Griddly

Release 1.6.7

Chris Bamford

Mar 16, 2023

ABOUT

Griddly documentation.

CHAPTER

ONE

INTRODUCTION

One of the most important things about AI research is data. In many Game Environments the rate of data (rendered frames per second, or state representations per second) is relatively slow meaning very long training times. Researchers can compensate for this problem by parallelizing the number of games being played, sometimes on expensive hardware and sometimes on several servers requiring network infrastructure to pass states to the actual learning algorithms. For many researchers and hobbyists who want to learn AI, this approach is unobtainable and accessible only for the research teams with lots of funding and engineers supporting the hardware and infrastructure required.

Griddly provides a solution to this issue.

Griddly is an open-source project aimed to be a all-encompassing platform for grid-world based research. Griddly provides a highly optimized game state and rendering engine with a flexible high-level interface for configuring environments. Not only does Griddly offer simple interfaces for single, multi-player and RTS games, but also multiple methods of rendering, configurable partial observability and interfaces for procedural content generation.

Here are some of the highlighted features:

1.1 Flexibility

Griddly games are defined using a simple configuration language GDY in which you can configure the number of players, how inputs are converted into game mechanics, the objects and how they are rendered and what design of the levels.

Read more about *GDY here*

1.2 Speed + Memory Usage

The Griddly engine is written entirely in c++ and and uses the Vulkan API to render observational states. This means that all the games have significantly faster frame rates. Griddly also offers lightweight vectorized state rendering, which can render games states at 30k+ FPS in some games.

1.3 Pre-Defined Games

Visit the *games section* here to see which games are currently available. Several games have been ported from the GVGAI and MiniGrid RL environments, which can now be run at significantly higher speeds and less memory overhead.

Note: More games are being added as Griddly is being developed. Feel free to design your own games and let the discord community see what your have built!

1.4 OpenAI Gym Interface

Griddly provides an open ai gym interface out-of-the-box which wraps the underlying raw API making Reinforcement Learning research significantly easier.

CHAPTER

TWO

COMMUNITY

Come join the Griddly Discord community, get support and share game levels that you have created.

Griddly is written and maintained by Chris Bamford.

Twitter: @Bam4d Github: Bam4d

Note: Please help me :D

CHAPTER

THREE

HALL OF FAME

If you create a project that uses Griddly, please let us know and we will link it here. This includes if you use Griddly in any papers, use the Griddly engine in another game project and want to share your work.

Note: You can Be the first!

3.1 Academia

Please use the following snippet to reference the Griddly project:

```
@misc{bamford2020griddly,
   title={Griddly: A platform for AI research in games},
   author={Chris Bamford and Shengyi Huang and Simon Lucas},
   year={2020},
   eprint={2011.06363},
   archivePrefix={arXiv},
   primaryClass={cs.AI}
}
```

CHAPTER

FOUR

ARTWORK

The Artwork is provided by the Oryx Design Lab.

You may not use these images in any product without purchasing a license.

CHAPTER FIVE

FREQUENTLY ASKED QUESTIONS

Nothing here yet!

CHAPTER

SIX

INSTALLATION

6.1 Python

Griddly supports versions of python 3.6+.

Warning: some features of OpenAI gym do not work with python 3.8 (using pygame for playing gym environments for example)

On most platforms Griddly can be easily installed using:

`pip install griddly`

Virtual environments such as conda are highly recommended to make sure the dependencies of projects using Griddly do not interfere with your other projects.

To create a conda environment with Griddly installed:

```
conda create --name griddly python=3.7
conda activate griddly
pip install griddly
```

6.2 Prerequisites

Griddly uses Vulkan to render environments. Most modern hardware will support vulkan and the required libraries should be pre-installed on your system.

If you are using docker, you can find images with vulkan pre-installed which may be helpful.

6.3 Other Languages

There is no support currently for languages other than python. A java version may be supported in the future.

CHAPTER

SEVEN

OPENAI GYM INTERFACE

Games defined with *GDY* files can easily be wrapped by OpenAI's gym interface.

The simplest way to use a pre-made environment is to just use the following code:

```
import gym
import griddly

if __name__ == '__main__':

    env = gym.make('GDY-Sokoban-v0')
    env.reset()
```

The `gym.make` command can also send several Griddly specific parameters to the environment:

```
env = gym.make(
    'GDY-Sokoban-v0'
    player_observer_type=gd.ObserverType.VECTOR,
    global_observer_type=gd.ObserverType.VECTOR,
    level=0,
    max_steps=None,
)
```

See also:

More examples and a full listing of all the games can be found on the page for each game in *Games*

7.1 Registering Your Own Environment

It's easy to register your own GDY files and wrap them with OpenAI gym using the `GymWrapperFactory`:

```
import gym
from griddly import GymWrapperFactory, gd

if __name__ == '__main__':
    wrapper = GymWrapperFactory()

    wrapper.build_gym_from_yaml('MyNewEnvironment', 'my_new_env_gdy.yaml')

    env = gym.make('GDY-MyNewEnvironment-v0')
    env.reset()
```

7.2 Observer Types

When generating an environment you can specify how you want the environment to be rendered. You can do this by setting the `player_observer_type` and `global_observer_type` parameters in the `gym.make` function, or the `build_gym_from_yaml` function.

See also:

For more information about observation spaces, states and event history see *Observation Spaces*

7.3 The Global Observer

The global observer can be used alongside any of the other observers and will always render the entire environment regardless of how other observers are defined. This means that you can pass vector observations to your agents and then render with sprites or blocks to make awesome demos!

```
env = gym.make(f'GDY-Sokoban-Adv-v0', global_observer_type=gd.ObserverType.SPRITE_2D)
env.reset()

env.render(observer='global')
```

CHAPTER

EIGHT

GRIDDLY DESCRIPTION YAML

Griddly Description YAML (GDY) is is the description language the Griddly uses to create environments and configure how to control the objects within it.

GDY files are typically split into 3 parts:

- **Environment** - Define levels, players, action mappings..
- **Actions** - Define the game mechanics
- **Objects** - Define the different objects in the environment and their properties

A GDY file looks like this:

```yaml
Version: "0.1"
Environment:
  Name: sokoban
  Observers:
    Sprite2D:
      TileSize: 24
      BackgroundTile: gvgai/newset/floor2.png
  Player:
    Count: 1 # This is only a single player game
    AvatarObject: avatar # The player can only control a single avatar in the game
  Termination:
    Win:
      - eq: [box:count, 0] # If there are no boxes left
  Levels:
    - |
      wwwwwww
      w..hA.w
      w.whw.w
      w...b.w
      whbb.ww
      w..wwww
      wwwwwww
    - |
      wwwwwwwww
      ww.h....w
      ww...bA.w
      w....w..w
      wwwbw...w
      www...w.w
      wwwh....w
```

(continues on next page)

(continued from previous page)

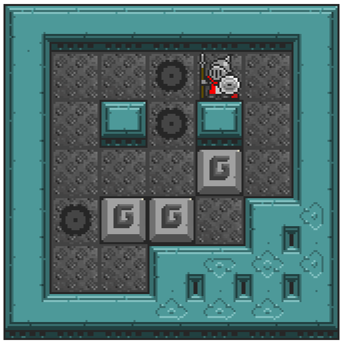
```
      wwwwwwwwww

Actions:
# Define the move action
- Name: move
  Behaviours:
    # The agent can move around freely in empty space and over holes
    - Src:
        Object: avatar
        Commands:
          - mov: _dest
      Dst:
        Object: [_empty, hole]

    # Boxes can move into empty space
    - Src:
        Object: box
        Commands:
            - mov: _dest
      Dst:
        Object: _empty

    # The agent can push boxes
    - Src:
        Object: avatar
        Commands:
          - mov: _dest
      Dst:
        Object: box
        Commands:
          - cascade: _dest

    # If a box is moved into a hole remove it
    - Src:
        Object: box
        Commands:
          - remove: true
          - reward: 1
      Dst:
        Object: hole

Objects:
- Name: box
  Z: 2
  MapCharacter: b
  Observers:
    Sprite2D:
      - Image: gvgai/newset/block1.png

- Name: wall
  MapCharacter: w
  Observers:
```

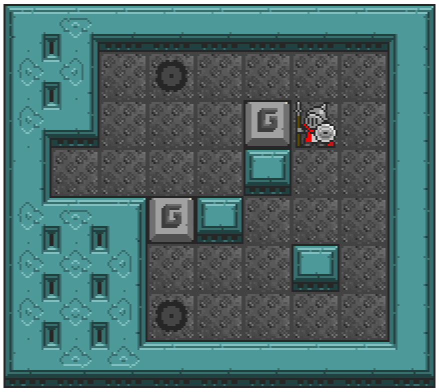
(continues on next page)

(continued from previous page)

```
    Sprite2D:
      - TilingMode:
          WALL_16
        Image:
          - gvgai/oryx/wall3_0.png
          - gvgai/oryx/wall3_1.png
          - gvgai/oryx/wall3_2.png
          - gvgai/oryx/wall3_3.png
          - gvgai/oryx/wall3_4.png
          - gvgai/oryx/wall3_5.png
          - gvgai/oryx/wall3_6.png
          - gvgai/oryx/wall3_7.png
          - gvgai/oryx/wall3_8.png
          - gvgai/oryx/wall3_9.png
          - gvgai/oryx/wall3_10.png
          - gvgai/oryx/wall3_11.png
          - gvgai/oryx/wall3_12.png
          - gvgai/oryx/wall3_13.png
          - gvgai/oryx/wall3_14.png
          - gvgai/oryx/wall3_15.png

- Name: hole
  Z: 1
  MapCharacter: h
  Observers:
    Sprite2D:
    - Image: gvgai/oryx/cspell4.png

- Name: avatar
  Z: 2
  MapCharacter: A
  Observers:
    Sprite2D:
    - Image: gvgai/oryx/knight1.png
```

This is all thats needed to define a game of Sokoban that has two levels which look like this:





8.1 GDY Schema

To make GDY slightly less scary, there is a GDY json schema which can be integrated with most IDEs to help write GDY files.

You can *Find the tutorial here*

CHAPTER NINE

ACTION SPACES

9.1 Overview

Griddly provides a common interface for action spaces in python which can be access using:

```
env = gym.make('GDY-[your game here]-v0')

# This contains a description of the action space
env.action_space
```

All actions follow the following format:

```
action = [

    # (Only required if there is no avatar)
    x, # X coordinate of action to perform.
    y, # Y coordinate of action to perform.

    # (Only required if there is more than one action type)
    action_type, # The type of action to perform (move, gather, attack etc....,

    # (Always required)
    action_id, # The ID of the action (These are defined by InputMapping in GDY)
]

env.step(action)
```

All values in this array are integers.

x, y These coordinates are required when the environment does not specify that there is an avatar to control. The coordinates chosen become the location of the action that will be performed.

For example in a game like chess, or checkers, the coordinates would correspond to the piece that the player wants to move.

action_type The action type refers to the index of the action type as defined in the GDY. For example *move*, *gather*, *push* etc…

A list of the registered (and correctly ordered for use in actions) types can be found using `env.gdy.get_action_names()`.

action_id The action id is commonly used for the "direction" component of the action. The action_id directly corresponds to the `InputMapping` of the action.

Note: if no `InputMapping` is set for an action, a default of 4 action ids is applied. These action ids resolve to "UP", "DOWN", "LEFT" and "RIGHT"

Note: All action types include action_id *0* which corresponds to a no-op

9.1.1 Sampling

Sampling the action space is the same as any other environment:

env.action_space.sample() This will always produce the correct format of actions for the environment that is loaded.

9.1.2 Sampling Valid Actions

In many environment, certain actions may have no effects at all, for example moving an avatar into an immovable object such as a wall. Or attacking a tile that has no objects.

Griddly provides some helper methods for reducing the action spaces to only sample valid actions and produce masks for calculating valid policies

env.game.get_available_actions(player_id) Returns a dict of locations of objects that can be controlled and the actions that can be used at those locations

Warning: player_id=0 is reserved for NPCs and internal actions

env.game.get_available_action_ids(location, action_names) Returns a dict of available action_ids at the given location for the given action_names.

ValidActionSpaceWrapper

In order to easily support games with large action spaces such as RTS games, several helper functions are included a wrapper `ValidActionSpaceWrapper`. The `ValidActionSpaceWrapper` has two functions:

- Sampling actions using this wrapper only returns valid actions in the environment.
- Two helper functions are available to create action masks which can be applied during neural network training to force the network to choose only valid actions.

env.get_unit_location_mask(player_id, mask_type='full') Returns a mask of all the locations in the grid which can be selected by a particular player.

If `mask_type == 'full'` then a mask of dimensions (grid_height, grid_width) is returned. This mask can be used in the case where a one-hot representation of the entire grid is used for location selection.

If `mask_type == 'reduced'` then two masks are returned. One for `grid_height` and one for `grid_width`. This mask can be used when two separate one-hot representations are used for `x` and `y` selection.

Warning: player_id=0 is reserved for NPCs and internal actions

env.get_unit_action_mask(location, action_names, padded=True) Returns a mask for the `action_type` and and `action_id`

If `padded == True` all masks will be returned with the length padded to the size of the largest number of action ids across all the actions.

If `padded == False` all masks are returned with the length of the number of action ids per action.

```
env.reset() # Wrapper must be applied after the reset

env = ValidActionSpaceWrapper(env)

unit_location_mask = env.get_unit_location_mask(player_id, mask_type='full')
unit_action_mask = env.get_unit_action_mask(location, action_names, padded=True)
```

See also:

A Closer Look at Action Masking in Policy Gradient Algorithms: https://arxiv.org/abs/2006.14171

Valid Action Trees

Valid action trees can be used to construct Conditional Action Trees, which can be used to iteratively apply masks to complex action spaces depending on the previous actions selected.

env.game.build_valid_action_trees() Returns a valid action tree for the current state for each player.

See also:

You can find several examples of Conditional Action Trees being used with Griddly and RLLib here: https://github.com/Bam4d/conditional-action-trees

9.2 Examples

In this section we break down some example action spaces. In all Griddly environments, `env.action_space.sample()` can be used to see what valid action spaces look like.

Here are some explanations of valid actions in different environments are and how to use them.

9.2.1 Single Player

Single Action Type

If the environment has a single action type then only the `action_id` needs to be sent to `env.step`.

This is usually the case in environments where there is an avatar that can only be moved and there are no special actions defined like `attack` or `pick_up`.

Assuming that our only `action_type` in the environment is `move` then the following code can be used to move the avatar in a particular direction:

```
# env.step(action_id)
# OR env.step([action_id])

env.step(3) # Move the avatar right
env.step(1) # Move the avatar left
```

Multiple Action Types

In the case where there may be a more complicated action space, for example if there is an avatar that can "move", but also "attack" in any direction around it, the `action_type` and `action_id` must both be supplied.

For example:

```
# env.step([action_type, action_id])

env.step([0, 3]) # Move the avatar right
env.step([1, 1]) # Attack to the left of the avatar
```

9.2.2 Multi-Agent

Multiple Player Actions

In multi-agent environments, `env.step` expects a list of actions for all players. To send actions to individual players in a call to `env.step`, set `action_id = 0` for any of the players that are not performing an action.

for example:

```
env.step([
  1, # Action for player 1
  0 # Action for player 2 (which is a no-op)
])
```

Single Action Type

If there is only a single action type available, a list of `action_id` values can be sent directly to `env.step`

```
env.step([
  1, # Action for player 1
  2 # Action for player 2
])
```

Multiple Action Types

If there are multiple action types available, `env.step` must contain a list of values for each player giving the `action_type` and `action_id`:

Given that there are two action types "move" and "attack" and each action type has default `InputMapping`, the following code can be used to send "move left" to player 1 and "attack forward" to player 2.

```
env.step([
  [0, 1], # Action for player 1 (move left)
  [1, 2]  # Action for player 2 (attack forward)
])
```

9.2.3 Real Time Strategy (RTS)

Multiple players, Multiple Action Types, Action Coordinates

In RTS games, multiple actions for multiple players can be performed in single time-steps.

Lets say our RTS game has units that have an action `move` and an action `gather` (to gather resources). Leta also say that there are three units for each player. We can control them in one call to `env.step`.

```
# env.step([
#   [ # List of actions for player 1
#     [x1, y1, action_type1, action_id1],
#     [x2, y2, action_type2, action_id2],
#     ...
#   ],
#   [ # List of actions for player 2
#     [x1, y1, action_type1, action_id1],
#     [x2, y2, action_type2, action_id2],
#     ..
#   ],
# ])

env.step([
  # Player 1
  [
    [3, 10, 0, 3], # Move the unit at [3,10] right
    [4, 7, 1, 1], # The unit at [4,7] will gather resources in front of it
    [4, 4, 0, 0] # The unit at [4, 4] will do nothing. (this can also be ommitted with␣
↪the same effect)
  ],

  # Player 2
  [
    [10, 4, 1, 3], # The unit at [10,4] will gather resources to the right
    [13, 2, 1, 1] # The unit at [13,2] will gather resources to the left
  ]
])
```

CHAPTER

TEN

OBSERVATION SPACES

10.1 Overview

Observation spaces in Griddly are highly configurable. In addition to providing pixel-based and vector-based states of environments, Griddly also provides methods of accessing semantic information about the game state itself, such as state data and event history. For pixel and vector-based representations Griddly provides different **observers**.

10.1.1 What is an Observer?

An **observer** in Griddly converts the state of an environment to something that is consumable by a machine learning algorithm. It effectively *creates* the observations. In Griddly there are many different options for creating observations. This page will outline how all of these methods can be used.

Every environment in Griddly has at least two configurable observers; the **player** observer(s) and the **global** observer.

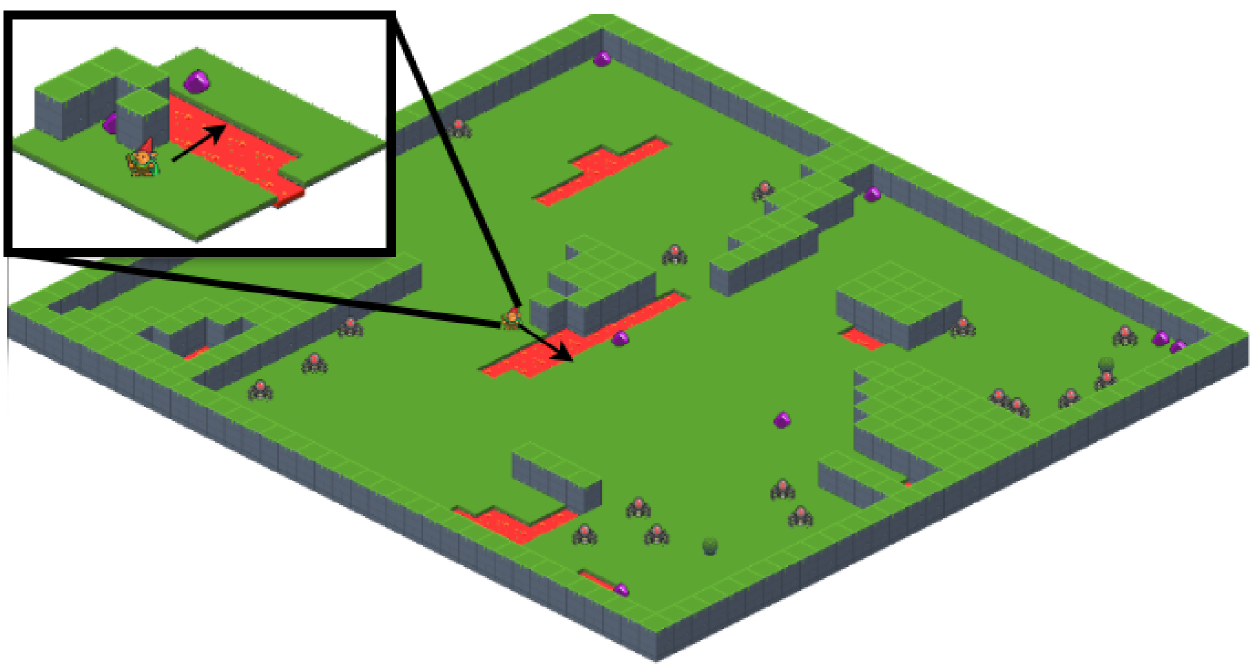


Fig. 1: The "Spider Nest" environment with an isometric player observer (inset) and isometric global observer configured.

The **player** observer(s) are what the agent in the environment *sees*. The agent might be able to see only a small area around itself (this is known as egocentric partial observability), or it might be able to see the entire environment.

The **global** observer always gives a view of the entire environment regardless of the number of players and and size of the environment.

Both observers have a standard OpenAI Gym observation space which can be accessed with `env.observation_space` describing the player observation space and `env.global_observation_space` describing the global observation space.

10.1.2 Single Agent Observations

The observations for environments where a single avatar is being controlled are returned in the following way:

```
obs = env.reset()

# obs = np.array([ ... ]) # Player observation

obs, reward, done, info = env.step( ... )

# obs = np.array([ ... ]) # Player observation
```

10.1.3 Multi-Agent Observations

When there are multiple agents in an environment the `env.reset(...)` and `env.step(...)` functions will return the observations of all the of agents as an array of `np.array` objects. Each observation in the array will be consistent with the shape of `env.observation_shape`.

As an example in an environment with 2 players, the result of `env.reset(...)` and `env.step(...)` will be:

```
obs = env.reset()

#  obs = [
#    np.array([ ... ]), # Player 1 observation
#    np.array([ ... ]) # Player 2 observation
# ]

obs, reward, done, info = env.step([ ... ])

#  obs = [
#    np.array([ ... ]), # Player 1 observation
#    np.array([ ... ]) # Player 2 observation
# ]
```

The global observer can also be returned in the `env.reset()` function by setting the `global_observations` parameter to `True`. In this case a dictionary is returned with the `global` and `player` keys for both observation types.

```
obs = env.reset(global_observations=True)

#  obs = {
#      'global': np.array([ ... ]), # Global observation
#      'player': [
#        np.array([ ... ]), # Player 1 observation
```

(continues on next page)

(continued from previous page)

```
#         np.array([ ... ]) # Player 2 observation
#      ]
# }
```

10.2 Pixels

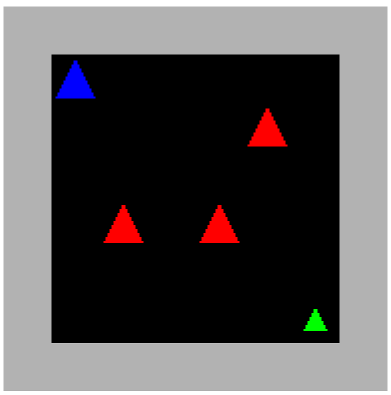
Note: For pixel-based observers, each object must define how it should be rendered with an *Observer Configuration*

The observer type can be chosen when creating the environment with `gym.make()`

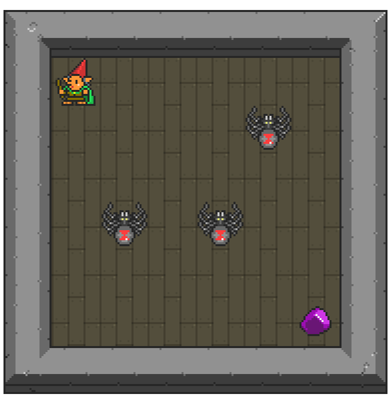
```
env = gym.make(
  'GDY-MyEnvironmentName-v0',
  player_observer_type=XXX,
  global_observer_type=YYY
  ...
)
```

The options for both the `player_observer_type` and `global_observer_type` are given below.

gd.ObserverType.BLOCK_2D Renders sprites defined by the GDY object *Block2D* descriptions.



gd.ObserverType.SPRITE_2D Renders sprites defined by the GDY object *Sprite2D* descriptions.



gd.ObserverType.ISOMETRIC Renders sprites defined by the GDY object *Isometric* descriptions.

10.3 Vector

Vector observers will return a tensor of shape [*objects, player ids, object_rotation, variables*, *width*, *height*] where each value is either 0 or 1 denoting that there is an object of that type in a particular location.

The data contained in the cell can be configured using the vector options in the GDY observer configuration.

Objects

Each cell always contains a multi-label representation of whether an object is present (1) in that cell or not (0).

The order of the object index in each [x,y] location can be retrieved by calling `env.game.get_object_names()`.

IncludePlayerId If this option is set, each cell of the observation tensor also contains a one-hot representation of which player an object belongs to.

> **Warning:** In multi-agent scenarios, every agent sees themselves as player 1.

IncludeRotation This option appends a one-hot to the cell representing the rotation of the object at that position.

IncludeVariables If set, the local variables of each object are provided. The order of the variables can be retrieved by calling `env.game.get_object_variable_names()`

As an example, in an 5x5 environment that has three types of object: *avatar*, *wall* and *goal* and no other options are set:

```
obs_shape = env.observation_space.shape

# obs_shape == (3,5,5)

obs, reward, done, info = env.step( ... )

# obs = [
  [ # avatar in these locations
    [0,0,0,0,0],
    [0,1,0,0,0],
    [0,0,0,0,0],
    [0,0,0,0,0],
    [0,0,0,0,0]
  ],
  [ # wall in these locations
    [1,1,1,1,1],
```

(continues on next page)

(continued from previous page)

```
    [1,0,0,0,1],
    [1,0,0,0,1],
    [1,0,0,0,1],
    [1,1,1,1,1]
  ],
  [ # goal in these locations
    [0,0,0,0,0],
    [0,0,0,0,0],
    [0,0,0,0,0],
    [0,0,0,1,0],
    [0,0,0,0,0]
  ]
]
```

10.4 Semantic State

A breakdown of the entire environment including internal variable values that the objects may have can be recovered using `env.get_state()`.

env.get_state() This function will return data in the following format:

```
{
  'GameTicks': 1,
  'GlobalVariables': {},
  'Objects': [
    {
      'Name': 'avatar',
      'Location': [1, 3],
      'Orientation': 'NONE',
      'PlayerId': 1,
      'Variables': {
        '_y': 3,
        '_playerId': 1,
        '_x': 1
      }
    },
    {
      'Name': 'goal',
      'Location': [1, 3],
      'Orientation': 'NONE',
      'PlayerId': 1,
      'Variables': {
        '_y': 3,
        '_playerId': 1,
        '_x': 1
      }
    },
    {

      ...
```

(continues on next page)

(continued from previous page)

```
    }
  ]
}
```

10.5 Event History

Event history in Griddly contains all the information about actions that have been made by the player, any internal actions that have been executed by any game mechanics, or any delayed actions that have been performed.

Event history is gathered during `env.step()` and returned in the `info['History']` variable, but it must be enabled first.

To enable event history, `env.enable_history(True)` can be called after `gym.make()`

The format of event history looks like this:

```
[
  {
    'PlayerId': 1,
    'ActionName': 'move',
    'Tick': 0,
    'Reward': 0,
    'Delay': 0,
    'SourceObjectName': 'avatar',
    'DestinationObjectName': '_empty',
    'SourceObjectPlayerId': 1,
    'DestinationObjectPlayerId': 0,
    'SourceLocation': [2.0, 3.0],
    'DestinationLocation': [1.0, 3.0]
  },
  {
    'PlayerId': 1,
    'ActionName': 'move',
    'Tick': 0,
    'Reward': 0,
    'Delay': 0,
    'SourceObjectName': 'ball',
    'DestinationObjectName': '_empty',
    'SourceObjectPlayerId': 1,
    'DestinationObjectPlayerId': 0,
    'SourceLocation': [1.0, 3.0],
    'DestinationLocation': [0.0, 3.0]
  },
  ...
]
```

CHAPTER

ELEVEN

VISUALIZATION

To make it easy for you to create high quality and interesting demonstrations of AIs, or be able to analyse behaviour of trained agents, Griddly provides many tools to make this easy.

11.1 Live Rendering

OpenAI gym allows gym environments to be rendered using the `env.render()` function. However in many Griddly environments you may have a choice between different players and global observations to render.

11.1.1 Rendering Different Observers

To render global observations you simply need to add the parameter `observer='global'` to your render function. Additionally if you want to render a particular player you can use `observer=P` where P is the 0-indexed player id.

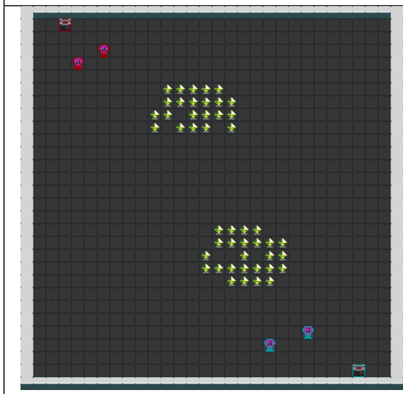
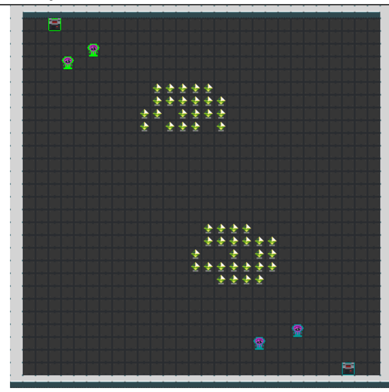
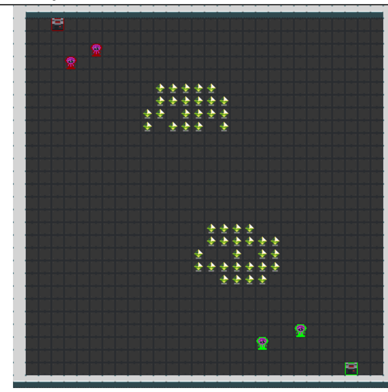
```
env.render(observer='global') # Will render the global observer

env.render(observer=0) # Will render the observer for player 1

env.render(observer=1) # Will render the observer for player 2
```

Player Highlighting

In games with multiple players, Griddly will highlight all the objects in the environment based on which observer is being used.

- The global observer will highlight player owned objects with colours which are automatically picked
- The player observer will highlight objects belonging to themselves in green

Table 1: Observer Highlighting

Global	Player 1	Player 2
		

11.2 Saving Images

Griddly includes a tool `RenderToFile` that can directly save observations to disk as png files.

```
from griddly.RenderTools import RenderToFile

render_to_file = RenderToFile()

...

visualization = env.render(observer=.., mode='rgb_array') # Get the observation as an array

render_to_file.render(visualization, 'my_observation.png') # save the image to disk
```

11.3 Saving Videos

Also saving videos with Griddly is simple using the `VideoRecorder`.

Note: you will need to make sure ffmpeg is installed in your system.

```
from griddly.RenderTools import VideoRecorder

video_recorder = VideoRecorder()

...

# Start the video recording
observation = env.reset()
video_recorder.start("video_test.mp4", env.observation_space.shape)

...
```

(continues on next page)

(continued from previous page)

```
# Step the environment and record the next frame
obs, reward, done, env = env.step( ... )
video_recorder.add_frame(obs)

...

# Clean up
video_recorder.close()
```

CHAPTER

TWELVE

PROCEDURAL CONTENT GENERATION

Reinforcement learning can be prone to over-fitting in environments where the initial conditions are limited and the environment dynamics are deterministic. Procedural content generation is an important tool in Reinforcement learning, as it allows level maps to be created on-the-fly. This gives the agent a much more complex challenge, and stops it from being able to overfit on a small dataset of levels.

12.1 Level Maps

Levels in Griddly environments are defined by strings of characters. The `MapCharacter` used are defined in the GDY files of the game. These `MapCharacter` can be found in the GDY files or in the game's documentation.

12.1.1 Basic Map

```
W   W   W   W   W   W
W   A   .   .   .   W
W   .   .   .   .   W
W   .   .   .   .   W
W   .   .   .   g   W
W   W   W   W   W   W
```

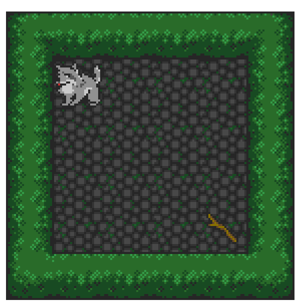


Fig. 1: How the above Doggo level is rendered.

You can see in this map example above that the `A` character defines the Dog and the `g` character defines the goal. `W` defines the walls and `.` is reserved for empty space.

This is a basic example and generating levels for this environment might not be too interesting...

12.2 Clusters Level Generator

A much more complicated example would be to use the *Clusters* game and generate new levels. The aim of the Clusters game is for the agent to push coloured blocks together to form "clusters", whilst avoiding spikes. The game is fully deterministic and there are only 5 levels supplied in the original GDY file. This makes it a perfect candidate for building new levels and testing if Reinforcement Learning can still solve these levels!

12.2.1 Level Generator Class

Here's an example of a level generator for the cluster's game. Levels are generated with simple configurable heuristics such as maximum number of each coloured boxes and maximum numbers of spikes. The boxes and spikes are randomly placed in the grid to create the initial game layout. The agent is also added to the grid in a random position.

The `LevelGenerator` class can be used as a base class. Only the `generate` function needs to be implemented.

```python
class ClustersLevelGenerator(LevelGenerator):
    BLUE_BLOCK = 'a'
    BLUE_BOX = '1'
    RED_BLOCK = 'b'
    RED_BOX = '2'
    GREEN_BLOCK = 'c'
    GREEN_BOX = '3'

    AGENT = 'A'

    WALL = 'w'
    SPIKES = 'h'

    def __init__(self, config):
        super().__init__(config)
        self._width = config.get('width', 10)
        self._height = config.get('height', 10)
        self._p_red = config.get('p_red', 1.0)
        self._p_green = config.get('p_green', 1.0)
        self._p_blue = config.get('p_blue', 1.0)
        self._m_red = config.get('m_red', 5)
        self._m_blue = config.get('m_blue', 5)
        self._m_green = config.get('m_green', 5)
        self._m_spike = config.get('m_spike', 5)

    def _place_walls(self, map):

        # top/bottom wall
        wall_y = np.array([0, self._height - 1])
        map[:, wall_y] = ClustersLevelGenerator.WALL

        # left/right wall
        wall_x = np.array([0, self._width - 1])
        map[wall_x, :] = ClustersLevelGenerator.WALL

        return map
```

(continues on next page)

(continued from previous page)

```
    def _place_blocks_and_boxes(self, map, possible_locations, p, block_char, box_char,␣
↪max_boxes):
        if np.random.random() < p:
            block_location_idx = np.random.choice(len(possible_locations))
            block_location = possible_locations[block_location_idx]
            del possible_locations[block_location_idx]
            map[block_location[0], block_location[1]] = block_char

            num_boxes = 1 + np.random.choice(max_boxes - 1)
            for k in range(num_boxes):
                box_location_idx = np.random.choice(len(possible_locations))
                box_location = possible_locations[box_location_idx]
                del possible_locations[box_location_idx]
                map[box_location[0], box_location[1]] = box_char

        return map, possible_locations

    def generate(self):
        map = np.chararray((self._width, self._height), itemsize=2)
        map[:] = '.'

        # Generate walls
        map = self._place_walls(map)

        # all possible locations
        possible_locations = []
        for w in range(1, self._width - 1):
            for h in range(1, self._height - 1):
                possible_locations.append([w, h])

        # Place Red
        map, possible_locations = self._place_blocks_and_boxes(
            map,
            possible_locations,
            self._p_red,
            ClustersLevelGenerator.RED_BLOCK,
            ClustersLevelGenerator.RED_BOX,
            self._m_red
        )

        # Place Blue
        map, possible_locations = self._place_blocks_and_boxes(
            map,
            possible_locations,
            self._p_blue,
            ClustersLevelGenerator.BLUE_BLOCK,
            ClustersLevelGenerator.BLUE_BOX,
            self._m_blue
        )

        # Place Green
        map, possible_locations = self._place_blocks_and_boxes(
```

(continues on next page)

(continued from previous page)

```
            map,
            possible_locations,
            self._p_green,
            ClustersLevelGenerator.GREEN_BLOCK,
            ClustersLevelGenerator.GREEN_BOX,
            self._m_green
        )

        # Place Spikes
        num_spikes = np.random.choice(self._m_spike)
        for k in range(num_spikes):
            spike_location_idx = np.random.choice(len(possible_locations))
            spike_location = possible_locations[spike_location_idx]
            del possible_locations[spike_location_idx]
            map[spike_location[0], spike_location[1]] = ClustersLevelGenerator.SPIKES

        # Place Agent
        agent_location_idx = np.random.choice(len(possible_locations))
        agent_location = possible_locations[agent_location_idx]
        map[agent_location[0], agent_location[1]] = ClustersLevelGenerator.AGENT

        level_string = ''
        for h in range(0, self._height):
            for w in range(0, self._width):
                level_string += map[w, h].decode().ljust(4)
            level_string += '\n'

        return level_string
```

This generates levels like the following:

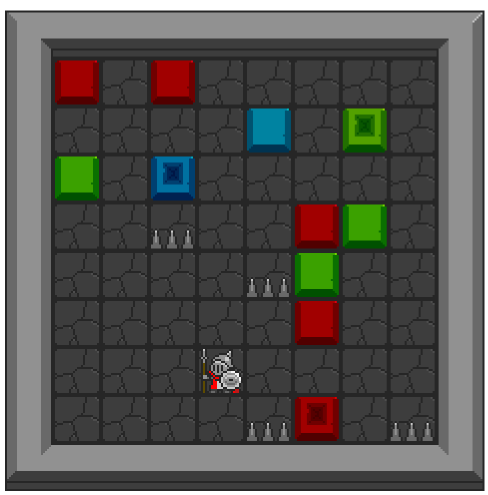


Fig. 2: A 10x10 map generated by the above code.

12.2.2 Using LevelGenerator

In the most simple case, the level generator can be used just before the level resets and the generated string can be passed to `env.reset(level_string=...)`

```
if __name__ == '__main__':

    config = {
        'width': 10,
        'height': 10
    }

    renderer = RenderToFile()

    level_generator = ClustersLevelGenerator(config)

    env = gym.make('GDY-Clusters-v0')
    env.reset(level_string=level_generator.generate())

    ...
```

12.2.3 Using LevelGenerators with RLLib

The `LevelGenerator` base class is compatible with RLLib and can be used and configured through the standard RLLib configuration.

For example, the level generator and its parameters can be set up in the `env_config` in the following way:

```
'config': {

    ...

    'env_config': {
        'generate_valid_action_trees': True,
        'level_generator': {
            'class': ClustersLevelGenerator,
            'config': {
                'width': 6,
                'height': 6,
                'p_red': 0.7,
                'p_green': 0.7,
                'p_blue': 0.7,
                'm_red': 4,
                'm_blue': 4,
                'm_green': 4,
                'm_spike': 4
            }
    },

    ...
}
```

CHAPTER

THIRTEEN

SINGLE-PLAYER

13.1 Spiders

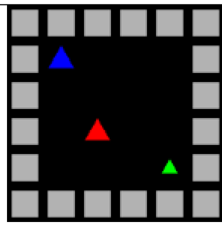
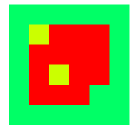
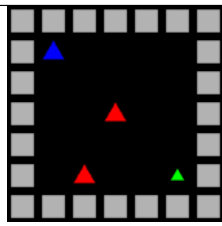
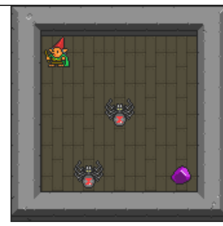
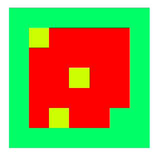
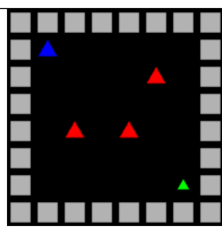
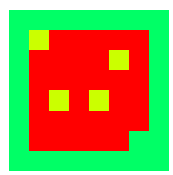
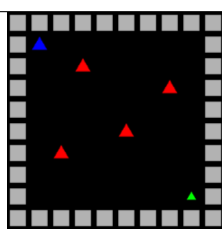
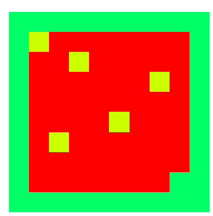
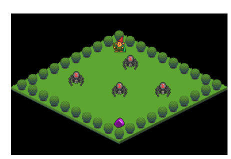
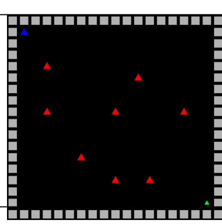
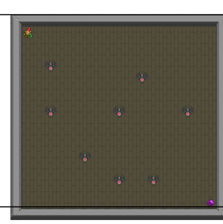
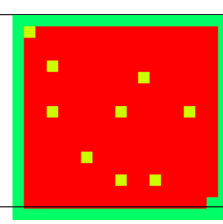
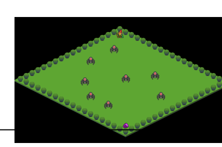
```
Single-Player/Mini-Grid/minigrid-spiders.yaml
```

13.1.1 Description

A port of the games provided in the https://github.com/maximecb/gym-minigrid Dynamic obstacles environment, but you're a gnome avoiding ghosts to get to a gem.

13.1.2 Levels

Table 1: Levels

	Block2D	Sprite2D	Vector	Isometric
Level ID: 0 Size: 6x6				
Level ID: 1 Size: 7x7				
Level ID: 2 Size: 8x8				
Level ID: 3 Size: 10x10				
Level ID: 4 Size: 19x18				

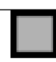
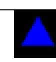
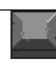
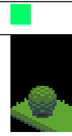
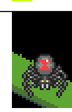
13.1.3 Code Example

The most basic way to create a Griddly Gym Environment. Defaults to level 0 and SPRITE_2D rendering.

```python
import gym
import griddly

if __name__ == '__main__':

    env = gym.make('GDY-Spiders-v0')
    env.reset()

    # Replace with your own control algorithm!
    for s in range(1000):
        obs, reward, done, info = env.step(env.action_space.sample())
        env.render() # Renders the environment from the perspective of a single player

        env.render(observer='global') # Renders the entire environment

        if done:
            env.reset()
```

13.1.4 Objects

Table 2: Tiles

Name ->	wall	spider	gem	gnome
Map Char ->	*W*	*G*	*g*	*A*
Block2D				
Sprite2D				
Vector				
Isometric				

13.1.5 Actions

move

Relative The actions are calculated relative to the object being controlled.

Action Id	Mapping
1	Rotate left
2	Move forwards
3	Rotate right
4	Move Backwards

random_movement

Relative The actions are calculated relative to the object being controlled.

Internal This action can only be called from other actions, not by the player.

Action Id	Mapping
1	Rotate left
2	Move forwards
3	Rotate right

13.1.6 YAML

```yaml
Version: "0.1"
Environment:
  Name: Spiders
  Description: A port of the games provided in the https://github.com/maximecb/gym-
↪minigrid Dynamic obstacles environment, but you're a gnome avoiding ghosts to get to a␣
↪gem.
  Observers:
    Sprite2D:
      TileSize: 24
      BackgroundTile: oryx/oryx_fantasy/floor2-2.png
    Isometric:
      TileSize: [32, 48]
      IsoTileHeight: 16
      IsoTileDepth: 4
      BackgroundTile: oryx/oryx_iso_dungeon/grass-1.png
    Block2D:
      TileSize: 24
  Player:
    Observer:
      RotateWithAvatar: true
      TrackAvatar: true
      Height: 7
      Width: 7
      OffsetX: 0
      OffsetY: 3
    AvatarObject: gnome
  Termination:
    Win:
      - eq: [gem:count, 0] # If there are no boxes left
    Lose:
      - eq: [gnome:count, 0] # If there are no boxes left
  Levels:
    - |
      W  W  W  W  W  W
      W  A  .  .  .  W
      W  .  .  .  .  W
      W  .  G  .  .  W
      W  .  .  .  g  W
      W  W  W  W  W  W
```

(continues on next page)

(continued from previous page)

```
    - |
      W  W  W  W  W  W  W
      W  A  .  .  .  .  W
      W  .  .  .  .  .  W
      W  .  .  G  .  .  W
      W  .  .  .  .  .  W
      W  .  G  .  .  g  W
      W  W  W  W  W  W  W
    - |
      W  W  W  W  W  W  W  W
      W  A  .  .  .  .  .  W
      W  .  .  .  .  G  .  W
      W  .  .  .  .  .  .  W
      W  .  G  .  G  .  .  W
      W  .  .  .  .  .  .  W
      W  .  .  .  .  .  g  W
      W  W  W  W  W  W  W  W
    - |
      W  W  W  W  W  W  W  W  W  W
      W  A  .  .  .  .  .  .  .  W
      W  .  .  G  .  .  .  .  .  W
      W  .  .  .  .  .  .  G  .  W
      W  .  .  .  .  .  .  .  .  W
      W  .  .  .  .  G  .  .  .  W
      W  .  G  .  .  .  .  .  .  W
      W  .  .  .  .  .  .  .  .  W
      W  .  .  .  .  .  .  .  g  W
      W  W  W  W  W  W  W  W  W  W
    - |
      W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W
      W  A  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  W
      W  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  W
      W  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  W
      W  .  .  G  .  .  .  .  .  .  .  .  .  .  .  .  .  .  W
      W  .  .  .  .  .  .  .  .  .  .  G  .  .  .  .  .  .  W
      W  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  W
      W  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  W
      W  .  .  G  .  .  .  .  .  G  .  .  .  .  .  G  .  .  W
      W  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  W
      W  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  W
      W  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  W
      W  .  .  .  .  .  G  .  .  .  .  .  .  .  .  .  .  .  W
      W  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  W
      W  .  .  .  .  .  .  .  .  G  .  .  G  .  .  .  .  .  W
      W  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  W
      W  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  g  W
      W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W

Actions:
  # Define action that cannot be controlled by the player. (In this case the spider␣
↪movement)
  - Name: random_movement
```

(continues on next page)

(continued from previous page)

```
  InputMapping:
    Inputs:
      1:
        Description: Rotate left
        OrientationVector: [-1, 0]
      2:
        Description: Move forwards
        OrientationVector: [0, -1]
        VectorToDest: [0, -1]
      3:
        Description: Rotate right
        OrientationVector: [1, 0]
    Relative: true
    Internal: true
  Behaviours:
    # Spider rotates on the spot
    - Src:
        Object: spider
        Commands:
          - rot: _dir
          - exec:
              Action: random_movement
              Delay: 3
              Randomize: true
      Dst:
        Object: spider

    # The gnome and the spider can move into empty space
    - Src:
        Object: spider
        Commands:
          - mov: _dest
          - exec:
              Action: random_movement
              Delay: 3
              Randomize: true
      Dst:
        Object: _empty

    # The spider will not move into the wall or the gem, but it needs to keep moving
    - Src:
        Object: spider
        Commands:
          - exec:
              Action: random_movement
              Delay: 3
              Randomize: true
      Dst:
        Object: [wall, gem]

    # If the gnome moves into a spider
    - Src:
```

(continues on next page)

(continued from previous page)

```
          Object: spider
        Dst:
          Object: gnome
          Commands:
            - remove: true
            - reward: -1

  # Define the move action
  - Name: move
    InputMapping:
      Inputs:
        1:
          Description: Rotate left
          OrientationVector: [-1, 0]
        2:
          Description: Move forwards
          OrientationVector: [0, -1]
          VectorToDest: [0, -1]
        3:
          Description: Rotate right
          OrientationVector: [1, 0]
        4:
          Description: Move Backwards
          VectorToDest: [0, 1]
          OrientationVector: [0, -1]
      Relative: true
    Behaviours:
      # Tell the gnome to rotate if it performs an action on itself (Rotate left and␣
↪Rotate right actions)
      - Src:
          Object: gnome
          Commands:
            - rot: _dir
        Dst:
          Object: gnome

      # If the gnome moves into a spider
      - Src:
          Object: gnome
          Commands:
            - remove: true
            - reward: -1
        Dst:
          Object: spider

      # The gnome and the spider can move into empty space
      - Src:
          Object: gnome
          Commands:
            - mov: _dest
        Dst:
          Object: _empty
```

(continues on next page)

(continued from previous page)

```
    # If the gnome moves into a gem object, the stick is removed, triggering a win␣
↪condition
    - Src:
        Object: gnome
        Commands:
          - reward: 1
      Dst:
        Object: gem
        Commands:
          - remove: true

Objects:
  - Name: wall
    MapCharacter: W
    Observers:
      Sprite2D:
        - TilingMode: WALL_16
          Image:
            - oryx/oryx_fantasy/wall2-0.png
            - oryx/oryx_fantasy/wall2-1.png
            - oryx/oryx_fantasy/wall2-2.png
            - oryx/oryx_fantasy/wall2-3.png
            - oryx/oryx_fantasy/wall2-4.png
            - oryx/oryx_fantasy/wall2-5.png
            - oryx/oryx_fantasy/wall2-6.png
            - oryx/oryx_fantasy/wall2-7.png
            - oryx/oryx_fantasy/wall2-8.png
            - oryx/oryx_fantasy/wall2-9.png
            - oryx/oryx_fantasy/wall2-10.png
            - oryx/oryx_fantasy/wall2-11.png
            - oryx/oryx_fantasy/wall2-12.png
            - oryx/oryx_fantasy/wall2-13.png
            - oryx/oryx_fantasy/wall2-14.png
            - oryx/oryx_fantasy/wall2-15.png
      Block2D:
        - Shape: square
          Color: [0.7, 0.7, 0.7]
          Scale: 1.0
      Isometric:
        - Image: oryx/oryx_iso_dungeon/bush-1.png

  - Name: spider
    InitialActions:
      - Action: random_movement
        Delay: 3
        Randomize: true
    MapCharacter: G
    Observers:
      Sprite2D:
        - Image: oryx/oryx_fantasy/avatars/spider1.png
      Block2D:
```

(continues on next page)

(continued from previous page)

```
      - Shape: triangle
        Color: [1.0, 0.0, 0.0]
        Scale: 0.8
    Isometric:
      - Image: oryx/oryx_iso_dungeon/avatars/spider-1.png

- Name: gem
  MapCharacter: g
  Observers:
    Sprite2D:
      - Image: oryx/oryx_fantasy/ore-6.png
    Block2D:
      - Shape: triangle
        Color: [0.0, 1.0, 0.0]
        Scale: 0.5
    Isometric:
      - Image: oryx/oryx_iso_dungeon/ore-6.png

- Name: gnome
  MapCharacter: A
  Observers:
    Sprite2D:
      - Image: oryx/oryx_fantasy/avatars/gnome1.png
    Block2D:
      - Shape: triangle
        Color: [0.0, 0.0, 1.0]
        Scale: 0.8
    Isometric:
      - Image: oryx/oryx_iso_dungeon/avatars/gnome-1.png
```

13.2 Eyeball

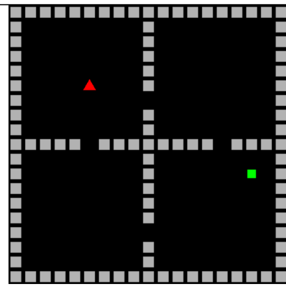
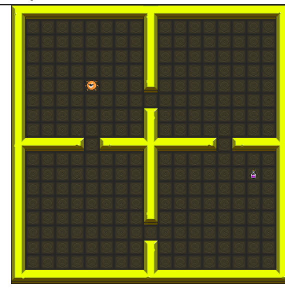
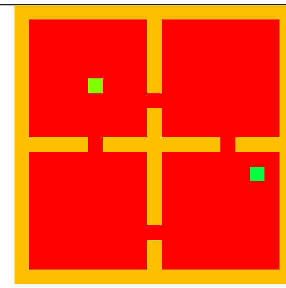
```
Single-Player/Mini-Grid/minigrid-eyeball.yaml
```

13.2.1 Description

A port of the games provided in the https://github.com/maximecb/gym-minigrid 4 Rooms environment, but you're a giant eye looking for it's eyedrops because everything is yellow and it hurts to look at.

13.2.2 Levels

Table 3: Levels

<table>
<tr><th></th><th>Block2D</th><th>Sprite2D</th><th>Vector</th></tr>
<tr><td><table><tr><td>Level ID</td><td>0</td></tr><tr><td>Size</td><td>19x19</td></tr></table></td><td></td><td></td><td></td></tr>
</table>

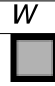
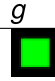
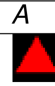
13.2.3 Code Example

The most basic way to create a Griddly Gym Environment. Defaults to level 0 and SPRITE_2D rendering.

```python
import gym
import griddly

if __name__ == '__main__':

    env = gym.make('GDY-Eyeball-v0')
    env.reset()

    # Replace with your own control algorithm!
    for s in range(1000):
        obs, reward, done, info = env.step(env.action_space.sample())
        env.render() # Renders the environment from the perspective of a single player

        env.render(observer='global') # Renders the entire environment

        if done:
            env.reset()
```

13.2.4 Objects

Table 4: Tiles

Name ->	wall	eye_drops	eyeball
Map Char ->	*W*	*g*	*A*
Block2D			
Sprite2D			
Vector			

13.2.5 Actions

move

Relative The actions are calculated relative to the object being controlled.

Action Id	Mapping
1	Rotate left
2	Move forwards
3	Rotate right

13.2.6 YAML

```yaml
Version: "0.1"
Environment:
  Name: Eyeball
  Description: A port of the games provided in the https://github.com/maximecb/gym-minigrid 4 Rooms environment, but you're a giant eye looking for it's eyedrops because everything is yellow and it hurts to look at.
  Observers:
    Sprite2D:
      TileSize: 24
      BackgroundTile: oryx/oryx_fantasy/floor7-1.png
  Player:
    Observer:
      RotateWithAvatar: true
      TrackAvatar: true
      Height: 7
      Width: 7
      OffsetX: 0
      OffsetY: 3
    AvatarObject: eyeball
  Termination:
    Win:
      - eq: [eye_drops:count, 0] # If there are no boxes left
  Levels:
    - |
      W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W
      W  .  .  .  .  .  .  .  .  W  .  .  .  .  .  .  .  .  W
      W  .  .  .  .  .  .  .  .  W  .  .  .  .  .  .  .  .  W
      W  .  .  .  .  .  .  .  .  W  .  .  .  .  .  .  .  .  W
      W  .  .  .  .  .  .  .  .  W  .  .  .  .  .  .  .  .  W
      W  .  .  .  .  A  .  .  .  W  .  .  .  .  .  .  .  .  W
      W  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  W
      W  .  .  .  .  .  .  .  .  W  .  .  .  .  .  .  .  .  W
      W  .  .  .  .  .  .  .  .  W  .  .  .  .  .  .  .  .  W
      W  W  W  W  W  .  W  W  W  W  W  W  W  W  .  W  W  W  W
      W  .  .  .  .  .  .  .  .  W  .  .  .  .  .  .  .  .  W
      W  .  .  .  .  .  .  .  .  W  .  .  .  .  .  .  g  .  W
      W  .  .  .  .  .  .  .  .  W  .  .  .  .  .  .  .  .  W
      W  .  .  .  .  .  .  .  .  W  .  .  .  .  .  .  .  .  W
```

(continues on next page)

(continued from previous page)

```
      W  .  .  .  .  .  .  .  .  W  .  .  .  .  .  .  .  .  W
      W  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  W
      W  .  .  .  .  .  .  .  .  W  .  .  .  .  .  .  .  .  W
      W  .  .  .  .  .  .  .  .  W  .  .  .  .  .  .  .  .  W
      W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W

Actions:
  # Define the move action
  - Name: move
    InputMapping:
      Inputs:
        1:
          Description: Rotate left
          OrientationVector: [-1, 0]
        2:
          Description: Move forwards
          OrientationVector: [0, -1]
          VectorToDest: [0, -1]
        3:
          Description: Rotate right
          OrientationVector: [1, 0]
      Relative: true
    Behaviours:
      # Tell the agent to rotate if the eyeball performs an action on itself
      - Src:
          Object: eyeball
          Commands:
            - rot: _dir
        Dst:
          Object: eyeball

      # The agent can move around freely in empty and always rotates the direction it is␣
↪travelling
      - Src:
          Object: eyeball
          Commands:
            - mov: _dest
        Dst:
          Object: _empty

      # If the eyeball moves into a eye_drops object, the eye_drops is removed,␣
↪triggering a win condition
      - Src:
          Object: eyeball
          Commands:
            - reward: 1
        Dst:
          Object: eye_drops
          Commands:
            - remove: true

Objects:
```

(continues on next page)

(continued from previous page)

```
  - Name: wall
    MapCharacter: W
    Observers:
      Sprite2D:
        - TilingMode: WALL_16
          Image:
            - oryx/oryx_fantasy/wall7-0.png
            - oryx/oryx_fantasy/wall7-1.png
            - oryx/oryx_fantasy/wall7-2.png
            - oryx/oryx_fantasy/wall7-3.png
            - oryx/oryx_fantasy/wall7-4.png
            - oryx/oryx_fantasy/wall7-5.png
            - oryx/oryx_fantasy/wall7-6.png
            - oryx/oryx_fantasy/wall7-7.png
            - oryx/oryx_fantasy/wall7-8.png
            - oryx/oryx_fantasy/wall7-9.png
            - oryx/oryx_fantasy/wall7-10.png
            - oryx/oryx_fantasy/wall7-11.png
            - oryx/oryx_fantasy/wall7-12.png
            - oryx/oryx_fantasy/wall7-13.png
            - oryx/oryx_fantasy/wall7-14.png
            - oryx/oryx_fantasy/wall7-15.png
      Block2D:
        - Shape: square
          Color: [0.7, 0.7, 0.7]
          Scale: 1.0

  - Name: eye_drops
    MapCharacter: g
    Observers:
      Sprite2D:
        - Image: oryx/oryx_fantasy/potion-1.png
      Block2D:
        - Shape: square
          Color: [0.0, 1.0, 0.0]
          Scale: 0.8

  - Name: eyeball
    MapCharacter: A
    Observers:
      Sprite2D:
        - Image: oryx/oryx_fantasy/avatars/eye1.png
      Block2D:
        - Shape: triangle
          Color: [1.0, 0.0, 0.0]
          Scale: 1.0
```

13.3 Drunk Dwarf

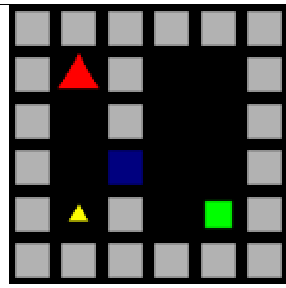
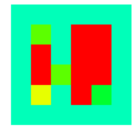
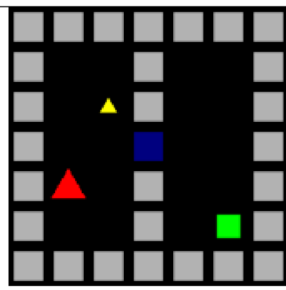
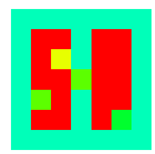
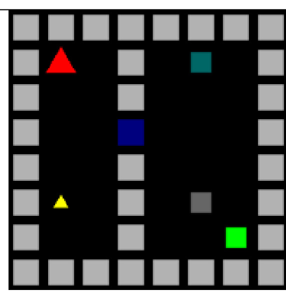
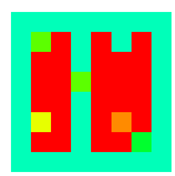
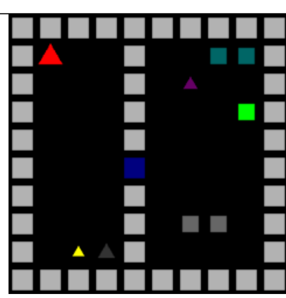
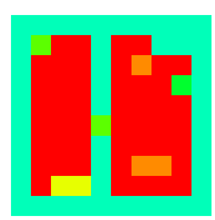
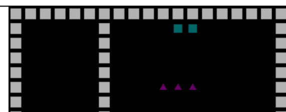
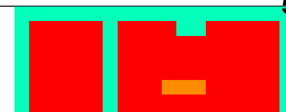
```
Single-Player/Mini-Grid/minigrid-drunkdwarf.yaml
```

13.3.1 Description

A port of the games provided in the https://github.com/maximecb/gym-minigrid environment, but you're a drunk dwarf trying find your keys that you've dropped to get to your bed (which is a coffin?? Wierd.).

13.3.2 Levels

Table 5: Levels

	Block2D	Sprite2D	Vector
Level ID: 0 Size: 6x6			
Level ID: 1 Size: 7x7			
Level ID: 2 Size: 8x8			
Level ID: 3 Size: 10x10			
			

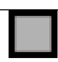
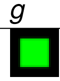
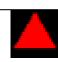
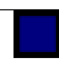
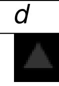
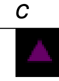
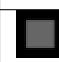
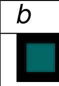
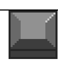
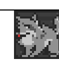
13.3.3 Code Example

The most basic way to create a Griddly Gym Environment. Defaults to level 0 and SPRITE_2D rendering.

```
import gym
import griddly

if __name__ == '__main__':

    env = gym.make('GDY-Drunk-Dwarf-v0')
    env.reset()

    # Replace with your own control algorithm!
    for s in range(1000):
        obs, reward, done, info = env.step(env.action_space.sample())
        env.render() # Renders the environment from the perspective of a single player

        env.render(observer='global') # Renders the entire environment

        if done:
            env.reset()
```

13.3.4 Objects

Table 6: Tiles

Name ->	wall	coffin_bed	drunk_dwarf	door	doggo	chair	table	bookshelf	key
Map Char ->	*W*	*g*	*A*	*D*	*d*	*c*	*t*	*b*	*k*
Block2D									
Sprite2D									
Vector									

13.3.5 Actions

stumble

Relative The actions are calculated relative to the object being controlled.

Action Id	Mapping
1	Rotate left
2	Move forwards
3	Rotate right

13.3.6 YAML

```yaml
Version: "0.1"
Environment:
  Name: Drunk Dwarf
  Description: A port of the games provided in the https://github.com/maximecb/gym-minigrid environment, but you're a drunk dwarf trying find your keys that you've dropped to get to your bed (which is a coffin?? Wierd.).
  Observers:
    Sprite2D:
      TileSize: 24
      BackgroundTile: oryx/oryx_fantasy/floor1-1.png
  Player:
    AvatarObject: drunk_dwarf # The player can only control a single drunk_dwarf in the game
    Observer:
      RotateWithAvatar: true
      TrackAvatar: true
      Height: 7
      Width: 7
      OffsetX: 0
      OffsetY: 3
  Termination:
    Win:
      - eq: [coffin_bed:count, 0] # If there are no boxes left
  Levels:
    - |
      W  W  W  W  W  W
      W  A  W  .  .  W
      W  .  W  .  .  W
      W  .  D  .  .  W
      W  k  W  .  g  W
      W  W  W  W  W  W
    - |
      W  W  W  W  W  W  W
      W  .  .  W  .  .  W
      W  .  k  W  .  .  W
      W  .  .  D  .  .  W
      W  A  .  W  .  .  W
      W  .  .  W  .  g  W
      W  W  W  W  W  W  W
    - |
      W  W  W  W  W  W  W  W
      W  A  .  W  .  b  .  W
      W  .  .  W  .  .  .  W
      W  .  .  D  .  .  .  W
      W  .  .  W  .  .  .  W
      W  k  .  W  .  t  .  W
      W  .  .  W  .  .  g  W
      W  W  W  W  W  W  W  W
    - |
      W  W  W  W  W  W  W  W  W  W
      W  A  .  .  W  .  .  b  b  W
```

(continues on next page)

(continued from previous page)

```
      W  .  .  .  W  .  c  .  .  W
      W  .  .  .  W  .  .  .  g  W
      W  .  .  .  W  .  .  .  .  W
      W  .  .  .  D  .  .  .  .  W
      W  .  .  .  W  .  .  .  .  W
      W  .  .  .  W  .  t  t  .  W
      W  .  k  d  W  .  .  .  .  W
      W  W  W  W  W  W  W  W  W  W
    - |
      W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W
      W  .  .  .  .  .  W  .  .  .  .  b  b  .  .  .  .  .  W
      W  .  .  .  .  .  W  .  .  .  .  .  .  .  .  .  .  .  W
      W  .  .  .  .  .  W  .  .  .  .  .  .  .  .  .  .  .  W
      W  .  .  .  .  .  W  .  .  .  .  .  .  .  .  .  .  .  W
      W  .  .  .  .  .  W  .  .  .  c  c  c  .  .  .  .  .  W
      W  .  .  .  .  .  W  .  .  .  .  .  .  .  .  .  .  .  W
      W  .  .  .  .  .  W  .  .  .  .  .  .  .  .  .  .  .  W
      W  .  .  .  .  .  W  .  .  .  .  .  .  .  .  .  .  .  W
      W  .  .  .  .  .  W  .  .  .  .  .  .  .  .  .  .  .  W
      W  .  .  .  .  .  D  .  .  .  .  .  g  .  .  .  .  .  W
      W  .  .  .  .  .  W  .  .  .  .  .  .  .  .  .  .  .  W
      W  .  .  k  .  .  W  .  .  .  .  .  .  .  .  .  .  .  W
      W  .  .  .  .  .  W  .  .  .  .  .  .  .  .  t  .  .  W
      W  .  .  .  .  .  W  .  .  .  .  .  .  .  .  .  .  .  W
      W  .  .  .  .  .  W  .  .  t  .  .  .  .  .  .  .  .  W
      W  A  .  .  .  .  W  .  .  .  .  .  .  .  .  .  .  .  W
      W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W

Actions:
  # Define the move action
  - Name: stumble
    InputMapping:
      Inputs:
        1:
          Description: Rotate left
          OrientationVector: [-1, 0]
        2:
          Description: Move forwards
          OrientationVector: [0, -1]
          VectorToDest: [0, -1]
        3:
          Description: Rotate right
          OrientationVector: [1, 0]
      Relative: true
    Behaviours:
      # Tell the agent to rotate if the drunk_dwarf performs an action on itself
      - Src:
          Object: drunk_dwarf
          Commands:
            - rot: _dir
        Dst:
          Object: drunk_dwarf
```

(continues on next page)

(continued from previous page)

```
    # The agent can move around freely in empty and always rotates the direction it is␣
↪travelling
    - Src:
        Object: drunk_dwarf
        Commands:
          - mov: _dest
      Dst:
        Object: [_empty, open_door]

    # If the drunk_dwarf moves into a coffin_bed object, the coffin_bed is removed,␣
↪triggering a win condition
    - Src:
        Object: drunk_dwarf
        Commands:
          - reward: 1
      Dst:
        Object: coffin_bed
        Commands:
          - remove: true

    # Keys and Locks
    - Src:
        Preconditions:
          - eq: [has_key, 1]
        Object: drunk_dwarf
        Commands:
          - mov: _dest
      Dst:
        Object: door
        Commands:
          - change_to: open_door
          - reward: 1

    # Avatar picks up the key
    - Src:
        Object: drunk_dwarf
        Commands:
          - mov: _dest
          - incr: has_key
          - reward: 1
      Dst:
        Object: key
        Commands:
          - remove: true

Objects:
  - Name: wall
    MapCharacter: W
    Observers:
      Sprite2D:
        - TilingMode: WALL_16
```

(continues on next page)

(continued from previous page)

```
        Image:
          - oryx/oryx_fantasy/wall1-0.png
          - oryx/oryx_fantasy/wall1-1.png
          - oryx/oryx_fantasy/wall1-2.png
          - oryx/oryx_fantasy/wall1-3.png
          - oryx/oryx_fantasy/wall1-4.png
          - oryx/oryx_fantasy/wall1-5.png
          - oryx/oryx_fantasy/wall1-6.png
          - oryx/oryx_fantasy/wall1-7.png
          - oryx/oryx_fantasy/wall1-8.png
          - oryx/oryx_fantasy/wall1-9.png
          - oryx/oryx_fantasy/wall1-10.png
          - oryx/oryx_fantasy/wall1-11.png
          - oryx/oryx_fantasy/wall1-12.png
          - oryx/oryx_fantasy/wall1-13.png
          - oryx/oryx_fantasy/wall1-14.png
          - oryx/oryx_fantasy/wall1-15.png
    Block2D:
      - Shape: square
        Color: [0.7, 0.7, 0.7]
        Scale: 1.0

- Name: coffin_bed
  MapCharacter: g
  Observers:
    Sprite2D:
      - Image: oryx/oryx_fantasy/coffin-1.png
    Block2D:
      - Shape: square
        Color: [0.0, 1.0, 0.0]
        Scale: 0.8

- Name: drunk_dwarf
  MapCharacter: A
  Z: 1
  Variables:
    - Name: has_key
      InitialValue: 0
  Observers:
    Sprite2D:
      - Image: oryx/oryx_fantasy/avatars/dwarf1.png
    Block2D:
      - Shape: triangle
        Color: [1.0, 0.0, 0.0]
        Scale: 1.0

- Name: door
  MapCharacter: D
  Observers:
    Sprite2D:
      - Image: oryx/oryx_fantasy/door-1.png
    Block2D:
```

(continues on next page)

(continued from previous page)

```
      - Shape: square
        Color: [0.0, 0.0, 0.5]
        Scale: 1.0

- Name: open_door
  Observers:
    Sprite2D:
      - Image: oryx/oryx_fantasy/open_door-1.png
    Block2D:
      - Shape: square
        Color: [0.0, 0.0, 0.0]
        Scale: 0.0

- Name: doggo
  MapCharacter: d
  Observers:
    Sprite2D:
      - Image: oryx/oryx_fantasy/avatars/doggo1.png
    Block2D:
      - Shape: triangle
        Color: [0.2, 0.2, 0.2]
        Scale: 0.7

- Name: chair
  MapCharacter: c
  Observers:
    Sprite2D:
      - Image: oryx/oryx_fantasy/chair-1.png
    Block2D:
      - Shape: triangle
        Color: [0.4, 0.0, 0.4]
        Scale: 0.6

- Name: table
  MapCharacter: t
  Observers:
    Sprite2D:
      - Image: oryx/oryx_fantasy/table-1.png
    Block2D:
      - Shape: square
        Color: [0.4, 0.4, 0.4]
        Scale: 0.8

- Name: bookshelf
  MapCharacter: b
  Observers:
    Sprite2D:
      - Image: oryx/oryx_fantasy/bookshelf-1.png
    Block2D:
      - Shape: square
        Color: [0.0, 0.4, 0.4]
        Scale: 0.8
```

(continues on next page)

(continued from previous page)

```yaml
  - Name: key
    MapCharacter: k
    Observers:
      Sprite2D:
        - Image: oryx/oryx_fantasy/key-3.png
      Block2D:
        - Shape: triangle
          Color: [1.0, 1.0, 0.0]
          Scale: 0.5
```

13.4 Doggo

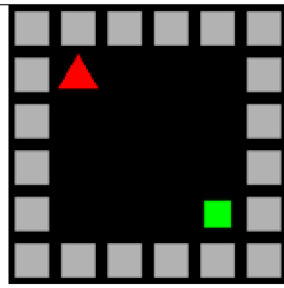
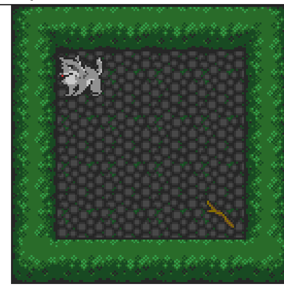
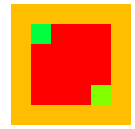
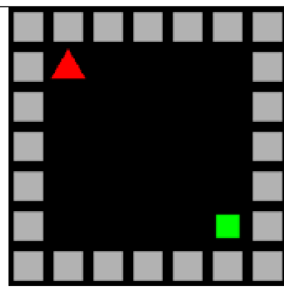
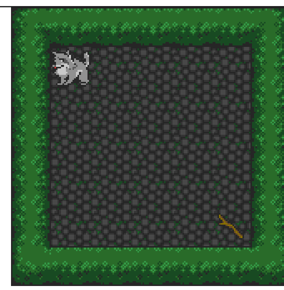
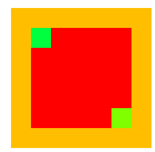
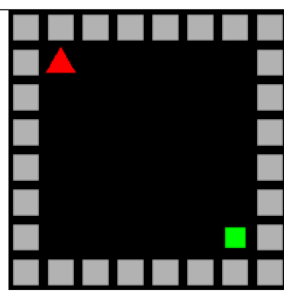
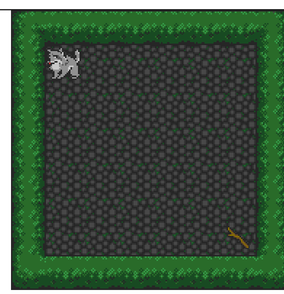
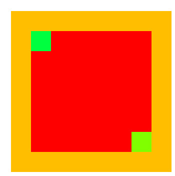
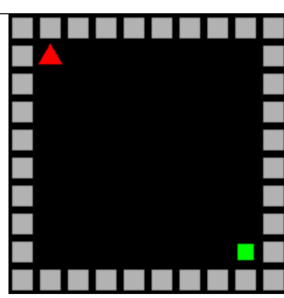
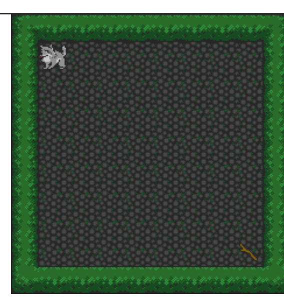
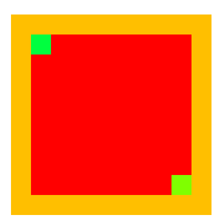
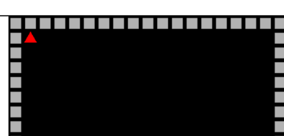
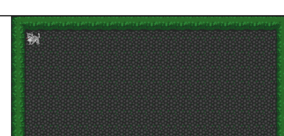
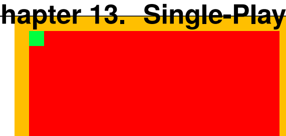
```
Single-Player/Mini-Grid/minigrid-doggo.yaml
```

13.4.1 Description

A port of the games provided in the https://github.com/maximecb/gym-minigrid Empty environment, but you're a doggo fetching a stick.

13.4.2 Levels

Table 7: Levels

	Block2D	Sprite2D	Vector
Level ID: 0 Size: 6x6			
Level ID: 1 Size: 7x7			
Level ID: 2 Size: 8x8			
Level ID: 3 Size: 10x10			
			

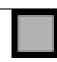
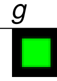
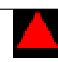
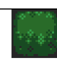
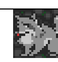
13.4.3 Code Example

The most basic way to create a Griddly Gym Environment. Defaults to level 0 and SPRITE_2D rendering.

```python
import gym
import griddly

if __name__ == '__main__':

    env = gym.make('GDY-Doggo-v0')
    env.reset()

    # Replace with your own control algorithm!
    for s in range(1000):
        obs, reward, done, info = env.step(env.action_space.sample())
        env.render() # Renders the environment from the perspective of a single player

        env.render(observer='global') # Renders the entire environment

        if done:
            env.reset()
```

13.4.4 Objects

Table 8: Tiles

Name ->	wall	stick	doggo
Map Char ->	*W*	*g*	*A*
Block2D			
Sprite2D			
Vector			

13.4.5 Actions

move

Relative The actions are calculated relative to the object being controlled.

Action Id	Mapping
1	Rotate left
2	Move forwards
3	Rotate right

13.4.6 YAML

```yaml
Version: "0.1"
Environment:
  Name: Doggo
  Description: A port of the games provided in the https://github.com/maximecb/gym-minigrid Empty environment, but you're a doggo fetching a stick.
  Observers:
    Sprite2D:
      TileSize: 24
      BackgroundTile: oryx/oryx_fantasy/floor9-3.png
  Player:
    Observer:
      RotateWithAvatar: true
      TrackAvatar: true
      Height: 7
      Width: 7
      OffsetX: 0
      OffsetY: 3
    AvatarObject: doggo
  Termination:
    Win:
      - eq: [stick:count, 0] # If there are no boxes left
  Levels:
    - |
      W   W   W   W   W   W
      W   A   .   .   .   W
      W   .   .   .   .   W
      W   .   .   .   .   W
      W   .   .   .   g   W
      W   W   W   W   W   W
    - |
      W   W   W   W   W   W   W
      W   A   .   .   .   .   W
      W   .   .   .   .   .   W
      W   .   .   .   .   .   W
      W   .   .   .   .   .   W
      W   .   .   .   .   g   W
      W   W   W   W   W   W   W
    - |
      W   W   W   W   W   W   W   W
      W   A   .   .   .   .   .   W
      W   .   .   .   .   .   .   W
      W   .   .   .   .   .   .   W
      W   .   .   .   .   .   .   W
      W   .   .   .   .   .   .   W
      W   .   .   .   .   .   g   W
      W   W   W   W   W   W   W   W
    - |
      W   W   W   W   W   W   W   W   W   W
      W   A   .   .   .   .   .   .   .   W
      W   .   .   .   .   .   .   .   .   W
      W   .   .   .   .   .   .   .   .   W
```

(continues on next page)

(continued from previous page)

```
      W  .  .  .  .  .  .  .  .  W
      W  .  .  .  .  .  .  .  .  W
      W  .  .  .  .  .  .  .  .  W
      W  .  .  .  .  .  .  .  .  W
      W  .  .  .  .  .  .  .  g  W
      W  W  W  W  W  W  W  W  W  W
    - |
      W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W
      W  A  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  W
      W  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  W
      W  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  W
      W  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  W
      W  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  W
      W  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  W
      W  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  W
      W  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  W
      W  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  W
      W  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  W
      W  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  W
      W  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  W
      W  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  W
      W  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  W
      W  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  W
      W  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  g  W
      W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W

Actions:
  # Define the move action
  - Name: move
    InputMapping:
      Inputs:
        1:
          Description: Rotate left
          OrientationVector: [-1, 0]
        2:
          Description: Move forwards
          OrientationVector: [0, -1]
          VectorToDest: [0, -1]
        3:
          Description: Rotate right
          OrientationVector: [1, 0]
      Relative: true
    Behaviours:
      # Tell the agent to rotate if the doggo performs an action on itself
      - Src:
          Object: doggo
          Commands:
            - rot: _dir
        Dst:
          Object: doggo

      # The agent can move around freely in empty and always rotates the direction it is
↪travelling
```

(continues on next page)

(continued from previous page)

```yaml
      - Src:
          Object: doggo
          Commands:
            - mov: _dest
        Dst:
          Object: _empty

      # If the doggo moves into a stick object, the stick is removed, triggering a win␣
↪condition
      - Src:
          Object: doggo
          Commands:
            - reward: 1
        Dst:
          Object: stick
          Commands:
            - remove: true

Objects:
  - Name: wall
    MapCharacter: W
    Observers:
      Sprite2D:
        - TilingMode: WALL_16
          Image:
            - oryx/oryx_fantasy/wall9-0.png
            - oryx/oryx_fantasy/wall9-1.png
            - oryx/oryx_fantasy/wall9-2.png
            - oryx/oryx_fantasy/wall9-3.png
            - oryx/oryx_fantasy/wall9-4.png
            - oryx/oryx_fantasy/wall9-5.png
            - oryx/oryx_fantasy/wall9-6.png
            - oryx/oryx_fantasy/wall9-7.png
            - oryx/oryx_fantasy/wall9-8.png
            - oryx/oryx_fantasy/wall9-9.png
            - oryx/oryx_fantasy/wall9-10.png
            - oryx/oryx_fantasy/wall9-11.png
            - oryx/oryx_fantasy/wall9-12.png
            - oryx/oryx_fantasy/wall9-13.png
            - oryx/oryx_fantasy/wall9-14.png
            - oryx/oryx_fantasy/wall9-15.png
      Block2D:
        - Shape: square
          Color: [0.7, 0.7, 0.7]
          Scale: 1.0

  - Name: stick
    MapCharacter: g
    Observers:
      Sprite2D:
        - Image: oryx/oryx_fantasy/stick-0.png
      Block2D:
```

(continues on next page)

(continued from previous page)

```yaml
      - Shape: square
        Color: [0.0, 1.0, 0.0]
        Scale: 0.8

- Name: doggo
  MapCharacter: A
  Observers:
    Sprite2D:
      - Image: oryx/oryx_fantasy/avatars/doggo1.png
    Block2D:
      - Shape: triangle
        Color: [1.0, 0.0, 0.0]
        Scale: 1.0
```

13.5 Butterflies and Spiders

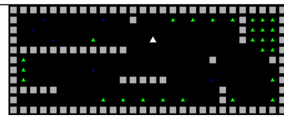
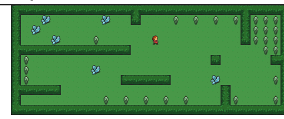
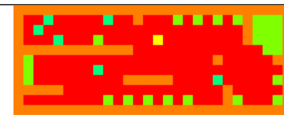
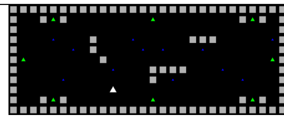
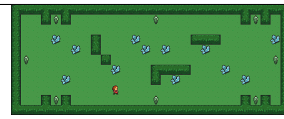
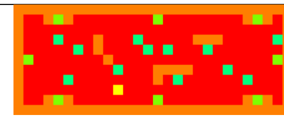
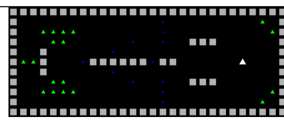
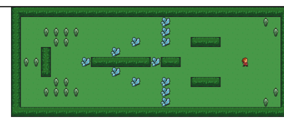
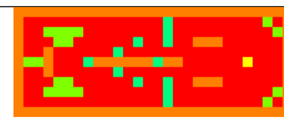
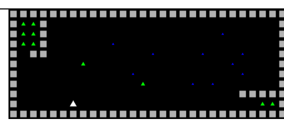
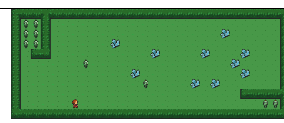
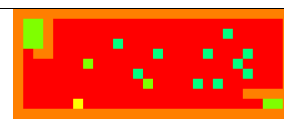
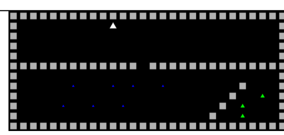
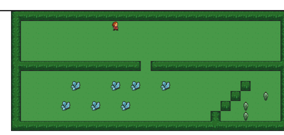
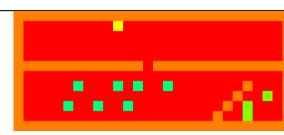
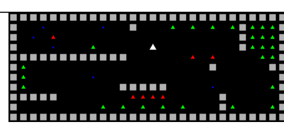
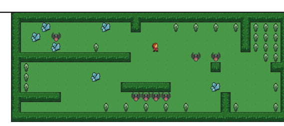
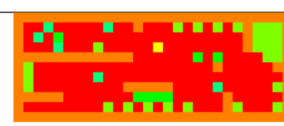
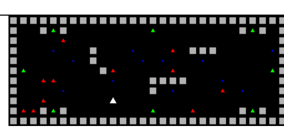
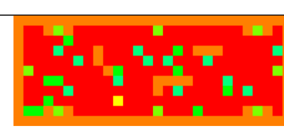
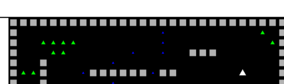
```
Single-Player/GVGAI/butterflies.yaml
```

13.5.1 Description

You want to catch all of the butterflies while also avoiding the spiders. Butterflies spawn slowly from cocoons. The butterflies are also eaten by the spiders so you need to be fast to collect them. You win the level as soon as there are no butterflies on the screen.

13.5.2 Levels

Table 9: Levels

	Block2D	Sprite2D	Vector
Level ID: 0 Size: 28x11			
Level ID: 1 Size: 28x11			
Level ID: 2 Size: 28x11			
Level ID: 3 Size: 28x11			
Level ID: 4 Size: 28x12			
Level ID: 5 Size: 28x11			
Level ID: 6 Size: 28x11			
			

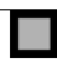
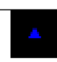
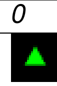
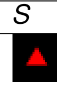
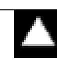
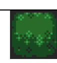
13.5.3 Code Example

The most basic way to create a Griddly Gym Environment. Defaults to level 0 and SPRITE_2D rendering.

```python
import gym
import griddly

if __name__ == '__main__':

    env = gym.make('GDY-Butterflies-and-Spiders-v0')
    env.reset()

    # Replace with your own control algorithm!
    for s in range(1000):
        obs, reward, done, info = env.step(env.action_space.sample())
        env.render() # Renders the environment from the perspective of a single player

        env.render(observer='global') # Renders the entire environment

        if done:
            env.reset()
```

13.5.4 Objects

Table 10: Tiles

Name ->	wall	butterfly	cocoon	spider	catcher
Map Char ->	*w*	*1*	*0*	*S*	*A*
Block2D					
Sprite2D					
Vector					

13.5.5 Actions

move

Action Id	Mapping
1	Left
2	Up
3	Right
4	Down

spider_random_movement

Relative The actions are calculated relative to the object being controlled.

Internal This action can only be called from other actions, not by the player.

Action Id	Mapping
1	Rotate left
2	Move forwards
3	Rotate right

spawn_butterfly

Internal This action can only be called from other actions, not by the player.

Action Id	Mapping
1	Left
2	Up
3	Right
4	Down

butterfly_random_movement

Internal This action can only be called from other actions, not by the player.

Action Id	Mapping
1	Left
2	Up
3	Right
4	Down

13.5.6 YAML

```
Version: "0.1"
Environment:
  Name: Butterflies and Spiders
  Description: |
    You want to catch all of the butterflies while also avoiding the spiders.␣
↪Butterflies spawn slowly from cocoons.
    The butterflies are also eaten by the spiders so you need to be fast to collect them.
    You win the level as soon as there are no butterflies on the screen.
  Observers:
    Sprite2D:
      TileSize: 24
      BackgroundTile: gvgai/oryx/grass_15.png
    Block2D:
      TileSize: 24
  Player:
    AvatarObject: catcher
```

(continues on next page)

(continued from previous page)

```yaml
  Termination:
    Win:
      - eq: [butterfly:count, 0] # If there are no butterflies
    Lose:
      - eq: [catcher:count, 0] # If the catcher gets killed
  Levels:
    - |
      w w w w w w w w w w w w w w w w w w w w w w w w w w w w
      w . . 1 . . . . . 1 . . w . . . 0 . 0 . 0 . 0 w 0 0 0 w
      w . 1 . . . . . . . . . . . . . . . . . . . . w 0 0 0 w
      w . . . 1 . . . 0 . . . . . A . . . . . . . . w 0 0 0 w
      w w w w w w w w w w w w . . . . . . . . . . . . . 0 0 w
      w 0 . . . . . . . . . . . . . . . . . . w . . . . . w w
      w 0 . . . . . . 1 . . . . . . . . . . . . . . . . . . w
      w 0 . . . . . . . . . w w w w w . . . . 1 . . . . . 0 w
      w w w w w . . . . . . . . . . . . . . . . w . . . . . w
      w . . . . . . . . 0 . 0 . 0 . 0 . 0 . . . w 0 . . . 0 w
      w w w w w w w w w w w w w w w w w w w w w w w w w w w w
    - |
      w w w w w w w w w w w w w w w w w w w w w w w w w w w w
      w . . w 0 w . . . . . . . . 0 . . . . . . . . w 0 w . w
      w . . . . . . . . . . . . . . . . . . . . . . . . . . w
      w . . . 1 . . . w . . . 1 . . . . . w w w . . . . . 1 w
      w . . . . . 1 . w . . . . 1 . 1 . . . 1 . . . . . . . w
      w 0 . . . . . . . w . . . . . . . . . . . . . . . . 0 w
      w . . . . . . . . . 1 . . . w w w w . . . 1 . . . . . w
      w . . . . 1 . . . . . . . . w . 1 . . . . . . 1 . . . w
      w . . . . . . . . . A . . . . . . . . . . . . . . . . w
      w . . w 0 w . . . . . . . . 0 . . . . . . . . w 0 w . w
      w w w w w w w w w w w w w w w w w w w w w w w w w w w w
    - |
      w w w w w w w w w w w w w w w w w w w w w w w w w w w w
      w . . . . . . . . . . . . . . 1 . . . . . . . . . 0 . w
      w . . 0 0 0 0 . . . . . . . . 1 . . . . . . . . . . 0 w
      w . . . 0 0 . . . . . . 1 . . 1 . . w w w . . . . . . w
      w . . w . . . . . . 1 . . . . . . . . . . . . . . . . w
      w 0 0 w . . . 1 w w w w w w 1 w w . . . . . . A . . . w
      w . . w . . . . . . 1 . . . . . . . . . . . . . . . . w
      w . . . 0 0 . . . . . . 1 . . 1 . . w w w . . . . . . w
      w . . 0 0 0 0 . . . . . . . . 1 . . . . . . . . . . 0 w
      w . . . . . . . . . . . . . . 1 . . . . . . . . . 0 . w
      w w w w w w w w w w w w w w w w w w w w w w w w w w w w
    - |
      w w w w w w w w w w w w w w w w w w w w w w w w w w w w
      w 0 0 w . . . . . . . . . . . . . . . . . . . . . . . w
      w 0 0 w . . . . . . . . . . . . . . . . . 1 . . . . . w
      w 0 0 w . . . . . . 1 . . . . . . . . . . . . . . . . w
      w . w w . . . . . . . . . . 1 . . . . 1 . . . 1 . . . w
      w . . . . . . 0 . . . . . . . . . . . . . . 1 . . . . w
      w . . . . . . . . . . . 1 . . . . . . . . . . 1 . . . w
      w . . . . . . . . . . . . 0 . . . . 1 . 1 . . . . . . w
      w . . . . . . . . . . . . . . . . . . . . . . w w w w w
```

(continues on next page)

(continued from previous page)

```
  w . . . . . A . . . . . . . . . . . . . . . . . . 0 0 w
  w w w w w w w w w w w w w w w w w w w w w w w w w w w w
- |
  w w w w w w w w w w w w w w w w w w w w w w w w w w w w
  w . . . . . . . . . A . . . . . . . . . . . . . . . . w
  w . . . . . . . . . . . . . . . . . . . . . . . . . . w
  w . . . . . . . . . . . . . . . . . . . . . . . . . . w
  w . . . . . . . . . . . . . . . . . . . . . . . . . . w
  w w w w w w w w w w w w w . w w w w w w w w w w w w w w
  w . . . . . . . . . . . . . . . . . . . . . . . . . . w
  w . . . . . 1 . . . 1 . 1 . . 1 . . . . . . . w . . . w
  w . . . . . . . . . . . . . . . . . . . . . w . . 0 . w
  w . . . . 1 . . 1 . . 1 . . . . . . . . . w . 0 . . . w
  w . . . . . . . . . . . . . . . . . . . w . . 0 . . . w
  w w w w w w w w w w w w w w w w w w w w w w w w w w w w
- |
  w w w w w w w w w w w w w w w w w w w w w w w w w w w w
  w . . 1 . . . . . 1 . . w . . . 0 . 0 . 0 . 0 w 0 0 0 w
  w . 1 . S . . . . . . . . . . . . . . . . . . w 0 0 0 w
  w . . . 1 . . . 0 . . . . . A . . . . . . . . w 0 0 0 w
  w w w w w w w w w w w w . . . . . . S . S . . . . 0 0 w
  w 0 . . . . . . . . . . . . . . . . . . w . . . . . w w
  w 0 . . . . . . 1 . . . . . . . . . . . . . . . . . . w
  w 0 . . . . . . . . . w w w w w . . . . 1 . . . . . 0 w
  w w w w w . . . . . . . S S S S . . . . . w . . . . . w
  w . . . . . . . . 0 . 0 . 0 . 0 . 0 . . . w 0 . . . 0 w
  w w w w w w w w w w w w w w w w w w w w w w w w w w w w
- |
  w w w w w w w w w w w w w w w w w w w w w w w w w w w w
  w . . w 0 w . . . . . . . . 0 . . . . . . . . w 0 w . w
  w . . . . S . . . . . . . . . . . . . . . . . . . . . w
  w . . . 1 . . . w . . . 1 . . . S . w w w . . . . . 1 w
  w . . . . . 1 . w . . . . 1 . 1 . . . 1 . . . . . . . w
  w 0 . . . . . . . w S . . . . . S . . . . . . . . . 0 w
  w . . S S . . . . . 1 . . . w w w w . . . 1 . . . . . w
  w . . . . 1 . . . . . . . . w . 1 . . . . S . 1 . . . w
  w . . S . . . . . . A . . . . . . . . . . . . . . . . w
  w S S w 0 w . . . . . . . . 0 . . . S . . . . w 0 w . w
  w w w w w w w w w w w w w w w w w w w w w w w w w w w w
- |
  w w w w w w w w w w w w w w w w w w w w w w w w w w w w
  w . . . . . . . . . . . . . . 1 . . . . . . . . . 0 . w
  w . . 0 0 0 0 . . . . . . . . 1 . . . . . . . . . . 0 w
  w . . . 0 0 . . . . . . 1 . . 1 . . w w w . . . . . . w
  w . . w . . . . . . 1 . . . . . . . . . . . . . . . . w
  w 0 0 w . . . 1 w w w w w w 1 w w . . . . . . A . . . w
  w . . w . . . . . . 1 . . . . . . . . . . . . . . . . w
  w . . . 0 0 . . . . . . 1 . . 1 . . w w w . . . . . . w
  w . . 0 0 0 0 . . . . . . . . 1 . . . . . . . . . . 0 w
  w . . . . . . . . . . . . . . 1 . . . . . . . . . 0 . w
  w w w w w w w w w w w w w w w w w w w w w w w w w w w w
- |
```

(continues on next page)

(continued from previous page)

```
   w w w w w w w w w w w w w w w w w w w w w w w w w w w w
   w 0 0 w . . . . . . . . . . . . . . . . . . . . . . . w
   w 0 0 w . . . . . . . . . . . . S . . . . 1 . . . . . w
   w 0 0 w . . . . . . 1 . . . . . S . . . . . . . . . . w
   w . w w . . . . . . . . . . 1 . S . . 1 . . . 1 . . . w
   w . . . . . . 0 . . . . . . . . S . . . . . 1 . . . . w
   w . . . . . . . . . . . 1 . . . S . . . . . . 1 . . . w
   w . . . . . . . . . . . . 0 . . S . 1 . 1 . . . . . . w
   w . . . . . . . . . . . . . . . S . . . . . . w w w w w
   w . . . . . A . . . . . . . . . . . . . . . . . . 0 0 w
   w w w w w w w w w w w w w w w w w w w w w w w w w w w w
  - |
   w w w w w w w w w w w w w w w w w w w w w w w w w w w w
   w . . . S . . . . . A . . . . . . . . . S . . . . . . w
   w . . . . . . . . . . . . . . . S . . . . . . S . . . w
   w . . . . . . S . . S . . . . . . . . . . . . . . . . w
   w . . . . . . . . . . . . . . . S . . . . . S . . . . w
   w w w w w w w w w w w w w . w w w w w w w w w w w w w w
   w . . . . . . . . . . . . . . . . . . . . . . . . . . w
   w . . . . . 1 . . . 1 . 1 . . 1 . . . . . . . w . . . w
   w . . . . . . . . . . . . . . . . . . . . . w . . 0 . w
   w . . . . 1 . . 1 . . 1 . . . . . . . . . w . 0 . . . w
   w . . . . . . . . . . . . . . . . . . . w . . 0 . . . w
   w w w w w w w w w w w w w w w w w w w w w w w w w w w w

Actions:
 - Name: spawn_butterfly
  InputMapping:
   Internal: true
  Behaviours:
   - Src:
     Object: cocoon
     Commands:
      - spawn: butterfly
      - exec:
        Action: spawn_butterfly
        Delay: 50
        Randomize: true

    Dst:
     Object: _empty

   - Src:
     Object: cocoon
     Commands:
      - exec:
        Action: spawn_butterfly
        Delay: 50
        Randomize: true
    Dst:
     Object: [cocoon, butterfly, catcher, wall]
```

(continues on next page)

(continued from previous page)

```yaml
  # Butterfly movement is different to spider movement
  - Name: butterfly_random_movement
    InputMapping:
      Internal: true
    Behaviours:

      # The butterfly moves into an empty space
      - Src:
          Object: butterfly
          Commands:
            - mov: _dest
            - exec:
                Action: butterfly_random_movement
                Delay: 3
                Randomize: true
        Dst:
          Object: _empty

      # if the butterfly tries to move into anything but an empty spot
      - Src:
          Object: butterfly
          Commands:
            - exec:
                Action: butterfly_random_movement
                Delay: 3
                Randomize: true
        Dst:
          Object: [ wall, spider, catcher, butterfly, cocoon ]

  # Define spider movement
  - Name: spider_random_movement
    InputMapping:
      Inputs:
        1:
          Description: Rotate left
          OrientationVector: [-1, 0]
        2:
          Description: Move forwards
          OrientationVector: [0, -1]
          VectorToDest: [0, -1]
        3:
          Description: Rotate right
          OrientationVector: [1, 0]
      Relative: true
      Internal: true
    Behaviours:
      # Spider rotates on the spot
      - Src:
          Object: spider
          Commands:
            - rot: _dir
            - exec:
```

(continues on next page)

(continued from previous page)

```
            Action: spider_random_movement
            Delay: 3
            Randomize: true
    Dst:
      Object: spider

  # The catcher and the spider can move into empty space
  - Src:
      Object: spider
      Commands:
        - mov: _dest
        - exec:
            Action: spider_random_movement
            Delay: 3
            Randomize: true
    Dst:
      Object: _empty

  # The spider will not move into the wall or the gem, but it needs to keep moving
  - Src:
      Object: spider
      Commands:
        - exec:
            Action: spider_random_movement
            Delay: 3
            Randomize: true
    Dst:
      Object: wall

  # If the spider moves into a butterfly it dies
  - Src:
      Object: spider
      Commands:
        - mov: _dest
        - exec:
            Action: spider_random_movement
            Delay: 3
            Randomize: true
    Dst:
      Object: butterfly
      Commands:
        - remove: true
        - reward: -1
  # if the spider moves into the catcher it dies
  - Src:
      Object: spider
    Dst:
      Object: catcher
      Commands:
        - remove: true
        - reward: -10
```

(continues on next page)

(continued from previous page)

```yaml
  # Define the move action
  - Name: move
    Behaviours:

      # If the catcher moves into a spider
      - Src:
          Object: catcher
          Commands:
            - remove: true
            - reward: -1
        Dst:
          Object: spider

      # The catcher move into an empty space
      - Src:
          Object: catcher
          Commands:
            - mov: _dest
        Dst:
          Object: _empty

      # If the catcher moves into a butterfly object, the butterfly is caught YAY!
      - Src:
          Object: catcher
          Commands:
            - mov: _dest
            - reward: 1
        Dst:
          Object: butterfly
          Commands:
            - remove: true

Objects:
  - Name: wall
    MapCharacter: 'w'
    Observers:
      Sprite2D:
        - TilingMode: WALL_16
          Image:
            - oryx/oryx_fantasy/wall9-0.png
            - oryx/oryx_fantasy/wall9-1.png
            - oryx/oryx_fantasy/wall9-2.png
            - oryx/oryx_fantasy/wall9-3.png
            - oryx/oryx_fantasy/wall9-4.png
            - oryx/oryx_fantasy/wall9-5.png
            - oryx/oryx_fantasy/wall9-6.png
            - oryx/oryx_fantasy/wall9-7.png
            - oryx/oryx_fantasy/wall9-8.png
            - oryx/oryx_fantasy/wall9-9.png
            - oryx/oryx_fantasy/wall9-10.png
            - oryx/oryx_fantasy/wall9-11.png
            - oryx/oryx_fantasy/wall9-12.png
```

(continues on next page)

(continued from previous page)

```
          - oryx/oryx_fantasy/wall9-13.png
          - oryx/oryx_fantasy/wall9-14.png
          - oryx/oryx_fantasy/wall9-15.png
    Block2D:
      - Shape: square
        Color: [0.7, 0.7, 0.7]
        Scale: 0.9

- Name: butterfly
  InitialActions:
    - Action: butterfly_random_movement
      Delay: 3
      Randomize: true
  MapCharacter: '1'
  Observers:
    Sprite2D:
      - Image: gvgai/newset/butterfly1.png
    Block2D:
      - Shape: triangle
        Color: [0.0, 0.0, 1.0]
        Scale: 0.3

- Name: cocoon
  MapCharacter: '0'
  InitialActions:
    - Action: spawn_butterfly
      Delay: 50
      Randomize: true
  Observers:
    Sprite2D:
      - Image: gvgai/newset/cocoonb1.png
    Block2D:
      - Shape: triangle
        Color: [0.0, 1.0, 0.0]
        Scale: 0.5

- Name: spider
  InitialActions:
    - Action: spider_random_movement
      Delay: 3
      Randomize: true
  MapCharacter: 'S'
  Observers:
    Sprite2D:
      - Image: oryx/oryx_fantasy/avatars/spider1.png
    Block2D:
      - Shape: triangle
        Color: [1.0, 0.0, 0.0]
        Scale: 0.5

- Name: catcher
  MapCharacter: 'A'
```

(continues on next page)

(continued from previous page)

```
    Observers:
      Sprite2D:
        - Image: gvgai/newset/girl5.png
      Block2D:
        - Shape: triangle
          Color: [1.0, 1.0, 1.0]
          Scale: 0.8
```

13.6 Partially Observable Sokoban - 2

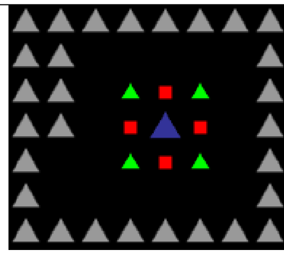
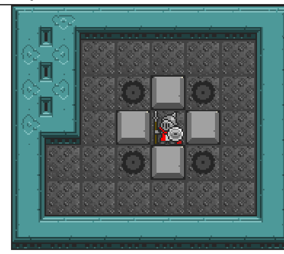
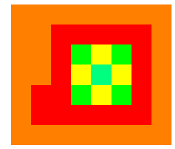
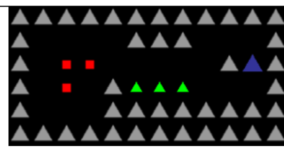
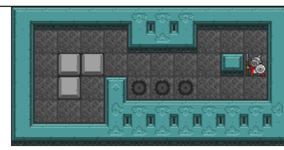
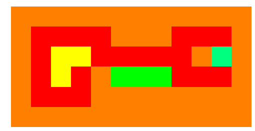
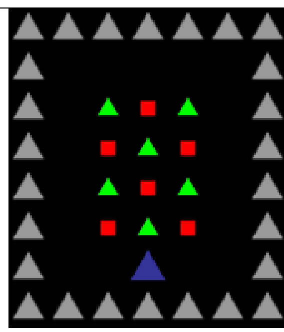
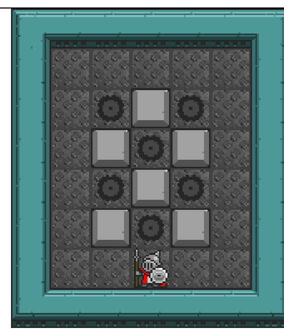
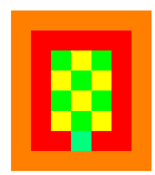
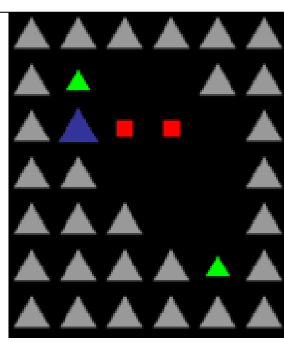
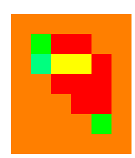
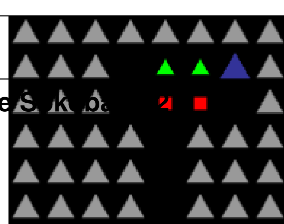
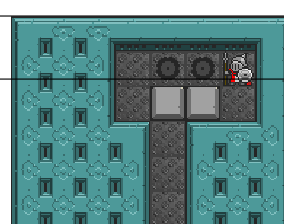
```
Single-Player/GVGAI/sokoban2_partially_observable.yaml
```

13.6.1 Description

Push the boxes onto the marked spaces, once a box has moved onto a space, it cannot be moved

13.6.2 Levels

Table 11: Levels

<table>
<tr><th></th><th>Block2D</th><th>Sprite2D</th><th>Vector</th></tr>
<tr><td><table><tr><td>Level ID</td><td>0</td></tr><tr><td>Size</td><td>8x7</td></tr></table></td><td></td><td></td><td></td></tr>
<tr><td><table><tr><td>Level ID</td><td>1</td></tr><tr><td>Size</td><td>12x6</td></tr></table></td><td></td><td></td><td></td></tr>
<tr><td><table><tr><td>Level ID</td><td>2</td></tr><tr><td>Size</td><td>7x8</td></tr></table></td><td></td><td></td><td></td></tr>
<tr><td><table><tr><td>Level ID</td><td>3</td></tr><tr><td>Size</td><td>6x7</td></tr></table></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td></tr>
</table>

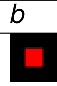
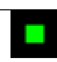
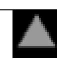
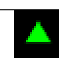
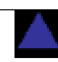
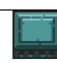
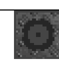
13.6.3 Code Example

The most basic way to create a Griddly Gym Environment. Defaults to level 0 and SPRITE_2D rendering.

```python
import gym
import griddly

if __name__ == '__main__':

    env = gym.make('GDY-Partially-Observable-Sokoban---2-v0')
    env.reset()

    # Replace with your own control algorithm!
    for s in range(1000):
        obs, reward, done, info = env.step(env.action_space.sample())
        env.render() # Renders the environment from the perspective of a single player

        env.render(observer='global') # Renders the entire environment

        if done:
            env.reset()
```

13.6.4 Objects

Table 12: Tiles

Name ->	box	box_in_place	wall	hole	avatar
Map Char ->	*b*	*f*	*w*	*h*	*A*
Block2D					
Sprite2D					
Vector					

13.6.5 Actions

move

Relative The actions are calculated relative to the object being controlled.

Action Id	Mapping
1	Rotate left
2	Move forwards
3	Rotate right

13.6.6 YAML

```yaml
Version: "0.1"
Environment:
  Name: Partially Observable Sokoban - 2
  Description: Push the boxes onto the marked spaces, once a box has moved onto a space,␣
↪it cannot be moved
  Observers:
    Sprite2D:
      TileSize: 24
      BackgroundTile: gvgai/newset/floor2.png
  Player:
    Observer:
      RotateWithAvatar: true
      TrackAvatar: true
      Height: 5
      Width: 5
      OffsetX: 0
      OffsetY: 0
    AvatarObject: avatar # The player can only control a single avatar in the game
  Termination:
    Win:
      - eq: [box:count, 0] # If there are no boxes left
  Levels:
    - |
      wwwwwwwww
      ww.....w
      ww.hbh.w
      ww.bAb.w
      w..hbh.w
      w......w
      wwwwwwwww
    - |
      wwwwwwwwwwwww
      w....www...w
      w.bb.....wAw
      w.b.whhh...w
      w...wwwwwwww
      wwwwwwwwwwwww
    - |
      wwwwwwww
      w.....w
      w.hbh.w
      w.bhb.w
      w.hbh.w
      w.bhb.w
      w..A..w
      wwwwwwww
    - |
      wwwwww
      wh..ww
      wAbb.w
      ww...w
```

(continues on next page)

(continued from previous page)

```
      www..w
      wwwwhw
      wwwwww
    - |
      wwwwwwww
      www.hhAw
      www.bb.w
      wwww.www
      wwww.www
      wwww.www
      wwww.www
      w....www
      w.w...ww
      w...w.ww
      www...ww
      wwwwwwww

Actions:
  # Define the move action
  - Name: move
    InputMapping:
      Inputs:
        1:
          Description: Rotate left
          OrientationVector: [-1, 0]
        2:
          Description: Move forwards
          OrientationVector: [0, -1]
          VectorToDest: [0, -1]
        3:
          Description: Rotate right
          OrientationVector: [1, 0]
      Relative: true
    Behaviours:

      # Avatar rotates
      - Src:
          Object: avatar
          Commands:
            - rot: _dir
        Dst:
          Object: avatar

      # The agent can move around freely in empty space and over holes
      - Src:
          Object: avatar
          Commands:
            - mov: _dest
        Dst:
          Object: [_empty, hole]

      # Boxes can move into empty space
```

(continues on next page)

(continued from previous page)

```
      - Src:
          Object: box
          Commands:
            - mov: _dest
        Dst:
          Object: _empty

      # The agent can push boxes
      - Src:
          Object: avatar
          Commands:
            - mov: _dest
        Dst:
          Object: [box, box_in_place]
          Commands:
            - exec:
                Action: move

      # If a box is moved into a hole, it should change to in-place box
      - Src:
          Object: [box, box_in_place]
          Commands:
            - mov: _dest
            - change_to: box_in_place
            - reward: 1
        Dst:
          Object: hole

      # If in-place box is moved into empty space, it should be a plain box
      - Src:
          Object: box_in_place
          Commands:
            - mov: _dest
            - change_to: box
            - reward: -1
        Dst:
          Object: _empty

Objects:
  - Name: box
    Z: 2
    MapCharacter: b
    Observers:
      Sprite2D:
        - Image: gvgai/newset/block2.png
      Block2D:
        - Shape: square
          Color: [1.0, 0.0, 0.0]
          Scale: 0.5

  - Name: box_in_place
    Z: 2
```

(continues on next page)

(continued from previous page)

```
    MapCharacter: f
    Observers:
      Sprite2D:
        - Image: gvgai/newset/block1.png
      Block2D:
        - Shape: square
          Color: [0.0, 1.0, 0.0]
          Scale: 0.5

  - Name: wall
    MapCharacter: w
    Observers:
      Sprite2D:
        - TilingMode: WALL_16
          Image:
            - gvgai/oryx/wall3_0.png
            - gvgai/oryx/wall3_1.png
            - gvgai/oryx/wall3_2.png
            - gvgai/oryx/wall3_3.png
            - gvgai/oryx/wall3_4.png
            - gvgai/oryx/wall3_5.png
            - gvgai/oryx/wall3_6.png
            - gvgai/oryx/wall3_7.png
            - gvgai/oryx/wall3_8.png
            - gvgai/oryx/wall3_9.png
            - gvgai/oryx/wall3_10.png
            - gvgai/oryx/wall3_11.png
            - gvgai/oryx/wall3_12.png
            - gvgai/oryx/wall3_13.png
            - gvgai/oryx/wall3_14.png
            - gvgai/oryx/wall3_15.png
      Block2D:
        - Shape: triangle
          Color: [0.6, 0.6, 0.6]
          Scale: 0.9

  - Name: hole
    Z: 1
    MapCharacter: h
    Observers:
      Sprite2D:
        - Image: gvgai/oryx/cspell4.png
      Block2D:
        - Shape: triangle
          Color: [0.0, 1.0, 0.0]
          Scale: 0.6

  - Name: avatar
    Z: 2
    MapCharacter: A
    Observers:
      Sprite2D:
```

(continues on next page)

(continued from previous page)

```
      - Image: gvgai/oryx/knight1.png
    Block2D:
      - Shape: triangle
        Color: [0.2, 0.2, 0.6]
        Scale: 1.0
```

13.7 Labyrinth

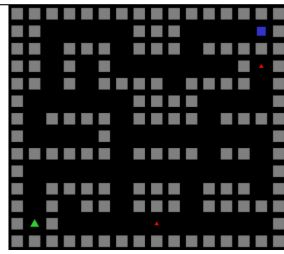
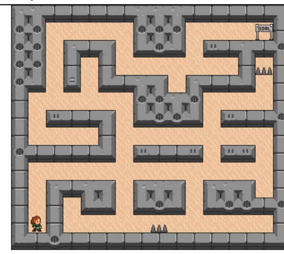
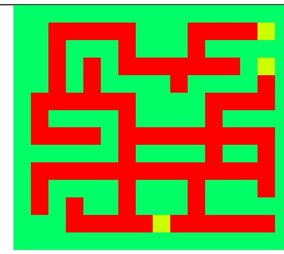
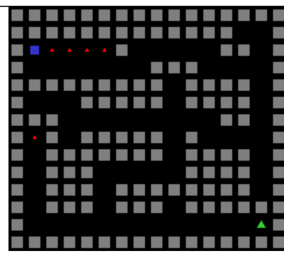
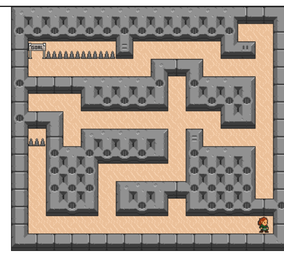
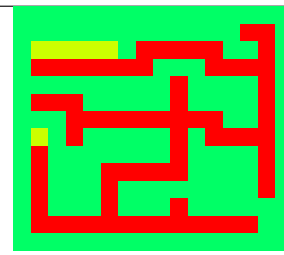
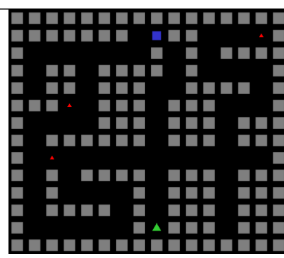
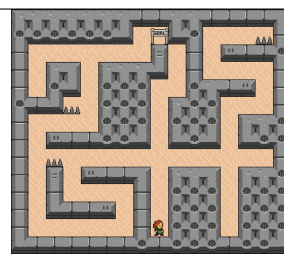
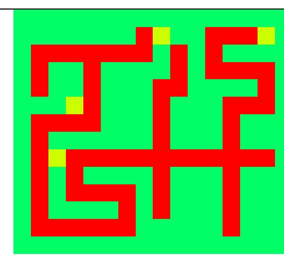
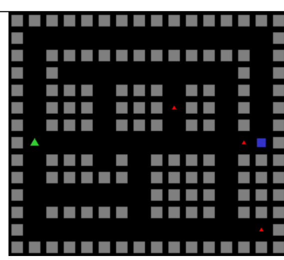
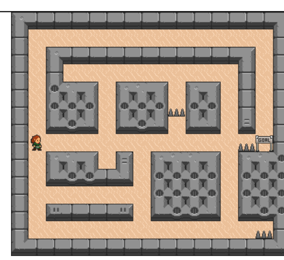
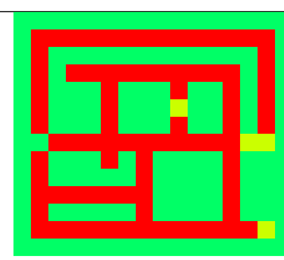
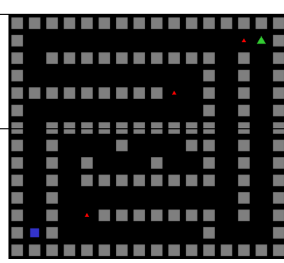
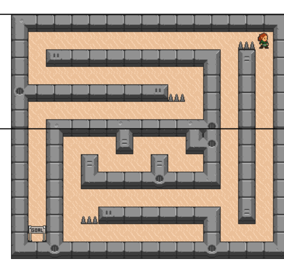
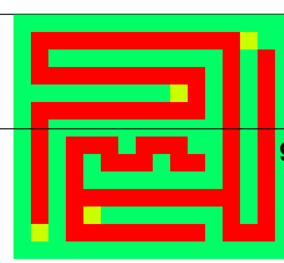
```
Single-Player/GVGAI/labyrinth.yaml
```

13.7.1 Description

Its a maze, find your way out. Watch out for spikey things.

13.7.2 Levels

Table 13: Levels

	Block2D	Sprite2D	Vector
Level ID: 0 Size: 16x14			
Level ID: 1 Size: 16x14			
Level ID: 2 Size: 16x14			
Level ID: 3 Size: 16x14			
			

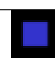
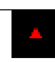
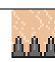
13.7.3 Code Example

The most basic way to create a Griddly Gym Environment. Defaults to level 0 and SPRITE_2D rendering.

```
import gym
import griddly

if __name__ == '__main__':

    env = gym.make('GDY-Labyrinth-v0')
    env.reset()

    # Replace with your own control algorithm!
    for s in range(1000):
        obs, reward, done, info = env.step(env.action_space.sample())
        env.render() # Renders the environment from the perspective of a single player

        env.render(observer='global') # Renders the entire environment

        if done:
            env.reset()
```

13.7.4 Objects

Table 14: Tiles

Name ->	avatar	exit	trap	wall
Map Char ->	*A*	*x*	*t*	*w*
Block2D				
Sprite2D				
Vector				

13.7.5 Actions

move

Action Id	Mapping
1	Left
2	Up
3	Right
4	Down

13.7.6 YAML

```yaml
Version: "0.1"
Environment:
  Name: Labyrinth
  Description: Its a maze, find your way out. Watch out for spikey things.
  Observers:
    Sprite2D:
      TileSize: 24
      BackgroundTile: gvgai/oryx/backBiege.png
  Player:
    AvatarObject: avatar
  Termination:
    Win:
      - eq: [exit:count, 0]
    Lose:
      - eq: [avatar:count, 0]
  Levels:
    - |
      w w w w w w w w w w w w w w w w
      w w . . . . . w w w . . . . x w
      w w . w w w . w w w . w w w w w
      w w . w . w . . . . . . . w t w
      w w . w . w w w w . w w w w . w
      w . . . . . . w w w w . . . . w
      w . w w w w . w w w w . w w w w
      w . . . . w . . . . . . . . . w
      w w w w w w . w w w w . w w . w
      w . . . . . . . . . . . . . . w
      w . w w w w . w w w . w w w . w
      w . w . w w . w w w . w w w w w
      w A w . . . . . t . . . . . . w
      w w w w w w w w w w w w w w w w
    - |
      w w w w w w w w w w w w w w w w
      w w w w w w w w w w w w w . . w
      w x t t t t w . . . . . w w . w
      w . . . . . . . w w w . . . . w
      w w w w w w w w w . w w w w . w
      w . . . w w w w w . w w w w . w
      w w w . . . . . . . . . w w . w
      w t w . w w w w w . w . . . . w
      w . w w w w w w w . w w w w . w
      w . w w w . . . . . w w w w . w
      w . w w w . w w w w w w w w . w
      w . w w w . w w w . w w w w w w
      w . . . . . . . . . . . . . A w
      w w w w w w w w w w w w w w w w
    - |
      w w w w w w w w w w w w w w w w
      w w w w w w w . x w w . . . t w
      w . . . . . . . w . w . w w w w
      w . w w . w w w w . w . . . . w
```

(continues on next page)

(continued from previous page)

```
      w . w w . w w w . . w w w w . w
      w w w t . w w w . w w w . . . w
      w . . . . w w w . w w w . w w w
      w . w w w w w w . w w w . w w w
      w . t . . . . . . . . . . . . w
      w . w . w w w w . w w w . w w w
      w . w . . . . w . w w w . w w w
      w . w w w w . w . w w w . w w w
      w . . . . . . w A w w w . w w w
      w w w w w w w w w w w w w w w w
    - |
      w w w w w w w w w w w w w w w w
      w . . . . . . . . . . . . . . w
      w . w w w w w w w w w w w w . w
      w . w . . . . . . . . . . w . w
      w . w w w . w w w . w w . w . w
      w . w w w . w w w t w w . w . w
      w . w w w . w w w . w w . w . w
      w A . . . . . . . . . . . t x w
      w . w w w . w . w w w w . w w w
      w . w w w w w . w w w w . w w w
      w . . . . . . . w w w w . w w w
      w . w w w w w . w w w w . w w w
      w . . . . . . . . . . . . . t w
      w w w w w w w w w w w w w w w w
    - |
      w w w w w w w w w w w w w w w w
      w . . . . . . . . . . . . t A w
      w . w w w w w w w w w w . w . w
      w . . . . . . . . . . w . w . w
      w w w w w w w w w t . w . w . w
      w . . . . . . . . . . w . w . w
      w . w w w w w w w w w w . w . w
      w . w . . . w . . . w w . w . w
      w . w . w . . . w . . w . w . w
      w . w . w w w w w w w w . w . w
      w . w . . . . . . . . . . w . w
      w . w . t w w w w w w w . w . w
      w x w . . . . . . . w . . . . w
      w w w w w w w w w w w w w w w w

Actions:
  # Define the move action
  - Name: move
    Behaviours:
      # Avatar can move into empty space
      - Src:
          Object: avatar
          Commands:
            - mov: _dest
        Dst:
          Object: _empty
```

(continues on next page)

(continued from previous page)

```
      # If Avatar hits a trap, remove it
      - Src:
          Object: avatar
          Commands:
            - remove: true
            - reward: -1
        Dst:
          Object: trap

      # If Avatar hits the exit, remove the exit
      - Src:
          Object: avatar
          Commands:
            - reward: 1
        Dst:
          Object: exit
          Commands:
            - remove: true

Objects:
  - Name: avatar
    MapCharacter: A
    Observers:
      Sprite2D:
        - Image: gvgai/newset/girl1.png
      Block2D:
        - Shape: triangle
          Color: [0.2, 0.8, 0.2]
          Scale: 0.6

  - Name: exit
    MapCharacter: x
    Observers:
      Sprite2D:
        - Image: gvgai/newset/exit2.png
      Block2D:
        - Shape: square
          Color: [0.2, 0.2, 0.8]
          Scale: 0.7

  - Name: trap
    MapCharacter: t
    Observers:
      Sprite2D:
        - Image: gvgai/oryx/spike2.png
      Block2D:
        - Shape: triangle
          Color: [1.0, 0.0, 0.0]
          Scale: 0.3

  - Name: wall
```

(continues on next page)

(continued from previous page)

```
    MapCharacter: w
    Observers:
      Sprite2D:
        - TilingMode: WALL_16
          Image:
            - oryx/oryx_fantasy/wall8-0.png
            - oryx/oryx_fantasy/wall8-1.png
            - oryx/oryx_fantasy/wall8-2.png
            - oryx/oryx_fantasy/wall8-3.png
            - oryx/oryx_fantasy/wall8-4.png
            - oryx/oryx_fantasy/wall8-5.png
            - oryx/oryx_fantasy/wall8-6.png
            - oryx/oryx_fantasy/wall8-7.png
            - oryx/oryx_fantasy/wall8-8.png
            - oryx/oryx_fantasy/wall8-9.png
            - oryx/oryx_fantasy/wall8-10.png
            - oryx/oryx_fantasy/wall8-11.png
            - oryx/oryx_fantasy/wall8-12.png
            - oryx/oryx_fantasy/wall8-13.png
            - oryx/oryx_fantasy/wall8-14.png
            - oryx/oryx_fantasy/wall8-15.png
      Block2D:
        - Shape: square
          Color: [0.5, 0.5, 0.5]
          Scale: 0.9
```

13.8 Bait

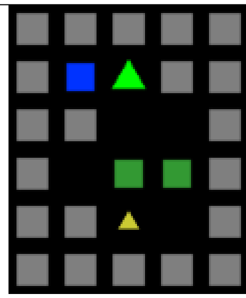
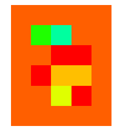
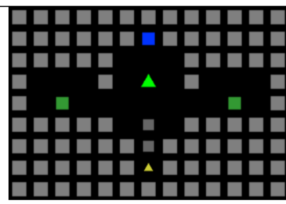
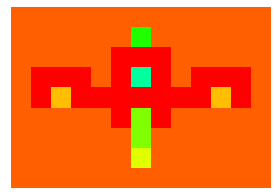
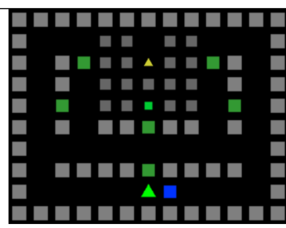
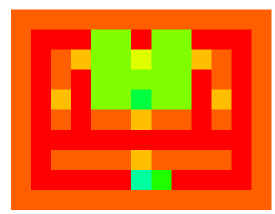
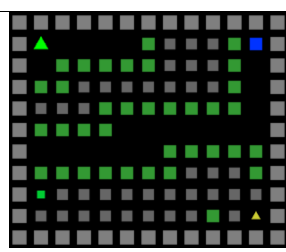
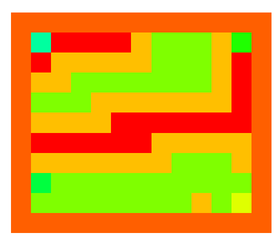
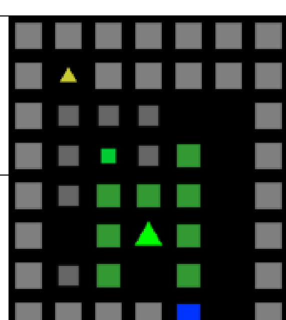
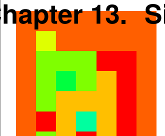
```
Single-Player/GVGAI/bait.yaml
```

13.8.1 Description

Get the key and unlock the door. Fill in the holes in the floor with blocks to get to the key.

13.8.2 Levels

Table 15: Levels

<table>
<tr><th></th><th>Block2D</th><th>Sprite2D</th><th>Vector</th></tr>
<tr><td><table><tr><td>Level ID</td><td>0</td></tr><tr><td>Size</td><td>5x6</td></tr></table></td><td></td><td></td><td></td></tr>
<tr><td><table><tr><td>Level ID</td><td>1</td></tr><tr><td>Size</td><td>13x9</td></tr></table></td><td></td><td></td><td></td></tr>
<tr><td><table><tr><td>Level ID</td><td>2</td></tr><tr><td>Size</td><td>13x10</td></tr></table></td><td></td><td></td><td></td></tr>
<tr><td><table><tr><td>Level ID</td><td>3</td></tr><tr><td>Size</td><td>13x11</td></tr></table></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td></tr>
</table>

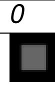
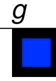
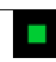
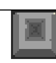
13.8.3 Code Example

The most basic way to create a Griddly Gym Environment. Defaults to level 0 and SPRITE_2D rendering.

```python
import gym
import griddly

if __name__ == '__main__':

    env = gym.make('GDY-Bait-v0')
    env.reset()

    # Replace with your own control algorithm!
    for s in range(1000):
        obs, reward, done, info = env.step(env.action_space.sample())
        env.render() # Renders the environment from the perspective of a single player

        env.render(observer='global') # Renders the entire environment

        if done:
            env.reset()
```

13.8.4 Objects

Table 16: Tiles

Name ->	avatar	hole	box	key	goal	mushroom	wall
Map Char ->	*A*	*0*	*1*	*k*	*g*	*m*	*w*
Block2D							
Sprite2D							
Vector							

13.8.5 Actions

move

Action Id	Mapping
1	Left
2	Up
3	Right
4	Down

13.8.6 YAML

```
# http://www.mobygames.com/game/bait

Version: "0.1"
Environment:
  Name: Bait
  Description: Get the key and unlock the door. Fill in the holes in the floor with␣
↪blocks to get to the key.
  Observers:
    Sprite2D:
      TileSize: 24
      BackgroundTile: gvgai/oryx/backLBrown.png
  Player:
    AvatarObject: avatar
  Termination:
    Win:
      - eq: [goal:count, 0]
    Lose:
      - eq: [avatar:count, 0]
  Levels:
    - |
      w w w w w
      w g A w w
      w w . . w
      w . 1 1 w
      w w k . w
      w w w w w
    - |
      w w w w w w w w w w w w w
      w w w w w w g w w w w w w
      w w w w w . . . w w w w w
      w . . . w . A . w . . . w
      w . 1 . . . . . . . 1 . w
      w w w w w . 0 . w w w w w
      w w w w w w 0 w w w w w w
      w w w w w w k w w w w w w
      w w w w w w w w w w w w w
    - |
      w w w w w w w w w w w w w
      w . . . 0 0 . 0 0 . . . w
      w . w 1 0 0 k 0 0 1 w . w
      w . w . 0 0 0 0 0 . w . w
      w . 1 . 0 0 m 0 0 . 1 . w
      w . w . w w 1 w w . w . w
      w . . . . . . . . . . . w
      w . w w w w 1 w w w w . w
      w . . . . . . A g . . . w
      w w w w w w w w w w w w w
    - |
      w w w w w w w w w w w w w
      w A . . . . 1 0 0 0 1 g w
      w . 1 1 1 1 1 0 0 0 1 . w
```

(continues on next page)

(continued from previous page)

```
      w 1 1 0 0 0 0 0 0 0 1 . w
      w 0 0 0 1 1 1 1 1 1 1 . w
      w 1 1 1 1 . . . . . . . w
      w . . . . . . 1 1 1 1 1 w
      w 1 1 1 1 1 1 1 0 0 0 1 w
      w m 0 0 0 0 0 0 0 0 0 0 w
      w 0 0 0 0 0 0 0 0 1 0 k w
      w w w w w w w w w w w w w
    - |
      w w w w w w w
      w k w w w w w
      w 0 0 0 . . w
      w 0 m 0 1 . w
      w 0 1 1 1 . w
      w . 1 A 1 . w
      w 0 1 . 1 . w
      w w w w g . w
      w w w w w w w

Actions:
  # Define the move action
  - Name: move
    Behaviours:
      # Avatar and boxes can move into empty space
      - Src:
          Object: [avatar, box]
          Commands:
            - mov: _dest
        Dst:
          Object: _empty

      # Boxes can be pushed by the avatar
      - Src:
          Object: avatar
          Commands:
            - mov: _dest
        Dst:
          Object: box
          Commands:
            - cascade: _dest

      # If a box falls into a hole, both disappear
      - Src:
          Object: box
          Commands:
            - remove: true
            - reward: 1
        Dst:
          Object: hole
          Commands:
            - remove: true
```

(continues on next page)

(continued from previous page)

```yaml
      # If the avatar falls into a hole remove the avatar
      - Src:
          Object: avatar
          Commands:
            - remove: true
            - reward: -1
        Dst:
          Object: hole

      # If the avatar picks up a mushroom, remove the mushroom
      - Src:
          Object: avatar
          Commands:
            - reward: 1
            - mov: _dest
        Dst:
          Object: mushroom
          Commands:
            - remove: true

      # Only an avatar with a key can
      - Src:
          Preconditions:
            - eq: [has_key, 1]
          Object: avatar
          Commands:
            - reward: 5
        Dst:
          Object: goal
          Commands:
            - remove: true

      # Avatar picks up the key
      - Src:
          Object: avatar
          Commands:
            - mov: _dest
            - incr: has_key
        Dst:
          Object: key
          Commands:
            - remove: true

Objects:
  - Name: avatar
    MapCharacter: A
    Variables:
      - Name: has_key
    Observers:
      Sprite2D:
        - Image: gvgai/oryx/swordman1_0.png
      Block2D:
```

(continues on next page)

(continued from previous page)

```
      - Shape: triangle
        Color: [0.0, 1.0, 0.0]
        Scale: 0.8

- Name: hole
  MapCharacter: "0"
  Observers:
    Sprite2D:
      - Image: gvgai/newset/hole1.png
    Block2D:
      - Shape: square
        Color: [0.4, 0.4, 0.4]
        Scale: 0.7

- Name: box
  MapCharacter: "1"
  Observers:
    Sprite2D:
      - Image: gvgai/newset/block3.png
    Block2D:
      - Shape: square
        Color: [0.2, 0.6, 0.2]
        Scale: 0.8

- Name: key
  MapCharacter: k
  Observers:
    Sprite2D:
      - Image: gvgai/oryx/key2.png
    Block2D:
      - Shape: triangle
        Color: [0.8, 0.8, 0.2]
        Scale: 0.5

- Name: goal
  MapCharacter: g
  Observers:
    Sprite2D:
      - Image: gvgai/oryx/doorclosed1.png
    Block2D:
      - Shape: square
        Color: [0.0, 0.2, 1.0]
        Scale: 0.8

- Name: mushroom
  MapCharacter: m
  Observers:
    Sprite2D:
      - Image: gvgai/oryx/mushroom2.png
    Block2D:
      - Shape: square
        Color: [0.0, 0.8, 0.2]
```

(continues on next page)

(continued from previous page)

```
        Scale: 0.5

  - Name: wall
    MapCharacter: w
    Observers:
      Sprite2D:
        - TilingMode: WALL_16
          Image:
            - gvgai/oryx/dirtWall_0.png
            - gvgai/oryx/dirtWall_1.png
            - gvgai/oryx/dirtWall_2.png
            - gvgai/oryx/dirtWall_3.png
            - gvgai/oryx/dirtWall_4.png
            - gvgai/oryx/dirtWall_5.png
            - gvgai/oryx/dirtWall_6.png
            - gvgai/oryx/dirtWall_7.png
            - gvgai/oryx/dirtWall_8.png
            - gvgai/oryx/dirtWall_9.png
            - gvgai/oryx/dirtWall_10.png
            - gvgai/oryx/dirtWall_11.png
            - gvgai/oryx/dirtWall_12.png
            - gvgai/oryx/dirtWall_13.png
            - gvgai/oryx/dirtWall_14.png
            - gvgai/oryx/dirtWall_15.png
      Block2D:
        - Shape: square
          Color: [0.5, 0.5, 0.5]
          Scale: 0.9
```

13.9 Partially Observable Zen Puzzle

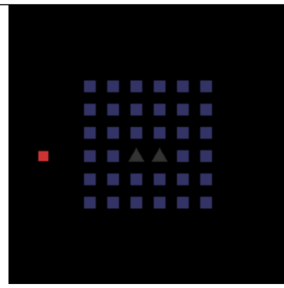
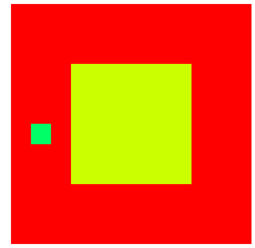
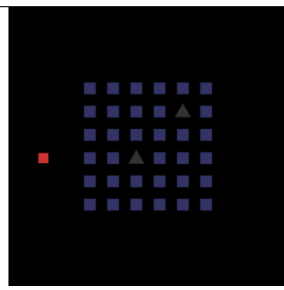
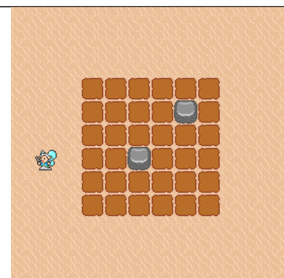
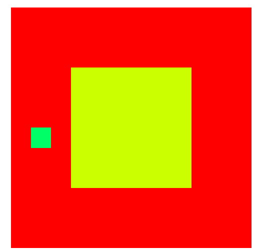
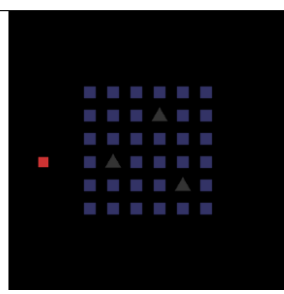
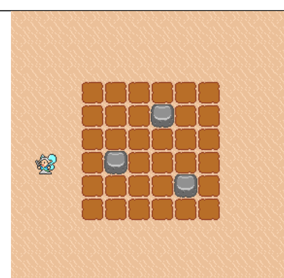
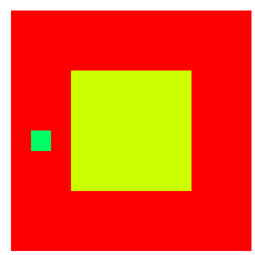
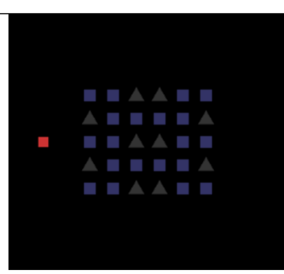
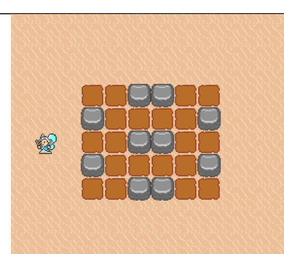
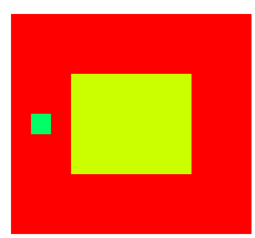
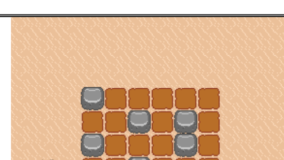
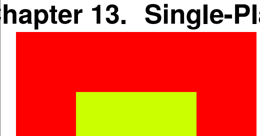
```
Single-Player/GVGAI/zenpuzzle_partially_observable.yaml
```

13.9.1 Description

Set all the tiles in the level to the same color, but you cannot move over a tile more than once! (Not even sure why this is zen its super frustrating)

13.9.2 Levels

Table 17: Levels

	Block2D	Sprite2D	Vector
Level ID: 0 Size: 12x12			
Level ID: 1 Size: 12x12			
Level ID: 2 Size: 12x12			
Level ID: 3 Size: 12x11			
			

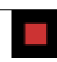
13.9.3 Code Example

The most basic way to create a Griddly Gym Environment. Defaults to level 0 and SPRITE_2D rendering.

```python
import gym
import griddly

if __name__ == '__main__':

    env = gym.make('GDY-Partially-Observable-Zen-Puzzle-v0')
    env.reset()

    # Replace with your own control algorithm!
    for s in range(1000):
        obs, reward, done, info = env.step(env.action_space.sample())
        env.render() # Renders the environment from the perspective of a single player

        env.render(observer='global') # Renders the entire environment

        if done:
            env.reset()
```

13.9.4 Objects

Table 18: Tiles

Name ->	avatar	ground	rock
Map Char ->	*A*	*g*	*r*
Block2D			
Sprite2D			
Vector			

13.9.5 Actions

move

Relative The actions are calculated relative to the object being controlled.

Action Id	Mapping
1	Rotate left
2	Move forwards
3	Rotate right

13.9.6 YAML

```yaml
Version: "0.1"
Environment:
  Name: Partially Observable Zen Puzzle
  Description: Set all the tiles in the level to the same color, but you cannot move␣
↪over a tile more than once! (Not even sure why this is zen its super frustrating)
  Observers:
    Sprite2D:
      TileSize: 24
      BackgroundTile: gvgai/oryx/backBiege.png
  Player:
    Observer:
      RotateWithAvatar: true
      TrackAvatar: true
      Height: 5
      Width: 5
      OffsetX: 0
      OffsetY: 2
    AvatarObject: avatar
  Termination:
    Win:
      - eq: [ground:count, 0]
    Lose:
      - eq: [_steps, 1000]
  Levels:
    - |
      ............
      ............
      ............
      ...gggggg...
      ...gggggg...
      ...gggggg...
      .A.ggrrgg...
      ...gggggg...
      ...gggggg...
      ............
      ............
      ............
    - |
      ............
      ............
      ............
      ...gggggg...
      ...ggggrg...
      ...gggggg...
      .A.ggrggg...
      ...gggggg...
      ...gggggg...
      ............
      ............
      ............
    - |
```

(continues on next page)

(continued from previous page)

```
      ............
      ............
      ............
      ...gggggg...
      ...gggrgg...
      ...gggggg...
      .A.grgggg...
      ...ggggrg...
      ...gggggg...
      ............
      ............
      ............
    - |
      ............
      ............
      ............
      ...ggrrgg...
      ...rggggr...
      .A.ggrrgg...
      ...rggggr...
      ...ggrrgg...
      ............
      ............
      ............
    - |
      ............
      ............
      ............
      ...rggggg...
      ...ggrgrg...
      ...rgggrg...
      .A.ggrggg...
      ...rgggrg...
      ...ggrggg...
      ............
      ............
      ............

Actions:
  # Define the move action
  - Name: move
    InputMapping:
      Inputs:
        1:
          Description: Rotate left
          OrientationVector: [-1, 0]
        2:
          Description: Move forwards
          OrientationVector: [0, -1]
          VectorToDest: [0, -1]
        3:
          Description: Rotate right
```

(continues on next page)

(continued from previous page)

```
        OrientationVector: [1, 0]
      Relative: true
    Behaviours:

      # Avatar rotates
      - Src:
          Object: avatar
          Commands:
            - rot: _dir
        Dst:
          Object: avatar

      # The agent can move around freely in empty space and over holes
      - Src:
          Object: avatar
          Commands:
            - mov: _dest

        Dst:
          Object: _empty

      - Src:
          Object: avatar
          Commands:
            - mov: _dest
        Dst:
          Object: ground
          Commands:
            - change_to: walked
            - reward: 1

Objects:
  - Name: avatar
    MapCharacter: A
    Z: 1
    Observers:
      Sprite2D:
        - Image: gvgai/oryx/angel1.png
      Block2D:
        - Shape: square
          Color: [0.8, 0.2, 0.2]
          Scale: 0.6

  - Name: ground
    MapCharacter: g
    Observers:
      Sprite2D:
        - Image: gvgai/oryx/floorTileOrange.png
      Block2D:
        - Shape: square
          Color: [0.2, 0.2, 0.4]
          Scale: 0.7
```

(continues on next page)

(continued from previous page)

```yaml
- Name: walked
  Z: 0
  Observers:
    Sprite2D:
      - Image: gvgai/oryx/floorTileGreen.png
    Block2D:
      - Shape: square
        Color: [0.2, 0.6, 0.2]
        Scale: 0.8

- Name: rock
  MapCharacter: r
  Observers:
    Sprite2D:
      - Image: gvgai/oryx/wall5.png
    Block2D:
      - Shape: triangle
        Color: [0.2, 0.2, 0.2]
        Scale: 0.8
```

13.10 Partially Observable Cook Me Pasta

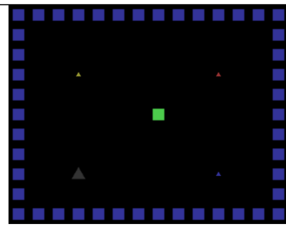
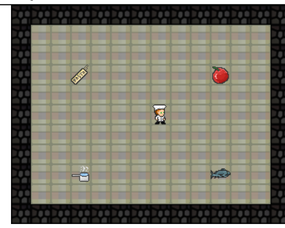
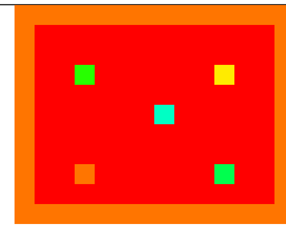
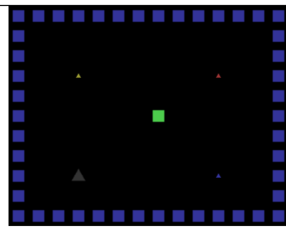
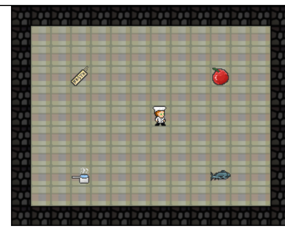
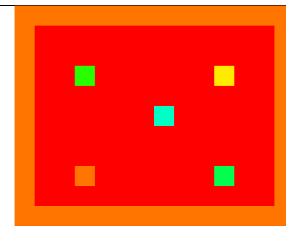
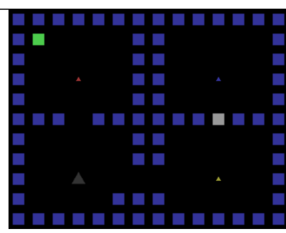
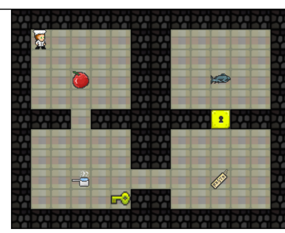
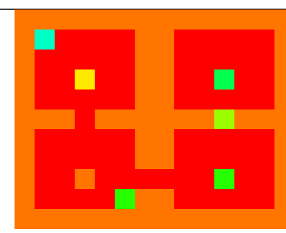
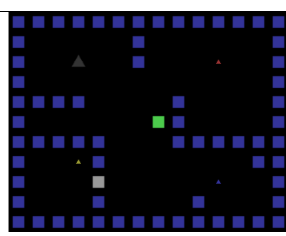
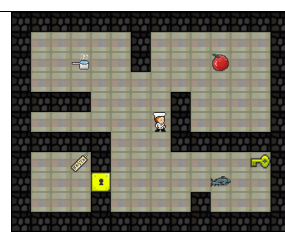
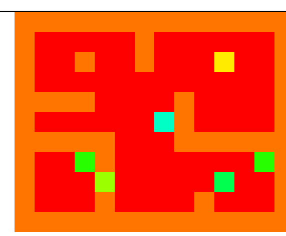
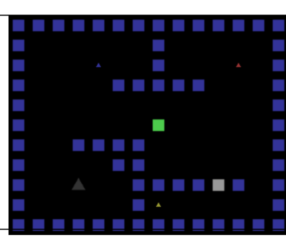
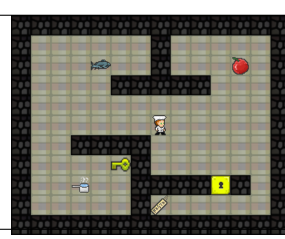
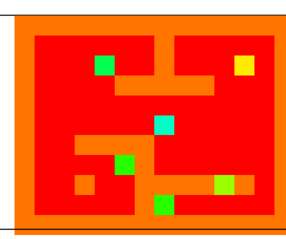
```
Single-Player/GVGAI/cookmepasta_partially_observable.yaml
```

13.10.1 Description

Help the chef create the meal, but make sure the ingredients are put together in the right order.

13.10.2 Levels

Table 19: Levels

	Block2D	Sprite2D	Vector
Level ID: 0 Size: 14x11			
Level ID: 1 Size: 14x11			
Level ID: 2 Size: 14x11			
Level ID: 3 Size: 14x11			
Level ID: 4 Size: 14x11			

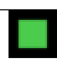
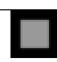
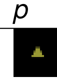
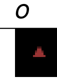
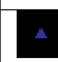
13.10.3 Code Example

The most basic way to create a Griddly Gym Environment. Defaults to level 0 and SPRITE_2D rendering.

```python
import gym
import griddly

if __name__ == '__main__':

    env = gym.make('GDY-Partially-Observable-Cook-Me-Pasta-v0')
    env.reset()

    # Replace with your own control algorithm!
    for s in range(1000):
        obs, reward, done, info = env.step(env.action_space.sample())
        env.render() # Renders the environment from the perspective of a single player

        env.render(observer='global') # Renders the entire environment

        if done:
            env.reset()
```

13.10.4 Objects

Table 20: Tiles

Name ->	avatar	wall	key	lock	boiling_water	raw_pasta	tomato	tuna
Map Char ->	*A*	*w*	*k*	*l*	*b*	*p*	*o*	*t*
Block2D								
Sprite2D								
Vector								

13.10.5 Actions

move

Relative The actions are calculated relative to the object being controlled.

Action Id	Mapping
1	Rotate left
2	Move forwards
3	Rotate right

13.10.6 YAML

```yaml
Version: "0.1"
Environment:
  Name: Partially Observable Cook Me Pasta
  Description: Help the chef create the meal, but make sure the ingredients are put together in the right order.
  Observers:
    Sprite2D:
      TileSize: 24
      BackgroundTile: gvgai/newset/floor6.png
  Player:
    Observer:
      RotateWithAvatar: true
      TrackAvatar: true
      Height: 5
      Width: 5
      OffsetX: 0
      OffsetY: 2
    AvatarObject: avatar # The player can only control a single avatar in the game
  Termination:
    Win:
      - eq: [complete_meal:count, 1]
    Lose:
      - eq: [wrong_place:count, 1]
  Levels:
    - |
      wwwwwwwwwwwwww
      w............w
      w............w
      w..p......o..w
      w............w
      w......A.....w
      w............w
      w............w
      w..b......t..w
      w............w
      wwwwwwwwwwwwww
    - |
      wwwwwwwwwwwwww
      w............w
      w............w
      w..p......o..w
      w............w
      w......A.....w
      w............w
      w............w
      w..b......t..w
      w............w
      wwwwwwwwwwwwww
    - |
      wwwwwwwwwwwwww
      wA....ww.....w
```

(continues on next page)

(continued from previous page)

```
      w.....ww.....w
      w..o..ww..t..w
      w.....ww.....w
      www.wwwwwwlwww
      w.....ww.....w
      w.....ww.....w
      w..b......p..w
      w....kww.....w
      wwwwwwwwwwwwww
    - |
      wwwwwwwwwwwwww
      w.....w......w
      w..b..w...o..w
      w............w
      wwww....w....w
      w......Aw....w
      wwwww...wwwwww
      w..pw.......kw
      w...l.....t..w
      w...w....w...w
      wwwwwwwwwwwwww
    - |
      wwwwwwwwwwwwww
      w......w.....w
      w...t..w...o.w
      w....wwwww...w
      w............w
      w......A.....w
      w..wwww......w
      w....kw......w
      w..b..wwwwlw.w
      w.....wp.....w
      wwwwwwwwwwwwww
    - |
      wwwwwwwwwwwwww
      w..lA........w
      w..wwwwwww.woww
      w..t.........w
      w..wwwwwwww..w
      w..w....k.w..w
      w..w..p...w..w
      w..w...wwww..w
      w..w...b.....w
      w..w.....w...w
      wwwwwwwwwwwwww

Actions:
  # Define the move action
  - Name: move
    InputMapping:
      Inputs:
        1:
```

(continues on next page)

(continued from previous page)

```
          Description: Rotate left
          OrientationVector: [-1, 0]
        2:
          Description: Move forwards
          OrientationVector: [0, -1]
          VectorToDest: [0, -1]
        3:
          Description: Rotate right
          OrientationVector: [1, 0]
      Relative: true
    Behaviours:

      # Avatar rotates
      - Src:
          Object: avatar
          Commands:
            - rot: _dir
        Dst:
          Object: avatar

      # The agent can move around freely in empty space and over holes
      - Src:
          Object: avatar
          Commands:
            - mov: _dest
        Dst:
          Object: [boiling_water, raw_pasta, tomato, tuna, cooked_pasta, pasta_sauce]
          Commands:
            - cascade: _dest
      - Src:
          Object: [avatar, boiling_water, raw_pasta, tomato, tuna, cooked_pasta, pasta_
↪sauce]
          Commands:
            - mov: _dest
        Dst:
          Object: _empty

      # Behaviour for boiling_water
      - Src:
          Object: boiling_water
          Commands:
            - remove: true
            - reward: 4
        Dst:
          Object: raw_pasta
          Commands:
            - change_to: cooked_pasta

      # Behaviour for raw_pasta
      - Src:
          Object: raw_pasta
          Commands:
```

(continues on next page)

(continued from previous page)

```
          - remove: true
          - reward: 4
    Dst:
      Object: boiling_water
      Commands:
        - change_to: cooked_pasta

  # Behaviours for tomato
  - Src:
      Object: tomato
      Commands:
        - remove: true
        - reward: 4
    Dst:
      Object: tuna
      Commands:
        - change_to: pasta_sauce

  # Behaviours for tuna
  - Src:
      Object: tuna
      Commands:
        - remove: true
        - reward: 4
    Dst:
      Object: tomato
      Commands:
        - change_to: pasta_sauce

  # Behaviours for cooked_pasta
  - Src:
      Object: cooked_pasta
      Commands:
        - remove: true
        - reward: 17
    Dst:
      Object: pasta_sauce
      Commands:
        - change_to: complete_meal
  # Behaviours for pasta_sauce
  - Src:
      Object: pasta_sauce
      Commands:
        - remove: true
        - reward: 17
    Dst:
      Object: cooked_pasta
      Commands:
        - change_to: complete_meal

  # If the wrong things are mixed together
  - Src:
```

(continues on next page)

(continued from previous page)

```
        Object: [raw_pasta, boiling_water]
        Commands:
          - remove: true
          - reward: -1
      Dst:
        Object: [tuna, tomato, pasta_sauce]
        Commands:
          - change_to: wrong_place

    - Src:
        Object: [tuna, tomato, pasta_sauce]
        Commands:
          - remove: true
          - reward: -1
      Dst:
        Object: [boiling_water, raw_pasta]
        Commands:
          - change_to: wrong_place


    # Keys and Locks
    - Src:
        Preconditions:
          - eq: [has_key, 1]
        Object: avatar
        Commands:
          - mov: _dest
      Dst:
        Object: lock
        Commands:
          - remove: true

    # Avatar picks up the key
    - Src:
        Object: avatar
        Commands:
          - mov: _dest
          - incr: has_key
      Dst:
        Object: key
        Commands:
          - remove: true

Objects:

  - Name: avatar
    MapCharacter: A
    Variables:
      - Name: has_key
    Observers:
      Sprite2D:
        - Image: gvgai/newset/chef.png
```

(continues on next page)

(continued from previous page)

```
      Block2D:
        - Shape: square
          Color: [0.3, 0.8, 0.3]
          Scale: 0.8

  - Name: wall
    MapCharacter: w
    Observers:
      Sprite2D:
        - Image: gvgai/newset/floor4.png
      Block2D:
        - Shape: square
          Color: [0.2, 0.2, 0.6]
          Scale: 0.8

  - Name: key
    MapCharacter: k
    Observers:
      Sprite2D:
        - Image: gvgai/oryx/key2.png
      Block2D:
        - Shape: square
          Color: [0.2, 0.2, 0.6]
          Scale: 0.8

  - Name: lock
    MapCharacter: l
    Observers:
      Sprite2D:
        - Image: gvgai/newset/lock1.png
      Block2D:
        - Shape: square
          Color: [0.6, 0.6, 0.6]
          Scale: 0.8

  - Name: boiling_water
    MapCharacter: b
    Observers:
      Sprite2D:
        - Image: gvgai/newset/boilingwater.png
      Block2D:
        - Shape: triangle
          Color: [0.2, 0.2, 0.2]
          Scale: 0.8
  - Name: raw_pasta
    MapCharacter: p
    Observers:
      Sprite2D:
        - Image: gvgai/newset/pasta.png
      Block2D:
        - Shape: triangle
          Color: [0.6, 0.6, 0.2]
```

(continues on next page)

(continued from previous page)

```
        Scale: 0.3
  - Name: tomato
    MapCharacter: o
    Observers:
      Sprite2D:
        - Image: gvgai/newset/tomato.png
      Block2D:
        - Shape: triangle
          Color: [0.6, 0.2, 0.2]
          Scale: 0.3
  - Name: tuna
    MapCharacter: t
    Observers:
      Sprite2D:
        - Image: gvgai/newset/tuna.png
      Block2D:
        - Shape: triangle
          Color: [0.2, 0.2, 0.6]
          Scale: 0.3

  - Name: cooked_pasta
    Observers:
      Sprite2D:
        - Image: gvgai/newset/pastaplate.png
      Block2D:
        - Shape: triangle
          Color: [0.6, 0.6, 0.6]
          Scale: 0.7
  - Name: pasta_sauce
    Observers:
      Sprite2D:
        - Image: gvgai/newset/tomatosauce.png
      Block2D:
        - Shape: triangle
          Color: [0.6, 0.0, 0.2]
          Scale: 0.7

  - Name: complete_meal
    Observers:
      Sprite2D:
        - Image: gvgai/newset/pastasauce.png
      Block2D:
        - Shape: triangle
          Color: [0.6, 0.0, 0.2]
          Scale: 0.7

  - Name: wrong_place
    Observers:
      Sprite2D:
        - Image: gvgai/oryx/slash1.png
      Block2D:
        - Shape: square
```

(continues on next page)

(continued from previous page)

```
        Color: [1.0, 0.0, 0.0]
        Scale: 1.0
```

13.11 Spider Nest

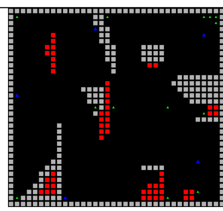
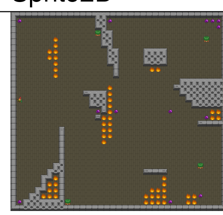
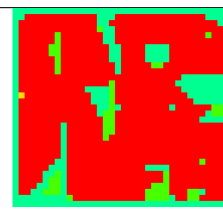
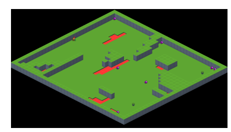
```
Single-Player/GVGAI/spider-nest.yaml
```

13.11.1 Description

A port of the games provided in the https://github.com/maximecb/gym-minigrid Dynamic obstacles environment, but you're a gnome avoiding ghosts to get to a gem.

13.11.2 Levels

Table 21: Levels

<table>
<tr><th></th><th>Block2D</th><th>Sprite2D</th><th>Vector</th><th>Isometric</th></tr>
<tr><td><table><tr><td>Level ID</td><td>0</td></tr><tr><td>Size</td><td>36x33</td></tr></table></td><td></td><td></td><td></td><td></td></tr>
</table>

13.11.3 Code Example

The most basic way to create a Griddly Gym Environment. Defaults to level 0 and SPRITE_2D rendering.

```python
import gym
import griddly

if __name__ == '__main__':

    env = gym.make('GDY-Spider-Nest-v0')
    env.reset()

    # Replace with your own control algorithm!
    for s in range(1000):
        obs, reward, done, info = env.step(env.action_space.sample())
        env.render() # Renders the environment from the perspective of a single player
```

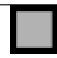
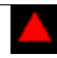
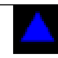
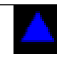
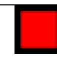
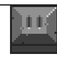
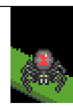
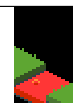
(continues on next page)

(continued from previous page)

```
        env.render(observer='global') # Renders the entire environment

        if done:
            env.reset()
```

13.11.4 Objects

Table 22: Tiles

Name ->	wall	spider	gem	gnome	nest	lava
Map Char ->	*W*	*G*	*g*	*A*	*N*	*L*
Block2D						
Sprite2D						
Vector						
Isometric						

13.11.5 Actions

move

Relative The actions are calculated relative to the object being controlled.

Action Id	Mapping
1	Rotate left
2	Move forwards
3	Rotate right
4	Move Backwards

spawn_spider

Internal This action can only be called from other actions, not by the player.

Action Id	Mapping
1	Left
2	Up
3	Right
4	Down

random_movement

Relative The actions are calculated relative to the object being controlled.

Internal This action can only be called from other actions, not by the player.

Action Id	Mapping
1	Rotate left
2	Move forwards
3	Rotate right

13.11.6 YAML


```
Version: "0.1"
Environment:
  Name: Spider Nest
  Description: A port of the games provided in the https://github.com/maximecb/gym-
↪minigrid Dynamic obstacles environment, but you're a gnome avoiding ghosts to get to a␣
↪gem.
  Observers:
    Sprite2D:
      TileSize: 24
      BackgroundTile: oryx/oryx_fantasy/floor2-2.png
    Isometric:
      TileSize: [32, 48]
      IsoTileHeight: 16
      IsoTileDepth: 4
      BackgroundTile: oryx/oryx_iso_dungeon/grass-1.png
    Block2D:
      TileSize: 24
  Player:
    Observer:
      RotateWithAvatar: true
      TrackAvatar: true
      Height: 7
      Width: 7
      OffsetX: 0
      OffsetY: 2
    AvatarObject: gnome
  Termination:
    Win:
      - eq: [gem:count, 0] # If there are no gems left
    Lose:
      - eq: [gnome:count, 0] # If the gnome gets killed
  Levels:
    - |
      W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W␣
↪ W  W  W  W  W  W  W  W
      W  g  .  .  .  .  .  .  .  .  .  .  .  .  W  W  g  .  .  .  .  .  .  .  .  .  .  .␣
↪ .  .  .  .  g  g  g  W
      W  .  .  .  .  .  .  .  .  .  .  .  .  .  W  W  .  .  .  .  .  .  .  .  .  .  .  .␣
↪ .  .  .  .  .  .  g  W
```


(continues on next page)

(continued from previous page)

```
W . . . . . . . . . . . . . N W W . . . . . . . . . . . . . . . . . . . W
W . . . . . . L . . . . . . . W W . . . . . . . . . . . . . . N . . W
W . . . . . . L . . . . . . . W W . . . . . . . . . . . . . . . . . . W
W . . . . . L L . . . . . . . . W W . . . . W W W W . . . . . . . . . W
W . . . . . L L . . . . . . . . W W . . . . W W W W . . . . . . . . . W
W . . . . . . L . . . . . . . . W W . . . . W W W W . . . . . . . . . W
W . . . . . . L . . . . . . . . . W . . . . . L L . . . . . . . . . . W
W . . . . . . L . . . . . . . . . W . . . . . . . . . . . . . . . . . W
W . . . . . . . . . . . . . . . . . . . . . . . . . . . W W W W W W W W
W . . . . . . . . . . . . . . . L . . . . . . . . . . W W W W W W W W W
W . . . . . . . . . . . W W W W L . . . . . . . . . . . W W W W W W W W
W A . . . . . . . . . . . W W W L . . . . . . . . . . . . W W W W W W W
W . . . . . . . . . . . . W W W L . . . . . . . . . . . . . W W W W W W
W . . . . . . . . . . . . . g W L g . . . . . . . . g . . . . . . L L W
W . . . . . . . . . . . . . W L L . . . . . . . . . . . . . . . g L L W
W . . . . . . . . . . . . . . L L . . . . . . . . . . . . . W W W W W W
W . . . . . . . W . . . . . . L L . . . . . . . . . . . . . . . . . . W
W . . . . . . . W . . . . . . L L . . . . . . . . . . . . . . . . . . W
W . . . . . . . W . . . . . . L . . . . . . . . . . . . . . . . . . . W
W . . . . . . . W . . . . . . . . . . . . . . . . . . . . . . . . . . W
W . . . . . . . W . . . . . . . . . . . . . . . . . . . . . . . . . . W
W . . . . . . . W . . . . . . . . . . . . . . . . . . . . . . . . . . W
W . . . . . . W W . . . . . . . . . . . . . . . . . . . . . . N . . . W
W . . . . . W W W . . . . . . . . . . . . . W W W W . . . . . . . . . W
W . . . . W W L W . . . . . . . . . . . . . . . . L . . . . . . . . . W
W . . . . W L L W . . . . . . . . . . . . . . . . L . . . . . . . . . W
```

(continues on next page)

(continued from previous page)

```
    W  .  .  .  W  W  L  L  W  .  .  .  .  .  .  .  .  .  .  .  .  .  .  L  L  L  .  .␣
↪ .  .  .  .  .  .  .  W
    W  .  .  W  W  L  L  L  W  .  .  .  .  .  .  .  .  .  .  .  .  .  L  L  L  L  .  .␣
↪ .  L  L  g  .  .  .  W
    W  g  W  W  W  W  W  W  W  N  .  .  .  .  .  .  .  .  .  .  .  .  L  L  L  L  g  .␣
↪ .  L  L  .  .  .  .  W
    W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W␣
↪ W  W  W  W  W  W  W  W

Actions:
  - Name: spawn_spider
    InputMapping:
      Internal: true
    Behaviours:
      - Src:
          Object: nest
          Commands:
            - spawn: spider
            - exec:
                Action: spawn_spider
                Delay: 50
                Randomize: true

        Dst:
          Object: _empty

      - Src:
          Object: nest
          Commands:
            - exec:
                Action: spawn_spider
                Delay: 50
                Randomize: true
        Dst:
          Object: [nest, spider, lava, wall, gnome, gem]


  # Define action that cannot be controlled by the player. (In this case the spider␣
↪movement)
  - Name: random_movement
    InputMapping:
      Inputs:
        1:
          Description: Rotate left
          OrientationVector: [-1, 0]
        2:
          Description: Move forwards
          OrientationVector: [0, -1]
          VectorToDest: [0, -1]
        3:
          Description: Rotate right
          OrientationVector: [1, 0]
```

(continues on next page)

(continued from previous page)

```
      Relative: true
      Internal: true
    Behaviours:
      # Spider rotates on the spot
      - Src:
          Object: spider
          Commands:
            - rot: _dir
            - exec:
                Action: random_movement
                Delay: 3
                Randomize: true
        Dst:
          Object: spider

      # The gnome and the spider can move into empty space
      - Src:
          Object: spider
          Commands:
            - mov: _dest
            - exec:
                Action: random_movement
                Delay: 3
                Randomize: true
        Dst:
          Object: _empty

      # The spider will not move into the wall or the gem, but it needs to keep moving
      - Src:
          Object: spider
          Commands:
            - exec:
                Action: random_movement
                Delay: 3
                Randomize: true
        Dst:
          Object: [wall, gem, nest]

      # If the spider runs into lava it dies
      - Src:
          Object: spider
          Commands:
            - remove: true
        Dst:
          Object: lava

      # If the gnome moves into a spider
      - Src:
          Object: spider
        Dst:
          Object: gnome
          Commands:
```

(continues on next page)

(continued from previous page)

```
          - remove: true
          - reward: -1

  # Define the move action
  - Name: move
    InputMapping:
      Inputs:
        1:
          Description: Rotate left
          OrientationVector: [-1, 0]
        2:
          Description: Move forwards
          OrientationVector: [0, -1]
          VectorToDest: [0, -1]
        3:
          Description: Rotate right
          OrientationVector: [1, 0]
        4:
          Description: Move Backwards
          VectorToDest: [0, 1]
          OrientationVector: [0, -1]
      Relative: true
    Behaviours:
      # Tell the gnome to rotate if it performs an action on itself (Rotate left and␣
↪Rotate right actions)
      - Src:
          Object: gnome
          Commands:
            - rot: _dir
        Dst:
          Object: gnome

      # If the gnome moves into a spider
      - Src:
          Object: gnome
          Commands:
            - remove: true
            - reward: -1
        Dst:
          Object: spider

       # If the gnome moves into lava
      - Src:
          Object: gnome
          Commands:
            - remove: true
            - reward: -1
        Dst:
          Object: lava

      # The gnome and the spider can move into empty space
      - Src:
```

(continues on next page)

(continued from previous page)

```
          Object: gnome
          Commands:
            - mov: _dest
        Dst:
          Object: _empty

      # If the gnome moves into a gem object, the stick is removed, triggering a win␣
↪condition
      - Src:
          Object: gnome
          Commands:
            - reward: 1
        Dst:
          Object: gem
          Commands:
            - remove: true

Objects:
  - Name: wall
    MapCharacter: 'W'
    Observers:
      Sprite2D:
        - TilingMode: WALL_16
          Image:
            - oryx/oryx_fantasy/wall8-0.png
            - oryx/oryx_fantasy/wall8-1.png
            - oryx/oryx_fantasy/wall8-2.png
            - oryx/oryx_fantasy/wall8-3.png
            - oryx/oryx_fantasy/wall8-4.png
            - oryx/oryx_fantasy/wall8-5.png
            - oryx/oryx_fantasy/wall8-6.png
            - oryx/oryx_fantasy/wall8-7.png
            - oryx/oryx_fantasy/wall8-8.png
            - oryx/oryx_fantasy/wall8-9.png
            - oryx/oryx_fantasy/wall8-10.png
            - oryx/oryx_fantasy/wall8-11.png
            - oryx/oryx_fantasy/wall8-12.png
            - oryx/oryx_fantasy/wall8-13.png
            - oryx/oryx_fantasy/wall8-14.png
            - oryx/oryx_fantasy/wall8-15.png
      Block2D:
        - Shape: square
          Color: [0.7, 0.7, 0.7]
          Scale: 1.0
      Isometric:
        - Image: oryx/oryx_iso_dungeon/wall-moss-1.png

  - Name: spider
    InitialActions:
      - Action: random_movement
        Delay: 3
        Randomize: true
```

(continues on next page)

(continued from previous page)

```
    MapCharacter: 'G'
    Observers:
      Sprite2D:
        - Image: oryx/oryx_fantasy/avatars/spider1.png
      Block2D:
        - Shape: triangle
          Color: [1.0, 0.0, 0.0]
          Scale: 0.8
      Isometric:
        - Image: oryx/oryx_iso_dungeon/avatars/spider-1.png

  - Name: gem
    MapCharacter: 'g'
    Observers:
      Sprite2D:
        - Image: oryx/oryx_fantasy/ore-6.png
      Block2D:
        - Shape: triangle
          Color: [0.0, 1.0, 0.0]
          Scale: 0.5
      Isometric:
        - Image: oryx/oryx_iso_dungeon/ore-6.png

  - Name: gnome
    MapCharacter: 'A'
    Observers:
      Sprite2D:
        - Image: oryx/oryx_fantasy/avatars/gnome1.png
      Block2D:
        - Shape: triangle
          Color: [0.0, 0.0, 1.0]
          Scale: 0.8
      Isometric:
        - Image: oryx/oryx_iso_dungeon/avatars/gnome-1.png

  - Name: nest
    MapCharacter: 'N'
    InitialActions:
      - Action: spawn_spider
        Delay: 10
        Randomize: true
    Observers:
      Sprite2D:
        - Image: oryx/oryx_fantasy/bush-1.png
      Block2D:
        - Shape: triangle
          Color: [0.0, 0.0, 1.0]
          Scale: 0.8
      Isometric:
        - Image: oryx/oryx_iso_dungeon/bush-1.png

  - Name: lava
```

(continues on next page)

(continued from previous page)

```
    MapCharacter: 'L'
    Observers:
      Sprite2D:
        - Image: oryx/oryx_fantasy/fire-1.png
      Block2D:
        - Shape: square
          Color: [1.0, 0.0, 0.0]
          Scale: 1.0
      Isometric:
        - Image: oryx/oryx_iso_dungeon/lava-1.png
          Offset: [0, 4]
          TilingMode: ISO_FLOOR
```

13.12 Partially Observable Labyrinth

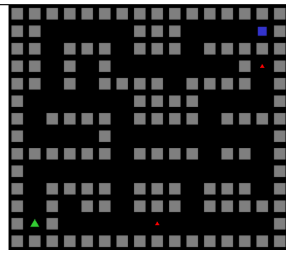
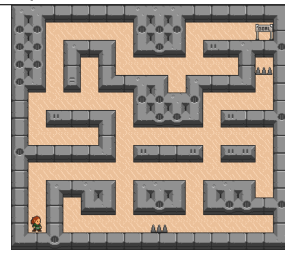
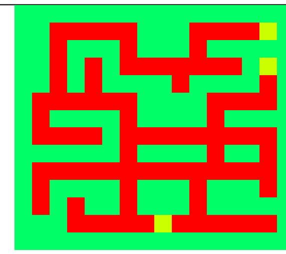
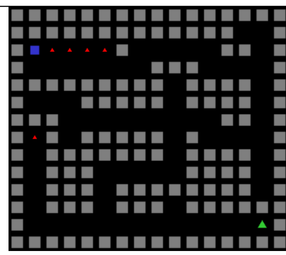
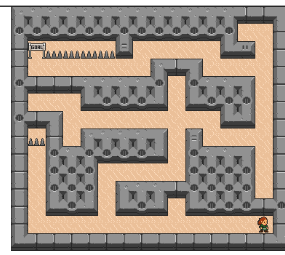
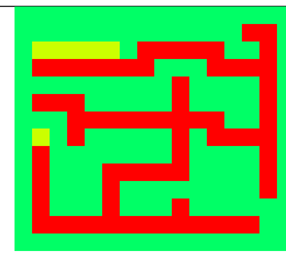
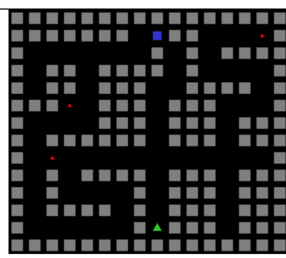
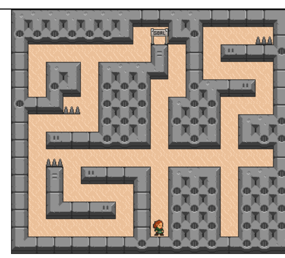
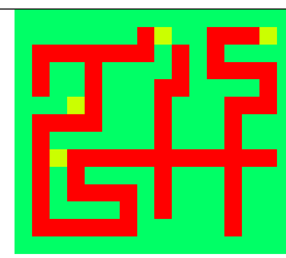
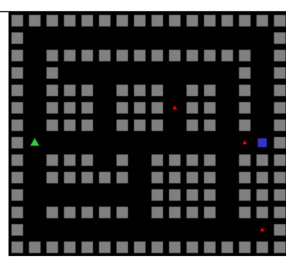
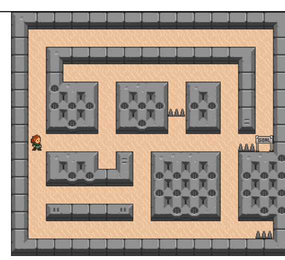
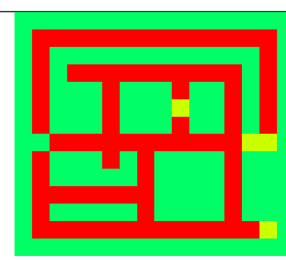
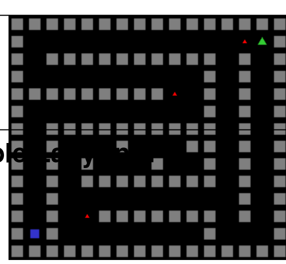
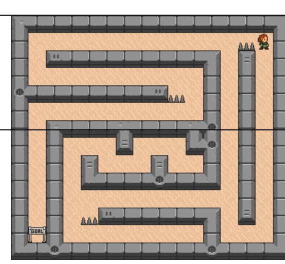
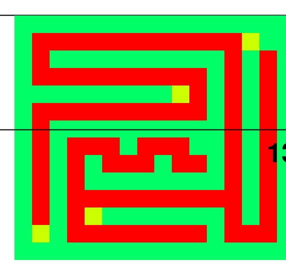
```
Single-Player/GVGAI/labyrinth_partially_observable.yaml
```

13.12.1 Description

Its a maze, find your way out. Watch out for spikey things. In this version the observation space for the player is partial.

13.12.2 Levels

Table 23: Levels

	Block2D	Sprite2D	Vector
Level ID: 0 Size: 16x14			
Level ID: 1 Size: 16x14			
Level ID: 2 Size: 16x14			
Level ID: 3 Size: 16x14			
			

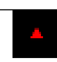
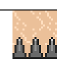
13.12.3 Code Example

The most basic way to create a Griddly Gym Environment. Defaults to level 0 and SPRITE_2D rendering.

```
import gym
import griddly

if __name__ == '__main__':

    env = gym.make('GDY-Partially-Observable-Labyrinth-v0')
    env.reset()

    # Replace with your own control algorithm!
    for s in range(1000):
        obs, reward, done, info = env.step(env.action_space.sample())
        env.render() # Renders the environment from the perspective of a single player

        env.render(observer='global') # Renders the entire environment

        if done:
            env.reset()
```

13.12.4 Objects

Table 24: Tiles

Name ->	avatar	exit	trap	wall
Map Char ->	*A*	*x*	*t*	*w*
Block2D				
Sprite2D				
Vector				

13.12.5 Actions

move

Relative The actions are calculated relative to the object being controlled.

Action Id	Mapping
1	Rotate left
2	Move forwards
3	Rotate right

13.12.6 YAML

```yaml
Version: "0.1"
Environment:
  Name: Partially Observable Labyrinth
  Description: Its a maze, find your way out. Watch out for spikey things. In this␣
↪version the observation space for the player is partial.
  Observers:
    Sprite2D:
      TileSize: 24
      BackgroundTile: gvgai/oryx/backBiege.png
  Player:
    Observer:
      RotateWithAvatar: true
      TrackAvatar: true
      Height: 5
      Width: 5
      OffsetX: 0
      OffsetY: 2
    AvatarObject: avatar
  Termination:
    Win:
      - eq: [exit:count, 0]
    Lose:
      - eq: [avatar:count, 0]
  Levels:
    - |
      w w w w w w w w w w w w w w w w
      w w . . . . . w w w . . . . x w
      w w . w w w . w w w . w w w w w
      w w . w . w . . . . . . . w t w
      w w . w . w w w w . w w w w . w
      w . . . . . . w w w w . . . . w
      w . w w w w . w w w w . w w w w
      w . . . . w . . . . . . . . . w
      w w w w w w . w w w w . w w . w
      w . . . . . . . . . . . . . . w
      w . w w w w . w w w . w w w . w
      w . w . w w . w w w . w w w w w
      w A w . . . . . t . . . . . . w
      w w w w w w w w w w w w w w w w
    - |
      w w w w w w w w w w w w w w w w
      w w w w w w w w w w w w w . . w
      w x t t t t w . . . . . w w . w
      w . . . . . . . w w w . . . . w
      w w w w w w w w w . w w w w . w
      w . . . w w w w w . w w w w . w
      w w w . . . . . . . . . w w . w
      w t w . w w w w w . w . . . . w
      w . w w w w w w w . w w w w . w
      w . w w w . . . . . w w w w . w
      w . w w w . w w w w w w w w . w
```

(continues on next page)

(continued from previous page)

```
      w . w w w . w w w . w w w w w w
      w . . . . . . . . . . . . . A w
      w w w w w w w w w w w w w w w w
    - |
      w w w w w w w w w w w w w w w w
      w w w w w w w . x w w . . . t w
      w . . . . . . . w . w . w w w w
      w . w w . w w w w . w . . . . w
      w . w w . w w w . . w w w w . w
      w w w t . w w w . w w w . . . w
      w . . . . w w w . w w w . w w w
      w . w w w w w w . w w w . w w w
      w . t . . . . . . . . . . . . w
      w . w . w w w w . w w w . w w w
      w . w . . . . w . w w w . w w w
      w . w w w w . w . w w w . w w w
      w . . . . . . w A w w w . w w w
      w w w w w w w w w w w w w w w w
    - |
      w w w w w w w w w w w w w w w w
      w . . . . . . . . . . . . . . w
      w . w w w w w w w w w w w w . w
      w . w . . . . . . . . . . w . w
      w . w w w . w w w . w w . w . w
      w . w w w . w w w t w w . w . w
      w . w w w . w w w . w w . w . w
      w A . . . . . . . . . . . t x w
      w . w w w . w . w w w w . w w w
      w . w w w w w . w w w w . w w w
      w . . . . . . . w w w w . w w w
      w . w w w w w . w w w w . w w w
      w . . . . . . . . . . . . . t w
      w w w w w w w w w w w w w w w w
    - |
      w w w w w w w w w w w w w w w w
      w . . . . . . . . . . . . t A w
      w . w w w w w w w w w w . w . w
      w . . . . . . . . . . w . w . w
      w w w w w w w w w t . w . w . w
      w . . . . . . . . . . w . w . w
      w . w w w w w w w w w w . w . w
      w . w . . . w . . . w w . w . w
      w . w . w . . . w . . w . w . w
      w . w . w w w w w w w w . w . w
      w . w . . . . . . . . . . w . w
      w . w . t w w w w w w w . w . w
      w x w . . . . . . . . w . . . w
      w w w w w w w w w w w w w w w w

Actions:
  # Define the move action
  - Name: move
```

(continues on next page)

(continued from previous page)

```
    InputMapping:
      Inputs:
        1:
          Description: Rotate left
          OrientationVector: [-1, 0]
        2:
          Description: Move forwards
          OrientationVector: [0, -1]
          VectorToDest: [0, -1]
        3:
          Description: Rotate right
          OrientationVector: [1, 0]
      Relative: true
    Behaviours:

      # Avatar rotates
      - Src:
          Object: avatar
          Commands:
            - rot: _dir
        Dst:
          Object: avatar

      # Avatar can move into empty space
      - Src:
          Object: avatar
          Commands:
            - mov: _dest
        Dst:
          Object: _empty

      # If Avatar hits a trap, remove it
      - Src:
          Object: avatar
          Commands:
            - remove: true
            - reward: -1
        Dst:
          Object: trap

      # If Avatar hits the exit, remove the exit
      - Src:
          Object: avatar
          Commands:
            - reward: 1
        Dst:
          Object: exit
          Commands:
            - remove: true

Objects:
  - Name: avatar
```

(continues on next page)

(continued from previous page)

```
    MapCharacter: A
    Observers:
      Sprite2D:
        - Image: gvgai/newset/girl1.png
      Block2D:
        - Shape: triangle
          Color: [0.2, 0.8, 0.2]
          Scale: 0.6

  - Name: exit
    MapCharacter: x
    Observers:
      Sprite2D:
        - Image: gvgai/newset/exit2.png
      Block2D:
        - Shape: square
          Color: [0.2, 0.2, 0.8]
          Scale: 0.7

  - Name: trap
    MapCharacter: t
    Observers:
      Sprite2D:
        - Image: gvgai/oryx/spike2.png
      Block2D:
        - Shape: triangle
          Color: [1.0, 0.0, 0.0]
          Scale: 0.3

  - Name: wall
    MapCharacter: w
    Observers:
      Sprite2D:
        - TilingMode: WALL_16
          Image:
            - oryx/oryx_fantasy/wall8-0.png
            - oryx/oryx_fantasy/wall8-1.png
            - oryx/oryx_fantasy/wall8-2.png
            - oryx/oryx_fantasy/wall8-3.png
            - oryx/oryx_fantasy/wall8-4.png
            - oryx/oryx_fantasy/wall8-5.png
            - oryx/oryx_fantasy/wall8-6.png
            - oryx/oryx_fantasy/wall8-7.png
            - oryx/oryx_fantasy/wall8-8.png
            - oryx/oryx_fantasy/wall8-9.png
            - oryx/oryx_fantasy/wall8-10.png
            - oryx/oryx_fantasy/wall8-11.png
            - oryx/oryx_fantasy/wall8-12.png
            - oryx/oryx_fantasy/wall8-13.png
            - oryx/oryx_fantasy/wall8-14.png
            - oryx/oryx_fantasy/wall8-15.png
      Block2D:
```

(continues on next page)

(continued from previous page)

```
        - Shape: square
          Color: [0.5, 0.5, 0.5]
          Scale: 0.9
```

13.13 Sokoban - 2

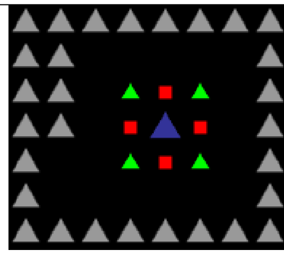
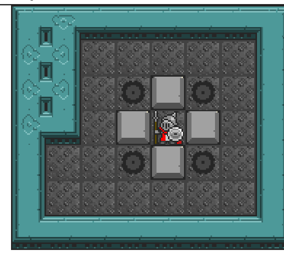
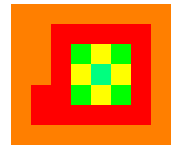
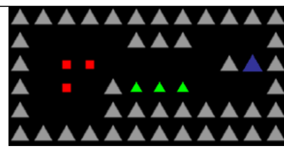
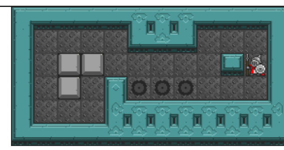
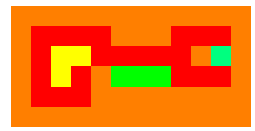
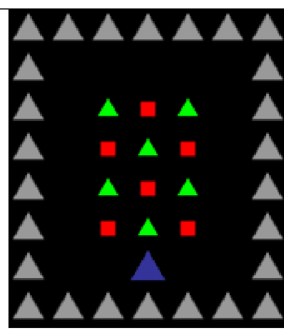
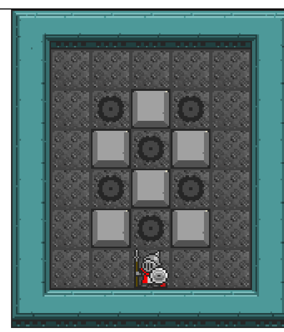
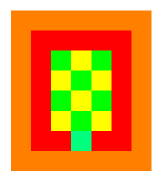
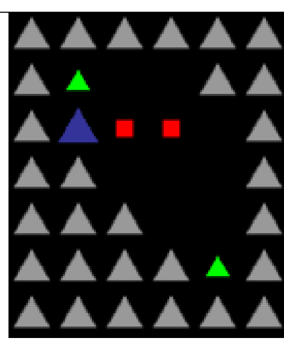
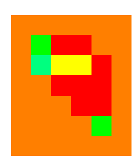
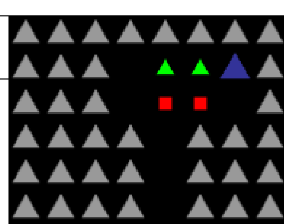
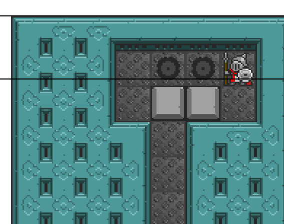
```
Single-Player/GVGAI/sokoban2.yaml
```

13.13.1 Description

Push the boxes onto the marked spaces, once a box has moved onto a space, it cannot be moved

13.13.2 Levels

Table 25: Levels

	Block2D	Sprite2D	Vector
Level ID: 0 Size: 8x7			
Level ID: 1 Size: 12x6			
Level ID: 2 Size: 7x8			
Level ID: 3 Size: 6x7			
			

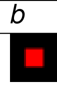
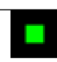
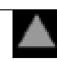
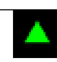
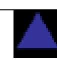
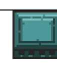
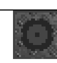
13.13.3 Code Example

The most basic way to create a Griddly Gym Environment. Defaults to level 0 and SPRITE_2D rendering.

```python
import gym
import griddly

if __name__ == '__main__':

    env = gym.make('GDY-Sokoban---2-v0')
    env.reset()

    # Replace with your own control algorithm!
    for s in range(1000):
        obs, reward, done, info = env.step(env.action_space.sample())
        env.render() # Renders the environment from the perspective of a single player

        env.render(observer='global') # Renders the entire environment

        if done:
            env.reset()
```

13.13.4 Objects

Table 26: Tiles

Name ->	box	box_in_place	wall	hole	avatar
Map Char ->	*b*	*f*	*w*	*h*	*A*
Block2D					
Sprite2D					
Vector					

13.13.5 Actions

move

Action Id	Mapping
1	Left
2	Up
3	Right
4	Down

13.13.6 YAML

```yaml
Version: "0.1"
Environment:
  Name: Sokoban - 2
  Description: Push the boxes onto the marked spaces, once a box has moved onto a space,␣
↪it cannot be moved
  Observers:
    Sprite2D:
      TileSize: 24
      BackgroundTile: gvgai/newset/floor2.png
  Player:
    AvatarObject: avatar # The player can only control a single avatar in the game
  Termination:
    Win:
      - eq: [box:count, 0] # If there are no boxes left
  Levels:
    - |
      wwwwwwwww
      ww.....w
      ww.hbh.w
      ww.bAb.w
      w..hbh.w
      w......w
      wwwwwwwww
    - |
      wwwwwwwwwwww
      w....www...w
      w.bb.....wAw
      w.b.whhh...w
      w...wwwwwwww
      wwwwwwwwwwww
    - |
      wwwwwww
      w.....w
      w.hbh.w
      w.bhb.w
      w.hbh.w
      w.bhb.w
      w..A..w
      wwwwwww
    - |
      wwwwww
      wh..ww
      wAbb.w
      ww...w
      www..w
      wwwwhw
      wwwwww
    - |
      wwwwwwww
      www.hhAw
      www.bb.w
```

(continues on next page)

(continued from previous page)

```
      wwww.www
      wwww.www
      wwww.www
      wwww.www
      w....www
      w.w...ww
      w...w.ww
      www...ww
      wwwwwwww

Actions:
  # Define the move action
  - Name: move
    Behaviours:
      # The agent can move around freely in empty space and over holes
      - Src:
          Object: avatar
          Commands:
            - mov: _dest
        Dst:
          Object: [_empty, hole]

      # Boxes can move into empty space
      - Src:
          Object: box
          Commands:
            - mov: _dest
        Dst:
          Object: _empty

      # The agent can push boxes
      - Src:
          Object: avatar
          Commands:
            - mov: _dest
        Dst:
          Object: [box, box_in_place]
          Commands:
            - exec:
                Action: move

      # If a box is moved into a hole, it should change to in-place box
      - Src:
          Object: [box, box_in_place]
          Commands:
            - mov: _dest
            - change_to: box_in_place
            - reward: 1
        Dst:
          Object: hole

      # If in-place box is moved into empty space, it should be a plain box
```

(continues on next page)

(continued from previous page)

```
      - Src:
          Object: box_in_place
          Commands:
            - mov: _dest
            - change_to: box
            - reward: -1
        Dst:
          Object: _empty

Objects:
  - Name: box
    Z: 2
    MapCharacter: b
    Observers:
      Sprite2D:
        - Image: gvgai/newset/block2.png
      Block2D:
        - Shape: square
          Color: [1.0, 0.0, 0.0]
          Scale: 0.5

  - Name: box_in_place
    Z: 2
    MapCharacter: f
    Observers:
      Sprite2D:
        - Image: gvgai/newset/block1.png
      Block2D:
        - Shape: square
          Color: [0.0, 1.0, 0.0]
          Scale: 0.5

  - Name: wall
    MapCharacter: w
    Observers:
      Sprite2D:
        - TilingMode: WALL_16
          Image:
            - gvgai/oryx/wall3_0.png
            - gvgai/oryx/wall3_1.png
            - gvgai/oryx/wall3_2.png
            - gvgai/oryx/wall3_3.png
            - gvgai/oryx/wall3_4.png
            - gvgai/oryx/wall3_5.png
            - gvgai/oryx/wall3_6.png
            - gvgai/oryx/wall3_7.png
            - gvgai/oryx/wall3_8.png
            - gvgai/oryx/wall3_9.png
            - gvgai/oryx/wall3_10.png
            - gvgai/oryx/wall3_11.png
            - gvgai/oryx/wall3_12.png
            - gvgai/oryx/wall3_13.png
```

(continues on next page)

(continued from previous page)

```
            - gvgai/oryx/wall3_14.png
            - gvgai/oryx/wall3_15.png
      Block2D:
        - Shape: triangle
          Color: [0.6, 0.6, 0.6]
          Scale: 0.9

  - Name: hole
    Z: 1
    MapCharacter: h
    Observers:
      Sprite2D:
        - Image: gvgai/oryx/cspell4.png
      Block2D:
        - Shape: triangle
          Color: [0.0, 1.0, 0.0]
          Scale: 0.6

  - Name: avatar
    Z: 2
    MapCharacter: A
    Observers:
      Sprite2D:
        - Image: gvgai/oryx/knight1.png
      Block2D:
        - Shape: triangle
          Color: [0.2, 0.2, 0.6]
          Scale: 1.0
```

13.14 Partially Observable Clusters

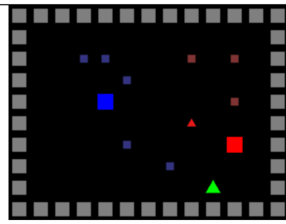
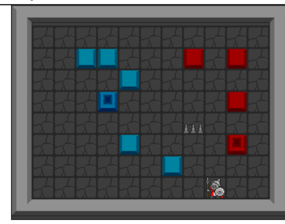
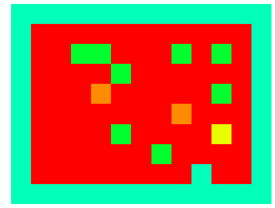
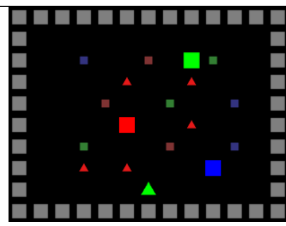
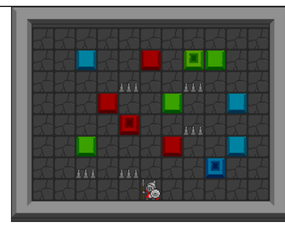
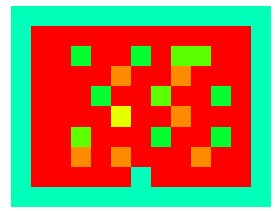
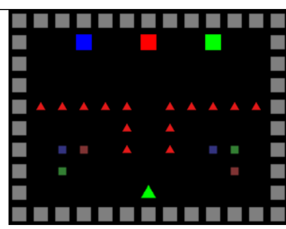
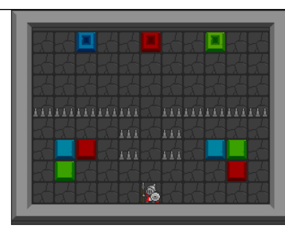
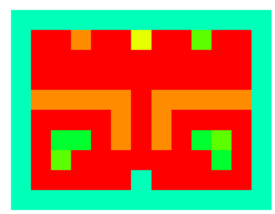
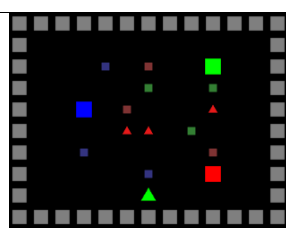
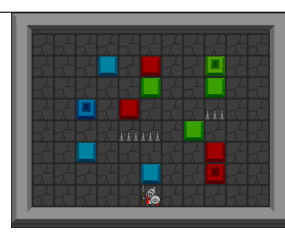
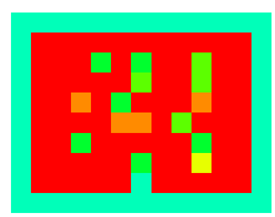
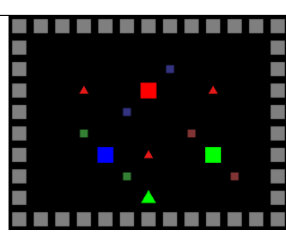
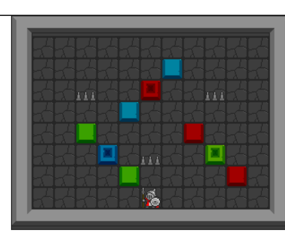
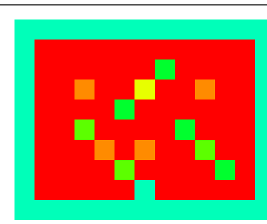
```
Single-Player/GVGAI/clusters_partially_observable.yaml
```

13.14.1 Description

Cluster the coloured objects together by pushing them against the static coloured blocks.

13.14.2 Levels

Table 27: Levels

	Block2D	Sprite2D	Vector
Level ID 0 Size 13x10			
Level ID 1 Size 13x10			
Level ID 2 Size 13x10			
Level ID 3 Size 13x10			
Level ID 4 Size 13x10			

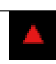
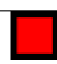
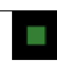
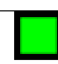
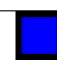
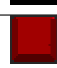
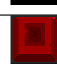
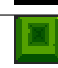
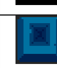
13.14.3 Code Example

The most basic way to create a Griddly Gym Environment. Defaults to level 0 and SPRITE_2D rendering.

```python
import gym
import griddly

if __name__ == '__main__':

    env = gym.make('GDY-Partially-Observable-Clusters-v0')
    env.reset()

    # Replace with your own control algorithm!
    for s in range(1000):
        obs, reward, done, info = env.step(env.action_space.sample())
        env.render() # Renders the environment from the perspective of a single player

        env.render(observer='global') # Renders the entire environment

        if done:
            env.reset()
```

13.14.4 Objects

Table 28: Tiles

Name ->	avatar	wall	spike	red_box	red_block	green_box	green_block	blue_box	blue_block
Map Char ->	*A*	*w*	*h*	*2*	*b*	*3*	*c*	*1*	*a*
Block2D									
Sprite2D									
Vector									

13.14.5 Actions

move

Relative The actions are calculated relative to the object being controlled.

Action Id	Mapping
1	Rotate left
2	Move forwards
3	Rotate right

box_counter

Internal This action can only be called from other actions, not by the player.

Action Id	Mapping
1	The only action here is to increment the box count

13.14.6 YAML

```yaml
Version: "0.1"
Environment:
  Name: Partially Observable Clusters
  Description: Cluster the coloured objects together by pushing them against the static␣
↪coloured blocks.
  Observers:
    Sprite2D:
      TileSize: 24
      BackgroundTile: oryx/oryx_fantasy/floor1-2.png
  Variables:
    - Name: box_count
      InitialValue: 0
  Player:
    Observer:
      RotateWithAvatar: true
      TrackAvatar: true
      Height: 5
      Width: 5
      OffsetX: 0
      OffsetY: 2
    AvatarObject: avatar # The player can only control a single avatar in the game
  Termination:
    Win:
      - eq: [box_count, 0]
    Lose:
      - eq: [broken_box:count, 1]
      - eq: [avatar:count, 0]
  Levels:
    - |
      w w w w w w w w w w w w w
      w . . . . . . . . . . . w
      w . . 1 1 . . . 2 . 2 . w
      w . . . . 1 . . . . . . w
      w . . . a . . . . . 2 . w
      w . . . . . . . h . . . w
      w . . . . 1 . . . . b . w
      w . . . . . . 1 . . . . w
      w . . . . . . . . A . . w
      w w w w w w w w w w w w w
    - |
      w w w w w w w w w w w w w
      w . . . . . . . . . . . w
```

(continues on next page)

(continued from previous page)

```
      w . . 1 . . 2 . c 3 . . w
      w . . . . h . . h . . . w
      w . . . 2 . . 3 . . 1 . w
      w . . . . b . . h . . . w
      w . . 3 . . . 2 . . 1 . w
      w . . h . h . . . a . . w
      w . . . . . . A . . . . w
      w w w w w w w w w w w w w
    - |
      w w w w w w w w w w w w w
      w . . a . . b . . c . . w
      w . . . . . . . . . . . w
      w . . . . . . . . . . . w
      w h h h h h . h h h h h w
      w . . . . h . h . . . . w
      w . 1 2 . h . h . 1 3 . w
      w . 3 . . . . . . . 2 . w
      w . . . . . . A . . . . w
      w w w w w w w w w w w w w
    - |
      w w w w w w w w w w w w w
      w . . . . . . . . . . . w
      w . . . 1 . 2 . . c . . w
      w . . . . . . 3 . . 3 . . w
      w . . a . 2 . . . h . . w
      w . . . . h h . 3 . . . w
      w . . 1 . . . . . 2 . . w
      w . . . . . . 1 . . b . . w
      w . . . . . . A . . . . w
      w w w w w w w w w w w w w
    - |
      w w w w w w w w w w w w w
      w . . . . . . . . . . . w
      w . . . . . . . 1 . . . . w
      w . . h . . b . . h . . w
      w . . . . 1 . . . . . . w
      w . . 3 . . . . 2 . . . w
      w . . . a . h . . c . . w
      w . . . . . 3 . . . . 2 . w
      w . . . . . . A . . . . w
      w w w w w w w w w w w w w

Actions:

  # A simple action to count the number of boxes in the game at the start
  # Not currently a way to do complex things in termination conditions like combine␣
↪multiple conditions
  - Name: box_counter
    InputMapping:
      Internal: true
      Inputs:
        1:
```

(continues on next page)

(continued from previous page)

```
        Description: "The only action here is to increment the box count"
  Behaviours:
    - Src:
        Object: [blue_box, red_box, green_box]
        Commands:
          - incr: box_count
      Dst:
        Object: [blue_box, red_box, green_box]

# Define the move action
- Name: move
  InputMapping:
    Inputs:
      1:
        Description: Rotate left
        OrientationVector: [-1, 0]
      2:
        Description: Move forwards
        OrientationVector: [0, -1]
        VectorToDest: [0, -1]
      3:
        Description: Rotate right
        OrientationVector: [1, 0]
    Relative: true
  Behaviours:

    # Avatar rotates
    - Src:
        Object: avatar
        Commands:
          - rot: _dir
      Dst:
        Object: avatar

    # Avatar and boxes can move into empty space
    - Src:
        Object: [avatar, blue_box, green_box, red_box]
        Commands:
          - mov: _dest
      Dst:
        Object: _empty

    # Boxes can be pushed by the avatar
    - Src:
        Object: avatar
        Commands:
          - mov: _dest
      Dst:
        Object: [blue_box, green_box, red_box]
        Commands:
          - cascade: _dest
```

(continues on next page)

(continued from previous page)

```
      # When boxes are pushed against the blocks they change
      - Src:
          Object: blue_box
          Commands:
            - change_to: blue_block
            - reward: 1
            - decr:  box_count
        Dst:
          Object: blue_block
      - Src:
          Object: red_box
          Commands:
            - reward: 1
            - change_to: red_block
            - decr:  box_count
        Dst:
          Object: red_block
      - Src:
          Object: green_box
          Commands:
            - reward: 1
            - change_to: green_block
            - decr:  box_count
        Dst:
          Object: green_block

      # Boxes break if they hit the spikes
      - Src:
          Object: [blue_box, green_box, red_box]
          Commands:
            - change_to: broken_box
            - reward: -1
        Dst:
          Object: spike

      # Avatar dies if it hits the spikes
      - Src:
          Object: avatar
          Commands:
            - remove: true
            - reward: -1
        Dst:
          Object: spike

Objects:
  - Name: avatar
    MapCharacter: A
    Observers:
      Sprite2D:
        - Image: gvgai/oryx/knight1.png
      Block2D:
        - Shape: triangle
```

(continues on next page)

(continued from previous page)

```
        Color: [0.0, 1.0, 0.0]
        Scale: 0.8

  - Name: wall
    MapCharacter: w
    Observers:
      Sprite2D:
        - TilingMode: WALL_16
          Image:
            - oryx/oryx_fantasy/wall1-0.png
            - oryx/oryx_fantasy/wall1-1.png
            - oryx/oryx_fantasy/wall1-2.png
            - oryx/oryx_fantasy/wall1-3.png
            - oryx/oryx_fantasy/wall1-4.png
            - oryx/oryx_fantasy/wall1-5.png
            - oryx/oryx_fantasy/wall1-6.png
            - oryx/oryx_fantasy/wall1-7.png
            - oryx/oryx_fantasy/wall1-8.png
            - oryx/oryx_fantasy/wall1-9.png
            - oryx/oryx_fantasy/wall1-10.png
            - oryx/oryx_fantasy/wall1-11.png
            - oryx/oryx_fantasy/wall1-12.png
            - oryx/oryx_fantasy/wall1-13.png
            - oryx/oryx_fantasy/wall1-14.png
            - oryx/oryx_fantasy/wall1-15.png
      Block2D:
        - Shape: square
          Color: [0.5, 0.5, 0.5]
          Scale: 0.9

  - Name: spike
    MapCharacter: h
    Observers:
      Sprite2D:
        - Image: gvgai/oryx/spike2.png
      Block2D:
        - Shape: triangle
          Color: [0.9, 0.1, 0.1]
          Scale: 0.5

  - Name: red_box
    MapCharacter: "2"
    InitialActions:
      - Action: box_counter
        ActionId: 1
    Observers:
      Sprite2D:
        - Image: gvgai/newset/blockR.png
      Block2D:
        - Shape: square
          Color: [0.5, 0.2, 0.2]
          Scale: 0.5
```

(continues on next page)

(continued from previous page)

```
  - Name: red_block
    MapCharacter: b
    Observers:
      Sprite2D:
        - Image: gvgai/newset/blockR2.png
      Block2D:
        - Shape: square
          Color: [1.0, 0.0, 0.0]
          Scale: 1.0

  - Name: green_box
    MapCharacter: "3"
    InitialActions:
      - Action: box_counter
        ActionId: 1
    Observers:
      Sprite2D:
        - Image: gvgai/newset/blockG.png
      Block2D:
        - Shape: square
          Color: [0.2, 0.5, 0.2]
          Scale: 0.5
  - Name: green_block
    MapCharacter: c
    Observers:
      Sprite2D:
        - Image: gvgai/newset/blockG2.png
      Block2D:
        - Shape: square
          Color: [0.0, 1.0, 0.0]
          Scale: 1.0

  - Name: blue_box
    MapCharacter: "1"
    InitialActions:
      - Action: box_counter
        ActionId: 1
    Observers:
      Sprite2D:
        - Image: gvgai/newset/blockB.png
      Block2D:
        - Shape: square
          Color: [0.2, 0.2, 0.5]
          Scale: 0.5
  - Name: blue_block
    MapCharacter: a
    Observers:
      Sprite2D:
        - Image: gvgai/newset/blockB2.png
      Block2D:
        - Shape: square
          Color: [0.0, 0.0, 1.0]
```

(continues on next page)

(continued from previous page)

```yaml
        Scale: 1.0

  - Name: broken_box
    Observers:
      Sprite2D:
        - Image: gvgai/newset/block3.png
      Block2D:
        - Shape: triangle
          Color: [1.0, 0.0, 1.0]
          Scale: 1.0
```

13.15 Bait With Keys

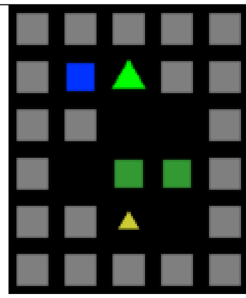
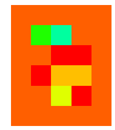
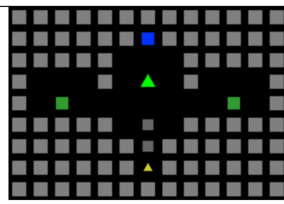
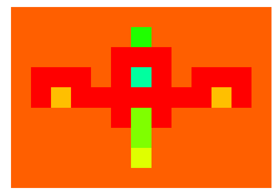
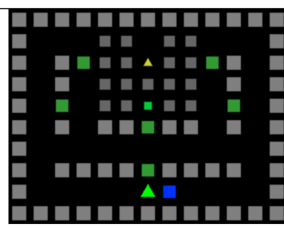
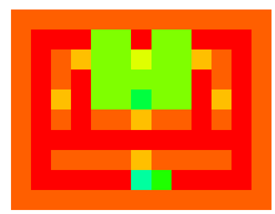
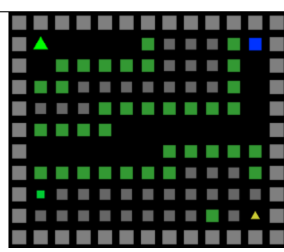
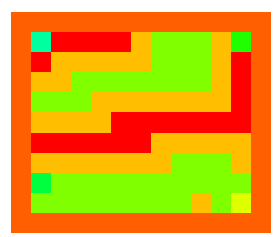
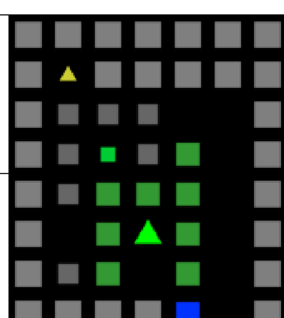
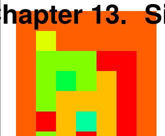
```
Single-Player/GVGAI/bait_keys.yaml
```

13.15.1 Description

Get the key and unlock the door. Fill in the holes in the floor with blocks to get to the key. (This environment is the same as the normal Bait environment, but if the avatar has the key, it is visible)

13.15.2 Levels

Table 29: Levels

	Block2D	Sprite2D	Vector
Level ID 0 Size 5x6			
Level ID 1 Size 13x9			
Level ID 2 Size 13x10			
Level ID 3 Size 13x11			
			

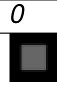
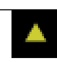
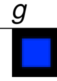
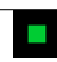
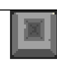
13.15.3 Code Example

The most basic way to create a Griddly Gym Environment. Defaults to level 0 and SPRITE_2D rendering.

```python
import gym
import griddly

if __name__ == '__main__':

    env = gym.make('GDY-Bait-With-Keys-v0')
    env.reset()

    # Replace with your own control algorithm!
    for s in range(1000):
        obs, reward, done, info = env.step(env.action_space.sample())
        env.render() # Renders the environment from the perspective of a single player

        env.render(observer='global') # Renders the entire environment

        if done:
            env.reset()
```

13.15.4 Objects

Table 30: Tiles

Name ->	avatar	hole	box	key	goal	mushroom	wall
Map Char ->	*A*	*0*	*1*	*k*	*g*	*m*	*w*
Block2D							
Sprite2D							
Vector							

13.15.5 Actions

move

Action Id	Mapping
1	Left
2	Up
3	Right
4	Down

13.15.6 YAML

```yaml
# http://www.mobygames.com/game/bait

Version: "0.1"
Environment:
  Name: Bait With Keys
  Description: Get the key and unlock the door. Fill in the holes in the floor with␣
↪blocks to get to the key. (This environment is the same as the normal Bait environment,
↪ but if the avatar has the key, it is visible)
  Observers:
    Sprite2D:
      TileSize: 24
      BackgroundTile: gvgai/oryx/backLBrown.png
  Player:
    AvatarObject: avatar
  Termination:
    Lose:
      - eq: [avatar:count, 0]
    Win:
      - eq: [goal:count, 0]
  Levels:
    - |
      w w w w w
      w g A w w
      w w . . w
      w . 1 1 w
      w w k . w
      w w w w w
    - |
      w w w w w w w w w w w w w
      w w w w w w g w w w w w w
      w w w w w . . . w w w w w
      w . . . w . A . w . . . w
      w . 1 . . . . . . . 1 . w
      w w w w w . 0 . w w w w w
      w w w w w w 0 w w w w w w
      w w w w w w k w w w w w w
      w w w w w w w w w w w w w
    - |
      w w w w w w w w w w w w w
      w . . . 0 0 . 0 0 . . . w
      w . w 1 0 0 k 0 0 1 w . w
      w . w . 0 0 0 0 0 . w . w
      w . 1 . 0 0 m 0 0 . 1 . w
      w . w . w w 1 w w . w . w
      w . . . . . . . . . . . w
      w . w w w w 1 w w w w . w
      w . . . . . A g . . . . w
      w w w w w w w w w w w w w
    - |
      w w w w w w w w w w w w w
      w A . . . . 1 0 0 0 1 g w
```

(continues on next page)

(continued from previous page)

```
      w . 1 1 1 1 1 0 0 0 1 . w
      w 1 1 0 0 0 0 0 0 0 1 . w
      w 0 0 0 1 1 1 1 1 1 1 . w
      w 1 1 1 1 . . . . . . . w
      w . . . . . . 1 1 1 1 1 w
      w 1 1 1 1 1 1 1 0 0 0 1 w
      w m 0 0 0 0 0 0 0 0 0 0 w
      w 0 0 0 0 0 0 0 0 1 0 k w
      w w w w w w w w w w w w w
    - |
      w w w w w w w
      w k w w w w w
      w 0 0 0 . . w
      w 0 m 0 1 . w
      w 0 1 1 1 . w
      w . 1 A 1 . w
      w 0 1 . 1 . w
      w w w w g . w
      w w w w w w w

Actions:
  # Define the move action
  - Name: move
    Behaviours:
    # Avatar and boxes can move into empty space
      - Src:
          Object: [avatar, box]
          Commands:
            - mov: _dest
        Dst:
          Object: _empty

      # Boxes can be pushed by the avatar
      - Src:
          Object: [avatar]
          Commands:
            - mov: _dest
        Dst:
          Object: box
          Commands:
            - cascade: _dest

      # If a box falls into a hole, both disappear
      - Src:
          Object: box
          Commands:
            - remove: true
            - reward: 1
        Dst:
          Object: hole
          Commands:
            - remove: true
```

(continues on next page)

(continued from previous page)

```
      # If the avatar falls into a hole remove the avatar
      - Src:
          Object: [avatar]
          Commands:
            - remove: true
            - reward: -1
        Dst:
          Object: hole

      # If the avatar picks up a mushroom, remove the mushroom
      - Src:
          Object: [avatar]
          Commands:
            - reward: 1
            - mov: _dest
        Dst:
          Object: mushroom
          Commands:
            - remove: true

      # Only an avatar with a key can
      - Src:
          Preconditions:
            - eq: [has_key, 1]
          Object: avatar
          Commands:
            - reward: 5
        Dst:
          Object: goal
          Commands:
            - remove: true

      # Avatar picks up the key
      - Src:
          Object: avatar
          Commands:
            - mov: _dest
            - incr: has_key
            - set_tile: 1
        Dst:
          Object: key
          Commands:
            - remove: true

Objects:
  - Name: avatar
    MapCharacter: A
    Variables:
      - Name: has_key
    Observers:
      Sprite2D:
```

(continues on next page)

(continued from previous page)

```
      - Image: gvgai/oryx/swordman1_0.png
      - Image: gvgai/oryx/swordmankey1_0.png
    Block2D:
      - Shape: triangle
        Color: [0.0, 1.0, 0.0]
        Scale: 0.8
      - Shape: triangle
        Color: [0.0, 1.0, 0.0]
        Scale: 0.9

- Name: hole
  MapCharacter: "0"
  Observers:
    Sprite2D:
      - Image: gvgai/newset/hole1.png
    Block2D:
      - Shape: square
        Color: [0.4, 0.4, 0.4]
        Scale: 0.7

- Name: box
  MapCharacter: "1"
  Observers:
    Sprite2D:
      - Image: gvgai/newset/block3.png
    Block2D:
      - Shape: square
        Color: [0.2, 0.6, 0.2]
        Scale: 0.8

- Name: key
  MapCharacter: k
  Observers:
    Sprite2D:
      - Image: gvgai/oryx/key2.png
    Block2D:
      - Shape: triangle
        Color: [0.8, 0.8, 0.2]
        Scale: 0.5

- Name: goal
  MapCharacter: g
  Observers:
    Sprite2D:
      - Image: gvgai/oryx/doorclosed1.png
    Block2D:
      - Shape: square
        Color: [0.0, 0.2, 1.0]
        Scale: 0.8

- Name: mushroom
  MapCharacter: m
```

(continues on next page)

(continued from previous page)

```
    Observers:
      Sprite2D:
        - Image: gvgai/oryx/mushroom2.png
      Block2D:
        - Shape: square
          Color: [0.0, 0.8, 0.2]
          Scale: 0.5

  - Name: wall
    MapCharacter: w
    Observers:
      Sprite2D:
        - TilingMode: WALL_16
          Image:
            - gvgai/oryx/dirtWall_0.png
            - gvgai/oryx/dirtWall_1.png
            - gvgai/oryx/dirtWall_2.png
            - gvgai/oryx/dirtWall_3.png
            - gvgai/oryx/dirtWall_4.png
            - gvgai/oryx/dirtWall_5.png
            - gvgai/oryx/dirtWall_6.png
            - gvgai/oryx/dirtWall_7.png
            - gvgai/oryx/dirtWall_8.png
            - gvgai/oryx/dirtWall_9.png
            - gvgai/oryx/dirtWall_10.png
            - gvgai/oryx/dirtWall_11.png
            - gvgai/oryx/dirtWall_12.png
            - gvgai/oryx/dirtWall_13.png
            - gvgai/oryx/dirtWall_14.png
            - gvgai/oryx/dirtWall_15.png
      Block2D:
        - Shape: square
          Color: [0.5, 0.5, 0.5]
          Scale: 0.9
```

13.16 Random butterflies

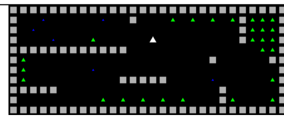
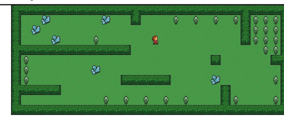
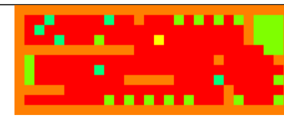
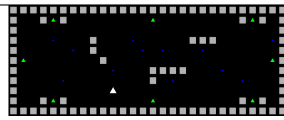
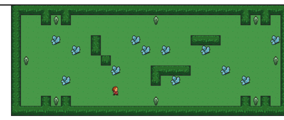
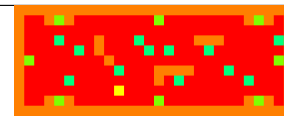
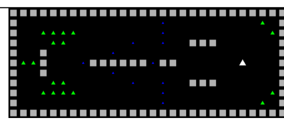
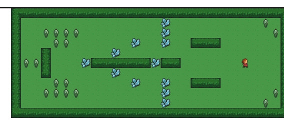
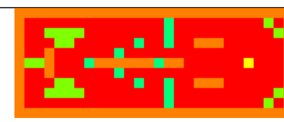
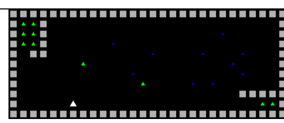
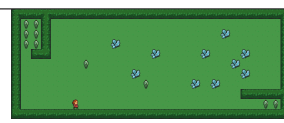
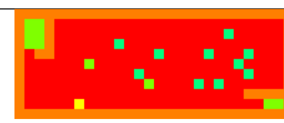
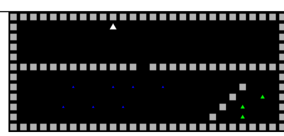
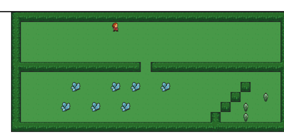
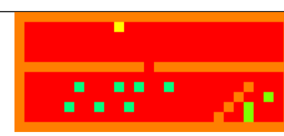
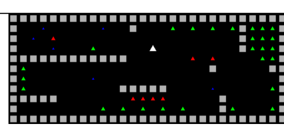
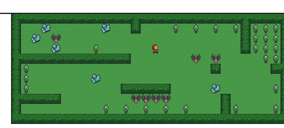
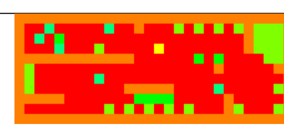
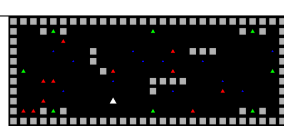
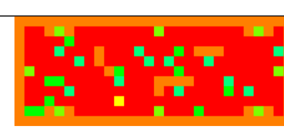
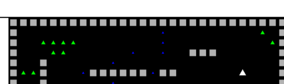
```
Single-Player/GVGAI/random_butterflies.yaml
```

13.16.1 Description

You want to catch all of the butterflies while also avoiding the spiders. Butterflies and spiders spawn randomly. The butterflies are also eaten by the spiders so you need to be fast to collect them. You win the level as soon as there are no butterflies left. The player also only has partial observability.

13.16.2 Levels

Table 31: Levels

	Block2D	Sprite2D	Vector
Level ID 0 Size 28x11			
Level ID 1 Size 28x11			
Level ID 2 Size 28x11			
Level ID 3 Size 28x11			
Level ID 4 Size 28x12			
Level ID 5 Size 28x11			
Level ID 6 Size 28x11			
			

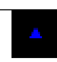
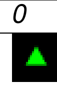
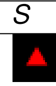
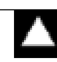
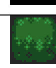
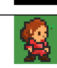
13.16.3 Code Example

The most basic way to create a Griddly Gym Environment. Defaults to level 0 and SPRITE_2D rendering.

```
import gym
import griddly

if __name__ == '__main__':

    env = gym.make('GDY-Random-butterflies-v0')
    env.reset()

    # Replace with your own control algorithm!
    for s in range(1000):
        obs, reward, done, info = env.step(env.action_space.sample())
        env.render() # Renders the environment from the perspective of a single player

        env.render(observer='global') # Renders the entire environment

        if done:
            env.reset()
```

13.16.4 Objects

Table 32: Tiles

Name ->	wall	butterfly	cocoon	spider	catcher
Map Char ->	*w*	*1*	*0*	*S*	*A*
Block2D					
Sprite2D					
Vector					

13.16.5 Actions

move

Action Id	Mapping
1	Left
2	Up
3	Right
4	Down

spider_random_movement

Relative The actions are calculated relative to the object being controlled.

Internal This action can only be called from other actions, not by the player.

Action Id	Mapping
1	Rotate left
2	Move forwards
3	Rotate right

spawn_butterfly

Internal This action can only be called from other actions, not by the player.

MapToGrid This action is mapped to any grid location.

butterfly_random_movement

Internal This action can only be called from other actions, not by the player.

Action Id	Mapping
1	Left
2	Up
3	Right
4	Down

13.16.6 YAML

```yaml
Version: "0.1"
Environment:
  Name: Random butterflies
  Description: |
    You want to catch all of the butterflies while also avoiding the spiders. Butterflies and spiders spawn randomly.
    The butterflies are also eaten by the spiders so you need to be fast to collect them.
    You win the level as soon as there are no butterflies left.
    The player also only has partial observability.
  Observers:
    Sprite2D:
      TileSize: 24
      BackgroundTile: gvgai/oryx/grass_15.png
    Block2D:
      TileSize: 24
  Player:
    AvatarObject: catcher
    Observer:
      TrackAvatar: true
      Height: 7
      Width: 7
```

(continues on next page)

(continued from previous page)

```
      OffsetX: 0
      OffsetY: 0
  Termination:
    Win:
      - eq: [butterfly:count, 0] # If there are no butterflies
    Lose:
      - eq: [catcher:count, 0] # If the catcher gets killed
  Levels:
    - |
      w w w w w w w w w w w w w w w w w w w w w w w w w w w w
      w . . 1 . . . . . 1 . . w . . . 0 . 0 . 0 . 0 w 0 0 0 w
      w . 1 . . . . . . . . . . . . . . . . . . . . w 0 0 0 w
      w . . . 1 . . . 0 . . . . . A . . . . . . . . w 0 0 0 w
      w w w w w w w w w w w w . . . . . . . . . . . . . 0 0 w
      w 0 . . . . . . . . . . . . . . . . . . w . . . . . w w
      w 0 . . . . . . 1 . . . . . . . . . . . . . . . . . . w
      w 0 . . . . . . . . . w w w w w . . . . 1 . . . . . 0 w
      w w w w w . . . . . . . . . . . . . . . . w . . . . . w
      w . . . . . . . . 0 . 0 . 0 . 0 . 0 . . . w 0 . . . 0 w
      w w w w w w w w w w w w w w w w w w w w w w w w w w w w
    - |
      w w w w w w w w w w w w w w w w w w w w w w w w w w w w
      w . . w 0 w . . . . . . . . 0 . . . . . . . . w 0 w . w
      w . . . . . . . . . . . . . . . . . . . . . . . . . . w
      w . . . 1 . . . w . . . 1 . . . . . w w w . . . . . 1 w
      w . . . . . 1 . w . . . . 1 . 1 . . . 1 . . . . . . . w
      w 0 . . . . . . . w . . . . . . . . . . . . . . . . 0 w
      w . . . . . . . . . 1 . . . w w w w . . . 1 . . . . . w
      w . . . . 1 . . . . . . . . w . 1 . . . . . . 1 . . . w
      w . . . . . . . . . A . . . . . . . . . . . . . . . . w
      w . . w 0 w . . . . . . . . 0 . . . . . . . . w 0 w . w
      w w w w w w w w w w w w w w w w w w w w w w w w w w w w
    - |
      w w w w w w w w w w w w w w w w w w w w w w w w w w w w
      w . . . . . . . . . . . . . . 1 . . . . . . . . . 0 . w
      w . . 0 0 0 0 . . . . . . . . 1 . . . . . . . . . . 0 w
      w . . . 0 0 . . . . . . 1 . . 1 . . w w w . . . . . . w
      w . . w . . . . . . 1 . . . . . . . . . . . . . . . . w
      w 0 0 w . . . 1 w w w w w w 1 w w . . . . . . A . . . w
      w . . w . . . . . . 1 . . . . . . . . . . . . . . . . w
      w . . . 0 0 . . . . . . 1 . . 1 . . w w w . . . . . . w
      w . . 0 0 0 0 . . . . . . . . 1 . . . . . . . . . . 0 w
      w . . . . . . . . . . . . . . 1 . . . . . . . . . 0 . w
      w w w w w w w w w w w w w w w w w w w w w w w w w w w w
    - |
      w w w w w w w w w w w w w w w w w w w w w w w w w w w w
      w 0 0 w . . . . . . . . . . . . . . . . . . . . . . . w
      w 0 0 w . . . . . . . . . . . . . . . . . 1 . . . . . w
      w 0 0 w . . . . . . 1 . . . . . . . . . . . . . . . . w
      w . w w . . . . . . . . . . 1 . . . . 1 . . . 1 . . . w
      w . . . . . . 0 . . . . . . . . . . . . . . 1 . . . . w
      w . . . . . . . . . . . 1 . . . . . . . . . . 1 . . . w
```

(continues on next page)

(continued from previous page)

```
      w . . . . . . . . . . . . 0 . . . . 1 . 1 . . . . . . w
      w . . . . . . . . . . . . . . . . . . . . . . w w w w w
      w . . . . . A . . . . . . . . . . . . . . . . . . 0 0 w
      w w w w w w w w w w w w w w w w w w w w w w w w w w w w
    - |
      w w w w w w w w w w w w w w w w w w w w w w w w w w w w
      w . . . . . . . . . A . . . . . . . . . . . . . . . . w
      w . . . . . . . . . . . . . . . . . . . . . . . . . . w
      w . . . . . . . . . . . . . . . . . . . . . . . . . . w
      w . . . . . . . . . . . . . . . . . . . . . . . . . . w
      w w w w w w w w w w w w w . w w w w w w w w w w w w w w
      w . . . . . . . . . . . . . . . . . . . . . . . . . . w
      w . . . . . 1 . . . 1 . 1 . . 1 . . . . . . . w . . . w
      w . . . . . . . . . . . . . . . . . . . . . w . . 0 . w
      w . . . . 1 . . 1 . . 1 . . . . . . . . . w . 0 . . . w
      w . . . . . . . . . . . . . . . . . . . w . . 0 . . . w
      w w w w w w w w w w w w w w w w w w w w w w w w w w w w
    - |
      w w w w w w w w w w w w w w w w w w w w w w w w w w w w
      w . . 1 . . . . . 1 . . w . . . 0 . 0 . 0 . 0 w 0 0 0 w
      w . 1 . S . . . . . . . . . . . . . . . . . . w 0 0 0 w
      w . . . 1 . . . 0 . . . . . A . . . . . . . . w 0 0 0 w
      w w w w w w w w w w w w . . . . . . S . S . . . . 0 0 w
      w 0 . . . . . . . . . . . . . . . . . . w . . . . . w w
      w 0 . . . . . . 1 . . . . . . . . . . . . . . . . . . w
      w 0 . . . . . . . . . w w w w w . . . . 1 . . . . . 0 w
      w w w w w . . . . . . . S S S S . . . . . w . . . . . w
      w . . . . . . . . 0 . 0 . 0 . 0 . 0 . . . w 0 . . . 0 w
      w w w w w w w w w w w w w w w w w w w w w w w w w w w w
    - |
      w w w w w w w w w w w w w w w w w w w w w w w w w w w w
      w . . w 0 w . . . . . . . . 0 . . . . . . . . w 0 w . w
      w . . . . S . . . . . . . . . . . . . . . . . . . . . w
      w . . . 1 . . . w . . . 1 . . . S . w w w . . . . . 1 w
      w . . . . . 1 . w . . . . 1 . 1 . . . 1 . . . . . . . w
      w 0 . . . . . . . w S . . . . . S . . . . . . . . . 0 w
      w . . S S . . . . . 1 . . . w w w w . . . 1 . . . . . w
      w . . . . 1 . . . . . . . . w . 1 . . . . S . 1 . . . w
      w . . S . . . . . . A . . . . . . . . . . . . . . . . w
      w S S w 0 w . . . . . . . . 0 . . . S . . . . w 0 w . w
      w w w w w w w w w w w w w w w w w w w w w w w w w w w w
    - |
      w w w w w w w w w w w w w w w w w w w w w w w w w w w w
      w . . . . . . . . . . . . . . 1 . . . . . . . . . 0 . w
      w . . 0 0 0 0 . . . . . . . . 1 . . . . . . . . . . 0 w
      w . . . 0 0 . . . . . . 1 . . 1 . . w w w . . . . . . w
      w . . w . . . . . . 1 . . . . . . . . . . . . . . . . w
      w 0 0 w . . . 1 w w w w w w 1 w w . . . . . . A . . . w
      w . . w . . . . . . 1 . . . . . . . . . . . . . . . . w
      w . . . 0 0 . . . . . . 1 . . 1 . . w w w . . . . . . w
      w . . 0 0 0 0 . . . . . . . . 1 . . . . . . . . . . 0 w
      w . . . . . . . . . . . . . . 1 . . . . . . . . . 0 . w
```

(continues on next page)

(continued from previous page)

```yaml
      w w w w w w w w w w w w w w w w w w w w w w w w w w w w
    - |
      w w w w w w w w w w w w w w w w w w w w w w w w w w w w
      w 0 0 w . . . . . . . . . . . . . . . . . . . . . . . w
      w 0 0 w . . . . . . . . . . . . S . . . . 1 . . . . . w
      w 0 0 w . . . . . . 1 . . . . . S . . . . . . . . . . w
      w . w w . . . . . . . . . . 1 . S . . 1 . . . 1 . . . w
      w . . . . . . 0 . . . . . . . . S . . . . . 1 . . . . w
      w . . . . . . . . . . . 1 . . . S . . . . . . 1 . . . w
      w . . . . . . . . . . . . 0 . . S . 1 . 1 . . . . . . w
      w . . . . . . . . . . . . . . . S . . . . . . w w w w w
      w . . . . . A . . . . . . . . . . . . . . . . . . 0 0 w
      w w w w w w w w w w w w w w w w w w w w w w w w w w w w
    - |
      w w w w w w w w w w w w w w w w w w w w w w w w w w w w
      w . . . S . . . . . A . . . . . . . . . S . . . . . . w
      w . . . . . . . . . . . . . . . S . . . . . . S . . . w
      w . . . . . . S . . S . . . . . . . . . . . . . . . . w
      w . . . . . . . . . . . . . . . S . . . . . S . . . . w
      w w w w w w w w w w w w w . w w w w w w w w w w w w w w
      w . . . . . . . . . . . . . . . . . . . . . . . . . . w
      w . . . . . 1 . . . 1 . 1 . . 1 . . . . . . . w . . . w
      w . . . . . . . . . . . . . . . . . . . . . w . . 0 . w
      w . . . . 1 . . 1 . . 1 . . . . . . . . . w . 0 . . . w
      w . . . . . . . . . . . . . . . . . . . w . . 0 . . . w
      w w w w w w w w w w w w w w w w w w w w w w w w w w w w

Actions:
  - Name: spawn_butterfly
    InputMapping:
      Internal: true
      MapToGrid: true
    Behaviours:
      - Src:
          Object: cocoon
          Commands:
            - spawn: butterfly
            - exec:
                Action: spawn_butterfly
                Delay: 200
                Randomize: true

        Dst:
          Object: _empty

      - Src:
          Object: cocoon
          Commands:
            - exec:
                Action: spawn_butterfly
                Delay: 3
                Randomize: true
```

(continues on next page)

(continued from previous page)

```
        Dst:
          Object: [cocoon, butterfly, catcher, wall]

  # Butterfly movement is different to spider movement
  - Name: butterfly_random_movement
    InputMapping:
      Internal: true
    Behaviours:

      # The butterfly moves into an empty space
      - Src:
          Object: butterfly
          Commands:
            - mov: _dest
            - exec:
                Action: butterfly_random_movement
                Delay: 3
                Randomize: true
        Dst:
          Object: _empty

      # if the butterfly tries to move into anything but an empty spot
      - Src:
          Object: butterfly
          Commands:
            - exec:
                Action: butterfly_random_movement
                Delay: 3
                Randomize: true
        Dst:
          Object: [ wall, spider, catcher, butterfly, cocoon ]

  # Define spider movement
  - Name: spider_random_movement
    InputMapping:
      Inputs:
        1:
          Description: Rotate left
          OrientationVector: [-1, 0]
        2:
          Description: Move forwards
          OrientationVector: [0, -1]
          VectorToDest: [0, -1]
        3:
          Description: Rotate right
          OrientationVector: [1, 0]
      Relative: true
      Internal: true
    Behaviours:
      # Spider rotates on the spot
      - Src:
          Object: spider
```

(continues on next page)

(continued from previous page)

```
        Commands:
          - rot: _dir
          - exec:
              Action: spider_random_movement
              Delay: 3
              Randomize: true
      Dst:
        Object: spider

    # The catcher and the spider can move into empty space
    - Src:
        Object: spider
        Commands:
          - mov: _dest
          - exec:
              Action: spider_random_movement
              Delay: 3
              Randomize: true
      Dst:
        Object: _empty

    # The spider will not move into the wall or the gem, but it needs to keep moving
    - Src:
        Object: spider
        Commands:
          - exec:
              Action: spider_random_movement
              Delay: 3
              Randomize: true
      Dst:
        Object: [wall, cocoon]

    # If the spider moves into a butterfly it dies
    - Src:
        Object: spider
        Commands:
          - mov: _dest
          - exec:
              Action: spider_random_movement
              Delay: 3
              Randomize: true
      Dst:
        Object: butterfly
        Commands:
          - remove: true

    # if the spider moves into the catcher it dies
    - Src:
        Object: spider
      Dst:
        Object: catcher
        Commands:
```

(continues on next page)

(continued from previous page)

```
          - remove: true
          - reward: -10

  # Define the move action
  - Name: move
    Behaviours:

      # If the catcher moves into a spider
      - Src:
          Object: catcher
          Commands:
            - remove: true
            - reward: -1
        Dst:
          Object: spider

      # The catcher move into an empty space
      - Src:
          Object: catcher
          Commands:
            - mov: _dest
        Dst:
          Object: _empty

      # If the catcher moves into a butterfly object, the butterfly is caught YAY!
      - Src:
          Object: catcher
          Commands:
            - mov: _dest
            - reward: 1
        Dst:
          Object: butterfly
          Commands:
            - remove: true

Objects:
  - Name: wall
    MapCharacter: 'w'
    Observers:
      Sprite2D:
        - TilingMode: WALL_16
          Image:
            - oryx/oryx_fantasy/wall9-0.png
            - oryx/oryx_fantasy/wall9-1.png
            - oryx/oryx_fantasy/wall9-2.png
            - oryx/oryx_fantasy/wall9-3.png
            - oryx/oryx_fantasy/wall9-4.png
            - oryx/oryx_fantasy/wall9-5.png
            - oryx/oryx_fantasy/wall9-6.png
            - oryx/oryx_fantasy/wall9-7.png
            - oryx/oryx_fantasy/wall9-8.png
            - oryx/oryx_fantasy/wall9-9.png
```

(continues on next page)

(continued from previous page)

```
            - oryx/oryx_fantasy/wall9-10.png
            - oryx/oryx_fantasy/wall9-11.png
            - oryx/oryx_fantasy/wall9-12.png
            - oryx/oryx_fantasy/wall9-13.png
            - oryx/oryx_fantasy/wall9-14.png
            - oryx/oryx_fantasy/wall9-15.png
      Block2D:
        - Shape: square
          Color: [0.7, 0.7, 0.7]
          Scale: 0.9

  - Name: butterfly
    InitialActions:
      - Action: butterfly_random_movement
        Delay: 3
        Randomize: true
    MapCharacter: '1'
    Observers:
      Sprite2D:
        - Image: gvgai/newset/butterfly1.png
      Block2D:
        - Shape: triangle
          Color: [0.0, 0.0, 1.0]
          Scale: 0.3

  - Name: cocoon
    MapCharacter: '0'
    InitialActions:
      - Action: spawn_butterfly
        Delay: 200
        Randomize: true
    Observers:
      Sprite2D:
        - Image: gvgai/newset/cocoonb1.png
      Block2D:
        - Shape: triangle
          Color: [0.0, 1.0, 0.0]
          Scale: 0.5

  - Name: spider
    InitialActions:
      - Action: spider_random_movement
        Delay: 3
        Randomize: true
    MapCharacter: 'S'
    Observers:
      Sprite2D:
        - Image: oryx/oryx_fantasy/avatars/spider1.png
      Block2D:
        - Shape: triangle
          Color: [1.0, 0.0, 0.0]
          Scale: 0.5
```

(continues on next page)

(continued from previous page)

```yaml
- Name: catcher
  MapCharacter: 'A'
  Observers:
    Sprite2D:
      - Image: gvgai/newset/girl5.png
    Block2D:
      - Shape: triangle
        Color: [1.0, 1.0, 1.0]
        Scale: 0.8
```

13.17 Cook Me Pasta

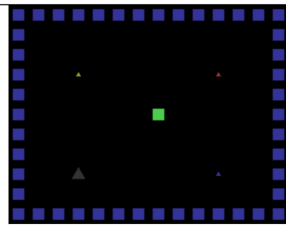
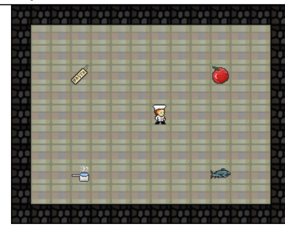
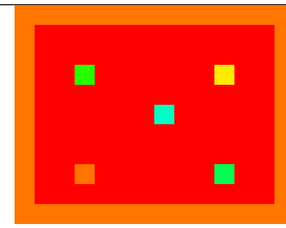
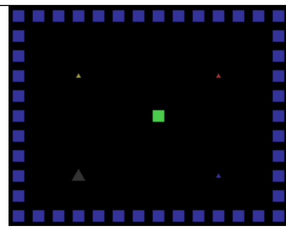
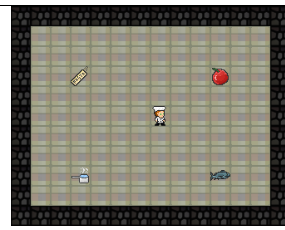
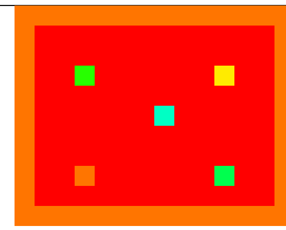
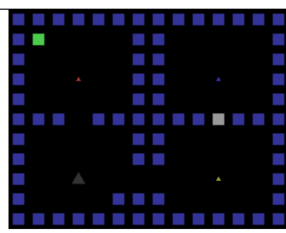
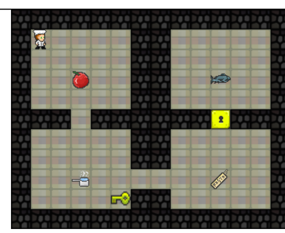
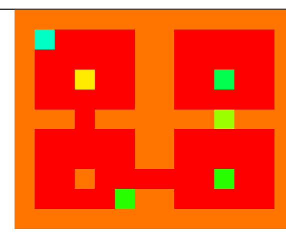
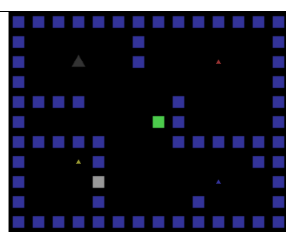
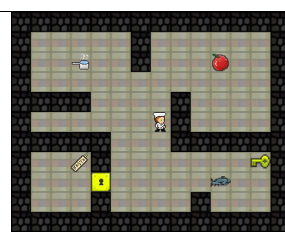
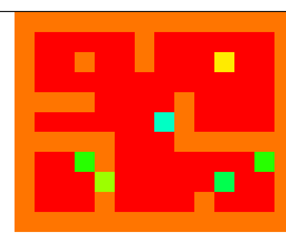
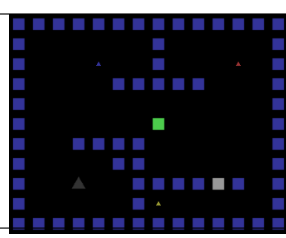
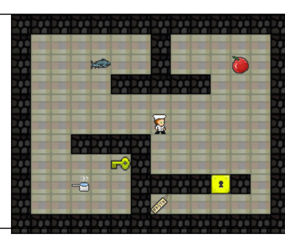
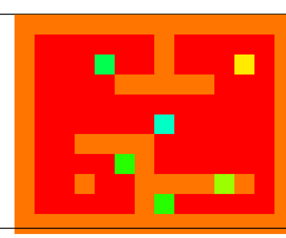
```
Single-Player/GVGAI/cookmepasta.yaml
```

13.17.1 Description

Help the chef create the meal, but make sure the ingredients are put together in the right order.

13.17.2 Levels

Table 33: Levels

	Block2D	Sprite2D	Vector
Level ID: 0 Size: 14x11			
Level ID: 1 Size: 14x11			
Level ID: 2 Size: 14x11			
Level ID: 3 Size: 14x11			
Level ID: 4 Size: 14x11			

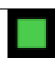
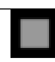
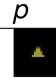
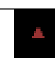
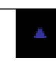
13.17.3 Code Example

The most basic way to create a Griddly Gym Environment. Defaults to level 0 and SPRITE_2D rendering.

```
import gym
import griddly

if __name__ == '__main__':

    env = gym.make('GDY-Cook-Me-Pasta-v0')
    env.reset()

    # Replace with your own control algorithm!
    for s in range(1000):
        obs, reward, done, info = env.step(env.action_space.sample())
        env.render() # Renders the environment from the perspective of a single player

        env.render(observer='global') # Renders the entire environment

        if done:
            env.reset()
```

13.17.4 Objects

Table 34: Tiles

Name ->	avatar	wall	key	lock	boiling_water	raw_pasta	tomato	tuna
Map Char ->	*A*	*w*	*k*	*l*	*b*	*p*	*o*	*t*
Block2D								
Sprite2D								
Vector								

13.17.5 Actions

move

Action Id	Mapping
1	Left
2	Up
3	Right
4	Down

13.17.6 YAML

```yaml
Version: "0.1"
Environment:
  Name: Cook Me Pasta
  Description: Help the chef create the meal, but make sure the ingredients are put together in the right order.
  Observers:
    Sprite2D:
      TileSize: 24
      BackgroundTile: gvgai/newset/floor6.png
  Player:
    AvatarObject: avatar # The player can only control a single avatar in the game
  Termination:
    Win:
      - eq: [complete_meal:count, 1]
    Lose:
      - eq: [wrong_place:count, 1]
  Levels:
    - |
      wwwwwwwwwwwwww
      w............w
      w............w
      w..p......o..w
      w............w
      w......A.....w
      w............w
      w............w
      w..b......t..w
      w............w
      wwwwwwwwwwwwww
    - |
      wwwwwwwwwwwwww
      w............w
      w............w
      w..p......o..w
      w............w
      w......A.....w
      w............w
      w............w
      w..b......t..w
      w............w
      wwwwwwwwwwwwww
    - |
      wwwwwwwwwwwwww
      wA....ww.....w
      w.....ww.....w
      w..o..ww..t..w
      w.....ww.....w
      www.wwwwwwlwww
      w.....ww.....w
      w.....ww.....w
      w..b......p..w
```

(continues on next page)

(continued from previous page)

```
      w....kww.....w
      wwwwwwwwwwwwww
    - |
      wwwwwwwwwwwwww
      w.....w......w
      w..b..w...o..w
      w............w
      wwww....w....w
      w......Aw....w
      wwwww...wwwwww
      w..pw.......kw
      w...l.....t..w
      w...w....w...w
      wwwwwwwwwwwwww
    - |
      wwwwwwwwwwwwww
      w......w.....w
      w...t..w...o.w
      w....wwwww...w
      w............w
      w......A.....w
      w..wwww......w
      w....kw......w
      w..b..wwwwlw.w
      w.....wp.....w
      wwwwwwwwwwwwww
    - |
      wwwwwwwwwwwwww
      w..lA........w
      w..wwwwww.woww
      w..t.........w
      w..wwwwwwww..w
      w..w....k.w..w
      w..w..p...w..w
      w..w...wwww..w
      w..w...b.....w
      w..w.....w...w
      wwwwwwwwwwwwww

Actions:
  # Define the move action
  - Name: move
    Behaviours:
      # The agent can move around freely in empty space and over holes
      - Src:
          Object: avatar
          Commands:
            - mov: _dest
        Dst:
          Object: [boiling_water, raw_pasta, tomato, tuna, cooked_pasta, pasta_sauce]
          Commands:
            - cascade: _dest
```

(continues on next page)

(continued from previous page)

```
    - Src:
        Object: [avatar, boiling_water, raw_pasta, tomato, tuna, cooked_pasta, pasta_
↪sauce]
        Commands:
          - mov: _dest
      Dst:
        Object: _empty

    # Behaviour for boiling_water
    - Src:
        Object: boiling_water
        Commands:
          - remove: true
          - reward: 4
      Dst:
        Object: raw_pasta
        Commands:
          - change_to: cooked_pasta

    # Behaviour for raw_pasta
    - Src:
        Object: raw_pasta
        Commands:
          - remove: true
          - reward: 4
      Dst:
        Object: boiling_water
        Commands:
          - change_to: cooked_pasta

    # Behaviours for tomato
    - Src:
        Object: tomato
        Commands:
          - remove: true
          - reward: 4
      Dst:
        Object: tuna
        Commands:
          - change_to: pasta_sauce

    # Behaviours for tuna
    - Src:
        Object: tuna
        Commands:
          - remove: true
          - reward: 4
      Dst:
        Object: tomato
        Commands:
          - change_to: pasta_sauce
```

(continues on next page)

(continued from previous page)

```
      # Behaviours for cooked_pasta
      - Src:
          Object: cooked_pasta
          Commands:
            - remove: true
            - reward: 17
        Dst:
          Object: pasta_sauce
          Commands:
            - change_to: complete_meal
      # Behaviours for pasta_sauce
      - Src:
          Object: pasta_sauce
          Commands:
            - remove: true
            - reward: 17
        Dst:
          Object: cooked_pasta
          Commands:
            - change_to: complete_meal

      # If the wrong things are mixed together
      - Src:
          Object: [raw_pasta, boiling_water]
          Commands:
            - remove: true
            - reward: -1
        Dst:
          Object: [tuna, tomato, pasta_sauce]
          Commands:
            - change_to: wrong_place

      - Src:
          Object: [tuna, tomato, pasta_sauce]
          Commands:
            - remove: true
            - reward: -1
        Dst:
          Object: [boiling_water, raw_pasta]
          Commands:
            - change_to: wrong_place


      # Keys and Locks
      - Src:
          Preconditions:
            - eq: [has_key, 1]
          Object: avatar
          Commands:
            - mov: _dest
        Dst:
          Object: lock
```

(continues on next page)

(continued from previous page)

```
          Commands:
            - remove: true

      # Avatar picks up the key
      - Src:
          Object: avatar
          Commands:
            - mov: _dest
            - incr: has_key
        Dst:
          Object: key
          Commands:
            - remove: true

Objects:

  - Name: avatar
    MapCharacter: A
    Variables:
      - Name: has_key
    Observers:
      Sprite2D:
        - Image: gvgai/newset/chef.png
      Block2D:
        - Shape: square
          Color: [0.3, 0.8, 0.3]
          Scale: 0.8

  - Name: wall
    MapCharacter: w
    Observers:
      Sprite2D:
        - Image: gvgai/newset/floor4.png
      Block2D:
        - Shape: square
          Color: [0.2, 0.2, 0.6]
          Scale: 0.8

  - Name: key
    MapCharacter: k
    Observers:
      Sprite2D:
        - Image: gvgai/oryx/key2.png
      Block2D:
        - Shape: square
          Color: [0.2, 0.2, 0.6]
          Scale: 0.8

  - Name: lock
    MapCharacter: l
    Observers:
      Sprite2D:
```

(continues on next page)

(continued from previous page)

```
        - Image: gvgai/newset/lock1.png
      Block2D:
        - Shape: square
          Color: [0.6, 0.6, 0.6]
          Scale: 0.8

  - Name: boiling_water
    MapCharacter: b
    Observers:
      Sprite2D:
        - Image: gvgai/newset/boilingwater.png
      Block2D:
        - Shape: triangle
          Color: [0.2, 0.2, 0.2]
          Scale: 0.8
  - Name: raw_pasta
    MapCharacter: p
    Observers:
      Sprite2D:
        - Image: gvgai/newset/pasta.png
      Block2D:
        - Shape: triangle
          Color: [0.6, 0.6, 0.2]
          Scale: 0.3
  - Name: tomato
    MapCharacter: o
    Observers:
      Sprite2D:
        - Image: gvgai/newset/tomato.png
      Block2D:
        - Shape: triangle
          Color: [0.6, 0.2, 0.2]
          Scale: 0.3
  - Name: tuna
    MapCharacter: t
    Observers:
      Sprite2D:
        - Image: gvgai/newset/tuna.png
      Block2D:
        - Shape: triangle
          Color: [0.2, 0.2, 0.6]
          Scale: 0.3

  - Name: cooked_pasta
    Observers:
      Sprite2D:
        - Image: gvgai/newset/pastaplate.png
      Block2D:
        - Shape: triangle
          Color: [0.6, 0.6, 0.6]
          Scale: 0.7
  - Name: pasta_sauce
```

(continues on next page)

(continued from previous page)

```
    Observers:
      Sprite2D:
        - Image: gvgai/newset/tomatosauce.png
      Block2D:
        - Shape: triangle
          Color: [0.6, 0.0, 0.2]
          Scale: 0.7

  - Name: complete_meal
    Observers:
      Sprite2D:
        - Image: gvgai/newset/pastasauce.png
      Block2D:
        - Shape: triangle
          Color: [0.6, 0.0, 0.2]
          Scale: 0.7

  - Name: wrong_place
    Observers:
      Sprite2D:
        - Image: gvgai/oryx/slash1.png
      Block2D:
        - Shape: square
          Color: [1.0, 0.0, 0.0]
          Scale: 1.0
```

13.18 Partially Observable Bait

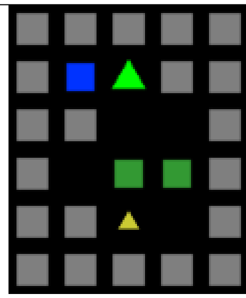
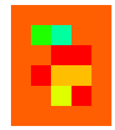
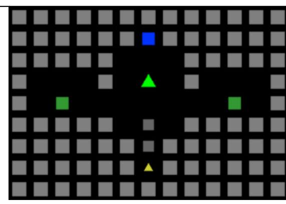
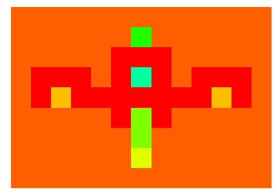
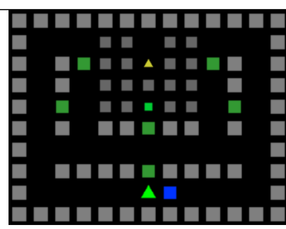
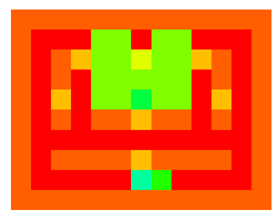
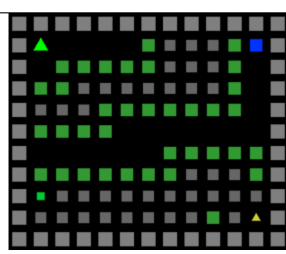
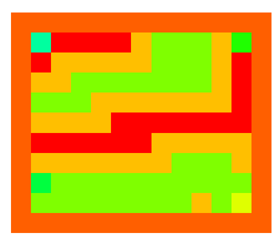
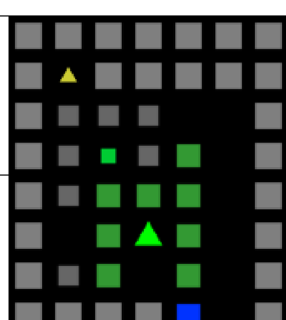
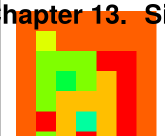
```
Single-Player/GVGAI/bait_partially_observable.yaml
```

13.18.1 Description

Get the key and unlock the door. Fill in the holes in the floor with blocks to get to the key.

13.18.2 Levels

Table 35: Levels

	Block2D	Sprite2D	Vector
<table><tr><td>Level ID</td><td>0</td></tr><tr><td>Size</td><td>5x6</td></tr></table>			
<table><tr><td>Level ID</td><td>1</td></tr><tr><td>Size</td><td>13x9</td></tr></table>			
<table><tr><td>Level ID</td><td>2</td></tr><tr><td>Size</td><td>13x10</td></tr></table>			
<table><tr><td>Level ID</td><td>3</td></tr><tr><td>Size</td><td>13x11</td></tr></table>			
			

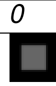
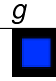
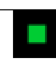
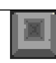
13.18.3 Code Example

The most basic way to create a Griddly Gym Environment. Defaults to level 0 and SPRITE_2D rendering.

```python
import gym
import griddly

if __name__ == '__main__':

    env = gym.make('GDY-Partially-Observable-Bait-v0')
    env.reset()

    # Replace with your own control algorithm!
    for s in range(1000):
        obs, reward, done, info = env.step(env.action_space.sample())
        env.render() # Renders the environment from the perspective of a single player

        env.render(observer='global') # Renders the entire environment

        if done:
            env.reset()
```

13.18.4 Objects

Table 36: Tiles

Name ->	avatar	hole	box	key	goal	mushroom	wall
Map Char ->	*A*	*0*	*1*	*k*	*g*	*m*	*w*
Block2D							
Sprite2D							
Vector							

13.18.5 Actions

move

Relative The actions are calculated relative to the object being controlled.

Action Id	Mapping
1	Rotate left
2	Move forwards
3	Rotate right

13.18.6 YAML

```
# http://www.mobygames.com/game/bait

Version: "0.1"
Environment:
  Name: Partially Observable Bait
  Description: Get the key and unlock the door. Fill in the holes in the floor with␣
↪blocks to get to the key.
  Observers:
    Sprite2D:
      TileSize: 24
      BackgroundTile: gvgai/oryx/backLBrown.png
  Player:
    Observer:
      RotateWithAvatar: true
      TrackAvatar: true
      Height: 5
      Width: 5
      OffsetX: 0
      OffsetY: 2
    AvatarObject: avatar
  Termination:
    Win:
      - eq: [goal:count, 0]
    Lose:
      - eq: [avatar:count, 0]
  Levels:
    - |
      w w w w w
      w g A w w
      w w . . w
      w . 1 1 w
      w w k . w
      w w w w w
    - |
      w w w w w w w w w w w w w
      w w w w w w g w w w w w w
      w w w w w . . . w w w w w
      w . . . w . A . w . . . w
      w . 1 . . . . . . . 1 . w
      w w w w w . 0 . w w w w w
      w w w w w w 0 w w w w w w
      w w w w w w k w w w w w w
      w w w w w w w w w w w w w
    - |
      w w w w w w w w w w w w w
      w . . . 0 0 . 0 0 . . . w
      w . w 1 0 0 k 0 0 1 w . w
      w . w . 0 0 0 0 0 . w . w
      w . 1 . 0 0 m 0 0 . 1 . w
      w . w . w w 1 w w . w . w
      w . . . . . . . . . . . w
```

(continues on next page)

(continued from previous page)

```
      w . w w w w 1 w w w w . w
      w . . . . . . A g . . . . w
      w w w w w w w w w w w w w
    - |
      w w w w w w w w w w w w w
      w A . . . . 1 0 0 0 1 g w
      w . 1 1 1 1 1 0 0 0 1 . w
      w 1 1 0 0 0 0 0 0 0 1 . w
      w 0 0 0 1 1 1 1 1 1 1 . w
      w 1 1 1 1 . . . . . . . w
      w . . . . . . . 1 1 1 1 1 w
      w 1 1 1 1 1 1 1 0 0 0 1 w
      w m 0 0 0 0 0 0 0 0 0 0 w
      w 0 0 0 0 0 0 0 0 1 0 k w
      w w w w w w w w w w w w w
    - |
      w w w w w w w
      w k w w w w w
      w 0 0 0 . . w
      w 0 m 0 1 . w
      w 0 1 1 1 . w
      w . 1 A 1 . w
      w 0 1 . 1 . w
      w w w w g . w
      w w w w w w w

Actions:
  # Define the move action
  - Name: move
    InputMapping:
      Inputs:
        1:
          Description: Rotate left
          OrientationVector: [-1, 0]
        2:
          Description: Move forwards
          OrientationVector: [0, -1]
          VectorToDest: [0, -1]
        3:
          Description: Rotate right
          OrientationVector: [1, 0]
      Relative: true
    Behaviours:

      # Avatar rotates
      - Src:
          Object: avatar
          Commands:
            - rot: _dir
        Dst:
          Object: avatar
```

(continues on next page)

(continued from previous page)

```
    # Avatar and boxes can move into empty space
    - Src:
        Object: [avatar, box]
        Commands:
          - mov: _dest
      Dst:
        Object: _empty

    # Boxes can be pushed by the avatar
    - Src:
        Object: avatar
        Commands:
          - mov: _dest
      Dst:
        Object: box
        Commands:
          - cascade: _dest

    # If a box falls into a hole, both disappear
    - Src:
        Object: box
        Commands:
          - remove: true
          - reward: 1
      Dst:
        Object: hole
        Commands:
          - remove: true

    # If the avatar falls into a hole remove the avatar
    - Src:
        Object: avatar
        Commands:
          - remove: true
          - reward: -1
      Dst:
        Object: hole

    # If the avatar picks up a mushroom, remove the mushroom
    - Src:
        Object: avatar
        Commands:
          - reward: 1
          - mov: _dest
      Dst:
        Object: mushroom
        Commands:
          - remove: true

    # Only an avatar with a key can
    - Src:
        Preconditions:
```

(continues on next page)

(continued from previous page)

```
            - eq: [has_key, 1]
          Object: avatar
          Commands:
            - reward: 5
        Dst:
          Object: goal
          Commands:
            - remove: true

      # Avatar picks up the key
      - Src:
          Object: avatar
          Commands:
            - mov: _dest
            - incr: has_key
        Dst:
          Object: key
          Commands:
            - remove: true

Objects:
  - Name: avatar
    MapCharacter: A
    Variables:
      - Name: has_key
    Observers:
      Sprite2D:
        - Image: gvgai/oryx/swordman1_0.png
      Block2D:
        - Shape: triangle
          Color: [0.0, 1.0, 0.0]
          Scale: 0.8

  - Name: hole
    MapCharacter: "0"
    Observers:
      Sprite2D:
        - Image: gvgai/newset/hole1.png
      Block2D:
        - Shape: square
          Color: [0.4, 0.4, 0.4]
          Scale: 0.7

  - Name: box
    MapCharacter: "1"
    Observers:
      Sprite2D:
        - Image: gvgai/newset/block3.png
      Block2D:
        - Shape: square
          Color: [0.2, 0.6, 0.2]
          Scale: 0.8
```

(continues on next page)

(continued from previous page)

```
- Name: key
  MapCharacter: k
  Observers:
    Sprite2D:
      - Image: gvgai/oryx/key2.png
    Block2D:
      - Shape: triangle
        Color: [0.8, 0.8, 0.2]
        Scale: 0.5

- Name: goal
  MapCharacter: g
  Observers:
    Sprite2D:
      - Image: gvgai/oryx/doorclosed1.png
    Block2D:
      - Shape: square
        Color: [0.0, 0.2, 1.0]
        Scale: 0.8

- Name: mushroom
  MapCharacter: m
  Observers:
    Sprite2D:
      - Image: gvgai/oryx/mushroom2.png
    Block2D:
      - Shape: square
        Color: [0.0, 0.8, 0.2]
        Scale: 0.5

- Name: wall
  MapCharacter: w
  Observers:
    Sprite2D:
      - TilingMode: WALL_16
        Image:
          - gvgai/oryx/dirtWall_0.png
          - gvgai/oryx/dirtWall_1.png
          - gvgai/oryx/dirtWall_2.png
          - gvgai/oryx/dirtWall_3.png
          - gvgai/oryx/dirtWall_4.png
          - gvgai/oryx/dirtWall_5.png
          - gvgai/oryx/dirtWall_6.png
          - gvgai/oryx/dirtWall_7.png
          - gvgai/oryx/dirtWall_8.png
          - gvgai/oryx/dirtWall_9.png
          - gvgai/oryx/dirtWall_10.png
          - gvgai/oryx/dirtWall_11.png
          - gvgai/oryx/dirtWall_12.png
          - gvgai/oryx/dirtWall_13.png
          - gvgai/oryx/dirtWall_14.png
```

(continues on next page)

(continued from previous page)

```
          - gvgai/oryx/dirtWall_15.png
    Block2D:
      - Shape: square
        Color: [0.5, 0.5, 0.5]
        Scale: 0.9
```

13.19 Zelda Sequential

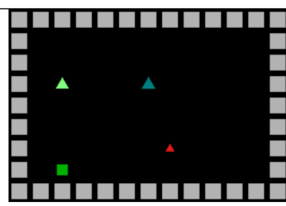
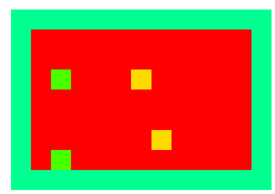
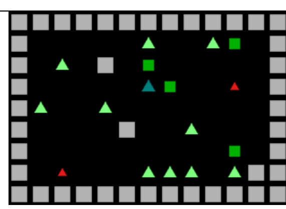
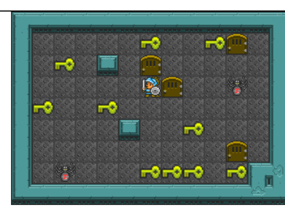
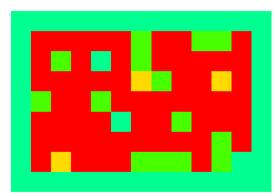
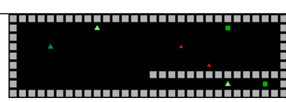
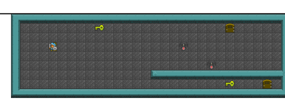
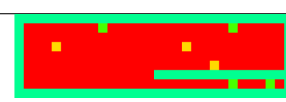
```
Single-Player/GVGAI/zelda_sequential.yaml
```

13.19.1 Description

A port of the GVGAI game "Zelda". Pick up keys to reach doors in the correct order and avoid enemies. For example, previously you could go – key -> door -> door. But now you would need to go – key -> door –> key –> door.

13.19.2 Levels

Table 37: Levels

	Block2D	Sprite2D	Vector
Level ID: 0 Size: 13x9			
Level ID: 1 Size: 13x9			
Level ID: 2 Size: 30x9			

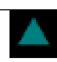
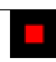
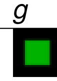
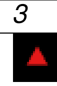
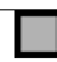
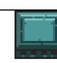
13.19.3 Code Example

The most basic way to create a Griddly Gym Environment. Defaults to level 0 and SPRITE_2D rendering.

```python
import gym
import griddly

if __name__ == '__main__':

    env = gym.make('GDY-Zelda-Sequential-v0')
    env.reset()

    # Replace with your own control algorithm!
    for s in range(1000):
        obs, reward, done, info = env.step(env.action_space.sample())
        env.render() # Renders the environment from the perspective of a single player

        env.render(observer='global') # Renders the entire environment

        if done:
            env.reset()
```

13.19.4 Objects

Table 38: Tiles

Name ->	avatar	attack_fire	key	goal	spider	wall
Map Char ->	*A*	*x*	*+*	*g*	*3*	*w*
Block2D						
Sprite2D						
Vector						

13.19.5 Actions

attack

Relative The actions are calculated relative to the object being controlled.

Action Id	Mapping
1	attack front

move

Relative The actions are calculated relative to the object being controlled.

Action Id	Mapping
1	Rotate left
2	Move forwards
3	Rotate right
4	Move Backwards

random_movement

Internal This action can only be called from other actions, not by the player.

Action Id	Mapping
1	Left
2	Up
3	Right
4	Down

remove_sprite

Internal This action can only be called from other actions, not by the player.

Action Id	Mapping
1	Left
2	Up
3	Right
4	Down

13.19.6 YAML

```yaml
Version: "0.1"
Environment:
  Name: Zelda Sequential
  Description: A port of the GVGAI game "Zelda". Pick up keys to reach doors in the correct order and avoid enemies. For example, previously you could go -- key -> door -> door. But now you would need to go -- key -> door --> key --> door.
  Observers:
    Sprite2D:
      TileSize: 24
      BackgroundTile: gvgai/newset/floor2.png
  Player:
    AvatarObject: avatar
  Termination:
    Win:
      - eq: [goal:count, 0]
    Lose:
      - eq: [avatar:count, 0]
```

(continues on next page)

(continued from previous page)

```yaml
  Levels:
    - |
      wwwwwwwwwwwww
      w...........w
      w...........w
      w.+...A.....w
      w...........w
      w...........w
      w......3....w
      w.g.........w
      wwwwwwwwwwwww
    - |
      wwwwwwwwwwwww
      w.....+..+g.w
      w.+.w.g.....w
      w.....Ag..3.w
      w+..+.......w
      w....w..+...w
      w.........g.w
      w.3...+++.+ww
      wwwwwwwwwwwww
    - |
      wwwwwwwwwwwwwwwwwwwwwwwwwwwwww
      w........+.............g.....w
      w............................w
      w...A.............3..........w
      w............................w
      w....................3.......w
      w..............wwwwwwwwwwwwwww
      w......................+...g.w
      wwwwwwwwwwwwwwwwwwwwwwwwwwwwww
Actions:
  # Define action that cannot be controlled by the player.
  # (In this case the spider movement)
  - Name: random_movement
    InputMapping:
      Internal: true
    Behaviours:
      # The gnome and the spider can move into empty space
      - Src:
          Object: spider
          Commands:
            - mov: _dest
            - exec:
                Action: random_movement
                Delay: 5
                Randomize: true
        Dst:
          Object: _empty
      # The spider will not move into the wall, but it needs to keep moving
      - Src:
          Object: spider
```

(continues on next page)

(continued from previous page)

```
          Commands:
            - exec:
                Action: random_movement
                Delay: 5
                Randomize: true
        Dst:
          Object: [wall, key, goal, spider]
      # If the gnome moves into a spider
      - Src:
          Object: spider
        Dst:
          Object: avatar
          Commands:
            - remove: true
            - reward: -1
  # remove the recently spawned attack animation sprite
  - Name: remove_sprite
    InputMapping:
      Internal: true
    Behaviours:
      - Src:
          Object: attack_fire
          Commands:
            - remove: true
        Dst:
          Object: attack_fire
  # Define the move action
  - Name: move
    InputMapping:
      Inputs:
        1:
          Description: Rotate left
          OrientationVector: [-1, 0]
        2:
          Description: Move forwards
          OrientationVector: [0, -1]
          VectorToDest: [0, -1]
        3:
          Description: Rotate right
          OrientationVector: [1, 0]
        4:
          Description: Move Backwards
          VectorToDest: [0, 1]
          OrientationVector: [0, -1]
      Relative: true
    Behaviours:
      # Tell the gnome to rotate if it performs an action on itself (Rotate left and␣
↪Rotate right actions)
      - Src:
          Object: avatar
          Commands:
            - rot: _dir
```

(continues on next page)

(continued from previous page)

```
    Dst:
     Object: avatar
   # Only an avatar with a key can win
   - Src:
     Preconditions:
      - eq: [src.has_key, 1]
     Object: avatar
     Commands:
      - reward: 1
      - decr: has_key
      - mov: _dest
      - set_tile: 0
    Dst:
     Object: goal
     Commands:
      - remove: true
   # If the gnome moves into a gem object, the stick is removed, triggering a win condition
   - Src:
     Object: avatar
     Commands:
      - mov: _dest
      - eq:
        Arguments: [ src.has_key, 0 ]
        Commands:
         - incr: has_key
         - reward: 1
         - set_tile: 1
    Dst:
     Object: key
     Commands:
      - eq:
        Arguments: [ src.has_key, 0 ]
        Commands:
         - remove: true
   # If the gnome moves into a spider
   - Src:
     Object: avatar
     Commands:
      - remove: true
      - reward: -1
    Dst:
     Object: spider
   # The gnome and the spider can move into empty space
   - Src:
     Object: avatar
     Commands:
      - mov: _dest
    Dst:
     Object: _empty
 - Name: attack
  InputMapping:
```

(continues on next page)

(continued from previous page)

```
      Inputs:
        1:
          Description: attack front
          OrientationVector: [ -1, 0 ]
          VectorToDest: [-1, 0]
      Relative: true
    Behaviours:
      - Src:
          Object: avatar
          Commands:
            - spawn: attack_fire
        Dst:
          Object: spider
          Commands:
            - remove: true
      - Src:
          Object: avatar
          Commands:
            - spawn: attack_fire
        Dst:
          Object: _empty
Objects:
  - Name: avatar
    Z: 3
    MapCharacter: A
    Variables:
      - Name: has_key
    Observers:
      Sprite2D:
        - Image: gvgai/oryx/swordman1_0.png
        - Image: gvgai/oryx/swordmankey1_0.png
      Block2D:
        - Shape: triangle
          Color: [0.0, 0.5, 0.5]
          Scale: 0.75
        - Shape: triangle
          Color: [0.3, 0.5, 0.2]
          Scale: 1.0
  - Name: attack_fire
    Z: 1
    InitialActions:
      - Action: remove_sprite
        Delay: 3
    MapCharacter: x
    Observers:
      Sprite2D:
        - Image: gvgai/oryx/fire1.png
      Block2D:
        - Shape: square
          Color: [1.0, 0.0, 0.0]
          Scale: 0.5
  - Name: key
```

(continues on next page)

(continued from previous page)

```yaml
    Z: 2
    MapCharacter: "+"
    Observers:
      Sprite2D:
        - Image: gvgai/oryx/key2.png
      Block2D:
        - Shape: triangle
          Color: [0.5, 1.0, 0.5]
          Scale: 0.7
  - Name: goal
    Z: 2
    MapCharacter: g
    Observers:
      Sprite2D:
        - Image: gvgai/oryx/doorclosed1.png
      Block2D:
        - Shape: square
          Color: [0.0, 0.7, 0.0]
          Scale: 0.7
#   - Name: chaser
#     Z: 2
#     MapCharacter: "3"
#     Observers:
#       Sprite2D:
#         - Image: gvgai/oryx/skeleton1.png
  - Name: spider
    Z: 2
    InitialActions:
      - Action: random_movement
        Delay: 5
    MapCharacter: "3"
    Observers:
      Sprite2D:
        - Image: oryx/oryx_fantasy/avatars/spider1.png
      Block2D:
        - Shape: triangle
          Color: [0.9, 0.1, 0.1]
          Scale: 0.5
  - Name: wall
    MapCharacter: w
    Observers:
      Sprite2D:
        - TilingMode: WALL_16
          Image:
            - gvgai/oryx/wall3_0.png
            - gvgai/oryx/wall3_1.png
            - gvgai/oryx/wall3_2.png
            - gvgai/oryx/wall3_3.png
            - gvgai/oryx/wall3_4.png
            - gvgai/oryx/wall3_5.png
            - gvgai/oryx/wall3_6.png
            - gvgai/oryx/wall3_7.png
```

(continues on next page)

(continued from previous page)

```
          - gvgai/oryx/wall3_8.png
          - gvgai/oryx/wall3_9.png
          - gvgai/oryx/wall3_10.png
          - gvgai/oryx/wall3_11.png
          - gvgai/oryx/wall3_12.png
          - gvgai/oryx/wall3_13.png
          - gvgai/oryx/wall3_14.png
          - gvgai/oryx/wall3_15.png
    Block2D:
      - Shape: square
        Color: [0.7, 0.7, 0.7]
        Scale: 1.0
```

13.20 Partially Observable Zelda

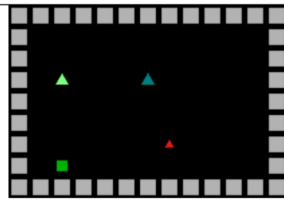
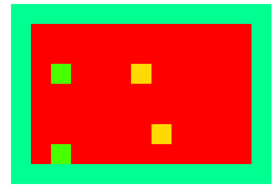
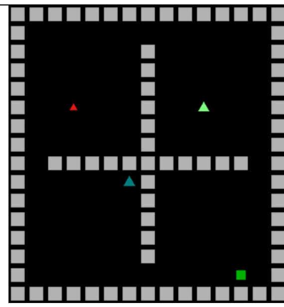
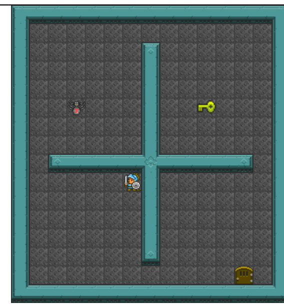
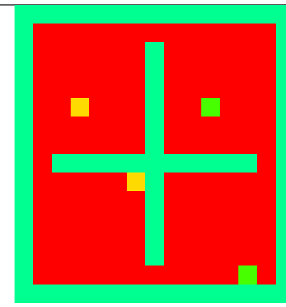
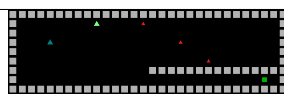
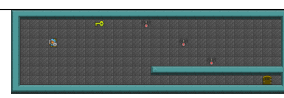
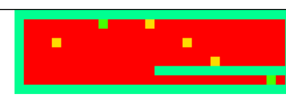
```
Single-Player/GVGAI/zelda_partially_observable.yaml
```

13.20.1 Description

A port of the GVGAI game “Zelda”, but partially observable. Pick up keys to reach doors and avoid enemies.

13.20.2 Levels

Table 39: Levels

<table>
<tr><th></th><th>Block2D</th><th>Sprite2D</th><th>Vector</th></tr>
<tr><td><table><tr><td>Level ID</td><td>0</td></tr><tr><td>Size</td><td>13x9</td></tr></table></td><td></td><td></td><td></td></tr>
<tr><td><table><tr><td>Level ID</td><td>1</td></tr><tr><td>Size</td><td>15x16</td></tr></table></td><td></td><td></td><td></td></tr>
<tr><td><table><tr><td>Level ID</td><td>2</td></tr><tr><td>Size</td><td>30x9</td></tr></table></td><td></td><td></td><td></td></tr>
</table>

13.20.3 Code Example

The most basic way to create a Griddly Gym Environment. Defaults to level 0 and SPRITE_2D rendering.

```python
import gym
import griddly

if __name__ == '__main__':

    env = gym.make('GDY-Partially-Observable-Zelda-v0')
    env.reset()

    # Replace with your own control algorithm!
    for s in range(1000):
        obs, reward, done, info = env.step(env.action_space.sample())
        env.render() # Renders the environment from the perspective of a single player
```

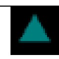
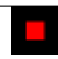
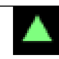
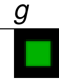
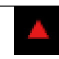
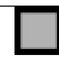
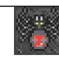
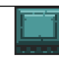
(continues on next page)

(continued from previous page)

```
        env.render(observer='global') # Renders the entire environment

        if done:
            env.reset()
```

13.20.4 Objects

Table 40: Tiles

Name ->	avatar	attack_fire	key	goal	spider	wall
Map Char ->	*A*	*x*	*+*	*g*	*3*	*w*
Block2D						
Sprite2D						
Vector						

13.20.5 Actions

attack

Relative The actions are calculated relative to the object being controlled.

Action Id	Mapping
1	attack front

move

Relative The actions are calculated relative to the object being controlled.

Action Id	Mapping
1	Rotate left
2	Move forwards
3	Rotate right
4	Move Backwards

random_movement

Internal This action can only be called from other actions, not by the player.

Action Id	Mapping
1	Left
2	Up
3	Right
4	Down

remove_sprite

Internal This action can only be called from other actions, not by the player.

Action Id	Mapping
1	Left
2	Up
3	Right
4	Down

13.20.6 YAML

```
Version: "0.1"
Environment:
  Name: Partially Observable Zelda
  Description: A port of the GVGAI game "Zelda", but partially observable. Pick up keys to reach doors and avoid enemies.
  Observers:
    Sprite2D:
      TileSize: 24
      BackgroundTile: gvgai/newset/floor2.png
  Player:
    Observer:
      RotateWithAvatar: true
      TrackAvatar: true
      Height: 5
      Width: 5
      OffsetX: 0
      OffsetY: 2
    AvatarObject: avatar
  Termination:
    Win:
      - eq: [goal:count, 0]
    Lose:
      - eq: [avatar:count, 0]
  Levels:
    - |
      wwwwwwwwwwwww
      w...........w
      w...........w
      w.+...A.....w
      w...........w
      w...........w
      w......3....w
      w.g.........w
      wwwwwwwwwwwww
    - |
      wwwwwwwwwwwwwww
      w.............w
      w......w......w
      w......w......w
```

(continues on next page)

(continued from previous page)

```yaml
      w......w......w
      w..3...w..+...w
      w......w......w
      w......w......w
      w.wwwwwwwwwww.w
      w.....Aw......w
      w......w......w
      w......w......w
      w......w......w
      w......w......w
      w...........g.w
      wwwwwwwwwwwwwww
    - |
      wwwwwwwwwwwwwwwwwwwwwwwwwwwwww
      w........+....3..............w
      w............................w
      w...A.............3..........w
      w............................w
      w....................3.......w
      w..............wwwwwwwwwwwwwww
      w..........................g.w
      wwwwwwwwwwwwwwwwwwwwwwwwwwwwww
Actions:
  # Define action that cannot be controlled by the player.
  # (In this case the spider movement)
  - Name: random_movement
    InputMapping:
      Internal: true
    Behaviours:
      # The gnome and the spider can move into empty space
      - Src:
          Object: spider
          Commands:
            - mov: _dest
            - exec:
                Action: random_movement
                Delay: 5
                Randomize: true
        Dst:
          Object: _empty
      # The spider will not move into the wall, but it needs to keep moving
      - Src:
          Object: spider
          Commands:
            - exec:
                Action: random_movement
                Delay: 5
                Randomize: true
        Dst:
          Object: [wall, key, goal, spider]
      # If the gnome moves into a spider
      - Src:
```

(continues on next page)

(continued from previous page)

```
          Object: spider
        Dst:
          Object: avatar
          Commands:
            - remove: true
            - reward: -1
  # remove the recently spawned attack animation sprite
  - Name: remove_sprite
    InputMapping:
      Internal: true
    Behaviours:
      - Src:
          Object: attack_fire
          Commands:
            - remove: true
        Dst:
          Object: attack_fire
  # Define the move action
  - Name: move
    InputMapping:
      Inputs:
        1:
          Description: Rotate left
          OrientationVector: [-1, 0]
        2:
          Description: Move forwards
          OrientationVector: [0, -1]
          VectorToDest: [0, -1]
        3:
          Description: Rotate right
          OrientationVector: [1, 0]
        4:
          Description: Move Backwards
          VectorToDest: [0, 1]
          OrientationVector: [0, -1]
      Relative: true
    Behaviours:
      # Tell the gnome to rotate if it performs an action on itself (Rotate left and␣
↪Rotate right actions)
      - Src:
          Object: avatar
          Commands:
            - rot: _dir
        Dst:
          Object: avatar
      # Only an avatar with a key can win
      - Src:
          Preconditions:
            - eq: [src.has_key, 1]
          Object: avatar
          Commands:
            - reward: 1
```

(continues on next page)

(continued from previous page)

```
          # - decr: has_key
          - mov: _dest
          # - set_tile: 0
      Dst:
        Object: goal
        Commands:
          - remove: true
    # If the gnome moves into a gem object, the stick is removed, triggering a win␣
↪condition
    - Src:
        Object: avatar
        Commands:
          - mov: _dest
          - eq:
              Arguments: [ src.has_key, 0 ]
              Commands:
                - incr: has_key
                - reward: 1
                - set_tile: 1
      Dst:
        Object: key
        Commands:
          - eq:
              Arguments: [ src.has_key, 0 ]
              Commands:
                - remove: true
    # If the gnome moves into a spider
    - Src:
        Object: avatar
        Commands:
          - remove: true
          - reward: -1
      Dst:
        Object: spider
    # The gnome and the spider can move into empty space
    - Src:
        Object: avatar
        Commands:
          - mov: _dest
      Dst:
        Object: _empty
- Name: attack
  InputMapping:
    Inputs:
      1:
        Description: attack front
        OrientationVector: [ -1, 0 ]
        VectorToDest: [-1, 0]
    Relative: true
  Behaviours:
    - Src:
        Object: avatar
```

(continues on next page)

(continued from previous page)

```
          Commands:
            - spawn: attack_fire
        Dst:
          Object: spider
          Commands:
            - remove: true
      - Src:
          Object: avatar
          Commands:
            - spawn: attack_fire
        Dst:
          Object: _empty
Objects:
  - Name: avatar
    Z: 3
    MapCharacter: A
    Variables:
      - Name: has_key
    Observers:
      Sprite2D:
        - Image: gvgai/oryx/swordman1_0.png
        - Image: gvgai/oryx/swordmankey1_0.png
      Block2D:
        - Shape: triangle
          Color: [0.0, 0.5, 0.5]
          Scale: 0.75
        - Shape: triangle
          Color: [0.3, 0.5, 0.2]
          Scale: 1.0
  - Name: attack_fire
    Z: 1
    InitialActions:
      - Action: remove_sprite
        Delay: 3
    MapCharacter: x
    Observers:
      Sprite2D:
        - Image: gvgai/oryx/fire1.png
      Block2D:
        - Shape: square
          Color: [1.0, 0.0, 0.0]
          Scale: 0.5
  - Name: key
    Z: 2
    MapCharacter: "+"
    Observers:
      Sprite2D:
        - Image: gvgai/oryx/key2.png
      Block2D:
        - Shape: triangle
          Color: [0.5, 1.0, 0.5]
          Scale: 0.7
```

(continues on next page)

(continued from previous page)

```
- Name: goal
  Z: 2
  MapCharacter: g
  Observers:
    Sprite2D:
      - Image: gvgai/oryx/doorclosed1.png
    Block2D:
      - Shape: square
        Color: [0.0, 0.7, 0.0]
        Scale: 0.7
#   - Name: chaser
#     Z: 2
#     MapCharacter: "3"
#     Observers:
#       Sprite2D:
#         - Image: gvgai/oryx/skeleton1.png
- Name: spider
  Z: 2
  InitialActions:
    - Action: random_movement
      Delay: 5
  MapCharacter: "3"
  Observers:
    Sprite2D:
      - Image: oryx/oryx_fantasy/avatars/spider1.png
    Block2D:
      - Shape: triangle
        Color: [0.9, 0.1, 0.1]
        Scale: 0.5
- Name: wall
  MapCharacter: w
  Observers:
    Sprite2D:
      - TilingMode: WALL_16
        Image:
          - gvgai/oryx/wall3_0.png
          - gvgai/oryx/wall3_1.png
          - gvgai/oryx/wall3_2.png
          - gvgai/oryx/wall3_3.png
          - gvgai/oryx/wall3_4.png
          - gvgai/oryx/wall3_5.png
          - gvgai/oryx/wall3_6.png
          - gvgai/oryx/wall3_7.png
          - gvgai/oryx/wall3_8.png
          - gvgai/oryx/wall3_9.png
          - gvgai/oryx/wall3_10.png
          - gvgai/oryx/wall3_11.png
          - gvgai/oryx/wall3_12.png
          - gvgai/oryx/wall3_13.png
          - gvgai/oryx/wall3_14.png
          - gvgai/oryx/wall3_15.png
    Block2D:
```

(continues on next page)

(continued from previous page)

```
        - Shape: square
          Color: [0.7, 0.7, 0.7]
          Scale: 1.0
```

13.21 Sokoban

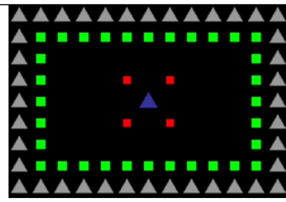
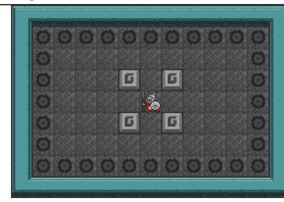
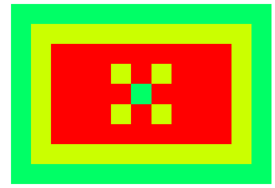
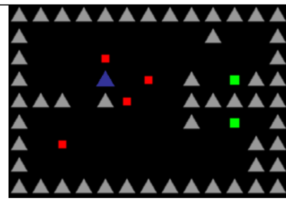
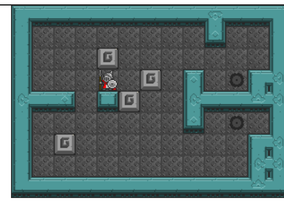
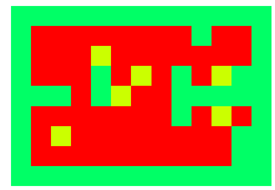
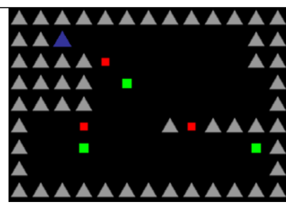
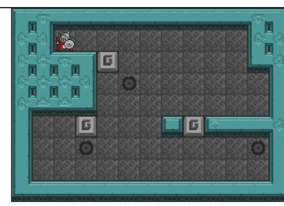
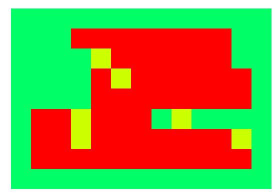
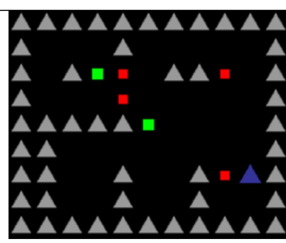
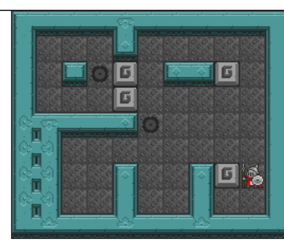
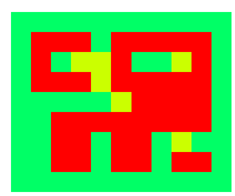
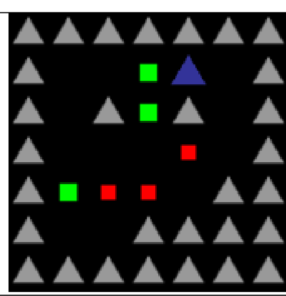
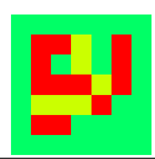
```
Single-Player/GVGAI/sokoban.yaml
```

13.21.1 Description

Push the boxes into holes.

13.21.2 Levels

Table 41: Levels

	Block2D	Sprite2D	Vector
Level ID 0 Size 13x9			
Level ID 1 Size 13x9			
Level ID 2 Size 13x9			
Level ID 3 Size 11x9			
Level ID 4 Size 7x7			
			

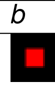
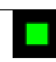
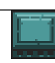
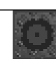
13.21.3 Code Example

The most basic way to create a Griddly Gym Environment. Defaults to level 0 and SPRITE_2D rendering.

```python
import gym
import griddly

if __name__ == '__main__':

    env = gym.make('GDY-Sokoban-v0')
    env.reset()

    # Replace with your own control algorithm!
    for s in range(1000):
        obs, reward, done, info = env.step(env.action_space.sample())
        env.render() # Renders the environment from the perspective of a single player

        env.render(observer='global') # Renders the entire environment

        if done:
            env.reset()
```

13.21.4 Objects

Table 42: Tiles

Name ->	box	wall	hole	avatar
Map Char ->	*b*	*w*	*h*	*A*
Block2D				
Sprite2D				
Vector				

13.21.5 Actions

move

Action Id	Mapping
1	Left
2	Up
3	Right
4	Down

13.21.6 YAML

```yaml
Version: "0.1"
Environment:
  Name: Sokoban
  Description: Push the boxes into holes.
  Observers:
    Sprite2D:
      TileSize: 24
      BackgroundTile: gvgai/newset/floor2.png
    Block2D:
      TileSize: 24
  Player:
    AvatarObject: avatar # The player can only control a single avatar in the game
  Termination:
    Win:
      - eq: [box:count, 0] # If there are no boxes left
  Levels:
    - |
      wwwwwwwwwwwww
      whhhhhhhhhhhw
      wh.........hw
      wh...b.b...hw
      wh....A....hw
      wh...b.b...hw
      wh.........hw
      whhhhhhhhhhhw
      wwwwwwwwwwwww
    - |
      wwwwwwwwwwwww
      w........w..w
      w...b.......w
      w...A.b.w.hww
      www.wb..wwwww
      w.......w.h.w
      w.b........ww
      w..........ww
      wwwwwwwwwwwww
    - |
      wwwwwwwwwwwww
      wwA........ww
      wwwwb......ww
      wwww.h......w
      wwww........w
      w..b...wbwwww
      w..h.......hw
      w...........w
      wwwwwwwwwwwww
    - |
      wwwwwwwwwww
      w...w.....w
      w.whb.wwb.w
      w...b.....w
```

(continues on next page)

(continued from previous page)

```
      wwwwwh....w
      ww........w
      ww..w..wbAw
      ww..w..w..w
      wwwwwwwwwww
    - |
      wwwwwww
      w..hA.w
      w.whw.w
      w...b.w
      whbb.ww
      w..wwww
      wwwwwww
    - |
      wwwwwwwww
      ww.h....w
      ww...bA.w
      w....w..w
      wwwbw...w
      www...w.w
      wwwh....w
      wwwwwwwww

Actions:
  # Define the move action
  - Name: move
    Behaviours:
      # The agent can move around freely in empty space and over holes
      - Src:
          Object: avatar
          Commands:
            - mov: _dest
        Dst:
          Object: [_empty, hole]

      # Boxes can move into empty space
      - Src:
          Object: box
          Commands:
            - mov: _dest
        Dst:
          Object: _empty

      # The agent can push boxes
      - Src:
          Object: avatar
          Commands:
            - mov: _dest
        Dst:
          Object: box
          Commands:
            - cascade: _dest
```

(continues on next page)

(continued from previous page)

```yaml
      # If a box is moved into a hole remove it
      - Src:
          Object: box
          Commands:
            - remove: true
            - reward: 1
        Dst:
          Object: hole

Objects:
  - Name: box
    Z: 2
    MapCharacter: b
    Observers:
      Sprite2D:
        - Image: gvgai/newset/block1.png
      Block2D:
        - Shape: square
          Color: [1.0, 0.0, 0.0]
          Scale: 0.5

  - Name: wall
    MapCharacter: w
    Observers:
      Sprite2D:
        - TilingMode: WALL_16
          Image:
            - gvgai/oryx/wall3_0.png
            - gvgai/oryx/wall3_1.png
            - gvgai/oryx/wall3_2.png
            - gvgai/oryx/wall3_3.png
            - gvgai/oryx/wall3_4.png
            - gvgai/oryx/wall3_5.png
            - gvgai/oryx/wall3_6.png
            - gvgai/oryx/wall3_7.png
            - gvgai/oryx/wall3_8.png
            - gvgai/oryx/wall3_9.png
            - gvgai/oryx/wall3_10.png
            - gvgai/oryx/wall3_11.png
            - gvgai/oryx/wall3_12.png
            - gvgai/oryx/wall3_13.png
            - gvgai/oryx/wall3_14.png
            - gvgai/oryx/wall3_15.png
      Block2D:
        - Shape: triangle
          Color: [0.6, 0.6, 0.6]
          Scale: 0.9

  - Name: hole
    Z: 1
    MapCharacter: h
```

(continues on next page)

(continued from previous page)

```
    Observers:
      Sprite2D:
        - Image: gvgai/oryx/cspell4.png
      Block2D:
        - Shape: square
          Color: [0.0, 1.0, 0.0]
          Scale: 0.6

  - Name: avatar
    Z: 2
    MapCharacter: A
    Observers:
      Sprite2D:
        - Image: gvgai/oryx/knight1.png
      Block2D:
        - Shape: triangle
          Color: [0.2, 0.2, 0.6]
          Scale: 1.0
```

13.22 Zelda

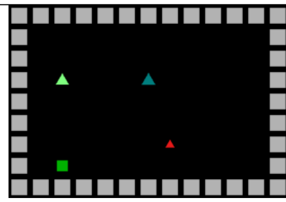
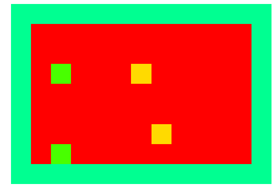
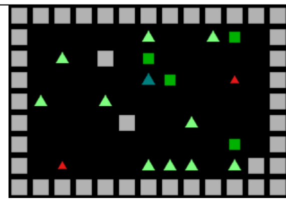
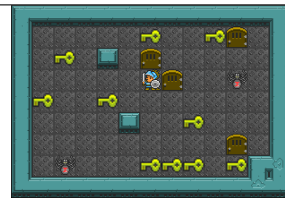
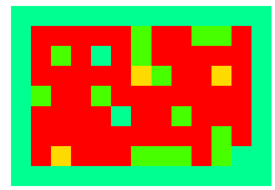
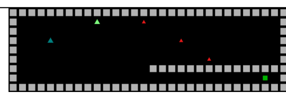
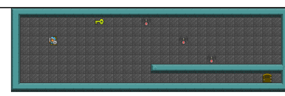
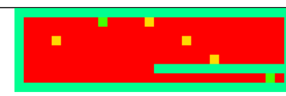
```
Single-Player/GVGAI/zelda.yaml
```

13.22.1 Description

A port of the GVGAI game “Zelda”. Pick up keys to reach doors and avoid enemies.

13.22.2 Levels

Table 43: Levels

<table>
<tr><th></th><th>Block2D</th><th>Sprite2D</th><th>Vector</th></tr>
<tr><td><table><tr><td>Level ID</td><td>0</td></tr><tr><td>Size</td><td>13x9</td></tr></table></td><td></td><td></td><td></td></tr>
<tr><td><table><tr><td>Level ID</td><td>1</td></tr><tr><td>Size</td><td>13x9</td></tr></table></td><td></td><td></td><td></td></tr>
<tr><td><table><tr><td>Level ID</td><td>2</td></tr><tr><td>Size</td><td>30x9</td></tr></table></td><td></td><td></td><td></td></tr>
</table>

13.22.3 Code Example

The most basic way to create a Griddly Gym Environment. Defaults to level 0 and SPRITE_2D rendering.

```
import gym
import griddly

if __name__ == '__main__':

    env = gym.make('GDY-Zelda-v0')
    env.reset()

    # Replace with your own control algorithm!
    for s in range(1000):
        obs, reward, done, info = env.step(env.action_space.sample())
        env.render() # Renders the environment from the perspective of a single player

        env.render(observer='global') # Renders the entire environment
```

(continues on next page)

(continued from previous page)

```
        if done:
            env.reset()
```

13.22.4 Objects

Table 44: Tiles

Name ->	avatar	attack_fire	key	goal	spider	wall
Map Char ->	*A*	*x*	*+*	*g*	*3*	*w*
Block2D						
Sprite2D						
Vector						

13.22.5 Actions

attack

Relative The actions are calculated relative to the object being controlled.

Action Id	Mapping
1	attack front

move

Relative The actions are calculated relative to the object being controlled.

Action Id	Mapping
1	Rotate left
2	Move forwards
3	Rotate right
4	Move Backwards

random_movement

Internal This action can only be called from other actions, not by the player.

Action Id	Mapping
1	Left
2	Up
3	Right
4	Down

remove_sprite

Internal This action can only be called from other actions, not by the player.

Action Id	Mapping
1	Left
2	Up
3	Right
4	Down

13.22.6 YAML

```yaml
Version: "0.1"
Environment:
  Name: Zelda
  Description: A port of the GVGAI game "Zelda". Pick up keys to reach doors and avoid␣
↪enemies.
  Observers:
    Sprite2D:
      TileSize: 24
      BackgroundTile: gvgai/newset/floor2.png
  Player:
    AvatarObject: avatar
  Termination:
    Win:
      - eq: [goal:count, 0]
    Lose:
      - eq: [avatar:count, 0]
  Levels:
    - |
      wwwwwwwwwwwww
      w...........w
      w...........w
      w.+...A.....w
      w...........w
      w...........w
      w......3....w
      w.g.........w
      wwwwwwwwwwwww
    - |
      wwwwwwwwwwwww
      w.....+..+g.w
      w.+.w.g.....w
      w.....Ag..3.w
      w+..+.......w
      w....w..+...w
      w.........g.w
      w.3...+++.+ww
      wwwwwwwwwwwww
    - |
      wwwwwwwwwwwwwwwwwwwwwwwwwwwwwww
```

(continues on next page)

(continued from previous page)

```
      w........+....3.............w
      w...........................w
      w...A............3.........w
      w...........................w
      w...................3.......w
      w.............wwwwwwwwwwwwwww
      w.........................g.w
      wwwwwwwwwwwwwwwwwwwwwwwwwwwwwww
Actions:
  # Define action that cannot be controlled by the player.
  # (In this case the spider movement)
  - Name: random_movement
    InputMapping:
      Internal: true
    Behaviours:
      # The gnome and the spider can move into empty space
      - Src:
          Object: spider
          Commands:
            - mov: _dest
            - exec:
                Action: random_movement
                Delay: 5
                Randomize: true
        Dst:
          Object: _empty
      # The spider will not move into the wall, but it needs to keep moving
      - Src:
          Object: spider
          Commands:
            - exec:
                Action: random_movement
                Delay: 5
                Randomize: true
        Dst:
          Object: [wall, key, goal, spider]
      # If the gnome moves into a spider
      - Src:
          Object: spider
        Dst:
          Object: avatar
          Commands:
            - remove: true
            - reward: -1
  # remove the recently spawned attack animation sprite
  - Name: remove_sprite
    InputMapping:
      Internal: true
    Behaviours:
      - Src:
          Object: attack_fire
          Commands:
```

(continues on next page)

(continued from previous page)

```
          - remove: true
        Dst:
          Object: attack_fire
  # Define the move action
  - Name: move
    InputMapping:
      Inputs:
        1:
          Description: Rotate left
          OrientationVector: [-1, 0]
        2:
          Description: Move forwards
          OrientationVector: [0, -1]
          VectorToDest: [0, -1]
        3:
          Description: Rotate right
          OrientationVector: [1, 0]
        4:
          Description: Move Backwards
          VectorToDest: [0, 1]
          OrientationVector: [0, -1]
      Relative: true
    Behaviours:
      # Tell the gnome to rotate if it performs an action on itself (Rotate left and␣
↪Rotate right actions)
      - Src:
          Object: avatar
          Commands:
            - rot: _dir
        Dst:
          Object: avatar
      # Only an avatar with a key can win
      - Src:
          Preconditions:
            - eq: [src.has_key, 1]
          Object: avatar
          Commands:
            - reward: 1
            # - decr: has_key
            - mov: _dest
            # - set_tile: 0
        Dst:
          Object: goal
          Commands:
            - remove: true
      # If the gnome moves into a gem object, the stick is removed, triggering a win␣
↪condition
      - Src:
          Object: avatar
          Commands:
            - mov: _dest
            - eq:
```

(continues on next page)

(continued from previous page)

```
                Arguments: [ src.has_key, 0 ]
                Commands:
                  - incr: has_key
                  - reward: 1
                  - set_tile: 1
        Dst:
          Object: key
          Commands:
            - eq:
                Arguments: [ src.has_key, 0 ]
                Commands:
                  - remove: true
      # If the gnome moves into a spider
      - Src:
          Object: avatar
          Commands:
            - remove: true
            - reward: -1
        Dst:
          Object: spider
      # The gnome and the spider can move into empty space
      - Src:
          Object: avatar
          Commands:
            - mov: _dest
        Dst:
          Object: _empty
  - Name: attack
    InputMapping:
      Inputs:
        1:
          Description: attack front
          OrientationVector: [ 1, 0 ]
          VectorToDest: [1, 0]
      Relative: true
    Behaviours:
      - Src:
          Object: avatar
          Commands:
            - spawn: attack_fire
        Dst:
          Object: spider
          Commands:
            - remove: true
      - Src:
          Object: avatar
          Commands:
            - spawn: attack_fire
        Dst:
          Object: _empty

Objects:
```

(continues on next page)

(continued from previous page)

```
- Name: avatar
  Z: 3
  MapCharacter: A
  Variables:
    - Name: has_key
  Observers:
    Sprite2D:
      - Image: gvgai/oryx/swordman1_0.png
      - Image: gvgai/oryx/swordmankey1_0.png
    Block2D:
      - Shape: triangle
        Color: [0.0, 0.5, 0.5]
        Scale: 0.75
      - Shape: triangle
        Color: [0.3, 0.5, 0.2]
        Scale: 1.0
- Name: attack_fire
  Z: 1
  InitialActions:
    - Action: remove_sprite
      Delay: 3
  MapCharacter: x
  Observers:
    Sprite2D:
      - Image: gvgai/oryx/fire1.png
    Block2D:
      - Shape: square
        Color: [1.0, 0.0, 0.0]
        Scale: 0.5
- Name: key
  Z: 2
  MapCharacter: "+"
  Observers:
    Sprite2D:
      - Image: gvgai/oryx/key2.png
    Block2D:
      - Shape: triangle
        Color: [0.5, 1.0, 0.5]
        Scale: 0.7
- Name: goal
  Z: 2
  MapCharacter: g
  Observers:
    Sprite2D:
      - Image: gvgai/oryx/doorclosed1.png
    Block2D:
      - Shape: square
        Color: [0.0, 0.7, 0.0]
        Scale: 0.7
#   - Name: chaser
#     Z: 2
#     MapCharacter: "3"
```

(continues on next page)

(continued from previous page)

```
#     Observers:
#       Sprite2D:
#         - Image: gvgai/oryx/skeleton1.png
- Name: spider
  Z: 2
  InitialActions:
    - Action: random_movement
      Delay: 5
  MapCharacter: "3"
  Observers:
    Sprite2D:
      - Image: oryx/oryx_fantasy/avatars/spider1.png
    Block2D:
      - Shape: triangle
        Color: [0.9, 0.1, 0.1]
        Scale: 0.5
- Name: wall
  MapCharacter: w
  Observers:
    Sprite2D:
      - TilingMode: WALL_16
        Image:
          - gvgai/oryx/wall3_0.png
          - gvgai/oryx/wall3_1.png
          - gvgai/oryx/wall3_2.png
          - gvgai/oryx/wall3_3.png
          - gvgai/oryx/wall3_4.png
          - gvgai/oryx/wall3_5.png
          - gvgai/oryx/wall3_6.png
          - gvgai/oryx/wall3_7.png
          - gvgai/oryx/wall3_8.png
          - gvgai/oryx/wall3_9.png
          - gvgai/oryx/wall3_10.png
          - gvgai/oryx/wall3_11.png
          - gvgai/oryx/wall3_12.png
          - gvgai/oryx/wall3_13.png
          - gvgai/oryx/wall3_14.png
          - gvgai/oryx/wall3_15.png
    Block2D:
      - Shape: square
        Color: [0.7, 0.7, 0.7]
        Scale: 1.0
```

13.23 Clusters

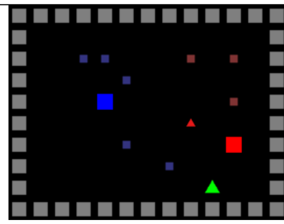
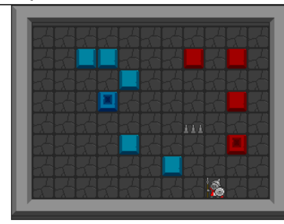
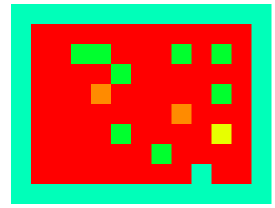
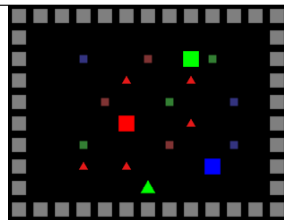
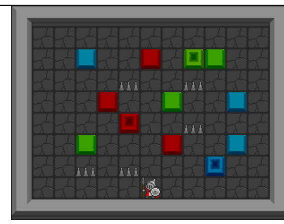
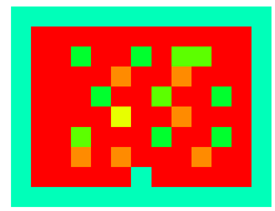
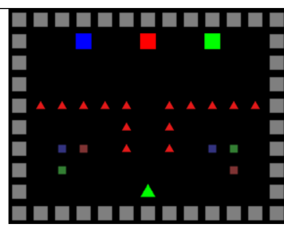
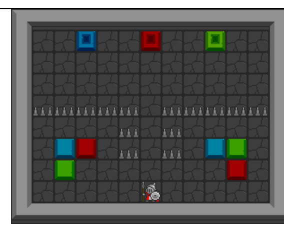
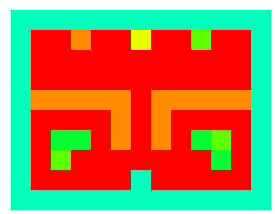
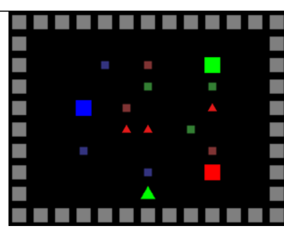
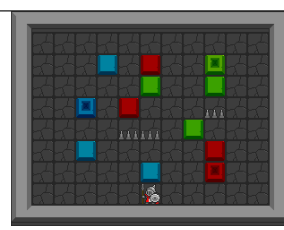
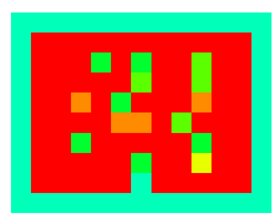
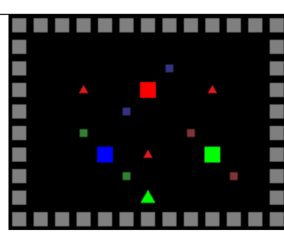
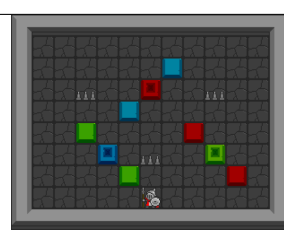
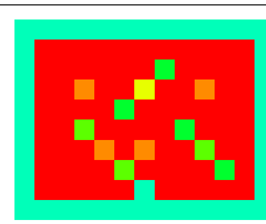
```
Single-Player/GVGAI/clusters.yaml
```

13.23.1 Description

Cluster the coloured objects together by pushing them against the static coloured blocks.

13.23.2 Levels

Table 45: Levels

	Block2D	Sprite2D	Vector
Level ID: 0 Size: 13x10			
Level ID: 1 Size: 13x10			
Level ID: 2 Size: 13x10			
Level ID: 3 Size: 13x10			
Level ID: 4 Size: 13x10			

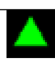
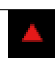
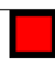
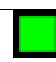
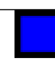
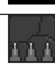
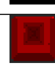
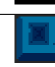
13.23.3 Code Example

The most basic way to create a Griddly Gym Environment. Defaults to level 0 and SPRITE_2D rendering.

```python
import gym
import griddly

if __name__ == '__main__':

    env = gym.make('GDY-Clusters-v0')
    env.reset()

    # Replace with your own control algorithm!
    for s in range(1000):
        obs, reward, done, info = env.step(env.action_space.sample())
        env.render() # Renders the environment from the perspective of a single player

        env.render(observer='global') # Renders the entire environment

        if done:
            env.reset()
```

13.23.4 Objects

Table 46: Tiles

Name ->	avatar	wall	spike	red_box	red_block	green_box	green_block	blue_box	blue_block
Map Char ->	*A*	*w*	*h*	*2*	*b*	*3*	*c*	*1*	*a*
Block2D									
Sprite2D									
Vector									

13.23.5 Actions

move

Action Id	Mapping
1	Left
2	Up
3	Right
4	Down

box_counter

Internal This action can only be called from other actions, not by the player.

Action Id	Mapping
1	The only action here is to increment the box count

13.23.6 YAML

```
Version: "0.1"
Environment:
  Name: Clusters
  Description: Cluster the coloured objects together by pushing them against the static␣
↪coloured blocks.
  Observers:
    Sprite2D:
      TileSize: 24
      BackgroundTile: oryx/oryx_fantasy/floor1-2.png
  Variables:
    - Name: box_count
      InitialValue: 0
  Player:
    AvatarObject: avatar # The player can only control a single avatar in the game
  Termination:
    Win:
      - eq: [box_count, 0]
    Lose:
      - eq: [broken_box:count, 1]
      - eq: [avatar:count, 0]
  Levels:
    - |
      w w w w w w w w w w w w w
      w . . . . . . . . . . . w
      w . . 1 1 . . . 2 . 2 . w
      w . . . . 1 . . . . . . w
      w . . . a . . . . . 2 . w
      w . . . . . . . h . . . w
      w . . . . 1 . . . . b . w
      w . . . . . . 1 . . . . w
      w . . . . . . . . A . . w
      w w w w w w w w w w w w w
    - |
      w w w w w w w w w w w w w
      w . . . . . . . . . . . w
      w . . 1 . . 2 . c 3 . . w
      w . . . . h . . h . . . w
      w . . . 2 . . 3 . . 1 . w
      w . . . . b . . h . . . w
      w . . 3 . . . 2 . . 1 . w
      w . . h . h . . . a . . w
      w . . . . . A . . . . . w
```

(continues on next page)

(continued from previous page)

```
        w w w w w w w w w w w w w
      - |
        w w w w w w w w w w w w w
        w . . a . . b . . c . . w
        w . . . . . . . . . . . w
        w . . . . . . . . . . . w
        w h h h h h . h h h h h w
        w . . . . h . h . . . . w
        w . 1 2 . h . h . 1 3 . w
        w . 3 . . . . . . . 2 . w
        w . . . . . A . . . . . w
        w w w w w w w w w w w w w
      - |
        w w w w w w w w w w w w w
        w . . . . . . . . . . . w
        w . . . 1 . 2 . . c . . w
        w . . . . . 3 . . 3 . . w
        w . . a . 2 . . . h . . w
        w . . . . h h . 3 . . . w
        w . . 1 . . . . . 2 . . w
        w . . . . . 1 . . b . . w
        w . . . . . A . . . . . w
        w w w w w w w w w w w w w
      - |
        w w w w w w w w w w w w w
        w . . . . . . . . . . . w
        w . . . . . . 1 . . . . w
        w . . h . . b . . h . . w
        w . . . . 1 . . . . . . w
        w . . 3 . . . . 2 . . . w
        w . . . a . h . . c . . w
        w . . . . 3 . . . . 2 . w
        w . . . . . A . . . . . w
        w w w w w w w w w w w w w

Actions:

  # A simple action to count the number of boxes in the game at the start
  # Not currently a way to do complex things in termination conditions like combine␣
↪multiple conditions
  - Name: box_counter
    InputMapping:
      Internal: true
      Inputs:
        1:
          Description: "The only action here is to increment the box count"
    Behaviours:
      - Src:
          Object: [blue_box, red_box, green_box]
          Commands:
            - incr: box_count
        Dst:
```

(continues on next page)

(continued from previous page)

```
        Object: [blue_box, red_box, green_box]

  # Define the move action
  - Name: move
    Behaviours:

      # Avatar and boxes can move into empty space
      - Src:
          Object: [avatar, blue_box, green_box, red_box]
          Commands:
            - mov: _dest
        Dst:
          Object: _empty

      # Boxes can be pushed by the avatar
      - Src:
          Object: avatar
          Commands:
            - mov: _dest
        Dst:
          Object: [blue_box, green_box, red_box]
          Commands:
            - cascade: _dest

      # When boxes are pushed against the blocks they change
      - Src:
          Object: blue_box
          Commands:
            - change_to: blue_block
            - reward: 1
            - decr:  box_count
        Dst:
          Object: blue_block
      - Src:
          Object: red_box
          Commands:
            - reward: 1
            - change_to: red_block
            - decr:  box_count
        Dst:
          Object: red_block
      - Src:
          Object: green_box
          Commands:
            - reward: 1
            - change_to: green_block
            - decr:  box_count
        Dst:
          Object: green_block

      # Boxes break if they hit the spikes
      - Src:
```

(continues on next page)

(continued from previous page)

```
          Object: [blue_box, green_box, red_box]
          Commands:
            - change_to: broken_box
            - reward: -1
        Dst:
          Object: spike

      # Avatar dies if it hits the spikes
      - Src:
          Object: avatar
          Commands:
            - remove: true
            - reward: -1
        Dst:
          Object: spike

Objects:
  - Name: avatar
    MapCharacter: A
    Observers:
      Sprite2D:
        - Image: gvgai/oryx/knight1.png
      Block2D:
        - Shape: triangle
          Color: [0.0, 1.0, 0.0]
          Scale: 0.8

  - Name: wall
    MapCharacter: w
    Observers:
      Sprite2D:
        - TilingMode: WALL_16
          Image:
            - oryx/oryx_fantasy/wall1-0.png
            - oryx/oryx_fantasy/wall1-1.png
            - oryx/oryx_fantasy/wall1-2.png
            - oryx/oryx_fantasy/wall1-3.png
            - oryx/oryx_fantasy/wall1-4.png
            - oryx/oryx_fantasy/wall1-5.png
            - oryx/oryx_fantasy/wall1-6.png
            - oryx/oryx_fantasy/wall1-7.png
            - oryx/oryx_fantasy/wall1-8.png
            - oryx/oryx_fantasy/wall1-9.png
            - oryx/oryx_fantasy/wall1-10.png
            - oryx/oryx_fantasy/wall1-11.png
            - oryx/oryx_fantasy/wall1-12.png
            - oryx/oryx_fantasy/wall1-13.png
            - oryx/oryx_fantasy/wall1-14.png
            - oryx/oryx_fantasy/wall1-15.png
      Block2D:
        - Shape: square
          Color: [0.5, 0.5, 0.5]
```

(continues on next page)

(continued from previous page)

```yaml
        Scale: 0.9

  - Name: spike
    MapCharacter: h
    Observers:
      Sprite2D:
        - Image: gvgai/oryx/spike2.png
      Block2D:
        - Shape: triangle
          Color: [0.9, 0.1, 0.1]
          Scale: 0.5

  - Name: red_box
    MapCharacter: "2"
    InitialActions:
      - Action: box_counter
        ActionId: 1
    Observers:
      Sprite2D:
        - Image: gvgai/newset/blockR.png
      Block2D:
        - Shape: square
          Color: [0.5, 0.2, 0.2]
          Scale: 0.5
  - Name: red_block
    MapCharacter: b
    Observers:
      Sprite2D:
        - Image: gvgai/newset/blockR2.png
      Block2D:
        - Shape: square
          Color: [1.0, 0.0, 0.0]
          Scale: 1.0

  - Name: green_box
    MapCharacter: "3"
    InitialActions:
      - Action: box_counter
        ActionId: 1
    Observers:
      Sprite2D:
        - Image: gvgai/newset/blockG.png
      Block2D:
        - Shape: square
          Color: [0.2, 0.5, 0.2]
          Scale: 0.5
  - Name: green_block
    MapCharacter: c
    Observers:
      Sprite2D:
        - Image: gvgai/newset/blockG2.png
      Block2D:
```

(continues on next page)

(continued from previous page)

```
      - Shape: square
        Color: [0.0, 1.0, 0.0]
        Scale: 1.0

- Name: blue_box
  MapCharacter: "1"
  InitialActions:
    - Action: box_counter
      ActionId: 1
  Observers:
    Sprite2D:
      - Image: gvgai/newset/blockB.png
    Block2D:
      - Shape: square
        Color: [0.2, 0.2, 0.5]
        Scale: 0.5
- Name: blue_block
  MapCharacter: a
  Observers:
    Sprite2D:
      - Image: gvgai/newset/blockB2.png
    Block2D:
      - Shape: square
        Color: [0.0, 0.0, 1.0]
        Scale: 1.0

- Name: broken_box
  Observers:
    Sprite2D:
      - Image: gvgai/newset/block3.png
    Block2D:
      - Shape: triangle
        Color: [1.0, 0.0, 1.0]
        Scale: 1.0
```

13.24 Zen Puzzle

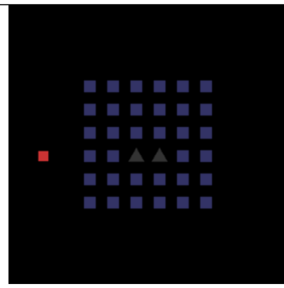
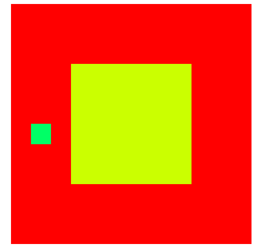
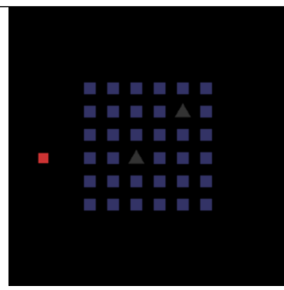
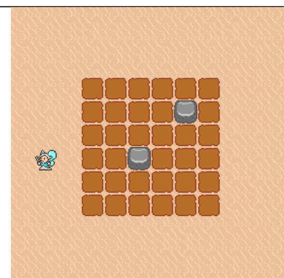
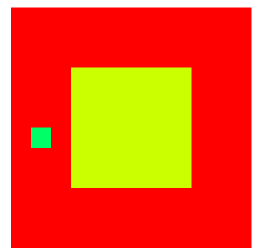
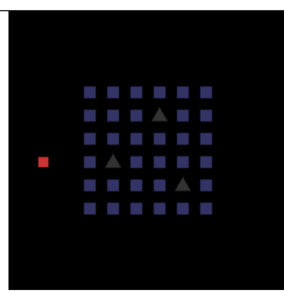
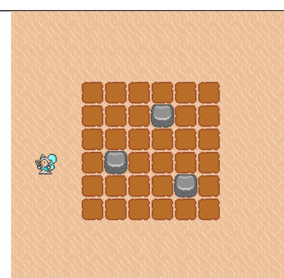
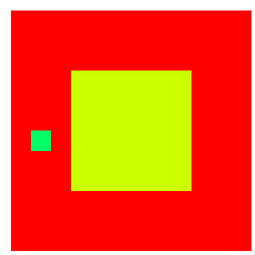
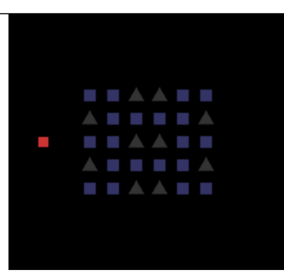
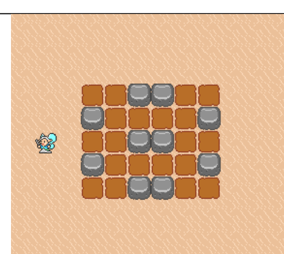
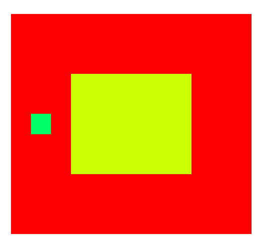
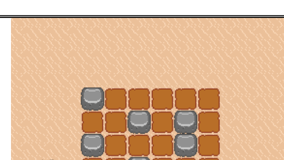
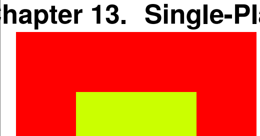
```
Single-Player/GVGAI/zenpuzzle.yaml
```

13.24.1 Description

Set all the tiles in the level to the same color, but you cannot move over a tile more than once! (Not even sure why this is zen its super frustrating)

13.24.2 Levels

Table 47: Levels

<table>
<tr><th></th><th>Block2D</th><th>Sprite2D</th><th>Vector</th></tr>
<tr><td><table><tr><td>Level ID</td><td>0</td></tr><tr><td>Size</td><td>12x12</td></tr></table></td><td></td><td></td><td></td></tr>
<tr><td><table><tr><td>Level ID</td><td>1</td></tr><tr><td>Size</td><td>12x12</td></tr></table></td><td></td><td></td><td></td></tr>
<tr><td><table><tr><td>Level ID</td><td>2</td></tr><tr><td>Size</td><td>12x12</td></tr></table></td><td></td><td></td><td></td></tr>
<tr><td><table><tr><td>Level ID</td><td>3</td></tr><tr><td>Size</td><td>12x11</td></tr></table></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td></tr>
</table>

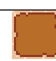
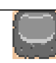
13.24.3 Code Example

The most basic way to create a Griddly Gym Environment. Defaults to level 0 and SPRITE_2D rendering.

```python
import gym
import griddly

if __name__ == '__main__':

    env = gym.make('GDY-Zen-Puzzle-v0')
    env.reset()

    # Replace with your own control algorithm!
    for s in range(1000):
        obs, reward, done, info = env.step(env.action_space.sample())
        env.render() # Renders the environment from the perspective of a single player

        env.render(observer='global') # Renders the entire environment

        if done:
            env.reset()
```

13.24.4 Objects

Table 48: Tiles

Name ->	avatar	ground	rock
Map Char ->	*A*	*g*	*r*
Block2D			
Sprite2D			
Vector			

13.24.5 Actions

move

Action Id	Mapping
1	Left
2	Up
3	Right
4	Down

13.24.6 YAML

```yaml
Version: "0.1"
Environment:
  Name: Zen Puzzle
  Description: Set all the tiles in the level to the same color, but you cannot move␣
↪over a tile more than once! (Not even sure why this is zen its super frustrating)
  Observers:
    Sprite2D:
      TileSize: 24
      BackgroundTile: gvgai/oryx/backBiege.png
  Player:
    AvatarObject: avatar
  Termination:
    Win:
      - eq: [ground:count, 0]
    Lose:
      - eq: [_steps, 1000]
  Levels:
    - |
      ............
      ............
      ............
      ...gggggg...
      ...gggggg...
      ...gggggg...
      .A.ggrrgg...
      ...gggggg...
      ...gggggg...
      ............
      ............
      ............
    - |
      ............
      ............
      ............
      ...gggggg...
      ...ggggrg...
      ...gggggg...
      .A.ggrggg...
      ...gggggg...
      ...gggggg...
      ............
      ............
      ............
    - |
      ............
      ............
      ............
      ...gggggg...
      ...gggrgg...
      ...gggggg...
      .A.grgggg...
```

(continues on next page)

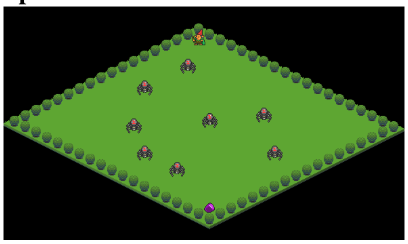
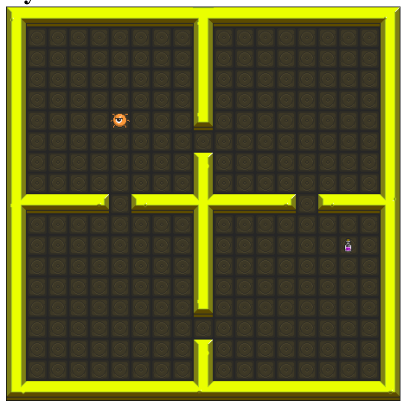
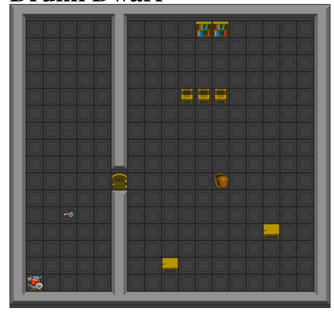
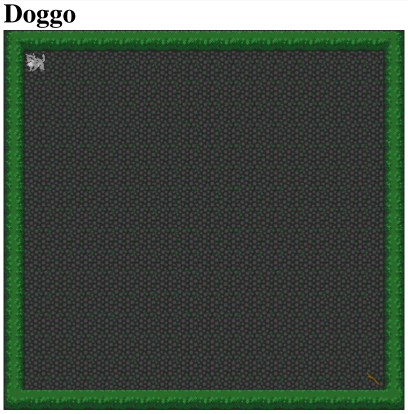
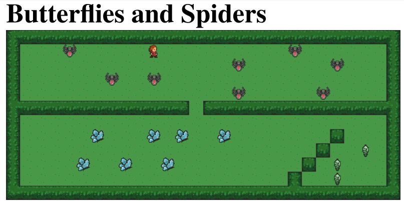
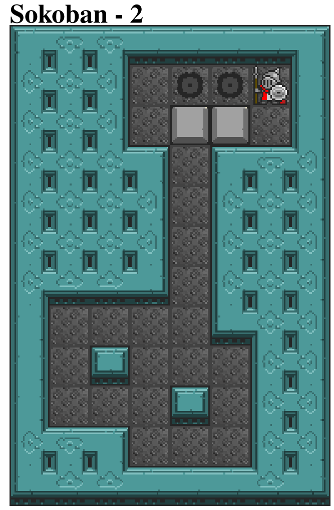
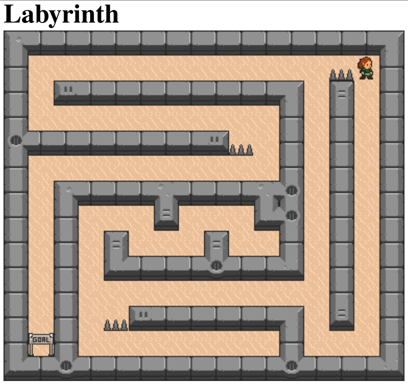
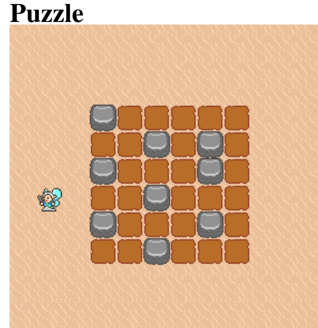
(continued from previous page)

```
      ...ggggrg...
      ...gggggg...
      ............
      ............
      ............
    - |
      ............
      ............
      ............
      ...ggrrgg...
      ...rggggr...
      .A.ggrrgg...
      ...rggggr...
      ...ggrrgg...
      ............
      ............
      ............
    - |
      ............
      ............
      ............
      ...rggggg...
      ...ggrgrg...
      ...rgggrg...
      .A.ggrggg...
      ...rgggrg...
      ...ggrggg...
      ............
      ............
      ............

Actions:
  # Define the move action
  - Name: move
    Behaviours:
      # The agent can move around freely in empty space and over holes
      - Src:
          Object: avatar
          Commands:
            - mov: _dest

        Dst:
          Object: _empty

      - Src:
          Object: avatar
          Commands:
            - mov: _dest
        Dst:
          Object: ground
          Commands:
            - change_to: walked
```

(continues on next page)

(continued from previous page)

```yaml
            - reward: 1

Objects:
  - Name: avatar
    MapCharacter: A
    Z: 1
    Observers:
      Sprite2D:
        - Image: gvgai/oryx/angel1.png
      Block2D:
        - Shape: square
          Color: [0.8, 0.2, 0.2]
          Scale: 0.6

  - Name: ground
    MapCharacter: g
    Observers:
      Sprite2D:
        - Image: gvgai/oryx/floorTileOrange.png
      Block2D:
        - Shape: square
          Color: [0.2, 0.2, 0.4]
          Scale: 0.7

  - Name: walked
    Z: 0
    Observers:
      Sprite2D:
        - Image: gvgai/oryx/floorTileGreen.png
      Block2D:
        - Shape: square
          Color: [0.2, 0.6, 0.2]
          Scale: 0.8

  - Name: rock
    MapCharacter: r
    Observers:
      Sprite2D:
        - Image: gvgai/oryx/wall5.png
      Block2D:
        - Shape: triangle
          Color: [0.2, 0.2, 0.2]
          Scale: 0.8
```

Spiders  A port of the games provided in the https://github.com/maximecb/gym-minigrid Dynamic obstacles environment, but you're a gnome avoiding ghosts to get to a gem.	**Eyeball**  A port of the games provided in the https://github.com/maximecb/gym-minigrid 4 Rooms environment, but you're a giant eye looking for it's eyedrops because everything is yellow and it hurts to look at.	**Drunk Dwarf**  A port of the games provided in the https://github.com/maximecb/gym-minigrid environment, but you're a drunk dwarf trying find your keys that you've dropped to get to your bed (which is a coffin?? Wierd.).
Doggo  A port of the games provided in the https://github.com/maximecb/gym-minigrid Empty environment, but you're a doggo fetching a stick.	**Butterflies and Spiders**  You want to catch all of the butterflies while also avoiding the spiders. Butterflies spawn slowly from cocoons. The butterflies are also eaten by the spiders so you need to be fast to collect them. You win the level as soon as there are no butterflies on the screen.	**Partially Observable Sokoban - 2**  Push the boxes onto the marked spaces, once a box has moved onto a space, it cannot be moved
Labyrinth  Its a maze, find your way out. Watch out for spikey things.	**Bait**  Get the key and unlock the door. Fill in the holes in the floor with blocks to	**Partially Observable Zen Puzzle**  Set all the tiles in the level to the same color, but you cannot move over a tile more than once! (Not even sure why this is zen its super frustrating)

CHAPTER

FOURTEEN

RTS

14.1 GriddlyRTS

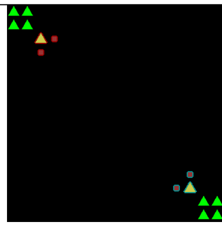
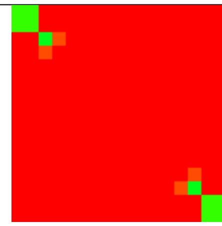
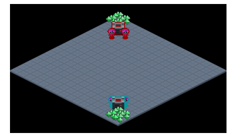
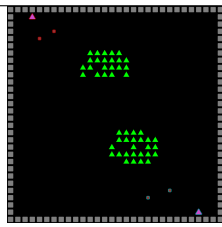
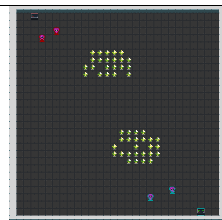
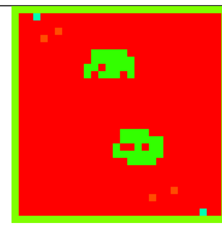
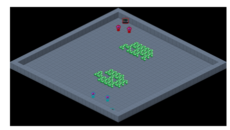
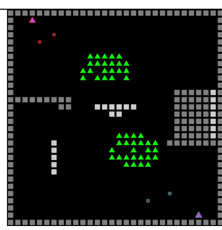
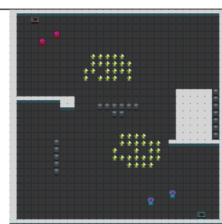
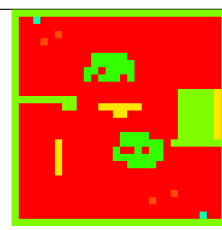
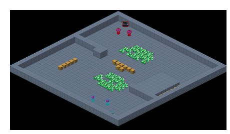
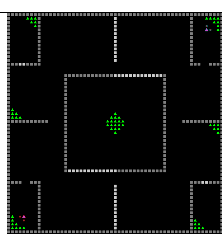
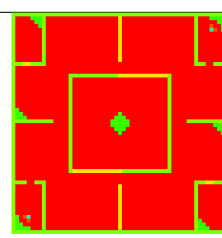
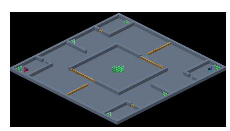
```
RTS/GriddlyRTS.yaml
```

14.1.1 Description

An RTS Game. There's aliens and stuff.

14.1.2 Levels

Table 1: Levels

	Block2D	Sprite2D	Vector	Isometric
Level ID: 0 Size: 16x16				
Level ID: 1 Size: 30x30				
Level ID: 2 Size: 30x30				
Level ID: 3 Size: 57x58				

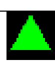
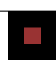
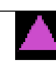
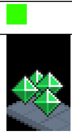
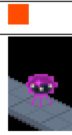
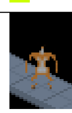
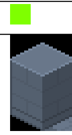
14.1.3 Code Example

The most basic way to create a Griddly Gym Environment. Defaults to level 0 and SPRITE_2D rendering.

```python
import gym
import griddly
from griddly.util.wrappers import InvalidMaskingRTSWrapper

if __name__ == '__main__':

    env = gym.make('GDY-GriddlyRTS-v0')
    env.reset()
    env = InvalidMaskingRTSWrapper(env)

    # Replace with your own control algorithm!
    for s in range(1000):
        obs, reward, done, info = env.step(env.action_space.sample())
        for p in range(env.player_count):
            env.render(observer=p) # Renders the environment from the perspective of a␣
↪single player

        env.render(observer='global') # Renders the entire environment

        if done:
            env.reset()
```

14.1.4 Objects

Table 2: Tiles

Name ->	minerals	worker	ranged	combat	fixed_wall	movable_wall	base	barracks_disabled	barracks
Map Char ->	*M*	*H*	*r*	*c*	*W*	*w*	*A*	*b*	*B*
Block2D									
Sprite2D									
Vector									
Isometric									

14.1.5 Actions

build_barracks

Action Id	Mapping
1	Left
2	Up
3	Right
4	Down

move

Action Id	Mapping
1	Left
2	Up
3	Right
4	Down

spawn_combat

Internal This action can only be called from other actions, not by the player.

Action Id	Mapping
1	Left
2	Up
3	Right
4	Down

attack

Action Id	Mapping
1	Left
2	Up
3	Right
4	Down

spawn_worker

Internal This action can only be called from other actions, not by the player.

Action Id	Mapping
1	Left
2	Up
3	Right
4	Down

construct_barracks

Internal This action can only be called from other actions, not by the player.

Action Id	Mapping
1	Completes construction of a barracks

build_combat

Action Id	Mapping
1	Build

gather

Action Id	Mapping
1	Left
2	Up
3	Right
4	Down

build_worker

Action Id	Mapping
1	Build

14.1.6 YAML

```yaml
Version: "0.1"
Environment:
  Name: GriddlyRTS
  Description: An RTS Game. There's aliens and stuff.
  Observers:
    Sprite2D:
      TileSize: 16
      BackgroundTile: oryx/oryx_tiny_galaxy/tg_sliced/tg_world/tg_world_floor_panel_
↪metal_a.png
    Isometric:
      TileSize: [ 32, 48 ]
      BackgroundTile: oryx/oryx_iso_dungeon/floor-1.png
      IsoTileHeight: 16
      IsoTileDepth: 4
    Vector:
      IncludePlayerId: true
      IncludeVariables: true
  Variables:
    - Name: player_resources
```

(continues on next page)

(continued from previous page)

```yaml
      InitialValue: 0
      PerPlayer: true
  Player:
    Count: 2
  Termination:
    Lose:
      - Conditions:
          - eq: [ base:count, 0 ] # If the player has no bases
        Reward: -10 # -10 for a loss
        OpposingReward: 10 # as the agent didnt lose
  Levels:
    - |
      M  M  .  .  .  .  .  .  .  .  .  .  .  .  .  .
      M  M  .  .  .  .  .  .  .  .  .  .  .  .  .  .
      .  .  A1 H1 .  .  .  .  .  .  .  .  .  .  .  .
      .  .  H1 .  .  .  .  .  .  .  .  .  .  .  .  .
      .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
      .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
      .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
      .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
      .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
      .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
      .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
      .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
      .  .  .  .  .  .  .  .  .  .  .  .  .  H2 .  .
      .  .  .  .  .  .  .  .  .  .  .  .  H2 A2 .  .
      .  .  .  .  .  .  .  .  .  .  .  .  .  .  M  M
      .  .  .  .  .  .  .  .  .  .  .  .  .  .  M  M
    - |
      W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W
      W  .  .  B1 .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  W
      W  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  W
      W  .  .  .  .  .  H1 .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  W
      W  .  .  .  H1 .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  W
      W  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  W
      W  .  .  .  .  .  .  .  .  .  .  M  M  M  M  M  .  .  .  .  .  .  .  .  .  .  .  .  .  W
      W  .  .  .  .  .  .  .  .  .  .  M  M  M  M  M  M  .  .  .  .  .  .  .  .  .  .  .  .  W
      W  .  .  .  .  .  .  .  .  .  M  M  .  M  M  M  M  .  .  .  .  .  .  .  .  .  .  .  .  W
      W  .  .  .  .  .  .  .  .  .  M  .  M  M  M  .  M  .  .  .  .  .  .  .  .  .  .  .  .  W
      W  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  W
      W  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  W
```

(continues on next page)

(continued from previous page)

```
W . . . . . . . . . . . . . . . . . . . . . . . . . . . . W
W . . . . . . . . . . . . . . . . . . . . . . . . . . . . W
W . . . . . . . . . . . . . . . . . . . . . . . . . . . . W
W . . . . . . . . . . . . . . . . . . . . . . . . . . . . W
W . . . . . . . . . . . . . . . . . . . . . . . . . . . . W
W . . . . . . . . . . . . . . M M M M . . . . . . . . . . W
W . . . . . . . . . . . . . . M M M M M M . . . . . . . . W
W . . . . . . . . . . . . . M . . M . M M . . . . . . . . W
W . . . . . . . . . . . . . M M M M M M M . . . . . . . . W
W . . . . . . . . . . . . . . . M M M M . . . . . . . . . W
W . . . . . . . . . . . . . . . . . . . . . . . . . . . . W
W . . . . . . . . . . . . . . . . . . . . . . . . . . . . W
W . . . . . . . . . . . . . . . . . . . . . . . . . . . . W
W . . . . . . . . . . . . . . . . . . . . . H2 . . . . . . W
W . . . . . . . . . . . . . . . . . . H2 . . . . . . . . . W
W . . . . . . . . . . . . . . . . . . . . . . . . . . . . W
W . . . . . . . . . . . . . . . . . . . . . . . . . B2 . . W
W W W W W W W W W W W W W W W W W W W W W W W W W W W W W W
  - |
W W W W W W W W W W W W W W W W W W W W W W W W W W W W W W
W . . B1 . . . . . . . . . . . . . . . . . . . . . . . . . W
W . . . . . . . . . . . . . . . . . . . . . . . . . . . . W
W . . . . . H1 . . . . . . . . . . . . . . . . . . . . . . W
W . . . H1 . . . . . . . . . . . . . . . . . . . . . . . . W
W . . . . . . . . . . . . . . . . . . . . . . . . . . . . W
W . . . . . . . . . . M M M M M . . . . . . . . . . . . . W
W . . . . . . . . . . M M M M M M . . . . . . . . . . . . W
```

(continues on next page)

(continued from previous page)

```
    W . . . . . . . . . M M . M M M M . . . . . . . . . . .␣
↪ . W
    W . . . . . . . . . M . M M M . M . . . . . . . . . . .␣
↪ . W
    W . . . . . . . . . . . . . . . . . . . . . . . . . . .␣
↪ . W
    W . . . . . . . . . . . . . . . . . . . . . . W W W W W␣
↪ w W
    W W W W W W W W W . . . . . . . . . . . . . . W W W W W␣
↪ w W
    W . . . . . . W W . . . w w w w w w . . . . . W W W W W␣
↪ w W
    W . . . . . . . . . . . . . w w . . . . . . . W W W W W␣
↪ w W
    W . . . . . . . . . . . . . . . . . . . . . . W W W W W␣
↪ w W
    W . . . . . . . . . . . . . . . . . . . . . . W W W W W␣
↪ w W
    W . . . . . . . . . . . . . . M M M M . . . . W W W W W␣
↪ w W
    W . . . . . w . . . . . . . . M M M M M M . W W W W W W␣
↪ W W
    W . . . . . w . . . . . . . M . . M . M M . . . . . . .␣
↪ . W
    W . . . . . w . . . . . . . M M M M M M M . . . . . . .␣
↪ . W
    W . . . . . w . . . . . . . . . M M M M . . . . . . . .␣
↪ . W
    W . . . . . w . . . . . . . . . . . . . . . . . . . . .␣
↪ . W
    W . . . . . . . . . . . . . . . . . . . . . . . . . . .␣
↪ . W
    W . . . . . . . . . . . . . . . . . . . . . . . . . . .␣
↪ . W
    W . . . . . . . . . . . . . . . . . . . . . H2 . . . . .␣
↪ . W
    W . . . . . . . . . . . . . . . . . . H2 . . . . . . . .␣
↪ . W
    W . . . . . . . . . . . . . . . . . . . . . . . . . . .␣
↪ . W
    W . . . . . . . . . . . . . . . . . . . . . . . . . B2 .␣
↪ . W
    W W W W W W W W W W W W W W W W W W W W W W W W W W W W␣
↪ W W
  - |
    W W W W W W W W W W W W W W W W W W W W W W W W W W W W␣
↪ W W W W W W W W W W W W W W W W W W W W W W W W W W W W W
    W . . . . M M M W . . . . . . . . . . . . . . . . . . .␣
↪ w . . . . . . . . . . . . . . . . . . . W . . . M M M M W
    W . . . . . M M W . . . . . . . . . . . . . . . . . . .␣
↪ w . . . . . . . . . . . . . . . . . . . W . . . . . M M W
    W . . . . . . M W . . . . . . . . . . . . . . . . . . .␣
↪ w . . . . . . . . . . . . . . . . . . . W . . . H2 . . M W
```

(continues on next page)

(continued from previous page)

```
    W . . . . . . . W . . . . . . . . . . . . . . . . . . .
↪ w . . . . . . . . . . . . . . . . . . . W . . . B2 H2 . M W
    W . . . . . . . W . . . . . . . . . . . . . . . . . . .
↪ w . . . . . . . . . . . . . . . . . . . W . . . . . . . W
    W . . . . . . . W . . . . . . . . . . . . . . . . . . .
↪ w . . . . . . . . . . . . . . . . . . . W . . . . . . . W
    W . . . . . . . W . . . . . . . . . . . . . . . . . . .
↪ w . . . . . . . . . . . . . . . . . . . W . . . . . . . W
    W . . . . . . . W . . . . . . . . . . . . . . . . . . .
↪ w . . . . . . . . . . . . . . . . . . . W . . . . . . . W
    W . . . . . . . W . . . . . . . . . . . . . . . . . . .
↪ w . . . . . . . . . . . . . . . . . . . W . . . . . . . W
    W . . . . . . . W . . . . . . . . . . . . . . . . . . .
↪ w . . . . . . . . . . . . . . . . . . . W . . . . . . . W
    W . . . . . . . W . . . . . . . . . . . . . . . . . . .
↪ w . . . . . . . . . . . . . . . . . . . W . . . . . . . W
    W . . . . . . . W . . . . . . . . . . . . . . . . . . .
↪ w . . . . . . . . . . . . . . . . . . . W . . . . . . . W
    W W W w w W W W W . . . . . . . . . . . . . . . . . . .
↪ . . . . . . . . . . . . . . . . . . . . W W W . . W W W W
    W . . . . . . . . . . . . . . . . . . . . . . . . . . .
↪ . . . . . . . . . . . . . . . . . . . . . . . . . . . . W
    W . . . . . . . . . . . . . . . . . . . . . . . . . . .
↪ . . . . . . . . . . . . . . . . . . . . . . . . . . . . W
    W . . . . . . . . . . . . . . W W W W W W W W W W W W W
↪ w w w w w w w w w w w w w W . . . . . . . . . . . . . . W
    W . . . . . . . . . . . . . . W . . . . . . . . . . . .
↪ . . . . . . . . . . . . . W . . . . . . . . . . . . . . W
    W . . . . . . . . . . . . . . W . . . . . . . . . . . .
↪ . . . . . . . . . . . . . W . . . . . . . . . . . . . . W
    W . . . . . . . . . . . . . . W . . . . . . . . . . . .
↪ . . . . . . . . . . . . . W . . . . . . . . . . . . . . W
    W . . . . . . . . . . . . . . W . . . . . . . . . . . .
↪ . . . . . . . . . . . . . W . . . . . . . . . . . . . . W
    W . . . . . . . . . . . . . . W . . . . . . . . . . . .
↪ . . . . . . . . . . . . . W . . . . . . . . . . . . . . W
    W . . . . . . . . . . . . . . W . . . . . . . . . . . .
↪ . . . . . . . . . . . . . W . . . . . . . . . . . . . . W
    W . . . . . . . . . . . . . . W . . . . . . . . . . . .
↪ . . . . . . . . . . . . . W . . . . . . . . . . . . . . W
    W . . . . . . . . . . . . . . W . . . . . . . . . . . .
↪ . . . . . . . . . . . . . W . . . . . . . . . . . . . . W
    W M . . . . . . . . . . . . . W . . . . . . . . . . . .
↪ . . . . . . . . . . . . . W . . . . . . . . . . . . . . W
    W M M . . . . . . . . . . . . W . . . . . . . . . . . .
↪ M . . . . . . . . . . . . W . . . . . . . . . . . . . . W
    W M M M . . . . . . . . . . . W . . . . . . . . . . . M
↪ M M . . . . . . . . . . . W . . . . . . . . . . . . . . W
    W W W W W W W W W W W . . . . W . . . . . . . . . . M M
↪ M M M . . . . . . . . . . W . . . . W W W W W W W W W W W
    W . . . . . . . . . . . . . . W . . . . . . . . . . M M
↪ M M M . . . . . . . . . . W . . . . . . . . . . . M M M W
```

(continues on next page)

(continued from previous page)

```
W . . . . . . . . . . . . . . W . . . . . . . . . . . M M M . . . . . . . . . . . W . . . . . . . . . . . . M M W
W . . . . . . . . . . . . . . W . . . . . . . . . . . . M . . . . . . . . . . . . W . . . . . . . . . . . . . M W
W . . . . . . . . . . . . . . W . . . . . . . . . . . . . . . . . . . . . . . . . W . . . . . . . . . . . . . . W
W . . . . . . . . . . . . . . W . . . . . . . . . . . . . . . . . . . . . . . . . W . . . . . . . . . . . . . . W
W . . . . . . . . . . . . . . W . . . . . . . . . . . . . . . . . . . . . . . . . W . . . . . . . . . . . . . . W
W . . . . . . . . . . . . . . W . . . . . . . . . . . . . . . . . . . . . . . . . W . . . . . . . . . . . . . . W
W . . . . . . . . . . . . . . W . . . . . . . . . . . . . . . . . . . . . . . . . W . . . . . . . . . . . . . . W
W . . . . . . . . . . . . . . W . . . . . . . . . . . . . . . . . . . . . . . . . W . . . . . . . . . . . . . . W
W . . . . . . . . . . . . . . W . . . . . . . . . . . . . . . . . . . . . . . . . W . . . . . . . . . . . . . . W
W . . . . . . . . . . . . . . W . . . . . . . . . . . . . . . . . . . . . . . . . W . . . . . . . . . . . . . . W
W . . . . . . . . . . . . . . W . . . . . . . . . . . . . . . . . . . . . . . . . W . . . . . . . . . . . . . . W
W . . . . . . . . . . . . . . W w w w w w w w w w w w w w W W W W W W W W W W W W W . . . . . . . . . . . . . . W
W . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . W
W . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . W
W W W W . . W W W . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . W W W w w W W W W
W . . . . . . . W . . . . . . . . . . . . . . . . . . . w . . . . . . . . . . . . . . . . . . . W . . . . . . . W
W . . . . . . . W . . . . . . . . . . . . . . . . . . . w . . . . . . . . . . . . . . . . . . . W . . . . . . . W
W . . . . . . . W . . . . . . . . . . . . . . . . . . . w . . . . . . . . . . . . . . . . . . . W . . . . . . . W
W . . . . . . . W . . . . . . . . . . . . . . . . . . . w . . . . . . . . . . . . . . . . . . . W . . . . . . . W
W . . . . . . . W . . . . . . . . . . . . . . . . . . . w . . . . . . . . . . . . . . . . . . . W . . . . . . . W
W . . . . . . . W . . . . . . . . . . . . . . . . . . . w . . . . . . . . . . . . . . . . . . . W . . . . . . . W
W . . . . . . . W . . . . . . . . . . . . . . . . . . . w . . . . . . . . . . . . . . . . . . . W . . . . . . . W
W . . . . . . . W . . . . . . . . . . . . . . . . . . . w . . . . . . . . . . . . . . . . . . . W . . . . . . . W
W M . H1 B1 . . . W . . . . . . . . . . . . . . . . . . . w . . . . . . . . . . . . . . . . . . . W . . . . . . . W
W M . . H1 . . . W . . . . . . . . . . . . . . . . . . . w . . . . . . . . . . . . . . . . . . . W M . . . . . . W
W M M . . . . . W . . . . . . . . . . . . . . . . . . . w . . . . . . . . . . . . . . . . . . . W M M . . . . . W
```

(continues on next page)

(continued from previous page)

```
      W  M  M  M  M  .  .  .  W  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .␣
↪ W  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  W  M  M  M  .  .  .  .  W
      W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W␣
↪ W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W

Actions:
  - Name: spawn_worker
    InputMapping:
      Internal: true
    Behaviours:
      - Src:
          Object: base
          Commands:
            - spawn: worker
            - set: [ is_busy, 0 ]
        Dst:
          Object: _empty

      - Src:
          Object: base
        Dst:
          Object: [ base, barracks, combat, worker, ranged, movable_wall ]
          Commands:
            # Try to spawn in another location on the next tick
            - exec:
                Action: spawn_worker
                Delay: 1
                Randomize: true
  - Name: spawn_combat
    InputMapping:
      Internal: true
    Behaviours:
      - Src:
          Object: barracks
          Commands:
            - spawn: combat
            - set: [ is_busy, 0 ]
        Dst:
          Object: _empty

      - Src:
          Object: barracks
        Dst:
          Object: [ base, barracks, combat, worker, ranged, movable_wall ]
          Commands:
            # Try to spawn in another location on the next tick
            - exec:
                Action: spawn_combat
                Delay: 1
                Randomize: true

  - Name: construct_barracks
```

(continues on next page)

(continued from previous page)

```
    InputMapping:
      Inputs:
        1:
          Description: Completes construction of a barracks
          VectorToDest: [ 0, 0 ]
      Internal: true
    Behaviours:
      - Src:
          Object: barracks_disabled
          Commands:
            - set: [ is_busy, 0 ]
            - change_to: barracks
        Dst:
          Object: barracks_disabled

  # worker costs 5 resources to build, get a reward when a worker is built
  - Name: build_worker
    InputMapping:
      Inputs:
        1:
          Description: Build
          VectorToDest: [ 0, 0 ]
    Behaviours:
      - Src:
          Object: base
          Preconditions:
            - gte: [ player_resources, 5 ]
            - eq: [ is_busy, 0 ]
          Commands:
            - set: [ is_busy, 1 ]
            - sub: [ player_resources, 5 ]
            - reward: 1
              # Queue a build which will take 10 seconds
            - exec:
                Action: spawn_worker
                Delay: 10
                Randomize: true
                Executor: action
        Dst:
          Object: base

  - Name: build_combat
    InputMapping:
      Inputs:
        1:
          Description: Build
          VectorToDest: [ 0, 0 ]
    Behaviours:
      - Src:
          Object: barracks
          Preconditions:
            - gte: [ player_resources, 5 ]
```

(continues on next page)

(continued from previous page)

```
          - eq: [ is_busy, 0 ]
        Commands:
          - set: [ is_busy, 1 ]
          - sub: [ player_resources, 5 ]
          - reward: 1
          - exec:
              Action: spawn_combat
              Delay: 10
              Randomize: true
              Executor: action
      Dst:
        Object: barracks

  - Name: build_barracks
    Behaviours:
      - Src:
          Object: worker
          Preconditions:
            - gte: [ player_resources, 20 ]
            - eq: [ is_busy, 0 ]
          Commands:
            - sub: [ player_resources, 20 ]
            - reward: 1
            - spawn: barracks_disabled
        Dst:
          Object: _empty

  - Name: gather
    Behaviours:
      - Src:
          Object: worker
          Preconditions:
            - lt: [ resources, 5 ]
            - eq: [ is_busy, 0 ]
          Commands:
            - incr: resources
            - reward: 1
        Dst:
          Object: minerals
          Commands:
            - decr: resources
            - lt:
                Arguments: [resources, 10]
                Commands:
                  - set_tile: 1
            - lt:
                Arguments: [ resources, 5 ]
                Commands:
                  - set_tile: 2
            - eq:
                Arguments: [ resources, 0 ]
                Commands:
```

(continues on next page)

(continued from previous page)

```
                  - remove: true
      - Src:
          Object: worker
          Preconditions:
            - eq: [ is_busy, 0 ]
            - gt: [ resources, 0 ]
            - eq: [ src._playerId, dst._playerId ]
          Commands:
            - decr: resources
            - reward: 1
        Dst:
          Object: base
          Commands:
            - incr: player_resources

  - Name: move
    Behaviours:
      - Src:
          Preconditions:
            - eq: [ is_busy, 0 ]
          Object: [ worker, combat, ranged ]
          Commands:
            - mov: _dest # mov will move the object, _dest is the destination location␣
↪of the action
        Dst:
          Object: _empty

      - Src:
          Object: ranged
          Commands:
            - mov: _dest # mov will move the object, _dest is the destination location␣
↪of the action
        Dst:
          Object: [ movable_wall, worker, combat ]
          Commands:
            - cascade: _dest # reapply the same action to the dest location of the action

  #   Name: ranged_attack
  - Name: attack
    Behaviours:

      - Src:
          Object: worker
          Preconditions:
            - neq: [ src._playerId, dst._playerId ]
            - eq: [ is_busy, 0 ]
          Commands:
            - reward: 1
        Dst:
          Object: [ base, combat, worker, ranged ]
          Commands:
            - sub: [ health, 1 ]
```

(continues on next page)

(continued from previous page)

```
            - lte:
                Arguments: [ health, 0 ]
                Commands:
                  - remove: true

      - Src:
          Object: combat
          Preconditions:
            - neq: [ src._playerId, dst._playerId ]
            - eq: [ is_busy, 0 ]
          Commands:
            - reward: 1
        Dst:
          Object: [ base, combat, worker, ranged, barracks ]
          Commands:
            - sub: [ health, 5 ]
            - lte:
                Arguments: [ health, 0 ]
                Commands:
                  - remove: true

Objects:
  - Name: minerals
    MapCharacter: M
    Variables:
      - Name: resources
        InitialValue: 20
    Observers:
      Sprite2D:
        - Image: oryx/oryx_tiny_galaxy/tg_sliced/tg_items/tg_items_crystal_green.png
          Scale: 1.0
        - Image: oryx/oryx_tiny_galaxy/tg_sliced/tg_items/tg_items_crystal_green.png
          Scale: 0.5
        - Image: oryx/oryx_tiny_galaxy/tg_sliced/tg_items/tg_items_crystal_green.png
          Scale: 0.3
      Block2D:
        - Shape: triangle
          Color: [ 0.0, 1.0, 0.0 ]
          Scale: 1.0
        - Shape: triangle
          Color: [ 0.0, 1.0, 0.0 ]
          Scale: 0.5
        - Shape: triangle
          Color: [ 0.0, 1.0, 0.0 ]
          Scale: 0.1
      Isometric:
        - Image: oryx/oryx_iso_dungeon/minerals-1-0.png
        - Image: oryx/oryx_iso_dungeon/minerals-1-1.png
        - Image: oryx/oryx_iso_dungeon/minerals-1-2.png

  - Name: worker
    MapCharacter: H
```

(continues on next page)

(continued from previous page)

```
    Variables:
      - Name: resources
        InitialValue: 0
      - Name: health
        InitialValue: 10
      - Name: is_busy
        InitialValue: 0
    Observers:
      Sprite2D:
        - Image: oryx/oryx_tiny_galaxy/tg_sliced/tg_monsters/tg_monsters_jelly_d1.png
      Block2D:
        - Shape: square
          Color: [ 0.6, 0.2, 0.2 ]
          Scale: 0.5
      Isometric:
        - Image: oryx/oryx_iso_dungeon/jelly-1.png

  - Name: ranged
    MapCharacter: r
    Variables:
      - Name: health
        InitialValue: 20
      - Name: is_busy
        InitialValue: 0
    Observers:
      Sprite2D:
        - Image: oryx/oryx_tiny_galaxy/tg_sliced/tg_monsters/tg_monsters_crawler_queen_
↪d1.png
      Block2D:
        - Shape: square
          Color: [ 0.2, 0.2, 0.6 ]
          Scale: 1.0
      Isometric:
        - Image: oryx/oryx_iso_dungeon/queen-1.png

  - Name: combat
    MapCharacter: c
    Variables:
      - Name: health
        InitialValue: 30
      - Name: is_busy
        InitialValue: 0
    Observers:
      Sprite2D:
        - Image: oryx/oryx_tiny_galaxy/tg_sliced/tg_monsters/tg_monsters_beast_d1.png
      Block2D:
        - Color: [ 0.2, 0.6, 0.6 ]
          Shape: square
          Scale: 0.8
      Isometric:
        - Image: oryx/oryx_iso_dungeon/beast-1.png
```

(continues on next page)

(continued from previous page)

```
- Name: fixed_wall
  MapCharacter: W
  Observers:
    Sprite2D:
      - TilingMode: WALL_2 # Will tile walls with two images
        Image:
          - oryx/oryx_tiny_galaxy/tg_sliced/tg_world_fixed/img33.png
          - oryx/oryx_tiny_galaxy/tg_sliced/tg_world_fixed/img40.png
    Block2D:
      - Color: [ 0.5, 0.5, 0.5 ]
        Shape: square
    Isometric:
      - Image: oryx/oryx_iso_dungeon/wall-grey-1.png

- Name: movable_wall
  MapCharacter: w
  Observers:
    Sprite2D:
      - Image: oryx/oryx_tiny_galaxy/tg_sliced/tg_world_fixed/img282.png
    Block2D:
      - Color: [ 0.8, 0.8, 0.8 ]
        Shape: square
    Isometric:
      - Image: oryx/oryx_iso_dungeon/crate-1.png

- Name: base
  MapCharacter: A
  Variables:
    - Name: health
      InitialValue: 50
    - Name: is_busy
      InitialValue: 0
  Observers:
    Sprite2D:
      - Image: oryx/oryx_tiny_galaxy/tg_sliced/tg_world_fixed/img324.png
    Block2D:
      - Color: [ 0.8, 0.8, 0.3 ]
        Shape: triangle
    Isometric:
      - Image: oryx/oryx_iso_dungeon/base-1.png

- Name: barracks_disabled
  MapCharacter: b
  InitialActions:
    - Action: construct_barracks
      Delay: 20
  Variables:
    - Name: health
      InitialValue: 20
    - Name: is_busy
      InitialValue: 1
  Observers:
```

(continues on next page)

(continued from previous page)

```
      Sprite2D:
        - Image: oryx/oryx_tiny_galaxy/tg_sliced/tg_world_fixed/img280.png
      Block2D:
        - Color: [ 0.3, 0.3, 0.3 ]
          Shape: triangle
          Size: 0.5
      Isometric:
        - Image: oryx/oryx_iso_dungeon/barracks-disabled-1.png

  - Name: barracks
    MapCharacter: B
    Variables:
      - Name: health
        InitialValue: 40
      - Name: is_busy
        InitialValue: 0
    Observers:
      Sprite2D:
        - Image: oryx/oryx_tiny_galaxy/tg_sliced/tg_world_fixed/img320.png
      Block2D:
        - Color: [ 0.8, 0.3, 0.8 ]
          Shape: triangle
      Isometric:
        - Image: oryx/oryx_iso_dungeon/barracks-1.png
```

14.2 Push Mania

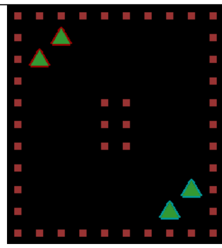
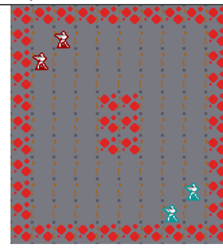
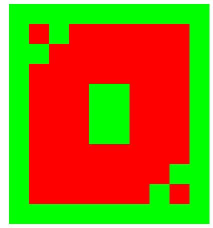
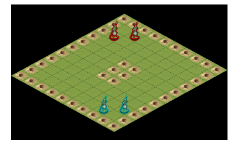
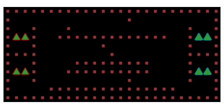
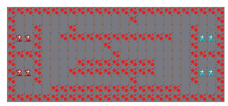
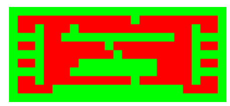
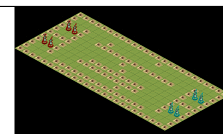
```
RTS/Stratega/push-mania.yaml
```

14.2.1 Description

Game environment ported from https://github.com/GAIGResearch/Stratega. You must push all your opponents pieces into the holes.

14.2.2 Levels

Table 3: Levels

	Block2D	Sprite2D	Vector	Isometric
Level ID: 0 Size: 10x11				
Level ID: 1 Size: 25x11				

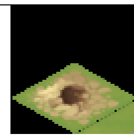
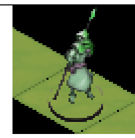
14.2.3 Code Example

The most basic way to create a Griddly Gym Environment. Defaults to level 0 and SPRITE_2D rendering.

```
import gym
import griddly
from griddly.util.wrappers import InvalidMaskingRTSWrapper

if __name__ == '__main__':

    env = gym.make('GDY-Push-Mania-v0')
    env.reset()
    env = InvalidMaskingRTSWrapper(env)

    # Replace with your own control algorithm!
    for s in range(1000):
        obs, reward, done, info = env.step(env.action_space.sample())
        for p in range(env.player_count):
            env.render(observer=p) # Renders the environment from the perspective of a single player

        env.render(observer='global') # Renders the entire environment

        if done:
            env.reset()
```

14.2.4 Objects

Table 4: Tiles

Name ->	hole	pusher
Map Char ->	*H*	*p*
Block2D		
Sprite2D		
Vector		
Isometric		

14.2.5 Actions

drain_health

Internal This action can only be called from other actions, not by the player.

Action Id	Mapping
1	Reduce the health

push

Action Id	Mapping
1	Left
2	Up
3	Right
4	Down

move

Action Id	Mapping
1	Left
2	Up
3	Right
4	Down

14.2.6 YAML

```
Version: "0.1"
Environment:
  Name: Push Mania
  Description: |
    Game environment ported from https://github.com/GAIGResearch/Stratega.
    You must push all your opponents pieces into the holes.
  Observers:
    Sprite2D:
      TileSize: 32
      BackgroundTile: oryx/oryx_tiny_galaxy/tg_sliced/tg_world_fixed/img23.png
    Isometric:
      TileSize: [64, 64]
      BackgroundTile: stratega/plain.png
      IsoTileHeight: 35
      IsoTileDepth: 0
    Vector:
      IncludePlayerId: true
      IncludeVariables: true
  Variables:
    - Name: unit_count
      InitialValue: 0
  Player:
    Count: 2
  Termination:
    Lose:
      - eq: [pusher:count, 0] # Player loses its king, it loses the game
  Levels:
    - |
      H  H  H  H  H  H  H  H  H  H
      H  .  p1 .  .  .  .  .  .  H
      H  p1 .  .  .  .  .  .  .  H
      H  .  .  .  .  .  .  .  .  H
      H  .  .  .  H  H  .  .  .  H
      H  .  .  .  H  H  .  .  .  H
      H  .  .  .  H  H  .  .  .  H
      H  .  .  .  .  .  .  .  .  H
      H  .  .  .  .  .  .  .  p2 H
      H  .  .  .  .  .  .  p2 .  H
      H  H  H  H  H  H  H  H  H  H
    - |
      H  H  H  H  H  H  H  H  H  H  H  H  H  H  H  H  H  H  H  H  H  H  H  H  H
      H  .  .  .  .  .  .  .  .  .  .  .  .  .  H  .  .  .  .  .  .  .  .  .  H
      H  .  .  H  .  .  .  H  .  .  .  .  .  .  .  .  .  .  .  .  .  H  .  .  H
      H  p1 p1 H  .  .  H  H  H  H  H  H  H  H  H  H  H  H  H  .  .  H  p2 p2 H
      H  .  .  H  .  .  .  .  .  .  .  H  .  .  .  .  .  .  .  .  .  H  .  .  H
      H  H  H  H  .  .  .  .  .  .  .  .  H  .  .  .  .  .  .  .  .  H  H  H  H
      H  .  .  H  .  .  .  .  H  H  H  H  H  H  H  H  H  .  .  .  .  H  .  .  H
      H  p1 p1 H  .  .  .  H  H  H  H  H  H  H  H  H  H  .  .  .  .  H  p2 p2 H
      H  .  .  H  .  .  .  .  .  .  .  .  .  .  H  .  .  .  .  .  .  H  .  .  H
      H  .  .  .  .  H  H  H  H  H  H  H  H  H  H  H  H  H  H  H  .  .  .  .  H
      H  H  H  H  H  H  H  H  H  H  H  H  H  H  H  H  H  H  H  H  H  H  H  H  H
```

(continues on next page)

(continued from previous page)

```
Actions:

  # Reduce all units health by an amount every 10 turns
  - Name: drain_health
    InputMapping:
      Internal: true
      Inputs:
        1:
          Description: "Reduce the health"
    Behaviours:
      - Src:
          Object: pusher
          Commands:
            - sub: [health, 10]
            # if the health is 0 then remove the player
            - exec:
                Action: drain_health
                ActionId: 1
                Delay: 50
            - lt:
                Arguments: [health, 1]
                Commands:
                  - remove: true
                  - reward: -1

        Dst:
          Object: pusher

  - Name: move
    Behaviours:
      # Healer and warrior can move in empty space
      - Src:
          Object: pusher
          Commands:
            - mov: _dest
        Dst:
          Object: _empty

      # Healer and warrior can fall into holes
      - Src:
          Object: pusher
          Commands:
            - remove: true
            - reward: -1
        Dst:
          Object: hole

  - Name: push
    Behaviours:
      # Pushers can push other pushers
      - Src:
```

(continues on next page)

(continued from previous page)

```
          Object: pusher
          Commands:
            - mov: _dest
        Dst:
          Object: pusher
          Commands:
            - cascade: _dest

Objects:

  - Name: hole
    MapCharacter: H
    Observers:
      Sprite2D:
        - Image: oryx/oryx_tiny_galaxy/tg_sliced/tg_world_fixed/img343.png
      Block2D:
        - Shape: square
          Color: [0.6, 0.2, 0.2]
          Scale: 0.5
      Isometric:
        - Image: stratega/hole.png

  - Name: pusher
    MapCharacter: p
    Variables:
      - Name: health
        InitialValue: 150
    InitialActions:
      - Action: drain_health
        ActionId: 1
        Delay: 50
    Observers:
      Sprite2D:
        - Image: oryx/oryx_tiny_galaxy/tg_sliced/tg_monsters/tg_monsters_astronaut_l1.png
      Block2D:
        - Shape: triangle
          Color: [0.2, 0.6, 0.2]
          Scale: 1.0
      Isometric:
        - Image: stratega/healer.png
```

14.3 Kill The King

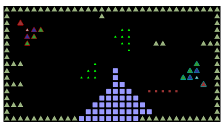
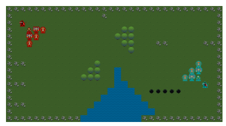
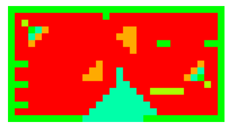
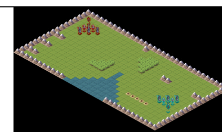
```
RTS/Stratega/kill-the-king.yaml
```

14.3.1 Description

Game environment ported from https://github.com/GAIGResearch/Stratega. Both you and your opponent must protect the king from being killed.

14.3.2 Levels

Table 5: Levels

	Block2D	Sprite2D	Vector	Isometric
Level ID: 0 Size: 32x17				

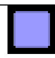
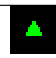
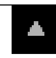
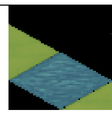
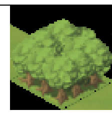
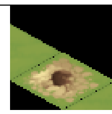
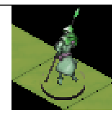
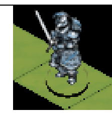
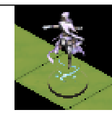
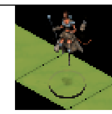
14.3.3 Code Example

The most basic way to create a Griddly Gym Environment. Defaults to level 0 and SPRITE_2D rendering.

```
import gym
import griddly
from griddly.util.wrappers import InvalidMaskingRTSWrapper

if __name__ == '__main__':

    env = gym.make('GDY-Kill-The-King-v0')
    env.reset()
    env = InvalidMaskingRTSWrapper(env)

    # Replace with your own control algorithm!
    for s in range(1000):
        obs, reward, done, info = env.step(env.action_space.sample())
        for p in range(env.player_count):
            env.render(observer=p) # Renders the environment from the perspective of a
↪single player

        env.render(observer='global') # Renders the entire environment

        if done:
            env.reset()
```

14.3.4 Objects

Table 6: Tiles

Name ->	moun-tain	water	forest	hole	healer	warrior	archer	king
Map Char ->	*M*	*W*	*F*	*H*	*h*	*w*	*a*	*k*
Block2D								
Sprite2D								
Vector								
Isometric								

14.3.5 Actions

warrior_attack

Action Id	Mapping
1	Left
2	Up
3	Right
4	Down

archer_attack

Action Id	Mapping
1	Left
2	Up
3	Right
4	Down

heal

Action Id	Mapping
1	Left
2	Up
3	Right
4	Down

move_2

Action Id	Mapping
1	
2	
3	
4	
5	
6	
7	
8	

move_1

Action Id	Mapping
1	Left
2	Up
3	Right
4	Down

14.3.6 YAML

```yaml
Version: "0.1"
Environment:
  Name: Kill The King
  Description: |
    Game environment ported from https://github.com/GAIGResearch/Stratega.
    Both you and your opponent must protect the king from being killed.
  Observers:
    Sprite2D:
      TileSize: 16
      BackgroundTile: oryx/oryx_tiny_galaxy/tg_sliced/tg_world_fixed/img125.png
    Isometric:
      TileSize: [64, 64]
      BackgroundTile: stratega/plain.png
      IsoTileHeight: 35
      IsoTileDepth: 0
    Vector:
      IncludePlayerId: true
      IncludeVariables: true
  Player:
    Count: 2
  Termination:
    Lose:
      - eq: [king:count, 0] # Player loses its king, it loses the game
  Levels:
    - |
      M  M  M  M  M  M  M  M  M  M  M  M  M  M  M  M  M  M  M  M  M  M  M  M  M  M  M  M  M  M  M  M
```

(continues on next page)

(continued from previous page)

```
      M  .  .  .  .  .  .  .  .  .  .  .  .  .  M  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  M
      M  .  k1 .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  M
      M  .  .  h1 a1 w1 .  .  .  .  .  .  .  .  .  .  .  F  F  .  .  .  .  .  .  .  .  .  .  .  .  M
      M  .  .  a1 w1 .  .  .  .  .  .  .  .  .  .  .  F  F  F  .  .  .  .  .  .  .  .  .  .  .  .  M
      M  .  .  w1 .  .  .  .  .  .  .  .  .  .  .  .  .  F  F  .  .  .  M  M  .  .  .  .  .  M  M  M
      M  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  F  .  .  .  .  .  .  .  .  .  .  .  .  M
      M  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  M
      M  M  M  .  .  .  .  .  .  .  .  .  .  F  .  .  .  .  .  .  .  .  .  .  .  .  .  .  w2 .  .  M
      M  .  .  .  .  .  .  .  .  .  .  .  F  F  .  .  W  .  .  .  .  .  .  .  .  .  .  w2 a2 .  .  M
      M  .  .  .  .  .  .  .  .  .  .  F  F  F  .  .  W  .  .  .  .  .  .  .  .  .  w2 a2 h2 .  .  M
      M  M  M  .  .  .  .  .  .  .  .  .  .  .  .  .  W  W  .  .  .  .  .  .  .  .  .  .  .  k2 .  M
      M  .  .  .  .  .  .  .  .  .  .  .  .  .  .  W  W  W  W  .  .  H  H  H  H  H  .  .  .  .  .  M
      M  .  .  .  .  .  .  .  .  .  .  .  .  .  W  W  W  W  W  W  .  .  .  .  .  .  .  .  .  .  .  M
      M  M  M  .  .  .  .  .  .  .  .  .  .  W  W  W  W  W  W  W  W  .  .  .  .  .  .  .  .  .  .  M
      M  .  .  .  .  .  .  .  .  .  .  .  W  W  W  W  W  W  W  W  W  W  .  .  .  .  .  .  .  .  .  M
      M  M  M  M  M  M  M  M  M  M  M  W  W  W  W  W  W  W  W  W  M  M  M  M  M  M  M  M  M  M  M  M

Actions:

  - Name: move_2
    InputMapping:
        Inputs:
          1:
            VectorToDest: [0, 1]
          2:
            VectorToDest: [0, 2]
          3:
            VectorToDest: [0, -1]
          4:
            VectorToDest: [0, -2]
          5:
            VectorToDest: [1, 0]
          6:
            VectorToDest: [2, 0]
          7:
            VectorToDest: [-1, 0]
```

(continues on next page)

(continued from previous page)

```
        8:
          VectorToDest: [-2, 0]
    Behaviours:
      # Healer and warrior can move in empty space
      - Src:
          Object: [warrior, archer, healer]
          Commands:
            - mov: _dest
        Dst:
          Object: _empty

      # Healer and warrior can fall into holes
      - Src:
          Object: [warrior, archer, healer]
          Commands:
            - remove: true
        Dst:
          Object: hole

  - Name: move_1
    Behaviours:
      # Healer and warrior can move in empty space
      - Src:
          Object: king
          Commands:
            - mov: _dest
        Dst:
          Object: _empty

      # Healer and warrior can fall into holes
      - Src:
          Object: king
          Commands:
            - remove: true
        Dst:
          Object: hole

  - Name: heal
    Behaviours:
      # Healer can heal adjacent warriors and other healers
      - Src:
          # Can only heal units on your own team
          Preconditions:
            - eq: [src._playerId, dst._playerId]
          Object: healer
        Dst:
          Object: [healer, warrior, king]
          Commands:
            - add: [health, 10]

  - Name: warrior_attack
    Behaviours:
```

(continues on next page)

(continued from previous page)

```
      # Warrior can damage adjacent warriors and healers
      - Src:
          # Can only attack units of different players
          Preconditions:
            - neq: [src._playerId, dst._playerId]
          Object: warrior
        Dst:
          Object: [healer, warrior]
          Commands:
            - sub: [health, 25]

  - Name: archer_attack
    Behaviours:
      # Warrior can damage adjacent warriors and healers
      - Src:
          # Can only attack units of different players
          Preconditions:
            - neq: [src._playerId, dst._playerId]
          Object: warrior
        Dst:
          Object: [healer, warrior]
          Commands:
            - sub: [health, 25]

Objects:
  - Name: mountain
    MapCharacter: M
    Observers:
      Sprite2D:
        - Image: oryx/oryx_tiny_galaxy/tg_sliced/tg_world_fixed/img355.png
      Block2D:
        - Shape: triangle
          Color: [0.6, 0.7, 0.5]
          Scale: 1.0
      Isometric:
        - Image: stratega/rock.png

  - Name: water
    MapCharacter: W
    Observers:
      Sprite2D:
        - Image: oryx/oryx_tiny_galaxy/tg_sliced/tg_world_fixed/img185.png
      Block2D:
        - Shape: square
          Color: [0.6, 0.6, 1.0]
          Scale: 1.0
      Isometric:
        - Image: stratega/water.png

  - Name: forest
    MapCharacter: F
    Observers:
```

(continues on next page)

(continued from previous page)

```
      Sprite2D:
        - Image: oryx/oryx_tiny_galaxy/tg_sliced/tg_world_fixed/img332.png
      Block2D:
        - Shape: triangle
          Color: [0.0, 7.0, 0.0]
          Scale: 0.5
      Isometric:
        - Image: stratega/forest.png

  - Name: hole
    MapCharacter: H
    Observers:
      Sprite2D:
        - Image: oryx/oryx_tiny_galaxy/tg_sliced/tg_world_fixed/img129.png
      Block2D:
        - Shape: square
          Color: [0.6, 0.2, 0.2]
          Scale: 0.5
      Isometric:
        - Image: stratega/hole.png

  - Name: healer
    MapCharacter: h
    Variables:
      - Name: health
        InitialValue: 40
    Observers:
      Sprite2D:
        - Image: oryx/oryx_tiny_galaxy/tg_sliced/tg_monsters/tg_monsters_civilian_m_l1.
↪png
      Block2D:
        - Shape: triangle
          Color: [0.7, 0.7, 0.7]
          Scale: 0.5
      Isometric:
        - Image: stratega/healer.png

  - Name: warrior
    MapCharacter: w
    Variables:
      - Name: health
        InitialValue: 200
    Observers:
      Sprite2D:
        - Image: oryx/oryx_tiny_galaxy/tg_sliced/tg_monsters/tg_monsters_beast_d1.png
      Block2D:
        - Color: [0.2, 0.6, 0.2]
          Shape: triangle
          Scale: 0.9
      Isometric:
        - Image: stratega/basicCloseRange.png
```

(continues on next page)

(continued from previous page)

```yaml
  - Name: archer
    MapCharacter: a
    Variables:
      - Name: health
        InitialValue: 100
    Observers:
      Sprite2D:
        - Image: oryx/oryx_tiny_galaxy/tg_sliced/tg_monsters/tg_monsters_drone_d1.png
      Block2D:
        - Color: [0.2, 0.2, 0.6]
          Shape: triangle
          Scale: 0.9
      Isometric:
        - Image: stratega/basicLongRange.png

  - Name: king
    MapCharacter: k
    Variables:
      - Name: health
        InitialValue: 400
    Observers:
      Sprite2D:
        - Image: oryx/oryx_tiny_galaxy/tg_sliced/tg_monsters/tg_monsters_lord_l1.png
      Block2D:
        - Color: [0.6, 0.2, 0.2]
          Shape: triangle
          Scale: 1.0
      Isometric:
        - Image: stratega/advancedCloseRange.png
```

14.4 Heal Or Die

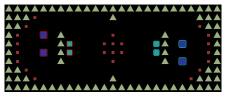
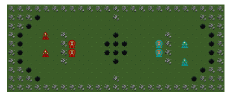
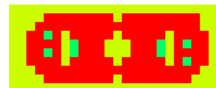
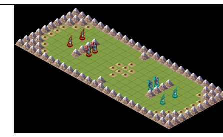
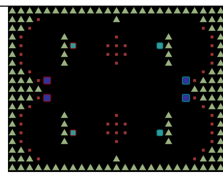
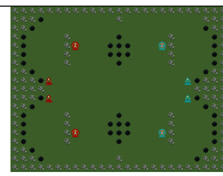
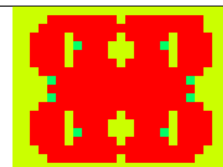
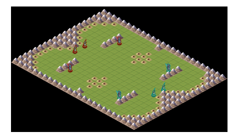
```
RTS/Stratega/heal-or-die.yaml
```

14.4.1 Description

Game environment ported from https://github.com/GAIGResearch/Stratega. You have units that heal and units that perform close combat. Additionally, on every turn, the health of your units decreases. Win the game by killing your opponents pieces first.

14.4.2 Levels

Table 7: Levels

<table>
<tr><th></th><th>Block2D</th><th>Sprite2D</th><th>Vector</th><th>Isometric</th></tr>
<tr><td><table><tr><td>Level ID</td><td>0</td></tr><tr><td>Size</td><td>25x10</td></tr></table></td><td></td><td></td><td></td><td></td></tr>
<tr><td><table><tr><td>Level ID</td><td>1</td></tr><tr><td>Size</td><td>25x19</td></tr></table></td><td></td><td></td><td></td><td></td></tr>
</table>

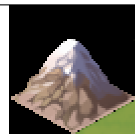
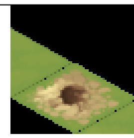
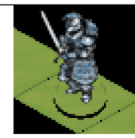
14.4.3 Code Example

The most basic way to create a Griddly Gym Environment. Defaults to level 0 and SPRITE_2D rendering.

```
import gym
import griddly
from griddly.util.wrappers import InvalidMaskingRTSWrapper

if __name__ == '__main__':

    env = gym.make('GDY-Heal-Or-Die-v0')
    env.reset()
    env = InvalidMaskingRTSWrapper(env)

    # Replace with your own control algorithm!
    for s in range(1000):
        obs, reward, done, info = env.step(env.action_space.sample())
        for p in range(env.player_count):
            env.render(observer=p) # Renders the environment from the perspective of a single player

        env.render(observer='global') # Renders the entire environment

        if done:
            env.reset()
```

14.4.4 Objects

Table 8: Tiles

Name ->	mountain	hole	healer	warrior
Map Char ->	*M*	*H*	*h*	*w*
Block2D				
Sprite2D				
Vector				
Isometric				

14.4.5 Actions

attack

Action Id	Mapping
1	Left
2	Up
3	Right
4	Down

heal

Action Id	Mapping
1	Left
2	Up
3	Right
4	Down

move

Action Id	Mapping
1	Left
2	Up
3	Right
4	Down

unit_counter

Internal This action can only be called from other actions, not by the player.

Action Id	Mapping
1	The only action here is to increment the unit count

drain_health

Internal This action can only be called from other actions, not by the player.

Action Id	Mapping
1	Reduce the health

14.4.6 YAML


```yaml
Version: "0.1"
Environment:
  Name: Heal Or Die
  Description: |
    Game environment ported from https://github.com/GAIGResearch/Stratega.
    You have units that heal and units that perform close combat.
    Additionally, on every turn, the health of your units decreases. Win the game by␣
↪killing your opponents pieces first.
  Observers:
    Sprite2D:
      TileSize: 16
      BackgroundTile: oryx/oryx_tiny_galaxy/tg_sliced/tg_world_fixed/img125.png
    Isometric:
      TileSize: [64, 64]
      BackgroundTile: stratega/plain.png
      IsoTileHeight: 35
      IsoTileDepth: 0
    Vector:
      IncludePlayerId: true
      IncludeVariables: true
  Variables:
    - Name: unit_count
      InitialValue: 0
      PerPlayer: true
  Player:
    Count: 2
  Termination:
    Lose:
      - eq: [unit_count, 0] # If the player has no bases
  Levels:
    - |
      M  M  M  M  M  M  M  M  M  M  M  M  M  M  M  M  M  M  M  M  M  M  M  M  M  M  M  M  M  M  M  M  M  M  M  M  M  M  M  M  M  M  M  M  M  M  M  M  M
      M  M  M  H  .  .  .  .  .  .  .  .  M  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  M  M  M
      M  M  H  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  H  M  M
```


(continues on next page)

(continued from previous page)

```
      M  H  .  .  h1 .  M  .  .  .  .  .  H  .  .  .  .  .  M  .  .  .  .  H  M
      M  H  .  .  .  .  M  w1 .  .  .  H  H  H  .  .  .  w2 M  .  h2 .  .  H  M
      M  H  .  .  h1 .  M  w1 .  .  .  H  H  H  .  .  .  w2 M  .  .  .  .  H  M
      M  H  .  .  .  .  M  .  .  .  .  .  H  .  .  .  .  .  M  .  h2 .  .  H  M
      M  M  H  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  H  M  M
      M  M  M  H  .  .  .  .  .  .  .  .  M  .  .  .  .  .  .  .  .  H  M  M  M
      M  M  M  M  M  M  M  M  M  M  M  M  M  M  M  M  M  M  M  M  M  M  M  M  M
    - |
      M  M  M  M  M  M  M  M  M  M  M  M  M  M  M  M  M  M  M  M  M  M  M  M  M
      M  M  M  H  .  .  .  .  .  .  .  .  M  .  .  .  .  .  .  .  .  .  M  M  M
      M  M  H  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  H  M  M
      M  H  .  .  .  .  M  .  .  .  .  .  H  .  .  .  .  .  M  .  .  .  .  H  M
      M  H  .  .  .  .  M  w1 .  .  .  H  H  H  .  .  .  w2 M  .  .  .  .  H  M
      M  H  .  .  .  .  M  .  .  .  .  H  H  H  .  .  .  .  M  .  .  .  .  H  M
      M  H  .  .  .  .  M  .  .  .  .  .  H  .  .  .  .  .  M  .  .  .  .  H  M
      M  M  H  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  H  M  M
      M  M  M  H  h1 .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  h2 H  M  M  M
      M  M  M  M  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  M  M  M  M
      M  M  M  H  h1 .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  h2 H  M  M  M
      M  M  H  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  H  M  M
      M  H  .  .  .  .  M  .  .  .  .  .  H  .  .  .  .  .  M  .  .  .  .  H  M
      M  H  .  .  .  .  M  .  .  .  .  H  H  H  .  .  .  .  M  .  .  .  .  H  M
      M  H  .  .  .  .  M  w1 .  .  .  H  H  H  .  .  .  w2 M  .  .  .  .  H  M
      M  H  .  .  .  .  M  .  .  .  .  .  H  .  .  .  .  .  M  .  .  .  .  H  M
      M  M  H  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  H  M  M
      M  M  M  H  .  .  .  .  .  .  .  .  M  .  .  .  .  .  .  .  .  H  M  M  M
      M  M  M  M  M  M  M  M  M  M  M  M  M  M  M  M  M  M  M  M  M  M  M  M  M

Actions:
  # Just a counter for the number of units per player
  - Name: unit_counter
    InputMapping:
      Internal: true
      Inputs:
        1:
          Description: "The only action here is to increment the unit count"
    Behaviours:
      - Src:
          Object: [healer, warrior]
          Commands:
            - incr: unit_count
        Dst:
          Object: [healer, warrior]

  # Reduce all units health by an amount every 10 turns
  - Name: drain_health
    InputMapping:
      Internal: true
      Inputs:
        1:
          Description: "Reduce the health"
    Behaviours:
```

(continues on next page)

(continued from previous page)

```
      - Src:
          Object: [healer, warrior]
          Commands:
            - sub: [health, 25]
            # if the health is 0 then remove the player
            - exec:
                Action: drain_health
                ActionId: 1
                Delay: 50
            - lt:
                Arguments: [health, 1]
                Commands:
                  - remove: true
                  - decr: unit_count
        Dst:
          Object: [healer, warrior]

  - Name: move
    Behaviours:
      # Healer and warrior can move in empty space
      - Src:
          Object: [healer, warrior]
          Commands:
            - mov: _dest
        Dst:
          Object: _empty

      # Healer and warrior can fall into holes
      - Src:
          Object: [healer, warrior]
          Commands:
            - remove: true
            - decr: unit_count
        Dst:
          Object: hole

  - Name: heal
    Behaviours:
      # Healer can heal adjacent warriors and other healers
      - Src:
          # Can only heal units on your own team
          Preconditions:
            - eq: [src._playerId, dst._playerId]
          Object: healer
        Dst:
          Object: [healer, warrior]
          Commands:
            - add: [health, 100]

  - Name: attack
    Behaviours:
      # Warrior can damage adjacent warriors and healers
```

(continues on next page)

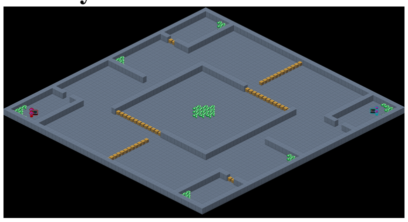
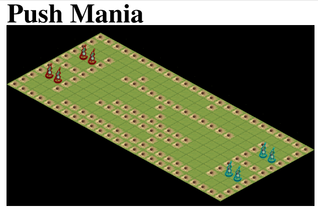
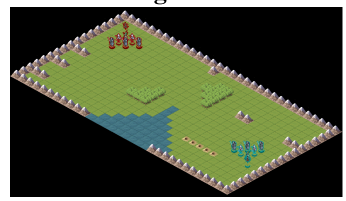
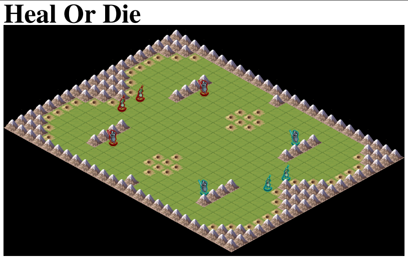
(continued from previous page)

```
      - Src:
          # Can only attack units of different players
          Preconditions:
            - neq: [src._playerId, dst._playerId]
          Object: warrior
        Dst:
          Object: [healer, warrior]
          Commands:
            - sub: [health, 25]
            - lt:
                Arguments: [health, 1]
                Commands:
                  - remove: true
                  - decr: unit_count

Objects:
  - Name: mountain
    MapCharacter: M
    Observers:
      Sprite2D:
        - Image: oryx/oryx_tiny_galaxy/tg_sliced/tg_world_fixed/img355.png
      Block2D:
        - Shape: triangle
          Color: [0.6, 0.7, 0.5]
          Scale: 1.0
      Isometric:
        - Image: stratega/rock.png

  - Name: hole
    MapCharacter: H
    Observers:
      Sprite2D:
        - Image: oryx/oryx_tiny_galaxy/tg_sliced/tg_world_fixed/img129.png
      Block2D:
        - Shape: square
          Color: [0.6, 0.2, 0.2]
          Scale: 0.5
      Isometric:
        - Image: stratega/hole.png

  - Name: healer
    MapCharacter: h
    Variables:
      - Name: health
        InitialValue: 150
    InitialActions:
      - Action: drain_health
        ActionId: 1
        Delay: 50
      - Action: unit_counter
        ActionId: 1
    Observers:
```

(continues on next page)

(continued from previous page)

```
      Sprite2D:
        - Image: oryx/oryx_tiny_galaxy/tg_sliced/tg_monsters/tg_monsters_civilian_m_l1.
↪png
      Block2D:
        - Shape: square
          Color: [0.2, 0.2, 0.6]
          Scale: 1.0
      Isometric:
        - Image: stratega/healer.png

  - Name: warrior
    MapCharacter: w
    Variables:
      - Name: health
        InitialValue: 200
    InitialActions:
      - Action: drain_health
        ActionId: 1
        Delay: 50
      - Action: unit_counter
        ActionId: 1
    Observers:
      Sprite2D:
        - Image: oryx/oryx_tiny_galaxy/tg_sliced/tg_monsters/tg_monsters_beast_d1.png
      Block2D:
        - Color: [0.2, 0.6, 0.6]
          Shape: square
          Scale: 0.8
      Isometric:
        - Image: stratega/basicCloseRange.png
```

GriddlyRTS  An RTS Game. There's aliens and stuff.	**Push Mania**  Game environment ported from https://github.com/GAIGResearch/Stratega. You must push all your opponents pieces into the holes.	**Kill The King**  Game environment ported from https://github.com/GAIGResearch/Stratega. Both you and your opponent must protect the king from being killed.
Heal Or Die  Game environment ported from https://github.com/GAIGResearch/Stratega. You have units that heal and units that perform close combat. Additionally, on every turn, the health of your units decreases. Win the game by killing your opponents pieces first.		

CHAPTER

FIFTEEN

MULTI-AGENT

15.1 Robot Tag 12v12

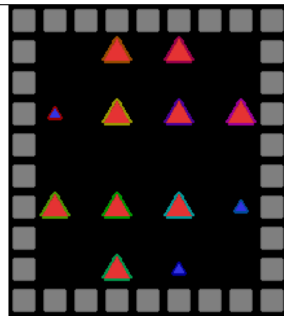
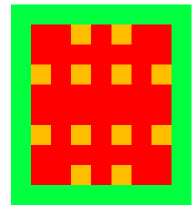
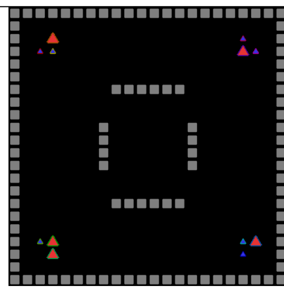
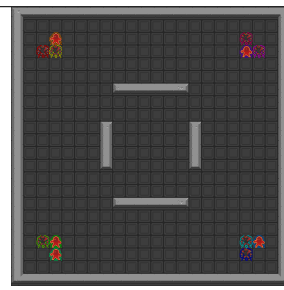
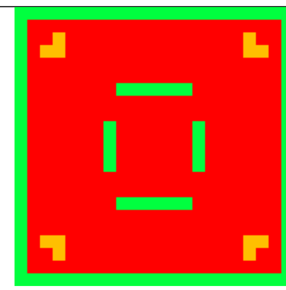
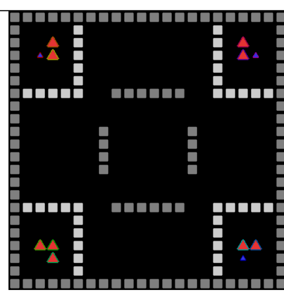
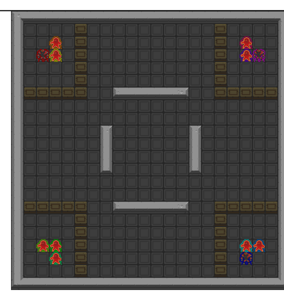
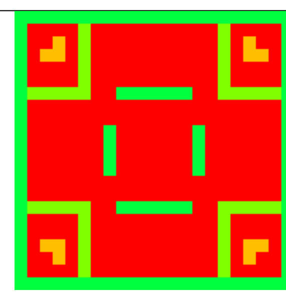
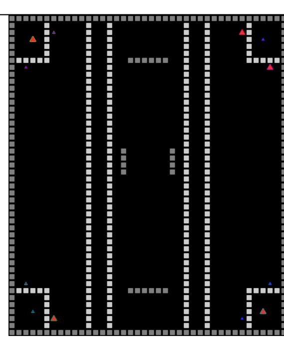
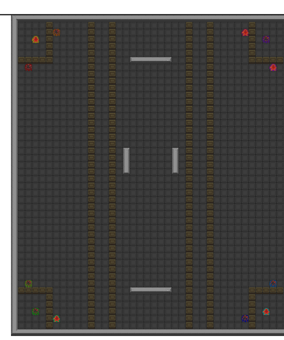
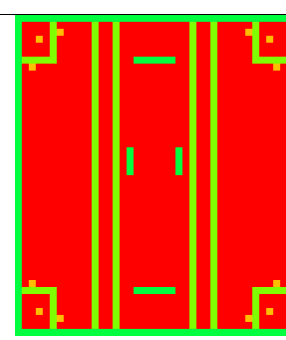
```
Multi-Agent/robot_tag_12.yaml
```

15.1.1 Description

Robots start randomly as “tagged” or not, robots can “tag” other robots. Any robot that is “tagged” 3 times dies.

15.1.2 Levels

Table 1: Levels

	Block2D	Sprite2D	Vector
Level ID 0 Size 9x10			
Level ID 1 Size 22x22			
Level ID 2 Size 22x22			
Level ID 3 Size 40x46			

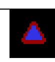
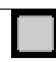
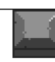
15.1.3 Code Example

The most basic way to create a Griddly Gym Environment. Defaults to level 0 and SPRITE_2D rendering.

```
import gym
import griddly

if __name__ == '__main__':

    env = gym.make('GDY-Robot-Tag-12v12-v0')
    env.reset()

    # Replace with your own control algorithm!
    for s in range(1000):
        obs, reward, done, info = env.step(env.action_space.sample())
        for p in range(env.player_count):
            env.render(observer=p) # Renders the environment from the perspective of a single player

        env.render(observer='global') # Renders the entire environment

        if done:
            env.reset()
```

15.1.4 Objects

Table 2: Tiles

Name ->	tagger	moveable_wall	fixed_wall
Map Char ->	*f*	*m*	*W*
Block2D			
Sprite2D			
Vector			

15.1.5 Actions

move

Action Id	Mapping
1	Left
2	Up
3	Right
4	Down

initialize_is_tagged

Internal This action can only be called from other actions, not by the player.

Action Id	Mapping
1	Initialize Tagged
2	Initialize Not Tagged

tag

Action Id	Mapping
1	Left
2	Up
3	Right
4	Down

15.1.6 YAML

```yaml
Version: "0.1"
Environment:
  Name: Robot Tag 12v12
  Description: Robots start randomly as "tagged" or not, robots can "tag" other robots.␣
↪Any robot that is "tagged" 3 times dies.
  Observers:
    Block2D:
      TileSize: 24
    Sprite2D:
      TileSize: 24
      BackgroundTile: oryx/oryx_fantasy/floor1-1.png
    Vector:
      IncludePlayerId: true
      IncludeVariables: true
  Variables:
    - Name: player_done
      InitialValue: 0
      PerPlayer: true
    - Name: tagged_count
      InitialValue: 0
  Player:
    Count: 12
    Observer:
      RotateWithAvatar: true
      TrackAvatar: true
      Height: 9
      Width: 9
      OffsetX: 0
      OffsetY: 0
    AvatarObject: tagger
  Termination:
```

(continues on next page)

(continued from previous page)

```
    End:
      - eq: [ tagged_count, 0 ]

  Levels:
    - |
      W   W   W   W   W   W   W   W   W
      W   .   .   f2  .   f12 .   .   W
      W   .   .   .   .   .   .   .   W
      W   f1  .   f3  .   f10 .   f11 W
      W   .   .   .   .   .   .   .   W
      W   .   .   .   .   .   .   .   W
      W   f4  .   f5  .   f7  .   f8  W
      W   .   .   .   .   .   .   .   W
      W   .   .   f6  .   f9  .   .   W
      W   W   W   W   W   W   W   W   W
    - |
      W   W   W   W   W   W   W   W   W   W   W   W   W   W   W   W   W   W   W   W   W ␣
↪ W
      W   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   . ␣
↪ W
      W   .   .   f2  .   .   .   .   .   .   .   .   .   .   .   .   .   .   f12 .   . ␣
↪ W
      W   .   f1  f3  .   .   .   .   .   .   .   .   .   .   .   .   .   .   f10 f11 . ␣
↪ W
      W   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   . ␣
↪ W
      W   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   . ␣
↪ W
      W   .   .   .   .   .   .   .   W   W   W   W   W   W   .   .   .   .   .   .   . ␣
↪ W
      W   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   . ␣
↪ W
      W   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   . ␣
↪ W
      W   .   .   .   .   .   .   W   .   .   .   .   .   .   W   .   .   .   .   .   . ␣
↪ W
      W   .   .   .   .   .   .   W   .   .   .   .   .   .   W   .   .   .   .   .   . ␣
↪ W
      W   .   .   .   .   .   .   W   .   .   .   .   .   .   W   .   .   .   .   .   . ␣
↪ W
      W   .   .   .   .   .   .   W   .   .   .   .   .   .   W   .   .   .   .   .   . ␣
↪ W
      W   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   . ␣
↪ W
      W   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   . ␣
↪ W
      W   .   .   .   .   .   .   .   W   W   W   W   W   W   .   .   .   .   .   .   . ␣
↪ W
      W   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   . ␣
↪ W
      W   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   . ␣
↪ W
```

(continues on next page)

(continued from previous page)

```
    W . f4 f5 . . . . . . . . . . . . . . f7 f8 . W
    W . . f6 . . . . . . . . . . . . . . f9 . . W
    W . . . . . . . . . . . . . . . . . . . . W
    W W W W W W W W W W W W W W W W W W W W W W
  - |
    W W W W W W W W W W W W W W W W W W W W W W
    W . . . . m . . . . . . . . . . m . . . . W
    W . . f2 . m . . . . . . . . . . m . f12 . . W
    W . f1 f3 . m . . . . . . . . . . m . f10 f11 . W
    W . . . . m . . . . . . . . . . m . . . . W
    W . . . . m . . . . . . . . . . m . . . . W
    W m m m m m . . W W W W W W . . m m m m m W
    W . . . . . . . . . . . . . . . . . . . . W
    W . . . . . . . . . . . . . . . . . . . . W
    W . . . . . . W . . . . . . W . . . . . . W
    W . . . . . . W . . . . . . W . . . . . . W
    W . . . . . . W . . . . . . W . . . . . . W
    W . . . . . . W . . . . . . W . . . . . . W
    W . . . . . . . . . . . . . . . . . . . . W
    W . . . . . . . . . . . . . . . . . . . . W
    W m m m m m . . W W W W W W . . m m m m m W
    W . . . . m . . . . . . . . . . m . . . . W
    W . . . . m . . . . . . . . . . m . . . . W
    W . f4 f5 . m . . . . . . . . . . m . f7 f8 . W
    W . . f6 . m . . . . . . . . . . m . f9 . . W
    W . . . . m . . . . . . . . . . m . . . . W
    W W W W W W W W W W W W W W W W W W W W W W
```

(continues on next page)

(continued from previous page)

```
  - |
    W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W
    W  .  .  .  .  m  .  .  .  .  .  m  .  .  m  .  .  .  .  .  .  .  .  .  .  m  .  .  m  .  .  .  .  .  m  .  .  .  .  W
    W  .  .  .  .  m  f2 .  .  .  .  m  .  .  m  .  .  .  .  .  .  .  .  .  .  m  .  .  m  .  .  .  .  f12 m  .  .  .  .  W
    W  .  .  f3 .  m  .  .  .  .  .  m  .  .  m  .  .  .  .  .  .  .  .  .  .  m  .  .  m  .  .  .  .  .  m  .  f10 .  .  W
    W  .  .  .  .  m  .  .  .  .  .  m  .  .  m  .  .  .  .  .  .  .  .  .  .  m  .  .  m  .  .  .  .  .  m  .  .  .  .  W
    W  .  .  .  .  m  .  .  .  .  .  m  .  .  m  .  .  .  .  .  .  .  .  .  .  m  .  .  m  .  .  .  .  .  m  .  .  .  .  W
    W  m  m  m  m  m  .  .  .  .  .  m  .  .  m  .  .  W  W  W  W  W  W  .  .  m  .  .  m  .  .  .  .  .  m  m  m  m  m  W
    W  .  f1 .  .  .  .  .  .  .  .  m  .  .  m  .  .  .  .  .  .  .  .  .  .  m  .  .  m  .  .  .  .  .  .  .  .  f11 .  W
    W  .  .  .  .  .  .  .  .  .  .  m  .  .  m  .  .  .  .  .  .  .  .  .  .  m  .  .  m  .  .  .  .  .  .  .  .  .  .  W
    W  .  .  .  .  .  .  .  .  .  .  m  .  .  m  .  .  .  .  .  .  .  .  .  .  m  .  .  m  .  .  .  .  .  .  .  .  .  .  W
    W  .  .  .  .  .  .  .  .  .  .  m  .  .  m  .  .  .  .  .  .  .  .  .  .  m  .  .  m  .  .  .  .  .  .  .  .  .  .  W
    W  .  .  .  .  .  .  .  .  .  .  m  .  .  m  .  .  .  .  .  .  .  .  .  .  m  .  .  m  .  .  .  .  .  .  .  .  .  .  W
    W  .  .  .  .  .  .  .  .  .  .  m  .  .  m  .  .  .  .  .  .  .  .  .  .  m  .  .  m  .  .  .  .  .  .  .  .  .  .  W
    W  .  .  .  .  .  .  .  .  .  .  m  .  .  m  .  .  .  .  .  .  .  .  .  .  m  .  .  m  .  .  .  .  .  .  .  .  .  .  W
    W  .  .  .  .  .  .  .  .  .  .  m  .  .  m  .  .  .  .  .  .  .  .  .  .  m  .  .  m  .  .  .  .  .  .  .  .  .  .  W
    W  .  .  .  .  .  .  .  .  .  .  m  .  .  m  .  .  .  .  .  .  .  .  .  .  m  .  .  m  .  .  .  .  .  .  .  .  .  .  W
    W  .  .  .  .  .  .  .  .  .  .  m  .  .  m  .  .  .  .  .  .  .  .  .  .  m  .  .  m  .  .  .  .  .  .  .  .  .  .  W
    W  .  .  .  .  .  .  .  .  .  .  m  .  .  m  .  .  .  .  .  .  .  .  .  .  m  .  .  m  .  .  .  .  .  .  .  .  .  .  W
    W  .  .  .  .  .  .  .  .  .  .  m  .  .  m  .  .  .  .  .  .  .  .  .  .  m  .  .  m  .  .  .  .  .  .  .  .  .  .  W
    W  .  .  .  .  .  .  .  .  .  .  m  .  .  m  .  W  .  .  .  .  .  .  W  .  m  .  .  m  .  .  .  .  .  .  .  .  .  .  W
    W  .  .  .  .  .  .  .  .  .  .  m  .  .  m  .  W  .  .  .  .  .  .  W  .  m  .  .  m  .  .  .  .  .  .  .  .  .  .  W
    W  .  .  .  .  .  .  .  .  .  .  m  .  .  m  .  W  .  .  .  .  .  .  W  .  m  .  .  m  .  .  .  .  .  .  .  .  .  .  W
    W  .  .  .  .  .  .  .  .  .  .  m  .  .  m  .  W  .  .  .  .  .  .  W  .  m  .  .  m  .  .  .  .  .  .  .  .  .  .  W
    W  .  .  .  .  .  .  .  .  .  .  m  .  .  m  .  .  .  .  .  .  .  .  .  .  m  .  .  m  .  .  .  .  .  .  .  .  .  .  W
    W  .  .  .  .  .  .  .  .  .  .  m  .  .  m  .  .  .  .  .  .  .  .  .  .  m  .  .  m  .  .  .  .  .  .  .  .  .  .  W
    W  .  .  .  .  .  .  .  .  .  .  m  .  .  m  .  .  .  .  .  .  .  .  .  .  m  .  .  m  .  .  .  .  .  .  .  .  .  .  W
```

(continues on next page)

(continued from previous page)

```
      W   .   .   .   .   .   .   .   .   .   .   m   .   .   m   .   .   .   .   .   .   .   .   .   .   m   .   .   m   .   .   .   .   .   .   .   .   .   .   W
      W   .   .   .   .   .   .   .   .   .   .   m   .   .   m   .   .   .   .   .   .   .   .   .   .   m   .   .   m   .   .   .   .   .   .   .   .   .   .   W
      W   .   .   .   .   .   .   .   .   .   .   m   .   .   m   .   .   .   .   .   .   .   .   .   .   m   .   .   m   .   .   .   .   .   .   .   .   .   .   W
      W   .   .   .   .   .   .   .   .   .   .   m   .   .   m   .   .   .   .   .   .   .   .   .   .   m   .   .   m   .   .   .   .   .   .   .   .   .   .   W
      W   .   .   .   .   .   .   .   .   .   .   m   .   .   m   .   .   .   .   .   .   .   .   .   .   m   .   .   m   .   .   .   .   .   .   .   .   .   .   W
      W   .   .   .   .   .   .   .   .   .   .   m   .   .   m   .   .   .   .   .   .   .   .   .   .   m   .   .   m   .   .   .   .   .   .   .   .   .   .   W
      W   .   .   .   .   .   .   .   .   .   .   m   .   .   m   .   .   .   .   .   .   .   .   .   .   m   .   .   m   .   .   .   .   .   .   .   .   .   .   W
      W   .   .   .   .   .   .   .   .   .   .   m   .   .   m   .   .   .   .   .   .   .   .   .   .   m   .   .   m   .   .   .   .   .   .   .   .   .   .   W
      W   .   .   .   .   .   .   .   .   .   .   m   .   .   m   .   .   .   .   .   .   .   .   .   .   m   .   .   m   .   .   .   .   .   .   .   .   .   .   W
      W   .   .   .   .   .   .   .   .   .   .   m   .   .   m   .   .   .   .   .   .   .   .   .   .   m   .   .   m   .   .   .   .   .   .   .   .   .   .   W
      W   .   .   .   .   .   .   .   .   .   .   m   .   .   m   .   .   .   .   .   .   .   .   .   .   m   .   .   m   .   .   .   .   .   .   .   .   .   .   W
      W   .   .   .   .   .   .   .   .   .   .   m   .   .   m   .   .   .   .   .   .   .   .   .   .   m   .   .   m   .   .   .   .   .   .   .   .   .   .   W
      W   .   f4  .   .   .   .   .   .   .   .   m   .   .   m   .   .   .   .   .   .   .   .   .   .   m   .   .   m   .   .   .   .   .   .   .   .   f8  .   W
      W   m   m   m   m   m   .   .   .   .   .   m   .   .   m   .   .   W   W   W   W   W   W   .   .   m   .   .   m   .   .   .   .   .   m   m   m   m   m   W
      W   .   .   .   .   m   .   .   .   .   .   m   .   .   m   .   .   .   .   .   .   .   .   .   .   m   .   .   m   .   .   .   .   .   m   .   .   .   .   W
      W   .   .   .   .   m   .   .   .   .   .   m   .   .   m   .   .   .   .   .   .   .   .   .   .   m   .   .   m   .   .   .   .   .   m   .   .   .   .   W
      W   .   .   f5  .   m   .   .   .   .   .   m   .   .   m   .   .   .   .   .   .   .   .   .   .   m   .   .   m   .   .   .   .   .   m   .   f7  .   .   W
      W   .   .   .   .   m   f6  .   .   .   .   m   .   .   m   .   .   .   .   .   .   .   .   .   .   m   .   .   m   .   .   .   .   f9  m   .   .   .   .   W
      W   .   .   .   .   m   .   .   .   .   .   m   .   .   m   .   .   .   .   .   .   .   .   .   .   m   .   .   m   .   .   .   .   .   m   .   .   .   .   W
      W   W   W   W   W   W   W   W   W   W   W   W   W   W   W   W   W   W   W   W   W   W   W   W   W   W   W   W   W   W   W   W   W   W   W   W   W   W   W   W

Actions:

  # Taggers have a random chance of starting in a tagged state
  - Name: initialize_is_tagged
    InputMapping:
      Internal: true
      Inputs:
        1:
          Description: Initialize Tagged
        2:
```

(continues on next page)

(continued from previous page)

```
        Description: Initialize Not Tagged
        VectorToDest: [ -1, 0 ]

  Behaviours:
    - Src:
        Object: tagger
        Preconditions:
          - eq: [ src._playerId, dst._playerId ]
        Commands:
          - set_tile: 1
          - set: [ is_tagged, 1 ]
          - incr: tagged_count
      Dst:
        Object: tagger

- Name: tag
  Behaviours:
    - Src:
        Object: tagger
        Preconditions:
          - eq: [ src.is_tagged, 1 ]
          - eq: [ dst.is_tagged, 0 ]
        Commands:
          - reward: 2
          - set_tile: 0
          - set: [ is_tagged, 0 ]
      Dst:
        Object: tagger
        Commands:
          - set_tile: 1
          - set: [ is_tagged, 1 ]
          - reward: -2
          - incr: times_tagged
          - eq:
              Arguments: [ times_tagged, 3 ]
              Commands:
                - set: [ player_done, 1 ]
                - decr: tagged_count
                - reward: -5
                - remove: true

- Name: move
  Behaviours:
    - Src:
        Object: [ tagger, moveable_wall ]
        Commands:
          - mov: _dest # mov will move the object, _dest is the destination location␣
↪of the action
      Dst:
        Object: _empty

    - Src:
```

(continues on next page)

(continued from previous page)

```
          Object: tagger
          Commands:
            - mov: _dest
        Dst:
          Object: moveable_wall
          Commands:
            - cascade: _dest

Objects:
  - Name: tagger
    MapCharacter: f
    InitialActions:
      - Action: initialize_is_tagged
        Randomize: true
    Variables:
      - Name: is_tagged
        InitialValue: 0
      - Name: times_tagged
        InitialValue: 0
    Observers:
      Sprite2D:
        - Image: oryx/oryx_fantasy/avatars/robot1.png
        - Image: oryx/oryx_fantasy/avatars/fireguy1.png
      Block2D:
        - Shape: triangle
          Color: [ 0.2, 0.2, 0.9 ]
          Scale: 0.5
        - Shape: triangle
          Color: [ 0.9, 0.2, 0.2 ]
          Scale: 1.0

  - Name: moveable_wall
    MapCharacter: m
    Observers:
      Sprite2D:
        - Image: oryx/oryx_fantasy/wall4-0.png
      Block2D:
        - Color: [ 0.8, 0.8, 0.8 ]
          Shape: square

  - Name: fixed_wall
    MapCharacter: W
    Observers:
      Sprite2D:
        - TilingMode: WALL_16
          Image:
            - oryx/oryx_fantasy/wall2-0.png
            - oryx/oryx_fantasy/wall2-1.png
            - oryx/oryx_fantasy/wall2-2.png
            - oryx/oryx_fantasy/wall2-3.png
            - oryx/oryx_fantasy/wall2-4.png
            - oryx/oryx_fantasy/wall2-5.png
```

(continues on next page)

(continued from previous page)

```
            - oryx/oryx_fantasy/wall2-6.png
            - oryx/oryx_fantasy/wall2-7.png
            - oryx/oryx_fantasy/wall2-8.png
            - oryx/oryx_fantasy/wall2-9.png
            - oryx/oryx_fantasy/wall2-10.png
            - oryx/oryx_fantasy/wall2-11.png
            - oryx/oryx_fantasy/wall2-12.png
            - oryx/oryx_fantasy/wall2-13.png
            - oryx/oryx_fantasy/wall2-14.png
            - oryx/oryx_fantasy/wall2-15.png
      Block2D:
        - Color: [ 0.5, 0.5, 0.5 ]
          Shape: square
```

15.2 Robot Tag 8v8

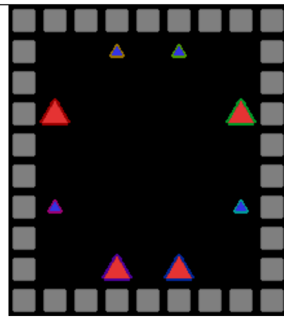
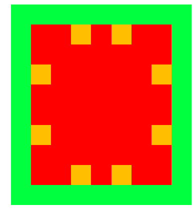
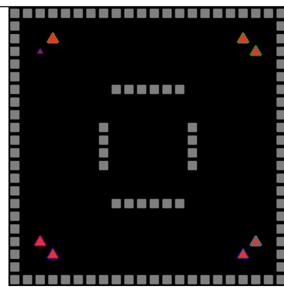
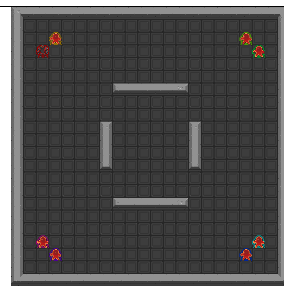
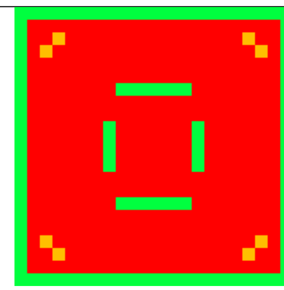
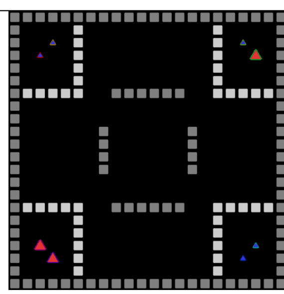
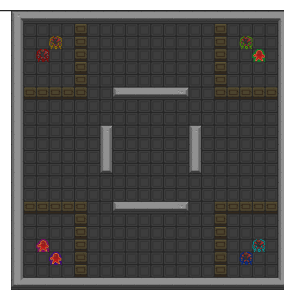
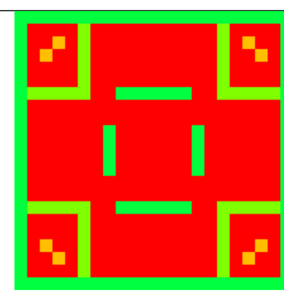
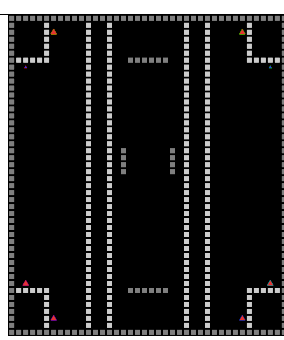
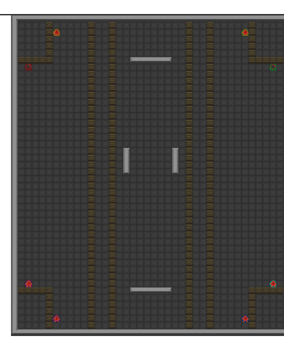
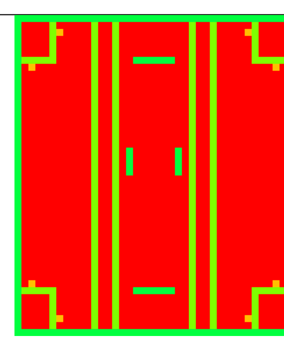
```
Multi-Agent/robot_tag_8.yaml
```

15.2.1 Description

Robots start randomly as “tagged” or not, robots can “tag” other robots. Any robot that is “tagged” 3 times dies.

15.2.2 Levels

Table 3: Levels

<table>
<tr><th></th><th>Block2D</th><th>Sprite2D</th><th>Vector</th></tr>
<tr><td><table><tr><td>Level ID</td><td>0</td></tr><tr><td>Size</td><td>9x10</td></tr></table></td><td></td><td></td><td></td></tr>
<tr><td><table><tr><td>Level ID</td><td>1</td></tr><tr><td>Size</td><td>22x22</td></tr></table></td><td></td><td></td><td></td></tr>
<tr><td><table><tr><td>Level ID</td><td>2</td></tr><tr><td>Size</td><td>22x22</td></tr></table></td><td></td><td></td><td></td></tr>
<tr><td><table><tr><td>Level ID</td><td>3</td></tr><tr><td>Size</td><td>40x46</td></tr></table></td><td></td><td></td><td></td></tr>
</table>

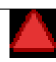
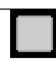
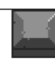
15.2.3 Code Example

The most basic way to create a Griddly Gym Environment. Defaults to level 0 and SPRITE_2D rendering.

```python
import gym
import griddly

if __name__ == '__main__':

    env = gym.make('GDY-Robot-Tag-8v8-v0')
    env.reset()

    # Replace with your own control algorithm!
    for s in range(1000):
        obs, reward, done, info = env.step(env.action_space.sample())
        for p in range(env.player_count):
            env.render(observer=p) # Renders the environment from the perspective of a single player

        env.render(observer='global') # Renders the entire environment

        if done:
            env.reset()
```

15.2.4 Objects

Table 4: Tiles

Name ->	tagger	moveable_wall	fixed_wall
Map Char ->	*f*	*m*	*W*
Block2D			
Sprite2D			
Vector			

15.2.5 Actions

move

Action Id	Mapping
1	Left
2	Up
3	Right
4	Down

initialize_is_tagged

Internal This action can only be called from other actions, not by the player.

Action Id	Mapping
1	Initialize Tagged
2	Initialize Not Tagged

tag

Action Id	Mapping
1	Left
2	Up
3	Right
4	Down

15.2.6 YAML

```
Version: "0.1"
Environment:
  Name: Robot Tag 8v8
  Description: Robots start randomly as "tagged" or not, robots can "tag" other robots.␣
↪Any robot that is "tagged" 3 times dies.
  Observers:
    Block2D:
      TileSize: 24
    Sprite2D:
      TileSize: 24
      BackgroundTile: oryx/oryx_fantasy/floor1-1.png
    Vector:
      IncludePlayerId: true
      IncludeVariables: true
  Variables:
    - Name: player_done
      InitialValue: 0
      PerPlayer: true
    - Name: tagged_count
      InitialValue: 0
  Player:
    Count: 8
    Observer:
      RotateWithAvatar: true
      TrackAvatar: true
      Height: 9
      Width: 9
      OffsetX: 0
      OffsetY: 0
    AvatarObject: tagger
  Termination:
```

(continues on next page)

(continued from previous page)

```
    End:
      - eq: [ tagged_count, 0 ]

  Levels:
    - |
      W   W   W   W   W   W   W   W   W
      W   .   .   f2  .   f3  .   .   W
      W   .   .   .   .   .   .   .   W
      W   f1  .   .   .   .   .   f4  W
      W   .   .   .   .   .   .   .   W
      W   .   .   .   .   .   .   .   W
      W   f8  .   .   .   .   .   f5  W
      W   .   .   .   .   .   .   .   W
      W   .   .   f7  .   f6  .   .   W
      W   W   W   W   W   W   W   W   W
    - |
      W   W   W   W   W   W   W   W   W   W   W   W   W   W   W   W   W   W   W   W   W   W
      W   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   W
      W   .   .   f2  .   .   .   .   .   .   .   .   .   .   .   .   .   .   f3  .   .   W
      W   .   f1  .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   f4  .   W
      W   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   W
      W   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   W
      W   .   .   .   .   .   .   .   W   W   W   W   W   W   .   .   .   .   .   .   .   W
      W   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   W
      W   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   W
      W   .   .   .   .   .   .   W   .   .   .   .   .   .   W   .   .   .   .   .   .   W
      W   .   .   .   .   .   .   W   .   .   .   .   .   .   W   .   .   .   .   .   .   W
      W   .   .   .   .   .   .   W   .   .   .   .   .   .   W   .   .   .   .   .   .   W
      W   .   .   .   .   .   .   W   .   .   .   .   .   .   W   .   .   .   .   .   .   W
      W   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   W
      W   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   W
      W   .   .   .   .   .   .   .   W   W   W   W   W   W   .   .   .   .   .   .   .   W
      W   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   W
      W   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   W
```

(continues on next page)

(continued from previous page)

```
W . f8 . . . . . . . . . . . . . . . . f5 . W
W . . f7 . . . . . . . . . . . . . . f6 . . W
W . . . . . . . . . . . . . . . . . . . . W
W W W W W W W W W W W W W W W W W W W W W W
 - |
W W W W W W W W W W W W W W W W W W W W W W
W . . . . m . . . . . . . . . . m . . . . W
W . . f2 . m . . . . . . . . . . m . f3 . . W
W . f1 . . m . . . . . . . . . . m . . f4 . W
W . . . . m . . . . . . . . . . m . . . . W
W . . . . m . . . . . . . . . . m . . . . W
W m m m m m . . W W W W W W . . m m m m m W
W . . . . . . . . . . . . . . . . . . . . W
W . . . . . . . . . . . . . . . . . . . . W
W . . . . . . W . . . . . . W . . . . . . W
W . . . . . . W . . . . . . W . . . . . . W
W . . . . . . W . . . . . . W . . . . . . W
W . . . . . . W . . . . . . W . . . . . . W
W . . . . . . . . . . . . . . . . . . . . W
W . . . . . . . . . . . . . . . . . . . . W
W m m m m m . . W W W W W W . . m m m m m W
W . . . . m . . . . . . . . . . m . . . . W
W . . . . m . . . . . . . . . . m . . . . W
W . f8 . . m . . . . . . . . . . m . . f5 . W
W . . f7 . m . . . . . . . . . . m . f6 . . W
W . . . . m . . . . . . . . . . m . . . . W
W W W W W W W W W W W W W W W W W W W W W W
```

(continues on next page)

(continued from previous page)

```
    - |
      W W W W W W W W W W W W W W W W W W W W W W W W W W W W W W W W W W W W W W W W
      W . . . . m . . . . . m . . m . . . . . . . . . . m . . m . . . . . m . . . . W
      W . . . . m f2 . . . . m . . m . . . . . . . . . . m . . m . . . . f3 m . . . . W
      W . . . . m . . . . . m . . m . . . . . . . . . . m . . m . . . . . m . . . . W
      W . . . . m . . . . . m . . m . . . . . . . . . . m . . m . . . . . m . . . . W
      W . . . . m . . . . . m . . m . . . . . . . . . . m . . m . . . . . m . . . . W
      W m m m m m . . . . . m . . m . . W W W W W W . . m . . m . . . . . m m m m m W
      W . f1 . . . . . . . . m . . m . . . . . . . . . . m . . m . . . . . . . . f4 . W
      W . . . . . . . . . . m . . m . . . . . . . . . . m . . m . . . . . . . . . . W
      W . . . . . . . . . . m . . m . . . . . . . . . . m . . m . . . . . . . . . . W
      W . . . . . . . . . . m . . m . . . . . . . . . . m . . m . . . . . . . . . . W
      W . . . . . . . . . . m . . m . . . . . . . . . . m . . m . . . . . . . . . . W
      W . . . . . . . . . . m . . m . . . . . . . . . . m . . m . . . . . . . . . . W
      W . . . . . . . . . . m . . m . . . . . . . . . . m . . m . . . . . . . . . . W
      W . . . . . . . . . . m . . m . . . . . . . . . . m . . m . . . . . . . . . . W
      W . . . . . . . . . . m . . m . . . . . . . . . . m . . m . . . . . . . . . . W
      W . . . . . . . . . . m . . m . . . . . . . . . . m . . m . . . . . . . . . . W
      W . . . . . . . . . . m . . m . . . . . . . . . . m . . m . . . . . . . . . . W
      W . . . . . . . . . . m . . m . . . . . . . . . . m . . m . . . . . . . . . . W
      W . . . . . . . . . . m . . m . W . . . . . . W . m . . m . . . . . . . . . . W
      W . . . . . . . . . . m . . m . W . . . . . . W . m . . m . . . . . . . . . . W
      W . . . . . . . . . . m . . m . W . . . . . . W . m . . m . . . . . . . . . . W
      W . . . . . . . . . . m . . m . W . . . . . . W . m . . m . . . . . . . . . . W
      W . . . . . . . . . . m . . m . . . . . . . . . . m . . m . . . . . . . . . . W
      W . . . . . . . . . . m . . m . . . . . . . . . . m . . m . . . . . . . . . . W
      W . . . . . . . . . . m . . m . . . . . . . . . . m . . m . . . . . . . . . . W
```

(continues on next page)

(continued from previous page)

```
      W  .  .  .  .  .  .  .  .  .  .  m  .  .  m  .  .  .  .  .  .  .  .  .  .  m  .  .  m  .  .  .  .  .  .  .  .  .  .  W
      W  .  .  .  .  .  .  .  .  .  .  m  .  .  m  .  .  .  .  .  .  .  .  .  .  m  .  .  m  .  .  .  .  .  .  .  .  .  .  W
      W  .  .  .  .  .  .  .  .  .  .  m  .  .  m  .  .  .  .  .  .  .  .  .  .  m  .  .  m  .  .  .  .  .  .  .  .  .  .  W
      W  .  .  .  .  .  .  .  .  .  .  m  .  .  m  .  .  .  .  .  .  .  .  .  .  m  .  .  m  .  .  .  .  .  .  .  .  .  .  W
      W  .  .  .  .  .  .  .  .  .  .  m  .  .  m  .  .  .  .  .  .  .  .  .  .  m  .  .  m  .  .  .  .  .  .  .  .  .  .  W
      W  .  .  .  .  .  .  .  .  .  .  m  .  .  m  .  .  .  .  .  .  .  .  .  .  m  .  .  m  .  .  .  .  .  .  .  .  .  .  W
      W  .  .  .  .  .  .  .  .  .  .  m  .  .  m  .  .  .  .  .  .  .  .  .  .  m  .  .  m  .  .  .  .  .  .  .  .  .  .  W
      W  .  .  .  .  .  .  .  .  .  .  m  .  .  m  .  .  .  .  .  .  .  .  .  .  m  .  .  m  .  .  .  .  .  .  .  .  .  .  W
      W  .  .  .  .  .  .  .  .  .  .  m  .  .  m  .  .  .  .  .  .  .  .  .  .  m  .  .  m  .  .  .  .  .  .  .  .  .  .  W
      W  .  .  .  .  .  .  .  .  .  .  m  .  .  m  .  .  .  .  .  .  .  .  .  .  m  .  .  m  .  .  .  .  .  .  .  .  .  .  W
      W  .  .  .  .  .  .  .  .  .  .  m  .  .  m  .  .  .  .  .  .  .  .  .  .  m  .  .  m  .  .  .  .  .  .  .  .  .  .  W
      W  .  .  .  .  .  .  .  .  .  .  m  .  .  m  .  .  .  .  .  .  .  .  .  .  m  .  .  m  .  .  .  .  .  .  .  .  .  .  W
      W  .  f8 .  .  .  .  .  .  .  .  m  .  .  m  .  .  .  .  .  .  .  .  .  .  m  .  .  m  .  .  .  .  .  .  .  .  f5 .  W
      W  m  m  m  m  m  .  .  .  .  .  m  .  .  m  .  .  W  W  W  W  W  W  .  .  m  .  .  m  .  .  .  .  .  m  m  m  m  m  W
      W  .  .  .  .  m  .  .  .  .  .  m  .  .  m  .  .  .  .  .  .  .  .  .  .  m  .  .  m  .  .  .  .  .  m  .  .  .  .  W
      W  .  .  .  .  m  .  .  .  .  .  m  .  .  m  .  .  .  .  .  .  .  .  .  .  m  .  .  m  .  .  .  .  .  m  .  .  .  .  W
      W  .  .  .  .  m  .  .  .  .  .  m  .  .  m  .  .  .  .  .  .  .  .  .  .  m  .  .  m  .  .  .  .  .  m  .  .  .  .  W
      W  .  .  .  .  m  f7 .  .  .  .  m  .  .  m  .  .  .  .  .  .  .  .  .  .  m  .  .  m  .  .  .  .  f6 m  .  .  .  .  W
      W  .  .  .  .  m  .  .  .  .  .  m  .  .  m  .  .  .  .  .  .  .  .  .  .  m  .  .  m  .  .  .  .  .  m  .  .  .  .  W
      W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W

Actions:

  # Taggers have a random chance of starting in a tagged state
  - Name: initialize_is_tagged
    InputMapping:
      Internal: true
      Inputs:
        1:
          Description: Initialize Tagged
        2:
```

(continues on next page)

(continued from previous page)

```
        Description: Initialize Not Tagged
        VectorToDest: [ -1, 0 ]

  Behaviours:
    - Src:
        Object: tagger
        Preconditions:
          - eq: [ src._playerId, dst._playerId ]
        Commands:
          - set_tile: 1
          - set: [ is_tagged, 1 ]
          - incr: tagged_count
      Dst:
        Object: tagger

- Name: tag
  Behaviours:
    - Src:
        Object: tagger
        Preconditions:
          - eq: [ src.is_tagged, 1 ]
          - eq: [ dst.is_tagged, 0 ]
        Commands:
          - reward: 2
          - set_tile: 0
          - set: [ is_tagged, 0 ]
      Dst:
        Object: tagger
        Commands:
          - set_tile: 1
          - set: [ is_tagged, 1 ]
          - reward: -2
          - incr: times_tagged
          - eq:
              Arguments: [ times_tagged, 3 ]
              Commands:
                - set: [ player_done, 1 ]
                - decr: tagged_count
                - reward: -5
                - remove: true

- Name: move
  Behaviours:
    - Src:
        Object: [tagger, moveable_wall]
        Commands:
          - mov: _dest # mov will move the object, _dest is the destination location␣
↪of the action
      Dst:
        Object: _empty

    - Src:
```

(continues on next page)

(continued from previous page)

```
          Object: tagger
          Commands:
            - mov: _dest
        Dst:
          Object: moveable_wall
          Commands:
            - cascade: _dest

Objects:
  - Name: tagger
    MapCharacter: f
    InitialActions:
      - Action: initialize_is_tagged
        Randomize: true
    Variables:
      - Name: is_tagged
        InitialValue: 0
      - Name: times_tagged
        InitialValue: 0
    Observers:
      Sprite2D:
        - Image: oryx/oryx_fantasy/avatars/robot1.png
        - Image: oryx/oryx_fantasy/avatars/fireguy1.png
      Block2D:
        - Shape: triangle
          Color: [ 0.2, 0.2, 0.9 ]
          Scale: 0.5
        - Shape: triangle
          Color: [ 0.9, 0.2, 0.2 ]
          Scale: 1.0

  - Name: moveable_wall
    MapCharacter: m
    Observers:
      Sprite2D:
        - Image: oryx/oryx_fantasy/wall4-0.png
      Block2D:
        - Color: [ 0.8, 0.8, 0.8 ]
          Shape: square

  - Name: fixed_wall
    MapCharacter: W
    Observers:
      Sprite2D:
        - TilingMode: WALL_16
          Image:
            - oryx/oryx_fantasy/wall2-0.png
            - oryx/oryx_fantasy/wall2-1.png
            - oryx/oryx_fantasy/wall2-2.png
            - oryx/oryx_fantasy/wall2-3.png
            - oryx/oryx_fantasy/wall2-4.png
            - oryx/oryx_fantasy/wall2-5.png
```

(continues on next page)

(continued from previous page)

```
            - oryx/oryx_fantasy/wall2-6.png
            - oryx/oryx_fantasy/wall2-7.png
            - oryx/oryx_fantasy/wall2-8.png
            - oryx/oryx_fantasy/wall2-9.png
            - oryx/oryx_fantasy/wall2-10.png
            - oryx/oryx_fantasy/wall2-11.png
            - oryx/oryx_fantasy/wall2-12.png
            - oryx/oryx_fantasy/wall2-13.png
            - oryx/oryx_fantasy/wall2-14.png
            - oryx/oryx_fantasy/wall2-15.png
      Block2D:
        - Color: [ 0.5, 0.5, 0.5 ]
          Shape: square
```

15.3 Foragers

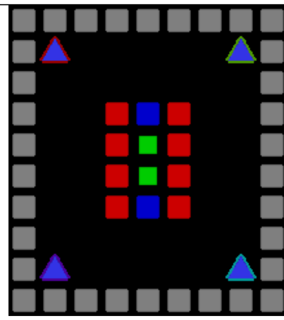
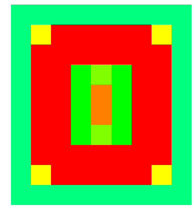
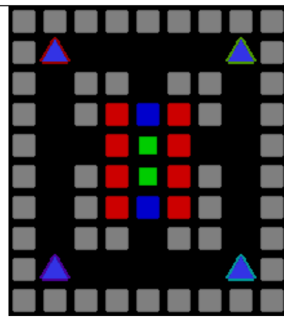
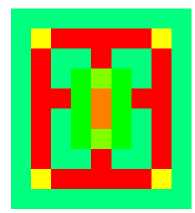
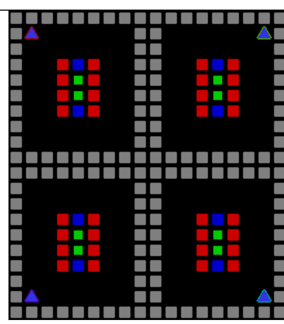
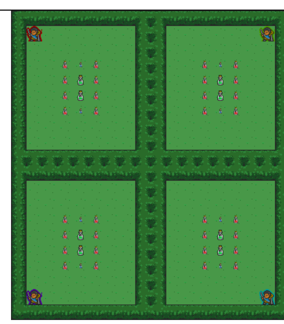
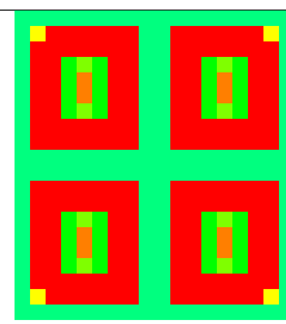
```
Multi-Agent/foragers.yaml
```

15.3.1 Description

A very simple multi-agent game. Agents must collect the coloured potions

15.3.2 Levels

Table 5: Levels

<table>
<tr><th></th><th>Block2D</th><th>Sprite2D</th><th>Vector</th></tr>
<tr><td><table><tr><td>Level ID</td><td>0</td></tr><tr><td>Size</td><td>9x10</td></tr></table></td><td></td><td></td><td></td></tr>
<tr><td><table><tr><td>Level ID</td><td>1</td></tr><tr><td>Size</td><td>9x10</td></tr></table></td><td></td><td></td><td></td></tr>
<tr><td><table><tr><td>Level ID</td><td>2</td></tr><tr><td>Size</td><td>18x20</td></tr></table></td><td></td><td></td><td></td></tr>
</table>

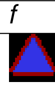
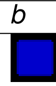
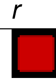
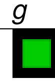
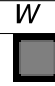
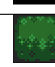
15.3.3 Code Example

The most basic way to create a Griddly Gym Environment. Defaults to level 0 and SPRITE_2D rendering.

```python
import gym
import griddly

if __name__ == '__main__':

    env = gym.make('GDY-Foragers-v0')
    env.reset()

    # Replace with your own control algorithm!
    for s in range(1000):
        obs, reward, done, info = env.step(env.action_space.sample())
        for p in range(env.player_count):
            env.render(observer=p) # Renders the environment from the perspective of a single player

        env.render(observer='global') # Renders the entire environment

        if done:
            env.reset()
```

15.3.4 Objects

Table 6: Tiles

Name ->	harvester	potion1	potion2	potion3	fixed_wall
Map Char ->	*f*	*b*	*r*	*g*	*W*
Block2D					
Sprite2D					
Vector					

15.3.5 Actions

move

Action Id	Mapping
1	Left
2	Up
3	Right
4	Down

init_potion

Internal This action can only be called from other actions, not by the player.

Action Id	Mapping
1	The only action here is to increment the potion count

gather

Action Id	Mapping
1	Left
2	Up
3	Right
4	Down

15.3.6 YAML

```yaml
Version: "0.1"
Environment:
  Name: Foragers
  Description: A very simple multi-agent game. Agents must collect the coloured potions
  Observers:
    Sprite2D:
      TileSize: 24
      BackgroundTile: gvgai/oryx/grass_15.png
    Block2D:
      TileSize: 24
  Player:
    Count: 4
    Observer:
      TrackAvatar: true
      Height: 5
      Width: 5
      OffsetX: 0
      OffsetY: 0
    AvatarObject: harvester
  Variables:
    - Name: potion_count
      InitialValue: 0
  Termination:
    End:
      - eq: [potion_count, 0]

  Levels:
    - |
      W   W   W   W   W   W   W   W   W
      W   f1  .   .  .    .   .   f2  W
      W   .   .   .   .   .   .   .   W
      W   .   .   r   b   r   .   .   W
      W   .   .   r   g   r   .   .   W
```

(continues on next page)

(continued from previous page)

```
      W   .   .   r   g   r   .   .   W
      W   .   .   r   b   r   .   .   W
      W   .   .   .   .   .   .   .   W
      W   f4  .   .   .   .   .   f3  W
      W   W   W   W   W   W   W   W   W
    - |
      W   W   W   W   W   W   W   W   W
      W   f1  .   .   .   .   .   f2  W
      W   .   W   W   .   W   W   .   W
      W   .   W   r   b   r   W   .   W
      W   .   .   r   g   r   .   .   W
      W   .   W   r   g   r   W   .   W
      W   .   W   r   b   r   W   .   W
      W   .   W   W   .   W   W   .   W
      W   f4  .   .   .   .   .   f3  W
      W   W   W   W   W   W   W   W   W
    - |
      W   W   W   W   W   W   W   W   W   W   W   W   W   W   W   W   W   W
      W   f1  .   .  .    .   .   .   W   W   .   .   .  .    .   .   f2  W
      W   .   .   .   .   .   .   .   W   W   .   .   .   .   .   .   .   W
      W   .   .   r   b   r   .   .   W   W   .   .   r   b   r   .   .   W
      W   .   .   r   g   r   .   .   W   W   .   .   r   g   r   .   .   W
      W   .   .   r   g   r   .   .   W   W   .   .   r   g   r   .   .   W
      W   .   .   r   b   r   .   .   W   W   .   .   r   b   r   .   .   W
      W   .   .   .   .   .   .   .   W   W   .   .   .   .   .   .   .   W
      W   .   .   .   .   .   .   .   W   W   .   .   .   .   .   .   .   W
      W   W   W   W   W   W   W   W   W   W   W   W   W   W   W   W   W   W
      W   W   W   W   W   W   W   W   W   W   W   W   W   W   W   W   W   W
      W   .   .   .  .    .   .   .   W   W   .   .   .  .    .   .   .   W
      W   .   .   .   .   .   .   .   W   W   .   .   .   .   .   .   .   W
      W   .   .   r   b   r   .   .   W   W   .   .   r   b   r   .   .   W
      W   .   .   r   g   r   .   .   W   W   .   .   r   g   r   .   .   W
      W   .   .   r   g   r   .   .   W   W   .   .   r   g   r   .   .   W
      W   .   .   r   b   r   .   .   W   W   .   .   r   b   r   .   .   W
      W   .   .   .   .   .   .   .   W   W   .   .   .   .   .   .   .   W
      W   f4  .   .   .   .   .   .   W   W   .   .   .   .   .   .   f3  W
      W   W   W   W   W   W   W   W   W   W   W   W   W   W   W   W   W   W


Actions:
  - Name: init_potion
    InputMapping:
      Internal: true
      Inputs:
        1:
          Description: "The only action here is to increment the potion count"
    Behaviours:
      - Src:
          Object: [ potion1, potion2, potion3 ]
          Commands:
            - incr: potion_count
        Dst:
```

(continues on next page)

(continued from previous page)

```
        Object: [ potion1, potion2, potion3 ]

  - Name: gather
    Behaviours:
      - Src:
          Object: harvester
          Commands:
            - reward: 1
        Dst:
          Object: [potion1, potion2, potion3]
          Commands:
            - decr: value
            - eq:
                Arguments: [ value, 0 ]
                Commands:
                  - decr: potion_count
                  - remove: true

  - Name: move
    Behaviours:
      - Src:
          Object: harvester
          Commands:
            - mov: _dest
        Dst:
          Object: _empty

Objects:
  - Name: harvester
    MapCharacter: f
    Observers:
      Sprite2D:
        - Image: oryx/oryx_fantasy/avatars/man1.png
      Block2D:
        - Shape: triangle
          Color: [ 0.2, 0.2, 0.9 ]
          Scale: 1.0

  - Name: potion1
    MapCharacter: b
    InitialActions:
      - Action: init_potion
        ActionId: 1
    Variables:
      - Name: value
        InitialValue: 5
    Observers:
      Sprite2D:
        - Image: oryx/oryx_fantasy/potion-0.png
          Scale: 0.5
      Block2D:
        - Color: [ 0.0, 0.0, 0.8 ]
```

(continues on next page)

(continued from previous page)

```
        Shape: square

  - Name: potion2
    MapCharacter: r
    InitialActions:
      - Action: init_potion
        ActionId: 1
    Variables:
      - Name: value
        InitialValue: 10
    Observers:
      Sprite2D:
        - Image: oryx/oryx_fantasy/potion-2.png
          Scale: 0.8
      Block2D:
        - Color: [ 0.8, 0.0, 0.0 ]
          Shape: square

  - Name: potion3
    MapCharacter: g
    InitialActions:
      - Action: init_potion
        ActionId: 1
    Variables:
      - Name: value
        InitialValue: 20
    Observers:
      Sprite2D:
        - Image: oryx/oryx_fantasy/potion-3.png
          Scale: 1.0
      Block2D:
        - Color: [ 0.0, 0.8, 0.0 ]
          Shape: square
          Scale: 0.8

  - Name: fixed_wall
    MapCharacter: W
    Observers:
      Sprite2D:
        - TilingMode: WALL_16
          Image:
            - oryx/oryx_fantasy/wall9-0.png
            - oryx/oryx_fantasy/wall9-1.png
            - oryx/oryx_fantasy/wall9-2.png
            - oryx/oryx_fantasy/wall9-3.png
            - oryx/oryx_fantasy/wall9-4.png
            - oryx/oryx_fantasy/wall9-5.png
            - oryx/oryx_fantasy/wall9-6.png
            - oryx/oryx_fantasy/wall9-7.png
            - oryx/oryx_fantasy/wall9-8.png
            - oryx/oryx_fantasy/wall9-9.png
            - oryx/oryx_fantasy/wall9-10.png
```

(continues on next page)

(continued from previous page)

```
          - oryx/oryx_fantasy/wall9-11.png
          - oryx/oryx_fantasy/wall9-12.png
          - oryx/oryx_fantasy/wall9-13.png
          - oryx/oryx_fantasy/wall9-14.png
          - oryx/oryx_fantasy/wall9-15.png
      Block2D:
        - Color: [ 0.5, 0.5, 0.5 ]
          Shape: square
```

15.4 Robot Tag 4v4

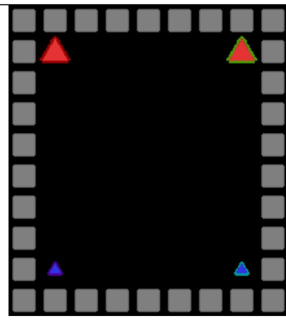
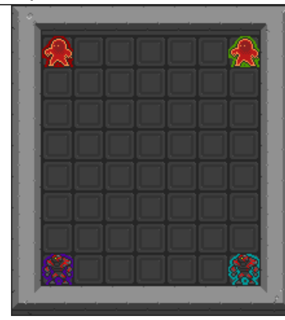
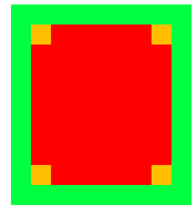
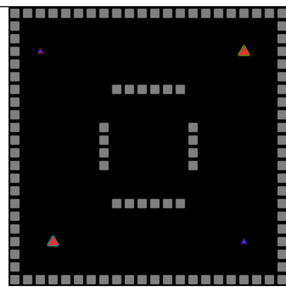
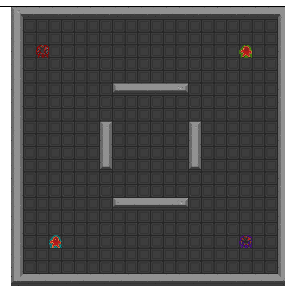
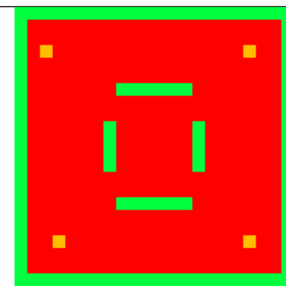
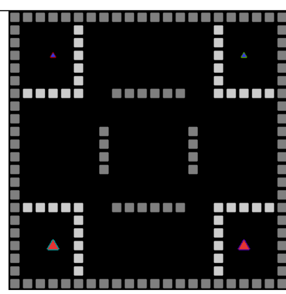
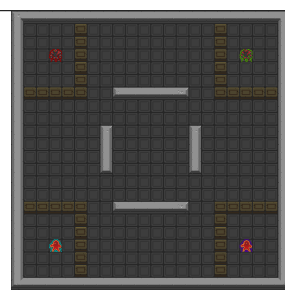
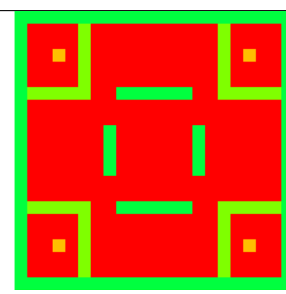
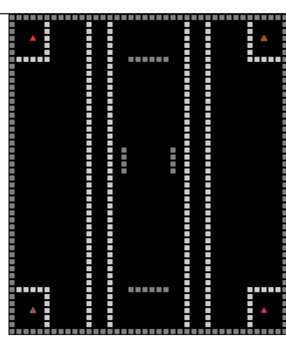
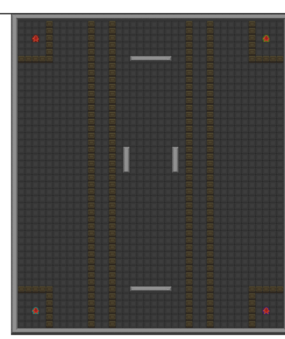
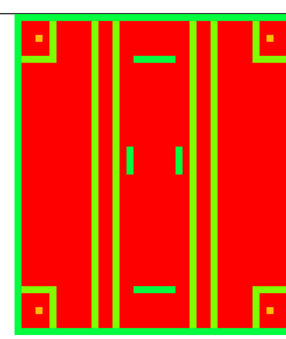
```
Multi-Agent/robot_tag_4.yaml
```

15.4.1 Description

Robots start randomly as "tagged" or not, robots can "tag" other robots. Any robot that is "tagged" 3 times dies.

15.4.2 Levels

Table 7: Levels

	Block2D	Sprite2D	Vector
Level ID 0 Size 9x10			
Level ID 1 Size 22x22			
Level ID 2 Size 22x22			
Level ID 3 Size 40x46			

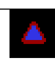
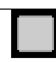
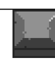
15.4.3 Code Example

The most basic way to create a Griddly Gym Environment. Defaults to level 0 and SPRITE_2D rendering.

```
import gym
import griddly

if __name__ == '__main__':

    env = gym.make('GDY-Robot-Tag-4v4-v0')
    env.reset()

    # Replace with your own control algorithm!
    for s in range(1000):
        obs, reward, done, info = env.step(env.action_space.sample())
        for p in range(env.player_count):
            env.render(observer=p) # Renders the environment from the perspective of a single player

        env.render(observer='global') # Renders the entire environment

        if done:
            env.reset()
```

15.4.4 Objects

Table 8: Tiles

Name ->	tagger	moveable_wall	fixed_wall
Map Char ->	*f*	*m*	*W*
Block2D			
Sprite2D			
Vector			

15.4.5 Actions

move

Action Id	Mapping
1	Left
2	Up
3	Right
4	Down

initialize_is_tagged

Internal This action can only be called from other actions, not by the player.

Action Id	Mapping
1	Initialize Tagged
2	Initialize Not Tagged

tag

Action Id	Mapping
1	Left
2	Up
3	Right
4	Down

15.4.6 YAML

```yaml
Version: "0.1"
Environment:
  Name: Robot Tag 4v4
  Description: Robots start randomly as "tagged" or not, robots can "tag" other robots.␣
↪Any robot that is "tagged" 3 times dies.
  Observers:
    Block2D:
      TileSize: 24
    Sprite2D:
      TileSize: 24
      BackgroundTile: oryx/oryx_fantasy/floor1-1.png
    Vector:
      IncludePlayerId: true
      IncludeVariables: true
  Variables:
    - Name: player_done
      InitialValue: 0
      PerPlayer: true
    - Name: tagged_count
      InitialValue: 0
  Player:
    Count: 4
    Observer:
      RotateWithAvatar: true
      TrackAvatar: true
      Height: 9
      Width: 9
      OffsetX: 0
      OffsetY: 0
    AvatarObject: tagger
  Termination:
```

(continues on next page)

(continued from previous page)

```
    End:
      - eq: [ tagged_count, 0 ]

  Levels:
    - |
      W   W   W   W   W   W   W   W   W
      W   f1  .   .   .   .   .   f2  W
      W   .   .   .   .   .   .   .   W
      W   .   .   .   .   .   .   .   W
      W   .   .   .   .   .   .   .   W
      W   .   .   .   .   .   .   .   W
      W   .   .   .   .   .   .   .   W
      W   .   .   .   .   .   .   .   W
      W   f4  .   .   .   .   .   f3  W
      W   W   W   W   W   W   W   W   W
    - |
      W   W   W   W   W   W   W   W   W   W   W   W   W   W   W   W   W   W   W   W   W   W
      W   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   W
      W   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   W
      W   .   f1  .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   f2  .   .   W
      W   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   W
      W   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   W
      W   .   .   .   .   .   .   .   W   W   W   W   W   W   .   .   .   .   .   .   .   W
      W   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   W
      W   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   W
      W   .   .   .   .   .   .   W   .   .   .   .   .   .   W   .   .   .   .   .   .   W
      W   .   .   .   .   .   .   W   .   .   .   .   .   .   W   .   .   .   .   .   .   W
      W   .   .   .   .   .   .   W   .   .   .   .   .   .   W   .   .   .   .   .   .   W
      W   .   .   .   .   .   .   W   .   .   .   .   .   .   W   .   .   .   .   .   .   W
      W   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   W
      W   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   W
      W   .   .   .   .   .   .   .   W   W   W   W   W   W   .   .   .   .   .   .   .   W
      W   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   W
      W   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   W
```

(continues on next page)

(continued from previous page)

```
    W  .  .  f3 .  .  .  .  .  .  .  .  .  .  .  .  .  .  f4 .  .  W
    W  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  W
    W  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  W
    W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W
  - |
    W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W
    W  .  .  .  .  m  .  .  .  .  .  .  .  .  .  .  m  .  .  .  .  W
    W  .  .  .  .  m  .  .  .  .  .  .  .  .  .  .  m  .  .  .  .  W
    W  .  .  f1 .  m  .  .  .  .  .  .  .  .  .  .  m  .  f2 .  .  W
    W  .  .  .  .  m  .  .  .  .  .  .  .  .  .  .  m  .  .  .  .  W
    W  .  .  .  .  m  .  .  .  .  .  .  .  .  .  .  m  .  .  .  .  W
    W  m  m  m  m  m  .  .  W  W  W  W  W  W  .  .  m  m  m  m  m  W
    W  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  W
    W  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  W
    W  .  .  .  .  .  .  W  .  .  .  .  .  .  W  .  .  .  .  .  .  W
    W  .  .  .  .  .  .  W  .  .  .  .  .  .  W  .  .  .  .  .  .  W
    W  .  .  .  .  .  .  W  .  .  .  .  .  .  W  .  .  .  .  .  .  W
    W  .  .  .  .  .  .  W  .  .  .  .  .  .  W  .  .  .  .  .  .  W
    W  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  W
    W  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  W
    W  m  m  m  m  m  .  .  W  W  W  W  W  W  .  .  m  m  m  m  m  W
    W  .  .  .  .  m  .  .  .  .  .  .  .  .  .  .  m  .  .  .  .  W
    W  .  .  .  .  m  .  .  .  .  .  .  .  .  .  .  m  .  .  .  .  W
    W  .  .  f3 .  m  .  .  .  .  .  .  .  .  .  .  m  .  f4 .  .  W
    W  .  .  .  .  m  .  .  .  .  .  .  .  .  .  .  m  .  .  .  .  W
    W  .  .  .  .  m  .  .  .  .  .  .  .  .  .  .  m  .  .  .  .  W
    W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W  W
```

(continues on next page)

(continued from previous page)

```
- |
  W W W W W W W W W W W W W W W W W W W W W W W W W W W W W W W W W W W W W W W W
  W . . . . m . . . . . m . . m . . . . . . . . . . m . . m . . . . . m . . . . W
  W . . . . m . . . . . m . . m . . . . . . . . . . m . . m . . . . . m . . . . W
  W . . f1 . m . . . . . m . . m . . . . . . . . . . m . . m . . . . . m . f2 . . W
  W . . . . m . . . . . m . . m . . . . . . . . . . m . . m . . . . . m . . . . W
  W . . . . m . . . . . m . . m . . . . . . . . . . m . . m . . . . . m . . . . W
  W m m m m m . . . . . m . . m . . W W W W W W . . m . . m . . . . . m m m m m W
  W . . . . . . . . . . m . . m . . . . . . . . . . m . . m . . . . . . . . . . W
  W . . . . . . . . . . m . . m . . . . . . . . . . m . . m . . . . . . . . . . W
  W . . . . . . . . . . m . . m . . . . . . . . . . m . . m . . . . . . . . . . W
  W . . . . . . . . . . m . . m . . . . . . . . . . m . . m . . . . . . . . . . W
  W . . . . . . . . . . m . . m . . . . . . . . . . m . . m . . . . . . . . . . W
  W . . . . . . . . . . m . . m . . . . . . . . . . m . . m . . . . . . . . . . W
  W . . . . . . . . . . m . . m . . . . . . . . . . m . . m . . . . . . . . . . W
  W . . . . . . . . . . m . . m . . . . . . . . . . m . . m . . . . . . . . . . W
  W . . . . . . . . . . m . . m . . . . . . . . . . m . . m . . . . . . . . . . W
  W . . . . . . . . . . m . . m . . . . . . . . . . m . . m . . . . . . . . . . W
  W . . . . . . . . . . m . . m . . . . . . . . . . m . . m . . . . . . . . . . W
  W . . . . . . . . . . m . . m . . . . . . . . . . m . . m . . . . . . . . . . W
  W . . . . . . . . . . m . . m . W . . . . . . W . m . . m . . . . . . . . . . W
  W . . . . . . . . . . m . . m . W . . . . . . W . m . . m . . . . . . . . . . W
  W . . . . . . . . . . m . . m . W . . . . . . W . m . . m . . . . . . . . . . W
  W . . . . . . . . . . m . . m . W . . . . . . W . m . . m . . . . . . . . . . W
  W . . . . . . . . . . m . . m . . . . . . . . . . m . . m . . . . . . . . . . W
  W . . . . . . . . . . m . . m . . . . . . . . . . m . . m . . . . . . . . . . W
  W . . . . . . . . . . m . . m . . . . . . . . . . m . . m . . . . . . . . . . W
```

(continues on next page)

(continued from previous page)

```
      W   .   .   .   .   .   .   .   .   .   .   m   .   .   m   .   .   .   .   .   .   .   .   .   .   m   .   .   m   .   .   .   .   .   .   .   .   .   .   W
      W   .   .   .   .   .   .   .   .   .   .   m   .   .   m   .   .   .   .   .   .   .   .   .   .   m   .   .   m   .   .   .   .   .   .   .   .   .   .   W
      W   .   .   .   .   .   .   .   .   .   .   m   .   .   m   .   .   .   .   .   .   .   .   .   .   m   .   .   m   .   .   .   .   .   .   .   .   .   .   W
      W   .   .   .   .   .   .   .   .   .   .   m   .   .   m   .   .   .   .   .   .   .   .   .   .   m   .   .   m   .   .   .   .   .   .   .   .   .   .   W
      W   .   .   .   .   .   .   .   .   .   .   m   .   .   m   .   .   .   .   .   .   .   .   .   .   m   .   .   m   .   .   .   .   .   .   .   .   .   .   W
      W   .   .   .   .   .   .   .   .   .   .   m   .   .   m   .   .   .   .   .   .   .   .   .   .   m   .   .   m   .   .   .   .   .   .   .   .   .   .   W
      W   .   .   .   .   .   .   .   .   .   .   m   .   .   m   .   .   .   .   .   .   .   .   .   .   m   .   .   m   .   .   .   .   .   .   .   .   .   .   W
      W   .   .   .   .   .   .   .   .   .   .   m   .   .   m   .   .   .   .   .   .   .   .   .   .   m   .   .   m   .   .   .   .   .   .   .   .   .   .   W
      W   .   .   .   .   .   .   .   .   .   .   m   .   .   m   .   .   .   .   .   .   .   .   .   .   m   .   .   m   .   .   .   .   .   .   .   .   .   .   W
      W   .   .   .   .   .   .   .   .   .   .   m   .   .   m   .   .   .   .   .   .   .   .   .   .   m   .   .   m   .   .   .   .   .   .   .   .   .   .   W
      W   .   .   .   .   .   .   .   .   .   .   m   .   .   m   .   .   .   .   .   .   .   .   .   .   m   .   .   m   .   .   .   .   .   .   .   .   .   .   W
      W   .   .   .   .   .   .   .   .   .   .   m   .   .   m   .   .   .   .   .   .   .   .   .   .   m   .   .   m   .   .   .   .   .   .   .   .   .   .   W
      W   .   .   .   .   .   .   .   .   .   .   m   .   .   m   .   .   .   .   .   .   .   .   .   .   m   .   .   m   .   .   .   .   .   .   .   .   .   .   W
      W   m   m   m   m   m   .   .   .   .   .   m   .   .   m   .   .   W   W   W   W   W   W   .   .   m   .   .   m   .   .   .   .   .   m   m   m   m   m   W
      W   .   .   .   .   m   .   .   .   .   .   m   .   .   m   .   .   .   .   .   .   .   .   .   .   m   .   .   m   .   .   .   .   .   m   .   .   .   .   W
      W   .   .   .   .   m   .   .   .   .   .   m   .   .   m   .   .   .   .   .   .   .   .   .   .   m   .   .   m   .   .   .   .   .   m   .   .   .   .   W
      W   .   .   f3  .   m   .   .   .   .   .   m   .   .   m   .   .   .   .   .   .   .   .   .   .   m   .   .   m   .   .   .   .   .   m   .   f4  .   .   W
      W   .   .   .   .   m   .   .   .   .   .   m   .   .   m   .   .   .   .   .   .   .   .   .   .   m   .   .   m   .   .   .   .   .   m   .   .   .   .   W
      W   .   .   .   .   m   .   .   .   .   .   m   .   .   m   .   .   .   .   .   .   .   .   .   .   m   .   .   m   .   .   .   .   .   m   .   .   .   .   W
      W   W   W   W   W   W   W   W   W   W   W   W   W   W   W   W   W   W   W   W   W   W   W   W   W   W   W   W   W   W   W   W   W   W   W   W   W   W   W   W

Actions:

  # Taggers have a random chance of starting in a tagged state
  - Name: initialize_is_tagged
    InputMapping:
      Internal: true
      Inputs:
        1:
          Description: Initialize Tagged
        2:
```

(continues on next page)

(continued from previous page)

```
        Description: Initialize Not Tagged
        VectorToDest: [ -1, 0 ]

    Behaviours:
      - Src:
          Object: tagger
          Preconditions:
            - eq: [ src._playerId, dst._playerId ]
          Commands:
            - set_tile: 1
            - set: [ is_tagged, 1 ]
            - incr: tagged_count
        Dst:
          Object: tagger

  - Name: tag
    Behaviours:
      - Src:
          Object: tagger
          Preconditions:
            - eq: [ src.is_tagged, 1 ]
            - eq: [ dst.is_tagged, 0 ]
          Commands:
            - reward: 2
            - set_tile: 0
            - set: [ is_tagged, 0 ]
        Dst:
          Object: tagger
          Commands:
            - set_tile: 1
            - set: [ is_tagged, 1 ]
            - reward: -2
            - incr: times_tagged
            - eq:
                Arguments: [ times_tagged, 3 ]
                Commands:
                  - set: [ player_done, 1 ]
                  - decr: tagged_count
                  - reward: -5
                  - remove: true

  - Name: move
    Behaviours:
      - Src:
          Object: [ tagger, moveable_wall ]
          Commands:
            - mov: _dest
        Dst:
          Object: _empty

      - Src:
          Object: tagger
```

(continues on next page)

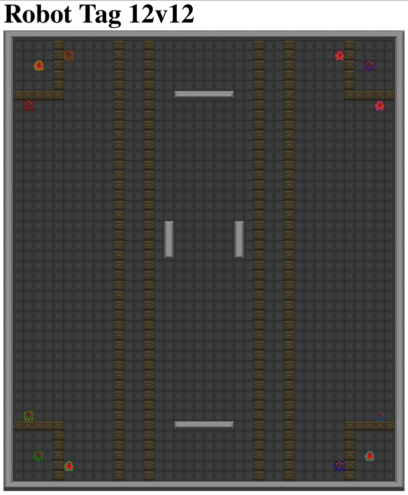
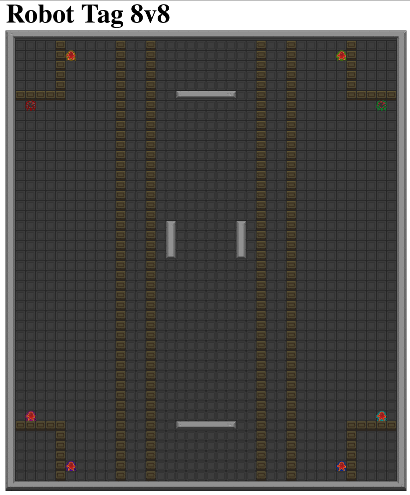
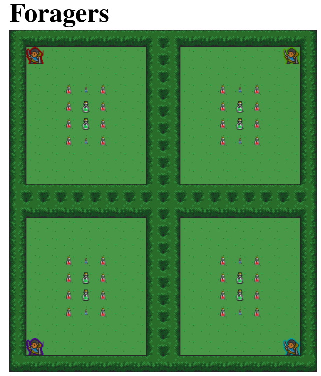
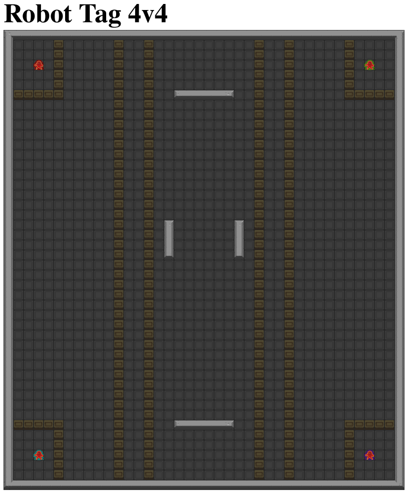
(continued from previous page)

```
          Commands:
            - mov: _dest
        Dst:
          Object: moveable_wall
          Commands:
            - cascade: _dest

Objects:
  - Name: tagger
    MapCharacter: f
    InitialActions:
      - Action: initialize_is_tagged
        Randomize: true
    Variables:
      - Name: is_tagged
        InitialValue: 0
      - Name: times_tagged
        InitialValue: 0
    Observers:
      Sprite2D:
        - Image: oryx/oryx_fantasy/avatars/robot1.png
        - Image: oryx/oryx_fantasy/avatars/fireguy1.png
      Block2D:
        - Shape: triangle
          Color: [ 0.2, 0.2, 0.9 ]
          Scale: 0.5
        - Shape: triangle
          Color: [ 0.9, 0.2, 0.2 ]
          Scale: 1.0

  - Name: moveable_wall
    MapCharacter: m
    Observers:
      Sprite2D:
        - Image: oryx/oryx_fantasy/wall4-0.png
      Block2D:
        - Color: [ 0.8, 0.8, 0.8 ]
          Shape: square

  - Name: fixed_wall
    MapCharacter: W
    Observers:
      Sprite2D:
        - TilingMode: WALL_16
          Image:
            - oryx/oryx_fantasy/wall2-0.png
            - oryx/oryx_fantasy/wall2-1.png
            - oryx/oryx_fantasy/wall2-2.png
            - oryx/oryx_fantasy/wall2-3.png
            - oryx/oryx_fantasy/wall2-4.png
            - oryx/oryx_fantasy/wall2-5.png
            - oryx/oryx_fantasy/wall2-6.png
```

(continues on next page)

(continued from previous page)

```
            - oryx/oryx_fantasy/wall2-7.png
            - oryx/oryx_fantasy/wall2-8.png
            - oryx/oryx_fantasy/wall2-9.png
            - oryx/oryx_fantasy/wall2-10.png
            - oryx/oryx_fantasy/wall2-11.png
            - oryx/oryx_fantasy/wall2-12.png
            - oryx/oryx_fantasy/wall2-13.png
            - oryx/oryx_fantasy/wall2-14.png
            - oryx/oryx_fantasy/wall2-15.png
      Block2D:
        - Color: [ 0.5, 0.5, 0.5 ]
          Shape: square
```

Robot Tag 12v12  Robots start randomly as "tagged" or not, robots can "tag" other robots. Any robot that is "tagged" 3 times dies.	**Robot Tag 8v8**  Robots start randomly as "tagged" or not, robots can "tag" other robots. Any robot that is "tagged" 3 times dies.	**Foragers**  A very simple multi-agent game. Agents must collect the coloured potions
Robot Tag 4v4  Robots start randomly as "tagged" or not, robots can "tag" other robots. Any robot that is "tagged" 3 times dies.		

CHAPTER

SIXTEEN

REINFORCEMENT LEARNING WITH RLLIB

Griddly provides support for reinforcement learning using the RLLib reinforcement learning library.

While RLLib doesn't support OpenAI Gym registered environments, it does provide a similar interface which is supported by Griddly's `RLLibEnv` environment.

Griddly provides two classes, `RLLibEnv` and `RLLibMultiAgentWrapper` which abstract away all the tedious parts of wrapping environments for RL and leaves you to concentrate on training algorithms, designing networks and game mechanics.

Examples for *single-agent* and *multi-agent* training are provided.

> **Warning:** All examples and networks are implemented using PyTorch. Some examples may be modified to work with Tensorflow, but we do not provide explicit support for Tensorflow.

16.1 Examples Setup

Griddly installs most of the dependencies for you automatically when it is installed, however you will need to install RLLlib and Pytorch to run the provided examples.

You can install RLLib and pytorch using the following command:

```
pip install ray[rllib]==2.1.0
```

16.2 Environment Parameters

Parameters for the environments, such as the *GDY* file for the game and *Observer options* can be sent to the environment using the `env_config` dictionary.

Most of the parameters here are the same as the parameters that can be given to the `gym.make()` command when creating a *Griddly environment for OpenAI Gym*.

```
'env_config': {
    'yaml_file': 'Single-Player/GVGAI/clusters_partially_observable.yaml',

    'global_observer_type': gd.ObserverType.SPRITE_2D,
    'record_video_config': {
        'frequency': 100000
    },
```

(continues on next page)

(continued from previous page)

```
    'random_level_on_reset': True,
    'max_steps': 1000,
},
```

The above example will also record a video of the environment (rendered using the SPRITE_2D renderer) for one episode every 100000 steps. Finally the max_steps of the environment will be override to be 1000 steps before the environment is reset automatically.

16.3 Level Randomization

Partially observable games have a fixed observations space regardless of the size of the levels. Additionally several games have levels of fixed size.

With these games, the level can be randomized at the end of every episode using the `random_level_on_reset` option in the `env_config` section of RLLib's config.

```
'env_config': {

    'random_level_on_reset': True,

    ...
```

If this is set to true then the agent will be placed in one of the random levels described in the GDY file each time the episode restarts.

16.4 Agents

We provide a few custom agent models that can be used with any Griddly environment.

16.4.1 Simple Convolutional agent

The simple convolutional agent stacks three convolutional layers that preserve the size of the input. After these layers the representation is flattened and linear layers are then used for the actor and critic heads.

To use `SimpleConvAgent`, register the custom model with RLLib and then use it in your training `config`:

```
ModelCatalog.register_custom_model('SimpleConv', SimpleConvAgent)

...

config = {

    'model': {
        'custom_model': 'SimpleConv'
        'custom_model_config': .....
    }

    ...
```

(continues on next page)

(continued from previous page)

```
}
```

SimpleConvAgent

```
class SimpleConvAgent(TorchModelV2, nn.Module):
"""
Simple Convolution agent that calculates the required linear output layer
"""

    def __init__(self, obs_space, action_space, num_outputs, model_config, name):
        super().__init__(obs_space, action_space, num_outputs, model_config, name)
        nn.Module.__init__(self)

        self._num_objects = obs_space.shape[2]
        self._num_actions = num_outputs

        linear_flatten = np.prod(obs_space.shape[:2])*64

        self.network = nn.Sequential(
            layer_init(nn.Conv2d(self._num_objects, 32, 3, padding=1)),
            nn.ReLU(),
            layer_init(nn.Conv2d(32, 64, 3, padding=1)),
            nn.ReLU(),
            nn.Flatten(),
            layer_init(nn.Linear(linear_flatten, 1024)),
            nn.ReLU(),
            layer_init(nn.Linear(1024, 512)),
            nn.ReLU(),
        )

        self._actor_head = nn.Sequential(
            layer_init(nn.Linear(512, 256), std=0.01),
            nn.ReLU(),
            layer_init(nn.Linear(256, self._num_actions), std=0.01)
        )

        self._critic_head = nn.Sequential(
            layer_init(nn.Linear(512, 1), std=0.01)
        )

    def forward(self, input_dict, state, seq_lens):
        obs_transformed = input_dict['obs'].permute(0, 3, 1, 2)
        network_output = self.network(obs_transformed)
        value = self._critic_head(network_output)
        self._value = value.reshape(-1)
        logits = self._actor_head(network_output)
        return logits, state

    def value_function(self):
        return self._value
```

16.4.2 Global Average Pooling

Griddly environments' observation spaces can differ between games, levels and visualization options. In order to handle this in a generic way using neural networks, we provide a Global Average Pooling agent *GAPAgent*, which can be used with any 2D environment with no additional configuration.

All you need to do is register the custom model with RLLib and then use it in your training `config`:

```
ModelCatalog.register_custom_model('GAP', GAPAgent)

...

config = {

    'model': {
        'custom_model': 'GAP'
        'custom_model_config': .....
    }

    ...

}
```

GAPAgent

```
class GAPAgent(TorchModelV2, nn.Module):
"""
Global Average Pooling Agent
This is the same agent used in https://arxiv.org/abs/2011.06363.

Global average pooling is a simple way to allow training grid-world environments␣
↪regardless o the size of the grid.
"""

def __init__(self, obs_space, action_space, num_outputs, model_config, name):
    super().__init__(obs_space, action_space, num_outputs, model_config, name)
    nn.Module.__init__(self)

    self._num_objects = obs_space.shape[2]

    self._num_actions = num_outputs

    self.network = nn.Sequential(
        layer_init(nn.Conv2d(self._num_objects, 32, 3, padding=1)),
        nn.ReLU(),
        layer_init(nn.Conv2d(32, 64, 3, padding=1)),
        nn.ReLU(),
        GlobalAvePool(2048),
        layer_init(nn.Linear(2048, 1024)),
        nn.ReLU(),
        layer_init(nn.Linear(1024, 512)),
        nn.ReLU(),
```

(continues on next page)

(continued from previous page)

```
    )

    self._actor_head = nn.Sequential(
        layer_init(nn.Linear(512, 256), std=0.01),
        nn.ReLU(),
        layer_init(nn.Linear(256, self._num_actions), std=0.01)
    )

    self._critic_head = nn.Sequential(
        layer_init(nn.Linear(512, 1), std=0.01)
    )

def forward(self, input_dict, state, seq_lens):
    obs_transformed = input_dict['obs'].permute(0, 3, 1, 2)
    network_output = self.network(obs_transformed)
    value = self._critic_head(network_output)
    self._value = value.reshape(-1)
    logits = self._actor_head(network_output)
    return logits, state

def value_function(self):
    return self._value
```

See also:

You can read more about agents that use Global Average Pooling here: https://arxiv.org/abs/2005.11247

16.5 Recording Videos

Griddly can automatically record videos during training by placing the `record_video_config` dictionary into the standard RLLib `env_config`.

```
'env_config':
    'record_video_config': {
        'frequency': 20000
        'directory': '/home/griddlyuser/my_experiment_videos'
        'include_global': True,
        'include_agents': False,
    },

    ...
}
```

Warning: the `directory` value must be an absolute path, as the working directory of workers is controlled by Ray.

Videos can be recorded from the perspective of the agent and the perspective of the global observer. `include_global` and `include_agents` will set which videos are recorded.

See also:

For more information on how to configure observers see *Observation Spaces*

The triggering of videos is configured using the `frequency` variable. The `frequency` variable refers to the number of steps in each environment that pass before the recording is triggered.

Once triggered, the next episode is recorded in full. Videos of episodes are recorded on the first environment in every worker in RLLib.

16.5.1 Uploading Videos to WandB

To automatically upload videos to WandB, the `VideoCallback` can be set in the RLLib config:

```
'config': {
    ...,

    'callbacks': VideoCallback,

    ...
}
```

16.6 Recording Environment Actions

Fig. 1: An example of logged events for each agent in an environment during training. Can help to diagnose problems with reward shaping and track exploration.

Griddly's RLLib integration hooks into the *Event History* and records all the frequency of the actions that are being taken by agents during training. This event history can then be picked up in the agent's `info` in RLLib's callback methods, e,g `on_episode_step`

```
'env_config':
     'record_actions': True,

    ...
}
```

16.6.1 Uploading Environment Events to WandB

To automatically upload action events to WandB, the `ActionTrackerCallback` can be set in the RLLib config:

```
'config': {
    ...,

    'callbacks': ActionTrackerCallback,

    ...
}
```

CHAPTER

SEVENTEEN

SINGLE AGENT GAMES

The Griddly RLLibEnv wrapper allows any of the single-agent games to be trained with many of the single agent RLLib Algorithms.

```
register_env('my-single-agent-environment', RLlibEnv)
```

17.1 Full Example

The example below uses PPO to train on the "GridMan" Environment.

The agent in the "GridMan" environment has a 7x7 partially observable ego-centric view.

By default the agent sees a *VECTOR* view of the environment. This view is passed to a *Simple Conv Agent* to produce the policy.

See also:

To use a different game, or specific level, just change the `yaml_file` or set a `level` parameter in the `env_config`. Other options can be found *here*

```
import os
import sys

from griddly.util.rllib.callbacks import VideoCallbacks
from griddly.util.rllib.environment.core import RLlibEnv
from ray.air.callbacks.wandb import WandbLoggerCallback
from ray.rllib.algorithms.ppo import PPOConfig
from ray.rllib.models import ModelCatalog
from ray.tune import register_env, tune

from rllib_single_agent_example.gap_agent import GAPAgent
from rllib_single_agent_example.simple_conv_agent import SimpleConvAgent

# You have to put this here so that rllib can find griddly libraries when it starts new␣
↪workers
sep = os.pathsep
os.environ["PYTHONPATH"] = sep.join(sys.path)

environment_name = "GridMan"
environment_yaml = "gridman/gridman.yaml"
model_name = "SimpleConvAgent"
```

(continues on next page)

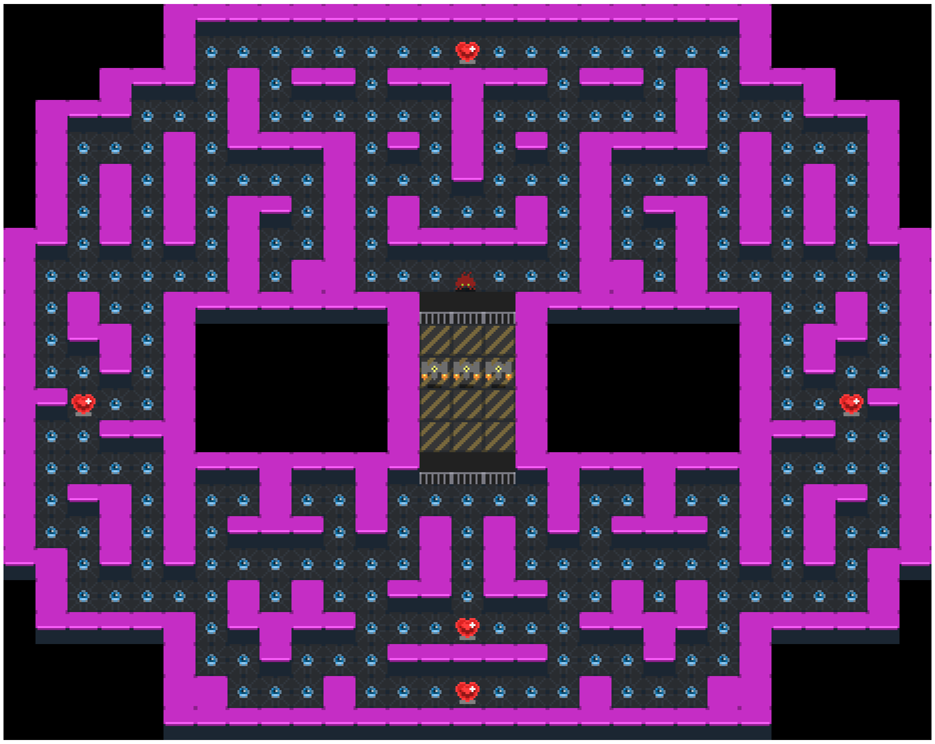


Fig. 1: The GridMan environment as seen from the "Global Observer" view.

(continued from previous page)

```
# Register the environment with RLlib
register_env(environment_name, lambda config: RLlibEnv(config))

model_class = None
if model_name == "SimpleConvAgent":
    model_class = SimpleConvAgent
elif model_name == "GlobalAveragePoolingAgent":
    model_class = GAPAgent

# Register the model with RLlib
ModelCatalog.register_custom_model(model_name, model_class)

test_dir = f"./results/{environment_name}"
video_dir = f"videos"

config = (
    PPOConfig()
    .rollouts(num_rollout_workers=8, num_envs_per_worker=16, rollout_fragment_length=128)
    .callbacks(VideoCallbacks)
    .training(
        model={
            "custom_model": model_name
        },
        train_batch_size=16384,
```

(continues on next page)

(continued from previous page)

```
        lr=1e-4,
        gamma=0.95,
        lambda_=0.9,
        use_gae=True,
        clip_param=0.4,
        grad_clip=None,
        entropy_coeff=0.1,
        vf_loss_coeff=0.5,
        sgd_minibatch_size=2048,
        num_sgd_iter=4,
    )
    .environment(
        env_config={
            # A video every 50 iterations
            'record_video_config': {
                'frequency': 1000,
                'directory': video_dir,

                # Will record a video of the global observations
                'include_global': True,

                # Will record a video of the agent's perspective
                'include_agents': False,
            },
            'random_level_on_reset': True,
            'yaml_file': environment_yaml,
            'global_observer_type': "GlobalSpriteObserver",
            'player_observer_type': "Vector",
            'max_steps': 2000,
        },
        env=environment_name, clip_actions=True)
    .debugging(log_level="ERROR")
    .framework(framework="torch")
    .resources(num_gpus=int(os.environ.get("RLLIB_NUM_GPUS", "1")))
)

result = tune.run(
    "PPO",
    name="PPO",
    stop={"timesteps_total": 10000000},
    local_dir=test_dir,
    config=config.to_dict(),
    callbacks=[
        WandbLoggerCallback(project="RLLib Gridman", group="griddlyai")
    ]
)
```

17.2 Github Repository

You can find a full working example here: https://github.com/GriddlyAI/rllib_single_agent_example

CHAPTER

EIGHTEEN

MULTI AGENT

Griddly automatically wraps multi-agent games for compatibility with RLLib using the *RLLibMultiAgentWrapper*.

To register the multi-agent Griddly environment for usage with RLLib, the environment can be wrapped in the following way:

```
# Create the environment and wrap it in a multi-agent wrapper for self-play
register_env(environment_name, lambda config: RLlibMultiAgentWrapper(RLlibEnv(config)))
```

18.1 Handling agent done

If a multi-agent environment has the conditions in which agents in the environment can be removed, for example they are defeated and are not longer in the episode, a RLLib needs to know that this agent no longer can receive actions.

Griddly's `RLlibMultiAgentWrapper` handles this by detecting a `player_done_variable`, defined per-player in the GDY. When this variable is set to 1 for a player, RLLib will consider this player has been removed.

18.2 Full Example

In this example we use a multi-agent version of the "GridMan" environment, but we train both "Gridman" and the agents that are chasing him!

Gridman has a 9x9 observation space and the chasers only have a 7x7 observation space, which gives them a disadvantage. However there are three of them!

See also:

To use a different game, or specific level, just change the `yaml_file` or set a `level` parameter in the `env_config`. Other options can be found *here*

```
import os
import sys

import gym
from griddly.util.rllib.callbacks import VideoCallbacks
from griddly.util.rllib.environment.core import RLlibEnv, RLlibMultiAgentWrapper
from ray.air.callbacks.wandb import WandbLoggerCallback
from ray.rllib.algorithms.ppo import PPOConfig
from ray.rllib.models import ModelCatalog
from ray.rllib.policy.policy import PolicySpec
```

(continues on next page)

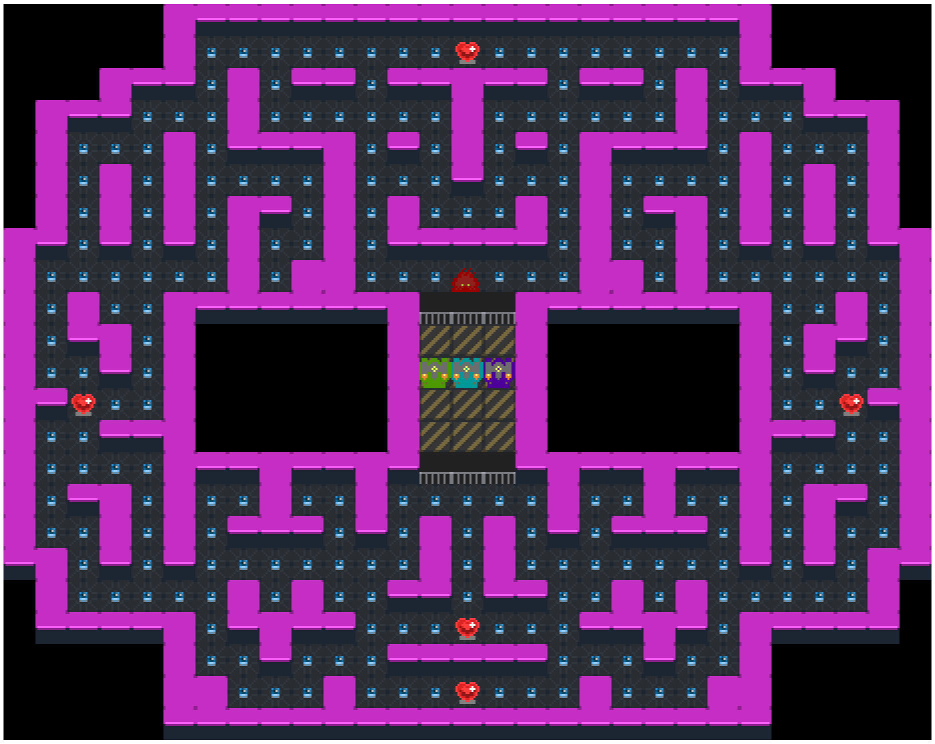


Fig. 1: The GridMan environment as seen from the "Global Observer" view.

(continued from previous page)

```
from ray.tune import register_env, tune

from rllib_multi_agent_example.gap_agent import GAPAgent
from rllib_multi_agent_example.simple_conv_agent import SimpleConvAgent

# You have to put this here so that rllib can find griddly libraries when it starts new␣
↪workers
sep = os.pathsep
os.environ["PYTHONPATH"] = sep.join(sys.path)

environment_name = "GridmanMultiAgent"
environment_yaml = "gridman/gridman_multiagent.yaml"
model_name = "SimpleConvAgent"

# Register the environment with RLlib
register_env(environment_name, lambda config: RLlibMultiAgentWrapper(RLlibEnv(config)))

model_class = None
if model_name == "SimpleConvAgent":
    model_class = SimpleConvAgent
elif model_name == "GlobalAveragePoolingAgent":
    model_class = GAPAgent

# Register the model with RLlib
ModelCatalog.register_custom_model(model_name, model_class)
```

(continues on next page)

(continued from previous page)

```
test_dir = f"./results/{environment_name}"
video_dir = f"videos"

# multi-agent policies
policies = {
    # Use the PolicySpec namedtuple to specify an individual policy:
    "gridman_policy": PolicySpec(
        observation_space=gym.spaces.Box(0, 255, (9, 9, 11), float),
        config={"gamma": 0.95},
    ),
    "enemy_policy": PolicySpec(
        observation_space=gym.spaces.Box(0, 255, (7, 7, 11), float),
        config={"gamma": 0.95},
    ),
}


def policy_mapping_fn(agent_id, episode, worker, **kwargs):
    if agent_id == 1:
        return "gridman_policy"
    else:
        return "enemy_policy"


config = (
    PPOConfig()
    .rollouts(num_rollout_workers=8, num_envs_per_worker=16, rollout_fragment_length=128)
    .callbacks(VideoCallbacks)
    .multi_agent(policies=policies, policy_mapping_fn=policy_mapping_fn)
    .training(
        model={
            "custom_model": model_name
        },
        train_batch_size=16384,
        lr=1e-4,
        gamma=0.95,
        lambda_=0.9,
        use_gae=True,
        clip_param=0.4,
        grad_clip=None,
        entropy_coeff=0.1,
        vf_loss_coeff=0.5,
        sgd_minibatch_size=2048,
        num_sgd_iter=4,
    )
    .environment(
        env_config={
            # A video every 50 iterations
            'record_video_config': {
                'fps': 20,
                'frequency': 1000,
```

(continues on next page)

(continued from previous page)

```
                'directory': video_dir,

                # Will record a video of the global observations
                'include_global': True,

                # Will record a video of the agent's perspective
                'include_agents': False,
            },
            'random_level_on_reset': True,
            'yaml_file': environment_yaml,
            'global_observer_type': "GlobalSpriteObserver",
            'player_observer_type': ["VectorGridMan", "VectorEnemy", "VectorEnemy",
↪"VectorEnemy"],
            'max_steps': 2000,
        },
        env=environment_name, clip_actions=True)
    .debugging(log_level="ERROR")
    .framework(framework="torch")
    .resources(num_gpus=int(os.environ.get("RLLIB_NUM_GPUS", "1")))
)

result = tune.run(
    "PPO",
    name="PPO",
    stop={"timesteps_total": 10000000},
    local_dir=test_dir,
    config=config.to_dict(),
    callbacks=[
        WandbLoggerCallback(project="RLLib Gridman MultiAgent", entity="griddlyai")
    ]
)
```

18.3 Github Repository

You can find a full working example here: https://github.com/GriddlyAI/rllib_multi_agent_self_play_example

CHAPTER

NINETEEN

GDY TUTORIAL - MAKING SOKOBAN

This tutorial will take you through building the game Sokoban using Griddly's Game Description Yaml (GDY).

The tutorial comes in three stages covering the main areas of the GDY configuration file `Enviroment`, `Actions` and `Objects`.

```
Version: 0.1
Enviroment: ...

Actions: ...

Objects: ...
```

Choose from the following options to learn how to configure each section:

Enviroment - Define how the player (or players) interact with the environment and design the levels.

Actions - Define the mechanics of the environment. This is how the different objects interact with one another.

Objects - Define all the objects that might exist and how they will be rendered on screen.

Finally you can load the game in python and play it:

–> *How to play the tutorial game*

19.1 Objects

In this section we define the objects in the game and what they will look like when being rendered.

In the game Sokoban we control a player `avatar` which moves around and pushes `boxes` into `holes`, there are `walls` that we cannot push and we cannot pass.

This means we have 4 objects in our simple Sokoban game. So we need to define them!

19.1.1 Step 1 - Avatar object

Each object we have to give a unique name to. That unique name can then be used to reference that object in other parts of the GDY configuration. In for the avatar object we will call it `avatar` unsurprisingly.

We're going to use this image:

```yaml
Objects:

  - Name: avatar
    Z: 2
    MapCharacter: A
    Observers:
      Sprite2D:
        Image: images/gvgai/oryx/knight1.png
```

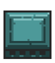
`MapCharacter` defines the character we will use to describe initial positions in the levels defined in the `Environment` section of the GDY file.

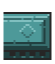
`Z` allows us to define that objects can occupy the same location in the grid, as long as they have different Z indexes. It also defines the rendering order when rendering the game. Higher Z-indexes mean the objects will be rendered on top.

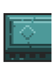
The `Observers` block defines how each observer type will render this particular object. We are defining a `Sprite2D` observer here so we need to supply an image to it.

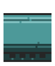
19.1.2 Step 2 - Wall Objects

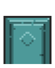
Wall objects are slightly more complicated because when they are rendered they actually use 15 different images, for example corner peices, T-peices etc. . . .











lets look at the object definition for walls!

```yaml
- Name: wall
  MapCharacter: w
  Observers:
    Sprite2D:
      TilingMode: WALL_16
      Image:
        - images/gvgai/oryx/wall3_0.png
        - images/gvgai/oryx/wall3_1.png
        - images/gvgai/oryx/wall3_2.png
        - images/gvgai/oryx/wall3_3.png
        - images/gvgai/oryx/wall3_4.png
        - images/gvgai/oryx/wall3_5.png
        - images/gvgai/oryx/wall3_6.png
        - images/gvgai/oryx/wall3_7.png
        - images/gvgai/oryx/wall3_8.png
        - images/gvgai/oryx/wall3_9.png
        - images/gvgai/oryx/wall3_10.png
        - images/gvgai/oryx/wall3_11.png
        - images/gvgai/oryx/wall3_12.png
        - images/gvgai/oryx/wall3_13.png
        - images/gvgai/oryx/wall3_14.png
        - images/gvgai/oryx/wall3_15.png
```

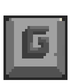
Here we define the `Name` and the `MapCharacter` like we did when the avatar was defined. We dont need a Z index because nothing interacts with the `wall` objects.

In the `Sprite2D` Object there is now a `TilingMode` object which can either be `WALL_2`, or `WALL_16`. These tiling modes use 2 or 16 images respectively to render the walls in the game environment. The order of the walls is important to render the walls correctly.

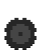
19.1.3 Step 3 - Boxes and holes

Boxes and holes are very similar to `avatar` objects. The only difference is that `hole` objects have a different `Z` value which allows the `avatar` object to move on top of them.

`box`:



`hole`:



```yaml
- Name: box
  Z: 2
  MapCharacter: b
  Observers:
    Sprite2D:
      Image: images/gvgai/newset/block1.png

- Name: hole
  Z: 1
  MapCharacter: h
  Observers:
    Sprite2D:
      Image: images/gvgai/oryx/cspell4.png
```

19.1.4 Putting it all together

Thats it! We've defined our objects and some properties about how they will look in the game.

The completed `Object` section of our GDY file looks like this:

```yaml
Objects:
 - Name: box
   Z: 2
   MapCharacter: b
   Observers:
     Sprite2D:
       Image: images/gvgai/newset/block1.png

 - Name: wall
   MapCharacter: w
   Observers:
     Sprite2D:
       TilingMode: WALL_16
       Image:
         - images/gvgai/oryx/wall3_0.png
         - images/gvgai/oryx/wall3_1.png
         - images/gvgai/oryx/wall3_2.png
         - images/gvgai/oryx/wall3_3.png
         - images/gvgai/oryx/wall3_4.png
         - images/gvgai/oryx/wall3_5.png
         - images/gvgai/oryx/wall3_6.png
         - images/gvgai/oryx/wall3_7.png
         - images/gvgai/oryx/wall3_8.png
         - images/gvgai/oryx/wall3_9.png
         - images/gvgai/oryx/wall3_10.png
         - images/gvgai/oryx/wall3_11.png
         - images/gvgai/oryx/wall3_12.png
         - images/gvgai/oryx/wall3_13.png
         - images/gvgai/oryx/wall3_14.png
         - images/gvgai/oryx/wall3_15.png

 - Name: hole
```

(continues on next page)

(continued from previous page)

```
  Z: 1
  MapCharacter: h
  Observers:
    Sprite2D:
      Image: images/gvgai/oryx/cspell4.png

- Name: avatar
  Z: 2
  MapCharacter: A
  Observers:
    Sprite2D:
      Image: images/gvgai/oryx/knight1.png
```

19.2 Actions

Actions are the "mechanics" of any Griddly game.

A single action defines what happens between two objects (or sets of objects) in an environment.

The **source** of an action is the object which perfoms a particular action. The **destination** of an action is the object that is affected by the action.

Lets look at a few examples to make these ideas more concrete!

19.2.1 Step 1 - Movement

We are building the game "Sokoban" so we will first define that the `avatar` object can move around in empty space. To do that we can define a "move" action as follows:

```
Actions:
 # Define the move action
 - Name: move
   Behaviours:
   # The agent can move around freely in empty space and over holes
     - Src:
         Object: avatar
         Commands:
           - mov: _dest
       Dst:
         Object: _empty
```

We have named the above action "move" and defined a single behaviour. The behaviour object contains the `Src` key with the `Object` value `avatar` meaning that we are defing what happens if the `avatar` object performs the "move" action. We also define the `Dst` key with the `Object` value `_empty`. The `_empty` keyword is a special object that refers to "empty space". This action therefore is only executed when the `avatar` performs an action on an `_empty` space.

Finally we have `Commands` object in the `Src` key. The `Commands` object contains a list of instructions that will be executed by the `Src` object. The command we have here is *mov: _dest* which tells the environment to move the object to the destination of the action. The *_dest* keyword is another special keyword used in actions which contains the location of the destination of the action.

For more information about possible commands that can be run on either the Src or Dst objects you can refer to the schema docs *here*

19.2.2 Step 2 - Pushing boxes

To define that we want box objects to move when the avatar object moves into them we can add the following code to our Behaviours list:

```
# Boxes can move into empty space
- Src:
    Object: box
    Commands:
        - mov: _dest
  Dst:
    Object: _empty

# The agent can push boxes
- Src:
    Object: avatar
    Commands:
        - mov: _dest
  Dst:
    Object: box
    Commands:
        - cascade: _dest
```

Here we are actually defining two behaviours. The first behaviour is similar to the one in the previous example. We define that the box object has the mechanic allowing it to move into empty space.

The second behaviour we define allows the avatar object to interact with the box object. The mov: _dest command tells avatar to move to the destination location when the action is executed. The box object also needs to be moved in the same direction as the avatar. This can be achieved by applying the same "move" action again, but on the destination object. cascade: _dest re-applies (or cascades) the same action on the destination object, which will move the box!

Note: We have only allowed the box object to "move" into empty space. If the _dest location is not empty, i.e. it contains a wall object or a hole object, the command will not be executed. This will stop also the avatar from moving.

19.2.3 Step 3 - Pushing a box into a hole

Now we can push boxes around in empty space, but we have no defined what will happen if we push the box into a hole object. We want to reward the player and also remove the box object.

```
# If a box is moved into a hole remove it
     - Src:
         Object: box
         Commands:
           - remove: true
           - reward: 1
```

(continues on next page)

(continued from previous page)

```
      Dst:
        Object: hole
```

Most of this behaviour is hopefully self-explanatory by this point. There are two new commands introduced here `reward: 1` which gives the player a reward of value 1 and `remove: true` which removes the *Src* object.

19.2.4 Putting It All Together

In order to put all these actions together, there is only one minor change to make to the first behaviour. We need to add `hole` to the `Dst` objects. This adds the ability for avatars to be able to walk on top of `hole` objects!

So the entire actions section of the game **Sokoban** looks like this:

```
Actions:
# Define the move action
- Name: move
  Behaviours:
    # The agent can move around freely in empty space and over holes
    - Src:
        Object: avatar
        Commands:
          - mov: _dest
      Dst:
        Object: [_empty, hole]

    # Boxes can move into empty space
    - Src:
        Object: box
        Commands:
            - mov: _dest
      Dst:
        Object: _empty

    # The agent can push boxes
    - Src:
        Object: avatar
        Commands:
          - mov: _dest
      Dst:
        Object: box
        Commands:
          - cascade: _dest

    # If a box is moved into a hole remove it
    - Src:
        Object: box
        Commands:
          - remove: true
          - reward: 1
      Dst:
        Object: hole
```

19.3 Environment

The `Environment` configuration contains the definition of how the game will be controlled by the player, the conditions underwhich the game ends, the configurations of observers and the levels that are contained in the game.

19.3.1 Step 1 - Player Configuration

The `Player` configuration

```
Player:
  AvatarObject: avatar
```

19.3.2 Step 2 - Termination Conditions

Termination conditions are the rules which decide when the game episodes are complete, and whether the agent wins or loses.

Rules can be defined seperately for `Win` and `Lose`. In Sokoban, the player "wins" by pushing all of the boxes into holes. This is how it is defined in GDY.

```
Termination:
    Win:
      - eq: [box:count, 0]
```

We define here that the agent wins in the case that the number of `box` objects in the environment reaches 0. The `:count` attribute can be appended to any object name to return the number of those objects.

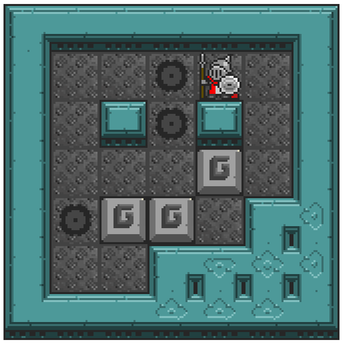
19.3.3 Step 3 - Levels

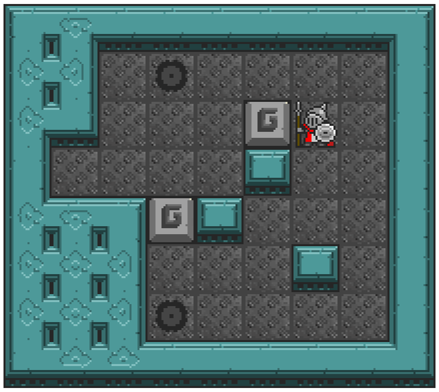
In our Sokoban game we are going to define two levels. The layout of each level is defined by a level string which is made up of `MapCharacter` characters that are defined in the *Objects* section of this tutorial.

the dot `.` character means that the space in the map in unoccupied.

```
Levels:
    - |
      wwwwwww
      w..hA.w
      w.whw.w
      w...b.w
      whbb.ww
      w..wwww
      wwwwwww
    - |
      wwwwwwwww
      ww.h....w
      ww...bA.w
      w....w..w
      wwwbw...w
      www...w.w
      wwwh....w
      wwwwwwwww
```

the two defined levels will look like this when rendered:





19.3.4 Step 4 - TileSize and Background Image

```
Observers:
  Sprite2D:
    TileSize: 24
    BackgroundTile: gvgai/newset/floor2.png
  Block2D:
    TileSize: 24
```

Here we specify the size of the tiles in pixels `TileSize` for both the Sprite2D and Block2D observers. Also if we want to use an image for the background when there are no objects present we can supply a `BackgroundTile` image.

19.3.5 Putting it all together

The environment definition with all the parts described looks like this:

```
Environment:
   Name: sokoban
   Observers:
   Sprite2D:
     TileSize: 24
     BackgroundTile: gvgai/newset/floor2.png
   Block2D:
     TileSize: 24
   Player:
     AvatarObject: avatar
```

(continues on next page)

(continued from previous page)

```
  Termination:
    Win:
      - eq: [box:count, 0] # If there are no boxes left
  Levels:
    - |
      wwwwwww
      w..hA.w
      w.whw.w
      w...b.w
      whbb.ww
      w..wwww
      wwwwwww
    - |
      wwwwwwwww
      ww.h....w
      ww...bA.w
      w....w..w
      wwwbw...w
      www...w.w
      wwwh....w
      wwwwwwwww
```

19.4 Playing Griddly Games

In this short tutorial you will learn how to load a GDY file, convert it to an OpenAI Gym interface and then use the OpenAI Gym interface to play the game with the w,a,s,d keys on your keyboard.

19.4.1 Step 1 - Imports

To play games with the keyboard using the gym interface, the `play` function can be used to wrap a gym environment

The only griddly import that is required is the `GymWrapperFactory`, this is used to create gym wrappers for any Griddly environments

```
import gym
from gym.utils.play import play

from griddly import GymWrapperFactory
```

19.4.2 Step 2 - Load the GDY

the `build_gym_from_yaml` builds the Griddly environment from the GDY file and loads a particular level.

This can then be loaded by OpenAI gym's `make` command. The name of the environment will be `GDY-[your environment name]-v0`. In this case the environment name will be `GDY-Sokoban-v0`

```
# This is what to use if you want to use OpenAI gym environments
wrapper = GymWrapperFactory()
```

(continues on next page)

(continued from previous page)

```
wrapper.build_gym_from_yaml('SokobanTutorial', 'sokoban.yaml', level=0)
```

19.4.3 Step 3 - Play

All thats left is to play the game!

```
# Create the Environment
env = gym.make(f'GDY-SokobanTutorial-v0')

# Play the game
play(env, fps=10, zoom=2)
```

CHAPTER

TWENTY

GDY SCHEMA TUTORIAL

This tutorial will show you how to set up your IDE to help write GDY files.

The GDY Schema defines all the required and optional properties that can be used to create games. Find your IDE below and follow the instructions to set up the schema validator!

You can either download the schema locally or reference it from the github repository

20.1 Visual Studio Code

- Install the `redhat.vscode-yaml` extension from the visual studio code marketplace.

This extension allows json schema files to be used to add syntax support for yaml files.

- Edit the your workspace settings to add the following keys:

```
{
 "[yaml]":{
     "editor.insertSpaces": true,
     "editor.tabSize": 2,
     "editor.quickSuggestions": {
       "other": true,
       "comments": false,
       "strings": true
     },
     "editor.autoIndent": "none",
   },
   "yaml.schemas": {
     "gdy-schema.json": "[path to your gdy files]/*.yaml"
   }
 }
```

20.2 PyCharm

PyCharm has a feature for YAML validation with JSON schema files built in.

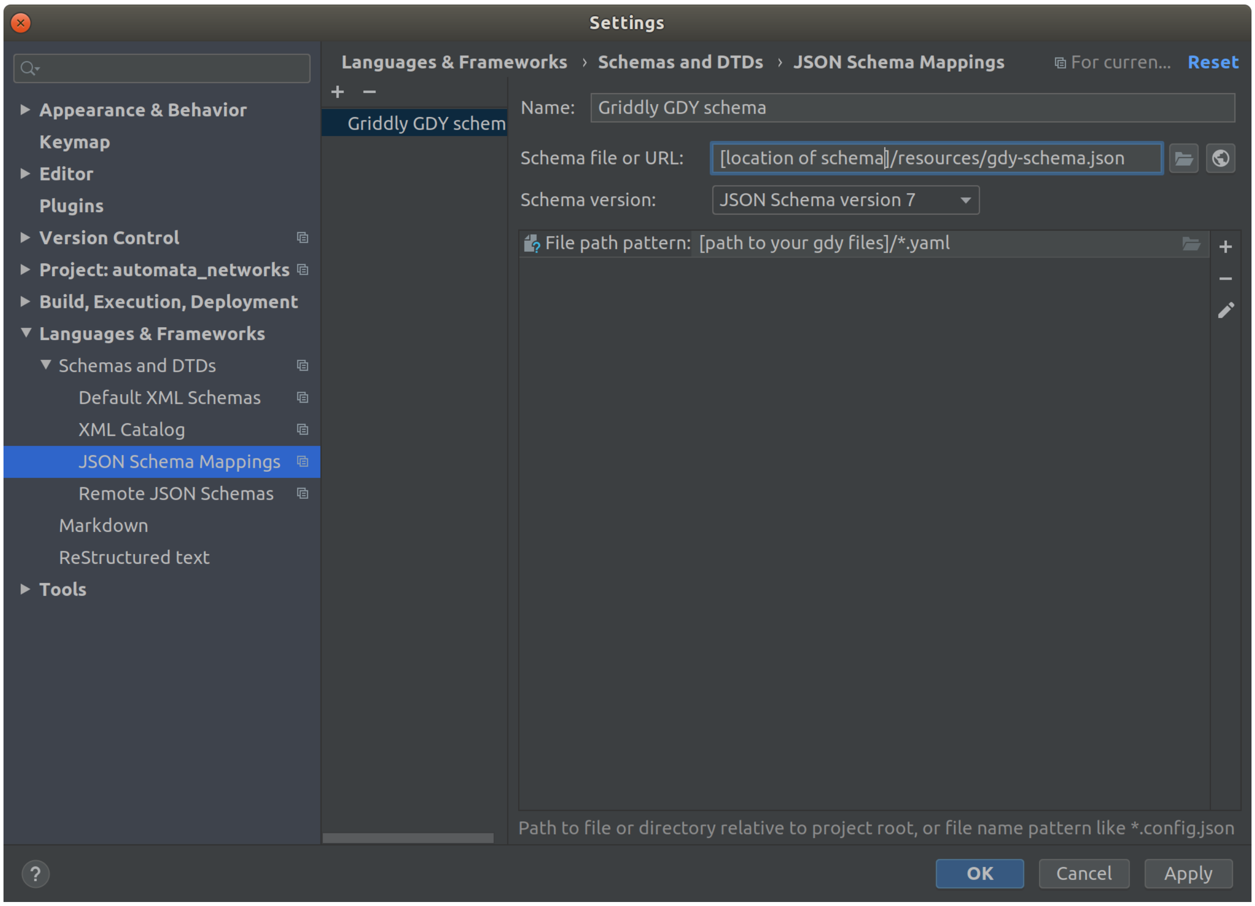
In settings navigate to `Languages And Frameworks -> Schemas and DTDs -> JSON Schema Mappings`

You will need to the schema location to either the schema location on github

`https://raw.githubusercontent.com/Bam4d/Griddly/develop/resources/gdy-schema.json`

or download the schema file locally and point to it on your local machine.

You will also need to set a *Filepath Pattern* to point to the location you are storing your GDY YAML files.



CHAPTER

TWENTYONE

PROXIMITY TRIGGERS

An important mechanic in many environments is to allow events to happen when objects are within certain ranges.

In this example we will create an environment where if the agent (a spider) gets close to lava it may catch fire, and if it goes near water it will douse the flames! Poor spider.

21.1 Step 1 - Create the lava, water and spider objects

```yaml
- Name: spider
  Variables:
    - Name: on_fire
      InitialValue: 0
  MapCharacter: s
  Observers:
    Isometric:
      - Image: oryx/oryx_iso_dungeon/avatars/spider-1.png
      - Image: oryx/oryx_iso_dungeon/avatars/spider-fire-1.png
    Block2D:
      - Shape: triangle
        Color: [ 0.2, 0.2, 0.9 ]
        Scale: 0.5
      - Shape: triangle
        Color: [ 0.9, 0.2, 0.2 ]
        Scale: 1.0

- Name: lava
  MapCharacter: l
  Observers:
    Isometric:
      - Image: oryx/oryx_iso_dungeon/lava-1.png
        Offset: [0, 4]
    Block2D:
      - Color: [ 0.8, 0.0, 0.0 ]
        Shape: square

- Name: water
  MapCharacter: w
  Observers:
    Isometric:
      - Image: oryx/oryx_iso_dungeon/water-1.png
```

(continues on next page)

(continued from previous page)

```
        Offset: [0, 4]
    Block2D:
      - Color: [ 0.0, 0.0, 0.8 ]
        Shape: square
```

21.2 Step 2 - Set up the proximity trigger for lava

For the lava, we want the spider to be catch fire instantly if it is next to the lava, but have a small chance of catching fire if it is close, but not right next to it.

We can achieve this by using two `RANGE_BOX_BOUNDARY` triggers. With `RANGE_BOX_BOUNDARY` triggers, only objects that are at a specific distance away can cause the action to trigger.

Therefor using two `RANGE_BOX_BOUNDARY` triggers, one with `Range: 1` and one with `Range: 2` we can produce the desired effect.

Additionally you can set a `Probability` for an action to set how likely the action is to be executed.

Note: Action Probabilities can be used on any action, not just those with triggers

```
- Name: set_spider_on_fire_close
  Probability: 1.0
  Trigger:
    Type: RANGE_BOX_BOUNDARY
    Range: 1
  Behaviours:
    - Src:
        Object: lava
      Dst:
        Object: spider
        Commands:
          - set_tile: 1
          - set: [ on_fire, 1 ]


- Name: set_spider_on_fire
  Probability: 0.1
  Trigger:
    Type: RANGE_BOX_BOUNDARY
    Range: 2
  Behaviours:
    - Src:
        Object: lava
      Dst:
        Object: spider
        Commands:
          - set_tile: 1
          - set: [ on_fire, 1 ]
```

21.3 Step 3 - Set up the proximity trigger for water

The following action uses a `RANGE_BOX_AREA` with `` `Range: `` 2 meaning that anything within a box that is 2 blocks away from the water activates this action

The action has a `Probability` of 0.1 of being executed.

```
- Name: douse_spider
  Probability: 0.1
  Trigger:
    Type: RANGE_BOX_AREA
    Range: 2
  Behaviours:
    - Src:
        Object: water
      Dst:
        Object: spider
        Commands:
          - set_tile: 0
          - set: [ on_fire, 0 ]
```

21.4 Full GDY Description

There's a bit more boiler plate to fill out … but otherwise proximity triggers are that simple!

See also:

for more information about the boilerplate for GDY files please see *this tutorial on GDY files*

```
Version: "0.1"
Environment:
  Name: SpiderFire
  Description: Just an example, not a real environment, also not real lava, the spider␣
↪is real though... too real.
  Observers:
    Block2D:
      TileSize: 24
    Isometric:
      TileSize: [ 32, 48 ]
      IsoTileHeight: 16
      IsoTileDepth: 4
      BackgroundTile: oryx/oryx_iso_dungeon/grass-1.png
    Vector:
      IncludePlayerId: true
  Player:
    AvatarObject: spider
  Levels:
    - |
      W  W  W  W  W  W  .  .  .  .  .  .  .  .  .  .
      W  W  W  W  W  W  .  .  .  .  .  .  .  .  .  .
      W  W  .  .  .  .  .  .  .  .  .  .  .  .  .  .
      W  W  .  .  .  .  .  .  S  .  .  .  .  .  .  .
```

(continues on next page)

(continued from previous page)

```
    w  w  .  .  .  .  .  .  .  .  .  .  .  .  l  l
    w  w  .  .  .  .  .  .  .  .  .  .  .  .  l  l
    .  .  .  .  .  .  .  .  .  .  .  .  l  l  l  l
    .  .  .  .  .  .  .  .  .  .  .  .  l  l  l  l

Actions:

  - Name: set_spider_on_fire_close
    Probability: 1.0
    Trigger:
      Type: RANGE_BOX_BOUNDARY
      Range: 1
    Behaviours:
      - Src:
          Object: lava
        Dst:
          Object: spider
          Commands:
            - set_tile: 1
            - set: [ on_fire, 1 ]


  - Name: set_spider_on_fire
    Probability: 0.1
    Trigger:
      Type: RANGE_BOX_BOUNDARY
      Range: 2
    Behaviours:
      - Src:
          Object: lava
        Dst:
          Object: spider
          Commands:
            - set_tile: 1
            - set: [ on_fire, 1 ]

  - Name: douse_spider
    Probability: 0.1
    Trigger:
      Type: RANGE_BOX_AREA
      Range: 2
    Behaviours:
      - Src:
          Object: water
        Dst:
          Object: spider
          Commands:
            - set_tile: 0
            - set: [ on_fire, 0 ]


  - Name: move
```

(continues on next page)

(continued from previous page)

```
    Behaviours:
      - Src:
          Object: spider
          Commands:
            - mov: _dest
        Dst:
          Object: _empty

Objects:
  - Name: spider
    Variables:
      - Name: on_fire
        InitialValue: 0
    MapCharacter: s
    Observers:
      Isometric:
        - Image: oryx/oryx_iso_dungeon/avatars/spider-1.png
        - Image: oryx/oryx_iso_dungeon/avatars/spider-fire-1.png
      Block2D:
        - Shape: triangle
          Color: [ 0.2, 0.2, 0.9 ]
          Scale: 0.5
        - Shape: triangle
          Color: [ 0.9, 0.2, 0.2 ]
          Scale: 1.0

  - Name: lava
    MapCharacter: l
    Observers:
      Isometric:
        - Image: oryx/oryx_iso_dungeon/lava-1.png
          Offset: [0, 4]
      Block2D:
        - Color: [ 0.8, 0.0, 0.0 ]
          Shape: square

  - Name: water
    MapCharacter: w
    Observers:
      Isometric:
        - Image: oryx/oryx_iso_dungeon/water-1.png
          Offset: [0, 4]
      Block2D:
        - Color: [ 0.0, 0.0, 0.8 ]
          Shape: square
```

CHAPTER

TWENTYTWO

CUSTOM SHADERS

GDY provides an easy way of scripting most game mechanics. But what if you want to do something interesting with the visualization of the environment?

Griddly achieves high-speed rendering using hardware accelerated SPIR-V shaders.

Shaders are complicated beasts, but the following tutorials will help to understand how Griddly uses them. We will also show how you can customize them to produce much more visually complex environments.

This tutorial is not intended to teach how computer graphics pipelines work, you will probably require a basic grasp of several topics. I've included some good resources here that should be helpful:

Model View Projection Matrices

Shaders Basics (this is based around the Unity engine, but many of the concepts are similar.

Healthbars, Signed Distance Fields & Lighting (again around Unity, but we use Signed Distance Fields in Example 2 to make a health bar!

22.1 Griddly Graphics Pipeline Basics

Griddly's rendering pipeline uses a single *Vertex Shader* and a single *Fragment Shader*. The entire game state (global variables, object locations, object variable values etc.) is sent to both the vertex and fragment shader. This allows the shader to read the states and render the environment accordingly.

Griddly has default shaders for both `SPRITE_2D` and `BLOCK_2D` renderers which are automatically included when you install Griddly. The code for those shaders (specifically for when you use the `SPRITE_2D`) is shown below:

22.1.1 Shader Memory Layout

The default vertex shader is slightly more complicated than the fragment shader, as we include all of the possible `uniform buffer objects` and `storage buffer objects` that are accessible from the shader.

Note: All uniforms are available to the vertex and fragment shader.

We use a **Push Constants** to index the object in the `ObjectData` that we are currently rendering on the GPU. Each ObjectData contains all the information of a single object as defined in the GDY, such as `color`, `textureIdx`, `objectType` and position (`modelMatrix`). We also have a persistent `EnvironmentData` object which includes global information about the environment, including the `playerId` that the shader is constructing the observation for. We also have access to information about the other players in the `PlayerInfo` buffer. (Currently this is limited to the automatically generated `playerColor`).

Finally we have the `ObjectVariableBuffer` and `GlobalVariableBuffer` which contain the data for **object variables** and **global variables** (as defined in the GDY files under the Shader options).

The full shader layout is shown below.:

```
struct GlobalVariable {
  int value;
};

struct PlayerInfo {
  vec4 playerColor;
  vec4 playerObservableGrid;
};

struct ObjectData {
  mat4 modelMatrix;
  vec4 color;
  vec4 gridPosition;
  vec2 textureMultiply;
  int textureIndex;
  int objectType;
  int playerId;
};

layout(std140, binding = 1) uniform EnvironmentData {
  mat4 projectionMatrix;
  mat4 viewMatrix;
  vec4 globalObserverAvatarHighlightColor;
  vec2 gridDims;
  int playerCount;
  int playerId;
  int globalVariableCount;
  int objectVariableCount;
  int globalObserverAvatarMode;
  int highlightPlayers;
}
environmentData;

layout(std430, binding = 2) readonly buffer PlayerInfoBuffer {
  PlayerInfo variables[];
}
playerInfoBuffer;

layout(std430, binding = 3) readonly buffer ObjectDataBuffer {
  uint size;
  ObjectData variables[];
}
objectDataBuffer;

layout(std430, binding = 4) readonly buffer GlobalVariableBuffer {
  GlobalVariable variables[];
}
globalVariableBuffer;
```

(continues on next page)

(continued from previous page)

```
layout(std430, binding = 5) readonly buffer ObjectVariableBuffer {
  ObjectVariable variables[];
}
objectVariableBuffer;

layout(push_constant) uniform PushConsts {
  int idx;
}
pushConsts;
```

22.1.2 Default Vertex Shader

The vertex shader is mostly very simple, it takes the object defined by the current `pushConsts.idx` and applies the *model view projection* matrix of that object to each vertex.

The sprite image for the object is stored in a *texture array* indexed by `object.textureIndex`. This is used to generate the fragment coordinates that are sent through to the fragment shader in the `outFragTextureCoords` variable.

Additionally, in the default vertex shader, we allow players to be highlighted by color. This can be seen in many of the multi-agent and RTS games. The player color is stored in the `outPlayerColor` variable, which is also sent to the fragment shader.

```
void main() {
  ObjectData object = objectDataBuffer.variables[pushConsts.idx];
  PlayerInfo objectPlayerInfo = playerInfoBuffer.variables[object.playerId - 1];

  outFragTextureCoords = vec3(
      inFragTextureCoords.x * object.textureMultiply.x,
      inFragTextureCoords.y * object.textureMultiply.y,
      object.textureIndex);

  mat4 mvp = environmentData.projectionMatrix * environmentData.viewMatrix * object.
↪modelMatrix;

  gl_Position = mvp * vec4(
                         inPosition.x,
                         inPosition.y,
                         inPosition.z,
                         1.);

  if (environmentData.highlightPlayers == 1) {
    if (object.playerId > 0 && object.playerId == environmentData.playerId) {
      outPlayerColor = vec4(0.0, 1.0, 0.0, 1.0);
    } else {
      outPlayerColor = objectPlayerInfo.playerColor;
    }

    outHighlightPlayers = 1;
  } else {
    outHighlightPlayers = 0;
    outPlayerColor = vec4(0.0);
```

(continues on next page)

(continued from previous page)

```
  }
}
```

22.1.3 Default Fragment Shader

Most customization for visualization in Griddly environments is undertaken in the fragment shader.

The default fragment shader samples the color of each pixel from the texture, given the texture coordinates from the vertex shader.

In this default shader we also keep player highlighting code which adds an outline to the sprite image based on it's transparency.

```
void main()
{
    if(highlightPlayers==1){
        // Just multiply by the alpha channel of the object
        vec4 color=texture(samplerArray,inFragTextureCoords);

        vec2 tex_dims=vec2(textureSize(samplerArray,0));

        vec2 pixel_size=2./tex_dims;

        vec4 colorU=texture(samplerArray,vec3(inFragTextureCoords.x,max(pixel_size.y,
↪inFragTextureCoords.y-pixel_size.y),inFragTextureCoords.z));
        vec4 colorD=texture(samplerArray,vec3(inFragTextureCoords.x,min(tex_dims.y,
↪inFragTextureCoords.y+pixel_size.y),inFragTextureCoords.z));
        vec4 colorL=texture(samplerArray,vec3(min(tex_dims.x,inFragTextureCoords.x+pixel_
↪size.x),inFragTextureCoords.y,inFragTextureCoords.z));
        vec4 colorR=texture(samplerArray,vec3(max(0.,inFragTextureCoords.x-pixel_size.x),
↪inFragTextureCoords.y,inFragTextureCoords.z));

        outFragColor=color;

        float thresh1=.7;
        float thresh2=.4;

        if(color.a<=thresh1&&(colorU.a>thresh2||colorD.a>thresh2||colorL.a>
↪thresh2||colorR.a>thresh2)){
            outFragColor=playerColor;
        }

    }else{
        outFragColor=texture(samplerArray,inFragTextureCoords);
    }

}
```

22.2 Customising Shaders

In order to customize these shaders, firstly the custom shaders have to be coded and compiled and secondly we have to point Griddly to these compiled shaders.

To point Griddly to a new shader directory, the `shader_path` parameter needs to be supplied in the `gym.make` or `GymWrapper` function:

```
env = GymWrapper('object_lighting.yaml',
                 shader_path='shaders',
                 player_observer_type=gd.ObserverType.SPRITE_2D,
                 global_observer_type=gd.ObserverType.SPRITE_2D,
                 ...
               )
```

22.2.1 Compiling shaders with glslc

There are many shader languages that can be compiled into the SPIR-V format that Griddly is compatible with. In all the examples, and default shaders we use the GLSL shader language and compile it to SPIR-V using `glslc`

glslc can be obtained by installing the Vulkan SDK or by forking the github repo and compiling from scratch.

Once you have the glslc tool, you can compile the fragment and vertex shaders using the following commands:

```
glslc triangle-textured.frag -o $SHADER_OUTPUT_DIR/triangle-textured.frag.spv
glslc triangle-textured.vert -o $SHADER_OUTPUT_DIR/triangle-textured.vert.spv
```

Note: For SPRITE_2D and ISOMETRIC shaders, the compiled fragment and vertex shaders must be named `triangle-textured.frag.spv` and `triangle-textured.vert.spv`. For BLOCK_2D, the shaders need to be named `triangle.frag.spv` and `triangle.vert.spv`.

22.3 Examples

22.3.1 Global Lighting

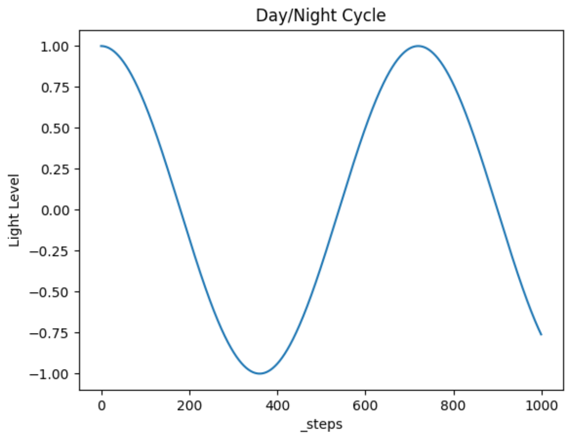
In this tutorial we modify the *Bufferflies* environment to have a day/night cycle. In environments that use pixel-based observations this can provide an additional challenge to agents, as the agent has to encode the pixel representations of the envrionment in a way that allows for many different lighting levels.

Naive Day/Night Cycle

We want the value of the pixels to cycle from black (night) to their *actual* (daylight) values. We can do this by multiplying the RGB values by a **light level** value L that oscillates (with respect to `_steps` s) between 0 and 1.

We could just use:

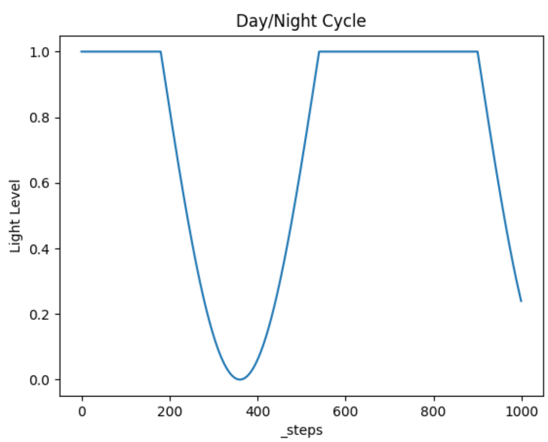
$$L = \cos\left(\frac{\pi s}{360}\right)$$



but this would mean that we would only reach full daylight and complete darkness once every 360 steps. What we actually want is light for maybe 50% of the time and then a short-ish night.

For this we can use:

$$L = \min(1, \cos\left(\frac{\pi s}{360}\right) + 1)$$



Now we have an algorithm for calculating the light level coefficient, how do we apply this in our custom shaders?

Using global variables in shaders

To implement our day/night cycle using our L value, we first have to understand a few features that are available in the Griddly engine.

How does Griddly handle "time"?

Griddly has a built-in step counter called `_steps` which can be accessed as a global variable. This value is equal to the number of steps that have passed in a particular episode.

How can we pass the `_steps` parameter to the shader?

The `_steps` parameter is automatically passed to the shader by default. Other global variables can be passed to the shader by specifying it in the GDY Shader options. An example of this is shown below.

```
Observers:
  Sprite2D:
    Shader:
    # The _steps variable is automatically exported to all shaders, so we cannot add it␣
↪here,
    # but this is how we would add custom global variables
      GlobalVariables: [global_variable1, global_variable2]
```

How can we read the `_steps` parameter and use it?

Global variables specified in the `Shader` configuration of the GDY file, are sent to the shader in the same order that they are specified in the GDY. However the `_steps` variable is also included by default in position `0`.

In the shader, we can use the `GlobalVariable` uniform buffer:

```
layout(std430, binding = 4) readonly buffer GlobalVariableBuffer {
  GlobalVariable variables[];
}
globalVariableBuffer;

# globalVariableBuffer.variables[0] # this is the value of _steps
```

Now we know how to access the variable in the shader, how can we customize the shaders to modify the pixel values to what we want?

Note: more information on compiling custom shaders and using them in Griddly envs can be found *here*

Global Lighting Shaders

Vertex

Most of the code in the vertex shader is standard code required for drawing the observation.

We calculate the lighting level in the vertex shader (so we don't need to calculate it for every pixel) and pass it to the fragment shader using `outLightLevel`

```
#version 460

layout(location = 0) in vec3 inPosition;
```

(continues on next page)

(continued from previous page)

```
layout(location = 1) in vec2 inFragTextureCoords;

layout(location = 0) out vec4 outLightLevel;
layout(location = 1) out vec3 outFragTextureCoords;

out gl_PerVertex {
  vec4 gl_Position;
};

struct GlobalVariable {
  int value;
};

struct ObjectVariable {
  int value;
};

struct PlayerInfo {
  vec4 playerColor;
  vec4 playerObservableGrid;
};

struct ObjectData {
  mat4 modelMatrix;
  vec4 color;
  vec4 gridPosition;
  vec2 textureMultiply;
  int textureIndex;
  int objectType;
  int playerId;
};

layout(std140, binding = 1) uniform EnvironmentData {
  mat4 projectionMatrix;
  mat4 viewMatrix;
  vec4 globalObserverAvatarHighlightColor;
  vec2 gridDims;
  int playerCount;
  int playerId;
  int globalVariableCount;
  int objectVariableCount;
  int globalObserverAvatarMode;
  int highlightPlayers;
}
environmentData;

layout(std430, binding = 2) readonly buffer PlayerInfoBuffer {
  PlayerInfo variables[];
}
playerInfoBuffer;

layout(std430, binding = 3) readonly buffer ObjectDataBuffer {
```

(continues on next page)

(continued from previous page)

```
  uint size;
  ObjectData variables[];
}
objectDataBuffer;

layout(std430, binding = 4) readonly buffer GlobalVariableBuffer {
  GlobalVariable variables[];
}
globalVariableBuffer;

layout(std430, binding = 5) readonly buffer ObjectVariableBuffer {
  ObjectVariable variables[];
}
objectVariableBuffer;

layout(push_constant) uniform PushConsts {
  int idx;
}
pushConsts;

#define PI 3.1415926538

void main() {
  ObjectData object = objectDataBuffer.variables[pushConsts.idx];

  float steps = float(globalVariableBuffer.variables[0].value);

  // 360 steps is roughly 1 day
  float lightLevel = clamp(cos(PI*steps/360)+1.0, 0.0, 1.0);
  outLightLevel = vec4(lightLevel,lightLevel,lightLevel,1.0);

  outFragTextureCoords = vec3(
      inFragTextureCoords.x * object.textureMultiply.x,
      inFragTextureCoords.y * object.textureMultiply.y,
      object.textureIndex);

  mat4 mvp = environmentData.projectionMatrix * environmentData.viewMatrix * object.
↪modelMatrix;

  gl_Position = mvp * vec4(
                          inPosition.x,
                          inPosition.y,
                          inPosition.z,
                          1.);
}
```

Fragment

In the fragment shader, we simply multiply our `inLightLevel` which is passed from the vertex shader by the texture fragment color (this is our RGB values).

Note that here we have also removed the code for highlighting the players thats present in the default shader as we don't need it for this environment.

```
#version 460

layout(binding = 0) uniform sampler2DArray samplerArray;

layout(location = 0) in vec4 inLightLevel;
layout(location = 1) in vec3 inFragTextureCoords;

layout(location = 0) out vec4 outFragColor;

void main() {
  outFragColor = texture(samplerArray, inFragTextureCoords) * inLightLevel;
}
```

Full Code Example

Full code examples can be found here!

22.3.2 Health Bars

We create a simple 4v4 environment where agents control a *robot*. Each robot has a health value and can attacks the other robots in the environment. The aim of this environment is for the robots to defeat the otehr robots.

In this tutorial we use a custom shader and signed distance fields to draw health bars.

Calculating Health Bars

Let's say each *robot* unit has a current health value `health` and a maximum health value `max_health`.

We can set the `max_health` and initial `health` value in the GDY for the *robot* units like this:

```
- Name: robot
  MapCharacter: f
  Variables:
    - Name: health
      InitialValue: 10
    - Name: max_health
      InitialValue: 10
```

We can then expose these variables to the shader by passing them in the `Shader` config for the `Sprite2D` renderer.

```
Sprite2D:
   Shader:
     ObjectVariables: [ health, max_health ]
```

These variables can then be accessed in the shader using the following helper function.

```
int getObjectVariable(in int objectIndex, in int variableIndex, in int numVariables) {
    return objectVariableBuffer.variables[objectIndex*numVariables+variableIndex].value;
}
```

`objectIndex` will be the value of the current object index being drawn. This value is sent to the shader as the *push constant* `idx`. `variableIndex` is the index of the variable in the `ObjectVariables`, for example, 0 for `health` and 1 for `max_health`. Finally `numVariables` is the number of variables that is provided in the `ObjectVariables` configuration. In our case this is 2.

We can retrieve the `health` and `max_health` values in the shader by calling this function with the following arguments:

```
int health = getObjectVariable(pushConsts.idx, 0, environmentData.objectVariableCount);
int maxHealth = getObjectVariable(pushConsts.idx, 1, environmentData.
↪objectVariableCount);

if(object.objectType == 2) {
 outNormalizedHealth = float(health)/float(maxHealth);
} else {
   outNormalizedHealth = -1.0;
}
```

We also normalize the health value for `robot` objects, but set it to -1.0 for all other objects. We normalize this value here because then it makes it very simple to adjust the initial and maximum health values without changing and recompiling the shader. It also means that this code can be re-used in environments with many objects that have many objects with different health values. In the fragment shader, we ignore drawing health bars when the normalized health value is less than 0.

Note: `object.objectType` here is the index of *alphabetically* sorted object names. The ordered list of object names can also returned by calling `env.get_object_names()`.

We dont want to calculate this for every pixel, as this is inefficient. So we put this code into the *vertex* shader and then send this value to the *fragment* shader.

Drawing Health Bars with Signed Distance Fields

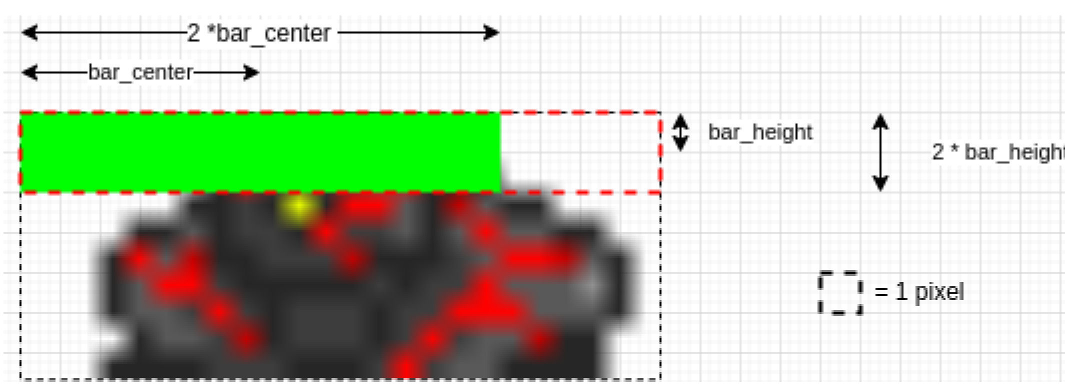


Fig. 1: `bar_center` and `bar_height` are used to calculate whether or not to change the colour of a pixel in the fragment shader. We calculate if the pixel is *within* the health bar and change its color accordingly.

The fragment shader operates on every pixel that we are drawing. The coordinates of the pixel of the *texture* that we are currenty drawing are given by `inFragTextureCoords`. This means we can override the current pixel color with health bar if the pixel itself is *in* the area that we want to health bar to occupy.

Our health bar always has a constant height, so we can check that the current pixel is at the **top** of the texture using the following code:

```
vec2 tex_dims=vec2(textureSize(samplerArray,0));
float bar_height=1.0/tex_dims.y;

bool isAtTop = distance(inFragTextureCoords.y,bar_height)<bar_height;
```

Notice that we also have to normalize the height value by the texture dimensions, as the texture coordinates in `inFragTextureCoords` are also normalized.

This gives us a health bar of size two pixels, because the `distance` function will effectively check 1 pixel above and below the `bar_height` (1-1=0 to 1+1=2).

We can calculate the width of the bar by calculating the *horizontal center* of where we want the bar and then checking if we are close enough to this point to be part of the "health bar".

We know that when health = 1.0 (full normalized health), we want the horizontal center to be at 0.5. In this example we also align the health bar to the left. This conveniently means we can just set `bar_center_x` to be equal to `inNormalizedHealth/2.0`.

```
float bar_center_x=inNormalizedHealth/2.0;

bool isCloseToHorizontalCenter = distance(inFragTextureCoords.x,bar_center_x)<bar_center_
↪x;
```

We can tie these two `x` and `y` distance fields together into a single check, and then we can change the color of the bar based on the value of `inNormalizedHealth`. We also want to only do this check if the `inNormalizedHealth` value is larger than 0. As we mentioned before, we set `inNormalizedHealth = -1` if there is no health bar to be rendered.

```
if(inNormalizedHealth>0){
    vec2 tex_dims=vec2(textureSize(samplerArray,0));
    float bar_height=1.0/tex_dims.y;
    float bar_center_x=inNormalizedHealth/2.0;

    if(distance(inFragTextureCoords.x,bar_center_x)<bar_center_x&&
↪distance(inFragTextureCoords.y,bar_height)<bar_height){
        if(inNormalizedHealth > 0.5) {
            outFragColor=vec4(0.0,1.0,0.,1.);
        } else if(inNormalizedHealth > 0.25) {
            outFragColor=vec4(1.0,1.0,0.,1.);
        } else {
            outFragColor=vec4(1.0,0.0,0.,1.);
        }
        isHealthBar = true;
    }
}
```

Health Bar Shaders

We can now tie all of this together in our vertex and fragment shaders!

Vertex

Again in our vertex shader we have the standard boiler plate code which gives us access to the variables from the Griddly Engine. How this ties in with the explanation of the normalized health calculations can be seen in full here:

```
#version 460

layout(location = 0) in vec3 inPosition;
layout(location = 1) in vec2 inFragTextureCoords;

layout(location = 0) out float outNormalizedHealth;
layout(location = 1) out vec3 outFragTextureCoords;

out gl_PerVertex {
  vec4 gl_Position;
};

struct GlobalVariable {
  int value;
};

struct ObjectVariable {
  int value;
};

struct PlayerInfo {
 vec4 playerColor;
 vec4 playerObservableGrid;
};

struct ObjectData {
  mat4 modelMatrix;
  vec4 color;
  vec4 gridPosition;
  vec2 textureMultiply;
  int textureIndex;
  int objectType;
  int playerId;
};

layout(std140, binding = 1) uniform EnvironmentData {
  mat4 projectionMatrix;
  mat4 viewMatrix;
  vec4 globalObserverAvatarHighlightColor;
  vec2 gridDims;
  int playerCount;
  int playerId;
  int globalVariableCount;
```

(continues on next page)

(continued from previous page)

```
  int objectVariableCount;
  int globalObserverAvatarMode;
  int highlightPlayers;
}
environmentData;

layout(std430, binding = 2) readonly buffer PlayerInfoBuffer {
  PlayerInfo variables[];
}
playerInfoBuffer;

layout(std430, binding = 3) readonly buffer ObjectDataBuffer {
  uint size;
  ObjectData variables[];
}
objectDataBuffer;

layout(std430, binding = 4) readonly buffer GlobalVariableBuffer {
  GlobalVariable variables[];
}
globalVariableBuffer;

layout(std430, binding = 5) readonly buffer ObjectVariableBuffer {
  ObjectVariable variables[];
}
objectVariableBuffer;

layout(push_constant) uniform PushConsts {
  int idx;
}
pushConsts;

int getObjectVariable(in int objectIndex, in int variableIndex, in int numVariables) {
  return objectVariableBuffer.variables[objectIndex*numVariables+variableIndex].value;
}

void main() {
  ObjectData object = objectDataBuffer.variables[pushConsts.idx];

  int health = getObjectVariable(pushConsts.idx, 0, environmentData.objectVariableCount);
  int maxHealth = getObjectVariable(pushConsts.idx, 1, environmentData.
↪objectVariableCount);

  if(object.objectType == 2) {
    outNormalizedHealth = float(health)/float(maxHealth);
  } else {
    outNormalizedHealth = -1.0;
  }

  PlayerInfo objectPlayerInfo = playerInfoBuffer.variables[object.playerId - 1];

  outFragTextureCoords = vec3(
```

(continues on next page)

(continued from previous page)

```
        inFragTextureCoords.x * object.textureMultiply.x,
        inFragTextureCoords.y * object.textureMultiply.y,
        object.textureIndex);

  mat4 mvp = environmentData.projectionMatrix * environmentData.viewMatrix * object.
↪modelMatrix;

  gl_Position = mvp * vec4(
                        inPosition.x,
                        inPosition.y,
                        inPosition.z,
                        1.);

}
```

Fragment

The entire Fragment shader can be seen here. Notice also that we check if the pixel being drawn is a health bar or not, and if it is *not* we just sample from the texture array to get the pixel for the sprite image.

```
#version 460

layout(binding = 0) uniform sampler2DArray samplerArray;

layout(location = 0) in float inNormalizedHealth;
layout(location = 1) in vec3 inFragTextureCoords;

layout(location = 0) out vec4 outFragColor;

void main() {
   bool isHealthBar = false;
   // Draw health bar at the top of the sprite using distance fields
   if(inNormalizedHealth>0){
         vec2 tex_dims=vec2(textureSize(samplerArray,0));
         float bar_height=1.0/tex_dims.y;
         float bar_center_x=inNormalizedHealth/2.0;

         if(distance(inFragTextureCoords.x,bar_center_x)<bar_center_x&&
↪distance(inFragTextureCoords.y,bar_height)<bar_height){
            if(inNormalizedHealth > 0.5) {
               outFragColor=vec4(0.0,1.0,0.,1.);
            } else if(inNormalizedHealth > 0.25) {
               outFragColor=vec4(1.0,1.0,0.,1.);
            } else {
               outFragColor=vec4(1.0,0.0,0.,1.);
            }
            isHealthBar = true;
         }
   }

   if (!isHealthBar) {
```

(continues on next page)

(continued from previous page)

```
        outFragColor=texture(samplerArray,inFragTextureCoords);
    }
}
```

Full Code Example

Full code examples can be found here!

22.3.3 Object Lighting

In this tutorial we will learn how to create a custom shader that performs per-object lighting.

We modify the *Partially Observable Labyrinth* environment to be completely dark apart from around the goal state and the agent itself.

Also, because this environment is partially observable, the agent itself will see observations like below:

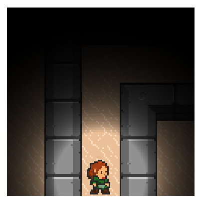


Fig. 2: Partially observable view of agent using the object shaders.

Lighting Individual Objects

Vertex shaders operate on each vertex (each corner of a sprite image), and fragment shaders operate on the individual pixels. To get the desired lighting effects, we don't need to do any calculations in the vertex shader.

In the Fragment shader however, we need to calculate how close we are to objects that emit light. Then we need to calculate how *bright* a pixel is.

Light Emitting Objects

we can define light emitting objects using object variables in the GDY:

For the avatar object:

```yaml
Name: avatar
MapCharacter: A
Variables:
  - Name: is_light
    InitialValue: 1
```

For the exit object:

```yaml
Name: exit
MapCharacter: x
Variables:
  - Name: is_light
    InitialValue: 1
```

We can then expose these variables to the shader by passing them in the `Shader` config for the `Sprite2D` renderer.

```yaml
Sprite2D:
  Shader:
    ObjectVariables: [ is_light ]
```

In the shader, we can access object variables using the following helper function:

```
int getObjectVariable(in int objectIndex, in int variableIndex, in int numVariables) {
    return objectVariableBuffer.variables[objectIndex*numVariables+variableIndex].value;
}
```

`objectIndex` will be the value of the current object index being drawn. This value is sent to the shader as the *push constant* `idx`. `variableIndex` is the index of the variable in the `ObjectVariables`, for example, 0 for `is_light`. If we required other variables, they would be available at subsequenct indexes. Finally `numVariables` is the number of variables that is provided in the `ObjectVariables` configuration. In our case this is 1 as we only define `is_light`.

To retrieve the `is_light` value for any object i, we can call the helper function with the following arguments:

```
int isLight = getObjectVariable(i, 0, 1);
```

Calculating Light Levels

To create the desired effect, we need to come up with a function that is lighter closer to the object, but fades out further away from the object.

we can use the following equation to work out the *brightness* B of a pixel as a function of the inverse square of distance to each light emitting object O. We also clip the values between 0.0 and 1.0:

$$L = \max\left(0, \min\left(1, \sum_{i=0}^{n}\left(\frac{1}{a|\overrightarrow{BO_i}|^2} - b\right)\right)\right)$$

n here is the total number of light emitting objects. a and b are parameters we can adjust to get the desired lighting effects.

Note: We use the values $a = 1/30$ and $b = 0.1$ in this tutorial but feel free to modify them.

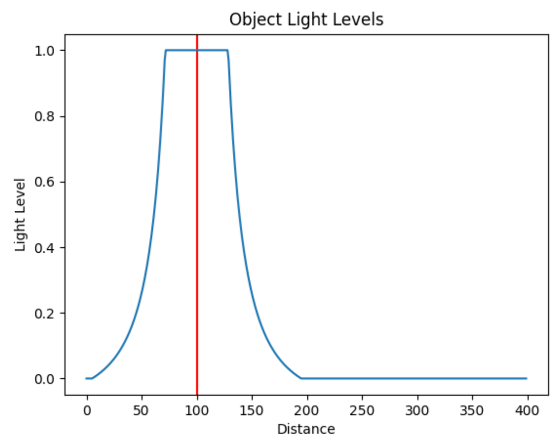
With a single object at $O_0 = 100$, the plot of Lighting level L against distance $|\overrightarrow{BO_i}|$ looks like this:

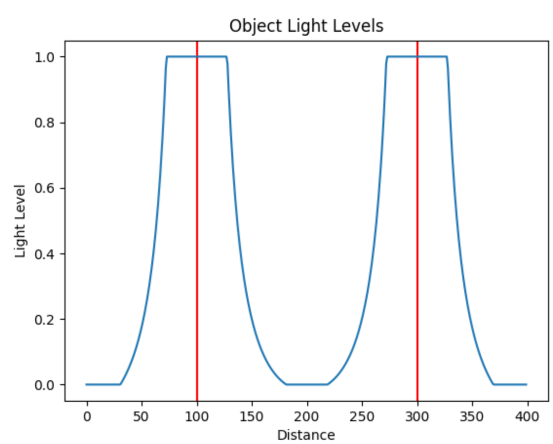
With multiple objects, $O_0 = 100$ and $O_1 = 300$:

To do this in the fragment shader, we iterate through all objects, check that the object is an object with lights (in this case, object type 0 and object type 1). We then calculate the above equation to get the light level:

```
float lightLevel = 0.0;
for (int i = 0; i < objectDataBuffer.size; i++) {
  ObjectData object = objectDataBuffer.variables[i];
```

(continues on next page)


Object Light Levels
Light Level
Distance
1.0
0.8
0.6
0.4
0.2
0.0
0
50
100
150
200
250
300
350
400


Object Light Levels
Light Level
Distance
1.0
0.8
0.6
0.4
0.2
0.0
0
50
100
150
200
250
300
350
400

(continued from previous page)

```
  if (object.objectType == 0 || object.objectType == 1) {
    int isLight = getObjectVariable(i, 0, 1);
    if (isLight == 1) {
      mat4 mv = environmentData.viewMatrix * object.modelMatrix;
      vec4 position = mv * vec4(0, 0, 0, 1);
      float dist_to_pixel = distance(position.xy, gl_FragCoord.xy);
      lightLevel += 1.0/pow(dist_to_pixel / 30.0, 2.0) - 0.1;
    }
  }
}

lightLevel = max(0, min(1.0, lightLevel));
```

We can then calculate the final pixel value by multiplying the light level by the RGB components of the texture that is being rendered:

```
outFragColor = texture(samplerArray, inFragTextureCoords) * vec4(lightLevel, lightLevel,␣
↪lightLevel, 1.0);
```

Object Lighting Shaders

Putting all of this together in our fragement and vertex shaders we have the following:

Vertex

```
#version 460

layout(location = 0) in vec3 inPosition;
layout(location = 1) in vec2 inFragTextureCoords;

layout(location = 0) out vec4 outColor;
layout(location = 1) out vec3 outFragTextureCoords;
layout(location = 2) out vec4 outPlayerColor;

out gl_PerVertex {
  vec4 gl_Position;
};

struct ObjectData {
  mat4 modelMatrix;
  vec4 color;
  vec4 gridPosition;
  vec2 textureMultiply;
  int textureIndex;
  int objectType;
  int playerId;
};

layout(std140, binding = 1) uniform EnvironmentData {
```

(continues on next page)

(continued from previous page)

```
  mat4 projectionMatrix;
  mat4 viewMatrix;
  vec4 globalObserverAvatarHighlightColor;
  vec2 gridDims;
  int playerCount;
  int playerId;
  int globalVariableCount;
  int objectVariableCount;
  int globalObserverAvatarMode;
  int highlightPlayers;
}
environmentData;

layout(std430, binding = 3) readonly buffer ObjectDataBuffer {
  uint size;
  ObjectData variables[];
}
objectDataBuffer;

layout(push_constant) uniform PushConsts {
  int idx;
}
pushConsts;


void main() {
  ObjectData object = objectDataBuffer.variables[pushConsts.idx];

  outFragTextureCoords = vec3(
      inFragTextureCoords.x * object.textureMultiply.x,
      inFragTextureCoords.y * object.textureMultiply.y,
      object.textureIndex);

  mat4 mvp = environmentData.projectionMatrix * environmentData.viewMatrix * object.
↪modelMatrix;

  gl_Position = mvp * vec4(
                          inPosition.x,
                          inPosition.y,
                          inPosition.z,
                          1.);
}
```

Fragment

```
#version 460

layout(binding = 0) uniform sampler2DArray samplerArray;

layout(location = 0) in vec4 inColor;
layout(location = 1) in vec3 inFragTextureCoords;
layout(location = 2) in vec4 playerColor;

layout(location = 0) out vec4 outFragColor;

struct GlobalVariable {
  int value;
};

struct ObjectVariable {
  int value;
};

struct PlayerInfo {
  vec4 playerColor;
  vec4 playerObservableGrid;
};

struct ObjectData {
  mat4 modelMatrix;
  vec4 color;
  vec4 gridPosition;
  vec2 textureMultiply;
  int textureIndex;
  int objectType;
  int playerId;
};

layout(std140, binding = 1) uniform EnvironmentData {
  mat4 projectionMatrix;
  mat4 viewMatrix;
  vec4 globalObserverAvatarHighlightColor;
  vec2 gridDims;
  int playerCount;
  int playerId;
  int globalVariableCount;
  int objectVariableCount;
  int globalObserverAvatarMode;
  int highlightPlayers;
}
environmentData;

layout(std430, binding = 3) readonly buffer ObjectDataBuffer {
  uint size;
  ObjectData variables[];
```

(continues on next page)

(continued from previous page)

```
}
objectDataBuffer;

layout(std430, binding = 4) readonly buffer GlobalVariableBuffer {
  GlobalVariable variables[];
}
globalVariableBuffer;

layout(std430, binding = 5) readonly buffer ObjectVariableBuffer {
  ObjectVariable variables[];
}
objectVariableBuffer;

int getObjectVariable(in int objectIndex, in int variableIndex, in int numVariables) {
  return objectVariableBuffer.variables[objectIndex * numVariables + variableIndex].
→value;
}

void main() {

  float lightLevel = 0.0;
  for (int i = 0; i < objectDataBuffer.size; i++) {
    ObjectData object = objectDataBuffer.variables[i];

    if (object.objectType == 0 || object.objectType == 1) {
      int isLight = getObjectVariable(i, 0, 1);
      if (isLight == 1) {
        mat4 mv = environmentData.viewMatrix * object.modelMatrix;
        vec4 position = mv * vec4(0, 0, 0, 1);
        float dist_to_pixel = distance(position.xy, gl_FragCoord.xy);
        lightLevel += 1.0/pow(dist_to_pixel / 30.0, 2.0) - 0.1;
      }
    }
  }

  lightLevel = max(0, min(1.0, lightLevel));

  outFragColor = texture(samplerArray, inFragTextureCoords) * vec4(lightLevel,␣
→lightLevel, lightLevel, 1.0);
}
```

Full Code Example

Full code examples can be found here!

Global Lighting In this tutorial we use the global variable `_steps` in the fragment shader to change the lighting level of the entire environment. *Go to tutorial* | Go to code
Heath Bars In this tutorial we use the `health` and `max_health` variables that we define in the GDY for each object to create a "health bar" showing the health of the agents as they battle. The heath bars are created in the fragment shader using signed distance fields. *Go to tutorial* | Go to code
Object Lighting In this tutorial we use variables that we define in the GDY to create lights around certain objects. Again we used signed distance fields in the fragment shader to create the light effects. *Go to tutorial* | Go to code

CHAPTER

TWENTYTHREE

PROJECTILES

Pew Pew! Sometimes in games we want to create objects that move across the environment under their own power. In this tutorial we learn how to do this using *GDY*. We build an environment where the `jelly` agent can shoot projectiles to break `box` es that are sitting on an island in the middle of an ocean of slime. The agent receives a reward of 1 for every time a projectile hits a box, and a reward of 10 when all boxes are destroyed.

There's several game mechanics here that we will explain in detail:

- **Spawning Objects**
- **Initial Actions**
- **Input Mappings**
- **Delayed Actions**
- **Internal Actions**
- **Action Spaces**
- **Collisions**

23.1 Spawning the "flame" object

To spawn an object in a particular direction we use the following code:

```
- Name: flame_shoot
  InputMapping:
    Inputs:
      1:
        OrientationVector: [ 0, -1 ]
        VectorToDest: [ 0, -1 ]
        MetaData:
          image_idx: 0
      2:
        OrientationVector: [ 1, 0 ]
        VectorToDest: [ 1, 0 ]
        MetaData:
          image_idx: 1
      3:
        OrientationVector: [ 0, 1 ]
        VectorToDest: [ 0, 1 ]
        MetaData:
          image_idx: 2
```

(continues on next page)

(continued from previous page)

```
      4:
        OrientationVector: [ -1, 0 ]
        VectorToDest: [ -1, 0 ]
        MetaData:
          image_idx: 3
  Behaviours:
    - Src:
        Object: jelly
      Dst:
        Object: [ grass, _empty ]
        Commands:
          - spawn: flame
```

Firstly want to be able to `spawn` our pink `flame` object in a particular direction from our `jelly` agent.

We can do this by defining an action which we will call `flame_shoot`. In this action, we have 4 `Input` objects, with ids 1-4 (0 is reserved for NOP actions).

Each Input is associated with a particular vector, which defines the direction and magnitude of the action. In this particular case the 4 actions correspond to `up`, `right`, `down` and `left` respectively.

23.1.1 Behaviours

We define which objects can perform actions and which objects can be the destination of actions by using `Behaviours`.

In our snippet above we only have a single `Behaviour`. This definition says that: If the object `jelly` has `grass` or `_empty` (a special object name for "an empty space") in the *destination location* of the action, then `spawn` a `flame` object there.

The *destination location* is calculated as the location of the *source object* (`jelly`) plus the vector given in `VectorToDest`.

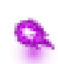
The action `flame_shoot` will automatically be exposed as an `action_type` with 4 `action_ids` in the environment's action space.

See also:

You can find much more information about action spaces *here*

23.2 Setting the `flame` tile image and initial direction

There are 4 images that we are going to use for the `flame` object:

`tile_id`	0	1	2	3
Image				

When the `flame` spawns, we want to make sure we set the correct tile based on the direction. For this we can use action `MetaData` variables and `InitialActions`:

23.2.1 Action MetaData

```
MetaData:
  image_idx: 0
```

In the previous section, we defined the the `flame_shoot` action. In each defined `action_id` of the `InputMapping` of this action, we include the `VectorToDest` and also the `MetaData` of this action. For each `action_id` you can define as many `MetaData` variables as you like. Think of them as constants that are available in the behaviour of the action. For each of the `action_ids` we set a `image_idx` variable which we can then use to set the current tile on the `flame` object.

In the GDY we define 4 tiles which can be used to render the `flame` object:

```
Objects:
  - Name: flame
    ...
    Observers:
      Isometric:
        - Image: oryx/oryx_iso_dungeon/fire-pink-up.png
        - Image: oryx/oryx_iso_dungeon/fire-pink-right.png
        - Image: oryx/oryx_iso_dungeon/fire-pink-down.png
        - Image: oryx/oryx_iso_dungeon/fire-pink-left.png
```

Now we have defined our 4 images for UP, DOWN, LEFT and RIGHT and our `image_idx` for each direction, we can make sure the right image is selected using `InitialActions`

23.2.2 Initial Actions

For this game in particular, we are going to create **two** initial actions. The first will only set the correct tile for the corresponding direction and the second will set the `flame` object in motion.

```
- Name: flame
  ...
  InitialActions:
    - Action: set_flame_direction
    - Action: flame_projectile_movement
      Delay: 2
```

set_flame_direction

```
- Name: set_flame_direction
  InputMapping:
    Internal: true
  Behaviours:
    - Src:
        Object: flame
        Commands:
          - set_tile: meta.image_idx
      Dst:
        Object: [ grass, _empty, flame, box ]
```

When an object is spawned, it automatically inherits the `MetaData` and `VectorToDest` of the *spawning action* (in this case `flame_shoot`). This means that the *destination location* for the `Behaviours` will be calculated relative to the *source object* using the previous `VectorToDest`.

For example: * The `jelly` at $[5, 5]$ spawns a `flame` object using `action_id` 2. The *destination location* of the action is $[6, 5]$ * The `flame` object is spawned at location $[6, 5]$ * The `flame` object then executes `set_flame_direction`. This also uses `action_id` 2 from the previous action, meaning the *destination location* will be $[7, 5]$

We don't really care what is in location $[7, 5]$, so we can set the possible destination objects as any of the possible objects in the environment.

Finally we perform a `set_tile` command using the action `MetaData`. We can reference this variable using the `meta.` prefix:

```
Commands:
  - set_tile: meta.image_idx
```

flame_projectile_movement

We add a delay to the `flame_projectile_movement` action so that it's only called after 3 game ticks.

Like the `set_flame_direction` this action will inherit the action `MetaData` and `VectorToDest`. We don't need the `MetaData` in the `flame_projectile_movement` action as we have already set the tile, but the `VectorToDest` can be used to set the direction of travel of the projectile.

We will cover this in the next section!

23.3 Projectile movement

```
- Name: flame_projectile_movement
  InputMapping:
    Internal: true
  Behaviours:
    - Src:
        Object: flame
        Commands:
          - mov: _dest
          - eq:
              Arguments: [ range, 0 ]
              Commands:
                - remove: true
          - gt:
              Arguments: [ range, 0 ]
              Commands:
                - decr: range
          - exec:
              Action: flame_projectile_movement
              Delay: 3
      Dst:
        Object: [ _empty, grass ]
```

When `flame_projectile_movement` is called, we check the *destination location* (using the inherited `VectorToDest`) of the object to see if there is `_empty` or `grass` object. If there is, we run some commands. Lets break these down line by line:

- Firstly move the flame object to the _dest variable, which contains the calculated *destination location*.

```
- mov: _dest
```

- Next we check a range variable. This is initialized in the flame object. If the range variable is 0. We remove the flame object.

```
- eq:
  Arguments: [ range, 0 ]
  Commands:
    - remove: true
```

- Then we check the range variable again, but this time we are looking if its larger than 0. If it *is*, then we decrement the value by 1.

```
- gt:
    Arguments: [ range, 0 ]
    Commands:
      - decr: range
```

- Finally we call the flame_projectile_movement function from within itself. But with a delay of 3 game ticks. So the process repeats again!

```
- exec:
    Action: flame_projectile_movement
    Delay: 3
```

Putting all of these commands together, the flame object moves one square in the initial direction every 3 game ticks. If the flame object moves more than it's range. Then it will be removed.

However, what happens if the flame encounters something thats not _empty or grass? What we **want** to happen is that we want the flame to destroy boxes, we also want to make sure that flames that bump into each other, or go off the edge of the map disappear.

This can be achieved by adding two more Behaviours that handle these collisions.

23.4 Projectile Collisions

```
Behaviours:
  ...
  - Src:
      Object: flame
      Commands:
        - remove: true
        - reward: 1
    Dst:
      Object: box
      Commands:
        - remove: true
  - Src:
      Object: flame
      Commands:
        - remove: true
```

(continues on next page)

(continued from previous page)

```
    Dst:
      Object: [flame, _boundary]
```

In the snippet above, we have two `Behaviours` the first one executes if the `flame` object has the *destination location* of a `box` object. In this case, the we remove both the `flame` and the `box` and give a reward of 1.

The second `Behaviour` will remove the flame if it has the *destination location* of another `flame` or the `_boundary` object (which is a special pseudo object referring to the boundary of the environment.)

23.5 Gym Interface

23.5.1 Load the GDY and create a gym environment

Loading the environment is super simple, you can just point the `GymWrapper` class at the `projectiles.yaml`:

```
env = GymWrapper('projectiles.yaml', player_observer_type=gd.ObserverType.ISOMETRIC)
env.reset()
```

You now have an `env` that you can use in Reinforcement Learning or any other experiments.

23.5.2 Action Space

So how can we now use this environment? How are the actions that we have defined exposed in the gym interface?

We have defined 4 actions in our GDY:

- **`move`**
  - Move the `jelly` (UP,DOWN,LEFT,RIGHT)
  - We didn't actually mention this one in the tutorial above because its super simple, just a single behaviour that uses the `mov: _dest` command and the default `InputMapping` (UP,DOWN,LEFT,RIGHT).
- **`flame_projectile_movement`**
  - Defines the movement of projectiles
- **`flame_shoot`**
  - Shoot a projectile in a particular direction (UP,DOWN,LEFT,RIGHT)
- **`set_flame_direction`**
  - Defines the movement of projectiles

But we only want to be able to expose the `move` and `flame_shoot` actions. All actions defined in GDY are exposed by default, so to **stop** an action being exposed we use the following:

```
InputMapping:
  Internal: true
```

This tells the Griddly engine that these actions are only used internally in the game, and cannot be called by an agent.

The actions that are exposed can then be used in the `env.step` function:

```
env.step([0, 1]) # move UP
env.step([0, 2]) # move RIGHT
env.step([0, 3]) # move DOWN
env.step([0, 4]) # move LEFT

env.step([1, 1]) # flame_shoot UP
env.step([1, 2]) # flame_shoot RIGHT
env.step([1, 3]) # flame_shoot DOWN
env.step([1, 4]) # flame_shoot LEFT
```

See also:

For more information on how Griddly deals with any action space you should look *here*

And thats about it for this tutorial!

23.6 Full Code Example

Full code examples can be found here!

CHAPTER

TWENTYFOUR

STOCHASTICITY

In this tutorial you will learn all about how to add stochasticity to environments. We will build an example environment where cows wander around a field and might eat some grass occaisionally. Your job is to replant the grass!

Additionally in this example we use some custom assets from the Crafter Reinforcement Learning environment.

There's several game mechanics here that we will explain in detail:

- **Action Execution Probability**
- **Random Action Choices**
- **Initial Actions**
- **Input Mappings**
- **Delayed Actions**
- **Using Custom Assets**
- **Internal Actions**
- **Changing Object Type**

24.1 Random Cow Movement

To achieve random movement of any object in Griddly, there are a few components that are required.

Firstly you need an action that defines how the object will move. In our case, we just want the cow to move UP, DOWN, LEFT and RIGHT. The GDY for this is the same as us defining actions for the agent:

```
- Name: cow_random_movement
  InputMapping:
    Internal: true
```

By not defining the `Inputs` key here, Griddly will use the default `action_id``s for UP, DOWN, LEFT, RIGHT. Note here we also set the action ``InputMapping to Internal: true`. This is so the `cow_random_movement` action cannot be access by the agent.

We then need to define the `Behaviours` of the `cow_random_movement`:

```
- Src:
    Object: cow
    Commands:
      - mov: _dest
      - exec:
```

(continues on next page)

(continued from previous page)

```
          Action: cow_eat_grass
      - exec:
          Action: cow_random_movement
          Delay: 1
          Randomize: true
  Dst:
    Object: [ grass, dirt ]
```

This action defines what will happen when the `cow_random_movement` action is performed on the `cow` object when the *destination location* has grass or dirt objects. There are three `Commands`; the first one will move the cow to the *destination location* of the action, the second will execute the `cow_eat_grass` action (we will get to this later). The third will re-execute the `cow_random_movement` with a small delay. It will also **randomize** the `action_id` that is executed. This means that it will randomly choose UP, DOWN, LEFT or RIGHT for the next action.

What happens if the `cow` is not next to `grass` or `dirt`? We don't want to move the cow, but we **do** want to try to move the cow again with a short delay. If we don't do this the cow will get stuck and no longer move!

```
- Src:
    Object: cow
    Commands:
      - exec:
          Action: cow_random_movement
          Delay: 1
          Randomize: true
  Dst:
    Object: [ _empty, _boundary, cow, player ]
```

Finally, we need to initialize the `cow_random_movement` for when the cows are generated at the start of the game. This is done using `InitialActions`.

```
- Name: cow
  InitialActions:
    - Action: cow_random_movement
      Randomize: true
```

24.2 Setting the Probability of a `cow` Eating Grass

Now we need to define the `cow_eat_grass` action to only execute a percentage of the time that it is called by the `cow_random_movement` command.

```
- Name: cow_eat_grass
  Probability: 0.05
  InputMapping:
    Internal: true
    Inputs:
      1:
        VectorToDest: [0, 0]
  Behaviours:
    - Src:
        Object: cow
      Dst:
```

(continues on next page)

(continued from previous page)

```
        Object: grass
        Commands:
          - change_to: dirt
```

This action contains a `Probability` property meaning the action will only be executed with a probability of 0.05 every time it is called. When the action **is** executed the `grass` object under the `cow` will be changed to a `dirt` object.

24.3 Planting the `grass` that a `cow` has Eaten.

The dirt object can then be changed back into `grass` by the agent:

```
- Name: plant_grass
  InputMapping:
    Inputs:
      1:
        VectorToDest: [ 0, -1 ]
  Behaviours:
    - Src:
        Object: player
        Commands:
          - reward: 1
      Dst:
        Object: dirt
        Commands:
          - change_to: grass
```

We also restrict the `plant_grass` action to the square above the `player`.

24.4 Using Custom Assets for your Environments

To use custom assets in your environment all you have to do is put all the assets in one directory, and then set the `image_path` in the `GymWrapper` when creating the environment:

```
env = GymWrapper('stochasticity.yaml',
            player_observer_type=gd.ObserverType.SPRITE_2D,
            global_observer_type=gd.ObserverType.SPRITE_2D,
            image_path='./assets/',
            level=0)
```

To make sure that your assets look how you expect them. Make sure that they are all the same dimensions. Internally Griddly will re-size them all to the `TileSize` (default 24x24) set in the Environment Observer definition:

```
Environment:
  ...
  Observers:
    Sprite2D:
      TileSize: 48
```

24.5 Stacking Objects

In Griddly, objects can be stacked on-top of each other. The stacking order (or Z-index) of the objects must be consistent with the z-index of the objects when they are defined. The object on "top" of the stack is always the first object that can be interacted with in actions.

Defining a Z index of an object is done in the object definition, for example in the following snippet, we define that the `player` sprite will always be rendered on top of the `grass` sprite:

```
- Name: player
  MapCharacter: p
  Z: 2
  Observers:
    Sprite2D:
      - Image: player.png
- Name: grass
  MapCharacter: G
  Z: 1
  Observers:
    Sprite2D:
      - Image: grass.png
```

We can also stack objects in the level map itself by using the / character:

```
Levels:
  - |
    G   G   G   G   G   G   G   G
    G   G   G   G   G   G   c/G G
    G   G   G   G   G   G   G   G
    G   G   G   G   G   G   G   G
    G   G   G   p/G G   G   G   G
    G   G   G   G   G   G   G   G
    G   G   G   G   G   G   G   G
    G   c/G G   G   G   G   G   G
    G   G   G   G   G   G   G   G
```

The Map characters in combination with the / character have the following meanings:

Map Characters	Meaning
`G`	`grass`
`c/G`	`cow` on `grass`
`p/G`	`player` on `grass`

24.6 Full Code Example

Full code examples can be found here!

CHAPTER

TWENTYFIVE

LEVEL DESIGN

In this tutorial we will learn how to design levels for our environments. In GDY, Levels are built entirely using **level strings**. A level string contains characters divided by spaces, each different character corresponds to a different object in the environment. The characters that are used to place each object are configurable.

```
p   p   p   p   p   p   p   p
p   .   .   .   .   .   t   p
p   .   c1  .   .   h1  .   p
p   .   .   .   .   .   .   p
p   .   .   t   .   .   .   p
p   .   .   .   .   .   .   p
p   .   c2  .   .   h2  .   p
p   .   .   t   .   .   .   p
p   p   p   p   p   p   p   p
```

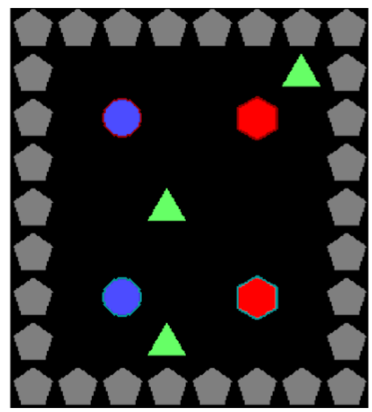


Fig. 1: A level string describing 4 different types of objects. Pentagons `p`, triangles `t`, circles `c` and hexagons `h`. Numbers after characters refer to the player that *owns* this object.

- **Basic Level Strings**
- **Object Ownership**
- **Stacking Objects**
- **Defining Multi-Agent Levels**
- **Player Highlighting**

25.1 Level String Basics

Every level has an associated `level_string` which defines the map of all the objects in that level. Level strings are a list of strings that make up a 2D coordinate mapping of the level. The list of strings is separated by a newline `\\n` character denoting a change in `y` coordinate. Within each string in the list, each character (ignoring whitespace and special characters) denotes a new object.

This explanation becomes much simpler with an example:

```
p   p   p   p   p   p   p   p
p   .   .   .   .   .   .   p
p   .   h   .   .   .   .   p
p   .   .   .   .   c   .   p
p   .   .   .   .   .   .   p
p   .   .   .   .   .   .   p
p   .   c   .   .   .   .   p
p   .   .   .   .   .   .   p
p   p   p   p   p   p   p   p
```

This level string defines an empty *room*, where the walls are made up of `p` objects there are `c` and `h` objects in the *room*. The **Map Characters** `p`, `c` and `h` are associated with objects in the GDY.

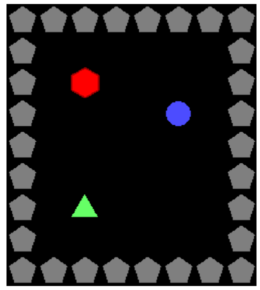
25.1.1 Object Map Chacters

```yaml
- Name: hexagon
 MapCharacter: h
 Observers:
   Block2D:
     - Shape: hexagon
       Color: [1,0,0]
       Scale: 1.0

- Name: circle
  MapCharacter: c
  Observers:
    Block2D:
      - Shape: circle
        Color: [0.3,0.3,1]

- Name: pentagon
  MapCharacter: p
  Observers:
    Block2D:
      - Shape: pentagon
        Color: [0.5,0.5,0.5]
```

The `MapCharacter` in the object definitions tells Griddly which object to place in the each level when it is created. In this case the *wall* (`p`) of the room will be the `pentagon` object, the `c` object a `circle` and finally the `h` object is a `hexagon`.

the `.` characters denote that there are no objects in that location. The number of spaces between each object does not matter (they are ignored).

When rendered… this environment looks like this:



25.2 Object Ownership

Lets's say we have an environment we want to build where we need to associate rewards with certain objects, but these objects are never interacted with by the player… For example maybe the object will disappear after a certain amount of time and the avatar loses a reward…

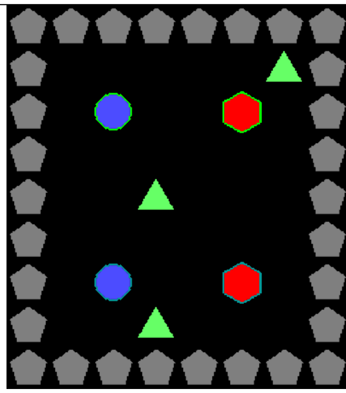
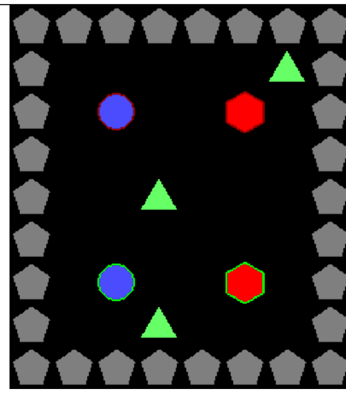
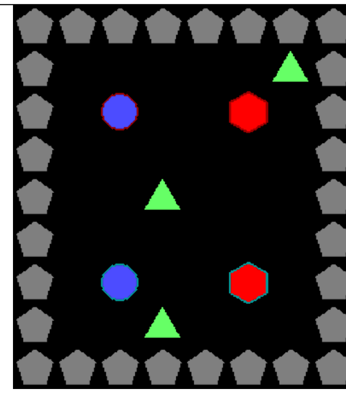
For this purpose we can define objects in the map as *owned* by a player. We can do this by adding a number after the map character:

```
p   p   p   p   p   p   p   p
p   .   .   .   .   .   t   p
p   .   c1  .   .   h1  .   p
p   .   .   .   .   .   .   p
p   .   .   t   .   .   .   p
p   .   .   .   .   .   .   p
p   .   c2  .   .   h2  .   p
p   .   .   t   .   .   .   p
p   p   p   p   p   p   p   p
```

In this map, the `c` and `h` objects are owned by different agents `t` and `p` objects are not owned by any.

25.2.1 Multi-Agent perspectives

In multi-agent environments, the perspective of the agent is always changed so that each agent sees itself as "player 1". This is handled in all observer types. For example in the `Block2D` observer each agent will see it's own objects with green highlighting:

Agent Perspective	1	2	Global
Image			

25.3 Layering Objects

If we have more complex environments for example where different rooms may have different types of background or floors, or where objects may start `on top` of other objects, we can use the / character to define that an object is **on top** of another object.

```
p    p    p    p    p    p    p    p
p    s    s    s    s    s    s    p
p    s    s    s    s    h2/s s    p
p    s    s    s    s    s    s    p
p    s    s    s    s    s    s    p
p    s    s    s    s    s    s    p
p    s    h1/s s    s    s    s    p
p    s    s    s    s    s    s    p
p    p    p    p    p    p    p    p
```

It's also very important to define the ordering of the objects in the Z axis in the GDY. This makes sure that behaviours happen in the same priority order as they are rendered.

For example with the square `s` and `h` objects, we define them to have `Z` values of 2 and 1 respectively:

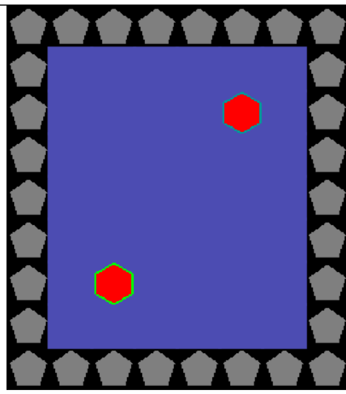
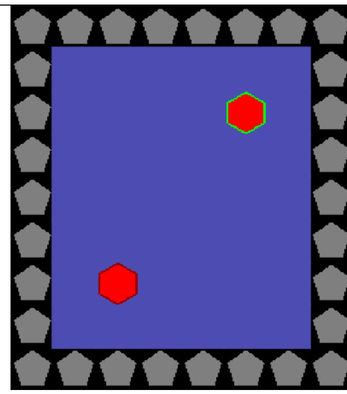
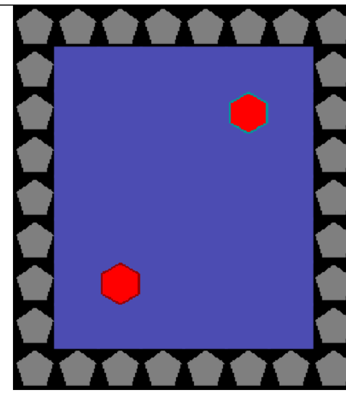
```yaml
- Name: square
  Z: 1
  MapCharacter: s
  Observers:
    Block2D:
      - Shape: square
        Color: [0.3,0.3,0.7]
        Scale: 1.5

- Name: hexagon
  MapCharacter: h
  Z: 2
  Observers:
```

(continues on next page)

(continued from previous page)

```
    Block2D:
      - Shape: hexagon
        Color: [1,0,0]
        Scale: 1.0
```

When rendered the hexagon objects are always rendered on top of the `square` objects.

Agent Perspective	1	2	Global
Image			

See also:

we show another example of this in the *stochasticity tutorial*.

25.4 Defining Levels Programmatically

If we don't want environments with fixed maps, we can also generate maps programmatically using any algorithm we choose. There's only two simple steps to this:

- Generate the level string
- Pass the level string to the `env.reset` function.

```
level_string = \
""". c c c c c c . . c c c c c c . . c c . c c c c c c . . c c c c c c . . c c . . . . .␣
↪. c c . . . . c c .
c c . . . . . . . c c . . . c c . c c . c c . . . c c . c c . . . c c . c c . . . . . . .
↪ c c . . c c . .
c c . . . c c c . c c c c c c . . c c . c c . . . c c . c c . . . c c . c c . . . . . . .
↪ . c c c c . . .
c c . . . . c c . c c . . . c c . c c . c c . . . c c . c c . . . c c . c c . . . . . . .
↪ . . c c . . . .
. c c c c c c . . c c . . . c c . c c . c c c c c c . . c c c c c c . . c c c c c c c . .
↪ . . c c . . . .
"""
env.reset(level_string=level_string)
```

This gives us a level that looks like this:

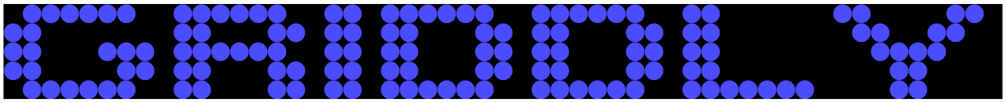


Fig. 2: We build an environment that spells out "GRIDDLY" by rendering ascii art as an environment!

See also:

If you want to know more about procedural content generation in Griddly, there's some documentation on it *here*

Full code examples can of all the above can found here!

CHAPTER

TWENTYSIX

A* SEARCH

In this tutorial we will learn how to give objects in the environment a small amount of intelligence by allowing them to use the A-Star Search algorithm to do pathfinding.

We build a simple environment where the agent must find a goal state while being chased by a gross spider. We build two versions of this environment where the `spider` has different movement characteristics.

In the first environment the `spider` can only move up, down left and right. In the second environment the `spider` can only rotate left and right and move forwards.

26.1 Defining Spider Movement

Firstly we need to define how the `spider` moves. In the example where we just want the `spider` to be able to move UP, DOWN, LEFT and RIGHT, we can just define a default `chase` action:

```
- Name: chase
  InputMapping:
    Internal: true
```

This action will have the default action mapping of UP, DOWN, LEFT and RIGHT. Also note that this action has `Internal:  true` so that this action cannot be performed by any controlling agents.

Alternatively for the environment where we want our `spider` to rotate left and right, and only move in the direction that it is travelling you can do the following:

```
- Name: chase
  InputMapping:
    Inputs:
      1:
        Description: Rotate left
        OrientationVector: [ -1, 0 ]
      2:
        Description: Move forwards
        OrientationVector: [ 0, -1 ]
        VectorToDest: [ 0, -1 ]
      3:
        Description: Rotate right
        OrientationVector: [ 1, 0 ]
    Relative: true
    Internal: true
```

See also:

You can find much more information about action spaces *here*

26.2 Using the Search option

The goal of out environment is to make the spider object find a path from its current location to a particular destination using the mocements we defined in chase. In GDY this is super simple to do and is the same for both of the cases above. You just need to tell the Griddly engine which actions it can use and which objects are *impassable*.

We can do that by using the Search option in an exec command.

```
- exec:
    Action: chase
    Delay: 10
    Search:
      ImpassableObjects: [ wall ]
      TargetObjectName: catcher
```

In the Griddly engine, this uses the A* search algorithm to find the best actionId (in this case which direction) to get the spider closer to the catcher object. In this case the catcher object is the name of the avatar we control. We also tell the A* algorithm that you cannot move through wall objects.

Now all we need to do is make sure the exec command is called when the spider moves. We can do that by adding to the Behaviours of the chase action:

```
- Name: chase
  ...
  Behaviours:
  - Src:
      Object: spider
      Commands:
        - mov: _dest
        - exec:
            Action: chase
            Delay: 10
            Search:
              ImpassableObjects: [ wall ]
              TargetObjectName: catcher
    Dst:
      Object: _empty

  # We only need this Behaviour if we are using the rotating version of the chase action
  - Src:
      Object: spider
      Commands:
        - rot: _dir
        - exec:
            Action: chase
            Delay: 0
            Search:
              ImpassableObjects: [ wall ]
              TargetObjectName: catcher
```

(continues on next page)

(continued from previous page)

```
    Dst:
      Object: spider
```

What we are doing here is telling telling the Griddly engine to execute another search operation every time the `spider` moves, (or rotates). We also only execute the `chase` action after a small delay of 10 if the spider actually moves to a new location. If the spider just rotates on the spot, we immediately execute another `chase` action so it moves as well as rotates.

26.3 Full Code Example

Full code examples can be found here!

CHAPTER

TWENTYSEVEN

GRIDDY DESCRIPTION YAML

This section contains a reference for the GDY DSL. All potential parameters are documented here.

Tutorials for building a Griddy Game Environment can be found *here*

27.1 Environment

Description Contains the definition of how the player interacts with the environment and how it is percieved by the players

Data Type	YAML Key
object	`Environment`

Properties

Property	Required
Name	`true`
Description	
Observers	
Player	
Variables	
Termination	
Levels	

27.1.1 Name

Description The name of the environment

Data Type	YAML Key
string	`Name`

27.1.2 Description

Description A description of the environment.

Data Type	YAML Key
string	`Description`

27.1.3 Observers

Description Default properties for observers

Data Type	YAML Key
object	`Observers`

27.1.4 Player

Description Defines how players (algorithms, agents, humans) interact with the environment.

Data Type	YAML Key
object	`Player`

Properties

Property	Required
Count	
AvatarObject	
Observer	

Player Count

Description The number of players in the environment

Data Type	YAML Key	Default Value
integer	`Count`	1

Avatar Object

Description The player will control a single object in the environment.

Data Type	YAML Key
string	`AvatarObject`

Observer Configuration

Description Defines how the observations will be generated for the players in the game.

Data Type	YAML Key
object	`Observer`

Properties

Property	Required
HighlightPlayers	
RotateWithAvatar	
RotateAvatarImage	
TrackAvatar	
Height	
Width	
OffsetX	
OffsetY	

Highlight Players

Description Add highlights to the players in visual observers.

Data Type	YAML Key	Default Value
boolean	`HighlightPlayers`	`True`

Rotate with avatar

Description The observer view will rotate to follow the orientation of the avatar.

Data Type	YAML Key	Default Value
boolean	`RotateWithAvatar`	`False`

Rotate Avatar Image

Description The observer will rotate the avatar image if the orientation changes.

Data Type	YAML Key	Default Value
boolean	`RotateAvatarImage`	`True`

Track Avatar

Description The observer view will track the position of the avatar.

Data Type	YAML Key	Default Value
boolean	`TrackAvatar`	`False`

Height

Description Height of the observation window

Data Type	YAML Key
integer	`Height`

Width

Description Width of the observation window.

Data Type	YAML Key
integer	`Width`

OffsetX

Description X offset of the observer window.

Data Type	YAML Key	Default Value
integer	`OffsetX`	`0`

OffsetY

Description Y offset of the observation window.

Data Type	YAML Key	Default Value
integer	`OffsetY`	`0`

27.1.5 Global Variables

Description Definition global variables that can be access from any action.

Data Type	YAML Key
array	`Variables`

Array Type

Type	Description
Variable	Variable

Variable

Description Define a global variable such as number of items collected, actions performed etc.

Data Type
object

Properties

Property	Required
Name	`true`
InitialValue	
PerPlayer	

Variable name

Description The name for the variable

Data Type	YAML Key
string	`Name`

Variable Initial Value

Description The initial value of the variable

Data Type	YAML Key	Default Value
integer	`InitialValue`	`0`

Per Player Variable

Description The variable is not shared between players, each player has their own version of this variable

Data Type	YAML Key	Default Value
boolean	`PerPlayer`	`False`

27.1.6 Termination

Description Definition of the termination conditions of the environment.

Data Type	YAML Key
object	`Termination`

Properties

Property	Required
Lose	
Win	
End	

Lose Conditions

Description If any of these conditions are met, the player associated with this condition will lose the game.

Data Type	YAML Key
array	`Lose`

Array Types

Type	Description
Termination Conditions V1	When a termination condition is met, the game will reset itself. If there are multiple players, the termination arguments are expanded internally "per player". This can be used to find the first player to a certain number of objects, or the first player to reach a certain score
Termination Conditions V2	When a termination condition is met, the game will reset itself. If there are multiple players, the termination arguments are expanded internally "per player". This can be used to find the first player to a certain number of objects, or the first player to reach a certain score

Termination Conditions V1

Description When a termination condition is met, the game will reset itself. If there are multiple players, the termination arguments are expanded internally "per player". This can be used to find the first player to a certain number of objects, or the first player to reach a certain score

Data Type
object

Properties

Property	Required
eq	
neq	
gt	
gte	
lt	
lte	

Equals

Description Check if the arguments are equal

Data Type	YAML Key	Max Items	Min Items
array	`eq`	2	2

Array Type

Type	Description
Termination Arguments	An argument to the termination condition. If there are multiple players, then these arguments expand internally as "per player"

Termination Arguments

Description An argument to the termination condition. If there are multiple players, then these arguments expand internally as "per player"

Not Equals

Description Check if the arguments are not equal

Data Type	YAML Key	Max Items	Min Items
array	`neq`	2	2

Array Type

Type	Description
Termination Arguments	An argument to the termination condition. If there are multiple players, then these arguments expand internally as "per player"

Greater Than

Description Check if the first argument is greater than the second

Data Type	YAML Key	Max Items	Min Items
array	`gt`	2	2

Array Type

Type	Description
Termination Arguments	An argument to the termination condition. If there are multiple players, then these arguments expand internally as "per player"

Greater Than Or Equal

Description Check if the first argument is greater than or equal to the second

Data Type	YAML Key	Max Items	Min Items
array	`gte`	2	2

Array Type

Type	Description
Termination Arguments	An argument to the termination condition. If there are multiple players, then these arguments expand internally as "per player"

Less Than

Description Check if the first argument is less than the second

Data Type	YAML Key	Max Items	Min Items
array	`lt`	2	2

Array Type

Type	Description
Termination Arguments	An argument to the termination condition. If there are multiple players, then these arguments expand internally as "per player"

Less Than Or Equal

Description Check if the first argument is less than or equal to the second

Data Type	YAML Key	Max Items	Min Items
array	`lte`	2	2

Array Type

Type	Description
Termination Arguments	An argument to the termination condition. If there are multiple players, then these arguments expand internally as "per player"

Termination Conditions V2

Description When a termination condition is met, the game will reset itself. If there are multiple players, the termination arguments are expanded internally "per player". This can be used to find the first player to a certain number of objects, or the first player to reach a certain score

Data Type
object

Properties

Property	Required
Conditions	
Reward	
OpposingReward	

Conditions

Description If any of these conditions are met, the game will end and distribute rewards to the associated players.

Data Type	YAML Key
array	`Conditions`

Array Type

Type	Description
Termination Conditions V1	When a termination condition is met, the game will reset itself. If there are multiple players, the termination arguments are expanded internally "per player". This can be used to find the first player to a certain number of objects, or the first player to reach a certain score

Reward

Description The reward given to the agent if this conditions is met.

Data Type	YAML Key
integer	`Reward`

Opposing Reward

Description The reward given to other agents if this condition is met.

Data Type	YAML Key
integer	`OpposingReward`

Win Conditions

Description If any of these conditions are met, the player associated with this condition will win the game.

Data Type	YAML Key
array	`Win`

Array Types

Type	Description
Termination Conditions V1	When a termination condition is met, the game will reset itself. If there are multiple players, the termination arguments are expanded internally "per player". This can be used to find the first player to a certain number of objects, or the first player to reach a certain score
Termination Conditions V2	When a termination condition is met, the game will reset itself. If there are multiple players, the termination arguments are expanded internally "per player". This can be used to find the first player to a certain number of objects, or the first player to reach a certain score

End Conditions

Description If any of these conditions are met, the game will end.

YAML Key
`End`

Array Types

Type	Description
Termination Conditions V1	When a termination condition is met, the game will reset itself. If there are multiple players, the termination arguments are expanded internally "per player". This can be used to find the first player to a certain number of objects, or the first player to reach a certain score
Termination Conditions V2	When a termination condition is met, the game will reset itself. If there are multiple players, the termination arguments are expanded internally "per player". This can be used to find the first player to a certain number of objects, or the first player to reach a certain score

27.1.7 Game Level Maps

Description Level Strings which define the levels in the game environment.

Data Type	YAML Key
array	`Levels`

27.2 Actions

Description Actions define all the game mechanics.

Data Type	YAML Key
array	`Actions`

Array Type

Type	Description
Action Definition	Action Definition

27.2.1 Action Definition

Description A single action.

Data Type
object

Properties

Property	Required
Probability	
Trigger	
InputMapping	
Name	`true`
Behaviours	`true`

Probability

Description The probability that this action is executed.

Data Type	YAML Key
number	`Probability`

Trigger

Description Triggers can be used to set proximity and other triggers for actions.

Data Type	YAML Key
object	`Trigger`

Properties

Property	Required
Type	
Range	
Relative	
Offset	

Type

Description The type of trigger to use.

YAML Key	Allowed Values	Default Value
`Type`	`RANGE_BOX_AREA`, `RANGE_BOX_BOUNDARY`	`RANGE_BOX_AREA`

Range

Description The proximity of the objects required for this action to be executed.

Data Type	YAML Key
integer	`Range`

Relative

Description If the Offset of the collision detector should be relative to the direction that the object is facing.

Data Type	YAML Key
boolean	`Relative`

Offset

Description Offset for proximity sensing

Data Type	YAML Key	Max Items	Min Items
array	`Offset`	2	2

Array Type

Type	Description
Offset Coordinate	Offset Coordinate

Offset Coordinate

Description Coordinate for rendering offset of isometric tiles

Data Type
integer

Input Mappping

Description Map action Ids to in-environment actions

Data Type	YAML Key
object	`InputMapping`

Properties

Property	Required
Inputs	
Relative	
Internal	
MapToGrid	

Inputs

Description Each Key maps a single Action Id to a vector to the action's destination and an orientation vector.

Data Type	YAML Key
object	`Inputs`

Relative

Description If the vectors in the actions should be calculated relative to the orientation of the object that is being acted upon.

Data Type	YAML Key
boolean	`Relative`

Relative

Description If the action is set as internal, it cannot be used by any players.

Data Type	YAML Key
boolean	`Internal`

MapToGrid

Description Maps the action to the entire grid rather than locations relative to an object.

Data Type	YAML Key
boolean	`MapToGrid`

Name

Description The name given to an action such as "move" or "push".

Data Type	YAML Key
string	`Name`

Behaviours

Description The list of behaviours that define what happens to objects when actions are performed.

Data Type	YAML Key
array	`Behaviours`

Array Type

Type	Description
Behaviour Definition	Behaviour Definition

Behaviour Definition

Description Behaviour definitions are how Griddly builds the mechanics of environments. Behaviour definitions are made up of two components, the "source" behaviour and "destination" behaviour.

Data Type
object

Properties

Property	Required
Src	`true`
Dst	`true`
Probability	

Source

Description Define the behaviour of the source object of an action.

Data Type	YAML Key
object	`Src`

Properties

Property	Required
Object	`true`
Commands	
Preconditions	

Behaviour Objects

Description The object or list of objects that this behaviour applies to.

Possible Values

Value	Type	Description
`[string]`	string	A single object the behaviour is applied to.
List of Objects	array	A list of object the behaviour is applied to

Command Options

Description Commands or conditional commands

Data Type
array

Array Types

Type	Description
Behaviour Command	A command to run as part of an action behaviour.
Conditional Behaviour Command	A conditional command to run as part of an action behaviour.

Behaviour Command

Description A command to run as part of an action behaviour.

Properties

Property	Required
mov	
rot	
add	
sub	
set	
decr	
incr	
cascade	
reward	
remove	
change_to	
set_tile	
spawn	
exec	

Move

Description Move the object to the location provided.

YAML Key
`mov`

Choose Between

Type	Description
Destination	The destination location of the action.
Source	The source location of the action.
Command Arguments	Arguments supplied to the commmand, can be integer values or variable names.

Destination

Description The destination location of the action.

Data Type	Allowed Values
string	_dest

Source

Description The source location of the action.

Data Type	Allowed Values
string	_src

Command Arguments

Description Arguments supplied to the commmand, can be integer values or variable names.

Data Type	Max Items	Min Items
array	2	2

Array Type

Type	Description
Command Argument	An argument to a behaviour command.

Command Argument

Description An argument to a behaviour command.

Possible Values

Value	Type	Description
[string]	string	Any variables defined on the object can be used
[integer]	integer	Any Integer value

Rotate

Description Rotate the object to the direction supplied

Data Type	YAML Key
string	rot

Choose Between

Type	Description
Direction	The direction of the action.

Direction

Description The direction of the action.

Data Type	Allowed Values
string	`_dir`

Add

Description Add a value from a variable. For example "- add: [my_variable 10]" will add 10 to "my_variable".

Data Type	YAML Key	Max Items	Min Items
array	`add`	2	2

Array Type

Type	Description
Command Argument	An argument to a behaviour command.

Sub

Description Subtract a value from a variable. For example "- sub: [my_variable 10]" will subtract 10 from "my_variable".

Data Type	YAML Key	Max Items	Min Items
array	`sub`	2	2

Array Type

Type	Description
Command Argument	An argument to a behaviour command.

Set

Description Sets the value of a variable. For example "- set: [my_variable 10]" will set the value of "my_variable" to 10.

Data Type	YAML Key	Max Items	Min Items
array	`set`	2	2

Array Type

Type	Description
Command Argument	An argument to a behaviour command.

Decrement

Description Decrement the supplied variable

Data Type	YAML Key
string	`decr`

Increment

Description Increment the supplied variable

Data Type	YAML Key
string	`incr`

Cascade Action

Description Repeat this action again in the same direction.

YAML Key
`cascade`

Choose Between

Type	Description
Destination	The destination location of the action.
Source	The source location of the action.
Command Arguments	Arguments supplied to the commmand, can be integer values or variable names.

Reward

Description Increase the reward by the value specified. If the object is associated with a player, that player's reward is increased.

Data Type	YAML Key
integer	`reward`

Remove

Description Remove the object from the environment.

Data Type	YAML Key
boolean	`remove`

Change Object

Description Swaps this object for another one. (this does not preserve any properties of the current object)

Data Type	YAML Key
string	`change_to`

Set Tile

Description When multiple tiles are defined for an object in the Observer configurations, this can be used to set the current tile the object is using to render.

Data Type	YAML Key
integer	`set_tile`

Spawn

Description Spawns an object. For example "- spawn: ghost" will spawn a ghost object.

Data Type	YAML Key
string	`spawn`

Execute Action

Description Executes an action.

Data Type	YAML Key
object	`exec`

Properties

Property	Required
Action	`true`
Randomize	
Delay	
ActionId	
Search	

Action

Description The name of the action to perform.

Data Type	YAML Key
string	`Action`

Randomize

Description If set to true, a random action is chosen from the input mapping.

Data Type	YAML Key
boolean	`Randomize`

Delay

Description The action will be executed after this number of game ticks.

Data Type	YAML Key
integer	`Delay`

ActionId

Description The ID of the action in action mappings to perform.

Data Type	YAML Key
integer	`ActionId`

AStar Search

Description Executes the action with the action Id that is on the optimal search path to the destination object.

Data Type	YAML Key
object	`Search`

Properties

Property	Required
ImpassableObjects	
TargetObjectName	
MaxDepth	
Mode	
TargetLocation	

Impassable Objects

Description Objects that should be considered impassable by the search algorithm.

Data Type	YAML Key
array	`ImpassableObjects`

Array Type

Type	Description
Object Name	Object Name

Object Name

Description Names of the objects that are impassable.

Data Type
string

Target Object Name

Description The search algorithm navigate to the closest of these objects.

Data Type	YAML Key
string	`TargetObjectName`

Max Depth

Description Budget for AStar search depth.

Data Type	YAML Key
integer	`MaxDepth`

Mode

Description Whether the object will seek or flee from the target object.

YAML Key	Allowed Values	Default Value
`Mode`	`SEEK`, `FLEE`	`SEEK`

Target Location

Description Instead of navigating to a particular object, we can navigate to a particular location

Data Type	YAML Key	Default Value
string	`TargetLocation`	`[0, 0]`

Array Type

Type	Description
Target Coordinate	Target Coordinate

Target Coordinate

Description Coordinate for rendering offset of isometric tiles

Data Type
integer

Conditional Behaviour Command

Description A conditional command to run as part of an action behaviour.

Properties

Property	Required
eq	
lt	
lte	
gt	
gte	

Equals

Description The specified commands will only be run if the arguments are equal.

Data Type	YAML Key
object	`eq`

Properties

Property	Required
Arguments	
Commands	

Command Options

Description Commands

Data Type
array

Array Type

Type	Description
Behaviour Command	A command to run as part of an action behaviour.

Less Than

Description The specified commands will only be run if the value of the first argument is less than the second.

Data Type	YAML Key
object	`lt`

Properties

Property	Required
Arguments	
Commands	

Less Than Or Equal

Description The specified commands will only be run if the value of the first argument is less than or equal to the second.

Data Type	YAML Key
object	`lte`

Properties

Property	Required
Arguments	
Commands	

Greater Than

Description The specified commands will only be run if the value of the first argument is greater than the second.

Data Type	YAML Key
object	`gt`

Properties

Property	Required
Arguments	
Commands	

Greater Than Or Equal

Description The specified commands will only be run if the value of the first argument is greater than or equal to the second.

Data Type	YAML Key
object	`gte`

Properties

Property	Required
Arguments	
Commands	

Behaviour Preconditions

Description A list of checks that have to be performed before this action is executed.

Data Type	YAML Key
array	`Preconditions`

Array Type

Type	Description
Behaviour Precondition	A check that must be performed before any action. This can be used to change the behaviour of objects based on their internal variables. For example checking whether an object has a key before opening a door.

Behaviour Precondition

Description A check that must be performed before any action. This can be used to change the behaviour of objects based on their internal variables. For example checking whether an object has a key before opening a door.

Properties

Property	Required
eq	
neq	
gt	
gte	
lt	
lte	

Equals

Description Check if the arguments are equal

Data Type	YAML Key	Max Items	Min Items
array	`eq`	2	2

Array Type

Type	Description
Command Argument	An argument to a behaviour command.

Not Equals

Description Check if the arguments are not equal

Data Type	YAML Key	Max Items	Min Items
array	`neq`	2	2

Array Type

Type	Description
Command Argument	An argument to a behaviour command.

Greater Than

Description Check if the first argument is greater than the second

Data Type	YAML Key	Max Items	Min Items
array	`gt`	2	2

Array Type

Type	Description
Command Argument	An argument to a behaviour command.

Greater Than Or Equal

Description Check if the first argument is greater than or equal to the second

Data Type	YAML Key	Max Items	Min Items
array	`gte`	2	2

Array Type

Type	Description
Command Argument	An argument to a behaviour command.

Less Than

Description Check if the first argument is less than the second

Data Type	YAML Key	Max Items	Min Items
array	`lt`	2	2

Array Type

Type	Description
Command Argument	An argument to a behaviour command.

Less Than Or Equal

Description Check if the first argument is less than or equal to the second

Data Type	YAML Key	Max Items	Min Items
array	`lte`	2	2

Array Type

Type	Description
Command Argument	An argument to a behaviour command.

Dest

Description Define the behaviour of the destination object of an action.

Data Type	YAML Key
object	Dst

Properties

Property	Required
Object	true
Commands	

Probability

Description The probability that this particular behaviour is executed.

Data Type	YAML Key
number	Probability

27.3 Objects

Description An explanation about the purpose of this instance.

Data Type	YAML Key
array	Objects

Array Type

Type	Description
Object Definition	Object Definition

27.3.1 Object Definition

Description An explanation about the purpose of this instance.

Data Type
object

Properties

Property	Required
Z	
InitialActions	
Name	`true`
MapCharacter	
Variables	
Observers	

Z Index

Description The Z index of an object

Data Type	YAML Key	Default Value
integer	`Z`	`0`

Initial Actions

Description Actions that are performed as soon as the object is initialized. Can be for constant and random movements.

Data Type	YAML Key
array	`InitialActions`

Array Type

Type	Description
Initial Action	Initial Action

Initial Action

Description A list of actions that are performed by this object when it is initialized into the environment.

Properties

Property	Required
Action	`true`
Randomize	
Delay	
ActionId	

Action

Description The name of the action to perform.

Data Type	YAML Key
string	`Action`

Randomize

Description If set to true, a random action is chosen from the input mapping.

Data Type	YAML Key
boolean	`Randomize`

Delay

Description The action will be executed after this number of game ticks.

Data Type	YAML Key
integer	`Delay`

ActionId

Description The ID of the action in action mappings to perform.

Data Type	YAML Key
integer	`ActionId`

Name

Description An explanation about the purpose of this instance.

Data Type	YAML Key
string	`Name`

Map Character

Description The character this object is represented by in the level mapping.

Data Type	YAML Key
string	`MapCharacter`

Variable Definitions

Description An explanation about the purpose of this instance.

Data Type	YAML Key
array	`Variables`

Array Type

Type	Description
Variable	Variable

Variable

Description Define an object variable, such as health, items collected etc…

Data Type
object

Properties

Property	Required
Name	`true`
InitialValue	

Variable Name

Description The name for the variable

Data Type	YAML Key
string	`Name`

Variable Initial Value

Description The initial value of the variable

Data Type	YAML Key	Default Value
integer	`InitialValue`	`0`

Observers configuration

Description Configure how the observers render the object

Data Type	YAML Key
object	`Observers`