
Griddly

Release 1.2.14

Chris Bamford

Feb 22, 2022

ABOUT

1	Introduction	3
2	Community	5
3	Hall of Fame	7
4	Artwork	9
5	Frequently Asked Questions	11
6	Installation	13
7	OpenAI Gym Interface	15
8	Griddly Description YAML	17
9	Action Spaces	21
10	Observation Spaces	27
11	Visualization	33
12	Procedural Content Generation	37
13	Single-Player	43
14	RTS	251
15	Multi-Agent	291
16	Reinforcement Learning with RLLib	333
17	Single Agent Games	341
18	Multi Agent	345
19	Real Time Strategy	349
20	GDY Tutorial - Making Sokoban	353
21	GDY Schema Tutorial	365
22	Proximity Triggers	367



Griddly documentation.

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 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 you 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.

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

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}  
}
```


ARTWORK

The Artwork is provided by the [Oryx Design Lab](#).

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

FREQUENTLY ASKED QUESTIONS

Nothing here yet!

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.

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')
```

GRIDDLY DESCRIPTION YAML

Griddly Description YAML (GDY) 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:

```
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:
    - |
      wwwwww
      w..hA.w
      w.whw.w
      w...b.w
      whbb ww
      w..www
      wwwwww
    - |
      wwwwwwww
      ww.h...w
      ww...bA.w
      w...W..W
      wwbw...w
      www...W.w
      wwwh...w
```

(continues on next page)

(continued from previous page)

```

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

  # 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)

```

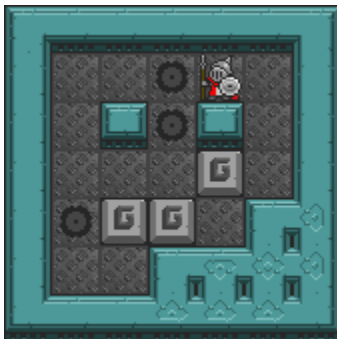
Sprite2D:
- TilingMode:
  WALL_16
  Image:
    - gvgai/oryx/wall3_0.png
    - gvgai/oryx/wall3_1.png
    - gvgai/oryx/wall3_2.png
    - gvgai/oryx/wall3_3.png
    - gvgai/oryx/wall3_4.png
    - gvgai/oryx/wall3_5.png
    - gvgai/oryx/wall3_6.png
    - gvgai/oryx/wall3_7.png
    - gvgai/oryx/wall3_8.png
    - 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/cspell14.png

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

```

This is all that's 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](#)

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 `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 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). Let's 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
  ]
])
```


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.

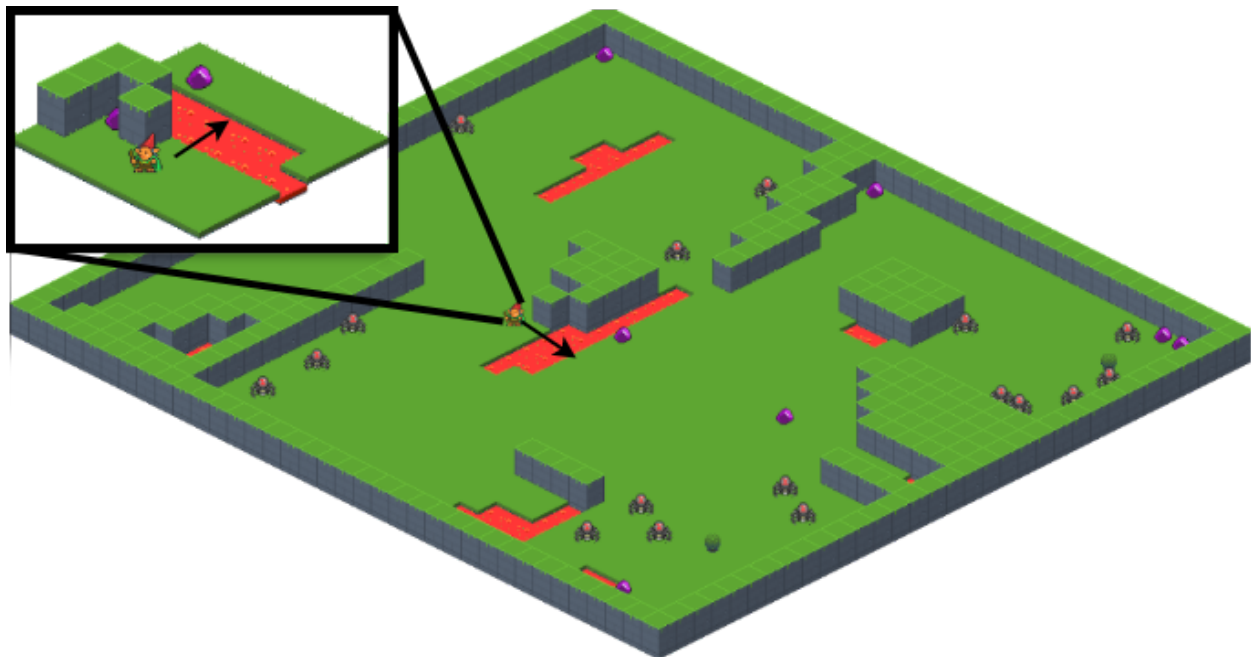


Fig. 1: The “Spider Nest” environment with an isometric player observer (inset) and isometric global observer configured.

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 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 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

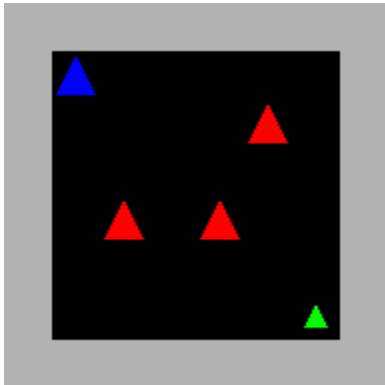
Note: For pixel-based observers, each object must define how it should be rendered with an *Observer Configuration*

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]  
  },  
  ...  
]
```

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.

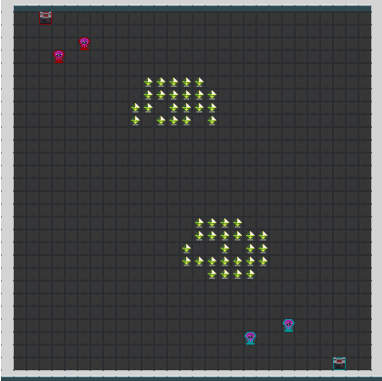
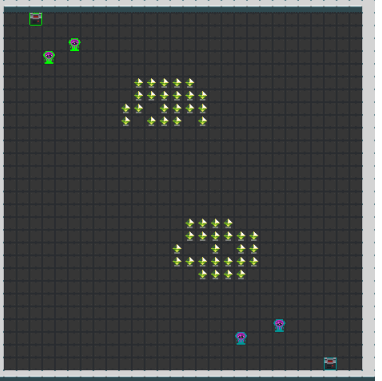
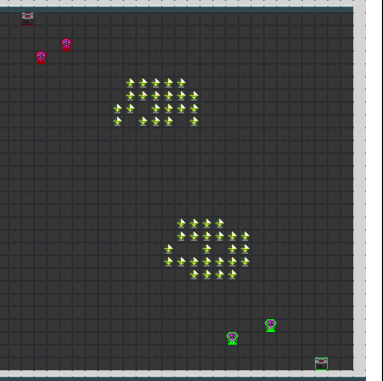
```
env.render(observer='global') # Will render the global observer
env.render(observer=0) # Will render the observer for player 1
env.render(observer=1) # Will render the observer for player 2
```

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()
```


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
```

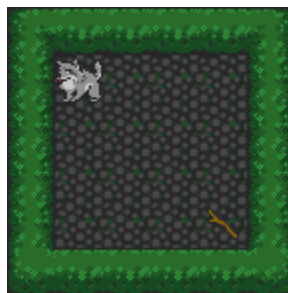


Fig. 1: How the above Doggo level is rendered.

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.

```
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:

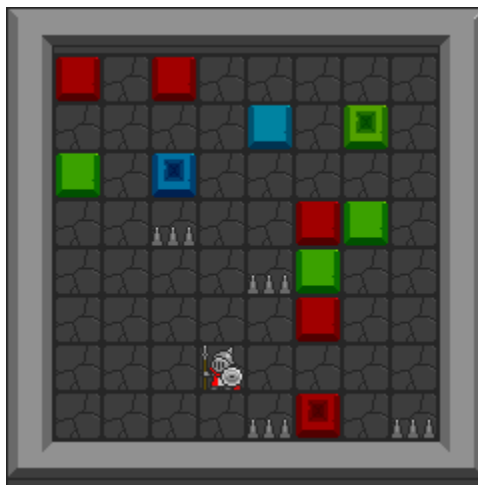


Fig. 2: A 10x10 map generated by the above code.

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
            }
        },
        ...
    }
```


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

	Vector	Sprite2D	Block2D	Isometric				
<table><tr><td>Level ID</td><td>0</td></tr><tr><td>Size</td><td>6x6</td></tr></table>	Level ID	0	Size	6x6				
Level ID	0							
Size	6x6							
<table><tr><td>Level ID</td><td>1</td></tr><tr><td>Size</td><td>7x7</td></tr></table>	Level ID	1	Size	7x7				
Level ID	1							
Size	7x7							
<table><tr><td>Level ID</td><td>2</td></tr><tr><td>Size</td><td>8x8</td></tr></table>	Level ID	2	Size	8x8				
Level ID	2							
Size	8x8							
<table><tr><td>Level ID</td><td>3</td></tr><tr><td>Size</td><td>10x10</td></tr></table>	Level ID	3	Size	10x10				
Level ID	3							
Size	10x10							
<table><tr><td>Level ID</td><td>4</td></tr><tr><td>Size</td><td>19x18</td></tr></table>	Level ID	4	Size	19x18				
Level ID	4							
Size	19x18							

13.1. Spiders

45

13.1. Spiders

13.1.3 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-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
Vector				
Sprite2D				
Block2D				
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

```

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 . . . . 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
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 . . . . . . . . . . . . . . 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 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

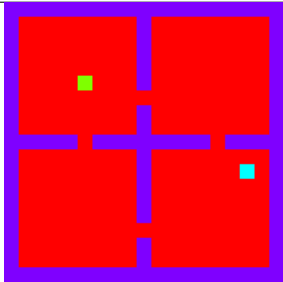
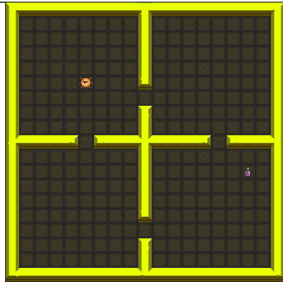
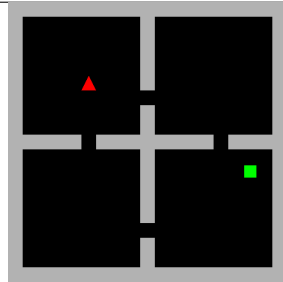
Single-Player/Mini-Grid/minigrid-eyeball.yaml

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

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

13.2.3 Code Example

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

```
import gym
import griddly

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
Vector			
Sprite2D			
Block2D			

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

```
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 . . . . 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 . . . . . . 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

```

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

	Vector	Sprite2D	Block2D				
<table><tr><td>Level ID</td><td>0</td></tr><tr><td>Size</td><td>6x6</td></tr></table>	Level ID	0	Size	6x6			
Level ID	0						
Size	6x6						
<table><tr><td>Level ID</td><td>1</td></tr><tr><td>Size</td><td>7x7</td></tr></table>	Level ID	1	Size	7x7			
Level ID	1						
Size	7x7						
<table><tr><td>Level ID</td><td>2</td></tr><tr><td>Size</td><td>8x8</td></tr></table>	Level ID	2	Size	8x8			
Level ID	2						
Size	8x8						
<table><tr><td>Level ID</td><td>3</td></tr><tr><td>Size</td><td>10x10</td></tr></table>	Level ID	3	Size	10x10			
Level ID	3						
Size	10x10						
13.3. Drunk Dwarf							

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 ->	<i>W</i>	<i>g</i>	<i>A</i>	<i>D</i>	<i>d</i>	<i>c</i>	<i>t</i>	<i>b</i>	<i>k</i>
Vector									
Sprite2D									
Block2D									

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

```

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 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 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
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 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)

```
- 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

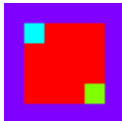
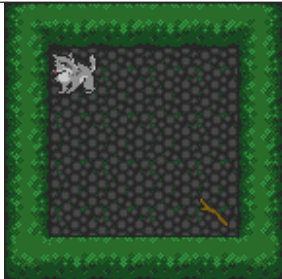
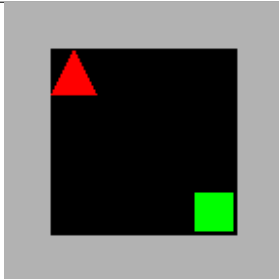
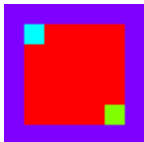
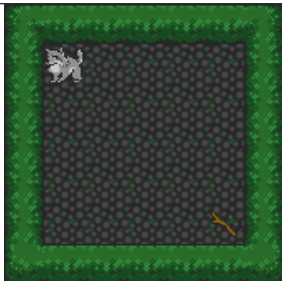
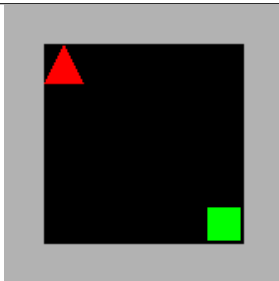
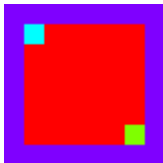
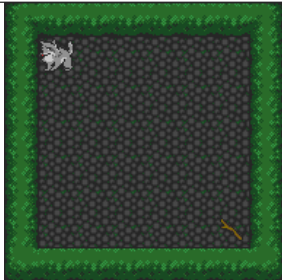
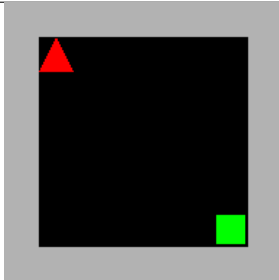
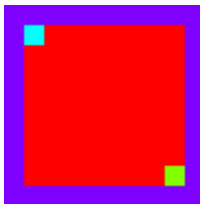
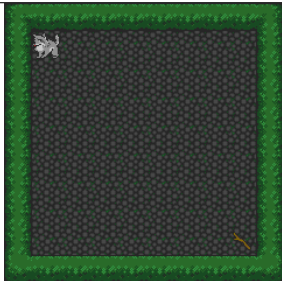
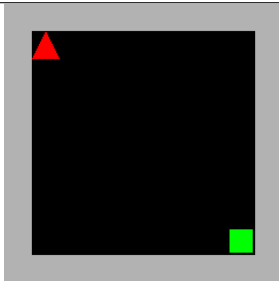
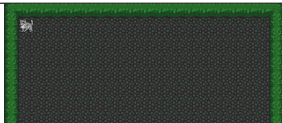
```
Single-Player/Mini-Grid/minigrid-doggo.yaml
```

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

	Vector	Sprite2D	Block2D				
<table><tr><td>Level ID</td><td>0</td></tr><tr><td>Size</td><td>6x6</td></tr></table>	Level ID	0	Size	6x6			
Level ID	0						
Size	6x6						
<table><tr><td>Level ID</td><td>1</td></tr><tr><td>Size</td><td>7x7</td></tr></table>	Level ID	1	Size	7x7			
Level ID	1						
Size	7x7						
<table><tr><td>Level ID</td><td>2</td></tr><tr><td>Size</td><td>8x8</td></tr></table>	Level ID	2	Size	8x8			
Level ID	2						
Size	8x8						
<table><tr><td>Level ID</td><td>3</td></tr><tr><td>Size</td><td>10x10</td></tr></table>	Level ID	3	Size	10x10			
Level ID	3						
Size	10x10						
68							

13.4.3 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-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 ->	<i>W</i>	<i>g</i>	<i>A</i>
Vector			
Sprite2D			
Block2D			

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

```

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 . . . . g 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

```

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)

```

- 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)

```
- 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

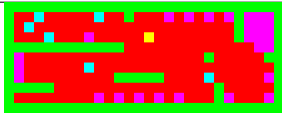
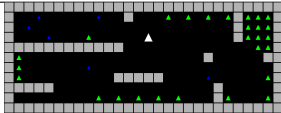
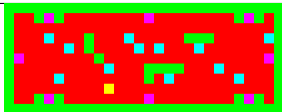
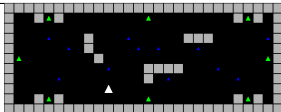
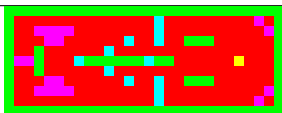
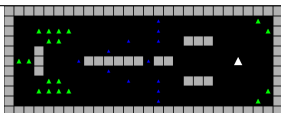
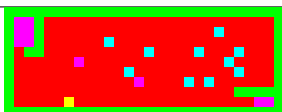
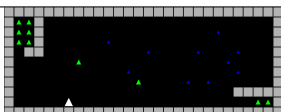
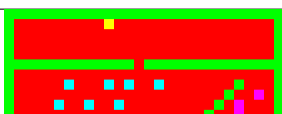
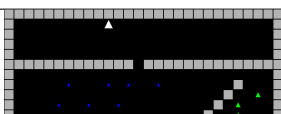
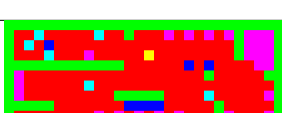
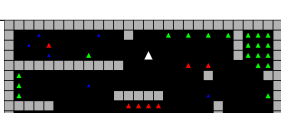
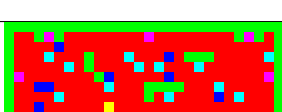
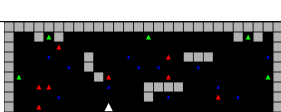
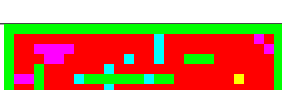
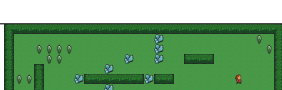
Single-Player/GVGAI/butterflies.yaml

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

	Vector	Sprite2D	Block2D				
<table><tr><td>Level ID</td><td>0</td></tr><tr><td>Size</td><td>28x11</td></tr></table>	Level ID	0	Size	28x11			
Level ID	0						
Size	28x11						
<table><tr><td>Level ID</td><td>1</td></tr><tr><td>Size</td><td>28x11</td></tr></table>	Level ID	1	Size	28x11			
Level ID	1						
Size	28x11						
<table><tr><td>Level ID</td><td>2</td></tr><tr><td>Size</td><td>28x11</td></tr></table>	Level ID	2	Size	28x11			
Level ID	2						
Size	28x11						
<table><tr><td>Level ID</td><td>3</td></tr><tr><td>Size</td><td>28x11</td></tr></table>	Level ID	3	Size	28x11			
Level ID	3						
Size	28x11						
<table><tr><td>Level ID</td><td>4</td></tr><tr><td>Size</td><td>28x12</td></tr></table>	Level ID	4	Size	28x12			
Level ID	4						
Size	28x12						
<table><tr><td>Level ID</td><td>5</td></tr><tr><td>Size</td><td>28x11</td></tr></table>	Level ID	5	Size	28x11			
Level ID	5						
Size	28x11						
<table><tr><td>Level ID</td><td>6</td></tr><tr><td>Size</td><td>28x11</td></tr></table>	Level ID	6	Size	28x11			
Level ID	6						
Size	28x11						
13.5. Butterflies and Spiders			75				
							

13.5.3 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-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
Vector					
Sprite2D					
Block2D					

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)

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 W W W W 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
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)

```

# 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)

```

# 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

	Vector	Sprite2D	Block2D				
<table><tr><td>Level ID</td><td>0</td></tr><tr><td>Size</td><td>8x7</td></tr></table>	Level ID	0	Size	8x7			
Level ID	0						
Size	8x7						
<table><tr><td>Level ID</td><td>1</td></tr><tr><td>Size</td><td>12x6</td></tr></table>	Level ID	1	Size	12x6			
Level ID	1						
Size	12x6						
<table><tr><td>Level ID</td><td>2</td></tr><tr><td>Size</td><td>7x8</td></tr></table>	Level ID	2	Size	7x8			
Level ID	2						
Size	7x8						
<table><tr><td>Level ID</td><td>3</td></tr><tr><td>Size</td><td>6x7</td></tr></table>	Level ID	3	Size	6x7			
Level ID	3						
Size	6x7						
13.6. Partially Observable Sokoban - 2							

13.6.3 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-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 ->	<i>b</i>	<i>f</i>	<i>w</i>	<i>h</i>	<i>A</i>
Vector					
Sprite2D					
Block2D					

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

```

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:
    - |
      WWWWWWWW
      WW.....W
      ww.hbh.w
      ww.bAb.w
      w..hbh.w
      W.....W
      WWWWWWWW
    - |
      WWWWWWWWWWWW
      W....WWW...W
      w.bb....wAw
      w.b.whhh...w
      W...WWWWWWW
      WWWWWWWWWWWW
    - |
      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
- |
wwwwwwwww
www.hhAw
www.bb.w
wwww.www
wwww.www
wwww.www
wwww.www
W...WWW
W.W...WW
W..W.WW
WWW...WW
wwwwwwwww

```

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

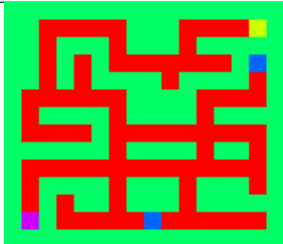
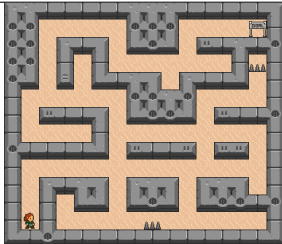
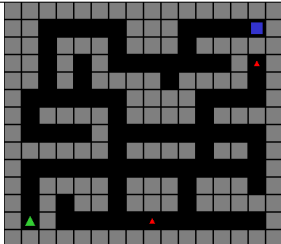
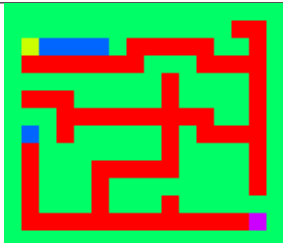
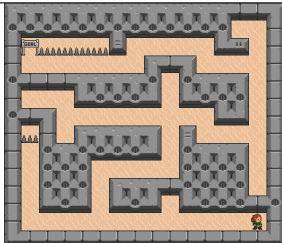
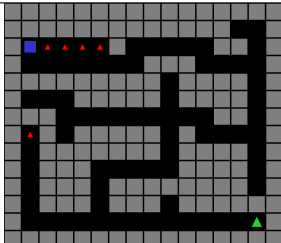
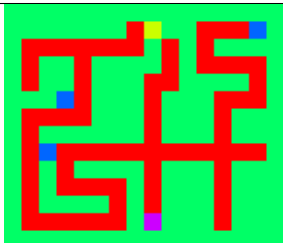
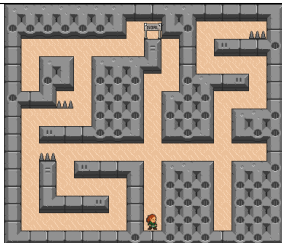
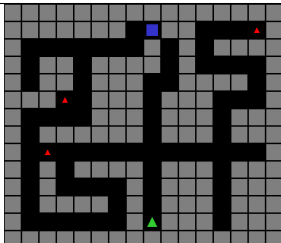
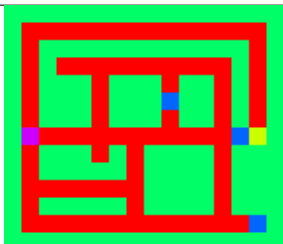
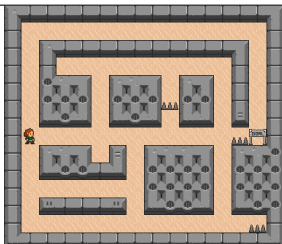
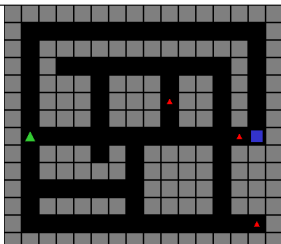
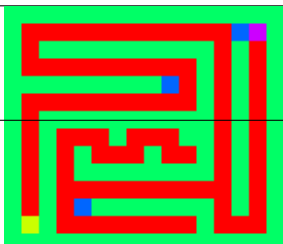
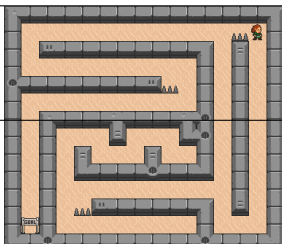
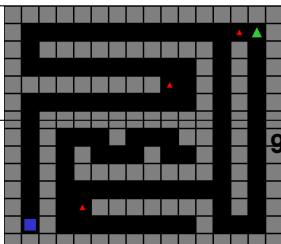
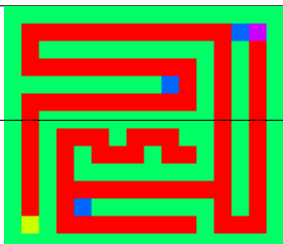
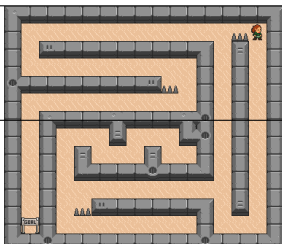
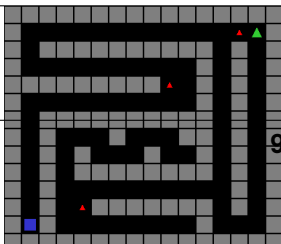
Single-Player/GVGAI/labyrinth.yaml

13.7.1 Description

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

13.7.2 Levels

Table 13: Levels

	Vector	Sprite2D	Block2D				
<table><tr><td>Level ID</td><td>0</td></tr><tr><td>Size</td><td>16x14</td></tr></table>	Level ID	0	Size	16x14			
Level ID	0						
Size	16x14						
<table><tr><td>Level ID</td><td>1</td></tr><tr><td>Size</td><td>16x14</td></tr></table>	Level ID	1	Size	16x14			
Level ID	1						
Size	16x14						
<table><tr><td>Level ID</td><td>2</td></tr><tr><td>Size</td><td>16x14</td></tr></table>	Level ID	2	Size	16x14			
Level ID	2						
Size	16x14						
<table><tr><td>Level ID</td><td>3</td></tr><tr><td>Size</td><td>16x14</td></tr></table>	Level ID	3	Size	16x14			
Level ID	3						
Size	16x14						
<table><tr><td>Level ID</td><td>3</td></tr><tr><td>Size</td><td>16x14</td></tr></table>	Level ID	3	Size	16x14			
Level ID	3						
Size	16x14						
13.7. Labyrinth							

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
Vector				
Sprite2D				
Block2D				

13.7.5 Actions

move

Action Id	Mapping
1	Left
2	Up
3	Right
4	Down

13.7.6 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 . . . X 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 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 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 . . . . . . . . . . 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 . 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 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 . 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

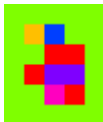
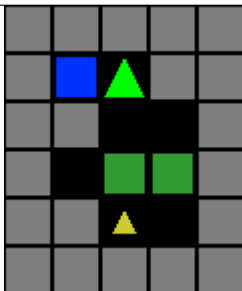
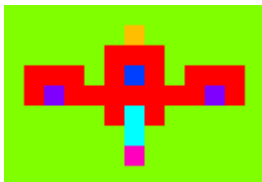
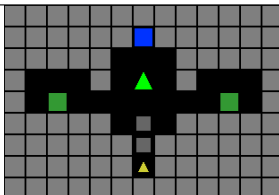
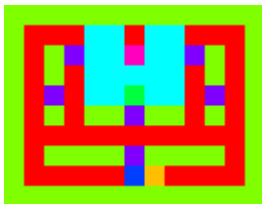
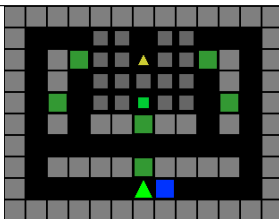
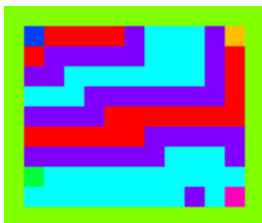
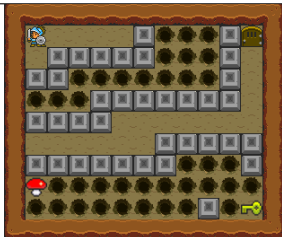
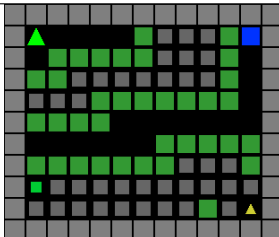
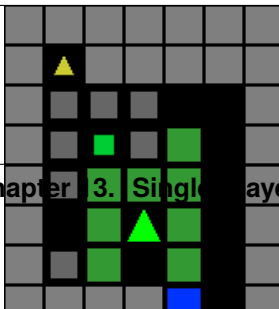
Single-Player/GVGAI/bait.yaml

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

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

13.8.3 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-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 ->	<i>A</i>	<i>o</i>	<i>1</i>	<i>k</i>	<i>g</i>	<i>m</i>	<i>w</i>
Vector							
Sprite2D							
Block2D							

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 . 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
  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

	Vector	Sprite2D	Block2D				
<table><tr><td>Level ID</td><td>0</td></tr><tr><td>Size</td><td>12x12</td></tr></table>	Level ID	0	Size	12x12			
Level ID	0						
Size	12x12						
<table><tr><td>Level ID</td><td>1</td></tr><tr><td>Size</td><td>12x12</td></tr></table>	Level ID	1	Size	12x12			
Level ID	1						
Size	12x12						
<table><tr><td>Level ID</td><td>2</td></tr><tr><td>Size</td><td>12x12</td></tr></table>	Level ID	2	Size	12x12			
Level ID	2						
Size	12x12						
<table><tr><td>Level ID</td><td>3</td></tr><tr><td>Size</td><td>12x11</td></tr></table>	Level ID	3	Size	12x11			
Level ID	3						
Size	12x11						
110			Ch...er				

13.9.3 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-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 ->	<i>A</i>	<i>g</i>	<i>r</i>
Vector			
Sprite2D			
Block2D			

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

```

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...
...gggrg...
...gggggg...
.....
.....
.....
- |
.....
.....
.....
...ggrrgg...
...rggggr...
.A.ggrrgg...
...rggggr...
...ggrrgg...
.....
.....
.....
- |
.....
.....
.....
...rggggg...
...ggrgrg...
...rgggrg...
.A.ggrrgg...
...rgggrg...
...ggrrgg...
.....
.....
.....

```

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)

```
- 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/wall15.png
  Block2D:
    - Shape: triangle
      Color: [0.2, 0.2, 0.2]
      Scale: 0.8
```

13.10 Partially Observable Cook Me Pasta

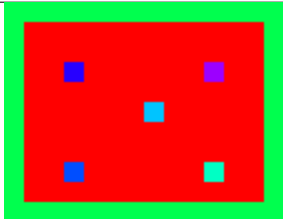
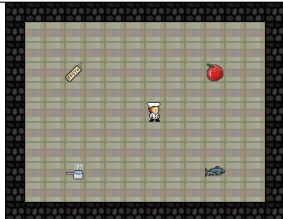
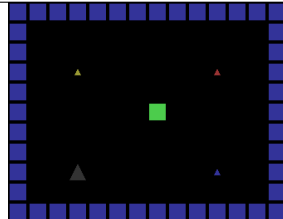
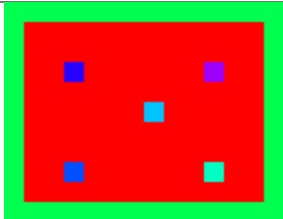
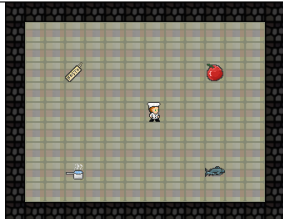
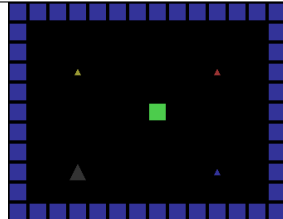
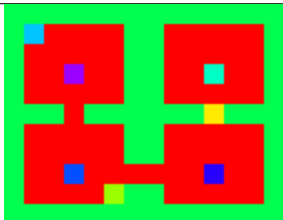
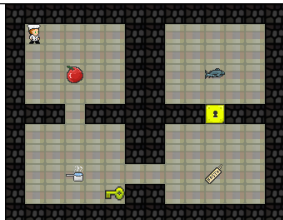
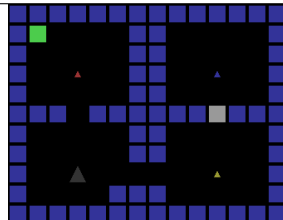
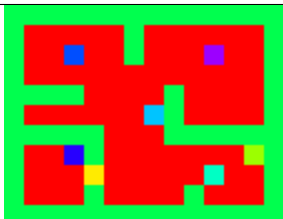
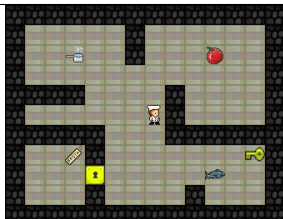
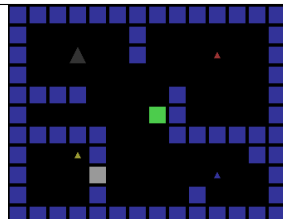
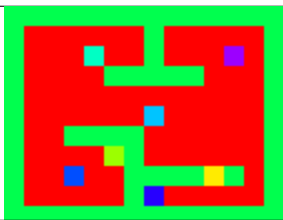
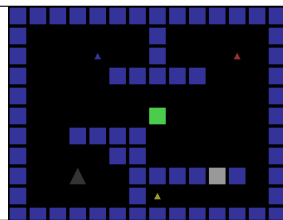
Single-Player/GVGAI/cookmepasta_partially_observable.yaml

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

	Vector	Sprite2D	Block2D				
<table><tr><td>Level ID</td><td>0</td></tr><tr><td>Size</td><td>14x11</td></tr></table>	Level ID	0	Size	14x11			
Level ID	0						
Size	14x11						
<table><tr><td>Level ID</td><td>1</td></tr><tr><td>Size</td><td>14x11</td></tr></table>	Level ID	1	Size	14x11			
Level ID	1						
Size	14x11						
<table><tr><td>Level ID</td><td>2</td></tr><tr><td>Size</td><td>14x11</td></tr></table>	Level ID	2	Size	14x11			
Level ID	2						
Size	14x11						
<table><tr><td>Level ID</td><td>3</td></tr><tr><td>Size</td><td>14x11</td></tr></table>	Level ID	3	Size	14x11			
Level ID	3						
Size	14x11						
<table><tr><td>Level ID</td><td>4</td></tr><tr><td>Size</td><td>14x11</td></tr></table>	Level ID	4	Size	14x11			
Level ID	4						
Size	14x11						

13.10. Partially Observable Cook Me Pasta

Level ID	4
Size	14x11

117

13.10.3 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-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 ->	<i>A</i>	<i>w</i>	<i>k</i>	<i>l</i>	<i>b</i>	<i>p</i>	<i>o</i>	<i>t</i>
Vector								
Sprite2D								
Block2D								

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

(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
www.wwwwwwwwwww
- |
www.wwwwwwwwwww
W.....W.....W
W..b..w...O..W
W.....W.....W
www.wwwwwwwww
W.....Aw....W
www.wwwwwwwww
W..pw.....kw
W...l.....t..W
W..W...W...W
www.wwwwwwwwwww
- |
www.wwwwwwwwwww
W.....W.....W
W...t..w...O..W
W...wwwwwwwww
W.....W.....W
W.....A.....W
W..www.....W
W...kw.....W
W..b..wwwwlw..W
W.....wp.....W
www.wwwwwwwwwww
- |
www.wwwwwwwwwww
W..lA.....W
W..wwwwww.woww
W..t.....W
W..wwwwwwww..W
W..W...k..W..W
W..W..p...W..W
W..W...www..W
W..W...b.....W
W..W...W...W
www.wwwwwwwwwww

```

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

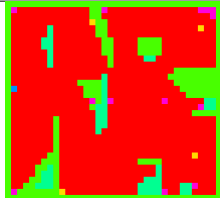
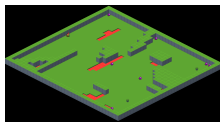
Single-Player/GVGAI/spider-nest.yaml

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

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

13.11.3 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-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 ->	<i>W</i>	<i>G</i>	<i>g</i>	<i>A</i>	<i>N</i>	<i>L</i>
Vector						
Sprite2D						
Block2D						
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 g . . . . . . . . . . W W g . . . . . . . . . .
      . . . g g g W
      W . . . . . . . . . . W W . . . . . . . . . .
      . . . . . . q 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 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

```

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]
```

13.12 Partially Observable Labyrinth

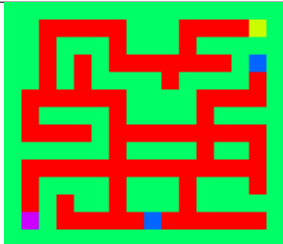
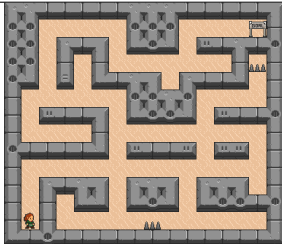
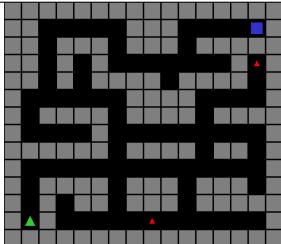
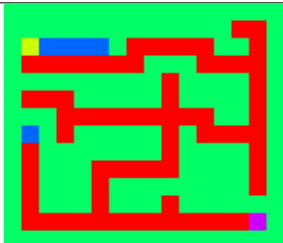
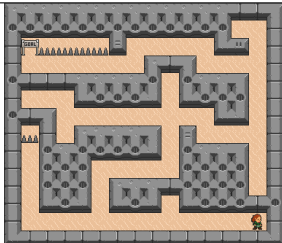
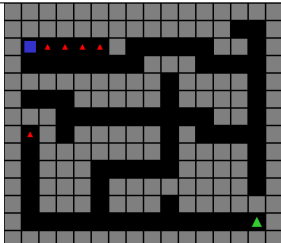
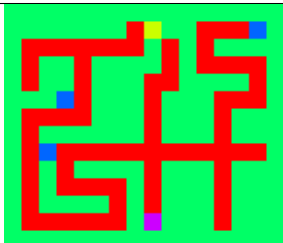
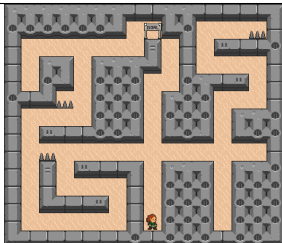
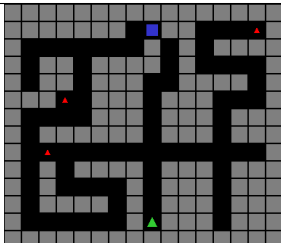
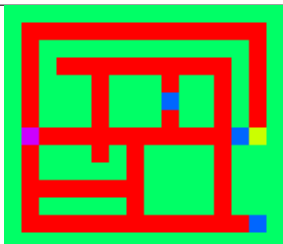
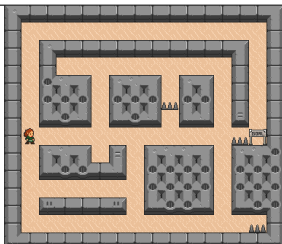
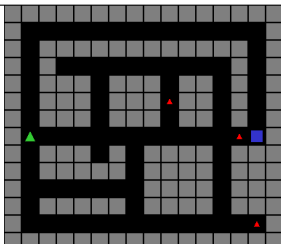
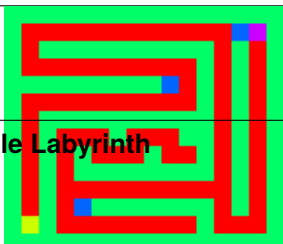
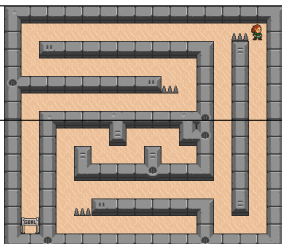
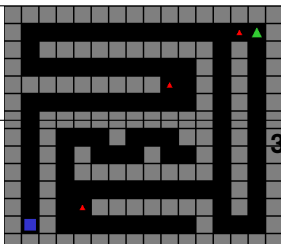
```
Single-Player/GVGAI/labyrinth_partially_observable.yaml
```

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

	Vector	Sprite2D	Block2D				
<table><tr><td>Level ID</td><td>0</td></tr><tr><td>Size</td><td>16x14</td></tr></table>	Level ID	0	Size	16x14			
Level ID	0						
Size	16x14						
<table><tr><td>Level ID</td><td>1</td></tr><tr><td>Size</td><td>16x14</td></tr></table>	Level ID	1	Size	16x14			
Level ID	1						
Size	16x14						
<table><tr><td>Level ID</td><td>2</td></tr><tr><td>Size</td><td>16x14</td></tr></table>	Level ID	2	Size	16x14			
Level ID	2						
Size	16x14						
<table><tr><td>Level ID</td><td>3</td></tr><tr><td>Size</td><td>16x14</td></tr></table>	Level ID	3	Size	16x14			
Level ID	3						
Size	16x14						
							

13.12. Partially Observable Labyrinth

37

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
Vector				
Sprite2D				
Block2D				

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

```

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 . . . 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 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

```

(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 . 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 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 . . . . . . . . . . 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

```

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

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

13.13.3 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-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 ->	<i>b</i>	<i>f</i>	<i>w</i>	<i>h</i>	<i>A</i>
Vector					
Sprite2D					
Block2D					

13.13.5 Actions

move

Action Id	Mapping
1	Left
2	Up
3	Right
4	Down

13.13.6 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:
    - |
      WWWWWWWW
      WW.....W
      ww.hbh.w
      ww.bAb.w
      w..hbh.w
      W.....W
      WWWWWWWW
    - |
      WWWWWWWWWWWW
      W....WWW...W
      w.bb....wAw
      w.b.whhh...w
      W...WWWWWWW
      WWWWWWWWWWWW
    - |
      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
      www..w
      wwwwhw
      WWWWWW
    - |
      WWWWWWWW
      www.hhAw
      www.bb.w

```

(continues on next page)

(continued from previous page)

```

WWW.WWW
WWW.WWW
WWW.WWW
WWW.WWW
W...WWW
W.W...W
W..W.W
WWW..W
WWWWWWW

```

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

	Vector	Sprite2D	Block2D				
<table><tr><td>Level ID</td><td>0</td></tr><tr><td>Size</td><td>13x10</td></tr></table>	Level ID	0	Size	13x10			
Level ID	0						
Size	13x10						
<table><tr><td>Level ID</td><td>1</td></tr><tr><td>Size</td><td>13x10</td></tr></table>	Level ID	1	Size	13x10			
Level ID	1						
Size	13x10						
<table><tr><td>Level ID</td><td>2</td></tr><tr><td>Size</td><td>13x10</td></tr></table>	Level ID	2	Size	13x10			
Level ID	2						
Size	13x10						
<table><tr><td>Level ID</td><td>3</td></tr><tr><td>Size</td><td>13x10</td></tr></table>	Level ID	3	Size	13x10			
Level ID	3						
Size	13x10						
<table><tr><td>Level ID</td><td>4</td></tr><tr><td>Size</td><td>13x10</td></tr></table>	Level ID	4	Size	13x10			
Level ID	4						
Size	13x10						

152

Chapter 13. Single-Player

13.14.3 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-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
Vector									
Sprite2D									
Block2D									

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

```
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)

```
    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

	Vector	Sprite2D	Block2D				
<table><tr><td>Level ID</td><td>0</td></tr><tr><td>Size</td><td>5x6</td></tr></table>	Level ID	0	Size	5x6			
Level ID	0						
Size	5x6						
<table><tr><td>Level ID</td><td>1</td></tr><tr><td>Size</td><td>13x9</td></tr></table>	Level ID	1	Size	13x9			
Level ID	1						
Size	13x9						
<table><tr><td>Level ID</td><td>2</td></tr><tr><td>Size</td><td>13x10</td></tr></table>	Level ID	2	Size	13x10			
Level ID	2						
Size	13x10						
<table><tr><td>Level ID</td><td>3</td></tr><tr><td>Size</td><td>13x11</td></tr></table>	Level ID	3	Size	13x11			
Level ID	3						
Size	13x11						
162							

13.15.3 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-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 ->	<i>A</i>	<i>o</i>	<i>1</i>	<i>k</i>	<i>g</i>	<i>m</i>	<i>w</i>
Vector							
Sprite2D							
Block2D							

13.15.5 Actions

move

Action Id	Mapping
1	Left
2	Up
3	Right
4	Down

13.15.6 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 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

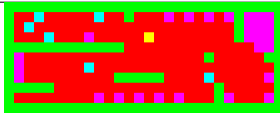
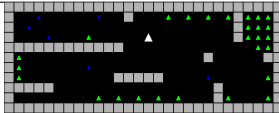
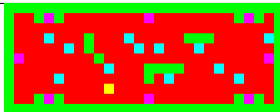
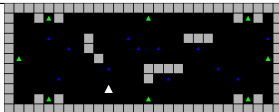
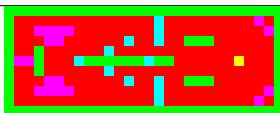
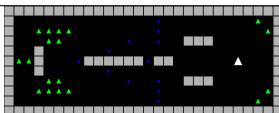
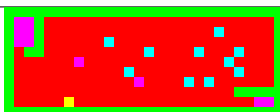
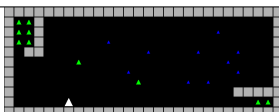
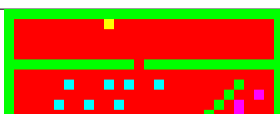
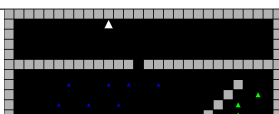
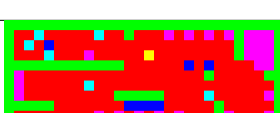
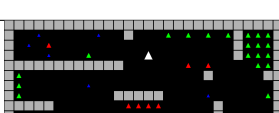
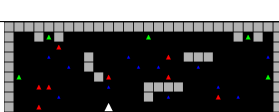
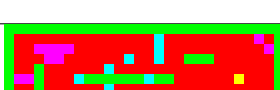
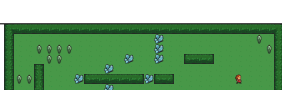
```
Single-Player/GVGAI/random_butterflies.yaml
```

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

	Vector	Sprite2D	Block2D				
<table><tr><td>Level ID</td><td>0</td></tr><tr><td>Size</td><td>28x11</td></tr></table>	Level ID	0	Size	28x11			
Level ID	0						
Size	28x11						
<table><tr><td>Level ID</td><td>1</td></tr><tr><td>Size</td><td>28x11</td></tr></table>	Level ID	1	Size	28x11			
Level ID	1						
Size	28x11						
<table><tr><td>Level ID</td><td>2</td></tr><tr><td>Size</td><td>28x11</td></tr></table>	Level ID	2	Size	28x11			
Level ID	2						
Size	28x11						
<table><tr><td>Level ID</td><td>3</td></tr><tr><td>Size</td><td>28x11</td></tr></table>	Level ID	3	Size	28x11			
Level ID	3						
Size	28x11						
<table><tr><td>Level ID</td><td>4</td></tr><tr><td>Size</td><td>28x12</td></tr></table>	Level ID	4	Size	28x12			
Level ID	4						
Size	28x12						
<table><tr><td>Level ID</td><td>5</td></tr><tr><td>Size</td><td>28x11</td></tr></table>	Level ID	5	Size	28x11			
Level ID	5						
Size	28x11						
<table><tr><td>Level ID</td><td>6</td></tr><tr><td>Size</td><td>28x11</td></tr></table>	Level ID	6	Size	28x11			
Level ID	6						
Size	28x11						
170		Chapter 13. Single-Player					
							

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
Vector					
Sprite2D					
Block2D					

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

```
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 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 . . . . . . . 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 . . . . . 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 . . 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 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 . . . . . . . . . . . . . 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)

```

W W W W W W W W W W W W W W W W W W W W W W W W W W W W W W
- |
W W W W W W W 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 . . . . . . . . . . . . . . . . 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
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 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
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

```

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)

```
- 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

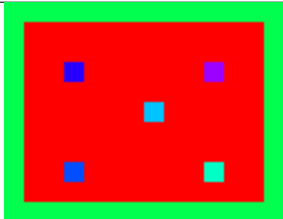
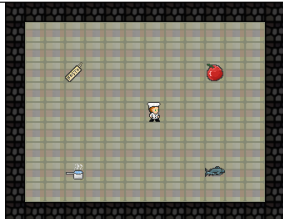
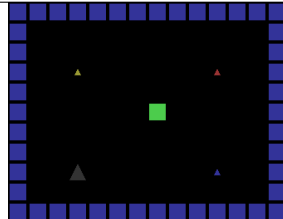
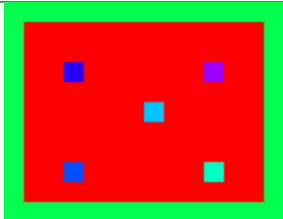
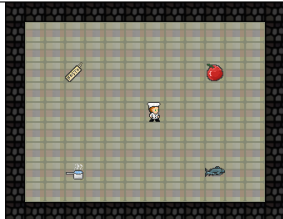
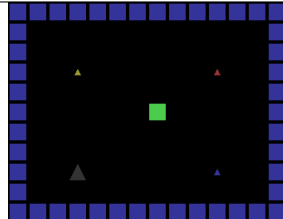
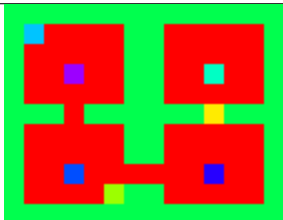
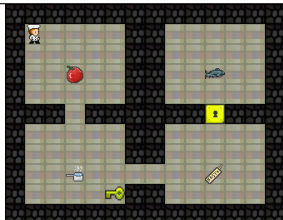
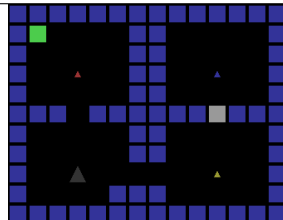
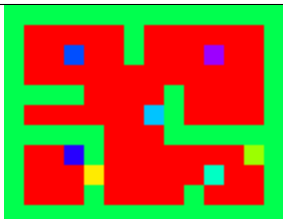
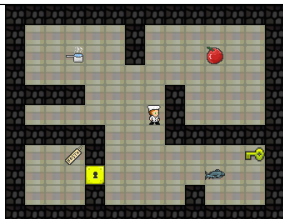
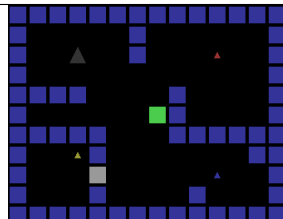
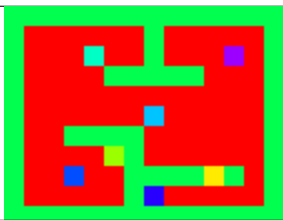
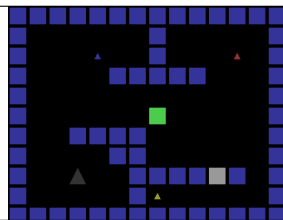
```
Single-Player/GVGAI/cookmepasta.yaml
```

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

	Vector	Sprite2D	Block2D				
<table><tr><td>Level ID</td><td>0</td></tr><tr><td>Size</td><td>14x11</td></tr></table>	Level ID	0	Size	14x11			
Level ID	0						
Size	14x11						
<table><tr><td>Level ID</td><td>1</td></tr><tr><td>Size</td><td>14x11</td></tr></table>	Level ID	1	Size	14x11			
Level ID	1						
Size	14x11						
<table><tr><td>Level ID</td><td>2</td></tr><tr><td>Size</td><td>14x11</td></tr></table>	Level ID	2	Size	14x11			
Level ID	2						
Size	14x11						
<table><tr><td>Level ID</td><td>3</td></tr><tr><td>Size</td><td>14x11</td></tr></table>	Level ID	3	Size	14x11			
Level ID	3						
Size	14x11						
<table><tr><td>Level ID</td><td>4</td></tr><tr><td>Size</td><td>14x11</td></tr></table>	Level ID	4	Size	14x11			
Level ID	4						
Size	14x11						

182

Chapter 13. Single-Player

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
Vector								
Sprite2D								
Block2D								

13.17.5 Actions

move

Action Id	Mapping
1	Left
2	Up
3	Right
4	Down

13.17.6 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:
    - |
      WWWWWWWWWWWWWWW
      W.....W
      W.....W
      W..p.....O..W
      W.....W
      W.....A.....W
      W.....W
      W.....W
      W..b.....t..W
      W.....W
      WWWWWWWWWWWWWWW
    - |
      WWWWWWWWWWWWWWW
      W.....W
      W.....W
      W..p.....O..W
      W.....W
      W.....A.....W
      W.....W
      W.....W
      W..b.....t..W
      W.....W
      WWWWWWWWWWWWWWW
    - |
      WWWWWWWWWWWWWWW
      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.....w
www...w...w
w.....Aw....w
www...www...w
w..pw.....kw
w..l.....t..w
w..w...w...w
wwwwwwwwwwwwww
- |
wwwwwwwwwwwwww
w.....w.....w
w..t..w...O..w
w...www...w
w.....w.....w
w.....A.....w
w..www.....w
w...kw.....w
w..b..wwwlw.w
w....wp....w
wwwwwwwwwwwwww
- |
wwwwwwwwwwwwww
w..lA.....w
w..www...w...w
w..t.....w
w..www...w...w
w..w...k..w..w
w..w..p...w..w
w..w...www...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

	Vector	Sprite2D	Block2D				
<table><tr><td>Level ID</td><td>0</td></tr><tr><td>Size</td><td>5x6</td></tr></table>	Level ID	0	Size	5x6			
Level ID	0						
Size	5x6						
<table><tr><td>Level ID</td><td>1</td></tr><tr><td>Size</td><td>13x9</td></tr></table>	Level ID	1	Size	13x9			
Level ID	1						
Size	13x9						
<table><tr><td>Level ID</td><td>2</td></tr><tr><td>Size</td><td>13x10</td></tr></table>	Level ID	2	Size	13x10			
Level ID	2						
Size	13x10						
<table><tr><td>Level ID</td><td>3</td></tr><tr><td>Size</td><td>13x11</td></tr></table>	Level ID	3	Size	13x11			
Level ID	3						
Size	13x11						
192							

Chapter 3. Single layer

13.18.3 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-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 ->	<i>A</i>	<i>o</i>	<i>1</i>	<i>k</i>	<i>g</i>	<i>m</i>	<i>w</i>
Vector							
Sprite2D							
Block2D							

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 g 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 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 . . . 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

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

```
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
Vector						
Sprite2D						
Block2D						

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

```
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)

Levels:

[illegible]

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)

```

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

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

13.20.3 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-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 ->	<i>A</i>	<i>x</i>	<i>+</i>	<i>g</i>	<i>3</i>	<i>w</i>
Vector						
Sprite2D						
Block2D						

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
    - |
      WWWWWWWWWWWWW
      W.....W
      W....W....W
      W....W....W

```

(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

	Vector	Sprite2D	Block2D				
<table><tr><td>Level ID</td><td>0</td></tr><tr><td>Size</td><td>13x9</td></tr></table>	Level ID	0	Size	13x9			
Level ID	0						
Size	13x9						
<table><tr><td>Level ID</td><td>1</td></tr><tr><td>Size</td><td>13x9</td></tr></table>	Level ID	1	Size	13x9			
Level ID	1						
Size	13x9						
<table><tr><td>Level ID</td><td>2</td></tr><tr><td>Size</td><td>13x9</td></tr></table>	Level ID	2	Size	13x9			
Level ID	2						
Size	13x9						
<table><tr><td>Level ID</td><td>3</td></tr><tr><td>Size</td><td>11x9</td></tr></table>	Level ID	3	Size	11x9			
Level ID	3						
Size	11x9						
<table><tr><td>Level ID</td><td>4</td></tr><tr><td>Size</td><td>7x7</td></tr></table>	Level ID	4	Size	7x7			
Level ID	4						
Size	7x7						

218

Chapter 13. Single-Player

13.21.3 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-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 ->	<i>b</i>	<i>w</i>	<i>h</i>	<i>A</i>
Vector				
Sprite2D				
Block2D				

13.21.5 Actions

move

Action Id	Mapping
1	Left
2	Up
3	Right
4	Down

13.21.6 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..www
      w.....w.h.w
      w.b.....ww
      w.....ww
      wwwwwwwwwwwww
    - |
      wwwwwwwwwwwww
      wwA.....ww
      wwwwb.....ww
      wwww.h.....w
      wwww.....w
      w..b...wbwww
      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..www
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)

```

# 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

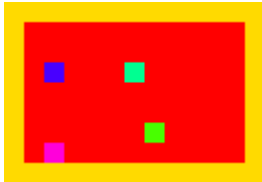
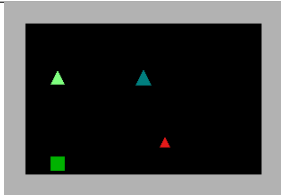
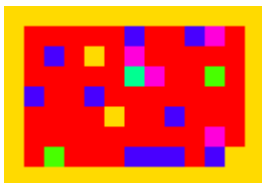
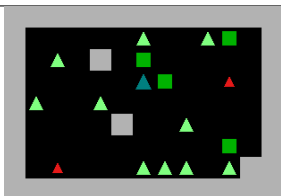
```
Single-Player/GVGAI/zelda.yaml
```

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

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

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 ->	<i>A</i>	<i>x</i>	<i>+</i>	<i>g</i>	<i>3</i>	<i>w</i>
Vector						
Sprite2D						
Block2D						

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

```
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:
    - |
      WWWWWWWWWWWWWW
      W.....W
      W.....W
      W.+...A....W
      W.....W
      W.....W
      W.....W
      W.....3....W
      W.g.....W
      WWWWWWWWWWWWWW
    - |
      WWWWWWWWWWWWWW
      W.....+..g.w
      W.+..w.g....w
      W.....Ag..3.w
      W+.+. ....W
      W....W..+...W
      W.....g.w
      W.3...+++..ww
      WWWWWWWWWWWWWW
    - |
      WWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWW
```

(continues on next page)

(continued from previous page)

```

W.....+....3.....W
W.....W
W...A.....3.....W
W.....W
W.....3.....W
W.....WWWWWWWWWWWW
W.....g.W
WWWWWWWWWWWWWWWWWWWW

```

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

	Vector	Sprite2D	Block2D				
<table><tr><td>Level ID</td><td>0</td></tr><tr><td>Size</td><td>13x10</td></tr></table>	Level ID	0	Size	13x10			
Level ID	0						
Size	13x10						
<table><tr><td>Level ID</td><td>1</td></tr><tr><td>Size</td><td>13x10</td></tr></table>	Level ID	1	Size	13x10			
Level ID	1						
Size	13x10						
<table><tr><td>Level ID</td><td>2</td></tr><tr><td>Size</td><td>13x10</td></tr></table>	Level ID	2	Size	13x10			
Level ID	2						
Size	13x10						
<table><tr><td>Level ID</td><td>3</td></tr><tr><td>Size</td><td>13x10</td></tr></table>	Level ID	3	Size	13x10			
Level ID	3						
Size	13x10						
<table><tr><td>Level ID</td><td>4</td></tr><tr><td>Size</td><td>13x10</td></tr></table>	Level ID	4	Size	13x10			
Level ID	4						
Size	13x10						

234

Chapter 13. Single-Player

13.23.3 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-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
Vector									
Sprite2D									
Block2D									

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 . . 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 . . . . 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 . . . . . . 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

```

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)

```
    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

	Vector	Sprite2D	Block2D				
<table><tr><td>Level ID</td><td>0</td></tr><tr><td>Size</td><td>12x12</td></tr></table>	Level ID	0	Size	12x12			
Level ID	0						
Size	12x12						
<table><tr><td>Level ID</td><td>1</td></tr><tr><td>Size</td><td>12x12</td></tr></table>	Level ID	1	Size	12x12			
Level ID	1						
Size	12x12						
<table><tr><td>Level ID</td><td>2</td></tr><tr><td>Size</td><td>12x12</td></tr></table>	Level ID	2	Size	12x12			
Level ID	2						
Size	12x12						
<table><tr><td>Level ID</td><td>3</td></tr><tr><td>Size</td><td>12x11</td></tr></table>	Level ID	3	Size	12x11			
Level ID	3						
Size	12x11						
244			Ch...er				

13.24.3 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-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 ->	<i>A</i>	<i>g</i>	<i>r</i>
Vector			
Sprite2D			
Block2D			

13.24.5 Actions

move

Action Id	Mapping
1	Left
2	Up
3	Right
4	Down

13.24.6 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)

```

...gggrg...
...ggggg...
.....
.....
.....
- |
.....
.....
.....
...grrgg...
...rgggr...
.A.grrgg...
...rgggr...
...grrgg...
.....
.....
.....
- |
.....
.....
.....
...rgggg...
...grrg...
...rgggr...
.A.grggg...
...rgggr...
...grrgg...
.....
.....
.....

```

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)

```
- reward: 1
```

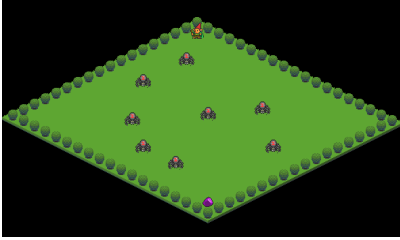
Objects:

- **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

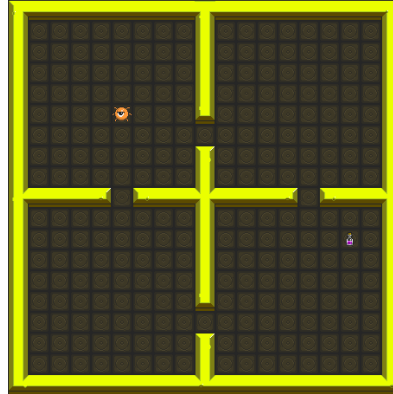
- **Name:** ground
MapCharacter: g
Observers:
Sprite2D:
 - **Image:** gvgai/oryx/floorTileOrange.png**Block2D:**
 - **Shape:** square
Color: [0.2, 0.2, 0.4]
Scale: 0.7

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

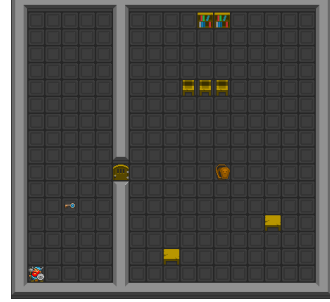
- **Name:** rock
MapCharacter: r
Observers:
Sprite2D:
 - **Image:** gvgai/oryx/wall15.png**Block2D:**
 - **Shape:** triangle
Color: [0.2, 0.2, 0.2]
Scale: 0.8

Spiders

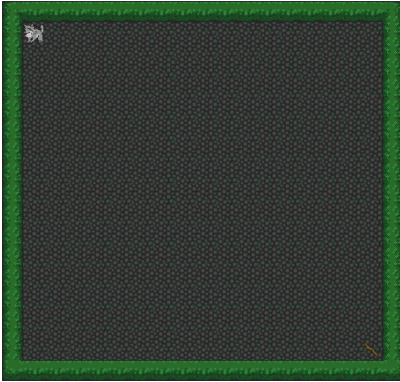
A port of the games provided in the <https://github.com/maximecb/gym-minigrid> Dynamic obstacles environment, but you're a gnome avoiding ghosts to get to a gem.

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

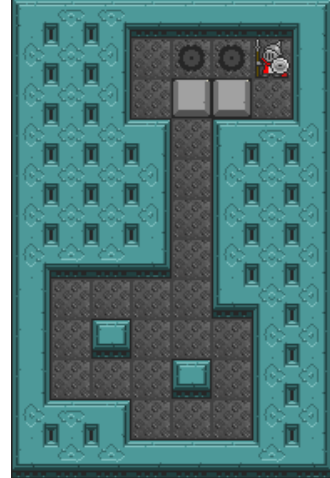
A port of the games provided in the <https://github.com/maximecb/gym-minigrid> environment, but you're a drunk dwarf trying find your keys that you've dropped to get to your bed (which is a coffin?? Wierd.).

Doggo

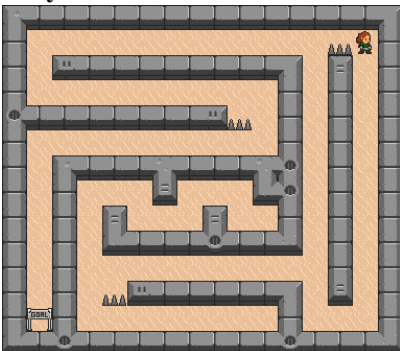
A port of the games provided in the <https://github.com/maximecb/gym-minigrid> Empty environment, but you're a doggo fetching a stick.

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

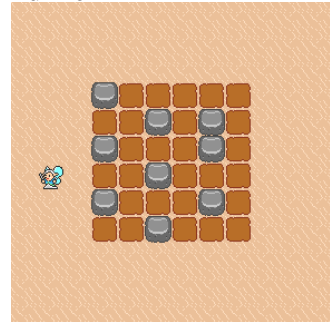
Push the boxes onto the marked spaces, once a box has moved onto a space, it cannot be moved

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)

14.1 GriddlyRTS

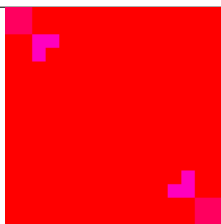
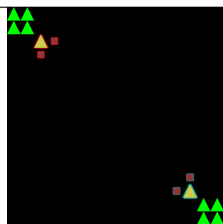
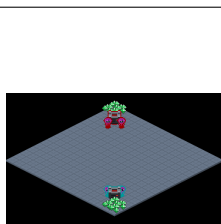
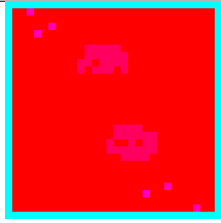
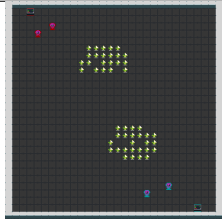
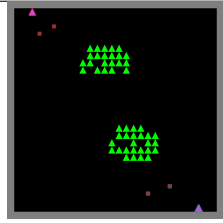
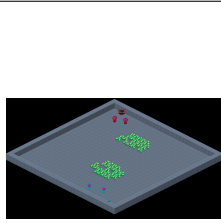
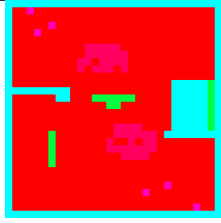
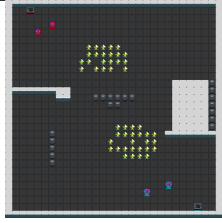
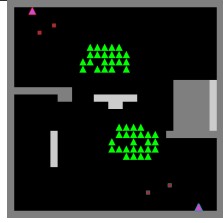
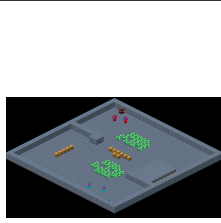
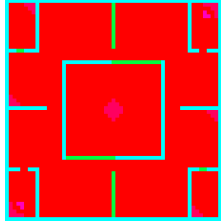
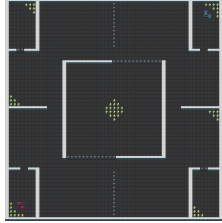
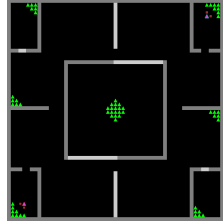
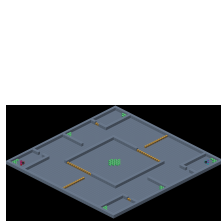
RTS/GriddlyRTS.yaml

14.1.1 Description

An RTS Game. There's aliens and stuff.

14.1.2 Levels

Table 1: Levels

	Vector	Sprite2D	Block2D	Isometric				
<table><tr><td>Level ID</td><td>0</td></tr><tr><td>Size</td><td>16x16</td></tr></table>	Level ID	0	Size	16x16				
Level ID	0							
Size	16x16							
<table><tr><td>Level ID</td><td>1</td></tr><tr><td>Size</td><td>30x30</td></tr></table>	Level ID	1	Size	30x30				
Level ID	1							
Size	30x30							
<table><tr><td>Level ID</td><td>2</td></tr><tr><td>Size</td><td>30x30</td></tr></table>	Level ID	2	Size	30x30				
Level ID	2							
Size	30x30							
<table><tr><td>Level ID</td><td>3</td></tr><tr><td>Size</td><td>57x58</td></tr></table>	Level ID	3	Size	57x58				
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.

```
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 ->	miner-als	worker	ranged	com-bat	fixed_wall	mov-able_wall	base	bar-racks_disabled	bar-racks
Map Char ->	M	H	r	c	W	w	A	b	B
Vector									
Sprite2D									
Block2D									
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

```
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)

[illegible]

(continues on next page)

(continued from previous page)

[illegible]

(continues on next page)

(continued from previous page)

[illegible]

(continues on next page)

Actions:

(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
      Block2D:
        - Shape: triangle
          Color: [ 0.0, 1.0, 0.0 ]
          Scale: 1.0
      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
    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:

```

(continues on next page)

(continued from previous page)

```

    - 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

- 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

```

(continues on next page)

(continued from previous page)

```

    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:
    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

```

(continues on next page)

(continued from previous page)

```
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

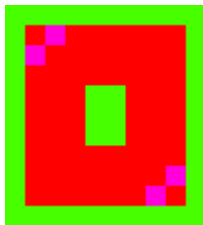
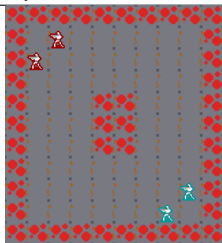
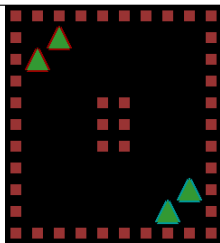
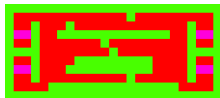
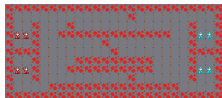
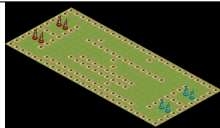
```
RTS/Stratega/push-mania.yaml
```

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

	Vector	Sprite2D	Block2D	Isometric				
<table><tr><td>Level ID</td><td>0</td></tr><tr><td>Size</td><td>10x11</td></tr></table>	Level ID	0	Size	10x11				
Level ID	0							
Size	10x11							
<table><tr><td>Level ID</td><td>1</td></tr><tr><td>Size</td><td>25x11</td></tr></table>	Level ID	1	Size	25x11				
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)

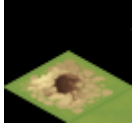
    # Replace with your own control algorithm!
    for s in range(1000):
        obs, reward, done, info = env.step(env.action_space.sample())
        for p in range(env.player_count):
            env.render(observer=p) # Renders the environment from the perspective of a
            ↪ single player

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

        if done:
            env.reset()
```

14.2.4 Objects

Table 4: Tiles

Name ->	hole	pusher
Map Char ->	<i>H</i>	<i>p</i>
Vector		
Sprite2D		
Block2D		
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 . . H
    H p1 p1 H . . H 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 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 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

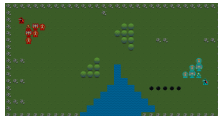
RTS/Stratega/kill-the-king.yaml

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

		Vector	Sprite2D	Block2D	Isometric				
<table><tr><td>Level ID</td><td>0</td></tr><tr><td>Size</td><td>32x17</td></tr></table>		Level ID	0	Size	32x17				
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)

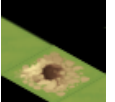
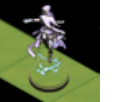
    # Replace with your own control algorithm!
    for s in range(1000):
        obs, reward, done, info = env.step(env.action_space.sample())
        for p in range(env.player_count):
            env.render(observer=p) # Renders the environment from the perspective of a
↪ single player

        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 ->	<i>M</i>	<i>W</i>	<i>F</i>	<i>H</i>	<i>h</i>	<i>w</i>	<i>a</i>	<i>k</i>
Vector								
Sprite2D								
Block2D								
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

move_1

14.3.6 YAML

[illegible]

(continues on next page)

[illegible]

```
- 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)

```

- 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

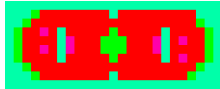
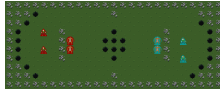
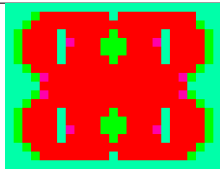
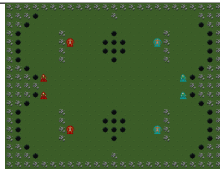
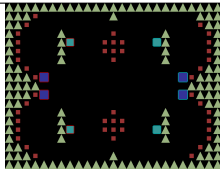
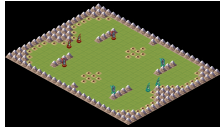
RTS/Stratega/heal-or-die.yaml

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

	Vector	Sprite2D	Block2D	Isometric				
<table><tr><td>Level ID</td><td>0</td></tr><tr><td>Size</td><td>25x10</td></tr></table>	Level ID	0	Size	25x10				
Level ID	0							
Size	25x10							
<table><tr><td>Level ID</td><td>1</td></tr><tr><td>Size</td><td>25x19</td></tr></table>	Level ID	1	Size	25x19				
Level ID	1							
Size	25x19							

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)

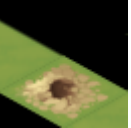
    # Replace with your own control algorithm!
    for s in range(1000):
        obs, reward, done, info = env.step(env.action_space.sample())
        for p in range(env.player_count):
            env.render(observer=p) # Renders the environment from the perspective of a
↪ single player

        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 ->	<i>M</i>	<i>H</i>	<i>h</i>	<i>w</i>
Vector				
Sprite2D				
Block2D				
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

```

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 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:

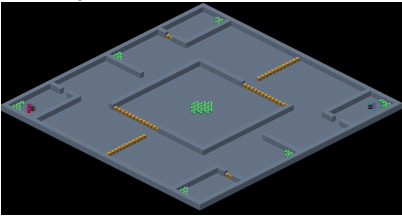
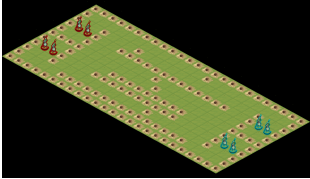
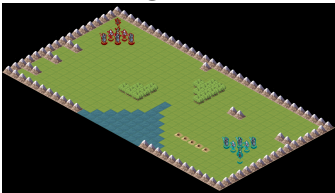
```

(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.		

15.1 Robot Tag 12v12

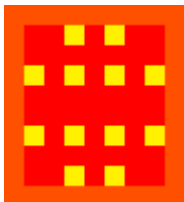
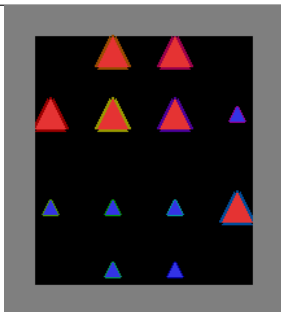
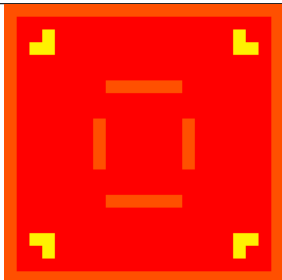
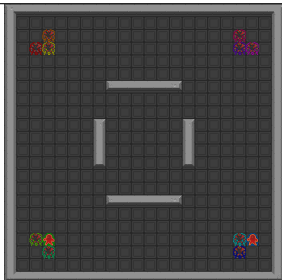
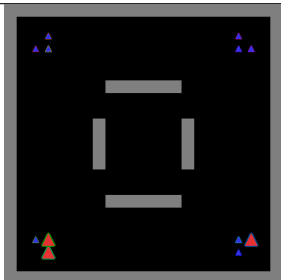
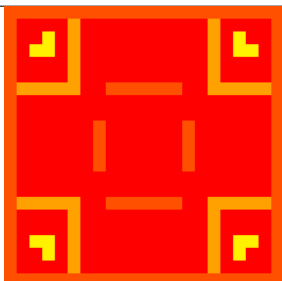
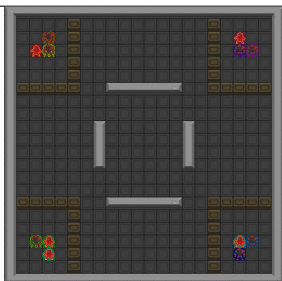
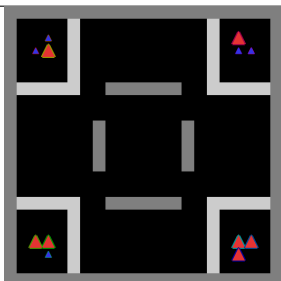
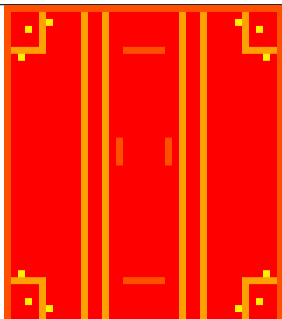
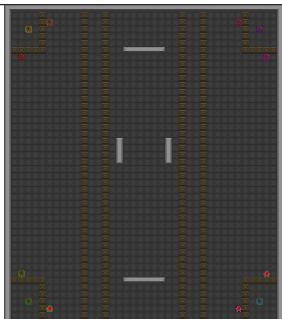
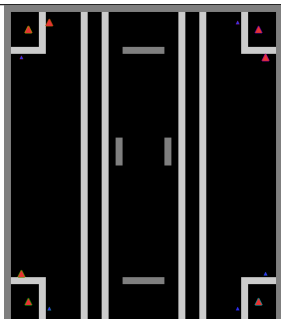
Multi-Agent/robot_tag_12.yaml

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

	Vector	Sprite2D	Block2D				
<table><tr><td>Level ID</td><td>0</td></tr><tr><td>Size</td><td>9x10</td></tr></table>	Level ID	0	Size	9x10			
Level ID	0						
Size	9x10						
<table><tr><td>Level ID</td><td>1</td></tr><tr><td>Size</td><td>22x22</td></tr></table>	Level ID	1	Size	22x22			
Level ID	1						
Size	22x22						
<table><tr><td>Level ID</td><td>2</td></tr><tr><td>Size</td><td>22x22</td></tr></table>	Level ID	2	Size	22x22			
Level ID	2						
Size	22x22						
<table><tr><td>Level ID</td><td>3</td></tr><tr><td>Size</td><td>40x46</td></tr></table>	Level ID	3	Size	40x46			
Level ID	3						
Size	40x46						

15.1. Robot Tag 12v12

293

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 ->	<i>f</i>	<i>m</i>	<i>W</i>
Vector			
Sprite2D			
Block2D			

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

```

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

- 1

[illegible]

(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	.	.	.	.	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	.	.	.	⌋
→	W	W	m	m	m	m	m	.	.	W	W	W	W	W	W	.	.	m	m	⌋
→	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	⌋
→	W	W	.	.	.	.	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	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	⌋

(continues on next page)

(continued from previous page)

[illegible]

(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

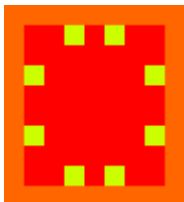
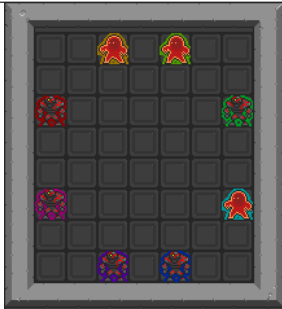
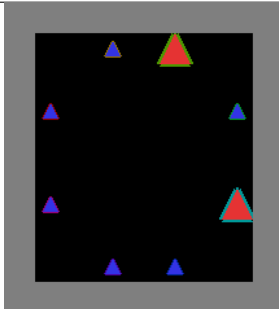
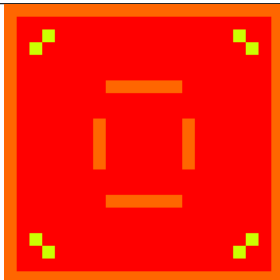
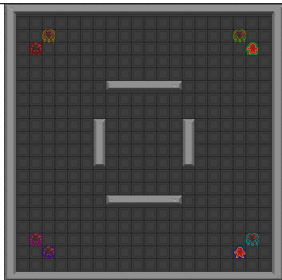
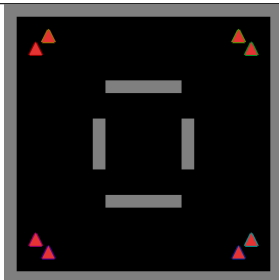
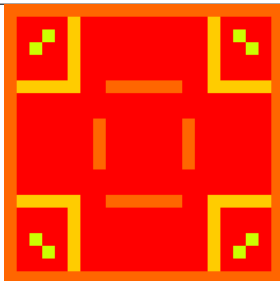
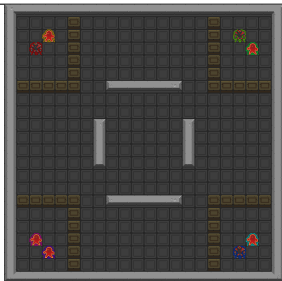
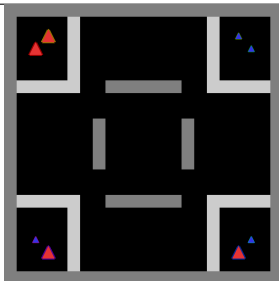
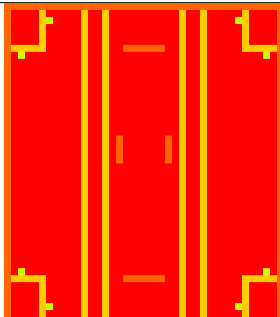
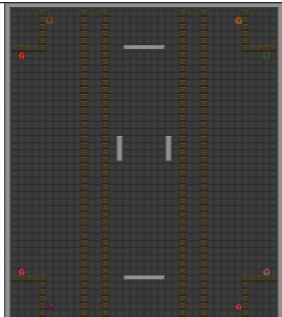
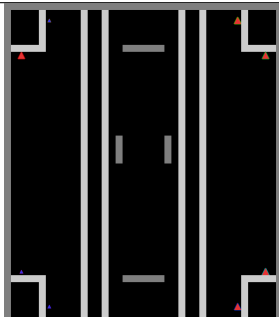
```
Multi-Agent/robot_tag_8.yaml
```

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

	Vector	Sprite2D	Block2D				
<table><tr><td>Level ID</td><td>0</td></tr><tr><td>Size</td><td>9x10</td></tr></table>	Level ID	0	Size	9x10			
Level ID	0						
Size	9x10						
<table><tr><td>Level ID</td><td>1</td></tr><tr><td>Size</td><td>22x22</td></tr></table>	Level ID	1	Size	22x22			
Level ID	1						
Size	22x22						
<table><tr><td>Level ID</td><td>2</td></tr><tr><td>Size</td><td>22x22</td></tr></table>	Level ID	2	Size	22x22			
Level ID	2						
Size	22x22						
<table><tr><td>Level ID</td><td>3</td></tr><tr><td>Size</td><td>40x46</td></tr></table>	Level ID	3	Size	40x46			
Level ID	3						
Size	40x46						

304

Chapter 15. Multi-Agent

15.2.3 Code Example

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

```
import gym
import griddly

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 ->	<i>f</i>	<i>m</i>	<i>W</i>
Vector			
Sprite2D			
Block2D			

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)

- **eq**: [tagged_count, 0]

Levels:

- 1

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

- 1

[illegible]

(continues on next page)

(continued from previous page)

→ W	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	.	.	.	.	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
→ W	W	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	⌋
→ W	W	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	⌋
→ 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
→ 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	⌋

(continues on next page)

(continued from previous page)

[illegible]

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

	Vector	Sprite2D	Block2D				
<table><tr><td>Level ID</td><td>0</td></tr><tr><td>Size</td><td>9x10</td></tr></table>	Level ID	0	Size	9x10			
Level ID	0						
Size	9x10						
<table><tr><td>Level ID</td><td>1</td></tr><tr><td>Size</td><td>9x10</td></tr></table>	Level ID	1	Size	9x10			
Level ID	1						
Size	9x10						
<table><tr><td>Level ID</td><td>2</td></tr><tr><td>Size</td><td>18x20</td></tr></table>	Level ID	2	Size	18x20			
Level ID	2						
Size	18x20						

15.3.3 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-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 ->	<i>f</i>	<i>b</i>	<i>r</i>	<i>g</i>	<i>W</i>
Vector					
Sprite2D					
Block2D					

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

```

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 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 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 . . 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

```

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

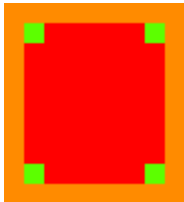
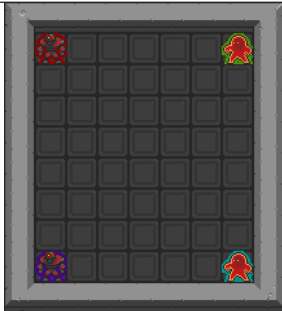
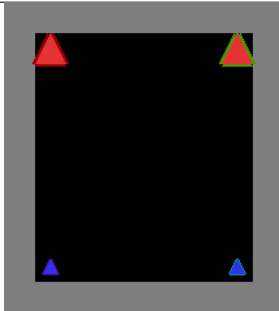
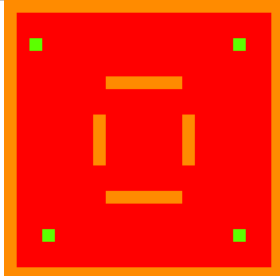
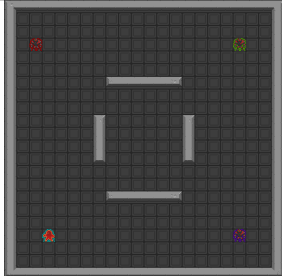
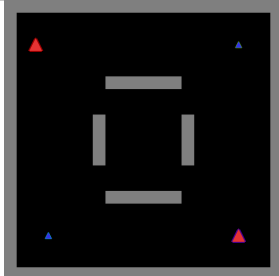
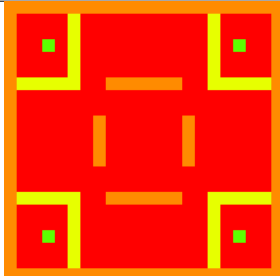
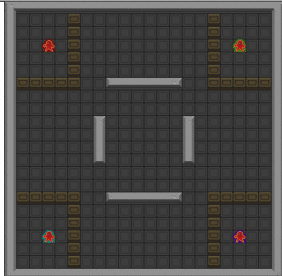
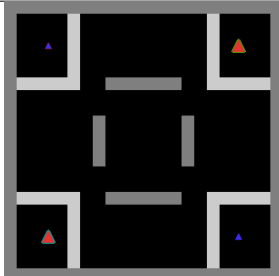
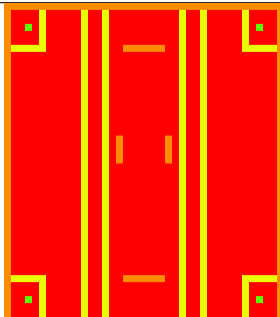
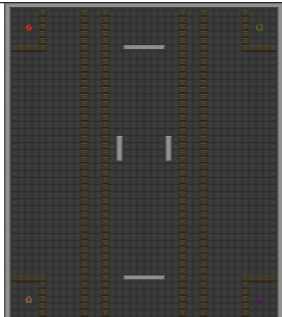
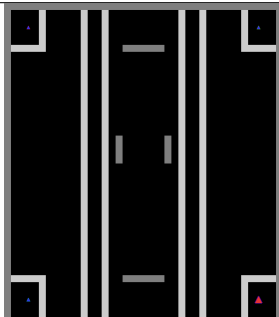
```
Multi-Agent/robot_tag_4.yaml
```

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

	Vector	Sprite2D	Block2D				
<table><tr><td>Level ID</td><td>0</td></tr><tr><td>Size</td><td>9x10</td></tr></table>	Level ID	0	Size	9x10			
Level ID	0						
Size	9x10						
<table><tr><td>Level ID</td><td>1</td></tr><tr><td>Size</td><td>22x22</td></tr></table>	Level ID	1	Size	22x22			
Level ID	1						
Size	22x22						
<table><tr><td>Level ID</td><td>2</td></tr><tr><td>Size</td><td>22x22</td></tr></table>	Level ID	2	Size	22x22			
Level ID	2						
Size	22x22						
<table><tr><td>Level ID</td><td>3</td></tr><tr><td>Size</td><td>40x46</td></tr></table>	Level ID	3	Size	40x46			
Level ID	3						
Size	40x46						

322

Chapter 15. Multi-Agent

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 ->	<i>f</i>	<i>m</i>	<i>W</i>
Vector			
Sprite2D			
Block2D			

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

```
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)

- **eq**: [tagged_count, 0]

- 1

(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			.	.	m	.	.	.	.	.	.	.	.	.	m	.	.	.	⌋	
→	W			.	.	m	.	.	.	.	.	.	.	.	.	m	.	.	.	⌋	
→	W			.	.	m	.	.	.	.	.	.	.	.	.	m	.	.	.	⌋	
→	W			f1	.	m	.	.	.	.	.	.	.	.	.	m	.	f2	.	⌋	
→	W			.	.	m	.	.	.	.	.	.	.	.	.	m	.	.	.	⌋	
→	W			.	.	m	.	.	.	.	.	.	.	.	.	m	.	.	.	⌋	
→	W	m	m	m	m	m	.	.	W	W	W	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	m	m	m	m	m	.	.	W	W	W	W	W	W	.	.	m	m	m	⌋	
→	W			.	.	m	.	.	.	.	.	.	.	.	.	.	m	.	.	⌋	
→	W			.	.	m	.	.	.	.	.	.	.	.	.	.	m	.	.	⌋	
→	W			.	.	m	.	.	.	.	.	.	.	.	.	.	m	.	.	⌋	
→	W			f3	.	m	.	.	.	.	.	.	.	.	.	.	m	.	f4	⌋	
→	W			.	.	m	.	.	.	.	.	.	.	.	.	.	m	.	.	⌋	
→	W			.	.	m	.	.	.	.	.	.	.	.	.	.	m	.	.	⌋	
→	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	⌋	
→	W																			⌋	

(continues on next page)

(continues on next page)

(continued from previous page)

[illegible]

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

```

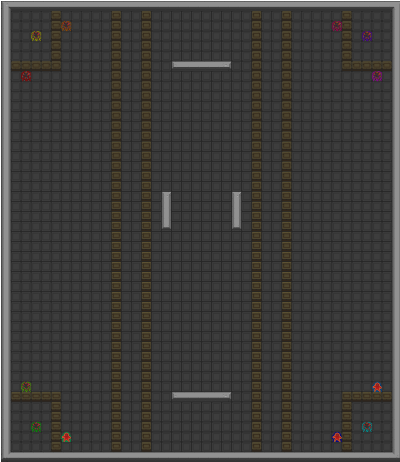
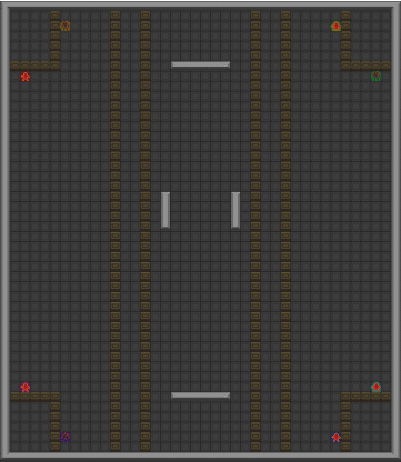
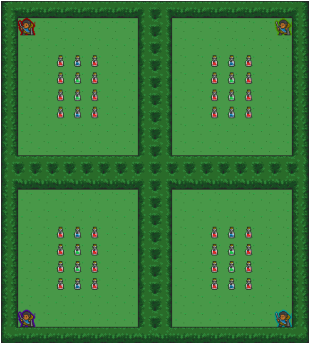
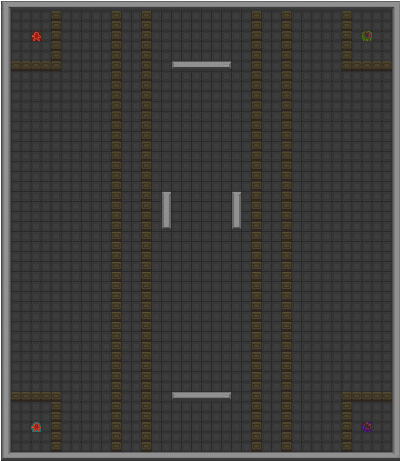
(continues on next page)

(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.		

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 RLLib and Pytorch to run the provided examples.

You can install RLLib and pytorch using the following command:

```
pip install ray[rllib]==1.2.0 torch==1.8.0
```

All RLLib examples can be found in `python/examples/rllib/`

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': {
```

(continues on next page)

(continued from previous page)

```
    'frequency': 100000
},

'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 the simple `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)

```

(continues on next page)

(continued from previous page)

```

    return logits, state

def value_function(self):
    return self._value

```

16.4.2 Global Average Pooling

Griddly environments' observation spaces 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(),

```

(continues on next page)

(continued from previous page)

```

        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(),
    )

    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 Weights and Biases (WandB)

16.6 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,
    },

```

(continues on next page)

(continued from previous page)

```
}  
    ...  
}
```

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.

16.6.1 Uploading Videos to WandB

To automatically upload videos to WandB, the `VideoCallback` can be set in the RLLib config:

```
'config': {  
    ...,  
  
    'callbacks': VideoCallback,  
  
    ...  
}
```

16.7 Recording Environment Actions

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,  
  
    ...  
}
```



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.

16.7.1 Uploading Environment Events to WandB

To automatically upload action events to WandB, the `ActionTrackerCallback` can be set in the RLLib config:

```
'config': {
    ...,
    'callbacks': ActionTrackerCallback,
    ...
}
```


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 IMPALA to train on the *Partially Observable Clusters* Environment.

The agent in the *Partially Observable Clusters* environment has a 5x5 partially observable ego-centric view.

By default the agent sees a *VECTOR* view of the environment. This view is passed to a *Global Average Pooling 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](#)

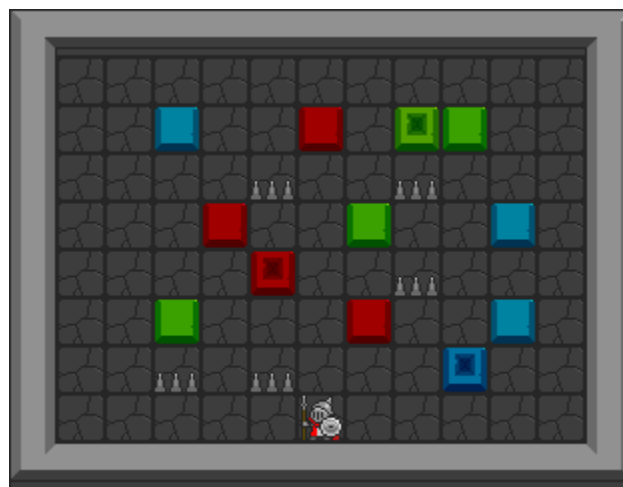


Fig. 1: The Clusters environment as seen from the “Global Observer” view.

```
import os
import sys
```

(continues on next page)

(continued from previous page)

```

import ray
from ray import tune
from ray.rllib.agents.impala import ImpalaTrainer
from ray.rllib.models import ModelCatalog
from ray.tune.registry import register_env

from griddly import gd
from griddly.util.rllib.torch import GAPAgent
from griddly.util.rllib.wrappers.core import RLlibEnv

if __name__ == '__main__':

    ray.init(num_gpus=1)

    env_name = "ray-griddly-env"

    register_env(env_name, RLlibEnv)
    ModelCatalog.register_custom_model("GAP", GAPAgent)

    max_training_steps = 1000000000

    config = {
        'framework': 'torch',
        'num_workers': 8,
        'num_envs_per_worker': 4,

        'model': {
            'custom_model': 'GAP',
            'custom_model_config': {}
        },
        'env': env_name,
        'env_config': {
            'record_video_config': {
                'frequency': 100000
            },

            'random_level_on_reset': True,
            'yaml_file': 'Single-Player/GVGAI/clusters_partially_observable.yaml',
            'global_observer_type': gd.ObserverType.SPRITE_2D,
            'max_steps': 1000,
        },
        'entropy_coeff_schedule': [
            [0, 0.01],
            [max_training_steps, 0.0]
        ],
        'lr_schedule': [
            [0, 0.0005],
            [max_training_steps, 0.0]
        ]
    }

    stop = {

```

(continues on next page)

(continued from previous page)

```
    "timesteps_total": max_training_steps,  
}  
  
result = tune.run(ImpalaTrainer, config=config, stop=stop)
```


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
def _create_env(env_config):
    env = RLLibEnv(env_config)
    return RLLibMultiAgentWrapper(env, env_config)

register_env(env_name, _create_env)
```

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 the *Foragers* environment is trained for 10M steps using IMPALA and a *Simple Convolutional Agent* `<simple_conv_agent>`.

The Foragers environment as seen from the “Global Observer” view.

The following code will run the Foragers example for 10M steps using IMPALA to train.

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 ray
from ray import tune
from ray.rllib.agents.ppo import PPOTrainer
```

(continues on next page)



(continued from previous page)

```

from ray.rllib.models import ModelCatalog
from ray.tune.registry import register_env

from griddly import gd
from griddly.util.rllib.torch.agents.conv_agent import SimpleConvAgent
from griddly.util.rllib.wrappers.core import RllibMultiAgentWrapper, RllibEnv

if __name__ == '__main__':
    sep = os.pathsep
    os.environ['PYTHONPATH'] = sep.join(sys.path)

    ray.init(num_gpus=1)

    env_name = 'ray-ma-env'

    # Create the environment and wrap it in a multi-agent wrapper for self-play
    def _create_env(env_config):
        env = RllibEnv(env_config)
        return RllibMultiAgentWrapper(env, env_config)

    register_env(env_name, _create_env)

    ModelCatalog.register_custom_model('SimpleConv', SimpleConvAgent)

    max_training_steps = 100000000

    config = {
        'framework': 'torch',
        'num_workers': 8,
        'num_envs_per_worker': 2,

        'num_gpus': 1,

        'model': {
            'custom_model': 'SimpleConv',

```

(continues on next page)

(continued from previous page)

```

        'custom_model_config': {}
    },
    'env': env_name,
    'env_config': {
        # in the griddly environment we set a variable to let the training_
↪environment
        # know if that player is no longer active
        # The Foragers game does not have a condition in which agents can be removed.
        # 'player_done_variable': 'player_done',

        'record_video_config': {
            'frequency': 20000 # number of rollouts
        },

        'random_level_on_reset': True,
        'yaml_file': 'Multi-Agent/foragers.yaml',
        'global_observer_type': gd.ObserverType.SPRITE_2D,
        'max_steps': 500,
    },
    'entropy_coeff_schedule': [
        [0, 0.01],
        [max_training_steps, 0.0]
    ],
    'lr_schedule': [
        [0, 0.0005],
        [max_training_steps, 0.0]
    ]
}

stop = {
    'timesteps_total': max_training_steps,
}

result = tune.run(PPOTrainer, config=config, stop=stop)

```


REAL TIME STRATEGY

Griddly also supports Strategy games! Strategy games in the context of Griddly are games where the player can control multiple “units” at at a single time.

RTS environments similar to multi-agent environments, but the units are controlled by individually selecting them and then performing actions. In this example, only a single action can be send to a particular unit on each turn.

19.1 Conditional Action Trees

This example uses Conditional Action Trees in order to handle multiple units and invalid action masking. Conditional Action Trees are implemented in RLLib in the `ConditionalActionImpalaTrainer`.

See also:

For more information on Conditional Action Trees, see the github repository: <https://github.com/Bam4d/conditional-action-trees> and the paper: <https://arxiv.org/abs/2104.07294>.

19.2 Full Example

In this example, self-play is used to train the agent controlling both armies. We are using a single IMPALA-CNN policy to teach both agents. This means the policy must learn to play from both perspectives (starting from the bottom of the map or the top).

```
import os
import sys

import ray
from ray import tune
from ray.rllib.models import ModelCatalog
from ray.tune.registry import register_env
from rts.models import ImpalaCNNAgent

from griddly import gd
from griddly.util.rllib.callbacks import MultiCallback, ActionTrackerCallback
from griddly.util.rllib.environment.core import RLLibMultiAgentWrapper, RLLibEnv
from griddly.util.rllib.torch.conditional_actions.conditional_action_policy_trainer_
    import \
        ConditionalActionImpalaTrainer

if __name__ == '__main__':
```

(continues on next page)

(continued from previous page)

```

sep = os.pathsep
os.environ['PYTHONPATH'] = sep.join(sys.path)

ray.init(num_gpus=1)

env_name = "griddly-rts-env"

def _create_env(env_config):
    env = RLlibEnv(env_config)
    return RLlibMultiAgentWrapper(env, env_config)

register_env(env_name, _create_env)
ModelCatalog.register_custom_model("ImpalaCNN", ImpalaCNNAgent)

max_training_steps = 1000000000

config = {
    'framework': 'torch',
    ,
    'num_workers': 8,
    'num_envs_per_worker': 5,

    'callbacks': ActionTrackerCallback,

    'model': {
        'custom_model': 'ImpalaCNN',
        'custom_model_config': {}
    },
    'env': env_name,
    'env_config': {
        'generate_valid_action_trees': True,
        'yaml_file': 'RTS/Griddly_RTS.yaml',
        'global_observer_type': gd.ObserverType.ISOMETRIC,
        'level': 0,
        'record_actions': True,
        'max_steps': 1000,
    },
    ,
    'record_video_config': {
        'frequency': 20000, # number of rollouts
        'directory': 'videos'
    },
    ,
    'entropy_coeff_schedule': [
        [0, 0.001],
        [max_training_steps, 0.0]
    ],
    ,
    'lr_schedule': [
        [0, args.lr],
        [max_training_steps, 0.0]
    ]
}

```

(continues on next page)

(continued from previous page)

```
    ],  
  
    }  
  
    stop = {  
        "timesteps_total": max_training_steps,  
    }  
  
    result = tune.run(ConditionalActionImpalaTrainer, config=config, stop=stop)
```


GDY TUTORIAL - MAKING SOKOBAN

Note: It's recommended to follow the short *GDY Schema tutorial* which will help validate any issues with GDY files as they are being created.

This tutorial will take you through building a simple game with GDY. At the end of this tutorial you will have recreated the Sokoban game in Griddy.

The tutorial comes in three main stages which represent each of the three main configuration stages, **Enviroment**, **Actions** and **Objects**.

The top level of GDY files always look like this:

```
Version: 0.1
Enviroment: ...

Actions: ...

Objects: ...
```

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.

Afterwards the environment can be loaded by Griddy and used in other projects. In this tutorial the environment will be loaded into an OpenAI gym wrapper and can be played with using your keyboard.

→ *How to play the tutorial game*

20.1 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!

20.1.1 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:



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.

20.1.2 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!

```
- 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.

20.1.3 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:



```
- 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
```

20.1.4 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:

```
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/cspell14.png

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

```

20.2 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 performs 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!

20.2.1 Step 1 - Movement

We are building the game “Sokoban” so we will first define that our `avatar` object which we defined before can move around in empty space. To do that we can define our “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 defining 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 keyword that the behaviour only applies when the action is performed on a space that has no objects present.

Finally we have `Commands` object in the `Src` key. The `Commands` object contains a list of instructions that will happen to the `Src` object if this action is executed. 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 go [here](#)

20.2.2 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 command tells avatar it can move in the same way it would just moving into empty space. The box object however now 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 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 the previous action also, stopping the avatar from moving also.

20.2.3 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
  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.

20.2.4 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

```

20.3 Environment

The Environment object contains the definition of how the game will be controlled by the player, the conditions under which the game ends, and the levels that are contained in the game.

20.3.1 Step 1 - Player Configuration

The Player configuration

```
Player:
  AvatarObject: avatar
```

20.3.2 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 separately for Win and Lose. In Sokoban, we just have a win condition where the player wins if all of the boxes have fallen into the 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` option can be appended to any object name to return the number of any of the objects.

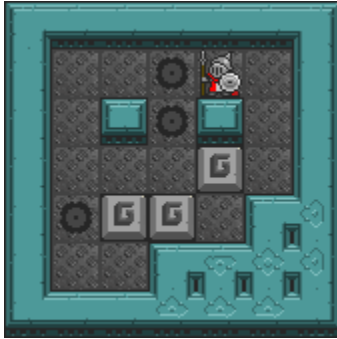
20.3.3 Step 3 - Levels

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 the MapCharacter characters that were defined in the *Objects* section of this tutorial.

the dot `.` character means that the space in the map is unoccupied.

```
Levels:
- |
  wwwwww
  w..hA.w
  w.whw.w
  w...b.w
  whbb.ww
  w..www
  wwwwww
- |
  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:



20.3.4 Step 4 - Name, TileSize and Background Image

– code-block:: YAML

Name: sokoban TileSize: 24 BackgroundTile: images/gvgai/newset/floor2.png

Here we specify the size of the tiles in pixels `TileSize`. Also if we want to use an image for the background when there are no objects present we can supply a `BackgroundTile` image. The `BackgroundTile` is ignored any Observers except from `Sprite2D`.

Additionally every environment must have a unique `Name` to differentiate it from other environments.

20.3.5 Putting it all together

The environment definition with all the parts described looks like this:

```
Environment:
  Name: sokoban
  TileSize: 24
  BackgroundTile: images/gvgai/newset/floor2.png
  Player:
    AvatarObject: avatar
  Termination:
    Win:
      - eq: [box:count, 0] # If there are no boxes left
  Levels:
    - |
```

(continues on next page)

(continued from previous page)

```
wwwwwww
w..hA.w
w.whw.w
w...b.w
whbb.ww
w..www
wwwwwww
- |
wwwwwww
ww.h...w
ww...bA.w
w...w..w
wwwbw...w
www...w.w
www.h...w
wwwwwww
```

20.4 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.

20.4.1 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's 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
```

20.4.2 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()

wrapper.build_gym_from_yaml('SokobanTutorial', 'sokoban.yaml', level=0)
```

20.4.3 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)
```


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

21.1 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"
  }
}
```

21.2 PyCharm

PyCharm has a feature for YAML validation with JSON schema files built in.

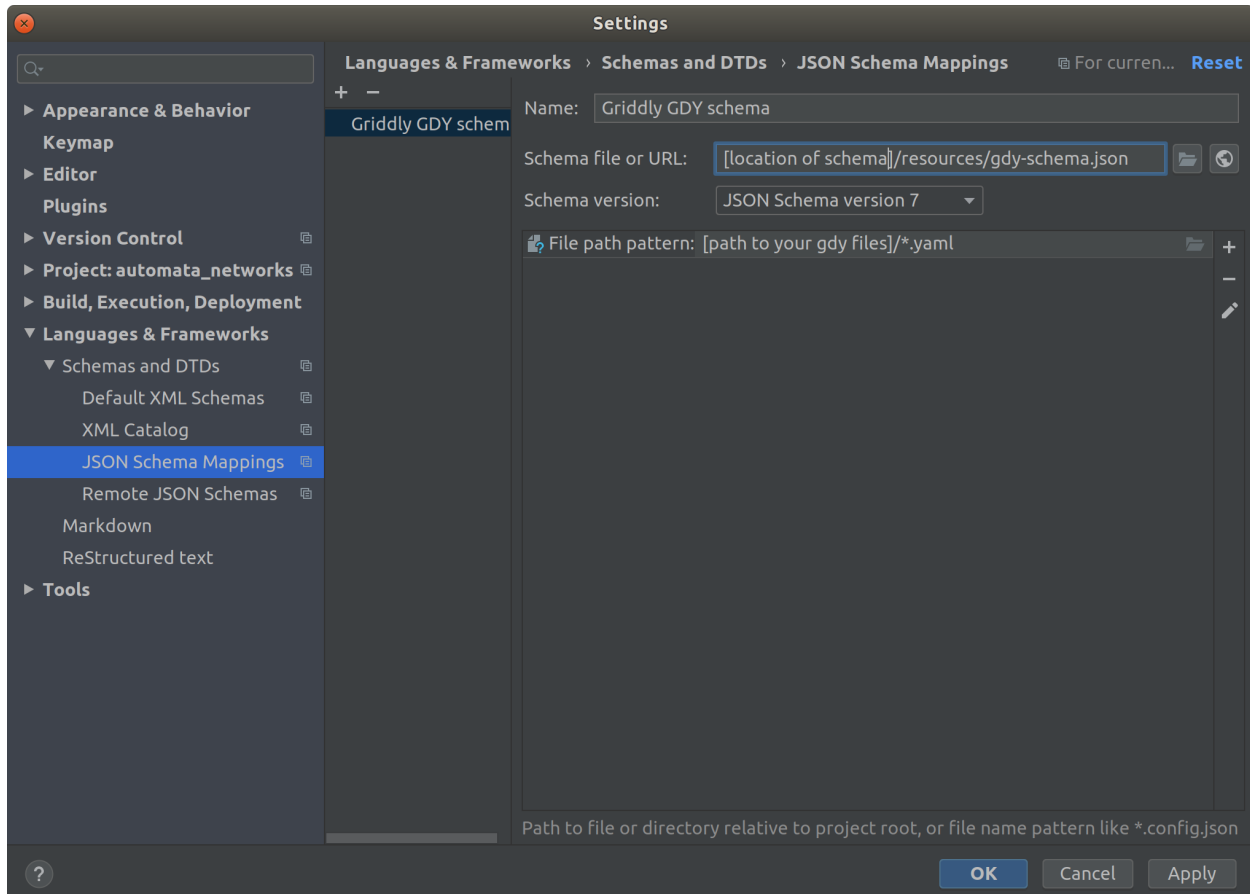
In settings navigate to Languages And Frameworks -> Schemas and DTDs -> JSON Schema Mappings

You will need 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.



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.

22.1 Step 1 - Create the lava, water and spider 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
```

(continues on next page)

(continued from previous page)

```
    Offset: [0, 4]
Block2D:
  - Color: [ 0.0, 0.0, 0.8 ]
    Shape: square
```

22.2 Step 3 - 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 ]
```

22.3 Step 4 - 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 ]
```

22.4 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 . . . . . . . . . . . . . . 1 1
W W . . . . . . . . . . . . . . 1 1
. . . . . . . . . . . . . . 1 1 1 1
. . . . . . . . . . . . . . 1 1 1 1

```

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

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](#)

23.1 Environment

Description Contains the definition of how the player interacts with the environment and how it is perceived by the players

Data Type	YAML Key
object	Environment

Properties

Property	Required
<i>Name</i>	true
<i>Description</i>	
<i>Observers</i>	
<i>Player</i>	
<i>Variables</i>	
<i>Termination</i>	
<i>Levels</i>	

23.1.1 Name

Description The name of the environment

Data Type	YAML Key
string	Name

23.1.2 Description

Description A description of the environment.

Data Type	YAML Key
string	Description

23.1.3 Observers

Description Default properties for observers

Data Type	YAML Key
object	Observers

Properties

Property	Required
<i>Vector</i>	
<i>Sprite2D</i>	
<i>Block2D</i>	
<i>Isometric</i>	

Vector

Description Configuration options for the vector renderer.

Data Type	YAML Key
object	Vector

Properties

Property	Required
<i>IncludePlayerId</i>	
<i>IncludeRotation</i>	
<i>IncludeVariables</i>	

Player Id

Description Includes the player id in the vector representation of each tile.

Data Type	YAML Key
boolean	IncludePlayerId

Rotation

Description Includes the rotation of the object in the vector representation of each tile.

Data Type	YAML Key
boolean	IncludeRotation

Variables

Description Includes the value of variables in vector representation of each tile.

Data Type	YAML Key
boolean	IncludeVariables

Sprite 2D

Description Configuration options for the sprite 2D renderer.

Data Type	YAML Key
object	Sprite2D

Properties

Property	Required
<i>TileSize</i>	
<i>BackgroundTile</i>	

Tile Size

Description Tile size to use for sprites.

Possible Values

Value	Type	Description
[integer]	integer	A single value for both the height and width of the tiles
Rectangular Tile	array	Width and height of the tiles

YAML Key
TileSize

Background Tile

Description The image that should be used as a background tile

Data Type	YAML Key
string	BackgroundTile

Block 2D

Description Configuration options for the block 2D renderer.

Data Type	YAML Key
object	Block2D

Properties

Property	Required
<i>TileSize</i>	

Tile Size

Description Tile size to use for blocks.

Possible Values

Value	Type	Description
[integer]	integer	A single value for both the height and width of the tiles
Rectangular Tile	array	Width and height of the tiles

YAML Key
TileSize

Observers

Description Configuration options for the isometric renderer.

Data Type	YAML Key
object	Isometric

Properties

Property	Required
<i>TileSize</i>	
<i>BackgroundTile</i>	
<i>IsoTileHeight</i>	
<i>IsoTileDepth</i>	

Tile Size

Description Configuration options for sprite 2D renderer.

Possible Values

Value	Type	Description
[integer]	integer	A single value for both the height and width of the tiles
Rectangular Tile	array	Width and height of the tiles

YAML Key
TileSize

Background Tile

Description The image that should be used as a background tile

Data Type	YAML Key
string	BackgroundTile

IsoTileHeight

Description The height of a floor tile in pixels of the isometric tile. This is measure from the bottom of the tile

Data Type	YAML Key
integer	IsoTileHeight

IsoTileDepth

Description The number of pixels used to cover the depth effect of isometric tiles

Data Type	YAML Key
integer	IsoTileDepth

23.1.4 Player

Description Defines how players (algorithms, agents, humans) interact with the environment.

Data Type	YAML Key
object	Player

Properties

Property	Required
<i>Count</i>	
<i>AvatarObject</i>	
<i>Observer</i>	

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
<i>RotateWithAvatar</i>	
<i>TrackAvatar</i>	
<i>Height</i>	
<i>Width</i>	
<i>OffsetX</i>	
<i>OffsetY</i>	

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

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

23.1.5 Global Variables

Description Definition global variables that can be access from any action.

Data Type	YAML Key
array	Variables

Array Type

Type	Description
<i>Variable</i>	Variable

Variable

Description Define a global variable such as number of items collected, actions performed etc.

Data Type
object

Properties

Property	Required
<i>Name</i>	true
<i>InitialValue</i>	
<i>PerPlayer</i>	

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

23.1.6 Termination

Description Definition of the termination conditions of the environment.

Data Type	YAML Key
object	Termination

Properties

Property	Required
<i>Lose</i>	
<i>Win</i>	
<i>End</i>	

Lose Conditions

Description If any of these conditions are met, the player associated with this condition will lose the game.

Possible Values

Value	Type	Description
V1 Termination Conditions	array	V1 Termination Conditions
V2 Termination Conditions	array	V2 Termination Conditions

YAML Key
Lose

Win Conditions

Description If any of these conditions are met, the player associated with this condition will win the game.

Possible Values

Value	Type	Description
V1 Termination Conditions	array	V1 Termination Conditions
V2 Termination Conditions	array	V2 Termination Conditions

YAML Key
Win

End Conditions

Description If any of these conditions are met, the game will end.

Possible Values

Value	Type	Description
V1 Termination Conditions	array	V1 Termination Conditions
V2 Termination Conditions	array	V2 Termination Conditions

YAML Key
End

23.1.7 Game Level Maps

Description Level Strings which define the levels in the game environment.

Data Type	YAML Key
array	Levels

23.2 Actions

Description Actions define all the game mechanics.

Data Type	YAML Key
array	Actions

Array Type

Type	Description
<i>Action Definition</i>	Action Definition

23.2.1 Action Definition

Description A single action.

Data Type
object

Properties

Property	Required
<i>Probability</i>	
<i>Trigger</i>	
<i>InputMapping</i>	
<i>Name</i>	true
<i>Behaviours</i>	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
<i>Type</i>	
<i>Range</i>	

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

Input Mapping

Description Map action Ids to in-environment actions

Data Type	YAML Key
object	InputMapping

Properties

Property	Required
<i>Inputs</i>	
<i>Relative</i>	
<i>Internal</i>	
<i>MapToGrid</i>	

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
<i>Behaviour Definition</i>	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
<i>Src</i>	true
<i>Dst</i>	true

Source

Description Define the behaviour of the source object of an action.

Data Type	YAML Key
object	Src

Properties

Property	Required
<i>Object</i>	true
<i>Commands</i>	
<i>Preconditions</i>	

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
<i>Behaviour Command</i>	A command to run as part of an action behaviour.
<i>Conditional Behaviour Command</i>	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
<i>mov</i>	
<i>rot</i>	
<i>add</i>	
<i>sub</i>	
<i>set</i>	
<i>decr</i>	
<i>incr</i>	
<i>cascade</i>	
<i>reward</i>	
<i>remove</i>	
<i>change_to</i>	
<i>set_tile</i>	
<i>spawn</i>	
<i>exec</i>	

Move

Description Move the object to the location provided.

YAML Key
mov

Choose Between

Type	Description
<i>Destination</i>	The destination location of the action.
<i>Source</i>	The source location of the action.
<i>Command Arguments</i>	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
<i>Command Argument</i>	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
<i>Direction</i>	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
<i>Command Argument</i>	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
<i>Command Argument</i>	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
<i>Command Argument</i>	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
<code>cascade</code>

Choose Between

Type	Description
<i>Destination</i>	The destination location of the action.
<i>Source</i>	The source location of the action.
<i>Command Arguments</i>	Arguments supplied to the command, 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	<code>reward</code>

Remove

Description Remove the object from the environment.

Data Type	YAML Key
boolean	<code>remove</code>

Change Object

Description Swaps this object for another one. (this does not preserve any properties of the current object)

Data Type	YAML Key
string	<code>change_to</code>

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	<code>set_tile</code>

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
<i>Action</i>	true
<i>Randomize</i>	
<i>Delay</i>	
<i>ActionId</i>	

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

Conditional Behaviour Command

Description A conditional command to run as part of an action behaviour.

Properties

Property	Required
<i>eq</i>	
<i>lt</i>	
<i>lte</i>	
<i>gt</i>	
<i>gte</i>	

Equals

Description The specified commands will only be run if the arguments are equal.

Data Type	YAML Key
object	eq

Properties

Property	Required
<i>Arguments</i>	
<i>Commands</i>	

Command Options

Description Commands

Data Type
array

Array Type

Type	Description
<i>Behaviour Command</i>	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
<i>Arguments</i>	
<i>Commands</i>	

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
<i>Arguments</i>	
<i>Commands</i>	

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
<i>Arguments</i>	
<i>Commands</i>	

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
<i>Arguments</i>	
<i>Commands</i>	

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
<i>Behaviour Precondition</i>	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
<i>eq</i>	
<i>neq</i>	
<i>gt</i>	
<i>gte</i>	
<i>lt</i>	
<i>lte</i>	

Equals

Description Check if the arguments are equal

Data Type	YAML Key	Max Items	Min Items
array	eq	2	2

Array Type

Type	Description
<i>Command Argument</i>	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
<i>Command Argument</i>	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
<i>Command Argument</i>	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
<i>Command Argument</i>	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
<i>Command Argument</i>	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
<i>Command Argument</i>	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
<i>Object</i>	true
<i>Commands</i>	

23.3 Objects

Description An explanation about the purpose of this instance.

Data Type	YAML Key
array	Objects

Array Type

Type	Description
<i>Object Definition</i>	Object Definition

23.3.1 Object Definition

Description An explanation about the purpose of this instance.

Data Type
object

Properties

Property	Required
<i>Z</i>	
<i>InitialActions</i>	
<i>Name</i>	true
<i>MapCharacter</i>	
<i>Variables</i>	
<i>Observers</i>	

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
<i>Initial Action</i>	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
<i>Action</i>	true
<i>Randomize</i>	
<i>Delay</i>	
<i>ActionId</i>	

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
<i>Variable</i>	Variable

Variable

Description Define an object variable, such as health, items collected etc...

Data Type
object

Properties

Property	Required
<i>Name</i>	true
<i>InitialValue</i>	

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

Properties

Property	Required
<i>Sprite2D</i>	
<i>Block2D</i>	
<i>Isometric</i>	

Sprite2D Observer

Description The Sprite2D render uses Vulkan to render the images to make the environment look super nice.

Data Type	YAML Key
array	Sprite2D

Properties

Property	Required
<i>TilingMode</i>	
<i>Image</i>	true

Tiling Mode

Description Tiling mode of the image.

YAML Key	Allowed Values	Default Value
TilingMode	NONE, WALL_2, WALL_16	NONE

Sprite Image

Description Either a list of image tiles or a single image

Possible Values

Value	Type	Description
[string]	string	A single image to use for ever occurence of this object.
Image Tiles	array	A list of image tiles to combine for animations or walls.

YAML Key
Image

Block2D Observer

Description The Sprite2D render uses Vulkan to render simple shapes.

Data Type	YAML Key
array	Block2D

Properties

Property	Required
<i>Shape</i>	true
<i>Color</i>	true
<i>Scale</i>	

Shape

Description The shape to use for this particular object

Data Type	YAML Key
string	Shape

Color

Description The color of the shape

Data Type	YAML Key	Max Items	Min Items
array	Color	3	3

Array Type

Type	Description
<i>Color value</i>	Color value

Color value

Description A float value for RGB

Data Type
number

Scale

Description The scale of the shape. if 1.0 the shape will fill the grid location.

Data Type	YAML Key	Min Value	Max Value	Default Value
number	Scale	0.0	1.0	1.0

Isometric Observer

Description The Isometric render uses Vulkan to render the images to make the environment look super nice.

Data Type	YAML Key
array	Isometric

Properties

Property	Required
<i>Offset</i>	
<i>TilingMode</i>	
<i>Image</i>	true

Offset

Description Rendering offset for this specific tile.

Data Type	YAML Key	Max Items	Min Items
array	Offset	2	2

Array Type

Type	Description
<i>Offset Coordinate</i>	Offset Coordinate

Offset Coordinate

Description Coordinate for rendering offset of isometric tiles

Data Type
integer

Tiling Mode

Description Tiling mode of the image.

YAML Key	Allowed Values	Default Value
TilingMode	NONE, ISO_FLOOR	NONE

Sprite Image

Description Either a list of image tiles or a single image

Possible Values

Value	Type	Description
[string]	string	A single image to use for ever occurrence of this object.
Image Tiles	array	A list of image tiles to combine for animations or walls.

YAML Key
Image