

UNIVERSITY OF WESTERN AUSTRALIA

CITS3001

AGENTS, ALGORITHMS AND ARTIFICIAL INTELLIGENCE

Super Mario Project

August 11, 2023



Figure 1: Sourced from XKCD [1]

Contents

1	Project Overview	2
1.1	Requirements	2
1.1.1	Gym-Super-Mario-Bros Environment Setup	2
1.1.2	AI Algorithm Implementations	2
1.2	Final Project Report	3
1.2.1	Analysis	3
1.2.2	Performance Metrics	3
1.2.3	Visualization and Debugging	3
1.3	Marks Distribution	3
1.4	Additional Notes	4
2	Submission Guidelines	5
3	Interviews	5
4	Environment Setup	6
4.1	Poetry	6
4.1.1	Tool Installation	6
4.1.2	Package Installation	6
4.2	Conda	6
4.2.1	Tool Installation	6
4.2.2	Environment Creation	6
4.2.3	Package Installation	6
5	Test Code	6
6	Example Issues and How to Fix Them	7
6.1	Tuple unpacking API Issue	7

1 Project Overview

In this project, you will develop AI agents to control the iconic character Mario in the classic game Super Mario Bros using the gym-super-mario-bros environment. The main objective is to implement at least two distinct AI algorithms/methods and compare their performance, strengths, and weaknesses in the context of playing the game.

You can undertake this project in teams of 2 that you select, if you are looking for a partner please reach out to your lab demonstrators by emails

1.1 Requirements

1.1.1 Gym-Super-Mario-Bros Environment Setup

- Set up the gym-super-mario-bros [2] environment on your local machine or any designated platform, see 4.2.2 Environment Creation.



Figure 2: Example gym-super-mario-bros

1.1.2 AI Algorithm Implementations

- Choose and implement at least two AI algorithms. You may wish to consider the following:
 - Reinforcement Learning: Q-learning [3], TD(λ) [4]
 - Rule-Based AI: logic and heuristics.
 - Monte Carlo Tree Search (MCTS) [5]

You are welcome to use more advanced algorithms that utilise deep learning such as DQN's [6] or Proximal Policy optimisation etc. but these are not covered in the unit and lab facilitators may not be able to assist with your implementations. These algorithms will also have to be referenced in your project report.

1.2 Final Project Report

To demonstrate your understanding of your implementations you will be required to write a final project report. Your report must conform to the following guidelines

- At least 3 pages
- No longer than 5 pages
- No smaller than size 12 font

You are allowed to add appendices with extra figures and words but these may not be marked. Your report should cover the following areas:

1.2.1 Analysis

- Analyze and contrast the performance of the chosen AI methods.
- Discuss their respective strengths, weaknesses, and suitability for playing Super Mario Bros.

1.2.2 Performance Metrics

You will notice that gym-super-mario-bros reward function assumes the objective of the game is to move as far right as possible. You are encouraged to come up with other performance and evaluation metrics for your agents. Novel and interesting metrics that you come up with will be rewarded.

1.2.3 Visualization and Debugging

- Include what visualization techniques you used to gain insights into the agent's decision-making process.
- Include what debugging/profiling tools you utilised to optimize the algorithms and enhance performance.

1.3 Marks Distribution

- AI Algorithm Implementations (40%)
 - Successful implementation of two or more AI methods (20%)
 - Code quality, readability, and efficiency (10%)
 - Integration with the gym-super-mario-bros environment (10%)
- Report Comparison and Contrast (30%)
 - In-depth analysis of the algorithms' strengths and weaknesses (15%)
 - Properly conducted experiments and results presentation (10%)
 - Effective comparison of AI methods (5%)
- Performance Metrics and Visualization (20%)
 - Selection and definition of appropriate performance metrics (10%)

- Quality of visualization techniques (10%)
- Report Quality (10%)
 - Clarity and organization of the report (5%)
 - Overall presentation and writing quality (5%)

1.4 Additional Notes

- You are encouraged to use external libraries and frameworks to support your implementation, but remember to provide proper citations and acknowledgments.
- Regularly check in with the instructor for progress assessments and guidance during the project timeline.
- Ensure that your work is original and properly referenced to maintain academic integrity throughout the project.

2 Submission Guidelines

You will be required to submit the following:

- A zip file containing the code for your agents
- Your final report as a PDF file

Please ensure your code submission contains a README.md file explaining how to setup the environment for your project and how to run the two separate agents individually

3 Interviews

If markers suspect instances of plagiarism or excessive reliance on ChatGPT or any external sources, a procedural measure has been established. Students found in such situations may be requested to attend an interview with their respective lab instructor. The purpose of this interview is to provide students with an opportunity to demonstrate their understanding of the project. It is worth noting that consistent attendance in laboratory sessions significantly mitigate the likelihood of being selected for an interview.

4 Environment Setup

You are using an official public package, so please don't spam the github issues. I have created a fork for our unit on github that you can file specific requests under

4.1 Poetry

4.1.1 Tool Installation

See poetry documentation on how to install for your system.

```
curl -sSL https://install.python-poetry.org | python3 -
```

4.1.2 Package Installation

```
poetry add gym-super-mario-bros
```

4.2 Conda

4.2.1 Tool Installation

Follow the instructions on anaconda documentation.

4.2.2 Environment Creation

The following is just an example to get you started.

```
conda create -n mario
```

```
conda activate mario
```

4.2.3 Package Installation

```
conda install gym-super-mario-bros
```

5 Test Code

```
from nes_py.wrappers import JoypadSpace
import gym_super_mario_bros
from gym_super_mario_bros.actions import SIMPLE_MOVEMENT
import gym

env = gym.make('SuperMarioBros-v0', apply_api_compatibility=True, render_mode="human")
env = JoypadSpace(env, SIMPLE_MOVEMENT)

done = True
env.reset()
for step in range(5000):
    action = env.action_space.sample()
    obs, reward, terminated, truncated, info = env.step(action)
    done = terminated or truncated

    if done:
```

```
state = env.reset()

env.close()
```

6 Example Issues and How to Fix Them

6.1 Tuple unpacking API Issue

```
DeprecationWarning: 'np.bool8' is a deprecated alias for 'np.bool_'. (Deprecated NumPy 1.24)
Traceback (most recent call last):
  File "/home/dadams/Desktop/Other/super-mario/test.py", line 11, in <module>
    state, reward, done, info = env.step(env.action_space.sample())
                                ~~~~~~
  File "site-packages/nes_py/wrappers/joypad_space.py", line 74, in step
    return self.env.step(self._action_map[action])
           ~~~~~~
  File "gym/wrappers/time_limit.py", line 50, in step
    observation, reward, terminated, truncated, info = self.env.step(action)
    ~~~~~~
ValueError: not enough values to unpack (expected 5, got 4)
```

This issue was caused by an api cmpatability error so simply put:

```
env = gym.make('SuperMarioBros-v0', apply_api_compatibility=True, render_mode="human")
```


References

- [1] Randall Munroe. Seashell. <https://xkcd.com/1236/>, 2013. [Online; accessed October 20, 2019].
- [2] Christian Kauten. Super Mario Bros for OpenAI Gym. GitHub, 2018.
- [3] Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8:279–292, 1992.
- [4] Harm Seijen and Rich Sutton. True online td (λ). In *International Conference on Machine Learning*, pages 692–700. PMLR, 2014.
- [5] Cameron B Browne, Edward Powley, Daniel Whitehouse, Simon M Lucas, Peter I Cowling, Philipp Rohlfshagen, Stephen Tavener, Diego Perez, Spyridon Samothrakis, and Simon Colton. A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in games*, 4(1):1–43, 2012.
- [6] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.