

## Final Project Proposal Super Mario Agent with Reinforcement Learning

### Problem Statement:

The goal of our project is to develop an AI agent that can play that classic video game Super Mario. We will try to pick out a specific section of the video game and try to make the agent, in this case Mario, know when to jump and move accordingly as the level progresses. It will be a challenging task to employ the specific algorithms that would co-ordinate with the dynamics of the game and its changing environment. We will have to define specifics with the environment and the states and define the actions as well.

### References:

S. Padhye, "Playing super mario bros. with reinforcement learning," Medium, <https://sohumpadhye.medium.com/playing-super-mario-bros-with-reinforcement-learning-81ee0c235372> (accessed Feb. 10, 2024).

"Train a mario-playing RL agent¶," Train a Mario-playing RL Agent - PyTorch Tutorials 2.2.0+cu121 documentation, [https://pytorch.org/tutorials/intermediate/mario\\_rl\\_tutorial.html](https://pytorch.org/tutorials/intermediate/mario_rl_tutorial.html) (accessed Feb. 10, 2024).

"Build an mario AI model with Python | Gaming Reinforcement Learning," YouTube, <http://www.youtube.com/watch?v=2eeYqJOuBKE> (accessed Feb. 10, 2024).

### Proposed Approach:

We will be using reinforcement learning to approach towards a solution for this problem. Our plan is to execute the algorithms that specifically decide what to do when the environment and the state of the problem keeps changing considering this is a changing dynamic environment. As far as I have researched, we will be using an approach called Deep Q Learning and the plan to use it is still tentative until we gather more information.

We will be using a game screen, position of the agent, the enemy's position and power ups as the game state or states and observations used. The actions available would be to jump, go left and go right and other ones as we see fit. We still are deciding on using a reward function or not based on the complications we will be seeing.

So far, we plan on using these libraries, credit goes to jiseongHAN on gitlab,  
numpy  
torch>=1.6.0  
torchvision  
gym==0.23

nes-py  
gym-super-mario-bros==7.2.3  
opencv-python  
matplotlib

### **Work Breakdown (Tentative):**

#### **March 1<sup>st</sup>, 2024:**

Have MarioNet Source code is installed on both the machines and is running properly.  
(Cedric and Aryan).

#### **March 1<sup>st</sup>, 2024 to March 15<sup>th</sup>, 2024:**

Start with the reinforcement learning track and note the progress down. (Aryan)

Start with Deep Q Learning and note the progress down. (Cedric)

Start with Prediction and Regression and note the progress down. (Aryan)

Get information about progress and complications. Note down what we need help with and what we need advice on. (Aryan and Cedric)

#### **March 16<sup>th</sup>, 2024 to April 5<sup>th</sup>, 2024:**

Finalize work on the above topics and start debugging and noting errors and what we still can't figure out. (Aryan and Cedric)

This schedule is tentative and we need more information and planning on what we need to know to figure a better schedule out and have shorter sprints for more tasks.

Document progress and start preparations for Presentation and Demonstration of the project.  
(Aryan and Cedric).

#### **April 6<sup>th</sup>, 2024 to April 19<sup>th</sup>, 2024:**

Finalize and start testing for Regression, Prediction, Reinforcement Learning and Deep Q Learning.  
(Aryan and Cedric)

Work on Project Report and Presentation Slides.  
(Aryan and Cedric)

### **References:**

- [1] "ArXiv.org e-print archive," arXiv.org e-Print archive, <https://arxiv.org/> (accessed Feb. 10, 2024).
- [2] "Part 1: Key concepts in RL¶," Part 1: Key Concepts in RL - Spinning Up documentation, [https://spinningup.openai.com/en/latest/spinningup/rl\\_intro.html](https://spinningup.openai.com/en/latest/spinningup/rl_intro.html) (accessed Feb. 10, 2024).
- [3] yfeng997, "YFENG997/madmario: Interactive tutorial to build a learning mario, for first-time RL learners," GitHub, <https://github.com/yfeng997/MadMario> (accessed Feb. 10, 2024).
- [4] jiseongHAN, "Jiseonghan/Super-Mario-RL: ¶reinforcement learning: Super mario bros with dueling DQN¶," GitHub, <https://github.com/jiseongHAN/Super-Mario-RL> (accessed Feb. 10, 2024).

- [5] "Build an mario AI model with Python | Gaming Reinforcement Learning," YouTube, <http://www.youtube.com/watch?v=2eeYqJ0uBKE> (accessed Feb. 10, 2024).
- [6] "Train a mario-playing RL agent¶," Train a Mario-playing RL Agent - PyTorch Tutorials 2.2.0+cu121 documentation, [https://pytorch.org/tutorials/intermediate/mario\\_rl\\_tutorial.html](https://pytorch.org/tutorials/intermediate/mario_rl_tutorial.html) (accessed Feb. 10, 2024).
- [7] S. Padhye, "Playing super mario bros. with reinforcement learning," Medium, <https://sohum-padhye.medium.com/playing-super-mario-bros-with-reinforcement-learning-81ee0c235372> (accessed Feb. 10, 2024).