| | | |
|---|---|---|
| **Name**: Parashara Ramesh | **NUSNET ID**: E1216292 | **Student ID**: A0285647M |
| **Name**: Sriram Srikanth | **NUSNET ID:** E1132261 | **Student ID**: A0276528R |
| **Name**: Grace Ngu Sook Ern | **NUSNET ID:** E1101580 | **Student ID**: A0268281X |
| **Name**: Xian He | **NUSNET ID:** E1101724 | **Student ID**: A0268425X |

## 1. Background

As a team we plan to implement some of the most popular Reinforcement Learning algorithms on the classic "Super Mario Bros" video game (which is a single agent game).
Our focus would mainly be to compare and contrast the performances of each algorithm while extracting insights about its merits / demerits from first principles.
As a baseline, we plan to select some of the levels and stages as we attempt to play them manually to understand how well an average human with little practice can achieve in this game and then move onwards to the reinforcement learning algorithms.

## 2. Proposal

Our plan is to leverage this game environment implementation from github and attempt to implement the following algorithms each with its own minor "architectural" variants after taking inspiration from available open source implementations. Here are the algorithms we hope to implement:
a. Proximal Policy Optimization; b. MuZero; c. Deep Q Networks; d. Actor Critic
Given enough time, we also plan to conduct a tournament of each agent where we individually time each agent (along with human agents i.e. this project team) to determine the best performing agents for particular levels.
For each algorithm and we plan to train some variants separately which are namely:

a. Training on a single level and testing its performance across other unseen levels
b. Training on all levels and testing its generic performance
c. Training without any human generated data (if possible)
d. Training variants with minor architectural differences/reward motivations (e.g. fastest time or completion %)

With this approach, we hope to comprehensively evaluate the performance of each agent holistically.

## 3. Timelines / Split of work

Since we have started off with 4 algorithms in mind, our plan is for each team member to separately train each algorithm (and its variants) and then evaluate their performances at the end.

| **Task** | **Timeline** | **Work Distribution** |
|---|---|---|
| Clone the repository and enable manual playing to understand each environment levels | 10th March | All members |
| Create a random agent and watch it fail spectacularly and learn insights from it | 10th March | All members |
| Baseline performance collection: Capture human data from our manual playing of selected levels or obtain from online resources | 17th March | 2 members: Manual Playing<br>2 members: Literature Review |
| Research about evaluation metrics and implement those functions | 17th March | All members |
| Start working on each algorithm and its variants parallely | 31st March | One algorithm each |
| Evaluate each agent and capture metrics | 31st March | All members |
| Have a tournament to decide the winner | 13th April | 2 members: Run tournament for the trained algorithms |
| Report Preparation & Submission | 13th April | 2 members: Write up |