

1 Related work

Tampuu, Matiisen et al[3] explored multi-agent environments and observed how the agents performed based on the reward system. They performed the experiments on Pong game. The most interesting experiment which is relevant to our project is simulating a collaborative and competitive environment and observe how agents interact. In the work done by Tampuu, Matiisen et al[3], the player was rewarded when the opponent lost in a competitive environment, whereas in a collaborative environment both players were penalized when any one of them lost. They clearly observed different agent behavior for these cases.

We would like to experiment this on a multi-agent snake game where the snakes can be rewarded not only on the amount of food consumed but how many snakes are alive at any given moment. In a collaborative environment the snakes should avoid killing other snakes and avoid death to consume the available food. However, in a competitive environment a snake should focus on how it is able to stay alive and consume more food.

2 JavaScript Pygame

We are using OpenAI gym[2] as an environment to train the agent. OpenAI gym is an environment which provides some APIs to interact with the game environment. It is a toolkit for developing and comparing reinforcement learning algorithms and provides some pre-built game environments. For these games we can focus primarily on the algorithm and not worry about how the game works. Interactions with the game include initializing a game, ending or resetting the ongoing game and stepping through the game. The step function takes one action from an allowed set of actions called “action space”. On completion of the step function, an “observation space” is returned. The observation space contains a game state and varies from game to game. It can be as simple as the current game snapshot or any complex data like coordinates and speed of each entity in the game. It also contains a reward for the current action and a flag which states whether the game is over or not.

Our first requirement was to build a snake game environment aligned with OpenAI gym principles. One of the snake game implementations which was similar to what we required was implemented by Loonride[1]. We modified the game and added a layer on top of it to create a gym environment. The game was built in JavaScript using Phaser game engine. For the ease of use and make minimal changes we simulated the browser experience in a headless mode using selenium web driver. This simulation encapsulated the game and provided gym environment APIs.

We modified the game to expose game information like current position of the snakes, scores, whether the snakes is alive or not, food positions and the cur-

rent game snapshot. This information is used by the agent algorithm to learn the game play. Furthermore, the input to the snake bot was also changed to take inputs from the action space instead of input devices such as mouse or keyboard. The actions tell the snake where to go. We divided the space into 36 segments of 10 degrees each to simplify the action space and have a finite set of directions. Based on the action provided the snake moved in that direction.

We trained the agent based on this gym environment and perceived it to be slow. Therefore, we moved to a simpler game environment, although this was a more realistic environment.

References

- [1] <https://github.com/Loonride/slither.io-clone>
- [2] <https://github.com/openai/gym>
- [3] Ardi Tampuu, Tambet Matiisen, Dorian Kodelja, Ilya Kuzovkin, Kristjan Korjus, Juhan Aru, Jaan Aru, Raul Vicente, “Multiagent Cooperation and Competition with Deep Reinforcement Learning”.