|  |
| --- |
| The University of Sheffield |
| **Artificial Intelligence Agents for Games** |
| - Design a Game of your choice |

|  |
| --- |
|  |

**Name: Junjin Chen**

**Supervisor: Eleni Vasilaki**

**Module Code: EV-UG-4**

**Date: 25/09/2017**

Google DeepMind has done many research on Artificial intelligence and leading the world in this field. It has developed programs that can learn to play classic Atari 2600 games from high-dimensional sensory inputs (e.g. pixels in the screen and score, same as human player would experience). The algorithm behind it uses the deep reinforcement learning method that makes a system learn from rewards or punishments. After each training, data will be processed in the neural network and it will make some improvement to the system. That is basically how the agent learns. As in practical, Demis Hassabis, the co-founder and CEO of DeepMind, said that he was surprised that after the algorithm played the Space Invaders about 8 hrs, it was able to make a predict shot to kill the last alien, and after it played the Breakout 500 times, it discovered the optimal strategy which is to dig a tunnel round the left-hand and then send the ball round the back. Another famous example of this algorithm is Alpha Go, who beat the human professional Go player with 9 dan (highest rank in Go) Lee Sedol with the score of 4-1. AlphaGo was initially given 100,000 games to mimic human player, after that it was ordered to play the games itself 30,000,000 times, which made the Alpha Go stronger than human. Those examples show that deep reinforcement learning algorithm that DeepMind has been using is pretty strong. Demis also emphasised that the machine learns automatically from raw inputs, i.e. without pre-programmed, and thus that the same system can operate across a wide range of tasks. This reflects the ‘deep’ in ‘deep reinforcement learning’.

In this project, I will use the deep reinforcement learning algorithm to learn to play the game of my choice, and compare this algorithm with other different algorithms. The game I have decided is called StarCraft2. It is a real-time strategy video game developed by Blizzard Entertainment. There are many different units with different abilities and other elements in the game, these elements make the game colourful, complex and has many strategic possibilities. Here are the reasons that I chose the game: Firstly, it is one of my favourite game, I used to play a lot of this games and I have plenty knowledge of this game including its culture. Secondly, this game is really fun to play due to its complexity, unlike Go or any Atari 2600 game which has much simpler control. Hence Starcraft2 is more challenging for the algorithm to learn. And Finally, DeepMind has the source of Starcraft2 learning environment called PySC2, and it is open for everyone, which is really a good opportunity for me doing this project. On the other side, there are already some AI in this game, but those AI are not strong enough comparing to the master human player and they do not learn. Therefore, after I finish my code, I will compare it with the existing AI in order to see the difference between these algorithms and my algorithm. In summary, the problem is to apply the deep reinforcement learning to play the Starcraft2 and compare with the other algorithms in the game. The technique is the reinforcement learning algorithm, and the tool is the Starcraft2 learning environment PySC2.

Here is a more detailed plan: In the beginning, I will be looking through the exist AI in Starcraft2 and test it, to see how it performs and also to get familiar with the PySC2. This could possibly take 1-2 weeks. Next, I will look through some materials related to the deep reinforcement algorithm, and then I will design the model of my algorithm. This is expected to take 2 weeks. After that I will implement my design and program the deep reinforcement learning algorithm in the PySC2. Then I will run the program and start training it. Some data will be recorded from this experiment, in order to comparing with other algorithms and see how far it goes. This whole process including coding and training could take 5-6 weeks. Finally, I will evaluate the rank of my algorithm, either by finding a volunteer who is master at this game and invite him/her to test the algorithm, or by making it play with itself and I will analyse the game. This could take 2 weeks. In every week, I will do some research about the theory behind the reinforcement learning and other related topic (e.g. neural network). And after implementing my plan, I will analyse and summarise what I have done in that week. I estimate that I will spend 16 hours every week in this project (including weekend).