

Detailed Proposal

Final Year Project – Reinforcement Learning

Student Name: Yuhang Song
Student ID: 201403970

Supervisor: Dr. Wojtczak, Dominik

| Project Description

This project involves approach to implement a reinforcement learning algorithm together with some convolutional neural network techniques, to train and enable an artificial agent to play one of the Atari games^[0], named 'DemonAttack'.

| Aims & Objectives

- **Aim:** To implement a reinforcement learning algorithm, which would allow the intelligent agent to action on game DemonAttack with iterative better policies.

Objective: Implement a Deep Q-Learning Network algorithm that takes game information of current state as its inputs, such as rewards, whether the game is done, etc., and processing it to eventually output a reasonable action that try to get better rewards.

- **Aim:** To implement a convolutional neural network that takes inputs (screen shot) from the game simulator, and then extract information regarding to current game state of it.

Objective: Implement a suitable Convolutional Neural Network by using third-party frames that correctly extract state information of current, includes the location of the agent & enemy, score shown on screen, remaining attempt time, enemies' bullets. Furthermore, pass this information to the DQN algorithm for further processing.

| Key Literature & Background Reading

According to my research, it is possible to use Reinforcement Learning together with Convolutional Neural Network to train an agent to play Atari games.

As we all know, convolutional neural networks are very suitable for image processing, it could be used to do tasks such as object recognizing/classification and edge detection, the earlier one would help our intelligent agent to locate itself and the enemies, the later one however, would also be useful for the agent to extract key literal information from the game such as the score, remaining attempt time, etc.

Deep Q-Learning Network, which is a combination of both Deep Q-Learning algorithm and Neural Networks, it is an algorithm that obtain the so call 'best' strategy of a state based on estimating values under each choose, Deep Q-Learning algorithm is then build a Neural Network that contains Q values so that those values would be able to update themselves through training.

Also, there are good artificial intelligence programming frame available there for us to simulate the game, like 'gym'^[1] library in Python.

| Development & Implementation Summary

This project will be implemented by Python because python itself has plenty of well-known third-party libraries that are convenient for mathematical calculations and common-used by the artificial intelligent community.

There are in total two key parts of the project, Convolutional Neural Network for image observations extrication, and a Deep Q-Learning Network for processing and decision making.

The first part will be implemented by using Deep-Learning and Neural Network frames (or functions) from TensorFlow^[2] and Keras^[3]. TensorFlow is an well-known artificial intelligent developing frame especially in neural networks and deep learning development, whereas, the Keras in other hand, is a deep learning library based on TensorFlow and Theano (A deep learning library from Université de Montréal), which packages the TensorFlow frame once more for quick implementation. Both TensorFlow and Keras supports GPU accelerate, which means it is suitable for the project, since we want to tuning the parameters to get better performance of the agent, that means a significant times of training will be performed.

Competitively, another available choice would be Pytorch, which is also an artificial intelligent library, however, it is mainly known for NLP(Natural Language Processing) use. Furthermore, the syntax of TensorFlow, to me, is easier to understand and use, that is also the reason why I choice to use TensorFlow&Keras rather than Pytorch.

Second part will involve some self-attempt of the DQN algorithm, however, some of the third-party libraries are also considered to be used to carry out mathematical calculations or establish neural networks.

Lastly, the game will be simulated by the gym library of python. The development workflow will be carried out by the classical Water Fall model.

| Data Source

A copy of game`s ROM is required for this project, the source data is obtained from <https://wowroms.com/en/roms/atari-2600/demon-attack-usa/30597.html> which they have permissions for internet users on non-commercial usage. This is used specifically to better try out the rules of the game.

For the training environment, the 'gym' library simulates the game itself, with its internal game ROM, using MIT license.

| Testing & Evaluation

Certainly, testing would be carried out in later development period of this project, on the following aspect:

1. Pressure testing (How many or how complex the takes could be for the agent to carries out)

2. Ability testing (If the agent correctly played the game)
3. Reliability testing (Whether the agent actually learning from the previous knowledge, that refers to if the award gained gets higher over a certain time period)

The above tests would eventually be resulting to a sense of evaluating the agent (algorithm).

| Ethical Considerations

Data Source

Since this project will not be considered using any data from the public domain, there is no ethical considerations related to data source.

Testing Software with Human Participants

The testing and evaluation part of project would only be sufficient for may be inviting few friends to play the game simultaneously with the agent, and thus compare scores won, there is therefore no need for a larger pool of people.

Protecting Participant Data

None of the personal data would be collected by the algorithm or the intelligent agent, so there is no need for considering this part of ethical requirements.

| BCS Project Criteria

In this section, I would explain below what the BCS criteria are and how are they satisfied by my project:

- An ability to apply practical and analytical skills gained during the degree programme.
The COMP338 – Computer Vision, COMP305-Biocomputation, ELEC319-Image Processing, and some other programmes I studied last year gains knowledge of general approaches and interpretation of artificial intelligence.
- Innovation and/or creativity.
Although some of the university modules does include ideas and methods of artificial intelligence, however there is not a module that studies reinforcement learning, therefore this project is a new attempt to me.
- Synthesis of information, ideas and practices to provide a quality solution together with an evaluation of that solution.

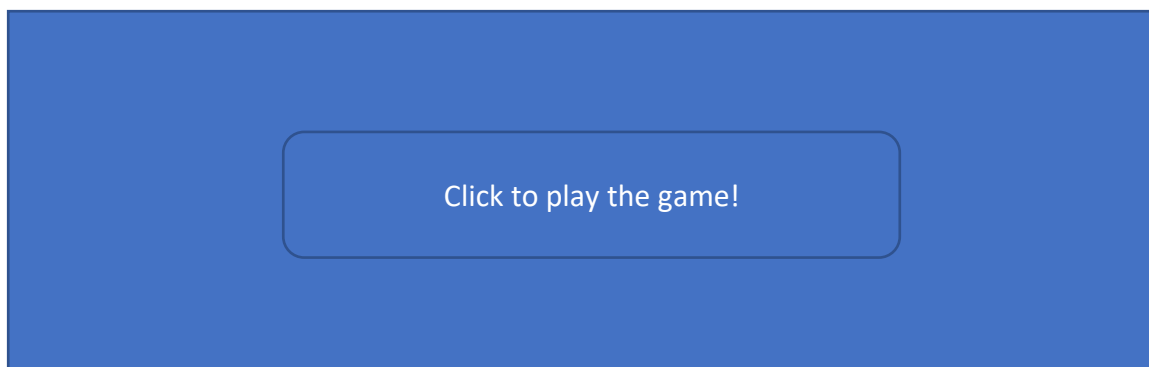
This part refers to the Development & Implementation Summary and Background Reading section above, which includes my idea of combining techniques to achieve the project requirements.

- That your project meets a real need in a wider context.
An intelligent agent that can performs high-level analyzable information and choose reasonable action is needed nowadays in many fields that requires autonomous systems.
- An ability to self-manage a significant piece of work.
The source code would be stored and maintained in both my local PC and github.
- Critical self-evaluation of the process.
Please refer to Testing & Evaluation section above for this part.

| UI/UX Mockup

User Interface (UI):

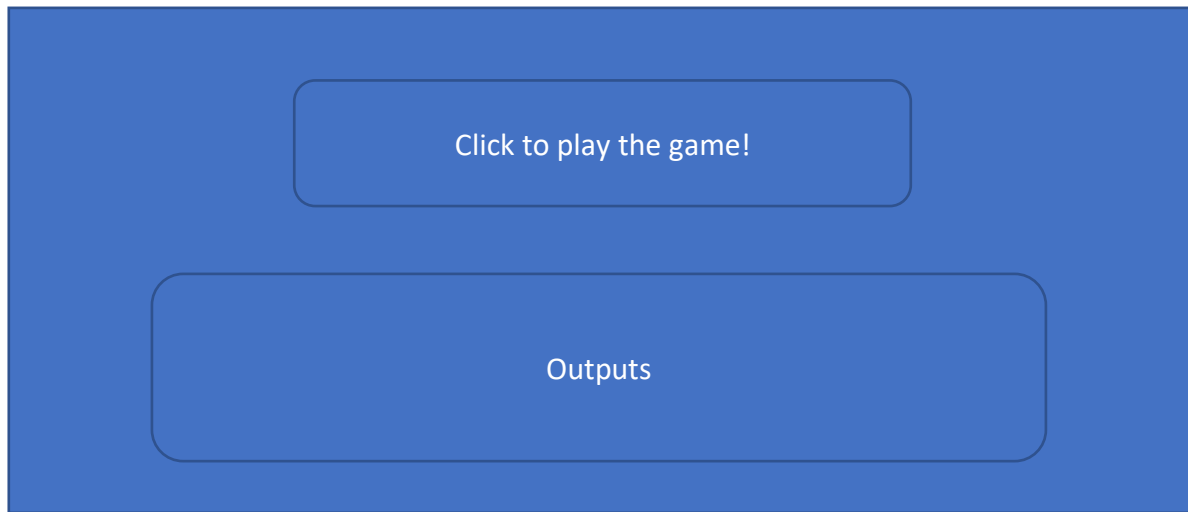
This project is more of an algorithm-based project; therefore, no further user interface or user experience design would be considered apart from a single big button design shown below:



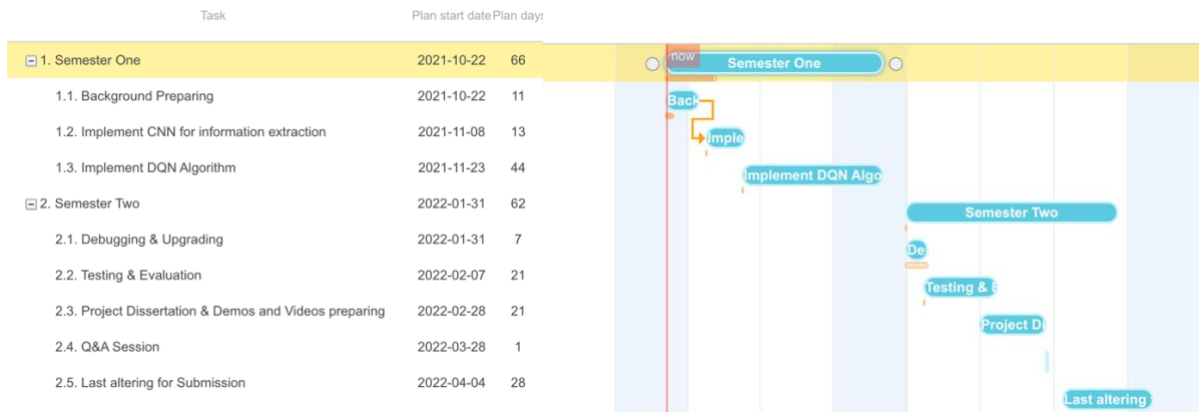
User Experience Consideration (UX):

Although this is an algorithm-based project, I would like to transfer to the users more visualizable experiences during the use. Specifically, the output of the training information (Award of this round, time taken...) would be shown on the UI diagram above, that is, other than the big begin bottom, a window is also shown to allow those outputs.

This information will help both user and me to analyze or understand what the intelligent agent is doing, and to illustrate what state it is currently stands on.



| Project Plan



| Risks & Contingency Plan

Risks	Contingencies	Likelihood	Impact
Software Failure	Recover from backup files	Not likely	High
Running out of time Software Failure	Apply for later submission	Medium Likelihood	High
Programming problems	Implement a easier version which takes readings and inputs directly from gym	Medium Likelihood	Medium

	and pass them to existing DQN agent		
--	-------------------------------------	--	--

| References

[0] 'Atari games'^[0]: <https://www.atari.com/>

[1] 'gym'^[1]: <https://gym.openai.com/>

[2] TensorFlow^[2]: <https://www.tensorflow.org/>

[3] Keras^[3]: <https://keras.io/>

TensorFlow vs Pytorch: <https://towardsdatascience.com/pytorch-vs-tensorflow-spotting-the-difference-25c75777377b>

Deep Reinforcement Learning for Atari Games Python Tutorial | AI Plays Space Invaders : <https://www.youtube.com/watch?v=hCeJeq8U0lo>

BCS Project Criteria: <https://www.bcs.org/media/1209/accreditation-guidelines.pdf>

Deep Q-Learning Network: <https://towardsdatascience.com/deep-q-learning-tutorial-mindqn-2a4c855abffc>