****

**EE 393 – Semester Project**

**Prepared By:**

Baki Koçer

Koray Eser

Umut Çakan

Mehmet Batu Özmeteler

**Relevant Links:**

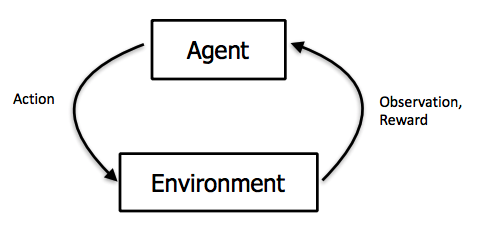
https://github.com/OctThe16th/PPO-Keras

https://blog.openai.com/openai-baselines-ppo/

**What is Reinforcement Learning?**

Reinforcement learning, in the context of artificial intelligence, is a type of dynamic programming that trains algorithms using a system of reward and punishment.

A reinforcement learning algorithm, or agent, learns by interacting with its environment. It receives a reward state depending on how it performs. In contrast, it also receives a penalty for working incorrectly. Therefore, gradually, it makes decisions to maximize its reward and minimize its penalty. Given the fact that this approach is inspired by behaviorist psychology, the advantage is that it allows an AI program to learn without a programmer spelling out how an agent should perform the task.



**A Glimpse of the Working Process of Our Project**

In our project, we aim to use a reinforcement learning algorithm with optimal accuracy. Firstly, we have done a profound research about this specific subject since it was our first-time diving into such an area. We have learnt a lot via reading many articles and following courses from Udemy, Youtube. We worked as a team but parted the subject in 2 divisions. Batu and Umut mostly worked on refactoring the algorithm, choosing the correct learning parameters for the process whilst Baki and Koray studied papers, wrote the code for our environment. Generally, we had a hard time while partitioning everything into classes. Lastly, we tested and analyzed the results obtained from the algorithm after combining all our work.

**-How Does Our Algorithm Work with Our Own Environment?**

We created a basic game which is simply our environment that consists of a 9x3 matrix. We imagined every row as a different level and put 1’s for obstacles and 0’s for empty spaces. We allowed our agent to choose an option out of three which are literally the column indices of matrix at each level. After selecting an option, it either encounters an obstacle or an empty space ending the episode and leaving the agent with an observation feedback. Thanks to such provision, it gets to decide whether if it wants to try or not try that option depending on the relevant feedback.