

**JOMO KENYATTA UNIVERSITY OF AGRICULTURE
AND TECHNOLOGY
SCHOOL OF COMPUTING AND INFORMATION
TECHNOLOGY
DEPARTMENT OF COMPUTING**

**NAME: KELVIN MACHARIA KARANI
REG NO: SCT211-9012/2015
UNIT NAME: KNOWLEDGE BASED SYSTEMS
UNIT CODE: ICS2405**

**REPORT FOR THE AUTOMATION OF THE SPACE
INVADERS GAME.**

1. Introduction

1.1. simple overview

The space invaders game is an arcade game that has been in existent since the 90's. This game basically involves having a ship at the bottom of the screen that can shoot bullets at other ships that are trying to invade. In the game one has 3 chances. i.e. lives. If the invader ships shoot the defender ship 3 times, the game is over, and also if the invader ships manage to get to the bottom where the defender ship is located, the game is also over.

In this report, we look up at the process of how to automate the game. This is a common challenge in the artificial intelligence community and it is usually commonly complex to automate besides the fact that the mechanism of how the game works are relatively easy.

Since it is a common challenge in AI, the automation of games, dedicated learning environments have cropped up that can make it easier to be able to get the state of the game and the actions that are supposed to happen in the game. This is so as to leave the machine learning engineer with only the part of choosing and coming up with efficient algorithms. Such learning environments include the Arcade Learning Environment(ALE), the Pygame Learning Environment(PLE) also the Open AI Gym environment.

For this project, I decided to use the Open AI Gym environment, which usually includes arcade atari games among the SpaceInvaders.

2. Methodology

Having obtained the environment for the game, I needed to know how I would go about the tackling of the problem and therefore automating the game. I decided upon using the Reinforcement Learning approach.

Reinforcement Learning is a subfield of machine learning whereby the agent (here we can think of the agent as our defender ship) learns how to behave in an environment(the environment here is the space invaders game) by performing actions and seeing the results of those actions.

The basic process of Reinforcement Learning is as follows:

- (a) The agent receives a state, S , from the environment.*
- (b) The agent takes an action, A , based on the state S .*
- (c) The environment transitions to a new state S' .*
- (d) The environment gives some reward R to the agent.*

Thus Reinforcement Learning is based on the idea of the reward hypothesis and the goal can be described by the maximisation of the expected cumulative reward.

Some Architectures of Reinforcement Learning include:

- a) Q learning.*
- b) Deep Q learning.*
- c) Double Q learning.*
- d) Dueling Q learning.*
- e) Actor Critic method.*

Upon some research and trial of the above methods, the best method that I settled with was the Actor Critic method. Though Deep Q learning, Double Q learning and Dueling Q learning also produced similar results.

In the Actor Critic method, it can be seen as basically that an actor decides which action to take and the critic tells the actor how good the action was and how to adjust it.

3. Implementation

To solve, I used deep learning, with the help of the most popular deep learning library Tensorflow. Before solving with Tensorflow though, some basic modification and adjustments need to be done and this is done with the help of other libraries.

The major modules that one needs to install to be able to train and run the program include:

1. *Tensorflow: pip install tensorflow*
2. *Open Cv: pip install opencv-python*
3. *Gym: pip install gym*
4. *Tensorpack: pip install tensorpack*

Ofcourse one needs to have **python 3** installed. For easier installation of all the required modules, in the folder there is a *requirements.txt* file that lists all the modules required, one can install them using ***pip install -r requirements.txt***.

In the openai gym environment, the aim is to train the agent so that it maximises the score. The main file is the *spaceinvaders.py* file. In this script, there are functions both for training the agent so that it can be more intelligent and also for testing the agent, that is basically playing the game. After training of the game, the file “*SpaceInvaders-v0.tfmodel*” is created which contains the weights for the model learned.

To run the *spaceinvaders* script for playing the game:

python3 --env SpaceInvaders-v0 --task play --load SpaceInvaders-v0.tfmodel

Here the command line options mean as follows:

- (a) *--env: It is basically the option for the open ai gym environment that we need, in my case I need space invaders and thus the argument is, SpaceInvaders-v0*
- (b) *--task: This is the option for what we need the agent to do or what we wish to be done to the agent, in this case we want the agent to play the game and thus the argument is, play*
- (c) *--load: This is the option for the weights file of the model that was trained, in our case the weights file for the spaceinvaders game is SpaceInvaders-v0.tfmodel and thus the argument will be, SpaceInvaders-v0.tfmodel.*

The game is run for 100 iterations. Through these 100 iterations, the output from the game is the graphics output which shows the agent as it plays and the score and also the output on the command line for the total reward that the agent is able to get.

Since in Reinforcement Learning we aim to maximise the cumulative reward, and in our case the reward is the overall score of the game, and the only way to maximise the score of the game is by not dieing, the agent learns to play the game without dieing. Over the course of the 100 iterations, the agents gets very high scores of upto 10000.

I have include a sample video (“short playing sample.mp4”) that shows the agent as it plays the game.

4. References

1. TensorFlow. (n.d.). Retrieved June 30, 2020, from <http://www.tensorflow.org/>
2. FreeCodeCamp.org. (2020, April 07). An introduction to Reinforcement Learning. Retrieved June 30, 2020, from <https://www.freecodecamp.org/news/an-introduction-to-reinforcement-learning-4339519de419/>
3. Ecoffet, A. (2017, October 31). Beat Atari with Deep Reinforcement Learning! (Part 1: DQN). Retrieved June 30, 2020, from <https://becominghuman.ai/lets-build-an-atari-ai-part-1-dqn-df57e8ff3b26>.