

## Midterm Project Proposal

# Neural Networks Applied to Game: Snake case

*Spring Semester 2018, Machine Learning class*

Department of Computer Science, Research Area: IoT. Kookmin University.

### Important date

Proposal due Tue May 2018

### Participant(s)

Alban Landry MOUNGALA: **M2017568**

**Github Account:** *albanLandry*

**Url:** [https://github.com/albanlandry/Neural\\_network\\_snake](https://github.com/albanlandry/Neural_network_snake)

### Project Overview

During the machine learning class, we studied a wide range of different concepts and algorithms taking part in a machine learning implementation process. Neural Network, one of the main tools used nowadays for machine learning can be used for a broad range of applications. For this project, we chose to leverage machine learning concepts in a simple *snake game*.

This is a simple project in which we aim to design a neural network algorithm that will be able to learn how to play snake game by himself. Especially, it should learn how to survive and how to grow by eating.

This project is to allow the participant to exercise the different concepts learned in class and show their ability to solve a problem using machine learning. It also provide to the student a hand on practical python programming.

Through this project, we attend to use separately CNN and RNN for implementing of the learning model. And compare the out coming experiment results with humans' data. The goal is to achieve the highest score possible.

The realization needs to be done with respect to some steps and requirements.

## Steps

### 1. Game implementation

- Make a fully functional Snake Game:
  - i. Define all the game logic and interactions: A snake should not hit the borders (Or labyrinth) or his own body. It should be able to achieve the highest score by eating a fruit.
- Generate the API used by the neural network
- Generate model features

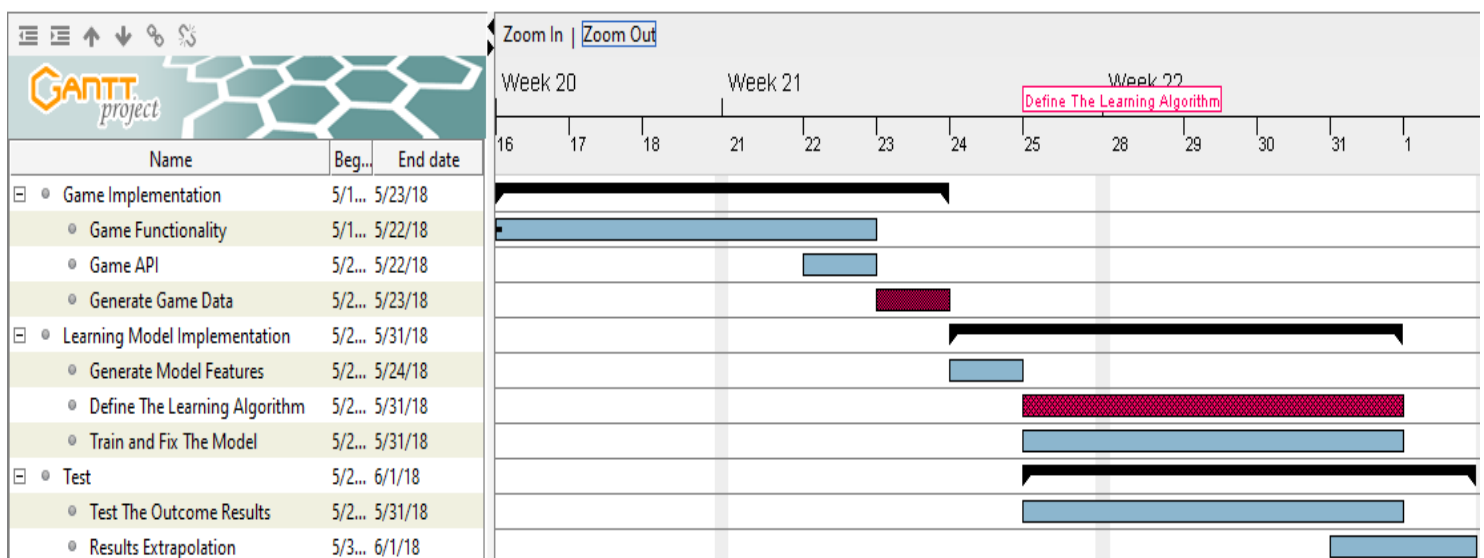
### 2. Learning

- Divide the data into training and test set
- Train the neural network

### 3. Test

- Test the outcome results

Below is the complete schedule made for the realization of the project.



## Tools

1. hardware:
  - laptop with Linux/Windows OS 64bits CPU
2. software:

- Python 3.5;
- Keras 2 for Python 3;
- TensorFlow 1.5 for Python 3;

**Skills needed**

- Programming language(python);
- Familiar with Linux, TensorFlow, Keras
- Deep learning, NNs (Neural Networks).

**Remarks**

The tools used for the project might be subjects to some changes depending on the problems we might face with some machine learning libraries.