**Project Proposal**

*Jumana Darwish*

**Mastermind**

**Project Description**

I will be creating a game that is an implementation of the board game Mastermind. Mastermind is a code breaking game for two players, a code maker and a codebreaker. The code is a pattern of four colored pegs. The code breaker attempts to guess the code and receives a score on the accuracy of his/her guess. The score is represented as black and white pegs. Black pegs represent the number of colored pegs in their correct position and white pegs represent the number of correct pegs that are not in the correct position. The goal of the game is to break the code using the minimum number of guess possible without exceeding eight attempts.

My project will implement the game using two settings. The computer against the player and the player against the computer. When the player is the codebreaker, the computer generates a random code for the player to guess. If the player wishes to be the code maker, the computer will attempt to guess the code using a machine learning algorithm based on neural networks.

**Competitive Analysis**

I have seen a few versions of the Mastermind game online and in all of them the player only has the option to be a codebreaker. My implementation is very similar when it comes to this part. However, what makes my project unique and what differentiates it from the other ones is the option I’m giving the user to be the code maker and have the computer guess his/her code.

**Structural Plan**

* Class for basic game – generates new boards ( by computer or player), handles scoring
* Class for machine learning – will load my model and connect it to game
* Class for training the machine learning model – will generate the data/examples, train my ML model
* Class for graphics

**Algorithmic Plan**

The trickiest part of my project is coming up with the best way to train my machine learning model. To do that, I plan to generate a few different sets of examples to train the model, with each set containing thousands of random examples.

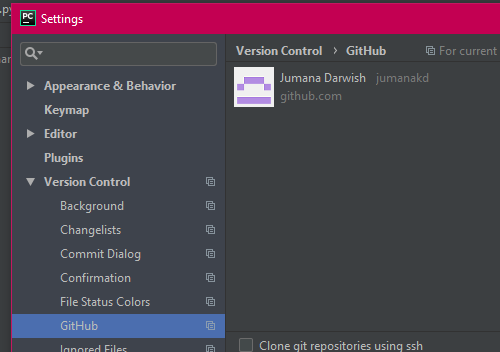
1. A large set of examples each containing a guessed code, its evaluation based on the hidden board, a second guessed code and finally a score. The score is based on the number or black and white pegs the guess gets. Black is represented by 2 and white represented by one, so the higher the score the better.
2. My other sets of examples will be the same as the previous set except that each contains the current guess and ***n*** previous guesses.

**Timeline Plan**

* **TP1**: by TP1, I plan to have my basic game setup. I will have the basic game class and the training model class. My first machine learning model will be trained.
* **TP2**: by TP2, my goal is to have a fully functioning game with a working AI and all the features would be implemented.
* **TP3**: by TP3 I will have the full game running with all the graphics done.

**Version Control Plan**

To back up my code, I plan to use GitHub. I directly connected my GitHub account with the IDE I use (PyCharm) which makes a clone of my project on GitHub and updates it whenever I save my code.



**Module List**

* Pygame
* Keras – using tensorflow backend
* Additional helpful libraries –
  + Numpy
  + Pandas
  + ~~Sklearn~~

**TP2 update**

No changes were made to my plan. My backend is now fully functioning with the machine learning AI working pretty well!