**Project Specification**

Unumandakh Bayandelger

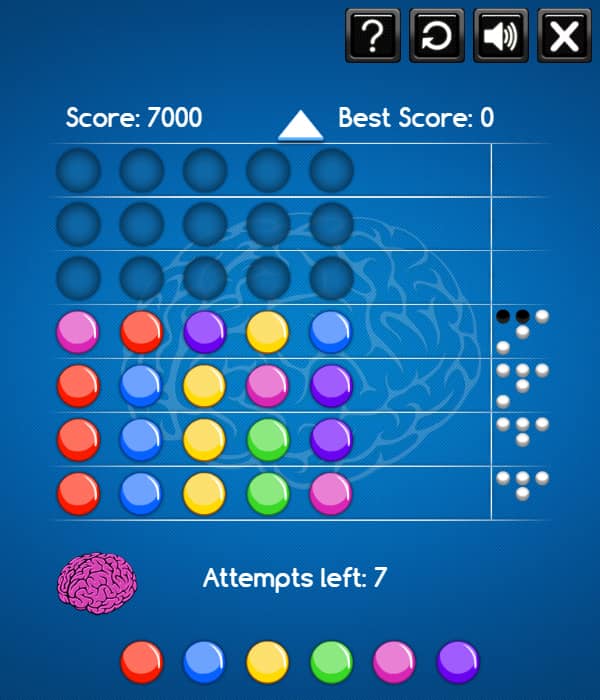
Prof. Vladimir Georgiev

COS221a\_S2019

15 March 2019

1. **Description**

The project I have chosen will be a simple GUI game based on [Mastermind](https://en.wikipedia.org/wiki/Mastermind). I will be mainly using [SFML library](https://www.sfml-dev.org/) for the GUI implementation, with [minimax algorithm](https://en.wikipedia.org/wiki/Minimax) for the bot to play against. Game’s progress will be available on my [github profile](https://github.com/uno-b/SFML_Mastermind). Game will be similar to the following picture:



1. **Class hierarchy**
2. Asset manager will handle the textures, sprites and fonts.
3. Input manager will listen to events during the program.
4. Game logic will be included in their respective states (e.g. gameplay state).
5. The constants such as the speed of the game, screen size etc. will be in the definitions class.

This hierarchy is only based on the current version, and may be subject to change.

1. **Statements**

Currently, some of the data structures to be included are:

* Array for most part, particularly in game logic
* Stack or queue for storing user scores
* Binary tree just for the sake of it (learning purposes)

There will be a polymorphic function called *TakeTurn()*, which will be changed in classes PlayerVSAI and PlayerVSPlayer from their parent class GameplayState. The *TakeTurn()* function in PlayerVSAI will allow the user’s turn to come immediately after the bot finished its move. In PlayerVSPlayer, the player’s perspective changes.

May introduce TCP/UDP for multiplayer feature from different applications/locations.