**CIS 434 Software Engineering – Cleveland State University Department of Electrical Engineering and Computer Science**

**Spring 2019**

**TITLE: CHESS GAME**

**Course Project Proposal Submission**

**02/28/2019**

**Group members**

* + **Ethan Kline**
  + **Megan Keleman**
  + **Victor Ipinmoroti**
  + **Matthew Nathal**

**Project Description**

The purpose of the project is to create a chess board game with a simple but clean UI. The board will be in the main window with a side or top bar that describes the movements each player makes. There will be two modes: player vs player and player vs computer. The GUI will highlight all legal movements for the turn player’s currently selected chess piece. Given this we are aware that we need to:

1. Create a graphical user interface for the board and game menu
2. Manipulate this interface to change based on certain events.
3. Create the movement set for each peace on the board.
4. Create an AI for someone to play against.
5. Connect the game logic with the board GUI.

**Method and tools that will be used**

To accomplish the above tasks, we will need to keep track of what each team member is doing. This will be done by using the GIT platform. As such, we can use different IDEs (i.e. intellij and eclipse) for our chosen programming language (JAVA). To create the GUI of the board and menu we will use javaFX or swing; we have yet to decide which library to use but intend to pick one at a later date.

An overview of how the GUI will function is that the turn player will be able to click on any of their movable pieces which highlights the legal movements for that piece. The turn player can then click on the square they wish to move the selected piece to or click on the chess piece again to cancel out of their current selection. Such functionality enables intuitive move selection for the user and prevents an inexperienced player from attempting illegal moves.

We will also be implementing a sidebar that displays the previous moves made during the game. This allows each player to review how they got to the current game state and potentially where they went wrong in their move choice. The sidebar will also allow us to implement a way to revert the game state to a previous turn. This is useful for testing move options not previously considered until a later point in the game.

Aside from this we also have the logic aspect of the game to consider which will be bound to the GUI later. The board will be an 8x8 matrix of buttons and chess pieces will be icons stored in separate instances of a class that dictates their behavior on the board based on the type of chess piece represented by the icon. Such behavior includes the movement of the chess piece and special properties for that piece such as en passant and castling.

Another possibility would be to have a superclass called ChessPiece with subclasses for each type of piece. We have yet to decide on the exact implementation for each chess piece. As for the AI, we still need to do a lot of research to even begin implementing it. It is the last thing we will do to finalize the project.

**Team member responsibilities**

Since we are still in the initial stages of the project each team member’s role is not finalized and is subject to change as time goes on but for now these are the roles that have been given:

* + Ethan Kline - Game logic and conceptual design
  + Megan Keleman - AI research and implementation
  + Victor Ipinmoroti - Graphical user interface and logic binding
  + Matthew Nathal - Chess piece movement and programming

**Conclusion**

At the end of the project we will have created a game of chess that has a streamlined user interface which will allow users to play against one-another or against an easy to moderate level AI. Ultimately, we aim to simplify the game enough so that anyone can enjoy playing.