Chess Engine

https://github.com/comp195/senior-project-spring-2023-chess-engine-classic-chess-game.git

Robert Kelly (r_kelly5@u.pacific.edu)

Lorena Aguilera (l_aguilera2@u.pacific.edu)

Paxton O'Bryan (p_obryan@u.pacific.edu)

Last Revision: 2/3/2023

Overview

      Our goal in this senior design project is to create a single player chess game. This will be split into two different elements. The first element will be the UI of the project, creating a board to move pieces and represent the moves of the player and the moves of the engine. The second element is to create the chess engine itself, which will take into account the state of the board and make decisions accordingly.
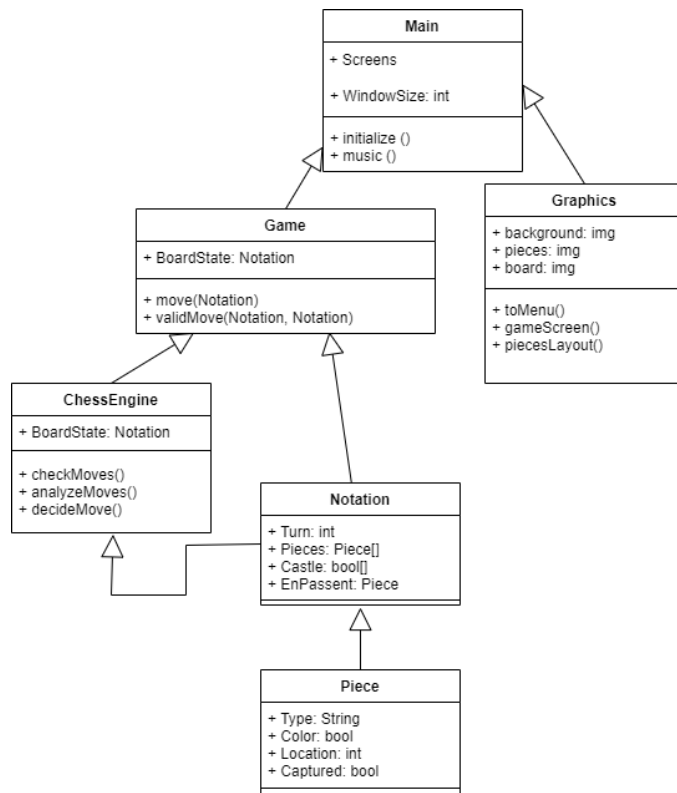
System Architecture
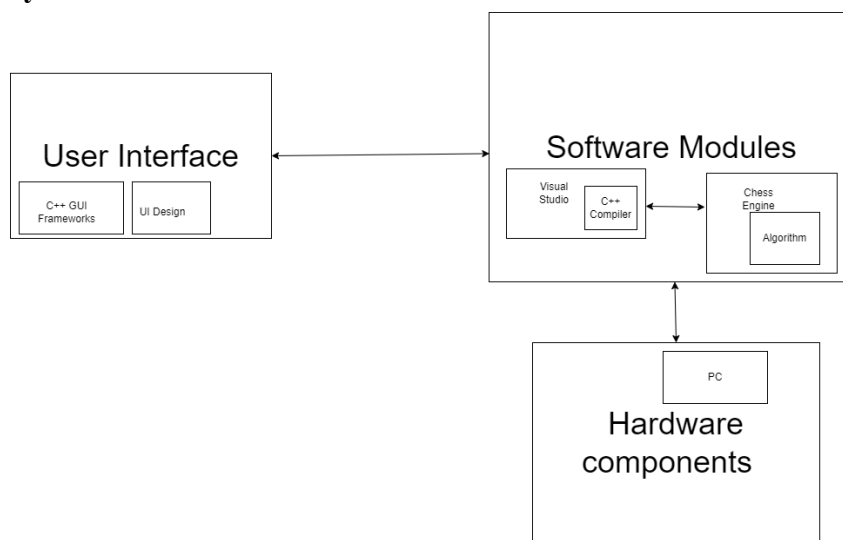
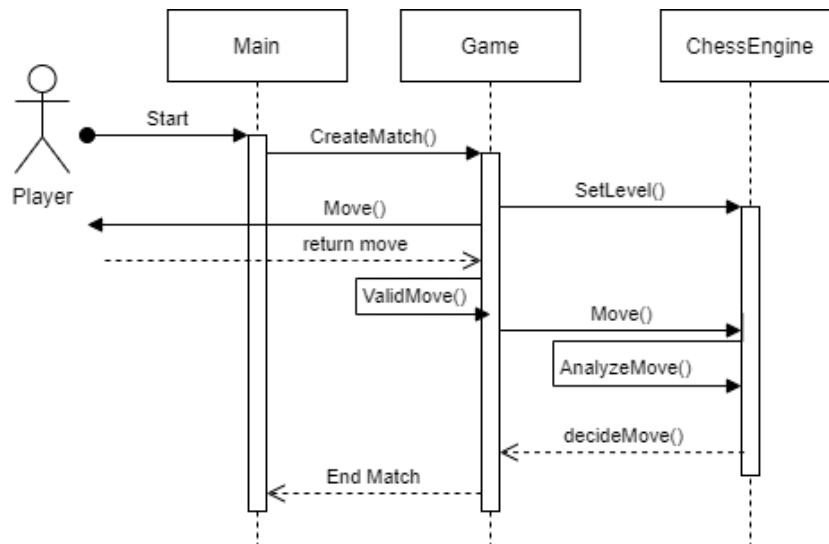**Hardware, Software and System Requirements**

      Unknown what other libraries we may need at this point. A pc will be needed to run the program, but it is unknown what system requirements are needed at this time. At this stage, we recommend a newer gen computer, with higher processing power.

Software Design
      Class Diagram
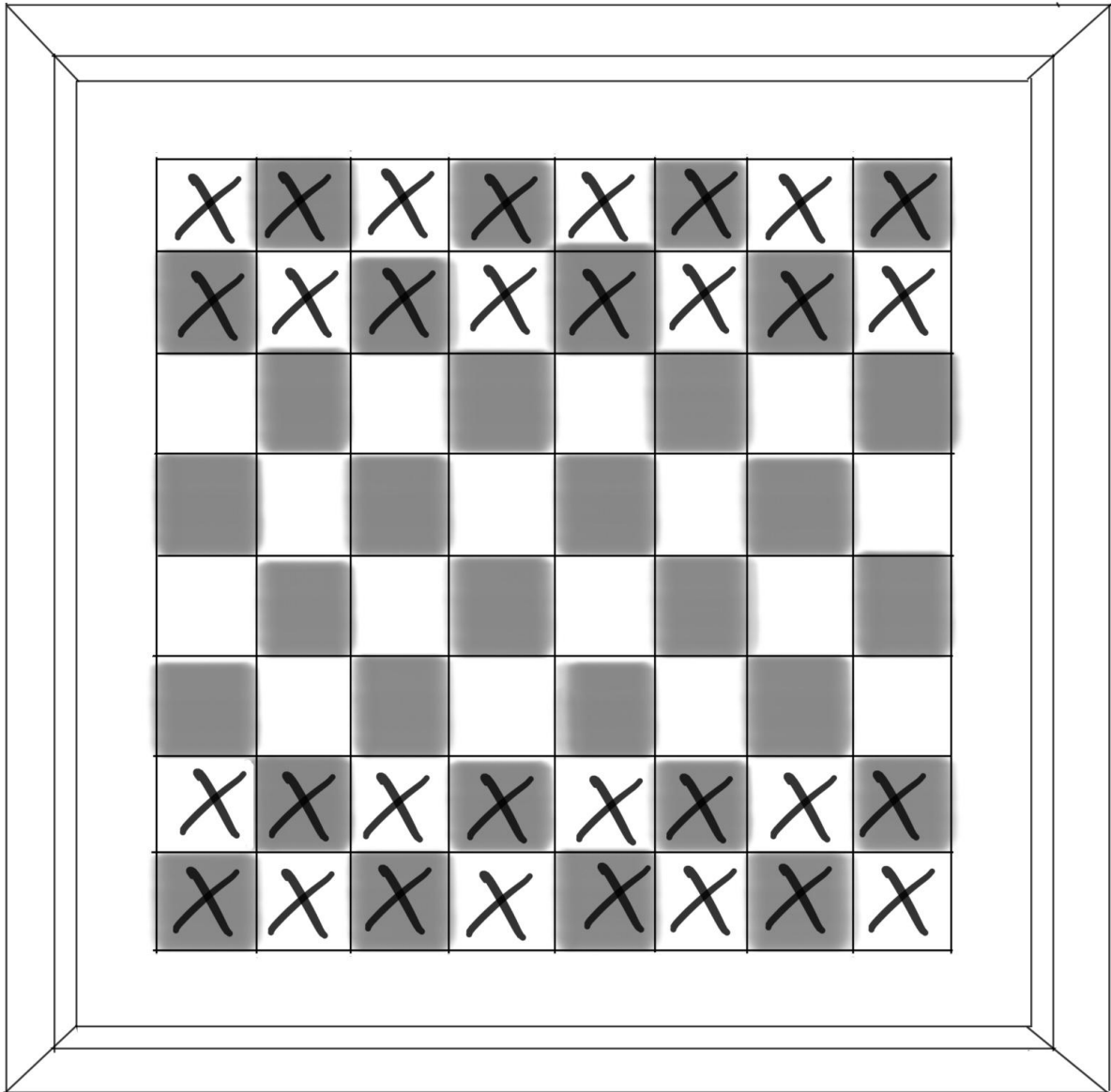
Main

+ Screens

+ WindowSize: int

+ initialize ()
+ music ()

Graphics

+ background: img
+ pieces: img
+ board: img

+ toMenu()
+ gameScreen()
+ piecesLayout()

Game

+ BoardState: Notation

+ move(Notation)
+ validMove(Notation, Notation)

ChessEngine

+ BoardState: Notation

+ checkMoves()
+ analyzeMoves()
+ decideMove()

Notation

+ Turn: int
+ Pieces: Piece[]
+ Castle: bool[]
+ EnPassent: Piece

Piece

+ Type: String
+ Color: bool
+ Location: int
+ Captured: bool

Interaction Diagrams

Main

Game

ChessEngine

Player

Start

CreateMatch()

SetLevel()

Move()

return move

ValidMove()

Move()

AnalyzeMove()

decideMove()

End Match

Design Consideration

User Interface Design

Wireframe

- Dimensions may change in respect to the software
- Border should appear to look "wooden"
- Options to change board color may be incorporated
- Gold trim will be applied around the board

Glossary of Terms

None at this stage

References

None at this stage