

Group 8 Final Presentation

...

By Salvador, Christianna, Corey,
Nicholas, and Hamza

Introduction

- Our goal with this project was to create a chess application written in Python and built with Tkinter with player vs player and player vs bot game modes.
- Our chess application offers three levels of difficulty, Easy, Normal, and Hard, when playing against the bot. The bot uses the Stockfish chess engine to create moves.
- The players, and bot's score of wins, losses, and ties are tracked across games.
- Our application also prevents illegal moves from being selected. For example, if a King is in check, the only moves selectable will be ones that will take the player's King out of check.

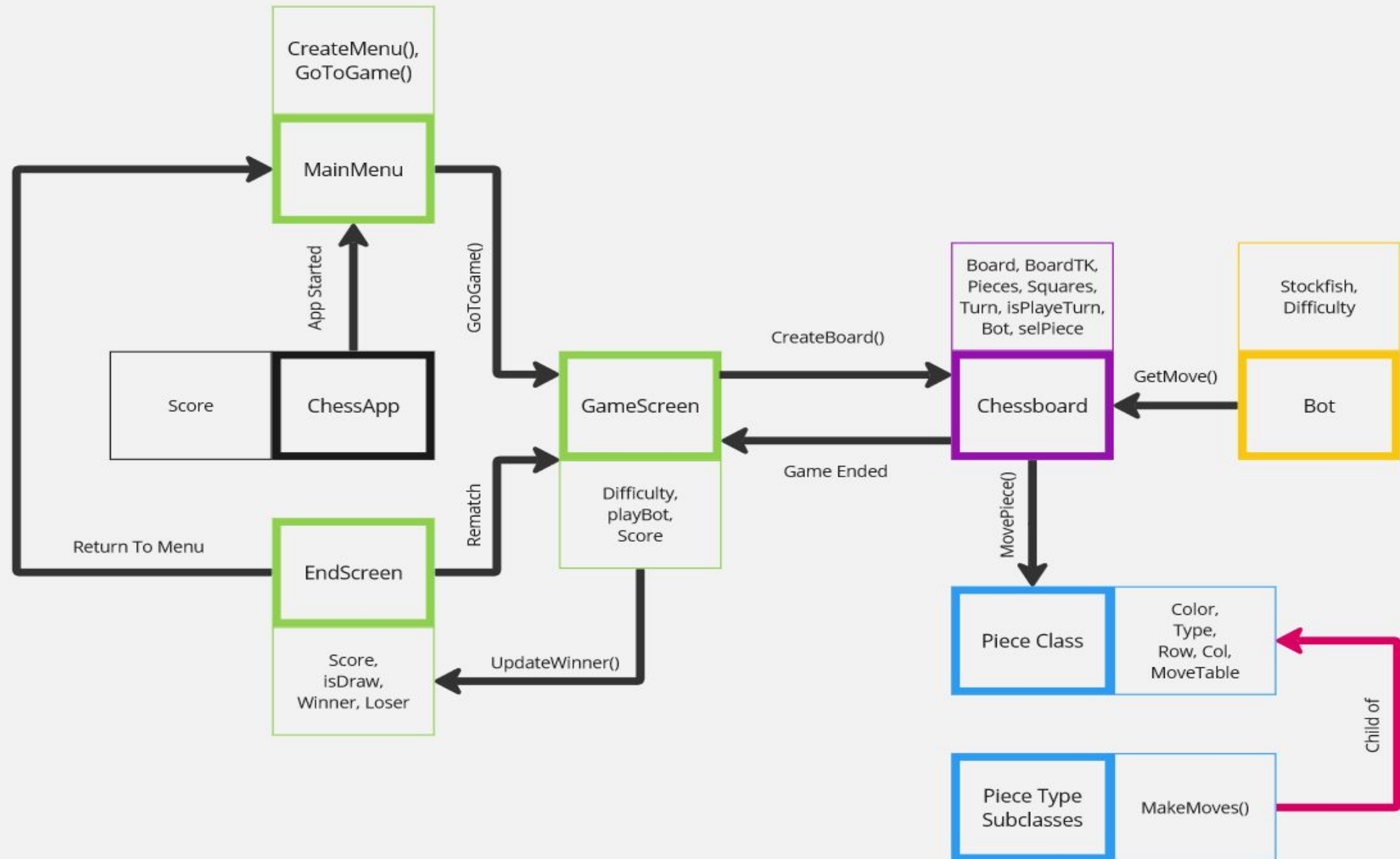
Technical Details

- As pointed out earlier, the application is written in Python. The graphical interface of the application is built with the natively installed Tkinter library.
- Python was chosen as it had Tkinter natively installed and because Python is an object oriented language. Because of Python's object oriented design, the GUI, chessboard, and pieces are organized in classes.
- The user interface screens and frames are all classes that have their own properties. The chessboard itself is a class that handles keeping track of the board, and representing the board through the graphical interface. All chess pieces are subclasses of a parent piece class.

Technical Details Cont.

- As stated before, the Stockfish engine is used to determine the best move for the bot. Following the object oriented design of the application, the bot also has a class for itself.
- Why classes? Using classes made it so group members could isolate their work in their designated section. This made working together as a group more manageable as members weren't interfering with each other's work.
- Using classes also made it very manageable when it came to working with each type of chess piece. Because of polymorphism, each piece has their own unique moves while all sharing the same properties and general methods from the parent class.

Class Diagram



Challenges

- Although designing our project with classes did make the project more manageable in a group, there were still some challenges.
- Figuring out how to prevent illegal moves from being selectable was very difficult and required the chessboard logic to undergo two massive reworks.
- This challenge was overcome by simulating a player's moves to see if that move will put their King in check. If the simulation of the move did put the King in check, it will remove that move from the piece's list of moves.

Future Work

- While we were able to accomplish our core objectives, there were some minor goals we were not able to finish due to time.
- One of the minor goals that could be implemented due to time is the right submenu in the game screen. This menu would have displayed the current turn, the player's stats, chess pieces taken, and a time limit.
- Another thing we couldn't implement because of time restriction is adding animation to the chess pieces when moved. This would've shown a much smoother transition between turns. Instead, the current lack of animations makes it hard to understand the moves done by both players.

LIVE DEMO