**Term Project Proposal**

**Project Description:**

Project Name: The Queen’s Gambit

Project Description: I will be making a 3D rendered chess game using the Panda3D library. Furthermore, I will be incorporating a multiplayer feature using the socket library which we have already dealt with in our homework and lectures to allow 2 remote players to play against each other.

**Competitive Analysis:**

This project will be similar to any other chess game in terms of implementing the rules of chess, but will stand out by implementing a 3D user interface and a multiplayer feature. Other unique features include: changing the camera view, highlighting the squares on the board for the valid moves of a selected piece, simple animations when moving a piece, selecting/changing the chess board theme, being able to: resign, offer a draw, send a predefined message, or ask for an undo/rematch to the opponent.

**Structural Plan**:

I plan to have 3 main python files for this project: one for implementing the rules of chess (the engine), one displaying the interactive 3D UI and managing the client side of the socket connection (user interface and client code), and one for managing the clients connected to the server (server code). In the engine, I will have 3 classes: GameLogic, Moves, and ChessPieces where the GameLogic handles making moves, updating the board, and handling checkmates/stalemates, Moves to store a Move object in the move log, which allows the undo function to work, and ChessPieces to calculate the valid moves of a chess piece.

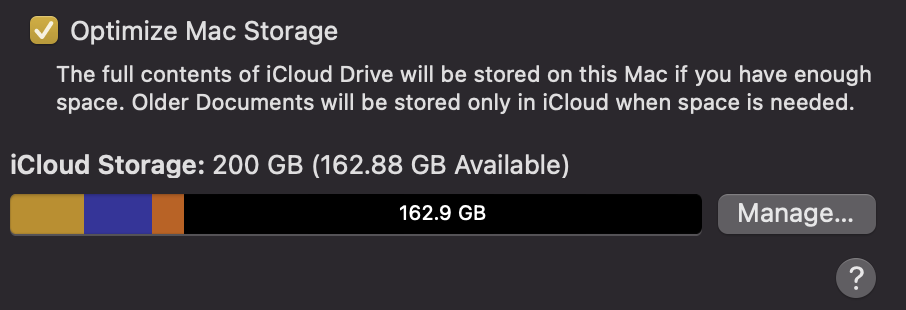
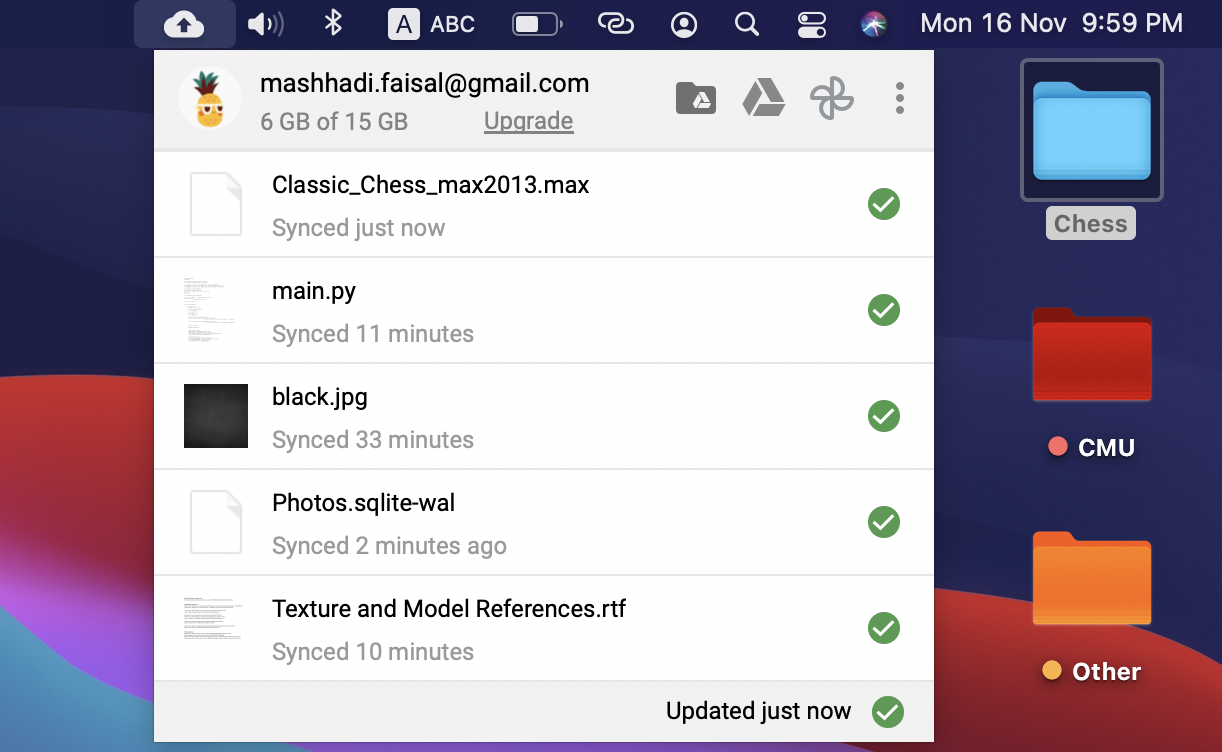
**Algorithmic Plan**:

The trickiest part of the project is calculating for checkmates/stalemates: the winning or drawing condition for a chess game. To implement checkmates/stalemates, I will need to create a function that calculates, from the King’s position, all pinned pieces and all attacking pieces. If the king is in check, then the algorithm will need to evaluate whether it is also a checkmate, if the king is not in check, and the current player has no valid moves, then that means the game is a stalemate (draw).

**Timeline Plan**:

By TP1, I plan to finish the majority of the 3D user interface and plan to have a decent start in the engine code. By TP2, I plan to have finished the 3D user interface and the engine of the chess game, this includes features such as highlighting all possible moves for a given chess piece and making simple animations for a move. By TP3, the client code and server code should be finished, so my chess game will be available to play via multiplayer.

**Version Control Plan**:

I am backing up my code (and computer desktop) using Google Drive, and my computer using iCloud.

**Module List**:

* Panda3D
* Sockets

**TP2 Update:**

I have implemented the majority of my design ideas in my project except for the multiplayer feature. I will still be implementing a multiplayer feature; however, I will not include the initial design plan of allowing players to send pre-defined messages to each other. Players will still be able to offer a draw, resign, undo a move or play again. Additionally, I will not be writing my own server code, I will be using the same server as I did in homework 7 and 8. Due to this change, I will also be including a simple chess AI.

**TP3 Update:**

I have implemented all my design plans, this includes having a multiplayer feature with the necessary protocols that allows communication between 2 players, additionally being able to play against a reasonably intelligent AI that uses the minimax algorithm with alpha beta pruning to make moves. I have also included background music and clicking sound effects when playing.