

Mona Lin (monal)

Gomoku AI

This project will focus on implementing two main modes of the game gomoku, player vs. player mode and AI mode. To pass MVP, basic player vs. player and AI modes will be implemented using the Minimax algorithm. For post, MVP, sockets and better UI will be emphasized,

Competitive Analysis:

<https://www.geeksforgeeks.org/minimax-algorithm-in-game-theory-set-1-introduction/?ref=lbp>

Looking at the geeks for geeks page on minimax algorithms, I plan to implement a similar pseudocode. Most of the minimax algorithms I looked at used Tik-Tac-Toe as an example, and my code will be similar in that, based on the state of the game, my AI will calculate the optimal move of the game. However, gomoku is on a vastly bigger board, with more combinations of moves.

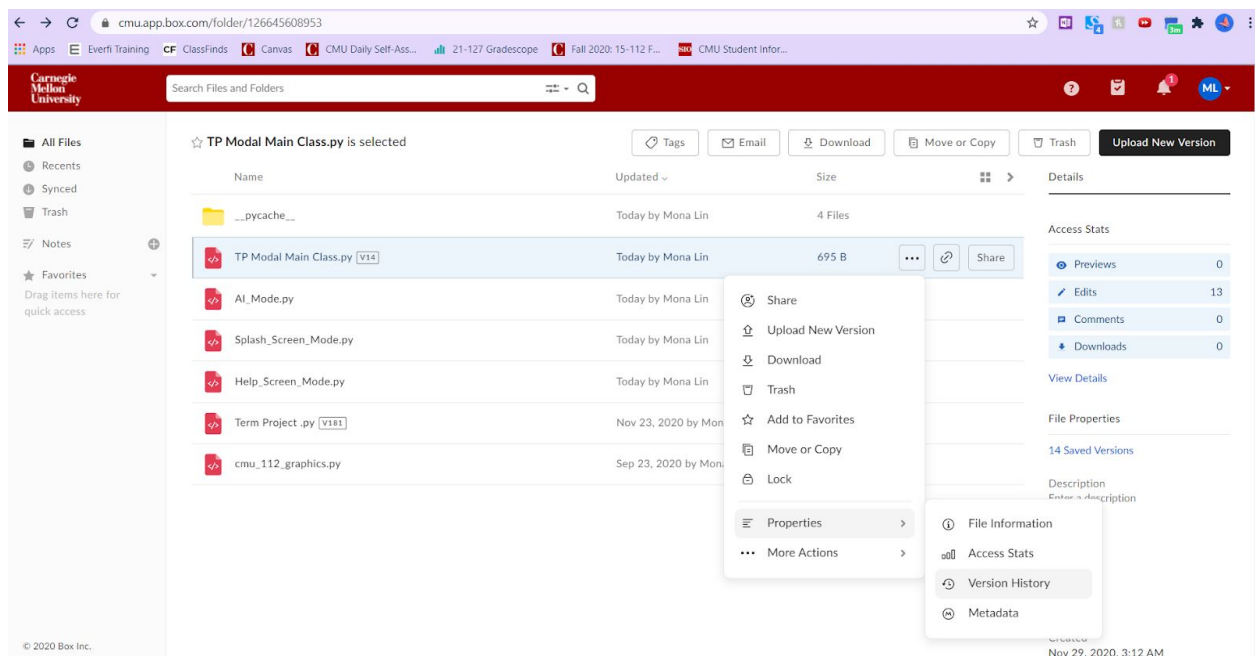
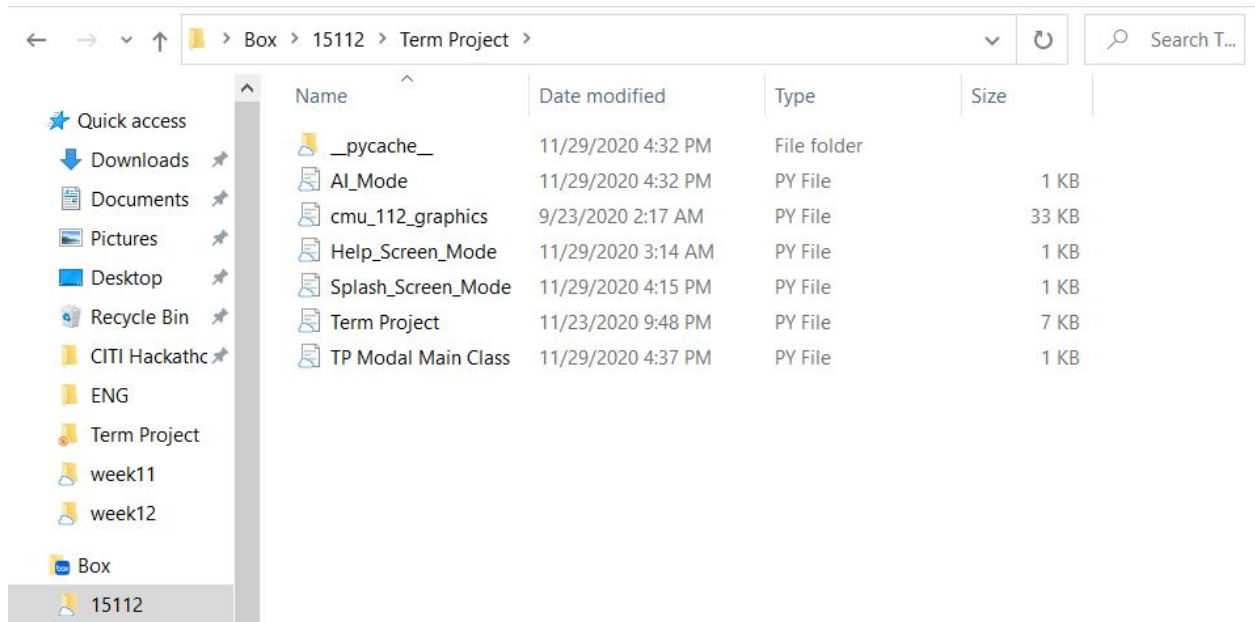
Structural Plan: I will be using a Main Modal class to organize all my files. Each mode will have it's own class, and ideally its own python file, which will be imported and run from the main file. Thus far, the files and classes I plan to implement are a Splash Screen Mode, a Main Menu Mode, Help Screen Mode, a Player and Player Mode, an AI Mode, and an Online Mode. If I get everything done, I plan to implement a socket mode to enable remote player vs. player gameplay.

Algorithmic Plan: To approach the AI portion of my project, I will use a minimax algorithm to design a unique heuristic function for my AI gomoku. Thus far, my heuristic will assign points on whether...

1. The maximizer (our AI) can get 5 pieces in a row, in which case, the board is assigned a score of infinity
2. The minimizer (the player) can get 5 pieces in a row, and the board is assigned negative infinity
3. The maximizer can get two connected 3 in a rows where none of the ends are blocked -> infinity score
4. The maximizer can get a a disconnected 3 in a row 0 -> +10 points
5. The maximizer can place a piece next to another piece -> 3 points
6. For the rest scenarios which aren't listed yet, the algorithm will assign points based on similar reasoning.

Timeline Plan: By TP1, I hope to fully complete the player vs. player mode, splash screens, main menus, and help menu, and the other modes. If I have time, I would like to also get started on the minimax algorithm. By the TP2 deadline, I plan to get the minimax algorithm working for the AI to pass MVP. Post-MVP, I plan to work on sockets and polishing up the UI before I turn the term-project in.

Version Control Plan: All my code is backed up in the desktop version of Box. In addition to saving my code in real time, Box also has the option to restore previous versions of the code.



Module List: No modules will be used pre-MVP.

TP2 Update

No changes were made to the design plan. However clarifications were made about my Gomoku AI needing 3 difficulties. I plan to tweak the UI slightly to accommodate these changes

TP3 Update

I had difficulty using sockets, but music and sound effects were successfully implemented using the pygame module. Additionally, the online feature using sockets was replaced with a leaderboard.