Project Proposal: 3D Connect4 Game

Riya Bhatia

1. Description

The name of this project is 3D Connect4, but this may be changed later in the term project season. 3D Connect4 is essentially a multi-mode Connect4 game created in 2.5D isometric, with a human-human mode and a human-computer player mode. The human-human mode will feature a 2-player game that will allow two unique players to play Connect4 via the application, while the human-computer mode will feature an AI player that will compete against a singular human player.

2. Similar Projects

Connect4 has been designed before, and there are many algorithms online that will allow me to create a 2D Connect4 game. These 2D games consist of a simplistic, 2D grid that doesn't demonstrate the way in which pieces fall into the grid and other such realistic features. They simply involve placing the pieces in specific areas in the grid, which doesn't provide an engaging experience for users. Connect4 also exists as a game in real life (not virtually), which people can readily find in stores.

My project will be different in the sense that it will provide a realistic game experience when users play Connect4. Specifically, it will feature a 3D model of the Connect4 game and piece features. Moreover, in the human-computer portion of the game, it will use the minimax algorithm, which has been utilized in Connect4 games previously, but on top of this, it will use alpha-beta pruning to refine the current AI model for players.

3. Structural Plan

The final project will be split into different functions, and the code will be modularized, such that:

- Script for 3D Connect4 board
- Script for 3D game pieces (both red and yellow coins)
- Functions for human-human player in the Connect4 game, and algorithm to process the pieces placed into the board [mode 1]
- Functions for human-AI player in the Connect4 game [mode2]
    - Algorithm for AI player using minimax algorithm
    - Refinement of AI player using alpha-beta pruning algorithm
- Main function to piece all scripts and functions together

4. Algorithmic Plan

The portions of my project that are most difficult algorithmically is the human-AI player portion of the game. Specifically, I want to create an AI player that will be able to play Connect4 alongside a human player and be able to respond to the human player's moves.

In order to solve this problem, I plan to implement the minimax algorithm, which is a recursive algorithm that uses backtracking to choose the next specific move for a multi-player game. Since recursion and backtracking has been covered in 15-112 lectures and recitations, this can be implemented. However, minimax has been applied to Connect4 in the past multiple times, so in order to improve existing solutions, I plan to implement alpha-beta pruning, which is a specific search algorithm. Here, the nodes that have already been established in the search tree are removed, which provides a more efficient minimax algorithm.

These features can both be implemented relatively easily using Python. Minimax is implemented using recursion and backtracking, and alpha-beta pruning is an optimization to the existing minimax algorithm.

5. Timeline Plan

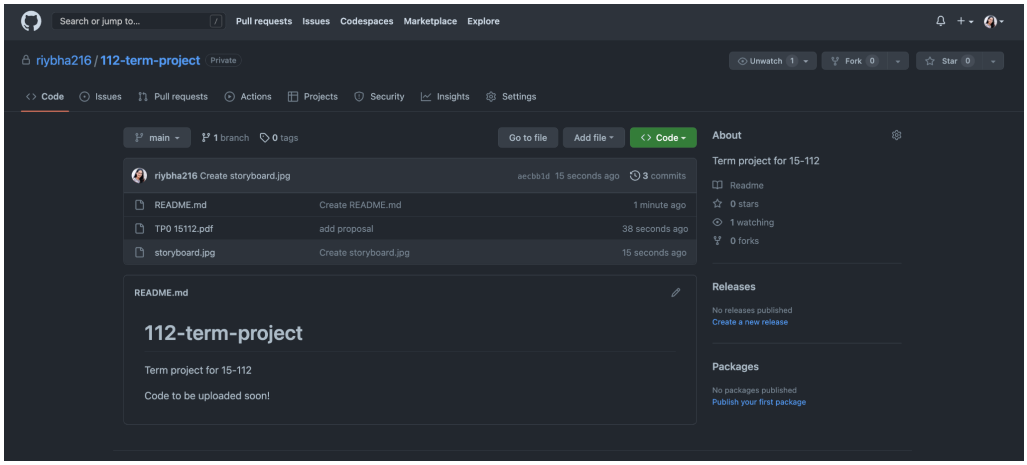
My current timeline is as follows:

1. Finish 2D Connect4 model by TP1 deadline, including minimax algorithm and algorithm for the high-level Connect4 game (Nov 20)
2. Create 3D Connect4 pieces by (Nov 23)
3. Create 3D Connect4 board (Nov 27)
4. Put all the pieces together, resolve bugs, have production-ready code (Nov 30)

Thus, all work should be finished by the TP2 deadline.

6. Version Control Plan

In order to back up my code, I am using a private GitHub repository (called 112-term-project). Since GitHub stores code on the Cloud, this is not a local process. However, in order to push my code to the repository, I will use GitHub Desktop, which allows you to make changes to code effectively. I can also use Git commands on my Terminal, such as git push.



7. Module List

I am not using any additional modules before MVP. However, after MVP, I may use numpy in order to more effectively draw the 3D board.

TP1 Update

I have finished creating a 2D implementation of Connect4, with both the human-human mode and the human-AI mode using the minimax algorithm and alpha-beta pruning. The minimax can still be improved slightly (as in its accuracy can be improved), which will be done before the TP2 deadline. This is according to my timeline that I mentioned in the proposal as well.

TP2 Update

Because of extenuating circumstances and a misunderstanding on what a 3D board looked like in the first place, unfortunately, I was not able to complete the core of my project's code. I implemented a portion of the code for what I initially thought a 3D board looked like, but since that was not truly a "3-dimensional" visual of the Connect4 board, I had to restart.

After this realization, I finished implementing the visualization of the 3D board, including the grid of cells and 3D visualization of the board. The player can also click on individual cells (so the board is responsive to user input), and play Connect4 with the flat squares as tokens. A win detection is also partially implemented (a backtracker was initially implemented, but that was slow for the 3D board), and a starter page is implemented as well.

TP3 Update

A couple of new features have been added since the TP2 deadline:

- Implementation of the 2.5D isometric board so that all cells are responsive to user input, regardless of the number of rows, columns, or total number of grids
- Cube generation such that the cube starts at the topmost board when the user clicks on a specific grid cell
- Animation of the cube falling onto its specified board location
- Showing which player has the current move due at the top center of the screen
- Three distinct levels (easy, medium, and hard) that feature different board sizes, which increase with difficulty
- All mouse presses disabled if column is full or if game is over
- Pieces closer to the user are drawn first to create the illusion that there is motion along the z-axis
- 2D map that maps each individual cell on the 3D board to 2D so that players can view where they can place pieces more effectively
- Players can pop out pieces from the bottom grid only by clicking on the piece they want to remove (they can remove their piece or their opponent's piece as well!)

Because of the minimax algorithm being exceptionally slow and poor at detecting optimal moves, I have decided to remove the human-AI portion of this project and instead, refined the

user experience for the two-player game by adding levels, the 2D map, ability to pop out, animation, and more listed above in the newly-added features list.