Engineering Track: Final Project Proposal

B351 / Q351

Basic Information

B351 Final Project: Connect-4

Short Project Statement
Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. A blind text should contain all letters of the alphabet (but this particular one doesn't) and it should be written in the original language.

Team Members

1. Name: Cassandra Chaput
2. Name: Jack Kelly
3. Name: Mahesh Chaganti

1 Problem Space

1. The problem space: Connect-Four is a two-player game, where players alternate placing pieces on a vertical board. The board is 7 columns wide and 6 rows tall. Each player is designated by yellow or red. The objective is to win the game by being the first to place four pieces in a horizontal, vertical, or diagonal line. Some of the challenges include the boards position. Since it is vertical, the pieces drop to the lowest available row within that column. A constraint of the game is a columns size. A column is full when it contains 6 pieces. A full column means no other pieces can be placed in the column. The number of pieces is also a constraint. Players start with 21 pieces of their color, and the first player to achieve a line of four connected pieces wins the game. If all 42 pieces are placed and neither player has four pieces in a row, the game is a draw. Variations in this problem space include the colors designating players (sometimes black and red pieces)and the order in which the color (or player) places a piece first.
2. Historically, players often follow an offense or defense strategy to achieve the objective. For example, if player 1 was focused on achieving four pieces in a row, theyd be on the offense. A player is on defense if they are focused on blocking the other player from achieving four pieces in a row. This is usually done by thinking of the players next move and how you could react or how you could prevent a move.

2 Algorithms

1. We will implement a minimax search algorithm. The minimax tree alternates between Player 1 and Player 2 . The minimax algorithm will use a heuristic to assign a value to each board state on the fringe. The advantageous states will be represented as larger for the advantage of yellow, and smaller for the advantage of red. This will be useful when implementing the use of forward checking (check the value of the board before deciding if it is a potential positive move) before expanding the state. This will cut down on the states expanded, as the non advantageous states will be ruled out immediately rather than being expanded.
2. Due to to the nature of the minimax algorithm paired with an efficient heuristic function, will allow us to gain the full advantage of A* search. Prior to this being implemented, we do aim to incorporate a

Breadth First Search algorithm before the implementation of A*. This will be done to ensure the data structure of the board and player classes are appropriate, as changing things too late will result in a total rewrite. This will also be done to showcase the difference that an efficient algorithm can have on a final programs runtime.

3 Third-Party Libraries and Technologies

If you intend to use third-party tools or technologies, please explain the following for each technology:

1. What technology will you be using?
2. What will it be used for / how will it assist you in your project?
3. How will you demonstrate your knowledge of the topic area despite off-loading work to the third-party technology?

1. Replit will be used in order for all of the group members to have the ability to contribute to the coding section of the project. Replit has the capability for real-time editing between multiple users at once. There is a live chat feed, for easy communication between all of us as well.
2. NumPy has a function called zeros(). This will create a matrix full of zeroes, representing the empty slots/positions. The flip() function will reverse the order of an array of elements along a specified axis, preserving the shape of the array. We can use this to print the board for each player. This package will also allow us to create the board object as it allows for n-th row matrices.
3. Pygame has a module for drawing shapes (pygame.draw). We will use this to draw the game board, and pieces. The board will be one large rectangle with 42 white circles indicating open spaces/holes in the board (7 columns by 6 rows). We will also be using pygame.draw to illustrate each players move/drop into the board. There will be up to 21 yellow circles and 21 red circles, each indicating a players move/drop. In order for our game to be displayed we will use pygame.display() and pygame.display.update()

List this for **all** non-standard libraries you will use. For example, the first item for many Python developers might be numpy, and the first item for many Javascript developers might be jquery. You may always opt to use more third-party tools later by presenting the proposal modification request form to your mentors at one of your check-ins.

4 Project Goals

In this section, please list the specific action items that you intend to complete by the end of the project. Include a range of reach (A-range), target (B-range), and safe (C-range) goals. Each set of goals should build on the previous set. This section will serve as a rubric used to assign a majority of your overall project grade, so be as specific as possible. You may use a bulleted format. This section should be no longer than 1 page single-spaced.

4.1 C-range Goals

Fallback goals for the project should go here.

1. Program/AI is able to play connect-4
2. Program is able to make moves and impact the game board.
3. Program verifies each of the opponent's moves is valid, and executes them. If the move is invalid, state the error (e.g. space occupied, column full, lower spaces unoccupied... etc.)

4.2 B-range Goals

Your target goals for the project should go here. These must go beyond the specific algorithms covered in class.

1. Program produces only valid moves
2. Search algorithms provide a winning outcome
3. Heuristic results in useful values for evaluating states
4. Forward checking results in appropriate moves, sorted by most advantageous first.
5. Ability to play with 2 input players, rather than against the AI.

4.3 A-range Goals

Ambitious goals for the project should go here.

1. Heuristic is admissible AND consistent.
2. Implementation of a graphical interface for the program. Displays the configuration of the board during the game.
3. Ability to save state of game to custom file type, and load later for save/load functionality.
4. AI defeats the average human player in games.

Timeline

Please delineate the major milestones of your project (no milestone should take more than a week to accomplish). The milestones should have accompanying descriptions of everything they entail.

Then, for each milestone, specify when you will have it completed (specific date). Additionally, make it clear which milestones you will have completed by each check-in date.

Acknowledgement

Instructor Mentor 1 _____ Signature _____

Instructor Mentor 2 _____ Signature _____

Team Member 1 _____ Signature _____

Team Member 2 _____ Signature _____

Team Member 3 _____ Signature _____