Widchard Faustin, wf77@drexel.edu, CS338, Project Stage #1

January 27th, 2019

Chinese Checkers: The Board Game of Family Relocation

Basic Description and Motive:

The goal of this app is to create a software application that will allow a user to play a comprehensive game of Chinese Checkers with up to five other players. Chinese Checkers is a popular game, but there aren't many ways to play it online, and the ones I researched were buggy and not as coherent for new players. On top of this fact, it only supported player versus CPU. I want to create a more comprehensive software that lets a user play with other users, and I want to have multiple people be able to play the game.

Users:

Although there aren't many Chinese Checkers enthusiasts, those users would not be the only people who would want to use this software. Those who love playing simple board games that includes strategy would also have a blast playing this game. This software could be played against another real-life user, so users can go toe-to-toe against another user. As a stretch goal, I also want users to be able to use this software even when others don't have the ability to play with them, so those who do like to play games against CPU opponents over real-life users would also be interested in this game. However, the main draw of this software is the fact that it isn't just a person versus person (PvP) board game, but it has the ability to be played in teams of 2, as well as accommodating up to 6 users in a single game, so people who would want to play a simple strategy board game with a group of people would also be interested in this game.

System:

The user will be able to do a lot of things in order to find both fun and success using this software, starting with the most important thing: playing the game. As I stated before, the Chinese Checkers software that I did encounter in my research, while sufficient on a functional level, were buggy, and the most problems I found in the software dealt with illegal movement of pieces. Users using the software that I plan to develop will be able to: move pieces to empty, adjacent spaces in any direction; jump over either their own or enemy pieces to an empty space adjacent to the nonmoving piece as long as the empty space, the non-moving piece's space, and the moving piece's space creates a straight line segment; and be able to continue jumping pieces using the second rule in the same turn. More info about the rules can be found in the link (https://www.ducksters.com/games/chinese_checkers.php). Additionally, users will be able to look up the rules of the game on the software, choose how many people are playing, choose the color that they will play as, and choose (if the game is an even number of players), if the game will be individual or team based. Finally, the stretch goal, would be to include the ability to face computer generated opponents, so users wouldn't necessarily have to play against human opponents to use the software.

Project Plan:

	Current Work	**Future Preparation**
Week 1	Make Preliminary Screen Designs; Plan Out Data Type Architectures; Start Designing Chinese Checkers Board and Pieces	Start Planning and Implementing Game Logic; Start Implementing Rough Draft of Main Menus; Research Best Strategies for Chinese Checkers to Program AI
Week 2	Continue Implementing Game Logic; Finalize The Majority of Screens I've Created For Week 1; Start Bug Testing Game Logic	Continue Bug Testing; Start Creating Any Art Needed For the Software; Start Programming AI Nature (stretch goal based on time)
Week 3	Finish Implementing Game Logic; Start and Continue Implementing Any Minor Screens That Have Not Been Finished; Finalize Any Minor Artwork Needed For the Software For Implementation	Continue Implementing Artwork; Create Basic Questionnaire For Users for Testing; Finish AI Implementation
Week 4	Finish Implementing Art; User Testing; Making Minor Adjustments As Needed	Final Tweaking

Data Structures and Technologies Needed:

In terms of technologies needed in order to make this software that are directly related to the coding process, I still need to do research on image generation and how that works in Java/Swing. However, I don't believe much else other than that will be needed. In terms of technologies not linked to the coding process, I do plan to use Photoshop to create the board and pieces, as well as design any menu items needed (if time permits).

In terms of data structures, I do need to create a two-dimensional array that can simulate the board. I believe that should be the most complicated data structure that is needed to be made. However, as I get into programming, I will continue defining more complex structures that need to be noted in this document.