CPSC 2150 Project 1

Connect 4

Michael Ellis, Cooper Taylor, Ryan Chen, Adam Niemczura

Requirements Analysis

Functional Requirements:

- As a player, I need to place a marker on the board, so that I can play the game.
- As a player, I should be able to choose which board I can play on, so that I can play the game on a variety of machines without limitation
- As a player, I should be able to pick another column if the column I picked was full, so that I can place a piece during my turn
- As a player, I should be able to pick a column again if the one I picked was invalid, so I can place a piece on the board during my turn.
- As a player, I need to be able to pick my piece character, so that I can identify where I am on the board
- As a player, I should be able to customize the size of the board, so that I can play on a variety of sizes with my friends.
- As a player, I should be able to vary the number of markers required to win, so that I can play as intended on customized boards.
- As a player, I should be able to play until I get my chosen number of markers in a row, so that I can win the game.
- As a player, I should be able to play until I get my chosen number of markers in a column, so that I can win the game.
- As a player, I should be able to play until I get my chosen number of markers diagonally, so the game ends in a win. As a player, I should be able to play the game until the board is full, so that the game ends in a tie.
- As a player, I should be able to take turns with my opponents, so that we can play

against each other.

- As a player, I should know whose turn it is currently, so that I can determine who needs to place a marker.
- As a player, I should be asked to play again, so that I can keep playing with my opponent after the current game ends.
- As a player, I need the board to be printed on the screen after each turn, so that I can see the game as it is being played.

Non-Functional Requirements

- The board should have a minimum of 3 rows and 3 columns
- The board should have a maximum of 100 rows and 100 columns
- There should be a maximum of 10 players
- There should be a minimum of 2 players
- A player must have at least 3 tokens in a row either horizontally, vertically, or diagonally to win
- A player must have a maximum of 25 tokens in a row either horizontally, vertically, or diagonally to win
- Gravity should affect the tokens
- The number of tokens in a column should not exceed the number of rows
- An error message should be returned if a player tries to place a token in an invalid column.
- The program should have 2 end conditions: win or tie
- The game will start with player 1
- A piece can be represented by a single character
- Two players cannot have the same piece character
- Players should alternate placing tokens
- The game should be able to repeat once the game has ended
- Empty positions on the board should be indicated by a single space character
- The program should run on Linux and Windows

- The program will be a command-line application, and run in the terminal
- The program needs to be written in Java
- There should be a memory efficient and speed efficient version of the game board

System Design – (UML diagrams)