CPSC 2150 Project 1

Checkers

Jada Hall, Elise James, Molly McKenzie, & Sarah Mendoza

**Requirements Analysis**

**Functional Requirements:**

1. As a player, I need to be prompted to choose a piece to move so that I can select where to move my token
2. As a player I need to be asked which direction I want to move my piece so that I can make sure my piece is moved in the correct direction
3. As a player, I can jump another players token as long as there is an empty space in that direction, so that I can advance in the game
4. As a player, I can have my token become kinged once it reaches the opposite side of the board, so that I can move that token in all 4 diagonal directions
5. As a player, I can see whose turn it is before each move, so that I can play the game fairly
6. As a player, I need the game to end when I or the other player have no tokens left on the board, so that I know someone has won the game
7. As a player, I need to know if I pick a token that cannot move anywhere, so that I can choose another token to move that turn.
8. As a player, I need to be notified if I select a space or a piece that belongs to the opponent to choose a valid piece to move
9. As a player, I need to be notified if I select a space that is empty so that I can make my move
10. As a player, I need the board to be printed after each turn, so that I know where all the tokens are.
11. As a player, I need to be asked if I would like to play again or terminate the program after the game, so that I can move on in the program
12. As a player, I can input a diagonal direction to move my piece in, so that I can take my turn and advance the game
13. As a player, if I enter an invalid input (e.g., a non-integer or an out-of-bounds coordinate), the program should notify me and prompt me to enter a valid input

**Non-Functional Requirements**

1. If a user inputs invalid input, the program will not crash
2. The program needs to alternate between two players for their turns
3. An 8x8 board will be generated at the beginning of the game for the players to play on
4. Player 1’s pieces will be represented by x and X when kinged
5. Player 1 should start on the top of the board
6. Player 2’s pieces will be represented by o and O when kinged
7. Player 2 should start on the bottom of the board
8. The “black tiles” on the board will each be represented by \*
9. Each player will start with 12 pieces
10. At the beginning of each turn, the status of the board should be printed to the terminal
11. Players can only make one jump per turn
12. The player will not be required to make a jump just because one is available
13. Player 1’s pieces should be initialized as “x”s
14. Player 2’s pieces should be initialized as “o”s
15. The game should be runnable with JDK17 and Junit 4
16. The game should handle exactly two players who alternate turns
17. The player should be made aware if they cannot move their piece to a certain location
18. The game should run on both Windows and Linux OSs

**System Design – (UML diagrams)**