CPSC2150-Checkers

Team Name: Pour Over Java
Team Member 1: Michael Funchess
Team Member 2: Will Lovin
Team Member 3: Curren Patel
Team Member 4: Jarret Wilson

Functional Requirements: As a <userRole>, I <what/need/can><goal> so that <reason>.

Functional Requirement User Stories:

1. As a player, I need to be able to specify my desired piece so that the program can give me options to move.
2. As a player, I need to be able to see the board every turn so that I can determine what move to make next.
3. As a player, I need to be able to input my move so that I can advance the board state.
4. As a player, when I run out of pieces the game should end and my opponent should win so that the game has a win condition.
5. As a player, when I eliminate all of my opponent’s pieces the game should end and I should win so that the game doesn’t run forever.
6. As a player, if I select a position that is out of bounds the system should alert me and prompt me to reselect a valid position so that I do not make an invalid move.
7. As a player, if I select a position that belongs to my opponent the system should alert me and prompt me to select a position that I can move to so that I do not erase my opponents pieces.
8. As a player, I need to be able to have my piece crowned whenever I reach the end of the board so that I am rewarded for playing well.
9. As a player, I need to be able to jump my opponent’s pieces so that I am rewarded for noticing a blank space behind my opponent’s piece.
10. As a player, I need to be able to choose to play again so that I can continue playing, if desired.
11. As a player, I need to be able to see the boundaries of the board so that I can tell how far out I can play my pieces.

12. As a player, I need to be able to know which pieces are kings or not so that I can tell which pieces are threatening my board state.
13. As a player, I need to be able to tell the row that my pieces are on so that I can locate my pieces accurately.
14. As a player, I should be given the option to jump pieces if possible, so that I can advance to the other side of the board more quickly.
15. As a player, If I have a kinged piece it should be able to move in all 4 cardinal directions so that I am rewarded for getting that piece to the other side of the board.
16. As a player, I should be able to choose between a speed efficient and a memory efficient version of Checkers so that I can choose whichever version would be more preferable for my current runtime environment.
17. As a player, I should be able to specify the board size so that I can play Checkers on a larger board size.
18. As a player, I should be able to specify what letter represents my pieces so that I can make it obvious that I am the one playing.

Non-Functional Requirements:

1. The program needs to be able to store and move information so that it can be presented to the players.
2. The program should be written in Java 17.
3. The game needs to be able to run on both Windows and Linux architectures.
4. The program should take input from and output to the console.
5. The program should be able to handle having two players with alternating input.
6. The program needs to have the ability to distinguish between which player is currently making a move.
7. The program should be able to handle the player inputting a string instead of an int.
8. The program should be able to handle players inputting board coordinates that are out of bounds.
9. The program should be able to determine which tiles are unable to be traversed on.
10. The program should distinguish between black and white tiles so that I can tell which tiles are playable or not.

11. The program should be able to generate a board size from 8x8 to 16x16 (only even numbers).
12. The program should always generate 2 empty rows between the two players.
13. The program should be able to differentiate between the two players regardless of what character they chose to represent their pieces.
14. The program should contain two versions of the game, a memory-efficient version and a speed-focused version.
15. The memory-efficient version of the game should generate the board as a HashMap.
16. The speed-focused version of the game should generate the board as a 2D array.