JavaFx Graphics

Checkers

Checkers is a strategy game played by 2 players on an 8 * 8 board. Each player begins the game with 12 soldiers on the closest three rows. The soldiers are placed with one square between each soldier. The aim of the game is to defeat the other player by taking all of their pieces. The pieces are constrained by either moving diagonally or jumping. Jumping is the medium through which you kill/take an enemy piece. When a soldier gets to the other side of the board he receives a field commission and is promoted. A King can move forwards and backwards and is a formidable agent to have on the game.

I originally began my project with the intent of making a game of chess. After a day or so it quickly dawned on me the mammoth task I had chosen so I changed my focus towards Checkers. This game although superficially simple turned out to be quite difficult. The actual game setup was not too hard, I forged ahead with setting the board up and placing pieces upon it. I separated out my boardLogic from my GUI. However, moving the pieces was incredibly hard and I got stuck on this for some time. My initial approaches were riddled with circular dependencies. This was between the BoardLogic and BoardGUI (Now called Checkers). I eventually succeeded in eliminating the circular dependency. This was done through using if statements. The Checker class calls the BoardLogic to check whether moves are valid. If the logic denies I.e. returns false, the piece is returned back to its original location. If the logic returns a true, this piece may move there. At all times there is a running parallelism between boardLogic position and the GUI.

Javafx was quite perplexing to be frank. The flow of information does not seem particularly ordered to me. It was confusing at first as we have been taught the dogma that programming happens from the bottom up. I tried to use while loops at the beginning to loop my programme around as we did in the noughts and crosses. This led to a number of runtime errors, so I had to back out of this way.

In future I will build on this framework laid down and try build a chess set. I believe I have broken the back of the hardest feature which to me was the movement. Chess would now be a matter of sorting each piece's type's logic and just assigning a different symbol to each piece as part of the class. Throughout the design cycle I tried to embrace the principles of agile development. I set myself loose targets throughout the design process and tried to keep to these. Once again however I get a little caught up in the details and have a tendency of scrapping everything and restarting. In all I am happy with this bit of work. I think it's a good attempt at using Javafx graphics and I would do it again. I hope you like it!

To run the programme lead with java Checkers.