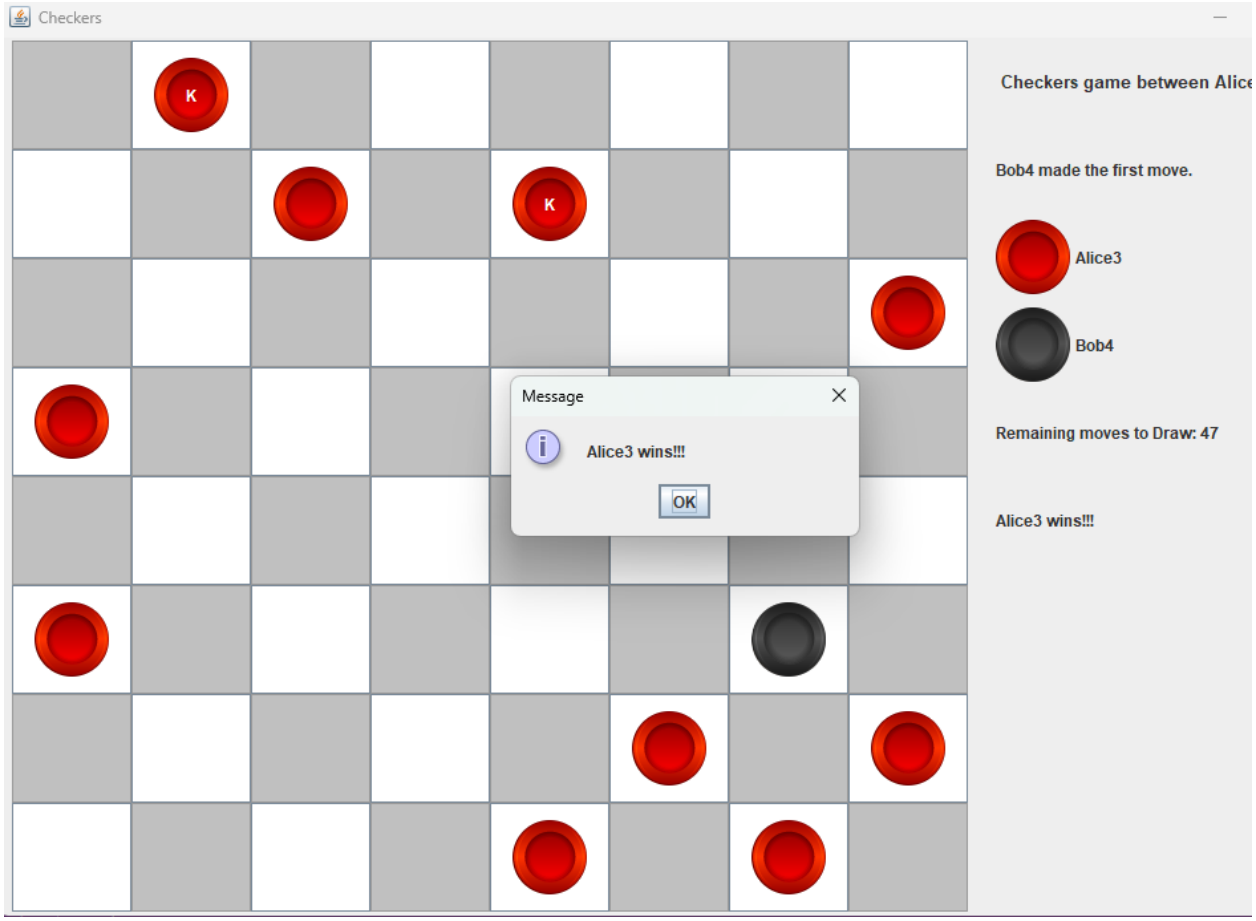
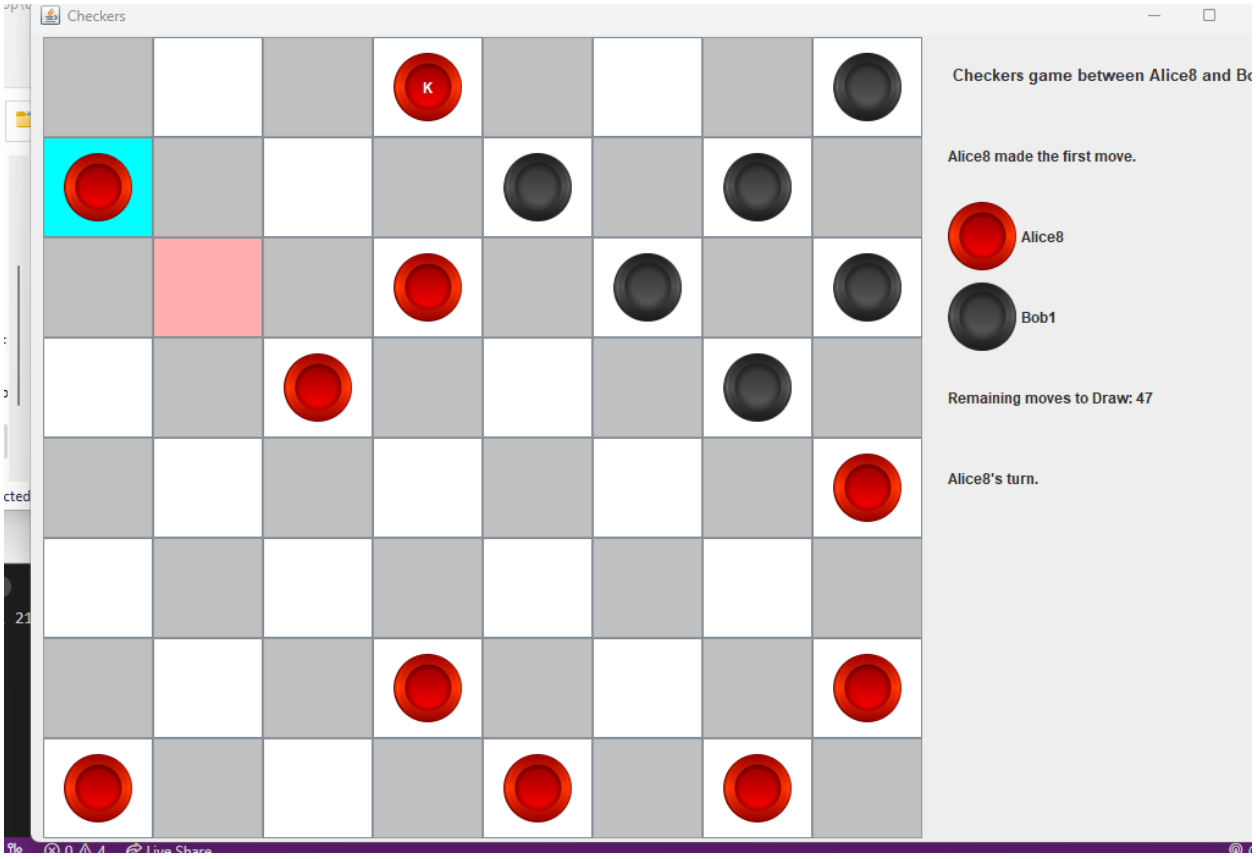
Elvis KImara
Borrowed the game and GUI from the internet. (https://github.com/KAI10/CheckersAI).
I added my own to it by making the Heuristics class, move class, and UseNumberOfPawnAndKingHeuristic and UsePositionOnBoardHeuristic.java. These last two are my evaluation functions. I made UsePawnsKingsMyPositionValHeuristic.java but the idea for this was borrowed from the internet.

3. I made a moveGenerator (line 84) in MinimaxCheckersAgent.java.

6. Compare the effect of increasing search depth (come up with a method to demonstrate your point).
I managed to make the computers verse each other and most of the time, the one with the higher depth won most of the time

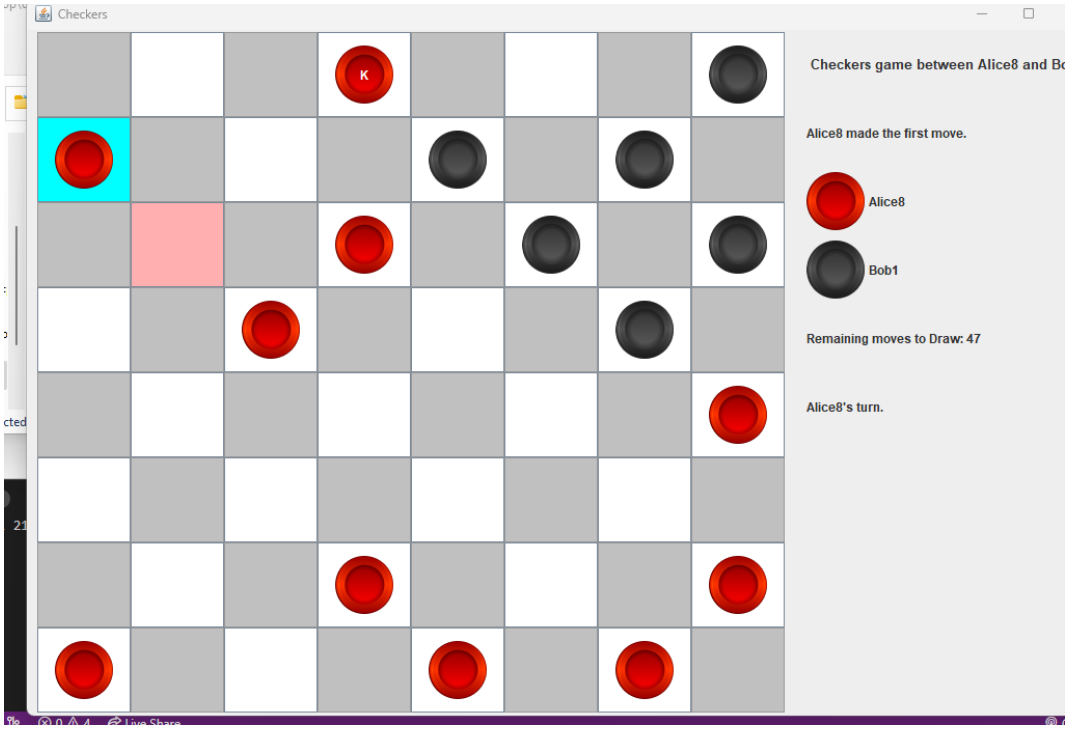


I recorded and tried it many times. Even with the positional heuristic were the agent is rewarded for moving up a board. The one with depth 7 destroyed the one with depth 1 as seen below



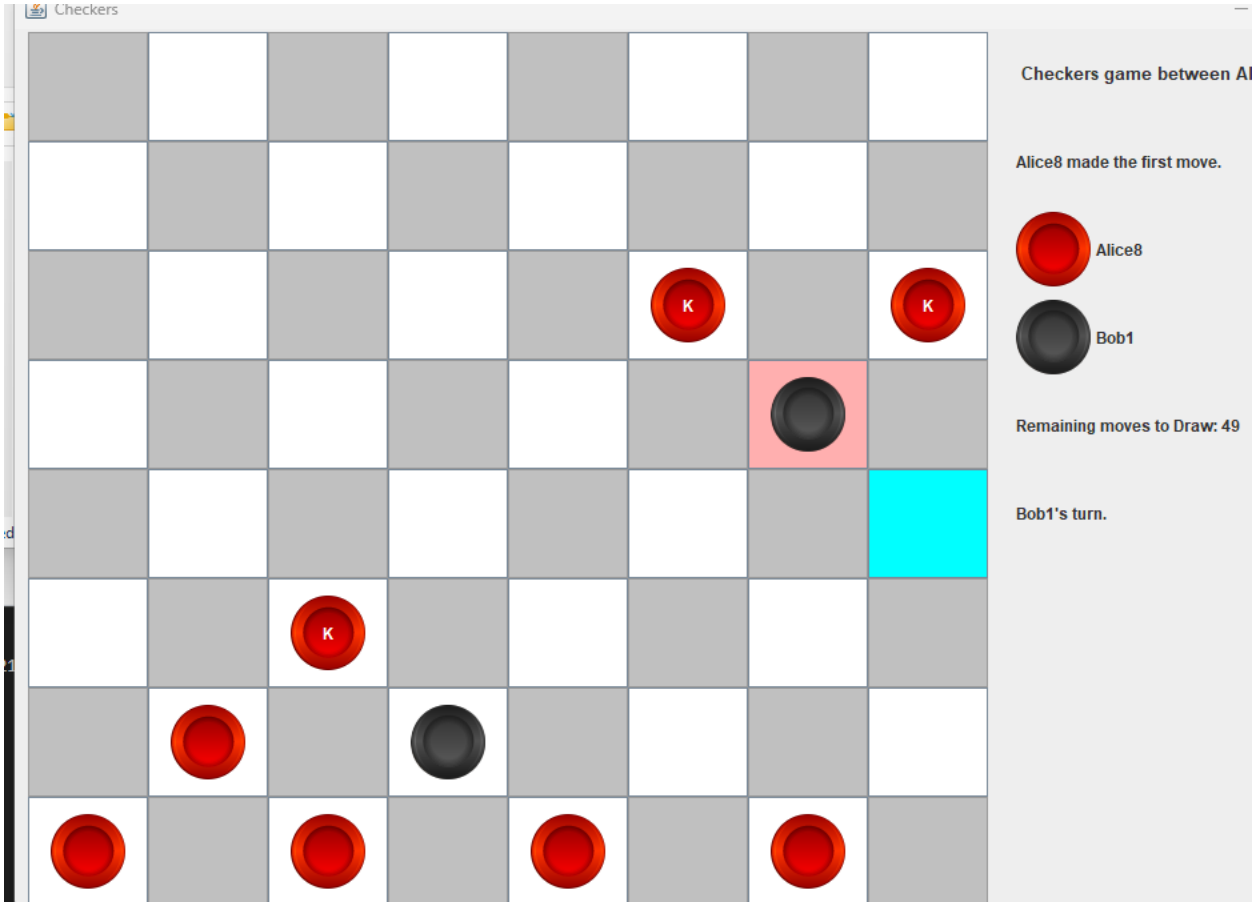
7. Implement at least two evaluation functions with vary quality. Compare the effect of improving the evaluation function

I made two evaluation functions of my own. One took its reward by moving up the board like this one:



It made kills but it prioritized moving up more than killing. So you can see the red moving more towards the left since theres empty space. In this case, the red has depth 8 vs black with depth 1.

The second one based on number of pawns and kings. It is way better than the first one since it prioritizes winning. In this case, the red has depth 8 vs black with depth 1.



As you see, by improving the function like I did in the second one, the black pieces were easily surrounded and the game ended quickly since the game proritized killing/winning instead of moving up the board.