CS-171 Checkers Final AI Report

Team name:

Member #1 (name and NetID): Mohamed Kharaev mkharaev
Member #2 (name and NetID or N/A): John Quinzon jquinzon

I. In about 1/2 page of text, describe what you did to make your Final AI agent "smart."

To make our Final AI agent smart, our team implemented a minimax algorithm along with a custom heuristic for the checkers board. This heuristic was based on different properties of the game that could be interpreted as giving a player an advantage in the game. We ranked these properties in importance, and created a points system based on these ranks. (Pawn Piece - 1000 points, King Piece - 2000 points, Rows away from king if pawn - up to 1000, Distance from an enemy if King Piece - up to 1000, Amount of pieces - 100 per piece, Randomization - 10). The randomization was implemented because there are many moves in the beginning stages of the game where multiple pieces get the same evaluation, so the randomization was to prevent the AI from being too algorithmic.

II. In about 1/4 page of text, describe problems you encountered and how you solved them.

One problem we had with the AI was when implementing minimax. Even after implementing iterative depth first search, our algorithm was taking too long to produce results. This was when we had to implement alpha-beta pruning for our minimax tree, which reduce the amount of time our algorithm took tremendously, and even allowed us to take our I-DFS algorithm further down levels.

III. In about 1/4 page of text, provide suggestions for improving this project.

Our team does not think there are any major improvements to be made to the project. It is structured well and teaches us how to implement "smart" AI. Possibly some logistical or technical improvements such as making the student manual a bit more specific or having different versions of the AI's working on time.