

Simon Yoon

Professor Sable

ECE469

My project implemented checkers with AI options. I compiled my code using Cygwin. You can choose which player goes first, cyan or red, but red will be placed at the bottom of the screen, while cyan will be at the top. The program allows for any permutation of human and AI competition and allows for the setting of the AI's maximum time allotted per move. The time taken is outputted, but a generous buffer of a fraction of a second is always there so as to not go over the time limit. The program is also capable of reading from the file format given on your website to access any particular game state. The game offers the option to save a game and quit.

My project implements minimax search with alpha-beta pruning. It numbers and lists all legal moves regardless of type of player and turn. A heuristic is also implemented in order to provide some strategic backbone. The project used lowercase and uppercase O to represent pawns and kinging a piece will be displayed by the piece turning into a K. The heuristic takes into account the desire to king, keep friendly pawns guarded and surrounded, spacing from enemy pawns, and centering of pieces. And given a necessary take, the AI does the move automatically.

I used ANSI escape codes to color and format the game and this should be functional for most terminals. I used it using this reference as aid to my discretion.¹

¹<https://unix.stackexchange.com/questions/548158/in-2019-is-it-safe-to-assume-that-all-terminals-underst-and-ansi-escape-codes>