Marko Smiljanic

CSc 180 – Intelligent Systems

Spring 2019

Project 2 – Adversarial Search

a) Programs Name

- My program's name is Regulator, as he appears to be attempting to keep the piece. Although he does actually seek victory.

b) Instructions on Installation and Use

- To run Regulator, save Regulator.cpp to any directory on Athena that you can access. From there run '`-g++ Regulator.cpp -o Regulator`' and the enter 'Regulator' to begin the game. From there, Regulator will ask if you would like to play first or not. Enter '`Y`' or '`N`' to decide Yes or No. Enter in all movements as a sequence of '`A1B1`' to ensure they work properly.

c) Code Language

- Regulator is written in C++.

d) Techniques Used

- For this first iteration of Regulator, I have only included the minimum requirements; that is, minimax and alpha-beta pruning. For the final competition, I have plans to include iterative deepening and scoring tables to increase speed and refine decision making.

e) Typical Search Depth

- Regulator will print out the number of nodes he has visited and will also print the average plies deep that he is searching. This is usually around 3.7 at the beginning of the game and around 3.4 near the end of the game.

f) Terminal Evaluation Function

- For this version, the Evaluation function is very simple. The program iterates through all spaces and counts human pieces as negative based on their value, 5000 for kings, 2000 for bishops, 1000 for knights, and 250 for pawns. Computer pieces are counted positive of the same values. I will be readjusting these points and adding tables that make pieces have different values based on location in my competition version.

g) Program Strength

- I will be honest and say that Regulator is not strong at all. He will move towards winning the game, but he makes silly decisions like sacrificing kings and powerful pieces when it is not necessary. I have not beat him, but I am also not good at chess at all.

h) Anything Unusual

- I cannot point out anything unusual with my code.

i)

- I am not aware of any bugs that make him crash or perform illegal moves. I am fairly certain I have tested for the majority of those and prevented them from happening.

j)

- There are no bugs that make him perform weak moves, he only has weak evaluation that make him perform weak moves.