Alan Huang

Dr. Daisy Tang

CS 4200

CS 4200 Project 3 Report

Approach:

After reading the project task requirement file I found out that there is an example tic tac toe C++ code to use as our project's skeleton. I check the code and found out that the example code didn't come with alpha-beta pruning, evaluation function, cut-off test which are the requirements that we needed to make for the project. I only know how to use C# and Java, and I would love to try to use C++ to learn this language but the deadline for this project is short; I decide to use Java to implement this project to make sure I can finish it on time.

I first transfer the simple C++ code to java and start adding alpha-beta pruning into the program. Again the textbook pseudocode is very confused to read, and I have to read a few articles on the internet to correctly adding the alpha-beta pruning. The reason I knew that I successfully implement alpha-beta pruning is that I noticed that the AI executing time decreased. Without alpha-beta pruning, my program would take forever to run the depth of 5, but

with alpha-beta, it was able to run the depth of 5.

Second I start to add a cut-off, so I can make sure that my program won't run past 5 seconds. There are many useful ideas I found on the internet to implement this requirement. Also when the cut-off is activated the program will start using the evaluation function to calculate the best move. All the tasks so far are very simple to do, and that is why I left the evaluation function at the very end to do because I felt it wouldn't be that simple.

I noticed that the AI is smart only with depth setting is large, and with lower depth, the AI movement doesn't make too much sense. From the internet, most of the website only talks about tic tac toe and connect 4 evaluation function, but connect 4 is only allowed the player to drop chess from the top to the bottom which is different than our project.

I found a simple method for tic tac toe which taught me to use + 100 for each 3 in a line for AI + 10 fir each 2 in a line and +1 for each 1 in a line, but after testing my AI for this 8x8 board tic tac toe still perform a very bad move. After a few days of trial, I decided why not just teach the AI to block a player's move? I give the AI a few examples moves such as XOO and OOX to let the algorithm knew that place a move next to the player will gain move score. I also wrote XO and OX to block the player's move as soon as possible since

this is a connect 4 tie tac toe. My AI’s move finally starts making sense after I implement my evaluation function to block players' move as a priority, but sometimes the AI is acting a little weird.

Finding:

I think from this project the hardest part is to implement evaluation function. A well-designed evaluation function can make the AI win all the time, but since there are lots of probability so it’s difficult to make a bulletproof algorithm for the evaluation function. I think if I can have a little more time for this project I can probably improve my evaluation function and redesign the data structure to improve efficiency. I barely finished this project 1 day before the deadline, but this project helped me understand how to design a simple AI program.