

Xiaowen (Angel) Zhang

Prof. Sable

ECE 469 AI-IS

11/13/2016

Othello Project Report

Objective

The purpose of this project was to create a game-playing program, which plays Othello against the user, using the alpha-beta search and iterative deepening.

Game Features

This program supports three game modes: a human play against a human player, a human player against a computer, and a computer against itself. At the beginning, a game lobby will provide a game mode selection along with other available options including who plays first, how long would like the computer player to think within a limit between three to sixty seconds, and where you want to start the game on the board. As you play, you will be given available moves for your turn, and you will only be able to click on the legal moves. If you are out of move, the system will automatically switch to the other player, or you could click on the “pass” button, and this button is only available when there is no legal move. However, when you have legal moves to choose from, you can take your time. Also, you will be able to see your opponent’s legal moves, but your clicking on the board would not affect the game. On the upper left corner, there is a scrollable textbox, which records each moves that have been made so far, and if there is a computer player, its search information—time and depth will so be recorded. Next to this text box, there is a score board indicating the current number of pieces on the board for each player. The game ends automatically if no one can move, and you will receive a message indicating the winner. Anytime you want to start over you can select “restart” button.

Program Details

This program consists of eight classes: Lobby and Board classes are used for game display as well as using a timer to handling the switching between players; game, move, and node classes are tracking the information about game status, moves that have been made, and potential moves; Player, as the super class, along with ai and human classes maintain the information about the players, and for the ai class, it makes the game winning strategy.

To accomplish computer player part, I used minimax search algorithm to find out the optimal

move, and used alpha-beta pruning to enable the computer to search in a larger depth within given time limit. I also implemented Othello game strategies including using point system or random moves with a purpose of testing in early stage. I looked up possible heuristic functions on line, and I decided to an evaluation function combining mobility, position, and current scores in different weights. Also, considering the distribution of the pieces on the board might be different from early game to late game, I modified the heuristic function in terms of the weights for the features and the points that I assigned to each square. For example, when there are 20 – 40 pieces on the board in total, which defined as mid game, I valued the corner and the favorable edge positions even more so as punished the unfavorable positions. Here is the points table in the upper left corner in the early game:

[50]	[-1]	[5]	[2]
[-1]	[-10]	[1]	[1]
[5]	[1]	[1]	[1]
[2]	[1]	[1]	[0]

At last, I enjoyed working on this assignment, and I truly learned a lot, not only the AI part, but also the game itself as I never really played this game before ☺