Hillel Weintraub

11/3/13

**AI Project#1**

I chose to implement Othello in C++ using Cygwin for my development environment. I have provided a makefile to compile the code and produce an executable called Othello.exe. My implementation uses an ASCII interface and legal moves are automatically marked on the board. I have provided the option to read in a position from a file, and a sample file named “b” is included in the folder. The file must contain the current players turn on the first line (1 or 2), and each subsequent line contains 8 space separated characters to represent the board state. Zero is for an empty location, one is for a location occupied by player1, and two is for a location occupied by player2.

The two classes I implemented for this program are the Board class, implemented in Board.cpp, and the Player class, implemented in Player.cpp. The Board class represents the state of the game, and implements the logic necessary to traverse the state tree, namely finding the legal moves, and applying a move to transition to the next state. The state is represented with a 2d 8x8 char array for the board, a 2d bool array for the legal moves, and a char for whose turn it is. The Board class also implements the heuristic function, as well as a method to display the board. The Player class stores the state of the player, whether it is player one or two and whether it is a computer or not. It also implements the alphabeta search. Finally, the main program which contains the overarching game framework is implemented in Othello.cpp.

I use several features in my heuristic function. First I use the piece count of both players. I also statically assigned values for the different positions on the board and take these into account. I also compute the corner occupation, and look at the locations next to unoccupied corners. Finally, I look at the actual and potential mobility of both players, where potential mobility just corresponds to the number of pieces next to open squares. I weight these features differently giving the most weight to corner occupation and the least weight to number of pieces. However, if the state is an end of game state then only the number of pieces is used, and weighted so as to dominate non end of game states.