Artificial Intelligence Programming Exercise

Participants

Pablo Panero (Student id: 2442996)
Satish Kumar Rajagopal (Student id: 2365620)
Raúl Jímenez Redondo (Student id: 244162)

Algorithm

Our algorithm (Jarvis) implements Reversi game using iterative deepening search with no depth limit. Therefore, the only constraint is the given time, meaning that it will keep looking for a better solution according to the heuristics.
Alpha-Beta pruning is used to cut off branches of the tree according to Wikipedia pseudocode[1]. However due to the usage of different parameters, values, and making it in a single function instead of two (one for max, and another for min) it has been slightly changed.

Heuristics

Coin placement

Concerning the heuristics, a two dimensional matrix contains weight values for each position in the game board. Weights are assigned according to strategically importance of the position in Reversi game[2]. The main idea is that borders weight more than the rest of the board. The corner is the most important position. For the same reason squares that are in the second line (from the limits) of the board are considered worse positions, because the opponent could take borders or corners, and therefore have lower weights. However, if is a corner is available as a movement, it is taken without further checking because of its high value.

10000	-3000	1000	800	800	1000	-3000	10000
-3000	-5000	-450	-500	-500	-450	-5000	-3000
1000	-450	30	10	10	30	-450	1000
8000	-500	10	50	50	10	-500	800
8000	-500	10	50	50	10	-500	800
1000	-450	30	10	10	30	-450	1000
-3000	-5000	-450	-500	-500	-450	-5000	-3000
10000	-3000	1000	800	800	1000	-3000	10000

[1] http://en.wikipedia.org/wiki/Alpha%E2%80%93beta_pruning#Pseudocode
[2] http://othellomaster.com/OM/Report/HTML/report.html#SECTION00063000000000000000

Coin parity

In addition the amount of coins of each player is taken into account[3]. The difference between our amount of coins and the opponents' one is divided by the total amount of coins in the board. This value is multiplied by 1000 in order for it to have effect along with the matrix values.

$$1000 * \frac{Player\ coins - Opponent\ coins}{Player\ coins + Opponent\ coins}$$

Mobility

In order to avoid running out of moves and therefore losing its mobility value is taken into account. If a movement leads to 0 moves for us, it is discarded. If not, both ours and the opponents' amount of moves are taken into account. Like coin parity, this value is multiplied by 200 in order for it to have effect along with the matrix values.

$$200 * \frac{Player\ moves - Opponent\ moves}{Player\ moves + Opponent\ moves}$$

[3] https://kartikkukreja.wordpress.com/2013/03/30/heuristic-function-for-reversiothello/