



Háskólinn í Reykjavík

Reykjavik University

Spring 2015

Artificial Intelligence

T-622-ARTI

Teacher: Stephan Schiffel

Adversarial Search

Programming Assignment 2

:: NerðirMeðSkapgerðir ::

Ægir Már Jónsson - aegir13@ru.is

Arnar Freyr Bjarnason - arnarb12@ru.is

Atli Sævar Guðmundsson - atlisg12@ru.is

Stefanís Bergljót Stefánsdóttir - stefania13@ru.is

Dags: 27. jan. 2015

Heuristic description

The evaluation function looks at all the rows, columns and diagonals of length 4 (lets call them quads) and checks how close the players are to getting all 4 spaces. If the quad is empty or there are both white and red discs in it, the rating given is 0. Otherwise, 1 disc gives 1 point, 2 discs give 15 points and 3 discs give 40 points. The player's points are positive but the opponent's are negative. We then sum up all the ratings of the quads. The function uses the rating 500 as a starting point and returns the sum of 500 and the sum of the ratings of the quads.

Experiments

In our AlphaBetaSearch function, when faced with two equally good options, the agent picked at random. This means that our agents are not deterministic so the games between the agents did not always go the same way.

Results

Time	# of tests	White	Red
1 sec.	10	9,2	14,0
5 sec.	10	12,2	18,4
10 sec.	5	10,2	14,0

Table 1. Our agent against the random player. Shows the average number of steps needed to win.

Time	# of tests	White	Red
1 sec.	12	30,2*	35,3
5 sec.	6	33,5	31,0
10 sec.	4	35,0	34,0

Table 2. Our evaluation function against the old evaluation function. Shows the average number of steps needed to win.

* In one case, the agent using the new evaluation function and the new evaluation function tied.

At the start of the game, our agent could search about 8 moves ahead when the playclock was 1 second and about 10 moves ahead with 5 or 10 seconds.

Conclusions

We pitted Nilli against himself with different ratings in the `rate()` function. We mainly experimented with changing two ratings, that is, how many points to return if we find 2 of our discs in a quad and how many points for 3. We ended up using 15 for 2 discs and 40 for 3 discs, and of course 1 point for 1 disc.

Our agent won almost all of the games against the agent with the original evaluation function ($\text{<number of white discs that are adjacent to another white disc>} - \text{<number of red discs that are adjacent to another red disc>}$), so it is clearly a big improvement. The differences between 1, 5 and 10 second playclocks did not have a significant impact on the results, but this makes sense since the branching factor is about 7, so the time spent on searching each level is exponential, so going from 1 to 5 to 10 seconds may not increase the depth of the search a lot.

When playing against the random player, Nilli was not better when he had more time to think about his moves, which was expected, because his heuristics assumes he is playing against a player that carefully chooses his moves as well. So, when playing against a random player, it does not help Nilli to think many moves ahead, instead he would be better off with a super simple strategy, like always placing his disc on top of his own disc, unless he cannot, then he places his disc on the floor, until he wins.

We are so proud of our little Nilli, not only has he learned to vacuum a room until it is clean in an efficient way, but also learned to play Connect 4 in an efficient and vicious way. Not only did he win the old heuristics almost always (only one tie) but none of us were able to beat him no matter how hard we tried.

