Homework for Artificial Intelligence for Robotics Assignment 10

Bastian Lang

June 8, 2015

1 Practical Part

1.1 Task

Implement a simple *Connect 4* game to demonstrate the use of adversarial search for deterministic, fully observable, two player turn-taking zero-sum games.
For the adversarial search problem, implement a minmax and alpha-beta pruning agent that allows you to play against the computer and/or the two algorithms against each other. Compare these two approaches based on their search time, space requirement and other information that you think is important. What was your evaluation function to rate the board?

1.2 Approach

1.2.1 Game

In the beginning the user is asked to decide the two players. He has the opportunity to choose between a human player, the **MinMax** agent or the **Alpha-Beta-Pruning** agent.
My implementation uses a *GameValidator* - class to check the current state of a board. This validator checks if the board is full and if in a single line, row or diagonal there are four connected stones of a kind.
The main part of the game runs in a while loop until the state has changed to either *draw* or one of the two players won. In every cycle one of the two players is asked for his next move. The board gets checked afterwards and the current player changes.

1.2.2 MinMax

Because the search space is way too big to check it completely, I limited the depth of the search to 4 levels.
In the beginning I used an **evaluation function** that only checks terminal states, but with the depth limit and returning 0 if the limit is reached this will result in poor behaviour.
I thought about several strategies. One is to check the number of connected stones each row, diagonal and column and assign them a value depending on empty fields on both sides of the chain, on only one side or no empty fields left. But due to time issues and problems while implementing I instead went for a simpler solution.
In every row, diagonal and column I now check each possible set of neighbouring four stones. I evaluate each of those sets with respect to the number of missing stones to have four connected stones. I then compute the sum over all those sets and subtract the same for the opponents stones.

1.3 Result

I did not manage to implement the alpha-beta pruning agent, so I cannot compare both agents. But I expect the latter one to consume less space ;-)