Team: Ronald Jenkins, Caelan Mayberry, Joshua Walton

Game: Khet

Intro/Description of problem:

The game we have chosen to program an AI for is the laser game Khet. Khet is a lot like chess. It is a grid with all pieces known and each player is given the task of defeating the enemy pharaoh. The difference between Khet and Chess is that instead of moving pieces to take out the opponents pharaoh, each player has a ‘sphinx’ which fires a laser at the end of each his turn. Both player’s pieces on the board are either mirrors to reflect the laser or blocks to stop he laser. Mirrored pieces change the direction of the beam and are kept in play, but if a piece is hit on a non-mirrored side, then it is removed. Certain pieces have mirrors on both sides and are irremovable from the game, which makes it impossible for the game to end because neither side has any pieces left. The goal of Khet is to hit your opponent’s pharaoh with your laser while also protecting yours from getting hit. Our goal is to program both the game to be playable by humans through simple console commands and by our own programmed agents.

Solutions we found and their merits:

            Because Khet is a relatively new game compared with games like chess or Go, it does not have much critical research devoted specifically to it. We were, however, able to find one master’s thesis which looks deeply into both minimax/alpha-beta pruning solutions and Monte Carlo solutions for Khet. [1] In order to have a sufficiently broad take on these algorithms and their advantages and disadvantages, we looked more closely at other examples of their employment. We found one paper which describes a variation on a Scouting algorithm called Negascout. [2] We took notice of this paper because it claims that its algorithm explores 20 to 30 percent fewer nodes than does alpha-beta pruning. We also found a simple paper describing the use of Monte-Carlo tree search in a game context. [3]

          We currently use our own minimax/alpha-beta pruning. Below we have the results of our AI against other players. Due to memory constraints the AI’s have to be limited to a short horizon, Khet is on a similar level of complexity to Chess and as such is not currently possible to for a complete search of the game tree. [1]

Our Solutions:

|  |  |
| --- | --- |
| Matches:5 | Win/Loss |
| Horizon: 1 | ? |
| Horizon: 2 | ? |
| Horizon: 3 | ? |

Currently we do not have any data, as it is still being generated.

Conclusions:

Hopefully that with the agent working it will show improvement as its horizon increases. We expect that with only a horizon of 1 it will most likely lose its games. With a horizon of 2 I expect better chances of success but still not great. However with a horizon of 3 I expect it will start a decent chance of success.

Outline:

Intro/Description of problem

Solutions we found and their merits

Our solutions

How it performed

Conclusion

Citations

[1] Using Intelligent Search Techniques to Play the Game Khet.

<https://project.dke.maastrichtuniversity.nl/games/files/msc/pmthesis.pdf>

[2] An improvement to the scout tree search algorithm.pdf. (n.d.).

<http://www.top-5000.nl/ps/An%20improvement%20to%20the%20scout%20tree%20search%20algorithm.pdf>

[3] Chaslot, G., Bakkes, S., Szita, I., & Spronck, P. (2006). Monte-Carlo Tree Search : A New Framework for Game AI, 216–217.

<http://www.aaai.org/Papers/AIIDE/2008/AIIDE08-036.pdf>