CITS3001 PROJECT

Introduction

The 2018 CITS3200 assignment requires us to research and implement relevant algorithms/methods to create two agents that can play the Hanabi card game. This report will document the design choices made for the two agents, including research, implementation and results.

The game

Hanabi is a cooperative card game where players work as pyrotechnicians and aim to hastily create fireworks for an upcoming festival. The game is played with coloured cards ranging from numbers 1-5 with colours of white, red, blue, yellow and green. The goal is to build 5 fireworks, where each firework is a set of each colour in correct numerical order (1,2,3,4,5). The catch of this game is that no one can look at their own cards, only everyone else’s.

A player has a choice of three options a turn:

* Give a clue
* Discard a card
* Play a card

Give a clue

If a player decides to give a clue to another player, it costs a blue counter (of which there is 8 initially). There are two types of clues allowed:

* A colour clue
  + Where you can identify the cards that have a certain colour
* A value clue
  + Where you can identify the cards that have a certain value

When giving a clue all the information is given, i.e. if a player has two white cards and another player gives a hint about the first player’s white cards, then both must be identified and not just one of them.

Discard a card

If a player decides to discard a card then a blue counter is gained, a card from their hand is put in the discard pile face up and a new unknown card is added to their hand. This action cannot be undertaken if there are already 8 blue counters available.

Play a card

If a player decides to play a card, they place it face up in front of them. If the card can successfully be added to one of the fireworks (is the number next in line for the stack), then it can be placed on that stack or the card is discarded, and a red counter is gained.

The Score

The score of the game is calculated by adding all the values of the top cards on each of the firework stacks. Using the ‘International Federation of Master Pyrotechnicians’ scale, you can then work out your Quality of Display.

Research

Programming language

Two languages were considered for project, being Java and Python.

Potential Implementation Algorithms/Techniques

We know that Hanabi is an imperfect information game (due to not knowing one’s own cards) and with our version, also time limited to one second a turn. When researching possible implementations, these conditions were taken into consideration.

Monte Carlo Tree Based Search 1 & 5

The Monte Carlo Tree Search algorithm loops 4 steps over a given time frame to select the next best move for a finite set of moves. These steps are selection, expansion, simulation and backpropagation.

Starting from root node R, where each node is a game state (with R being the current game state), the algorithm chooses a node from the child nodes of R and expands until a leaf node is reached (an unexpanded node).When selecting, the algorithm is more likely to select the most successful child node, allowing for a better analysis of the favourable option. If the selected node is not going to bring the game to an end, expand the node and create the child nodes. Select one of these nodes and play random moves until an end game is reached and the outcome is either a win, loss or draw. This technique is also known as a playout. Use this outcome to update the parent nodes “win tally” from the leaf node back to the root node and repeat the cycle.

An advantage of this algorithm is that it will always have a result. This is advantageous for the agent because you can cap the tree search to one second to meet the game conditions of a turn. Another advantage is that it is domain dependent. This means it does not require an evaluation function since it instead uses the number of “victory paths” for each child of the root node to value each option.

While this algorithm can be very efficient for some games, I can also be quite slow for others. Games with a larger number of options will create larger trees that require more time to search. This can create a suboptimal result that takes much longer to reach.

Van Den Bergh Steps (Rule Based Play) 1

As outlined in [1] “The Van Den Bergh Steps agent is the best rule-based agent” for Hanabi. It relies on the

Design choices

* Programming language
* AI considerations
* Algorithm considerations
* Implementation
* Results

Bibliography

1

<http://teaching.csse.uwa.edu.au/units/CITS3001/project/2018/papers/MCTSHanabi.pdf>

2

http://teaching.csse.uwa.edu.au/units/CITS3001/project/2018/papers/HanabiNP.pdf

3

http://teaching.csse.uwa.edu.au/units/CITS3001/project/2018/papers/hanabi-strategy.pdf

4

http://teaching.csse.uwa.edu.au/units/CITS3001/project/2017/paper1.pdf

5

https://www.youtube.com/watch?v=Fbs4lnGLS8M