

Coursework 3 CST3140 – Pandemic Game with AI Agent

Rodrigo Ferreira
Middlesex University
London, UK
RL604@live.mdx.ac.uk

ABSTRACT

Implementation of Pandemic Game with an AI Agent player and assistant chatbot.

INTRODUCTION

For this coursework it was asked to deliver an implementation of the Pandemic Game in Java. Such implementation should include an AI agent to assist the human player(s) and chatbot interaction.

GAME IMPLEMENTATION

For this implementation a board for 2 players, a human and the AI Agent of 48 cities corresponding to 48 player cards drawn 2 at a time at the end of each turn by each player at the end of their round, each player round has 4 moves. There are four coloured diseases, blue, yellow, red and black that can be cured by trading 5 same colour cards to build a research center in a same colour city and trading another same colour cards (there are 12 of each colour available on the deck for a total of 48 cards plus 5 epidemic cards, which spread 3 waves of 3 cube diseases to 3 pseudorandom cities from the infection deck, which itself constantly contains the 48 cities at any time. The players draw 4 cards each from a shuffled deck containing only the cities, this deck is then split into 5 piles where an epidemic card is inserted, put back together and shuffled. When the game begins 9 cities are picked pseudorandomly to distribute 3, 2 and 1 cubes of that cities colour to it. There are 24 cubes for each disease colour, when more than three cubes are assigned to a city nearby cities are contaminated with 1 same colour cube independently of that adjacent city colour. When a group of 24 cubes runs out the players lose. Other ways of losing include running out of cards on the player deck or hitting the 8 on the outbreak mark, which increases by 1 for each epidemic card drawn and 1 by each city outbreak when it hits more than 3 cubes in a single city. Available moves from classic rules include move to an adjacent city on the board, trade a card of the current located city to travel anywhere, trade a card of a city to travel there, travel to a research center, build research center, find a cure by trading 5 same colour cards, trading a card with another player and pass. In the digital version as we are using simple TUI the player can do the following moves for free, list actions, list stations, list city colour, list cities, list infections, list adjacent cities, list players hands and locations and communicate with the AI Agent.

AI AGENT

The AI Agent uses a simple greedy algorithm to move around pseudorandomly while trying to find infected

adjacent cities to move to and remove cubes. It can be instructed through chatbot TUI communication to move somewhere, remove cube from a location and trade cards with the human player.

REFLECTIONS

An algorithm such as Monte Carlo Search Tree as used by DeepMind to beat Garry Kasparov at chess would probably be a more effective regardless of the pseudorandomness. This algorithm essentially selects a tree node, expands it, simulates the game and updates the nodes in the tree to find an optimal solution. This is a reinforcement learning algorithm so it needs a reward function for optimal solutions, such as winning the game. Although slower to the greedy implementation due to the complexity of the search space, it implements tree searching inside tree searching thus it increasing exponentially even though this could be increased with good policies and giving more importance to some nodes and allowing their children to be searched first, it would be able to simulate and predict to a good degree of accuracy, as with enough simulations the pseudorandomness could be mapped, the best outcome moves to win the game, play them and advise the human player on what to do (as the human would be statistically inferior to it).

CONCLUSION

This project served as a great opportunity to learn more about AI algorithms for pseudorandom adversarial search and search spaces, software programming by reading and understanding instruction for a game and implementing it in Java and chatbots through processing text input and deferring intent for the output response.

REFERENCES

1. Artificial Intelligence, A Modern Approach – Stuart Russel and Peter Norvic
2. <https://cwa.mdx.ac.uk/cst3170/cst3170.html> – Chris Huyck, CST3170 Middlesex University London
3. <https://www.deepmind.com/publications/monte-carlo-tree-search-as-regularized-policy-optimization> - Yunhao Tang , Jean-bastien Grill, Florent Altché, Thomas Hubert, Remi Munos, Michal Valko, Ioannis Antonoglou
4. <https://www.ultraboardgames.com/pandemic/game-rules.php>

