CPSC 490 Project Proposal

Daniel Kim
Advisor: Prof. James Glenn

February 1, 2023

1 Game Description

Settlers of Catan is a three or four-player turn-based stochastic/Markov game with limited information that pits players against each other to gather resources and build a series of outposts on the fictional landmass of Catan.

Catan is represented as a series of randomly-generated interlocking hexagons; each hexagon is associated with one of five differing resources (wood, brick, sheep, stone, wheat), and a number from 2-12 (excluding 7). Players can place settlements on the vertices of these hexagons to gather resources; on a player's turn, they roll two standard six-sided dice, add up the total, and then all players gather resources for every settlement-hexagon connection that they have created. Afterwards, the player can choose to spend these resources on a number of various improvements to their colony, including:

- Roads, which are placed on the edges of a hexagon, and are required to be connected to a player's existing roads or settlements/cities. New settlements must be placed next to a player's existing roads, and a given hexagon's edge can only contain one player's road, making road-building (especially early in the game) essential for expanding one's colony and achieving victory. Roads are cheap, costing only one wood and brick, counterbalanced by the large number of roads players will inevitably build over the course of the game.
- Additional settlements, which cost one wood, brick, sheep, and wheat. Settlements must be placed at least two hexagon edges away from all other settlements as well as the road connection restriction mentioned above, restricting the number of potential settlement spots available to players and adding strategic complexity to the game. Players can build up to five of these.
- Cities, which are upgrades to settlements, doubling the resources per city-hexagon connection that players can gather. They cost three stone and two wheat, and replace an already-existing settlement on the game board. Players can build up to four of these.
- Development Cards, which can contain a variety of useful effects, such as stealing all of one particular resource from every other player or allowing the player to instantly place two roads without paying the normal cost for them. They cost one sheep, wheat, and stone.

If players roll a 7, instead of gathering resources, the robber is activated. The player who rolled a 7 can place the robber on any hex in the game; afterwards, that player can choose another player with a settlement/city connected to that hex and steal one resource from their hand at random, after which the building phase continues as normal. The "soldier" development card, which can be activated on one's turn, also has the same effect of the robber.

The aim of the game is to earn a total of 10 Victory Points ("VPs"); settlements are worth 1 VP, cities 2 VPs, having the longest road (at least a length of 5) 2 VPs, and having the largest army (made up of soldiers gained from drawing development cards, with at least a size of 3) 2 VPs. Players are initially given two settlements and a road for each of them, and play does not stop until a player achieves the 10 VP goalpost.

Note that the full game includes a couple more mechanics, including discarding cards upon rolling a 7, trading resources with the bank at a heavily inefficient rate, docks on the outskirts of the map for mitigating that inefficiency, and trading resources with other players, which can be done on one's turn after rolling the dice and resolving their outcome. However, given the potential complexity of including those mechanics into my computational intelligence, we will exclude those for now and include them as possible stretch goals in case development proceeds faster than anticipated.

2 Project Design

When considering creating an agent for Catan, there are a couple things of note. Given that Catan is potentially infinite (for example, if players continuously roll 7s or numbers not connected to any player's city or settlement), attempting to create a complete tree of the game is infeasible. Regardless of that fact, the branching factor and the depth of a given game's possible moves is sufficiently large that creating a full minimax tree would take a enourmous amount of time anyway, so we must choose a method of creating computational intelligence that can work with a limited tree depth while also producing workable results in the context of the game.

The random nature of the game must also be considered; any intelligence must utilize an algorithm that takes that randomness into consideration; successfully gauging the likelihood of gathering a particular resource on a hexagon, as not all hexagons are of equal value, is essentially to the creation of a sufficiently powerful agent. Additionally, not all resources have equal value, and the value of said resources constantly shifts over the course of the game. For example, wood and brick are essential in the early game when the creation of roads is important for reaching good settlement spots before one's opponent, while stone is not as useful. However, once many settlements have been created, stone and wheat become paramount in order to upgrade settlements into cities.

Of course, given the popularity of Catan, I am not the first person to analyze this game in the context of computational intelligence, not even at Yale; Omar Ashraf and Christopher Kim's Senior Project in 2018 came up with the idea of creating a computational intelligence for Catan independently of my research proposal, where they utilized deep Q-learning to effectively build an agent that could play Catan. A small number of other researchers outside of Yale have also made Catan agents, each with differing methods; Szita, Chaslot, and Spronck used MCTS when the algorithm was still relatively new (2010), Pfeiffer opted for a reinforcement learning-based approach (2004), and Guhe and Lascarides created a symbolic model with various heuristics to effectively improve an agent's prior distribution for analyzing game states and next moves. (2014).

Catan is an interesting case for this type of algorithm because very little is technically hidden from view of the players; all resource gathering is fully visible to everyone. In theory, a player (or computer) with a perfectly eidetic memory could keep track of every resource that players gather, with the only source of variability being which cards players randomly steal from each other upon activating the robber. While that source of variability necessitates a model that doesn't contain perfect information, and human players do not typically retain this level of information (except, perhaps, at the very highest levels of play) this is still a characteristic of note in the case of Catan.

In order to tread new ground concerning this particular game, I am proposing utilizing a version of MCTS called Information Set MCTS (ICMCTS) with Partially Observable Moves (POMs) first pioneered in the context of games by Cowling, Powley, and Whitehouse, and suited for use in stochastic games with imperfect information such as Catan (2012). ICMCTS differs from the standard well-trodden MCTS algorithms by dealing with information sets as the primary nodes in the minimax tree, which resolves many of the weaknesses of using a purely determinization-based approach. Regular MCTS will likely be used as a sort of benchmark, with ICMCTS being used as a comparison.

As for how MCTS will be set up, I plan to utilize a series of meta-actions in order to guide the actions of my agents - instead of looking at specific actions (build a road here), I plan to instead set general guidelines for my agent (use roads to expand, focus on cities, get development cards, etc.) as the potential branches for my MCTS algorithm, which the agents will follow accordingly.

3 Deliverables

- Settlers of Catan engine that can successfully simulate games of Catan.
- Baseline agent that uses standard, determinized MCTS to play Catan.
- Agent using ISMCTS to play Catan, to compare with above.
- Final report.

Note that all code is planned to be delivered to the CPSC 490 directory, unless directed otherwise.

4 Timeline

- by Feb 14: Settlers of Catan engine complete.
- by Mar 7: Baseline agent complete.
- by Apr 21: ICMCTS agent complete.
- by May 4: Final report complete.

References

[1] Omar Ashraf and Christopher Kim. *CPSC 490 Project Proposal: Computational Intelligence for Settlers of Catan.* PhD thesis, 2018.

[2] Peter I. Cowling, Edward J. Powley, and Daniel Whitehouse. Information set monte carlo tree search. *IEEE Transactions on Computational Intelligentce and AI in Games*, 4(2), June 2012.

[3] Markus Guhe and Alex Lascarides. Game strategies for the settlers of catan. *2014 IEEE Conference on Computational Intelligence and Games*, 2014.

[4] Michael Pfeiffer. Reinforcement learning of strategies for settlers of catan. *International Conference on Computer Games: Artificial Intelligence, Design and Education 2004*, 2004.

[5] Istvan Szita, Guillaume Chaslot, and Pieter Spronck. Monte-carlo tree search in settlers of catan. *Advances in Computer Games*, 2009.