CPSC 490 Project Proposal: Computational Intelligence for

*Settlers of Catan*

Omar Ashraf and Christopher Kim
Advisor: James Glenn

Introduction and Game Mechanics

This work seeks to address two main goals: 1) develop a mobile application for the board game, *Settlers of Catan* (henceforth referenced to as *Catan*), that allows humans players to play against other human players and artificial intelligence agents, and 2) apply computational intelligence concepts to design and implement an artificial intelligence agent to competitively play *Catan*.

*Catan* is a stochastic board game played between 2-4 players on a board of hexagonal tiles. Each tile is assigned a particular resource (brick, wood, sheep, wheat, or ore) as well as a numeric value between 2 and 12 (inclusive). The configuration of these tiles is randomly generated and subject to several constraints to ensure fairness. The goal of the game is to accumulate 10 “victory points” which can be earned through the following ways:

1. **Settlements** - Settlements are placed on the corners of hexagonal tiles and is the primary means of acquiring resources. On each player’s turn, two dice are rolled and if a player has a settlement adjacent to a tile that contains the number that was rolled, he or she receives a resource card that matches the resource of the tile. Players place two settlements in the initial stage of the game and can build up to 5 settlements by trading in specific resource cards. Each settlement counts as one victory point.

2. **Cities** - Cities are upgraded settlements and can be built by trading in specific resource cards. If a player has a city adjacent to a tile that contains the number that was rolled, he or she receives two resource cards of the corresponding resource. Each city counts as two victory points.

3. **Longest Road** - Settlements and cities are connected to one another by roads. If a player builds 5 connecting roads, he or she attains *Longest Road* which corresponds to two victory points. Unlike settlements and cities, the owner of *Longest Road* can change throughout the game as players build additional roads.

4. **Largest Army** - Resource cards can also be used to buy development cards. There are 3 main types of development cards: ability cards, which allow players to use one of three abilities; knight cards; and victory point cards. Knight cards allow players to steal resources from other players. When a player activates 3 knight cards, he or she attains *Largest Army* which
corresponds to two victory points. Similar to Longest Road, the owner of Largest Army can change throughout the game as players activate additional knight cards.

5. **Victory Point Card** - The final category of development cards is the victory point card. When a player draws a victory point card, he or she is awarded a single victory point.

**Project Details**

This project consists of two main parts. Part I will involve the development of an Android-based mobile application that will allow human players to play against one another and against AI agents. This will be divided into two main components: the front-end Graphical User Interface and the game framework (decoupled from the application that will allow us to both run automated simulations to test AI performance and manually play the game). Part II will involve the design and implementation of an AI agent to competitively play Catan. Our current plan is to utilize a hierarchical deep reinforcement learning approach that combines deep Q learning with a priori heuristic-based knowledge to develop an agent that can rival a human’s performance. By hierarchical, we distinguish between lower-level strategies (such as build-a-settlement, build-a-city, build-a-road, or buy-a-development-card) and higher-level strategies (such as roads-and-settlements, cities-and-development, or port-opening). We will use heuristics to map from lower-level strategies to higher-level strategies and employ deep reinforcement learning to learn lower-level strategies. We build on Pfeiffer’s work, Reinforcement Learning of Strategies for Settlers of Catan (Pfeiffer, 2003), which uses a similar hierarchical approach but uses model trees instead of Q learning for learning state-action values.

**Division of Work and Deliverables**

I will first work on developing the general framework for the game with Christopher Kim. This will not only involve implementing the basic game rules but also creating a random benchmark agent that always randomly picks an action at each step. We anticipate that the implementation of the game framework will be complex enough for two people to work on simultaneously. It is important that this framework is in place before other components of the project are started since it will serve as the foundation for both the mobile application and the training, testing, and validation of the AI.

Once finished, I will move on to reviewing previous work that has been done by researchers for Catan as well as similar board games employing learning techniques that we can generalize for this specific project. I anticipate DeepMind’s AlphaGo (2016) and Tesauro’s TD-gammon (1995) to be particularly helpful in the brainstorming and evaluation of different model architectures that can be trained and tested through self-play.
After deciding on an initial architecture, I will work on the implementation of the model, while Chris works on developing effective heuristics to bridge the two tiers associated with the hierarchical Reinforcement Learning approach. Finally, we expect training and testing to be a time-intensive process, one that will require us to go back and tweak our architecture to improve model performance.

**Stretch Goals**

If the deliverables defined above have been completed before the deadline, we want to address two major stretch goals. The first is to utilize some other computational intelligence technique to compare and/or improve upon the performance of the AI agent. An example of such a technique is Monte-Carlo Tree Search which has been applied to similar stochastic games. The second stretch goal is to build a trading component into the gameplay of the mobile application. We have purposely omitted trading for the sake of simplification, as the inclusion of trading would significantly increase the size of the action space, making it difficult to train our model.

**Timeline**

09/20/2018 - Submission of Project Proposal  
09/21/2018 - 10/13/2018 (~3 wks) - Implementation of Game Framework  
10/14/2018 - 10/27/2018 (~2 wks) - Reinforcement Learning Literature Review  
10/28-2018 - 11/10/2018 (~2 wks) - Implementation of AI Model  
11/11/2018 - 12/04/2018 (~3.5 wks) - Training, Testing, and Refinement of Model  