Settlers of Catan

Ankeet Mutha, Brooks Willis, Emily Guthrie, Josh Langowitz

This semester we will be creating a software implementation of the board game Settlers of Catan. The game Settlers of Catan is a popular multiplayer civilization game for 2-6 people. Each player represents a group of settlers who are racing to expand and develop using the resources at their disposal. Each player has a turn in which they roll dice, collect resources based on the value of the roll, and buy materials to aid in their expansion. They can build new things onto their civilization during their turn and also can engage in trades with other players for desired resources. We'd like to create a means of playing this game without the need for the physical board and pieces, and ideally without needing all the players in the same room.

At a minimum our version of the game will require background python code to establish rules, game play, and interaction between players and a graphical interface using programs like Pixi or SVG.js to visualize the python code. We envision that the board displayed graphically would show the different tiles' values and resources, as well as where players had built on the board. Resource hands of the players would be displayed around the board to help facilitate trading. This would all be handled through the the GUI. The python code would orchestrate the rotation of turns, the rolling of the dice, the collection of the resources, the movement of the robber, and the execution of trades. Additionally, if things go horribly wrong, we can choose not to implement the rules and instead rely on the players to make sure nobody cheats, but we do not expect that to be necessary.

Assuming completion of the minimum goals we have for the game, we have several additional features we would like to add. First of all, the game has 3 expansions, so we could implement support for the new rules added in those. Secondly, we would like to provide a web server to play the game online, ideally with multiple lobbies to allow more than one game to be played at once. We would additionally like to make a mobile web application that will allow each player to view their hand of cards individually on their phone. The idea is that either a group of players will get together and display the board on a computer or projector, using phones for their hands, or each player will have the board and his/her hand on a computer, allowing the game to be played online at a distance. This would also involve creating a more involved trading system that allows several trades to occur at the same time or a queue to keep track of the order in which trade requests are made. Finally, we would like to keep track of statistics about the game, such as what resources players get and use during the game on average, and how that compares between winning and losing players.

The largest problems that we currently envision in the first stage of our project are making a GUI and networking, since we have not learned either of these skills yet. After this, we will need to learn how to write web applications for mobile devices. We all have some programming experience prior to this class so we should be able to figure out the things we need to learn. Another problem we face is the balance between creating a project that includes all the features that would make it most playable and working within the scope of our current and class acquired knowledge. While the web interface would be extremely useful, it also forces us to learn material not taught to us in Software Design. We'll have to balance our time between learning the new material and making sure we focus enough on demonstrating we've learned the material from the class.

Our first step will be to create a plan for the overall structure of our system. We start here because this project will involve many subsystems that have individual requirements for functionality. We need to be able to identify the block of the program that will handle each subsection's action and define means to write, test, and debug them. It will also aid in our ability to divide up responsibilities in a way that won't make block integration challenging later. Following this we will begin scoping some of our larger goals because of the time required for their completion. If they seem feasible, we will begin to learn the tools necessary to complete them. After a rough schematic of the system is completed we'll create a code structure that mimics our intended structure.