CS 3110 Final Report

Max Whitton, Leo Davies, Sasha Boguraev

December 2023

1 Vision

Our vision for this project is threefold, and has evolved since its conception and since previous milestones. The first goal for this application and the one that inspired its creation is to make a working implementation of a popular board game to allow players to enjoy the strategy and competition of the world of Catan without the physical board, pieces and cards. This makes the game faster, more portable, and more accessible. The second goal, which came to be through the later iterations of the application, is to automate some of the more tedious aspects of the physical board game, such as resource collection. These aspects can often lead to errors even with experienced players. With the help of our application, experienced players can play the board game but use our work as a tool to roll the dice and keep track of when they should receive resources. Lastly, Settlers of Catan can be a very hard game to learn. Seeing this, we implemented many checks to verify that all of the rules of the game are followed. With this functionality, beginner players can use our application as a learning tool to understand where and when they can build roads and settlements before transitioning to the full board. Whether our application improves access to Settlers of Catan, automates the mundane parts for experienced players, or helps beginners learn the rules, our hope is that our application can spread the world of Catan.

2 Summary of Progress

For the second milestone, we implemented a data structure that represented the Catan map as a graph, with nodes representing building locations, and edges representing potential roads connecting buildings. This data structure supported building, querying for the status of nodes and edges. At this time, the data structure had minimal code in place to enforce the rules of the game. We also implemented a data structure that represented players, and supported resource management and querying for and setting attributes. Both of these data structures were tested lightly using an OUnit test suite. Lastly, we began an executable file to welcome the players to Catan and help them set up the board.

For the third and final milestone, having most of the infrastructure in place

to begin implementing the executable game, we set our sights on implementing gameplay using our existing data structures. This required implementing a recursive loop to represent a single turn, implementing user-initiated building and trading, and implementing logic to support ending the game. For each turn, we implemented rolling a random die, collecting resources, and an optional building/trading phase. For building, we combined board updates and player updates that we'd written for the second milestone. To end the game, we added logic to monitor either player winning, or either player asking to stop. In addition to implementing the executable game, we also tested every possible category of rule violation and incorrect gameplay on our previous work, which required adding functionality to the earlier work to check the game's rules. These rules consisted of both restrictions on players and restrictions on the board. To do this, we introduced a rigorous OUnit test suite and used a tool called bisect to measure and improve upon our coverage of the rules we added.

3 Activity Breakdown

3.1 Max

- Utilized bisect to analyze and improve OUnit testing coverage
- Implemented board logic to check for valid road locations, settlement locations, and city locations
- Modified the board's data structure to support traversal in order to expand
- Tested placing each type of piece when it was illegal given the current state of the board
- Tested placing each type of piece when it was illegal given the player
- Implemented query functions for all attributes of players and boards
- Worked on the text presented to players when executing the game and added functionality to cancel an initiated build or trade
- Hours spent working (on this milestone) TODO

3.2 Leo

- Added logic to handle bad inputs
- Added functionality to let players indicate what actions they would like to take on each turn
- Implemented road and city building

- Implemented trading through pair programming with Sasha
- Wrote instruction manual
- Added turn functionality
- Hours spent working (on this milestone) 25

3.3 Sasha

- Implemented dice rolling
- Implemented turns
- Implemented resource collection during turns
- Implemented logic to increase points counter of players
- Implemented logic to detect whe players have won and ending game
- Debugging minor bugs
- Implemented trading through pair programming with Leo
- Wrote up README
- Fixed formatting for terminal print out
- Hours spent working (on this milestone) 25

4 Productivity Analysis

Generally, we found that we were less productive in terms of functionality than we'd projected, given the inherent complication of a graph-based game with so many rules. However, we felt that our productivity in terms of correctness and writing code to handle all types of bad input was very good and only improved as we worked through intermediate iterations of the application. We ended up implementing a smaller board and a less complex scoring system than we'd set out to, simplifying the game greatly, but every simplification we made traded off to great advantages in terms of automation, rule checking, and input handling. In the end, defining new goals to stay productive through each sprint shifted our scope slightly from making an extremely complex and strategical game to making a foolproof tool to aid the Catan community.