Senior Design Settlers of Stepán Proposal

1 Introduction

The Settlers of Catan is a beloved board game that is known among a large portion of the student population. In this game, players gather and trade resources to build a colony on an island, competing to become the greatest civilization on the board. As a game with a variety of simple rules and a customizable board, it is a good candidate to recreate in a senior design project.

2 Problem Description

Although Settlers of Catan is iconic, it has a few notable pitfalls that make playing it difficult. These problems include:

- Using the advanced setup is not friendly to newer players as one needs to have a basic understanding of the rules in order to select good locations for their starting settlements.
- It is common for players to forget when one of their numbers is rolled, resulting in them missing out on resources
- Rules mistakes are not uncommon, such as accidentally placing two settlements too close to each other, which can be difficult to undo if not spotted immediately
- There are a lot of pieces that can easily be lost, which can make the game more difficult to play

3 Proposed Solution

Our solution involves creating an automated board that facilitates most game functions. Here are some general blocks for the project:

- Include an option upon setup for "quick start". There is a "beginners' setup" intended to simplify the start of the game for new players, this initial map is often forgotten.
- Die rolling system
- Automatic resource distribution/management (includes "Longest Road" and "Largest Army")
- Board display (i.e. tile resources, player improvements, ports)
- Board consolidated to few pieces
- Easier construction of roads, settlements, and cities
- Display development cards
- "Failsafes" in controls to prevent breaking of rules

4 Demonstrated Features

Primary features that solve the original problems:

- "Quick start" allows players to easily set up the beginning scenario for new players
- Automated resource distribution helps players worry less about forgetting their numbers when rolled
- "Failsafes" prevent accidental rules violations
- The integrated interface eliminates the need to use small pieces, which prevents them from being lost.

5 Available Technologies

The primary component requirements for this project must cover:

- Player input
- Individual resource display
- Contents of variable resource tiles
- Locations of settlements and roads
- Internal processing of the rules and board state
- The game board itself

It should be noted that the technologies that follow tend to use common stock components that have a negligible cost. The primary costs are anticipated to come from the processor and the game board itself, which should be possible to acquire within a $500 budget.

Player Input

The players need to be able to input their actions. This may be accomplished through simple push buttons.

Resource Display

There must be a way to represent the amount of each resource that the players own. A set of LED lights or an LCD number display may be used to portray the amount of each resource, and a static inscription on the game board could be used to show which resource each display represents.

Contents of Variable Resource Spaces

The three most important parts of a hexagonal space are:

- The type of resource produced
- The number on which the tile produces resources
- Whether the robber is on the tile

The type of resource produced may be represented by putting shapes with a backlight on each hexagon and lighting up whichever one the hexagon produces during the game, like the engine status lights on cars. The number can be shown by an LCD number display, and the robber can be represented either by an additional light or use of the LCD display, as players will not be using a tile's number while the robber is present.

Additionally, some amount of budget may need to be allocated to ICs that will distribute the control or pins needed for individual outputs. That is, a very large number of LEDs will need to be individually controlled and may exceed the number of pins available on a single microcontroller, necessitating some form of a master-servant system.

Locations of Settlements and Roads

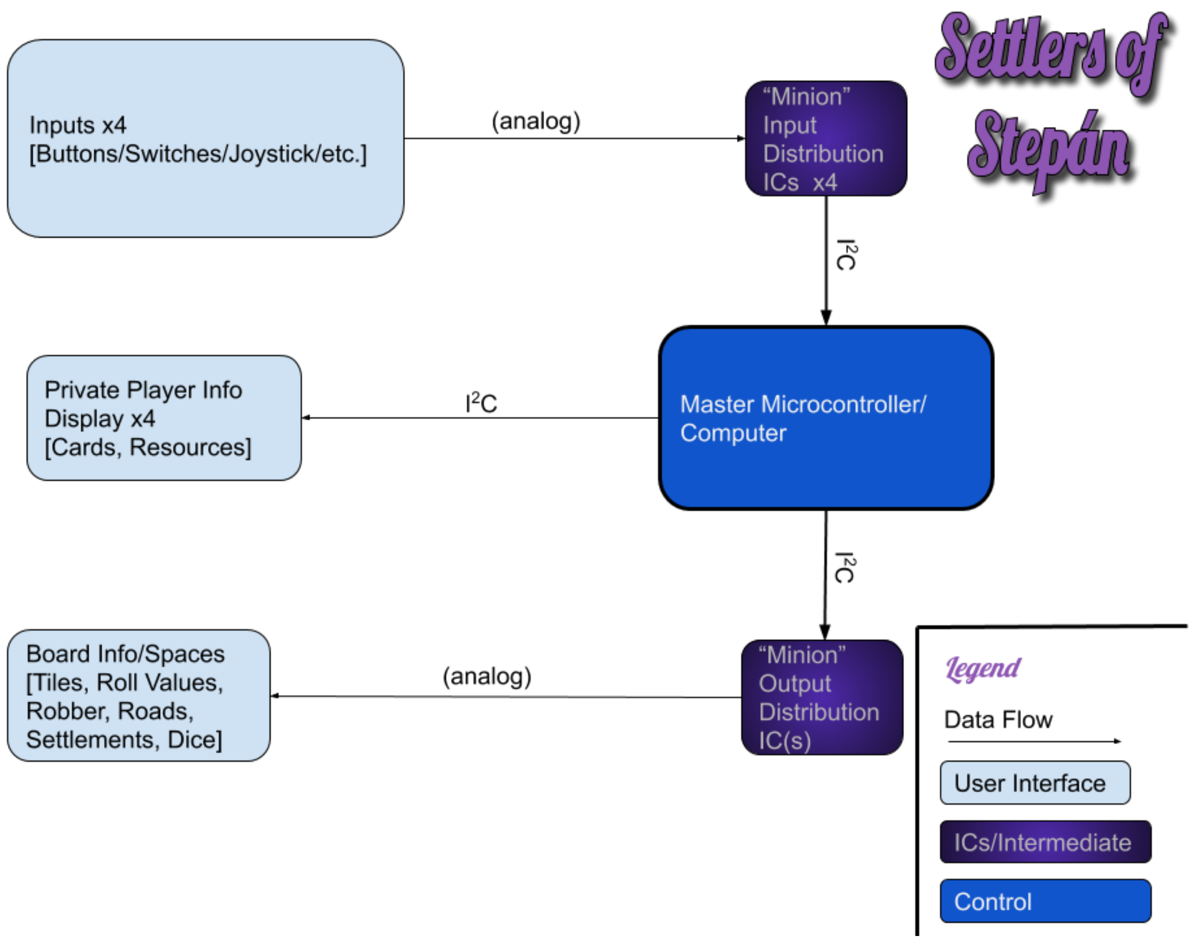
There are three structures that players can build and own on the board itself: roads, settlements, and cities. Roads exist on the edges between hexagons and are the only structure that can be built along the edges, so color controlled LEDs may be used along these edges to light up with the appropriate color for the player that owns a road along that edge. Likewise, LEDs may be used at the corners/junctions between hexagons to represent settlements and cities. Since cities are improved settlements, a secondary LED may be used around each junction to represent if the settlement has been turned into a city.

Internal Processing of Game State

There must be a software component to track the attributes and pieces of the board and the stage of the game, such as whose turn it is. This software component can be held on a number of processors, including the PIC32, an Arduino, and Raspberry Pi.

6 Engineering Content

The following design blocks use the technologies and be engineered as outlined below, which should include an appropriate amount of content.



"Quick Start" Option

This could simply involve some form of memory storage or matrix that contains a default board state that could be used in lieu of a random setup. This would also require some form of display or pair of start buttons to choose between the random and quick start setup.

This block is a minor feature that would not require much development after the core blocks are created.

Die Rolling

A pseudo random number generator may be created to simulate die rolls.

This would be a minor feature that would not require much development as software options for pseudo random number generators already exist.

Resource Distribution and Management

The resource distribution and management could be considered to be the storage of "cards in hand." Some software will be needed to store and manage these variables and integrate a hardware display to show what is "in hand" for each player. There are multiple options that would need to be discussed for the display itself and would require some testing and prototyping.

The management would be tied to die rolling and draw upon the memory for the board state itself, requiring an interface between the two blocks that would need to be developed concurrently.

This block would require a moderate amount of development and design

Board Display

This block involves the "actual game board" which would involve all of the resource hexagons and roads, settlements, and cities on the board. This would require a significant amount of design since there are many levels and methods of distributing this display between various master-servant IC interactions. This would also involve connecting a large number of LEDs and 7-segment number displays on the hardware side to show the board. There would also be a fair amount of software to generate and store the board state for processing.

Construction of Roads, Settlements, and Cities

The construction of improvements would be integrated with the board display. The most significant difference is that the board display block focuses more on features that remain stationary throughout the game, such as the resource types on each hexagon or the number assigned to each space. Improvements are dynamic and would need to be changed throughout the game as they are built. This would require giving the processor(s) access to the software and hardware components that store and display the status of improvements.

Improvement construction also requires player input, interfacing with that block, and would require significant software and development of a state machine that allows players to select the locations of different improvements within the confines of the rules. There would also be some minor hardware considerations to allow the display to have a state that shows a "tentative" location for an improvement when its location is being selected.

Overall, this design block also requires a very significant amount of design and development because it integrates software, hardware, player input, display output, and crosses over multiple states within a state machine.

Display Development Cards

The development cards would need to be denoted as a resource that players have access to. The software requires a moderate amount of effort because, although storing which types of development cards players own is not difficult, there are five different types of development cards with their own rules implications and actions. This would also require a significant amount of design and moderate amount of development on the hardware side as there are many ways for players to see which types of development cards they have, and there are not many clear ways to communicate necessary information (such as what each type of card does) while also fitting within limited physical space.

This would require a significant amount of hardware and software design as this is one of the most complex parts of the game rules and requires an elegant solution for playability.

Trading Resources

The players must be capable of trading resources amongst each other and with "the bank" in order for the game to work properly. This would require a moderate amount of software development as there would need to be multiple game states used to generate, offer, and approve of trades. There would also be a fair amount of hardware involved as players need to be able to use physical controls to initiate a trade, select the number of resources traded, who they would trade with, and whether they approve of the trade.

This would require a moderate amount of design and development because, although this would require a significant amount of player investment and it may be difficult to create an intuitive interface, the mechanics and display should not be much more difficult than displaying the complex development cards or selecting locations to build roads, settlements, and cities.

7 Conclusions

In conclusion, the proposed project is an appropriate senior design project. There are tangible problems with the current physical board game that can be resolved by an electronic game board. This proposal shows that an electronic game board with some automated capabilities can be broken into manageable blocks and presents an appropriate level of design challenge for the class.