# CATAN®

Owen Hillary, Sam Rios, Owen Westerkamp

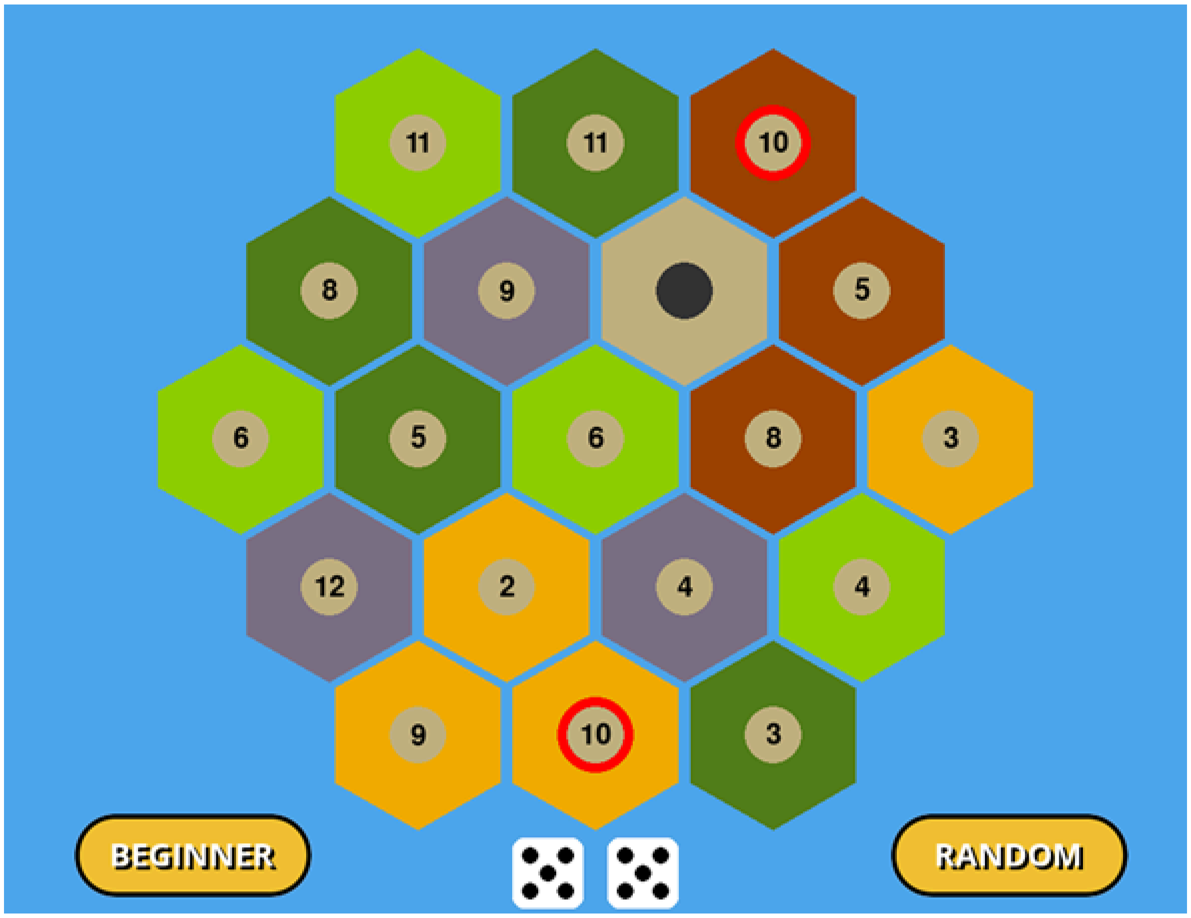
GitHub: https://github.com/arctucDangkla/Catan

## Project Objective:

This project aims to recreate the popular board game *Catan* (also known as *The Settlers of Catan*) as a digital, multiplayer game using Python. The game will replicate the core mechanics of the original board game, including resource management, tile placement, trading, and building settlements, cities, and roads. The project will focus on creating a user-friendly interface, implementing the game rules, and enabling multiplayer functionality.

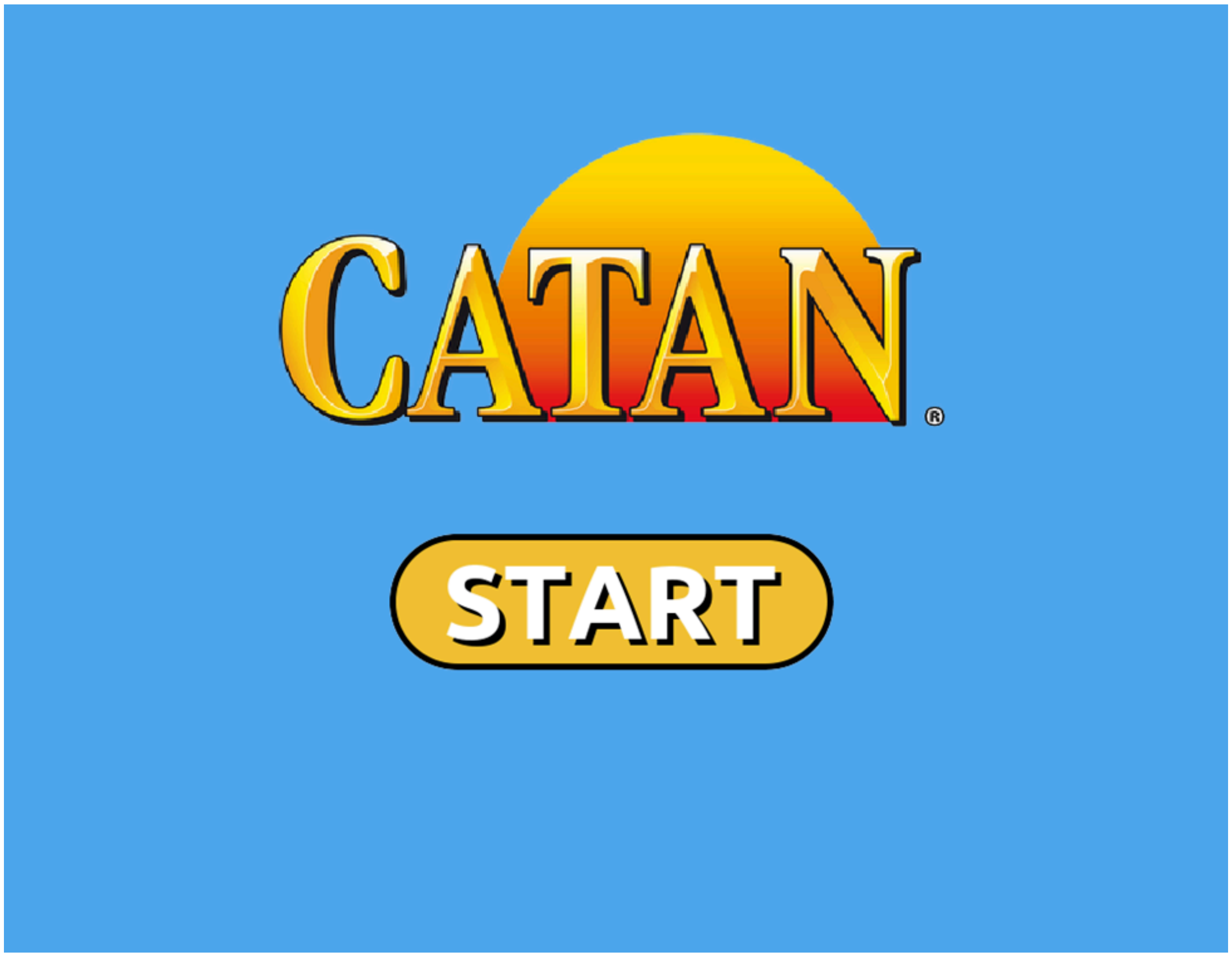
### Functionalities

After our first delivery, where we aimed to flesh out the game board and make it presentable, our board looked like this

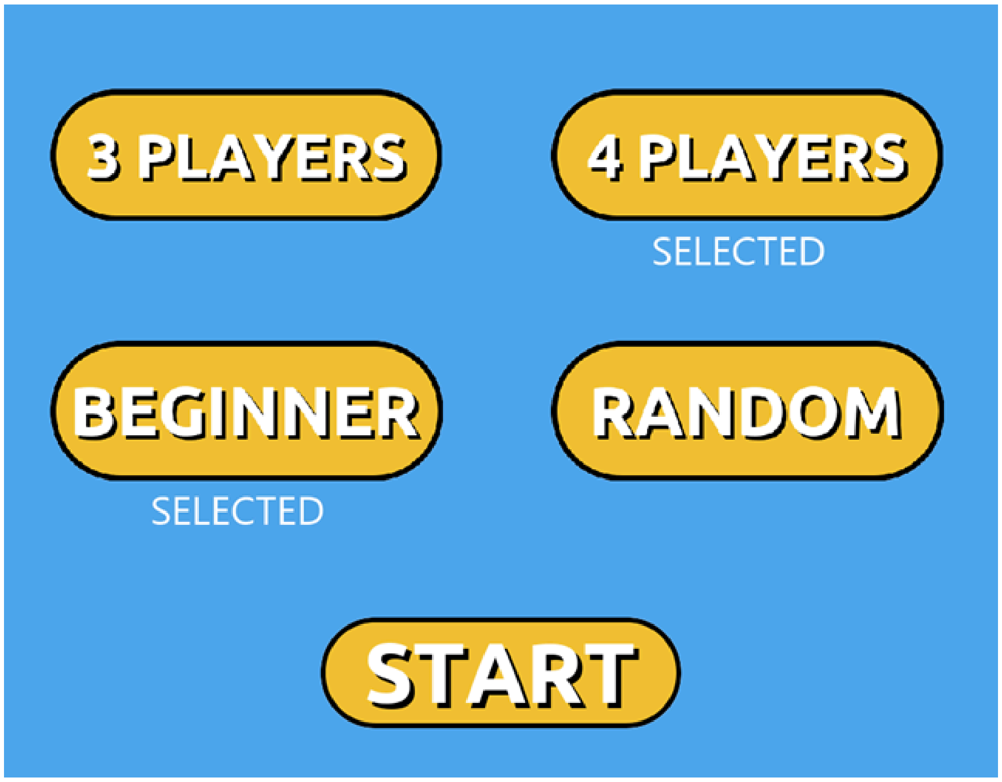


This board would be changed a lot, and lots of our buttons would be moved around. One of our decisions was to make a menu where you could choose the board instead of having it at the bottom of the screen, this also paved the way for much more space to work with.

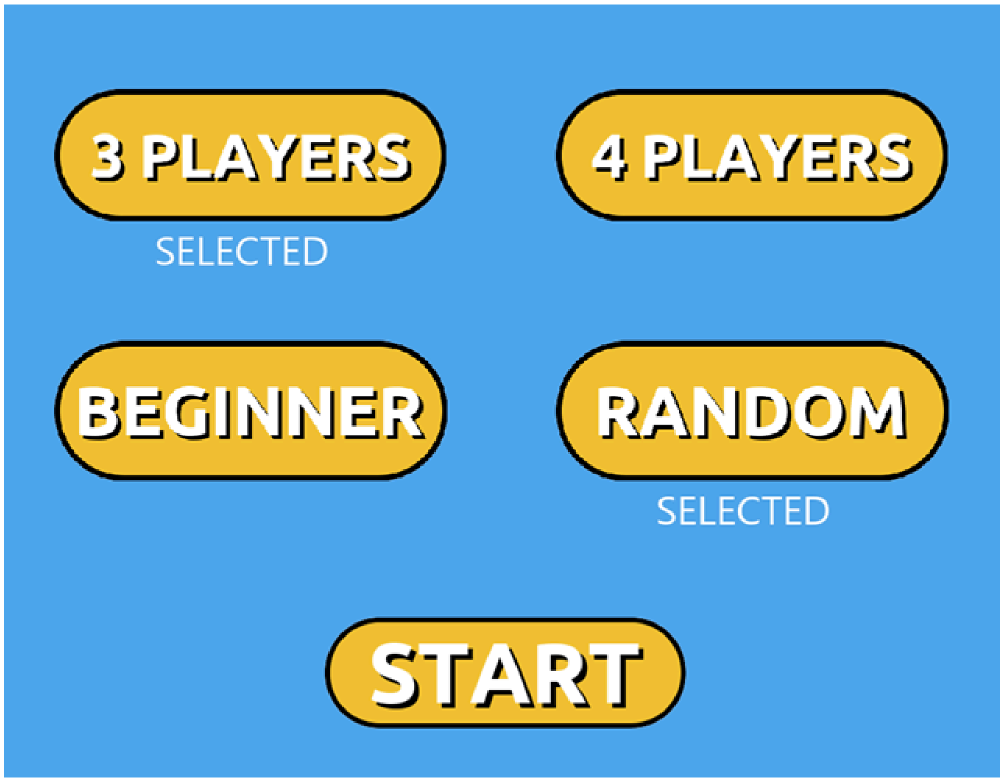
The first screen you see when you start up the game looks like this



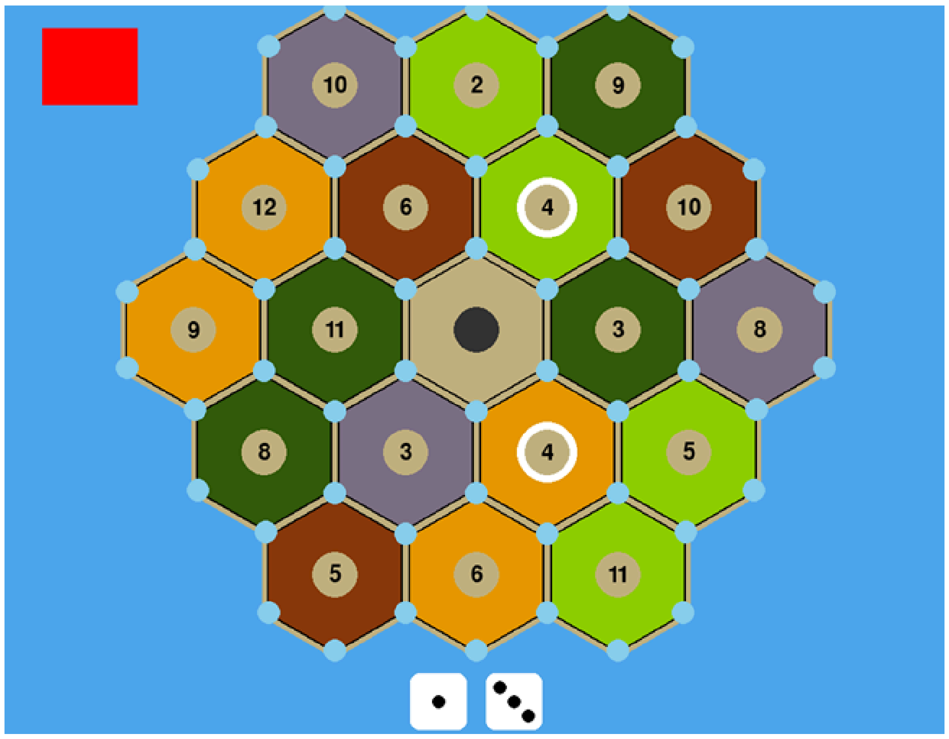
After you click the start button, you are greeted with this page



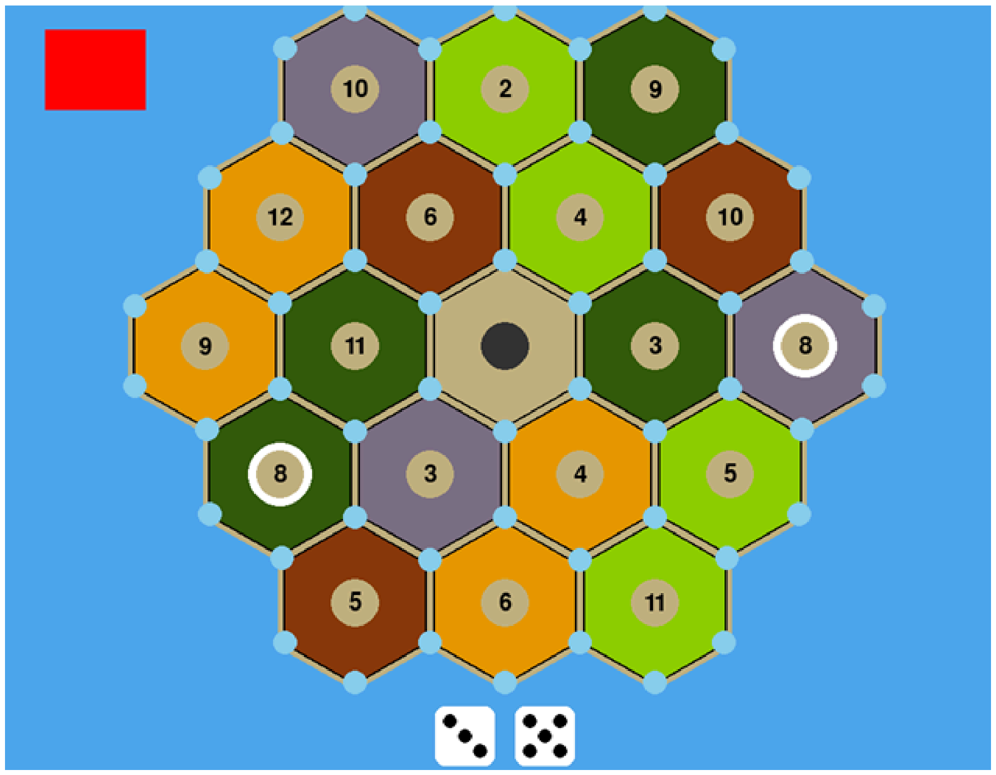
This shows that you can select the other options

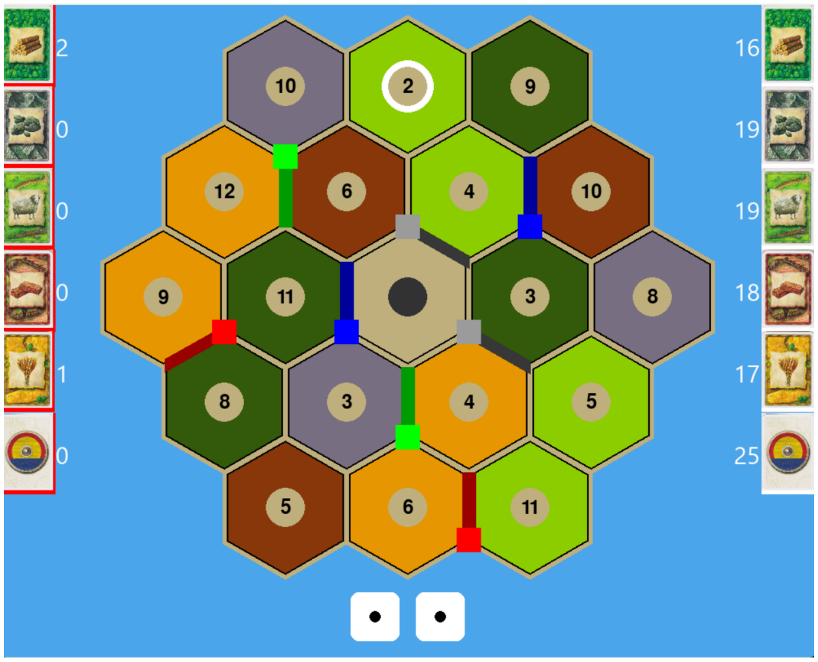


The beginner Board looked like this

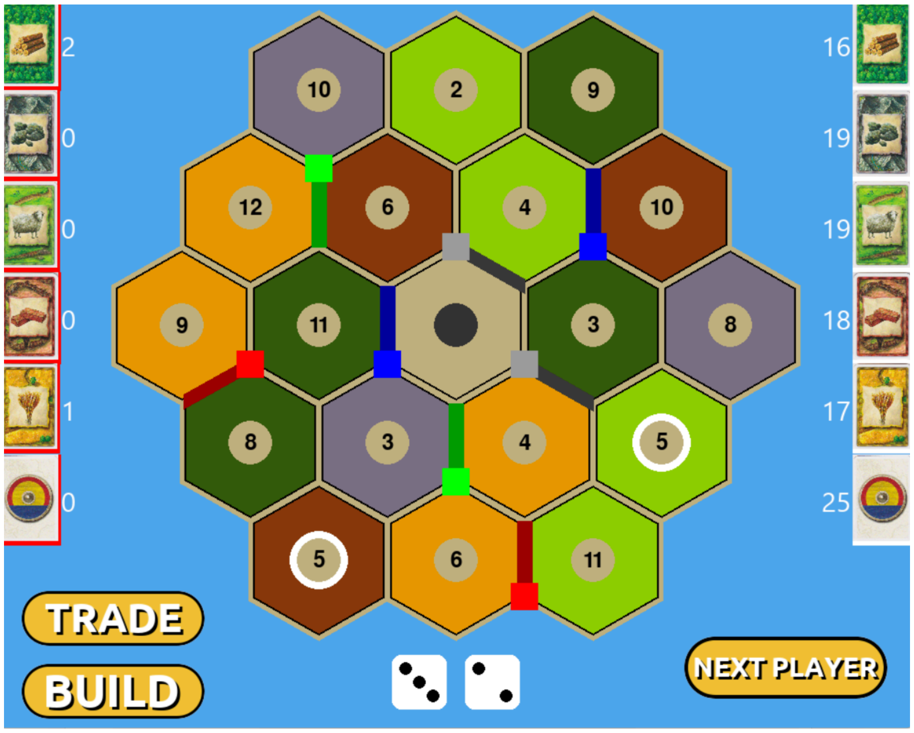


And after a dice roll





Players then take turns setting up initial settlements and roads, and then Player 1 begins the game, as noted by the background color of the cards of the left side. The player must click the dice to roll them to do anything else.



Player has options to trade, build or move on to the next player.

Additionally, with these changes we added the cards to the sidebar

Wood

Dev Card

Ore

Sheep

Wheat

Brick

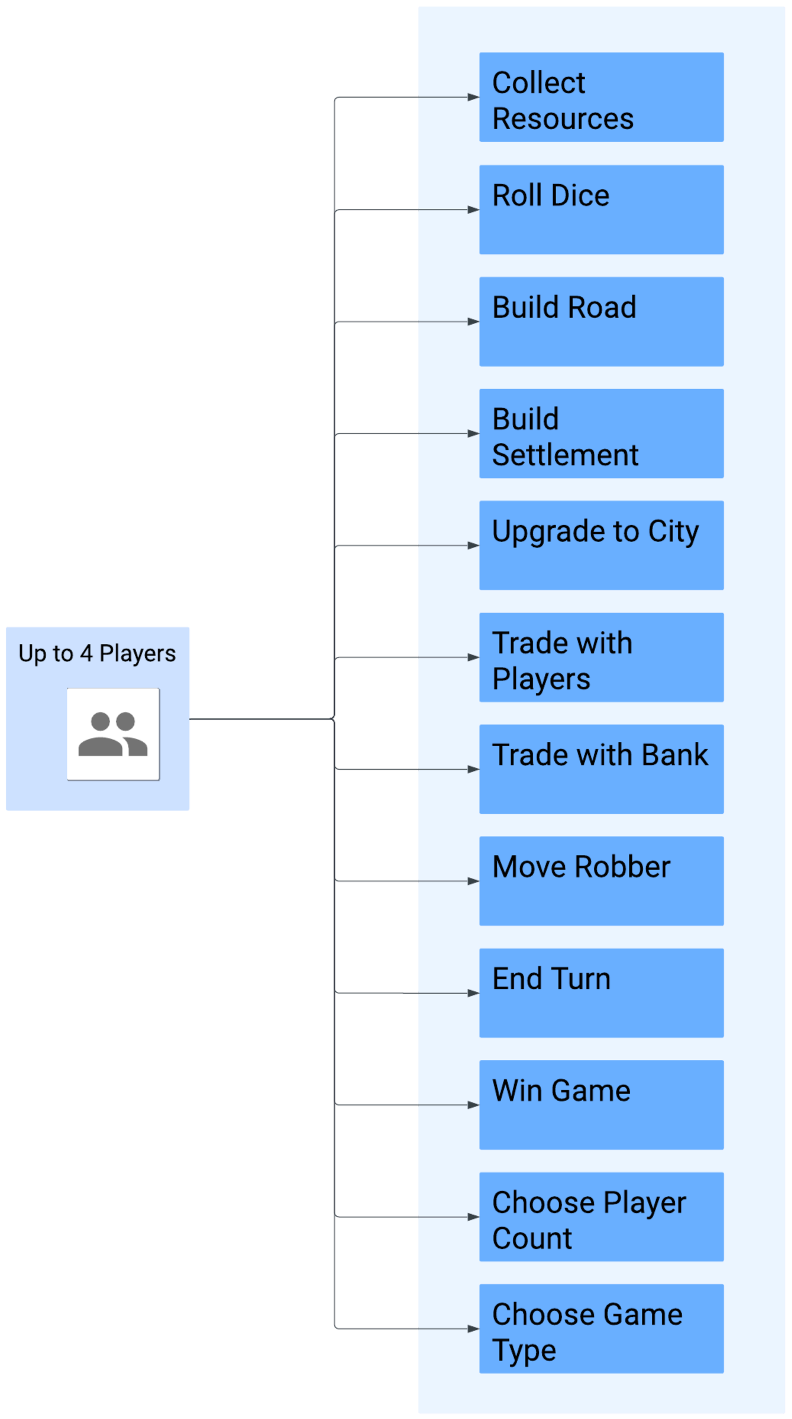
Added More buttons for building

ROAD
SETTLEMENT
CITY
BUILD

Cycling through Players

PLAYER 3
PLAYER 2
PLAYER 1
NEXT PLAYER
PLAYER 4

## Use Cases

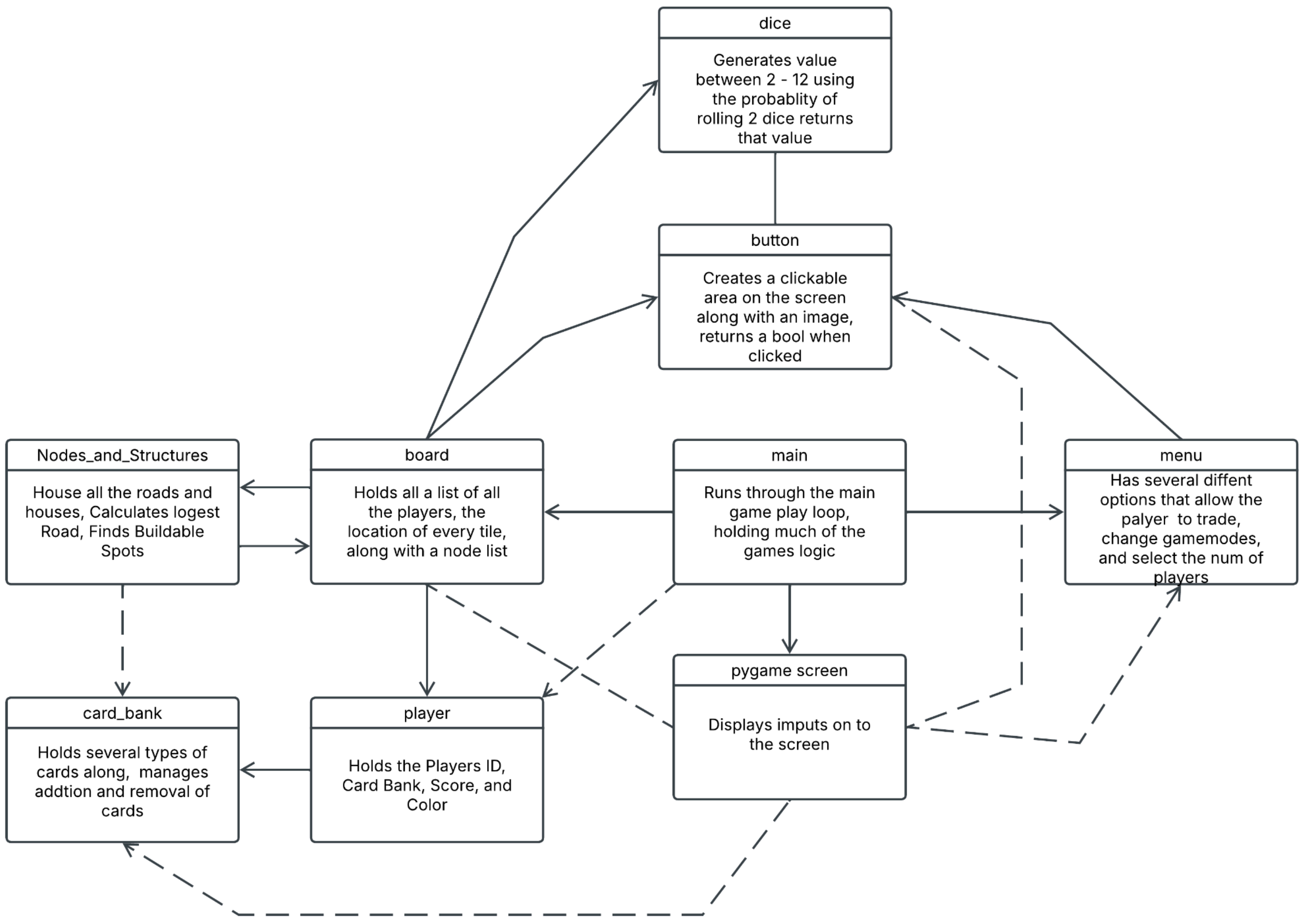

Up to 4 Players
Collect Resources
Roll Dice
Build Road
Build Settlement
Upgrade to City
Trade with Players
Trade with Bank
Move Robber
End Turn
Win Game
Choose Player Count
Choose Game Type

| Name | ID | Brief Description | Actors | Triggers | Preconditions |
|---|---|---|---|---|---|
| Roll Dice | UC1 | Simulates the dice roll at the start of a player's turn to determine which hexes produce resources. | Primary Actor: Player | Start of a player's turn. | It is the player's turn. |
| Collect Resources | UC2 | Players collect resources based on settlements/cities adjacent to activated hexes. | Primary Actor: Player | Triggered after a dice roll. | Valid board setup; dice have been rolled. |
| Build Road | UC3 | Allows the player to spend resources to place a road. | Primary Actor: Player | Player selects 'Build Road' option during their turn. | Player has at least 1 brick and 1 lumber. |
| Build Settlement | UC4 | Allows the player to place a new settlement on the board. | Primary Actor: Player | Player chooses to build a settlement. | Player has the correct resources and placement follows game rules (e.g., distance rule). |
| Upgrade to City | UC5 | Allows the player to upgrade an existing settlement to a city. | Primary Actor: Player | Player chooses to upgrade. | Player owns the settlement and has required resources. |
| Trade with Player | UC6 | Allows two players to exchange resources during a turn. | Primary Actor: Player, Secondary Actor: Other Player | Initiated by a player on their turn. | Both players agree on the trade. |
| Trade with Bank or Ports | UC7 | Allows players to trade resources with the bank or via ports at specified rates. | Primary Actor: Player | Player initiates trade during their turn. | Player has enough resources for the trade; port access if using non-bank rates. |
| Choose Player Count | UC8 | Allows the player to pick how many players are in the game (max - 4, min -3) | Primary Actor: Player | Player initiates | The Game has not started |

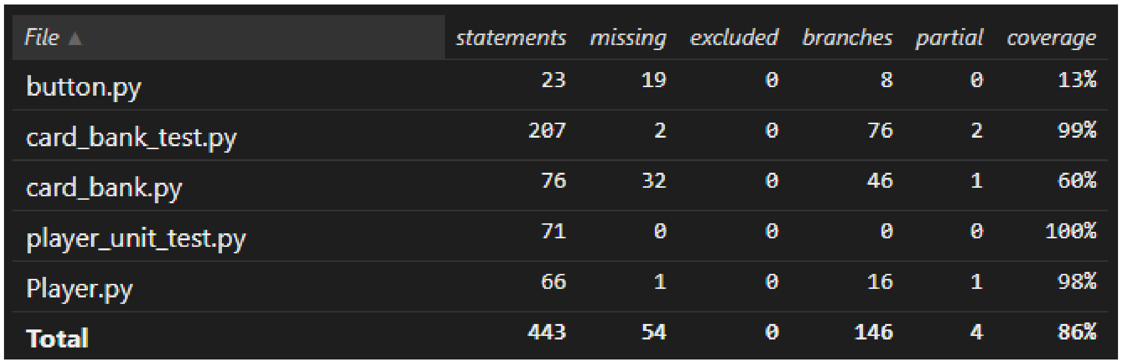
| | | | | | |
|---|---|---|---|---|---|
| Choose Game Type | UC9 | Choose between a random board and a preset board layout | Primary Actor: Player | That board is played on for the rest of the game | The Game has not started |
| End Turn | UC10 | Ends the current player's turn and passes control to the next player. | Primary Actor: Player | Player chooses to end their turn. | All current actions are completed. |
| Win Game | UC11 | Triggers when a player reaches 10 victory points. | Primary Actor: Player | Player action results in total VP reaching 10. | Player has sufficient VP. |

# Class Diagram

This diagram was made using Lucid Charts, a clearer version can be found at: https://lucid.app/lucidchart/f1dec094-2d06-4e42-814b-65b919653d88/edit?viewport_loc=-573%2C-684%2C1655%2C1990%2C0_0&invitationId=inv_b134ff99-0ee7-4f89-8646-58a3481ccdc1

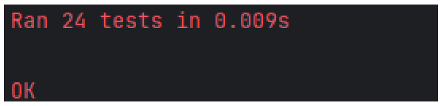

dice
Generates value between 2 - 12 using the probablity of rolling 2 dice returns that value
button
Creates a clickable area on the screen along with an image, returns a bool when clicked
Nodes_and_Structures
House all the roads and houses, Calculates logest Road, Finds Buildable Spots
board
Holds all a list of all the players, the location of every tile, along with a node list
main
Runs through the main game play loop, holding much of the games logic
menu
Has several diffent options that allow the palyer to trade, change gamemodes, and select the num of players
card_bank
Holds several types of cards along, manages addtion and removal of cards
player
Holds the Players ID, Card Bank, Score, and Color
pygame screen
Displays imputs on to the screen

## Code Coverage



| File ▲ | statements | missing | excluded | branches | partial | coverage |
|---|---|---|---|---|---|---|
| button.py | 23 | 19 | 0 | 8 | 0 | 13% |
| card_bank_test.py | 207 | 2 | 0 | 76 | 2 | 99% |
| card_bank.py | 76 | 32 | 0 | 46 | 1 | 60% |
| player_unit_test.py | 71 | 0 | 0 | 0 | 0 | 100% |
| Player.py | 66 | 1 | 0 | 16 | 1 | 98% |
| **Total** | 443 | 54 | 0 | 146 | 4 | 86% |

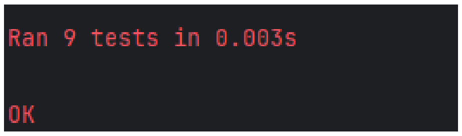
*Coverage report on automatic test files*



```
Ran 24 tests in 0.009s

OK
```

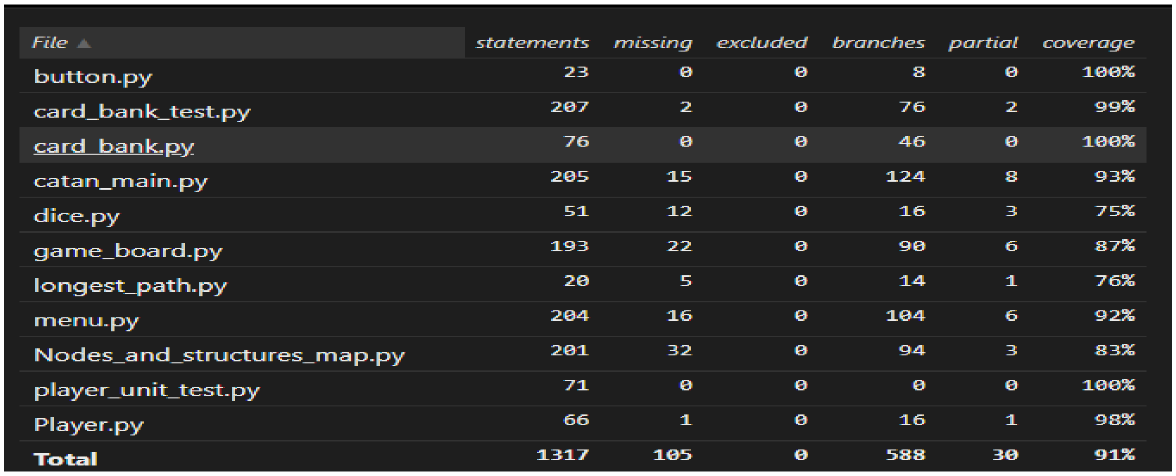
*Card_bank_test.py output*



```
Ran 9 tests in 0.003s

OK
```

*Player_unit_test.py output*



| File ▲ | statements | missing | excluded | branches | partial | coverage |
|---|---|---|---|---|---|---|
| button.py | 23 | 0 | 0 | 8 | 0 | 100% |
| card_bank_test.py | 207 | 2 | 0 | 76 | 2 | 99% |
| card_bank.py | 76 | 0 | 0 | 46 | 0 | 100% |
| catan_main.py | 205 | 15 | 0 | 124 | 8 | 93% |
| dice.py | 51 | 12 | 0 | 16 | 3 | 75% |
| game_board.py | 193 | 22 | 0 | 90 | 6 | 87% |
| longest_path.py | 20 | 5 | 0 | 14 | 1 | 76% |
| menu.py | 204 | 16 | 0 | 104 | 6 | 92% |
| Nodes_and_structures_map.py | 201 | 32 | 0 | 94 | 3 | 83% |
| player_unit_test.py | 71 | 0 | 0 | 0 | 0 | 100% |
| Player.py | 66 | 1 | 0 | 16 | 1 | 98% |
| **Total** | 1317 | 105 | 0 | 588 | 30 | 91% |

*Coverage with manual testing*

## Justification:

The automated tests ran to cover all data holding items. The card_bank.py file has methods that involve the GUI, such as creating cards for the visualization of the bank. The other 60% is code that was able to be tested on. The button.py file was included because of the card_bank __init__ which includes initializing some buttons. The manual tests are difficult to achieve, so as much was covered as possible to be able to make them as close to 100% code coverage as possible.

# Team Breakdown

## Owen Hillary

Role: Product Dev Team

Responsibilities: Created UML diagrams and Project Plans, in addition to numbers and titles randomization.

Self Reflection: As a fan of the board game I have found the motive for this project easy to find and project work fun.  This is my first time working on a project that requires coding with others and I find it's nice to see other developers' viewpoints.  I’ve worked on games in python and other languages in the past before allowing me to apply previously learned skills to this project while learning new techniques.  I think the second half of this project will be fun and I can't wait to see the challenges we’ll find.

## Sam Rios

Role: Product Dev Team

Responsibilities: Created various GUI elements and listeners to facilitate actions of buttons in the GUI.

Self Reflection: I have enjoyed my time so far working on this project! It’s interesting to see all the differences that Java and Python have, since I primarily have used Java in the past, more notably in my MTU classes. Being able to figure out how to make a GUI using pygame was also fun and challenging. I enjoy working together as a team with my group members and I just enjoy this game in general, so to be able to make it was really cool!

## Owen Westerkamp

Role: Product Dev Team

Responsibilities: Worked on backend classes and test cases, such as the Player class

Self Reflection: The project so far has been very satisfying and fulfilling to complete with my groupmates, I grew up playing Catan with my family so it has been a lot of fun to design a Python game after it. This project has helped me understand project design better and teamwork. I’ve worked on simple games like Chess and Checkers but those games are a lot more simpler than a game with so many rules like Catan.