# CS 342: Project 4

## Team 3:

Lam, Samantha

Martinez, Manuel

Omene, Ese

Ramirez Gomez, Christian

**Logic Description:** The server and client communicate by sending a WordInfo object between each other. The client just sends the category it's picking, the letter it's guessing, and whether it's playing again or quitting. On the other hand, the server would send a word's length and the results of a guess. The client kept track of it's guess, lives, and what categories it had cleared. This seemed easier than having to send a lot of variables inside the wordInfo object and having to copy them over each time they were sent.

**UI and UX:** We wanted to make the game look as simple as possible, while also making great use of the space we had. From the title page to the end page, each narration and button gives clear direction on what the user should do, and uses images and colors as extra features to give the player a sense of direction. Every page utilizes the negative space effectively by being directly in the center of the screen without being congested.

**How we divided up work:** We discussed and chose what parts we wanted to work on.

**Collaboration:** We mostly used Discord to message each other and a few times though audio to explain things in more depth. We also set up a GitHub repository to be able to share the project.

**What worked and what did not:**

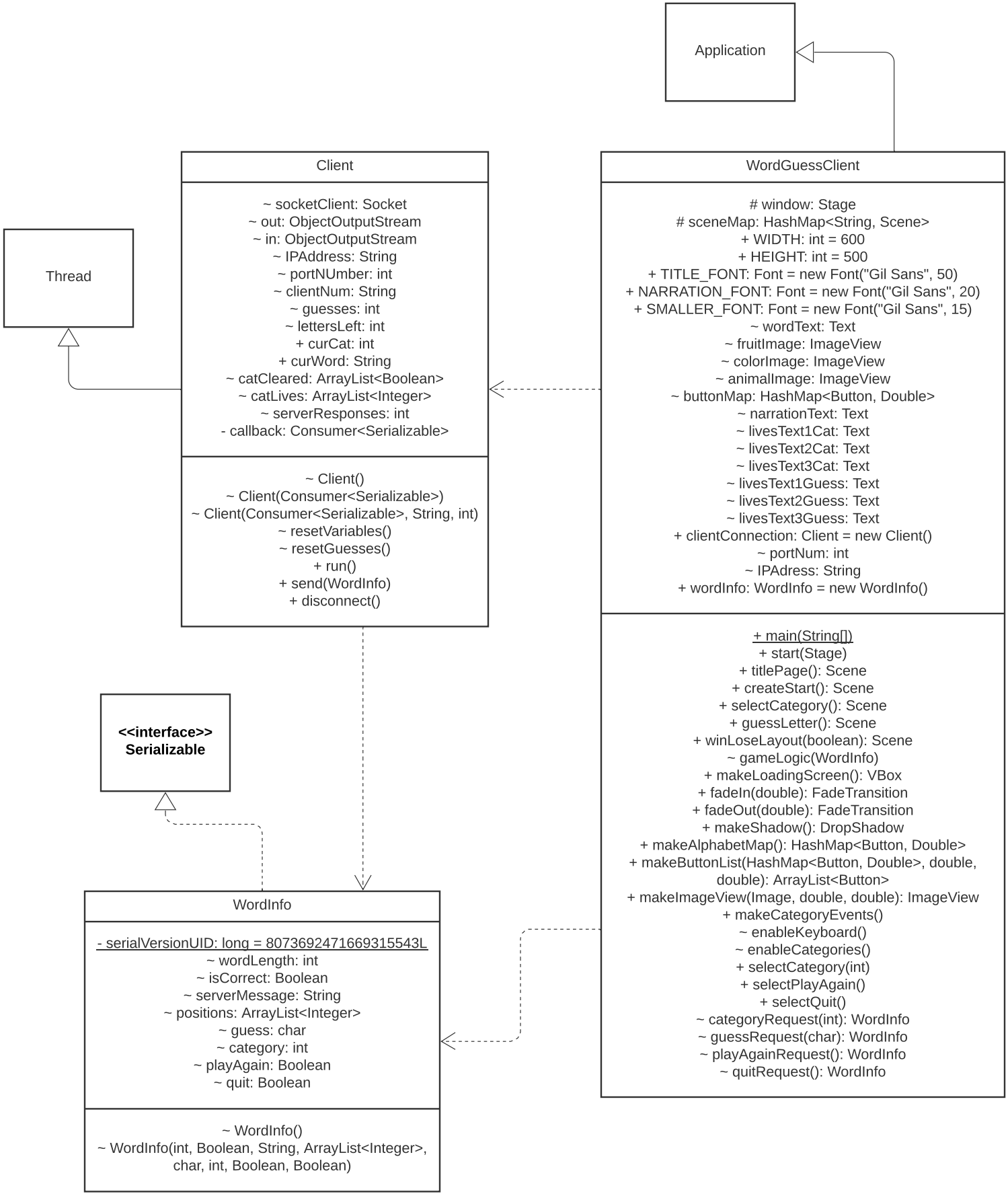
- What worked: Talking to each other when we had questions. Keeping each other updated on progress. Holding each other accountable. Suggesting ideas to each other. Discord allowed us to communicate in a direct manner.
- What did not work: The group experience went relatively smooth. We did not run into many problems. For the small problems we did run into we were able to solve by brainstorming. Discord allowed us to talk to each other, but meeting in person might have been better.

**The work was divided as follows:**

- Samantha: I created the wireframe for the Server and the UML diagrams for both the Server and Client.
- Manuel: I coded the test cases for both the client and the server. Coded part of the Server Logic and the Server GUI.
- Ese: I created the wireframe for the Client side of the game. I also created the GUI for the game.
- Christian: I did the server/ client classes to be able to communicate between them. I also did the Activity Diagram, the game logic inside the client, and part of the logic on the server side.

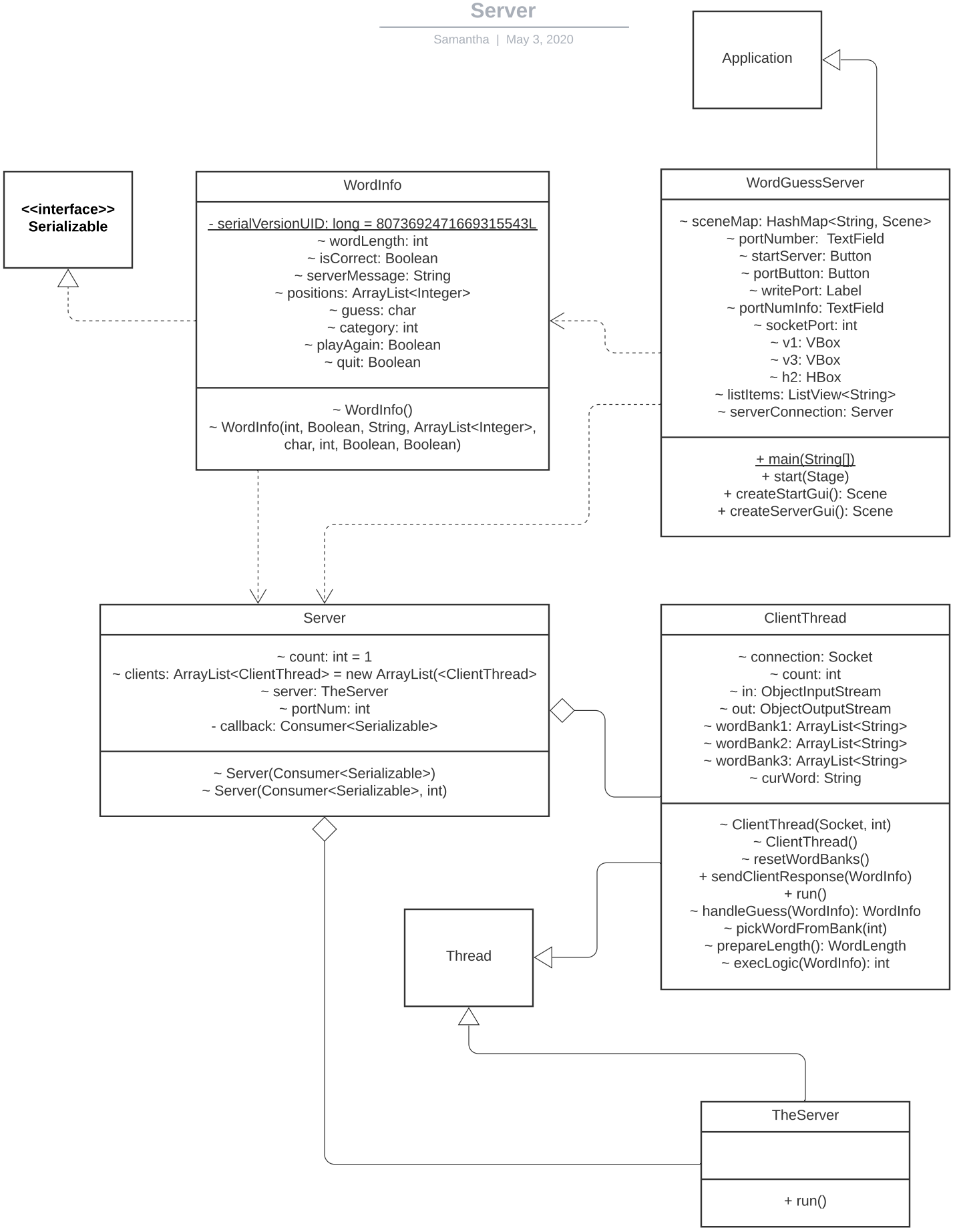
# Client

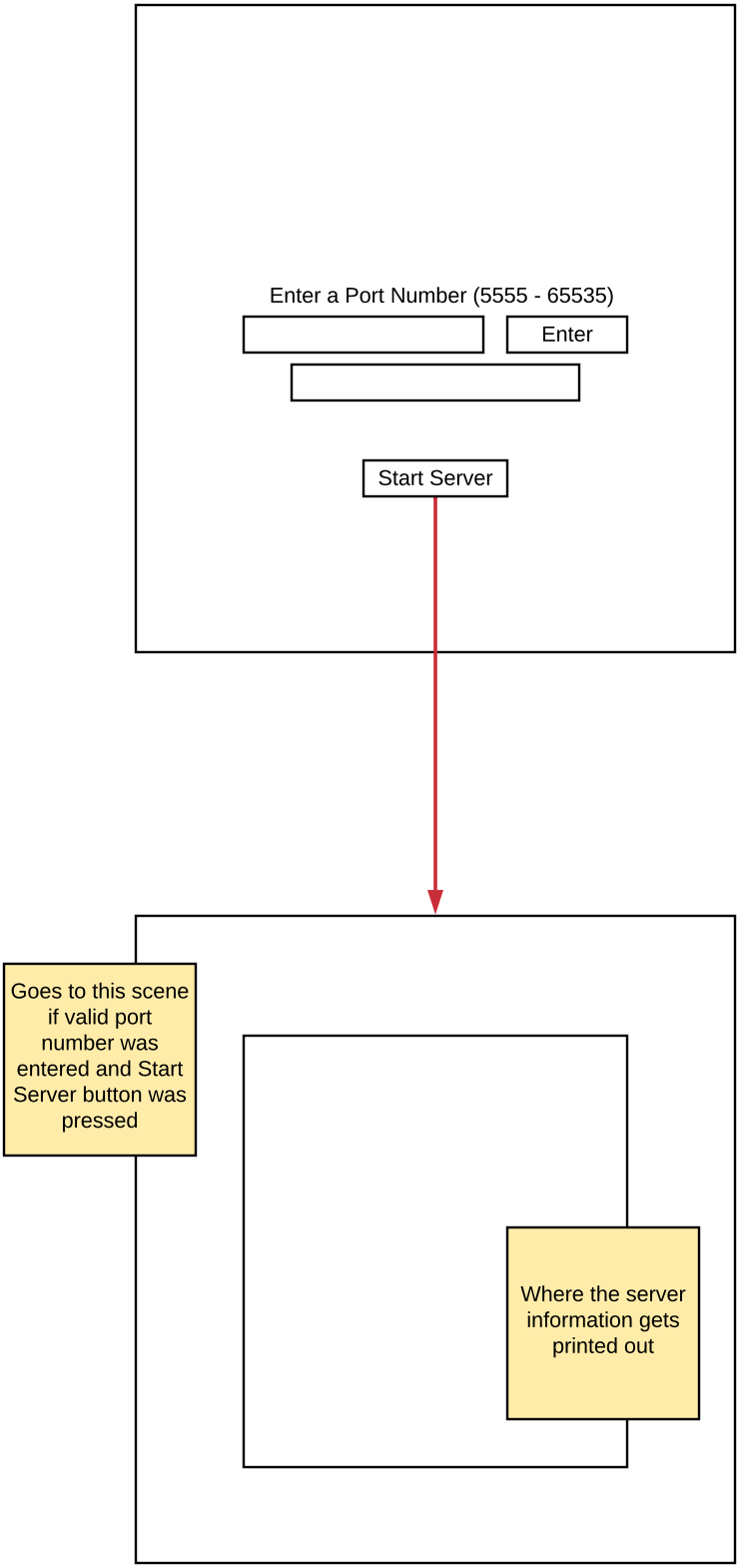
Samantha | May 3, 2020

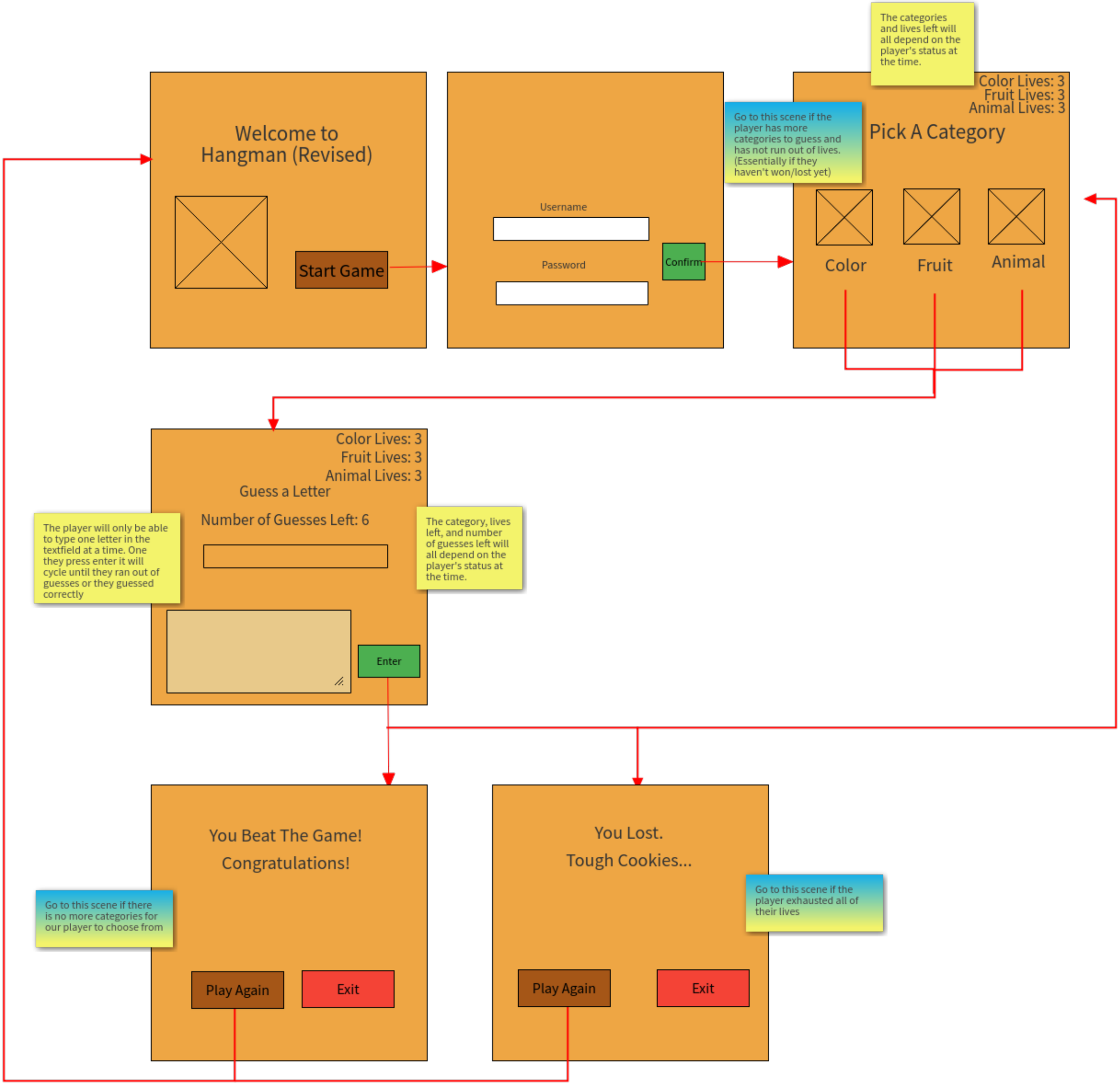


# Server

Samantha | May 3, 2020




Enter a Port Number (5555 - 65535)
Enter
Start Server
Goes to this scene if valid port number was entered and Start Server button was pressed
Where the server information gets printed out


Welcome to Hangman (Revised)
Start Game
Username
Password
Confirm
Go to this scene if the player has more categories to guess and has not run out of lives. (Essentially if they haven't won/lost yet)
The categories and lives left will all depend on the player's status at the time.
Color Lives: 3
Fruit Lives: 3
Animal Lives: 3
Pick A Category
Color
Fruit
Animal
Color Lives: 3
Fruit Lives: 3
Animal Lives: 3
Guess a Letter
Number of Guesses Left: 6
The player will only be able to type one letter in the textfield at a time. One they press enter it will cycle until they ran out of guesses or they guessed correctly
The category, lives left, and number of guesses left will all depend on the player's status at the time.
Enter
You Beat The Game!
Congratulations!
Go to this scene if there is no more categories for our player to choose from
Play Again
Exit
You Lost.
Tough Cookies...
Go to this scene if the player exhausted all of their lives
Play Again
Exit

# Activity diagram

Kurisu | April 20, 2020