Team Good

Xuejian Tom Yu
Ruoyi Pu
Julia Lou

Game Technology Assignment 3 Part 2 Report

So far, we've created a menu that pops up when you first start the game. Clicking the "single player" button in the menu will take you to the original game that we turned in for Assignment 2. As you play the game pressing the escape button will take you to a pause menu, which give you options for either exiting the game or returning to the game. The "Multiplayer" button has not been implemented yet, so there is currently no reaction when you click on it. Ideally we hope for the "Multiplayer" button to lead to another menu that allows the players to pick either "1st player" or "2nd player". The "1st player" button will lead to a window that allows that player to input a port number. The "2nd player" button will lead to a window that allows that player to input a port number(ideally the same port number that the 1st player inputted) and the hostname(entering the IP address also works) of the 1st player. To run this demo follow the instructions below.

Input into the terminal:
./buildit
./Project1

We've also created a sample code that tests a network wrapper that we wrote. After weighing the pros and cons of TCP versus UDP we decided to use TCP. The sample is meant to be run simultaneously in two different terminals so that they can communicate with each other. One terminal acts as the server while the other acts as the client. Each terminal can only send one message to the other in order to test if we can connect and send and receive messages properly. Then the test program terminates. To run the test code, follow the instructions below.

In order to run this test code you will have to edit the Makefile.am. First uncomment the originally commented out lines. Then comment out the line directly above those lines. Run ./ buildit to compile the code. Then run ./Project1 in two different terminals so that they can communicate with each other.

Since we have this working network code, we plan on merging it with our game code. Our game code base has been modified with a more advanced framework to accommodate for the network code and different game states. We are using the server/client model to represent our network. We plan on using the menu that allows for easy navigation to the right sub-menu to initiate network connection and gameplay. Both players game state will differ due to the different processing and input needs of the client and server.