Real-Time Network Multiplayer Game

COP701: Software Systems Laboratory
Assignment 2

by
Dharmisha Sharma - 2022MCS2062
Raajita Bhamidipaty - 2022MCS2053

Guide: Prof. Rahul Narain

Department of Computer Science and Engineering
INDIAN INSTITUTE OF TECHNOLOGY DELHI

Acknowledgement

We would like to thank Prof. Rahul Narain for granting the opportunity of working on this interesting assignment and for his invaluable help and guidance. We would like to express our deep sense of gratitude to our fellow classmates for their important help and technical suggestions. Finally we would like to thank our families and friends for their constant moral support.

Abstract

Reach the Goal, is a multi-player real network competitive game. It allows two players to connect via a network and play against each other. A goal has been defined in the game and the player who reaches the goal first wins. Since it is a competitive game the players need to be as fast as they can for their way to victory.

A server is maintained for the interaction of both the players which act as clients of that server. Each player receives a question from the main server. The one who correctly answers the questions before the other player wins a point a moves a step closer to the goal.

Contents

List of Figures

1 Introduction

The main components of a multiplayer real network game are :

1. Players (Clients)
2. Server (Through which clients will connect to each other)
3. GUI (an interactive platform where the game will be played)

The main idea of our game is that two players will connect via a server. After connecting to the server they can access the GUI.

There is a goal point towards which each players has to proceed faster than the other, since it is a competitive game.

The server will send an arithmetic problem to each player via the GUI. Each time a player answers the problem correctly the server is notified and based on whichever player has answered first, he/she steps ahead towards the goal and the other player will step back away from the goal. After a player has answered a problem correctly, i.e. written the correct answer in the provided textbox, the client notifies the server and the apt step ahead/back are done. The server then sends a new question to each player. This continues till a player reaches the goal, i.e. he/she wins the game.

2 GUI Design

The GUI design is done via a python library : PyGame, which is a Python wrapper for the SDL library, which stands for Simple DirectMedia Layer.

The GUI displays :

1. A main window - where all the components of the GUI will be displayed, each player has his/her own window on their machine.
2. Two Players - Two objects of the player class which are basically the clients
3. Scores of both the players - which are updated each time a player gives a correct answer.
4. Arithmetic Question - randomly generated by server via a Random Question Generator class.
5. A textbox to input the answer - where each player inputs their answer
6. A goal - the goal point where each player aim to reach by stepping ahead
7. Path that leads the players to the goal - a path to trace the steps of each player.
8. Some audio effects - Audio effects are that play whenever a player gives wrong/right answer and at the end when a winner is declared.

2.1 Screenshots :

Figure 1: The initial GUI page

Figure 2: TextBox Entry

3 Working of the Various Components

1. Start the server via the run server file where the host and port number are set.
2. Connect a client by running the run client file. Once a client is connected to the server, he/she waits for the other client.

Figure 3: Wait Page

3. Connect the second client by running the run client file on the second player's machine. Once he/she joins the GUI gets updated for both the players, where the score and the random question sent by the server are updated.

Figure 4: Start Page

4. Each player can write his/her in the textbox present below the question.

Figure 5: Textbox

5. Once a player gives the correct answer he/she moves further, the score gets incremented and a newly generated random question appears on the screen.

Figure 6: Updated Position, Score and question

6. Once a player reaches the goal he/she wins and a message is displayed.

Figure 7: End Page

4 Software Architecture

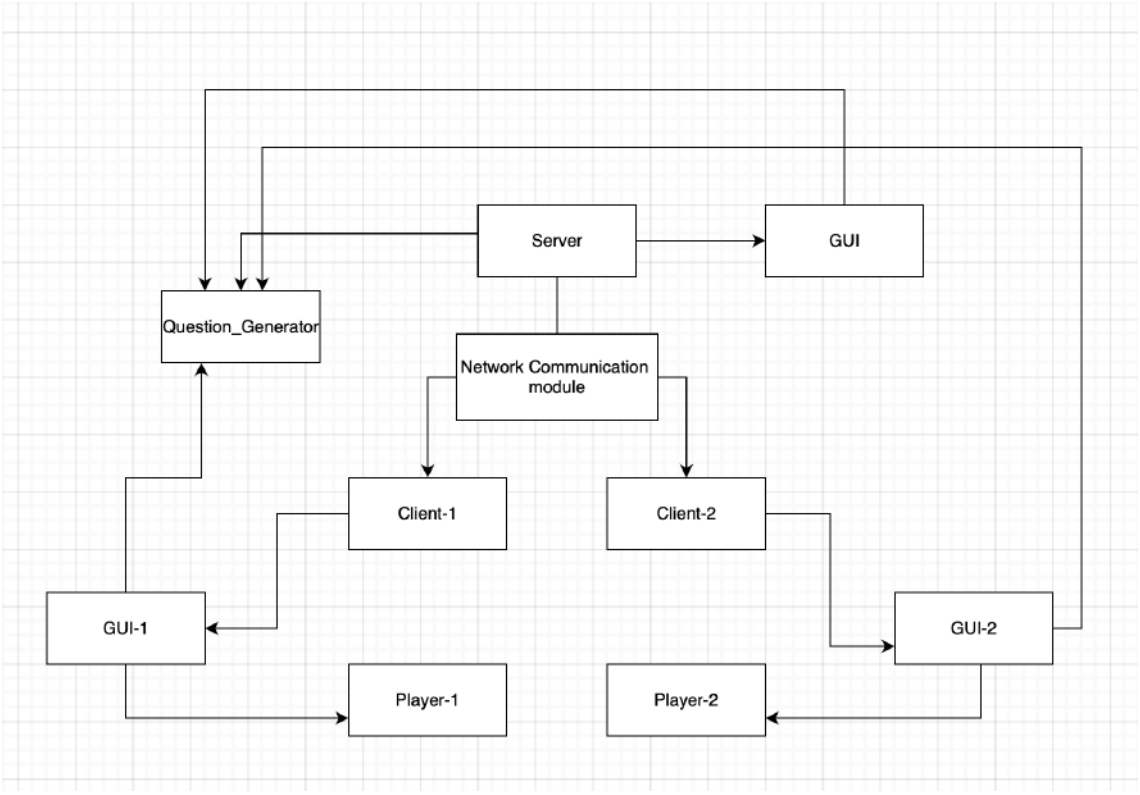

Figure 8: Software Architecture

4.1 Rationale behind the architecture

The main requirement was to build a multiplayer game where the players can connect over a network and play the game.

The main idea of the game is that each player receives a randomly generated question, so for that we have created a Question Generator class that returns a randomly generated question and it's answer. Since, the game is a multiplayer game, we have made a player class. Each player in the game is an object of this player class.

The game needs to have an interactive environment, so we made a class GUI which creates the game window and it's various components and displays it on the user screen for inter activeness.

Now since the GUI and players are made, the players also need to communicate over a network, so we made server and client classes. The client will connect over a server and use the Player class and GUI class for playing. The server will pass the required messages to the client, in our case these messages are : the randomly generated arithmetic question, the score, the message that which player has answered correctly.

5 Individual Contributions

2022MCS2062 - Built the GUI and the functions required for integrating the GUI with client and server. Worked for fulfilling the additional requirements: creating python package and version control.

2022MCS2053 - Set up the client server connection and interaction of clients over the server, integration with GUI, Worked for fulfilling the additional requirements: version control.

6 References

1. https://www.pygame.org/
2. https://www.freepik.com
3. https://www.techwithtim.net
4. https://toppng.com/
5. https://realpython.com/pygame-a-primer/