Multiplayer Bingo Game

Chinmayi C.R. - 181IT113
Department of Informaion Technology
National Institute of Technology
Karnataka Surathkal, India 575025
Email: chinmayicr27@gmail.com

Meghna Kashyap - 181IT127
Department of Informaion Technology
National Institute of Technology
Karnataka Surathkal, India 575025
Email: kashyap.meghna13@gmail.com

K Keerthana - 181IT221
Department of Informaion Technology
National Institute of Technology
Karnataka Surathkal, India 575025
Email:
keerthanakanapuram@gmail.com

Socket programming is a way of connecting two nodes on a network to communicate with each other. One socket(node) listens on a particular port at an IP, while other socket reaches out to the other to form a connection. Server forms the listener socket while client reaches out to the server. We aim to implement a multiplayer game in which two clients can interact with each other on a server using socket programming.

I. INTRODUCTION

- This project is a representation of a Bingo game using a server and multiple clients.
- C++ is used to code the different components of the game. The concept of socket programming is implemented.
- Two C++ programs are run simultaneously on separate terminals. The server is known as '**Game**' and the clients are known as '**Player**'.

II. LITERATURE SURVEY

A. UDP Based Chat Application.

By Akshith Malhotra, Vaibhav Sharma, Prateek Gandhi, Neetesh Purohit.

The paper depicts the implementation of socket programming using Java which includes features like peer to peer communication, multicasting and file sharing.

The main advantages of this IEEE paper are that UDP supports multicasting and broadcasting and is seen to provide a reliable connection.

The drawbacks taken from this paper depict System and Network failure.

B. Client- Server based Application in a distributed computing environment

by Rolou Lyn R. Maata, Ronald Cordova.

This paper focuses on development of a client-server-based application called OpTel Billing System. It uses Java NetBeans and TCP datagram.

The application is seen to be very efficient and consists of direct communication. For further ease of implementation, Java NetBeans covers a wide range of functions and classes.

The drawbacks taken from the paper are connection loss problems and speed-oriented problems which surface during implementation.

III. PROBLEM STATEMENT

This project implements a multiplayer game Bingo in which two clients from different devices establish a connection with the server to implement the game BINGO. The game broadcasts information via a multicast socket, uses multiple synchronous threads, and communicates with RPC and connects two players to the virtual game world using socket programming.

IV. METHODOLOGY

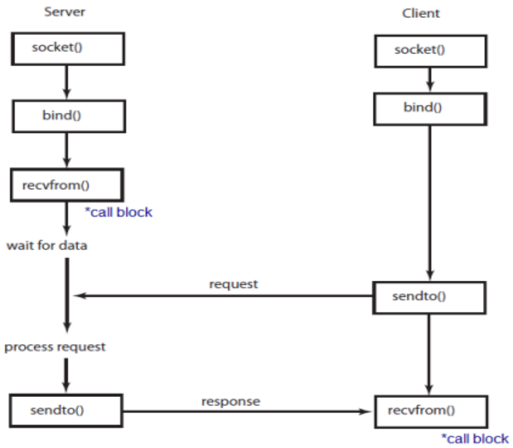
Algorithm:

This project includes two programs: one for the client and one for the server. The client module includes functions for getting sign up details and inputs for the game from the user. The server module includes the code to execute the game.

- A 5X5 matrix will be displayed to each player and the player is allowed to enter numbers between a certain range.
- While entering numbers in the matrix, it is ensured that duplication of numbers does not occur.
- Using randomize() function, server generates a random number in the given range and is displayed on the screen.
- The player is asked if the number exists in their matrix.
- If yes, the player is asked the row and column number in which the number exists and their answer is verified by the server.

  If the number does not exist in that particular cell, the user is asked whether he/she would like to re-enter the row and column or exit.
- When a number exists in a given row and column, the counter assigned for that row and column is incremented by 1.
- The player wins if there is a count of 5 for either the same row or column.

Block diagram depicting client-server connection:



V. CONCLUSION

- The server creates a 5x5 matrix for each player.
- User is allowed to enter numbers from a range of numbers.
- When a player gets bingo, server sends a message to the winner stating that they won and a losing message to the remaining players.

REFERENCES

[1] Jouni Smed and Timo Kaukoranta, A Review on Networking and Multiplayer Computer Games, 2002.

[2] T. Alexander. Massively Multiplayer Game Development II. Charles River Media, 2005.

[3] J.B. Postel and J.K. Reynolds, File Transfer Protocol, [RFC 959], October 1985

[4] W.R. Stevens, Unix Network Porgramming, Prentice-Hall, Englewood Cliffs, N.J.