## A Mini- Project Report

**on**

## Interactive Multiplayer Jackpot Rush Game

Submitted to the

Pune Institute of Computer Technology, Pune In partial fulfillment for the award of the Degree of Bachelor of Engineering

in

#### Information Technology

by

|  |  |
| --- | --- |
| Kaiwalya Aney | 43202 |
| Pankaj Atram | 43206 |
| Yash Behare | 43207 |
| Swapnil Chitalkar | 43214 |

Under the guidance of

## Prof. Sachin Pande



#### Department Of Information Technology

Pune Institute of Computer Technology College of Engineering

Sr. No 27, Pune-Satara Road, Dhankawadi, Pune - 411 043.

**2022-2023**

## CERTIFICATE

## Interactive Multiplayer Jackpot Rush Game

**Submitted by**

|  |  |
| --- | --- |
| Kaiwalya Aney | 43202 |
| Pankaj Atram | 43206 |
| Yash Behare | 43207 |
| Swapnil Chitalkar | 43214 |

is a bonafide work carried out by them under the supervision of Prof Sachin Pande and it is approved

for the partial fulfillment of the requirement of **Lab Practice-V** for the award of the Degree of Bachelor of Engineering (Information Technology)

|  |  |
| --- | --- |
| **Prof Sachin Pande** | **Dr. A. S. Ghotkar** |
| Lab Teacher | Head of Department |
| Department of Information Technology | Department of Information Technology |
|  |  |
|  |  |
|  |  |

Place: Date:

II

## ACKNOWLEDGEMENT

We thank everyone who have helped and provided valuable suggestions for successfully creating a wonderful project.

We are very grateful to our Prof Sachin Pande, Head of Department Dr. A. S. Ghotkar and our principal Dr. S.T. Gandhe. They have been very supportive and have ensured that all facilities remained available for smooth progress of the project.

We would like to thank our professor and mentor Prof. Sachin Pandefor providing very valuable and timely suggestions and help.

Kaiwalya Aney

Pankaj Atram

Yash Behare

Swapnil Chitalkar

III

## ABSTRACT

Jackpot Rush is a multiplayer game in which a server generates a unique number and multiple clients guess the number. The game continues until one of the clients guesses the correct number. This game can be implemented using a client-server model based on distributed systems. The server generates the random number and manages the game state, while the clients send their guesses to the server and receive updates on the game state. This report describes the architecture and implementation of a Jackpot Rush game based on distributed systems and presents some results of running the game

IV

## LIST OF FIGURES

|  |  |  |
| --- | --- | --- |
| Figure Number | Figure Title | Page Number |
| 1 | Flow Diagram | 10 |
| 2 | Random number output at server | 13 |
| 3 | Winning and losing Scenario | 13 |

V

## LIST OF TABLES

|  |  |  |
| --- | --- | --- |
| Table Number | Table Title | Page Number |

**LIST OF TOPIC**

1. INTRODUCTION
2. SCOPE AND OBJECTIVE
3. PROJECT FLOW
4. CODE AND SNAPSHOT
5. RESULT
6. CONCLUSION AND FUTURE SCOPE
7. REFERENCES

x

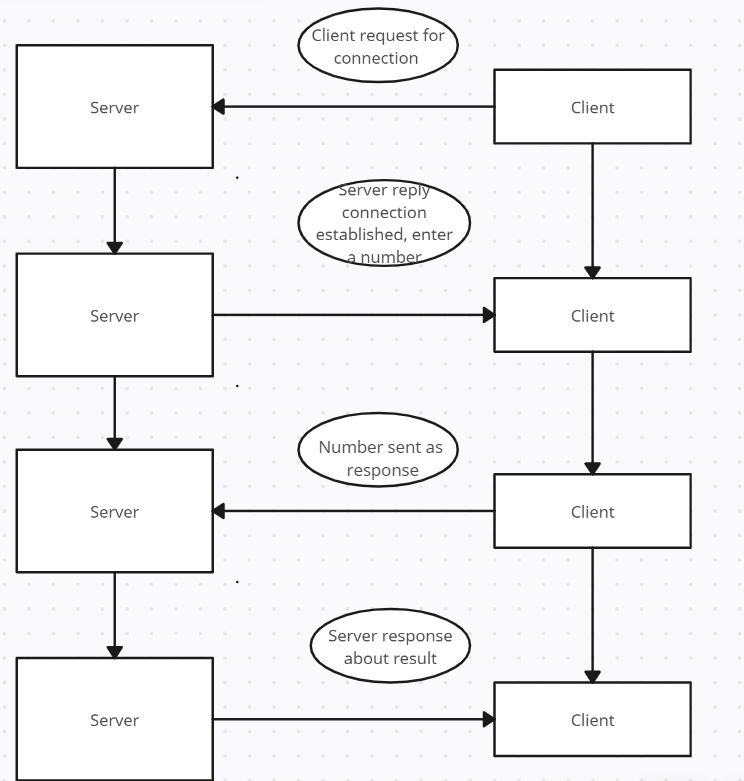
Introduction

Jackpot Rush is a multiplayer game in which the server generates a unique number and multiple clients guess the number until one of them correctly guesses the number. The game can be implemented using a client-server model based on distributed systems. In this report, we describe the architecture and implementation of a Jackpot Rush game based on distributed systems, using a client-server model

Scope and Objective

* The scope of the project is limited to a simple version of the Jackpot Rush game, with no additional features such as graphics or sound.
* The objective of this project is to develop a Jackpot Rush game using a client-server model based on distributed systems. The game should allow multiple clients to connect to a server, which generates a unique number and manages the game state. The clients should send their guesses to the server and receive updates on the game state.

Project Flow



Code and Snapshot

Client -

import socket

def client\_program():

    host = socket.gethostname()  # as both code is running on same pc

    port = 5000  # socket server port number

    client\_socket = socket.socket()  # instantiate

    client\_socket.connect((host, port))  # connect to the server

    message = input("Enter the number -> ")  # take input

    while message.lower().strip() != 'bye':

        client\_socket.send(message.encode())  # send message

        data = client\_socket.recv(1024).decode()  # receive response

        print('Received from server: ' + data)  # show in terminal

        message = input(" -> ")  # again take input

    client\_socket.close()  # close the connection

if \_\_name\_\_ == '\_\_main\_\_':

    client\_program()

Server code-

import socket

import random

def server\_program():

    # get the hostname

    host = socket.gethostname()

    port = 5000  # initiate port no above 1024

    lottery = random.randint(0,101)

    print(lottery)

    server\_socket = socket.socket()  # get instance

    # look closely. The bind() function takes tuple as argument

    server\_socket.bind((host, port))  # bind host address and port together

    # configure how many client the server can listen simultaneously

    while True:

      server\_socket.listen(100)

      conn, address = server\_socket.accept()  # accept new connection

      print("Connection from: " + str(address))

      # receive data stream. it won't accept data packet greater than 1024 bytes

      data = conn.recv(1024).decode()

      if int(data) == lottery:

        conn.send("\*\*\*Won\*\*\*".encode())

        conn.close()

      else:

        conn.send("Lost".encode())

if \_\_name\_\_ == '\_\_main\_\_':

    server\_program()

Results

The Jackpot Rush game based on distributed systems was successfully implemented using a client-server model. The server generated a unique number, maintained the game state, and communicated with the clients. The clients sent their guesses to the server and received updates on the game state. The game continued until one of the clients correctly guessed the unique number.

A screen shot of a computer

Description automatically generated with low confidence

Fig -2 Random number output at server

A picture containing text, screenshot, font

Description automatically generated

Fig 3.  Winning and loosing Scenario

Conclusion and Future Scope

In this project, we developed a Jackpot Rush game using a client-server model based on distributed systems. The game allowed multiple clients to connect to a server, which generated a unique number and managed the game state. The clients sent their guesses to the server and received updates on the game state. The game continued until one of the clients correctly guessed the unique number.

In future work, we could enhance the Jackpot Rush game with additional features such as graphics, sound, or a leaderboard. We could also improve the game's scalability by using load balancing and clustering techniques. Furthermore, we could implement security measures to prevent cheating and ensure the integrity of the game.

REFERENCES

1. Python Socket Programming: <https://docs.python.org/3/library/socket.html>
2. Distributed Systems: <https://www.sciencedirect.com/topics/computer-science/distributed-system>
3. Client-Server Model: <https://en.wikipedia.org/wiki/Client%E2%80%93server_model>
4. Random Number Generation: <https://en.wikipedia.org/wiki/Random_number_generation>