NITTE EDUCATION TRUST | N.M.A.M. INSTITUTE OF TECHNOLOGY
(An Autonomous Institution affiliated to Visvesvaraya Technological University, Belagavi)
Nitte – 574 110, Karnataka, India
(ISO 9001:2015 Certified), Accredited with 'A' Grade by NAAC
☎: 08258 - 281039 – 281263, Fax: 08258 – 281265
Department of Computer Science and Engineering
B.E. CSE Program Accredited by NBA, New Delhi from 1-7-2018 to 30-6-2021

Report on Mini Project

"Client-Server Architecture"

Course Code: 18CS607

Course Name: Computer Graphics Lab

Semester: 6th Section: C

Submitted To,

Mr. Puneeth K R

Asst Prof Gd III, Dept of CSE, NMAMIT

Submitted By:

Name: Pratheeksha P USN:4NM18CS118
Name: Sajja Manoj Kumar USN:4NM18CS147

Date of submission: 28/05/2021

Signature of Course Instructor

CLIENT-SERVER ARCHITECTURE

ABSTRACT

The aim of this project is to illustrate the concepts and usage of functionalities present in OpenGL. The project deals with basic network model. This network model is further used to illustrate certain functionalities of client-server architecture present in TCP/IP network. Functionalities involve establishment and disconnection of link between client (user) and server, and also the data transfer between user and another user.

INDEX

INTRODCTION

A computer network extends interpersonal communications by electronic means with various technologies, such as email, instant messaging, online chat, voice and video telephone calls, and video conferencing. A network allows sharing of network and computing resources. Users may access and use resources provided by devices on the network, such as printing a document on a shared network printer or use of a shared storage device. Most modern computer networks use protocols based on packet-mode transmission. A network packet is a formatted unit of data carried by a packet-switched network.

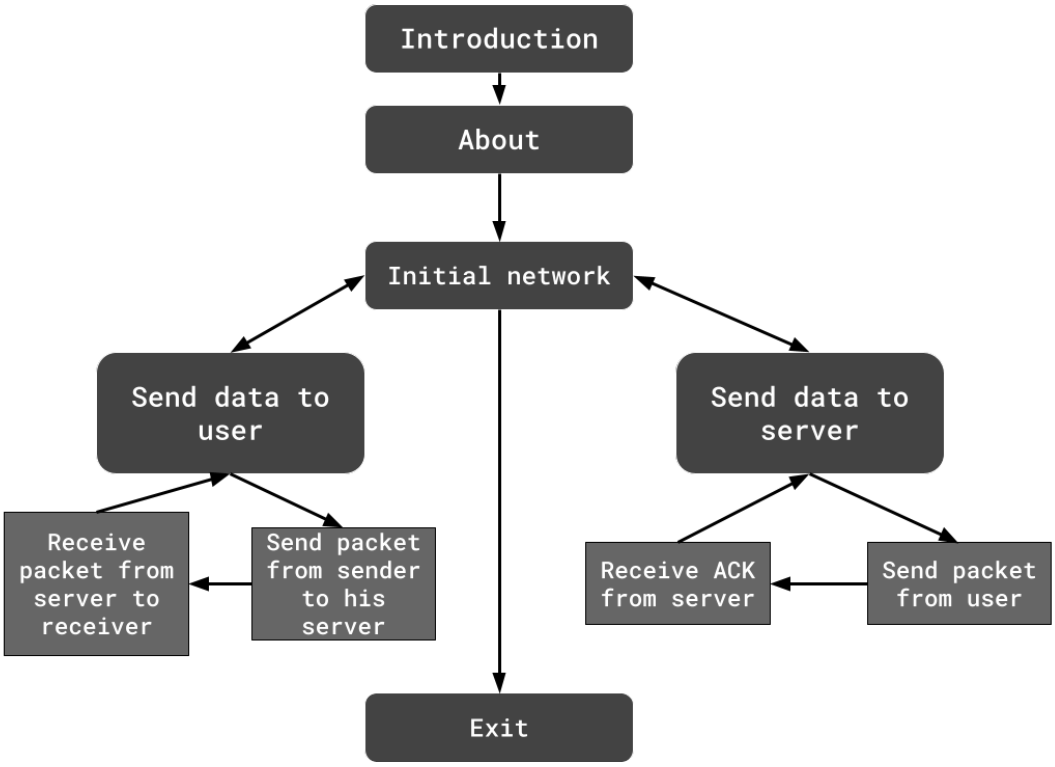
Let us consider a simple network with servers and users. Here, user can transfer packets to the server, followed by an acknowledgment (ACK) from the server itself. Each packet present in network is unique. To demonstrate this property, we have the packet colour as the colour of the sender itself. Each user is assigned to a dedicated server. Furthermore, user can send data to other users present in the network. The users can be present across different servers. To achieve this transfer of data, first sender sends the packets to his dedicated server. Followed by this, receiver will download this packet from his server. To mark the packets sent from certain user to another, the packet colour is set to the colour of sender. Additional properties such as heavy data rate is demonstrated by increased speed of packet transfer between 2 entities when the demand for inflow/outflow of packets is high between that link.

The data transfer between user and server is TCP/IP network. It is often used in real life when ACK is important to conform if packet is received or not, e.g., Upload of data to cloud. The data transfer between 2 users represent UDP network, where the ACK is not necessary and all data is sent in form of packets assuming no data loss has occurred. This type of network is often used in real life data transfer like Video streaming online.

IMPLEMENTATION

1. Work-Flow:

The working flow of the project is show in below figure:



1. Introduction: This consists of details about the project and the team.
2. About: Here, the instructions on how the project can be used, is mentioned.
3. Initial network: We have constructed the basic network with 2 servers and 3 clients. Server 1 has User 1 and User 2 as its dedicated users. Server 2 is dedicated is user 3.
   Users can establish connection with their server in TCP/IP format and send packets between them. Furthermore, users can also send packets to other users present in the network.

2. Keyboard/Mouse Functions:

User can react with system with the help of keyboard and mouse action. These actions are associated with certain functionality in the program. For example, by pressing the key "N" on keyboard you can go next, similarly "B" for back and "E" for exit. In main action platform right click on mouse gives you a menu of operation you will be able to perform on the initial network.

There exists a variable "*msgCount"* with initial value as 0. This value will decide on which state our program will be, e.g., when msgCount = 6, initially network is built. Hence, by incrementing or decrementing this value will let us go to next or previous slide/state. Therefore, when key "N" is clicked, this value is incremented and when key "B" is clicked, this value is decremented. Like so when the msgCount value is 7, the state of program is to exit. Hence, at this state if key "E" is clicked, the program is terminated/exited.

```
void keyboard (unsigned char key, int _, int __) {
    if key is 'N' or 'n' {
         increment msgCount;
    }
    else if key is 'B' or 'b' {
         decrement msgCount;
    }
    else if key is 'E' or 'e' and msgCount is 7 {
         exit ();
    }
}
```

The menu that displays on right mouse button click should only be displayed when the msgCount is 6. Hence, when the msgCount value becomes 6, we can link the menu with right mouse button.

```
If msgCount is 6 {
    glutAttachMenu (GLUT_RIGHT_BUTTON);
}
```

At every state, set of messages are displayed using OpenGL functions.

3. Data transfer between Client and Server:

When the basic network is constructed, the network has set of users and each user has a server allocated to him (some users share same server as well). One of the illustrated functionalities was allowing the user to transit packets of data with server and receive ACK packet for each packet he sends. This will demonstrate the basic working of TCP/IP network protocol.

This service should be available only after the basic network is built. On selecting this option, first a link is established between user and server. After the link is established, packets from user are transmitted towards server. As the packet leaves the user, a packet from server is also deployed into the network which transmits towards user. The link is represented by a line between user and server.

```
void drawLine (int x1, int y1, int x2, int y2, int thickness) {
    line from (x1, y1) to (x2, y2) with thickness (thickness);
}
```

The packet is built as a rectangle with white border and colour as the user colour or server colour depending on from where the packet is traveling.

```
void packet (int x, int y, int r, int g, int b) {
    1 px white border at (x, y), (x+10, y+10);
    rectangle with colour (r, g, b) (x, y), (x+10, y+10);
}
```

This packet will travel from sender to receiver. This is done by redrawing packet in new positions which is a bit ahead then the previous position at every frame till the packet reaches the receiver.

```
For ‘move’ from sender position to receiver position {
    packet (x+move, y, r, b, g);
}
```

4. Data transfer between Client and Client:

The users present in the network can transmit packets amongst them. That is one user can send messages to another user. Unlike Client-Server link, this follows UDP protocol. This process can be divided into two stages. First sender will have to send the packets to his server. Second receiver will download the packets from his server.

For sender to send data to server, first the link to server should be established. This is represented by a link drawn between user and server. Once the link is established, the user will send the packets to server. This is achieved in same fashion as we discussed in previous section.

```
drawLine (sender (x, y), server (x, y), thickness);
for ‘move’ from sender position to server position {
    packet (sender (x+move, y), r, b, g);
}
```

For receiver to download data from server, first a link should be established between receiver and server. After that packets will move/download from server to receiver.

```
drawLine (server (x, y), receiver (x, y), thickness);
for ‘move’ from server position to receiver position {
    packet (server (x+move, y), r, b, g);
}
```

CONCLUSION

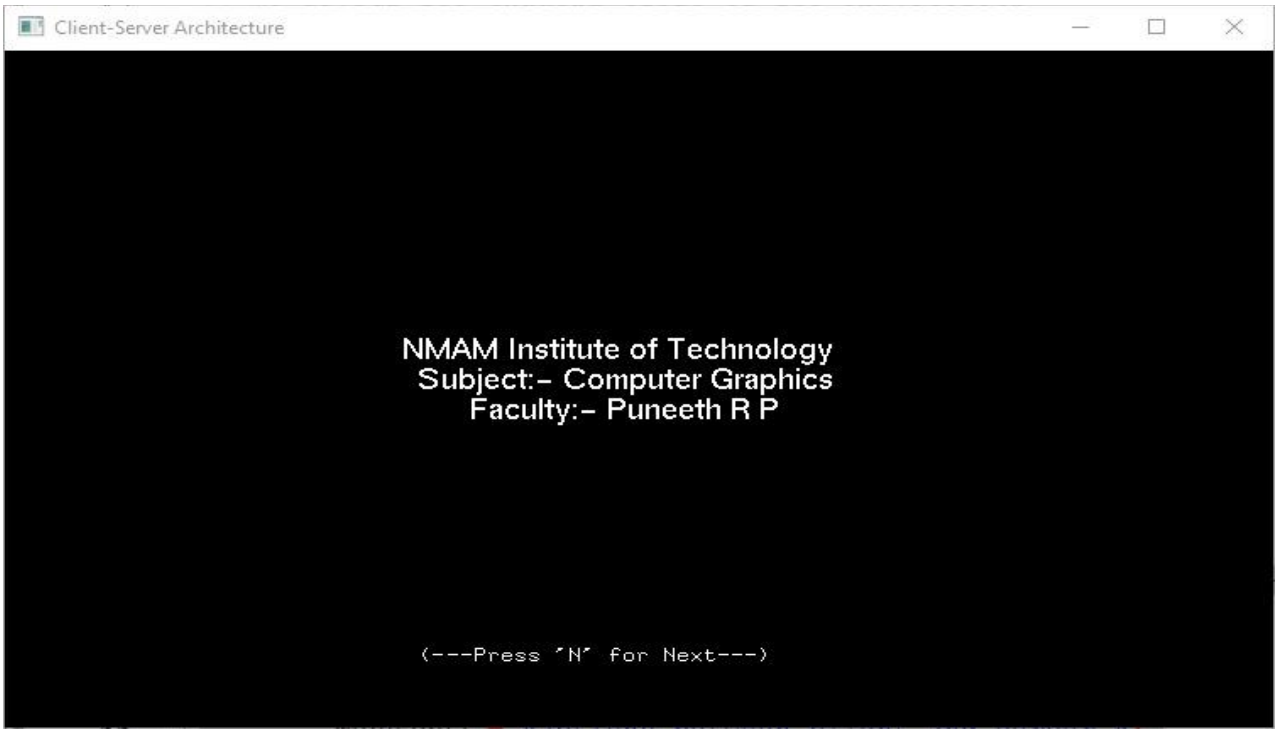
We have successfully constructed and demonstrated the basic network and working of client-server architecture. We have demonstrated the following functionalities:

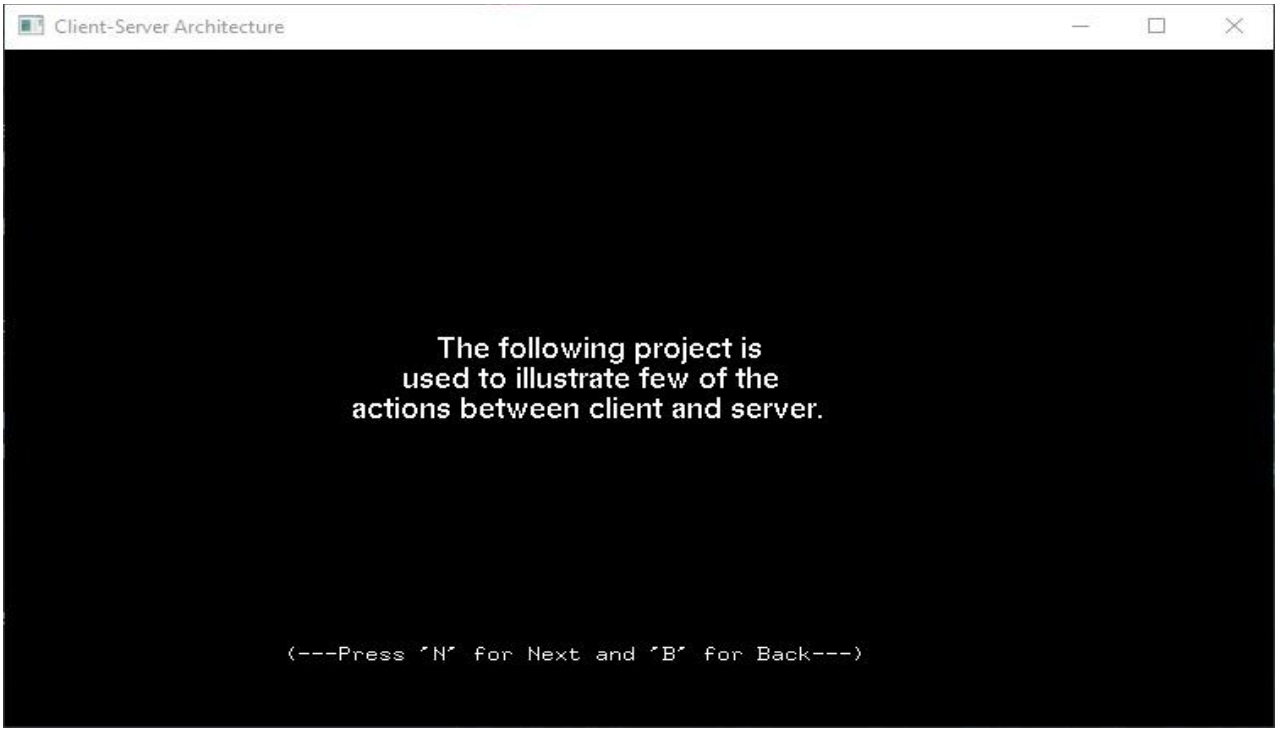
1. **Transfer of packets between Client and Server in TCP/IP fashion.**
2. **Transfer of data between multiple users present in the network in UDP fashion.**

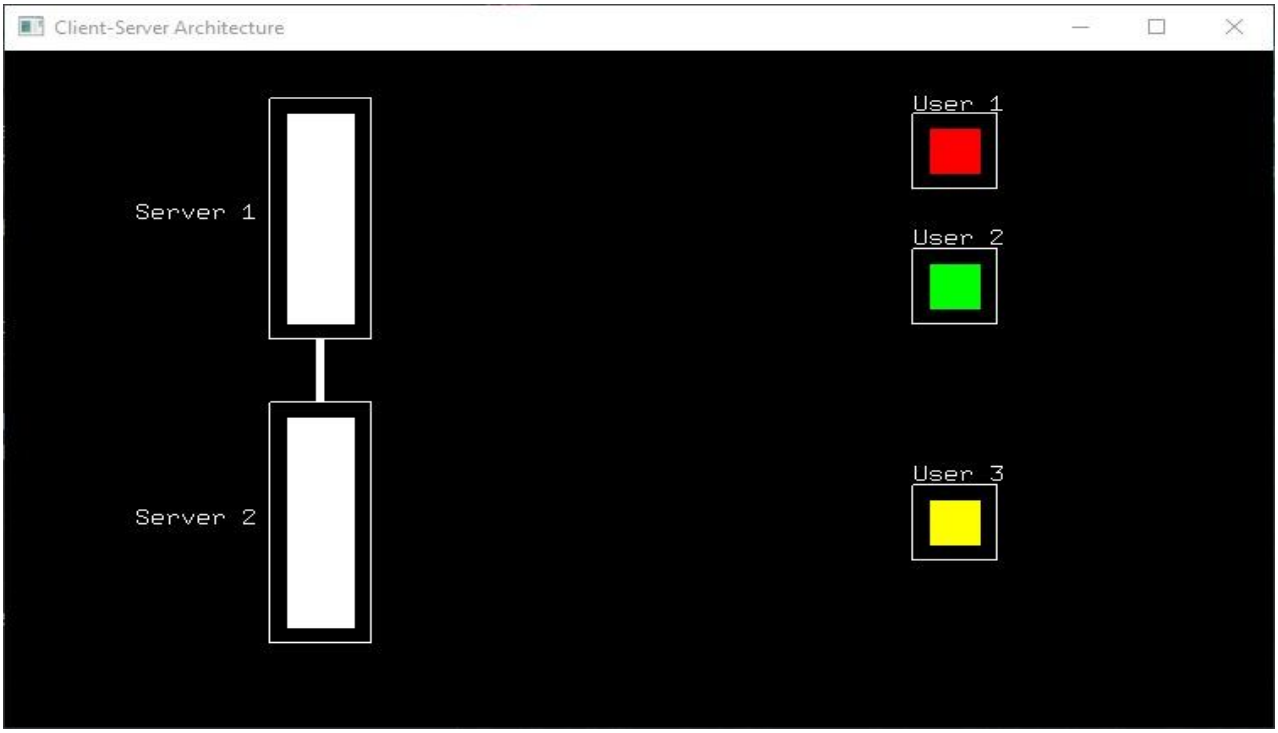
REFERENCES

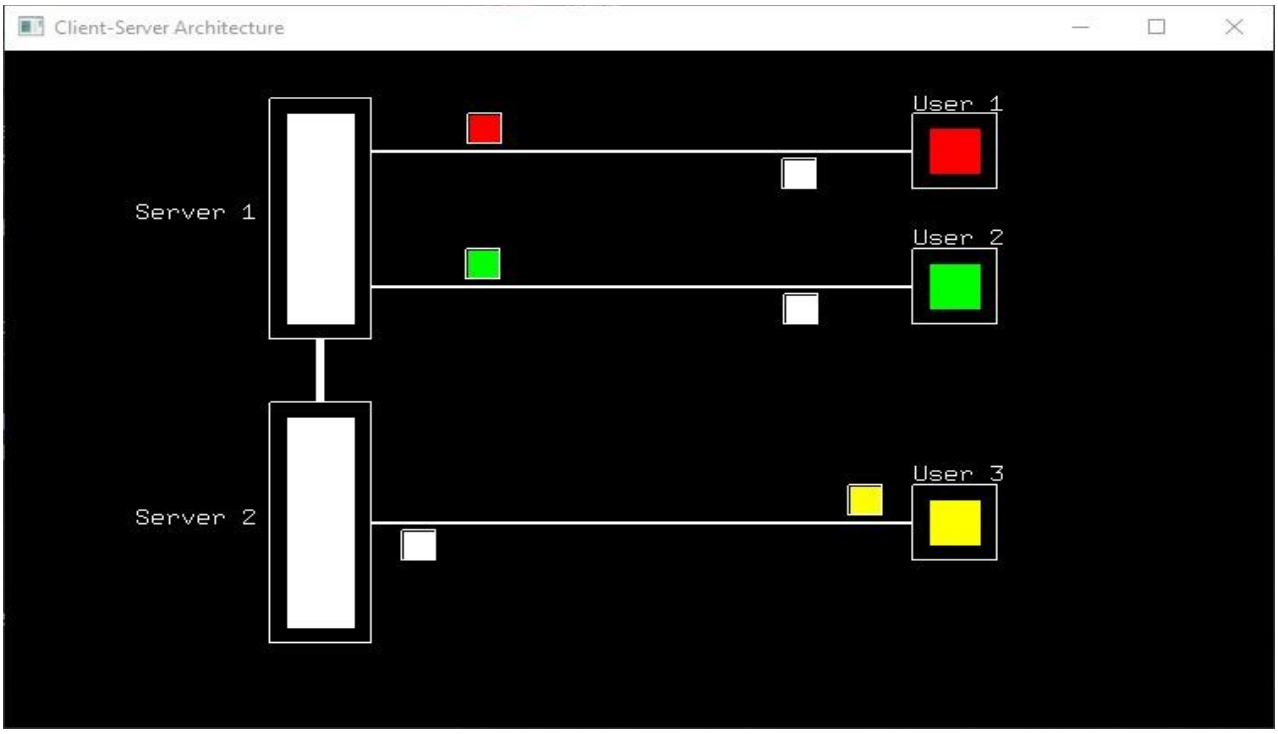
- https://www.opengl.org/resources/
- https://www.khronos.org/registry/OpenGL-Refpages/gl4/
- https://www.khronos.org/opengl/
- https://learnopengl.com/Getting-started/OpenGL
- https://en.wikipedia.org/wiki/Client%E2%80%93server_model
- https://en.wikipedia.org/wiki/Transmission_Control_Protocol
- https://en.wikipedia.org/wiki/User_Datagram_Protocol
- https://stackoverflow.com
- https://www.youtube.com/

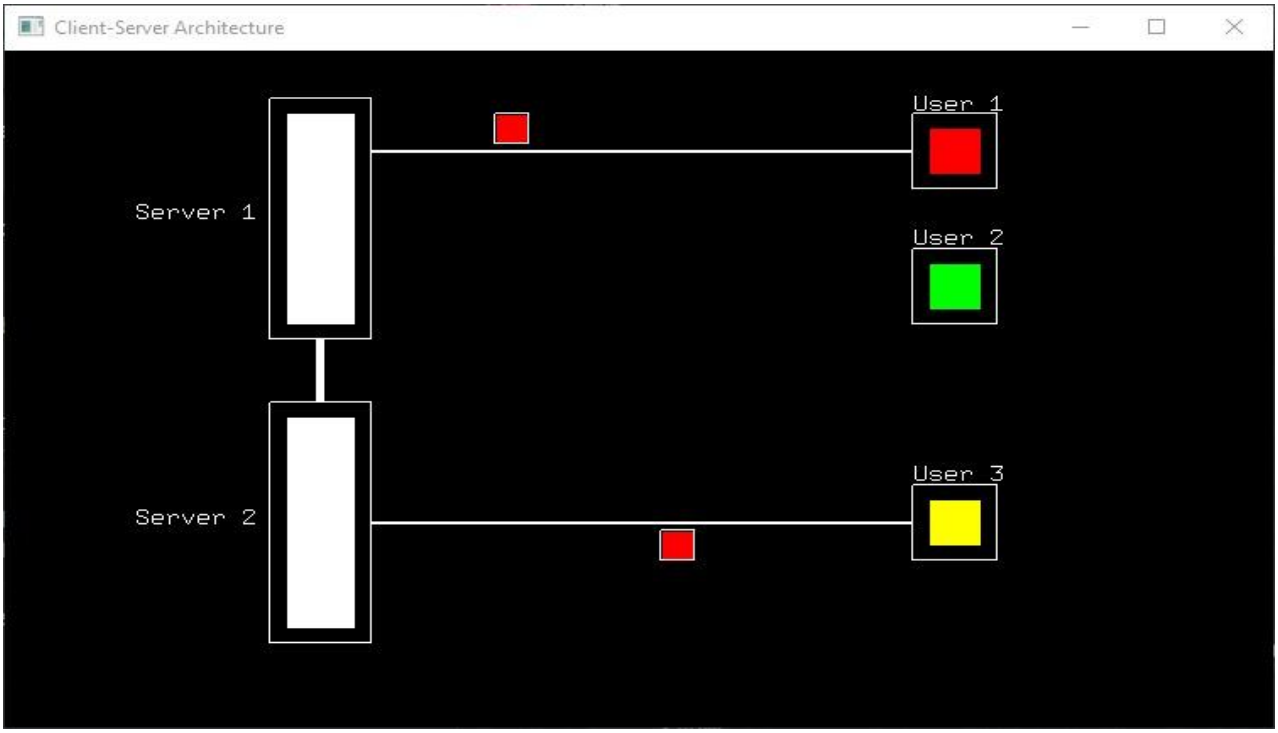
APPENDIX


Client-Server Architecture
NMAM Institute of Technology
Subject:- Computer Graphics
Faculty:- Puneeth R P
(---Press 'N' for Next---)


Client-Server Architecture
The following project is
used to illustrate few of the
actions between client and server.
(---Press 'N' for Next and 'B' for Back---)


Client-Server Architecture
Server 1
Server 2
User 1
User 2
User 3


Client-Server Architecture
Server 1
Server 2
User 1
User 2
User 3


Client-Server Architecture
User 1
Server 1
User 2
User 3
Server 2