

MULTIPLAYER NETWORK GAME

Summer Project, 2011
Programming Club
Science and Technology Council
IIT Kanpur

By:

Rahul Arora

Ramena Gopi

Rohit Gupta

Mentor: Adarsh Jagannatha

1. Aims:

To develop a multiplayer network game with 3D graphics.

2. Our Choices:

2.1: Why C?

- As opposed to other languages like Java, Python, where GUI development is easier, C is more optimized. It is also quite faster than other languages.
- Also socket programming which is the core part of our networking can also be easily implemented in C.

2.2: Why OpenGL Library?

- Our first option was using Game Engines as they are easy to implement & had pre-defined functions for many of the features. But it had some problems like importing of a map. Also we couldn't get to learn more if we used a game engine. We had also looked into other libraries of C like Allegro & AllegroGL. Finally we decided to build the game from scratch using the basic OpenGL library.

3. Implementing the Game:

- The Game currently runs in Linux with OpenGL & Glut packages installed. These can be simply installed by typing the following command at the linux terminal:

```
sudo apt-get install freeglut3 freeglut3-dev
```

- GUI development:

Basic OpenGL functions are used for creating game features like lighting, drawing the user car & remote car, drawing ground, world etc.

An example as to how we created lighting in the map is illustrated by the following code:

```
void createLights() {
GLfloat light0Pos[] = {0.0,1.0,1.0,0.0};
GLfloat light0Diffuse[] = {0.3,0.3,0.3,1.0};
GLfloat light0Specular[] = {1.0,1.0,0.0,1.0};
GLfloat ambLight[] = {0.0,0.0,0.0,1.0};
glLightModelfv(GL_LIGHT_MODEL_AMBIENT,ambLight);
glLightfv(GL_LIGHT0,GL_POSITION,light0Pos);
glLightfv(GL_LIGHT0,GL_SPECULAR,light0Specular);
glLightfv(GL_LIGHT0,GL_DIFFUSE,light0Diffuse);
glLightfv(GL_LIGHT1,GL_SPECULAR,light0Specular);
}
```

```
glLightfv(GL_LIGHT1, GL_AMBIENT, light0Diffuse);
glLightModeli(GL_LIGHT_MODEL_LOCAL_VIEWER, GL_TRUE);
glLightfv(GL_LIGHT1, GL_POSITION, light1Pos);
glLightfv(GL_LIGHT1, GL_SPOT_DIRECTION, light1Direction);
glLightf(GL_LIGHT1, GL_SPOT_CUTOFF, 35.0);
glLightf(GL_LIGHT1, GL_SPOT_EXPONENT, 50.0);
glLightf(GL_LIGHT1, GL_CONSTANT_ATTENUATION, 0.3);
glEnable(GL_LIGHTING);
glEnable(GL_LIGHT0);
glEnable(GL_LIGHT1);
glEnable(GL_DEPTH_TEST);
glEnable(GL_AUTO_NORMAL);
glEnable(GL_BLEND);
glBlendFunc(GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA);
}
```

Similarly other OpenGL functions like `glPushMatrix`, `glPopMatrix`, `glVertex3fv`, `glMaterialfv`, `glNormalfv`, `glRotatef`, `glTranslatef`, etc. were used to create the cars & the rest of the world in the game.

Only three parameters namely r , α and ϕ were used to control the user car as well as the remote car.

- Networking:

The game used sockets to communicate with the other remote computer. Initially we used TCP sockets. But due to their blocking nature

the game had to wait for the packets to arrive. So we changed our sockets to UDP. Due to its non-blocking nature, the packets were sent & received independently.

On a keypress event, the corresponding r , α and ϕ parameters were changed & their values were passed as a structure via sockets to the other remote computer. We used port number 5000 to communicate between the computers. The ip address of the other remote machine was required to be entered by the user on starting the game. IP validation was also done to check whether its a valid ip or not.

4. Further Improvement

In its current state the program serves only as a demo of the features of OpenGL and Socket Programming. The full potential of these tools can be realised with the help of creatively oriented designer-cum-programmer.

There is also a scope of making it run in different platforms. Presently it runs in linux. It requires a small modification to be run in Windows. Similarly it can be explored about how to run it in other OS

like Mac, etc.

5.References

- The OpenGL Programming Guide 5th Edition:The Official Guide to Learning OpenGL Version 2.1 a.k.a The Redbook
url:<http://fly.cc.fer.hr/~unreal/theredbook/>
- Beej's Guide to Network Programming
url:<http://beej.us/guide/bgnet/>

6.Acknowledgements

- Our Mentor:Adarsh Jagannatha
- Other Programming Club Coordinators: Ankit Mahato and Ankesh Kumar Singh
- SnT Council and Students Gymkhana