CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION TO COMPUTER GRAPHICS

Computer Graphics is concerned with all aspects of producing pictures or images using a computer. Graphics provides one of the most natural means of communicating within a computer, since our highly developed 2D and 3D pattern-recognition abilities allow us to perceive and process pictorial data rapidly and effectively. Interactive computer graphics is the most important means of producing pictures since the invention of photography and television.

Applications of Computer Graphics

1. Display of information
2. Design
3. Simulation and animation
4. User interfaces

The Graphics Architecture

Graphics Architecture can be made up of seven components:

1. Display processors
2. Pipeline architectures
3. The graphics pipeline
4. Vertex processing
5. Clipping and primitive assembly
6. Rasterization
7. Fragment processing

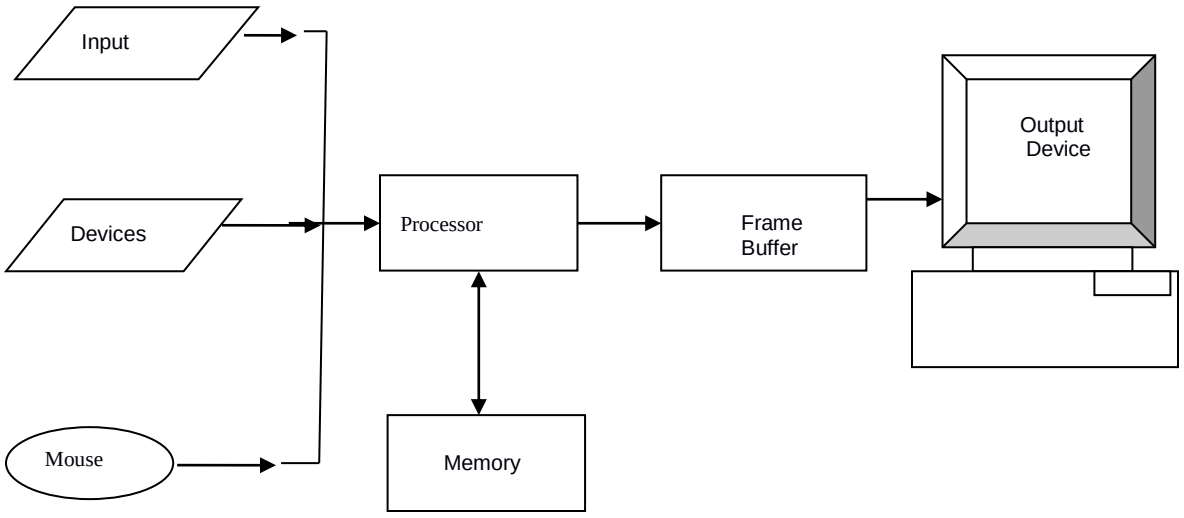

Figure 1.1 Components of Graphics Architecture and their working.

1.2 INTRODUCTION TO OPENGL

OpenGL is software used to implement computer graphics. The structure of OpenGL is similar to that of most modern APIs including Java 3D and DirectX. OpenGL is easy to learn, compared with other.

APIs are nevertheless powerful. It supports the simple 2D and 3D programs. It also supports the advanced rendering techniques. OpenGL API explains following 3 components

1. Graphics functions
2. Graphics pipeline and state machines
3. The OpenGL interfaces

There are so many polygon types in OpenGL like triangles, quadrilaterals, strips and fans.

There are 2 control functions, which will explain OpenGL through,

1. Interaction with window system
2. Aspect ratio and view ports

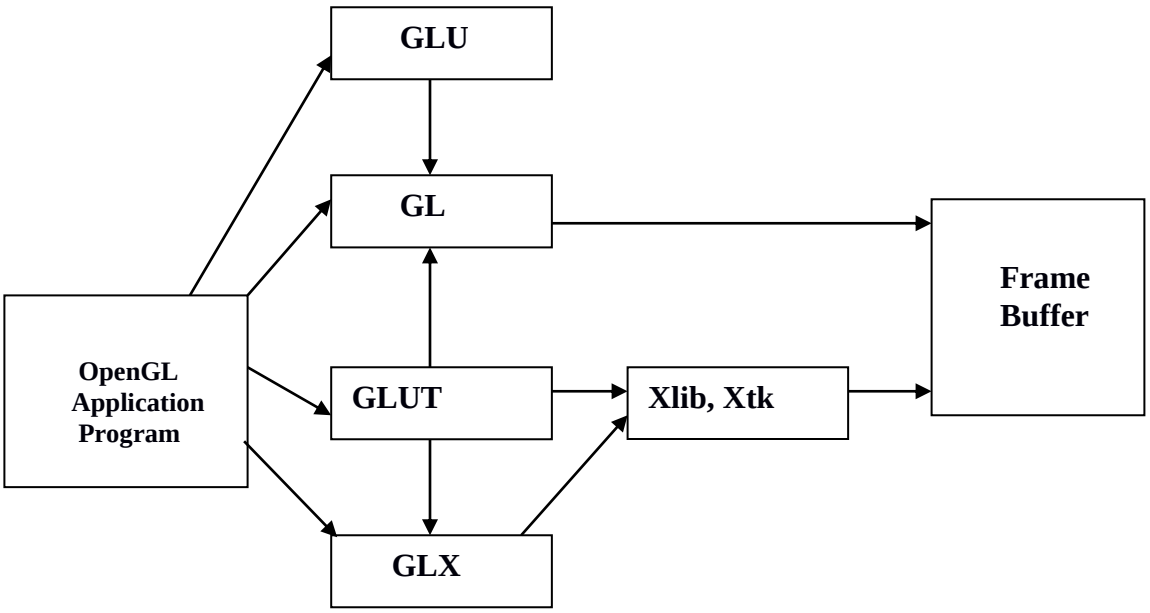

Figure 1.2 OpenGL Library organization

Most implementations of OpenGL have a similar order of operations, a series of processing stages called the OpenGL rendering pipeline. This ordering, as shown in Figure 1.2, is not a strict rule of how OpenGL is implemented but provides a reliable guide for predicting what OpenGL will do. The following diagram shows the assembly line approach, which OpenGL takes to process data. Geometric data (vertices, lines, and polygons) follow the path through the row of boxes that includes evaluators and per-vertex operations, while pixel data (pixels, images, and bitmaps) are treated differently for part of the process. Both types of data undergo the same final steps (rasterization and per-fragment operations) before the final pixel data is written into the frame buffer.

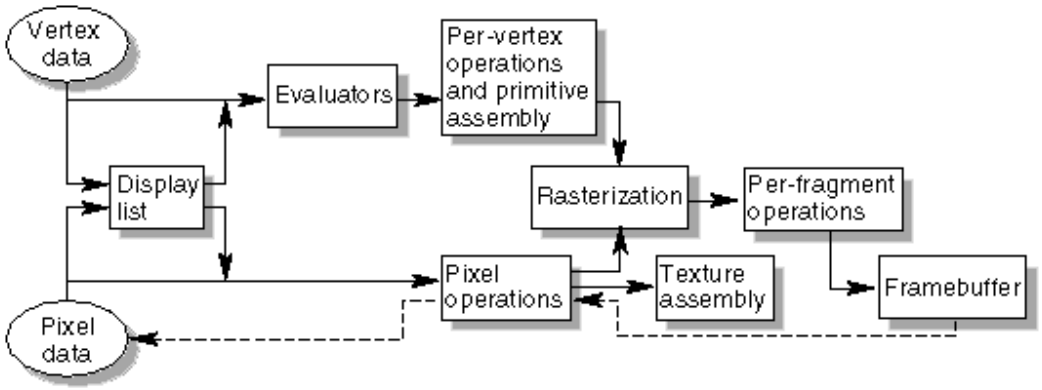

Figure 1.3 OpenGL Order of Operations

CHAPTER 2

REQUIREMENTS SPECIFICATION

2.1 SOFTWARE REQUIREMENTS

- Operating system – Windows XP
- Microsoft Visual Studio 2012
- OPENGL library files – GL, GLU, GLUT
- Language used is C/C++

2.2 HARDWARE REQUIREMENTS

- Processor - Pentium Pro
- Memory - 128MB RAM
- 20GB Hard Disk Drive
- Mouse or other pointing device
- Keyboard
- Display device

.

CHAPTER 3

SYSTEM DEFINITION

The aim of the project is to draw a **skating stickman** and control his motions using keyboard interrupts and detecting collision between stickman and obstacles.

The projectdraws askating stickmanwith a backdrop. Stickman will skate along the road. The player has to ensure that the stickman should jump across the interrupting obstacles on the road without colliding with the obstacles. For each jump across the obstacles score point the player get increases. The speed of the stickman will increases along with the time. Game ends when the player failed to jump across the obstacles successfully.

3.1 USER DEFINED FUNTION

- The **display()** function to display the whole object which is created using the various OpenGL functions.
- The **background()** function renders buildings in the background of stickman.
- The **window()** function renders the windows on the buildings.
- The **moon()** function renders the full moon.
- The **stickman()** function renders the stickman on skateboard.
- The **keyboard()** function respond for the input from the keyboard and used to control the movement of stickman and to change the mode of playing the game (day or night).
- The **mouse()** function changes the color of the stickman.

3.2 FLOWCHART

The figure shows the flow diagram of Skating Stickman

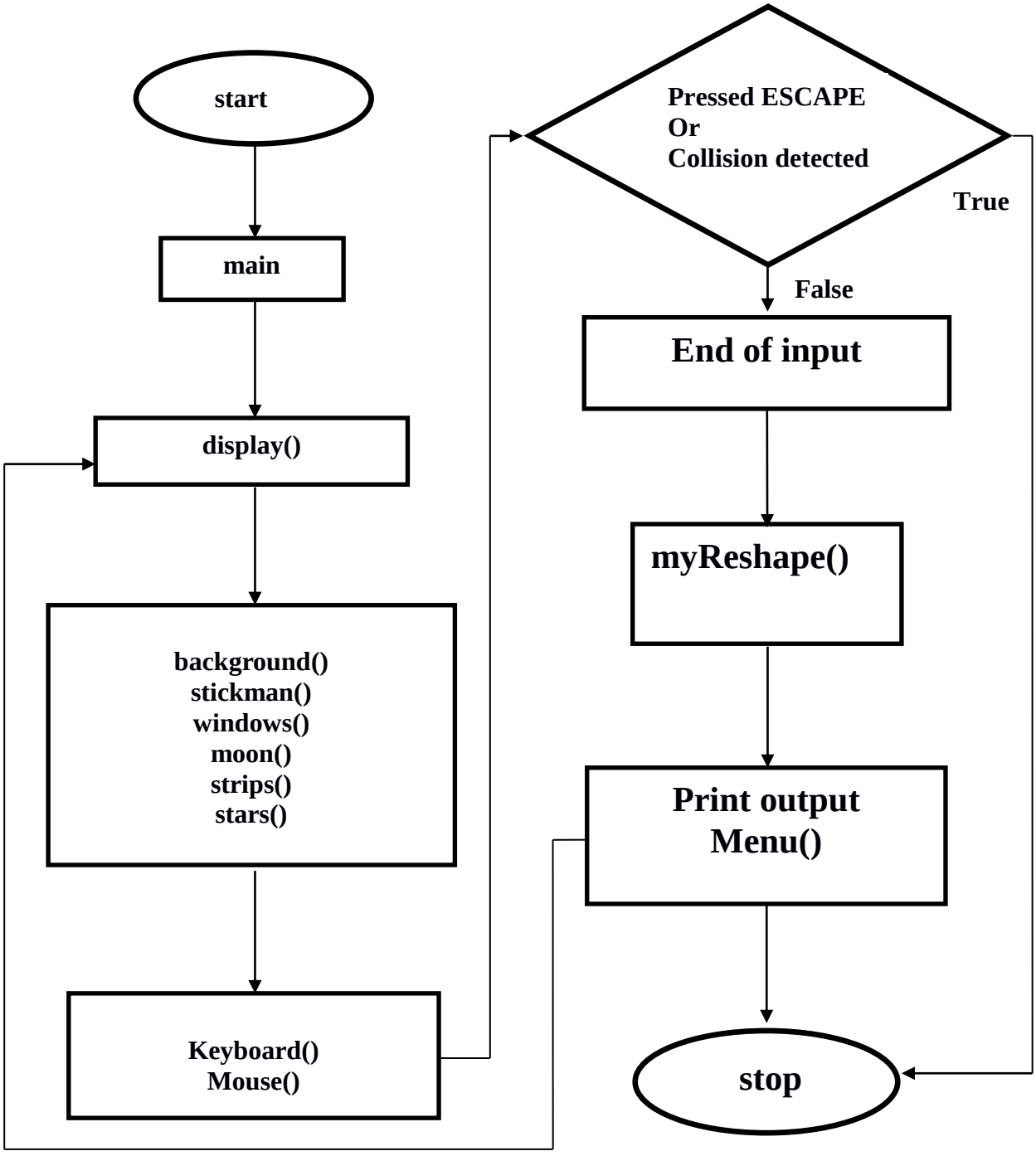

Figure 3.1 Flow Control of the program

CHAPTER 4

IMPLEMENTATION

4.1 SOURCE CODE

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<windows.h>>
#include <glut.h>
#include <gl.h>
#include <glu.h>

#define ESCAPE 27
float a1=0.5;
float escape=-1.4;
float rtri =2.0f;
float v=0,up=0;
float rquad = 0.0f;
float a=0,x=-
2.0,y,R=1,G=1,B=1,Rb=0.25,Gb=0.25,Bb=0.5f,Rb1=1,Gb1=1,Bb1=1,Rb2=0.5,Gb2=0.5,Bb2=1,
m1=1,m2=1,m3=0.3;
float speed=40,ibigx=0,ibigy=0,hbigx=0,hbigy=0,pbigx=0,pbigy=0;
const float z=45.5;
int window,mov=0,day=1;
int stop=0,score=0,ok=1,help=0,info=0,okinfo=0,okhelp=0,firstspace=0;
/*Text*/
void    Text0(){
        glBegin(GL_LINE_LOOP);
        glVertex3f(1,1,0);
        glVertex3f(-1,1,0);
        glVertex3f(-1,-1,0);/*draw 0*/
```

```
        glVertex3f(1,-1,0);
        glEnd();

}
void   Text1(){
        glBegin(GL_LINE_LOOP);
        glVertex3f(0,1,0);
        glVertex3f(0,-1,0);
        glEnd();
}
void      Text2(){
        glBegin(GL_LINES);
        glVertex3f(-1,1,0);
        glVertex3f(1,1,0);
        glEnd();
        glBegin(GL_LINES);
        glVertex3f(1,1,0);
        glVertex3f(1,0,0);
        glEnd();
        glBegin(GL_LINES);
        glVertex3f(1,0,0);
        glVertex3f(-1,0,0);
        glEnd();
        glBegin(GL_LINES);
        glVertex3f(-1,0,0);
        glVertex3f(-1,-1,0);
        glEnd();
        glBegin(GL_LINES);
        glVertex3f(-1,-1,0);
        glVertex3f(1,-1,0);
        glEnd();
```

```
}
void	Text3(){
	glBegin(GL_LINES);
	glVertex3f(-1,1,0);
	glVertex3f(1,1,0);
	glEnd();
	glBegin(GL_LINES);
	glVertex3f(1,1,0);
	glVertex3f(1,0,0);
	glEnd();
	glBegin(GL_LINES);
	glVertex3f(1,0,0);
	glVertex3f(-1,0,0);
	glEnd();
	glBegin(GL_LINES);
	glVertex3f(1,0,0);
	glVertex3f(1,-1,0);
	glEnd();
	glBegin(GL_LINES);
	glVertex3f(1,-1,0);
	glVertex3f(-1,-1,0);
	glEnd();

}
void	Text4(){
	glBegin(GL_LINES);
	glVertex3f(1,1,0);
	glVertex3f(1,-1,0);
	glEnd();
	glBegin(GL_LINES);
```

```
        glVertex3f(-1,0,0);
        glVertex3f(1.5,0,0);
        glEnd();
        glBegin(GL_LINES);
        glVertex3f(1,1,0);
        glVertex3f(-1,0,0);
        glEnd();
}
void    Text5(){
        glBegin(GL_LINES);
        glVertex3f(1,1,0);
        glVertex3f(-1,1,0);
        glEnd();

        glBegin(GL_LINES);
        glVertex3f(-1,1,0);
        glVertex3f(-1,0,0);
        glEnd();

        glBegin(GL_LINES);
        glVertex3f(-1,0,0);
        glVertex3f(1,0,0);
        glEnd();

        glBegin(GL_LINES);
        glVertex3f(1,0,0);
        glVertex3f(1,-1,0);
        glEnd();

        glBegin(GL_LINES);
        glVertex3f(1,-1,0);
```

```
        glVertex3f(-1,-1,0);
        glEnd();

}
void      Text6(){
        glBegin(GL_LINES);
        glVertex3f(1,1,0);
        glVertex3f(-1,1,0);
        glEnd();

        glBegin(GL_LINES);
        glVertex3f(-1,1,0);
        glVertex3f(-1,0,0);
        glEnd();

        glBegin(GL_LINES);
        glVertex3f(-1,0,0);
        glVertex3f(1,0,0);
        glEnd();

        glBegin(GL_LINES);
        glVertex3f(1,0,0);
        glVertex3f(1,-1,0);
        glEnd();

        glBegin(GL_LINES);
        glVertex3f(1,-1,0);
        glVertex3f(-1,-1,0);
        glEnd();
        glBegin(GL_LINES);
        glVertex3f(-1,0,0);
```

```
        glVertex3f(-1,-1,0);
        glEnd();

}
void    Text7(){
        glBegin(GL_LINES);
        glVertex3f(-1,1,0);
        glVertex3f(1,1,0);
        glEnd();
        glBegin(GL_LINES);
        glVertex3f(1,1,0);
        glVertex3f(1,-1,0);
        glEnd();

}
void    Text8(){
        glBegin(GL_LINE_LOOP);
        glVertex3f(1,1,0);
        glVertex3f(-1,1,0);
        glVertex3f(-1,-1,0);/*draw 0*/
        glVertex3f(1,-1,0);
        glEnd();
        glBegin(GL_LINES);
        glVertex3f(1,0,0);
        glVertex3f(-1,0,0);
        glEnd();

}
void    Text9(){
        glBegin(GL_LINE_LOOP);
        glVertex3f(1,1,0);/*draw 9*/
```

```
        glVertex3f(-1,1,0);
        glVertex3f(-1,0,0);
        glVertex3f(1,0,0);
        glVertex3f(1,-1,0);
        glEnd();

}
void    TextA(){
        glBegin(GL_LINES);
        glVertex3f(1.0,-1,0);
        glVertex3f(0,1,0);
        glEnd();
        glBegin(GL_LINES);
        glVertex3f(-1,-1,0);
        glVertex3f(0,1,0);
        glEnd();
        glBegin(GL_LINES);
        glVertex3f(0.2,0,0);
        glVertex3f(-0.2,0,0);
        glEnd();
}
void    TextB(){
        Text8();
}
void    TextC(){
        glBegin(GL_LINES);
        glVertex3f(1,1,0);
        glVertex3f(-1,1,0);
        glEnd();
        glBegin(GL_LINES);
        glVertex3f(-1,1,0);
```

```
        glVertex3f(-1,-1,0);
        glEnd();
        glBegin(GL_LINES);
        glVertex3f(1,-1,0);
        glVertex3f(-1,-1,0);
        glEnd();

}
void    TextD(){
        glBegin(GL_LINES);
        glVertex3f(1,1,0);
        glVertex3f(1,-1,0);
        glEnd();
        glBegin(GL_LINES);
        glVertex3f(-1,1,0);
        glVertex3f(1,1,0);
        glEnd();
        glBegin(GL_LINES);
        glVertex3f(-1,-1,0);
        glVertex3f(1,-1,0);
        glEnd();
        glBegin(GL_LINES);
        glVertex3f(-0.5,1,0);
        glVertex3f(-0.5,-1,0);
        glEnd();
}
void    TextF(){
        glBegin(GL_LINES);
        glVertex3f(-1,1,0);
        glVertex3f(-1,-1,0);
        glEnd();
```

```
        glBegin(GL_LINES);
        glVertex3f(-1,1,0);
        glVertex3f(1,1,0);
        glEnd();
        glBegin(GL_LINES);
        glVertex3f(-1,0,0);
        glVertex3f(1,0,0);
        glEnd();
}
void    TextE(){

        TextF();
        glBegin(GL_LINES);
        glVertex3f(-1,-1,0);
        glVertex3f(1,-1,0);
        glEnd();
}

void    TextG(){
        TextC();
        glBegin(GL_LINES);
        glVertex3f(1,-1,0);
        glVertex3f(1,0,0);
        glEnd();
        glBegin(GL_LINES);
        glVertex3f(0,0,0);
        glVertex3f(1,0,0);
        glEnd();
}
void    TextH(){
        glBegin(GL_LINES);
```

```
        glVertex3f(-1,1,0);
        glVertex3f(-1,-1,0);
        glEnd();
        glBegin(GL_LINES);
        glVertex3f(-1,0,0);
        glVertex3f(1,0,0);
        glEnd();
        glBegin(GL_LINES);
        glVertex3f(1,1,0);
        glVertex3f(1,-1,0);
        glEnd();
}
void    TextI(){
        glBegin(GL_LINES);
        glVertex3f(0,1,0);
        glVertex3f(0,-1,0);
        glEnd();
        glBegin(GL_LINES);
        glVertex3f(-1,1,0);
        glVertex3f(1,1,0);
        glEnd();
        glBegin(GL_LINES);
        glVertex3f(-1,-1,0);
        glVertex3f(1,-1,0);
        glEnd();
}
void    TextJ(){
        glBegin(GL_LINES);
        glVertex3f(0,1,0);
        glVertex3f(0,-1,0);
        glEnd();
```

```
        glBegin(GL_LINES);
        glVertex3f(-1,1,0);
        glVertex3f(1,1,0);
        glEnd();
        glBegin(GL_LINES);
        glVertex3f(-1,-1,0);
        glVertex3f(0,-1,0);
        glEnd();
        glBegin(GL_LINES);
        glVertex3f(-1,-1,0);
        glVertex3f(-1,0,0);
        glEnd();
}
void    TextK(){
        Text1();
        glBegin(GL_LINES);
        glVertex3f(0,0,0);
        glVertex3f(1.5,1.2,0);
        glEnd();
        glBegin(GL_LINES);if(x>1121&&x<1251&&y>17&&y<42)
        glVertex3f(0,0,0);
        glVertex3f(1.5,-1.2,0);
        glEnd();
}
void    TextL(){
        glBegin(GL_LINES);
        glVertex3f(-1,1,0);
        glVertex3f(-1,-1,0);
        glEnd();
        glBegin(GL_LINES);
        glVertex3f(-1,-1,0);
```

```
        glVertex3f(1,-1,0);
        glEnd();
}
void    TextM(){
        glBegin(GL_LINES);
        glVertex3f(-1,-1,0);
        glVertex3f(-1,1,0);
        glEnd();
        glBegin(GL_LINES);
        glVertex3f(-1,1,0);
        glVertex3f(0,0,0);
        glEnd();
        glBegin(GL_LINES);
        glVertex3f(0,0,0);
        glVertex3f(1,1,0);
        glEnd();
        glBegin(GL_LINES);
        glVertex3f(1,1,0);
        glVertex3f(1,-1,0);
        glEnd();
}
void    TextN(){
        glBegin(GL_LINES);
        glVertex3f(-1,-1,0);
        glVertex3f(-1,1,0);
        glEnd();
        glBegin(GL_LINES);
        glVertex3f(-1,1,0);
        glVertex3f(1,-1,0);
        glEnd();
        glBegin(GL_LINES);
```

```
        glVertex3f(1,-1,0);
        glVertex3f(1,1,0);
        glEnd();
}
void TextO()
{
        Text0();
}
void  TextP(){
        glBegin(GL_LINES);
        glVertex3f(-1,1,0);
        glVertex3f(-1,-1,0);
        glEnd();
        glBegin(GL_LINES);
        glVertex3f(1,1,0);
        glVertex3f(-1,1,0);
        glEnd();
        glBegin(GL_LINES);
        glVertex3f(1,1,0);
        glVertex3f(1,0,0);
        glEnd();
        glBegin(GL_LINES);
        glVertex3f(1,0,0);
        glVertex3f(-1,0,0);
        glEnd();
}
void  TextQ(){
        Text0();
        glBegin(GL_LINES);
        glVertex3f(1.5,-1.5,0);
        glVertex3f(0,0,0);
```

```
        glEnd();
}
void      TextR(){
   TextP();
        glBegin(GL_LINES);
        glVertex3f(1,-1,0);
        glVertex3f(0,0,0);
        glEnd();
}
void      TextS(){
        Text5();
}
void      TextT(){
        Text1();
        glBegin(GL_LINES);
        glVertex3f(-1,1,0);
        glVertex3f(1,1,0);
        glEnd();
}
void      TextU(){
        glBegin(GL_LINES);
        glVertex3f(-1,1,0);
        glVertex3f(-1,-1,0);
        glEnd();
        glBegin(GL_LINES);
        glVertex3f(-1,-1,0);
        glVertex3f(1,-1,0);
        glEnd();
        glBegin(GL_LINES);
        glVertex3f(1,-1,0);
        glVertex3f(1,1,0);
```

```
        glEnd();
}
void      TextV(){
        glBegin(GL_LINES);
        glVertex3f(-1,1,0);
        glVertex3f(0,-1,0);
        glEnd();
        glBegin(GL_LINES);
        glVertex3f(0,-1,0);
        glVertex3f(1,1,0);
        glEnd();
}
void      TextW(){
        glRotatef(180,0,0,1);
        TextM();
}
void      TextX(){
        glBegin(GL_LINES);
        glVertex3f(-1,1,0);
        glVertex3f(1,-1,0);
        glEnd();
        glBegin(GL_LINES);
        glVertex3f(1,1,0);
        glVertex3f(-1,-1,0);
        glEnd();
}
void   TextY(){
        glBegin(GL_LINES);
        glVertex3f(0,-1,0);
        glVertex3f(0,0,0);
        glEnd();
```

```
        glBegin(GL_LINES);
        glVertex3f(0,0,0);
        glVertex3f(1,1,0);
        glEnd();
        glBegin(GL_LINES);
        glVertex3f(0,0,0);
        glVertex3f(-1,1,0);
        glEnd();
}
void  TextZ(){
        glBegin(GL_LINES);
        glVertex3f(-1,1,0);
        glVertex3f(1,1,0);
        glEnd();
        glBegin(GL_LINES);
        glVertex3f(1,1,0);
        glVertex3f(-1,-1,0);
        glEnd();
        glBegin(GL_LINES);
        glVertex3f(-1,-1,0);
        glVertex3f(1,-1,0);
        glEnd();
}
void TextEditor(GLfloat posx,GLfloat posy,GLfloat posz,GLfloat scalex,GLfloat scaley,GLfloat
scalez,GLfloat red,GLfloat green,GLfloat blue){
        glPushMatrix();
        glTranslatef(posx,posy,posz);
        glScalef(scalex,scaley,scalez);
        glColor3f(red,green,blue);
}
```

```
void MenuButtonText(GLfloat posx,GLfloat posy,GLfloat posz,GLfloat scalex,GLfloat
scaley,GLfloat scalez,GLfloat red,GLfloat green,GLfloat blue){
        glLoadIdentity();
        TextEditor(posx,posy,posz,scalex,scaley,scalez,red,green,blue);//Global
        TextG();
        glPopMatrix();
        TextEditor(posx+0.02,posy,posz,scalex,scaley,scalez,red,green,blue);
        TextL();
        glPopMatrix();
        TextEditor(posx+(0.02*2),posy,posz,scalex,scaley,scalez,red,green,blue);
        TextO();
        glPopMatrix();
        TextEditor(posx+(0.02*3),posy,posz,scalex,scaley,scalez,red,green,blue);
        TextB();
        glPopMatrix();
        TextEditor(posx+(0.02*4),posy,posz,scalex,scaley,scalez,red,green,blue);
        TextA();
        glPopMatrix();
        TextEditor(posx+(0.02*5),posy,posz,scalex,scaley,scalez,red,green,blue);
        TextL();
        glPopMatrix();

        TextEditor(posx+(0.02*7),posy,posz,scalex,scaley,scalez,red,green,blue);//Academy
        TextA();
        glPopMatrix();
        TextEditor(posx+(0.02*8),posy,posz,scalex,scaley,scalez,red,green,blue);
        TextC();
        glPopMatrix();
        TextEditor(posx+(0.02*9),posy,posz,scalex,scaley,scalez,red,green,blue);
        TextA();
        glPopMatrix();
```

```
TextEditor(posx+(0.02*10),posy,posz,scalex,scaley,scalez,red,green,blue);
TextD();
glPopMatrix();
TextEditor(posx+(0.02*11),posy,posz,scalex,scaley,scalez,red,green,blue);
TextE();
glPopMatrix();
TextEditor(posx+(0.02*12),posy,posz,scalex,scaley,scalez,red,green,blue);
TextM();
glPopMatrix();
TextEditor(posx+(0.02*13),posy,posz,scalex,scaley,scalez,red,green,blue);
TextY();
glPopMatrix();

TextEditor(posx+(0.02*15),posy,posz,scalex,scaley,scalez,red,green,blue);//of
TextO();
glPopMatrix();
TextEditor(posx+(0.02*16),posy,posz,scalex,scaley,scalez,red,green,blue);
TextF();
glPopMatrix();


TextEditor(posx+(0.02*18),posy,posz,scalex,scaley,scalez,red,green,blue);//Technology
TextT();
glPopMatrix();
TextEditor(posx+(0.02*19),posy,posz,scalex,scaley,scalez,red,green,blue);
TextE();
glPopMatrix();
TextEditor(posx+(0.02*20),posy,posz,scalex,scaley,scalez,red,green,blue);
TextC();
glPopMatrix();
TextEditor(posx+(0.02*21),posy,posz,scalex,scaley,scalez,red,green,blue);
```

```
        TextH();
        glPopMatrix();
        TextEditor(posx+(0.02*22),posy,posz,scalex,scaley,scalez,red,green,blue);
        TextN();
        glPopMatrix();
        TextEditor(posx+(0.02*23),posy,posz,scalex,scaley,scalez,red,green,blue);
        TextO();
        glPopMatrix();
        TextEditor(posx+(0.02*24),posy,posz,scalex,scaley,scalez,red,green,blue);
        TextL();
        glPopMatrix();
        TextEditor(posx+(0.02*25),posy,posz,scalex,scaley,scalez,red,green,blue);
        TextO();
        glPopMatrix();
        TextEditor(posx+(0.02*26),posy,posz,scalex,scaley,scalez,red,green,blue);
        TextG();
        glPopMatrix();
        TextEditor(posx+(0.02*27),posy,posz,scalex,scaley,scalez,red,green,blue);
        TextY();
        glPopMatrix();

}
void DepartmentText(GLfloat posx,GLfloat posy,GLfloat posz,GLfloat scalex,GLfloat
scaley,GLfloat scalez,GLfloat red,GLfloat green,GLfloat blue){
        glLoadIdentity();
        TextEditor(posx,posy,posz,scalex,scaley,scalez,red,green,blue);//Department
        TextD();
        glPopMatrix();
        TextEditor(posx+0.02,posy,posz,scalex,scaley,scalez,red,green,blue);
        TextE();
        glPopMatrix();
```

```
TextEditor(posx+(0.02*2),posy,posz,scalex,scaley,scalez,red,green,blue);
TextP();
glPopMatrix();
TextEditor(posx+(0.02*3),posy,posz,scalex,scaley,scalez,red,green,blue);
TextA();
glPopMatrix();
TextEditor(posx+(0.02*4),posy,posz,scalex,scaley,scalez,red,green,blue);
TextR();
glPopMatrix();
TextEditor(posx+(0.02*5),posy,posz,scalex,scaley,scalez,red,green,blue);
TextT();
glPopMatrix();
TextEditor(posx+(0.02*6),posy,posz,scalex,scaley,scalez,red,green,blue);
TextM();
glPopMatrix();
TextEditor(posx+(0.02*7),posy,posz,scalex,scaley,scalez,red,green,blue);
TextE();
glPopMatrix();
TextEditor(posx+(0.02*8),posy,posz,scalex,scaley,scalez,red,green,blue);
TextN();
glPopMatrix();
TextEditor(posx+(0.02*9),posy,posz,scalex,scaley,scalez,red,green,blue);
TextT();
glPopMatrix();


TextEditor(posx+(0.02*11),posy,posz,scalex,scaley,scalez,red,green,blue);//of
TextO();
glPopMatrix();
TextEditor(posx+(0.02*12),posy,posz,scalex,scaley,scalez,red,green,blue);
TextF();
```

```
        glPopMatrix();

        TextEditor(posx+(0.02*14),posy,posz,scalex,scaley,scalez,red,green,blue);//Computer
        TextC();
        glPopMatrix();
        TextEditor(posx+(0.02*15),posy,posz,scalex,scaley,scalez,red,green,blue);
        TextS();
        glPopMatrix();
        TextEditor(posx+(0.02*16),posy,posz,scalex,scaley,scalez,red,green,blue);
        TextE();
        glPopMatrix();

}
void MyText(GLfloat posx,GLfloat posy,GLfloat posz,GLfloat scalex,GLfloat scaley,GLfloat
scalez,GLfloat red,GLfloat green,GLfloat blue){
        glLoadIdentity();


        TextEditor(posx,posy   ,posz,scalex,scaley,scalez,red,green,blue);//Sai
        TextS();
        glPopMatrix();
        TextEditor(posx+0.02,posy   ,posz,scalex,scaley,scalez,red,green,blue);
        TextH();
        glPopMatrix();
        TextEditor(posx+(0.02*2),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        TextA();
        glPopMatrix();
        TextEditor(posx+(0.02*3),posy   ,posz,scalex,scaley,scalez,red,green,blue);//Nandan
        TextN();
        glPopMatrix();
        TextEditor(posx+(0.02*4),posy   ,posz,scalex,scaley,scalez,red,green,blue);
```

```
TextT();
glPopMatrix();
TextEditor(posx+(0.02*5),posy   ,posz,scalex,scaley,scalez,red,green,blue);
TextH();
glPopMatrix();
TextEditor(posx+(0.02*6),posy   ,posz,scalex,scaley,scalez,red,green,blue);
TextA();
glPopMatrix();

TextEditor(posx+(0.02*8),posy   ,posz,scalex,scaley,scalez,red,green,blue);
TextK();
glPopMatrix();
TextEditor(posx+(0.02*9),posy   ,posz,scalex,scaley,scalez,red,green,blue);
TextU();
glPopMatrix();
TextEditor(posx+(0.02*10),posy   ,posz,scalex,scaley,scalez,red,green,blue);//Pai
TextM();
glPopMatrix();
TextEditor(posx+(0.02*11),posy   ,posz,scalex,scaley,scalez,red,green,blue);
TextA();
glPopMatrix();
TextEditor(posx+(0.02*12),posy,posz,scalex,scaley,scalez,red,green,blue);
TextR();
glPopMatrix();

TextEditor(posx+(0.02*14),posy,posz,scalex,scaley,scalez,red,green,blue);//M
TextS();
glPopMatrix();

TextEditor(posx+0.02*2,posy-0.05,posz,scalex,scaley,scalez,red,green,blue);
Text1();
```

```
        glPopMatrix();
        TextEditor(posx+0.02*3,posy-0.05,posz,scalex,scaley,scalez,red,green,blue);
        TextG();
        glPopMatrix();
        TextEditor(posx+(0.02*4),posy-0.05,posz,scalex,scaley,scalez,red,green,blue);
        TextA();
        glPopMatrix();
        TextEditor(posx+(0.02*5),posy-0.05,posz,scalex,scaley,scalez,red,green,blue);
        Text1();
        glPopMatrix();
        TextEditor(posx+(0.02*6),posy-0.05,posz,scalex,scaley,scalez,red,green,blue);
        Text4();
        glPopMatrix();
        TextEditor(posx+(0.02*7),posy-0.05,posz,scalex,scaley,scalez,red,green,blue);
        TextC();
        glPopMatrix();
        TextEditor(posx+(0.02*8),posy-0.05,posz,scalex,scaley,scalez,red,green,blue);
        TextS();
        glPopMatrix();
        TextEditor(posx+(0.02*9),posy-0.05,posz,scalex,scaley,scalez,red,green,blue);
        Text1();
        glPopMatrix();
        TextEditor(posx+(0.02*10),posy-0.05,posz,scalex,scaley,scalez,red,green,blue);
        Text2();
        glPopMatrix();
        TextEditor(posx+(0.02*11),posy-0.05,posz,scalex,scaley,scalez,red,green,blue);
        Text3();
        glPopMatrix();

}
```

```
void LabText(GLfloat posx,GLfloat posy,GLfloat posz,GLfloat scalex,GLfloat scaley,GLfloat
scalez,GLfloat red,GLfloat green,GLfloat blue){
        glLoadIdentity();
        TextEditor(posx,posy,posz,scalex,scaley,scalez,red,green,blue);//Computer
        TextC();
        glPopMatrix();
        TextEditor(posx+(0.02*1),posy,posz,scalex,scaley,scalez,red,green,blue);
        TextG();
        glPopMatrix();
        TextEditor(posx+(0.02*4),posy,posz,scalex,scaley,scalez,red,green,blue);
        TextL();
        glPopMatrix();
        TextEditor(posx+(0.02*5),posy,posz,scalex,scaley,scalez,red,green,blue);
        TextA();
        glPopMatrix();
        TextEditor(posx+(0.02*6),posy,posz,scalex,scaley,scalez,red,green,blue);
        TextB();
        glPopMatrix();


        TextEditor(posx+(0.02*8),posy,posz,scalex,scaley,scalez,red,green,blue);//Graphics
        Text1();
        glPopMatrix();
        TextEditor(posx+(0.02*9),posy,posz,scalex,scaley,scalez,red,green,blue);
        Text0();
        glPopMatrix();
        TextEditor(posx+(0.02*10),posy,posz,scalex,scaley,scalez,red,green,blue);
        TextC();
        glPopMatrix();
        TextEditor(posx+(0.02*11),posy,posz,scalex,scaley,scalez,red,green,blue);
        TextS();
```

```
        glPopMatrix();
        TextEditor(posx+(0.02*12),posy,posz,scalex,scaley,scalez,red,green,blue);
        TextL();
        glPopMatrix();
        TextEditor(posx+(0.02*13),posy,posz,scalex,scaley,scalez,red,green,blue);
        Text6();
        glPopMatrix();
        TextEditor(posx+(0.02*14),posy,posz,scalex,scaley,scalez,red,green,blue);
        Text7();
        glPopMatrix();
}

void InfoButtonText(GLfloat posx,GLfloat posy,GLfloat posz,GLfloat scalex,GLfloat
scaley,GLfloat scalez,GLfloat red,GLfloat green,GLfloat blue){
        glLoadIdentity();
        TextEditor(posx,posy,posz,scalex,scaley,scalez,red,green,blue);
        TextI();
        glPopMatrix();
        TextEditor(posx+0.02,posy,posz,scalex,scaley,scalez,red,green,blue);
        TextN();
        glPopMatrix();
        TextEditor(posx+0.02*2,posy,posz,scalex,scaley,scalez,red,green,blue);
        TextF();
        glPopMatrix();
        TextEditor(posx+0.02*3,posy,posz,scalex,scaley,scalez,red,green,blue);
        TextO();
        glPopMatrix();
}
void HelpButtonText(GLfloat posx,GLfloat posy,GLfloat posz,GLfloat scalex,GLfloat
scaley,GLfloat scalez,GLfloat red,GLfloat green,GLfloat blue){
        glLoadIdentity();
```

```
        TextEditor(posx,posy,posz,scalex,scaley,scalez,red,green,blue);
        TextH();
        glPopMatrix();
        TextEditor(posx+0.02,posy,posz,scalex,scaley,scalez,red,green,blue);
        TextE();
        glPopMatrix();
        TextEditor(posx+0.02*2,posy,posz,scalex,scaley,scalez,red,green,blue);
        TextL();
        glPopMatrix();
        TextEditor(posx+0.02*3,posy,posz,scalex,scaley,scalez,red,green,blue);
        TextP();
        glPopMatrix();
}
//$$$$$$$$$$$$$$$$$$$ help $$$$$$$$$$$$$$$$$$$$$$$$$$$$$
void space(GLfloat posx,GLfloat posy,GLfloat posz,GLfloat scalex,GLfloat scaley,GLfloat
scalez,GLfloat red,GLfloat green,GLfloat blue)
{
        glLoadIdentity();
        TextEditor(posx+(0.02*8),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        TextP();
        glPopMatrix();
        TextEditor(posx+(0.02*9),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        TextR();
        glPopMatrix();
        TextEditor(posx+(0.02*10),posy   ,posz,scalex,scaley,scalez,red,green,blue);//Pai
        TextE();
        glPopMatrix();
        TextEditor(posx+(0.02*11),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        TextS();
        glPopMatrix();
        TextEditor(posx+(0.02*12),posy,posz,scalex,scaley,scalez,red,green,blue);
```

```
        TextS();
        glPopMatrix();

    glColor3f(1,0,0);
        TextEditor(posx+(0.02*14),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        TextS();
        glPopMatrix();
        TextEditor(posx+(0.02*15),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        TextP();
        glPopMatrix();
        TextEditor(posx+(0.02*16),posy   ,posz,scalex,scaley,scalez,red,green,blue);//Pai
        TextA();
        glPopMatrix();
        TextEditor(posx+(0.02*17),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        TextC();
        glPopMatrix();
        TextEditor(posx+(0.02*18),posy,posz,scalex,scaley,scalez,red,green,blue);
        TextE();
        glPopMatrix();

        TextEditor(posx+(0.02*20),posy,posz,scalex,scaley,scalez,red,green,blue);//of
        TextT();
        glPopMatrix();
        TextEditor(posx+(0.02*21),posy,posz,scalex,scaley,scalez,red,green,blue);
        TextO();
        glPopMatrix();

        glPopMatrix();
        TextEditor(posx+(0.02*23),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        TextJ();
        glPopMatrix();
```

```
        TextEditor(posx+(0.02*24),posy   ,posz,scalex,scaley,scalez,red,green,blue);//Pai
        TextU();
        glPopMatrix();
        TextEditor(posx+(0.02*25),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        TextM();
        glPopMatrix();
        TextEditor(posx+(0.02*26),posy,posz,scalex,scaley,scalez,red,green,blue);
        TextP();
        glPopMatrix();
}
void esc(GLfloat posx,GLfloat posy,GLfloat posz,GLfloat scalex,GLfloat scaley,GLfloat
scalez,GLfloat red,GLfloat green,GLfloat blue)
{
   glLoadIdentity();
        TextEditor(posx+(0.02*8),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        TextP();
        glPopMatrix();
        TextEditor(posx+(0.02*9),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        TextR();
        glPopMatrix();
        TextEditor(posx+(0.02*10),posy   ,posz,scalex,scaley,scalez,red,green,blue);//Pai
        TextE();
        glPopMatrix();
        TextEditor(posx+(0.02*11),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        TextS();
        glPopMatrix();
        TextEditor(posx+(0.02*12),posy,posz,scalex,scaley,scalez,red,green,blue);
        TextS();
        glPopMatrix();

        glColor3f(1,0,0);
```

```
        TextEditor(posx+(0.02*14),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        TextE();
        glPopMatrix();
        TextEditor(posx+(0.02*15),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        TextS();
        glPopMatrix();
        TextEditor(posx+(0.02*16),posy   ,posz,scalex,scaley,scalez,red,green,blue);//Pai
        TextC();
        glPopMatrix();

        TextEditor(posx+(0.02*18),posy,posz,scalex,scaley,scalez,red,green,blue);//of
        TextT();
        glPopMatrix();
        TextEditor(posx+(0.02*19),posy,posz,scalex,scaley,scalez,red,green,blue);
        TextO();
        glPopMatrix();

        glPopMatrix();
        TextEditor(posx+(0.02*21),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        TextQ();
        glPopMatrix();
        TextEditor(posx+(0.02*22),posy   ,posz,scalex,scaley,scalez,red,green,blue);//Pai
        TextU();
        glPopMatrix();
        TextEditor(posx+(0.02*23),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        TextI();
        glPopMatrix();
        TextEditor(posx+(0.02*24),posy,posz,scalex,scaley,scalez,red,green,blue);
        TextT();
        glPopMatrix();
}
```

```
void daymode(GLfloat posx,GLfloat posy,GLfloat posz,GLfloat scalex,GLfloat scaley,GLfloat
scalez,GLfloat red,GLfloat green,GLfloat blue)
{
   glLoadIdentity();
        TextEditor(posx+(0.02*8),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        TextP();
        glPopMatrix();
        TextEditor(posx+(0.02*9),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        TextR();
        glPopMatrix();
        TextEditor(posx+(0.02*10),posy   ,posz,scalex,scaley,scalez,red,green,blue);//Pai
        TextE();
        glPopMatrix();
        TextEditor(posx+(0.02*11),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        TextS();
        glPopMatrix();
        TextEditor(posx+(0.02*12),posy,posz,scalex,scaley,scalez,red,green,blue);
        TextS();
        glPopMatrix();


        TextEditor(posx+(0.02*14),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        Text1();
        glPopMatrix();

        TextEditor(posx+(0.02*16),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        TextF();
        glPopMatrix();
        TextEditor(posx+(0.02*17),posy   ,posz,scalex,scaley,scalez,red,green,blue);//Pai
        TextO();
        glPopMatrix();
```

```
   TextEditor(posx+(0.02*18),posy,posz,scalex,scaley,scalez,red,green,blue);//of
        TextR();
        glPopMatrix();

        TextEditor(posx+(0.02*20),posy,posz,scalex,scaley,scalez,red,green,blue);
        TextD();
        glPopMatrix();
        TextEditor(posx+(0.02*21),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        TextA();
        glPopMatrix();
        TextEditor(posx+(0.02*22),posy   ,posz,scalex,scaley,scalez,red,green,blue);//Pai
        TextY();
        glPopMatrix();

        TextEditor(posx+(0.02*24),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        TextM();
        glPopMatrix();
        TextEditor(posx+(0.02*25),posy,posz,scalex,scaley,scalez,red,green,blue);
        TextO();
        glPopMatrix();
        TextEditor(posx+(0.02*26),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        TextD();
        glPopMatrix();
        TextEditor(posx+(0.02*27),posy,posz,scalex,scaley,scalez,red,green,blue);
        TextE();
        glPopMatrix();
}
void nightmode(GLfloat posx,GLfloat posy,GLfloat posz,GLfloat scalex,GLfloat scaley,GLfloat
scalez,GLfloat red,GLfloat green,GLfloat blue)
{
   glLoadIdentity();
```

```
        TextEditor(posx+(0.02*8),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        TextP();
        glPopMatrix();
        TextEditor(posx+(0.02*9),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        TextR();
        glPopMatrix();
        TextEditor(posx+(0.02*10),posy   ,posz,scalex,scaley,scalez,red,green,blue);//Pai
        TextE();
        glPopMatrix();
        TextEditor(posx+(0.02*11),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        TextS();
        glPopMatrix();
        TextEditor(posx+(0.02*12),posy,posz,scalex,scaley,scalez,red,green,blue);
        TextS();
        glPopMatrix();


        TextEditor(posx+(0.02*14),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        Text2();
        glPopMatrix();

        TextEditor(posx+(0.02*16),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        TextF();
        glPopMatrix();
        TextEditor(posx+(0.02*17),posy   ,posz,scalex,scaley,scalez,red,green,blue);//Pai
        TextO();
        glPopMatrix();
    TextEditor(posx+(0.02*18),posy,posz,scalex,scaley,scalez,red,green,blue);//of
        TextR();
        glPopMatrix();
```

```
        TextEditor(posx+(0.02*20),posy,posz,scalex,scaley,scalez,red,green,blue);
        TextN();
        glPopMatrix();
        TextEditor(posx+(0.02*21),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        TextI();
        glPopMatrix();
        TextEditor(posx+(0.02*22),posy   ,posz,scalex,scaley,scalez,red,green,blue);//Pai
        TextG();
        glPopMatrix();
        TextEditor(posx+(0.02*23),posy   ,posz,scalex,scaley,scalez,red,green,blue);//Pai
        TextH();
        glPopMatrix();
        TextEditor(posx+(0.02*24),posy   ,posz,scalex,scaley,scalez,red,green,blue);//Pai
        TextT();
        glPopMatrix();

        TextEditor(posx+(0.02*26),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        TextM();
        glPopMatrix();
        TextEditor(posx+(0.02*27),posy,posz,scalex,scaley,scalez,red,green,blue);
        TextO();
        glPopMatrix();
        TextEditor(posx+(0.02*28),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        TextD();
        glPopMatrix();
        TextEditor(posx+(0.02*29),posy,posz,scalex,scaley,scalez,red,green,blue);
        TextE();
        glPopMatrix();
}
void color(GLfloat posx,GLfloat posy,GLfloat posz,GLfloat scalex,GLfloat scaley,GLfloat
scalez,GLfloat red,GLfloat green,GLfloat blue)
```

```
{
   glLoadIdentity();
        TextEditor(posx+(0.02*8),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        TextC();
        glPopMatrix();
        TextEditor(posx+(0.02*9),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        TextL();
        glPopMatrix();
        TextEditor(posx+(0.02*10),posy   ,posz,scalex,scaley,scalez,red,green,blue);//Pai
        TextI();
        glPopMatrix();
        TextEditor(posx+(0.02*11),posy   ,posz,scalex,scaley,scalez,red,green,blue);//Pai
        TextC();
        glPopMatrix();
        TextEditor(posx+(0.02*12),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        TextK();
        glPopMatrix();

        TextEditor(posx+(0.02*14),posy,posz,scalex,scaley,scalez,red,green,blue);
        TextO();
        glPopMatrix();
   TextEditor(posx+(0.02*15),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        TextN();
        glPopMatrix();

   TextEditor(posx+(0.02*17),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        TextC();
        glPopMatrix();
        TextEditor(posx+(0.02*18),posy   ,posz,scalex,scaley,scalez,red,green,blue);//Pai
        TextO();
        glPopMatrix();
```

```
   TextEditor(posx+(0.02*19),posy,posz,scalex,scaley,scalez,red,green,blue);//of
        TextL();
        glPopMatrix();
        TextEditor(posx+(0.02*20),posy,posz,scalex,scaley,scalez,red,green,blue);
        TextO();
        glPopMatrix();
        TextEditor(posx+(0.02*21),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        TextR();
        glPopMatrix();

        TextEditor(posx+(0.02*23),posy   ,posz,scalex,scaley,scalez,red,green,blue);//Pai
        TextB();
        glPopMatrix();
        TextEditor(posx+(0.02*24),posy   ,posz,scalex,scaley,scalez,red,green,blue);//Pai
        TextO();
        glPopMatrix();
        TextEditor(posx+(0.02*25),posy   ,posz,scalex,scaley,scalez,red,green,blue);//Pai
        TextX();
        glPopMatrix();
}
void color1(GLfloat posx,GLfloat posy,GLfloat posz,GLfloat scalex,GLfloat scaley,GLfloat
scalez,GLfloat red,GLfloat green,GLfloat blue)
{
   glLoadIdentity();


   TextEditor(posx+(0.02*27),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        TextT();
        glPopMatrix();
        TextEditor(posx+(0.02*28),posy,posz,scalex,scaley,scalez,red,green,blue);
        TextO();
```

```
        glPopMatrix();
        TextEditor(posx+(0.02*30),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        TextC();
        glPopMatrix();
        TextEditor(posx+(0.02*31),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        TextH();
        glPopMatrix();
        TextEditor(posx+(0.02*32),posy,posz,scalex,scaley,scalez,red,green,blue);
        TextA();
        glPopMatrix();
        TextEditor(posx+(0.02*33),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        TextN();
        glPopMatrix();
        TextEditor(posx+(0.02*34),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        TextG();
        glPopMatrix();
        TextEditor(posx+(0.02*35),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        TextE();
        glPopMatrix();
}
void color2(GLfloat posx,GLfloat posy,GLfloat posz,GLfloat scalex,GLfloat scaley,GLfloat
scalez,GLfloat red,GLfloat green,GLfloat blue)
{
   glLoadIdentity();
        TextEditor(posx+(0.02*26),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        TextS();
        glPopMatrix();
        TextEditor(posx+(0.02*27),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        TextT();
        glPopMatrix();
```

```
        TextEditor(posx+(0.02*28),posy   ,posz,scalex,scaley,scalez,red,green,blue);//Pai
        TextI();
        glPopMatrix();
        TextEditor(posx+(0.02*29),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        TextC();
        glPopMatrix();
        TextEditor(posx+(0.02*30),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        TextK();
        glPopMatrix();
    TextEditor(posx+(0.02*31),posy,posz,scalex,scaley,scalez,red,green,blue);
        TextM();
        glPopMatrix();
    TextEditor(posx+(0.02*32),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        TextA();
        glPopMatrix();
        TextEditor(posx+(0.02*33),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        TextN();
        glPopMatrix();


    TextEditor(posx+(0.02*35),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        TextC();
        glPopMatrix();
        TextEditor(posx+(0.02*36),posy   ,posz,scalex,scaley,scalez,red,green,blue);//Pai
        TextO();
        glPopMatrix();
    TextEditor(posx+(0.02*37),posy,posz,scalex,scaley,scalez,red,green,blue);//of
        TextL();
        glPopMatrix();
        TextEditor(posx+(0.02*38),posy,posz,scalex,scaley,scalez,red,green,blue);
        TextO();
        glPopMatrix();
```

```
        TextEditor(posx+(0.02*39),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        TextR();
        glPopMatrix();

}
void printscrore(GLfloat posx,GLfloat posy,GLfloat posz,GLfloat scalex,GLfloat scaley,GLfloat
scalez,GLfloat red,GLfloat green,GLfloat blue)
{
   glLoadIdentity();

        TextEditor(posx+(0.02*3),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        TextS();
        glPopMatrix();
        TextEditor(posx+(0.02*4),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        TextC();
        glPopMatrix();
        TextEditor(posx+(0.02*5),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        TextO();
        glPopMatrix();
        TextEditor(posx+(0.02*6),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        TextR();
        glPopMatrix();
        TextEditor(posx+(0.02*7),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        TextE();
        glPopMatrix();
}
void score1(GLfloat posx,GLfloat posy,GLfloat posz,GLfloat scalex,GLfloat scaley,GLfloat
scalez,GLfloat red,GLfloat green,GLfloat blue)
{
   glLoadIdentity();
   if(score==0)
```

```
{
        TextEditor(posx+(0.02*1),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        Text0();
        glPopMatrix();
        TextEditor(posx+(0.02*2),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        Text0();
        glPopMatrix();
}

if(score==1)
{
        TextEditor(posx+(0.02*1),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        Text0();
        glPopMatrix();
        TextEditor(posx+(0.02*2),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        Text1();
        glPopMatrix();
}

if(score==2)
{
        TextEditor(posx+(0.02*1),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        Text0();
        glPopMatrix();
        TextEditor(posx+(0.02*2),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        Text2();
        glPopMatrix();
}

if(score==3)
{
```

```
        TextEditor(posx+(0.02*1),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        Text0();
        glPopMatrix();
        TextEditor(posx+(0.02*2),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        Text3();
        glPopMatrix();
    }

    if(score==4)
    {
        TextEditor(posx+(0.02*1),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        Text0();
        glPopMatrix();
        TextEditor(posx+(0.02*2),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        Text4();
        glPopMatrix();
    }

    if(score==5)
    {
        TextEditor(posx+(0.02*1),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        Text0();
        glPopMatrix();
        TextEditor(posx+(0.02*2),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        Text5();
        glPopMatrix();
    }

    if(score==6)
    {
        TextEditor(posx+(0.02*1),posy   ,posz,scalex,scaley,scalez,red,green,blue);
```

```
        Text0();
        glPopMatrix();
        TextEditor(posx+(0.02*2),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        Text6();
        glPopMatrix();
    }

    if(score==7)
    {
        TextEditor(posx+(0.02*1),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        Text0();
        glPopMatrix();
        TextEditor(posx+(0.02*2),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        Text7();
        glPopMatrix();
    }

    if(score==8)
    {
        TextEditor(posx+(0.02*1),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        Text0();
        glPopMatrix();
        TextEditor(posx+(0.02*2),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        Text8();
        glPopMatrix();
    }

    if(score==9)
    {
        TextEditor(posx+(0.02*1),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        Text0();
```

```
        glPopMatrix();
        TextEditor(posx+(0.02*2),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        Text9();
        glPopMatrix();
    }

    if(score==10)
    {
        TextEditor(posx+(0.02*1),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        Text1();
        glPopMatrix();
        TextEditor(posx+(0.02*2),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        Text0();
        glPopMatrix();
    }

    if(score==11)
    {
        TextEditor(posx+(0.02*1),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        Text1();
        glPopMatrix();
        TextEditor(posx+(0.02*2),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        Text1();
        glPopMatrix();
    }

    if(score==12)
    {
        TextEditor(posx+(0.02*1),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        Text1();
        glPopMatrix();
```

```
        TextEditor(posx+(0.02*2),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        Text2();
        glPopMatrix();
    }

    if(score==13)
    {
        TextEditor(posx+(0.02*1),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        Text1();
        glPopMatrix();
        TextEditor(posx+(0.02*2),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        Text3();
        glPopMatrix();
    }

    if(score==14)
    {
        TextEditor(posx+(0.02*1),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        Text1();
        glPopMatrix();
        TextEditor(posx+(0.02*2),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        Text4();
        glPopMatrix();
    }

    if(score==15)
    {
        TextEditor(posx+(0.02*1),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        Text1();
        glPopMatrix();
        TextEditor(posx+(0.02*2),posy   ,posz,scalex,scaley,scalez,red,green,blue);
```

```
        Text5();
        glPopMatrix();
    }

    if(score==16)
    {
        TextEditor(posx+(0.02*1),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        Text1();
        glPopMatrix();
        TextEditor(posx+(0.02*2),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        Text6();
        glPopMatrix();
    }

    if(score==17)
    {
        TextEditor(posx+(0.02*1),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        Text1();
        glPopMatrix();
        TextEditor(posx+(0.02*2),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        Text7();
        glPopMatrix();
    }

    if(score==18)
    {
        TextEditor(posx+(0.02*1),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        Text1();
        glPopMatrix();
        TextEditor(posx+(0.02*2),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        Text8();
```

```
        glPopMatrix();
    }

    if(score==19)
    {
        TextEditor(posx+(0.02*1),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        Text1();
        glPopMatrix();
        TextEditor(posx+(0.02*2),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        Text9();
        glPopMatrix();
    }

    if(score==20)
    {
        TextEditor(posx+(0.02*1),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        Text2();
        glPopMatrix();
        TextEditor(posx+(0.02*2),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        Text0();
        glPopMatrix();
    }

    if(score==21)
    {
        TextEditor(posx+(0.02*1),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        Text2();
        glPopMatrix();
        TextEditor(posx+(0.02*2),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        Text1();
        glPopMatrix();
```

```
}

if(score==22)
{
        TextEditor(posx+(0.02*1),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        Text2();
        glPopMatrix();
        TextEditor(posx+(0.02*2),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        Text2();
        glPopMatrix();
}

if(score==23)
{
        TextEditor(posx+(0.02*1),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        Text2();
        glPopMatrix();
        TextEditor(posx+(0.02*2),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        Text3();
        glPopMatrix();
}

if(score==24)
{
        TextEditor(posx+(0.02*1),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        Text2();
        glPopMatrix();
        TextEditor(posx+(0.02*2),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        Text4();
        glPopMatrix();
}
```

```
if(score==25)
{
        TextEditor(posx+(0.02*1),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        Text2();
        glPopMatrix();
        TextEditor(posx+(0.02*2),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        Text5();
        glPopMatrix();
}

if(score==26)
{
        TextEditor(posx+(0.02*1),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        Text2();
        glPopMatrix();
        TextEditor(posx+(0.02*2),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        Text6();
        glPopMatrix();
}

if(score==27)
{
        TextEditor(posx+(0.02*1),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        Text2();
        glPopMatrix();
        TextEditor(posx+(0.02*2),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        Text7();
        glPopMatrix();
}
```

```
if(score==28)
{
        TextEditor(posx+(0.02*1),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        Text2();
        glPopMatrix();
        TextEditor(posx+(0.02*2),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        Text8();
        glPopMatrix();
}

if(score==29)
{
        TextEditor(posx+(0.02*1),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        Text2();
        glPopMatrix();
        TextEditor(posx+(0.02*2),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        Text9();
        glPopMatrix();
}

if(score==30)
{
        TextEditor(posx+(0.02*1),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        Text3();
        glPopMatrix();
        TextEditor(posx+(0.02*2),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        Text0();
        glPopMatrix();
}

if(score==31)
```

```
{
        TextEditor(posx+(0.02*1),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        Text3();
        glPopMatrix();
        TextEditor(posx+(0.02*2),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        Text1();
        glPopMatrix();
}

if(score==32)
{
        TextEditor(posx+(0.02*1),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        Text3();
        glPopMatrix();
        TextEditor(posx+(0.02*2),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        Text2();
        glPopMatrix();
}

if(score==33)
{
        TextEditor(posx+(0.02*1),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        Text3();
        glPopMatrix();
        TextEditor(posx+(0.02*2),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        Text3();
        glPopMatrix();
}

if(score==34)
{
```

```
        TextEditor(posx+(0.02*1),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        Text3();
        glPopMatrix();
        TextEditor(posx+(0.02*2),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        Text4();
        glPopMatrix();
    }

    if(score==35)
    {
        TextEditor(posx+(0.02*1),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        Text3();
        glPopMatrix();
        TextEditor(posx+(0.02*2),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        Text5();
        glPopMatrix();
    }

    if(score==36)
    {
        TextEditor(posx+(0.02*1),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        Text3();
        glPopMatrix();
        TextEditor(posx+(0.02*2),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        Text6();
        glPopMatrix();
    }

    if(score==37)
    {
        TextEditor(posx+(0.02*1),posy   ,posz,scalex,scaley,scalez,red,green,blue);
```

```
        Text3();
        glPopMatrix();
        TextEditor(posx+(0.02*2),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        Text7();
        glPopMatrix();
    }

    if(score==38)
    {
        TextEditor(posx+(0.02*1),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        Text3();
        glPopMatrix();
        TextEditor(posx+(0.02*2),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        Text8();
        glPopMatrix();
    }

    if(score==39)
    {
        TextEditor(posx+(0.02*1),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        Text3();
        glPopMatrix();
        TextEditor(posx+(0.02*2),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        Text9();
        glPopMatrix();
    }

    if(score==40)
    {
        TextEditor(posx+(0.02*1),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        Text4();
```

```
        glPopMatrix();
        TextEditor(posx+(0.02*2),posy   ,posz,scalex,scaley,scalez,red,green,blue);
        Text0();
        glPopMatrix();
    }
}

//_____________________________________________________building.cpp_____________
_____________________________________________
void DrawGLScene(float x,float y)
{
    glBegin(GL_POLYGON);
    glColor3f(Rb,Gb,Bb);
    glVertex3f(-2.0f+x, 2.0f-y, -0.1f);
    glVertex3f(-1.25f+x, 2.0f-y, -0.1f);
    glVertex3f(-1.25f+x, 0.0f, -0.10f);
    glVertex3f(-2.0f+x,0.0f, -0.10f);
    glEnd();

    // back ground**************************************************************

    glBegin(GL_POLYGON);
    glColor3f(Rb1,Gb1,Bb1);
    glVertex3f(-2.0f+x,1.30f, -0.5f);
    glVertex3f(18.0f+x, 1.3f, -0.5f);
    glColor3f(Rb2,Gb2,Bb2);
    glVertex3f(18.0f+x, -0.5f, -0.5f);
    glVertex3f(-2.0f+x,-0.5f, -0.5f);
    glEnd();

    //road  black  above upper yellow strip ******************
```

```
  glBegin(GL_POLYGON);
   glColor3f(0.20f,0.20f,0.20f);
   glVertex3f(-2.0f+x,0.0f, -0.3f);
   glVertex3f(18.0f+x, 0.0f, -0.3f);
   glColor3f(0.0f,0.0f,0.0f);
   glVertex3f(18.0f+x, -2.5f, -0.3f);
   glVertex3f(-2.0f+x,-2.5f, -0.3f);
   glEnd();


   glBegin(GL_POLYGON);
   glColor3f(Rb2,Gb2,Bb2);
   glVertex3f(18.0f+x, 2.7f, -0.5f);
   glVertex3f(-2.0f+x,2.7f, -0.5f);
   glColor3f(Rb1,Gb1,Bb1);
   glVertex3f(-2.0f+x,1.30f, -0.5f);
   glVertex3f(18.0f+x, 1.3f, -0.5f);
   glEnd();

   glBegin(GL_POLYGON);
   glColor3f(0.0f,0.0f,0.0f);
   glVertex3f(-2.0f+x,1.30f, -0.5f);
   glVertex3f(18.0f+x, 1.3f, -0.5f);
   glColor3f(0.0f,0.0f,0.0f);
   glVertex3f(18.0f+x, -3.0f, -0.5f);
   glVertex3f(-2.0f+x,-3.0f, -0.5f);
   glEnd();

   // Road  upper yello strip
**********************************************************
```

```
    glBegin(GL_POLYGON);
    glColor3f(1.0f,1.0f,0.0f);
    glVertex3f(-1.0f+x, -0.15f, -0.05f);
    glVertex3f(-2.0f+x,-0.15f, -0.05f);
   // glColor3f(1.0f,1.0f,1.0f);5
    glVertex3f(-2.0f+x, -0.22f, -0.05f);
    glVertex3f(-1.0f+x,-0.22f, -0.05f);
    glEnd();



    glBegin(GL_POLYGON);
    glColor3f(1.0f,1.0f,0.0f);
    glVertex3f(-1.0f+x, -1.95f, -0.0f);
    glVertex3f(-2.0f+x,-1.95f, -0.0f);
    //glColor3f(1.0f,1.0f,1.0f);
    glVertex3f(-2.0f+x, -2.02f, -0.0f);
    glVertex3f(-1.0f+x,-2.02f, -0.0f);
    glEnd();
}

void background()
 {
    DrawGLScene(-5,0);
    DrawGLScene(-4,0.5 );
    DrawGLScene(-3,0.3 );
    DrawGLScene(-2,0.5 );
    DrawGLScene(-1,0.2 );
    DrawGLScene(0,0.1 );
    DrawGLScene(1,0.0 );
    DrawGLScene(2,0.5 );
```

```
    DrawGLScene(3,0.3 );
    DrawGLScene(4,0.5 );
    DrawGLScene(5,0.2 );
    DrawGLScene(6,0.1 );
    DrawGLScene(7,0 );
    DrawGLScene(8,0.5 );
    DrawGLScene(9,0.3 );
    DrawGLScene(10,0.5 );
    DrawGLScene(11,0.2 );
    DrawGLScene(12,0.1 );
    DrawGLScene(13,0.0 );
    DrawGLScene(14,0.5 );

 }
//_____moon.cpp_____
_____

void drawBall(void)
{
float ballX = 2.0f;
float ballY = 2.2f;
float ballZ = -0.5f;
    glTranslatef(ballX,ballY,ballZ);
    glutSolidSphere(0.4,100,2);

}

//_____sticman.cpp_____
_____
```

```
void stickman()
{
float ballX = 0;
float ballY = -1.5;
float ballZ = 0;

float ballX1 = -0.5;
float ballY1 = -3.15;
float ballZ1 = 0;



float ballX2 = 0.27;
float ballY2 = -3.15;
float ballZ2 = 0;

//color
changing##############################################################################
###################################
   glPushMatrix();
   glTranslatef(ballX-2.25,ballY+3,ballZ);
   glRotatef(50,0,0,1);
   glRotatef(15,0,1,0);
   glColor3f(1,1,1);
   glutSolidSphere(0.15,4,2);
   glPopMatrix();

   glPushMatrix();
   glTranslatef(ballX-1.9,ballY+3,ballZ);
   glRotatef(50,0,0,1);
   glRotatef(15,0,1,0);
   glColor3f(0,0,1);
```

```
    glutSolidSphere(0.15,4,2);
    glPopMatrix();

    glPushMatrix();
    glTranslatef(ballX-1.57,ballY+3,ballZ);
    glRotatef(50,0,0,1);
    glRotatef(15,0,1,0);
    glColor3f(1,0,1);
    glutSolidSphere(0.15,4,2);
    glPopMatrix();

    glPushMatrix();
    glTranslatef(ballX-1.26,ballY+3,ballZ);
    glRotatef(50,0,0,1);
    glRotatef(15,0,1,0);
    glColor3f(1,1,0);
    glutSolidSphere(0.15,4,2);
    glPopMatrix();

    glPushMatrix();
    glTranslatef(ballX-0.95,ballY+3,ballZ);
    glRotatef(50,0,0,1);
    glRotatef(15,0,1,0);
    glColor3f(0,1,1);
    glutSolidSphere(0.15,4,2);
    glPopMatrix();
//#############################################################################
##############################################
    glColor3f(R,G,B);
    glPushMatrix();
    glTranslatef(ballX,ballY,ballZ);
```

```
glutSolidSphere(0.15,100,2);
glPopMatrix();

glPushMatrix();
glTranslatef(-0.49,-2.2,0);
glutSolidSphere(0.04,100,2);
glPopMatrix();

glPushMatrix();
glTranslatef(0.5,-1.95,0);
glutSolidSphere(0.05,100,2);
glPopMatrix();

glPushMatrix();
glTranslatef(-0.39,-1.87,0);
glutSolidSphere(0.03,100,2);
glPopMatrix();

glPushMatrix();
glTranslatef(0,-1.5f,-1.0f);
glLineWidth(5);
glBegin(GL_LINE_STRIP);
glVertex3f(-0.4,-1.5,1);
glVertex3f(-0.3,-1.1,1);
glVertex3f(-0.05,-0.7,1);
glVertex3f(0.13,-1.1,1);
glVertex3f(0.16,-1.5,1);
glEnd();
glBegin(GL_LINE_STRIP);
glVertex3f(-0.5,-0.7,1);
glVertex3f(-0.4,-0.35,1);
```

```
    glVertex3f(0,-0.25,1);
    glVertex3f(0.2,-0.45,1);
    glVertex3f(0.5,-0.45,1);
    glEnd();
    glBegin(GL_LINE_STRIP);
    glVertex3f(0,-0.25,1);
    glVertex3f(-0.05,-0.7,1);
    glEnd();
    //skateboard*****************************
    glBegin(GL_LINE_STRIP);
    glColor3f(1.0, 1.0, 1.0);
    glVertex3f(-0.7,-1.4,1);
    glVertex3f(-0.6,-1.5,1);
    glVertex3f(0.35,-1.5,1);
    glVertex3f(0.5,-1.4,1);
    glEnd();
    glPopMatrix();

    glPushMatrix();
    glTranslatef(ballX1,ballY1,ballZ1);
    glColor3f(0.0, 1.0, 1.0);
    glutSolidSphere(0.05,100,2);
    glPopMatrix();

    glPushMatrix();
    glTranslatef(ballX2,ballY2,ballZ2);
    glColor3f(0.0, 1.0, 1.0);
    glutSolidSphere(0.05,100,2);
    glPopMatrix();
}
```

```
//_____strips.cpp_____
_____

void drawstrips()
{
   glBegin(GL_POLYGON);
   glColor3f(1.0f,1.0f,0.0f);
   glVertex3f(-5.2f, -1.08f, -0.1f);
   glVertex3f(-6.7f,-1.08f, -0.1f);
  // glColor3f(1.0f,1.0f,1.0f);
   glVertex3f(-6.7f, -1.17f, -0.1f);
   glVertex3f(-5.2f,-1.17f, -0.1f);
   glEnd();


   glBegin(GL_POLYGON);
   glColor3f(1.0f,1.0f,0.0f);
   glVertex3f(-4.95f+1, -1.08f, -0.1f);
   glVertex3f(-3.45f+1,-1.08f, -0.1f);
  // glColor3f(1.0f,1.0f,1.0f);
   glVertex3f(-3.45f+1, -1.17f, -0.1f);
   glVertex3f(-4.95f+1,-1.17f, -0.1f);
   glEnd();

   glBegin(GL_POLYGON);
   glColor3f(1.0f,1.0f,0.0f);
   glVertex3f(-3.2f+2, -1.08f, -0.1f);
   glVertex3f(-1.7f+2,-1.08f, -0.1f);
   //glColor3f(1.0f,1.0f,1.0f);
   glVertex3f(-1.7f+2, -1.17f, -0.1f);
```

```
glVertex3f(-3.20f+2,-1.17f, -0.1f);
glEnd();

glBegin(GL_POLYGON);
glColor3f(1.0f,1.0f,0.0f);
glVertex3f(-1.5f+3, -1.08f, -0.1f);
glVertex3f(0.0f+3,-1.08f, -0.1f);
// glColor3f(1.0f,1.0f,1.0f);
glVertex3f(0.0f+3, -1.17f, -0.1f);
glVertex3f(-1.5f+3,-1.17f, -0.1f);
glEnd();

glBegin(GL_POLYGON);
glColor3f(1.0f,1.0f,0.0f);
glVertex3f(0.2f+4, -1.08f, -0.1f);
glVertex3f(1.7f+4,-1.08f, -0.1f);
//glColor3f(1.0f,1.0f,1.0f);
glVertex3f(1.7f+4, -1.17f, -0.1f);
glVertex3f(0.2f+4,-1.17f, -0.1f);
glEnd();
glBegin(GL_POLYGON);
glColor3f(1.0f,1.0f,0.0f);
glVertex3f(1.9f+5, -1.08f, -0.1f);
glVertex3f(3.4f+5,-1.08f, -0.1f);
//glColor3f(1.0f,1.0f,1.0f);
glVertex3f(3.4f+5, -1.17f, -0.1f);
glVertex3f(1.9f+5,-1.17f, -0.1f);
glEnd();

glBegin(GL_POLYGON);
glColor3f(1.0f,1.0f,0.0f);
```

```
glVertex3f(3.6f+6, -1.08f, -0.1f);
glVertex3f(5.1f+6,-1.08f, -0.1f);
// glColor3f(1.0f,1.0f,1.0f);
glVertex3f(5.1f+6, -1.17f, -0.1f);
glVertex3f(3.6f+6,-1.17f, -0.1f);
glEnd();



glBegin(GL_POLYGON);
glColor3f(1.0f,1.0f,0.0f);
glVertex3f(5.3f+7, -1.08f, -0.1f);
glVertex3f(6.8f+7,-1.08f, -0.1f);
//glColor3f(1.0f,1.0f,1.0f);
glVertex3f(6.8f+7, -1.17f, -0.1f);
glVertex3f(5.3f+7,-1.17f, -0.1f);
glEnd();


glBegin(GL_POLYGON);
glColor3f(1.0f,1.0f,0.0f);
glVertex3f(7.0f+8, -1.08f, -0.1f);
glVertex3f(8.5f+8,-1.08f, -0.1f);
//glColor3f(1.0f,1.0f,1.0f);
glVertex3f(8.5f+8, -1.17f, -0.1f);
glVertex3f(7.0f+8,-1.17f, -0.1f);
glEnd();


glBegin(GL_POLYGON);
glColor3f(1.0f,1.0f,0.0f);
```

```
    glVertex3f(8.7f+9, -1.08f, -0.1f);
    glVertex3f(10.2f+9,-1.08f, -0.1f);
    //glColor3f(1.0f,1.0f,1.0f);
    glVertex3f(10.2f+9, -1.17f, -0.1f);
    glVertex3f(8.7f+9,-1.17f, -0.1f);
    glEnd();



    glBegin(GL_POLYGON);
    glColor3f(1.0f,1.0f,0.0f);
    glVertex3f(10.4f+10, -1.08f, -0.1f);
    glVertex3f(11.9f+10,-1.08f, -0.1f);
    //glColor3f(1.0f,1.0f,1.0f);
    glVertex3f(11.9f+10, -1.17f, -0.1f);
    glVertex3f(10.4f+10,-1.17f, -0.1f);
    glEnd();

}
//THORNS ON THE
ROAD&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
void star()
{
    float y;
    for(y=0;y<0.015;y+=0.005)
    {
    glColor3f(0,0,0);
    glTranslatef(1.5,-0.4-y,-0.1);
    glutSolidSphere(0.2,3,2);
    glTranslatef(0.5,0,0);
    glutSolidSphere(0.2,3,2);
```

```
    glTranslatef(0.5,0,0);
    glutSolidSphere(0.2,3,2);
    glTranslatef(0.5,0,0);
    glutSolidSphere(0.2,3,2);
    glTranslatef(0.5,0,0);
    glutSolidSphere(0.2,3,2);
    glTranslatef(0.5,0,0);
    glutSolidSphere(0.2,3,2);
glTranslatef(-4,0,0.1);
    }
}

//_____windows.cpp_____
_____

//the function to display windows on
buildings**********************************************************************
void windwos(float h)
{
    float i=0,j=0;
    while(i<=21)
    {
        if(j==3)
        {
          i=i+0.37;
          j=0;
        }
        j++;
        glBegin(GL_POLYGON);
        glColor3f(1.0f,1.0f,0.0f);
        glVertex3f(-6.50f+i, 0.2f+h,-0.1f);
```

```
        glVertex3f(-6.40f+i, 0.2f+h,-0.1f);
        glColor3f(1.0f,1.0f,1.0f);
        glVertex3f(-6.40f+i, 0.1f+h,-0.1f);
        glVertex3f(-6.50f+i,0.1f+h,-0.1f);
        glEnd();
        i=i+0.21;
    }

}
//the function to display more row of windows on building
void window1()
{

    windwos(0);
    windwos(0.3);
    windwos(0.6);
    windwos(0.9);

}

//_____________________________________________________main.cpp_____________
_____________________________________________________


void InitGL(int Width, int Height)
{
    glClearColor(1.0f, 1.0f, 1.0f, 1.0f);
    glClearDepth(1.0);
    glDepthFunc(GL_LESS);
    glEnable(GL_DEPTH_TEST);
    glShadeModel(GL_SMOOTH);
```

```
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluPerspective(45.0f,(GLfloat)Width/(GLfloat)Height,0.1f,100.0f);
    glMatrixMode(GL_MODELVIEW);
}
void ReSizeGLScene(int Width, int Height)
{
    if(Height==0)
    Height=1;
    glViewport(0,0,Width,Height);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluPerspective(45.0f,(GLfloat)Width/(GLfloat)Height,0.1f,100.0f);
    glMatrixMode(GL_MODELVIEW);
}
void display()
{
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glLoadIdentity();

    InfoButtonText(0.25,0.19,-0.5,0.005+ibigx,0.005+ibigy,0,0,1,1);
    if(okinfo==1)
    {
    MenuButtonText(-0.25,0.15,-0.5,0.005,0.01,0,0,1,1);
          DepartmentText(-0.15,0.10,-0.5,0.005,0.01,0,0,1,1);
          LabText(-0.125,0.05,-0.5,0.005,0.01,0,0,1,1);
          MyText(-0.125,0,-0.5,0.005,0.01,0,0,1,1);
    }
    HelpButtonText(0.25,0.17,-0.5,0.005+hbigx,0.005+hbigy,0,0,1,1);
    if(okhelp==1)
    {
```

```
        space(-0.5,0.15,-0.5,0.005,0.01,0,0,1,1);
        esc(-0.5,0.11,-0.5,0.005,0.01,0,0,1,1);
        daymode(-0.5,0.07,-0.5,0.005,0.01,0,0,1,1);
        nightmode(-0.5,0.03,-0.5,0.005,0.01,0,0,1,1);
        color(-0.5,-0.025,-0.5,0.005,0.01,0,0,1,1);
        color1(-0.45,-0.01,-0.5,0.005,0.01,0,0,1,1);
        color2(-0.45,-0.04,-0.5,0.005,0.01,0,0,1,1);
    }
    score1(-0.26,0.17,-0.5,0.005,0.005,0,0,1,1);
    printscrore(-0.42,0.17,-0.5,0.005,0.005,0,0,1,1);
    glPushMatrix();
    glTranslatef(rtri/3,0.0f,-6.0f);
    background();
    glPopMatrix();

    glPushMatrix();
    glTranslatef(rtri/3-0.355,0.0f,-5.99f);
    window1();
    glPopMatrix();

    glPushMatrix();
    glTranslatef(rtri,0.0f,-6.0f);
    drawstrips();
    star();
    glPopMatrix();

    glPushMatrix();
    glTranslatef(0,0,-6.0);
    glColor3f(m1,m2,m3);
    drawBall();
    glPopMatrix();
```

```
    glPushMatrix();
    glScalef(0.7,0.7,1);
    glTranslatef(-4,up+1.885,-6.0);
    stickman();
    glPopMatrix();
    glutSwapBuffers();
    if(mov==1)
    {
       if(rtri<=-4&&rtri>=-7.5&&up<=0)
       {
          mov=0;
          Sleep(700);
          glutDestroyWindow(window);
          Sleep(1000000000);
       }
       Sleep(speed);
       rtri-=0.25f;
       if(rtri==-8.5)
          score++;
       if(rtri==-15.75)
       {
       rtri=2.5f;
       }
       rquad-=15.0f;
    }
}
void mouse_routine(int x, int y)
{
if(x>1121&&x<1251&&y>17&&y<42)
```

```
{
   ibigx=0.004;
   ibigy=0.002;
}
else
{
   ibigx=0;
   ibigy=0;
}
if(x>1120&&x<1251&&y>53&&y<73)
{
   hbigx=0.004;
   hbigy=0.002;
}
else
{
   hbigx=0;
   hbigy=0;
}
if(x>58&&x<228&&y>94&&y<116)
{
   pbigx=0.004;
   pbigy=0.002;
}
else
{
   pbigx=0;
   pbigy=0;
}

}
```

```
void mouse(int btn,int state,int x,int y)
{
   if(btn==GLUT_LEFT_BUTTON&&state==GLUT_DOWN)
   {
      if(x>=14&&x<38&&y>=8&&y<25)
      {
         R=1;
         G=1;
         B=1;
      }
      if(x>=49&&x<75&&y>=8&&y<25)
      {
         R=0;
         G=0;
         B=1;
      }
      if(x>=85&&x<110&&y>=8&&y<25)
      {
         R=1;
         G=0;
         B=1;
      }
      if(x>=118&&x<138&&y>=8&&y<25)
      {
         R=1;
         G=1;
         B=0;
      }
      if(x>=150&&x<174&&y>=8&&y<25)
      {
         R=0;
```

```
        G=1;
        B=1;
      }
      if(x>1121&&x<1251&&y>17&&y<42)
      {
        okinfo=1;
      }
      else
      {
        okinfo=0;
      }
      if(x>1120&&x<1251&&y>53&&y<73)
      {
        okhelp=1;
      }
      else
      {
        okhelp=0;
      }
                if(x>1119&&x<1251&&y>53&&y<73)
      {
        mov=0;
      }
   }
printf("%d       %d\n",x,y);
}
void keyPressed(unsigned char key, int x, int y)
{
   if (key == ESCAPE)
   {
      glutDestroyWindow(window);
```

```
    exit(0);
  }
  if(key=='1')
  {
    Rb=0.25f,Gb=0.25f,Bb=0.5f;
    Rb1=1,Gb1=1,Bb1=1;
    Rb2=0.5,Gb2=0.5f,Bb2=1;
    m1=1,m2=1,m3=0.3;
  }
  if(key=='2')
  {
    Rb=0.05,Gb=0.05,Bb=0.05f;
    Rb1=0,Gb1=0,Bb1=0.5;
    Rb2=0,Gb2=0,Bb2=0;
    m1=0.8,m2=0.8,m3=0.8;
  }

  if(key == ' ')
  {
    ok=0;
    if(mov==1)
    {
      while(1)
      {
        up=up+0.2;
        display();
        if(up>=1.5)
        break ;
      }
      while(up!=0)
      {
```

```
            up=up-0.1;
            display();
            if(up<=0)
            break ;
          }
          ok=1;
       }
      mov=1;
       speed=speed-0.99999;
       system("cls");
       printf("score :%d",score);
   }
}
int main(int argc, char **argv)
{
   glutInit(&argc,argv);
   glutInitDisplayMode(GLUT_RGBA | GLUT_DOUBLE | GLUT_ALPHA | GLUT_DEPTH);
   glutInitWindowSize(1366,768);
   glutInitWindowPosition(0,0);
   window = glutCreateWindow("STICK MAN GAME");
   glutDisplayFunc(&display);
   glutIdleFunc(&display);
   glutReshapeFunc(&ReSizeGLScene);
   glutKeyboardFunc(&keyPressed);
   glutMouseFunc(mouse);
   glutPassiveMotionFunc(mouse_routine);
   InitGL(1366,768);
   glutMainLoop();
   return 1;
}
```

CHAPTER 5

TESTING AND RESULTS

5.1 DIFFERENT TYPES OF TESTING

1. Unit Testing

Individual components are tested to ensure that they operate correctly. Each component is tested independently, without other system components.

2. Module Testing

A module is a collection of dependent components such as a object class, an abstract data type or some looser collection of procedures and functions. A module related components, so can be tested without other system modules.

3. System Testing

This is concerned with finding errors that result from unanticipated interaction between sub-system interface problems.

4. Acceptance Testing

The system is tested with data supplied by the system customer rather than simulated test data.

5.2 OUTPUT SNAPSHOTS

The below figure 5.1 shows the Skating Stickman at day.

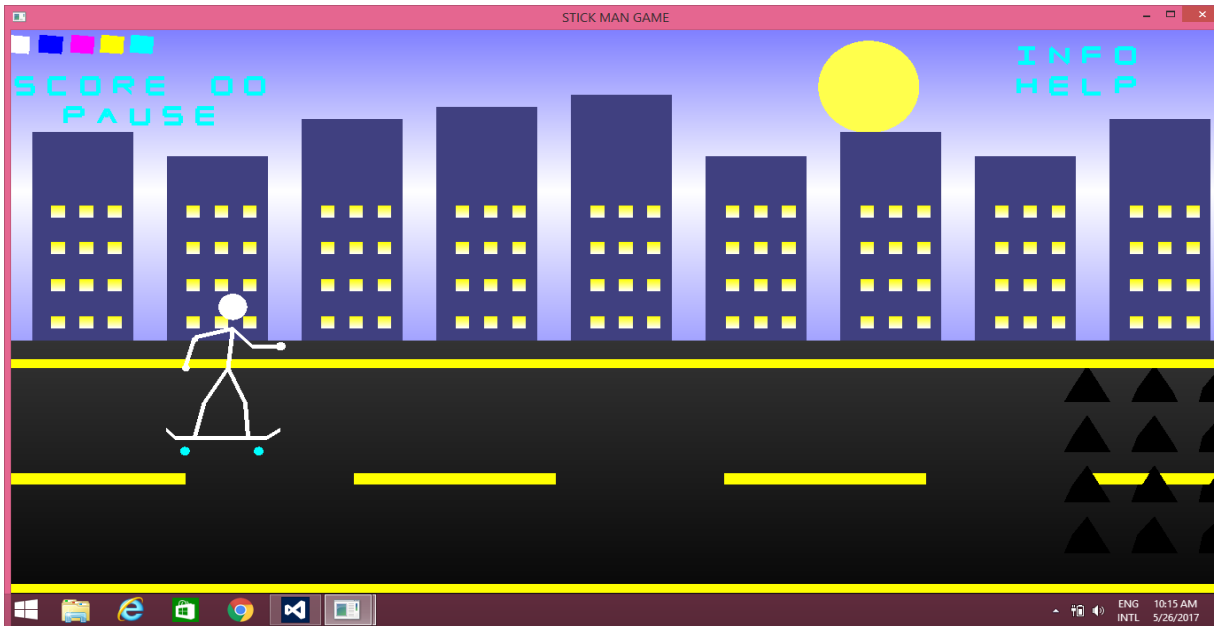

Figure 5.1 Day mode

The below figure 5.2 shows the Skating Stickman at night.

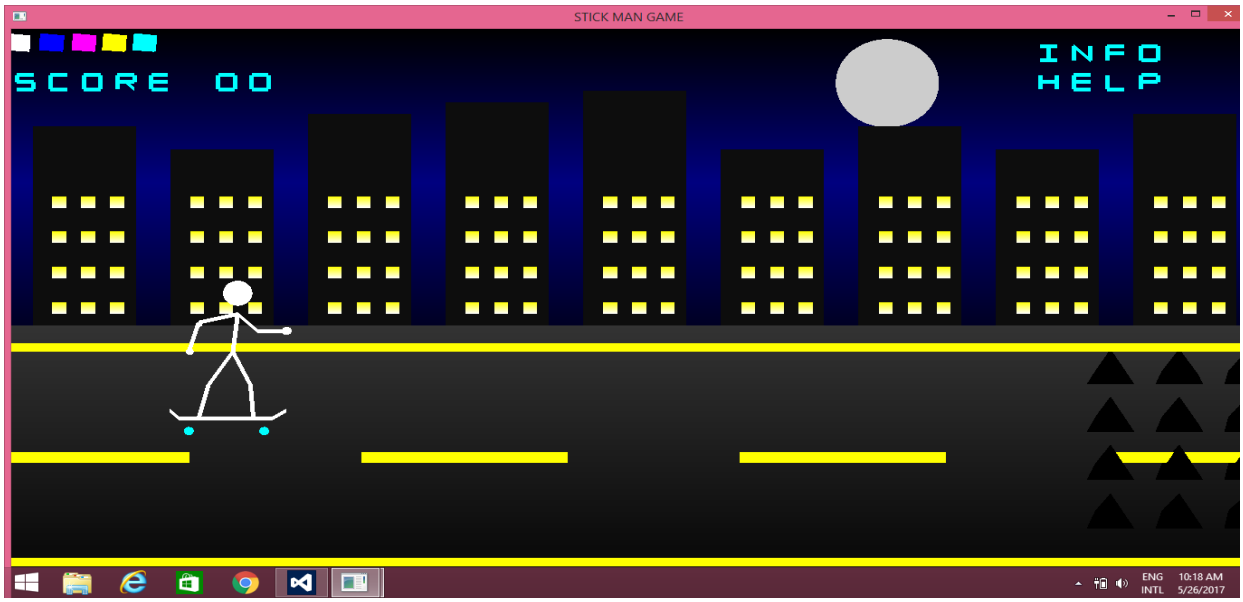

Figure 5.2 Night mode

The below figure 5.3 shows the stickman jumping using keyboard interface.

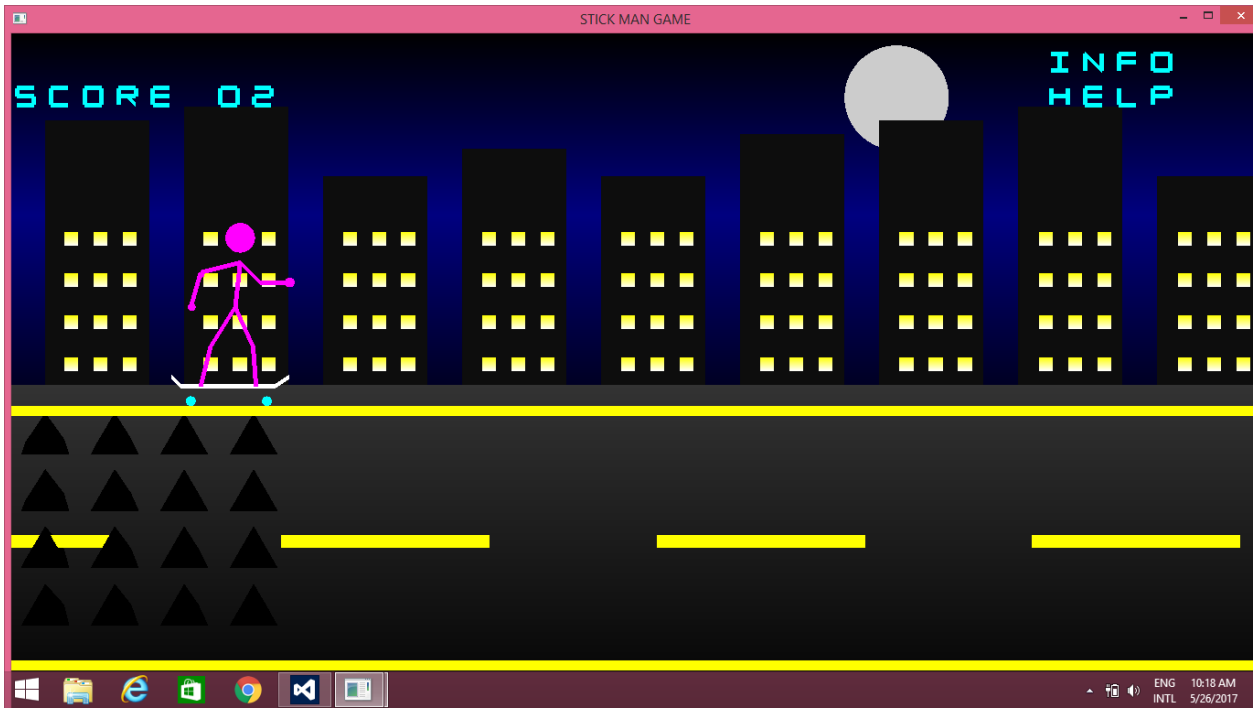

Figure 5.3 Jumping of stickman

The below figure 5.4 shows the information about the project.

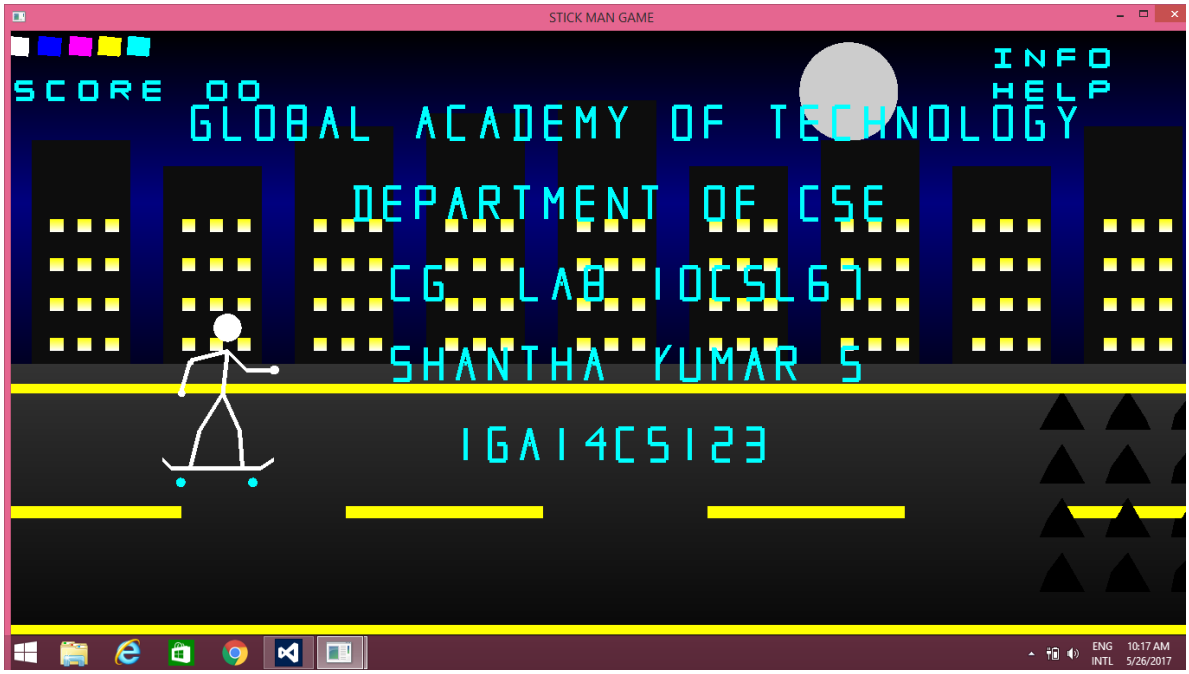

Figure 5.4 Information of the project.

CONCLUSION

The project “3D-Structures Orientation in OpenGL” was designed and implemented by solely by its creators as the exercise of “Computer Graphics and Visualization” Laboratory. The concepts used in the design are that of our own, but certain referrals were made to my lecturers and others regarding some technical issues. More features to make it a competitive. Further work on projects like this would enable greater knowledge in OpenGL.

This project has met its objectives to produce a good scene. The development of this project was very helpful to develop our programming skills and to know Visual C++ better. It helped us to learn about computer graphics, design of graphical interfaces, interface to the user, user interaction handling and mouse coordinates. The experience gained during the course of developing this project will prove helpful in future endeavors.

.

BIBLIOGRAPHY

Books

[1]. Edward Angel, "Interactive Computer Graphics", 5th edition, Addison-Wesley, 2008

[2]. F.S. Hill, Jr.: Computer Graphics Using OpenGL, 2nd Edition, Pearson Education, 2001

[3]. "Interactive Computer Graphics a top down approach with openGL"- Edward Angel, 5thEdition, Addison-Wesley, 2008.

[4]. "OpenGL Programming Guide", 6th edition, Addison-Wesley Publishing Company.

[5]. "OpenGL Reference Manual", release-1, Addison-Wesley Publishing Company.

Websites

[6]. http://www.opengl.org/sdk/docs/man/xhtml/glRasterPos.xml

[7]. http://www.swiftless.com/opengltuts.html

[8]. http://www.3dcodingtutorial.com/Basic-OpenGL-functions/Translate-and-Rotate-functions.html