Project Reports

Guidede by:- Vaibhav Vasani
Members: Nileshkumar Jha - 1711086
Ansh Bandral - 1911067
Shubham K Dathia - 1911072

INTRODUCTION

Computer graphics is concerned with all aspects of producing images using a computer. The field began humbly 50 years ago, with the display of few lines on a cathode ray tube (CRT); now, we can create images by computer that are indistinguishable from photographs of real objects.
In this chapter we discuss about the application of computer graphics, overview of the graphic system, graphics architectures. In this project we discuss about OpenGL- open graphics library API- which is used to develop the application program which is the matter
in question.

APPLICATIONS OF COMPUTER GRAPHICS

The applications of computer graphics are many and varied. However we can divide them into four major areas:

- Display of Information
- Design
- Simulation and Animation
- User interfaces

Display of Information

Classical graphics techniques arose as a medium to convey information among people. Although spoken and written languages serve this purpose, images are easier tocommunicate with and hence the computer graphics plays an important role.Information visualization is becoming increasingly important in the field of security, medicine weather forecasting and mapping. Medical imaging like CT, MRI, ultrasound scan are produced by medical imaging systems but are displayed by computer graphics
system.

Design

Professions such as engineering and architecture are concerned with design which is an iterative process. The power of the paradigm of humans interacting with the computer displays was recognized by Ivan Sutherland.

Simulation and Animation

Graphics system, now are capable of generating sophisticated images in real time. Hence engineers and researchers use them as simulators. The most important use has been in the training of pilots.

User interfaces

User interaction with computers has become dominated by a visual paradigm that includes windows, icons, menus and pointing devices. Applications such as office suites, web browsers are accustomed to this style.

CODE SNIPPETS

```
#include<cstdio>
#include<iostream>
#include<cstdlib>
#include<GL\gl.h>
#include<GL\glu.h>
#include<GL\glut.h>
#include<ctime>
#include<windows.h>
int win1, win2;
void Write(float x, float y, float z, float scale, char *s)
{
    int i, l;
    l = strlen(s);
    glPushMatrix();
    glTranslatef(x, y, z);
    glScalef(scale, scale, scale);
    for (i = 0; i < l; i++)
        glutStrokeCharacter(GLUT_STROKE_ROMAN, s[i]);
    glPopMatrix();
}

void frontsheet()
{
    glClearColor(0, 0, 0, 1);
    glClear(GL_COLOR_BUFFER_BIT);
    glColor3f(1.0, 1.0, 0.0);
    Write(-0.50, 0.9, 1, 0.0007, (char*)"K J Somaiya College Of
Engineering");
    Write(-0.55, 0.8, 1, 0.0006, (char*)"    Department of
CSE");
    glColor3f(1.0, 0.0, 0.0);
    Write(-0.45, 0.6, 0.0, 0.0007, (char*) " 2D Helicopter
Game");
    glColor3f(1.0, 1.0, 0.5);
    Write(-0.4, -0.8, 0.0, 0.0006, (char*) "Press 'C' to
continue");
```

```
    glColor3f(1, 1, 0.0);
    Write(-1.0, 0.1, 0.0, 0.0007, (char*)" Submitted BY:");
    glColor3f(1.0, 1.0, 1.0);
    Write(-1.0, -0.03, 0.0, 0.0006, (char*) "1. Nileshkumar
Jha:1711086");
    Write(-1.0, -0.13, 0.0, 0.0006, (char*) "2. Shubham K
Dathia: 1911072");
    Write(-1.0, -0.13, 0.0, 0.0006, (char*) "3. Ansh Bandral:
1911067");
    glColor3f(1, 1, 0.0);
    Write(-1.0, -0.4, 0.0, 0.0007, (char*) " Under the guidance
of: ");
    glColor3f(1.0, 1.0, 1.0);
    Write(0.15, -0.415, 0.0, 0.0006, (char*) " 1. Vaibhav
Vasani");
    glFlush();
}

float bspd = 0.02; // block dx value

//char name[25];

float b1x = 50.0, b1y = 0;//block 1 init position

float hm = 0.0;//copter moving dy value

int i = 0, sci = 1; float scf = 1; // for increment score
score_int score_flag

char scs[20], slevel[20];
//to store score_string using itoa() and level as well
int level = 1, lflag = 1, wflag = 1; //level_flag & welcome_flag
init w/1
void init()
{
    srand(time(0));//it generates a random no each time code is
executed
    b1y = (rand() % 45) + 10;//b/w 10 to 44
    glClearColor(0.0, 0.0, 0.0, 0.0);
    glShadeModel(GL_SMOOTH);
    glLoadIdentity();
    glOrtho(0.0, 100.0, 0.0, 100.0, -1.0, .0);
}

void drawcopter();

void drawcopter() {
    glColor3f(0.5, 1.0, 0.3);
```

```
    glRectf(10, 49.8, 19.8, 44.8);//body
    glRectf(2, 46, 10, 48);//tail
    glRectf(2, 46, 4, 51);//tail up
    glRectf(14, 49.8, 15.8, 52.2);//propeller stand
    glRectf(7, 53.6, 22.8, 52.2);//propeller*/
}

void renderBitmapString(float x, float y, float z, void *font,
const char *string) {
    const char *c;
    glRasterPos3f(x, y, z);
    for (c = string; *c != '\0'; c++)
    {
        glutBitmapCharacter(font, *c);
    }
}

void renderBitmapString(float x, float y, float z, void *font,
const char*string);

void display(void)
{
    glClear(GL_COLOR_BUFFER_BIT);
    //GameOver Checking
    if((i == 730 || i == -700)
       //top and bottom checking

       ||
       (((int)b1x == 10 || (int)b1x == 7 || (int)b1x == 4 ||
(int)b1x == 1) && (int)b1y < 53 + (int)hm && (int)b1y + 35>53 +
(int)hm)
       // propeller front checking

       ||
       (((int)b1x == 9 || (int)b1x == 3 || (int)b1x == 6) &&
(int)b1y < 45 + (int)hm && (int)b1y + 35>45 + (int)hm)
       //lower body checking

       ||
       (((int)b1x == 0) && (int)b1y < 46 + (int)hm && (int)b1y +
35>46 + (int)hm))
        // lower tail checking
    {
        glColor3f(0.0, 0.0, 1.0);
        glRectf(0.0, 0.0, 100.0, 100.0);
        glColor3f(1.0, 0.0, 0.0);
        renderBitmapString(40, 70, 0, GLUT_BITMAP_HELVETICA_18,
"GAME OVER!!!");
```

```
        glColor3f(1.0, 1.0, 1.0);
        renderBitmapString(30, 58, 0,
GLUT_BITMAP_TIMES_ROMAN_24, "THANKS FOR PLAYING THE GAME!!");
        //renderBitmapString(45, 58, 0,
GLUT_BITMAP_TIMES_ROMAN_24, "scored:");
        renderBitmapString(70, 58, 0,
GLUT_BITMAP_TIMES_ROMAN_24, scs);
        glutSwapBuffers();
        glFlush();
        printf("\nGAME OVER\n\n");
        system("pause");
        //printf("%s\You scored  %s", name, scs);
        printf("\n\nClose the console window to exit...\n");
        exit(0);
        //getch();
    }

    else
    {
        //on every increase by 50 in score in each level
        if (sci % 50 == 0 && lflag == 1)
        {
            lflag = 0; //make level_flag=0
            level++;//increase level by 1
            bspd += 0.01;//increase block_dx_speed by 0.01
        }
            //within every level make level_flag=1
        else if (sci % 50 != 0 && lflag != 1)
        {
            lflag = 1;
        }
        glPushMatrix();
        glColor3f(0.0, 0.5, 0.7);
        glRectf(0.0, 0.0, 100.0, 10.0);      //ceil
        glRectf(0.0, 100.0, 100.0, 90.0);    //floor

        glColor3f(0.0, 0.0, 0.0);   //score
        //  renderBitmapString(1, 3, 0,
GLUT_BITMAP_TIMES_ROMAN_24, "Distance:");
        //glColor3f(0.7,0.7,0.7);

        printf(slevel, "%d", level);   //level
        //  renderBitmapString(80, 3, 0,
GLUT_BITMAP_TIMES_ROMAN_24, "Level:");
        renderBitmapString(93, 3, 0, GLUT_BITMAP_TIMES_ROMAN_24,
slevel);
```

```
        scf += 0.025;               //so less as program run very
fast
        sci = (int)scf;
        printf(scs, "%d", sci);
        //from int to char convertion to display score
        renderBitmapString(20, 3, 0, GLUT_BITMAP_TIMES_ROMAN_24,
scs);
        glTranslatef(0.0, hm, 0.0);
        // hm(=dy) changes occur by mouse func
        drawcopter();
        //code for helicopter
        //if wall move towards left & get out of projection
volume
        if (b1x < -10)
        {
            b1x = 50;            //total width is 50
            b1y = (rand() % 25) + 20;
            //10 for selling+10 for giving enough space
            // block bottom limit 0+20 & top limit 24+20=44
        }

        else
            b1x -= bspd;
        //within the projection volume dec its x value by
block_speed

        glTranslatef(b1x, -hm, 0.0);

        glColor3f(1.0, 0.0, 0.0);
        glRectf(b1x, b1y, b1x + 5, b1y + 35);//block 1

        glPopMatrix();

        glutSwapBuffers();
        glFlush();
    }
}

void moveHeliU(void)
{

    hm += 0.05;
    i++;
    glutPostRedisplay();

}

void moveHeliD()
```

```
{

    hm -= 0.05;
    i--;
    glutPostRedisplay();

}
void mouse(int button, int state, int x, int y)
{
    switch (button)
    {
        case GLUT_LEFT_BUTTON:

            if (state == GLUT_DOWN)
                glutIdleFunc(moveHeliU);

            else if (state == GLUT_UP)
                glutIdleFunc(moveHeliD);
            break;
        default: break;
    }
}
void keys(unsigned char key, int x, int y)
{
    if (key == 'w') glutIdleFunc(moveHeliU);
    if (key == 'm') glutIdleFunc(moveHeliD);
}
void keyboards(unsigned char key, int x4, int y4)
{
    if (key == 'c' || key == 'C')
    {
        glutDestroyWindow(win1);
        glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB |
GLUT_DEPTH);
        win2 = glutCreateWindow("2D Helicopter Game");
        glClearColor(0.0, 0.0, 0.0, 0.0);
        glFlush();
        glutDisplayFunc(display);
        gluOrtho2D(-1000, 1000, 0, 1000);
        init();
        glutMouseFunc(mouse);
        glutKeyboardFunc(keys);
    }
}
int main(int argc, char** argv)
{
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
```

```
    glutInitWindowSize(800, 600);
    glutInitWindowPosition(200, 200);
    win1=glutCreateWindow("Mini Project");
    glFlush();
    glutDisplayFunc(frontsheet);
    glutKeyboardFunc(keyboards);
    glutMainLoop();
}
```

REFERENCES

[1]. Interactive Computer Graphics: A Top- down Approach Using OpenGL, Fifth Edition by Edward Angel, Pearson education, 2009.

[2]. Computer Graphics with OpenGL, Third Edition, by Hearn & Baker, Pearson education.

[3]. http://www.cs.uccs.edu/~ssemwal/glman.html

[4]. http://www.opengl.czweb.org/ewtoc.html

[5]. http://www.opengl.org

[6]. http://www.en.wikipedia.org/wiki/OpenGL

[7]. https://www.opengl.org/documentation/specs/glut/spec3/node49.html

CHAPTER 6

CONCLUSION

6.1 Conclusion of the Project

We have successfully implemented a simple game in this project using OpenGL. OpenGl supports enormous flexibility in the design and the use of OpenGl graphics programs. The presence of many built in classes methods take care of much functionality and reduce the job of coding as well as makes the implementation simpler. This game shows the use of computer graphics throughout an application especially when it comes to interaction of computers with humans. In this program, we saw how alphabetical
characters and stored data like scores are rendered on screen. We also saw how the game or animated characters- like the helicopter in this game- are created, and how objects can be moved from one co-ordinate position to another representing the movement of the object.