# Department of Computer Science and Engineering

## Shivanirudh S G, 185001146, Semester VII

19 October 2021

---

## UCS1712 - Graphics and Multimedia Lab

---

### Exercise 10: 3-D Scene in C++ using OpenGL

**Objective:**

Write a C++ program using Opengl to draw atleast four 3D objects. Apply lighting and texture and render the scene. Apply transformations to create a simple 3D animation.

**Code:**

```
1  #ifndef LOPENGL_H
2  #define LOPENGL_H
3
4  #include <GL/freeglut.h>
5  #include <GL/gl.h>
6  #include <GL/glu.h>
7  #include <math.h>
8  #include <stdio.h>
9  #include<iostream>
10 #include<vector>
11 #include<ctime>
12 #include <unistd.h>
13 using namespace std;
14
15 #endif
```

```
1  #ifndef LUTIL_H
2  #define LUTIL_H
3
4  #include "Headers.h"
5
6  //Screen Constants
7  const int SCREEN_WIDTH = 640;
8  const int SCREEN_HEIGHT = 480;
9  const int SCREEN_FPS = 60;
10 const int POINT_SIZE=2;
11
12 int INC = 1;
13
14 void initGL();
15
16 void render(int state);
17
18 void keyboardKeys(unsigned char key, int x, int y);
19
20 void drawAxes();
21
22
23 #endif
```

```
1  #include "Signatures.h"
2
3  void initGL(){
4      glClearColor(0.0, 0.0, 0.0, 1.0);
5      glShadeModel(GL_SMOOTH);
6      GLfloat light_diffuse[] = { 1.0, 1.0, 1.0, 1.0 };
7      GLfloat light_position[] = { 1, 0, 1, 0 };
8      glLightfv(GL_LIGHT0, GL_DIFFUSE, light_diffuse);
9      glLightfv(GL_LIGHT0, GL_POSITION, light_position);
10     glEnable(GL_LIGHTING);
11     glEnable(GL_LIGHT0);
12     glEnable(GL_DEPTH_TEST);
13 }
14
15
16 void render(int state){
17     if (state == 0)
18         INC = 1;
19     else if (state == 10)
20         INC = -1;
21     glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
22     glLoadIdentity();
23     gluLookAt(0.0, 1.0, 7.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0);
24     glMatrixMode(GL_MODELVIEW);
25     // Cube
26     glPushMatrix();
27     GLfloat cube_color[] = { 0.85, 0.078, 0.23, 1.0 };
28     glMaterialfv(GL_FRONT, GL_DIFFUSE, cube_color);
29     glScalef(0.4*state, 0.35*state, 1.0);
30     glTranslatef(0.4, -1.0, 0.0);
31     glutSolidCube(0.8);
32     glPopMatrix();
33     // Teapot
34     glPushMatrix();
```

```
35      GLfloat teapot_color[] = { 0,1,0, 0.0 };
36      GLfloat mat_shininess[] = { 100 };
37      glMaterialfv(GL_FRONT, GL_DIFFUSE, teapot_color);
38      glMaterialfv(GL_FRONT, GL_SHININESS, mat_shininess);
39      glTranslatef(3.0, 2, 0.0);
40      glRotatef(-0.2 * state, 0, 0, 1);
41      glutSolidIcosahedron();
42      glPopMatrix();
43      // Ramp
44      glPushMatrix();
45      GLfloat ramp_color[] = { 0, 0.74, 1, 1.0 };
46      glMaterialfv(GL_FRONT, GL_DIFFUSE, ramp_color);
47      glTranslatef(-1.2, -0.2*state, 0);
48      glutSolidTorus(0.3, 0.7, 10, 10);
49      glPopMatrix();
50
51      glPushMatrix();
52      GLfloat ball_color[] = { 0.29, 0, 0.50, 1.0 };
53      glMaterialfv(GL_FRONT, GL_DIFFUSE, ball_color);
54      glRotatef(-0.1 * state, 0, 0, 1);
55      glTranslatef(-2.5 - 0.25*state, -2, 0);
56      glutSolidRhombicDodecahedron();
57      glPopMatrix();
58      glutSwapBuffers();
59      glutTimerFunc(1000 / 30, render, state + INC);
60  }

1   #include "Helpers.h"
2
3   void reshape(int w, int h);
4   void runMainLoop();
5
6
7   int main(int argc, char *args[])
8   {
9
10      glutInit( &argc, args );
11
12      glutInitContextVersion( 2, 1 );
13
14      glutInitDisplayMode( GLUT_SINGLE|GLUT_RGB );
15      glutInitWindowSize( SCREEN_WIDTH, SCREEN_HEIGHT );
16      glutCreateWindow( "OpenGL" );
17
18      initGL();
19      glutDisplayFunc(runMainLoop);
20      glutReshapeFunc(reshape);
21
22      glutMainLoop();
23
24      return 0;
25  }
26
27  void reshape(int w, int h) {
28      glViewport(0, 0, (GLsizei)w, (GLsizei)h);
29      glMatrixMode(GL_PROJECTION);
30      glLoadIdentity();
31      gluPerspective(75, 1, 1, 20);
```

```
32      glMatrixMode(GL_MODELVIEW);
33 }
34 void runMainLoop() {
35
36      glutTimerFunc(1000 / 60, render, 0);
37 }
```

**Output:**