

Name- Suryasen Singh
Enrollment No- IIT2018069

Assignment 2

9th August 2020

Original code is written below -

```
/*
 * GLUT Shapes Demo
 *
 * Written by Nigel Stewart November 2003
 *
 * This program is test harness for the sphere, cone
 * and torus shapes in GLUT.
 *
 * Spinning wireframe and smooth shaded shapes are
 * displayed until the ESC or q key is pressed. The
 * number of geometry stacks and slices can be adjusted
 * using the + and - keys.
 */
#include<windows.h>
#include<bits/stdc++.h>
#ifdef __APPLE__
#include <GLUT/glut.h>
#else
#include <GL/glut.h>
#endif

#include <stdlib.h>

static int slices = 16;
static int stacks = 16;

/* GLUT callback Handlers */

static void resize(int width, int height)
{
    const float ar = (float) width / (float) height;

    glViewport(0, 0, width, height);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    glFrustum(-ar, ar, -1.0, 1.0, 2.0, 100.0);
```

```

    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity() ;
}

static void display(void)
{
    const double t = glutGet(GLUT_ELAPSED_TIME) / 1000.0;
    const double a = t*90.0;

    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glColor3d(1,0,0);

    glPushMatrix();
        glTranslated(-2.4,1.2,-6);
        glRotated(60,1,0,0);
        glRotated(a,0,0,1);
        glutSolidSphere(1,slices,stacks);
    glPopMatrix();

    glPushMatrix();
        glTranslated(0,1.2,-6);
        glRotated(60,1,0,0);
        glRotated(a,0,0,1);
        glutSolidCone(1,1,slices,stacks);
    glPopMatrix();

    glPushMatrix();
        glTranslated(2.4,1.2,-6);
        glRotated(60,1,0,0);
        glRotated(a,0,0,1);
        glutSolidTorus(0.2,0.8,slices,stacks);
    glPopMatrix();

    glPushMatrix();
        glTranslated(-2.4,-1.2,-6);
        glRotated(60,1,0,0);
        glRotated(a,0,0,1);
        glutWireSphere(1,slices,stacks);
    glPopMatrix();

    glPushMatrix();
        glTranslated(0,-1.2,-6);
        glRotated(60,1,0,0);

```

```

        glRotated(a,0,0,1);
        glutWireCone(1,1,slices,stacks);
        glPopMatrix();

        glPushMatrix();
        glTranslated(2.4,-1.2,-6);
        glRotated(60,1,0,0);
        glRotated(a,0,0,1);
        glutWireTorus(0.2,0.8,slices,stacks);
        glPopMatrix();

        glutSwapBuffers();
    }

```

```

static void key(unsigned char key, int x, int y)
{
    switch (key)
    {
        case 27 :
        case 'q':
            exit(0);
            break;

        case '+':
            slices++;
            stacks++;
            break;

        case '-':
            if (slices>3 && stacks>3)
            {
                slices--;
                stacks--;
            }
            break;
    }

    glutPostRedisplay();
}

```

```

static void idle(void)
{

```

```

    glutPostRedisplay();
}

const GLfloat light_ambient[] = { 0.0f, 0.0f, 0.0f, 1.0f };
const GLfloat light_diffuse[] = { 1.0f, 1.0f, 1.0f, 1.0f };
const GLfloat light_specular[] = { 1.0f, 1.0f, 1.0f, 1.0f };
const GLfloat light_position[] = { 2.0f, 5.0f, 5.0f, 0.0f };

const GLfloat mat_ambient[] = { 0.7f, 0.7f, 0.7f, 1.0f };
const GLfloat mat_diffuse[] = { 0.8f, 0.8f, 0.8f, 1.0f };
const GLfloat mat_specular[] = { 1.0f, 1.0f, 1.0f, 1.0f };
const GLfloat high_shininess[] = { 100.0f };

/* Program entry point */

int main(int argc, char *argv[])
{
    glutInit(&argc, argv);
    glutInitWindowSize(640,480);
    glutInitWindowPosition(10,10);
    glutInitDisplayMode(GLUT_RGB | GLUT_DOUBLE | GLUT_DEPTH);

    glutCreateWindow("GLUT Shapes");

    glutReshapeFunc(resize);
    glutDisplayFunc(display);
    glutKeyboardFunc(key);
    glutIdleFunc(idle);

    glClearColor(1,1,1,1);
    glEnable(GL_CULL_FACE);
    glCullFace(GL_BACK);

    glEnable(GL_DEPTH_TEST);
    glDepthFunc(GL_LESS);

    glEnable(GL_LIGHT0);
    glEnable(GL_NORMALIZE);
    glEnable(GL_COLOR_MATERIAL);
    glEnable(GL_LIGHTING);

    glLightfv(GL_LIGHT0, GL_AMBIENT, light_ambient);
    glLightfv(GL_LIGHT0, GL_DIFFUSE, light_diffuse);

```

```

glLightfv(GL_LIGHT0, GL_SPECULAR, light_specular);
glLightfv(GL_LIGHT0, GL_POSITION, light_position);

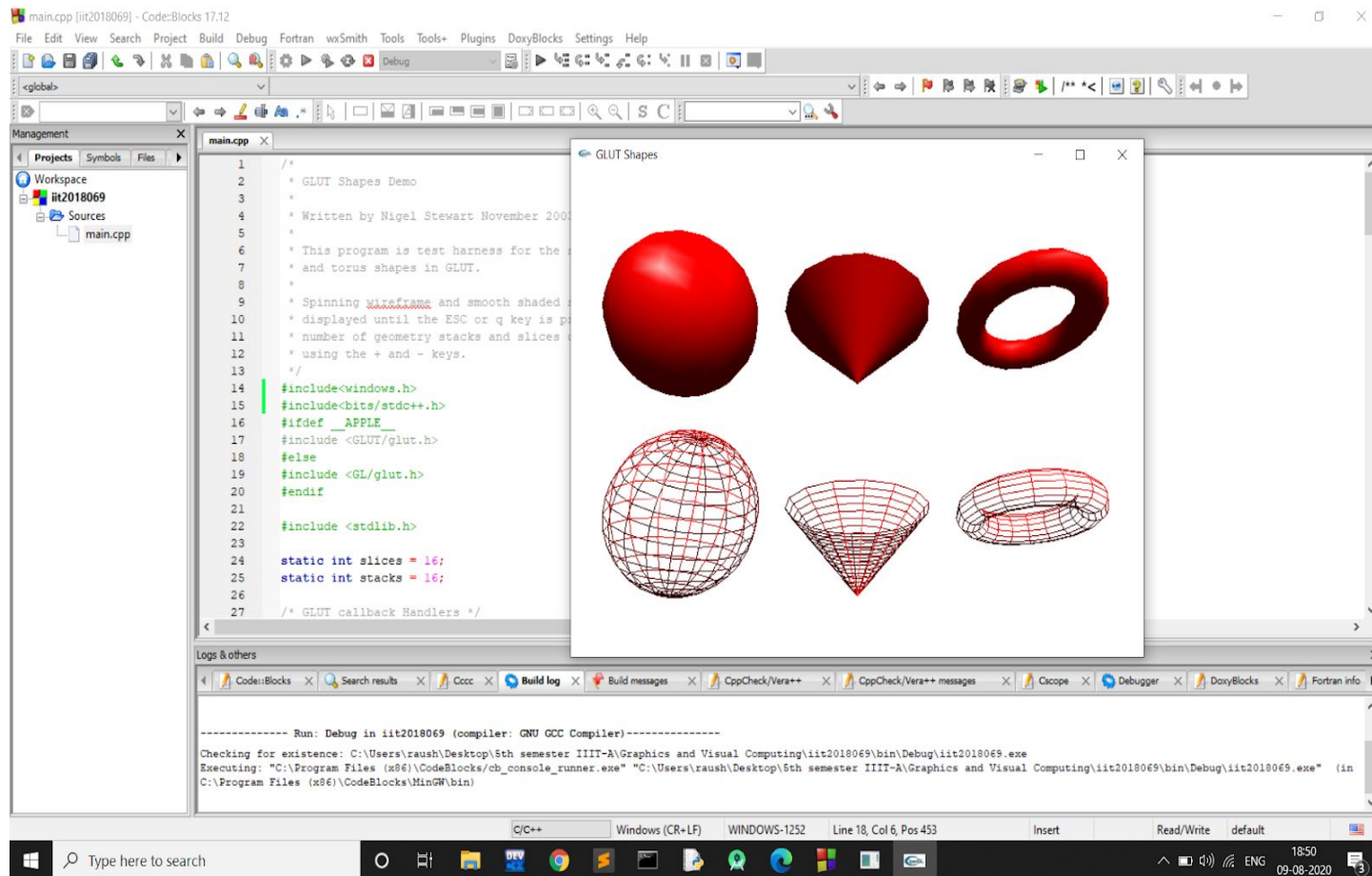
glMaterialfv(GL_FRONT, GL_AMBIENT, mat_ambient);
glMaterialfv(GL_FRONT, GL_DIFFUSE, mat_diffuse);
glMaterialfv(GL_FRONT, GL_SPECULAR, mat_specular);
glMaterialfv(GL_FRONT, GL_SHININESS, high_shininess);

glutMainLoop();

return EXIT_SUCCESS;
}

```

Output of the code written above is attached below for your reference-



Now , I've done some modification of the original code and the modified code is as follows-

```
/*
 * GLUT Shapes Demo
 *
 * Written by Nigel Stewart November 2003
 *
 * This program is test harness for the sphere, cone
 * and torus shapes in GLUT.
 *
 * Spinning wireframe and smooth shaded shapes are
 * displayed until the ESC or q key is pressed. The
 * number of geometry stacks and slices can be adjusted
 * using the + and - keys.
 */
#include<windows.h>
#include<bits/stdc++.h>
#ifdef __APPLE__
#include <GLUT/glut.h>
#else
#include <GL/glut.h>
#endif

#include <stdlib.h>

static int slices = 10;
static int stacks = 5;

/* GLUT callback Handlers */

static void resize(int width, int height)
{
    const float ar = (float) width / (float) height;

    glViewport(0, 0, width, height);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    glFrustum(-ar, ar, -1.0, 0.5, 1.0, 50.0);

    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity() ;
}
```

```

static void display(void)
{
    const double t = glutGet(GLUT_ELAPSED_TIME) / 100.0;
    const double a = t*90.0;

    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glColor3d(1,0,0);

    glPushMatrix();
        glTranslated(-2.4,1.2,-6);
        glRotated(100,1,0,0);
        glRotated(a,0,0,1);
        glutSolidSphere(1,slices,stacks);
    glPopMatrix();

    glPushMatrix();
        glTranslated(0,1.2,-6);
        glRotated(100,1,0,0);
        glRotated(a,0,0,1);
        glutSolidCone(1,1,slices,stacks);
    glPopMatrix();

    glPushMatrix();
        glTranslated(2.4,1.2,-6);
        glRotated(100,1,0,0);
        glRotated(a,0,0,1);
        glutSolidTorus(0.2,0.8,slices,stacks);
    glPopMatrix();

    glPushMatrix();
        glTranslated(-2.4,-1.2,-6);
        glRotated(100,1,0,0);
        glRotated(a,0,0,1);
        glutWireSphere(1,slices,stacks);
    glPopMatrix();

    glPushMatrix();
        glTranslated(0,-1.2,-6);
        glRotated(100,1,0,0);
        glRotated(a,0,0,1);
        glutWireCone(1,1,slices,stacks);
    glPopMatrix();
}

```

```

    glPushMatrix();
        glTranslated(2.4,-1.2,-6);
        glRotated(100,1,0,0);
        glRotated(a,0,0,1);
        glutWireTorus(0.2,0.8,slices,stacks);
    glPopMatrix();

    glutSwapBuffers();
}

```

```

static void key(unsigned char key, int x, int y)
{
    switch (key)
    {
        case 27 :
        case 'q':
            exit(0);
            break;

        case '+':
            slices++;
            stacks++;
            break;

        case '-':
            if (slices>3 && stacks>3)
            {
                slices--;
                stacks--;
            }
            break;
    }
}

```

```

    glutPostRedisplay();
}

```

```

static void idle(void)
{
    glutPostRedisplay();
}

```



```

const GLfloat light_ambient[] = { 0.0f, 0.0f, 0.0f, 1.0f };
const GLfloat light_diffuse[] = { 1.0f, 1.0f, 1.0f, 1.0f };
const GLfloat light_specular[] = { 1.0f, 1.0f, 1.0f, 1.0f };
const GLfloat light_position[] = { 2.0f, 5.0f, 5.0f, 0.0f };

const GLfloat mat_ambient[] = { 0.7f, 0.7f, 0.7f, 1.0f };
const GLfloat mat_diffuse[] = { 0.8f, 0.8f, 0.8f, 1.0f };
const GLfloat mat_specular[] = { 1.0f, 1.0f, 1.0f, 1.0f };
const GLfloat high shininess[] = { 100.0f };

/* Program entry point */

int main(int argc, char *argv[])
{
    glutInit(&argc, argv);
    glutInitWindowSize(700,500);
    glutInitWindowPosition(100,100);
    glutInitDisplayMode(GLUT_RGB | GLUT_DOUBLE | GLUT_DEPTH);

    glutCreateWindow("GLUT Shapes");

    glutReshapeFunc(resize);
    glutDisplayFunc(display);
    glutKeyboardFunc(key);
    glutIdleFunc(idle);

    glClearColor(1,1,1,1);
    glEnable(GL_CULL_FACE);
    glCullFace(GL_BACK);

    glEnable(GL_DEPTH_TEST);
    glDepthFunc(GL_LESS);

    glEnable(GL_LIGHT0);
    glEnable(GL_NORMALIZE);
    glEnable(GL_COLOR_MATERIAL);
    glEnable(GL_LIGHTING);

    glLightfv(GL_LIGHT0, GL_AMBIENT, light_ambient);
    glLightfv(GL_LIGHT0, GL_DIFFUSE, light_diffuse);
    glLightfv(GL_LIGHT0, GL_SPECULAR, light_specular);
    glLightfv(GL_LIGHT0, GL_POSITION, light_position);

```

```

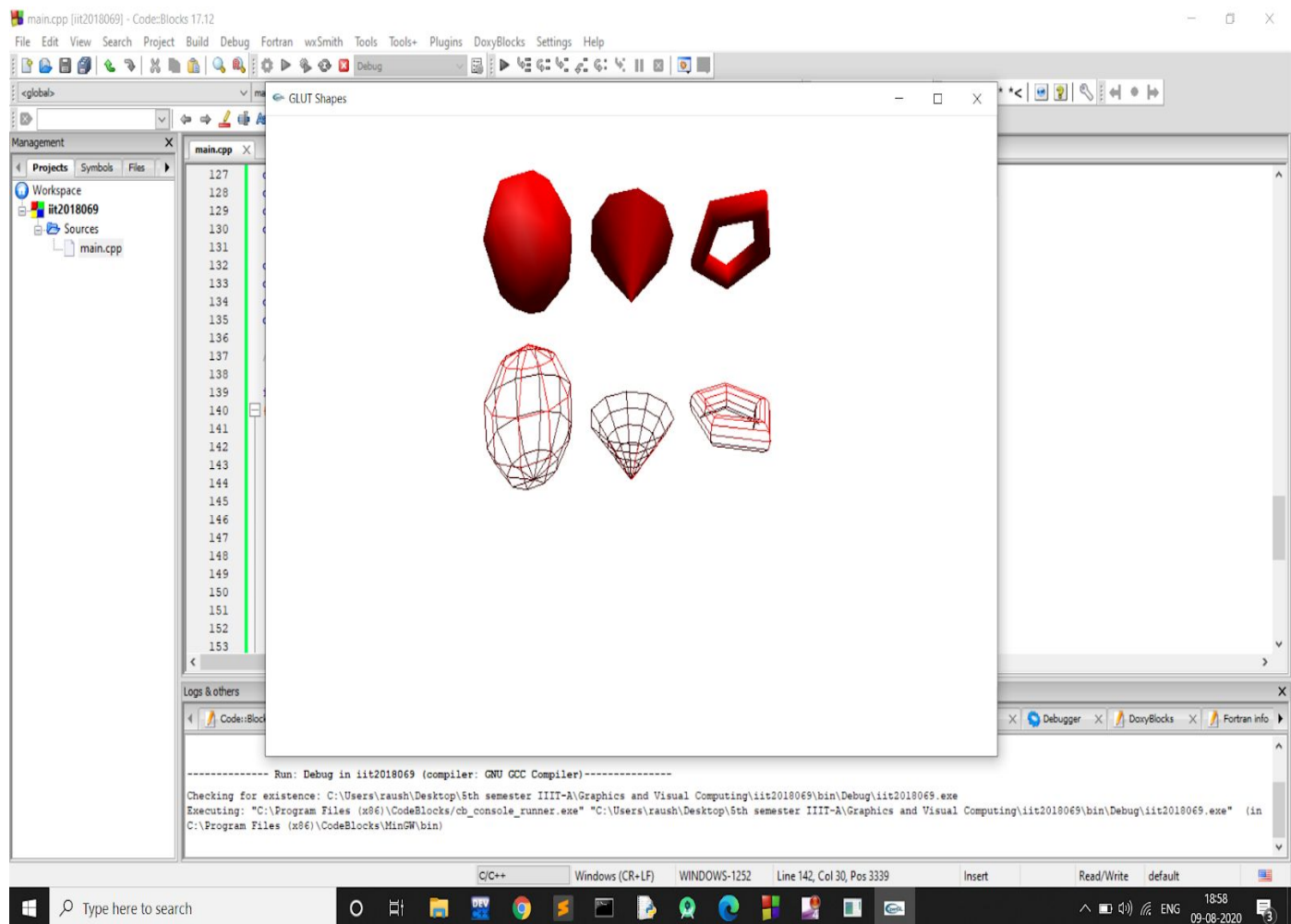
glMaterialfv(GL_FRONT, GL_AMBIENT, mat_ambient);
glMaterialfv(GL_FRONT, GL_DIFFUSE, mat_diffuse);
glMaterialfv(GL_FRONT, GL_SPECULAR, mat_specular);
glMaterialfv(GL_FRONT, GL_SHININESS, high_shininess);

glutMainLoop();

return EXIT_SUCCESS;
}

```

Output of the code written above is attached below for your reference-



Conclusion-

Based on my observation, I observed that when I modified the original code and assigned a less value to some parameters like the size of stacks and slices, I got the GLUT shapes even a little bit smaller as compared to the 1st one shape (output of the original code). In the modified code I have also increased the value of some parameters like window size and window position.

Thankyou