Name- Raushan Raj **Graphics and Visual Computing**
Enrollment No- IIT2018031 **Assignment 2** **8th August 2020**

This is the original code that was available in main.cpp of my project named “2nd assignment”. I created this project using glut project via Code Blocks.

The original codes are 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>
#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();
```

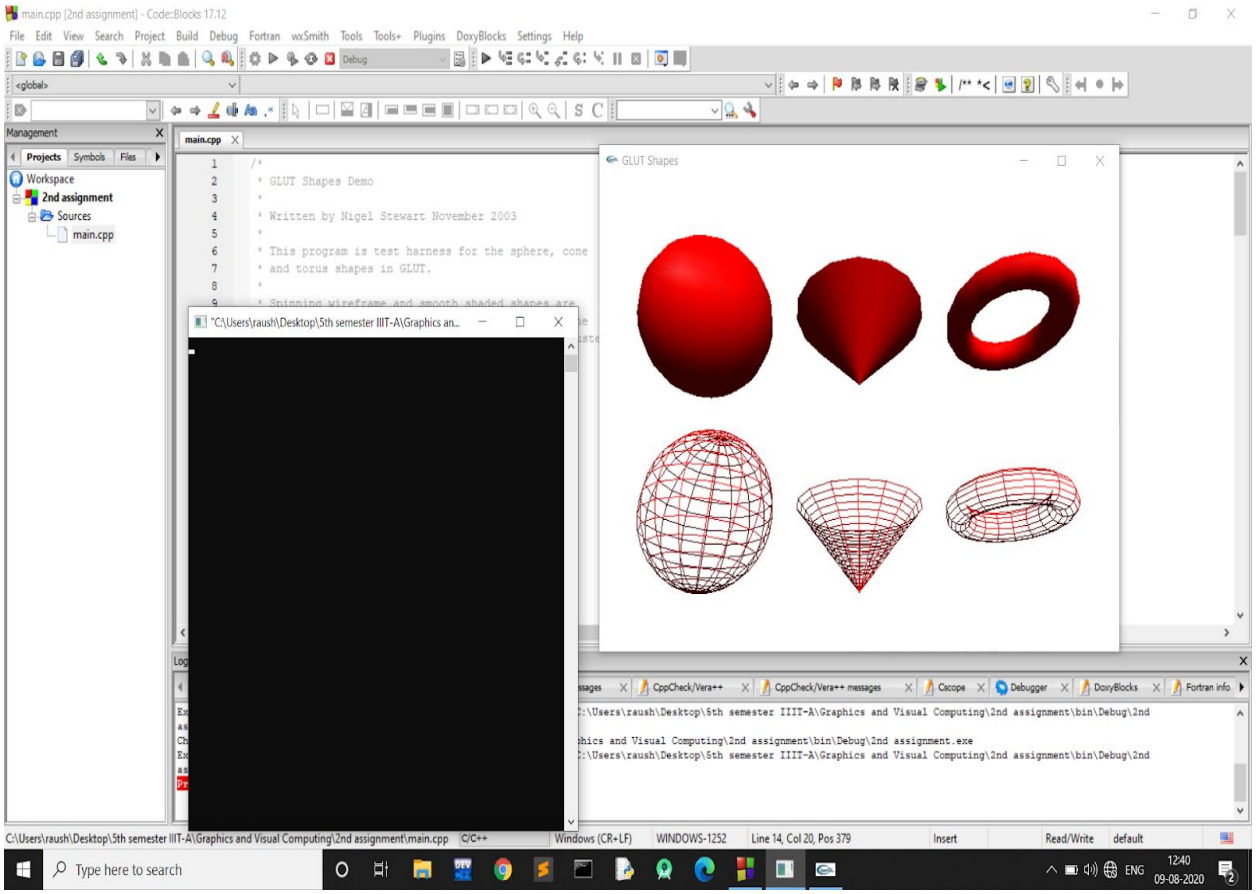
```
    glFrustum(-ar, ar, -1.0, 1.0, 2.0, 100.0);

    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;
}
```

This is the output of the above code.
You may see the GLUT shapes in the screenshot attached below.



Now this is the modified code. So what I have done here in this code is nothing but the modification of the parameter of the original code.

I have changed some parameters like the size of slices and stacks , window size,window position, matrix size (matrix rotation) and gl size etc .

For the reference, you may go through the code written below.

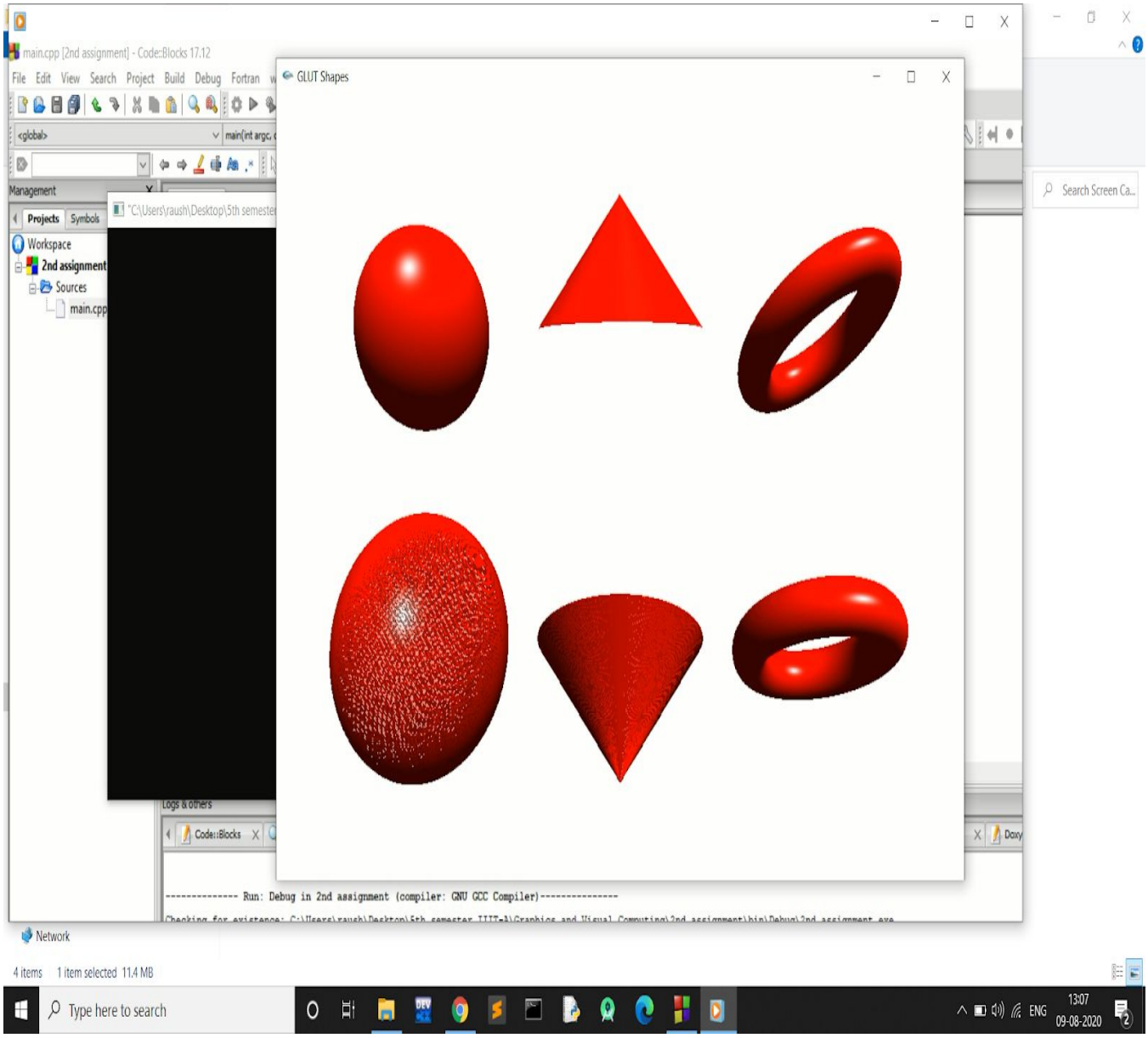
```
/*
 * GLUT Shapes Demo
 *
 * Written by Nigel Stewart November 2003
 *
 * This program is test harness for the sphere, cone
 * and torus shapes in GLUT.
 *
 * Spinning wireframe and smooth shaded shapes are
 * displayed until the ESC or q key is pressed.  The
 * number of geometry stacks and slices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

#include <stdlib.h>

static int slices = 700;
static int stacks = 400;

/* 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(40,0,0,0);
       glRotated(a,0,0,1);
       glutSolidSphere(1,slices,stacks);
    glPopMatrix();

    glPushMatrix();
       glTranslated(0,1.2,-6);
       glRotated(200,15,65,75);
       glRotated(a,0,0,1);
       glutSolidCone(1,1,slices,stacks);
    glPopMatrix();

    glPushMatrix();
       glTranslated(2.4,1.2,-6);
       glRotated(80,2,0,1);
       glRotated(a,0,0,1);
       glutSolidTorus(0.2,0.8,slices,stacks);
    glPopMatrix();

    glPushMatrix();
       glTranslated(-2.4,-1.2,-6);
       glRotated(100,2,1.5,0);
    glutInitWindowSize(925,658);
    glutInitWindowPosition(75,65);
    glutInitDisplayMode(GLUT_RGB | GLUT_DOUBLE | GLUT_DEPTH);
```

```
    glutCreateWindow("GLUT Shapes");

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

    glClearColor(20,30,40,100);
    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;
}
```

This is the output of the above code (which is the modification of original code)

You may see the GLUT shapes in the screenshot attached below.



Conclusion -

When I modified the parameters of the original code I got the solid GLUT shapes(as clearly visible in the above screenshot of the modified code). I have increased the parameters to a larger value . Things will be very clear when you compare both the images.You will clearly imagine/understand the difference between these two screenshots attached in this document.

--- The End ---