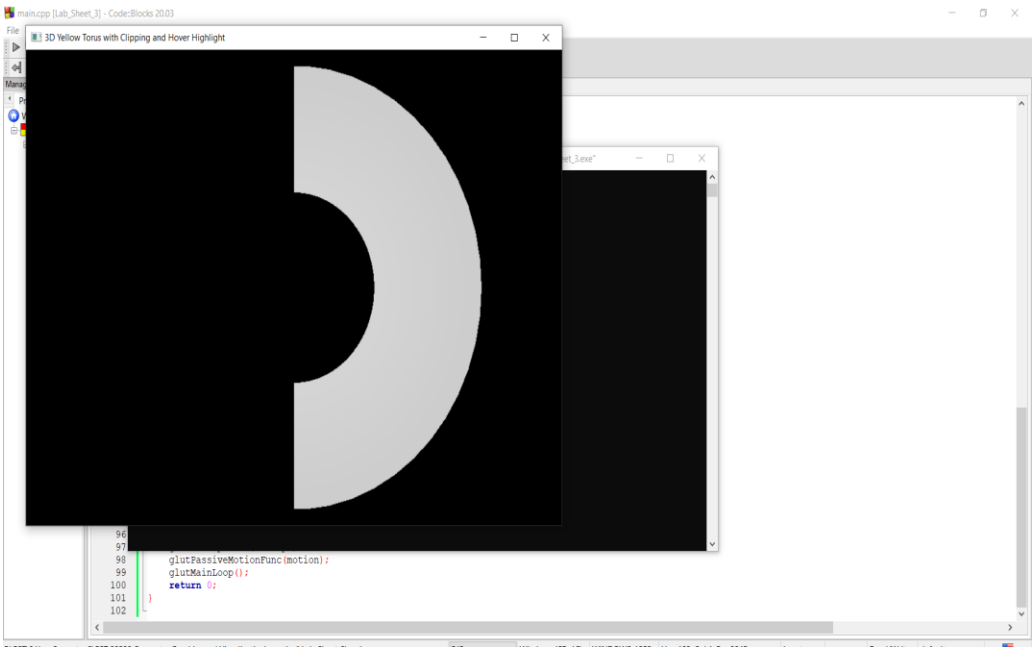
Lab Sheet No : 03
Index No : 19APP3936
Name : W.H.S.N.Rathnaweera
Date : 01.02.2025

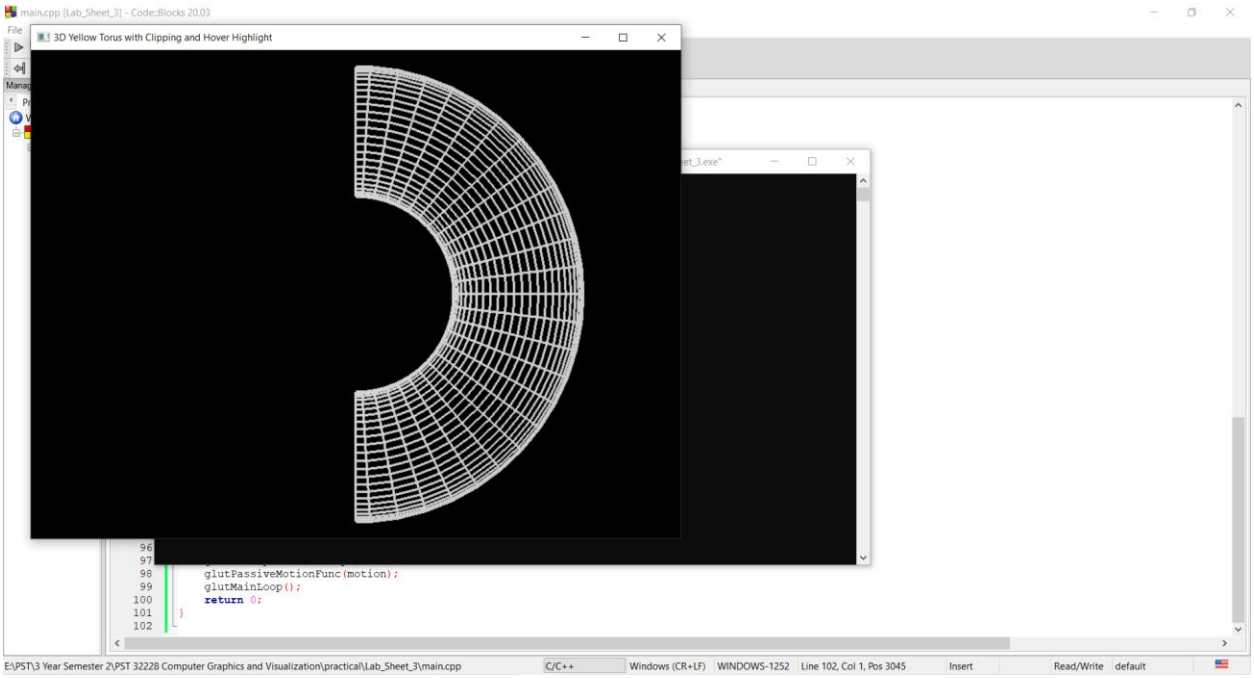
Q1.

```
  #ifdef __APPLE__
#include <GLUT/glut.h>
#else
#include <GL/glut.h>
#endif
#include <cmath>

int W = 800, H = 600;
bool isHighlighted = false;
GLdouble clipPlane[] = {1.0, 0.0, 0.0, 0.0};

void setColor() {
   // Change color to yellow (RGB: 1.0, 1.0, 0.0)
   glColor3f(1.0f, 1.0f, 0.0f);
}

void drawTorus(float innerRadius, float outerRadius, int numSides, int numRings) {
   for (int i = 0; i < numRings; i++) {
      float theta1 = (i * 2.0 * M_PI) / numRings;
      float theta2 = ((i + 1) * 2.0 * M_PI) / numRings;
      glBegin(GL_QUAD_STRIP);
      for (int j = 0; j <= numSides; j++) {
         float phi = (j * 2.0 * M_PI) / numSides;

         float x1 = (outerRadius + innerRadius * cos(phi)) * cos(theta1);
         float y1 = (outerRadius + innerRadius * cos(phi)) * sin(theta1);
         float z1 = innerRadius * sin(phi);
         setColor();  // Set color to yellow
         glVertex3f(x1, y1, z1);

         float x2 = (outerRadius + innerRadius * cos(phi)) * cos(theta2);
         float y2 = (outerRadius + innerRadius * cos(phi)) * sin(theta2);
         float z2 = innerRadius * sin(phi);
         setColor();  // Set color to yellow
         glVertex3f(x2, y2, z2);
      }
```

```
        glEnd();
    }
}

void display() {
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glLoadIdentity();
    glTranslatef(0.0f, 0.0f, -2.5f);
    glEnable(GL_CLIP_PLANE0);
    glClipPlane(GL_CLIP_PLANE0, clipPlane);

    if (isHighlighted) {
        glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
        glLineWidth(2.5);
    } else {
        glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
    }

    drawTorus(0.4, 1.0, 50, 50); // Slightly reduced size of the torus
    glDisable(GL_CLIP_PLANE0);
    glutSwapBuffers();
}

void reshape(int width, int height) {
    W = width;
    H = height;
    glViewport(0, 0, width, height);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    float aspect = (float)width / height;
    if (aspect >= 1.0f)
        glOrtho(-1.5 * aspect, 1.5 * aspect, -1.5, 1.5, -10, 10);
    else
        glOrtho(-1.5, 1.5, -1.5 / aspect, 1.5 / aspect, -10, 10);
    glMatrixMode(GL_MODELVIEW);
}

void motion(int x, int y) {
    if (x > W / 3 && x < 2 * W / 3 && y > H / 3 && y < 2 * H / 3)
        isHighlighted = true;
    else
        isHighlighted = false;
```

```
    glutPostRedisplay();
}

void initGL() {
    glEnable(GL_DEPTH_TEST);
    glEnable(GL_LIGHTING);
    glEnable(GL_LIGHT0);
    GLfloat light_pos[] = {0.0f, 0.0f, 2.0f, 1.0f};
    glLightfv(GL_LIGHT0, GL_POSITION, light_pos);
}

int main(int argc, char **argv) {
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB | GLUT_DEPTH);
    glutInitWindowSize(W, H);
    glutCreateWindow("3D Yellow Torus with Clipping and Hover Highlight");
    initGL();
    glutDisplayFunc(display);
    glutReshapeFunc(reshape);
    glutPassiveMotionFunc(motion);
    glutMainLoop();
    return 0;
}
```




main.cpp [Lab_Sheet_3] - Code::Blocks 20.03
3D Yellow Torus with Clipping and Hover Highlight
glutPassiveMotionFunc(motion);
glutMainLoop();
return 0;
}
E:\PST\3 Year Semester 2\PST 32228 Computer Graphics and Visualization\practical\Lab_Sheet_3\main.cpp
C/C++
Windows (CR+LF)
WINDOWS-1252
Line 102, Col 1, Pos 3045
Insert
Read/Write
default