

LISTA OPEN GL

```
from OpenGL.GL import *
from OpenGL.GLU import *
from OpenGL.GLUT import *
import sys

mouse_x = 0
mouse_y = 0
window_width = 800
window_height = 600
def init():
    glClearColor(0.0, 0.0, 0.0, 1.0)
    glEnable(GL_DEPTH_TEST)

    glEnable(GL_LIGHTING)
    glEnable(GL_LIGHT0)
    glEnable(GL_COLOR_MATERIAL)
    glColorMaterial(GL_FRONT_AND_BACK, GL_AMBIENT_AND_DIFFUSE)

    # Configuração da projeção
    glMatrixMode(GL_PROJECTION)
    glLoadIdentity()
    gluPerspective(45, window_width / window_height, 0.1, 50.0)
    glMatrixMode(GL_MODELVIEW)

def display():
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)
    glLoadIdentity()

    gluLookAt(0, 0, 8,
              0, 0, 0,
              0, 1, 0)
    glColor3f(0.0, 0.8, 0.9)

    glutSolidTorus(0.3, 1.0, 30, 30)

    glutSwapBuffers()
def reshape(width, height):
    global window_width, window_height
    window_width = width
    window_height = height

    glViewport(0, 0, width, height)
    glMatrixMode(GL_PROJECTION)
    glLoadIdentity()
    gluPerspective(45, width / height, 0.1, 50.0)
    glMatrixMode(GL_MODELVIEW)

def keyboard(key, x, y):
    global mouse_x, mouse_y

    key = key.decode('utf-8')

    if key == 'b' or key == 'B':
        print(f"Coordenadas do mouse: ({mouse_x}, {mouse_y})")

    elif key == 'q' or key == 'Q':
        glutLeaveMainLoop()
        sys.exit(0)
def mouse_passive_motion(x, y):
    global mouse_x, mouse_y
    mouse_x = x
```

LISTA OPEN GL

```
mouse_y = y
def main():
    glutInit(sys.argv)
    glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB | GLUT_DEPTH)
    glutInitWindowSize(window_width, window_height)
    glutInitWindowPosition(100, 100)

    glutCreateWindow(b"OpenGL - Torus 3D (Pressione 'b' para coordenadas, 'q' para sair)")

    init()
    glutDisplayFunc(display)
    glutReshapeFunc(reshape)
    glutKeyboardFunc(keyboard)
    glutPassiveMotionFunc(mouse_passive_motion)

    glutMainLoop()
if __name__ == "__main__":
    main()
```

