

Universidade Federal de Ouro Preto - UFOP
Departamento de Ciência da Computação - DECOM

Relatório atividade 1

BCC327 - COMPUTACAO GRAFICA

Kayo Xavier Nascimento Cavalcante Leite - 21.2.4095

Professor: Rafael Alves Bonfim

Ouro Preto
30 de março de 2025

Sumário

1	Código e enunciado.	1
---	---------------------	---

Lista de Figuras

1	Visualização da aplicação gráfica.	1
---	--	---

Lista de Códigos Fonte

1	Implementação Atividade 1.	1
---	------------------------------------	---

1 Código e enunciado.

Na primeira atividade, foi requerido o uso de alguma API gráfica das listadas (OpenGL, WebGL ou DirectX.) para gerar uma aplicação gráfica. Por ser apenas o início da matéria, uma aplicação simples satisfaz a demanda e para tal, escolhi fazer a representação visual de um quadrado que se adapta ao tamanho da tela.

O código da atividade se encontra anexado junto ao pdf com o relatório e, para executá-lo, o necessário instalar o python e a biblioteca Pyopengl, para tal pode-se utilizar o seguinte comando :

Instalando bibliotecas

```
no linux:  
sudo pip3 install pyopengl  
ou  
sudo apt-get install -y python-opengl  
para o windows:  
pip install pyopengl
```

Após a instalação das bibliotecas, para rodar o código basta utilizar o terminal para entrar no diretório da atividade e usar o seguinte comando:

Rodando o exercício

```
python atv1.py
```

A seguir, a visualização da tela da aplicação com o quadrado preto no fundo branco.

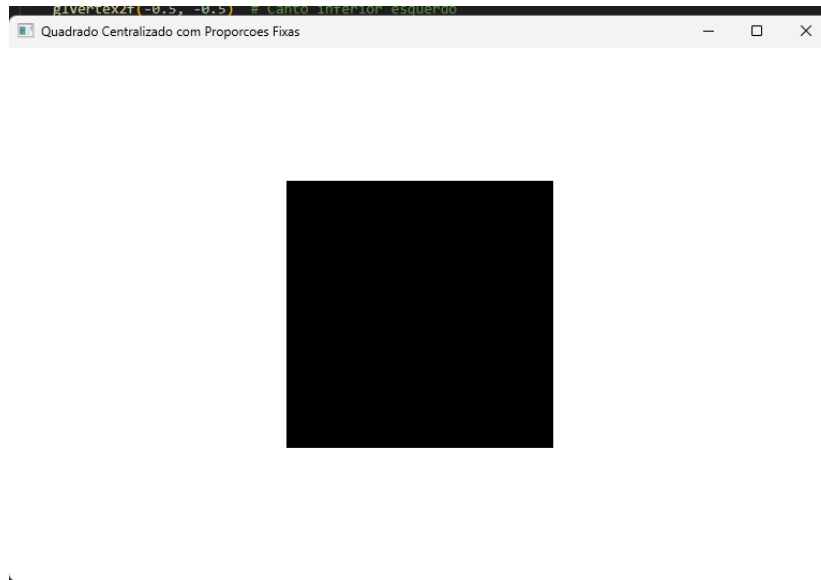


Figura 1: Visualização da aplicação gráfica.

O código se encontra todo comentado e formatado. A seguir a visualização do código completo da implementação:

```
1 from OpenGL.GL import *  
2 from OpenGL.GLUT import *  
3 from OpenGL.GLU import *  
4  
5 def init():  
6     # Configura a cor de fundo (Branco)  
7     glClearColor(1.0, 1.0, 1.0, 1.0)
```

```

8
9 def draw():
10     glClear(GL_COLOR_BUFFER_BIT)
11     glLoadIdentity()
12
13     # Define a cor do quadrado (Preto)
14     glColor3f(0.0, 0.0, 0.0)
15
16     # Desenha o quadrado centralizado
17     glBegin(GL_QUADS)
18     glVertex2f(-0.5, -0.5) # Canto inferior esquerdo
19     glVertex2f( 0.5, -0.5) # Canto inferior direito
20     glVertex2f( 0.5,  0.5) # Canto superior direito
21     glVertex2f(-0.5,  0.5) # Canto superior esquerdo
22     glEnd()
23
24     glutSwapBuffers()
25
26 def reshape(width, height):
27     # Define a viewport para cobrir toda a janela
28     glViewport(0, 0, width, height)
29     glMatrixMode(GL_PROJECTION)
30     glLoadIdentity()
31
32     # Preserva a razão de aspecto ajustando os limites da projeção ortográfica
33     if width <= height:
34         aspect = float(height) / float(width)
35         # Se a largura é o fator limitante, ajusta os limites verticais
36         gluOrtho2D(-1.0, 1.0, -1.0 * aspect, 1.0 * aspect)
37     else:
38         aspect = float(width) / float(height)
39         # Se a altura é o fator limitante, ajusta os limites horizontais
40         gluOrtho2D(-1.0 * aspect, 1.0 * aspect, -1.0, 1.0)
41
42     glMatrixMode(GL_MODELVIEW)
43
44 def main():
45     glutInit()
46     glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB)
47     glutInitWindowSize(400, 400) #define o tamanho inicial da janela
48     glutCreateWindow(b"Quadrado Centralizado com Proporções Fixas") #cria a
49     #janela (a string representa o título da aba ao ser aberta)
50     init()
51     glutDisplayFunc(draw) #desenha o quadrado através da função draw
52     glutReshapeFunc(reshape) # atualiza o tamanho do quadrado de acordo com as
53     #proporções da janela
54     glutMainLoop()
55
56 if __name__ == '__main__':
57     main()

```

Código 1: Implementação Atividade 1.