

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

Relatório atividade 2

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.	3
2	Visualização da aplicação gráfica.	3

Lista de Códigos Fonte

1	Implementação daAtividade	1
---	-------------------------------------	---

1 Código e enunciado.

Na primeira atividade, foi requerido realizar uma aplicação gráfica empregando conceitos de programação gráfica interativas: eventos, manipulação em tempo real e interatividade.

Para tal, decidi utilizar a biblioteca pygame para gerar uma aplicação que satisfaça a demanda. Foi escolhido fazer a representação visual de um quadrado e um círculo que se movimenta ao receber os inputs de teclado de usuário, atualizando sempre a sua posição de acordo com a direção utilizada.

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

Instalando bibliotecas

```
no linux:  
sudo pip3 install pygame  
para o windows:  
pip install pygame
```

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

Rodando o exercício

```
python MovimentacaoCirculo.py  
e  
python MovimentacaoQuadrado.py
```

O código se encontra todo comentado e formatado. A seguir a visualização do código completo da implementação (neste caso, para o quadrado):

```
1 import pygame  
2 import sys  
3  
4 # Kayo Xavier Nascimento Cavalcante Leite - 21.2.4095  
5  
6 # Inicializa o do Pygame  
7 pygame.init()  
8  
9 # Defini es da tela  
10 WIDTH, HEIGHT = 800, 600  
11 screen = pygame.display.set_mode((WIDTH, HEIGHT))  
12 pygame.display.set_caption("Jogo do quadrado")  
13  
14 # Defini es de cores  
15 BLACK = (0, 0, 0)  
16 WHITE = (255, 255, 255)  
17 RED = (255, 0, 0)  
18 BLUE = (0, 0, 255)  
19  
20 # Defini es do quadrado  
21 object_x, object_y = 400, 300 # Initial position  
22 object_size = 50  
23 speed = 5  
24  
25 # Main game loop  
26 running = True  
27 while running:  
28  
29     # Limpa a tela  
30     screen.fill(WHITE)  
31
```

```

32     # Captura Eventos
33     for event in pygame.event.get():
34         if event.type == pygame.QUIT:
35             running = False
36
37     # Movimenta o da bolinha
38     keys = pygame.key.get_pressed()
39     if keys[pygame.K_LEFT]:
40         object_x -= speed
41     if keys[pygame.K_RIGHT]:
42         object_x += speed
43     if keys[pygame.K_UP]:
44         object_y -= speed
45     if keys[pygame.K_DOWN]:
46         object_y += speed
47
48     # Verifica colisões com as bordas
49     object_x = max(0, min(object_x, WIDTH - object_size))
50     object_y = max(0, min(object_y, HEIGHT - object_size))
51
52     # Desenha quadrado:
53     pygame.draw.rect(screen, BLACK, (object_x, object_y, object_size,
54                                     object_size))
55
56     # Atualiza o display
57     pygame.display.flip()
58     pygame.time.Clock().tick(60) # Cap at 60 FPS
59
60 pygame.quit()
sys.exit()

```

Código 1: Implementação daAtividade .

A seguir, a visualização da tela da aplicação com o quadrado o círculo e o quadrado preto no fundo branco. Além disto, na pasta da atividade se encontra um vídeo da execução e movimentação dos objetos

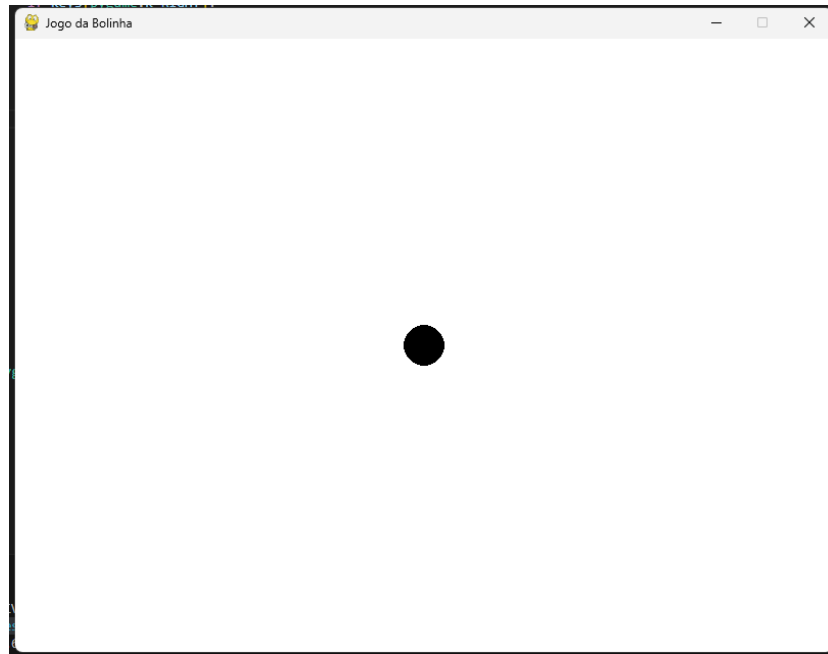


Figura 1: Visualização da aplicação gráfica.

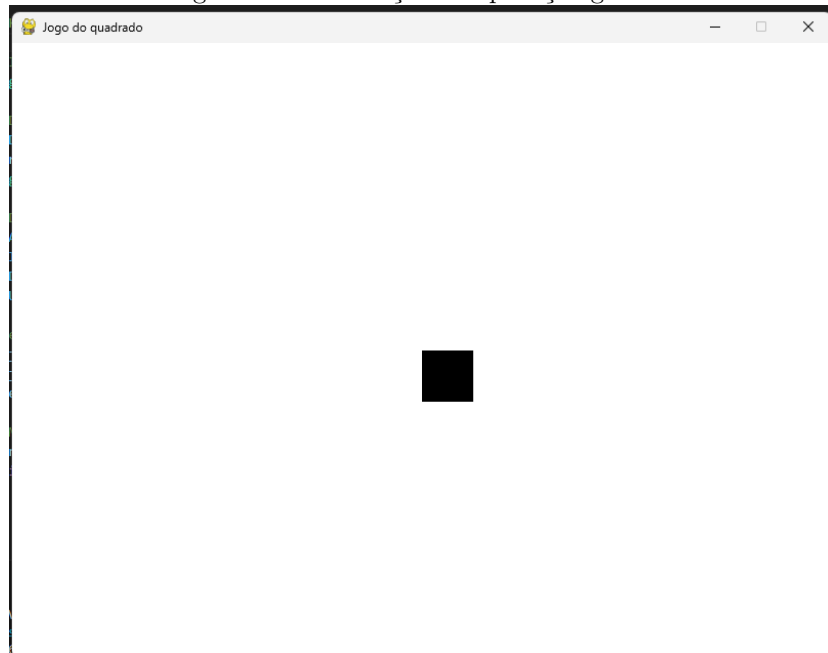


Figura 2: Visualização da aplicação gráfica.