

Nome: Felipe Braz Marques

Disciplina: BCC327 - Computação Gráfica

Professor: Rafael Alves Bonfim de Queiroz

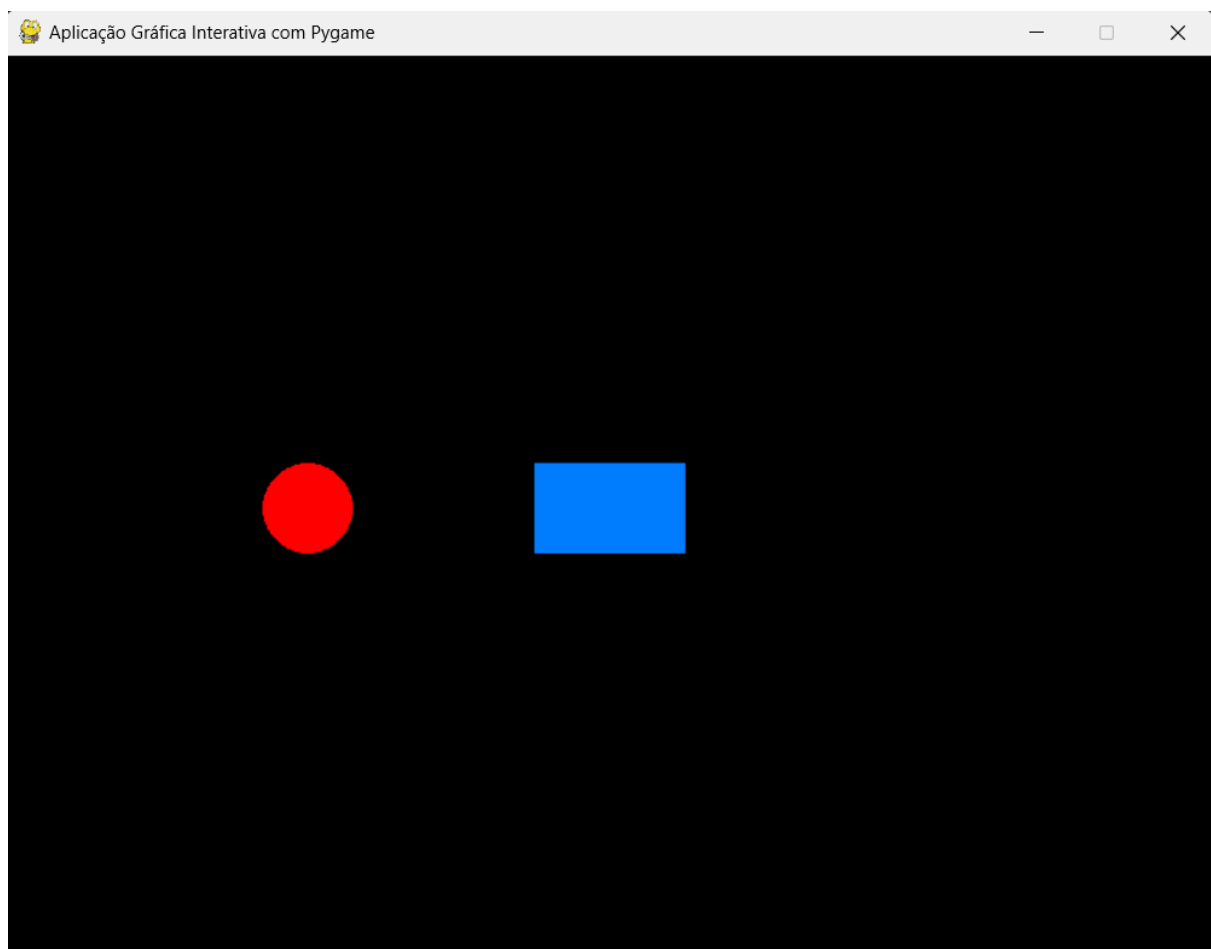
Descrição da atividade:

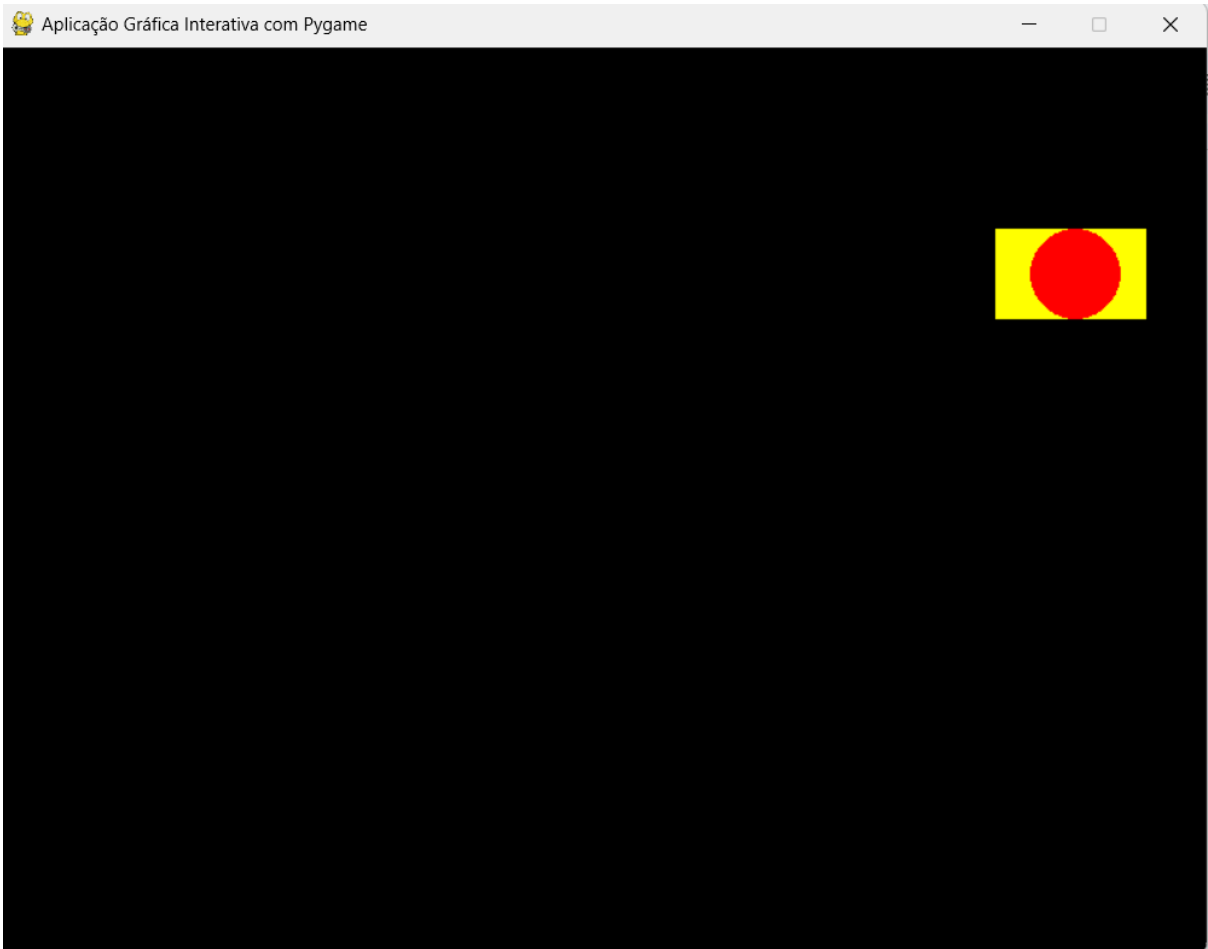
A atividade propôs o desenvolvimento de uma aplicação gráfica interativa utilizando a biblioteca pygame, explorando conceitos de programação gráfica como captura de eventos, manipulação em tempo real e interatividade. A aplicação foi projetada para demonstrar como diferentes tipos de entrada do usuário (teclado e mouse) podem ser utilizados para manipular elementos gráficos na tela de forma dinâmica.

Na aplicação desenvolvida, dois elementos principais são exibidos:

- Um retângulo azul, que pode ser movimentado usando as setas do teclado e tem sua cor alterada ao pressionar a tecla Espaço.
- Um círculo vermelho, que pode ser clicado e arrastado com o mouse em qualquer lugar da tela.

Essas interações permitem ao usuário explorar o comportamento dos elementos gráficos e sua resposta em tempo real às ações realizadas.





```

import pygame
from pygame.locals import *

# Inicializa o pygame
pygame.init()

# Configurações da tela
WIDTH, HEIGHT = 800, 600
screen = pygame.display.set_mode((WIDTH, HEIGHT))
pygame.display.set_caption("Aplicação Gráfica Interativa com Pygame")

# Configurações do retângulo
rect_width, rect_height = 100, 60
rect_x, rect_y = WIDTH // 2 - rect_width // 2, HEIGHT // 2 - rect_height // 2
rect_color = (0, 128, 255) # Azul
rect_speed = 5

# Configurações do círculo
circle_radius = 30
circle_color = (255, 0, 0) # Vermelho
circle_pos = [WIDTH // 4, HEIGHT // 2]

# Loop principal
running = True
dragging_circle = False

while running:
    for event in pygame.event.get():
        # Fecha a aplicação
        if event.type == QUIT:
            running = False

        # Eventos do mouse
        elif event.type == MOUSEBUTTONDOWN:
            mouse_x, mouse_y = event.pos
            # Verifica se clicou no círculo
            if (mouse_x - circle_pos[0]) ** 2 + (mouse_y - circle_pos[1]) ** 2 <= circle_radius ** 2:
                dragging_circle = True
        elif event.type == MOUSEBUTTONUP:
            dragging_circle = False
        elif event.type == MOUSEMOTION:
            if dragging_circle:
                circle_pos = list(event.pos)

    # Movimentação do retângulo com o teclado
    keys = pygame.key.get_pressed()
    if keys[K_UP]:
        rect_y -= rect_speed
    if keys[K_DOWN]:
        rect_y += rect_speed
    if keys[K_LEFT]:
        rect_x -= rect_speed
    if keys[K_RIGHT]:
        rect_x += rect_speed

    # Limita o retângulo dentro da tela
    rect_x = max(0, min(WIDTH - rect_width, rect_x))
    rect_y = max(0, min(HEIGHT - rect_height, rect_y))

    # Alteração de cor do retângulo com espaço
    if keys[K_SPACE]:
        rect_color = (255, 255, 0) # Amarelo

    # Atualização da tela
    screen.fill((0, 0, 0)) # Cor de fundo: preto
    pygame.draw.rect(screen, rect_color, (rect_x, rect_y, rect_width, rect_height))
    pygame.draw.circle(screen, circle_color, circle_pos, circle_radius)

```

```
pygame.display.flip()
```

```
# Encerra o pygame
```

```
pygame.quit()
```