

Atividade 6: Sistemas de Coordenadas e Movimentos de Objetos

Discente: Nicolas Expedito Lana Mendes

Matrícula: 22.1.4028

Docente: Rafael Alves Bonfim de Queiroz

Universidade Federal de Ouro Preto

Departamento de Computação

BCC327 - Computação Gráfica

Março de 2025

1 Introdução

Este relatório apresenta a implementação de uma aplicação gráfica interativa utilizando a biblioteca **Pygame**. A atividade abrange dois exercícios que demonstram o uso de sistemas de coordenadas e hierarquia de movimentos:

1. Exercício 1: Robô com Braço Animado

Um robô simples é modelado com um corpo, um braço e um antebraço. O corpo gira lentamente no sistema global; o braço oscila em torno do corpo; e o antebraço gira continuamente a partir do braço. Essa hierarquia de movimentos é calculada utilizando funções trigonométricas, demonstrando como a posição de um membro depende da posição de outro.

2. Exercício 2: Sistema Solar Simples

Um sistema solar é modelado com o Sol fixo no centro da tela, a Terra orbitando o Sol em uma trajetória circular e a Lua orbitando a Terra. Os movimentos orbitais são implementados através da atualização contínua dos ângulos, calculados com funções trigonométricas.

O usuário pode alternar entre os exercícios pressionando as teclas 1 (Robô) e 2 (Sistema Solar).

2 API Utilizada

A aplicação foi desenvolvida em **Python** utilizando a biblioteca **Pygame**. Esta biblioteca oferece suporte à renderização 2D, gerenciamento de eventos e manipulação de superfícies, permitindo a criação de animações interativas. Funções trigonométricas da biblioteca **math** são utilizadas para calcular as posições dos objetos em movimento, simulando rotações e órbitas.

3 Descrição do Código

O código foi organizado de forma a suportar dois modos de operação, cada um correspondendo a um dos exercícios.

3.1 Exercício 1: Robô com Braço Animado

- **Corpo:** Representado por um círculo, o corpo gira lentamente em torno do centro da tela. Um marcador interno indica o ângulo de rotação.
- **Braço:** O braço está fixado a uma determinada distância do centro do corpo. Ele oscila de um lado para o outro, com sua oscilação calculada através de uma função seno.
- **Antebraço:** Anexado ao final do braço, o antebraço gira continuamente, demonstrando um movimento hierárquico em que seu ângulo depende do braço.

Os cálculos são realizados da seguinte forma:

```
% Atualiza o ângulo do corpo e a oscilação do braço
body_angle += body_speed
robot_time += 1
arm_oscillation = arm_amplitude * math.sin(robot_time * 0.05)
forearm_offset += forearm_speed

% Calcula a posição de fixação do braço (ponto base)
base_x = robot_center[0] + arm_attachment_offset * math.cos(
    body_angle)
base_y = robot_center[1] + arm_attachment_offset * math.sin(
    body_angle)
arm_base = (base_x, base_y)

% Ângulo total do braço = ângulo do corpo + oscilação
total_arm_angle = body_angle + arm_oscillation
arm_end_x = arm_base[0] + arm_length * math.cos(total_arm_angle)
arm_end_y = arm_base[1] + arm_length * math.sin(total_arm_angle)
arm_end = (arm_end_x, arm_end_y)

% O antebraço gira continuamente a partir do braço
total_forearm_angle = total_arm_angle + forearm_offset
forearm_end_x = arm_end[0] + forearm_length * math.cos(
    total_forearm_angle)
forearm_end_y = arm_end[1] + forearm_length * math.sin(
    total_forearm_angle)
forearm_end = (forearm_end_x, forearm_end_y)
```

Em seguida, os componentes são desenhados: o corpo é desenhado como um círculo, e o braço e o antebraço como linhas com pontos de articulação destacados.

3.2 Exercício 2: Sistema Solar Simples

- **Sol:** Fixo no centro da tela, desenhado como um círculo amarelo.

- **Terra:** Orbita o Sol em uma trajetória circular, com sua posição calculada a partir de um ângulo que é incrementado continuamente.
- **Lua:** Orbita a Terra, com sua posição determinada pela posição da Terra e um ângulo adicional.

O cálculo das posições é feito da seguinte forma:

```
% Atualiza os ngulos de rota o
earth_angle += earth_speed
moon_angle += moon_speed

% Posi o da Terra
earth_x = sun_center[0] + earth_orbit_radius * math.cos(
    earth_angle)
earth_y = sun_center[1] + earth_orbit_radius * math.sin(
    earth_angle)
earth_pos = (int(earth_x), int(earth_y))

% Posi o da Lua em rela o Terra
moon_x = earth_x + moon_orbit_radius * math.cos(moon_angle)
moon_y = earth_y + moon_orbit_radius * math.sin(moon_angle)
moon_pos = (int(moon_x), int(moon_y))
```

Em seguida, os corpos celestes são desenhados e as órbitas são indicadas com círculos de contorno.

3.3 Loop Principal e Alternância de Modos

O loop principal processa eventos, atualiza os ângulos e renderiza a cena de acordo com o modo selecionado:

```
while running:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False
        elif event.type == pygame.KEYDOWN:
            if event.key == pygame.K_1:
                mode = 1
            elif event.key == pygame.K_2:
                mode = 2
    % Atualiza e desenha os objetos de acordo com o modo
    pygame.display.flip()
    clock.tick(60)
pygame.quit()
sys.exit()
```

4 Instruções de Execução

1. Instale o Python e a biblioteca **Pygame** (por exemplo, utilizando `pip install pygame`).

2. Salve o código em um arquivo denominado `atividade6.py`.
3. Execute o script com o comando `python atividade6.py`.
4. Pressione a tecla 1 para visualizar o robô com braço animado (Exercício 1) e a tecla 2 para visualizar o sistema solar simples (Exercício 2).

5 Funcionalidade e Resultados

- **Exercício 1: Robô com Braço Animado**

O robô é composto por um corpo que gira lentamente, um braço que oscila em relação ao corpo e um antebraço que gira continuamente. A hierarquia dos movimentos é calculada com base em funções trigonométricas, evidenciando como a posição dos membros depende dos ângulos acumulados.

- **Exercício 2: Sistema Solar Simples**

O sistema solar é modelado com o Sol fixo no centro da tela, a Terra orbitando o Sol e a Lua orbitando a Terra. Os movimentos circulares são obtidos através da atualização contínua dos ângulos de rotação, demonstrando a aplicação prática de sistemas de coordenadas.

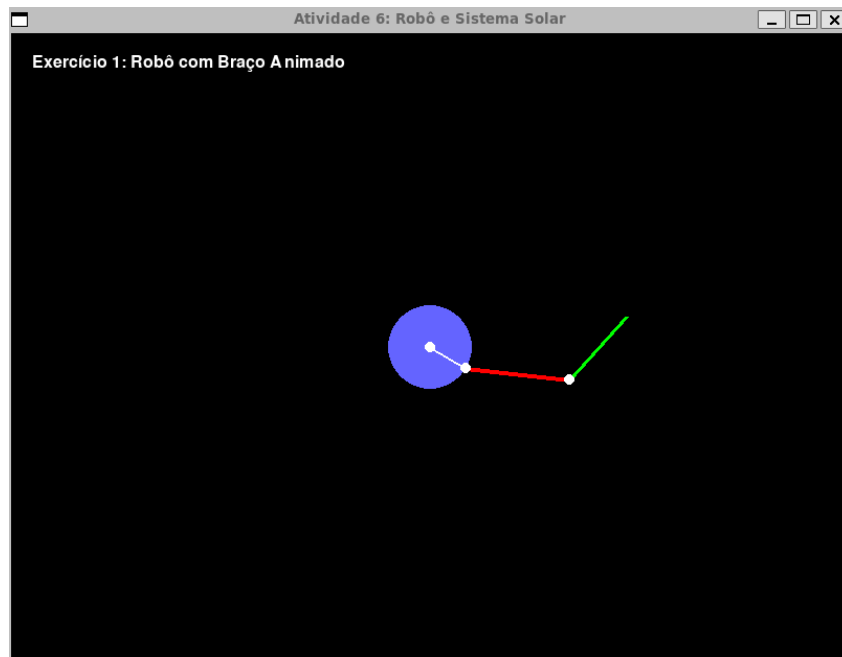


Figure 1: Exercício 1: Robô com braço animado. Parte 1

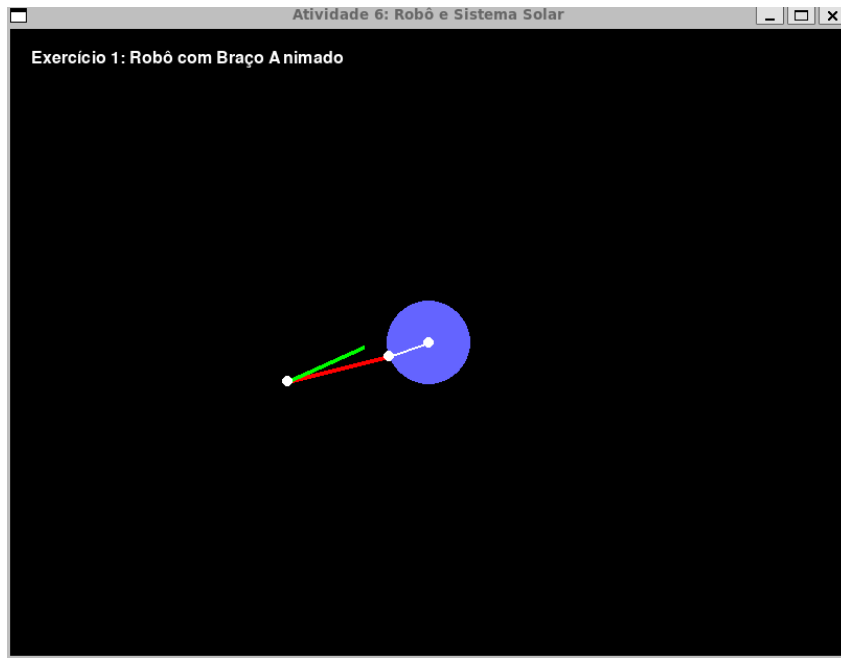


Figure 2: Exercício 1: Robô com braço animado. Parte 2

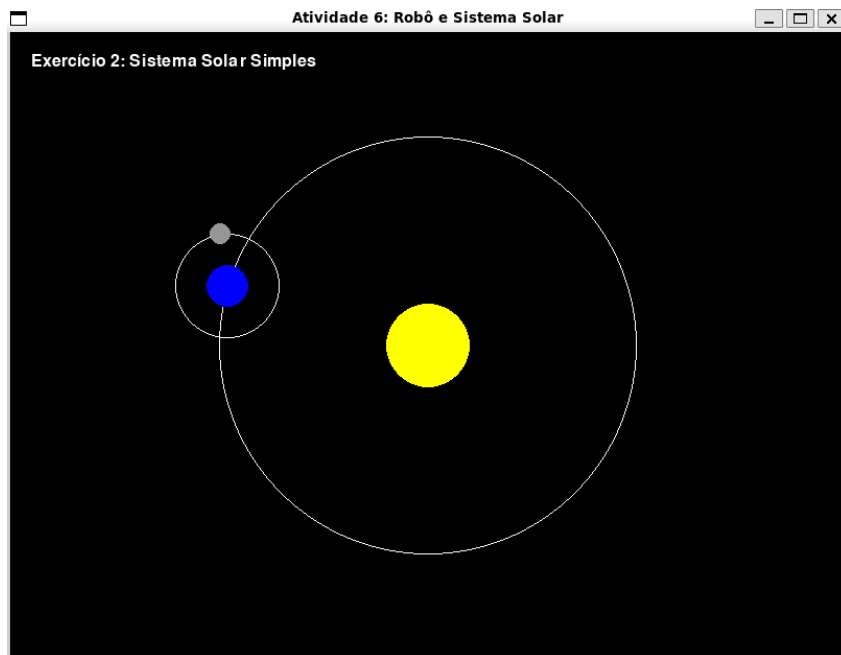


Figure 3: Exercício 2: Sistema Solar Simples. Parte 1

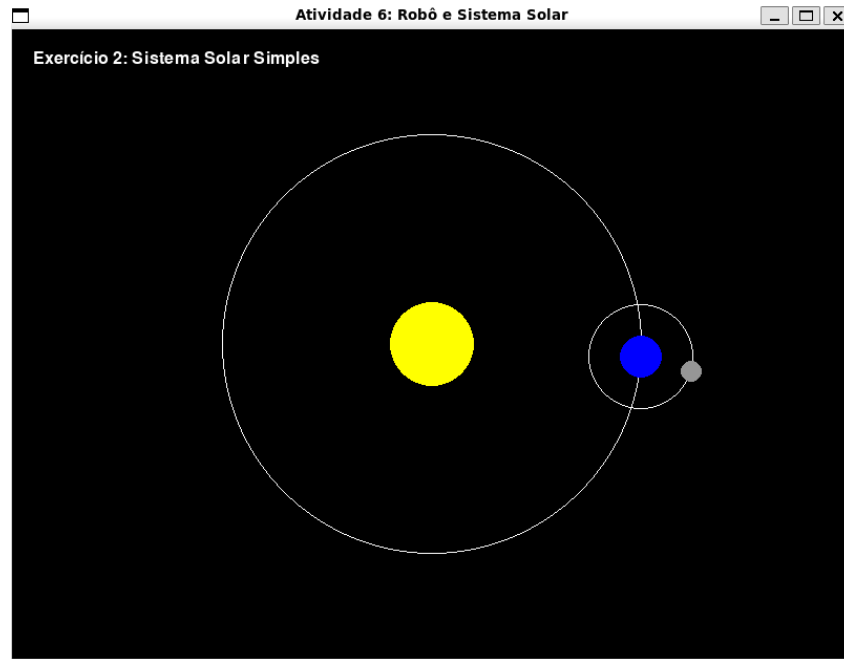


Figure 4: Exercício 2: Sistema Solar Simples. Parte 2

6 Conclusão

A implementação desta atividade demonstrou a aplicação dos conceitos de sistemas de coordenadas e hierarquia de movimentos em programação gráfica. No Exercício 1, a modelagem do robô com braço animado evidenciou como movimentos complexos podem ser simulados através de cálculos trigonométricos e da composição de ângulos. No Exercício 2, a simulação do sistema solar simples ilustrou a utilização de órbitas circulares, onde a posição dos corpos é atualizada de forma dinâmica. Ambas as implementações consolidam os fundamentos da computação gráfica e servem como base para projetos mais avançados.