



Jogando com Python



Da Lógica à Criação de Jogos Interativos

Igor Cardoso

Igor Silva de Oliveira Cardoso

Jogando com Python

Da Lógica Básica à Criação de Jogos Interativos

Brasília, 2025

SUMÁRIO

CAPÍTULO 1 - PYGAME

2.1 O QUE É O PYGAME?	80
2.2 INSTALAÇÃO	80
2.3 TESTES	80
2.4 ESTRUTURA BÁSICA DE UM JOGO EM PYGAME	80
2.5 COMANDOS UPDATE E SET_MODE	80
2.5.1 CONFIGURANDO A JANELA DO JOGO COM SET_MODE	80
2.5.2 ATUALIZANDO A TELA COM UPDATE	80
2.5 CORES E PREENCHIMENTO DA TELA	80
2.5 EVENTOS E CONTROLE DE ENTRADA	80
2.5 DESENHANDO FORMAS GEOMÉTRICAS	80
2.5 TRABALHANDO COM IMAGENS E SPRITES	80
2.5 MOVIMENTAÇÃO DE OBJETOS NA TELA	80
2.5 SONS E MÚSICAS NO JOGO	80

CAPÍTULO 2 - PYGAME

Neste capítulo, você aprenderá como instalar a biblioteca Pygame em múltiplos sistemas operacionais e também fará um teste rápido para garantir que tudo está funcionando corretamente. Dominar a instalação é o primeiro passo crucial para começar a criar jogos incríveis com seus alunos. Sem um ambiente configurado, não é possível colocar a mão na massa e explorar o potencial do Pygame. Este capítulo prático e direto ao ponto te dará a base necessária para seguir adiante com confiança.



Figura 4 - Logotipo do Paygame

O Pygame oferece uma excelente porta de entrada para o ensino de programação, combinando o aprendizado de conceitos fundamentais com a criação de projetos práticos e envolventes. Ao contrário de abordagens puramente teóricas, o desenvolvimento de jogos com Pygame motiva os alunos ao apresentar resultados visuais imediatos e a possibilidade de criar algo divertido e interativo.

2.1 O QUE É O PYGAME?

O Pygame é uma biblioteca poderosa e versátil para o desenvolvimento de jogos e aplicações multimídia em Python. Ela oferece um conjunto abrangente de ferramentas e funcionalidades que simplificam tarefas complexas, permitindo que desenvolvedores de todos os níveis criem jogos 2D incríveis com relativa facilidade. Construída sobre a biblioteca SDL (Simple DirectMedia Layer), o Pygame oferece portabilidade multiplataforma, o que significa que seus jogos podem ser executados em diversos sistemas operacionais, como Windows, macOS, Linux e até mesmo em dispositivos móveis, com poucas ou nenhuma modificação no código.

O que é uma biblioteca?

Imagine uma biblioteca física, cheia de livros. Cada livro contém informações e conhecimentos sobre um assunto específico. Em programação, uma biblioteca é como essa biblioteca, mas em vez de livros, ela contém códigos prontos que realizam tarefas específicas. Você pode 'pegar emprestado' esses códigos (usá-los no seu programa) em vez de ter que escrever tudo do zero. Isso economiza tempo e esforço.

2.2 INSTALAÇÃO

Nesta etapa, realizaremos a instalação da biblioteca Pygame. Se tudo ocorreu bem no Capítulo 1, "Preparando o Ambiente Python", você já deve ter o Python instalado e o **pip** (gerenciador de pacotes do Python) configurado. O **pip** é essencial para instalar bibliotecas externas como o Pygame.

O que é um gerenciador de pacotes?


Um gerenciador de pacotes é uma ferramenta de software que automatiza o processo de instalação, atualização, configuração e remoção de programas de computador de uma forma consistente.

Instalando o Pygame com o pip

A maneira mais recomendada e simples de instalar o Pygame é utilizando o **pip**, o gerenciador de pacotes do Python. Antes de prosseguir com a instalação do Pygame, é importante verificar se o **pip** está instalado e atualizado. Para isso, abra o terminal ou prompt de comando do seu sistema operacional e execute o seguinte comando:


```
pip --version
```

Com o ambiente Python configurado e o `pip` verificado (conforme explicado anteriormente), execute o seguinte comando para instalar o Pygame:



```
python -m pip install -U pygame --user
```


O seguinte retorno é esperado:



```
Collecting pygame
  Using cached pygame-2.6.1-cp313-cp313-win_amd64.whl.metadata (13 kB)
Downloading pygame-2.6.1-cp313-cp313-win_amd64.whl (10.6 MB)
  ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 10.6/10.6 MB 5.4 MB/s eta 0:00:00
Installing collected packages: pygame
Successfully installed pygame-2.6.1
```

2.3 TESTES

O Pygame inclui alguns exemplos que podem ser usados para testar a instalação. Para executar um deles, abra o seu terminal ou prompt de comando e execute o seguinte comando:



```
python -m pygame.examples.aliens
```

Se o exemplo for executado corretamente, você verá uma janela com um jogo simples, como ilustrado abaixo:



2.4 COMANDOS UPDATE E SET_MODE

Ao criar um jogo com Pygame, precisamos de uma janela onde tudo será exibido e de um método para atualizar essa tela constantemente. Dois comandos essenciais para isso são **set_mode**, que cria a tela do jogo, e **update**, que atualiza a tela para exibir as mudanças.

Sem esses comandos, nosso jogo não teria onde ser renderizado e os elementos gráficos não apareceriam corretamente. Neste capítulo, vamos entender como usá-los para dar vida ao nosso jogo.

2.4.1 CONFIGURANDO A JANELA DO JOGO COM SET_MODE

O primeiro passo para criar um jogo com Pygame é configurar a janela onde ele será exibido. Para isso, usamos o comando **pygame.display.set_mode()**.


Sintaxe:

```
pygame.display.set_mode((largura, altura), flags, profundidade)
```

- **(largura, altura)** → Define o tamanho da janela em pixels.
- **flags** (opcional) → Permite configurar o comportamento da janela, como tela cheia (`pygame.FULLSCREEN`).
- **profundidade** (opcional) → Define a profundidade de cores (geralmente deixamos o padrão).

Exemplo: Criando uma janela de 800x600 pixels

```
1 import pygame
2
3 pygame.init() # Inicializa o Pygame
4 tela = pygame.display.set_mode((800, 600)) # Cria a janela
5 pygame.display.set_caption("Meu Jogo com Pygame") # Define o título da janela
6
7 rodando = True
8 while rodando:
9     # Fechar a janela
10    for evento in pygame.event.get():
11        if evento.type == pygame.QUIT:
12            rodando = False
```



Você já conhece as instruções `while` e `for`? Esses são laços de repetição fundamentais, permitindo a execução de um bloco de código várias vezes, de acordo com uma condição ou uma sequência de elementos.

Para uma explicação detalhada: https://docs.python.org/pt-br/3/reference/compound_stmts.html

2.4.2 ATUALIZANDO A TELA COM UPDATE

Depois de configurar a janela, precisamos atualizar a tela para que as mudanças apareçam. É aqui que entra o comando `pygame.display.update()`.

Como funciona?

Toda vez que desenhamos algo na tela, como um personagem ou um fundo, ele não aparece imediatamente. Somente quando chamamos `update()`, a tela é atualizada para exibir as mudanças.

Sintaxe:



```
pygame.display.update( )
```

Exemplo: Desenhando um retângulo e atualizando a tela

```
1 import pygame
2
3 pygame.init()
4 tela = pygame.display.set_mode((800, 600))
5 pygame.display.set_caption("Atualizando a Tela")
6
7 rodando = True
8 while rodando:
9     for evento in pygame.event.get():
10         if evento.type == pygame.QUIT:
11             rodando = False
12
13     tela.fill((0, 0, 0)) # Preenche a tela com preto
14
15     # Desenha um retângulo vermelho
16     pygame.draw.rect(tela, (255, 0, 0), (100, 100, 200, 150))
17
18     pygame.display.update() # Atualiza a tela para mostrar o retângulo
19
20 pygame.quit()
```