Conhecendo as bibliotecas de Numpy e Pygame

Aula 18

Ana Júlia Lima

Desenvolvido para
Young 1 - sexta #5228

Ctrl Play
Brasil
Novembro 2025

Sumário

1 Bibliotecas NumPy e Pygame em Python

1.1 Introdução

Python é uma linguagem extremamente versátil — além de ser usada para ciência de dados e automação, também é muito popular para desenvolver jogos e realizar cálculos matemáticos complexos.

Existem 2 ferramentas perfeitas para projetos que unem matemática, física e programação visual, sendo elas Numpy e Pygame.

2 NumPy — Cálculos Numéricos

NumPy (Numerical Python) é usada para trabalhar com vetores e matrizes e realizar cálculos rápidos. É amplamente usada em:

- Ciência de dados e aprendizado de máquina;
- Simulações físicas e matemáticas;
- Processamento de imagens;
- Jogos e animações com cálculos de movimento.

2.0.1 Instalação e Importação

No terminal para instalar:

```
pip install numpy
```

No documento de programação para importar e colocando um apelido:

```
import numpy as np
```

2.0.2 Criando arrays

```
import numpy as np

a = np.array([1, 2, 3, 4, 5])
print(a)
```

2.0.3 Funções úteis

- `np.array()` — cria um array;
- `np.arange()` — cria uma sequência numérica;
- `np.zeros()` / `np.ones()` — arrays com 0 ou 1;
- `np.random.rand()` — gera números aleatórios;
- `np.mean()`, `np.median()`, `np.std()` — estatísticas;
- `np.sum()`, `np.max()`, `np.min()` — operações gerais.

2.0.4 Operações matemáticas

```
a = np.array([2, 4, 6])
b = np.array([1, 3, 5])

print(a + b)
print(a * b)
print(np.sqrt(a))
```

3 Pygame — Desenvolvimento de Jogos em Python

3.1 Introdução

O **Pygame** é uma biblioteca usada para criar jogos 2D e aplicações multimídia em Python. Ela oferece ferramentas para trabalhar com gráficos, imagens, sons, eventos de teclado e mouse, colisões e animações. O objetivo desta seção é apresentar os conceitos principais e ensinar como estruturar um jogo completo.

3.2 Instalação

Para instalar a biblioteca, use o seguinte comando no terminal:

```
pip install pygame
```

Para utilizá-la:

```
import pygame
```

3.3 Estrutura Básica de um Jogo

Todo jogo no Pygame segue a mesma estrutura geral:

1. Inicializar o Pygame;
2. Criar a janela;
3. Criar variáveis do jogo;
4. Executar o *game loop*;
5. Encerrar o programa.

```
import pygame

pygame.init()

tela = pygame.display.set_mode((800, 600))
pygame.display.set_caption("Meu Jogo")

rodando = True

while rodando:
    for evento in pygame.event.get():
        if evento.type == pygame.QUIT:
            rodando = False

    tela.fill((0, 0, 0))      # limpa a tela
    pygame.display.update()   # atualiza a tela

pygame.quit()
```

3.4 Controle de FPS

FPS (frames por segundo) controla a velocidade do jogo.

```
clock = pygame.time.Clock()

while rodando:
    clock.tick(60)  # limita o jogo a 60 FPS
```

3.5 Cores em Pygame (RGB)

Cores são definidas como tuplas representando `(R, G, B)`:

```
BRANCO = (255, 255, 255)
PRETO = (0, 0, 0)
VERMELHO = (255, 0, 0)
```

3.6 Desenhando na Tela

3.6.1 Retângulos

```
pygame.draw.rect(tela, VERMELHO, (x, y, largura, altura))
```

3.6.2 Círculos

```
pygame.draw.circle(tela, cor, (x, y), raio)
```

3.6.3 Linhas

```
pygame.draw.line(tela, cor, (x1, y1), (x2, y2), espessura)
```

3.7 Movimentação e Teclas

3.7.1 Capturar teclas pressionadas continuamente

```
teclas = pygame.key.get_pressed()

if teclas[pygame.K_LEFT]:
    x -= 5
if teclas[pygame.K_RIGHT]:
    x += 5
```

3.7.2 Evento de tecla apertada uma única vez

```
for evento in pygame.event.get():
    if evento.type == pygame.KEYDOWN:
        if evento.key == pygame.K_SPACE:
            print("Espaço pressionado!")
```

3.8 Eventos Importantes

- `pygame.QUIT` — fechar janela;
- `pygame.KEYDOWN` — tecla pressionada;
- `pygame.KEYUP` — tecla solta;
- `pygame.MOUSEBUTTONDOWN` — clique do mouse;
- `pygame.MOUSEMOTION` — movimento do mouse.

3.9 Carregando e Exibindo Imagens

```
jogador_img = pygame.image.load("jogador.png")
tela.blit(jogador_img, (x, y))
```

Para melhor performance:

```
jogador_img = pygame.image.load("jogador.png").convert()
```

Para imagens com transparência:

```
jogador_img = pygame.image.load("jogador.png").convert_alpha()
```

3.10 Sprites e Grupos de Sprites

Sprites representam personagens, inimigos e objetos com comportamento próprio.

```
class Jogador(pygame.sprite.Sprite):
    def __init__(self):
        super().__init__()
        self.image = pygame.Surface((50, 50))
        self.image.fill((0, 255, 0))
        self.rect = self.image.get_rect()
        self.rect.center = (400, 300)

    def update(self):
        teclas = pygame.key.get_pressed()
        if teclas[pygame.K_LEFT]:
            self.rect.x -= 5
        if teclas[pygame.K_RIGHT]:
            self.rect.x += 5
```

3.10.1 Criando grupos

```
todos = pygame.sprite.Group()
jogador = Jogador()
todos.add(jogador)

todos.update()
todos.draw(tela)
```

3.11 Colisões

3.11.1 Entre dois sprites

```
if pygame.sprite.collide_rect(sprite1, sprite2):
    print("Colisão detectada!")
```

3.11.2 Entre sprite e grupo

```
colisoes = pygame.sprite.spritecollide(jogador, inimigos, False)
```

3.12 Sons e Música

3.12.1 Música de fundo

```
pygame.mixer.music.load("musica.mp3")
pygame.mixer.music.play(-1)  # repete indefinidamente
```

3.12.2 Efeitos sonoros

```
tiro = pygame.mixer.Sound("tiro.wav")
tiro.play()
```

3.13 Animações (Frames)

Uma animação é criada alternando várias imagens:

```
frame = 0
frames = [img1, img2, img3, img4]

def animar(self):
    global frame
    frame = (frame + 1) % len(frames)
    self.image = frames[frame]
```

3.14 Física Simples

3.14.1 Gravidade

```
gravidade = 0.5
vel_y = 0

vel_y += gravidade
y += vel_y
```

3.14.2 Pulo

```
vel_y = -10
```

3.15 Exemplo de Jogo Completo

```
import pygame

pygame.init()
tela = pygame.display.set_mode((800, 600))
clock = pygame.time.Clock()

x, y = 400, 300
vel = 5

rodando = True
while rodando:
    clock.tick(60)

    for evento in pygame.event.get():
        if evento.type == pygame.QUIT:
            rodando = False

    teclas = pygame.key.get_pressed()
    if teclas[pygame.K_w]: y -= vel
    if teclas[pygame.K_s]: y += vel
    if teclas[pygame.K_a]: x -= vel
    if teclas[pygame.K_d]: x += vel

    tela.fill((20, 20, 20))
    pygame.draw.rect(tela, (0, 200, 50), (x, y, 40, 40))
    pygame.display.update()

pygame.quit()
```

3.16 Melhores Práticas

- Limite o FPS com `clock.tick()`;
- Nunca carregue imagens dentro do loop principal;
- Utilize sprites para organizar melhor o código;
- Centralize configurações (cores, tamanhos, velocidade);
- Separe arquivos (ex.: `jogo.py`, `sprites.py`);
- Use `convert()` e `convert_alpha()` para otimizar imagens.

4 Integração Pygame + NumPy

NumPy pode gerar movimentos aleatórios, partículas ou cálculos físicos avançados.

```
import numpy as np

particulas = np.random.randint(0, 800, (100, 2))

for p in particulas:
    p[1] += np.random.randint(1, 4)
    pygame.draw.circle(tela, (255,255,255), p, 2)
```

4.1 Usando NumPy com Pygame

```
import pygame
import numpy as np

pygame.init()
tela = pygame.display.set_mode((600, 400))
preto = (0, 0, 0)
azul = (0, 0, 255)

x, y = 300, 200
rodando = True

while rodando:
    for evento in pygame.event.get():
        if evento.type == pygame.QUIT:
            rodando = False

    tela.fill(preto)
    dx, dy = np.random.randint(-5, 6), np.random.randint(-5, 6)
    x += dx
    y += dy
    pygame.draw.circle(tela, azul, (x, y), 20)
    pygame.display.update()

pygame.quit()
```

5 Referências

NumPy

- Documentação oficial do NumPy: https://numpy.org/doc/
- Guia de Introdução ao NumPy (NumPy User Guide): https://numpy.org/doc/stable/user/index.html
- NumPy Quickstart Tutorial (oficial): https://numpy.org/doc/stable/user/quickstart.html
- Repositório oficial no GitHub: https://github.com/numpy/numpy
- NumPy — SciPy Lectures (material educacional): https://scipy-lectures.org/intro/numpy/index.html

Pygame

- Documentação oficial do Pygame: https://www.pygame.org/docs/
- Página oficial do projeto Pygame: https://www.pygame.org/
- Pygame Tutorials (oficial): https://www.pygame.org/wiki/tutorials
- Repositório oficial no GitHub: https://github.com/pygame/pygame
- Pygame Community Tutorials (Coleção da comunidade): https://www.pygame.org/wiki/GettingStarted