

PROGRAMAÇÃO EM PYTHON



PROGRAMAÇÃO EM PYTHON



Por meio de uma parceria com o SENAI-SP, a Prefeitura de Santana de Parnaíba está disponibilizando um curso focado em Python, com o objetivo de preparar os alunos para o desenvolvimento de programas e aplicações.



PROGRAMAÇÃO EM PYTHON



Plano de Aula

- Conteúdo:**
- Fundamentos de programação gráfica em 2D;
 - Atividades;

Início:

As informações deste conteúdo visam compreender o conteúdo do curso.



PROGRAMAÇÃO EM PYTHON



Objetivo

Capacitar profissionais para desenvolver aplicações em linguagem Python, por meio de técnicas de programação, seguindo boas práticas, procedimentos e normas.

PROGRAMAÇÃO EM PYTHON



Fundamentos de Programação Gráfica em 2D com Python



Conceitos Básicos

- **Ambiente Gráfico:** Em 2D, trabalhamos com um sistema de coordenadas (x, y) onde os elementos são posicionados e desenhados na tela.
- **Primitivas Gráficas:** São os elementos básicos como pontos, linhas, retângulos, círculos, que podem ser desenhados para compor cenas ou interfaces.
- **Eventos e Interatividade:** Programas gráficos geralmente respondem a eventos, como cliques do mouse e pressionamento de teclas, permitindo interatividade.



PROGRAMAÇÃO EM PYTHON



Fundamentos de Programação Gráfica em 2D com Python



Renderização e Atualização

- **Renderização:** Consiste em desenhar os elementos na tela a cada ciclo do programa.
- **Loop Principal:** Um loop que atualiza a tela, trata eventos e gerencia a lógica do programa, mantendo a interface responsiva e a animação fluida.



PROGRAMAÇÃO EM PYTHON



Fundamentos de Programação
Gráfica em 2D com Python



Pygame

O **Pygame** é uma biblioteca Python bastante utilizada para desenvolvimento de jogos e aplicações gráficas em 2D. Ele simplifica tarefas como criação de janelas, desenho de elementos gráficos, tratamento de eventos e reprodução de sons.

Instalação: No terminal digite `pip install pygame`

Utilização: Sempre que for utilizar a biblioteca no começo do script faça a importação, com o comando `import pygame`. O Pygame permite capturar eventos do teclado, mouse e outros dispositivos, possibilitando interatividade.

PROGRAMAÇÃO EM PYTHON



Fundamentos de Programação Gráfica em 2D com Python

Vamos
realizar
juntos!



```
import pygame

pygame.init()
largura = 800
altura = 600
tela = pygame.display.set_mode((largura, altura))
pygame.display.set_caption("Exemplo com Pygame")
rodando = True
while rodando:
    for evento in pygame.event.get():
        if evento.type == pygame.QUIT:
            rodando = False

    tela.fill((255, 255, 255))

    # Aqui você desenha os elementos gráficos
    pygame.display.update()

pygame.quit()
```


PROGRAMAÇÃO EM PYTHON



Fundamentos de Programação
Gráfica em 2D com Python



Interface do Usuário (UI)

Botões e Labels: Elementos comuns em uma interface, que podem ser desenhados como retângulos ou imagens e interagir com o usuário.

Entrada de Dados: Campos de texto e outros widgets podem ser criados para capturar informações.



PROGRAMAÇÃO EM PYTHON



Fundamentos de Programação Gráfica em 2D com Python

Vamos
realizar
juntos!



```
import pygame

pygame.init()
tela = pygame.display.set_mode((400, 300))
fonte = pygame.font.SysFont(None, 36)

def desenhar_botao(texto, posicao, cor):
    texto_render = fonte.render(texto, True, (0, 0, 0))
    retangulo = texto_render.get_rect(center=posicao)
    pygame.draw.rect(tela, cor, retangulo.inflate(20, 20))
    tela.blit(texto_render, retangulo)
    return retangulo

rodando = True
while rodando:
    for evento in pygame.event.get():
        if evento.type == pygame.QUIT:
            rodando = False
        if evento.type == pygame.MOUSEBUTTONDOWN:
            if botao.collidepoint(evento.pos):
                print("Botão clicado!")

    tela.fill((220, 220, 220))
    botao = desenhar_botao("Clique Aqui", (200, 150), (100, 200, 100))
    pygame.display.update()

pygame.quit()
```

PROGRAMAÇÃO EM PYTHON



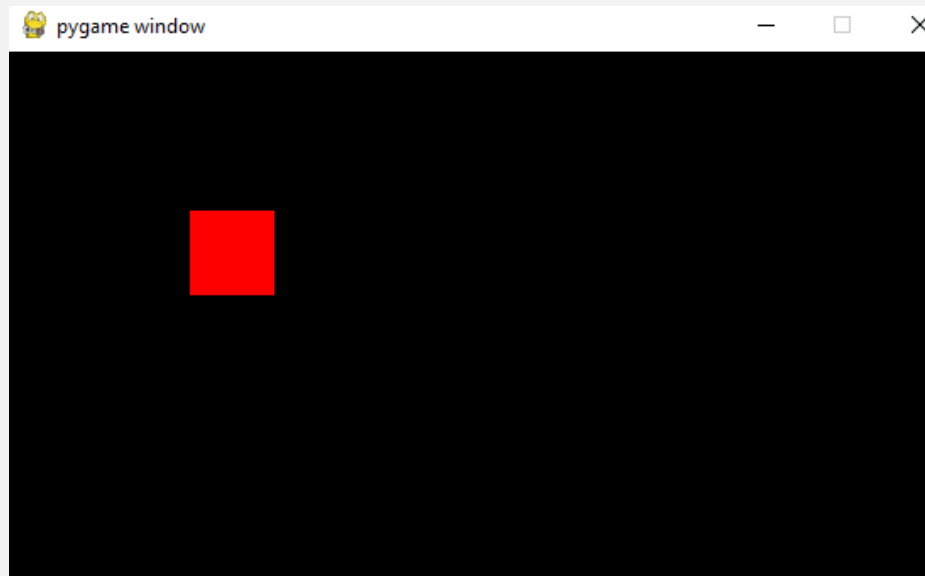
Fundamentos de Programação
Gráfica em 2D com Python



Animação

Frames e Taxa de Atualização: A animação é feita atualizando a tela em intervalos regulares (frames por segundo), criando a ilusão de movimento.

Sprites: Elementos gráficos que representam objetos animados. Sprites podem ser imagens ou formas que mudam de posição, tamanho ou aparência.



PROGRAMAÇÃO EM PYTHON



Fundamentos de Programação Gráfica em 2D com Python

Vamos
realizar
juntos!



```
import pygame

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

x = 50
y = 50
velocidade_x = 5
velocidade_y = 5
largura_ret, altura_ret = 50, 50

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

    # Atualiza a posição do retângulo
    x += velocidade_x
    y += velocidade_y

    # Inverte a direção se atingir as bordas
    if x + largura_ret > largura or x < 0:
        velocidade_x *= -1
    if y + altura_ret > altura or y < 0:
        velocidade_y *= -1

    tela.fill((0, 0, 0)) # Fundo preto
    pygame.draw.rect(tela, (255, 0, 0), (x, y, largura_ret, altura_ret))
    pygame.display.update()
    clock.tick(60) # Limita a 60 frames por segundo

pygame.quit()
```

PROGRAMAÇÃO EM PYTHON




Fundamentos de Programação Gráfica em 2D com Python



Documentação

Informações sobre a biblioteca **pygame**, você encontra aqui:
<https://www.pygame.org/docs/>

 pygame documentation	<p>Most useful stuff: Color display draw event font image key locals mixer mouse Rect Surface time music pygame</p> <p>Advanced stuff: cursors joystick mask sprite transform BufferProxy freetype gfxdraw midi PixelArray pixelcopy sndarray surfarray math</p> <p>Other: camera controller examples fastevent scrap tests touch version</p>
--	--

É importante acessar o site para obter informações sobre os comandos, métodos e instruções da biblioteca Pygame.

PROGRAMAÇÃO EM PYTHON



Fundamentos de Programação Gráfica em 2D com Python

Vamos
realizar
juntos!



PROGRAMAÇÃO EM PYTHON



Fundamentos de Programação Gráfica em 2D com Python

Vamos
realizar
juntos!



```
import pygame, random, sys
pygame.init()
largura, altura = 600, 400
tela = pygame.display.set_mode((largura, altura))
relogio = pygame.time.Clock()
tamanho = 10
x = largura // 2
y = altura // 2
vel_x = vel_y = 0
cobra = [[x, y]]
comida = [random.randrange(0, largura - tamanho, tamanho), random.randrange(0, altura - tamanho, tamanho)]
while True:
    for evento in pygame.event.get():
        if evento.type == pygame.QUIT:
            pygame.quit(); sys.exit()
        if evento.type == pygame.KEYDOWN:
            if evento.key == pygame.K_UP and vel_y == 0:
                vel_x = 0; vel_y = -tamanho
            if evento.key == pygame.K_DOWN and vel_y == 0:
                vel_x = 0; vel_y = tamanho
            if evento.key == pygame.K_LEFT and vel_x == 0:
                vel_x = -tamanho; vel_y = 0
            if evento.key == pygame.K_RIGHT and vel_x == 0:
                vel_x = tamanho; vel_y = 0
    x += vel_x; y += vel_y
    if x < 0 or x >= largura or y < 0 or y >= altura: pygame.quit(); sys.exit()
    cobra.insert(0, [x, y])
    if x == comida[0] and y == comida[1]:
        comida = [random.randrange(0, largura - tamanho, tamanho), random.randrange(0, altura - tamanho, tamanho)]
    else:
        cobra.pop()
    tela.fill((0, 0, 0))
    pygame.draw.rect(tela, (255, 0, 0), (comida[0], comida[1], tamanho, tamanho))
    for seg in cobra:
        pygame.draw.rect(tela, (0, 255, 0), (seg[0], seg[1], tamanho, tamanho))
    for seg in cobra[1:]:
        if seg == [x, y]: pygame.quit(); sys.exit()
    pygame.display.update()
    relogio.tick(15)
```

REFERÊNCIAS

ALVES, William Pereira. Lógica de programação de computadores. São Paulo: Érica, 2012.

ASCENCIO, Ana F. G.; CAMPOS, Edilene A. V. Fundamentos da programação de computadores. 3. ed. São Paulo: Pearson, 2012.

FARRELL, Joyce. Lógica e design de programação. São Paulo: Cengage Learning, 2010.

MANZANO, J. A. N. G.; OLIVEIRA, J. F. Algoritmos: Lógica para desenvolvimento de programação de computadores. 27. ed. rev. São Paulo: Érica, 2014.

MEDINA, Marco; FERTIG, Cristina. Algoritmos e programação. Teoria e prática. São Paulo: Novatec, 2005.

PEREIRA, Silvio do Lago. Algoritmos e lógica de programação em C: Uma abordagem didática. São Paulo: Érica, 2010.

Importante:

Os conteúdos disponibilizados são específicos para este curso/turma, a divulgação ou reprodução do material para outras pessoas/organização não é autorizada.



daniel.petinice@sp.senai.br



Daniel Tadeu Petinice



Instrutor de Formação Profissional III

