



Pygame

Disciplina: Introdução à Programação para Engenharias

Professor: André Smania

Alunos: Matheus Herberts Rios de Lima

N° USP: 15653174

Raphael Parra

N° USP: 15554782

Vinícius França dos Santos

N° USP: 15652322

São Carlos

18/06/2024

1. BIBLIOTECAS EM PYTHON

Bibliotecas em Python são conjuntos de código pré-escrito que fornecem uma série de funcionalidades e recursos adicionais para facilitar o desenvolvimento de programas nessa linguagem de programação. Elas são como “caixas de ferramentas” que os programadores podem utilizar para realizar tarefas específicas sem precisar escrever todo o código do zero. As bibliotecas em Python são extremamente úteis, pois permitem que os desenvolvedores economizem tempo e esforço, aproveitando soluções já prontas e testadas pela comunidade.

2. PYGAME

Pygame, trata-se de uma biblioteca voltada para o desenvolvimento de jogos em Python. Ela fornece um conjunto de módulos projetados para facilitar a criação de jogos e aplicações multimídia. Com o Pygame, é possível manipular gráficos, sons e eventos de entrada de maneira simplificada.

Pygame abstrai muitos dos detalhes complexos relacionados à manipulação de hardware e gráficos, permitindo que desenvolvedores se concentrem na lógica e na criatividade do jogo. Entre suas funcionalidades, estão:

1. **Manipulação de Gráficos:** Pygame permite desenhar formas geométricas, carregar e manipular imagens, e renderizar gráficos em 2D com facilidade. Ele suporta várias operações de desenho, como preenchimento de superfícies, transformação de imagens e aplicação de efeitos visuais.
2. **Controle de Sons:** A biblioteca oferece suporte para reprodução de sons e músicas, incluindo efeitos sonoros e trilhas sonoras. É possível manipular o volume, controlar a reprodução e fazer mixagem de áudio.
3. **Eventos de Entrada:** Pygame simplifica a captura e tratamento de eventos de entrada do usuário, como pressionamento de teclas, movimento do mouse e cliques. Isso permite a criação de interações complexas de maneira intuitiva.

4. **Gestão de Janelas e Exibição:** Com Pygame, é fácil criar e gerenciar janelas de jogo, configurar resoluções e definir modos de exibição, incluindo tela cheia e modos escaláveis.

3. APLICAÇÃO DO PYGAME

Para esse trabalho, o grupo escolheu fazer um jogo utilizando as funcionalidades do Pygame. O jogo escolhido foi o “Jogo da Cobrinha” e as seguintes funções foram utilizadas:

- **pygame.init():** Inicializa todos os módulos do Pygame. É necessário para começar a usar qualquer funcionalidade do Pygame.

- **pygame.display.set_mode((win_width, win_height)):** Configura a janela do jogo com as dimensões especificadas (largura e altura).

- **pygame.display.set_caption("Jogo da Cobrinha"):** Define o título da janela do jogo.

- **pygame.font.SysFont(None, 70) e pygame.font.SysFont(None, 35):** Cria objetos de fonte para desenhar texto na tela. O primeiro parâmetro define o tipo de fonte (None usa a fonte padrão), e o segundo define o tamanho da fonte.

- **pygame.image.load():** Carrega uma imagem de um arquivo. No caso deste código, a imagem é carregada a partir de um objeto BytesIO criado a partir do conteúdo de uma resposta HTTP.

- **pygame.transform.scale(image, (snake_block, snake_block)):** Redimensiona a imagem carregada para caber no tamanho especificado (neste caso, do tamanho de um bloco da cobra).

- **pygame.draw.rect(win, color, [x, y, width, height]):** Desenha um retângulo na superfície especificada (win) com a cor e dimensões fornecidas. Utilizado para desenhar os segmentos da cobra e a comida quando a imagem não está disponível.

□ **pygame.font.Font(None, size).render(msg, True, color):** Renderiza uma mensagem de texto na tela com a fonte e cor especificadas. O parâmetro True ativa a suavização de bordas do texto.

□ **text_surface.get_rect(center=(win_width / 2, win_height / 2 + y_displace)):** Obtém um retângulo que engloba o texto renderizado e centraliza este retângulo na tela com base nas coordenadas fornecidas.

□ **win.blit(source, dest):** Desenha uma superfície (ou parte dela) em outra superfície. Utilizado para desenhar a imagem da comida na tela do jogo.

□ **pygame.event.get():** Obtém todos os eventos da fila de eventos. Utilizado para lidar com a entrada do usuário (como pressionar teclas ou fechar a janela).

□ **pygame.KEYDOWN:** Constante que indica que uma tecla foi pressionada.

□ **pygame.K_LEFT, pygame.K_RIGHT, pygame.K_UP, pygame.K_DOWN:** Constantes que representam as teclas de seta do teclado. Utilizadas para mudar a direção da cobra.

□ **pygame.QUIT:** Constante que indica que o usuário fechou a janela do jogo.

□ **win.fill(color):** Preenche a superfície (win) com a cor especificada. Utilizado para limpar a tela antes de redesenhar os elementos do jogo.

□ **pygame.display.update():** Atualiza a tela com todas as mudanças feitas desde a última chamada. Necessário para que as alterações apareçam na janela do jogo.

□ **pygame.time.Clock():** Cria um objeto de relógio que pode ser usado para controlar a velocidade do jogo.

□ **clock.tick(snake_speed)**: Controla a taxa de quadros (FPS - frames por segundo) do jogo, definindo quantas vezes o loop principal deve ser executado por segundo.

□ **pygame.quit()**: Encerra todos os módulos do Pygame. Deve ser chamado antes de encerrar o programa para limpar corretamente todos os recursos do Pygame.

Para mais informações sobre o projeto, acessar o [GitHub](#).

4. DIFICULDADES

Não é uma biblioteca difícil de ser usada, mesmo tendo que aprende-la do zero, tivemos muito apoio de toda a documentação e das inúmeras comunidades que existem de Pygame. Sempre que ficamos em dúvida com alguma coisa, era só buscarmos no Google que alguém já tinha comentado e explicado a questão.

Então mesmo sem familiaridade com a biblioteca, foi consideravelmente simples desenvolver um jogo básico, como o da “cobrinha”.

5. CONCLUSÃO

Após utilizarmos a biblioteca para desenvolver um simples jogo, notamos que pode ser uma biblioteca muito útil para construção de jogos simples, tendo em vista que a biblioteca foi projetada para jogos em 2D e não inclui ferramentas específicas para renderização 3D. Para criação de jogos em 3D, é mais recomendável a utilização de bibliotecas como a OpenGL e a Panda3D.

6. REFERÊNCIAS

<https://awari.com.br/bibliotecas-em-python-descubra-a-quantidade-e-a-importancia/#:~:text=As%20bibliotecas%20em%20Python%20desempenham,a%20roda%20a%20cada%20projeto>

<https://www.pygame.org/docs/>

<https://github.com/mende1/guia-definitivo-de-markdown>

<https://stackoverflow.com/>

<https://chatgpt.com/>