**Pyglet**

O Pyglet é uma biblioteca Python para desenvolvimento de aplicações multimídia, especialmente jogos e outras aplicações gráficas 2D. Com o Pyglet, é possível criar janelas, desenhar gráficos e reproduzir áudio e vídeo de maneira integrada, sem a necessidade de dependências externas, pois ele oferece uma integração nativa com OpenGL, uma API amplamente utilizada para gráficos acelerados por hardware.

**Histórico e Criador**

O Pyglet foi desenvolvido por Alex Holkner em 2006. Holkner criou a biblioteca com o objetivo de fornecer uma alternativa eficiente e simples para o desenvolvimento de jogos e aplicativos multimídia em Python, destacando-se pela facilidade de uso e pela integração com OpenGL. Desde sua criação, o Pyglet tem sido popular entre desenvolvedores independentes e educadores por ser leve e fácil de aprender.

O Pyglet é uma biblioteca de multimídia cruzada para Python, que fornece uma API fácil de usar para criar aplicativos gráficos, incluindo jogos, visualizações e apresentações. Aqui estão algumas das principais características e funcionalidades do Pyglet:

**Características**

**1. Multiplataforma:**  O Pyglet é compatível com Windows, macOS e Linux.

**2. Gráficos 2D e 3D:** O Pyglet suporta gráficos 2D e 3D, incluindo renderização de sprites, texturas e modelos 3D.

**3. Áudio:** O Pyglet suporta reprodução de áudio em vários formatos, incluindo WAV, MP3 e OGG.

**4. Entrada do usuário:** O Pyglet suporta entrada do usuário por meio de teclado, mouse e joystick.

**5. Eventos:** O Pyglet fornece um sistema de eventos que permite que os aplicativos respondam a eventos do usuário, como cliques do mouse e pressionamentos de teclas.

**Principais Funcionalidades do Pyglet**

**Manipulação de Janelas:** Pyglet permite criar e manipular janelas para exibir gráficos e interagir com o usuário.

**Gráficos e Imagens:** Suporta desenho de formas geométricas e manipulação de imagens de forma simples.

**Entrada de Usuário:** Possui captura de eventos de teclado e mouse, facilitando a criação de controles para os jogos.

**Reprodução de Áudio e Vídeo:** Suporta reprodução de áudio e vídeo em diversos formatos, sendo útil para jogos que precisam de sons e efeitos visuais.

**Animações:** O Pyglet permite criar animações e transições de imagens.

**Integração com OpenGL:** Para quem deseja personalizar os gráficos de forma avançada, o Pyglet permite o uso de código OpenGL diretamente, aproveitando a renderização acelerada.

**Efeitos visuais:** O Pyglet fornece uma variedade de efeitos visuais, como sombras, reflexos e transparência.

**Integração com outras bibliotecas:** O Pyglet pode ser integrado com outras bibliotecas Python, como NumPy e SciPy.

**Utilização do Pyglet em Jogos**

Embora a biblioteca Pyglet seja mais comumente usada para projetos educacionais e pequenos jogos, não há muitos jogos famosos que utilizam exclusivamente essa biblioteca. A maioria dos jogos conhecidos tende a usar bibliotecas mais robustas como Pygame ou frameworks mais avançados.

Alguns jogos que utilizam desse framework são,:

1. Asteroids: Um clássico jogo de nave espacial onde o jogador precisa destruir asteroides.

2. Jogo da Forca: Um jogo simples onde o jogador tenta adivinhar uma palavra letra por letra.

3. Simulações 2D: Vários projetos de simulações interativas com objetos em movimento.

4. Jogos de Tiro: Jogos simples de tiro com gráficos básicos e interação do usuário.

**Exemplo de Código com Pyglet**

Abaixo está um exemplo básico de como usar o Pyglet para criar uma janela e exibir uma imagem:

import pyglet

from pyglet.window import key

janela = pyglet.window.Window(800, 600, "Movimento do Círculo")

janela.set\_location(100, 100)

posicao\_x = 400

posicao\_y = 300

raio = 30

velocidade = 200

teclas\_pressionadas = set()

@janela.event

def on\_draw():

janela.clear()

pyglet.shapes.Circle(posicao\_x, posicao\_y, raio, color=(50, 200, 50)).draw()

@janela.event

def on\_key\_press(symbol, modifiers):

if symbol in (key.LEFT, key.RIGHT, key.UP, key.DOWN):

teclas\_pressionadas.add(symbol)

@janela.event

def on\_key\_release(symbol, modifiers):

if symbol in teclas\_pressionadas:

teclas\_pressionadas.remove(symbol)

def atualizar(dt):

global posicao\_x, posicao\_y

if key.LEFT in teclas\_pressionadas:

posicao\_x -= velocidade \* dt

if key.RIGHT in teclas\_pressionadas:

posicao\_x += velocidade \* dt

if key.UP in teclas\_pressionadas:

posicao\_y += velocidade \* dt

if key.DOWN in teclas\_pressionadas:

posicao\_y -= velocidade \* dt

pyglet.clock.schedule\_interval(atualizar, 1/60.0)

pyglet.app.run()