



Universidad Gerardo Barrios

Facultad:
Ciencia y Tecnología

Carrera:
Ingeniería en Sistemas y Redes Informáticas

Asignatura:
Programación III

Tema:
Bibliotecas externas de Python

Docente:
Ing. Willian Montes

Estudiantes:
Kevin Antonio Castro Araujo
Jeremías Neftaly Fuentes Méndez
Yensi Elizabeth Valladares Ventura

Librería Pygame

Introducción

Pygame es una de las librerías más conocidas y potentes del lenguaje de programación Python, especialmente diseñada para el desarrollo de videojuegos, simulaciones, y aplicaciones multimedia interactivas. Desde su lanzamiento inicial en el año 2000, esta librería ha ganado una gran popularidad por su sencillez, portabilidad y capacidad de crear juegos 2D de manera rápida y eficaz.

A diferencia de otros motores de videojuegos más complejos, Pygame se centra en la facilidad de aprendizaje. No requiere conocimientos profundos sobre motores gráficos, lo que la convierte en una excelente herramienta educativa y en una alternativa ideal para estudiantes, docentes y programadores principiantes que desean aprender sobre animaciones, control de eventos, reproducción de sonido y manipulación de imágenes. En este documento se detallará el funcionamiento de la librería, sus principales características, funciones más usadas, ejemplos prácticos y casos reales donde su aplicación ha sido útil.

Índice

Contenido

Librería Pygame.....	2
Introducción.....	2
Índice.....	3
¿Qué hace la librería Pygame?.....	4
Casos de uso de la vida real	5
Funciones más utilizadas.....	6
Bibliografía	7

¿Qué hace la librería Pygame?

Pygame es una librería de código abierto que permite crear videojuegos 2D y programas multimedia en Python. Está construida sobre la base de SDL (Simple DirectMedia Layer), una biblioteca escrita en C que facilita la creación de gráficos, sonido y manejo de eventos en diferentes plataformas. Esto significa que los juegos desarrollados en Pygame pueden ejecutarse en Windows, macOS y Linux sin necesidad de realizar grandes cambios en el código.

Entre sus principales características se encuentran la posibilidad de manejar sprites (objetos gráficos en movimiento), detectar colisiones entre objetos, controlar la velocidad de actualización del juego (FPS), administrar múltiples sonidos o pistas musicales y realizar animaciones complejas mediante bucles lógicos controlados.

Otra de las ventajas más notables de Pygame es su capacidad para integrarse con otras librerías de Python como NumPy o OpenCV, lo que permite desarrollar simulaciones físicas, sistemas de inteligencia artificial simples, o experiencias interactivas con visión por computadora. Su naturaleza multiplataforma y su licencia libre hacen que sea ampliamente utilizada tanto en proyectos personales como educativos y académicos.

Casos de uso de la vida real

El alcance de Pygame va mucho más allá del desarrollo de videojuegos tradicionales. Gracias a su simplicidad y potencia, ha sido usada en proyectos reales de muy diversas áreas:

- Educación: utilizada en escuelas y universidades para enseñar conceptos de programación, lógica y matemáticas mediante la creación de juegos interactivos.
- Videojuegos independientes: muchos desarrolladores indie crean juegos 2D con Pygame debido a su rapidez y facilidad de despliegue.
- Simulaciones científicas: ideal para representar comportamientos físicos, trayectorias de partículas, o experimentos visuales controlados.
- Visualización de datos: permite crear gráficos y animaciones interactivas para representar resultados estadísticos o matemáticos.
- Inteligencia artificial: algunos proyectos combinan Pygame con algoritmos de IA para simular agentes autónomos, aprendizaje por refuerzo o detección de objetos.
- Interfaces gráficas personalizadas: se usa para crear aplicaciones interactivas con botones, imágenes y sonido sin necesidad de librerías complejas como Tkinter o PyQt.

Funciones más utilizadas

Pygame está compuesta por múltiples módulos que trabajan de forma conjunta. Los más usados y sus respectivas funciones son los siguientes:

Función / Módulo	Descripción detallada
<code>pygame.init()</code>	Inicializa todos los módulos internos necesarios, como audio, video y eventos.
<code>pygame.display.set_mode()</code>	Crea la ventana del juego con una resolución específica, donde se dibujan todos los elementos.
<code>pygame.display.update()</code>	Actualiza la pantalla para mostrar los cambios realizados en cada frame.
<code>pygame.event.get()</code>	Recupera todos los eventos del teclado, ratón o joystick ocurridos en un ciclo del juego.
<code>pygame.image.load()</code>	Carga imágenes en formatos como PNG o JPG para usarlas como fondos, personajes o íconos.
<code>pygame.draw.rect()</code>	Permite dibujar figuras geométricas, como rectángulos o líneas, directamente sobre la pantalla.
<code>pygame.mixer.Sound()</code>	Reproduce efectos de sonido cortos (por ejemplo, saltos, disparos o clics).
<code>pygame.mixer.music.play()</code>	Reproduce música de fondo durante el juego, soportando formatos MP3 y OGG.
<code>pygame.time.Clock()</code>	Controla la velocidad de ejecución del programa para evitar que el juego corra demasiado rápido.
<code>pygame.sprite.Sprite</code>	Clase base que representa un objeto visible en la pantalla con su propia imagen y comportamiento.

Bibliografía

- <https://realpython.com/pygame-a-primer/>
- <https://www.geeksforgeeks.org/python/pygame-tutorial/>
- [Installations pygame · PyPI](#)
- [Pygame Front Page — pygame v2.6.0 documentation](#)

Diapositiva

https://www.canva.com/design/DAG1tPMNT8w/v8RV8-vzcHk5Q1w8s8XgpA/edit?utm_content=DAG1tPMNT8w&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton