

Main Render Loop

Semestre 02, 2025

Objetivo

- Comprender qué es y cómo funciona el **Main Render Loop**.
- Saber cómo manejar **entradas de usuario** (teclado, mouse, etc.) dentro del loop.
- Aplicar buenas prácticas para mantener un **framerate constante** y un loop eficiente.

Definición

El **Main Render Loop** es el ciclo principal que ejecuta cualquier aplicación gráfica o videojuego.

Permite:

- Procesar entradas de usuario.
- Actualizar el estado de la aplicación.
- Renderizar la escena en pantalla de manera continua.

🎮 Así se logra la **ilusión de movimiento** y la **interactividad en tiempo real**.

Importancia

Sin el render loop, las aplicaciones gráficas serían **estáticas**.

Con el loop:

- Se puede responder al usuario en tiempo real.
- Se pueden crear animaciones fluidas.
- Se puede mantener actualizada la lógica de la aplicación.

Funcionamiento

En cada iteración del loop se realizan 3 pasos principales:

1. **Handle Input** – procesar entradas de teclado, mouse, etc.
2. **Update State** – actualizar posiciones, físicas, animaciones.
3. **Render Frame** – dibujar la nueva escena en pantalla.

Flujo

```
1
2  init_window(width, height)
3
4
5
6  while not window_should_close():
7
8      process_input()
9
10     update_scene()
11
12     render_frame()
13
14     swap_buffers()
15
16     limit_framerate()
17
```

Detalle

Inicialización

- Crear la ventana.
- Cargar recursos: texturas, modelos, shaders.

Handle Input

El render loop escucha eventos de entrada:

- Teclado
- Mouse
- Joysticks o controladores

Eventos:

- `KEY_DOWN` : tecla presionada.
- `KEY_UP` : tecla liberada.
- `MOUSE_MOVE` : movimiento del cursor.
- `MOUSE_BUTTON` : clics.

Update State

- Calcula nueva posición de objetos.
- Aplica físicas y lógica del juego.
- Controla animaciones basadas en el tiempo transcurrido.

Render Frame

- Dibuja la escena actualizada en el framebuffer.
- Limpia la pantalla antes de dibujar para evitar artefactos.
- Muestra el resultado en la ventana.

Swap Buffers / Sincronización

Swap Buffers:

- Cambia entre el framebuffer actual y el próximo.
- Evita parpadeos (flickering).

Limitación de FPS:

- Mantiene un framerate estable con `vsync` o retrasos manuales.

Buenas prácticas

- Limpia el framebuffer al inicio de cada frame.
- Usa `delta time` para animaciones independientes del FPS.
- No cargues archivos dentro del loop → hazlo en inicialización.
- Mantén un framerate constante para evitar “stutter”.