




Físicas de Videojuegos

Máquinas de estados



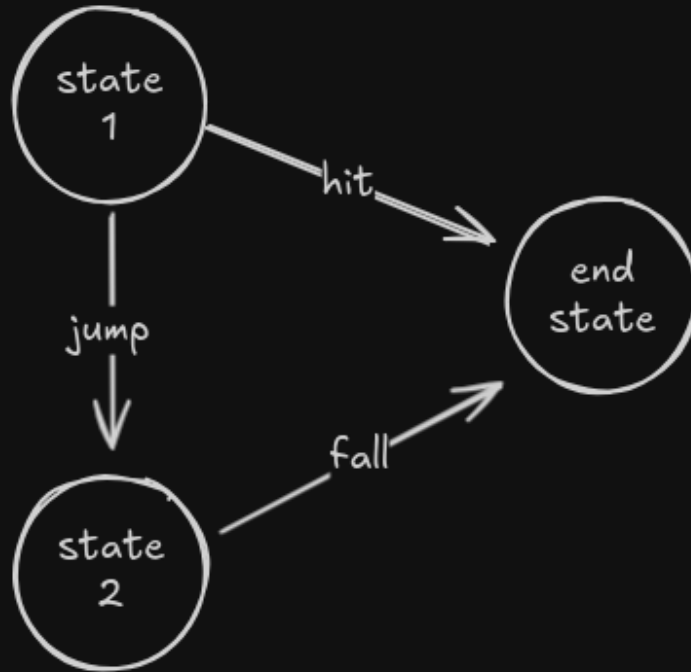
Introducción

¿De Qué vamos a hablar?

- Máquinas de estados y su utilización para la creación de videojuegos.
- Máquinas de estados
 - Definición
 - Uso
 - Aplicaciones
- Ejemplo en C con Raylib

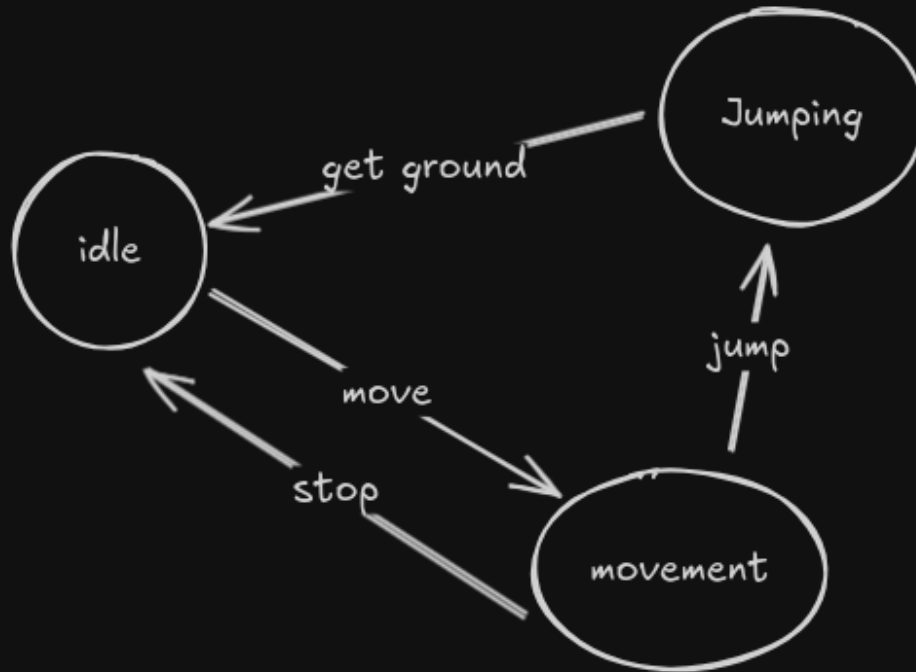
Máquinas de estados

En muchas ocasiones, necesitamos establecer que "acciones" o parámetros tener en función del estado "actual". Esto es lo que conocemos como máquina de estados.



Máquina de estados (Definición)

Una máquina de estados nos permite definir un "estado actual" por el cual en función de diferentes parámetros, podemos establecer a que estados puede viajar. Estableciendo diferentes restricciones podemos utilizarlo para definir la lógica o comportamiento de un personaje, la, Menú, etc.



Ventajas de las máquinas de estados

Los puntos clave o ventajas de las máquinas de estados pueden ser:

- Mantienen el código organizado y modular.
- Facilitan controlar el flujo del juego.

Aplicaciones

Se aplican a:

- Personajes (IA, animaciones)
- Menús y pantallas
- Niveles y escenas
- Diálogos y misiones

Construir una máquina de estados

Veamos como construir una máquina de estados en el lenguaje C por ejemplo:

Necesitaremos:

- Una serie de estados definidos.
- Un estado actual.
- Una función que los gestione.

```
1  enum states{
2      MENU,
3      GAME,
4      GAME_OVER
5  };
6
7  enum currentState = states.MENU;
8
9  void UpdateState(){
10     switch(currentState){
11         case MENU:
12             if(start_pressed)
13                 currentState = GAME;
14             break;
15         case GAME:
16             DrawGame();
17             if(lives==0)
18                 currentState=GAME_OVER;
19             break;
20         case GAME_OVER:
21             DrawGameOver();
22             break;
23     }
24 }
```

Ejemplo realizado con Raylib

Se ha creado un ejemplo con el lenguaje C y la librería Raylib.

Puedes encontrarlo en:

https://github.com/makeclassicgames/state_machine

Conclusión

1. Código más limpio
2. Fácil de ampliar o modificar
3. Ideal para IA, menús y control de flujo
4. Compatible con cualquier motor o lenguaje