




Patrón de Diseño

Entity-Component-System (ECS)



Introducción

¿De Qué vamos a hablar?

- Patrones de Diseño
- Entity-Component-System (ECS)
 - Definición
 - Aplicación en los videojuegos
 - Como implementarlo
- Ejemplo en C con Raylib de un ECS simple.

Patrones de diseño

Uno de los aspectos importantes en el desarrollo de software es la organización del código y de la posibilidad de reutilizarlo de manera clara y limpia.

Es por ello, que se utilizan los patrones de diseño, que son soluciones reutilizables a problemas comunes en el desarrollo de software.

Existen muchos patrones de diseño; aunque en el desarrollo de videojuegos, uno de los más utilizados es el patrón Entity-Component-System (ECS).

Patrón Entity-Component-System (ECS)

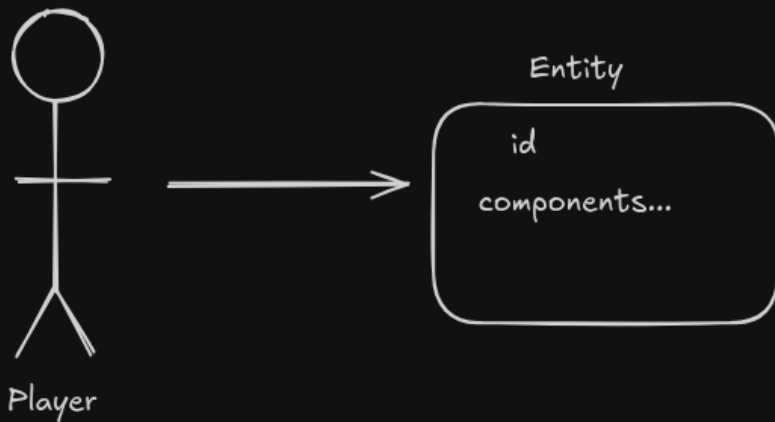
El patrón Entity-Component-System (ECS) es una arquitectura de software que se utiliza principalmente en el desarrollo de videojuegos para gestionar y organizar los datos y comportamientos de los objetos del juego.

Se basa en tres conceptos principales:

- **Entidades (Entities):** Son los objetos del juego que representan cosas como personajes, enemigos, objetos, etc. Las entidades son identificadores únicos que no contienen datos ni comportamientos por sí mismos.
- **Componentes (Components):** Son contenedores de datos que almacenan información específica sobre una entidad. Por ejemplo, un componente de posición podría almacenar las coordenadas X e Y de una entidad en el espacio del juego. Los componentes son modulares y pueden ser añadidos o eliminados de las entidades según sea necesario.
- **Sistemas (Systems):** Son responsables de procesar y actualizar los datos de los componentes. Los sistemas contienen la lógica del juego y operan sobre las entidades que tienen los componentes necesarios. Por ejemplo, un sistema de movimiento podría actualizar la posición de todas las entidades que tienen un componente de velocidad.

Entidades

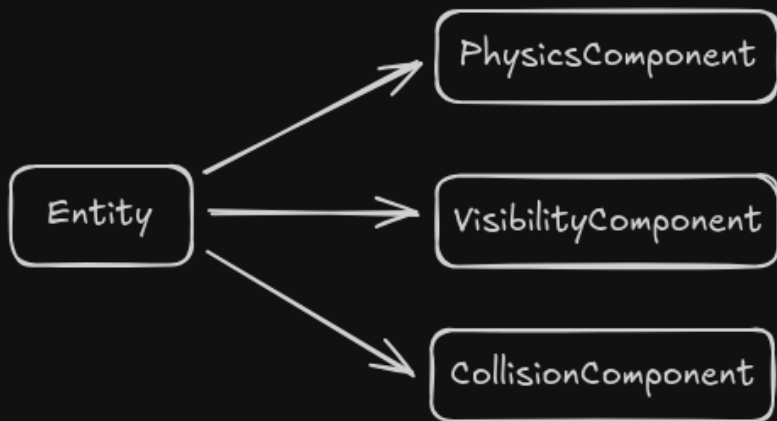
Las entidades son los objetos del juego que representan cosas como personajes, enemigos, objetos, etc. Las entidades son identificadores únicos que no contienen datos ni comportamientos por sí mismos.



Un jugador o cualquier objeto, es una entidad que tiene una serie de propiedades o componentes que definen su comportamiento y características.

Componentes

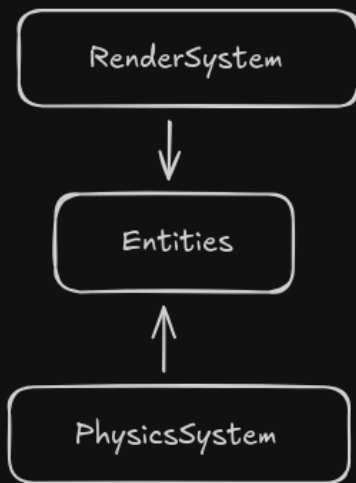
Son contenedores de datos que almacenan información específica sobre una entidad. Por ejemplo, un componente de posición podría almacenar las coordenadas X e Y de una entidad en el espacio del juego. Los componentes son modulares y pueden ser añadidos o eliminados de las entidades según sea necesario.



Como vemos en la imagen, una entidad puede tener varios componentes que definen sus características y comportamientos.

Sistemas

Son responsables de procesar y actualizar los datos de los componentes. Los sistemas contienen la lógica del juego y operan sobre las entidades que tienen los componentes necesarios. Por ejemplo, un sistema de movimiento podría actualizar la posición de todas las entidades que tienen un componente de velocidad.



Cada Sistema se encarga de una tarea específica y opera sobre las entidades que tienen los componentes necesarios para esa tarea.

Ejemplo de ECS en C con Raylib

Vamos a ver un ejemplo simple de cómo implementar un patrón ECS en C utilizando la biblioteca Raylib.

<https://github.com/makeclassicgames/ecs-raylib>

Conclusiones

1. El patrón Entity-Component-System (ECS) es una arquitectura de software que se utiliza principalmente en el desarrollo de videojuegos para gestionar y organizar los datos y comportamientos de los objetos del juego.
2. El patrón ECS se basa en tres conceptos principales: Entidades, Componentes y Sistemas.
3. El patrón ECS permite una mayor flexibilidad y modularidad en el diseño del juego, lo que facilita la adición y eliminación de características y comportamientos de los objetos del juego.