

Tema 2 - Arquitectura y modelos para entornos virtuales.

2.1 Grafos de escena y modelos jerárquicos.

Germán Arroyo, Juan Carlos Torres

5 de febrero de 2021

Tema 2: Arquitectura y modelos para entornos virtuales.

- 2.1 Grafos de escena y modelos jerárquicos.
- 2.2 Métodos básicos de representación.
- 2.3 Sistemas básicos de iluminación y cámaras.
- 2.4 Modelos de generación procesal.

2.1 Grafos de escena y modelos jerárquicos

Motor de grafos de escena:

- Trabajar con escenas.
- Edición y visualización 3D.
- Captura de interacción.
- Generación de cálculos y simulaciones.

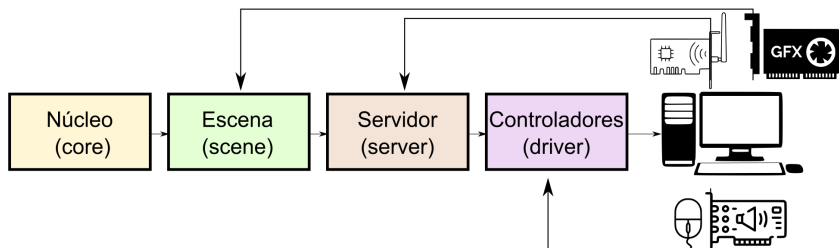


Figura 1: Esquema abstracto de motor de grafos de escena.

Escena (I)

La **Escena** contiene todos los elementos geométricos, animaciones, cálculos, etc. que serán visibles en el entorno.

Procesos de dibujo de la escena.

- Cargar recursos y transferirlos a memoria (CPU).
- Procesar/optimizar el modelo.
- Transferirlos al procesador gráfico (GPU).

Escena (II)

Una escena está definida por un grafo de escena:

- Pueden ser visualizadas o no.
- Pueden grabarse en disco y cargarse.
- Pueden ser instanciadas (una o más veces).
- Ejecutar una escena significa mostrar el mundo virtual que representa.

Recuerda.

Podemos ver la escena como un archivo ZIP que contiene una serie de elementos que conforman una porción del mundo virtual.

Grafos de escena (I)

Un **Grafo de escena** es un grafo (usualmente acíclico dirigido) que representa la estructura de una escena.

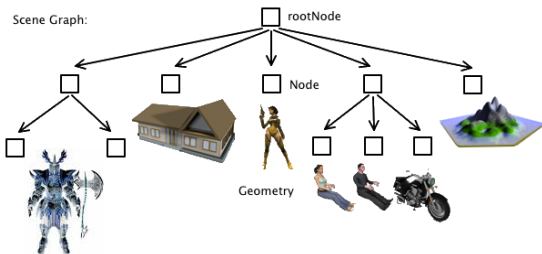
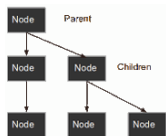


Figura 2: Representación de un grafo de escena.

Grafos de escena (II)

Los grafos de escena son útiles para:

- Entender las dependencias entre objetos.
- Diseñar el modelo geométrico.
- Editar y animar el modelo.
- Reutilizar geometría.

Nodos (I)

Un **Nodo** es la unidad básica del grafo de escena. Puede contener elementos geométricos, pero también abstractos.

- El nodo raíz representa una escena.
- Los nodos pueden contener primitivas y objetos geométricos simples (cubos, etc.), pero no está limitado a este contenido (cámaras, luces, cajas de colisión, etc.).
- Los nodos contienen propiedades (como transformaciones, color, etc.), recursos (imágenes, sonido, etc.) y funciones (código de programación).
- Una escena puede ser también un nodo (como en Godot).

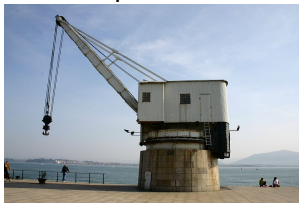
Nodos (II)

Los nodos:

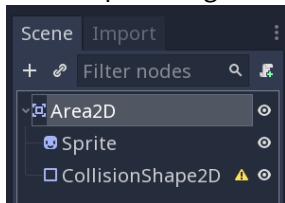
- Tienen nombre, que los identifica unívocamente.
- Tiene propiedades que heredan (de una clase).
- Puede recibir eventos mediante el uso de funciones (*callbacks*).
- Pueden extenderse (para añadir funcionalidad).
- Puede tener cualquier cantidad de nodos hijos asociados.
- Siempre tiene padre, salvo el nodo raíz que representa la escena.

Jerarquía de nodos

- Un nodo padre domina las transformaciones de un nodo hijo.



- El nodo padre algunas veces afecta al comportamiento del hijo.



Tipos de nodos (I)

Los nodos suelen estar agrupados en distintas categorías.

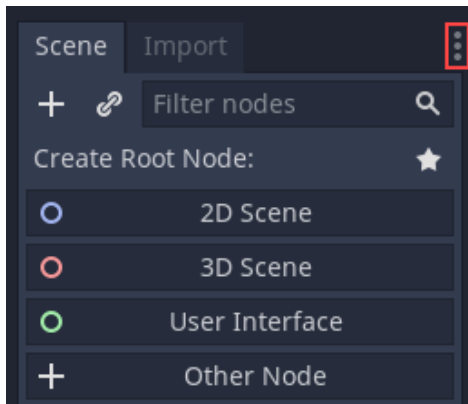


Figura 3: Tipos de nodos en Godot.

Tipos de nodos (II)

Además, cada nodo tiene una función específica en la escena que depende de la **Clase** de la que hereda.

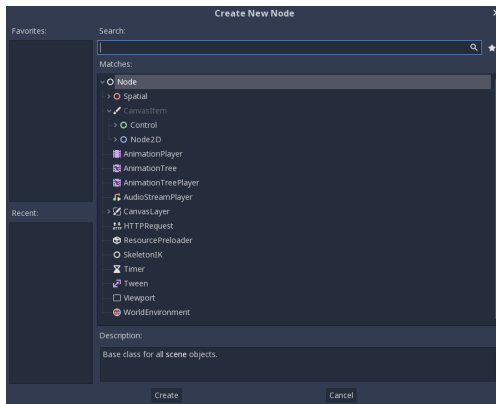


Figura 4: Algunas clases de nodos en Godot.

Propiedades de los nodos

Las propiedades de los nodos tienen tipos:

- **Sencillos:** texto, enteros, reales, etc.
- **Mapas:** texturas, imágenes, etc.
- **Referencias:** a objetos perteneciente a una clase que no llegan a ser nodos (o sí).

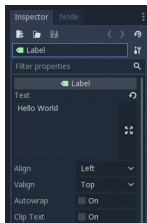


Figura 5: Propiedades de un nodo en Godot.

Instanciación de escenas (I)

Usar una escena como si fuera un nodo se llama **instanciar una escena**.

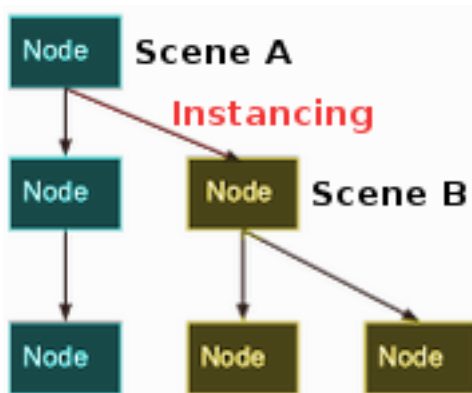


Figura 6: Instanciación de una escena.

Instanciación de escenas (II)

La instanciación de escenas permite crear objetos, comportamientos, etc. Con un comportamiento común, que luego pueden ser instanciados.

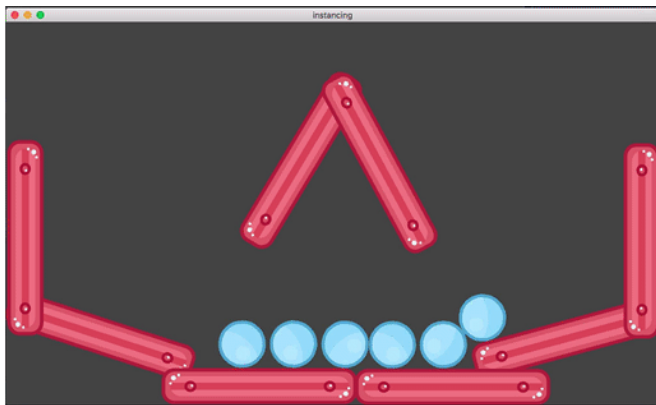


Figura 7: Ejemplo de uso de instancias.

Diseño de entornos virtuales

A la hora de diseñar entornos virtuales tenemos varias opciones:

- Modelo vista-controlador (*Model View Controller*).
- Modelo sistema de componentes de entidad (*Entity Component System*).
- **Modelo de grafo** (*Graph Model*).

Modelo vista-controlador

El patrón de diseño **Modelo vista-controlador** especifica que una aplicación consta de un modelo de datos, información de presentación e información de control.

- Cada uno de los elementos se tienen que separar en diferentes objetos.
- Fuerte relación entre la interfaz de usuario con la capa de interacción de la aplicación.

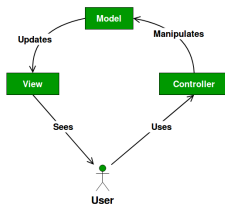


Figura 8: Esquema MVC.

Modelo sistema de componentes de entidad

- **Entidad:** la entidad es un objeto de propósito general de identidad única (ID único).
- **Componente:** los datos del objeto (desligados de la entidad) para un aspecto de la entidad (etiquetas de comportamiento).
- **Sistema:** cada sistema se ejecuta constantemente, realizando las acciones globales para cada entidad que posea un determinado tipo de componente.

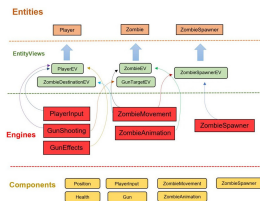
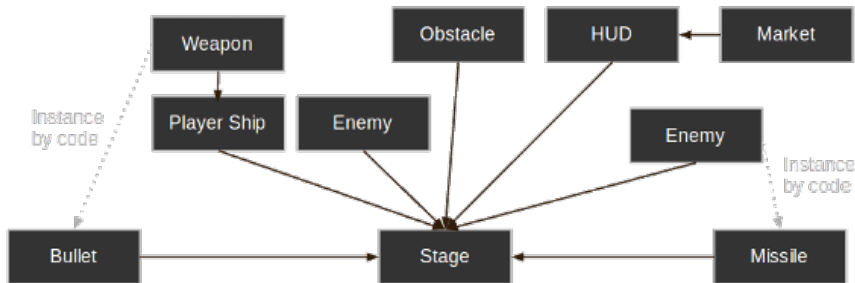


Figura 9: Esquema Entidad-Componente-Sistema.

Modelo de grafo

- Escribe las partes del mundo virtual que se visualizarán (desde un punto de vista ajeno al programador).
- Cada flecha representa pertenencia de un componente a otro.
- Las líneas punteadas indican relación mediante programación.



Diseñar usando modelos de grafo de escena

- 1 Define las escenas que habrá (habitaciones, escenarios, etc.).
- 2 Decide en que lugar empezará el usuario.
- 3 Decide muebles y objetos no interactivos.
- 4 Decide como se conectarán las escenas (que habitación conecta con otra).
- 5 Diseña los elementos interactivos y móviles.
- 6 Coloca los elementos interactivos en cada escena.

Ampliar es trivial

Si luego amplias a una casa, puedes crear una única escena que referencie tus habitaciones. Si amplias a una ciudad, tendrás una nueva escena con instancias a la casa.

Ejercicio Teórico-Práctico:

Creación del modelo de grafo para tus prácticas.

Crea el grafo para el modelo final de la última práctica, añade al menos 3 habitaciones, sus conexiones y algunos elementos animados e interactivos.