UNIVERSIDAD DE GRANADA

PRÁCTICA 3: Grafos de escena

INFORMÁTICA GRÁFICA

Joaquín Cruz Lorenzo

DNI: 79074896E

Grupo: B3

Universidad de Granada
Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación
Departamento de Lenguajes y Sistemas Informáticos

Curso 2025-2026
Octubre 2025

1. Introducción

El objetivo de esta tercera práctica de Informática Gráfica es aplicar los conceptos fundamentales de **modelos jerárquicos** y **grafos de escena**. Un grafo de escena es una estructura de datos clave en gráficos 3D que permite organizar los objetos de forma jerárquica. Esto posibilita la creación de modelos articulados complejos, donde la transformación (posición, rotación) de un nodo hijo es dependiente de la de su nodo padre.

Para cumplir con los objetivos, he implementado un modelo articulado de un **móvil de cuna**. Este modelo se ha construido usando una combinación de primitivas CSG para las estructuras y modelos '.glb' importados para las figuras. El modelo implementa múltiples grados de libertad con varias capas de dependencia.

2. Desarrollo de la práctica

El desarrollo se ha centrado en dos fases: la construcción del grafo de escena en Godot (Actividades 1 y 2) y la implementación de la animación y la interacción mediante un script genérico (Actividades 3 y 4).

2.1. Actividad 1 y 2: Diseño del Grafo de Escena

El modelo del móvil de cuna se ha diseñado con **seis grados de libertad** (articulaciones) y múltiples niveles de dependencia, superando los tres grados de libertad y dos dependencias requeridos.

La estructura del grafo de escena implementada en el archivo `escena_principal.tscn` es la siguiente:

```
Raiz (Raíz de la práctica)
|
+-- Ejes (script de ejes.gd)
|
+-- Camera3D (cámara de la práctica)
|
+-- DirectionalLight3D (Luz par la escena principal)
|
+-- GrafoP3 (Nodo principal de la P3)
    |
    +-- Soporte_Mesh, Soporte_2_Mesh, Cable_Conecta_Pivote (Objetos ESTÁTICOS)
    |
    +-- Pivote_Principal (Primer nodo con rotación: Gira con la tecla 1)
        |
        +-- Varilla_1_Mesh, Varilla_2_Mesh, Varilla_3_Mesh, Varilla_5_Mesh (Visuales)
        |
        +-- Figura_A_Node (Segundo nodo con rotación: Gira con la tecla 3)
            |
            +-- Figura_A_Mesh (Instancia de Planet.glb)
```

```
|
+-- Pivote_Secundario (Tercer nodo con rotación: Gira con la tecla 2)
    |
    +-- Varilla_4_Mesh (Visual)
|
+-- Pivote_Terciario (Cuarto nodo: Gira solo, sin tecla)
    |
    +-- Varilla_6_Mesh, Varilla_7_Mesh, Varilla_8_Mesh (Visuales)
    |
    +-- Figura_B_Node (Quinto nodo: Gira con la tecla 4)
        |
        +-- Figura_B_Mesh (Instancia de SpaceShip.glb)
    |
    +-- Figura_C_Node (Sexto nodo: Gira con la tecla 4)
        |
        +-- Figura_C_Mesh (Instancia de Planet2.glb)
```

Esta jerarquía asegura que las dependencias funcionen correctamente:

- `Pivote_Secundario` y `Figura_A_Node` dependen de `Pivote_Principal`.
- `Pivote_Terciario` depende de `Pivote_Secundario` (y por tanto, también de `Pivote_Principal`).
- Las figuras B y C dependen de `Pivote_Terciario` (y por tanto, de todos sus ancestros).

2.2. Actividad 3 y 4: Animación e Interacción

Se ha utilizado un único script genérico, `rotador_simple.gd`, para controlar todas las articulaciones. Este script se instancia en cada uno de los 6 nodos pivote.

2.2.1. Código del script (rotador_simple.gd)

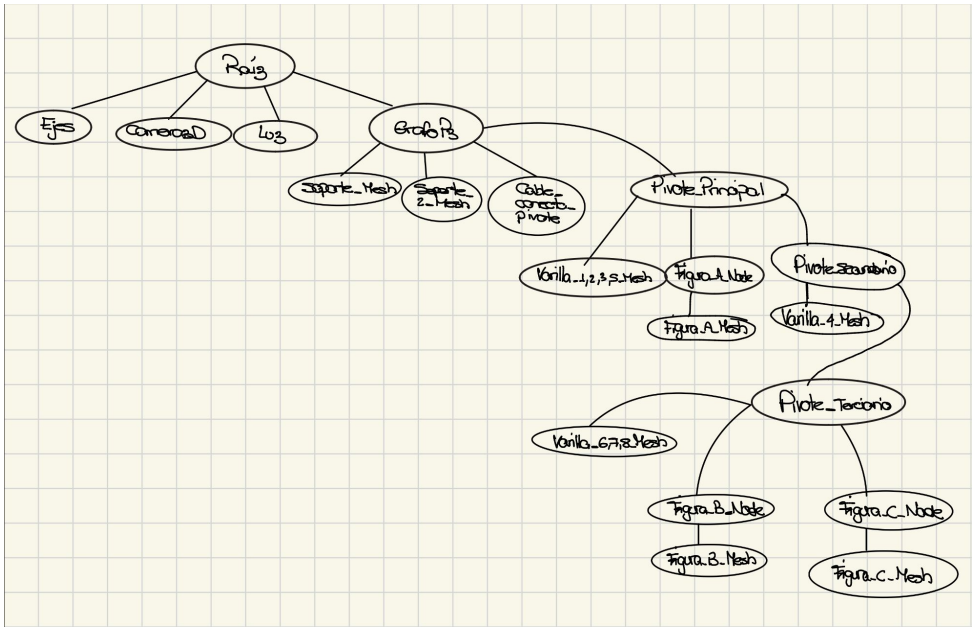
El script implementa la lógica de las Actividades 3 y 4. Utiliza variables `@export` para configurar la velocidad de rotación y la acción de teclado correspondiente desde el editor de Godot, tal como sugiere el guión.

Listing 1: Contenido de rotator_simple.gd

```
1 extends Node3D
2
3 # Definimos una variable "exportable"
4 # Esto significa que puedo cambiar la velocidad
5 # desde el editor para cada pivote.
6 @export var rotation_speed_deg := 20.0 # grados por segundo
7
8 # Esta variable nos permite decir que accion (tecla) usa este
     nodo
9 @export var activar_accion := "toogle_principal"
```

```
10
11  # Esto guarda si la animacion esta encendida o apagada
12  var activa := true
13
14  func _process(delta):
15      # 1. Comprueba si se ha pulsado la tecla que le hemos
            asignado
16      if Input.is_action_just_pressed(activar_accion):
17          activa = !activa # Invierte el valor (true -> false |
                false -> true)
18
19      # 2. Solo rotamos si la variable 'activa' es verdadera
20      if activa:
21          rotation.y += deg_to_rad(rotation_speed_deg * delta)
```

2.3. Nodo de la práctica a mano



2.3.1. Configuración de la Interacción

En el archivo `project.godot`, se definieron las acciones de entrada para las teclas 1, 2, 3 y 4.

- `toogle_principal`: Asignada a la tecla **1**.
- `toogle_secundario`: Asignada a la tecla **2**.
- `toogle_figura`: Asignada a la tecla **3**.
- `toogle_figuras_juntas`: Asignada a la tecla **4**.

En el editor de escenas, se asignó a cada nodo pivote su `activar_action` correspondiente (ej. `Pivote_Secundario` usa `"toogle_secundario"`). Esto permite un control individualizado de cada articulación.