

**Ciclo:** Animaciones 3D, Juegos y Entornos Interactivos

**Curso:** 2023/24

**Módulo:** Desarrollo de Entornos Interactivos Multidispositivo

**Turno de Mañana**

## PRÁCTICA 01 JUEGO DE LABERINTO

Vamos a crear un juego basado en físicas y colisiones, inspirado en el clásico juego del laberinto:



### 1. Niveles y escenarios

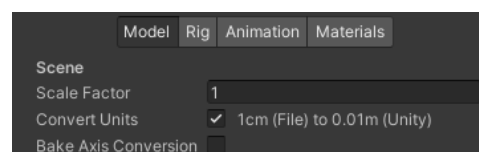
Vamos a crear 3 niveles de juego, con sus correspondientes escenas. Para ello, se ha preparado una serie de archivos .fbx que servirán de terreno de juego. Descárgalos de la carpeta compartida de OneDrive e impórtalos en tu proyecto.

Verás 3 archivos:

1. Un fbx con la malla del nivel 1
2. Un fbx con el colisionador de baja poligonación para el "mesh collider" del nivel 1
3. Un fbx que contendrá tanto la malla de alta poligonación como la de baja poligonación

NOTA: verás que los archivos .fbx son en realidad contenedores que pueden incluir maya, materiales, animaciones, etc. Funcionan como prefabs que puedes incluso desempaquetar en la ventana de jerarquía (RMB > prefab > unpack)

NOTA2: hay que tener cuidado con las medidas. Maya funciona en cmts., algo que puedes desactivar en la pestaña de "Model". También puedes ajustar el factor de escala.





Ajusta la configuración de la importación para tener unos terrenos sobre los que harás rodar la bola de metal. Las dimensiones las decides tú, pero evita terrenos muy pequeños o grandes, y ten en cuenta que todo deberá ajustarse para que sea realista: masas, resistencias, empujes, etc.

Crea una bola que pueda caber por el agujero del terreno. Te interesa crear un prefab con la bola para poder reutilizarla, con sus materiales, componentes, medidas, peso, etc (aunque algunos scripts dependerán de la instancia, no del prefab).

Crea materiales de texturas y materiales físicos para dotar al terreno y a la bola de realismo.

### **Estos son lo 3 niveles del juego:**

#### Nivel 1:

Usando el terreno 1 que será estático, deberemos empujar la bola hasta que caiga por el agujero.

#### Nivel 2

Usando el terreno 2, deberemos añadir torsiones al terreno para que la bola resbale por él por su propio peso hasta que caiga por el agujero.

#### Nivel 3

Igual que el nivel 2, pero añadiremos paredes que tendremos que ir usando para esquivar o ayudar a la bola hasta llegar al final.

RECOMENDACIÓN: los obstáculos deben ser hijos del terreno para que se muevan con él, y tener sus propios colisionadores. Si no son hijos, se comportarán como objetos intependientes.

### **Mecánicas:**

Siempre que la bola se caiga del terreno, volveremos a empezar ese nivel.

Cuando llegue al agujero final, cargaremos el nivel siguiente (salvo que sea el último). Para ello tendremos que agregar la librería [SceneManager](#) y usar el método [LoadScene](#).



## 2. Controles y físicas

Vamos a usar los 2 joysticks del gamepad para mover la bola o el terreno: el izquierdo para mover la bola en horizontal o girar el terreno en su eje Z, y el derecho para mover la bola en profundidad o girar el terreno en su eje X (opcionalmente, puedes usar las teclas que quieras)

El movimiento de la cámara ajústalo según creas que sea necesario. Es posible que tengas que ajustar el valor "Clipping Planes" que indica a qué distancia de los objetos la cámara deja de verlos, ya que los objetos pueden estar muy cerca.

Para que el juego sea realista mediante físicas, deberás ajustar los siguientes elementos hasta lograr el efecto deseado:

- Rigidbody aplicado al objeto al que vayas a aplicar fuerzas o torsiones (si es necesario, restringiendo movimientos)
- Masas relativas de los objetos. Intenta ser realista, recuerda que una masa de 1 equivale a 1KG (si la bola pesa mucho, empujará el terreno hacia abajo)
- Materiales físicos. Es recomendable crear "[physics materials](#)" para la bola y el terreno y así ajustar su elasticidad y fricción
- Fuerza aplicada con los controles. Opcionalmente, puedes ajustar la sensibilidad de los ejes en el Input Manager de Unity.
- Resistencia al aire de los objetos, tanto lineal como angular.

IMPORTANTE: como vamos a aplicar fuerzas, es mejor cambiar el método Update por el [FixedUpdate](#)

## 3. Colisiones

Para este juego es necesario crear colisionadores de dos tipos:

[Mesh Colliders](#) y [Sphere/Box Colliders](#)



Los terrenos en FBX incluyen una malla adicional de baja poligonación que te permitirá añadirla como colisionador de malla ("[mesh collider](#)"). Si es necesario, "desempaqueta" el FBX en el escenario para separarla de la malla del terreno.

La bola del juego y las paredes que añadas llevan incluidas sus propias mallas de colisión ("Sphere Collider / Box Collider")

NOTA: el mesh collider, para que reconozca el agujero, NO puede ser de tipo Convex, y por tanto no puede combinarse con un Rigidbody que no sea Kinemático. Para lograrlo, tendrás que mover un padre con Rigid Body y aplicar el collider a un hijo.

### Colisionadores de tipo disparador

Puedes usar objetos vacíos con colisionador con el trigger activado para detectar cuándo el usuario ha llegado al final.

---

### Entrega y evaluación

El juego se desarrollará en clase y la fecha de entrega será el **viernes 20 de octubre**.

Para dar por finalizada la práctica, **se subirá a OneDrive del alumnos un archivo comprimido en ZIP con el proyecto de Unity SIN LA CARPETA LIBRARY, y sin los archivos que no se usen. El archivo tendrá el siguiente nombre:**

**ApellidoNombre\_PR01.ZIP**

Obtén el enlace compartido del archivo y añádelo a la entrega.

Este ejercicio **será calificado con nota**, siguiendo los siguientes criterios:

1. Creación de escenas y terrenos
2. Controles del usuario
3. Ajustes de los parámetros para crear físicas
4. Colisionadores