



Ciclo: Animaciones 3D, Juegos y Entornos Interactivos

Curso: 2021/22

Módulo: Desarrollo de Entornos Interactivos Multidispositivo – Mañana **16/11/2021**

EXAMEN PRÁCTICO – 1ª EV

Antes de comenzar, descarga los materiales y lee atentamente estas instrucciones y visiona el vídeo de muestra, disponible en la plataforma. A continuación, realiza las siguientes tareas:

1.- GitHub

Crea un repositorio PRIVADO en tu cuenta de Git Hub con el nombre:

ApellidoNombreEx1EV

Deberás autorizarme a acceder a él con permisos de Administrador:

Settings > Manage Access > Add People > algo

A partir de ese momento, trabaja con el clon de **TU** repositorio. Realiza commits cada vez que des por cerrada una tarea (aunque no esté concluida del todo).

La entrega será el enlace a este repositorio.

2.- Creación y desarrollo del proyecto Unity

Crea en el repositorio clonado un Proyecto Unity que usarás para crear el videojuego del examen.

Llama al proyecto de Unity con tu apellido y nombre (no te olvides de copiar el archivo “gitignore” dentro de la carpeta del proyecto antes de realizar el primer *commit*).

Crea una escena con los elementos y los scripts necesarios para lograr un videojuego similar al del vídeo. Deberás añadir estos elementos:

1. Un plano que servirá de suelo de madera (el del vídeo está escalado X:0,4 – Z:2)
2. Una pared trasera y un segundo suelo que evita que la bola se caiga por los bordes



3. Una bola (la del vídeo de muestra está escalada 0,4 en todos los ejes)
4. Bolos (deberán ser instanciados, por lo que no deben estar inicialmente en la escena)
5. Un puntero que determinará hacia dónde lanzamos la bola. Puede ser una caja pequeña
6. Los menús y los contadores de la UI

CONSEJO: asegúrate de que los elementos se ubican correctamente respecto a los ejes de la escena y trata de mantener unas escalas adecuadas. Y GUARDA siempre la escena.

3.- Cuestiones a resolver en el juego

Tu proyecto deberá resolver los siguientes retos:

A) Interactividad y control del tiempo

- a. El puntero que determina la dirección del lanzamiento se mueve en la horizontal con el joystick izquierdo o con las flechas del teclado
- b. El movimiento está restringido a los laterales para no poder mandar la bola muy fuera de la pista.
- c. Una cuenta atrás desde 4 indicará el momento en el que dejaremos de poder mover el puntero y haremos que desaparezca. Este contador se reiniciará con cada lanzamiento

B) Fuerzas y físicas

- a. Se aplicará un empuje a la bola al pulsar el botón "A" del mando o la barra espaciadora del teclado. La dirección del empuje lo determina la posición del puntero.
- b. Ese empuje es único, y no puede repetirse una vez está la bola en movimiento.
- c. Tampoco es recomendable que se pueda lanzar la bola cuando el juego se ha terminado
- d. Los bolos deben responder a la gravedad y a las colisiones. Para ello, deberás usar la malla de baja poligonación que se incluye dentro del FBX.



RECUERDA: si usas un mesh collider junto a un Rigid Body, el collider tiene que ser de tipo Convex.

C) Movimiento de la bola y seguimiento con la cámara

- La cámara seguirá a la bola en su movimiento por la pista, pero solo en su posición en el eje Z (en profundidad), desde una posición elevada y mirándola en todo momento (debe conseguirse con código)
- Unos segundos después de lanzar, la bola volverá a su posición inicial y la cámara la seguirá (es preferible un seguimiento con cierto retardo)

NOTA: Es posible que tengas que eliminar las fuerzas de la bola tras el lanzamiento, para ello puedes usar este código (donde "rb" es el componente RigidBody de la bola):

```
rb.velocity = Vector3.zero;
```

```
rb.angularVelocity = Vector3.zero;
```

D) Creación de elementos (bolos) mediante instanciación y prefabs

- En el archivo con los materiales encontrarás un FBX con el bolo. Incluye una malla de alta poligonación y otra de baja que usarás para el colisionador (deberás "desempaquetar" el FBX para usar solo una)
- Al comienzo del juego se instanciarán 9 bolos al final de la pista, ordenados como aparece en el vídeo
- Para hacerlo se deberán usar bucles y preferiblemente un "empty object" que sirva de referencia

E) Colisiones:

- Tanto la bola como los bolos deben estar sometidos a la gravedad y colisionar entre ellos, como en el vídeo de muestra
- Los bolos deben tener un "mesh Collider" pero deberán usar la malla de baja poligonación que se incluye en el fbx

F) UI: deberás crear los menús y los textos que aparecen en el juego:

- Edita la hoja de sprites de los materiales para poder crear botones cuya apariencia cambie según el estado del botón (reposo, seleccionado, hover y pulsado).
RECUERDA: tienes que instalar el paquete "2D Sprite" para poder editarlo.



- b. Añade navegación para poder pasar de unas escenas a otras: inicio, instrucciones, juego (tendrás que crear unos botones al final de la partida para volver al inicio o volver intentarlo)
- c. Añade contadores:
 - i. Bolos derribados (deben contarse solo una vez)
 - ii. Intentos realizados (se ponen a cero al comenzar la partida)
 - iii. Segundos que pasan en la preparación del lanzamiento. Al derribar todos los bolos este texto cambiar por un aviso de victoria.
- d. Usa la tipografía que encontrarás en los materiales para los contadores (para los botones puedes usar la Arial)

G) Materiales y texturas

- a. Crea los materiales para dotar de realismo a la escena
- b. Crea materiales físicos para que las físicas se comporten adecuadamente

Recuerda hacer *commits* de forma regular para seguir el proceso de desarrollo, y añade descripciones y comentarios al código para facilitar su lectura.

Entrega y evaluación

El tiempo para realizar el examen es de 4 horas.

Cuando se termine el plazo para realizar el examen, copia la dirección de tu repositorio y pégala en el texto de entrega el examen.

Criterios de calificación del examen práctico

- 1. Git Hub (**1 punto**)
 - a. Realizar correctamente el repositorio y las acciones necesarias



- b. Hacer *commits* de forma correcta y cada vez que se produce un cambio significativo en el proyecto
- 2. Organización del proyecto y de la escena y coding correcto (**1,5 puntos**)
 - a. Los *assets* son organizados y nombrados de forma correcta en el proyecto y en la escena
 - b. El control de los elementos de la escena es correcto
 - c. La organización del código es adecuada
 - d. Se usa el código de forma adecuada y eficiente
- 3. Interactividad, movimiento y control de tiempo (**2,5 puntos**)
 - a. El puntero se mueve según las indicaciones y con las restricciones indicadas
 - b. Se realiza correctamente una cuenta atrás que desactiva el movimiento y que se reinicia de nuevo en cada lanzamiento
- 4. Instanciar elementos (**1 punto**)
 - a. Uso correcto de los *prefabs* y su instanciación
 - b. Creación de ubicaciones correctamente mediante bucles
- 5. Físicas y colisiones (**2,5 puntos**)
 - a. Se crean las físicas adecuadas al juego
 - b. Se añaden los componentes necesarios para detectar colisiones
 - c. Se realizan las tareas necesarias tras producirse una colisión
- 6. UI – User Interface (**2,5 puntos**)
 - a. Se han editado la hoja de sprites adecuadamente
 - b. Se han creado los menús con la navegabilidad exigida
 - c. Se crean contadores y botones en la escena, interactuando con el juego y con el usuario
- 7. Materiales (**1 punto**)
 - a. Se añaden materiales con texturas y materiales físicos

La nota final del examen será un nº con decimales entre 0 y 10 puntos