Ciclo: Animaciones 3D, Juegos y Entornos Interactivos **Curso:** 2020/21

Módulo: Desarrollo de Entornos Interactivos Multidispositivo

Nombre y apellidos:

EXAMEN TEÓRICO **Grupo A**

Escribe tu nombre y apellidos en la cabecera de este documento, y a continuación explica qué herramientas vistas a lo largo del curso utilizarías para lograr los objetivos planteados en el ejercicio práctico, explicando por qué esas y no otras:

Interactividad y restricción de movimiento

Para crear la interactividad, primero debemos crear el movimiento de la esfera. Para esto serializamos un float llamado velocidad que nos sirva para variar la velocidad de la misma.

A continuación, necesitamos crear un método que nos permita mover la nave. Necesitaremos crear dos variables para que estas cambien de valor conforme uses los joysticks (Input.GetAxis), y le asignas el eje horizontal y el vertical a cada valor.

Por otro lado, creas dos variables de posición x y z respectivamente, igualándolas a la propiedad transform.position.x/z, de manera que sirvan para definir la posición de la esfera como variables.

Lo siguiente es utilizar condicionales de if, para definir las capacidades del movimiento de la esfera, restringiéndolo al interior del plano. Se pueden utilizar || entre las condiciones para que el programa interprete "o" y el código se optimice.

Osea posición menor que x && mayor que x ||...

En el interior usamos un transform.Translate(Vector3.right...) el cual define la dirección del eje de movimiento dentro de la configuración predeterminada de unity, así como lo multiplicamos por velocidad, Time.deltaTime para asegurarse de que se ejecuta uniformemente sin importar el ordenador, todo ello por el valor de la variable de desplazamiento del joystick.

Con esto se mueve en uno de los ejes junto con la restricción. Se repite para el otro.

Para que se ejecute, se llama al método en update, para que se calcule la posición cada frame.

Seguimiento del jugador con la cámara

Para situar la cámara sobre la esfera necesitamos una variable de la esfera como GameOb ject, creamos un vector3 que represente la posición relativa a tiempo real.

En start igualamos el vector de posición relativa al vector transform.position (del objeto de la cámara, donde va el script).

En el void update necesitamos igualar ese vector transform.position de la cámara al de la esfera + la posición relativa.

Es decir, que en el primer frame se define la distancia entre objetos en unity, y en el upda te se actualiza la posición de la cámara cada frame en función de la posición de la esfera.

Creación de elementos (columnas) de forma aleatoria.

Se utiliza una corrutina de la misma manera que está explicado en la interfaz. En este caso utilizas math random para randomizar los valores, con dos variables random diferentes para poder instanciar la posición de las columnas en zonas aleatorias dentro del rango de los ejes x,z del plano. Importante Quaternion.identity para no preocuparse de la rotación de las instancias. Todo en un loop infinito.

User Interface (tanteos)

Añades la librería correspondiente y utilizas corrutinas para crear la función asíncrona que se realice constantemente. Primero generas una variable serializada de text (con la nueva librería), en start inicias la corrutina. Tienes que crear la corrutina con Ienumerator, y crear un bucle infinito en ella con for, de manera que reutilizas la variable n que aumenta en 1 cada loop, para modificar el valor del string de texto que ordenas que la corrutina te devuelva. Y bueno, todo esto relacionando las cosas en unity (en la variable MyText necesitas insertar un texto... ese tipo de cosas.) Con las columnas similar pero desde la condición de los loops de creación de columnas.

Colisiones:

Para crear las colisiones necesitamos dar la propiedad de rigidbody a la esfera y a las columnas. De esta manera, al crear un script usando void OnTriggerEnter(...), lo que conseguimos es que, al poner un if condicionante, si un elemento con nombre esfera toca las instancias de la columna(other.gameObject.name), estas se destruyen con Destroy (gameObject). Lo que

sucede es que están chocando hitboxes, y cuando se detecta este choque en las columnas estas se borran.

En casi todos es necesario relacionar parámetros de unos elementos con otros.

Entrega y evaluación

Cuando tengas completo el documento, expórtalo a pdf con este formato:

Apellidos_nombre_ExTco1EV.pdf

Guárdalo en el misma carpeta que el documento Word, dentro del repositorio, y súbelo a un commit de GitHub.