

Ejercicios Práctica 2:

1. Crear una escena simple sobre la que probar diferentes configuraciones de objetos físicos en Unity. La escena debe tener un plano a modo de suelo, una esfera y un cubo.
 - a. Ninguno de los objetos será físico.
 - b. La esfera tiene físicas, el cubo no.
 - c. La esfera y el cubo tienen físicas.
 - d. La esfera y el cubo son físicos y la esfera tiene 10 veces la masa del cubo.
 - e. La esfera tiene físicas y el cubo es de tipo `IsTrigger`.
 - f. La esfera tiene físicas, el cubo es de tipo `IsTrigger` y tiene físicas.
 - g. La esfera y el cubo son físicos y la esfera tiene 10 veces la masa del cubo, se impide la rotación del cubo sobre el plano XZ.

En el repositorio de las prácticas crear un documento Markdown en el que expliques los resultados obtenidos.

2. Sobre la escena que has trabajado ubica un cubo que represente un personaje que vas a mover. Se debe implementar un script que haga de `CharacterController`. Cuando el jugador pulse las teclas de flecha (o `aswd`) el jugador se moverá en la dirección que estos ejes indican.
 - a. Crear un script para el personaje que lo desplace por la pantalla, sin aplicar simulación física.
 - b. Agregar un campo público que permita graduar la velocidad del movimiento desde el inspector de objetos.
 - c. Estar a la escucha de si el usuario ha utilizado los ejes virtuales. Elegir cuáles se va a permitir utilizar: flechas, `aswd`.

Ayuda: Utilizar la clase `Input`.

- El recorrido virtual realizado con los controladores (teclas) debe ser proporcional a lo que se desplaza el jugador:
 - Si sólo pulsa una vez, corresponderá a una unidad, Unity asigna +1 o -1 según la dirección del movimiento
 - Si se mantiene pulsado, el jugador debe avanzar en un movimiento continuo, así que Unity asigna un valor entre 0 y 1 ó 0 y -1
 - Una vez que tenemos la proporción del desplazamiento, este también debe ser proporcional a la velocidad que hemos establecido para el objeto. El objeto debe cumplir los siguientes requisitos en el movimiento:
 - Se debe mover exclusivamente en el plano XZ de su sistema de referencia (su suelo)
 - El avance se debe producir hacia adelante.

Recuerda: los parámetros que debas usar que sean vectores tienen que ser del tipo Vector3, por tanto, debes crear una referencia a un objeto de tipo Vector3.

Recuerda: Vector3 dispone de la propiedad forward.

Ayuda: la traslación la puedes calcular usando el sistema de referencia del objeto.

Ayuda: Debes utilizar la función GetComponent

- Elegir otros ejes virtuales para el giro y girar al jugador sobre el eje OY (up).
-
3. Sobre la escena que has trabajado programa los scripts necesarios para las siguientes acciones:
 - Se deben incluir varios cilindros sobre la escena. Cada vez que el objeto jugador colisione con alguno de ellos, deben aumentar su tamaño y el jugador aumentar puntuación.
 - Agregar cilindros de tipo A, en los que además, si el jugador pulsa la barra espaciadora lo mueve hacia fuera de él.
 - Se deben incluir cilindros que se alejen del jugador cuando esté próximo.
 - Ubicar un tercer objeto que sea capaz de detectar colisiones y que se mueva con las teclas: I, L, J, M
 - Debes ubicar cubos que aumentan de tamaño cuando se le acerca una esfera y que disminuye cuando se le acerca el jugador.