Guía de ejercicios

Día 1

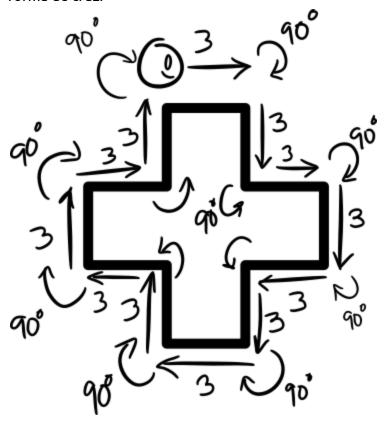
- 1. Crear un nivel con los assets de la carpeta Prefabs/Platformer
- Crear un nuevo prefab llamado Persuiter a partir de un Enemy que, cada vez que el jugador intersecte un detector especial, el persuiter aumente su velocidad. Para lograrlo, arreglar el script Scripts/Persuiter.cs, y añadir un Prefabs/Technical Stuff xD/detector como hijo del transform del enemigo.
- 3. Crear un nuevo prefab llamado *Follower* a partir de un *Enemy* que, cada vez que el jugador entre a un collider esférico que funciona como trigger, persiga al jugador. Cuando el jugador salga del collider, el enemigo debe volver a su comportamiento normal.
 - a. Para hacerlo, crear un game object vacío como hijo del enemigo llamado *player* detector.
 - b. Poner al *player detector* en una capa física que sólamente colisione con el jugador.
 - c. Añadirle al player detector un sphere collider que funcione como trigger, y configurar la esfera... no debe ser muy pequeña, pero tampoco muy grande. Configúralo y pruébalo hasta que sea divertido! :) Recuerda que el jugador ya tiene un Rigidbody! Así que no es necesario que player detector tenga un rigidbody también.
 - d. Añadirle al *player detector* el script *Follower*.
 - e. Arreglar Scripts/Follower.cs
- 4. Crear un script: *Scripts/ProximityExpirable.cs* que haga lo mismo que *Scripts/Expirable.cs*, pero que ejecute la corrutina *EventuallyDie()* sólamente cuando el jugador entre a un sphere collider que funciona como trigger.

Día 2

- 1. Cuando el jugador muera, hacer que los enemigos se detengan durante una cantidad de tiempo, y luego vuelvan a la normalidad.
- Practicando!! Modificar el prefab Prefabs/Platformer/enemy spawner para que isDestroyed se vuelva true cuando el jugador salte sobre él, y que luego de cierta cantidad de segundos determinada por una variable public float regenerateCooldown; isDestroyed vuelva a ser false.
- 3. Modificar el prefab *Prefabs/Cannon* para que dispare balas cada cierto tiempo. Las balas deben ser destruidas luego de transcurrido cierto tiempo. Para ello, crear un script *Scripts/Cannon.cs.* El cañón tiene un script con un método *Disparar()* que corre una animación de disparo, éste método debe ser invocado por el script

Scripts/Cannon.cs cuando el cañón dispare la bala. La bala debe ser un prefab... debes crear el prefab de la bala y el script que haga que la bala vaya hacia adelante, además, la bala debe matar al jugador al impactar con éste.

4. Crear una corrutina recursiva que haga que un game object cualquiera se mueva en forma de cruz:



Día 3

Para este día, puedes consultar los scripts del juego en la carpeta *persistence-in-videogames*. (https://github.com/VengadoraVG/persistence-in-videogames)

- 1. **Practicando!!** Crear un nivel con los prefabs de la carpeta *Prefabs*
- 2. Crear una escena llamada *Scenes/Selection* de selección de personaje para la escena *Scenes/Persistence*
 - a. El jugador debe ser capaz de elegir entre un cubo, una esfera y un cilindro.
- 3. Para la escena *Scenes/Persistence*, hacer que los enemigos recuerden su posición luego de cerrar el juego.
- 4. Crear un prefab llamado *Prefabs/powerup* que, cuando el jugador se acerque a él, incremente el score del juego.
- 5. Hacer que el juego recuerde el score del jugador luego de cerrar el juego.

Día 4

- 1. Practicando!! Crear un prefab llamado Prefab/bomb a partir del Prefab/follower creado en el día 1, ejercicio 3, que cuando el enemigo comience a seguir al jugador, haga que el enemigo se destruya eventualmente (para ello se puede usar el script Scripts/ProximityExpirable.cs creado en el día 1, en el ejercicio 4), activando un sphere collider de "explosión" que cause que el jugador muera si se encuentra dentro de ésta área.
- 2. Crear la escena *Scenes/Store* en la que el jugador sea capaz de comprar 50pts de score.