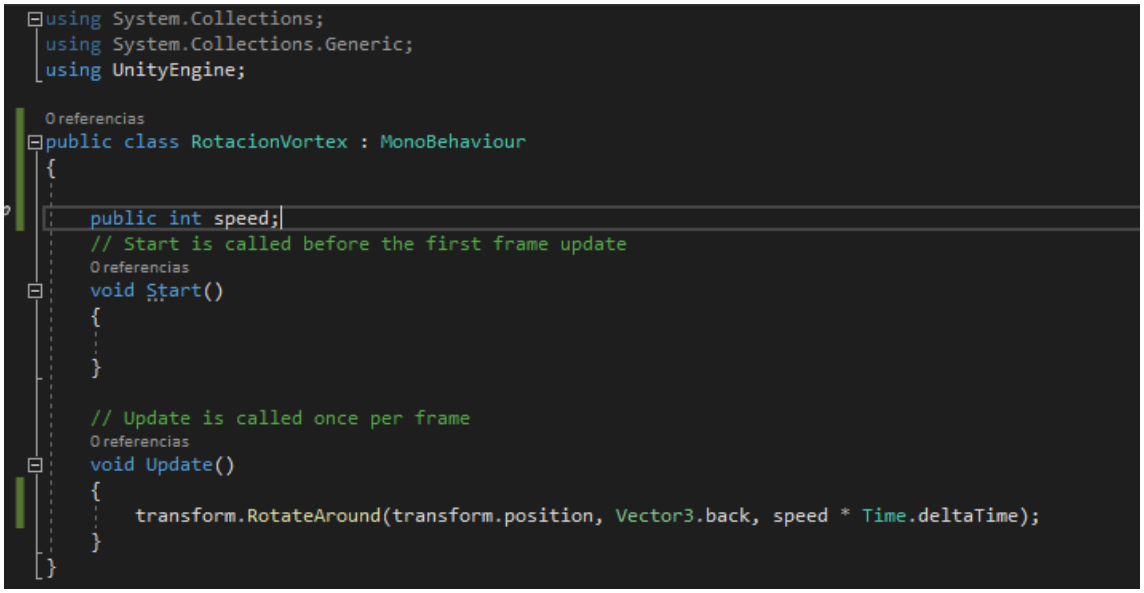
1. Implementar una animación sobre el Vòrtex haciéndolo rotar sobre el eje Z. (Se puede hacer en un Script).

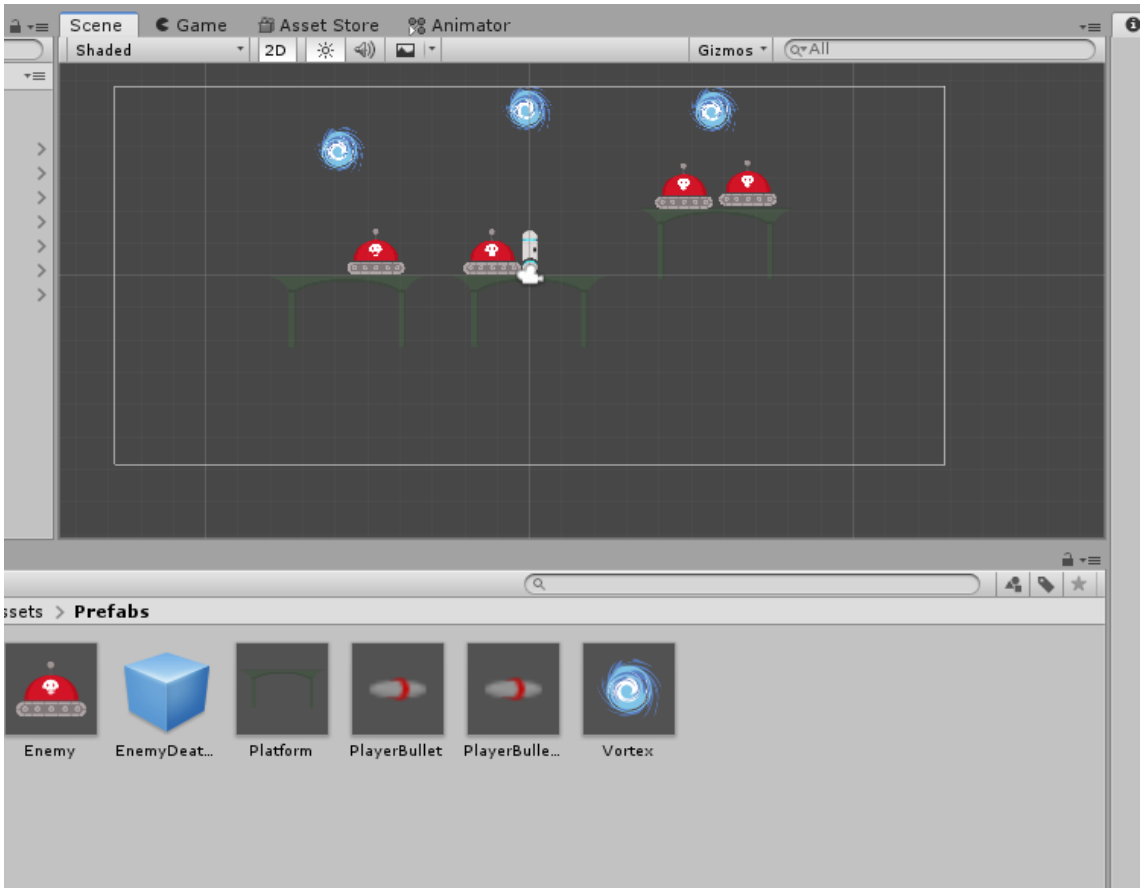


```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

0 referencias
public class RotacionVortex : MonoBehaviour
{

    public int speed;
    // Start is called before the first frame update
    0 referencias
    void Start()
    {

    }

    // Update is called once per frame
    0 referencias
    void Update()
    {
        transform.RotateAround(transform.position, Vector3.back, speed * Time.deltaTime);
    }
}
```

- Creamos un Script para la rotación del Vortex, le he puesto RotacionVortex y se lo he asignado al Vortex. En el Script declaro una variable entera publica que usare luego en el método de void Update(). Aquí llamaremos al transform.RotateAround que es el método que existe ya por defecto y le pasaremos 3 parametros. El primero es la posición, que es el propio centro de nuestro Vortex, luego asignaremos el eje que queremos rotar. Vector3 guarda el eje X, Y y Z, por ello asignamos el back que es el Z, y finalmente utilizamos la variable speed asignada arriba que nos servirá para verla visualmente en Unity y poder modificarla. Lo multiplicamos por Time.deltaTime que lo trae por defecto Unity esto se utiliza porque cada maquina tiene una potencia distinta y un refresco de frames diferentes y este Time.deltaTime se adapta.

2. Convertir el objecto Vortex en prefab, borrar el objecto Vortex original y poner unos cuantos Vortex sobre el terreno del juego. Probar el juego.

- Únicamente he arrastrado el objeto Vortex, a la carpeta Prefabs lo he eliminado del Hierarchy y he arrastrado el prefabs del Vortex a los lugares que he querido declararlo en mi Scene.



3. Comentar el código i explicar el funcionamiento del Script EnemyRespawner. Pensar como es podría modificar el código para que la regeneración de enemigos sea más o menos frecuente.

- Explico lo que hace el script en cometarios en el código.

```
using UnityEngine;
using System.Collections;
0 referencias
public class EnemyRespawner : MonoBehaviour
{
    //Variable que saca un enemigo
    public GameObject spawnEnemy = null;
    // Variable que sirve para decir cuanto tiempo tardara en aparecer otro enemigo de os vorteces cuando uno a muerto.
    float respawnTime = 0.0f;

    // Estos dos metodos los que hacen es detectar cuando un enemigo a muerto entonces lanzar el metodo scheduleRespawn
    0 referencias
    void OnEnable()
    {
        EnemyControllerScript.enemyDied += scheduleRespawn;
    }
    0 referencias
    void OnDisable()
    {
        EnemyControllerScript.enemyDied -= scheduleRespawn;
    }
    // Este metodo lo que hace es que cuando muere enemigo llaman a este metodo y el random del if decide si reparecen los enemigos o no.
    // Si reaparecen reaparecen en el tiempo que hemos marcado en el respawnTime, en este caso Time.time + 2.5f
    2 referencias
    void scheduleRespawn(int enemyScore)
    {
        // Randomly decide if we will respawn or not
        if (Random.Range(0, 10) < 5)
            return;
        respawnTime = Time.time + 2.5f;
    }
```

```
// Este metodo comprueba que el respawnTime es mas grande que 0.0f para que puedan reaparecer
// Una vez que es mayor revisa que el respawnTime es menor que el Time.time que por norma declarado arriba va a ser menor porque al Time.time se le suma el 2.5f
// Finalmente el respawnTime se reseta a 0.0f y se inicia la reaparacion del enemygo en la posicion transform.position que es la posicion que le tenemos asignada
// en unity en los vortices.
0 referencias
void Update()
{
    if (respawnTime > 0.0f)
    {
        if (respawnTime < Time.time)
    {
            respawnTime = 0.0f;
            GameObject newEnemy =
            Instantiate(spawnEnemy) as GameObject;
            newEnemy.transform.position = transform.position;
        }
    }
}
```

4. Implementar el marcador del juego utiliza la herramienta de generación de GUI de Unity. Explicar los pasos que se han seguido.

   - He buscado y buscado y mirados videos y no termino de entender como funciona el GUI para crear la puntuación. Porque al crear el UI – Text y al modificar (creo yo) el script de la bala que es la que colisiona con el objeto Enemy creo dos variables Public Text textContador y una private int puntuación = 0; al ir a la bala me sale la nueva variable del Script pero no me deja asignar el text contador que es el objeto UI que tengo dentro de Canvas. Luego creo un método OnTriggerEnter con un if para el tag Enemy a si solo se sumara puntuación cuando la bala haga collinder con Enemy, pero no he conseguido que me funcione porque no termino de entenderlo.

5. Hacer un listado con los conceptos introducidos en este documento, indicar la página y explicar qué son y para que se utilizan.

- Los conceptos nuevos introducidos han sido hacer un contador que sume puntos cada vez que mate a un enemigo.

Para ello hemos creado un objeto puntuación como hijo de la MainCamera con el fin de que la puntuación este continuamente en la pantalla del player.

Finalmente se crean dos Script uno para el objeto nuevo creado y otro para el enemy que escuchara el evento de cuando el enemy es eliminado y llamara al script del Score para sumar puntos.

- El otro concepto ha sido regenerar enemigos cuando son eliminados.

Para ello he creado una nueva textura de un Vortex que es donde regenerare los enemigos cuando mueran. Hago un nuevo Script que es el que esta comentado y explicado en la pregunta 3 asignándolo al Vortex