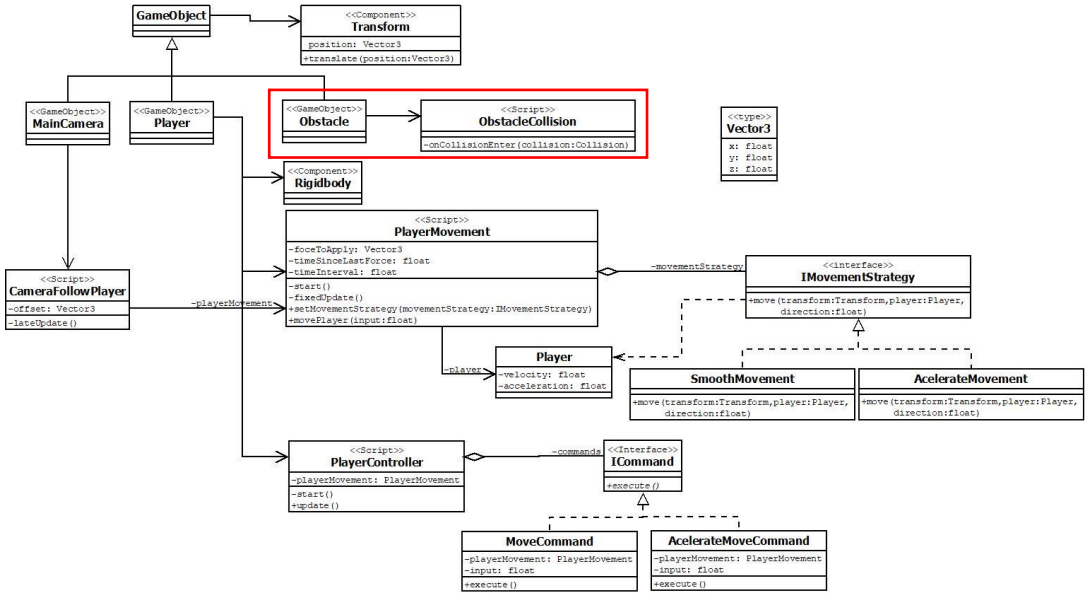
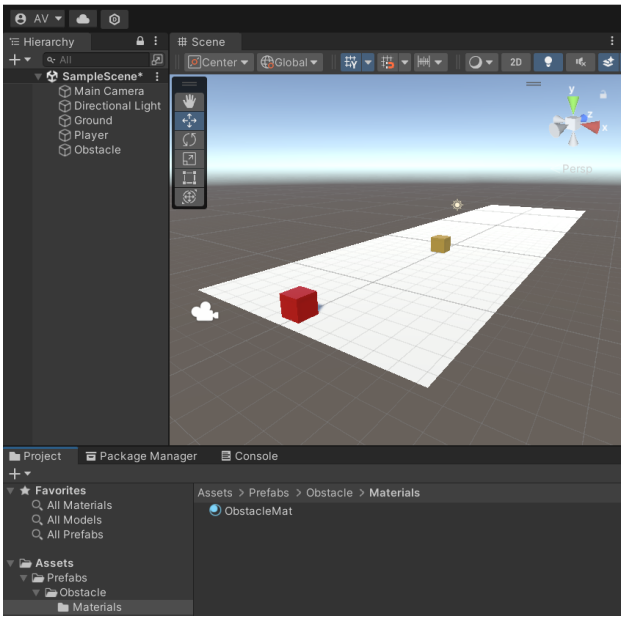
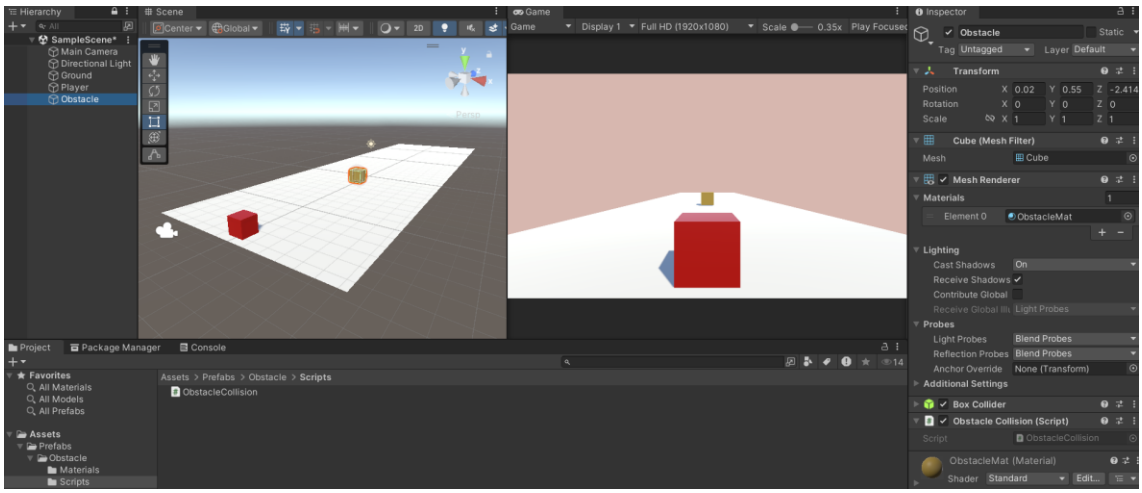
Historia de Usuario: Destrucción de obstáculos	
Identificador: UH06	Usuario: Jugador
Prioridad en negocio: -	Riesgo en desarrollo: -
Puntos estimados: -	Iteración asignada: -
Programador Responsable: Ariel Vega	
Descripción Como jugador quiero los obstáculos que el player choque sean destruidos para liberar el camino de estos obstáculos.	
Observaciones La colisión puede ser evaluada desde un evento por parte del player, como la del obstáculo. En este caso se solicita que sea a partir del obstáculo. Diagrama de clases  	
Criterios de aceptación • Los obstáculos se destruyen cuando el player los choca.	

Actividad 01: Crear un cubo con componente Rigidbody, y denominarlo Obstacle. Colocarlo en el camino. Además asignarle un material.



Actividad 02: Agregar un Script al Obstacle denominado ObstacleCollision.



Considere que, en este caso, los cubos son gameObjects que por defecto poseen ciertos componentes. Uno de ellos es BoxCollider. Este componente permite trabajar con las colisiones. Para responder a una colisión detectada puede trabajar con el método OnCollisionEnter()

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

Script de Unity | 0 referencias
public class ObstacleCollision: MonoBehaviour
{
    Mensaje de Unity | 0 referencias
    private void OnCollisionEnter(Collision collision)
    {
        Debug.Log("Está colisionando");
    }
}
```

Si ejecutamos el juego se notará que al momento de que el Player colisione con el obstáculo se visualizará el mensaje en la consola. Puede notar además que, si bien el obstáculo está en contacto con el suelo del camino, este no detecta colisión. Esto se debe a que el camino no tiene asignado un componente Collider.

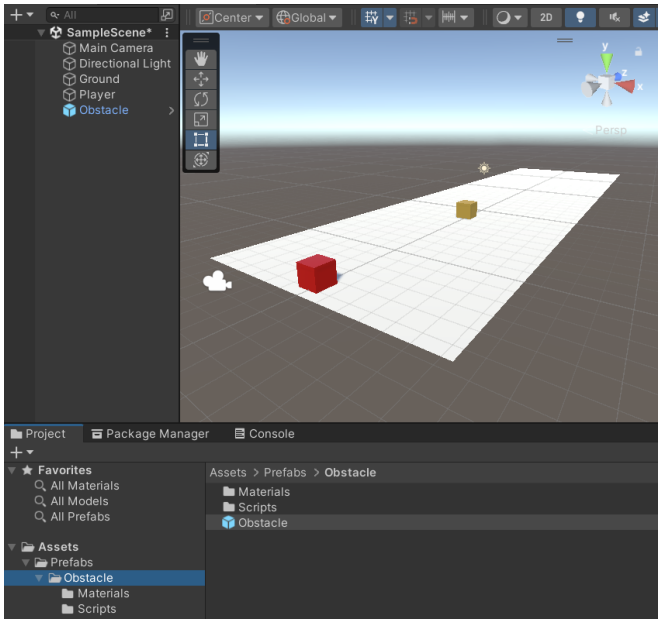
Actividad 03: Destruir el obstáculo cuando la colisión sea provocada por el Player. Aquí debemos agregar el siguiente código

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

Script de Unity | 0 referencias
public class ObstacleCollision: MonoBehaviour
{
    Mensaje de Unity | 0 referencias
    private void OnCollisionEnter(Collision collision)
    {
        if(collision.collider.name == "Player")
        {
            Destroy(gameObject);
        }
    }
}
```

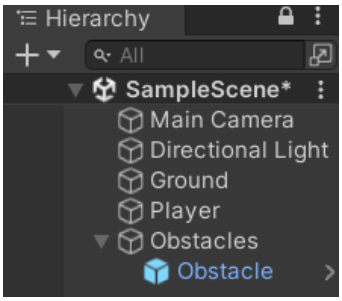
En este caso, se utiliza el nombre el gameObject para determinar si se ha colisionado con él. El nombre es el asignado en la jerarquía de gameObjects del juego. Puede usar también Tag, para lo cual primero debe asignarle al gameObject un tag adecuado.

Actividad 04: Hacer del gameObject Obstacle un prefab, y luego generar varios obstáculos en pantalla. Para hacer que un gameObject sea un prefab, arrástrelo a la carpeta prefabs destinada para que quede almacenado.

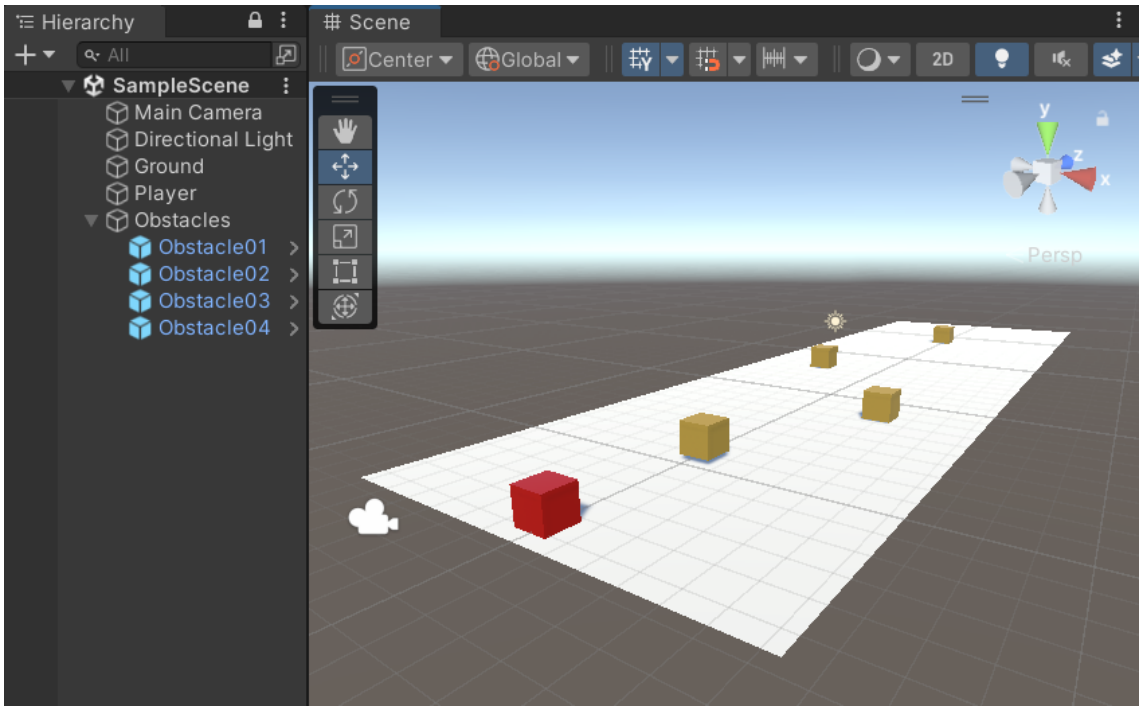


Puede notar que el gameObject toma un color celeste dentro de la jerarquía. Esto denota que es un obstáculo.

Ahora cree un Empty GameObject denominado obstacles y arrastre el obstáculo a ese gameObject



Duplique los gameObjects de tipo Obstacle y asígnele un nombre distintivo. Luego ubíquelos en la sección del camino que desee



Conclusión

Unity proporciona un mecanismo sencillo de detección de colisiones y programación de respuestas a través de los Scripts y los componentes Collider. Estos se deben optimizar para que la carga del programa no se vea afectada. Algunos de estos aspectos se desarrollarán más adelante.