

Prototipo 1: Simulador de conducción

1. Antes de empezar:

- Crear un nuevo proyecto en Unity (selecciona el Template 3D)
- Importar el paquete correspondiente al prototipo (Assets > Import Package > Custom Package).

2. Organizamos la escena:

- Abre la escena correspondiente al prototipo (la escena Sample Scene se puede borrar sin guardar)
- Selecciona un vehículo de los disponibles en la carpeta de Course Library en la ventana de proyecto y arrástralo a la escena.
- Selecciona un obstáculo de la librería y posicóinalo delante del vehículo
- Mueve la cámara para que enfoque la parte trasera del vehículo.
- Reconfigura la disposición de las ventanas de Unity a tu gusto.

3. Comenzamos a configurar el movimiento del vehículo:

- Comenzamos creando una carpeta en la ventana de proyecto para guardar los scripts.
- Dentro de la nueva carpeta creamos un nuevo script de C# con el nombre PlayerController.
- Una vez creado, lo arrastramos hasta el vehículo. Asegúrate en la ventana de Inspector que el script se ha añadido al vehículo.
- Edita el script y añade el código para que quede como:

```
public class PlayerController : MonoBehaviour
{
    public float velocidad = 20.0f;
    // Start is called before the first frame update
    void Start()
    {
    }
    // Update is called once per frame
    void Update()
    {
        transform.Translate(Vector3.forward * Time.deltaTime * velocidad);
    }
}
```

- Comprueba que al ejecutar el prototipo el vehículo se mueve y atraviesa los obstáculos.

4. Ahora damos cuerpo y masa a los objetos para que interaccionen:

- Ve seleccionando al vehículo y a los obstáculos y añádeles a cada uno un componente Rigidbody (Add Component en la ventana de Inspector de cada uno de ellos).
- Incrementa la masa de los objetos hasta una cantidad que pudiera ser realista y ejecuta de nuevo el prototipo.
- Por último, coloca nuevos obstáculos. Una forma fácil es seleccionarlos en la ventana de Jerarquía y tecleando Ctrl+D para duplicarlos.

5. Reposicionamos la cámara y la hacemos seguir al vehículo:

- Creamos un nuevo script al que vamos a llamar FollowPlayer y lo añadimos a la cámara

```
public class FollowPlayer : MonoBehaviour
{
    public GameObject player;
    // Start is called before the first frame update
    void Start()
    {
    }
    // Update is called once per frame
    void Update()
    {
        transform.position = player.transform.position;
    }
}
```

- Ahora seleccionamos la cámara y arrastramos el objeto correspondiente al vehículo a la variable del script dentro de la ventana del inspector.
- En la sección update del script lo que se está haciendo es asignar a la cámara la posición del jugador. No obstante, si probamos a ejecutar la cámara estará debajo del vehículo. Debemos ajustar su posición con un desplazamiento u offset.
- Para arreglar la posición de la cámara debemos modificar la línea del script:

```
transform.position = player.transform.position;
```

por, por ejemplo:

```
transform.position = player.transform.position + new Vector3(0, 5, -7);
```

- También podemos declarar una variable Vector3 para poder modificar el valor del offset de la cámara con más facilidad:

```
public class FollowPlayer : MonoBehaviour
{
    public GameObject player;
    private Vector3 offset = new Vector3(0, 5, -7);
    // Start is called before the first frame update
    void Start()
    {
    }
    // Update is called once per frame
    void Update()
    {
        //transform.position = player.transform.position;
        transform.position = player.transform.position + offset;
    }
}
```

- Podemos comprobar al ejecutar que la imagen que proporciona la cámara tiembla. Para solucionarlo y suavizar la actualización de la cámara con la posición del vehículo podemos cambiar el método Update del script por el método LateUpdate. Este método se llama después del método Update.

6. Vamos a tomar el control del vehículo de manera que podamos hacerlo girar a derecha e izquierda, acelerar y frenar detectando las pulsaciones sobre las teclas.

- En primer lugar vamos a permitir que el vehículo se mueva de derecha a izquierda. Desde el menú superior pinchando **Edit > Project Settings** y seleccionando **Input Manager** podemos ver la configuración de los controles.
Para llevarlo a la práctica modificamos el script PlayerController

```
public class PlayerController : MonoBehaviour
{
    public float velocidad = 20.0f;
    public float velocidadGiro = 45.0f;
    public float horizontalInput;
    // Start is called before the first frame update
    void Start()
    {
    }
    // Update is called once per frame
    void Update()
    {
        horizontalInput = Input.GetAxis("Horizontal");
        transform.Translate(Vector3.forward * Time.deltaTime * velocidad);
        transform.Translate(Vector3.right * Time.deltaTime * velocidadGiro *
horizontalInput);
    }
}
```

- Ahora vamos a controlar la aceleración y la frenada. Para ello modificamos el script de la siguiente manera:

```
public class PlayerController : MonoBehaviour
{
    public float velocidad = 20.0f;
    public float velocidadGiro = 45.0f;
    public float horizontalInput;
    public float forwardInput;
    // Start is called before the first frame update
    void Start()
    {
    }
    // Update is called once per frame
    void Update()
    {
        horizontalInput = Input.GetAxis("Horizontal");
        forwardInput = Input.GetAxis("Vertical");

        transform.Translate(Vector3.forward * Time.deltaTime * velocidad *
forwardInput);
        transform.Translate(Vector3.right * Time.deltaTime * velocidadGiro *
horizontalInput);
    }
}
```

- En este punto vemos que el vehículo, no gira sino que se desliza de un lado a otro. Podemos cambiar el comportamiento modificando el script:

```

public class PlayerController : MonoBehaviour
{
    public float velocidad = 20.0f;
    public float velocidadGiro = 45.0f;
    public float horizontalInput;
    public float forwardInput;
    // Start is called before the first frame update
    void Start()
    {
    }
    // Update is called once per frame
    void Update()
    {
        horizontalInput = Input.GetAxis("Horizontal");
        forwardInput = Input.GetAxis("Vertical");

        transform.Translate(Vector3.forward * Time.deltaTime * velocidad *
forwardInput);
        //transform.Translate(Vector3.right * Time.deltaTime * velocidadGiro *
horizontalInput);
        transform.Rotate(Vector3.up, velocidadGiro * horizontalInput *
Time.deltaTime);
    }
}

```

Actividades propuestas sobre el prototipo 1

1. **Grupos de obstáculos apilados:** crea pilas, montones o pirámides de obstáculos para que el vehículo los atraviese.
2. **Vehículos en dirección contraria:** añade un par de vehículos más que circulen automáticamente por la carretera en la dirección opuesta, que el jugador también tiene que evitar.
3. **Cambio de cámara:** permite que el jugador presione una tecla en el teclado para cambiar las vistas de la cámara. La misma tecla alternaría entre dos vistas, una arriba y detrás del vehículo, y la otra, desde la perspectiva del asiento del conductor.
4. **Opción multijugador:** modifica el prototipo para convertirlo en un juego de pantalla dividida “multijugador” con dos vehículos, donde el auto de un jugador esté controlado por las teclas WASD y el otro está controlado por las teclas de flecha.

Soluciones a las actividades propuestas

2. Vehículos en dirección contraria:

Añadimos desde la carpeta de Assets un vehículo a la escena. Le añadimos un componente Rigid Body y ajustamos su masa.

Añadimos un script a la carpeta Scripts que va a mover el vehículo en sentido contrario. Su código podría ser:

```
public class SentidoContrario : MonoBehaviour
{
    public int velocidad;
    // Start is called before the first frame update
    void Start()
    {
    }
    // Update is called once per frame
    void Update()
    {
        transform.Translate(Vector3.forward * velocidad * Time.deltaTime);
    }
}
```

Añade el script al vehículo que hayas escogido y ajusta la velocidad a tu gusto.

Crea una carpeta dentro de Assets llamada Prefabs si no la tienes y arrastra tu vehículo modificado a ella. Si aparece una ventana, selecciona **Original Prefab**.

Duplica el vehículo en la escena y colócalos en el otro extremo de la carretera y orientados en sentido contrario.

3. Cambio de cámara:

Modificamos el script PlayerController:

- añadimos las variables antes del método start:

```
public Camera mainCamera;
public Camera conductorCamera;
public KeyCode switchKey;
```

- en el método Update añadimos las líneas de código:

```
if(Input.GetKeyDown(switchKey))
{
    mainCamera.enabled = !mainCamera.enabled;
    conductorCamera.enabled = !conductorCamera.enabled;
}
```

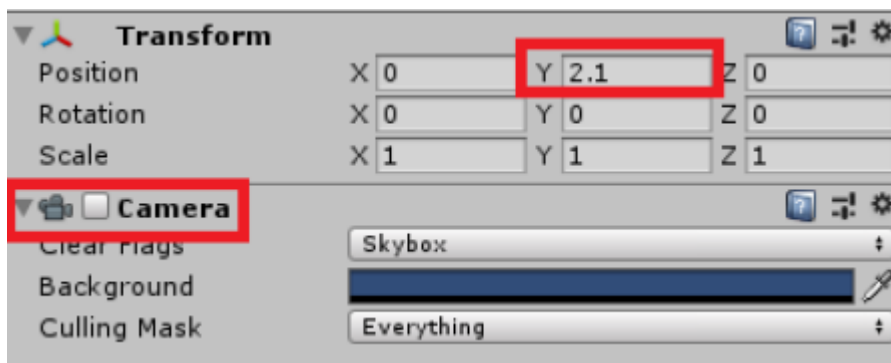
El script debe quedar:

```
public class PlayerController : MonoBehaviour
{
    public float velocidad = 20.0f;
    public float velocidadGiro = 45.0f;
    public float horizontalInput;
    public float forwardInput;
    public Camera mainCamera;
    public Camera conductorCamera;
    public KeyCode switchKey;
    // Start is called before the first frame update
    void Start()
    {
    }
    // Update is called once per frame
    void Update()
    {
        horizontalInput = Input.GetAxis("Horizontal");
        forwardInput = Input.GetAxis("Vertical");

        transform.Translate(Vector3.forward * Time.deltaTime * velocidad *
forwardInput);
        //transform.Translate(Vector3.right * Time.deltaTime * velocidadGiro *
horizontalInput);
        transform.Rotate(Vector3.up, velocidadGiro * horizontalInput *
Time.deltaTime);

        if (Input.GetKeyDown(switchKey))
        {
            mainCamera.enabled = !mainCamera.enabled;
            conductorCamera.enabled = !conductorCamera.enabled;
        }
    }
}
```

Después de guardar el script, crea una nueva cámara en la ventana de jerarquía y llámala conductorCamera. Arrástrala dentro del objeto vehículo y colocala de modo que enfoque desde la posición del conductor (valor de y aproximadamente 2.1). Deshabilítala de modo que el juego comience utilizando la cámara principal:



Por último, elimina de esta segunda cámara el componente Audio Listener que no se va a utilizar. Viene a ser un micrófono acoplado a la cámara y para la mayoría de aplicaciones tiene más sentido adjuntar el listener a la Camera principal.

En la misma ventana de jerarquía selecciona el objeto del vehículo del jugador y arrastra las dos cámaras al componente PlayerController y selecciona una tecla para cambiar de una a otra.

4. Opción multijugador:

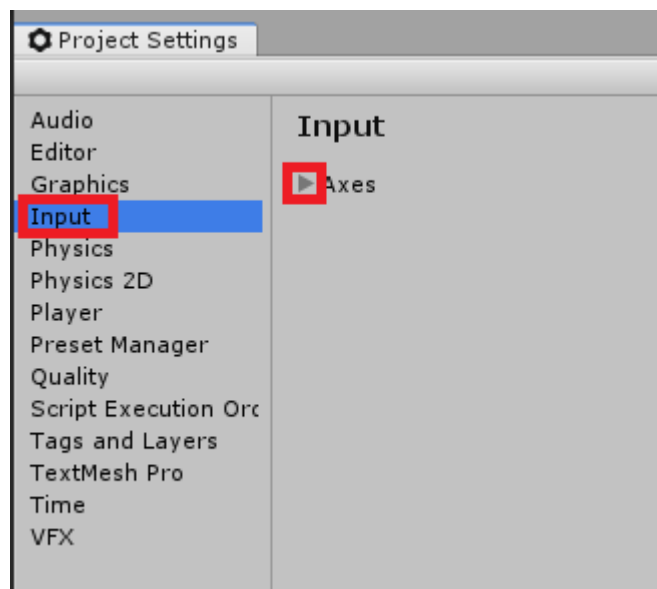
En el scrip PlayerController añadimos una nueva variable antes del método Start:

```
public string inputID;
```

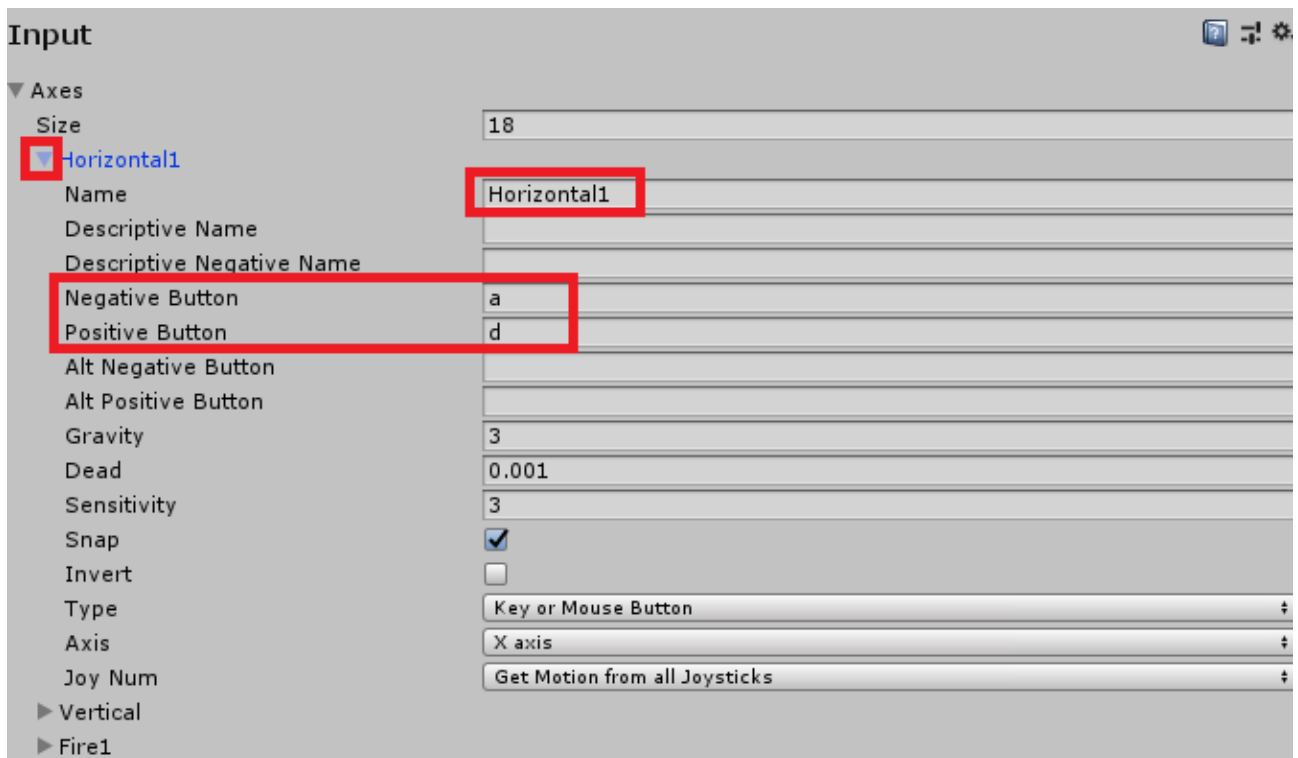
Dentro del método Update modificamos cómo leemos el teclado:

```
horizontalInput = Input.GetAxis("Horizontal" + inputID);  
forwardInput = Input.GetAxis("Vertical" + inputID);
```

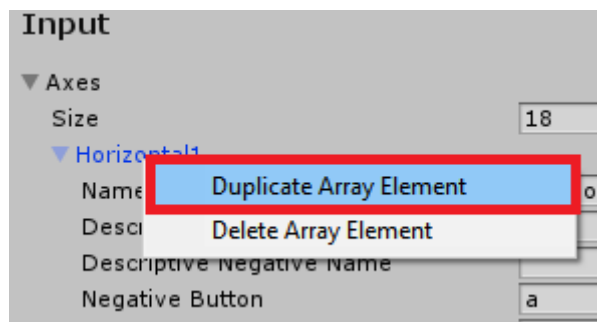
Ahora nos vamos a modificar el **Input Manager** al que se accede desde la pestaña **Edit > Project Settings**



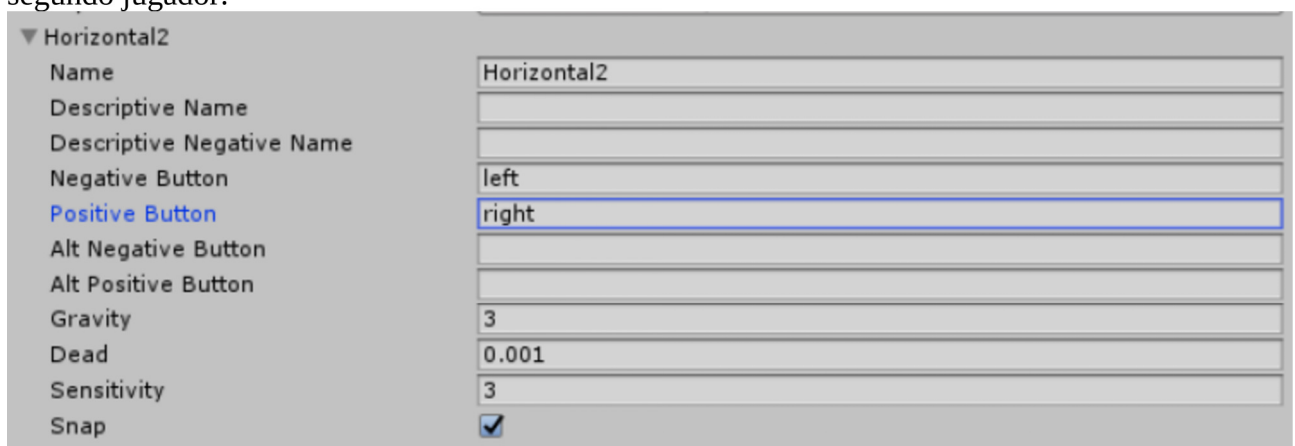
Renombramos el eje horizontal a Horizontal1 y cambiamos las teclas para controlar al vehículo:



Pulsamos el botón derecho del ratón con el cursor sobre **Horizontal1** y seleccionamos **Duplicate Array Element**



Renombramos el nuevo elemento como Horizontal2 y modificamos las teclas de control para el segundo jugador:



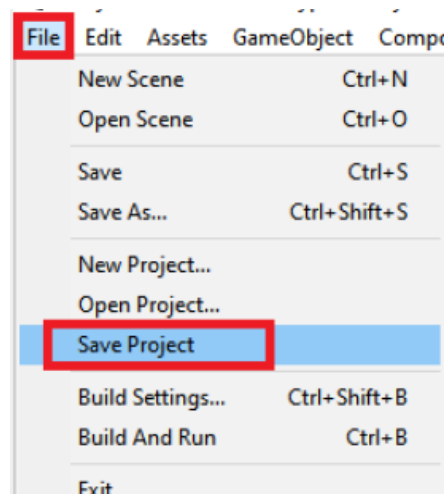
Hemos terminado con los controles horizontales. Hacemos lo mismo con los verticales. Renombramos **Vertical** como **Vertical1** y modificamos las teclas de control:

▼ Vertical1	
Name	Vertical1
Descriptive Name	
Descriptive Negative Name	
Negative Button	s
Positive Button	w
Alt Negative Button	
Alt Positive Button	
Gravity	3
Dead	0.001
Sensitivity	3
Snap	<input checked="" type="checkbox"/>
Invert	<input type="checkbox"/>
Type	Key or Mouse Button
Axis	X axis
Joy Num	Get Motion from all Joysticks

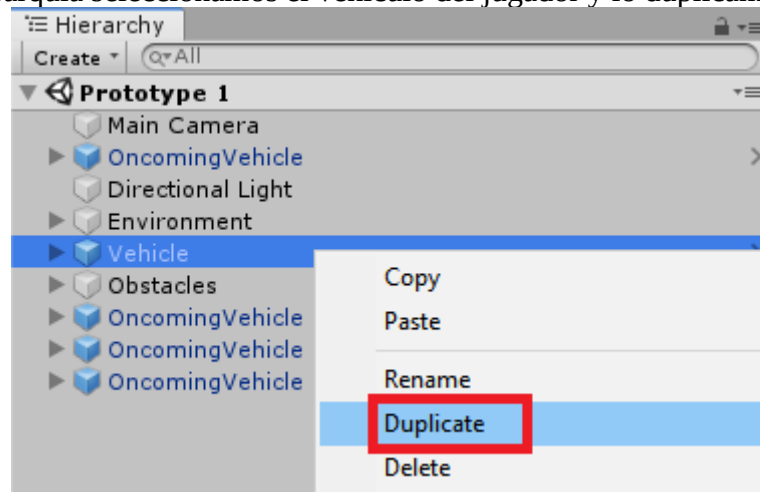
Duplicamos **Vertical1** como antes hicimos con **Horizontal1** y le cambiamos el nombre al duplicado a **Vertical2**. Cambiamos las teclas del segundo jugador:

▼ Vertical2	
Name	Vertical2
Descriptive Name	
Descriptive Negative Name	
Negative Button	down
Positive Button	up
Alt Negative Button	
Alt Positive Button	
Gravity	3
Dead	0.001
Sensitivity	3
Snap	<input checked="" type="checkbox"/>

Ahora guardamos el proyecto.

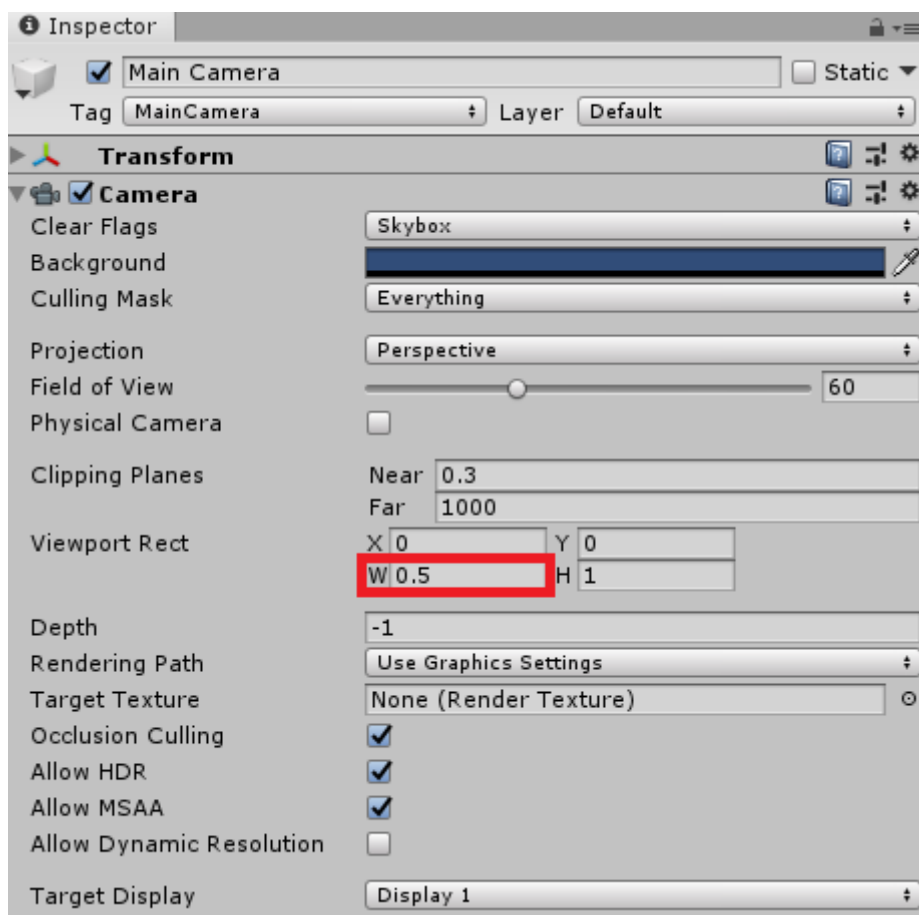


En la ventana de jerarquía seleccionamos el vehículo del jugador y lo duplicamos.



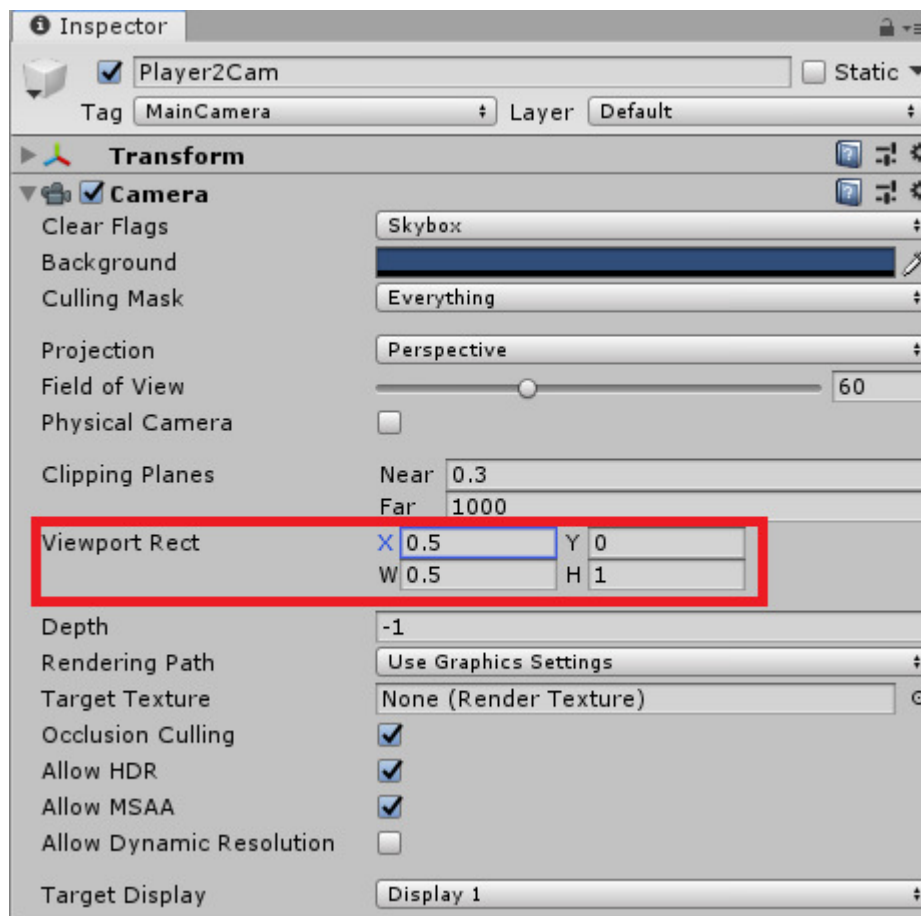
Hacemos lo mismo con la cámara principal y la duplicamos.

Una vez duplicada la cámara seleccionamos la cámara principal y en la ventana de inspección modificamos los valores del **Viewport Rect**:



Seleccionamos dentro del vehículo del primer jugador su cámara del conductor (conductorCamera) y ajustamos sus valores de **Viewport Rect** para que coincidan con los valores de su cámara principal.

A continuación seleccionamos la cámara secundaria y en la ventana de inspección modificamos los valores del **Viewport Rect**:

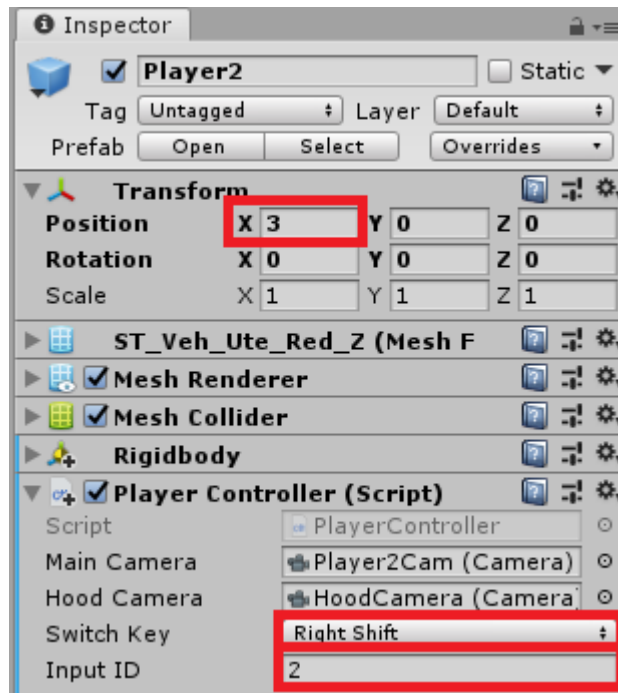


Podemos eliminar su componente Audio Listener como hicimos con la cámara del conductor anteriormente.

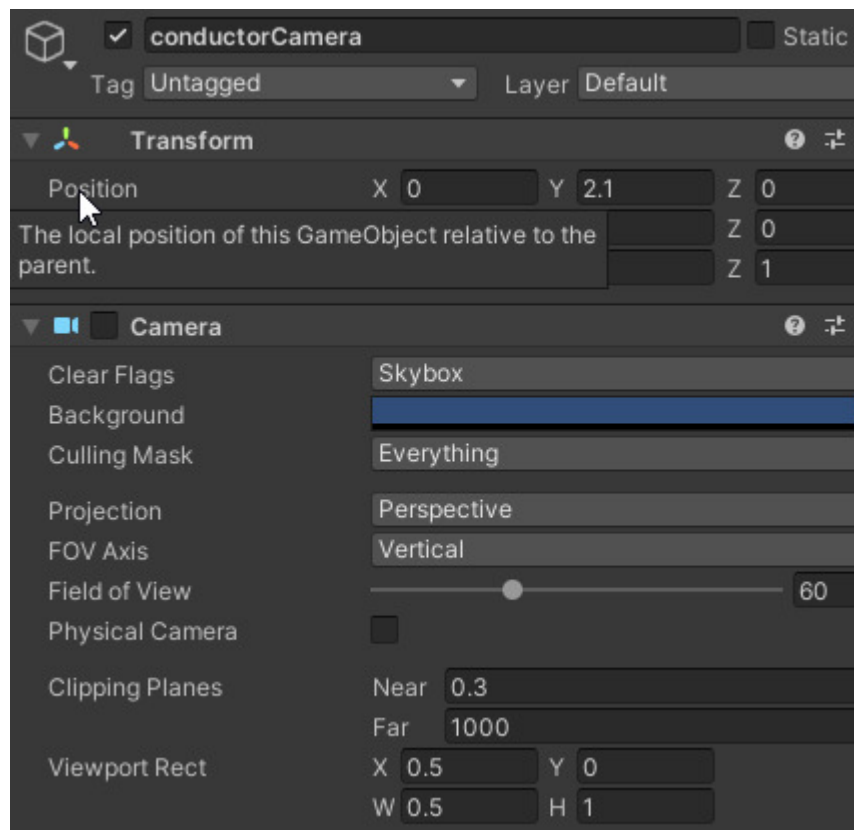
Sin salir de la ventana de inspección de esta cámara para el segundo jugador, arrastramos el objeto del segundo vehículo al script **Follow Player**.

Ahora seleccionamos en la ventana de jerarquía el objeto del segundo jugador y ajustamos los valores de su script **Player Controller**: hay que sustituir su Main Camera y la tecla para intercambiar cámara.

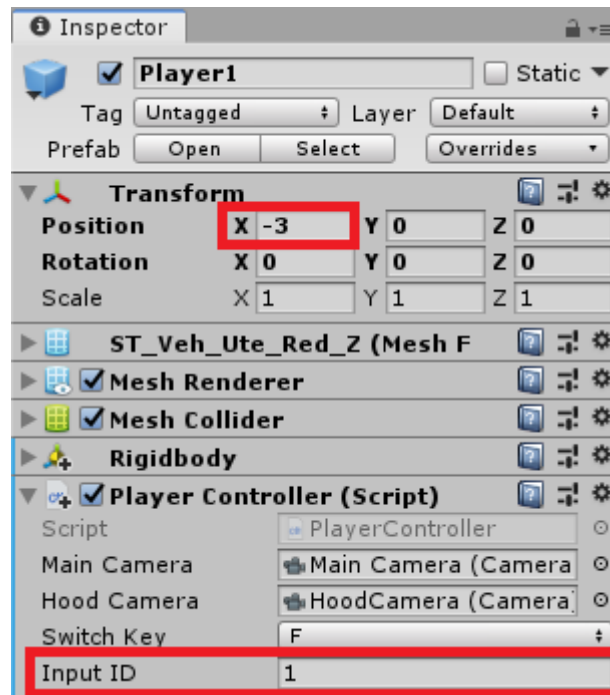
Aprovechamos para modificar la coordenada X de su posición para recolocararlo en la escena y modificamos el valor de su propiedad Input ID para que valga 2.



En la ventana de jerarquía seleccionamos la cámara del conductor del segundo jugador y modificamos los valores de Viewport Rect para que coincidan con los de su cámara principal:



Por último seleccionamos en la ventana de jerarquía al vehículo correspondiente al primer jugador y modificamos la coordenada X de su posición para que valga -3 y el valor de su propiedad Input ID para que valga 1.



Hemos terminado. Guardamos el proyecto y le damos a play.