



**Tecnológico
de Monterrey**

Laboratorio 1

Reporte

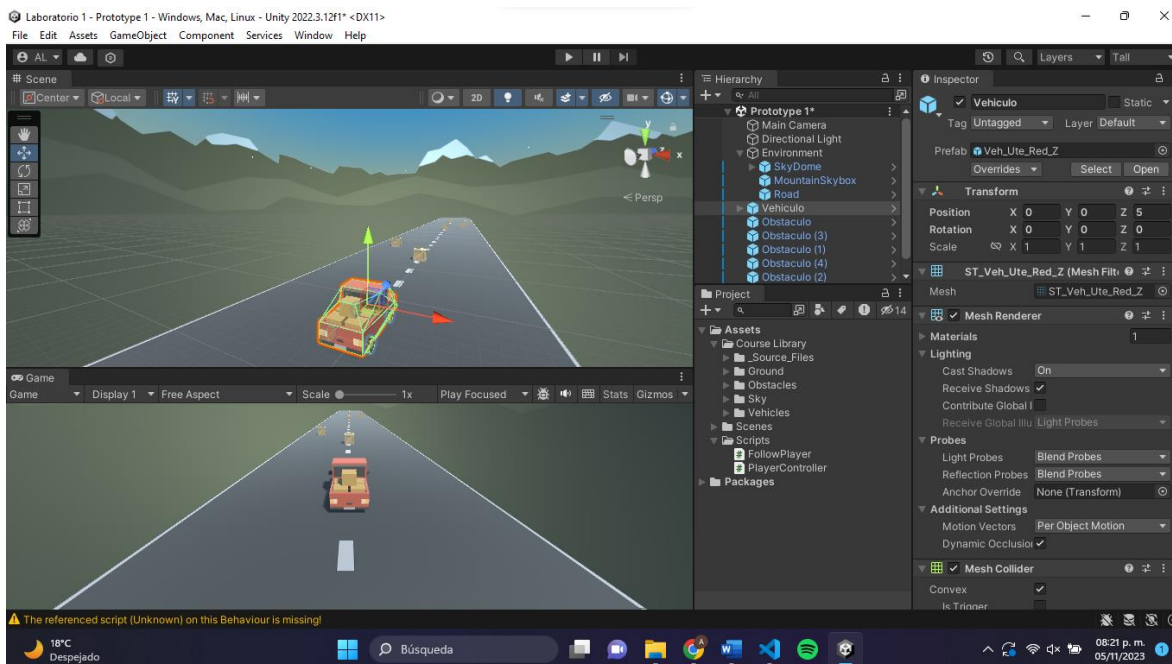
Arturo Díaz López A01709522

TC2008B.301

06/11/2023

Layout Seleccionado

Decidí seleccionar este layout por la facilidad que tiene para navegar entre cámara de edición y cámara de juego. Además, tener las pestañas de inspección / jerarquía / proyecto en la parte inferior le quitaba mucho espacio a los apartados de escena y juego, por lo que me pareció mejor idea moverlo a la derecha. Por último, la pestaña de inspección es más larga que ancha, por lo que es más práctico scrollar con el mouse de arriba para abajo para ver todos sus atributos y editarlos si es necesario.

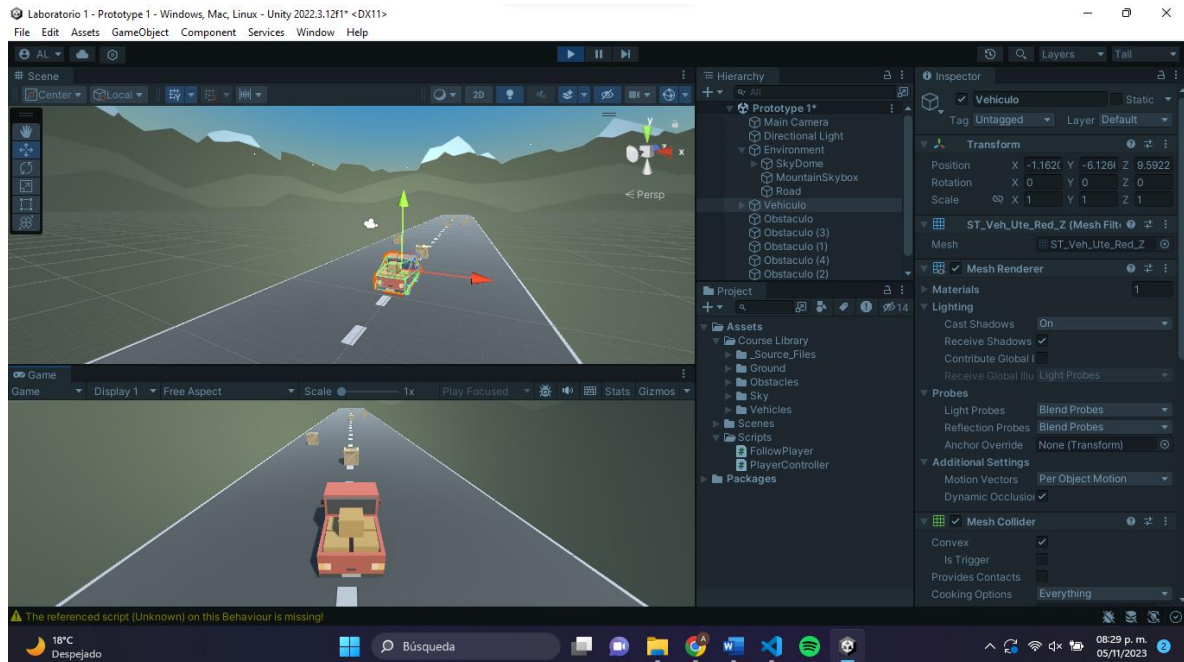


Reflexión Laboratorio 1-1

Este laboratorio fue una excelente introducción al desarrollo de aplicaciones gráficas en general, me ayudó a comprender de qué manera se ligan y relacionan los assets o interfaces con la lógica para que funcionen correctamente esos assets. Siempre he sido un aficionado de los videojuegos, y tener un acercamiento más profundo a su desarrollo fue algo que siempre quise experimentar, la verdad es que fue muy divertido moverle a todas las opciones y probar con distintos valores el funcionamiento del coche y sus obstáculos.

Fecha de realización del laboratorio: 03/11/2023

Layout de color en play mode:



Lista de errores que hasta el momento existen en el juego:

- La gravedad del coche funciona de manera extraña.
- Al llegar al borde del camino y tratar de incorporarte a él otra vez, el coche sale volando.
- Cuando el camino se acaba, el coche cae en un precipicio infinito.
- Deberían existir límites que no permitan que salgas del camino.
- El coche nunca se detiene (aunque caiga de cabeza o realice otras acciones poco realistas, seguirá avanzando).
- El coche puede atravesar la carretera (noclipping).

Código documentado de la clase FollowPlayer.cs

```
1  /// <summary>
2  /// This follow player class will update the events from the main camera.
3  /// Standar coding documentarion can be found in
4  /// https://learn.microsoft.com/en-us/dotnet/csharp/language-reference/language-specification/documentation-comments
5  /// </summary>
6
7  using System.Collections;
8  using System.Collections.Generic;
9  using UnityEngine;
10
11 public class FollowPlayer : MonoBehaviour
12 {
13     public GameObject player;
14     private Vector3 offset = new Vector3(0,6,-7);
15
16     /// <summary>
17     /// This method is called before the first frame update
18     /// </summary>
19     void Start() {
20
21     }
22
23     /// <summary>
24     /// This method is called once per frame
25     /// </summary>
26     void LateUpdate() {
27         transform.position = player.transform.position + offset;
28     }
29 }
30
```

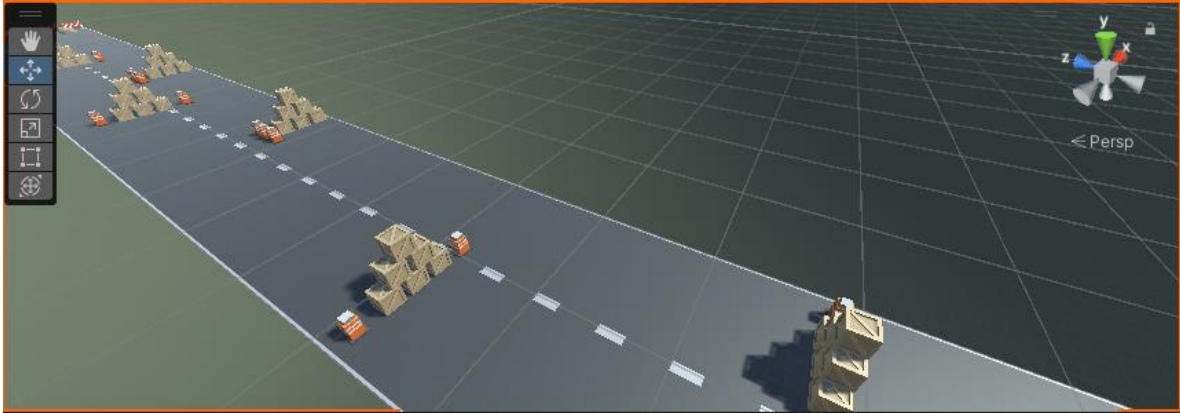
Reflexión Laboratorio 1-2

Este segundo laboratorio me permitió entender mejor cómo se construyen los scripts que van ligados a los objetos de un videojuego, y cómo los mismos nos ayudan a darle comportamiento e incluso vida. Una física real de un objeto le brinda una experiencia placentera al usuario, por lo que es muy importante prestarle la debida atención a los scripts donde se declaran estas propiedades.

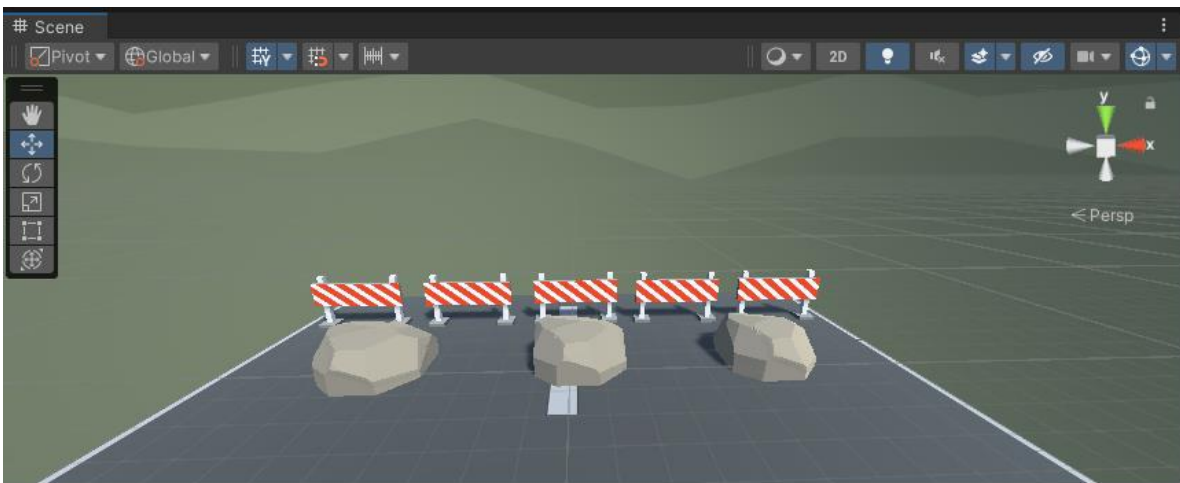
Fecha de realización: 04/11/2023

Prefabs de obstáculos

1. Cones: Agregué conos a las orillas de las pirámides para agregar dificultad a la hora de esquivar los obstáculos.



2. Rocks: Agregué piedras al final del camino para dificultar el paso del coche al precipicio y evitar que el jugador caiga en un infinito (o al menos que caiga con estilo).
3. Barriers: Con el mismo fin, agregué barreras que delimitan el límite del camino manejable.



Resultado Final

[Laboratorio 1.mp4](#)

Reflexión Laboratorio 1-3

Este último laboratorio me ayudó a comprender de qué manera funcionan los juegos cooperativos, y me causó mucha intriga de qué manera se resolvería el manejar los inputs para ambos jugadores. Antes de ver la solución hice un intento por solucionar este problema por mi cuenta, hardcodeando el horizontalInput1, 2 y verticalInput 1, 2, pero este acercamiento no funcionaba ya que seguía sin distinguir para qué jugador se debía realizar la acción, en cambio, dio un resultado gracioso, cuando presionaba las dos teclas al mismo tiempo (por ejemplo, w y up para ir hacia adelante), el coche se movía el doble de rápido ya que calculaba la velocidad para cada tecla y la sumaba.

En general, estas primeras 3 actividades me ayudaron a identificar de qué manera funciona Unity, la creación de componentes, objetos y scripts que pueden dar vida a un juego o aplicación interactiva.

Fecha de realización: 06/11/2023