



Tecnológico de Monterrey

Instituto Tecnológico de Estudios Superiores Monterrey

CAMPUS QUERÉTARO

Modelación de sistemas multiagentes con gráficas computacionales

Introducción a Unity

Alumno

Erick Alfredo García Huerta - A01708119

Fecha:

07 de noviembre de 2022

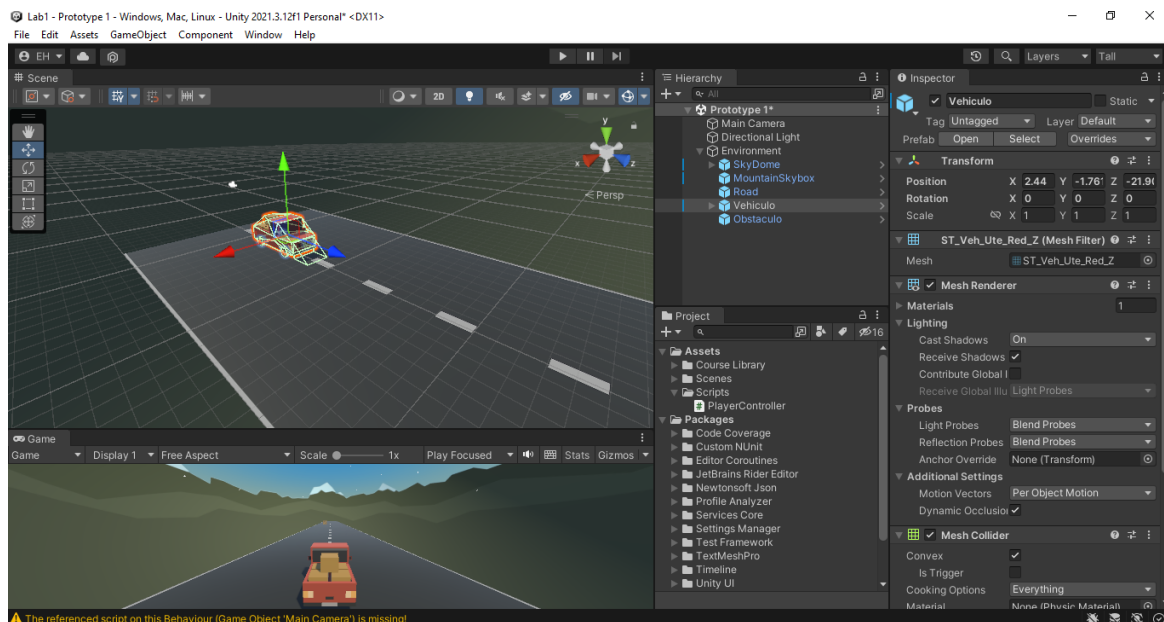
Laboratorio 1 – Starting Unity

Reflexión (01 – noviembre – 2022)

Antes de comenzar a usar Unity me documenté un poco en las cosas que se podían hacer con él, así como las distintas capas de funcionamiento que tenía, esto me intimidó bastante y eso dejando de lado que tenía miedo de que mi computadora no fuera capaz de correr los proyectos que tenía que llevar a cabo para los cursos.

Sin embargo, una vez hecho el laboratorio me he dado cuenta de que es una herramienta bastante intuitiva una vez puedes ubicar sus componentes más importantes, así como que mi computadora puede correr los proyectos sin más problemas que unos cuantos segundos de espera mientras los proyectos cargan para correr. Así que el mayor aprendizaje que me llevo es que Unity puede ser una herramienta que con el debido entrenamiento puede ser bastante fácil de usar para un primerizo. Esto sin su buena tanda de contras como que uno va a necesitar de un mouse externo para llevar a cabo la mayoría de las acciones, porque la aplicación no está pensada para ser usada en dispositivos portátiles, así como la necesidad de una o más pantallas se hace presente para tener un flujo de trabajo menos pausado.

Layout seleccionado



Justificación

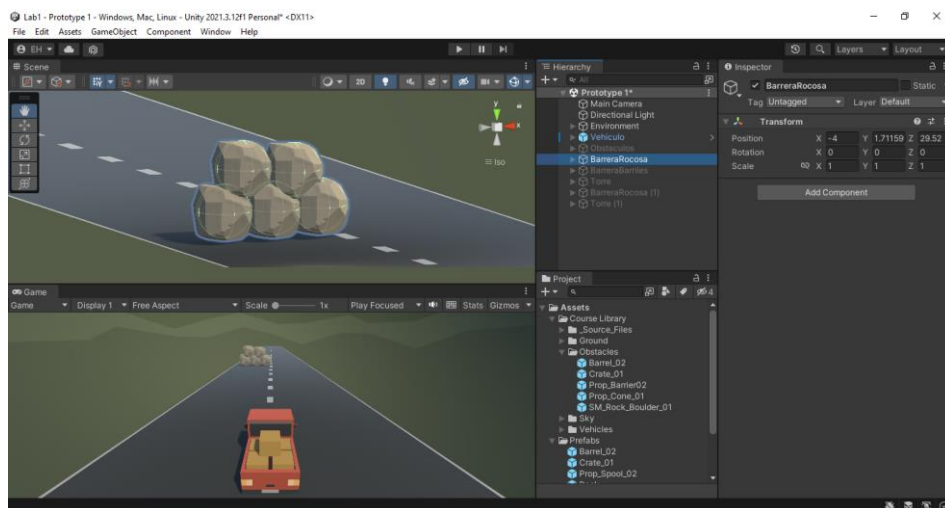
La principal razón por la que escogí este layout es porque me permite ver todas las partes importantes que necesito para modificar las propiedades de los objetos, dejando espacio suficiente para las vistas de la cámara y el acomodo de los distintos elementos en el mapa. Además de que el acomodo de los elementos en el layout deja a la altura de mis ojos elementos que se ocupan en todo momento como lo son la jerarquía y la escena, dan espacio suficiente para el inspector para que la mayoría de las características ocupen espacio y deja accesibles las carpetas del proyecto para arrastrar lo elementos necesarios sin andar navegando demasiado entre pestañas.

Errores presentes en el juego

- Las hit boxes de los elementos en el mapa son más grandes que el objeto en cuestión.
- El comportamiento del automóvil una vez es lanzado al aire es errático e imposible de controlar correctamente.
- A veces cuando colisiona el coche con cualquier objeto, atraviesa el suelo.
- El movimiento del auto es antinatural y robótico.
- Aunque las colisiones funcionan correctamente a lo largo de la pista, de vez en cuando se atraviesan los objetos a pesar de que estos cuentan con un cuerpo rígido.

Obstáculos originales

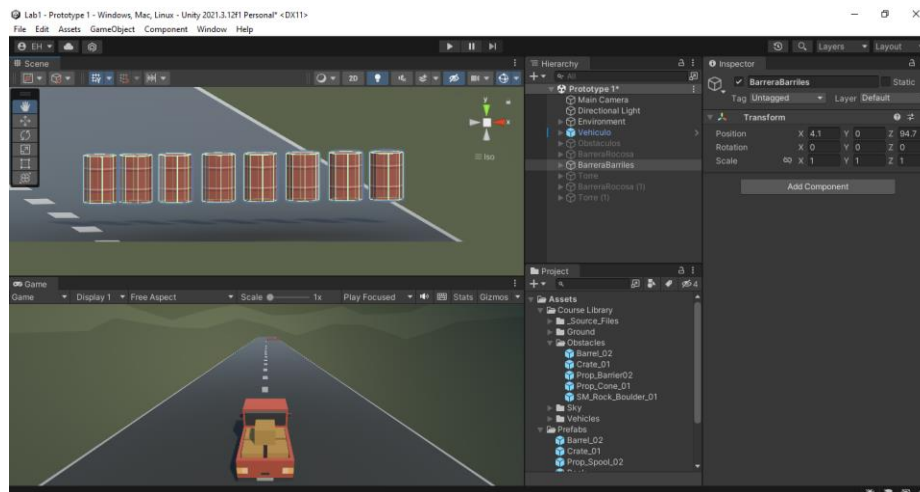
1. Barrera de rocas



Justificación

Me gustan mucho los juegos de plataformas y la mayoría de ellos tienen en sus niveles iniciales barreras de roca que bloquean el camino, así que intenté hacer una para mi primer proyecto de Unity. Además, debido a las colisiones irregulares que tiene el vehículo, chocar contra una barrera de este tipo haría que este comportamiento se viera incrementado y el control del auto en el aire es muy caótico, por lo que sería ideal para probar qué tanto podría el auto salir volando si chocaba contra un objeto medianamente esférico.

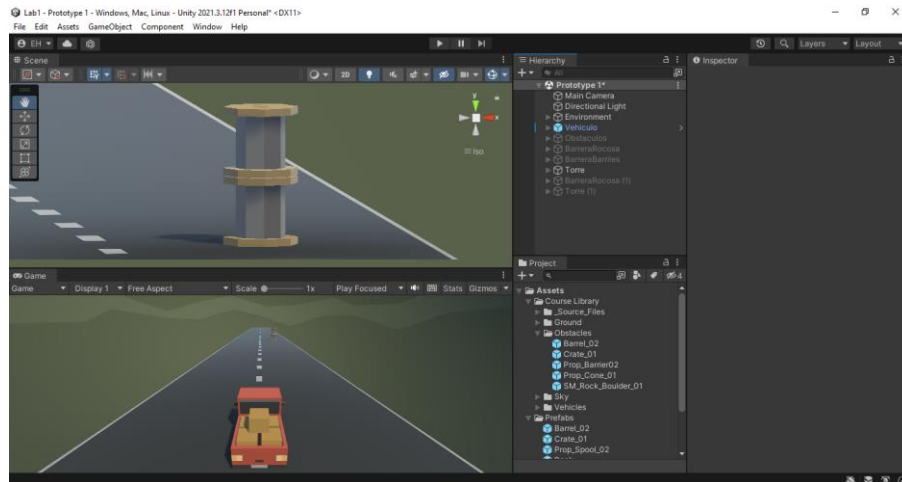
2. Barrera de barriles



Justificación

La idea de estos barriles era que cubrieran toda una sección de la carretera para forzar al jugador a tomar el otro carril y tener que cruzarlo con el tiempo suficiente para que los autobuses no le atropellasen.

3. Torre



Justificación

Este obstáculo es uno que estaría junto con las cajas debido a que no ocupa demasiado espacio en la pista y ya estaba demasiado llena, además de que sería fácil poner varias sin que los autobuses pasaran arriba de ellas y las lanzaran por todo el mapa.

Reflexión (02 – noviembre – 2022)

Con el agregado del rigid body y los obstáculos, Unity cada vez parece más una herramienta para crear videojuegos u otras piezas de medios interactivos. Del mismo modo, la forma tan extraña en la que interactúan los objetos, y todos los procesos que tiene que seguirse para que sus acciones tengan cierto semblante de autonomía y naturalidad demuestra que toma mucho tiempo llegar a controlar las herramientas con precisión.

Además de que la construcción de estos objetos me ha parecido demasiado tardada, por lo que me lleva a pensar que en un futuro debe de haber una forma más eficiente de hacer estas construcciones, además de que una vez se me presentaron los medios para hacer construcciones personalizables y fácilmente replicables me encontré usando mucho esta herramienta. Por desgracia mi aplicación crasheó poco después de que terminara de probar los elementos que creé, por lo que me hice más consciente de que, si bien estas herramientas son muy útiles, hacer uso de ellas inicialmente toma mucho tiempo y es muy seguro que en un futuro me encuentre guardando compulsivamente el archivo para que no se repita el suceso.

Código de la clase FollowPlayer

```
using System.Collections;
```

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

```
using UnityEngine;
```

```
public class FollowPlayer : MonoBehaviour
```

```
{
```

```
    // Variable que hace que liga la cámara con el objeto que designemos como nuestro jugador
```

```
    public GameObject player;
```

```
    // Variable que toma los valores del offset con la posición de la cámara
```

```
    private Vector3 offset = new Vector3(0,6,-7);
```

```
    // Start is called before the first frame update
```

```
    void Start()
```

```
    {
```

```
    }
```

```
    // Update is called once per frame
```

```

//void Update()

// LateUpdate se manda a llamar después del Update

void LateUpdate()

{

    // transform.position = player.transform.position;

    // transform.position = player.transform.position + new Vector3(0,6,-7);

    // Modifica la posición de la cámara con en base al movimiento del jugador.

    transform.position = player.transform.position + offset;

}

}

```

Reflexión (03 – noviembre – 2022)

Nunca había escrito nada en C# y por un segundo tenía miedo de que mi falta de conocimiento se volviera un problema para llevar a cabo las actividades del laboratorio, pero una vez pude unir los puntos de cómo funcionaba la asignación de elementos y cómo los scripts son reutilizables para múltiples elementos. Además de que, si bien hay que hacer bastantes líneas de código y saber bien qué variables están ligadas a qué atributos de los elementos de Unity, es bastante fácil prever cómo se comportarán los elementos basados en los nombres de los comandos.

Además, me sorprendió lo fácil que sería implementar un multijugador local en el juego, así como la división de controles. Si bien en un inicio tuve algunos errores que impedían que los

autos se pudieran mover, prontamente pude repararlos. Además de que me he acostumbrado cada vez más a la construcción de obstáculos y el acomodo de los elementos en el programa, por lo que ahora mi ritmo de trabajo ha aumentado y tengo menos problemas con posicionamientos extraños de los objetos y varias ideas de qué se puede hacer con el programa.

Enlace al video: <https://youtu.be/ug1MyOwN3Xg>

Explicación del juego

El objetivo del juego es simple, cruzar la intersección sin caer del mapa y derribar los barriles que se encuentran al final de la recta. En la demostración se dejó de lado al jugador dos y sólo se tomó control del jugador uno.