Tecnológico de Monterrey

Modelación de Sistemas Multiagentes con Gráficas Computacionales (Grupo 4)

M4-1. Actividad

Alumno

Javier E. Agostini Castilla

Profesor

Jorge Mario Cruz Duarte

Fecha

25 de Agosto del 2021

M4-1. Actividad

Introducción

Como parte de la actividad 1 del módulo 4 del bloque TC2008B, se diseñó un vehículo utilizando ProBuilder dentro de Unity, se le agregó un material y se codificó para que existiera cierta cantidad de vehículos y que todos estuvieran girando con respecto a un pivote, simulando a grandes rasgos el movimiento en una glorieta o rotonda.

Desarrollo

Esta actividad se fue desarrollando durante las sesiones de clase. El profesor nos dio la libertad de diseñar nuestro vehículo como quisiéramos, por lo que se decidió diseñar un camión de carga o tráiler.

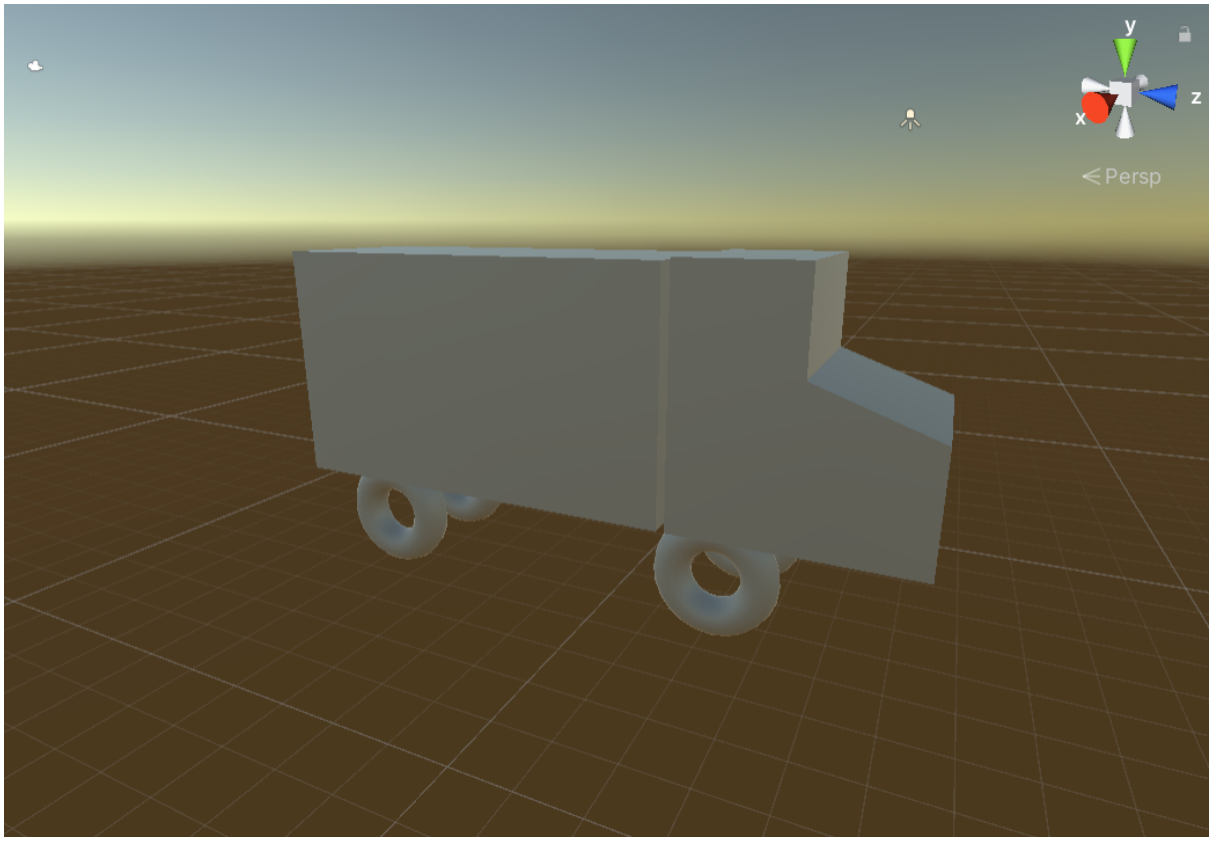


Figura 1.1 Imagen que muestra el diseño del tráiler.

Una vez diseñado el vehículo, se agruparon todos los componentes y se creó un prefab. Después, se creó el material que sería asignado al prefab para darle un poco de diseño.

Figura 2.1 Imagen que muestra el material diseñado. Creado modificando las variables Albedo y Specular.

Figura 2.2 Imagen que muestra el prefab del vehículo después de habérsele asignado el material creado.

Una vez asignado el material al prefab, se prosiguió a desarrollar el código en C#, el cuál permitiría que se crearan una cantidad definida de vehículos y que posteriormente estos estuvieran girando alrededor de un pivote.

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

public class Carros : MonoBehaviour
{
    // Inspector de Unity
    public GameObject PrefabCarro;
    public int NumeroCarros = 10;

    // Start is called before the first frame update
    void Start()
    {
        for(int i = 0; i < NumeroCarros; i++){
            float x = Random.Range(-25, 25);
            float y = 0;
            float z = Random.Range(-25, 25);

            Instantiate(PrefabCarro, new Vector3(x, y, z),
            Quaternion.Euler(90,0,0));
        }
    }

    // Update is called once per frame
    void Update()
    {

    }
}
```

Figura 3.1 Código de la clase Carros, la cuál define la cantidad de vehículos que aparecerán en la simulación, su posición, ángulo, así como el prefab a asignarle.

```
void TransformacionCarro()
{

    Vector3 newPosition =  position;

    Matrix4x4 rotation =  Transformations.RotateM(angle,
    Transformations.AXIS.AX_Y);
    newPosition = dehomogenise(rotation * homogenise(newPosition));

    GetComponent<MeshFilter>().gameObject.transform.position = newPosition;

}
```

Figura 3.2 Función que forma parte de la clase CarrosMoving, la cual define cómo se moverán los vehículos. Esta función realiza las rotaciones y traslaciones de los objetos por medio de matrices.

Una vez desarrollado el código, es necesario crear un GameObject vacío y asignarle el script Carros, para después asignarle el prefab que se hizo anteriormente.

Después, se cambió la luz de direccional a spot, así como se ajustó el rango, el ángulo y la posición de la luz. A pesar de que no se aprecie mucha diferencia a simple vista, se podrá observar este cambio al correr la simulación.

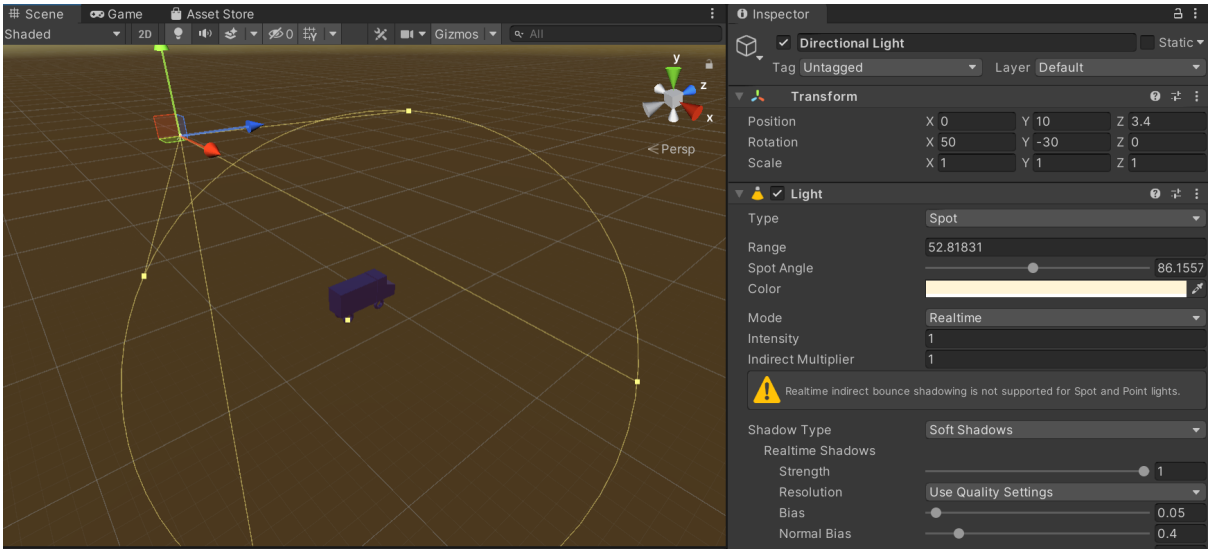


Figura 4.1 Captura de pantalla que muestra la luz de spot dentro de la simulación.

Finalmente, se ajustó la posición y ángulo de la cámara principal, para que, al correr la simulación, se puedan apreciar de mejor manera los vehículos rotando.

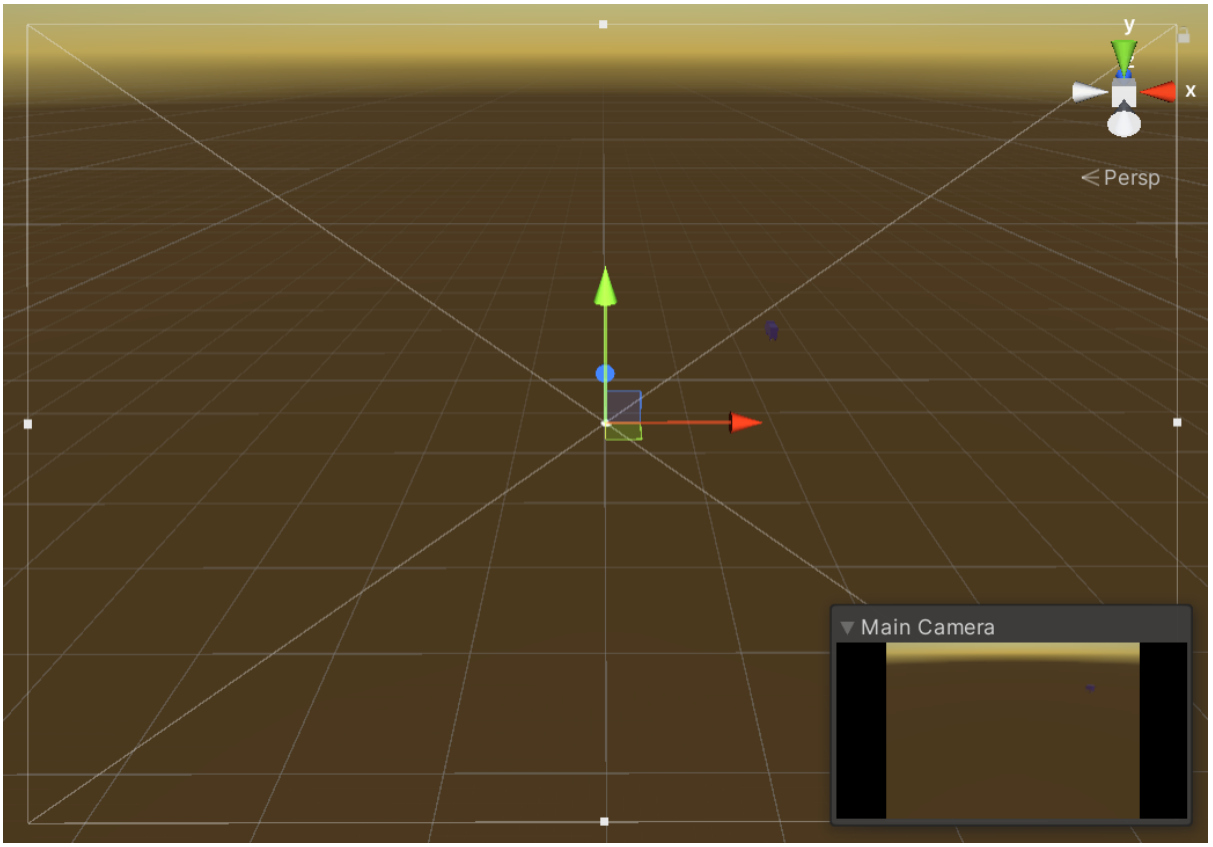
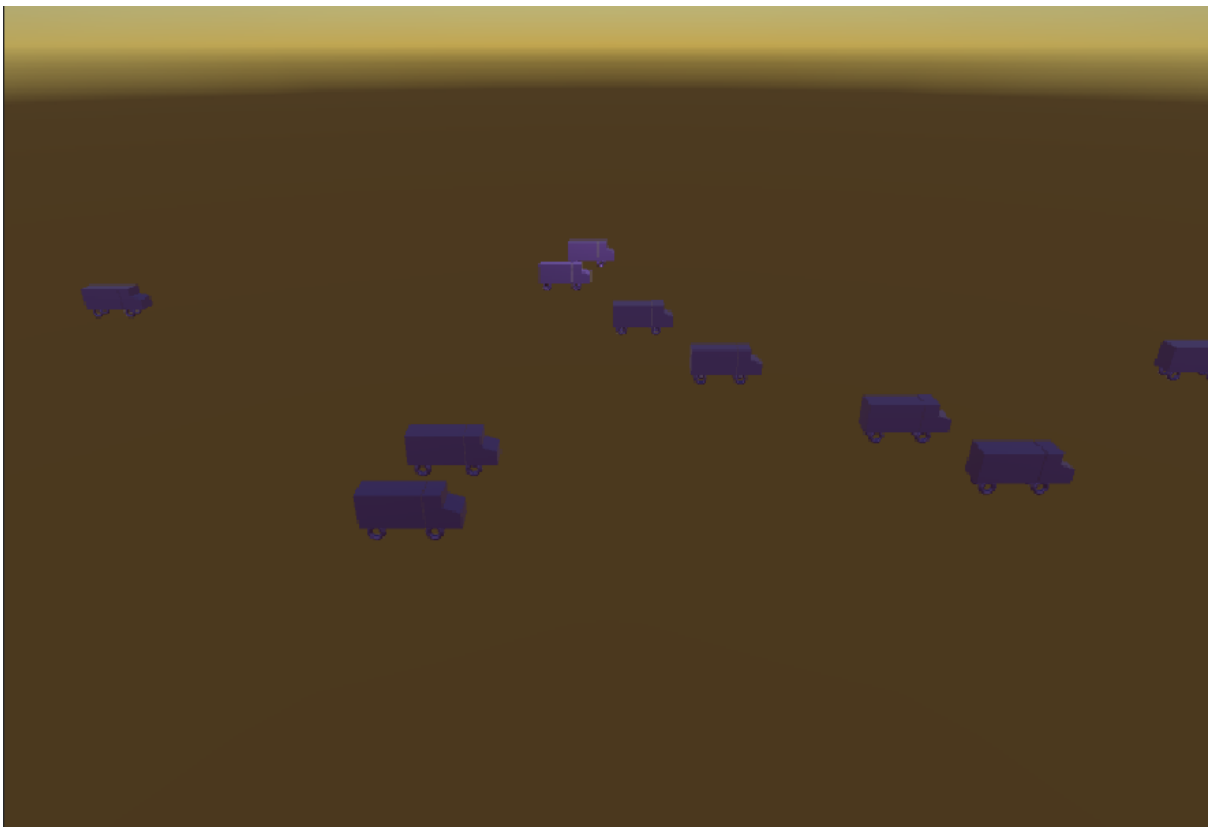
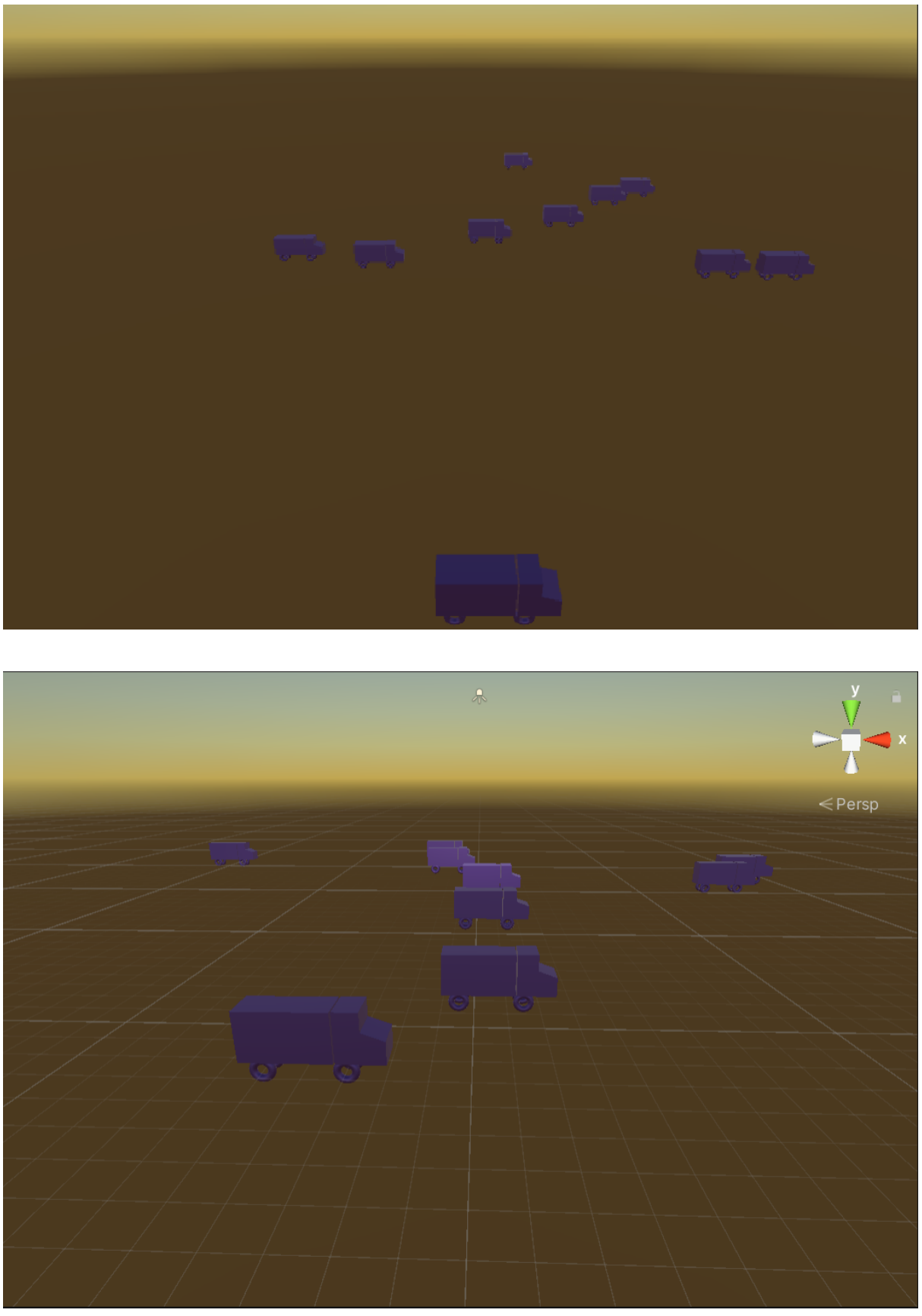


Figura 5.1 Imagen que muestra el posicionamiento de la cámara.

Una vez realizado todo esto, es posible correr la simulación y observar la rotación de los vehículos. A continuación, se muestran algunas imágenes de la simulación:





Figuras 6.1, 6.2 y 6.3 Imágenes que muestran la simulación corriendo. Los vehículos se colocan en posiciones aleatorias y comienzan a girar con respecto a un pivote central.

Conclusión

En conclusión, esta actividad me permitió entender de mejor manera el rol que tiene Unity en la solución del reto, así como diversas aplicaciones como traslación y rotación de objetos por medio de matrices. Aprendí sobre el diseño de objetos con ProBuilder, así como la utilización de materiales y texturas para darles un toque más realista o estético a los objetos que estemos utilizando.