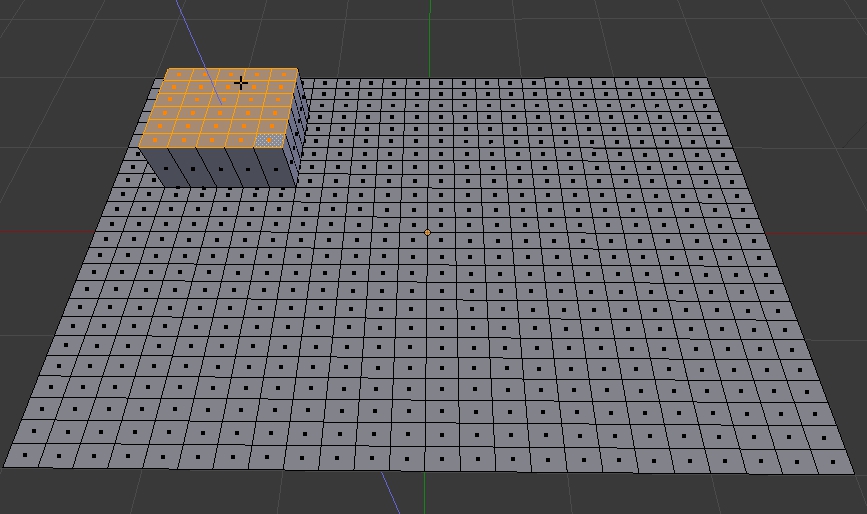
# Capítulo 1: Diseño y modelos

## Diseño del Mapa

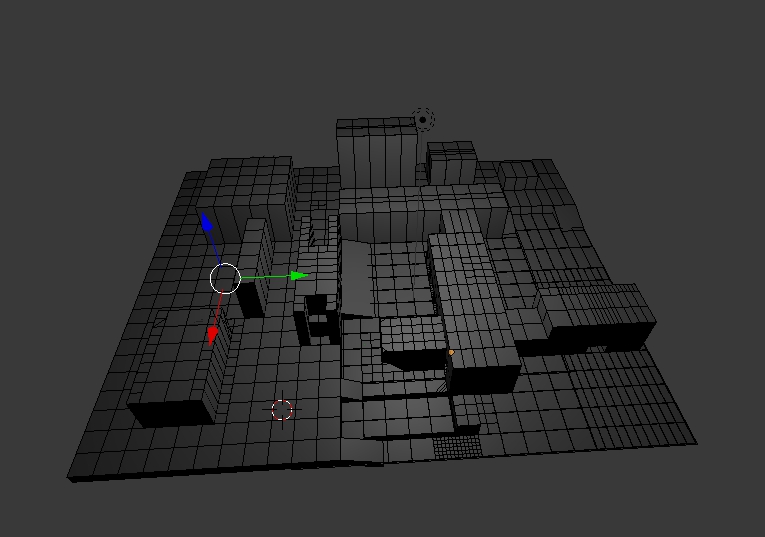
Para modelar el mapa y respectivamente los edificios se consideró tomar fotografías a todo el plantel, dado que este proceso llevaría bastante más tiempo para el modelado se optó por realizar los modelos basados en aproximaciones únicamente.

Los edificios se modelaron a partir de un plano base subdividido para facilitar un poco el trabajo de diseño tal y como se aprecia en la figura 1.1.



**Figura 1.1**

Este proceso se utilizó para todos los edificios e instalaciones para crear un único modelo que los contuviera a todos, dando el resultado mostrado en la figura 1.2.



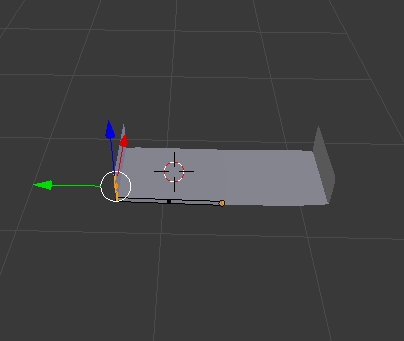
**Figura 1.2**

Después de tener el modelo completo, se diseñaron texturas para algunas zonas de este, solo para mejorar su efecto visual en juego, tal y como se muestra en la figura 1.3.

**/\* Edgar mete imagen de las texturas \*/**

## Modelado de pistas

Para diseñar las pistas se diseñó un fragmento únicamente de la pista apreciable en la figura 1.4.



**Figura 1.4**

Creado el fragmento, se duplicó varias veces a lo largo de una curva con el diseño del circuito para darle la forma deseada y se utilizó el mismo proceso para el diseño de las demás pistas tal y como se muestra en la figura 1.5.



**Figura 1.5**

## Modelado de coche y otros objetos

Para el coche y otros objetos dentro del juego, se optó por buscar los diseños en internet y facilitar un poco la tarea de modificación o adaptación de estos al juego.

Los distintos modelos utilizados son diseños gratuitos y libres.

# Capítulo 2: Diseño de escenas

## Menú principal y vistas

El menú principal, de opciones y de ajuste grafico se diseñaron usando canvas proporcionados por el propio editor de Unity, véase figura 2.1.



**Figura 2.1**

Cada uno de los botones maneja funciones similares cuando son presionados a excepción de los que cargan las escenas del juego y los de ajuste grafico, estos fueron programados definiendo una nueva función que cargara una escena específica, tal y como se aprecia a continuación.

public function Boton() {  
    Application.LoadLevel("Escena a cargar");  
}

Para los botones de ajuste de gráficos se utiliza un código diferente.

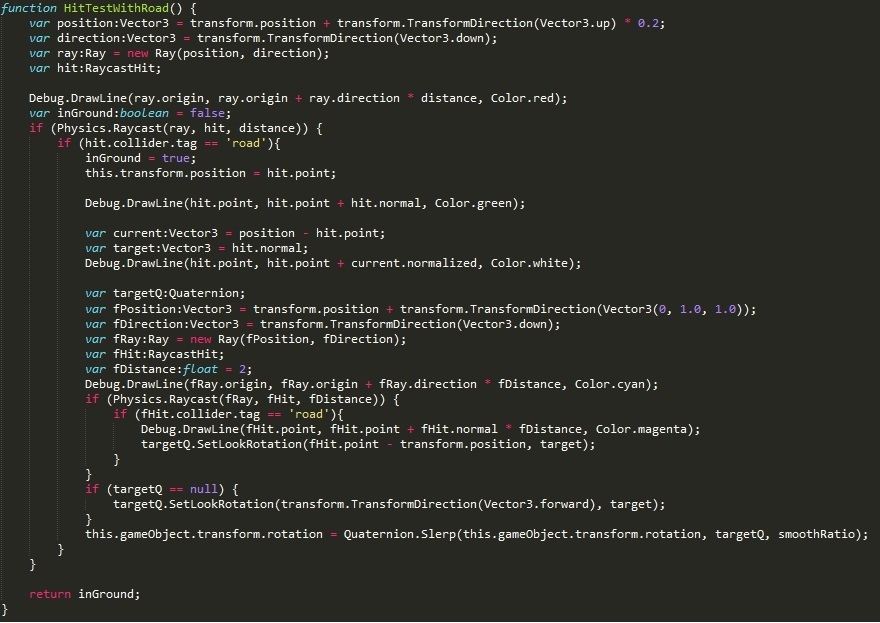
public function GraficosBuenos() {  
    QualitySettings.currentLevel = QualityLevel.Good;  
}

Esta función ajusta el valor de los gráficos en uno ya predefinido por Unity como “bueno”, de manera similar las demás funciones ajustan a distintos valores ya definidos para lograr otro efecto en los gráficos del juego;

# Capítulo 3: Programación

## Programación movimiento del carro

El movimiento del carro se basa en que el objeto que se maneja se adhiera automáticamente a la malla más cercana cuya etiqueta sea la correcta.



**Figura 3.1**

En el código anterior (figura 3.1) se puede ver la función principal del movimiento, funciona buscando un objeto con el cual colisionar que use la etiqueta “road” y respectivamente obtiene la posición del objeto en el cual se colocó el script para hacer que se acerque y se mantenga en la pista.

Para obtener los movimientos del jugador se toman las teclas definidas como “Vertical” y “Horizontal”, estas ajustan la velocidad y la torsión del carro, tal y como se muestra en el siguiente código.

var x:Number = Input.GetAxis("Horizontal");

var y:Number = Input.GetAxis("Vertical");

this.GetComponent.<Rigidbody>().velocity += y \* transform.forward \* forwardSpeed;

this.GetComponent.<Rigidbody>().AddTorque(transform.up \* x \* steerAngle, ForceMode.Acceleration);