



UNIVERSIDAD DE CHILE  
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS  
ESCUELA DE INGENIERÍA Y CIENCIAS  
CC3501 MODELACIÓN Y COMPUTACIÓN GRÁFICA PARA INGENIEROS

## TAREA 3B

### Road Star

Daniela Campos  
danielacamposfischer@gmail.com

Profesora:  
Nancy Hitschfeld K.

Auxiliares:  
Pablo Pizarro R.  
Pablo Polanco

Fecha:  
28 de Junio del 2017

## Índice

<b>1</b>	<b>Descripción del Problema</b>	<b>2</b>
<b>2</b>	<b>Descripción de la Solución</b>	<b>3</b>
2.1	Pistas . . . . .	3
2.1.1	Construcción . . . . .	3
2.1.2	Componentes del las Pistas . . . . .	3
2.2	Vehículo . . . . .	4
2.2.1	Modelo . . . . .	4
2.2.2	Componentes . . . . .	4
2.2.3	Cámara . . . . .	5
2.3	Menú . . . . .	5
2.3.1	Menu Principal . . . . .	5
2.3.2	Menu de Selección de Pista . . . . .	6
2.4	Otros . . . . .	7
2.4.1	Pausa . . . . .	7
2.4.2	Volver al Menú . . . . .	7
<b>3</b>	<b>Dificultades Encontradas</b>	<b>8</b>
<b>4</b>	<b>Conclusion</b>	<b>9</b>
<b>5</b>	<b>Anexos</b>	<b>10</b>

---

## 1 Descripción del Problema

Se debe diseñar e implementar un juego similar a RoadStar, el cual consiste en que el usuario (vehículo) recorre una pista cíclica hasta caerse de esta. El juego deberá poseer las siguientes características:

1. El vehículo siempre avanza automáticamente y el usuario tiene control sólo sobre el giro.
2. Después de girar se activa un turbo que aumenta la velocidad por un momento.
3. En cada vuelta la velocidad del vehículo debe aumentar.

El viraje del vehículo es controlado por las Flechas Izquierda y Derecha. Deben existir tres pistas las cuales deben contener segmentos rectos, dos virajes y dos segmentos con diferencia de altura. Al ser apretados los botones P y M el juego se debe pausar e ir al menú principal respectivamente. El menú debe tener dos secciones, un menú del juego y uno de selección de pistas. El menú del juego debe tener un botón para ir al menú de selección de pistas y un botón para salir del juego, por otro lado, el menú de selección de pistas debe incluir, una lista de las pistas a elegir, un botón para elegir cada pista y un botón para volver al menú principal.

El vehículo deberá mostrar animación para el derrape y para el turbo y al girar la cámara deberá mostrar el lado para el cual se está girando.

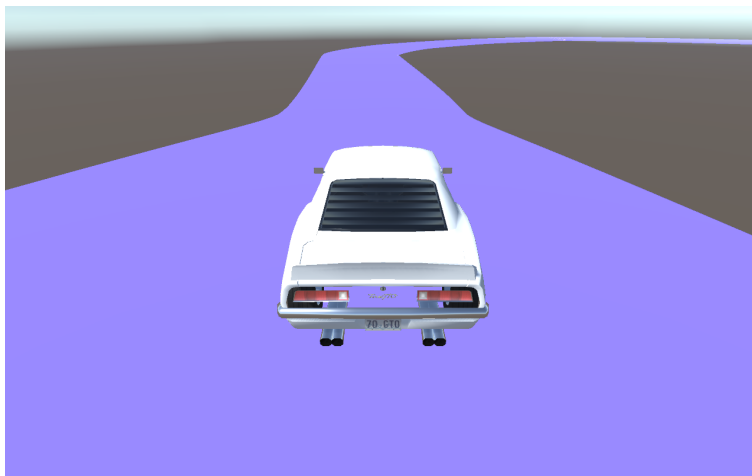


Figura 1: Ejemplo del Juego

---

## 2 Descripción de la Solución

A continuación se expondrá como se implementó el juego solicitado y los métodos y clases utilizados para hacerlo.

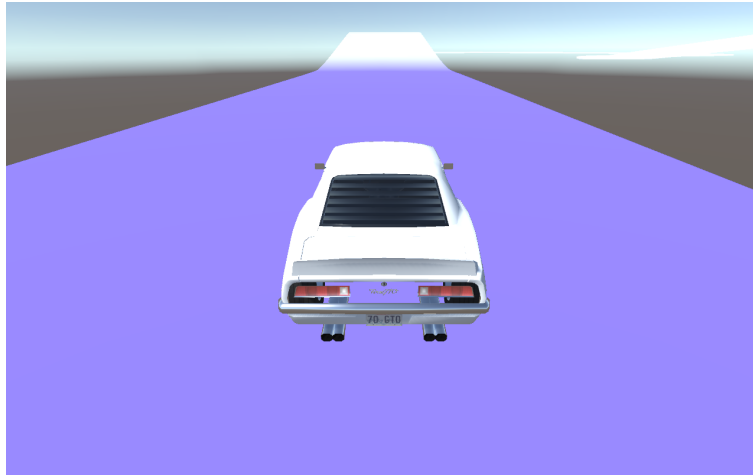


Figura 2: Juego en curso

### 2.1 Pistas

#### 2.1.1 Construcción

La modelación de las pistas fue realizada utilizando el programa **Blender** y luego fueron exportadas como archivo *.3ds* a la carpeta de Assets del Proyecto del juego.

#### 2.1.2 Componentes del las Pistas

Una vez en Unity las pistas son incluidas en la escena y poseen 3 elementos:

1. Curva: Aparte de los componentes que vienen con la curva, fue agregado un Mesh Collider, el cuál evita que el vehículo atravesase la pista.
2. Pista Física: Es lo que permite que la pista posea forma y no sea sólo una curva. No fue necesario agregarle otro componente.
3. Choque: Es un objeto vacío que contiene un Box Collider, el cuál permite detectar si es que el vehículo a dado una vuelta a la Pista, esto se logra gracias a la clase AumentarVelocidad, la cual será explicada a continuación.

- AumentarVelocidad: Tiene la función Start , la cual define el número de vueltas en un inicio como 0, y la función OnTriggerEnter, que toma como argumento el Objeto Vacío Choque, de tal manera de que si este es "activado" se aumenta en uno el valor de las vueltas.<sup>1</sup>

## 2.2 Vehículo

### 2.2.1 Modelo

El modelo del vehículo fue obtenido de la Librería Cars, la cuál se encuentra en el Asset Store de Unity.

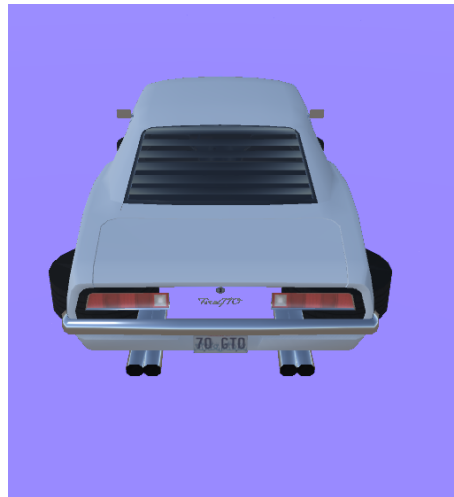


Figura 3: Modelo del Auto

### 2.2.2 Componentes

Los componentes que fueron incluidos al vehículo fueron: Rigidbody, para poder trabajar con el vehículo como un objeto físico, Box Collider, para modelar el movimiento del auto como si fuera una caja. Los Scripts que fueron incluidos serán explicados a continuación:

1. HoverMotor: Clase que permite el movimiento del vehículo. Contiene a los métodos: Awake, que permite trabajar con el Rigidbody, Update, que toma el input del usuario y con el rota el vehículo según corresponda y FixedUpdate, que le aplica las fuerzas necesarias al vehículo para que se mueva hacia adelante.<sup>2</sup>

---

<sup>1</sup>El código para la Clase Aumentar Velocidad puede ser encontrado en el Anexo 1

<sup>2</sup>El código para la Clase HoverMotor puede ser encontrado en el Anexo 2

2. Turbo: Esta Clase contiene las instrucciones para ejecutar el turbo que ocurre cuando el vehículo gira. Contiene los métodos: Start, que inicializa la cantidad de tiempo que se ha estado doblando en 0 y Update, que actualiza el tiempo que se ha estado doblando y en el caso de que esta sea mayor a 0.5 segundos actualiza la variable turbo a 2.<sup>3</sup>
3. Partículas2: Esta Clase hace aparecer las partículas cuando es necesario para el turbo. Contiene los métodos: Start, que inicializa en 0 el tiempo que ha sido ejecutado el Turbo y desactiva el sistema de partículas, y Update que activa el el sistema de partículas cuando es necesario y lo desactiva una vez que hayan pasado 5 segundos.<sup>4</sup>

### 2.2.3 Cámara

La cámara está anclada al vehículo y por lo tanto su posición es está condicionada por la posición del vehículo, de esta manera tenemos una vista de "primera persona". Le fue agregado un complemento de AudioSource, el cuál permite incluirle música al juego.

## 2.3 Menú

El menú fue creado utilizando la Librería Sample UI disponible en el Asset Store de Unity y se ocupó un Canvas que contenía tanto el Menu Principal como el Menú de Selección de Pistas

### 2.3.1 Menu Principal

El menu principal contiene dos botones, los cuales serán presentados a continuación:

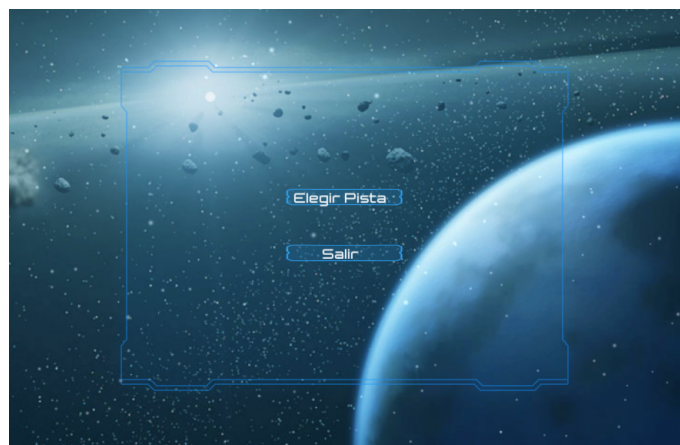


Figura 4: Menu Principal

---

<sup>3</sup>El código para la Clase Turbo puede ser encontrado en el Anexo 3

<sup>4</sup>El código para la Clase Particulas2 puede ser encontrado en el Anexo 4

1. Botón Elegir Pista: Este botón al ser seleccionado lleva al Menú de Selección de Pistas, esto se hace utilizando la instancia `OnClick` que tienen los botones, la cual al ser presionado el botón, desactiva la vista del Menú Principal y activa la del Menu de Selección de Pistas.
2. Boton Salida: Este botón le permite al jugador salir del juego, esto se lleva a cabo agregandole como componente al botón el script `QuitOnClick` el cuál tiene una única función `Quit`, que para el juego.<sup>5</sup>

### 2.3.2 Menu de Selección de Pista

El Menú de Selección de Pistas contiene 4 botones, de los cuales tres permiten seleccionar pistas y uno lleva al usuario de vuelta al Menú Principal.

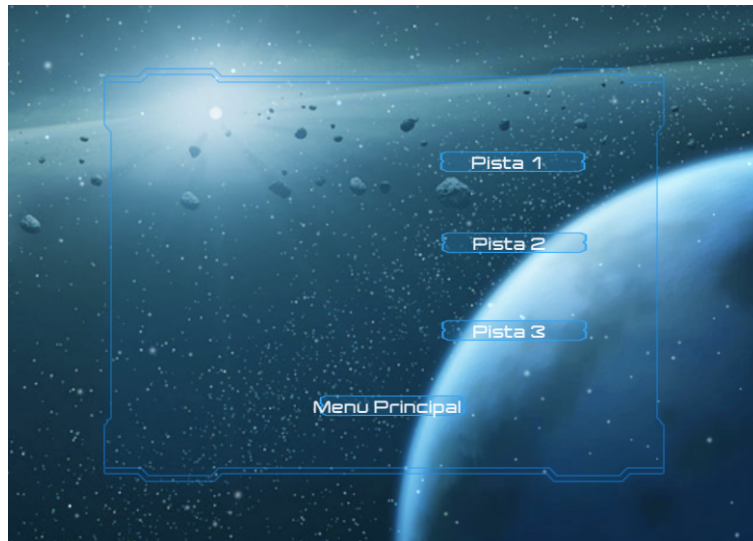


Figura 5: Menu de Selección de Pistas

1. Botón Pista i: Este botón permite cambiar la escena a la de la pista seleccionada. Esto se logra gracias al script `LoadSceneOnClick`, el cual es anclado al botón y posee un único método `LoadByIndex`, el cuál gracias al `SceneManager`, te lleva a la escena i.<sup>6</sup>
2. Botón Menú Principal: Este botón lleva al usuario de vuelta al Menú Principal y lo hace utilizando la instancia `OnClick` disponible en los botones, la cual activa el Menú Principal y desactiva el actual.

---

<sup>5</sup>El código para la Clase `QuitOnClick` se encuentra en el Anexo 5

<sup>6</sup>El código para la Clase `LoadSceneOnClick` se encuentra en el Anexo 6

## 2.4 Otros

A continuación se explicará como se implementaron las pausas y vueltas al Menú disponibles en el juego. Ambas instancias fueron implementadas agregando un Objeto Vacío a las escenas y anclandole los scripts pertinentes.

### 2.4.1 Pausa

Una vez en el juego, al apretar el botón P, este se pausará, esto se implementó con la Clase PauseScript, la cuál tiene un único método Update, que identifica si es que está siendo presionada la tecla P, de ser así pausa el juego. Es importante mencionar, que si es que se pausa el juego en una escena y posteriormente se cambia de escena, la nueva escena también estará en pausa, dado que PauseScript altera la escala de tiempo del juego entero.<sup>7</sup>

### 2.4.2 Volver al Menú

Si se desea volver al menú una vez en el juego, se debe presionar la tecla M, esto se logra gracias a la Clase BackToMenu, la cuál con su método Update identifica si es que la tecla M está siendo presionada y de ser así lleva al usuario de vuelta al Menú Principal.<sup>8</sup>

---

<sup>7</sup>El código para la Clase Pause Script se encuentra en el Anexo 7

<sup>8</sup>El código para la Clase BackToMenu se encuentra en el Anexo 8



---

### 3 Dificultades Encontradas

La mayor dificultad a la hora de implementar el juego fue lograr un buen movimiento del auto, dado que, se buscaron varias configuraciones que permitieran un movimiento realista del vehículo, pero con ninguna se llegó a un resultado realmente satisfactorio. La configuración final elegida fué utilizando un Box Collider, pero debido a que la pista es curva en algunas ocasiones el auto se queda estancado y es necesario girarlo para que continúe con su movimiento.

Otra dificultad encontrada fue implementar el Turbo dado que no se tenía mucha información sobre como hacerlo, se logró aumentar la velocidad cuando el auto giraba por una cantidad considerable de tiempo, pero lamentablemente no fue posible agregar las partículas para darle un efecto visual al turbo.

Es importante recalcar que un obstaculo importante fue la cantidad de tiempo necesaria para aprender a hacer el juego, dado que hay que investigar mucho el funcionamiento de Unity para tener el nivel de conocimiento necesario para crear el juego.

---

## 4 Conclusion

Tras utilizar el juego creado, se puede concluir que Unity es una herramienta muy útil a la hora de crear juego 3D que requieren movimiento, ya que permite incorporarle propiedades físicas a los modelos presentes en el juego lo que hace que todo se vea bastante realista.

Existen muchas cosas en el juego que pueden ser mejoradas, como el turbo y lograr un movimiento del auto mas continuo, pero estas podrían ser resueltas estudiando y practicando más con Unity.

---

## 5 Anexos

### 1. Anexo 1 :

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

public class AumentarVelocidad : MonoBehaviour {
    public static float vueltas = 1f;
    void Start () {
        vueltas = 1;
    }
    void OnTriggerEnter(Collider Choque) {
        vueltas++;
    }
}
```

---

## 2. Anexo 2 :

```
using UnityEngine;
using System.Collections;

public class HoverMotor : MonoBehaviour {

    public float speed = 90f;
    public float turnSpeed = 5f;
    private float powerInput;
    private float turnInput;
    private Rigidbody carRigidbody;

    void Awake ()
    {
        carRigidbody = GetComponent <Rigidbody>();
    }

    void Update ()
    {
        turnInput = Input.GetAxis ("Horizontal");
        transform.Rotate(new Vector3(0f, Time.deltaTime *
            turnInput * turnSpeed, 0f));
    }

    void FixedUpdate()
    {
        if (AumentarVelocidad.vueltas == 1f) {
            if (Turbo.turbo == 1f) {
                carRigidbody.AddRelativeForce (0f, 0f,
                    (speed * -1f));
            } else {
                carRigidbody.AddRelativeForce (0f, 0f,
                    (speed + speed*0.5f )*-1f);
            }
        } else {
            if (Turbo.turbo == 1f) {
                carRigidbody.AddRelativeForce (0f, 0f,
                    (speed +
                    AumentarVelocidad.vueltas
                    * 0.1f) * -1f);
            } else {
                carRigidbody.AddRelativeForce (0f, 0f,
                    (speed +
```

---

```
    AumentarVelocidad.vueltas
    * 0.1 f +speed*0.5 f )*-1 f );
}
}
}
```

---

### 3. Anexo 3 :

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

public class Turbo : MonoBehaviour {
    public static float tiempoDoblar = 0.0f;
    public static bool izquierda = false;
    public static bool derecha = false;
    public static float turbo = 1f;
    public static bool verdadTrubo = false;
    // Use this for initialization
    void Start () {
        tiempoDoblar = 0.0f;
    }

    // Update is called once per frame
    void Update () {
        if (turbo == 2f) {
            turbo = 1f;
        } else if (Input.GetKeyDown (KeyCode.LeftArrow) &&
            derecha == false) {
            izquierda = true;
            tiempoDoblar = tiempoDoblar + Time.deltaTime;
        } else if (Input.GetKeyUp (KeyCode.LeftArrow) &&
            tiempoDoblar > 0.5f && izquierda == true) {
            tiempoDoblar = 0.0f;
            turbo = 2f;
            izquierda = false;

        } else if (Input.GetKeyDown (KeyCode.RightArrow) &&
            izquierda == false) {
            derecha = true;
            tiempoDoblar = tiempoDoblar + Time.deltaTime;
        } else if (Input.GetKeyUp (KeyCode.RightArrow) &&
            tiempoDoblar > 0.5f && derecha == true) {
            tiempoDoblar = 0.0f;
            turbo = 2f;
            derecha = false;
        }
    }
}
```

---

#### 4. Anexo 4 :

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

public class Particulas2 : MonoBehaviour {

    public static float tiempoTurbo = 0.0f;
    public GameObject bla;
    void Start ()
    {
        tiempoTurbo = 0.0f;
        bla.SetActive(false);
    }

    // Update is called once per frame
    void Update () {
        if (Turbo.verdadTrubo == true && tiempoTurbo < 5f) {
            tiempoTurbo = tiempoTurbo + Time.deltaTime;
            bla.SetActive(true);
        } else if (tiempoTurbo > 5f) {
            Turbo.verdadTrubo = false;
            tiempoTurbo = 0f;
            bla.SetActive(true);
        }
    }
}
```

---

5. **Anexo 5 :**

```
using UnityEngine;
using System.Collections;

public class QuitOnClick : MonoBehaviour {

    public void Quit()
    {
        #if UNITY_EDITOR
        UnityEditor.EditorApplication.isPlaying = false;
        #else
        Application.Quit ();
        #endif
    }
}
```

6. **Anexo 6 :**

```
using UnityEngine;
using System.Collections;
using UnityEngine.SceneManagement;

public class LoadSceneOnClick : MonoBehaviour {

    public void LoadByIndex(int sceneIndex)
    {
        SceneManager.LoadScene (sceneIndex);
    }
}
```



---

## 7. Anexo 7 :

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

public class PauseScript : MonoBehaviour {

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

        if (Input.GetKey (KeyCode.P)) {
            if (Time.timeScale == 1)
                Time.timeScale = 0;
            else
                Time.timeScale = 1;
        }

    }

}
```

## 8. Anexo 8 :

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

public class BackToMenu : MonoBehaviour {

    // Update is called once per frame
    void Update () {
        if (Input.GetKey (KeyCode.M))
            SceneManager.LoadScene (0);
    }

}
```