



Logos del juego

|  |  |
| --- | --- |
| *Profesor:* | M.I. Marco Antonio Martínez Quintana |
| *Asignatura:* | estructuras de datos y algoritmos |
| *Grupo:* | 17 |
| *Nombre:* | Villanueva Corona Miguel Ángel |
| *Nombre del Proyecto:* | Super Speed Riders |
| *No. de Lista o Brigada:* | 41 |
| *Semestre:* | 2020-2 |
| *Fecha de entrega:* | 24 de marzo del 2020 |

**Objetivo**

El objetivo de este proyecto es crear un videojuego que sirva para el entretenimiento de la gente el cual consiste en la generación de mundo relativamente infinito a través de estructuras de datos.

**Alcance**

El alcance de este proyecto es la generación de un mundo lineal e infinito el cual sea recorrido por el usuario evadiendo obstáculos para lograr una mayor puntuación.

**Introducción**

El juego se trata de un coche el cual es manejado por el usuario a través de una calle lineal la cual se va generando conforme se vaya avanzando en el camino, el usuario tiene que ir esquivando los diferentes tipos de obstáculos los cuales le van a hacer perder vidas, cabe resaltar que el puntaje a obtener se va a basar en que el usuario avance lo más que pueda en el escenario.

El juego está planeado para una edad de 10 años en adelante ya que se pretende que el juego no sea ni muy rápido ni lento y no tan fácil, pero a la vez no tan difícil por eso esta predestinado para una audiencia mayor o igual a los 10 años

**Desarrollo**

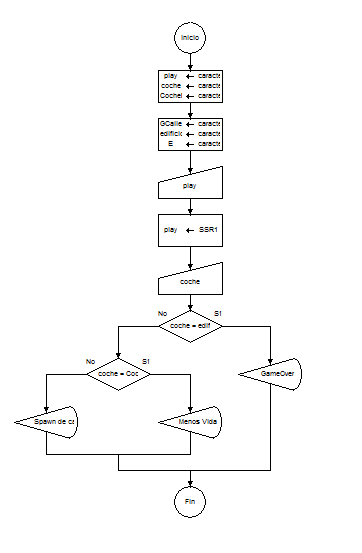
La realización de este proyecto se basa en una estructura de dato llamada pila la con la cual se utiliza en este proyecto para la creación de autopistas al encolar una calle después de otra calle y desencolar la calle que va delante de esta. Para la realización de esta dinámica en el juego se utilizo el software de Unity, además de que se utilizó Visual Studio para hacer la programación y Paint y Photoshop para el diseño de las calles.

Algoritmo

1. Dar cinco caracteres de tipo cadena el primero indica el botón y se almacena en Play, el segundo indica al coche con el que se choca y se almacena en CocheM, otra que indica a los edificios del entorno con el cual puede colisionar y se almacena en edificio, el cuarto indica la generación de calles y se almacena en GCalle y el ultimo indica a el coche y se almacenara en coche
2. ----
3. Dos caracteres de tipo cadena el primero indica el cambio de escena y se almacena en SSR1 y uno que represente la perdida de vida el cual es representado por MenosV Y un carácter de tipo cadena que indique el fin del juego almacenado en GameOver.
4. 1. Dar clic en el botón Play
      1. Realizar la transición a la escena SSR1
         1. Si coche=edificio
            1. Mostrar GameOver
         2. En caso contrario
            1. Si coche=CocheM

Mostrar MenosV

* + - * 1. En caso contrario

Realizar GCalles

Códigos básicos para el funcionamiento

Motor de calles

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class MotorCalles : MonoBehaviour

{

    public GameObject motorCalles;

    public GameObject[] contenedorCalles;

    public float speed;

    public int numSelectorCalle;

    public int contadorCalles = 0;

    public bool cuentaRegresivaT;

    public bool juegoT;

    void Start()

    {

        juegoT = false;

        InicioJuego();

    }

    public void InicioJuego()

    {

        CreaCalles();

        SpeedCarretera();

        cuentaRegresivaT = false;

    }

    void Update()

    {

        if(cuentaRegresivaT && juegoT == false)

        {

            motorCalles.transform.Translate(Vector3.down \* speed \* Time.deltaTime);

        }

    }

    public void CreaCalles()

    {

        numSelectorCalle = Random.Range(0,5);

        GameObject Motor = (GameObject)Instantiate(contenedorCalles[numSelectorCalle],

                                                    new Vector3(-1,500,100),

                                                    transform.rotation);

        Motor.SetActive(true);

        contadorCalles ++;

        Motor.name = "Motor"+contadorCalles;

        Motor.transform.parent = motorCalles.transform;

        GameObject piezaAux = GameObject.Find("Motor"+(contadorCalles-1));

        Motor.transform.position = new Vector3(transform.position.x,

                                                 piezaAux.GetComponent<Renderer>().bounds.size.y +

                                                 piezaAux.transform.position.y,

                                                 piezaAux.transform.position.z);

    }

    public void SpeedStop()

    {

        speed = 0;

    }

    public void SpeedArcen()

    {

        speed = 5;

    }

    public void SpeedCarretera()

    {

        speed = 15;

    }

    public void SpeedCocheMalo1()

    {

        speed = 3;

    }

    public void FinalizarJuego()

    {

        SpeedStop();

    }

}

Limite de calles

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class limiteCarreteras : MonoBehaviour

{

    public MotorCalles motorCallesScript;

    public void OnTriggerEnter2D ( Collider2D cInfo)

    {

        if(cInfo.gameObject.tag == "limite")

        {

            Destroy(cInfo.transform.parent.gameObject);

            motorCallesScript.CreaCalles();

        }

    }

}

Cronometro

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.UI;

public class cronometro : MonoBehaviour

{

    public GameObject motorCalles;

    public MotorCalles MotorCallesScript;

    public Text textoTiempo;

    public Text textoMetros;

    public float distancia;

    public float tiempo;

    public bool CronometroEncendido;

    public Image masTiempo;

    public Image masDinero;

    public GameObject popfinalGO;

    public POPfinal POPfinalScript;

    void Start()

    {

        textoTiempo.text = "0:00";

        textoMetros.text = "0";

        masTiempo.CrossFadeAlpha(0,0,false);

    }

Cuenta atrás

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class CuentaAtras : MonoBehaviour

{

    public GameObject motorCalles;

    public GameObject musicaJuego;

    public GameObject sonidoStart;

    public GameObject sonidoMotorCoche;

    public GameObject numeros;

    public Sprite[] numerosIMG;

    public MotorCalles MotorCallesScript;

    public cronometro cronometroScript;

    float tiempoDeEspera = 4;

    public bool pararCuenta = false;

    public bool finContador = false;

    public AudioSource reproductor;

    public AudioClip[] sonidosContador;

    void Start ()

    {

        reproductor = this.gameObject.GetComponent<AudioSource>();

        IniciarCuentaAtras();

    }

    public void IniciarCuentaAtras()

    {

        StartCoroutine(EmpezarCuentaAtras());

    }

    public IEnumerator EmpezarCuentaAtras()

    {

        sonidoStart.GetComponent<AudioSource>().Play();

        yield return new WaitForSeconds(1);

        InvokeRepeating("Contando",1.0f,1.00f);

    }

    void Contando ()

    {

        tiempoDeEspera --;

        if(tiempoDeEspera >= 3)

        {

            numeros.GetComponent<SpriteRenderer>().sprite = numerosIMG[3];

            reproductor.clip = sonidosContador[0];

            reproductor.Play();

        }

        if(tiempoDeEspera <= 3 && tiempoDeEspera >=2)

        {

            numeros.GetComponent<SpriteRenderer>().sprite = numerosIMG[2];

            reproductor.clip = sonidosContador[0];

            reproductor.Play();

        }

        if(tiempoDeEspera <= 2 && tiempoDeEspera >=1)

        {

            numeros.GetComponent<SpriteRenderer>().sprite = numerosIMG[1];

            reproductor.clip = sonidosContador[0];

            reproductor.Play();

        }

        if(tiempoDeEspera <= 1 && finContador == false)

        {

            finContador = true;

            cronometroScript.CronometroEncendido = true;

            MotorCallesScript.cuentaRegresivaT = true;

            numeros.GetComponent<SpriteRenderer>().sprite = numerosIMG[0];

            reproductor.clip = sonidosContador[1];

            reproductor.Play();

            musicaJuego.GetComponent<AudioSource>().Play();

            sonidoMotorCoche.GetComponent<AudioSource>().Play();

        }

        if(tiempoDeEspera <= 0)

        {

            CancelInvoke();

        }

    }

    void Update()

    {

        if(tiempoDeEspera == 0 && pararCuenta == false)

        {

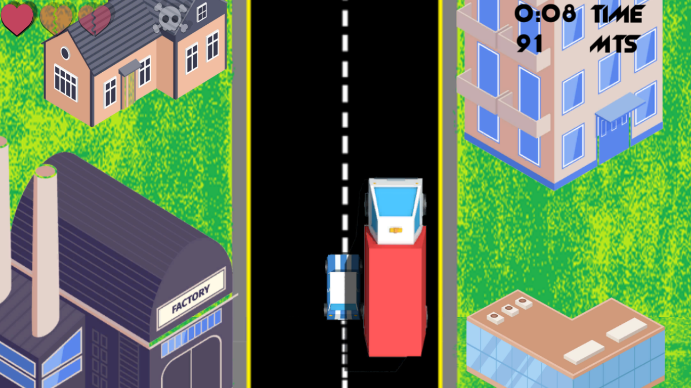
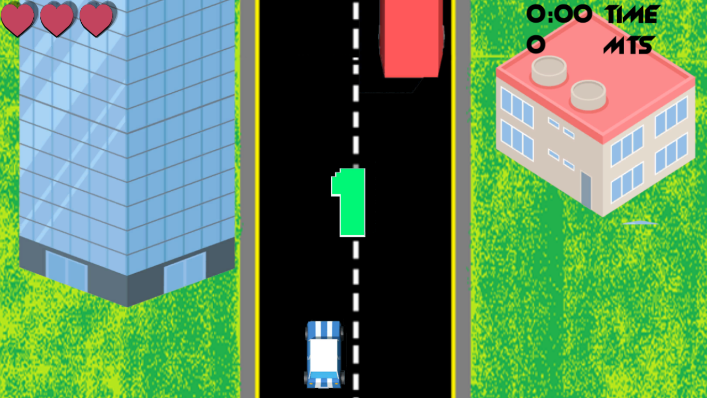
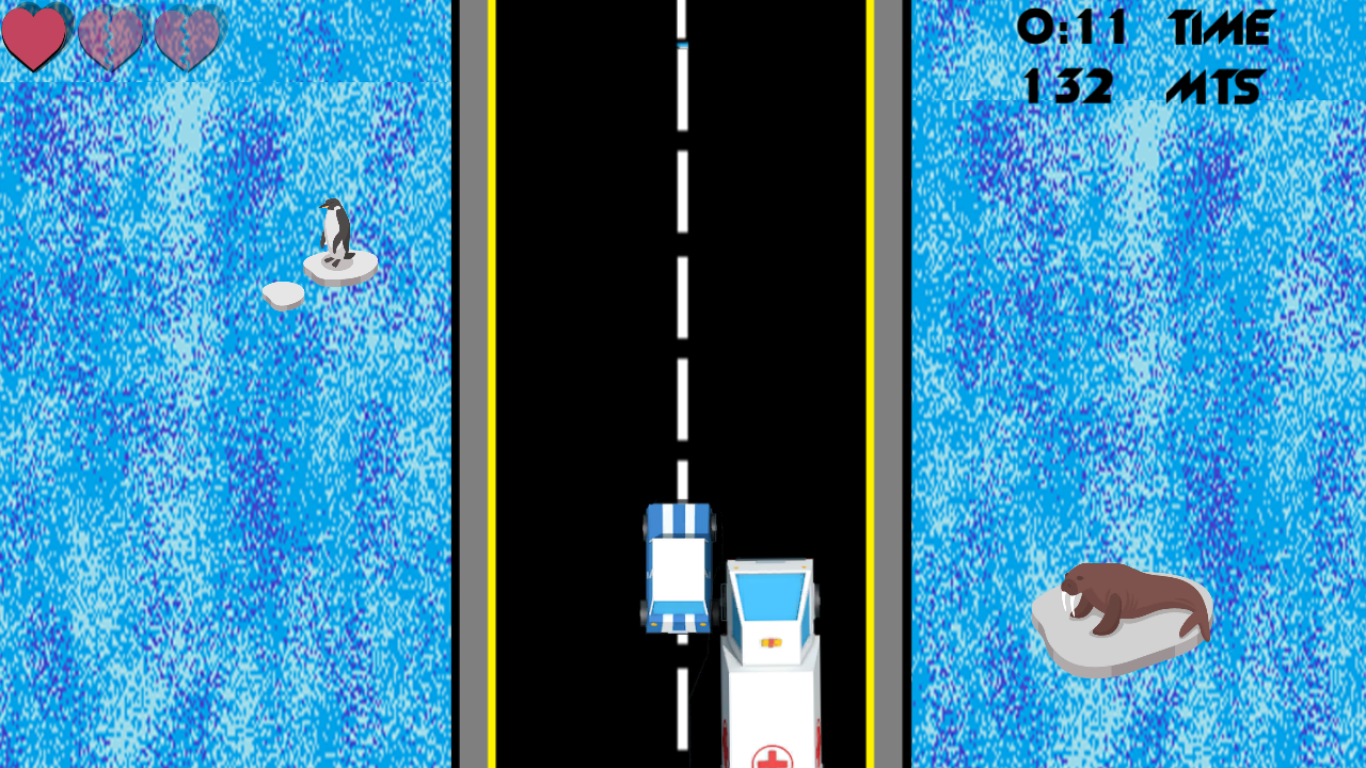
            pararCuenta = true;

            numeros.SetActive(false);

        }

    }

}

**********Resultados**

Video de ejecución del juego

<https://www.youtube.com/watch?v=Sb9D0P7ts-0>

El audio no se reproduce debido al software de gravado utilizado

**Conclusión**

Aunque por el momento el juego se encuentra en una fase de Beta se espera modificar más el juego para poder hacer del algo un poco más dinámico ya sea el respawn independiente de obstáculos, así como la implementación de más escenarios por el cual el usuario puede pasar, además, se desea hacer la implementación de PowerUps los cuales ayuden al usuario a mejorar su puntaje o sobrevivir más tiempo en el juego, además de introducir un menú de selección de coche así como un menú de selección de modos de juego.

La conclusión es que por el momento el juego no se piensa ser distribuido con la gente ya que todavía se pueden y se deben de detallar algunas cuestiones del juego además de que sea implementar otros tipos de dinámicas que hagan de este un juego bueno

Ya por ultimo quiero mencionar que este es el primer juego que realizo en unity y realmente fue un reto ya que anteriormente yo solo tenia un poco de noción de como programar en unity y cabe resaltar que, aunque sea mi primera vez haciendo un juego desde cero me encanto hacer este tipo de cosas las cuales si te estancas pueden ser un poco frustrante pero cuando realmente te salen las cosas como tu lo esperas es una experiencia muy buena.

**Referencias**

<https://www.youtube.com/watch?v=SFjtqrDdu7I>

<https://www.domestika.org/es/courses/86-diseno-y-programacion-de-videojuegos-con-unity-5/units/376-publicacion-del-videojuego-en-pc-mac-osx-y-paginas-web>

<https://incompetech.com/music/royalty-free/music.html>

<https://www.sshhtt.com/categories/ambient>