



Gráficos y Multimedia

Sesión 3: Introducción a Unity (2ª Parte) Trabajo de Unity



Script PlayerController (C#) [1]

```
using UnityEngine;
using UnityEngine.UI;
using System.Collections;

public class PlayerController : MonoBehaviour {

    public float velocidad;
    public Text countText;
    public Text winText;

    private Rigidbody rb;
    private int contador;

    void Start()
    {
        rb = GetComponent<Rigidbody>();
        contador = 0;
        SetCountText();
        winText.text = "";
    }
}
```



Script PlayerController (C#) [2]

```
void FixedUpdate()
{
    float posH = Input.GetAxis("Horizontal");
    float posV = Input.GetAxis("Vertical");

    Vector3 movimiento = new Vector3(posH, 0.0f, posV);

    rb.AddForce(movimiento * velocidad);
}

void OnTriggerEnter(Collider other)
{
    if (other.gameObject.CompareTag("mono"))
    {
        other.gameObject.SetActive(false);
        contador = contador + 1;
        SetCountText();
    }
}
```



Script PlayerController (C#) [3]

```
void SetCountText()
{
    countText.text = "Contador: " + contador.ToString();
    if (contador >= 4)
    {
        winText.text = "Ganaste!!";
    }
}
}
```



Script CameraController (C#)

```
using UnityEngine;
using System.Collections;

public class CameraController : MonoBehaviour {

    public GameObject player;

    Private Vector3 desplazamiento;

    // Use this for initialization
    void Start () {
        desplazamiento = transform.position - player.transform.position;
    }

    void LateUpdate () {
        transform.position = player.transform.position + desplazamiento;
    }

}
```



Script MonkeyController (C#)

```
using UnityEngine;
using System.Collections;

public class MonkeyController : MonoBehaviour {

    // Update is called once per frame
    void Update () {
        transform.Rotate(new Vector3(0, 0, 10) * Time.deltaTime);
    }
}
```




Algoritmo de Laberintos Aleatorios

Algoritmo para generar laberintos aleatorios

- Características
 - Entrada y salida aleatoria
 - Laberinto siempre con solución
- Técnica (Algoritmo de Prim)
 - Avanzar dos posiciones
 - Emplear una pila para recorrer las casillas no visitadas
- Ejemplos:
 - <https://nopointer.wordpress.com/2015/10/04/generar-laberinto-aleatorio-con-el-algoritmo-de-prims/>



Trabajo en Unity

Minijuego en Unity

- Realizar un tablero de juego con agujeros, habrá objetos que se podrán recoger (**collectables**) y que deberán puntuar distinto, al menos tres distintos. Se puede emplear algún otro marcador o una posición determinada para indicar la llegada.
- El juego deberá contar con al menos 5 tableros diferentes donde se incrementará la dificultad de forma gradual.
- Se valorará especialmente el nivel jugabilidad y de acabado, así como, las texturas y sonidos empleados.
- Con los controles se moverá el tablero (rotación X y Z). Habrá una pelota que se moverá a medida que se incline el tablero. Habrá que llegar a la salida del tablero obteniendo el máximo de puntos.
- Para el movimiento del tablero, en un móvil o tablet, podremos hacer uso de los giroscopios.
- Si la pelota se cuelga por un agujero perdemos, podemos contar con una serie de vidas, podemos colocar un objeto para detectar la caída y terminar el juego.



Trabajo en Unity

- Para la entrega se deberá proporcionar el código fuente y un ejecutable para Windows.
- La entrega se realizará mediante Moodle, fecha tope **10 de Marzo de 2022**, inclusive.

Reto para M.H.

- Realizar tablero aleatorio empleando el algoritmo de Prim.
- Consultar:
 - <https://docs.unity3d.com/es/current/Manual/RandomNumbers.html>
 - <https://docs.unity3d.com/Manual/InstantiatingPrefabs.html>



¿Preguntas...?