



Curso desarrollo de videojuegos con Unity - Málaga

M.3011.001.021

API Unity. Scripting Básico

Violeta Sáez Botella



Cofinanciado por
la Unión Europea



Fondos Europeos



Sistema Nacional de
Garantía Juvenil



Índice

- _01 Translate y Rotate
- _02 LookAt
- _03 GetKey
- _04 GetAxis
- _05 Awake y Start
- _06 GetComponent
- _07 OnMouseDown
- _08 Activar/Desactivar GameObjects
- _09 Habilitar/Deshabilitar componentes
- _10 Instantiate
- _11 Destroy
- _12 Invoke
- _13 InvokeRepeating

— 01

Translate y Rotate

Cambiar posición y rotación de un GameObject.

Dirección * velocidad * Time.deltaTime

```
if (Input.GetKey(KeyCode.UpArrow))
    transform.Translate(Vector3.forward * moveSpeed * Time.deltaTime);

if (Input.GetKey(KeyCode.DownArrow))
    transform.Translate(-Vector3.forward * moveSpeed * Time.deltaTime);

if (Input.GetKey(KeyCode.LeftArrow))
    transform.Rotate(Vector3.up, -turnSpeed * Time.deltaTime);

if (Input.GetKey(KeyCode.RightArrow))
    transform.Rotate(Vector3.up, turnSpeed * Time.deltaTime);
```

— 02

LookAt

Se usa para hacer que la dirección **forward** de un **GameObject** (es decir, "hacia donde está mirando") apunte hacia otro **GameObject**.

```
using UnityEngine;
using System.Collections;

public class CameraLookAt : MonoBehaviour
{
    public Transform target;

    void Update ()
    {
        transform.LookAt(target);
    }
}
```

— 03

GetKey

```
④ Mensaje de Unity | Referencias
void Update()
{
    if (Input.GetKeyDown(KeyCode.A))
    {
        Debug.Log("He presionado la tecla A");
    }
    else if (Input.GetKey(KeyCode.A))
    {
        Debug.Log("Estoy presionando la tecla A");
    }
    if (Input.GetKeyUp(KeyCode.A))
    {
        Debug.Log("He dejado de presionar la tecla A");
    }
}
```

GetKey

Método que se usa para obtener los input de teclado y se accede a través de la clase **Input**.

DEVUELVE UN VALOR BOOLEANO.

— 04

GetAxis

Devuelve un valor tipo float entre -1 y 1.

ProjectSettings -> Input -> Axes

Cada axis tiene un nombre y lleva asociado una tecla positiva y negativa.

```
void Update()
{
    float h_ = Input.GetAxis("Horizontal");
}
```

El axis horizontal mapea las teclas AD. Si el usuario está pulsado la A devuelve -1, si no está pulsando nada devuelve 0 y si está pulsando D devuelve 1.

— 05

Awake y Start

```
using UnityEngine;
using System.Collections;

public class AwakeAndStart : MonoBehaviour
{
    void Awake ()
    {
        Debug.Log("Awake called.");
    }

    void Start ()
    {
        Debug.Log("Start called.");
    }
}
```

Awake y Start

Son dos funciones que se invocan automáticamente cuando se carga un script.

Awake se invoca antes.

— 06

GetComponent

Para acceder al componente de un **GameObject** usamos **GetComponent**.

```
Rigidbody rb;  
    Mensaje de Unity | 0 referencias  
void Awake()  
{  
    rb = GetComponent<Rigidbody>();  
}
```



```
Light _light;  
    Mensaje de Unity | 0 referencias  
void Awake()  
{  
    _light = GetComponent<Light>();  
}
```

— 07

OnMouseDown

Detecta el click del ratón sobre un objeto que tiene un collider.

```
➊ Mensaje de Unity | 0 referencias
private void OnMouseDown()
{
    Debug.Log("Has pulsado con el botón izquierdo del ratón.");
}

➋ Mensaje de Unity | 0 referencias
private void OnMouseUp()
{
    Debug.Log("Has soltado el botón izquierdo del ratón.");
}

➌ Mensaje de Unity | 0 referencias
private void OnMouseEnter()
{
    Debug.Log("Has pasado el ratón por encima del GameObject.");
}

➍ Mensaje de Unity | 0 referencias
private void OnMouseDrag()
{
    Debug.Log("Estás arrastrando el cursor del ratón.");
}
```

— 08

Activar/Desactivar GameObjects

Para activar o desactivar un **GameObject** en la escena.

```
public GameObject myObject;  
    ↗ Mensaje de Unity | 0 referencias  
void Update()  
{  
    myObject.SetActive(true);  
}
```



Activa el **GameObject** al que haga referencia la variable **myObject**.

Desactiva el **GameObject** que lleva el script "**ExampleScript**".



```
public class ExampleScript : MonoBehaviour  
{  
    ↗ Mensaje de Unity | 0 referencias  
    void Update()  
    {  
        gameObject.SetActive(false);  
    }  
}
```

Habilitar/Deshabilitar componentes

Podemos **habilitar** o
deshabilitar componentes
desde **código**.

```
Light _light;
➊ Mensaje de Unity | 0 referencias
void Awake()
{
    _light = GetComponent<Light>();

    _light.enabled = true; //Habilita el componente
    _light.enabled = false; //Deshabilita el componente
}
```

Habilitar/Deshabilitar Componentes

Destroy

Destroy

Es una función que usamos para eliminar GameObjects de la escena.



```
public GameObject myObject;  
    Mensaje de Unity | 0 referencias  
private void Update()  
{  
    if(Input.GetMouseButtonDown(0))  
    {  
        DestroyObject();  
    }  
}  
1 referencia  
void DestroyObject()  
{  
    Destroy(myObject);  
}
```

Podemos meterle un tiempo de espera, es decir, el objeto se destruirá a los 5 segundos



```
public GameObject myObject;  
    Mensaje de Unity | 0 referencias  
private void Update()  
{  
    if(Input.GetMouseButtonDown(0))  
    {  
        DestroyObject();  
    }  
}  
1 referencia  
void DestroyObject()  
{  
    Destroy(myObject, 5);  
}
```

Instantiate

Instantiate

Instantiate es una función que se usa para crear clones de prefabs.

```
public GameObject myObject;  
    Mensaje de Unity | 0 referencias  
private void Update()  
{  
    if(Input.GetMouseButtonDown(0))  
    {  
        CreateObjects();  
    }  
}  
1 referencia  
void CreateObjects()  
{  
    Instantiate(myObject);  
}
```

Cada vez que pulsamos el **botón izquierdo del ratón**, creamos un clone del prefab. Como no le decimos donde colocarlo, lo pondrá en el (0,0,0) de la escena.

Instantiate

Le indico la posición y la rotación de salida del clone mediante una variable pública tipo **Transform**.

```
public GameObject myObject;
public Transform posRotObject;
@ Mensaje de Unity | 0 referencias
private void Update()
{
    if(Input.GetMouseButtonDown(0))
    {
        CreateObjects();
    }
}
1 referencia
void CreateObjects()
{
    Instantiate(myObject, posRotObject.position, posRotObject.rotation);
}
```

Invoke

Invoke

Te permite meterle retardo de tiempo a la llamada de una función.

```
public GameObject target;
public Transform posRotTarget;
// Mensaje de Unity | 0 referencias
void Start()
{
    //La función SpawnObject se ejecutará cuando pasen 2 segundos
    Invoke("SpawnObject", 2);
}
0 referencias
void SpawnObject()
{
    Instantiate(target, posRotTarget.position, posRotTarget.rotation);
}
```

InvokeRepeating

InvokeRepeating

Permite llamar a una función cada X segundos.

```
public GameObject target;
public Transform posRotTarget;
↳ Mensaje de Unity | 0 referencias
void Start()
{
    //La función SpawnObject tardará en ejecutarse 1 segundo LA PRIMERA VEZ QUE
    //ES LLAMADA. Luego se ejecutará cada 3 segundos
    InvokeRepeating("SpawnObject", 1,3);
}
0 referencias
void SpawnObject()
{
    Instantiate(target, posRotTarget.position, posRotTarget.rotation);
}
```