



15 DECEMBER 2018

**GLOBAL AI&MR**  
**BOOTCAMP** MADRID  
NEW INTELLIGENT WORLDS **2018**

# Build Mixed Reality Apps

Marcos Rodríguez Calderón  
Mixed Reality Technical Leader

@MRCalderon3D

# GRACIAS A NUESTROS SPONSORS!!

**encamina**  
PIENSA EN COLORES

**Insight** 

**TOKIOTA**

**plain concepts** 

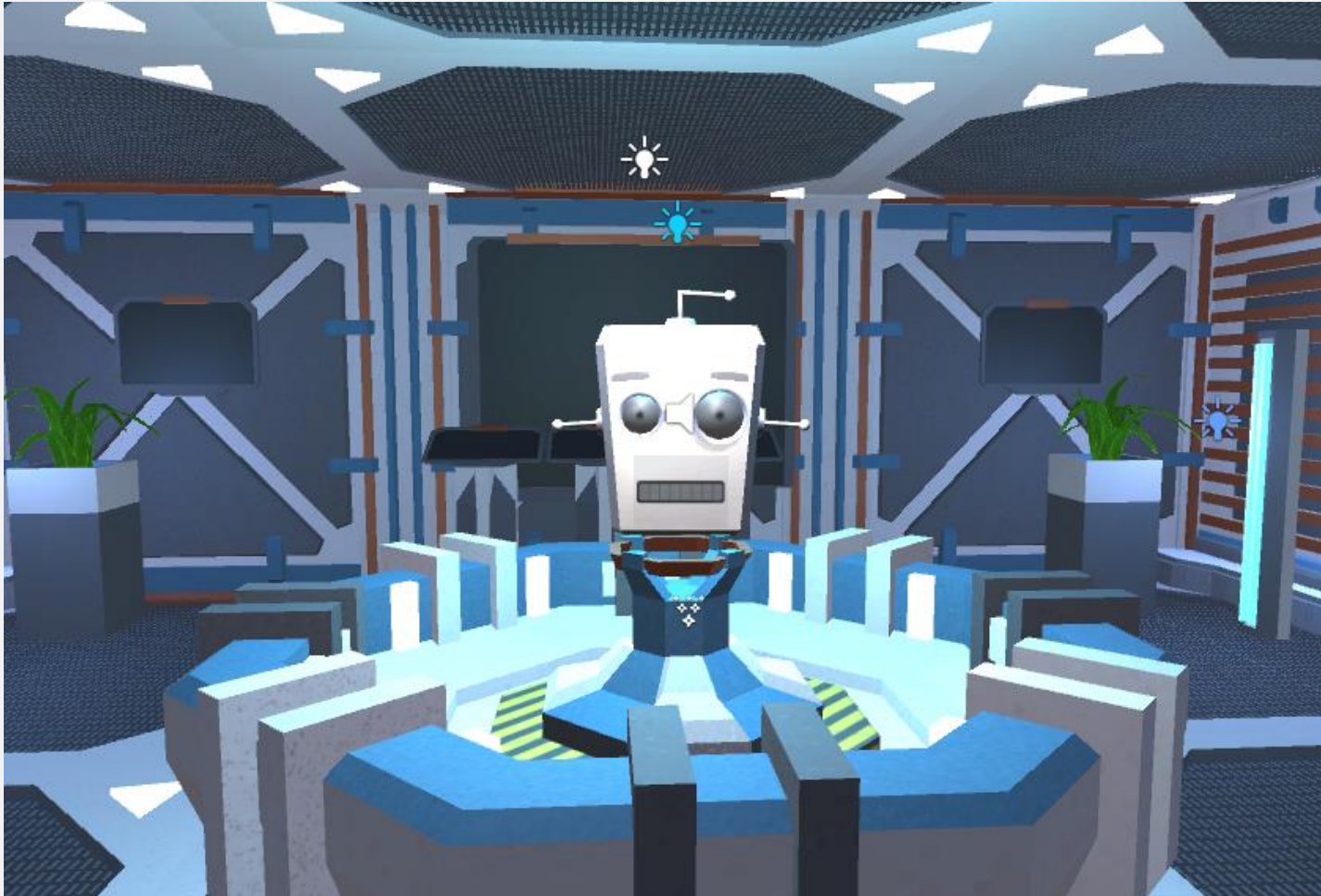
**everis**  
an NTT DATA Company



 **Microsoft**

# Vista previa de proyecto

Hacemos una ruta por el proyecto abriendo puertas y hablando con el robot.



# Set Up del proyecto

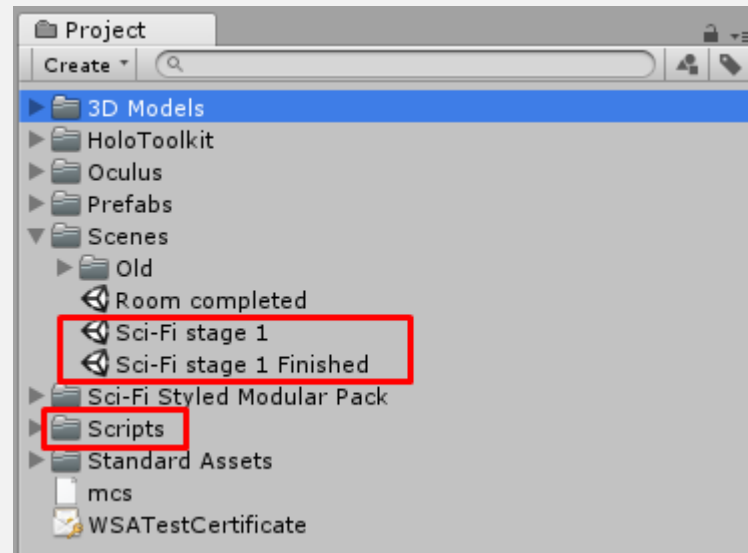
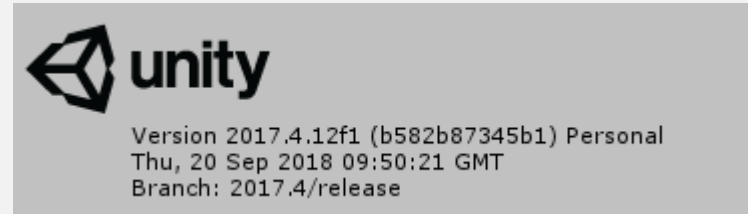
Versiones de software:

- Unity 3D 2017.4.12f1
- Visual Studio 2017

El proyecto tiene dos escenas; "**Sci-Fi stage 1 Finished**" que contiene la practica finalizada y "**Sci-Fi stage 1**" que está preparada para realizar el taller.

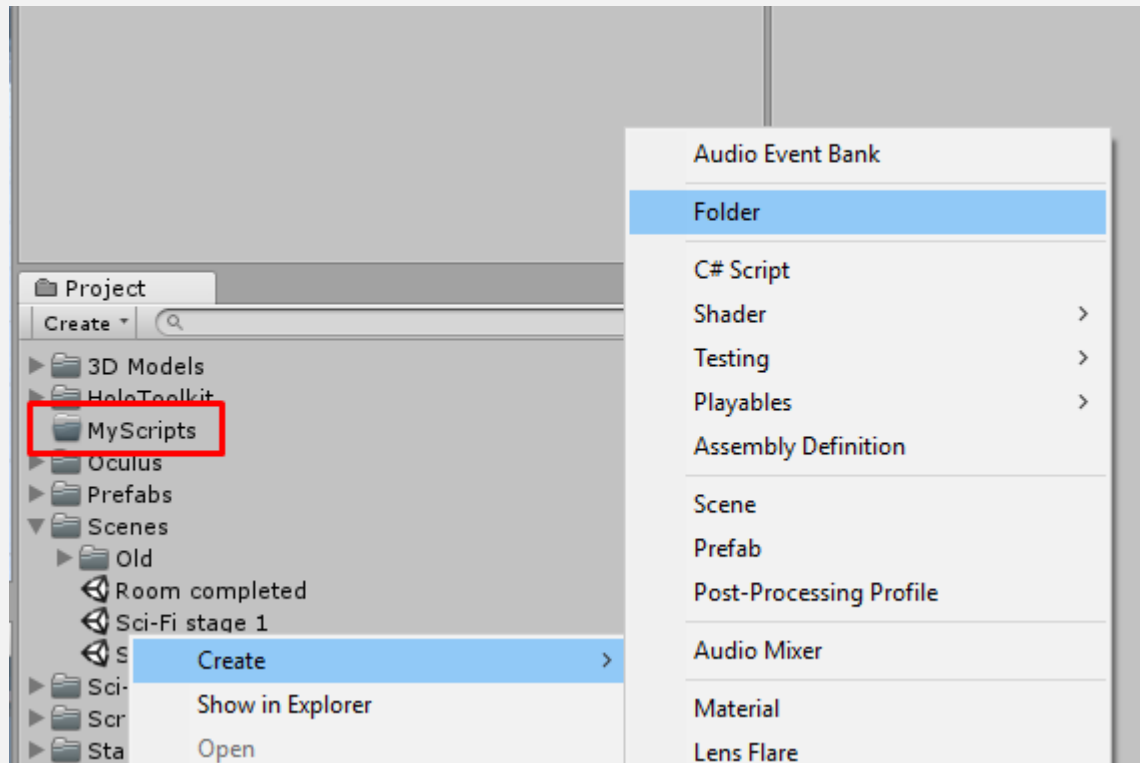
También hay una carpeta con los scripts necesarios para el taller.

Cada script nuevo se llamará "My[nombre del script]" que coincidirá con el script que ya existe dentro de la carpeta "Scripts"



# Set Up del proyecto

Nos creamos una carpeta que se llamara "MyScripts" para ir creando nuestros scripts.



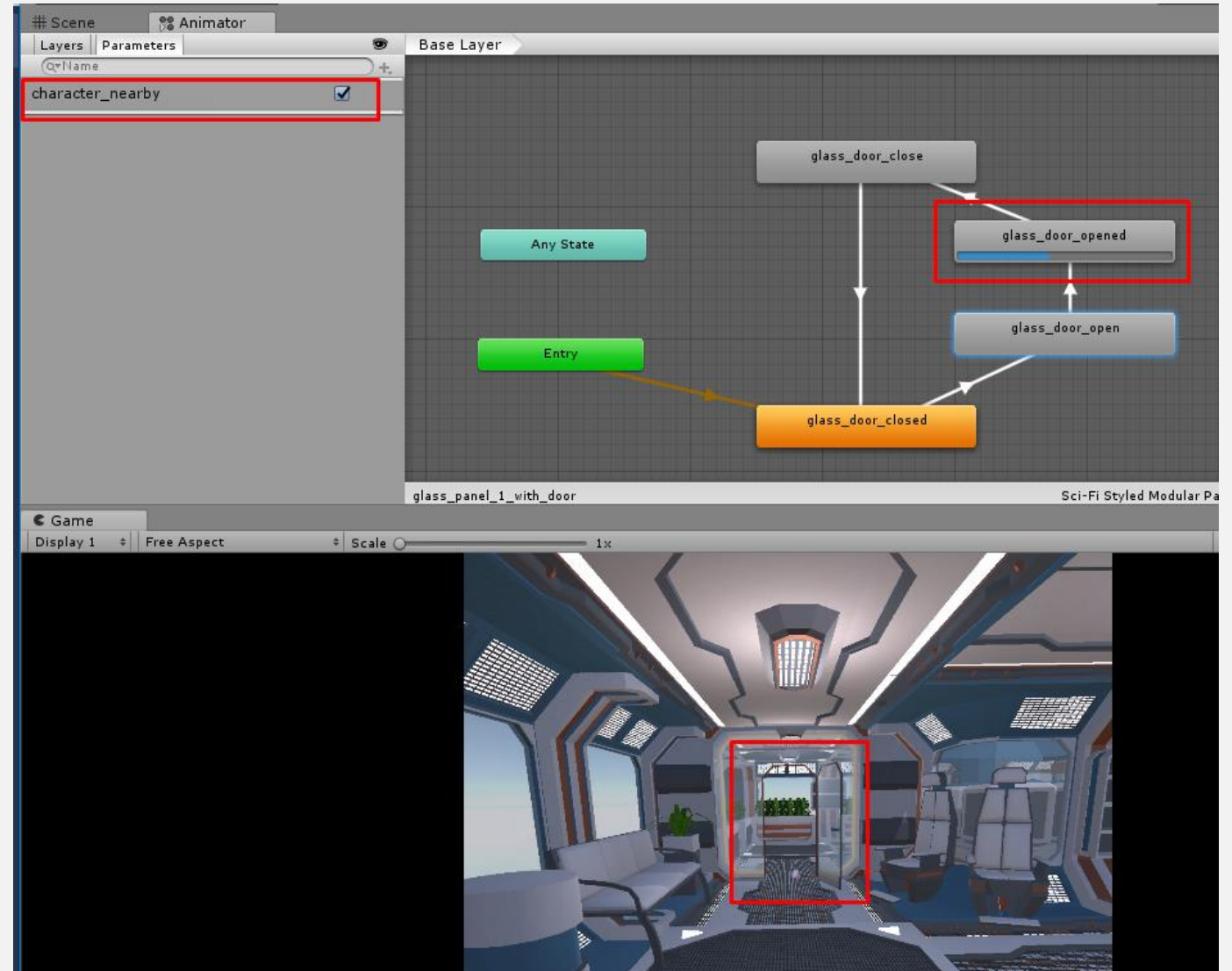
# Funcionalidad Puertas

Antes de darle funcionalidad a las puertas, vamos a echar un vistazo a las animaciones que tienen establecidas en el AnimatorController

**"glass\_panel\_1\_with\_door"**

Como vemos en la pestaña de parámetros, hay un parámetro Boolean que nos servirá para indicar cuando se abran las puertas.

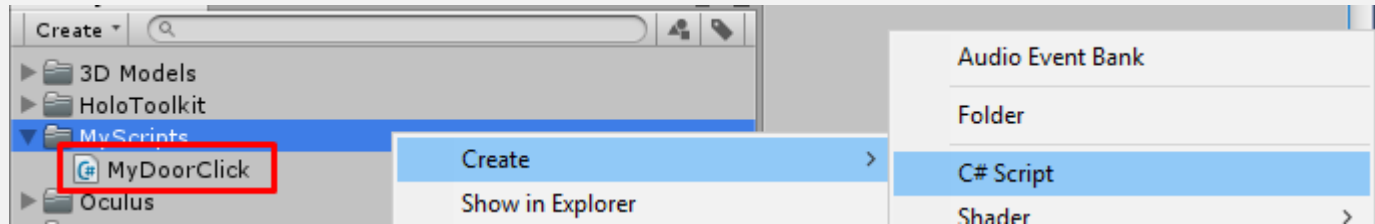
Si le damos a play y lo marcamos con el ratón, podemos ver como se abre la puerta y como avanzan los estados en la Ventana de Animator



# Funcionalidad Puertas

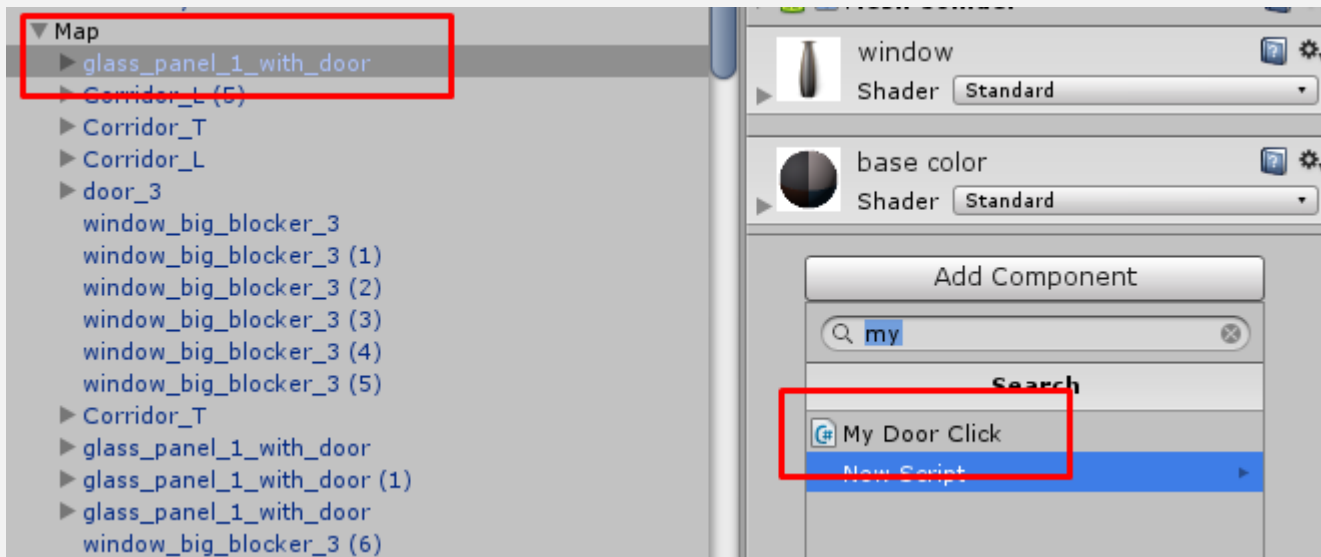
Creamos un nuevo Script para darle funcionalidad a las puertas.

Botón derecho sobre la carpeta **MyScripts** -> **Create** -> **C# Script** y lo nombramos "**MyDoorClick**"



# Funcionalidad Puertas

Añadimos el script "**MyDoorClick**" al GameObject "**glass\_panel\_1\_with\_door**" y abrimos el Script en Visual Studio para darle funcionalidad.





# Funcionalidad Puertas

```
using UnityEngine;
using Holotoolkit.Unity.InputModule;

public class MyDoorClick : MonoBehaviour, IInputClickHandler
{
    Animator m_Animator;
    void Start()
    {
        //Get the Animator attached to the GameObject you are intending to animate.
        m_Animator = gameObject.GetComponent<Animator>();
    }

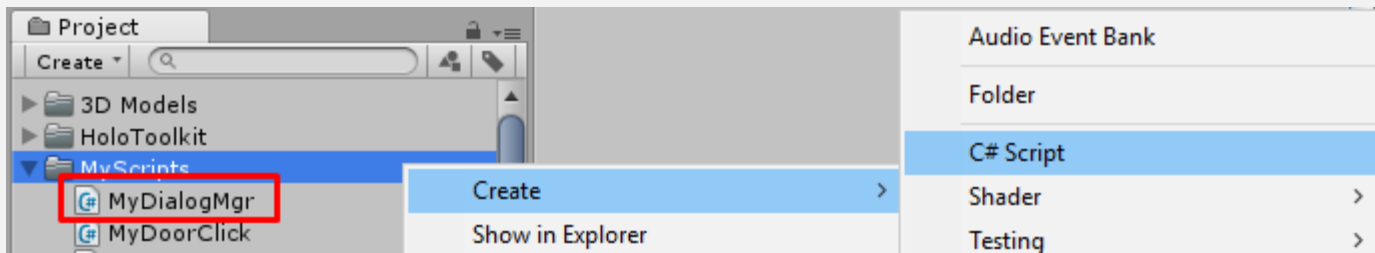
    public void OnInputClicked(InputClickedEventData eventData)
    {
        if (m_Animator != null)
        {
            bool _state = m_Animator.GetBool("character_nearby");
            m_Animator.SetBool("character_nearby", !_state);
        }

        eventData.Use(); // Mark the event as used, so it doesn't fall through to other handlers.
    }
}
```

# Funcionalidad Robot

Creamos un nuevo Script para darle funcionalidad al dialogo del Robot.

Botón derecho sobre la carpeta **MyScripts** -> **Create** -> **C# Script** y lo nombramos "**MyDialogMgr**"



# Funcionalidad Robot

```
using UnityEngine;

public class MyDialogMgr : MonoBehaviour
{
    private bool conversationStarted;
    public enum robotState { Inactivo, hablando, escuchando }
    public robotState state;
    private float count;
    public bool activo;
    public Renderer e_i, e_d;
    public Material normal, rojo, verde;
}
```

```
private void Update()
{
    switch (state)
    {
        case robotState.hablando:
            e_i.material = rojo;
            e_d.material = rojo;
            count += Time.deltaTime;
            if (!MyRobotMgr.instance.speechManager.audioSource.isPlaying && count > 2)
            {
                if (activo)
                {
                    state = robotState.escuchando;
                }
                else
                {
                    state = robotState.Inactivo;
                }
            }
            break;
        case robotState.escuchando:
            e_i.material = verde;
            e_d.material = verde;
            MyRobotMgr.instance.microphoneManager.StartCapturingAudio();
            break;
        default:
            e_i.material = normal;
            e_d.material = normal;
            break;
    }
}
```

```
public void DictationProcess(string dic)
{
    if (activo)
    {
        string texto_out = dic.ToLower();
        if (!conversationStarted)
        {
            conversationStarted = true;
            texto_out = "Hola. Soy Robochachi. Puedes hablar conmigo, pero solo te escucharé cuando tenga los ojitos verdes";
        }
        else
        {
            if (dic.Contains("desconecta"))
            {
                texto_out = "Muy bien. Hasta luego machote";
                activo = false;
                conversationStarted = false;
            }
            else
            {
                if (dic.Contains("tiempo") && dic.Contains("mañana"))
                {
                    texto_out = "Creo que mañana va a llover. Eso no es bueno para mis circuitos.";
                }
                else if (dic.Contains("realidad") && dic.Contains("mixta"))
                {
                    texto_out = "Sin duda la realidad mixta es el futuro... por la cuenta que me trae.";
                }
                else if (dic.Contains("xxxxx"))
                {
                    texto_out = "Apagando sistema....";
                    activo = false;
                    conversationStarted = false;
                }
            }
        }
    }
}
```

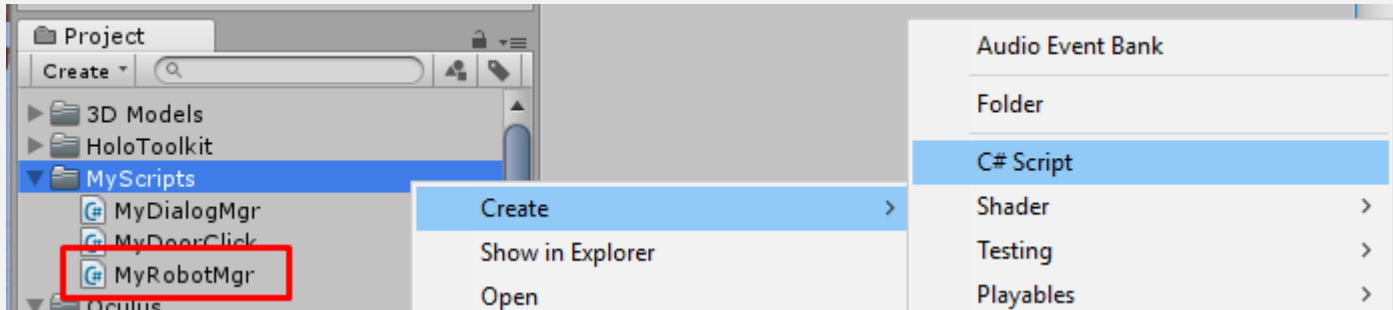
```
        else if (dic.Contains("repite conmigo"))
        {
            texto_out = "Has dicho... " + texto_out;
        }
        else
        {
            int r = Random.Range(0, 6);
            switch (r)
            {
                case 1:
                    texto_out = "¿que tiempo hará mañana?... eso si es una pregunta interesante.";
                    break;
                case 2:
                    texto_out = "No se que quieres decir. Pero si te sirve de algo, mi color favorito es el azul.";
                    break;
                case 3:
                    texto_out = "La vida de un robot es dura. A veces no entiendo a los humanos.";
                    break;
                case 4:
                    texto_out = "Porque no me preguntas sobre la realidad mixta... por ejemplo.";
                    break;
                case 5:
                    texto_out = "También se repetir tus palabras... si me lo pides.";
                    break;
                default:
                    texto_out = "En realidad no soy tan listo aún. Dame un poco de tiempo.";
                    break;
            }
        }
    }
}

MyRobotMgr.instance.speechManager.Speak(texto_out); //no existe aun
count = 0;
state = robotState.hablando;
}
```

# Funcionalidad Robot

Creamos un nuevo Script para darle funcionalidad al Robot.

Botón derecho sobre la carpeta **MyScripts** -> **Create** -> **C# Script** y lo nombramos "**MyRobotMgr**"



# Funcionalidad Robot

```
using UnityEngine;

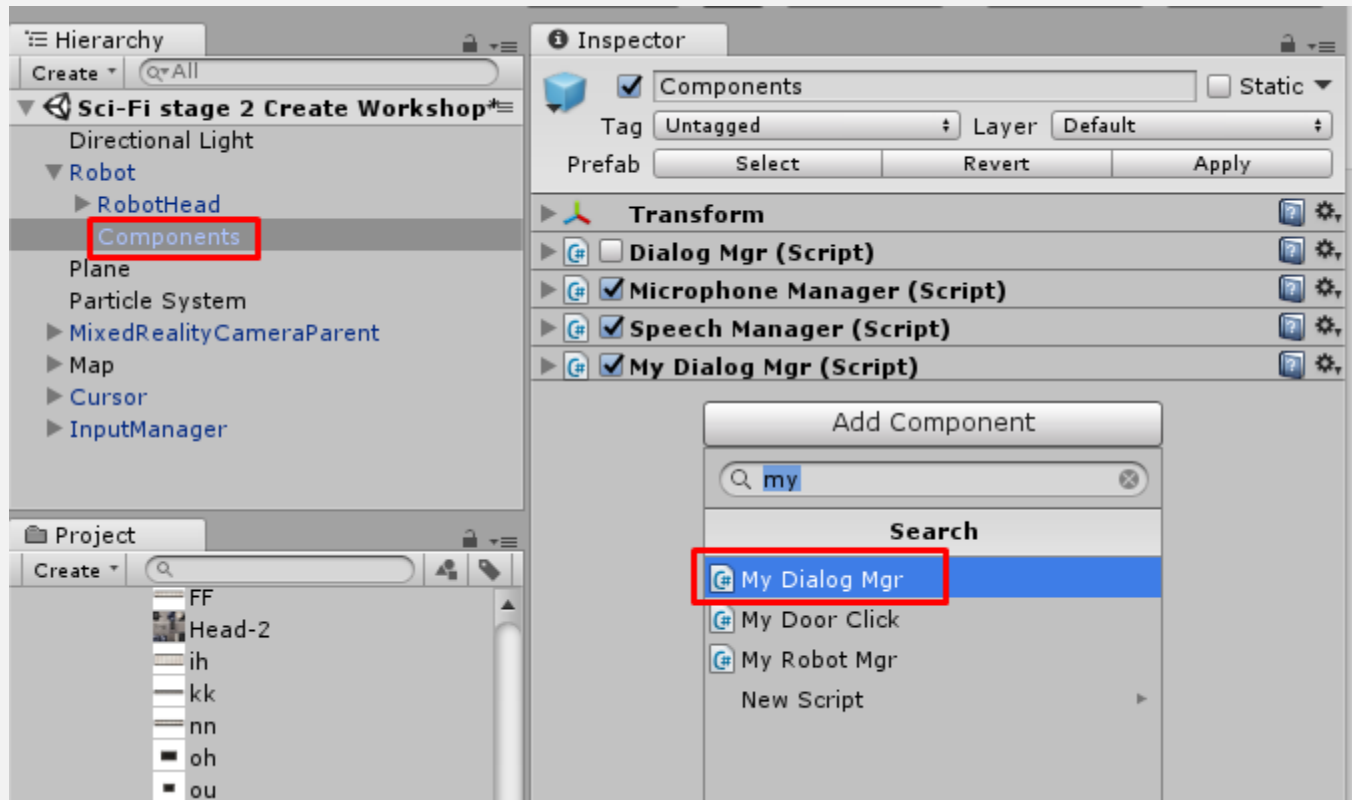
public class MyRobotMgr : MonoBehaviour
{
    public MyDialogMgr dialogMgr;
    public SpeechManager speechManager;
    public MicrophoneManager microphoneManager;
    public static MyRobotMgr instance;
    public bool speaking;

    private void Awake()
    {
        dialogMgr = GetComponent<MyDialogMgr>();
        speechManager = GetComponent<SpeechManager>();
        microphoneManager = GetComponent<MicrophoneManager>();
        instance = this;
    }
}
```



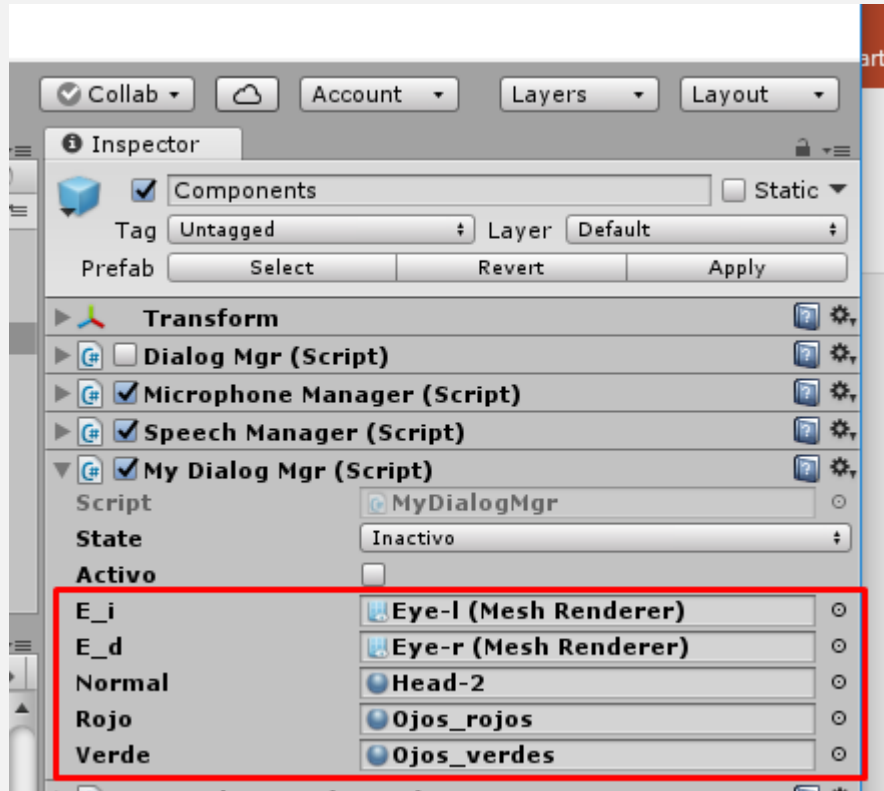
# Funcionalidad Robot

Añadimos el script "**MyDialogMgr**" al GameObject "**Components**"



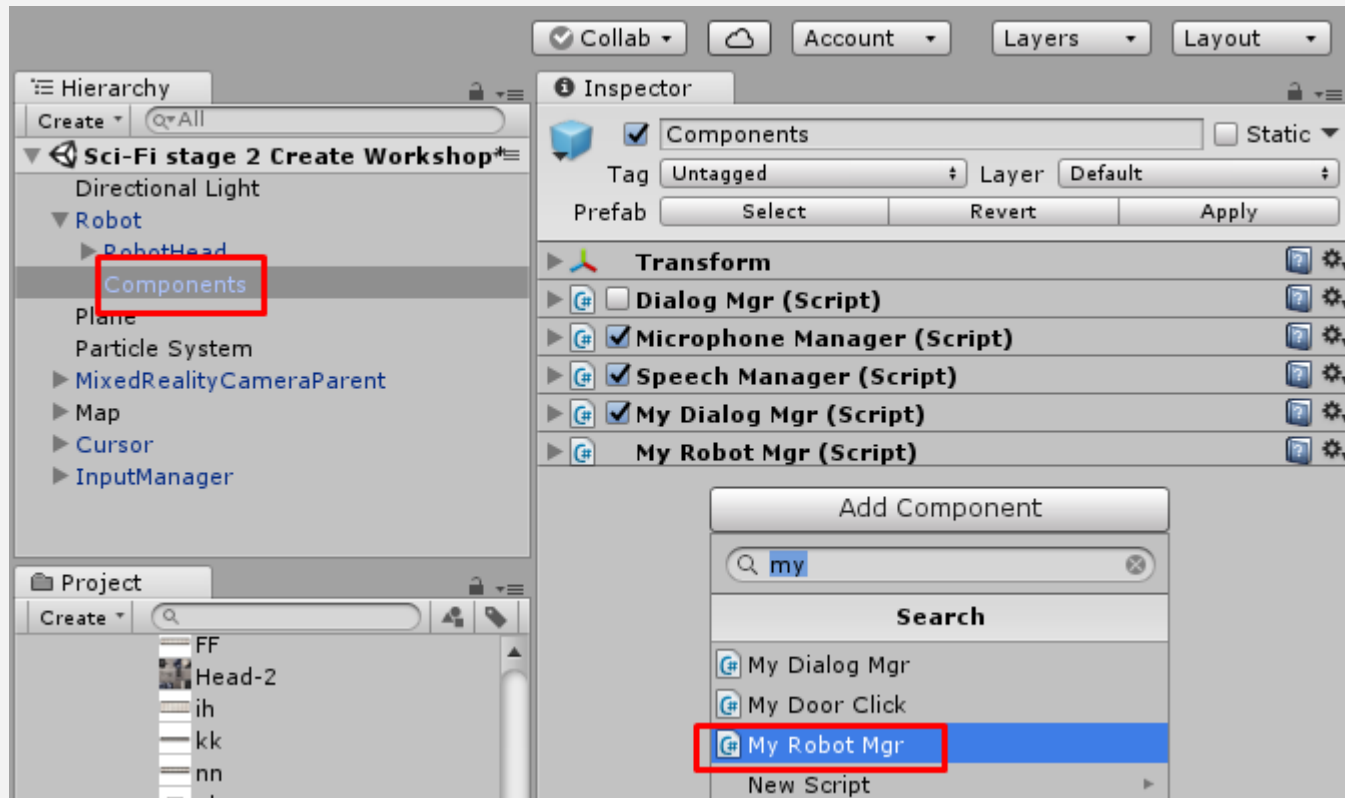
# Funcionalidad Robot

Añadimos las referencias de los **Mesh Renderer** y **Materiales** que usamos en el script.



# Funcionalidad Robot

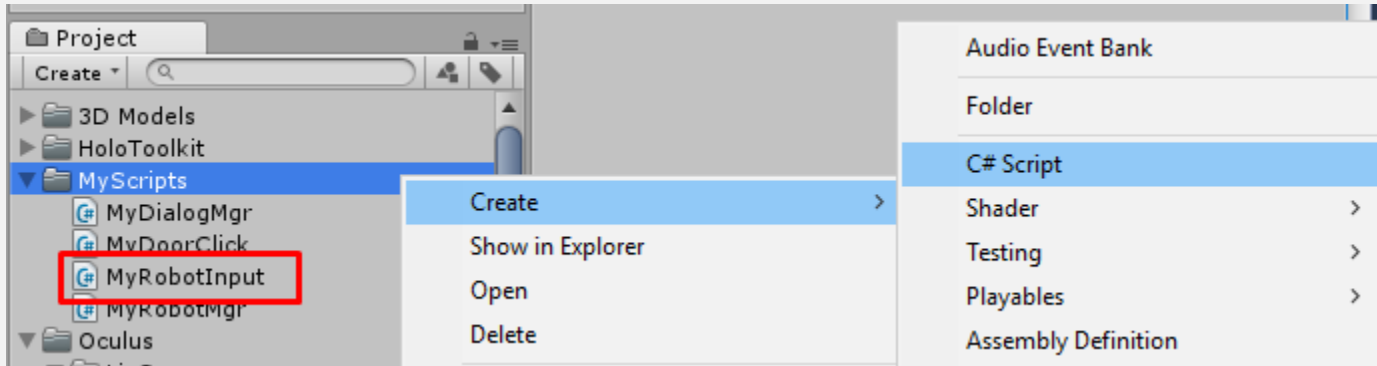
Añadimos el script **"MyRobotMgr"** al GameObject **"Components"**



# Funcionalidad Robot

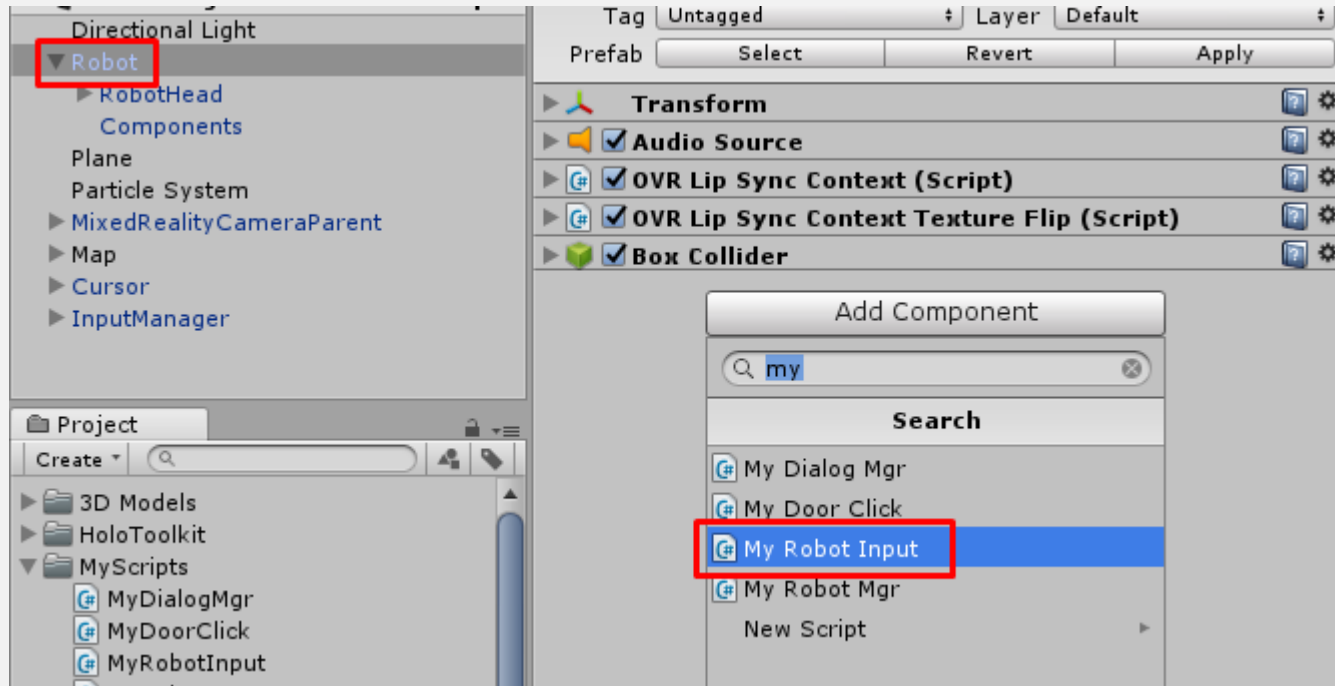
Creamos un nuevo Script para darle funcionalidad al Robot.

Botón derecho sobre la carpeta **MyScripts** -> **Create** -> **C# Script** y lo nombramos "**MyRobotInput**"



# Funcionalidad Robot

Añadimos el script "**MyRobotInput**" al GameObject "**Robot**" y abrimos el Script en Visual Studio para darle funcionalidad.



# Funcionalidad Robot

```
using HoloToolkit.Unity.InputModule;
using UnityEngine;
using UnityEngine.XR.WSA;

public class MyRoboInput : MonoBehaviour, IInputClickHandler
{
    public void OnInputClicked(InputClickedEventData eventData)
    {
        if (HolographicSettings.IsDisplayOpaque)
        {
            MyRobotMgr.instance.dialogMgr.activo = true;
            MyRobotMgr.instance.dialogMgr.DictationProcess("Hola machote");
        }
        eventData.Use(); // Mark the event as used, so it doesn't fall through to other handlers.
    }
}
```

# Microsoft Learn

Friction-free online learning. Have fun, unlock achievements, and get certified.



# Docs.microsoft.com

One global service. The latest in technical documentation, training and solutions.



