

tags: unity openai

Referencias:

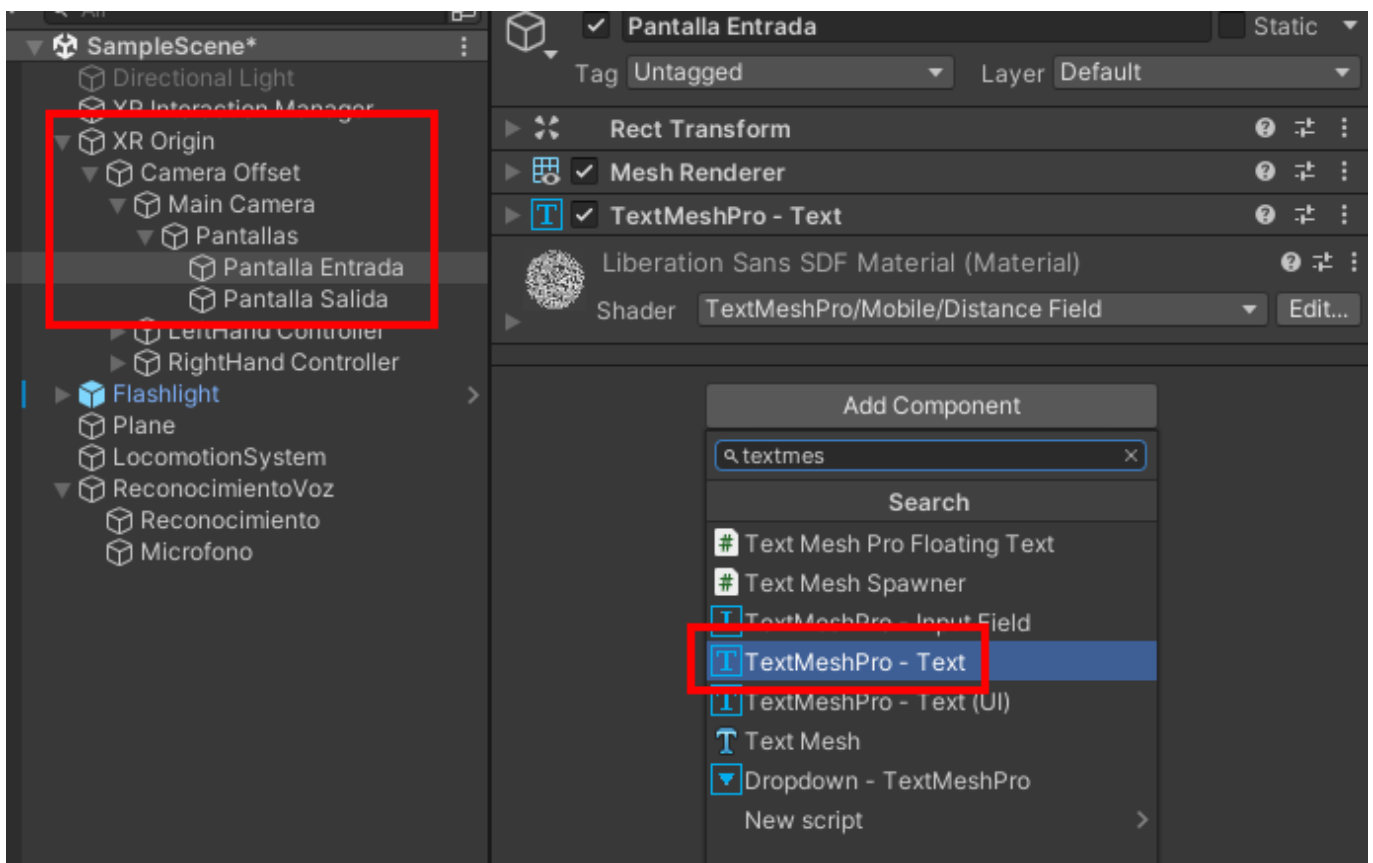
<https://www.youtube.com/watch?v=MQfVCY9qgEU&t=243s>

<https://github.com/srcnalt/OpenAI-Unity>

Añadir pantallas de texto a la visión

Primero añadimos dos pantallas, que será una de entrada y otra de salida que se moverán con nuestra vista y que las podemos esconder pulsando el botón A

Creamos dos pantallas de Texto. Una de texto de entrada y otra de texto de Salida. Creamos dos GameObjects y los colgamos de "Main Camera" de nuestro jugador. Si queremos quitar las dos a la vez lo ponemos dentro de otro GameObject que he llamado Pantalla. Asignamos un "TextMeshPro - Text" a cada gameobject.



Script para que aparezca la pantalla si deajo pulsado el botón A del mando derecho:

```
using UnityEngine;
using UnityEngine.XR;
using TMPro;
```

```
public class Botones : MonoBehaviour
{
    private InputDevice oculusController;
    private bool primaryButtonPressed = false;
    [SerializeField] private GameObject Pantallas;

    void Start()
    {
        // Get the input device for the Right hand
        oculusController = InputDevices.GetDeviceAtXRNode(XRNode.RightHand);
    }

    void Update()
    {
        // Check if the primary button A is pressed
        if (oculusController.TryGetFeatureValue(CommonUsages.primaryButton,
out bool primaryButtonValue) && primaryButtonValue && !primaryButtonPressed)
        {
            primaryButtonPressed = true;
            Debug.Log("button pressed");
            Pantallas.SetActive(true);
        }

        // Check if the primary button A is released
        if (oculusController.TryGetFeatureValue(CommonUsages.primaryButton,
out bool newPrimaryButtonValue) && !newPrimaryButtonValue &&
primaryButtonPressed)
        {
            primaryButtonPressed = false;
            Debug.Log("button released");
            Pantallas.SetActive(false);
        }
    }
}
```

Poner y quitas con un click:

```
using UnityEngine;
using UnityEngine.XR;
using TMPro;

public class Botones : MonoBehaviour
{
    private InputDevice oculusController;
    private bool primaryButtonPressed = false;
    [SerializeField] private GameObject Pantallas;
    int i=0;
```

```
void Start()
{
    // Get the input device for the Right hand
    oculusController = InputDevices.GetDeviceAtXRNode(XRNode.RightHand);
}

void Update()
{
    // Check if the primary button A is pressed
    if (oculusController.TryGetFeatureValue(CommonUsages.primaryButton,
out bool primaryButtonValue) && primaryButtonValue && !primaryButtonPressed)
    {
        primaryButtonPressed = true;
        Debug.Log("button pressed");
        Pantallas.SetActive(true);
    }

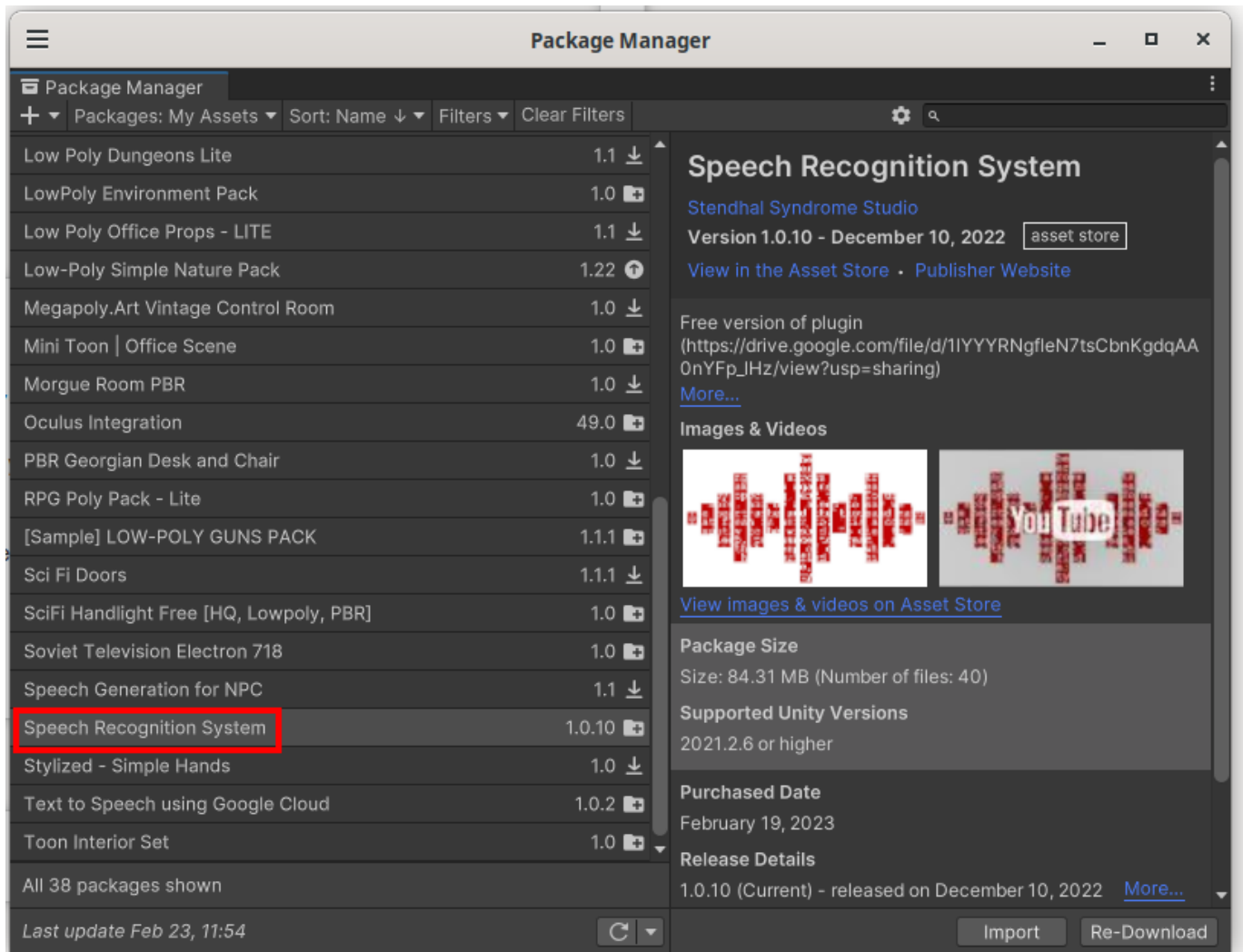
    // Check if the primary button A is released
    if (oculusController.TryGetFeatureValue(CommonUsages.primaryButton,
out bool newPrimaryButtonValue) && !newPrimaryButtonValue &&
primaryButtonPressed)
    {
        primaryButtonPressed = false;
        Debug.Log("button released");
        if ( i==0)
        {
            i=1;
        } else {
            Pantallas.SetActive(false);
            i=0;
        }
    }
}
}
```

Speech to Text

Instalación de Speech to text. Asset de unityStore que cuesta 49.89€ (no es dinero)

<https://assetstore.unity.com/packages/tools/audio/speech-recognition-system-187171>

Lo importamos, se llama: "Speech Recognition System"



Descargamos el modelo de lenguaje “Spanish” está en el fichero de documentación y es un enlace a Google Drive (***investigar para conseguir el original que creo que es de Apache***)

<https://drive.google.com/file/d/11BvWNcp5EapeW4rbSjvl4MIPUhrz2DdD/view>

Copiamos el directorio en Assets/StreamingAssets/SpeechRecognitionSystem/model/spanish

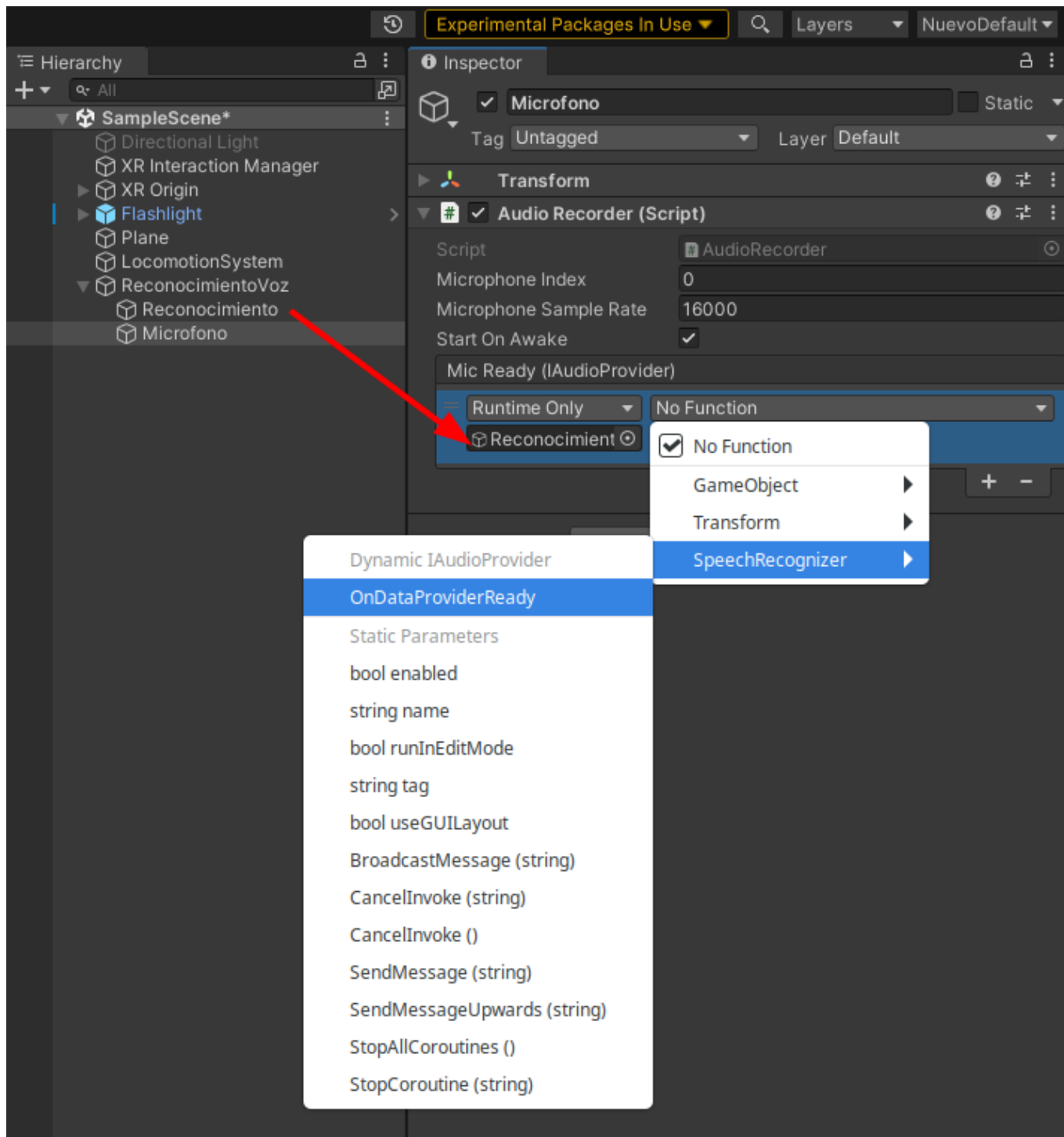
Creamos dos GameObjects:

Reconocimiento
Microfono

Y les asignamos los scripts: “SpeechRecognitionSystem > Scripts” estos dos: “SpeechRecognizer” y “Audio recorder”

Reconocimiento -> Script "SpeechRecognizer"
Microfono -> Script "Audio recorder"

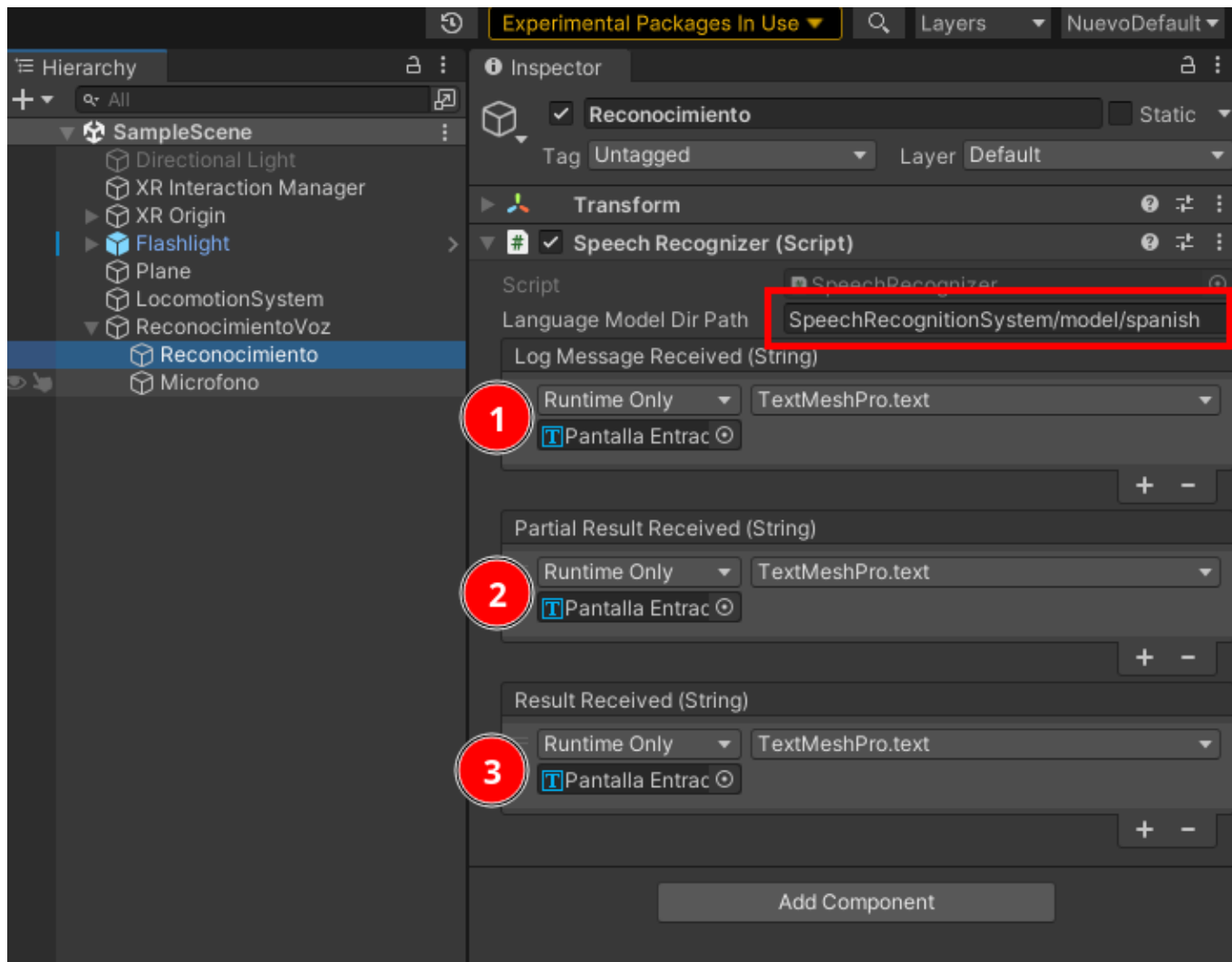
Configuramos así el script AudioRecorder: le asignamos el GameObject de Reconocimiento y seleccionamos la función “SpeechRecognizer.OnDataProviderReady”



Lo cambiamos en el script:

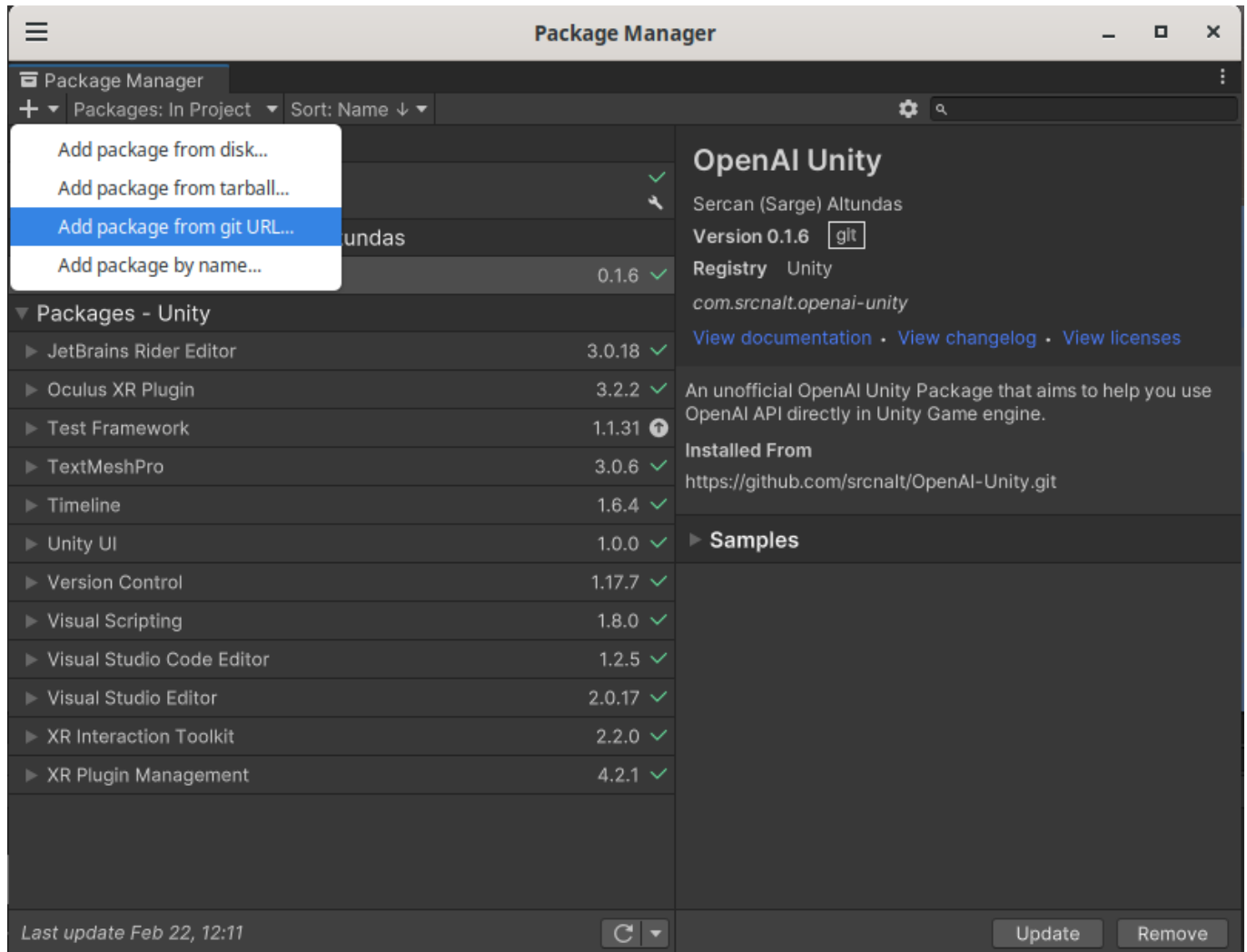
```
public string LanguageModelDirPath =
"SpeechRecognitionSystem/model/spanish";
```

Y por si acaso también en Unity, en el GameObject de Reconocimiento, cambiamos en el Script el modelo a Spanish y que la salida sea la pantalla de Entrada. La primera es de log, la segunda el partial (que es el que usaremos) y la tercera el final



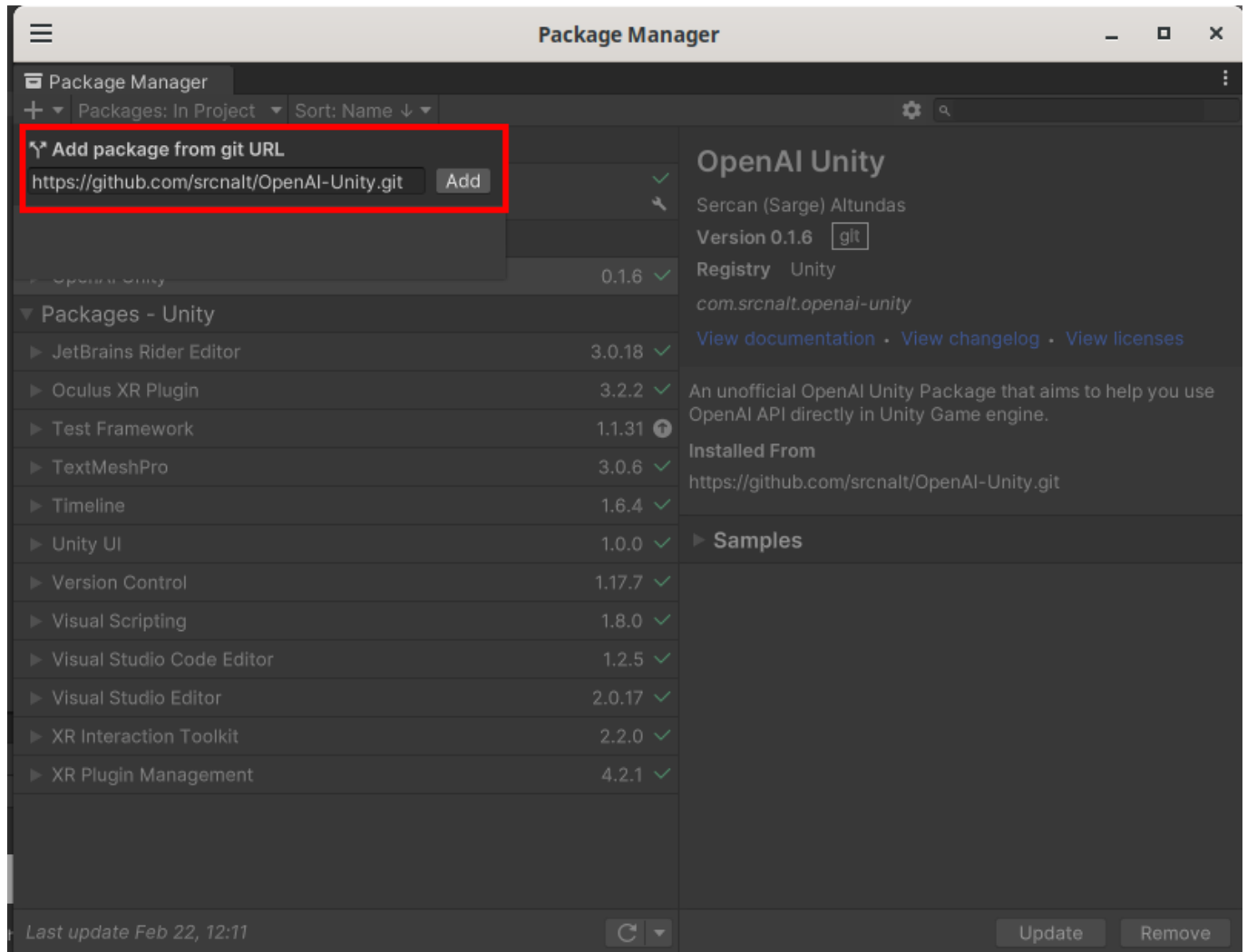
Openai

Instalamos el Asset de openai desde el repositorio de GIT. Vamos a “Window > Package Manager” y pulsamos en el + y seleccionamos “Add Package from git URL...”

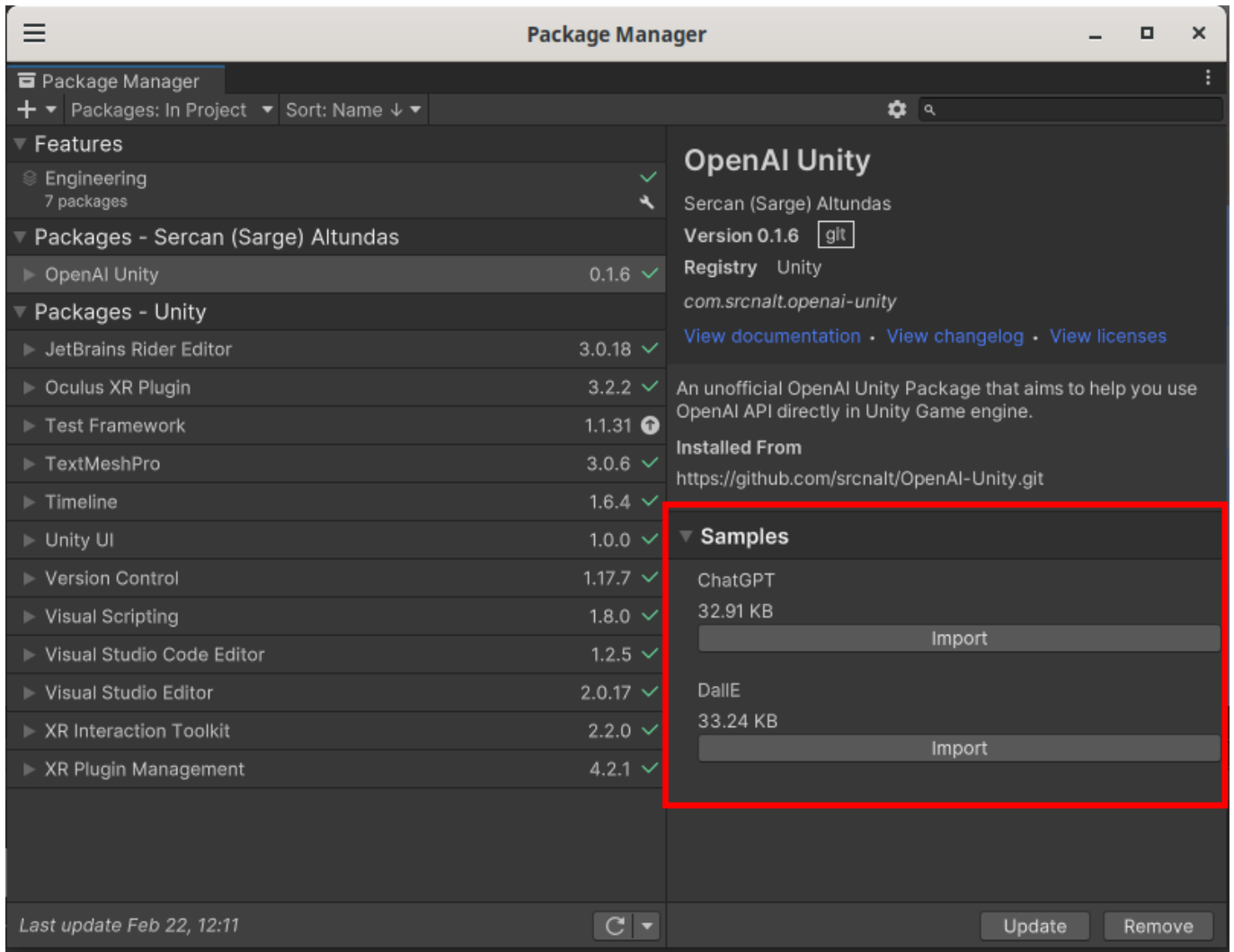


Introducimos la URL que es:

<https://github.com/srcnalt/OpenAI-Unity.git>



A la derecha importamos los ejemplos:



Nos aparecerá en “Samples > OpenAI Unity > 0.1.6 > ChatGPT” Aquí creamos un GameObject que se llame ChatGPT y añadimos el script y lo modificamos.

En la línea

```
private OpenAIApi openai = new OpenAIApi();
```

Introducimos nuestra API_KEY de openai:

```
private OpenAIApi openai = new OpenAIApi(sk-
CxRE16KA2qgjtowRM6tyT3BlbkFJBBoXbXxTCnSi0GAJ1xes);
```

El script de OpenAi es una pantalla con un prompt abajo y la tecla de enviar. Lo cambiamos simplemente por dos pantallas, entrada y salida
Añadimos que use la biblioteca de TextMeshPro

```
using TMPro;
```

Cambiamos también las variables de entrada:

```
[SerializeField] private InputField inputField;
[SerializeField] private Button button;
```

```
[SerializeField] private Text textArea;
```

Por estas.

```
[SerializeField] private TextMeshPro inputField;  
[SerializeField] private TextMeshPro textArea;
```

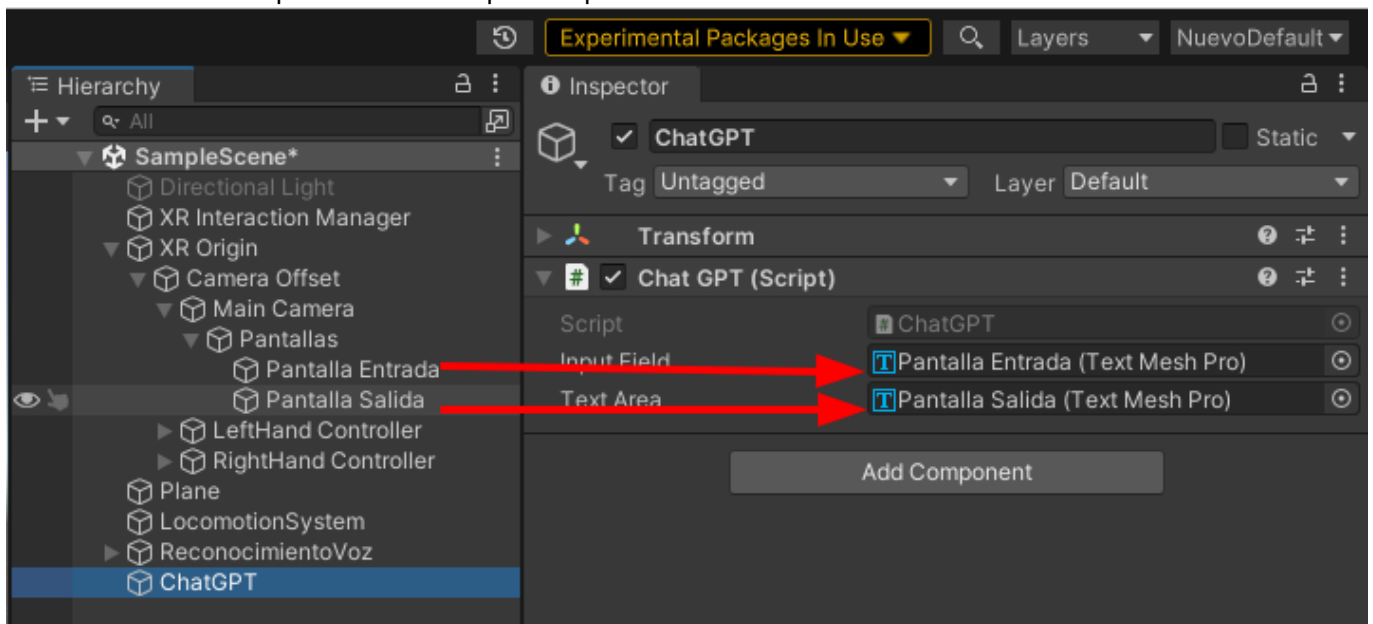
Quitamos las referencias a button, ya que llamaremos a OpenAI desde el reconocimiento de voz:

```
private void Start()  
{  
    //button.onClick.AddListener(SendReply);  
}
```

```
//button.enabled = false;
```

```
//button.enabled = true;
```

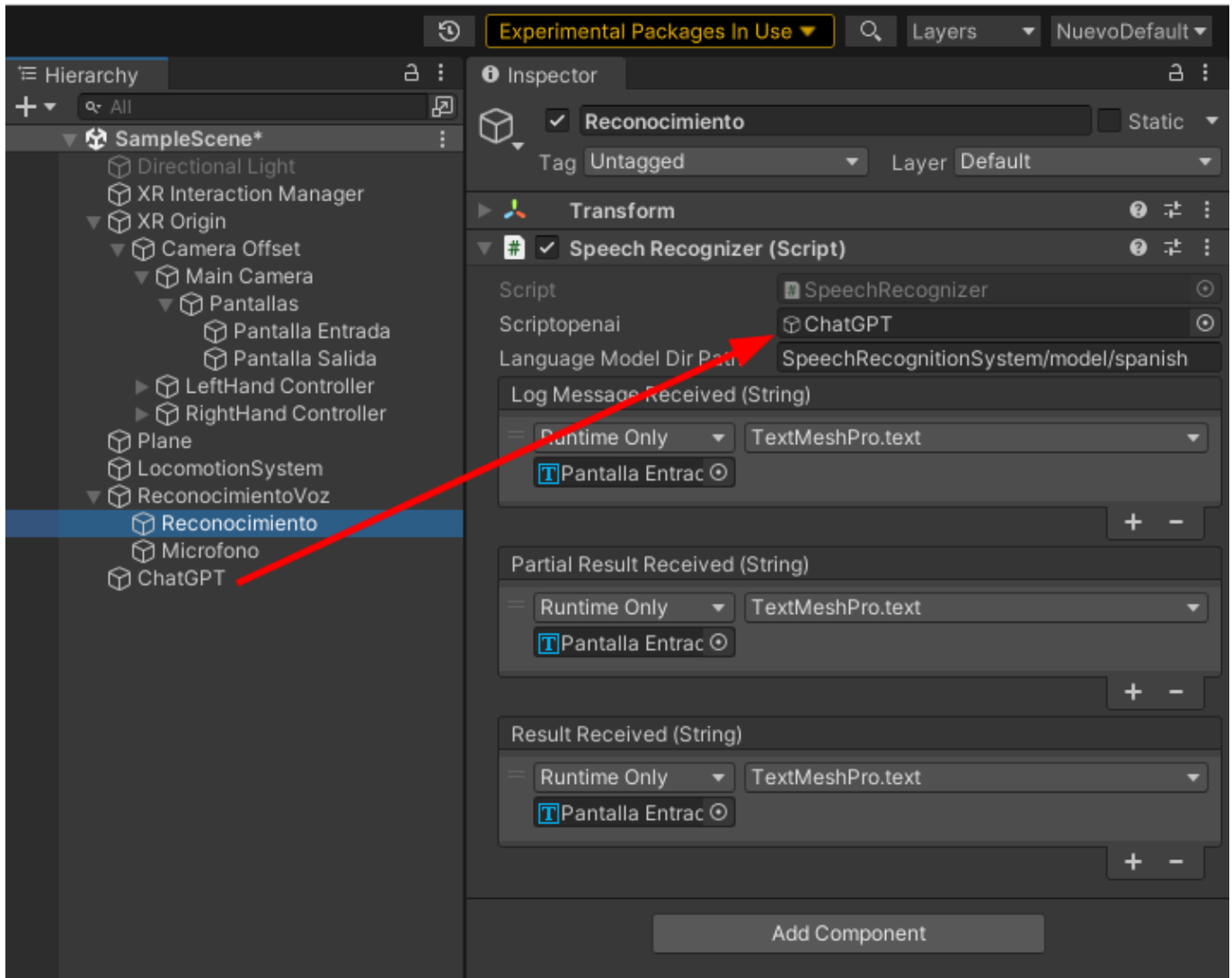
Ahora añadimos las pantallas al Script de OpenAI



Ahora nos faltaría añadir que cuando acabe de escuchar el texto, haga la llamada a OpenAI.

Vamos al Script de "SpeechRecognizer" añadimos esta línea y arrastramos el GameObject donde está el script de chatGPT:

```
[SerializeField] private GameObject scriptopenai;
```



Ahora vamos donde detecta que hemos dejado de hablar, es en la línea:

```
if ( _recognitionFinalResultsQueue.TryDequeue( out string result ) ) {
    if ( result != string.Empty )
        ResultReceived?.Invoke( result );
}
```

Y añadimos la llamada a Sendreply:

Text to Speech

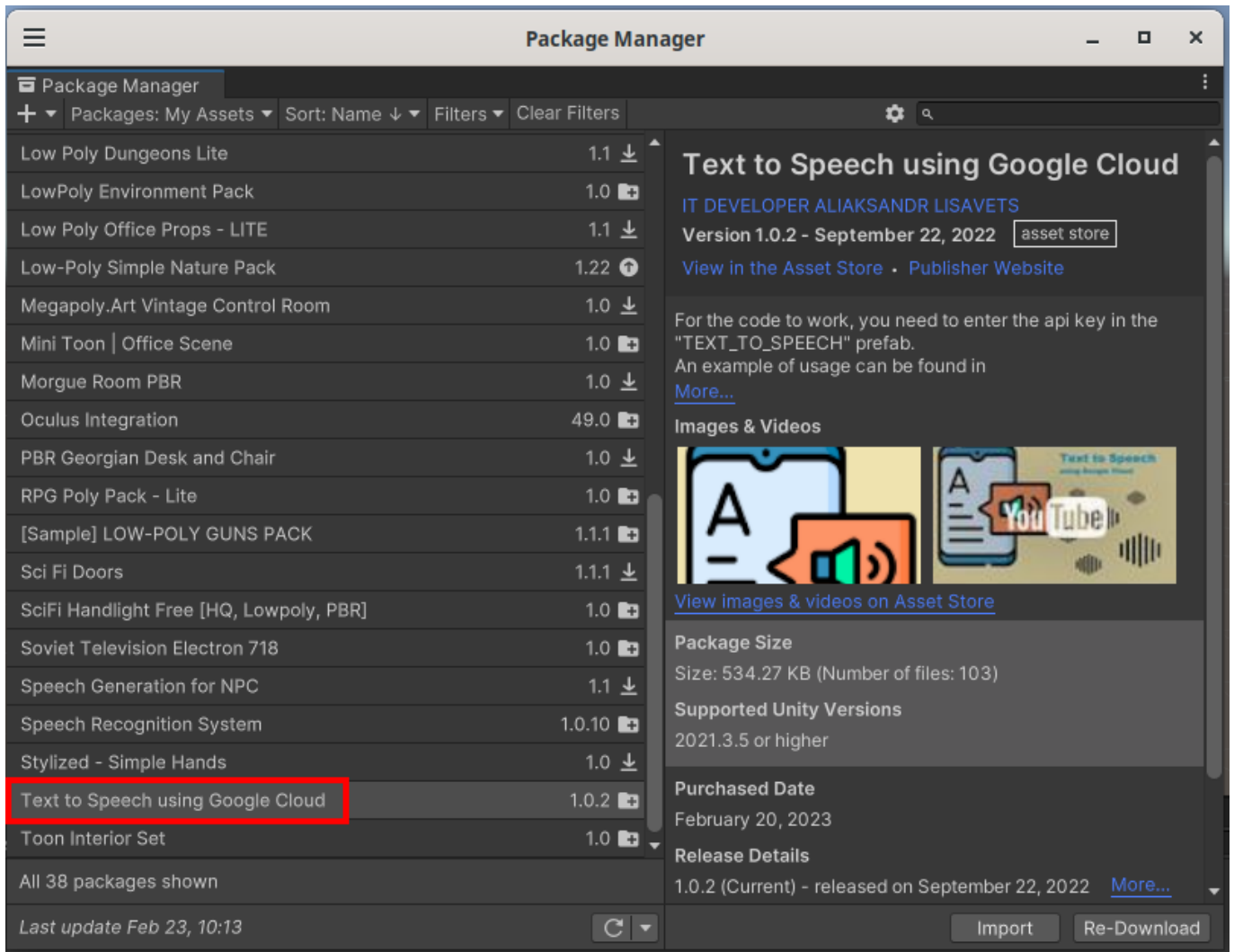
Facturación de Google:

<https://console.cloud.google.com/apis/api/texttospeech.googleapis.com/metrics?hl=es&project=unityts-378321>

Usamos el Asset "Text to Speech using Google Cloud" que vale 4.98€:

<https://assetstore.unity.com/packages/tools/utilities/text-to-speech-using-google-cloud-232515>

Se añade el Asset en Unity:



Añadimos un GameObject que se llame TextToSpeech. Le añadimos 3 componentes, un AudioSource y dos Scripts:

```
GoogleTextToSpeech > Scripts > TextToSpeech (para la API)
GoogleTextToSpeech > Scripts > Example > TextToSpeechExample (para la configuración)
```

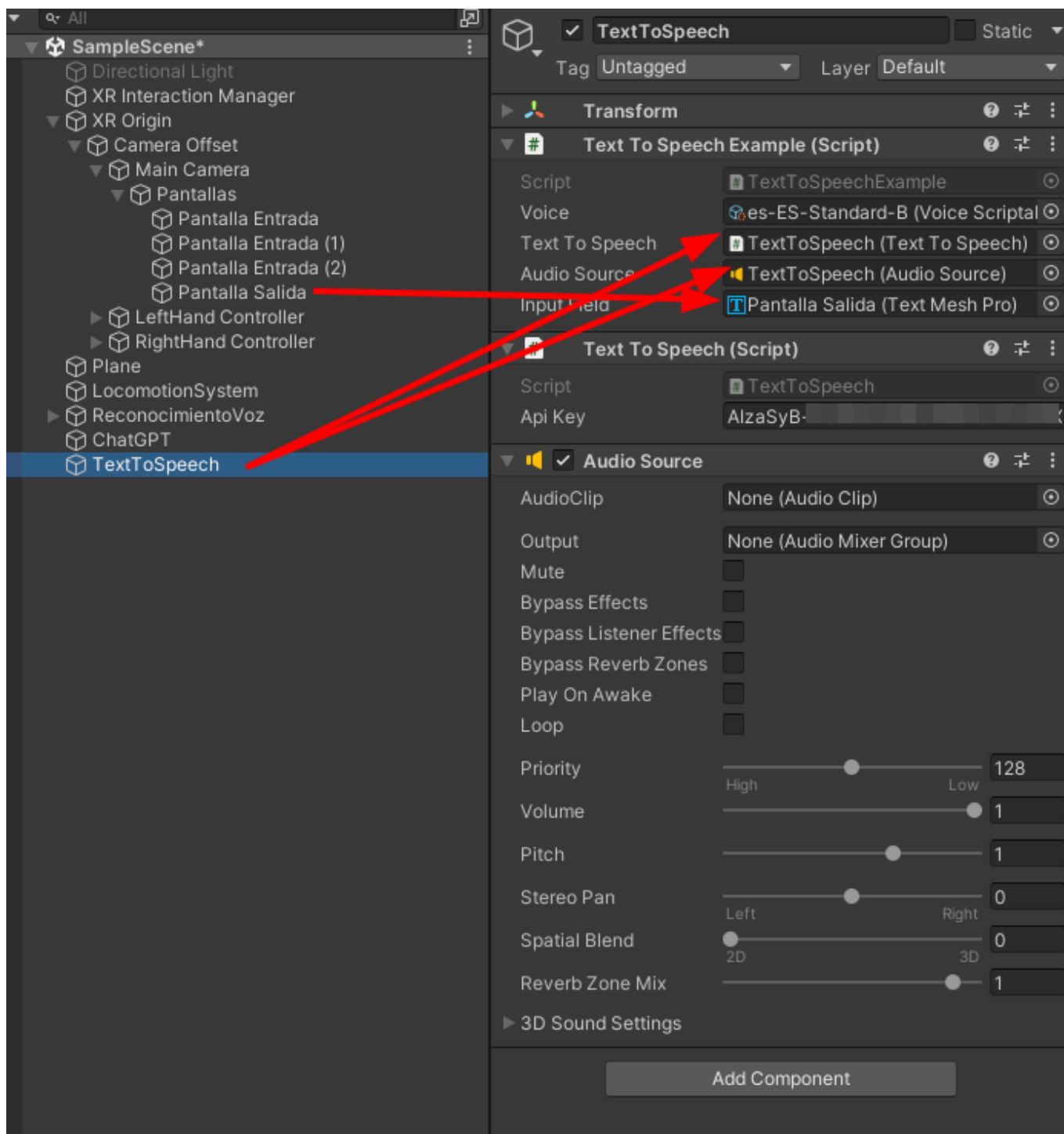
Cambiamos la línea del script TextToSpeechExample de TextMeshPro a TextMeshPro

```
[SerializeField] private TextMeshPro inputField;
```

Añadimos los siguientes campos en el GameObject TextToSpeech:

```
Voice: es-ES-Standard-B
Text To Speech: (arrastramos el GameObject TextToSpeech)
Audio Source: (arrastramos el GameObject TextToSpeech y quitamos Play On Awake)
Input Field: (arrastramos la salida de Openai, que es el GameObject Pantalla Salida)
```

Quedaría mas o menos así:



Ahora hacemos que cuando acabe la llamada a openAI, llame a TextToSpeech. En el script ChatGPT.cs añadimos estas líneas para poder llamar al método PressBtn() que es el que lanza TextToSpeech desde otro script:

```
using GoogleTextToSpeech.Scripts.Example;
```

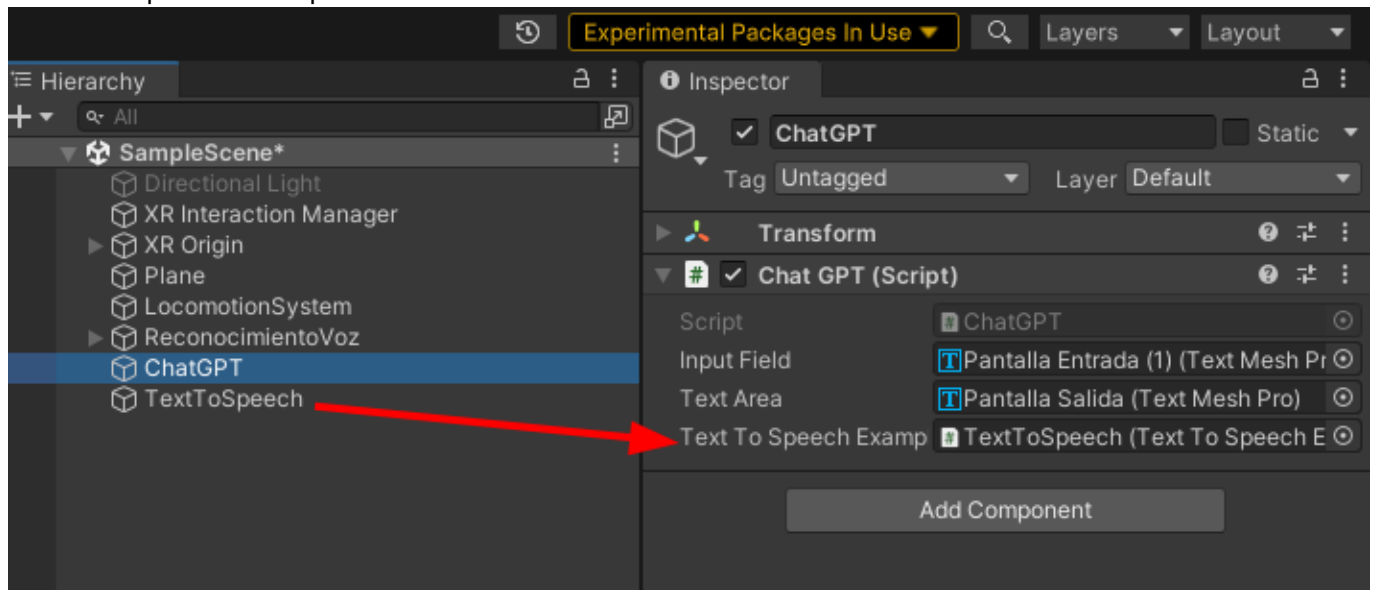
```
[SerializeField] private TextToSpeechExample textToSpeechExample ;
```

Ahora al final del SendReply() añadimos:

```
textToSpeechExample.PressBtn();
```

Ahora señalamos el gameObject chatGPT y arrastramos el gameObject TextToSpeech dentro de:

"Text To Speech Example"



From:
<http://wiki.legido.com/> - **Legido Wiki**

Permanent link:
http://wiki.legido.com/doku.php?id=informatica:realidad_virtual:unity:openai

Last update: **2023/03/03 22:29**

