

PRÁCTICA 5. RECONOCIMIENTO DE VOZ

Funcionamiento general de la práctica

Para esta práctica he querido implementar un reconocimiento de voz que permita a 2 cubos cambiar su color. Esto ha sido hecho tanto con el KeywordRecognizer como con el DictationRecognizer.

Cuando pulsemos la 'X' el DictationRecognizer se activará y empezará a reconocer palabras que digamos, se parará su ejecución si pulsamos la 'Z' y finalmente liberará los recursos si pulsamos la 'D'. Los colores que reconoce el DictationRecognizer son el amarillo, violeta, negro, azul y naranja.

Por otro lado, la tecla de activación del KeywordRecognizer es la 'Q', la de Stop es la barra espaciadora y para liberar sus recursos (Dispose) habrá que pulsar la tecla 'E'. Se reconocen los colores amarillo, rojo, verde, azul y blanco.

DictationRecognizer

En el script he definido una serie de variables: 1 variable para cada uno de los 5 colores (variables tipo Color), un MeshRenderer que me ayudará a cambiar el color al objeto al que está asignado el script y una variable DictationRecognizer. Los colores deberán ser asignados desde el inspector del objeto en Unity.

```
DictationRecognizer dictationRecognizer;  
  
public Color amarillo;  
public Color violeta;  
public Color negro;  
public Color azul;  
public Color naranja;  
MeshRenderer myRenderer;
```

En el start lo primero que hago es construir un nuevo objeto DictationRecognizer y asignarle la funcionalidad de sus 4 funciones de escucha mediante 4 funciones realizadas en el script.

```

void Start () {
    dictationRecognizer = new DictationRecognizer();

    dictationRecognizer.DictationResult += onDictationResult;
    dictationRecognizer.DictationHypothesis += onDictationHypothesis;
    dictationRecognizer.DictationComplete += onDictationComplete;
    dictationRecognizer.DictationError += onDictationError;

    //dictationRecognizer.Start();
}

```

La función que llevará la voz cantante será 'onDictationResult', que será la función que cambiará el color al objeto dependiendo de la palabra leída.

```

void onDictationResult(string text, ConfidenceLevel confidence)
{
    // write your logic here
    //Debug.LogFormat("Dictation result: " + text);
    if (text == "amarillo") {
        myRenderer = GetComponent<MeshRenderer>();
        myRenderer.material.color = amarillo;
    }
    if (text == "violeta") {
        myRenderer = GetComponent<MeshRenderer>();
        myRenderer.material.color = violeta;
    }
    if (text == "negro") {
        myRenderer = GetComponent<MeshRenderer>();
        myRenderer.material.color = negro;
    }
    if (text == "naranja") {
        myRenderer = GetComponent<MeshRenderer>();
        myRenderer.material.color = naranja;
    }
    if (text == "azul") {
        myRenderer = GetComponent<MeshRenderer>();
        myRenderer.material.color = azul;
    }
}

```

Finalmente en el Update controlaremos el funcionamiento del script.

```
void Update() {  
    if (Input.GetKey(KeyCode.X)) {  
        dictationRecognizer.Start();  
    }  
    if (Input.GetKey(KeyCode.Z)) {  
        dictationRecognizer.Stop();  
    }  
    if (Input.GetKey(KeyCode.D)) {  
        dictationRecognizer.Dispose();  
    }  
}
```

KeywordRecognizer

Este script tiene el mismo concepto que el anterior, solo que añadiremos una variable más: un array de strings que contendrá las palabras que se reconocen y que definiremos en el Start.

```
private KeywordRecognizer kr;  
public string[] keywords;  
public Color amarillo;  
public Color verde;  
public Color rojo;  
public Color azul;  
public Color blanco;  
MeshRenderer myRenderer;  
// Start is called before the first frame update  
void Start()  
{  
    keywords = new string[5];  
    keywords[0] = "amarillo";  
    keywords[1] = "rojo";  
    keywords[2] = "verde";  
    keywords[3] = "azul";  
    keywords[4] = "blanco";  
    kr = new KeywordRecognizer(keywords);  
}
```

En el Update controlaremos el funcionamiento del script de la siguiente manera:

```
void Update()
{
    if (Input.GetKey(KeyCode.Q)) {
        kr.OnPhraseRecognized += OnKeywordsRecognized;
        kr.Start();
    }
    if (Input.GetKey(KeyCode.Space)) {
        kr.OnPhraseRecognized -= OnKeywordsRecognized;
        kr.Stop();
    }
    if (Input.GetKey(KeyCode.E)) {
        kr.Dispose();
    }
}
```

Como se puede observar en los 2 primeros if, hay un atributo del objeto kr que se llama 'OnPhraseRecognized' que activa un evento del tipo Phrase Recognizer para poder reconocer frases o palabras. Este atributo está suscrito a una función tipo void que recibe como parámetro un evento PhraseRecognizer. Esta función es el núcleo de nuestro script y es lo que permite al objeto cambiar su color.

```
void OnKeywordsRecognized(PhraseRecognizedEventArgs args) {
    if (args.text == keywords[0]) {
        myRenderer = GetComponent<MeshRenderer>();
        myRenderer.material.color = amarillo;
    }
    if (args.text == keywords[1]) {
        myRenderer = GetComponent<MeshRenderer>();
        myRenderer.material.color = rojo;
    }
    if (args.text == keywords[2]) {
        myRenderer = GetComponent<MeshRenderer>();
        myRenderer.material.color = verde;
    }
    if (args.text == keywords[3]) {
        myRenderer = GetComponent<MeshRenderer>();
        myRenderer.material.color = azul;
    }
    if (args.text == keywords[4]) {
        myRenderer = GetComponent<MeshRenderer>();
        myRenderer.material.color = blanco;
    }
}
```