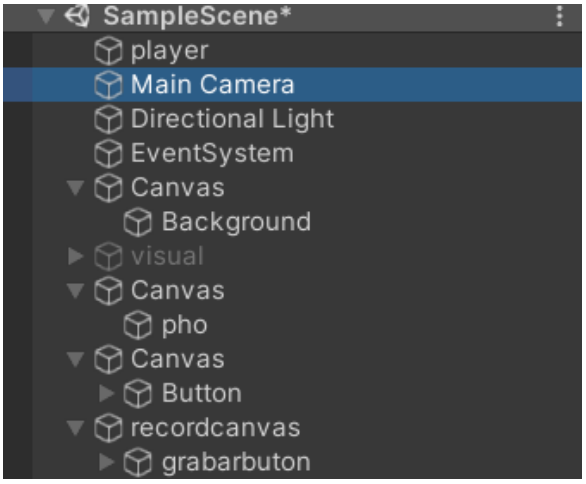
Interfaces inteligentes
Práctica 4
Micrófono y cámara

Por: ADRIÁN FLEITAS DE LA ROSA

La aplicación muestra en pantalla la webcam en tiempo real, permite pausar/reanudar el vídeo con un botón, permite grabar/reproducir la voz del jugador, hacer una captura y mover/cambiar el color de un cubo con comandos de voz. La interfaz luce así:

He tenido que crear dos botones, un cubo , un canvas para el vídeo y otro para la captura.



Lo primero fue capturar la webcam y mostrarlos como raw image de un canvas.

```
}
cam = new WebCamTexture (devices[0].name, Screen.width, Screen.height);
cam.Play();
background.texture = cam;
camAvaliable = true;
```

```
void Update()
{
    if (!camAvaliable)
      return;
    float ratio = (float)cam.width / cam.height;
    fit.aspectRatio = ratio;
    float scaleY = cam.videoVerticallyMirrored ? -1f: 1f;
    background.rectTransform.localScale = new Vector3(1f,scaleY, 1f);
    int orient = -cam.videoRotationAngle;
    background.rectTransform.localEulerAngles = new Vector3(0, 0, orient);
}
```

Los eventos no asociados al botón de grabar son manejados con un controlador de juego, es decir, el reconocedor de voz lanza el evento que corresponda al comando a través del controlador.

```
0 references
void Start()
{
    keywordActions.Add("左", controladorjuego.left);
    keywordActions.Add("動く", controladorjuego.right);
    keywordActions.Add("変えて", controladorjuego.change);
    keywordActions.Add("up", controladorjuego.up);
    keywordActions.Add("down", controladorjuego.down);
    keywordActions.Add("写真", controladorjuego.photo);
    keywordActions.Add("遠い", controladorjuego.away);
    keywordActions.Add("近く", controladorjuego.close);
    keywordRecognizer = new KeywordRecognizer(keywordActions.Keys.ToArray());
    keywordRecognizer.OnPhraseRecognized += OnKeywordsRecognized;
    keywordRecognizer.Start();
}
```

Por lo tanto, el jugador tiene su script asociado que le permite tanto moverse como cambiar de color lanzado por comando a través del controlador y la clase reconocedor.
Para moverse se usa el transform del cubo, por ejemplo:

```
transform.position = new Vector3(transform.position.x - Speed,
transform.position.y, transform.position.z);
```

Y el cambio de color (aleatorio):

```
1 reference
void ChangeColor() {
  random1 = UnityEngine.Random.Range(0f, 1f);
  random2 = UnityEngine.Random.Range(0f, 1f);
  random3 = UnityEngine.Random.Range(0f, 1f);
  color = new Color(random1, random2, random3);
  playerRenderer.material.SetColor("_Color", color);
}
// Start is called before the first frame update
0 references
void Start()
{
    playerRenderer = GetComponent<Renderer>();
    controladorjuego.left += MoveLeft;
    controladorjuego.right += MoveRigth;
    controladorjuego.away += MoveAway;
    controladorjuego.close += MoveClose;
    controladorjuego.down += MoveDown;
    controladorjuego.up += MoveUp;
    controladorjuego.change += ChangeColor;
}
```

En cuanto a hacer una captura al usar el controlador realmente se usa el método TakePhoto de la clase camera. La clase camera y reconocedor están enlazadas a la main camera para que no dependan de que esté o no otro objeto en pantalla.
Llamando a través del controlador al método cuando se recibe el comando de voz:

```
1 reference
void TakePhoto() {
  Texture2D photo = new Texture2D(cam.width, cam.height);
  photo.SetPixels(cam.GetPixels());
  photo.Apply();
  image.texture = photo;
}
```

Cambiando este la textura de la imagen del canvas.
La clase del botón que pausa el vídeo de nuevo utiliza el controlador por tanto en su propia clase solo llama a pause que es en realidad el método manage de la clase camera:

```
0 references
void Start()
{
    GetComponent<Button>().onClick.AddListener(manage);
    GetComponentInChildren<Text>().text = "Pause/Resume";
}
1 reference
void manage() {
  controladorjuego.pause();
}
```

```
1 reference
void Manage() {
  if (cam.isPlaying) {
    cam.Pause();
  } else {
    cam.Play();
  }
}
```

Por último el botón de grabar/reproducir audio se puede resumir en una llamada al método Microphone.Start en la primera pulsación y Microphone.End, audio.Play en la segunda.

```
void graba() {
  if (!press) {
    Debug.Log("graba");
    recording = Microphone.Start(Microphone.devices[0], false, 10, 44100);
    press = true;
  } else {
    audioSource.clip = recording;
    Microphone.End(Microphone.devices[0]);
    audioSource.Play();
    Debug.Log("play");
    press = false;
  }
```