

Reconocimiento de voz

Por: Adrián Fleitas de la Rosa

Reconocer palabras clave

- Dictionary<string,Action> al que luego se añaden palabras a reconocer
- OnkeyWordRecognizer(), invoca la Action correspondiente
- Suscribirse a comenzar y pause del controlador en Start()
- Función Pause /comienza para encender y parar el reconocedor

```
private Dictionary<string, Action> keywordActions = new Dictionary<string, Action>();  
6 references  
private KeywordRecognizer keywordRecognizer;  
1 reference  
void OnKeywordsRecognized(PhraseRecognizedEventArgs args) {  
    Debug.Log("Keyword: " + args.text);  
    keywordActions[args.text].Invoke();  
}  
0 references  
void Start()  
{  
    controladorjuego.comenzar += comienza;  
    controladorjuego.pause += Pause;  
    keywordActions.Add("左", controladorjuego.left);  
    keywordActions.Add("へん", controladorjuego.model);  
    keywordActions.Add("動<", controladorjuego.right);  
}
```

Dictado de palabras

- Añadir los eventos DictationResult, DictationHypothesis, DictationComplete, Dictation Error.
- Además añadí las Action para empezar y terminar el dictado
- Cambia el texto
- Activa audio si

dices bakamitai

en japonés

0 references

```
void Start()
{
    dictationRecognizer = new DictationRecognizer();
    dictationRecognizer.DictationResult += Resolution;
    dictationRecognizer.DictationHypothesis += hypo;
    dictationRecognizer.DictationComplete += completo;
    dictationRecognizer.DictationError += errores;
    controladorjuego.dictStart += comienza;
    controladorjuego.dictStop += termina;
}
```

Botones

- Parar/Encender Reconocedor
- Parar/Encender Dictado

```
1 reference
void Start()
{
    GetComponent<Button>().onClick.AddListener(manage);
}

1 reference
void manage() {
    if (!pressed) {
        controladorjuego.dictStart();
        pressed = true;
    } else {
        controladorjuego.dictStop();
        pressed = false;
    }
}
```

```
1 reference
void manage() {
    if (!press) {
        Debug.Log("Reconoce");
        controladorjuego.comenzar();
        press = true;
    } else {
        Debug.Log("Deja de reconocer");
        controladorjuego.pause();
        press = false;
    }
}
```

Movimiento

- Cubo -> Aprovechando la práctica anterior pero esta vez en 3D
- Personaje - Asset free - Cambio del código Idlechanger.

```
void Start ()
{
    // 各参照の初期化
    controladorjuego.model += ChangePose;
    anim = GetComponent<Animator> ();
    currentState = anim.GetCurrentAnimatorStateInfo (0);
    previousState = currentState;
    // ランダム判定用関数をスタートする
    StartCoroutine ("RandomChange");
}

// Update is called once per frame
0 references
void Update ()
{
    // ↑キー/スペースが押されたら、ステートを次に送る処理
    if (Input.GetKeyDown ("up") || Input.GetButton ("Jump") || change) {
        // ブーリアンNextをtrueにする
        Debug.Log("Entro");
        anim.SetBool ("Next", true);
        change = false;
    }
}
```

Reproducir música

- Escucha al controlador de juego - Usuario dice bakamitai - Reproduce
- Reproduce solo una vez usando un bool y la función PlayOneShot()
- Es lanzado a través del controlador de juego por el dictado

```
1 reference
void OnPlay() {
    if (!alreadyPlayed) {
        audio.PlayOneShot(SoundToPlay, Volume);
        alreadyPlayed = true;
    }
}
```

Texto reconocido

- Suscribe a la Action del controlador
- Si recibe bakamitai:
cambia texto y pone audio
- Si no, cambia el texto a lo que dictes

```
void Start()
{
    texto.text = "";
    controladorjuego.swearing += Swear;
}
```

1 reference

```
void Swear(string words) {
    if (words == "ばかみたい") {
        texto.text = "Don't swear";
        controladorjuego.playAudio();
        //controladorjuego.playVideo();
    } else {
        texto.text = words;
    }
}
```

```
// Update is called every frame
```

Proyecto final

