

Sistema de diálogos.

Generales

Este repositorio fue creado con el objetivo de hacer que un sistema de diálogo sea funcional, a este mismo se le agregó un script el cual es el encargado de la funcionalidad.

1. Se creo un script llamado "Diálogo"
2. A este se le agrego varias variables

```
public GameObject Admiracion; //El signo de admiración para hscer la alerta.
public GameObject Mouse; //El mouse que explica como activar.
private bool isPlayerInRange; //Esta me va a indicar si el jugador esta cerca del rango.
private bool didDialogueStart; //Nos va a indicar que el dialogo comenzo.
private int lineIndex; //Este nos va a mostrar que linea de dialogo nos esta mostrando.
public float typingTime = 0.05f; //Tiempo de los caracteres.

public GameObject PanelDeDialogo; //Para tener una referencia al panel del dialogo.
public TMP_Text TextoDeDialogo; //Para tener una referencia al texto de la ui.
[SerializeField, TextArea(4, 6)] private string[] dialogueLines; //Aqui vamos a escribir lo que va a salir en el dialogo.
```

3. En el update se agrego el momento en el cual se comienza el dialogo y la manera en como pasar al dialogo siguiente, también se podrá detener el dialogo.

```
void Update()
{
    if(isPlayerInRange && Input.GetButtonDown("Fire1")) //El dialogo va a comenzar cuando el player este en el area y presiona el click
    {
        if(!didDialogueStart)
        {
            StartDialogue();
        }
        else if(TextoDeDialogo.text == dialogueLines[lineIndex]) //Si esta mostrando toda la frase completa pasamos a la siguiente linea
        {
            NextDialogueLine();
        }
        else //Podre detener el dialogo y adelantarlo.
        {
            StopAllCoroutines();
            TextoDeDialogo.text = dialogueLines[lineIndex];
        }
    }
}
```

4. En esta área es donde se activan los paneles para mostrar el texto, los símbolos que activan el dialogo se desactivan mientras el dialogo pase y todo en el juego se detiene hasta que acabe el dialogo.

```
void StartDialogue()
{
    didDialogueStart = true; //Le estamos diciendo que ya comenzo el dialogo.
    PanelDeDialogo.SetActive(true); //Se activa el panel para mostrar el texto.
    Admiracion.SetActive(false); //Se desactiva el signo de admiración cuando esta el dialogo.
    Mouse.SetActive(false); //Se desactiva el mouse cuando esta el dialogo.
    lineIndex = 0; // Para que siempre que reactivemos el dialogo comience de 0.
    Time.timeScale = 0f; //Todo se detiene completamente mientras esta el dialogo.

    StartCoroutine(ShowLine());
}
```

5. Aquí se activa el segundo dialogo al momento de terminar el primero y aquí se sabe si hay mas diálogos o ya finalizo el dialogo (si ya finalizo se activan los símbolos y el tiempo regresa a la normalidad.).

```
private void NextDialogueLine() //Para poder pasar a la siguiente linea de dialogo.
{
    lineIndex++; //Esto me ayuda a pasar a la siguiente linea.
```

```

        if(lineIndex < dialogueLines.Length) //si este es menor quiere decir que aun no llegamos a la ultima linea.
        {
            StartCoroutine(ShowLine());
        }
        else //Si no hay mas lineas el dialogo ah terminado.
        {
            didDialogueStart = false;
            PanelDeDialogo.SetActive(false);
            Admiracion.SetActive(true);//Se activa el signo de admiración.
            Mouse.SetActive(true);//Se activa el mouse.
            Time.timeScale = 1f; // Regresa el juego a la normalidad una vez acabado el dialogo.
        }
    }
}

```

6. Aquí se va a controlar como va a funcionar el dialogo, comenzamos sin texto pero se va a comenzar a escribir letra por letra y el tiempo que toma de una letra a otra.

```

private IEnumerator ShowLine()
{
    TextoDeDialogo.text = string.Empty; //El texto va a comenzar vacio

    foreach(char ch in dialogueLines[lineIndex])//Se va a ir escribiendo caracter por caracter.
    {
        TextoDeDialogo.text += ch;
        yield return new WaitForSecondsRealtime(typingTime); //El tiempo de letra en letra.
    }
}

```

7. Este se activa cuando el jugador este dentro del área del objeto el cual va a tener el dialogo, lo cual va a activar los símbolos que indican el siguiente paso para activar el dialogo.

Dentro de esta área se va a poder activar el dialogo.

Se le dijo al script que solo el objeto que pase sobre el y tenga la etiqueta "Player" podrá activarlo.

```

private void OnTriggerEnter2D(Collider2D collision) //Cuando pasen por el trigger este se va a activar.
{
    if(collision.gameObject.CompareTag("Player")) // Solo el que tenga el tag de player lo puede activar.
    {
        isPlayerInRange = true;
        Admiracion.SetActive(true);//Se activa el simbolo de admiración al momento de estar cerca de la misión.
        Mouse.SetActive(true);//Se activa el mouse al momento de estar cerca de la misión.
    }
}

```

8. Este se desactiva cuando el jugador sale del área del objeto, lo cual desactiva los símbolos que indican el siguiente paso y ya no se puede iniciar el dialogo.

```

private void OnTriggerExit2D(Collider2D collision) //Cuando salga del trigger este se va a desactivar.
{
    if (collision.gameObject.CompareTag("Player")) // Solo el que tenga el tag de player lo puede desactivar
    {
        isPlayerInRange = false;
        Admiracion.SetActive(false);//Se desactiva el simbolo de admiración al momento de alejarse de la misión.
        Mouse.SetActive(false);//Se desactiva el mouse.
    }
}

```

Al momento de tener el código, en el unity se van a calar el panel y el texto a sus cuadros correspondientes.

De igual manera se jalen las imágenes que se vayan a ocupar para que esta actuen como los símbolos que te avisan como sirve el juego.