

Skybox con video
Y cambio de escenario

Para crear un Skybox con video:

- Se debe crear un material nuevo de tipo Skybox/Panoramic.
- Para poder reproducir el video se crea un Render Texture.
- Al Skybox se le asigna como textura el Render Texture.
- Con ayuda de un GameObject vacío se le añade un componente Video Player, en render mode se selecciona Render Texture y en Target Texture ponemos el Render Texture que creamos.
- Para que se vea el Skybox en la cámara principal debemos cambiar la opción de Clear Flags a Skybox.

Para configurar los botones:

- Se busca la opción On_Click() y se le agrega el GameObject que contenga la función que se va a ejecutar cuando se presione el botón.
- En caso de querer usar una función en un script creado por uno mismo se debe declarar esta función como pública para que se pueda visualizar en las opciones.
- Para cambiar el diseño del botón se cambia el Source Image, además los colores que aparecen abajo son cambios en tonalidades para cuando se presiona el botón, o cuando el cursor pasa por encima de este, entre otras opciones.

Para el script:

- Si se desea que el script manipule escenas se debe incluir la biblioteca UnityEngine.SceneManagement.
- La función GetActiveScene() regresa la escena que esta corriendo, si se le añade al final un .buildIndex regresa en forma de int el índice que le pertenece a la escena.
- SceneManager.sceneCountInBuildSettings regresará el número de escenas que están configuradas en el build.
- SceneManager.LoadScene() carga la escena que se le indique, puede ser por nombre o por índice.

Adicionales:

- Revisar jerarquía entre objetos ya que puede ocasionar que el jugador no pueda interactuar con el GUI.
- No mover elementos del GUI en dirección Z porque el jugador tampoco podrá interactuar con ese elemento.
- Es mejor escalar el GUI que el resto de los objetos.
- Para escalar el GUI utilizar el Width y Height ya que alterar la escala también alterará los puntos de interacción.

```

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.SceneManagement;
5
6  public class SceneChange : MonoBehaviour
7  {
8      public int maxScene;
9      public int actualScene;
10
11     private void Start()
12     {
13         actualScene = SceneManager.GetActiveScene().buildIndex;
14         maxScene = SceneManager.sceneCountInBuildSettings - 1;
15     }
16
17     public void PreviousScene()
18     {
19         if (actualScene < 0)
20         {
21             actualScene = maxScene;
22             SceneManager.LoadScene(actualScene);
23         }
24         else
25         {
26             actualScene--;
27             SceneManager.LoadScene(actualScene);
28         }
29     }
30
31     public void NextScene()
32     {
33         if (actualScene > maxScene)
34         {
35             actualScene = 0;
36             SceneManager.LoadScene(actualScene);
37         }
38         else
39         {
40             actualScene++;
41             SceneManager.LoadScene(actualScene);
42         }
43     }
44 }
45
46

```

Código utilizado en la práctica.