

```
1 using System;
2 using UnityEngine;
3
4 /// <summary>
5 /// CursorManager handles the switching of mouse textures within each scene.
6 /// </summary>
7 public class CursorManager : MonoBehaviour
8 {
9     // Singleton state reference
10    private static CursorManager instance;
11
12    /// <summary>
13    /// Texture that is utilized when the cursor is in use.
14    /// </summary>
15    [SerializeField]
16    Texture2D cursorTexture;
17
18    /// <summary>
19    /// Texture that is utilized when the link is in use.
20    /// </summary>
21    [SerializeField]
22    Texture2D linkTexture;
23
24    /// <summary>
25    /// The pixel position the cursor texture is centered around.
26    /// </summary>
27    private static Vector2 cursorPosition = Vector2.zero;
28
29    /// <summary>
30    /// The pixel position the link texture is centered around.
31    /// </summary>
32    private static Vector2 linkPosition = new Vector2(12, 0);
33
34    // Enforces a singleton state pattern
35    private void Awake()
36    {
37        if (instance != null)
38        {
39            Destroy(this);
40            throw new Exception("CursorManager instance already established; terminating.");
41        }
42
43        instance = this;
44    }
45
46    /// <summary>
47    /// Sets a new mouse texture based on the specified value.
```

```
48     /// </summary>
49     /// <param name="useCursorTexture">Whether the cursor texture should be ↗
        used.</param>
50     public static void SetMouseTexture(bool useCursorTexture)
51     {
52         if (useCursorTexture) Cursor.SetCursor(instance.cursorTexture, ↗
            cursorPosition, CursorMode.ForceSoftware);
53
54         else Cursor.SetCursor(instance.linkTexture, linkPosition, ↗
            CursorMode.ForceSoftware);
55     }
56 }
```