

Documento Técnico

Este documento técnico describe la implementación interna del plugin de ambientación emocional desarrollado para Unity. Su objetivo es gestionar la iluminación de un entorno de juego según combinaciones de emociones, utilizando el modelo de Plutchik como base teórica. Este documento está dirigido a desarrolladores que deseen integrar, modificar o extender la funcionalidad del plugin.

1. Requisitos del sistema

- **Motor:** Unity 2022.3.4f1
- **Lenguaje:** C#
- **Editor:** Visual Studio 2022
- **Sistema operativo:** Windows 10 o superior
- **Memoria RAM recomendada:** 8 GB mínimo
- **Otros:** Permisos de escritura en disco para persistencia JSON

2. Arquitectura del sistema

El sistema sigue una arquitectura modular basada en el patrón *Controlador-Vista-Modelo* (MVC).

- **Controlador:** LightingController.cs
- **Vista:** Interfaz construida sobre un Canvas en Unity
- **Modelo:** Estructuras de datos EmotionProfile, diccionarios de emociones y archivos JSON de configuración

3. Descripción de componentes

- **LightingController:** clase central. Controla los sliders, calcula los colores, guarda/carga configuraciones y aplica los efectos.
- **EmotionalLight:** componente que reciben el color emocional calculado y lo aplican a la iluminación de la escena.
- **Canvas UI:** sliders, botones, rueda emocional y marcador.
- **emotion_profiles.json:** archivo donde se almacenan los perfiles personalizados.
- **settings.json:** archivo que guarda el último estado emocional aplicado.

4. Lógica de funcionamiento

- El usuario ajusta sliders o hace clic en la rueda.
- Se calcula una mezcla ponderada de colores según las intensidades.
- Se actualiza la iluminación de los objetos con EmotionalLight.
- Se guarda automáticamente el estado.
- También es posible activar perfiles emocionalmente preconfigurados desde scripts.

5. Interacción con el juego

Los desarrolladores pueden activar perfiles emocionales directamente desde eventos del juego usando `lightingController.ActivateProfile("nombrePerfil");`

Esto permite cambios contextuales de ambientación durante el juego (zonas tristes, eventos felices, etc.).

6. Persistencia de datos

- Guardado automático en `settings.json` tras cada cambio.
- Guardado manual por el usuario en `emotion_profiles.json`.
- Los archivos son accesibles desde `Application.persistentDataPath`.

7. Posibilidades de extensión

- Integración con audio, clima o música.
- Detección automática de emociones mediante sensores o decisiones del jugador.
- Expansión a emociones terciarias o personalizadas.

8. Mantenimiento y recomendaciones

- No modificar directamente los archivos JSON si no se entiende su estructura.
- Usar nombres únicos y descriptivos para los perfiles.
- Asegurar que todos los objetos de iluminación incluyan el componente EmotionalLight.