

The Sleepless - GDD Sintetico

Core Concept

Survival psicologico sottrattivo: la perdita di controllo è il tema centrale. La follia del protagonista impedisce il sonno, necessario per recuperare energia. Senza energia, le azioni essenziali per mantenere la barca sono impossibili.



Game Overview

Game Sheet: Survival narrativo, single-player, durata stimata 1–2 ore, piattaforma: PC.

Game Structure: Introduzione del protagonista e routine di cura della barca; progressiva perdita della possibilità di dormire; cutscenes che segnano i passaggi critici.

Game Loop:

- (1) Dormire per recuperare energia;
- (2) usare energia rimanente per manutenzione;
- (3) la barca degrada se le azioni di manutenzione non sono eseguite;
- (4) Elementi di trama impediranno di dormire.

Game Pillars: Sottrazione, Perdita di controllo, Resonance (coerenza tematica), Atmosfera immersiva.

Game Structure & Architecture

Struttura modulare con **manager centralizzati** (Cutscene Manager, UI Manager, Game State Manager).

Ogni manager è un **Singleton** per garantire un unico punto di accesso e gestione.

Comunicazione tra sistemi tramite **eventi** (UnityEvents / C# Events) per ridurre le dipendenze dirette.

Gestione stati di gioco

Game State Manager: controlla i macro-stati (Lucido, Paranoia, Deprivazione, Collasso).

Utilizzo di un approccio a **Finite State Machine (FSM)** semplice, centralizzata nel manager, senza moltiplicare logiche sui singoli oggetti.

Ogni stato abilita/disabilita blocchi di input e interazioni.

Cutscenes: Sistema modulare basato su **Timeline** e **Animator**.

Trigger gestiti dal **Cutscene Manager** per garantire sincronizzazione con gameplay.

Uso di **DOTween** per transizioni UI e effetti di interfaccia.

Interfaccia utente (UI): Architettura a Canvas singolo con pannelli modulari attivati/disattivati.

Feedback chiaro tramite **DOTween** (fade, scaling, transizioni).

Separazione logica tra presentazione (UI Manager) e logica (Game State Manager).

IA & Navigazione: Elementi interattivi e narrativi minori gestiti con **NavMeshAgent** (se introdotti per simulazioni di movimento).

ProBuilder per prototipazione rapida delle scene e adattamento della NavMesh.

Persistenza e dati: Gestione leggera degli stati di partita (nessun salvataggio avanzato richiesto vista la durata del gioco).

Uso di **ScriptableObject** per configurazioni (es. soglie energia, parametri UI, velocità degrado).

Sfide tecniche

Dare feedback chiaro quando un'azione non è disponibile (UI, input bloccati).

Sincronizzare il collasso progressivo del gameplay con le cutscenes senza introdurre incoerenze.

Garantire che la struttura a manager non diventi un collo di bottiglia, mantenendo il codice modulare e testabile.