Institut Puig Castellar
Projecte / Crèdit de síntesi
Document funcional

DOCUMENTO FUNCIONAL DEL PROYECTO

ALUMNO/GRUPO: Juan José Alonso Romero

---

1. Introducción y contexto

Objetivo: Se trata de un videojuego 3D de terror psicológico con estética PSX.

Describe:

- El problema o necesidad que se quiere resolver.

En el mercado actual de videojuegos indie existe una tendencia creciente hacia el terror psicológico con estética retro. Sin embargo, la mayoría de propuestas son o demasiado simples en su narrativa o demasiado complejas técnicamente para un solo desarrollador. Existe una necesidad de demostrar que un único alumno puede crear una experiencia de terror inmersiva, con narrativa ambiental coherente y sistemas funcionales, usando herramientas gratuitas como Godot Engine.

- Quién será el usuario o cliente final.

El juego está dirigido a jugadores de entre 15 y 35 años, aficionados al género de terror indie y al PSX horror. No requiere conocimientos técnicos previos ni habilidades especiales, sólo curiosidad y tolerancia al miedo sutil y psicológico.

- Qué solución se propone y con qué propósito.

Se propone el desarrollo y publicación de They Left Notes, un videojuego de terror en primera persona ambientado en un bosque nocturno aislado. El jugador aparece junto a un coche averiado y debe explorar el entorno, recoger notas que cuentan una historia fragmentada y encontrar una garrafa de gasolina para escapar. La tensión se genera mediante la ambientación, el audio ambiental aleatorio, un susto controlado con una sombra y una narrativa que revela que el jugador está atrapado en un bucle del que nunca puede escapar.

2. Análisis de requisitos

2.1. Requisitos funcionales (RF)

Qué debe hacer el sistema. Enumera las funciones principales, numeradas como RF1, RF2, etc.

Código	Descripción del requisito funcional
RF1	El juego permite al jugador moverse libremente por el entorno 3D del bosque con WASD y controlar la cámara con el ratón.
RF2	El jugador puede interactuar con objetos del entorno (notas, garrafa de gasolina, coche) pulsando la tecla I.
RF3	Al interactuar con una nota, se muestra su contenido en pantalla y el juego pausa mientras se lee.
RF4	El jugador puede teletransportarse entre el exterior e interior de la cabaña pulsando la tecla F.
RF5	Pasado un tiempo aleatorio (30-60 segundos), una sombra aparece frente al jugador bloqueando temporalmente el movimiento y la cámara.
RF6	El juego reproduce audio ambiental aleatorio (pasos, ramas, pájaros, chase) a intervalos variables para generar tensión.
RF7	Al recoger la garrafa de gasolina y volver al coche pulsando I, se activa la pantalla final con fade a negro, texto narrativo y cierre automático.
RF8	El juego aplica un shader PSX que simula la estética visual de PlayStation 1 en toda la pantalla.
RF9	Al inicio de la partida se muestran diálogos del personaje que contextualizan la situación narrativa.

2.2. Requisitos no funcionales (RNF)

Cómo debe comportarse el sistema. Incluye aspectos como rendimiento, seguridad, compatibilidad o facilidad de uso.

Código	Descripción del requisito no funcional
RNF1	El juego funciona a un mínimo de 30 FPS en hardware de gama media (Intel i5, 8 GB RAM, GPU integrada).
RNF2	El juego es compatible con Linux. No se garantiza compatibilidad con móviles ni consolas.
RNF3	El código está modularizado en escenas independientes para facilitar el mantenimiento y futuras ampliaciones.
RNF4	El juego no sufre errores críticos (crashes) durante la jugabilidad normal.
RNF5	La interfaz es intuitiva y todas las acciones principales están indicadas en el

	propio juego.

2.3. Restricciones

Condiciones o limitaciones del proyecto.

- Lenguajes o tecnologías obligatorias: GDScript, Godot Engine 4.x, GitHub, HTML/CSS/JS para la página web del estudio, Itch.io para la publicación.
- Recursos disponibles: 6-7 meses de desarrollo, un ordenador con Godot instalado, repositorio en GitHub y acceso a plataformas de assets gratuitos (Sketchfab, Itch.io, Freesound.org).
- Dependencias técnicas: Godot Engine 4.x (el juego depende directamente de esta versión). Assets externos gratuitos de Itch.io y Sketchfab bajo licencias CC0/CC-BY.
- Limitaciones técnicas: El juego está diseñado y optimizado exclusivamente para Linux. No se garantiza su correcto funcionamiento en Windows, macOS, móviles o consolas. Es un juego individual sin soporte multijugador.

3. Análisis de usuarios y roles

Objetivo: identificar quién usará el sistema y qué podrá hacer.

Rol	Descripción	Permisos principales
Desarrollador	Juan José Alonso Romero, responsable del desarrollo, mantenimiento y publicación del juego.	Puede modificar el código, los assets, la narrativa, exportar nuevas versiones y gestionar la página de Itch.io.
Jugador	Cualquier persona que descarga y ejecuta el juego sin necesidad de registro ni cuenta.	Puede jugar la partida completa, leer las notas, explorar el entorno e interactuar con todos los objetos.
Tester	Persona encargada de probar el juego para detectar errores o desequilibrios en la dificultad o la narrativa.	Puede jugar todas las zonas en cualquier estado del desarrollo y reportar fallos, pero no puede modificar nada.

4. Casos de uso / Escenarios de uso

Objetivo: mostrar cómo interactúan los usuarios con el sistema.

Código	Nombre del caso de uso	Actor principal	Descripción	Resultado esperado
CU1	Iniciar partida	Jugador	El jugador abre el juego, ve los diálogos iniciales y empieza a explorar el bosque.	El juego carga la escena principal con el jugador junto al coche averiado en el bosque.
CU2	Recoger una nota	Jugador	El jugador se acerca a una nota, pulsa I para inspeccionarla y vuelve a pulsar I para guardarla.	La nota desaparece del mundo, su texto queda guardado y el jugador recupera el control.
CU3	Entrar en la cabaña	Jugador	El jugador se acerca a la puerta de la cabaña abierta y pulsa F.	El jugador se teletransporta al interior de la cabaña donde puede encontrar la garrafa de gasolina.
CU4	Susto con la sombra	Sistema	Pasado un tiempo aleatorio, la sombra aparece frente al jugador con un sonido de susto.	El jugador pierde el control durante 2 segundos mientras la cámara gira hacia la sombra. Luego recupera el control.
CU5	Activar final del juego	Jugador	El jugador recoge la garrafa de gasolina, vuelve al coche y pulsa I.	La pantalla se funde a negro, aparece el texto final del bucle narrativo y el juego se cierra.

5. Modelo de datos o estructura de la información

Entidades principales:

Jugador: guarda el estado actual del jugador durante la partida. Incluye su posición en el mundo, si la linterna está encendida o apagada, si el susto ya ha ocurrido y si la garrafa de gasolina ha sido recogida.

Nota: representa cada documento coleccionable repartido por el mapa. Tiene una posición fija en el mundo, un texto narrativo asociado y un estado (recogida o no recogida). Cada nota es una instancia independiente de la escena nota.tscn.

Garrafa de gasolina: objeto interactivo único que constituye el objetivo principal del juego. Tiene una posición fija en el interior de la cabaña y un estado (recogida o no recogida). Al ser recogida activa el objetivo final.

Sombra: entidad invisible por defecto que se activa pasado un tiempo aleatorio. Tiene una posición calculada en tiempo real (siempre delante del jugador) y un estado (activa o inactiva).

PantallaFinal: entidad UI (CanvasLayer) que contiene el ColorRect para el fade a negro y el Label con el texto narrativo final. Inactiva hasta que se cumplen las condiciones del final del juego.

Relaciones:

- El jugador interactúa con las notas para recogerlas y leerlas.
- El jugador interactúa con la garrafa de gasolina para recogerla.
- El jugador interactúa con el coche sólo si la garrafa está recogida, activando la secuencia final.
- La sombra interactúa con el jugador bloqueando temporalmente sus controles.

6. Diseño de la interfaz

Pantallas principales del juego:

- Menú principal

Función: Pantalla de inicio con título del juego y tres opciones: Play, Options y Exit. Estética oscura con fuente de terror coherente con la identidad visual del juego.

- Pantalla de juego (partida en curso)

Función: Pantalla principal donde transcurre la acción. Muestra el mundo 3D en primera persona con el shader PSX activo, la linterna como única fuente de luz y el entorno del bosque. No hay HUD permanente para mantener la inmersión.

- Pantalla de lectura de notas

Función: Al recoger una nota aparece un CanvasLayer con el texto de la nota sobre fondo oscuro semitransparente. El jugador puede leerla con calma y pulsar I para cerrarla y continuar.

- Pantalla de diálogo inicial

Función: Al iniciar la partida se muestran los diálogos del personaje con efecto glitch que contextualizan la situación. Casos de uso relacionados: CU1.

- Pantalla final (fade a negro)

Función: Al activar el final del juego, la pantalla se funde a negro en 2 segundos, aparece el texto narrativo del bucle y tras 5 segundos el juego se cierra. Casos de uso relacionados: CU5.

7. Planificación técnica

Lenguajes y frameworks:

- Godot Engine 4.x (GDScript): desarrollo del juego y su lógica principal.
- HTML, CSS y JavaScript: página web del estudio Night Tape Studio.
- GitHub: control de versiones y almacenamiento del proyecto.
- Itch.io: plataforma de publicación del juego.

Herramientas de diseño o edición:

- Blender: ajustes menores en modelos 3D (rotaciones, escalas).
- Audacity: edición básica de efectos de sonido.
- Sketchfab e Itch.io: fuentes de assets 3D bajo licencias CC0/CC-BY.
- Freesound.org: efectos de sonido libres.

Fases del proyecto:

Fase	Descripción	Duración estimada
Octubre - Noviembre	Configuración de Godot, menú principal, movimiento del jugador, assets del bosque y cabañas.	2 meses
Diciembre - Enero	Linterna, diálogos, sistema de notas, teletransporte y recuperación de bugs.	2 meses
Febrero - Marzo	Audio ambiental, sistema de susto, final del juego, shader PSX, página web y publicación en Itch.io.	2 meses
Marzo - Mayo	Redacción de la memoria, preparación de la presentación y entrega final.	2 meses

8. Análisis de riesgos

8.1. Identificación de riesgos

Posibles riesgos:

- Bugs graves que rompen el proyecto: ya ocurrió al implementar el teletransporte, lo que obligó a recuperar una versión anterior desde GitHub.
- Funcionalidades que no se pueden implementar por falta de tutoriales o documentación (ej: lluvia con partículas en Godot 4).
- Pérdida del progreso del proyecto o archivos dañados.
- Problemas de exportación y compatibilidad con plataformas de publicación (Itch.io web).

8.2. Valoración y respuesta

Riesgo	Probabilidad	Impacto	Plan de prevención o contingencia
Bugs graves en el proyecto	Media	Alta	Uso de GitHub para guardar versiones estables. Recuperar la última versión funcional si ocurre un bug.
Funcionalidad no implementable	Media	Baja	Descartar la funcionalidad si no es primordial para el juego y continuar con el resto del desarrollo.
Pérdida de datos	Baja	Alta	Uso de GitHub con commits frecuentes y copias de seguridad en la nube.
Problemas de exportación web	Alta	Media	Exportar para Linux como alternativa si la exportación web no funciona en Itch.io.
Falta de tiempo	Alta	Alta	Priorizar las funcionalidades esenciales y dejar las opcionales para futuras versiones.

9. Validación y criterios de éxito

Objetivo: definir cómo sabremos que el proyecto funciona correctamente.

Criterios de aceptación:

1. El juego se ejecuta correctamente en Linux sin errores de inicio.

2. El jugador puede moverse, usar la linterna, leer notas e interactuar con objetos sin errores ni comportamientos inesperados.

3. El teletransporte (F) entre exterior e interior de la cabaña funciona en ambas direcciones.

4. El sistema de susto se activa correctamente pasado el tiempo aleatorio, posicionando la sombra siempre frente al jugador.

5. El audio ambiental (pasos, ramas, pájaros, chase) suena a intervalos aleatorios y se pausa correctamente durante el susto.

6. Al recoger la garrafa y volver al coche, la pantalla final se activa con el fade a negro y el texto narrativo correcto.

Pruebas previstas:

- Pruebas funcionales: comprobar que todos los sistemas del juego funcionan correctamente (movimiento, notas, teletransporte, susto, audio, final).
- Pruebas de jugabilidad: sesiones de prueba con compañeros del instituto para evaluar la atmósfera, la tensión y la experiencia general.
- Pruebas de portabilidad: ejecutar el juego en diferentes equipos con Linux para asegurar compatibilidad.

Indicadores de calidad:

- Juego fluido a 30 FPS mínimos.
- Atmósfera de terror inmersiva y coherente con la estética PSX.
- Narrativa comprensible a través de las notas del juego.
- Tensión psicológica generada sin necesidad de jumpscares directos.

10. Conclusión

Objetivo: cerrar el análisis y preparar la siguiente fase.

Durante esta fase del proyecto se han definido los aspectos más importantes de They Left Notes:

- Funciones principales: movimiento del jugador, exploración del bosque, recogida de notas, teletransporte, sistema de susto, audio ambiental aleatorio y final del juego con narrativa.
- Tecnologías utilizadas: Godot Engine 4.x con GDScript, GitHub, HTML/CSS/JS para la página web e Itch.io para la publicación.
- Valor del proyecto: crear un videojuego de terror psicológico inmersivo con estética PSX que demuestre que un solo alumno puede desarrollar un proyecto completo y publicarlo con herramientas gratuitas.

Próximos pasos:

- Publicación definitiva en Itch.io con la build final.
- Presentación oral del proyecto ante el tribunal.
- Entrega de toda la documentación (memoria, seguimiento, diagrama de Gantt y documento funcional).

Reflexión final:

El análisis funcional ha servido para tener una visión clara y estructurada de todos los sistemas implementados en They Left Notes. El proyecto ha demostrado que es posible crear una experiencia de terror psicológico inmersiva con recursos limitados, un motor gratuito y trabajo individual constante. A partir de aquí, el objetivo es presentar el juego y la documentación de forma profesional.