



UNIVERSITAT
POLITÀCNICA
DE VALÈNCIA



Escola Tècnica
Superior d'Enginyeria
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

D.A.N.G.E.R – Disaster Prevention

Desarrollo de niveles para la gestión y evaluación
de emergencia en Unity 3D

Trabajo Fin de Grado

Grado en Ingeniería Informática

Autor: Xinyu Jiang

Tutor: Ramón Pascual Mollá Vayá

Segundo Tutor: Teresa María Pellicer Armiñana

Curso 2021-2022

Resumen

Videojuego de simulación social y estrategia en Unity 3D. El proyecto parte del trabajo realizado por dos alumnos en el curso 2020-2021.

Actualmente el juego cuenta con dos modos del juego:

El primer modo es el modo de la construcción, donde te permite construir salas de acuerdo con las normativas.

El segundo modo es el modo de la estrategia, el usuario debe guiar a todos los personajes de la sala a salir de la situación de emergencia.

Y vamos a añadir el tercer modo del juego, en este modo los usuarios solo tienen control de un personaje y tiene que tomar muchas decisiones y ayudar a otros personajes no controlables a escapar del incendio.

Esta vez vamos a cooperar con una estudiante de Máster de Prevención de Riesgos Laborales. Procederemos juntos a realizar tareas como: la ampliación de interfaz de usuario, el diseño de los nuevos niveles, el diseño de nuevas funcionalidades, el análisis del resultado del juego y más cambios u otros complementos que consideremos necesario para complementar el juego. Todos nuestros cambios serán detalladamente documentados en el documento de diseño del juego.

El objetivo principal de este juego es aprender las medidas y comportamientos que deben tomarse en caso de emergencia en un entorno interactivo, seguro y divertido.

Palabras clave: Videojuego, programación, gamificación, prevención de riesgo, Unity 3D.

Summary

Social simulation and strategy videogame made with Unity 3D. The project is based on the work done by two students in the 2020-2021 academic year.

Currently the game has two game modes:

The first mode is the construction mode, where it allows you to build rooms according to the regulations.

The second mode is the strategy mode, users must guide all the characters in the room to get out of the emergency.

And we are going to add the third mode of the game, in this mode the users only have control of one character, and he must make a lot of decisions and help other non-playable characters to escape from the fire.

This time we are going to cooperate with a master's student in Occupational Risk Prevention. We will proceed together to do tasks such as: the extension of the user interface, the design of the new levels, the design of new functionalities, the analysis of the result of the game and more changes or other complements that we consider necessary to complement the game. All our changes will be clearly documented in the game design document.

The main objective of this game is to learn the measures and behaviors that must be taken in an emergency in an interactive, safe and fun environment.

Keywords: Video game, programming, gamification, risk prevention, Unity 3D.

Resum

Videojoc de simulació social i estratègia a Unity 3D. El projecte parteix del treball realitzat per dos alumnes durant el curs 2020-2021.

Actualment el joc compta amb dues maneres del joc:

La primera manera és la manera de la construcció, on us permet construir sales d'acord amb les normatives.

La segona manera és la manera de l'estratègia, l'usuari ha de guiar a tots els personatges de la sala a sortir de la situació d'emergència.

I afegirem el tercer mode del joc, en aquest mode els usuaris només tenen control d'un personatge i han de prendre moltes decisions i ajudar altres personatges no controlables a escapar de l'incendi.

Aquesta vegada cooperarem amb una estudiant de Màster de Prevenció de Riscos Laborals. Procedirem junts a realitzar tasques com: l'ampliació d'interfície d'usuari, el disseny dels nous nivells, el disseny de noves funcionalitats, l'anàlisi del resultat del joc i més canvis o altres complements que considerem necessaris per complementar el joc. Tots els nostres canvis seran detalladament documentats al document de disseny del joc.

L'objectiu principal d'aquest joc és aprendre les mesures i els comportaments que cal prendre en cas d'emergència en un entorn interactiu, segur i divertit.

Paraules clau: Videojoc, programació, gamificació, prevenció de risc, Unity 3D.

Tabla de contenidos

1.	Introducción	12
1.1	Contexto	12
1.2	Motivación	13
1.3	Objetivos	14
2.	Estado del arte	15
2.1	Juego y su relación con la educación	15
2.2	Gamificación	17
2.2.1	Ejemplos de la gamificación	18
2.3	Análisis del mercado	22
3.	Herramientas	27
3.1	Introducción	27
3.2	Unity (Motor de videojuego)	27
3.3	Visual Studio (Entorno de desarrollo integrado)	29
3.4	Github (Controlador de versión)	30
3.5	Angular (Framework para la página web)	32
4.	Diseño	33
4.1	Introducción	33
4.2	Análisis de posibles mejoras	36
4.3	Diseño de las mejoras	36
4.3.1	Problema 1: No da pista de cómo jugar.	37
4.3.2	Problema 2: Pocas retroalimentaciones	37
4.3.3	Problema 3: Poco interactivo	38
4.3.4	Problema 4: Página web difícil de lanzar.....	39
4.3.5	Problema 5: Página web redundante	39
4.3.6	Problema 6: Documento de diseño impreciso.....	40

5.	Implementación.....	41
5.1	Introducción	41
5.2	Implementación de las mejoras.....	42
5.2.1	Sistema de tutorial integrado.....	42
5.2.2	Mejora de las retroalimentaciones	43
5.2.3	Nuevo nivel.....	47
5.2.4	Sistema de bonificación y penalización	48
5.2.5	Sistema de Inventarios	49
5.2.6	Sistema de visualización de temperaturas.....	50
5.2.7	Sistema de las preguntas	54
5.2.8	Sistema del tiempo límite.....	57
5.2.9	Página web	58
6.	Conclusiones	61
6.1	Comparación con la última versión (changelog)	61
6.2	Relación con los estudios.....	61
6.3	Conclusiones y trabajos futuros	62
7.	Agradecimientos	64
8.	Bibliografía	65
	Anexo I: Glosario	66
	Anexo II: Documento del diseño del juego.....	67
	SECCIÓN 1: DATOS PRINCIPALES	67
	SECCION 2: Pitch doc.....	68
	SECCIÓN 3: Análisis competitivo.....	70
	SECCIÓN 4: Estructura narrativa, personajes	74
	SECCIÓN 5: Escenas, Niveles.....	76
	SECCIÓN 6 Assets:	81
	SECCIÓN 7: Controles	90
	SECCIÓN 8: Interfaz	91
	SECCIÓN 9: Herramientas complementarias	97
	Anexo III: Objetivo de desarrollo sostenible.....	98
	OBJETIVOS DE DESARROLLO SOSTENIBLE	98



Tabla de figuras

Figura 1. Intervenciones por incendio o explosión (VÍCTIMAS DE INCENDIOS EN ESPAÑA EN 2019, Fundación MAPFRE).....	13
Figura 2. Evolución del número de víctimas mortales (VÍCTIMAS DE INCENDIOS EN ESPAÑA EN 2019, Fundación MAPFRE).....	13
Figura 3. El avatar pisando sobre el suelo de fuego.	15
Figura 4. Consola Elinor, permite interacciona mediante dos mandos amarillos, de dicha forma todos los movimientos serán leídos por la consola.	16
Figura 5. Cinco procesos de la gamificación.....	17
Figura 6. Flood Resilience Game.	19
Figura 7. The Period Game.....	19
Figura 8. Saturo Itawa, el cuarto presidente de Nintendo (Izquierda) y Eikō Harada el director de McDonald’s Japón (derecha).....	20
Figura 9. Underground.	21
Figura 10. Ring Fit Adventure.....	22
Figura 11. The Great Fire of London 1666.	23
Figura 12. Flame Over.....	24
Figura 13. Ludus.....	25
Figura 14. Pokémon Diamante Brillante y Pokémon Perla Reluciente.....	28
Figura 15. Interfaz de Unity.	29
Figura 16. Interfaz de Visual Studio.....	30
Figura 17. Interfaz de Github.	32
Figura 18. Menú de inicio de la versión 2020-2021.....	33
Figura 19. Modo construcción.....	34
Figura 20. Modo estrategia.....	34
Figura 21. Diagrama del flujo del juego versión 2020-2021.....	35
Figura 22. Página web y completa el juego.....	35

Figura 23. Captura de pantalla de Persona 5.	37
Figura 24. Plano de la planta del nivel uno.	38
Figura 25. Diagrama de flujo de la versión 2021-2022.....	41
Figura 26. Menú de inicio de la versión 2020-2021.....	42
Figura 27. Ejemplo de una página de tutoría.....	42
Figura 28. Hacer que un botón atienda a un evento en Unity.	43
Figura 29. Destino del personaje dibujado en la pantalla (Remarcado con el cuadro rojo).	43
Figura 30. Detalle de script <i>3D Target Cursor Marker</i>	44
Figura 31. Notificación que aparece cuando pulsa el botón de alarma de incendio.	45
Figura 32. Detalle del script de colector de información.	46
Figura 33. Página de resultado cuando el jugador activa la alarma de incendio (izquierda) y cuando no activa la alarma (derecha).....	46
Figura 34. La Figura 17 en 3D.	47
Figura 35. Ejemplo de un recurso gratuito que ofrece componentes sueltos para poder construir un laboratorio entero.	47
Figura 36. Notificación que aparece cuando el jugador no respeta la señal de ruta de evacuación.	49
Figura 37. Notificación que aparece cuando encuentra una toalla en la escena....	50
Figura 38. Zona segura (No hay indicador).....	50
Figura 39. Zona alta temperatura (indicador en amarillo, remarcado con cuadro rojo).	51
Figura 40. Zona muy peligrosa (indicador en rojo, remarcado con cuadro rojo)...	51
Figura 41. <i>Tags</i> y <i>colliders</i> que se asigna a zona alta temperatura (izquierda) y zona muy peligrosa (derecha).	52
Figura 42. <i>Collider</i> del fuego (izquierda) y <i>collider</i> del personaje (derecha).	53
Figura 43. Cuando <i>collider</i> del fuego penetra con el <i>collider</i> del personaje.....	53
Figura 44. Cuando se asigna <i>collider</i> y <i>tag</i> de zona caliente para una puerta, fija que aparece el indicador de alta temperatura (remarcado con cuadro rojo).....	54
Figura 45. Ejemplo de una pregunta.....	54
Figura 46. Ejemplo de contestar bien la pregunta, pero recibe penalización.	56
Figura 47. Preguntas con tiempo límite.....	57

Figura 48. Tiempo restante (remarcado con cuadro rojo).	58
Figura 49. Página principal de la web.	59
Figura 50. Repaso de las preguntas sin marcar respuestas.	59
Figura 51. Repaso de las preguntas visualizando las respuestas.	60

Tabla de tablas

Tabla 1. Datos de los cinco sujetos.....	16
Tabla 2. Resultados. *Sujeto C pierde un día de juego debido por el fallo de hardware.	16
Tabla 3. Análisis DAFO de Danger con simulacro (Desde la perspectiva de Danger).	23
Tabla 4. Análisis DAFO de Danger con The Great Fire of London 1666 (Desde la perspectiva de Danger).	24
Tabla 5. análisis DAFO de Danger con Flame Over (Desde la perspectiva de Danger).	25
Tabla 6. Analisis DAFO de Danger con Ludus (Desde la perspectiva de Danger).	26
Tabla 7. Tabla comparative entre Unreal, CryEngine, Godot y Unity.....	27
Tabla 8. Tabla comparativa entre SVN y Git.....	31



1. Introducción

1.1 Contexto

El presente proyecto es una continuación del trabajo *Danger*, que habían desarrollado los compañeros, Adrián Sánchez Lavarias y Pablo Querol Ballester en el curso académico 2020-2021.

Danger es un juego serio que no solamente ofrece la diversión, sino que también es una herramienta simuladora de incendios para que los jugadores puedan aprender técnicas de autoprotección en el caso de incendio sin ponerse en situaciones de riesgo.

Como es un proyecto grande, el juego se construye en varias etapas. Gracias a los esfuerzos de Adrián Sánchez Lavarias y Pablo Querol Ballester, ya han realizado tareas de:

- Análisis de problemas y requisitos.
- El estudio del mercado.
- El diseño del concepto y la jugabilidad.
- La estructura básica del juego.
- El diseño de una página web para complementar el juego.

Por lo tanto, mi trabajo sería analizar las posibles mejoras y añadir más mecanismos para enriquecer la jugabilidad, mejorar la página web y mejorar el documento de diseño del juego.

Además, este año se coopera con Yennifer Ramirez Lopez, una estudiante de máster de Prevención de Riesgos Laborales. Ella trabajará como una consultora técnica que da y evalúa las ideas desde un punto de vista de profesional para que el juego actúe de acuerdo con situaciones reales, teniendo en cuenta las normativas del sector, al tiempo que mantenga su valor educativo.

1.2 Motivación

Según la investigación y la estadística de la Asociación Profesional de Técnicos de Bomberos (APTB) y la fundación de Mutualidad de la Agrupación de Propietarios de Fincas Rústicas de España (Mapfre), durante el año 2019, se han computado un total de 129.522 siniestros por incendio y se registraron 165 muertos¹.

Intervenciones por incendio o explosión

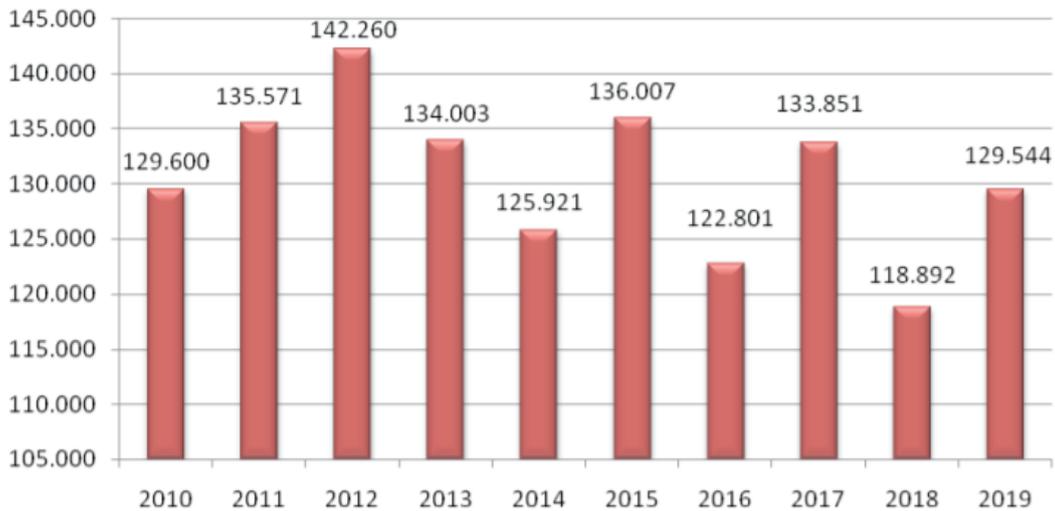


Figura 1. Intervenciones por incendio o explosión (VÍCTIMAS DE INCENDIOS EN ESPAÑA EN 2019, Fundación MAPFRE).

Evolución del número de víctimas mortales

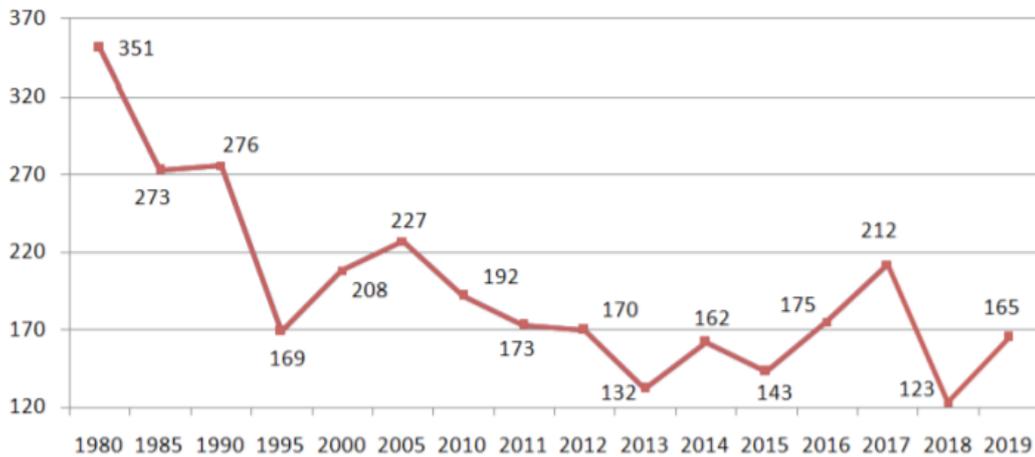


Figura 2. Evolución del número de víctimas mortales (VÍCTIMAS DE INCENDIOS EN ESPAÑA EN 2019, Fundación MAPFRE).

¹ https://www.tecnifuego.org/recursos/arxius/20201217_11062020_Estudio_victimas_incendios_en_2019_APTB_y_MAPFRE.pdf

Por lo tanto, urge la necesidad de crear una herramienta didáctica para poder disminuir el número de incendio y enseñar a la gente cómo actuar en el caso de incendio.

También hay que representarlo en forma de juego, ya que aporta muchas ventajas como ofrecer un entorno seguro, motivar más a los jugadores...y otras características que se explica en las siguientes secciones.

1.3 Objetivos

Los objetivos dentro del videojuego son:

- Hacer una simulación real, donde todos los detalles del juego deben ser creíbles.
- Finalidad didáctica, que todos los jugadores tengan conocimiento sobre gestión de emergencia tras jugar, en consecuente, bajar número de muertes por el incendio.
- Captar interés de los jugadores, ya que aprender es un procedimiento lento y monótono, por lo tanto, el juego será divertido para que, tanto el tiempo para que, tanto el tiempo como la concentración con la que se realice el entrenamiento, aumente y, con ello, mejore el aprendizaje, tanto en calidad como en cantidad.
- Fácil de manejar, intenta hacer que el juego sea fácil de entender y evitar controles complejos ya que va dirigido para cualquier rango de edad.

Por otra parte, se pretende:

- Crear un documento de diseño del videojuego (GDD) claro y conciso para que cualquier persona que continúe este proyecto pueda comprender lo que hay lograr.
- Desarrollar el videojuego con código fuente fácil de entender, bien documentado siguiendo las normativas de desarrollo del software; flexible y extensible para que el sucesor del trabajo cueste lo menos posible para entender los códigos.

2. Estado del arte

2.1 Juego y su relación con la educación

Antes continuar con la implementación el proyecto, hay que parar a pensar: ¿Por qué la herramienta de aprendizaje que busca debe ser un juego, y no otra forma de representación como un libro, una película o cualquier otro soporte?

Para resolver esta duda, en primer lugar, hay que saber cuáles son las características de un juego.

De acuerdo con Chris Crawford “un juego es un sistema formal cerrado que subjetivamente representa un subconjunto de la realidad” (The Art of Computer Game Design, 1997, pág. 8).

A partir de su definición se saca siguientes conclusiones:

- **Un juego es un sistema formal, es decir, tiene reglas explícitas.** Para este juego también se diseña reglas que restringe los jugadores, porque la finalidad es que, obedeciendo estas reglas, los jugadores puedan aprender las técnicas de autoprotección en situaciones de emergencia.
- **Un juego es un subconjunto de la realidad, por lo tanto, ofrece un mundo virtual y de fantasía.** Para la simulación de un incendio es necesario un mundo virtual ya que de esa manera los jugadores no van a ponerse ante la situación del riesgo, pero al mismo tiempo garantiza una experiencia semejante al incendio real.

Por ejemplo, en el juego Danger, cuando el jugador encuentra ante la situación de la Figura 3 una de las reglas es que, si el avatar del jugador pisa sobre el suelo con fuego, pierde el punto de salud y cuando el punto de salud cae a cero, el personaje muere, pero todo eso pasa en un mundo virtual, el jugador no será afectado en ningún modo.



Figura 3. El avatar pisando sobre el fuego.

En segundo lugar, la relación entre el juego y aprendizaje es un tema clásico en el área de psicología cognitiva. Doctora Fran Blumber afirma que el videojuego no es algo independiente de la educación, es más, se puede utilizar para mejorar el aprendizaje de las cosas: Los niños y los adolescentes adquiere y refina su habilidad cognitiva durante el progreso del juego (Learning by Playing: Video Gaming in Education, 2014).

El proyecto Elinor es un caso de estudio interesante (Maiga Chang, Rita Kuo, Kinshuk, Gwo-Dong Chen, & Michitaka Hirose, 2009), que consiste en ofrecer actividades basados en videojuego para rehabilitar la motricidad gruesa.

El grupo objetivo de este proyecto son los pacientes con accidente cerebrovascular, para ello, desarrollaron una consola de juego a medida de los requisitos .



Figura 4. Consola Elinor, permite interacciona mediante dos mandos amarillos, de dicha forma todos los movimientos serán leídos por la consola.

Para el experimento, buscaron cinco pacientes con accidente cerebrovascular.

Sujeto	Edad	Sexo	Hemiparesias	Experiencia con computador	Experiencia con juego de computador
A	64	Femenino	Derecho	Poco	Poco
B	71	Femenino	Izquierdo	Nada	Nada
C	64	Femenino	Izquierdo	Poco	Poco
D	52	Femenino	Izquierdo	Poco	Moderado
E	7	Masculino	Izquierdo	Nada	Nada

Tabla 1. Datos de los cinco sujetos.

El grupo de investigación no dieron recomendación de qué jugar, cuándo juega o qué candidatas de juegos hay que jugar, todo depende del incentivo de los sujetos. Los expertos médicos estimaron una cantidad de entrenamiento de quince minutos al día y cinco días a la semana, en total 375 minutos, sin embargo, el resultado de la Tabla 2 muestra que los sujetos gastaron de media 1324 minutos jugando, que son casi triple de la expectativa.

Sujeto	Tiempos de juego (minutos)	Sesiones	Días
A	703	40	29/35
B	2187	61	31/35
C	1866	88	33/35
D	1175	59	26/35
E	691	38	18/35

Tabla 2. Resultados. *Sujeto C pierde un día de juego debido por el fallo de hardware.

En cuando las retroalimentaciones, son mayormente positivos, les parecieron desafiante e inmersivo. Además, los sujetos nunca quejaron por falta de instrucción, el aprendizaje de los juegos parece relativamente fluidos y las habilidades desarrollaron inconscientemente.

La conclusión para este experimento es que, cuatro de cada cinco pacientes han tenido buena experiencia. El efecto de rehabilitación no está dentro del alcance de este proyecto, pero cabe destacar que un fisioterapeuta reveló que su paciente ha tenido una mejora significativa tanto el movimiento como el uso del brazo afectado.

Con estos dos análisis podemos afirmar que el juego es una herramienta potente para hacer una simulación de una situación de emergencia, además, los jugadores reflexionan y aprenden nuevas habilidades inconscientemente y a veces el propio desafío del juego motiva los jugadores a aprender, como demostraba el proyecto Elinor. En el siguiente punto se presenta cómo se llama la técnica de convertir una simulación en un juego, y cuáles son sus ventajas.

2.2 Gamificación

La gamificación consiste en ofrecer una experiencia similar al juego o utilizar elementos de juego con el fin de proporcionar una diversión. Exististe unas diferentes interpretaciones sobre ella, por ejemplo:

- Diseño de un software en un contexto no relacionado con el juego utilizando mecanismos del juego (Kevin Werbach & Dan Hunter, 2012): Como los sistemas de puntos, logros, ranking...etc. Estos tipos de software suelen tener un diseño antropométrico, es decir, son orientados para despertar interés de sus usuarios y dando les más motivación para seguir utilizando su software.
- Resolver problemas sociales (Gómez Tringueros & Ortega Sánchez, 2019): Diseño de juegos que enfoca en problemas que la gente normalmente no se toma en serio y convierte las situaciones no jugables en un conjunto de mecánicas jugables.

Aunque cada una de las definiciones se interpreta la gamificación desde un punto de vista diferente, pero el núcleo de la idea es captar interés de la gente con el entretenimiento, porque la diversión es la forma más fácil de cambiar el comportamiento de las personas para ser mejor (Volkswagen, 2009).

En cuando el proceso de la gamificación, Wendy Hsin-Yuan Huang y Dilip Soman propusieron cinco pasos para ludificar un programa para el sistema educativo (A Practitioner's Guide To Gamification Of Education, 2013, pág. 7):

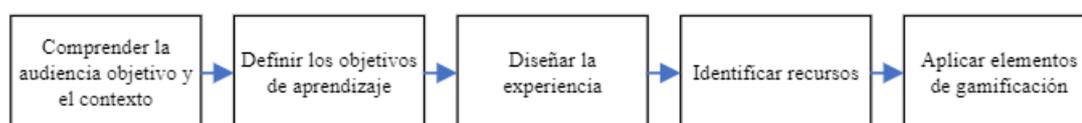


Figura 5. Cinco procesos de la gamificación.

Lo cual, cada uno de los procesos representan:

- Comprender la audiencia objetivo y el contexto: ¿A quién o quiénes va dirigido? Estos análisis determinarán factores como grupo de edad, nivel de enseñanza, el tamaño de grupo...etcétera y facilitará a diseñar un programa que capacite al estudiante.

- Definir los objetivos de aprendizaje: ¿Qué quiere que logre los estudiantes al finalizar el programa? El éxito depende cómo definir claramente el objetivo de aprendizaje que subyacen al programa educativo.
- Diseñar la experiencia: ¿Cómo se puede dividir el programa de aprendizaje y cuáles son los puntos débiles? El diseño de hitos hace que el objetivo final parezca más alcanzable y medible, asimismo facilita la identificación de obstáculos de cada etapa.
- Identificar recursos: ¿Qué recursos necesitan para ludificar la enseñanza? Teniendo los hitos identificados, resulta mucho más fácil determinar los recursos necesarios.
- Aplicar elementos de gamificación: ¿Qué elementos de la gamificación hay que aplicar? El efecto de la enseñanza y la reacción de estudiantes dependen de técnicas aplicadas, y el objetivo principal siempre es motivar los estudiantes.

El uso de la gamificación en el sistema educativo conlleva muchas ventajas, Wendy Hsin-Yuan Huang y Dilip Soman estudiaron casos reales de diferentes escuelas y sacaron siguientes conclusiones:

- **La enseñanza con el juego es más motivante**, ya que minimiza las emociones negativas que suele encontrar en las formas tradicionales de educación, como el factor de vergüenza por fallo que se encuentra en la educación en el aula.
- **El juego proporciona retroalimentaciones a tiempo real**, no como el sistema educativo tradicional, que, por general, reciben su calificación después de una fecha de vencimiento (por ejemplo, después de los exámenes).
- **El juego motiva a los jugadores a aplicar conocimientos justo después de la enseñanza**, eso ayuda mucho a los estudiantes que tienen dificultades para recordar las lecciones.
- **El juego incide la competición**. Los jugadores puede estar motivado por el aprendizaje por elementos sociales como la tabla de la calificación y el sistema de los logros.

2.2.1 Ejemplos de la gamificación

La gamificación no solamente se encuentra en el área de videojuego, sino que se usa en el juego en sentido más genérico, es decir, tanto juego con herramienta como juego sin herramienta, estos son unos ejemplos.

- **Flood Resilience Game² (El juego de resistencia a las inundaciones)** es un juego de mesa creado por unión de varias organizaciones sanitarias. Fue diseñado para resolver problema de inundaciones en Indonesia y Perú. En este juego, los jugadores tienen que seleccionar sus roles entre: miembros de la comunidad, gobierno local y funcionarios de la junta de aguas. Durante el juego los jugadores experimentarán qué prevenciones hay que hacer antes de la inundación, cómo distribuir recursos, cómo actuar durante el siniestro, y cómo reorganizar la sociedad tras el desastre.

² <https://floodresilience.socialsimulations.org/>



Figura 6. Flood Resilience Game.

- **The Period Game³ (El juego del período)** es un juego de mesa diseñado para enseñar el conocimiento de la menstruación de una manera divertida y atractiva. El diseñador pretende resolver el problema de falta de conocimiento sobre salud sexual de los adolescentes. Durante el juego, los jugadores aprenderán cómo tratar con los problemas del período menstrual, así como las enfermedades que puede sufrir.



Figura 7. The Period Game.

- **McDonald's Training Game⁴ (El juego de entrenamiento de McDonald's)** es un videojuego educativo creado por McDonald's y publicado únicamente en Japón en

³ <https://www.periodgame.com/store/periodgame>

⁴ <https://gizmodo.com/mcdonalds-japan-to-begin-training-employees-with-ninten-5498542>

la consola Nintendo DS. El objetivo del juego es hacer formación a los empleos para saber cómo se trabaja en McDonald's. En cuando la jugabilidad, el juego te proporciona unos manuales de instrucciones para leer, y posteriormente hacer entrenamientos o bien haciéndote preguntas, o bien simular una escena de un restaurante McDonald con unos problemas que hay que resolver siguiendo con las instrucciones del manual.



Figura 8. Saturo Itawa, el cuarto presidente de Nintendo (Izquierda) y Eikō Harada el director de McDonald's Japón (derecha).

- **Underground**⁵ es un videojuego para entrenar las habilidades de los médicos cirujanos. El juego hace una metáfora que relaciona los retos y los problemas de la cirugía con un mundo subterráneo llenos de monstruo y obstáculos. El jugador tiene que controlar un brazo robótico a destrozar los obstáculos a salvar a los peones, como un médico profesional operando al paciente usando su cuchillo de cirujano.

⁵ <https://grendelgames.com/spotlight/underground/>



Figura 9. Underground.

- **Ring Fit Adventure**⁶ es un videojuego que intenta motivar a los jugadores a entrenar y hacer ejercicio dentro de la casa. El juego viene con varios componentes físicos con sensores y hay que atarse en los manos y las piernas. El juego consiste en disfrutar una aventura en el mundo de juego controlando los movimientos (como correr o atacar) del personaje con tus propios movimientos gracias a los sensores que te ofrecen.

Un factor interesante es que durante la pandemia Covid19, cuando todo el mundo quedaba confinado en casa por la situación de emergencia sanitaria, Ring Fit Adventure fue el videojuego que ha resuelto el problema de falta de entrenamiento de mucha gente.

⁶ <https://www.nintendo.com/store/products/ring-fit-adventure-switch/>



Figura 10. Ring Fit Adventure.

“El método de enseñanza expositiva es mucho más eficiente, pero la conversación interactiva es más efectiva. Durante centenarios, los humanos hemos aprendido que la ganancia de eficiencia supera la pérdida de efectividad, por eso decidimos basarnos siempre en método de exposición (...) pero ahora tenemos la tecnología que cambia todo, con el computador podemos expresarnos en forma de algoritmo y podemos grabarlo en disco y difundirlo por todo el mundo.” (Crawford, Chris Crawford Dragon Speech "I Had a Dream", 1992). Chris Crawford, uno de los primeros diseñadores del juego, expuso estas ideas en el 1992, que coincide con las ideas de gamificación antes del nacimiento del dicho concepto.

Efectivamente, una charla de prevención de riesgo es eficiente y puede propagar a mayoría de la gente, pero, la información se convierte más fácil en conocimiento cuando hay una interacción, por lo tanto, Danger es teóricamente una herramienta muy potente para la enseñanza, además, gracias a Internet, hace que Danger realmente posible propagar por todo el mundo.

De esta forma, la pregunta de por qué elegir el videojuego como el soporte ya se queda completamente contestada.

2.3 Análisis del mercado

Actualmente, existe diversa modos para hacer formación de prevención de riesgo contra los incendios, por ejemplo, la enseñanza mediante: el simulacro, el juego, el videojuego o el simulacro mediante realidad virtual.

A continuación, desde la perspectiva de Danger, se procede a comparar con otras formaciones de prevención de riesgos, y sacar el análisis de debilidades, amenazas, fortalezas y oportunidades (también conocido como análisis DAFO), que es una herramienta potente para conocer la situación de un producto analizando tanto sus características internas (amenazas y oportunidades) como sus características externas (debilidades y fortalezas).

El **simulacro de incendio** consiste en hacer una simulación para practicar cómo se evacuaría en el caso de incendio, asimismo ayuda a una organización a determina si existen insuficiencias en los equipos de protección contra incendios y analizar las posibles mejoras.

	De origen interno	De origen externo
N e g a t i v o	<p>Debilidades:</p> <ul style="list-style-type: none"> - Requiere que el usuario dispone interés con el videojuego. - El usuario puede dejar de jugar. 	<p>Amenazas:</p> <ul style="list-style-type: none"> - El usuario no puede revisar su equipamiento real contra incendio. - A veces es menos eficaz que un simulacro.
P o s i t i v o	<p>Fortalezas:</p> <ul style="list-style-type: none"> - Ofrece una experiencia realista. - Fácil de adquirir. - Fácil de manejar. - Dispone recursos complementarios. <ul style="list-style-type: none"> - Promueve competición. 	<p>Oportunidades:</p> <ul style="list-style-type: none"> - Puede jugar en cualquier momento. - Puede entrenar en diferentes escenas. - Puede enfrentar casos imprevistos.

Tabla 3. Análisis DAFO de Danger con simulacro (Desde la perspectiva de Danger).

The Great Fire of London 1666⁷ (El Gran Incendio de Londres 1666) es un ejemplo de formación mediante juego de mesa. El juego está basado en el evento histórico real que ocurrió en el año 1666, cuando quemó gran parte del Londres y casi el noventa por ciento de la población quedaba sin hogar. En este juego el jugador no es un simple espectador de esta tragedia, sino que lidera un grupo de persona para combatir con el incendio.

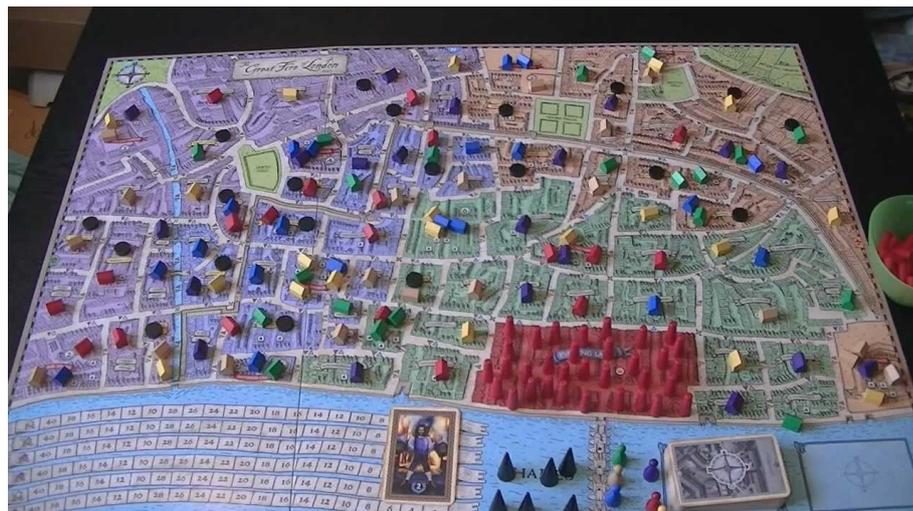


Figura 11. The Great Fire of London 1666.

⁷ <https://boardgamegeek.com/boardgame/41569/great-fire-london-1666>

	De origen interno	De origen externo
N e g a t i v o	<p>Debilidades:</p> <ul style="list-style-type: none"> - Requiere que el usuario dispone interés con el videojuego. - El usuario puede dejar de jugar. 	<p>Amenazas:</p> <ul style="list-style-type: none"> - Es menos divertido, ya que normalmente se juega solo.
P o s i t i v o	<p>Fortalezas:</p> <ul style="list-style-type: none"> - Ofrece una experiencia realista. - Fácil de adquirir. - Fácil de manejar. - Dispone recursos complementarios. - Incide competición. 	<p>Oportunidades:</p> <ul style="list-style-type: none"> - Danger está basado en sociedad actual. - Normalmente enfrentas con problemas reales, mientras que enfrentar con la situación de <i>The Great Fire of London 1666</i> es muy poco probable.

Tabla 4. Análisis DAFO de Danger con The Great Fire of London 1666 (Desde la perspectiva de Danger).

Flame Over⁸ es un videojuego que controla un bombero para rescatar a las personas y los animales atrapados en un edificio devorado por fuego. Durante el juego, el jugador tomará la decisión de elegir equipamiento oportuno para extinción de incendios, y aprovechar los recursos dentro del edificio como extintores o tanque de agua.

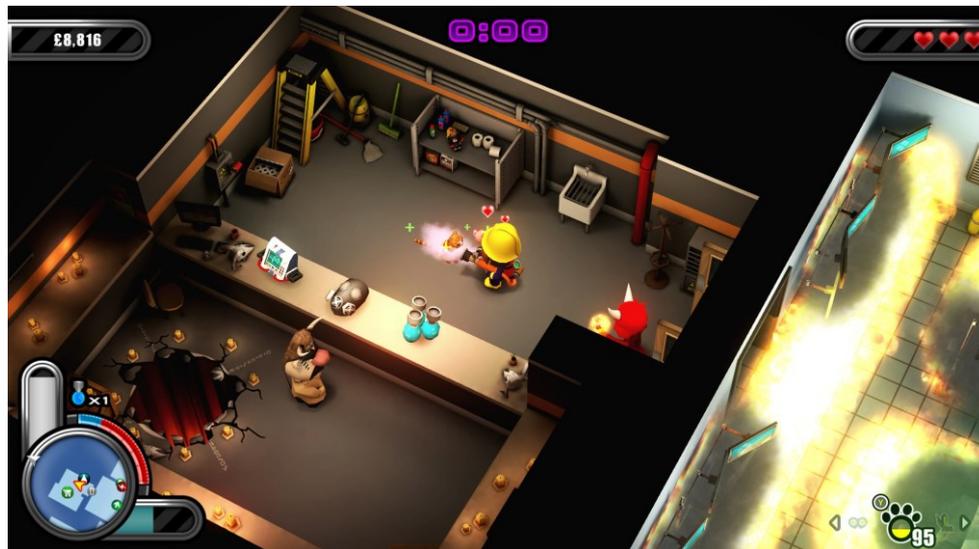


Figura 12. Flame Over.

	De origen interno	De origen externo
--	-------------------	-------------------

⁸ <http://www.flameovergame.com/>

N e g a t i v o	<p>Debilidades:</p> <ul style="list-style-type: none"> - Requiere que el usuario dispone interés con el videojuego. - El usuario puede dejar de jugar. 	<p>Amenazas:</p> <ul style="list-style-type: none"> - En Danger no puedes actuar como un bombero. - La gráfica es menos llamativa.
	<p>Fortalezas:</p> <ul style="list-style-type: none"> - Ofrece una experiencia realista. - Fácil de adquirir. - Fácil de manejar. - Dispone recursos complementarios. - Incide competición. 	<p>Oportunidades:</p> <ul style="list-style-type: none"> - Dispone gráfica más realista. - Dispone un sistema de preguntas y respuestas para consolidar el conocimiento.

Tabla 5. análisis DAFO de Danger con Flame Over (Desde la perspectiva de Danger).

Una simulación mediante realidad virtual consiste en crear un mundo virtual donde todos los objetos tienen apariencia real, y los usuarios pueden someterse como si fuera un mundo real. Para utilizar este sistema es imprescindible disponer un periférico como gafas o cascos de realidad virtual.

Ludus⁹ es una plataforma para hacer formaciones empresariales mediante accesorios de la realidad virtual. En dicha plataforma ofrece formaciones como prevención de riesgos planta, trabajos en altura, ergonomía postural...etcétera. Entre ellos se encuentra la extinción de incendios, que está orientado al entrenamiento del uso de extintores para apagar pequeños incendios.



Figura 13. Ludus.

	De origen interno	De origen externo
N e g a t i v o	<p>Debilidades:</p> <ul style="list-style-type: none"> - Requiere que el usuario dispone interés con el videojuego. - El usuario puede dejar de jugar. 	<p>Amenazas:</p> <ul style="list-style-type: none"> - Danger es menos inmersivo comparando un producto de realidad virtual. - Es menos interactivo comparando un producto de realidad virtual.

⁹ <https://ludusglobal.com/>

v o		
P o s i t i v o	<p>Fortalezas:</p> <ul style="list-style-type: none"> - Ofrece una experiencia realista. - Fácil de adquirir. - Fácil de manejar. - Dispone recursos complementarios. - Incide competición. 	<p>Oportunidades:</p> <ul style="list-style-type: none"> - Dispone niveles que se genera procedualmente, mientras que Ludus solamente dispone un nivel. - Es más barato, no requiere cascos de realidad virtual.

Tabla 6. Analisis DAFO de Danger con Ludus (Desde la perspectiva de Danger).

Una vez hecho todos los análisis, es mejor parar para hacer unas reflexiones:

- ¿Cómo puede hacer desaparecer las debilidades?

El videojuego por sí mismo requiere que la gente disponga interés para poder empezar a jugar, por lo tanto, captar interés de nuevos jugadores es un trabajo imprescindible, pero también es difícil de hacer. La única solución que propongo es hacer que el juego sea más amigable para que cualquier público pueda aceptar. Además, es posible que el jugador queda atascado y no sabe por dónde continuar y por el consecuente, deja de jugar, para ello, se debe ofrecer ayudas de alguna forma.

- ¿Cómo defenderse de cada amenazas?

Unos competidores tienen barreras insuperables por su naturaleza (como el simulacro o realidad virtual), por lo tanto, es mejor no pretender sustituir a ellos. Pero sí que existen posibles mejoras como la estética, añadir nuevos modos o añadir más funcionalidades online para que los jugadores puedan jugar con los demás.

En la sección 4 presentaré detalladamente qué cambios considero que son necesarios para superar todas las desventajas.

3. Herramientas

3.1 Introducción

Para poder llevar a cabo un proyecto grande es necesario utilizar muchas herramientas, tanto software como hardware: un motor de videojuego, un entorno de desarrollo integrado, un controlador de versión, un framework para el desarrollo de la página web.

A continuación, se presenta qué herramientas existe en el mercado, porqué se decide utilizar esta herramienta y cuáles son sus características.

3.2 Unity¹⁰ (Motor de videojuego)

Un motor de videojuego es un framework que ofrece herramientas útiles y potentes que facilita el desarrollo de un proyecto. Un motor de videojuego suele contener herramientas como calculadora física, gestor de sonidos, manejador de recurso de internet, manejador de hardware...etcétera. Gracias a su aparición, ahora a la hora de crear un videojuego es menos costoso y más rápido.

De entre todos los motores que existen actualmente en el mercado, para poder desarrollar un videojuego, se han seleccionado cuatro que, por las prestaciones que ofrecen, su implantación y calidad de entorno de desarrollo, se han considerado para desarrollo de este TFG. En concreto, se ha elegido para comparar entre ellos, los motores de videojuego como Unreal¹¹, CryEngine¹², Godot¹³ y Unity, unas de las características importes son:

	Unreal	CryEngine	Godot	Unity
Plataforma que soporta	PC + Consola+ teléfono móvil	PC + Consola	PC + Consola+ teléfono móvil	PC + Consola+ teléfono móvil
Lenguaje de programación	C++	C# y C++	GScript/VisualScript/C#...	C#
Open Source	Sí	Sí	Sí	No
Precio	Gratuito para proyectos pequeños	Gratuito para proyectos pequeños	Totalmente gratuitos	Gratuito para proyectos pequeños

Tabla 7. Tabla comparative entre Unreal, CryEngine, Godot y Unity.

Empezando comparar por el precio, son todos gratuitos para proyectos pequeños o incluso gratuito para todos (Godot) por lo tanto, son válidos para este proyecto.

¹⁰ <https://unity.com/>

¹¹ <https://www.unrealengine.com/en-US/>

¹² <https://www.cryengine.com/>

¹³ <https://godotengine.org/>

Pero CryEngine tiene una gran desventaja es que no soporta el desarrollo para videojuegos en teléfono móvil, que es una de las metas de este proyecto, entonces se queda descartada.

Unreal es open source, eso significa que es posible modificar el propio motor y dar una mayor flexibilidad a la hora de crear el videojuego. Pero no debe olvidar que el juego Danger ya tenía construido una base con la herramienta Unity y el lenguaje de programación C# mientras que Unreal se utiliza C++, el trabajo de la misma migración puede ser costoso, además hasta ahora no se ha presentado ninguna deficiencia con Unity y C#, por lo tanto, no es necesario cambiarse a Unreal.

Godot es la herramienta más ligera comparando con los otros, pero tiene un comportamiento inestable a la hora de crear videojuegos de 3D¹⁴, uno de los mayores problemas es que actualmente falta rendimiento 3D en Godot y su motor físico también es deficiente. Como crear un juego inmersivo también es uno de nuestro objetivo y la representación en 3D es esencial. Por lo tanto, se decide continuar utilizando Unity 3D.

Unity 3D es el un motor de videojuego multiplataforma desarrollado por Unity Technologies, fue creado el objetivo de “democratizar” el proceso de la creación del videojuego. Ahora Unity ya soporta crear el juego en más de 20 plataformas distintas. Entre los juegos creado con Unity destaca: *Pokémon Diamante Brillante* y *Pokémon Perla Reluciente*¹⁵, *Fall Guys*¹⁶.



Figura 14. Pokémon Diamante Brillante y Pokémon Perla Reluciente.

Trabajar con Unity conlleva muchas ventajas, estos son las 5 razones fundamentales para que yo elija esta plataforma:

- Interfaz usuario amigable: el trabajo de la programación es más visual, por lo tanto, agiliza la edición de script. También permite editar fácilmente las escenas y las animaciones de los personajes.

¹⁴ https://www.reddit.com/r/godot/comments/oq8eya/why_cant_you_make_3d_games_in_godot/

¹⁵ <https://diamondpearl.pokemon.com/es-es/>

¹⁶ <https://www.fallguys.com/en-US>

- Programación orientada a componentes: hace que el proyecto sea más modular y los recursos son fácilmente reutilizable para el mismo proyecto o incluso para el otro proyecto.
- Multiplataforma: permite exportar fácilmente tu juego a diferentes plataformas de computador, teléfono móvil y video consola.
- Lenguaje de programación C#: Unity te permite programa en C#, eso implica que es más fácil de programar, más rápido a la hora de compilar, más fácil de depurar, y tiene mejor integración con Visual Studio.
- Tienda virtual: en la tienda virtual oficial de Unity (Unity Asset Store) se puede encontrar muchos modelos 2D/3D, scripts, plugin...etcétera. Eso ayuda a el usuario a cubrir su punto débil, por ejemplo, si un artista no sabe programar, pero quiere desarrollar un videojuego, encontrará muchas herramientas en la tienda que hace que la programación sea más visual y evita escribir códigos.
- Herramienta gratuita para desarrolladores pequeños, incipientes o estudiantes.

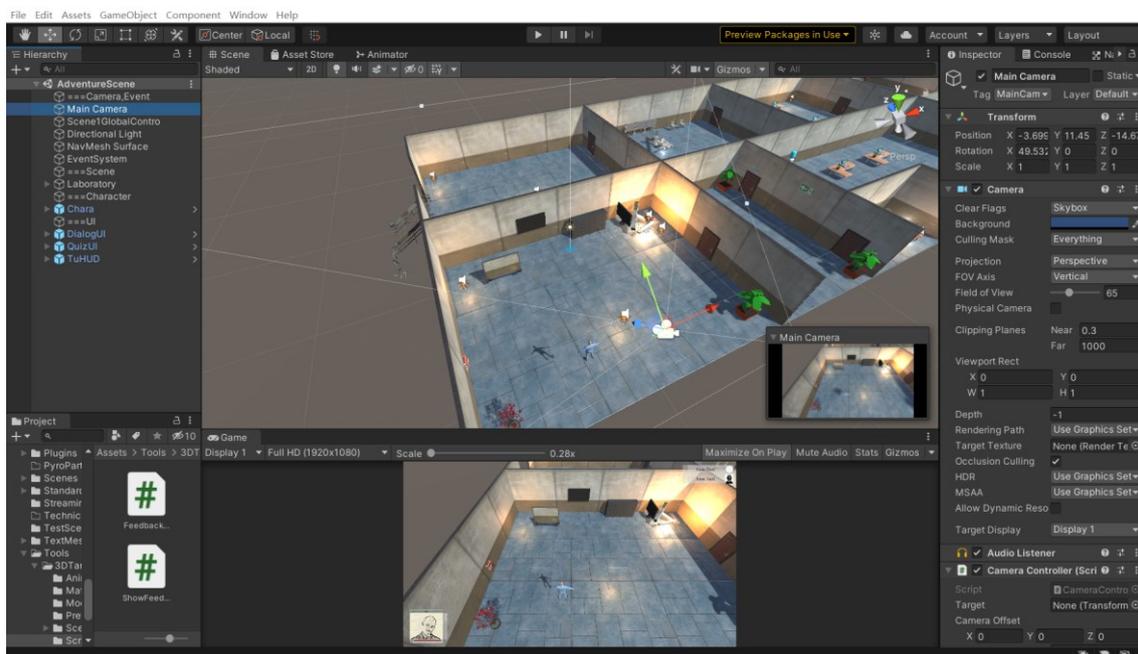


Figura 15. Interfaz de Unity.

3.3 Visual Studio¹⁷ (Entorno de desarrollo integrado)

Un entorno de desarrollo integrado es un software que proporciona un entorno para facilitar la programación. Generalmente se incluye herramientas como un editor de código, un compilador, un depurador y una interfaz gráfica de usuario. El propósito de un entorno de desarrollo integrado es ahorrar tiempo que se pierde navegando entre diferentes programas y hacer que el desarrollo sea más rápido y conveniente gracias a su organizador los recursos, notificador los errores y los atajos.

¹⁷ <https://visualstudio.microsoft.com/>

Trabajar con Unity implica el uso del lenguaje de programación C#, y existe herramientas como JetBrains Rider¹⁸, MonoDevelop¹⁹ o Visual Studio.

Pero en este caso no hace falta comparar, es mejor utilizar Visual Studio, ya que es la herramienta recomendada oficialmente por Unity²⁰.

Visual Studio es un entorno de desarrollo integrado desarrollado por Microsoft, ofrece la mayoría de las herramientas necesarios para todo el ciclo de vida de un software, aparte de herramientas esenciales como compilador de código, depurador de código, compilador, la interfaz usuario, también proporciona herramientas para diseñar los diagramas de clase, herramientas de control de código...etcétera.

Utilizando Visual Studio con Unity hay muchos beneficios:

- Un ambiente de desarrollo con C# sofisticado.
- Autocompletado inteligente.
- Resaltado de sintaxis más inteligente.
- Herramienta gratuita para desarrolladores pequeños, incipientes o estudiantes.

```

1  using UnityEngine;
2
3  namespace Graphics.Feedback.Scripts
4  {
5      /// <summary>
6      /// Handles the feedback pointer position.
7      /// </summary>
8
9      public class FeedbackPointer
10     {
11         /// <summary>
12         /// Instance of the move indicator created from _moveToIndicator.
13         /// </summary>
14         private static GameObject _moveToIndicatorInstance;
15
16         /// <summary>
17         /// Creates an instance of given prefab and scales it.
18         /// </summary>
19         /// <param name="prefabInstance">Prefab to instantiate.</param>
20         /// <param name="scale">Scale of the pointer.</param>
21
22         public void PreparePointer(GameObject prefabInstance, float scale)
23         {
24             if (prefabInstance == null) return;
25
26             if (_moveToIndicatorInstance == null)
27                 _moveToIndicatorInstance = Object.Instantiate(prefabInstance);
28
29             _moveToIndicatorInstance.SetActive(false);
30             _moveToIndicatorInstance.transform.localScale = Vector3.one * scale;
31         }
32     }
33 }

```

Figura 16. Interfaz de Visual Studio

3.4 Github (Controlador de versión)

El sistema controlador de versión sirve para gestionar cambios de un programa. El sistema tiene tres funcionalidades esenciales para la administración: el almacenamiento de archivos que deseas controlar, la posibilidad modificación de código, la visualización de cambio y registro histórico de las acciones.

Los controladores de versiones más comunes que se encuentra en el mercado son: Apache Subversion (SVN)²¹ y Git²².

¹⁸ <https://www.jetbrains.com/>

¹⁹ <https://www.monodevelop.com/>

²⁰ <https://docs.unity3d.com/es/2018.4/Manual/VisualStudioIntegration.html>

²¹ <https://subversion.apache.org/>

²² <https://git-scm.com/>

Se puede hacer una tabla comparativa para visualizar sus características:

	SVN	Git
Características	Sistema de control centralizado	Sistema de control distribuido
Flexibilidad	Si falla el servidor central, pierde el control	Se puede trabajar en local
Permiso de acceso a los archivos	Muy restricto, puede definir roles	Poca restricción
Gestión	Existen programas de complemento para gestionar.	Existen programas de complemento y grandes plataformas como Github

Tabla 8. Tabla comparativa entre SVN y Git

Subversion está diseñado más orientado hacia el desarrollo paralelo y centralizado de la empresa, mientras Git se usa más para proyectos corporativos personales. Pero en caso práctico, ambas herramientas presentan un comportamiento muy similar puesto que no todos los proyectos necesitan aprovechar todas las funcionalidades, sino que simplemente necesita una plataforma para controlar los cambios, como este proyecto.

Por lo tanto, es más importante elegir programa de complemento que facilita la gestión, y Git lleva ventaja de que existen mucha plataforma online que ofrece servicios de alojamiento de código fuente que utiliza Git como base. Para este proyecto se utiliza la plataforma que tiene una comunidad más grande según la estadística de StackOverflow en el año 2021²³, Github²⁴.

Github es una plataforma de alojamiento para proyectos. Es una herramienta potente, ya que además de alojamiento del repositorio de código y la interfaz de administración a través de la página web, también ofrece funciones como suscripción, grupo de discusión, informe...etcétera, gracias a su comunidad tecnológica. Actualmente tiene muchos proyectos muy grandes alojada en dicha plataforma.

El uso de controlador de versión es esencial para un proyecto grande, entre las ventajas hay que destacar:

- Almacenamiento de las versiones: es muy importante guardar los cambios del proyecto con frecuencia, con el sistema de control de versiones, te permite a ti y tus compañeros de equipo visualizar fácilmente los cambios en cada versión.
- Fácil de recuperación: si te has dado cuenta de que has cometido un error en tu cambio y quieres quitarlo, simplemente poder deshacerlo con unos pocos clics.
- Copia de seguridad: cada miembro del equipo tendrá una copia completa del proyecto en su entorno local, incluida la historia de todo el proyecto. Si el servidor que contiene tu proyecto sufre errores, puedes recuperar todos los archivos con un repositorio local.
- Trabajo en equipo: cada miembro del equipo puede modificar cualquier archivo en cualquier momento y luego el sistema puede fusionar todos los cambios en una versión común.

²³ <https://insights.stackoverflow.com/survey/2021#stack-overflow-community-now-new-other-comms-names-other>

²⁴ <https://github.com/>



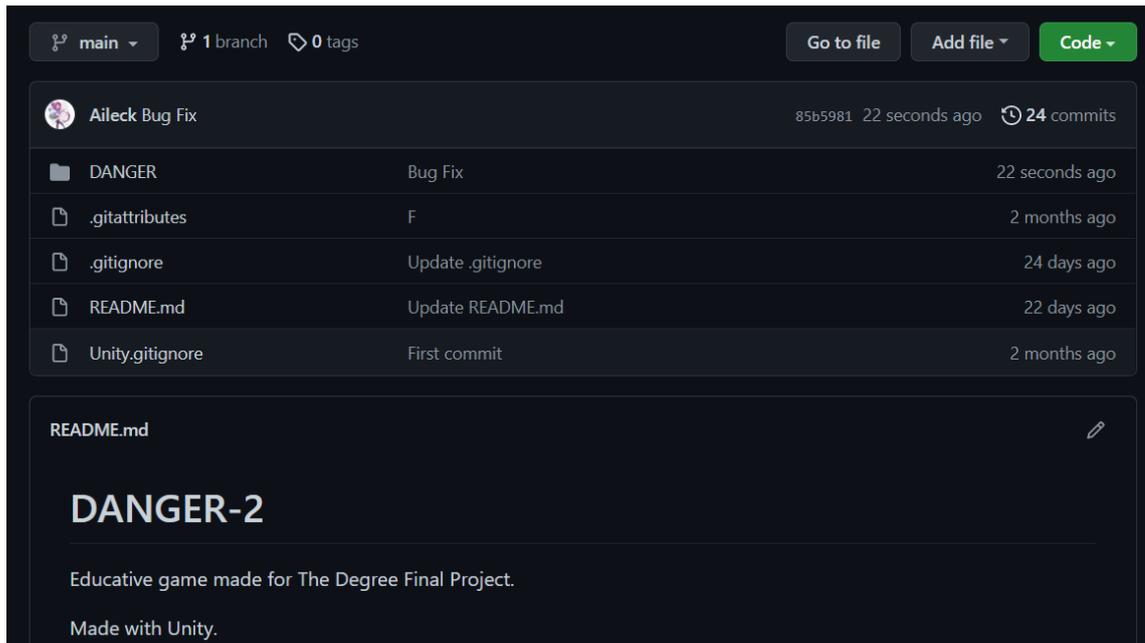


Figura 17. Interfaz de Github.

3.5 Angular (Framework para la página web)

Un framework para la página web facilita tanto el proceso de desarrollo de la página web como el tratamiento de las peticiones de la web, el buen uso de dicho componente puede mejorar en gran medida la eficiencia del desarrollo.

Como el caso de Unity, vamos a continuar usando la herramienta de eligieron: Angular.

Angular es uno de los framework con eficiente para diseñar software complejas y sofisticada.

Las ventajas de trabajar con Angular son:

- Se basa en el patrón de modelo-vista-controlador, eso hace que la aplicación sea más escalable y fácil de reutilizar códigos.
- Es relativamente completo, viene con muchas funcionalidades que facilidad el desarrollo como plantillas, enlace bidireccional de los datos y modularización.
- Es creado por Google, por lo tanto, tiene una comunidad sólida.

4. Diseño

4.1 Introducción

Tal como presentaba en la introducción, Adrián Sánchez Lavarias²⁵ y Pablo Querol Ballester²⁶ habían diseñado y creado una estructura básica del juego. En esta sección se da un repaso de sus trabajos.

El videojuego empieza con una cinemática introductoria y luego nos lleva al menú principal, en este menú encontrarás cinco botones que corresponde a las funcionalidades de: créditos, configuraciones del juego, modo Build, modo Escape y salir del juego.

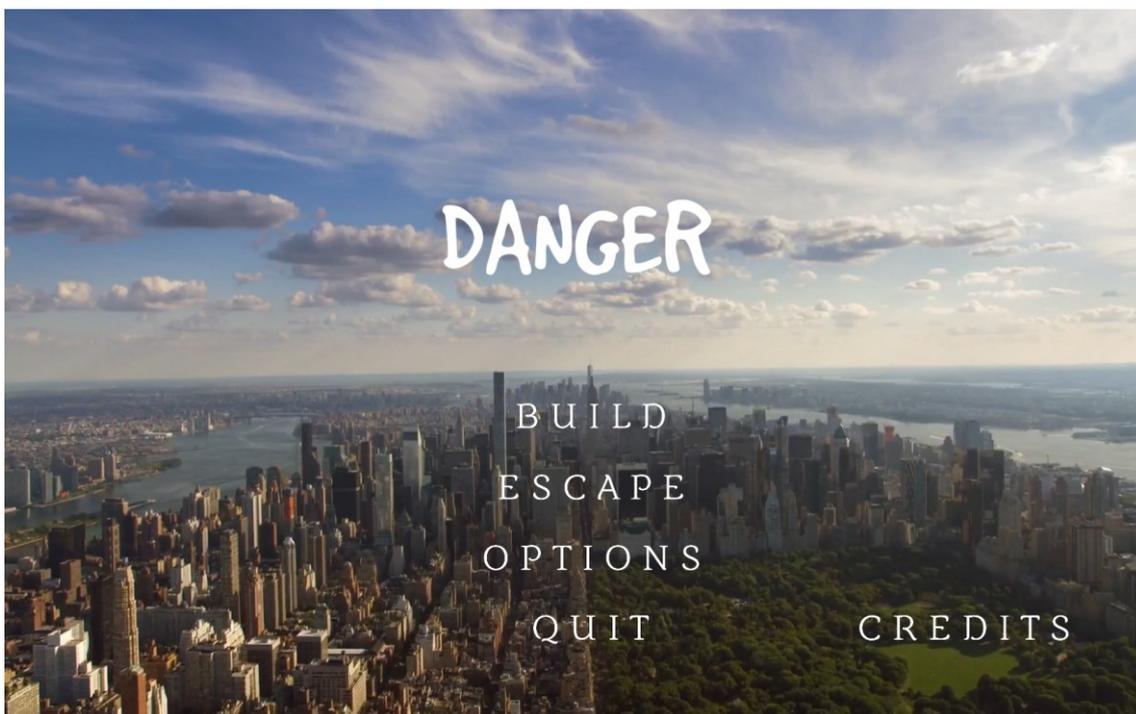


Figura 18. Menú de inicio de la versión 2020-2021.

Los dos modos de juegos mencionados anteriormente forman el núcleo didáctico del juego, ya que el objetivo es que todos los jugadores puedan aprender después de haberlo jugado. Estos dos modos del juego son:

- Modo Build: El modo de construcción te da un espacio abierto y te permite colocar bloques como suelo, paredes, mesas...etcétera. El objetivo del jugador es crear una habitación o una sala de acuerdo con las limitaciones establecidas. El jugador dispone un panel que muestra estadísticas en tiempo real y un botón *Scan* para saber si la habitación construida es correcta o no.

²⁵ <https://riunet.upv.es/handle/10251/174398>

²⁶ <https://riunet.upv.es/handle/10251/174568>

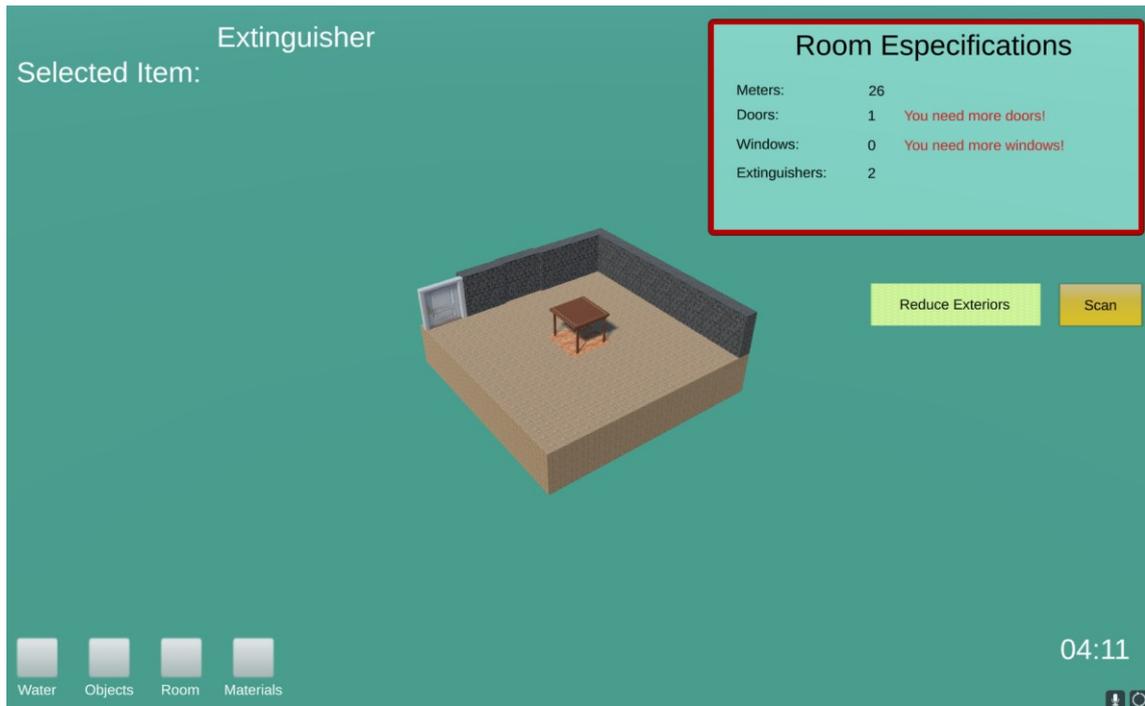


Figura 19. Modo construcción.

- Modo Escape: El modo de estrategia a tiempo real, el jugador debe escapar de una escena generada de forma procedural, es decir, cada vez que se reinicia el juego la escena es diferente. El juego consiste en la simulación de incendios en una oficina y el jugador debe guiar a todas las oficinistas a la salida evitando contactos con el fuego. En este modo, todos los personajes controlables tienen dos atributos muy importantes: puntos de sangre y estado de ánimo, si cualquier de los dos valores caen a cero, el personaje muere.

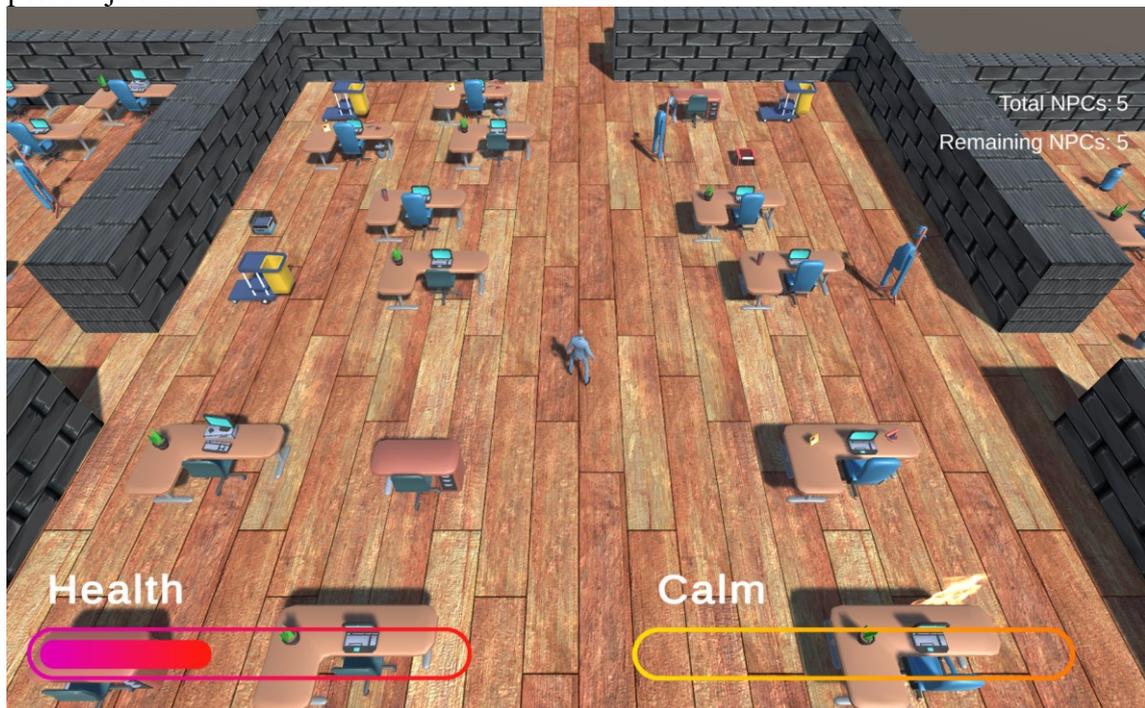


Figura 20. Modo estrategia.

Por lo tanto, el juego puede reducirse en un diagrama de flujo de la siguiente forma:

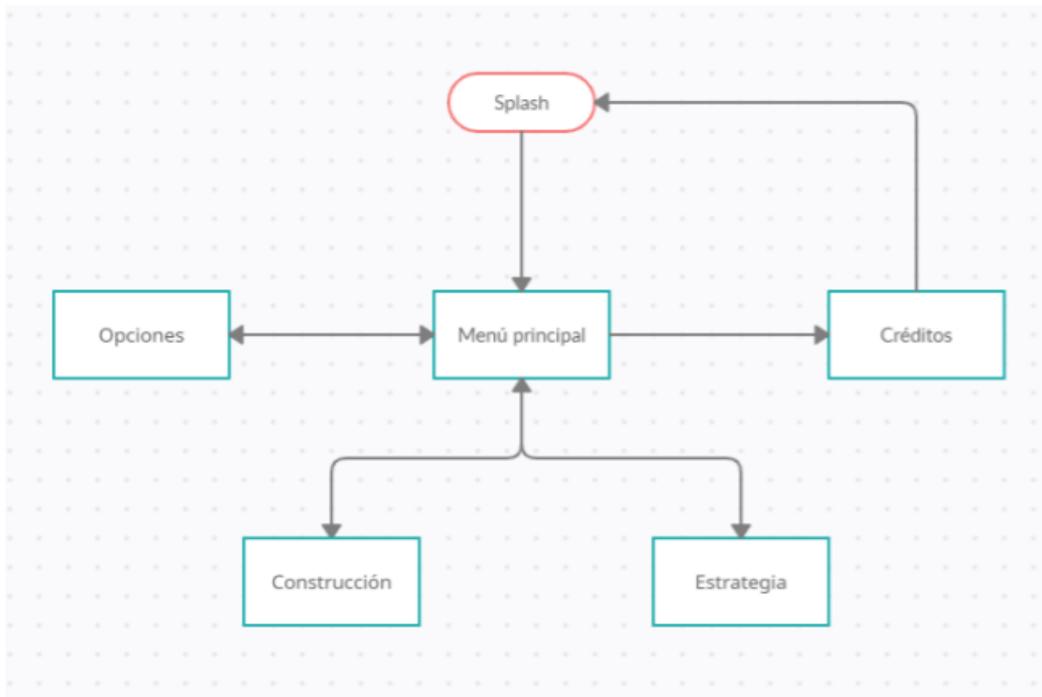


Figura 21. Diagrama del flujo del juego versión 2020-2021

También diseñaron una página web usando el framework Angular, para ofrecer informaciones extras sobre cómo actuar en una situación de emergencia.

Además, conectaron el juego a la herramienta Firebase, que es una plataforma para conectar a Unity con la página web para poder gestionar datos e intercambiar información entre ellos.



Figura 22. Página web y completa el juego.

4.2 Análisis de posibles mejoras

En este apartado se pasa a analizar los aspectos que se puede mejorar.

- El juego en sí **no da pista de cómo jugar**, ni ofrece un tutorial de cómo controlar. Un jugador que no lee el manual (que tampoco lo tenemos) no sabrá cómo jugar ni sabe cuál es su objetivo.
- El juego hay muy **pocas retroalimentaciones** y hay poca comunicación con los jugadores. Por ejemplo, en modo Escape cuando seleccionas un personaje, el juego no resalta el personaje que está seleccionando, en modo Build, no puedes previsualizar el objeto que quieres colocar. Además, no marcan el final del juego ni evalúa el comportamiento del jugador, el jugador quedará encerrado en este espacio virtual y no percibe ninguna retroalimentación del juego, en consecuencia, el jugador aprenderá muy poco.
- El juego **es poco interactivo**, por lo tanto, no es divertido. En el modo Escape, el jugador solamente puede mover el personaje, y no puede interactuar con ningún elemento del entorno.
- La **página web es difícil de lanzar**, hay que leer una instrucción muy extensa para poder visualizar el contenido de la página web.
- La **página web no complementa al juego**. La existencia de la página web es prácticamente redundante ya que solamente ofrece informaciones extras y no hay contenido acerca del juego.
- **Completar el modo build**, la versión actual hay muy pocos componentes para construir una habitación. Además, solamente de ha definido una regla para limitar el comportamiento de los jugadore, que en mundo real, existe otras más restricciones a la hora de construir²⁷.
- **Mejorar el algoritmo de generación de escena procedural**, ya que presenta muy poca variedad y a veces produce errores.
- La **estética**, se debe mejorar para que el juego sea más llavativo y cualquier persona tenga interes para seguir jugando.

Fuera del contexto del videojuego, el documento de diseño de videojuego que crearon también fue impreciso y falta muchas informaciones. Los puntos mejorables son:

- **Falta de información:** Algunos puntos se describía con palabras muy simples y no entran al detalle.
- **Falta de apartados:** Falta de algunos apartados muy importantes como concepto abstracto, comparación con competencias, historia y personajes...etcétera.

4.3 Diseño de las mejoras

Para este proyecto se intentará mejorar siguientes puntos:

- El juego en sí no da pista de cómo jugar.
- El juego hay muy pocas retroalimentaciones.
- El juego es poco interactivo.
- La página web es difícil de lanzar.
- La página web no complementa al juego.
- Rediseño del documento de diseño de videojuego.

²⁷ <https://www.codigotecnico.org/pdf/Documentos/SI/DBSI.pdf>

4.3.1 Problema 1: No da pista de cómo jugar.

Existen tres posibles soluciones:

En primer lugar, se puede diseñar una página web que contiene todas instrucciones de cómo jugar. La ventaja es que los jugadores pueden consultar al manual del juego en cualquier momento sin tener que abrir el videojuego. Sin embargo, es posible que los jugadores no vean este tutorial debido por varias razones como: no saber la existencia de la página web, jugar en un entorno sin internet o el jugador no quiere cerrar el juego para leer un tutorial fuera del juego.

En segundo lugar, se puede hacer que el juego muestra todas las posibles acciones a tiempo real. Como muestra en la Figura 23:



Figura 23. Captura de pantalla de Persona 5.

De esta forma, el jugador siempre dispone todas las informaciones de qué cosas puede hacer en cada instante. Aunque eso supone muchos tiempos de programación y tiempos de testeo, además como nuestro juego será multiplataforma, es difícil adaptar para cada sistema.

La última propuesta es integrar una tutorial dentro del juego, es sencillo y llega a todos los usuarios con facilidad, el inconveniente es que un jugador que ya conocía los comandos también tiene que volver a leer, pero eso se resuelve fácilmente con un botón para saltar el tutorial.

En resumen, la mejor solución para este problema es añadir un **tutorial integrado en el videojuego**.

4.3.2 Problema 2: Pocas retroalimentaciones

Una buena interfaz gráfica proporciona retroalimentaciones clara y significativa para cada acción del usuario. La falta de retroalimentación obliga a los usuarios a verificar dos veces para ver si se han realizado las acciones previstas y eso ocurre muchas veces en el juego, por ejemplo, cuando pasa a controlar otro personaje, no da ninguna retroalimentación visual de que se ha seleccionado dicho personaje o cuando se mueve

un personaje, tampoco se sabe que, si está moviendo aleatoriamente o está dirigiendo el destino que le está indicando, hay muchos detalles de este tipo que se puede mejorar.

También se crear un sistema de colección de datos para que al final del juego pueda valorar todos los comportamientos de los jugadores indicándoles qué es lo que han hecho bien y qué es lo que han hecho mal, para que realmente puedan aprender del juego, para eso, se debe de diseñar un **sistema de dialogo** para poder conversar con los usuarios.

4.3.3 Problema 3: Poco interactivo

Para enriquecer el contenido del juego, mi compañera de máster de prevención ha propuesto una serie de ideas, y yo ordeno sus ideas y lo convierto en mecanismos del videojuego. A continuación, voy a exponer las ideas suyas.

Las escenas que se construye serán de forma manual (no como el modo Escape, que se genera de forma procedural), la ventaja de escena manual es que el comportamiento de los jugadores es más controlable, y un nivel bien diseñado es más fácil despertar intereses de los jugadores e incluso pueden estar completamente involucrados en el juego para aprender más.

Este es el mapa del primer nivel:

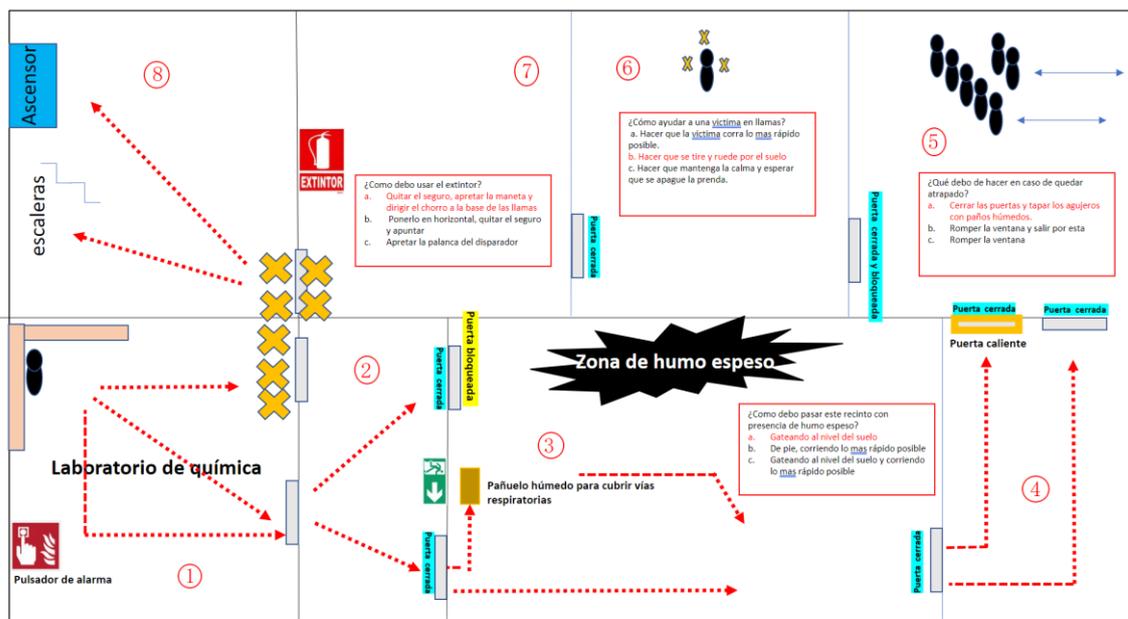


Figura 24. Plano de la planta del nivel uno.

En este mapa se ve una planta de un edificio, a su vez, también se observa que dicha planta se encuentra dividida en ocho habitaciones. En la Figura 24 se ve las habitaciones y su número de identificación asociado.

El jugador comienza desde la habitación número uno, en esta habitación se encuentra con dos puertas, una se encuentra bloqueada por el fuego y la otra no. Opcionalmente, el jugador puede pulsar la alarma para recibir una bonificación. Aquí aparece la primera funcionalidad que hay que implementar: **el sistema de bonificación**.

La habitación dos también se encuentra con dos puertas, y ambas se pueden abrir. En el plano se observa que hay una señal de salida con flecha, si el jugador no sigue las

indicaciones de la señal, recibirá una penalización. Como antes, hay implementar **el sistema de penalización**.

La tercera habitación es una zona con humo oscuro, si el jugador coje el pañuelo húmedo para cubrir vías respiratorias recibirá otra recompensa, y si quiere seguir avanzando, tiene que contestar una pregunta sobre cómo pasar el recinto con presencia de humo espeso. Aquí aparecen dos nuevas funcionalidades: **el sistema de inventario** y **el sistema de preguntas**.

En la cuarta habitación otra vez se encuentra con dos puertas, una de ella está caliente y no se debe abrir en un incendio, si el jugador intentar abrir, se penaliza. Para este mecanismo se diseñar **un sistema de visualización de temperatura**.

La habitación quinta y sexta se encuentra con puertas bloqueadas y personajes no controlables atrapadas en esta habitación, el jugador debe responder preguntas para poder salvar a todos ellos y abrir las puertas.

En la séptima habitación se encuentra con un extintor y una única puerta bloqueada por el fuego. Para poder continuar, el jugador debe contestar la pregunta de cómo usar el extintor para poder apagar el fuego con éxito.

En la última habitación hay que tomar decisiones para evacuar: escapar por el ascensor o escapar por la escalera, si decide escapar por el ascensor recibirá otra penalización.

Aparte del diseño del mapa, también se aumenta la dificultad del juego añadiendo **el sistema de tiempo límite**, ya que el fuego se propaga con el tiempo, de esta forma obliga a los jugadores a tomar decisiones rápidas para ofrecerles una experiencia inmersiva.

4.3.4 Problema 4: Página web difícil de lanzar

Existen dos posibles soluciones:

La primera opción es desplegar una aplicación web subiendo todos los archivos a un servidor ajeno. De esta forma, dicha página web pasará de estar en local a online, y cualquier usuario pueda visualizar esta página web conociendo el URL (Localizador de recurso uniforme).

La segunda opción es refactorizar el código con el fin de reducir pasos para lanzar la web en local.

La primera opción es la óptima y amigable para los usuarios, pero uno de los problemas es que para desplegar la aplicación web, hay que pagar por el servidor por el consumo de los recursos. Como dicho proyecto sigue en fase de prueba, no se precipita en subirlo, aunque una vez publicado el juego al público, publicar la página web en Internet es esencial.

Sin duda, la segunda opción es la más oportuna para ahora.

4.3.5 Problema 5: Página web redundante

Para aprovechar la página web, se puede añadir más funcionalidades relacionado con el juego.

Como se integran series de preguntas dentro del juego, entonces en la página web se puede dejar las respuestas para que los jugadores no quedan atascados cuando enfrentan



a una pregunta que no sabe contestar. Por su parte, estas respuestas también pueden servir como repastos de las preguntas del juego sin tener que abrir el juego.

Adicionalmente, es posible añadir más comentarios o enlaces de las documentaciones relacionadas para cada una de las preguntas, que es redundante si se integra dentro del juego, pero es útil si lo pone en la página web.

4.3.6 Problema 6: Documento de diseño impreciso

Un documento de diseño de videojuego es imprescindible, ya que su contenido es altamente descriptivo acerca del videojuego, además es sumamente importante para este proyecto porque cada año pasará por mano de un desarrollador diferente. Por lo tanto, se procede a realizar una reorganización de ideas y una ampliación a dicho documento, para evitar posibles confusiones futuras.

Las tareas se realizan en este trabajo son:

- Reorganizar las secciones.
- Añadir secciones:
 - Análisis de las competencias.
 - Estructura narrativa y los personajes.
 - Herramientas complementarias.
- Añadir apartados:
 - Temática del juego.
 - Concepto abstracto.
 - Finalidad del juego.
 - Elementos jugables.
 - Niveles.
 - Escenas.
- Ampliar contenidos:
 - Resumen.
 - Estética.
 - Gameplay.
 - Público dirigido.
 - Assets.

Además, es un documento vivo, es decir se edita durante el desarrollo del juego, por lo tanto, todas las ideas mencionadas anteriormente y otros pequeños cambios para mejorar la experiencia del juego también están anotado en el documento de diseño del videojuego.

La última versión del documento de diseño de videojuego se encuentra en el anexo II.

5. Implementación

5.1 Introducción

En este apartado presenta cómo se implementa las funcionalidades comentadas en el apartado anterior.

Como el nuevo nivel es parecido como el modo Escape, pero tiene dos diferencias fundamentales:

- La escena no se genera de forma procedural, sino que está planificado y diseñado manualmente.
- El jugador ya no puede controlar todos los personajes de la escena, ahora solamente puede controlar un solo personaje y su objetivo es ayudar y guiar a otros personajes no controlables a escapar del incendio.

Por lo tanto, será tratado como un nuevo modo, y pasará a llamar Adventure Mode (Modo aventura). Así que se procede a rediseñar el diagrama de flujo y el menú principal.

El nuevo diagrama tras la actualización se quedará en la siguiente forma:

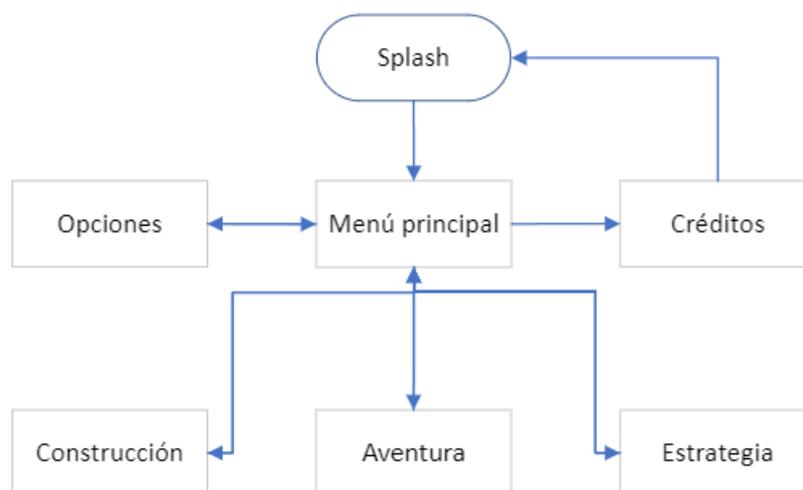


Figura 25. Diagrama de flujo de la versión 2021-2022.

Y el menú principal del juego se queda así:



Figura 26. Menú de inicio de la versión 2020-2021.

5.2 Implementación de las mejoras

En los apartados 4.2 y 4.3 ya se presentó cuáles son los puntos que se puede mejorar, cómo se puede mejorar y porqué se hace de esta forma, ahora pasa a explicar detalladamente cómo se implementa estos cambios.

5.2.1 Sistema de tutorial integrado

Este tutorial se visualizará una vez el jugador entra al modo aventura.



Figura 27. Ejemplo de una página de tutoría.

El panel informativo que de la Figura 28 está hecho en con la herramienta *Canvas* de Unity, que es un área donde se coloca los elementos de interfaz de usuario como cuadros de texto, imágenes y botones.

Como para este tutorial quiero hacer un diseño multipágina, es decir, parecido a un libro físico y pueda navegar entre las páginas con un botón, entonces se hace que el botón “Continuar” se atienda a los eventos de clics (o eventos de pulsar el ratón).



Figura 28. Hacer que un botón atienda a un evento en Unity.

Cuando el botón “Continuar” percibe *clics*, intenta cargar el siguiente Canvas hasta que no queda más Canvas en la lista y en este caso, entra a la siguiente escena del juego. Esto reducido en pseudo código es:

```
if (currentPage != lastPage)
    currentPage = nextPage
else
    load(nextScene)
```

Y el botón “Saltar” es para que los jugadores con experiencias puedan saltarse de este tutorial, por lo tanto, su pseudo código es tan simple como

```
load(nextScene)
```

5.2.2 Mejora de las retroalimentaciones

Se procede a diseñar tres cambios para mejorar la experiencia del juego.

En primer lugar, ahora cuando indica el destino de un personaje, se visualiza en la pantalla.

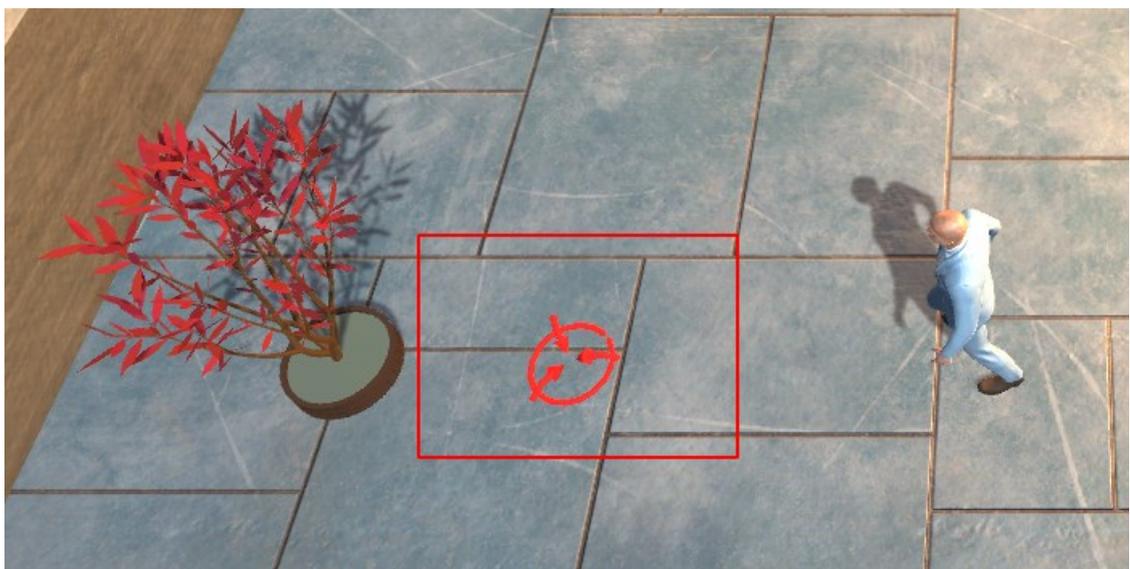


Figura 29. Destino del personaje dibujado en la pantalla (Remarcado con el cuadro rojo).

En la Figura 29 se observa que hay tres flechas apuntando hacia en suelo, estas flechas indican hacia qué sitio se desplazará el personaje.

Esta funcionalidad no lo hemos implementado nosotros, sino que es un script gratuito (*3D Target Cursor Marker*²⁸) de la tienda virtual de Unity. El uso de esta script es tan fácil como asignar a un GameObject de suelo y automáticamente te dibuja las flechas en la pantalla.

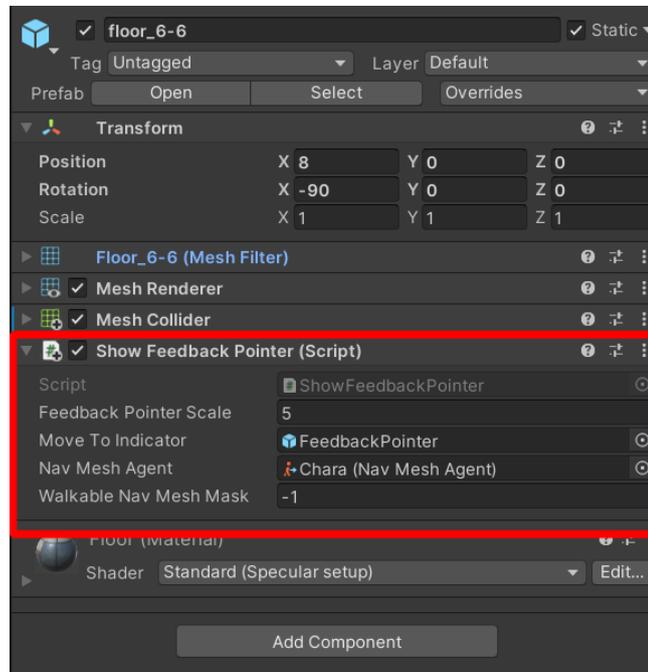


Figura 30. Detalle de script *3D Target Cursor Marker*.

La segunda mejora es el sistema de dialogo, gracias a él, ahora se notifica lo que está ocurriendo en la escena en cualquier instante. Es muy parecida como paneles de tutorial puesto que sus implementaciones son prácticamente iguales.

²⁸ <https://assetstore.unity.com/packages/3d/gui/3d-target-cursor-marker-155582>

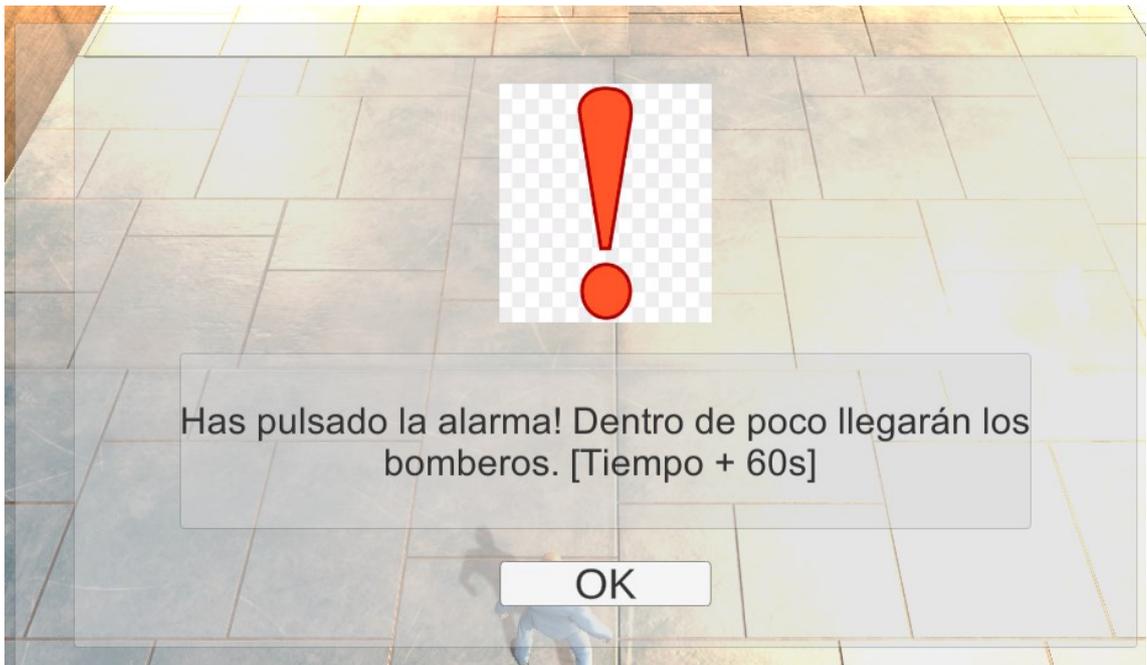


Figura 31. Notificación que aparece cuando pulsa el botón de alarma de incendio.

La última mejora para el sistema de retroalimentación es que se ha registrado todas las acciones del jugador, gracias a dicha información, hace posible decir al jugador qué cosas ha hecho bien y cuáles ha hecho mal.

Para lograr eso simplemente he creado un *GameObject* vacío, y le he asignado un script que colecciona unas series de informaciones, y en cada punto de programa comprueba si el jugador a respetado la normativa de prevención de riesgo.

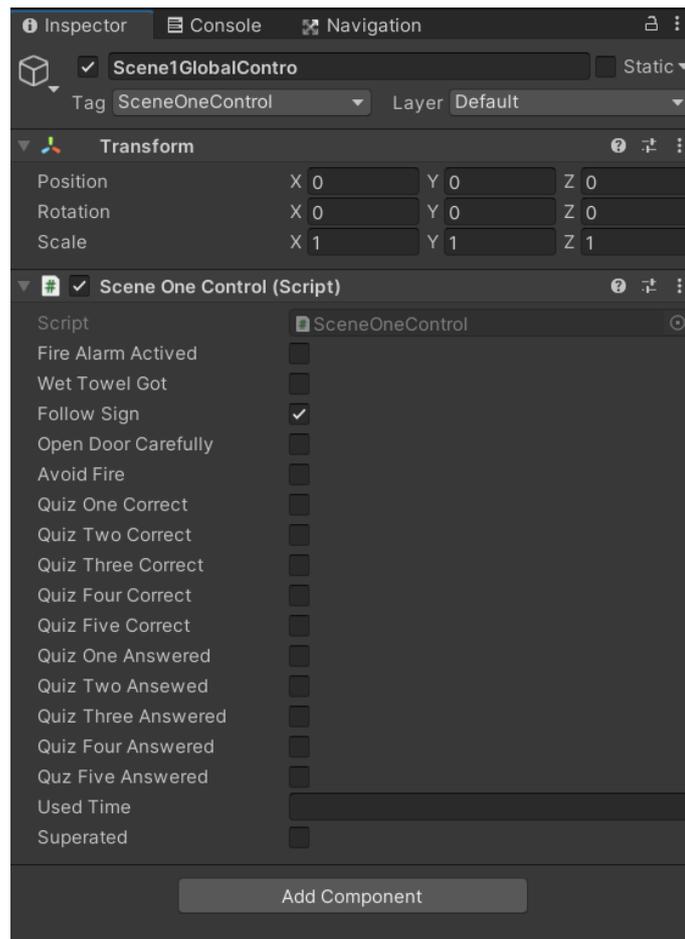


Figura 32. Detalle del script de colector de información.

Por ejemplo:

Fijar en la variable *Fire Alarm Activated* de la Figura 32, se observa que inicialmente está marcado como *false*.

Si durante el transcurso del juego, el jugador ha pulsado la alarma de incendio, aparte de recibir una bonificación, el programa también envía señal al script de la Figura 32 diciéndole que sí que ha activado la alarma (la variable *Fire Alarm Activated* pasa a ser *true*). Cuando el jugador supera el nivel, en la página de resultado marcará un tic en el campo de “Has activado la cámara”, en el caso contrario, marcaría con una cruz.

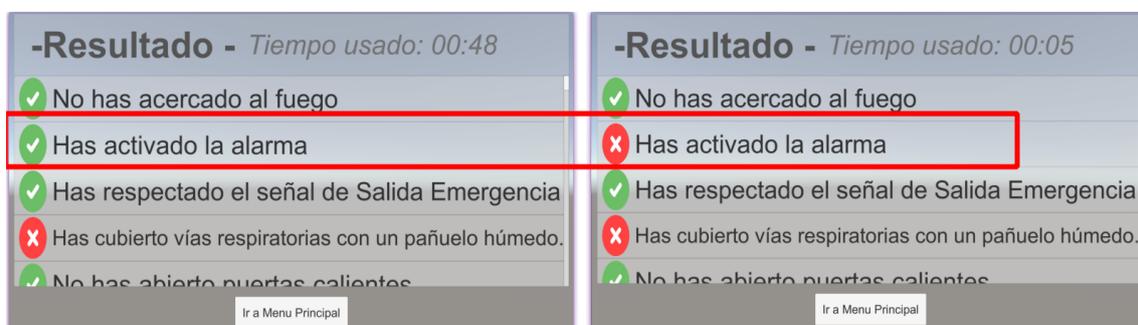


Figura 33. Página de resultado cuando el jugador activa la alarma de incendio (izquierda) y cuando no activa la alarma (derecha)

5.2.3 Nuevo nivel

En la siguiente ilustración podemos ver el modelo 3D de la escena basándose en el mapa de la figura.

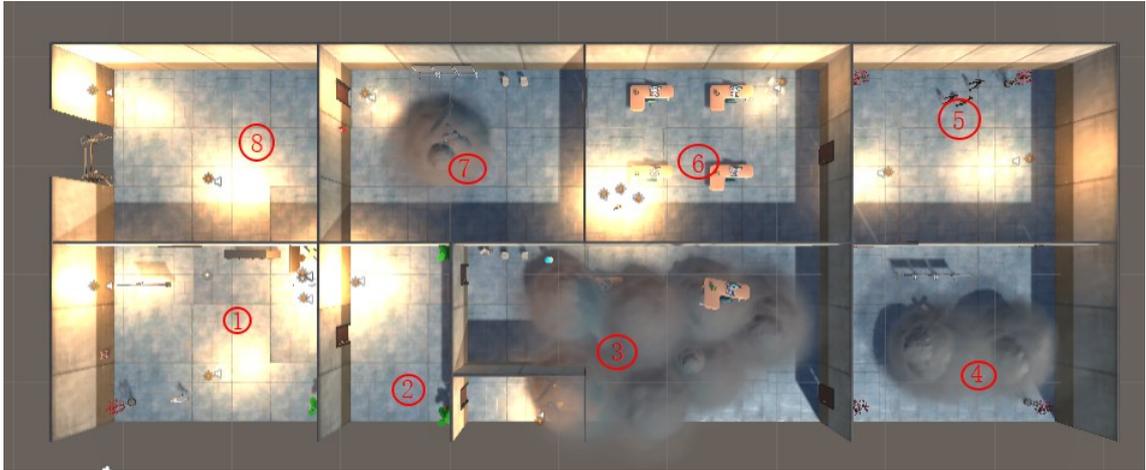


Figura 34. La Figura 24 en 3D.

En cada una de las salas añadiremos eventos que habíamos comentado en el punto 4.3.3, por lo tanto, aquí no entraremos en detalle, y las nuevas funcionalidades que hacen falta para completar el nivel lo explicaremos en los siguientes puntos.

Cabe destacar que, todos los componentes de la escena son proporcionados en la tienda de Unity, gracias a él, hace posible crear una escena sin tener un modelador de 3D en el equipo.

Para construir la escena de la Figura 34, se ha empleado siguientes recursos: el entorno de laboratorio²⁹, efectos de partículas de fuegos y humos³⁰.

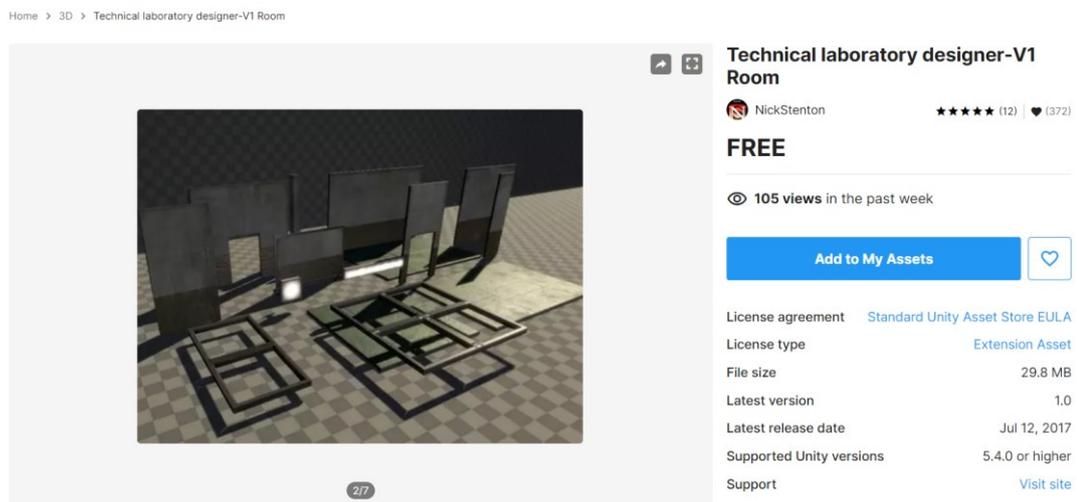


Figura 35. Ejemplo de un recurso gratuito que ofrece componentes sueltos para poder construir un laboratorio entero.

²⁹ <https://assetstore.unity.com/packages/3d/technical-laboratory-designer-v1-room-93699>

³⁰ <https://assetstore.unity.com/packages/vfx/particles/pyro-particle-texture-pack-33161?locale=zh-CN>

5.2.4 Sistema de bonificación y penalización

Para garantizar que cada acción de los jugadores tenga consecuencia, el sistema de bonificación y penalización son esenciales para el juego.

El avatar de los jugadores dentro del juego tiene tres atributos decisivos para poder continuar el juego (puntos de sangre, estado de ánimo y tiempos restantes), entonces estos valores deben ser escalables, eso es fácil lograrlo con funciones de *setter* (función que asigna un valor) y funciones de *getter* (función que recibir un valor).

Por ejemplo, para retornar o modificar el valor de puntos de sangre es:

```
public void GetHp()
{
    //Función getter de puntos de sangre
    return CurrentHp;
}

public void AddHp(double i)
{
    //Función setter de puntos de sangre
    CurrentHp += i;
}

public void LossHp(double i)
{
    //Función setter de puntos de sangre
    CurrentHp -= i;
}
```

Y para el resto de los atributos son prácticamente iguales.

Un ejemplo de bonificación es que si el jugador activa la alarma, el juego mostrará el mensaje de la Figura 31.

Esto traducido en código sería:

```
AddTime (60)
```

En contrario, para el caso de penalización, si el jugador abre una puerta que no debería abrir, mostraría:

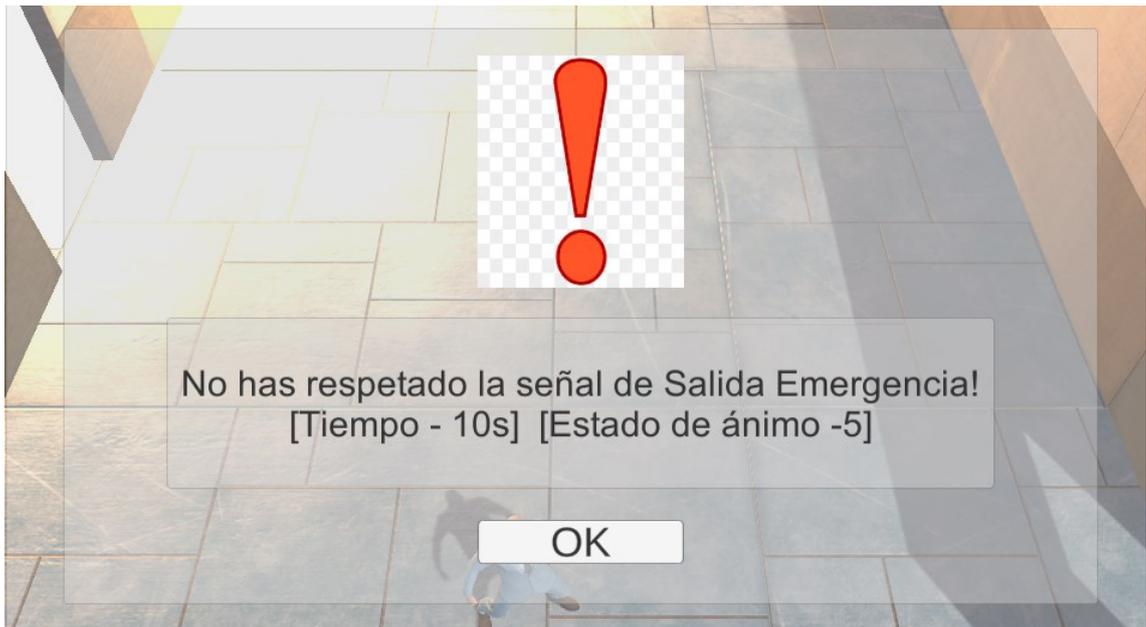


Figura 36. Notificación que aparece cuando el jugador no respeta la señal de ruta de evacuación.

Y esto en código es:

```
LossTime(10)
LossStress(5)
```

Y cuando el sistema comprueba cualquier de los tres atributos reducen a cero, directamente lleva el jugador a la escena del fin de juego.

```
if (GetHP() == 0 || GetStress() == 0 || GetTime() == 0)
    loadScene("Game Over")
```

5.2.5 Sistema de Inventarios

Como durante el transcurso del juego se encuentra con muchos objetos coleccionables, es óptimo crear una clase para guardarlos, para ello, se el tipo enumeración para definirlo, que es una herramienta potente para asociar un constante con un valor numérico, de esta forma se consigue códigos más limpios.

Para ello, hay que crear la clase *Items*.

```
public class Items
{
    public enum ItemType {
        ALARM,
        TOWEL,
        WET_TOWEL
    }
}
```

Por otra parte, al personaje también le asocia con una lista de objetos.

```
public class Character
{
```

```
private List<Items.ItemType> ItemList
}
```

De esta forma, cuando el jugador encuentra una toalla, verá la alerta de la figura.

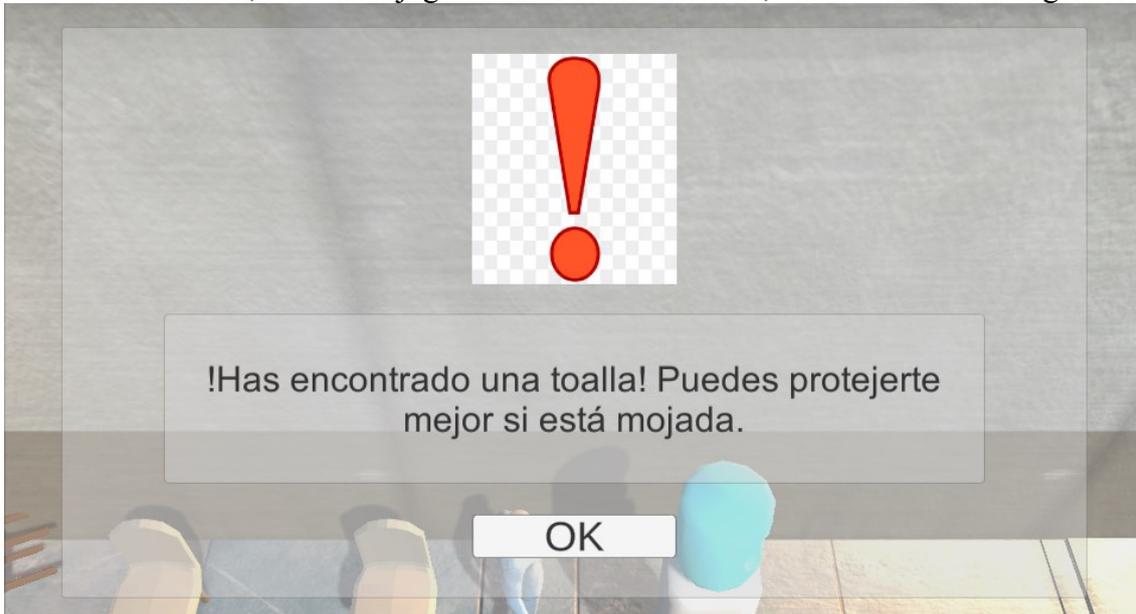


Figura 37. Notificación que aparece cuando encuentra una toalla en la escena.

Internamente, ejecuta:

```
addItem(Items.ItemType.TOWEL);
```

5.2.6 Sistema de visualización de temperaturas

La información de temperatura es muy importante para los jugadores en una simulación de incendio, sin embargo, es muy difícil transmitir dicha información en un videojuego, por lo tanto, hay que buscar una alternativa para informarlos.

Una de la solución es añadir un indicador de la temperatura, como muestra en las siguientes figuras:



Figura 38. Zona segura (No hay indicador).

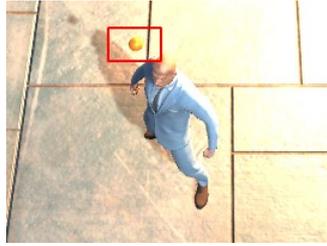


Figura 39. Zona alta temperatura (indicador en amarillo, remarcado con cuadro rojo).



Figura 40. Zona muy peligrosa (indicador en rojo, remarcado con cuadro rojo).

Se observa que, si el oficinista está en una zona segura, no se visualiza ningún indicador.

Pero cuando está cerca del fuego, la temperatura se aumenta y está recibiendo el daño del humo, se ve un punto amarillo al lado del oficinista, indicando que está entrando a una zona peligrosa.

Finalmente, si el oficinista ya está dentro del fuego, el indicador vuelve rojo para notificar que la situación es mucha emergencia y debe salirse de ahí lo antes posible. Por supuesto, también significa que está percibiendo una temperatura muy alta.

Este mecanismo es muy fácil de implementar gracias a la librería de colisionadores de Unity.

El *collider* (colisionador) sirve para detectar si dos objetos han sido colisionados (no tiene por qué ser físicamente). El problema es que, durante el transcurso del juego, es muy probable colisionar con otros objetos de la escena, la solución es filtrar los colisionadores por su tag, que es para identificar un grupo de *GameObject* con el mismo comportamiento.

Por lo tanto, se diseña un sistema que:

- Cuando el colisionador del jugador ha colisionado con otro *GameObject* con tag de *DangerFire*, aparece el indicador de la temperatura en color amarillo.
- Cuando el colisionador del jugador ha colisionado con otro *GameObject* con tag de *Fire*, aparece el indicador de la temperatura en color rojo.
- Cuando el colisionador del jugador deja de colisionado con *GameObject* con tag de *Fire*, aparece el indicador de la temperatura en color amarillo.
- Cuando el colisionador del jugador deja de colisionado con *GameObject* con tag de *DangerFire*, desaparece el indicador de la temperatura.

Haciendo esto posible hay que colocar un *asset* del fuego en la escena, y a este *asset* se asigna el tag de *Fire* y un *box collider* ajustando el tamaño de *asset*. Luego sobre ese

asset se asigna un objeto padre vacío, le asigna el tag de *DangerFire* y un colisionador más grande que el tamaño del fuego. Y para el personaje controlable se opera lo mismo, pero le asignando con un tag de *Player* y un *capsule collider* ya que tiene una forma de cápsula.

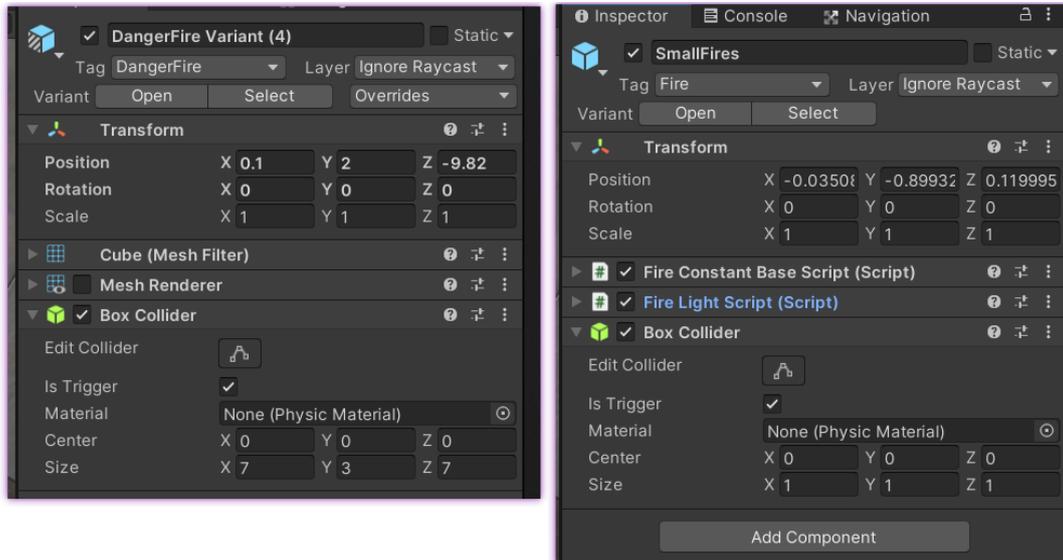


Figura 41. *Tags* y *colliders* que se asigna a zona alta temperatura (izquierda) y zona muy peligrosa (derecha).

La librería de colisionadores también contiene funciones que ayuda a detectar cuándo colisionan y cuándo dejan de colisionar.

```
private void OnTriggerEnter(Collider other)
{
    //Si hemos entrado a zona peligro
    if (other.tag == "Fire")
        EnableDetector (Red)
    else if (other.tag == "DangerFire")
        EnableDetector (Yellow)
}

private void OnTriggerExit(Collider other)
{
    //Si hemos salido de zona peligro
    if (other.tag == "Fire")
        EnableDetector (Yellow)
    else if (other.tag == "DangerFire")
        DisableDetector ()
}
```

Visualmente se observa que los *colliders* son así:

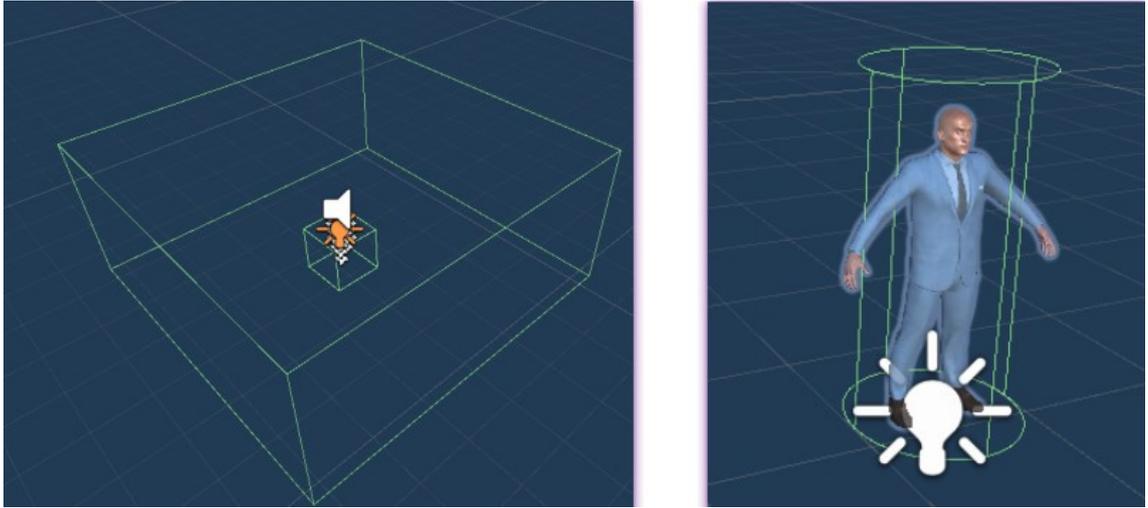


Figura 42. *Collider* del fuego (izquierda) y *collider* del personaje (derecha).

Y cuando *colliders* penetran entre sí:

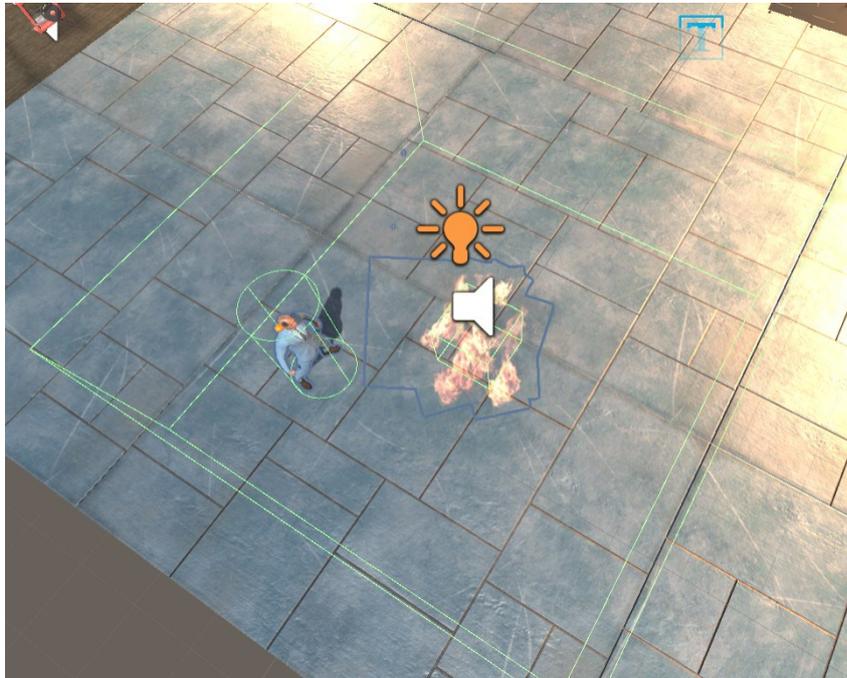


Figura 43. Cuando *collider* del fuego penetra con el *collider* del personaje.

Con todo eso, consigue a crear algunas puertas calientes que no se debe abrir:

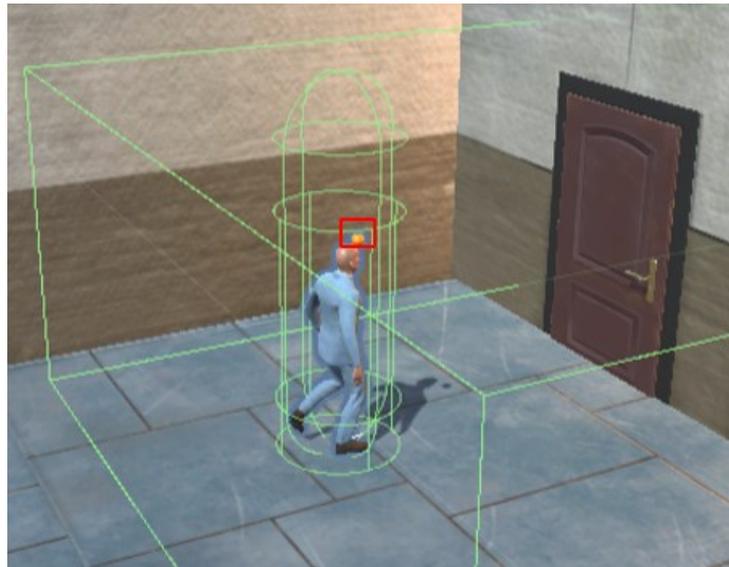


Figura 44. Cuando se asigna *collider* y *tag* de zona caliente para una puerta, fija que aparece el indicador de alta temperatura (remarcado con cuadro rojo).

Como se ve en la Figura 44, aunque el jugador no ve ningún fuego cerca, sigue pudiendo saber que la puerta tiene temperatura alta.

5.2.7 Sistema de las preguntas

Para que el jugador realmente pueda tomar decisiones durante el transcurso del juego, se introduce el sistema de las preguntas.



Figura 45. Ejemplo de una pregunta.

Para implementar este sistema hay que volver a utilizar *Canvas* ya que se puede aprovechar muchos mecanismos de *Canvas* como el cuadro de texto, las barras de progreso y los botones.

La interfaz de preguntas está formada por cinco partes principales:

- Un contador para que los jugadores siempre tengan en cuenta el tiempo que queda.
- Las barras de estado ya que tampoco se debe perder la vista.
- Una imagen del personaje controlando.
- Un título de la pregunta.
- Tres posibles respuestas.

Para simplificar el proceso de crear una pregunta, he creado una función que facilita su creación:

```
SetQuiz(int id,string quiz,string optionA,string optionB,string optionC,int correctOption)
```

Sus parámetros son:

- *id*: El identificador de la pregunta, para poder tratarlo individualmente posteriormente.
- *quiz*: El texto de la pregunta.
- *optionA*: El texto de la opción A.
- *optionB*: El texto de la opción B.
- *optionC*: El texto de la opción C.
- *correctOption*: La opción correcta de rango {1,2,3} que corresponde {A,B,C}.

Por ejemplo:

```
SetQuiz(0,"2+2=?", "3", "4", "5", 2)
```

Con esto se lanza una pregunta con identificador 0, y preguntar por cuál es el resultado de $2 + 2$, el jugador debe elegir entre 3, 4 o 5, y poner que la opción B es correcta que corresponde al número 4.

Si el jugador ha elegido la opción incorrecta, entonces se penaliza quitándole un poco el punto de estado de ánimo, y le deja volver a elegir, en el caso contrario, desaparece la interfaz del sistema de preguntas y el jugador puede continuar.

Con eso se obtiene un prototipo básico del sistema, pero se puede profundizar aún más para mejorar la jugabilidad.

La primera extensión es combinar con el sistema de inventario.

Por ejemplo, una de las preguntas requiere que el jugador tapa a los agujeros de la puerta con una toalla mojada, como disponemos un identificador para cada objeto, se puede añadir una comprobación extra para que el jugador aparte de tener que elegir la opción correcta también hay que tener objeto requerido, si no, también se considera como si ha fracasado y recibe penalización.



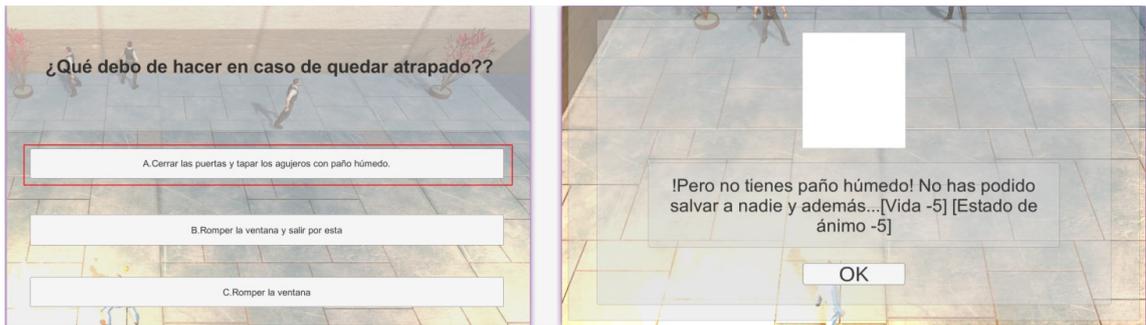


Figura 46. Ejemplo de contestar bien la pregunta, pero recibe penalización.

Para lograr eso, hay que extender un poco la función de antes:

```
SetQuiz(int id, string quiz, string optionA, string optionB, string optionC, int correctOption, QuizExtraInfo extra)
```

Se añade un parámetro que se llama *extra* de clase *QuizExtraInfo*. Esta clase contiene muchas funcionalidades para extender la funcionalidad del sistema de las preguntas, entre ellos se destaca:

Class QuizExtraInfo

```
public Items.ItemType[] ItemsToCheck;

//Penalizaciones de tiempo, puntos de sangre, estado de ánimo
public int timeToLoss;
public float hearthToLoss;
public float stressToLoss;

public string infoToShowOnFail; //Texto que muestra cuando falla
```

- **ItemsToCheck:** Lista de objetos que requiere para la pregunta.
- **timeToLoss:** La penalización del tiempo.
- **hearthToLoss:** La penalización del punto de sangre.
- **stressToLoss:** La penalización del estado de ánimo
- **infoToShowOnFail:** El texto que muestra cuando falla.

Con todo eso ya se puede replicar el caso de la Figura 46.

Y la segunda extensión es añadir un tiempo límite para preguntas concretas, ya que puede encontrar ante una situación crítica como hay que salvar a alguien en llamas.



Figura 47. Preguntas con tiempo límite.

Para ello hay que volver a expandir la clase QuizExtraInfo añadiendo un atributo más:

```

Class QuizExtraInfo
public double[] TimeLimits

```

- TimeLimits: Lista de tiempos restantes para la pregunta.

La implementación del sistema del tiempo está en el punto 5.2.8.

5.2.8 Sistema del tiempo límite

Para que el juego sea más estimulante, se añade un descontador en la parte superior derecha la pantalla notificando el tiempo restante.



Figura 48. Tiempo restante (remarcado con cuadro rojo).

Para disminuir el coste computacional y mejorar el rendimiento del juego, se utiliza la corrutina para el trabajo de descontar tiempo.

El uso de la corrutina es una técnica para aliviar un proceso y reducir la sobrecarga de la unidad central de procesamiento. En Unity, es tan fácil como llamar las siguientes funciones:

```
StartCoroutine(countDown())
IEnumerator countDown()
```

Con eso Unity se encargará de ejecutar la función *countDown* en su totalidad antes de retornar.

5.2.9 Página web

Se destaca dos cambios principales:

En primer lugar, se reestructura y refactoriza la página web para mejorar el redimiendo, también se añade una página principal, el cual sirve como un portal para presentar al usuario todos los servicios que ofrece esta página web.

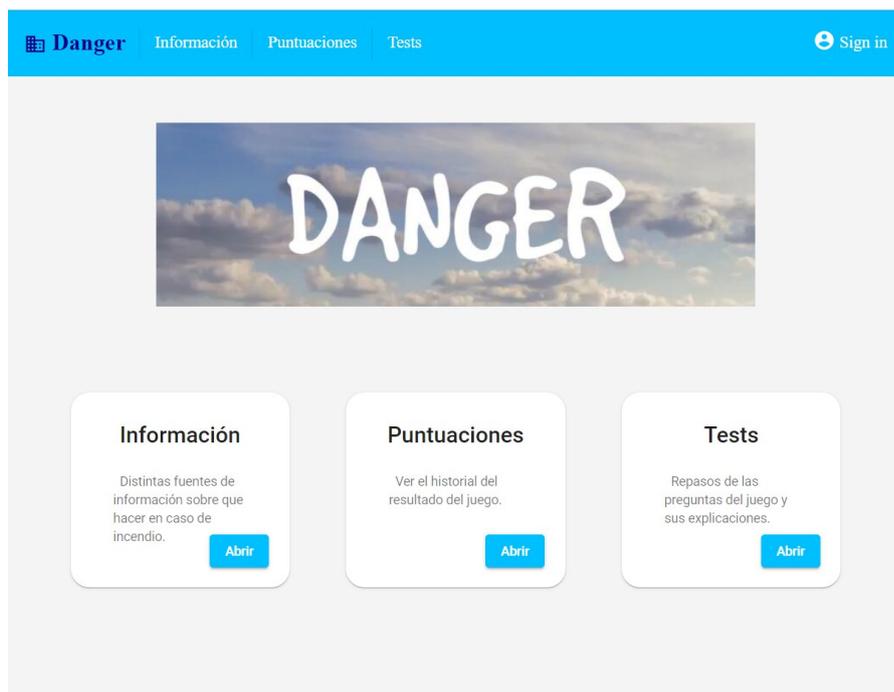


Figura 49. Página principal de la web.

Tal como presentaba el punto 4, Adrián Sánchez Lavarias y Pablo Querol Ballester crearon funcionalidades de:

- Información: Acceder distintas fuentes de información sobre qué hacer en caso de incendio.
- Puntuaciones: Ver el historial del resultado del juego.

Y en este proyecto se incorpora la nueva funcionalidad “Test”, que consiste en hacer repasos de las preguntas del juego sin tener que abrirlo. También dispone un botón de “Mostrar respuestas” que sirve para ver las respuestas junto con una breve explicación con las referencias para que los jugadores puedan entender mejor porqué la respuesta es así.

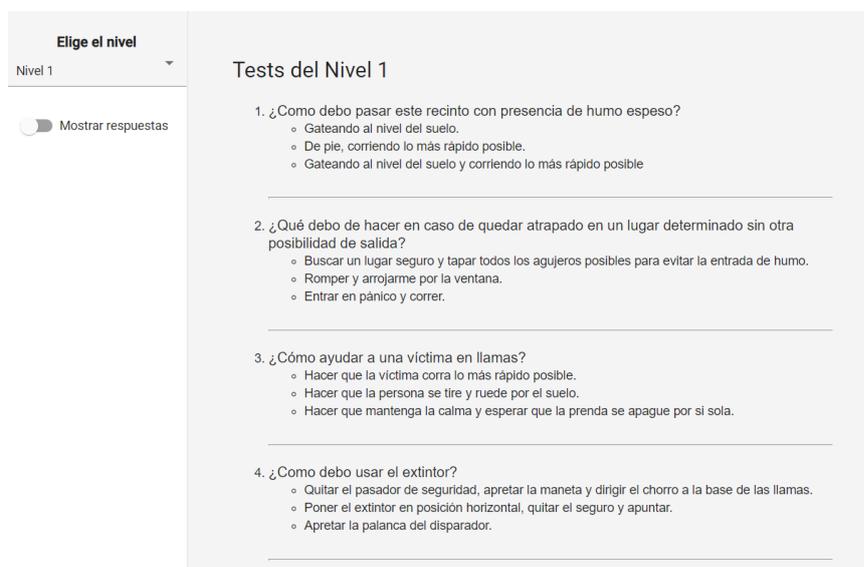


Figura 50. Repaso de las preguntas sin marcar respuestas.

Elige el nivel
Nivel 1

Mostrar respuestas

Tests del Nivel 1

1. ¿Como debo pasar este recinto con presencia de humo espeso?

- Gateando al nivel del suelo.
- De pie, corriendo lo más rápido posible.
- Gateando al nivel del suelo y corriendo lo más rápido posible

Explicación:
Cúbra nariz y boca con alguna prenda o paño. Si la circunstancias lo permite humidézcalo (Sánchez López, 2018). Respirar aire caliente es mortal porque quema las vías respiratorias. Gatear a nivel del suelo es vital porque el calor y el humo de desplazan y se mantienen en la parte superior lo cual permite que a nivel del suelo el aire este relativamente más limpio (BBC Mundo, 2017).

2. ¿Qué debo de hacer en caso de quedar atrapado en un lugar determinado sin otra posibilidad de salida?

- Buscar un lugar seguro y tapar todos los agujeros posibles para evitar la entrada de humo.
- Romper y arrojarme por la ventana.
- Entrar en pánico y correr.

Explicación:
En caso de quedar atrapado y no poder salir de un sitio: 1. Tapa rendijas u orificios con paños o cualquier prenda que tengas a la mano y si es posible humidécela, para evitar la entrada de humo en la habitación (Fundación MAPFRE, 2021). 2. Cuando hayas cubierto todos los agujeros, mantén la calma y llama al 112 para informar de tu situación. 3. Presta atención: que personas no se pueden evacuar por sus propios medios, también asegúrate que todos han salido. En caso tal que alguien no pueda salir avisa a emergencias (Sánchez López, 2018).

Figura 51. Repaso de las preguntas visualizando las respuestas.

En segundo lugar, se reorganiza estructura de los archivos para facilitar el proceso para lanzar la página web.

Para empezar, se debe tener instalado Nodejs para poder tener un entorno de ejecución para JavaScript. Para comprobar si Nodejs se ha instado correctamente, puede utilizar siguiente comando:

```
node -v
```

También es necesario instalar CLI de Angular con el comando.

```
npm install -g @angular/cli
```

Luego ejecutar:

```
npm install
```

Este comando sirve para acceder al archivo para descargar todas las dependencias necesarias para lanzar el web en nuestro local, estas dependencias se encuentran definidas en el archivo *package.json*.

Finalmente, lanzar la página web con el siguiente comando:

```
ng server -open
```

Con esto, automática abre un navegador configurado por defecto y accede a <http://localhost:4200/> para visualizar la página web.

6. Conclusiones

6.1 Comparación con la última versión (changelog)

A continuación, se procede a analizar los cambios, las mejoras y las nuevas funcionalidades que se han implementado.

- Se ha añadido un nuevo modo de juego, la escena se construye de forma manual.
- Se ha añadido un tutorial antes de entrar al juego.
- Se ha cambiado las barras de estado, ahora es más estético y siempre muestra por la pantalla.
- Se ha sustituido la barra de estrés por la barra de estado de ánimo y hemos cambiado su comportamiento.
- Ahora cuando indicamos el destino del personaje haciendo un clic en el suelo, aparece una flecha como retroalimentación.
- Ahora cuando poner el cursor sobre algo que se puede interactuar, cambia el gráfico del puntero.
- Ahora el juego puede comunicarse con el jugador a través de una ventana de diálogo.
- Hemos añadido objetos coleccionables en la escena.
- Ahora el personaje cuando recibe el daño desde el fuego se notifica.
- El daño recibido por acercarse al fuego ya no es constante, sino que depende de tu inventario (Por ejemplo, si tapas la nariz y boca con una toalla húmeda recibirás menos daños).
- Ahora cuando superas el nivel, el sistema te hace un informe del resultado.
- Ahora cuando el personaje muere, te lleva a la pantalla de fin del juego.
- Ahora el personaje cuando recibe el daño desde el fuego se notifica.
- Ahora el personaje puede interactuar con el entorno.
- Ahora el personaje tiene que contestar las preguntas para poder avanzar el juego.
- Ahora el personaje tiene que contestar las preguntas para poder salvar a otras personas.
- Ahora el jugador debe escapar antes de superar el tiempo límite.
- Se ha añadido diferentes puertas, no todas las puertas se deben abrir, el jugador debe pensarlo bien antes de abrirlas.
- Se ha añadido nuevos personajes.
- Se ha añadido nuevas animaciones.

6.2 Relación con los estudios

Al tratar de diseñar un videojuego con tanta complejidad, exige conocimientos muy completos sobre el procedimiento de diseño de un software. En esta sección, presentamos conocimientos requeridos en cada fase y su relación con los estudios cursados.

En primer lugar, tenemos la fase de diseño, para especificar más, podemos dividirla en: diseño del videojuego, diseño del software y gestión de proyecto.

En la fase de diseño de videojuego se pretende elaborar un documento de diseño del juego, aquí documentamos los aspectos principales del juego como: tipo de juego, historia, diseño de niveles, diseño de interfaces...etcétera. Por lo tanto, es sumamente



importante las asignaturas como Interfaces persona computador e Introducción de la programación de videojuego.

Luego en la fase de diseño de software, se debe planificar y diseñar un diagrama de clase con el lenguaje unificado de modelado (también conocido como UML), en dicho diagrama, debe especificar las clases del sistema, sus atributos, los métodos y las relaciones entre las clases. Este conocimiento se imparte en los temarios de la Ingeniería de software y Gestión de bases de datos.

Finalmente, también es importante la gestión de proyecto, en esta fase debemos determinar la metodología de desarrollo de software, por ejemplo, se puede hacer de forma de cascada, forma espiral, forma cíclica...etcétera. También vamos a desarrollar un diagrama de Gantt para dividir tareas en actividades más pequeñas, luego estimar tiempo que vamos a dedicar en cada subtarea. Adicionalmente hay que elegir un controlador de versión para una mejor colaboración, en nuestro caso hemos elegido a Github. Todo lo anterior se estudia en la asignatura de Ingeniería de software y Gestión de proyecto.

En segundo lugar, llegamos al desarrollo del proyecto, en esta parte tenemos el desarrollo de la página web y el desarrollo del videojuego.

Para el desarrollo de la web, se requiere el manejo del *framework* Angular, para ello, es necesario saber la programación en TypeScript y uso de HTML complementando con CSS. Parte de ese conocimiento lo vemos en la asignatura Tecnología de sistemas de información en la red.

Y para el desarrollo del videojuego, hay que saber el uso de la herramienta Unity y la programación en C#, por lo tanto, es necesario aplicar conocimientos aprendidos en la asignatura de Programación e Introducción a la programación de videojuego.

6.3 Conclusiones y trabajos futuros

En conclusión, estoy muy satisfecho con el resultado del juego. Como indicaba en el principio de esta memoria, es un proyecto largo y es parecido a una carrera de relevo. He corrido todas mis trayectorias y es el momento de traspasar al siguiente compañero. En los apartados anteriores he organizado todo que hecho durante mi “recorrido”, ahora vemos qué nos falta para completar la carrera:

Para empezar, la estética. Como no hay el responsable de arte en el equipo, no hay más remedio que utilizar recursos de licencias abiertas de internet, por consecuente, no hay estilo de arte y la calidad de la gráfica tampoco es llamativa, además, las animaciones también se pueden mejorar.

En segundo lugar, la jugabilidad. En esta versión hay intentos de transmitir el conocimiento en forma de preguntas y respuestas, pero desde mi punto de vista, sigue siendo muy monótono y los jugadores puede perder sus intereses rápidamente, lo ideal es diseñar niveles más interactivos haciendo que los jugadores puedan usar todos lo que ha aprendidos en el juego con la finalidad de consolidar el conocimiento.

En tercer lugar, la colección de datos. Es preferible coleccionar más datos dentro del marco establecido por la ley, ya que tiene dos grandes usos:

- Es útil para el jugador, se puede generar informe personalizado para cada jugador, indicándoles qué han hecho bien y dónde se puede mejorar.

- Es útil para la sociedad, se puede analizar el comportamiento de todos los jugadores, eso ayuda mucho hacer una estadística y estudiar de ella, podemos sacar informaciones como: qué error suele cometer la gente, cómo podemos hacer que la gente evite estos errores...etcétera.

Y finalmente, también hay que completar y mejorar otros modos de juego:

- Para el modo *Build*
 - Hay que mejorar las interfaces para que sea más fácil de utilizar.
 - Introducir más componentes para construir.
 - Análisis o informe sobre el resultado de la construcción.
- Para el modo *Escape*
 - Incorporar las nuevas funcionalidades introducidos en el modo Adventure.
 - Mejorar el algoritmo de generación de la escena.



7. Agradecimientos

Principalmente quiero agradecer a mi amigo Hui Yuan, que sin él, yo no estaría tan interesado en el tema de videojuego.

También quiero agradecer a Ramón y Teresa por ayudarme durante este periodo de tiempo.

A mi novia Cai que se ha dedicado un montón de tiempo para explicarme la programación en TypeScript y características de Angular.

A Yennifer por ayudarme con el tema de prevención de riesgo y haber diseñado un nivel de alta calidad.

A Pablo Querol y Adrián Sanchez por haber realizado un trabajo tan excelente.

A mis compañeros de equipo de la asignatura “Programación e Introducción a la programación de videojuego”, Jesús García Sastre y Pepe Zhou por tener buena experiencia de trabajar juntos con un videojuego y me dan confianza en poder terminar este trabajo.

Y sobre todo a mis familiares que por todos los esfuerzos para que yo pueda estar aquí.

8. Bibliografía

APTB, & Fundación Mapfre. (2019). *VÍCTIMAS DE INCENDIOS EN ESPAÑA EN 2019*. Obtenido de https://www.tecnifuego.org/recursos/arxius/20201217_11062020_Estudio_victimas_incendios_en_2019_APTB_y_MAPFRE.pdf

Blumberg, F. C. (2014). *Learning by Playing: Video Gaming in Education*. Obtenido de <https://oxford.universitypressscholarship.com/view/10.1093/acprof:osobl/9780199896646.001.0001/acprof-9780199896646>

Crawford, C. (1992). Chris Crawford Dragon Speech "I Had a Dream". Obtenido de <https://www.youtube.com/watch?v=VwZi58u1FjI>

Crawford, C. (1997). *The Art of Computer Game Design*. Obtenido de https://www.digitpress.com/library/books/book_art_of_computer_game_design.pdf

Gómez Tringueros, I., & Ortega Sánchez, D. (2019). *Gamification, social problems, and gender in the teaching of social sciences: Representations and discourse of trainee teachers*. Obtenido de <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0218869>

Kevin Werbach, & Dan Hunter. (2012). *For the Win: How Game Thinking can Revolutionize your Business*. Obtenido de https://www.researchgate.net/publication/273946893_For_the_Win_How_Game_Thinking_can_Revolutionize_your_Business

Maiga Chang, Rita Kuo, Kinshuk, Gwo-Dong Chen, & Michitaka Hirose. (2009). *Learning by Playing: Game-based Education System*. Obtenido de <https://link.springer.com/book/10.1007/978-3-642-03364-3>

Querol Ballester, P. (2021). *D.A.N.G.E.R - Disaster Prevention Videojuego educativo para la gestión de emergencias en Unity 3D : IA, Agentes Inteligentes y Generación Procedural*. Obtenido de <https://riunet.upv.es/handle/10251/174568>

Sánchez Lavarias, A. (2021). *D.A.N.G.E.R. Disaster Prevention: videojuego educativo para la gestión de emergencias en Unity 3D: interfaz de usuario, gestión y tratamiento de la información*. Obtenido de <https://riunet.upv.es/handle/10251/174398>

Volkswagen. (2009). *The Fun Theory*. Obtenido de <http://www.thefuntheory.com/>

Wendy Hsin-Yuan Huang, & Dilip Soman. (2013). *A Practitioner's Guide To Gamification Of Education*. Obtenido de https://www.academia.edu/33219783/A_Practitioners_Guide_To_Gamification_Of_Education



Anexo I: Glosario

- **Assets:** Son objetos de Unity que podemos utilizar en nuestro juego, puede ser un modelo 3D, imagen 2D, un archivo de audio...etc.
- **Avatares:** Son personajes en el videojuego controlado por el jugador.
- **Engine:** También conocido como Motor de juego, es un software que facilita el diseño y la creación de un videojuego.
- **Escena:** Una escena en Unity es el lugar donde vamos a representar nuestro juego, puede crear cualquier número de escena en el juego.
- **Estrategia:** (No tengo muy claro que tiene que ver esto con el juego XD)
- **Game Over:** Es una escena del videojuego, y cuando aparece cuando el juego ha terminado cuando el jugador se ha fracasado o ha terminado el juego.
- **GameObject:** Son objetos en Unity que representan personajes, objetos y el escenario.
- **Hardware:** Son partes tangibles de un sistema informático un juego puede presentar diferentes comportamientos (mejor o peor) en diferente hardware.
- **UI:** También conocido como interfaz de usuario. Son paneles visuales para poder comunicar o interactuar con el usuario.
- **Juego:** Actividad con o sin apoyo de las herramientas con la finalidad de entretenerse respetando las reglas.
- **Niveles:** Desafío de un juego a los jugadores. Los niveles suelen estar asociado con un número, dicho número suele tener una relación proporcional con la dificultad.
- **Objetivos:** Es el propósito concreto que determine lo que el jugador debe hacer para superar el nivel.
- **Página Web:** Son archivos guardados en los servidores, tiene una dirección asociada para poder navegar desde cualquier sitio.
- **Prefab:** Un Prefab es un tipo de Assets de Unity que permite crear, configurar y guardar un GameObject con todos sus valores, propiedades, componentes hijos...etc.
- **Raycast:** Un Raycast en Unity es una función de física que proyecta un rayo en la escena.
- **Reglas:** Norma que regula el comportamiento de los jugadores.
- **Software:** Son partes intangibles de un sistema informático, formado por códigos y algoritmos.
- **Usuario:** Son las personas que adquieren la copia de software, también denominamos jugadores en el caso de videojuego.
- **Videojuego:** Son juegos con herramienta (terminal) y representado en forma de software.

Anexo II: Documento del diseño del juego

SECCIÓN 1: DATOS PRINCIPALES

- **Título:** “D.A.N.G.E.R – Disaster Prevention”
- **Concepto abstracto:** Danger es un juego con propósito pedagógico con el que aprender normativas y procedimientos aplicables a la prevención de incendios.
- **Tema:** Sociedad actual, gestión de emergencia, simulación.
- **Género:** Construcción y estrategia en tiempo real.
- **Plataforma:** Windows/Android/IOS.
- **Mercado:** Todos los públicos.
- **PEGI:** Para 7 años o más. Intentamos hacer para que el juego sea aceptable para todos los rangos de edad, aun así, pueda haber unas escenas que muestra crueldad de los incendios, y no son muy apropiados para niños menor que 7 años.



FIGURA GDD 1. PEGI 7

Equipo:

1. **Desarrolladores:**

- 1.1. Adrián Sánchez Lavarias (email: adsanla@inf.upv.es)
- 1.2. Pablo Querol Ballester (email: pabqueba@inf.upv.es)
- 1.3. Xinyu Jiang (email: xinjia@inf.upv.es)

2. **Consultores técnicos:**

- 2.1. Yennifer Ramirez Lopez (email: yenralo@posgrado.upv.es)

SECCION 2: Pitch doc

Resumen:

En juego puedes elegir entre actuar como el diseñador de un edificio (modo construcción) o un público que accidentalmente queda atrapado en un incendio (modo estrategia o aventura).

Por lo tanto, cuando juegas como un diseñador, podrás construir salas y ponerlas a prueba para comprobar si cumplen con la normativa.

Y si quedas atrapado en un edificio en llamas, hay que guiarlas para que logren escapar antes de que les alcance el fuego.

Gameplay:

Mezclará los géneros de construcción, puzzle y estrategia en tiempo real en tercera persona, enfatizando en la toma de decisiones a lo largo del videojuego

En modo de construcción, tendrás que aprovechar todos los bloques y objetos que te ofrecen, arrastrarlas (Con ratón o pantalla táctil) para crear viviendas u oficinas que cumplen la normativa de presión de incendio.

Es modo de estrategia de tiempo real, también tienes que usar ratón o pantalla táctil para indicar el destino de los personajes para que pueda salir de la situación de emergencia. En este modo también encontrarán una serie de obstáculos como puzzles o preguntas que hay que resolver para poder continuar con la evacuación.

Para el modo de estrategia de tiempo real se dispone dos formas de representar la escena:

- Escena generada proceduralmente: Escenas aleatorias creado con algoritmos, por lo tanto, cada vez que entra a jugar es un nuevo nivel, para mantener intereses de los jugadores (modo escape).
- Escena construida manualmente: Niveles prediseñados, servirán tanto para niveles introductorios como niveles finamente diseñados para garantizar dificultades razonables (modo aventura).

Finalidad del juego:

El objetivo es hacer un juego:

- Interactivo en 3D, con el fin de enseñar la normativa y metodología en caso de incendio. Para ello se han creado 3 modos de juego de los géneros construcción con puzzles y estrategia en tiempo real. Lo que permite desarrollar la creatividad del usuario y colocarlo en una simulación de una situación real.

- Diferenciador, la dualidad de modos de juego que desarrollan la creatividad, la capacidad de estrategia y reacción del jugador. Ambos modos de juegos, además, fomentan la rejugarabilidad del título.

Estética:

El juego será en 3D, el modo construcción será en primera persona, mientras que el modo construcción y aventura es de tercera persona. Los tres modos del juego presentarán la misma estética.

Dado que la escena del juego es el mundo real y tiene un tema serio, por lo tanto, los personajes, las escenas y las animaciones del juego será realista acortando todos los detalles con el mundo real.



FIGURA GDD 2. Imagen conceptual, fuente GTA 5

SECCIÓN 3: Análisis competitivo

Referentes

Nuestras competencias serán todas las metodologías para realizar la formación de prevención de riesgo contra los incendios.

A grandes rasgos, está:

- Fuera de contexto del juego: simulacro de incendios.
- Juegos/Juegos de mesas.
- Videojuegos.
- Producto de realidad virtual.

Se procede a realizar análisis DAFO para hacer una comparativa con otro producto:

- **Simulacro de incendios.**

	De origen interno	De origen externo
N e g a t i v o	<p>Debilidades:</p> <ul style="list-style-type: none"> - Requiere que el usuario dispone interés con el videojuego. - El usuario puede dejar de jugar. 	<p>Amenazas:</p> <ul style="list-style-type: none"> - El usuario no puede revisar su equipamiento real contra incendio. - A veces es menos eficaz que un simulacro.
P o s i t i v o	<p>Fortalezas:</p> <ul style="list-style-type: none"> - Ofrece una experiencia realista. - Fácil de adquirir. - Fácil de manejar. - Dispone recursos complementarios. - Incide competición. 	<p>Oportunidades</p> <ul style="list-style-type: none"> - Puede jugar en cualquier momento. - Puede entrenar en diferentes escenas. - Puede enfrentar casos imprevistos.

FIGURA GDD 3. Análisis DAFO de Danger con simulacro (Desde la perspectiva de Danger).

- **Juegos/Juegos de mesas:** The Great Fire of London 1666.



Figura GDD 4. The Great Fire of London 1666.

	De origen interno	De origen externo
N e g a t i v o	<p>Debilidades:</p> <ul style="list-style-type: none"> - Requiere que el usuario dispone interés con el videojuego. - El usuario puede dejar de jugar. 	<p>Amenazas:</p> <ul style="list-style-type: none"> - Es menos divertido, ya que normalmente se juega solo.
P o s i t i v o	<p>Fortalezas:</p> <ul style="list-style-type: none"> - Ofrece una experiencia realista. - Fácil de adquirir. - Fácil de manejar. - Dispone recursos complementarios. - Incide competición. 	<p>Oportunidades</p> <ul style="list-style-type: none"> - Danger está basado en sociedad actual. - Normalmente enfrentas con problemas reales, mientras que enfrentar la situación de <i>The Great Fire of London 1666</i> es muy poco probable.

FIGURA GDD 5. Análisis DAFO de Danger con The Great Fire of London 1666 (Desde la perspectiva de Danger).

- **Videojuegos: Flame Over.**

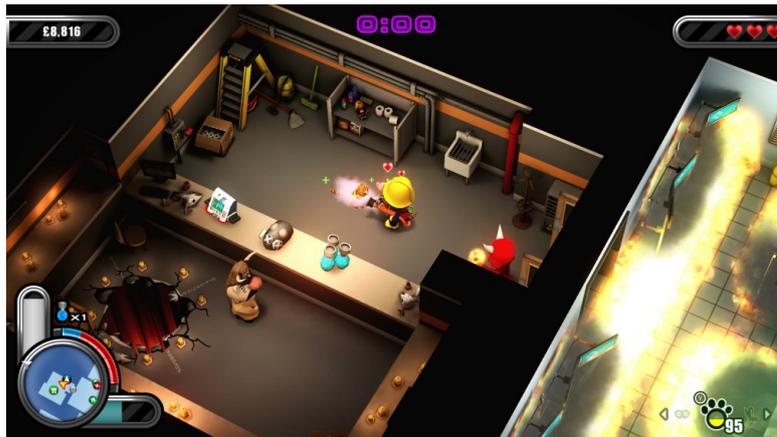


Figura GDD 6. Flame Over.

	De origen interno	De origen externo
N e g a t i v o	<p>Debilidades:</p> <ul style="list-style-type: none"> - Requiere que el usuario dispone interés con el videojuego. - El usuario puede dejar de jugar. 	<p>Amenazas:</p> <ul style="list-style-type: none"> - En Danger no puedes actuar como un bombero. - La gráfica es menos llamativa.
P o s i t i v o	<p>Fortalezas:</p> <ul style="list-style-type: none"> - Ofrece una experiencia realista. - Fácil de adquirir. - Fácil de manejar. - Dispone recursos complementarios. - Incide competición. 	<p>Oportunidades</p> <ul style="list-style-type: none"> - Dispone gráfica más realista. - Dispone un sistema de preguntas y respuestas para consolidar el conocimiento.

FIGURA GDD 7. Análisis DAFO de Danger con Flame Over (Desde la perspectiva de Danger).

- **Producto de realidad virtual: Ludus.**



Figura GDD 8. Ludus.

	De origen interno	De origen externo
N e g a t i v o	<p>Debilidades:</p> <ul style="list-style-type: none"> - Requiere que el usuario dispone interés con el videojuego. - El usuario puede dejar de jugar. 	<p>Amenazas:</p> <ul style="list-style-type: none"> - Danger es menos inmersivo comparando un producto de realidad virtual. - Es menos interactivo comparando un producto de realidad virtual.
P o s i t i v o	<p>Fortalezas:</p> <ul style="list-style-type: none"> - Ofrece una experiencia realista. - Fácil de adquirir. - Fácil de manejar. - Dispone recursos complementarios. - Incide competición. 	<p>Oportunidades</p> <ul style="list-style-type: none"> - Dispone niveles que se genera proceduralmente, mientras que Ludus solamente dispone un nivel. - Es más barato, no requiere cascos de realidad virtual.

FIGURA GDD 9. Análisis DAFO de Danger con Ludus (Desde la perspectiva de Danger).

SECCIÓN 4: Estructura narrativa, personajes

Estructura narrativa:

Los modos de construcción y escape son modos sin narrativas, el jugador simplemente puede elegir el modo y empezar a disfrutar el juego.

En el modo de aventura, tampoco hay una historia principal, pero sí que hay descripciones de lo que está sucediendo en la escena y también se explica cuáles son los objetivos de los jugadores.

Personajes:

Modo	Nombre	Descripción	Foto
Modo construcción	Protagonista	El protagonista del modo de construcción.	No visible dentro del juego
Modo escape	Protagonista	El protagonista del modo de aventura.	
Modo aventura	Protagonista – Nivel 1	Un oficinista queda atrapado en un edificio, tienes que ayudarlo a escapar.	

<p>Modo aventura</p>	<p>Personajes no controlables – Nivel 1</p>	<p>Compañeros de trabajo del protagonista, puedes decidir si quieres (o si eres capaz) de salvarles.</p>	
<p>Todos los modos</p>	<p>Puede aparecer más personajes</p>		

SECCIÓN 5: Escenas, Niveles

Escenas

El juego comienza con una Cinemática introductoria, se puede esperar o saltar con la tecla espacio o escape. Sirve como primer contacto con el jugador, por lo que hemos creado una escena móvil con música para introducir al jugador a nuestro juego.

Una vez finalizado el Splash nos encontraremos en el menú principal, desde aquí podemos acceder a las demás escenas.

Desde menú principal puedes acceder a diferentes opciones: créditos, configuración, modo construcción, modo estrategia, modo aventura.

Configuración: Esta escena servirá para que el jugador pueda cambiar entre pantalla completa o ventana, modificar el volumen, los gráficos del videojuego o la resolución a la que se muestra. Con el botón “back” volveremos al menú principal.

Créditos: En esta escena hemos colocado las distintas tareas de las que nos hemos encargado cada uno en el proyecto en un fondo negro con música. Esta escena conecta con el Splash para que la navegación sea completa.

Modo construcción: Este es nuestro primer modo jugable, mantiene la música de las anteriores escenas, se compone por un entorno interactivo en el que construir salas de cualquier forma y tamaño.

Modo estrategia: Para nuestro segundo modo, contamos con un número de salas generadas de forma procedural, donde encontraremos un modo de juego de estrategia en tiempo real.

Modo aventura: El tercer modo, es parecido como modo de estrategia, pero con niveles predefinidos.

Este es un diagrama de flujo para ver relaciones entre las escenas:

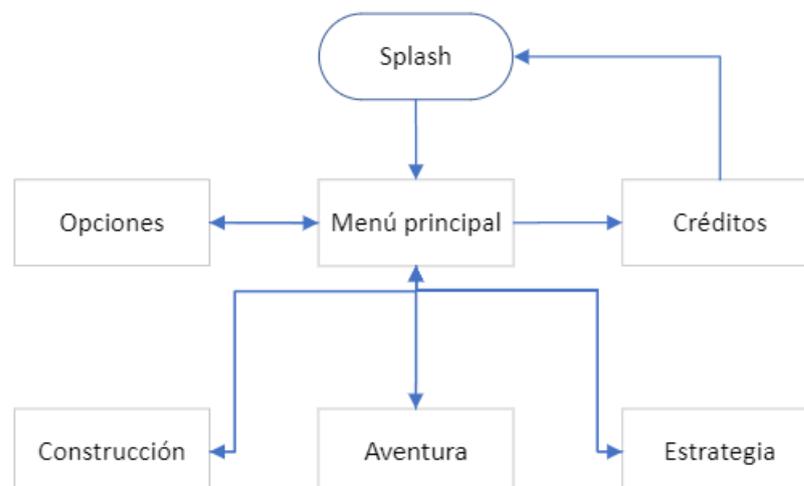


FIGURA GDD 10. Diagrama de flujo.

Modo construcción

En el modo de juego “build” encontramos un nivel con un solo bloque base desde donde el jugador comenzará a crear el conjunto de salas deseado. Estas salas tendrán unos requisitos dependiendo del tamaño total de las mismas que deben cumplirse para finalizar el nivel.

Elementos jugables:

- Construcción de escena: Elegir bloque y construir una habitación de tu imaginación.



FIGURA GDD 11. Una habitación en construcción.

- Verificación de resultado: Con el botón “Scan” compruebas si tu habitación cumple requisitos.



FIGURA GDD 12. Comprobando si la habitación cumple requisitos con el botón Scan.

Nivel:

El juego ofrece un “sandbox” que permite a los jugadores construir cualquier escena y luego comprobar y cumple la normativa, por lo tanto, no hay niveles explícitos.

Modo estrategia

Los niveles del modo “escape” se generan de forma procedural, por lo que cada vez que se ejecuta el juego es diferente. Sin embargo, todos comparten que son salas conexas entre sí con forma rectangular poblada con mobiliario de oficina.

Elementos jugables:

- Control de personajes: Controla tu personaje para escapar de aquí.

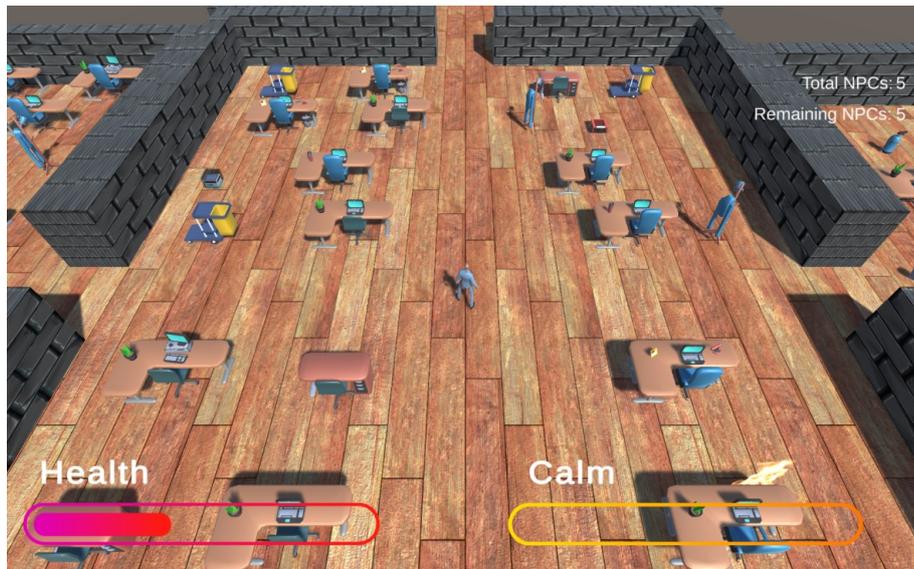
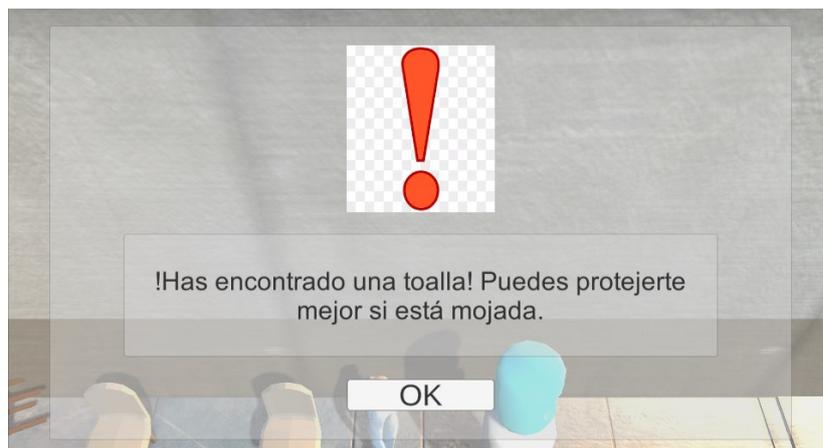


FIGURA GDD 13. Personaje que se encuentra en una oficina.

- Interacción con el entorno: ¡Interacciona y colecciona los objetos, te puede salvar la vida!



- Gestión de situación de emergencia: Aprovecha todos los elementos del entorno y tu inventario para salir de la emergencia.



FIGURA GDD 14. Personaje rodeado por la llama.

- Contestar las preguntas: Ante la situación de emergencia, debes tomar decisión correcta para poder avanzar.



Nivel:

Gracias al algoritmo de generar escenas automáticamente, el juego existe infinitas posibles escenas, por lo tanto, existe infinitos niveles.

Modo aventura

También con como modo “adventure”, es parecido como el modo “escape” pero las salas se construyen de forma manual, de esa forma, podemos ofrecer a los usuarios unos niveles bien diseñados y dificultades controlables.

Elementos jugables:

Será igual como el modo escape, ya que por su definición es un modo igual como modo escape, pero con niveles predefinidos.

Nivel 1:

El jugador comienza desde la habitación número uno, en esta habitación vamos a encontrar dos puertas, una se encuentra bloqueada por el fuego y la otra no. Opcionalmente, el jugador puede pulsar alarma para recibir una bonificación.

La habitación dos también se encuentra con dos puertas, y ambas se pueden abrir. Pero fijamos que hay una señal de salida con flecha, si el jugador abre la puerta que no está apuntada por la señal, recibirá una penalización.

La tercera habitación es una zona con humo oscuro, si el jugador coje el pañuelo húmedo para cubrir vías respiratorias recibirá otra recompensa, y si quiere seguir avanzando, tiene que contestar una pregunta sobre cómo pasar el recinto con presencia de humo espeso.

En la cuarta habitación otra vez encontramos con dos puertas, una de ella está caliente y no se debe abrir en un incendio, si el jugador intentar abrir, le penalizamos.

La habitación quinta y sexta encontramos con puertas bloqueadas y personajes no controlables atrapadas en esta habitación, el jugador debe responder preguntas para poder salvar a todos ellos y abrir las puertas.

En la séptima habitación se encuentra con un extintor y una única puerta bloqueada por el fuego. Para poder continuar, el jugador debe contestar la pregunta de cómo usar el extintor para poder apagar el fuego con éxito.

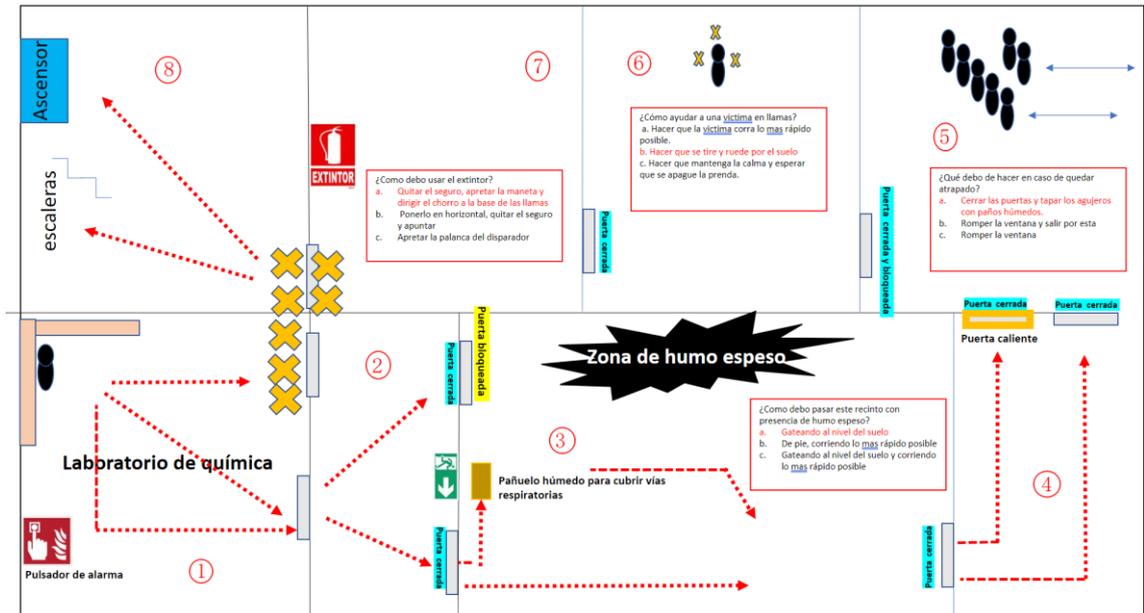
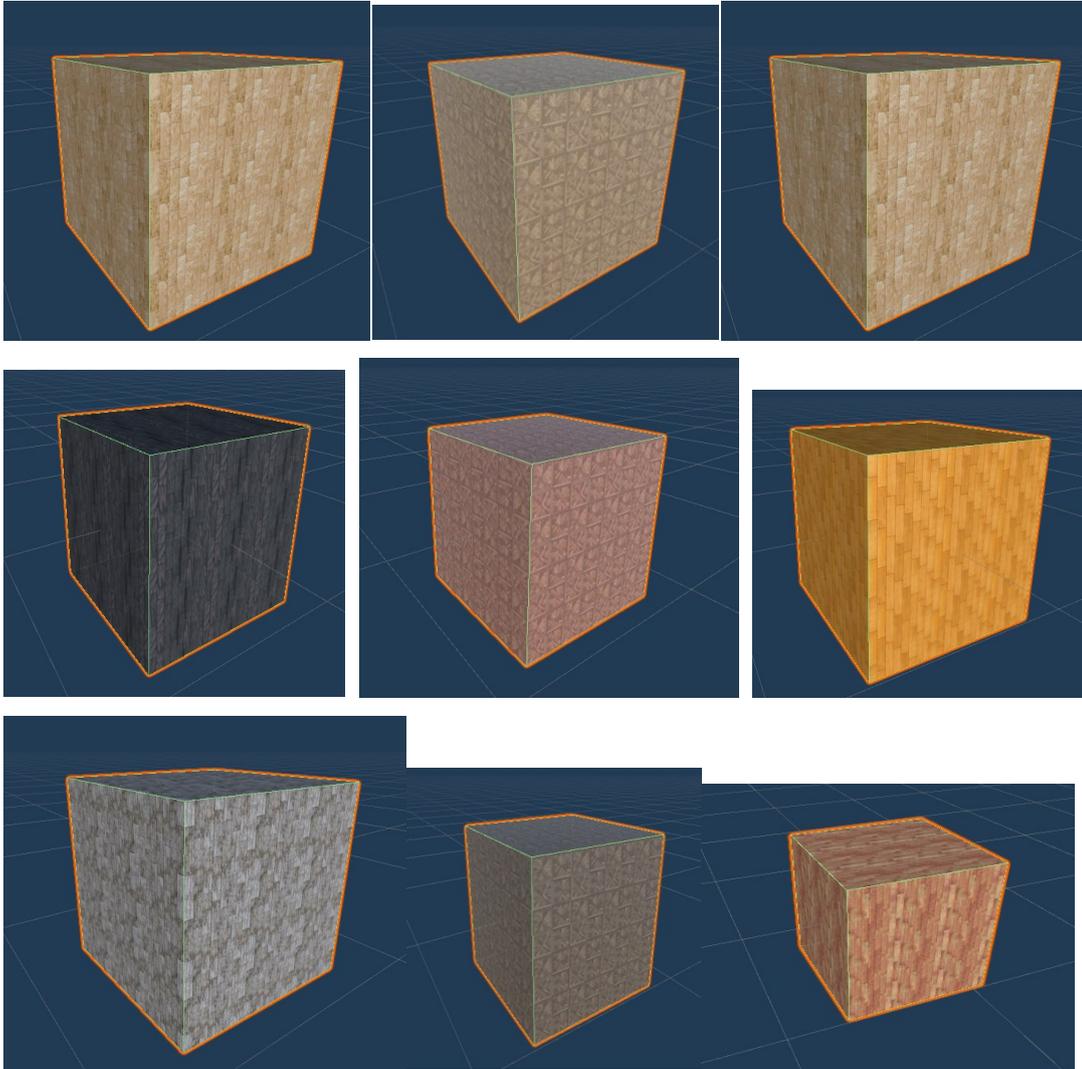


FIGURA GDD 15. Esquema conceptual del nivel 1.

SECCIÓN 6 Assets:

Modo construcción:

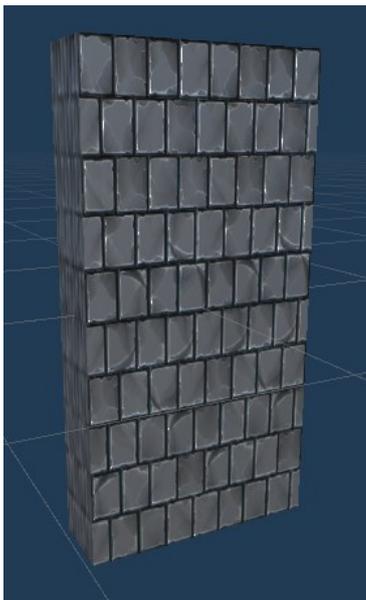
- **Bloques:** Sirve para colocar en el suelo de las habitaciones, la diferencia no sólo está en los colores, también influirá cómo propagar el fuego.



- **Puertas:** Sirve para parar el fuego de manera débil, debe tener al menos una para ser reconocido como una habitación.



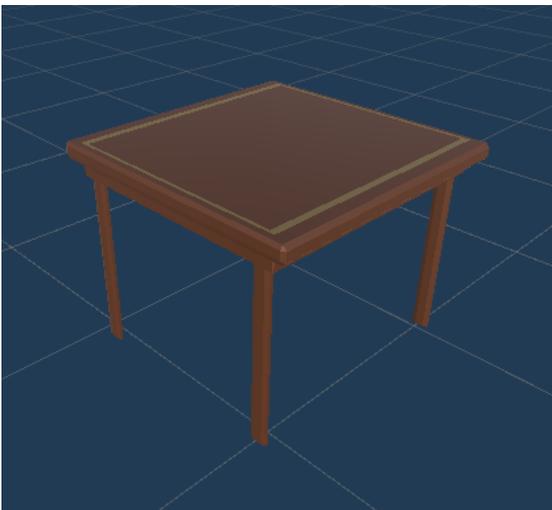
- **Muro:** Sirve para parar el fuego, debe rodear sobre bloques de suelo para ser reconocido como una habitación. También sirve para colgar los extintores.



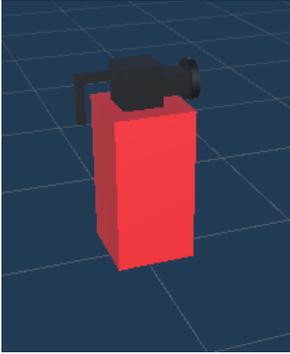
- **Ventana:** Para ventilar, a la hora de construir una habitación hay que tener cantidad necesaria para no saltarse de la normativa.



- **Mesa:** Elemento decorativo, puede dejar objetos sobre ella.



- **Extintor:** Para apagar el fuego.



Modo escape y modo aventura

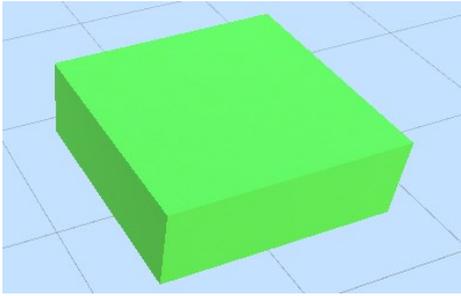
- **Protagonista:** Personaje controlable, hay que guiarle a la salida.



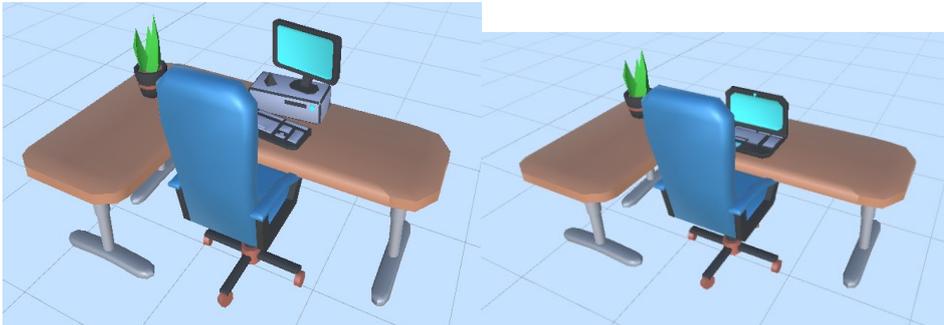
- **Personaje no controlable:** Personaje no controlable, puedes ayudarlo a escapar juntos.



- **Salida:** Tienes que llegar aquí para salir.



- **Mesas:** Decoraciones, interaccionable cuando es necesario.



- **Papelera:** Decoraciones, interaccionable cuando es necesario.



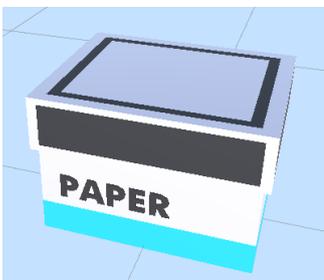
- **Armario:** Decoraciones, interaccionable cuando es necesario.



- **Carro de limpieza:** Decoraciones, interaccionable cuando es necesario.



- **Caja de papeles:** Decoraciones, interaccionable cuando es necesario.



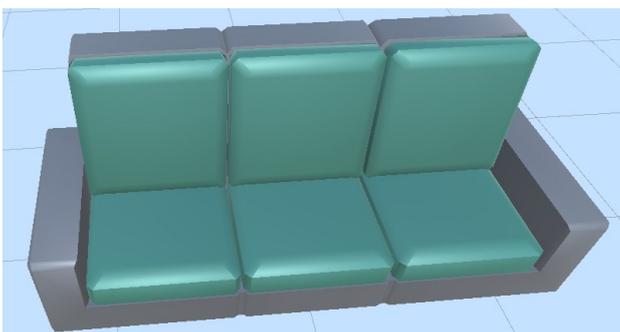
- **Planta:** Decoraciones, interaccionable cuando es necesario.



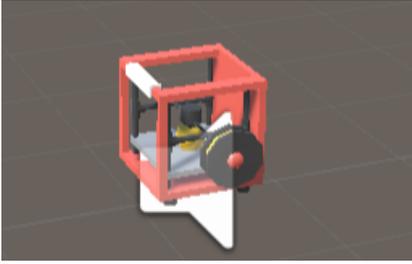
- **Máquina de agua:** Decoraciones, interaccionable cuando es necesario, como para mojar la toalla.



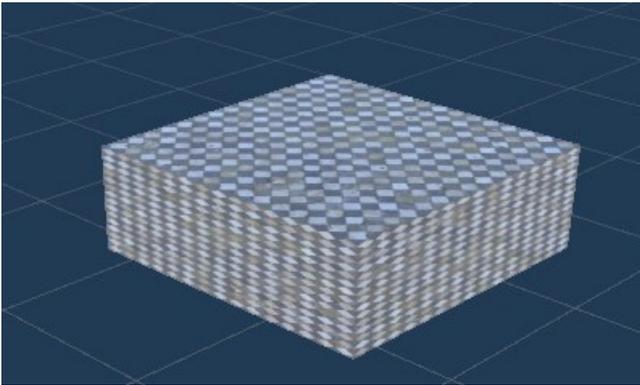
- **Sofá:** Decoraciones, interaccionable cuando es necesario.



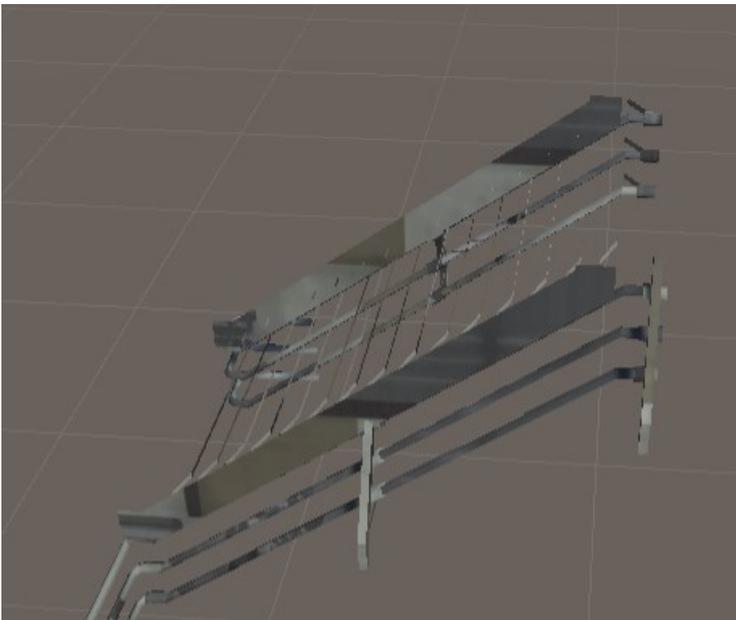
- **Alarma:** Objeto interaccionable, normalmente da bonus de tiempos extras.



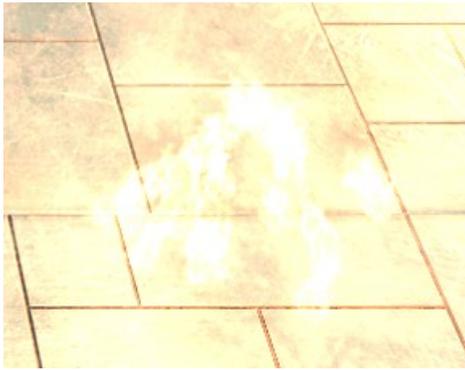
- **Toalla:** Objeto coleccionable, normalmente tienes que mojar en máquina de agua para poder reducir daño de fuego.



- **Escalera:** Hay que llegar a aquí para ir de una escena a otra.



- **Fuego:** Hace daño cuando estás sobre él y produce humos.



SECCIÓN 7: Controles

En este apartado se tratará de profundizar en detalle en todo lo referido a los controles que usaremos, para así poder realizar las distintas acciones que estarán definidas dentro del videojuego.

Tenemos la intención de lanzar nuestro videojuego en PC y dispositivos móviles, sin embargo, los controles de la versión de móvil no están implementados todavía.

Teclas por defecto para el modo de juego construcción (Build)

- Desplazamiento de la cámara: Control Izq + Clic Izquierdo.
- Rotación de la cámara: Control Izq + Clic Derecho.
- Zoom de la cámara: Rueda del ratón.
- Colocar un objeto: Clic Derecho.
- Eliminar un objeto: Clic Izquierdo.

Teclas por defecto para el modo de juego de estrategia (Escape) y aventura (Adventure)

- Desplazamiento de la cámara: teclas WASD.
- Seleccionar personaje: Clic izquierdo.
- Seleccionar destino: Clic izquierdo.
- Aumentar velocidad del movimiento: Shift.
- Abrir/cerrar mapa: tecla M.
- Interacción con entorno: Clic derecho.
- Contestar preguntas: Clic izquierdo.
- Cerrar el juego: Esc.

Menú de Acceso

- Seleccionar destino: Clic izquierdo.
- Interaccionar con el entorno: Clic derecho.
- Seleccionar una respuesta: Clic izquierdo.
- Cerrar el juego: Esc.

SECCIÓN 8: Interfaz

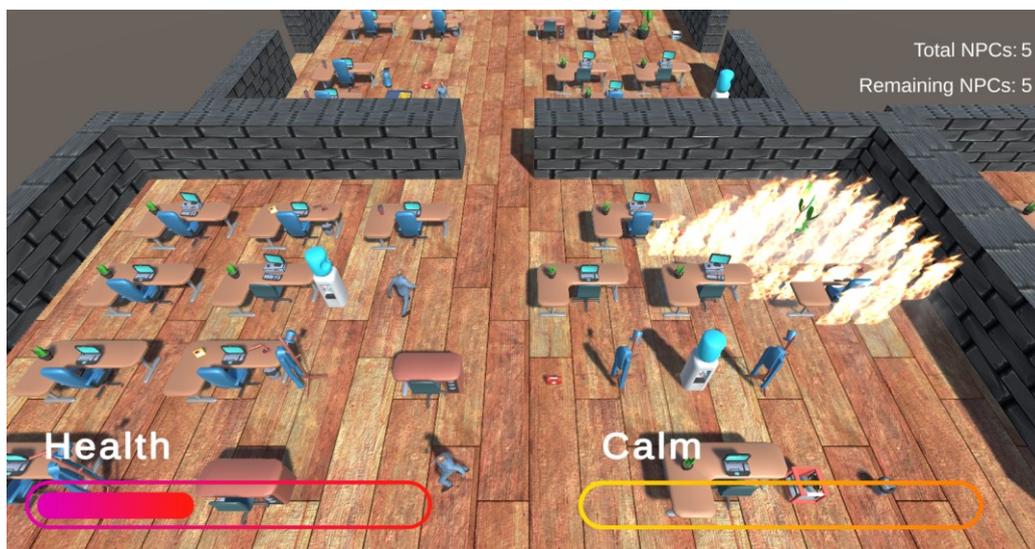
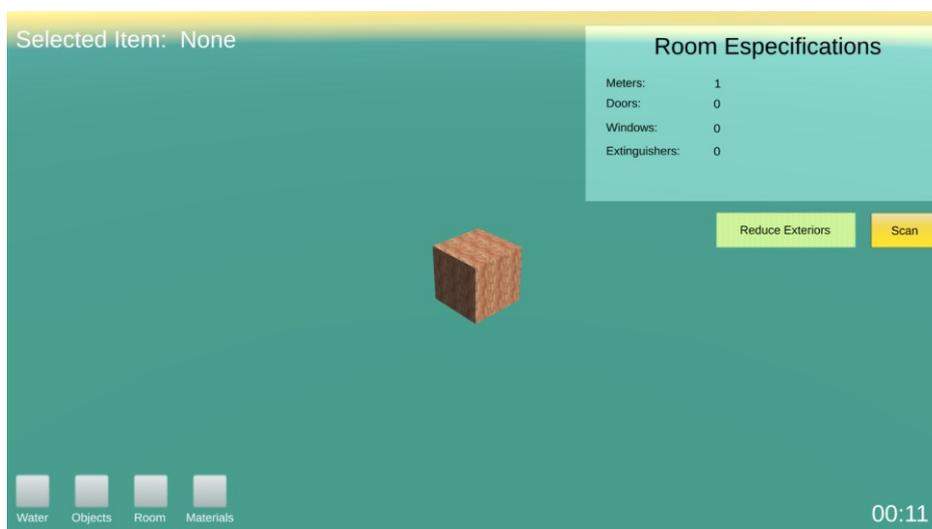
En esta sección concretamos la interfaz del jugador en detalle, especificaremos los distintos menús y cómo moverse entre ellos. Dentro de esta parte encontraremos el menú principal con sus distintas opciones, el HUD y el inventario.

Cámara

Respecto a la cámara tendremos una vista en 3ª persona controlable por el jugador, de manera que el la sitúe en la dirección y ángulo que quiera en cada momento.

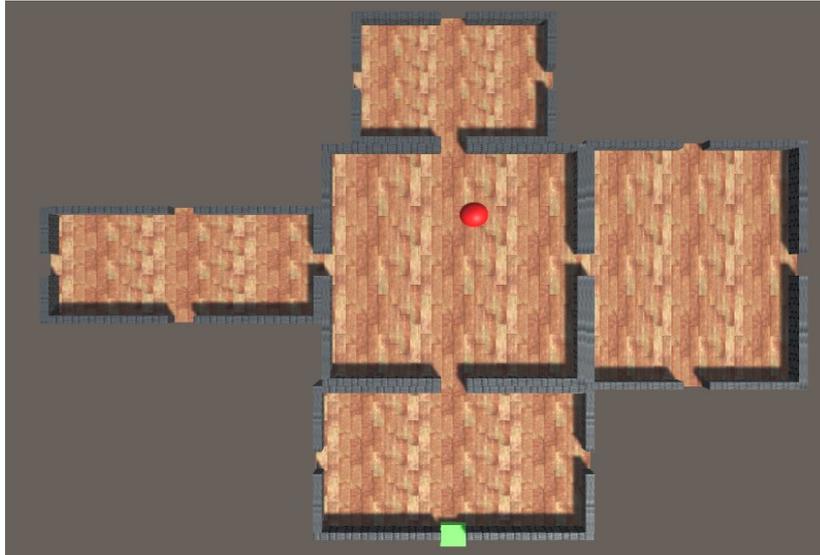
Vista estándar:

De manera estándar la cámara se situará en una posición concreta.



Vistas Alternativas

En el modo de estrategia (Escape) disponemos de una vista alternativa en forma de mapa, desde esta vista es la única donde pueden verse la posición actual del jugador en el escenario (punto rojo) y la salida del edificio donde tienen que llevar los personajes para ganar (cuadrado verde)



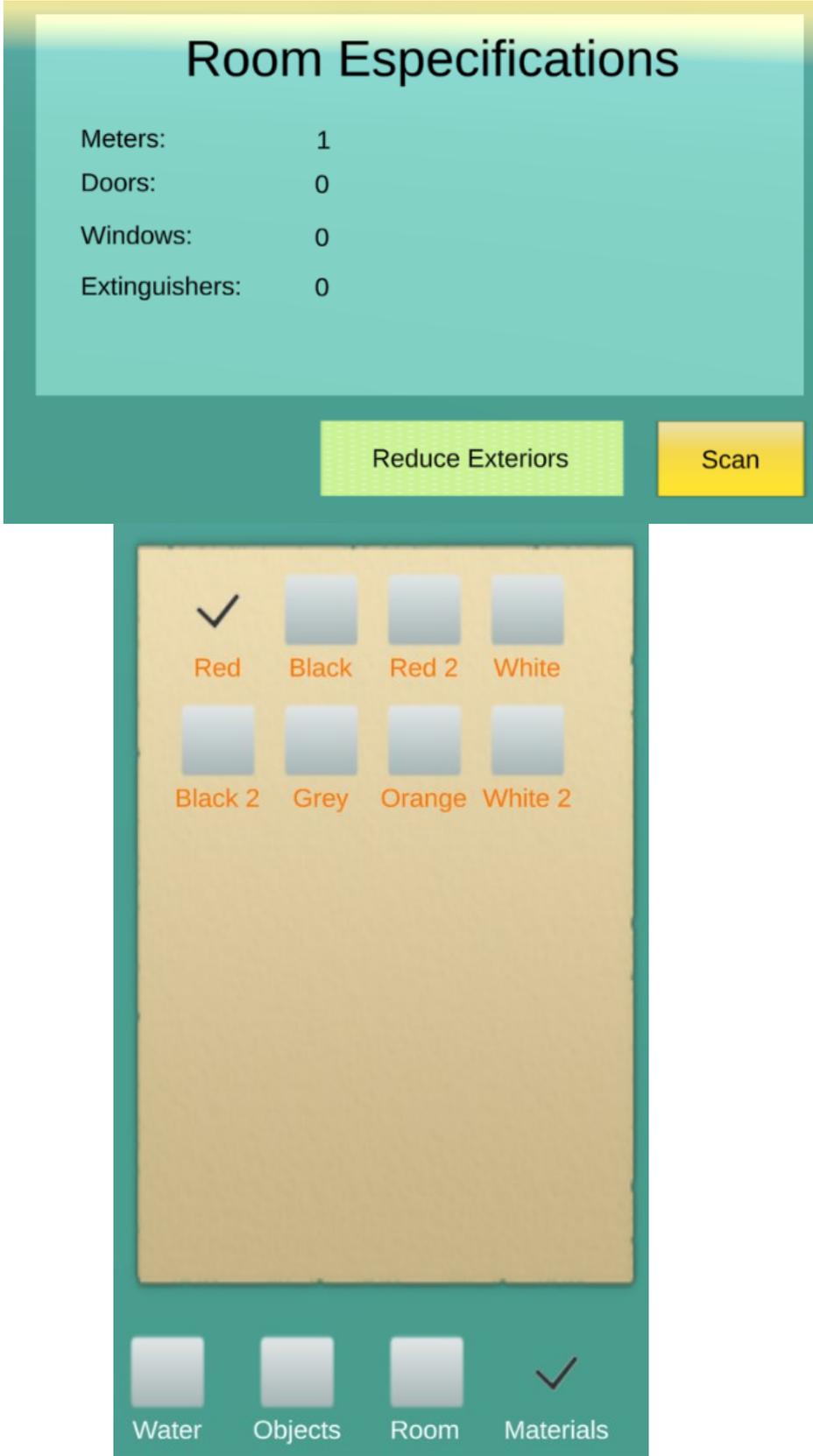
Opciones controlables por el Jugador:

Utilizando el ratón el jugador podrá mover la cámara a voluntad.

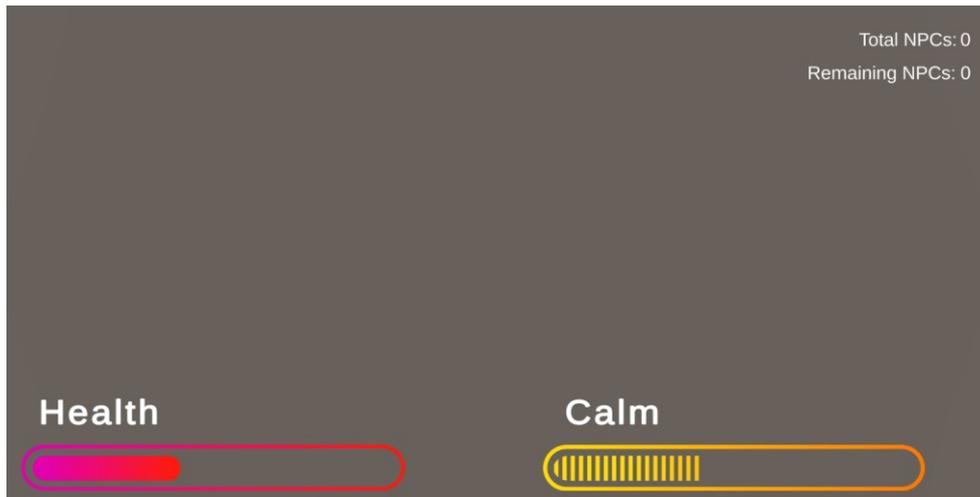
HUD

Modo de construcción:

Selected Item: CubeRedWood



Modo de estrategia



Modo de Aventura



Información de estado

En la vista del jugador podemos observar los distintos estados e información que le damos al jugador:

Modo escape y modo adventure

- Salud: Visible en la esquina inferior izquierda del HUD y administrada mediante una barra de color gradiente entre rosa y rojo. Cuando el fuego te alcanza al personaje el tamaño de la barra disminuye. Si te quedas sin salud pierdes el juego.
- Estado de ánimo: Visible en la esquina inferior izquierda del HUD y administrada mediante una barra de color gradiente entre amarillo y naranja. Cuando el personaje detecta fuego o humo este nivel aumenta. Cuando el personaje lleva un tiempo sin

ver elementos peligrosos comienza a disminuir gradualmente. Cuanto más alto es este nivel mayor será la velocidad del personaje y menor será el tiempo de espera del personaje antes de pasar al modo autónomo. También aparecerá un texto sobre la barra que se intercambiará entre: “Calm”, “Stress” y “Pánico” según vaya aumentando o disminuyendo el nivel de estrés.

- NPCs totales y restantes: Contadores visibles en la esquina superior derecha con el número de NPCs restantes y totales.

Modo Build

- Selected Item: En la esquina superior izquierda encontraremos este elemento que nos indicará qué objeto concreto tiene el jugador seleccionado en cada momento.

- Room Specifications: Este elemento se encuentra en la esquina superior derecha, se compone por un panel con la información relativa a todas las salas creadas por el jugador en la escena.

- Item selectors: Conjunto de botones donde los diferentes objetos se clasifican, al pulsar estos botones abrimos sus páginas correspondientes para poder seleccionar el objeto deseado.

Menús

Menú principal



Menús de opciones



SECCIÓN 9: Herramientas complementarias

Página web

Hay que crear una página web con siguientes finalidades:

- Tener fácil acceso a los distintos enlaces de interés sobre cómo actuar en medidas de emergencia y prevenirlas.
- Registrar los puntos de todas las partidas.
- Repasar las preguntas que aparecían en el juego.

Por lo tanto, que tiene siguiente forma:

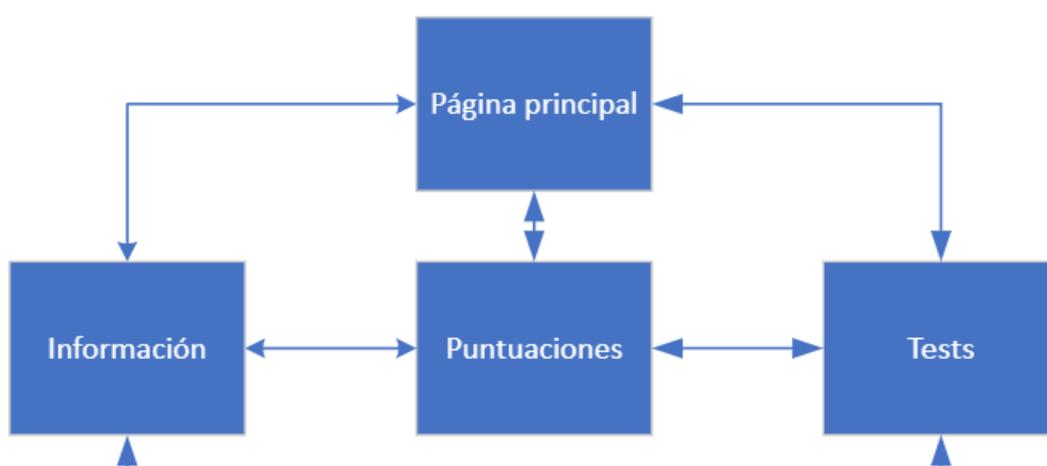


Figura GDD 16. Diagrama de flujo página web.

Y cada una de las funcionalidades corresponden a:

- Página principal: Portal para navegar a otras distintas páginas.
- Información: Acceder distintas fuentes de información sobre qué hacer en caso de incendio.
- Puntuaciones: Ver el historial de los resultados del juego.
- Tests: Consiste en hacer repastos de las preguntas del juego sin tener que abrirlo. También dispone un botón de “Mostrar respuestas” que sirve para ver las respuestas junto con una breve explicación con las referencias para que los jugadores puedan entender mejor porqué la respuesta es así.

Anexo III: Objetivo de desarrollo sostenible

OBJETIVOS DE DESARROLLO SOSTENIBLE

Grado de relación del trabajo con los Objetivos de Desarrollo Sostenible (ODS).

Objetivos de Desarrollo Sostenibles	Alto	Medio	Bajo	No Procede
ODS 1. Fin de la pobreza.				X
ODS 2. Hambre cero.				X
ODS 3. Salud y bienestar.	X			
ODS 4. Educación de calidad.	X			
ODS 5. Igualdad de género.				X
ODS 6. Agua limpia y saneamiento.				X
ODS 7. Energía asequible y no contaminante.				X
ODS 8. Trabajo decente y crecimiento económico.				X
ODS 9. Industria, innovación e infraestructuras.				X
ODS 10. Reducción de las desigualdades.				X
ODS 11. Ciudades y comunidades sostenibles.	X			
ODS 12. Producción y consumo responsables.				X
ODS 13. Acción por el clima.				X
ODS 14. Vida submarina.				X
ODS 15. Vida de ecosistemas terrestres.			X	
ODS 16. Paz, justicia e instituciones sólidas.	X			
ODS 17. Alianzas para lograr objetivos.		X		

Reflexiones:

El incendio siempre ha sido un tema serio y difícil de tratar. Un siniestro de incendio mal gestionado puede provocar daños muy graves hasta muertes, además su explosión y humo puede dañar e intoxicar el ambiente. Sin duda, si dicho proyecto puede lograr éxito (que el público sabe cómo evacuar en caso de emergencia o incluso evitar la aparición de incendio), puede alcanzar muchos de los objetivos de ODS.

Para empezar, mi trabajo de final de grado está sólidamente relacionado con los cuatro objetivos de desarrollo sostenible:

Salud bienestar: El juego DANGER ayuda a reducir número de incendios y enseñan a la gente cómo autoprotgerse en el caso de emergencia, en consecuencia, fortalece las capacidades de alerta temprana personal, reduce riesgo y gestiona riesgo de incendio a nivel internacional, por lo tanto, bajarán la tasa de mortalidad,

Educación de calidad: Como mencionaba en la memoria, juego y la educación siempre está vinculado. Además, el videojuego es más accesible para cualquier público y da una formación equitativa a cualquier jugador de Danger.

Ciudades y comunidades sostenibles: El juego reducirá las muertes por desastres y número de personas afectadas. Si los jugadores aprenden técnicas de evitar incendios, incluso es posible reducir impacto nocivo y mejorar la calidad de aire de las ciudades debido por reducción de humos. También reduce gastos públicos para la reconstrucción tras siniestro.

Paz, justicia e instituciones sólidas: La reducción de número de incendios puede hacer que la sociedad sea más tranquila y pacífica.

También está algo relacionado con:

Alianzas para lograr objetivos, ya que uno de los objetivos dentro del juego es ayudar a otros personajes a escapar juntos. Me gustaría que los jugadores puedan llevar esta virtud a fuera del juego.

Vida ecosistemas terrestres: Los bosques estarán mejor protegidos ya que evita los incendios provocados por acción humana en los bosques, esto favorece tanto a los bosques como los animales y las plantas que viven en ahí.