javier granizo

DoomTD

Contenido

[1.Análisis del problema 3](#_Toc184757090)

[1.1. Introducción 3](#_Toc184757091)

[1.2. Objetivos 3](#_Toc184757092)

[1.3. Funciones y rendimientos deseados 3](#_Toc184757093)

[1.4. Planteamiento y evaluación de diversas soluciones 4](#_Toc184757094)

[1.5. Justificación de la solución elegida. 4](#_Toc184757095)

[1.6. Modelado de la solución 5](#_Toc184757096)

[1.6.1. Recursos humanos 5](#_Toc184757097)

[1.6.2. Recursos hardware 5](#_Toc184757098)

[1.6.3. Recursos software 5](#_Toc184757099)

[1.7. Planificación temporal 6](#_Toc184757100)

[2. Diseño e implementación del proyecto 6](#_Toc184757101)

[2.1. Arquitectura de la Aplicación 7](#_Toc184757102)

[2.1.1. Interfaz de Usuario (UI): 7](#_Toc184757103)

[2.1.2. Gestión de Estado: 7](#_Toc184757104)

[2.1.3. Servidor del lado del Usuario 7](#_Toc184757105)

[2.1.3.1 . Controladores: 7](#_Toc184757106)

[2.1.3.2. Lógica de Negocio: 7](#_Toc184757107)

[2.1.3.3. Base de Datos: 7](#_Toc184757108)

[2.1.3.4. Autenticación y Autorización: 8](#_Toc184757109)

[2.2. Tipos de Usuarios y Operaciones 8](#_Toc184757110)

[2.2.1. Usuario Registrado 8](#_Toc184757111)

[2.3. Mapa de Navegación de la APP 8](#_Toc184757112)

[Menú de Log in 8](#_Toc184757113)

[Menú inicial 9](#_Toc184757114)

[Menú Ajustes 12](#_Toc184757115)

[Menú de Créditos 13](#_Toc184757116)

[Menú de Niveles 14](#_Toc184757117)

[Niveles 14](#_Toc184757118)

[3. Fase de pruebas 17](#_Toc184757119)

[3.1. Pruebas de la Interfaz de Usuario 17](#_Toc184757120)

[3.2. Pruebas de las Mecánicas del Juego 18](#_Toc184757121)

[3.3. Pruebas de Rendimiento 20](#_Toc184757122)

[3.4. Pruebas de Progresión y Balance 20](#_Toc184757123)

[3.5. Pruebas de Integración Usuario-Sistema 21](#_Toc184757124)

[3.6. Pruebas sobre la Resiliencia y Errores 21](#_Toc184757125)

[3.7. Pruebas Estéticas y de la Experiencia del Usuario 22](#_Toc184757126)

[4. Documentación de la aplicación. 23](#_Toc184757127)

[4.1. Introducción a la aplicación (Getting Started) 23](#_Toc184757128)

[4.2. Manual de Instalación 24](#_Toc184757129)

[4.3. Manual de usuario. 24](#_Toc184757130)

[1. Introducción al Juego 24](#_Toc184757131)

[2. Interfaz Principal 25](#_Toc184757132)

[3. Cómo Empezar a Jugar 25](#_Toc184757133)

[4. Mecánicas Principales 26](#_Toc184757134)

[5. Consejos Básicos para Progresar en el Juego 27](#_Toc184757135)

[6. Resolución de Problemas Comunes 27](#_Toc184757136)

[5. Conclusiones finales 28](#_Toc184757137)

[6. Bibliografía 28](#_Toc184757138)

# 1.Análisis del problema

## Introducción

Este proyecto se centra en el desarrollo de un juego de estilo Tower Defense, un género popular donde los jugadores deben construir torres defensivas para proteger su base de oleadas de enemigos. El objetivo del proyecto es crear un juego entretenido, desafiante y que ofrezca una experiencia de usuario única.

## Objetivos

Desarrollar un juego de Tower defense Completo: Crear un juego funcional con múltiples niveles y dificultades, diversas torres y enemigos.

Proporcionar una experiencia de usuario atractiva: Asegurar que el juego sea visualmente atractivo y fácil de jugar.

Implementar mecánicas de juego balanceadas: Garantizar que las torres y los enemigos estén bien equilibrados para mantener un nivel de desafío adecuado.

Optimización y rendimiento: Asegurar que el juego funcione sin problemas en una variedad de dispositivos.

## Funciones y rendimientos deseados

Múltiples niveles y dificultades: Diferentes niveles que aumentan en dificultad para mantener el interés del jugador.

Diversidad de torres y enemigos: Varias torres con diferentes habilidades y enemigos con distintas características.

Interfaz de usuario intuitiva: Controles y menús fáciles de usar.

Gráficos y efectos de sonido atractivos: Diseño visual y audio que mejoren la experiencia del jugador.

Puntuaciones y logros: Sistema de puntuación y logros para aumentar la rejugabilidad.

## Planteamiento y evaluación de diversas soluciones

Motor de juego: Evaluación de Unity y Unreal Engine. Unity es elegido por su facilidad de uso, amplia comunidad y soporte para 2D.

Diseño de niveles: Uso de herramientas como Tiled o el editor de niveles interno de Unity.

Optimización de rendimiento: Uso de técnicas como la carga diferida y la optimización de sprites.

Implementación de IA para enemigos: Uso de scripts para crear patrones de movimiento y comportamiento de los enemigos.

## Justificación de la solución elegida.

Se eligió Unity como motor de juego debido a su flexibilidad y facilidad de uso para desarrolladores independientes. Unity permite una rápida iteración en el diseño y desarrollo del juego, y tiene una amplia gama de recursos y tutoriales disponibles.

## Modelado de la solución

### Recursos humanos

Desarrollador principal: Encargado del diseño y la implementación del juego.

Javier Granizo Aguilar

Artista gráfico: Responsable de los gráficos y animaciones del juego.

Javier Granizo Aguilar

Diseñador de sonido: Encargado de crear y seleccionar los efectos de sonido y la música del juego.

Javier Granizo Aguilar

Tester: Encargado de probar el juego para detectar errores y asegurarse de que todo funcione correctamente.

Javier Granizo Aguilar

### Recursos hardware

Ordenadores de desarrollo: Pc personal bajos recursos.

Dispositivos de prueba: Pc personal bajos recursos.

### 1.6.3. Recursos software

Unity: Motor de juego para el desarrollo.

Visual Studio: IDE para programar en C#.

SQLitelStudio: Visualizador de la base de datos.

Piskel: Creador de esprites

## Planificación temporal

**Fase 1: Planificación (1 semanas)**

* Definición de objetivos y funcionalidades.
* Investigación y elección de herramientas.

**Fase2: Desarrollo (6 semanas)**

* Implementación del motor de juego y mecánicas básicas.
* Desarrollo de niveles y programación de IA de enemigos.
* Integración de gráficos y efectos de sonido.

**Fase 3: Diseño (1 semanas)**

* Diseño de niveles y mecánicas de juego.
* Creación de arte conceptual y diseño de personajes.

**Fase 4: Pruebas (1 semanas)**

* Pruebas de usabilidad y rendimiento.
* Corrección de errores y optimización.

**Fase 5: Revisión final y entrega (1 semana)**

* Últimos ajustes y mejoras.
* Preparación de la documentación y presentación del proyecto.

# Diseño e implementación del proyecto

## 2.1. Arquitectura de la Aplicación

### 2.1.1. Interfaz de Usuario (UI):

La interfaz gráfica por la que los usuarios interactúan está creada completamente con Unity utilizando Canvas. Esto permite una experiencia de usuario rica y visualmente atractiva.

### 2.1.2. Gestión de Estado:

La gestión del estado de la aplicación en el cliente se maneja utilizando scripts en C# dentro de Unity. No se utilizan librerías externas, sino que se implementa la lógica de estado directamente en Unity.

## 2.1.3. Servidor del lado del Usuario

### 2.1.3.1 . Controladores:

Los controladores se encargan de la gestión de las peticiones que llegan al sistema, ejecutando la lógica de negocio y devolviendo la respuesta. Esto puede hacerse con el uso de scripts en Unity que hagan de controladores gestionando eventos y entradas.

### 2.1.3.2. Lógica de Negocio:

La lógica de negocio está completamente implementada en C# dentro de Unity. Los scripts implementan las reglas y los procesos que son necesarios para la funcionalidad de esta aplicación.

### 2.1.3.3. Base de Datos:

La aplicación hace uso de SQLite para el almacenamiento local de datos. SQLite es una base de datos ligera y fácil de integrar con Unity, esto es llevado a cabo mediante el uso de plugins como es en mi caso NuGet el cual permite una integración de la base de datos.

### 2.1.3.4. Autenticación y Autorización:

La gestión de la seguridad del sistema comprueba la identidad de los usuarios y sus permisos, que pueden gestionar validando credenciales almacenadas en la base de datos SQLite y añadiendo mecanismos de autorización en Unity.

## 2.2. Tipos de Usuarios y Operaciones

### 2.2.1. Usuario Registrado

Operaciones:

Crear su perfil.

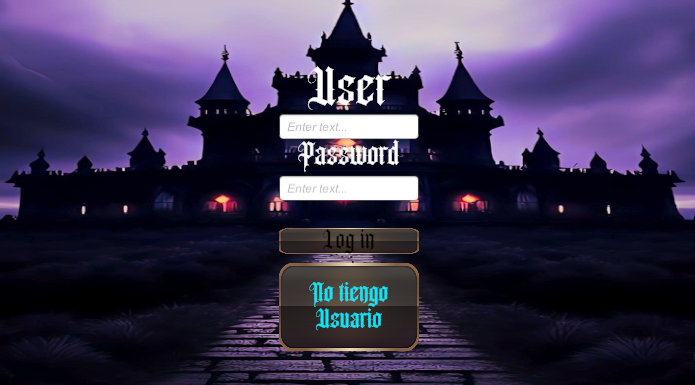
Editar su perfil.

Borrar su perfil.

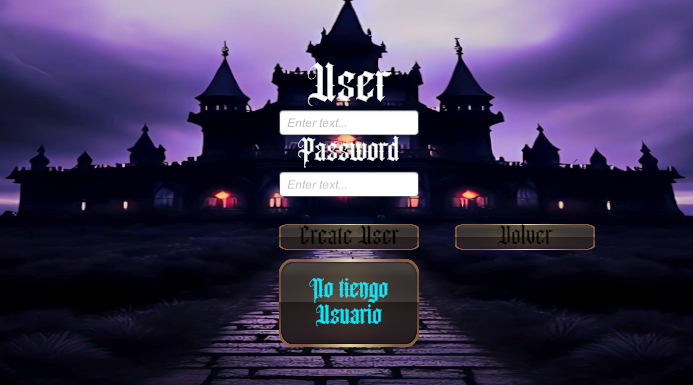
## 2.3. Mapa de Navegación de la APP

### Menú de Log in

Hacer log in con un usuario.



Crear Usuario



### Menú inicial

#### Funcionalidades



Botón para ir a los ajustes de usuario.

Botón para ir a la escena de Niveles.

Botón para ir a la escena de Ajustes.

Botón para escena de Créditos.

Botón para salir del juego.

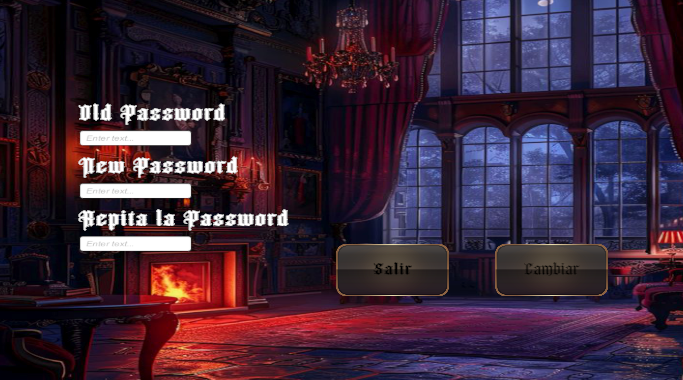
#### Ajustes de usuario



Edición del nombre del usuario.



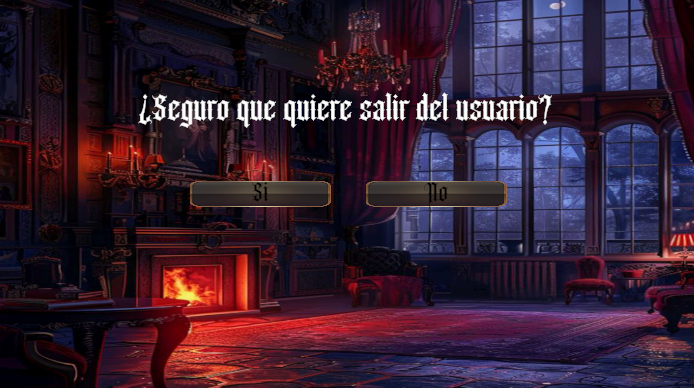
Edición de la Contraseña del usuario.



Cambiar de usuario te pregunta si quieres salir o no .

Si dices que si te manda al menú de usuario.

Si No te manda a el menú de inicio.



Volver a las funcionalidades.

### Menú Ajustes

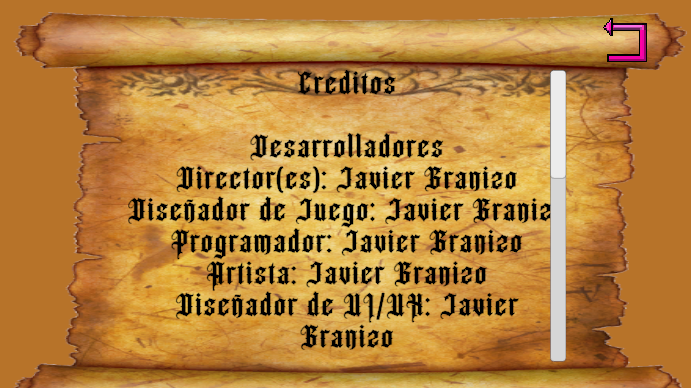


Seleccionador de Resolución.

Slider para indicar el volumen General.

Botón para volver al Menú Inicial

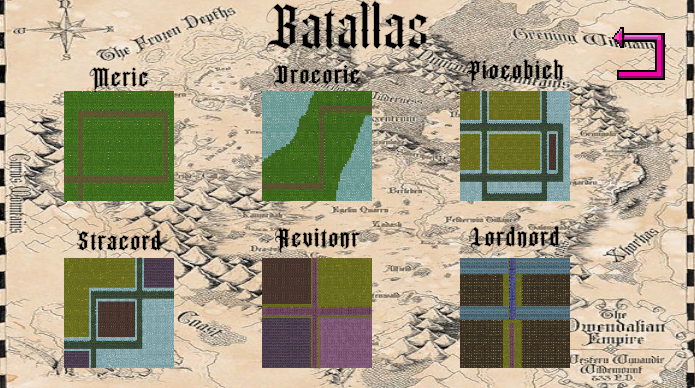
### Menú de Créditos



En este aparecen los créditos del juego

Botón para volver al Menú Inicial

### Menú de Niveles



Botones enlazados a sus respectivos niveles

Botón que devuelve al menú inicial

### Niveles

Todos los niveles siguen la misma estructura de movimiento entre distintas funcionalidades

#### Menú elección



Seis botones cada uno con su personaje asociado

Botón de pausa lleva al menú de Pausa

#### Menú Pausa



Botón para salir del nivel.

Botón para continuar.

Botón Para menú de ajustes en nivel.

#### Menú Ajustes



Seleccionador de Resolución.

Slider para indicar el volumen General.

Botón para salir de este menú

#### Menú de Mejoras



En este puedes seleccionar Las mejoras deseadas teniendo en cuenta que solo puedes tener con 2 mejoras o mas un camino que seria o el primer botón o el segundo

Puedes seleccionar a que tipo de globo ataca el personaje entre: Primero, Ultimo, Débil y Fuerte

Con el icono de la bolsa del dinero vende el personaje

Salir del Menú de Mejoras

# Fase de pruebas

## Pruebas de la Interfaz de Usuario

Objetivo: Comprobar que la interfaz se comporte de forma adecuada y que sea lo más intuitiva posible.

Prueba 3.1.1: Comprobar que los botones del menú principal (Jugar, Configuraciones, Salir) respondan cuando se hagan clics o toques en ellos.

Funciona de manera correcta

Prueba 3.1.2: Confirmar que la sección del mapa sea visible y que las torres se puedan arrastrar o colocar sin solaparse entre sí y con el resto de los elementos que conforman el mapa.

Funciona de manera correcta

Prueba 3.1.3: Verificar que la barra de recursos cambie en tiempo real cuando se ganan o se gastan recursos.

Funciona de manera correcta

Prueba 3.1.4: Probar el juego en distintos tamaños de pantalla (dispositivos móviles, tabletas, PC).

Hay algunas resoluciones las cuales se pierde el diseño original.

Prueba 3.1.5: Comprobar que el botón de "Pausa" detenga el juego correctamente y que ofrezca la opción de reanudar, reiniciar o salir de partida.

Funciona de manera correcta

## 3.2. Pruebas de las Mecánicas del Juego

Objetivo: Comprobar que las reglas y las dinámicas se comporten como se espera.

3.2.1: Colocar una torre en una posición válida y comprobar que ataque físicamente a los enemigos que se encuentren dentro del rango de alcance de la torre.

Funciona de manera correcta

3.2.2: Colocar una torre en una posición que no sea válida (fuera de la zona de juego) y comprobar que no se pueda realizar la acción.

Funciona de manera correcta

3.2.3: Comprobar que los enemigos continúen por la ruta establecida sin desviarse de ella.

Funciona de manera correcta

3.2.4: Comprobar que el jugador pierda puntos de vida por cada enemigo que llegue al final de la ruta.

Funciona de manera correcta

3.2.5: Confirmar que las mejoras de torre (mayor daño, mayor rango de ataque, mayor velocidad de disparo, etc..) se comporten de la forma esperada tras su obtención.

Funciona de manera correcta

3.2.6: Comprobar el comportamiento de los enemigos en distintas velocidades de avance y con sus atributos especiales (resistencia, invisibilidad y regeneracion).

Funciona de manera correcta

## 3.3. Pruebas de Rendimiento

Finalidad: Evaluar el rendimiento del juego en distintas circunstancias.

3.3.1: Ejecutar el juego con numerosos enemigos en pantalla para comprobar que su rendimiento permanece estable (sin caídas notables de FPS).

Cuando colocas demasiadas torres al final de la ronda da un tirón por el tiempo que tarda en guardar en la base de datos.

## 3.4. Pruebas de Progresión y Balance

Finalidad: Comprobar la existencia de una experiencia justa o equilibrada para todos los jugadores.

3.4.1: Comprobar que los niveles más avanzados sean progresivamente más complicados, pero alcanzables.

Después de la beta testing y unos cuantos arreglos a parámetros se pueden jugar todos los niveles y llegar a pasarse, eso no implica que sea fácil.

3.4.2: Comprobar que los recursos obtenidos (dinero, experiencia) son suficientes para comprar torres o mejoras acordes al nivel.

3.4.3: Comprobar que las torres iniciales son útiles, sin ser más fuertes que las avanzadas.

Se llega a la conclusión que cada una tiene su funcionalidad, y si sabes posicionarlas son útiles.

3.4.4: Revisar que los enemigos "jefes" sean desafiantes, pero derrotables.

Si se consigue llegar a los Boses estos deberían de resultar faciales en comparación con las rondas anteriores.

3.4.5: Comprobar que se puede reiniciar un nivel en cualquier momento sin errores.

Por las pruebas que se han hecho no deberían de pasar errores, lo que no quita la posibilidad de un error, de alguna manera la cual no se allá probado.

## 3.5. Pruebas de Integración Usuario-Sistema

Finalidad: Comprobar la integración de los componentes del sistema, así como las interacciones con los usuarios.

3.5.1: Crear una cuenta de usuario y comprobar que el progreso del jugador se guarda como es debido.

Se crea y guarda correctamente.

## 3.6. Pruebas sobre la Resiliencia y Errores

Objetivo: Que el juego mantenga el funcionamiento esperado en situaciones inesperadas.

3.6.1: Cerrar el juego en una partida y volver a entrar para comprobar que se reanude con normalidad.

Como esta creado guarda cada vez que se termine una ronda por lo que te devuelve al inicio de la ronda.

3.6.2: Realizar acciones inválidas (por ejemplo, colocar 7 torres en una unidad), y estudiar que el sistema lo pueda asumir.

El juego esta preparado para estos estilos de casos como:

Colocar más torres de las permitidas en una unidad de tiempo.

Colocar torres fuera del área permitida.

Colocar torres superpuestas.

Colocar torres sin suficientes recursos.

Intentar mejorar una torre al nivel máximo por ambos caminos o paths.

Gastar recursos después de perder la partida.

3.6.3: Verificar que el juego sea capaz de volver a la normalidad luego de haber sido cerrado.

Se reanuda correctamente.

## 3.7. Pruebas Estéticas y de la Experiencia del Usuario

Objetivo: Testar la experiencia estética y la experiencia de juego en general.

3.7.1: Revisar que las animaciones -si las hubo- se dieran de forma fluida y fueran satisfactoria de ver.

Las animaciones en general funcionan bien pero algunas no son muy buenas o poco fluidas

3.7.2: Recabar que los colores y fuentes de texto se dieran de forma legible, y no fuese cansado de ver.

Los colores por la información recabada por diferentes sujetos parecen ser acertada.

3.7.3: Asegurar que los sonidos y la música no llegasen a ser repetitivos o irritantes.

Los sonidos de las torre pueden llegar a irita a algunas personas

4. Documentación de la aplicación.

Importante: esta documentación se debe

incluir en la distribución del proyecto. Dependiendo de la naturaleza del

proyecto serán necesarios todos o alguno de los manuales mencionados y el

modo de inclusión será el adecuado al tipo de proyecto.

## 4.1. Introducción a la aplicación (Getting Started)

El Juego al que va a disponerse a jugar trata sobre el rey demonio intentando conquistar las tierras humanas, tu juegas como el rey demonio el cual comandara sus tropas a la batalla.

Cada vez que mates a un humano venderás su carne para conseguir dinero y llevar a tus tropas al campo de batalla o mejorar las que estén hay. Este juego se trata de un Tower Defence en el que los enemigos tendrán un camino predefinido y tu como jugador deveras matarlos antes de que alcance el final del camino momento en el que te descontara vida dependiendo de su dificultad, entre estos hay los camuflados los cuales tendrán un aura verdosa, los regenerables los cuales son rojas sus armaduras y los acorazados los cuales aguantan mucho daño si o es del tipo adecuado, junto con sus fusiones y los temidos boses los cuales serán un desafío que indica las rondas claves como la ronda 40, la ronda 80, la ronda 90 y la ronda 100.

Juego para todos los públicos, pero dirigido a un público joven.

## 4.2. Manual de Instalación

Dirijirse al repositorio de github:

<https://github.com/javiergranizoaguilar/Dungeon-td/tree/main>

Descargar el zip llamado: DundeonTDGame.zip

Extraerlo e iniciar Doom TD.exe

## 4.3. Manual de usuario.

### 1. Introducción al Juego

Te damos la bienvenida a Doom TD, un emocionante y desafiador juego del tipo Tower Defense. En este manual podrás encontrar instrucciones precisas sobre los pasos y, así pues, podrás comenzar a jugar, comprender las mecánicas más importantes del juego y mejorar tu propio conocimiento sobre cómo conseguir mejor tus habilidades estratégicas y defender tu base de las diferentes oleadas de enemigos.

### 2. Interfaz Principal

Cuando abras el juego te encontrarás con las siguientes secciones:

Menú de usuario:

Log in: coloca tu usuario y contraseña.

Crear Usuario: Crea un usuario.

Menú Principal:

Jugar: Accede al mapa de niveles para poder ya iniciar una partida, entre otras cosas.

Configuración: Ajusta el volumen, los gráficos.

Perfil: Puedes editar el nombre y contraseña de tu usuario.

Menú Niveles:

Visión del juego:

Mapa del Nivel: Este es el mapa que muestra el recorrido de los enemigos y las zonas en donde puedes situar la torre, por lo tanto.

Barra de Recursos: Indica la cantidad actual que tienes de Dinero, Rondas y Vidas.

Barra de Torretas: Contiene las torres que te encuentras disponible para colocar, entre otras cosas.

### 3. Cómo Empezar a Jugar

Al iniciar el juego por primera vez, crea un usuario.

Entra en tu susario.

Dele a jugar.

Pulsa Jugar en el menú principal para abrir el mapa de niveles.

Los niveles se desbloquean progresivamente a medida que avanzas.

### Mecánicas Principales

4.1 Colocar Torres:

Elige una torre en la barra derecha

Arrastrar y soltar la torre en una de las zonas válidas del mapa.

Asegúrate de que los enemigos estén en el rango de ataque de la torre para su función.

4.2 Mejorar la Torre, Vender la Torre

Mejorar:

Presiona una torre que ya esté colocada.

Pulsa el botón Mejorar -> Aumenta el daño, el alcance o la velocidad.

Cada mejora tiene el precio en monedas.

Solo se puede seguir uno de los dos path hasta segunda o tercera mejora y el otro solo al primero.

Vender:

Mientras sujetas una torre clic derecha.

Presiona una torre que ya no necesites.

Pulsa el botón Vender -> Para ti, el precio de volver a tu parte de recursos gastados.

4.3 Gestionar los Recursos:

Ganar monedas al vencer a los enemigos, o a través de completar las oleadas.

Las monedas se gastan en colocar torres o en mejorarlas, pero no gastes toda tu utilidad de una vez, algunos enemigos requieren estrategias muy avanzadas.

### 5. Consejos Básicos para Progresar en el Juego

Se ha de planificar con anterioridad:

Hay que observar el camino de los rivales y colocar torres en lugares estratégicos.

Hay que diversificar tus torres:

Se debe utilizar torres diferentes (daño rápido, ataque fuerte, lentificación, veneno) para adaptarse a los enemigos.

Hay que gestionar tus recursos:

No gastes todas tus monedas de una vez, pero ahorrarlas para oleadas más duras.

Hay que mejorar prudentemente:

Hay que mejorar primero las torres bien situadas antes de añadir otras torres nuevas.

Hay que conocer a los enemigos:

Hay enemigos inmunes a ciertos efectos; revisa las habilidades de los enemigos para decidir tu estrategia.

### 6. Resolución de Problemas Comunes

No puedes colocar una torre: Asegúrate de que esté en una zona válida (marcada en verde) y de tener suficientes recursos.

Perdiste la partida rápidamente: Prueba diferentes combinaciones de torres y analiza el recorrido de los enemigos.

# 5. Conclusiones finales

El proyecto titulado DoomTD se ha materializado con éxito, ya que finalmente se ha elaborado un juego funcional y entretenido en el género Tower Defense, que abarca las mecánicas básicas para ello, como la colocación de torres, la gestión de recursos y avanzar a través de niveles de dificultad creciente. El diseño de la interfaz se ha hecho suficientemente intuitivo y atractivo, y ha contribuido así a que el jugador adquiera una experiencia de usuario positiva.

En consonancia con las diferentes fases del desarrollo, han surgido áreas de mejora en las que se podrían recuperar, de manera óptima, el modo en el que se efectúan las animaciones o los efectos sonoros que acompañen al juego y, así, dar un paso más en la experiencia inmersiva del jugador. También se pueden contemplar extensiones, como la de nuevos tipos de torres y nuevos tipos de enemigos, niveles extra, para dar un tope a la rejugabilidad y aumentar el alcance del mismo.

Finalmente, DoomTD se establece como una base suficiente para seguir avanzando en el género de los Tower Defense, equilibrado entre diversión y el desafío estratégico.

# 6. Bibliografía

Documentación del motor Unity:

Unity Technologies. Unity User Manual. Disponible en: <https://unity.com>

SQLite para almacenamiento local:

SQLite. Official Documentation. Disponible en: <https://sqlite.org>

Tutoriales de diseño de juegos en Unity:

Brackeys. How to make a Tower Defense Game in Unity. Disponible en: <https://youtube.com/brackeys>