

3DSLug



19 de junio de 2022

JORGE ARENAS SORIANO

Contenido

[¿En qué consiste? 2](#_Toc99907432)

[Opciones de dificultad. 2](#_Toc99907433)

[Análisis de requisitos. 2](#_Toc99907434)

[Requisitos funcionales 2](#_Toc99907435)

[Requisitos no funcionales 3](#_Toc99907436)

[Requisitos de información. 3](#_Toc99907437)

# ¿En qué consiste?

Un videojuego de supervivencia por rondas en las cuales aparecerán enemigos en función de la ronda en la que se encuentre el jugador y la dificultad escogida por el mismo. Por tanto, el objetivo de este juego es sobrevivir el mayor número de rondas posibles llegando a la ronda número 100 para ganar. El jugador dispone de una barra de salud de 0 a 100 que empezará al completo. Cuando los enemigos golpeen al jugador este perderá vida, en función del daño que cause, que podrá recuperar cogiendo los diferentes botiquines que aparecerán en puntos preestablecidos del mapa. Si la salud del jugador llega a 0, la partida termina automáticamente indicando el número de rondas que has superado. Así mismo, el jugador dispondrá también de una bolsa de granadas. Las granadas se obtienen con una probabilidad del 10% al matar a un enemigo y serán lanzadas por el jugador en la dirección a la que mire el personaje, eliminando a todo enemigo dentro del rango cubierto por la explosión eléctrica.

Además de esto, se pretende añadir una funcionalidad de compra de armas por puntos conseguidos al eliminar enemigos. Se colocarán siluetas de diferentes armas por todo el mapa de forma que, al acercarse el jugador, podrá comprarlas si dispone de los puntos suficientes. Facilitando de esta forma la supervivencia frente a enemigos cada vez más fuertes y diversos.

## Opciones de dificultad.

El juego dispone de 3 modos de dificultad (Fácil, Medio y Difícil) que funcionarán de la siguiente manera:

* **Modo Fácil:** Se generarán enemigos entre el número de ronda y el doble. Por ejemplo: en la ronda 5 se generarán entre 5 y 10 enemigos. El precio de las armas será el preestablecido.
* **Modo Medio:** Se generarán enemigos entre el doble del número de ronda y el cuádruple. Por ejemplo: en la ronda 5 se generarán entre 10 y 20 enemigos. El precio de las armas será un 25% más caro.
* **Modo Difícil:** Se generarán enemigos entre el triple del número de ronda y el séxtuple. Por ejemplo: en la ronda 5 se generarán entre 15 y 30 enemigos. El precio de las armas será un 50% más caro.

En todos los modos de dificultad la salud y el daño de los enemigos serán el mismo, tan solo cambiará el número de enemigos generados y el precio de las armas.

# Análisis de requisitos.

## Requisitos funcionales

Un requisito funcional define una función del sistema de software o sus componentes. Una función es descrita como un conjunto de entradas, comportamientos y salidas.

Los requisitos funcionales del usuario de 3Dslug son los siguientes:

1. Manejar al personaje.
2. Posicionar la cámara.
3. Atacar a los enemigos. Habrá dos tipos de ataque (cuerpo a cuerpo y a distancia) que cambiarán en función del arma equipada.
4. Lanzar objetos arrojadizos.
5. Recoger botiquines dispersos por el mapa.
6. Comprar armas empleando los puntos conseguidos al abatir enemigos.
7. Cambiar entre un total de 2 armas que se guardarán en el inventario. Al comprar un arma nueva, esta ocupará el espacio de la que tenga equipada en ese momento, dejándola caer al suelo.
8. Pasar de un escenario a otro sobreviviendo a un número determinado de rondas (Lo decidiré mas adelante) con la salud, puntos y armas conseguidos hasta el momento, pero perdiendo todas las armas que queden tiradas en el suelo.
9. Recoger las armas que se hayan dejado caer al comprar nuevas, siguiendo la misma normativa de intercambio por el arma equipada.
10. Parar el juego en cualquier momento.
11. Guardar el estado de la partida en cualquier momento. De forma que al volver a abrir el juego, el usuario pueda retomar la partida por donde la dejó, sin perder sus puntos ni armas.

## Requisitos no funcionales

Un requisito no funcional o atributo de calidad es, en la ingeniería de sistemas y la ingeniería de software, un requisito que especifica criterios que pueden usarse para juzgar la operación de un sistema en lugar de sus comportamientos específicos, ya que estos corresponden a los requisitos funcionales.

Por tanto, se refieren a todos los requisitos que no describen información a guardar, ni funciones a realizar, sino características de funcionamiento. Por esto, suelen denominarse atributos de calidad de un sistema. Queda entonces el requisito no funcional, que son las restricciones o condiciones que impone el cliente al programa que necesita, por ejemplo, el tiempo de entrega del programa, el lenguaje o la cantidad de usuarios.

Entonces, los requisitos no funcionales de nuestra aplicación serán los siguientes:

1. El Front-End se desarrollará en Unity, utilizando assets públicos prediseñados y animaciones descargadas y diseñadas. Se utilizarán Prefabs con aquellos objetos que lo precisen, como pueden ser los enemigos, las armas o incluso el personaje jugador.
2. El Back-End se desarrollará en visual studio utilizando lenguaje C# apropiado para juegos de Unity, utilizando los métodos heredados de la clase MonoBehaviour. Estos scripts irán asignados a los GameObjects o Prefabs a los que vaya destinado su código.
3. Para guardar el estado de la partida, se generará un fichero de datos que será almacenado en una carpeta interna del móvil. (Estudiable utilizar firebase)
4. Se utilizará base de datos No-SQL con firebase para guardar un top de los usuarios que hayan conseguido las mejores puntuaciones.
5. Se pedirá un nombre al usuario al terminar la partida para guardarlo en el top de los mejores indicado anteriormente.

## Requisitos de información.

Los requisitos de información son aquellos que representan entidades e información relevante con las que el producto software va a operar.

En el caso de 3DSlug, Estos requisitos son:

**Usuario →**

* Nombre otorgado por teclado al terminar la partida.
* Puntuación obtenida durante la partida, contando los puntos con los que cuente al terminar más la suma de todos los gastos que haya hecho el jugador al comprar armas. Por ejemplo, si tiene 6000 puntos acumulados al terminar la partida y ha gastado 24000 puntos en armas, el puntaje final será 6000+24000 = 30000 ptos.

**Partida →**

* Se guardará el estado de la escena en el momento de presionar la opción de guardar partida. De forma que, al volver a iniciar el juego, el usuario pueda retomar la partida tal como la dejó.