Miembros del grupo

|  |  |
| --- | --- |
| **Nombre** | **Apellidos** |
| Víctor  Manuel  David  Juan Alberto  Fernando | Alonso Manavi  Barrenechea González  Fernández Cid  Gómez Vega  Murua Alcazar |

Versiónes

|  |  |  |
| --- | --- | --- |
| **Versión** | **Fecha** | **Comentarios** |
| 1.0.0. Alfa | 1-02-20 | Menú principal |
| 1.0.1 Alfa | 15-02-20 | Menú de opciones y movimiento básico de jugador |
| 1.0.2 Alfa | 29-02-20 | Menú para creación de juego, nuevos enemigos implementado en el juego. |
| 1.0.3 Alfa | 10-03-20 | Ataques del personaje |
| 1.1.0 Alfa | 15-03-20 | Unión de las interfaces principales con la primera pantalla del juego y el personaje. |

Especificación de Requisitos de Software

Formato IEEE 830

[**Introducción**](#_npbjqvklurii) **2**

[1.1 Propósito](#_vgwhlvydyflq) 2

[1.2 Alcance](#_3iupoz3hz3yz) 2

[1.3 Definiciones, siglas y abreviaturas](#_pn85vcrayoqt) 2

[1.4 Referencias](#_dyc67je6wpfy) 2

[**Descripción global**](#_ni14snwmfxl) **2**

[2.1 Perspectiva del producto](#_5i1gydnccy2y) 2

[2.2 Funciones del producto](#_b59y1d9x0isf) 3

[2.3 Características del usuario](#_xmsbjdpnxcyh) 3

[2.4 Restricciones](#_t86wm43dboq8) 3

[2.5 Suposiciones y Dependencias](#_sks1bcxzgtfo) 3

[2.6 Requisitos Futuros](#_iwj7gw6nqcco) 3

[**Requisitos Específicos**](#_9sxyzcqgclpt) **4**

[3.1 Interfaces externas](#_1zstwsnnohx3) 5

[3.2 Funciones](#_o477g2sdhruf) 5

[3.3 Requisitos de rendimiento](#_77ox7212z8qg) 6

[3.4 Restricciones de diseño](#_wgqfqie9iqr) 6

[3.5 Atributos del sistema](#_ubfl91nn8z5t) 6

[3.6 Otros requisitos](#_mp2if0w0k4sl) 6

[**Apéndices**](#_5sd0lcjml53) **6**

## Introducción

### 1.1 Propósito

El propósito de este documento es definir las especificaciones funcionales y no funcionales para un videojuego tipo Metroidvania.

### 1.2 Alcance

Nuestro producto se llama World off Mind. Es una aventura de acción 2D ambientada en un gran mundo interconectado. El usuario interactúa con el videojuego mediante el teclado y el ratón. El jugador adquirirá diferentes habilidades a lo largo del juego que le permitirá ir pasando por las diferentes pantallas hasta la finalización del juego.

### 1.3 Definiciones, siglas y abreviaturas

WOM : World off mind

### 1.4 Referencias

|  |  |
| --- | --- |
| **Título del documento** | **Referencia** |
| Enlace de github al proyecto | [*https://github.com/bgmanuel99/World-off-Mind*](https://github.com/bgmanuel99/World-off-Mind) |
| Especificación del videojuego | *Se encuentra en la carpeta documentación* |
| Plan de trabajo | *Se encuentra en la carpeta documentación* |
| Plan de riesgos | *Se encuentra en la carpeta documentación* |

## Descripción global

### 2.1 Perspectiva del producto

Nuestro videojuego será experienciado por el jugador mediante una serie de interfaces que le brindarán la oportunidad de personalizar las opciones a su gusto. Será controlado por teclado y ratón dando la posibilidad de modificar los atajos de teclado.

### 2.2 Funciones del producto

- El usuario podrá seleccionar en el menú entre empezar juego, opciones, logros y salir.

- El usuario podrá iniciar el juego desde el principio.

- El usuario podrá continuar el juego desde cinco slots diferentes.

- El usuario podrá personalizar el video, el audio, el juego, el idioma y los atajos de teclado.

- El usuario podrá guardar los avances del juego en cualquier punto del mismo.

### 2.3 Características del usuario

|  |  |
| --- | --- |
| **Tipo de usuario** | Jugador |
| **Criterio** | El usuario no cuenta con una contraseña para el acceso, podrá acceder desde su dispositivo en el que haya guardado su progreso, siendo este cualquier usuario. |

### 2.4 Restricciones

* No es apto para ejecutarlo en entornos laborales.

### 2.5 Suposiciones y Dependencias

El videojuego es ejecutable en sistema Windows 10 (64 bits).

### 2.6 Requisitos Futuros

Implementación en el market de Android y en otras plataformas diferentes a Windows.

## Requisitos Específicos (Pendiente)

*Esta sección contiene los requisitos a un nivel de detalle suficiente como para permitir a los diseñadores diseñar un sistema que satisfaga estos requisitos, y que permita al equipo de pruebas planificar y realizar las pruebas que demuestren si el sistema satisface, o no, los requisitos. Todo requisito aquí especificado describirá comportamientos externos del sistema, perceptibles por parte de los usuarios, operadores y otros sistemas.*

*Esta es la sección más larga e importante de la ERS. Deberán aplicarse los siguientes principios:*

* *El documento debería ser perfectamente legible por personas de muy distintas formaciones e intereses.*
* *Deberán referenciarse aquellos documentos relevantes que poseen alguna influencia sobre los requisitos.*
* *Todo requisito debería ser unívocamente identificable mediante algún código o sistema de numeración adecuado.*
* *Lo ideal, aunque en la práctica no siempre realizable, es que los requisitos posean las siguientes características*
  + *Corrección: La corrección de la ERS implica que el sistema implementado será el sistema deseado.*
  + *No son ambiguos: Cada requisito tiene una sola interpretación. Para eliminar la ambigüedad inherente a los requisitos expresados en lenguaje natural, se deberán utilizar gráficos o notaciones formales. En el caso de utilizar términos que, habitualmente, poseen más de una interpretación, se definirán con precisión en el glosario*
  + *Completos: Todos los requisitos relevantes han sido incluidos en la ERS. Deben incluir todas las posibles respuestas del sistema a los datos de entrada, tanto válidos como no válidos*
  + *Consistentes: Los requisitos no pueden ser contradictorios. Un conjunto de requisitos contradictorio no es implementable.*
  + *Clasificados: Normalmente, no todos los requisitos son igual de importantes. Los requisitos pueden clasificarse por importancia (esenciales, condicionales u opcionales) o por estabilidad (cambios que se espera que afecten al requisito). Esto sirve, ante todo, para no emplear excesivos recursos en implementar requisitos no esenciales.*
  + *Verificables: La ERS es verificable si y sólo si todos sus requisitos son verificables. Un requisito es verificable (testeable) si existe un proceso finito y no costoso para demostrar que el sistema cumple con el requisito. Un requisito ambiguo no es, en general, verificable. Expresiones como a veces, bien, adecuado, etc. introducen ambigüedad en los requisitos. Requisitos como “en caso de accidente la nube tóxica no se extenderá más allá de 25Km” no es verificable por el alto costo que conlleva.*

### 3.1 Interfaces externas

*Se describirán los requisitos que afecten a la interfaz de usuario, interfaz con otros sistemas (hardware y software) e interfaces de comunicaciones. Debe ser una descripción detallada de todas las entradas y salidas del sistema del software, incluyendo:*

* *la descripción de propósito*
* *la fuente de entrada o destino de salida*
* *el rango válido*
* *exactitud y/o tolerancia*
* *las unidades de medida*
* *tiempos*
* *las relaciones a otras entradas/salidas*
* *el formato de pantalla /organización*
* *el formato de ventanas/organización*
* *los formatos de los datos*
* *los formatos de los comandos*
* *fin de mensajes*

### 3.2 Funciones

*Esta subsección (la más larga del documento) deberá especificar todas aquellas acciones (funciones) que deberá llevar a cabo el software. Si se considera necesario, podrán utilizarse notaciones gráficas.*

*Las funciones pueden organizarse de múltiples formas, sugiriendo las siguientes:*

* *Por tipos de usuario: Distintos usuarios poseen distintos requisitos. Para cada clase de usuario que exista en la organización, se especificarán los requisitos funcionales que le afecten o tengan mayor relación con sus tareas.*
* *Por objetos: Los objetos son entidades del mundo real que serán reflejadas en el sistema. Para cada objeto, se detallarán sus atributos y sus funciones. Los objetos pueden agruparse en clases. Esta organización de la ERS no quiere decir que el diseño del sistema siga el paradigma de Orientación a Objetos.*
* *Por objetivos: Un objetivo es un servicio que se desea que ofrezca el sistema y que requiere una determinada entrada para obtener su resultado. Para cada objetivo o sub- objetivo que se persiga con el sistema, se detallarán las funciones que permitan llevarlo a cabo.*
* *Por estímulos: Se especificarán los posibles estímulos que recibe el sistema y las funciones relacionadas con dicho estímulo.*
* *Por jerarquía funcional: funciones que comparten entradas, salidas o datos internos. Se detallarán las funciones (entrada, proceso, salida) y las subfunciones del sistema.*

### 3.3 Requisitos de rendimiento

**Requisitos mínimos:**

*Sistema operativo: Windows 7*

*Procesador: Intel Core i5 4460*

*Memoria:*

*Tarjeta gráfica:*

*Almacenamiento:*

*DirectX: versión 11.*

**Requisitos recomendados:**

*Sistema operativo: Windows 10*

*Procesador:*

*Memoria:*

*Tarjeta gráfica:*

*Almacenamiento:*

*DirectX: versión 11.*

*Se detallarán los requisitos relacionados con la carga que se espera tenga que soportar el sistema. Por ejemplo, el número de terminales, el número esperado de usuarios simultáneamente conectados, número de transacciones por segundo que deberá soportar el sistema, etc. También, si es necesario, se especificarán lo requisitos de datos, es decir, aquellos requisitos que afecten a la información que se guardará en la base de datos. Por ejemplo, la frecuencia de uso, las capacidades de acceso y la cantidad de registros que se espera almacenar (decenas, cientos, miles o millones).*

*Todos que estos requisitos deben declararse en las condiciones mensurables. Por ejemplo, 95% de las transacciones se procesarán en menos de 1 seg.*

### 3.4 Restricciones de diseño

*Todo aquello que restrinja las decisiones relativas al diseño de la aplicación: Restricciones de otros estándares, limitaciones del hardware, etc. Por ejemplo, un requisito de rastro de auditoría puede declarar que deben grabarse todos los cambios a un banco de datos de la nómina en un archivo del rastro con los valores antes del proceso y después del proceso.*

### 3.5 Atributos del sistema

*Se detallarán los atributos de calidad del sistema: Fiabilidad, mantenibilidad, portabilidad, y la seguridad. Debería especificarse qué tipos de usuario están autorizados, o no, a realizar ciertas tareas, y cómo se implementarán los mecanismos de seguridad (por ejemplo, por medio de un login y una password).*

### 3.6 Otros requisitos

*Cualquier otro requisito que no encaje en otra sección*

## Apéndices

*Pueden contener todo tipo de información relevante para la ERS pero que, propiamente, no forme parte de la ERS. Por ejemplo: formatos de entrada/salida de datos, por pantalla o en listados o resultados de análisis de costes.*