**Final Rouge**



**Un juego de aventura, orientado a la exploración y a la supervivencia.**

**Integrantes:** *Alpuin Schunk Leonardo, Iaizzo Stefano, Ellacopulos Jonathan Jesús.*

**24 de Octubre de 2019**

**FINAL ROGUE**  
  
**DESCRIPCIÓN**  
 El personaje despierta en una habitación desconocida. Pronto descubre que está inmerso en un laberinto. Para descubrir cómo salir debe sobrevivir al mismo, el cual está plagado de entes extraños, a los cuales tiene que sobrevivir. La exploración, adquisición e implementación de los objetos encontrados guiarán al jugador a la escapatoria, o a la muerte del prisionero encarnado.

**MECANICAS DE JUEGO**

**FINAL ROGUE** es un juego del estilo Roguelike, el cual consiste en la exploración y supervivencia en un mapa generado proceduralmente. Esto significa que la experiencia de juego varía cada vez que se juega al mismo, dado a que la configuración del mapa es diferente cada vez que se inicia un juego nuevo.

**CLASES**

Según la elección de la clase, al inicio del juego, se dispondrá de las habilidades y atributos únicos de:

GUERRERO, MAGO, O PICARO. Cada uno con un enfoque diferente a la hora de avanzar las etapas del juego.

**GUERRERO:** Su estrategia central es destruir a cuanto enemigo se le cruce para adquirir los elementos necesarios para sobrevivir y avanzar en la mazmorra.  
**MAGO:** Su gran ventaja yace en la manipulación de la magia, aunque adquiridas con cuantiosas horas de estudio. Este anciano posee la fragilidad del cristal pero la fuerza destructiva de un volcán. Sin embargo su gran sabiduría y conocimiento serán sus principales ventajas  
**PICARO:** La filosofía del pícaro es el engaño, las trampas y el sigilo. Avanza con cautela pero con precisión quirúrgica.

**OBJETIVO**

Para progresar en la partida se deberá explorar el mapa en busca de “ítems” que faciliten la supervivencia del personaje, ya sean armas, llaves, o consumibles. Algunas zonas del mapa serán inaccesibles sin objetos clave. A su vez, la necesidad de recolección de estos objetos llevará al jugador a enfrentarse con los monstruos que habitan en la mazmorra.  
Para escapar de la misma se deberá vencer al guardia que protege la salida.

**CONTROLES**  
Es un juego manejado por completo con teclado.

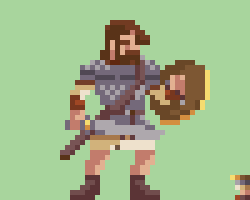
Personajes:



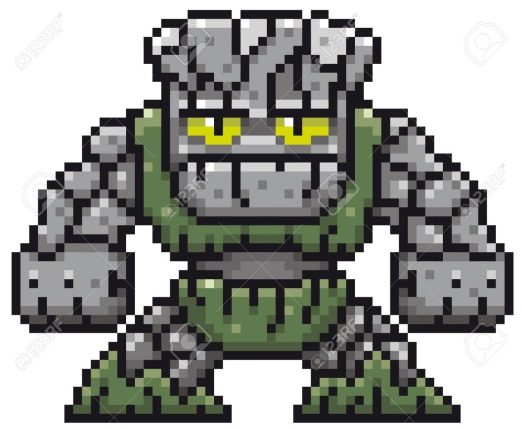
Rogue, un aventurero novato, pero muy astuto, que va aprendiendo y evolucionando sus tácticas de combate y supervivencia conforme avanza su exploración. Se caracteriza por su agilidad y destreza. Tiene una habilidad especial para las trampas y para desactivar mecanismos.



Wizard, un sabiondo de la los hechizos y la magia, su principal característica son sus conjuros y su capacidad de entablar combates a distancia. No son muy resistentes y tampoco poseen gran cantidad de salud. De todas formas, tienden a ser muy útiles en las aventuras.



Warrior, un poderoso guerrero de mano firme, capaz de derribar a cualquiera que se interponga en su camino blandiendo su poderosa espada. Posee mucha fuerza y salud, es muy bueno en los combates cuerpo a cuerpo. Sin embargo, es un tanto más lento. La destreza y la agilidad no son sus más fieles aliados.

Monstruo de piedra, una abominación de 2 metros de alto, muy aterradora y altamente resistente en combates cuerpo a cuerpo pero débil ante los hechizos y conjuros. Puede ser muy agresiva si se le molesta, pero usualmente son bastante torpes.



Monstruo de lava, esta extraña criatura tiene la característica de calentar las cosas nada más con su presencia. No son muy usuales pero pueden llegar a ser muy peligrosos y agresivos. Además de poseer poderosos ataques que pueden derretir a sus víctimas.



Monstruo de hielo, probablemente el más peligroso y despiadado de los 3. No tiene compasión con sus víctimas, persigue hasta matar. Para fortuna del protagonista son lentos y no poseen tanta fuerza. Además de ser poco resistentes a los ataques cuerpo a cuerpo.

ARCHIVOS USADOS:

* **Main.cpp**

**Inicia el programa con un menú con opciones de las que se derivan los siguientes archivos:**

* + - **“Partida guardada.bkp” contiene un registro del mapa con sus cambios, ítems que posee el jugador y posición de Personas jugador, enemigo, objetos, a través de sus clases**
    - **“Mapa.dat” genera un mapa nuevo con una distribución procedural de enemigos y objetos, ubica la posición de persona jugador, a través de sus clases.**
* **Mapa.dat**
  + - **Se genera a través de un algoritmo procedural al inicio de un juego nuevo.**
* **Imágenes.bmp**
  + - **Se sustraen imágenes de este archivo para la parte grafica del juego.**
* **Items.dat**
  + - **Cantidad finita y preestablecida de objetos del juego.**
* **Equipo.dat**
  + - **Equipo inicial, según la clase selecta.**
* **Equipo.bkp**
  + - **Equipo adquirido con el progreso del juego.**
* **Enemigo.dat**
  + - **Enemigos distintos prediseñados, los cuales van a ser ubicados de forma procedural.**
* **Jugador.dat**
  + - **Contiene las estadísticas de las tres distintas clases de personaje.**

**DIAGRAMA DE CLASES**

***\*Registro items***

***\*Registro de ítems a parte***

Equipo:

Public:

Métodos:

Equipar();

Desequipar()

C

Items:

Private:

* Fuerza
* Destreza
* Inteligencia
* Vida

Métodos:

* Drop();
* Efectp();
* Gets();
* Sets();

Persona:

Private:

* Eje\_x\_relativo (int)
* Eje\_y\_relativo (int)
* Eje\_x\_mapa (int)
* Eje\_y\_mapa (int)
* Vida(int)
* Nombre(char)

Public:

Métodos:

* Movimiento();
* Gets();
* Sets();

Hijo

Jugador:

Private:

* Fuerza (int)
* Destreza (int)
* Inteligencia (int)
* Equipo

Métodos:

* Movimiento();
* Accion();
* Gets();
* Sets();

Hijo

Enemigo:

Private:

* Atributo\_mas (int)
* Atributo\_menos (int)

Métodos:

* Drop();
* Comportamiento();
* Gets();
* Sets();

***\*Actualizacio de mapa***

***\*Registro salvado***

Mapa:

Private:

* Mapa\_Guia[][] (int)
* Mapa\_Juego[][](int)

Métodos:

Getters();

Setters();

***\*Interaccion***

***\*Salvar y guardar***

***\*El juego cicla en main luego del menú inicial siempre preguntando si hubo alguna interacción.***

***\*Los atributos sin tipo de dato aclarado, son clases***

Salvar:

Public:

* Persona
* Mapa
* Item

Metodos:

* Guardar();
* Cargar();

Jugabilidad:

Public:

Métodos:

* Interaccion()
* Baja\_persona()
* Validar\_mapa()
* Opciones();
* Guardar();