DUNGEONS

Versión 0.2a

MANUAL DE USUARIO

DESARROLLADORES:

Quintana, Eduardo Nicolás

Vignoli, Lucas

INDICE:

¿Como Jugar?........................................................................2

Controles......................................................................3

¿Como Crear un Nivel?..........................................................6

Informacion de desarrollo......................................................7

Nombre de archivo........................................................7

Estructura de cabecera...................................................7

Lista de funciones..........................................................7

¿COMO JUGAR?

Para jugar, deberá existir un nivel creado previamente, de lo contrario no se podrá ejecutar el juego. Si desea saber cómo crear un nivel, o incorporar uno creado por otro jugador, por favor diríjase a la sección ¿Como crear un nivel? pag. 4 .

Para iniciar un juego, deberá seleccionar la opción "Jugar" del menú principal.

El nivel se cargará e inmediatamente podrá comenzar a jugar.

REGLAS:

Las reglas son simples, el jugador deberá explorar los calabozos, hasta encontrar reliquias, las cuales deberán ser regresadas a las zonas de entrega. En el camino, deberá resolver laberintos y problemas que se le presentaran debido al diseño del nivel, y a los dragones que pueden existir en las habitaciones.

Tenga cuidado!, los dragones pueden volar por encima de las paredes, e intentaran atraparlo lo más rápido posible.

En ocasiones podrá ver trampas en los calabozos, y deberá utilizarlas para inutilizar a los dragones. Para ello deberá conducirlos y hacer que entren en las trampas. Una vez allí, un dragón no se moverá.

Tenga en cuenta que al salir de la habitación, los dragones se liberan!

Para recoger una reliquia basta con pasar sobre ella. Recuerde que no podrá recoger más de una reliquia a la vez.

Si posee una reliquia, bastara con pasar sobre una zona de entrega para poder entregarla.

En el pie del mapa se mostrará información acerca de cuantas reliquias quedan para ganar.

El juego termina cuando el jugador completa el objetivo, o es atrapado por un dragón.

CONTROLES

Para mover al jugador solo tendrá que usar las flechas del teclado.

¿COMO CREAR UN NIVEL?

Antes de empezar se recomienda tener el programa abierto, y un diseño básico de las posiciones y conexiones de cada escena.

PROCEDIMIENTO:

1. Para crear un nivel deberá ir a la opción "Crear un nuevo Juego" del menú principal. Tenga en cuenta que al finalizar la creación de un nuevo nivel, si existía uno antes, será reemplazado.
2. Una vez que hayamos entrado en dicha opción, se nos pedirá la cantidad de escenas, una escena es una habitación o un calabozo.
3. Luego se nos pedirá la cantidad de filas y columnas, que definen las dimensiones de cada una de las escenas.
4. A continuación comenzara el proceso de configuración de puerta.

En este, se nos pedirá que coloquemos el identificador de la escena contigua a la actual, y a la cual podremos llegar atravesando una puerta en particular. Para comprender mejor como configurar una puerta es importante tener en mente el siguiente esquema:

Puerta 1

Puerta 2

Puerta 4

Escena X

Puerta 3

Las reglas son simples, si en una escena existe una conexión con otra, la puerta correspondiente deberá tener el identificador de la escena con la cual se conecta. Esto significa que una puerta en una escena determinada no puede dirigir hacia la misma escena. También quiere decir que no es posible que una puerta dirija hacia una escena que no existe. (Estas limitaciones están consideradas en el programa).

De esta forma, si tenemos 2 escenas, una al lado de la otra, la configuración para la escena 1 seria:

Puerta 1 = 0

Puerta 2 = 2 (id de la escena 2)

Puerta 3 = 0

Puerta 4 = 0

Y para la escena 2:

Puerta 1 = 0

Puerta 2 = 0

Puerta 3 = 0

Puerta 4 = 1 (id de la escena 1)

1. Una vez configuradas las puertas, se concluye con el diseño de la escena en sí, es decir el mapa de la escena. Para ello contara con una unica herrmanienta llamada pincel la cual se mueve con las flechas del teclado.

El pincel dibuja por defecto un camino por el cual el jugador podra moverse. De aqui en mas es su eleccion diseñar un nivel simple o complejo.

Podra observar debajo del mapa una lista de referencia que le permitira dibujar otros elementos ademas de caminos.

Estas opciones cambian la cabeza del pincel (el cual seguirá dibujando caminos por defecto). Para seleccionar una nueva cabeza de pincel, presione el número correspondiente al elemento que desea pintar. Notara que si sigue moviendo el pincel, este dibujara solo caminos.

Para dibujar el elemento seleccionado en la cabeza del pincel deberá presionar las teclas WASD de acuerdo a la dirección en la que desea pintar.

Recuerde que no puede pintar fuera de las dimensiones del mapa.

Para dibujar una puerta solo debe dibujar un camino que llegue hasta el límite del mapa. Si no lo hace y la escena tenía una puerta configurada, el jugador jamás podrá pasarla.

Una vez dibujado completamente el mapa lo único que queda es guardarlo.

CONSEJO: Para evitar problemas a la hora de jugar, se recomienda tener en cuenta la simetria con respecto a las puertas.

1. Para guardar el mapa dibujado, simplemente presione Escape. El proceso se repetirá hasta haber configurado y dibujado todas las escenas.

ADVERTENCIA!: En el punto 5, recuerde dibujar al menos una reliquia y una zona de entrega, de lo contrario su juego podrá ser recorrido pero nunca podrá ser ganado.

INFORMACION DE DESARROLLO

NOMBRE DE ARCHIVO:

"Nivel.dat"

ESTRUCTURA DE CABECERA:

Cant. de escenas | Cant. de Columnas | Cant. de Filas | Cant. de Reliquias

LISTA DE FUNCIONES Y DESCRIPCIÓN:

Scene \* Create\_Level()

Función que crea una lista enlazada de escenas y retorna la escena inicial (siempre será la escena en la que aparezca el jugador).

Scene \* New\_Scene(Header \* header, int id)

Función que crea una nueva escena vacía con id especifico. Recibe la cabecera y el id y retorna la escena completamente creada.

void Create\_Empty\_Map(Map \* map, int amount\_rows, int amount\_columns)

Función que crea un mapa vacio (solo contiene paredes). Recibe como parámetros un puntero al mapa, y las dimensiones del mismo.

void Draw\_Map(Map \* map);

Función que dibuja en pantalla un mapa. Recibe un puntero al mapa a dibujar.

void Draw\_Map(Map \* map, Enemy\* enemies[50], int \* countEnemy); Función que dibuja el mapa. Específicamente usada durante el juego y no la creación de mapas. Similar al anterior, pero en este caso, checkea la existencia de enemigos en el mapa y llena el vector de punteros, así también como asigna la cantidad de enemigos.

Recibe un puntero al mapa, un vector de puntero a enemigos y un puntero al contador de enemigos.

void Draw\_Palette();

Función que dibuja la paleta que indica cuales son las posibles brushheads (cabezas de pincel).

void Draw\_Brush(Brush \* brush);

Función que dibuja el pincel en las coordenadas correspondientes. Recibe un puntero al pincel.

void Input\_Check\_Brush( Map \* map, Brush \* brush, int \* key, int\*items); Función que realiza el checkeo de teclas presionadas durante la creación de un mapa.

Recibe un puntero al mapa, un puntero al pincel, un puntero a la tecla presionada y un puntero al contador de reliquias.

bool Brush\_Movement(Map \* map, Brush \* brush, int \* key);

Función que mueve el pincel, retorna verdadero si las teclas presionadas son para mover el pincel (flechas).

Recibe un puntero al mapa, un puntero al pincel y un puntero a la tecla presionada.

bool Brush\_Head\_Selection(Brush \* brush, int \* key);

Función que selecciona el tipo de cabeza de pincel a pintar, devuelve verdadero si las teclas presionadas corresponden a la selección de cabeza de pincel(1,2,3,4,5). Recibe un puntero al pincel y un puntero a la tecla presionada

bool Draw\_Brushhead(Map \* map, Brush \* brush , int \* key, int\* items); Función que pinta la cabeza de pincel, retorna verdadero si las teclas presionadas son para pintar la cabeza de pincel (WASD). Si se pinta sobre una reliquia, se disminuye la cantidad de reliquias totales del juego. Si se pinta una reliquia, se aumenta este contador. Recibe un puntero al mapa, un puntero al pincel, un puntero a la tecla presionada y un puntero al contador de reliquias

void Generate\_Way(Map \* map, Brush \* brush, int \* items);

Función que genera el camino tras el pincel. Si el pincel pasa sobre una reliquia disminuye la cantidad de reliquias totales en el juego. Recibe un puntero al mapa, un puntero al pincel y un puntero al contador de reliquias.

Scene\* Search\_Scene\_By\_Id(Scene\* mainScene, int id);

Función que devuelve una escena dentro de una lista enlazada, cuyo id es el especificado. Recibe un puntero a la escena inicial de la lista enlazada y un entero que representa el id de la escena a buscar.

void Save\_Level (Scene\* Aux, Header \* header);

Función que guarda el nivel en un archivo binario. Recibe un puntero a la escena inicial, y un puntero a la cabecera.

Scene\* Load\_All\_Scenes(int\* items);

Función que carga todas las escenas, armando una lista enlazada y devolviendo la primera escena cargada. Recibe un puntero al contador de reliquias.

Scene\* Load\_Scene(FILE\* file, int dimy, int dimx);

Función que carga una escena de un archivo binario y devuelve un puntero a ella. Recibe un puntero al archivo y las dimensiones del mapa.

int Get\_Amount\_Scenes(Scene \* scene);

Función que dada un escena inicial de una lista enlazada, retorna la cantidad de escenas en dicha lista. Recibe un puntero a la escena a partir de la cual se desea contar.

void Modify();

Función que permite la modificación de escenas concretas.

void Modify\_Scene(Scene \* mainscene, Scene \* scene, int \* items, int amount);

Función que hace uso de funciones anteriores para modificar una escena en particular. Recibe un puntero a la escena inicial de la lista enlazada, un puntero a la escena que se desea modificar, un puntero al contador de reliquias y un entero que contiene la cantidad de escenas de todo el nivel.

void Menu();

Función que contiene toda la lógica de interacción del menú.

void HowToPlay(int \*);

Función que muestra la información sobre cómo jugar el juego. Recibe un puntero a entero que representa la opción de menú a la que se retornara luego de leer la información.

void HowToCreate(int \*);

Función que muestra la información sobre cómo crear un nivel completo. Recibe un puntero a entero que representa la opción de menú a la que se retornara luego de leer la información.

bool GameLoop(Scene \* mainScene, int \*items);

Función que contiene el juego propiamente dicho. Se encarga del movimiento del jugador y enemigos, control de teclas, control de reliquias y carga de escenas. Recibe un puntero a la escena inicial de la lista enlazada y un puntero al contador de reliquias.

bool Enemy\_Movement(Enemy \* enemy, Jugador \* jugador, Map \* map);

Función que se encarga de manejar la inteligencia artificial de los enemigos. Recibe un puntero al enemigo que se desea mover, un puntero al jugador y un puntero al mapa.

void \* Save\_Memory(int amount);

Función para reservar memoria. Recibe un entero que representa el tamaño de la reserva.