

Mazmorras en Java

Introducción

En este proyecto de programación, nos proponemos desarrollar un juego clásico de mazmorras utilizando el lenguaje Java. Las mazmorras, con su ambiente misterioso y desafiante, han sido una parte integral del mundo de los videojuegos, ofreciendo a los jugadores emocionantes aventuras llenas de peligros y tesoros por descubrir.

Nuestro objetivo principal es crear una experiencia de juego inmersiva y divertida, aprovechando los principios de la programación orientada a objetos y los algoritmos de juego. Englobaremos todos los aspectos del desarrollo desde la planificación inicial hasta la implementación final del juego, abordando aspectos como la utilización de estructuras eficientes de datos, la inteligencia artificial de los enemigos y la interfaz de usuario.

Descripción de la aplicación

La aplicación iniciará con una pantalla de bienvenida donde se pedirá al jugador que introduzca su nombre. Al pulsar el botón de jugar, la aplicación cargará toda la información de una serie de ficheros a las clases correspondientes, tras lo cual se mostrará el tablero de juego con los distintos elementos dispuestos en él.

El tablero consiste en una matriz de casillas. Las casillas pueden ser suelo o paredes. Solo las casillas de suelo pueden contener elementos mientras que las paredes no se pueden atravesar de ninguna manera.

En el tablero se disponen los elementos interactivos del juego. Los más importantes son los personajes. El juego contará con un personaje principal y una serie de enemigos. Todos los personajes cuentan con un nombre, puntos de vida, fuerza, defensa y velocidad. El personaje principal, además, contará con un objeto equipable en su mano derecha y un segundo en su mano izquierda. Por su parte, los enemigos tendrán un valor de percepción que indicará la distancia a la que pueden ver al jugador.

En la interacción entre el jugador y los enemigos intervienen una serie de habilidades y resistencias. Un ejemplo de posibles habilidades es: contundente, perforante, cortante, fuego, magia y abrir. Los enemigos tienen asignadas una serie de habilidades que pueden usar. Por su parte, el jugador tiene acceso a esas habilidades a través de sus objetos equipables. Todos los personajes contarán con una serie de resistencias que influyen en la cantidad de daño que reciben de una habilidad. Los distintos tipos de resistencias son los siguientes:

Letal	El ataque acaba con el personaje	Resistente	Mitad de daño
Frágil	Cuatro veces el daño	Irrompible	Un cuarto del daño
Vulnerable	Doble de daño	Absorbente	Recupera vida con el daño

Neutral	Daño normal	Inmune	No recibe daño
---------	-------------	--------	----------------

Los personajes, por tanto, contarán con una relación entre las distintas habilidades y la resistencia a cada habilidad. Si una habilidad no aparece en la lista de resistencia se considera que es inmune.

El jugador interactúa con el juego a través de las flechas de dirección. El personaje principal se mueve por el tablero usando las flechas. Podrá moverse a una casilla de tipo suelo que no esté ocupada. No se podrá mover hacia una pared. Si la casilla a la que se mueve contiene un elemento interactivo, interactuará con él.

Los enemigos actúan por su propia cuenta. La inteligencia artificial del enemigo funciona de la siguiente manera:

- Si el jugador está a su lado, interactuará con él.
- Si el jugador está en su rango de visión (atributo percepción) le perseguirá
- Si el jugador está más allá de su rango de visión se moverá a una casilla adyacente al azar.

El jugador basa sus capacidades en los distintos objetos que tiene a su disposición. Encontramos dos tipos de objetos: los equipables y los consumibles. Los objetos equipables son aquellos que el personaje principal puede llevar en alguna de sus manos y que tienen las habilidades y los atributos que permiten atacar e interactuar con los elementos del entorno. Cuentan además con una serie de datos asociados que serán usados para los cálculos de daño. Por otro lado, los objetos consumibles recuperarán una cantidad de vida al personaje principal y desaparecerán. Deberá poder existir un tercer tipo de objeto llamado artefacto que será equipable y consumible al mismo tiempo. Los objetos podrán obtenerse al derrotar enemigos o al abrir o romper ciertos elementos en el entorno y serán equipados o usados al momento de obtenerlos.

En el escenario se pueden encontrar distintos objetos que dificultan o ayudan con el avance del jugador (como puertas o cofres) Estos elementos tendrán un nombre, una cantidad de “vida”, una barrera de daño y una forma de reflejar si está o no roto. Además contendrá una lista de resistencias a habilidades al igual que los personajes. Estos elementos tendrán la posibilidad de soltar algún objeto al romperse/abrirse (Lo normal sería que estos objetos tengan una resistencia de tipo letal a la habilidad abrir).

Cuando el jugador se mueva hacia algún elemento interactivo, interactuará con él. La interacción encapsulará toda la información que será recibida por el elemento (apuntar, daño, habilidades que intervienen). El jugador realizará un uso de su mano derecha y otro de su mano izquierda. El azar tendrá un papel importante en el éxito o fracaso de las interacciones:

- Apuntar: el jugador tendrá un valor para apuntar igual al resultado de lanzar un dado de 20 caras más la suma de su fuerza.
- Daño: el valor de daño de la interacción será igual al resultado de una serie de dados asociados con el objeto equipable más la fuerza del jugador. (Por ejemplo, puede lanzar dos dados de 6 caras) Con la mano secundaria no se suma la fuerza del jugador en el daño.

Los ataques de los enemigos funcionan de forma parecida, siendo estos los que tienen definidos una serie de datos para el cálculo de daño.

Dependiendo del elemento interactivo que reciba la interacción el comportamiento será diferente:

Si se trata de un personaje (el personaje principal o un enemigo), primero se busca cual es la resistencia recibida de la interacción más desfavorable. Posteriormente, se compara el valor de puntería de la interacción con su defensa, si la supera se reduce la vida una cantidad igual al valor de daño (con la modificación correspondiente por las resistencias).

Si es un elemento del entorno no se utiliza el valor de puntería. Como contraparte, el valor de daño, con las modificaciones por la resistencia tiene que superar la barrera de daño del elemento para que tenga algún efecto.

La acción del juego se organiza en turnos. Los turnos son organizados con la pulsación de una tecla de dirección y supone la realización de un movimiento o interacción por parte del jugador o de los enemigos. El orden de actuación se organiza por medio del atributo velocidad de los personajes, siendo los más rápidos los que realizarán su acción primero.

Se dedicará una clase gestor de juego a la organización del juego (gestión de turnos, creación y borrado de elementos, cambios de pantalla...).

Extras

Se podrán incluir algunas modificaciones en el proyecto para aumentar la puntuación:

- Objetos defensivos: Se trata de objetos equipables que aumentan las capacidades defensivas del personaje principal.
- Objetos consumibles complejos: Los objetos consumibles pueden tener un tipo asociado y realizar otras acciones además de la curación de heridas (aumento de algún atributo o cualquier otro efecto a discreción del programador)
- Inventario: En lugar de usar o equipar los objetos al recogerlos, se almacenarán en un inventario de objetos. Se deberá añadir una interfaz gráfica para el manejo de objetos desde donde se podrán usar o equipar.
- Bosses: Los bosses actuarán de la misma manera que los enemigos, pero tendrán una acción especial y poderosa con una serie de turnos de recarga.
- NPCs amigables: Se podrán encontrar personajes en la mazmorra que no sean hostiles. Estos personajes tendrán alguna línea de diálogo y podrán otorgar objetos al jugador.
- Interfaz gráfica: Se valorará una interfaz gráfica personalizada, haciendo uso de elementos gráficos y de hojas de estilo.
- Otras modificaciones: Se escucharán propuestas.

A continuación se proponen una serie de modificaciones especialmente complicadas.

- Generación aleatoria de escenario: Además de la generación mediante fichero, se ofrecerá un modo de generación aleatoria. El escenario generado debe tener sentido (con pasillos y salas).

- Algoritmos de IA complejos: Se implementará un algoritmo de búsqueda de camino (path tracking) para los enemigos que les permita esquivar mejor los obstáculos (Algoritmo A*).

Entregables

A lo largo de la realización del proyecto se realizarán entregas parciales. La entrega final consistirá en:

- Documentación del proyecto.
 - Guía de usuario.
 - Guía del programador.
 - Documentación interna con formato Javadoc.
- Repositorio del proyecto.