

Juego de Mazmorras en Java - Fase 1

Objetivo

El propósito de este proyecto es diseñar y desarrollar un videojuego clásico de mazmorreo en el que un personaje protagonista se enfrentará a los peligros de un laberinto.

Requisitos

Requisitos Funcionales

El juego contará con las siguientes características:

1. Escenario:

- El escenario se cargará desde un fichero y se almacenará internamente en una matriz.
- Cada celda de la matriz representará un suelo o una pared.
- Las paredes serán obstáculos y no podrán ser atravesadas por ningún personaje.
- El escenario se visualizará mediante una GridPane u otro componente adecuado de JavaFX, con una representación visual clara de cada tipo de celda.
- En cada casilla podrá situarse un personaje. Solo puede situarse uno por casilla.

2. Personajes:

- Habrá un único **personaje protagonista** controlado por el jugador. La información sobre las estadísticas del personaje protagonista será introducida por el jugador en una vista dedicada a ello al iniciar la aplicación.
- Se incluirán **enemigos** repartidos por el escenario, cuya información se cargará desde un fichero.
- El protagonista podrá desplazarse utilizando las teclas de flecha (↑ ↓ ← →) o las teclas **WASD**.
- Los enemigos se moverán automáticamente:
 - Si están a una distancia inferior o igual a su atributo de **percepción**, se moverán en dirección al jugador.
 - Si están más lejos, se moverán en una **dirección aleatoria**.
- Los personajes actuarán secuencialmente, uno después de otro. Para decidir el orden de actuación se incluirá un atributo de velocidad.

3. Combate:

- Si un personaje se desplaza hacia una casilla ocupada por otro personaje (enemigo o protagonista), en lugar de realizar el movimiento **se producirá un ataque**.
- El diseño del sistema de combate (cálculo de daño, estadísticas, puntos de vida, etc.) será responsabilidad del programador.
- La partida acabará cuando el personaje protagonista pierda todos sus puntos de vida.

- Cuando no queden enemigos en el tablero ocurrirá una de las siguientes situaciones (a elegir):
 - Acaba la partida.
 - Se genera un nuevo tablero con nuevos enemigos.

4. Carga de datos:

- El escenario y los enemigos se cargarán desde **ficheros externos**.
- Los datos del jugador se pedirán en una vista al iniciar la aplicación.
- Se espera que el juego sea capaz de interpretar estos ficheros correctamente al inicio y reflejar su contenido en la interfaz.

5. Interfaz gráfica:

La interfaz del juego debe mostrar, como mínimo, la siguiente información:

- El **tablero de juego** con:
 - Suelos y paredes.
 - Posiciones actualizadas del protagonista y los enemigos.
- Las **estadísticas del protagonista** (por ejemplo: salud, fuerza, defensa, etc.).
- La **salud actual de cada enemigo**.
- El **orden de turnos**, indicando quién actúa a continuación.

Requisitos Técnicos

1. Tecnología principal

- La interfaz gráfica debe estar implementada utilizando **JavaFX**.
- Los nodos gráficos se crearán combinando archivos FXML y generación dinámica en código Java.

2. Arquitectura del proyecto

- El proyecto debe seguir el **patrón Modelo-Vista-Controlador (MVC)**:
 - **Modelo**: Contendrá los datos del juego (escenario, personajes, lógica de movimiento y combate).
 - **Vista**: Será responsable de representar gráficamente el escenario y la información del juego.
 - **Controlador**: Gestionará las entradas del usuario y coordinará las actualizaciones entre el modelo y la vista.

3. Comunicación entre capas

- Se deben utilizar **observadores** para notificar a la vista de los cambios en el modelo:
 - Cuando el modelo cambie (por ejemplo, un personaje se mueva, reciba daño o muera), la vista debe actualizarse automáticamente.
 - Se utilizará el modelo de observadores y suscripción.

Documentación

Junto con el proyecto se deberá entregar.

Guía del programador: Documentación javadoc exportada a html.

Guía de usuario: PDF breve donde se describa el funcionamiento de la aplicación.

Entrega

El proyecto será realizado en grupos de dos o tres personas. La entrega será realizada a través de GitHub, con las siguientes consideraciones:

- Uno de los miembros de la pareja deberá crear un repositorio donde ambos trabajarán.
- Se realizarán commits habitualmente al repositorio.
- Se revisarán las contribuciones de los miembros del equipo, por lo que debe utilizarse GitHub correctamente.
- El proyecto incluirá bajo una etiqueta de tipo @author el nombre de los participantes.
- En la entrega se añadirá un enlace al repositorio (deberá ser público).

Evaluación

Criterio	Puntos	Criterio	Puntos
1. Escenario	10	7. Carga de datos	10
2. Movimiento del protagonista	10	8. Creación del protagonista	5
3. Movimiento de enemigos	10	9. Arquitectura MVC	10
4. Combate	15	10. Observadores	10
5. Turnos	5	11. Documentación	5
6. Interfaz gráfica	10	12. Defensa	APTO / NO APTO

Penalizaciones

Las siguientes situaciones supondrán una calificación de 0:

- La entrega está fuera de plazo o no se realiza por los mecanismos solicitados.
- El juego no es jugable.
- No se incluye documentación de ningún tipo.
- No se aporta al equipo de trabajo.
- El proyecto está plagado.

Defensa

Para comprobar la autoría del proyecto se realizará una defensa del mismo.

- La defensa se realizará de manera individual.
- Consistirá en una modificación del proyecto original en un tiempo limitado.
- La defensa deberá ser APTA para que el proyecto sea evaluado.
- Se realizará el día 14 de Mayo.