

A. Enunciado proyecto UML

Proyecto UML - Entornos de Desarrollo

Objetivo: Comprender y aplicar los diagramas UML en la planificación y estructuración de un minijuego. El trabajo se realizará en parejas.

Minijuego: “Aventuras en el Laberinto”, donde el jugador navega por un laberinto lleno de trampas y tesoros.

Tareas:

1. **Identificar las Clases:** Jugador, Laberinto, Trampa, Tesoro.
2. **Definir Atributos y Métodos:** vida y puntuación para Jugador y Tesoro respectivamente. Métodos como `moverJugador()` y `activarTrampa()`.
3. **Establecer Relaciones:** Mostrar asociaciones y herencia si es aplicable.

Entrega del Proyecto: El proyecto ha de tener portada e índice y constará de dos partes principales:

Parte Teórica: Presentar y explicar los tres tipos de diagramas UML: Clases, Secuencia y Estados. Explicar qué es cada tipo de diagrama, la nomenclatura utilizada y proporcionar ejemplos.

Parte Práctica: Explicar el minijuego creado, describir las clases utilizadas y cómo el código Java implementa estas estructuras y comportamientos. Generar y explicar los tres tipos de diagramas UML para el juego.

Este proyecto permitirá aplicar la teoría a la práctica, visualizando cómo los diagramas UML se traducen en código real y funcional para un videojuego.

B. Explicación detallada:

1. Como futuros desarrolladores de software, es crucial que comprendan cómo los diagramas UML pueden ayudarles a visualizar y planificar la estructura de un videojuego. El trabajo se ha de hacer en grupos de 2 personas. El minijuego puede ser de temas libres o similar al siguiente.

Por ejemplo, trabajarán (en un minijuego llamado "Aventuras en el Laberinto", donde el jugador debe navegar a través de un laberinto lleno de trampas y tesoros).

2. Objetivos del Proyecto:

- Crear *Diagramas de Clases UML* para definir las entidades del juego, sus atributos y métodos, y las relaciones entre ellas.
- Desarrollar *Diagramas de Secuencia UML* para ilustrar la interacción entre objetos durante eventos clave del juego.
- Elaborar *Diagramas de Estados UML* para representar los cambios de estado de la entidad 'Jugador' en respuesta a eventos del juego.

3. Descripción del Minijuego:

- El jugador comienza en la entrada del laberinto y debe encontrar la salida.
- Se encuentran trampas que pueden disminuir la vida del jugador.
- Hay tesoros que aumentan la puntuación del jugador.

4. Tareas Específicas:

- *Identificar las Clases*: Determinen las clases necesarias, como `Jugador`, `Laberinto`, `Trampa`, y `Tesoro`.
- *Definir Atributos y Métodos*: Asignen atributos como `vida` para `Jugador` y `puntuación` para `Tesoro`. Definan métodos como `moverJugador()` y `activarTrampa()`.
- *Establecer Relaciones*: Utilicen la nomenclatura UML para mostrar asociaciones, como la relación entre `Jugador` y `Laberinto`, y herencia, si es aplicable.

Entrega del Proyecto:

El proyecto ha de tener portada e índice. El proyecto constará de dos partes principales:

Parte Teórica: Los estudiantes deben presentar y explicar los tres tipos de diagramas UML creados: Diagramas de Clases, Diagramas de Secuencia y Diagramas de Estados. Deben explicar qué es cada tipo de diagrama, la nomenclatura utilizada en estos diagramas y proporcionar ejemplos.

1. **Diagramas de Clases UML:** Deben definir qué es un Diagrama de Clases, la nomenclatura utilizada y cómo se representan las entidades, atributos, métodos y relaciones entre ellas.
2. **Diagramas de Secuencia UML:** Deben explicar qué es un Diagrama de Secuencia, la nomenclatura utilizada y cómo se ilustra la interacción entre objetos durante eventos clave.
3. **Diagramas de Estados UML:** Deben describir qué es un Diagrama de Estados, la nomenclatura utilizada y cómo se representan los cambios de estado de una entidad en respuesta a eventos.

Parte Práctica: En grupos de dos personas, los estudiantes deben explicar el minijuego que han creado. Deben describir qué hace el juego, describir las clases utilizadas y cómo el código Java implementa estas estructuras y comportamientos. Además, deben generar y explicar los tres tipos de diagramas UML para su juego.