

Proyecto Global

Diseño y Desarrollo de un Juego de Combate

Introducción

El objetivo de este proyecto es desarrollar un juego de ordenador en el que el protagonista vaya combatiendo contra los enemigos que se va encontrando, utilizando para ello diferentes armas, tipos de ataque, etc.

Dinámica del juego:

- El juego se juega por consola, por lo que no tendrá interfaz gráfica. Pero si alguien sabe y/o quiere puede hacerla.
- Cada combate se resolverá por turnos: se elige al azar quién ataca primero, y a partir de ese momento el jugador y la máquina se van turnando.
- En cada turno, el jugador podrá elegir entre una serie de acciones disponibles (diferentes tipos de ataques, armas, defenderse, no hacer nada...).
- Tras cada ataque, el programa evaluará los daños causados al oponente. Pierde el combate aquel contrincante que primero se queda sin vida.
- La evaluación de los daños se realizará de manera algorítmica (podéis inventar la fórmula que queráis para el reparto de los puntos).
- El juego continúa combate tras combate, contra distintos tipos de enemigos, hasta que el jugador pierde. Con cada combate ganado el jugador acumula puntos.
- Los combates se desarrollarán en distintos mundos/escenarios.

Personajes:

- Todos los personajes, tanto el jugador como los enemigos, tendrán una serie de atributos que afectarán a los resultados de los combates (fuerza, resistencia, agilidad...).
- Al principio de la partida el jugador definirá los valores que su personaje va a tener para cada atributo.
- Habrá distintas categorías de enemigos (guerreros, hechiceros, mutantes...), cada uno con sus características propias.



- Las mismas categorías aplican en todos los mundos/escenarios, pero en cada uno de dichos mundos cada categoría de enemigo presenta características diferentes (por ejemplo, los brujos de la selva podrían hacer vuelos cortos y los brujos del mar vuelos largos).
- Cada enemigo se definirá también por una determinada estrategia, de entre varias disponibles (defensiva, agresiva...).
- Un personaje puede pasar por varios estados a lo largo de un combate (herido, paralizado, activo...). Pasará de un estado a otro como consecuencia de los distintos eventos que vayan ocurriendo a lo largo del combate (pérdida de mucha vida, recuperación de vida por consumo de alguna poción, paralizado por culpa de un hechizo...).

Tareas a realizar

Al final del cuatrimestre se deberá entregar un juego funcionalmente completo que incluya como mínimo los siguientes desarrollos:

1. **(1 punto)** Se deberá utilizar el patrón **Strategy** para gestionar el conjunto de estrategias posibles y la estrategia que en cada combate adopte el enemigo de turno.
2. **(1 punto)** Se deberá emplear un patrón **Decorator** para gestionar las distintas acciones a realizar por los personajes: se decorará la acción básica con los modificadores adecuados dependiendo de la estrategia y las características del personaje. Por ejemplo, si el personaje realiza un 'ataque con espada', este ataque habrá que decorarlo para hacerlo un 'ataque poderoso con espada' si el personaje tiene un alto valor de su atributo Fuerza.
3. **(1 punto)** Se deberá emplear un patrón **State** para controlar el estado de los personajes durante el combate, de forma que los personajes vayan pasando de un estado a otro dependiendo de las incidencias del combate.
4. **(2 puntos)** Se deberá emplear un patrón **Abstract Factory** para crear los diferentes enemigos. Existirán varios tipos de enemigos, comunes a todos los mundos/escenarios pero adaptados a cada mundo/escenario.
5. **(1 punto)** Se deberá emplear un patrón **Singleton** en la clase encargada de hacer los cálculos sobre el resultado de los ataques, para que sólo haya una instancia de dicho 'calculador' en el sistema.
6. **(1 punto)** Se deberá emplear un patrón **Template Method** para implementar en cada clase Enemigo el algoritmo para decidir cuál será la siguiente acción a realizar.
7. **(2 puntos)** Se deberá realizar un documento de memoria que recoja:
 - La descripción del juego.
 - Un pequeño manual de uso.
 - El diagrama de clases de cada patrón empleado, convenientemente explicado.
8. **(1 punto)** Se reserva esta puntuación para la claridad, sencillez y estilo del código. Se valorarán los comentarios, indentación, etc.

Normativa de realización, entrega y evaluación del proyecto:

- El proyecto se realizará y entregará por parejas.
- El proyecto se realizará únicamente mediante el lenguaje de programación Java.
- La entrega deberá hacerse mediante *Blackboard* antes del 22-05-2020 a las 23:59 horas (hora peninsular en España).
- La entrega se compondrá de un único fichero comprimido que contendrá el proyecto exportado de eclipse, o el código fuente si se hace con otro IDE, y la memoria.
- El nombre del fichero será el nombre del enunciado de la práctica junto con el de los alumnos y su primer apellido escritos en letras mayúsculas y separados mediante un guión bajo '_'. Ejemplo: PROYECTO_ITZIAR_MORALES_LUCAS_GARCÍA.zip
- El último apartado puntuará si (y sólo si) al menos algún otro apartado se ha resuelto correctamente.
- Los apartados puntúan como un todo. Si un apartado se resuelve al completo correctamente, el alumno obtendrá los puntos asignados a dicho apartado. En cualquier otro caso, no obtendrá ningún punto.
- Cualquier sospecha de copia entre dos o más prácticas derivará en la calificación de 0 para todos los alumnos involucrados.
- Los proyectos que den errores de compilación no obtendrán ningún punto.