TPO:
Juego de Aventura

Etapa 1

*Facultad de Ingeniería y Ciencias Exactas (**FAIN**)*

Ingeniería en Informática

Materia: Paradigma Orientado a Objetos

Día-Turno: Viernes - Mañana

Profesor: Sauczuk Martín Pablo

Alumnos: Araujo Nicolás - Charelli Franco - Di Lalla Thiago - Nemi Santiago - Sanchez Santiago

Año 2024

Estrategia General del Diseño

El diseño del sistema se centra en un juego de combate, donde los personajes (Personaje) participan en peleas (Pelea), utilizando atributos, herramientas y tipos para superar a sus oponentes. Aquí explicamos las estrategias clave para cada elemento del sistema:

La estrategia en la que se basa este sistema corresponde al juego de una serie de interacciones entre las diferentes clases de personajes, de armas, de tipos de elementos y, por último, de reglas de combate que gestionan las batallas.
El flujo de la simulación comienza con la creación de los tipos (Fuego, Planta, Agua), los cuales son gestionados a partir de la clase Tipo. Cada tipo puede tener un "contrario", el cual define la ventaja y desventaja en la batalla. Por otro lado, tenemos que Fuego posee ventaja contra Planta, pero no contra Agua, y así sucesivamente. Esta relación entre tipos se especifica cuando se utiliza el método setContrario(), el cual establece las reglas básicas en las que se basan las batallas posteriores.

A continuación, se crea un personaje principal (en este caso, "Batman"), utilizando la clase Personaje. Este personaje tiene atributos como salud, ataque, defensa y tipo (en este caso, Fuego). Además, se le asignan herramientas (como armas y protecciones), que afectan su rendimiento durante la batalla. Las herramientas son gestionadas a través de la clase abstracta Herramienta, que se extiende en las clases Arma y Protección. Las herramientas tienen un nombre, tipo y atributos específicos (por ejemplo, daño o defensa), y se utilizan en la batalla para modificar las estadísticas del personaje. Las herramientas también tienen un número limitado de usos, que se decrementa con cada uso y pueden romperse si se agotan.

La clase Enemigo genera una lista de enemigos con diferentes tipos, armas y defensas, y cada uno tiene herramientas asociadas. Estos enemigos se enfrentan al héroe en diferentes rondas. Por ejemplo, el "Pinguino" utiliza una espada de fuego, que es un arma débil contra el tipo Fuego de Batman. Cada enemigo es creado con su propio conjunto de habilidades, lo que aumenta la complejidad de las batallas.

El sistema de batalla se gestiona a través de la interfaz Peleable y la clase Pelea. La interfaz define los métodos que deben implementarse para iniciar, ejecutar y finalizar las peleas entre dos personajes. La clase Pelea gestiona la dinámica del combate, donde los personajes atacan,

usan objetos y se protegen dependiendo de sus herramientas y habilidades. El combate continúa hasta que uno de los personajes se quede sin salud, lo que determina al ganador.

Durante la ejecución de la pelea se imprimen los estados de los personajes en cada ronda, encontrando así la salud que tienen cada uno de ellos y los objetos que se encuentran en uso. Mientras se encuentra un personaje derrotado, se imprime un mensaje de acabado y la ronda en la que se ha caído un personaje en pelea.

Para acabar la clase TorreDelPoder organizará las peleas en rondas y gestionará la lógica que comprueba si el personaje ha vencido a los enemigos o ha sido vencido frente a estos últimos. Cada ronda lleva a subir el nivel del personaje, llevándolo a subir sus atributos (salud, ataque, defensa) en base a avanzar sobre las rondas preseteadas. Si el personaje cae en alguna ronda la pelea finaliza.

En conclusión, el código implementa una serie de peleas complejas mediante reglas de tipo y uso de herramientas y características de personajes. La interacción entre las clases de tipos, personajes y herramientas permiten llevar a cabo la experiencia de pelea donde la forma de atacar y defender en cada round depende de la combinación de habilidades y objetos. Cada enemigo es más difícil que el anterior y el héroe tendrá que gestionar bien sus recursos si desea avanzar en la Torre del Poder y alcanzar el triunfo.

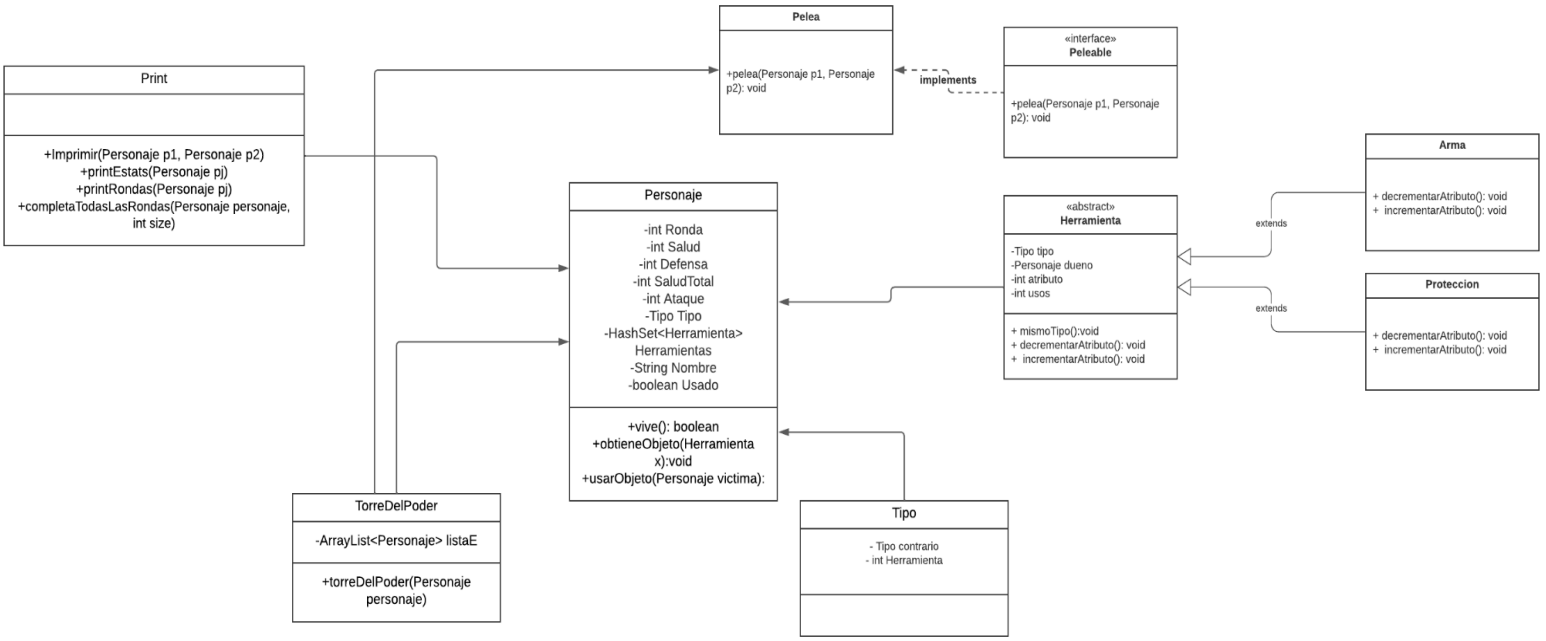
Conclusión

El sistema desarrollado demuestra el uso adecuado de las técnicas aprendidas a lo largo de la cursada. Aprendimos mejor cómo separar las responsabilidades entre las diferentes clases planteadas en el programa, permitiendo de cierta forma que futuras actualizaciones sean más sencillas de establecer. Esto es un claro ejemplo de cómo hemos podido emplear nuestras técnicas en el programa.

Este trabajo nos ayudó a comprender la importancia que necesita conocer las responsabilidades de cada clase, enseñándonos una forma más sencilla de poder acceder a ciertos elementos los cuales necesitamos a lo largo de una ejecución. Además, nos hizo mejorar en no tener tanto código repetido gracias a las clases abstractas las cuales utilizamos ampliamente.

Para finalizar, el trabajo es una demostración acorde a lo visto en clase, donde se aplican los métodos, clases y diversos recursos vistos a lo largo de la cursada. El trabajar en grupo nos facilitó el planteo y desarrollo de las ideas para este programa. Esto permitió diferentes perspectivas frente a las adversidades encontradas, hallando soluciones
más óptimas.

Diagrama de clases



Link del repositorio:

https://github.com/sdnxdxd/TPOparadigmagrupo13