

Trabajo Práctico 1

¿Cómo complicar el código?

Como tengo un Makefile por carpeta, para correr el código es nomás abrir la carpeta y escribir en la terminal "make"

Ejercicio 1

Personajes

Antes que nada explico, que por más comodidad separo cada clase en una carpeta para mantener un orden.

- la clase personajes es una interfaz (como pide el enunciado), donde los únicos métodos virtuales que tiene son básicamente los Get de todo, el manejo de las armas y las formas "básicas" de atacar (ataque_rapido y atacar_con_armas que lo tienen todos).
- Después viene la estructura de las clases abstractas, donde su utilidad básicamente es definir todas las variables básicas (vida, daño, armadura, etc.). Para que sean abstractas tienen ahí una función virtual cumplen la propiedad de abstracta.
- Bueno acá me tome realmente en serio lo que es la libertad de creatividad, puedo decir que todos los personajes derivados tienen 4 métodos únicos (menos el caballero, que bueno hasta ahí llego mi creatividad). También muchos de estos tienen atributos únicos relacionados a los métodos. Por ejemplo el oro que roba el mercenario le da la posibilidad de hacer críticos.
- Es así que cada personaje tiene todos los métodos y atributos propios por ser Mago/Guerrero sumado a los únicos que tienen.

Mecánicas extras

- También existe una mecánica que es global para todos los personajes que son los efectos (en el código está escrito como EFECTO). Existen 9 únicos efectos, los cuales dan distintos resultados al ser aplicados al personajes. 7 se consideran "negativos" y 2 "positivos"
- algunos de los ejemplos son:
 - PARALIZAR = no te deja atacar esa ronda
 - QUEMADURA = te hace daño y te haces menos daño
 - PROTECCION = el daño que te hacen es reducido entre un 20% a 100% (se decide por un rand()):

- ¿Cómo se consiguen estos efectos? por ataques de armas (después se explica cómo) o por ciertas habilidades (métodos) únicas, el hechicero tiene la opción de aplicarte el efecto “confusión”. Los efectos se guardan en un Vector que guarda un struct llamado EfectoActivo. Este struct tiene el efecto y su duración, por lo que (si se hace un juego de rondas, en donde cada ronda se hace un ataque/habilidad) cada ronda se procesan los efectos de todos los personajes (gracias a la función de personajes “procesar_efectos”).
- Aprovechando que estoy hablando de mecánicas extras, quería comentar la utilidad de todas los enum/struct repartidos por el código:
 - EFECTO = es un enum que contiene todos los posibles efectos que un jugador puede recibir. con esto se trabaja la decisión de cuál va a ser el resultado cuando se aplique un efecto en un jugador
 - TIPO_DAÑO = Permite distinguir qué tipo de daño va a recibir el jugador. Como hay distinta armadura para cada tipo de daño, no es lo mismo que te ataquen con daño físico que mágico
 - Marca = El mercenario tiene una mecánica parecida a los efectos, la cual trata de “marcar” enemigos en los cuales después, al pasar las rondas, se verán con un resultado dependiendo la marca. Por ejemplo está el “sicario”, el cual si pasan 2 rondas y el personajes se encuentra a menos de 15%, lo elimina.
 - InvocacionAnimal = El conjurador tiene la habilidad de poder invocar animales como aliados suyos. Estos podrían atacar si se les indica y además siempre recibirán primero el daño ellos antes que su invocador. Este struct guarda, la vida, daño, efecto que puede dar, probabilidad de aplicar y que tipo de invocación es.
 - INVOCACIONES_ANIMALES = Este enum es el que decide qué tipo de animal es, es una variable de InvocacionAnimal, el cual ayuda a decidir todas las estadísticas de la invocacion
 - INVOCACIONES_MUERTAS = es lo mismo al enum de arriba, solo que este tipo de invocación no tiene su propio struct. Las invocaciones se guardan en un vector que contiene la vida y que tipo de invocación es.
 - MATERIAL Y ENCANTAMIENTO = ambas son muy parecidas. Pertenecen a la clase de armas y les permite tener estadísticas extras. Cada arma va a tener un material (ArmaCombate) o un encantamiento (Item_magicos). cada uno se consigue aleatoriamente en el constructo de la clase. Por un lado los materiales le otorgan más o menos daño y el encantamiento tiene la posibilidad de poder aplicar el efecto al impacto.

Armas

Podría decir que la clase de armas es una versión “mini me” de lo que serían los personajes. No tiene el desarrollo que tuvo las armas pero aun cada arma tiene al menos dos ataques únicos del resto

- La clase interfaz “arma” sirve para poder incluir todos los getter básicos y también la función principal que es “atacar”. Esta función es la que es llamada cuando el personaje quiere usar las armas para atacar.

- Las clases abstractas sirven para poder ser iniciadoras de todas las variables básicas que puede tener un arma. Las mayores diferencias entre las armas de combate y los ítems mágicos es que, por un lado las de combate tienen material y únicamente daño físico y por el lado de los mágicos, encantamiento y daño mágico.
- Las clases derivadas no son un gran misterio, tienen, como ya dije 2 ataques distintos, que suelen distinguir en el gasto de durabilidad del arma y el daño que hace. Algunos tienen atributos únicos, como por ejemplo la lanza que puedes partirla a la mitad, o la espada que si llega a 20, puede aplicar hemorragia. También que la poción tiene la posibilidad que explote, causando mucho daño pero rompiendo el arma por completo

Equipo

Es una clase extra hecha tanto para la funcionalidad de ciertos personajes, como también para facilitar ciertas cosas en el ejercicio 2.

- Esta clase contiene como atributos al nombre de equipo, el tamaño, todos sus jugadores y los jugadores vivos en un vector aparte.
- Esta decisión de tener a un vector de jugadores vivos es útil para la pelea ya que los que van muriendo se van eliminando del vector de vivos así no estorban.
- Sus métodos son simples, puede ingresar jugadores, elimina los muertos del vector de vivos, devuelve la cantidad de vivos actuales, devuelve el vector de todos los personajes y también otro método devuelve únicamente los vivos.

Descripción general de los métodos de los personajes

Hago una breve descripción de las habilidades de los personajes, que es lo más complejo de mi código.

Bárbaro

- tiene la mecánica de la rabia que le aumenta el daño que hace, dependiendo de la rabia.
- esa misma rabia te permite hacer dos habilidades nuevas, como lo son romper huesos (daño que ignora la armadura), aplicar el efecto de miedo a los enemigos y también su habilidad definitiva que no puede morir por 3 rondas.

Caballero

- tiene una única habilidad: proteger a los compañeros.

Gladiador

esta bastante completito este personaje

- Su mecánica principal es la adrenalina. Esta se obtiene haciendo daño básicamente.
- Está adrenalina otorga tanto armadura extra como capacidad de esquivar los ataques

- tiene una habilidad, que consume adrenalina la cual aplica PARALIZAR al enemigo, además de hacer daño
- la capacidad de esquivar es activada cuando el gladiador decide “entretener al público”. Es ahí donde por cada golpe esquivado gana honor, pero si lo golpean pierde todo su honor y adrenalina.
- este honor le otorga más daño, además de habilitarlo a la habilidad de ignorar los efectos negativos

Mercenario

la mecánica principal del mercenario es **robar oro**

- el oro le otorga posibilidad de pegar crítico en los ataques
- con el oro puede contratar a un sicario que, pasado 2 rondas ejecuta al jugador marcado (marca que se procesa como los efectos) si tiene menos de 15% de vida.
- Tiene la habilidad definitiva que ataca con un cañón al enemigo marcado y hace bastante daño.

también tiene mecánicas distintas

- robar armas a los enemigos y conservarlas
- eliminar un compañero y ganar muchas estadísticas (mucho oro y vida, daño, etc.). Esto tiene una contra, que es si no hay compañeros vivos, se muere el.

Paladín

- puede usar el escudo que protege por completo un ataque, pero hace reducido mientras esté el escudo activo
- el paladín gana fe atacando dependiendo la cantidad de daño que hace. esta fe le otorga mayor estamina y vida.
- puede gastar fe en “bendecir las armas” (pegan más)
- también puede consumir fe en hacer un ataque que ignora la armadura

Brujo

- puede maldecir enemigo, que le otorga un efecto negativo random
- controlar a un enemigo, que lo haga atacar a uno random de su equipo
- si lo matan, entra en un estado, que si mata a uno en el siguiente ataque, se salva de la muerte
- puede activar un pacto, que sacrifica vida y mana, pero gana mucho daño por 3 rondas

Conjurador

- puede paralizar a un enemigo (efecto PARALIZAR)
- aplica un efecto aleatorio a un enemigo

su mecánica principal son las invocaciones de animales (que utiliza el struct y el enum)

- aca puede atacar con estos animales que puede llegar a aplicar ciertos efectos
- estos mismos pueden protegerlo cuando lo atacan (recibir el daño por el)

Su habilidad fuerte es invocar a un dragón que ataca a uno y le aplica quemadura

Hechizero

- aplica confusión a los enemigos (el enemigo tiene la posibilidad de atacarse a él mismo)
- aplicarse un escudo que protege de un daño específico (mágico o físico)
- restablecer parte de su mana
- clonarse (esto hace que el enemigo pueda errar el tiro y pegarle a un clon)
- potenciar el daño a un aliado
- tirar un rayo que pega mágico y tiene la posibilidad de pegar crítico

Nigromante

- puede revivir a los compañeros muertos con menos vida.
- drena la vida a un enemigo (le saca vida y se la suma a él)
- invocar muertos que pueden atacar al enemigo y también protegerlo de ataques
- mandar al reino de los muertos a un enemigo, donde le roba gana 20% de las estadísticas del enemigo, donde si muere ahí, el nigromante se queda esas estadísticas. Si No las pierde

Ejercicio 2

Aca creó la fábrica de personajes pedida.

- todas sus funciones son estáticas, así no es necesario una instancia para poder usar sus métodos

Las fábricas consisten de 3 métodos:

- crear personajes
- crear armas
- equipar personajes

Todas las clases reciben un parámetro que deciden si los personajes y/o armas se crean de manera aleatoria o de manera manual.

- Toda la construcción de personajes/armas es fundamentada gracias a 4 enum que representan todas las clases derivadas que hay para facilitar el código.

Añadido a esto, la función de equipar personajes también recibe como parámetro la cantidad de armas que desde la terminal se decide que tenga el personaje. Como es especificado, tiene que recibir un número entre 0 y 2. de ahí se crea un personaje y sus respectivas armas.

Sumado a esto, para la creación de los equipos usó la clase Equipo ya creada, donde ahí ya tengo una función para mostrar los personajes del equipo.

Tanto el tamaño del equipo, la cantidad de armas por jugador es creado con la función "rand". Además, para diferenciar una ejecución con la otra, importo <ctime> así creó una semilla srand(time(0)) que es distinta cada ejecución.

Ejercicio 3

El jugador pelea contra la "máquina" en un juego de piedra, papel o tijeras.

Nuevamente utilizó el srand(time(0)) así cada ejecución es distinta.

Cómo funciona el código:

- función `Decision_juego` = la función básica para decidir qué “ataque” hace en esa ronda el jugador
- De ahí convierto esa decisión en uno de los valores del enum definido `DECISION_JUEGO`, el cual contiene los 3 posibles movimientos del juego.
- de ahí toma las decisiones y las recibe la función de Ganador, que decide quién es el ganador del “piedra papel o tijera”
- Después de esto, concluye todo a una última función la cual dependiendo los resultados de la ronda, pinte en terminal quien atacó y con que, sumado a la vida de los jugadores