Proyecto diseño de software

Diseño y desarrollo de un juego de combate

INSO2

David Domínguez y Unai Puelles

# Descripción del juego

Este juego tiene una temática fantástica y ficticia de personajes genéricos. No hemos decidido un tema en específico, sino que, hemos optado por coger personajes de diferentes para que sea más divertido.

Este juego consiste en eliminar a dos rivales en dos mapas diferentes, por lo cual, va a haber un total de cuatro combates diferentes. Si en algún combate te quedas sin vida, el juego habrá terminado y si consigues ganar los cuatro combates, ganarás el juego.

# Manual de uso

Selección de personaje

Una vez empezada la ejecución del juego nos aparecerá la selección del personaje. Elegiremos entre: Guerrero, Mago y Gigante. Dependiendo de que personaje elijamos tendrá diferentes parámetros en los atributos. Lo siguiente que aparecerá es un menú en el que incrementaremos los valores de los atributos poniendo el número que tienen junto al nombre. Se incrementará de uno en uno y tendremos un total de ocho valores para incrementar en cualquiera de los atributos.

Además, si eliges bien los atributos del personaje que has elegido tendrá ataques especiales.

Combate

Hay un total de cuatro combates que seguirán el siguiente orden de enemigos y mapas:

1. **Combate**
   1. Enemigo: Dragón
   2. Mapa: Castle
2. **Combate**
   1. Enemigo: Orc
   2. Mapa: Meadow
3. C**ombate**
   1. Enemigo: Dragón
   2. Mapa: Meadow
4. **Combate**
   1. Enemigo: Orc
   2. Mapa: Castle

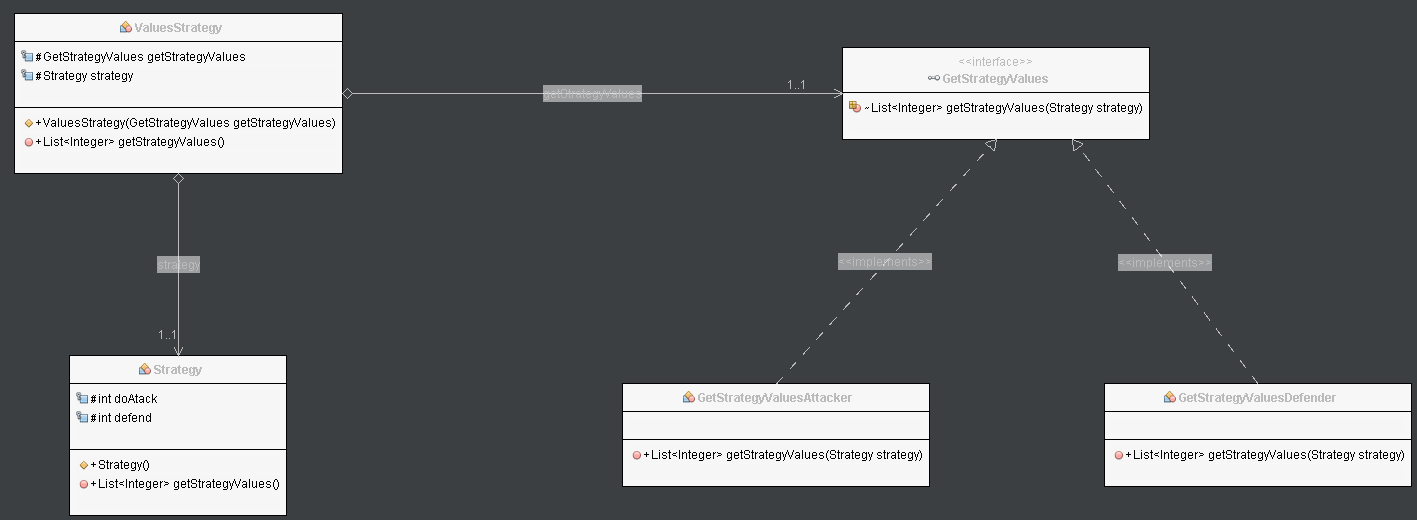
El dragón tiene una mejora cuando está en el mapa de Castle y el Orco tiene una mejora de los ataques cuando se encuentra en el mapa de Meadow.

El combate funciona por rondas y en cada ronda podrás atacar o curarte. Cuidado, porque puedes fallar cualquiera de las dos, perdiendo así el turno. Mediante los menús que van apareciendo elegirás la acción que quieres hacer y si eliges que quieres realizar un ataque, mediante el menú que aparece elegirás el ataque a realizar.

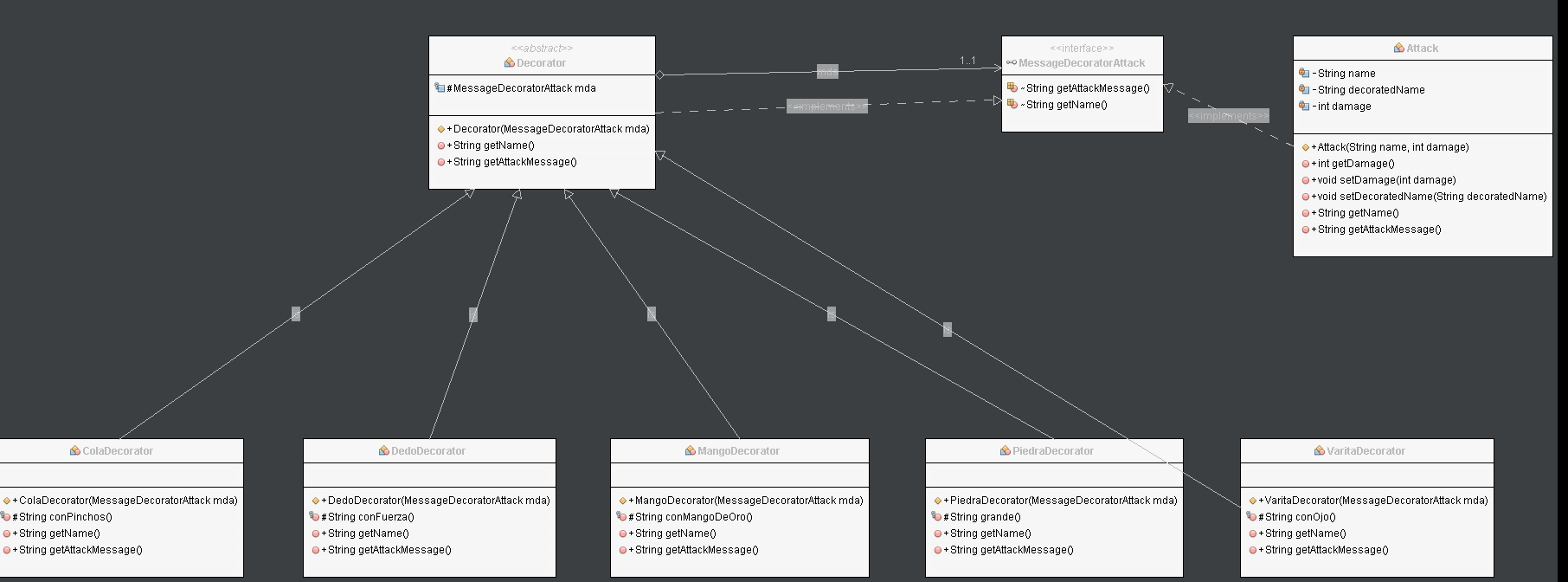
# Diagramas de clases

## Strategy

Este patrón se ha utilizado para definir la estrategia que va a utilizar el enemigo en el combate. Va a afectar en la decisión de atacar o curarse. Dependiendo de que estrategia se ha establecido va a haber más opciones de realizar un ataque o de recuperar vida. Establece diferentes valores según la clase que se instancia.

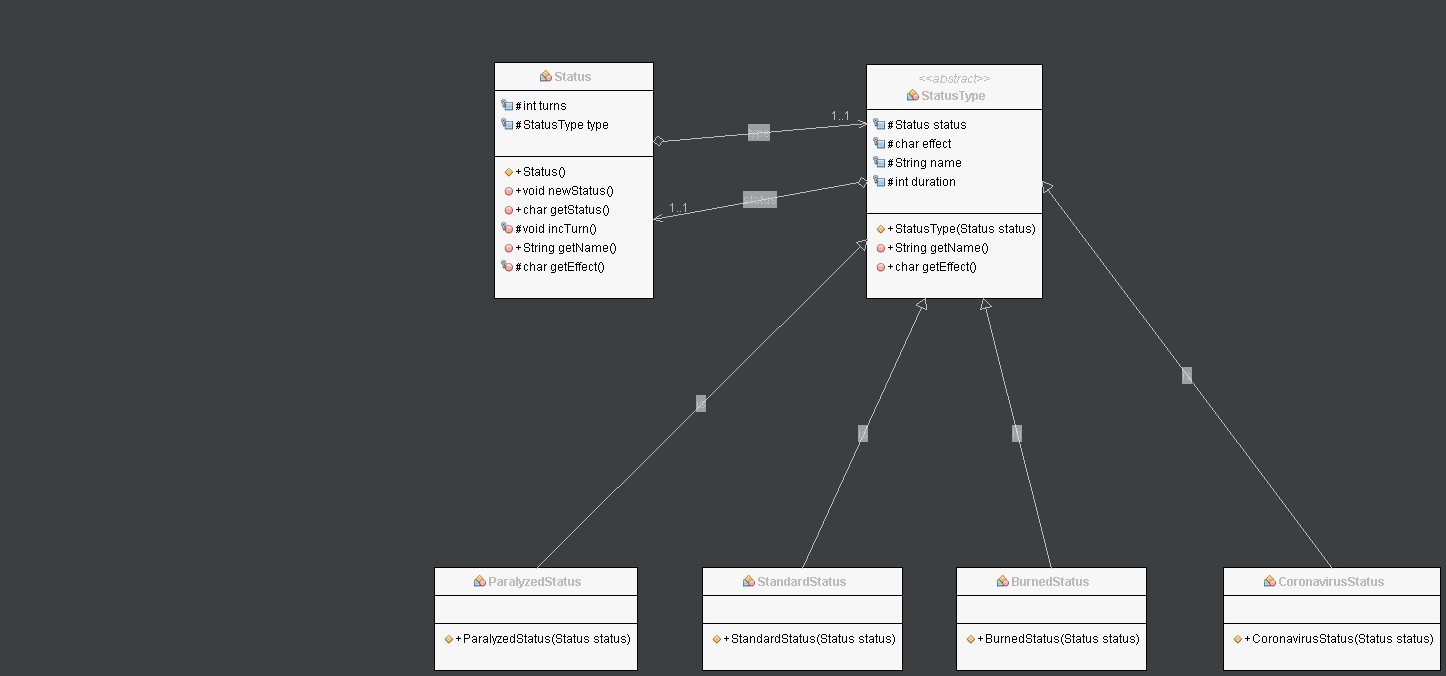


## Decorator

Tal y como se comentaba en el enunciado, el objetivo de este patrón es gestionar los ataques de los personajes y nosotros lo hemos implementado de la manera en que, si un personaje o enemigo tiene más de un valor en varios atributos, que se le mejore (decore) el ataque y que se haga más fuerte. El patrón lo que hace es un “override” al método para recoger el nombre del ataque y nos lo devuelve decorado.

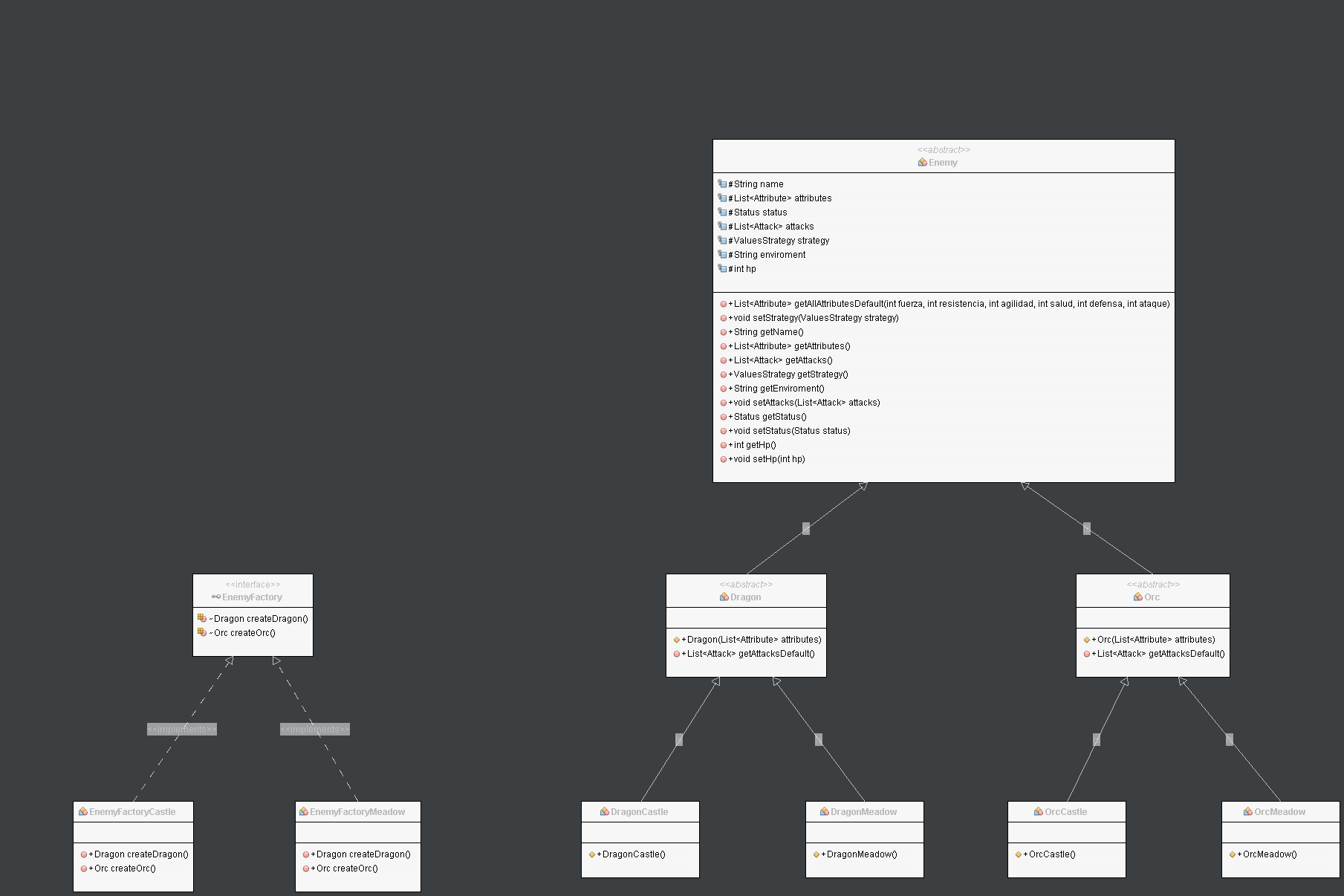
## State

El patrón state nos va a servir para determinar en que estado están los personajes. En nuestro caso los hemos dividido en normal; que como su nombre indica no va a pasar nada y es el estado por defecto del personaje/enemigo, paralizado; que no va a poder realizar nada en esa ronda, quemado y coronavirus; que nos resta vida cada ronda que pasa.

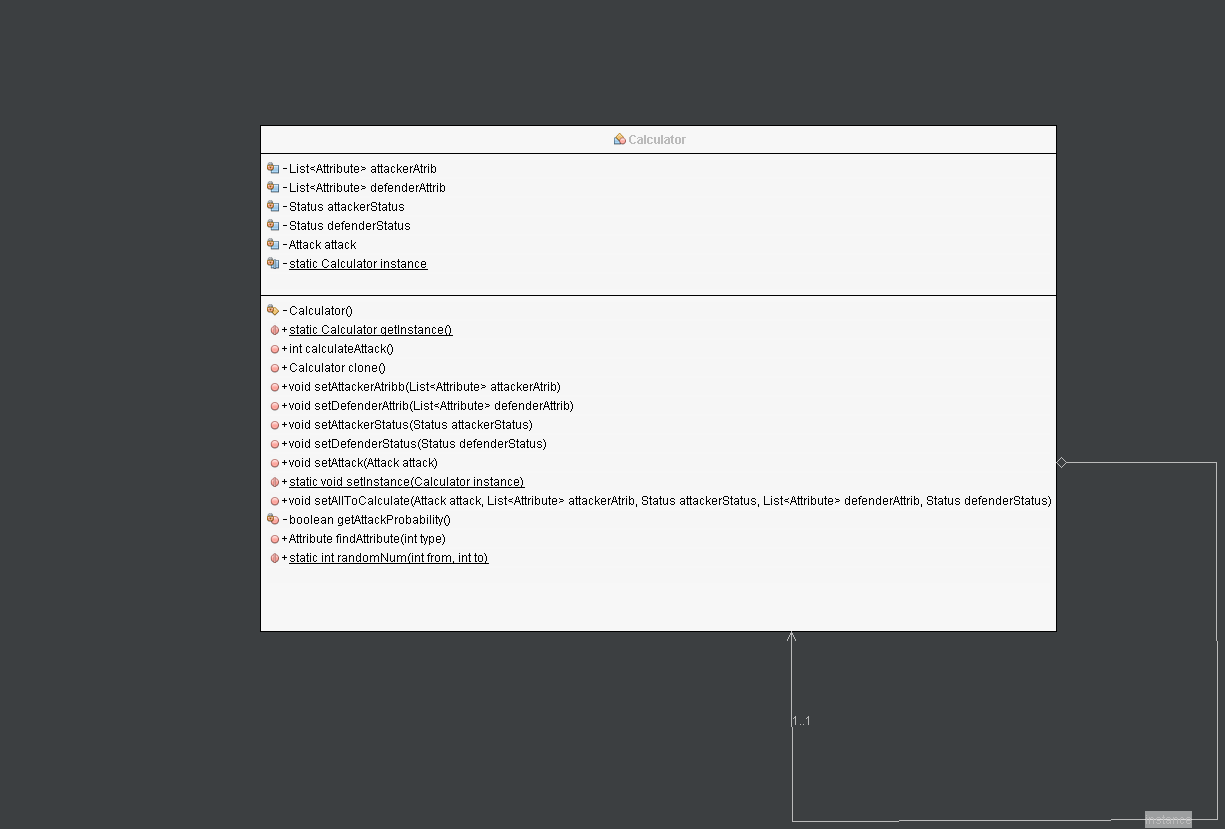


## Abstract Factory

Se ha utilizado para la creación de los enemigos en los diferentes mapas. Dependiendo de que enemigo y en que mapa estuviese se le establecen diferentes atributos que hacen a ese personaje o más fuerte o débil. Se han definido dos enemigos: dragón y orco y dos mapas: Castle (castillo) y Meadow (pradera).



## Singleton

Realiza los cálculos del combate. Se han realizado varias funciones a las que antes de llamarlas les pasas los atributos y estados del atacante y defensor y mediante varios algoritmos hace ul calculo del daño que se le ha hecho al enemigo. Tiene en cuenta: los atributos, estados, mejoras… Para que el ataque sea diferente en cada uno. Por otro lado, también tiene una probabilidad de que el ataque o la cura falle para que no siempre se pueda realizar de manera correcta.

## Template method