

# Programación Avanzada

## Trabajo Practico N° 3

### Programación Orientada a Objetos

2° cuatrimestre 2017 (ambos turnos)

DIIT - Universidad Nacional de La Matanza

## 1. Introducción

Vamos a modelar y desarrollar las distintas unidades de un juego de estrategia.

## 2. Especificación

### 2.1. Unidades

- Los soldados pueden atacar cuerpo a cuerpo a otras unidades si tienen suficiente energía. Cada ataque les consume 10 puntos de energía, y comienzan con 100. Restauran energía si reciben la poción de agua. Infringen un daño de 10 puntos y comienzan con 200 de salud.
- Los arqueros pueden atacar a una distancia de entre 2 y 5 respecto de su enemigo, y si tienen suficientes flechas. Comienzan con 20 flechas en su carcaj, y pueden recargar si reciben un paquete de seis flechas. Infringen un daño de 5 puntos, y comienzan con 50 de salud.
- Los lanceros pueden atacar a una distancia de 1 a 3, sin condición. Infringen un daño de 25 puntos, y comienzan con 150 de salud.
- Los caballeros pueden atacar a una distancia de 1 a 2, siempre que su caballo no se haya puesto rebelde. Infringe un daño de 50 puntos y comienza con 200 de salud. Un caballo se pone rebelde luego de 3 ataques, y puede calmarse si recibe una poción de agua.

### 2.2. Items

Las unidades de este juego de estrategia pueden equiparse con uno o más items. Los mismos no son excluyentes, serán:

- Un escudo, que aumenta la defensa ante ataques. Quien posea un escudo recibirá solamente el 40 % del valor del ataque.
- Una capa, que duplica la energía de la unidad. Sin embargo, reduce la fuerza de los ataques en un 10 % porque obstaculiza su movimiento.
- Un puñal, que aumenta la fuerza de los ataques en 3 puntos pero reduce la defensa también en 3 puntos por inutilizar la otra mano.

## 3. Sobre la resolución

Se pide realizar un buen diseño con objetos. Asimismo, se solicita que utilicen los patrones de diseño *Template Method* y *Decorator*.

## 4. Entrega

Se deberán entregar los siguientes elementos:

1. Diagrama de clases
2. Complejidad computacional
3. Javadocs
4. Casos de prueba exhaustivos con JUnit
5. Métricas de desarrollo en Loom