

Dept. d'Informàtica i Telecomunicacions	Curs 2018 - 2019
Grup: DAM2T1	
M03 Programació	
UF5 Pràctica 1	
Nom Professor/a: Gonzalo Blanca Bonilla	
Data 15/10/2018	

Práctica 1: Herencia StucomRoyal

Un grupo de alumnos de Stucom ha decidido hacer una aplicación para simular un juego de estrategia al que le gusta jugar mucho. Es un juego donde cada jugador tiene una serie de cartas que representan diferentes tipos de ataques o defensas que permiten “luchar” en batallas con otros jugadores.

Cada jugador tiene nombre de usuario, password y número de trofeos conseguidos. Además dispone de un grupo de cartas que ha ido consiguiendo durante el juego y que podrá utilizar en sus batallas.

Todas las cartas tienen un nombre, un nivel de ataque, un nivel de defensa, un coste de elixir (que estará entre 1 y 5) y un nivel de vida (entre 1 y 100). Cada tipo de carta atacará y se defenderá en combate de forma propia.

Las cartas de tipo “tropa”, tendrán un nivel de fuerza que estará entre 1 y 5. Cuando una carta tropa ataque, su nivel de ataque se multiplicará por su nivel de fuerza. No tienen nada especial para su defensa.

Las cartas tipo “estructura”, tendrán un nivel de escudo, que estará entre 1 y 5. Cuando una carta de tipo estructura es atacada, su nivel de defensa se multiplicará por su nivel de escudo. No tienen nada especial para su ataque.

Las cartas de tipo “hechizo”, tendrán un nivel de alcance (entre 5 y 10). Cuando una carta hechizo es atacada, su nivel de alcance se le sumará a su nivel de defensa. En el ataque se sumará a su nivel de ataque el alcance.

Crea las clases necesarias para poder guardar los datos de los jugadores y las cartas.

1. En el método principal de la aplicación, crea al menos 3 cartas de cada tipo (tropa, estructura y hechizo) y guárdalas como las cartas disponibles para el juego.
2. Crea al menos 4 jugadores. Inicialmente su nº de trofeos debe ser 0 y no deben tener ninguna carta disponible.
3. Crea un menú principal con las siguientes opciones:
 - a. Conseguir cartas. Se pedirá usuario y password del jugador. Si los datos no son correctos se dará el correspondiente mensaje de error y se le mandará al menú principal. Si el usuario se autentifica correctamente se le mostrará el listado de cartas y se le permitirá añadir cartas a su listado de cartas disponibles. Cada vez que escoja una carta se comprobará que no la tenga. Si ya la tiene se le informará de que ya dispone de dicha carta. Si no la tiene se le incorporará a su grupo de cartas disponibles. En cualquier caso, se le dará opción al usuario a obtener más cartas siempre que no haya llegado al máximo permitido. Un jugador puede tener como máximo 6 cartas disponibles. Cuando se tenga lleno su cupo de cartas se le informará de ello, se le mostrarán todas sus cartas y se le mandará al menú principal.

b. Batalla. Deberán autenticarse dos jugadores mediante usuario y password. Una vez autenticados los dos, cada jugador deberá escoger tres cartas para la lucha. La suma del elixir de las cartas que escoja no podrá ser superior a 10. Una vez escogidas las cartas de los dos jugadores empezará la batalla. La batalla corresponde a un ataque con cada carta por cada carta del jugador. Atacará primero el jugador que menos trofeos tenga. En caso de empate se escogerá aleatoriamente quien empiece. En el ataque, se calcula el ataque de la carta que ataca, la defensa de la carta del jugador que recibe el ataque y se restan. El resultado de la resta se le quita a la vida de la carta que ha recibido el ataque, siempre que la resta no dé un resultado negativo, en cuyo caso el ataque habrá sido neutro para la ataque que lo recibe y se deberá informar de ello. Ganará el jugador cuya cartas tengan más vida al finalizar la batalla (se debe sumar la vida de las cartas). Obtendrá 5 trofeos. El jugador perdedor obtendrá 1. La aplicación debe mostrar en cada ataque , el valor del ataque y de defensa de cada carta, así como los datos de las dos cartas al finalizar el ataque. También se mostrará la suma de vidas final y el los datos de los jugadores al finalizar la batalla: nombre y n.º de trofeos, tanto del ganador como el perdedor.

c. Obtener el ranking de jugadores por nº de trofeos. Deberá mostrarse el nombre del jugador y el nº de trofeos que tiene, ordenador de mayor a menor.

Consideraciones

La entrega deberá hacerse del proyecto entero en un fichero .zip con el nombre y primer apellido del alumno. Ejemplo: gonzaloblanca.zip.

Deberán tenerse en cuenta todos los posibles errores que puede haber en cada opción del menú e informar al usuario cuando se produzcan. Es importante el feedback con el usuario.

Criterios de corrección

Si la práctica funciona correctamente y hace lo que se pide la nota es de un 5.

Si se utilizan los mecanismos, librerías y funcionalidades explicados en clase de forma correcta, la nota puede llegar hasta un 7.

Si el código está optimizado y limpio, la nota puede llegar hasta un 8.

Si el código está ordenado y debidamente documentado, la nota puede llegar a un 9. Es imprescindible que se comenten las decisiones de diseño. Los argumentos y retornos de las funciones se deben documentar y justificar.

Si se llega más allá de lo pedido en la práctica, la nota puede llegar a un 10.