

Rocagigante_solucion.pdf



Damaga



Metodología de la Programación



1º Grado en Ingeniería Informática en Tecnologías de la Información



**Escuela Politécnica de Ingeniería de Gijón
Universidad de Oviedo**



Descarga la APP de Wuolah.
Ya disponible para el móvil y la tablet.



Nombre y Apellidos:										
UO:	CRC32:	Firma:								
	<table border="1"> <tr> <td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> </table>									

ESCUELA POLITÉCNICA DE INGENIERÍA DE GIJÓN
METODOLOGÍA DE LA PROGRAMACIÓN
Control Práctico de Mayo. Jueves 28 de Abril de 2016.

Se trata de añadir a la jerarquía del Jeroquest una nueva clase de personajes: la **RocaGigante**. Es un tipo de personaje que tiene una capacidad de ataque y defensa impresionante, en realidad si pilla a un enemigo lo aplasta de modo que el enemigo tiene una probabilidad muy pequeña de sobrevivir; pero si la roca gigante pierde la mitad de sus puntos de cuerpo se puede fracturar de forma espontánea.

Las características y comportamientos de la **RocaGigante** son los siguientes:

1. Inicialmente tiene los siguientes atributos: Ataque=50, Defensa=50, Cuerpo=100, Movimiento= (un valor aleatorio entre 5 y 10).

Debemos crear una subclase Personaje denominada RocaGigante. No necesitamos la constante MOVIMIENTO ya que va a ser variable. En el constructor al invocar al constructor de la superclase indicamos el movimiento inicial como un 4+Dado.tira()

2. Su representación como char es 'R' y en modo gráfico es el icono 'roca.gif' a descargar del CV.

En esta clase la implementación de la interface Ficha devuelve una 'R' en toChar(), y el icono indicado para getImagen() de la interface ElementoGrafico.

3. Atacan a todos sus objetivos posibles, en lugar de atacar a uno solo como hacen los demás. Además atacan a todos los personajes, incluidas otras rocas gigantes.

Sobrescribimos AccionCombatar() de forma que combata con todos los elementos del vector objetivosValidos. Para que pueda atacar a todos los tipos de personajes, sobrescribimos esEnemigo() de forma que devuelva siempre cierto.

4. Son capaces de detener un impacto si al lanzar un dado de defensa sacan 3 o más puntos.

El método defender() es abstracto en Personaje, así que debe ser implementado en RocaGigante, podemos copiar el de cualquier personaje concreto (ej. Goblin) y cambiar el umbral de bloque por un 3 o más.

5. Al final de cada ronda, las rocas gigantes que tengan menos de la mitad de los puntos de cuerpo iniciales se fracturan con una probabilidad del 50%. Si una roca se fractura entonces se queda en el tablero pero a partir de ese momento no hace nada en su turno.

- *Debemos añadir la propiedad booleana RocaGigante.fracturada con su get/set (inicialmente falsa).*
- *Al final de cada ronda habría que recorrer los personajes de la partida, identificar cada pj que sea RocaGigante (con instanceof), comprobar vía downcast que no esté fracturada y que tiene menos cuerpo que la mitad de su*

cuerpo inicial. En ese caso tiramos un dado de 2 caras y si sale 1 activamos su atributo fracturada.

- *Sobrescribimos `RocaGigante.resuelveTurno()` y si su atributo `fracturada` es cierto no hace nada, en otro caso invoca a `super.resuelveTurno()`*

Por otra parte, algunos tipos de personajes se comportan como acróbatas, lo que significa que pueden esquivar los ataques de las rocas gigantes debido a su agilidad.

6. Este comportamiento viene descrito por los siguientes métodos:
 - `int` `getAgilidad()`: consulta la agilidad del acróbata
 - `void` `setAgilidad(int a)`: fija la agilidad del acróbata
 - `boolean` `esquivar(RocaGigante r)`: cierto si consigue esquivar la roca, falso en otro caso

*Creamos una interface **Acrobata** con estos tres métodos, ya que va afectar a distintos tipos de personajes sin necesariamente relación jerárquica entre todos ellos.*

7. Se consideran acróbatas a los héroes y a los goblins
 - El valor de la agilidad en los goblins es 2 y en los héroes 5.
 - Los goblins y enanos esquivan a una roca que intenta aplastarlos si en 1D10 (una tirada de un dado de 10 caras) sacan un valor igual o menor que su agilidad.
 - Los bárbaros esquivan si su agilidad es mayor o igual que el movimiento de la roca.

*Las clases **Heroe** y **Goblin** implementan la interface **Acrobata**. **Heroe** fija su agilidad a 5 en su constructor, y **Goblin** a 2, con `setAgilidad()`.*

`Goblin.esquivar()` se implementa comprobando `Dado.tira(10) <= this.getAgilidad()`

*La operación `esquivar()` no se define en **Heroe** sino en **Barbaro** y **Enano** ya es diferente en cada caso.*

*`Enano.esquivar()`: su código es como el del **Goblin***

`Barbaro.esquivar()`: su código será `r.getMovimiento() <= this.getAgilidad()`

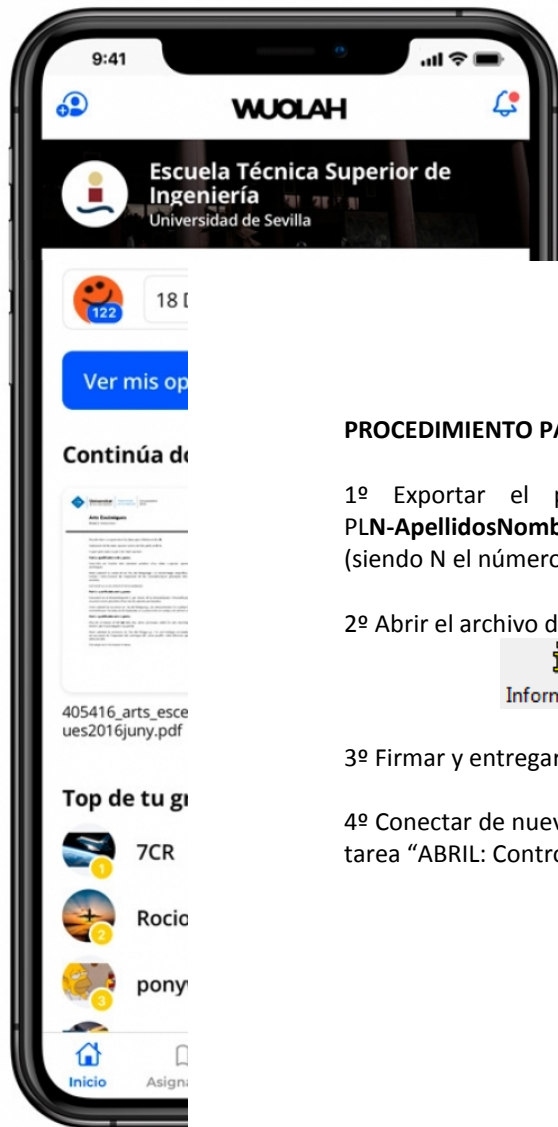
8. Cuando una **RocaGigante** ataca a un enemigo que es acróbata, éste trata de esquivarla. Si tiene éxito, el acróbata no sufre daño, en caso contrario recibe el ataque correspondiente.

En `RocaGigante.AccionCombatir()` si comprobamos que (`objetivo instanceof Acrobata`) permitimos que el objetivo intente esquivar vía casting a la interface: (`(Acrobata)objetivo.esquivar(this)`). Si lo consigue entonces la roca no le ataca.

Finalmente, hay que cambiar la estrategia del juego de modo que

9. En cada ronda, los personajes actúan en orden de mayor a menor capacidad de movimiento actual.

Creamos el método privado auxiliar `Jeroquest.OrdenaPersonajes(Personaje[] pjs)` que ordena los personajes de acuerdo a los puntos de movimiento que tienen de mayor a menor. En `Jeroquest.resuelveRonda()` en primer lugar invocamos al método `ordenaPersonajes()` con el argumento `partidaActual.getPersonajes()`.



Descarga la APP de Wuolah.

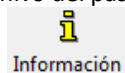
Ya disponible para el móvil y la tablet.



PROCEDIMIENTO PARA ENTREGAR EL EXAMEN:

1º Exportar el proyecto a disco. El nombre del fichero a subir debe ser: **PLN-ApellidosNombre.{zip/7z}**
(siendo N el número de grupo de prácticas del alumno).

2º Abrir el archivo del paso 1 y copiar el CRC en la hoja de examen.



En 7zip pulsa el icono Información para obtener ese dato

3º Firmar y entregar la hoja al profesor.

4º Conectar de nuevo a la red el ordenador y subir el archivo al Campus Virtual a través de la tarea "ABRIL: Control práctico" y enviar para calificar.