METODOLOGÍA DE LA PROGRAMACIÓN

Guion de prácticas de laboratorio

Profesores: Jorge Puente, Ramiro Varela, Juan Luis Mateo, Alfredo Alguero, Eugenia Díaz

TEMA 2. Programación Orientada a Objetos. POLIMORFISMO (SEGUNDA PARTE)

- Jerarquía de clases de la versión JEROQUEST 2.0

PRACTICA 2.5. JEROQUEST 2.0

Objetivo: Aplicar todos nuestros conocimientos de Herencia y Polimorfismo en una aplicación no trivial: el JEROQUEST

Ejercicio 2.5.1. Estudiar la versión (un poco incompleta) del JEROQUEST

a) Importar el proyecto JEROQUEST1.5_INCOMPLETE

b) Observa los diagramas de clases UML a través de los ficheros .gaphor que ya están construidos.

c) Accede a la documentación de la aplicación, generada con Javadoc, a través de la subcarpeta doc /index.html. Elimina esta documentación y genérala otra vez con la opción Project/Generate Javadoc …

d) Observa la implementación de algunos métodos. Por ejemplo, la del método **`opponetsLeft`** (quedan contrincantes vivos) de la clase **`Controller`**. Estudia bien la implementación de este método e indica mediante comentarios en el código qué tipo de esquemas de programa se utilizan.

Ejercicio 2.5.2.- Esta es la parte más dura: se trata de jugar un poco (aunque el juego no está completo, porque algunos métodos no responden a las especificaciones)

a) Ejecuta el programa tal y como está y observa la salida de texto y la salida gráfica. Lo que se muestra es el estado resultante de cada acción (de momento sólo moverse) que realizan los personajes en su turno. Una vez resuelto el turno de todos los personajes acaba la ronda actual y empieza una nueva ronda.

b) Cambia los parámetros para que se creen más personajes, se cambie el tamaño del tablero, y se juegue un total de rondas inferior a 20. Juega unas cuantas veces (no muchas) y relaciona bien las salidas gráficas y de texto.

Ejercicio 2.5.3.- Ahora se trata de completar los métodos que están incompletos o que son incorrectos

a) Habrás observado que los personajes se mueven y se apelotonan en la esquina inferior izquierda del tablero. Esto es debido a que los métodos que calculan las posiciones al norte y al este de otra posición funcional mal. Corrígelo para que los personajes se muevan en todas las direcciones.

b) Habrás observado también que los personajes no se atacan. El motivo es que el método **`validTargets`** (objetivos válidos) siempre devuelve un vector de personajes vacío. Debes corregirlo para que devuelva siempre un vector con los

enemigos vivos y que estén en una casilla adyacente ortogonalmente con la del personaje que ataca.

Ejercicio 2.5.4.- Y ahora se trata de cambiar algunas cosas:

a) Cambia la forma de elegir contrincante de modo que cada personaje elija siempre al personaje del otro bando que tenga más puntos de cuerpo, y que esté a tiro, por supuesto.

b) La representación del vector de personajes. En lugar de un vector estático uno dinámico **`DynamicVectorCharacters`**. Modifica el código de manera que los personajes muertos desaparezcan del vector y haz que todos los métodos tengan en cuenta este hecho.

c) Cambia la estrategia de juego de cada ronda de modo que los personajes no jueguen siempre en el mismo orden. Para ello lo que hay que hacer es "barajar" el vector de personajes antes de cada ronda. Una forma de barajar un vector es recorrerlo y para cada posición i elegir otra posición j aleatoriamente e intercambiar los contenidos de las posiciones i y j.

d) Cambia la estrategia de juego de los héroes de la siguiente forma: en lugar de atacar al principio y luego moverse, lo que harán es intentar atacar, si no pueden se moverán una posición y una vez allí intentarán atacar otra vez, si no lo consiguen seguirán moviéndose y así sucesivamente. Si consiguen atacar una vez, entonces ya no lo intentarán más veces y simplemente se moverán las veces que les queden. Así cada héroe solo podrá atacar una vez como mucho en cada ronda.

e) Añadir al proyecto una clase **`Statistics`** para controlar el número de ataques y el daño producido por cada uno de los bandos.

DISTRIBUCIÓN TEMPORAL

Sesión 14. PRÁCTICA 2.5. Ejercicios 2.5.1, 2.5.2 Y 2.5.3.

Sesión 15. PRÁCTICA 2.5. Ejercicio 2.5.4

Sesión 16. PRÁCTICA 2.5. Ejercicio 2.5.4

Sesión 17. PRÁCTICA 2.5. Ejercicio 2.5.4