**FIUBA - 75.07**

**Algoritmos y programación III**

*Trabajo práctico 2: AlgoFormers*

1er cuatrimestre, 2016

Alumnos:

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Padrón** | **Mail** |
| Carbón, Joaquín | 97.537 | joakoracing@hotmail.com |
| García Guarrochena, Carlos Ignacio | 87.299 | cignaciogarcia@gmail.com |
| La Penna, Mariano | 98.432 | mlapenna79@hotmail.com |

***Fecha de entrega final***: Jueves 23/06/2016 - Lunes 27/06/2016

***Tutor***: Gabriel Devoto

***Comentarios***:

4ta y última Entrega:

Trabajo Práctico completo funcionando, con interfaz gráfica final, sonidos e informe completo.

**Avance respecto a la entrega previa**

Se agrego la interfaz gráfica completa, con un tablero, los Algoformers de cada jugador y sus respectivas barras (de ataque, vida, distancia de ataque y velocidad) y los botones con las acciones que puede realizar el Algoformer.

En cuanto al modelo, a grandes rasgos, se agrego la clase Modo la cuál administra las diferencias para movimiento y ataque según el modo del algoformer (humanoide o alterno). Algoformer dejó de conocer a Tablero y Movimiento. También se agregó la clase Turno que administra lo que su nombre sugiere, a la cual conoce el Algoformer. Por último, se implemento la lógica de la victoria por obtener la chispa o matar a todos los Algoformers rivales.

Se agregaron nuevos tests para probar las modalidades y clases nuevas y se modificaron los de las entregas pasadas para que acepten estos cambios en el modelo (por ejemplo constructores a los que se les pasan parámetros de nuevas clases).

Algunas modificaciones relevantes:

* Turno
* Excepciones nuevas para las modalidades agregadas.
* Modo: dividida en alterno, combinado y humanoide.
* Vistas: (las más importantes)

**-TableroVista:** la vista tanto del Tablero como de las barras de los Algoformers.

**-CasilleroVista**: setea las superficies y es la Vista de cada Casillero.

**-AlgoformerVista:** da las imágenes a cada Algoformer y es la vista de los mismos.

**-** **ChispaSupremaVista**: lo que su nombre indica.

**-JuegoVista:** Arma la ventana con los botones, contiene la lógica de crear el juego y llama a las otras vistas.

-**BarraJugadorVista:** los paneles con la información de cada robot.

-**MostrarAlUsuarioError**: lee de cada excepción el título y contenido a mostrar y la muestra mediante AlertHandler.

* Controladores:

**-MouseClickHandler**: contiene la lógica de la interacción del jugador y el juego mediante el “click” en el tablero (no los botones).

-**Juego:** Contiene el main.

**-AtacarButtonHandler**, **CambiarMapaButtonHandler**, **CombinarButtonHandler**, **MoverButtonHandler**, **TransformarButtonHandler:** responden al evento clic de cada uno de estos cinco botones en el juego.

**-AlertHandler**: crea las ventanas javafx Alert.

**Supuestos**

* Cuando se crea un Algoformer se crea en forma Humanoide
* Se carga el mapa ya creado en JSON
* Se considera que hay dos jugadores y ninguno más.
* Los movimientos que se pueden realizar son horizontales, verticales y diagonales puros, igual a la “dama” o “reina” en ajedrez (pero dentro de la velocidad máxima). No se permiten mezclas de ambos.
* Cuando las espinas reducen la vida de una unidad que lo atraviesa, definimos que el 5 por ciento se multiplique por cantidad de casilleros atravesados. O sea si por ejemplo una unidad terrestre con 100 de vida avanza desde (0, 0) hasta (3, 0) siendo los casilleros (1,0), (2,0) y (3,0) de superficie espinosa, la unidad termina el recorrido con 85 de vida.
* Al pasar una unidad terrestre por al menos un casillero con pantano, se reduce el recorrido a la mitad de distancia independientemente de la cantidad de pantanos existentes en el recorrido. Por ejemplo, si se realiza un movimiento de 4 casilleros en el que el primero es pantano y los otros no, se moverá dos casilleros.
* Al pasar por un pantano con una cantidad de casilleros es impar (por ejemplo 5), la cantidad de casilleros que se desplazará (por la condición de que cueste el doble atravesarlo) se redondea para abajo (se mueve 2 casilleros).
* Para combinarse, los algoformers deben estar alineados verticalmente y sin espacio entre ellos, o sea, uno al lado de otro en la misma columna.
* Los algoformers combinados no podrán descombinarse (ni transformarse).

**Desarrollo**

Para realizar el juego utilizando el modelo orientado a objetos, creamos las clases necesarias que interactúan tanto como es necesario para el buen funcionamiento y fácil comprensión, mínima repetición, etc.

La clase principal es Algoformer, esta es abstacta y de ella heredan dos clases que son Autobot y Decepticon, también abstractas, de las cuales heredaran las clases de cada algoformer en particular, donde se definirá todo lo necesario para diferenciarlos y darles sus características.

Para que el algoformer se mueva y la superficie que haya lo afecte, se utiliza la clase Movimiento, pero el algoformer no la conoce, solo conoce a la clase Modo, la cual tiene un atributo de clase Movimiento que se diferencia también en 3 hijas.

Para la base del juego se utiliza la clase Tablero, la cual maneja un ArrayList de ArrayList<Casillero>, siendo Casillero la clase elemental que contiene las superficies aérea y terrestre (cada superficie es una clase que sabe cómo afectar al algoformer que pasa por ella) y en el Casillero se alberga un “Contenido”, este contenido es una clase de la que heredan Algoformer, Vacio y Chispa. El casillero devuelve su contenido siendo hoy uno de esos 3 tipos posibles.

Los tableros se guardan en un archivo JSON que es leído e interpretado al iniciar el juego.

Por último, las clases Juego y Jugador son las que crean el juego (“armar” el tablero, crear los algoformers, etc.) y lo organizan.

Otras clases no principales son:

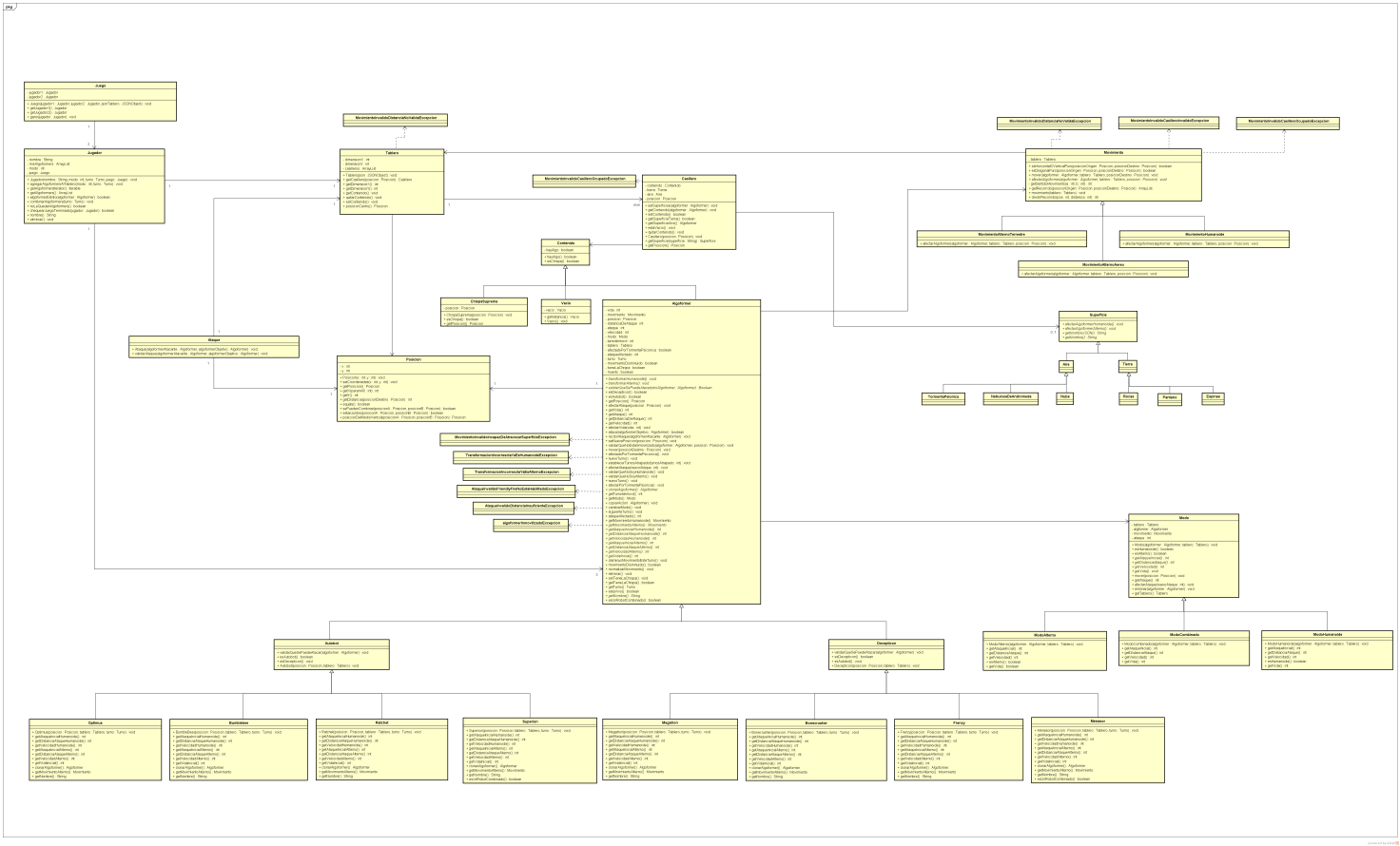
* Ataque: clase mediante la cual un algoformer ataca a otro.
* Posición: lógica de posicionamiento en el tablero.
* Turno: organiza los turnos y avisa cuándo termina el juego.
* 14 Excepciones más una de la que heredan todas.

Se utilizó el patrón MVC para incorporar la interfaz gráfica. Ya fue mencionado cada componente de vista y controlador en detalle previamente.

DIAGRAMAS

Se pueden ver también desde el archivo .asta disponible en ela misma carpeta que este documento, o bien hacer zoom acá para ver detalle.

Clases



Diagramas de Secuencia

