

PRÁCTICA ROBOCODE OPCIONAL

REALIZADO POR:

- Rocío Márquez Molero
- Marta López Pérez

CLASES IMPLEMENTADAS

- Problema

En esta clase se genera en unas coordenadas al azar donde nuestro robot aparecerá, al igual que ocurre con el robot objetivo y los obstáculos.

Funciones a destacar:

- Problema (constructor): dado un número de Filas (nF), columnas (nC), obstáculos (nO) y una semilla (s), se creará una matriz de nF x nC que representará el tablero. A partir de la semilla, se generará un número aleatorio (clase Random) con el que se irá recorriendo tantos valores como sean necesarios para que todos los obstáculos, nuestro robot y su objetivo tengan una posición diferente entre ellos.

- Estado

Esta clase como bien su nombre indica, guarda la información de donde está un robot en un determinado momento mediante la fila, la columna y el problema que se quiere resolver.

Funciones a destacar:

- sePuedeMover: esta función devuelve true si se puede transitar de un estado a otro sucesor o false si no se puede.

- heurístico: es una función que calcula el heurístico que se basa en la distancia octil entre el estado en que estamos y el estado final. Esta devuelve el valor.

- finalP: esta función devuelve true si el estado es final o false si no lo es.

- Dato

En esta clase se almacena un estado del problema, el coste real para llegar a él y el paso previo. Implementa la interfaz Comparable para lograr una ordenación correcta.

Funciones a destacar:

- getFN: devuelve el coste estimado en función de la heurística para llegar a la posición final

- Bot3

Esta clase se encarga de controlar el funcionamiento del robot, que se moverá desde una posición inicial hasta una final, si encuentra un camino válido.

Funciones a destacar:

- Funciones de movimiento: estas funciones se encargan de orientar el robot hacia donde se va a mover y lo mueve una casilla. Se implementan 8 movimientos.
- Algoritmo A*: es el algoritmo principal que se encarga de que mientras no se haya acabado lista de sucesores siga intentando llegar a la solución, utilizando como heurístico la distancia octil.

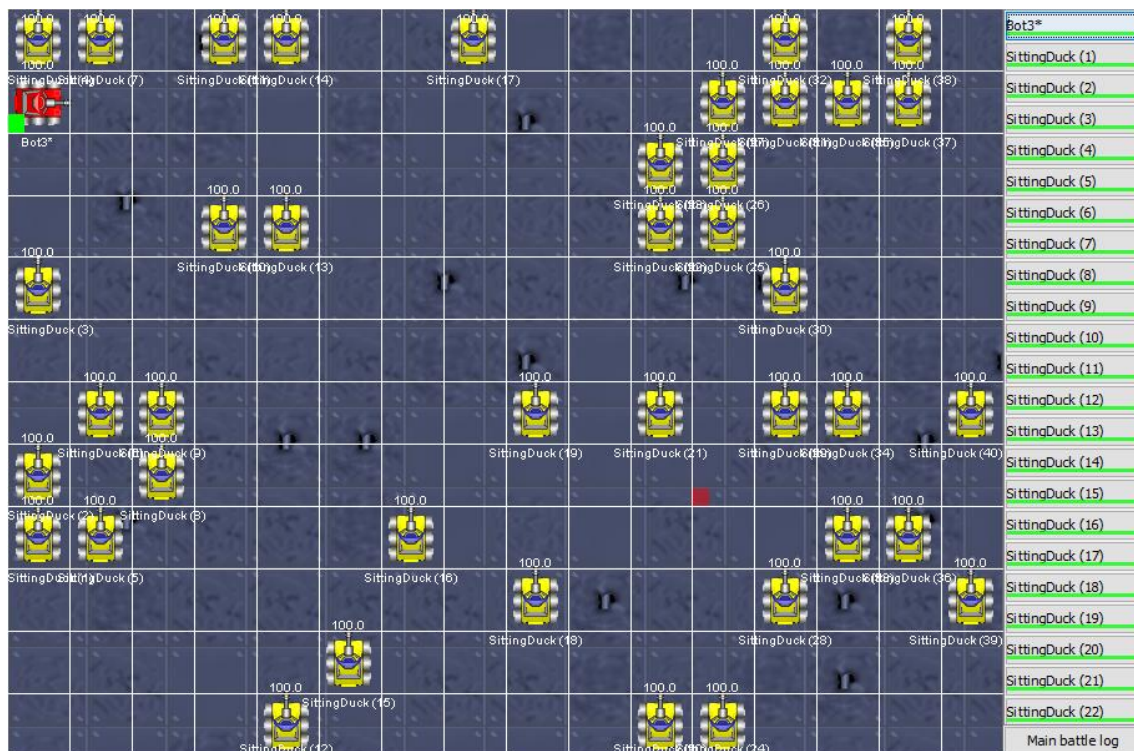
- RouteFinder

Esta es la clase main del proyecto, en ella se inicia Robocode, se genera el problema y se colocan los obstáculos y nuestro robot (bot3).

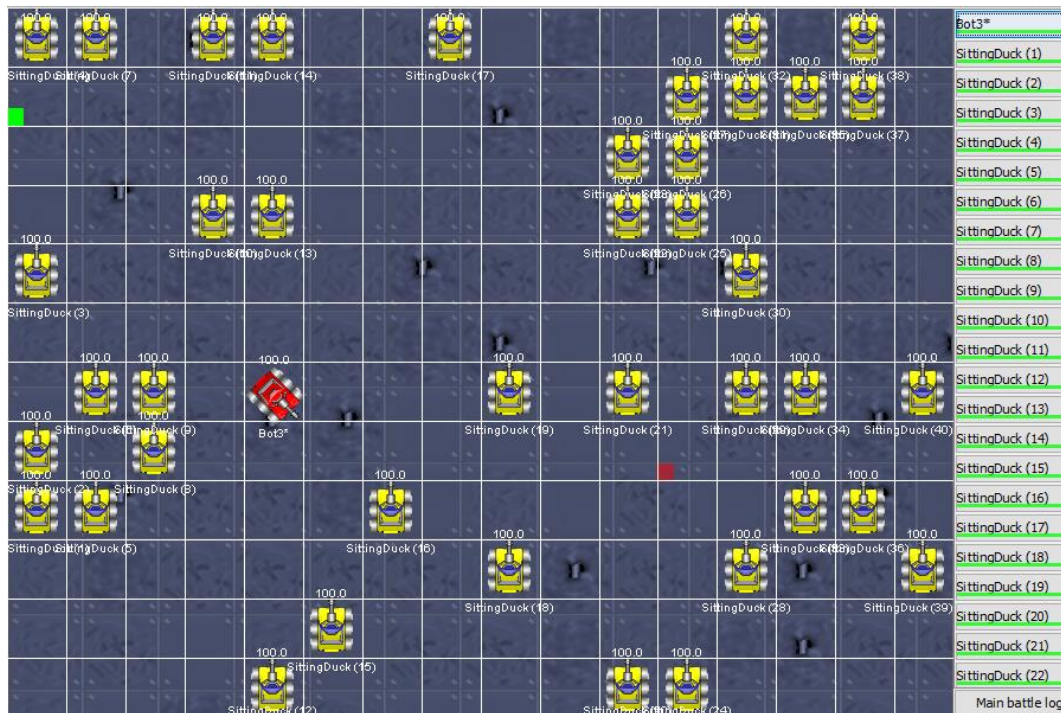
EJECUCIÓN

Para ejecutarlo, se abre eclipse con la ruta donde se ha guardado el proyecto y se compila la clase RouteFinder. Automáticamente se abre la aplicación Robocode con nuestro robot y los respectivos obstáculos. A la derecha si pinchamos en Bot3, se nos abre una ventana de compilación con los pasos que se siguen y si le damos a Paint se pintan las casillas.

El objetivo es llegar a la posición final (**cuadradito rojo**) partiendo desde la posición inicial (**cuadradito verde**) evitando a los otros robots y siguiendo el camino más óptimo.

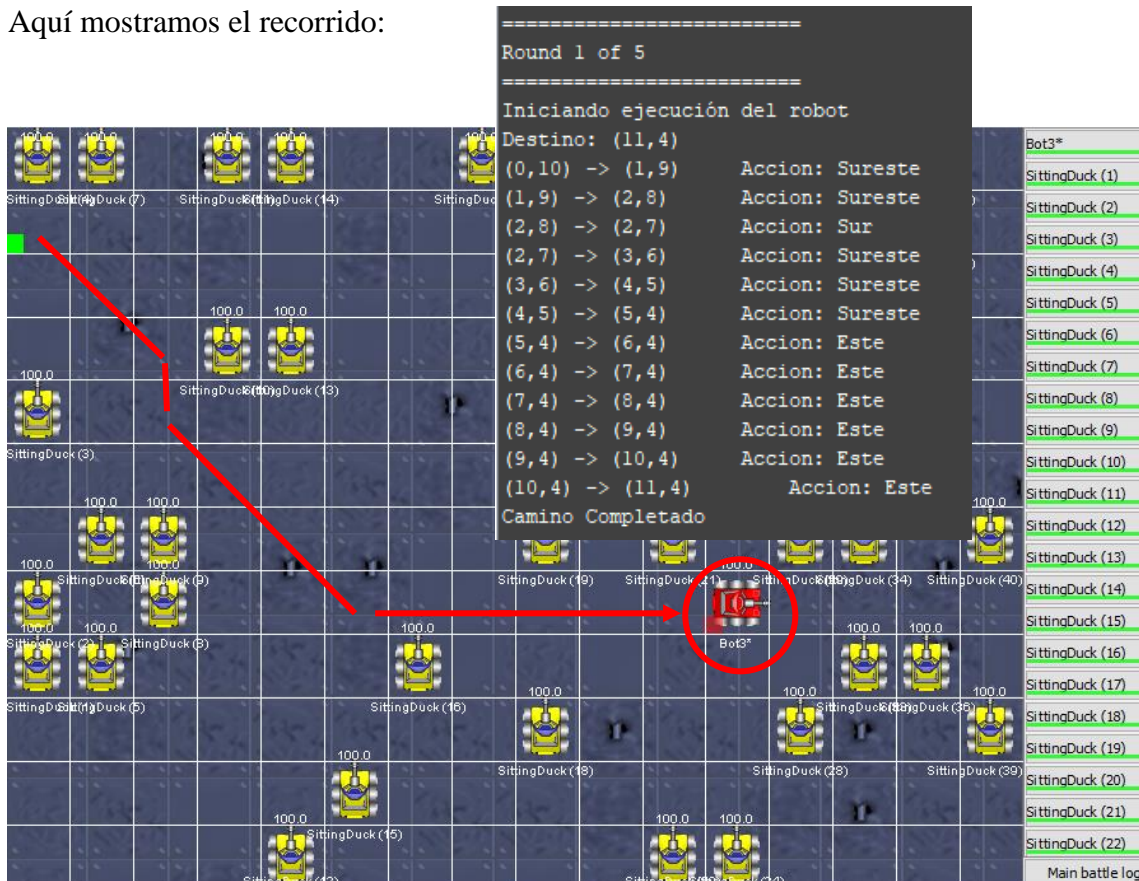


Para tener más movilidad, nuestro robot tiene una amplia movilidad al haberle programado con 8 movimientos, facilitándole moverse no solo en dirección Norte, Sur, Este y Oeste.



Durante todo el camino, en la consola se recogen todos los movimientos que realiza el robot indicando la posición inicial y final además de la orientación que tienen.

Aquí mostramos el recorrido:



CONCLUSION

Este proyecto es muy interesante y entretenido. Al principio puede llegar a ser muy lioso, ya que la instalación, si nunca has trabajado con cosas parecidas anteriormente, es muy nuevo. Es una manera de aprender a programar algo nuevo de una manera muy divertida, y el poder ir viendo las batallas y como progresa tu robot es increíble.

El código en sí nos ha costado un poco en cuanto a las clases que se han añadido a las dadas como plantillas, pero gracias a un poco de ayuda de otros compañeros hemos conseguido que funcione.