

Práctica 2 – Resolución de problemas con búsqueda informada

Introducción

En esta práctica continuamos con el problema de la aspiradora, pero utilizando técnicas de búsqueda informada, que son capaces de trabajar con espacios de búsqueda más grandes.

Instrucciones

Descargar el zip con el esqueleto de la práctica 1 desde el campus virtual. Descomprimir y ejecutar `mvn install`. Ahora descargar el zip con el esqueleto de la práctica 2 desde el campus virtual. Descomprimir y ejecutar los siguientes ejemplos:

- Correr todos los tests con `mvn test`. Observar que se ejecutan sin fallos.
- Ejecutar los tests por separado
 - Búsqueda informada A* con heurística “basura total”: `mvn test -Dtest=AStarSearchHGlobalDirtTest`
 - Búsqueda informada A* con heurística “basura total + distancia a la basura”: `mvn test -Dtest=AStarSearchHGlobalDirtAndDistanceTest`
- Ejecutar el programa `VacuumApp`, que realiza una simulación sencilla con el `VacuumSearchAgent`, utilizando A* y la heurística “basura total”: `mvn exec:java -Dexec.mainClass=ssii.p2.VacuumApp`

Importar el proyecto en Eclipse. Si Eclipse da fallos con los import de los paquetes `ssii.p1.*`, abrir la perspectiva “Maven Repositories” con `Window -> Show View -> Other` y buscando `Maven -> Maven Repositories`. En esa vista seleccionar `Local Repositories -> Local Repository` y con el botón derecho elegir “Rebuild Index”. Después de eso hacer `Project -> Clean` y el proyecto debería compilarse correctamente con Eclipse. Realizar las siguientes modificaciones del código:

- Compara la efectividad de las dos heurísticas. Para ello emplea tableros más grandes, o con basura en varias casillas.
- Escribir tests adicionales considerando tableros de 10x10, variando la casilla inicial de la que parte la aspiradora, e incluyendo más casillas con suciedad. Los tests deben comprobar que no queda suciedad después de ejecutar la solución.
- Diseña una nueva heurística. Por ejemplo investiga la [distancia de Manhattan](#). Añade tests para compararla con las otras heurísticas.
- Investigar como de grande puedes hacer el tablero y seguir encontrando una solución en menos de 30 segundos, con las distintas heurísticas.

Presentación de la práctica

Subir a la entrega en el campus virtual un zip con la carpeta del proyecto (la que contiene el archivo pom.xml). Solo hace falta que uno de los miembros del grupo realice la entrega en el campus virtual. La práctica deberá estar modificada como sigue:

- El proyecto se puede importar en Eclipse correctamente.
- Se han escrito tests adicionales considerando tableros de 10x10, variando la casilla inicial de la que parte la aspiradora, e incluyendo más casillas con suciedad. Todos los tests pasan.
- Se ha definido una nueva heurística y añadido tests usando la nueva heurística.

También añade un archivo preguntas.txt al zip que se entrega en el campus virtual, con respuestas a las siguientes preguntas.

1. De todos los algoritmos vistos en esta y la práctica anterior, ¿cuál funciona mejor? ¿Por qué?
2. En la búsqueda informada, ¿como valorarías el efecto de cada heurística? ¿Varía mucho o poco? Justifica la respuesta
3. ¿Cuál de las heurísticas encuentra la solución más rápidamente?
4. ¿Qué heurísticas son admisibles y cuales no lo son? Justifica tu respuesta
5. ¿Cómo de grande puedes hacer la matriz? ¿Depende de cuentas basuras añadidas? ¿Depende de la posición de la basuras?

Para que quede claro que respuesta corresponde a cada pregunta en el archivo preguntas.txt, copia en dicho archivo la lista de preguntas y añade cada respuesta a continuación de cada pregunta.