

IA – Curso 2016-2017

Práctica 2

Representación y búsqueda en la librería AIMA

El objetivo de esta práctica es conocer las técnicas de representación de problemas y los algoritmos de búsqueda disponibles en la librería Java asociada al libro AIMA (Artificial Intelligence – A Modern Approach 3rd Edition).

La entrega de la práctica 2 se realizará a través del campus virtual en un fichero zip que tendrá como nombre el número de grupo seguido por el primer apellido de cada uno de los integrantes del grupo (NumGrupoApellidoAlumno1ApellidoAlumno2.zip).

Instalación

- Descarga el archivo **aima-java-AIMA3e.zip** disponible en el Campus Virtual.
- Si descomprimes el zip aparecen 4 directorios:
 - aima-core: proyecto que contiene los algoritmos implementados en la 3ª edición.
 - aima-gui: proyecto que contiene demos de aplicaciones desarrolladas utilizando aima-core.
 - aimax-osm: proyecto que contiene aplicaciones y librerías relacionadas con Open Street Map (OSM). Framework para desarrollar sistemas de navegación.
 - aima-all: para crear releases del resto de proyectos.
- Nota: En el Campus Virtual también puedes encontrar la versión es **aima-java-1.7.0-Chp4-Complete.zip**. No tiene el directorio aima-all sino release, donde ya están los jar generados.

Ejecución de la práctica

- Importa en Eclipse los proyectos aima-core, aima-gui y aimax-osm (puedes seguir las instrucciones para Eclipse en <https://github.com/aima-java/aima-java/wiki/AIMA3e-Using-Eclipse-IDE>).

1) Ejecuta AimaDemoApp (aima.gui.applications). Observa cómo funciona en modo aplicación (ejecución completa o paso a paso) y en modo demo, seleccionando distintos problemas, estados iniciales y algoritmos.

2) Ejecuta AimaDemoApp en modo aplicación para el problema del puzzle de 8 con el ejemplo extremo, probando la búsqueda en anchura, el método voraz y el A* con diferentes heurísticas. Haz una tabla con los resultados para cada algoritmo e incorpórala a la memoria. Interpreta y discute los resultados que has obtenido.

- 3) Ejecuta AimaDemoApp en modo aplicación para la búsqueda de caminos en el mapa de Rumanía, desde Arad a Bucharest, utilizando la búsqueda en profundidad, en anchura y el algoritmo A* con la heurística de la distancia en línea recta. Haz una tabla con los resultados obtenidos con cada algoritmo e incorpórala a la memoria. Interpreta y discute los resultados. ¿Por qué el algoritmo de búsqueda en anchura no obtiene el camino mínimo?
- 4) Ejecuta SearchDemoOsmAgentApp (aimax.osm.aplications) para buscar el camino entre Altheim y Leibi, utilizando los algoritmos de búsqueda en profundidad, coste uniforme y A* con la heurística de la distancia en línea recta. Haz una tabla con los resultados para cada algoritmo e incorpórala a la memoria. Interpreta y discute los resultados que has obtenido.
- 5) Explora el código de la librería y localiza la definición del estado y de los operadores en el problema del puzle de 8. Indica en la memoria de este apartado la localización de dichas definiciones. Indica también cómo está representado el estado del puzle de 8. ¿En qué paquete y clase se implementa la heurística del número de fichas mal colocadas? Indícalo en la memoria.
- 6) Explora el código de la librería y localiza la definición del estado y de los operadores en el problema de la búsqueda de caminos en el mapa de Rumanía (citado en el apartado 3). Indica en la memoria la localización de dichas definiciones. Indica también cómo están representados el estado y los operadores.

Fecha límite de entrega: 6/11/2016 23:55