IEC9818 SISTEMAS INTELIGENTES

Implementación de búsqueda – Algoritmo A\*

Fecha de entrega: 25 de Octubre del 2017

Problema: Desarrollar un programa que implemente el Algoritmo A\* para resolver el siguiente problema:

Está basado en el juego “atomix”, en el que la finalidad es ensamblar moléculas a partir de distintos átomos. El juego se desarrolla en un *grid* 2D, no necesariamente cuadrado, donde ciertas posiciones están bloqueadas, también hay paredes que delimitan el área de trabajo y el resto de los espacios son libres. En la posición inicial, todos los átomos necesarios para construir la molécula se encuentran dispersos en el *grid*, cada átomo ocupando alguna celda. A partir de esa posición, es posible realizar movimientos sobre cada átomo, en forma independiente, para irlos juntando y formar la configuración exacta que se pide, o sea la formación de la molécula. El modelo de molécula a construir se encuentra visible en todo momento a un lado del *grid*. En algunas versiones de este juego, se trabaja con un límite de tiempo y se van resolviendo distintos niveles, cada uno de mayor complejidad. Cada nivel representa el ensamble de una molécula diferente.

Los movimientos están restringidos a desplazamientos horizontales y verticales (no en diagonal). También se ven restringidos por la presencia de los obstáculos del *grid*, las paredes, o incluso de otros átomos que se encuentren en su trayectoria.

Para tener una idea más concreta del aspecto, objetivo y grado de dificultad de este juego, se puede visitar la siguiente página, la cual implementa una versión llamada “katomic”, la cual es un clon del “atomix” original (para ejecutarlo es necesario que el navegador pueda desplegar applets de java)

<http://www.heni-online.de/java/atomicplay_js.php>

Manual disponible en: <http://docs.kde.org/stable/en/kdegames/katomic/index.html>

Si se tienen problemas con el navegador, se puede ejecutar una versión Windows, hay una modalidad de descargar un .zip y desde ahí correr el ejecutable (sin instalar nada). Para esto visitar el sitio: <http://watomic.sourceforge.net/> Este .zip también se encuentra disponible en el Aula Virtual.

**Consideraciones a tomar en cuenta entre el juego original y este proyecto con el fin de simplificar la solución:** En este proyecto se definirá anticipadamente la ubicación final de la molécula dentro del *grid*, cosa que no ocurre en la versión original del juego. También, otra simplificación consistirá en que cada átomo que formará la molécula es diferente a los demás. En la versión original del juego hay casos (no tan frecuente) en los que la molécula utiliza varias veces un mismo átomo.

Criterios de solución e implementación:

1) El programa puede realizarse en cualquier lenguaje de programación, sin embargo, el ejecutable debe poder correrse en Windows. Independientemente del lenguaje, el ejecutable entregado o en su caso el “instalador” de la aplicación debe incluir todo lo necesario para poder ejecutar la aplicación de manera que no indique errores de *runtime* por falta de librerías, componentes, etc.

2) La solución propuesta **necesariamente** debe basarse en la implementación del **algoritmo A\***

3) El programa debe funcionar con el nivel 2 (molécula del metano) de la implementación katomic.

4) La representación interna del problema y las estructuras de datos asociadas al grafo de búsqueda, así como otros factores y variables consideradas para implementar el algoritmo son libres. Cada quien decide que utilizar para este fin.

5) Es requisito mostrar como solución una interfaz visual que muestre exactamente la posición inicial y la final. Si no se desea implementar una animación gráfica con los movimientos de la solución, es posible mostrar solamente la secuencia de movimientos en forma de texto, de alguna manera que no deje duda para cada movimiento, qué átomo está involucrado y hacia qué dirección se mueve. Además de mostrar la secuencia de la solución, el programa debe contabilizar todos los estados que fueron considerados durante el proceso para poder mostrar al final de la ejecución el total de nodos considerados o visitados. **La cantidad de estados visitados es una medida directa de lo bien o mal implementada que fue la heurística utilizada (o lo bien o mal implementado que fue el algoritmo A\* en sí).**

Criterios de calificación

* Se debe trabajar en equipo.
* La solución es estrictamente en equipo, no hay ninguna razón justificable para que dos equipos presenten soluciones semejantes. Es decir, aunque todos los programas se basarán en el algoritmo A\*, los detalles de la codificación y la adaptación del algoritmo para resolver este problema en particular implica que no hay razón para tener códigos semejantes. De no cumplirse lo anterior, los programas involucrados quedarán sin calificación.
* Es necesario que el programa, al ser ejecutado, mande a la pantalla alguna evidencia del proceso que está efectuando, por ejemplo, la información de los nodos que va explorando; y también al final, la ruta o secuencia que muestre paso a paso la solución encontrada.
* Evite que el programa no muestre nada durante el proceso, o que al terminar solo muestre la posición final de la molécula, ya que esto no proporcionará ninguna información del trabajo interno del programa.
* Es necesario entregar en forma impresa la secuencia que muestre los movimientos desde el inicio hasta la solución. Lo anterior no aplicará en el caso en que el programa no haya sido capaz de encontrar una solución o que sea una cantidad excesivamente grande de pasos, que haga impráctica la impresión en papel.
* Es necesario subir al Aula Virtual el código fuente y el ejecutable. (si por el tamaño del ejecutable fuera inapropiado subirlo al Aula Virtual, se puede entregar de una manera alterna).

No es necesario entregar impreso el código fuente.

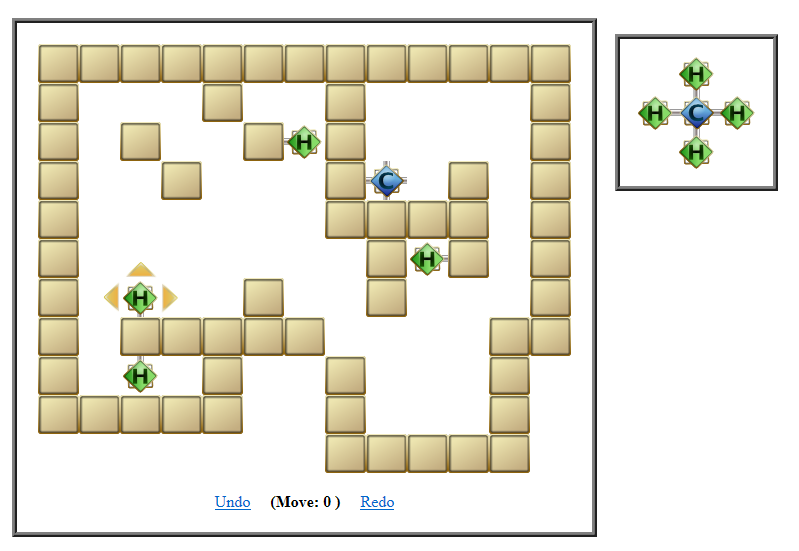
* La calificación será otorgada en dos fases, primero, una calificación igual a todo el equipo en base a los criterios de la rúbrica anexa. Luego, para calcular la calificación final (individual), esta será el 100%, el 50% o el 0% (de la obtenida por el equipo) de acuerdo al nivel de suficiencia mostrado al contestar preguntas específicas.
* Si la solución ofrecida no está basada en un algoritmo de “búsqueda informada” se obtendrá solamente un 3.0, si al menos es resultado de una búsqueda no-informada, o 0.0 si es en base a cualquier otra lógica.
* Para la calificación en equipo se tomará en cuenta: los **resultados obtenidos**, la **documentación entregada** y la **integración** **como equipo**.
* Para la calificación individual se tomará en cuenta: **las respuestas individuales que se den a preguntas específicas el día de la entrega y demostración del proyecto**. Las preguntas irán encaminadas a establecer el dominio del tema y el conocimiento de los detalles de la implementación.
* Mostrar un programa que resuelve el problema no garantiza obtener calificación. Es necesario mostrar suficiencia respecto a que el programa en realidad fue desarrollado por el equipo. Ante una duda razonable acerca de la autoría del programa, se solicitará a los integrantes del equipo explicaciones más detalladas del código y su funcionamiento, quedando sujeta la calificación al resultado de esa explicación.

Nota: cualquier duda o caso no previsto en este documento se resolverá en clase.

**La calificación obtenida será de acuerdo a los siguientes criterios:**



Nivel 2 – Molécula de Metano (Configuración inicial)



Una solución al problema es la siguiente (se puede tomar esta ubicación como la meta para efecto de este proyecto):

