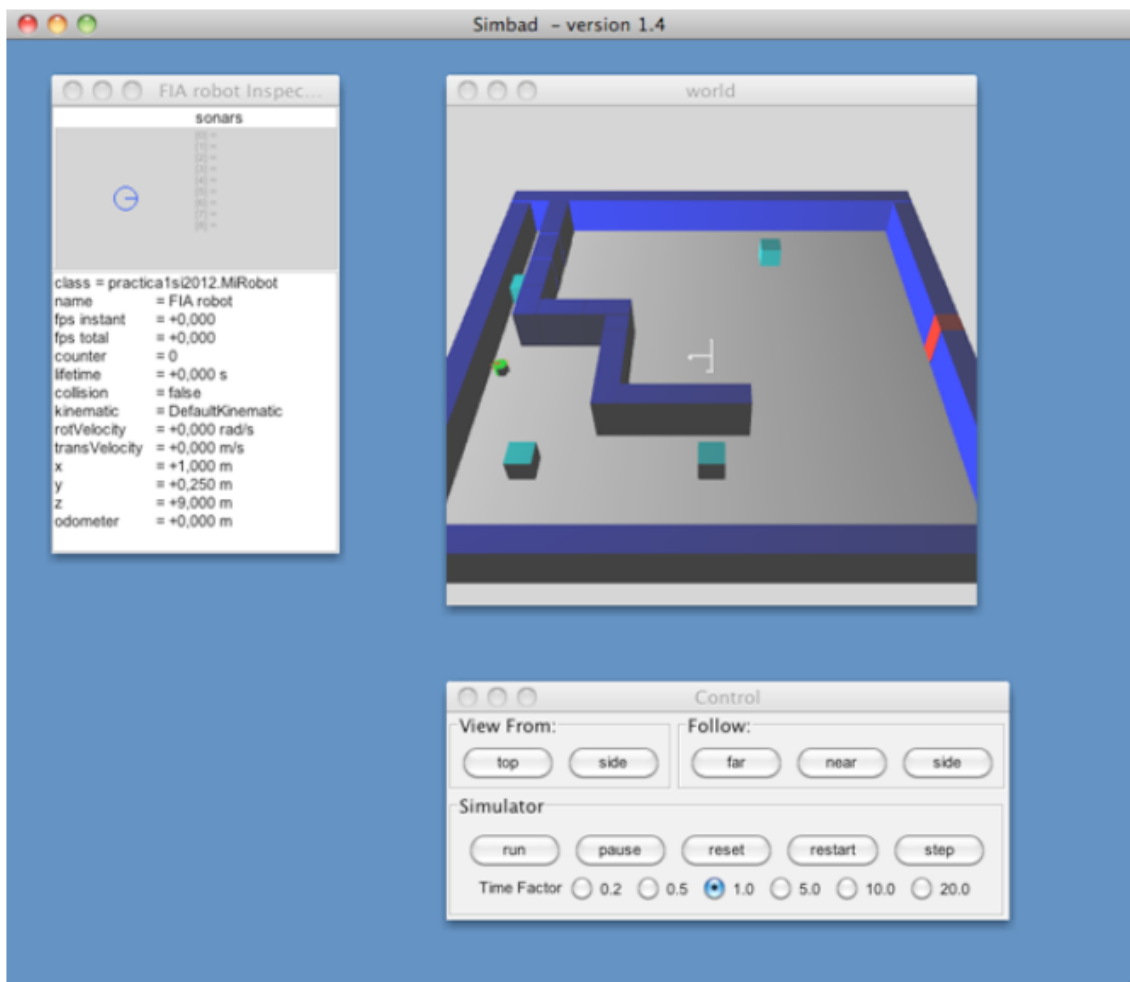


BÚSQUEDA HEURÍSTICA. A*

INTRODUCCIÓN

En esta práctica y en la siguiente se desarrollará un sistema capaz de guiar a un robot en su entorno desde el punto de partida hasta la meta. En primer lugar, en esta práctica, se utilizará el algoritmo A* para calcular el camino óptimo para llegar a la meta y posteriormente, en la siguiente práctica, se utilizará un sistema experto difuso para guiar al robot hasta la meta siguiendo el camino calculado por A*.



ALGORITMO A*

En la implementación del algoritmo A* se ha seguido el algoritmo que parece en la práctica. Para hacer más fácil el algoritmo se han creado dos clases nuevas: *Position* y *Nodo*.

- *Position*: contiene la información sobre una posición del mundo (coordenada X e Y).
- *Nodo*: contiene la información sobre una casilla del mundo. Está formado por:
 - *Position*: Posición que ocupa la casilla.
 - *Padre*: Referencia a un objeto *Nodo*, padre del objeto actual.
 - *g*: coste del camino que llevamos hasta el momento.
 - *h*: heurística. Aproximación al coste que falta hasta llegar al destino.
 - Además cuenta con una función *F()* que devuelve el valor de $f(g + h)$.

Se ha implementado el algoritmo A* visto en teoría y que aparece en la práctica:

```
Alg A*
  listaInterior = vacío
  listaFrontera = inicio
  mientras listaFrontera no esté vacía
    n = obtener nodo de listaFrontera con menor  $f(n) = g(n) + h(n)$ 
    listaFrontera.del(n)
    listaInterior.add(n)
    si n es meta
      devolver
      reconstruir camino desde la meta al inicio siguiendo los
      punteros
    fsi
    para cada hijo m de n que no esté en listaInterior
       $g'(m) = n.g + c(n, m)$  //g del nodo a explorar m
      si m no está en listaFrontera
        almacenar la f, g y h del nodo en (m.f, m.g, m.h)
        m.padre = n
        listaFrontera.add(m)
      sino si  $g'(m)$  es mejor que m.g //Verificamos si el nuevo
      camino es
        mejor
        m.padre = n
        recalcular f y g del nodo m
    fsi
  fpara
  fmientras
  devolver no hay solución
falg
```

A continuación lo explicaremos paso a paso:

- Inicializamos las dos listas. La lista interior está vacía y metemos el nodo inicial (punto de partida del robot) en la lista frontera. Para poder ser expandido.

```
listaInterior = vacío
listaFrontera = inicio
```

- A continuación seleccionamos el nodo con menor *f* de la lista frontera. Este es el nodo que vamos a expandir. Lo eliminamos de la lista frontera y no insertamos en la lista interior.

```
n = obtener nodo de listaFrontera con menor  $f(n) = g(n) + h(n)$ 
listaFrontera.del(n)
listaInterior.add(n)
```

- Ahora vamos a expandir el nodo. Pero primero comprobamos si el nodo es meta (hemos llegado al destino). En ese caso reconstruimos el camino y acabamos la ejecución del A*.

```

si n es meta
devolver
reconstruir camino desde la meta al inicio siguiendo los punteros
fsi

```

- Si no es el nodo meta. Expandimos el nodo. Primero generamos todos los hijos del nodo actual y para cada nodo que no esté en la lista interior ejecutamos el siguiente bucle:

```

para cada hijo m de n que no esté en listaInterior

```

- Primero actualizamos del nodo hijo su g:

```

g'(m) = n.g + c(n, m) //g del nodo a explorar m

```

- Si el nodo m no está en la lista frontera, almacenamos su valor de g y h y lo añadimos a la lista frontera. Asignándole como padre el nodo que estamos expandiendo.

```

si m no está en listaFrontera
almacenar la f, g y h del nodo en (m.f, m.g, m.h)
m.padre = n
listaFrontera.add(m)

```

- Por otro lado si ya está en lista frontera y la g'(m) calculada es menor que la g que ya tiene es que nos encontramos ante una solución menos costosa (un camino de llegada más corto) por tanto actualizamos los valores del nodo y le asignamos como padre el nodo que estamos expandiendo.

```

sino si g'(m) es mejor que m.g //Verificamos si el nuevo
camino es
mejor
m.padre = n
recalcular f y g del nodo m
fsi

```

- Si la ejecución sale del bucle porque se vacía la lista frontera es que no hay solución.

Siguiendo el algoritmo podemos representarlo en función de que opciones tenemos cuando evaluamos nuevos nodos.

Vamos a partir de la siguiente configuración:

- Lista Interior: [(4,1) G: 0 H: 4 F: 4] [(4,2) G: 1 H: 3 F: 4] [(3,1) G: 1 H: 3 F: 4] [(4,3) G: 1 H: 3 F: 4]
- Lista Frontera: [(2,1) G: 1 H: 3 F: 4] [(4,4) G: 1 H: 3 F: 4]
- Nodo seleccionado: [(2,1) G: 1 H: 3 F: 4]

Y vamos a ver cómo se comporta el algoritmo para el nodo generado [(2,2) G: 1 H: 2 F: 3].

- El nodo no está en lista Interior
 - Si el nodo tampoco está en lista frontera, lo añadimos:

Lista Frontera: [(4,4) G: 1 H: 3 F: 4] [(2,2) G: 1 H: 2 F: 3].

Vamos a partir de la siguiente configuración:

- Lista Interior: [(4,1) G: 0 H: 4 F: 4] [(4,2) G: 1 H: 3 F: 4] [(3,1) G: 1 H: 3 F: 4] [(4,3) G: 1 H: 3 F: 4]
- Lista Frontera: [(2,1) G: 1 H: 3 F: 4] [(4,4) G: 1 H: 3 F: 4] [(2,2) G: 3 H: 2 F: 5].
- Nodo seleccionado: [(2,1) G: 1 H: 3 F: 4]

Y vamos a ver cómo se comporta el algoritmo para el nodo generado [(2,2) G: 1 H: 2 F: 3].

- El nodo no está en lista Interior
 - Si el nodo está en lista frontera, por tanto comprobamos si mejora el camino recorrido (g) anterior. En este caso si cumple esa condición, por tanto lo actualizamos.

Lista Frontera: [(2,1) G: 1 H: 3 F: 4] [(4,4) G: 1 H: 3 F: 4] [(2,2) G: 1 H: 2 F: 3].

HEURÍSTICA

Vamos a trabajar con un conjunto de 4 mapas y 3 heurísticas:

Según la heurística que empleemos podemos encontrar o no la solución óptima:

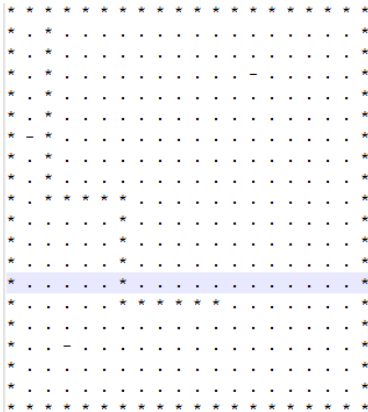
Si $h'(x)$ nunca sobrestima a $h(x)$ (o subestima), se garantiza encontrar el camino óptimo, pero se desperdicia esfuerzo explorando otras rutas que parecieran buenas.

Si $h'(x)$ sobrestima a $h(x)$, no puede garantizarse la consecución del camino del menor coste

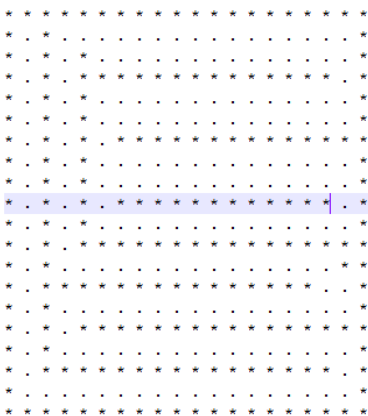
Estas son las 3 heurísticas que vamos a probar:

- $h'(x) = 0$: No vamos a tener en cuenta la heurística. Por tanto nos garantizamos obtener el camino óptimo aunque con un mayor coste porque se explorarían rutas no óptimas.
- Distancia Manhattan.
- Distancia Euclídea.

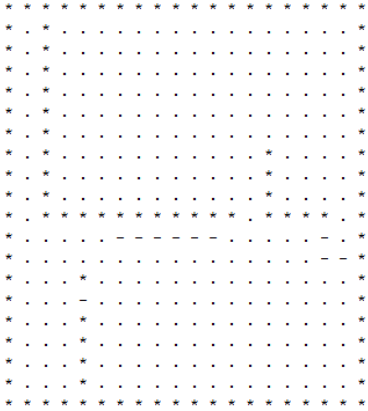
A continuación vamos a probar las distintas heurísticas con cada uno de los cuatro mapas.



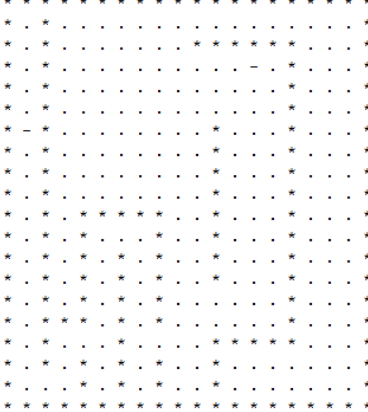
mundo.txt	$H(x) = 0$	Distancia Manhattan	Distancia Euclídea
Nodos explorados	178	80	118
Coste del camino	28	28	28



mundo2.txt	$H(x) = 0$	Distancia Manhattan	Distancia Euclídea
Nodos explorados	199	199	189
Coste del camino	136	136	136



mundo3.txt	$H(x) = 0$	Distancia Manhattan	Distancia Euclídea
Nodos explorados	250	129	169
Coste del camino	30	30	30



mundo4.txt	$H(x) = 0$	Distancia Manhattan	Distancia Euclídea
Nodos explorados	207	183	199
Coste del camino	62	62	62

Sin heurística la cantidad de nodos que se explora es mucho mayor. Por este motivo la mejor heurística, y la que vamos a utilizar en la práctica esa la de la distancia Manhattan. Porque suele explorar menos nodos que utilizando la distancia Euclídea.

TRAZA DE UN PROBLEMA SENCILLO

Vamos a realizar una traza de un problema sencillo, con un mapa de 6x6. Vamos a ir mostrando cómo evolucionan las listas frontera e interior, así como que nodo se escoge para explorar.

El mapa es el siguiente (mundo5.txt):



Iteración	Estado
0	Lista Interior: Lista Frontera: [(4,1) G: 0 H: 6 F: 6] Nodo seleccionado: [(4,1) G: 0 H: 6 F: 6]
1	Lista Interior: [(4,1) G: 0 H: 6 F: 6] Lista Frontera: [(4,2) G: 1 H: 5 F: 6] [(3,1) G: 1 H: 5 F: 6] Nodo seleccionado: [(4,2) G: 1 H: 5 F: 6]
2	Lista Interior: [(4,1) G: 0 H: 6 F: 6] [(4,2) G: 1 H: 5 F: 6] Lista Frontera: [(3,1) G: 1 H: 5 F: 6] [(4,3) G: 2 H: 4 F: 6] Nodo seleccionado: [(3,1) G: 1 H: 5 F: 6]
3	Lista Interior: [(4,1) G: 0 H: 6 F: 6] [(4,2) G: 1 H: 5 F: 6] [(3,1) G: 1 H: 5 F: 6] Lista Frontera: [(4,3) G: 2 H: 4 F: 6] [(2,1) G: 2 H: 4 F: 6] Nodo seleccionado: [(4,3) G: 2 H: 4 F: 6]
4	Lista Interior: [(4,1) G: 0 H: 6 F: 6] [(4,2) G: 1 H: 5 F: 6] [(3,1) G: 1 H: 5 F: 6] [(4,3) G: 2 H: 4 F: 6] Lista Frontera: [(2,1) G: 2 H: 4 F: 6] [(4,4) G: 3 H: 3 F: 6] Nodo seleccionado: [(2,1) G: 2 H: 4 F: 6]
5	Lista Interior: [(4,1) G: 0 H: 6 F: 6] [(4,2) G: 1 H: 5 F: 6] [(3,1) G: 1 H: 5 F: 6] [(4,3) G: 2 H: 4 F: 6] [(2,1) G: 2 H: 4 F: 6] Lista Frontera: [(4,4) G: 3 H: 3 F: 6] [(2,2) G: 3 H: 3 F: 6] [(1,1) G: 3 H: 3 F: 6] Nodo seleccionado: [(4,4) G: 3 H: 3 F: 6]
6	Lista Interior: [(4,1) G: 0 H: 6 F: 6] [(4,2) G: 1 H: 5 F: 6] [(3,1) G: 1 H: 5 F: 6] [(4,3) G: 2 H: 4 F: 6] [(2,1) G: 2 H: 4 F: 6] [(4,4) G: 3 H: 3 F: 6] Lista Frontera: [(2,2) G: 3 H: 3 F: 6] [(1,1) G: 3 H: 3 F: 6] Nodo seleccionado: [(2,2) G: 3 H: 3 F: 6]
7	Lista Interior: [(4,1) G: 0 H: 6 F: 6] [(4,2) G: 1 H: 5 F: 6] [(3,1) G: 1 H: 5 F: 6] [(4,3) G: 2 H: 4 F: 6] [(2,1) G: 2 H: 4 F: 6] [(4,4) G: 3 H: 3 F: 6] [(2,2) G: 3 H: 3 F: 6] Lista Frontera: [(1,1) G: 3 H: 3 F: 6] [(2,3) G: 4 H: 2 F: 6] Nodo seleccionado: [(1,1) G: 3 H: 3 F: 6]
8	Lista Interior: [(4,1) G: 0 H: 6 F: 6] [(4,2) G: 1 H: 5 F: 6] [(3,1) G: 1 H: 5 F: 6] [(4,3) G: 2 H: 4 F: 6] [(2,1) G: 2 H: 4 F: 6] [(4,4) G: 3 H: 3 F: 6] [(2,2) G: 3 H: 3 F: 6] [(1,1) G: 3 H: 3 F: 6] Lista Frontera: [(2,3) G: 4 H: 2 F: 6]

	Nodo seleccionado: [(2,3) G: 4 H: 2 F: 6]
9	Lista Interior: [(4,1) G: 0 H: 6 F: 6] [(4,2) G: 1 H: 5 F: 6] [(3,1) G: 1 H: 5 F: 6] [(4,3) G: 2 H: 4 F: 6] [(2,1) G: 2 H: 4 F: 6] [(4,4) G: 3 H: 3 F: 6] [(2,2) G: 3 H: 3 F: 6] [(1,1) G: 3 H: 3 F: 6] [(2,3) G: 4 H: 2 F: 6] Lista Frontera: [(2,4) G: 5 H: 1 F: 6] [(1,3) G: 5 H: 1 F: 6] Nodo seleccionado: [(2,4) G: 5 H: 1 F: 6]
10	Lista Interior: [(4,1) G: 0 H: 6 F: 6] [(4,2) G: 1 H: 5 F: 6] [(3,1) G: 1 H: 5 F: 6] [(4,3) G: 2 H: 4 F: 6] [(2,1) G: 2 H: 4 F: 6] [(4,4) G: 3 H: 3 F: 6] [(2,2) G: 3 H: 3 F: 6] [(1,1) G: 3 H: 3 F: 6] [(2,3) G: 4 H: 2 F: 6] [(2,4) G: 5 H: 1 F: 6] Lista Frontera: [(1,3) G: 5 H: 1 F: 6] [(1,4) G: 6 H: 0 F: 6] Nodo seleccionado: [(1,3) G: 5 H: 1 F: 6]
11	Lista Interior: [(4,1) G: 0 H: 6 F: 6] [(4,2) G: 1 H: 5 F: 6] [(3,1) G: 1 H: 5 F: 6] [(4,3) G: 2 H: 4 F: 6] [(2,1) G: 2 H: 4 F: 6] [(4,4) G: 3 H: 3 F: 6] [(2,2) G: 3 H: 3 F: 6] [(1,1) G: 3 H: 3 F: 6] [(2,3) G: 4 H: 2 F: 6] [(2,4) G: 5 H: 1 F: 6] [(1,3) G: 5 H: 1 F: 6] Lista Frontera: [(1,4) G: 6 H: 0 F: 6] Nodo seleccionado: [(1,4) G: 6 H: 0 F: 6]

```

Coste del camino: 7
Solución del A*
Camino
. . . . .
. . . . X .
. X X X X .
. X . . . .
. X . . . .
. . . . .
Nodos explorados
-1 -1 -1 -1 -1 -1
-1 7 -1 10 11 -1
-1 4 6 8 9 -1
-1 2 -1 -1 -1 -1
-1 0 1 3 5 -1
-1 -1 -1 -1 -1 -1

```