



### 1. Objetivo

Programar de forma autónoma el **PROBLEMA DEL MAPA**.

Se trata de obtener **la ruta más óptima** (*pathfinding*) entre el origen y el destino en un mapa con un número arbitrario de celdas cada una con acceso a las 8 adyacentes comparando y **analizando** diversas técnicas de búsqueda.

### 2. Elementos a utilizar:

- Lenguaje Python
- Librerías: numérica *NumPy*, estructuras de datos *pandas*, gráfica *Matplotlib* (opcional si se quieren implementar gráficas).
- Entorno Anaconda y Editor Jupyter

### 3. Implementación

1. El programa deberá leer un mapa a partir de un fichero `.map` con la siguiente estructura:

```
type octile
height x
width y
map
```

Donde x, y son las respectivas altura y anchura del mapa, el cual se encuentra almacenado a partir de la línea siguiente a “map” como una malla de caracteres ASCII. La **esquina superior izquierda** del mapa es la coordenada (0,0).

Cada celda del mapa puede ser uno de los siguientes caracteres:

```
. - terreno (transitable)
G - terreno (transitable)
@ - muro (no transitable)
O - muro (no transitable)
T - árboles (no transitable)
S - pantano (transitable y accesible desde terreno)
W - agua (transitable si estás dentro, pero no accesible desde terreno)
```

Una vez cargado el mapa pintarlo en pantalla con un código de colores razonable para el valor de cada casilla (terreno=blanco, muro, árboles, pantano y agua).

2. Leer un fichero de escenarios `.scen`, en el que cada línea del fichero contiene coordenadas origen-destino para realizar una búsqueda del camino más corto. El número de líneas es arbitrario. La primera línea del fichero es:

```
version 1
```

Cada línea posterior tiene 9 campos (Podemos ignorar el parámetro bucket):

Bucket	Map name	map width	map height	start x-coordinate	start y-coordinate	goal x-coordinate	goal y-coordinate	optimal length
--------	----------	-----------	------------	--------------------	--------------------	-------------------	-------------------	----------------

3. Ejecutar los algoritmos de búsqueda Avara y A\* sobre cada escenario del fichero `.scen`.

Para cada escenario y algoritmo de búsqueda, deberá de producir la siguiente salida:

1. Mapa completo incluyendo:
  - a. Las celdas cerradas (**gris**)
  - b. Las celdas abiertas (frontera) (**amarillo**)
  - c. Las celdas de la solución (**rojo**)
2. Comparación del coste final con la longitud óptima



3. Secuencia de coordenadas que sigue el camino encontrado, con el valor de  $f$ ,  $g$  y  $h$  para cada coordenada.

**Nota:** el mapa (bootybay.map) es un escenario del juego Warcraft III (publicado en Julio de 2002).

### 4. Ejercicios

1. La heurística a utilizar viene representada por la siguiente función:

```
function heuristica(celda) =  
    dx=abs(celda.x-goal.x)  
    dy=abs(celda.y-goal.y)  
    return D*(dx+dy)+(D2-2*D)*min(dx,dy)
```

donde

- D= distancia horizontal/vertical entre celdas
- D2= distancia diagonal entre celdas

Analizar y explicar la formula en función de los distintos valores de D y D2. Compararla con distancias conocidas (Manhattan, Chebyshev...).

2. Suponiendo que
  - La distancia horizontal/vertical es  $1 \cdot \text{coste}$  de la celda en la que estamos.
  - La distancia en diagonal es  $\sqrt{2} \cdot \text{coste}$  de la celda en la que estamos
  - El coste de todas las celdas transitables es 1
  - A. Derivar de la fórmula anterior la heurística más adecuada para el problema y justificarlo. ¿Es dominante, es admisible, es consistente?
  - B. Comparar las longitudes obtenidas con cada algoritmo con las longitudes óptimas del fichero de escenarios. ¿Qué algoritmo encuentra la mejor ruta? ¿Cuál expande menos celdas?
3. Suponiendo que ahora
  - La distancia en diagonal es  $1 \cdot \text{coste}$  de la celda en la que estamos
  - A. Derivar de la fórmula anterior la heurística más adecuada para el problema y justificarlo.
  - B. Comparar las longitudes obtenidas con cada algoritmo con las longitudes obtenidas en el ejercicio 2.
4. Suponiendo que ahora
  - El coste de todas las celdas transitables es

Terreno	1	Árboles	2 (ahora es transitable)
Pantano	3	Agua	4 (ahora es transitable y accesible)

  - A. Derivar de la fórmula anterior la heurística más adecuada para el problema y justificarlo.
  - B. Analizar el resultado en función del número y distribución de tipos de celda que haya en el mapa
  - C. ¿Qué pasa si cambiamos las relaciones de coste de las celdas? ¿Cómo se acelera o desacelera A\*?

### 5. Forma de entrega de la Práctica:

La entrega **GRUPAL** consistirá en un fichero comprimido RAR con nombre **LAB1-GRXX.RAR** subido a la tarea **Programación de BI** que **contenga únicamente**

1. **Por cada ejercicio (2, 3 y 4)** un notebook de Jupyter (archivos con extensión **.ipynb**).
2. Una **memoria del laboratorio** en Word.

**Las entregas que no se ajusten exactamente a esta norma NO SERÁN EVALUADAS.**

### 6. Corrección de la Práctica:

1. **IMPLEMENTACIÓN (factor 0/1)** Un 0 en este apartado implica el suspenso de la práctica



Siendo una práctica de IA, todos los aspectos de programación se dan por supuesto. La implementación será:

- Original: Código fuente no copiado de internet. Grupos con igual código fuente serán suspendidos
- Comentada: Inclusión (**obligatoria**) de comentarios
- Correcta: Los algoritmos de búsqueda están correctamente programados. El programa funciona y ejecuta correctamente todo lo planteado en el apartado “Ejercicios” de cada práctica. El programa funciona con otros casos de prueba planteados por el profesor.

**Se podrá llamar a los alumnos sobre cuyo código haya algún tipo de duda** para que lo defiendan oralmente. Si la defensa no es satisfactoria se considerará suspensa la práctica.

### 2. MEMORIA DE LA PRÁCTICA

Se evaluará

- La calidad del código y de la memoria entregada. Es obligatorio una redacción clara y correcta ortográfica/gramaticalmente de la memoria con la siguiente estructura:
  - *Portada con el nombre de los componentes del grupo y el número del grupo*
  - *Índice*
  - *Solución lógica: Espacio de estados, algoritmos de búsqueda*
  - *Resolución de los ejercicios y discusión*
  - *Bibliografía*
- Corrección de la solución lógica,
- Resolución correcta de todos los ejercicios (especialmente en lo referente a la heurística), uso de gráficas y discusión de los resultados de una manera lógica. Se valorará si se aporta trabajo propio sobre los ejercicios propuestos.