



Universidad Politécnica
de Madrid



Escuela Técnica Superior de
Ingenieros Informáticos

Máster Universitario en Inteligencia Artificial

Robots Autónomos

Sistema de Planificación de Ruta

Alejandro Francisco Toral

Madrid, 8 de enero de 2021

1. Instrucciones de uso

1.1. Descripción del trabajo realizado

La práctica se ha implementado en lenguaje Python, utilizando Jupyter en Windows 10. Consiste en un sistema de planificación de ruta basado en un algoritmo de búsqueda A*.

La imagen se lee y se convierte a una matriz binaria, donde los 0s son muros u obstáculos y los 1s, camino que recorrer.

Tras tratar la imagen, se inicializan dos puntos, uno de comienzo y otro del fin de la ruta aleatoriamente. Estos puntos no pueden ser iguales, ni pueden corresponder con un 0 ni pueden estar inicializados a los bordes de cada matriz. Internamente, el píxel de inicio será un 2 y el de fin, un 3.

Se ejecuta el algoritmo de estrella hasta encontrar el camino óptimo. El algoritmo A* utiliza una función de evaluación $f(n) = g(n) + h'(n)$ donde $h'(n)$ representa el valor heurístico del nodo a evaluar desde el actual, n, hasta el final, y $g(n)$, el coste real del camino recorrido para llegar a dicho nodo, n, desde el nodo inicial.

En este caso, la heurística utilizada es la **distancia euclídea** desde cada punto a la meta, es decir, la raíz cuadrada de la suma de las diferencias al cuadrado entre las coordenadas del punto final y el punto actual.

$$\sqrt{(i_{final} - i)^2 + (j_{final} - j)^2}$$

El algoritmo A* devuelve tanto el camino como el coste total. Finalmente, se convierte la matriz en una imagen png que muestra gráficamente el camino.

1.2. Requisitos

- Versión de Python 3.8
- Librerías:
 - cv2
 - matplotlib.pyplot
 - numpy
 - sys
 - PIL
 - random
 - matplotlib.cm
- Jupyter Notebook

En la misma carpeta se ha de tener lo siguiente:

- Cuaderno de Jupyter *path_planning.ipynb*
- Carpeta "imgs" con las imágenes de ejemplo

1.3. A tener en cuenta

Las imágenes deben ser minimizadas una vez se lean para que el algoritmo pueda terminar en poco tiempo. De no ser así, la ejecución puede tardar más de lo deseado.

1.4. Pasos a seguir

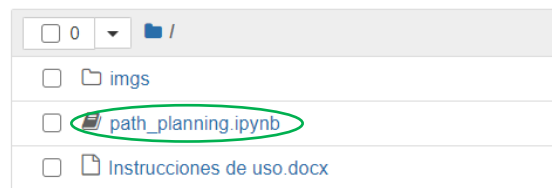
1. Abrir la consola de comandos desde la carpeta donde se encuentran los archivos y conectar con Jupyter Notebook

```
C:\Windows\System32\cmd.exe - jupyter notebook
Microsoft Windows [Versión 10.0.18363.1198]
(c) 2019 Microsoft Corporation. Todos los derechos reservados.
C:\Users\alexb\archivos_de_alex\Universidad\master\Asignaturas\Robots\Practica 3>jupyter notebook
```

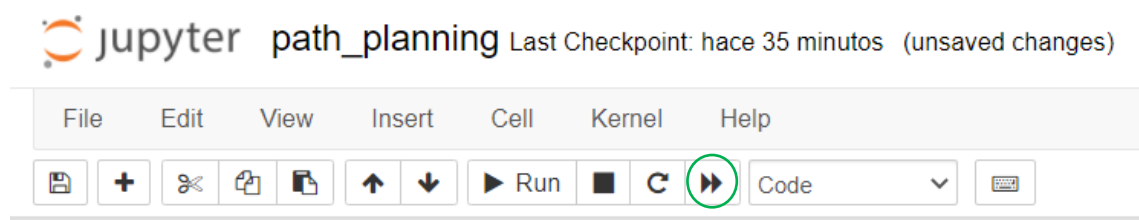
Se abrirá en el navegador predeterminado una ventana mostrando los archivos de la ruta seleccionada



2. Acceder al cuaderno



3. Pulsar en “Restart Kernel and run all”. El cuaderno viene por defecto con un ejemplo seleccionado.



4. En caso de querer cambiar de archivo, se debe descender hasta el apartado de “Selección de variables” y cambiar las variables “file_path”, “resize_w” y “resize_h”.
 - Si se desea añadir una imagen nueva, basta con sustituir la variable “file_path” por “imgs/nombre_archivo” y seleccionar unos nuevos valores de ajustado de la imagen. Se recomienda usar unos valores entre 10 y 15 para conseguir una ejecución óptima.

Apartado de selección de variables

```
# Read the image and get the binary matrix
"""
maze_file ==> Maze from the example assets of Moodle. Ideal size:
resize_w = 7
resize_h = 7

maze2_file ==> Maze from https://www.dreamstime.com/vector-illustration-maze-gray-background-layers-grouped-easy-editing-illu
Ideal size:
resize_w = 60
resize_h = 60

lab_file ==> Maze from the example assets of Moodle. Ideal size:
resize_w = 15
resize_h = 15

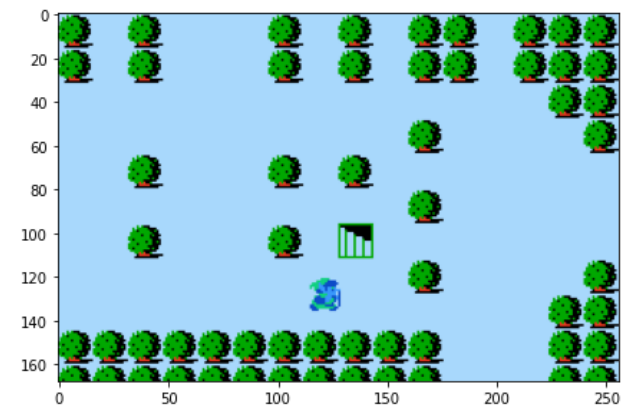
prado_file ==> Floor plan of the Prado Museum from the example assets of Moodle. Ideal size:
resize_w = 9
resize_h = 9

zelda_file ==> Screenshot from The Legend of Zelda (1986) for NES (Nintendo(R)). Ideal size:
resize_w = 15
resize_h = 15
"""

# Selected image
file_path = maze_file

# Times that the image will be shrinked
resize_w = 7
resize_h = 7
```

2. Ejemplo de la ejecución

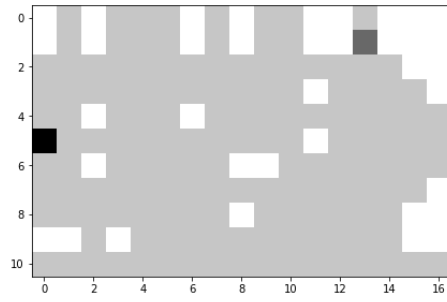


Base

```
01011101011001000
01011101011001000
11111111111111100
11111111111011110
11011101111111111
11111111111011111
11011111001111111
11111111111111110
11111111011111100
00101111111111100
11111111111111111
```

Maze with random start and goal

```
Start: (1, 13)
End: (5, 0)
```



Final path

Path taken: [(1, 13), (2, 13), (3, 13), (4, 13), (4, 12), (4, 11), (4, 10), (5, 10), (5, 9), (5, 8), (5, 7), (5, 6), (5, 5), (5, 4), (5, 3), (5, 2), (5, 1), (5, 0)]
 Final cost: 17

