Docs - Rescate - Manuel Ortega Puyol

Preambulo

Se suponen las siguientes limitaciones :

- Una iteracion equivale a un movimiento del conjunto de robots
- Radio de alcance de visión de 20 pixeles
- Radio de alcance de radiofrecuencia de 10 pixeles
- Los robots comparten:
  - Mapas locales
  - Objetivos conocidos
  - Objetivos rescatados conocidos
  - Niebla mas cercana
- Los robots tomaran un objetivo como rescatado **unicamente** si :
  - Llegan a pararse sobre él
  - Un objetivo previamente no rescatado vuelve a entrar a su viewport estando ya rescatado
  - Otro robot le comparte que ese objetivo ya esta rescatado

Clase *Robot* y *Estrategias*

Movimiento

Los robots intentarán ir a un objetivo siempre que este sea alcanzable, sino se pondran a explorar el mapa.

Busqueda de objetivos/victimas

Cada robot tiene tres sets y un contador :

- `self.objetivos_conocidos = set()`
- `self.objetivos_rescatados = set()`
- `self.objetivos_ignorados = set()`
- `self.unignore_countdown = self.UNIGNORE_COUTDOWN = 10`

💡 Cada vez que una victima entra en su campo de vision la agregan al set de *conocidos*

💡 Si no encuentran camino hacia el objetivo entonces este pasa a la lista de *ignorados*

- `unignore_countdown` es una cuenta atras que vuelve a tomar en cuenta los objetivos ignorados despues de una cantidad de pasos `UNIGNORE_COUTDOWN`

- Los objetivos que se rescatan salen de la lista de *conocidos* y entran a la lista de *rescatados*

Exploración

Cada robot calcula la niebla mas cercana a su alcance de vision y guarda los puntos, luego haran un A* intentar alcanzar el punto de niebla mas cercano

> El A* esta modificado para que si toca otro punto de niebla mientras calcula ruta de exploracion entonces se dirija a ese punto 😉 asi se vuelve aun mas eficiente moverse a la niebla mas cercana

Clase *TipoCasilla*

Clase `Enum` que representa:

```
NADA = 0
PARED = 1
NIEBLA = 4
ROBOT = 5
VICTIMA = 6
RESCATADO = 7
```

Función *convertir_imagen_a_matriz(url)*

Funcion del archivo `CreadorMapa.py` que recibe una String url de la cual cargara una imagen, evalua cada `pixel` y retorna una nueva matriz de `Casillas` que seran utilizadas como mapa.

Clase *Main*

Esta clase se encarga de tener todos los datos y pintar el mapa. En su bucle principal llama a cada robot a moverse.

En la simulacion se puede ver:

En general:

- Gris: Niebla
- Negro: Pared
- Rojo: Victimas
- Verde: Rescatados / Ruta de A estrella actual

- Amarillo: Niebla mas cercana al robot
- Cyan: Niebla mas cercana siendo evaluada

A estrella

- Rojo: Nodos cerrados
- Cyan: Nodos de la lista abierta

La simulacion puede verse de de menor a mayor detalle descomentando algunos bloques que pintan pixeles

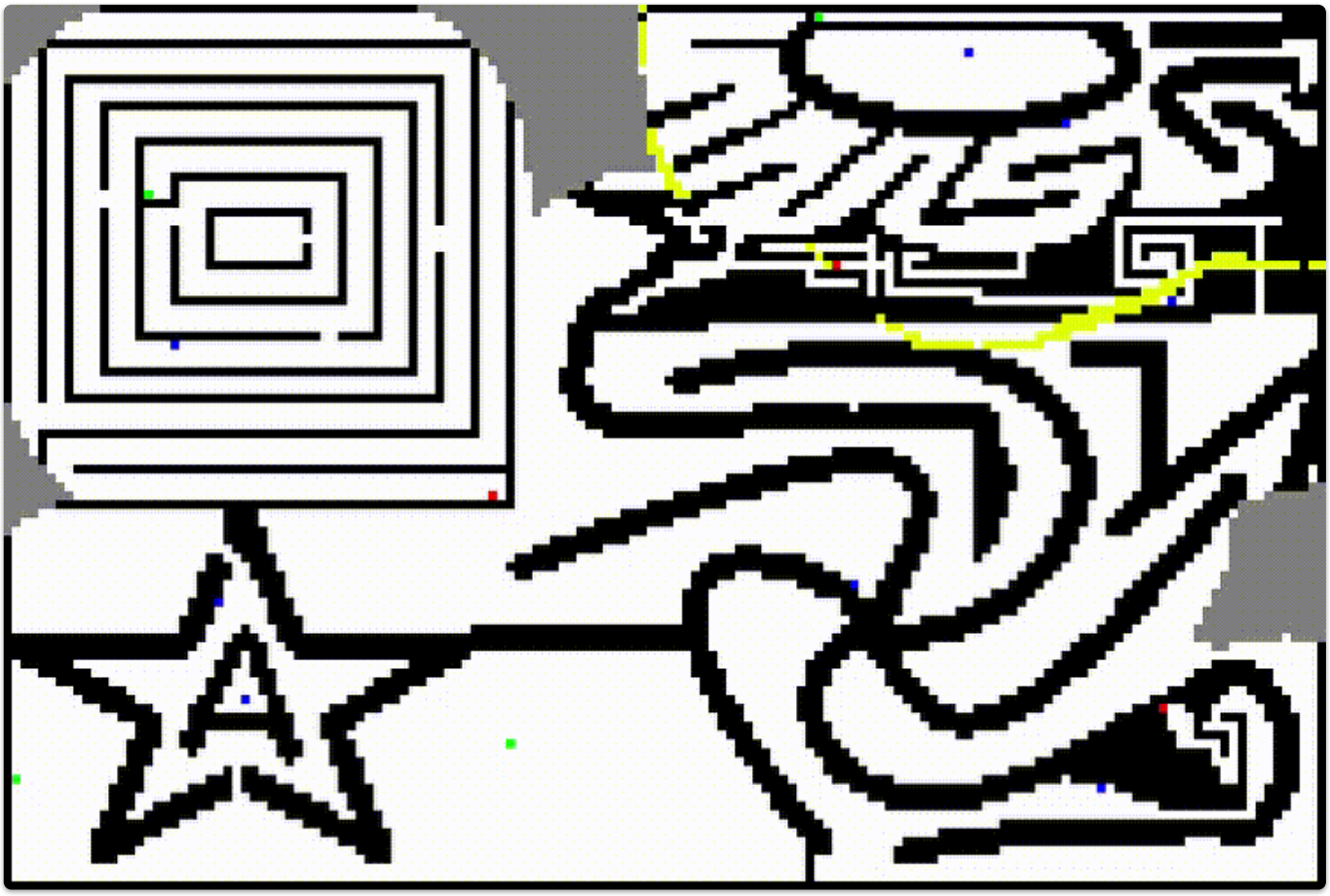
Local Rapida

```
        # Bucle principal dentro de Robot
        for i, fila in enumerate(robot.mapaLocal):
                for j, casilla in enumerate(fila):
```

General Rapida

```
        # Bucle principal dentro de Robot
```

```
        for i, fila in enumerate(self.niebla):
                for j, casilla in enumerate(fila):
```

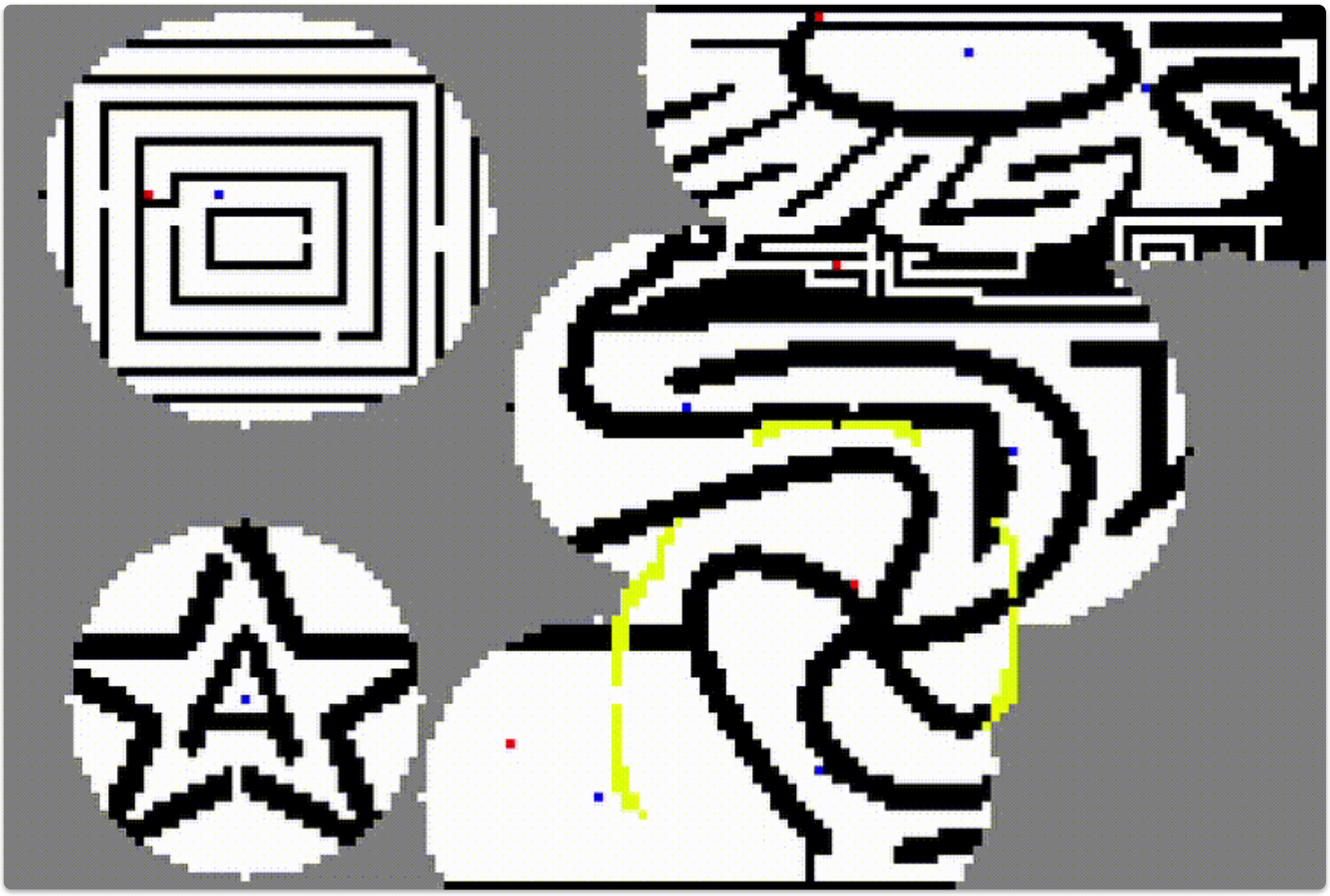


A*

```
        # Descomentar en main.py
                # elif robot.rutaAEstrella is not None and (i, j) in
robot.rutaAEstrella:
                #     color = GREEN

        # Descomentar en Aestrella.py
        #...
        x, y = current_node.coord
        # pygame.draw.rect(pantalla, (255, 0, 0), (y * 5, x * 5, 5, 5))
        # pygame.display.flip()
        neighbors = [
        #...
        #...
        #...
                open_list.append(neighbor)
        # pygame.draw.rect(pantalla, (0, 255, 255), (neighbor_coord[1] * 5,
```

```
neighbor_coord[0] * 5, 5, 5))
        # pygame.display.flip()
```



Atributos

- `ruta_imagen`: Ruta a la imagen
- `matriz_resultante`: Matriz de `Casillas` que representa el contenido de la imagen
- `filas`: Alto de la imagen
- `columnas`: Ancho de la imagen
- `tamano_casilla`: Entero que representa el tamaño de cada pixel en el display de pygame
- `iteraciones`: Entrero que representa el numero de iteraciones
- `rescatados`: Set con las victimas rescatadas usado unicamente para mostrarlo y detener el juego
- `niebla`: Matriz de `Casillas` usada para mostrar niebla en el display de pygame e inicializar los mapas locales de los robots
- `pantalla`: Objeto pantalla de pygame

Metodos

- `__init__`: Inicializa cada atributo

- `mapa_vacio` : retorna un mapa lleno de niebla