



Planificación Automática

Robot Control

Integrantes:

Ana Cortés Cercadillo

Carlos Javier Hellín Asensio

Path-planning

La implementación de Theta* se puede ver en r2p2/thetaStar.py

Para implementar las heurísticas se ha realizado el siguiente código en r2p2/heuristics.py

```
def manhattan(point, point2):
    """
    Function that performs Manhattan heuristic.
    """
    dx = abs(point.grid_point[0] - point2.grid_point[0])
    dy = abs(point.grid_point[1] - point2.grid_point[1])

    return dx + dy

pp.register_heuristic('manhattan', manhattan)

def naive(point, point2):
    """
    Function that performs a naive heuristic.
    """
    return 1

pp.register_heuristic('naive', naive)

def euclidean(point, point2):
    """
    Function that performs euclidean heuristic.
    """
    dx = (point.grid_point[0] - point2.grid_point[0]) ** 2
    dy = (point.grid_point[1] - point2.grid_point[1]) ** 2

    return sqrt(dx + dy)

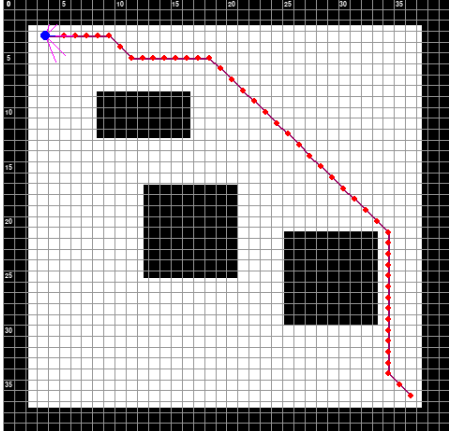
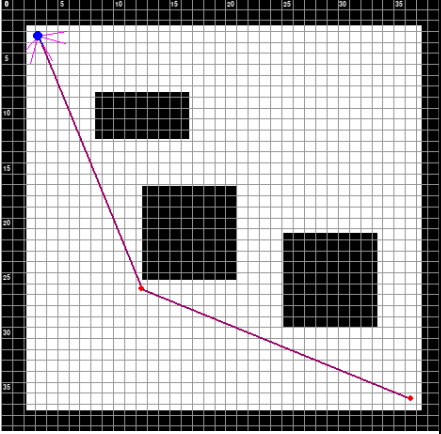
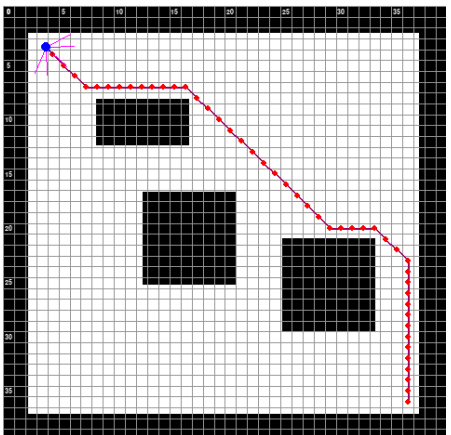
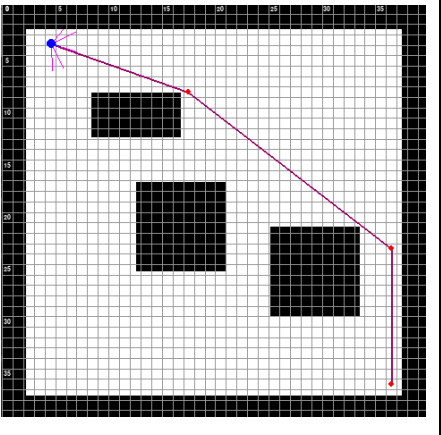
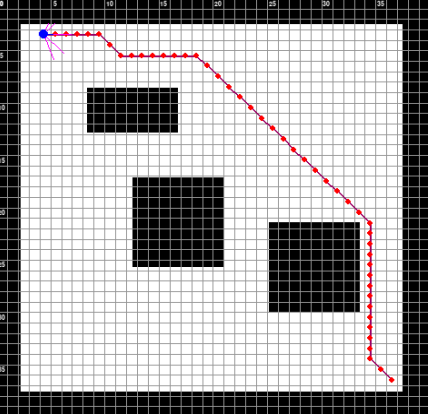
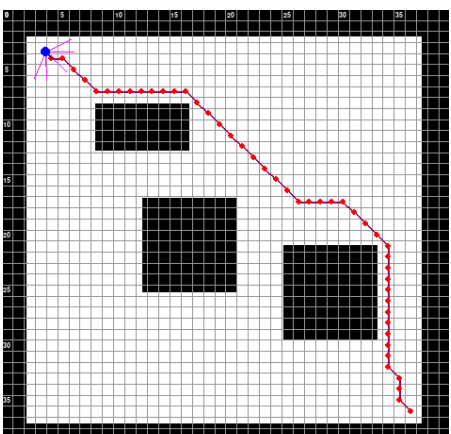
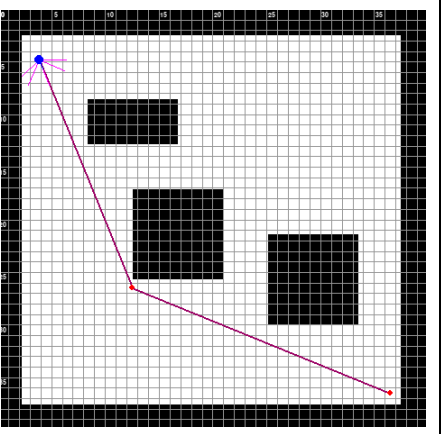
pp.register_heuristic('euclidean', euclidean)

def octile(point, point2):
    """
    Function that performs octile heuristic.
    """
    dx = abs(point.grid_point[0] - point2.grid_point[0])
    dy = abs(point.grid_point[1] - point2.grid_point[1])
    D = 1
    D2 = sqrt(2)

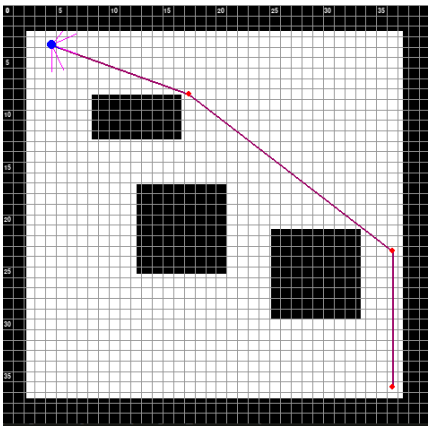
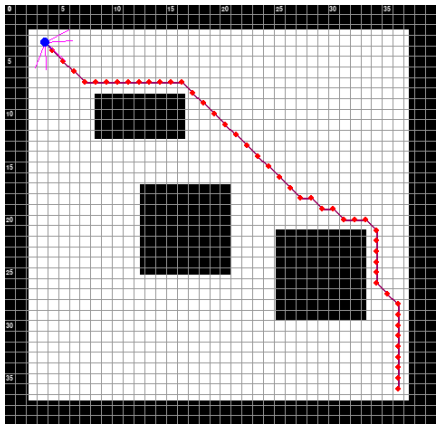
    return D * (dx + dy) + (D2 - 2 * D) * min(dx, dy)

pp.register_heuristic('octile', octile)
```

La siguiente tabla muestra los resultados de ejecutar los tres algoritmos cambiando las heurísticas:

Heurística/ Algoritmo	A*	Theta*	Dijkstra
Naive			
Manhattan			
Euclidean			

Octile



Integration

Se ha generado un plan del rover y se ha pegado en el planning.txt con lo siguiente añadiendo el primer move:

```
(move r p1011 p1010)

0.000: (drill rover1 p1010) [10.000]
0.000: (take_picture rover1 p1010) [2.000]
0.000: (extends-solar-panels rover1) [4.000]
6.628: (recharge_batteries rover1) [20.000]
10.001: (analyse-samples rover1 p1010) [8.000]
18.002: (earth-communication rover1 p1010) [8.000]
26.002: (move rover1 p1010 p0506) [0.625]
26.627: (drill rover1 p0506) [10.000]
36.628: (analyse-samples rover1 p0506) [8.000]
42.628: (take_picture rover1 p0506) [2.000]
44.629: (earth-communication rover1 p0506) [8.000]
49.630: (extends-solar-panels rover1) [4.000]
52.629: (move rover1 p0506 p1314) [1.000]
53.630: (recharge_batteries rover1) [32.000]
59.629: (drill rover1 p1314) [10.000]
69.630: (analyse-samples rover1 p1314) [8.000]
75.630: (take_picture rover1 p1314) [2.000]
77.631: (earth-communication rover1 p1314) [8.000]
85.631: (move rover1 p1314 p1234) [1.250]
86.881: (drill rover1 p1234) [10.000]
96.882: (analyse-samples rover1 p1234) [8.000]
102.882: (take_picture rover1 p1234) [2.000]
104.883: (earth-communication rover1 p1234) [8.000]
```

Se ha dejado como por defecto el algoritmo Theta* y heurística euclidean en el fichero controller-planning.json

```
{
  "class": "pddl_executor.PDDL_Executor",
  "plan_path": "../res/planning.txt",
  "algorithm": "Theta*",
  "heuristic": "euclidean"
}
```

Reflexion for building intelligent robots

A lo largo de la práctica hemos usado planos de dimensiones consideradas pequeñas. En los planificadores que se usan para los robots reales, necesitan tener mapas más grandes para conocer el terreno sobre el que se mueven. Para las divisiones de estos mapas se necesitan rejillas más grandes de las que hemos usado en esta pequeña simulación. Esto nos lleva a la conclusión de que para manejar las rejillas más grandes se necesitan máquinas más potentes.

También creemos que como algo necesario para mejorar la inteligencia de los robots es disponer de otro fichero aparte y más complejo que permita trabajar con entornos 3D. Esto serviría para los algoritmos de planificación de caminos el tener un mapa de costes pero, en este caso, en 3D buscando un entorno más realista.

Y por último, desde el punto de vista de la planificación, sería mejorar la selección de acciones guiadas por una función coste que sirva para conocer las veces que una acción falla debido a un camino inalcanzable.