Internet of Things

Pathfinding algorythm

Introduction

This algorithm will try to scan a room thanks to device on wheels, he will try to go accross the greatest surface possible and he will log his path. The device will send his current position and his future direction each time the server ask it.

The device will send data in JSON about his distance from an obstacle and his rotation angle. The server will analyze these data and then send an uptaded order if needed with the rights parameters to execute it.

Each new order will send parameters about the rotation angles for the wheels, the vehicle’s state (rolling or stop, backward.), the order’s execution time and finally the order`s number (in order to make a backlog).

The device will send his data under the following form.

```
{
Distance : ?
Degree : ?
}
```

Distance = is the distance in centimeters between the device and an obstacle
Degree = This is his current position from the north (example: 180 will means that the device’s direction is heading the north)

The server will send those parameters for his orders

```
{
rotation = ?
direction = ?
time = ?
order = ?
}
```

Rotation = indicates that the device’s wheels will have to turn until they reach the asked rotations degree.

Direction = Direction represents the vehicle’s state. If Direction = 0, that means that the car is stopped. If equal 1, that’s means that the vehicle is heading forward. If 2, that’s means that the car is heading backward.
Time = The number of millisecond while the car will execute the order to send.

Order = Is the order’s number, if this is the third order sent since the beginning of the scan, the Order’s value will be 3.

Step 1 : Initialisation

In this actual implementation, the device will be in the corner of a room and always scanning the room heading the opposite wall to his right.

The default position of the wheels in degree is 180, so the device will check in there is no obstacle is before him regarding these parameters.

“if (distance > 10 && degree == 180 || degree == 0)”

“Distance > 10” means that if no obstacle is before the device below 10 centimeters.

“degree == 180 || degree == 0” if the device is heading in the same direction than his initial direction or heading the opposite direction.

If these conditions are respected, the device will begin to scan straight ahead, and will do same each time that those conditions are respected, just like when he is rolling straight ahead in order to join the opposite wall.

The server sends in this case the order with the following parameters:

Rotation = 180 if the device is in the same direction than the initial one
Direction = 1 to indicate to the device to roll forward.
Time = 5000 to indicate that the device will have to execute the order during 5 seconds.
Order = 1 if this the first order sent by the server.

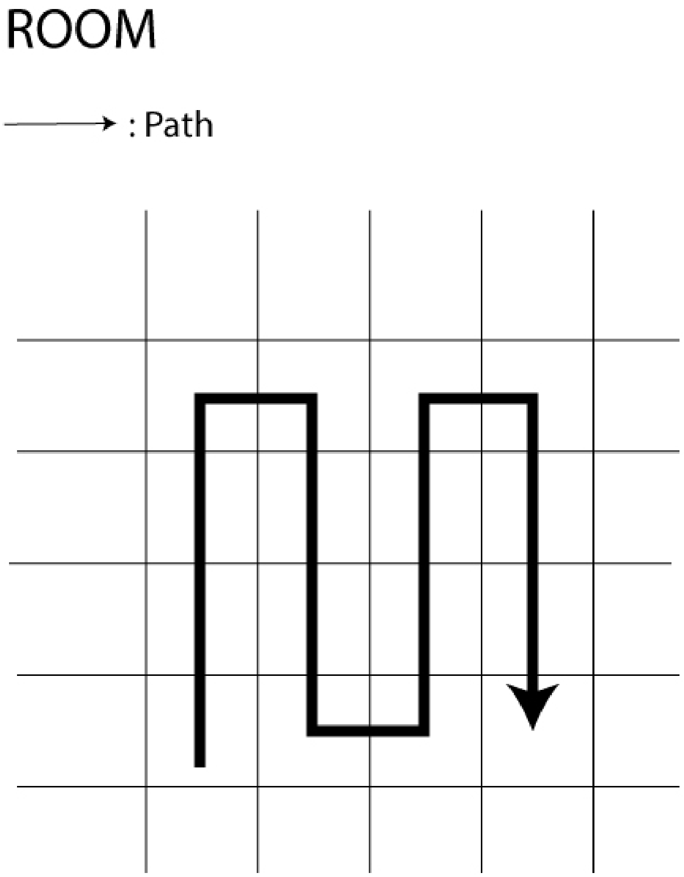
ROOM

? : Startpoint

	?			?	
	?			?	

Step 2 : Path

When the device begin to move, he will try to go the furthest right to the direction to his right or left, where his the opposite corner of the room. He go straight until he meet an obstacle (wall) and then he will make a Uturn (or turn back), his direction will be define by his startpoint. He will continue like that until he meets the well at his left or right and unable to turn left or right.



If the starpoint was in the right/south corner of the room, the path would have been the opposite.

Step 3 : Process

When the device is making his way into the room, he will be applying a command whom will be updated for diverse reasons :

- If the device meet an obstacle will tell him to turn in a direction in order to avoid it and to continue the scanning. If the car meet a wall that means that the data “distance” have a value under 10. That means that the obstacle is in the distance less than ten.

“if (distance < 10 && degree == 180)” means that the obstacle is nearby and that the device is heading forward from the bottom to the top of the room. The next order sent by the server will indicate to the car to make a Uturn, to turn back in order to go into the opposite direction closer to the opposite wall (right).

“if (distance < 10 && degree == 180)” means that the obstacle is nearby and that the device is heading forward from the top to the bottom of the room. The next order sent by the server will indicate to the car to make a Uturn, to turn back in order to go into the opposite direction closer to the opposite wall (left).

When the device makes a Uturn, his wheels will turn of 45 degrees and will go forward and rotate until his direction the opposite one depending on his current direction.

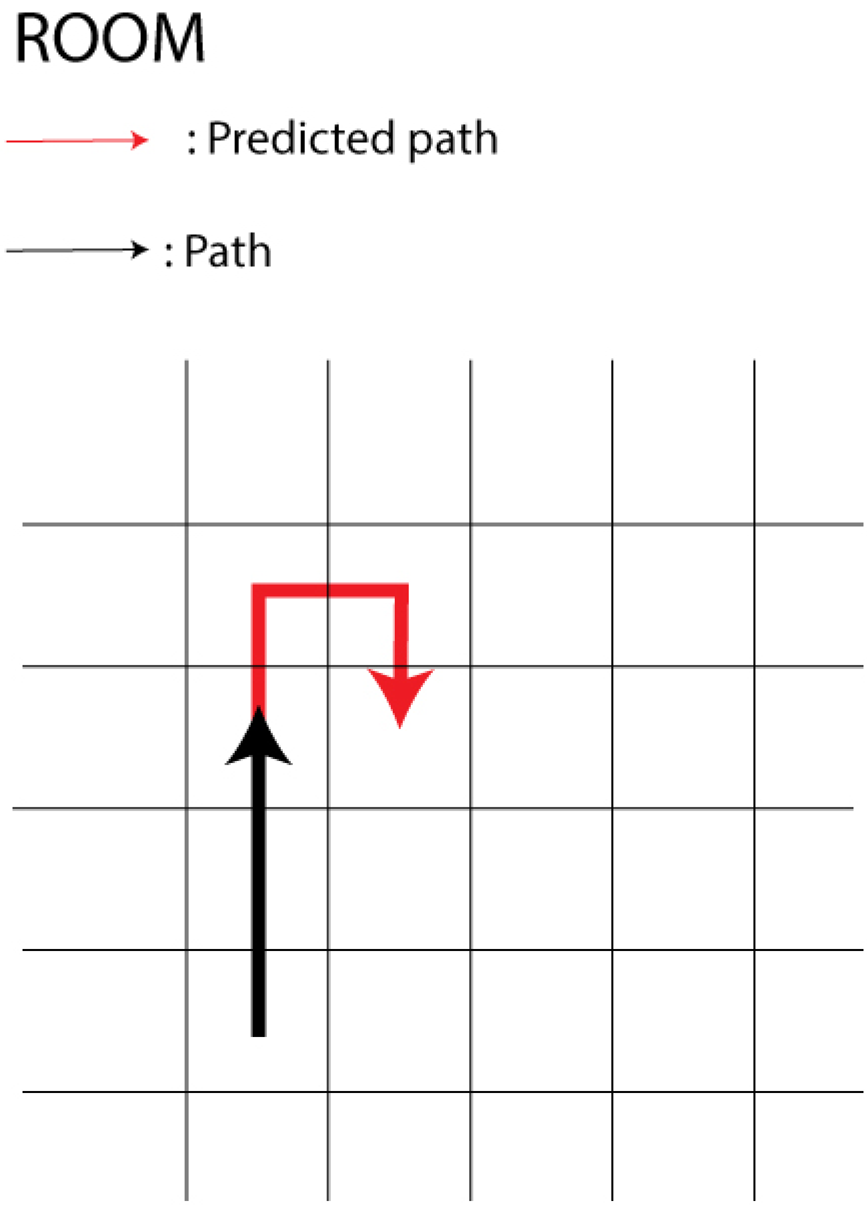
Example:

“if (distance < 10 && degree == 180)” the order with the returned parameters by the server will be

Rotation = 0 who indicate the opposite current direction
Direction = 1 to indicate to the device to roll forward.
Time = 3000 to indicate that the device will have to execute the order during 5 seconds.
Order = ?

So the device will turn until his direction reach 0.

- So the device will send to the server his current position and his future path.


ROOM
: Predicted path
: Path

Step 4 : Special case

If the device meet an obstacle who is not right in front, the device will do as always, turn back and then go further to his right or left. The main preoccupation is to ensure that the device is able to go right into the north or south without visiting the already saw path.

The end of the scan is known when the device have no way to turn furthermore, at the end of his last order, if the obstacle is always here, the car will simply