I didn’t used RUBY before, but I did some searching on the Internet, i found PYTHON have similar with RUBY. I just started learning PYTHON by myself recently, so I took the time to look at the tutorial of PYTHON.

First things come up on my mind is a tank game, very similarly mechanism, except that the left and right directions doesn’t change the X-axis variable. (Turning 90 degree).

Before go into the coding, I want to go through the mathematical logic first.

So far we know that, if we need to find the distance between two coordinate,

The formula will be:

Start point: P1(x1, y1),

End point: P2(x2, y2)

Distance =

However, the robot can’t go directly back home, it mean the formula need to change. We know the robot can only go in north, south, east, west directions, so that robot only do the horizontal and vertical moving on the coordinate.

So I change the formula into this:

Start point: P1(x, y),

End point: P2 (a, b)

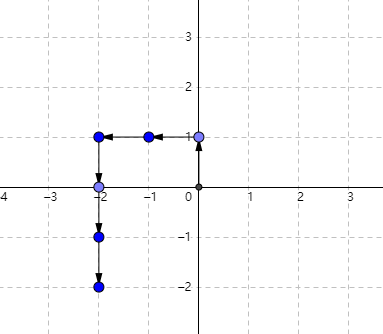
Distance =

Sum the both absolute value.

Now let’s take a look the example. If we have radar and coordinate.



So the trajectory of robot will be like:



Start point: P1 (0, 0),

End point: P2 (-2,-2)

Distance = = 4

Match the answer.

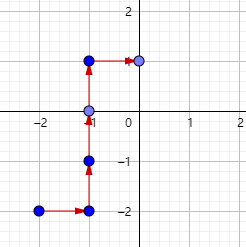
Even if we want to calculate the minimum amount of distance to get back to the first stopping point, let’s try:

The point that you want to go back to, in this case will be: P3 (0, 1)

End point: P2 (-2,-2)

Distance = = 5

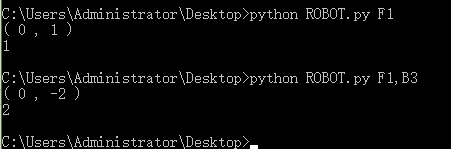
And it’s 5 unit length.



Once we figure out the mathematical logic, we can write code with a clear idea.

We know that command code `R` and `L` don’t move the robot. So I go with the forward and backward first.

Testing:



For the turning degree, I spent too much time on it, at first I was trying to use some mathematical function to pass by the value, because consider of the extensibility and robot should work in any situation.

Also, I had try the recursion method, however error happened. Then I realized that it could make it simple. So I add facing direction estimate.

We give value to the facing directions

Facing = [“north”, “south”, “east”, “west”]

West: 3

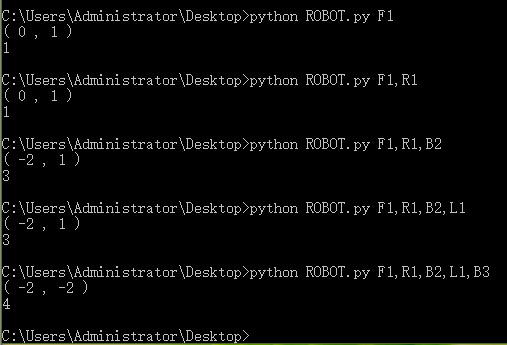
South: 1

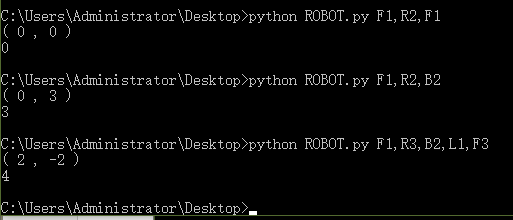
East: 2

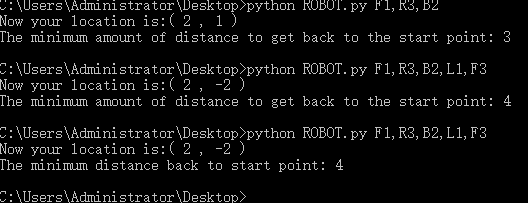
North: 0

Every time when user input command, check the facing directions value first, then change the value depend on turning. Also change the “Forward” and “Backward” command formula if needed, because in some case, Forward/Backward could be X-axis adding or subtracting.

Testing:







Should be fine.