

## Aryan Gupta

```
def calc_encoder_count (distance):
    return math magic that converts distance to how many encoder counts needed

def calc_degrees_to_distance (degrees):
    return math magic that converts degrees to a distance from the wheelbase and diameter

def turn (degrees, num_wheels):
    distance = call calc_degrees_to_distance with degrees
    count = call calc_encoder_count with distance

    if using both wheels:
        if (degrees < 0): # turn right (CW)
            raise not implemented
        else: # turn left (CCW)
            while both counts absolute value are less than count:
                call rotate_left_motor_speed with -5%
                call rotate_right_motor_speed with 5%
    else:
        raise not implemented

def travel_straight (distance):
    count = call calc_encoder_count with distance

    call set_both_motor_speed with 10%

    while both counts are less than count:
        if left and right counts vary by more than 2 counts:
            if left count is larger than right count:
                call increase_left_motor_speed with 5%
                call decrease_right_motor_speed with 5%

            if right count is larger than left count:
                call increase_right_motor_speed with 5%
                call decrease_decrease_motor_speed with 5%

def loop (void):
    call reset_encoder

    repeat 4 times:
        call travel_straight with 0.5 meter
        call turn with 90 deg and both wheel

    repeat 3 times:
        call travel_straight with 0.5 meter
        call turn with 60 deg and both wheel
```