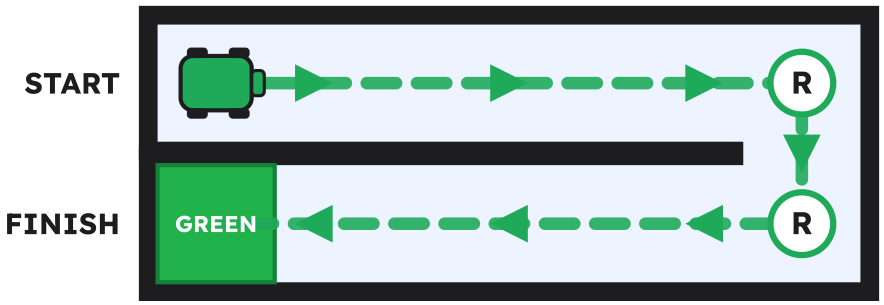
MISSION 1 · SOLUTION

The Right Hook

There is more than one right answer. Yours may look different — that is fine if it works.

THE ROUTE



Turns in order: 1: Right | 2: Right | then green → stop

FIRST

Make these My Blocks

turn right
motor D go to position 60° → move forward for 20 cm → motor D go to position 0° · Start values — tune until the car ends up facing sideways.

drive to the wall
start moving forward → wait until distance sensor B closer than 20 cm → stop moving · The turret is facing forward (position 0).

celebrate
set hub light to green → play beep 80 for 0.5 s → play beep 100 for 0.5 s · Sound + light on the tile.

THE PROGRAM

Word Blocks

when program starts

set movement motors to A + E
use the ports from YOUR car

set movement speed to 30 %

drive to the wall
My Block (turret facing forward)

turn right
My Block

move forward for 36 cm
one lane + one wall — measure yours

turn right
My Block

start moving forward

wait until color sensor F sees green
the belly sensor has reached the tile

stop moving

celebrate
My Block: sound + light

– Same thing in Python

```
from hub import port, sound, light_matrix
import runloop, motor, motor_pair, distance_sensor, color_sensor, color

DRIVE      = motor_pair.PAIR_1
STEER      = port.D            # large motor, steers the back
EYES       = port.B            # distance sensor (rides on the turret)
BELLY      = port.F            # color sensor pointing down
STOP_MM    = 200               # wall closer than 20 cm ahead = stop and look
LANE_CM    = 36                # one lane + one wall
DEG_PER_CM = 20                # wheel degrees per cm: tune
TURN_CM    = 20                # roll while steering: tune

def wall_closer_than(mm):
    d = distance_sensor.distance(EYES)
    return 0 < d < mm

def on_color(c):
    return color_sensor.color(BELLY) == c

async def drive_cm(cm, speed=300):
    await motor_pair.move_for_degrees(DRIVE, int(cm * DEG_PER_CM), 0, velocity=speed)

async def drive_until_wall():
    motor_pair.move(DRIVE, 0, velocity=400)
    await runloop.until(lambda: wall_closer_than(STOP_MM))
    motor_pair.stop(DRIVE)

async def drive_until_green():
    motor_pair.move(DRIVE, 0, velocity=300)
    await runloop.until(lambda: on_color(color.GREEN))
    motor_pair.stop(DRIVE)

async def turn_right():
    await motor.run_to_absolute_position(STEER, 60, 500)
    await drive_cm(TURN_CM)
    await motor.run_to_absolute_position(STEER, 0, 500)

async def celebrate():
    light_matrix.show_image(light_matrix.IMAGE_HAPPY)
    await sound.beep(880, 300)
    await sound.beep(1175, 500)

async def main():
    motor_pair.pair(DRIVE, port.A, port.E)      # your drive motor ports
    await drive_until_wall()
    await turn_right()
    await drive_cm(LANE_CM)
    await turn_right()
    await drive_until_green()
    await celebrate()

runloop.run(main())
```

Starting values only. Steering angle, roll distances, the OPEN limit and the line target depend on *your* car and maze. Measure, then

tune one at a time. I have not run these on a car — expect to adjust.

LEVEL UP

Ready for more?

- Make one My Block called *U-turn right* that does all three moves.
- Make the sound different for each mission you finish.

Keep going: Hints → `mission-1-hints.pdf` | All missions → `index.pdf` | Next mission → `mission-2-challenge.pdf`
The same content is in the matching .html files.