



8.2 Collecting data

Name _____

Class _____

Date _____

What this lesson is about

Readings with labels, and why bad data makes bad robots.

- Four situations
- Getting to the spots
- Bad data makes bad robots

Questions

6 marks in all

1. In this module, what is a sample? [1 mark]

- ☐ A A view (the 16 level readings) paired with a label
- ☐ B Just the 16 level readings
- ☐ C The label the robot guessed
- ☐ D The robot's position when it recorded

2. Which of these produce a robot that does the wrong thing with total confidence? [1 mark]

Tick every answer that is true.

- ☐ A Recording the wall once with the label open
- ☐ B Recording open only at the start
- ☐ C Recording three headings at each spot
- ☐ D Using labels that mean where the robot is, not what it sees
- ☐ E Recording after the robot has settled

3. Why does `drive_to` end with `wait(0.4)` before a sample is recorded? [1 mark]

- ☐ A So the readings are taken after the robot has settled, not while it is still sliding
- ☐ B The depth sensor needs 0.4 s to switch on
- ☐ C To give you time to type the label
- ☐ D `tof_grid()` only works after a wait

4. What does this program print? [1 mark]

```
data = []
data.append([54, 52, 51], "open")
data.append([16, 16, 16], "wall-ahead")
print(len(data), data[-1][1], data[0][0][1])
```

5. Who decides the label on a sample?

[1 mark]

- ☐ A You do, and if you get it wrong the model learns your mistake
- ☐ B The robot works it out from the readings
- ☐ C The depth sensor measures it
- ☐ D The model fixes wrong labels during training

6. What does the label `gap-left` mean?

[1 mark]

- ☐ A The wall ends on my left, wherever I happen to be
- ☐ B The robot is standing on the left spot, (-38, 45)
- ☐ C Turn left now
- ☐ D There is a gap somewhere on the left of the mat

The task: collect data

Visit the four spots in order, record the level rows at each with its label, and print `recorded <label>` each time.

```
# the two lines every program starts with: the commands, then the robot
from bugbot import *
connect()
# maths: atan2, hypot, sin, cos, radians
import math

def wrapped(h):
    return (h + 180) % 360 - 180

def go_to(x, y, speed=60):
    # where am I?
    px, py = position()
    a = math.radians(wrapped(math.degrees(math.atan2(x - px, y - py)) - heading()))
    # forward, sideways, rotation: -100 to 100 each, until the next command
    drive(speed * math.cos(a), speed * math.sin(a), wrapped(0 - heading()) * 3)

def near(x, y, cm=4):
    # where am I?
    px, py = position()
    return math.hypot(x - px, y - py) < cm

def drive_to(x, y):
    while not near(x, y):
        go_to(x, y)
        # pause 0.1 s (the robot keeps doing what it was told)
        wait(0.1)
    # all motors off
    stop()
    # pause 0.4 s (the robot keeps doing what it was told)
    wait(0.4)
drive_to(0, 10)
print('recorded', 'open')
```

Plan your program here, then type it in and press Run.



Do it on the robot

www.bugbotlab.com/learn/8-2-collecting-data/

The simulator checks it and tells you when it passes. Nothing to install, no account.

Challenges

1. Record three samples at each spot: facing 10 degrees left, straight, and 10 degrees right.
2. Print `data` at the end. That text is your dataset; lesson 8.7 shows how to keep it.
3. Add a fifth label, `corner`, recorded in a corner of the mat. What does the robot see there?