ANSWERS

2.3 The depth grid

Sensing · Robot club · about 20 min

What this lesson is about

64 readings, 8 columns: seeing left and right, not just ahead.

- tof_grid()
- Seeing it as a picture
- Columns are directions
- Left and right

Questions

7 marks in all

1. The grid is 8 rows of 8, and a reading is at `grid[row * 8 + col]`. At what position in the list is row 2, column 5? [1 mark]

> **Answer:** 21. Rows and columns both count from 0. Row 2 starts at 2 times 8, which is 16, and column 5 is 5 along from there: 21.

2. Which rows of the grid look level, straight ahead? [1 mark]

- ☐ **A** Rows 2 and 3
- ☐ **B** Rows 0 and 1
- ☐ **C** Rows 6 and 7
- ☐ **D** All eight rows

> **Answer:** A. Row 0 is the top of the view and looks over most things. The bottom rows see the mat. Rows 2 and 3 are the ones to read.

3. Why are the bottom rows of the grid always close readings? [1 mark]

- ☐ **A** The sensor sits 3 cm above the mat, so the bottom rows see the mat a few centimetres ahead
- ☐ **B** There is always a wall in front
- ☐ **C** The bottom rows are broken
- ☐ **D** The robot is always driving uphill

> **Answer:** A. The view tilts down at the bottom, and the mat is right there, so those readings are always short.

4. Which direction does column 0 look? [1 mark]

- ☐ **A** About 20 degrees to the left
- ☐ **B** About 20 degrees to the right
- ☐ **C** Straight ahead
- ☐ **D** Straight down at the mat

> **Answer:** A. Columns go left to right: 0 looks about 20 degrees left, 7 about 20 degrees right, and 3 and 4 straight ahead.

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

```
row = [30, 90, 60, 20]
line = ""
for d in row:
    line += "#" if d < 40 else ("+" if d < 80 else ".")
print(line)
```

> **Answer:**
>
> ```
> #.+#
> ```
>
> Under 40 is `#`, under 80 is `+`, and anything further is `.`. 30 and 20 are close, 60 is medium and 90 is far.

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

```
left_side = 45
right_side = 80
if left_side < right_side:
    print("more room on the right")
else:
    print("more room on the left")
```

> **Answer:**
>
> ```
> more room on the right
> ```
>
> The left reading is smaller, so things are closer on the left, and there is more room on the right.

7. The level-row readings for columns 0 to 7 are 35, 38, 90, 120, 110, 40, 33, 30. Which way is the way through? [1 mark]

- ☐ **A** Column 3, about straight ahead
- ☐ **B** Column 0, the left edge
- ☐ **C** Column 7, the right edge
- ☐ **D** Column 5

> **Answer:** A. The largest reading, 120, is in column 3. The biggest distance is where there is most room to drive.

The task: widest gap

There are two blocks ahead, one left and one right, with a gap between. Without moving, print `clear: column <n>` for the column with the largest reading in the level rows: that is the way through.

```
# the two lines every program starts with: the commands, then the robot
from bugbot import *
connect()

# 64 distances, 8 rows of 8
grid = tof_grid()
best_col = 0
best = 0
# find the column whose reading in the level rows is the largest
print("clear: column", best_col)
```

The hint students can ask for: Without moving, look at tof_grid() and print the column (0 to 7) with the largest reading: that is the way through.

A solution

```
from bugbot import *
connect()
grid = tof_grid()
best_col = 0
best = 0
for col in range(8):
    d = min(grid[row * 8 + col] for row in (2, 3))   # the level rows
    if d > best:
        best = d
        best_col = col
print('clear: column', best_col)
```

Any program that meets the task's checks is marked correct in the simulator; this is one way, not the only way.