



10.3 Watching the others

What this lesson is about

others() and the camera tags, and giving way early so contact never costs you the game.

- Where everyone is
- Giving way

Questions

6 marks in all

1. In the traffic task, a student calls `others()` to find the moving robots. What happens? [1 mark]

- ☐ A It does not work: `others()` is only there in games, so in a task use the camera
- ☐ B It lists both robots with their positions
- ☐ C It returns what the camera can see
- ☐ D It sends a message asking where they are

Answer: A. `others()` is the game's own view. In a task, `robots()` shows what is in front of you, each one found by its dome colour.

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

```
import math
bots = [{"name": "red-1", "x": 30, "y": 40}, {"name": "blue-2", "x": -6, "y": 8}, {"name": "blue-3", "x": 20, "y": 0}]
closest, best = None, 1e9
for bot in bots:
    d = math.hypot(bot["x"], bot["y"])
    if d < best:
        closest, best = bot["name"], d
print(closest, best)
```

Answer:

blue-2 10.0

Measured from (0, 0) the distances are 50, 10 and 20, so blue-2 is closest at 10.0 cm.

3. You touch another robot that drove into you. Whose fault is it, in most games? [1 mark]

- ☐ A Yours: touching another robot is your fault, whoever moved
- ☐ B Theirs, because they moved
- ☐ C Nobody's
- ☐ D Whoever has the lower score

Answer: A. That is why seeing other robots early and giving way matters.

4. Which habits keep you out of trouble on a busy mat?

[1 mark]

Tick every answer that is true.

- ☐ A Look before you commit
- ☐ B Go round with `left()` or `right()`
- ☐ C Give way early
- ☐ D Speed up to get past before they arrive
- ☐ E Push through, since they moved into you

Answer: A, B, C. A quick look, a sidestep and a little slowing cost almost nothing. A collision costs the game.

5. `robots()` comes back as `[['red', 160, 118, 35.0]]`. What is 35.0?

[1 mark]

- ☐ A How far away that robot is, in centimetres
- ☐ B Its bearing in degrees
- ☐ C How big its dome looks, in pixels
- ☐ D Its id

Answer: A. Each robot is `[colour, cx, cy, distance]`, nearest first. The bearing comes from `cx : (cx - 160) * 120 / 320`.

6. What does this program print?

[1 mark]

```
for gap in [40, 25, 12]:
    if gap < 25:
        print(gap, "step aside")
    else:
        print(gap, "carry on")
```

Answer:

```
40 carry on
25 carry on
12 step aside
```

`gap < 25` is false when the gap is exactly 25, so only 12 is close enough to step aside.

The task: through the traffic

Two robots are driving back and forth across the mat, 35 cm and 55 cm ahead of you. Get from the start to the green zone without touching either of them, or anything else.

```
from bugbot import *
connect()

while position()[1] < 80:
    forward(50)
    wait(0.1)

stop()
```

The hint students can ask for: Two robots cross your path, one red and one blue. Watch their domes with `robots()` and `wait` or `go round`.

A solution

```
from bugbot import *
connect()
def robot_near():
    return any(r[3] < 30 for r in robots())
while position()[1] < 76:
    if robot_near():
        stop()
    else:
        forward(45)
        wait(0.1)
stop()
```

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