



10.1 A program in a game

What this lesson is about

info(), your own log, and the loop every game program is built from.

- What changes
- The shape of a game program
- Try one

Questions

6 marks in all

1. A student copies the Sprint and Stop program into the task. It stops with `NameError: name 'info' is not defined`. Why? [1 mark]

- ☐ A `info()` only exists in a game, and in a task no game is telling you anything
- ☐ B `info` needs importing separately
- ☐ C The stop box is called something else in tasks
- ☐ D Tasks do not allow while loops

Answer: A. `info()` is how a game talks to you. In a task, work from `position()` instead: the target is 55 cm from the start.

2. Which of these are true of a program in a game? [1 mark]

Tick every answer that is true.

- ☐ A Other robots are on the mat
- ☐ B The game tells you things through `info()`
- ☐ C Everyone in the game can read your `print()` lines
- ☐ D There is no task to pass, just a result
- ☐ E The program keeps running until the game ends

Answer: A, B, D, E. Your log is your own: nobody else sees it, so print as much as you like.

3. Put the four parts of a game loop in order.

[1 mark]

Number the lines 1 to 4 to put them in the right order.

- ___ Look: read the sensors and what the game says
- ___ Wait a moment
- ___ Decide
- ___ Move

Answer:

```
Look: read the sensors and what the game says
Decide
Move
Wait a moment
```

Look, decide, move, wait: the behaviour loop from Module 5, with other robots in the way.

4. What does this program print?

[1 mark]

```
finish_y = 55
for y in [10, 40, 52]:
    gap = finish_y - y
    if gap > 25:
        print(gap, "forward(90)")
    elif gap > 4:
        print(gap, "forward(30)")
    else:
        print(gap, "stop")
```

Answer:

```
45 forward(90)
15 forward(30)
3 stop
```

Far away it drives at 90, closer it creeps at 30, and within 4 cm it stops.

5. In the game loop, why does each pass end with `wait(0.1)` ?

[1 mark]

- ☐ A To let a moment pass, so the robot keeps hearing from you
- ☐ B To make the robot slower than the bots
- ☐ C Because `info()` only updates once a second
- ☐ D The game ends if you do not wait

Answer: A. Without it the loop spins flat out and the robot has no time to act on what you told it.

6. In the task the target is 55 cm up the mat and the red strip starts 5 cm beyond it. `position()[1]` reads 58. How many cm are you from the red strip?

[1 mark]

Answer: 2. The strip starts at $55 + 5 = 60$ cm, and $60 - 58 = 2$ cm. Close, but safe.

The task: stop on the line

Same idea, on your own: drive 55 cm up the mat and stop within 5 cm of that point, without crossing into the red strip five centimetres beyond it. Remember `position()` counts from where the robot started.

```
from bugbot import *
connect()

# position() counts from the start, so the line is 55 cm up from here
forward(70)
wait(1)
stop()
print("stopped at", position())
```

The hint students can ask for: Drive fast while the line is far away, creep when it is close, and remember the robot slides a little after `stop()`.

A solution

```
from bugbot import *
connect()
while position()[1] < 38:
    forward(80)
    wait(0.1)
while position()[1] < 53:
    forward(25)
    wait(0.1)
stop()
print('stopped at', position())
```

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