WORKSHEET

4.2 Where is it?

Vision · Robot club · about 15 min

Name _____ Class _____ Date _____

What this lesson is about

Pixels to degrees, and a controller that turns to face a tag.

- Pixels to degrees
- Turn to face it
- Stopping cleanly

Questions

6 marks in all

1. Using `bearing = (cx - 160) * 120 / 320`, what is the bearing in degrees of a tag at `cx = 240`? [1 mark]

2. Using `bearing = (cx - 160) * 120 / 320`, where is a tag at `cx = 100`? [1 mark]

- ☐ **A** To the left, about 22 degrees
- ☐ **B** To the right, about 22 degrees
- ☐ **C** Straight ahead
- ☐ **D** To the left, about 100 degrees

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

```
def bearing_of(cx):
    return (cx - 160) * 120 / 320

for cx in [0, 160, 320]:
    print(bearing_of(cx))
```

4. The loop runs `drive(0, 0, bearing_of(cx) * 3)` and the tag is at `cx = 220`. What does the robot do? [1 mark]

- ☐ **A** Turns clockwise, towards the tag
- ☐ **B** Turns anticlockwise, away from the tag
- ☐ **C** Drives forward
- ☐ **D** Slides right without turning

5. A student writes `error = (160 - cx) * 120 / 320` and uses `drive(0, 0, error * 3)`. What happens? [1 mark]

- ☐ **A** The robot turns away from the tag until it loses it
- ☐ **B** The robot faces the tag, just more slowly
- ☐ **C** Nothing different, because the sign does not matter
- ☐ **D** The robot drives towards the tag

6. Once the error is under 1.5 degrees, the program stops, waits 0.3 s and checks again. Why check again? [1 mark]

- ☐ **A** The robot coasts after stopping, so the tag may have slid off centre
- ☐ **B** The camera needs 0.3 s to switch on
- ☐ **C** The tag might have changed its id
- ☐ **D** To let the battery recover

The task: face the marker

Turn until marker 3 is in the middle of the picture, then print `centred`. Do not drive.

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

# camera: the tag detector
set_cv("apriltag")
print(apriltags())
```

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

Do it on the robot

www.bugbotlab.com/learn/4-2-where-is-it/

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

Challenges

1. Print the heading before and after, and compare the change with the bearing you computed.
2. Face the marker, then turn 180 away from it and face it again.
3. Work out the bearing exactly with `math.degrees(math.atan((cx - 160) / 92.4))` (after `import math`). What does it give for marker 3? Compare the two formulas at cx 160, 200, 240, 280 and 320: where is the difference biggest?