



4.4 Colour

What this lesson is about

The blob detector: one colour at a time, area and box width as distance.

- Pick a colour
- Which side?
- How far?
- Follow the colour

Questions

7 marks in all

1. A blob is `[cx, cy, area, x0, y0, x1, y1, aspect]`. What is `blob[2]`? [1 mark]

- ☐ A The area, in pixels
- ☐ B The distance, in cm
- ☐ C The left edge of its box
- ☐ D Its colour

Answer: A. Index 2 is the area: how many pixels it covers. There is no distance in a blob.

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

```
blob = [220, 110, 900, 205, 95, 235, 125, 1.0]
width = blob[5] - blob[3]
print(width)
print("left" if blob[0] < 160 else "right")
```

Answer:

```
30
right
```

The box width is `x1 - x0`, `235 - 205 = 30`. `cx` is 220, over 160, so the blob is on the right.

3. The program has read the red blobs. How does it read the blue ones? [1 mark]

- ☐ A Call `set_cv("blob", "blue")`, then `blobs()` again
- ☐ B Call `blobs("blue")`
- ☐ C Look further down the list from `blobs()`
- ☐ D Call `set_cv("blue")`

Answer: A. The blob detector looks for one colour at a time. Switch colour with `set_cv`, then read again.

4. Why is the box width a better guide to distance than the area? [1 mark]
- ☐ A Area grows with the square of the size, so it changes much faster
 - ☐ B Area is always 0 for a ball
 - ☐ C Width is measured in cm
 - ☐ D Width does not change as the ball gets closer

Answer: A. Twice as close looks twice as wide but four times the area. Width goes up steadily, which is easier to use.

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

```
red = [100, 120, 400, 90, 110, 110, 130, 1.0]
blue = [250, 118, 1600, 230, 98, 270, 138, 1.0]
balls = {"red": red, "blue": blue}
nearer = max(balls, key=lambda name: balls[name][5] - balls[name][3])
print(nearer, "is nearer")
```

Answer:

blue is nearer

Red is 20 pixels wide and blue is 40. Same size ball, looks bigger, so blue is the nearer one.

6. A blob's box is 60 pixels wide and 20 pixels high. What is its aspect ? [1 mark]

Answer: 3. Aspect is width divided by height: $60 / 20 = 3$. A ball is about 1, so this looks more like a stick lying down.

7. Why does a blob have no distance , when a tag does? [1 mark]

- ☐ A The detector does not know how big the red thing really is
- ☐ B The camera cannot measure distance at all
- ☐ C Colours are always far away
- ☐ D You have to call `distance()` inside `blobs()`

Answer: A. A tag has a known printed size, so the detector can work out distance. For a blob, only you know the real size.

The task: red or blue

Without driving, print `left: <colour>` and `right: <colour>` for the two balls.

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

# camera: look for red blobs
set_cv("blob", "red")
print(blobs())
```

The hint students can ask for: Look for each colour in turn and use `cx` to say which side it is on: print `left: <colour>` and `right: <colour>`. Do not drive.

A solution

```
from bugbot import *
connect()
set_cv('blob', 'red')
red = blobs()[0]
set_cv('blob', 'blue')
blue = blobs()[0]
if red[0] < blue[0]:
    print('left: red')
    print('right: blue')
else:
    print('left: blue')
    print('right: red')
```

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