



5.6 Project: rescue

Name _____

Class _____

Date _____

What this lesson is about

Search, approach, return: camera, servoing and states in one machine.

- The plan, drawn
- Search and approach
- Home
- What you have built
- Where next

Questions

7 marks in all

1. Put the rescue states in the order the robot goes through them.

[1 mark]

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

- ___ home
- ___ done
- ___ search
- ___ approach

2. Which condition moves the machine from search to approach?

[1 mark]

- ☐ A Tag 9 is in view
- ☐ B The robot is closer than 12 cm
- ☐ C The robot is near the start
- ☐ D The loop has run 300 ticks

3. Why does the code say `if tags and tags[0][3] < 12` rather than just `if tags[0][3] < 12`?

[1 mark]

- ☐ A If the list is empty, `tags[0]` would be an error, and `and` stops before reading it
- ☐ B `and` makes the test run faster
- ☐ C It checks the tag is number 9
- ☐ D Without it the robot would never stop

4. What does this program print?

[1 mark]

```
def step(state, tag_dist, at_home):
    if state == "search" and tag_dist is not None:
        return "approach"
    if state == "approach" and tag_dist is not None and tag_dist < 12:
        return "home"
    if state == "home" and at_home:
        return "done"
    return state

state = "search"
for tag_dist, at_home in [(None, False), (60, False), (30, False), (11, False), (None, False), (None, True)]:
    state = step(state, tag_dist, at_home)
    print(state)
```

5. How does the home state know where home is?

[1 mark]

- ☐ A Positions are relative to the start, so home is (0, 0)
- ☐ B It searches for a home tag
- ☐ C It reverses every move it made
- ☐ D It follows the distance sensor back

6. The task starter stops at the marker with `break`. What turns it into the full rescue?

[1 mark]

- ☐ A Set `state = "home"` instead of breaking, and add an `elif state == "home":` branch
- ☐ B Add a second loop after the first one that searches again
- ☐ C Change the 12 to 0
- ☐ D Call `set_cv("apriltag")` again at the marker

7. The rescue ends with `print("home:", position(), "after", tick / 10, "seconds")`. If the loop ended on tick 437, how many seconds does it print? Give the exact number.

[1 mark]

The task: rescue

Find marker 9, print `found 9`, drive to it, and return home, without touching the box.

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

# maths: atan2, hypot, sin, cos, radians
import math

def wrapped(h):
    return (h + 180) % 360 - 180

def go_to(x, y, speed=70):
    # where am I?
    px, py = position()
    a = math.radians(wrapped(math.degrees(math.atan2(x - px, y - py)) - heading()))
    # forward, sideways, rotation: -100 to 100 each, until the next command
    drive(speed * math.cos(a), speed * math.sin(a), wrapped(0 - heading()) * 3)

def near(x, y, cm=6):
    # where am I?
    px, py = position()
    return math.hypot(x - px, y - py) < cm

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

# camera: the tag detector
set_cv('apriltag')
state = 'search'
# do this 590 times (tick counts from 0)
for tick in range(590):
    tags = [t for t in apriltags() if t[0] == 9]
    if state == 'search':
        if tags:
            print('found 9')
            state = 'approach'
        else:
            # spin clockwise on the spot at 40
            turn_right(40)
    elif state == 'approach':
        if tags and tags[0][3] < 12:
            # leave the loop
            break
        elif tags:
            # forward, sideways, rotation: -100 to 100 each, until the next command
            drive(60, 0, bearing_of(tags[0][1]) * 3)
        else:
            # spin clockwise on the spot at 30
            turn_right(30)
    # pause 0.1 s (the robot keeps doing what it was told)
    wait(0.1)
# all motors off
stop()
```

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



Do it on the robot

www.bugbotlab.com/learn/5-6-project-rescue/

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

Challenges

1. Add an "avoid" layer from lesson 5.4 above all three states, and move the box into the way.
2. Rescue two markers, 9 and then 8 (add one to the scene in the free-play editor), and come home once.
3. Blink the LED in a different pattern in each state, so a person can tell what the robot is thinking.