04_04_escape_real_crazyflie_student_hs

May 2, 2025

1 Escape the Room – Student Version

Your challenge is to write a program that lets the Crazyflie escape from a room.

Use sensor data to detect walls, avoid obstacles, and find an open exit.

1.1 Helpful Commands Reference

Command	Description	Parameters (units/type)
`takeoff()`	Take off and hover	height (m/float), speed (m/s/float)
`land()`	Land the drone	speed (m/s/float)
`forward()`	Move forward	distance (m/float), speed (m/s/float)
`left()` / `right()`	Move sideways (strafe)	distance (m/float), speed (m/s/float)
`rotate()`	Rotate (yaw)	angle (degrees/int), duration (s/float)
`get_distances()`	Read all sensor values	None
`get_distances()['front']`	Distance to object in front	meters/float
`get_distances()['left']`	Distance to object on the left	meters/float
`get_distances()['right']`	Distance to object on the right	meters/float
`get_yaw()`	Get the yaw rotation of the drone	None
`get_status()`	Check if drone is flying or landed	None

[ ]:
```
#  Setup your drone simulator
from crazyflie_sim import CrazyflieSimulator
import time
drone = CrazyflieSimulator(real=False)
```

1.2 Objective

Write code that takes off, navigates around walls and obstacles, and escapes from a room with an opening.

- Use loops and conditionals
- Check distances in multiple directions
- Use print statements to debug your decisions
- Don't crash into walls!

Hint: Use `if` statements and `get_distances()` to decide where to go. Hint: `rotate()` can help you face a different direction if you're blocked.

[ ]:
```
#  Your code goes here:
```

1.3 Exercise 1: Try escaping with fewer steps by checking for the largest open space first

[ ]:
```

```

1.4 Exercise 2: Stop immediately if all sides are blocked (trap logic)

[ ]:
```

```