04_04_escape_real_crazyflie_teacher_

May 2, 2025

1 Escape the Room: Real Crazyflie Navigation

In this version of the Escape the Room challenge, we'll assume you're using a **real Crazyflie** equipped with the **MultiRanger deck**. Your goal: write logic to help your drone avoid walls and exit the room using sensor data only — no visual rendering.

1.1 Scenario

- You are flying in a real room, approximately **2m x 2m**.
- The drone starts in the center.
- There's a door/opening on the front wall.
- You must write a navigation loop to detect open space and escape.

[ ]:
```
#  Setup for Real Crazyflie
import time
from crazyflie_sim import CrazyflieSimulator

# Set real=True for hardware mode
drone = CrazyflieSimulator(real=True)
```

1.2 Step 1: Take Off

[ ]:
```
drone.takeoff(1.0, 0.3)
time.sleep(2)
```

1.3 Step 2: Read the Sensors

The drone will use MultiRanger to measure distance to walls: - `front`, `left`, `right`, `back`, `up` - Units are in **meters** - Less than 0.5 meters = obstacle close!

[ ]:
```
distances = drone.get_distances()
print(distances)  # You should see something like: {'front': 0.6, 'left': 0.4, .
  ↪..}
```

1.4 Step 3: Escape Logic

Keep moving forward if clear, otherwise rotate or back up.

[ ]:
```
# Basic escape loop
steps = 0
while steps < 15:
    d = drone.get_distances()
    print(f"Step {steps} | Sensors: {d}")

    if d['front'] > 0.5:
        drone.forward(0.2, 0.2)
    elif d['left'] > 0.5:
        drone.rotate(90, 1)
    elif d['right'] > 0.5:
        drone.rotate(-90, 1)
    else:
        drone.backward(0.2, 0.2)

    time.sleep(1)
    steps += 1
```

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

[ ]:
```
# d = drone.get_distances()
# # Implement your own logic to choose the best direction
# # e.g., if d['right'] is most open, rotate right and go
# ...
```

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

[ ]:
```
# d = drone.get_distances()
# if min(d['front'], d['left'], d['right'], d['back']) < 0.3:
#     print("Trapped! Emergency landing.")
#     drone.land()
#     exit()
```

1.7 Step 4: Land and Disconnect

[ ]:
```
drone.land(0.3)
drone.close()
```