

04_02_obstacles__

May 2, 2025

Part of the InnovatED STEM and DroneBlocks Land, Air, and Sea Robotics Curriculum
Licensed for educational use in schools only.
Redistribution, commercial use, or resale is strictly prohibited.
© 2025 InnovatED STEM & DroneBlocks. All rights reserved.

1 Simple Obstacle Avoidance with MultiRanger

This notebook teaches how to combine sensor input and logic to help your drone avoid obstacles automatically using `while`, `if`, and sensor readings.

Obstacle detection is simulated using the `CrazyflieSimulator` with realistic distance values.

1.1 Command Reference Table

Command	Description	Parameters (units/type)
<code>takeoff()</code>	Drone takes off and hovers	height (m/float), speed (m/s/float)
<code>land()</code>	Lands the drone	speed (m/s/float)
<code>forward()</code>	Move forward	distance (m/float), speed (m/s/float)
<code>rotate()</code>	Rotate the drone left or right	angle (degrees/int), time (s/float)
<code>get_distances()</code>	Returns dictionary of sensor distances	None
<code>get_distances()['Distance']</code>	Distance to object in front (m/float)	None
<code>get_distances()['Left']</code>	Distance to the left (m/float)	None
<code>get_distances()['Right']</code>	Distance to the right (m/float)	None

```
[ ]: # Setup
import time
from crazyflie_sim import CrazyflieSimulator

drone = CrazyflieSimulator(real=False)
```

1.2 Takeoff

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

1.3 Obstacle Avoidance Logic

The idea is simple: - If there's something **in front** ($< 0.5\text{m}$) \rightarrow stop and move **back** or **turn** -
Otherwise \rightarrow keep moving **forward**

```
[ ]: print(drone.get_distances()) # See example sensor values
```

```
[ ]: # Simple forward motion with obstacle check
     for _ in range(10):
         distances = drone.get_distances()
         if distances['front'] < 0.5:
             print("Obstacle ahead! Backing up.")
             drone.backward(0.2, 0.2)
             drone.rotate(90, 1)
         else:
             drone.forward(0.2, 0.2)
         time.sleep(1)
```

1.4 Exercise 1: Try using left/right sensors to dodge obstacles

Hint: If there's an object in front and space on one side, turn that way instead of backing up.

```
[ ]:
```

1.5 Exercise 2: Loop until you detect an obstacle within 0.4m in any direction

```
[ ]:
```

1.6 Land and Close

```
[ ]:
```

1.7 Exercise 3: Turn around if blocked on both front and sides

Use logic to rotate 180° if there's no space in front, left, or right.

```
[ ]:
```

1.8 Exercise 4: Cautiously move forward until object is closer than 0.6m

Use a loop and stop at safe distance.

```
[ ]:
```

1.9 Exercise 5: Try a mini square pattern, only if no object ahead

[]: