PRESENTATION 4 · LEARNING WITH PYTHON

Simple maze

Keep one hand on the wall: following a wall with the ultrasonic sensors, step by step.

TheBotCup 2026 · AiBotCup Simulator



REMEMBER FROM PRESENTATION 3

Three ideas we need today

Three ears

US_Front ahead, **US_Left** and **US_Right** at 45°.

A comparison = a question

US_Front <= 25: “Is a wall close ahead?”

Order matters

The first true branch wins. The most urgent goes on top.

PART 1

The mission: Session 4

Find the way from START to FINISH

SESSION 4 · SIMPLE MAZE

Through the maze to FINISH

4

Session 4 · Simple maze

Find the way through the maze from START (bottom left) to the pink FINISH pad (top right).

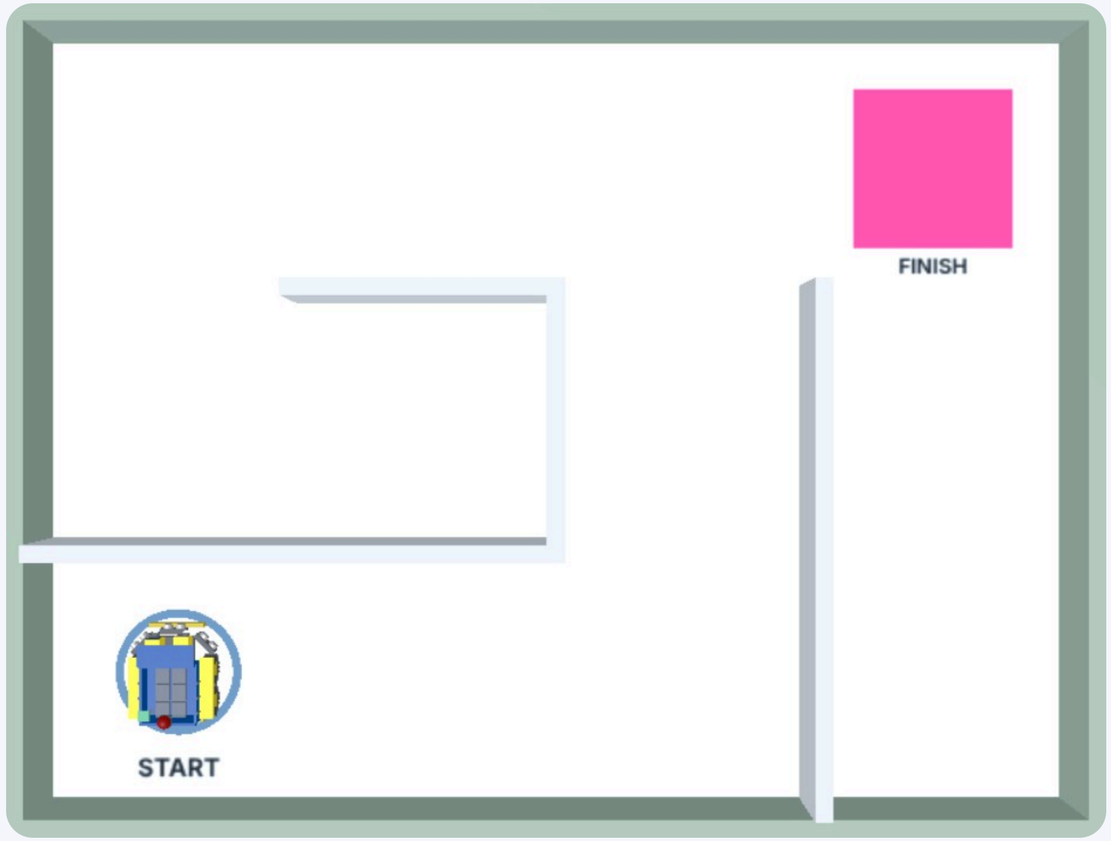
Reach FINISH: +150. Time limit 5:00.

Not done yet

START bottom left, facing **up**, a wall right in front.

FINISH pink pad, top right: **+150**.

Field 240 × 180 cm: **4 × 3 cells** of 60 cm. Time limit **5:00**.



MAZE WORDS

Four words to talk about mazes

Cell

One square of the grid, here **60 × 60 cm**.

Corridor

A path between two walls.

Junction

A place where you can choose a way.

Dead end

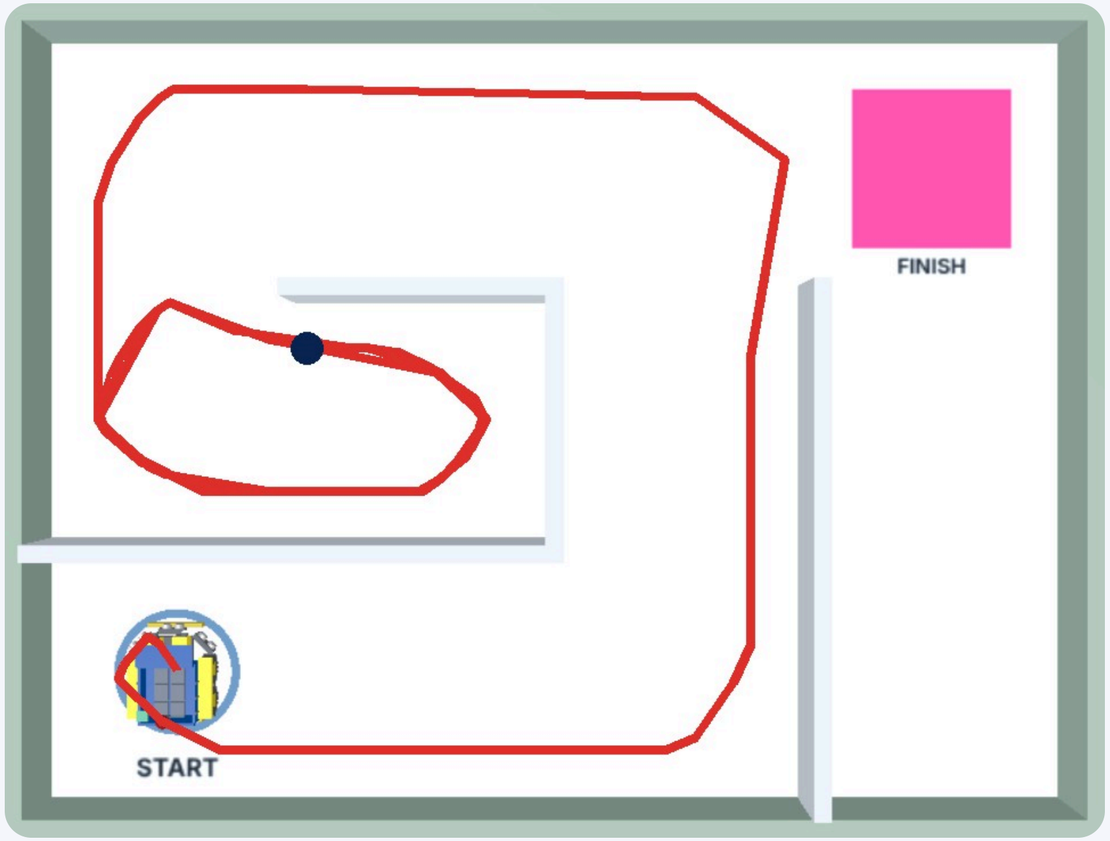
A corridor with no way out: turn around.

The robot does **not** know the maze. It only hears what is near it.

PART 2

How do you solve a maze?

A simple idea, its problem, and a better rule



FIRST IDEA · TESTED

“Wall ahead? Spin left. Else forward.”

Two branches, if and else, like Session 3.

The robot wanders, then gets **stuck going round** in the dead end.

After 1 minute: still **0 points**. It needs a plan for the whole maze.

THE RIGHT-HAND RULE

Keep your right hand on the wall, and never let go.

Wall ends?

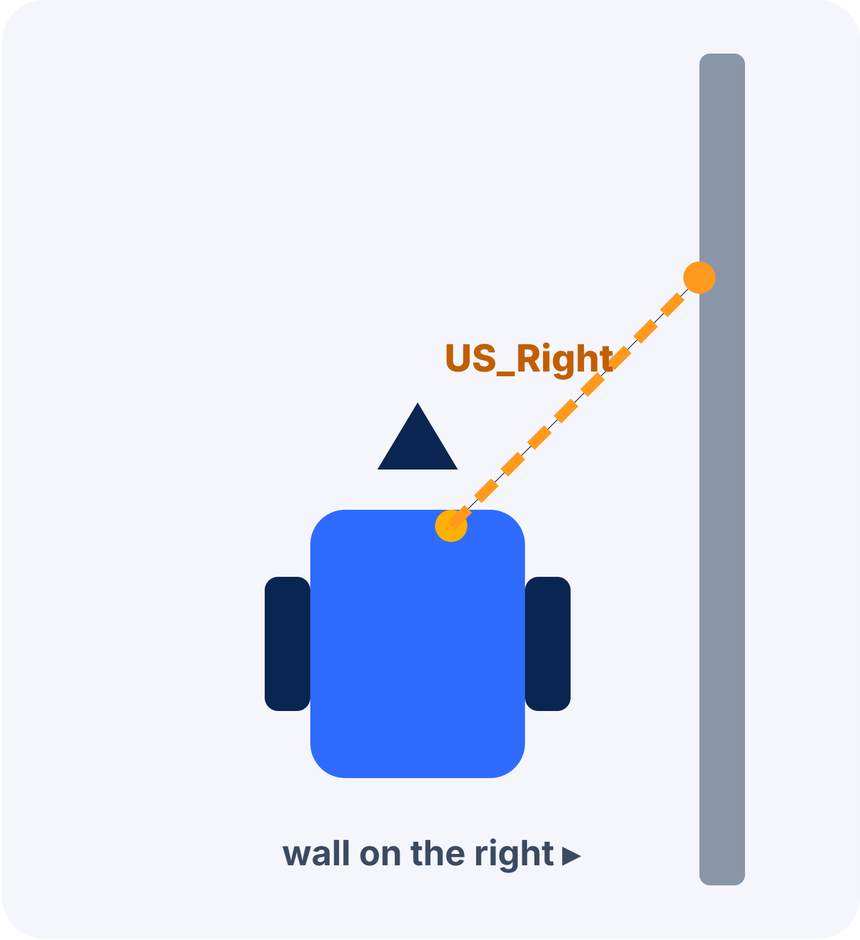
Turn right around it.

Wall in front?

Turn left, hand still on the wall.

Dead end?

Follow the wall around and come back out.



THE “HAND” IS US_RIGHT

US_Right tells how far the right wall is

0 – 22 · too close: steer **left**

23 – 44 · good: drive **straight** (in the middle of a corridor it reads about 31)

45 – 255 · lost the wall: steer **right** to find it again

FROM RULE TO CODE

Four situations, four moves

1 · AHEAD

Wall in front

US_Front <= 20 → **spin left** (−2, 2)

2 · TOO CLOSE

Right wall near

US_Right <= 22 → **curve left** (2.5, 4.5)

3 · LOST

Right wall gone

US_Right >= 45 → **curve right** (4.5, 1.5)

4 · FORWARD

Everything else

else → **straight** (4.5, 4.5)

Curves keep moving forward: one wheel slower than the other (Presentation 1).

PART 3

Step by step in Python

Write · order · build · run

STEP 1 · WRITE

The whole program: four branches

```
# Session 4 · Simple maze

def game1():
    if US_Front <= 20:                          # Ahead
        WheelLeft = -2; WheelRight = 2
    elif US_Right <= 22:                        # TooClose
        WheelLeft = 2.5; WheelRight = 4.5
    elif US_Right >= 45:                        # Lost
        WheelLeft = 4.5; WheelRight = 1.5
    else:                                       # Forward
        WheelLeft = 4.5; WheelRight = 4.5
```

Ahead US_Front <= 20 → −2, 2

TooClose US_Right <= 22 → 2.5, 4.5

Lost US_Right >= 45 → 4.5, 1.5

Forward else → 4.5, 4.5

No Time, no Position: it works on **any** maze like this.

STEP 2 · ORDER

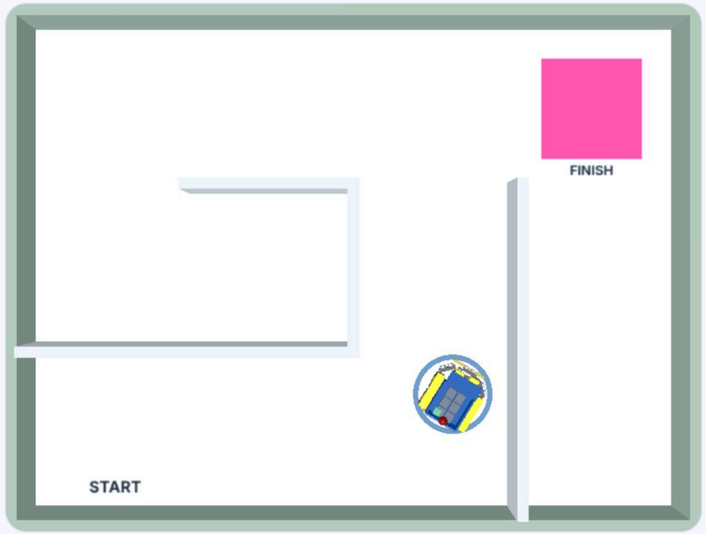
Most urgent on top

1st
Ahead
A crash is worst: check first.

2nd
TooClose
Don’t scrape the side.

3rd
Lost
Find the wall again.

4th
Forward
The else: everything else.





At 9 s: Front **28** (not <= 20), Right **11** → **TooClose** wins: curve left.

STEP 3 · CHECK

Read it like the robot: from the top

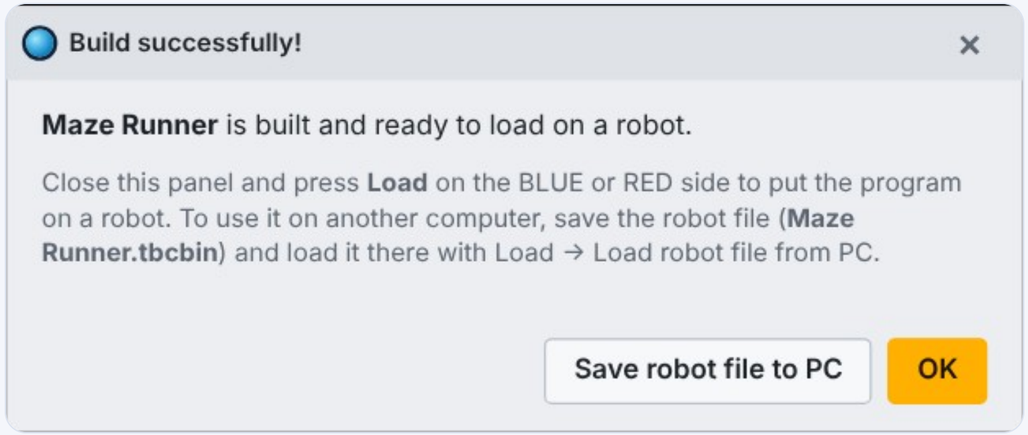
US_Front	US_Right	First true branch	Move
<= 20	any	Ahead	spin left · −2, 2
> 20	<= 22	TooClose	curve left · 2.5, 4.5
> 20	23 – 44	Forward · else	straight · 4.5, 4.5
> 20	>= 45	Lost	curve right · 4.5, 1.5

Between 23 and 44 cm the robot is in the good band: straight on.

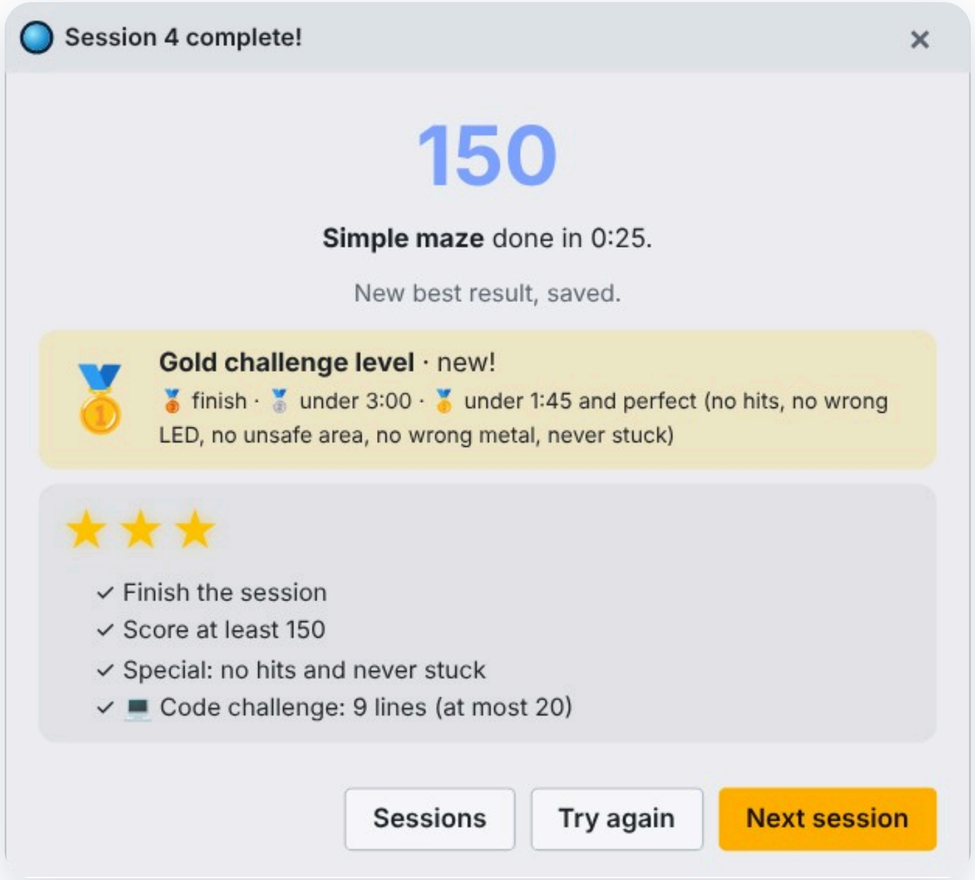
STEP 4 · TEST

Build, save, load on BLUE, play

1 Build → OK → **2 Save to PC**: we reuse it next time! → **3** BLUE **Load** → Maze Runner → **4 ▶ Play**



Use the **Top** camera and watch the robot keep the wall on its right. Open **Debug** to see US_Right go up and down.



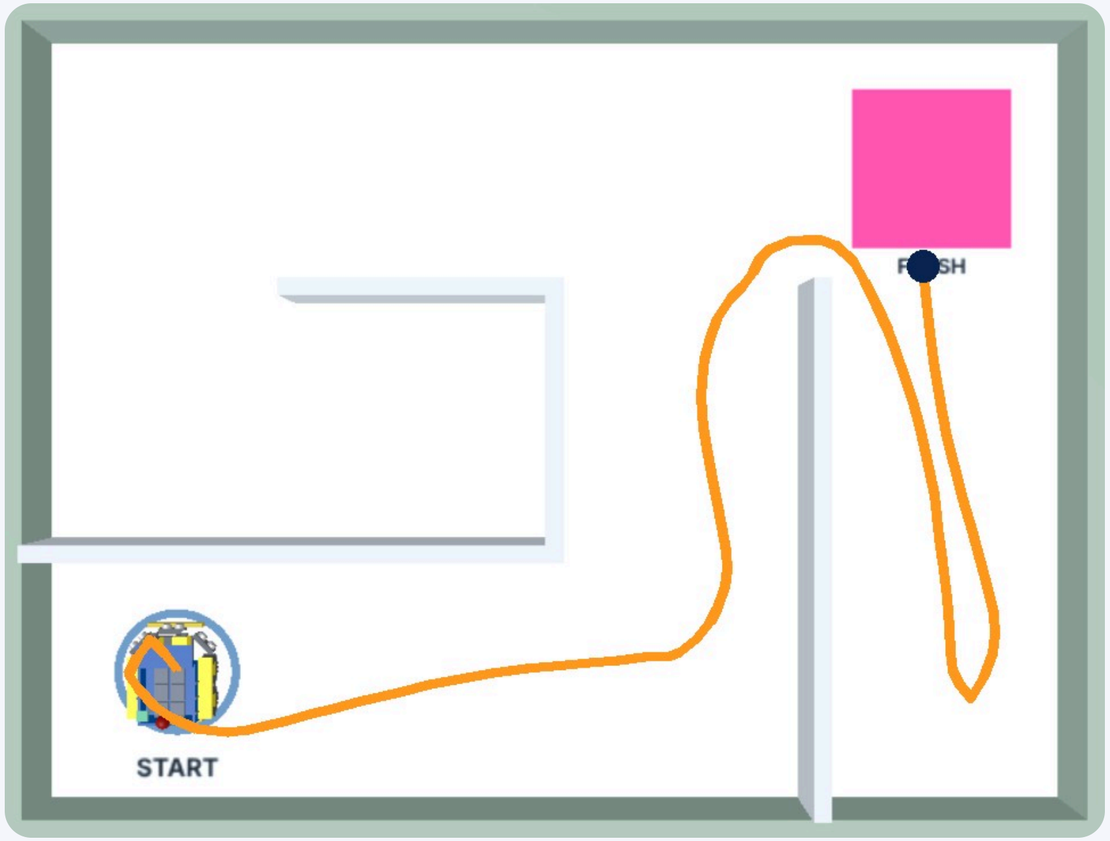
RESULT · TESTED

150 points in 0:25 · Gold

Medals · Bronze: finish · Silver: under 3:00 · Gold: under 1:45 and perfect.

Stars · finish · score 150 · no hits and never stuck.

Code challenge · at most 20 lines. We used 9.



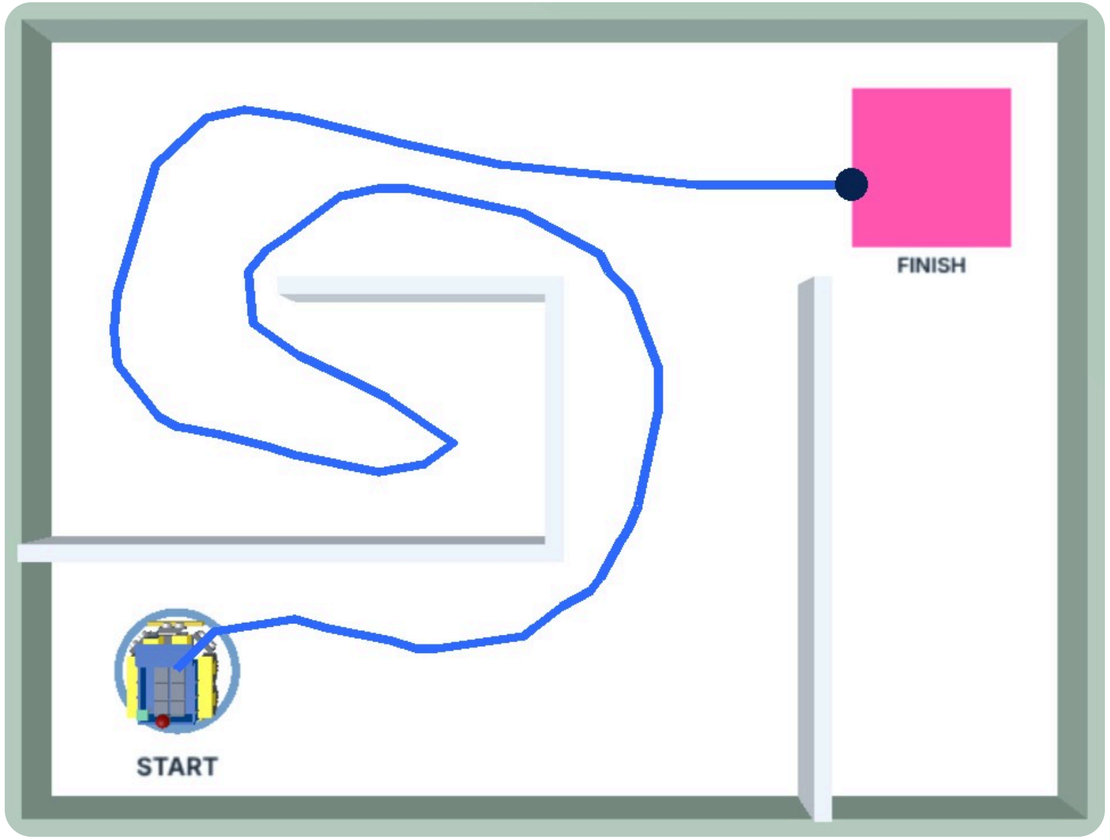
THE PATH · RECORDED

Follow the orange line

Spin left at START (wall ahead), then right along the bottom.

Up and over the long wall.

The right wall leads **down into a pocket**: the robot turns around and comes back up to FINISH.



CHALLENGE · TESTED

The left-hand rule works too

Mirror everything: US_**Left**, spin **right** (2, −2), curves (4.5, 2.5) and (1.5, 4.5).

It visits the **dead end** and comes back out.

Right hand: **0:25** · Left hand: **0:31**. Both 150 points, no hits.

DID NOT WORK?

Check the four branches

What you see	What to do
Build error with a line number	Colon : and indentation
Drives into a wall in front	Ahead must be first: **US_Front <= 20**
Scrapes along the right wall	TooClose: **US_Right <= 22**, wheels 2.5 and 4.5
Wobbles left and right	Keep a gap: TooClose ends at 22, Lost starts at 45
Never turns at wall ends	Lost: wheels 4.5 and 1.5 (right wheel slower)
Goes in circles	Use the same side everywhere: US_Right with the right-hand moves

NEXT TIME

Presentation 5 · Hard maze

30 cells, narrower corridors, more dead ends. **Will our four branches still work?**

Bring your Maze Runner file from Save to PC.