PRESENTATION 3 · LEARNING WITH PYTHON

Avoid the walls

Ultrasonic sensors: how they hear walls, how to use them in Python, and Session 3 step by step.

TheBotCup 2026 · AiBotCup Simulator



SO FAR

Timelines work, until the world is different

Presentation 1

Python code, two wheels, the **Time** sensor.

Presentation 2

Position and **Compass**; small turn errors add up.

Today

The robot **senses the walls** and decides for itself when to turn.

PART 1

The mission: Session 3

Two walls · one FINISH pad · no touching

SESSION 3 · AVOID THE WALLS

Steer around both walls

3

Session 3 · Avoid the walls

Steer around both walls and stop on the pink FINISH pad.

Each wall passed without touching anything: +50. Each hit: −10. Reach FINISH: +100. Time limit 4:00.

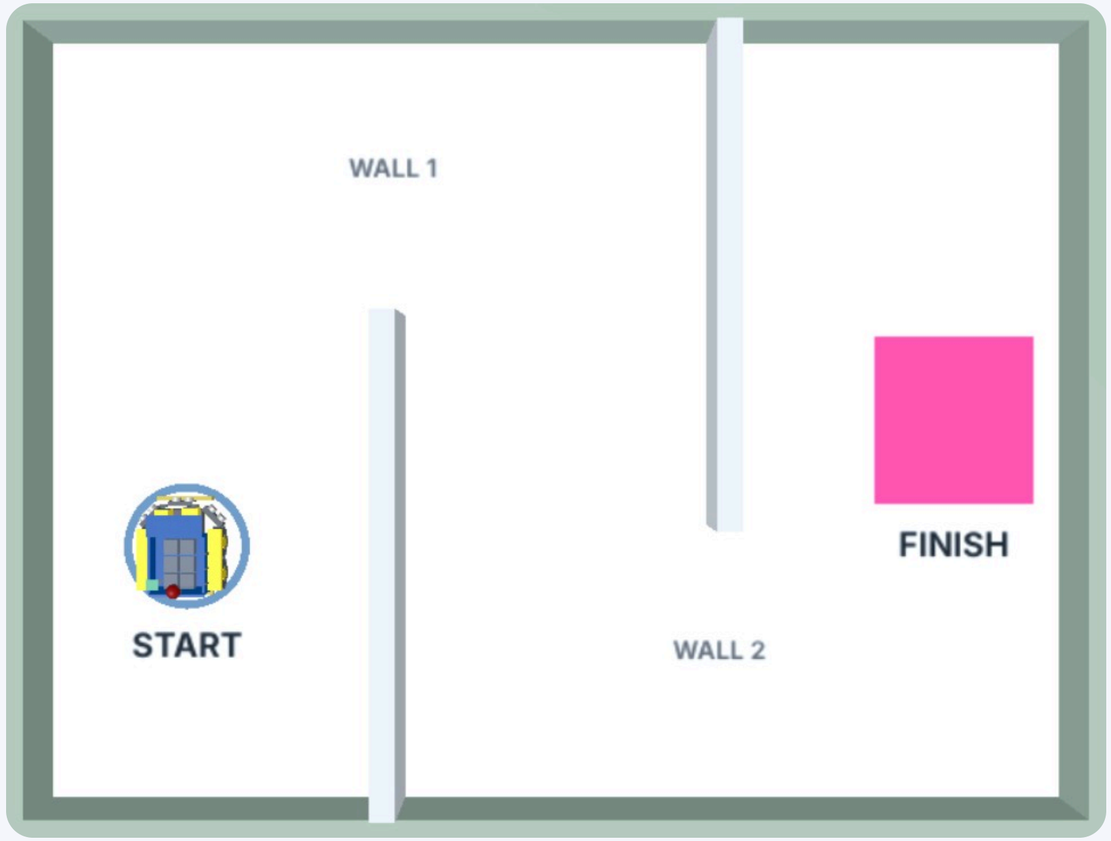
Not done yet

START bottom left, facing **up**.

Wall 1 grows up from the bottom: the gap is at the **top**.

Wall 2 hangs down from the top: the gap is at the **bottom**.

FINISH pink pad on the right.



POINTS

200 points for a clean run

+50

Past **wall 1** without touching anything

+50

Past **wall 2** without touching anything

+100

Reach the **FINISH** pad

−10

Every time the robot **hits** a wall

Time limit **4:00**. Touch a wall and you also lose the +50 for passing it.

PART 2

What is an ultrasonic sensor?

Seeing with sound, like a bat

ULTRASONIC SENSOR

Send a sound, listen for the echo



Too high to hear

“Ultra-sonic” = above what people hear. Many sensors use 40,000 vibrations a second.

Bounces back

A hard surface reflects the sound, like a ball off a wall.

Time = distance

A longer wait for the echo means the wall is farther away.

ULTRASONIC SENSOR

From echo time to centimetres

distance = speed of sound × time ÷ 2

÷ 2 because the sound goes **there and back**.

Speed of sound

About 343 m/s in air = **34.3 cm every millisecond**.

Echo after 2 ms

34.3 × 2 ÷ 2 = **34 cm** to the wall.

Wall 1 m away

The echo takes about **5.8 ms**: fast!

ULTRASONIC SENSOR

Real robot and simulator

Real sensor

The sound spreads out like a **cone**.

Soft or slanted surfaces may send back a weak echo.

Readings can jump a little.

In the simulator

One thin, straight **line** from the sensor.

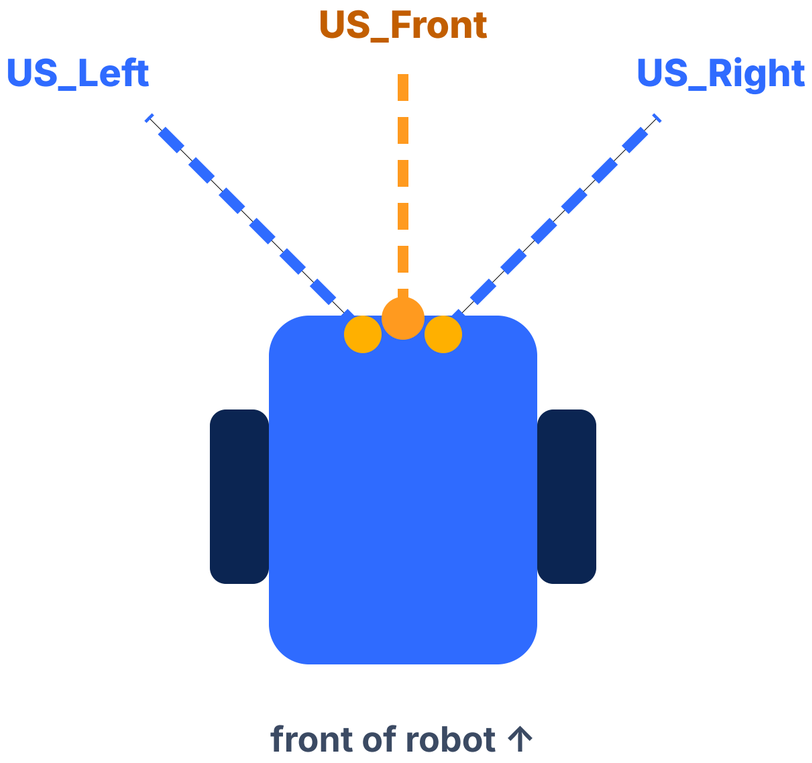
Whole centimetres, **0 to 255**. Nothing ahead = 255.

Sees walls, obstacles and the other robot, **not** colours on the floor.

PART 3

Ultrasonic in the simulator

US_Front · US_Left · US_Right



THREE SENSORS

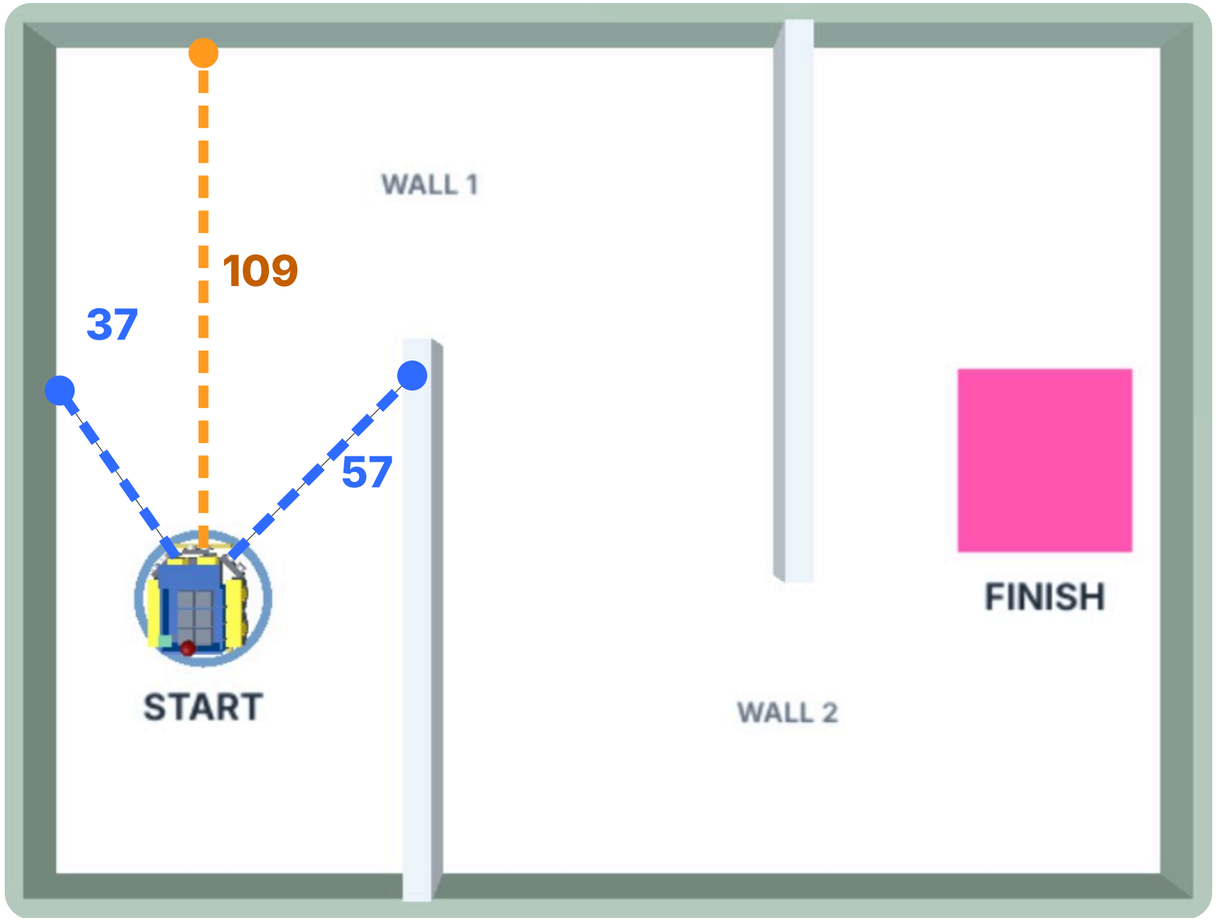
Ahead, front-left and front-right

US_Front: straight ahead.

US_Left: 45° to the left.

US_Right: 45° to the right.

They sit at the **front** of the robot. The robot's centre is about **11 cm** behind US_Front.



AT START

What the robot hears

Ultrasonic Sensors

Front	109	(US_Front)
Left	37	(US_Left)
Right	57	(US_Right)

109: the top wall, far ahead.

37: the left wall.

57: wall 1, front-right.

ULTRASONIC IN PYTHON

A comparison turns a distance into a question

US_Front <= 25

US_Front: “Is a wall **closer than 25 cm** ahead?”

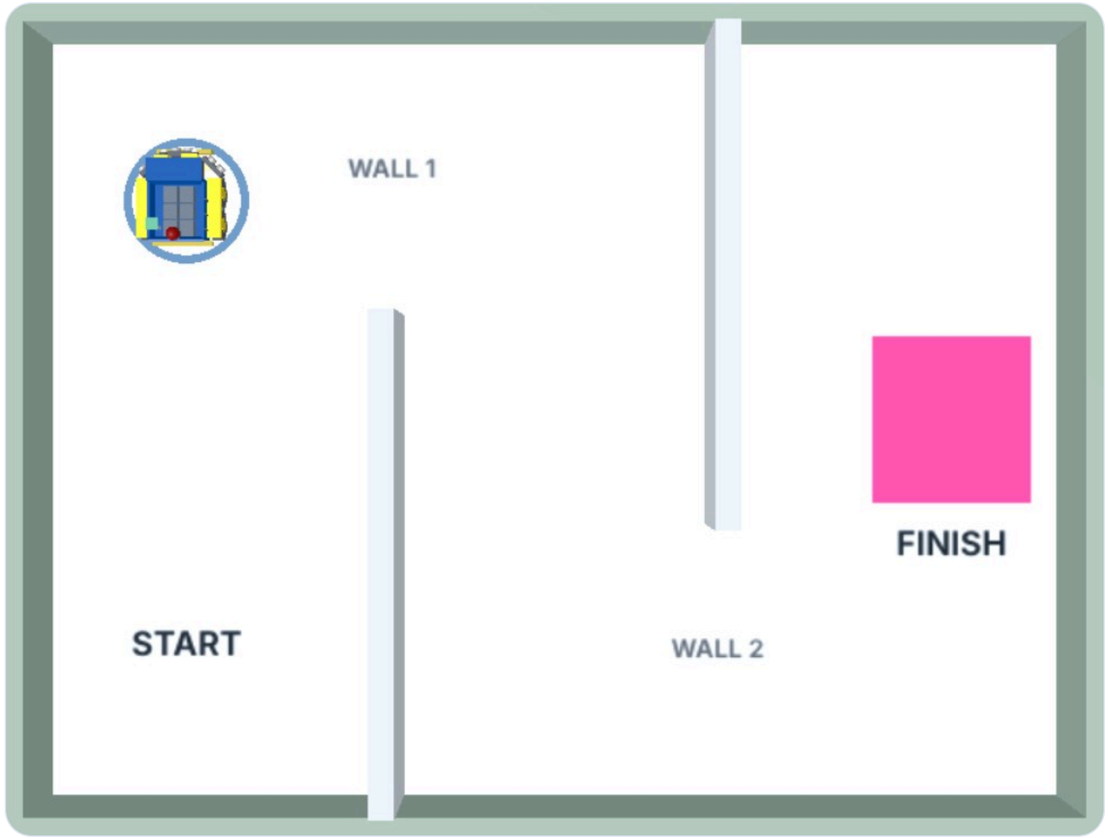
US_Front > 25

US_Front: “Is the way ahead **clear**?”

not used

Not in the condition: this sensor **does not matter**.

One if can ask about **several sensors at once** with and: **all** parts must be true.



CHOOSING THE NUMBER

Why 25 cm?

Ultrasonic Sensors

Front	27	(US_Front)
Left	37	(US_Left)
Right	41	(US_Right)

At 27 cm/s the robot moves only **1.6 cm** between two checks (60 ms).

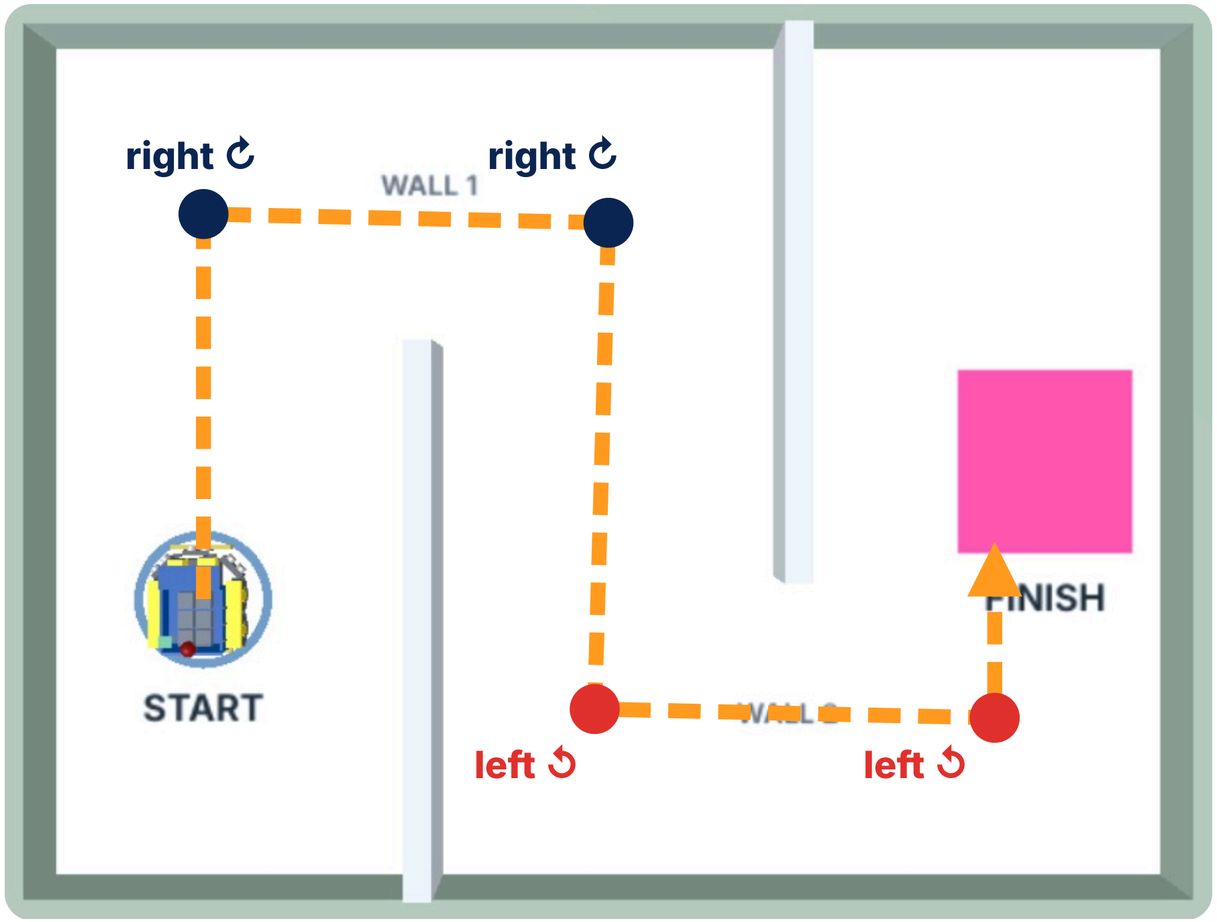
The robot is **22 cm** wide: it needs room to spin.

25 stops it close enough to pass the gaps, far enough not to touch.

PART 4

Solving Session 3

Plan · first idea · better idea · test



STEP 1 · PLAN

Drive until a wall, then turn

Up → right → down → right → up.

Every turn happens when a **wall is ahead**.

At the **top**: two right turns.

At the **bottom**: two left turns.

STEP 2 · FIRST IDEA

“Wall ahead? Turn right.”

IF · TURNRIGHT

WHEN US_Front <= 25
DO spin right

ELSE · FORWARD

WHEN anything else
DO drive 4.5, 4.5

At the top it works: **+50** for wall 1.

At the bottom it turns right, **back towards START**, and drives in circles.

STEP 3 · BETTER IDEA

Two sensors in one if

Near the top

US_Front <= 25 **and** **PositionY** >= 100

→ turn **right**

Near the bottom

US_Front <= 25 **and** **PositionY** <= 99

→ turn **left**

The field is 180 cm tall, so **Y = 100** splits it into top and bottom.

STEP 4 · THE TURN

Duration = 980: finish the whole turn

NO DURATION · TESTED

The wall leaves the front after a **small turn**, so Forward takes over. The robot drives at an angle and hits walls: only **60 points**.

DURATION = 980 · TESTED

The turn keeps going for **0.98 s**: −2 and 2 ≈ 92° a second → about **90°**. Clean run: **200 points**.

The wheels are held until the **Duration** ends, then game1() runs again.

STEP 5 · WRITE

The whole program: three branches

```
# Session 3 · Avoid the walls

def game1():
    if US_Front <= 25 and PositionY >= 100:     # TurnRight
        WheelLeft = 2; WheelRight = -2
        Duration = 980                          # finish the turn
    elif US_Front <= 25 and PositionY <= 99:    # TurnLeft
        WheelLeft = -2; WheelRight = 2
        Duration = 980
    else:                                       # Forward
        WheelLeft = 4.5; WheelRight = 4.5
```

TurnRight wall ahead and top → 2 and −2 for 980 ms

TurnLeft wall ahead and bottom → −2 and 2 for 980 ms

Forward the else → 4.5 and 4.5

No Time anywhere: the robot **reacts** to the walls.

STEP 5 · WHY IT WORKS

The robot asks two questions every 60 ms

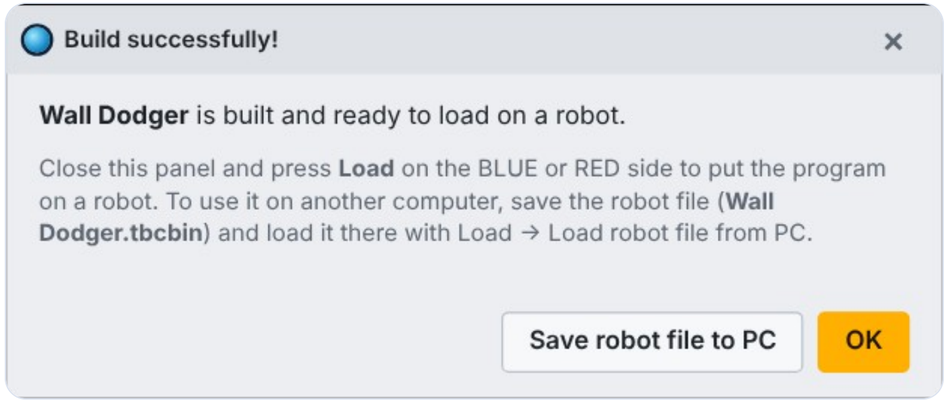
US_Front	PositionY	Branch that runs	Wheels
> 25 (clear)	any	Forward · else	4.5 · 4.5
<= 25 (wall)	>= 100 (top)	TurnRight	2 · −2 for 980 ms
<= 25 (wall)	<= 99 (bottom)	TurnLeft	−2 · 2 for 980 ms

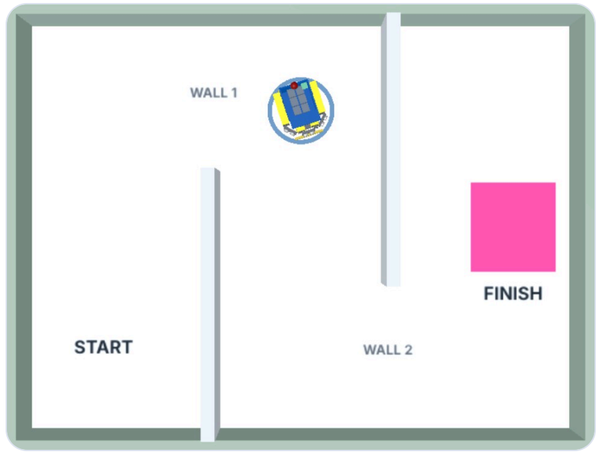
Every case is covered: the robot always knows what to do.

STEP 6 · TEST

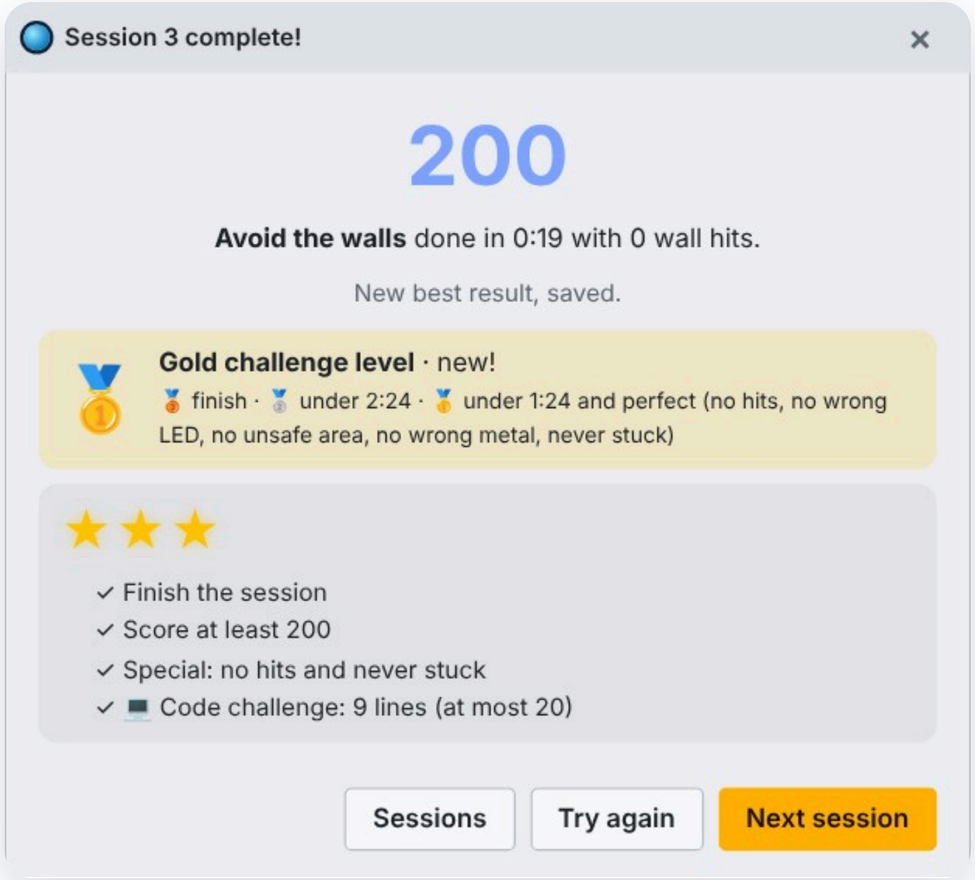
Build, Load on BLUE, Play

1 Build → OK → **2 Save to PC** → **3** BLUE **Load** → Wall Dodger → **4 ▶ Play**





At 8 s the robot has passed wall 1 (**+50**) and turns right in front of wall 2.



RESULT · TESTED

200 points in 0:19 · Gold

Medals · Bronze: finish · Silver: under 2:24 · Gold: under 1:24 and perfect.

Stars · finish · score 200 · no hits and never stuck.

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

DID NOT WORK?

Watch the numbers, change one thing

What you see	What to do
Build error with a line number	Colon : after if / elif / else, and indentation
Drives straight into the top wall	Check **US_Front <= 25** in both turn branches
Never turns at all	The turns must be above the else
Turns a little, then drives at an angle	Each turn needs **Duration = 980**
Turns the wrong way at the bottom	Check the PositionY numbers and the wheel signs
Spins on the spot forever	Forward must be 4.5 and 4.5, both the same

TRY IT YOURSELF

Three challenges

Faster

Try wheels 5 and 5. Do you need a bigger number than 25?

Watch the sensors

Open **Debug** and follow the Ultrasonic graph during the run.

Next session

Presentation 4 · **Simple maze**: follow a wall with a side sensor.