PRESENTATION 2 · LEARNING WITH STATEMENTS

Three targets

Position and Compass sensors, planning a route with four turns, and Session 2 step by step.

TheBotCup 2026 · AiBotCup Simulator



REMEMBER FROM PRESENTATION 1

Four ideas we use again today

Top to bottom

Every 60 ms the first statement that matches runs.

Straight

4.5 and 4.5 = **27 cm** every second.

Spin left

−2 and 2 = about **92°** every second.

Time sensor

Whole seconds since Play. Time 0–5 = 6 seconds.

TODAY

From one turn to four

1

New sensors

Position and Compass.

2

Plan

A route that visits 1, 2, 3 in order.

3

Write & run

Ten statements, then Play.

4

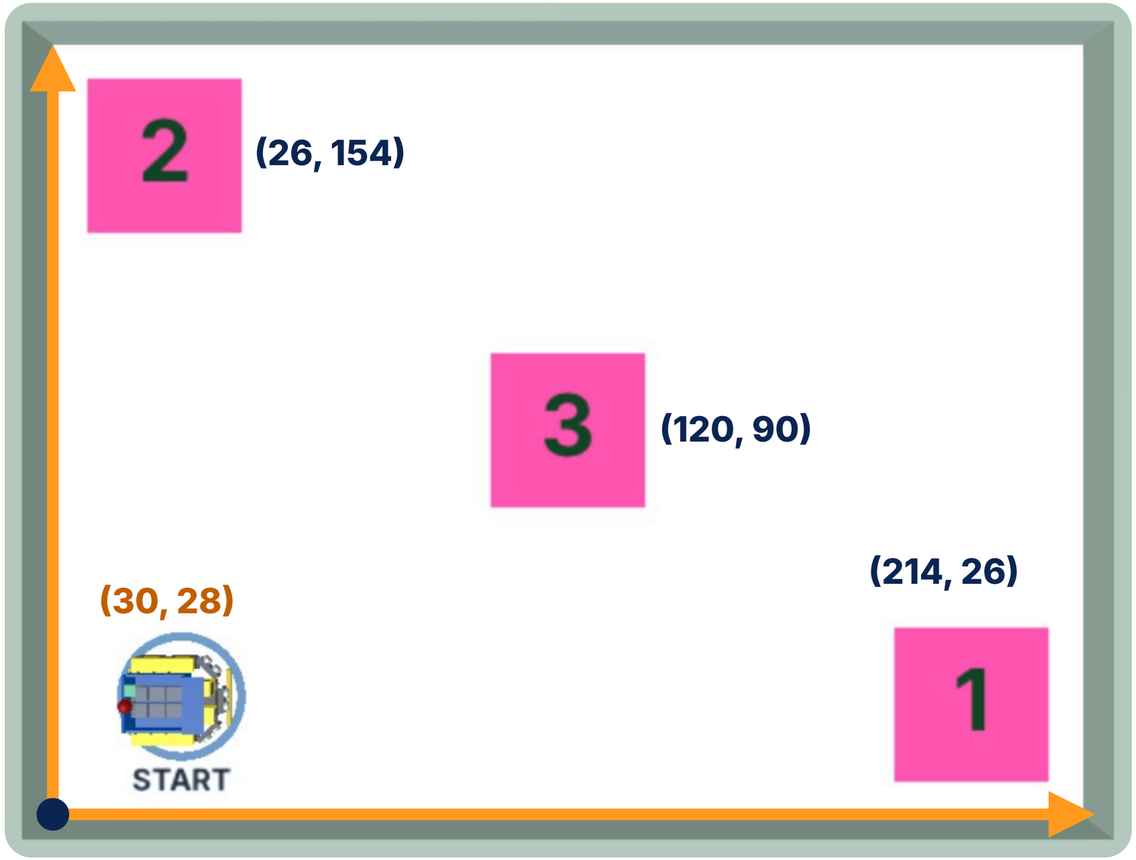
Improve

Why the path bends, and how to fix it.

PART 1

Where am I? Which way?

The Position and Compass sensors



POSITION SENSOR

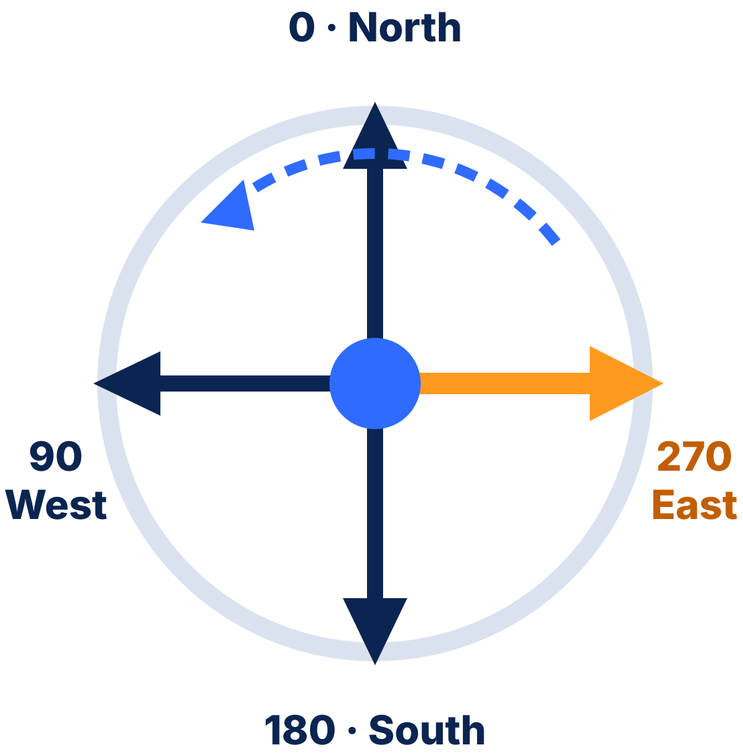
Where is the robot?

PositionX: cm from the left wall, 0 → 240.

PositionY: cm from the bottom wall, 0 → 180.

(0, 0) is the **bottom-left** corner. The robot starts at **(30, 28)**.

Target centres: **1** (214, 26) · **2** (26, 154) · **3** (120, 90). Each target is 36 × 36 cm.



COMPASS SENSOR

Which way is the robot facing?

Compass goes from 0 to 359 degrees.

Up = **0**, left = **90**, down = **180**, right = **270**.

At START the robot faces right: **270**.

Turning **left** makes the number go **up**: 270 → 359 → 0 → 90 → 180.

TRY IT

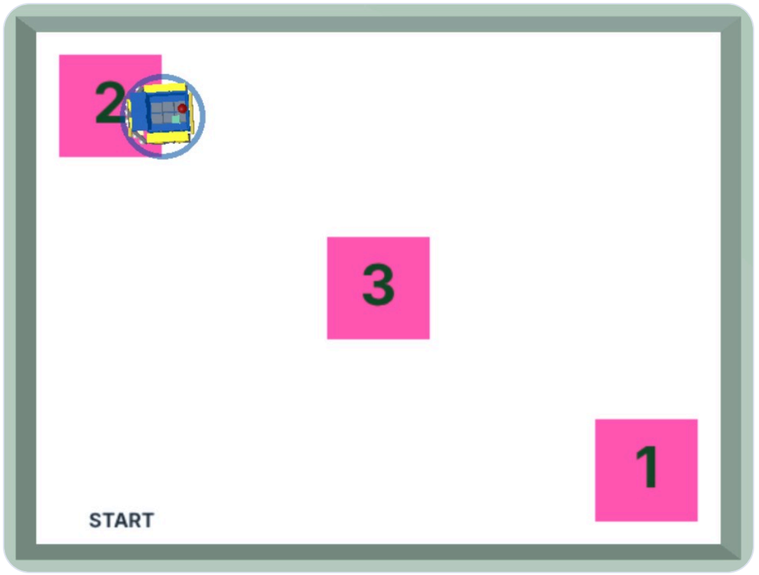
Drive by hand and read the sensors

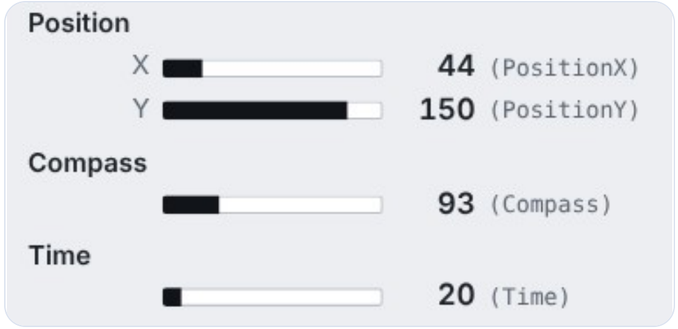
1 Open Session 2. Use the **joystick** or the arrow keys.

2 Drive onto each target. Write down **PositionX**, **PositionY** and **Compass**.

3 Press **Restart session** before running your program.

Driving by hand never counts for the session: it is for exploring.





PART 2

Plan the route

Look · plan · calculate · timeline

STEP 1 · LOOK

The mission

2

Session 2 · Three targets

Visit target 1 (bottom right), then target 2 (top left), then target 3 (centre), in that order.

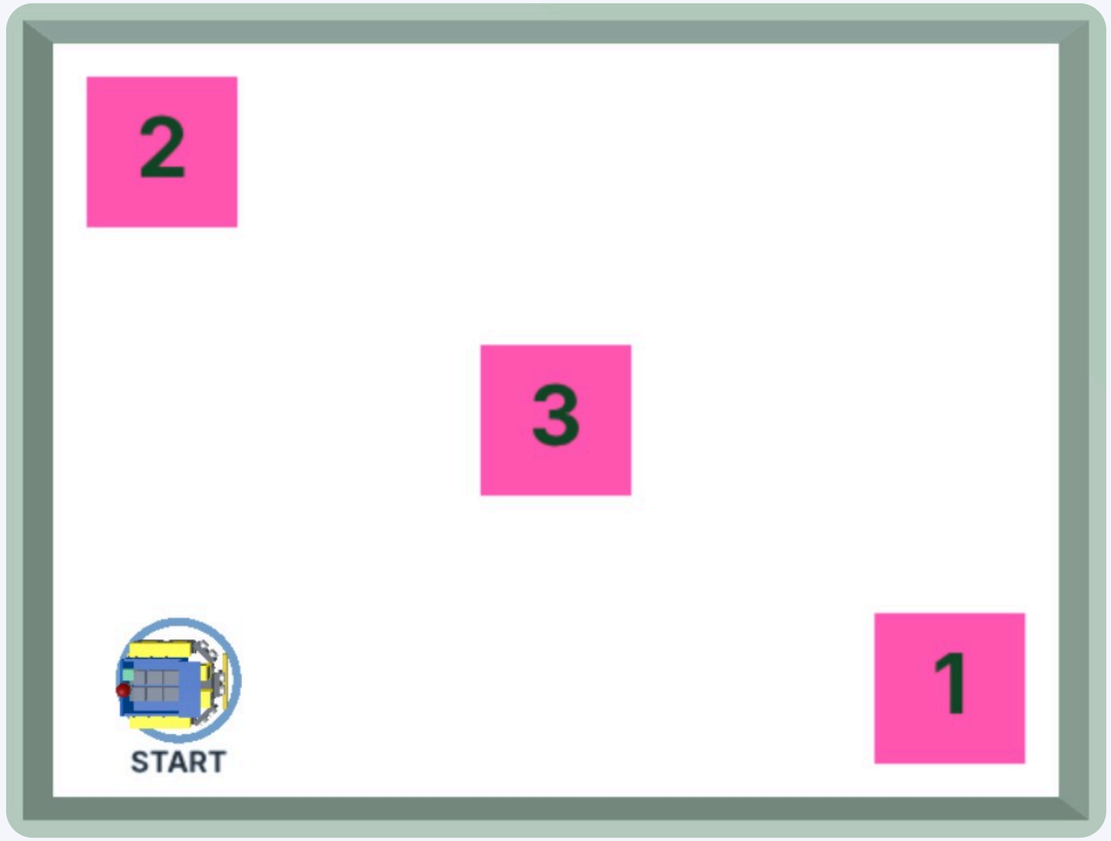
Target 1: +50. Target 2: +50. Target 3: +100. Only the next target in order counts. Time limit 4:00.

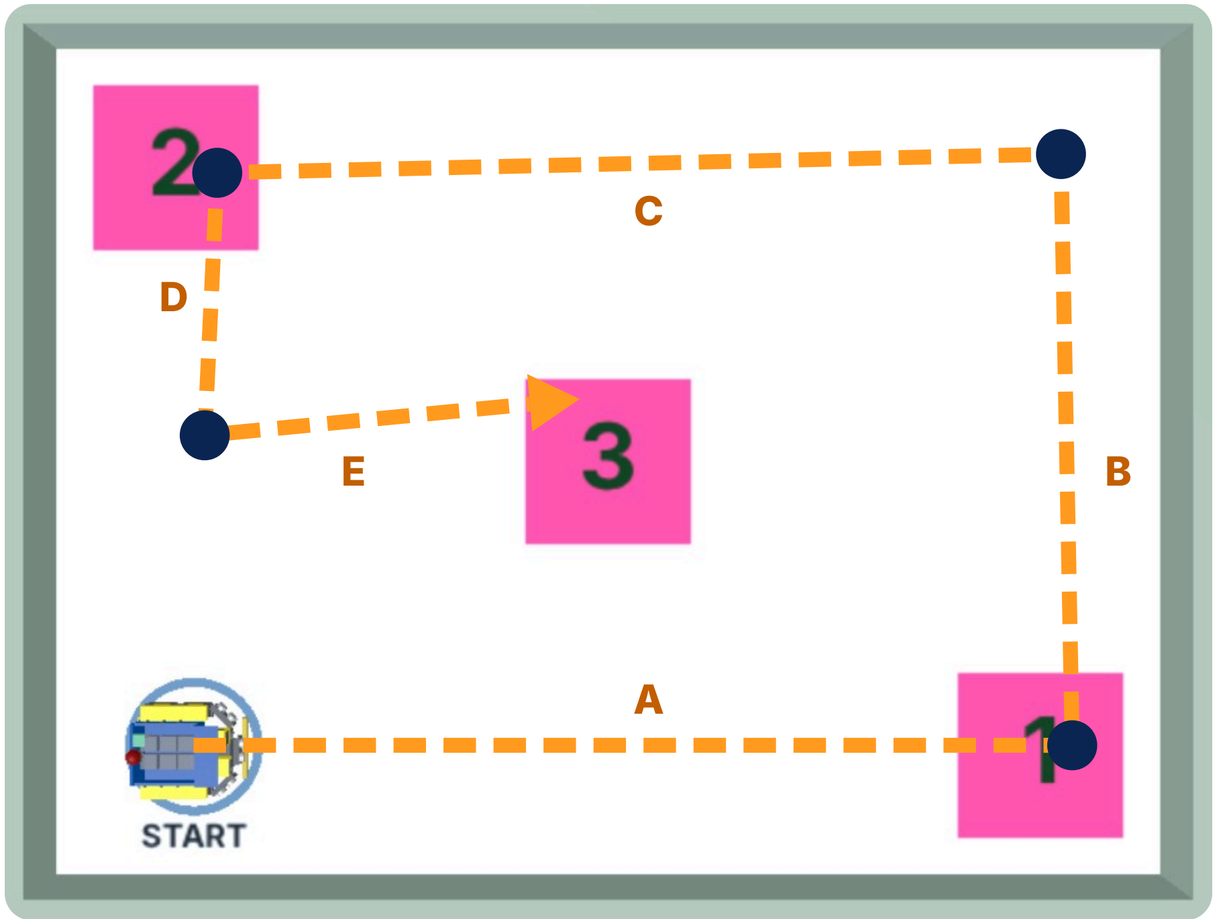
Not done yet

Visit **1 → 2 → 3** in that order. Only the next one counts.

Points: **+50**, **+50**, **+100** = 200, within **4:00**.

Same field as Session 1, with a wall around it.





STEP 2 · PLAN

Five straight legs, four left turns

A right, through **target 1**

B up the right side

C left, into **target 2**

D down a little

E right, into **target 3**

Going around the edge keeps us away from target 3 until the end.

STEP 3 · CALCULATE

How long is each leg?

Leg	Where to	Seconds × 27 cm/s	Ends near
A	X 30 → inside target 1 (196–232)	7 s = 189 cm	X ≈ 220
B	Y 28 → near the top	5 s = 135 cm	Y ≈ 160
C	X 220 → inside target 2 (8–44)	7 s = 189 cm	X ≈ 38
D	Y 150 → level with target 3 (72–108)	2 s = 54 cm	Y ≈ 95
E	X 31 → inside target 3 (102–138)	about 3 s = 81 cm	X ≈ 110

Each turn: **−2 and 2 for 1 second** ≈ a quarter turn left.

STEP 4 · TIMELINE

The whole trip in 30 seconds



Blue = straight

4.5 and 4.5

Orange = turn left

−2 and 2, one second

No gaps

Each range starts right after the last.

PART 3

Write and run

Ten statements · Build · Load · Play

STEP 5 · WRITE

Ten statements, in this order

Statement	Time Min–Max	WheelLeft	WheelRight
1 Forward1 · A	0 – 6	4.5	4.5
2 Turn1	7 – 7	−2	2
3 Forward2 · B	8 – 12	4.5	4.5
4 Turn2	13 – 13	−2	2
5 Forward3 · C	14 – 20	4.5	4.5
6 Turn3	21 – 21	−2	2
7 Forward4 · D	22 – 23	4.5	4.5
8 Turn4	24 – 24	−2	2
9 Forward5 · E	25 – 29	4.5	4.5
10 Stop	(none)	0	0

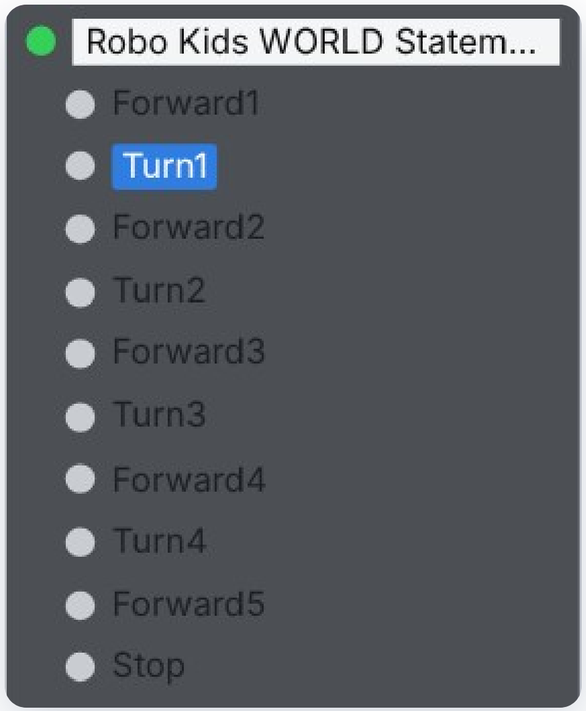
All **Default**, Duration **60**, LED 0.

Only the **Time** row changes in the conditions.

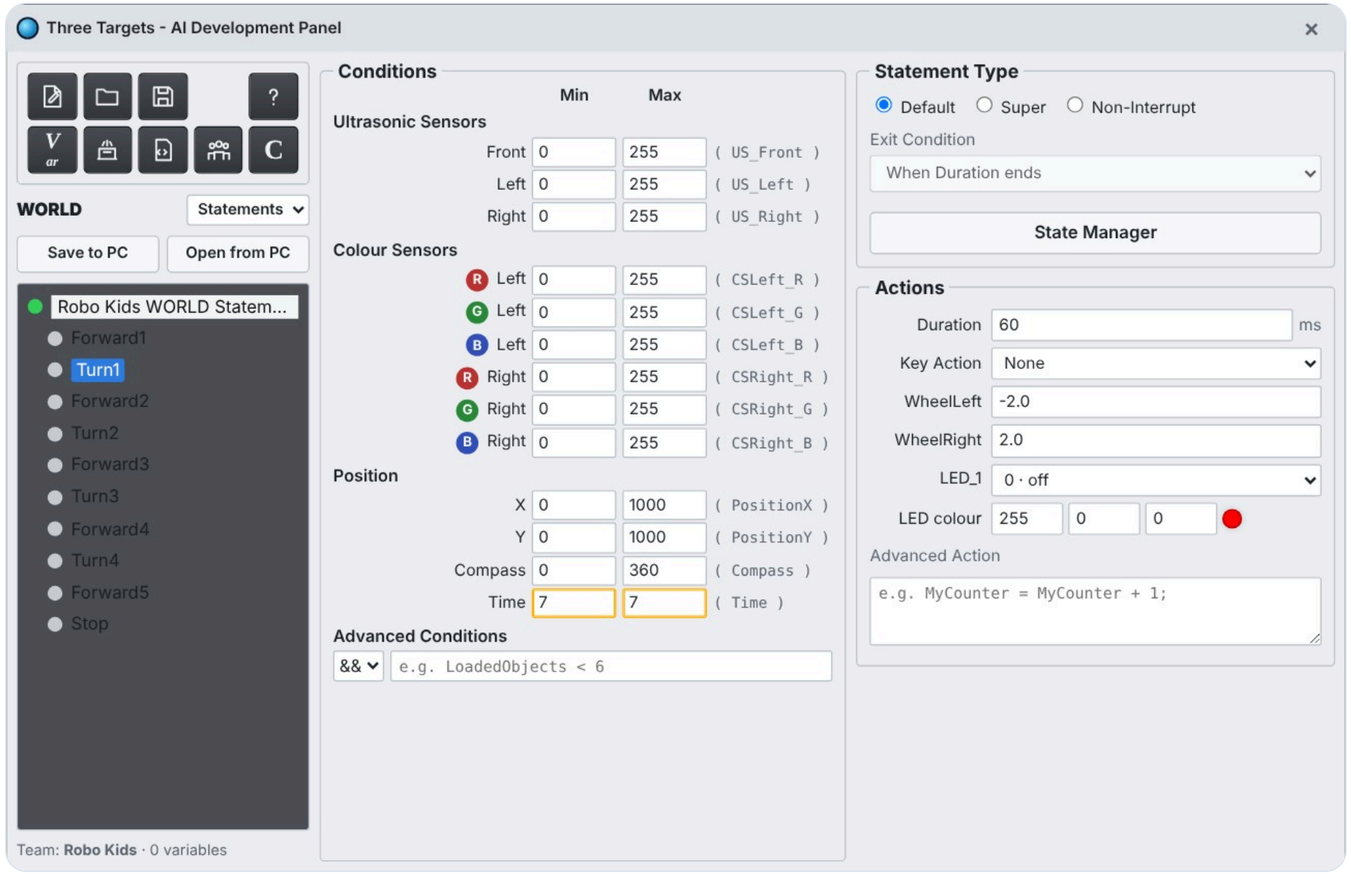
Code challenge: at most **10** statements. We use exactly 10.

STEP 5 · WRITE

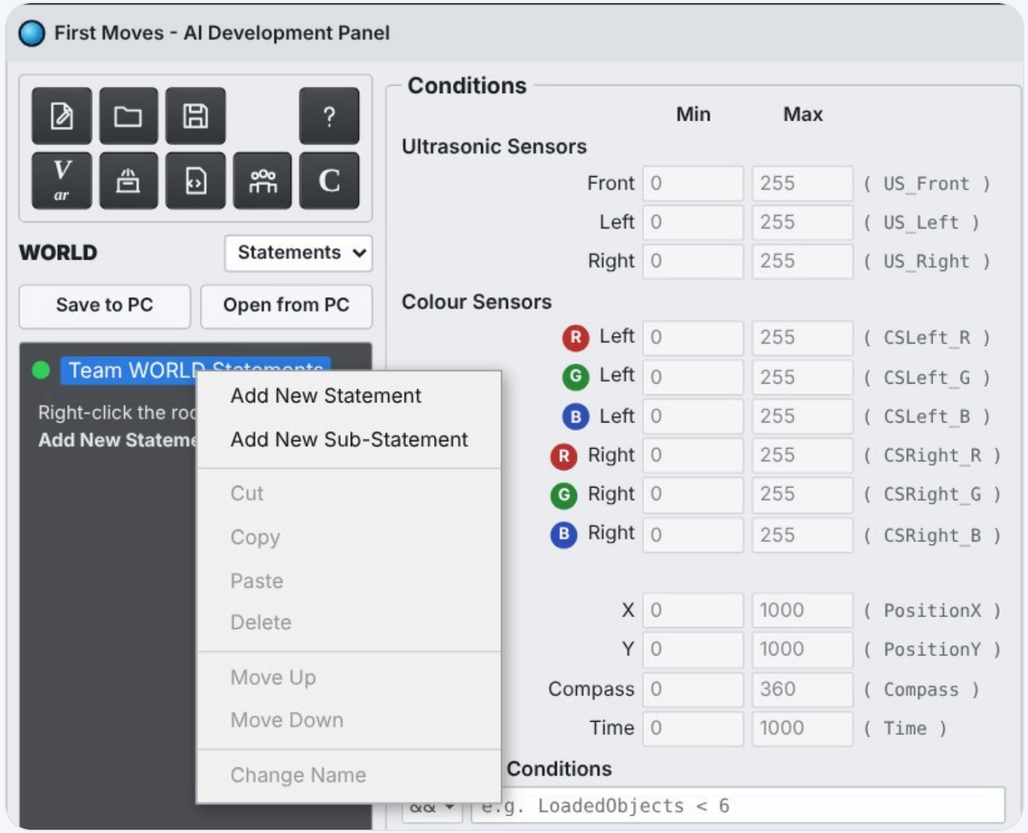
The program in the AI panel



The list: top to bottom



Turn1: Time 7 to 7 · WheelLeft −2 · WheelRight 2 · Duration 60



WORK FASTER

Copy, paste, change

1 Make **Forward1** and **Turn1** first.

2 Right-click a statement → **Copy**.

3 Right-click the last one → **Paste**: the copy goes right below it.

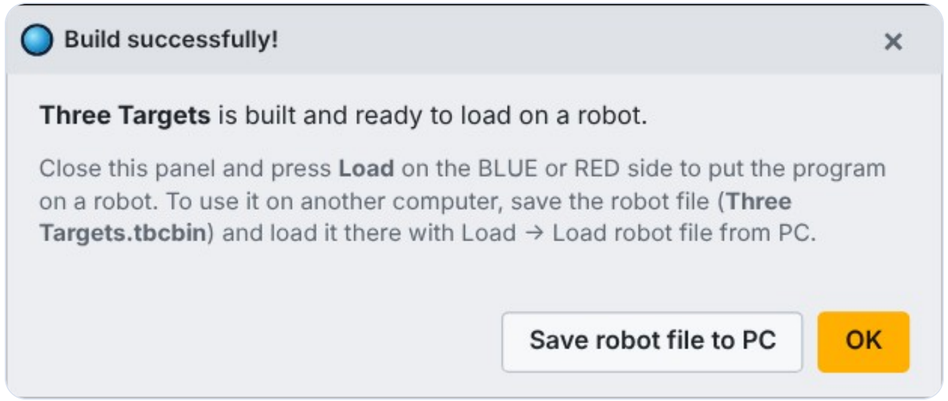
4 Double-click to rename it, then change only the **Time**.

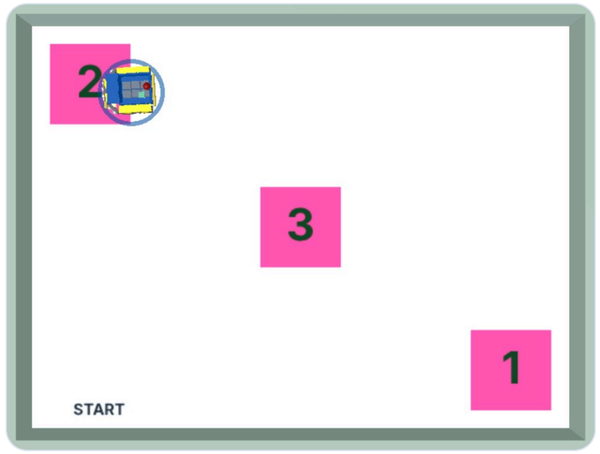
Wrong place? **Move Up** / **Move Down**.

STEP 6 · TEST

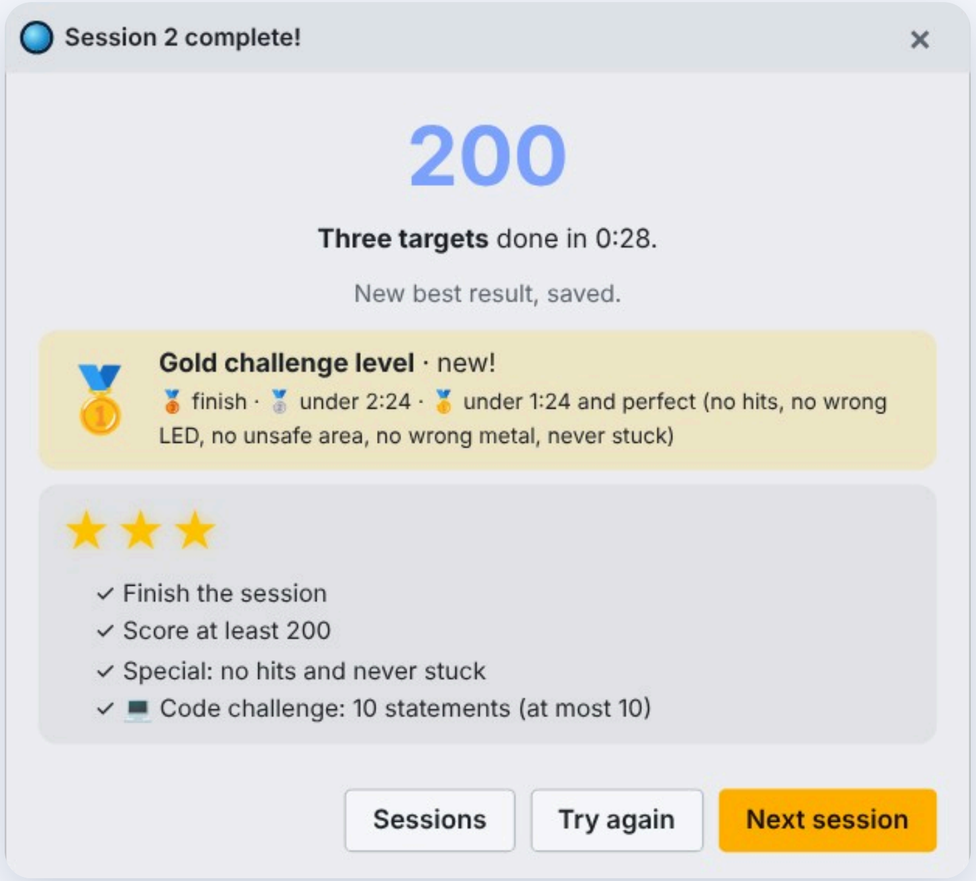
Build, Load on BLUE, Play

1 Build → OK → **2 Save to PC** (keep a copy) → **3** BLUE **Load** → Three Targets → **4 ▶ Play**





At **Time 20** the robot reaches target 2. The feed says **Target 1 reached +50**.



RESULT

200 points in 0:28 · 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 10 statements. We used 10.

PART 4

Why does the path bend?

Small errors add up

CHECK WITH THE COMPASS

Each turn is a little more than 90°

After	We wanted	Compass read	Off by
Turn1	0	**2**	2°
Turn2	90	**93**	3°
Turn3	180	**185**	5°
Turn4	270	**277**	7°

−2 and 2 for 1 second = **91.7°**, not 90°.

The extra bit **adds up** after every turn.

Leg E climbs: target 3 was reached at **Y ≈ 103**. Its top edge is **108**.

IMPROVE

Fine-tune the turn, then re-time

CHANGE 1

Smaller turn

All four turns: WheelLeft **−2**, WheelRight **1.9** ≈ **89°** a second.

CHANGE 2

Longer leg D

Forward4 **22–24** · Turn4 **25** · Forward5 **26–30**.

Tested: target 3 is reached at **Y ≈ 80**, near its middle, in **0:29**. Later we will use the Compass in conditions to stop each turn at exactly the right heading.

DID NOT WORK?

Find the leg, change one thing

What you see	What to change
No “Target 1 reached”	**Forward1**: Time Max 6 → 5 or 7
Hits the top wall on leg B	**Forward2**: Time 8–12 → 8–11
“Target 2 first”	The robot crossed target 3 too early: check the turn Times
Misses target 3 above or below	**Forward4** Time, or the turn wheels (−2, 1.9)
Stops before target 3	**Forward5**: make the Time Max bigger
Stops in the middle of the trip	A gap between two Time ranges

TRY IT YOURSELF

Three challenges

Nine statements

Remove one statement and still finish. Hint: what happens when nothing matches?

Straighter

Use the fine-tuned turns and check the Compass after each one.

Next session

Presentation 3 · **Avoid the walls** with the ultrasonic sensors.