PRESENTATION 12 · LEARNING WITH C

Follow the wall

Around a big block to FINISH with the wall follower we already know, and a good choice of which wall.

TheBotCup 2026 · AiBotCup Simulator



RECAP · PRESENTATIONS 4 AND 5

The right-hand wall follower

Ahead
wall in front → spin left

TooClose
right wall near → ease left

Lost
right wall far → curve right

Forward
otherwise drive

Four branches, a hand on the wall, and it solved two mazes. Today: the **same four**.

PART 1

The mission: Session 12

Around the block to FINISH

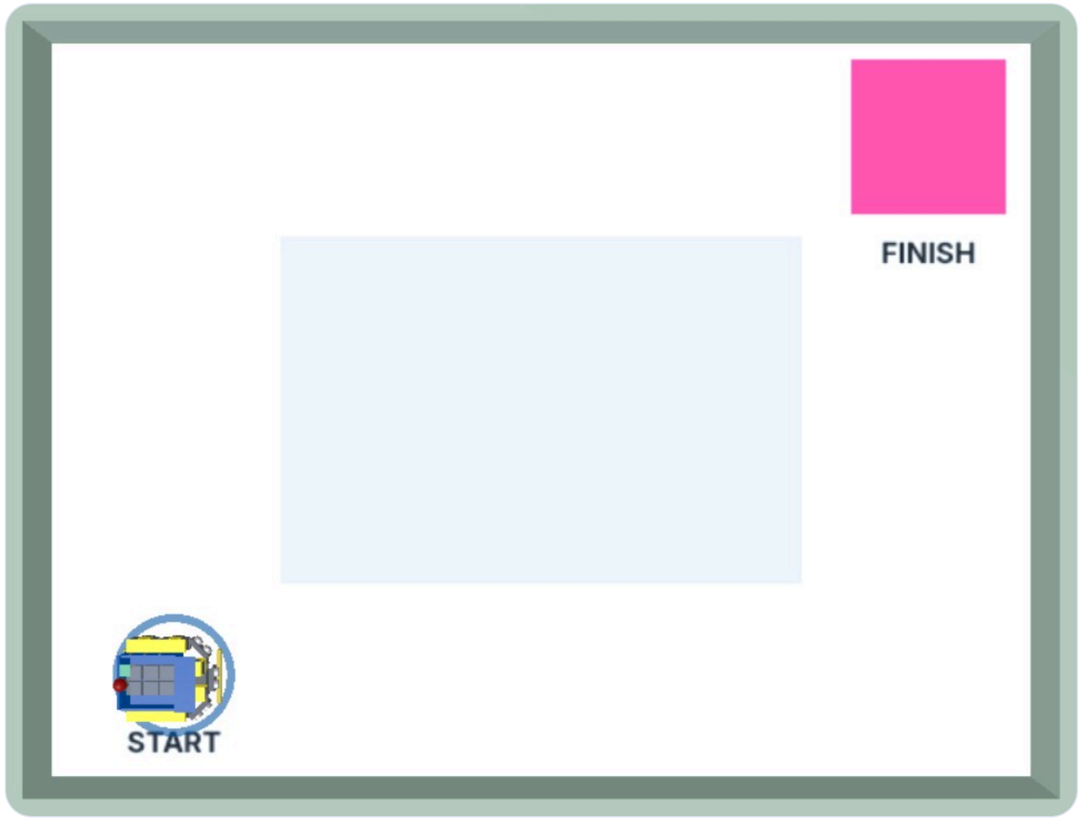
SESSION 12 · FOLLOW THE WALL

Keep a wall on one side, all the way

Reach the pink **FINISH** pad: **+150**.

Time limit **4:00**. Code challenge: at most **20** lines.





THE FIELD

A big block in the way

START bottom left, facing **right**.

A big block fills the middle.

FINISH sits in the **top-right corner**, against the outer walls.

PART 2

Which wall?

The block, or the outer wall

WE TESTED BOTH

Follow the wall that touches FINISH

TESTED · DID NOT FINISH

Block on the left

The robot circled the block and kept missing the corner. **0 points** after 4:00.

TESTED · WORKED

Outer wall on the right

Along the bottom, up the right side, straight into FINISH. **0:12**.

FINISH is in a corner of the **outer** wall, so following that wall must pass over it.

THE PLAN

Right hand on the outer wall

1

Drive right, **bottom wall** on the right.

2

Corner ahead: **Ahead** spins left.

3

Drive up, **right wall** on the right, into FINISH.

The block is on the robot’s left the whole time, so we still go **around the block**.

PART 3

Step by step in C

Four branches, already known

STEP 1 · THE WHOLE PROGRAM

The right-hand rule, unchanged

Branch	Condition	Wheels
1 Ahead	US_Front ≤ 20	−2 / 2
2 TooClose	US_Right ≤ 22	2.5 / 4.5
3 Lost	US_Right ≥ 45	4.5 / 1.5
4 Forward	the else	4.5 / 4.5

Between 23 and 44 cm on the right, nothing above matches: **Forward**.

STEP 2 · LINES 3–7

Ahead and TooClose

```
3  void Game1() {
4      if (US_Front <= 20) {                // Ahead: wall in front
5          WheelLeft = -2; WheelRight = 2;
6      } else if (US_Right <= 22) {         // TooClose: drift left
7          WheelLeft = 2.5; WheelRight = 4.5;
```

Ahead: US_Front ≤ 20 → spin left, −2 / 2. Wall in front: turn away.

TooClose: US_Right ≤ 22 → 2.5 / 4.5, a gentle turn left, away from the wall.

STEP 3 · LINES 8–13

Lost and Forward

```
 8       } else if (US_Right >= 45) {          // Lost: curve right
 9           WheelLeft = 4.5; WheelRight = 1.5;
10       } else {                              // Forward
11           WheelLeft = 4.5; WheelRight = 4.5;
12       }
13   }
```

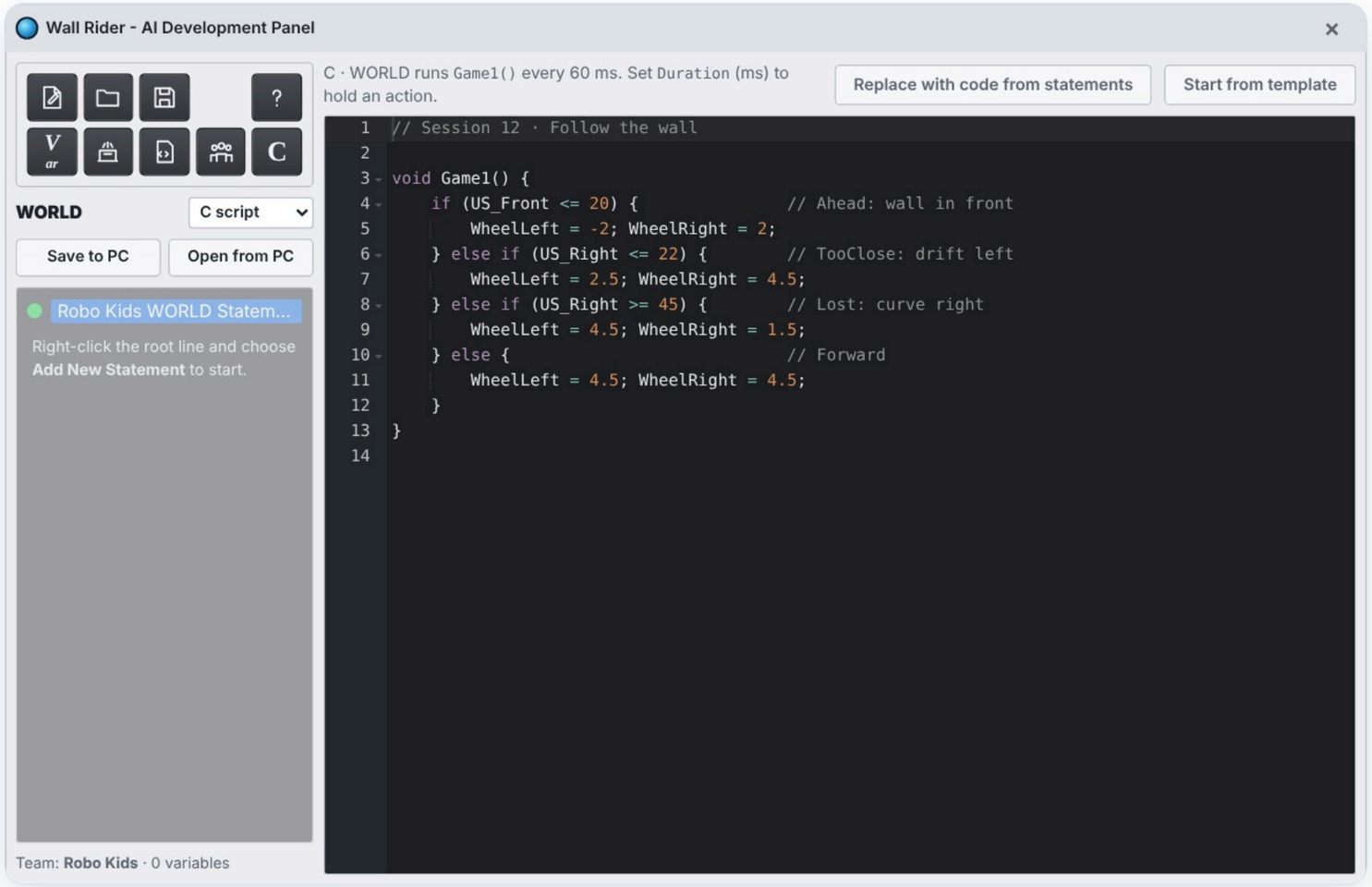
Lost: US_Right ≥ 45 → 4.5 / 1.5, curve right to find the wall again.

Forward: the else. Right wall 23–44 cm away → drive 4.5 / 4.5.

STEP 4 · BUILD AND LOAD

The whole program, built

1. WORLD menu: **C script**.
2. Type the program, name it **Wall Rider**.
3. Press **Build**, then **OK**.
4. Press **Load** on the BLUE side.





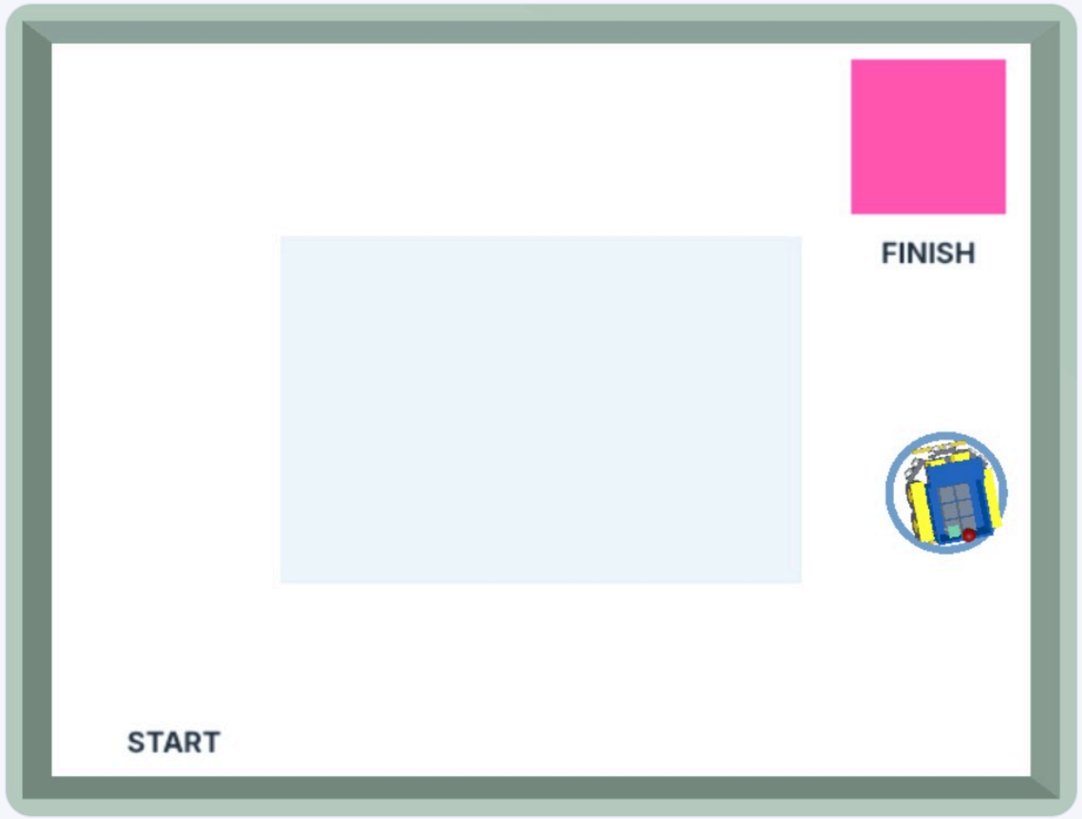
STEP 5 · RUN · BOTTOM WALL

Between the wall and the block

Ultrasonic Sensors

Front	119	(US_Front)
Left	27	(US_Left)
Right	27	(US_Right)

Right **27**: in the good band. Front **118**: clear. So **Forward**.



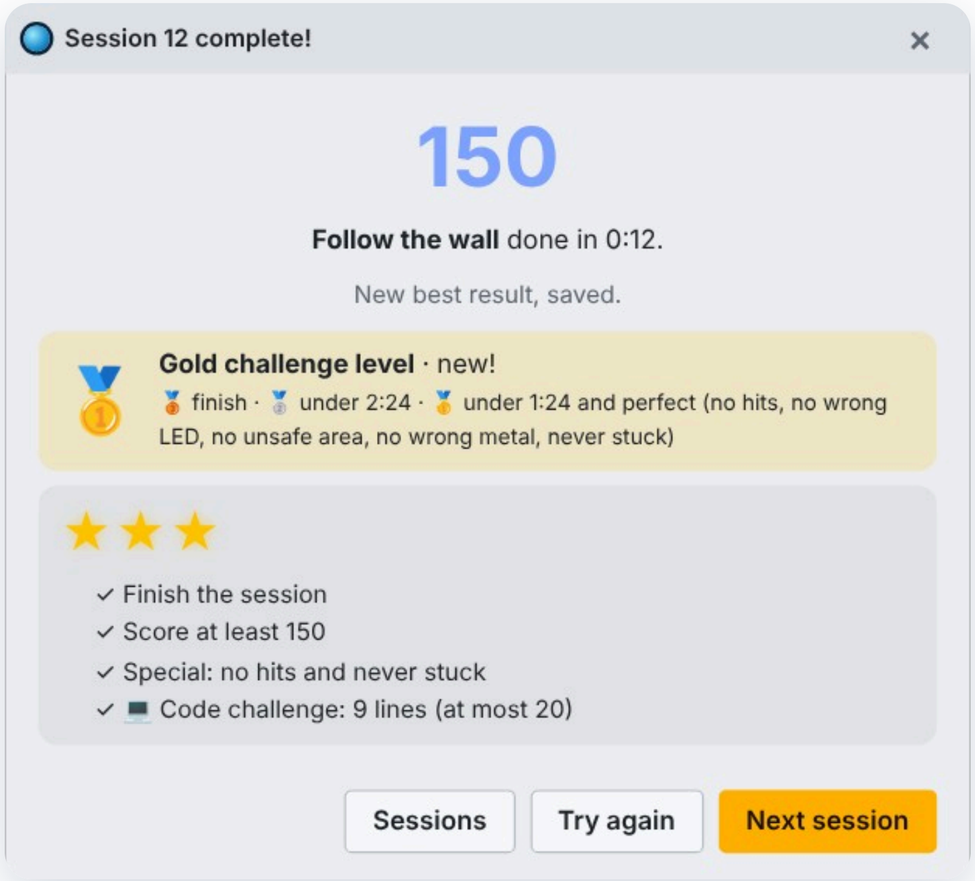
STEP 6 · RUN · RIGHT WALL

Round the corner and up

Ultrasonic Sensors

Front	98	(US_Front)
Left	37	(US_Left)
Right	31	(US_Right)

After the corner, the right wall is the outer wall again: Right **31**, straight up into FINISH.



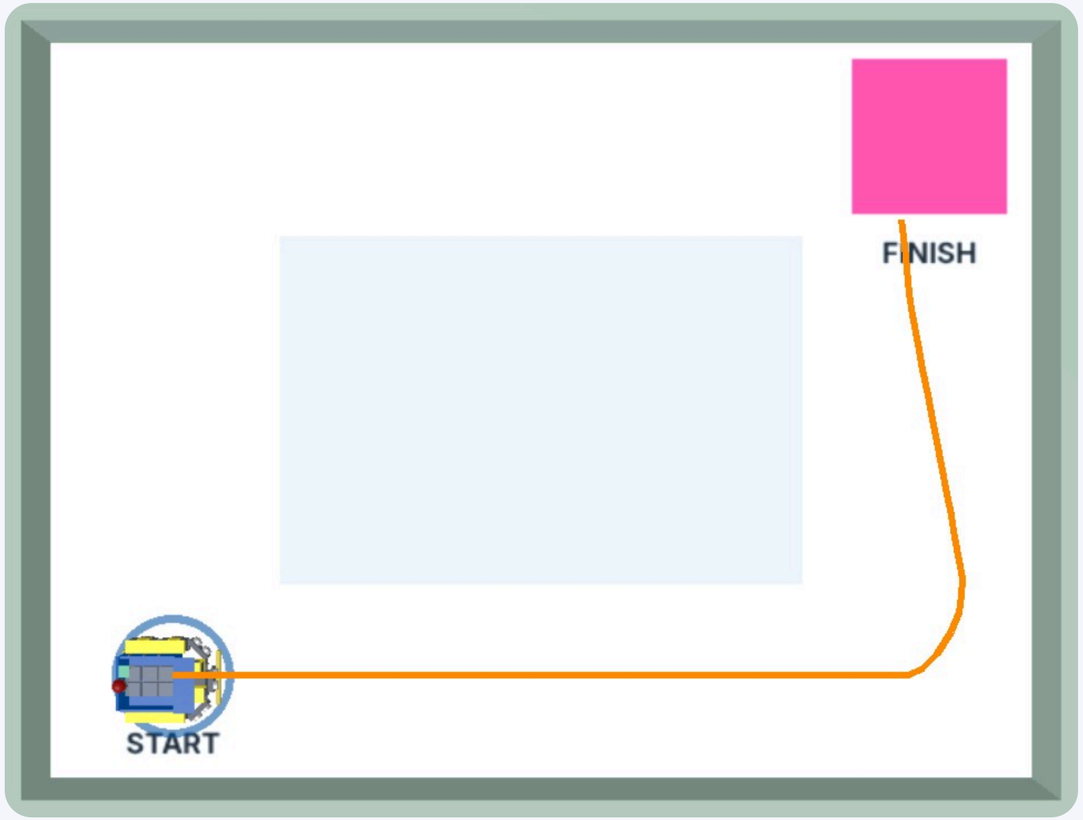
RESULT · TESTED

150 points in 0:12 · Gold

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

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

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



THE PATH · RECORDED

An L-shape round the block

Along the bottom, one corner, then up into FINISH.

Same program as the mazes. The **choice of wall** made it work.

DID NOT WORK?

Check first, then tune carefully

What you see	What to do
Goes round and round the block	You are following the block: use **US_Right**, not US_Left
Turns the wrong way at the corner	Ahead must spin **left**: −2 / 2
Wobbles along the wall	Check TooClose ≤ 22 and Lost ≥ 45
Old program still on the robot	**Build** again, then **Load** on BLUE

NEXT TIME

Presentation 13 · Water pond

Mining in water: a new function and a **blue LED**.