PRESENTATION 16 · LEARNING WITH C

Around the flags

Plan a route, then drive it with position and compass: no flag touched.

TheBotCup 2026 · AiBotCup Simulator



RECAP · PRESENTATION 11

The compass in the simulator

0	90	180	270
north · up	west · left	south · down	east · right

Turning **left** makes the compass go **up**; turning right makes it go down. Position X and Y say **where** the robot is.

PART 1

The mission: Session 16

To FINISH, no flag touched

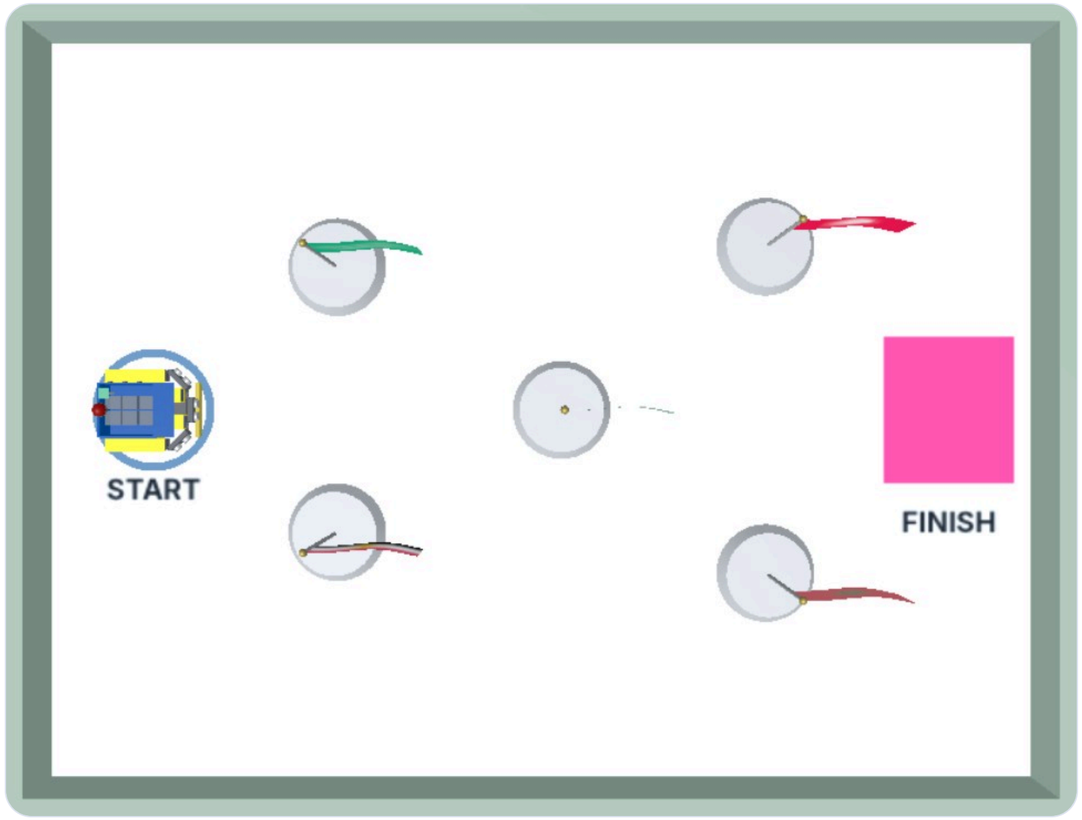
SESSION 16 · AROUND THE FLAGS

START to FINISH, flags untouched

Reach FINISH: **+150**. Each touch: **−10**.

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





THE FIELD

Five flags in the way

START left middle, facing **right** (compass 270).

Five flags on round bases in the middle.

FINISH on the right. The bottom strip is **free**.

PART 2

Plan a route

Zones, headings and drives

WE TESTED BOTH

React, or follow a plan?

TESTED · DID NOT WORK

“See a flag, turn away”

Thin flags are hard to see: it touched flags, or got stuck between two of them.

TESTED · WORKED

A planned route

Down, along the free bottom strip, then up into FINISH. **0 touches, 0:14**.

THE PLAN · THREE LEGS

Each zone: turn, then drive

Leg	Zone (where)	Heading	Compass
1 South	X ≤ 199 · Y ≥ 22	down	about 180
2 East	X ≤ 199 · Y ≤ 21	right	about 270
3 North	X ≥ 200	up	about 360 (= 0)

In every zone: **Turn** while the compass is wrong, otherwise **Go** straight.

PART 3

Step by step in C

Six branches, three pairs

STEP 1 · THE WHOLE PROGRAM

Turn and Go, three times

Branch	Position	Compass	Wheels
1 TurnSouth	X ≤ 199 · Y ≥ 22	≥ 185	2 / −2 (turn right)
2 GoSouth	X ≤ 199 · Y ≥ 22	any	4.5 / 4.5
3 TurnEast	X ≤ 199 · Y ≤ 21	≤ 265	−2 / 2 (turn left)
4 GoEast	X ≤ 199 · Y ≤ 21	any	4.5 / 4.5
5 TurnNorth	X ≥ 200	180–355	−2 / 2 (turn left)
6 GoNorth	X ≥ 200	any	4.5 / 4.5

Each **Turn** comes before its **Go**: it wins until the compass is right.

STEP 2 · LEG 1 · LINES 4–7

Face south, drive down

```
1  // Session 16 · Around the flags
2
3  void Game1() {
4      if (PositionX <= 199 && PositionY >= 22 && Compass >= 185) {              // TurnSouth
5          WheelLeft = 2; WheelRight = -2;
6      } else if (PositionX <= 199 && PositionY >= 22) {                         // GoSouth
7          WheelLeft = 4.5; WheelRight = 4.5;
8      } else if (PositionX <= 199 && PositionY <= 21 && Compass <= 265) {       // TurnEast
9          WheelLeft = -2; WheelRight = 2;
10     } else if (PositionX <= 199 && PositionY <= 21) {                         // GoEast
11         WheelLeft = 4.5; WheelRight = 4.5;
12     } else if (PositionX >= 200 && Compass >= 180 && Compass <= 355) {        // TurnNorth
13         WheelLeft = -2; WheelRight = 2;
14     } else if (PositionX >= 200) {                                            // GoNorth
15         WheelLeft = 4.5; WheelRight = 4.5;
16     }
17 }
```

TurnSouth: zone && Compass ≥ 185 → turn right, 2 / −2.

GoSouth: same zone, any compass → 4.5 / 4.5.

STEP 3 · LEG 2 · LINES 8–11

Face east, drive along the bottom

```
1  // Session 16 · Around the flags
2
3  void Game1() {
4      if (PositionX <= 199 && PositionY >= 22 && Compass >= 185) {              // TurnSouth
5          WheelLeft = 2; WheelRight = -2;
6      } else if (PositionX <= 199 && PositionY >= 22) {                         // GoSouth
7          WheelLeft = 4.5; WheelRight = 4.5;
8      } else if (PositionX <= 199 && PositionY <= 21 && Compass <= 265) {       // TurnEast
9          WheelLeft = -2; WheelRight = 2;
10     } else if (PositionX <= 199 && PositionY <= 21) {                         // GoEast
11         WheelLeft = 4.5; WheelRight = 4.5;
12     } else if (PositionX >= 200 && Compass >= 180 && Compass <= 355) {        // TurnNorth
13         WheelLeft = -2; WheelRight = 2;
14     } else if (PositionX >= 200) {                                            // GoNorth
15         WheelLeft = 4.5; WheelRight = 4.5;
16     }
17 }
```

TurnEast: bottom strip && Compass ≤ 265 → turn left, −2 / 2.

GoEast: same zone → drive right.

STEP 4 · LEG 3 · LINES 12–15

Face north, drive into FINISH

```
1  // Session 16 · Around the flags
2
3  void Game1() {
4      if (PositionX <= 199 && PositionY >= 22 && Compass >= 185) {              // TurnSouth
5          WheelLeft = 2; WheelRight = -2;
6      } else if (PositionX <= 199 && PositionY >= 22) {                         // GoSouth
7          WheelLeft = 4.5; WheelRight = 4.5;
8      } else if (PositionX <= 199 && PositionY <= 21 && Compass <= 265) {       // TurnEast
9          WheelLeft = -2; WheelRight = 2;
10     } else if (PositionX <= 199 && PositionY <= 21) {                         // GoEast
11         WheelLeft = 4.5; WheelRight = 4.5;
12     } else if (PositionX >= 200 && Compass >= 180 && Compass <= 355) {        // TurnNorth
13         WheelLeft = -2; WheelRight = 2;
14     } else if (PositionX >= 200) {                                            // GoNorth
15         WheelLeft = 4.5; WheelRight = 4.5;
16     }
17 }
```

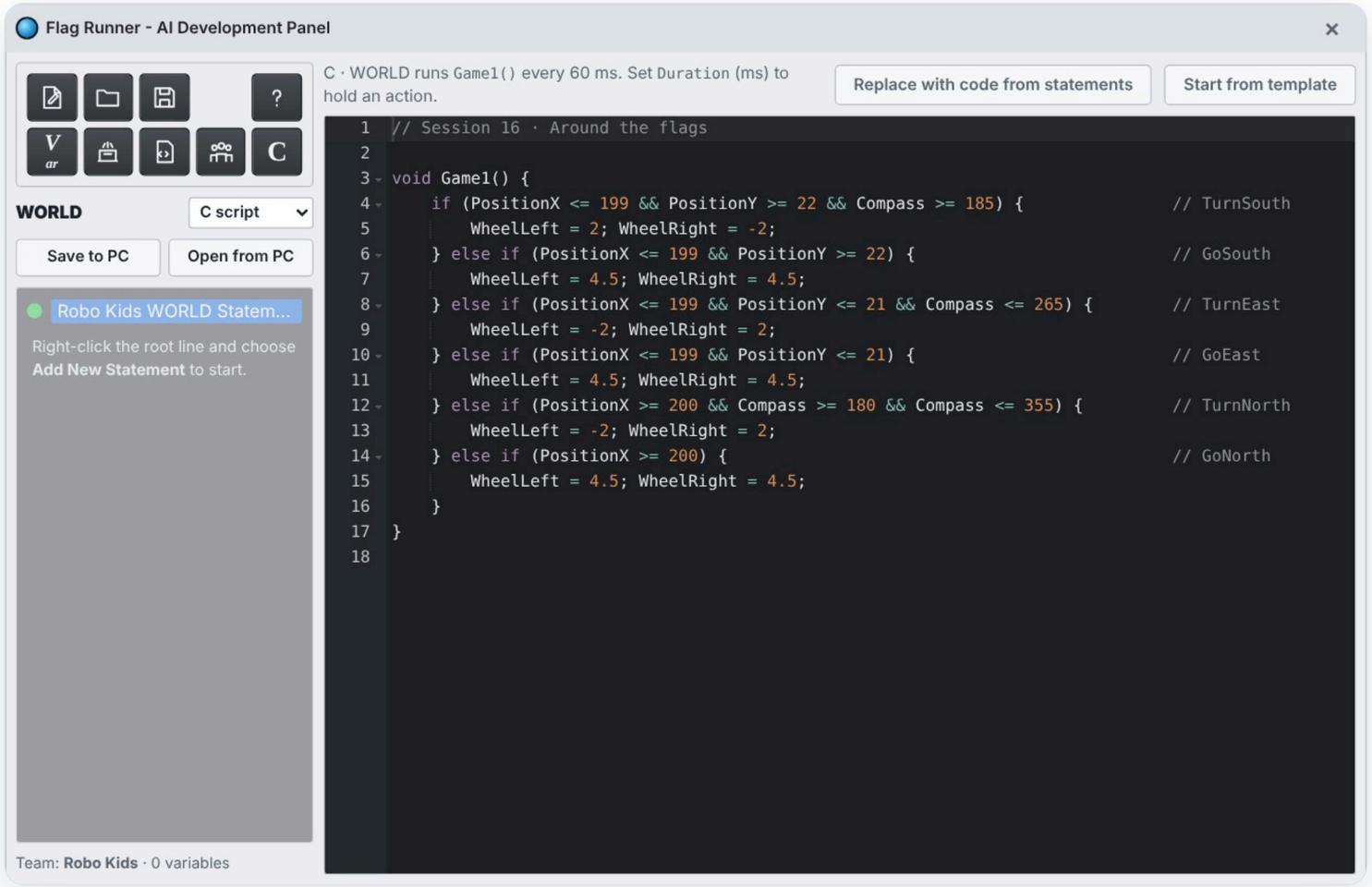
TurnNorth: PositionX ≥ 200 && Compass 180–355 → turn left.

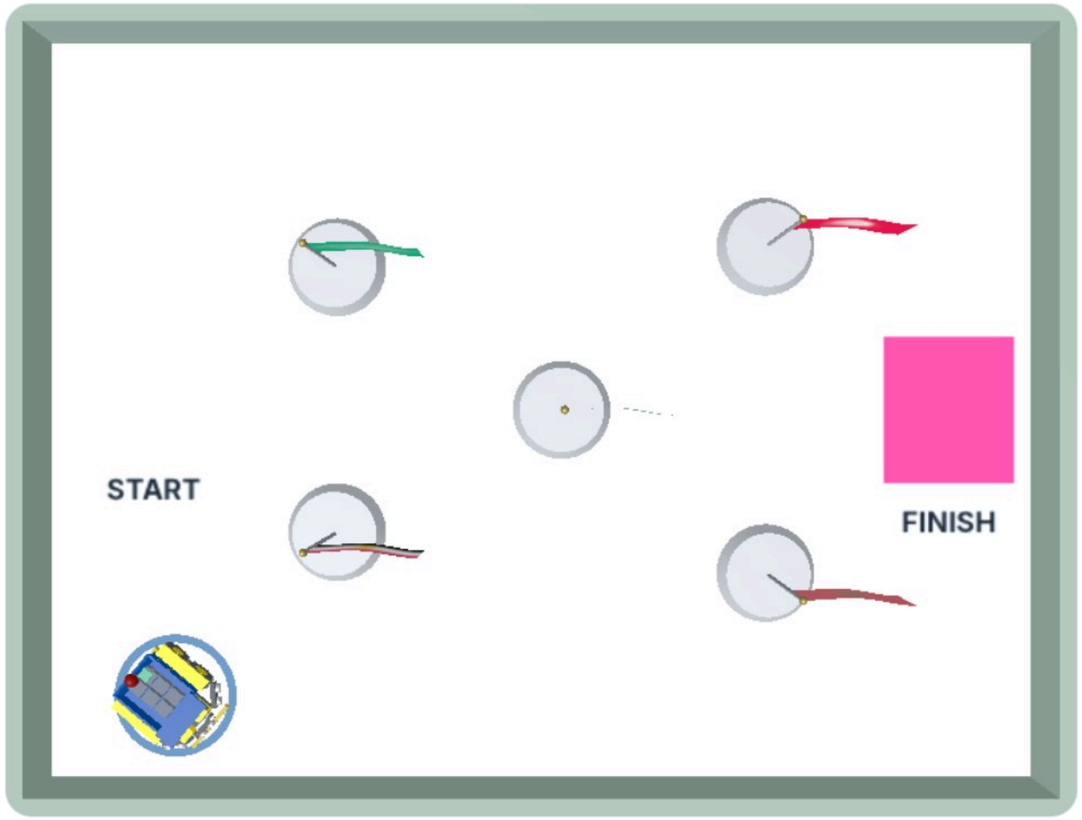
GoNorth: PositionX ≥ 200 → drive up into FINISH.

STEP 5 · BUILD AND LOAD

The whole program, built

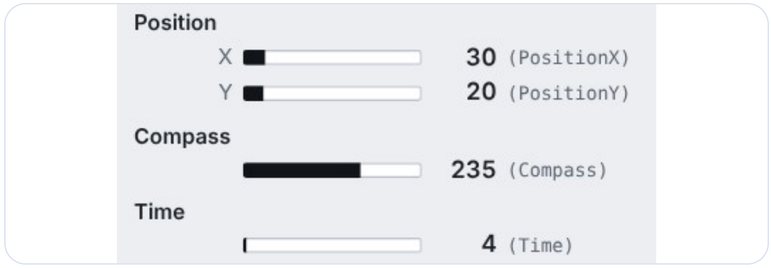
1. WORLD menu: **C script**.
2. Type the program, name it **Flag Runner**.
3. Press **Build**, then **OK**.
4. Press **Load** on the BLUE side.



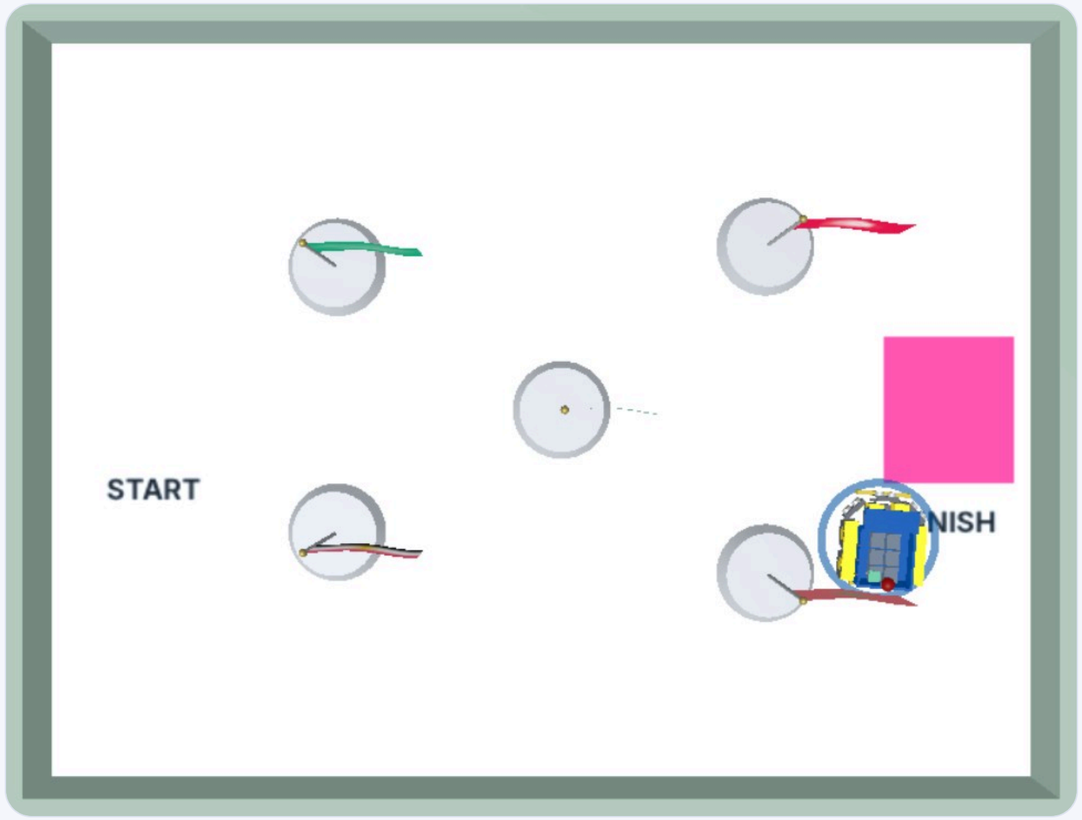


STEP 6 · RUN · 4.1 S

The first corner

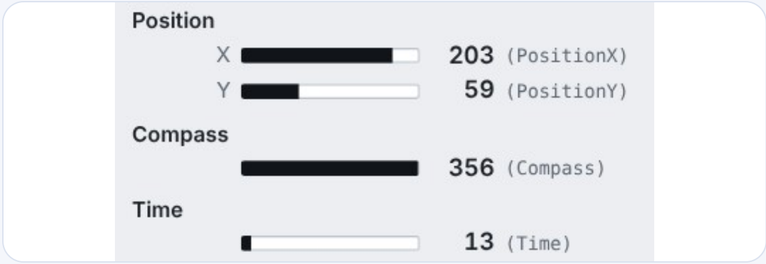


Y **20**: bottom strip. Compass **235** < 266: **TurnEast** is turning.

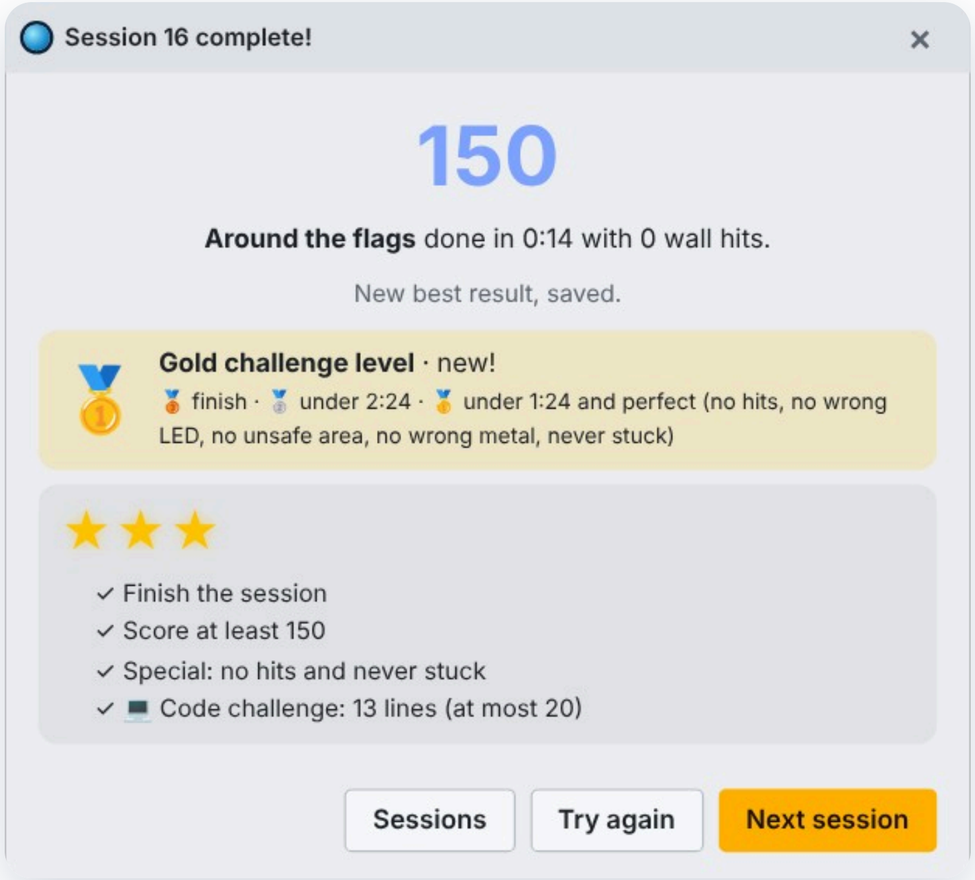


STEP 7 · RUN · 13.1 S

Up into FINISH



X **203**, compass **356**: **GoNorth**, past the last flag.



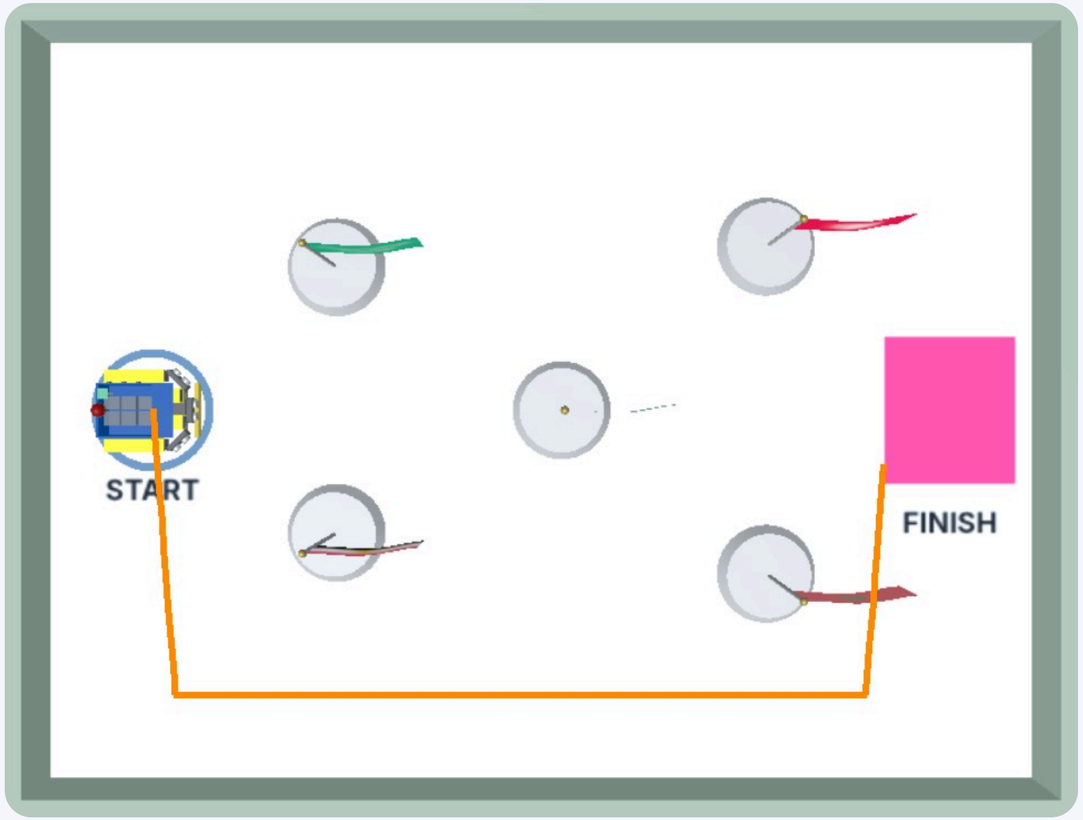
RESULT · TESTED

150 points in 0:14 · Gold

Touches · 0. No hits, never stuck.

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

Code challenge · at most 20 lines. We used 13.



THE PATH · RECORDED

A U-shape below the flags

Down, right, up: three straight legs, as planned.

When sensors are not enough, **position and compass** drive a plan.

DID NOT WORK?

Check first, then tune carefully

What you see	What to do
Spins in place forever	Turn direction: right = 2 / −2, left = −2 / 2
Drives off at the wrong angle	Check each Turn’s compass range
Never starts a leg	Zones must touch: Y 22 / 21, X 199 / 200
Old program still on the robot	**Build** again, then **Load** on BLUE

NEXT TIME

Presentation 17 · Timer

Using **Time** as a condition: drive for a set number of seconds.