PRESENTATION 1 · LEARNING WITH C

First moves

How C code runs, how two wheels move the robot, the Time sensor, and Session 1 step by step.

TheBotCup 2026 · AiBotCup Simulator



TODAY

Four steps to your first program

1

C

How the robot runs your code.

2

Two wheels

Differential drive: going straight and turning.

3

Time sensor

Use the clock to say WHEN each move happens.

4

Session 1

Solve First moves, step by step.

PART 1

How coding in C works

if · else · the 60 ms loop

C

The four parts of a program

```
1  // Session 1 · First moves
2
3  void Game1() {
4      if (Time <= 5) {                          // Forward1
5          WheelLeft = 4.5; WheelRight = 4.5;
6      } else if (Time == 6) {                   // TurnLeft
7          WheelLeft = -2; WheelRight = 2;
8      } else if (Time <= 12) {                  // Forward2
9          WheelLeft = 4.5; WheelRight = 4.5;
10     } else {                                  // Stop
11         WheelLeft = 0; WheelRight = 0;
12     }
13 }
```

1 · Game1() the function the robot runs again and again.

2 · Conditions after if and else if, like **Time <= 5**. True or false.

3 · Actions the lines inside: WheelLeft, WheelRight, LED and **Duration** in ms.

4 · Comments after //: a name for each branch, like **Forward1**. The robot ignores them.

C

What the robot does every 60 ms

STEP 1

Still busy?

If the last action’s Duration has not ended, keep going.

STEP 2

Run Game1()

Run Game1() from the **top**.

STEP 3

First true branch runs

Its lines set the outputs, held for the Duration.

Order matters. Two conditions can both be true: only the higher branch runs.

Nothing matches? The wheels keep their last values. A last else avoids surprises.

C

Conditions are questions: true or false

Write	Means	Example
<=	at most	Time <= 5 · 0, 1, 2, 3, 4, 5
>=	at least	Time >= 7 · 7, 8, 9 …
==	exactly (two = signs)	Time == 6 · only 6
!=	not equal	Time != 6 · everything but 6
&&	both must be true	Time >= 7 && Time <= 12
||	one is enough	Time == 6 || Time == 13

One **=** sets a value: **WheelLeft = 4.5**. Two **==** ask a question: **Time == 6**.

C

Duration: how long to keep an action

1000 ms

= 1 second. Duration is always in milliseconds.

60 ms

No Duration: Game1() runs again after 60 ms. **We use this today.**

400 ms

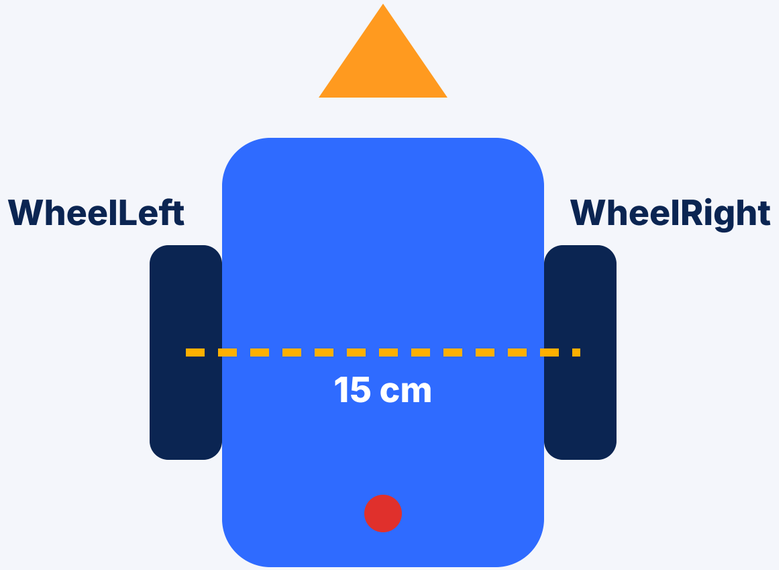
Duration = 400 holds the action for 0.4 s before Game1() runs again.

No Duration + conditions = the robot checks again every 60 ms and reacts quickly.

PART 2

Differential drive: how the robot moves

Two wheels · no steering wheel



DIFFERENTIAL DRIVE

Two motors, one on each side

WheelLeft and **WheelRight** each go from **−5** (full reverse) through **0** (stop) to **5** (full forward), in steps of 0.1.

Each unit = **6 cm per second**. Full speed 5 = 30 cm/s.

Same speeds → **straight**. Different speeds → **it turns** toward the slower wheel.

DIFFERENTIAL DRIVE

Six moves from two numbers



Forward

5 and **5**



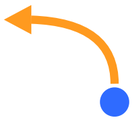
Backward

−5 and **−5**



Stop

0 and **0**



Curve left

2 and **5**



Spin left

−2 and **2**



Spin right

2 and **−2**

Values are **WheelLeft** then **WheelRight**. Opposite values spin the robot on the spot, without moving forward.

DIFFERENTIAL DRIVE

Handy numbers for planning

WheelLeft	WheelRight	What happens
5	5	Straight, **30 cm** each second (fastest)
4.5	4.5	Straight, **27 cm** each second
2.5	2.5	Straight, **15 cm** each second
−2	2	Spin left, about **92°** each second
−1	1	Spin left, about **46°** each second
2	−2	Spin right, about **92°** each second

Speed

(Left + Right) ÷ 2 × 6 cm/s

Distance

speed × time
27 cm/s × 6 s = **162 cm**

PART 3

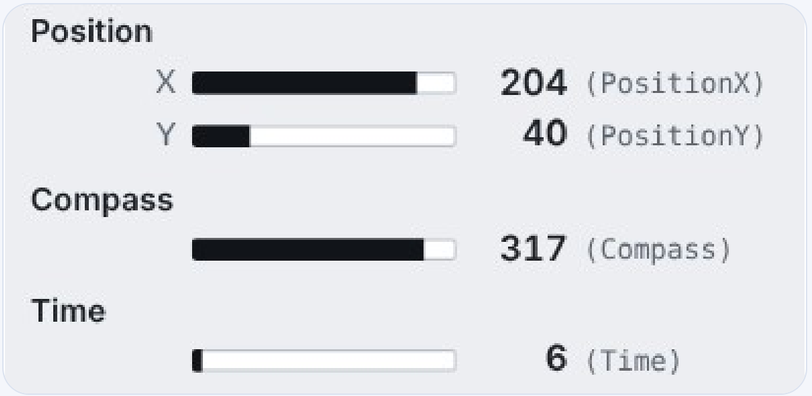
The Time sensor

The robot’s own stopwatch

TIME SENSOR

Seconds since you pressed Play

- Starts at **0** and counts up: 0, 1, 2, 3 …
- Whole seconds only: from 6.0 to 6.9 it reads **6**.
- In code: **Time** is a whole number, like **Time <= 5**.
- Watch it live in the Sensors list.



6.5 s after Play: **Time = 6**

TIME SENSOR

Plan your program as a timeline



Time <= 5

= **6 seconds** (0.0 to 5.9), because Time counts whole seconds.

Time == 6

= exactly **1 second**: just right for a 92° turn.

No gaps

With else if, each branch starts where the last one ended. The else catches the rest.

PART 4

Solving Session 1 · First moves

Look · plan · calculate · write · test

STEP 1 · LOOK

The mission

1

Session 1 · First moves

Drive the robot from START into the green deposit area.

Reach the deposit area: +100. Time limit 3:00.

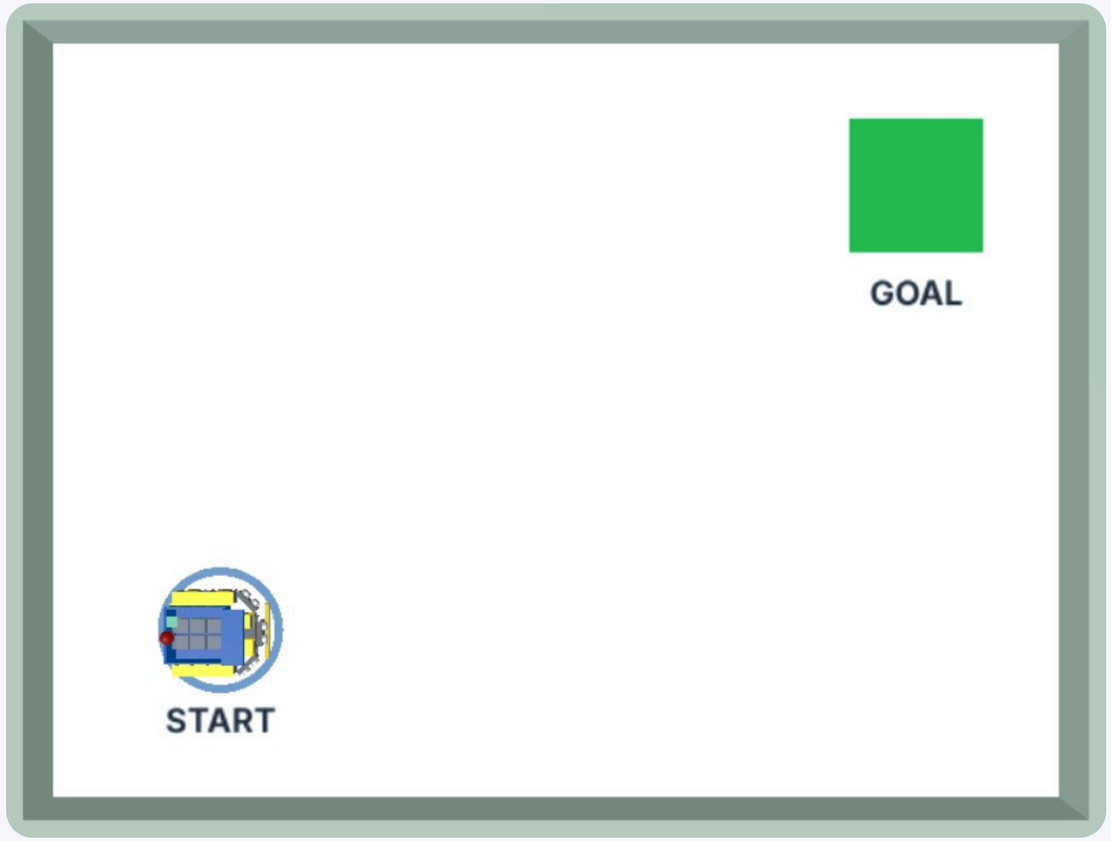
Not done yet

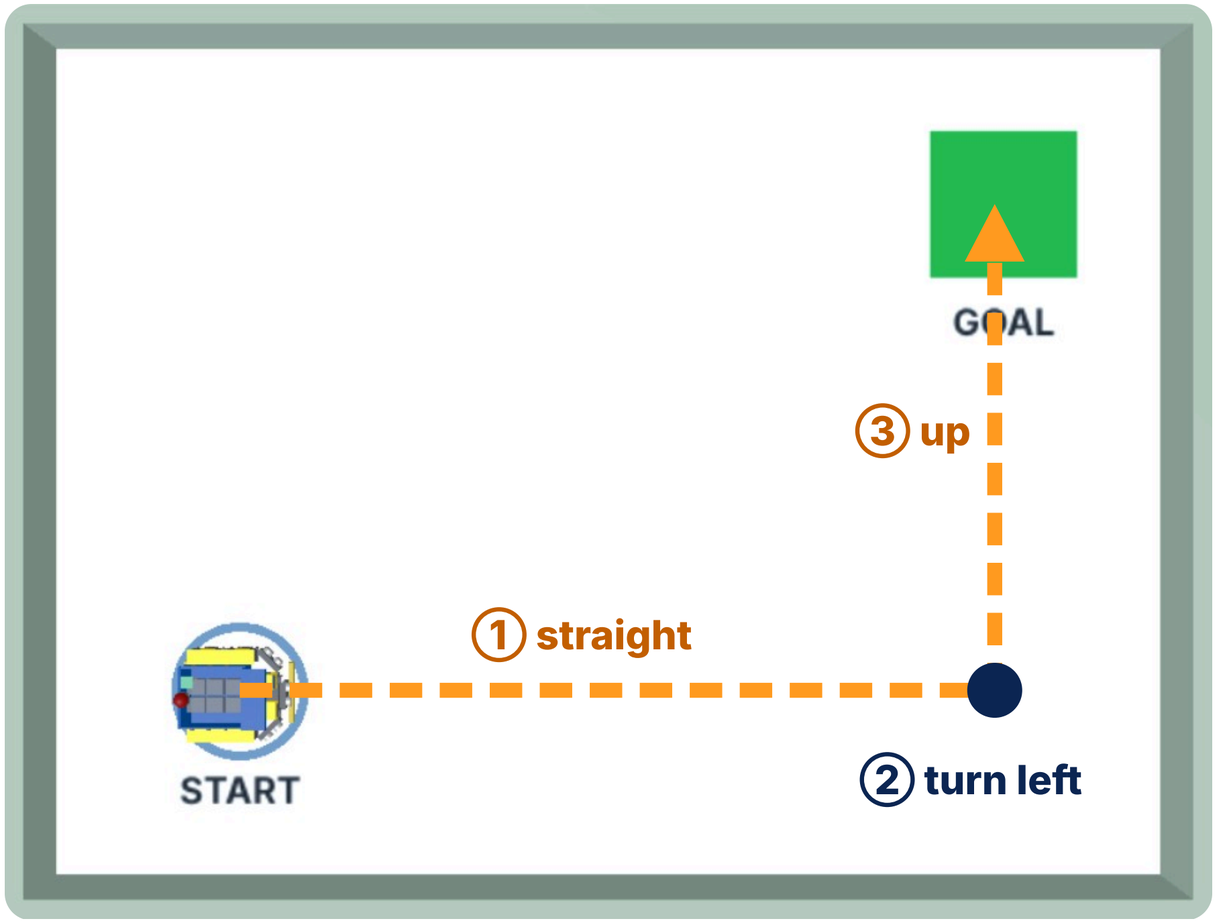
Field 240 × 180 cm with a wall around it.

START bottom left, robot facing **right**.

GOAL green square, 32 × 32 cm, top right.

Win robot centre inside the green: **+100**, within 3:00.





STEP 2 · PLAN

An L-shaped route

① Drive straight right until under the goal.

② Spin left a quarter turn (90°).

③ Drive straight up into the green.

Moves we know: straight and spin. No curves needed.

STEP 3 · CALCULATE

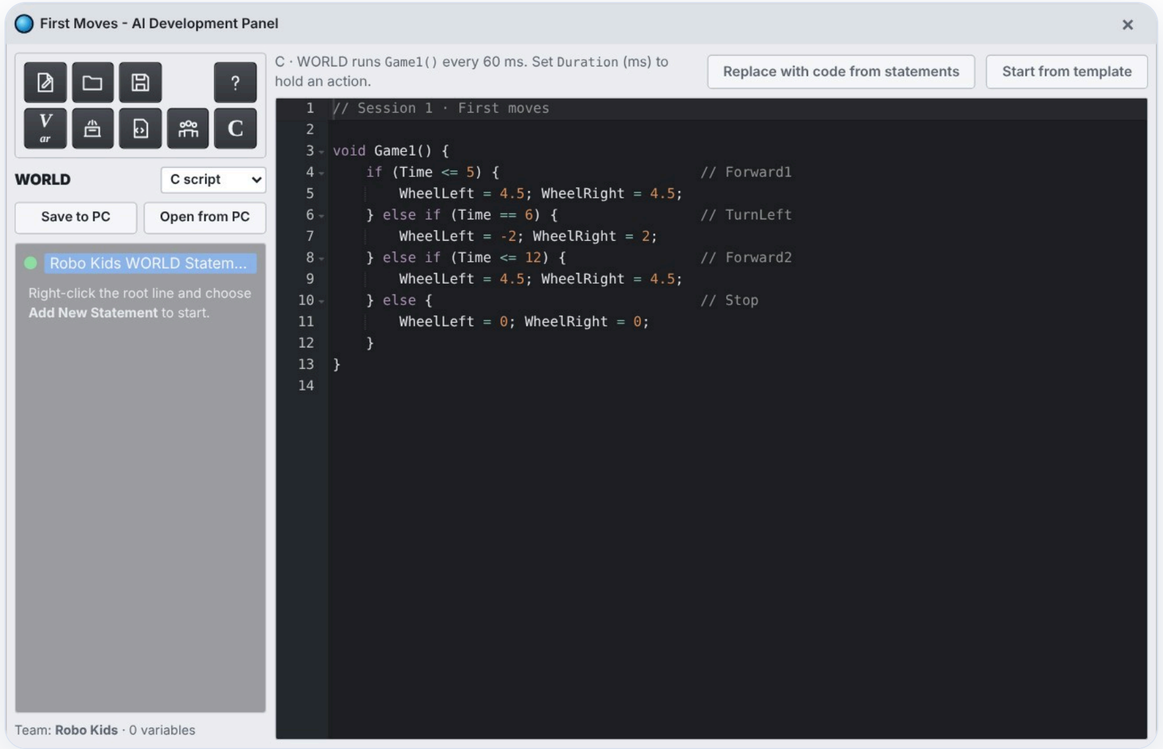
Turn the route into numbers

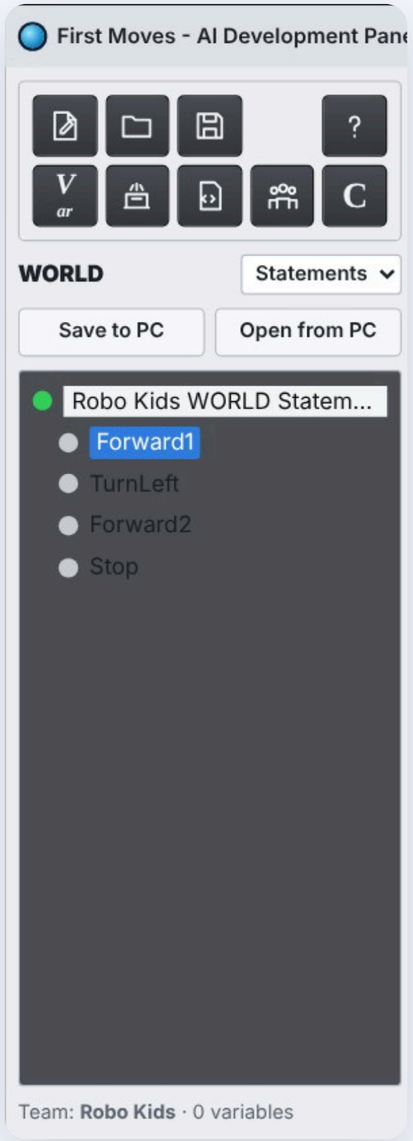
① **Straight**	Goal centre is ≈ 166 cm to the right. Wheels 4.5 = 27 cm/s. **6 s × 27 = 162 cm** → Time 0–5.
② **Turn**	Wheels −2 and 2 spin ≈ 92° each second. **1 second** → Time 6.
③ **Up**	The green starts 90 cm above. 90 ÷ 27 ≈ **3.3 s**. Give it extra time → Time 7–12.
④ **Stop**	No conditions: the safe else at the bottom.

STEP 4 · WRITE

Open the AI panel and choose C

1. **Learning** → Sessions → **Session 1**.
2. Press **AI**. First button = New Project: name it **First Moves**.
3. People button = **Team Name**, e.g. Robo Kids.
4. WORLD menu: choose **C script**.
5. Type the program in the code editor, starting with **Game1()**.





STEP 4 · WRITE

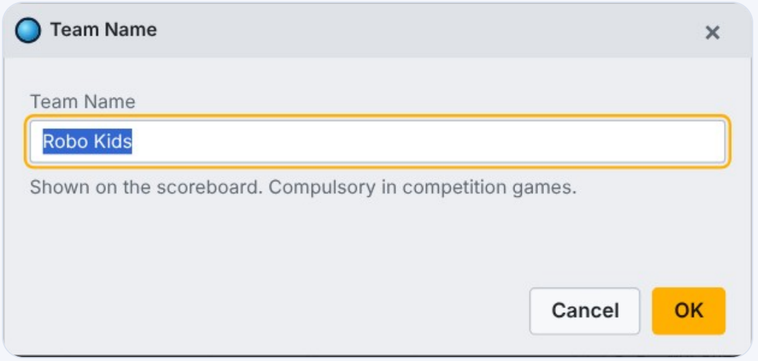
Set or change your Team Name

1 In the AI panel press the **people** button (2nd row, 4th).

2 Type the name: up to **24** characters.

3 Press **OK**. The bottom line shows **Team: Robo Kids**.

4 Press **Build** again so the robot gets the new name.



The team name shows on the **scoreboard**. In competition it is **compulsory**: no name = a warning, the second time you are disqualified from that game.

STEP 4 · WRITE

The whole program: four branches

```
// Session 1 · First moves

void Game1() {
    if (Time <= 5) {                          // Forward1
        WheelLeft = 4.5; WheelRight = 4.5;
    } else if (Time == 6) {                   // TurnLeft
        WheelLeft = -2; WheelRight = 2;
    } else if (Time <= 12) {                  // Forward2
        WheelLeft = 4.5; WheelRight = 4.5;
    } else {                                  // Stop
        WheelLeft = 0; WheelRight = 0;
    }
}
```

Forward1 Time <= 5 → 4.5 and 4.5

TurnLeft Time == 6 → −2 and 2

Forward2 Time <= 12 → 4.5 and 4.5

Stop else → 0 and 0

STEP 4 · WHY IT WORKS

From the top: the first true branch wins

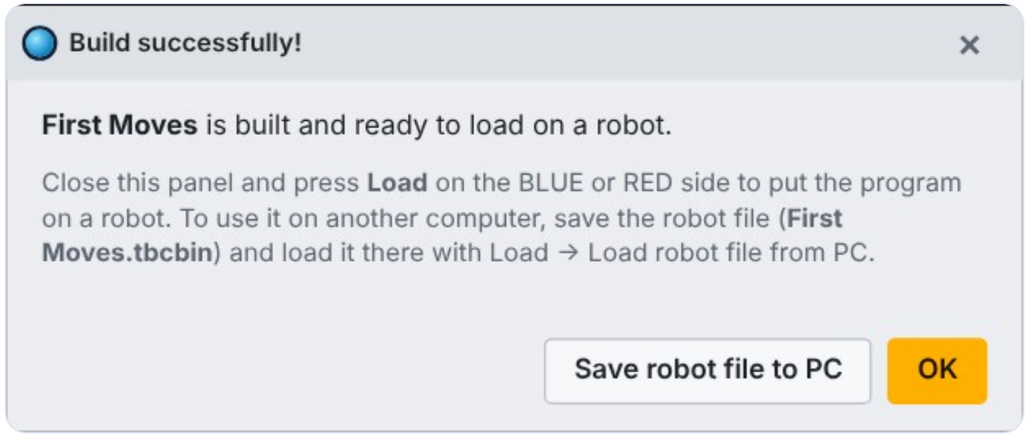
Time	Forward1 · ≤ 5	TurnLeft · == 6	Forward2 · ≤ 12	Runs
0 – 5	**true**	(skipped)	(skipped)	**Forward1**
6	false	**true**	(skipped)	**TurnLeft**
7 – 12	false	false	**true**	**Forward2**
13 +	false	false	false	**Stop** (else)

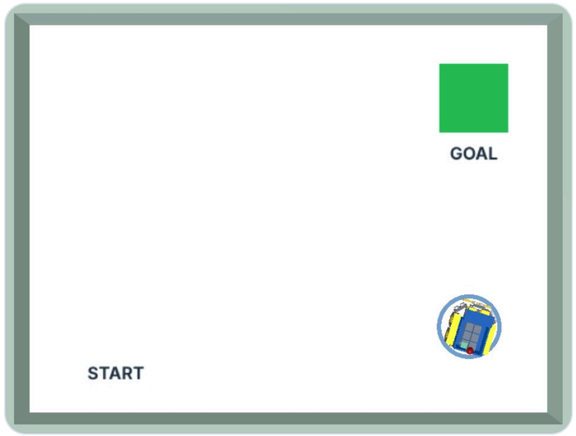
Swap Forward1 and Forward2 and the robot would never turn.

STEP 5 · TEST

Build, Load on BLUE, Play

1 Press **Build** → 2 OK, close the panel → 3 BLUE **Load** → First Moves → 4 Press ▶ **Play**





At **Time 6** the robot is under the goal and turning left.

KEEP YOUR WORK

Download your code to your PC

TO KEEP EDITING

Project file

AI panel → **Save to PC**

File: **First Moves.tbcbot.json**

Back later: **Open from PC** → choose the file.

READY TO RUN

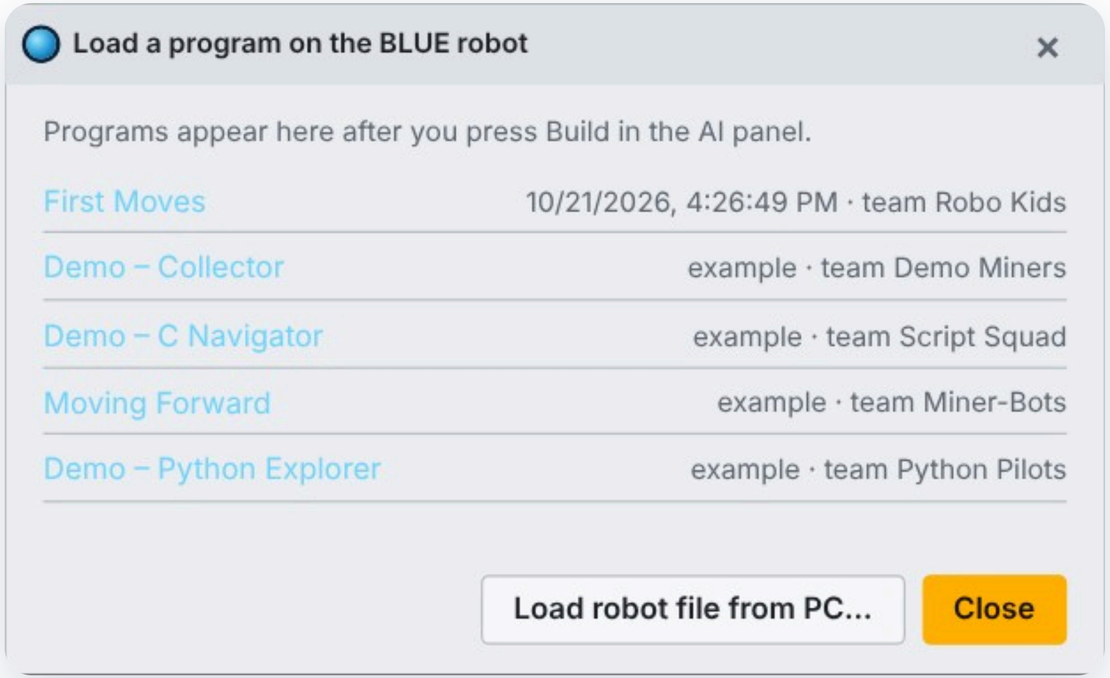
Robot file

Build → **Save robot file to PC**

File: **First Moves.tbcbin**

For another computer or a match. It cannot be edited.

The **disk** button (Save Project) keeps a copy in this browser only. Use **Save to PC** to be safe.



RUN YOUR CODE

Upload your code to the robot

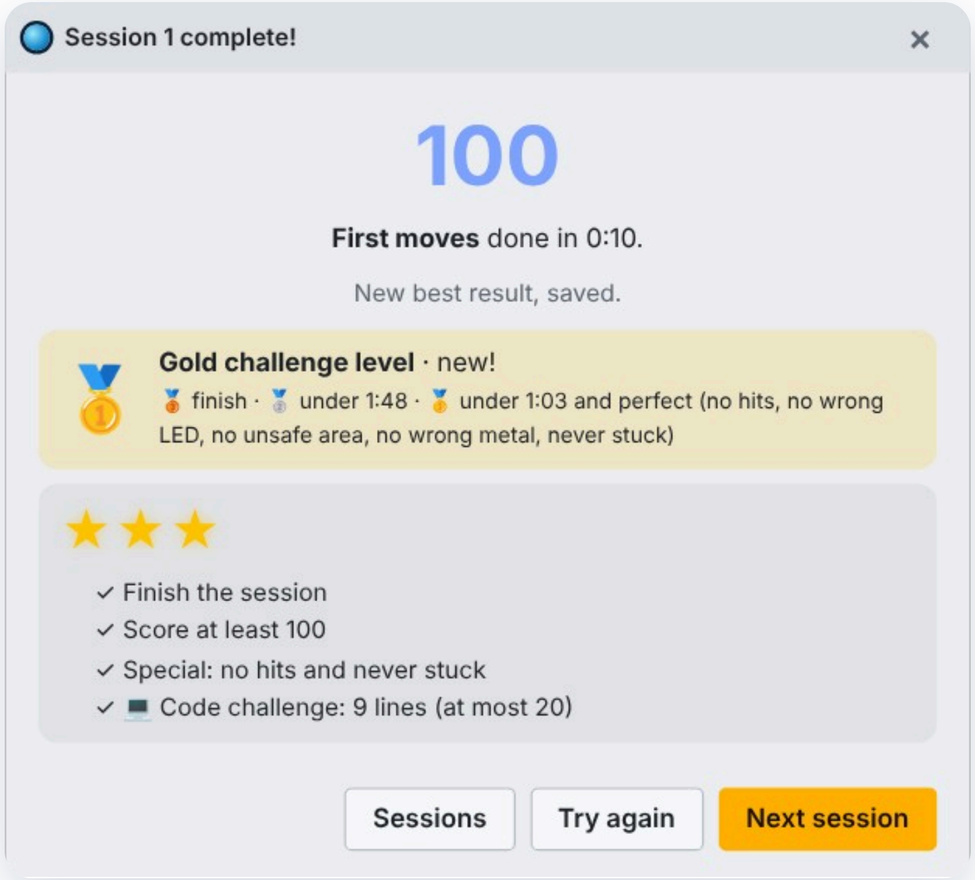
1 Press **Load** on the **BLUE** side.

2a Built here? Click **First Moves** in the list.

2b File on your PC? **Load robot file from PC...** → choose the **.tbcbin** (or .tbcbot.json).

3 Check the name under the BLUE buttons.

4 Press ▶ **Play**.



RESULT · TESTED

100 points in 0:10 · Gold

Medals · Bronze: finish · Silver: under 1:48 · Gold: under 1:03 and perfect.

Stars · finish · score 100 · no hits and never stuck.

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

DID NOT WORK?

Find the problem, change one thing

What you see	What to do
Build error with a line number	Check the ; at the end of each line and the { } brackets
Robot does not move at all	Build again, then Load on BLUE: the name shows under the buttons
Turns too early or too late	Forward1: change **Time <= 5** or the speed (4.5)
Turns too much or too little	TurnLeft: −1.9 and 1.9 turns ≈ 87° in a second
Stops below the goal	Forward2: make the 12 bigger
Never turns	Check the order: Forward1, TurnLeft, Forward2, then else

TRY IT YOURSELF

Three challenges

Other way round

Turn left first, drive up, then spin **right** (2, −2) and drive right.

Hard mode 🔥

Tick Hard mode in Sessions and run your program again.

Next session

Presentation 2 · **Three targets**: visit 1, 2, 3 in order.