PRESENTATION 18 · LEARNING WITH C

Search and return

Leave home, find two objects, come back and deposit: the green-deposit program with a new number.

TheBotCup 2026 · AiBotCup Simulator



RECAP · PRESENTATION 14

Collect, then deposit on green

Deposit, on top

both sensors green · LoadedObjects == full

Six collects

each with LoadedObjects < full

Last time “full” was 3. Today it is **2**, and the green box is where we **start**.

PART 1

The mission: Session 18

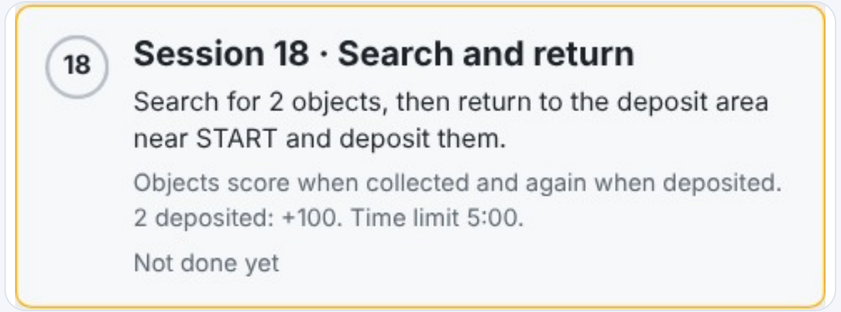
Out, find two, back home

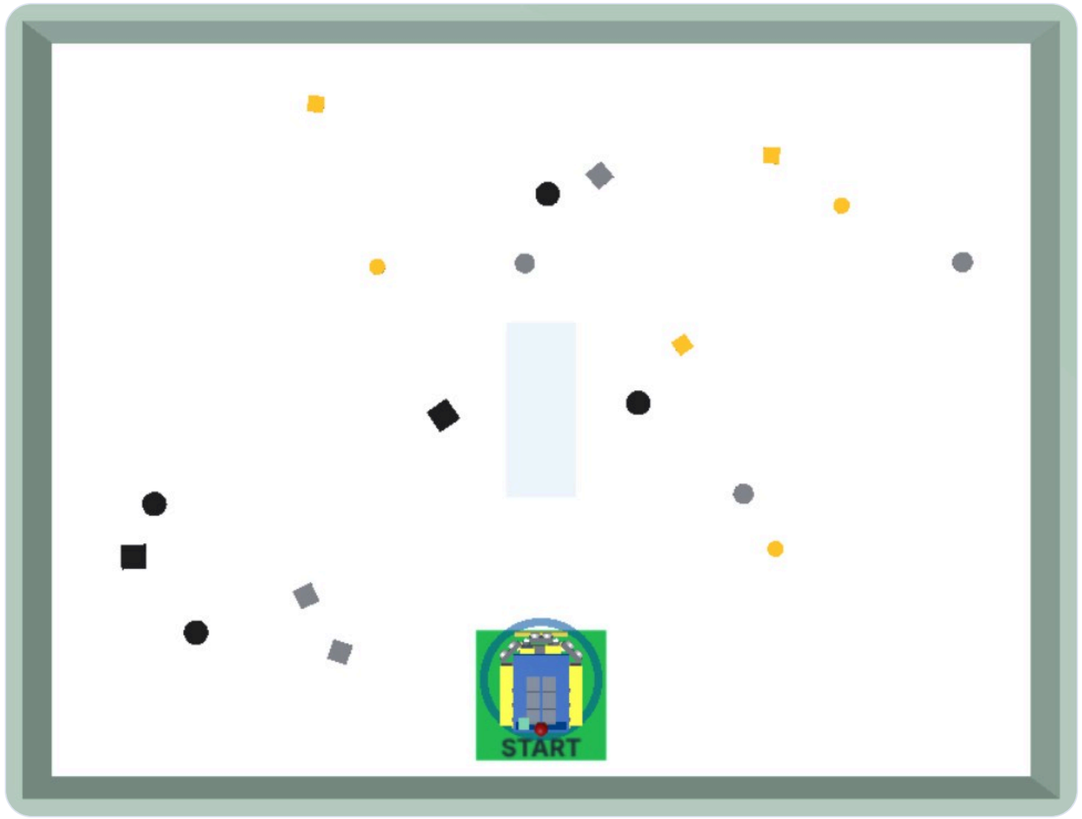
SESSION 18 · SEARCH AND RETURN

Find 2, return to the deposit area near START

Objects score when collected and again when deposited. 2 deposited: **+100**.

Time limit **5:00**. Code challenge: at most **45** lines.





THE FIELD

Home is green

The robot **starts inside** the green deposit box.

Metals all over the field; several close to home.

It must **leave**, then **find its way back**.

PART 2

Starting on green

Why doesn’t it deposit at once?

THE IDEA · CONDITIONS WORK TOGETHER

Green is not enough on its own

Moment	On green?	LoadedObjects	Deposit?
At START	yes	0	no, drives on
Searching	no	0 or 1	no
Back home	yes	2	**yes**

Every part joined by && must be true. Deposit needs LoadedObjects == 2 **&&** onGreen().

PART 3

Step by step in C

Presentation 14's program, 3 → 2

STEP 1 · THE WHOLE PROGRAM

Nine branches, “full” = 2

Branch	Condition	Also	Action
1 Deposit	onGreen()	LoadedObjects == 2	deposit()
2 IronLeft	left iron	LoadedObjects < 2	collect()
3 IronRight	right iron	LoadedObjects < 2	collect()
4 SilverLeft	left silver	LoadedObjects < 2	collect()
5 SilverRight	right silver	LoadedObjects < 2	collect()
6 GoldLeft	left gold	LoadedObjects < 2	collect()
7 GoldRight	right gold	LoadedObjects < 2	collect()
8 Ahead	US_Front ≤ 20	—	−2 / 2 · Duration 700
9 Forward	the else	—	4.5 / 4.5

STEP 2 · LINES 3–23

The tools from Presentation 14

```
3  int iron(int r, int g, int b) { return r <= 60 && g <= 60 && b <= 60; }
4  int silver(int r, int g, int b) {
5      return r >= 90 && r <= 135 && g >= 92 && g <= 137 && b >= 98 && b <= 142;
6  }
7  int gold(int r, int g, int b) { return r >= 195 && g >= 135 && g <= 200 && b <= 80; }
8  int green(int r, int g, int b) { return r <= 75 && g >= 130 && g <= 200 && b >= 40 && b <= 110; }
9  int onGreen() {
10     return green(CSLeft_R, CSLeft_G, CSLeft_B) && green(CSRight_R, CSRight_G, CSRight_B);
11 }
12
13 void collect() {
14     WheelLeft = 0; WheelRight = 0;
15     LED_1 = 2; LED_R = 255; LED_G = 0; LED_B = 0;    // flashing red
16     Duration = 3200;                                 // hold 3.2 seconds
17 }
18
19 void deposit() {
20     WheelLeft = 0; WheelRight = 0;
21     LED_1 = 1; LED_R = 0; LED_G = 255; LED_B = 0;    // steady green
22     Duration = 3200;                                 // hold 3.2 seconds
23 }
```

STEP 3 · LINES 25–28

Deposit when home with 2

```
25  void Game1() {
26      LED_1 = 0;
27      if (LoadedObjects == 2 && onGreen()) {                    // Deposit
28          deposit();
```

LoadedObjects == 2: the only change in this branch.

At START the robot is on green, but the load is 0: false.

STEP 4 · LINES 29–47

Collect while fewer than 2

```
29      } else if (LoadedObjects < 2 && iron(CSLeft_R, CSLeft_G, CSLeft_B)) {     // IronLeft
30          collect();
31      } else if (LoadedObjects < 2 && iron(CSRight_R, CSRight_G, CSRight_B)) {  // IronRight
32          collect();
33      } else if (LoadedObjects < 2 && silver(CSLeft_R, CSLeft_G, CSLeft_B)) {   // SilverLeft
34          collect();
35      } else if (LoadedObjects < 2 && silver(CSRight_R, CSRight_G, CSRight_B)) { // SilverRight
36          collect();
37      } else if (LoadedObjects < 2 && gold(CSLeft_R, CSLeft_G, CSLeft_B)) {     // GoldLeft
38          collect();
39      } else if (LoadedObjects < 2 && gold(CSRight_R, CSRight_G, CSRight_B)) {  // GoldRight
40          collect();
41      } else if (US_Front <= 20) {                                              // Ahead
42          WheelLeft = -2; WheelRight = 2;
43          Duration = 700;
44      } else {                                                                  // Forward
45          WheelLeft = 4.5; WheelRight = 4.5;
46      }
47  }
```

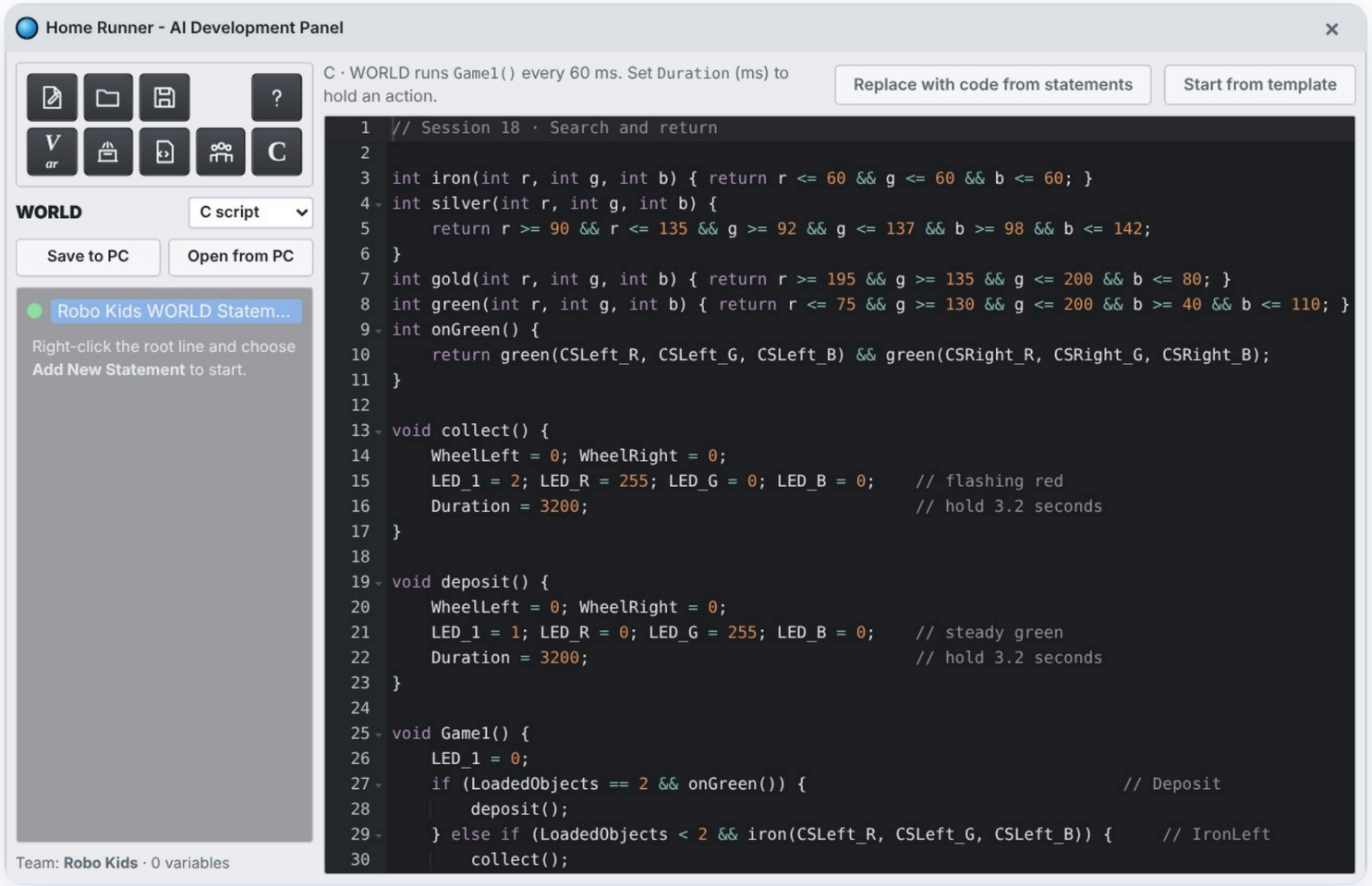
All six collects: **LoadedObjects < 2 &&** …

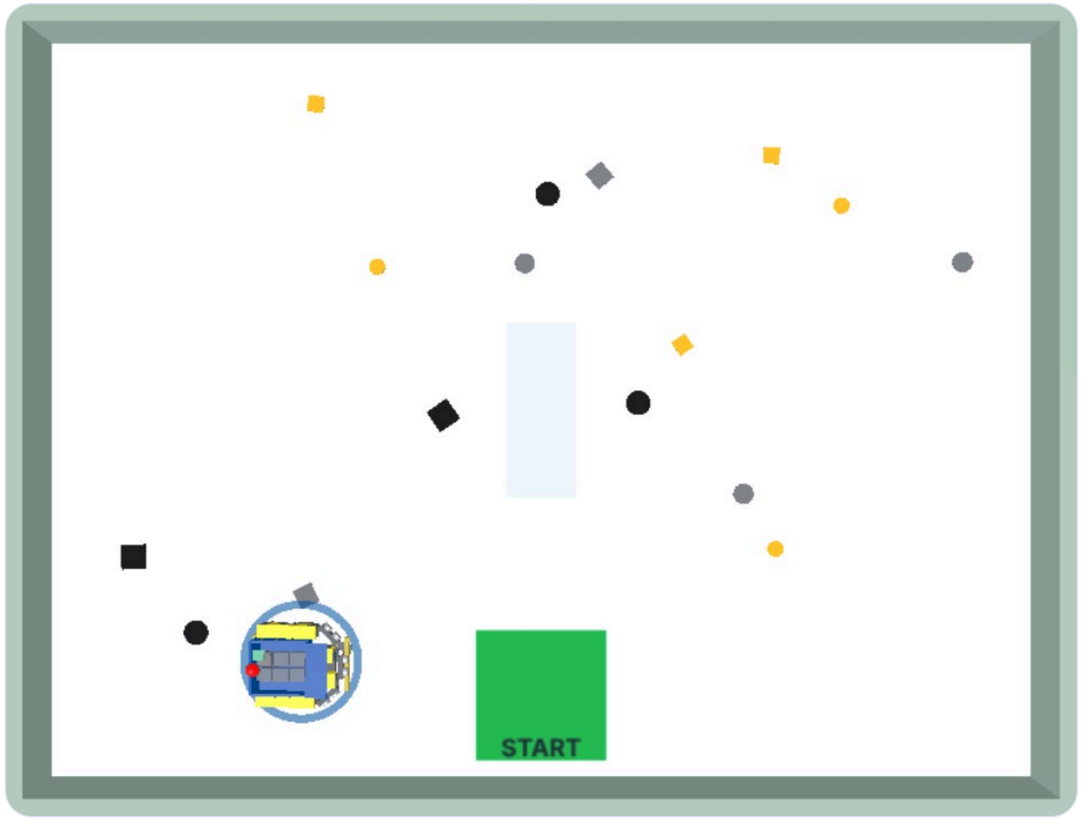
Ahead and the else **Forward**: unchanged.

STEP 5 · BUILD AND LOAD

The whole program, built

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





Colour Sensors

Left R	111	(CSLeft_R)
G	115	(CSLeft_G)
B	117	(CSLeft_B)

left sees this colour

Right R	227	(CSRight_R)
G	233	(CSRight_G)
B	239	(CSRight_B)

right sees this colour

Robot

Wheels L **0.0** · R **0.0** · LED **2**

Load **1**/6 · Fe 1 Ag 0 Au 0

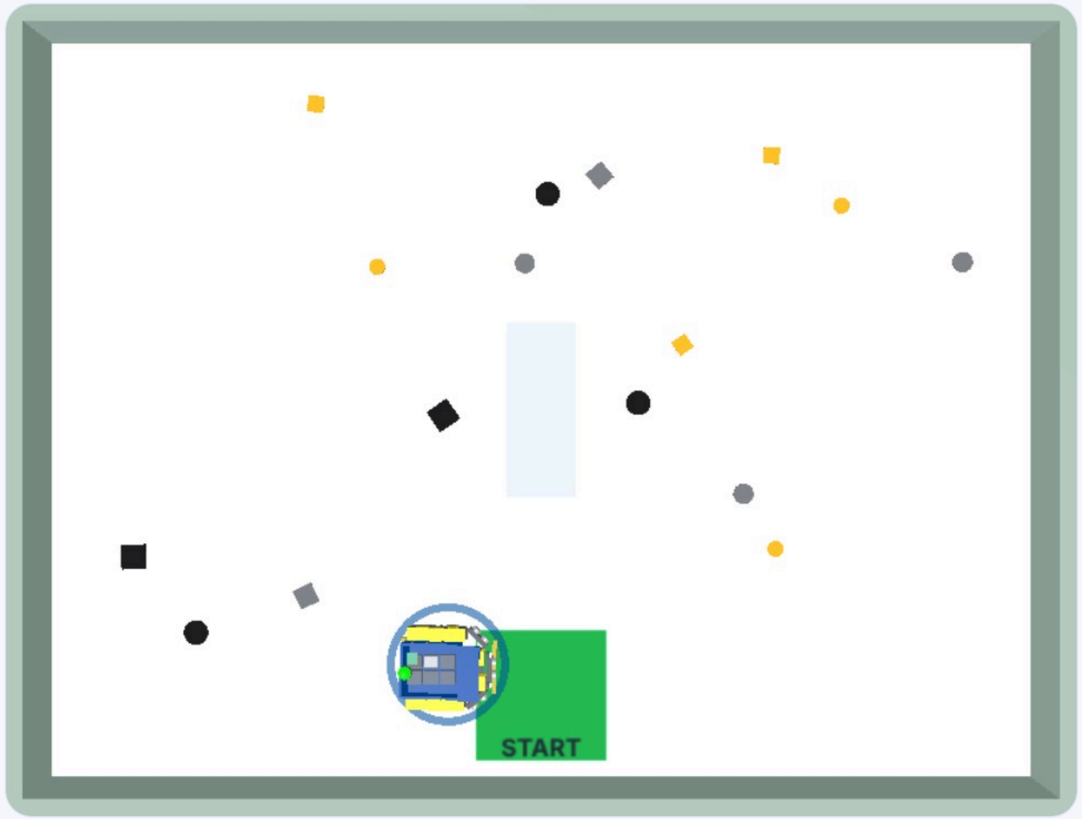
On **collecting 1.9 / 3 s**

STEP 6 · RUN · 15.3 S

The second object

Load **1** (an iron at 7.2 s). Left sees silver: **SilverLeft**.

Now LoadedObjects = **2**: no more collects.



Colour Sensors

Left R	29	(CSLeft_R)
G	159	(CSLeft_G)
B	71	(CSLeft_B)

left sees this colour

Right R	31	(CSRight_R)
G	160	(CSRight_G)
B	73	(CSRight_B)

right sees this colour

Robot

Wheels L **0.0** · R **0.0** · LED **1**

Load **2**/6 · Fe 1 Ag 1 Au 0

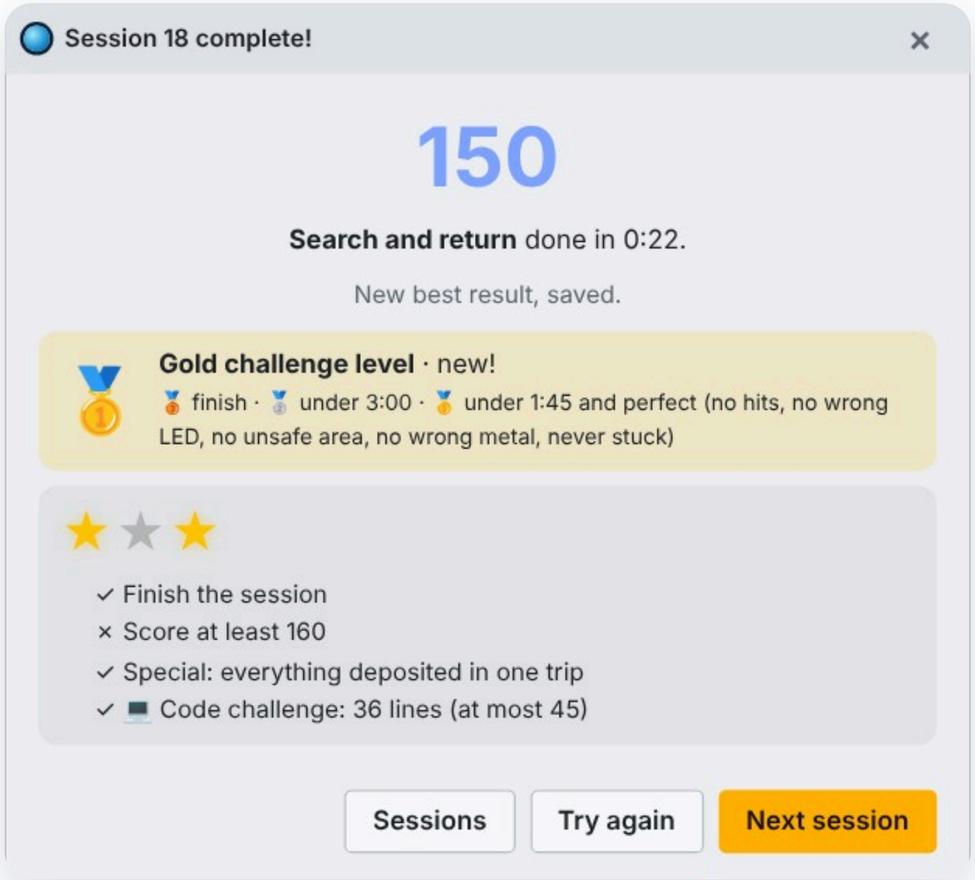
On **deposit area · depositing 1.9 / 3 s**

STEP 7 · RUN · 19.8 S

Back home

Both sensors green, load **2**: **Deposit**.

Both metals were near home, so the trip was short.



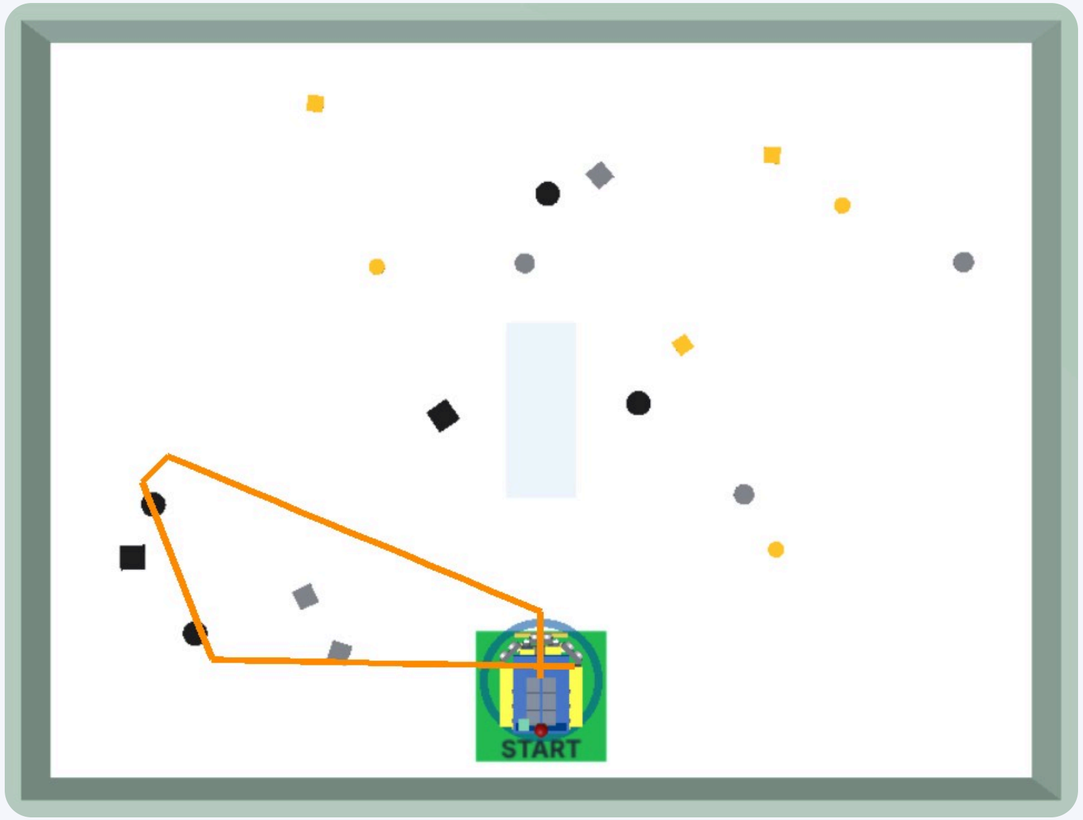
RESULT · TESTED

150 points in 0:22 · Gold

Score · iron 10 + silver 15, twice, + bonus 100.

Special · one trip. The “score 160” star was not reached.

Code challenge · at most 45 lines. We used 36.



THE PATH · RECORDED

A loop out and home

Out to the left, two metals, back into the box.

One program, one changed number: **reuse** what works.

DID NOT WORK?

Check first, then tune carefully

What you see	What to do
Collects a third object	Every collect needs **LoadedObjects < 2**
Comes home but never deposits	Deposit must say **== 2**, not == 3
Stops at START with nothing	Deposit is missing **LoadedObjects == 2 &&**
Old program still on the robot	**Build** again, then **Load** on BLUE

NEXT TIME

Presentation 19 · Mini mission

Water, land and deposit together: **everything we have learned** in one program.