PRESENTATION 14 · LEARNING WITH C

Green deposit

Collect three, then bring them home: a built-in counter and a green-LED deposit.

TheBotCup 2026 · AiBotCup Simulator



RECAP

Two things we already know

Collect

A colour range finds iron, silver or gold. collect() stops and flashes the red LED.

Variables

A named number, used in any condition (Presentation 7).

Today the robot gives us a variable **for free**.

PART 1

The mission: Session 14

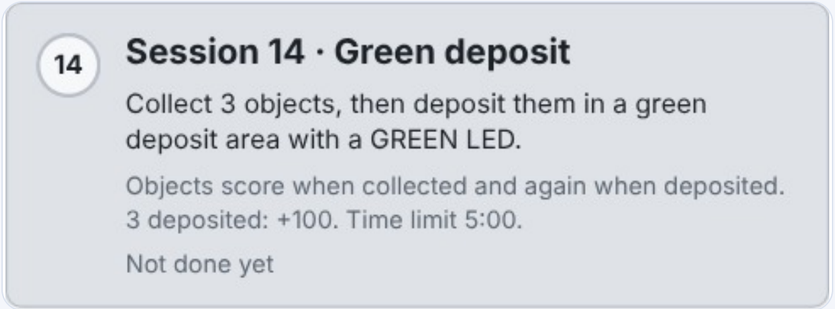
Collect three, deposit on green

SESSION 14 · GREEN DEPOSIT

Collect 3, then deposit with a GREEN LED

Objects score when **collected** and again when **deposited**. 3 deposited: **+100**.

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





THE FIELD

Two green corners

START at the bottom centre.

Iron, silver and gold all over the field.

Two **green deposit areas** in the top corners: either one works.

PART 2

Two new ideas

How full am I? Where do I unload?

IDEA 1 · A BUILT-IN VARIABLE

LoadedObjects counts for us

0 → 3

+1 each time an object is **really on board**

→ 0

back to 0 after a **deposit**

No need to create it. Just use it in a condition: **LoadedObjects < 3** or **LoadedObjects == 3** (two equals signs).

IDEA 2 · THE DEPOSIT

Both sensors on green, then Deposit

Green range

R 0–75 · G 130–200 · B 40–110, on **left and right**.

deposit()

Stops, **LED_1 = 1** (steady) in **green**, 3.2 s: the robot unloads.

Both sensors on green means the robot is fully inside the area.

PART 3

Step by step in C

Nine branches, Deposit on top

STEP 1 · THE WHOLE PROGRAM

Deposit, six collects, two drivers

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

STEP 2 · LINES 3–11

Is this green? Are both sensors on green?

```
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 }
```

green(r, g, b): R ≤ 75, G 130–200, B 40–110.

onGreen() calls green() for the left && the right sensor.

STEP 3 · LINES 13–23

Two actions: collect() and deposit()

```
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  }
```

collect(): LED_1 = 2, flashing **red**.

deposit(): LED_1 = 1, **steady green** (0, 255, 0).

Both stop the wheels and hold for **3200** ms.

STEP 4 · LINES 25–28

Deposit comes first

```
25  void Game1() {
26      LED_1 = 0;
27      if (LoadedObjects == 3 && onGreen()) {                    // Deposit
28          deposit();
```

LoadedObjects == 3: two equals signs ask “is it equal?”.

&& **onGreen()** → deposit(). First in the chain, so it always wins.

STEP 5 · LINES 29–40

Collect only while not full

```
29      } else if (LoadedObjects < 3 && iron(CSLeft_R, CSLeft_G, CSLeft_B)) {     // IronLeft
30          collect();
31      } else if (LoadedObjects < 3 && iron(CSRight_R, CSRight_G, CSRight_B)) {  // IronRight
32          collect();
33      } else if (LoadedObjects < 3 && silver(CSLeft_R, CSLeft_G, CSLeft_B)) {   // SilverLeft
34          collect();
35      } else if (LoadedObjects < 3 && silver(CSRight_R, CSRight_G, CSRight_B)) { // SilverRight
36          collect();
37      } else if (LoadedObjects < 3 && gold(CSLeft_R, CSLeft_G, CSLeft_B)) {     // GoldLeft
38          collect();
39      } else if (LoadedObjects < 3 && gold(CSRight_R, CSRight_G, CSRight_B)) {  // GoldRight
40          collect();
```

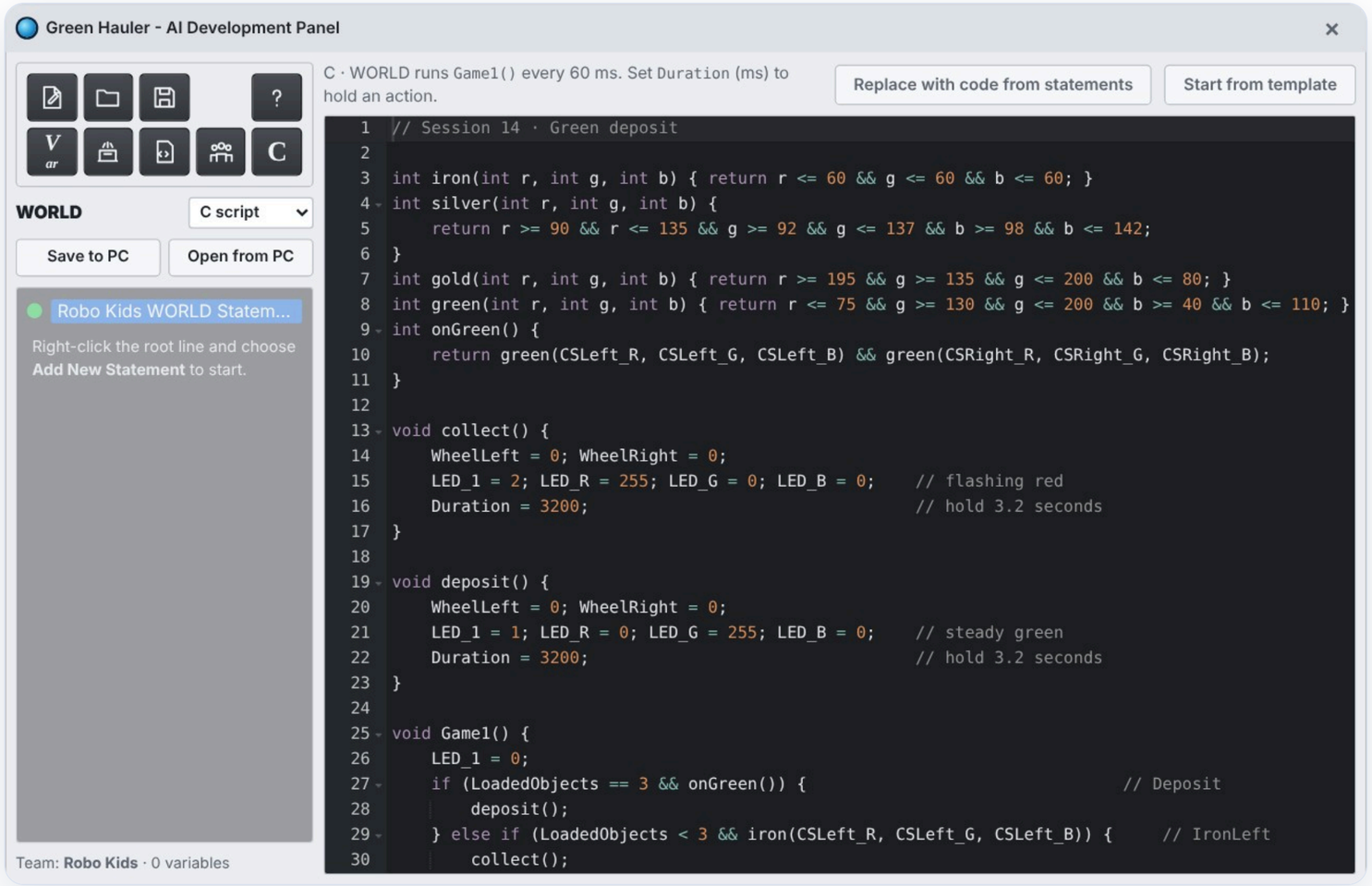
Every collect starts with **LoadedObjects < 3 &&**.

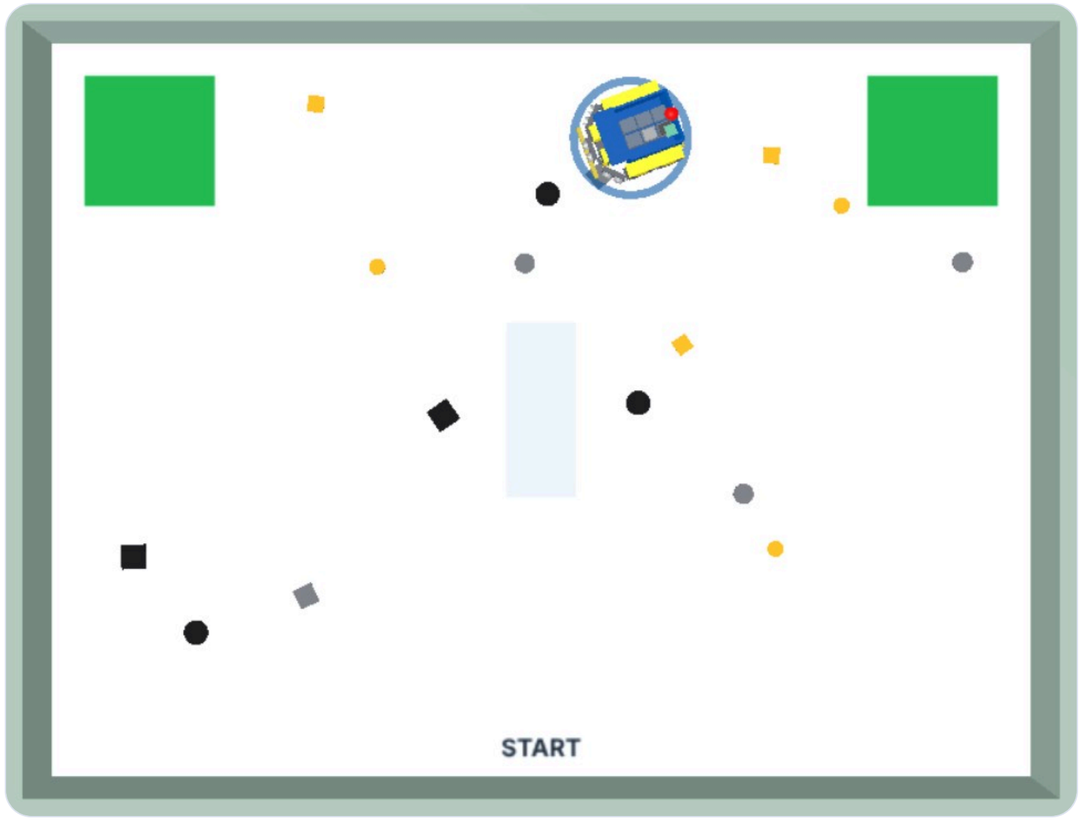
Then **Ahead** and the else **Forward** as before (lines 41–47).

STEP 6 · BUILD AND LOAD

The whole program, built

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





Colour Sensors

Left R	108	(CSLeft_R)
G	110	(CSLeft_G)
B	120	(CSLeft_B)

left sees this colour

Right R	224	(CSRight_R)
G	235	(CSRight_G)
B	238	(CSRight_B)

right sees this colour

Robot

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

Load **2**/6 · Fe 1 Ag 1 Au 0

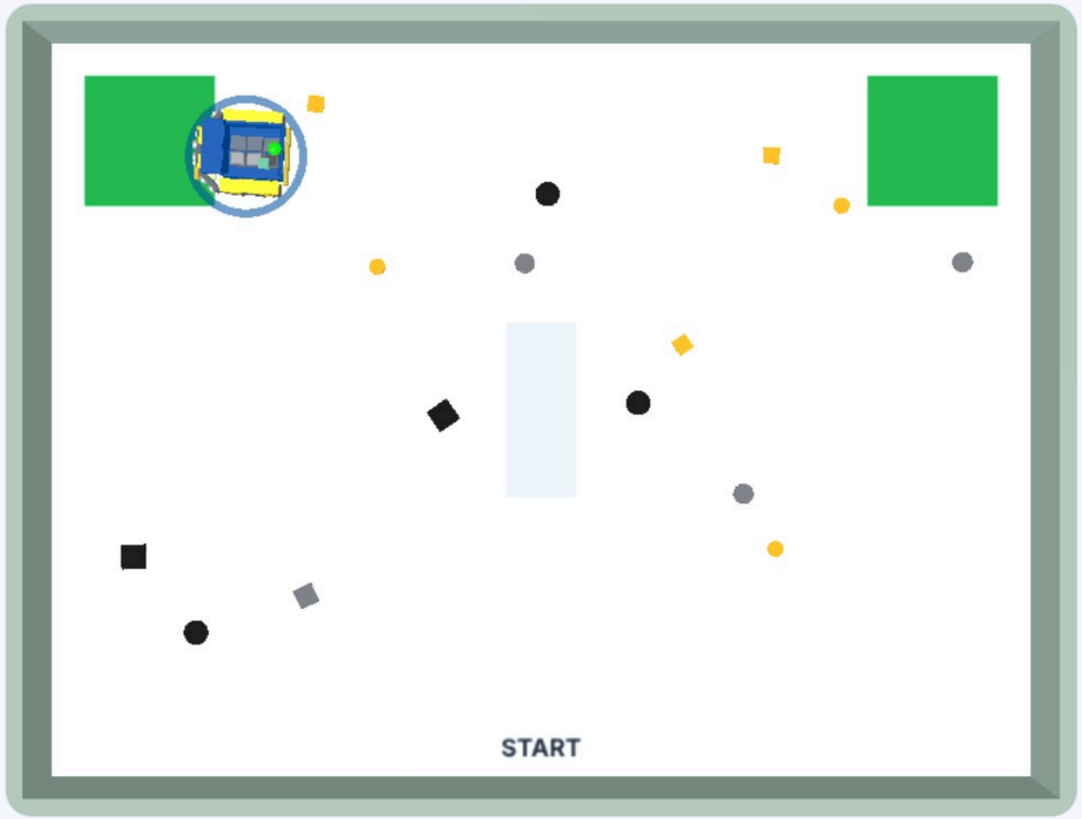
On **collecting 1.9 / 3 s**

STEP 7 · RUN · 32.4 S

The third object

Load **2** < 3, left sees silver (110, 110, 117): **SilverLeft**.

After this one, LoadedObjects is **3**: no more collects.



Colour Sensors

Left R	27	(CSLeft_R)
G	160	(CSLeft_G)
B	68	(CSLeft_B)

left sees this colour

Right R	32	(CSRight_R)
G	157	(CSRight_G)
B	71	(CSRight_B)

right sees this colour

Robot

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

Load **3**/6 · Fe 1 Ag 2 Au 0

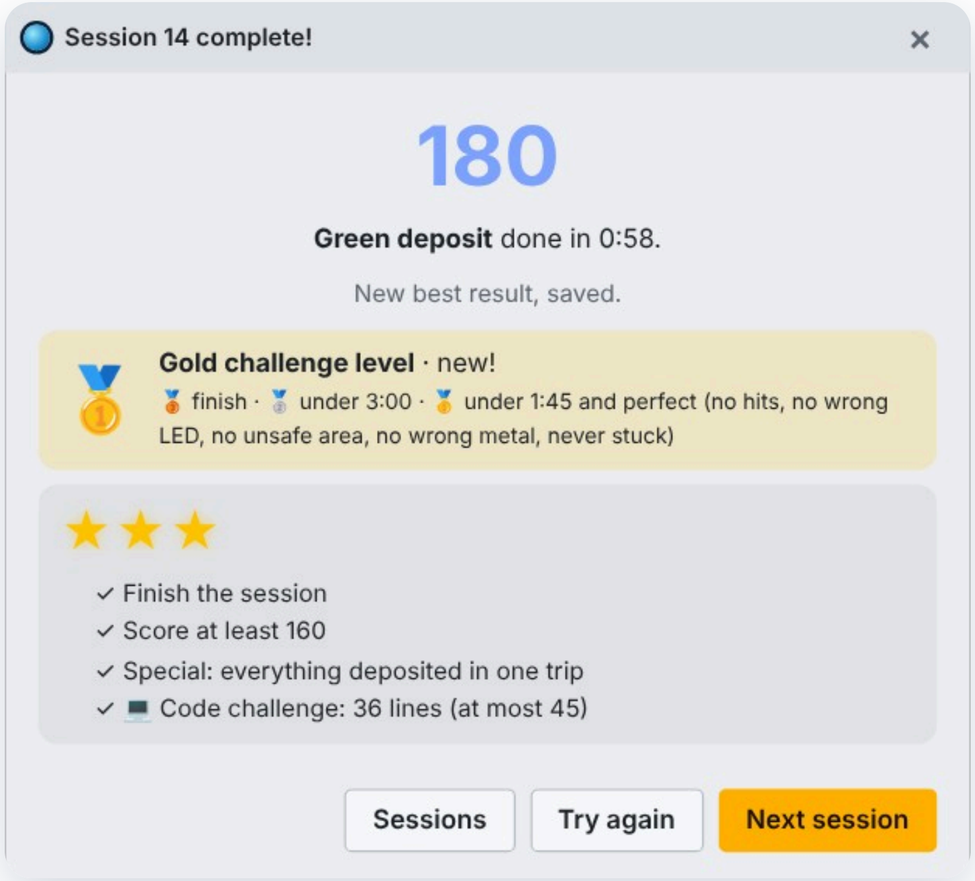
On **deposit area · depositing 1.4 / 3 s**

STEP 8 · RUN · 54.7 S

Home on green

Both sensors green, load **3**: **Deposit**, green LED.

Full robot drives over metals until it finds green.



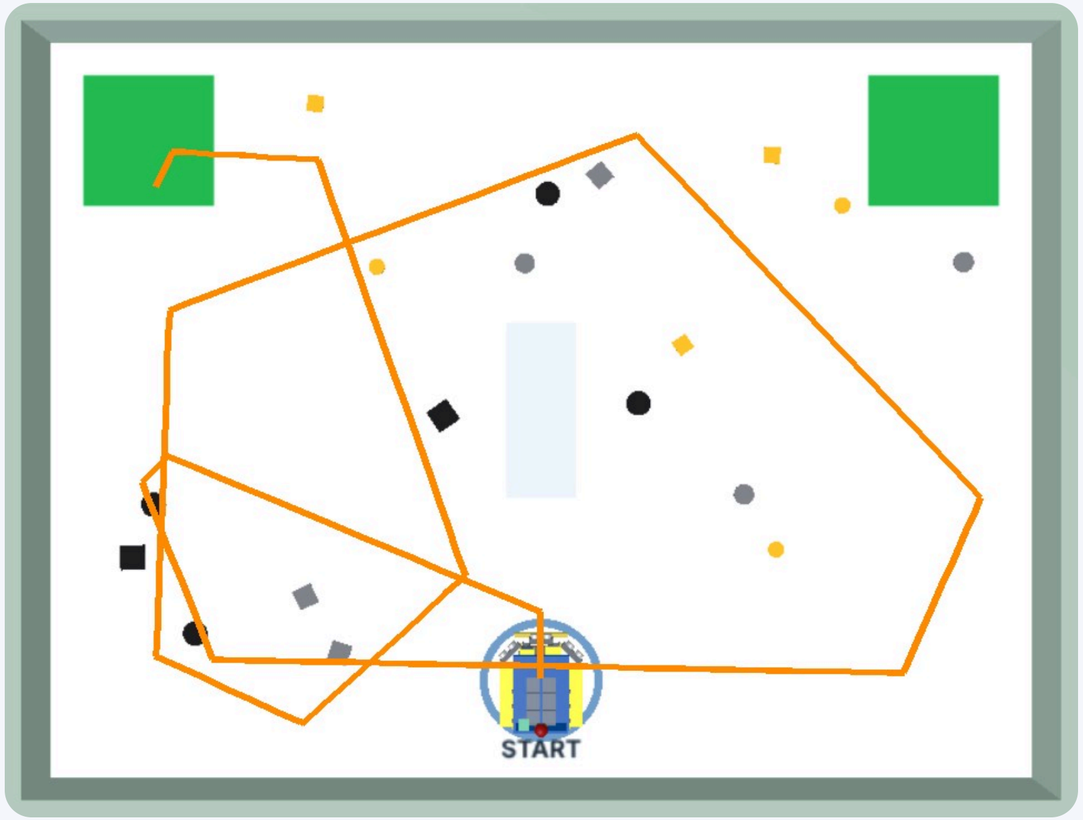
RESULT · TESTED

180 points in 0:58 · Gold

Score · collected 40 + deposited 40 + bonus 100.

Special star · everything deposited in one trip.

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



THE PATH · RECORDED

Collect, then search for green

The path ends in the **top-left** green corner.

One number, **LoadedObjects**, switches the robot from collecting to going home.

DID NOT WORK?

Check first, then tune carefully

What you see	What to do
Keeps collecting after 3	Add **LoadedObjects < 3 &&** to every collect
Drives over green when full	Use **==** (two signs) and check the green range
Deposits at the edge, nothing happens	onGreen() checks **both** sensors, not just one
Old program still on the robot	**Build** again, then **Load** on BLUE

NEXT TIME

Presentation 15 · One of each

Exactly one iron, one silver and one gold, with **LoadedIron, LoadedSilver, LoadedGold**.