PRESENTATION 13 · LEARNING WITH C

Water pond

Mining inside the water: a blue LED, double points, and branches that know where the pond is.

TheBotCup 2026 · AiBotCup Simulator



RECAP · COLLECTING METALS

A colour range finds the metal

Iron

R, G, B all 0–60

Silver

R 90–135 · G 92–137 · B 98–142

Gold

R 195–255 · G 135–200 · B 0–80

One branch per metal and per sensor, calling **collect()**, which stops and flashes red. Today the metals are **under water**.

PART 1

The mission: Session 13

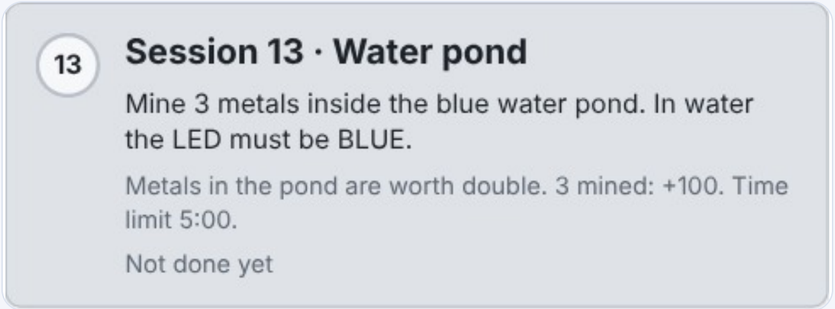
Three metals from the water

SESSION 13 · WATER POND

Mine 3 metals inside the pond

In water the LED must be **BLUE**. Pond metals are worth **double**.

3 mined: **+100**. Time limit **5:00**. Code challenge: at most **45** lines.





THE FIELD

A big blue pond

The **pond** fills the left side, with iron, silver and gold inside.

START is just right of the pond.

Metals on **dry land** do not help this mission.

PART 2

Two new ideas

A water collect, and knowing where the pond is

IDEA 1 · A NEW FUNCTION

In water, collect with a blue LED

ON DRY LAND

collect()

Red LED. Normal points.

IN THE POND

collectWater()

Blue LED. **Double** points: iron 20, silver 30, gold 40.

A red-LED collect in the pond is **not counted**. The function must match the place.

IDEA 2 · POSITION IN THE CONDITION

Is the robot in the pond?

The pond’s box

The pond covers **X 8–104** and **Y 8–132**. Every pond collect also tests **PositionX ≤ 104** && **PositionY ≤ 132**.

Stay in the water

Near the pond, if the left sensor sees **white floor**, turn back in.

PART 3

Step by step in C

Nine branches

STEP 1 · THE WHOLE PROGRAM

Six pond collects, then three drivers

Branch	Condition	Action
1 PondIronLeft	left iron · in the pond	collectWater()
2 PondIronRight	right iron · in the pond	collectWater()
3 PondSilverLeft	left silver · in the pond	collectWater()
4 PondSilverRight	right silver · in the pond	collectWater()
5 PondGoldLeft	left gold · in the pond	collectWater()
6 PondGoldRight	right gold · in the pond	collectWater()
7 StayIn	left white · near the pond	−2 / 2 · Duration 700
8 Ahead	US_Front ≤ 20	−2 / 2 · Duration 700
9 Forward	the else	4.5 / 4.5

STEP 2 · LINES 3–10 · NEW IN CODE

Functions that answer yes or no

```
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 white(int r, int g, int b) { return r >= 200 && g >= 200 && b >= 200; }
9  int inPond() { return PositionX <= 104 && PositionY <= 132; }
10 int nearPond() { return PositionX <= 112 && PositionY <= 140; }
```

iron(r, g, b) gets three numbers and **returns** 1 when all are ≤ 60.

inPond() needs no numbers: it reads PositionX and PositionY itself.

Write the colour range **once**, use it for the left and the right sensor.

STEP 3 · LINES 12–16

collectWater(): stop and flash blue

```
12  void collectWater() {
13      WheelLeft = 0; WheelRight = 0;
14      LED_1 = 2; LED_R = 0; LED_G = 0; LED_B = 255;    // flashing blue
15      Duration = 3200;                                  // hold 3.2 seconds
16  }
```

Wheels 0: stop on the metal.

LED_1 = 2, colour **0, 0, 255**: flashing blue.

Duration = 3200: hold it 3.2 s.

STEP 4 · LINES 18–31

Six pond collects: metal && in the pond

```
18  void Game1() {
19      LED_1 = 0;
20      if (inPond() && iron(CSLeft_R, CSLeft_G, CSLeft_B)) {                // PondIronLeft
21          collectWater();
22      } else if (inPond() && iron(CSRight_R, CSRight_G, CSRight_B)) {      // PondIronRight
23          collectWater();
24      } else if (inPond() && silver(CSLeft_R, CSLeft_G, CSLeft_B)) {       // PondSilverLeft
25          collectWater();
26      } else if (inPond() && silver(CSRight_R, CSRight_G, CSRight_B)) {    // PondSilverRight
27          collectWater();
28      } else if (inPond() && gold(CSLeft_R, CSLeft_G, CSLeft_B)) {         // PondGoldLeft
29          collectWater();
30      } else if (inPond() && gold(CSRight_R, CSRight_G, CSRight_B)) {      // PondGoldRight
31          collectWater();
```

Each branch asks **inPond()** && one metal for one sensor.

Only the sensor and the metal change from line to line.

STEP 5 · LINES 32–41

StayIn, Ahead and Forward

```
32      } else if (nearPond() && white(CSLeft_R, CSLeft_G, CSLeft_B)) {      // StayIn
33          WheelLeft = -2; WheelRight = 2;
34          Duration = 700;
35      } else if (US_Front <= 20) {                                         // Ahead
36          WheelLeft = -2; WheelRight = 2;
37          Duration = 700;
38      } else {                                                             // Forward
39          WheelLeft = 4.5; WheelRight = 4.5;
40      }
41  }
```

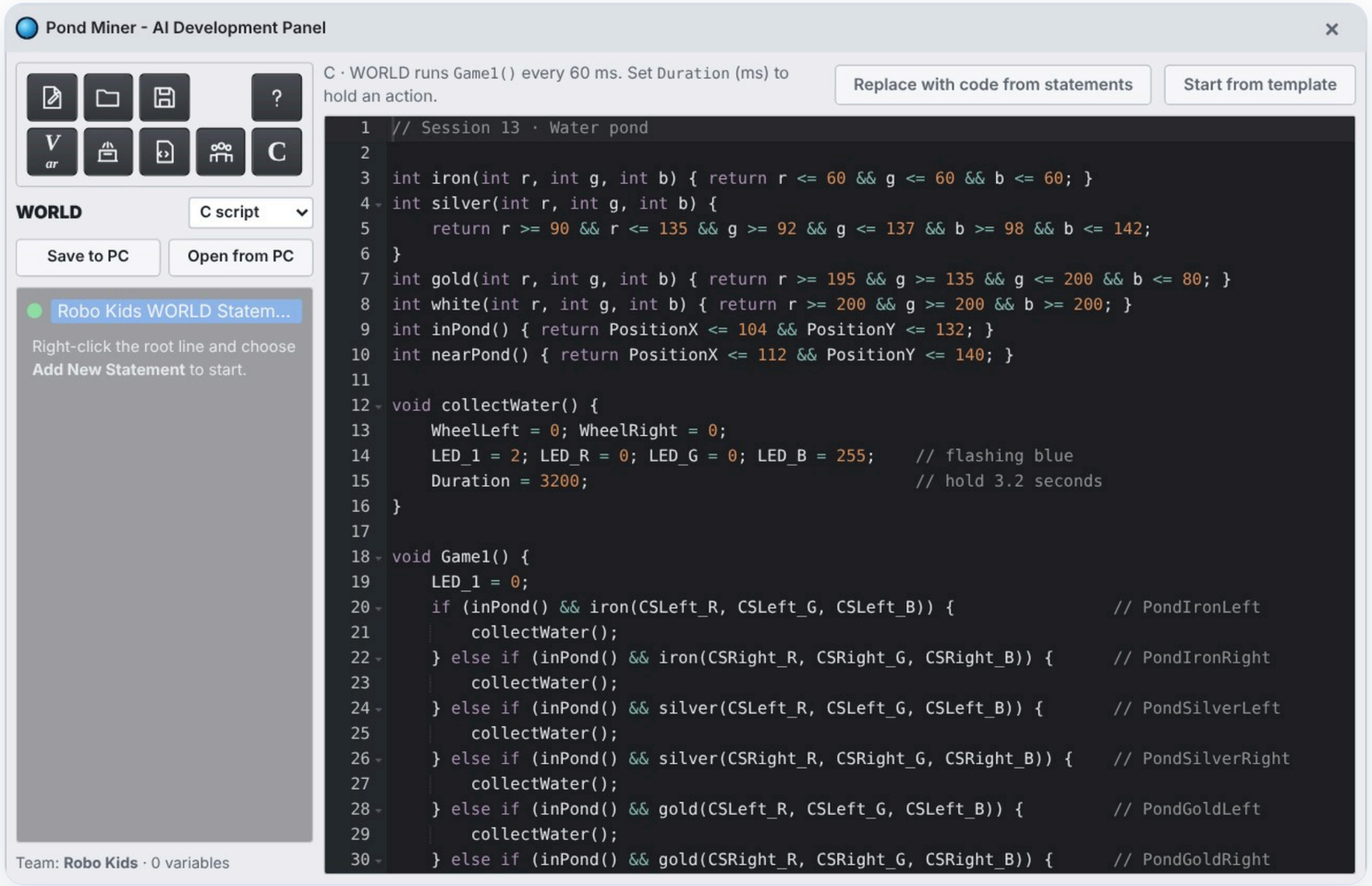
StayIn: near the pond && left sensor on white floor → turn back, 700 ms.

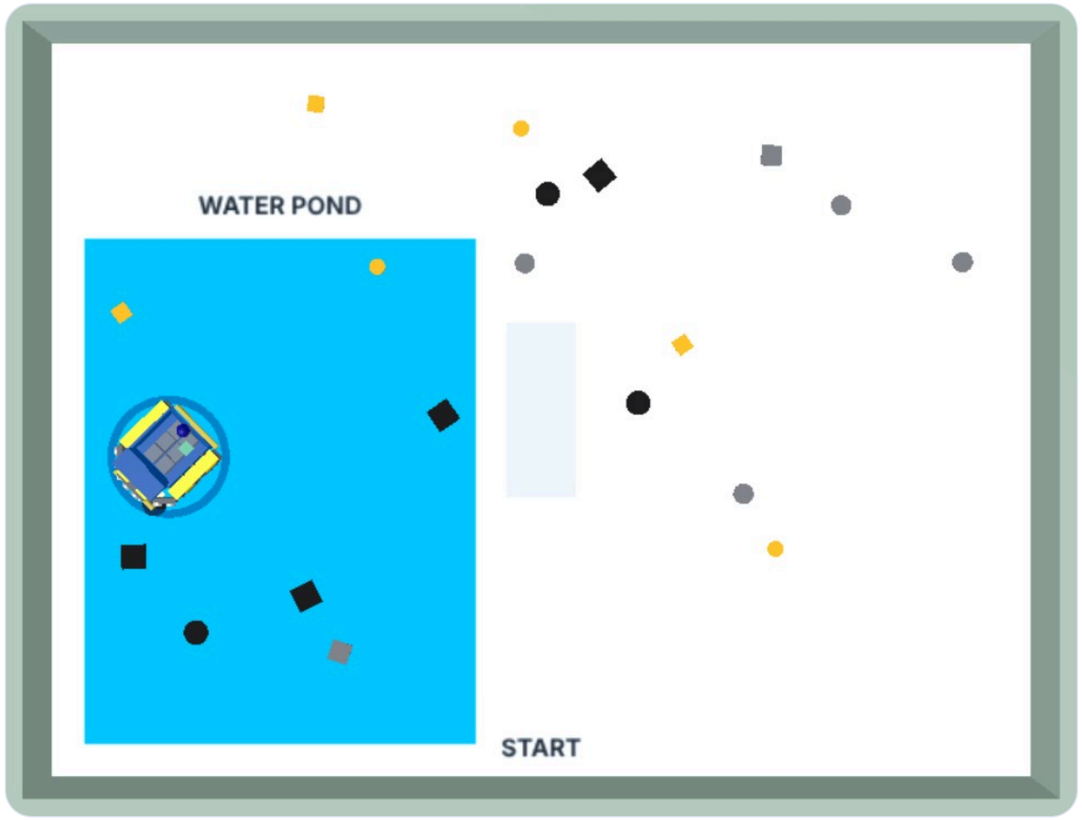
Ahead: wall in front → turn. **Forward**: the else.

STEP 6 · BUILD AND LOAD

The whole program, built

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





Colour Sensors

Left R	26	(CSLeft_R)
G	24	(CSLeft_G)
B	27	(CSLeft_B)

left sees this colour

Right R	1	(CSRight_R)
G	170	(CSRight_G)
B	240	(CSRight_B)

right sees this colour

Robot

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

Load **0**/6 · Fe 0 Ag 0 Au 0

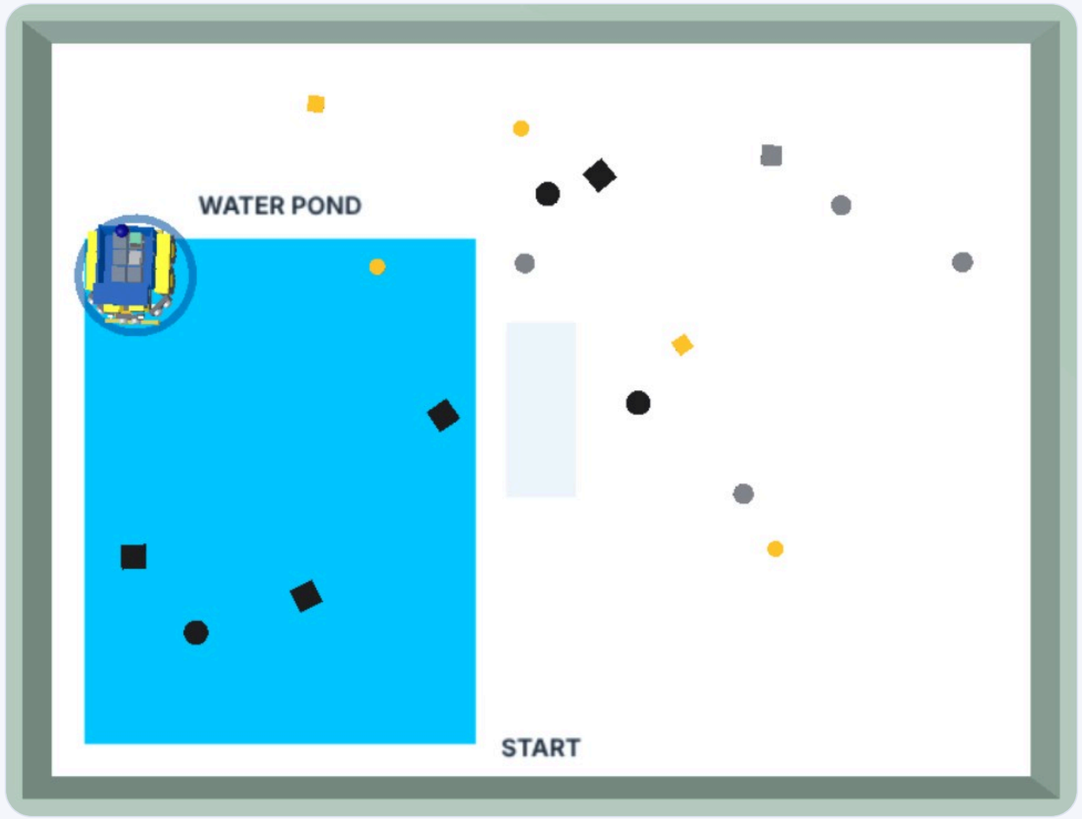
On **water pond · collecting 2.1 / 3 s**

STEP 7 · RUN · 7.8 S

First iron

Left **27, 21, 28** = iron. X 29, Y 79 = in the pond.

PondIronLeft: blue LED, collecting.



Colour Sensors

Left R	0	(CSLeft_R)
G	169	(CSLeft_G)
B	241	(CSLeft_B)

left sees this colour

Right R	221	(CSRight_R)
G	169	(CSRight_G)
B	34	(CSRight_B)

right sees this colour

Robot

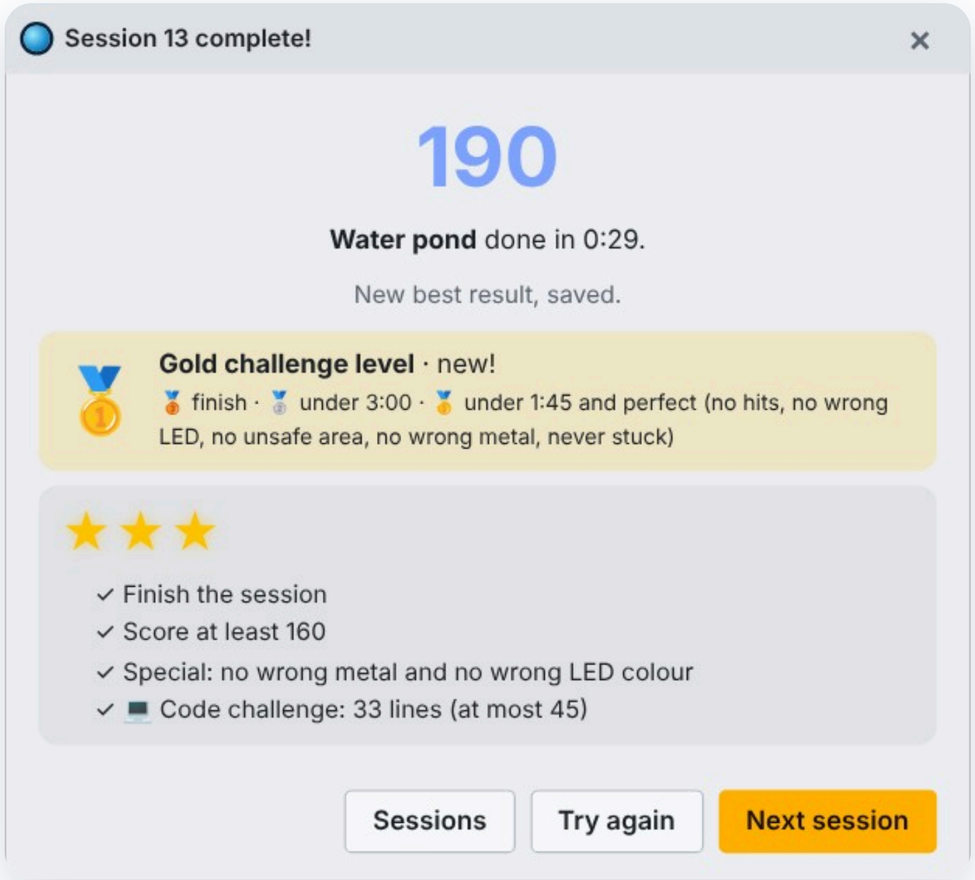
Wheels	L **0.0** · R **0.0** · LED **2**
Load	**2**/6 · Fe 1 Ag 1 Au 0
On	**water pond · collecting 1.8 / 3 s**

STEP 8 · RUN · 27.9 S

The third metal

Iron **+20** and silver **+30** already on board.

Right **222, 166, 36** = gold: **PondGoldRight**.



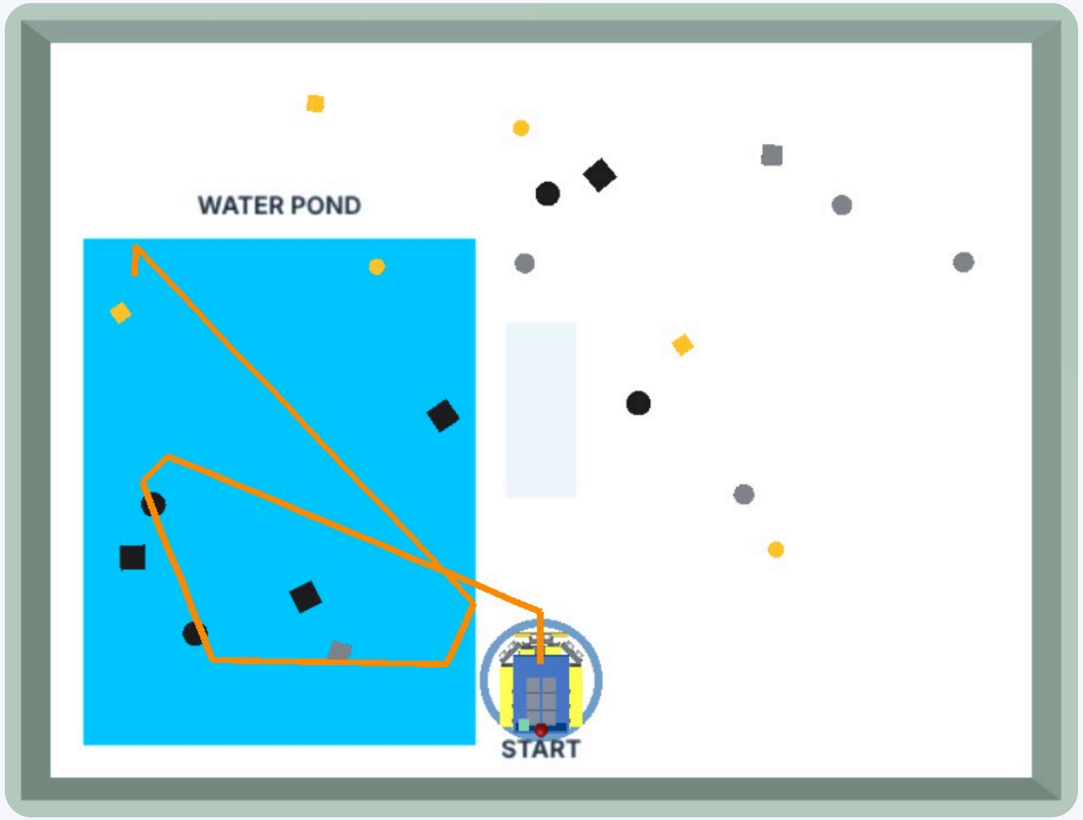
RESULT · TESTED

190 points in 0:29 · Gold

Score · iron 20 + silver 30 + gold 40 + bonus 100.

Medals · Silver: under 3:00 · Gold: under 1:45 and perfect.

Code challenge · at most 45 lines. We used 33.



THE PATH · RECORDED

Into the pond, and kept there

Every bend at the edge of the water is **StayIn** at work.

Position conditions keep the robot **where the points are**.

DID NOT WORK?

Check first, then tune carefully

What you see	What to do
Red LED in the pond, not counted	Call **collectWater()**, not collect()
Drives out of the pond	Check nearPond(): PositionX ≤ 112, PositionY ≤ 140
Drives over pond metals	Check the metal functions and inPond(): X ≤ 104, Y ≤ 132
Old program still on the robot	**Build** again, then **Load** on BLUE

NEXT TIME

Presentation 14 · Green deposit

Bring the metals home: a full load, then the **green deposit** area.