PRESENTATION 19 · LEARNING WITH PYTHON

Mini mission

Water, land and deposit in one program: red, blue and green LEDs, and three minutes on the clock.

TheBotCup 2026 · AiBotCup Simulator



RECAP · PRESENTATIONS 13 AND 14

Three action functions, three LEDs

collect()

red LED · on the field

collectWater()

blue LED · in the pond, double points

deposit()

green LED · both sensors on green

Plus **LoadedObjects** to know when we are full. Today: all of it at once.

PART 1

The mission: Session 19

Four objects, three minutes

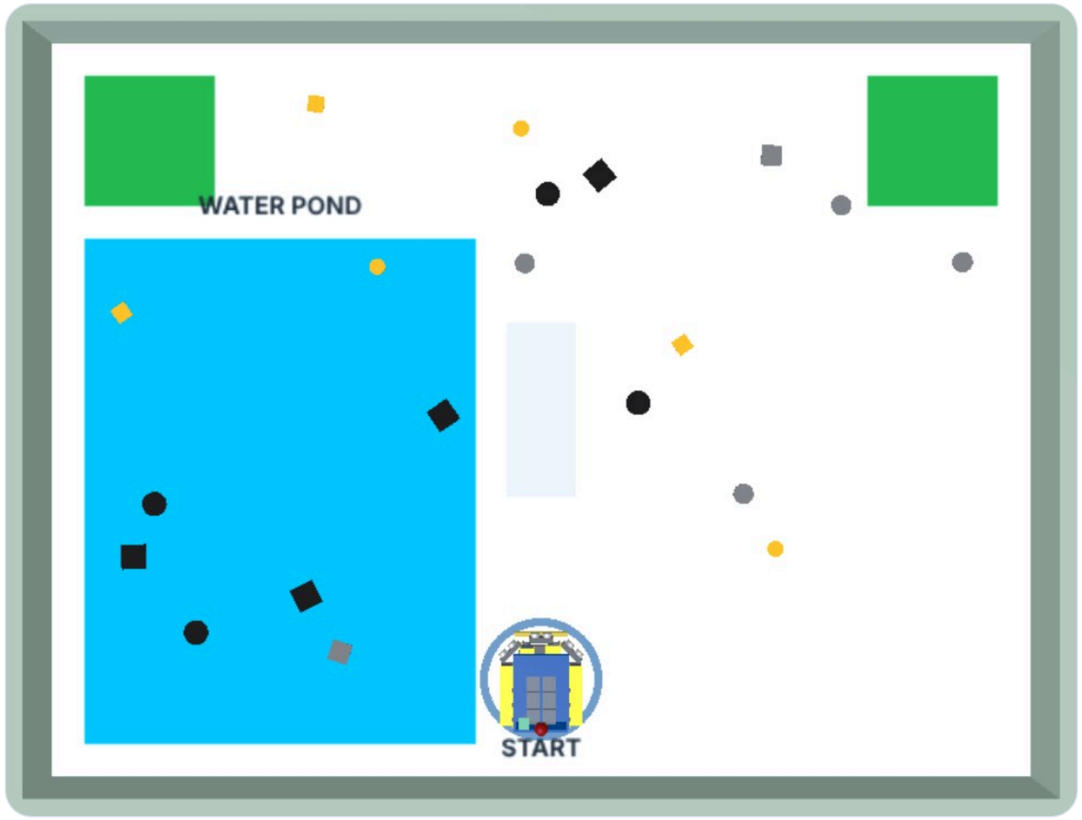
SESSION 19 · MINI MISSION

Mine and deposit 4 objects

Red on the field, **blue** in water, **green** to deposit.

All metal points count. 4 deposited: **+150**. Time limit **3:00**. Code challenge: at most **45** lines.





THE FIELD

Pond, land, two green corners

The **pond** on the left: double points.

Metals on **land** on the right.

Green **deposit** squares in both top corners.

PART 2

Two smart choices

The order, and what to leave out

CHOICE 1 · THE ORDER

Water collects above land collects

In the pond

Iron makes **PondIronLeft** and **IronLeft** both true. The one that comes first wins: **blue LED**.

On land

The pond branches need inPond(): they are false. **IronLeft** wins: red LED.

The first true branch wins, so the **more specific** branch goes first.

CHOICE 2 · 45 LINES AT MOST

Leave silver out

Everything	Our choice
Tools, start, Deposit, drivers = 27	Tools, start, Deposit, drivers = 27
Pond: 3 metals × 2 sensors × 2 lines = 12	Pond: 2 metals × 2 × 2 lines = 8
Land: 3 metals × 2 × 2 lines = 12	Land: 2 metals × 2 × 2 lines = 8
+ silver() = 52, too many	**43, fits**

Iron is everywhere, gold is worth the most: **silver** is the one to drop.

PART 3

Step by step in Python

Eleven branches

STEP 1 · THE WHOLE PROGRAM

Deposit, pond, land, drivers

Branch	Condition	Also	Action
1 Deposit	onGreen()	LoadedObjects == 4	deposit()
2 PondIronLeft	left iron · in the pond	LoadedObjects < 4	collectWater()
3 PondIronRight	right iron · in the pond	LoadedObjects < 4	collectWater()
4 PondGoldLeft	left gold · in the pond	LoadedObjects < 4	collectWater()
5 PondGoldRight	right gold · in the pond	LoadedObjects < 4	collectWater()
6 IronLeft	left iron	LoadedObjects < 4	collect()
7 IronRight	right iron	LoadedObjects < 4	collect()
8 GoldLeft	left gold	LoadedObjects < 4	collect()
9 GoldRight	right gold	LoadedObjects < 4	collect()
10 Ahead	US_Front ≤ 20	—	−2 / 2 · Duration 700
11 Forward	the else	—	4.5 / 4.5

STEP 2 · LINES 3–23

All our tools in one program

```
3  def iron(r, g, b): return r <= 60 and g <= 60 and b <= 60
4  def gold(r, g, b): return r >= 195 and 135 <= g <= 200 and b <= 80
5  def green(r, g, b): return r <= 75 and 130 <= g <= 200 and 40 <= b <= 110
6  def onGreen():
7      return green(CSLeft_R, CSLeft_G, CSLeft_B) and green(CSRight_R, CSRight_G, CSRight_B)
8  def inPond(): return PositionX <= 104 and PositionY <= 132
9
10 def collect():
11     WheelLeft = 0; WheelRight = 0
12     LED_1 = 2; LED_R = 255; LED_G = 0; LED_B = 0     # flashing red
13     Duration = 3200
14
15 def collectWater():
16     WheelLeft = 0; WheelRight = 0
17     LED_1 = 2; LED_R = 0; LED_G = 0; LED_B = 255     # flashing blue
18     Duration = 3200
19
20 def deposit():
21     WheelLeft = 0; WheelRight = 0
22     LED_1 = 1; LED_R = 0; LED_G = 255; LED_B = 0     # steady green
23     Duration = 3200
```

STEP 3 · LINES 25–36

Deposit, then four pond collects

```
25  def game1():
26      LED_1 = 0
27      if LoadedObjects == 4 and onGreen():
28          deposit()                                   # Deposit
29      elif LoadedObjects < 4 and inPond() and iron(CSLeft_R, CSLeft_G, CSLeft_B):
30          collectWater()                              # PondIronLeft
31      elif LoadedObjects < 4 and inPond() and iron(CSRight_R, CSRight_G, CSRight_B):
32          collectWater()                              # PondIronRight
33      elif LoadedObjects < 4 and inPond() and gold(CSLeft_R, CSLeft_G, CSLeft_B):
34          collectWater()                              # PondGoldLeft
35      elif LoadedObjects < 4 and inPond() and gold(CSRight_R, CSRight_G, CSRight_B):
36          collectWater()                              # PondGoldRight
```

Deposit: LoadedObjects == 4 and onGreen().

Pond: **load < 4 and inPond() and** metal → collectWater().

STEP 4 · LINES 37–49

Land collects, then the drivers

```
37      elif LoadedObjects < 4 and iron(CSLeft_R, CSLeft_G, CSLeft_B):
38          collect()                                   # IronLeft
39      elif LoadedObjects < 4 and iron(CSRight_R, CSRight_G, CSRight_B):
40          collect()                                   # IronRight
41      elif LoadedObjects < 4 and gold(CSLeft_R, CSLeft_G, CSLeft_B):
42          collect()                                   # GoldLeft
43      elif LoadedObjects < 4 and gold(CSRight_R, CSRight_G, CSRight_B):
44          collect()                                   # GoldRight
45      elif US_Front <= 20:
46          WheelLeft = -2; WheelRight = 2              # Ahead
47          Duration = 700
48      else:
49          WheelLeft = 4.5; WheelRight = 4.5           # Forward
```

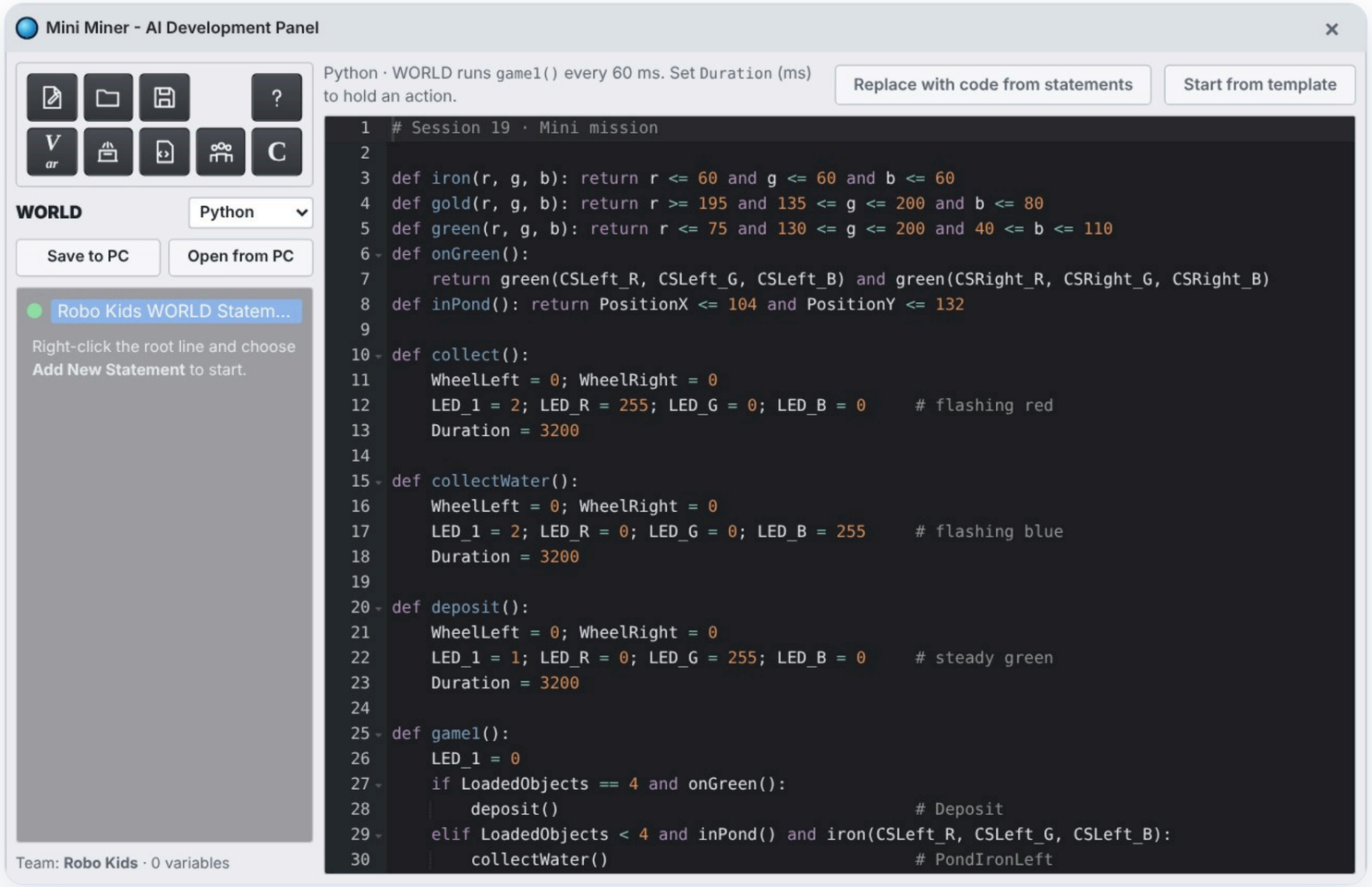
Land: **load < 4 and** metal → collect(), red LED.

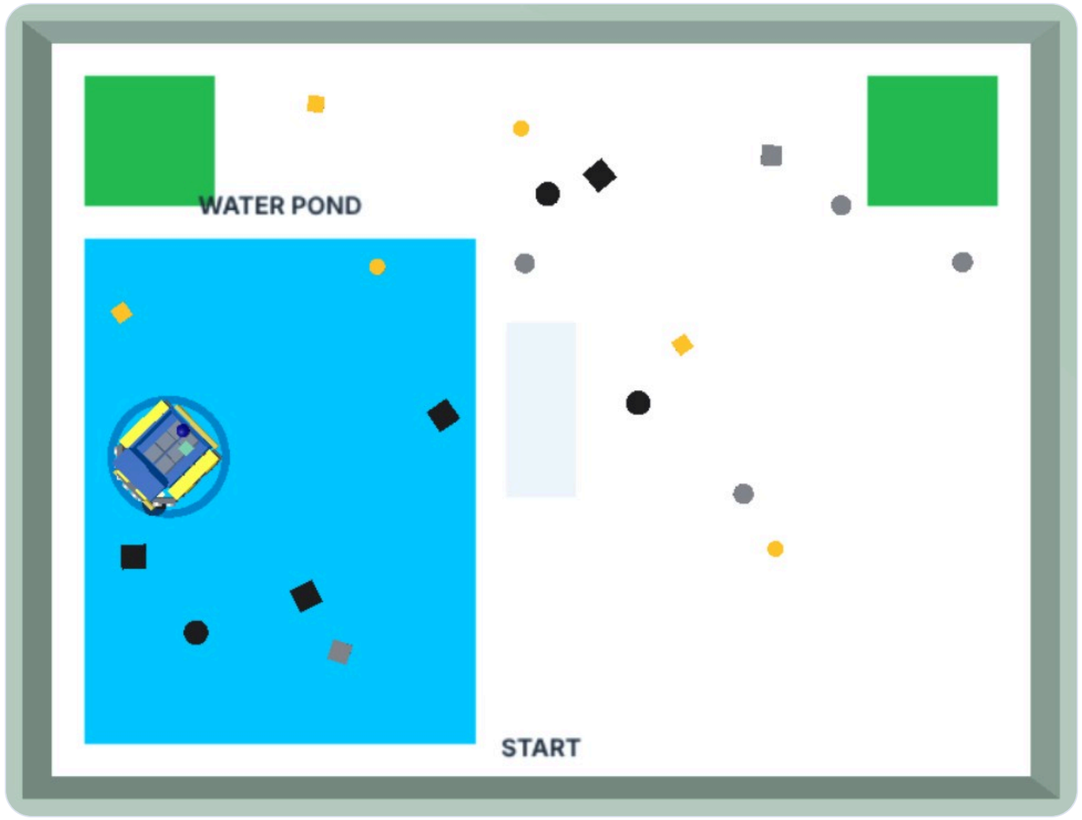
Ahead (700 ms) and the else, Forward, close the chain.

STEP 5 · BUILD AND LOAD

The whole program, built

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





Colour Sensors

Left R	27	(CSLeft_R)
G	21	(CSLeft_G)
B	23	(CSLeft_B)

left sees this colour

Right R	1	(CSRight_R)
G	172	(CSRight_G)
B	243	(CSRight_B)

right sees this colour

Robot

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

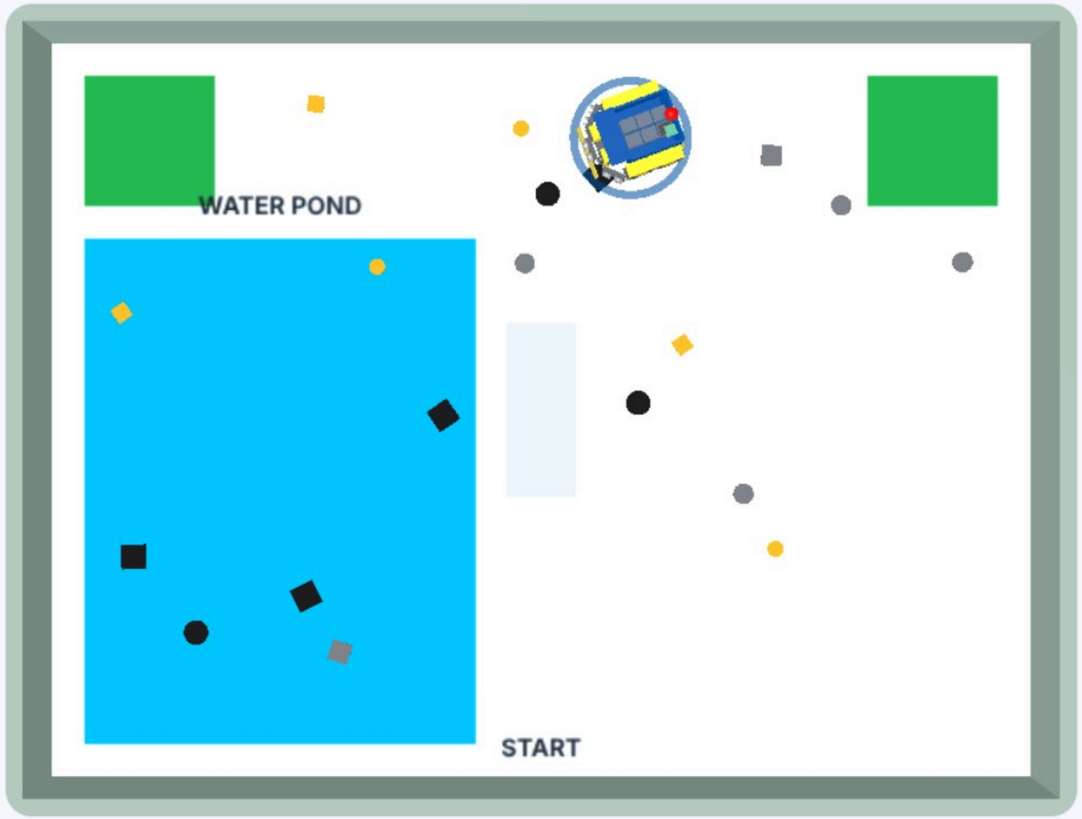
Load **0**/6 · Fe 0 Ag 0 Au 0

On **water pond · collecting 1.8 / 3 s**

STEP 6 · RUN · 7.7 S

Iron in the water

PondIronLeft and IronLeft are both true. **PondIronLeft** comes first: blue LED.



Colour Sensors

Left R	27	(CSLeft_R)
G	23	(CSLeft_G)
B	27	(CSLeft_B)

left sees this colour

Right R	227	(CSRight_R)
G	233	(CSRight_G)
B	238	(CSRight_B)

right sees this colour

Robot

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

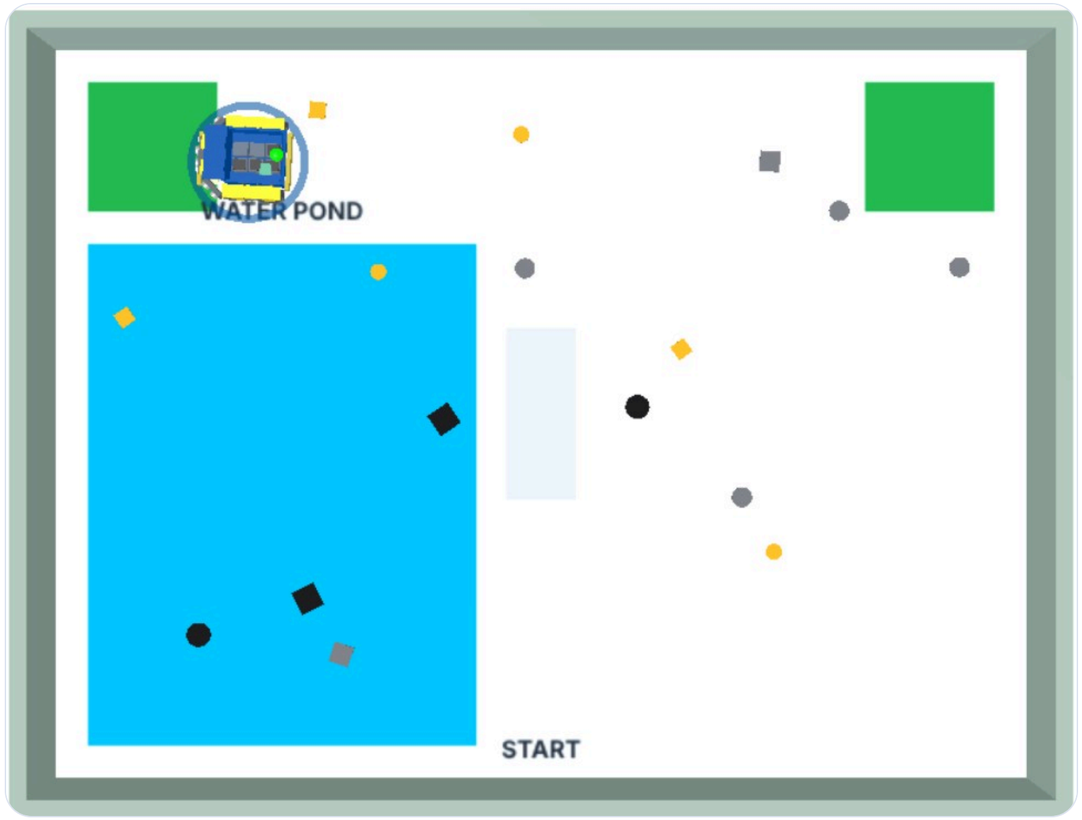
Load **1**/6 · Fe 1 Ag 0 Au 0

On **collecting 1.7 / 3 s**

STEP 7 · RUN · 29.1 S

Iron on land

Outside the pond box, so inPond() is false. **IronLeft**: red LED.



Colour Sensors

Left R	28	(CSLeft_R)
G	162	(CSLeft_G)
B	68	(CSLeft_B)

left sees this colour

Right R	29	(CSRight_R)
G	161	(CSRight_G)
B	69	(CSRight_B)

right sees this colour

Robot

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

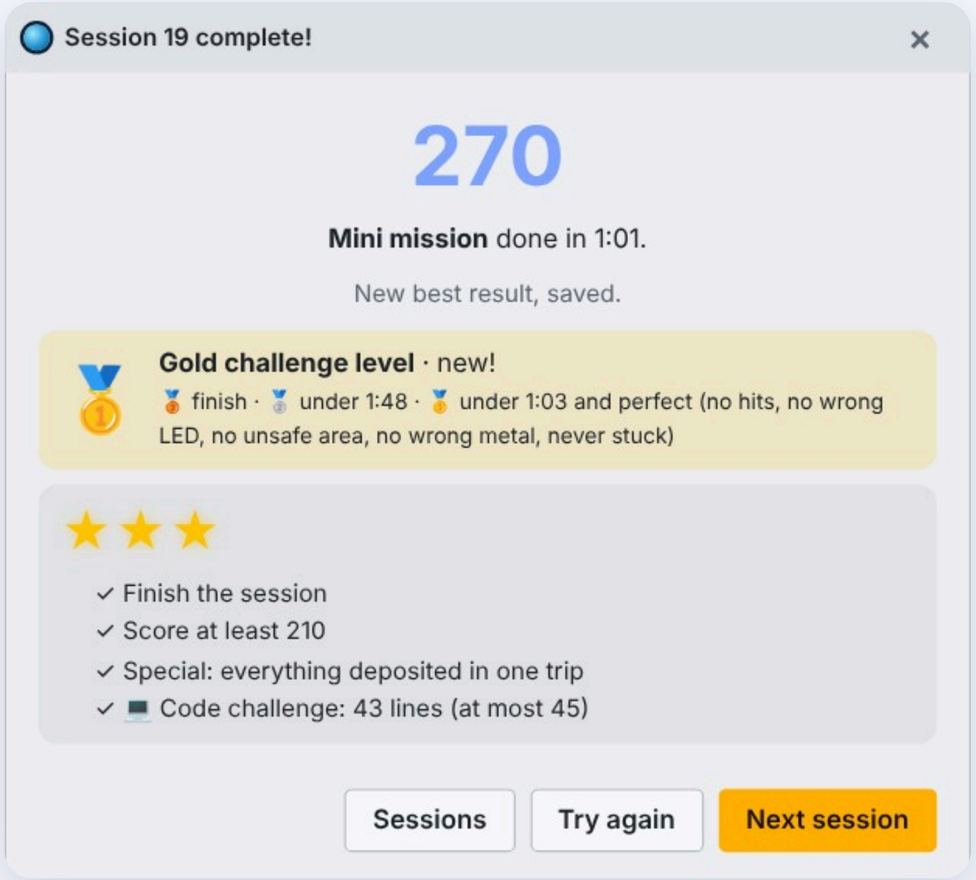
Load **4**/6 · Fe 4 Ag 0 Au 0

On **deposit area · depositing 1.6 / 3 s**

STEP 8 · RUN · 58.2 S

Four on board, on green

Load **4** (four irons), both sensors green: **Deposit**.



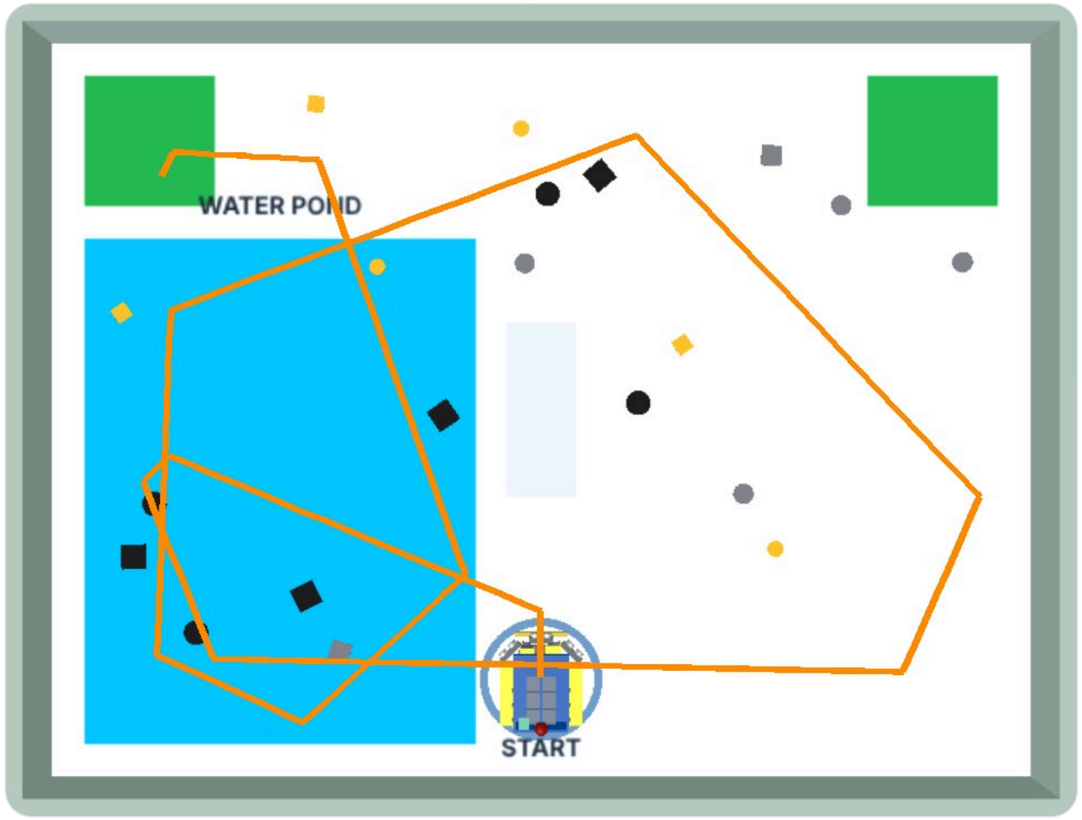
RESULT · TESTED

270 points in 1:01 · Gold

Score · metals 60 + deposited 60 + bonus 150.

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

Code challenge · at most 45 lines. We used 43.



THE PATH · RECORDED

Water, land, water, home

The right LED in every place, decided by **order** and **position**.

Specific branches first, general ones after: **a rule for every program**.

DID NOT WORK?

Check first, then tune carefully

What you see	What to do
Red LED in the pond, not counted	Put the pond branches **before** the land ones
Blue LED on land	Every pond branch needs **inPond() and**
Too slow for gold	Try LoadedObjects 3 and compare the score
Old program still on the robot	**Build** again, then **Load** on BLUE

NEXT TIME

Presentation 20 · Final challenge

The full Juniors challenge, with **warning bands** to avoid.