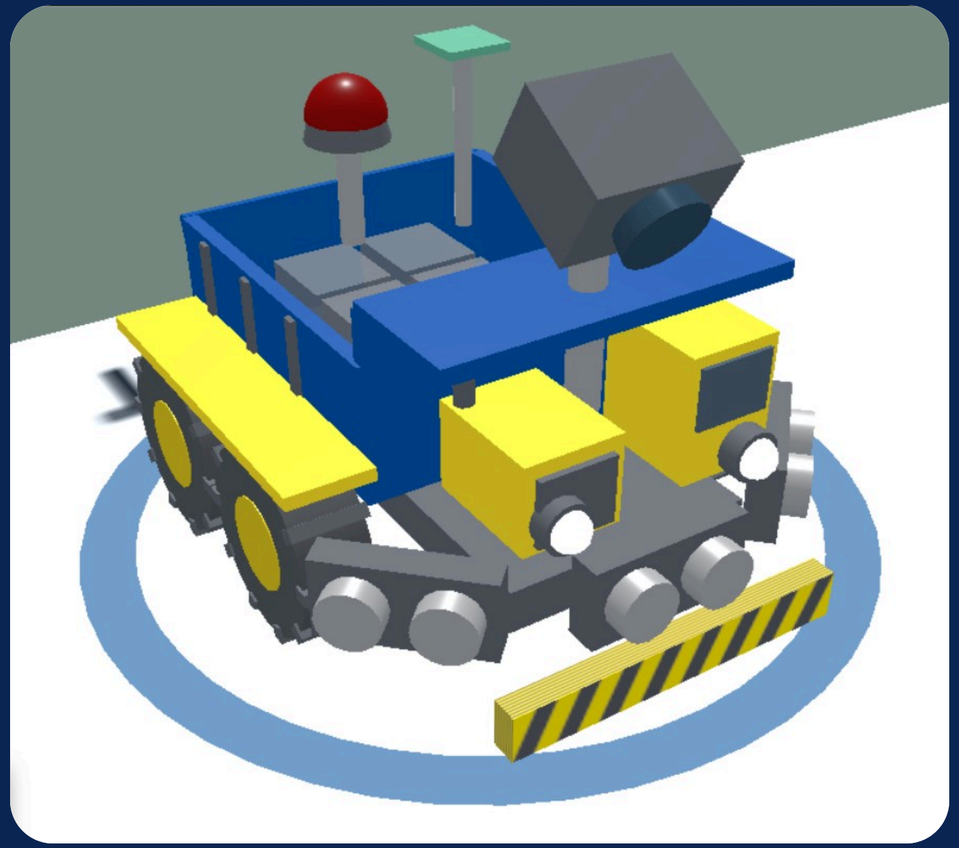
UNIVERSITY · PRESENTATION 9 · LEARNING WITH PYTHON

Final challenge

The pond, unsafe areas, Oil and Rocks, two deposit areas: mine six and bring them home in one trip.

TheBotCup 2026 · AiBotCup Simulator · Advanced



RECAP

Everything we built

Sessions 12–13

Position

atan2d and angleDiff aim at any point.

Sessions 14–16

Camera objects

Colour tests, shapes, line up by column.

Sessions 17–18

Camera places

Water under us; red on each side.

Session 19 · last time

Two modes

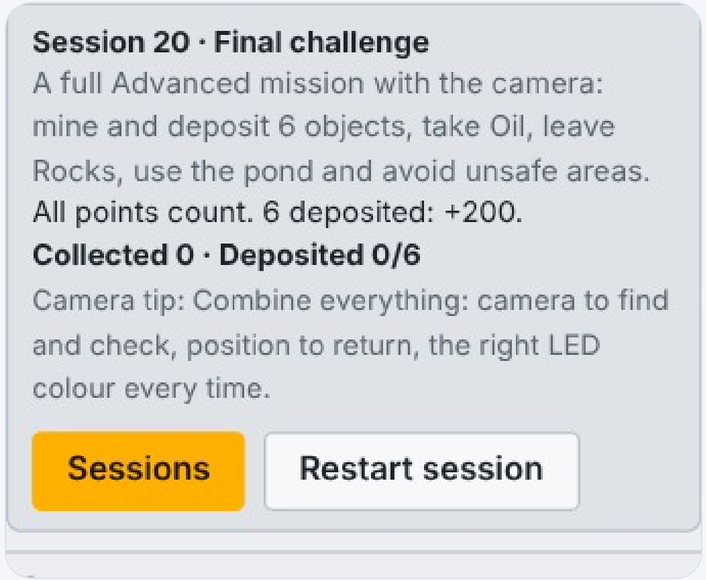
Search by camera, return by position.

Today: **one program**, six rules in a clear order, and a strategy that keeps it under 45 lines.

PART 1

The mission: Session 20

Mine and deposit 6 objects



SESSION 20 · FINAL CHALLENGE

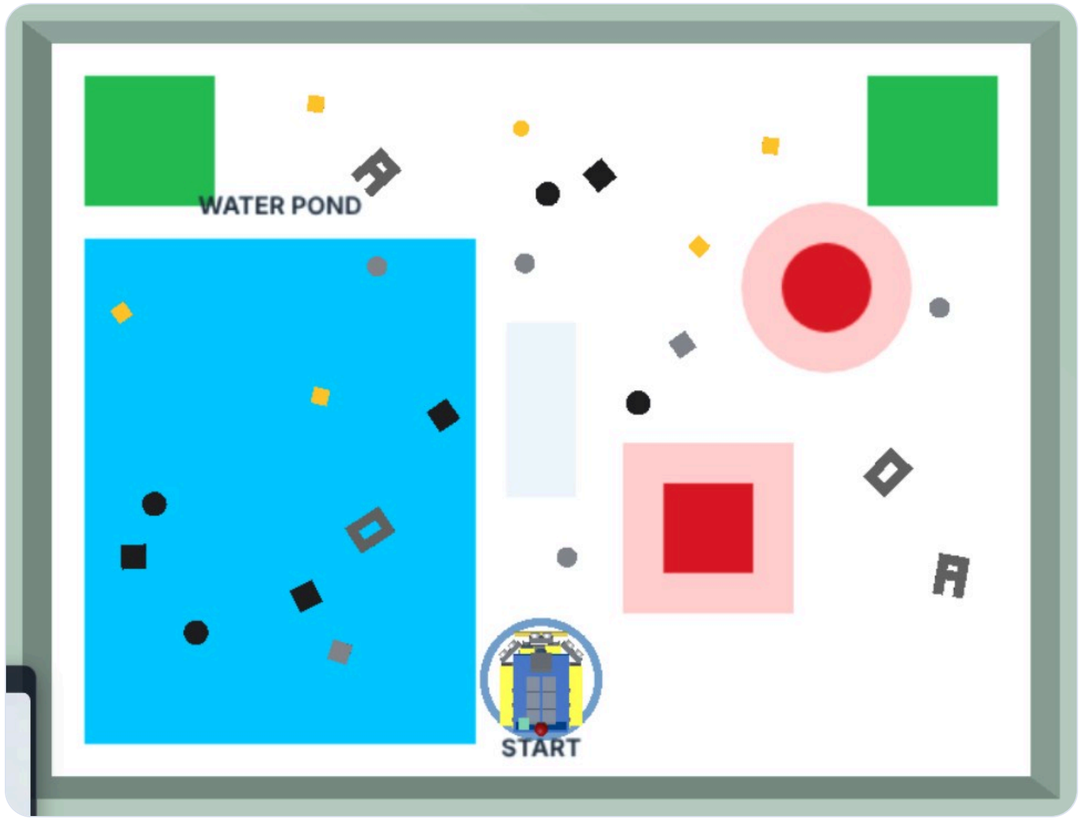
The full Advanced mission

6 deposited: +200. All metal points count, pond metals double.

Leave **Rocks (R)**: trying costs −10. Avoid the unsafe areas.

Right LED every time: **blue** in water, **red** on land, **green** to deposit.

Time **5:00** · gold: under 1:45 and perfect · star: all six **in one trip**.



THE FIELD

Every element at once

Pond x 8–104, y 8–132 · **2 deposits** in the top corners, centres (24, 156) and (216, 156).

2 unsafe areas: a square at (150–172, 50–72) and a circle at (190, 120).

Letters: 2 O (Oil), 2 R (Rocks). A **block** in the middle.

START (120, 24), facing north.

PART 2

Strategy before code

Decide what to mine, then how

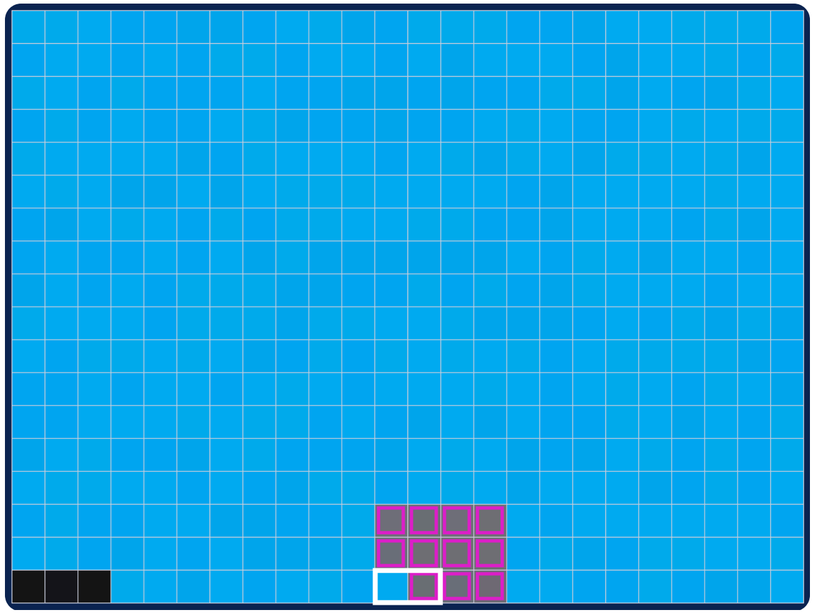
WHAT IS ON THE FIELD

Choose the easy six

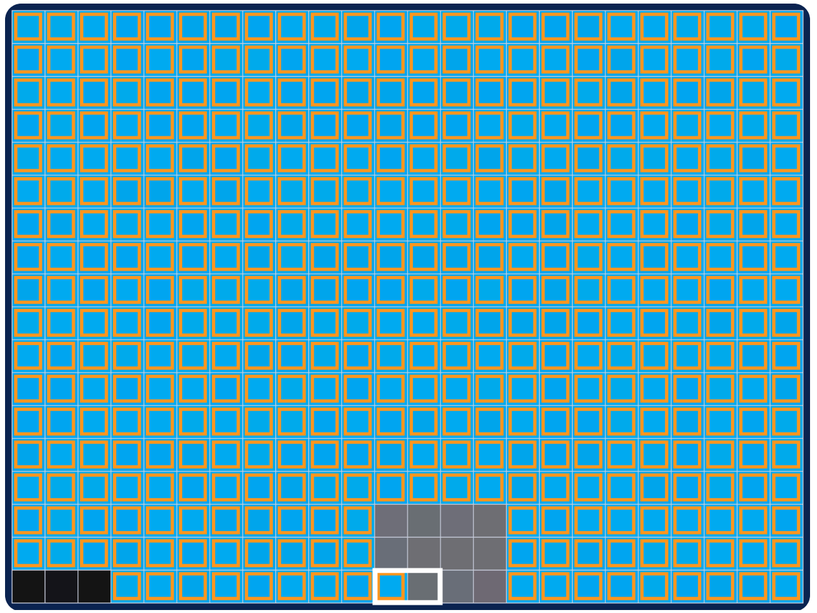
Object	On the field	Camera colour	Our choice
Gold	6 (2 in the pond)	220, 169, 34	**collect**
Silver	6 (2 in the pond)	110, 113, 118	**collect**, but only B < 124: the block is 126
Iron	8 (5 in the pond)	25, 23, 25	leave: looks like an R letter
O (Oil)	2 (1 in the pond)	black ring	leave: not needed
R (Rocks)	2	black, ink in the centre	never: −10

ONE REAL IMAGE · TWO MASKS

Silver in the water: which LED?



Metal mask: silver in the box, **108, 114, 115**. The iron is ignored.



Pond mask: 58 of 72 bottom pixels are water → **blue**.

One scan answers three questions at once: **nearest metal**, **red left / right**, **water under us**.

TWO DEPOSIT AREAS

Go to the nearer one

hx = 24 if PositionX < 120, else 216
Left half of the field → left deposit, right half → right deposit.

goal = atan2d(156 − PositionY, hx − PositionX) − 90
The same geometry as in Session 19.

When	Position	Compass	Deposit	goal	off by
6th catch, 55.1 s	(75, 163)	85°	left (24, 156)	98°	13°

Full at the top of the field, close to the left deposit: a short, straight trip home.

TUNING · MEASURED WHILE DEVELOPING

Faster exploring, better medal

Explore wheels	Approach speed	Time	Medal
3 / 3.3	3	1:57	silver
4 / 4.4	4	2:04	silver
5 / 4.5	**4**	**1:03**	**gold**

PART 3

Step by step in Python

Six rules in one function

THE PLAN

Safety first, then the job

1. Wall or block ahead (US_Front < 15) → **turn** for 0.6 s.
2. More than 3 warning pixels → **steer away** from the redder side.
3. Six on board → **home** to the nearer deposit; green in the box → deposit.
4. Gold or silver in the box → **collect**: blue if water under us, else red.
5. Gold or silver in view → **steer to the nearest**; nothing → **explore**.

Safety rules sit above the job rules, so they win even when the robot is carrying six objects home.

STEP 1 · LINES 3–8

Two colour tests

```
3  def metal(p):                        # gold or silver (iron looks like Rocks)
4      if p[0] > 195 and 135 < p[1] < 200 and p[2] < 80: return True      # gold
5      return 95 < p[0] < 130 and 95 < p[1] < 130 and 100 < p[2] < 124   # silver, not the block
6
7  def green(p):                        # deposit area
8      return p[0] < 75 and 130 < p[1] < 200 and 40 < p[2] < 110
```

metal: gold, or silver with B < 124 (not the block).

green: the deposit colour.

STEP 2 · LINES 10–15

Two actions: hold() and drive()

```
10  def hold(r, g, b, led):             # stop, show the LED, hold 3.2 s
11      WheelLeft = 0; WheelRight = 0; Duration = 3200
12      LED_1 = led; LED_R = r; LED_G = g; LED_B = b
13
14  def drive(turn):                    # turn > 0: to the left
15      WheelLeft = 4 - turn; WheelRight = 4 + turn
```

hold(r, g, b, led): collect or deposit, 3.2 s.

drive(turn): speed 4, turn > 0 to the left.

STEP 3 · LINES 17–30

One scan, three answers

```
17  def game1():
18      LED_1 = 0
19      mid = CAM_W // 2
20      row = -1; col = 0; redL = 0; redR = 0; water = 0
21      for y in range(CAM_H):
22          for x in range(CAM_W):
23              p = Camera[y][x]
24              if metal(p) and y > row:
25                  row = y; col = x
26              elif p[0] > 230 and 145 < p[1] < 205 and 145 < p[2] < 205:     # warning band
27                  if x < mid: redL += 1
28                  else: redR += 1
29              elif p[0] < 60 and p[1] > 130 and p[2] > 200 and y >= CAM_H - 3:
30                  water += 1                                                  # pond under us
```

row, col: nearest metal.

redL, redR: warning band per half.

water: pond in the 3 bottom rows.

STEP 4 · LINES 31–41

Safety, then home

```
31      base = Camera[CAM_H - 1]                                    # bottom row
32      if US_Front < 15:                                           # wall or block ahead
33          WheelLeft = -2; WheelRight = 2; Duration = 600
34      elif redL + redR > 3:                                       # unsafe area ahead
35          drive(-2.5 if redL > redR else 2.5)
36      elif LoadedObjects >= 6:                                    # full: deposit all 6
37          if green(base[mid - 1]) and green(base[mid]):
38              hold(0, 255, 0, 1)                                  # green, steady
39          else:
40              hx = 24 if PositionX < 120 else 216                 # nearer deposit area
41              drive(angleDiff(atan2d(156 - PositionY, hx - PositionX) - 90, Compass) * 0.05)
```

Band: **drive(−2.5)** or **drive(2.5)**.

Full: nearer deposit **hx**, then the heading.

STEP 5 · LINES 42–50

Mine: the right LED every time

```
42      elif metal(base[mid - 1]) or metal(base[mid]):
43          if water > 3:
44              hold(0, 0, 255, 2)                      # in the pond: BLUE
45          else:
46              hold(255, 0, 0, 2)                      # on land: red
47      elif row >= 0:
48          drive(-(col - (CAM_W - 1) / 2) * 0.25)      # steer to the metal
49      else:
50          WheelLeft = 5; WheelRight = 4.5             # explore
```

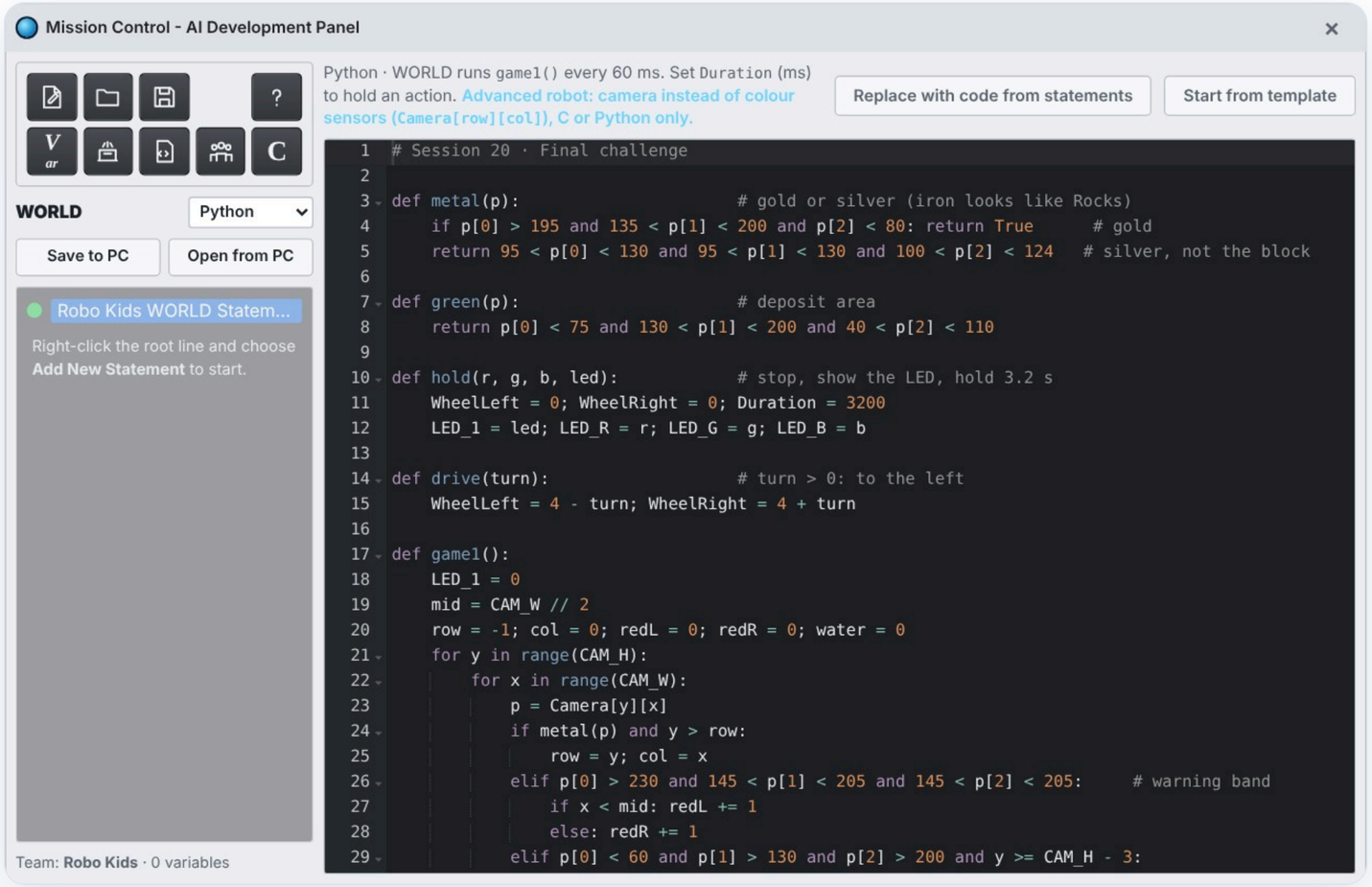
water > 3 → blue **0, 0, 255**, else red **255, 0, 0**.

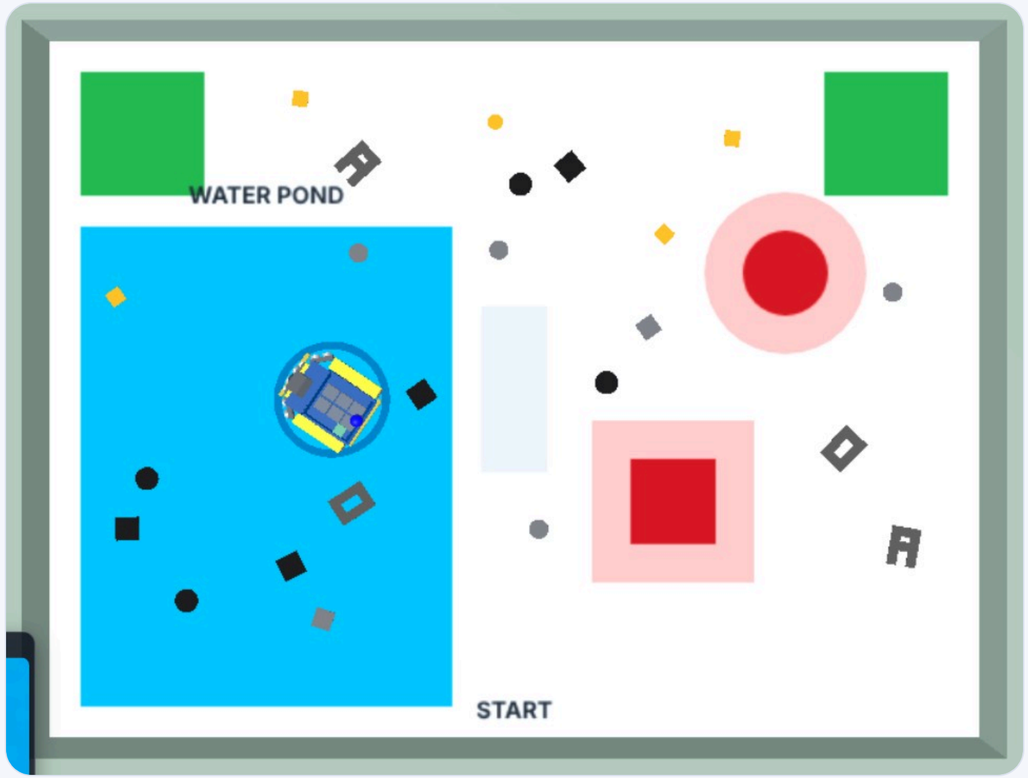
Explore: **5 / 4.5**, the tuned values.

BUILD AND LOAD

The whole program, built

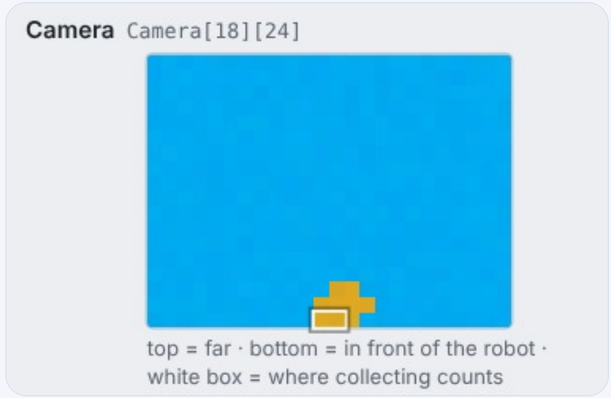
1. WORLD menu: **Python**.
2. Type the program, name it **Mission Control**.
3. Press **Build**, then **OK**.
4. Press **Load** on the BLUE side.





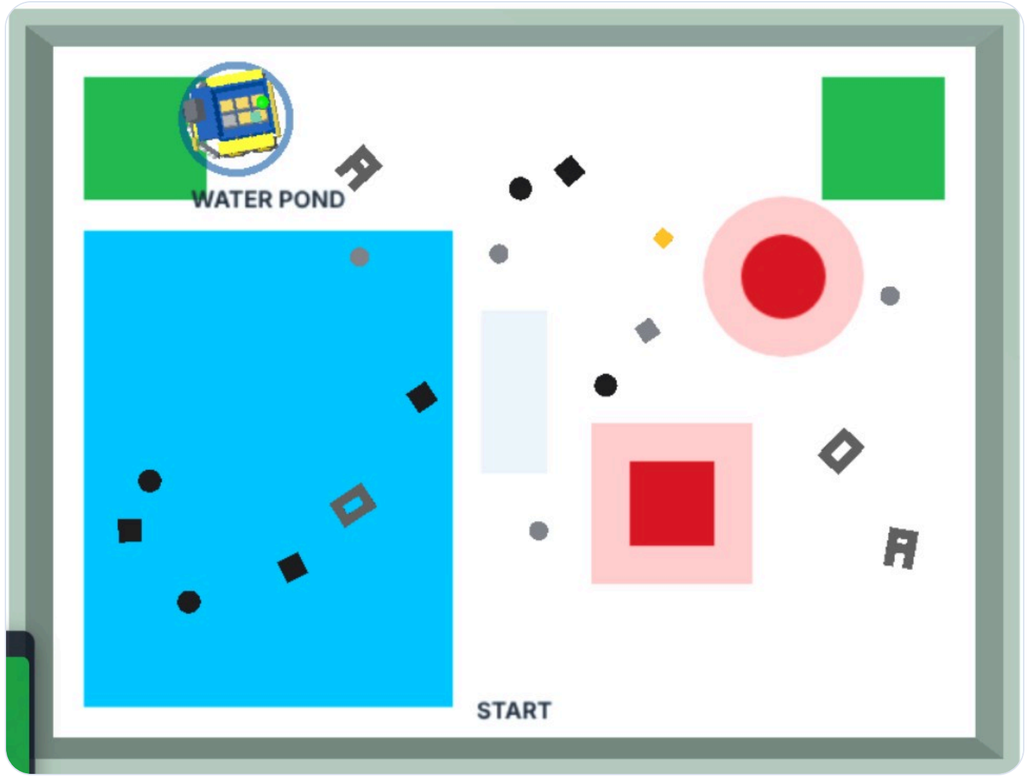
RUN · 6.0 S

First catch: gold in the pond



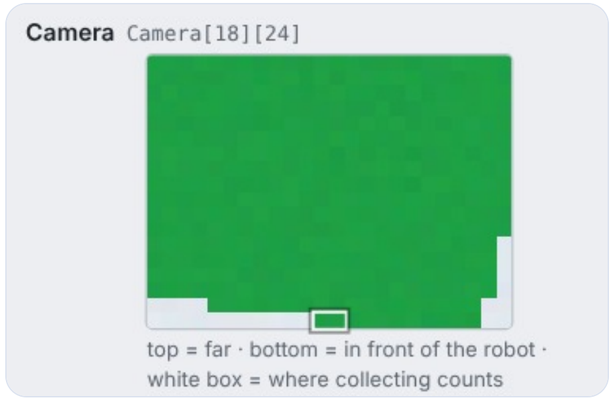
At (**73, 87**): gold **220, 169, 37**, water **63** of 72.

Blue LED · pond gold is worth **40**.



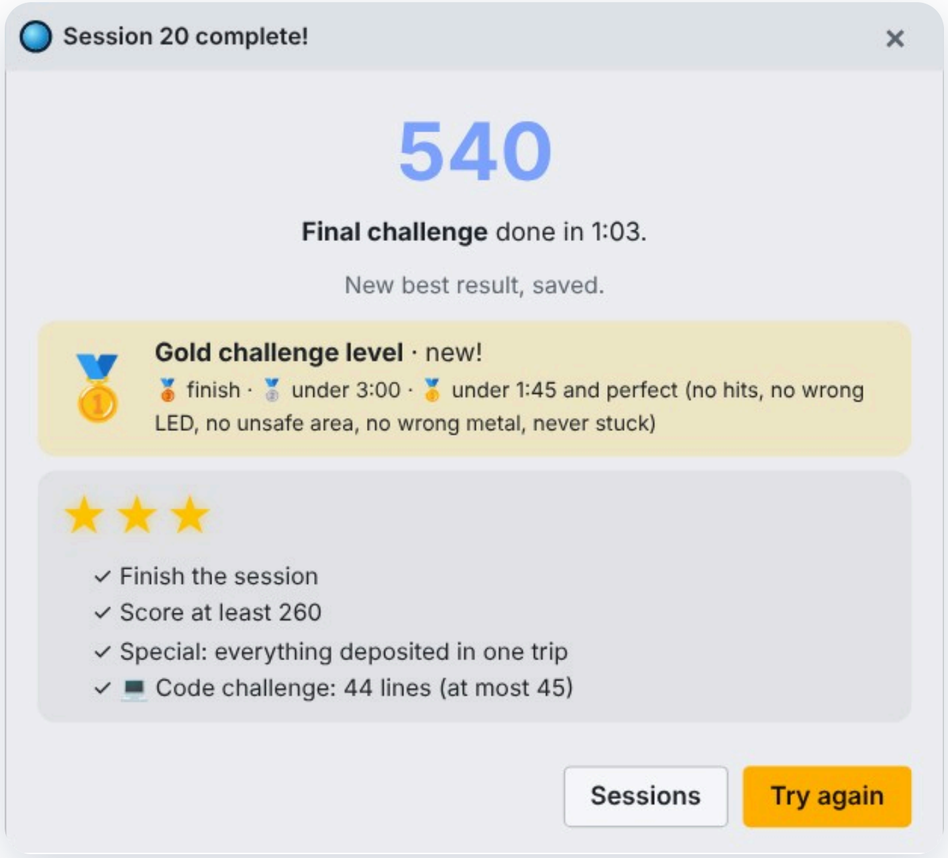
RUN · 59.2 S

Six on board: deposit



Three pond catches, three land golds, then home to the **left** deposit.

At (**48, 161**), box **31, 158, 71** → green, steady.



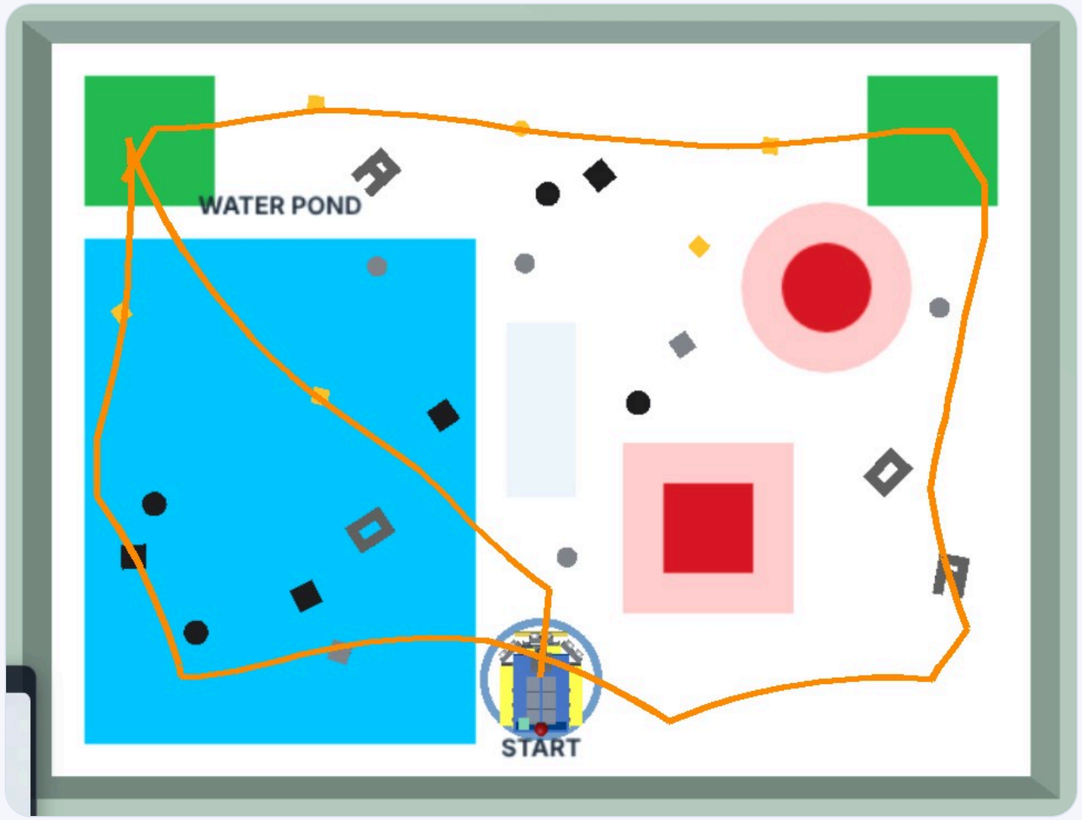
RESULT · TESTED

540 points in 1:03 · Gold

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

Stars · finish · score 260 · everything deposited in one trip.

Code challenge · at most 45 lines. We used 44.



THE PATH · RECORDED

One lap, six catches, one trip

Pond first, then a lap of the field, round the dangers.

It never touched an R, an unsafe area or the block.

A clear **order of rules** is a strategy you can read.

DID NOT WORK?

Check first, then tune carefully

What you see	What to do
Pushes against the block or a wall	Silver test: **B < 124** keeps the grey block out
Red LED in the water	water counts the 3 bottom rows; collect with **blue** when water > 3
Touches an unsafe area	The band rule must come **before** home and mining
Takes Rocks, −10	Do not count black as a metal
Slow, silver medal	Tune one number at a time: explore **5 / 4.5**

WELL DONE

University course complete

Now beat our tested run: **1:03**, **540 points**. Can you take Oil safely too?