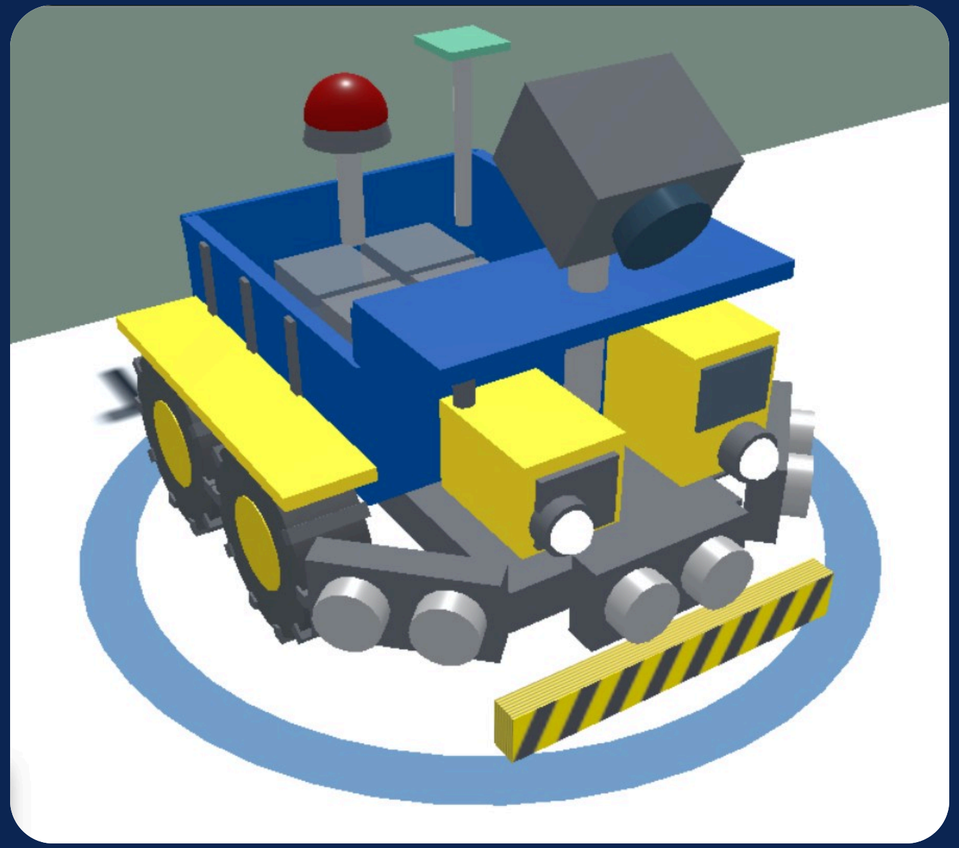
UNIVERSITY · PRESENTATION 8 · LEARNING WITH C

Deposit by position

Search with the camera, return with geometry: collect 3 objects, then drive home to the deposit area.

TheBotCup 2026 · AiBotCup Simulator · Advanced



RECAP

Two tools, one robot

Session 17

Pond by camera

A colour mask tells where we are.

Session 18 · last time

Avoid by camera

Count red per half; aim at FINISH with atan2d.

Session 19 · today

Deposit by position

Camera to search, position to return.

New today: a **mode**. The same **Game1()** behaves differently before and after the robot is loaded.

PART 1

The mission: Session 19

Collect 3, then deposit them at (216, 156)

Session 19 · Deposit by position

Collect 3 objects with the camera, then use PositionX, PositionY and Compass to drive to the deposit area at (216, 156) and deposit (green LED).

All metal points count. 3 deposited: +150.

Collected 0 · Deposited 0/3

Camera tip: Search with the camera, return with position: heading = atan2d(156 − PositionY, 216 − PositionX) − 90.

Sessions | Restart session

SESSION 19 · DEPOSIT BY POSITION

Bring three home

Collect **3 objects** (red LED), any metal.

Drive to the deposit area at **(216, 156)** and deposit: **green** LED, steady (LED_1 = 1).

3 deposited: **+150** · time limit **6:00**.

The star: deposit all three **in one trip**.



THE FIELD

Metals everywhere, home in the corner

18 metals: 6 iron, 6 silver, 6 gold.

Deposit area: x 200–232, y 140–172, top right.

A **block** in the middle: (112–128, 70–110).

START (120, 24), facing north.

PART 2

Search, then return

Two modes, two kinds of navigation

ONE NUMBER DECIDES THE MODE

Search mode and home mode

	Search mode	Home mode
When	LoadedObjects < 3	LoadedObjects ≥ 3
Sensor	Camera: the nearest metal	PositionX, PositionY, Compass
Steering	By the column error	By angleDiff to (216, 156)
Action	Metal in the white box → collect, red	Green in the white box → deposit, green
Navigation	Local: what is in front of me	Global: where I am on the field

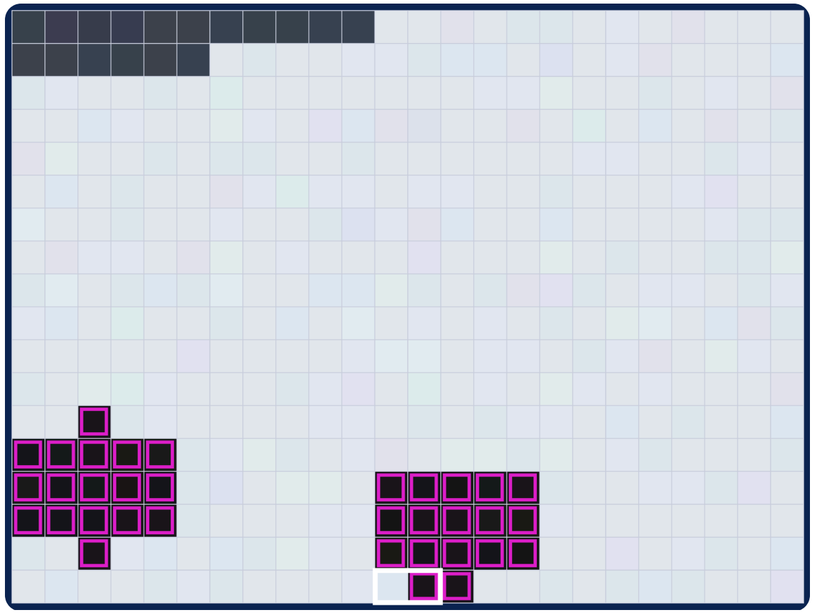
CHOOSE WHAT TO COLLECT

Silver looks like the block

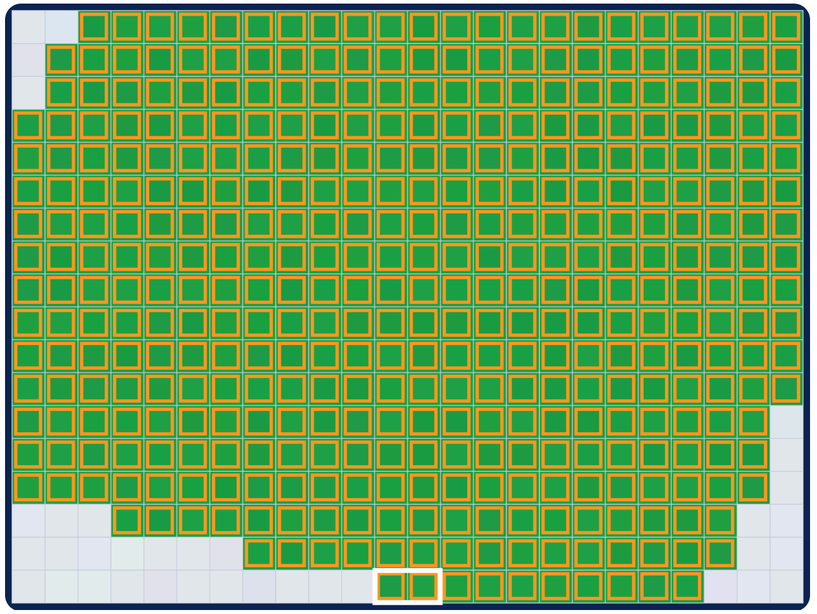
Surface	R, G, B (measured)	Our choice
Iron	23, 24, 23	**collect**: all three below 60
Gold	221, 167, 32	**collect**: R > 195, B < 80
Silver	110, 113, 118	leave it
Block top	112, 118, 126	never collect a wall
Deposit area	31, 157, 72	**deposit**: R < 75 · G 130–200 · B 40–110

TWO REAL IMAGES

Something to collect, somewhere to deposit



8.9 s · search mode: iron in the box, **27, 24, 26**.



35.9 s · home mode: green in the box, **32, 159, 72**.

Both decisions use the same two pixels: **Camera[17][11]** and **Camera[17][12]**.

HOME MODE · THE GEOMETRY

Point the robot at (216, 156)

goal = atan2d(156 − PositionY, 216 − PositionX) − 90
The heading from here to the deposit centre.

turn = angleDiff(goal, Compass) × 0.05
WheelLeft = 3 − turn, WheelRight = 3 + turn.

When	Position	Compass	goal	off by	turn
3rd catch, 18.7 s	(29, 45)	220°	301°	81°	4.05

A big turn first, then the robot drives a nearly straight line home: at the deposit it was off by only **-1°**.

PART 3

Step by step in C

Search mode and home mode in one function

THE PLAN

Five rules, one mode switch

1. Wall or block ahead (US_Front < 15) → **turn** for 0.6 s.
2. LoadedObjects ≥ 3 → **home**: green in the box → deposit, else steer to (216, 156).
3. Iron or gold in the box → **collect**, red LED.
4. Iron or gold in view → **steer to the nearest**.
5. Nothing in view → **explore** on a gentle curve.

Rule 2 is the mode switch: once it is true, rules 3 to 5 never run again.

STEP 1 · LINES 3–18

Two tests and one action

```
3  int thing(int y, int x) {                  // iron or gold (silver looks like the block)
4      int r = Camera[y][x][0], g = Camera[y][x][1], b = Camera[y][x][2];
5      return max(r, max(g, b)) < 60 || (r > 195 && b < 80);
6  }
7
8  int green(int y, int x) {                  // deposit area
9      int r = Camera[y][x][0], g = Camera[y][x][1], b = Camera[y][x][2];
10     return r < 75 && g > 130 && g < 200 && b > 40 && b < 110;
11 }
12
13 void hold(int r, int g, int b, int led) { // stop, show the LED, hold 3.2 s
14     WheelLeft = 0; WheelRight = 0;
15     LED_1 = led;
16     LED_R = r; LED_G = g; LED_B = b;
17     Duration = 3200;
18 }
```

thing: iron (max < 60) or gold.

green: the deposit colour.

hold(r, g, b, led): collect and deposit share it.

STEP 2 · LINES 20–25

Find the nearest metal

```
20  void Game1() {
21      LED_1 = 0;
22      int mid = CAM_W / 2, last = CAM_H - 1, row = -1, col = 0;
23      for (int y = 0; y < CAM_H; y++)
24          for (int x = 0; x < CAM_W; x++)
25              if (thing(y, x) && y > row) { row = y; col = x; }
```

row, col: the lowest iron or gold pixel.

base: last = CAM_H − 1, the bottom row.

STEP 3 · LINES 26–35

Safety, then home mode

```
26      if (US_Front < 15) {                                    // wall or block ahead
27          WheelLeft = -2; WheelRight = 2; Duration = 600;
28      } else if (LoadedObjects >= 3) {                        // full enough: go home
29          if (green(last, mid - 1) && green(last, mid)) {
30              hold(0, 255, 0, 1);                             // deposit: green, steady
31          } else {
32              float goal = atan2d(156 - PositionY, 216 - PositionX) - 90;
33              float turn = angleDiff(goal, Compass) * 0.05; // > 0: turn left
34              WheelLeft = 3 - turn; WheelRight = 3 + turn;
35          }
```

Both box pixels green → **hold(0, 255, 0, 1)**: deposit.

Else: **goal** and **turn** as in Session 18.

STEP 4 · LINES 36–44

Search mode

```
36      } else if (thing(last, mid - 1) || thing(last, mid)) {
37          hold(255, 0, 0, 2);                              // collect: red, flashing
38      } else if (row >= 0) {                               // steer to the nearest object
39          float err = col - (CAM_W - 1) / 2.0;
40          WheelLeft = 2 + err * 0.25; WheelRight = 2 - err * 0.25;
41      } else {
42          WheelLeft = 3; WheelRight = 3.3;                 // explore
43      }
44  }
```

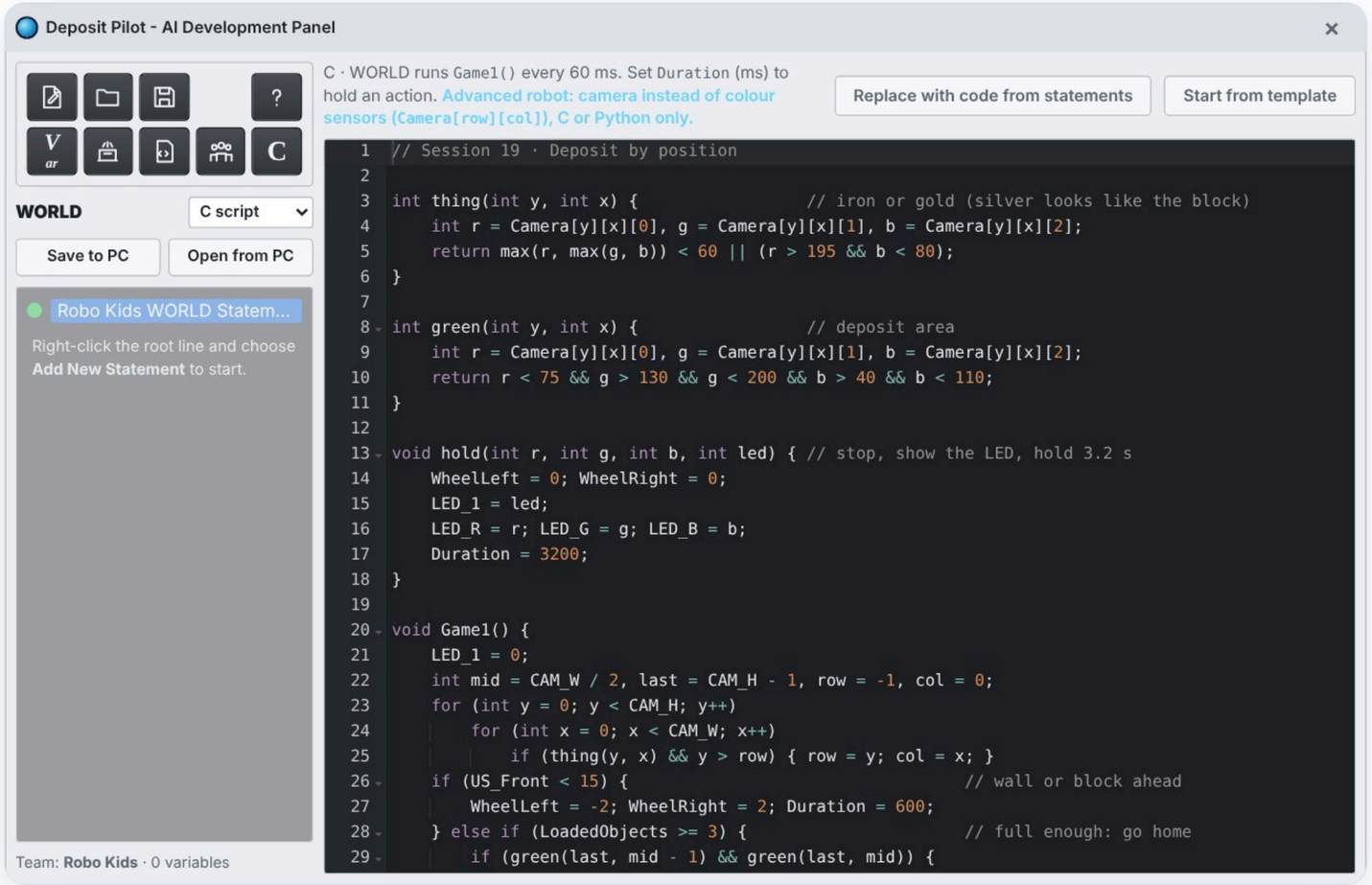
Metal in the box → **hold(255, 0, 0, 2)**: collect.

Steer at speed 2: slow enough to stop on it.

BUILD AND LOAD

The whole program, built

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





RUN · 8.9 S

Search mode: the first catch



At (**36, 63**): iron in the box, **27, 24, 26**.

Two more iron nearby: the next catches come at 13.3 s and 18.7 s.



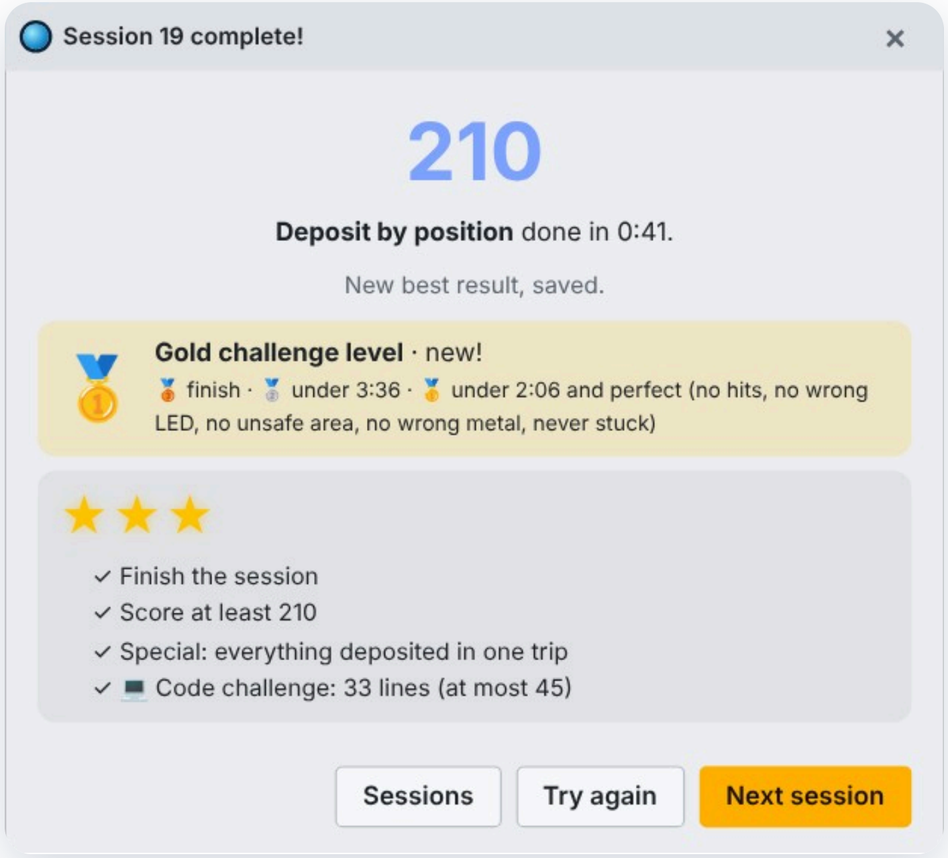
RUN · 35.9 S

Home mode: deposit



At (**193, 149**), Compass **288**, heading off by **-1°**.

Box: **32, 159, 72** → green, steady LED, all three in one trip.



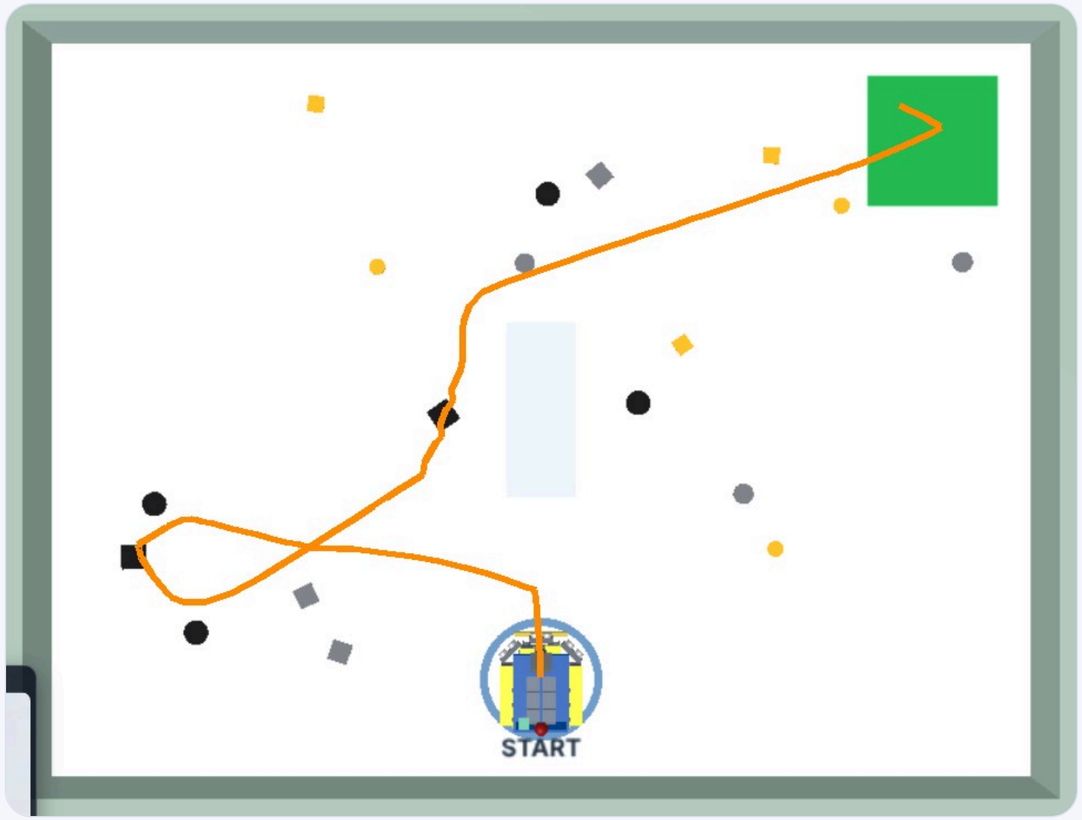
RESULT · TESTED

210 points in 0:41 · Gold

Medals · Bronze: finish · Silver: under 3:36 · Gold: under 2:06 and perfect.

Stars · finish · score 210 · everything deposited in one trip.

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



THE PATH · RECORDED

A wiggle, then a straight line home

Search mode: short moves between nearby iron.

Home mode: one long, nearly straight line to the corner.

Camera for **near**, position for **far**.

DID NOT WORK?

Check first, then tune carefully

What you see	What to do
Never goes home	The mode test is **LoadedObjects >= 3** and must come before the search rules
Arrives but does not deposit	Check green(): **R < 75, G 130–200, B 40–110**, both box pixels
Pushes against the block	It is chasing grey: keep silver out, the block top is **112, 118, 126**
Drives the wrong way home	Check **atan2d(156 – PositionY, 216 – PositionX) – 90**
Deposit LED wrong	Deposit: **LED_1 = 1**, green **0, 255, 0**

NEXT TIME

University Presentation 9 · Final challenge

Everything together: the pond, unsafe areas, Oil and Rocks, two deposit areas and **six objects in one trip**.