

ABOUT THE PROJECT

Code-A-Bot is an interactive digital game which asks the player to program robot workers in the fictional recycling plant DERP (Department of Environment Resource Processing).

The robots are programmed to move around the facility's recycling plant and use manipulators to pick up different types of rubbish: metal, glass, plastic, paper, and e-waste. The robot must navigate to the correct bin using the April tags (bin labels) to guide them. When the robot successfully recycles a piece of rubbish, DERP's efficiency score increases.



Robots in the DERP recycling facility

GAME MISSION

The robots (bots) are loaded with basic code and attachments, but they are waiting for you to improve their efficiency by upgrading their hardware and coding instructions.

On screen activity | stage 1 | robot design

The robots must be fitted with sensors to collect information about their environment.

Sensors, cameras, lights, and aerials can be added to the bot by touching the coloured flashing "ports".

Ports on the front of the robot:

- Blue port = Sensors
- Red port = Manipulators

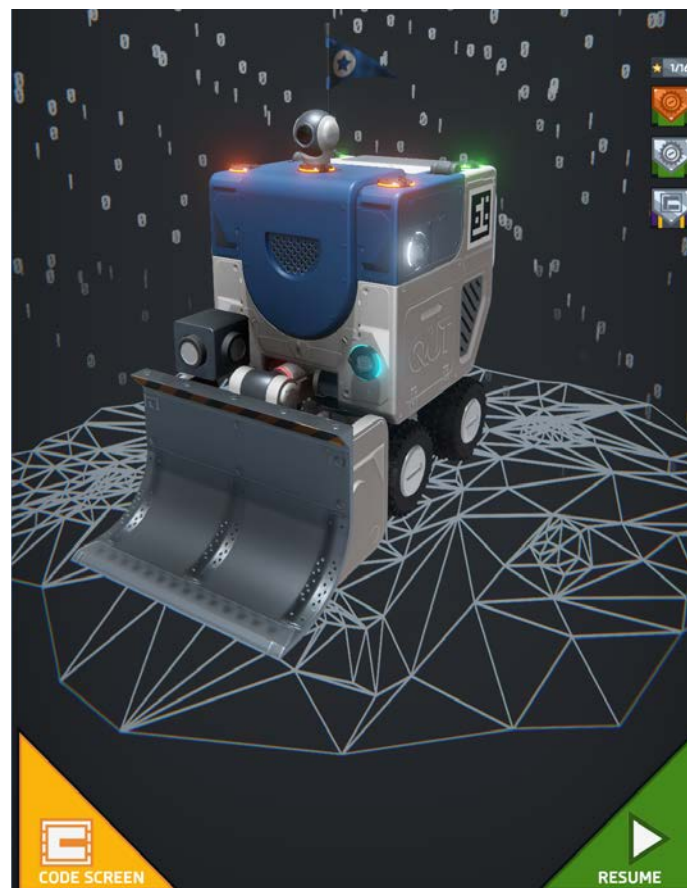
Ports on top of the robot:

- Orange port = Cameras
- Green port = Aerials

Ports on the side of the robot:

- White port = Lights

Hint: Aerials and lights are not required but can be added to customise your robot.



Robot with coloured flashing sensor ports

Sensors

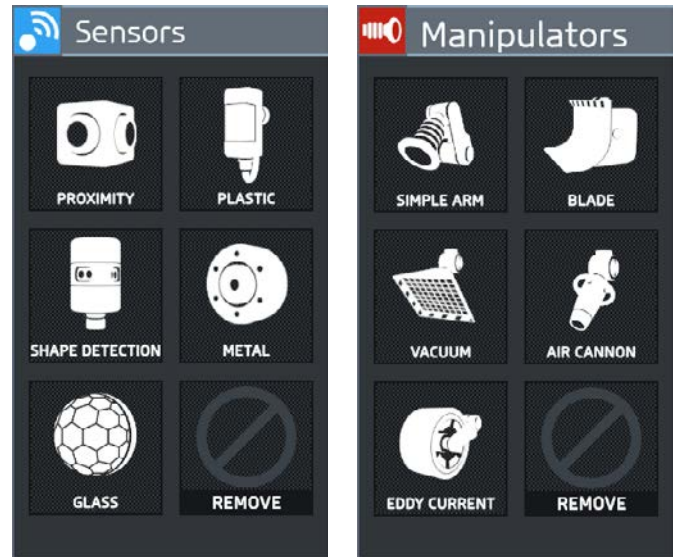
Touch the blue flashing ports to find the Sensor Menu. This menu shows the options for different types of sensors:
Proximity, Plastic, Shape Detection, Metal, Glass.

Hint: You can fit up to 2 sensors to each bot, but without a Proximity sensor the bot will not be able to navigate around the arena.

Manipulators

Your bot needs to be fitted with a manipulator to collect rubbish. Touch the attachment located on the centre front of the bot to reveal the Manipulators Menu. Manipulators include: Simple Arm, Blade, Vacuum, Air Cannon, Eddy Current.

Hint: The manipulator you choose will affect the code options which are available in the code panel.



The sensor and manipulator panels

On screen activity | stage 2 | code

Code Screen

Access the code screen by the touching the triangular yellow “code screen” button, or by touching the code pictured on the back of the robot.

The code screen contains:

1. Arena Mini-map – At the top of the screen is a diagram showing the position and direction of your bot in the recycling arena.

2. Code Menu – Code statements and movements are listed in the code menu. Select the icon on the top menu for the item you want to code.

Lights | Manipulators | Sensors | Cameras | Statements



3. Code Panel – In the centre of the screen is a floating code panel into which you drag code blocks from the code menu to program your bot.

Test: Try dragging and dropping some of the code blocks into the code panel. To delete the code blocks, drag the block back out of the code panel toward the code menu which will highlight red to “delete”.



Code screen with code menu and panel

CODE-A-BOT ON SCREEN CHALLENGES

Challenge 1: Move Forward

- Use the movement blocks to code your bot to go forward.
- Test your code by hitting the green triangle “run your code” button.
- Try out other basic code commands, observe what the robot does when you run the code.

Tip: View your robot moving in the main area on the projection screen and on the mini-map above the code menu. To return to the code screen touch the code panel located on the back of your bot.

Code Statements

Review the Code Statements:

- **IF Statement** – executes instructions IF a statement is TRUE.
- **WHILE Statement** – carries out the instructions WHILE a statement is TRUE and will continue to carry out the code held within its block until the statement is no longer true.
- **LOOP for** – A loop is a block that causes the commands contained within the block to repeat for the chosen amount of times.

Challenge 2: Proximity Sensor

- Instruct your bot to move forward, and while in proximity (close to) the arena wall, turn right.
- Run your code.

Did your bot avoid running into the arena wall?

Tip: Click the + “avoid objects” button at the bottom of the code menu to reveal sample code which includes a while statement and proximity sensor.

Congratulations, you added a proximity sensor and used code to ensure your bot did not run into the arena wall.

CODE STATEMENTS







Statements and movement:

Use these code statements to build up your code to enable your bot to perform more complicated tasks.

if

while

loop for

wait

▲ forward

◀ left

▶ right

✓ reverse

Code statements menu panel

CODE-A-BOT ON SCREEN CHALLENGES

Challenge 3: Pick up Plastic

Before you begin, you must fit the bot with the sensors and manipulators required for this challenge:

- To move from the code screen to the robot touch the triangular blue “load out” button.
- Fit a “proximity” sensor and a “plastic” sensor.
- Fit the “simple arm” manipulator.
- Code your bot to go (in any direction) and pick up (arm grab) a piece of plastic rubbish.

Tip: Remember to break down each step required and test as you go. Check your work by fitting a simple arm manipulator to the bot, and then click the + “Pick Up Plastic” button at the bottom of the Code Panel to view an example of this code.

Challenge 4: Rubbish to Bin (advanced)



- Code your bot to pick up a piece of rubbish, take it to the recycling bin, and drop it in the correct bin.
- First fit your bot with a High Definition (HD) camera.
- You will need to use WHILE statements.
- Don’t forget to instruct your bot on all steps, including dropping the rubbish into the bin.

Tip: Check your work, click the + “Go to Plastic Bin” button below the code menu to see some sample code for getting the bot to the plastic bin.



Robot fitted with a simple arm manipulator



Code-A-Bot Achievements

Did you know *Code-A-Bot* has a set of 16 achievements you can work through to advance your skills? The achievements are revealed by touching the 3 icons running down the right-hand side of the screen.

Congratulations on trying Code-A-Bot! Code-A-Bot encourages problem solving and computational thinking by asking us to code a robot. Well done and thanks for helping to improve the efficiency of the DERP recycling facility!