

First choose roles

**Designer****Designer** places the items on the grid and plans the route the robot will take to get to them**Writer****Writer** records the steps with a whiteboard and pen / piece of paper and pencil**Tester****Tester** takes the whiteboard and follows the instructions that are written down

How about swapping roles as you go, so that everyone has a turn?

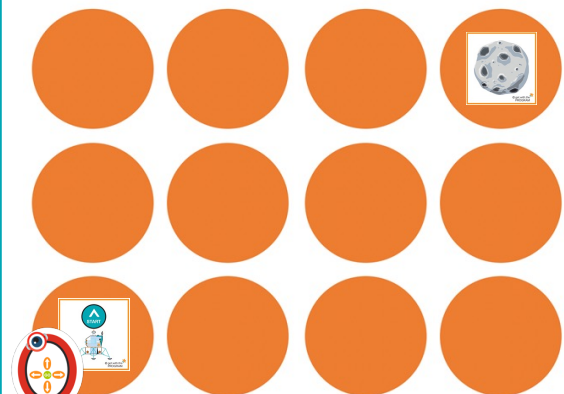


1

Challenge 1 – Get to the item

Can you move across the moon to collect the moon rock?

- 1. Designer:** Place the lunar lander for 'start' and the moon rock for 'finish' anywhere on the grid. Use the robot to **design** the route the tester needs to follow to get across the moon to the moon rock.
- 2. Writer:** Record the route (**algorithm**) the tester needs to take on the whiteboard using forward, left and right arrows.
- 3. Tester:** Follow the instructions on the whiteboard to **test** the algorithm. You must follow the instructions exactly, even if you know they aren't right – you might go off the grid! If this happens, don't worry. Work with the designer and writer to **debug** the **algorithm** and try again!

Example grid layout:

Essential Skills being developed:



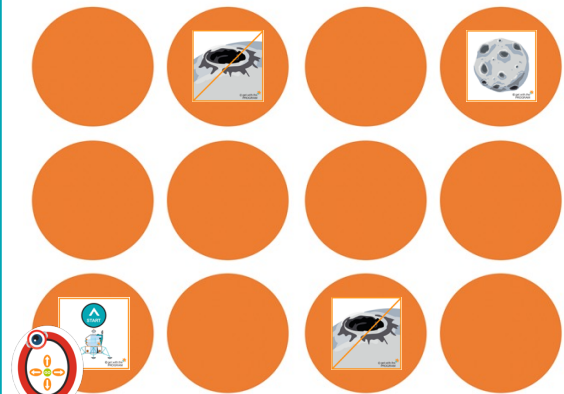
2

Challenge 2 – avoid items

There is another moon rock to collect but some craters we want to avoid. Can you program the robot to miss these items?

- 1. Designer:** Choose where to add in the extra craters. **Design** the **algorithm** as before, making sure to go around the 'avoid' items to get to the 'finish' item.
- 2. Writer:** Record the **algorithm** on the whiteboard using forward, left and right arrows. It will probably be a bit more complicated than last time!
- 3. Tester:** Use the whiteboard to follow the **algorithm** to **test** it. Did you get it right? **Debug** if needed and try again.

Example grid layout:



Essential Skills being developed:



3

Challenge 3 – creative coding

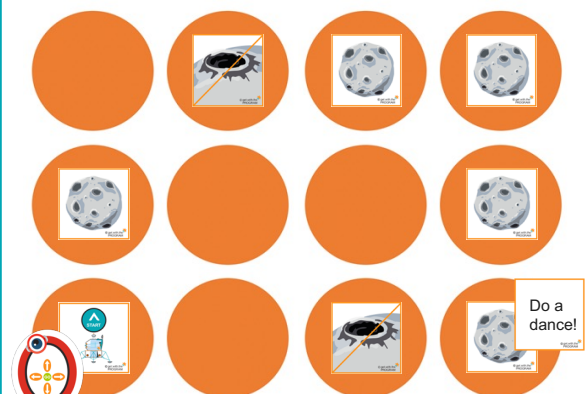
Now, can you use your imaginations to think of some other things to include in your algorithm?

- Maybe you can add in lots of extra moon rocks, and design a route for the robot to collect them all?
- Could you create a specific instruction to pick up items? What would this look like in the algorithm?
- Maybe the robot can jump over 'avoid' items?
- Maybe they should do a happy dance when they've successfully collected the moon rocks?

What other creative coding can you think of?

- 1. Designer:** Plan what steps to add in, and where they should go in the **algorithm**. You can use one of your blank squares to draw on or cut up some paper.
- 2. Writer:** Record the **algorithm** on the whiteboard – make up new pictures for the new actions.
- 3. Tester:** Use the whiteboard to follow the **algorithm** to **test** it. **Debug** if needed and try again.

Example grid layout:



Essential Skills being developed:



4

Extension challenges

These extension challenges build towards Scratch programming and use more complex key terms to do it! Here is some information about the terms



Repetition

Repetition is looping or repeating sections of an algorithm so the steps can be reused



Selection

Selection is making a choice to take a path in the algorithm, based on other information you have



Input

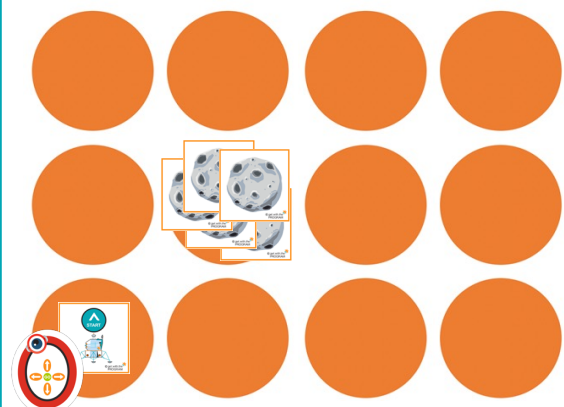
Input is when you have some information that comes into the algorithm

Extension challenge 1 – repetition

For this challenge, we are going to put all the moon rocks in one place, and program the robot to collect them one-by-one and bring them back to the start.

- 1. Designer:** Can you program the robot to do this?
The robot can only carry one item at a time so will need to return to the start to repeat the instructions again.
*A loop that is repeated like this is called **repetition**.*
- 2. Writer:** Record the **algorithm** on the whiteboard using forward, left and right arrows, and including a **repeat** instruction - think about what this might look like.
- 3. Tester:** Use the whiteboard to follow the **algorithm** to test it. Did you get there and back without any problems? Don't worry if it didn't go as planned – **debug** and try again!

Example grid layout:
(also see extension demo video / slides)



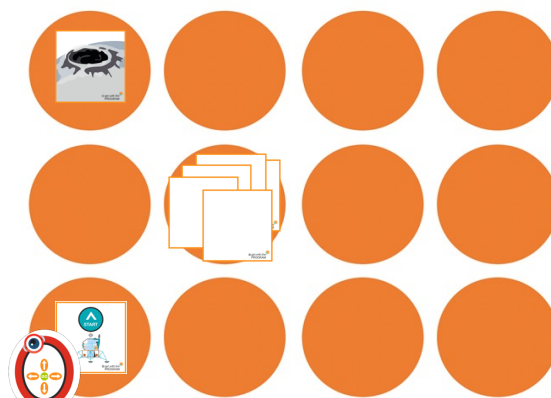
Hint: What position does the robot need to be in at the end before it can repeat the algorithm?

Extension challenge 2 - selection

For this challenge, can you help the robot make the right choice when deciding between two options of moon rocks?

- 1. Designer:** Shuffle the cards and lay them face down on one spot somewhere on the grid. As before, you will need to plan the route the robot should take to get to the items. The robot will pick up the items one-by-one – if a moon rock is whole, the robot needs to take it back to the 'start' lunar lander. If it is a broken one, the robot needs to take it to the crater to throw it away.
- 2. Writer:** Record the **algorithm** as planned out by the designer on the whiteboard using forward, left and right arrows. When you get to the cards, you will need to include pick-up and **selection** instructions. Choose how you will write this selection (e.g. a choice of path or 'IF' step) in your instructions.
- 3. Tester:** Use the whiteboard to follow the **algorithm** to test it. Don't forget to test both paths! Don't worry if anything doesn't go as planned – **debug** and try again!

Example grid layout:
(also see extension demo video / slides)



Essential Skills being developed:



7

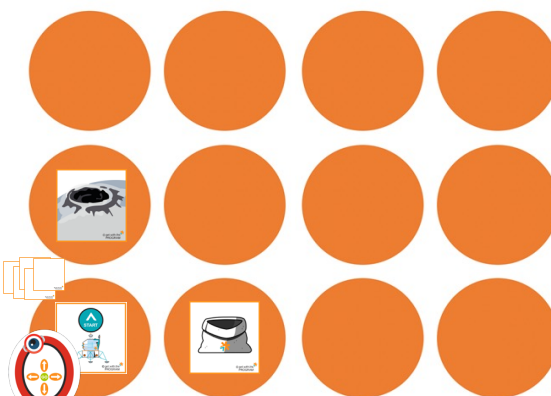
Extension super challenge – with added input

This challenge builds on the previous two extension challenges (repetition and selection) with added input:

For this challenge, the designer or writer is going to be in charge of **input** – decide who it will be before you start. When the **algorithm** begins, you will need to stand next to the start and hand the tester the cards one-by-one, this is the **input**. When the tester is handed a card, they must decide whether it is a whole moon rock, and take it to the new moon rock bag, or if it is a broken one, in which case it goes into the crater.

- 1. Designer:** Place the robot on the 'start' lunar lander and put the moon rock bag and crater nearby. Hand all the moon rock item cards face down to the person controlling the **input**. Plan the route the robot takes to both the moon rock bag and crater.
- 2. Writer:** Record the **algorithm** on the whiteboard. You will need to use an 'IF' step at the beginning, for **selection**, then use forward, left and right arrows to get to the bag or crater, and include a **repetition** or loop step at the end. What might these look like?
- 3. Tester:** Turn over the card when you are handed it and use the 'IF' step on the whiteboard to decide whether to take it to the bag or crater. Follow the **algorithm** to **test** it. **Debug** if needed.

Example grid layout:
(also see extension demo video / slides)



Essential Skills being developed:



8