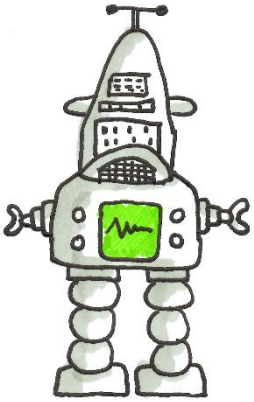
Mission to Mars – Algorithms



Age range: These lesson activities are pitched at Year 2 to Year 3 pupils but it could be adapted for younger or less able pupils.

Differentiation:

- Do it all verbally
- Make picture instruction cards for pupils to order

Lesson 1

Learning Objectives:

Pupils will:

- Create, test and refine precise instructions for a specific purpose
- Know what an algorithm is
- Express an algorithm using simple symbols

This activity works best in a large space

Pupils will work in groups – each group needs one pupil to act as the 'robot'

Resources:

Each group will need some PE equipment (benches, rounders' bases, skipping ropes etc.) arranged as obstacles. Paper or individual white boards and pens. A copy of the Robot Code Library

Set the scene

Tell pupils that they are space explorers. Their space craft uses B-Rainium as fuel. They have run out of fuel! They have made an emergency landing on a planet that has a lot of B-Rainium rocks. Unfortunately, the planet's environment is too hostile for humans (even in space suits). The only way to collect the B-Rainium and refuel the space ship is to use the ship's robot.

The robot only does exactly what it is told to do. Each group need to create an algorithm for the robot to follow so that it can move across the planet's surface (the school hall) around various obstacles (PE Equipment) and collect some B-Rainium (a ball or box).

Warm up activity

Each group of pupils instructs a 'robot' pupil through the obstacles to the B-Rainium using natural language. The 'robot' child must only do what they are instructed to do. At first pupils are likely to give vague instructions such as 'walk' or 'go round the mat'. Robots can't follow these sorts of instructions. Encourage children to give more precise instructions such as 'walk forward 3 steps'.

A set of precise instructions for a special purpose is called an algorithm

Discuss any issues/problems that arise.

Main activity

Explain that robots do not understand natural language and that the robot on the space ship only understands a few basic commands (see the Robot Code Library).

In groups, write an **algorithm,** using the Robot Code Library, for the robot child to follow.

An algorithm to walk forward six steps, turn right and walk two steps might look like this

Repeat the warm up activity but this time with the robot following the algorithm exactly.

Most groups are likely to have some 'bugs' or errors in their algorithm meaning that the robot will not be able to navigate through the obstacle course.

Debugging is an important part of programming. It doesn't mean that you have failed! It is an essential ingredient in programme development. Allow pupils to find and correct their own (or each other's) errors.

Allow pupils to call the robot back to base, debug (correct) the algorithm and re-run the programme.

Repeat as necessary.

Development

Pupils will have been drawing an arrow for each step or input a move forward command for each step. If the robot or floor turtle needs to move several steps this can get pretty tedious. Can pupils think of a way to instruct the robot to move several steps without having to write a command for each individual step? Many programming languages use a 'repeat' command followed by the number of repetitions required.

This command for walking eight steps could become

How could pupils adapt the  and  (turn right and turn left) commands to allow greater accuracy and finer control?

Allow pupils to write (and debug) an algorithm for guiding the robot back to the ship using their new commands. What impact do these commands have on their algorithm?

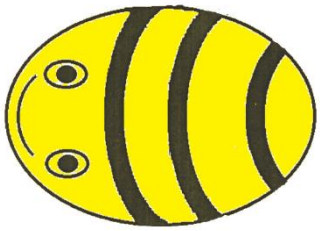
Plenary

Ask pupils what would happen in a real situation if the algorithm was wrong.

What could the pupils have done increase their chances of getting the algorithm right before they sent the robot out? Discuss the importance of planning, trial and error and debugging.

Lesson 2 Mission to Mars - Implement anAlgorithm on a device

Learning objectives:



Pupils will:

- Create and debug simple programs
- Know how algorithms are implemented as programs on digital devices
- Know that programs execute by following precise and unambiguous instructions

Lesson Prep

Pupils are going to work in groups to program a floor turtle/ beebot to follow a pre-defined route around a maze. The maze could be created with piles of 'junk' or could be drawn on a large sheet of paper. The route through the maze must be big enough for the turtle/ beebot to pass through easily.

Use the fakebot template to create a fakebot for each group or pair of pupils. (A fakebot is a card representation of the floor robot/ beebot that can be used to test the algorithm. One pupil reads out the instructions whilst another moves the fakebot in response to the instructions.

Recap

Remind pupils of their work with 'robots' last lesson and recap algorithms and the importance of precise instructions.

Main activity

In groups pupils should:

1) Discuss/write instructions for guiding a fakebot through the maze

2) Discuss/write instructions for guiding the floor turtle/ beebot using the device's commands – For younger or less able pupils provide picture instruction cards

3) Test the algorithm with the fakebot. Are all of the instructions correct? Debug the instructions and retest the algorithm using a 'fakebot'.

4) Input the algorithm into the floor turtle/ beebot. Test the programme. Find and fix any bugs.

Plenary

Discuss issues and check understanding. Does the sequence of commands matter? Does turn right followed by move 4 have the same effect as walk 4 turn right? Compare algorithms – are they are the same? Are some more or less efficient?