

Learning Intention

Design, test, and debug an algorithm in Scratch to control Sprites to move and communicate!

Resources

- Moon Landing Coding Adventure Show
- Computers with internet access to <https://scratch.mit.edu/>
- Teaching guide PowerPoint
- Moon Landing Scratch Programming activity explainer video
- Activity detailed guide pdf
- (Optional) Printable worksheets and certificates

Keywords

- | | | |
|-------------|----------------|--------------|
| • Algorithm | • Sequence | • Repetition |
| • Design | • Test | • Debug |
| • Input | • Forever loop | |

Introduction

Watch the Moon Landing Coding Adventure Show with the class, discuss key points, and ask pupils questions to determine their understanding.

Next, watch the Moon Landing Scratch Programming Activity explainer video to guide you through the activities. If children have previously used Scratch, recap some of the features and functions.

Ask children questions throughout to determine their understanding of the tasks. Ensure children are working in appropriate pairs and differentiate accordingly.

Alternatively, coding steps can also be followed using the more detailed teaching guide, and students can be given the quick guides or detailed guides to explain the challenges.

If time, watch the Moon Landing Scratch Programming Activity Extension explainer video, or show the extension slides, to explain the extension activities.

Plenary

As a class, discuss the successes and challenges of the tasks. Was everybody able to successfully program their code for AL the Robot to sort the rocks going back to Earth? Were they able to make AL the Robot talk? Did anyone need to debug their code and if so, how?

Learning Outcomes

- Design an algorithm to achieve a goal
- Test and debug an algorithm
- Include repetition and understand how it makes the algorithm more efficient
- Use examples of input

Key Learning Goals*

Pupils should be taught to:

- design, write, and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts
- use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs

*These learning goals align to the National Computing Curriculum in England - Key Stage 2 (ages 7-11)

Health and Safety

Ensure children are supervised when using computers. Monitor safe use of internet access and ensure e-safety.

Main Activity

Set up: Set the scene – delete the Cat sprite and change the stage backdrop to Moon.

Challenge 1: Create Rock, Tick and Cross sprites. Follow the instructions to rename, resize, and re-position these, and change the Cross sprite's 'costume' to red. Program your Rock (or sorting) sprites to glide to the tick or cross when clicked and return to their starting position when reset.

Challenge 2: Program an AL the Robot sprite to sort the rocks. Follow the instructions to add code for all the sprites to go to AL the Robot before going to the Tick or Cross and return to starting position when reset.

Extension Challenges: Program the rock sprites to move around the screen. Try to 'catch' them to send to AL the Robot for sorting. Program AL tell you when she's sorted the sprite and if time, follow the instructions to create a timer for your game. Investigate the different outcomes when following the suggested steps in the instructions.

Assessment Questions

- What is it called when you have some information that comes into the algorithm?
- What input did you have in your code?
- What examples of repetition did you have?