

3. AUTOSUB6000'S OCEAN FLOOR MISSION

AGE 9-11

Objectives

To create a project on Scratch to allow Autosub6000 to explore the ocean floor and photograph the specimens found.

To understand the sequence of instructions that makes a computer program.

To be able to debug faulty programs.

The big questions

How is the submersible Autosub6000 controlled?

How can Scratch be used to simulate this?

Unit summary

Learners will use Scratch, to debug and then improve a program to move Autosub6000 around the ocean floor, photographing samples found. When encountering a creature, Autosub6000 will say "Photographing sample..." before then saying "You have found a ..." The sample will then disappear. The remote movement will be controlled through a keyboard's arrow keys initially and then the children will be challenged to create a program which will move Autosub6000 autonomously.

Learners will need a basic understanding of Scratch before carrying out this Unit. Guides and online tutorials are available here if needed <https://scratch.mit.edu/help/>

Background

Autosub6000 is a type of autonomous underwater vehicle (AUV). It is used to explore the world's oceans without a pilot or any tether, including the Polar Regions as it can work under ice. Before the submersible is launched, it is programmed with instructions about where to go, how deep to go and what to measure. More information can be found at: <http://noc.ac.uk/facilities/marine-autonomous-robotic-systems/autosubs>

Scratch allows children to create programs to control sprites. Children will complete a program which allows them to control an Autosub6000 sprite along the ocean floor, photographing specimens, using a keyboard's arrow keys initially and then creating a program to move Autosub6000 autonomously.

The scratch reference guide

<https://download.scratch.mit.edu/ScratchReferenceGuide14.pdf> explains the meaning of the blocks you will need to use to create the scripts which program the sprites.

Curriculum links

Computing KS2

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