Plan and program a monologue

Computing Science Concepts

-Sequence
-Order can be important in sequences
-Algorithm
-Use, modify, create
-Four levels of abstraction
-Inputs

National Curriculum Programs of Study

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 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

Research Focus

This planning uses **use, modify, create**[1] Use involves exploring premade code before modifying it and finally building your own program. A process that takes the user from non ownership to ownership.

Computer scientists have **four levels of abstraction**[2]. The ideas level, Planning level (which includes the algorithm), code level, and execution level (testing the code).

Monologue Planning Version B Use, Modify, Create

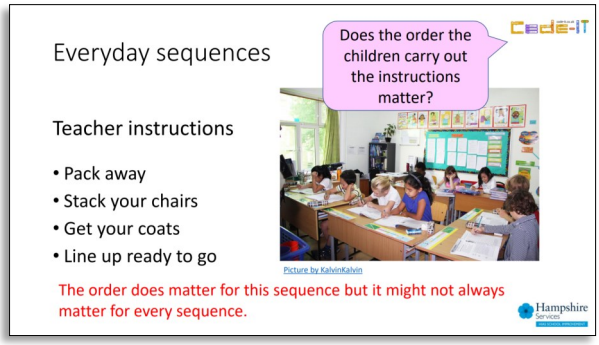
Overview

1. Introduce the concept of sequence through everyday examples
2. Extend the concept of sequence through role play
3. Introduce Scratch if it has not been used before
4. Introduce the challenge by sharing the idea, algorithm and example code for pupils to modify [Use, modify, create] ⇔
5. Pupils plan their own short monologue
6. Pupils turn their algorithmic planning into their own programmed monologue
7. Pupils fill in the summative assessment form

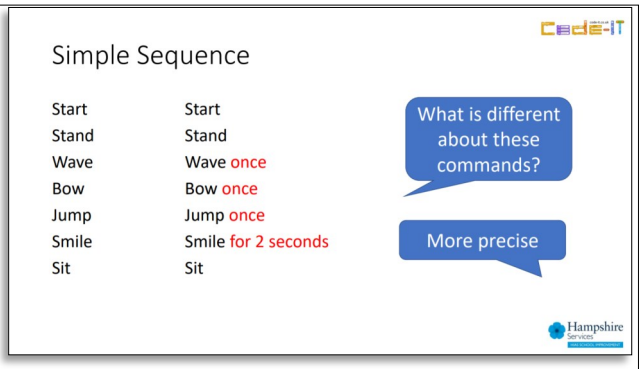
1. Introduce the concept of sequence through everyday examples

Download Everyday computing concepts PDF from http://code-it.co.uk/knowledge or directly from http://code-it.co.uk/wp-content/uploads/2019/04/everydaycomputingconcepts.pdf

Use the first eight slides to introduce the idea of sequence in our everyday lives. The answer mostly appear in red text. There are some sequences where the order is less important and some where the order is paramount. The same will be true for programming sequences. By linking the concept to its everyday use you are linking to known knowledge which means pupils are more likely to assimilate the idea.



2. Extend the concept of sequence through role play



Download Concepts before coding PDF from http://code-it.co.uk/knowledge or directly from http://code-it.co.uk/wp-content/uploads/2019/04/conceptbeforecoding.pdf Follow the links in the menu to simple sequence. Use those six slides to roleplay and write simple fun sequences. These slides introduce the idea that the more precise a sequence is the more useful it is. Stop when you get to the dance slide.

Formative assessment opportunity

While pupils are writing their own sequences go round and check them all. Is anyone struggling? Have they copied the one on the board exactly? This is often an indication that they are not sure how to create their own or that spelling is an issue. A good supportive activity is to get them to tell you about their own sequence that you scribe for them.

3. Introduce Scratch if it has not been used before

If pupils have never used Scratch before it is worth going over the basics. I have included some videos here Scratch 2.0 https://youtu.be/bNoyArexVns or Scratch 3.0 https://youtu.be/gtqMauyKE_w but I recommend that you watch them and introduce it yourself in short bursts.

Repeat until basic Scratch introduced

Show pupils basic feature **Introducing Scratch algorithm**

Pupils have time to try it out

Sprites are pictures that we can give instructions too through dragging blocks and snapping them together in the scripts area.

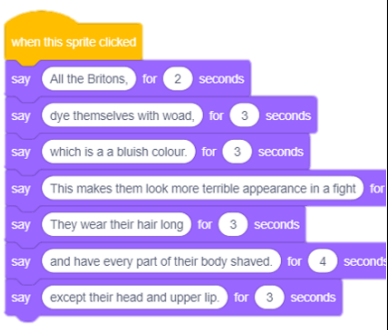
There are lots of different types of blocks and they are colour coded with similar blocks. Many of these blocks will contain ideas that we won't know how to use until upper KS2 or KS3.

Starting blocks have a curved top and can be found in the events section. Drag out the when this sprite is clicked starting block to show pupils.

4. Introduce the challenge by sharing the idea, algorithm and example code for pupils to modify [Use, modify, create]

Which template files to use

Start by giving pupils one example of planning either (say and wait blocks or timed say blocks) as well as the code so they can see what the author intended to make and why. Direct them to the other form once they have finished exploring the first.



Julius Caesar on the Britons

What challenge to use

This module comes with four pre-made challenges.

Tables –3 times tables (say and wait blocks)

Victorian –Victorian working conditions (say and wait blocks)

Julius C –His account of the Britons when he invaded (timed say blocks)

Churchill –His speech in 1940 (Timed say blocks)

You can of course make your own really easily and there is a blank word template to create the idea and algorithm. Scratch supports copy and paste.

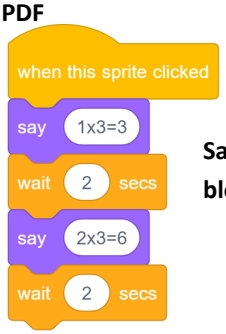
Idea and algorithm planner

If pupils are only writing a monologue use the A4 monologue planner.

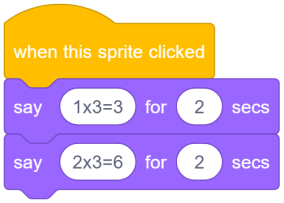
monologuealgorithmplannerA4 PDF

If pupils are going to complete the stage and sound module then use the left half of the A3 monologue planner.

monologuealgorithmplannerA4 PDF



Say and wait blocks



Timed say blocks

Using the template file

Direct pupils to open the template file. This can be found either on the Scratch website at https://scratch.mit.edu/projects/303987509/

Or as a downloadable file to put on your network

Scratch 2.0 **monologueVBscratch2 ZIP**

Scratch 3.0 **monologueVBscratch3 ZIP**

You could also adapt this planning to make it read the text or translate it but this would increase the complexity of the project.

Introducing the challenge

Look at the planning first. What did the author intend to create? Why have they included seconds?

Use

Ask pupils to run the code by clicking on the sprite. What happens? Is the code the same as what is shown on the screen? Does it follow the same order? How do we know the time wait blocks are working?

Modify

Encourage Pupils to change one thing in the code that is not a background or sprite and then see how this affects how it runs. Tel the class that you will ask them to describe their changes and the difference or lack of difference this made. Give them time to explore the examples and parts of the program?

Ask pupils to describe their changes? What affects did they have?

Some pupils are happy with this open ended approach to modifying and others need more direction. There is an additional sheet with some modify tasks that you can give to those pupils that need it called **mono_modify_tasks.**

5. Pupils plan their own short monologue

Give out the idea and algorithm planner. You can simplify this by giving pupils an idea or you could let them choose their own idea. This could have any cross curricular theme.

Formative assessment

Check pupils algorithms as they are creating. Do they make sense? Have they kept to their idea? Have they included punctuation? Have they included timings? Do they know what type of blocks their algorithm can easily convert into? Although it is easy to adjust an algorithm to output in either say and wait or timed say blocks.

6. Pupils turn their algorithmic planning into their own programmed monologue

Give pupils time to do this. Those that finish earlier than others can create a different type of monologue.

7, Pupils fill in the summative assessment form

You can find a summative assessment quick Kahoot Quiz linked at http://code-it.co.uk/gold/

Whilst Kahoot is a limited assessment tool it is free and it is easy for teachers to pass the results back to code-it via phil.bagge@code-it.me so I can look at which method provides best short test results. Not conclusive but useful.

If you pass the results back please

1, Anonymise the results by removing the names

2, Ask the head teacher for permission

3, In the email title say which module you are doing (ie Animation D)

Research References

[1] **Use, modify, create** Irene Lee et al Computational thinking for Youth in practice (2011)

[2] **Four levels of abstraction**

This article includes an example of the four levels of abstraction and sign posts the work of Waite and Armoni in using them with school level pupils.

http://code-it.co.uk/algprogdiff/

Inspired by Behaviour Rubric created with @MarkDorling and linked at http://code-it.co.uk/attitudes/