Plan and program a dialogue

Computing Science Concepts

-Sequence

-Order is important for some sequences

-Inputs

-Algorithm

-Four levels of abstraction

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

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).

Dialogue Planning Version D Modelled Offline

Overview

1. Introduce the concept of sequence through everyday examples

2. Extend the concept of sequence through role play

3. Introduce the challenge by modelling it offline

4. Pupils plan their own short dialogue

5. Introduce Scratch if it has not been used before

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

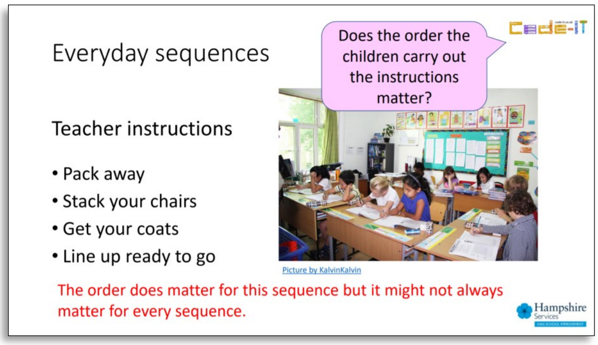
7. Pupils fill in the summative assessment form

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

If pupil used this material when creating a dialogue skip this step.

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.



Everyday sequences

Does the order the children carry out the instructions matter?

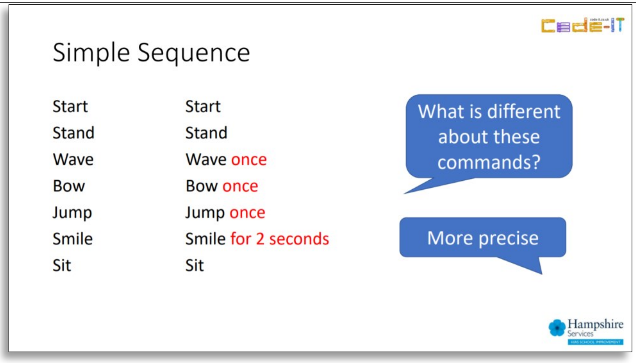
Teacher instructions

- Pack away
- Stack your chairs
- Get your coats
- Line up ready to go

Picture by KalvinKalvin

The order does matter for this sequence but it might not always matter for every sequence.

2. Extend the concept of sequence through role play



Simple Sequence

Start	Start
Stand	Stand
Wave	Wave once
Bow	Bow once
Jump	Jump once
Smile	Smile for 2 seconds
Sit	Sit

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.

If pupils used this material as part of their dialogue planning then skip to the sequence dance challenge which was not included in the monologue plan.

Formative assessment opportunity

While pupils are writing their own sequences go round and check them. 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 the challenge by modelling it offline

Write WAIT in large letters on two whiteboards. Hand one to a child or adult helper and keep the other one yourself. Have a very simple conversation about something you both know about. When you are speaking the child should hold up wait. When they are speaking you hold up wait. Explain that they are waiting for the same time that you are talking and visa versa. Now open up an interactive whiteboard or use your flip board chart to model a few lines of conversation similar to what is shown on the right.

Idea and algorithm planners

If pupils are only writing a dialogue use the **A4 dialogue planner.**

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

Character 1 Modern Girl	Time secs	Character 2 Viking Boy	Time secs
Do your parents know you have that sharp knife?	3	Wait	3
Wait Zig Zag Pattern	3 Total Time	All the children in our village carry knives apart from the slaves.	3 Total Time
You have slaves, that is so wrong!	3 Total Time	Wait	3 Total Time

Look for the zig zag pattern in the algorithm

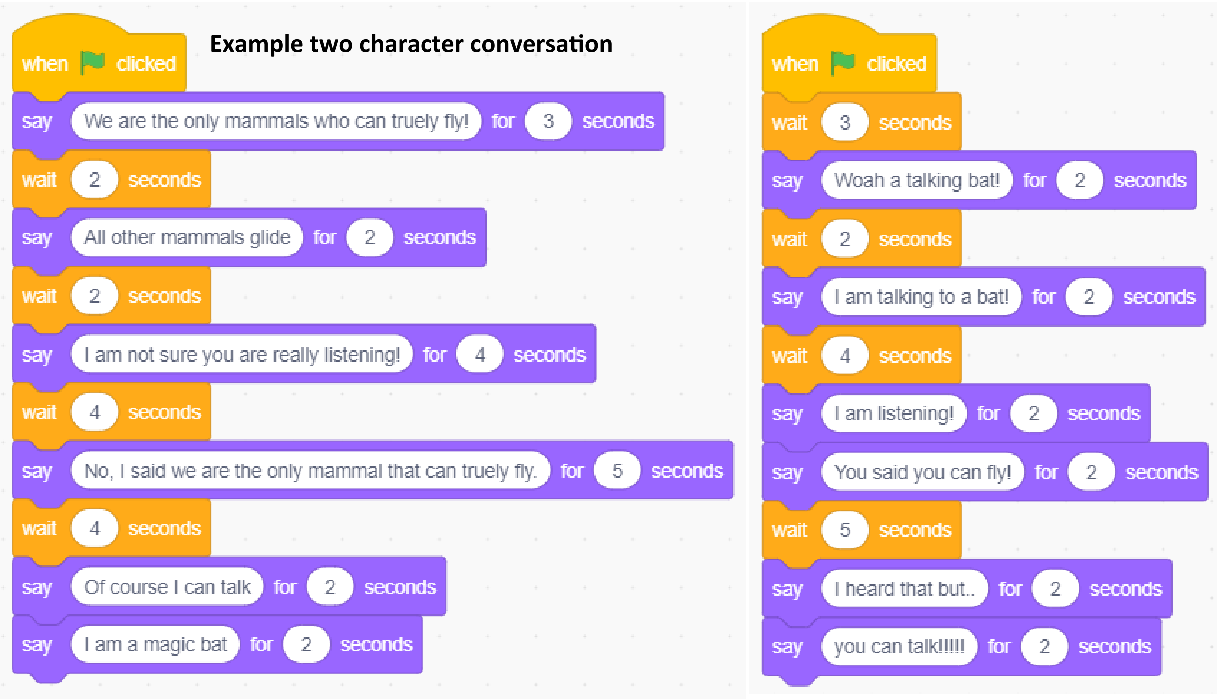
4. Pupils plan their own short dialogue

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? Can you see one character talk while the other waits?

Check pupils algorithms before they turn them into code.



Example two character conversation

```
when 🏴 clicked
say [We are the only mammals who can truely fly!] for (3) seconds
wait (2) seconds
say [All other mammals glide] for (2) seconds
wait (2) seconds
say [I am not sure you are really listening!] for (4) seconds
wait (4) seconds
say [No, I said we are the only mammal that can truely fly.] for (5) seconds
wait (4) seconds
say [Of course I can talk] for (2) seconds
say [I am a magic bat] for (2) seconds
```

```
when 🏴 clicked
wait (3) seconds
say [Woah a talking bat!] for (2) seconds
wait (2) seconds
say [I am talking to a bat!] for (2) seconds
wait (4) seconds
say [I am listening!] for (2) seconds
say [You said you can fly!] for (2) seconds
wait (5) seconds
say [I heard that but..] for (2) seconds
say [you can talk!!!!!] for (2) seconds
```

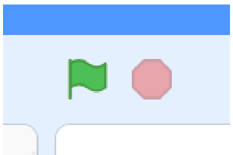
5. Introduce Scratch if it has not been used before

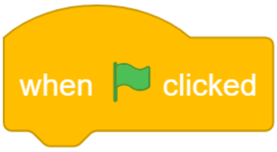
If pupils have never used Scratch before it is worth going over the basics. I have included some videos here Scratch 2.0 https://youtu.be/bNoyArexVns or Scratch 3.0 https://youtu.be/gtqMauyKE_w but I recommend that you watch them but introduce it yourself in short bursts.

Repeat until basic Scratch introduced

Show pupils basic feature

Pupils have time to try it out





```
when 🏴 clicked
```

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 green flag clicked starting block and show pupils where this can be triggered on the Scratch display.

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

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

Optional WAGGOL

10/15 minutes into the session ask pupils to talk to their neighbours and discuss what criteria they will use to determine if their program works. Draw out that the conversation will make sense. That one character will talk while another will listen. That there will be enough time to read the speech bubbles but not so much time that the reader gets bored. That the subject matter will be interesting and engage the reader.

Pupils could use this list to peer assess each others work.

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

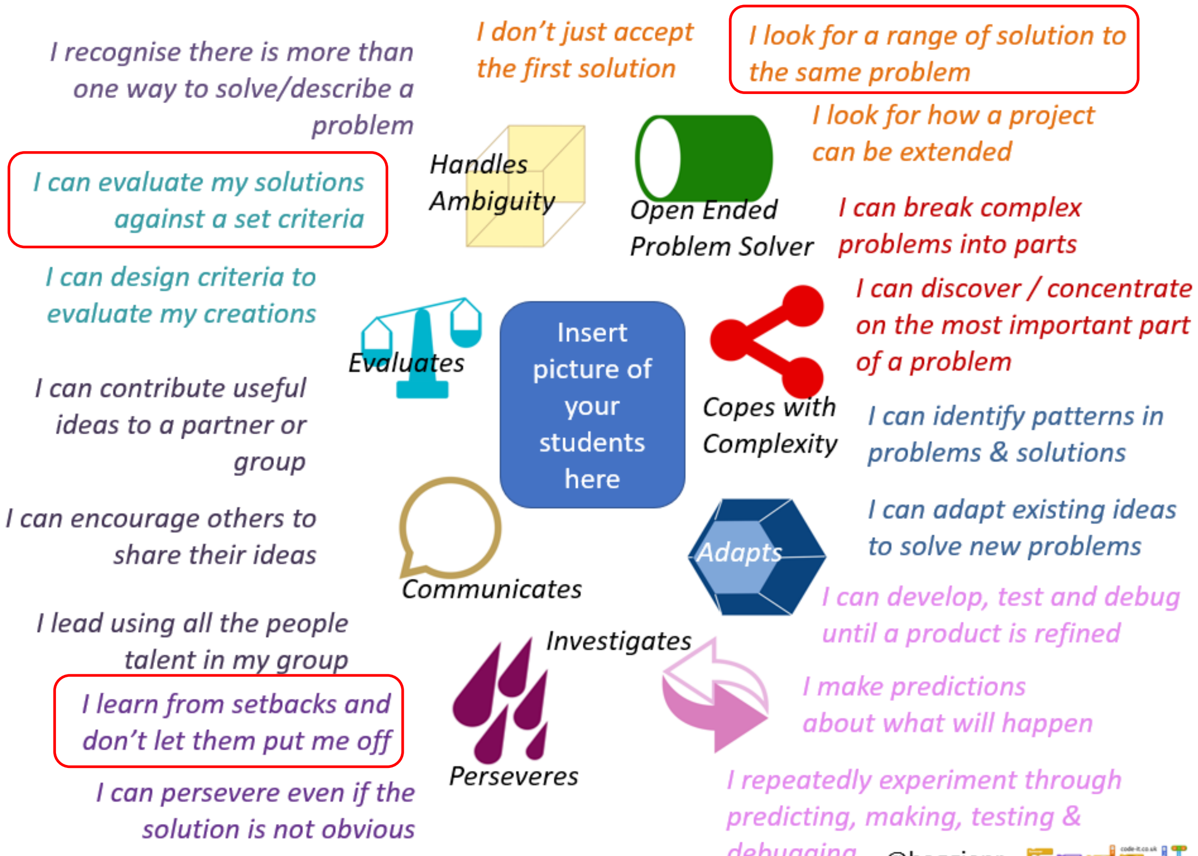
3, In the email title say which module you are doing (ie Animation D)

Research References 📖

[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/



I recognise there is more than one way to solve/describe a problem

I can evaluate my solutions against a set criteria

I can design criteria to evaluate my creations

I can contribute useful ideas to a partner or group

I can encourage others to share their ideas

I lead using all the people talent in my group

I learn from setbacks and don't let them put me off

I can persevere even if the solution is not obvious

I don't just accept the first solution

I look for a range of solution to the same problem

I look for how a project can be extended

I can break complex problems into parts

I can discover / concentrate on the most important part of a problem

I can identify patterns in problems & solutions

I can adapt existing ideas to solve new problems

I can develop, test and debug until a product is refined

I make predictions about what will happen

I repeatedly experiment through predicting, making, testing & debugging

@baggiepr

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