

Captain Meow and the rise of repetition

Can your pupils guide Meow to the colours she loves to play with and learn about how useful repetition is.

Scratch 2.0 Planning
by Phil Bagge
Code-it.co.uk

- 1, Sequence activities away from the computer
- 2, Guiding Meow using a sequence.
- 3, Repetition away from the computer
- 3, Guiding Meow using repetition
- 4, Rules for repetition

Module Aims

Pupils will gain a clear understanding of sequence and simple repeat so many times loops. They will know when and when not to use repeat x times loops.

Computer Science Concepts

- sequence
- Repetition

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**

Prior experience needed

-None although simple projects like Smoking car, Magic Carpet or Conversation give them more confidence in manipulating Scratch code. Ideal for Year 3.

Preparation before the lesson

-Download a copy of Meow or find an easy way of sharing the link to Meow on the Scratch website.

-Print out the copy of the repetition and sequence knowledge sheets for pupils to keep and reference during their primary programming journey.

Approximate Time needed

50-90 Minutes depending on your pupils opening and saving skills and previous Scratch use.

1. Sequence activities away from the computer

Hand out the sequence sheets. Pupils can keep these in their computing folders or books and refer to them as needed in the future. If you use online storage such as Seesaw or Google Docs then you can upload a PDF copy. Use the definition at the top, A set of instructions that follow each other. Either give a few select children three instructions to do in order or get them to give each other a set of three instructions to do. Draw out that one instruction finishes before the next happens.

2. Guiding Meow using a sequence

Open a copy of Captain Meow and show pupils the code blocks under the green flag. Instruct them not to delete the cat or change any of the define blocks or the big block of code with the colours. They are only to use one green flag block and the blocks in the more blocks area. Can they drag out the more blocks and click on them to see what they do? Give them a few minutes to do this before explaining that they need to snap the blocks together in order to navigate Captain Meow to all of the colours one by one. It helps to demo this without any repeat loops. Make sure you press the **main green flag** to start the program,



Don't just click on the script. Otherwise, code such as returning Captain Meow to the start if she goes through the pink walls will not happen.

Formative Assessment

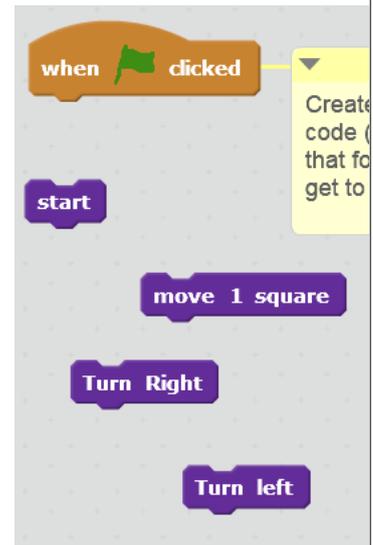
Go round and look to see if pupils are creating a sequence of code. Occasionally some pupils will attach buttons to each block and navigate the maze using these. Or they are just clicking on the separated code blocks to move Captain Meow.



Incorrect Solution using buttons

Corrective intervention

Congratulate them on remembering how to make their own keyboard input buttons, then explain that their solution doesn't use a sequence and that they are only allowed to run the code by clicking on the green flag which will start the sequence one code block at a time. Point to your demo sequence.



What do the code blocks do?



Demo a sequence

3. Repetition away from the computer

Stop pupils after 15-20 minutes and ask them if they have ended up off the page as their script has grown longer and longer? Ask them if they would like to know a way to do the same instructions but use much fewer code blocks?

Give out the repetition sheet for future reference and point them to the top left example. Explain that programmers came up with a great solution for avoiding having to put one code block on top of another by inventing repetition. Code that allows something to be easily repeated. I also talked to my class about Ada Lovelace the first programmer.

Give a pupil an instruction and ask them to repeat it three or four times.

Repeat three times, wave your hands in the air.

Repeat four times, stand up and sit down.

Repeat six times, put your tongue out and in.

Can they give their partners repeat instructions?

4. Guiding Meow using repetition

Demonstrate how many duplicated blocks could be replaced by just two blocks. Make sure you ask if blocks that happen once should be repeated.

Instruct them to either continue their code using repeat loops or start again and remake all the code. I favour the later approach as it clearly shows how repeat loops create shorter code. They could detach their earlier sequence only code and move it to the right so they can compare it later.

Give time to create code using sequence and repetition.

If any pupil is a quick finisher then they can create their own maze using the blank ones inside Stage Backdrops.

Saving for offline Scratch Users on Windows

Don't forget that pupils will need time to **save as** their work to their own personal home areas. You may need to talk this through or provide a cheat sheet as by default Scratch 2.0 likes to save to the desktop.

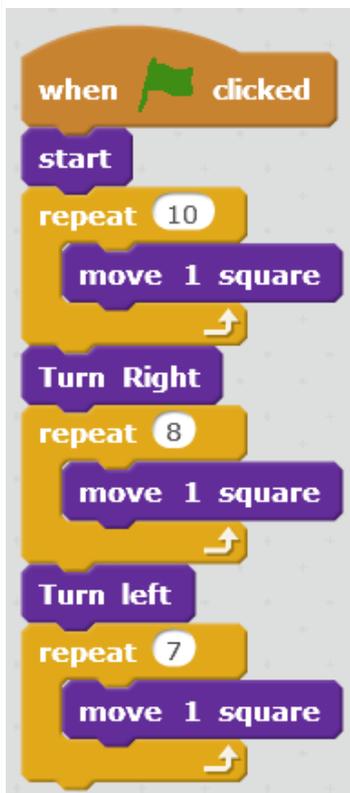
5. Rules for repetition

Ask pupils if they can think of any rules for repetition?

Draw out that if the same code is used more than once then a repeat loop is a very useful programming tool.

Code that is not repeated should not use a repeat loop.

You could draw a repeat loop on the classroom wall and look for everyday repeats to put inside it. Can your class find any more?



Teachers Notes

Please note that the build a block procedures are simplifying the start point, moves and turns so that younger pupils don't get lost in coordinates, degrees or steps that are pixel lengths (dots on the screen). Computing scientists call this abstraction as we are removing complexity and allowing the user to concentrate on what they absolutely need to know.

This might also be a useful project to show pupils who are just starting to learn about simple non adaptable procedures or being introduced to the concept of abstraction.

Assessment

Can pupils use repetition when it is useful in this project and avoid using it when code is not repeated?

Can pupils identify and describe sequence and repetition in their code?

Can pupils spot other forms of everyday repetition?

Can pupils use repetition in their own coding projects?

<http://code-it.co.uk/scratch/actionassess/actionoverview>

Sharing and adapting

You are welcome to share this project and adapt it internally for your school.

If you wish to modify it and republish it yourself please get in touch with the me via twitter or email. I am normally ok with this as long as credit for the original work has been given and there is substantial addition or difference in the adaptation.

Without written consent the following creative commons licence applies.

Phil Bagge 17th May 2018 V1



<http://code-it.co.uk/contact/>