

BWJS COMPUTING

Making a Maths Quiz Year Five

Unit Overview

This unit re-visits Scratch, and the world of programming. It gives children the opportunity to create a game for others to play, using selection and variables.

This unit is based on Phil Bagge's 'Quiz' - <http://www.code-it.co.uk/scratch/scratchquiz.html>

Children will cover:

- Decomposition as a way of analysing how a programme works.
- How to create simple programming blocks that link input to output.
- Making blocks of programming which complement each other.
- Using a variety of inputs to control different outputs.
- Planning, testing and refining algorithms.

Expectations	Curriculum Links	Prior Skills
Children will be able to create a maths quiz which uses selection to determine responses to answers, and variables to construct a score. They will be able to change backgrounds on the stage by using broadcasts.	This is a brand new unit to BWJS. Its most obvious link is to maths, but really, this unit will work most logically as a standalone unit.	Children will have some pre-existing knowledge of how Scratch works. They will be used to adding sprites and backgrounds, and how to create some blocks of programming.
		Software Used <ul style="list-style-type: none">- Scratch.

BWJS COMPUTING

Making a Maths Quiz Year Five

Assessment Strands Relevant to this Unit

Programming & Gaming

Bronze	Silver	Gold
Understand that elements in a game or simulation can be controlled. Control a device or on screen character using simple commands. Programme a sequence of commands for a device or screen character.	Programme a series of commands to achieve a specific outcome. Adapt commands to change the eventual outcome. Programme complementary sequences of commands to control more than one thing. Use repetition in a programme. Use selection in a programme (if/else).	Use variables in a programme. Test and refine a series of commands. Evaluate commands and make changes where necessary.

Making a Maths Quiz Year Five

	Learning Objectives	Key Skills	Concepts	Lesson Content
1	<ul style="list-style-type: none"> - To understand what an algorithm is. - To decompose a simple programme. - To use selection in a programming block. 	<ul style="list-style-type: none"> - Insert a background into a Scratch project. - Insert a new sprite into a Scratch project. - Use a selection block in a Scratch project. 	<ul style="list-style-type: none"> - To know what an algorithm is, and how to put one together. - To know that a programme can be broken into smaller steps. 	<p>Show children Quiz Example 4 (or an equivalent of). Remind children of what is meant by decomposition, and apply this to example. What aspects will we need to programme? (Sprite, background, questions and answers, keeping a score, changing the background). Explain that this is a project we will put together over a run of sessions. Give children a few moments to choose a sprite and a background. Remind children of the link between input and output. Add the green flag block and the welcome message (children should remember this from their work in Year 4). Introduce the concept of selection through the 'if', 'else' block. Give illustrations of selection in the real world: IF it rains, I get wet, ELSE I stay dry. IF I eat, I'll have energy, ELSE I'll stay hungry. Children may well be able to think of some other examples. Bring out the selection block to show how it works. It allows the programme to go down two different paths – in this case, the right answer or the wrong answer. To use the selection block, we need to add an 'ask' block (from the 'sensing' bank). This is very different to the 'say' or 'think' blocks from last year, as they require the user to give an answer. Put a maths question into the ask block. We also need to supply the answer to our question. To do this, you need an 'answer' oval and a '=' operator. Arrange these components so they match the example of the layout. Finally, we need an output for each answer – a 'Correct; or 'Wrong' response for our sprite to say. Drag these into the correct places. Test and debug, giving a right and wrong answer. Ask children why we're doing a maths quiz as opposed to any other type. More able children may be able to recognise that this is because of the black and white nature or calculations – the answer is right or wrong, and there are no shades of grey (bad spelling, irregular capitalisation, etc).</p>

BWJS COMPUTING

2	<ul style="list-style-type: none"> - To use selection to create a series of questions which can be answered. 	<ul style="list-style-type: none"> - Duplicate and adapt blocks of programming. 	<ul style="list-style-type: none"> - That programmers use short cuts – they don't need to programme everything from scratch if they can duplicate programming they have already done. 	<p>Pull up Quiz example 2 (or an equivalent file of your own). How has this project developed since last week? There is now more than one question for the quiz to work through.</p> <p>The rest of these questions could be programmed one step at a time, but there is a much faster way to generate them. Show children how to duplicate a block of code. Then give children plenty of time to create a run of questions. They will need to...</p> <ul style="list-style-type: none"> - Change the question each time - Make sure each question is linked to a correct answer <p>Children may want to vary the response given by the sprite to right or wrong answers for some variety.</p> <p>Remind children to test and debug frequently through each step, testing right and wrong answers.</p>
3	<ul style="list-style-type: none"> - To create a variable within your project 	<ul style="list-style-type: none"> - Add a variable to a scratch project. 	<ul style="list-style-type: none"> - That a computer can keep track of a variable which changes depending on how the programme is used. 	<p>We've spent the last two sessions creating a working quiz. Are there any traditional aspects of a quiz that we don't have in ours? Children should be able to think of adding a score.</p> <p>To introduce a score to our game, we need to use a variable. A variable is something that can change as the programme progresses. The computer will keep track of the variable, adjusting it whenever it is programmed to do so.</p> <p>Open your example quiz from last week (or Example 2 provided). Make a variable and call it 'score'. Show that the variable can be displayed or hidden – make sure children know that it will continue working when it is hidden.</p> <p>Talk about where you add a 'Change score by 1' block. It will need to go somewhere after each question, but where? Discuss options, test a few.</p> <p>Children then need to place these blocks in their own quiz.</p> <p>Play through your quiz. Notice that when you play your quiz a second time, your score will not reset. Demonstrate adding a 'Set score to 0' block at the beginning of your quiz.</p> <p>Test and debug, as usual.</p>
<p>Once children have added a score system to their quiz, they may want to extend this idea by altering the score whenever there is a wrong answer (using a 'Change score by -1' command block, etc. This could be an additional session, or an extension task in session three for those who are able.</p>				

BWJS COMPUTING

4	<ul style="list-style-type: none"> - To create broadcast messages to change a background. 	<ul style="list-style-type: none"> - Switch between backgrounds in a project. - Use broadcasts as a form of input. 	<ul style="list-style-type: none"> - That inputs can be placed within the programme as broadcasts which can be sent and received, controlling other elements within the project. 	<p>Play through the fourth Quiz example file. Children to use their decomposition skills to work out what additional programming has been done for this version of the quiz. Children should notice that a message appears on the background when a question is answered correctly or incorrectly.</p> <p>Open the example quiz you used in the last session (or the third quiz example file). Select the stage and show how to make a copy of the background. Edit the copy, adding a 'well done' message through the paintbrush or text tool. Repeat this to create an equivalent message for a wrong answer. Name each background (including the original) so they can be easily identified.</p> <p>Switch back to your sprite's script area. We're now going to use a new form of input and output – our input will be a broadcast; something which sends a message from one part of the project to another. Create a broadcast block for when an answer is correct and create a new broadcast for it to send (call it 'correct', or 'right', or 'yes' – something easily identifiable. Place the broadcast block inside the 'if – else' selection block. Switch to the Stage's script tab, and add a 'When I receive' block. Choose the correct broadcast from the available options. Then add a 'Switch to background' block from the 'Looks' bank, and choose the background which is relevant. Test the programme. Children will notice that the background will now be stuck in the new version. To reset backgrounds, add a 'Wait' block to the stage script, and a 'Switch to background' block that reverts to normal.</p> <p>Can children repeat this process to switch to your other background after a wrong answer?</p> <p>When achieved, the same commands will need to be added for each question. Test and debug at every stage to make sure the programme works.</p> <p>Depending on how each project is set up, you may need to add one more section of code to the stage scripts – a green flag block linked to a 'Switch to background' block which will reset the background whenever the quiz is started.</p>
5	<ul style="list-style-type: none"> - To programme a suitable ending based on selection and variables. 	<ul style="list-style-type: none"> - Combine variables and selection using an operator. 	<ul style="list-style-type: none"> - To understand that selection can be used to choose between a range of options. 	<p>Show Example 5 of the maths quiz examples. Look at the two possible endings given for the quiz. One is a selection block which chooses between a high score and everything else. One is a series of selection blocks which use the +, < and > signs within operator blocks to give a range of outcomes. Talk through both options and give children the chance to create an ending for their own quiz.</p>

BWJS COMPUTING

As an extension to this final lesson, children may want to send another broadcast message to change the background to a 'game over' screen. Children would need to decide where this broadcast should be placed. Children may also want to 'hide' the sprite when this message is broadcast.

Depending on timing, allow children time to develop their own version of the game they have made. What kind of changes could they make?

- The character and background.
- The type of questions – text based, or multiple choice.
- The scoring system.

Or, indeed, any more creative ideas the children may have.