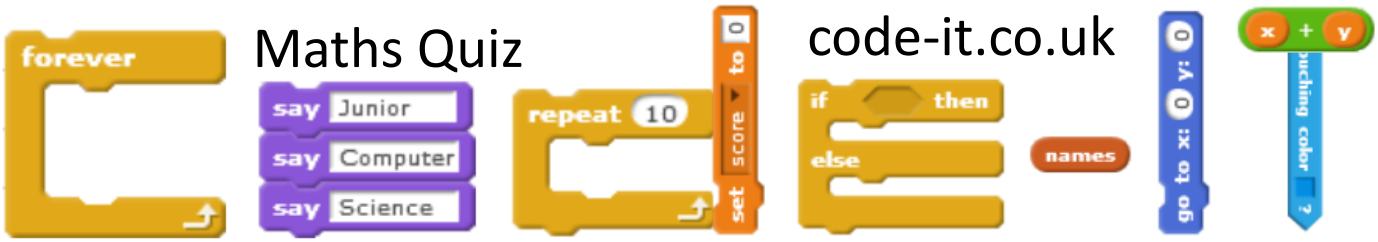


Computer Science Concepts

-if else selection
-Broadcasting to trigger other blocks
-Variable for score
-Algorithm to code
-User typed input into program

Program Aim Create a Maths quiz

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**

Maths Concepts -Equals means the same as

Extension

4a, Explain how selection works using comment

6a, Take away score if get wrong answer

6d, Questions that use < or >
6e, Timer
6f, Congratulations
6g, Repeat until answer correct

1, Think Through Quiz
2, Welcome
4, Selection
5, Multiple Questions
6, Score Variable

Learning Path

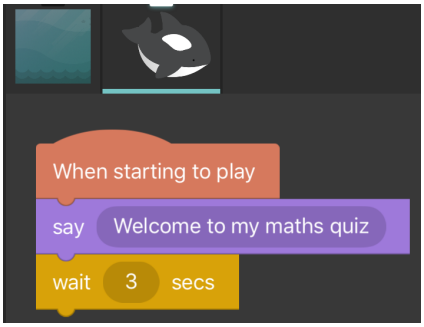
3, User input ask block

1, Think through steps (algorithm) to ask a quiz question

Ask a simple maths quiz question to the class. Get them in pairs to work out the steps to ask that question and record them on a whiteboard. Make sure they realise that this is ***nothing to do with code*** and it is what a humans would do if asking this question as part of a quiz. Once they have completed this draw out that the quiz master would need to think of the question and then think of the answer before they can ask the question. Once the user has answered the question ask them how they will know if the question is right or wrong. Indicate that the quiz master has got the original thought through answer and the answer the quiz user gave them. Pretending to hold both in either hand and look alternately at both like you are examining them helps pupils. You could write **MY QUIZ ANSWER** on one piece of paper and **USER ANSWER** on another. You are looking for a pupil to indicate that they need to be the same. Often pupils will come up with the idea of comparing the answers which is a step in the right direction. Push pupils to be more specific and ask for a type of comparison. If a pupil says same you can then ask what the maths symbol for the same is (=). Then go through http://code-it.co.uk/wp-content/uploads/2015/05/quiz.pdf which highlights the process graphically.

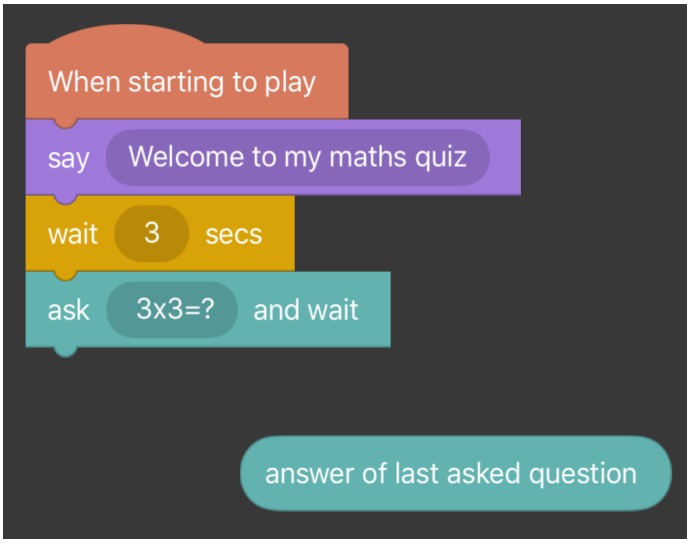
2, Welcome

Explain that this program will mostly be one sequence of instructions from start to finish. Can pupils choose a good starting block and a way to welcome users to their quiz? Have they tested their code?



Latest version of tickle will speak the say blocks.

For those converting from Scratch you need to set the length of the say block using a wait



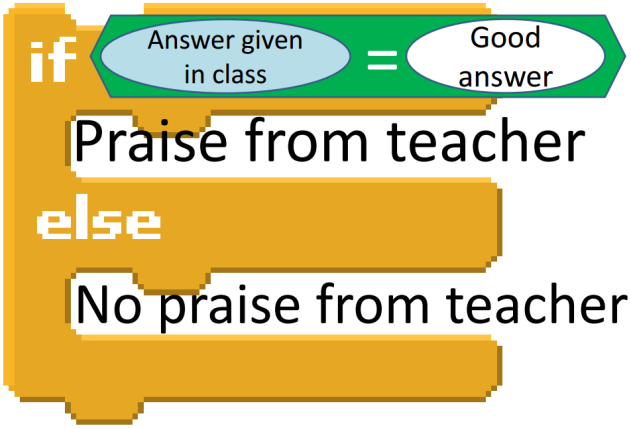
3, User Input Ask Block

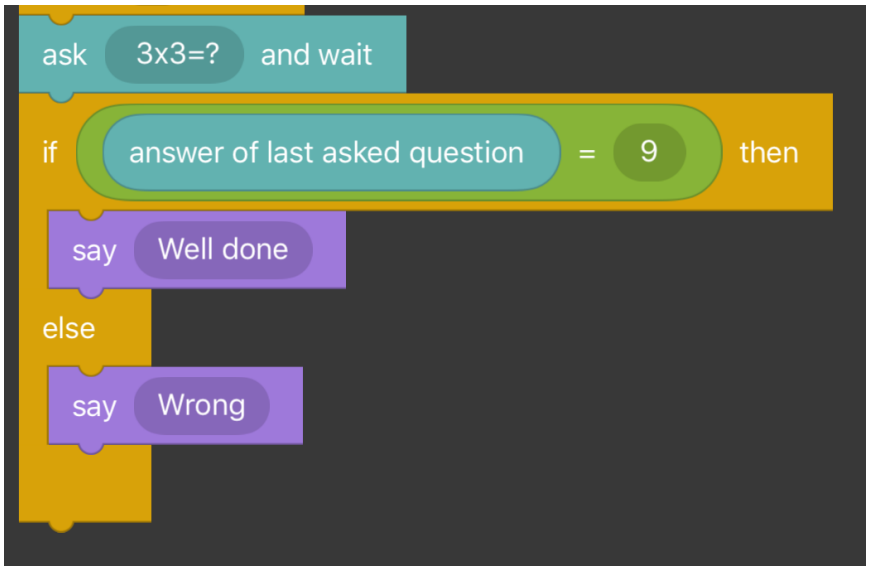
Drag the *ask* and the *answer of last question* blocks out. Explain that this is a user **input** block. It **puts in** information from the person taking the quiz. Explain that whatever is typed into the ask input block goes inside the **answer of last asked question** block.

Type some rubbish in such as 'fishfinger' ask the class where it is now. Ham this up with some bad overacting! Just make sure they all get where the inputted information goes.

4, Selection

If this is the first time that pupils have met the idea of selection it is worth using some real world examples which you can find at http://code-it.co.uk/wp-content/uploads/2015/05/selectioninrealworld.pdf This last examples is important as it paves the way for the code. Make sure in the last example you draw out that equals means the same as.





Drag out these blocks and arrange them like this. Make sure you draw attention to the shape especially when inserting the equals block. Notice that the equals block goes inside the if else block and then the **answer of last question** block goes inside the left hand side of the equals block.

Talk through the program line by line before running it to help pupils understand what is happening.

Pretend to be a user who gets an answer right then. Pretend to type the answer and ask the pupils where the answer has gone in in the code? (the light green **answer of last question** block)

The program then checks to see if it is the same (point to equals sign) as the right answer this means that the condition has been met and only the top correct block is run. Do the same for a wrong answer pointing out that the answer is not the same as the right answer so only the else block is run.

This work is licensed under the Creative Commons Attribution-NonCommercial 3.0 Unported License. To view a copy of this license, visit http://creativecommons.org/licenses/by-nc/3.0/ or send a letter to Creative Commons, 444 Castro Street, Suite 900, Mountain View, California, 94041, USA.

5, Multiple Questions

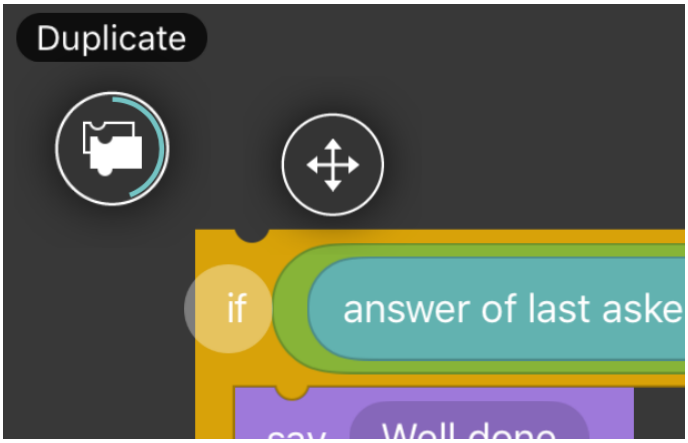
Ask pupils what three blocks we wouldn't need to create when adding a new question? (Top 3)

Don't show pupils how to duplicate the blocks until they have done a questions block by block.

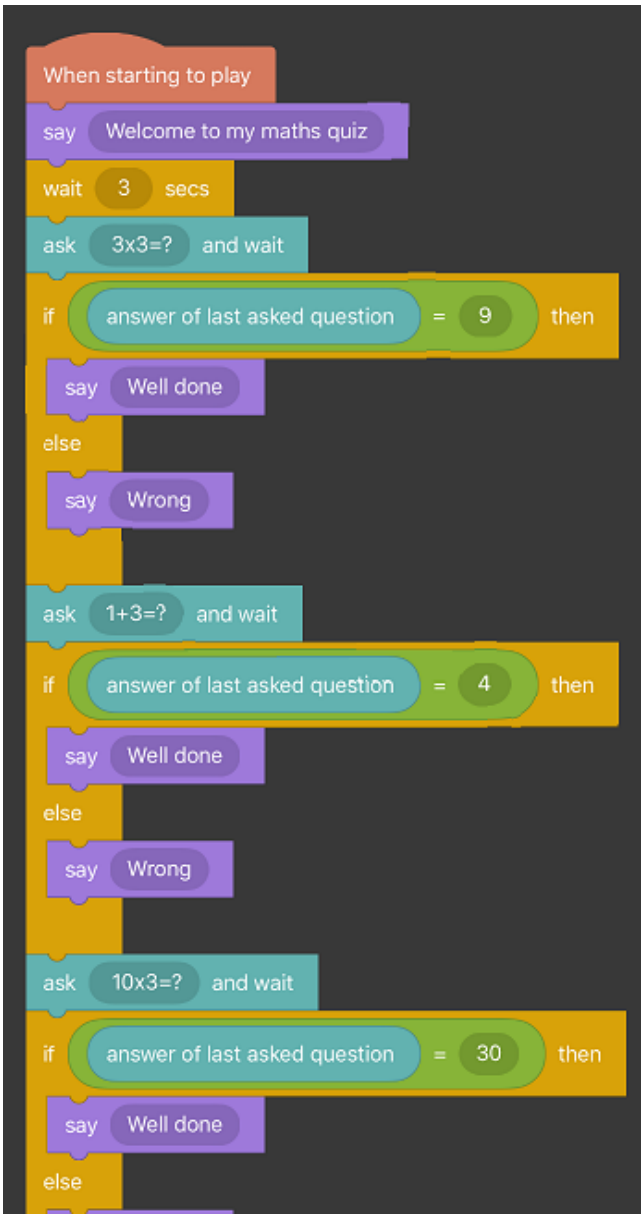
Now demonstrate how to duplicate blocks by dragging out a single question and holding your finger on the top most block and sliding up to select duplicate.

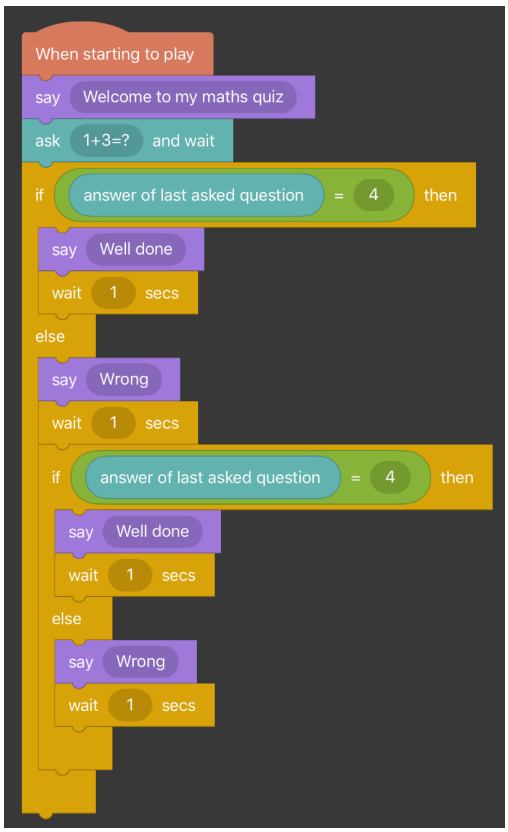
Pupils may need to pinch out to create more space.

Hold finger on top block to duplicate



Slide up to select duplicate until the circle is complete





Common Bugs

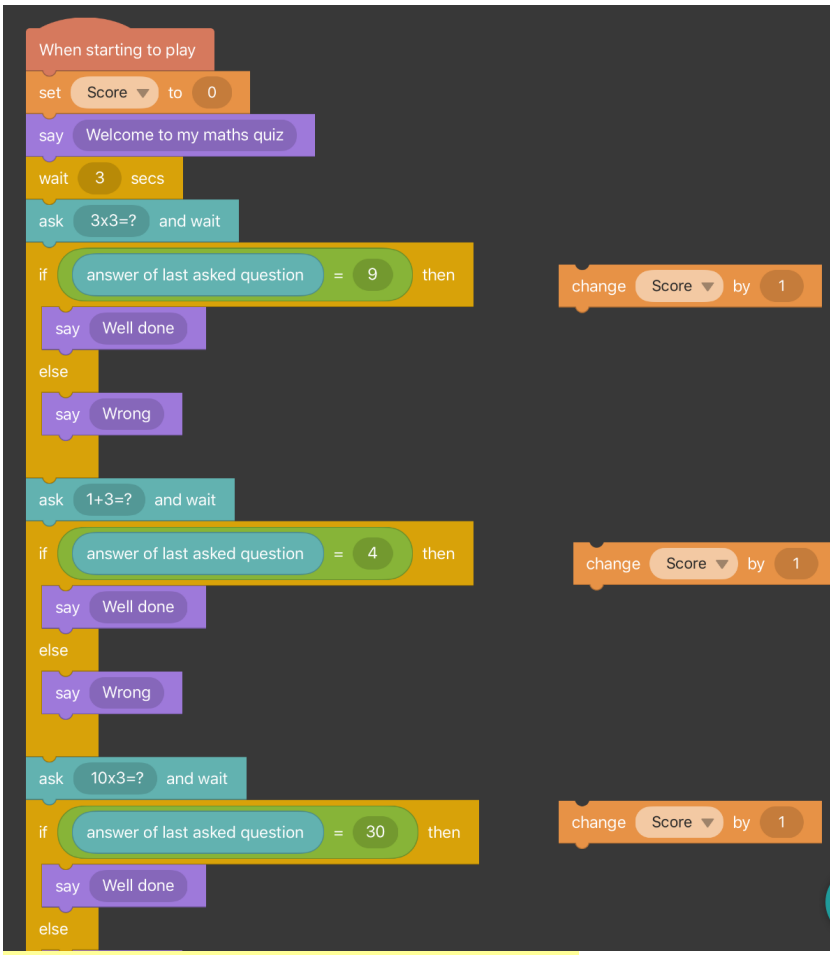
Once pupils have started to duplicate it is very easy to drag the question inside the bottom else part of the selection. Can they tell you when the second question would be asked? ANSWER Only if you got question 1 wrong.

If done on purpose this could be a device to allow an easier question to be asked if the first question was answered incorrectly.

6, Score Variable

Ask pupils what it means if we say that the weather is variable? Draw out that it means changeable. Explain that we can get the program to keep score for us by creating a variable which is like a pencil pot which we can get the computer to put pencils/ numbers into to help us keep score when someone gets the answers right.

Ask pupils what possible scores the user could get from a three question quiz if each question is worth one mark each? (0,1,2 or 3) Ask them if we know what score a user will get before they take the quiz? This is important as many pupils won't have considered the range of possibilities and may have already awarded a *'mark'* via a say command at the end of each question.



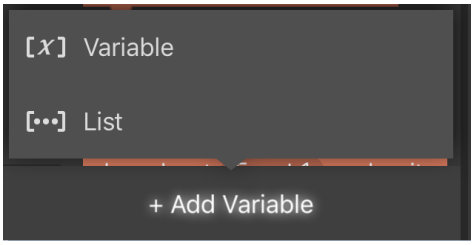
Common Bugs

Inserting **set score to 1** blocks instead of **change score by 1**. This sets the score back to 1 every time it is used.

Missing out a **set score to 0** block

This means that when the quiz is run the second time the old score is still inside the variable and will be added to instead of having an empty score when you start.

Create a variable called score



Click on the + add variable button (bottom left)

Select variable

Type in a name

Ask pupils what score the user should start with? Drag out the **set score to 0** block and ask pupils where it should go? (At the top so the quiz always starts with zero points)

Now drag out multiple **change score by 1** blocks and ask pupils to decide where they should go to put a point into the variable pot if the user gets the question correct?

Give them time to discover the correct location themselves and test it.

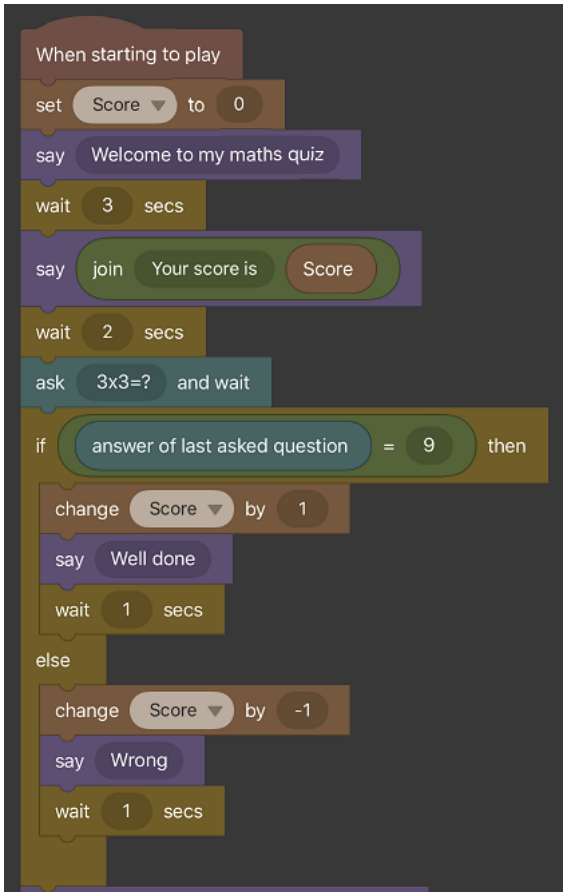
When pupils have come up with the right location talk through the code to explain it before running it. A great way to do this is with a box for the variable containing 4 pencils labelled with score that a pupil holds. Model setting the score to 0 by removing all the pencils. Model adding a pencil into the box if you get the answer right.

Finally run the code getting some answers right and some wrong.

6a Takeaway score point if user gets answer wrong

Challenge pupils to takeaway a point if the user gets the question wrong. They do this by adding a change score by –1 into the else part of the if else selection block.

You may also wish to show pupils how you can get the sprite to say the score through the use of a say, join and score variable. Pupils love the cleverness of this.



When tickle code runs it highlights the block which is running

Here the code has all finished

6d, Code that uses less than or greater than

Can pupils create questions such as type a number less than 56. Use a > or < block instead of an = one

6e, Timer

Can pupils create a timer for the quiz which ends the game after a set period of time. This could be further extended so that the user could select a difficulty level which results in a longer or shorter time period. Pupils could use simple wait for x seconds time blocks although this will not show the user the time counting down or a time variable inside a loop which is the better solution.

6f, Congratulations

Add code at the end of the quiz that congratulates the player IF they got all of the questions correct ELSE it tells them their score. Pupils would need to use an if else selection blocks.

6g, Repeat until answer correct

Can pupils get a question to repeat until the user answers it correctly? Ask if this solution would affect a scoring variable? The repeat until loop is useful for this.

Cross Curricular Opportunities

-Maths

Pupils create a quiz for other members of their set or maths group based on recent maths topics. Pupils directed to create questions of a specific type geared to the needs of their immediate Maths curriculum.

-Other areas of the curriculum

Pupils instructed to create a quiz around a specific topic such as Roman or Neolithic Britain, light and sound, keeping safe on the web or sacred texts from the great world religions.

For lots more great free programming planning why not subscribe to code-it.co.uk for free

For lots of great Scratch planning and advice on how to teach programming purchase How to teach primary programming using Scratch

If you use Twitter why not follow Phil Bagge @baggiepr who can also be contacted for advice on computing excellence in your school.

This work is licensed under the Creative Commons Attribution-NonCommercial 3.0 Unported License. To view a copy of this license, visit http://creativecommons.org/licenses/by-nc/3.0/ or send a letter to Creative Commons, 444 Castro Street, Suite 900, Mountain View, California, 94041, USA.