

Seamer and Irton CP School – Computing (H.Griffiths)

Topic – Selection in quizzes

Year 5 – Summer 2

Strand – Programming

Prior Learning

In Year 4 – Summer 1 – Repetition in shape

Learners explored the concept of repetition in programming using the Scratch environment. Learners looked at the difference between count-controlled and infinite loops and used their knowledge to modify existing animations and games using repetition. Their final project was to design and create a game which used repetition, applying stages of programming design throughout.

Through previous programming units, learners will have prior experience of programming using block-based construction (eg Scratch) and understand the concepts of ‘sequence’ and ‘repetition’, and have some experience of using ‘selection’

Key Knowledge I need to understand

I need to understand that:

Programming is when we make a set of instructions for computers to follow.

Scratch is a program that we can use in order to code our own quizzes, stories, animations and games.

We can input questions using the ‘ask’ command blocks.

We can use selections and conditions in order to ensure that there are different outcomes depending upon a user’s response.

We use algorithms (a set of instructions to perform a task) to sequence movements, actions and sounds in order to program effective animations.

Learners will develop their knowledge of ‘selection’ by revisiting how ‘conditions’ can be used in programming, and then learning how the ‘if... then... else...’ structure can be used to select different outcomes depending on whether a condition is ‘true’ or ‘false’. They represent this understanding in algorithms, and then by constructing programs in the Scratch programming environment. They learn how to write programs that ask questions and use selection to control the outcomes based on the answers given. They use this knowledge to design a quiz in response to a given task and implement it as a program. To conclude the unit, learners evaluate their program by identifying how it meets the requirements of the task, the ways they have improved it, and further ways it could be improved.

How I will show what I have learned

To explain how selection is used in computer programs

- I can recall how conditions are used in selection
- I can identify conditions in a program
- I can modify a condition in a program

To relate that a conditional statement connects a condition to an outcome

- I can use selection in an infinite loop to check a condition
- I can identify the condition and outcomes in an ‘if... then... else...’ statement
- I can create a program that uses selection to produce different outcomes

To explain how selection directs the flow of a program

- I can explain that program flow can branch according to a condition
- I can design the flow of a program that contains ‘if... then... else...’
- I can show that a condition can direct program flow in one of two ways

To design a program which uses selection

- I can outline a given task
- I can use a design format to outline my project
- I can identify the outcome of user input in an algorithm

To create a program which uses selection

- I can implement my algorithm to create the first section of my program
- I can test my program
- I can share my program with others

To evaluate my program

- I can identify ways the program could be improved
- I can identify the setup code I need in my program
- I can extend my program further

What vocabulary I need to know

Selection, condition, true, false, count-controlled loop, outcomes, conditional statement (the linking together of a condition and outcomes), algorithm, program, debug, question, answer, task, design, input, selection, implement, test, run, share, evaluate, constructive

The following Glossary may be useful

<https://icompute-uk.com/ewExternalFiles/iCompute-Glossary.pdf>

What’s next

In Year 6 - Variables in Games – Spring 1 This unit explores the concept of variables in programming through games in Scratch. First, learners find out what variables are and relate them to real-world examples of values that can be set and changed. Then they use variables to create a simulation of a scoreboard. In Lessons 2, 3, and 5, which follow the Use-Modify-Create model, learners experiment with variables in an existing project, then modify them, before they create their own project. In Lesson 4, learners focus on design. Finally, in Lesson 6, learners apply their knowledge of variables and design to improve their games in Scratch

Assessment

National Curriculum Computing links

- 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
- select, use and combine a variety of software (including internet services) on a range of digital devices to design and create a range of programs, systems and content that accomplish given goals, including collecting, analysing, evaluating and presenting data and information

Assessment

Formative assessment opportunities are highlighted in each of the lesson plan documents. The learning objective and success criteria will be introduced at the beginning of each lesson and then reviewed at the end. Learners should assess how well they feel they have met the learning objective using the teacher's chosen method.

Summative assessment document included - multiple choice questions. This should be used, alongside teacher judgement, to complete summative assessment on ScholarPack

<https://teachcomputing.org/curriculum/key-stage-2/programming-b-selection-in-quizzes>

Teacher Subject Knowledge

Please use a teacher accounts in Scratch to help with assessment throughout this unit. For guidance on setting up teacher accounts, please [visit the Scratch website](http://scratch.mit.edu/educators/faq) (scratch.mit.edu/educators/faq).

This unit focuses on developing learners' understanding of selection in an on-screen context. It highlights what 'conditions' are and how they are used as part of 'selection'. This unit also develops learners' understanding of design in programming, using the approach outlined below.

Levels of abstraction

When programming, there are four levels which can help describe a project (known as levels of abstraction). Research suggests that this structure can support learners in understanding how to create a program and how it works:

- Task - this is what is needed
- Design - this is what it should do
- Code - this is how it is done
- Running the code - this is what it does

Spending time at the 'Task' and 'Design' levels before engaging in code-writing aids learners in assessing the 'do-ability' of their programs and reduces a learner's cognitive load during programming. Learners will move between the different levels throughout the unit and this is highlighted within each lesson plan.

Conditions

'Conditions' are statements that need to be met for a set of actions to be carried out. They can be used in algorithms and programs to control the flow of actions. When a condition is met it is referred to as 'true' and when it is not met it is referred to as 'false'. You need to be able to identify and use conditions in algorithms in the form of statements to both start and stop sets of action. Additionally, you need to understand that conditions can be used in loops, and when they are, that the set of actions in the loop will be carried out repeatedly until the condition is true. For example, 'until button 'A' is pressed'.

Selection

When designing programs, there are often points where a decision must be made. These decisions are known as 'selection', and are commonly implemented in programming using 'if' statements. Selection is used to control the flow of actions in algorithms and programs by checking whether a condition (see above) has been met. If it has been met, the identified actions will be carried out. When selection is used in programs, infinite loops (see above) are often used to instruct the device to check the condition repeatedly. Without using loops, the condition would only be checked once following the sequence of the code.