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

Topic – Repetition in Games	Year 4 – Summer 2	Strand – Programming

Prior Learning

In Year 4 – Spring 1 - Repetition in Shapes learners created programs by planning, modifying, and testing commands to create shapes and patterns. They used Logo, a text-based programming language.
Pupils have prior experience of coding in KS1 (Floor robot and ScratchJR) and in KS2 using Scratch to develop their coding in a progressive manor having completed the following previous units
Year 3 – Spring 1 - Sequence in Music
Year 3 – Summer 2 - Event and Actions

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 stories, animations and games.

We can use repeat and loop operator blocks in order to make our programs more logical and efficient. These help to run code continuously or for a set number of times.

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

In day-to-day life, we use many patterns of repetition. This may include things like; brushing your teeth, performing a dance routine, creating a piece of music, finding a clapping rhythm.

Learners will explore the concept of repetition in programming using the Scratch environment. The unit begins with a Scratch activity similar to that carried out in Logo in Programming unit A, where learners can discover similarities between two environments. Learners look at the difference between count-controlled and infinite loops, and use their knowledge to modify existing animations and games using repetition. Their final project is to design and create a game which uses repetition, applying stages of programming design throughout.

How I will show what I have learned

To develop the use of count-controlled loops in a different programming environment	- I can list an everyday task as a set of instructions including repetition - I can predict the outcome of a snippet of code - I can modify a snippet of code to create a given outcome
To explain that in programming there are infinite loops and count controlled loops	- I can modify loops to produce a given outcome - I can choose when to use a count-controlled and an infinite loop - I can recognise that some programming languages enable more than one process to be run at once
To develop a design that includes two or more loops which run at the same time	- I can choose which action will be repeated for each object - I can explain what the outcome of the repeated action should be - I can evaluate the effectiveness of the repeated sequences used in my program
To modify an infinite loop in a given program	- I can identify which parts of a loop can be changed - I can explain the effect of my changes - I can re-use existing code snippets on new sprites
To design a project that includes repetition	- I can evaluate the use of repetition in a project - I can select key parts of a given project to use in my own design - I can develop my own design explaining what my project will do
To create a project that includes repetition	- I can refine the algorithm in my design - I can build a program that follows my design - I can evaluate the steps I followed when building my project

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 5 – Spring 1 – Selection in quizzes** learners 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 using 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.

Assessment

Completed in ScholarPack against the following National Curriculum 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 provided throughout each of the lesson plan documents. The learning objectives and success criteria are introduced in the slide decks at the beginning of each lesson and then reviewed at the end.

Summative assessment – the assessment rubric document should be used to assess student's work. The rubric should be completed digitally and stored in individual pupil folders and then used alongside teacher judgement to complete ScholarPack
https://teachcomputing.org/curriculum/key-stage-2/programming-b-repetition-in-games

Teacher Subject Knowledge

It is recommended that learners use desktop or laptop computers to access Scratch (scratch.mit.edu). Please set up a teacher account in Scratch to make it easier to manage student accounts. For guidance on setting up teacher accounts, please visit the Scratch website. (https://scratch.mit.edu/educators/faq)

This unit focuses on developing learners' understanding of repetition within the Scratch programming environment. Repetition is where actions or commands in programming are repeated. The repeating commands can also be referred to as a 'loop'. Loops can be repeated indefinitely (known as 'infinite loops'), or for a set number of times (known as 'count-controlled loops'). This unit also develops learners' understanding of design in programming, using the approach outlined below.
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 - what is needed
- Design - what it should do
- Code - how it is done
- Running the code - what it does

Spending time at the 'task' and 'design' levels before engaging in code-writing can aid learners in assessing the 'do-ability' of their programs. It also reduces a learner's cognitive load during programming.