Medium Term Plan: Supporting Implementation of LTP/Progression Grid

Subject: Computing – Programming: Repetition in Shapes Year: LKS2 – Year B – Autumn

NC/PoS:
- 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

Prior Learning (what pupils already know and can do)
Understanding giving and following instructions, using floor robots to create and debug programs, creating a sequence of commands to follow a routed, using Scratch Jnr to create a program using blocks, how to add music and link to motion, how to use the pen tool within programmes

End Points (what pupils MUST know and remember)
- To identify that accuracy in programming is important
- To create a program in a text-based language
- To explain what 'repeat' means
- To modify a count-controlled loop to produce a given outcome
- To decompose a task into small steps
- To create a program that uses count-controlled loops to produce a given outcome

Key Vocabulary: screen-turtle, command box, pixels, clear screen, angles/degrees, algorithm, design, debug, pattern, repeat, repetition, count-controlled loop, value, decompose, procedure, debug

Recommended Resources:
https://tinyurl.com/lks2-repetitioninshapes

Session 1: Turtle Academy

What is meant by a 'text-based programming language'? How can we control the 'screen-turtle'? What is the purpose of the command box? What is needed to move the turtle forward and backward? How can we tell the turtle how far we want it to go? What are 'pixels'? Why do we need to use 'clear screen'? What commands are needed to turn the turtle left and right? How can we combine commands?

Vocabulary: Logo, screen-turtle, command box, pixels, clear screen, degrees/angles

Session 2: Drawing

How could we use the turtle to draw our initials? What angles will need to be used? How do we create a gap between the letters? How can we use the 'home' command? Why is the order of commands important? How can check our algorithm is correct?

Vocabulary: algorithm, design, debug,

Session 3: Repeated Patterns

How can we simply our programmes? When creating patterns, or drawing shapes, is part of the programme repetitive? How can we use the 'repeat' command? What does the value after the repeat command tell us? Why do we need to put our direction commands in square brackets? Can we predict what shape will be made from a given algorithm? Can we use these algorithms to create our own? What happens if the value of the repeat command is changed?

Medium Term Plan: Supporting Implementation of LTP/Progression Grid

Vocabulary: pattern, repeat, repetition, count-controlled loop, value
Session 4: Creating Procedures What is a procedure? Why might they be needed? Why do each of our procedures need to be given different names? Would the procedure for a square be the same as that for triangle? Why do we need to consider angles when creating our procedures? How can we use procedures within our programmes? Can we use procedures to create our own designs? How can we check our programmes work? Vocabulary: repeat, count-controlled loop, decompose, procedure, debug
Future learning this content supports: The content of this unit will support other units on programming and creating effective algorithms.