**Elworth CE Primary School**

**LKS2 Programming Skills Progression**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Y3 (A)** | **Y3 (B)** | **Y4©** | **Y4 (D)** |
| **LKS2 Progression** | To explore a new programming environment* I can identify the objects in a Scratch project (sprites, backdrops)
* I can explain that objects in Scratch have attributes (linked to)
* I can recognise that commands in Scratch are represented as blocks
 | To explain how a sprite moves in an existing project* I can explain the relationship between an event and an action
* I can choose which keys to use for actions and explain my choices
* I can identify a way to improve a program
 | To identify that accuracy in programming is important* I can program a computer by typing commands
* I can explain the effect of changing a value of a command
* I can create a code snippet for a given purpose
 | 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
 |
| I can identify that each sprite is controlled by the commands I choose* I can choose a word which describes an on-screen action for my design
* I can create a program following a design
 | To create a program to move a sprite in four directions* I can choose a character for my project
* I can choose a suitable size for a character in a maze
* I can program movement
 | To create a program in a text-based language* I can use a template to create a design for my program
* I can write an algorithm to produce a given outcome
* I can test my algorithm in a text-based language
 | 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 explain that a program has a start* I can start a program in different ways
* I can create a sequence of connected commands
* I can explain that the objects in my project will respond exactly to the code
 | To adapt a program to a new context* I can use a programming extension
* I can consider the real-world when making design choices
* I can choose blocks to set up my program
 | To explain what ‘repeat’ means* I can identify everyday tasks that include repetition as part of a sequence, eg brushing teeth, dance moves
* I can identify patterns in a sequence, eg ‘step 3 times’ means the same as ‘step, step, step’
* I can use a count-controlled loop to produce a given outcome
 | To develop a design which 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 recognise that a sequence of commands can have an order* I can explain what a sequence is
* I can combine sound commands
* I can order notes into a sequence
 | To develop my program by adding features* I can identify additional features (from a given set of blocks)
* I can choose suitable keys to turn on additional features
* I can build more sequences of commands to make my design work
 | To modify a count-controlled loop to produce a given outcome* I can identify the effect of changing the number of times a task is repeated
* I can predict the outcome of a program containing a count-controlled loop
* I can choose which values to change in a loop
 | 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 change the appearance of my project* I can build a sequence of commands
* I can decide the actions for each sprite in a program
* I can make design choices for my artwork
 | To identify and fix bugs in a program* I can test a program against a given design
* I can match a piece of code to an outcome
* I can modify a program using a design
 | To decompose a program into parts* I can identify ‘chunks’ of actions in the real world
* I can use a procedure in a program
* I can explain that a computer can repeatedly call a procedure
 | 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 from a task description* I can identify and name the objects I will need for a project
* I can relate a task description to a design
* I can implement my algorithm as code
 | To design and create a maze based challenge* I can make design choices and justify them
* I can implement my design
* I can evaluate my project
 | To create a program that uses count-controlled loops to produce a given outcome* I can design a program that includes count-controlled loops
* I can make use of my design to write a program
* I can develop my program by debugging it
 | 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
 |