



ALMOND HILL JUNIOR SCHOOL MEDIUM TERM PLAN

TOPIC TITLE/SUBJECT: Computing - Programming B – Repetition in games

YEAR GROUP: 4

TERM : Summer

<p style="text-align: center;">Vocabulary</p> <p>Algorithm Sprite Program Backdrops Code Block code Count controlled loops Outcomes Infinite loops</p>	<p style="text-align: center;">Skills</p> <p>-To develop the use of count-controlled loops in a different programming environment (list everyday tasks as instructions with repetition/predict outcomes from snippets of code/modify code snippets to create given outcomes) -To explain that in programming there are infinite loops and count-controlled loops (modify loops to produce given outcome/choose when to use count-controlled and infinite loops/recognise programming languages that 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 (choose which action will be repeated for which object/explain what the outcome of a repeated action will be/evaluate effectiveness of repeated sequences used in programs) -To modify an infinite loop in a given program (identify which parts of loops can be changed/explain effects of any changes/re-use existing code snippets) -To design a project that includes repetition (evaluate the use of repetition on a project/select key parts of a project to include in own design/develop designs and explain what they will do) -To create a project that includes repetition (refine algorithms/build programs that match designs/evaluate the steps)</p>	<p style="text-align: center;">What we already know</p> <p>This unit assumes that learners will have some prior experience of programming. The KS1 -floor robots and ScratchJr, and Scratch is introduced in the Year 3 programming units. Year 4 Programming unit A – using repetition in shapes.</p>
<p>Application/Outcomes</p> <p><u>1 Using loops to create shapes:</u> Look at real-life examples of repetition and identify which parts of instructions are repeated. Use Scratch, a block-based programming environment, to create shapes using count-controlled loops. Consider what the different values in each loop signify, then use existing code to modify and create new code, and work on reading code and predicting what the output will be once the code is run.</p> <p><u>2 Different loops:</u> Look at different types of loops: infinite loops and count-controlled loops. Practise using these within Scratch and think about which might be more suitable for different purposes.</p> <p><u>3 Animate your name:</u> Create designs for an animation of the letters in their names. The animation uses repetition to change the costume (appearance) of the sprite. The letter sprites will all animate together when the event block (green flag) is clicked. When they have designed their animations, the learners will program them in Scratch. After programming, learners then evaluate their work, considering how effectively they used repetition in their code.</p> <p><u>4 Modifying a game:</u> Look at an existing game and match parts of the game with the design. Make changes to a sprite in the existing game to match the design. Look at a completed design and implement the remaining changes in the Scratch game. Add a sprite, re-use and modify code blocks within loops, and explain the changes made.</p> <p><u>5 Designing a game:</u> Look at a model project that uses repetition. Design their own games based on the model project, producing designs and algorithms for sprites in the game. Share these designs with a partner and have time to make any changes to their design as required.</p> <p><u>6 Creating your games:</u> Learners build their games, using the designs they created in Lesson 5. Follow their algorithms, fix mistakes, and refine designs in their work as they build. Evaluate their work once it is completed and showcase their games at the end.</p>		
<p>Other/Cross Curricular Links</p> <p>Maths – Shape, angles Art- Designing aesthetic images DT-Plan, make, evaluate</p>		<p>Adaptation for SEND</p> <ul style="list-style-type: none"> ● Adapted tasks ● Adapted resources ● Additional support