# Year 4 — Repetition in games

| Lesson | Brief overview | Learning objectives |
|---|---|---|
| 1 Using loops to create shapes | In the first lesson, learners look at real-life examples of repetition, and identify which parts of instructions are repeated. Learners then use Scratch, a block-based programming environment, to create shapes using count-controlled loops. They 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. | To develop the use of count-controlled loops in a different programming environment<br>• I can list an everyday task as a set of instructions including repetition<br>• I can predict the outcome of a snippet of code<br>• I can modify a snippet of code to create a given outcome |
| 2 Different loops | In this lesson, learners look at different types of loops: infinite loops and count-controlled loops. They practise using these within Scratch and think about which might be more suitable for different purposes. | To explain that in programming there are infinite loops and count-controlled loops<br>• I can modify loops to produce a given outcome<br>• I can choose when to use a count-controlled and an infinite loop<br>• I can recognise that some programming languages enable more than one process to be run at once |
| 3 Animate your name | In this lesson, learners 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 | To develop a design that includes two or more loops which run at the same time<br>• I can choose which action will be repeated for each object |

| | | |
|---|---|---|
| | 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. | • I can explain what the outcome of the repeated action should be<br>• I can evaluate the effectiveness of the repeated sequences used in my program |
| 4 Modifying a game | In this lesson, learners look at an existing game and match parts of the game with the design. They make changes to a sprite in the existing game to match the design. They then look at a completed design, and implement the remaining changes in the Scratch game. They add a sprite, re-use and modify code blocks within loops, and explain the changes made. | To modify an infinite loop in a given program<br>• I can identify which parts of a loop can be changed<br>• I can explain the effect of my changes<br>• I can re-use existing code snippets on new sprites |
| 5 Designing a game | In this lesson, learners look at a model project that uses repetition. They then design their own games based on the model project, producing designs and algorithms for sprites in the game. They share these designs with a partner and have time to make any changes to their design as required. | To design a project that includes repetition<br>• I can evaluate the use of repetition in a project<br>• I can select key parts of a given project to use in my own design<br>• I can develop my own design explaining what my project will do |
| 6 Creating your games | In this lesson, learners build their games, using the designs they created in Lesson 5. They follow their algorithms, fix mistakes, and refine designs in their work as they build. They evaluate their work once it is completed, and showcase their games at the end. | To create a project that includes repetition<br>• I can refine the algorithm in my design<br>• I can build a program that follows my design<br>• I can evaluate the steps I followed when building my project |