**Year 6 Computing Curriculum – Summer Term 2**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Theme: Variables in games | | | | | | | | |
| **Curriculum objectives** | | | **Vocabulary** | | | | | **Links across the curriculum** |
| - 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 | | | **Keyword** | Definition | sequences | a pattern or process in which one thing follows another. | | [**National curriculum links**](https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study/national-curriculum-in-england-computing-programmes-of-study)   * 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 | |
| design | to think up and plan out in the mind | decompose | Break down into smaller and manageable ‘chunks’ | |
| Count-controlled loop | A count-controlled loop is a form of repetition in which a set of commands are carried out a specific number of times. | Condition-controlled loop | A condition-controlled loop is a form of repetition in which a set of commands stop being carried out when a condition is met. | |
| debug | to fix | conditions | Conditions are statements that need to be met for a set of actions to be carried out | |
| commands | to order or instruct | program | a plan of what will be done | |
| Infinite loop | An infinite loop is a loop that commands the instruction/set of instructions to repeat forever. | algorithms | a determined and finite procedure for solving a problem | |
| **Prior Knowledge:**  EYFS – To follow two step instructions. Year 1 – Commands for a robot. Year 2 – plan and debug algorithm  Year 3 - Sequencing Sounds Year 4 – repetition in Shapes  Year 5 - Selection In physical computing | | | | | **Future Knowledge:**  Year 6 - Sensing | | | |
| **Lesson Sequence** | | **Key Knowledge** | | | | | **Key Skills** | |
| 1 Introducing variables | | * Learners are introduced to variables. They see examples of real-world variables (score and time in a football match) before they explore them in a Scratch project. Learners then design and make their own project that includes variables. Finally, learners identify that variables are named and that they can be letters (strings) as well as numbers. | | | | | To define a ‘variable’ as something that is changeable   * I can identify examples of information that is variable * I can explain that the way a variable changes can be defined * I can identify that variables can hold numbers or letters | |
| 2 Variables in programming | | * Learners understand that variables are used in programs, and that they can only hold a single value at a time. They complete an unplugged task that demonstrates the process of changing variables. Then, learners explore why it is important to name variables and apply their learning in a Scratch project in which they make, name, and update variables. | | | | | To explain why a variable is used in a program   * I can identify a program variable as a placeholder in memory for a single value * I can explain that a variable has a name and a value * I can recognise that the value of a variable can be changed | |
| 3 Improving a game | | * Learners apply the concept of variables to enhance an existing game in Scratch. They predict the outcome of changing the same change score block in different parts of a program, then they test their predictions in Scratch. Learners also experiment with using different values in variables, and with using a variable elsewhere in a program. Finally, they add comments to their project to explain how they have met the objectives of the lesson. | | | | | To choose how to improve a game by using variables   * I can decide where in a program to change a variable * I can make use of an event in a program to set a variable * I can recognise that the value of a variable can be used by a program | |
| 4 Designing a game | | * Learners work at the ‘design’ level of abstraction, where they create their artwork and algorithms. Learners first design the sprites and backgrounds for their project, then they design their algorithms to create their program flow. | | | | | To design a project that builds on a given example   * I can choose the artwork for my project * I can create algorithms for my project * I can explain my design choices | |
| 5 Design to code | | * Learners implement the algorithms that they created in Lesson 4. In doing this, they identify variables in an unfamiliar project and learn the importance of naming variables. They also have the opportunity to add another variable to enhance their project. | | | | | To use my design to create a project   * I can create the artwork for my project * I can choose a name that identifies the role of a variable * I can test the code that I have written | |
| 6 Improving and sharing | | * Learners build on the project that they created in Lesson 5. They consider how they could improve their own projects and make small changes to achieve this. Learners then have the opportunity to add a variable independently. Finally, learners evaluate each other’s projects; they identify features that they liked and features that could be improved. | | | | | To evaluate my project   * I can identify ways that my game could be improved * I can use variables to extend my game * I can share my game with others | |
| **Themes and links** | | | | | | | | |
| **Computing themes** | **Where these are covered:** | | | | | | | |
| **Technology around us**  Autumn 1 | * Scratch links to the real world and computer games the children know. | | | | | | | |
| **Digital painting**  Autumn 2 | * Design, make and evaluate process | | | | | | | |
| **Programming A**  Spring 1 | * the concept of selection in programming using the Scratch programme | | | | | | | |
| **Data /information**  Spring 2 | * Storing the commands and the effect on language on the outcome of your commands. | | | | | | | |
| **Creating media**  Summer 1 | * Your own designs of Scratch | | | | | | | |
| **Programming B**  Summer 2 | * Using scratch to implement an algorithm as a code | | | | | | | |