# Year 6 – Programming A – Variables in games

## Unit introduction

This unit explores the concept of variables in programming through games in Scratch. First, learners find out what variables are and relate them to real-world examples of values that can be set and changed. Then they use variables to create a simulation of a scoreboard. In Lessons 2, 3, and 5, which follow the Use-Modify-Create model, learners experiment with variables in an existing project, then modify them, before they create their own project. In Lesson 4, learners focus on design. Finally, in Lesson 6, learners apply their knowledge of variables and design to improve their games in Scratch.

**If learners are using the online version of Scratch, be aware this allows them to share and comment on projects. A simplified version of the Scratch's community guidelines can also be found at the end of this unit guide. For the full guidelines, see the Scratch website.**

## Software and Hardware requirements

Learners will need to have access to Scratch for this unit. The online version of Scratch runs via a web browser and can be accessed on desktops, laptops and tablets. You may want to consider setting up a teacher account, to create logins for learners to save and access their projects. If internet connectivity is an issue in school, Scratch can be accessed offline via the Scratch app.

If you've adapted this unit to better suit your school, please share your adapted resources with fellow teachers in the STEM community. Alternatively, if this unit isn't quite right for your school, why not see if an adapted version which better suits has already been shared?

## Overview of lessons

| Lesson | Brief overview | Learning objectives |
|---|---|---|

| | | |
|---|---|---|
| 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<br>• I can identify examples of information that is variable<br>• I can explain that the way a variable changes can be defined<br>• 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<br>• I can identify a program variable as a placeholder in memory for a single value<br>• I can explain that a variable has a name and a value<br>• 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<br>• I can decide where in a program to change a variable<br>• I can make use of an event in a program to set a variable<br>• I can recognise that the value of a variable can be used by a program |

| | | |
|---|---|---|
| 4 Becoming a games designer | Learners will take on the role of a games designer. They will work at the 'design' level of abstraction, where they create artwork and plan 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<br>• I can choose the artwork for my project<br>• I can create algorithms for my project<br>• I can explain my design choices |
| 5 Design to code | Continuing to use the work of games designers as a model, 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<br>• I can create the artwork for my project<br>• I can choose a name that identifies the role of a variable<br>• 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. they identify features that they liked and features that could be improved. | To evaluate my project<br>• I can identify ways that my game could be improved<br>• I can use variables to extend my game<br>• I can share my game with others |

## Request a computing ambassador

This unit is ideal for linking to the world of careers, and a computing ambassador can support this. Through the STEM ambassador platform, you can search for a computing ambassador. If you cannot find a computing ambassador with an offer to support this unit, then the following request will help to match you with the right person. You will need to edit the areas in red to ensure the request is right for your school.

*Year 6 (ages 10-11) are learning about variables in programming  though the Teach Computing Curriculum unit of six lessons. Within these lessons, pupils will learn the skills needed to design and create a game on Scratch*

*Our lessons are taking place from *date* to *date* and we would appreciate someone with skills in this area to offer some real-world experience to this unit. The unit uses the programming interface of Scratch, with pupils coding via drag and drop block-based code, and focuses on the following areas:*

- *understand variables as placeholders and how these can be used to create a score*
- *modify existing animations and games using variables*
- *design a game, including both the artwork and the algorithm for their project*
- *build a game following my design algorithm*

*We require an ambassador who can support in any of these areas. We are hoping for an ambassador who would be willing to join us *in the classroom/virtually* to support our learning by *providing some handy hints and tips for our projects/giving us constructive feedback on our final projects/discussing how programming to create games is used within their profession and in the real-world.**

## Subject knowledge and CPD opportunities

This unit focuses on developing learners' understanding of variables in Scratch, a block-based programming language. It emphasises where variables can be used and how they can be set and changed through the running of a program.

**Variables**

You need to be aware of the concept of variables in programming. In this lesson, a 'variable' is defined as something that can be set and changed throughout the running of a program. You need to know that a variable is a placeholder for a single value in the memory of a computer, and that all variables are uniquely named. You need to know that when the value of a variable is updated, the original value is replaced.

**Continual Professional Development**

Enhance your subject knowledge to teach this unit through the following free CPD:

- Getting started in Year 6 – short course
- Introduction to primary computing remote or face to face
- Teaching programming to 5- to 11-year-olds
- Introduction to Programming with Scratch

- Teaching programming using Scratch and Scratch Jr

**Teach primary computing certificate**
To further enhance your subject knowledge, enrol on the teach primary computing certificate. This will support you to develop your knowledge and skills in primary computing and gain the confidence to teach great lessons, all whilst earning a nationally recognised certificate!

## Progression

This unit assumes that learners will have prior experience of programming using block-based construction (e.g. Scratch), understand the concepts of 'sequence' (Year 3 units: Sequencing Sounds and Events and actions in programs), 'repetition' (Year 4 units: Repetition in shapes and Repetition in games), and 'selection' (Year 5 units: Selection in Physical Computing and Selection in quizzes). The constructs covered in the previous year groups will include at least one unit that develops the concept through the use of Scratch.

## Common Misconceptions

When introducing variables, it is important to ensure that learners understand the meaning of the word within programming, as they are likely already familiar with the word from Science where it has a slightly different meaning. To avoid misconceptions forming, pupils need to know that a variable holds a piece of data, it can only hold one piece of data at once and this can change throughout the program.

The analogy of a small box is often used to help pupils understand variables – the box can hold a piece of data, but because it is small, we have to get rid of the last piece of data to put a new one in. The box can only hold one piece of data. We can replace it, but it can only ever fit one piece of data in. Sticky notes are also used in the unit to demonstrate this. Ensure pupils remove the last sticky note before placing their new one, to demonstrate that the last value is gone.

Score is often used to introduce variables, and is used in this unit. This is because score is a familiar variable that pupils have likely experienced before. However, it is important to emphasise that score is one example of a variable, but that not all variables are 'score'. This is covered in lesson two, but as score is used many times throughout the unit as a familiar example, pupils would benefit from being reminded of this.

## Curriculum links

**Computing**

- 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

## Assessment

**Formative assessment**

Assessment opportunities are detailed in each lesson plan. The learning objectives and success criteria are introduced in the slide deck at the beginning of each lesson, and then reviewed at the end. Learners are invited to assess how well they feel they have met the learning objective using thumbs up, thumbs sideways, or thumbs down.

We recommend the use of teacher accounts in Scratch to help with assessment throughout this unit. For guidance on setting up teacher accounts, please visit the Scratch website (scratch.mit.edu/educators/faq).

**Summative assessment**

Please see the summative assessment document of multiple-choice questions for this unit. This can be downloaded as a paper copy, with answers, or in a digital format to be shared.

## Scratch guidelines

- **Stay Safe Online**: Don't share personal info like your full name, address, or phone number. Also, don't share details about where you go to school or your social media accounts.

- **Be Kind and Helpful**: When you comment on someone's project, say something nice about it and offer suggestions in a friendly way. Don’t be mean or spammy.

- **Share and Collaborate**: You can use other people's stuff on Scratch to make your own cool projects but remember to give credit. And when you share your work, others can use it too, as long as they give credit and make changes.

- **Be Honest**: Always tell the truth and be yourself when you're on Scratch. Don’t pretend to be someone else.

- **Keep Scratch Friendly**: Make sure your creations and chats are friendly for everyone. If you see something mean or inappropriate, you can click the link that says "report" on any project, comment, discussion post, studio, or profile page. If you're unsure or it's a bit complicated, you can ask your teacher or a trusted adult to get in touch with us. The Scratch team will take care of it.