Year 3 – Programming B – Events and actions in programs

Unit introduction

This unit explores the links between events and actions, while consolidating prior learning relating to sequencing. Learners begin by moving a sprite in four directions (up, down, left, and right). They then explore movement within the context of a maze, using design to choose an appropriately sized sprite. This unit also introduces programming extensions, through the use of **Pen** blocks. Learners are given the opportunity to draw lines with sprites and change the size and colour of lines. The unit concludes with learners designing and coding their own maze-tracing program.

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 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 Moving a sprite	In this lesson, learners will investigate how characters can be moved using 'events'. They will analyse and improve an existing project, and then apply what they have learned to their own projects. They will then extend their learning to control multiple sprites in the same project.	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
2 Maze movement	In this lesson, learners will program a sprite to move in four directions: up, down, left, and right. They will begin by choosing a sprite and sizing it to fit in with a given background. Learners will then create the code to move the sprite in one direction before duplicating and modifying it to move in all four directions. Finally, they will consider how their project could be extended to prove that their sprite has successfully navigated a maze.	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
3 Drawing lines	This lesson will introduce learners to extension blocks in Scratch using the **Pen** extension. Learners will use the pen down block to draw lines, building on the movement they created for their sprite in Lesson 2. Learners will then decide how to set up their project every time it is run.	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
4 Adding features	In this lesson, learners will be given the opportunity to use additional **Pen** blocks. They will predict the functions of new blocks and experiment with them, before designing features to add to their own projects. Finally, they will add these features to their projects and test their effectiveness.	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
5 Debugging movement	This lesson explores the process of debugging, specifically looking at how to identify and fix errors in a program. Learners will review an existing project against a given design and identify bugs within it. They will then correct the errors, gaining independence as they do so. Learners will also develop their projects by considering which new setup blocks to use.	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
6 Making a project	In this lesson, learners will design and create their own projects. Using a template (which can be blank or partially completed), learners will complete projects to move a sprite around a maze, with the option to leave a pen trail showing where the sprite has moved. Ideally, projects will include setup blocks to position the sprite at the start of the maze and clear any lines already on the screen.	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

Subject knowledge and CPD opportunities

This unit focuses on the links between 'events' and 'actions' in programming, while also developing learners' understanding of sequencing. It highlights that events cause actions, and that the order of those actions can have an impact on the outcome of a program. This unit also further develops learners' understanding of design in programming, using the approach outlined below. As with unit A, videos are embedded in lesson files which demonstrate the use of Scratch, but pedagogically it is more beneficial to model these live with pupils.

When programming, there are four levels that help to describe the stages of a project, known as levels of abstraction. Research suggests that this structure can support learners in understanding how to create a program and how it works.

- Task — this is what is needed
- Design — this is what it should do
- Code — this is how it is done

- Running the code — this is what it does

Spending time at the Task and Design levels before engaging in code writing aids learners in assessing the 'do-ability' of their programs and reduces a learner's cognitive load during programming. Learners will move between the different levels throughout the unit. This is highlighted within each lesson plan.

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

- Getting started in Year 3 – short course
- Introduction to primary computing remote or face to face
- 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 some prior experience of programming; via the KS1 NCCE units. They will have experienced programming via floor robots; Moving A Robot Year 1 and Robot algorithms Year 2, alongside the use of ScratchJr through Programming animations Year 1 and Programming quizzes Year 2. The Year 3 Programming A unit introduces the Scratch programming environment and the concept of sequences.

Common Misconceptions

When developing code for left and right keys, learners need to be aware that Scratch uses x and y coordinates. The stage works as a grid, so x and y coordinate numbers indicate locations on the stage. The x coordinate gives the left-right position, and the y coordinate gives the up-down position.

Learners may get confused when changing x/y by 10/-10. Searching ‘xy-grid’ on the backdrops of Scratch, and using the provided background image, may support learners.

When introducing the pen extension, ‘pen down’ may cause confusion. Some may understand this as putting a pen down on the table, instead of putting the pen down, nib first, to draw on the screen.

Learners may also have misconceptions between the ‘pen up’ and ‘erase all’ blocks. The pen up block stops the sprite from drawing, and the erase all block removes all lines from the screen. When using the erase all block, learners may not be aware that the sequence of blocks is important. For example, if the ‘erase all’ block is before the ‘go to’ block, a line will be drawn when the sprite returns to the start position.

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 objective 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.

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.

We recommend the use of teacher and learner accounts in Scratch to help with assessment throughout this unit. For guidance on setting up teacher accounts, visit scratch.mit.edu/educators/faq. A teacher account enables you to manage learners' accounts and organise projects into studios. If you are unable to use teacher and learner accounts, work can be saved offline to local devices.

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.