# Year 3 – Programming B - Events and actions

## Unit introduction

This unit explores the links between events and actions, whilst consolidating prior learning relating to sequencing. Learners will begin by moving a sprite in four directions (up, down, left and right). They will 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.

There are two year 3 programming units:

* Programming A - Sequence in music
* Programming B - Events and actions

This is unit B which should be delivered after unit A.

## Overview of lessons

|  |  |  |
| --- | --- | --- |
| **Lesson** | **Brief overview** | **Learning objectives** |
| Online Safety – Unit Opener | In this lesson, children understand and investigate the importance of keywords in ensuring they find the information they need safely when using search engines. | To demonstrate how to use key phrases in search engines to gather accurate information online. |
| * Moving a sprite

[ncce.io/pg3b-1-p](https://ncce.io/pg3a-1-p) | 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. 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
 |
| * Maze movement

[ncce.io/pg3b-2-p](https://ncce.io/pg3a-2-p) | In this lesson, learners will be programming 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 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
 |
| * Drawing lines

[ncce.io/pg3b-3-p](https://ncce.io/pg3a-3-p) | This lesson introduces 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 the previous lesson. 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
 |
| * Adding features

[ncce.io/pg3b-4-p](https://ncce.io/pg3a-4-p) | In this lesson learners are given the opportunity to use additional pen blocks. They will predict the function of new blocks, experiment with them, before designing features to add into their own project. Finally they will add these features to their project 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
 |
| * Debugging movement

[ncce.io/pg3b-5-p](https://ncce.io/pg3a-5-p) | 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 the set up of their project by considering which new 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
 |
| * Making a project

[ncce.io/pg3b-6-p](https://ncce.io/pg3a-6-p) | In this lesson, learners will design and create a project. Using a template (which can be blank or partially completed) learners will complete a project 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 set up 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
 |

## Progression

This unit assumes that learners will have some prior experience of programming. The KS1 NCCE units focus on floor robots and Scratch Jr, however experience of other languages or environments may also be useful. The Year 3 Programming A unit introduces the Scratch programming environment and the concept of sequences.

[Y3 Events and actions – Learning graph](https://docs.google.com/drawings/d/1iXog0viYavrvPFi15b02bpOWrobvvvc_Kf8rQM2rNHY) (ncce.io/pg3b-lg)

## Curriculum links

[**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

## 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**

* [Assessment questions](https://docs.google.com/document/d/1VJo87GO0aUUdssA8EOAwoEjUPW6sAavpe0CCBCOFhzQ) (ncce.io/pg3b-saq) and [answers](https://docs.google.com/document/d/1eNWbODFY6Fa-DEctwE3GWRyd3ClC3xDZSQr3nLEX624) (ncce.io/pg3b-saa) are included at the end of the unit to support summative assessment.

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, please follow [this link](https://scratch.mit.edu/educators/faq). (<https://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.

## Subject knowledge

This unit focuses on the links between events and actions in programming, whilst 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.

When programming, there are four levels which can help describe 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 and this is highlighted within each lesson plan.

Enhance your subject knowledge to teach this unit through the following training opportunities:

### Online training courses

* [Raspberry Pi Foundation online training courses](https://www.futurelearn.com/partners/raspberry-pi)

### Face-to-face courses

* [National Centre for Computing Education face-to-face training courses](https://teachcomputing.org/courses)

This resource is available online at [ncce.io/pg3b-o](https://docs.google.com/document/d/1YhMbzRlm1Jz9l_ruO77QHF8cZX97EGiTt3RduDgnpEw). Resources are updated regularly — please check that you are using the latest version.

This resource is licensed under the Open Government Licence, version 3. For more information on this licence, see [ncce.io/ogl](https://ncce.io/ogl).