Year 5 – Programming A – Selection in physical computing

Unit introduction

In this unit, learners will use physical computing to explore the concept of selection in programming through the use of the Crumble programming environment. Learners will be introduced to a microcontroller (Crumble controller) and learn how to connect and program it to control components (including output devices — LEDs and motors). Learners will be introduced to conditions as a means of controlling the flow of actions in a program. Learners will make use of their knowledge of repetition and conditions when introduced to the concept of selection (through the 'if...then...' structure) and write algorithms and programs that utilise this concept. To conclude the unit, learners will design and make a working model of a fairground carousel that will demonstrate their understanding of how the microcontroller and its components are connected, and how selection can be used to control the operation of the model. Throughout this unit, learners will apply the stages of programming design.

Software and Hardware requirements

The unit has been designed to make use of the components provided in the Crumble starter kit. Learners will also need devices capable of running the Crumble software with a USB port to connect the crumble. If access to crumble kits will prevent you teaching this unit, your local hub may be able to support with a loan kit. Find out more by visiting Physical Computing Kits - Teach Computing

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 Connecting Crumbles	In this lesson, your learners will become familiar with the Crumble controller and the programming environment used to control it. Learners will connect a Sparkle to a Crumble and then program the Crumble to make the Sparkle flash different colour patterns. Learners will also use infinite loops, which were introduced to the learners in the previous school year.	To control a simple circuit connected to a computer • I can create a simple circuit and connect it to a microcontroller • I can program a microcontroller to make an LED switch on • I can explain what an infinite loop does
2 Combining output components	In this lesson, learners will connect a Sparkle and a motor to the Crumble controller. Learners will design sequences of actions for these components. They will then apply their understanding of repetition by using count-controlled loops when implementing their design as a program.	To write a program that includes count-controlled loops • I can connect more than one output component to a microcontroller • I can use a count-controlled loop to control outputs • I can design sequences that use count-controlled loops
3 Controlling with conditions	In this lesson, learners will be introduced to conditions, and how they can be used in programs to control their flow. They will identify conditions in statements, stating if they are true or false. Learners will be introduced to a Crumble switch, and learn how it can provide the Crumble controller with an input that can be used as a condition. They will explore how to write programs that use an input as a condition.	To explain that a loop can stop when a condition is met • I can explain that a condition is either true or false • I can design a conditional loop • I can program a microcontroller to respond to an input
4 Starting with selection	In this lesson, learners will develop their understanding of how the flow of actions in algorithms and programs can be controlled by conditions. They will be introduced to selection and then represent conditions and actions using	To explain that a loop can be used to repeatedly check whether a condition has been met

	the 'if…then…' structure. Learners will create algorithms that include selection. They will use their algorithms to guide their program writing. Learners will see that infinite repetition is required to repeatedly check if a condition has been met.	• I can explain that a condition being met can start an action • I can identify a condition and an action in my project • I can use selection (an 'if…then…' statement) to direct the flow of a program
5 Drawing designs	In this lesson, learners will apply their understanding of microcontrollers and selection when designing a project to meet the requirements of a given task. To support their understanding, learners will identify how selection might be used in real-world situations, then they will consider how they can apply this knowledge to design their project. Learners will produce design sketches to show how their model will be made and how they will connect the microcontroller to its components.	To design a physical project that includes selection • I can identify a real-world example of a condition starting an action • I can describe what my project will do • I can create a detailed drawing of my project
6 Writing and testing algorithms	In this final lesson of the unit, learners will develop Crumble programs to control the model of a fairground ride they built in Lesson 5. First, learners will identify how they are going to use selection before writing an algorithm to meet the requirements of the given task. They will then implement their algorithms as code. Learners will run their programs to identify any bugs, and then return to the code or algorithm to debug it where necessary. Finally, to conclude the unit, learners will evaluate their designs.	To create a program that controls a physical computing project • I can write an algorithm that describes what my model will do • I can use selection to produce an intended outcome • I can test and debug my project

Subject knowledge

This unit focuses on physical computing, which allows learners to control real-life projects through the construction of programs. When learners undertake physical computing, they write programs that control real-world objects, like LEDs and motors, using a computer. The tangible effect of seeing the

commands that they entered into a computer being carried out on a physical item, rather than on screen, can be highly motivational for learners. Physical computing also offers the opportunity to take a more project-based approach to learning, and allows learners to make choices about the purpose, design, and program of their product.

For this unit, you will need experience of constructing programs using the Crumble programming software. It uses the same drag-and-drop style as Scratch. You will need to write programs that turn LEDs (Sparkles) on and off, change LED colours, spin motors, use push switches as inputs, and combine a number of these components. Additionally, you will connect the Crumble controller to battery packs, Sparkles, motors, and push switches. For further support on using Crumbles, see the Crumble 'Getting Started' guide at redfernelectronics.co.uk/crumble-getting-started.

Repetition

You will need to know that repetition is used in programming to give the same instruction or set of instructions several times. Repetition uses loops as the means to give these instructions. This unit makes use of two types of loops: infinite and count-controlled, which have been defined below:

- Infinite loop: An infinite loop is a loop that commands the instruction/set of instructions to repeat forever. When an infinite loop is used in a program, there is no way of ending the program, as the command(s) within the loop will be repeated endlessly. For this reason, infinite loops should only be used when writing a program that is intended to run forever. The exception to this is when using selection in physical computing, as you will see throughout this unit.
- 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. Count-controlled loops should only be used when it is known how many times a set of commands needs to be repeated.
- 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. The condition could be anything from when the 'score' in a game reaches a certain value to when a key on a keyboard has been pressed.

Conditions

Conditions are statements that need to be met for a set of actions to be carried out. They can be used in algorithms and programs to control the flow of actions. When a condition is met, it is referred to as 'true' and when it is not met, it is referred to as 'false'. You will need to be able to identify and use conditions in algorithms in the form of statements to both start and stop sets of action. Additionally, you will need to understand that conditions can be used in loops, and when they are, that the set of actions in the loop will be carried out repeatedly until the condition is true, for example, 'until button A is pressed'.

Selection

Selection is "part of a program where, if a condition is met, then a set of commands are run". Selection is implemented in programming using `if...then...` statements. Selection is used to control the flow of actions in algorithms and programs by checking if a condition (see above) has been met. If it has been met, the identified actions will be carried out. When selection is used in programs, loops (see above) often have to be used to instruct the device to check the condition repeatedly. Without using loops, the condition would only be checked once. It's important to understand that each loop cycle will complete before the condition is checked again. In the Crumble programming software, selection is implemented through the `if...then...` command block.

In addition to the above, you will also need to understand that programs are an implementation of an algorithm, and that when the program does not produce the required output, the algorithm should be debugged. This should then be implemented in the program.

Continual Professional Development

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

- Getting started in Year 5
- Introduction to primary computing
- Introduction to Crumble in KS2
- Teaching programming to 5- to 11-year-olds

Primary Certificate

To further enhance your subject knowledge, enrol on the primary 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 a block-based language (e.g. Scratch) and understand the concepts of sequence and repetition. The National Centre for Computing Education Key Stage 1 units focus on floor robots and ScratchJr, however, experience of other languages or environments may also be useful.

See the learning graph for this unit for more information about progression.

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

Science – Electricity (Year 4)

- Construct a simple series electrical circuit, identifying and naming its basic parts, including cells, wires, bulbs, switches, and buzzers

Design and Technology (Key stage 2)

Design

- Generate, develop, model, and communicate their ideas through discussion, annotated sketches, cross-sectional and exploded diagrams, prototypes, pattern pieces, and computer-aided design

Make

- Select from and use a wider range of tools and equipment to perform practical tasks [for example, cutting, shaping, joining, and finishing], accurately
- Select from and use a wider range of materials and components, including construction materials, textiles, and ingredients, according to their functional properties and aesthetic qualities

Evaluate

- Evaluate their ideas and products against their own design criteria and consider the views of others to improve their work

Technical knowledge

- Understand and use electrical systems in their products [for example, series circuits incorporating switches, bulbs, buzzers, and motors]
- Apply their understanding of computing to program, monitor, and control their products

Assessment

Formative assessment

Assessment opportunities are detailed in each lesson plan. The learning objectives and success criteria are introduced in the slide decks 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 assessment rubric document for this unit. The rubric can be used to assess student's work from lessons 5 and 6.

Resources are updated regularly — the latest version is available at: ncce.io/tcc.

Attribution statement

Year 5 – Programming B – Selection in quizzes

Unit introduction

Learners will develop their knowledge of 'selection' by revisiting how 'conditions' can be used in programming, and then learning how the 'if… then… else…' structure can be used to select different outcomes depending on whether a condition is 'true' or 'false'. They represent this understanding in algorithms, and then by constructing programs in the Scratch programming environment. They learn how to write programs that ask questions and use selection to control the outcomes based on the answers given. They use this knowledge to design a quiz in response to a given task and implement it as a program. To conclude the unit, learners evaluate their program by identifying how it meets the requirements of the task, the ways they have improved it, and further ways it could be improved.

This unit uses the Scratch programming platform to support pupils' learning in computing. Scratch offers an engaging environment for developing key programming skills through creativity and experimentation. However, it is essential that teachers using Scratch understand how to do so safely.
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.

Teachers have a statutory duty to protect pupils from potential risks associated with using online platforms, including those that enable content sharing and interaction. The Department for Education's guidance on *Keeping Children Safe in Education* makes clear that safeguarding extends to online activity and digital tools used in school.

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
Exploring conditions	In this lesson, learners revisit previous learning on 'selection' and identify how 'conditions' are used to control the flow of actions in a program. They are introduced to the blocks for using conditions in programs using the Scratch programming environment. They modify the conditions in an existing program and identify the impact this has.	To explain how selection is used in computer programs • I can recall how conditions are used in selection • I can identify conditions in a program • I can modify a condition in a program
Selecting outcomes	In this lesson, learners will develop their understanding of selection by using the 'if… then… else...' structure in algorithms and programs. They will revisit the need to use repetition in selection to ensure that conditions are repeatedly checked. They identify the two outcomes in given programs and how the condition informs which outcome will be selected. Learners use this knowledge to write their own programs that use selection with two outcomes.	To relate that a conditional statement connects a condition to an outcome • I can use selection in an infinite loop to check a condition • I can identify the condition and outcomes in an 'if… then… else…' statement

		• I can create a program that uses selection to produce different outcomes
Asking questions	In this lesson, learners consider how the 'if… then... else…' structure can be used to identify two responses to a binary question (one with a 'yes or no' answer). They identify that the answer to the question is the 'condition', and use algorithms with a branching structure to represent the actions that will be carried out if the condition is true or false. They learn how questions can be asked in Scratch, and how the answer, supplied by the user, is used in the condition to control the outcomes. They use an algorithm to design a program that uses selection to direct the flow of the program based on the answer provided. They implement their algorithm as a program and test whether both outcomes can be achieved.	To explain how selection directs the flow of a program • I can explain that program flow can branch according to a condition • I can design the flow of a program that contains 'if… then… else…' • I can show that a condition can direct program flow in one of two ways
Designing a quiz	In this lesson, learners will be provided with a task: to use selection to control the outcomes in an interactive quiz. They will outline the requirements of the task and use an algorithm to show how they will use selection in the quiz to control the outcomes based on the answer given. Learners will complete their designs by using design templates to identify the questions that will be asked, and the outcomes for both correct and incorrect answers. To demonstrate their understanding of how they are using selection to control the flow of the program, learners will identify which outcomes will be selected based on given responses.	To design a program that uses selection • I can outline a given task • I can use a design format to outline my project • I can identify the outcome of user input in an algorithm
Testing a quiz	In this lesson, learners will use the Scratch programming environment to implement the first section of their algorithm as a program. They will run the first section of their program to test whether they have correctly used selection to control the outcomes, and debug their program if required. They will then continue implementing their algorithm as a program. Once completed, they will	To create a program that uses selection • I can implement my algorithm to create the first section of my program • I can test my program

	consider the value of sharing their program with others so that they can receive feedback. Learners conclude the lesson by using another learner's quiz and providing feedback on it.	• I can share my program with others
Evaluating a quiz	In this lesson, learners will return to their completed programs and identify ways in which the program can be improved. They will focus on issues where answers similar to those in the condition are given as inputs, and identify ways to avoid such problems. Learners will also consider how the outcomes may change the program for subsequent users, and identify how they can make use of 'setup' to provide all users with the same experience. They will implement their identified improvements by returning to the Scratch programming environment and adding to their programs. They conclude the unit by identifying how they met the requirements of the given task, and identifying the aspects of the program that worked well, those they improved, and areas that could improve further.	To evaluate my program • I can identify ways the program could be improved • I can identify the setup code I need in my program • I can extend my program further

Subject knowledge and CPD opportunities

This unit focuses on developing learners' understanding of selection in an on-screen context. It highlights what 'conditions' are and how they are used as part of 'selection'.

Conditions

'Conditions' are statements that need to be met for a set of actions to be carried out. They can be used in algorithms and programs to control the flow of actions. When a condition is met it is referred to as 'true' and when it is not met it is referred to as 'false'. You need to be able to identify and use conditions in algorithms in the form of statements to both start and stop sets of action. Additionally, you need to understand that conditions can be used in loops, and when they are, that the set of actions in the loop will be carried out repeatedly until the condition is true. For example, 'until button 'A' is pressed'.

Selection

When designing programs, there are often points where a decision must be made. These decisions are known as 'selection', and are commonly

implemented in programming using 'if' statements. Selection is used to control the flow of actions in algorithms and programs by checking whether a condition (see above) has been met. If it has been met, the identified actions will be carried out. When selection is used in programs, infinite loops (see above) are often used to instruct the device to check the condition repeatedly. Without using loops, the condition would only be checked once following the sequence of the code.

Continual Professional Development

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

- Getting started in Year 5
- Introduction to primary computing
- Teaching programming to 5- to 11-year-olds
- Teaching programming using Scratch and Scratch Jr
- Introduction to Programming with Scratch

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) and 'repetition' (Year 4 units: Repetition in shapes and Repetition in games), and have some experience of using 'selection'. Ideally, learners will have completed 'Programming A – Selection in physical computing' before undertaking this unit, as this will provide them with the required knowledge of 'selection'.

Please see the learning graph for this unit for more information about progression.

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. Pupils are invited to assess how well they feel they have met the learning objectives 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.

Resources are updated regularly — the latest version is available at: ncce.io/tcc.

Attribution statement

Year 5 – Video production

Unit introduction

Learners will learn how to create short videos by working in pairs or groups. As they progress through this unit, they will be exposed to topic-based language and develop the skills of capturing, editing, and manipulating video. Learners are guided with step-by-step support to take their idea from conception to completion. At the conclusion of the unit, learners have the opportunity to reflect on and assess their progress in creating a video.

Software and Hardware requirements

To teach this unit, you will need video recording equipment such as digital/video cameras or tablets with video capabilities. The lesson videos show Windows Video Editor which, unfortunately is no longer available, however the skills within this unit can be transferred and it can be delivered with any video editing software. For those schools with access to iPads, iMovie is a suitable alternative to complete the unit.

Video editing is also possible on Canva Video Editor, which is web based can be used on any device. There is more advice on Canva, and on setting up Canva for Education accounts, as part of the unit guide for Year 3 Desktop publishing.

Overview of lessons

Lesson	Brief overview	Learning objectives
1 What is video?	Learners will be introduced to video as a media format. They will see examples of videos featuring production and editing techniques that they will work towards	To explain what makes a video effective • I can explain that video is a visual media format

	using their own videos. Learners will begin by explaining what the medium of video is before analysing and comparing examples of videos.	• I can identify features of videos • I can compare features in different videos • I know what to do if I see any content online that makes me feel uncomfortable
2 Filming techniques	Learners will explore the capabilities of a digital device that can be used to record video. Once they are familiar with their device, learners will experiment with different camera angles, considering how different camera angles can be used for different purposes.	To use a digital device to record video • I can identify and find features on a digital video recording device • I can experiment with different camera angles • I can make use of a microphone
3 Using a storyboard	Learners will use a storyboard to explore a variety of filming techniques, some of which they will use in their own video project later in the unit. They will evaluate the effectiveness of these techniques before offering feedback on others' work.	To capture video using a range of techniques • I can suggest filming techniques for a given purpose • I can capture video using a range of filming techniques • I can review how effective my video is
4 Planning a video	Learners will plan a video by creating a storyboard. Their storyboard will describe each scene, and will include a script, camera angles, and filming techniques. Learners will use their storyboards to film the first scene of their videos.	To create a storyboard • I can outline the scenes of my video • I can decide which filming techniques I will use • I can create and save video content
5 Importing and editing	Learners will film the remaining scenes of their video, and then import their	To identify that video can be improved

video	content to video editing software. Learners will be introduced to the job role of a video producer and will take on the role of a video producer to edit their creations. They will then explore key editing techniques and decide whether sections of their video can be edited or need to be shot again.	through reshooting and editing • I can store, retrieve, and export my recording to a computer • I can explain how to improve a video by reshooting and editing • I can select the correct tools to make edits to my video
6 Video evaluation	Learners will complete their video by removing unwanted content and reordering their clips. They will then export their finished video and evaluate the effectiveness of their edits. Finally, they will consider how they could share their video with others.	To consider the impact of the choices made when making and sharing a video • I can make edits to my video and improve the final outcome • I can recognise that my choices when making a video will impact the quality of the final outcome • I can evaluate my video and share my opinions

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 5 (ages 9-10) are learning about video production though the Teach Computing Curriculum unit of six lessons. Within these lessons, pupils will learn how to create their own short videos.

*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 https://www.getpaint.net/*insert software* on *insert devices* and focuses on the following areas:*

- *understanding the different production and editing techniques for creating videos, such as camera angles and framing, and static cameras, zoom, pan and tilt*
- *understand storyboards as tools for planning videos*
- *filming and improving a video by reshooting and editing*
- *evaluating our video creations*

*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 video production is used within their profession and in the real-world.**

Subject knowledge and CPD opportunities

This unit focuses on the skills associated with planning, recording, editing, and creating a video. You will need to be able to explain that video is the recording, reproducing, or broadcasting of moving visual images. You will also need to be familiar with a number of shot types and filming techniques, which are introduced in Lessons 2 and 3. A storyboard is used as a planning tool. It will be useful if you are familiar with the format of the storyboard.

Once learners begin filming, you will need to be familiar with the device they are using, including how to start and stop recording, how to zoom in and out, and how to download content from the device to a computer for editing. It is important that you are familiar with the devices and apps or programs that you will use to import and edit video content. You need to know where to locate the video files and where to save them for easy retrieval.

Once recording has been completed, learners will need to import their video files to the video editing software, so you will need to be familiar with this process, including where videos will be stored. You will need to have a clear understanding of how to edit and complete the video creation process, deleting or reordering clips. Finally, you should be able to demonstrate how to export the video project into an *.mp4 format for viewing.

Continual Professional Development

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

- Getting started in Year 5
- Introduction to primary computing

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 progresses learners' knowledge and understanding of creating media by guiding them systematically through the process involved in creating a video. The unit builds on the Year 4 Photo editing unit where composition is introduced and the Year 3 unit 'Stop-frame animation' where learners explored some of the features of video production. By the end of this unit, learners will have developed the skills required to plan, record, edit, and share a video.

Common Misconceptions

Learners may believe that editing is just about cutting out mistakes, however this unit will show that it is a creative process that involves arranging clips, adding effects, transitions, audio and ensuring the video flows coherently. They may have the misconception that the pre-production elements (planning, scripting, and storyboarding), are unnecessary, but it is crucial for the structure and vision of the project. Learners may assume that any footage taken with a camera will look good, but this unit will highlight that good footage requires an understanding of camera angles, framing, lighting and steady camera movements. When editing, they may assume that must edit the film in the order it was shot, however this process does not have to be linear, and reshooting can take place at any point. They may assume that adding lots of special effects will automatically make a video better, however overusing these can make a video look cluttered and distract from the content.

Curriculum links

Computing

- Use search technologies effectively, appreciate how results are selected and ranked, and be discerning in evaluating digital content
- 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
- Use technology safely, respectfully, and responsibly; recognise acceptable/unacceptable behaviour; identify a range of ways to report concerns about content and contact

Education for a Connected World links

Online relationships

- I can explain how someone can get help if they are having problems and identify when to tell a trusted adult.

Assessment

Formative assessment

Assessment opportunities are detailed in each lesson plan. The learning objectives and success criteria are introduced in the slide decks 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 assessment rubric document for this unit. The rubric can be used to assess student's work from lessons 4 to 6.