Game lesson title	**Game Builder - Level 1**		
Brief intro	Perfect for students who have had little to no experience making a videogame. This unit will take students through all aspects of making a video game including: ● Planning a narrative and designing characters ● Learning the basics of logic and coding ● Building and playtesting their first game Level 1 uses resources and programs that are all available for free use online (see Game Builder Lever 1). Students should keep track of logins and URLs for their projects, as well as save and backup the assets they create throughout.		
Lesson hook	Making videogames - the perfect introduction for first-time game makers.		
Suggested year level	3-8	**Suggested age level**	8-14
Suggested duration	8 x 45 min sessions (we recommend working in sequence order, some may require more time)	**Costs involved?** (detail below)	Free
Author	You are free to copy, communicate and adapt this lesson plan which was created by Garry Westmore and ACMI, and licensed under a Creative Commons Attribution 2.0		
Subject/s			
☒ Technologies, Mathematics, English, Arts, Media Arts			
Curriculum/Capabilities Alignment (VIC/AC) and Skills			
VCAMAE029 - Explore representations, characterisations and viewpoints of people in their community, using stories, structure, settings, and genre conventions in images, sounds and text. VCELT284 - Discuss how authors and illustrators make stories exciting, moving and absorbing and hold readers' interest by using various techniques VCDTDI039 - Manage, create and communicate interactive ideas, information and projects collaboratively online, taking safety and social contexts into account			

VCDTCD031 - Design a user interface for a digital system, generating and considering alternative design ideas

VCAMAM036 - Plan, structure and design media artworks for a range of purposes that engage audiences using media elements, technologies and production processes

Game/Program Used	A Short Trip, Tetris, Pong: free games used as reference Scratch 3.0: a free, easy to use online block coding program Beepbox: a chiptune music making program Piskel: an online pixel-art editor		
Game play required?	Optional (can also watch playthroughs online)		
CLASSIFICATION	**CONSOLE**	**INTERNET REQUIRED?**	**COST (RRP)**
G	Mac or PC computer, tablet, or phone	Yes	Free

Important note about Game Classification

As classifications can change, teachers are responsible for checking the latest videogame classification and suitability for their class age group. Please visit https://www.classification.gov.au/ and https://www.commonsense.org/education/ to guide you.

How are games used by students in the lesson?

☐ **Watchers** – observing, analysing and evaluating. Learning about the world and ourselves through understanding the impact of games culture and industry.

☐ **Players** - learning by playing videogames - learning things applicable to life outside of (and in) the game e.g. flight simulators, esports, etc.

☒ **Makers** - learning through making games (coding, creative production, teamwork, leadership)

☐ **Explorers** (Minecraft) – imaginative, self-directed, exploratory/sandbox learning.

eSafety Considerations

N/A

Prior knowledge/skills (Required/Recommended/References)

No prior knowledge required.

In this unit we start by riffing on Paper Scissors Rock. If there's another simple game that's more culturally relevant to your students, feel free to substitute here.

LESSON 1 SEQUENCE - WHAT MAKES A GOOD GAME	
Student outcomes & Learning goals	In this lesson, students analyse the different elements that make up a good game. **By the end of this lesson students should be able to:** Identify the key elements that make up a game such as gameplay, graphics and audio **By the end of this lesson students will have:** Conducted meaningful conversations about what they enjoy about videogames Analysed student-made games from Screen It.
Prep & introduction activities	*Setup screen to watch videos, and paper and writing materials. Print any worksheets linked below.* ACTIVITY: **Games we love** Different people have different ideas about what games are good, but game developers and game players probably agree on at least a few elements. Watch Bajo and Hex talk about what they think makes a great game in the video below. (4 min watch)
Main lesson activities	After, discuss with your group what you think makes a good game. Did you have different ideas to Bajo and Hex? Choose a game to play as a group. After playing, brainstorm what you enjoyed. Focus on the gameplay but feel free to talk about graphics, music, sound effects and any other fun elements. There are some game suggestions below. Play student-made games from ACMI's Screen It competition here: acmiscreenit.itch.io/ Or if you prefer, you can play a browser-based game instead we recommend A Short Trip. Play it online, it takes about 5 minutes to complete!
Reflection activity	This module includes interviews with ACMI staff who love videogames. We think that to make good videogames, you should discuss with others what you like about the videogames you play. Watch ACMI staff talking about the videogames they like and consider what it is that keeps you coming back to a game.
Student Homework/Further Work	
Play your favourite game and note the how the aspects we discussed today.	

LESSON 2 SEQUENCE - WHAT ARE GAME MECHANICS?	
Student outcomes & Learning goals	In this lesson, students start to understand what game mechanics are, and how they influence gameplay. Through designing and testing their own paper games students can start to think about how testing can result in improved gameplay. **By the end of this lesson students should be able to:** Display an understanding of what game mechanic are and how they influence gameplay Understand 'bugs' and the importance of playtesting **By the end of this lesson students will have:** Discussed and articulated an understanding of game mechanics through examples Created in a group a paper game with game mechanics and gameplay Tested and revised a paper game through playtesting
Prep & introduction activities	*Setup screen to watch videos, and paper and writing materials. Print any worksheets linked below.* What are game mechanics? Game mechanics are the rules and interactive elements of a game. These rules and elements make up the gameplay. To create good gameplay, game designers think about how the game mechanics work together, and the gameplay they want players to experience. Good game developers try not to make a game too repetitive, too easy, or frustratingly difficult. Ideally, the game is easy to pick up, and hard to master. If you're starting out, just creating game mechanics that result in actual gameplay is a win. From there you might tweak the mechanics of your videogame or add more elements to improve gameplay. You can experiment with game mechanics by designing other types of games first – mini-sports, board games and playground games all involve rules and methods of play you can test. Watch ACMI staff talk about their favourite game mechanics.
Main lesson activities	BRAINSTORM: **What are game mechanics?** 1. Students form small groups 2. Within your group talk about a videogame you really enjoy. Record your notes on the worksheet Game builder: breaking down gameplay. 3. Explain the basic rules of the game. For example, a racing game might require you to drive through certain checkpoints, or impose a time limit. 4. Describe the game mechanics in terms of movement, what can you do in the game? In a racing game, for instance, you can accelerate, brake, steer.

	5. List the methods required to experience and succeed in the game. In a racing game, you balance speed with steering, and maybe contact other vehicles to get ahead or avoid contact as it will destroy your vehicle. 6. Describe the appeal of these game mechanics and the gameplay. How challenging is the gameplay? What is fun or different about it? ACTIVITY: **Make a simple game** 1. Create a game from a single sheet of paper. You can pair up or be in a group and use the Game builder: paper game testing worksheet to record your discoveries. 2. You can cut up, fold, tear, scrunch the paper, do whatever you need to do. You can use the surrounding environment as a base, or platform or even a goal for your game. 3. Decide how players interact with the game. What is the aim of the game? What are the rules and other mechanics? 4. Share your game with another group, explaining its goal and the rules. You don't have to describe methods of gameplay, instead, wait to see if they use tactics you didn't anticipate. They might try to break the game or discover 'bugs'. 5. After the gameplay, discuss your findings. Did players uncover unexpected methods of play? Did they try to break the game? Did you have to explain more rules? 6. Ask them what they thought of the mechanics and the gameplay. Could the mechanics be tweaked to make the gameplay more enjoyable, or challenging? Were there any bugs that meant the gameplay couldn't be achieved? 7. Games need to be tested a lot during development. From your observations of players, produce a new version of your game. Iron out any bugs, and enhance gameplay where necessary.
Reflection activity	What worked and what didn't? What surprised you? What would you do differently next time?
Modification	This game could be adapted to use whatever you have around a classroom.
Student Homework/Further Work	
Can you create a game without objects? Using only words? Only gestures? What else could you make a game out of?	

LESSON 3 SEQUENCE - CREATING NARRATIVES	
Student outcomes & Learning goals	In this lesson, students consider how narrative can enhance the engagement of a videogame and come up with their own stories for existing retro games. From their, students can start forming their own take on and narrative outline for their version of Paper, Scissors, Rock. **By the end of this lesson students should be able to:** Understand how story and characters make games more engaging Understand the concept of stakes and player consequences **By the end of this lesson students will have:** Developed narrative ideas for existing retro games Developed a simple narrative for own version of Rock, Paper, Scissors (or other if the teacher decides)
Prep & introduction activities	*Setup screen to watch videos, and paper and writing materials. Print any worksheets linked below.* Games are more engaging if they involve a story, or narrative. Through story you can add stakes to your player's actions. By stakes we mean consequences for the player. So, does the fate of the planet rest on the result of the Paper, Scissors, Rock game? By including a narrative, you might also think of interesting story moments or dialogue that could be written into the gameplay. If there's a non-playing character or opponent in the game, you could give them a backstory and dialogue. In the Paper, Scissors, Rock game we're creating, we are going to have a beginning, middle and end to our story. The beginning involves the information the player learns at the start of the game, the middle is the gameplay, and the end is the resulting win or loss. Hear from ACMI staff (2 min watch) as they talk about their favourite video game storylines, and think about any game storylines you've really loved. THINK: Think about a game you really love for its storyline and what you like about it. Who are the characters? What are the stakes? What happens in the beginning, middle and end that makes it so engaging? What is the role of the player in driving the narrative? PAIR: Discuss in pairs the similarities and differences between your two games. SHARE: Brainstorm any games that do something unusual with narrative structure. Do they play with time, choice, or character perspective?

Main lesson activities	Here are two classic games that don't have much of a storyline: Play each game online: Tetris. Pong. ACTIVITY: **Imagining storylines for these simple games** You may find it helpful to use the Game Builder Story and Character worksheet. BASICS: ● Thinking about Tetris or Pong, Get imaginative. What could be the backstory to these blocks flying around? ● What happens when a player wins? ● What happens when a player loses? CHARACTERS: ● We love memorable characters like Mario. Think about the kinds of characters who might be involved in your storyline ● Focusing on either your Tetris or Pong story, create characters for the game. Who are they? Why are they here in your story? What have they got to lose or gain? STORYLINE: ● Now that you know who's involved, flesh out the beginning, middle and end. ● What information is communicated to the player at the beginning, before the gameplay starts? ● What information will the player get during gameplay, eg. when players score or take hits? ● What information will the player get at the end, when they win or lose? **Your Game** Use the Paper, Scissors, Rock example and create a story around the gameplay – what has taken place before the gameplay starts? What happens to the winner or loser? Could you change the game story and the world by changing the objects? For example, a science-fiction version might keep the original rules of the game but involve a Force Field Shield, Laser Knife, and Ion Cannon.
Reflection activity	Share your Paper, Scissors, Rock stories so far. What is the beginning, middle, and end? Brainstorm as a class if there are any gaps. Can you get inspiration from your fellow Game Makers to help give your game a unique twist?
Student Homework/Further Work	
Continue with the Story and Character worksheet for the Paper, Scissors, Rock game.	

LESSON 4 SEQUENCE - GRAPHICS AND ARTWORK	
Student outcomes & Learning goals	In these lessons, students consider the differences between sprites, costumes and backdrops in regards to the coding program Scratch. Students will develop concept artwork that considers the role of colour in videogames, before using Piskel to build their asset library. It's anticipated it will take students at least two lessons to design, create and export all their sprites, backdrops and costumes. **By the end of this lesson students should be able to:** Understand the difference between sprites, costumes, and backdrops Understand the role of concept artwork and colour palettes in designing artwork for a videogame **By the end of this lesson students will have:** Used Piskel to create simple sprites for videogame Created concept artwork and colour palettes pre-sprite design Completed sprites for their videogame
Prep & introduction activities	*Setup screen to watch videos, and computer devices with internet access for students to use.* **Words To Know** ***Sprites**:* the two-dimensional d*esign for a character or object in a game. Anything that can be interacted with.* ***Costumes**: specific to the Scratch program and belong to your sprites. For example, when your character runs or jumps you'll want a different costume for each different movement.* ***Backdrop**: sits behind your sprites. They paint the world of your game and add mood and atmosphere.* Watch ACMI staff discuss the design and art in their favourite games. (2 minute watch.) ACTIVITY: **Rock Paper Scissors game** Think about the colour scheme of your game. The colours relate to the genre and world of your videogame, and communicate mood and atmosphere. **Concept Artwork** ● Discuss what you want your game to look like. This might be influenced by your story. For example, is the world of your game fun or serious? Do you want your sprites to look realistic, or cartoonish? ● Hand draw, or use a digital sketching program, to produce sketches of the backgrounds and characters for your version of Paper, Scissors, Rock. You could also create a mood board for your project.

	Colour Palettes ● Find a videogame similar to yours, and make note of the colour palette. How do the colours help the overall feeling of the game? ● Try sites such as Colourmind or Coolors to create a colour palette. You can save these palettes for when you create backdrops and sprites, and copy and paste the HTML colour codes into your sprite making software. **Planning** ● Make a list of the characters and objects (sprites) as well as any costumes you need. Keep track of your list as you go. Keep it manageable and don't create heaps of costumes for every sprite.
Main lesson activities	**Create sprites** You can create your sprites in a variety of ways, using Photoshop, Microsoft Paint, or any program where you can produce images. You can also hand draw your artwork and scan it onto a computer. You could even take photos of your own hands to use for the game. We recommend making Pixel art sprites using Piskel, a free, online pixel sprite creator. **Import sprites into Scratch** You are going to import your artwork into Scratch which supports many file types. However, by the end of the process, your files need to be JPEGS or PNG image files. Check out our tutorial on how to use Piskel here. (6 min watch) If you'd like to learn how to animate your Piskels, we recommend this YouTube tutorial. (4 min watch) **Create a costume** A costume is another state a sprite can be in. Let's keep it simple so that when a character loses, they flash red like they are taking damage. In Piskel you can duplicate your sprite, and use the paint fill tool to make the sprite red. Remember to export this costume as an individual image, and label it something obvious like: character1_costume_lose Later when you build your game, you can add code so that your sprite changes costume when it loses. Watch this video about coding costume changes. It also gives you a visual explanation of what costumes are, and how they work. **Create a background** Once you've created your sprites and imported them into Scratch, you can design your background. Here is a tutorial video that shows you how. (3 min watch) Once you have created your backdrops, sprites and at least one costume, move onto the next lesson.

Reflection activity	THINK: About the process from planning your character, to making it and importing it. Were there any challenges? PAIR: Discuss in pairs what you found easy or difficult, and how you overcame the challenges. SHARE: If you had to start again, what would you do differently?
Differentiation: modification, extension and inclusion notes	If there is limited time the planning component can be simplified.
Student Homework/Further Work	
Continue with creating and importing your sprites.	

LESSON 5 SEQUENCE - LOGIC AND CODING	
Student outcomes & Learning goals	In this lesson, students consider what coding is through listing coding actions for a real-world game. Students will then be introduced to object-orientated coding and complete further coding lists for their own videogame project. **By the end of this lesson students should be able to:** Understand what coding is Understand what object-orientated coding is **By the end of this lesson students will have:** Applied logic to basic coding lists, including an object orientated list Created a coding list for their game Created object-orientated programming list for their game
Prep & introduction activities	*Set up paper and writing materials, and print worksheets below.* Coding is a list of instructions you give to a computer software so it can run a program, a game, or even display a website. Let's start with the game you created earlier using a piece of paper. To replicate that as a videogame, you need to use the specific coding instructions known as syntax. Syntax means the rules for how coding is written -- the words and symbols used and how they're ordered. Different coding languages have different syntax, but let's begin by focusing on the thinking behind coding. We want you to think about the Paper, Scissors, Rock game. Instead of verbally describing the rules, write out a list of instructions for how to play the game. EXAMPLE: **Hide and Seek** 1. Choose a player to 'seek' - they are the seeker. 2. Seeker closes their eyes 3. Seeker starts counting down from 60 – they cannot open their eyes or move until the 60 seconds is up 4. All other players, the 'hiders' search for somewhere to hide 5. Hiders assess the quality of hiding spots 6. Players choose hiding spots, and hide 7. Once Seeker reaches '0' second, they say 'ready or not, here I come' loudly 8. Seeker starts searching 9. They search locations and find the hiders 10. The last remaining hider wins.

	Now do the same for Paper, Scissors, Rock in the left-hand table of the Game builder: logic and coding worksheet. The list is not in coding language a computer will recognise but is a little closer to how a computer would understand the game's operations.
Main lesson activities	**Object Oriented Programming** When you use Scratch you can code to objects. These objects can interact with and react to one another. This is called object-oriented programming, and it just means you can give instructions to each sprite within the game, as well as backgrounds. This kind of coding, or programming, is great because it means the code the computer reads doesn't have to be in a particular sequence or order and can read instructions coming from all the objects in the game at different times. EXAMPLE: Here is the list of instructions broken down and split up so they're applied to the different objects, in this case, players, in the game. Object = seeker 1. Start game 2. Close eyes 3. Count backward from 60 4. When reaches zero say 'ready or not here I come' 5. Can move in all directions and search 6. If they find player say: 'found you!' Object = player 1. Start game 2. When seeker closes eyes and starts counting, search for a hiding spot 3. Enter hiding spot 4. If seeker reaches '0' and no hiding spot has been entered player loses. 5. If discovered by seeker, game over 6. If all other players are found, the player wins, and stop the game. Working with your partner or team, identify where the instructions for the different objects affect one another and the game. How could you adjust the mechanics of the game to make the gameplay different? Because we'll be making our own version of an existing real-world game, it is a good idea to think about altering the mechanics of the game to make it unique. PLAY: A game can be made very different by changing the mechanics. Try out this version of Pong where instead of controlling the paddles, you control the ball. How is the gameplay different to the original?

	ACTIVITY: **Your game** Using the right-hand section of the Game builder: logic and coding worksheet you started earlier, create an object-oriented coding list for the version of Paper, Scissors, Rock you're developing. Try to stamp your own personality on the game mechanics and gameplay.
Reflection activity	What surprised you about this process? Why is it so important that the instructions are so specific? Can you think of a game where the rules were not clear? What was the outcome?
Student Homework/Further Work	
Continue filling out the object-oriented coding list for your Paper, Scissors, Rock game.	

LESSON 6 SEQUENCE - CODING IN SCRATCH	
Student outcomes & Learning goals	In these lessons, students build or enhance their block-coding skills using Scratch, and build their game. Before building their own game students will identify the coding knowledge they will need and identify any gaps in their knowledge. They will then seek out further tutorials or instructionals before starting the build of their game. It's anticipated familiarising or refamiliarising themselves with Scratch, and building their game will take at least two lessons. **By the end of these lessons students should be able to:** Understand a variety of Scratch blocks and their uses Understand the actions of their game and the coding skills they'll need in Scratch **By the end of these lessons students will have:** A consolidated list of assets before learning Scratch A list of actions and coding skills Identified Scratch tutorials they should complete in order to be able to code their game A rough version of their game
Prep & introduction activities	*Setup computer devices with internet access for students to use, screen to show videos, and print any worksheets below.* We're going to block code using Scratch. Block coding means that instead of typing out all the code and instructions for a computer program, Scratch has the instructions built into blocks that you can click, drag, drop, and combine with other coding blocks to create a game. Many of these blocks require inputs from you. We highly recommend learning how to use Scratch before developing your game. Visit the Scratch website and make sure you have an account set up for yourself, or your group, to use. There are lots of different coding blocks to learn about, and we recommend this video playlist which explains a lot of them in sixty seconds. These videos were made for Scratch 2.0, and there is now Scratch 3.0 but a majority of the blocks have remained the same. Have Scratch open as you watch these so you can practise and experiment as you go. Scratch also has a number of tutorials to go with the new 3.0 version which was released January 2019. Check out the list and pick four tutorials to work through, based on the mechanics you have planned for your game. Keep a list of the ones you have completed and do some reflection with your group members on the new Scratch skills you picked up.
Main lesson activities	Now you have an understanding of what you can do with Scratch, it's time to think about applying that understanding by creating a list of coding actions for your game. EXAMPLE: **Paper, Scissors, Rock**

We've done it for our version of Paper, Scissors, Rock, but yours should be different.

1. Game starts.
2. Computer player and player 1 enter.
3. Computer player challenges player 1.
4. Player 1 sees three options, paper, scissors and rock.
5. Player 1 clicks one option using the mouse.
6. Computer player randomly selects paper, scissors or rock.
7. Icons for each player's choice are shown.
8. If player 1 wins they say 'I win' or if computer wins they say 'I beat you!'
9. If draw both players say 'it's a draw'
10. Each time someone wins a point is added to their score
11. The first player to reach 5 points wins
12. Player who wins says 'I win!'
13. The losing player explodes.

From this list, we get an idea of some of the coding required. The mouse will be used as a controller, and we need to code in the dialogue and some random selections for the computer player. We need to do some coding so that the program recognises who actually wins each round, as well as some variables to keep track of the score.

Once one of the players reaches 5, we'll need the program to recognise this and react with the 'I win' line, and make the losing player explode.

ACTIVITY: **Coding Actions**

Using the worksheet Game builder: Scratch game worksheet, write up a list of coding actions for your version of the game. Try to keep the list to around 15 – 20 actions. Any longer and your game might get too big.

Afterwards, go through your list and brainstorm some of the coding you will need to do in Scratch.

From that list, break down your coding into 'I/we know how to:' and 'I/we still need to learn how to:'

Before you make your game, see if you can find out as a team how to do the things you haven't learnt yet. Return to the tutorials list of Scratch and see if you can fill the blanks.

Make Your Game

Using your list of coding actions, go step by step and add code to the sprites you already have imported in your game.

You can watch the tutorials alongside each step to refresh your memory.

Reflection activity	Play our version of Paper, Scissors, Rock below. What features of the game do you like? Are there any you had planned to include in your game? Remember, you can always look 'inside' Scratch projects. If you'd like to see the code for this project you'll find it here. If you notice some problems with the game, make a note of them for the playtesting lesson. Play our buggy version of Paper, Scissors, Rock on Scratch
Student Homework/Further Work	
Continue building your game and refining your code.	

LESSON 7 SEQUENCE – SOUND DESIGN	
Student outcomes & Learning goals	In this lesson, students consider the role of music and sound effects on the mood and genre of videogames. They'll pick up some basic music theory before creating their own piece or pieces of music for their game. They'll also consider sound effects and how to make original sound effects. Once their audio assets are complete, students will code the into their game build. **By the end of this lesson students should be able to:** Understand what sound design is, and the relationship between music, sound effects, and mood/ genre of game Understand basic musical theory, including beat, tempo and musical keys How to use Beepbox to create original music **By the end of this lesson students will have:** Completed planning sheet outlining music and sound effects One or more pieces of music made specifically for their game that shows an intent to link to the game, either thematically or in terms of pace or gameplay Import music and sound effects and code into game build
Prep & introduction activities	*Setup computer devices with internet access for students to use, screen to show videos, and print any worksheets below.* Watch ACMI staff discuss sound design in their favourite games. **Introduction** Sound design is an important part of any videogame. When we refer to sound design, we're talking about music and sound effects, and how they work together to enhance the gaming experience. Music and sound effects contribute to the tone and feel of your game, and should aim to reflect elements of the game itself. If your game is all about adventure, the music should reflect that. If your game is retro-looking then maybe some 8 or 16-bit music might suit it? If your game is set during a certain time, you might create music that sounds like it is of that time. Your sound design should complement something about your game. Perhaps the music can be fast-paced to match the fast-moving gameplay you've created, or eerie to match the dark setting of your monster-fighting game. It's up to you, but consider how it can enhance the player experience. **PLAN:** Now your game is functioning, think about the different pieces of music you need, and where they will be used or activated within your game. Using our example, we decided we want three pieces of music. One long looped piece for the general gameplay, one for if the player wins, and one if they lose.

	The last two pieces will be short as they'll be playing at the conclusion of the game. We've mapped out where it will be used (this is helpful as you'll know where you'll need to code the music), and the required length of the music. Open the worksheet Game Builder: planning music and sound effects, and complete for your game. ACTIVITY: **Creating chiptune music** You can use programs like Beepbox to create your own chiptune music, also known as 8-bit music, even if you don't play an instrument yourself. You might recognise this type of music from older videogames, so your game will have a cool retro feel about it. **Check out our instructional video on how to use Beepbox** here. Make 'General Gameplay' music - this piece should be quite long, so it can play throughout the game. You can create loops in Beepbox so don't worry about having to compose a really long piece of music. Make 'Win' and 'Lose' music - this music should be quite short. Before you make your music, think about things like: Tempo – do you want your music to be fast-paced, plodding, or somewhere in the middle? The tempo (similar to speed) will affect how the player experiences the game. Key – a key is a series of musical notes that work together. You can experiment with different keys in Beepbox, as you might find a key such as the default key of C sounds nice and upbeat. You might find other keys sound darker, or gloomier, such as G minor. Here's a great breakdown that associates different musical keys with different emotions, feelings and tones. **BONUS: Sound effects** Scratch has in-built sound effects but we suggest creating your own because it's fun and makes your game feel more original. Check out the footage from Llama That Wandered a Bit Too Far, winner of the Junior Game category for Screen It 2018. You can also download the game, as well as other winning Screen It videogames.

Main lesson activities	**Creating your music** PREPARE: ● Go through your game and create a list of all the sound effects you could create. Focus especially on actions within the game. GATHER YOUR MATERIALS: ● It's fantastic if you have a microphone and a computer program to record into. If not, you can use voice record applications on a smartphone or tablet. Scratch also has an option to record directly into the program via your computer. RECORD ● Record your sound effects in a controlled environment. This is a quiet space free of noise you can't control. Make sure you review and edit or trim your sound clips, so there's no unnecessary space or noise at the beginning or end of your audio clip. ● Check your worksheets and see if you have all the music and sound effects for your game. IMPORT: ● Import all your music and sound effects files into your Scratch build, then code them all in to play at the appropriate points of your game.
Reflection activity	**Playtesting and improving** Remember in lesson two when you made a game out of paper and tested it out? The point of that exercise was to identify any bugs with the game and any issues with the mechanics or gameplay. Now that you've coded a game, and included all your artwork and audio elements, the chances are it will have some bugs. They might be large bugs that prevent the game from working at all, or they might be minor bugs like a costume or sound effect not working as they should. Either way, you need to identify them and work to solve them. To do this you'll need to do some playtesting. First, you should play the game yourself to identify any problems with the game. Keep a list of bugs or problems you identify, and as a team see if you can fix them. If you can't solve within your team you might have to get some advice from other groups. After you've fixed the bugs you identified, playtest with others. Choose people who haven't been involved in the creation of the game. Let them play and merely observe. Don't interrupt and tell them how to play, or ask or answer questions. Just observe and make note of any bugs you notice, or moments where the player seemed confused, unsure of what to do, or got stuck. Afterward, go through your notes with the player and ask them questions about their experience. Take your list and make some final tweaks to the gameplay.

	Congratulations! Once you're done, time to celebrate! Share your finished product with other groups, and play each other's games.

Student Homework/Further Work

Get some practice by playtesting our version below. Have one or more member of your group play the game whilst others observe and take notes on issues and glitches.

Don't forget, you can see inside the project and even copy it and edit the coding yourself. After identifying bugs, have a go at fixing some of the issues yourself.

Access the project here.

Ready for your next challenge? Move on to Game Builder: Level 2.

LESSON 8 SEQUENCE – PLAYTESTING & IMPROVING	
Student outcomes & Learning goals	Once a rough version of their game had been built, students will playtest their games and make notes. Following playtesting, students create a list of possible edits, alternations or additions to their game as they see fit. **By the end of this lesson students should be able to:** Know how to conduct a playtest session that will result in meaningful observation for their game **By the end of this lesson students will have:** Produced a list of observations from playtesting Devised a shortlist of edits, alternations or additions to their game
Prep & introduction activities	*Setup computer devices with internet access for students to use, screen to show videos, and print any worksheets below.* **Playtesting and improving** Remember in lesson two when you made a game out of paper and tested it out? The point of that exercise was to identify any bugs with the game and any issues with the mechanics or gameplay. Get some practice by playtesting our version below. Have one or more member of your group play the game whilst others observe and take notes on issues and glitches. Don't forget, you can see inside the project and even copy it and edit the coding yourself. After identifying bugs, have a go at fixing some of the issues yourself. Access the project here.

Main lesson activities	Now that you've coded a game, and included all your artwork and audio elements, the chances are it will have some bugs. They might be large bugs that prevent the game from working at all, or they might be minor bugs like a costume or sound effect not working as they should. Either way, you need to identify them and work to solve them. To do this you'll need to do some playtesting. First, you should play the game yourself to identify any problems with the game. Keep a list of bugs or problems you identify, and as a team see if you can fix them. If you can't solve within your team you might have to get some advice from other groups. After you've fixed the bugs you identified, playtest with others. Choose people who haven't been involved in the creation of the game. Let them play and merely observe. Don't interrupt and tell them how to play, or ask or answer questions. Just observe and make note of any bugs you notice, or moments where the player seemed confused, unsure of what to do, or got stuck. Afterward, go through your notes with the player and ask them questions about their experience. Take your list and make some final tweaks to the gameplay. Congratulations! Once you're done, time to celebrate! Share your finished product with other groups, and play each other's games.
Student Homework/Further Work	
Ready for your next challenge? Move on to Game Builder: Level 2.	

Game Builder was made possible with support of the Telematics Trust.

ACMI Game Lessons acknowledges the support of the Department of Education Training, Victoria, through the Strategic Partnerships Program.