

Game lesson title	Game Builder - Level 2		
Brief intro	<p>Designed for students with some experience to help take their skills to the next level.</p> <p>This unit will take students through the more advanced aspects of making a video game including:</p> <ul style="list-style-type: none"> • Detailed ideation and structuring methods • How to create complex systems and code sequences • How to plan workflow, process and build your own game • Building videogame soundscapes and playtesting • Next steps in game making 		
Lesson hook	Advance game making - the perfect toolkit for students wanting to develop their game from ideation to final playtesting using more complex game engines.		
Suggested year level	5-10	Suggested age level	10 - 16
Suggested duration	8 x 45 min sessions (Students may need more time to complete more complex sections, at teachers own discretion.)	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			
<input checked="" type="checkbox"/> [subject] 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 VCDTCD031 - Design a user interface for a digital system, generating and considering alternative design ideas			

[VCDTDI039](#) - Manage, create and communicate interactive ideas, information and projects collaboratively online, taking safety and social contexts into account

[VCDTCD053](#) Develop modular programs, applying selected algorithms and data structures including using an object-oriented programming language

[VCAMAE040](#) Experiment with ideas and stories that manipulate media elements, and genre conventions to construct new and alternative viewpoints in images, sounds and text

Game/Program Used	<p>To build a game using this resource you'll need access to a game engine. We recommend Game Maker Studio 2, which has a free trial. Unity is also a good option, and teachers can apply for free Education licences. If you're needing to use a browser-based program we recommend Game Froot.</p> <p>Here are some free, accessible programs we recommend to make the music and artwork for your games:</p> <p>Beepbox: an online chiptune music creator. It's free and browser-based.</p> <p>Garageband: even non-musicians can create complex musical loops that can be used in gaming. Garageband is free on Apple iMac and Macbooks</p> <p>Piskel: a simple browser-based image editor that you can create pixel characters and objects with. You can also create gifs and animated sprites using Piskel</p> <p>Bfxr: a program to create original sound effects. Can be used online (needs Flash player) or downloaded for free.</p> <p>Leshy SFmaker: a program to create original, old-school sound effects</p>
Game play required?	No

CLASSIFICATION	CONSOLE	INTERNET REQUIRED?	COST (RRP)
N/A	PC or Mac computer	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.

Prior knowledge/skills (Required/Recommended/References)

This resource is designed to deepen the skills of more experienced students. We recommend it for students who have completed Game Builder Level 1 or have previous coding or game-making experience.

Student outcomes

Learning goals

KNOW: Advanced methods of game design including: ideation, planning workflow, complex structure and design, and advanced coding, audio, and graphics.

BE ABLE TO: Successfully plan, structure, and create a fully detailed video game. Create detailed process records, and approach game making with a producer mindset.

IMPROVE: Planning and communication skills, production skills, research and troubleshooting skills, as well as technical game making ability.

LESSON 1 SEQUENCE - IDEATION	
<p>Student outcomes & Learning goals</p>	<p>This lesson gives students a framework for developing ideas as well as tools for generating their own.</p> <p>By the end of this lesson students should be able to:</p> <ul style="list-style-type: none"> • Understand the role of a foundational premise / idea in game development • Be able to critically engage and explore questions of premise / idea development in games they have played • Have a set of foundation ideas for the development of their own platform game
<p>Prep & introduction activities</p>	<p><i>Set up screen to watch videos. Paper and writing materials needed.</i></p> <p>Ideas for games can come from anywhere. You might have something completely original that nobody else has done before, or you might take something you know really well and add a twist to it.</p> <p>Good ideas pose questions. What would it be like to control time? What would an interactive western look like? What if we could build bases and control territories?</p> <p>By working through those answers, you will come up with more new ideas and things to experiment with.</p> <p>Hear from an Australian videogame maker Grace Bruxner talk about how she comes up with ideas for her wonderful, quirky games. (2 min watch)</p> <p>After watching the interview, discuss where your ideas come from, and the kinds of things you draw inspiration from when thinking about creating videogames.</p> <p>Thinking about premise: Think about a game you love.</p> <p>What do you think the initial idea, or premise, for the game was?</p> <p>Sometimes this might be about a thing that the player can do, a world they want to explore, or a character.</p> <p>For example, a game like Portal is cleverly designed, has an interesting antagonist, but most likely started with the premise: what if there was a gun that could create portals?</p> <p>To help prompt ideas for a game, consider the following questions and fill in the blanks. This could spark an idea for different games, or you could answer all for one premise.</p>

	<p>ACTIVITY: Imaginative brainstorm</p> <p>1. What would _____ look like as a game? This might be a location or something from a story, like the Wild West or Space Pirates.</p> <p>2. What if the player was a _____ ? This might be about the player’s abilities, such as a detective or a race car driver.</p> <p>3. What if the player could _____ ? This would be about the types of things players can do in the game; travel through time, have superpowers etc.</p> <p>4. What if the game’s setting was _____ ? Perhaps it’s a haunted house, or a shutdown spaceship orbiting a black hole.</p> <p>5. What if we wanted the player to feel _____ ? Do you want them to feel frustrated? Powerful? Quiet? Loud?</p> <p>One way of working through your idea for a game is by asking these questions at the start of making it, and also while you’re in the thick of making it.</p>
<p>Main lesson activities</p>	<p>Different platformers</p> <p>Create a list of all the types of platform games there are out there and discuss as a group any similarities or differences amongst the games on your list.</p> <p>One way we can think about the elements of these games is to write down their similarities and differences as a list of opposites. Some of these might be:</p> <ul style="list-style-type: none"> ● Slow—Fast ● Single Screen—Scrolling ● Vertical—Horizontal ● No items / pickups / power ups—items / pickups / powerups ● No story—story <p>Take the list of opposites from above, find a dice, and for each pair of opposites roll the dice. For odd numbers put a circle around the left-hand side, for even numbers put a circle around the right.</p> <p>You should end up with a description of your game like: A fast, scrolling, horizontal platform game with no items and no story. Or: A fast vertical scroller with no items and a story. What did you end up with? Whatever you get can form the parameters of the game you’ll work on.</p> <p>Setting and character</p> <p>Now you have a skeleton for your game, think about where you’re going to set it, and how that setting relates to what the player will be doing, the tone or genre of the game. This can be anything, but some of the questions you should ask are:</p>

	<ol style="list-style-type: none"> 1. What games have settings that I like and would want to explore in my own game? 2. What books / films / comics / TV shows are set in worlds I like, or have characters I like? What do I like about them? And can I bring those into my game? 3. What sort of player character might exist in my type of game? <p>Using the above questions, write down as many ideas as you can, then mix and match your various responses until you find something you're excited to work on.</p> <p>EXAMPLE: Throughout these resources, we will develop a fictional game to provide you a working example.</p> <p>For this example, we will develop a horizontal scrolling, slow platform game with no items and a story. For our setting, we're going with a jungle adventurer ala Indiana Jones, Tomb Raider, or Pitfall. So we'll draw inspiration from those games and films.</p> <p>Our player character will be some sort of adventurer, treasure hunter or archaeologist (or all of the above).</p>
<p>Differentiation, extension and inclusion notes</p>	<p>As a next step, find an example game which has most, if not all of the attributes you're working with. Play that game if you can, discuss what works and what doesn't. This game might also provide a point of inspiration and help you problem-solve issues you face during the development of your game.</p>

LESSON 2 SEQUENCE - STRUCTURE AND DESIGN	
<p>Student outcomes & Learning goals</p>	<p>This lesson looks at player actions, and how they evolve over space and time.</p> <p>At the end of the module, students should be able to:</p> <ul style="list-style-type: none"> • List the actions that the player can do • Map out the spaces of their game • Show how their game spaces provide interesting sequences of interaction and playability
<p>Prep & introduction activities</p>	<p><i>Set up screen to show videos, and print any worksheets below. Paper and writing materials needed.</i></p> <p>Actions and design</p> <p>Games are built around actions that the player can do: run, walk, jump, shoot, and spaces where the player performs those actions. How we structure those actions and create gameplay is a key part of designing a game.</p> <p>In-game actions begin outside of the game with something that the player physically does. This might be pressing buttons on a controller or a keyboard, moving a mouse, or waving their arms if they're playing something in VR. These physical movements get transformed into something that happens in the game: press a button and a character jumps, move the mouse and you look around.</p> <p>Once you know about your individual game actions, consider how they will work together in sequence and the player's experience of performing those actions.</p> <p>Doing the same thing over and over can be boring, so mix it up and provide different objects to interact with, different levels that make you move in different ways, and a narrative that engages the player throughout.</p> <p>Your job as a designer is to hold all those things in your head, and always be asking: what do I want the player to do now?</p> <p>Check out the analysis of the level design from game Super Mario Bros. (6 min watch) This will help get you thinking about player actions and sequences, and how they relate to level design and narrative.</p> <p>Actions and objects</p> <p>Write down a list of all the actions the player character can perform in your game - these are your game's verbs. This might include things which the player themselves doesn't directly control - in a platform game, these could be 'land', 'skid', anything.</p> <p>Next, write down a list of all the objects you'll find in your game. Include things which the player might not interact with - clouds or any other background visuals - as well as the things the player will interact with. This might be 'platforms', 'spikes', 'goombas'.</p>

So in our example game, the player character's actions are:
 Run + Jump + Swing + Duck
 ...and the objects are:
 Platforms + Logs + Snakes + Tree branches

Finally, create a table like the one below with your player actions along the side and the objects along the top.

	Run	Jump	Swing	Duck
Platforms	Player runs along platforms	Player jumps onto and off platforms	Player can swing onto platforms	Player can duck below platforms
Logs		Player jumps over logs		Player can duck below swinging logs
Snakes		Player jumps over snakes or knocks them out with a whip	Player can swing onto snakes and knock them out	
Tree branches			Player can swing from tree branches	

For each cell in your table, write down what the interaction is. This table describes the interactions between actions and objects in your game. There will be gaps, and it's ok to have repetitions.

Main lesson activities

Sequencing

Get dice and paper. Now that we've seen how objects in your game interact with each other, we're going to look at how they might be organised to create a sequence for a level.

1. Roll your dice. This will be the number of objects in your sequence.
2. Assign a number 1-6 for each object in your list. Roll a dice once for every object in your sequence from 1). For each number that comes up, find the corresponding object from your list and make note of it.
3. Repeat at least four times and put all your info into a table like the one below. We're going to use this to help us design a gameplay sequence.

6	Platform	Platform	Log	Snake	Tree branch	Spike pit
3	Spike pit	Spike pit	Platform	-	-	-
4	Log	Snake	Platform	Snake	-	-
1	Log	Platform	-	-	-	-

The number in the left-hand column is the number of objects we rolled, with each object in the sequence also assigned randomly through rolling dice.

1. Illustrate the sequences from your table, adding notes along on what the player does and how the objects might interact with each other. You might like to see how levels were designed for Mario Bros. back in the day [using graph paper](#).
2. Repeat steps 1-4 until you have a pile of illustrated sequences.

Next, go through your pile and think about where they might come in your game. Are they easy or hard? Do they introduce a new object that players should learn about? Do they repeat things that the player has already done? Now order your sequences to create levels, considering the difficulty arch and overall experience of gameplay spaces. Feel free to put aside sequences you don't think will work.

**Differentiation:
modification,
extension
and
inclusion
notes**

You might like to get some inspiration for sequences, level design and player actions [from this presentation by the developers of Celeste](#). (31 min watch)

Itching to program some sequence right away? [PuzzleScript](#) is an open-source HTML5 puzzle game engine.

Try out some games, check out some simple game sequences and levels, then look at the First Steps section before coming up with your own.

LESSON 3 SEQUENCE - NARRATIVE DESIGN	
<p>Student outcomes & Learning goals</p>	<p>This lesson looks at story structure and the relationship between narrative and gameplay.</p> <p>By the end of this module, students should be able to:</p> <ul style="list-style-type: none"> • Describe the key events or acts of their game • Describe the key beats of their story if they have one • Show how their narrative and existing level designs coalesce
<p>Prep & introduction activities</p>	<p><i>Set up a screen to show videos. Paper and writing materials needed.</i></p> <p>If you are making a game with a story, that story can help you think about the player’s actions, how they interact with objects, and when to introduce new pieces of gameplay.</p> <p>You might also think about characters and what they say along the way. Maybe you have to rescue someone, someone gives you a quest, or they give you an item that endows you with a new ability.</p> <p>The story, with its beginning, middle, and end, can help you think about the game structure.</p> <p>We use the term narrative design rather than narrative writing because narrative and gameplay are linked in videogames. Considering how the player will experience the story and the gameplay is at the heart of narrative design.</p> <p>There are different views about when the narrative should come into the development of a project. For your project, you will first decide on a genre and setting for your game and have ideas for your player character, the game mechanics, player actions and game objects.</p> <p>Before going any further, flesh out the story of your game in relation to the levels and sequences you've roughly designed. This video outlines the relationship between narrative and game mechanics. (6 min watch)</p>
<p>Main lesson activities</p>	<p>Thinking in acts</p> <p>One way of thinking about a game’s story is in acts. An act is a series of actions, in our case, maybe bits of gameplay, or levels that build to a climax, and then send the characters and player into the next act.</p> <p>While stories can have lots of different numbers of acts, it’s useful to think of three to four acts; your beginning, middle, a twist/fake ending, and the end.</p> <p>Story can help you to think about where the player’s actions go, how they interact with objects, and when to introduce new pieces of gameplay. You might also be thinking about characters and what they say along the way. The amount of narrative you decide to work into your game is up to you; these are just suggestions.</p>

Act 1: Here you introduce the player's actions and characters.

Act 2: Give the player a chance to do some bigger things with their actions. The game isn't difficult yet, but it's getting harder.

Act 3: Here things get quite difficult. The player should be tested and the story moving towards a confrontation or climax. When that's reached, a final change sends the player into the final act.

Act 4: The final act of the game. Things are very difficult here, and the player must summon all the skills they've acquired throughout the game to complete it. At the climax, they must learn what they set out to learn and overcome any obstacles put in their way.

EXAMPLE:

For our example game, 'Stowe's Adventure', the main character is Jennifer Stowe, an adventurer and archaeologist searching for her friend Evelyn Quinn who went missing in a faraway jungle.

Act 1: Jennifer lands in the jungle, finds a camp and a map that Evelyn left behind. Jennifer heads to a location on the map, finds Evelyn and an idol. The two set off to find more idols.

Act 2: The two head into the jungle, making it through traps and puzzles. They find a cave and come across mercenary collector Archer Adley. They stop him stealing idols but he gets away and they are trapped in the cave and must escape.

Act 3: Jennifer and Evelyn pursue Archer. They reach a temple but Archer has beaten them to the next idol. Archer sets off a trap and falls to his doom. Jennifer and Evelyn flee.

Act 4: The two race through the jungle and encounter even more difficult traps and puzzles. They make it to their plane to find... Archer! They battle Archer to reach the plane first, eventually, he slips into quicksand. They reach the plane just in time. Do they decide to save or leave Archer to his fate?

[Hear from Brooke Maggs](#), an Australian narrative designer discussing the game The Gardens Between and consider the characters, mood and story of the game. (2 min watch)

Now it's your turn.

For each act, write down three 'beats' - beats are just moments or happenings within an act. This might be: learning a new skill or meeting a character. Think about how these fit together to build towards your climax. You will end up with twelve story beats that you can fit together with your levels and sequences.

Take Act 1 from Stowe's Adventure for example. Earlier we summarised the act, but here it is again with three more detailed beats:

Beat 1: Jennifer lands in the jungle on the edge where it's not so dangerous, to begin with. She begins her search, avoiding some minor threats in the jungle.

Beat 2: Halfway through this act Jennifer finds a camp and a map that Evelyn left behind with three numbered marks on it.

Beat 3: Jennifer heads towards the first one and there she finds Evelyn and an idol. Evelyn reveals she's here to document three idols. Evelyn and Jennifer head out to find the other two idols.

Once you have your twelve-story beats, think about how you're going to tell your story. Are you going to stop the action and have a small cutscene? Or are you going to have characters pop up while the game is in action that share snippets of what's happening?

As a group, refer to your **illustrated sequences and levels** from the last lesson and start positioning and matching them to the story acts and beats you just fleshed out.

You should now have a rough plan for your game's sequences, levels and how the narrative fits within and around them.

Before moving onto the next lesson, have an honest discussion as a group as to whether the plan for your game is too big. You can always consider proto-typing the first act or level of your game to begin with.

LESSON 4 SEQUENCE – SYSTEMS & CODE	
<p>Student outcomes & Learning goals</p>	<p>This lesson looks at problem solving and converting game designs into code, including breaking down the actions and rules of the game into component systems & functions, and also thinking about the parts of a game that you can't see – drawing sprites, handling collision, taking input, developing an understanding of how they interact with what students are making.</p> <p>By the end of this module, students should be able to:</p> <ul style="list-style-type: none"> • Break their game actions into pseudo code / flowcharts • Describe the functions the engine provides & how they interact with their game & code
<p>Prep & introduction activities</p>	<p><i>Set up computer devices with internet access for students to use, and screen to show videos.</i></p> <p>Converting to code</p> <p>To become a videogame, the actions and rules of your game need to be broken down into individual steps and converted to code that a computer can understand. Luckily, there are lots of languages and engines like Gamemaker Studio 2 and Unity which handle all the behind the scene stuff. You'll still need to write the code that is unique to your game though.</p> <p>Each piece of your game logic, actions, rules and so on will be built up of individual pieces of code, and some of that code will communicate with the game engine. If we think about a character jumping, the individual steps for that might be:</p> <ul style="list-style-type: none"> • get the input from the controller or keyboard • start playing the jump animation • start playing the jump sound • apply an upwards force to make them jump. <p>Then there might also be code that checks for when the player lands back on the ground, or miss the ground completely, or hit an enemy. Designing the code for your game is about capturing all of those possibilities and writing code that describes what to do.</p> <p>Some of these things like collision, animation and cameras will be handled by the game engine you've chosen, leaving you free to focus on your gameplay. The more games you make, the more you'll understand what your chosen engine can do and the more you'll be able to make more complex games. You might also move beyond visual scripting to more complex programming languages, which will give you even more options in what you make.</p> <p>Object-oriented coding</p> <p>Originally, code was written linearly in what we now call 'procedural code'. As videogames aren't procedural, programming in most videogames is now object-oriented. Watch this video for further insight into OOP. (2 min watch)</p>

Main lesson activities

Working out what you need from your engine

Before building your game you'll want to know if your game engine is equipped to perform the kinds of events you have sketched out for your game. Consider the outline of your game thus far, particularly actions and interactions, as well as the overall vision you have for the game. Write down what things are happening in the game that could be handled by the engine.

This could be things like:

- Checking if things collide
- Playing music and sound effects
- Handling input from the keyboard
- Drawing sprites and backgrounds

In our platform game Stowe's Adventure, we'll need to think about:

- Reading input from a keyboard, and translating that into Jennifer's movements
- Loading sprites and deciding which ones to play depending on what action Jennifer is performing
- Loading and playing sound effects
- Figuring out what happens when Jennifer hits a dangerous object

Make a list of what features you'll need for your game and which parts of the engine they'll speak to.

Check the documentation of the engine software you are using and ensure the engine can handle what you are planning.

You'll find Game Maker Studio 2's documentation [here](#), and Unity's [here](#). You can see an overview of all the support videos Game Froot have [here](#).

Based on your findings, make a decision as a group as to which game engine you'll use.

Optional: Coding

Before diving into the game engine you'll be using, you might want to brush up on your coding with some unfamiliar engines.

[CodeCombat](#): An amazing introduction to written coding that requires you to write basic code to actually play the game.

[Human Resource Machine](#): Another game that requires coding to play and complete.

LESSON 5 SEQUENCE - WORKFLOW, PROCESS AND BUILDING YOUR GAME	
Student outcomes & Learning goals	<p>This lesson looks at how to map out milestones for a project, and prompts students to prepare themselves for their game build</p> <p>By the end of this module, students should be able to:</p> <ul style="list-style-type: none"> • Show their work plan for their project • Explain how they are placed to build their game, considering their ability with their game engine and understanding of what their game demands of that engine
Prep & introduction activities	<p><i>Set up computer devices with internet access for students to use, and screen to show videos.</i></p> <p>Milestones and workflow</p> <p>Just as each game is unique, the process for making each game is also unique. Some games are made by huge teams hoping to sell millions of copies, and others are made by just one person with humbler intentions.</p> <p>Before you go any further, consider whether you're working alone or in a team. Either way, you or someone in your group is going to have to think like a producer. Ideally you should all be responsible for setting our project milestones and workflow for your project.</p> <p>Unfortunately, we can't guide you through every step when it comes to programming your game, only advise on how to approach your build and point you in the right direction.</p> <p>Watch this interview with Videogame Producer Lisy Kane (Armello) to get an idea of what a producer does, and how you might plan your project. (2 min watch)</p>
Main lesson activities	<p>Milestones are all the key progression points that exist between starting and completing your project.</p> <p>Depending on the size of your group, you might be able to divide the work and certain responsibilities (graphics, sound design, programming) within your team, or you might prefer helping each other across all areas.</p> <p>Either way, it's important to set out your milestones together, decide on a logical sequence of asset delivery, and touch base regularly throughout the project to discuss everyone's progress.</p> <p>You will have already compiled considerable documentation for your game, such as illustrated levels, a list of character actions, objects and interactions, and a narrative. Now it's time to set out the milestones from this point on to completion.</p>

Activity: setting milestones

Set out five clear milestones as a group, decide upon a logical sequence of these milestones, and roughly when they should be completed or achieved.

You'll also need to understand the file types your assets can handle, before exporting your sprites, audio files and the like.

Much of this depends on having an understanding of your chosen game engine.

Getting to know your game engine

It's obviously necessary to be familiar with your game engine, so at some stage, you'll need to work through at least one full-length game making tutorial. These are quite detailed tutorials, and building a platformer even through tutorials might take a considerable amount of time, so factor it into your process.

[How to make a 2D game in Unity playlist](#)
[Game Froot on YouTube](#)

Consider the list of things that happen in your game that your game engine will have to handle. Are you skilled enough with the engine to handle all those 'happenings'? Are there gaps in your knowledge?

If so, find more specific tutorials for your engine.

Once you're feeling more comfortable, you can test your planning and or programming skills by designing a version of an existing classic game, from scratch - without explicit tutorials.

Take the classic games of [Pong](#), or [Snake](#) for example.

Choose and play one of these games, then devise a plan for recreating the game yourself. Brainstorm any sprites, states, movements, and collisions contained within the game. What might the milestones for such a project be?

If you have time, test your skills by building your own version of these classic games.

LESSON 6 SEQUENCE - GAME GRAPHICS	
<p>Student outcomes & Learning goals</p>	<p>These lessons looks at the basics of visual communication and graphic creation including silhouetting, complimentary colour selection and different work pipelines. This will also cover animating sprites and creating sprites.</p> <p>Please note producing their in-game assets might take some time.</p> <p>At the end of this module, students should be able to:</p> <ul style="list-style-type: none"> • Break their game down into different states & show how they connect to graphics • Describe how their game is visually readable – backgrounds, players, enemies – through a combination of silhouetting, colour choice, etc. • Describe how the game communicates changes in state visually – what happens when the player is hit? What happens when an enemy is injured? • Create Sprites & animations which work in their game engine, and which communicate multiple actions and states
<p>Prep & introduction activities</p>	<p><i>Set up computer devices with internet access for students to use, and screen to show videos. Paper and writing materials needed.</i></p> <p>Graphics, meaning and mood</p> <p>As well as being a key part of a games’ world, game visuals also communicate with the player. They tell the player about success and failure, changes in state and the mood of the game or level.</p> <p>The use of colour and shape can also help set the mood. Think about games which are cheerful. What colours do they use? What shapes are the objects? What about scarier games, what colours and shapes do they use? Lastly, do the colours and shapes change as a player progresses? Does the colour scheme reflect the state of the game?</p> <p>These are things players might not be aware of, but the art of the game is working to evoke feelings in the player.</p> <p>More directly, games also communicate changes in player states. This might be obvious, like the screen flashing red if the player is injured, or an enemy flashing white when the player jumps on them, more subtle, like a small cloud of dust when the player takes off from a jump or lands. When games communicate changes in state, they help the player understand the rules and adds to the game's polish.</p> <p>Hear from Ngoc Vu, an Australian artist who has worked on a range of videogames including Necrobarrista. (2 min watch)</p> <p>Shapes and colour</p> <p>Check out this video on colour theory for game artists. It might give you a few more ideas before deciding on colours for your objects and states.</p>

Activity: Using colours

You'll need some paper and coloured pencils for this activity. Go back to your in-game objects and make a new list with three columns. Make each row big and the last column large enough to draw some shapes in.

In the first column, we're going to make some notes about how this object should feel to the player. Think about whether it is hostile, friendly, or neutral; and it is part of the foreground action or the background, quick or slow.

In the next column, think about what shapes and colours represent the in-game object and how you want the player to feel about it. If it's dangerous, should it be spiky? If it's an enemy, how will we tell it apart from the player? What colours might tell the player something is dangerous or friendly?

Use the third column to try out some different ideas. As you go, see if there are common elements you might want to emphasise, for instance, powerups might have different shapes but similar colour patterns, enemies might all have very pointy, sharp shapes so the player can read them quickly and know how to react.

Check out our example table below:

	How the player feels about the object	Shapes and colours	Notes
Platforms	Neutral	Brown, wooden, rectangles	A different shade of brown, maybe brighter than trees
Logs	Neutral	Realistic shape, realistic colours	Could share similar shade of brown to platforms so it's obvious player can use / interact positively with
Snakes	Hostile	Bright, alarming colours that stand out against the earthy colours of neutral objects	Bigger than a normal snake, colours that stand out
Tree branches	Neutral	Natural colours, brown and green	Brighter and bigger than other trees and foliage in background so they stand out. Brown could be similar to platforms and logs.

Main lesson activities

Object interactions and effects

Create a table for all your objects, with objects listed down the left-hand side column and across the top row.

Rather than describe how these objects interact with each other, fill in what effect should trigger when something happens, or the objects interact.

Effects might include:

Sparks, mud, or goo - these are known as particle effects

Objects flashing a colour - perhaps white when they're hit, or red when they're damaged

Full-screen effects - blurs or flashes again, communicating larger impacts to the player

Numbers floating out of something - shows how much damage you've done

Again, check out our table below as an example:

Standing still	Play idle animation. No effect
Running	Play running animation. Trigger small dust effects from feet
Stopping	Play stop animation. Play skid particle effect
Jumping	Play jump animation
Falling	Play falling animation
Landing	Play landing animation. Small dust cloud around feet
Taking damage	Play damage animation. Flash screen red

States

As well as effects when objects interact with each other, there might also be effects that show us changes in the state of the objects, or the game.

States might be simple things like moving left or right, but they can also be more complex such as 'spawning in', 'patrolling', or 'possessed by player.'

Check out [this instructional video](#) on Object States using Game Maker Studio. Although specific to that engine, the video also provides a great overview of object states in games so it's well worth a watch. (22min watch)

For each object in your game, make a list of its states and a short description of what the object is doing in that state.

Then, have a think about whether or not you might need an animation or effect to communicate when that object has moved into that state. Perhaps an enemy

is asleep until the player gets close enough, but then it does a little jump and flashes colour to signal that it is awake and dangerous?

Make a list of possible animation or effects for each state like ours below.

Standing still	Play idle animation. No effect
Running	Play running animation. Trigger small dust effects from feet
Stopping	Play stop animation. Play skid particle effect
Jumping	Play jump animation
Falling	Play falling animation
Landing	Play landing animation. Small dust cloud around feet
Taking damage	Play damage animation. Flash screen red

Create your own

Once you've got a list of object interactions and effects, as well as states and animations/effects, you'll need to create your characters, objects, effects, and states.

Depending on the size of your group and your approach to creating your graphical assets, creating all these assets might take some time.

It is possible to create within engines like Game Maker Studio 2, but you might want to use an external program also. Devise a plan of attack within your group. It's up to you.

We can recommend a great, free pixel-art tool in [Piskel](#) it's a browser-based animation program that's quite easy to use. You can create sprites, animations, states, and particles quite easily.

Here is a [detailed playlist of instructional videos](#) on how to create animated objects and characters using Piskel.

LESSON 7 SEQUENCE - MUSIC & SOUNDSCAPES	
<p>Student outcomes & Learning goals</p>	<p>This lesson looks at the use of audio across a complete game, encompassing different music for the different games states including intro and outro. Audio FX will also be picked apart and stand-out fx use will be highlighted and recreated.</p> <p>At the end of this module, students should be able to:</p> <ul style="list-style-type: none"> • Break down their game’s actions and state, creating a list of sound effects to play on each • Source suitable sound effects & implement them in game. • Source or write music for each of their games high-level states – front end, playing, end screen, boss fights.
<p>Prep & introduction activities</p>	<p><i>Set up computer devices with internet access for students to use, and screen to show videos.</i></p> <p>Sound design for videogames</p> <p>Like graphics, sound design also communicates things about the game, from changes in state to overall mood. Sound effects and music can tell the player things they can’t see or provide additional prompts on success and failure, enhancing the experience and gameplay.</p> <p>So, how do you want the player to feel at each stage or section? The music in the 'easy' level might conjure up images of freedom and expansiveness. In contrast, music for a boss battle might be scary and foreboding. Even if the player isn’t actively aware of it, the sound and music will have an impact, and thinking through what emotions you want players to feel will help you when it comes time to compose your soundscape.</p> <p>This same idea applies to sound effects, which you can use to communicate changes in state and other things that are happening in the game.</p> <p>Hear from sound designer / audio specialist Maize Wallin and their thoughts on designing sound for videogames. (2 min watch)</p> <p>States and audio effects</p> <p>Just as we did with our visual effects, we can map out our games audio effects and use those effects to communicate things to the player. Make a table that has all your objects (including the player) down the left-hand column and along the top row. As with our visual effects, we’re going to go through the objects and fill in what sound effect should be played when they interact.</p> <p>Think about the type of sound you want and how it should feel to the player. Is it a soft, squishy interaction, or is it a harder metallic sound? Is it communicating something positive, like an increase in score, or is it telling the player that they’ve done something wrong? We might also play sound effects that tell us that something in the game has changed.</p>

By layering audio and visual feedback together, we start to create a rich set of interactions that communicate with the player in lots of different ways. Check out our example below:

	Player	Platforms	Logs	Snakes
Player	-	Squishy leaf / dirt sound effect on landing	'Yelp' sound = player damage	'Yelp' sound = player damage
Platforms	-	-	Wood and dirt, smooshy, grindy sound	No interaction
Logs	-	-	-	Hissing sound from snake = damage
Snakes	-	-	-	-

Now, for each object in your game, make a list of its states and a short description of what the object is doing in that state.

Have a think about whether or not you might need a sound effect or effect to communicate when that object has moved into that state. For example, perhaps an enemy is asleep until the player gets close enough, but then it does a little jump and makes a beep sound to indicate it's seen the player and is awake. You can look out our example below to help you get started.

Standing still	No sound effect
Running	Footstep sound effect
Stopping	Quiet skid sound effect
Jumping	Quiet 'boing' or spring sound effect
Falling	Quiet, short 'swoosh' sound effect
Landing	Soft thud on dirt
Taking damage	'Yelp' sound effect

Main Activity

Time to either create or collect all the sound effects assets you need. Your game engine might have some sound effects available for use, otherwise. Otherwise you can check out this [online listing from SuperDevResources](#).

Another way to create simple effects yourself is using [Bfxr.net](#) or [as3sfxr](#). You can even watch a tutorial on how to use that site and bring sound effects into Game Maker [here](#) as well as another general tutorial [here](#).

For some inspiration, [check out this audio analysis](#) of all the sound effects from Limbo.

Music

Find a film or TV show that corresponds to your game's theme or setting. Watch it and pay special attention to the music. How does it capture the different moods of what is happening to the characters?

Next, play through one of your favourite games and do the same. How does the music change depending on where you are in the game, or what is happening?

Think about what the music is doing to support, or contradict the player's action. Go through each of your own gameplay spaces or levels and write down three words or phrases that capture the tone or mood of the level. Make at least one of these words or phrases imply what is going to happen next - so if you are in a dark, gloomy place, the third phrase might be hopeful, to show that there is a way out of the danger.

Here are our notes for Stowe's Adventure using some key moments where we imagine the music should change:

In the opening scenes of our game, the space is bright, open, and adventurous.

- Finding Evelyn is triumphant, bright, with a hint of danger to come
- Discovering Archer is dark & dangerous in the future
- Racing back through the jungle is frantic and very dangerous.
- Finally escaping on the plane is triumphant. Your turn. Using [Beepbox](#), try to create music that captures these three words at the same time.

You can check out our [Beepbox tutorial](#) first before you get started. (8 min watch)

LESSON 8 SEQUENCE - PLAYTESTING	
<p>Student outcomes & Learning goals</p>	<p>This lesson looks bug fixing and game improvement through playtesting. It also looks at what might be next for students' games. It encourages them to think beyond just using it as a portfolio or how game development skills might lead to a job, highlighting that this practice can sit within an amateur or hobbyist practice, a culture & events one, or even as something not to be taken any further.</p> <p>At the end of this module, students should be able to:</p> <ul style="list-style-type: none"> • Come up with a plan for playtesting their game, recording results, and making any necessary changes • Reflect on what might be needed for their game to be finished • Position their game alongside other students, experimental, and artistic games • Think about next steps - whether skills development, showcasing, or further play & tinkering • Be aware of and engage with their local games development scene & activity
<p>Prep & introduction activities</p>	<p><i>Set up computer devices with internet access for students to use, and screen to show videos.</i></p> <p>How to get feedback on your game (7 min watch)</p> <p>Playtesting is less about finding out if people like your game (although that's certainly useful) and more about fixing what might be broken and strengthening what is there. First off, play your own game. Try to forget what you know about it and play it as someone fresh. Make a list of the following:</p> <ol style="list-style-type: none"> 1. Bugs - these are things that you know are broken in the game. You can fix these later on, but you can also see if people point them out when they're playing as well. 2. Questions you have about whether the game is clear: is it communicating what you mean it to? Is some part overwhelming? Does everything make sense? 3. Questions about progression and difficulty: is a sequence too hard too early, or is a later stage too simple? <p>Keep your answers to the questions to yourself for now. We'll compare them to what your playtesters think.</p> <p>Based on what you've learned, make a short questionnaire with space for people to write their answers which asks open questions about your game. Questions might be:</p> <ul style="list-style-type: none"> • What bugs, if any, did you notice in the game? • What do you remember of the characters? • What parts, if any, did you find too difficult? • What parts, if any, did you find too easy?

	<ul style="list-style-type: none"> • If you could change one thing in the game, what would it be? <p>Find a small group of people that you trust to play your game while you watch without interrupting them to explain or to ask or answer questions. You don't want to influence their reactions. Make notes of areas where they get stuck or confused while they play.</p> <p>Afterwards, give them your questionnaire to fill out. You can answer some questions at this stage but try not to influence their opinions: remember, you're looking for bugs and ways to improve the game, not whether they liked it. When you have them all written up, compare what your playtesters pointed out to what you originally thought. Are they the same? Did they find new things? Does it pose new questions for you to explore? And did they uncover a bunch of new bugs? Take this on board to improve your game or set you up for the next version.</p>
<p>Main Activity</p>	<p>Critique and reflection</p> <p>Giving and receiving feedback is an essential part of any creative process. Learning how to do that sensitively and in a way that supports a project and its maker can make a big difference to the success of your project.</p> <p>This is different from playtesting where your finding bugs plus what does and doesn't work. In sourcing critiques and reflection, you're looking to answer the questions about whether your game is successful in what it does. Firstly, find a group of people you trust to play your game. Have them play it through without commenting or interference. When everyone is done, sit down and go through the following:</p> <p>Encourage people to focus on whether the game does what you intended. Emphasise that comments should be about the game, not its maker. Explain what you were trying to do. Ask if there were parts of the game which achieved that; and if there were parts that didn't.</p> <p>Avoid closed questions (questions with a yes or no answer) as much as you can. Ask if there might be places where things could be improved. Discourage people from giving solutions at this stage, you're looking to understand where things don't quite work and why.</p> <p>Dig into anything you're unsure through open questions and avoid justifying your decisions. Listen to the feedback as impartially as you can.</p> <p>Ask for solutions to things if people have them. These are people's opinions on your game, and it is your game so you can take their advice if you want, but you can also come up with your own solution to how things should work. This can be a difficult process but remember if you trust the people in your group that everyone just wants to make the game better.</p> <p>When you're done thank everyone and then put your notes and comments away for a little while. Come back to them when you feel ready and go through them,</p>

	<p>reflecting on whether these comments and critiques feel true and if people's solutions to problems are appropriate.</p> <p>Remember, you don't need to take everything on board. Keep in mind your intentions for your game and your original ideas and consider each comment and piece of advice as to whether it helps to achieve what you want to make.</p> <p>Then, when you know what is useful and what is not as useful, go back to your game and make the changes.</p>
<p>Reflection activity</p>	<p>Next steps</p> <p>With your game finished it's time to decide what's next. This varies from person to person and from game to game. You might want to get more people to play it and look into marketing your game and reaching an audience. You might decide to keep it personal and only have friends or family play it.</p> <p>This game is a starting point. Where you take it is up to you.</p> <p>If you want to continue making games there are lots of new directions to take and communities to connect with.</p> <p>itch.io is an online place to find and share games, and if you're keen to find a wider audience or find new things to play.</p> <p>If you're looking for communities to help inspire you there are online and in person game development challenges you could take part in like Ludum Dare or the Global Game Jam. You can play winning entries from ACMI's past Screen It competition at acmiscreenit.itch.io</p> <p>If you're thinking about ways to think more widely about games and technology, keeping an eye out for events, exhibitions, and talks near where you are might help you meet the makers of games you've played, or show you something incredible from the fringes of game development you hadn't thought of.</p> <p>It's a big wide world of different types of developers, artists and makers out there.</p> <p>Research</p> <p>Think about where you might want to take your project next. Log onto itch.io and find three games that feel like yours. Ask yourself what it is about them that is interesting? If they are free, why are they free? If they cost money, how much is it? Is there anything about them that is different: does your game do things they don't or are there things missing from your game? What do you think you'd need to do to put your game up there for people to play?</p> <p>Try to find other places where those games might be mentioned online. Has the maker done any marketing on them? Do they have a company? Do they have a development blog?</p>

acmi

Search for any games events near where you. Are they exhibitions, code clubs, festivals, or meetups? Do they let you show your game there? Try to go along to one and see if there are people like you there. If not, have a think about what help or support you might need or what you'd like to see at one. Or if one doesn't exist at all, could you start something? Maybe a small exhibition of your friend's work or a weekly get together where you make games together.

Look at worldwide games events and competitions, the [Independent Games Festival](#), or [Ludum Dare](#), or the [Global Game Jam](#). Find some of the games made for that, and research the people who made them. Who are they? What are they interested in? Where are they from?

Think about how your game sits within this research. What are you interested in? How did some of those other people get started in games? What are they doing now? What next steps might you take? Who might you need to talk to? What skills might you develop?

Good luck with your future gamemaking and well done on getting this far, having a completed game (even if it still has bugs or work to do to it) is a fantastic achievement.

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.