Digital Technologies – 5 and 6_ Creating a digital solution

	STRAND		Knowledge and understanding				Processes and production skills											
			Digital Systems		Representation of data		Collecting, managing and analysing data		*Creating Digital Solutions by:*									
									Investigating and defining		Generating and designing				Producing and implementing		Evaluating		Collaborating and managing	
	Content Description		Examine the main components of common digital systems and how they may connect together to form networks to transmit data (ACTDIK014)		Examine how whole numbers are used to represent all data in digital systems (ACTDIK015)		Acquire, store and validate different types of data, and use a range of software to interpret and visualise data to create information (ACTDIP016)		Define problems in terms of data and functional requirements drawing on previously solved problems (ACTDIP017)		Design a user interface for a digital system (ACTDIP018)		Design, modify and follow simple algorithms involving sequences of steps, branching, and iteration (repetition) (ACTDIP019)		Implement digital solutions as simple visual programs involving branching, iteration (repetition), and user input (ACTDIP020)		Explain how student solutions and existing information systems are sustainable and meet current and future local community needs (ACTDIP021)		Plan, create and communicate ideas and information, including collaboratively online, applying agreed ethical, social and technical protocols (ACTDIP022)	
Sequence of Lessons / Unit	**Approx. time rq'd (hrs)**	**Year 5or 6**	CD	Achievement standard #	CD	Achievement standard #	CD	Achievement standard #	CD	Achievement standard #	CD	Achievement standard #	CD	Achievement standard #	CD	Achievement standard #	CD	Achievement standard #	CD	Achievement standard #
Problem-solving processes	16 hrs	A	☐		☐		☐		☑	3	☐		☑	4	☑	4	☐	5	☐	

Levels 3 and 4 Achievement Standard	**Levels 5 and 6 Achievement Standard** The numbering of the Achievement Standards below is reflected in the grid above to show coverage across the 8 units.	**Levels 7 and 8 Achievement Standard**
By the end of Year 4 • Students describe how a range of digital systems (hardware and software) and their peripheral devices can be used for different purposes • They explain how the same data sets can be represented in different ways. • Students define simple problems, design and implement digital solutions using algorithms that involve decision-making and user input. • They explain how the solutions meet their purposes. • They collect and manipulate different data when creating information and digital solutions. • They safely use and manage information systems for identified needs using agreed protocols and describe how information systems are used.	By the end of Year 6: • Students explain the fundamentals of digital system components (hardware, software and networks) and how digital systems are connected to form networks. (1) • They explain how digital systems use whole numbers as a basis for representing a variety of data types. (2) • Students define problems in terms of data and functional requirements and design solutions by developing algorithms to address the problems. (3) • They incorporate decision-making, repetition and user interface design into their designs and implement their digital solutions, including a visual program. (4) • They explain how information systems and their solutions meet needs and consider sustainability. (5) • Students manage the creation and communication of ideas and information in collaborative digital projects using validated data and agreed protocols. (6)	By the end of Year 8 • Students distinguish between different types of networks and defined purposes. • They explain how text, image and audio data can be represented, secured and presented in digital systems. • Students plan and manage digital projects to create interactive information. • They define and decompose problems in terms of functional requirements and constraints. • Students design user experiences and algorithms incorporating branching and iterations, and test, modify and implement digital solutions. • They evaluate information systems and their solutions in terms of meeting needs, innovation and sustainability. • They analyse and evaluate data from a range of sources to model and create solutions. • They use appropriate protocols when communicating and collaborating online.

Creating a digital game
Year Level 6 TOPIC Creating a digital solution Time: 20 HOURS

Developing an online game provides a useful context for students to apply and develop the problem-solving processes so they can create a digital solution, namely a digital game. This would involve defining the target audience for the game and their needs, designing the rules/actions and the appearance of the game (interface design). The 'rules' are written as an algorithm that include decisions, options and ways that the users input instructions. This addresses what data is needed for the game and what features the game must perform (functional requirements). Students use a programming language to create the digital solution (a game) and then judge if their solution meets the intended requirements. A potential extension when creating the digital solution is to connect an input device such as a Makey Makey board to create a game controller.

Flow of activities				
Short text	Analyse existing digital games Identify common elements of a game to understand functional requirements.	Design a digital solution Plan and design their own digital game, consider purpose, functional requirements, features and types of data.	Visual programming solution Use a visual programming language to create their digital game.	Evaluate your game Evaluate the level to which the digital game met the needs of the target audience or its intended purpose.
Question to guide exploration	*What makes a good game?*	*Can you design a game for a particular audience?*	*How do I use visual programming language to create my game?*	*How can I evaluate my game?*
AC Alignment	*Investigating and defining (ACTDIP017)*	*Generating and designing (ACTDIP019)*	*Producing and Implementing (ACTDIP020)*	*Evaluating (ACTDIP021)*
What's this about?	Game design provides an opportunity to analyse existing games and replicate or build on their functionality and game play to create a new digital game. It involves finding out exactly what the users want in a game – what the game should be achieving. A key stage in this process is defining the problem and identifying the functional requirements and any data needed. Game control is an important consideration in design. How the user interacts with the game can have a bearing on the overall enjoyment and game play. The use of arrow keys is often incorporated in game play; for example, arrow keys can be used to move an avatar or character within the game.	Generating and designing is about understanding the audience and then determining how the solution will work and what it will look like (appearance and functionality). It is important that the game meets the purpose identified and is suitable for the age and interest of the people most likely to play the game. Generating is about considering alternative ways that the solution could be solved. This has close links to the Critical and Creative Thinking general capability, which requires students to engage in the process of ideation.	A visual programming language enables students to sequence commands (displayed as blocks) to create a program. The students' use of a programming language should allow users of the game to make choices/decisions (branching) and the instructions should include repetition until particular conditions are met such as guessing the correct number.	An important stage of the problem-solving process is for students to evaluate their solution. A comparison can be made of the original design and the final solution. Did it meet the intended purpose? Did it meet the user's needs? Evaluating at this level also includes considering the sustainability of the solution and of existing solutions, using criteria such as the cost and resource demands of the solutions.
The focus of the learning	Students investigate a range of games. They identify common elements of a game that can be categorised based on user engagement/enjoyment. Students interact with a game and determine how effective and engaging the game is based on set criteria (eg interface, user interaction, feedback provided to the player during and after the game). This investigation helps inform the data and functional requirements for the student's own digital solution. Provide an opportunity for students to clearly define their problem, the purpose of their game and the audience for the game. Students identify functional and data requirements.	Students plan and design their own digital games, considering purpose, functional requirements, features and types of data needed (eg text, images, sound). Plans include rules and an algorithm to show branching and suggest how elements of repetition can be included. It also identifies user input. Designs can be in the form of storyboards or flow charts. Students can draw diagrams of the screen layout (eg how the user will interact with the game; what the user will see and do when they play the game). At this level the process could also involve students generating at least two or three alternative design ideas	Use Scratch, Tynker, Snap or other similar visual programming language to create an online game. Alternatively use a specific game-making platform such as Sploder. Ensure that the design phase includes algorithm development and students' tests (debugging) and evaluate their implemented design. There are many Scratch projects available via the Scratch community that are based on game development. Review YouTube tutorials to overcome challenges or alternatively modify existing Scratch projects to create your new game. (Discuss crediting the original developer.)	Compare designs with the final digital solution. What changes might have been made during the implementation process and what were the reasons for these changes? How are students assessing whether the solution met the user's needs? What feedback did they seek or receive? When students are evaluating if their solutions and those developed by others are sustainable they could ask questions such as whether the solution could operate on a range of platforms so users do not have to purchase specific hardware or software, or whether the solution could be upgraded with little or no cost.

	Extension: This activity has links with exploring input devices in digital systems. Provide a programming board that can be used as an input device; for example, with the aid of a simple circuit board such as Makey Makey students can design their own controller. The controller can be designed based on user needs and how they intend the user to interact with the game. For students who can handle more of a challenge the Arduino and ScratchX combination can enable students to design their own game controller.	before they settle on one idea for further development. This could involve roughly sketching different home page layouts (user interfaces) showing alternative positioning and sizes of buttons or colours. Students could get feedback from the people most likely to play the game.	Games can be created within the context of another learning area (eg History, Mathematics or Civics and Citizenship).	
Supporting Resources	Board game design first steps What are the steps when designing a board game? What makes a good game? Students explore design thinking processes to investigate how games are designed, created and played. Design thinking process: Ideation Explore the design thinking process of ideation and reflect on different ways we can generate ideas to solve a problem with a design brief. Makey Makey projects These projects incorporate Scratch programming. ScratchX An Arduino extension is available on ScratchX, a Scratch sister site hosting experimental extensions. See the instructions.	Gliffy Gliffy is flow chart software. Storyboard Generator Students choose a script and create a storyboard.	Sploder This is a game-making platform designed for students. Forums are public and moderated. There are different options for different styles of games and no programming or coding experience is needed. Lego game creator This website allows students to create platform-style games using the characters from *The Lego Movie*. Students follow a set of instructions to create their games. Scratch Students drag and drop blocks to create games. Games can be shared, so students can explore games made by other students, and can copy and modify pre-existing designs. Storm survivor: Input, decision-making and loops Students use a visual programming language to create a game or quiz to help members of a community prepare for a severe weather event. Scratch wiki This wiki contains tutorials and how to program certain actions in Scratch. These tutorials may help when a student is looking for a solution to a Scratch programming issue.	Software evaluation checklist Students could select some checklist points for their solutions and convert them into questions to assist with evaluation.
Assessment	**Assessment task** • List of the functional requirements of the game • List of the data needed to create a game ***Define*** *problems in terms of data and functional requirements and* **design** solutions by **developing** algorithms to address the problems.	**Assessment task** • User interface • Two parts of the algorithm showing one example of branching and one of repetition (students can submit their entire algorithm and highlight the relevant parts for assessment) **Achievement standard** **Incorporate decision-making, repetition and user interface design into their designs** and implement their digital solutions, including a visual program. Define problems in terms of data and functional requirements and **design solutions by developing algorithms to address the problems**.	**Assessment task** • Use a visual programming language to create an online game. • Success criteria: see below **Achievement standard** Incorporate decision-making, repetition and user interface design into their designs and **implement their digital solutions, including a visual program.**	**Assessment task** • Three evaluation criteria posed as questions and student responses to these questions **Achievement standard** 'They explain how information systems and their solutions meet needs and consider sustainability.'

Assessment advice

It is important that teachers don't think they have to assess all of the activities. For example, the creation of an online game is the summative assessment task and the smaller tasks that students undertake during this process are the formative assessment items. It might be useful to somehow incorporate the simple ways to capture evidence of achievement; for example, a flow chart is a great way to find out if the student has understood repetition and iteration prior to even implementing the program in Scratch. Also, a few paragraphs

to compare and critique existing online games is a good way to find out if students have considered those elements that make a game engaging and user friendly.