Years 7/8: Game Design

Learning hook	The learning hook here involves students playing a range of games using the MaKey MaKey to – firstly – engage in the project and be interested in it, but also to start developing ideas about what ‘playability’ means in a game. This unit of work encourages students to think like an inventor, and utilise tools available to them to create interactive games. Teachers set up the following activities for students to rotate through in sets. Watch this video for inspiration of how to set up the stations. Students complete this worksheet as they are rotating through the activities. • Operation: Uses aluminium foil and pencils/chopsticks for controls. • Super Mario Bros: Use playdough as a controller. • Pacman: Use lead pencil on paper as a controller. • Tetris: For this one, have a box of things that students could use for controllers (plants, coins, bananas, Lego, foam) and let students construct their own controllers.
Learning map and outcomes	Implement and modify programs with user interfaces involving branching, iteration and functions in a general-purpose programming language (ACTDIP030). Design the user experience of a digital system, generating, evaluating and communicating alternative designs (ACTDIP028). Design algorithms represented diagrammatically and in English, and trace algorithms to predict output for a given input and to identify errors (ACTDIP029). Learning intentions • Think like a programmer. • Understand basic circuits. Success criteria • Design an algorithm to construct a game. • Program and construct the game.
Learning input	Learn the software • Students must create an online account with Scratch. The online editor is the easier one to use, as there is no need to install Adobe AIR and then the offline editor. If there is a reason not to use the online editor (blocked at school, slow internet) you can follow these instructions to install offline: a. Install Adobe AIR. b. Install Scratch offline. • If using the online editor, check that Adobe Shockwave Player is installed. • Schools with BYOD iPad programs can use Pyonkee, which is a mirror of Scratch, and uses the same programming principles that are used within this unit of work. • Discuss with students the advantages of modularisation of programming. Programmers work through getting one element of their program right before

DIGITAL TECHNOLOGIES HUB

moving onto other programs. This also is important when they are learning. Students need to learn one thing before the other.

- Discuss with students the importance of tracking and recording errors, so that when predictable errors come up you can look to a troubleshooting guide to fix them easily. It is important that students note that programming is a problem-solving process, that not everything will work straight away, and that it requires a growth mindset in order to program.
    a. Keep moving forward!
    b. Growth mindset vs fixed mindset
- Demonstrate the Scratch interface. Go through with students the importance of using the correct blocks. Note that each of the blocks are colour-coded with what they are doing. Eg, the purple blocks have to do with the sprite's look, whereas the yellow ones are about control (control structures like repetition and selection).
- Demonstrate to students the need to start with an event (such as the flag being clicked) in order to have some action to start the event. Other events are available under the 'events' tab in the blocks section, but generally look different to the action blocks as they don't have any jigsaw piece in the top for other blocks to connect into. You can use Card 2 to go through this with students.
- Demonstrate to students that the scripts are attached to sprites, and that these actions occur on sprites. You can create a new sprite and demonstrate to students that the programming does not transfer with the sprite, that each individual sprite has its own programming.
- Students work through the programming cards. These cards are designed for students to work through at their own pace. They provide questions and extensions where appropriate and map against features of algorithm design. Card 1 is designed to be printed, cut in half and given to students as a checklist. Questions are given on each card. These can be answered in the students' books. Extension activities are given on each card to encourage students to combine skills that they have already learnt, or to push themselves further. These cards give an introduction to the basic features that they need to learn from Scratch in order to complete their game.
- Additional video tutorials are available:
    a. Hit test and variables
    b. Move with mouse
    c. Move on keypress

Learn the MaKey MaKey

- Demonstrate to students the connection of the MaKey MaKey to the computer.
- Demonstrate to students the earthing process and explain that this creates a circuit with your body. Demonstrate how, if you are not touching the earth, programs will not work.
- Ask students to create a basic controller out of a piece of cardboard and playdough.
- Connect the alligator clips supplied with the MaKey MaKey to the up, down, left and right keys.
- Connect these alligator clips to pieces of playdough on the board.
- Students can use the key movement game that they created. Let students experiment with this for a while, using the different keys on the MaKey MaKey to move things around on their program.
- For inspiration and ideas of different things students can use, watch Hacking everyday objects.

	● Ask students to reflect back to the learning hook and list the different materials that they used in these. What do these materials have in common? What other materials could they possibly use? Why does the cardboard not conduct electricity? What could they use for the controller base aside from cardboard? Design the game ● Students watch the following How the inventor of Mario designs a game, a film on the design process that Nintendo used when designing some of their games: Students brainstorm different things that they like about games they play. ● Students brainstorm, in groups, different things that they can integrate into their game, and start to develop control ideas and navigation of their program.
Learning construction	Interaction design ● Students should design a storyboard for their game. This should be annotated with things like what the character will do when certain things happen. This then leads to the design of their algorithm. These should be rough drawings. Algorithm design Model for students as a class the construction of an algorithm for Tetris. This should be done by asking students to deconstruct how they play Tetris. Questions that could be asked might resemble: What keys do I press? What happens when I press each key? What are the conditions for winning? How would I lose? How do I get a score? These questions can be put up on the board as a list of rules/conditions, then an algorithm constructed out of this. Teachers can ask 'What's the first thing I would do as a program?' (test for a line of bricks being full). The teacher can question students for this information and the class can co-construct the algorithm on the board. ● Students draft the algorithms that they need to complete the game. Character design/background design ● Students to draft design ideas for characters and backgrounds that are more descriptive drawings than their screen designs. These designs will be those that are brought into Scratch for development of their project. Students should team up with another group and get some feedback on their ideas. Students can use the 'critical friends' process to guide their feedback.
Learning demo	Student exhibition of games is the public display of the product in this unit of work. For long projects, students should be asked to present – at different stages – their progress, and to demonstrate to other students exciting things they are doing in their game.
Learning reflection	Students exhibit games in an exhibition on game design. This could include a range of existing games, as well as the students' own game designs. The year group will be broken up into two and students asked to vote on a 'people's choice' game design for playability (fun) and for programming (how complex the program is). Students within these groups will be awarded prizes for their games. A teacher selection will also be made of the game that is the most playable and difficult. Students are to evaluate their games against others using this scaffold.

DIGITAL TECHNOLOGIES HUB