## Digital Technologies – 5 and 6 _ Creating a digital solution

### Levels 3 and 4 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.

### 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.

#### Creating Digital Solutions by:

<table>
<thead>
<tr>
<th>Processes and production skills</th>
<th>Investigating and defining</th>
<th>Generating and designing</th>
<th>Producing and implementing</th>
<th>Evaluating</th>
<th>Collaborating and managing</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Levels 7 and 8 Achievement Standard

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.

---

Mapping template © Victorian Curriculum and Assessment Authority (VCAA). [Creative Commons BY-NC-SA 3.0 AU](http://creativecommons.org/licenses/by-nc-sa/3.0/au/)
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.

<table>
<thead>
<tr>
<th>Flow of activities</th>
<th>Short text</th>
<th>Question to guide exploration</th>
<th>AC Alignment</th>
<th>What’s this about?</th>
<th>The focus of the learning</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Analyse existing digital games Identify common elements of a game to understand functional requirements.</td>
<td>Design a solution Plan and design their own digital game, consider purpose, functional requirements, features and types of data.</td>
<td>Visual programming solution Use a visual programming language to create their digital game.</td>
<td>Evaluate your game Evaluate the level to which the digital game met the needs of the target audience or its intended purpose.</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Can you design a game for a particular audience?</td>
<td>Generating and designing (ACTDIP019)</td>
<td>Producing and Implementing (ACTDIP020)</td>
<td>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</td>
</tr>
</tbody>
</table>
### 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.**

**Success criteria:**
- **Achievement standard**
  - Incorporate decision-making, repetition and user interface design into their designs and implement their digital solutions, including a visual program.

**Achievement standard**
- They explain how information systems and their solutions meet needs and consider sustainability.

**Supporting Resources**

<table>
<thead>
<tr>
<th>Board game design first steps</th>
<th>Gillyf</th>
<th>Sploder</th>
</tr>
</thead>
<tbody>
<tr>
<td>What are the steps when designing a board game?</td>
<td>Gillyf is flow chart software.</td>
<td>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.</td>
</tr>
<tr>
<td>What makes a good game?</td>
<td>Storyboard Generator</td>
<td>Lego game creator</td>
</tr>
<tr>
<td>Students explore design thinking processes to investigate how games are designed, created and played.</td>
<td>Students choose a script and create a storyboard.</td>
<td>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.</td>
</tr>
<tr>
<td>Design thinking process: ideation</td>
<td></td>
<td>Scratch</td>
</tr>
<tr>
<td>Explore the design thinking process of ideation and reflect on different ways we can generate ideas to solve a problem with a design brief.</td>
<td></td>
<td>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.</td>
</tr>
<tr>
<td>Makey Makey projects</td>
<td></td>
<td>Storm survivor: Input, decision-making and loops</td>
</tr>
<tr>
<td>These projects incorporate Scratch programming.</td>
<td></td>
<td>Students use a visual programming language to create a game or quiz to help members of a community prepare for a severe weather event.</td>
</tr>
<tr>
<td>ScratchX</td>
<td></td>
<td>Scratch wiki</td>
</tr>
<tr>
<td>An Arduino extension is available on ScratchX, a Scratch sister site hosting experimental extensions. See the instructions.</td>
<td></td>
<td>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.</td>
</tr>
</tbody>
</table>

### 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.