a

**Internal Assessment Resource**

**Digital Technologies & Hangarau Matihiko Level 2**

This resource supports assessment against Achievement Standards 91896 and 91897[[1]](#footnote-1)

**Standard title:**  91896 Use advanced programming techniques to develop a computer program (6 credits)

91897 Use advanced processes to develop a digital technologies outcome (6 credits)

**Credits:** 12

**Resource title:** Global Citizen Game

**Resource reference:** Digital Technologies & Hangarau Matihiko 2.7A\_2.8A

|  |
| --- |
| This resource:   * Clarifies the requirements of the achievement standard * Supports good assessment practice * Should be subjected to the school’s usual assessment quality assurance process * Should be modified to make the context relevant to students in their school/kura environment and ensure that submitted evidence is authentic |

|  |  |
| --- | --- |
| Date version published by Ministry of Education | December 2018 Version 1  To support internal assessment from 2019 |
| Authenticity of evidence | Teachers/kaiako must manage authenticity for any assessment from a public source, because students may have access to the assessment schedule or student/ākonga exemplar material.  Using this assessment resource without modification may mean that students’ work is not authentic. The teacher may need to change figures, measurements or data sources or set a different context or topic to be investigated or a different text to read or perform. |

**Internal Assessment Resource**

**Achievement Standards:** 91896 and 91897

**Standard title:**  Use advanced programming techniques to develop a computer program (6 credits)

Use advanced processes to develop a digital technologies outcome (6 credits)

**Credits:** 12

**Resource title:** Global Citizen Game

**Resource reference:** Digital Technologies & Hangarau Matihiko 2.7A\_2.8A

**Teacher/Kaiako guidelines**

The following guidelines are supplied to enable teachers/kaiako to carry out valid and consistent assessment using this internal assessment resource.

Teachers/kaiako need to be very familiar with the outcome being assessed by the achievement standards. The achievement criteria and the explanatory notes contain information, definitions, and requirements that are crucial when interpreting the standard and assessing students/ākonga against it.

**Context/Te Horopaki**

This is an integrated assessment activity supporting a project-based approach that assesses against two achievement standards. Students will use advanced processes to plan, test and trial a computer game program that teaches others about the importance of protecting our oceans and waterways from plastic. They are required to implement advanced programming techniques within their game program.

Using advanced processes will support the students to develop a refined computer game program that has been comprehensively tested and debugged. Both achievement standards require testing to improve the quality and functionality of the outcome.

Students may use any suitable programming environment/language, as long as it supports a range of advanced programming techniques. For example, the student may develop a text-based adventure game or a mobile/desktop/web-based game with GUI interface.

Teachers are encouraged to edit the default task/context to make it suitable for their student community.

Note: Students may either complete the task or ‘pitch’ an alternate idea, provided their idea allows them to use advanced techniques to develop an advanced computer program. They should be encouraged to develop their custom solution.

**Conditions/Ngā Tikanga**

It is recommended that students should have at least two identified checkpoints with their teacher as they work through this assessment activity to ensure they have an opportunity to ask questions and gather feedback.

Conditions of Assessment related to this achievement standard can be found at <http://ncea.tki.org.nz/Resources-for-Internally-Assessed-Achievement-Standards>

**Resource requirements/Ngā Rauemi**

Students will need access to software that will allow them to document their development process. This could be in the form of a document, presentation, online portfolio, series of screen casts, etc.

They will also need access to project management and version control tools (which may include a mixture of digital and non-digital resources).

**Internal Assessment Resource**

**Achievement Standards:** 91896 and 91897

**Standard title:**  91896 Use advanced programming techniques to develop a computer program (6 credits)

91897 Use advanced processes to develop a digital technologies outcome (6 credits)

**Credits:** 12

**Resource title:** Global Citizen Game

**Resource reference:** Digital Technologies & Hangarau Matihiko 2.7A\_2.8A Version 1

**Student/Ākonga instructions**

**Introduction/Kupu Arataki**

This assessment activity requires you to plan, trial, test and develop a computer program using advanced programming techniques. You will utilise a development process to help you make informed decisions throughout the coding, testing and trialling of your program and show ongoing refinement to improve the functionality and quality of your program.

You will be assessed on how effectively you plan your development, decompose the outcome into smaller components, and test and refine your program so that it is a high-quality response to the task (e.g. well-structured, logical, flexible, robust and comprehensively tested).

When planning and developing your program, you must ensure your program:

* uses variables storing at least two types of data (e.g. numeric, text, Boolean)
* uses sequence, selection and iteration control structures
* takes input from a user, sensor(s), or other external source(s)
* produces output

AND includes two or more advanced programming techniques, such as writing code that:

* modifies data stored in collections (e.g. lists, arrays, dictionaries)
* defines and manipulates multidimensional data in collections
* creates methods, functions, or procedures that use parameters and/or return values
* responds to events generated by a graphical user interface (GUI)
* requires non-basic string manipulation
* uses functionality of additional non-core libraries.

Teacher note: Insert due dates and timeframes

**Task/Hei Mahi**

**Scenario**

The Enviroschools club at your school would like an engaging way to help the students/school community learn about the damage plastics are doing to our oceans and waterways. Members of the club have read about how gamification has been used to promote environmental causes. They have approached you to develop a computer game program that will help promote awareness of this global problem and ways in which students can take positive action to reduce plastic waste.

1. Decide on an appropriate planning methodology, and what project management and version control tools you will use to manage your program development.
2. Set up any necessary planning/project management tools.
3. Research any content for your game and use your chosen methodology/tools to plan out the development and structure of your game. For example:
   * Will there be levels, lives, time limits?
   * How will you incorporate important environmental information for the players?
   * Will the program use graphical elements or be a text-based adventure game?
   * Explain what relevant implications are important to consider in the development of your game program?
   * Decompose your program into the different components you need to incorporate into the final game (e.g. start-up screens/GUI interface, different levels, sub-programs/functions/methods).
4. Throughout your development, you must trial multiple components. For example, this could include different ways to present the user interface, different ways of scoring, etc. You should also think about the advanced programming techniques that will best make your program flexible and robust. Select the best components and/or advanced programming techniques to include in your final program, based on the results of your testing and trialling.
5. Use your selected version control tools/techniques to save successive versions of your code and keep evidence of how you created the program in an ongoing manner (e.g. screenshots showing your file structure with appropriately named versions/program components, including brief annotations of the changes made in each version).
6. Ensure your testing and trialling includes both expected cases and relevant boundary cases (e.g. what happens when the score/lives gets to a certain number). You may want to get other students, your family/whānau, and/or the Enviroschools club members to test your program at each stage and provide feedback to help you improve your final game program. Using others to test the program will help to ensure it is comprehensively tested for many different cases (including expected and relevant boundary cases). Note the improvements that could be made based on the testing and implement your changes.
7. Throughout the development of your program code, ensure that you document your program with appropriate variable/module names and comments that describe code function and behaviour. Follow the common conventions of your programming language (e.g. naming conventions or rules for program layout).
8. Comprehensively test your final game to ensure that it functions correctly and is of high-quality (e.g. bug free, has a well-presented and easy-to-use interface, contains all the required information).

Note: Testing can be recorded by making a brief screencast showing the outcome being comprehensively tested. If desired, you can take screenshots of your screencast and annotate them. This is often easier than trying to screenshot whilst testing where it is easy to ‘forget’ to screenshot a key part of the test. If you prefer, you can record a voice-over of the screencast to explain your testing and simply submit the screencast.

1. Discuss how the information from planning, testing and trialling of the components of your game program assisted you to develop a high-quality outcome. This can be in the form of a screen-cast, document with annotated screen shots, online presentation or oral presentation to your teacher/class.
2. Show how your program has addressed the relevant implications.

**Assessment schedule/Mahere Aromatawai: Digital Technologies & Hangarau Matihiko 91896 – Global Citizen Game**

|  |  |  |
| --- | --- | --- |
| **Evidence/Judgements for Achievement/Paetae** | **Evidence/Judgements for Achievement with Merit/Kaiaka** | **Evidence/Judgements for Achievement with Excellence/Kairangi** |
| Use advanced programming techniques to develop a computer program.  The student has:   * written code for a program that performs a specified task * used advanced techniques in a suitable programming language   **For example (partial evidence):**  The student’s program allows text variable entered by players to be stored under the player’s name, uses Boolean variables to store game play choices. They have used sequence, selection and iteration control structures within their code. They have utilised input from the player to control the flow of the game.  **For example (partial evidence):**  Student has used a GUI with event handling and programmed appropriate responses to the events (e.g. button clicks, mouse overs). Functions/modules are used to check if user input is valid and/or do calculations which would otherwise be repeated in the code. The program uses an array/list to store and update game information (e.g. characters, environmental facts to display).   * set out the program code clearly and documents the program with comments   **For example (partial evidence):**  Most of the variable names are clear, sensible and the code includes some comments, but these comments don’t describe the code’s function.  Eg # Menu Display   * tested and debugged the program to ensure that it works on a sample of expected cases   **For example (partial evidence):**  Student has provided evidence of testing their program. The program works on expected input, but may crash on boundary or invalid input.  *The examples above are indicative samples only* | Use advanced programming techniques to develop an informed computer program.  The student has:   * documented the program with appropriate variable/module names and comments that describe code function and behaviour   **For example (partial evidence):**  The student uses descriptive variable and function name, e.g. the menu choice function might have been called ‘menu check’, the variable holding the value of the player name might be called ‘player name'. The code has comments at key points, e.g. ‘function checks that user input is a number that is between a given lower and upper bound’.   * followed common conventions for the chosen programming language   **For example (partial evidence):**  The student uses all lower case variable names for code written in Python. Function definitions are placed before or after the main function, as per the programming language. Layout conventions are followed, e.g. whitespace between definitions. Indentation and/or bracketing conventions are followed as per the programming language. The student has used an automated tool to check that their code follows common conventions.   * tested and debugged the program effectively to ensure that it works on a sample of both expected and relevant boundary cases.   **For example (partial evidence):**  The student has provided evidence of testing relevant components (or the complete program) to confirm that it works correctly on a range of boundary cases, e.g. for lives at 0, 1, -1 or menu selection at the upper and lower bounds of the choices.  Student testing methodology is effective within the context of the problem.  *The examples above are indicative samples only* | Use advanced programming techniques to develop a refined computer program.  The student has:   * ensured that the program is a well-structured, logical response to the task * made the program flexible and robust   **For example (partial evidence):**  The student has used abstractions where appropriate. Functions have been used to avoid repeated code. The code works for expected, unexpected and boundary values. It is easy to extend its functionality of the code (e.g. a function has been used to check the menu choices, so it would be easy to update menu to add another level or path). They have used derived values to iterate through a collection instead of hard coded values. The code works for expected, unexpected and boundary values. They have used the GUI to limit invalid input or used other appropriate techniques such as try/except to check for validity.   * comprehensively tested and debugged the program.   **For example (partial evidence):**  Student has supplied test plans and/or annotated screenshots/a screencast showing that the program components (and final program) have been tested to ensure that it works correctly, e.g. they have used others to comprehensively test their program to ensure it responds gracefully to a variety of input.  *The examples above are indicative samples only* |

Final grades will be decided using professional judgement based on a holistic examination of the evidence provided against the criteria in the Achievement Standard.

**Assessment schedule/Mahere Aromatawai: Digital Technologies & Hangarau Matihiko 91897 – Global Citizen Game**

|  |  |  |
| --- | --- | --- |
| **Evidence/Judgements for Achievement/Paetae** | **Evidence/Judgements for Achievement with Merit/Kaiaka** | **Evidence/Judgements for Achievement with Excellence/Kairangi** |
| Use advanced processes to develop a digital technologies outcome.  The student has:   * used an appropriate planning methodology to plan the development of a digital technologies outcome * decomposed the outcome into smaller components * trialled components of the digital technologies outcome   **For example (partial evidence):**  The student has decided to follow an Agile-based planning methodology. They have broken their outcome down into a series of components. For each component they have created a piece of code and tested that code. They have trialled different methods for providing a user interface for the app and added to/changed the interface as they progressed through the programming stages.  Towards the end of the process the components have been combined into a fully working version of the outcome.  The student has used Trello (or an off-line visual planning board) to manage their development process. Student code is clearly named and shows version numbers/indicates which part of the decomposition has been coded. They have created a team drive and added each version of the code to the drive so that the teacher has access and the program is backed up in the cloud.   * tested that the digital outcome functions as intended   **For example (partial evidence):**  The student has provided evidence of testing with expected cases. They have also provided screenshot/screencast evidence showing that the actual program works as expected. They may not have had others test the program or looked at boundary/unexpected cases.   * explained relevant implications   **For example (partial evidence):**   * Student has explained the importance of creating code which is functional and easy to use. * They explained why their legal obligations need to be honoured. * They have explained why their program needs to meet the requirements of the Enviroschools club and why the game needs to have relevance to important environmental issues.   *The examples above are indicative samples only* | Use advanced processes to develop an informed digital technologies outcome.  The student has:   * trialled multiple components and/or techniques and selected those which are most suitable * used information appropriately from testing and trialling to improve the functionality of the digital technologies outcome   **For example (partial evidence):**  The student trialled various techniques for programming the menu system and decided to create a function that could be called from any point within the program. They considered alternate ways of getting user input and laying out the GUI (or text-based menu system) of the game. They trialled the different ways with the members of the Enviroschools club and selected the one that was rated easiest to use and understand by the testers.   * used project management and version control tools and techniques to effectively manage the development of a digital technologies outcome   **For example (partial evidence):**  The student has updated their development plan, when they realised a new component was needed to improve the game play and make the game less predictable.  They have saved versions of the outcome (e.g. developing new versions of the working file) at each stage of the development. They have named each file with a version number and date.  Evidence shows that at each stage of the development a new version of the working file was created, and these were named appropriately.  They have used evidence from their trialling to inform their decisions to update their project plan.   * addressed relevant implications   **For example (partial evidence):**  The student has created a program which functions as intended and easy to use (rather than just explaining that these are important attributes). They have provided annotated screenshots of their program to illustrate what they have done to address functionality, usability, accessibility (etc.) implications. Note that students only need to address relevant implications – where the output is essentially text, aesthetics are less relevant than other implications. Students who have created a game with a GUI interface will have shown adherence to aesthetic design principles.  *The examples above are indicative samples only* | Use advanced processes to develop a refined digital technologies outcome.  The student has:   * *d*iscussed how the information from planning, testing and trialling of components assisted in the development of a high-quality outcome   **For example (partial evidence):**  The student presents a brief reflective summary of how the information from planning, testing and trialling of the components of their game program assisted them to develop a high-quality outcome. They provided annotated screen shots of the changes they have made throughout the process and how feedback from the users and the testing process helped them to refine their program. They also included screen shots of their Trello board with a reflection on how it guided their development process and helped them to complete all the components and keep on track with their time management.  *The examples above are indicative samples only* |

Final grades will be decided using professional judgement based on a holistic examination of the evidence provided against the criteria in the Achievement Standard.

1. Achievement Standards 91896 and 91897 are derived from both *The New Zealand Curriculum* and *Te* *Marautanga o Aotearoa.* [↑](#footnote-ref-1)