Level 6 – New Zealand Curriculum

DIGITAL TECHNOLOGIES & HANGARAU MATIHIKO

Teaching and learning programme

Project based learning – Interactive gaming

MINISTRY OF EDUCATION
TE TĀHUHU O TE MĀTAURANGA

Developed by Ruth Davey, Lincoln High School, 2018

External links to websites
The Ministry of Education does not accept any liability for the accuracy of information on external websites, nor for the accuracy or content of any third-party website accessed via a hyperlink from this resource. Links to other websites should not be taken as endorsement of those sites or of products offered on those sites. Some websites have dynamic content, and we cannot accept liability for the content that is displayed.

Published 2018 by the Ministry of Education
PO Box 1666, Wellington 6011, New Zealand

www.education.govt.nz

Summary of the teaching and learning programme

This themed, project-based learning programme asks students to create a game that will assist other students to learn about a myth or legend. Students work on various aspects of the project throughout the year, offering the possibilty of collaborative work, but each outcome will be unique.

By the end of this teaching and learning programme, students will be able to:

- research to find ideas and user preferences from existing games and myths and legends
- research programming techniques that will assist them in creating their own game
- create and analyse survey data to enhance their ideas
- iteratively develop and test code for their own game in the chosen computer language
- investigate sorting and searching algorithms
- competently design and evaluate good human computer interfaces.

Duration

This resource outlines a year-long programme. It details teaching and assessment for each term.

Key teaching and learning concepts – the big ideas

- An interactive game provides a platform to introduce myths and legends from the diverse people of New Zealand.
- Researching game development principles is essential for creating an interactive game.
- Iterative development results in an outcome that is fit for purpose.
- A well-designed game requires awareness of the target audience.

Alignment to NZC and/or Te Marautanga – (DTHM progress outcomes and progressions)

This material includes the following progressions:

Computational Thinking

Students will:

- independently decompose a problem to create an algorithm
- implement algorithms as a program, with comments
- use organised approach for testing and debugging
- apply modular structure to program
- store data in collections
- compare the costs of two different searching and/or sorting algorithms in relation to number of comparisons
- use an iterative process to design, develop, document and test a computer program.

Designing and Developing Digital Outcomes

Students will:

- select appropriate software and file types for particular purpose
- select best tools and techniques to solve a problem
- work through an iterative process to design, develop, create, store, test and evaluate a program to meet its purpose
- independently work through the above iterative process.

Technological Practice: Technology cycle

Students will:

- create a proposal for an outcome
- design and develop the outcome
- use clear stakeholder input to design an outcome that is fit for purpose for a proposed end-user group
- create a prototype from the design.

Links to other learning areas

This programme links to science in its involvement with fair testing and to mathematics in determining the best, worse and average values using experimental statistics.

The theme of myths and legends could link to Māori, English or classics.

Teaching and learning pedagogy

This is a project-based approach to planning an entire year's work. Students produce one body of work, although this may be broken down into term sections. Teachers need to ensure that students are correctly scaffolded and managed to complete the work and that they present all the evidence required by the standards.

Prior knowledge and place in learning journey

Students are expected to have prior experience of using Word, file and folder management, basic literacy and searching using a web browser; understanding of copyright and Creative Commons is required.

Resources required

- Teacher and student resources that have been developed for this programme are available on TKI
- A desktop or laptop computer
- An internet connection
- A web browser and access to YouTube videos.

Software used:

- Word processor
- Python 3.6 or higher
- Pygame 2.9
- Arcade
- Text editing software such as Notepad++ or IDE such as Wing 101, Pycharm or Sublime
- Survey software such as Microsoft Office 365 Forms or Google Forms
- Database software such as Microsoft Access

How you might adapt this in your classroom

A different programming language or game engine as well as the resources for editing and running them can be used. Some options are: App Inventor, Game Maker, GameFroot, Javascript with Phaser or CodeAvengers, Unity or even Scratch.

Note: it is wise to limit the development to a 2D game as there is not enough time or resources to easily develop the 3D graphics for a first-person shooter game. Students should also not attempt to make too many of the required graphics themselves. In coding a game, the important aspect is how the game functions.

- It isn't necessary to do all the assessments, but the entire programme should be taught. Some students need more time to learn some topics than others, so adapt the programme accordingly.
- The searching and sorting assessment could be moved to term 3, doing only some programming of searches and sorts in term 1 as a class exercise. Database analysis of user opinions would then move into term 1 and the rest of the programme would move forward accordingly.
- Some students struggle doing the iterative design and coding assessments together. They should attempt some form of both, but use your judgment about whether to only assess one of these.
- Some schools prefer not to do the external assessment with their students. In this case, leave out the HCI assessment.

- The approach of themed project-based learning can easily be adapted to suit a different theme, for example, transport problems, or a different digital outcome, such as producing an infographic, a website, a brochure or a printed board or card game. This only would change the nature of the research to be conducted and the skills to be taught. For example, instead of programming a game, a board or card game could be developed using tools like InDesign and Illustrator, or equivalent free software, such as GIMP and Inkscape. For this alternative, instead of learning to code and use algorithms, students would need to learn to use the chosen drawing software, the principles of design and how to incorporate these in their work.
- Additionally, it is possible to weave in aspects of Hangarau Matihiko if desired.

Assessment

AS 91877 Develop a proposal for a digital outcome (3 credits)

AS 91885 Demonstrate understanding of searching and sorting algorithms (3 credits)

AS 91879 Develop a digital outcome to manage data (4 credits)

AS 91883 Develop a computer program (4 credits)

AS 91884 Use basic iterative processes to develop a digital outcome (6 credits)

AS 91886 Demonstrate understanding of human–computer interaction (3 credits) - external

TERM 1 OUTLINE

Teaching and learning programme

This is a full year programme split into terms. Term one, students will be learning to code in Python, how to think like a coder (algorithms) and how to plan code and test cases for more involved programs. They will be investigating searching and sorting algorithms. It is important to continually teach good habits like planning and commenting, even for the simplest programs.

Consider weaving the learning so that they cover two topics together over the total time allocated. Devoting half the weeks' lessons to each topic works well when students are learning to code alongside learning computer science, thinking, analysing, etc. This also means that they continually do some coding almost every lesson.

What is being covered	Duration	Specific learning outcomes *Students will be able to:*	Learning activities	Resources *Access all resources here*
Introduction of teacher and students to the awesome possibilities in IT Introduction to the program and basic good practice Introduction to Python (or your choice of programming language)	1 week – or part thereof (First week back at school)	• understand that there is more to DT than just coding and assessment • manage their files and folders effectively • understand some ethics • write small, basic programs.	• Introductions, mihimihi and class rules • Explain overview of the course. • Best to warn students that they will be creating a good 2D game – not enough time or manpower to create amazing 3D graphics. This course is about creating a usable program fit for purpose. • "Find someone who ..." ice-breaker exercise (student resource 8) • Create file and folder structure for the year. • Write "hello world" program as an introduction to IDE and Python language. • Comments and program structure • Introduction to input and output • Student programming task - Exercise 1	YouTube videos from Made in Code series Find Someone Who worksheet Notes on Teaching Programming Python v3.6 Pygame Arcade Wing 101 Example code

What is being covered	Duration	Specific learning outcomes *Students will be able to:*	Learning activities	Resources *Access all resources here*
Basic Python	3 weeks	• work in the IDE to develop small programs • understand how to use: – variables – if statements – loops – functions.	• Some print formatting • Short practice exercise • Variables and constants – more conventions • Maths expressions • Variables exercise • If statement • Thinking – how to decompose a problem example • Conditional exercise • For loops – with worked example • For loops exercise • While loops – with worked example • While loops exercise • Simple try-except error handling • Introducing functions • Mixed loops exercise • Challenge exercise • Strings and lists can also be done if there is time (exercises included).	Python exercises Try-except example Try-except as a function

What is being covered	Duration	Specific learning outcomes *Students will be able to:*	Learning activities	Resources *Access all resources here*
Introduction to algorithms Searching and sorting algorithms Iterative program development Python programming	4 weeks	• describe an algorithm • program in Python using: – basic lists – functions • create a program using iterative development • write code to perform: – a search algorithm – a sort algorithm • understand how to calculate the cost of an algorithm.	• Introducing algorithms • Some more advanced Python: – lists – functions • Explore searching algorithms. • Explore sorting algorithms. (See Exploring Algorithms teachers' notes for details on how to teach this section.)	Exploring Algorithms teachers' notes Old phone book Scale and different weights CSFieldGuide.org.nz algorithms chapter Resources: • unsorted list • sorted list Search and sort cards • word cards • number cards
Investigation of searching and sorting algorithms assessment	2 weeks	• how to experiment to get a better idea of the implications of searching and sorting big datasets • complete a report on the findings.	• Exploring Costs – Costs experiments (see teachers' notes) • Complete the assessment.	Exploring Costs teachers' notes Programs developed or found in previous section Comparison experiment example Digital Investigators assessment

TERM 2 OUTLINE

Teaching and learning programme

This term, students will begin to develop a brief for their game and will learn about game development, how to capture and analyse user opinions, and how to use a game engine to assist them in creating their game. Remember to weave the learning through the week and discuss issues as they arise. For example, it's helpful to discuss some aspects of HCI and design at this early stage of the development.

Students should be continually practising the planning for code and test cases wherever appropriate. Once these are habits, iterative design becomes easier.

What is being covered	**Duration**	**Specific learning outcomes** *Students will be able to:*	**Learning activities**	**Resources** *Access all resources here*
Game development theory	½ week	• understand some game development theory • see that there is more to game creation that simply implementing their own idea • be more aware of the importance of stakeholder preferences.	• If possible, organise a visit from a game development professional sometime during this section. FutureInTech can help you with this. • A professional will inform students of some of the realities about game creation and reinforce your restrictions about time constraints – they will only have a few weeks to craft their game program. • Watch "So you want to be a game designer?" https://youtu.be/zQvWMdWhFCc • Watch "What is a game?" https://youtu.be/H0ReU2tvLFo • Research game flow theory (not the game flow) http://www.jenovachen.com/flowingames/flowtheory.htm https://gamedesignconcepts.wordpress.com/ • Complete Analysing Games worksheet. • Class discussion on what is good to include, what is to be avoided.	Analysing Games worksheet

What is being covered	Duration	Specific learning outcomes *Students will be able to:*	Learning activities	Resources *Access all resources* here
Using databases to analyse user opinion	3 weeks	• create a survey and capture results • create a database that includes a table, a report and some queries • interpret results of database queries.	• Research into opinion surveys – what makes a good survey? • Class discussion of results. • Research how to create good questions. • Create a simple class survey using Forms (either Google or Office 365). Choose a topic they are all familiar with so no other research is necessary. • Send out survey. • Investigate survey tools and produce a PMI chart. • Creating a database – basics: – what is a database? – what databases do you already use? – why are databases useful? – naming conventions for tables, fields, etc. – creating tables – data entry – creating forms – creating queries – creating reports – importing data to database table – using database to analyse results – smart queries – interpreting results. • Opinion survey and analysis exercise	Survey Tools list Opinion Survey exercise Access or another database Choice of survey tool

What is being covered	**Duration**	**Specific learning outcomes** *Students will be able to:*	**Learning activities**	**Resources** *Access all resources here*
Database assessment and begin research for initial brief	2 weeks	• conduct research on existing outcomes and topic • conduct research on user preferences • create a survey • analyse results • write a short report explaining the results.	• Research existing games – create PMI. • Research myths and legends. • Research user characteristics to determine possible game preferences. • Use research to create a thoughtful survey on user game preferences – to inform required proposal specifications. • Create queries to analyse user preferences from survey. • Send out survey and collect results (at least 10). • Create database to analyse survey results. • Create a range of initial game ideas (at least 3). • Report on user preferences and choice of stakeholders for project (no more than 3 stakeholders).	About that Game assessment
Developing initial proposal	2 weeks	• create a proposal.	Note: this section and the next one work well together sharing the week's lessons. As students learn to use the game engine, they firm up their ideas for their own game creation. • Use research information from About that Game assessment to create a proposal for the game. • Share game ideas with chosen stakeholders and ask for feedback. • Choose one idea that will be developed. • Determine purpose of outcome. • Determine user and stakeholder profiles. • Determine potential resources for game development. • Determine specifications. • Determine the outcome statement, scope and context of the game.	Continue About that Game assessment

What is being covered	Duration	Specific learning outcomes *Students will be able to:*	Learning activities	Resources *Access all resources here*
Learning to create games using Arcade search engine. Alternatively students can learn how to make simple text-based games in Python, or simple animated games in Scratch or GameFroot.	2 ½ weeks	• draw on computer • understand the game loop used in Pygame • understand the program structure required for a game • create a simple platform game.	• Teacher: watch “Teaching Python 3.6 using games” for background info on using games to teach Python https://www.youtube.com/watch?v=MbJUMMvNMqk or read http://2017-craven-webinar.readthedocs.io/en/latest/ • http://programarcadegames.com/ • Video tutorials are also in this online book resource (all of these may be shared with students at teacher discretion – author has granted permission). • The full list of functions available in Arcade, and other Arcade resources are available here: http://arcade.academy/arcade.html • Students explore and experiment with techniques that they will use when developing their game. The examples and explanations from the book will help develop their understanding and their ability to create exciting Arcade-type games of their own. They will be graded only on the parts of the program that they create on their own. Techniques include but are not limited to: – how to draw on your computer (Ref: chapter 2) – simple shape animation – array-backed grids – platformers and variations – mazes – sprites. • Students make basic notes as they investigate techniques to include in their outcome, for use in their brief development in next assessment. • Hold a class discussion about the interface of the game and what could make it more usable.	base_game_code.py (from Arcade resources) reference book http://programarcadegames.com/ Video tutorials are also in this online book resource (all of these may be shared with students at teacher discretion – author has granted permission).

TERM 3 OUTLINE

Teaching and learning programme

This term students will be creating their game. They will also craft their external report towards the end of the term, with two weeks in the last term to complete their external submission.

What is being covered	Duration	Specific learning outcomes *Students will be able to:*	Learning activities	Resources *Access all resources here*
Using Arcade game engine to create a game	1 week		More experimenting with Arcade code	
Iteratively creating a game and a computer program for the game	6 weeks	• create a game from their brief • create a refined brief • use an iterative process to design, create and test the game • create a final brief.	**Note:** *students who struggle with using Arcade and Pygame can create a simple quiz game or guessing game using only Python.* • Do the assessment. • Refine the brief. **Note:** *Teachers need to manage student work with regular checkpoints to establish progress.* • Students may talk to each other and discuss potential solutions or share techniques, but they may NOT copy from, or write code for, each other. • Complete the final brief, planning and game code.	Game On! assessment
HCI and crafting the external report	1 ½ weeks + (1 ½ weeks of school exams)	• understand why HCI is important • practise for the external assessment • create an external submission.	• Class discussion on usability of interfaces, simple HCI situations and examples • Class discussion on how to evaluate interfaces • Class discussion about heuristics • Do various HCI exercises from CS Unplugged, CS Field Guide or similar resource • Class exercises to evaluate Interfaces and suggest improvements • Reflect on the game they created, and the interface used and other examples. • Do the external assessment.	Assortment of devices HCI external assessment

TERM 4 OUTLINE

Teaching and learning programme

This term students will be completing any resubmissions not yet completed. They will also be practicing their coding skills.

What is being covered	Duration	Specific learning outcomes *Students will be able to:*	Learning activities	Resources *Access all resources here*
Term 4: Complete any outstanding assessment resubmissions. Do more programming exercises.	2 weeks	• coding practice	• Work on responding to feedback in order to improve and resubmit any outstanding assessments.	

Acknowledgments and Thanks

For their ideas and contributions to making this resource possible, I would like to acknowledge Prof Paul Craven and Julie McMahon.

DIGITAL INVESTIGATORS ASSESSMENT TASK

Achievement standard:	91885
Standard title:	Demonstrate understanding of searching and sorting algorithms
Total credits:	3

OVERVIEW

This assessment activity requires you to use the research, experiments and data you have collected to create a digital portfolio of evidence for searching and sorting algorithms.

You can use annotated photographs, annotated sketches, short videos or written descriptions to do this.

You may have worked with others while investigating the topic; however, you must each submit your own evidence in your own words for this assessment.

HOW WILL YOU BE ASSESSED?

You are going to be assessed on how comprehensively you demonstrate your understanding of searching and sorting algorithms.

TASK

Be sure to include evidence from your research, experiments and data to support your responses to the following topics:

1. Real world scenarios
   - What is an algorithm?
   - What sort of real world problems can be solved by using good searching and sorting algorithms?
   - Why is it important to choose good algorithms for these situations? Write a paragraph to explain your thinking.
   - Discuss possible implications of using an incorrect or inefficient algorithm for such real-world situations.
2. Searching algorithms
   - Which searching algorithm are you choosing to carry out?
   - Show evidence of you demonstrating this algorithm using equipment supplied by your teacher.
   - Describe the cost of the algorithm you used and how you arrived at that cost.
3. Sorting Algorithms
   - Which sorting algorithm are you choosing to demonstrate?
   - Show evidence of you demonstrating this algorithm using equipment supplied by your teacher.
   - Describe the cost of the algorithm you used and how you arrived at that cost.
4. Costs when dataset size increases - using either a searching or a sorting algorithm
   - Describe in your own words how the cost will change as the dataset size increases for the algorithm you used in **either** Task 2 or Task 3.

Achievement standard:	91885
Standard title:	Demonstrate understanding of searching and sorting algorithms
Total credits:	3

TASK (CONTINUED)

5. Comparing **two different** searching or sorting algorithms.
    - Using the experiments you did in class, with datasets of steadily increasing size, determine the best, worst and average cost of performing the algorithm.
    - Explain what you see happening to the cost of each algorithm as the dataset size increases.
    - Compare the two algorithms and say which you would choose to use for a large dataset and why.
    - Explain the implications of using this algorithm in a real-world situation.
6. Final evaluation
    - explain the relationship between searching and sorting algorithms
    - consider the searching algorithms you have studied. When do you think they are useful in the real world? Justify your answer
    - consider the best, average and worst-case costs of either the two related searching algorithms or the two related sorting algorithms you used and explain the implications of these in the real world. When would you choose to use them and why?

HAND IN

Your digital evidence in the form your teacher will indicate (a Word document, Google Doc, OneNote class notebook, portfolio, visual diary)

DIGITAL INVESTIGATORS TEACHER GUIDELINES

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

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

Please be aware that NZQA have read the assessment task but the task will still need to be checked by the teacher using the assessment to ensure it meets all requirements.

CONTEXT/TE HOROPAKI

This activity requires students to work through a set of tasks, guided by their teacher, and finally do an evaluation, to demonstrate their understanding of searching and sorting algorithms. Each task can be supported by the teacher with structured lessons, discussions and experiments. Alternatively, the students could use this as an inquiry guided by the teacher.

The tasks listed below provide a framework for this. Students will be creating a portfolio of evidence, including written, photographic or video evidence of them carrying out at least one searching algorithm and one sorting algorithm.

CONDITIONS/NGĀ TIKANGA

Where a group approach is used, the teacher needs to ensure that each student creates their own portfolio outlining their understanding of searching and sorting algorithms in their own words.

You may want to give students guidance on appropriate style and format for their portfolio. A written portfolio gives demonstrates a student's understanding, but classroom observation and electronic documents with short embedded videos can fit the same purpose. Annotated photographs are also useful in explanations.

Conditions of Assessment related to this achievement standard can be found at http://ncea.tki.org.nz

RESOURCE REQUIREMENTS/NGĀ RAUEMI

Students will need access to the web, digital devices and information from a variety of sources, such as extracts, and/or notes from textbooks, the Computer Science Field Guide or "CS-unplugged".

ASSESSMENT SCHEDULE: DIGITAL INVESTIGATORS AS 91885 – DEMONSTRATE UNDERSTANDING OF SEARCHING AND SORTING ALGORITHMS

EVIDENCE/JUDGMENTS FOR ACHIEVEMENT/PAETAE	EVIDENCE/JUDGMENTS FOR ACHIEVEMENT WITH MERIT/KAIAKA	EVIDENCE/JUDGMENTS FOR ACHIEVEMENT WITH EXCELLENCE/KAIRANGI
The student demonstrates understanding of searching and sorting algorithms.	The student demonstrates in-depth understanding of searching and sorting algorithms.	The student demonstrates comprehensive understanding of searching and sorting algorithms.
The student has: • described applications of searching and sorting algorithms *For example (partial evidence)* The student has described the application of searching and sorting algorithms. For example: Searching and sorting are used in a library to help us find the books that we want. The books are stored on the shelves in a sorted order which makes searching for the book you want a lot easier….the books are organised in a way that makes it easier for us to find them…. All of the names in a phone book are sorted so they appear in alphabetical order…this makes it much easier for us to find the name that we need…. **The student has:** • carried out a searching algorithm accurately *For example (partial evidence)* The student has provided evidence of understanding the steps required to undertake one of the searching algorithms like a linear or binary search. (This evidence can be supported by teacher observation if the written, annotated or video evidence within the report is incomplete or unclear).	**The student has:** • explained the relationship between searching and sorting algorithms *For example (partial evidence)* *"When we were given an unsorted list, we could not easily find the word we were looking for because …* *… the data has to be sorted before the algorithm works …* *… if they were not sorted before we tried to find something … this is because* *… you can only use binary search on data that has been sorted because …"*	**The student has:** • discussed real world usage and implications of searching and sorting algorithms *For example (partial evidence)* An example student response: *… with 3.5 billion searches a day on average … imagine how long it would take …* *… Amazon's 5 million digital documents … if the data wasn't sorted …* *… quicksort is used most commonly even on large datasets like …* *… but the memory requirements of the algorithm mean.*

ASSESSMENT SCHEDULE: DIGITAL INVESTIGATORS AS 91885 – DEMONSTRATE UNDERSTANDING OF SEARCHING AND SORTING ALGORITHMS

The student has: • carried out a sorting algorithm accurately. ***For example (partial evidence)*** The student has provided evidence of understanding the steps required to perform a sorting algorithm. For example the selection sort or quick sort. (This evidence can be supported by teacher observation if the written, annotated or video evidence is incomplete or unclear). **The student has:** • described how the cost for a chosen searching or sorting algorithm changes as the size of the problem increases. ***For example (partial evidence):*** An example student response: *"When we were given the bigger list of words to look through it would have taken us twice as long to check each word to find the one we needed because there were twice as many words…* *using binary search if the list was twice as long you would only need to do one more search.…* *but what if the list was twice as long? … you would have to…* *…but with a pile of 20 cards we would have to go through the algorithm…"*	**The student has:** • determined the best, average and worst case costs of two related searching or sorting algorithms and explains the implications. ***For example (partial evidence)*** An example student response: *"When we used the linear search the worst case would be that we might have to look through all the data in order to … this is not the case with the binary search because …* *the best case would be that we find the ball under the cup on the first try … but on average … this means that …* *if you weigh the heaviest container first …* *… the best case would be that … this means …* *"the average" case would be that I would need 6 tries … but if you use binary search …* *if the boxes were already in order … this would be the best case … but on average …"* *"Since the number of comparisons for … algorithm increases so much faster than it does for the … algorithm, I would choose … algorithm to sort lots of data or it could take years to do. However, if there are only 10 or less items to sort, it doesn't make much difference which algorithm you use."*	**The student has:** • investigated the cost of searching and sorting algorithms with different datasets. ***For example (partial evidence)*** An example student response: *… in our experiment it took 6 more iterations … this is in line with the O^2 notation because …* *… when the data was reversed … the algorithm still took …* *… when there were 10 times as many words the linear search would take… while the binary search would only take …* *… we see a more linear increase in the cost … when compared to … the difference is much more notable as the size of the sample increases …* *… as you would expect from the average case cost of O(n log n) … when we … This means that …*

Final grades will be determined on a holistic examination of the evidence provided against the criteria in the achievement standard.

All supporting materials are supplied with this programme and can be found on the TKI website.

ABOUT THE GAME! ASSESSMENT TASK

Achievement standard:	91879 & 91877
Standard title:	Develop a proposal for a digital outcome (3 credits) Develop a digital outcome to manage data (4 credits)
Total credits:	7

OVERVIEW

This assessment activity requires you to develop a proposal for an interactive 2D game that will introduce a myth, legend, or tradition, in a way that inspires other students. You will then develop the code to create the game.

Before you can develop your game proposal, you need to:

- explore the problem, the context, the scope, the purpose and the end users.
- conduct market research in the form of a survey to find out what kind of game you will develop.
- create a database outcome to manage and analyse the data you collect.

The actual game development takes place next term.

HOW WILL YOU BE ASSESSED?

You will be assessed on how you have created a proposal, researched the topic and game format, and collected and managed the data related to your game proposal.

You must create and demonstrate evidence of your own work, using research and feedback to improve the proposal and show clear links between the requirements, end users and resources.

If you are working as a group, you must ensure that there is opportunity to for you to provide evidence for all aspects of the standards.

ABOUT THE GAME! ASSESSMENT TASK

Achievement standard:	91879 & 91877
Standard title:	Develop a proposal for a digital outcome (3 credits) Develop a digital outcome to manage data (4 credits)
Total credits:	7

TASK 1 – RESEARCH

1. Explore the topic (myths and legends) to find:
   a. details of some myths or legends that may interest your end users
   b. the role that myths and legends play in communicating the history of a people group
   c. techniques used in passing on a myth or legend.

2. Explore the context (computer games) and the issue (developing a game that encourages your end users to learn about the topic):
   a. research what makes games enjoyable to play
   b. analyse existing games to determine the attributes commonly found in computer games (such as genre, reward methods, feedback etc) that may interest your target audience.

3. Research your end users. This information will be useful when deciding on the specifications of your proposed game. You will need to look at:
   a. the general characteristics of your end users
   b. the common preferences of your target audience
   c. the social environment in which they are likely to play your game (eg at school)
   d. the device(s) they are likely to use to play the game
   e. The software you will use to develop the game.

TASK 2 – COLLECTING END USER OPINIONS

1. Conduct research into topic and computer game preferences of your chosen target audience:
   a. plan and create a survey to determine end-user requirements
   b. create a database to organise your data
   c. create sensible queries to help you analyse the data
   d. communicate your findings on common preferences as requirements for the proposal
   e. capture evidence of your tests of procedures and data that ensure data integrity
   f. provide evidence to show how you have addressed the relevant implications such as privacy, data security etc.

ABOUT THE GAME! ASSESSMENT TASK

Achievement standard:	91879 & 91877
Standard title:	Develop a proposal for a digital outcome (3 credits) Develop a digital outcome to manage data (4 credits)
Total credits:	7

TASK 3 – IDEA DEVELOPMENT

1. Identify two or three end users to act as stakeholders who will give feedback on your proposal and provide their names. These can be other students in your class or persons from outside the class environment. Your survey analysis may be helpful in identifying like-minded end users to act as stakeholders.
2. Create a few different game ideas (2 or 3), informed by Task 1 and present these to your chosen end users for feedback. These are a quick, rough description of the proposed game idea. You can use annotated diagrams or sketches or text descriptions. These need to be detailed enough to show your idea but should not map out the entire development of the game.
3. Test your ideas by asking your stakeholders specific questions and use their opinions to assist you to choose ONE idea to develop further.

TASK 4 – PROPOSAL

1. Develop a proposal for your chosen idea, based on your research and feedback provided by your stakeholders. Your proposal should include (at least):
   a. A statement of the issue you are addressing
   b. Who your target end-users are (informed by your work in Task 1)
   c. Your game idea, including rough design and the purpose of the game
   d. An outline of any attributes you have identified that your game needs to include (this could include both images and text)
   e. The requirements you have identified from these attributes
   f. A description of how the requirements of the project will be able to be delivered in the established timeframe (the scope of the project)
   g. Any resources required that you have identified (for example, people, equipment)
   h. Justification of why your solution should be developed and how it meets the needs of your target end users

Ensure that you follow standard codes of practice. For example:

- If using photographs or illustrations in your work, use only those that you have taken yourself or have the rights to use (copyright and intellectual property compliance).
- Do not use photographs of people unless you have their permission to do so.

ABOUT THE GAME!
ASSESSMENT TASK

Achievement standard:	91879 & 91877
Standard title:	Develop a proposal for a digital outcome (3 credits) Develop a digital outcome to manage data (4 credits)
Total credits:	7

HAND IN

1. Your research identifying and detailing your target audience and the resources required by the project (survey, database with analysis and presentation of findings) including:
   a. Evidence of data integrity and data testing.
   b. Evidence that codes of practice for data used in this assessment have been followed and how you addressed these.
   c. Evidence of iterative improvement the database.
2. Who your stakeholders are and how they were chosen. (Task 3)
3. Your rough game ideas, your stakeholder feedback (via your questions) and your chosen idea. (Task 4)
4. Your proposal for the game you intend to make. (Task 5)

ABOUT THE GAME! TEACHER GUIDELINES

The following guidelines are supplied to support 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 standard/s. 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.

Please be aware that NZQA have read the assessment task but the task will still need to be checked by the teacher using the assessment to ensure it meets all requirements.

CONTEXT/TE HOROPAKI

This activity requires students to investigate games, conduct an opinion poll and analyse the results to determine requirements that should be included in the game they will be creating in the next assessment. This will help them to develop an initial proposal for this project and choose some users to assist in its development.

Students will:

- do research into theory of game development, existing games and myths and legends, to determine possible requirements for their game
- survey student opinions on their findings to determine which requirements they need to include
- create a database to assist in analysing their collected data and present the findings
- create a range of ideas for the game
- decide on one idea to develop with the help of a small group of end users
- create a proposal for the project

CONDITIONS/NGĀ TIKANGA

It is recommended that students should have regular progress checks with their teacher as they work through this assessment activity.

Where a group approach is used for the game outcome development, each student will need to create their own portfolio demonstrating how they individually developed their own digital outcome to manage data, and their own proposal for the game.

You need to guide the students in how you want them to capture evidence of their practice. This could be a written portfolio, electronic documents or even short embedded videos if the teacher deems this appropriate. Annotated screenshots are also useful in explanations.

Conditions of Assessment related to this achievement standard can be found at http://ncea.tki.org.nz

RESOURCE REQUIREMENTS/NGĀ RAUEMI

Students will need access to the internet, digital devices and database software.

ABOUT THE GAME! TEACHER GUIDELINES

ADDITIONAL INFORMATION

Some suggestions for attributes to include in the game:

- contain multiple levels (minimum of 3)
- have multiple objects or sprites
- have some way for the player to interact with it (eg press keys, click mouse etc).
- use simple graphics, or open source alternatives. No sprite rips. Must not contain any copyright material (images, sounds, music, or even other people’s code (unless you acknowledge the code as theirs)
- make use of conditional statements & loops
- include a Title screen with some way for users to select to “Play” and “Quit”
- include a Game Over screen and a Win screen (if it fits the game), as well as the High Score chart (if it fits the game)
- include a help page which has instructions for the game

ASSESSMENT SCHEDULE: ABOUT THE GAME! AS 91879 – DEVELOP A DIGITAL OUTCOME TO MANAGE DATA

EVIDENCE/JUDGMENTS FOR ACHIEVEMENT/PAETAE	EVIDENCE/JUDGMENTS FOR ACHIEVEMENT WITH MERIT/KAIAKA	EVIDENCE/JUDGMENTS FOR ACHIEVEMENT WITH EXCELLENCE/KAIRANGI
The student has developed an outcome to manage data	The student has developed an informed outcome to manage data	The student has developed a refined outcome to manage data
The student has: • created a database to store and organise the data related to their target audience preferences about games • used appropriate tools and techniques to structure, organise, query and present data for purpose to the end user *For example (partial evidence)* The student has: • named fields appropriately • used appropriate data types and data formatting • created at least one query which demonstrates selection of a subset of the table data • performed a sort or filter of the data • summarised the data (through queries, formulae, functions or graphs/charts) for use in the analysis of the results • created a presentation that displays the summary of the data collected which shows common target audience preferences.	**The student has:** • used information from testing procedures to improve the quality and functionality of the outcome *For example (partial evidence)* The student has: • asked a volunteer to enter data into the database and made suggested improvements based on testing • changed the field types from text boxes to drop down lists to improve data integrity and make it easier for the user • run some test queries and realised they would require another field to be able to get the desired results. **The student has:** • structured, organised and queried the data logically *For example (partial evidence):* The student has: • named fields in a consistent style • used appropriate data types and data formatting, for example including validation, suitable field sizes • used well-constructed queries, formulae, functions.	**The student has:** • shown iterative improvement throughout the development and testing process *For example (partial evidence)* The student has: • created a sample data gathering questionnaire/survey and tested it with a few classmates before setting up the structure of their database. • improved the questionnaire after realising a key piece of data was not being clearly asked for. Then improved the structure of their database to include an additional field. • made iterative changes to their database formatting and queries as they have developed their outcome. They have changed the data type of a field to enable more logical extraction of data. • tested different versions of their method of presentation to determine which one format is the most effective and meaningful way to present the data.

The student has: • applied appropriate data integrity and testing procedures ***For example (partial evidence):*** The student has: • checked the information has been imported correctly into the database and there are no duplicate or blank records • checked that filters and queries display expected results • checked that the summary calculations produce the correct results • checked that the presentation displays the information clearly and accurately. **The student has:** • described relevant implications. ***For example (partial evidence):*** The student has: • asked for permission from the participants to use their data in the summary of the findings • understood that the database may contain personal information and shouldn't be shared with others. • used correct field types that make it easier for the user and improve data integrity.	**The student has:** • addressed relevant implications. ***For example (partial evidence):*** The student has: • ensured that the database is stored securely in their network/cloud drive and that personal data in the database is not accessible to others • ensured that data displayed in their findings report would not make any single participant's data recognisable • used imagery/visuals they have created themselves or used those from the open-clip art library to respect intellectual property • developed their findings report with fonts that appeal to high school students (their target audience) • ensured the language used is easily readable by both students from the target audience and other interested adults.	**The student has:** • presented the data effectively for the purpose and to meet end user requirements. ***For example (partial evidence):*** The student has: • collected data in such a way that it was able to be effectively analysed and summarised (e.g. multi-choice, drop down lists, check boxes) • checked there are no grammatical or typographical errors • checked any imagery/visuals are not pixelated and accurately represent the data • checked the layout demonstrates effective application of design principles.

Final grades will be determined on a holistic examination of the evidence provided against the criteria in the achievement standard.

All supporting materials are supplied with this programme and can be found on the TKI website.

ASSESSMENT SCHEDULE: ABOUT THE GAME! AS91877 – DEVELOP A PROPOSAL FOR A DIGITAL OUTCOME

EVIDENCE/JUDGMENTS FOR ACHIEVEMENT/PAETAE	EVIDENCE/JUDGMENTS FOR ACHIEVEMENT WITH MERIT/KAIAKA	EVIDENCE/JUDGMENTS FOR ACHIEVEMENT WITH EXCELLENCE/KAIRANGI
The student has developed a proposal for a digital outcome	The student has developed an informed proposal for a digital outcome	The student has developed an effective proposal for a digital outcome
The student has: • proposed outcome statement includes problem and/or issue, scope, purpose and context. The student has explored myths and legends and existing computer games to determine positive and negative attributes of games and ideas which could be included in their own game. The student has created a survey to gather stakeholder preferences on myths and legends, computer games. The student then analysed the data to find the most popular preferences The student has provided a more detailed outline on what the proposed digital outcome is using sketches, storyboards, text descriptions, pseudo-code or flow charts to show what they game will look like on screen, and roughly how the game will work ***For example (partial evidence)*** *"I am going to make a computer game to help promote the Myth/Legend "…". Many students at our school don't know this story and I would like to make this game to bring it to the attention of the students. I want them to be able to "…" so that they will understand …". They can play my game at …*	**The student has:** • improved the proposal after feedback. They have gone through an improvement process based upon feedback from chosen stakeholders and made relevant changes to their proposal • consulted key stakeholders more than once to gain their opinions on decisions such as what sprites to include, what techniques to use, etc. ***For example (partial evidence)*** *After talking to my stakeholders … it would be better to use… in my game as this will be a more effective way to … As a result, I have decided to…*	**The student has:** • provided justifications with evidence for different aspects of their proposal, they have provided more than their own opinion, refined their proposal based on the feedback of others on their ideas, and provided justification and evidence of meeting the overall requirements ***For example (partial evidence)*** *"… although my stakeholders wanted …, my survey of target users suggested … As a result, the idea will now include … This addition will help to really get the message across.* *… should be an effective strategy because …* *This approach should be successful because …"*

The student has: • outlined the requirements that will make this project successful. The student has provided a relevant list of the requirements to make the digital outcome fit for purpose ***For example (partial evidence):*** Python: used to code my program so the game functions properly and can be played on … Gimp: an open-source art program to edit my images so the game characters look good Hardware: to play my game, students will need … Arcade techniques: platform code (or walls, etc) **The student has:** • stated who their end users are and outlined some expectations based upon evidence from research ***For example (partial evidence):*** *This game will be made for the students at this school this school in "…" most of them can access the game on …* *The students I surveyed wanted … included in the game. They also wanted the game to …*	**The student has:** • ensured the proposal shows clear links between the requirements, specifications, end users and resources required. The student has shown consideration for the end user's needs, the available resources (ie time/expertise etc) and the context in their development of their proposal. Their proposal clearly addresses the problem, issue, purpose or context. ***For example (partial evidence):*** *I will be using Python and … because … My target audience have indicated that they would like me to … I know this because when I asked them …, most of them said …* *My game also needs to have … This is because … said that … Also, from my survey I need to include …* *I will use … code from Arcade so that I can have … in my game. This will be necessary because my users wanted …* *I have decided that my game needs to have a start screen, help screen and … levels. It also needs to have … This is because … said that … Also, from I survey I need to include …*	

The student has: • outlined the resources required (people, equipment, time). The student lists the resources required for the digital outcome, outlines what they are, and why they are needed. ***For example (partial evidence):*** *I will need access to Python, Pygame, Arcade and Wing 101 at home and at school to code and test my game. I am familiar with the software and it is free, so I don't have to buy a copy to use at home. It lets me code in Python, and I can easily run the code as I develop to check for errors.*		

Final grades will be determined on a holistic examination of the evidence provided against the criteria in the achievement standard.

All supporting materials are supplied with this programme and can be found on the TKI website.

GAME ON!
ASSESSMENT TASK

Achievement standard:	91883 & 91884
Standard title:	Develop a computer program (4 credits) Use basic iterative processes to develop a digital outcome (6 credits)
Total credits:	10

OVERVIEW

This assessment activity requires you to develop a small interactive computer game that appeals to a broad target audience (eg, year 11 students), which will assist them to learn about a myth or legend.

You should have already created a proposal for your game, so it is now time to develop your game program, using the appropriate programming language you have selected.

HOW WILL YOU BE ASSESSED?

You are going to be assessed on how you develop your game, including testing and trialling your prototype.

This is an individual assessment task, so if you are working as a collaborative group, you must provide evidence of your own work for all aspects of the standards.

TASK

You need to:

1. Create an overall plan to develop the code by breaking the task down into smaller sub-tasks that can be coded, tested and debugged, prioritising the sub-tasks for development, and creating a timeline to help you track your development process.
2. Plan how the code for each sub-task will work, research techniques that could improve this sub-task and create a test plan to confirm that your program works for expected, boundary and relevant boundary cases.
3. Develop and test the code using your test cases, being sure to comment your code meaningfully.

HAND IN

- Your plan for program development
- Your program testing evidence
- All the versions of your program.

Note: You are doing one project, so there is no need to separate out the evidence by standard. Please use headings and subheadings so that your teacher can find all the relevant evidence required for the three assessments.

GAME ON! TEACHER GUIDELINES

The following guidelines are supplied to support 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 standard/s. 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.

Please be aware that NZQA have read the assessment task but the task will still need to be checked by the teacher using the assessment to ensure it meets all requirements.

CONTEXT/TE HOROPAKI

This assessment activity requires students to refine their proposal for a computer game and develop a basic computer program that is suitable for the intended audience. Students should demonstrate the use of basic iterative processes to plan, trial, and test their computer program.

Students will investigate resources and techniques, collecting end-user feedback on these decisions to refine their proposal. They will also accurately plan and construct a basic computer program and improve its quality in an ongoing manner. They need to ensure that they:

- Plan their process
- Test the outcome thoroughly
- Describe and address relevant implications and end-user considerations
- Complete a final proposal that defines the prototype they have developed.

CONDITIONS/NGĀ TIKANGA

It is recommended that students should have regular progress checks with their teacher as they work through this assessment activity.

The format of the final outcome is an interactive game.

If the project is done collaboratively, the teacher needs to ensure that it is broken up into sub-projects so that each student can provide evidence of meeting the standards in their portion of the project.

Conditions of Assessment related to this achievement standard can be found at http://ncea.tki.org.nz/

RESOURCE REQUIREMENTS/NGĀ RAUEMI

Students will need access to the internet, computers with Python 3.x, Pygame and Arcade installed, a text editor or integrated development environment tool such as Wing 101 and word-processing software to complete their documentation, Snipping Tool, and other recording software if desired.

ADDITIONAL INFORMATION

Teacher Support information

NB: Before attempting this assessment, students should have completed the About that Game assessment. Students should also have followed a program of the teaching and learning of the knowledge, concepts and skills required to prepare students for this assessment.

Planning:

Students should plan how they will develop their basic computer program. The planning should include research into resources they would like to include in their game and techniques they would like to use (eg, sprite sheets (copyright-free or Creative Commons), code for creating platform or maze games, etc). Their planning should also include evidence of breaking down of component parts that need to be included in their outcome. For example, when developing a basic computer program for the game, the components may include the start screen, the game loop, class code for platforms, walls or sprite-sheet handling, handling (and validating) user input, providing feedback to users, etc. This could involve using programming techniques, the use of exceptions where required, using loops or functions allowing students to avoid repeating code unnecessarily, variables, data collections and various user interface elements.

Planning should be undertaken in a manner that suits the outcome and could include flow-diagrams, pseudo code, to do lists, sprint lists, sketches, wireframes, storyboards or mock-ups. Students may use online interactive or collaborative planning tools, but this is not necessary.

Trialling:

Students should trial the components to be included and refine their outcome based on evidence of their trialling. Students should also provide evidence that they have planned out expected, boundary and unexpected use cases to test the outcome, and carry out testing iteratively to improve and refine their outcome. The final prototype should include evidence that the student has recognised and addressed a range of relevant social implications and end-user considerations.

Outcome:

Students may produce an individual basic computer program that is appropriate for the teaching and learning programme. Teachers should ensure the rigour of the outcome is appropriate for level 6 of the NZ Curriculum. The code in the basic computer program that the student is being assessed on should be original coding, which has been developed by the student. However, they may also incorporate other open-source or royalty-free media that they have not developed, as appropriate to the outcome and they may use program code from Arcade resources, but it should be acknowledged as such. For example, they may include an open-source or royalty-free sound track or images that they have not developed but have the permission (or appropriate license type) to include in the outcome. Students may choose to make a text based quiz or text adventure game, or animated game using Scratch or GameFroot instead of using Arcade and Pygame.

In the evidence of process produced by the students, evidence for the refined and completed proposal should also be shown. It is not necessary to burden the student with submitting separate evidence for each assessment, as long as all the evidence is present somewhere in the evidence that students submit to satisfy all three assessments.

GAME ON! ASSESSMENT SCHEDULE: AS 91884 – USE BASIC ITERATIVE PROCESSES TO DEVELOP A DIGITAL OUTCOME

EVIDENCE/JUDGMENTS FOR ACHIEVEMENT/PAETAE	EVIDENCE/JUDGMENTS FOR ACHIEVEMENT WITH MERIT/KAIAKA	EVIDENCE/JUDGMENTS FOR ACHIEVEMENT WITH EXCELLENCE/KAIRANGI
Use basic iterative processes to develop a digital outcome.	Use basic iterative processes to develop an informed digital outcome.	Use basic iterative processes to develop a refined digital outcome.
The student has: • planned a digital outcome to address a problem, need, opportunity or interest. *For example (partial evidence)* The student has researched issues relating to games, examined user preferences and planned how they were going to incorporate these through creating sketches and technique testing programs. The student has used an online tool to plan the game overall and each identified component. **The student has:** • developed the digital outcome by decomposing the problem *For example (partial evidence):* The student has decomposed their basic computer game into the components that needed to be developed and tested, such as user interface, game loop, walls, user-input handling, etc.	**The student has:** • used information from testing and trialling to improve the outcome *For example (partial evidence)* The student has provided screen shots with annotations that shows the improvements in the game mechanics after making changes. They have also provided a short video or write-up to demonstrate improved functionality after correcting a bug in the code. **The student has:** • trialled multiple components and/or techniques and selected those that ensure the outcome functions as intended *For example (partial evidence)* The student trialled two different techniques for performing a particular aspect of their game and selected the choice that did not cause functionality issues. The student trialled two different techniques for handling user input and chose the one that is most efficient.	**The student has:** • applied information from planning, testing and trialling of components to develop a high-quality outcome. *For example (partial evidence)* The student has provided evidence that their planning has allowed them to meet project timelines and include all the planned-for components and information. Their outcome functions as intended and has no obvious errors in functionality or presentation of the information. Evidence gained from trialling and thorough and organised testing has been integrated into the outcome in an ongoing manner to ensure that the outcome is of high quality, including aesthetics, functionality and usability.

The student has: • planned and trialled components of the outcome in an iterative manner ***For example (partial evidence)*** The student has planned and tested the code for the user interface. They next planned and tested the main game loop and then adding platforms, etc. They checked that the game is easy to use and understand. Each component was planned and tested in an iterative manner until the final game was produced. **The student has:** • tested that the programming outcome functions as intended ***For example (partial evidence)*** The student shows evidence of testing of the functionality of the game with various users throughout development to ensure that the game worked as expected.	**The student has:** • addressed the social implications and end-user considerations that are relevant to the outcome. ***For example (partial evidence)*** The student has provided screen shots with annotations that shows the improvements in the game mechanics after making changes. They have also provided a short video or write-up to demonstrate improved functionality after correcting a bug in the code. **The student has:** • trialled multiple components and/or techniques and selected those that ensure the outcome functions as intended ***For example (partial evidence)*** The student trialled two different techniques for performing a particular aspect of their game and selected the choice that did not cause functionality issues. The student trialled two different techniques for handling user input and chose the one that is most efficient.	

The student has: • described the relevant implications *For example (partial evidence):* The student the game preferences of their target audience to determine their game preferences. The student recognised that it is unethical to use copyrighted sprites or images in the game. They have recognised that game screen layout will affect the enjoyment of using the game. However, the student may not have chosen the best solution to address the considerations or could have more fully addressed these considerations.	**The student has:** • addressed the social implications and end user considerations that are relevant to the outcome. *For example (partial evidence)* The student addressed the fact that it is unethical to use copyrighted images by making up their own images or using Creative Commons material that has been correctly attributed or by using images that are copyright-free. They have addressed usability and aesthetic considerations through testing their game with a range of end-users and responding to user feedback.	

Final grades will be determined on a holistic examination of the evidence provided against the criteria in the achievement standard.

All supporting materials are supplied with this programme and can be found on the TKI website.

GAME ON! ASSESSMENT SCHEDULE: AS 91883 – DEVELOP A COMPUTER PROGRAM

EVIDENCE/JUDGMENTS FOR ACHIEVEMENT/PAETAE	EVIDENCE/JUDGMENTS FOR ACHIEVEMENT WITH MERIT/KAIAKA	EVIDENCE/JUDGMENTS FOR ACHIEVEMENT WITH EXCELLENCE/KAIRANGI
Develop a basic computer program.	Develop an informed computer program.	Develop a refined computer program.
The student has: • written a basic program that performs the specified task, using a suitable programming language ***For example (partial evidence)*** The student has written a simple, functional game program in a computer language. The program may not be structured very well. The game works as expected but they may only have included two functions without parameters. **The student has:** • set out the program code clearly and documented the program with comments ***For example (partial evidence):*** The student has written the code in a procedural manner, including use of loops and conditions. Comments are present but may not be particularly descriptive or frequent. eg, # this code creates the game loop	**The student has:** • documented the program with variable names and comments that describe code function and behaviour ***For example (partial evidence)*** The student has frequent, clear comments throughout the code that helps to describe relevant functions or sections of code. The variable names clearly describe the data they hold. eg. # this function tests that the user input is a number. It will continue to ask for input until the input is a number. e.g. answer = is_a_number(choice[i])	**The student has:** • ensured that the program is a well-structured, logical solution to the task ***For example (partial evidence)*** The student's final program consists of multiple general-purpose functions that pass parameters when appropriate. The program flow is clear and there is no unnecessary or redundant code. Functions are well named so that they are self-documenting. Student code is laid out so that all the functions appear before the main program. The student has used parameters and arguments to pass data into functions and has returned values for functions, using local variables. *Function to test input is an integer is named is_num* The code is clean, concise, efficient and easily readable. The main program may be short and consists of multiple reusable user defined functions which do most of the logic and processing.

GAME ON! ASSESSMENT SCHEDULE: AS 91883 – DEVELOP A COMPUTER PROGRAM

The student has: • tested and debugged the program to ensure that it works on a sample of expected cases. *For example (partial evidence):* The student has shown some evidence of expected use cases that were used to test and debug the program and shown that the program works when the user inputs data that is expected. Testing may be trial and error rather than clearly thought out. The student has written about some expected case data and has tested what happens when the program is run. Such testing may be observed by the teacher, presented in table form with minimal notes, or simply explained using examples.	**The student has:** • followed conventions of the chosen programming language *For example (partial evidence)* The student has followed most common programming conventions for their chosen language: • Python files and functions contain a docstring explaining the purpose of the program or function. • Constants are ALL_CAPS with underscores separating the words if required. • Variable names use underscore rather than Camel case. • Functions appear before loose lines of code and main section of code is all at bottom, not between the functions, thus making the program easier to read.	**The student has:** • made the program flexible and robust *For example (partial evidence)* The student has created a program with a flexible structure that can be easily used in another program by including the code. The program uses modules, actions and control structures effectively and without unnecessary repetition. The program uses constants, variables and derived values in place of literals. Such variables appear at the top of the code, where they can easily be seen and changed should those values change over time. The student has named SCREEN_NUM as a constant values rather than using 5 (or whatever value a correct answer rates).

	The student has: • tested and debugged the program in an organised way to ensure that it works on a sample of both expected and relevant boundary use cases. *For example (partial evidence)* The student tested frequently during development (observed) and the final program works when the user inputs the data that is expected and checks or handles when the data is outside specific thresholds. The student may have kept some form of notes showing what was tested and the outcome of that testing. Test use cases by the student include expected and boundary use cases. In a situation where a user is asked to "choose the answer 1,2,3 or 4", the program checks that the input is above 0 and under 5 The program might not correctly handle unexpected data and fail when a word is typed in where a number was expected.	**The student has:** • comprehensively tested and debugged the program. *For example (partial evidence)* The student makes sure the program checks the validity of input data and deals with expected, boundary and unexpected or invalid data. All use cases have been tested, including testing each condition within their code. Testing has been done in a systematic way. Test cases have been well thought out and notes may have been made showing that the code works as expected for all use cases. They have used "try-except" error handling to ensure that program handles even unexpected events or inputs.

Final grades will be determined on a holistic examination of the evidence provided against the criteria in the achievement standard.

All supporting materials are supplied with this programme and can be found on the TKI website.