INTERNAL ASSESSMENT RESOURCE

Digital Technologies

|  |  |  |  |
| --- | --- | --- | --- |
| **AS** | 91907 | **Credits** | 6 |
| **Level** | NCEA level 3 | **Assessment type** | Internal |
| **Date published** | 30 Mar 2018 | **Version** | 1 |

# Ngā whāinga paetae **Achievement Criteria**

|  |  |  |
| --- | --- | --- |
| **Paetae • Achieved** | **Kaiaka • Merit** | **Excellence** |
| Use complex processes to develop a digital technologies outcome. | Use complex processes to develop an **informed** digital technologies outcome. | Use complex processes to develop a **refined** digital technologies outcome. |

## Tō putanga Your grade

|  |  |  |  |
| --- | --- | --- | --- |
| Kāore i eke / NA | **Paetae / Achieved** | **Kaiaka / Merit** | **Kairangi / Excellence** |

* *There is no resubmission for this assessment.*

## Ngā tākapu Comments

|  |
| --- |
|  |

# Paearu Paetae **Assessment Schedule**

Use this marking schedule to keep track of what requirements you have met while developing your program. Type an X in the boxes to mark each criterion complete.

## Paetae Achieved

|  |  |  |  |
| --- | --- | --- | --- |
| **Criteria** | **Example evidence** | **Student** | **Teacher** |
| The student has used recognised and appropriate project management tools and techniques to plan the development of a digital technologies outcome. | The student has used at least one project management system, including:   * GitHub (Projects, Codespaces, etc.) * Microsoft Project * Paper-based project management (post-it notes, journal, etc.). |  |  |
| The student has used at least one project management technique, including:   * Agile * Kanban or scrum boards * Version control software, such as Git * Collaboration tools, such as GitHub (Issues, Discussions) or Teams * Managing assets, such as creating an appropriate folder structure to hold files, media, etc. |  |  |
| The student has decomposed the digital technologies outcome into smaller components. | The student has created either a list, table, or mind map of at least 10 components. This could be 10 separate components or 3 main components with 3 to 4 subcomponents each, etc. |  |  |
| The student has trialled components of the outcome. | The student has recorded evidence of trialling different at least 2 components. |  |  |
| The student has used trialling to inform decision making about components, such as the choice of components, techniques, designs, etc. |  |  |
| The student has tested that the digital technologies outcome functions as intended. | The student has recorded evidence of testing that the outcome works on **expected**, **boundary**, and **invalid** cases. The major focus is that the outcome functions as intended. |  |  |
| The student has addressed relevant implications. | The student has identified at least three relevant implications. |  |  |
| The student has defined the implications and explained how they are relevant to the outcome. |  |  |
| At the start of the project, the student has stated how they plan to address the implications. |  |  |
| By the end of the project, the student has addressed the implications according to their plan. |  |  |

## 

## Kaiaka Merit

|  |  |  |  |
| --- | --- | --- | --- |
| **Criteria** | **Example evidence** | **Student** | **Teacher** |
| The student has effectively used project management tools and techniques to manage development, feedback, and/or collaborative processes. | The student has effectively used project management tools, such as project boards and version control tools, to manage the development of their project. |  |  |
| The student has used project management tools to adjust timelines and resources. This might include the addition, removal, or modification of components. |  |  |
| If working in a pair/group, the student has made use of collaboration tools to ensure good communication between group members, that deadlines are met, etc. |  |  |
| The student has effectively trialled multiple components and/or techniques. | The student has effectively used trialling to inform decision making about multiple components across the whole project. |  |  |
| The student has trialled more than one way to solve a given sub-problem and selected the best way. |  |  |
| The student has explained their decisions based on the results of their trialling. |  |  |
| The student has effectively used information from testing and trialling to improve the functionality of the digital technologies outcome. | The student has recorded evidence of testing in a timely manner. When an issue is detected, it can be resolved either within the same sprint or next. |  |  |
| If applicable, the student has explained why some issues might take longer (i.e., more than one sprint) to resolve. |  |  |

## 

## Kairangi Excellence

|  |  |  |  |
| --- | --- | --- | --- |
| **Criteria** | **Example evidence** | **Student** | **Teacher** |
| The student has synthesised information gained from the planning, testing, and trialling of components. | The student has reflected on how testing and trialling has influenced the development of the outcome over time, especially when components turn out differently than initially planned. |  |  |
| The student has discussed how information gained from planning, testing, and trialling of components led to the development of a high-quality digital technologies outcome. | The student has justified the outcome’s fitness for purpose and any features/components that make the outcome high-quality. Unlike explanation, justification requires evidence; this should come from the results of testing, trialling, and how your outcome addresses relevant implications. |  |  |

# Ngā tikanga Conditions of Assessment

## Timeline

* You are expected to work on this assignment in class and at home.
* During part of this assessment, your classwork will be dedicated to AS91908. You will be expected to manage your time to continue working on this assessment at home, during study spells, etc.
* Your kaiako can offer extra time to catch up, get help, etc.
  + **RJA**: Wednesday spell 5 in Room 10
  + **VTA**: Thursday lunchtime in Room 7
* There will be assessment check-ins for your kaiako to give you feedback on your documentation:
  + Term 3, Week 2
  + Term 3, Week 4
* This assessment is due:
  + Term 4, Week 3
  + Friday, the 27th of October 2023

## As part of this assessment, you may

* Consult notes provided by your kaiako.
* Consult your own work stored on GitHub.
* Consult the internet for help with developing your outcome.
  + You must cite the sources of any code that you use, either as:
    - Comments in your code
    - Links in your project documentation
* Consult with other ākonga. However, you may not submit their work as your own.

## You may not

* Submit work that you did not create directly as your own.

# 

# He Mahi **Task**

**READ ALL INSTRUCTIONS CAREFULLY**

## Outcome requirements

You will develop a game, program, or website to fulfil one of the [outcome ideas](#_953vl09orjiu) in the next section. Each outcome idea can be developed into whatever kind of outcome you see fit. For example, you could make a:

* Python-based game
* Program in any programming language
  + If you wish to use a non-Python language, you are responsible for learning it if you aren’t already familiar with it.
  + Speak with your kaiako first about whether it is advisable to use a language other than Python.
* Game using a game-making engine. Many such engines use node-based programming which can be easier.
* Website or web-based game

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Python game** | **Program** | **Game engine** | **Web-based** |
| **Requires programming** | Yes | Yes | Not mandatory | Yes |
| **Taught at OC** | Yes | No | No | No |
| **Requires graphics** | Yes | No | Yes | Yes |
| **Programming language(s)** | Python | Python, Swift, Rust, C#, Java | **Godot**: GDScript  **Unity**: C#  **Unreal Engine**: C++  **Game Maker**: C++  **Minecraft Education**: Python  **Roblox**: Lua | JavaScript, TypeScript, PHP, Ruby, Python |
| **Possible frameworks** | PyGame  TkInter  PySide 6 | TkInter  PySide6  Pandas  Numpy | Godot  Unity  Unreal Engine  Game Maker  Construct  Minecraft Education  Roblox | Bootstrap  Rails  React  WordPress  Drupal  Joomla |

## 

# Outcome ideas

## Tabletop game (for budding game developers)

Develop your own version of a traditional tabletop game — adding your own twist.

As part of this, you will identify the game’s rules; if necessary, you might simplify them.

You will then design and implement the game to be played by one or two players (you do not need to create an AI player, although you can if you want).



You may choose from one of the following **OR** you can choose your own (after discussion with your kaiako):

|  |  |  |  |
| --- | --- | --- | --- |
| [Yacht](https://www.youtube.com/watch?v=olES-DowvEk&pp=ygUgZ2F0aGVyIHRvZ2V0aGVyIGdhbWVzIHlhY2h0IGRpY2U%3D) | [Reversi](https://www.youtube.com/watch?v=QioKbksiJnA) | [Last Card](https://www.youtube.com/watch?v=0pDXIWSOFg4) | [Mastermind](https://www.youtube.com/watch?v=Dn0iqlY5tMU) |
| [Mancala](https://www.youtube.com/watch?v=2T0jeY-BxU4) | [Gomoku](https://www.youtube.com/watch?v=-KD743yNDHc) | [Durak](https://www.youtube.com/watch?v=3JagmUmUJOc) (дурак) | [Backgammon](https://www.youtube.com/watch?v=KDvvKWi0ijs) |
| [Noughts and Crosses](https://www.youtube.com/watch?v=gk9KcRm5cIo&pp=ygUhZ2F0aGVyIHRvZ2V0aGVyIGdhbWVzIHRpYyB0YWMgdG9l) | [Go](https://www.youtube.com/watch?v=cFVM_hwh56s) | [President](https://www.youtube.com/watch?v=HYcH0bzsmsI&t=1s) (Ps & As, Scum) | [Draughts](https://www.youtube.com/watch?v=ScKIdStgAfU&pp=ygUeZ2F0aGVyIHRvZ2V0aGVyIGdhbWVzIGRyYXVnaHRz) (Checkers) |
| [Dots and Boxes](https://www.youtube.com/watch?v=uWrKllUAfm8) | [Four in a Row](https://www.youtube.com/watch?v=frousevsm34) | [Koikoi](https://www.youtube.com/watch?v=gkjjB3fYD2E&pp=ygUcZ2F0aGVyIHRvZ2V0aGVyIGdhbWVzIGdvbW9rdQ%3D%3D) or [Go-Stop](https://www.youtube.com/watch?v=HM8LWmb5Ky0) (Hanafuda) | [Shogi](https://www.wikihow.com/Play-Shogi) |

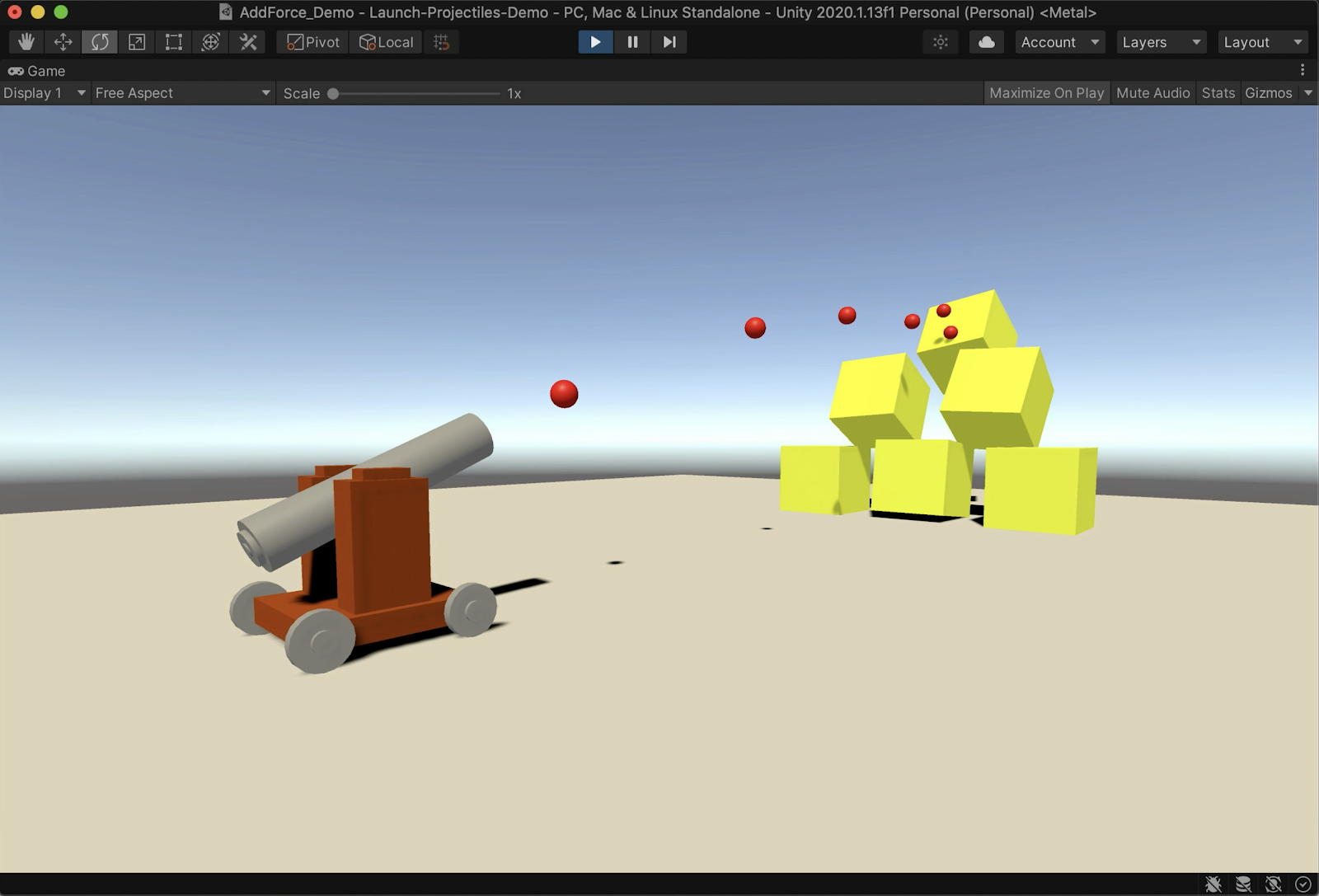
Twists could include:

1. Different shaped boards.
2. Different number or kind of cards.
3. A particular theme for the graphics, gameplay, etc.

## Projectile motion simulator (for physics students)

Develop a game or program based on a simulation of projectile motion. The outcome should be able to demonstrate:

1. A projectile launched from ground level at an angle between 0° and 90°
2. A projectile launched from an elevated position at an angle between 0° and 90°



The user should be able to specify:

1. The launch speed.
2. The launch angle.
3. The height of the elevated platform from the ground

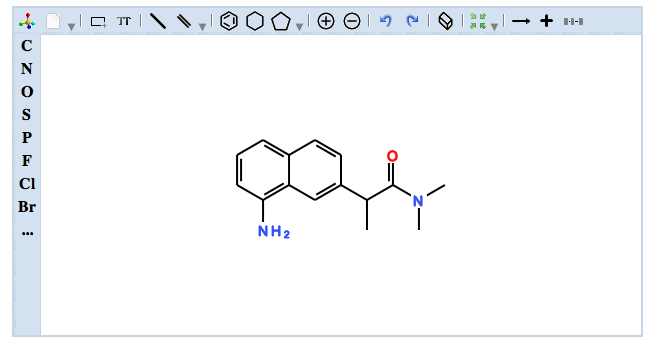
The program will display:

1. The maximum height reached by the projectile.
2. The total time of flight
3. The range (horizontal distance) travelled.
4. A real-time simulation of the projectile from firing to landing.

## 

## Chemical formula viewer (for chemistry students)

Develop a program that takes a chemical formula as an input and visualises the corresponding molecular structure. Your program should account for covalent bonding and the spatial arrangement of atoms.



Your program must:

1. accept a chemical formula as input. For instance, if the user inputs "H2O", the program should interpret this as two hydrogen atoms and one oxygen atom.
2. show a visualisation that clearly distinguishes between single, double, and triple bonds.
3. handle at least the first 20 elements of the periodic table.

## 

## Other idea (for budding developers)

You may also propose your own idea to your kaiako, who will help ensure the task is:

* suitable work for an NCEA standard
* realistically achievable within the given time frame
* able to meet the criteria for the standard.



One idea is to develop a feature for a Free and Open-Source Software (FOSS) project.

In this case, you must come up with a new feature to develop from scratch, separately from any existing on-going work in the project. Otherwise, you will not be able to complete all steps needed for this standard (decomposition, planning, etc.).

You would not be expected to contribute your feature back to the project in order to pass the standard, although you should comply with the project’s licence.

# Ngā herenga Requirements

## Authenticity requirements

As part of this project, you will be expected to use GitHub for version control. All your commits are evidence of your contribution to a project. If you commit work that is not your own without appropriate citation, you risk the project receiving a Not Achieved grade for plagiarism.

If you work in pairs/groups, each individual contributor to the project must:

1. Decompose the outcome with their group.
2. Contribute to a shared project board.
3. When components are moved from To Do to In Progress, clearly mark the components that person are working on (as opposed to other members of the group).
4. Commit to GitHub regularly, as evidence that they are contributing an appropriate amount of work to the project.
5. Reflect on *their own* contributions.

## Testing requirements

There are no specific requirements around testing in the standard — the act of testing is only required for paetae/achieved, and only to ensure the outcome functions correctly.

However, the development of a **high-quality outcome** is a requirement for kairangi/excellence. A high-quality outcome should not exhibit unexpected behaviour or crash, so you should continue to record evidence of boundary and invalid case tests.

You can record your testing in two ways. In either case, you must show evidence of testing *during development*, not just at the end:

1. Take screenshots and fill in the testing table in the documentation.
2. Record videos showing that the program works on expected, boundary, and invalid cases

Student Work

Throughout this document, you must fill in all the boxes highlighted in this colour. Failing to fill in these boxes means you risk receiving not achieving this standard.

## Problem Statement A

|  |
| --- |
| *Type an X in the box for the outcome idea you will develop in your project.* |

|  |  |  |  |
| --- | --- | --- | --- |
| **Tabletop game** | **Projectile simulation** | **Chemical formula viewer** | **Another project** |
| - | X | - | - |

## Decompose the outcome A

|  |
| --- |
| *Decompose your digital technologies outcome into smaller components. Rank the priority of each component. You can show this as:*   * *A to-do list.* * *A decomposition table.* * *A mind-map or brainstorm*   *Take a screenshot of your decomposition.* |

|  |
| --- |
| **Decomposition Screenshot** |
|  |

## Relevant implications A

|  |
| --- |
| *What relevant implications do you need to consider in the development of your outcome?*  *You must identify* ***at least*** *two implications for paetae/achieved. However, to ensure a high-quality outcome, you should identify as many as possible.*   1. *Cite a dictionary or industry-standard definition for each implication. Provide the source of each definition.* 2. *Describe how each implication is relevant to your project.* 3. *Explain how you plan to address each implication.* |

### Implication #1: FUNCTIONALITY

|  |  |
| --- | --- |
| **Define the implication here** | **Functionality** - The state of an app, website or database being useful. |
| **Describe how the implication is relevant to your outcome** | The program is based on being able to serve a purpose and be useful, this is because it is a tool. This is why it is relevant. |
| **Explain how you plan to address this implication** | As the program is fully based on having the purpose to provide data to the end user after end user inputs raw data, it needs to achieve this otherwise the program is useless. |

### Implication #2: INTELLECTUAL PROPERTY

|  |  |
| --- | --- |
| **Define the implication here** | **Intellectual Property** - Any of various products of the intellect that have commercial value |
| **Describe how the implication is relevant to your outcome** | As the program is a tool, it can be marketed to consumers. |
| **Explain how you plan to address this implication** | I would make the program open source for anyone to use as it is a useful tool that I'd like for anyone around the world to use. This allows people to use the IP but if it gets popular enough, I will have to patent it to make sure no one else takes it and makes you pay for it. |

### Implication #3: USABILITY

|  |  |
| --- | --- |
| **Define the implication here** | **Usability –** Allows end-users to easily access features, have error prevention, data recovery ect. |
| **Describe how the implication is relevant to your outcome** | The program needs to be easy to navigate and understand as it only has one purpose, allowing good usability allows end-users to have good user experience and efficient use. |
| **Explain how you plan to address this implication** | I will make the menus easy to understand, most things will be one or two clicks away from the main menu, text inputs will be very visible. |

### [**https://www.freybergdigital.co.nz/relevant-implications.html**](https://www.freybergdigital.co.nz/relevant-implications.html)

# Sprints

## Sprint 1

### Planning A

|  |  |
| --- | --- |
| **Start date**  When did this sprint start? | **End date**  When will this sprint end? |
| Start of assignment | 4/8/2023 |

### Components A

|  |
| --- |
| *Take a screenshot of your* ***entire*** *project board, showing what components you will work on. The components you plan to work on should be shown in the* ***In Progress*** *bucket.* |

|  |
| --- |
| **Project board screenshot** |
|  |

|  |  |
| --- | --- |
| **Explain why you have selected these components for this sprint** | This is the foundation of the program, the GUI will be built around it so I need to understand what it looks like and how it works. |

### Testing

|  |
| --- |
| *You must show evidence of testing that your program works on a sample of expected (paetae/achieved), boundary (kaiaka/merit), and/or invalid (kairangi/excellence) cases. This testing should be done over the course of development, not just at the very end.* |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test name** | **E/B/I** | **Input** | **Expected output** | **Actual output** (screenshot) | **Comments** |
| Showing graph and testing matplotlib.pyplot.subplots | E | Random plotted points | Graph that goes through [0, 1], [1, 3], [2, 5.5], [3, 7], [4,9] |  | Worked perfectly |
| Testing kinematics graphing calculations | E | 50m/s and 45\* angle |  |  |  |
| Testing animations | E | Inputting a few plots and graphing them using matplotlib.animation | A point be put down every 0,5 seconds |  | Works |
| Decreasing interval between points in order to get smoother animation | E | Interval = 0.033, increasing np.linspace num to 128 | Smooth animation to show parabola of kinematic formula |  |  |
| Accepting angles and speed of projectiles as text inputs | E | “80”, “30” |  |  |  |

### Evaluation M/E

|  |
| --- |
| *Reflect on what happened during this sprint. Think about what components you were able to develop/trial easily, what was trickier, and what might not have been completed at all.*  *In your project board, components that you completed should move to the Completed bucket. If any were not completed, they should either remain in In Progress if you intend to keep working on them, or get moved back to To Do if you plan to work on something else during the next sprint.* |

|  |  |
| --- | --- |
| **✔︎ Explain what components you completed in this sprint** | I finished the matplotlib animations, this allows a user to see in realtime the calculations taking place, the graph is run at 30FPS and the entire animations last for 4.26 seconds  I also finished the initial kinematics calculations such as d = vi \* t + ½ \* a \* t |
| **✘ Explain any components you were** not **able to complete this sprint, and why** | Adding initial height as part of the calculations, this would allow the user to specify the height at which the object starts, i was not able to start this as I didnt have the time. This will have to be done next sprint. |

|  |
| --- |
| *In your GitHub repository (not your Codespace), click on the commits symbol. Take a screenshot of the commits you made during this sprint. If you are working in a pair/group, take a screenshot of* ***all*** *commits, not just your own.* |

|  |
| --- |
| **Version control screenshot** |
|  |

(I just made a new repository for this)

## Sprint 2

### Planning A

|  |  |
| --- | --- |
| **Start date**  When did this sprint start? | **End date**  When will this sprint end? |
| 07/08/2023 | 21/08/2023 |

### Components A

|  |
| --- |
| *Take a screenshot of your* ***entire*** *project board, showing what components you will work on. The components you plan to work on should be shown in the* ***In Progress*** *bucket.*  *If you have finished any components, they should be shown in the Completed bucket. Components that you have not finished either remain in the In Progress bucket or return to the To Do bucket.* |

|  |
| --- |
| **Project board screenshot** |
|  |

|  |  |
| --- | --- |
| **Explain why you have selected these components for this sprint** | I will be mainly working on building the GUI text inputs, run graph function that displays the graph on the GUI and animating that graph. I chose these as I am ready to use the graphing class I created last sprint with parameters taken from the GUI text inputs to graph using user inputs. |

### 

### Testing

|  |
| --- |
| *You must show evidence of testing that your program works on a sample of expected (paetae/achieved), boundary (kaiaka/merit), and/or invalid (kairangi/excellence) cases. This testing should be done over the course of development, not just at the very end.* |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test name** | **E/B/I** | **Input** | **Expected output** | **Actual output** (screenshot) | **Comments** |
| Showing display to be used for GUI | E | nothing | White display |  |  |
| Rendering green boxes for data inputs | E |  | 3 green boxes rendered in a position in the window area |  |  |
| Labels for input data | E |  | Text next to each input box |  |  |
| Confirm button and text for running simulation | E |  | Button under input boxes for “start simulation” |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

|  |
| --- |
| *Reflect on what happened during this sprint. Think about what components you were able to develop/trial easily, what was trickier, and what might not have been completed at all.*  *In your project board, components that you completed should move to the Completed bucket. If any were not completed, they should either remain in In Progress if you intend to keep working on them, or get moved back to To Do if you plan to work on something else during the next sprint.* |

|  |  |
| --- | --- |
| **✔︎ Explain what components you completed in this sprint** | Explanation here |
| **✘ Explain any components you were** not **able to complete this sprint, and why** | Explanation here |

|  |
| --- |
| *In your GitHub repository (not your Codespace), click on the commits symbol. Take a screenshot of the commits you made during this sprint. If you are working in a pair/group, take a screenshot of* ***all*** *commits, not just your own.* |

|  |
| --- |
| **Version control screenshot** |
|  |

Sprint 3

### Planning A

|  |  |
| --- | --- |
| **Start date**  When did this sprint start? | **End date**  When will this sprint end? |
|  | 21/8/2023 |

### Components A

|  |
| --- |
| *Take a screenshot of your* ***entire*** *project board, showing what components you will work on. The components you plan to work on should be shown in the* ***In Progress*** *bucket.* |

|  |
| --- |
| **Project board screenshot** |
|  |

|  |  |
| --- | --- |
| **Explain why you have selected these components for this sprint** | I will be mainly working on GUI text inputs, as well as trialing the text inputs toward the end of the sprint. I chose this as I have all the backend required to run graphs and calculations |

### 

### 

### Trialling A/M

|  |
| --- |
| *In each sprint, at least three users should trial your outcome, either as it is developed or near the end of the sprint. As part of the standard, you need to trial* ***functionality*** *— the way the outcome works, how it achieves its goal, etc.*  *For Merit, you must trial multiple components over the lifetime of the outcome’s development. You can trial a given component more than once.*  *You should trial multiple approaches to a particular problem (*[*more details*](#q8pkvf8jfwdx)*). Decide on at least three approaches for a given problem, then gather feedback below.* |

|  |  |
| --- | --- |
| **Problem** | Explain the problem you are trying to solve |
| **Approach 1** | Explain, show, or link to your option here |
| **Approach 2** | Explain, show, or link to your option here |
| **Approach 3** | Explain, show, or link to your option here |

#### Feedback #1

|  |  |
| --- | --- |
| **Feedback giver name** | Sam McLiesh |
| **Which approach is best** | Animation of graph |
| **Explain why you think that approach is best** |  |
| **Suggest what you think could have improved the other approaches** | Add loading screen when graph is rendering, add confirmation that button is clicked, add units to input label |

#### Feedback #2

|  |  |
| --- | --- |
| **Feedback giver name** | Name goes here |
| **Which approach is best** | Response goes here |
| **Explain why you think that approach is best** | Explanation goes here |
| **Suggest what you think could have improved the other approaches** | Suggestions go here |

#### Feedback #3

|  |  |
| --- | --- |
| **Feedback giver name** | Name goes here |
| **Which approach is best** | Response goes here |
| **Explain why you think that approach is best** | Explanation goes here |
| **Suggest what you think could have improved the other approaches** | Suggestions go here |

### Testing

|  |
| --- |
| *You must show evidence of testing that your program works on a sample of expected (paetae/achieved), boundary (kaiaka/merit), and/or invalid (kairangi/excellence) cases. This testing should be done over the course of development, not just at the very end.* |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test name** | **E/B/I** | **Input** | **Expected output** | **Actual output** (screenshot) | **Comments** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

### Evaluation M/E

|  |
| --- |
| *Analyse the feedback that you received and make decisions about what you might change.*   1. *Explain if the responses were what you expected or not, and why.* 2. *Explain what changes you will make (such as additional features, different designs, etc.) in response to the feedback, and why.* 3. *Explain any proposals you will* ***not*** *make in response to the feedback, and why.* |

|  |  |
| --- | --- |
| **Explain if the responses were what you expected** | Explanation here |
| **✔︎ Explain changes you will make in response to feedback, and why** | Explanation here |
| **✘ Explain changes you will** not **make in response to feedback, and why** | Explanation here |
| **Explain how the changes have addressed or will address your outcome’s functionality** | Explanation here |

|  |
| --- |
| *Reflect on what happened during this sprint. Think about what components you were able to develop/trial easily, what was trickier, and what might not have been completed at all.*  *In your project board, components that you completed should move to the Completed bucket. If any were not completed, they should either remain in In Progress if you intend to keep working on them, or get moved back to To Do if you plan to work on something else during the next sprint.* |

|  |  |
| --- | --- |
| **✔︎ Explain what components you completed in this sprint** | Explanation here |
| **✘ Explain any components you were** not **able to complete this sprint, and why** | Explanation here |

|  |
| --- |
| *In your GitHub repository (not your Codespace), click on the commits symbol. Take a screenshot of the commits you made during this sprint. If you are working in a pair/group, take a screenshot of* ***all*** *commits, not just your own.* |

|  |
| --- |
| **Version control screenshot** |
| Screenshot here |

## Sprint 4

### Planning A

|  |  |
| --- | --- |
| **Start date**  When did this sprint start? | **End date**  When will this sprint end? |
| Start date here | End date here |

### Components A

|  |
| --- |
| *Take a screenshot of your* ***entire*** *project board, showing what components you will work on. The components you plan to work on should be shown in the* ***In Progress*** *bucket.* |

|  |
| --- |
| **Project board screenshot** |
| Screenshot here |

|  |  |
| --- | --- |
| **Explain why you have selected these components for this sprint** | Explanation here |

### 

### 

### Trialling A/M

|  |
| --- |
| *In each sprint, at least three users should trial your outcome, either as it is developed or near the end of the sprint. As part of the standard, you need to trial* ***functionality*** *— the way the outcome works, how it achieves its goal, etc.*  *For Merit, you must trial multiple components over the lifetime of the outcome’s development. You can trial a given component more than once.*  *You should trial multiple approaches to a particular problem (*[*more details*](#q8pkvf8jfwdx)*). Decide on at least three approaches for a given problem, then gather feedback below.* |

|  |  |
| --- | --- |
| **Problem** | Explain the problem you are trying to solve |
| **Approach 1** | Explain, show, or link to your option here |
| **Approach 2** | Explain, show, or link to your option here |
| **Approach 3** | Explain, show, or link to your option here |

#### Feedback #1

|  |  |
| --- | --- |
| **Feedback giver name** | Name goes here |
| **Which approach is best** | Response goes here |
| **Explain why you think that approach is best** | Explanation goes here |
| **Suggest what you think could have improved the other approaches** | Suggestions go here |

#### Feedback #2

|  |  |
| --- | --- |
| **Feedback giver name** | Name goes here |
| **Which approach is best** | Response goes here |
| **Explain why you think that approach is best** | Explanation goes here |
| **Suggest what you think could have improved the other approaches** | Suggestions go here |

#### Feedback #3

|  |  |
| --- | --- |
| **Feedback giver name** | Name goes here |
| **Which approach is best** | Response goes here |
| **Explain why you think that approach is best** | Explanation goes here |
| **Suggest what you think could have improved the other approaches** | Suggestions go here |

### Testing

|  |
| --- |
| *You must show evidence of testing that your program works on a sample of expected (paetae/achieved), boundary (kaiaka/merit), and/or invalid (kairangi/excellence) cases. This testing should be done over the course of development, not just at the very end.* |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test name** | **E/B/I** | **Input** | **Expected output** | **Actual output** (screenshot) | **Comments** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

### Evaluation M/E

|  |
| --- |
| *Analyse the feedback that you received and make decisions about what you might change.*   1. *Explain if the responses were what you expected or not, and why.* 2. *Explain what changes you will make (such as additional features, different designs, etc.) in response to the feedback, and why.* 3. *Explain any proposals you will* ***not*** *make in response to the feedback, and why.* |

|  |  |
| --- | --- |
| **Explain if the responses were what you expected** | Explanation here |
| **✔︎ Explain changes you will make in response to feedback, and why** | Explanation here |
| **✘ Explain changes you will** not **make in response to feedback, and why** | Explanation here |
| **Explain how the changes have addressed or will address your outcome’s functionality** | Explanation here |

|  |
| --- |
| *Reflect on what happened during this sprint. Think about what components you were able to develop/trial easily, what was trickier, and what might not have been completed at all.*  *In your project board, components that you completed should move to the Completed bucket. If any were not completed, they should either remain in In Progress if you intend to keep working on them, or get moved back to To Do if you plan to work on something else during the next sprint.* |

|  |  |
| --- | --- |
| **✔︎ Explain what components you completed in this sprint** | Explanation here |
| **✘ Explain any components you were** not **able to complete this sprint, and why** | Explanation here |

|  |
| --- |
| *In your GitHub repository (not your Codespace), click on the commits symbol. Take a screenshot of the commits you made during this sprint. If you are working in a pair/group, take a screenshot of* ***all*** *commits, not just your own.* |

|  |
| --- |
| **Version control screenshot** |
| Screenshot here |

## Sprint 5

### Planning A

|  |  |
| --- | --- |
| **Start date**  When did this sprint start? | **End date**  When will this sprint end? |
| Start date here | End date here |

### Components A

|  |
| --- |
| *Take a screenshot of your* ***entire*** *project board, showing what components you will work on. The components you plan to work on should be shown in the* ***In Progress*** *bucket.* |

|  |
| --- |
| **Project board screenshot** |
| Screenshot here |

|  |  |
| --- | --- |
| **Explain why you have selected these components for this sprint** | Explanation here |

### 

### 

### Trialling A/M

|  |
| --- |
| *In each sprint, at least three users should trial your outcome, either as it is developed or near the end of the sprint. As part of the standard, you need to trial* ***functionality*** *— the way the outcome works, how it achieves its goal, etc.*  *For Merit, you must trial multiple components over the lifetime of the outcome’s development. You can trial a given component more than once.*  *You should trial multiple approaches to a particular problem (*[*more details*](#q8pkvf8jfwdx)*). Decide on at least three approaches for a given problem, then gather feedback below.* |

|  |  |
| --- | --- |
| **Problem** | Explain the problem you are trying to solve |
| **Approach 1** | Explain, show, or link to your option here |
| **Approach 2** | Explain, show, or link to your option here |
| **Approach 3** | Explain, show, or link to your option here |

#### Feedback #1

|  |  |
| --- | --- |
| **Feedback giver name** | Name goes here |
| **Which approach is best** | Response goes here |
| **Explain why you think that approach is best** | Explanation goes here |
| **Suggest what you think could have improved the other approaches** | Suggestions go here |

#### Feedback #2

|  |  |
| --- | --- |
| **Feedback giver name** | Name goes here |
| **Which approach is best** | Response goes here |
| **Explain why you think that approach is best** | Explanation goes here |
| **Suggest what you think could have improved the other approaches** | Suggestions go here |

#### Feedback #3

|  |  |
| --- | --- |
| **Feedback giver name** | Name goes here |
| **Which approach is best** | Response goes here |
| **Explain why you think that approach is best** | Explanation goes here |
| **Suggest what you think could have improved the other approaches** | Suggestions go here |

### Testing

|  |
| --- |
| *You must show evidence of testing that your program works on a sample of expected (paetae/achieved), boundary (kaiaka/merit), and/or invalid (kairangi/excellence) cases. This testing should be done over the course of development, not just at the very end.* |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test name** | **E/B/I** | **Input** | **Expected output** | **Actual output** (screenshot) | **Comments** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

### Evaluation M/E

|  |
| --- |
| *Analyse the feedback that you received and make decisions about what you might change.*   1. *Explain if the responses were what you expected or not, and why.* 2. *Explain what changes you will make (such as additional features, different designs, etc.) in response to the feedback, and why.* 3. *Explain any proposals you will* ***not*** *make in response to the feedback, and why.* |

|  |  |
| --- | --- |
| **Explain if the responses were what you expected** | Explanation here |
| **✔︎ Explain changes you will make in response to feedback, and why** | Explanation here |
| **✘ Explain changes you will** not **make in response to feedback, and why** | Explanation here |
| **Explain how the changes have addressed or will address your outcome’s functionality** | Explanation here |

|  |
| --- |
| *Reflect on what happened during this sprint. Think about what components you were able to develop/trial easily, what was trickier, and what might not have been completed at all.*  *In your project board, components that you completed should move to the Completed bucket. If any were not completed, they should either remain in In Progress if you intend to keep working on them, or get moved back to To Do if you plan to work on something else during the next sprint.* |

|  |  |
| --- | --- |
| **✔︎ Explain what components you completed in this sprint** | Explanation here |
| **✘ Explain any components you were** not **able to complete this sprint, and why** | Explanation here |

|  |
| --- |
| *In your GitHub repository (not your Codespace), click on the commits symbol. Take a screenshot of the commits you made during this sprint. If you are working in a pair/group, take a screenshot of* ***all*** *commits, not just your own.* |

|  |
| --- |
| **Version control screenshot** |
| Screenshot here |

## Sprint 6

### Planning A

|  |  |
| --- | --- |
| **Start date**  When did this sprint start? | **End date**  When will this sprint end? |
| Start date here | End date here |

### Components A

|  |
| --- |
| *Take a screenshot of your* ***entire*** *project board, showing what components you will work on. The components you plan to work on should be shown in the* ***In Progress*** *bucket.* |

|  |
| --- |
| **Project board screenshot** |
| Screenshot here |

|  |  |
| --- | --- |
| **Explain why you have selected these components for this sprint** | Explanation here |

### 

### 

### Trialling A/M

|  |
| --- |
| *In each sprint, at least three users should trial your outcome, either as it is developed or near the end of the sprint. As part of the standard, you need to trial* ***functionality*** *— the way the outcome works, how it achieves its goal, etc.*  *For Merit, you must trial multiple components over the lifetime of the outcome’s development. You can trial a given component more than once.*  *You should trial multiple approaches to a particular problem (*[*more details*](#q8pkvf8jfwdx)*). Decide on at least three approaches for a given problem, then gather feedback below.* |

|  |  |
| --- | --- |
| **Problem** | Explain the problem you are trying to solve |
| **Approach 1** | Explain, show, or link to your option here |
| **Approach 2** | Explain, show, or link to your option here |
| **Approach 3** | Explain, show, or link to your option here |

#### Feedback #1

|  |  |
| --- | --- |
| **Feedback giver name** | Name goes here |
| **Which approach is best** | Response goes here |
| **Explain why you think that approach is best** | Explanation goes here |
| **Suggest what you think could have improved the other approaches** | Suggestions go here |

#### Feedback #2

|  |  |
| --- | --- |
| **Feedback giver name** | Name goes here |
| **Which approach is best** | Response goes here |
| **Explain why you think that approach is best** | Explanation goes here |
| **Suggest what you think could have improved the other approaches** | Suggestions go here |

#### Feedback #3

|  |  |
| --- | --- |
| **Feedback giver name** | Name goes here |
| **Which approach is best** | Response goes here |
| **Explain why you think that approach is best** | Explanation goes here |
| **Suggest what you think could have improved the other approaches** | Suggestions go here |

### Testing

|  |
| --- |
| *You must show evidence of testing that your program works on a sample of expected (paetae/achieved), boundary (kaiaka/merit), and/or invalid (kairangi/excellence) cases. This testing should be done over the course of development, not just at the very end.* |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test name** | **E/B/I** | **Input** | **Expected output** | **Actual output** (screenshot) | **Comments** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

### Evaluation M/E

|  |
| --- |
| *Analyse the feedback that you received and make decisions about what you might change.*   1. *Explain if the responses were what you expected or not, and why.* 2. *Explain what changes you will make (such as additional features, different designs, etc.) in response to the feedback, and why.* 3. *Explain any proposals you will* ***not*** *make in response to the feedback, and why.* |

|  |  |
| --- | --- |
| **Explain if the responses were what you expected** | Explanation here |
| **✔︎ Explain changes you will make in response to feedback, and why** | Explanation here |
| **✘ Explain changes you will** not **make in response to feedback, and why** | Explanation here |
| **Explain how the changes have addressed or will address your outcome’s functionality** | Explanation here |

|  |
| --- |
| *Reflect on what happened during this sprint. Think about what components you were able to develop/trial easily, what was trickier, and what might not have been completed at all.*  *In your project board, components that you completed should move to the Completed bucket. If any were not completed, they should either remain in In Progress if you intend to keep working on them, or get moved back to To Do if you plan to work on something else during the next sprint.* |

|  |  |
| --- | --- |
| **✔︎ Explain what components you completed in this sprint** | Explanation here |
| **✘ Explain any components you were** not **able to complete this sprint, and why** | Explanation here |

|  |
| --- |
| *In your GitHub repository (not your Codespace), click on the commits symbol. Take a screenshot of the commits you made during this sprint. If you are working in a pair/group, take a screenshot of* ***all*** *commits, not just your own.* |

|  |
| --- |
| **Version control screenshot** |
| Screenshot here |

# Project summary

## Relevant implications A

|  |
| --- |
| *Explain how you addressed the relevant implications that you identified at the start of the project. You should think about:*   1. *Whether you addressed them by doing what you originally planned or by doing something else* 2. *What evidence you have from testing/trialling to show the implications were addressed* |

|  |  |
| --- | --- |
| **Implication #1** | Explanation here |
| **Implication #2** | Explanation here |
| **Implication #3** | Explanation here |

## Evaluation E

|  |
| --- |
| *Explain how the process, tools, and techniques you used in each sprint helped shape the development of a high-quality outcome. Refer to evidence such as:*   * *Choice of components at decomposition, ones added/removed over the course of development* * *Changes made in response to trialling, such as improving functionality* * *Changes made in response to testing, such as confirming decisions* * *How the process, tools, and techniques helped your project get back on track when problems arose* |

|  |
| --- |
| **Reflection** |
| Explanation here |