

Project Proposal

04 February 2026

10:21

Francois Riches

2318399

Module: Changemakers - Creativity and Value Creation

Module Code: ACCA5033

Bachelor of Science: Computing (Software Engineering)

Link to this OneNote doc online:

[CM Assign1](#)

Task 1: Ideation & Innovation

04 February 2026 10:22

Innovation

Innovation Brings Change

Innovation often brings significant change, and in many cases these changes contribute to improvements in society. New tools, technologies, and medicines have helped address practical challenges and improve quality of life in many parts of the world. Many innovations initially reach only a small number of early adopters, but gradually spread more widely as production increases, costs fall, and access improves. However, it is also important to consider the incentives that shape innovation. The direction of innovation is frequently influenced by commercial interests, meaning that some developments prioritise control over products or data rather than broader social benefits.

Kondratieff Waves

Kondratieff waves describe long-term economic cycles driven by major technological developments. Each wave is associated with transformative innovations such as industrial manufacturing, electrification, or digital technologies, which reshape industries and economic growth [1]. These cycles are often measured through economic indicators such as investment patterns and stock market performance. While these measures can illustrate technological change within economies, they do not always reflect broader social outcomes. Economic expansion does not necessarily guarantee equal access to the benefits created by innovation.

Some commentators suggest that future waves may be driven by advances in healthcare and biotechnology. While these developments have the potential to improve quality of life, their impact will depend on how widely accessible such innovations become and whether they prioritise public wellbeing alongside economic growth proportionately [1].

Innovation Needs Creativity

Creativity plays a key role in the development of innovation. While innovation focuses on producing new tools, systems, or technologies, creativity is the process that generates the ideas behind these developments. In many cases, creativity involves finding new ways to combine existing technologies or applying them to solve different problems. Collaboration between individuals with different skills and perspectives can also encourage creative thinking, often leading to more effective solutions. In the field of computing, creativity can be seen in areas such as software design, algorithm development, and the creation of user-friendly interfaces. By encouraging experimentation and new approaches to problem solving, creativity helps drive the innovations that continue to shape modern technology and society.

Ideation

For ideas OpenAI's ChatGPT was asked for some project suggestions using 'Technology to help the homeless', 'Video game development', and 'Simulations utilising Markov chains' as inspiration. These were the results [2].

"1 Homelessness Support Resource Mapping Tool

Concept:

A simple web platform that maps nearby shelters, food banks, and support services, helping people quickly locate available assistance. [2]"

"2 Predictive Resource Allocation Using Markov Chains

Concept:

A simulation that models how individuals move between housing states (stable housing, temporary accommodation, homelessness) using Markov chains. The model could help visualise how interventions affect long-term outcomes [2]."

"3 Procedural World Generation Using Markov Chains (Game Dev)

Concept:

A game prototype where environments or narrative events are generated dynamically using Markov chain probabilities [2]."

In addition to the ChatGPT ideas and the idea I will be taking forwards.

Concept: A cozy farming game.

The game will be made in the Godot engine and will feature farming mechanics, the game loop will include planting, watering and tending to crops. In game currency will be used to purchase upgrades and newer more profitable crops.

Aim is,

To design and develop a playable prototype of a 2D farming game using the Godot engine that demonstrates core gameplay mechanics and progression systems.

Objectives are,

To create the game mechanics for basic game loops.

To create the UI and title menu to navigate the game.

To create a saving system for in game progress.

Add art to the functional mechanics to complete the game.

The method of doing the art last is to avoid expending too much time on a missing skillset or too much time browsing open-source art.

Research

1 The Godot game engine is a free open-source software that works very well for 2D game development [3].

2 Brackeys Youtube channel [4] has many tutorials for game development. Their video on GDScript [5] the programming language used in the Godot game engine will enable me to refresh on the syntax. The channel includes a range of other useful videos for this project.

3 Itch.io has a fantastic range of free asset packs for the art stage of the project [6].

Task 2: Project Management

04 February 2026 10:22

Project Scope

Project Solution:

The solution will be a playable prototype of a cozy farming game developed using the Godot game engine. The prototype will demonstrate the core gameplay loop, including planting crops, watering them, harvesting, and using in-game currency to purchase upgrades or additional crops. The project will focus on delivering functional gameplay systems first, followed by the addition of user interface elements and basic art assets. This approach ensures the game mechanics are fully operational before visual assets are integrated.

Technologies and Resources

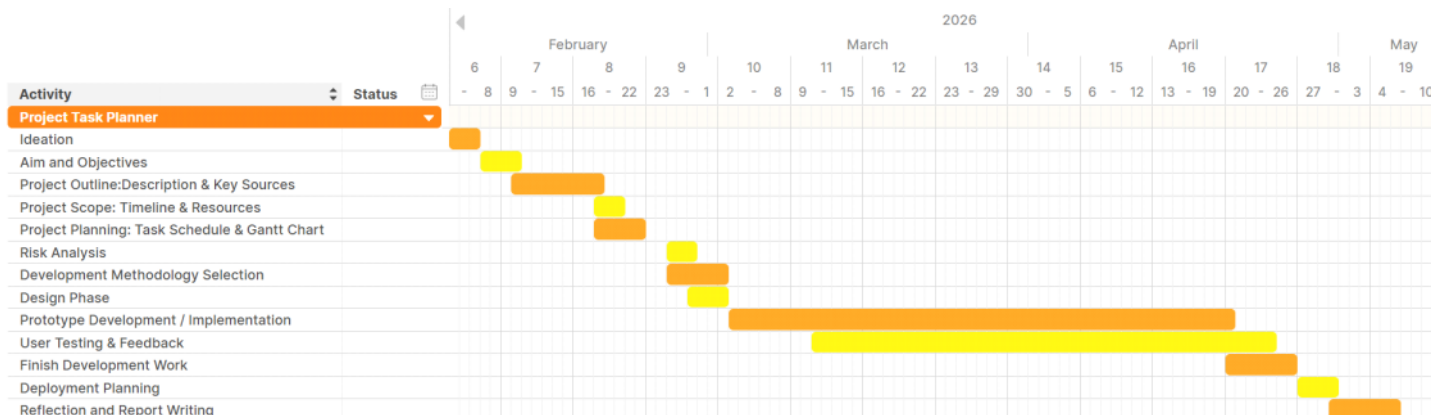
The project will use the Godot game engine for development and GDScript as the programming language. The code editor of choice will be Visual Studio. Learning resources will include tutorials from the Brackeys YouTube channel. Art assets will be sourced from open-source asset packs available on platforms such as Itch.io.

Task Schedule

TASK	START DATE & END DATE
Ideation	04/02/2026 – 06/02/2026
Aim and Objectives	07/02/2026 – 10/02/2026
Project Outline: Description & Key Sources	10/02/2026 – 18/02/2026
Project Scope: Timeline & Resources	18/02/2026 – 20/02/2026
Project Planning: Task Schedule & Gantt Chart	18/02/2026 – 22/02/2026
Risk Analysis	25/02/2026 – 27/02/2026
Development Methodology Selection	25/02/2026 – 02/03/2026
Design Phase	27/02/2026 – 02/03/2026
Prototype Development / Implementation	03/03/2026 – 20/04/2026
User Testing & Feedback	11/03/2026 – 24/04/2026
Finish Development Work	20/04/2026 – 26/04/2026
Deployment Planning	27/04/2026 – 30/04/2026
Reflection and Report Writing	30/04/2026 – 06/05/2026

Gantt Chart

Figure 1: Gantt chart created using tomsplanner.com [7]



Risks and Challenges

#	Risk	Impact	Likelihood	Contingency
1	Hardware Failure & Loss of Data	High	Low	Keeping a backup of files to ensure that no work is lost. Making use of the resources in the university, such as the available computers, in the event of home hardware failing.
2	Skillset	High	Medium	Make sure to keep the scope of the project just outside the boundaries of current skillset, enabling the development of new skills via learning resources.
3	Time Management	Medium	Medium	Use tools for project management, such as the Gantt chart to properly assign time to tasks as required.
4	Scope Creep	Medium	High	Make use of MoSCoW when designing possible features to issue each feature a priority. Once prioritised apply Minimum Viable Product (MVP) principle to define the scope of the project.
5	Developer Health	High	High	Make attempt to comply to the schedule even in the event of poor mental health. Be conscious of allowing the work to stack too much.

Task 3: Project Development Method & Design Work

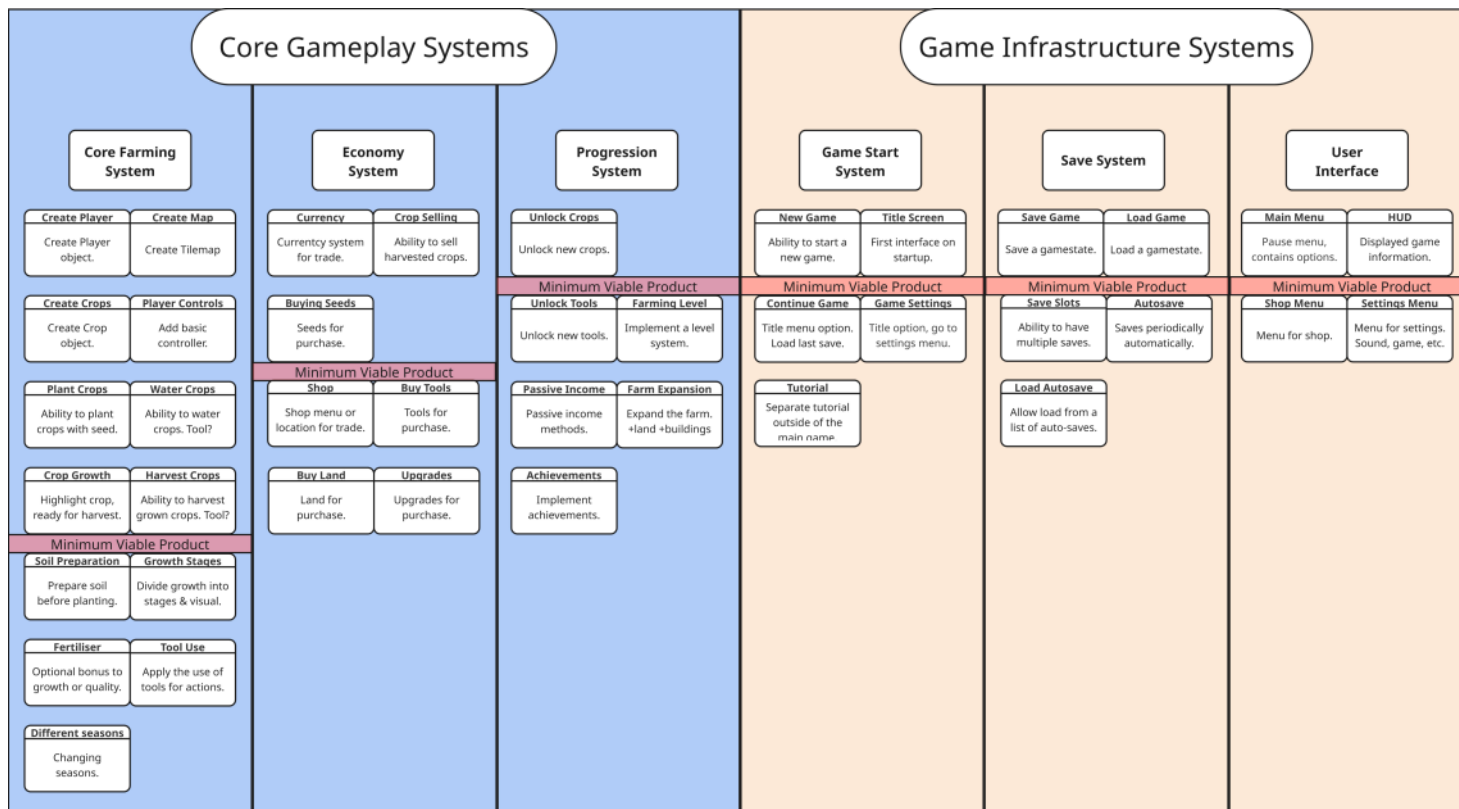
04 February 2026 10:22

Development Method

The project is organised using a **feature-driven task board** (see figure 2) that follows **Agile development principles**. Features are listed and prioritised to guide the order of development and to allow progress to be tracked visually. To determine the importance of each feature, the **MoSCoW prioritisation method** has been used. Features categorised as *Must have* are placed at the top of the board, while *Should have* and *Could have* features are reserved for future iterations. *Won't have* features are excluded from development. This approach helps ensure that the most important functionality is implemented first.

A **Minimum Viable Product (MVP)** line has also been drawn on the board to identify the features required for the first development cycle. These MVP features represent the minimum functionality needed for the system to operate effectively. The features identified for the MVP have been transferred into a **requirements table**, where they are tracked throughout development. As work progresses, each requirement will be marked as completed once the feature has been successfully implemented.

Figure 2: Feature Driven Development board created using Miro [8]



Design Work

MVP Requirement Specifications

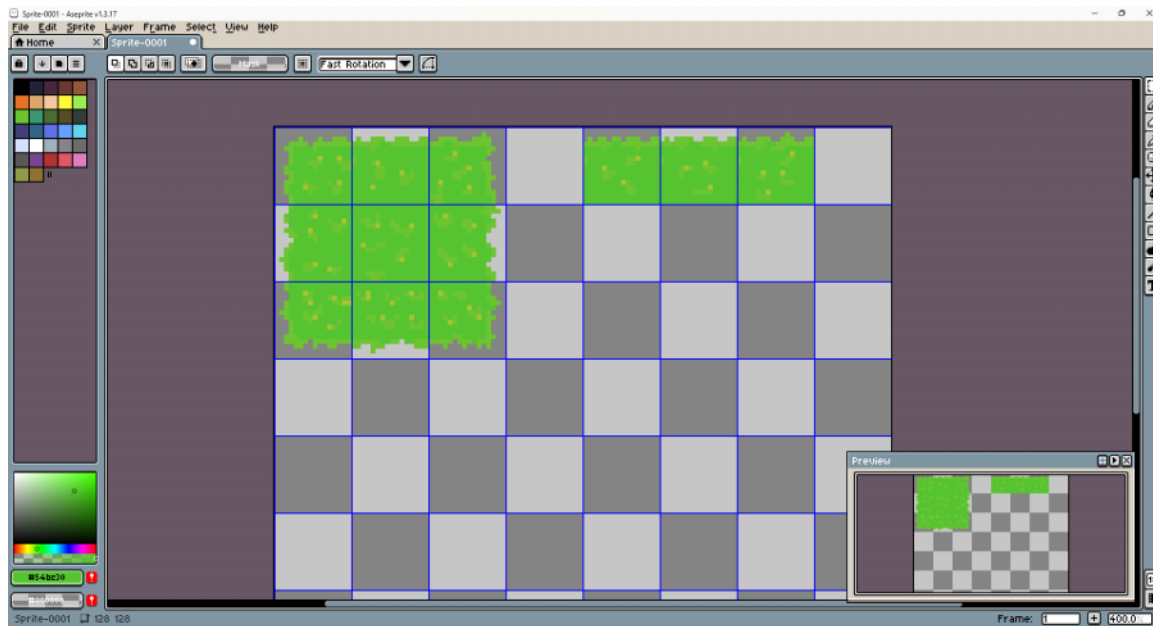
ID	Priority	Name	Description/Comments	Status
1	High	Create Player	Create the player object. Add variables such as position vector.	
2	High	Player Controller	Create a player controller to move the player and utilise a use button.	
3	High	Create Map	Create a tilemap as the ground for the game.	Started
4	High	Create Crops	Create a crop object for use with varied crops to be used in the game.	
5	High	Plant Crops	Define preset plots to allow the planting of crops, this can be later modified.	
6	High	New Game	Allow the player the ability to start a new game from the title menu.	
7	Medium	Title Screen	Create the title screen which will be loaded on startup.	
8	Medium	Save Game	Allow the player the ability to save their in game progress.	
9	Medium	Load Game	Allow the player the ability to load a previously saved game.	
10	Medium	Water Crops	Allow the ability for player to water planted crops. Include visual indicator.	
11	Medium	Crop Growth	A visual indicator to show that the crop has grown and is harvestable.	
12	Medium	Harvest Crops	Allow the ability for the player to harvest the crops. This should reset plot.	
13	Low	Main Menu	A pause screen with a menu of options which will include closing the game, saving,	

			and in time an options menu.	
14	Low	Currency	Integrate a currency system to hold integer values for numerical operations used for trade.	
15	Low	Crop Selling	Allow the player to sell crops for currency.	
16	Low	Buying Seeds	Allow the player to buy a selection of seeds for currency.	
17	Low	Unlock Crops	As part of the progression system allow the player to unlock new seeds.	
18	Low	HUD	Create all the on screen visual indicators for player information, such as currency.	

Creating Tilemap Art

The Aseprite pixel art editor is being used to begin creating the tilemap artwork for the farming game. A 16×16 pixel grid is used to design a grass overlay, which will sit on top of a 16×16 dirt tilemap. This layered approach allows the environment tiles to be combined while maintaining a consistent visual style. This set will suffice for testing but will require more edges for better map design.

Figure 3: Creating the tilemap art in Aseprite [9]



Testing Tilesets in Godot

Testing of layered tilemaps has been carried out within the Godot engine. This approach produces a strong visual result and may also support future mechanics such as terrain manipulation in later iterations of development. The dirt layer is currently simple, consisting of either standard dirt or tilled soil to represent planting plots. However, this could be expanded later by adding edge tiles to improve the visual transitions between terrain types. The grass layer will require a complete set of edge tiles to allow for more varied and visually appealing map design.

Tutorial videos are also being used to develop familiarity with the Godot game engine and its node-based system. The engine provides a wide range of features and tools, which gives confidence that it will support the required gameplay mechanics as development progresses.

This experimentation forms part of an iterative development process, where small features are tested and refined before being expanded in later stages of the project.

Figure 4: Creating the DirtMap tileset in Godot [3]

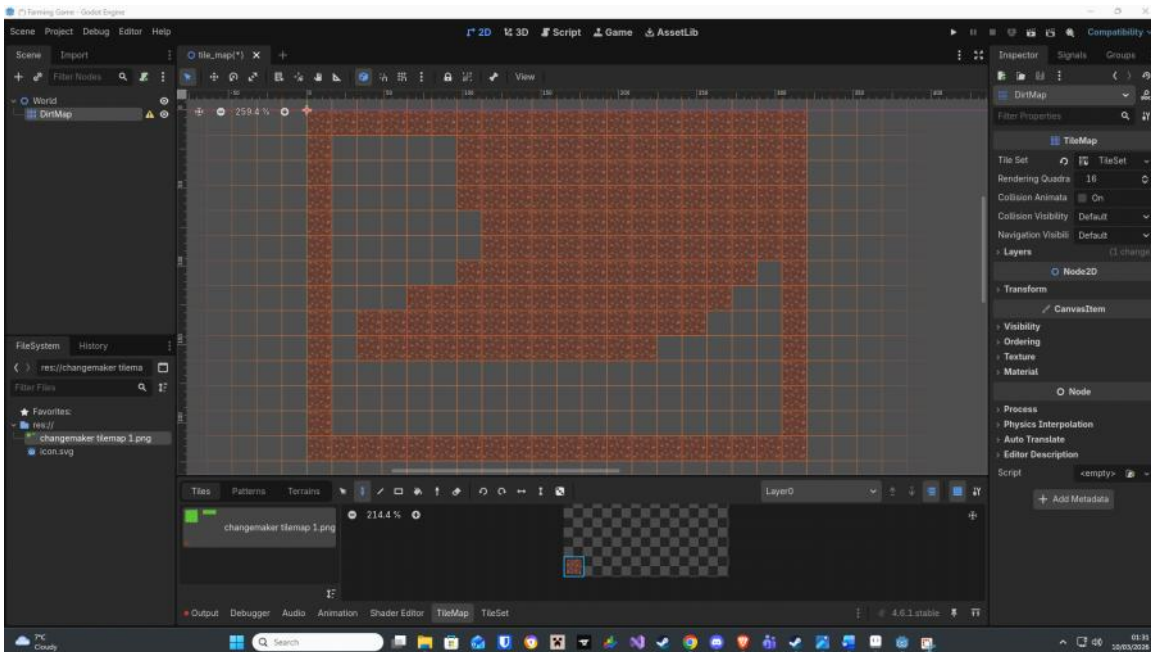


Figure 5: Creating the GrassMap tileset in Godot [3]

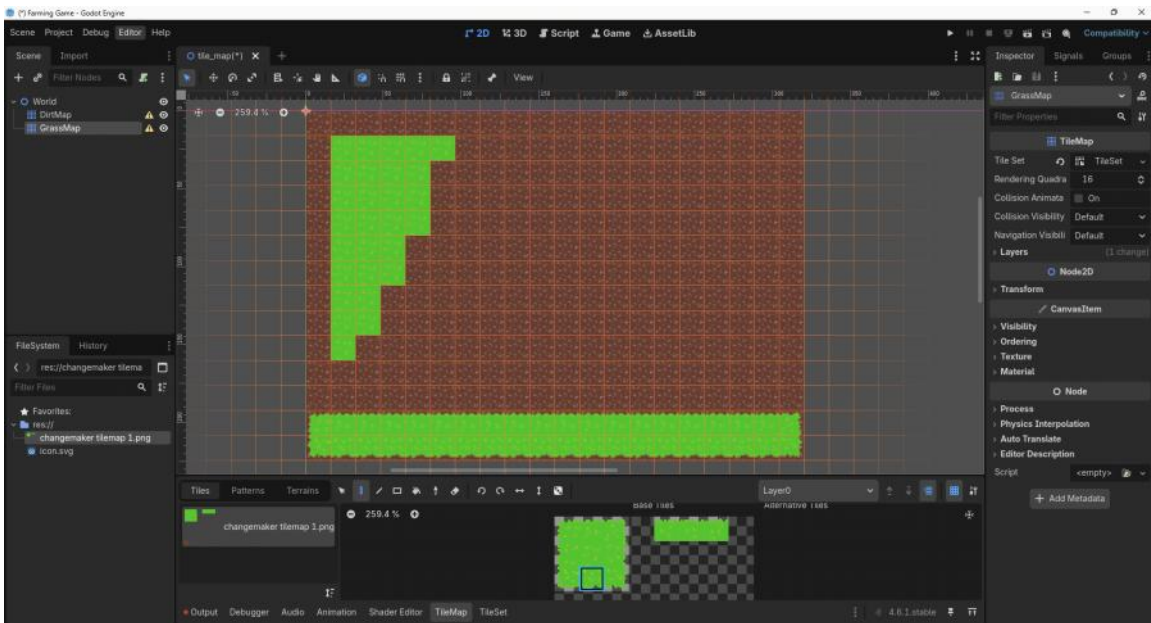
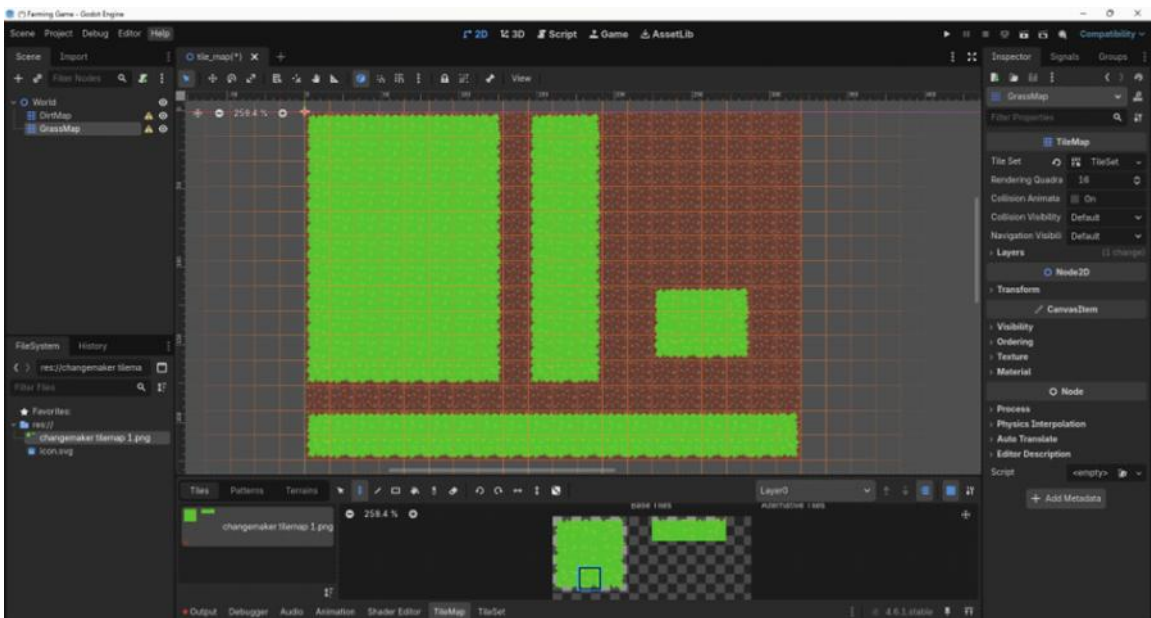


Figure 6: Finish testing the map layering process in Godot [3]



References

04 February 2026 10:22

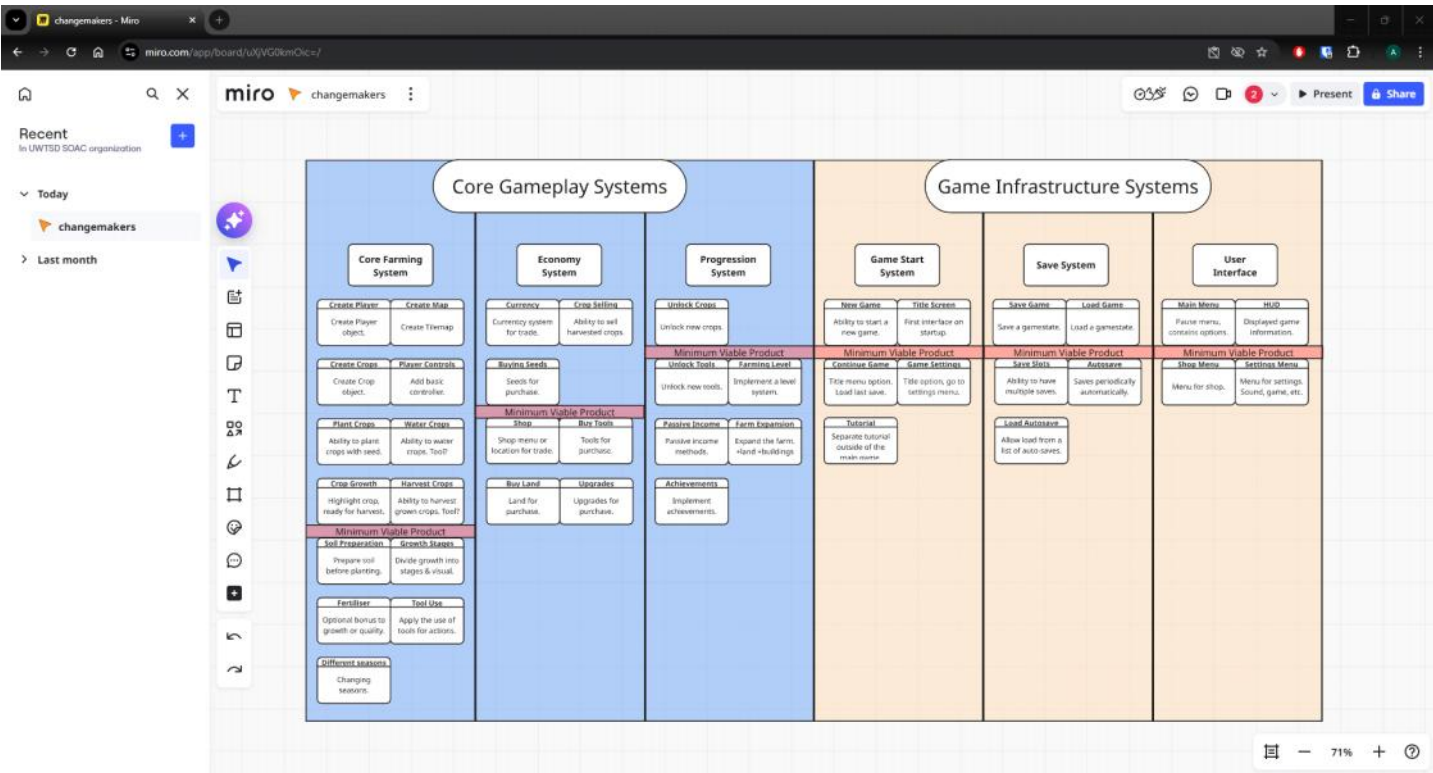
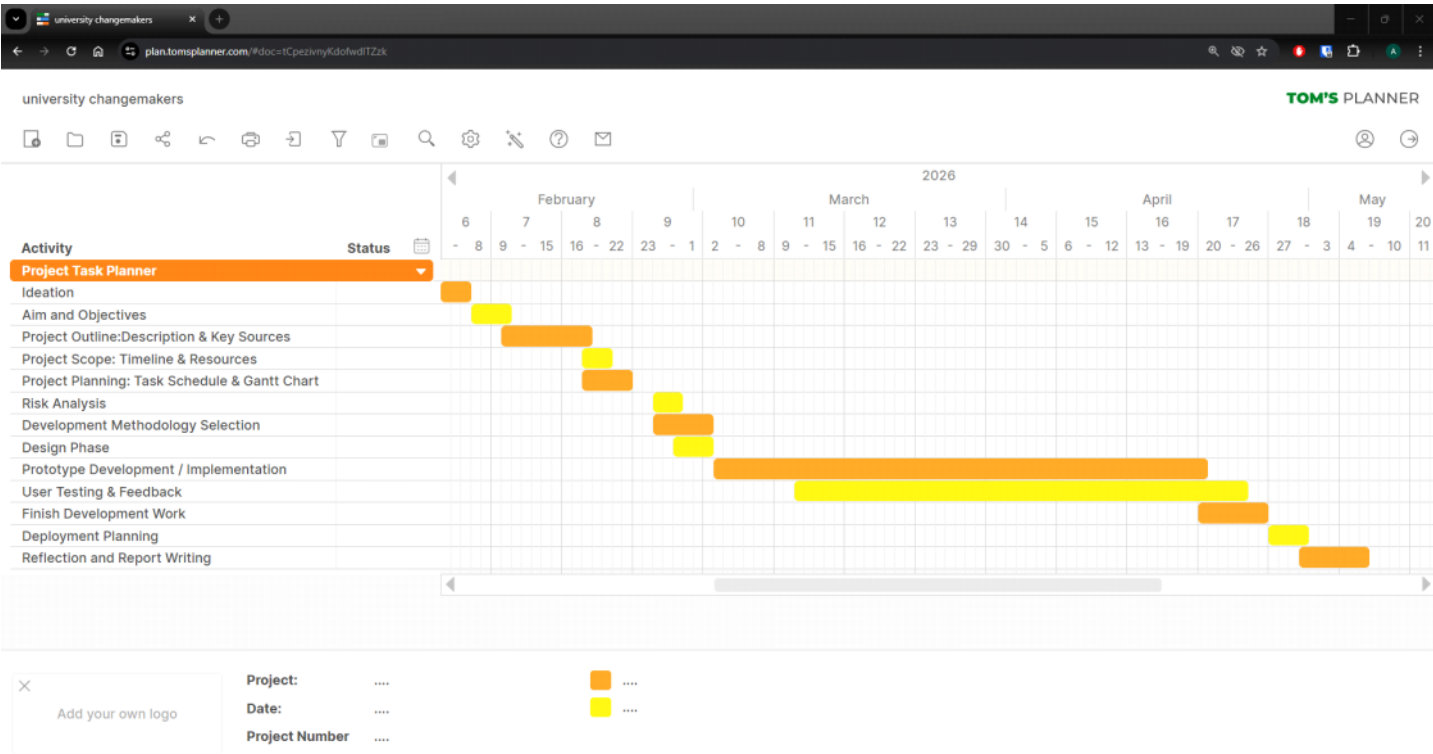
References

[1]	A. Korotayev, J. Zinkina and J. Bogevolnov, "Kondratieff waves in global invention activity (1900–2008)," <i>Technological Forecasting and Social Change</i> , vol. 78, no. 7, pp. 1280-1284, September 2011.
[2]	OpenAI's ChatGPT, <i>Private Communication</i> , 2026.
[3]	Godot Foundation, "Your free, open-source game engine," [Online]. Available: https://godotengine.org/ . [Accessed 6 March 2026].
[4]	Brackeys, "Brackeys," Brackeys, [Online]. Available: https://www.youtube.com/@Brackeys . [Accessed 6 March 2026].
[5]	Brackeys, "How to program in Godot, DGScript tutorial," Brackeys, [Online]. Available: https://www.youtube.com/watch?v=e1zJS31tr88 . [Accessed 6 March 2026].
[6]	Itch.io, "Itch.io Assets," [Online]. Available: https://itch.io/game-assets . [Accessed 6 March 2026].
[7]	Toms Planner, "tomsplanner," [Online]. Available: https://plan.tomsplanner.com/ .

	[Accessed 6 March 2026].
[8]	Miro, "Miro," [Online]. Available: https://miro.com/ . [Accessed 6 March 2026].
[9]	Aseprite, "Aseprite product page," [Online]. Available: https://www.aseprite.org/ . [Accessed 8 March 2026].

Use of Project Management Tools

Tom’s Planner was used to create the project Gantt chart, providing a visual timeline of development stages and key tasks. Miro was used to create a feature-driven design (FDD) board, allowing project features and systems to be organised and prioritised for development.



Continuing Professional Development

As part of continued professional development (CPD), online tutorial resources were used to support the development of relevant technical skills. Tutorial videos were followed to learn the process of applying tilemaps within the Godot game engine and to understand how to create tile sets using the Aseprite pixel art editor. These resources helped develop practical knowledge of the tools required for the project and supported the early implementation of the game environment.

