

The Development of PC-Based 2D Farming Simulation Role-Playing Game: Tranquil Harvest

Lee Bing Yang¹, Che Samihah Che Dalim^{1*}

¹ *Fakulti Sains Komputer dan Teknologi Maklumat,*

Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, 86400, MALAYSIA

*Corresponding Author: samihah@uthm.edu.my

DOI: <https://doi.org/10.30880/aitcs.2024.05.02.043>

Article Info

Received: 17 July 2024

Accepted: 16 October 2024

Available online: 15 December 2024

Keywords

PC-Based Game, Farming Simulation, Role-Playing Game, Local Vegetable, Local Plant

Abstract

Agriculture, essential for human civilization, faces disconnection due to urbanization. To bridge this gap, "Tranquil Harvest," a 2D farming simulation RPG for PC, educates players about the variety of vegetables commonly grown in Malaysia. The game offers a virtual farming experience, allowing players to engage in realistic agricultural activities, enhancing their awareness of food origins. Developed using the Game Development Life Cycle (GDLC), the game targets users aged 12 to 35. Alpha testing ensured functionality, while beta testing with 30 participants provided positive feedback on learning outcomes, user acceptance, and functionality. Players found the game educational and enjoyable, demonstrating its effectiveness in bridging the urban-agriculture knowledge gap.

1. Introduction

Tranquil Harvest is a comprehensive initiative aiming to develop a PC-Based 2D farming simulation role-playing game that uniquely merges entertainment and education. This endeavor is driven by the recognition of the growing global player base for PC games, with projections suggesting 1.3 billion players by the end of the year 2023. The two-dimensional visual style is a deliberate choice, not only for its unique aesthetic appeal but also for its compatibility with a broader range of devices, enhancing accessibility in addition to entertainment. As a simulation game, it immerses players in virtual farming activities, fostering cognitive abilities such as deep learning, critical thinking, and decision-making. The educational content extends to local plant knowledge, Malaysian flora, and cultural significance, making it a valuable tool for knowledge and awareness. The significance of this project lies in its potential to engage a broad audience, particularly young players, in learning about agriculture and sustainable farming practices, biodiversity, and the importance of protecting local plant species.

The review on existing apps and literature identifies critical issues in the current 2D farming simulation genre, highlighting the prevalent lack of educational content and the genre's limited distinctiveness. "Tranquil Harvest" seeks to address these challenges by infusing educational elements related to local plant knowledge and creating a unique and diverse gaming environment. The objectives of the project include designing and developing the game using the Unity game engine, with alpha and beta testing to refine the user experience. The project scope outlines the various features of "Tranquil Harvest," encompassing farming activities, a weather system, a day night shifting system, a time system and a shop system. The target audience ranges from children to adults, with a focus on users aged 12 to 35. The game integrates educational elements related to local plant knowledge, cultural significance, and agricultural practices. The expected result from this ambitious project is the creation of "Tranquil Harvest," a 2D farming simulation role-playing game offering an innovative and educational gaming experience. The game is anticipated to engage a broad audience and promote sustainability, biodiversity, and a deeper understanding of local plant knowledge and basic farming knowledge. This aligns with the project's overall goal

of providing a serene and enjoyable gaming experience while enriching players' knowledge and appreciation of nature and agriculture. In conclusion, "Tranquil Harvest" represents a significant contribution to the gaming industry by elevating the 2D farming simulation genre through the infusion of educational content. It seeks to set itself apart by offering a distinctive and enriching gaming experience, contributing to the advancement of educational gaming and fostering awareness of vital environmental and cultural aspects.

A literature review is presented in the next chapter, which delves into the project's study domain, the technology employed, and conducts comparative analyses of existing games. In chapter 3, the Game Development Life Cycle (GDLC) methodology selected for the project is explained while chapter 4 delves into the entire process of designing and developing game,

2. Related Work

2.1 Agriculture

Agriculture is important and it has a profound impact on the human society, leading to larger populations, complex civilizations, and technological innovations. Agriculture sheds light on its role in providing food security, raw materials, economic development, and job creation. Citing statistics from 2022 [1], the narrative underscores the contemporary challenges of chronic hunger and the need for urgent food assistance. Furthermore, it emphasizes agriculture's contribution to the global economy by providing essential raw materials for various products, showcasing corn as an example. The economic significance of agriculture is illustrated through the Annual Economic Statistics of Malaysia's agriculture sector, revealing growth in grow output value [2]. The Malaysian palm oil industry serves as a case study, highlighting its global impact, productivity enhancements, and job creation. Additionally, the role of agriculture in job creation in Malaysia is discussed, with over 1.5 million people employed in the sector [3]. The knowledge about agriculture is explored through personal experience and education. Personal experience, such as growing up on a farm or working on one, provide valuable firsthand knowledge of agricultural practices [4]. Education plays a crucial role in disseminating knowledge about agriculture, with schools and universities offering courses and degrees in various agricultural disciplines. The modernization of agricultural education through online courses, tutorials, and virtual farms also provides interactive and innovative learning opportunities [5].

2.2 Game Technology

In game technology, games are structured forms of play that provide entertainment, requiring skill, strategy, and luck. They encompass elements such as rules, goals, and challenges. Rules dictate gameplay, often encouraging less efficient methods, and goals provide quantifiable objectives for players, fostering a sense of achievement. Challenges, linked to complexity, vary based on player proficiency. Feedback mechanisms, including in-game hints and progress indicators, aid players in strategy and improvement [6].

Personal computers (PCs) are a dominant platform for gaming, offering a wide variety of titles, genres, and capabilities. PC games benefit from the processing power and graphical fidelity of modern PCs, enabling experiences that may not be possible on consoles. Role-Playing Games (RPGs) involve participants assuming character identities in fictional worlds, commonly featuring narratives, quests, levels, statistics, combat systems, equipment, interactive elements, and character classes. Simulation games, tracing back to "The Sumerian Game" in 1964, simulate real-world scenarios for entertainment and training, evolving to include fantastical elements. They are used across various industries, serving as crucial tools for skill development and training in fields such as military, healthcare, and education. The advent of virtual reality has enhanced the realism of simulation games. Farming simulation games, exemplified by "Harvest Moon," immerse players in the role of a farmer, providing educational insights into agriculture, economics, and sustainability. These games contribute to the development of soft skills like critical thinking and empathy. Lastly, 2D model technology involves games that use two-dimensional graphics, represented as sprites and tiles. Sprites, stored in sprite sheets, can be animated and created using pixel art, vector graphics, or hand-drawn art. Tiles, arranged in a grid and stored in tilesets, create backgrounds and elements in a game, also made using pixel art, vector graphics, or hand-drawn art.

2.3 Comparative Analysis

In this section, a comparison has been made between the existing game, such as Stardew Valley, Animal Crossing: New Horizon, and My Time at Sandrock and the proposed game. Table 1 shows the result of the comparative analysis. Based on Table 1, several advantages and limitations of the proposed application can be concluded. Firstly, 13 types of local vegetables and plants will be added to the game to introduce the different kinds of local vegetables in Malaysia.

Table 1 Comparative Analysis

Feature	Stardew Valley	Animal Crossing: New Horizon	My Time at Sandrock	Tranquil Harvest
Platform	PC, PlayStation 4, Xbox One, Nintendo Switch, iOS, Android	Nintendo Switch	PC, PlayStation 4, Xbox One, Nintendo Switch	PC
Pricing	RM 31.00	RM 195.00	RM 109.00	FREE
Ads	No	No	No	No
Design	2D pixel art	3D with cel-shaded graphics	3D with voxel-based graphics	2D pixel art
Advantages	Relaxing and addictive gameplay, charming characters, open-ended gameplay, high replayability	Charming characters, relaxing and addictive gameplay, seasonal events, creative freedom	Addictive gameplay, deep crafting system, focus on community and relationships	Relaxing gameplay with simple farming system that focus on teaching players about different types of local plants and vegetables.
Limitations	Can be repetitive after a while, focus on farming but no local plants or vegetables are introduced in-game.	Can be slow-paced, not as much focus on farming or resource management. Not many activities are based on farming.	Can be grindy, Simple farming mechanism but did not introduce any local plants or vegetables.	Can be repetitive after a while, limited activities for the players as farming and introducing about local plants and vegetable is the main goal.

3. Methodology

The proposed game, Tranquil Harvest is a PC-Based designed as a tool to introduce and learn about local plants and vegetables. The Game Development Life Cycle (GDLC) approach is used to guide the design and development of this game [7]. The 6 stages of the GDLC methodology are shown in Figure 1. The stages are the initiation phase, pre-production phase, production phase, testing phase, beta phase, and release phase. All the stages of the GDLC methodology will be explained in the next sub-section.

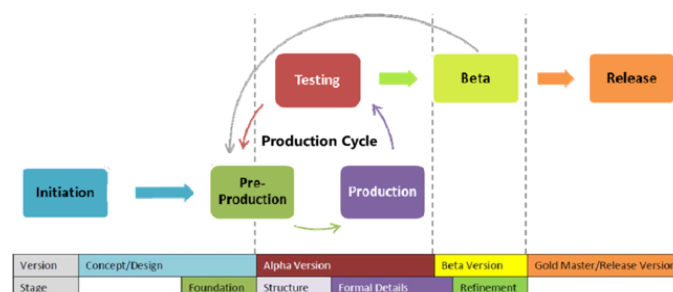


Fig. 1 GDLC Methodology [7]

3.1 Initiation Phase

In the initiation phase, the rough concept of the game is developed, defining the problem statement, objectives, and project scope. The target user demographic is identified as individuals between the ages of 12 to 35 [8].

3.2 Pre-production Phase

The pre-production phase focuses on design, involving the creation and revision of prototypes, interface designs, user analysis, navigation structure, and flowcharts. During this phase, essential questions regarding the game are answered, such as the genre, 2D or 3D, art style, gameplay mechanics, characters, and the game engine that will be used. A proof of concept is developed to determine the resources required to bring the game to life.

3.3 Production Phase

The production phase involves the development and design of the graphic assets, animation and the source codes of each main features of "Tranquil Harvest" in each scene. The game assets that required to develop "Tranquil Harvest" are the sprites and animation using Aseprite. The audio assets of the game are obtained from license-free online libraries. After designing and preparing all the required game assets, "Tranquil Harvest" is developed using Unity Game Engine with C# programming language. A functional prototype is produced in this stage.

3.4 Testing Phase

The testing phase includes alpha testing by developers or internal testers to identify errors and bugs, with immediate improvements made. After development, the game is rigorously tested to find and solve issues, improve gameplay, and verify that it works smoothly across all intended platforms.

3.5 Beta Phase

Beta testing is an important stage in the software development process that involves testing a product by real-world consumers in a controlled environment. In this project, beta testing involves external target user aged 12 to 35 to discover remaining bugs and gather feedback to ensure that all the components of the game are working as planned without any error.

3.6 Release Phase

The release phase, the final stage, involves releasing a stable, error-free version of the game to the market. The project planning outlines a timeline from October 11, 2023, to June 5, 2024, with specific tasks [8].

4. Result and Discussion

This section discusses the implementation and outcome derived from each phase of the development of "Tranquil Harvest," based on the GDLC methodology.

4.1 Implementation of Pre-Production Phase

In this section, both the hardware and software requirements will be discussed as both are important for the development of "Tranquil Harvest". The user analysis has been conducted with the subject matter expert, table 2 shows the user analysis. Navigation structure and flowchart are shown in Appendix A and Appendix B. Functional and non-functional requirements are listed in Table 3 and Table 4. Table 5 will show the button design and Figure 2(a) to Figure 2(f) will show the interface design for all the game scene such as the main menu, tutorial, in-house, farm, town and in-shop scene.

Table 2 *User analysis*

Stakeholder Category	Role in product	Design Implications	Action Needed
Subject matter expert	Provide knowledge and guidance in agriculture field, end-user of the system	Based on the interview,	Develop game with offline mode.
		<ul style="list-style-type: none"> Game should be able to play at any time, most importantly with free excess. 	<ul style="list-style-type: none"> Develop game that is free on the PC platform.
		<ul style="list-style-type: none"> Prefer English version. 	<ul style="list-style-type: none"> Develop a game in English language.

Table 2 *User analysis (cont)*

- Easy to understand and play.
 - Provide clear instruction and guidance.
 - Develop with a simple game style.
-
- Game with valuable contents
 - Provide knowledge about local plants and vegetable.
 - Provide description about the plants and vegetable in the game.






Table 3 *Functional Requirement*

No	Functional Requirement
1	The PC game should allow the user to swap between tools.
2	The PC game should allow the user to navigate through different scenes.
3	The PC game should allow the user to continue their last save.
4	The PC game should allow the user to start a new game.
5	The PC game should allow the user to access the tutorial.

Table 4 *Non-Functional Requirement*

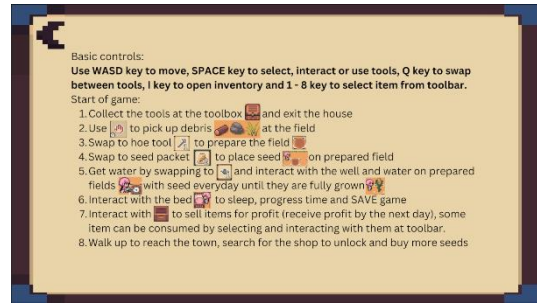
No	Non-Functional Requirement	Description
1	Implementation	The PC game should be able to access any Windows platform.
2	Usability	The PC game should be developed in the English language. The buttons should be designed in contrasting colors with thick border.
3	Reliability	The PC game should be accessible by the user at any time.
4	Performance	The response time of the PC game should be within one second.
5	Security	The PC game should allow the user to view the information in the PC game only, and not modify the data.

Table 5 *Button Design*

Button	Function Description
	<ul style="list-style-type: none"> • This is the tutorial button displayed in main menu interface.
	<ul style="list-style-type: none"> • This is the start new farm button displayed in the main menu interface.
	<ul style="list-style-type: none"> • This is the load last save button displayed in the main menu interface.
	<ul style="list-style-type: none"> • This is the back button displayed in tutorial interface.
	<ul style="list-style-type: none"> • This is the swap tools button displayed in the in-house interface, farm interface, town interface and in-shop interface.



(a)



(b)



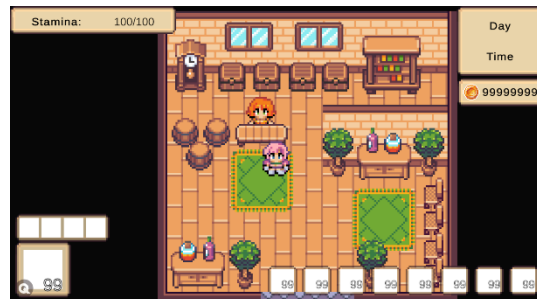
(c)



(d)



(e)



(f)

Fig. 2 Interface design for main menu scene (a), Interface design for tutorial scene (b), Interface design for in-house scene (c), Interface design for farm scene (d) Interface design for town scene (e), Interface design for in-shop scene (f)

4.2 Implementation of Production Phase

The interface design from pre-production phase depicted key scenes, such as the in-house scene, farm scene, town scene, and in-shop scene, illustrating player activities like planting seeds, watering plants, harvesting crops, and selling crops. This visual guide helped maintain a clear vision throughout the production phase.

4.2.1 Development of Graphic Assets

The graphic assets needed to develop “Tranquil Harvest” is designed using Aseprite, these graphic assets include the main character, the non-player character (NPC), the plants, crops and seeds, the environment graphic assets such as the trees, grass land, stone, houses, stone path, water well and furniture. The developed graphic assets will be shown in Figure 3, Figure 4 and Figure 5 below.

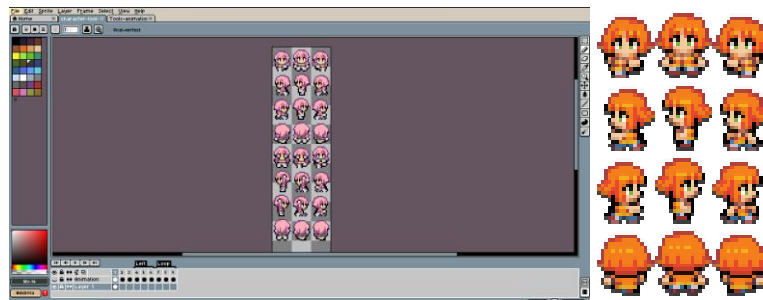


Fig. 3 Development of Main Character and Non-player Character Graphic Assets



Fig. 4 Development of Plants, Crops and Seeds Graphic Assets



Fig. 5 Development of Environment Graphic Assets

4.2.2 Development of Animation

The animation needed to develop “Tranquil Harvest” is designed using Aseprite, these animations include idle, walking, pick up, swinging and watering animation for the main character. After that in Unity Game Engine, a few blend trees in the animator tools are used to sort out the logic behind the playing of each animation, this is to ensure that all the available animations are played at the right moment and give the correct feedback to the player. Figure 6 shows the development of all the animations for the main character and Figure 7 shows the blend trees in animator tools and all the animations added in the animation tools in Unity Game Engine.



Fig. 6 Development of Main Character Animation

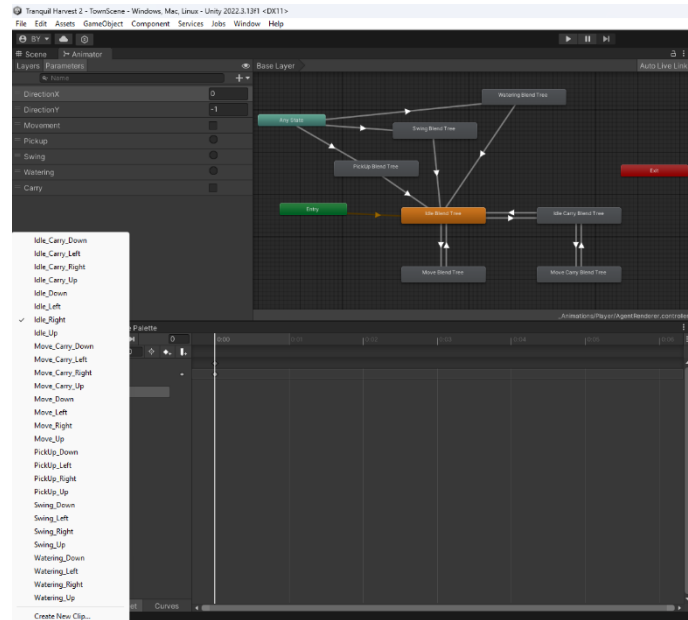


Fig. 7 Blend Trees, Animation and Animator Tools in Unity Game Engine

4.2.3 Code Implementation in Unity Game Engine

This section discussed about the codes that are implemented in each of the game scene. The programming language used is C#. Each game scene will be shown in this section following with the features in each game scene and the explanation of the code working behind these scenes to make these features work.

4.2.4 Code Implementation for Main Menu Scene

Figure 8 shows the main menu scene of the game, this will be the starting scene of the game. There are their buttons in the scene which is the button for the tutorial, the button to start a new game and the button to continue last game progress. The code implemented manages the main menu functionality, which include starting a new game, loading a saved game, and handling scene transitions. It utilizes various components and managers like `ScreenTransitionEffect`, `SceneTransitionManager`, and `SaveManager`. The script has serialized fields that can be assigned in the Unity editor, such as the name of the "in-house" scene and references to required components. The `Start` method checks for saved game data and triggers an event if none is found. The `CreateNewGame` method resets saved data, plays a transition effect, and loads the "in-house" scene after a delay. The `LoadSavedData` method saves the current game state, plays a transition effect, and loads the previously loaded scene. The `LoadScene` method is a coroutine that loads a specified scene after a delay, using the `SceneTransitionManager`.

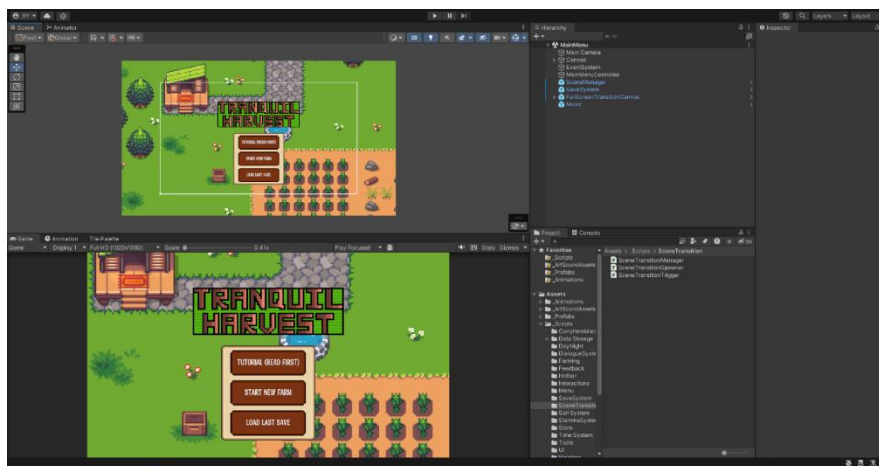


Fig. 8 Main Menu Scene

4.2.5 Code Implementation for In-house Scene

Figure 9 shows the in-house scene of the game, which is the second scene after the player starts a new game or loads a saved progress. This scene contains a bed that the player can interact with to progress the game time to the next day and save their progress. The scene also has a toolbox where the player can store tools like seeds, watering cans, and hoes. The `SaveManager` script is responsible for saving and loading game data. It uses `PlayerPrefs` and file saving, serializes/deserializes data from objects implementing `ISavable` interface, tracks unused data, can reset saved data, and has a flag to load from file or `PlayerPrefs`. Data is structured as a list of `ObjectsSaveData` with ID and serialized object data. JSON is used for serialization. The `ToolBox` script manages the toolbox inventory system. It has a serialized toolbox inventory, list of initial tools, and reference to an item database. `GetData` returns serialized toolbox data for saving. `RestoreData` restores toolbox from saved data or adds initial tools if no data. It uses an `InitialToolData` struct for tool ID and amount. The `SceneTransition` script is responsible for the scene transition from in-house to farm scene and vice versa. It has fields for target scene name, spawn point, trigger flag, and ID. It finds `SceneTransitionManager`, blocks player movement on trigger enter, plays a screen transition effect, and loads the target scene via `SceneTransitionManager` when effect finishes. It works with `Player` and `SceneTransitionManager` components.

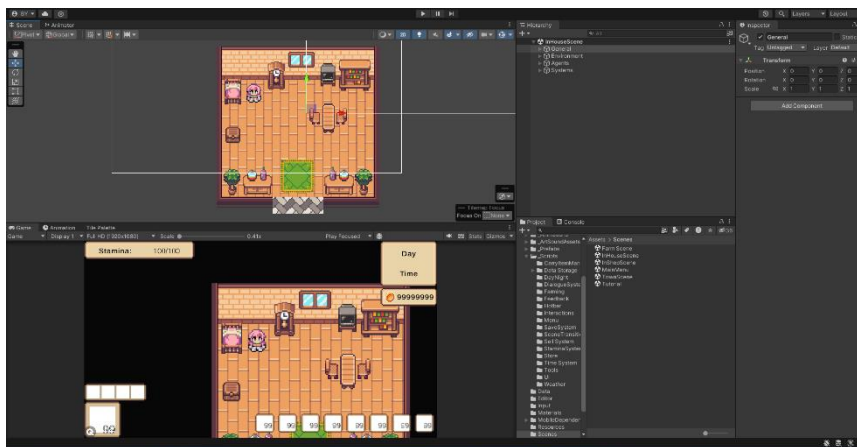


Fig. 9 In-house Scene

4.2.6 Code Implementation for Farm Scene

Figure 10 shows the farm scene of the game which is the third scene after the player exits the in-house scene. This scene contains a sell box for the player to sell items, a water well for the player to refill their watering can, a field where the player can plant seeds and the same scene transition script from the in-house scene is reused here, allowing the player to transition between the farm scene and the town scene. The `SellManager` script manages the selling system where items can be sold at a specific hour. It has fields for the sell hour, item database, player data, and a sell inventory. It subscribes to time events to trigger selling at the set hour and reset on a new day. `PerformSelling` method sells all items in the inventory, adds proceeds to the player's money, and clears the inventory. `GetData` and `RestoreData` method implement `ISavable` for saving/restoring state. The `FieldRenderer` script renders and manages the visual farm field. It uses a tilemap for prepared and watered tiles and creates, updates, removes crop and debris visualizations. It has methods for creating crops, wilting, making collectable, adding/removing debris, clearing fields, removing crops, watering crops with particle effects, and resetting watered fields. It works with farming/crop management systems. The `FillInWaterInteraction` script manage the logic that allows the player to refill their watering can tool. It has a `UsableTools` field specifying which tools can be used which is in default is just the watering can. `CanInteract` method checks if the agent's current tool is usable. `Interact` restores the agent's current tool and invokes `OnInteract`. It works with the `WateringCanTool` and `ToolsBag` script.

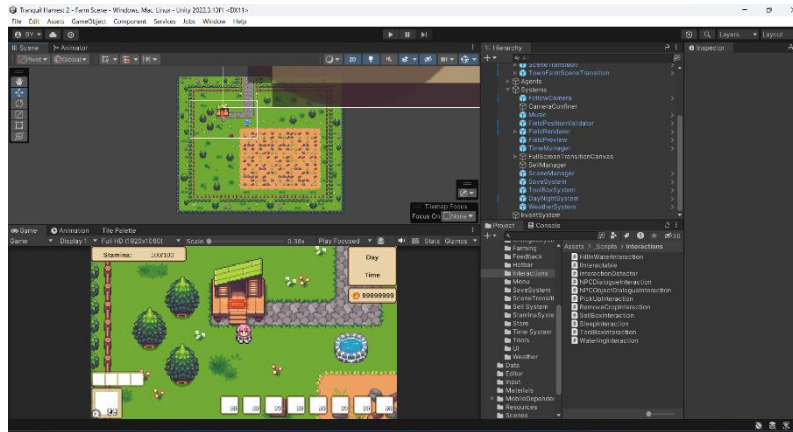


Fig. 10 Farm Scene

4.2.7 Code Implementation for Town Scene

Figure 11 shows the town scene of the game which is the fourth scene of the game. The player can transition from the farm scene to this scene by following the stone path at the farm scene. Only two scene transition script will be implemented in this scene which is the scene transition of the farm scene with the town scene and the town scene with the in-shop scene.

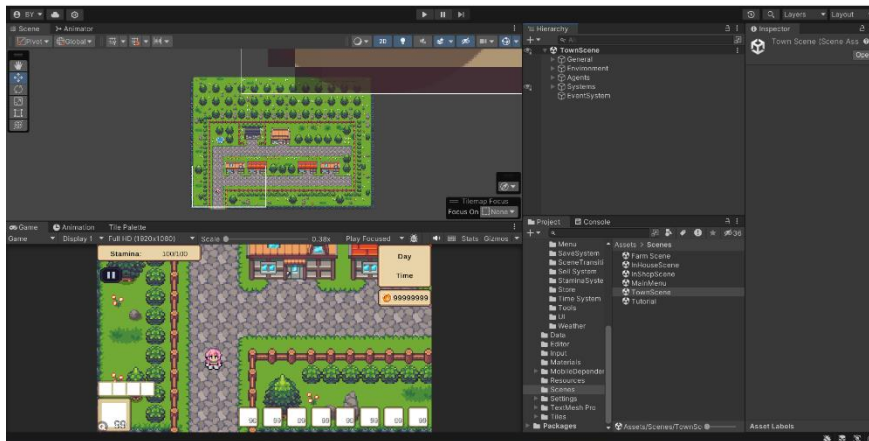


Fig. 11 Town Scene

4.2.8 Code Implementation for In-shop Scene

Figure 12 shows the in-shop scene of the game which is the fifth scene of the game. The player can transition from the town scene to this scene. In the in-shop scene, there is a shop system implemented on the game's only NPC (non-player character), the shopkeeper. Dialogue boxes that are shown if the player interacts with the shopkeeper. The `StoreController` script manages the in-game store functionality. It has fields for store inventory, item/tool databases, player inventories, and store UI. Start method initializes store inventory. `ShowUI` method displays the store UI with player agent and inventory. `MakePurchase` method handles purchasing items/tools - deducting cost from player money and adding to respective inventory. `AddToInventory` method adds items to player inventory. `AddToToolBag` method adds tools to tool inventory or toolbox if full. `OnToolTransferredToBox` event is invoked when a tool is moved to the toolbox due to full inventory. It works with inventory, database, and UI systems. The `NPCDialogueInteraction` script allows the player to interact with an NPC for dialogue. It has fields for `DialogueControllerUI` and `DialogueData` asset. Only the "Hand" tool can be used per `UsableTools`. `Awake` method finds `DialogueControllerUI`. `CanInteract` method checks if agent is player with "Hand" tool. `Interact` prepares dialogue UI by passing `DialogueData` to `DialogueControllerUI` method. This script works with dialogue management, UI, and player interaction systems.

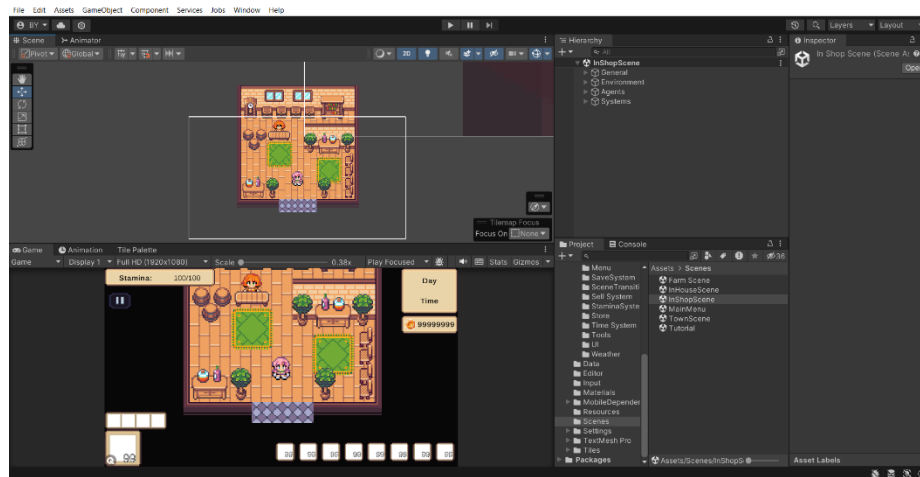


Fig. 12 In-shop Scene

4.2.9 Code Implementation Outside of Scenes

Other than the scripts related to the different scenes in the game, there are also scripts outside the scenes of the game which work with the scripts related to scenes, these scripts are the `WeatherSystem` script, the `DayNightShifting` script, the `TimeSystem` script, the `MoneyController` script, the `StaminaSystem` script and most importantly the `Player` scripts that connect all of these scripts together so that they can work flawlessly.

`WeatherSystem` script manages the weather system using a weather data asset with rainy day data. It finds the `RainControllers` and `TimeManager` scripts and subscribes to `TimeManager` clock and day events. The `SetUpWeather` method checks if current day is rainy based on data and sets `isRaining` flag. `ToggleRain` method enables or disables rain effects on `RainControllers` script while the `CallOnRainEvent` invokes `OnRaining` event when raining. Has `inHouseScene` flag to toggle weather activity.

The `DayNightShifting` script manages day and night cycle using a `Light2D` global light and the `TimeManager` script. It has fields for sunrise, noon, sunset, night hours and light intensities. It subscribes to `TimeManager`'s clock event. The `AdjustLight` method calculates light intensity based on current hour ranges and invokes `OnNightTime` event when day or night transitions for light sources.

The `TimeManager` script manages in-game time system, it has a clock and day events, and it uses `GameCalendar` method for day and week tracking. It also updates the in-game time per frame based on real and game time conversion. It progresses calendar on new days and provides time data with getting and restoring methods. The `TimeEventArgs` method contains the current in-game time details.

The `MoneyController` script manages displaying player's money. It has fields for player data and `MoneyUI` and subscribes to the player data's `OnDataUpdated` event to call `UpdateMoneyUI` method, this method is to update the displayed money value.

The `StaminaSystem` script manages the player's stamina system. It uses player data to store and update stamina data. There are methods in the script to check if enough there are stamina to perform action and stamina will be replenish or modify based on tool and consumables.

The `Player` scripts represent the player character, it has fields for movement, input, animation, interaction, field, UI, inventory, hotbar, carry item. It also has properties and events for state like blocking movement, toggling inventory, tool swapping. It also contains methods for performing actions, swapping tools, toggling inventory, hotbar selections, carrying/consuming items. Saving/restoring data, setting UI visibility, displaying info. This script is important as it coordinates the whole gameplay systems. There is an independent tutorial scene that provides guidance and controls tutorial before starting the game for the player. It has a `SceneTransition` script on a back button to transition to the main menu scene.

4.3 Implementation of Testing Phase

4.3.1 Alpha Testing

Alpha testing was conducted throughout the development phase to test the functionality of the game until the game was fully developed. The functionality of all the inputs, the logic of the scripts was checked and tested during alpha testing to ensure that no game breaking bug is found, and the game is running properly. Improvements and amendments will be made to the game during alpha testing. Problems are fixed in the testing process, for example the tool swapping for the tool bar UI is not functioning after pressing the Q key, therefore, corrective actions have been taken and the assignment of the key is double checked to ensure that the key is working as intended. Table 6 shows the result of the Alpha Testing based on the functionality in “Tranquil Harvest”.

Table 6 *Result of Alpha Testing*

Scene	Testing	Expected Result	Actual Result	Corrective Action
Main Menu	Tutorial button	Navigate to tutorial scene	Works as expected	Not needed
	Start new farm button	Navigate to in-house scene	Works as expected	Not needed
	Load last save button	Navigate to in-house scene if save data is available, else button is disabled	Works as expected	Not needed
Tutorial	Back button	Navigate to Main Menu scene	Works as expected	Not needed
In-House	Bed	Save progress	Works as expected	Not needed
		Progress to the next day	Works as expected	Not needed
	In-house entrance	Navigate to farm scene	Works as expected	Not needed
	Toolbox	Select tool and remove tools from tool bar and store in toolbox	Works as expected	Not needed
		Select tool and remove tools from toolbox and store in tool bar	Work as expected	Not needed
Farm	In-house entrance	Navigate to in-house scene	Works as expected	Not needed
	Farm to Town entrance	Navigate to town scene	Works as expected	Not needed
	Water well	Able to refill watering can	Works as expected	Not needed
	Farm field	Able to pick up debris with hand (Stone, weed, wood log)	Works as expected	Not needed
		Plough field with hoe tool	Works as expected	Not needed
		Plant seeds with different type of seed	Works as expected	Not needed
		Planted field can be watered with watering tool	Works as expected	Not needed
		Crop stop growth when progress to next day if not watered	Works as expected	Not needed
		Crop grow when progress to next day if watered	Works as expected	Not needed
		Crops wither when not watered for two consecutive day	Works as expected	Not needed

Table 6 *Result of Alpha Testing (cont)*

		Remove withered crops with hoe tool	Works as expected	Not needed
		Remove planted crop with hoe tool	Works as expected	Not needed
Sell box		Select item from inventory and remove item from inventory and store in sell box	Works as expected	Not needed
		Select item from sell box and remove item from sell box and store in inventory	Works as expected	Not needed
		Stored items in sell box removed when progress to next day	Works as expected	Not needed
		Add money to player's data according to the value of sold item	Works as expected	Not needed
Weather		Play raining effect based on weather data	Works as expected	Not needed
		Automatically water planted crop when raining	Works as expected	Not needed
		Stop raining effect based on weather data	Works as expected	Not needed
Day Night Shifting		Dim the environment after 7pm game time	Works as expected	Not needed
		Brighten the environment after 6am game time	Works as expected	Not needed
X-ray effect		X-ray effect when player goes behind trees	Works as expected	Not needed
Stamina		Reduce 1 stamina when using hoe tool on field once	Works as expected	Not needed
		Reduce 1 stamina when picking up one debris with hand	Works as expected	Not needed
		Reduce 1 stamina when using watering can on field once	Works as expected	Not needed
		Reduce 1 stamina when using watering can on water well once	Works as expected	Not needed
Town	Town-Farm entrance	Navigate to farm scene	Works as expected	Not needed
	Town to in-shop entrance	Navigate to in-shop scene	Works as expected	Not needed
Weather		Play raining effect based on weather data	Works as expected	Not needed
		Stop raining effect based on weather data	Works as expected	Not needed
Day Night Shifting		Dim the environment after 7pm game time	Works as expected	Not needed
		Brighten the environment after 6am game time	Works as expected	Not needed
In-shop	In-shop to town entrance	Navigate to town scene	Works as expected	Not needed
	NPC Dialogue box	Show dialogue box when interacting with NPC	Works as expected	Not needed
		Show next dialogue when pressed space bar	Works as expected	Not needed
Shop UI		Show shop UI after NPC dialogue	Works as expected	Not needed
		Show all items for sale in shop UI	Works as expected	Not needed

Table 6 Result of Alpha Testing (cont)

		Select item to buy in shop UI	Works as expected	Not needed
		Show confirmation UI after selected item	Works as expected	Not needed
		Change buy quantity in confirmation UI	Works as expected	Not needed
		Disable buy button if player doesn't have sufficient money	Works as expected	Not needed
		Enable buy button if play have sufficient money	Works as expected	Not needed
		Item added to inventory after selected buy button	Works as expected	Not needed
		Close shop UI after buying item	Works as expected	Not needed
		Bought item added to toolbox if tool bar is full	Works as expected	Not needed
In-house, farm, town, in-shop	Stamina UI	Stamina amount is shown correctly when transition from once scene to another	Works as expected	Not needed
		Stamina is replenished to 100 after progressing to next day using bed	Works as expected	Not needed
		Show "Not enough stamina" dialogue box if stamina is 0 when performing action that requires stamina	Works as expected	Not needed
	Hot bar UI	Show the top 8 items in inventory correctly	Works as expected	Not needed
		Show the top 8 items in inventory correctly if items is swapped in inventory	Works as expected	Not needed
		Player hold an item in inventory shown in hotbar when pressed on either 1 to 8 key	Works as expected	Not needed
		Player consume consumable item when holding consumable item after pressing space bar	Works as expected	Not needed
		Player remove unconsumable item when holding unconsumable item after pressing space bar	Works as expected	Not needed
		Player stop holding item after pressing ESC key when holding an item	Works as expected	Not needed
	Inventory UI	Show inventory UI after pressing i key	Works as expected	Not needed
Swap placement of two items in inventory UI		Works as expected	Not needed	
Close inventory UI after pressing i key or ESC key		Works as expected	Not needed	
Tool bar UI	Swap between tools when pressing q key	Tools not swapped when pressing q key	Make sure that the right key is assigned	

4.3.2 Beta Testing

Beta testing was conducted involving target user after the game was fully developed. This test was performed to get feedback from target users based on their user experience when playing "Tranquil Harvest". A set of questionnaires was prepared using Google Form, and graphs and charts were generated based on the results obtained from the questionnaires. A total of 30 children participated in the Beta test. Figure 13 shows the analysis of gender. There is a total of 17 out of 30 respondents (56.7%) are male and 13 out of 30 respondents (43.3%) are female.

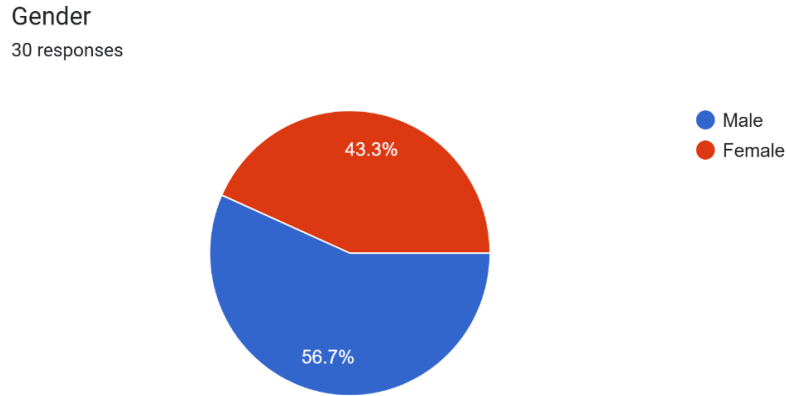


Fig. 13 Analysis of Gender

Figure 14 shows the Analysis of Component 1, which is the Learning Outcome Acquisition. In response to Question 1, 100% of respondents answered "Yes", 0% of respondents answered "No" and "Maybe". In response to Question 2, 96.7% of respondents answered "Yes", 0% of respondents answered "No" and 3.3% of respondents answered "Maybe". According to the analysis, most of the respondents provided a positive response, meaning that most of them had good learning outcome by playing "Tranquil Harvest". Participants showed positive inclination towards acquiring knowledge about growing different types of local plants and vegetables after playing this game and show high interest towards practicing it in real environment.

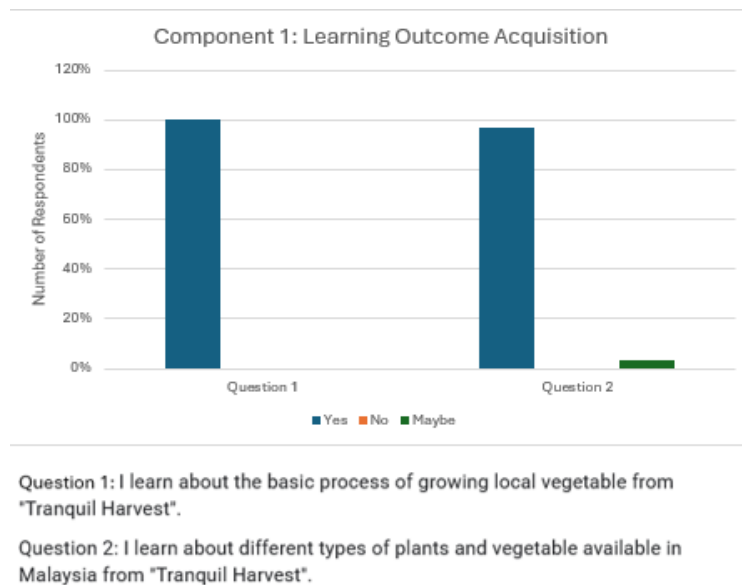


Fig. 14 Analysis of Component 1

Figure 15 shows the Analysis of Component 2, which is the User Acceptance Level. In response to Question 1, 100% of respondents answered "Yes", 0% of respondents answered "No" and "Maybe". In response to Question 2, 83.3% of respondents answered "Yes", 0% of respondents answered "No" and 16.7% of respondents answered "Maybe". In response to Question 3, 100% of respondents answered "Yes", 0% of respondents answered "No" and "Maybe". According to the analysis, most of the respondents agreed that "Tranquil Harvest" provides a good interface design with clear instructions that are easy to understand, and they enjoy playing the game. This result proves that Tranquil Harvest was designed by adhering to good design principles which eventually increases user experience.

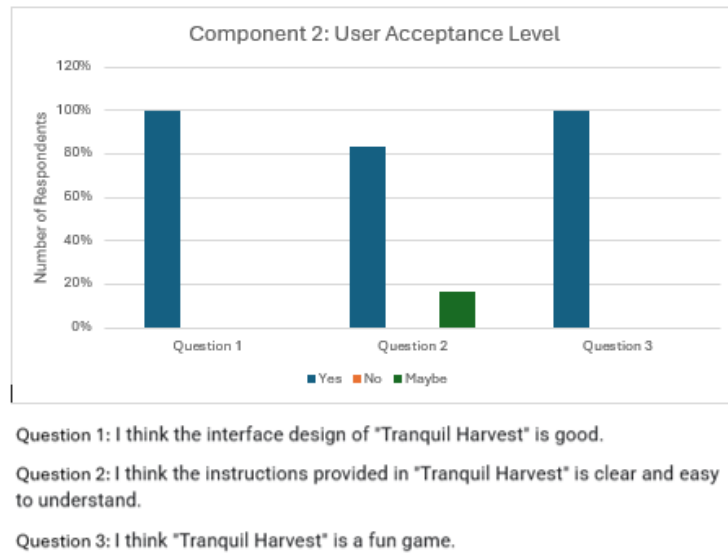


Fig. 15 Analysis of Component 2

Figure 16 shows the Analysis of Component 3, which is the Functionality. In response to Question 1, 96.7% of respondents answered “Yes”, 0% of respondents answered “No” and 3.3% of respondents answered “Maybe”. In response to Question 2, 100% of respondents answered “Yes”, 0% of respondents answered “No” and “Maybe”. From this result, most of the respondents agreed that they can understand the function of each input of button in the game and they did not encounter any errors in the game. Familiarity is an important aspect of design that enables users to understand how to use the interface. This result reflects that Tranquil Harvest indeed provided users with visible and clear functions that eventually ease the use of the interface.

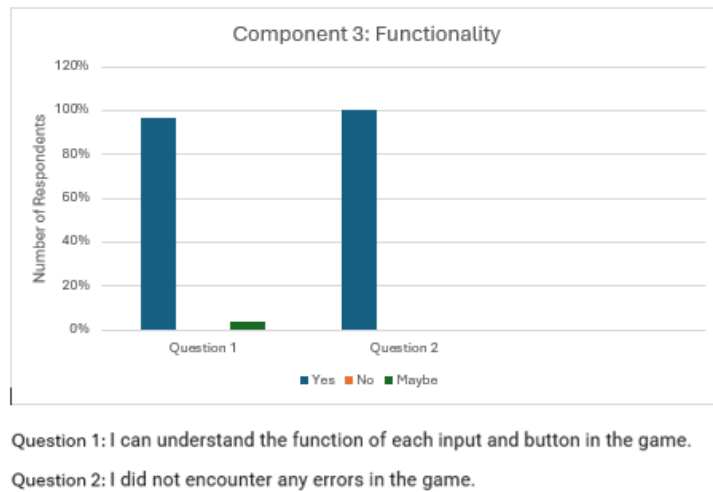


Fig. 16 Analysis of Component 3

Overall, the project was successfully completed, resulting in a fully functional game that met the project objectives. In the pre-production phase, the detailed planning and design ensured that the development phase proceeded smoothly, with a clear vision and structure. In the production phase, the use of Unity and Aseprite allowed for efficient development and iteration. The development of graphic assets, animations and the implemented features such as scene transitions and inventory management contributed to an engaging gameplay experience. In the testing phase, alpha testing phase was crucial in identifying and fixing initial bugs, while beta testing provided valuable feedback from real users. The positive responses from beta testers confirmed the game's effectiveness in educating players about local plants and vegetables and its overall enjoyability.

5. Conclusion and Future Work

The chapter discusses the achievements, limitations, and potential future work for the "Tranquil Harvest" game project. According to the result of the testing phase, it can be concluded that the objectives of "Tranquil Harvest" has been achieved. The objectives of this project are:

To design a 2D farming simulation role-playing game: Tranquil Harvest for PC platform. The flowchart, storyboard and the button and interface design of "Tranquil Harvest" was designed and discussed in Section 3 and 4.

To develop the Tranquil Harvest using Unity game engine. "Tranquil Harvest" was successfully developed for the Windows platform using Unity game engine. The development and build process of the game was discussed in Section 4.

To perform alpha and beta testing on the developed game to target user. User acceptance level testing was conducted during the beta testing phase to gather feedback from the target users. Thirty respondents took part in the beta testing and the result was discussed in Section 4.3.

Key advantages of "Tranquil Harvest" includes providing education about local plants/vegetables, using simple English, having an easy control system, and featuring a good graphics/interface. Limitations are also noted, such as only having 13 plants to unlock, being Windows-only, and lacking sufficient sound effects. To address the limitations, future work recommendations include adding more plants, porting to other platforms like iOS, Android, Mac, Linux, and enhancing sound effects with things like rain sounds and reminders about in-game time.

In conclusion, the project involved thorough planning and design, resulting in a functional and engaging game that educates players about local plants and vegetables through immersive gameplay. Feedback from alpha and beta testing confirmed its educational value and overall enjoyability. Despite its successes, the game has limitations, such as the number of plants and platform exclusivity. Future work aims to address these by expanding plant varieties, enhancing sound effects, and porting to additional platforms. This project effectively bridges the gap between urban populations and agricultural knowledge, fostering a deeper appreciation for local flora and sustainable farming practices.

Acknowledgement

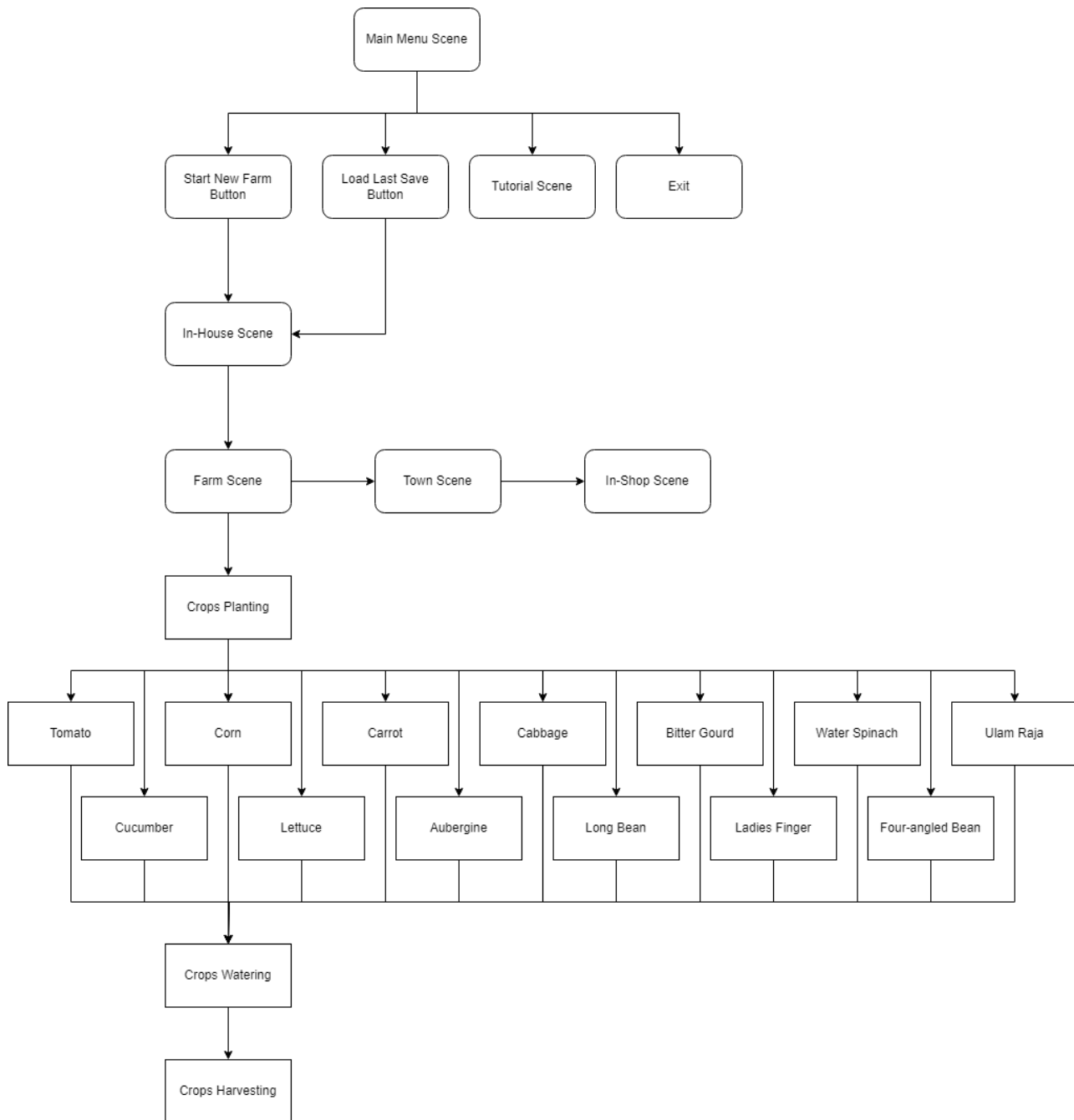
The authors would like to thank the Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia for its support.

The authors confirm contribution to the paper as follows: **study conception and design:** Lee Bing Yang, Che Samihah; **data collection:** Lee Bing Yang; **analysis and interpretation of results:** Lee Bing Yang, Che Samihah; **draft manuscript preparation:** Lee Bing Yang, Che Samihah. All authors reviewed the results and approved the final version of the manuscript.

References

- [1] M. Torero. (2023). "Global food security challenges and its drivers: conflicts and wars in Ukraine and other countries, slowdowns and downturns, and climate change". *Food and Agriculture Organization of the United Nations* [Online]. vol. 172, issue 5. Available: <https://www.fao.org/3/nl652en/nl652en.pdf>
- [2] Department of Statistics Malaysia. (2023, Feb. 28). *Annual Economic Statistics of the Agricultural Sector* [Online]. Available: <https://www.dosm.gov.my/portal-main/release-content/0ae5d1df-8ba9-11ed-96a6-1866daa77ef9>
- [3] International Trade Administration. (2022, Nov. 23). *Malaysia – Agricultural Sector* [Online]. <https://www.trade.gov/country-commercial-guides/malaysia-agriculturalsector#:~:text=Malaysia%20has%20a%20relatively%20strong,input%20subsidies%2C%20and%20consumer%20subsidies>
- [4] A.D. Rozhan. (2022). Overview of the Agricultural Sector during the 11th Malaysian Development Plan (2016-2020) [Online]. Available: <https://ap.fftc.org.tw/article/3010#:~:text=Agriculture%20is%20one%20of%20the,11%20development%20plans%20since%201952>
- [5] D. J. Lockwood., "Engaging people to think deeply about agriculture and its role in society: The key role of interdisciplinary education," *Journal of Agricultural Education*, vol. 40, pp. 8-14, 1999.
- [6] L.W. Charlotte, "Developing Goals and Objectives for Gameplay and Learning," in *Learning, Education and Games*, Carnegie Mellon University ETC Press, 2014, pp. 225-249.
- [7] R. Ramadan and Y. Widyani. (2013). "Game development life cycle guidelines". *2013 International Conference on Advanced Computer Science and Information Systems* [Online]. *ICACSIS 2013*, pp. 95-100. Available: <http://doi.org/10.1109/ICACSIS.2013.6761558>.

Appendix A



Appendix B

