Department of Electronic & Telecommunication Engineering (EN)
University of Moratuwa

Proposal – Hybrid Farm

EN3350 - Software Design Competition

1. Team Introduction

Name	Index Number
L. G. S. Clarance	200094R
V. Kurrshanth	200331T
C. M. C. C. Miranda	200396U
R. Sairisan	200552V

2. Game Concept Overview

2.1. Concept Summary [Hybrid Farm]

Our game is a 2.5D point-and-click strategic gameplay that enables the player to manage the production processes on a farm, make decisions to increase profit, and complete objectives in less time to secure a position on the leaderboard. The resources the player will be managing include different livestock (farm animals) such as chickens, pigs, and cows (not finalized). As the profit increases the player will have the ability to build small factories to produce byproducts from the base products. We have planned to have one base product from each livestock and to use it to generate two byproducts. There will be two types of byproduct factories for each livestock type. The player will use a transport vehicle to export their products to market and receive money and he will purchase grass from a dedicated store to feed the livestock. There are time delays between each grass purchase, producing batch of byproducts and receiving profits from transported goods.

There will be a time limit for the player to achieve specific objectives, such as reaching a profit target, producing a certain number of eggs, milk, or meat (base products), creating a certain number of egg powder packets, cakes, curd, cheese, bacon, and fast food (byproducts). Based on his finishing time, he will get gold, silver, or bronze trophies. Player's aim is to finish the objectives in less time to secure a place on the global leaderboard. There will be five upgrades available for the factories and the transport vehicle. Upgrading the factories will reduce production time for byproducts and increase the number of byproducts produced at once. Upgrading the transport vehicle will increase vehicle storage and decrease transportation time. The player has to make strategic decisions on which livestock to raise using his initial money, which factories to be built and which transport vehicle to be used as he continues the gameplay while making profit. When building and upgrading the factories, for each type of factory the player gets to choose from renewable energy powered and fuel-powered factories. When choosing the solar-powered factories, the player gets the advantage of extra time for completing his objectives but at the cost of more money for building. Similarly, when upgrading his transport vehicle, he gets to choose electric vehicle or fuel powered vehicle. When choosing the electric vehicle, he can get the goods delivered faster in turn make profit faster, but the upgrade will cost more. Strategic upgrades and decisions play a crucial role in helping the player climb the global leaderboard effortlessly.

At random times, the farm will come under attack from certain predators. The energy saving actions of the player which are retrieved through the API every 10 seconds will affect how often the farm comes under predator attack, how many predators will attack in one go and the level of the predators [Severity of the damage]. The player must capture the predator [by continuously clicking] before it kills the livestock. The player then can sell the captured predator or just keep it as it is captured. In addition, the API data also affects the time delays mentioned above for-profit generation via export, grass purchase and byproduct production.

To enhance the farm's defense, the player can raise a guardian dog, with five available upgrades. Each upgrade increases the dog's power of defending against predators. At higher upgrades, the dog gains extra power, allowing it to hint to the player about the power consumption, retrieved through the API. Dog upgrades become essential as higher power consumption attracts more powerful predators, which only high-level dogs can control. Similarly, there is an option to raise a cat responsible for collecting the base products of the livestock. Base products are time-sensitive, and if not collected promptly, they will disappear. The cat's upgrades increase the collection area and speed. Players must strategically utilize their resources to achieve objectives within the specified time limit.

2.2. Educational Objectives

The game inspires the player to make decisions on energy efficient upgrades in the game play to receive added benefits while instilling the importance and benefits of those decisions. Throughout the gameplay we motivate the player to go for energy efficient upgrades by providing them with the advantages they can get. For example, even though the initial cost is high they can make more profit and finish the objectives in less time to get himself on the global leaderboard as the gameplay goes on with these energy efficient upgrades. Since the primary goal is to secure a place on the leaderboard, choosing energy-efficient upgrades becomes essential. Without them, achieving objectives in the shortest time possible would be challenging.

3. Development Plan

3.1. Phase 0 - Player Authentication/Registration

We implement a component for handling the API calls for the whole game. This component will be the controller for getting and sending API responses and requests. There will be a component for loading the *main menu* which will talk with this *API controller* to get necessary values to authenticate the player, do the authentication and load the main menu.

3.2. Phase 1 - Player Profile

We will create another component for the *player profile*, and we retrieve already available information through our *API controller* and save it in our player profile component. In the *main menu* component, we implement an option for the player to create profile. When he clicks that we access the player profile component and retrieve already available values in a form like interface where there will be empty spaces for the users to fill in the missing details. After filling in the details and pressing submit button the missing values will be updated in the profile component locally, then the *API controller* will be called to save the profile details through API. If any information is not filled in there will be a prompt to fill in the missing information and after completing the profile the game environment will be visible.

3.3. Phase 2 – Questionnaire

For the first time players after the profile completion, the *main menu* component will prompt the user to click a button for the questionnaire asking the player to attempt. After clicking the button, the *main menu* component will redirect the player to the questionnaire web application in a separate browser tab and waits for the response with the boost score from the web app. Once the player enters the web application, the backend will retrieve the questions from the database one-by-one with the answer choices and send them to the front end, there it will format it for the user to select the answer. Once the answer is selected it will be sent to the backend, and the backend evaluates and stores the results in database. After completing every question, the backend retrieves the evaluation results from the database along with the correct answers and questions, it sends them to the frontend to show the results to the user. While this happens, the boost score of the player will also be calculated in the backend. After the player reviews the questions and clicks the button, the webapp will redirect the player to the main menu component with the boost score. To make this communication happen between these two applications, we will use the *API controller* to handle local HTTP communication.

3.4. Phase 3 - Game Environment

As mentioned in the game description the farm will come under attack from predators at certain random times. The frequency of these attacks, the number of attacking predators and the severity of the predators will depend on the player's energy consumption retrieved by the API calls every 10 seconds. We use the "*View daily power consumption by current month*" API provided to get the daily power consumption of the user for the current month, and we implement our own algorithm to calculate the nature of the above attacks. Also, the time delays for profit generations [Time to transport the good to market receive money], for consecutive grass purchases and for by-products production completion will change according to the player's energy consumption. This also will be done by the algorithm. So, a player will experience difficulty playing the game when his energy consumption increases so that it'll be very hard for him to finish the tasks in the least time. So that, he can't let himself in the leaderboard.

3.5. Phase 4 - Leaderboard

The player will receive trophies [Gold, silver, and bronze] according to the time he takes to complete the objectives and pass the objectives. There will be a defined score for each trophy that the player achieves. Also, the player will get extra score to his choices in energy efficient upgrades and the profit he makes. In the game environment there will be a button to view the leaderboard. After every objective, he can be able to view the leaderboard along with the time he took to finish the respective objective. In order to move higher on the leaderboard, he can replay the respective objective. But to finish in less time than before, he has to take necessary actions to reduce the power consumption in his home, so that he will get the advantages of lesser predator attacks and faster profit generation and so on. After finishing all his objectives, he can view the overall global leaderboard. To make himself on top he has to replay all the objectives with the taking power saving actions at home, which leads him to finish the objective faster. To make the leaderboard we use the "*View Player List*" API to get the list of each player. To calculate the leaderboard, we use the current player's score as the base value for which we use the retrieved other players' information to calculate a variation using our own algorithm and calculate the other players' score. This leaderboard will be calculated after each objective using the current time taken by the player and will be saved in the database and retrieved for the view when clicking then 'Leaderboard' button and after each objective.

4. Expected Challenges and Solutions

4.1. Challenges

- Difficulties in making the game art. [Models, Sprites]
- Remote team-members.
- Difficulties in finding the audio and sound effects for the game.
- Learning new language [C#, Java], software stack [Unity, Photoshop, Illustrator, React, Aseprite].
- Time constraint due to internship workload.

4.2. Solutions

- Learning graphic design for making simple and unavailable game arts.
- Using royalty-free game arts, sound effects and software.
- Using GIT, GitHub, and MS Teams to collaborate.
- Learning through online courses and other online platforms.
- Proper time management and workload management.