LBYCPEI Aquaman Final Project Proposal

In partial fulfillment of the course

in **LBYCPEI - EQ3**

Entrepreneur Simulator

Submitted by:

Lim, William Anderson C. -

Ty, Josh Angelo P. -

Tan, Irabelle Pristine Y. -

Submitted to:
Mr. Ramon Stephen Ruiz

I. Introduction/Description

Entrepreneur Simulator is a simple Java application akin to the famous Cookie Clicker game series, it teaches users to reinvest their money for continued and sustainable economic growth. It allows users to "work" (click) for money and reinvest that money into different investments that can grow overtime (investment profits/revenue). It features two ways to earn money, one for the shop and passive income (investment), and the other for active income ("work" - clicking). The goal of the game is to raise a certain amount of cash as fast as possible. The time it takes to get to that amount will be recorded and be shown in the end screen before taking them back to the main menu. There will also be an alternative game mode to gain as much cash as possible within a certain time limit. The score will similarly be shown in the end screen but using a different metric and send them back to the main menu.

II. Methodology

- Menus (Major Phase)
    - The developers will start by coding menu option displays, preparing to code for the interactions with the menu objects
- Graphic Interface (Major)
    - The developers will develop the User Interface (UI) of the game that will display the relevant statistics of the player (money, time passed, etc.)
- Game Design (Major)
    - The developers will code in the mechanics and functions used in the game, and tie them to the menu and user interface implemented in the earlier phases
    - Rules:
        - Earn money by starting tasks
        - Use earned money to invest and expand to other businesses and earn more money faster
        - Hire managers along the way to automate certain tasks
        - Reach a certain amount of money in the shortest time possible
    - Additional Features:
        - The Investment Shop
        - Clicking
        - Local Save
        - More to come
- Debugging (Minor)
    - Fixing up the code for any issues to ensure that it works as intended
- Four Pillars (Major)
    - Abstraction

- Abstraction in this program will be done through the use of JavaFX and SceneBuilder, making an application that acts as an interface for the user. Basically, it creates a graphical user interface to make the program easier to use. This allows the coders (us) to hide the code from the user, but still allow them to use the program through tool interaction.

- Encapsulation
  - Encapsulation will be used in this program through the Investment class wherein its attributes are under the protected access modifier. This will result in encapsulation as these parameters cannot be edited by external code and can only be modified/accessed by the class objects/instances themselves.
- Inheritance
  - Inheritance will be used through the subclasses of the Investment class, specifically all the different investment types in our game (i.e. Lemonade, Dog, Stock, Resto). This allows the subclasses to utilize the attribute characteristics and methods of the Investment class, and makes it easier to update/utilize them due to their similarity to each other under the investment class.
- Polymorphism
  - Polymorphism will be demonstrated also through the methods present in the Investment class. What we will specifically use is Static Polymorphism/Method Overloading. This allows the "same" method to be called but the return value is determined by the type of variable that was passed through it. This allowed us to utilize the "same" method to update different values when an event is done.

III. Project Description

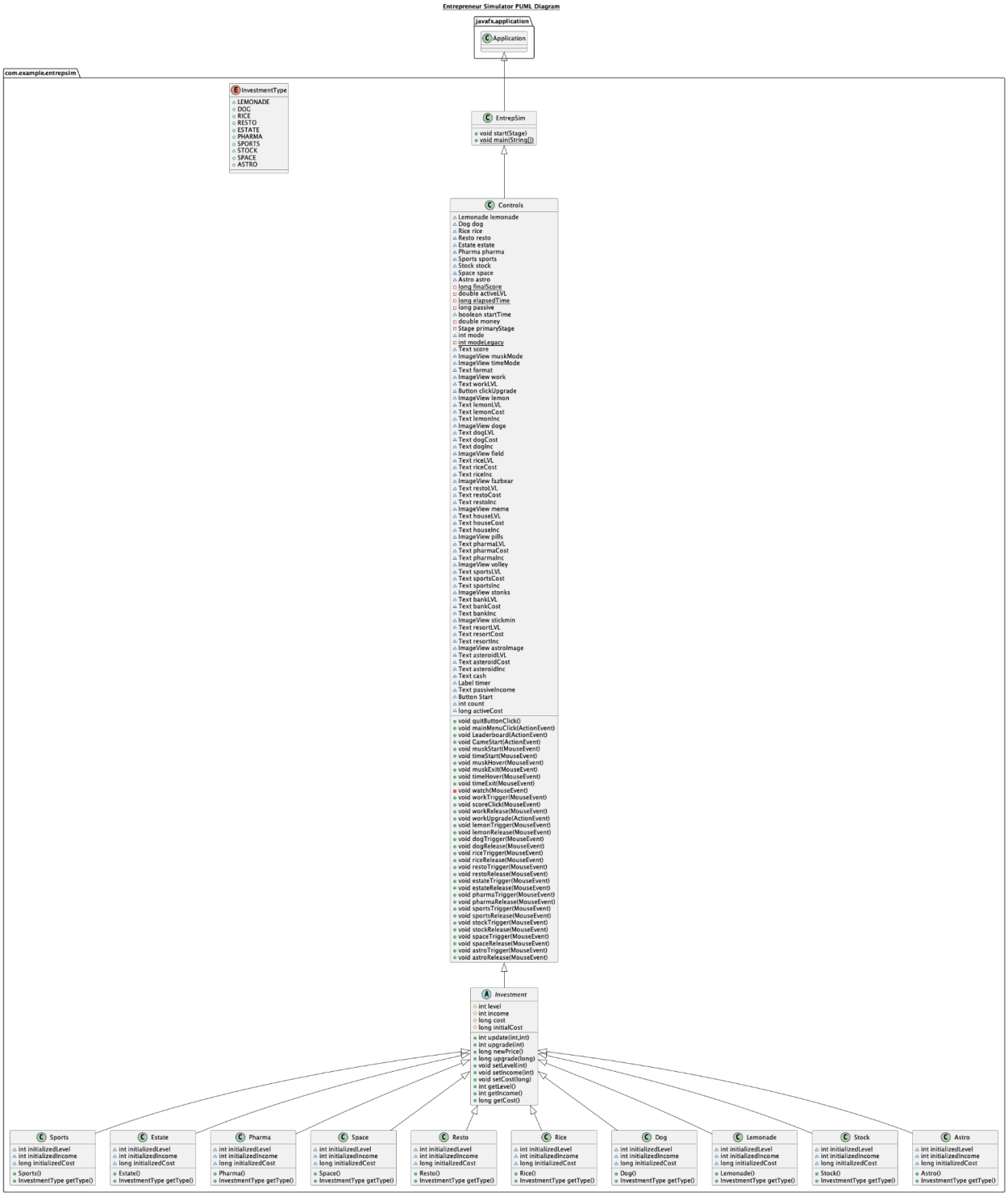
IPO:

Input	Process	Output
Pressing the Game Start Button	Runs GameStart(ActionEvent event)	Loads "Gamemodes.fxml"
Pressing the Credits Button	Runs Credits(ActionEvent event)	Loads "Credits.fxml"
Pressing the Quit Button	Runs quitButtonClick	Ends the Program
Pressing the "Mode: Elongated Muskrat" Button	Runs muskStart(MouseEvent event)	Loads "Game.fxml" with the gamemode of getting to $1,000,000,000 as the goal
Pressing the "Mode: Devouring Time" Button	Runs timeStart(MouseEvent event)	Loads "Game.fxml" with the gamemode of earning as much

<table>
<tr><td></td><td></td><td>money as possible in 30 minutes as the goal</td></tr>
<tr><td>Pressing the Lemonade Upgrade Button</td><td rowspan="10">Checks if the player has enough money to upgrade the component’s passive income value</td><td rowspan="6">Meets Condition: Subtracts the cost from the player’s available money, increases the passive income, and updates the new cost for upgrade, while displaying a yellow border around the button</td></tr>
<tr><td>Pressing the Dog Walk Upgrade Button</td></tr>
<tr><td>Pressing the Rice Field Upgrade Button</td></tr>
<tr><td>Pressing the Restaurant Upgrade Button</td></tr>
<tr><td>Pressing the Real Estate Upgrade Button</td></tr>
<tr><td>Pressing the Pharmaceuticals Upgrade Button</td></tr>
<tr><td>Pressing the Sports Team Upgrade Button</td><td rowspan="4">Doesn’t Meet Condition: Displays a red border around the button</td></tr>
<tr><td>Pressing the Bank Upgrade Button</td></tr>
<tr><td>Pressing the Space Resort Upgrade Button</td></tr>
<tr><td>Pressing the Asteroid Mining Upgrade Button</td></tr>
<tr><td>Pressing the Work Button</td><td>Starts the Timer on first click; adds the value of activeIncome to the available money of the player</td><td>Updates available money by increasing it by the amount in activeIncome</td></tr>
<tr><td rowspan="2">Pressing the Work Upgrade Button</td><td rowspan="2">Checks if the player has enough money to upgrade the active income value</td><td>Meets Condition: Subtracts the cost from the player’s available money, increases the active income, and updates the new cost for upgrade</td></tr>
<tr><td>Doesn’t Meet Condition: Does Nothing</td></tr>
<tr><td>Clicking on the Score Text</td><td>Extracts the type of gamemode to determine what score to display</td><td>Elongated Muskrat Mode: Displays the Elapsed Time as the score</td></tr>
</table>

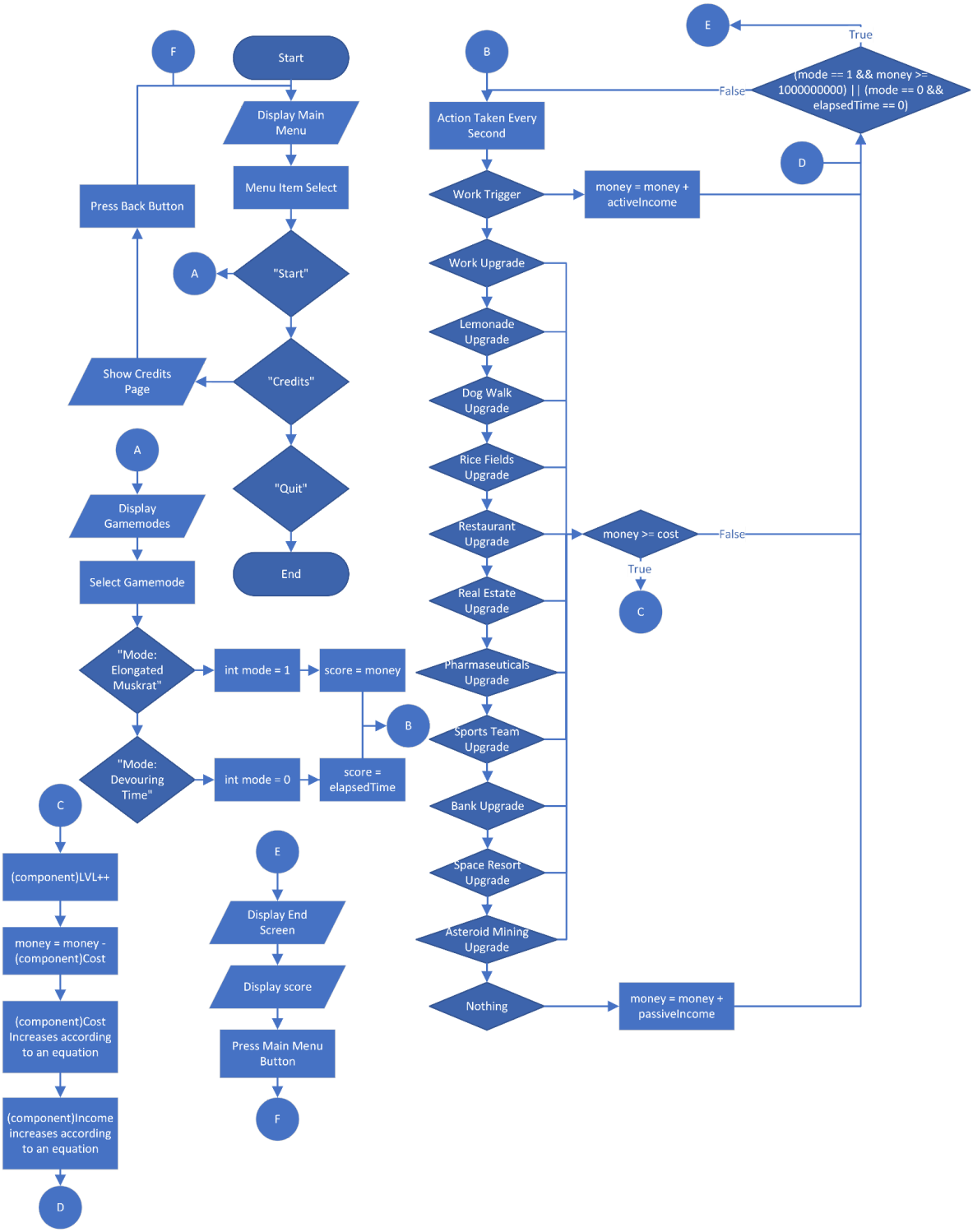
		Devouring Time Mode: Displays the amount of money in the player's possession as the score

UML:



https://drive.google.com/file/d/1GdxIReYRw6lFzp0oqVF0usDjvh-jt5hF/view?usp=sharing

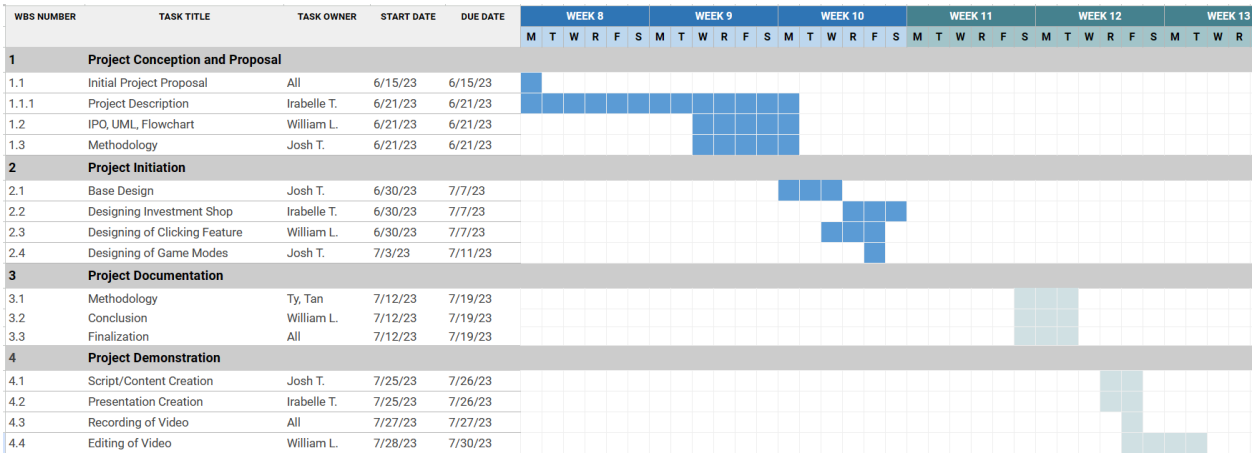
Flowchart:



https://drive.google.com/file/d/1sfFYQtrHVQUj5ukF_iAPy_vanV7AWGY7/view?usp=sharing

IV. Deliverables

GANTT Chart



WBS NUMBER	TASK TITLE	TASK OWNER	START DATE	DUE DATE
1	**Project Conception and Proposal**			
1.1	Initial Project Proposal	All	6/15/23	6/15/23
1.1.1	Project Description	Irabelle T.	6/21/23	6/21/23
1.2	IPO, UML, Flowchart	William L.	6/21/23	6/21/23
1.3	Methodology	Josh T.	6/21/23	6/21/23
2	**Project Initiation**			
2.1	Base Design	Josh T.	6/30/23	7/7/23
2.2	Designing Investment Shop	Irabelle T.	6/30/23	7/7/23
2.3	Designing of Clicking Feature	William L.	6/30/23	7/7/23
2.4	Designing of Game Modes	Josh T.	7/3/23	7/11/23
3	**Project Documentation**			
3.1	Methodology	Ty, Tan	7/12/23	7/19/23
3.2	Conclusion	William L.	7/12/23	7/19/23
3.3	Finalization	All	7/12/23	7/19/23
4	**Project Demonstration**			
4.1	Script/Content Creation	Josh T.	7/25/23	7/26/23
4.2	Presentation Creation	Irabelle T.	7/25/23	7/26/23
4.3	Recording of Video	All	7/27/23	7/27/23
4.4	Editing of Video	William L.	7/28/23	7/30/23

https://docs.google.com/spreadsheets/d/1yHnAVKcrwASdkrYdXBjK_bfYiJXMHS1EJivPpOp2xDg/edit?usp=sharing

V. References

- Adventure Capitalist (https://hyperhippo.com/games/adventure-capitalist/)
    - What makes our game stand out compared to other games is that we incorporate 2 subgenres of the game into one, which will utilize split canvas display to make it happen, and that it is a time-trial leaderboard that records how fast a player can reach a set monetary goal, which other games usually don't have, as they are designed to be endless.
- Cookie Clickers (http://orteil.dashnet.org/cookieclicker/)
    - The game won't just be a cookie clicker that requires endless monotonous clicking, but more akin to starting a timer for a task to be able to actually generate revenue.