Project Plan

This project’s goal is to make a program that allows the user to run a small farm. The player begins the game with a sum of money and no plants or supplies. The player can buy these items and begin managing their farm. They must water their plants to make them grow. Supplies can further help plants grow. The player then moves time forward to see their plants grow. The player can then sell plants to buy additional plants and equipment.

## Potential Classes

* Shop: The shop contains the items needed for the player to manage their farm.

Data and function:

* Int money
* Sup supplies (array of class sup)
* Plant plants (array of class plant)
* Void sell\_plant(plant)
* Void sell\_sup(sup)
* Void buy\_plant(plant)

Relationships:

* Child class of the same parent class as the player
* Contains arrays of other made classes (plant and supplies)
* Player: The player interacts with the shop and manages the farm.

Data and function:

* Int money
* Sup supplies (array of class sup)
* Plant plants (array of class plant)
* Void water\_plant(plant)
* Void sell\_plant(plant)
* Void buy\_plant(plant)

Relationships:

* Child class of the same parent class as the shop
* Contains arrays of other made classes (plant and supplies)
* Affects the state of other classes (plant and farm)
* Farm: The farm contains all the supplies and plants the player currently has.

Data and function:

* Int growth\_factor
* Sup supplies (array of class sup)
* Plant plants (array of class plant)
* Void add\_plant(plant)
* Void rem\_plant(plant)
* Plant \* return\_plants()
* Void add\_sup(sup)

Relationships:

* Contains arrays of other made classes (plant and supplies)
* Player can add and remove plants from the farm
* Supplies: The supplies are items bought from the shop that improve the farm.

Data and function:

* Int price
* Char name
* Int growth\_factor

Relationships:

* None ATM
* Plants: The plants are exchanged between the player and the shop and can grow at the player’s farm.

Data and function:

* Int price
* Char name
* Int growth

Relationships:

* None ATM

## Project Timeline

In week 9, the primary focus will be creating all the classes for the program and correctly implementing the relevant relationships such as inheritance and polymorphism.

In week 10, the focus will be making sure that all classes correctly interact with each other as intended within the program. Issues at this point in development may require us to rethink our classes or their interactions to overcome challenges.

In week 11, the majority of the program should be completed and the focus should be on improving user experience such as finalizing the user interface.

## User Interface

The user interface will be texted base as it allows for simple interaction between the user and the program. Text also allows for information to be easily presented to the use as well as making user input relatively simple.

## Testing and Debugging

Testing will be a continuous process throughout the development of the program. Test cases will be made for each class to test the relevant functions during their creation as well as ensure that the relationships between classes are successfully implemented. Once the program is relatively operational, testing will be more focused on attempting to ‘break’ the program in ways that may occur when a player uses the program. This will help to further polish the program.