FARM TYCOON

Created by

Nutthapong Dissanont 6530128321

Patthapol Kittikun 6530285121

2110215 Programming Methodology

Semester 1 Year 2023

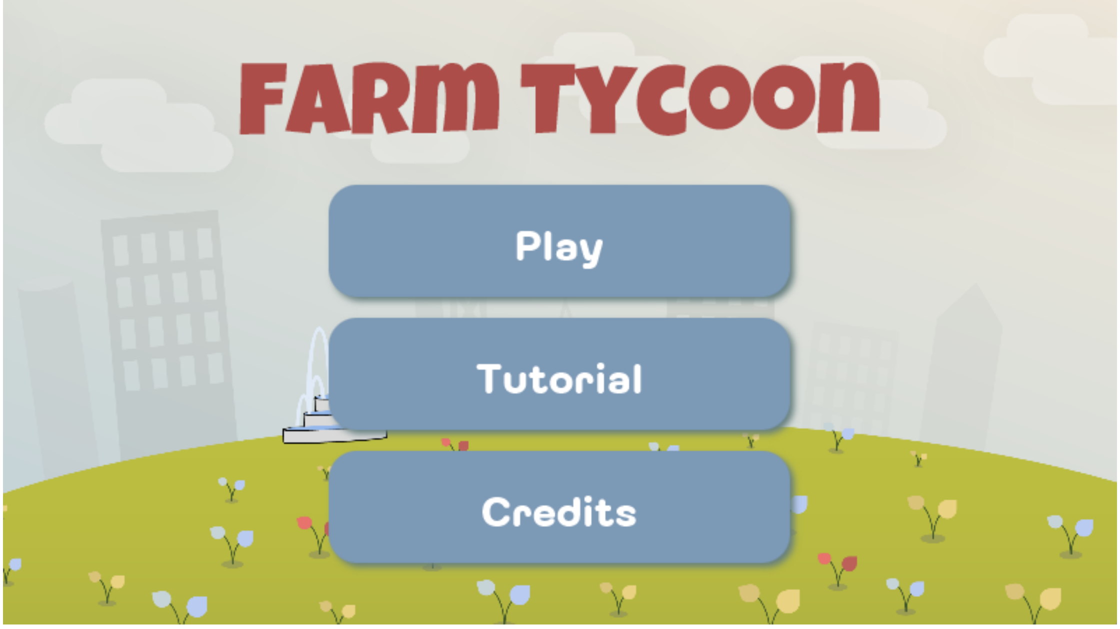
Chulalongkorn University

FARM TYCOON

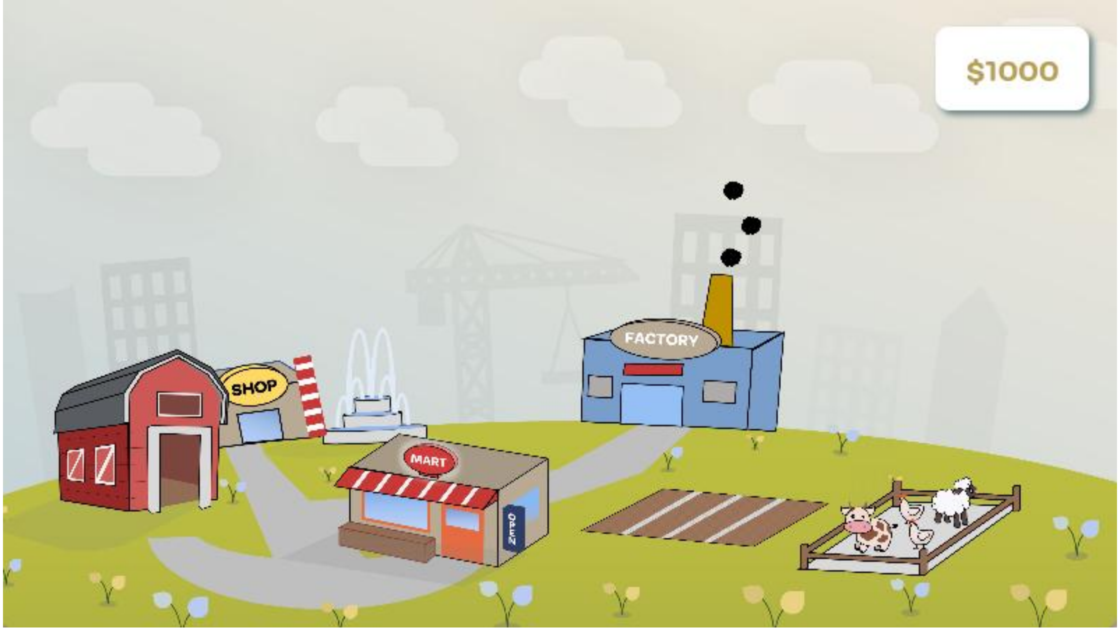
Introduction

Farm Tycoon is a farming simulator game inspired by Hay Day, in which players can plant diverse crops, raise adorable animals, sell products, and so much. Players can unlock various kinds of seeds, from crops to stunning flowers, and watch the farm flourish. The objective of this game is to build a thriving business and expand the farm.

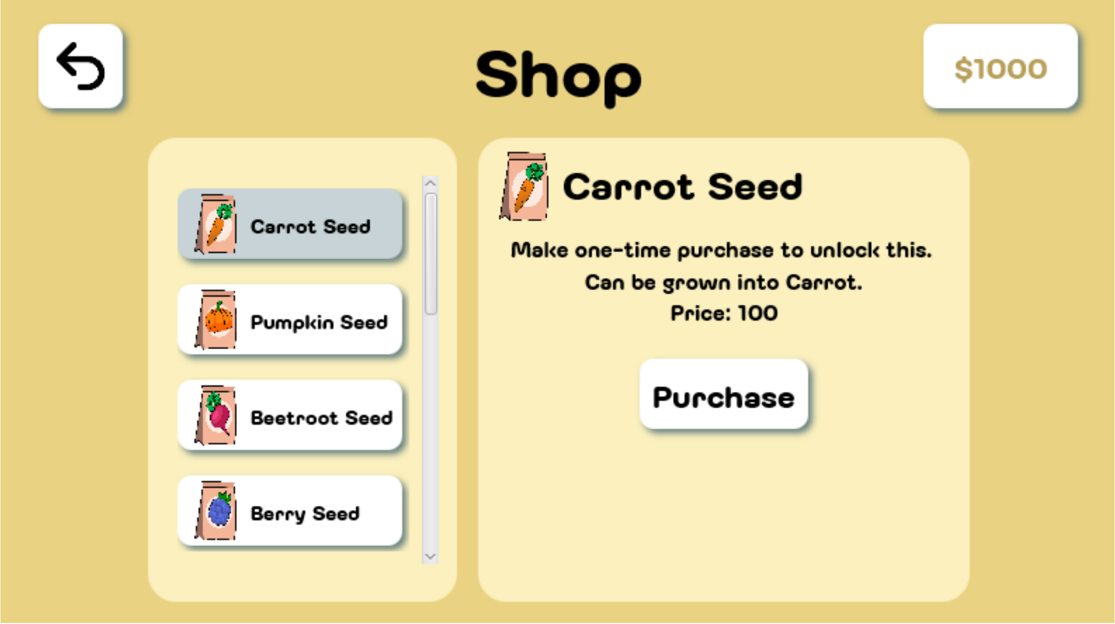
Main menu page



How to play



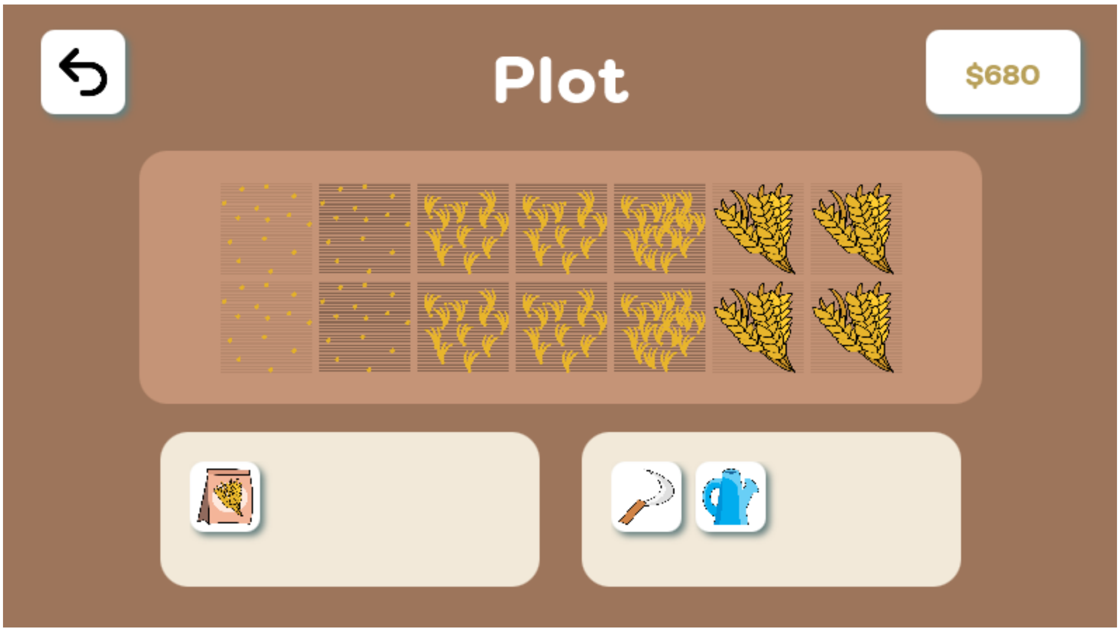
When start, the player will receive $1000 to initiate the farm business. There are 9 seeds in this game, i.e. wheat seed, carrot seed, pumpkin, seed, beetroot seed, berry seed, dandelion, seed, orchid seed, poppy seed, and tulip seed.



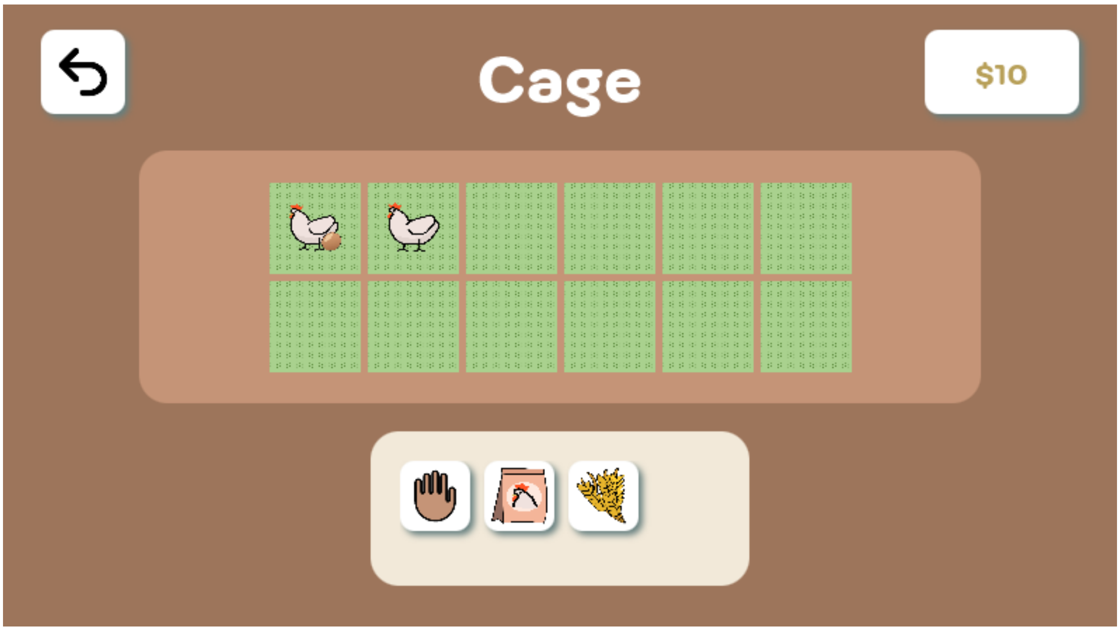
Carrot Seed

Make one-time purchase to unlock this.
Can be grown into Carrot.
Price: 100

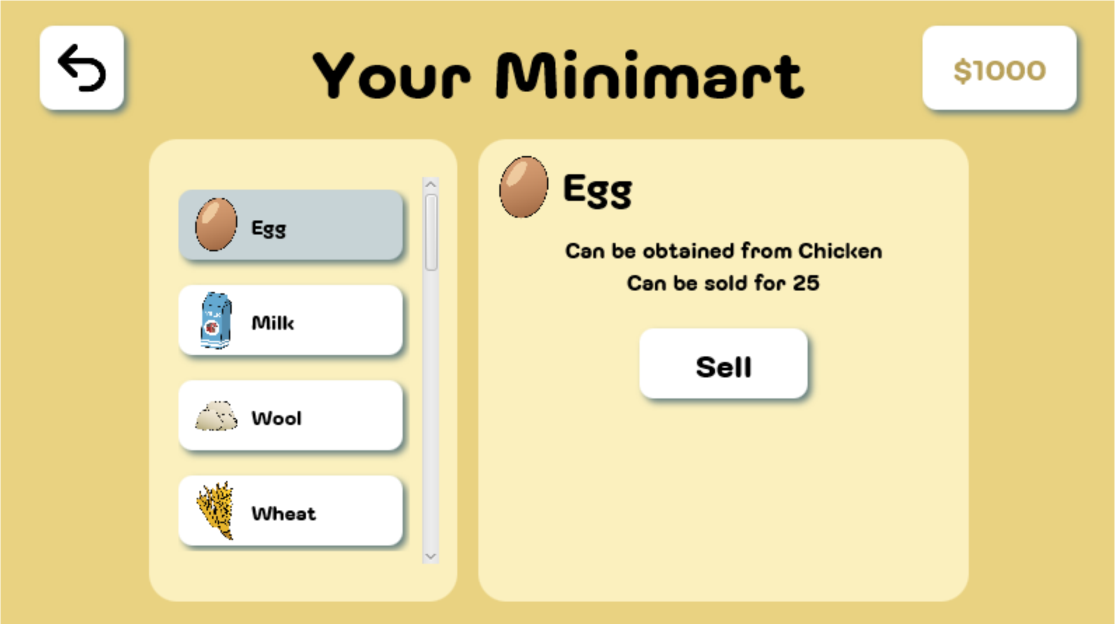
Each of them needs to be unlocked in the shop before they can plant except wheat seed that will be unlocked at the start of the game.



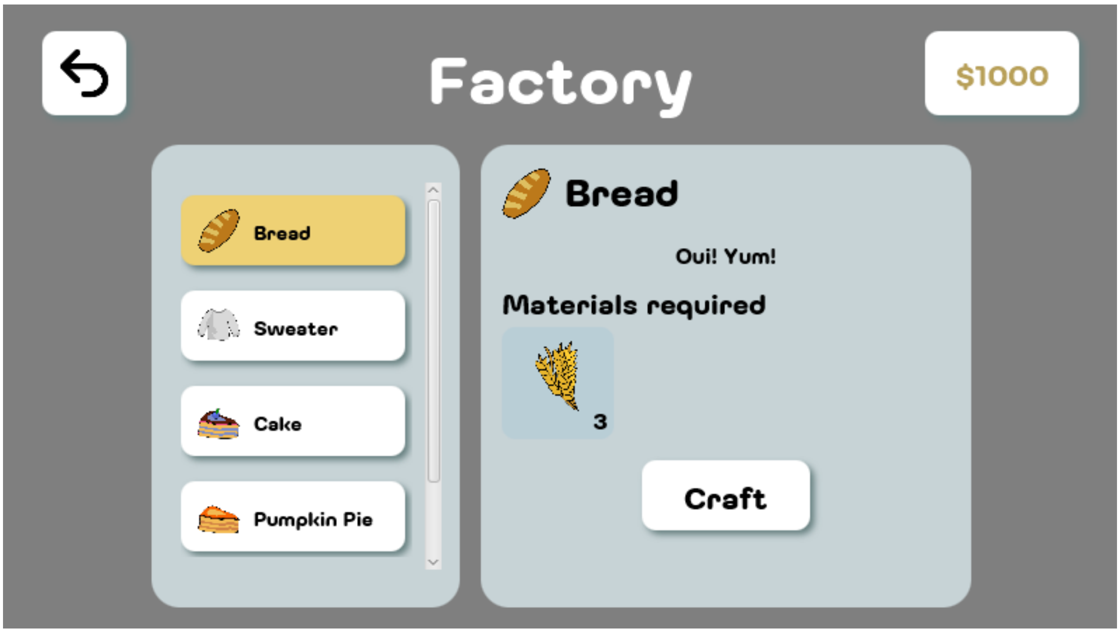
Players can plant all the seeds in plot and water them till they grow, and they can be harvested for products.



There are also 3 animals to raise including cow, chicken, and sheep which can have maximum of 4. Like seeds, the player can feed animals for products as well.



Can be obtained from Chicken
Can be sold for 25

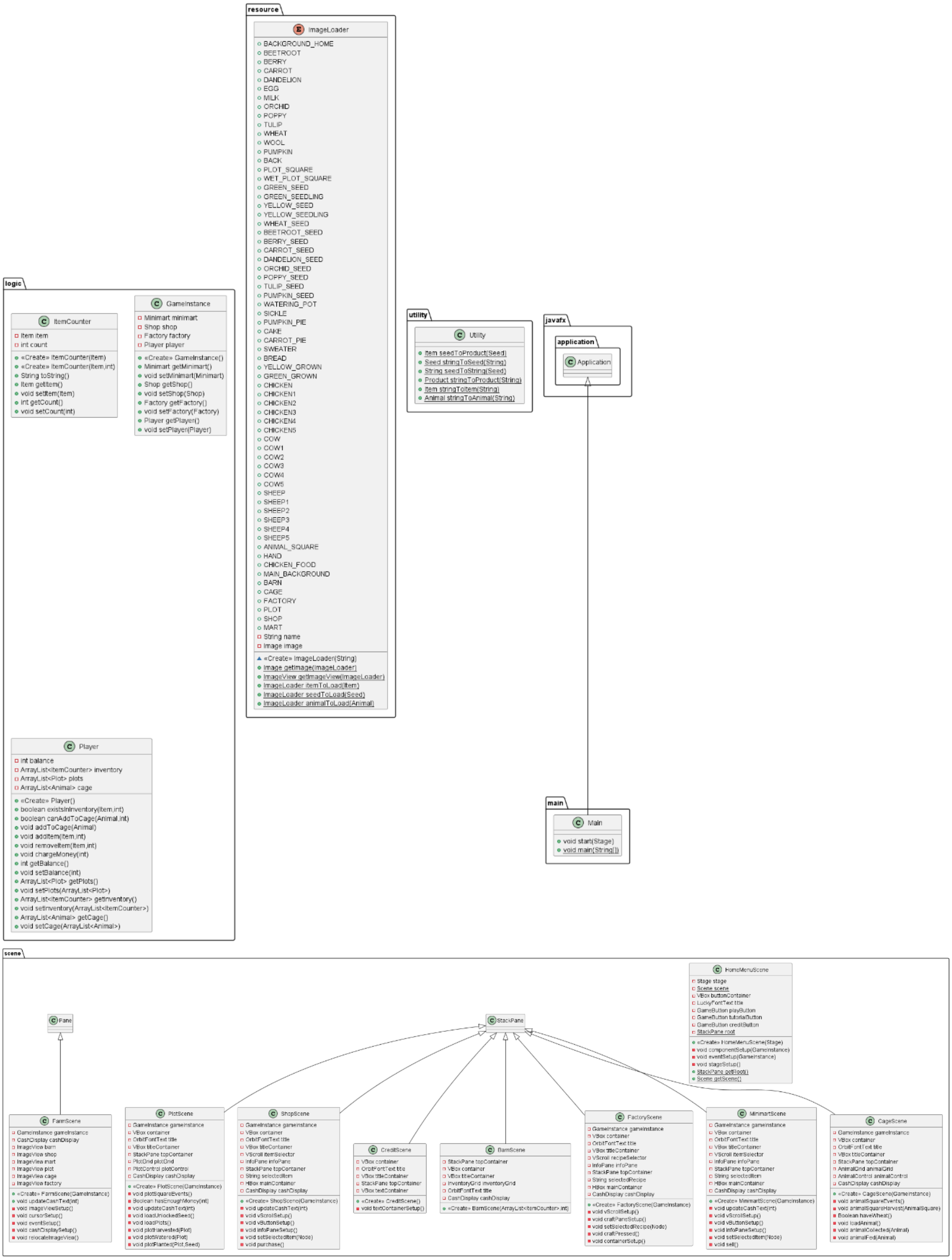


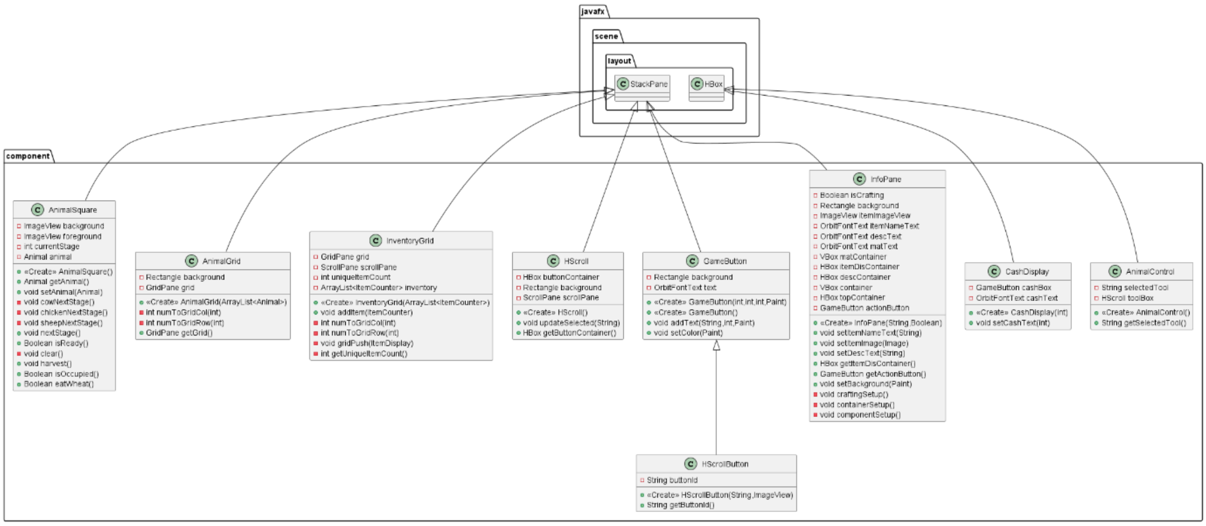
Oui! Yum!

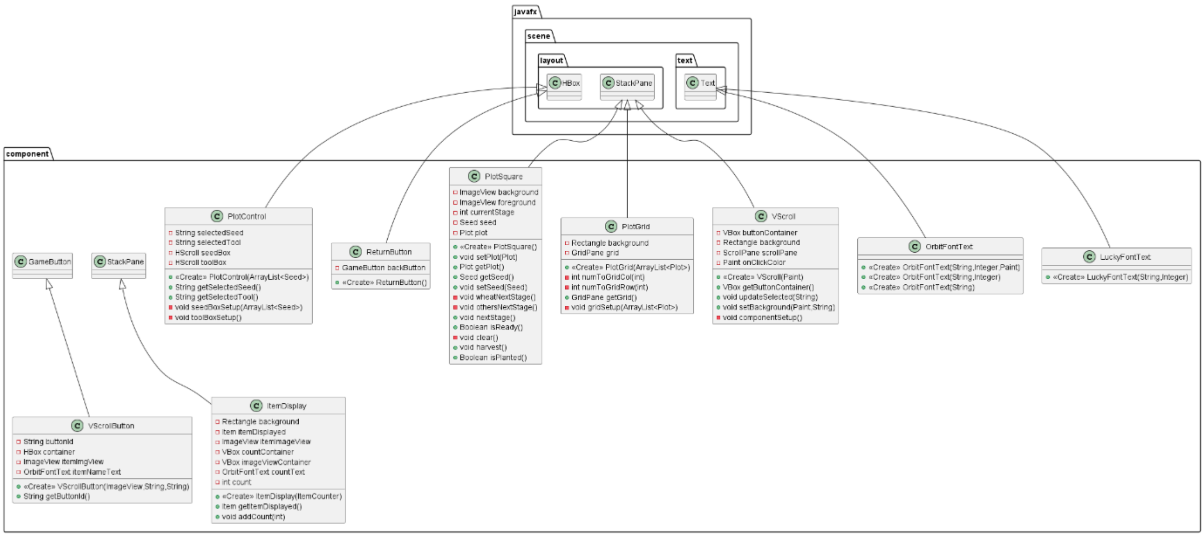
Materials required

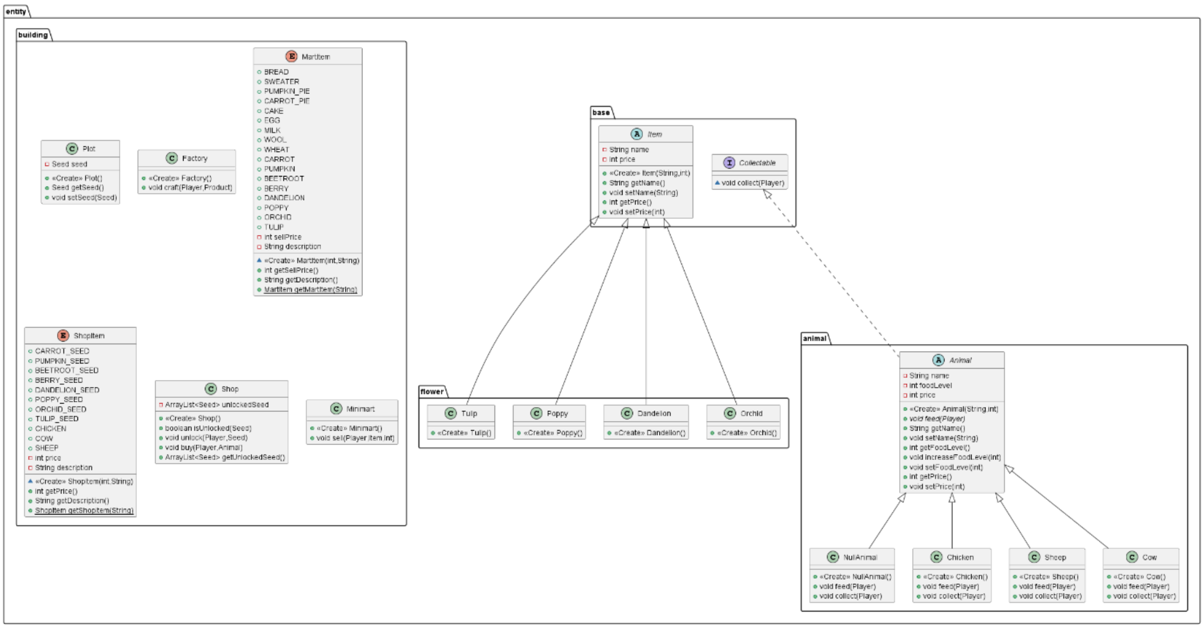
These products can be sold at minimart or can be used as raw materials in a factory to craft items such as bread, sweaters, cake, etc. for better selling price and become millionaire!

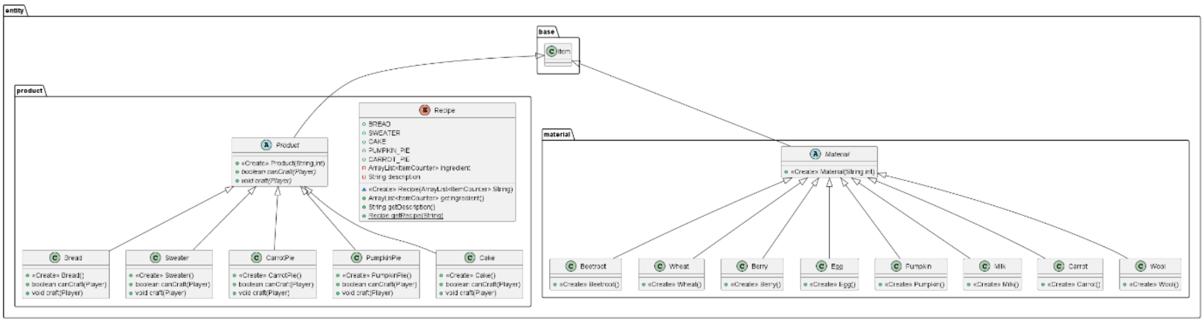
Class Diagram

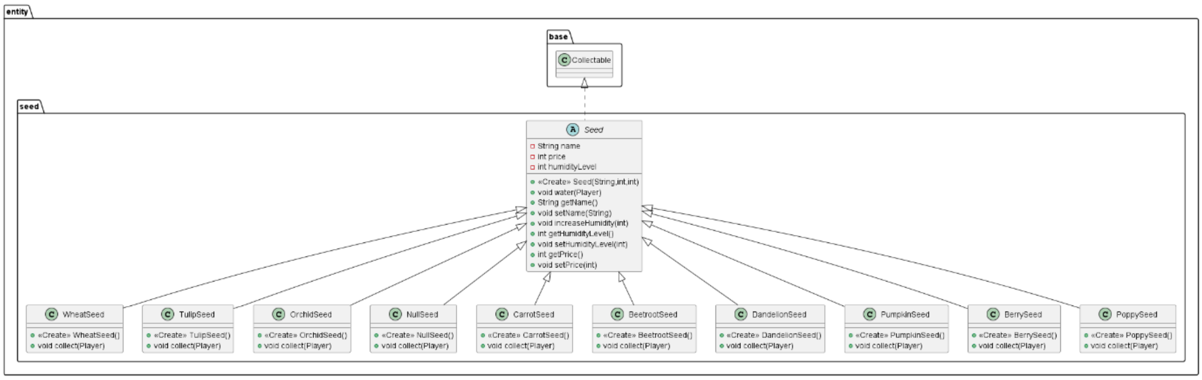



javafx
scene
layout
StackPane
HBox
component
AnimalSquare
ImageView background
ImageView foreground
int currentStage
Animal animal
«Create» AnimalSquare()
Animal getAnimal()
void setAnimal(Animal)
void cowNextStage()
void chickenNextStage()
void sheepNextStage()
void nextStage()
Boolean isReady()
void clear()
void harvest()
Boolean isOccupied()
Boolean eatWheat()
AnimalGrid
Rectangle background
GridPane grid
«Create» AnimalGrid(ArrayList<Animal>)
int numToGridCol(int)
int numToGridRow(int)
GridPane getGrid()
InventoryGrid
GridPane grid
ScrollPane scrollPane
int uniqueItemCount
ArrayList<ItemCounter> inventory
«Create» InventoryGrid(ArrayList<ItemCounter>)
void addItem(ItemCounter)
int numToGridCol(int)
int numToGridRow(int)
void gridPush(ItemDisplay)
int getUniqueItemCount()
HScroll
HBox buttonContainer
Rectangle background
ScrollPane scrollPane
«Create» HScroll()
void updateSelected(String)
HBox getButtonContainer()
GameButton
Rectangle background
OrbitFontText text
«Create» GameButton(int,int,int,Paint)
«Create» GameButton()
void addText(String,int,Paint)
void setColor(Paint)
HScrollButton
String buttonId
«Create» HScrollButton(String,ImageView)
String getButtonId()
InfoPane
Boolean isCrafting
Rectangle background
ImageView itemImageView
OrbitFontText itemNameText
OrbitFontText descText
OrbitFontText matText
VBox matContainer
HBox itemDisContainer
HBox descContainer
VBox container
HBox topContainer
GameButton actionButton
«Create» InfoPane(String,Boolean)
void setItemNameText(String)
void setItemImage(Image)
void setDescText(String)
HBox getItemDisContainer()
GameButton getActionButton()
void setBackground(Paint)
void craftingSetup()
void containerSetup()
void componentSetup()
CashDisplay
GameButton cashBox
OrbitFontText cashText
«Create» CashDisplay(int)
void setCashText(int)
AnimalControl
String selectedTool
HScroll toolBox
«Create» AnimalControl()
String getSelectedTool()


javafx
scene
layout
HBox
StackPane
text
Text
component
GameButton
StackPane
PlotControl
String selectedSeed
String selectedTool
HScroll seedBox
HScroll toolBox
«Create» PlotControl(ArrayList<Seed>)
String getSelectedSeed()
String getSelectedTool()
void seedBoxSetup(ArrayList<Seed>)
void toolBoxSetup()
ReturnButton
GameButton backButton
«Create» ReturnButton()
PlotSquare
ImageView background
ImageView foreground
int currentStage
Seed seed
Plot plot
«Create» PlotSquare()
void setPlot(Plot)
Plot getPlot()
Seed getSeed()
void setSeed(Seed)
void wheatNextStage()
void othersNextStage()
void nextStage()
Boolean isReady()
void clear()
void harvest()
Boolean isPlanted()
PlotGrid
Rectangle background
GridPane grid
«Create» PlotGrid(ArrayList<Plot>)
int numToGridCol(int)
int numToGridRow(int)
GridPane getGrid()
void gridSetup(ArrayList<Plot>)
VScroll
VBox buttonContainer
Rectangle background
ScrollPane scrollPane
Paint onClickColor
«Create» VScroll(Paint)
VBox getButtonContainer()
void updateSelected(String)
void setBackground(Paint,String)
void componentSetup()
OrbitFontText
«Create» OrbitFontText(String,Integer,Paint)
«Create» OrbitFontText(String,Integer)
«Create» OrbitFontText(String)
LuckyFontText
«Create» LuckyFontText(String,Integer)
VScrollButton
String buttonId
HBox container
ImageView itemImgView
OrbitFontText itemNameText
«Create» VScrollButton(ImageView,String,String)
String getButtonId()
ItemDisplay
Rectangle background
Item itemDisplayed
ImageView itemImageView
VBox countContainer
VBox imageViewContainer
OrbitFontText countText
int count
«Create» ItemDisplay(ItemCounter)
Item getItemDisplayed()
void addCount(int)


entity
building
MartItem
BREAD
SWEATER
PUMPKIN_PIE
CARROT_PIE
CAKE
EGG
MILK
WOOL
WHEAT
CARROT
PUMPKIN
BEETROOT
BERRY
DANDELION
POPPY
ORCHID
TULIP
int sellPrice
String description
«Create» MartItem(int,String)
int getSellPrice()
String getDescription()
MartItem getMartItem(String)
Plot
Seed seed
«Create» Plot()
Seed getSeed()
void setSeed(Seed)
Factory
«Create» Factory()
void craft(Player,Product)
ShopItem
CARROT_SEED
PUMPKIN_SEED
BEETROOT_SEED
BERRY_SEED
DANDELION_SEED
POPPY_SEED
ORCHID_SEED
TULIP_SEED
CHICKEN
COW
SHEEP
int price
String description
«Create» ShopItem(int,String)
int getPrice()
String getDescription()
ShopItem getShopItem(String)
Shop
ArrayList<Seed> unlockedSeed
«Create» Shop()
boolean isUnlocked(Seed)
void unlock(Player,Seed)
void buy(Player,Animal)
ArrayList<Seed> getUnlockedSeed()
Minimart
«Create» Minimart()
void sell(Player,Item,int)
base
Item
String name
int price
«Create» Item(String,int)
String getName()
void setName(String)
int getPrice()
void setPrice(int)
Collectable
void collect(Player)
flower
Tulip
«Create» Tulip()
Poppy
«Create» Poppy()
Dandelion
«Create» Dandelion()
Orchid
«Create» Orchid()
animal
Animal
String name
int foodLevel
int price
«Create» Animal(String,int)
void feed(Player)
String getName()
void setName(String)
int getFoodLevel()
void increaseFoodLevel(int)
void setFoodLevel(int)
int getPrice()
void setPrice(int)
NullAnimal
«Create» NullAnimal()
void feed(Player)
void collect(Player)
Chicken
«Create» Chicken()
void feed(Player)
void collect(Player)
Sheep
«Create» Sheep()
void feed(Player)
void collect(Player)
Cow
«Create» Cow()
void feed(Player)
void collect(Player)


entity
base
Item
product
Product
«Create» Product(String,int)
boolean canCraft(Player)
void craft(Player)
Recipe
BREAD
SWEATER
CAKE
PUMPKIN_PIE
CARROT_PIE
ArrayList<ItemCounter> ingredient
String description
«Create» Recipe(ArrayList<ItemCounter>,String)
ArrayList<ItemCounter> getIngredient()
String getDescription()
Recipe getRecipe(String)
Bread
«Create» Bread()
boolean canCraft(Player)
void craft(Player)
Sweater
«Create» Sweater()
boolean canCraft(Player)
void craft(Player)
CarrotPie
«Create» CarrotPie()
boolean canCraft(Player)
void craft(Player)
PumpkinPie
«Create» PumpkinPie()
boolean canCraft(Player)
void craft(Player)
Cake
«Create» Cake()
boolean canCraft(Player)
void craft(Player)
material
Material
«Create» Material(String,int)
Beetroot
«Create» Beetroot()
Wheat
«Create» Wheat()
Berry
«Create» Berry()
Egg
«Create» Egg()
Pumpkin
«Create» Pumpkin()
Milk
«Create» Milk()
Carrot
«Create» Carrot()
Wool
«Create» Wool()


entity
base
Collectable
seed
Seed
String name
int price
int humidityLevel
«Create» Seed(String,int,int)
void water(Player)
String getName()
void setName(String)
void increaseHumidity(int)
int getHumidityLevel()
void setHumidityLevel(int)
int getPrice()
void setPrice(int)
WheatSeed
«Create» WheatSeed()
void collect(Player)
TulipSeed
«Create» TulipSeed()
void collect(Player)
OrchidSeed
«Create» OrchidSeed()
void collect(Player)
NullSeed
«Create» NullSeed()
void collect(Player)
CarrotSeed
«Create» CarrotSeed()
void collect(Player)
BeetrootSeed
«Create» BeetrootSeed()
void collect(Player)
DandelionSeed
«Create» DandelionSeed()
void collect(Player)
PumpkinSeed
«Create» PumpkinSeed()
void collect(Player)
BerrySeed
«Create» BerrySeed()
void collect(Player)
PoppySeed
«Create» PoppySeed()
void collect(Player)

1. Package logic

1.1. public class GameInstance

Field

Name	Description
- Minimart minimart	Game minimart
- Shop shop	Game shop
- Factory factory	Game factory
- Player player	Game player

Constructor

Name	Description
+ GameInstance()	Create GameInstance with new minimart, shop, factory, and player.

Method

Name	Description
+ Minimart getMinimart()	Return game minimart
+ void setMinimart(Minimart minimart)	Set game minimart
+ Shop getShop()	Return game shop
+ void setShop(Shop shop)	Set game shop
+ Factory getFactory()	Return game factory
+ void setFactory(Factory factory)	Set game factory
+ Player getPlayer()	Return game player
+ void setPlayer(Player player)	Set game player

1.2. public class ItemCounter

Field

Name	Description
- Item item	Item
- int count	Quantity of item

Constructor

Name	Description
+ ItemCounter(Item item)	Initiate ItemCounter with the specified item. Set count to 1.
+ ItemCounter(Item item, int count)	Initiate ItemCounter with the specified item and quantity.

Method

Name	Description
+ String toString()	Return String in format “item name x count”
+ Item getItem()	Return item
+ void setItem(Item item)	Set item to specified item
+ int getCount()	Return item quantity
+ void setCount(int count)	Set count to specified information. If count is less than 0, set count to 0.

1.3. public class Player

Field

Name	Description
- int balance	Player’s balance
- ArrayList<ItemCounter> inventory	Player’s inventory
- ArrayList<Plot> plots	Player’s plots
- ArrayList<Animal> cage	Player’s animal cage

Constructor

Name	Description
+ Player()	Initiate Player with 1000 balance, empty inventory, 12 empty animal cage, and 14 empty plots.

Method

Name	Description
+ boolean existInventory(Item item, int amount)	Return whether player’s inventory has the item with specified amount.
+ boolean canAddToCage(Animal animal, int amount)	Return whether player’s animal cage has the animal with specified amount.
+ void addItem(Item newItem, int amount)	Add Item to player’s inventory. If newItem already exists in inventory add quantity of item to specified amount. If not exists add newItem with specified amount.
+ void removeItem(Item toRemove, int amount)	Remove Item from player’s inventory with specified amount. If the amount is less than 0 do nothing. When removed item quantity is less than 0 remove that item from inventory.
+ void chargeMoney(int amount)	Deduct player’s balance by specified amount given.
+ int getBalance()	Return player’s balance
+ void setBalance(int balance)	Set the player’s balance with specified amount. If amount is less than 0, set to 0.
+ ArrayList<Plot> getPlots()	Return player’s plots
+ void setPlot(ArrayList<Plot> plots)	Set the player’s plots
+ ArrayList<ItemCounter> getInventory()	Return player’s inventory
+ void setInventory(ArrayList< ItemCounter > inventory)	Set the player’s inventory
+ ArrayList<Animal> getCage()	Return player’s animal cage
+ void setCage(ArrayList<Animal> cage)	Set the player’s animal cage

2. Package entity

2.1. Package entity.animal

2.1.1. public abstract class Animal implements Collectable

Field

Name	Description
- String name	Name of animal
- int foodLevel	Food level of animal
- int price	Price of animal

Constructor

Name	Description
+ Animal(String name, int price)	Initiate Animal with the specified information. Set foodLevel to 0.

Method

Name	Description
+ abstract void feed(Player player)	Feed animal different by each variety of animal
+ void increaseFoodLevel(int amount)	Increase the food level by given amount. If end, food level is more than 100 set to 100.
+ void setFoodLevel(int foodLevel)	Set the food level by specified amount. If the food level is less than 0, set to 0. If food level is more than 100, set to 100.
+ int getFoodLevel()	Return animal’s food level
+ void setName(String name)	Set animal’s name
+ String getName()	Return animal’s name
+ void setPrice(int price)	Set animal’s price
+ int getPrice()	Return animal’s price

2.1.2. public class Chicken extends Animal

Constructor

Name	Description
+ Chicken()	Initiate with super constructor. Set name to “Chicken” and price to 1000.

Method

Name	Description
+ void feed(Player player)	Increase food level by 25
+ void collect(Player player)	If food level equal to 100, add an egg to player’s inventory and set food level to 0.

2.1.3. public class Cow extends Animal

Constructor

Name	Description
+ Cow()	Initiate with super constructor. Set name to “Cow” and price to 3000.

Method

Name	Description
+ void feed(Player player)	If player’s inventory has wheat, increase food level by 25 and remove 1 wheat from player’s inventory
+ void collect(Player player)	If food level equal to 100, add a milk to player’s inventory and set food level to 0.

2.1.4. public class Sheep extends Animal

Constructor

Name	Description
+ Sheep()	Initiate with super constructor. Set name to “Sheep” and price to 3000.

Method

Name	Description
+ void feed(Player player)	If player’s inventory has wheat, increase food level by 25 and remove 1 wheat from player’s inventory
+ void collect(Player player)	If food level equal to 100, add a wool to player’s inventory and set food level to 0.

2.1.5. public class NullAnimal extends Animal

Constructor

Name	Description
+ NullAnimal()	Initiate with super constructor. Set name to “” and price to 0.

Method

Name	Description
+ void feed(Player player)	Do nothing
+ void collect(Player player)	Do nothing

2.2. Package entity.base

2.2.1. Interface Collectable

Method

Name	Description
+ void collect(Player player)	Implement collect method unique by class.

2.2.2. public abstract class Item

Field

Name	Description
- String name	Name of item
- int price	Price of item

Constructor

Name	Description
+ Item(String name, int price)	Initiate item with the specified information.

Method

Name	Description
+ void setName(String name)	Set name of item
+ String getName()	Return name of item
+ void setPrice(int price)	Set price of item
+ int getPrice()	Return price of item

2.3. Package entity.building

2.3.1. public class Factory

Constructor

Name	Description
+ Factory()	Initiate factory

Method

Name	Description
+ void craft(Player player, Product product)	If player can craft product, craft the product by using method craft of product. If not, print message “Can’t craft + product name”

2.3.2. public enum MartItem

Field

Name	Description
- int sellPrice	Item sell price
- String description	Item description

Constructor

Name	Description
+ MartItem(int price, String description)	Initiate MartItem with the specified sell price and description

Method

Name	Description
+ int getSellPrice()	Return sellprice
+ String getDescription()	Return description
+ MartItem getMartItem(String s)	Return Item object with name s

2.3.3. public class Minimart

Constructor

Name	Description
+ Minimart()	Initiate minimart

Method

Name	Description
+ void sell(Player player, Item item, int amount)	If item exist in player’s inventory, remove item with specified amount and increase player’s balance by the price of item. If not, print out message “item name + doesn’t exists”

2.3.4. public class Plot

Field

Name	Description
- Seed seed	Plot’s seed

Constructor

Name	Description
+ Plot()	Initiate plot with null seed

Method

Name	Description
+ void setSeed(Seed seed)	Set plot’s seed with specified seed
+ Seed getSeed()	Return plot’s seed

2.3.5. public class Shop

Field

Name	Description
- ArrayList<Seed> unlockedSeed	List of seed that has been unlock

Constructor

Name	Description
+ Shop()	Initiate shop and add wheat seed to unlocked seed

Method

Name	Description
+ boolean isUnlocked(Seed seed)	Return true if seed is already unlocked, else return false
+ void unlock(Player player, Seed seed)	If player’s balance is more than seed price and seed is not already unlocked, unlock seed and charge player. If not, print out message “Can’t unlock + seed name)
+ void buy(Player player, Animal animal)	If player’s balance is more than animal price and this kind of animal is not exceeded 4, add animal to player’s cage and charge player. If not, print out message “Can’t buy + animal name)
+ ArrayList<Seed> getUnlockedSeed()	Return list of unlocked seed

2.3.6. public enum ShopItem

Field

Name	Description
- int price	Item price
- String description	Item description

Constructor

Name	Description
+ ShopItem(int price, String description)	Initiate ShopItem with the specified price and description

Method

Name	Description
+ int getPrice()	Return price
+ String getDescription()	Return description
+ ShopItem getShopItem(String s)	Return Item object with name s

2.4. Package entity.flower

2.4.1. public class Dandelion extends Item

Constructor

Name	Description
+ Dandelion()	Create with super constructor. Set name to “Dandelion” and price to 200.

2.4.2. public class Orchid extends Item

Constructor

Name	Description
+ Orchid()	Create with super constructor. Set name to “Orchid” and price to 250.

2.4.3. public class Poppy extends Item

Constructor

Name	Description
+ Poppy()	Create with super constructor. Set name to “Poppy” and price to 100.

2.4.4. public class Tulip extends Item

Constructor

Name	Description
+ Tulip()	Create with super constructor. Set name to “Tulip” and price to 100.

2.5. Package entity.material

2.5.1. public abstract class Material extends Item

Constructor

Name	Description
+ Material(String name, int price)	Create with super constructor. Set field to specified information

2.5.2. public class Beetroot extends Material

Constructor

Name	Description
+ Beetroot()	Create using super constructor. Set name to “Beetroot” and price to 50.

2.5.3. public class Berry extends Material

Constructor

Name	Description
+ Berry()	Create using super constructor. Set name to “Berry” and price to 50.

2.5.4. public class Carrot extends Material

Constructor

Name	Description
+ Carrot()	Create using super constructor. Set name to “Carrot” and price to 25.

2.5.5. public class Egg extends Material

Constructor

Name	Description
+ Egg()	Create using super constructor. Set name to “Egg” and price to 25.

2.5.6. public class Milk extends Material

Constructor

Name	Description
+ Milk()	Create using super constructor. Set name to “Milk” and price to 100.

2.5.7. public class Pumpkin extends Material

Constructor

Name	Description
+ Pumpkin()	Create using super constructor. Set name to “Pumpkin” and price to 25.

2.5.8. public class Wheat extends Material

Constructor

Name	Description
+ Wheat()	Create using super constructor. Set name to “Wheat” and price to 25.

2.5.9. public class Wool extends Material

Constructor

Name	Description
+ Wool()	Create using super constructor. Set name to “Wool” and price to 100.

2.6. Package entity.product

2.6.1. public abstract class Product extends Item

Constructor

Name	Description
+ Product(String name, int price)	Create with super constructor. Set field to specified information

Method

Name	Description
+ abstract boolean canCraft(Player player)	Check whether player has enough raw material to craft product
+ abstract void craft(Player player)	Craft product using player’s raw material

2.6.2. public class Bread extends Product

Constructor

Name	Description
+ Bread()	Create using super constructor. Set name to “Bread” and price to 100.

Method

Name	Description
+ boolean canCraft(Player player)	Return true if player’s inventory exists 3 wheats

+ void craft(Player player)	Remove 3 wheats from player's inventory and add bread to it

2.6.3. public class Cake extends Product

Constructor

Name	Description
+ Cake()	Create using super constructor. Set name to "Cake" and price to 1200.

Method

Name	Description
+ boolean canCraft(Player player)	Return true if player's inventory exists of 5 berries, 5 milks, and 5 eggs.
+ void craft(Player player)	Remove 5 berries, 5 milks, and 5 eggs from player's inventory and add 1 cake to player's inventory.

2.6.4. public class CarrotPie extends Product

Constructor

Name	Description
+ CarrotPie()	Create using super constructor. Set name to "Carrot Pie" and price to 1000.

Method

Name	Description
+ boolean canCraft(Player player)	Return true if player's inventory exists of 5 carrots, 5 milks, and 5 eggs.
+ void craft(Player player)	Remove 5 carrots, 5 milks, and 5 eggs from player's inventory and add 1 carrot pie to player's inventory.

2.6.5. public class PumpkinPie extends Product

Constructor

Name	Description
+ PumpkinPie()	Create using super constructor. Set name to “Pumpkin Pie” and price to 1000.

Method

Name	Description
+ boolean canCraft(Player player)	Return true if player’s inventory exists of 5 pumpkins, 5 milks, and 5 eggs.
+ void craft(Player player)	Remove 5 pumpkins, 5 milks, and 5 eggs from player’s inventory and add 1 pumpkin pie to player’s inventory.

2.6.6. public class Sweater extends Product

Constructor

Name	Description
+ Sweater()	Create using super constructor. Set name to “Sweater” and price to 600.

Method

Name	Description
+ boolean canCraft(Player player)	Return true if player’s inventory exists of 5 wools.
+ void craft(Player player)	Remove 5 wools from player’s inventory and add 1 sweater to player’s inventory.

2.6.7. public enum Recipe

Field

Name	Description
- ArrayList<ItemCounter> ingredient	Ingredient
- String description	Description

Constructor

Name	Description
+ Recipe(ArrayList<ItemCounter> ingredient, String description)	Initiate MartItem with the specified ingredient and description

Method

Name	Description
+ ArrayList<ItemCounter> getIngredient()	Return ingredient
+ String getDescription()	Return description
+ Recipe getRecipe(String s)	Return Recipe with item name s

2.7. Package entity.seed

2.7.1. public abstract class Seed implements Collectable

Field

Name	Description
- String name	Name of seed
- int price	Price of seed
- int humidityLevel	Humidity level of seed

Constructor

Name	Description
+ Seed(String name, int price, int humidityLevel)	Initiate seed with the specified information.

Method

Name	Description
+ void water(Player player)	If player’s balance is more than 5 and seed humidity not equal to 100, increase humidity and charge player $5. If not, print out the message “Can’t water plant. Not enough money.”
+ void increaseHumidity(int amount)	Increase the humidity level by given amount. If after increase, humidity level exceeded 100, set to 100.
+ void setHumidityLevel(int humidity)	Set humidity of seed. If humidity is less than 0, set humidity level to 0. If humidity level is more than 100, set humidity level to 100.
+ int getHumidityLevel()	Return humidity level of seed
+ void setName(String name)	Set name of seed
+ String getName()	Return name of seed
+ void setPrice(int price)	Set price of seed. If price is less than 0, set to 0.
+ int getPrice()	Return price of seed

2.7.2. public class BeetrootSeed extends Seed

Constructor

Name	Description
+ BeetrootSeed()	Initiate with super constructor. Set name to “Beetroot Seed”, price to 500, and humidity level to 0.

Method

Name	Description
+ void collect(Player player)	If humidity level equal to 100, add 1 beetroot to player’s inventory. If not

	print out message “Can’t collect + seed name”

2.7.3. public class BerrySeed extends Seed

Constructor

Name	Description
+ BerrySeed()	Initiate with super constructor. Set name to “Berry Seed”, price to 500, and humidity level to 0.

Method

Name	Description
+ void collect(Player player)	If humidity level equal to 100, add 1 berry to player’s inventory. If not print out message “Can’t collect + seed name”

2.7.4. public class CarrotSeed extends Seed

Constructor

Name	Description
+ CarrotSeed()	Initiate with super constructor. Set name to “Carrot Seed”, price to 100, and humidity level to 0.

Method

Name	Description
+ void collect(Player player)	If humidity level equal to 100, add 1 carrot to player’s inventory. If not print out message “Can’t collect + seed name”

2.7.5. public class PumpkinSeed extends Seed

Constructor

Name	Description
+ PumpkinSeed()	Initiate with super constructor. Set name to “Pumpkin Seed”, price to 100, and humidity level to 0.

Method

Name	Description
+ void collect(Player player)	If humidity level equal to 100, add 1 pumpkin to player’s inventory. If not print out message “Can’t collect + seed name”

2.7.6. public class WheatSeed extends Seed

Constructor

Name	Description
+ WheatSeed()	Initiate with super constructor. Set name to “Wheat Seed”, price to 0, and humidity level to 0.

Method

Name	Description
+ void collect(Player player)	If humidity level equal to 100, add 1 wheat to player’s inventory. If not print out message “Can’t collect + seed name”

2.7.7. public class DandelionSeed extends Seed

Constructor

Name	Description
+ DandelionSeed()	Initiate with super constructor. Set name to “Dandelion Seed”, price to 2000, and humidity level to 0.

Method

Name	Description
+ void collect(Player player)	If humidity level equal to 100, add 1 dandelion to player’s inventory. If not print out message “Can’t collect + seed name”

2.7.8. public class OrchidSeed extends Seed

Constructor

Name	Description
+ OrchidSeed()	Initiate with super constructor. Set name to “Orchid Seed”, price to 2500, and humidity level to 0.

Method

Name	Description
+ void collect(Player player)	If humidity level equal to 100, add 1 orchid to player’s inventory. If not print out message “Can’t collect + seed name”

2.7.9. public class PoppySeed extends Seed

Constructor

Name	Description
+ PoppySeed()	Initiate with super constructor. Set name to “Poppy Seed”, price to 1000, and humidity level to 0.

Method

Name	Description
+ void collect(Player player)	If humidity level equal to 100, add 1 poppy to player’s inventory. If not print out message “Can’t collect + seed name”

2.7.10. public class TulipSeed extends Seed

Constructor

Name	Description
+ TulipSeed()	Initiate with super constructor. Set name to “Tulip Seed”, price to 1000, and humidity level to 0.

Method

Name	Description
+ void collect(Player player)	If humidity level equal to 100, add 1 tulip to player’s inventory. If not print out message “Can’t collect + seed name”

2.7.11. public class NullSeed extends Seed

Constructor

Name	Description
+ NullSeed()	Initiate with super constructor. Set name to “”, price to 0, and humidity level to 0.

Method

Name	Description
+ void collect(Player player)	Do nothing

3. Package component

3.1. public class AnimalControl extends HBox

Field

Name	Description
- String selectedTool	The selected tool
- HScroll toolBox	Toolbox

Constructor

Name	Description
+ AnimalControl()	Initiate AnimalControl box

Method

Name	Description
+ String getSelectedTool()	Return selectedTool

3.2. public class AnimalGrid extends StackPane

Field

Name	Description
- Rectangle background	Grid background
- GridPane grid	Grid to contain AnimalSquare

Constructor

Name	Description
+ AnimalGrid()	Initiate AnimalGrid box

Method

Name	Description
- int numToGridCol(int x)	Return the column number of cell #x
- int numToGridRow(int x)	Return the row number of cell #x
+ GridPane getGrid()	Return grid
+ int getCount()	Return the amount of element in grid

3.3. public class AnimalSquare extends StackPane

Field

Name	Description
- ImageView background	Background of the square
- ImageView foreground	Foreground of the square
- int currentStage	Current animal stage
- Animal animal	Animal contained

Constructor

Name	Description
+ AnimalSquare()	Initiate AnimalSquare by setting currentStage to 0 and background.

Method

Name	Description
+ Animal getAnimal()	Return animal
+ void setAnimal(Animal animal)	Set animal
- void cowNextStage()	Increment the stage of cow by 1 and change foreground corresponding to the new stage
- void chickenNextStage()	Increment the stage of chicken by 1and change foreground corresponding to the new stage
- void sheepNextStage()	Increment the stage of sheep by 1 and change foreground corresponding to the new stage
+ void nextStage()	Increment the stage of animal by 1

+ Boolean isReady()	Return whether animal is ready to be harvested (currentStage is 4)
- void clear()	Set the stage of animal to 0 and change foreground corresponding to the new stage
+ void harvest()	If isReady is true, then clear the square
+ void isOccupied()	Return whether the square is occupied by certain animal or not

3.4. public class CashDisplay extends HBox

Field

Name	Description
- GameButton cashBox	Background
- OrbitFontText cashText	Text

Constructor

Name	Description
+ CashDisplay(int cash)	Initiate CashDisplay box with the specified amount of cash to display

Method

Name	Description
+ void setCashText(int cash)	Set cashText with the specified amount

3.5. public class GameButton extends StackPane

Field

Name	Description
- Rectangle background	Background
- OrbitFontText text	Text

Constructor

Name	Description
+ GameButton(int w, int h, int corner_r, Paint paint)	Initiate the GameButton by setting background width, height, corner radius, and background color.
+ GameButton()	Initiate the GameButton by setting background width, height, corner radius, and background color to be 60, 60, 20, and White respectively.

Method

Name	Description
+ void addText(String s, int size, Paint paint)	Set the text value, size, and color with the specified values
+ void setColor(Paint paint)	Set the background color with the specified color

3.6. public class HScroll extends StackPane

Field

Name	Description
- HBox buttonContainer	Container of all buttons
- Rectangle background	Background
- ScrollPane scrollPane	ScrollPane

Constructor

Name	Description
+ HScroll()	Initiate HScroll by setting the size, color, and children of all fields.

Method

Name	Description
+ void updateSelected(String selected)	Set the button inside container state to selected.
+ HBox getButtonContainer()	Return buttonContainer

3.7. public class HScrollButton extends GameButton

Field

Name	Description
- String buttonId	Button identifier

Constructor

Name	Description
+ HScrollButton(String buttonId, ImageView imgv)	Initiate new HScrollButton with the specified buttonId and ImageView

Method

Name	Description
+ String getButtonId()	Return buttonId

3.8. public class InfoPane extends StackPane

Field

Name	Description
- Boolean isCrafting	If the InfoPane object is created inside FactoryScene, the value must be set to true. Otherwise, set it to false.
- Rectangle background	Background
- ImageView itemImageView	ImageView with an image of item
- OrbitFontText itemNameText	Text with the name of an item
- OrbitFontText descText	Text with the item description
- OrbitFontText matText	Text set to “Materials required”
- VBox matContainer	Container of matText
- HBox itemDisContainer	Container of ItemDisplays

- HBox descContainer	Container of descText
- VBox container	Super container that contains all subcontainers
- HBox topContainer	Contains itemImageView and itemNameText
- GameButton actionButton	Button with its text set to certain string

Constructor

Name	Description
+ InfoPane(String action, Boolean isCrafting)	Initiate InfoPane

Method

Name	Description
+ void setItemNameText(String s)	Return String in format “item name x count”
+ void setItemImage(Image i)	Return item
+ void setDescText(String s)	Set item to specified item
+ HBox getItemDisContainer()	Return itemDisContainer
+ GameButton getActionButton()	Return actionButton
+ void setBackground(Paint paint)	Set the background color with the specified color
- void craftingSetup()	Additional setup in case isCrafting is true
- void containerSetup()	Setup all containers
- void componentSetup()	Setup all field components

3.9. public class InventoryGrid extends StackPane

Field

Name	Description
- GridPane grid	Grid that contains ItemDisplays
- ScrollPane scrollPane	ScrollPane
- int uniqueItemCount	The amount of ItemDisplay in grid
- ArrayList<ItemCounter> inventory	Player's inventory

Constructor

Name	Description
InventoryGrid(ArrayList<ItemCounter> inv)	Initialize new InventoryGrid with the specified inventory

Method

Name	Description
+ void addItem(ItemCounter itemCounter)	Converts ItemCounter to ItemDisplay then add it to grid
- int numToGridCol(int x)	Return the column number of cell #x
- int numToGridRow(int x)	Return the row number of cell #x
- void gridPush(ItemDisplay itemDisplay)	Add the itemDisplay to grid
- int getUniqueItemCount()	Return the amount of element inside grid

3.10. public class ItemDisplay extends StackPane

Field

Name	Description
- Rectangle background	Background
- Item itemDisplayed	Item to be displayed
- ImageView itemImageView	ImageView with an image of itemDisplayed
- VBox countContainer	Container of countText
- VBox imageViewContainer	Container of itemImageView
- OrbitFontText countText	Text to display count
- int count	The amount of itemDisplayed

Constructor

Name	Description
+ ItemDisplay(ItemCounter itemCounter)	Initialize ItemDisplay with the specified item and its amount from itemCounter

Method

Name	Description
+ Item getItemDisplayed()	Return itemDisplayed
+ void addCount(int a)	Increase count by a and update countText

3.11. public class LuckyFontText extends Text

Constructor

Name	Description
+ LuckyFontText(String t, int s)	Initiate LuckyFontText with its text value and size set to t and s respectively.

3.12. public class OrbitFontText extends Text

Constructor

Name	Description
+ OrbitFontText(String t, int s, Paint paint)	Initiate OrbitFontText with its text value, color, and size set to t, paint, and s respectively.
+ OrbitFontText(String t, int s)	Initiate OrbitFontText with its text value, color, and size set to t, BLACK, and s respectively.
+ OrbitFontText(String t)	Initiate OrbitFontText with its text value, color, and size set to t, BLACK, and 16 respectively.

3.13. public class PlotControl extends HBox

Field

Name	Description
- String selectedSeed	The selected seed
- String selectedTool	The selected tool
- HScroll seedBox	Seed box
- HScroll toolBox	Toolbox

Constructor

Name	Description
+ PlotControl(ArrayList<Seed> unlockedSeed)	Initiate PlotControl with the available seeds in seedBox set to the specified value

Method

Name	Description
+ String getSelectedSeed()	Return selectedSeed
+ String getSelectedTool()	Return selectedTool
- void seedBoxSetup(ArrayList<Seed> unlockedSeed)	Setup seedBox with the available seeds in seedBox set to the specified value
- void toolBoxSetup()	Setup toolBox

3.14. public class PlotGrid extends StackPane

Field

Name	Description
- Rectangle background	Background
- GridPane grid	Grid that contains PlotSquare

Constructor

Name	Description
+ PlotGrid(ArrayList<Plot> plots)	Initiate PlotGrid by converting each element in plots to PlotSquare then add them to grid

Method

Name	Description
- int numToGridCol(int x)	Return the column number of cell #x
- int numToGridRow(int x)	Return the row number of cell #x
+ GridPane getGrid()	Return grid
- void gridSetup(ArrayList<Plot> plots)	Setup grid by converting each element in plots to PlotSquare then add them to grid

3.15. public class PlotSquare extends HBox

Field

Name	Description
- ImageView background	Background
- ImageView foreground	Foreground
- int currentStage	Current stage of seed
- Plot plot	Plot
- Seed seed	Seed

Constructor

Name	Description
+ PlotSquare()	Initiate PlotSquare

Method

Name	Description
+ void setPlot(Plot plot)	Set plot with the specified value
+ Plot getPlot()	Return plot
+ Seed getSeed()	Return seed
+ void setSeed(Seed seed)	Set seed with the specified value then set the foreground image corresponding to seed type

- void wheatNextStage()	Increment the stage of wheat by 1 then change foreground and background corresponding to the new stage
- void othersNextStage()	Increment the stage of any seed types (except Wheat) by 1 then change foreground and background corresponding to the new stage
+ void nextStage()	Increment the stage of seed by 1
+ Boolean isReady()	Return whether if the seed is ready to be harvested. (currentStage is 4)
- void clear()	Set the stage of seed to 0 then change foreground and background corresponding to the new stage
+ void harvest()	If isReady is true, then clear the square
+ Boolean isPlanted()	Return whether there is a seed planted on the square or not

3.16. public class ReturnButton extends HBox

Field

Name	Description
- GameButton backButton	Button

Constructor

Name	Description
+ ReturnButton()	Initiate ReturnButton by initating backButton, add ImageView with the back icon to it, and set its event handler.

3.17. public class VScroll extends StackPane

Field

Name	Description
- VBox buttonContainer	Container of all buttons
- Rectangle background	Background
- ScrollPane scrollPane	ScrollPane
- Paint onClickColor	Set the button color to the specified color when clicked

Constructor

Name	Description
+ VScroll(Paint clickedColor)	Initiate VScroll with the specified onClickColor

Method

Name	Description
+ VBox getButtonContainer()	Return buttonContainer
+ void updateSelected(String selected)	Change the button color of a button if its buttonId is equal to the specified value
+ void setBackground(Paint paint, String s)	Set the background color of ScrollPane and background color to s and paint respectively. Note s must be a HEX color code
- void componentSetup()	Setup all components

3.18. public class VScrollButton extends StackPane

Field

Name	Description
- String buttonId	Button identifier
- HBox container	Container of itemImgView and itemNameText
- ImageView itemImgView	ImageView with its Image set to certain item
- OrbitFontText itemNameText	Text with its text value set to certain item’s name

Constructor

Name	Description
+ VScrollButton(ImageView imageView, String text, String buttonId)	Initiate VScrollButton with the specified values

Method

Name	Description
+ String getButtonId()	Return buttonId

4. Package scene

4.1. public class BarnScene extends HBox

Field

Name	Description
- StackPane topContainer	Contains cashDisplay and backButton
- VBox container	Super container that contains all sub containers
- VBox titleContainer	Container of title
- InventoryGrid inventoryGrid	InventoryGrid
- OrbitFontText title	Title of the scene that should be set to “Barn”
- CashDisplay cashDisplay	CashDisplay

Constructor

Name	Description
+ BarnScene(ArrayList<ItemCounter> inventory, int balance)	Initiate BarnScene by converting each element in inventory to ItemDisplay then add them to inventoryGrid. Next, set the CashDisplay value with the specified balance. Finally, initiate and setup all component fields.

4.2. public class CageScene extends StackPane

Field

Name	Description
- GameInstance gameInstance	GameInstance
- VBox container	Super container that contains all sub containers
- OrbitFontText title	Title of the scene that should be set to “Cage”
- VBox titleContainer	Container of title
- StackPane topContainer	Container of cashDisplay and backButton
- AnimalGrid animalGrid	AnimalGrid
- AnimalControl animalControl	AnimalControl
- CashDisplay cashDisplay	CashDisplay
- HScroll toolBox	ToolBox

Constructor

Name	Description
+ CageScene(GameInstance gameInstance)	Initiate CageScene by converting each element in player’s animal to AnimalSquare then add them to animalGrid. Next, set the CashDisplay value with player’s balance. Finally, initiate and setup all component fields.

Method

Name	Description
- void animalSquareEvents()	Set up event handler of all AnimalSquare

- void animalSquareHarvest(AnimalSquare i)	Set up event handler of all AnimalSquare when attempted harvest
- Boolean haveWheat()	Return true if player has Wheat in his/her inventory. Otherwise, return false.
- void loadAnimal()	Initialize AnimalGrid
- void animalCollected(Animal animal)	Animal harvested handler
- void animalFed(Animal animal)	Animal fed handler

4.3. public class CreditScene extends StackPane

Field

Name	Description
- VBox container	Super container that contains all sub containers
- OrbitFontText title	Title of the scene that should be set to “Credits”
- VBox titleContainer	Container of title
- StackPane topContainer	Contains backButton
- VBox textContainer	Contains credit text

Constructor

Name	Description
+ CreditScene()	Initiate CreditScene by setting up all component fields.

Method

Name	Description
- void textContainerSetup()	Setup textContainer

4.4. public class FactoryScene extends StackPane

Field

Name	Description
- GameInstance gameInstance	GameInstance
- VBox container	Super container that contains all sub containers
- OrbitFontText title	Title of the scene that should be set to “Factory”
- VBox titleContainer	Contains title
- VScroll recipeSelector	VScroll that contains all craftable item
- InfoPane infoPane	InfoPane which displays crafting recipe and description of the selected craftable item
- StackPane topContainer	Contains backButton and cashDisplay
- String selectedRecipe	String with the name of the selected craftable item
- HBox mainContainer	Contains recipeSelector and infoPane
- CashDisplay cashDisplay	CashDisplay

Constructor

Name	Description
+ FactoryScene(GameInstance gameInstance)	Initiate FactoryScene by adding all recipes from Recipe enum class. Next, set the CashDisplay value with player’s balance. Finally, initiate and setup all component fields.

Method

Name	Description
- void vScrollSetup()	Setup of recipeSelector
- void craftPaneSetup()	Setup of craftingPane
- void setSelectedRecipe(Node vb)	Set the selectedRecipe with the specified clicked button
- void craftPressed()	Crafting handler
- void containerSetup()	Setup of all containers

4.5. public class FarmScene extends StackPane

Field

Name	Description
- GameInstance gameInstance	GameInstance
- CashDisplay cashDisplay	CashDisplay
- ImageView barn	ImageView with IMAGELOADER.BARN
- ImageView shop	ImageView with IMAGELOADER.SHOP
- ImageView mart	ImageView with IMAGELOADER.MART
- ImageView plot	ImageView with IMAGELOADER.PLOT
- ImageView cage	ImageView with IMAGELOADER.CAGE
- ImageView factory	ImageView with IMAGELOADER.FACTORY

Constructor

Name	Description
FarmScene(GameInstance gameInstance)	Initiate FarmScene by calling all setup methods

Method

Name	Description
+ void updateCashText(int c)	Update cashDisplay text to the specified amount
- void imageViewSetup()	Initiate all ImageViews and set them up
- void cursorSetup()	Apply setCursor(CURSOR.HAND) to all ImageViews
- void eventSetup()	Event handler of all ImageViews
- void cashDisplaySetup()	Setup of cashDisplay
- void relocateImageView()	Set the position of each ImageView

4.6. public class HomeMenuScene

Field

Name	Description
- Stage stage	Stage
- *<u>Scene scene</u>*	Scene
- VBox buttonContainer	Container of all buttons
- GameButton playButton	Play button
- LuckyFontText title	Game title that should be set to “Farm Tycoon”
- GameButton tutorialButton	Tutorial button

- GameButton creditButton	Credit button
- *StackPane root*	Root pane

Constructor

Name	Description
+ HomeMenuScene(Stage stage)	Set stage with the specified value, initiate new GameInstance, and call of setup methods

Method

Name	Description
- void componentSetup(GameInstance gameInstance)	Setup of all component fields
- void eventSetup(GameInstance gameInstance)	Event handler of all component fields
- void stageSetup()	Setup of stage
+ StackPane getRoot()	Return root
+ Scene getScene()	Return scene

4.7. public class MinimartScene extends StackPane

Field

Name	Description
- GameInstance gameInstance	GameInstance
- VBox container	Super container that contains all sub containers
- OrbitFontText title	Title of the scene that should be set to “Your Minimart”
- VBox titleContainer	Container of title

- VScroll itemSelector	VScroll that contains all sellable item
- InfoPane infoPane	InfoPane which displays description and sellable price of the selected sellable item
- StackPane topContainer	Contains backButton and cashDisplay
- String selectedItem	String with the name of selected sellable item
- HBox mainContainer	Contains itemSelector and infoPane
- CashDisplay cashDisplay	CashDisplay

Constructor

Name	Description
+ MinimartScene(GameInstance gameInstance)	Initiate MinimartScene by adding all values from MartItem enum class. Next, set the CashDisplay value with player's balance. Finally, initiate and setup all component fields.

Method

Name	Description
- void updateCashText(int x)	Update cashText value
- void vScrollSetup()	Setup of itemSelector
- void vButtonSetup()	Setup of itemSelector's buttons
- void infoPaneSetup()	Setup of infoPane
- void setSelectedItem(Node vb)	Set the selectedItem with the specified clicked button
- void sell()	Selling handler

4.8. public class PlotScene extends StackPane

Field

Name	Description
- GameInstance gameInstance	GameInstance
- VBox container	Super container that contains all sub containers
- OrbitFontText title	Title of the scene that should be set to "Plot"
- VBox titleContainer	Container of title

- StackPane topContainer	Container of cashDisplay and backButton
- PlotGrid plotGrid	PlotGrid
- PlotControl plotControl	PlotControl
- CashDisplay cashDisplay	CashDisplay

Constructor

Name	Description
PlotScene(GameInstance gameInstance)	Initiate PlotScene by converting each element in player’s plots to PlotSquare then add them to plotGrid. Next set the cashDisplay value with the player’s balance. Finally, initiate and setup all component fields.

Method

Name	Description
- void plotSquareEvents() - Boolean hasEnoughMoney(int c)	Setup event handler of all PlotSquare
- void updateCashText(int x)	Update CashDisplay value with the specified amount
- void loadUnlockedSeed()	Add buttons to seedBox in plotGrid corresponding to player’s unlockSeeds
- void loadPlots()	Initialize PlotGrid
- void plotHarvested(Plot plot)	Plot harvested handler
- void plotWatered(Plot plot)	Plot watered handler
- void plotPlanted(Plot plot, Seed seed)	Plot planted handler

4.9. public class ShopScene extends StackPane

Field

Name	Description
- GameInstance gameInstance	GameInstance
- VBox container	Super container that contains all sub containers
- OrbitFontText title	Title of the scene that should be set to “Shop”
- VBox titleContainer	Container of title
- VScroll itemSelector	VScroll that contains all buyable item
- InfoPane infoPane	InfoPane which displays item price and description of the selected buyable item
- StackPane topContainer	Contains backButton and cashDisplay
- String selectedItem	String with the name of the selected buyable item
- HBox mainContainer	Contains itemSelector and infoPane
- CashDisplay cashDisplay	CashDisplay

Constructor

Name	Description
+ ShopScene(GameInstance gameInstance)	Initiate ShopScene by adding all buyable items from ShopItem enum class. Next set the CashDisplay value with the player’s balance. Finally, initiate and setup all component fields.

Method

Name	Description
- void updateCashText(int x)	Update cashText value with the specified amount
- void vScrollSetup()	Setup of itemSelector
- void vButtonSetup()	Setup of all buttons in VScroll
- void infoPaneSetup()	Setup of infoPane
- void setSelectedItem(Node vb)	Set the selectedItem with the specified clicked button
- void purchase()	Purchasing handler

5. Package utility

5.1. public class Utility

Name	Description
+ Item seedToProduct(Seed seed)	Return Item object corresponding to seed type
+ Seed stringToSeed(String s)	Return Seed object corresponding to name s
+ String seedToString(Seed s)	Return string of seed name of Seed object s
+ Product stringToProduct(String s)	Return Product object corresponding to name s
+ Item stringToItem(String s)	Return Item object corresponding to name s
+ Animal stringToAnimal(String s)	Return Animal object corresponding to name s

6. Package resource

6.1. public enum ImageLoader

Field

Name	Description
- final String name	Image name Note For instance, if the file name is "ProgMeth.png" it should be set to "ProgMeth".
- final Image image	Image

Constructor

Name	Description
+ ImageLoader(String name)	Initiate ImageLoader with the specified name

Method

Name	Description
+ Image getImage(ImageLoader i)	Return Image of the specified ImageLoader
+ ImageView getImageView(ImageLoader i)	Return ImageView of the specified ImageLoader
+ ImageLoader itemToLoad(Item item)	Return imageLoader of the specified Item object
+ ImageLoader seedToLoad(Seed seed)	Return imageLoader of the specified Seed object
+ ImageLoader animalToLoad(Animal animal)	Return imageLoader of the specified Animal object

7. Package main

7.1. public class Main extends Application

Method

Name	Description
+ void start(Stage primaryStage)	Create new HomeMenuScene
+ void main()	Launch application