UML Diagrams for the 'Mowing Before Hoeing' Game

Designed and Developed by Megan Steenkamp and Lewis Marshall

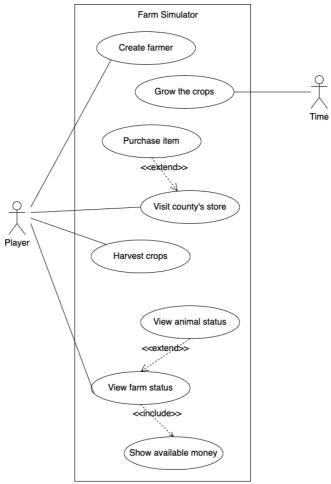


Figure 1: 'Mowing Before Hoeing' Game UML Use Case Diagram

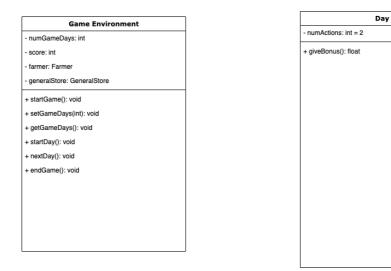


Figure 2: 'Mowing Before Hoeing' Game UML Class Diagrams for the Game Environment and Day Classes

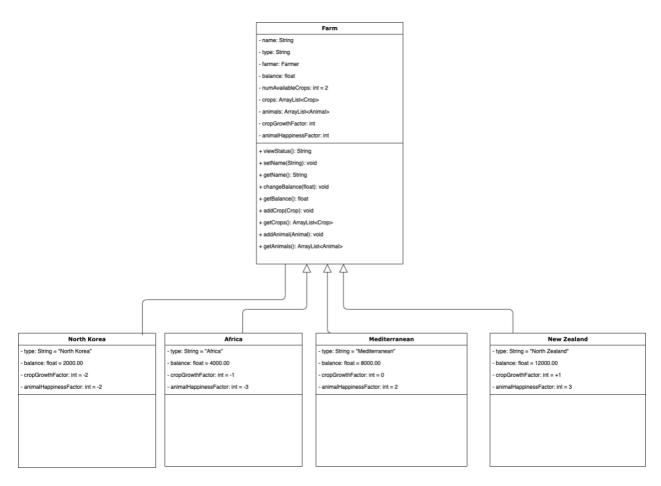


Figure 3: 'Mowing Before Hoeing' Game UML Class Diagrams for the Farm Class and Subsequent North Korea, Africa, Mediterranean, and New Zealand Farm Type Subclasses

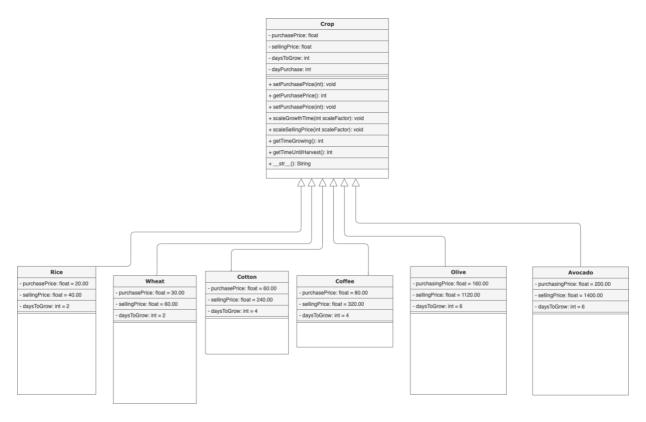


Figure 4: 'Mowing Before Hoeing' Game UML Use Case Diagram for the Crop Class and Subsequent Rice, Wheat, Cotton, Coffee, Olive, and Avocado Crop Type Subclasses

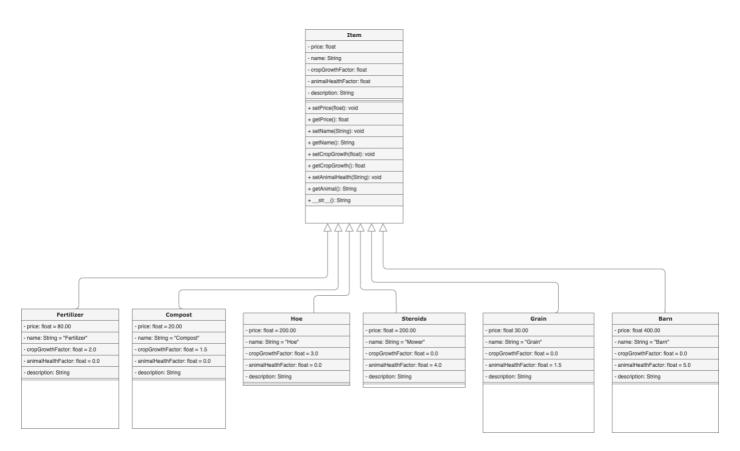


Figure 5: 'Mowing Before Hoeing' Game UML Use Case Diagram for the Item Class and Subsequent Fertilizer, Compost, Hoe, Steroids, Grain, and Barn Item Type Subclasses

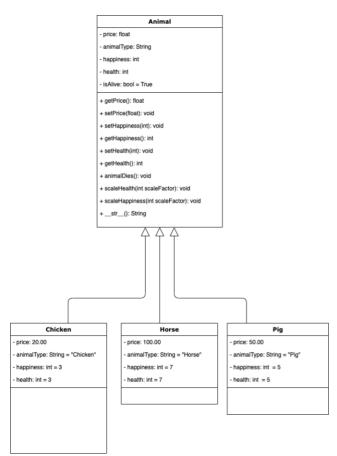


Figure 6: 'Mowing Before Hoeing' Game UML Use Case Diagram for the Animal Class and Subsequent Chicken, Horse, and Pig Animal Type Subclasses

Farmer	
- name: String	
- age: int	
- items: ArrayList <item></item>	
+ setItems(ArrayList <item>): void</item>	I
+ getItems(): ArrayList <item></item>	
+ setName(String): void	
+ getName(): String	
+ setAge(int): void	
+ getAge(): int	
+ enterStore(): void	
+ exitStore(): void	
+ tendToCrops(Item): void	
+ feedAnimals(Item): void	
+ playWithAnimals(): void	
+ harvestCrops(): void	
+ tendToFarm(): void	

Figure 7: 'Mowing Before Hoeing' Game UML Use Case Diagram for the Farmer Class

- items: ArrayList <item></item>
- crops: ArrayList <crop></crop>
- animals: ArrayList <animal></animal>
+ getItems(): String
+ getCrops(): String
+ getAnimals(): String
+ sellitem(Item): Item
+ sellCrop(Crop): Crop
+ sellAnimal(Animal): Animal
+ viewInventory(Farmer): String

Figure 8: 'Mowing Before Hoeing' Game UML Use Case Diagram for the General Store Class