

Farmers and Food

Newly Added Classes and its Responsibilities:

1. Farmer Class

- extends Human class as they share the same characteristics and abilities as human and therefore reduce the need of repeating codes.
- it has an attribute of FarmBehaviour Class that allows Farmer to do their objectives in the game which is farming and crops.
- it uses a parent class constructor which is human to initialise 3 attributes; String name, char displayChar, and int hitPoints.
- used to represent Farmer in the game and has an @override playTurn method that returns an action to play from FarmBehaviour class at a given turn.

2. FarmBehaviour Class

- implements Behaviour interface as it is a new type of behaviour for Farmer.
- it is used to create actions that Farmer can do such as sowing crops if Farmer actor is standing near a patch of dirt, which drops Crop item from inventory to the dirt ;Harvesting fully grown Wheat and drops it to the ground ;Fertilising crops to reduce the amount of turn required for it to be a fully grown Wheat.
- @Overrides getAction method in Behaviour Interface which returns a FarmAction.

3. FarmAction Class

- Extends Action abstract class as it is also a new type of action and therefore help to reduce duplication of code.
- @Overrides execute method which the Farmer performs an action and displays the result in the I/O.
- @Overrides menuDescription method which returns a string of a description to be displayed in the menu.

4. Food Interface

- It is an interface for all types of food items in the game.
- has attributes -int heal, which is the amount of hit points it will restore when being consumed by Human or Player.

- has an attribute- String verb, which is the verb used when displaying the results after consuming this Food. For example, “chews” and “drinks”.

5. FoodItem Abstract Class

- extends Item abstract class and implements Food interface.
- contains attributes- int heal, String verb
- Used to represent items that are used as a food that can be consumed to recover health points.
- has 2 accessor methods heal() and verb() which returns the amount of hit points recovered and the verb to be used to display. “Player drinks milk and recovers for 20 hit points.”

6. Crop Class

- Extends Item abstract class.
- is an item with attributes name = Crop, displayChar = ‘c’, and portable = true
- it also has an int countdown attribute that records the number of turns required to grow into a ripe crop which is Wheat. This countdown is updated using tick() method in Item abstract class.
- @Overrides tick() method in Item abstract class and decreases the countdown by 1 each time the tick() method is called. When the countdown attribute is 0 , it will spawn a Wheat in the same location.
- has a new spawnWheat() method that drops a newly instantiated object of type Wheat on the location.
- used by Farmer to sow crops.

7. Wheat Class

- extends FoodItem abstract class
- has attributes name = Wheat, displayChar = ‘C’, heal = 20, String = “consumes”, portable = true -can be picked up by Player and Human to be used as food.

8. EatAction

- extends Action abstract class and therefore can reduce duplication of code.

-it is used to consume FoodItem by removing it from an actor's or player's inventory and recover some health points of the actor.

@Overrides the execute method in Action and returns a String to be displayed to user at the end of each turn.

-@Overrides menuDescription method which returns a String of a description to be displayed in the menu. For example, "Player consumes Wheat and recovered 20 health points."

Rising from the Dead

9. HumanCorpse Class

-extends Item abstract class to reduce code duplication as it has same functionality.

-it is used to represent dead Human which has health points ≤ 0 . This item is dropped when a Human is killed by Zombie by modifying the execute method in AttackAction class where the (!target.isConscious()) is false, a new object is created of type HumanCorpse and added onto the map location.

-has attributes String name = humanCorpse, char displayChar = 'X', portable = false.

-It has a new attribute called counter of type int, int countdown = 5-10, randomly generated between 5 to 10 turns, where this represents the number of turns before it spawns a Zombie at the original location.

-modifies the method tick() in Item abstract class which decreases the countdown by 1 every time tick is called. When countdown reaches 0, calls spawnZombie() method and removes the humanCorpse item from the map.

-has a new method spawnZombie() that create a zombie at the current location using addActor(Actor actor) method in Location class, addActor(new Zombie("ZombieName"))

