FIT2099 Object Oriented Design And Implementation

ASSIGNMENT 1

Sutulova Tatiana, 30806151

Elaf

Descriptions of classes

overall responsibilities of the class need to be documented somewhere, as you will need this information to be able to begin implementation. This can be done in a separate text document if you prefer.

To help us understand how your system will work, you must also write a design rationale to explain the choices you made. You must explain both how your proposed system will work and why you chose to do it that way.

The design (which includes all the diagrams and text that you create) must clearly show:

what classes will exist in your extended system

what the roles and responsibilities of any new or significantly modified classes are

how these classes relate to and interact with the existing system

how the (existing and new) classes will interact to deliver the required functionality

You are not required to create new documentation for components of the existing system that you do not plan to change.

MAKE SURE TO MENTION IN THE THIRD COLUMN BOTH HOW CLASS INTERACTS AND WHERE IT COMES FROM

1 Description of new classes

1.1 Vending Machine

Class Name	**Description**	**Relation to/interaction with the existing system**
VendingMachine	Stores and sells items to a Player, which may be used by them for	Vending machine is located on the Ground and cannot be moved,

	different purposes.	removed or stepped on(canActorEnter() method returns false) by the actor. The Player can interact by paying EcoPoints and receiving a desired item in their inventory. The Item class and VendingMachine class are in association relationships, since VendingMachine creates an Item.

1.2 Eggs

Class Name	**Description**	**Relation to/interaction with the existing system**
<<abstract>> Egg	The Egg class cannot be instantiated because it is an abstract class. The Egg has: -age, which starts at 0, when the egg is laid and ends at 20, after which the egg will hatch and the Dinosaur object will be instantiated -turns, which starts at 0 and increases by 1 every turn when it is in Dinosaur inventory (before being laid) - the attributes, inherited from the Item class. -tick() methods, which are used to update age and turns -getDropAction() method, which is used by the Dinosaur to lay an Egg. -hatchEgg() method, which initializes a new Stegosaur / Allosaur object and deletes the StegosaurEgg/AllosaurEgg object.	It is inherited from the Item class and is a parent class for StegosaurEgg, AllosaurEgg classes.

StegosaurEgg	The egg is created (instantiated) if: + Dinosaurs of the same kind breed + Bought from the vending machine and it is a subclass of the Item class as it may be kept in the player's inventory and put on any location afterwards.	Subclasses of Egg and grandchildren classes of Item. A children classes which are derived from parent Egg class and grandparent Item class, inheriting the methods and attributes existing in them.
AllosaurEgg		

1.3 FoodItem

Class Name	Description	Relation to/interaction with the existing system
<<abstract>> FoodItem	The FoodItem has: - the attributes, inherited from the Item class. -the points, that will increase Dinosaurs food level when eaten	The FoodItem class cannot be instantiated because it is an abstract class. It is inherited from the Item class and is a parent class for <<abstract>> MealKit, Fruit.
<<abstract>> MealKit	The MealKit object has: - the attributes, inherited from the Item and FoodItem class.	The MealKit class cannot be instantiated because it is an abstract class. It is inherited from the FoodItem class and is a parent class for AllosaurMealKit, StegosaurMealKit.
AllosaurMealKit	Is used to feed the Allosaur once per kit, increasing Allosaurus food level to maximum.	Subclass of MealKit class, that inherits all the attributes and methods from it. Can be bought from the VendingMachine for a constant number of ecoPoints and can be stored in the user's inventory.
StegosaurMealKit	Is used to feed the Stegosaur once per kit, increasing the Stegosaur food level to maximum.	
Fruit	Is used to feed Stegosaur once per fruit and increases Stegosaur's food level by 30. Attributes: -turns, which starts at 0, when the	Subclass of FoodItem and grandchild of Item class. Fruit class inherits all the methods and attributes of super classes. Fruit can be bought from the

	Fruit is dropped on the ground and changes every turn, till it reaches 20, if 20 is reached it rots away. -tick() method, which is used to update turns	VendingMachine (association) for a constant number of ecoPoints and can be stored in Player's inventory without rotting. Fruit can be picked from the Tree(association) with the possibility of 40%. Fruit can be dropped on the ground by the Tree (possibility of 5%) and picked from the ground by the Player.
Hay	Is used to feed Stegosaur once per hay and increases Stegosaur's food level by 20.	Subclass of FoodItem and grandchild of the Item class. Hay class inherits all the methods and attributes of super classes. Hay can be bought from the VendingMachine (association) for a constant number of ecoPoints and can be stored in Player's inventory. Hay can be harvested from the grass.

1.4 Weapon

Class Name	**Description**	**Relation to/interaction with the existing system**
LaserGun	Is used by a Player to kill a Stegosaur. Its damage is 50, which is enough to kill the Stegosaur in one or two shots.	Subclass of WeaponItem and grandchild of the Item class. LaserGun class inherits all the methods and attributes of super classes. LaserGun can be bought from the VendingMachine (association) for a constant number of ecoPoints and can be stored in Player's inventory.

1.5 Dinosaur

Class Name	Description	Relation to/interaction with the existing system
<<abstract>> Dinosaur	The abstract class that has attributes and methods that are implemented by both Stegosaur and Allosaur objects. The attributes include: + The isConscious() method inherited from the Actor class and adds the additional functionality to make the Dinosaur unconscious if the food level is zero. + The playTurn method will be updated to make sure if the Dinosaur stays unconscious for 20 turns, it will die. + foodLevel: int (0-100), which is 10 if the Dinosaur is hatched from the egg. It decreases by 1 every turn. May be increased if the dinosaur is fed. + gender: String, which is needed for breeding + age: int, which is needed to see if the Dinosaur is old enough to breed + status: ALIVE, DEAD - If the food level is more than 50 and age is more than 30, the Dinosaur is able to breed. - When the unconscious level reaches 20, status becomes DEAD and the dinosaur can be only eaten by Allosaur.	*Subclass of Actor class and is parent for Stegosaur and Allosaur classes.* To create a new Dinosaur (any kind): + Player can buy an Egg in a VendingMachine, place it at any Location and wait for 10 turns for it to hatch. + Dinosaurs of opposite genders (same kind) may breed by meeting at adjacent cells and after 10 turns the female Dinosaur lays eggs that will hatch after 10 rounds. After the Egg hatches, the dinosaur is born with the age of 1 and food level of 10.

Stegosaur	Subclass of Dinosaur superclass. Unlike Allosaur, Stegosaur is a herbivore and implements only vegetarian behaviours: eats hay, grass, leaves and fruits, does not kill and attack. + foodLevel: int is 50 for a Stegosaur created by default when the game is started	*Subclass of the Dinosaur class, implementing all the attributes and methods mentioned above.* + can be fed by the Player with vegetarian food. + can be killed by the Player with the LaserGun. + when the Stegosaur steps on a new Location it checks whether it is a Grass or not. If it's the Grass and Stegosaur's food level is not full, then the Stegosaur eats the Grass, which increases his foodLevel by 5 points and leaves the Dirt afterwards.
Allosaur	Subclass of Dinosaur superclass. Unlike Stegosaur, Allosaur is a carnivore and implements only carnivore behaviours: moving towards the dead Dinosaur to eat it, attack the Stegosaur when it is close and eat eggs if they are close.	*Subclass of the Dinosaur class, implementing all the attributes and methods mentioned above.* + can be fed by the Player with non-vegetarian food (carnivore kit) . + can eat the dead dinosaur or an Egg if it is found at some location.

1.6 Actions

Class Name	**Description**	**Relation to/interaction with the existing system**
BuyItemAction		
FeedAction		

HarvestGrass		
SearchTreeAction	Only activates when the Player is standing on the same cell as the Tree. If the Fruit is dropped, it will be added to the Player inventory. If the Tree did not drop the Fruit, there is a 40% chance of player finding a fruit on tree and it will be added to the Player inventory.	This action allows the Player to find the Fruit from the Tree and keep it in its inventory.

1.7 Behaviour

Class Name	Description	Relation to/interaction with the existing system
HungryBehaviour		
BreedBehaviour		

2 Changes to existing classes

Class Name	Description
Player	Adding ecoPoints that are used to buy Items from the VendingMachine. EcoPoints are increased if: + 1 when Grass grows/harvested + 10 when Dinosaur is fed with hay + 15 when Dinosaur is fed with Fruit + 100 when StegosaurEgg hatches + 1000 when AllosaurEgg
Item	Adding price in ecoPoints to sell it in the VendingMachine to a Player.
Tree	Adding Fruit implementation, allowing the Player to collect a Fruit from the Tree or dropping the Fruit on the ground allowing to collect it.

GrassDirt	

3 Reason for choosing this design

Documentation quality Does your design documentation clearly communicate your proposed changes to the system? This can include:
UML notation consistency and appropriateness
consistency between artefacts
clarity of writing
level of detail (this should be sufficient but not overwhelming)
Explanation Can you adequately explain your design and the reasoning

4 Work Breakdown Agreement

We require you to create a simple Work Breakdown Agreement (WBA) to let us know how you plan to
divide the work between members of your team. There is more information on WBAs in the FIT2099 Assignment Rule