Design Rationale

• This file explains all the classes we are about to add and all the reason for the existence of the classes and their role in our game, it is also mentioned a relationship and a few functions given that we will use to implement the roles for these classes, also consists of a table of display character references and some implementation notes at the bottom.

1) Bushes Class – extends Ground and overrides the tick function to call the ripening() method every turn with a 10% chance of growing a fruit on the bush, there is also a 50% chance for The bush to die if it is stepped on by a brachiosaur which we have implemented using killBush() method.

2) Dirt class – extends Ground, it will override the tick() method and for every turn it will call growBush() method and this method will check if the Dirt is next to atleast to squares of bushes or next to a tree, if it is next to two or more squares of bushes the possibility for the square of dirt to trun into a bush will be 10% and if next to a tree it will have 0% chance, if both these conditions are not met it will have a 1% chance to grow a bush. When a Bush is grown a Bush instance is created and added to the location of where the Dirt is on the Game map.

3) FoodItem Class- An abstract class that extends PortableItem class because all food items are portable and contains the basic common functionality of all the food items in the game.

4) Fruit Class – extends FoodItem and overrides the tick() function to call a function rot() which keeps track of whether the Fruit has been on the floor for over 15 turns and should rot.

5) Enum FruitStatus – has values {ON_FLOOR, ON_TREE, ON_BUSH, IN_INVENTORY} to add the capability to the Fruit class for us to know where the fruit is located.

6) Tree Class- will be updated to have a ripening() method just like bush to check if the tree has produced fruit (chance of producing ripe fruit for a tree is 50%) implemented by ripening() method and this will update the static points variable of the Player and add 1 point, there is also a 5% chance for the tree to drop the food, this is implemented by dropsFruit() method.

7) Egg Class – extends the item class and overrides the tick function to find out how long it has been since the egg has been created and if the required number of turns for the egg is met then the Egg will run hatch() method which will create a dinosaur of the eggs species. The egg will also store what dinosaur the egg belongs to and will create a new instance of that specie of

dinosaur depending on its hatch time and the eggs will not hatch if there is an actor on its location.

- • 10 turns for stegosaur egg to hatch.
- • 20 turns for brachiosaur egg to hatch.
- • 50 turns for allosaur egg to hatch.

8) Points Class – A class used to set and get the points value for the Actor using the Points class as a variable.

9) Player Class – will be updated to use a static variable of Points, so that this variable will be updated by the required Classes such as Tree, VendingMachine, Egg and etc... So that when the task of a ripe fruit is produced on a tree or a dinosaur hatches or a Item is bought from a vending machine, the required classes will update the Points on the Player accordingly, by reducing or increasing of the points.

10) Dinosaur Class – an abstract class for all the dinosaurs being created and the common functions and behaviours that all the dinosaurs will have when any dinosaur is created it will randomly be decided whether it is a male or female. It will also have a second constructor for an Admin to create Dinosaurs at the beginning and set the genders he/she wants.

11) Brachiosaur Class – extends Dinosaur and Overrides the required methods from the Dinosaur class, is a dinosaur that can only feed on fruits on a tree for 5 hp gained per fruit. Baby brachiosaurs only start with 10 hp and grows to adult and has 100, with a max hp of 160, they also become hungry at food level 140, they only eat fruits on the floor and fed by players and vegetarian meal kits fed by the player.

12) Allosaur Class – extends Dinosaur and Overrides the required methods from the Dinosaur class. Allosaurs will use the AttackAction with their weapon "claws" and deal 20 damage to the target stegosaurs, Baby allosaurs only do 10. Allosaurs will also have a list of stegosaurs they already attacked and cannot attack those stegosaurs until 20 turns pass by. They also cannot follow Brachiosaurs so they cannot have Brachiosaurs as targets Dinosaur's species for FollowBehaviour, they are born with 20 hp and grow to get 50 hp and have a max hp of 100, they get hungry at a food level of 50, they eat Eggs of all species and Corpses of all dead dinosaurs and Carnivorous meal kits when fed by the player.

13) Stegosaur Class – extends Dinosaur and was made for us, but the changes made causes them to be born as babies with a food level of 10 and will grow up to have a food level of 80, they have a

max food level of 100 and only eat fruits that are on the ground and on bushes and fed by the player, and meal kits fed by the player.

Note – All Behaviours of the dinosaur below are used to check whether the required conditions for the Actions to happen are met and if it does it will create and return the Action that the dinosaur will do.

14) BreedingBehaviour– extends Behaviour and Creates and returns the Action BreedingAction if the Dinosaur has more than the required foodlevel needed for breeding. If it does it will use the FollowBehaviour to follow a Dinosaur of the same species and opposite gender and the female is not pregnant and they will both do the BreedingAction(which extends Action).

15) HungryBehaviour– extends Behaviour and Creates and returns the EatingAction, It will first check if the Dinosaur has the less than the required food level to become hungry and if It does it will try to find its food and it will return the EatingAction(which extends Action) to eat the Food it has found, this means its heal() method will be called and will heal by required amount depending on the FoodItem it is consuming and whether or not the Dinosaur is a baby and the targets food level will decrease(if it is a dinosaur).

16) GrowingBehaviour – extends Behaviour and creates and returns the GrowingAction if the Dinosaur is a baby, the GrowingAction(which extends Action) will keep track of the dinosaurs age(number of turns since it was born) and if it meets the required value.

    • Stegosaur - 30 turns to grow.
    • Allosaurs and Brachiosaurs – 50 turns to grow.

17) LayEggBehaviour – extends Behaviour and creates and returns the LayEggAction if the dinosaur is a pregnant female, the LayEggAction(which extends Action) keeps track of how long the dinosaur has been pregnant and when the number of turns to lay an egg is reached it will create a new Egg item and the dinosaur will no longer be pregnant.

18) DeathBehaviour – extends Behaviour and creates and returns the DeathAction if the dinosaur has been unconscious for over the required amount of turns for its species and DeathAction(which extends Action) causes the dinosaur to die and creates a DinoCorpse object for the specie of dinosaur.

19) DinoCorpse Class– extends item and overrides the tick function to keep track of the number of turns that pass after it has died and calls the decay() function when the dead corpse has been on the map for the required amount of turns for the species.

20) LazerGun Class – extends WeaponItem and sets the damage that the lazer gun instance does (approximately a 1-2 hit to kill a stegosaur), we can use the AttackAction and Actor.getWeapon() to get this weapon and attack dinosaurs targets using it.

21) VendingAction Class – extends Action and overrides the menudescription() method to show options of the items the user can buy from the machine, then it takes the user input for the key of the item it has to enter to the VendingMachine, it validates if the entered key is correct and if a valid key is entered it sends this data to the VendingMachine instance and lets it process the points and transaction and if successful will add the item to Players inventory and if unsuccessful will stop the transaction.

22) VendingMachine Class – extends ground and holds a list of all the items it has within it for sale, it checks if the price of the item and the amount of points the users static point variable has and if he doesn't have enough acknowledges the Player, he/she does not have enough points to purchase the item, other wise if they do then the VendingAction adds the item to the Players inventory.

23) FeedAction Class – will override the menuDescription() method to show the option to feed a target and will call the target Actors heal() method to heal the target by required amount depending on the FoodItem. The target can be fed an object of Fruit, MealKit or DinoCorpse depending on whether the species of the Dinosaur is carnivore or herbivore.

24) MealKit Class – An abstract class for the mealkits with all the common functions for both mealkits, MealKits will heal all Dinosaurs to max food level regardless of species.

25) Vege and Meat MealKit Classes – extend MealKit class and will call the Actors heal(maxHitPoints) method to increase its food level to its maximum Hit Point amount. VegeMealKit only works on Vegetarian Dinosaurs and MeatMealKit works only on carnivorous Dinosaurs.

26) PickFruitBushAction Class – is the action by the a player to pick a fruit from the bush, in its execute() it will have a 60% chance for the player to pick a ripe fruit if it does have any on the list of ripe fruits on the bush and will increase the players static points variable to increase the points the player has by 10 and will also remove a fruit from the list of ripe fruits on the bush if a fruit was picked.

27) PickFruitTreeAction Class - is the action by the a player to pick a fruit from the tree, in its execute() it will have a 60% chance for the player to pick a ripe fruit if it does have any on the list of ripe furits on the tree and will increase the players static points variable to increase the points the player has by 10 and will also remove a fruit from the list of ripe fruits on the tree if a fruit was picked.

28) Fish Class – extends food item and will create a fish, which has a heal value of 5. This is another type of food that can be eaten by the dinosaurs

29) FishAction Class – is the action for the player to fish for the fish from the lake that he is standing on, the player can only catch fish if there are fish in the lake that he is on, if not he cannot catch fish, once a fish is caught it is removed from the lake and added into the players inventory.

30) FlyingBehaviour Class – implements the Behaviour interface and will allow any dinosaur to fly, if the dinosaur does have a fuel variable, it checks if the dinosaur is on a tree and if yes gives it 30 fuel, if it is not it checks if the dinosaur has fuel and if it is hungry or thirsty, if yes it will fly to nearest food source, but if the dinosaur doesn't have fuel or does have fuel but isn't hungry or thirsty it will find the closest tree and get back to it.

31) Lake Class – extends ground and will save a list of all the fish that exist in the lake, it can have a maximum of 25 fish and stores the number of sips of water that the lake has. If a dinosaur drinks from the lake, the sips will decrease by one and if a dinosaur eats a fish or player catches a fish, a fish will be removed from the lake. Every turn it is also checked if it has rained which has a chance to occur every 10 turns and if it does it will add a random number of sips, depending on the rainfall to all the lakes in the map.

32) Sky Class – extends ground and will be used to rain to all the places in the map, the rain value is the amount it rained for when the rain occurs, it is a random integer from 2 to 12 according to the percentages, raining will occur every 10 turns with a 20% chance of successfully raining. This class will check if that 10th turn has arrived and if the probability to rain has been met.

33) SwitchMap Class – extends Ground to create an unenterable border that acts as a gateway between the maps. This exists at the very top of the southern old map, and at the very bottom of the northern new map. This cannot be entered by the player but can be interacted to allow the player to move between the maps

34) ThirstyBehaviour Class – implements Behaviour interface and checks if the conditions for the dinosaur to get thirsty and find and drink water are met and if yes it will cause the action for the dinosaur to either search for the closest lake if it is not near one and if it is next to one it will return a DrinkingAction which makes the dinosaur drink water from the lake it has found.

35) DrinkingAction Class - The action that causes the dinosaur to drink water from a lake, this will remove a sip from the lake that the dinosaur has drunk from and increase the water level of the dinosaur by the required amount

Some details of implementations used

1) The corpses are all 'X' since it is hard to identify between dead corpses until you go onto it, so to make it more realistic we made all corpses hard to distinguish and name will be told before picking it up.
2) We allowed the user to buy any number of one item only from the vending machine per turn.
3) We changed values of the hatch times, growth times, hunger levels and more for the dinosaurs which we have listed in the details above.
4) We also used the interfaces provided to include methods to drastically decrease the amount of down casting that should be done to the Actor, Ground, and Item classes to their subclasses.
5) We have fully utilized the Display class provided and have not used System.out.println() anywhere in our program.
6) Allosaurs will 100% land a hit when using AttackAction to attack, unlike the player. This helps improve the game balance a lot.

Item or Actor Name	Corresponding Display Character
Stegosaur	's' for baby and 'S' for adult
Brachiosaur	'b' for baby and 'B' for adult
Allosaur	'a' for baby and 'A' for adult
Pterodactyl	'p' for baby and 'P' for adult
Stegosaur Egg	'o'
Brachiosaur Egg	'O'
Allosaur Egg	'0'
Pterodactyl Egg	'Q'
Lazer Gun	'='
Vegetarian Meal Kit	'^'
Carnivorous Meal Kit	'%'
Fruit	'f'
Tree	'+', 't' age more than 10 and 'T' age more than 20
Floor	'_'
Wall	'#'
Vending Machine	'V'
Dirt	'.'
Bush	'*'
Player	'@'
Corpses of all dinosaurs	'X'
Sky and Border	'`'
Lakes	'~'
Fish	'!'