Design Rationale

By: Chong Chiu Gin (28842022) & Chin Wen Yuan (29975239)

Growing Plants implementation

Grass

A new class called Grass that extends Ground will be opened. A method will be added in Dirt class that enables it to have a chance to grow into Grass at every turn.

Tree

Another method will be implemented in Tree to ensure that every Ground neighbour of an existing alive Tree can grow into a Tree. To control the growth of tress, we will implement that only Dirt can grow into Tree. This is to ensure that other classes that extends Ground like Grass, Wall and Floor that is neighbour to Tree won't be replaced into a Tree.

Dinosaurs implementation

Protoceratops and Velociraptors extends Dinosaurs (abstract class)

A class called Dinosaurs is made into an abstract class so we can ensure no Dinosaurs objects are instantiated. Instead two classes called Protoceratops and Velociraptors are extending this Dinosaurs class. Both these subclasses contain same attributes and same methods which can be declared in the parent class Dinosaurs. If they have different implementation of methods, these methods can be declared abstract methods in the Dinosaurs class to obey the principle of "Do not repeat yourselves".

BreedAction

Dinosaurs has a collection of Behaviours. All dinosaurs come with a food level that can affect their behaviours. When the dinosaurs are sufficiently well-fed which means they have a sustaining food level, they are given a chance to breed by laying an egg. Random will be imported and called to decide whether the dinosaurs will breed as their next action. BreedAction that extends Action class will execute if it was decided that the dinosaurs will breed. The BreedAction class will create an Egg item in the execute() method that can be picked up by Player or left on the Ground.

HungryBehaviour, EatAction and AttackAction

If food level gets too low, the dinosaurs will have HungryBehaviour. HungryBehaviour is a class that implements Behaviour class and it is used to decide on what is the next action the dinosaurs should take when it is hungry. Protoceratops has HungryBehaviour and will perfrom EatAction on herbivore food or plants like trees and grass. Velociraptors will first check the Ground where the Velociraptors is standing on to see if there is food (Egg, corpse or carnivore food). Velociraptors will perform an EatAction which is a class extending Action class if there is. If there is no food on the Ground, then it will check to see it's surrounding for food and perform MoveActorAction(from the engine package) towards the food. When no food is found, the Velociraptors will then see if there is Protoceratops in the surrounding so it can perform AttackAction to reduce the hitpoints of Protoceratops.

FeedAction and TagAction

Dinosaurs' getAllowableAction will include FeedAction and TagAction that extends the Action class when Player is standing next to the dinosaurs. Both FeedAction and TagAction will have one target dinosaur. These two class has a dependency relation to Dinosaurs class to ensure the principle of "Classes are responsible for their own properties" is met. FeedAction allows Player to hand feed herbivore food to Protoceratops or carnivore food to Velociraptors whereas TagAction allows player to tag the dinosaurs that is to be sold.

Items

PortableDinoItem

Aside from Egg class, items like corpse, carnivore food, herbivore food and dinosaur tag are made and designed as instances of PortableDinoItem rather than a concrete class. This is to avoid redundancy classes that don't have particular functions or operations to implement since they are mostly being acted upon. This also follows the principle of reduce dependency because this design ensure there are lesser classes extending to PortableDinoItem.

Egg

Egg class is designed because dinosaur eggs have extra functionality like having an age attribute and a method called hatch() to create a new instance of dinosaurs and remove the Egg item from the GameMap. These dinosaurs will be baby dinosaurs for some period of turns before turning into adult. These baby dinosaurs cannot be perform BreedAction.

Buying and Selling at Shop

Shop with BuyAction and SellAction

Shop class is created and extends Ground class because Shop has to be displayed on GameMap as 'S' character in the building on the top left. The allowableAction method needs to be override in the Shop class so it can allow Player to perform BuyAction and SellAction whenever Player reaches the building and is standing next to 'S' character of the Shop on GameMap. Shop has a dependency relation with both BuyAction and SellAction class instead of Player to fullfill the principle of classes should be responsible for their own properties.