DESIGN RATIONALE

When a Zombie inflicts an attack such as a bite attack , the AttackAction and AttackBehaviour will handle this action, therefore no new classes are required to implement this requirement. However, to emulate zombies to cry out “Braaaaains”, a new Behaviour class is implemented, SpeakBehaviour class which extends from Behaviour class. This will generate a SpeakAction class which every 10% chance will have the zombie perform a SpeakAction.The SpeakBehaviour class will be added to the behaviours of the Zombie class.

Implementing the body anatomy of the zombie will be done in the Zombie class, where new attributes such as noOfZombieArms and noOfZombieLegs will represent the state of the zombie’s anatomy, and new methods will implement the conditions of the state of the zombie. There is also another attribute called knockOffChance which determines the probability that the zombie will lose one of its limbs. Once a Zombie knocks off its arms/legs,a ZombieArm or ZombieLeg which both extends from Item will drop at the Zombie’s Location. Therefore the responsibility of the ZombieArm and ZombieLeg is to represent the zombie’s knocked off arms and legs.

Weapon crafting is done through the WeaponCraftingAction class. This is created to ensure that the Player has a new action to craft the weapons if the conditions are met. A player can select the WeaponCraftingAction action if and only if they are holding a ZombieLeg or ZombieArm item. These Item classes can be transformed into ZombieClub or ZombieMace WeaponItem’s and are added into the Player’s inventory. These extends the WeaponItem class since they are weapons.

A new class is created to represent corpses, which is the Corpse class which extends from Item class. The responsibility of this class is used to represent corpses and to meet the requirements.

The new classes for farmers are the Farmer class which will extend from Human class. A new class called Crop will also be created to represent crops. The Farmer class will be able to use the CropBehaviour class, which extends from the Behaviour class. The responsibility of this class to determine the appropriate actions for both Player and Farmer class such as HarvestAction (action used to harvest a ripe Crop) and PlantAction (action used solely by Farmer to plant new Crop) and FertilizeAction (action used solely by Farmer to fertilize Crop), therefore CropBehaviour depends on these actions. There is also EatAction, which can be used by Human and Player class to eat a Crop.

Finally, a Crop can be converted into a food, therefore another class is created to represent food called Food class which extends PortableItem since it is portable and can be carried around.