**DESIGN RATIONALE**

ZOMBIE ATTACKS

Implementing Zombie Bite

All features described in this segment are depicted in Class Diagram – Zombie.

The Zombie Bite feature may be added by modifying the Zombie Class method getIntrinsicWeapon(). A probability check can be added that, if successful, will return a new IntrinsicWeapon(10, “bites”) instead of the default IntrinsicWeapon already written.

The damage number is subject to change, but the verb “bites” will be used to identify the weapon as a bite, and so will be unique to this type of Zombie attack.

The probability check used to determine the type of attack must be modifiable based on how many arms the Zombie has. This can be done by checking the value of the zombieArms variable in the Zombie object making the attack.

The AttackAction class will be modified, to implement characteristics specific to the Zombie’s bite attack. It will do this by checking the verb associated with the weapon when an attack is made; if IntrinsicWeapon.verb() == “bites”, a lower hit chance will be factored into the attack roll, via if/else switch.

The same IntrinsicWeapon.verb() == “bites” check will also determine whether a successfully landed attack restores health. If the attack lands and the verb check is successful, the bite will execute actor.heal(5) to heal the zombie.

Implementing Zombie Weapon Pickup

All features described in this segment are depicted in Class Diagram – Zombie and Sequence Diagram – Zombie Pick Up Weapon.

To accommodate Zombies picking up weapons, a new behavior class named PickUpBehavior will be added to the Zombie class behaviors ArrayList. This behavior will be lower priority than AttackBehavior, but higher than HuntBehavior or WanderBehavior.

PickUpBehavior will first check that the Zombie has arms, by checking the zombieArms variable in the Zombie object making the attack.

The presence of a weapon on the ground that is eligible to be picked up must then be checked. This can be done by using map.LocationOf(zombie) to get the zombie’s current map location, then calling getItems() on that location to fetch all items on the ground at that location, then calling item.asWeapon() to check the item’s status as a weapon.  
  
If there is a valid Weapon on the ground in the same location as the zombie, PickUpBehavior will return a pickUpItemAction for that Weapon, which Zombie will then execute.

Implementing Zombie Moan

Every time the playTurn method in Zombie is called, before the program loops over the Behavior ArrayList, a random probability will return a 10% chance that the system prints a zombie moan.

BEATING UP THE ZOMBIES

Implementing Zombie dismemberment