# Design Rationale

## Zombie Attacks

**Zombies should be able to bite. Give the Zombie a bite attack as well, with a 50% probability of**

**using this instead of their normal attack. The bite attack should have a lower chance of hitting**

**than the punch attack, but do more damage – experiment with combinations of hit probability and damage that make the game fun and challenging. (You can experiment with the bite probability too, if you like.)**

This will be implemented in the AttackAction class and the bite attack will be executed if weapon is set to a certain weapon (eg. teeth). If actor is a zombie object, then at the start of the execute method, rand.nextBoolean() will be executed to determine whether weapon will be set for a normal attack or a bite attack (50/50 chance). The probability of the attack missing will be determined by checking rand.nextDouble() <= X, meaning there will be a (X\*100)% chance of missing and X will be determined on what weapon is. If weapon is for a normal attack (ie. fists or a weapon), then X=0.5, meaning a miss probability of 50%. If weapon is for a bite attack (ie. teeth), then X=0.7, meaning a miss probability of 70%.\*

Creating a new class for the bite attack was not chosen to be the best design as it would be similar to the AttackAction class and therefore require duplicated code which goes against the “Don’t repeat yourself” principle.

\*these miss probability may change.

A successful bite attack restores 5 health points to the Zombie

The health restore will be done in the AttackAction class after the bite attack has been executed in the execute method. As the actor object is already passed as a parameter in the execute method, the health restore can be done with actor.heal(5).

If there is a weapon at the Zombie’s location when its turn starts, the Zombie should pick it up.

This means that the Zombie will use that weapon instead of its intrinsic punch attack (e.g. it might

“slash” or “hit” depending on the weapon)

A new class called ScanvengeBehaviour will be created which implements Behaviour. It will contain a getAction method that inspects the items at the actor’s location and picks up items that are a weapon. A ScanvengeBehaviour object will be the first element in the behaviours array in the Zombie class so that it will be executed at the start of its turn.

Creating a new class has been decided as the best design as it will group attributes and methods that depends on each other, following the “Group elements that must depend on each other together inside an encapsulation boundary” principle.

Every turn, each Zombie should have a 10% chance of saying “Braaaaains” (or something similarly Zombie-like)

This will be implemented in the Zombie class at the start of the playTurn method as this event is specific to only Zombie objects. A rand.nextDouble() <= 0.1 condition will be checked and if successful, the zombie will say “Braaaaains”.