**Design Rational for Assignment 1**

1. **ZombieActionClass**

This class is created that extends the AttackAction class to accommodate the need for zombie to decrease their probability of punch when they lose one or both their limb. Modifying the code at AttackAction class has a high risk of breaking it, because some major changes are needed in the execute method, making a new class not only shorten the code but also reduce the chance of breaking the getAllowableAction class which might affect the human class.

1. **WeaponLeg and WeaponHand**

This class is created to create a weapon obtained from fallen zombie’s limb. It inherits the WeaponItem class. We created this class, to make it easier to create an instance of this weapon every time a zombie’s limb fall off. It will be used in the in the *hurt* method in the zombie class, where every time a zombie is hurt, there will be a probability its limb will fall.

1. **Zombie**

*GetIntrinsicWeapon* method is changed to add the probability of having a bite attack. The reason it is added here instead of adding it to the weapon because biting it’s an attack from a part of the zombie (i.e. not weapon), hence its more suitable to have it in the *GetIntrinsicWeapon* method. Besides, adding it here can reduce the lines of code compared to creating a new attack class for it.

*PlayTurn* method is also modified to check for broken leg, so that it can reduce the movement speed on the zombie based on the last action of the zombie. Its implemented in this method because, it’s the only method that keeps track of the zombie last action and it’s in the zombie class, so it is also easier to access the condition of the zombie (e.g. how many legs left) which can reduce repetition compared to creating a new subclass of action for zombie with broken leg.

These designs reduce dependencies as most of the method that required access to the zombie class is implemented inside the zombie class.

1. **Zombie saying ‘Brainns’**

The probability of the zombies saying brains is implemented in the playTurn function where in every turn it takes it will have a probability of saying ‘Brainns’. This is implemented in playTurn instead of making it an action because, in every turn an actor can only take an action. We decided that we should not waste a turn for zombie just to say ‘Brainns’

1. **Limb class**

A class limb that keeps track of the number of limbs a zombie has left is created instead of putting a counter in the zombie class. This follows the design principle that ‘Classes should be responsible for their own properties’, where in this case, the purpose of this class is to keep track of a zombie’s limb. This makes future code changing easier because this code does not have relation to other method.