Tony Nguyen's General Observations and general ideas about Assignment 1

- Zombie attacks

So From my interpretation of the code it seems as though "weapons" are actually what determines the damage to do and are actually what the attack actually is (intrinsic weapons are basically the default weapon I'm guessing). For example zombie has "punches" as it's weapon thus when AttackAction calls .getWeapon() from the actor it is actually calling what type of attack it is (weapon).

So in the zombie class we could make it so it has a 50% chance to set it's weapon to (bite) before performing the action then we have specific control structure in that if an attack happens with bite it restores 5 hp in the zombie object (a heal class maybe?)

Most of this revolves around the playTurn() method in the Zombie class, we could add a behaviour that makes it pick up item as it's turn and here we could also add the 10% to say text

- Beating up Zombies

Where to do the percentage chance of taking off limbs is a question (maybe a method in the zombie class that is called when damage to determine if it does lose a limb).
The limbs could be Boolean variables to indicate if they are on the zombie or not
Dropping items could be a method in zombie class

The limbs are Boolean, thus we could have selection structure in the methods based on the limbs thus giving the movement/attack constraints to the zombie this way
Dropping limbs affects the map and it would technically be putting a weapon onto the map to solve the picking limbs up thing.
We may have to implement a pickup weapons method that happens at the start of its turn in the playTurn().
Something like Boolean hasLeftArm would be our implementation of limbs

- Crafting weapons

If we do the implementation that I have put above about dropping limbs, then if it is dropped onto the map then it already is a weapon so we may have to allow it to work with weapons as an input type.

I don't see any crafting classes in the UMLs provided.
Create a crafting class which interacts with interface and the inventory of the Human/Player classes and allows for many input types.
(Might need to also include a behaviour for this for humans as well).
This one seems to be the one where we create our own as the previous seems to me like we have to change and implement into what is already existing.

- Rise from the dead

Variable for number of turns passed after death.

Implementation could be See if the person is dead, see the number of turns that have passed, if yes then call a method that might (unknown chance) to make them a zombie.

Unsure if 5-10 means that guaranteed that there is a zombie by turn 10 or it has a chance at becoming a zombie from turns 5-10 but after that they cannot rise from the dead.

Second idea sounds better in game but I think maybe First idea in making it?

Unsure about that.

- Farmers

It could extend humans class.

New Behaviour class for potential to farm (farming behaviour needed?).

We could add a variable to dirt indicating whether it has a crop or not as well as the turns till ready.

For the actual implementations, I think the dirt would do a Boolean check before performing farmer actions.

Finally, we could have a food class that could maybe do something with our potential heal class.

General notes:

- cannot change Engine class so most likely changes only to game and interface class
- Assignment 1 is just planning and this is my thoughts out loud