

Right now the RNG for hit probability is happening in the AttackAction class (execute method). The generation is the same for each weapon, and is 50%.

We need to add a new intrinsic weapon to the zombie, called bite. But somehow we have to implement different hit probabilities for different weapons.

Idea 1:

Implement a new class called WeaponProb that has attributes weapon and also a separate integer called hitchance

Idea 2:

Extend WeaponItem and IntrinsicWeapon in the game package with hit probability as an attribute. Make all the characters only refer to these new classes. Override all references to the old weaponitem classes in all the classes in game.

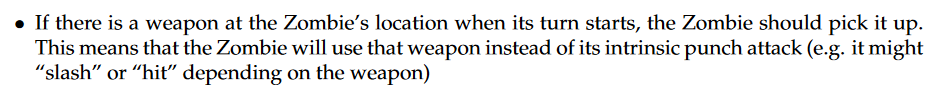
Look into extending the weapon interface

//Implement intrinsic weapons as private attributes of the zombie class, then have this method select randomly between them (spec says 50% probability)

**USE ACTION INTERFACES!**

C:\Users\cedri\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\3768EFE0.tmp

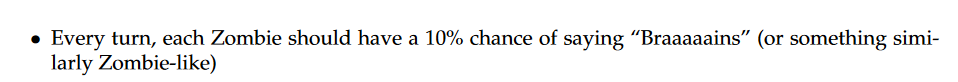
Call the Actor.heal when a bite attack occurs.



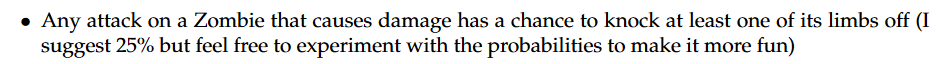
LOOK AT World.processActorTurns

Add new behaviour in the zombie class (as first priority) to pick something up if it can

Overwrite the inventory attribute in the zombie class to have only a length of 1.



Insert it into the playTurn class before the behaviour loop

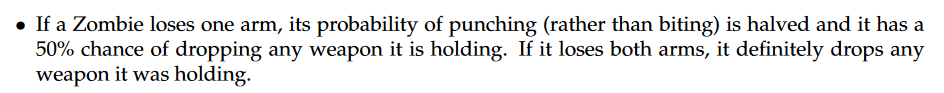


Implement new private attributes in the zombie class that correspond to the number of legs and arms it has.

Implement a new method in ActorInterfaces called takeDamage that handles all actors in game taking damage. Make it also have a reference to the location of the actor, by feeding in 'map', then overwrite the takeDamage(int damage, GameMap map) method in zombie to handle gamelogic about the subtraction of limb count and the creating of an object at the location.

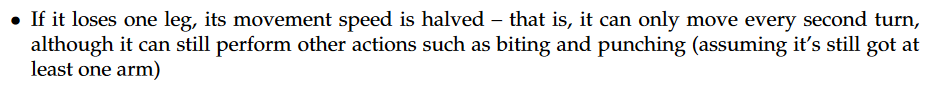
C:\Users\cedri\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\EBDB9538.tmp

Handle this with these attributes having set values on initiation



For the first part, change the gamelogic in the zombie class with regards to which intrinsic weapon it chooses to be dependent on the private attributes the number of arms.

For the second part, we have earlier decided to implement that the zombie can only have one weapon in its inventory. In this case, we can also handle the dropping of weapons in the zombie.takedamage method in the zombie class to call the DropItemAction



World already stores information about what each actor did in the previous turn. We can run a query on each zombie actor, checking if it moved (called the MoveActorAction) in the previous turn, then limit the behaviours it can choose depending on that.

This then will need to run a query on the zombie's private attribute (the number of legs it has remaining)

We have two options

1. Hardcode the behaviours into the playturn method in the zombie class, which lets us add our logic to the processing based on lastAction

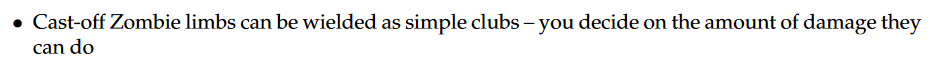
C:\Users\cedri\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\5811C2B2.tmp

Handle this in the same way (check zombie leg count, then only provide Attackbehaviour or donothingbehaviour as options if there are no legs)

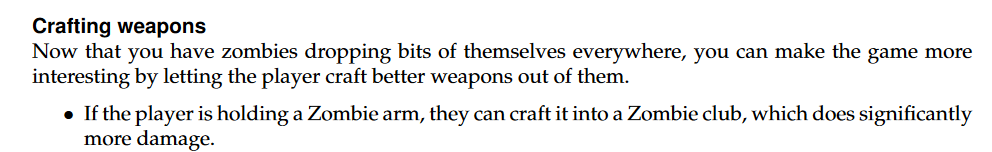
C:\Users\cedri\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\1ABAE590.tmp

First, we can handle the drop limb logic in the same section as we handle the losing of the limbs. Ie, in the takeDamage section in the zombie class.

We will need to call the addItem method in the Location class, which is an attribute of the GameMap class, which we have a reference to in the playturn method. This may need to be linked to the takeDamage method in the zombie class.



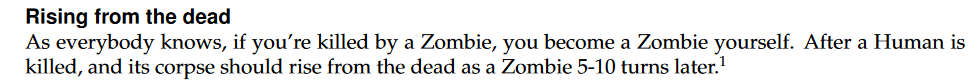
Create new classes of weapons



We need to create new class called CraftAction

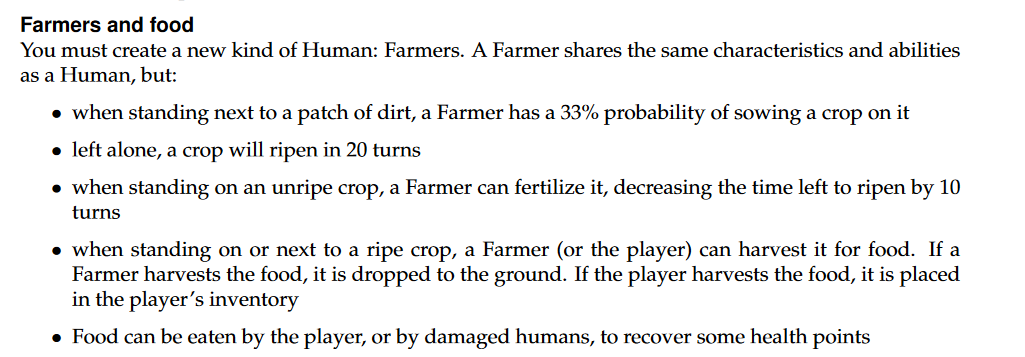
Store all the item associations in CraftAction in a hashmap, mapping a parent weapon to a child weapon. In the execute section write the code to handle the actual addition and removal of the item from the player's inventory.

Create these new weapon classes.



**ASK QUESTION ON FORUMS REGARDING THE TRANPORT OF CORPSES**

Create a new class called Corpse that experiences the passage of time. Give this Corpse a random value between 5-10 on initialisation. Overwrite the tick() class (in Item) to decrement the counter every turn. Create a new Zombie when this counter hits zero, and destroy the corpse.



Create new class called Farmer, which extends human. Add unripe crop as extension of ground, as well as ripe crop as an extension of ground.

Create a new class called SowAction, HarvestAction and FertiliseAction

Create a class called FarmingBehaviour

Create a new class called Food.

Create a new classed called ConsumeFoodAction