Assignment 1

The Human class initially has a list of behaviour similar to the zombie class. The behaviour to be included will be HarvestBehaviour, EatBehaviour and WanderBehaviour.

The HarvestBehaviour will always check if there is a ripe Crop on the Ground they are standing on and will return a HarvestAction which will harvest the Crop and convert it into a Food. The Food will be immediately dropped on to the coordinates and binded with a EatAction.

The EatBehaviour will check if the Human has received any damage. If the Human has received damage , the Human will always check if there is any Food on the Ground and only proceed to checking inventory. If there is a Food on the ground, Human will always call the EatAction binded to the Food.Else, Human will check for Food in the inventory and call Eat Action. The Human will restore health after executing the EatAction.

If the Human healthpoint is less than or equals 0, then the Human will become a corpse that will take 5 to 10 turns to turn into a zombie. Actor's isConcious function will be used to check if the Human is still alive.A Corpse will be created at the turn at the location of Human. We will use a number generator to decide which turn to rise and turn into a zombie . We will use the tick function in the Corpse to check if the turn is matched.

Player will have an extra CraftBehaviour that allows them to craft items such as Hand items and Leg items into weapons that deal higher damage .The CraftBehaviour will check for Item(Hand/Leg) at adjacent location and call the CraftAction which is binded when the limbs dropped. Hand can be further crafted into a ZombieClub , Leg can be crafted into a ZombieMace.

The abstract Farmer class inherits human class.This new farmer class has an inventory of size 0 and the food will be picked up and consumed if they are damaged. The farmer can only pick up food if it has taken damage. The Farmer class also has two extra behaviours, SowBehaviour and FertiliseBehaviour.

The SowBehaviour has a 33% chance to determine whether the Farmer will plant the crop on any adjacent dirt but not the patch of dirt they stand on. If it happens, the SowBehaviour returns a SowAction that takes in the coordinates of the dirt where the Crop will be planted and ripe in 20 turns.

The FertiliseBehaviour allows the farmer to fertilise an unripe crop if the Farmer is standing on top of the unripe crop and decreases the time left to ripen by 10 turns.The FertilizeAction will take in the coordinate of the crop that is to be fertilize and reduce the turn to ripe by 10 turns.If the turn to ripe is lesser than 10 turns, the crop will be immediately ripens.

Zombie class will have two more behaviours added. These behaviours are the PickUpBehaviour and the SpeakBehaviour. PickUpBehaviour will allow the zombie to pick up a WeaponItem when the Zombie is standing on that location. SpeakBehaviour will have a 10% of return SpeakActon. Zombie will only have an inventory of size 1, which means they can only pick up a Weapon and use it but they can never craft a Weapon. Bite is another intrinsic weapon besides punch of the Zombie. Bite is integrated into the AttackAction of the Zombie and has a different miss chance than punch. Zombie will contain 2 hands and legs which have a 25% of dropping one of them when they take damage. If they lose an arm they 50% of dropping the weapon if they are holding any and the probability of punching is halved. If they lose a leg, their movement speed is halved, moving only every two turns, if they lose both legs then they won't be able to move but still be able to attack. The limbs will drop on the adjacent ground of the zombie ,a CraftAction will be binded to the limbs.

Abstract class is created to follow the design principle of DRY(Do not repeat yourself). For example, Player and Farmer are extended from Human because they have some common characteristics and abilities so instead of having two different classes of Player and Farmer. We create an abstract class Human and contains the common characteristics and abilities and we can further extend the game later in time.

Attributes are declared private to prevent privacy leaks, setters and getters will be made to allow other objects to obtain an attribute from it. This design follows the design principle of Privacy Leaks

Assignment 2

The Human class now only will have two behaviours which are EatBehaviour and WanderBehaviour, at each turn will randomly select a behaviour to act.

EatBehaviour will check if the Human is damaged but Player will override the method of eat , so the Player can eat without getting damaged.

The Human will no longer have HarvestBehaviour instead Farmer and Player will have it, if a Farmer harvest the crop, the food is dropped on the ground , If a Player harvest the crop, the food is added into the Player’s inventory.

Crop now extends from ground instead of item . This is easier as it won't be recognised as an item and prevents the Farmer planting two Crop Objects on the same patch of Dirt.

The CraftBehaviour for Player now no longer exists, since the CraftAction is binded to the Hand/Leg and it will be handled by the playTurn Actions.Liskov substitution Principle is used in the hand/leg

Farmer will still have an inventory but won’t be used

Bite will be categorised as an IntrinsicWeapon and will be included in the AttackAction of the Zombie. We also introduce a PickUpBehaviour for the zombie to allow Zombie to pick up a Weapon from the ground.

The InfectBehaviour was not introduced into the game as it was deemed redundant, if Actor health falls below 0, it will be removed from the map and a corpse will be created at that position, using the tick to check the turn and let us spawn a Zombie at the location .

CraftAction is added into the allowableAction of the Hand/Leg upon drop,same goes to the HarvestAction is added upon Crop is ripe.EatAction is added upon the Food created.

Open-closed principle is applied in our design, all Actors will be able to harvest , craft, eat ,etc by just overriding a function that is already in the super type (Zombie Actor), So they are open to extension.

Liskov Substitution Principle is also used in our design as we are passing Item/Ground as argument instead of Hand/Leg etc and let polymorphism handle the input.

To prevent down casting methods was implemented in the ActorInterface that can be overridden in any sub classes extending from Actor. This is important as different sub classes have different variations of features we are implementing such as eat, craft, harvest, etc. Some methods were also added in the ItemInterface to prevent down casting ,such as craft and healing effect.

Assignment 3

Moving between maps is done by using MoveActorActionwith two references to two game maps. By adding MoveActorAction to the Vehicle allows the player to move between maps, moving to another map the actor and item in the map will still continue acting. This is done by the engine code in World.java.

We created another ZombieGameMap which extends from the GameMap in engine, By changing the tick function we check if the MamboMarie is conscious and the map doesn't contain the MamboMarie then we spawn the MamboMarie. The MamboMarie will remove itself from the GameMap when it reaches 30 turn.
We are tracking all the turn the MamboMarie was in the world , and simply use a modulo function to do ChantingAction,which will spawn 5 zombies at random location

QuitGame feature is implemented by removing the player from the game, the one that lets the player choose to quit is really easy as we can just remove the player in playTurn function , the tricky part is the one that indicates the player win/lose. We implemented both by extending the original World class and creating a new GameWorld that extends from World on our own. By overriding the processActorTurn & stillRunning method,we are able to tell if the player wins / loses. Iterating over all the actorLocation and checking if they are conscious .Also we added the processActorTurn's actions in GameWorld to include the ground's allowableAction.

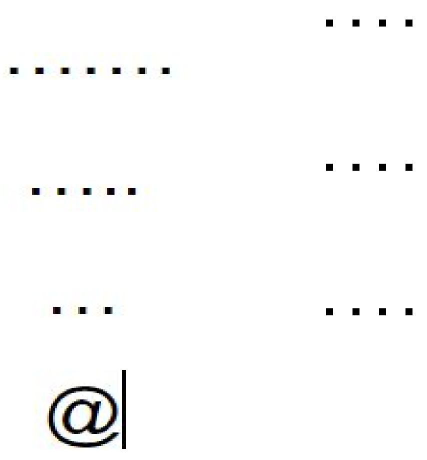
Sniper and Shotgun both extend from WeaponGuns class . Both of them have their own choose action(chooseShotgunAction & chooseSniperAction). Both of them also have their own attack action. Shotgun affects a cone shaped area from the player(shown below) . Sniper can be used to shoot any target in a range of 10 spaces from the player .

Ammunition extends from portableItem and has a reloadAction that allows the player to reload both ranged weapons.

Each item/actor is represented with a different char that is shown in the table below .

Inbuilt Actor/Item	Display Char	Add-on Actor/Item	Display Char
Player	@	Corpse	D
Zombie	Z	Crop	c
Human	H	Food	f
Farmer	F	Hand	y
Fence	#	Leg	w
Dirt	.	ZombieClub	Y
Plank	)	ZombieMace	W
Tree	t/T	MamboMarie	P
		Sniper	S
		Shotgun	s
		Ammunition	a
		Vehicle	V

<u>Shotgun affected area</u>



. . . @