Requirement 1 Class Breakdowns

Engine Classes

1) Location
   a. The location object is used to represent a position on the game map using (x, y) coordinates. In requirement 1, location objects are often passed into other objects as an argument in order to facilitate the spawning of enemies at specific locations or simply to determine the location of a specific actor, ground or item. Locations have an attribute of the ground class.

2) Ground (Abstract)
   a. Ground is an abstract class that is meant to represent terrain objects that exist at a location. Its inheritors can define unique properties for the terrain that effect the player and other game entities. Ground objects have a list of allowable actions that they provide to nearby actors and can experience the joys of time through the use of the tick method.

3) Actor (Abstract)
   a. The actor abstract class exists to provide a template for what a character in the game should be able to do. Both the player and all NPCs inherit from this class and will therefore be able to take damage, have an inventory and play turns through the use of the playTurn() method. Actor objects have a list of allowable actions that they provide to nearby actors.

4) Action (Abstract)
   a. Action objects are meant to represent actions that can be executed by actors such as attacking, picking up items, moving etc. This is achieved by calling the execute() method of the action object which will then execute some code specific to the action. Actions are mostly triggered one per turn by each actor in the game loop, however this is not a strict rule and actions can technically be triggered anywhere at any time within the codebase.

5) Weapon (Interface)
   a. Interface meant to define the capabilities of a weapon. Forces its inheritors to have a damage, verb and chance to hit. Also contains default method getSkill() which returns an action object meant to represent the weapons skill. This default method should be overridden as it currently returns a doNothing action.

6) IntrinsicWeapon (Implements Weapon)
   a. Intrinsic weapons are meant to represent weapons that are intrinsic to the actor such as the actor's hands, claws, fangs, etc.

7) WeaponItem (Abstract) (Extends Item) (Implements Weapon)
   a. WeaponItems are items that have the capabilities of weapons. Therefore they can have a damage and chance to hit stat and can be used in combat. They can also be dropped and picked up freely as they extend the item class.

Requirement 1 Class Breakdowns

Game Classes

Package: environments

1) Spawner (Abstract) (Extends Ground)
   a. Abstract class meant to represent a ground object with spawning capabilities. Children of this class will have an overridden tick() method that allows them to spawn enemy objects through the use of static methods from the EnemyFactory class. Spawners wont spawn anything if something is standing in top of them.

2) Graveyard (Extends Spawner)
   a. A spawner ground that has a 27% chance to spawn a HeavySkeletalSwordsman every turn if it is on the west side of the map, or a SkeletalBandit if it is on the east side of the map.

3) GustOfWind (Extends Spawner)
   a. A spawner ground that has a 33% chance to spawn a LoneWolf every turn if it is on the west side of the map, or a 4% chance to spawn a GiantDog on the east side of the map.

4) PuddleOfWater (Extends Spawner)
   a. A spawner ground that has a 2% chance to spawn GiantCrab every turn if it is on the west side of the map, or a 1% chance to spawn a GiantCrayfish if it is on the east side of the map.

Package: enemies

5) EnemyFactory
   a. Factory class used by multiple objects to create and return new enemies. Most of the methods in this class will be static and can be called without creating n instance of this class.

6) Mortal (Abstract) (Extends Actor)
   a. Abstract class that extends actor while implementing support for death actions. Meant to provide a template for actors who have the capacity to die and perform a death action. Has getDeathAction() abstract method.

7) Enemy (Abstract) (Extends Mortal)
   a. Abstract class meant to provide a template for enemy/hostile actors. Provides some additional logic and methods to facilitate their role in the game such as the use of Behaviour objects in their playTurn() method as a form of AI. All enemies belong to a species.

Requirement 1 Class Breakdowns

8) PileOfBones (Extends Enemy)
   a. Enemy that is spawned by the SkeletalHeavySwordsman on death. After 3 turns, if this enemy has not been destroyed/killed, it will call the EnemyFactory and spawn a new SkeletalHeavySwordsman. Drops Grossmesser on death. Skeletal type.

9) LoneWolf (Extends Enemy)
   a. Hostile creature that attacks the player with its intrinsic weapon. Canine type

10) GiantEnemy (Abstract) (Extends Enemy) (Implements Skilled)
   a. Represents a class of enemies who use the Slam SkillAction. Implements the skilled interface and therefore implements the getSkillAction() method which returns its associated skill action.

11) GiantCrab (Extends GiantEnemy)
   a. Hostile creature that attacks the player with its intrinsic weapon. Is a child of the giant enemy class and hence can use slam. Crustacean type.

12) GiantDog (Extends GiantEnemy)
   a. Hostile creature that attacks the player with its intrinsic weapon. Is a child of the giant enemy class and hence can use slam. Canine type.

13) GiantCrayfish (Extends GiantEnemy)
   a. Hostile creature that attacks the player with its intrinsic weapon. Is a child of the giant enemy class and hence can use slam. Crustacean type.

14) SkeletalEnemy (Abstract) (Extends Enemy)
   a. Represents a class of enemies who use the SkeletalDeathAction DeathAction. Hence, upon death, children of this class will execute the skeletal death action instead of the standard death action.

15) SkeletalBandit (Extends SkeletalEnemy)
   a. Hostile creature that attacks player with a WeaponItem called Scimitar. Since Scimitar is a child of the Sword class, it is possible for this enemy to use its weapon skill action, SpinningAttack. Skeletal type.

16) HeavySkeletalSwordsman (Extends SkeletalEnemy)
   a. Hostile creature that attacks player with a WeaponItem called Grossmesser. Since Grossmesser is a child of the Sword class, it is possible for this enemy to use its weapon skill action, SpinningAttack. Skeletal type.

Requirement 1 Class Breakdowns

Package: behaviours

17) Behaviour (Interface)
   a. An interface for behaviour objects. Behaviours just act as wrappers for action objects, they use conditional logic to determine what action to return or to return null. Enemies use this class as a form of AI and will progressively move down the list of behaviours until an action is returned.

18) SkillBehaviour (Implements Behaviour)
   a. Returns a SkillAction by using the skilled enemy's getSkillAction() method. Random logic is used here to determine if the skill or null is returned. Attack behaviour is performed after this. Not all enemies have this behaviour.

19) AttackBehaviour (Implements Behaviour)
   a. Returns an AttackAction if the player or any other enemies of a different species are close (one tile away), will return null otherwise. Follow behaviour is performed after this.

20) FollowBehaviour (Implements Behaviour)
   a. Returns a MoveAction if the player is close, will move in the direction of the player to follow them, if the player is not close it will simply return null. Wander behaviour is performed after this.

21) WanderBehaviour (Implements Behaviour)
   a. Returns a MoveAction, will move in a valid random direction. Always returns a move action.

Package: gameactions -> deathactions

22) DeathAction (Abstract) (Extends Action)
   a. A class of actions executed upon the death of its owner. Can be executed by other actions such as at the end of an AttackAction if the target is killed.

23) SkeletonDeathAction (Extends DeathAction)
   a. Death action returned by skeletal enemies. Will call the EnemyFactory to spawn a new PileOfBones.

24) StandardDeathAction (Extends DeathAction)
   a. The standard death action of most mortal enemies. Simply causes the enemy to drop its inventory weapons and runes upon death.

Requirement 1 Class Breakdowns

Package: gameactions -> skillactions

25) Skilled (Interface)
   a. Interface implemented by enemies and weapons that possess special offensive or non-offensive skills. Forces the implementation of the getSkillAction() method.

26) SkillAction (Abstract) (Extends Action)
   a. A class of actions that represent special skills that can be performed by actors to heal, damage, etc.

27) Slam (Extends SkillAction)
   a. Causes the actor to attack all actors around it (8 surrounding tiles) for intrinsic weapon damage and accuracy.

28) SpinningAttack (Extends SkillAction)
   a. Held by weapons. Causes the actor to attack all actors around it for the WeaponItem's damage and accuracy.

Package: gameactions -> standardactions

29) AttackAction (Extends Action)
   a. Initiates an attack on another actor using a WeaponItem or IntrinsicWeapon.

Package: gameweapons

30) Sword (Abstract) (Extends WeaponItem) (Implements Skilled)
   a. A class of WeaponItems who use the SpinningAttack skill action.

31) Grossmesser (Extends Sword)
   a. A child of the Sword class. Commonly held by HeavySkeletalSwordsman enemy.

32) Scimitar (Extends Sword)
   a. A child of the Sword class. Commonly held by SkeletonBandits.

Package: enums

1) Species (Enumeration)
   a. A list of all enemy species that are in the game such as skeletal, canine crustacean etc. Used during the attack behaviour to determine if an actor should be attacked or not.

Requirement 1 Class Breakdowns

2) Species (Enumeration)
   a. A list of all enemy species that are in the game such as skeletal, canine crustacean etc. Used during the attack behaviour to determine if an actor should be attacked or not.

Package: trader

3) Trader (Extends Actor)
   a. A class representing a trader that the player can buy and sell weapons from.