

Room

- description : String
- north : Room
- south : Room
- east : Room
- west : Room
- monsterHere : Monster
- hasMonster : boolean
- potionHere : Potion
- hasPotion : boolean
- randomGenerator : Random
- isBossRoom : boolean

«Constructor» Room(description : String)
«Constructor» Room(description : String, isBossRoom : boolean)

+ getDescription() : String
+ getNorth() : Room
+ getSouth() : Room
+ getEast() : Room
+ getWest() : Room
+ getExits() : String
+ getMonster() : Monster
+ hasMonster() : boolean
+ getPotion() : Potion
+ hasPotion() : boolean
+ setExits(north : Room, south : Room, east : Room, west : Room)
+ decideMonster(playerLevel : int)
- addMonster(playerLevel : int)
+ nullifyMonster()
+ decidePotion()
- addPotion()

Item

- name : String

«constructor» Item(name: String)
+ getName() String

Weapon

- attackBonus : int

«constructor» Weapon(name : String, attackBonus : int)
+ getAttackBonus() : int

Potion

- boosts : int[]

«constructor» Potion(name : String, boosts : int[])
+ getBoost() : int[]

PowerPotion

- NAME : String
- HEALTH_BOOST : int
- MANA_BOOST: int
- BOOSTS : int[]

«constructor» PowerPotion()

GameCharacter

- name : String
- health : int
- attackPower : int
- isAlive : boolean

«constructor» GameCharacter(name : String, health : int, attack : int)

+ getName() : String
+ getHealth() : int
+ getAttackPower() : int
+ isAlive() : boolean
+ setHealth(healthBoost : int)
+ setAttackPower(newAttack : int)
+ takeDamage(damage : int)
+ toString() : String
+ takeTurn(character GameCharacter)

Dungeon

- antechamber : Room
- tortureRoom : Room
- holdingCell : Room
- hall : Room
- chapel : Room
- passage1 : Room
- greenRoom : Room
- robeRoom : Room
- passage2 : Room
- altarRoom : Room
- statueRoom : Room

«constructor» Dungeon()
+ getRoom0() : Room

HomuncularPriest
<ul style="list-style-type: none"> - HOM_NAME : String - HOM_ATTACK : int - HOM_XP : int - randomGenerator : Random
<ul style="list-style-type: none"> «constructor» HomuncularPriest() + attack(player : Player)

Asenath
<ul style="list-style-type: none"> - instance : Asenath - AS_NAME : String - AS_HEALTH : int - randomGenerator : Random
<ul style="list-style-type: none"> - «constructor» Asenath() + getInstance() : Asenath + attack(player : Player) - lifeStealAttack(player : Player) - manaBurnAttack(player : Player) - levelBasedAttack(player : Player)

AncientOne
<ul style="list-style-type: none"> - ANC_NAME : String - ANC_ATTACK : int - ANC_XP : int - randomGenerator : Random - isManaAttacking : boolean
<ul style="list-style-type: none"> «constructor» AncientOne() + attack(player : Player) - manaAttack(player : Player) - physicallyAttack(player : Player) + takeDamage(damage : int)

Amphiboid
<ul style="list-style-type: none"> - AMPH_NAME : String - AMPH_ATTACK : int - AMPH_XP : int - randomGenerator : Random
<ul style="list-style-type: none"> «constructor» Amphiboid() + attack(player : Player) - printFailedAttackMessage()

HealthPotion
<ul style="list-style-type: none"> - NAME : String - HEALTH_BOOST : int - MANA_BOOST : int - BOOSTS : int[]

ManaPotion
<ul style="list-style-type: none"> - NAME : String - HEALTH_BOOST : int - MANA_BOOST : int - BOOSTS : int[]

Monster
<ul style="list-style-type: none"> - xP : int - hasWeapon : boolean - carriedWeapon : Weapon - hasPotion : boolean - carriedPotion : Potion - randomGenerator : Random + monsterWeaponsCache : ArrayList<Weapon>
<ul style="list-style-type: none"> «constructor» Monster(name : String, health : int, attackPower : int, xP : int) + getXP() : int] + getWeapon() : Weapon + getPotion() : Potion - decideWeapon() - addWeapon() - decidePotion() - addPotion() + takeTurn(player : GameCharacter) + attack(player : Player)

Player
<ul style="list-style-type: none"> - mana : int - level : int - xPUncspent : int - currentWeapon : Weapon - isMagicAttacking : boolean - BASIC_SWORD : Weapon - potionInventory : HashMap<String, Integer>
<ul style="list-style-type: none"> «constructor» Player(name : String) + getLevel() : int + getMana() : mana + getWeapon() : Weapon + hasPotionType(potionType: String) : boolean + getPotionInventory() : String + acquireXP(xPAcquired : int) - levelUp() - printLevelUpMessage() + addToPotionInventory(potion : Potion) + equipWeapon(newWeapon : Weapon) + storeWeapon() + takeTurn(monster : GameCharacter) + setAttackStyle(choiceOfAttack : boolean) - castSpell(monster : GameCharacter) + drinkPotion(potion : Potion) + takeManaBurn(burn : int) + attack(monster : GameCharacter) + toString() : String

GameRunner

- gameOver : boolean
- input : Scanner
- dungeon : Dungeon
- currentRoom : Room
- player : Player

+ main(args : String[])
- printIntro()
- printInstructions()
- nameHero() : String
- printInitialTurnOptions() : boolean
- usePotionOptions()
- printPotionOptions()
- getPotionChoice() : Potion
- tryPotionsAgain() : Potion
- postPotionUpdate(potionDrank : Potion)
- printOpponent(monster : Monster)
- printCombatOptions(monster : Monster) : boolean
- getCombatChoice(monster : Monster) : boolean
- printCombatUpdate(attacker : GameCharacter, defender : GameCharacter)
- plunderCorpse(monster : Monster)
- printPlunderString(amountOfPlunder : int, lootedWeapon : Weapon,
 lootedPotion : Potion, monster : Monster)
- canWeaponBeEquipped(lootedWeapon : Weapon) : boolean
- handleRoomPotion(roomPotion : Potion)
- parseYesNoInput() : boolean
- handleMovement()
- getPlayerChoice(choice : String)
- exitRoom(choice : String)
- printGameLostMessage(monster : Monster)
- printGameWonMessage()



