## - entities: List<Entity> - dungeonMap: int[[[] - adjMap: int[[[] - inventory: List<CollectableEntity> - buildables: List<String> - dungeonName: String - dungeonId: String - dungeonGoals: String - keyCounter: int - keyStatus: boolean - tickCounter: int - ncxCounter: int - entityCounter: int - peaceful: boolean - hard: boolean - invisibilityPotionCounter: int - invincibilityPotionCounter: int Dungeon(String, String, String) + getEntityCounter(): int + setEntityCounter(int): void + getInvincibilityPotionCounter(): int + setInvincibilityPotionCounter(): int + setInvisibilityPotionCounter(int): void + getTickCounter(): int + setTickCounter(int): void + getDungeonGoals(): String + setDungeonGoals(String): void + getEntities(): List<Entity> + setEntities(List<Entity>): void + getDungeonName(): String + setDungeonName(): String + setDungeonId(): String - setMeyCounter(): int - setKeyCounter(int): void

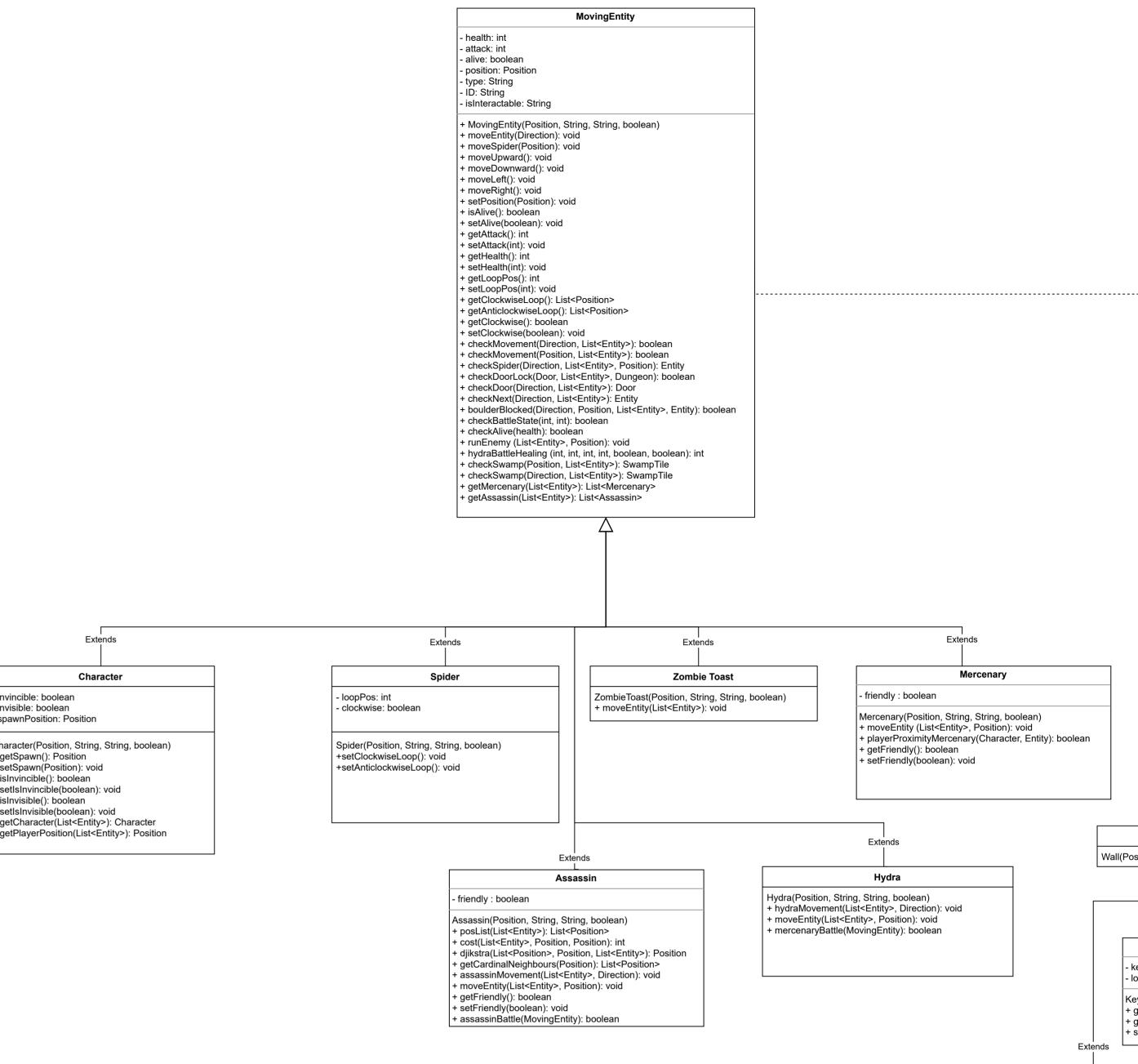
Dungeon

## DungeonManiaController

dungeons: List<Dungeons>
currDungeon: String
dungeonCounter: int
entityCounter: int
tickCounter: int

+ DungeonManiaController(): void + getSkin(): String + getLocalisation: String + getGameModes(): List<String> + dungeons(): List<String> + newGame(String, String): DungeonResponse + addEntitiesToList(String, Dungeon): void + getGoalsFromJson(String): String

		- invincible - invisible - spawnF  Characte + getSpa + setSpa + isInvine + setIsIn + isInvisi + setIsIn + getCha + getPla



## SwitchDoor - wireQueue: Queue<Wire> SwitchDoor(Position, String, String, bool + doorUnlock(): void + doorLock(): void + checkSwitchBoulder(Dungeon): boolea + boulderAndSwitch(List<Entity>): boole + checkWires(Dungeon): boolean + findWire(List<Entity>): Wire

