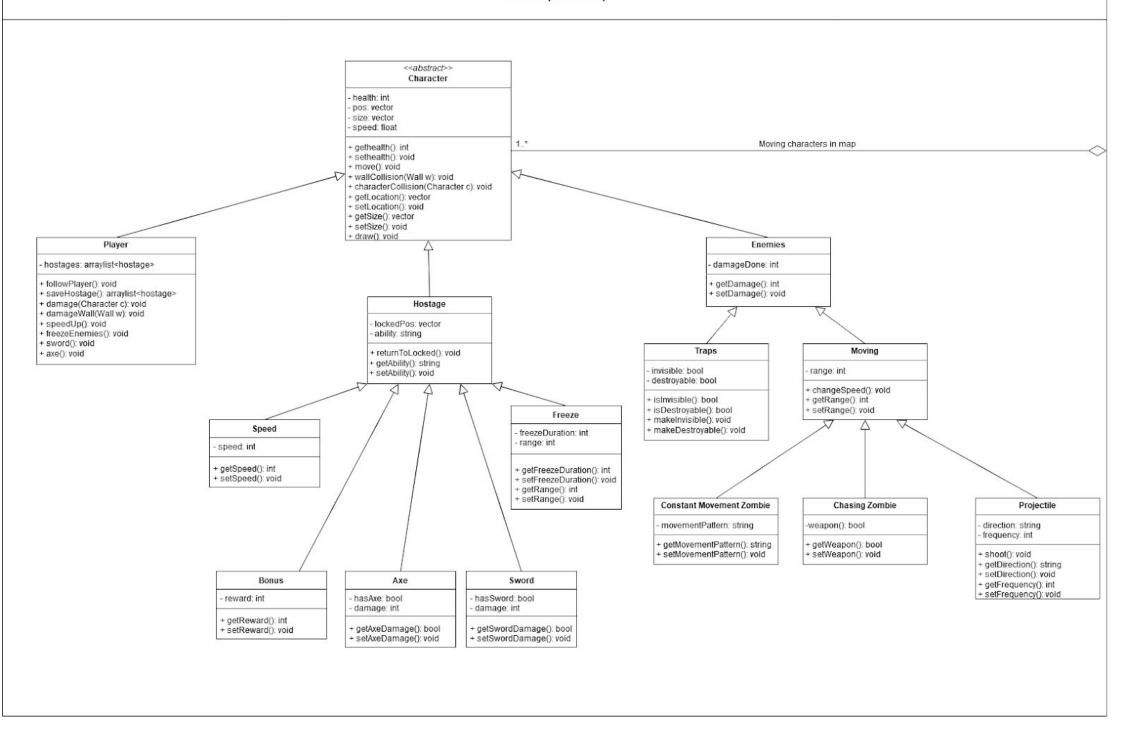
Classes (Character)



Classes (Map) Map walls: arraylist<wall> enemies: arratlist <enemy> size: vector unitSize: int unbreakableWallCount: int - breakableWallCount: int doorCount: int - movingEnemieyCount: int Hud trapCount: int hostageCount: int in-game information size: vector score: int time: float + draw(): void + update(): void running: boolean + achievements(): void + createStructure(): arraylist<wall> + createEnemies(): arraylist<enemy> + createHostages():arraylist <hostage> + destroyStructure(): arraylist<wall> + destroyEnemies(): arraylist<enemy> + destroyHostages() :arraylist <hostage> + draw(): void + startGame(): void + calculateScore(): int Map structure 1..* <<abstract>> Wall -size: vector -color: color -passable: bool -location: vector + getSize(): vector + setSize(): void + getColor(): color + setColor(): void + getPassable(): bool + setPassable(): void + getLocation(): vector + setLocation(): void Unbreakable boundaryWall: bool - hostageRequired: bool + getBoundaryWall: bool + setBoundaryWall: void + open(): void + getHostageRequired(): bool Breakable Door + setHostageRequired(): void durability: int keyRequired: bool destroy(): void open(): void getDurability(): void getKeyRequired(): bool setDurability(): void - setKeyRequired(): void