

Game
- in: Scanner - rand: Random - player: Player
+ main(args: String[]): void + Game() + createPlayer(): void + enterCombat(opponent: NPC): void

Character
- maxHealth: int - health: int - maxMana: int - mana: int - baseDamage: int - name: String - items: ArrayList<Consumable> - turnsVulnerable: int - turnsInvincible: int - turnsStunned: int - tempDamageBuff: double
+ Character(name: String, health: int, mana: int, damage: int) + getName(): String + getHealth(): int + getMaxHealth(): int + getMaxMana(): int + getMana(): int + getBaseDamage(): int + isAlive(): boolean + attack(other: Character): void + defend(other: Character): void + modifyHealth(modifier: int): void + setAsVulnerable(numTurns: int): void + isVulnerable(): boolean + decreaseTurnsVulnerable(): void + setAsInvincible(numTurns: int): void + isInvincible(): boolean + decreaseTurnsInvincible(): void + setAsStunned(numTurns: int): void + isStunned(): boolean + decreaseTurnsStunned(): void + setTempDamageBuff(buff: double): void + setMana(Tmana: int): void + obtain(item: Consumable): void + chargeUpMana(): void + castSpell(): void + useItem(owner: Character, other: Character, scanner: Scanner): void + hasItems(): boolean + <i>takeTurn(owner: Character, other: Character, scanner: Scanner): void</i>

Player
- (inherited from Character)
+ Player(name: String, health: int, mana: int, baseDamage: int) + takeTurn(other: Character): void

Consumable	
+ consume(owner: Character): void	
<div></div>	
<div></div>	<div></div>
HealingPotion	RagePotion
+ consume(owner: Character): void	+ consume(owner: Character): void
- calculateHealing(): int	