Character

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
+ obtain(item: Consumable): void
+ useItem(owner: Character, other: Character): void
+ hasItems(): boolean
+ *takeTurn(other: Character): void*

<<Interface>> Consumable

<<Interface>>
Consumable

+ consume(owner: Character): void

HealingPotion

HealingPotion

+ consume(owner: Character): void
- calculateHealing(): int

Game

Game

- player: Player
~ in: Scanner
~ opponent: NPC
- rand: Random

+ main(args: String[]): void
+ Game()
+ createPlayer(): void
+ enterCombat(opponent: NPC): void

Player

Player

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

NPC

NPC

+ NPC(name: String, health: int, mana: int, baseDamage: int)
+ takeTurn(other: Character): void