

MessageCallBack (Interface)
send(m: string): void

InputHelper
- regexUserInput: string
- regexMenu: string
+ playerGameInput(): InputProvider
+ menuInput(): char
- inputCatch(regex: string): char
- validateWithRegex(s: string, regex: string): boolean

Movement
+ gameBoard: Board
+ messageCallBack: MessageCallBack
+ movePlayer(inputProvider: InputProvider, player: Player): Tile
+ moveEnemy(inputProvider: InputProvider, enemy: Enemy): Tile
+ getNewPosition(inputProvider: InputProvider, position: Position): Position
+ validMove(position: Position): boolean

UnitsInRange
+ gameBoard: Board
+ enemiesInRange(player: Player, range: int): List<Enemy>

Board
+ height: int
+ width: int
- player: Player
- enemies: List<Enemy>
- board: string[][]
+ walls: HashMap<Position, Wall>
+ getTile(position: Position): Tile
+ buildBoard(file: File): void
+ buildArray(): void
+ remove(e: Enemy): void
+ arrayToString(): string
+ toString(): string

LevelManager
- board: Board
- countTicks: int
- messageCallBack: MessageCallBack
- filesOfLevels: List<File>
- unitList: List<Unit>
+ start(address: string): void
+ loadLevel(file: File): void
+ gameTick(): void
+ createListOfLevel(address: string): void
+ instruction(): void
- startLevel(): void
- getPlayerMenu(): void
+ printLevel(): void
+ printMenu(): void
+ printChosenPlayer(): void

UnitDB
+ playerPool: Map<string, Unit>
+ enemyPool: Map<string, Unit>

Tile
tile: char
position: Position
+ initialize(positon: Position): void
+ accept(player: Player): void
+ accept(enemy: Enemy): void
+ compareTo(tile: Tile): int
+ toString(): string

Empty
+ emptyTile="·": char
+ initialize(position: Position): void
+ accept(player: Player): void
+ accept(enemy: Enemy): void

Wall
+ wallTile="#": char
+ accept(player: Player): void
+ accept(enemy: Enemy): void

Unit
+ manager: LevelManager
r: Random
+ messageCallBack: MessageCallBack
name: String
health: Resource
attack: int
defense: int
+ isAlive: boolean
+ initialize(position: Position, messageCallBack: MessageCallBack): void
attack(): int
+ defend(): int
~ onDeath(u: Unit): void
~ accept(e: Enemy): void
~ accept(p: Player): void
~ onKill(e: Enemy): void
~ onKill(p: Player): void
~ gameTick(player: Player): void
~ copy(): Unit
battle(u: Unit): void
+ describe(): string

Player
+ playerTile="@": char
experience: int
expReq=50: int
attackBonus=4: int
defenseBonus=1: int
healthBonus=10: int
level: int
+ accept(e: Enemy): void
+ accept(p: Player): void
+ gameTick(player: Player): void
+ battle(u: Unit): void
+ onDeath(u: Unit): void
+ onKill(e: Enemy): void
+ onKill(p: Player): void
addExperience(experienceGained: int): void
levelUp(): void
+ toString(): string
gainHealth(): int
gainAttack(): int
gainDefense(): int
- levelUpRequirement(): int
+ describe(): string
~ castSpecialAbility(): void

Enemy
- experience: int
+ onDeath(u: Unit): void
+ onKill(e: Enemy): void
+ onKill(p: Player): void
+ battle(u: Unit): void
~ accept(e: Enemy): void
~ accept(p: Player): void

Monster
- visionRange: int
+ gameTick(player: Player): void
+ accept(e: Enemy): void
+ accept(p: Player): void
+ onDeath(u: Unit): void
+ onKill(e: Enemy): void
+ onKill(p: Player): void
+ describe(): string
+ copy(): Monster

Trap
+ visibiltyTime: int
+ invisibiltyTime: int
+ ticksCount: int
+ visible: boolean
+ gameTick(player: Player): void
+ accept(e: Enemy): void
+ accept(p: Player): void
+ onDeath(u: Unit): void
+ onKill(e: Enemy): void
+ onKill(p: Player): void
+ copy(): Monster

Warrior
- abCoolDown: int
- remCoolDown: int
- abilityRange=3: int
+ specialAbility="Avenger's Shield": string
levelUp(): void
+ gameTick(player: Player): void
+ castSpecialAbility(): void
+ describe(): string
+ copy(): Warrior

Mage
- mana: Resource
- manaCost: int
- spellPower: int
- hitsCount: int
- abilityRange: int
+ specialAbility="Blizzard": string
levelUp(): void
+ gameTick(player: Player): void
+ castSpecialAbility(): void
+ describe(): string
+ copy(): Mage

Rogue
- cost: int
- currEnergy: Resource
- abilityRange=2: int
- maxEnergy=100: int
+ specialAbility="Fan Of Knives": string
levelUp(): void
+ gameTick(player: Player): void
+ castSpecialAbility(): void
+ describe(): string
+ copy(): Mage

Dictionary:
- private
+ public
protected
~ abstract

Position
- x: int
- y: int
- up: Position
- down: Position
- right: Position
- left: Position
+ initialize(): void
+ compareTo(pos: Position): int
+ range(pos: Position): double
+ inRange(position: Position, range: int): boolean

Resource
- pool: int
- amount: int
+ increaseAmount(added: int): void
+ reduceAmount(reduce: int): boolean
+ addCapacity(gained: int): void
+ restore(): void

InputProvider (enum)
Up("w"), Down("s"), Left("a"), Right("d"), CastSpecialAbility("e"), Nothing("q")
- key: string
+ getRegex(): string
+ findByKey(key: string): InputProvider