CrazyPiece

idPeca: int idTipePeca: int idEquipa: int alcunha: String coordenadaX: int coordenadaY: int captured: boolean

CrazyPiece(idPeca: String, idTipoPeca: String, idEquipa: String, alcunha: String, coordenadaX: int, coordenadaY: int)

CrazyPiece(idPeca: int, idTipoPeca: int, idEquipa: int, alcunha: String, coordenadaX: int, coordenadaY: int)

getImagePNG(): String

getId(): int

getIdTipoPeca(): int getIdEquipa(): int getAlcunha(): String getCoordenadaX: int getCoordenadaY: int isCaptured(): boolean setCoordenadaX(): void setCoordenadaY(): void toString(): String



## Simulador

crazyList: List<CrazyPiece>

turnoCaptura: int turno: int

equipaAJogar: int jodaInvalidaBrancas: Integer

jodalnvalidaBrancas: Integer jodaValidaPretras: Int jodaValidaBrancas: int jodaValidaBrancas: int numeroPecas: int

capturas: List<CrazyPiece> listaDocumentos: List<String> autores: List<String> resultadoFinal: List<String>

Simulador()

Simulador(crazyList: List<CrazyPiece>, turno: int, equipaAJogar: int, capturas: List<CrazyPiece>)

iniciaJogo(ficheiroInicial: File): boolean

stringTest(peca: List<Lis<String>>, mapas: List<List<String>>): void

numberInt(number String): int

findCapture(x: int, y: int, i:D int, equipa: int): void findFriend(x: int, y: int, iD: int, equipa: int): boolean

removeCrazyList(): void invalidoBranco(): void invalidoPreto(): void jogoTerminado: boolean

contarPecasCapturadas(equipa: int): int

getTamanhoTabuleiro(): int

processaJogada(xO: int, yO: int, xD int, yD int ): boolean

getPecasMalucas(): List<CrazyPiece>

jogoTerminado(): boolean getAutores(): List<String> getIdPeca(int x, int y): int getIDPeca(x: int, y: int): int getIDEquipaAJogar(): int getResultados(): List<String> getCrazyList(): List<CrazyPiece> getTurno(): int

getCapturas(): List<CrazyPiece>