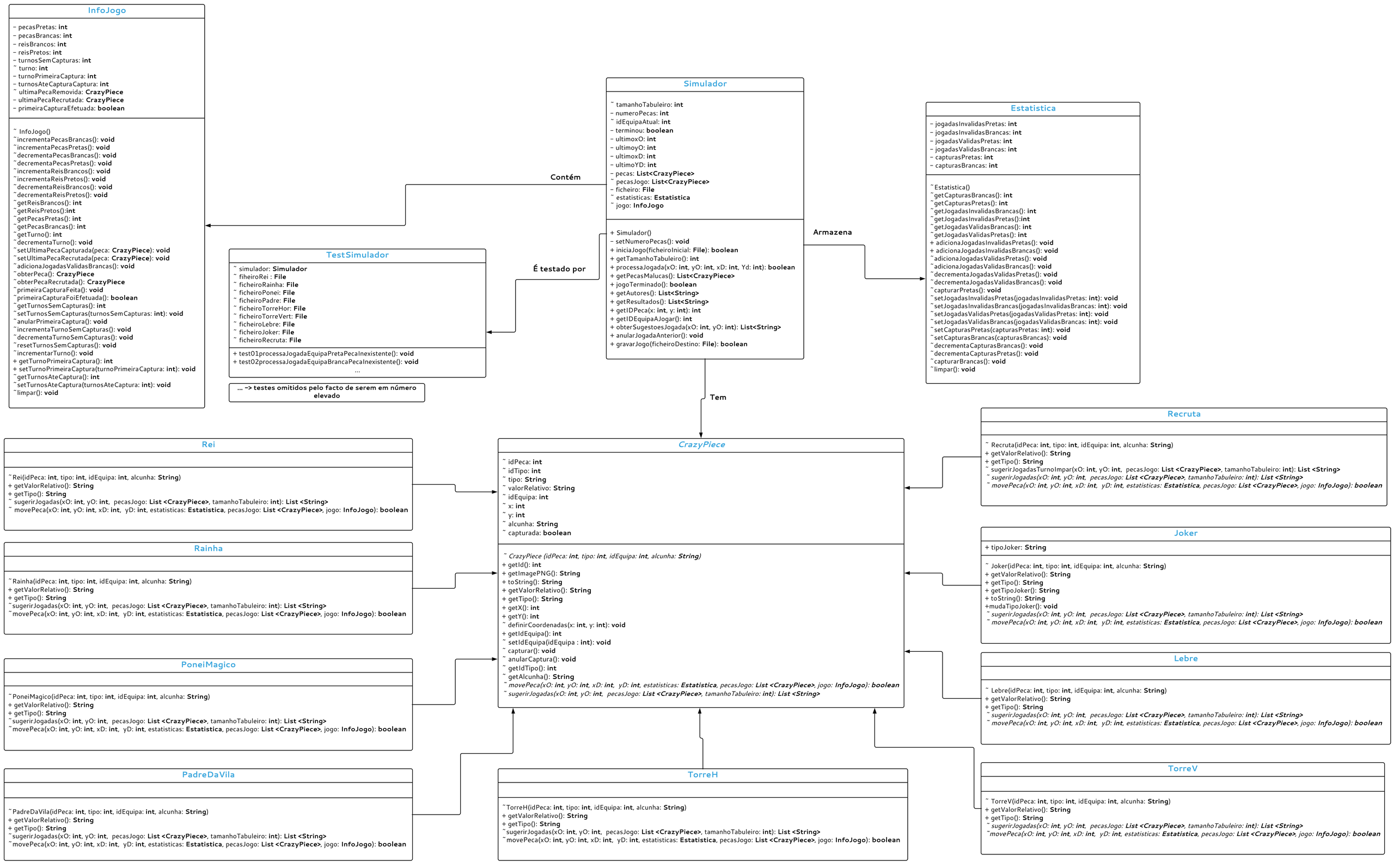
InfoJogo
- pecasPretas: int
- pecasBrancas: int
- reisBrancos: int
- reisPretos: int
- turnosSemCapturas: int
~ turno: int
- turnoPrimeiraCaptura: int
- turnosAteCapturaCaptura: int
~ ultimaPecaRemovida: CrazyPiece
- ultimaPecaRecrutada: CrazyPiece
- primeiraCapturaEfetuada: boolean
~ InfoJogo()
~incrementaPecasBrancas(): void
~incrementaPecasPretas(): void
~decrementaPecasBrancas(): void
~decrementaPecasPretas(): void
~incrementaReisBrancos(): void
~incrementaReisPretos(): void
~decrementaReisBrancos(): void
~decrementaReisPretos(): void
~getReisBrancos(): int
~getReisPretos():int
~getPecasPretas(): int
~getPecasBrancas(): int
~getTurno(): int
~decrementaTurno(): void
~setUltimaPecaCapturada(peca: CrazyPiece): void
~setUltimaPecaRecrutada(peca: CrazyPiece): void
~adicionaJogadasValidasBrancas(): void
~obterPeca(): CrazyPiece
~obterPecaRecrutada(): CrazyPiece
~primeiraCapturaFeita(): void
~primeiraCapturaFoiEfetuada(): boolean
~getTurnosSemCapturas(): int
~setTurnosSemCapturas(turnosSemCapturas: int): void
~anularPrimeiraCaptura(): void
~incrementaTurnoSemCapturas(): void
~decrementaTurnoSemCapturas(): void
~resetTurnosSemCapturas(): void
~incrementarTurno(): void
+ getTurnoPrimeiraCaptura(): int
+ setTurnoPrimeiraCaptura(turnoPrimeiraCaptura: int): void
~getTurnosAteCaptura(): int
~setTurnosAteCaptura(turnosAteCaptura: int): void
~limpar(): void
Contém
Simulador
~ tamanhoTabuleiro: int
- numeroPecas: int
~ idEquipaAtual: int
- terminou: boolean
- ultimoxO: int
- ultimoyO: int
- ultimoxD: int
- ultimoYD: int
- pecas: List<CrazyPiece>
~ pecasJogo: List<CrazyPiece>
- ficheiro: File
~ estatisticas: Estatistica
~ jogo: InfoJogo
+ Simulador()
- setNumeroPecas(): void
+ iniciaJogo(ficheiroInicial: File): boolean
+ getTamanhoTabuleiro(): int
+ processaJogada(xO: int, yO: int, xD: int, Yd: int): boolean
+ getPecasMalucas(): List<CrazyPiece>
+ jogoTerminado(): boolean
+ getAutores(): List<String>
+ getResultados(): List<String>
+ getIDPeca(x: int, y: int): int
+ getIDEquipaAJogar(): int
+ obterSugestoesJogada(xO: int, yO: int): List<String>
+ anularJogadaAnterior(): void
+ gravarJogo(ficheiroDestino: File): boolean
Armazena
Estatistica
- jogadasInvalidasPretas: int
- jogadasInvalidasBrancas: int
- jogadasValidasPretas: int
- jogadasValidasBrancas: int
- capturasPretas: int
- capturasBrancas: int
~Estatistica()
~getCapturasBrancas(): int
~getCapturasPretas(): int
~getJogadasInvalidasBrancas(): int
~getJogadasInvalidasPretas():int
~getJogadasValidasBrancas(): int
~getJogadasValidasPretas(): int
+ adicionaJogadasInvalidasPretas(): void
+ adicionaJogadasInvalidasBrancas(): void
~adicionaJogadasValidasPretas(): void
~adicionaJogadasValidasBrancas(): void
~decrementaJogadasValidasPretas(): void
~decrementaJogadasValidasBrancas(): void
~capturarPretas(): void
~setJogadasInvalidasPretas(jogadasInvalidasPretas: int): void
~setJogadasInvalidasBrancas(jogadasInvalidasBrancas: int): void
~setJogadasValidasPretas(jogadasValidasPretas: int): void
~setJogadasValidasBrancas(jogadasValidasBrancas: int): void
~setCapturasPretas(capturasPretas: int): void
~setCapturasBrancas(capturasBrancas): void
~decrementaCapturasBrancas(): void
~decrementaCapturasPretas(): void
~capturarBrancas(): void
~limpar(): void
É testado por
TestSimulador
~ simulador: Simulador
~ fiheiroRei : File
~ ficheiroRainha: File
~ ficheiroPonei: File
~ ficheiroPadre: File
~ ficheiroTorreHor: File
~ ficheiroTorreVert: File
~ ficheiroLebre: File
~ ficheiroJoker: File
~ ficheiroRecruta: File
+ test01processaJogadaEquipaPretaPecaInexistente(): void
+ test02processaJogadaEquipaBrancaPecaInexistente(): void
...
... –> testes omitidos pelo facto de serem em número elevado
Tem
Rei
~Rei(idPeca: int, tipo: int, idEquipa: int, alcunha: String)
+ getValorRelativo(): String
+ getTipo(): String
~ sugerirJogadas(xO: int, yO: int, pecasJogo: List <CrazyPiece>, tamanhoTabuleiro: int): List <String>
~ movePeca(xO: int, yO: int, xD: int, yD: int, estatisticas: Estatistica, pecasJogo: List <CrazyPiece>, jogo: InfoJogo): boolean
Rainha
~Rainha(idPeca: int, tipo: int, idEquipa: int, alcunha: String)
+ getValorRelativo(): String
+ getTipo(): String
~sugerirJogadas(xO: int, yO: int, pecasJogo: List <CrazyPiece>, tamanhoTabuleiro: int): List <String>
~movePeca(xO: int, yO: int, xD: int, yD: int, estatisticas: Estatistica, pecasJogo: List <CrazyPiece>, jogo: InfoJogo): boolean
PoneiMagico
~PoneiMagico(idPeca: int, tipo: int, idEquipa: int, alcunha: String)
+ getValorRelativo(): String
+ getTipo(): String
~sugerirJogadas(xO: int, yO: int, pecasJogo: List <CrazyPiece>, tamanhoTabuleiro: int): List <String>
~movePeca(xO: int, yO: int, xD: int, yD: int, estatisticas: Estatistica, pecasJogo: List <CrazyPiece>, jogo: InfoJogo): boolean
PadreDaVila
~PadreDaVila(idPeca: int, tipo: int, idEquipa: int, alcunha: String)
+ getValorRelativo(): String
+ getTipo(): String
~sugerirJogadas(xO: int, yO: int, pecasJogo: List <CrazyPiece>, tamanhoTabuleiro: int): List <String>
~movePeca(xO: int, yO: int, xD: int, yD: int, estatisticas: Estatistica, pecasJogo: List <CrazyPiece>, jogo: InfoJogo): boolean
CrazyPiece
~ idPeca: int
~ idTipo: int
~ tipo: String
~ valorRelativo: String
~ idEquipa: int
~ x: int
~ y: int
~ alcunha: String
~ capturada: boolean
~ CrazyPiece (idPeca: int, tipo: int, idEquipa: int, alcunha: String)
+ getId(): int
+ getImagePNG(): String
+ toString(): String
+ getValorRelativo(): String
+ getTipo(): String
+ getX(): int
+ getY(): int
~ definirCoordenadas(x: int, y: int): void
+ getIdEquipa(): int
~ setIdEquipa(idEquipa : int): void
~ capturar(): void
~ anularCaptura(): void
~ getIdTipo(): int
~ getAlcunha(): String
~ movePeca(xO: int, yO: int, xD: int, yD: int, estatisticas: Estatistica, pecasJogo: List <CrazyPiece>, jogo: InfoJogo): boolean
~ sugerirJogadas(xO: int, yO: int, pecasJogo: List <CrazyPiece>, tamanhoTabuleiro: int): List <String>
TorreH
~TorreH(idPeca: int, tipo: int, idEquipa: int, alcunha: String)
+ getValorRelativo(): String
+ getTipo(): String
~sugerirJogadas(xO: int, yO: int, pecasJogo: List <CrazyPiece>, tamanhoTabuleiro: int): List <String>
~movePeca(xO: int, yO: int, xD: int, yD: int, estatisticas: Estatistica, pecasJogo: List <CrazyPiece>, jogo: InfoJogo): boolean
Recruta
~ Recruta(idPeca: int, tipo: int, idEquipa: int, alcunha: String)
+ getValorRelativo(): String
+ getTipo(): String
~ sugerirJogadasTurnoImpar(xO: int, yO: int, pecasJogo: List <CrazyPiece>, tamanhoTabuleiro: int): List <String>
~ sugerirJogadas(xO: int, yO: int, pecasJogo: List <CrazyPiece>, tamanhoTabuleiro: int): List <String>
~ movePeca(xO: int, yO: int, xD: int, yD: int, estatisticas: Estatistica, pecasJogo: List <CrazyPiece>, jogo: InfoJogo): boolean
Joker
+ tipoJoker: String
~ Joker(idPeca: int, tipo: int, idEquipa: int, alcunha: String)
+ getValorRelativo(): String
+ getTipo(): String
+ getTipoJoker(): String
+ toString(): String
+mudaTipoJoker(): void
~ sugerirJogadas(xO: int, yO: int, pecasJogo: List <CrazyPiece>, tamanhoTabuleiro: int): List <String>
~ movePeca(xO: int, yO: int, xD: int, yD: int, estatisticas: Estatistica, pecasJogo: List <CrazyPiece>, jogo: InfoJogo): boolean
Lebre
~ Lebre(idPeca: int, tipo: int, idEquipa: int, alcunha: String)
+ getValorRelativo(): String
+ getTipo(): String
~ sugerirJogadas(xO: int, yO: int, pecasJogo: List <CrazyPiece>, tamanhoTabuleiro: int): List <String>
~ movePeca(xO: int, yO: int, xD: int, yD: int, estatisticas: Estatistica, pecasJogo: List <CrazyPiece>, jogo: InfoJogo): boolean
TorreV
~ TorreV(idPeca: int, tipo: int, idEquipa: int, alcunha: String)
+ getValorRelativo(): String
+ getTipo(): String
~ sugerirJogadas(xO: int, yO: int, pecasJogo: List <CrazyPiece>, tamanhoTabuleiro: int): List <String>
~movePeca(xO: int, yO: int, xD: int, yD: int, estatisticas: Estatistica, pecasJogo: List <CrazyPiece>, jogo: InfoJogo): boolean