grid::Grid

-bounds = Game.GRID_SIZE * Game.GRID_SIZE: int
-picked = false: boolean
-mines = new ArrayList<Integer>(): ArrayList<Integer>
+cellGrid = new ArrayList<Common_cell>(): ArrayList<Common_cell>
-grid: Common_cell[][]
-dimension: int
-mine: int
-craziness: int

+Grid(int dimension, int mine): ctor
+VGrid(GridLayout g, Handler h): void
-createCells(Handler h): void
-addCells(): void
+getDimensao(): int
+getMinas(): int
+getGrid(): Common_cell[][]
+printGrid(): void
+positionMines(): void
+placeMines(int line, int collum): void
+hideGrid(): void
+revealGrid(): void
+revealCells(int line, int collum): void
+checkMines(int line, int collum): boolean
+revealMines(): void
+checkWin(): boolean
+setFlag(int linha, int coluna): void
+placeCrazyness(int crazyLevel): void
+getCraziness(): int
+crazyCell(int line, int collum): boolean
+getVisible(int line, int collum): boolean

cells::Common_cell

-type: int
-discovered: boolean
-flagged: boolean
-crazyness: boolean
-frame: String
-handler: Handler

+Common_cell(): ctor
+clickButton(): void
+rightClickButton(): void
+setType(int type): void
+getType(): int
+getDiscovered(): boolean
+setDiscovered(boolean d): void
+setFlagged(boolean f): void
+getFlagged(): boolean
+setFrame(String frame): void
+getFrame(): String
+isEmptyCell(): boolean
+isMine(): boolean
+isNearMine(): boolean
+isCrazyCell(): boolean
+changeToCrazyCell(): void

«interface» cells::I_Cell

+setType(int type): void
+getType(): int
+setDiscovered(boolean discovered): void
+getDiscovered(): boolean
+setFlagged(boolean flagged): void
+getFlagged(): boolean
+setFrame(String frame): void
+getFrame(): String
+isEmptyCell(): boolean
+isMine(): boolean
+isNearMine(): boolean
+isCrazyCell(): boolean

«interface» grid::I_Grid

+printGrid(): void
+positionMines(): void
+placeMines(int line, int collum): void
+hideGrid(): void
+revealGrid(): void
+revealCells(int line, int collum): void
+checkMines(int line, int collum): boolean
+revealMines(): void
+checkWin(): boolean
+setFlag(int line, int collum): void
+getVisible(int line, int collum): boolean

main::Game

+WIDTH = 720, HEIGHT = 720: int
+GRID_SIZE = 10: int
+MINECOUNT = 15: int
-handler = new Handler(): Handler

+Game(): ctor
+main(String[] args): void

gameLogic::Handler

-current = new ArrayList<Common_cell>(): ArrayList<Common_cell>
-queue = new ArrayList<Common_cell>(): ArrayList<Common_cell>
-flaggedCells = 0: int

+click(Common_cell cell): void
+rightClick(Common_cell cell): void

cells::Empty_cell

+Empty_cell(): ctor
+isEmptyCell(): boolean
+isMine(): boolean
+isNearMine(): boolean

cells::Mine_cell

+Mine_cell(): ctor
+isEmptyCell(): boolean
+isMine(): boolean
+isNearMine(): boolean

cells::NearMine_cell

+NearMine_cell(): ctor
+isEmptyCell(): boolean
+isMine(): boolean
+isNearMine(): boolean

main::Window

-frame: JFrame
-title: String

+Window(int width, int height, int gridSize, String title, Game game, Handler handler): ctor
+update(int flagged): void

grid::Grid_easy

+Grid_easy(): ctor

grid::Grid_hard

+Grid_hard(): ctor

grid::Grid_medium

+Grid_medium(): ctor