UML Class Diagram

Digger
- SIZE: int
- digSolution():void

UndoSystem
- MAX_UNDOS: int
- numberUndos: int
- UndoSystem(Difficulty)
- addUndo(UndoMove):void
- getLastUndoMove():UndoMove
- getSize():int
- getMaxUndos():int

PuzzleCreator
- SIZE: int
- BLOCK: int
- NUM_GAMES: int
- MAX_SWAPS: int
- RAND_COLS: int
- EASY_BASE: int
- MEDIUM_BASE: int
- HARD_BASE: int
- EXPERT_BASE: int
- SOLUTION_1: int[][]
- SOLUTION_2: int[][]
- pGrid: int[][]
- solvedGrid: int[][]
- solutionGrid: int[][]
- other: int[][]
- PuzzleCreator()
- createSolution():void
- generateNewPuzzle(Difficulty):void
- validGame(int[][]):boolean
- retreiveInitialPuzzleState():Puzzle
- retreivePuzzleSolution():Puzzle
- copySolution():void
- createPuzzle(int):void
- digPuzzle(Digger):void
- getNumGivens(Difficulty):int
- setPuzzleGrid(int[][]):void
- transformGrid(int[][]):int[][]
- validRegions(int[][]):boolean
- validColumns(int[][]):boolean
- validRows(int[][]):boolean
- swapColumns(int[][],int,int,int):void
- swapColumnBlocks(int[][],int,int,int):void
- swapDigits(int[][],int,int,int):void
- printPuzzle(Puzzle):void
- printGrid(int[][]):void

DigOneByOne
- RESET_LIMIT: int
- givens: int
- DigOneByOne(PuzzleCreator,int)
- digSolution():void

DigJumpOne
- givens: int
- DigJumpOne(PuzzleCreator,int)
- digSolution():void

DigRandom
- NUM_LOOPS: int
- givens: int
- DigRandom(PuzzleCreator,int)
- digSolution():void

UndoMove
- x: int
- y: int
- oldValue: int
- newValue: int
- UndoMove(int,int,int,int)
- getOld():int
- getNew():int
- getX():int
- getY():int

GameStatsPanel
- tilesLeftValue: JLabel
- tilesPlacedValue: JLabel
- hintsValue: JLabel
- undosAvailable: JLabel
- time: JLabel
- FONT_SIZE: int
- timer: Timer
- timeValue: int
- GameStatsPanel(Board)
- intToTime(int):String
- getTime():String
- stopTimer():void
- update(Board):void
- getTilesLeftValue():int
- setTilesLeftValue(int):void
- getTilesPlacedValue():int
- setTilesPlacedValue(int):void

OptionsPanel
- buttonSize: int
- howToPlay: String
- OptionsPanel(UIFrame)
- getStatsPanel():GameStatsPanel
- showStatsPanel(Board):void

Game
- Game()
- generatePuzzle(Board,Difficulty):void
- setCell(Board,int,int,int,boolean):void
- getHintsRemaining():int
- showHint(Board):void
- resetCurrentPuzzle(Board):void
- checkSolution():boolean
- undoMove(Board):void
- getUndoSystem():UndoSystem

UIGridSquare
- field: JLabel
- previousColour: Color
- isProtected: boolean
- isHint: boolean
- HINT_MARKER: String
- CELL_SIZE: int
- position: Point
- UIGridSquare(Board,NumbersPanel,Point)
- createClickListener():void
- createHoverListener():void
- getValue():int
- setIsProtected(Boolean):void
- getIsProtected():boolean
- setHint(Boolean):void
- setValue(String):void
- setColor(Color):void
- setTextColour(Color):void

NumbersPanel
- group: ButtonGroup
- NumbersPanel()
- createMouseListener():void
- getCurrentlySelected():String
- keyPressed(KeyEvent):void
- keyReleased(KeyEvent):void
- keyTyped(KeyEvent):void

UIFrame
- serialVersionUID: long
- X_PIXELS: int
- Y_PIXELS: int
- UIFrame()
- getGame():Game
- getBoard():Board
- getOptions():OptionsPanel
- showGameOver():void
- startNewGame():void

Board
- HORIZONTAL_LENGTH: int
- VERTICAL_LENGTH: int
- isActive: Boolean
- Board(UIFrame,NumbersPanel)
- getFrame():UIFrame
- generateCells(NumbersPanel):void
- getBoardSize():Point
- get(int,int):int
- set(int,int,int):void
- setCellProtected(int,int,Boolean):void
- getCellProtected(int,int):boolean
- setHintCell(int,int):void
- unlockBoard():void
- resetBoard():void
- puzzleCompleted():void
- getIsActive():boolean
- setIsActive(Boolean):void

HintSystem
- RANGE: int
- hintsRemaining: int
- HintSystem(Difficulty)
- getHintsRemaining():int
- getHintCell(Puzzle):Point
- legal(int,int,int,Puzzle):boolean

Puzzle
- grid: int[][]
- NUM_ROWS: int
- NUM_COLS: int
- Puzzle()
- getRowCount():int
- getColCount():int
- get(int,int):int
- set(int,int,int):void
- equals(Object):boolean

Relationships / labels:
- -undoSystem 0..1
- -list 0..*
- -statsPanel 0..1
- -stats 0..1
- -optionsPanel 0..1
- -frame 0..1
- -game 0..1
- -pC 0..1
- -pC 0..1
- -numbers 0..1
- -numbersPanel 0..1
- -squares 0..*
- -board 0..1
- -frame 0..1
- -board 0..1
- -hintSystem 0..1
- -currentGameState
- -puzzle 0..1

Puzzle Creation Sequence

Game	PuzzleCreator	Puzzle	Digger	UniqueChecker

Game → PuzzleCreator: Request new puzzle

PuzzleCreator → PuzzleCreator: Initialise variables

PuzzleCreator → PuzzleCreator: create solution

PuzzleCreator → Digger: create a puzzle grid

loop [until enough cells removed]

Digger → Digger: attempt to remove a cell

Digger → UniqueChecker: check if unique solution maintained

UniqueChecker → Digger: return answer

Digger → PuzzleCreator: return a puzzle grid

PuzzleCreator → Puzzle: create solution (Puzzle)

Puzzle → PuzzleCreator: give solution (Puzzle)

PuzzleCreator → Puzzle: create puzzle (Puzzle)

Puzzle → PuzzleCreator: give puzzle (Puzzle)

PuzzleCreator → Game: give puzzle

PuzzleCreator → Game: give solution

Game	PuzzleCreator	Puzzle	Digger	UniqueChecker

Game

Puzzle board
Puzzle solution

generatePuzzleBoard()
resetCurrentPuzzle()
undoMove()
setCell()
getHints()
showHint()
getUndoSystem()

PuzzleCreator

Puzzle solution
Puzzle puzzle
int[][] puzzleGrid
int[][] solutionGrid

createSolution()
generateNewPuzzle()
retrieveInitialPuzzle()
retrievePuzzleSolution()

«Digger»

digSolution()

UniqueChecker

hasUniqueSolution()
solver()
isLegal()

Puzzle

int[][] grid

getValue()
setValue()

DigJumpOne

PuzzleCreator pC
int givens

digSolution()

DigOneByOne

PuzzleCreator pC
int givens

digSolution()

DigRandom

PuzzleCreator pC
int givens

digSolution()

1

1

1

1

1

1

1..*

1..*

1

1