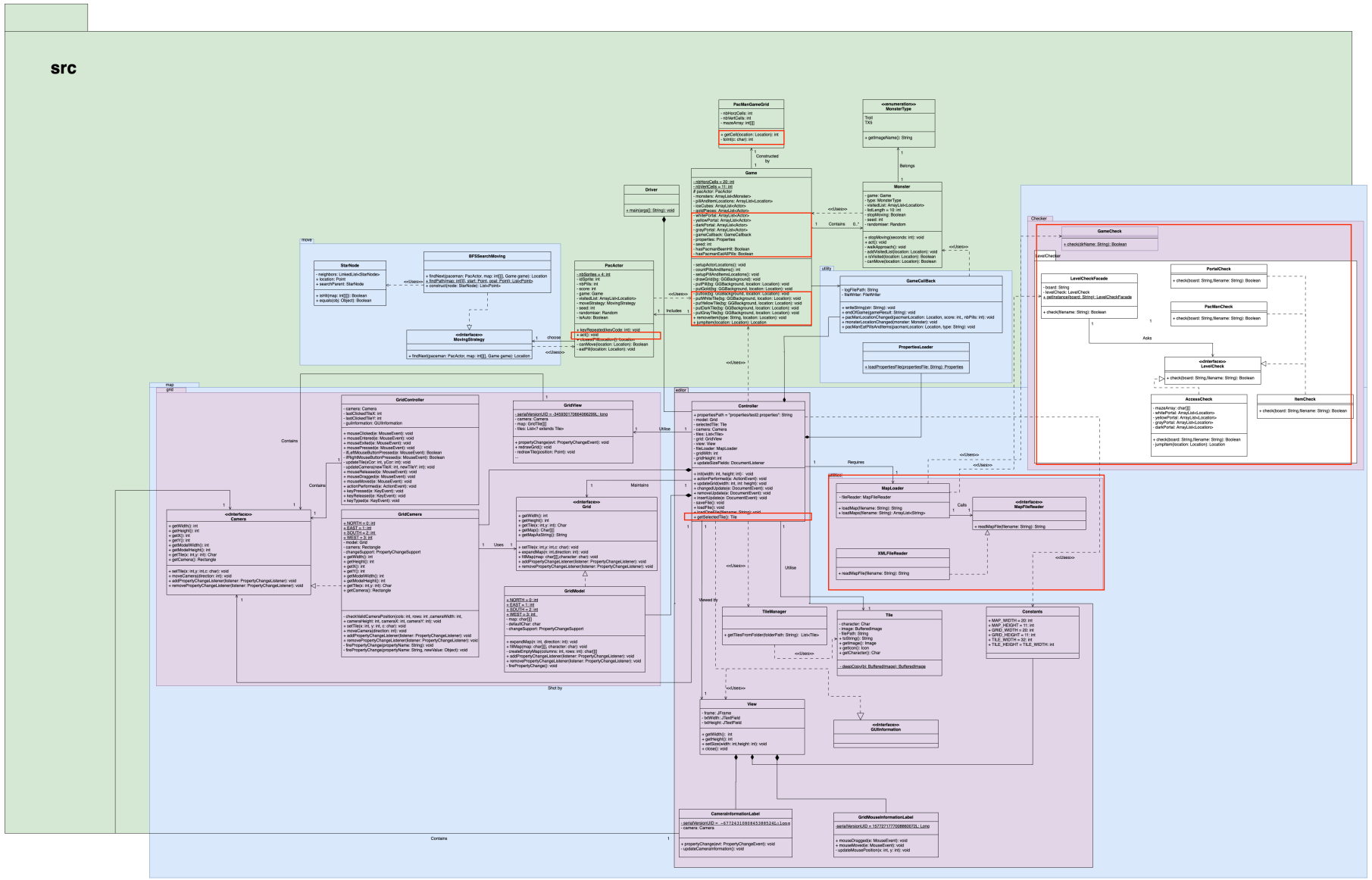
src


PacManGameGrid
- nbHorzCells: int
- nbVertCells: int
- mazeArray: int[][]
+ getCellLocation: Location): int
- toInt(c: char): int
Constructed by
<<enumeration>>
MonsterType
Troll
TXS
+ getImageName(): String
Belongs
Driver
+ main(args[]: String): void
Game
- nbHorzCells = 20: int
- nbVertCells = 11: int
pacActor: PacActor
- monsters: ArrayList<Monster>
- pillAndItemLocations: ArrayList<Location>
- iceCubes: ArrayList<Actor>
- addPlaces: ArrayList<Actor>
- whitePortal: ArrayList<Actor>
- yellowPortal: ArrayList<Actor>
- darkPortal: ArrayList<Actor>
- grayPortal: ArrayList<Actor>
- mazeStream: GameCallback
- properties: Properties
- seed: int
- hasPacmanBeenHit: Boolean
- hasPacmanEatAllPills: Boolean
- setupActorLocations(): void
- countPillsAndItems(): int
- setupPillAndItemsLocations(): void
- drawGrid(g): GUIBackground
- putPill(g): GUIBackground, location: Location): void
- putGold(g): GUIBackground, location: Location): void
- putIceCube(g): GUIBackground, location: Location): void
- putYellowTile(g): GUIBackground, location: Location): void
- putDarkTile(g): GUIBackground, location: Location): void
- putGrayTile(g): GUIBackground, location: Location): void
+ removeItem(type: String, location: Location): void
+ jumpItem(location: Location): Location
<<Uses>>
Contains
Monster
- game: Game
- type: MonsterType
- visitedList: ArrayList<Location>
- listLength = 10: int
- stopMoving: Boolean
- seed: int
- randomVar: Random
+ stopMoving(seconds: int): void
+ act(): void
- walkApproach(): void
- addVisitedList(location: Location): void
+ isVisited(location: Location): Boolean
- canMove(location: Location): Boolean
<<Uses>>
move
StarNode
- neighbors: LinkedList<StarNode>
+ location: Point
+ searchParent: StarNode
+ isHit(map: int[][]): Boolean
+ equals(obj: Object): Boolean
<<Uses>>
BFSSearchMoving
+ findNext(paceman: PacActor, map: int[][], Game game): Location
+ findPath(map: int[][], start: Point, goal: Point): ListePoint>
+ construct(node: StarNode): ListePoint>
<<interface>>
MovingStrategy
+ findNext(paceman: PacActor, map: int[][], Game game): Location
choose
<<Uses>>
PacActor
- nbSprites = 4: int
- idSprite: int
- nbPills: int
- score: int
- game: Game
- pillsAndItems: ArrayList<Location>
- moveStrategy: MovingStrategy
- seed: int
- randomVar: Random
- isAuto: Boolean
+ keyReceivedIntoKeyCode: int): void
+ act(): void
+ closestPillLocation(): location
- canMove(location: Location): Boolean
- eatPill(location: Location): void
Includes
<<Uses>>
Checker
GameCheck
+ check(dirName: String): Boolean
LevelChecker
LevelCheckFacade
- board: String
- levelCheck: LevelCheck
+ getInstance(board: String): LevelCheckFacade
+ check(filename: String): Boolean
PortalCheck
+ check(board: String,filename: String): Boolean
PacManCheck
+ check(board: String,filename: String): Boolean
Asks
<<interface>>
LevelCheck
+ check(board: String,filename: String): Boolean
AccessCheck
- mazeArray: char[][]
- whitePortal: ArrayList<Location>
- yellowPortal: ArrayList<Location>
- grayPortal: ArrayList<Location>
- darkPortal: ArrayList<Location>
+ check(board: String,filename: String): Boolean
- jumpItem(location: Location): Location
ItemCheck
+ check(board: String,filename: String): Boolean
Utility
GameCallBack
- logFilePath: String
- fileWriter: FileWriter
+ writeString(str: String): void
+ endOfGame(gameResult: String): void
+ pacManLocationChanged(pacmanLocation: Location, score: int, nbPills: int): void
+ monsterLocationChanged(monster: Monster): void
+ pacManEatPillsAndItems(pacmanLocation: Location, type: String): void
PropertiesLoader
+ loadPropertiesFile(propertiesFile: String): Properties
map
grid
GridController
- camera: Camera
- lastClickedTileX: int
- lastClickedTileY: int
- guiInformation: GUIInformation
+ mouseClicked(e: MouseEvent): void
+ mouseEntered(e: MouseEvent): void
+ mouseExited(e: MouseEvent): void
+ mousePressed(e: MouseEvent): void
- IsLeftMouseButtonPressed(e: MouseEvent): Boolean
+ mouseMoved(e: MouseEvent): void
- updateTileIsOn(e: MouseEvent): Boolean
- updateCamera(newTileX: int, newTileY: int): void
+ mouseReleased(e: MouseEvent): void
+ mouseDragged(e: MouseEvent): void
+ mouseMoved(e: MouseEvent): void
+ actionPerformed(a: ActionEvent): void
+ keyPressed(e: KeyEvent): void
+ keyReleased(e: KeyEvent): void
+ keyTyped(e: KeyEvent): void
GridView
- serialVersionUID = -348392172664066295L: long
- camera: Camera
- map: GridTile[]
- files: List<? extends Tile>
+ propertyChange(evt: PropertyChangeEvent): void
+ redrawGrid(): void
- redrawTile(position: Point): void
Utilise
Contains
Contains
<<interface>>
Camera
+ getWidth(): int
+ getHeight(): int
+ getX(): int
+ getY(): int
+ getModelWidth(): int
+ getModelHeight(): int
+ getTile(x: int,y: int): Char
+ getCamera(): Rectangle
+ setTile(x: int,y: int,c: char): void
+ moveCamera(direction: int): void
+ addPropertyChangeListener(listener: PropertyChangeListener): void
+ removePropertyChangeListener(listener: PropertyChangeListener): void
GridCamera
+ NORTH = 0: int
+ EAST = 1: int
+ SOUTH = 2: int
+ WEST = 3: int
- tileSize: int
- camera: Rectangle
- changeSupport: PropertyChangeSupport
+ getWidth(): int
+ getHeight(): int
+ getX(): int
+ getY(): int
+ getModelWidth(): int
+ getModelHeight(): int
+ getTile(x: int,y: int): Char
+ getCamera(): Rectangle
- checkValidCameraPosition(colx: int, rowx: int ,cameraWidth: int, cameraHeight: int, cameraX: int, cameraY: int): void
+ setTile(x: int, y: int, c: char): void
+ moveCamera(direction: int): void
+ addPropertyChangeListener(listener: PropertyChangeListener): void
+ removePropertyChangeListener(listener: PropertyChangeListener): void
- firePropertyChange(propertyName: String): void
- firePropertyChange(propertyName: String, newValue: Object): void
Uses
Maintains
<<interface>>
Grid
+ getWidth(): int
+ getHeight(): int
+ getTile(x: int,y: int): Char
+ getMap(): Char[][]
+ getMapAsString(): String
+ setTile(x: int,y: int,c: char): void
+ expandMap(n: int,direction: int): void
+ fillMap(map: char[][],character: char): void
+ addPropertyChangeListener(listener: PropertyChangeListener): void
+ removePropertyChangeListener(listener: PropertyChangeListener): void
GridModel
+ NORTH = 0: int
+ EAST = 1: int
+ SOUTH = 2: int
+ WEST = 3: int
- map: char[][]
- defaultChar: char
- changeSupport: PropertyChangeSupport
+ expandMap(n: int, direction: int): void
+ fillMap(map: char[][], character: char): void
- createEmptyMap(columns: int, rows: int): char[][]
+ addPropertyChangeListener(listener: PropertyChangeListener): void
+ removePropertyChangeListener(listener: PropertyChangeListener): void
- firePropertyChange(): void
Shot by
editor
Controller
+ propertiesPath = "properties/test2.properties": String
- model: Grid
- selectedTile: Tile
- camera: Camera
- tiles: List<Tile>
- grid: GridView
- view: View
- fileLoader: MapLoader
- gridWith: int
- gridHeight: int
- updateSizeFields: DocumentListener
+ initWidth(v: int, height: int): void
+ updateWidth(e: DocumentEvent): void
+ updateGridWidth(v: int, int height): void
+ changedUpdate(e: DocumentEvent): void
+ removeUpdate(e: DocumentEvent): void
+ insertUpdate(e: DocumentEvent): void
+ saveFile(): void
+ loadFile(): void
+ loadOneFile(filename: String): void
+ getSelectedTile(): Tile
Requires
<<Uses>>
<<Uses>>
Utilise
Utilise
Viewed by
<<Uses>>
<<Uses>>
<<Uses>>
MapLoader
- fileReader: MapFileReader
+ loadMap(filename: String): String
+ loadMaps(filename: String): ArrayList<String>
Calls
<<interface>>
MapFileReader
+ readMapFile(filename: String): String
XMLFileReader
+ readMapFile(filename: String): String
TileManager
+ getTilesFromFolder(folderPath: String): List<Tile>
Tile
- character: Char
- Image: BufferedImage
- filePath: String
+ toString(): String
+ getImage(): Image
+ getIcon(): Icon
+ getCharacter(): Char
- deepCopy(bi: BufferedImage): BufferedImage
Constants
+ MAP_WIDTH = 20: int
+ MAP_HEIGHT = 11: int
+ GRID_WIDTH = 20: int
+ GRID_HEIGHT = 11: int
+ TILE_WIDTH = 32: int
+ TILE_HEIGHT = TILE_WIDTH: int
View
- frame: JFrame
- txtWidth: JTextField
- txtHeight: JTextField
+ getWidth(): int
+ getHeight(): int
+ setSize(width: int,height: int): void
+ close(): void
<<interface>>
GUIInformation
CameraInformationLabel
- serialVersionUID = -6772431189345389524L: Long
- camera: Camera
+ propertyChange(evt: PropertyChangeEvent): void
- updateCameraInformation(): void
GridMouseInformationLabel
-serialVersionUID = 1577221777088600721L: Long
+ mouseDragged(e: MouseEvent): void
+ mouseMoved(e: MouseEvent): void
- updateMousePosition(x: int, y: int): void
Contains