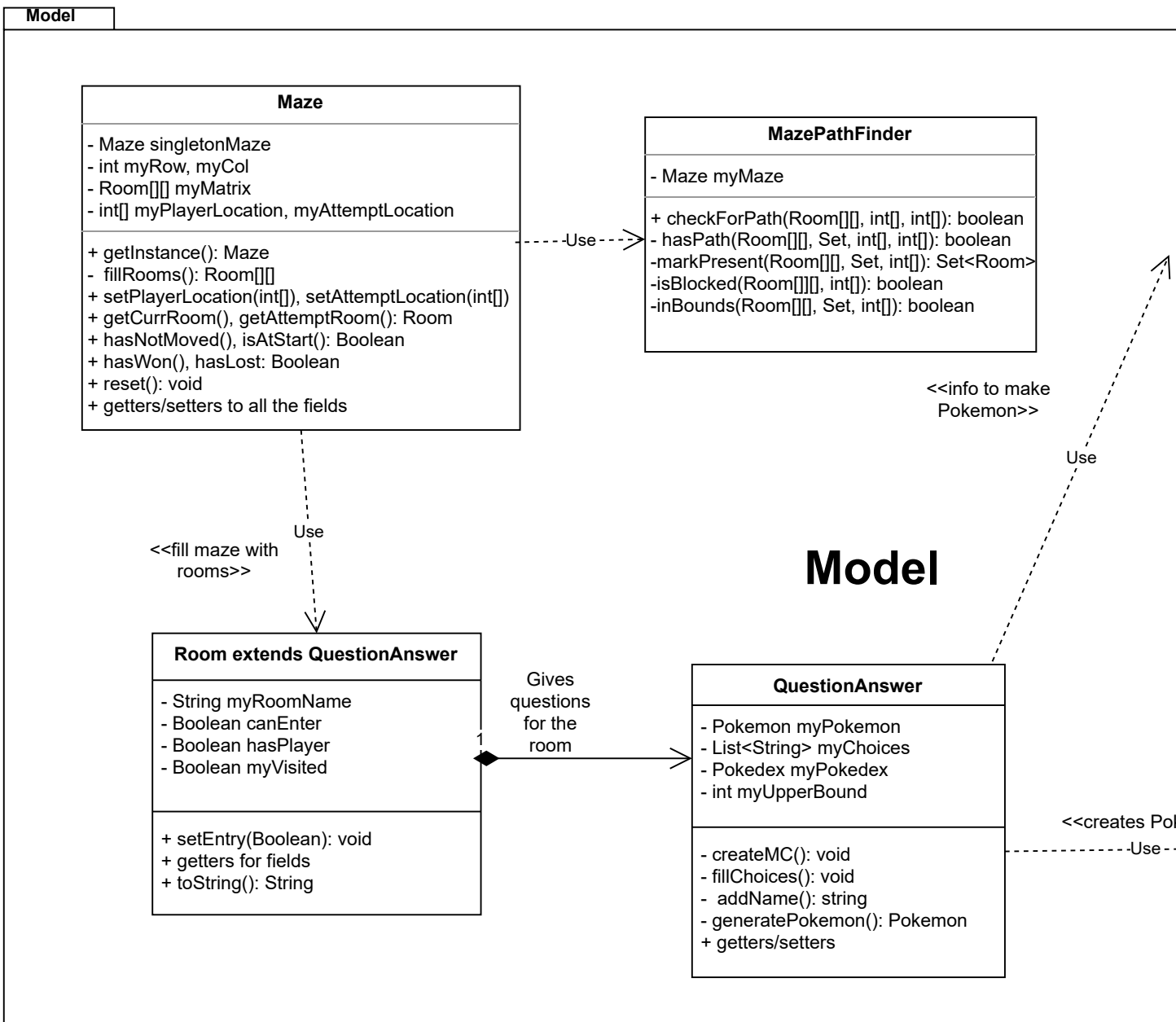

Model
Maze
- Maze singletonMaze
- int myRow, myCol
- Room[][] myMatrix
- int[] myPlayerLocation, myAttemptLocation
+ getInstance(): Maze
- fillRooms(): Room[][]
+ setPlayerLocation(int[]), setAttemptLocation(int[])
+ getCurrRoom(), getAttemptRoom(): Room
+ hasNotMoved(), isAtStart(): Boolean
+ hasWon(), hasLost: Boolean
+ reset(): void
+ getters/setters to all the fields
Use
MazePathFinder
- Maze myMaze
+ checkForPath(Room[][], int[], int[]): boolean
- hasPath(Room[][], Set, int[], int[]): boolean
-markPresent(Room[][], Set, int[]): Set<Room>
-isBlocked(Room[][], int[]): boolean
-inBounds(Room[][], Set, int[]): boolean
<<info to make Pokemon>>
Use
<<fill maze with rooms>>
Use
Model
Room extends QuestionAnswer
- String myRoomName
- Boolean canEnter
- Boolean hasPlayer
- Boolean myVisited
+ setEntry(Boolean): void
+ getters for fields
+ toString(): String
Gives questions for the room
1
QuestionAnswer
- Pokemon myPokemon
- List<String> myChoices
- Pokedex myPokedex
- int myUpperBound
- createMC(): void
- fillChoices(): void
- addName(): string
- generatePokemon(): Pokemon
+ getters/setters
<<creates Pok
Use
Manipulates
Controller
CON
AnswerFormatter
+formatAnswer(String): String
+formatMultipleChoice(String): String
+idConverter(String): String
SQLPokedexFiller
-SQLiteDataSource ds
+buildDatabase(String, int): void
-setupSQL(String):void
controller.movem
AbstractMoveme
-Maze myMaze
-PokemonPanel myPan
-ImageIcon myIcon
~AbstractMovementAct
ImageIcon, Pokemon
+movePlayer(int[]): void

View

Pokedex

+ Pokedex singletonDex
- Map<Integer, ArrayList<String,String> myPokedex
- Map<String, Integer> myNameDex
- Set<Integer> mySelectedGens
- int myCounter

+ getInstance(): Pokedex
- fillPokedex(int)
+ addGenToDex(int), addAllGensToDex()
+ removeGenFromDex(int)
+ restoreGensToDefault()
+ canRemoveGen(): Boolean
- resetPokedex(); void
+ addPokemon(String, String)
+ findPokemon(String), findPokemon(int): Pokemon
+ hasPokemon(): Boolean
+ toString(): String
+ getters for fields

Updates

Pokemon

- String myName
- String myID
- BufferedImage myPNG

+ getters for fields
+ compareTo(): int
+ toString(): String

<<Java Cl
MCPanel
view.viewHelper.Q
MCPanelListener()
actionPerformed(A

NTROLLER

ment_actions

ntAction

nel

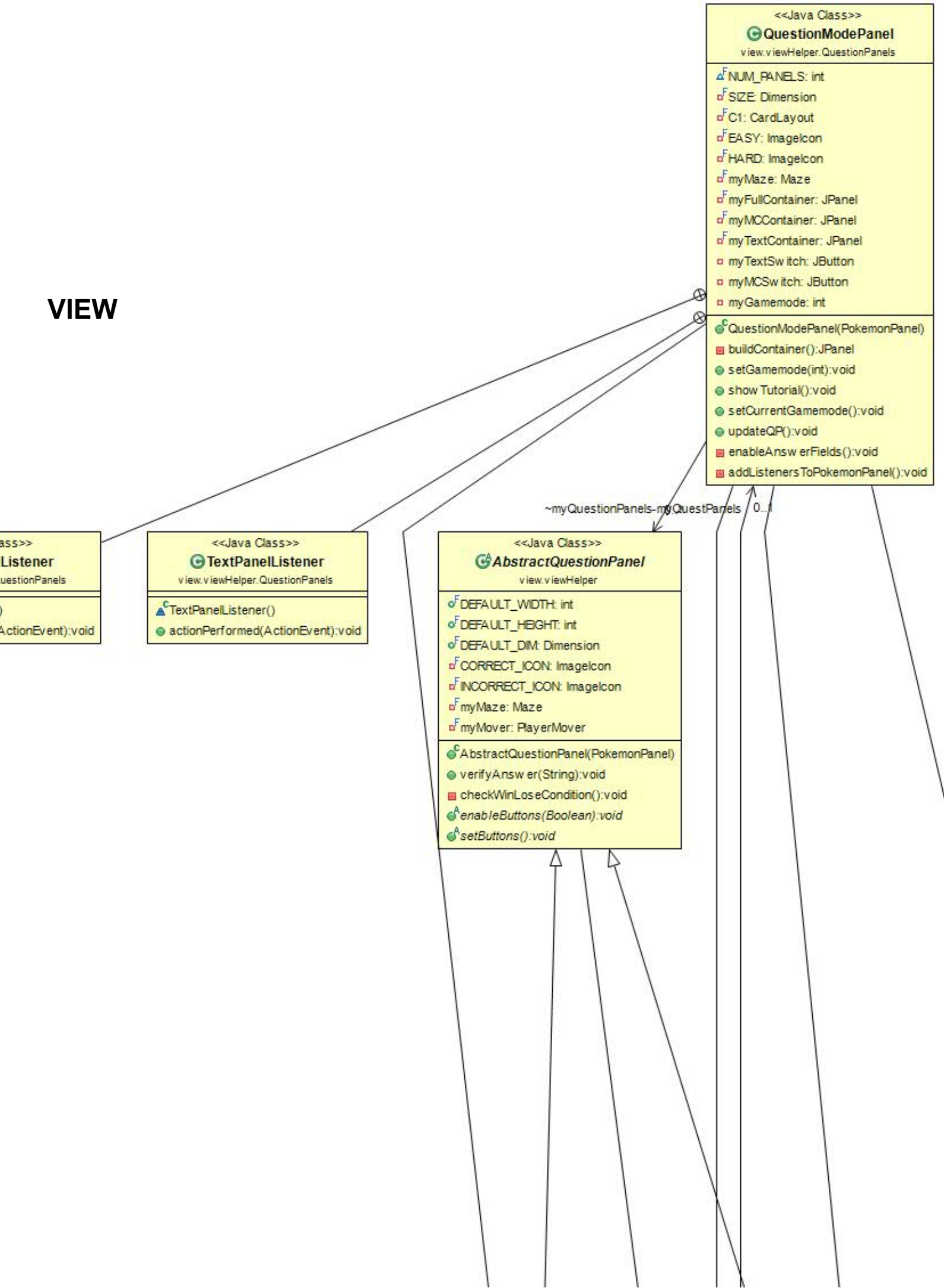
tion(String,
nPanel)
l

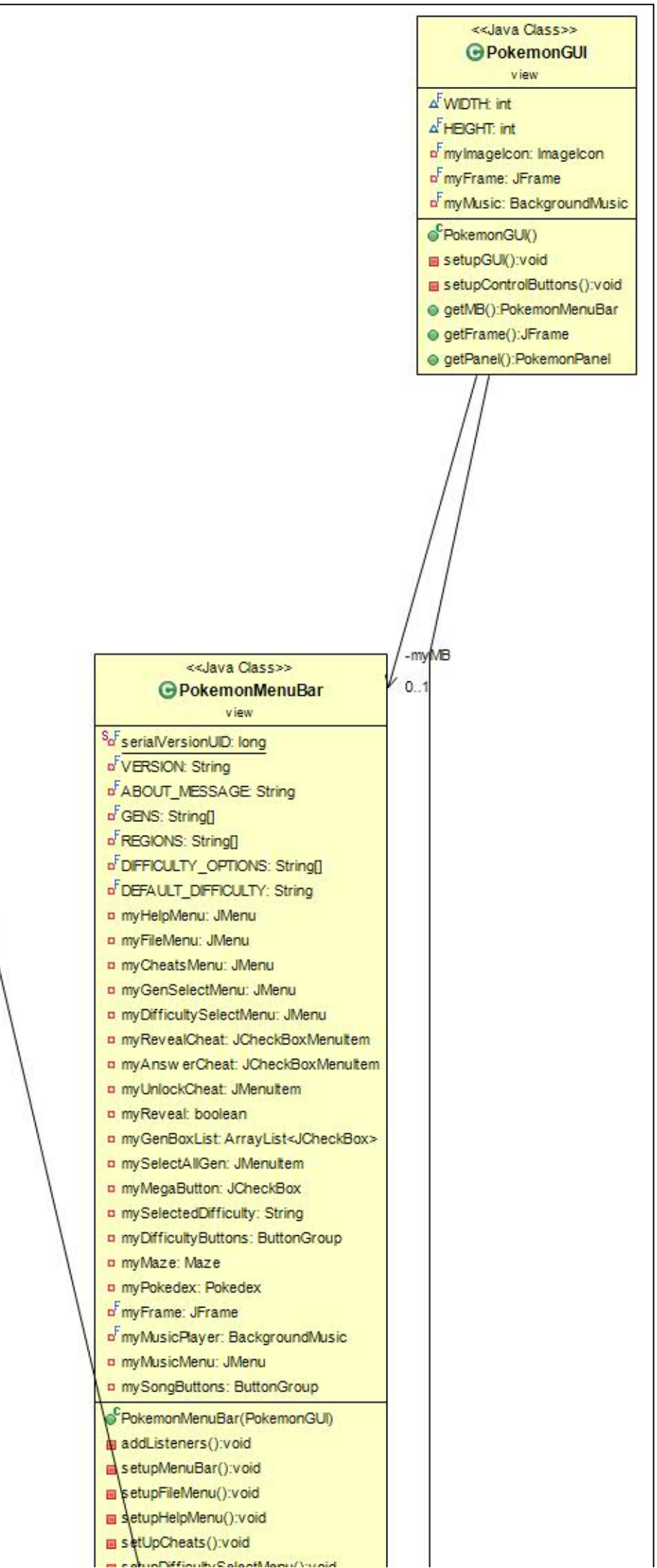
LeftAction

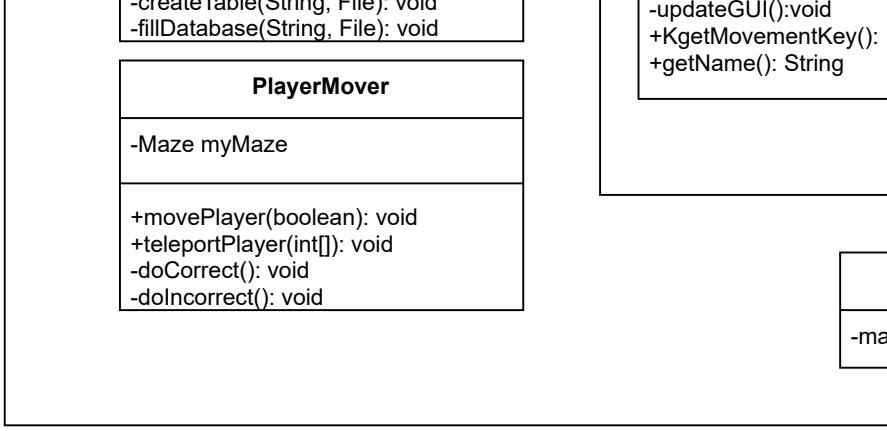
RightAction

UpAction

VIEW


<<Java Class>>
QuestionModePanel
view.viewHelper.QuestionPanels
NUM_PANELS: int
SIZE: Dimension
C1: CardLayout
EASY: ImageIcon
HARD: ImageIcon
myMaze: Maze
myFullContainer: JPanel
myMCContainer: JPanel
myTextContainer: JPanel
myTextSwitch: JButton
myMCSwitch: JButton
myGamemode: int
QuestionModePanel(PokemonPanel)
buildContainer():JPanel
setGamemode(int):void
showTutorial():void
setCurrentGamemode():void
updateQP():void
enableAnswerFields():void
addListenersToPokemonPanel():void
~myQuestionPanels-myQuestPanels
0..1
<<Java Class>>
TextPanelListener
view.viewHelper.QuestionPanels
TextPanelListener()
actionPerformed(ActionEvent):void
<<Java Class>>
AbstractQuestionPanel
view.viewHelper
DEFAULT_WIDTH: int
DEFAULT_HEIGHT: int
DEFAULT_DIM: Dimension
CORRECT_ICON: ImageIcon
INCORRECT_ICON: ImageIcon
myMaze: Maze
myMover: PlayerMover
AbstractQuestionPanel(PokemonPanel)
verifyAnswer(String):void
checkWinLoseCondition():void
enableButtons(Boolean):void
setButtons():void


<<Java Class>>
PokemonGUI
view
WIDTH: int
HEIGHT: int
myImageIcon: ImageIcon
myFrame: JFrame
myMusic: BackgroundMusic
PokemonGUI()
setupGUI():void
setupControlButtons():void
getMB():PokemonMenuBar
getFrame():JFrame
getPanel():PokemonPanel
-myMB
0..1
<<Java Class>>
PokemonMenuBar
view
serialVersionUID: long
VERSION: String
ABOUT_MESSAGE: String
GENS: String[]
REGIONS: String[]
DIFFICULTY_OPTIONS: String[]
DEFAULT_DIFFICULTY: String
myHelpMenu: JMenu
myFileMenu: JMenu
myCheatsMenu: JMenu
myGenSelectMenu: JMenu
myDifficultySelectMenu: JMenu
myRevealCheat: JCheckBoxMenuItem
myAnsw erCheat: JCheckBoxMenuItem
myUnlockCheat: JMenuItem
myReveal: boolean
myGenBoxList: ArrayList<JCheckBox>
mySelectAllGen: JMenuItem
myMegaButton: JCheckBox
mySelectedDifficulty: String
myDifficultyButtons: ButtonGroup
myMaze: Maze
myPokedex: Pokedex
myFrame: JFrame
myMusicPlayer: BackgroundMusic
myMusicMenu: JMenu
mySongButtons: ButtonGroup
PokemonMenuBar(PokemonGUI)
addListeners():void
setupMenuBar():void
setupFileMenu():void
setupHelpMenu():void
setUpCheats():void

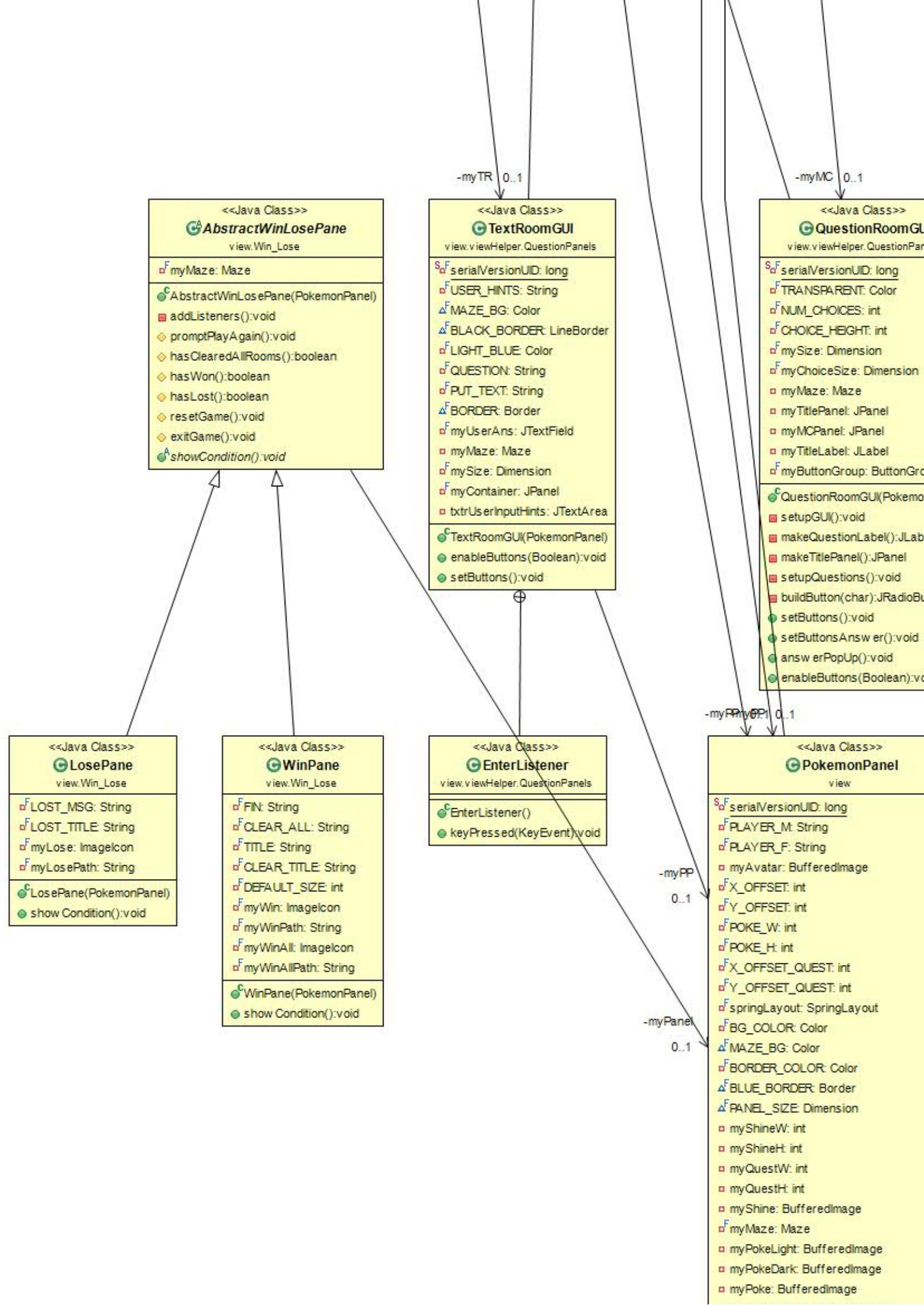


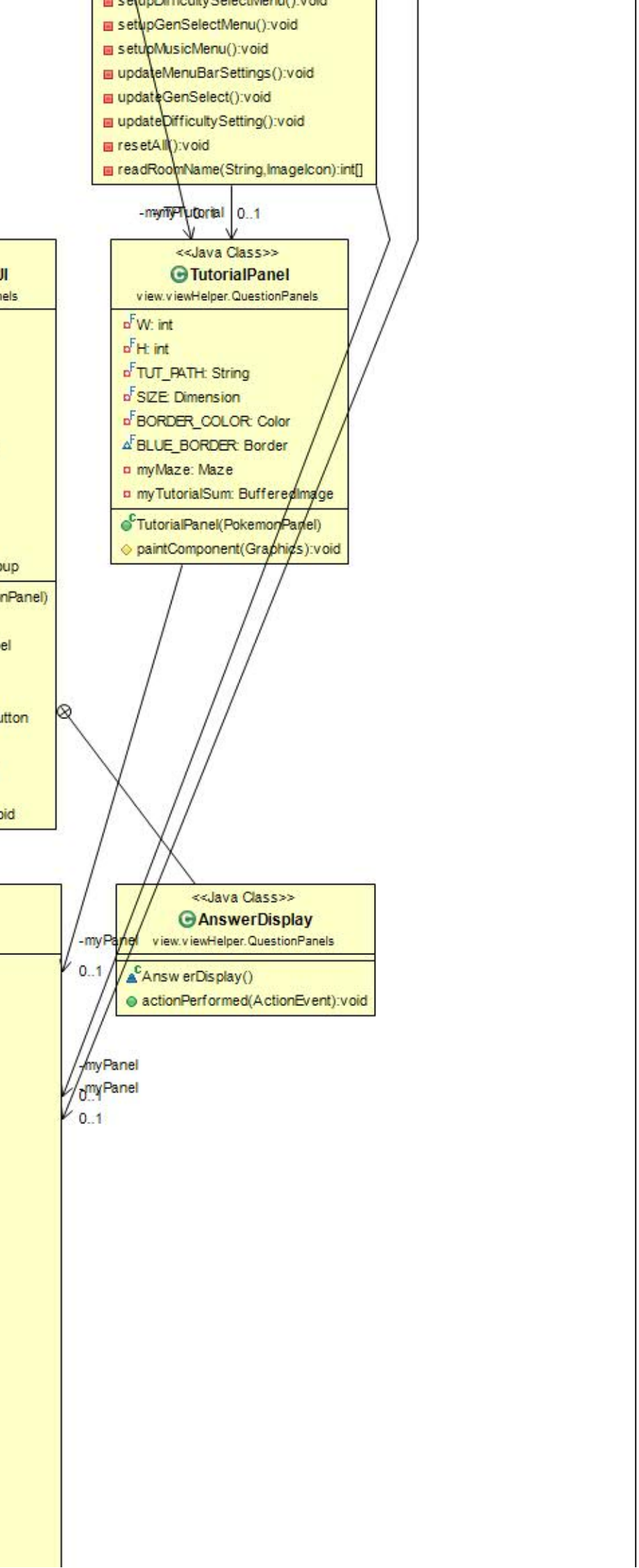
PlayerMover

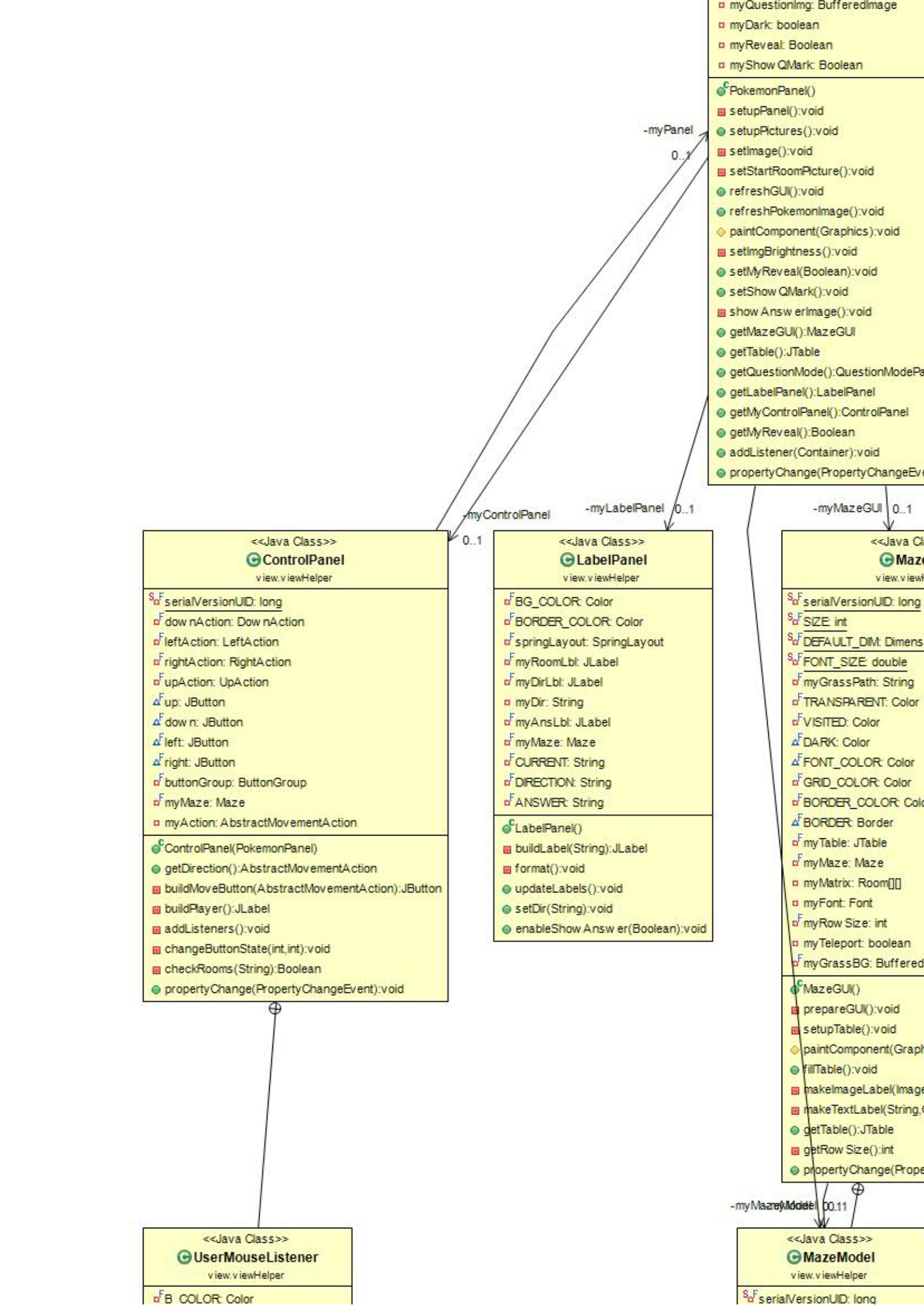
-Maze myMaze

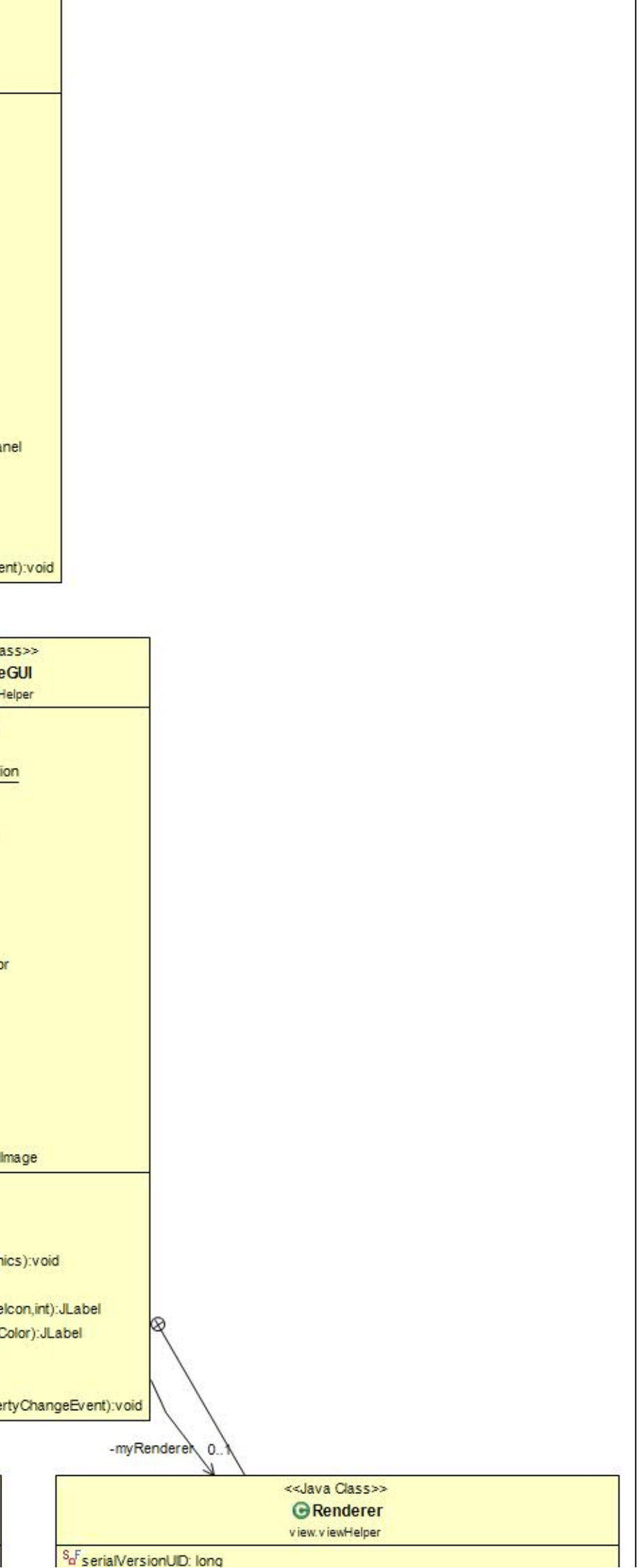
+movePlayer(boolean): void
+teleportPlayer(int[]): void
-doCorrect(): void
-doIncorrect(): void

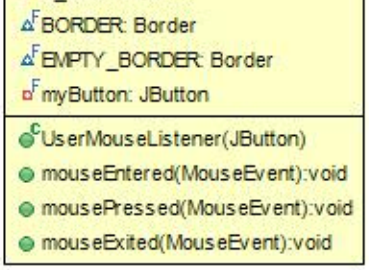
KeyStroke
DownAction
PkmnTriviaMain
ain runs program
Runs GUI

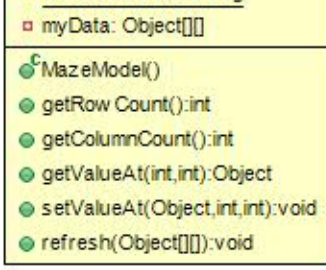

-myTR 0..1
-myMC 0..1
<<Java Class>>
AbstractWinLosePane
view.Win_Lose
myMaze: Maze
AbstractWinLosePane(PokemonPanel)
addListeners():void
promptPlayAgain():void
hasClearedAllRooms():boolean
hasWon():boolean
hasLost():boolean
resetGame():void
exitGame():void
showCondition():void
<<Java Class>>
TextRoomGUI
view.viewHelper.QuestionPanels
serialVersionUID: long
USER_HINTS: String
MAZE_BG: Color
BLACK_BORDER: LineBorder
LIGHT_BLUE: Color
QUESTION: String
PUT_TEXT: String
BORDER: Border
myUserAns: JTextField
myMaze: Maze
mySize: Dimension
myContainer: JPanel
txtrUserInputHints: JTextArea
TextRoomGUI(PokemonPanel)
enableButtons(Boolean):void
setButtons():void
<<Java Class>>
QuestionRoomGU
view.viewHelper.QuestionPan
serialVersionUID: long
TRANSPARENT: Color
NUM_CHOICES: int
CHOICE_HEIGHT: int
mySize: Dimension
myChoiceSize: Dimension
myMaze: Maze
myTitlePanel: JPanel
myMCPanel: JPanel
myTitleLabel: JLabel
myButtonGroup: ButtonGro
QuestionRoomGUI(Pokemo
setupGUI():void
makeQuestionLabel():JLab
makeTitlePanel():JPanel
setupQuestions():void
buildButton(char):JRadioBu
setButtons():void
setButtonsAnswer():void
answerPopUp():void
enableButtons(Boolean):vo
<<Java Class>>
LosePane
view.Win_Lose
LOST_MSG: String
LOST_TITLE: String
myLose: ImageIcon
myLosePath: String
LosePane(PokemonPanel)
show Condition():void
<<Java Class>>
WinPane
view.Win_Lose
FIN: String
CLEAR_ALL: String
TITLE: String
CLEAR_TITLE: String
DEFAULT_SIZE: int
myWin: ImageIcon
myWinPath: String
myWinAll: ImageIcon
myWinAllPath: String
WinPane(PokemonPanel)
show Condition():void
<<Java Class>>
EnterListener
view.viewHelper.QuestionPanels
EnterListener()
keyPressed(KeyEvent):void
-myPP
0..1
-myPanel
0..1
<<Java Class>>
PokemonPanel
view
serialVersionUID: long
PLAYER_M: String
PLAYER_F: String
myAvatar: BufferedImage
X_OFFSET: int
Y_OFFSET: int
POKE_W: int
POKE_H: int
X_OFFSET_QUEST: int
Y_OFFSET_QUEST: int
springLayout: SpringLayout
BG_COLOR: Color
MAZE_BG: Color
BORDER_COLOR: Color
BLUE_BORDER: Border
PANEL_SIZE: Dimension
myShineW: int
myShineH: int
myQuestW: int
myQuestH: int
myShine: BufferedImage
myMaze: Maze
myPokeLight: BufferedImage
myPokeDark: BufferedImage
myPoke: BufferedImage

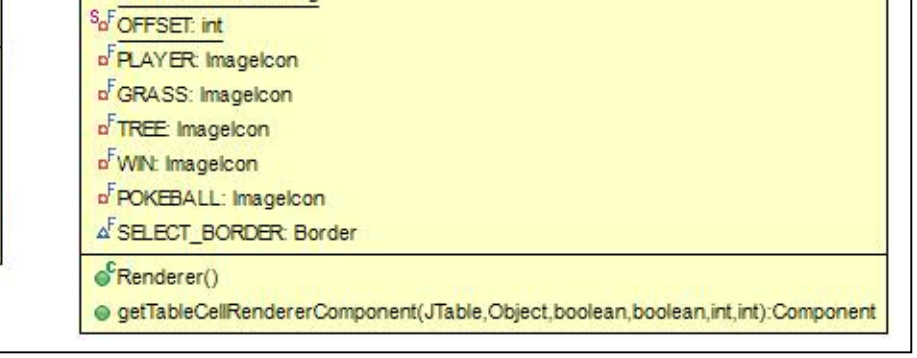

setupGenSelectMenu():void
setupMusicMenu():void
updateMenuBarSettings():void
updateGenSelect():void
updateDifficultySetting():void
resetAll():void
readRoomName(String,ImageIcon):int[]
-myTutorial
0..1
<<Java Class>>
TutorialPanel
view.viewHelper.QuestionPanels
W: int
H: int
TUT_PATH: String
SIZE: Dimension
BORDER_COLOR: Color
BLUE_BORDER: Border
myMaze: Maze
myTutorialSum: BufferedImage
TutorialPanel(PokemonPanel)
paintComponent(Graphics):void
<<Java Class>>
AnswerDisplay
view.viewHelper.QuestionPanels
AnswerDisplay()
actionPerformed(ActionEvent):void
-myPanel
0..1
-myPanel
0..1
0..1


myQuestionImg: BufferedImage
myDark: boolean
myReveal: Boolean
myShowQMark: Boolean
PokemonPanel()
setupPanel():void
setupPictures():void
setImage():void
setStartRoomPicture():void
refreshGUI():void
refreshPokemonImage():void
paintComponent(Graphics):void
setImgBrightness():void
setMyReveal(Boolean):void
setShowQMark():void
showAnswerImage():void
getMazeGUI():MazeGUI
getTable():JTable
getQuestionMode():QuestionModePa
getLabelPanel():LabelPanel
getMyControlPanel():ControlPanel
getMyReveal():Boolean
addListener(Container):void
propertyChange(PropertyChangeEve
-myPanel
0..1
-myControlPanel
0..1
-myLabelPanel
0..1
-myMazeGUI
0..1
<<Java Class>>
ControlPanel
view.viewHelper
serialVersionUID: long
downAction: DownAction
leftAction: LeftAction
rightAction: RightAction
upAction: UpAction
up: JButton
down: JButton
left: JButton
right: JButton
buttonGroup: ButtonGroup
myMaze: Maze
myAction: AbstractMovementAction
ControlPanel(PokemonPanel)
getDirection():AbstractMovementAction
buildMoveButton(AbstractMovementAction):JButton
buildPlayer():JLabel
addListeners():void
changeButtonState(int,int):void
checkRooms(String):Boolean
propertyChange(PropertyChangeEvent):void
<<Java Class>>
LabelPanel
view.viewHelper
BG_COLOR: Color
BORDER_COLOR: Color
springLayout: SpringLayout
myRoomLbl: JLabel
myDirLbl: JLabel
myDir: String
myAnsLbl: JLabel
myMaze: Maze
CURRENT: String
DIRECTION: String
ANSWER: String
LabelPanel()
buildLabel(String):JLabel
format():void
updateLabels():void
setDir(String):void
enableShowAnswer(Boolean):void
<<Java Cl
Maze
view.viewH
serialVersionUID: long
SIZE: int
DEFAULT_DIM: Dimens
FONT_SIZE: double
myGrassPath: String
TRANSPARENT: Color
VISITED: Color
DARK: Color
FONT_COLOR: Color
GRID_COLOR: Color
BORDER_COLOR: Colo
BORDER: Border
myTable: JTable
myMaze: Maze
myMatrix: Room[][]
myFont: Font
myRowSize: int
myTeleport: boolean
myGrassBG: Buffered
MazeGUI()
prepareGUI():void
setupTable():void
paintComponent(Graph
fillTable():void
makeImageLabel(Image
makeTextLabel(String,
getTable():JTable
getRowSize():int
propertyChange(Prope
-myMazeModel
0..1
<<Java Class>>
UserMouseListener
view.viewHelper
B_COLOR: Color
<<Java Class>>
MazeModel
view.viewHelper
serialVersionUID: long


anel
ent):void
ass>>
eGUI
Helper
ion
or
lImage
nics):void
elcon,int):JLabel
Color):JLabel
ertyChangeEvent):void
-myRenderer
0..1
<<Java Class>>
Renderer
view.viewHelper
serialVersionUID: long


BORDER: Border
EMPTY_BORDER: Border
myButton: JButton
UserMouseListener(JButton)
mouseEntered(MouseEvent):void
mousePressed(MouseEvent):void
mouseExited(MouseEvent):void


myData: Object[][]
MazeModel()
getRow Count():int
getColumnCount():int
getValueAt(int,int):Object
setValueAt(Object,int,int):void
refresh(Object[][]):void


OFFSET: int
PLAYER: ImageIcon
GRASS: ImageIcon
TREE: ImageIcon
WIN: ImageIcon
POKEBALL: ImageIcon
SELECT_BORDER: Border
Renderer()
getTableCellRendererComponent(JTable,Object,boolean,boolean,int,int):Component