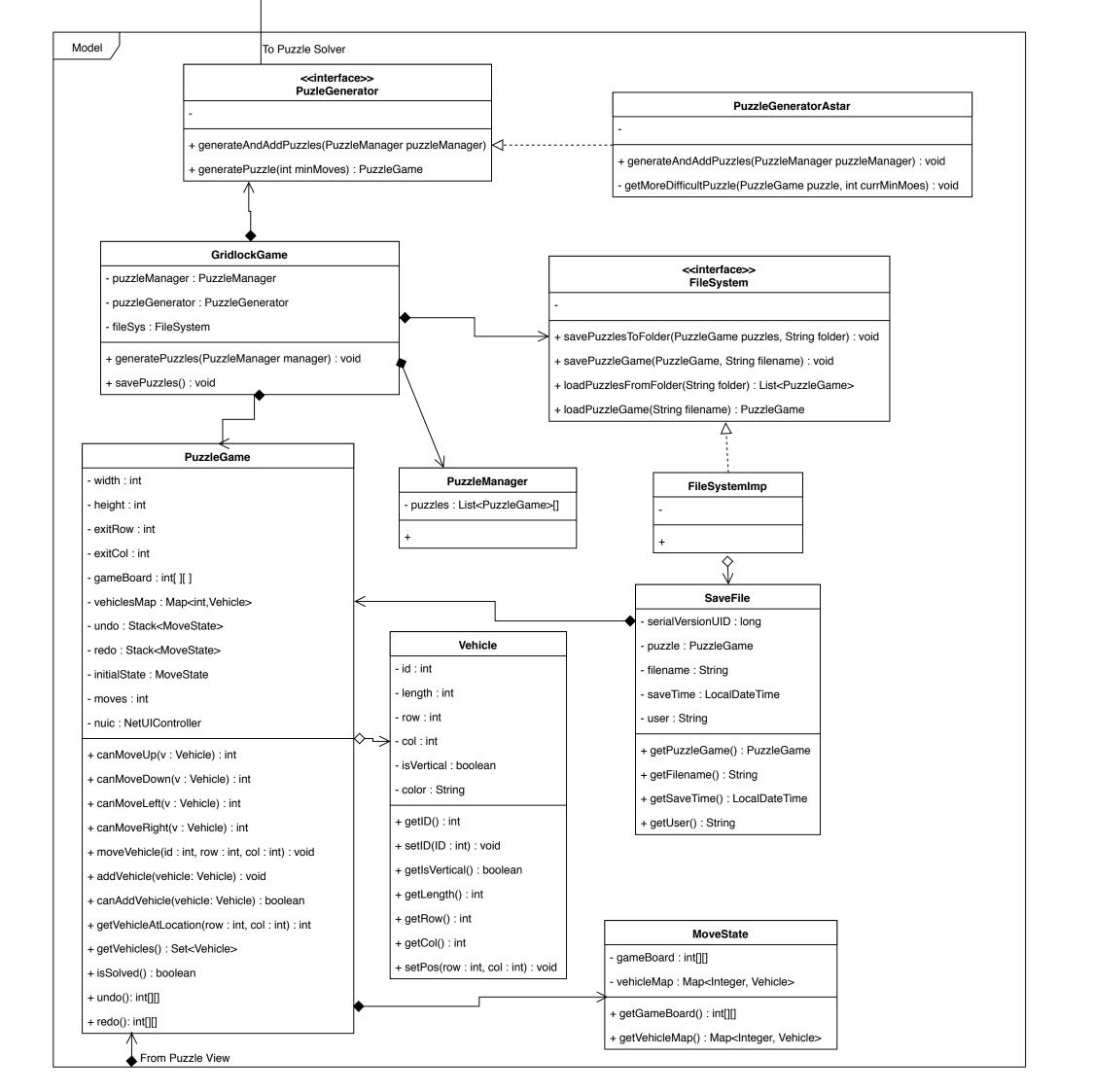


Network/Multiplayer <interface> RushHourServer RushHourServerEV NetUlController NetworkPanel NetworkUIController ss: ServerSocket eventQueue: Blocking Queue<String> np: NetworkPanel ChallengeBtn: JButton + getChallengeBtnListener(): ActionListener portNumber : int - port: Int - clients: Map<String, ClientInfo> Use₍) ConnectBtn:JButton - getSetUserBtnListener(): ActionListener - eventQueue: Blocking Queue<String> - userList: Map<String, Socket> host : String QuitBtn:JButton + getUpdateBtnListener(): ActionListener - rand: Random - clients: Map<String, ClientInfo> SetUserBtn:JButton + <constructor> NetUIController(np: NetworkPanel) + getConnectBtnListener(): ActionListener host: String, portNumber: int) - userList: Map<String, Socket> - puzzleGame: puzzleGame UpdateBtn:JButton + getQuitBtnListener(): ActionListener + getChallengeBtnListener() : ActionListener - evHandler: RushHourServerEV ForfeitBtn:JButton + <constructor> RushHourServerEV(eventQueue: + getForfeitBtnListener(): ActionListener BlockingQueue, clients: Map<String, ClientInfo>, + getSetUserBtnListener() : ActionListener LossLabel:JLabel + <constructor> RushHourServer(port:int) userList Map<String,Socket>) + getUpdateBtnListener() : ActionListener OppoLabel: JLabel + run: void + run: void UserLabel:JLabel + connectBtnListener() : ActionListener - createPuzzle(void): PuzzleGame + getOpponentName(): String + quitBtnListener() : ActionListener getList(username: String): void OnlineUsrLabel:JLabel + forfeitListener(): ActionListener send(user: String, message: String): void ClientInfo ∇ + resetPlayingAgainst(): void sendUsername(): boolean + setOnline(online : boolean) : void - challenge(user1: String, user2: String): void WinLabelLabel:JLabel ActionListener updateList(user : String) : void + setPlayingAgainst(playingAgainst : String) : void - acceptChallenge(user1: String, user2: String): void + setPlayingAgainst(player String): + run(): void UserTxtField: JTextField + getName() : String - sendPuzzle(user1: String, user2: String): void + lockButtons(): void processMessage(message: String): void - name : String - puzzleComplete(user1: String, user2: String): void currentOppo: String + isBusy() : boolean + acceptChallenge(message : String) : void - forfeit(user1: String, user2: String): void + unlockButtons(): void - busy : boolean message(message: String): void offlineUser(user: String): void + setOpponentName(user: String): v + getPlayingAgainst(): String declineUser(user1:String, user2: String): void initialise(): void -Extends + getLosses(): int - sendStats(user1:String, user2: String): void + lockConnectBtn(): void - isOnline : boolean + setUsername(name: String): void + resetPlayingAgainst(): void ConnectionHandler + updateWins(wins: Int): void - playingAgainst : String - sock : Socket Thread + getUsername(): String + getWins(): int queue: BlockingQueue<String> Extends + updateLosses(losses: Int): void - info : ClientInfo wins : int + setUserListBox(users:String[]): voi Data and View is: DataInputStream - losses : int + incrementWins(): void + run() : void + incrementLosses(): void + setBusy(busy : boolean) : void

PuzzleGame, GridGame, as these classes are on another page



Animation

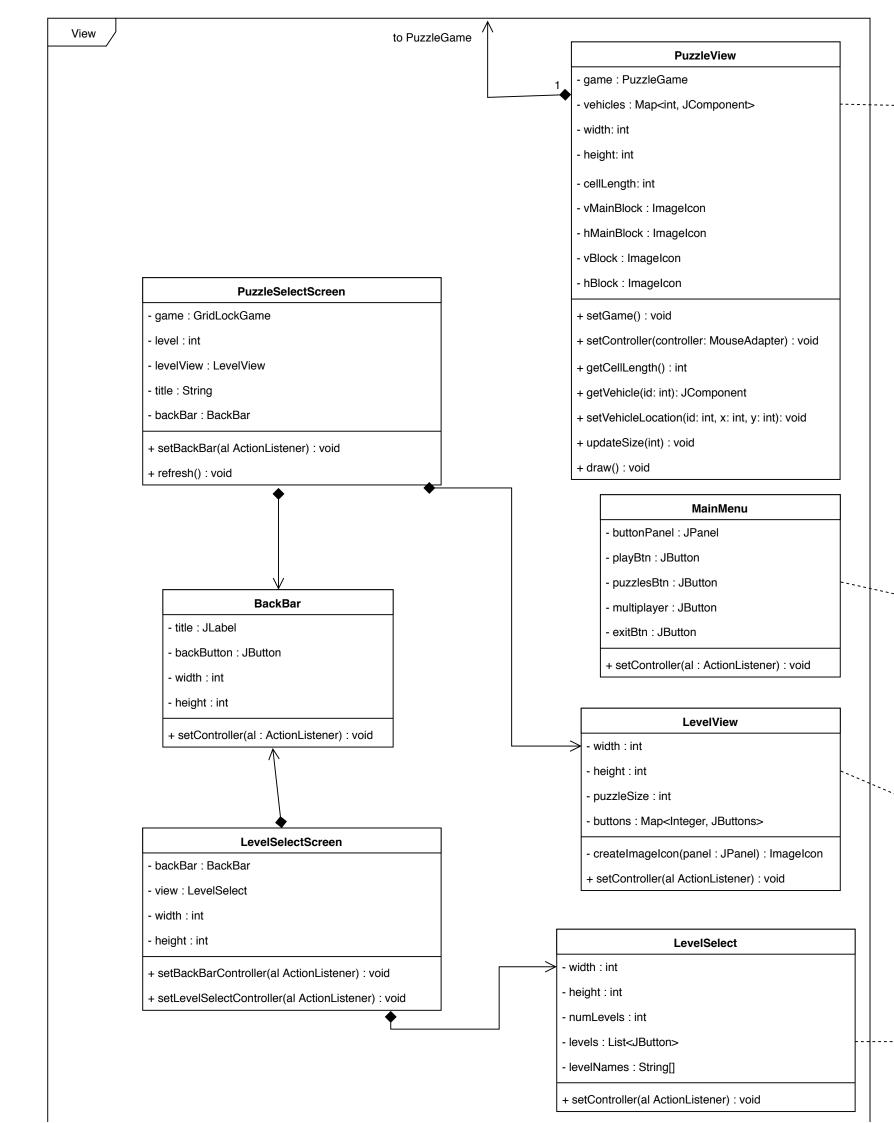
- f : JFrame

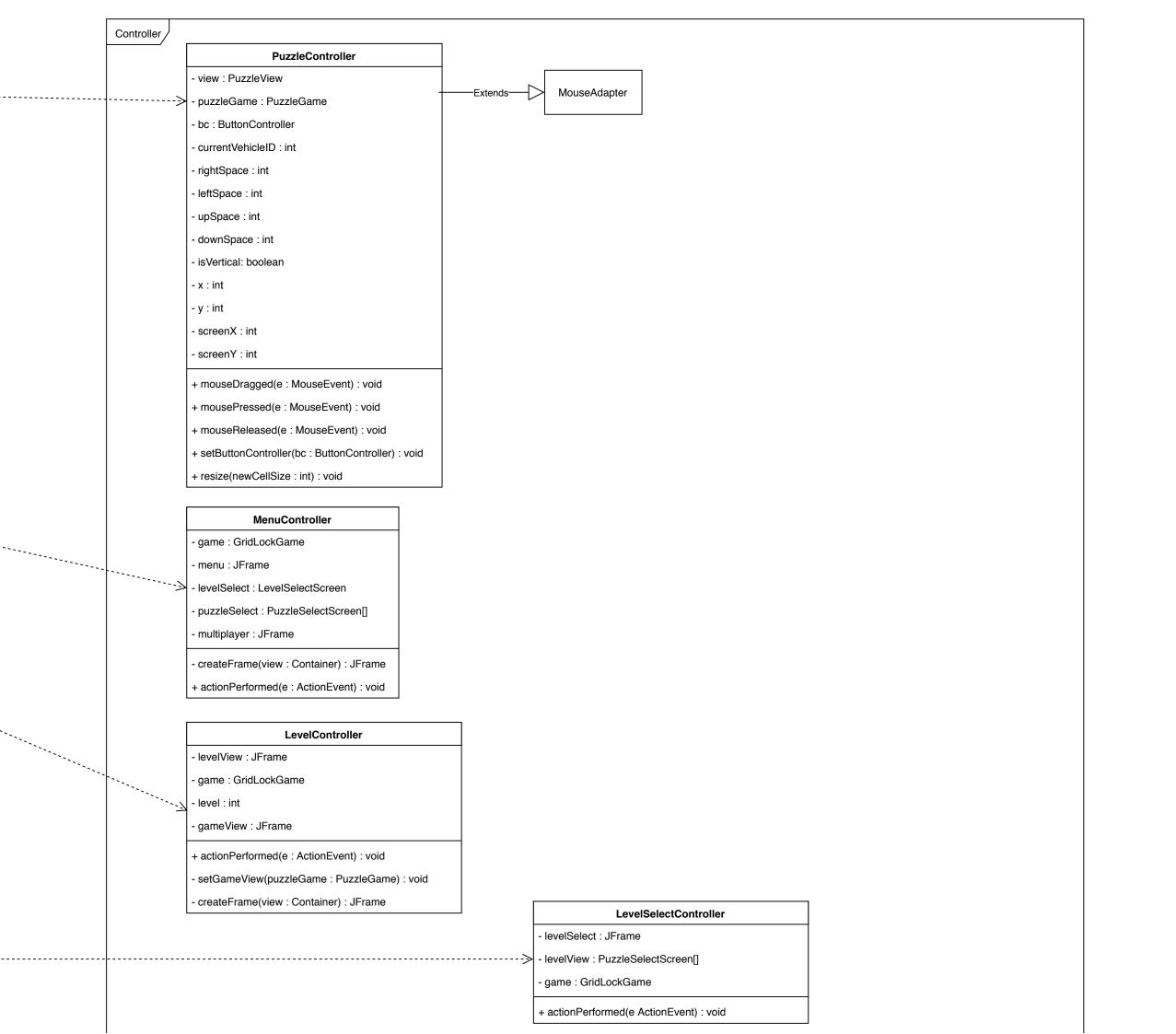
- timer : Timer

- mountain : GeneralPath

- van : GeneralPath

- paintComponent(g:Graphics):void





PuzzleSolvedPopUpScreen

- solvedMessage : JLabel
- okBtn : JButton
- + setOkButtonController(al : ActionListener) : void

<<interface>> Board

- + draw() : void
- + updateSize(cellSize : int) : void

BorderedPuzzleView

- puzzleView : PuzzleView
- + draw() : void
- + updateSize(cell : int) : void



GameView

- buttonPanel : ButtonPanel
- borderedPuzzleView : BorderedPuzzleView
- + getButtonPanel() : ButtonPanel
- + PaintComponent(g Graphics) : void



ButtonPanel

- menuButton : JButton
- MoveCount : JLabel
- redoBtn : JButton
- undoBtn : JButton
- jResetBtn : JButton
- timer : Timer
- + setController(c ButtonController) : void
- + startTimer() : void
- + endTimer() : void
- + displayTime(time : String) : void
- + displayMoves(moves: int) : void

PuzzleSolvedPopUpController - view : PuzzleView - game : GridLockGame level : int puzzleID : int puzzleGame : PuzzleGame currGameScreen : JFrame selectPuzzleScreen: JFrame + mouseReleased(e MouseEvent): void BorderedPuzzleController - borderedPuzzleView : BorderedPuzzleView + resize(newCellSize : int) : void GameController GameListener - gameView : GameView · borderedPuzzleController : BorderedPuzzleController - gameListener : GameListener + componentResized(e ComponentEvent): void + GameController(gameView : GameView, borderedPuzzleController : BorderedPuzzleController) <<interface>> SideButtonController **ButtonController** - puzzleView: PuzzleView + getRedoButtonListener() : ActionListener - puzzleGame: PuzzleGame + getUndoButtonListener() : ActionListener - bp: ButtonPanel + getLoadGameButtonListener() : ActionListener + updateView(): void + getCreateGameButtonListener() : ActionListener + getRedoButtonListener() : ActionListener + getHintButtonListener() : ActionListener + getUndoButtonListener() : ActionListener + getResetButtonListener() : ActionListener + getLoadGameButtonListener() : ActionListener + getTimerListener() : ActionListener + getCreateGameButtonListener() : ActionListener + getMouseAdapter(): MouseAdapter + getRedoButtonListener() : ActionListener + getTimeListener() : ActionListener + getMouseAdapter() : MouseAdapter