

server::Manage
-frame: JFrame -operatorModel: DefaultListModel -operatorList: JList -actionQueueList: JList -historyList: JList -IPTextField: JTextField -roundLabel: JLabel -statusLabel: JLabel -bankLabel: JLabel -incomeLabel: JLabel -timerLabel: JLabel ~director: Director -remaining: long -curRound: int -lastUpdate: long -roundTimer: RoundTimer -format: NumberFormat +registerDisabled: boolean
+Manage(String IP, Director director): ctor +setDirector(Director director): void -initialize(): void +refreshInfo(): void #setCurrentOperator(ServerOperator selected): void -setBankBalance(ServerOperator selected): void -getActionList(ServerOperator selected): Vector<Action> -setActionList(ServerOperator selected): void -copyActionList(ServerOperator selected): void -copyHistoryList(ServerOperator selected): void -setClipboard(Vector<Action> actions): void -getHistoryList(ServerOperator selected): Vector<Action> -setHistoryList(ServerOperator selected): void +addOperator(ServerOperator operator): void +setTimer(int curRound2, int numRounds, String string): void +setRoundTimer(long timeRemaining, int curRound): void +setStatus(String in): void

server::Director
~operators: ServerOperator[] ~grid: Grid +roundLength: long #numRounds: int #curRound = 0: int #gameRunning = true: boolean #winner = null: ServerOperator +timeStep: int ~timerHandler: EndRoundTimer -numOperators: int +manage: Manage
+Director(Manage manage, File xmlFile): ctor -loadData(File xmlFile): Grid +newClient(ServerMessages client, boolean register, String name, String password): void -sendTopOfGrid(ServerOperator client): void -sendGameInfo(ServerMessages client): void #unscheduledEndRound(): void #endRound(): void +setupAuction(): void +processSeismicRequests(): void +processDrillRequests(): void +changeRoundLength(long length): void +setRoundTime(long length): void +setNumRounds(int number): void +getNumRounds(): int +setTimeStep(int amount): void +getGrid(): Grid +serverStop(): void +endGame(): void +getWinner(): ServerOperator

server::Auction
~bids: LinkedList<Bid>[] -grid: Grid -priorityList: Vector<PriorityQueue<Bid>»
+Auction(LinkedList<Bid>[] bids, Grid grid): ctor +auction(): void -getQueue(Bid bid): PriorityQueue<Bid> +getResults(): LinkedList<Bid>

server::EndRoundTimer
-director: Director -toolkit: Toolkit -roundLength: long
+EndRoundTimer(Director director, long roundLength): ctor +run(): void +getTimeRemaining(): long

server::Server
+mainDirector: Director +localIP: InetAddress
+Server(Manage window, File file): ctor +main(String[] args): void +run(): void -waitConnections(): void

server::ServerOperator
#history: LinkedList<Action> +password: String
+ServerOperator(Director d, String name, String password, ServerMessages client): ctor +wonBid(Bid won): void +lostBid(Bid won): void +sendInfo(String message): void +sendExistingData(): void +sendGrid(): void +sendQueues(): void +sendSeismicLayer(Point p, Integer[] layer): void +sendDrillStuffs(Point p, String drillSocket, Integer[] gas, Integer[] oil, String gasSocket, String oilSocket, String rockSocket, LithologicType[] rock): void +sendDrillStuffs(String drillSocket, String cellToSocket): void +sendEndGame(ServerOperator winner): void +addHistory(Action s): void +getHistory(): LinkedList<? extends Action> +endRound(long timeRemaining, long roundLength, int curRound, int numRounds): void +setOwnership(Bid w): void +addIncome(int gasRate, int oilRate): void +setName(String newName): void +setBalance(int newBalance): void +getBalance(): int +sendWarning(String msg): void +toString(): String

server::ServerMessages
~director: Director
+ServerMessages(Director inDirector, Socket sock): ctor +parse(String in, Prefix pr): void +wonBid(String bid): void +lostBid(String bid): void +sendBidLimit(int amount): void +startTimer(long milliseconds, long roundLength, int roundNum, int numRounds): void +sendNewRound(int balance): void +endRound(String message): void +sendBankBalance(int amount): void +sendInfo(String message): void +sendWarning(String message): void +sendGridLimits(int xLimit, int yLimit, int numLayers, int[] layers): void +sendSeismicCosts(String in): void +sendDrillCost(int drillCost): void +sendLayer(String layers): void +sendGas(String gas): void +sendOil(String oil): void +sendRock(String rockSocket): void +sendCell(String c): void +sendDrill(String drillSocket): void +sendBidQueue(String socket): void +sendSeismicQueue(String socket): void +sendDrillQueue(String socket): void +sendOwner(String socket): void +sendEndGame(String socket): void

server::ParseSimulationXML
~elementMap: Map<String, Element> ~parentMap: Map<Element, Element> ~stack: Stack<Element> -grid: Grid -gridStart: boolean -numLayer: int -numRow: int -numCol: int -curLayer = 0: int -xmlFile: File
+ParseSimulationXML(File xmlFile, Grid grid): ctor +startElement(String uri, String localName, String qName, Attributes attributes): void +endElement(String uri, String localName, String qName): void +characters(char ch[], int start, int length): void +endDocument(): void +fatalError(SAXParseException e): void