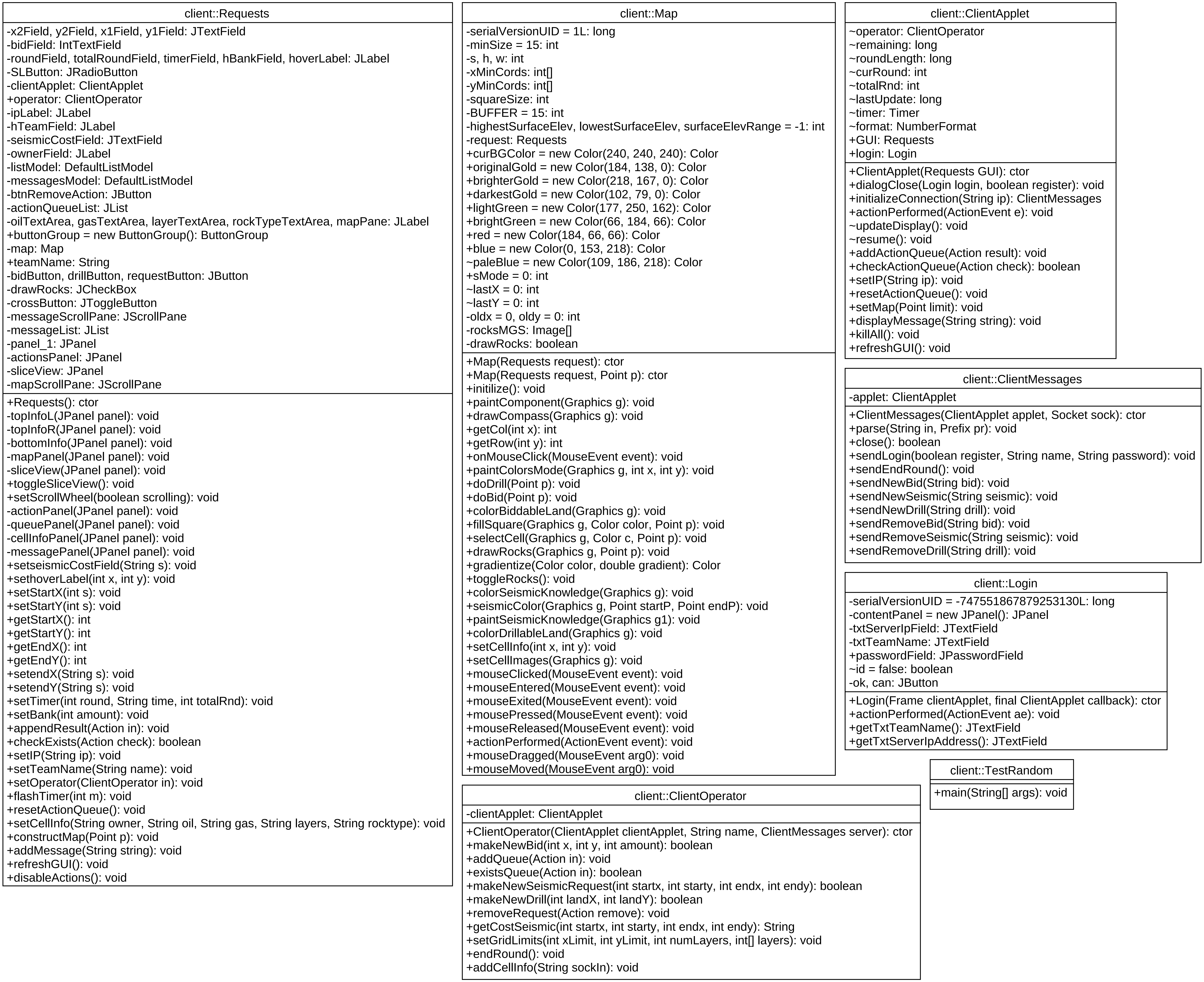

client::Requests

```
-x2Field, y2Field, x1Field, y1Field: JTextField
-bidField: IntTextField
-roundField, totalRoundField, timerField, hBankField, hoverLabel: JLabel
-SLButton: JRadioButton
-clientApplet: ClientApplet
+operator: ClientOperator
-ipLabel: JLabel
-hTeamField: JLabel
-seismicCostField: JTextField
-ownerField: JLabel
-listModel: DefaultListModel
-messagesModel: DefaultListModel
-btnRemoveAction: JButton
-actionQueueList: JList
-oilTextArea, gasTextArea, layerTextArea, rockTypeTextArea, mapPane: JLabel
+buttonGroup = new ButtonGroup(): ButtonGroup
-map: Map
+teamName: String
-bidButton, drillButton, requestButton: JButton
-drawRocks: JCheckBox
-crossButton: JToggleButton
-messageScrollPane: JScrollPane
-messageList: JList
-panel_1: JPanel
-actionsPanel: JPanel
-sliceView: JPanel
-mapScrollPane: JScrollPane

+Requests(): ctor
-topInfoL(JPanel panel): void
-topInfoR(JPanel panel): void
-bottomInfo(JPanel panel): void
-mapPanel(JPanel panel): void
-sliceView(JPanel panel): void
+toggleSliceView(): void
+setScrollWheel(boolean scrolling): void
-actionPanel(JPanel panel): void
-queuePanel(JPanel panel): void
-cellInfoPanel(JPanel panel): void
-messagePanel(JPanel panel): void
+setseismicCostField(String s): void
+sethoverLabel(int x, int y): void
+setStartX(int s): void
+setStartY(int s): void
+getStartX(): int
+getStartY(): int
+getEndX(): int
+getEndY(): int
+setendX(String s): void
+setendY(String s): void
+setTimer(int round, String time, int totalRnd): void
+setBank(int amount): void
+appendResult(Action in): void
+checkExists(Action check): boolean
+setIP(String ip): void
+setTeamName(String name): void
+setOperator(ClientOperator in): void
+flashTimer(int m): void
+resetActionQueue(): void
+setCellInfo(String owner, String oil, String gas, String layers, String rocktype): void
+constructMap(Point p): void
+addMessage(String string): void
+refreshGUI(): void
+disableActions(): void
```

client::Map

```
-serialVersionUID = 1L: long
-minSize = 15: int
-s, h, w: int
-xMinCords: int[]
-yMinCords: int[]
-squareSize: int
-BUFFER = 15: int
-highestSurfaceElev, lowestSurfaceElev, surfaceElevRange = -1: int
-request: Requests
+curBGColor = new Color(240, 240, 240): Color
+originalGold = new Color(184, 138, 0): Color
+brighterGold = new Color(218, 167, 0): Color
+darkestGold = new Color(102, 79, 0): Color
+lightGreen = new Color(177, 250, 162): Color
+brightGreen = new Color(66, 184, 66): Color
+red = new Color(184, 66, 66): Color
+blue = new Color(0, 153, 218): Color
~paleBlue = new Color(109, 186, 218): Color
+sMode = 0: int
~lastX = 0: int
~lastY = 0: int
-oldx = 0, oldy = 0: int
-rocksMGS: Image[]
-drawRocks: boolean

+Map(Requests request): ctor
+Map(Requests request, Point p): ctor
+initilize(): void
+paintComponent(Graphics g): void
+drawCompass(Graphics g): void
+getCol(int x): int
+getRow(int y): int
+onMouseClick(MouseEvent event): void
+paintColorsMode(Graphics g, int x, int y): void
+doDrill(Point p): void
+doBid(Point p): void
+colorBiddableLand(Graphics g): void
+fillSquare(Graphics g, Color color, Point p): void
+selectCell(Graphics g, Color c, Point p): void
+drawRocks(Graphics g, Point p): void
+gradientize(Color color, double gradient): Color
+toggleRocks(): void
+colorSeismicKnowledge(Graphics g): void
+seismicColor(Graphics g, Point startP, Point endP): void
+paintSeismicKnowledge(Graphics g1): void
+colorDrillableLand(Graphics g): void
+setCellInfo(int x, int y): void
+setCellImages(Graphics g): void
+mouseClicked(MouseEvent event): void
+mouseEntered(MouseEvent event): void
+mouseExited(MouseEvent event): void
+mousePressed(MouseEvent event): void
+mouseReleased(MouseEvent event): void
+actionPerformed(ActionEvent event): void
+mouseDragged(MouseEvent arg0): void
+mouseMoved(MouseEvent arg0): void
```

client::ClientOperator

```
-clientApplet: ClientApplet

+ClientOperator(ClientApplet clientApplet, String name, ClientMessages server): ctor
+makeNewBid(int x, int y, int amount): boolean
+addQueue(Action in): void
+existsQueue(Action in): boolean
+makeNewSeismicRequest(int startx, int starty, int endx, int endy): boolean
+makeNewDrill(int landX, int landY): boolean
+removeRequest(Action remove): void
+getCostSeismic(int startx, int starty, int endx, int endy): String
+setGridLimits(int xLimit, int yLimit, int numLayers, int[] layers): void
+endRound(): void
+addCellInfo(String sockIn): void
```

client::ClientApplet

```
~operator: ClientOperator
~remaining: long
~roundLength: long
~curRound: int
~totalRnd: int
~lastUpdate: long
~timer: Timer
~format: NumberFormat
+GUI: Requests
+login: Login

+ClientApplet(Requests GUI): ctor
+dialogClose(Login login, boolean register): void
+initializeConnection(String ip): ClientMessages
+actionPerformed(ActionEvent e): void
~updateDisplay(): void
~resume(): void
+addActionQueue(Action result): void
+checkActionQueue(Action check): boolean
+setIP(String ip): void
+resetActionQueue(): void
+setMap(Point limit): void
+displayMessage(String string): void
+killAll(): void
+refreshGUI(): void
```

client::ClientMessages

```
-applet: ClientApplet

+ClientMessages(ClientApplet applet, Socket sock): ctor
+parse(String in, Prefix pr): void
+close(): boolean
+sendLogin(boolean register, String name, String password): void
+sendEndRound(): void
+sendNewBid(String bid): void
+sendNewSeismic(String seismic): void
+sendNewDrill(String drill): void
+sendRemoveBid(String bid): void
+sendRemoveSeismic(String seismic): void
+sendRemoveDrill(String drill): void
```

client::Login

```
-serialVersionUID = -747551867879253130L: long
-contentPanel = new JPanel(): JPanel
-txtServerIpField: JTextField
-txtTeamName: JTextField
+passwordField: JPasswordField
~id = false: boolean
-ok, can: JButton

+Login(Frame clientApplet, final ClientApplet callback): ctor
+actionPerformed(ActionEvent ae): void
+getTxtTeamName(): JTextField
+getTxtServerIpAddress(): JTextField
```

client::TestRandom

```
+main(String[] args): void
```