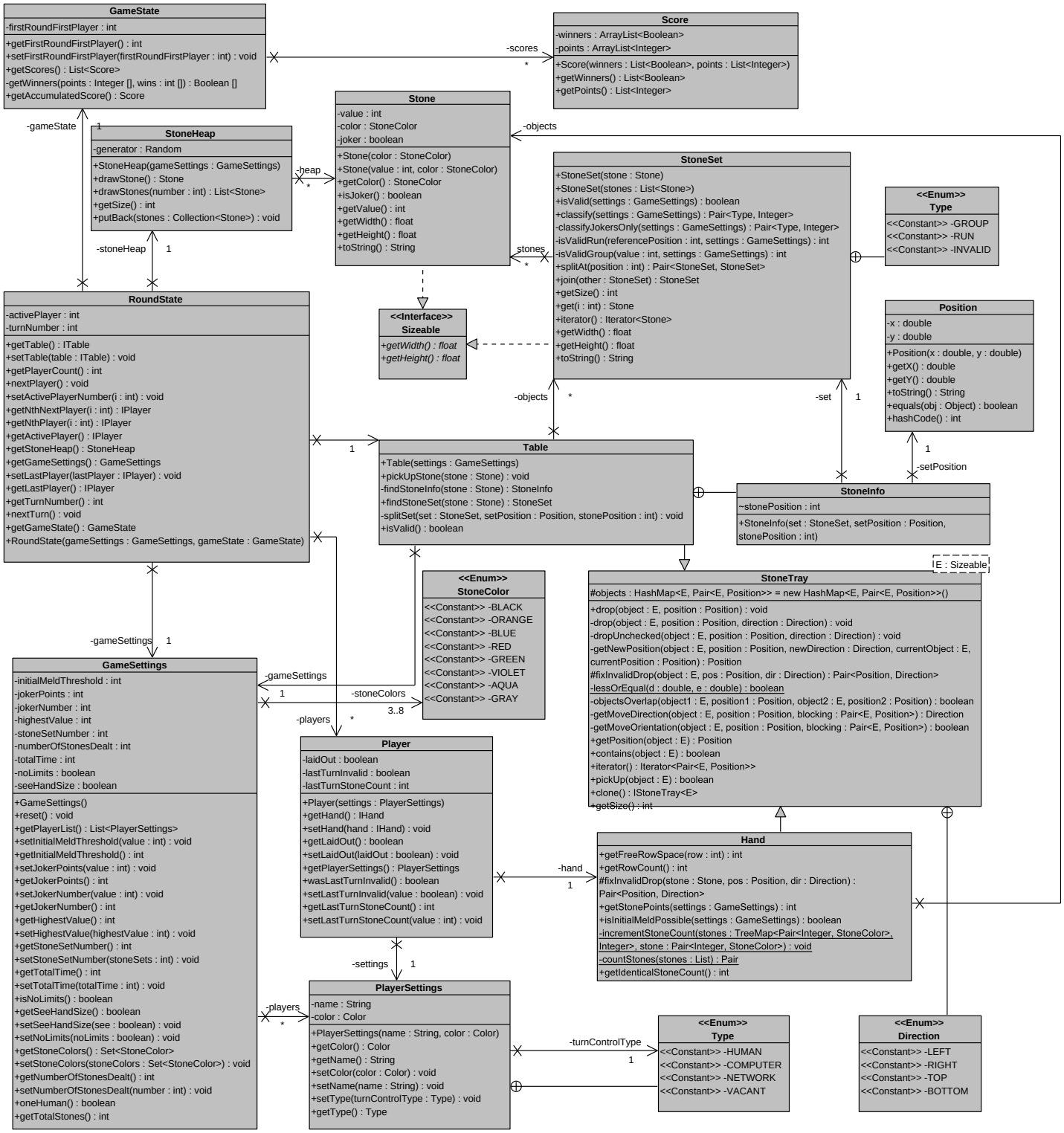
GameState
-firstRoundFirstPlayer : int
+getFirstRoundFirstPlayer() : int
+setFirstRoundFirstPlayer(firstRoundFirstPlayer : int) : void
+getScores() : List<Score>
-getWinners(points : Integer [], wins : int []) : Boolean []
+getAccumulatedScore() : Score
-scores
*
Score
-winners : ArrayList<Boolean>
-points : ArrayList<Integer>
+Score(winners : List<Boolean>, points : List<Integer>)
+getWinners() : List<Boolean>
+getPoints() : List<Integer>
-gameState
1
StoneHeap
-generator : Random
+StoneHeap(gameSettings : GameSettings)
+drawStone() : Stone
+drawStones(number : int) : List<Stone>
+getSize() : int
+putBack(stones : Collection<Stone>) : void
-heap
*
Stone
-value : int
-color : StoneColor
-joker : boolean
+Stone(color : StoneColor)
+Stone(value : int, color : StoneColor)
+getColor() : StoneColor
+isJoker() : boolean
+getValue() : int
+getWidth() : float
+getHeight() : float
+toString() : String
-objects
StoneSet
+StoneSet(stone : Stone)
+StoneSet(stones : List<Stone>)
+isValid(settings : GameSettings) : boolean
+classify(settings : GameSettings) : Pair<Type, Integer>
-classifyJokersOnly(settings : GameSettings) : Pair<Type, Integer>
-isValidRun(referencePosition : int, settings : GameSettings) : int
-isValidGroup(value : int, settings : GameSettings) : int
+splitAt(position : int) : Pair<StoneSet, StoneSet>
+join(other : StoneSet) : StoneSet
+getSize() : int
+get(i : int) : Stone
+iterator() : Iterator<Stone>
+getWidth() : float
+getHeight() : float
+toString() : String
<<Enum>>
Type
<<Constant>> -GROUP
<<Constant>> -RUN
<<Constant>> -INVALID
-stoneHeap
1
stones
*
<<Interface>>
Sizeable
+getWidth() : float
+getHeight() : float
Position
-x : double
-y : double
+Position(x : double, y : double)
+getX() : double
+getY() : double
+toString() : String
+equals(obj : Object) : boolean
+hashCode() : int
RoundState
-activePlayer : int
-turnNumber : int
+getTable() : ITable
+setTable(table : ITable) : void
+getPlayerCount() : int
+nextPlayer() : void
+setActivePlayerNumber(i : int) : void
+getNthNextPlayer(i : int) : IPlayer
+getNthPlayer(i : int) : IPlayer
+getActivePlayer() : IPlayer
+getStoneHeap() : StoneHeap
+getGameSettings() : GameSettings
+setLastPlayer(lastPlayer : IPlayer) : void
+getLastPlayer() : IPlayer
+getTurnNumber() : int
+nextTurn() : void
+getGameState() : GameState
+RoundState(gameSettings : GameSettings, gameState : GameState)
-objects
*
-set
1
1
Table
+Table(settings : GameSettings)
+pickUpStone(stone : Stone) : void
-findStoneInfo(stone : Stone) : StoneInfo
+findStoneSet(stone : Stone) : StoneSet
-splitSet(set : StoneSet, setPosition : Position, stonePosition : int) : void
+isValid() : boolean
1
-setPosition
StoneInfo
~stonePosition : int
+StoneInfo(set : StoneSet, setPosition : Position, stonePosition : int)
E : Sizeable
StoneTray
#objects : HashMap<E, Pair<E, Position>> = new HashMap<E, Pair<E, Position>>()
+drop(object : E, position : Position) : void
-drop(object : E, position : Position, direction : Direction) : void
-dropUnchecked(object : E, position : Position, direction : Direction) : void
-getNewPosition(object : E, position : Position, newDirection : Direction, currentObject : E, currentPosition : Position) : Position
#fixInvalidDrop(object : E, pos : Position, dir : Direction) : Pair<Position, Direction>
-lessOrEqual(d : double, e : double) : boolean
-objectsOverlap(object1 : E, position1 : Position, object2 : E, position2 : Position) : boolean
-getMoveDirection(object : E, position : Position, blocking : Pair<E, Position>) : Direction
-getMoveOrientation(object : E, position : Position, blocking : Pair<E, Position>) : boolean
+getPosition(object : E) : Position
+contains(object : E) : boolean
+iterator() : Iterator<Pair<E, Position>>
+pickUp(object : E) : boolean
+clone() : IStoneTray<E>
+getSize() : int
-gameSettings
1
<<Enum>>
StoneColor
<<Constant>> -BLACK
<<Constant>> -ORANGE
<<Constant>> -BLUE
<<Constant>> -RED
<<Constant>> -GREEN
<<Constant>> -VIOLET
<<Constant>> -AQUA
<<Constant>> -GRAY
GameSettings
-initialMeldThreshold : int
-jokerPoints : int
-jokerNumber : int
-highestValue : int
-stoneSetNumber : int
-numberOfStonesDealt : int
-totalTime : int
-noLimits : boolean
-seeHandSize : boolean
+GameSettings()
+reset() : void
+getPlayerList() : List<PlayerSettings>
+setInitialMeldThreshold(value : int) : void
+getInitialMeldThreshold() : int
+setJokerPoints(value : int) : void
+getJokerPoints() : int
+setJokerNumber(value : int) : void
+getJokerNumber() : int
+getHighestValue() : int
+setHighestValue(highestValue : int) : void
+getStoneSetNumber() : int
+setStoneSetNumber(stoneSets : int) : void
+getTotalTime() : int
+setTotalTime(totalTime : int) : void
+isNoLimits() : boolean
+getSeeHandSize() : boolean
+setSeeHandSize(see : boolean) : void
+setNoLimits(noLimits : boolean) : void
+getStoneColors() : Set<StoneColor>
+setStoneColors(stoneColors : Set<StoneColor>) : void
+getNumberOfStonesDealt() : int
+setNumberOfStonesDealt(number : int) : void
+oneHuman() : boolean
+getTotalStones() : int
-gameSettings
1
-stoneColors
3..8
-players
*
Player
-laidOut : boolean
-lastTurnInvalid : boolean
-lastTurnStoneCount : int
+Player(settings : PlayerSettings)
+getHand() : IHand
+setHand(hand : IHand) : void
+getLaidOut() : boolean
+setLaidOut(laidOut : boolean) : void
+getPlayerSettings() : PlayerSettings
+wasLastTurnInvalid() : boolean
+setLastTurnInvalid(value : boolean) : void
+getLastTurnStoneCount() : int
+setLastTurnStoneCount(value : int) : void
-hand
1
Hand
+getFreeRowSpace(row : int) : int
+getRowCount() : int
#fixInvalidDrop(stone : Stone, pos : Position, dir : Direction) : Pair<Position, Direction>
+getStonePoints(settings : GameSettings) : int
+isInitialMeldPossible(settings : GameSettings) : boolean
-incrementStoneCount(stones : TreeMap<Pair<Integer, StoneColor>, Integer>, stone : Pair<Integer, StoneColor>) : void
-countStones(stones : List) : Pair
+getIdenticalStoneCount() : int
-settings
1
-players
*
PlayerSettings
-name : String
-color : Color
+PlayerSettings(name : String, color : Color)
+getColor() : Color
+getName() : String
+setColor(color : Color) : void
+setName(name : String) : void
+setType(turnControlType : Type) : void
+getType() : Type
-turnControlType
1
<<Enum>>
Type
<<Constant>> -HUMAN
<<Constant>> -COMPUTER
<<Constant>> -NETWORK
<<Constant>> -VACANT
<<Enum>>
Direction
<<Constant>> -LEFT
<<Constant>> -RIGHT
<<Constant>> -TOP
<<Constant>> -BOTTOM