Network

Protocol
+ encodePacket(msg : String, ip : InetAdress, port : int) : DatagramPacket
+ decodePacket(packet : DatagramPacket) : String[]
+ encodeConnect(username : String) : String
+ encodeBoardPiece(bp : BoardPiece) : String
+ decodeBoardPiece(msg : String[]) : BoardPiece
+ encodeGoalCard(Stack<Box> goal, Box currentGoal) : String
+ decodeGoalCard(msg : String[]) : Stack<Box>
+ encodeTime(count : Countdown) : String
+ decodeTime(msg : String[]) : int
+ encodeProposition(username : String , nb : int) : String
+ encodeHand(bool : boolean) : String
+ decodeHand(msg : String[]) : boolean
+ encodeClient(players : List<Player>) : String
+ decodeClient(msg : String[]) : String
+ encodeMove(r : Robot, round : int) : String
+ encodeAskNextProposition() : String
+ encodeMessage(msg : String) : String
+ encodeOtherUserName(username : String) : String
+ encodeButtonValidate(buttonValidate : boolean) : String
+ encodeServerISDisconnect() : String

Client
- ipServer : InetAdress
- socket : DatagramSocket
- username : String
- hand : boolean
- bestProposition : int
- run : boolean
- controller : Controller
- count : Countdown
- clientThread : Thread

- listen()
- process(data : String)
+ connect()
+ stopClient()
+ processData(data : String[])
+ sendMove(newRound : int)
+ sendPropositionfromPlayer(username : String, i : Integer)
+ askNextProposition()

Server
- game : Game
- manager : PlayerManager
- socket : DatagramSocket
- controller : Controller
- round : int
- bestProposition : int
- hand : boolean
- run : boolean
- startCount : boolean
- buttonValidate : boolean
- username : String
- count : Countdown
- serverThread : Thread

+ stopServer()
- listen()
- process(data : String)
+ updatePlayers()
+ addPoint()
+ sendBoardToAllClients()
+ sendBoard(ip : InetAddress)
- sendOtherUserName(ip : InetAddress,username : String)
- sendCard(ip : InetAddress)
- sendRobots(ip : InetAddress)
+ sendTime(ip : InetAddress)
+ sendTimeToAllClients()
+ sendRefresh(ip : InetAddress)
+ sendUpdates()
+ updatePropositionfromPlayer()
+ startPlay()
- sendHands(ip : InetAddress, bool : boolean)
+ sendbestProposition(proposition : int)
+ resetPropositions()
+ sendMessage(msg : String)
+ nextPlayerProposition()
+ replaceRobotAtStart()
+ setButtonValidate(buttonValidate : boolean)
+ sendServerDisconnect()

CountDown
- timer : Timer
- time : int
- controller : Controller

+ createCountDown()
+ startCountDown()
+ stopCountDown()
+ resetCountDown()

PlayerManager
- players : Player[]
- currentHand : int

+ addPlayer(p : Player)
+ chooseHand()
+ sortPropositions(p : Player, prop : int)
+ deletePlayer(p : Player)
+ sortPropositions(user : String) : Player
+ doesEveryoneProposed() : boolean

userInterface

RasendeFrame
+ buildFrame (controller : controller)
+ buildJMenu(controller : controller)
- buildColumn(controller : controller)
+ refresh(g : Game)
+ println(s : String)

<<Interface>> RasendeViewInterface
+ display(game : Game,c : Controller)
+ displayDataInfo(game : Game,c : Controller)
+ displayPlayers(string : String)
+ displayBoard(game : Game)
+ displayHelp()
+ displayWin()
+ displayMoveLimit()
+ displayLicense()
+ displayConnectionLost()
+ displayStartServer() : String
+ displayJoinServer() : String[]
+ setFocusOnBoard()
+ println(string : String)
+ setOnlinePerspective(online : Boolean)
+ setEnabledForfeit(enabled : Boolean)
+ setEnabledValidate(enabled : Boolean)
+ getSuggestion() : int
+ dispose()

Solver

Solver
- tree : StructTree
- root : String
- leaf : String
- game : Game
- depth : int
- cut : int
- solved : boolean

+ initsolver(g : Game)
+ solve(g : Game)
+ createRobot()
- buildPossibilities() : boolean
- encodeKey(robots : List<Robots>) : String
- decodeKey(key:String) : List<Robot>

StructTree
+ possibilities : Map<String, Integer>
+ parent : Map<String, Integer>
+ solutionStack : Stack<String>

+ clear()
+ containsKey(key : String) : boolean
+ addParent(key : String,keyParent : String)
+ addPossibility(key : String,depth : int)
+ getKeyFromValue(depth : int) : Map<String, Integer>
- buildStack(leaf : String,root : String)
- findDirection(after : String,before : String) : String

Controller
- game : Game
- frame :RasendeViewInterface
- server : Server
- client : Client

- game : Game
- frame : RasendeFrame
- sever : Server
- client : Client
- moveRobotInDirection(dir: Direction)
- askToQuit()
+ moveLimit()
+ timeLimit()
+ gameOver()
- canMove() : boolean
- actionPerformed(e : ActionEvent)
- mouseClicked(e : MouseEvent)
- keyPresse(e : KeyEvent)
- checkEndRound(round : int)
- write(arg0 : int) throws IOException

Game

Box
- north : boolean
- east : boolean
- south : boolean
- west : boolean
- x, y : int
- color : Color
- type : BoxType

+ setWall(dir : Direction)
+ canContinue(dir : Direction) : boolean
- getImageIcon(Game game) : ImageIncon
+ getJPanel(Game game, int i, int j) : Jpannel

Game
- mBoard : Board
- mRobots : List<Robot>
- mPreviousPosition : Stack<List><robot>
- mNextPosition : Stack<List><robot>
- mSelectedRobot : Robot
- mGoalCards : Stack<Box>
- currentGoal : Box
- mSolver : Solver
- isOver : boolean
- theme : String

+ startNewGame()
+ placeRobots()
- generateBoard()
+ startNewLap()
- saveCurrentPosition(rob : Robot)
+ loadPreviousPosition()
+ loadNextposition()
+ moveSelectedRobot(dir : Direction)
+ isWin(rob : Robot) : boolean

Constant
+ <<enum>> BoxType
+ <<enum>> Color
+ <<enum>> Direction
-NomMembre

Robot
+ x, y, originX, originY : int
- color : Color

+ placeOnBoard(robots : List<Robot>)
+ robotIsHere(robots : List<Robot>) : boolean
+ equalsObject(r : Object) : boolean
+ newOrigin()

Board
- gameBoard : Box[][]
- numBoardPieces : List<Integer>
- boardPiecesSide[] : char

+ initBoard()
- initBoardPieces()
- randomizeNumBoardPieces()
- randomizeBoardPiecesSide()
+ setBoard()
+ putSurroundingWalls()
- putCentralBoxes()
+ oppositeWall()
+ getNewPosition(robot:Robot, direction : Direction, robots :List<Robot>) : boolean

BoardPiece
- xmlFile : File
- sxb : SAXBuilder
- initialLocation : int
- finalLocation : int
- boxes : List<Box>

+ initBoardPiece()
- rotateBoxes()
+ rotateRight()
+ rotateLeft()
- rotateWallsRight(box:Box)
- rotateWallsLeft(box:Box)
- adjustBoxes()
- adjustY()
- adjustX()
- extractBoxInfo(elem : Element)

Player
- ip : InetAddress
- pseudo : String
- points : int
- nbMoveProposed : int
- host : boolean
- hand : boolean