CShipPlacementController

- m_RNG:Random
- m_width:int
- m_height:int
- m_board:BoardState[]
- m_numBattleship:int
- m_BattleshipSize:int
- m_numCruiser:int
- m_CruiserSize:int
- m_numDestroyer:int
- m_DestroyerSize:int
- m_numSubmarines:int
- m_SubmarineSize:int

- CShipPlacementController(width:int,height:...
- resetBoard():void
- getRandomNumber(low:int,high:int):int
- getRandomDirection():Direction
- markBusy(x:int,y:int):void
- surroundWithBusy():void
- tryToPlace(x:int,y:int,dir:Direction):boolean
- placeShips(numShips:int,shipSize:int):void
- initShips():void
- getNewBoard():FieldState[]

CBattleshipGUI

- m_playingFieldLayout:GridLayout
- m_windowLayout:BorderLayout
- m_panel:JPanel
- m_controllerListener:Thread
- m_statusMsg:JLabel

- CBattleshipGUI(control:CPlayingFieldController)
- setEnemyPlayingField(stateVec:CPlayingFieldController.FieldState[]):v...
- setOwnPlayingField(stateVec:CPlayingFieldController.FieldState[]):void
- actionPerformed(ae:ActionEvent):void

CPlayingFieldPanel

- m_width:int
- m_height:int
- m_button:JButton[]
- m_layout:GridLayout
- m_panel:JPanel

- CPlayingFieldPanel(width:int,height:int,al:ActionListen...
- setState(x:int,y:int,s:FieldState):void
- setState(stateArray:FieldState[]):void
- applyButtonDesign(x:int,y:int,s:FieldState):void
- applyButtonDesign(pos:int,s:FieldState):void
- enable(x:int,y:int):void
- disable(states:FieldState[]):void
- getPanel():JPanel

CPlayingFieldController

- m_enemyState:FieldState[]
- m_ownState:FieldState[]
- m_width:int
- m_height:int
- m_inStream:BufferedReader
- m_outStream:BufferedWriter
- m_socket:Socket
- m_port:int
- m_host:String
- m_updateAvailable:boolean
- m_itsMyTurn:boolean
- m_gameState:GameState

- CPlayingFieldController(width:int,height:int,host:String,port:int)
- initShips():void
- getGameState():GameState
- getEnemyStateVec():FieldState[]
- getOwnStateVec():FieldState[]
- run():void
- areAnyShipsLefts():boolean
- isShipCompletlyDestroyed(x:int,y:int,board:FieldState[],oldX:int,oldY:int):boole...
- markWholeShipAsDestroyed(x:int,y:int,board:FieldState[]):void
- handleOp1(paramList:String):String
- handleOp2(paramList:String):String
- handleIncomingMessage(paramList:String):String
- defend():void
- attack(x:int,y:int):void
- send(msg:String):void
- getWidth():int
- getHeight():int
- getUpdatedFields():List
- isItMyTurn():boolean

«Singleton»
CMessageGenerator

- m_me:CMessageGenerator

- CMessageGenerator()
- attack(x:int,y:int):String
- respondAttack(x:int,y:int,s:FieldState):String
- defendFirst():String
- attackFirst():String
- generateStartSignal():String
- parseState(code:int):FieldState
- parseState(s:FieldState):int
- parseCommand(code:int):CommandState
- parseCommand(s:CommandState):int

1

2

1

1