ModelControllerDiagram

Model

Controller

Model

General: Model+View

View

Utils

ClientController
(from Model2)
- -commandController: CommandController
- -choiceController: ChoiceController
- +view: View
- +client: Client
- +clientModel: ClientModel
- +update(playerMessage: PlayerMessage)
- +updateBoard(moveMessage: moveMessage)
- +reconnection()

Observable
- +observers: Observer<T>
- +register(observer: Observer<T>)
- +deregister(observer: Observer<T>)
- //notify(message: T)

«interface»
Observer
- +update(message: T): void

ClientModel
(from Model)
- -clientBoard: ClientBoard
- -actualPlayer: Player
- -privateObjCard: PrivateObjCard
- -actualUser: User
- +setClientBoard(clientBoard: ClientBoard): void
- +setPrivateObjCard(): void
- +setActualPlayer(actualPlayer: Player): void
- +setActualUser(actualUser: User): void

Client
(from Model2)
- -clientStrategy
- -view: View
- -thisUser: User
- -nickname: String
- +Client(clientStrategy: ClientStrategy, view: View)
- +setNickname(nickname: String): void
- +connect(): void
- +update(message: T): void
- +reconnect(): void
- +send(playerMessage: PlayerMessage): void
- +close(): void

View
- -inputStrategy: InputStrategy
- -outputStrategy: OutputStrategy
- -clientModel: ClientModel
- +View(inputStrategy: InputStrategy, outputStrategy: OutputStrategy)
- +receive(playerMessage: PlayerMessage): void
- +reportError(id: int): void
- +showMessage(message: int): void
- +update(clientModel: ClientModel): void
- -reconnection(): void

PlayerSetupper
- -user: User
- -playerChoice: PlayerChoice
- -commandTypeEnum: CommandTypeEnum
- -messageMap: HashMap
- +playerSetupper()
- +newChoiceReceived(playerChoice: PlayerChoice): TypeOfInputAsked
- +newUserReceived(user: User): TypeOfInputAsked
- +setCommandType(): void
- +validCommand(inputReceived: String): TypeOfInputAsked
- +setFrameColour(inputReceived: String): TypeOfInputAsked
- +setCompleteMove(): void
- -setSchema(inputReceived: String): void
- +validNickname(inputReceived: String): TypeOfInputAsked
- -reconnect(message: String): void

«interface»
OutputStrategy
- +showGameboard(clientModel: ClientModel)
- +showScore(playerList: List<Player>)

«interface»
InputStrategy
- +yourTurn(clientBoard: ClientTBoard, playerMove: PlayerMove)
- +showMessage(idError: int)
- +showError(idError: int)
- +getSyntaxController(): SyntaxController
- +getPlayerSetupper(): PlayerSetupper
- +makeChoice(playerChoice: PlayerChoice)
- +choseNickname(nickname: String)
- +tryReconnection()
- +validPath()

«enumeration»
InputFormatEnum

TypeOfInputAsked
- -commandTypeEnum: CommandTypeEnum
- -inputFormat: InputFormat
- +TypeOfInputAsked()
- +Operation1()

SyntaxController
- -clientBoard: ClientBoard
- -playerMove: PlayerMove
- -comandTypeEnum: CommandTypeEnum
- -inputFormatEnum: InputFormatEnum
- -receivedPlayerMove: PlayerMove
- -commandTypeEnumLists: List<List<CommandTypeEnum>>
- +SyntaxController()
- +newMoveReceived(receivedPlayerMove: PlayerMove, clientBoard: ClientBoard): TypeOfInputAsked
- +validCommand(inputReceived: String): TypeOfInputAsked
- -setToolCardId(inputReceived: String, toolCardList: List<ToolCard>): void
- -getCommandFromString(getCommandFromString: OperationString, toolCard: ToolCard): CommandTypeEnum
- +validCommand(inputReceived: String): void
- -setTrackBoardIndex(inputReceived: String): void
- -setDiceSchemaWhereToTake(inputReceived: String): void
- -setDiceBoardIndex(inputReceived: String): void
- -separateCellValues(inputReceived: String): int
- -setDiceSchemaWhereToLeave(message: String): void
- -nextMessage(commandTypeEnum: CommandTypeEnum): int
- -setNextCommand(playerMove: PlayerMove): void
- -resetCommand(): void

CommandLineGraphic
- -hashMapColours: HashMap
- +CommandLineGraphic()
- +showBasicCard(card: Card): void
- -showPublicObjCard(publicObjCard: PublicObjCard): void
- -showPrivateObjCard(card: PrivateObjCard): void
- -showToolCard(card: Card): void
- +showSchemaCard(schemaCard: SchemaCard): void
- +printCardColour(colourCard: ColourEnum): void
- -separateString(string: String): List<String>

CommandLineInput
- -syntaxController: syntaxController
- -playerSetupper: PlayerSetupper
- +CommandLineInput()
- +yourTurn(clientBoard: ClientBoard, playerMove: PlayerMove): void
- -makeChoice(playerChoice: PlayerChoice): void
- -choseNickname(user: User): void
- +tryReconnection()

GuiOutput
- +setControllerMatchTable()
- +showGameBoard()
- +showScore()

GuiInput
- +GuiInput()
- +makeChoice()
- +showError()
- +yourTurn()

MessageLoader
- -errors: Map
- +MessageLoader()

ControllerMatchTable
- +initialize(location: URL, resources ResourceBundle): void
- +enablePane(): void
- +disablePane(): void
- +createNetwork(): void
- +request(): void
- +send(): void
- +createScheme(schemaCard: SchemCard): GridPane

RemoteView
- -userList: List<User>
- -serverManager: ServerManager
- +RemoteView(serverManager: ServerManager)
- +sendChoice(playerChoice: PlayerChoice): void
- +sendMove(playerMove: PlayerMessage): void
- +sendWinner(gameBoard: GameBoard): void
- +sendTable(moveMessage: MoveMessage): void
- +reportError(idError: int, nickname: String): void
- +isYourTurn(player: Player, playerMove: PlayerMove): void
- +setRecipient(nickname: String, playerMessage: PlayerMessage, userList: List<User>): void

ValueRequest
- +display(syntaxController: SyntaxController): void