```
+ GuiGame
  implements UlInterface
               ViewVisitor
              ViewModelVisitor
              ViewControllerVisitor
∃fields ----
 instance:GuiGame
 waitGame:WaitGame
 gameStage:Stage
 utilStage:Stage
 toolStage:Stage
  sceneGame:Scene
 scenelnit:Scene
  diceSource:String
  toolCardSource:String
  privateObjectSource:String
  publicObjectSource:String
 cardShow:ShowCardBox
 value:ShowValue
 popUpGame: AlertMessage
 popUpWait:AlertMessage
 init:boolean
 playerld:int
  opposingPlayers:VBox
 centerBox:VBox
 boxAllDataPlayer:HBox[]
  infoPlayer: VBox[]
 playersName:Text[]
 numberData:HBox[]
 favorTokenOfEachPlayer:Text[]
  pointsOfEachPlayer:Text[]
 vBoxesHandOfEachPlayer:VBox[]
 imageViewsHandPlayer : ImageView[][]
 boxWindowPlayer:VBox[]
  nameWindowPlayer : Text[]
  gridCellPlayer:GridPane[]
 imageViewsCellPlayer :ImageView[][][]
 difficultyWindowPlayer : Text[]
 boxAllWindowPoolChoice:HBox
 boxWindowPoolChoice:VBox[]
 nameWindowPoolChoice:Text[]
 gridCellPoolChoice:GridPane[]
  imageViewsCellPoolChoice :ImageView[][][]
  difficultyWindowPoolChoice:Text[]
 cardBox:VBox
  toolBox:HBox
 toolCardInfoBox:VBox[]
 costToolCard:Text[]
 toolCard:ImageView[]
  objectivePublicBox:HBox
  objectivePublicCard:ImageView[]
  objectivePrivateBox : HBox
  objectivePrivateCardOfEachPlayers :ImageView[]
  boxAllRound:HBox
 boxSingleRound:VBox[]
 textSingleRound:Text[]
 currentRound:Text
 currentTurn:Text
 comboBoxSingleRound:ComboBox[]
 flowPaneDicePool :FlowPane
  dicePool:LinkedList<|mageView>
 gameButton:Button[]
 constructors —
 GuiGame()
 methods —
 errPrintln (error: String):void
 getGuiGame():GuiGame
createGuiGame():GuiGame
 setGameWait (owner: Stage):void
 closeGame(stage:Stage):void
  setInit ():void
 setBoard ():void
 createNewImageViewForCard ():ImageView
 createNewImageViewForDicePool ():ImageView
 showEventView (eventClient: EventClient):void
 sendEventToNetwork(eventController: EventController):void
 loginOk():void
 restartConnection (cause: String):void
 visit(event: EventClientFromController):void
 visit(event: EventClientFromModel):void
 visit (event: UpdateDisconnectionDuringSetup):void
 visit (event: UpdateNamePlayersDuringSetUp):void
 visit(event: UpdateDisconnectionDuringGame):void
 visit (event: UpdatePlayerConnection ):void
 visit (event: EndGame): void
 visit(event: MoveTimeoutExpired):void
visit (event: LoginResponse):void
 visit (event: ConnectionDown): void
 visit(event: AskLogin):void
 visit(event: StartGame):void
 visit (event: StartPlayerTurn): void
 visit(event: WaitYourTurn):void
  visit (event: SelectCellOfWindow ):void
 activeWindow (indexWindow:int):void
  activeCell (indexWindow:int, indexRow:int, indexColumn:int):void
 disableWindow(indexWindow:int):void
 visit(event: SelectDiceFromRoundTrack):void
 activeRoundTrack():void
 activeSingleRound(indexRound:int):void
  disableAllRound():void
 visit (event: SelectValueDice ):void
 visit(event: SelectIncrementOrDecreaseDice ):void
  visit(event: SelectDiceFromDraftPool):void
 activeDiceOfDicePool (index:int):void
 disableDiceOfDicePool ():void
 visit(event: SelectToolCard):void
```

+ Ulinterface

+ loginOk():void

+ CliController

showEventView (eventClient: EventClient):void

restartConnection (message: String):void

ViewVisitor

ViewControllerVisitor

ViewModelVisitor

objectivePublicCards:ObjectivePublicCard[]

windowPatternCardsToChoose :WindowPatternCard[]

windowPatternCardOfEachPlayer :WindowPatternCard[]

objectivePrivateCardOfEachPlayers :ObjectivePrivateCard[]

errPrintln (message: String): void

implements UlInterface

instance:CliController

factory: ClientFactory

currentRound:int

currentTurn:int

cliMessage: CliMessage

roundTrack: DiceStack[]

toolCard:ToolCard[]

playersName:String[]

connected:boolean[]

playerId:int

ranking:int[][]

MUTEX: Object

constructors ----

- CliController()

methods ----

start (): void

login():void

turn():void

loginOk():void

currentTask:Thread

handOfEachPlayer:DiceStack[]

favorTokenOfEachPlayer:int[]

isInputActive:AtomicBoolean

getFactory():ClientFactory

checkShutDown():boolean

main(args:String[]):void

initConnection():void

getMyHand():DiceStack

getMyName (): String

errPrintln (error: String):void

CliController(client2:AbstractClient2)

sendEventToNetwork(packet: EventController):void

sendEventToNetwork(packet: EventServer):void

getMyWindowPatternCard ():WindowPatternCard

showEventView (eventClient: EventClient):void

visit(event: EventClientFromController):void

visit(event: EventClientFromModel):void

visit (event: LoginResponse): void

visit(event: ConnectionDown):void

visit (event: AskLogin): void

visit(event: StartGame):void

visit(event: ShowAllCards):void

visit(event: InitialEnded):void

visit (event: EndGame): void

sendInfo(info:int):void

visit (event: WaitYourTurn): void visit (event: StartPlayerTurn): void

visit(event: MoveTimeoutExpired):void

visit(event: SelectDiceFromDraftPool):void

visit(event: SelectDiceFromRoundTrack):void

visit (event: SelectIncrementOrDecreaseDice):void

visit (event: UpdateInitialWindowPatternCard):void

visit (event: SelectCellOfWindow):void

visit(event: SelectValueDice):void

sendInfo(info1:int, info2:int):void

visit(event: UpdateInfoCurrentTurn):void

visit(event: UpdateAllPublicObject):void

visit (event: UpdateNamePlayers): void

visit(event: UpdateInitDimRound):void

visit(event: UpdateDicePool):void

visit(event: UpdateSingleCell):void

visit (event: UpdateSingleWindow):void visit (event: UpdateStatPodium):void

visit (event: UpdatePlayerConnection):void

visit(event: UpdateCurrentPoint):void

visit(event: UpdateSingleToolCardCost):void

visit (event: UpdateSinglePlayerHand):void

visit (event: UpdateSinglePlayerToken):void visit (event: UpdateSinglePrivateObject):void

visit (event: UpdateSingleTurnRoundTrack):void

visit(event: UpdateDisconnectionDuringSetup):void

visit (event: UpdateNamePlayersDuringSetUp):void visit (event: UpdateDisconnectionDuringGame):void

visit (event: UpdateAllToolCard):void

visit(event: SelectToolCard):void

visit(event: MessageError):void

visit (event: MessageOk):void

visit(event: SelectInitialWindowPatternCard):void

restartConnection (message: String):void

pointsOfEachPlayer:int[]

dicePool:DiceStack

cliParser:CliParserNonBlocking

cliParserBlocking:CliParser client2:AbstractClient2

sendEventToNetwork(eventController: EventController):void

methods -

```
+ ControllerGUI
  implements UlInterface
               ViewVisitor
              ViewControllerVisitor
 fields ----
  instance:ControllerGUI
  factoryInstance:ClientFactory
 client2:AbstractClient2
  login:Login
  waitGame:WaitGame
 players:LinkedList<PlayerOnline>
 -constructors
 ControllerGUI()
methods —
 getFactoryInstance():ClientFactory
 setFactoryInstance (factoryInstance: ClientFactory):void
 getClient2 (): AbstractClient2
 setClient2 (client2: AbstractClient2):void
 main(args:String[]):void
  getGuilnstance():ControllerGUI
 getSingletonGUIInstance():ControllerGUI
 startGui():void
 getWaitGame ():WaitGame
  disconnect():void
 showEventView (eventClient: EventClient):void
 sendEventToNetwork(eventController: EventController):void
 restartConnection (message: String):void
errPrintln (message: String):void
 loginOk():void
 visit(eventView: EventClientFromController):void
 visit(eventView: EventClientFromModel):void
 visit(event: LoginResponse):void
 visit(event: ConnectionDown):void
 visit(event: AskLogin):void
 visit(event: StartGame):void
 visit(event: ShowAllCards):void
 visit(event: SelectInitialWindowPatternCard ):void
visit(event: InitialEnded):void
 visit(event: MessageError):void
 visit(event: MessageOk ):void
 visit(event: StartPlayerTurn):void
visit(event: SelectCellOfWindow):void
 visit(event: SelectDiceFromDraftPool):void
  visit(event: SelectToolCard):void
 visit(event: MoveTimeoutExpired):void
 visit(event: WaitYourTurn):void
+ visit(event: SelectDiceFromRoundTrack):void
+ visit(event: SelectValueDice):void
 visit(event: SelectIncrementOrDecreaseDice ):void
visit(event: EndGame):void
```

+ CliParser cliMessage : CliMessage ☐constructors CliParser () ∃methods readSplash():int parseInt():int parsePositiveInt (upperBound:int):int parseInt(upperBound:int):int parseNickname():String parselp():String + parsePort (lowBound:int):int + CliParserNonBlocking extends CliParser isInputActive:AtomicBoolean cliMessage : CliMessage CliParserNonBlocking (isInputActive: AtomicBoolean) methods readSplash():int parseInt():int parseInt (upperBound:int):int parsePositiveInt (upperBound:int):int flush(bufferedReader: BufferedReader):void

visit(event: MessageError):void visit(event: MessageOk):void visit(event: ShowAllCards):void

disableWindowChoice():void visit(event:InitialEnded):void

activeWindowChoice (index:int):void

+ visit(event: UpdateNamePlayers):void + visit(event: UpdateSinglePrivateObject):void + visit(event: UpdateInitialWindowPatternCard):void

visit (event: UpdateAllToolCard):void visit (event: UpdateAllPublicObject):void visit (event: UpdateInitDimRound):void visit (event: UpdateSingleWindow):void visit (event: UpdateSingleToolCardCost):void

visit(event: UpdateDicePool):void visit(event: UpdateInfoCurrentTurn):void visit(event: UpdateSinglePlayerHand):void visit(event: UpdateSingleCell):void

visit(event: UpdateCurrentPoint):void visit(event: UpdateStatPodium):void

visit(event: UpdateSinglePlayerToken):void visit(event: UpdateSingleTurnRoundTrack):void

visit(event: SelectInitialWindowPatternCard):void

~ ShowCardBox

☐ fields
— clicked:boolean
— clicklsOn:boolean
— stage:Stage
— maxWidth:int
— maxHeigh:int
☐ constructors
— ShowCardBox(owner:Stage, maxWidth:int, maxHeigh:int)
☐ methods
— setClickIsOn (clickIsOn: boolean):void
~ displayCard(imageViewToShow: ImageView, canCanUseCard: boolean):boolean

+ AlertMessage

fields
- stageMessage:Stage

constructors
+ AlertMessage (owner:Stage)
methods
- displayMessage (message:String):void

+ GuiMain extends Application

fields

primaryStage:Stage

game:GuiGame

constructors

methods

launchGui():void

setPrimaryStage (stage:Stage):void

start (primaryStage:Stage):void

closeProgram():void

+ SetUpConnection

fields
- stage:Stage
constructors
+ SetUpConnection(owner:Stage)
methods
+ display():void

isInt(input:TextField):boolean

~ CliMessage constructors— CliMessage () methods println():void eraseScreen ():void synchronized splashScreen():void synchronized showSocketRmi():void - synchronized showlpRequest():void - synchronized showPortRequest():void synchronized showConnectionSuccessful():void synchronized showConnectionFailed():void - synchronized showInsertNickname():void ~ synchronized showWelcomeNickname(nickname:String):void - synchronized showNicknameExists():void synchronized showGameStarted(playersName:String[]):void synchronized showInitialWindowPatternCardSelection ():void - synchronized showYourTurnScreen():void synchronized showWaitYourTurnScreen(name:String):void synchronized showWindowPatternCard (card:WindowPatternCard):void synchronized showObjectivePublicCardMessage ():void synchronized showObjectivePublicCard (card:ObjectivePublicCard):void synchronized showObjectivePrivateCardMessage ():void ~ synchronized showObjectivePrivateCard (card:ObjectivePrivateCard):void synchronized showOpponentWindowMessage ():void synchronized showOpponentInsertDice(name:String, line:int, column:int):void synchronized showOpponentWindow(name:String):void synchronized showToolCardMessage():void - synchronized showToolCard(card:ToolCard):void synchronized showDice(dice:Dice):void synchronized showRoundTrack(roundTrack:DiceStack[]):void synchronized showMainMenu():void ~ synchronized showChoiceRow():void ~ synchronized showChoiceColumn():void ~ synchronized showChoiceRound():void - synchronized showChoiceInRound():void synchronized showValueDice():void ~ synchronized showIncrementDecrement():void synchronized showDiceStack(diceStack: DiceStack):void synchronized showDicePool(diceStack: DiceStack):void synchronized showHandPlayer(hand:DiceStack):void synchronized showToolCardChoice(toolCard:ToolCard[]):void ~ synchronized showInputNotValid():void ~ synchronized showWaitInput():void synchronized showRoundAndTurn(round:int, turn:int):void synchronized showMessage (message: String):void - synchronized showGreenMessage (message: String):void synchronized showMoveTimeoutExpired():void ~ synchronized showReLogin():void

~ synchronized showEndGameScreen(ranking:int[][], playersName:String[], myld:int):void

+ ConfirmBox

- fields -- answer:boolean
- stage:Stage
--constructors -+ ConfirmBox(owner:Stage)
--methods -+ displayMessage (message:String):boolean

+ WaitGame ∃fields stage:Stage tableView : TableView < PlayerOnline > nameInput:TextField listPlayer:ObservableList<PlayerOnline> → constructors — WaitGame (owner: Stage) -}methods − getStage ():Stage displayMessage (messageWait: String):void addPlayerOnline (index:int, nickname:String, connected:boolean):void addPlayerOnline (names: String[]):void deletePlayerKicked (nickname:String):void deletePlayerKicked ():void getPlayerOnlineSingleton (): ObservableList < PlayerOnline > closeWait ():void