com.sagrada.ppp

com.sagrada.ppp.model

Cell

- color : Color
- diceOn : Dice
- value : Integer

+ equals(Object) : boolean
+ hasColorRestriction( ) : boolean
+ hasDiceOn( ) : boolean
+ hasValueRestriction( ) : boolean
+ hashCode( ) : int + toString( ) : String

Dice

- color : Color
- value : int

+ equals(Object) : boolean
+ hashCode( ) : int
~ isSimilar(Dice) : boolean
+ throwDice( ) : void
+ toString( ) : String

DicePlacedMessage

+ panel : WindowPanel
+ username : String
+ draftPool : ArrayList<Dice>

IstoolCardUsableResult

+ result : boolean
+ toolCardID : int

JoinGameResult

+ gameHashCode : int
+ playerHashCode : int
+ playersUsername : ArrayList<String>
+ timerStart : long
+ username : String

PlaceDiceResult

+ message : String
+ panel : WindowPanel
+ status : boolean
+ draftPool : ArrayList<Dice>

GameStartMessage

+ publicObjectiveCards : ArrayList<PublicObjectiveCard>
+ toolCards : ArrayList<ToolCard>
+ players : ArrayList<Player>
+ currentPlayer : Player
+ draftpool : ArrayList<Dice>

ReconnectionResult

+ gameStartMessage : GameStartMessage
+ message : String
+ result : boolean

EndTurnMessage

+ roundTrack : RoundTrack
+ turn : int
+ players : ArrayList<Player>
+ currentPlayer : Player
+ previousPlayer : Player
+ draftpool : ArrayList<Dice>

UseToolCardResult

+ dice : Dice
+ players : ArrayList<Player>
+ msg : String
+ result : boolean
+ roundTrack : RoundTrack
+ toolCardCost : int
+ toolCardId : int
+ draftpool : ArrayList<Dice>

LeaveGameResult

+ gameHashCode : int
+ status : LeaveGameResultStatus

DiceBag

- bag : ArrayList<Dice>

+ addDice(Dice) : void
+ equals(Object) : boolean
~ extractDices(int) : ArrayList<Dice>
+ extractRandomDice( ) : Dice
~ getDiceBag( ) : ArrayList<Dice>
+ hashCode( ) : int
- numberOf(int, Color) : int
+ size( ) : int

<<Enumerazione>> Color

RED,
GREEN,
YELLOW,
BLUE,
PURPLE

+ getColor(String) : Color
+ getRandomColor( ) : Color
+ valueOf(String) : Color
+ values( ) : Color[]

<<Enumerazione>> TimerStatus

START,
INTERRUPT,
FINISH

ToolCardNotificationMessage

+ draftPool : ArrayList<Dice>
+ toolCardCost : int
+ toolCardID : int
+ player : Player
+ roundTrack : RoundTrack

<<Interfaccia>> GameObserver

+ onDicePlaced(DicePlacedMessage) : void
+ onEndGame(ArrayList<PlayerScore>) : void
+ onEndTurn(EndTurnMessage) : void
+ onGameStart(GameStartMessage) : void
+ onPanelChoice(int, ArrayList<WindowPanel>, HashMap<String, WindowPanel>, Color) : void
+ onPlayerAFK(Player, boolean, Player) : void
+ onPlayerDisconnection(Player, boolean) : void
+ onPlayerReconnection(Player) : void
+ onToolCardUsed(ToolCardNotificationMessage) : void
+ rmiPing( ) : void

<<Enumerazione>> LeaveGameResultStatus

SUCCESS,
FAIL,
GAME_DELETED

RoundTrack

- rounds : int
- currentRound : int
- dicesOnTrack : ArrayList<ArrayList<Dice>>

+ getDice(int, int) : Dice
+ getDicesOnRound(int) : ArrayList<Dice>
~ nextRound( ) : void
+ setDice(int, int, Dice) : void
+ setDicesOnRound(int, ArrayList<Dice>) : void
+ toString( ) : String

Game

«transient»

- diceBag : DiceBag
- chosenPanelIndex : Integer «transient»
- currentTimerTask : MyTimerTask
«volatile»
- dicePlaced : boolean
- draftPool : ArrayList<Dice>
«volatile»
- endTurn : boolean
«volatile»
- gameEnded : boolean
- gameStatus : GameStatus
«volatile» - isSpecialTurn : boolean
- lobbyObservers : HashMap<Integer, LobbyObserver>
«transient»
- lobbyTimer : LobbyTimer
- gameObservers : HashMap<Integer, ArrayList<GameObserver>>
«volatile»
~ panelChoiceTimerExpired : boolean
- publicObjectiveCards : ArrayList<PublicObjectiveCard>
- roundTrack : RoundTrack
«transient» - service : Service
- toolCards : ArrayList<ToolCard>
- turn : int
«volatile»
- turnTimeout : boolean
«volatile»
- usedToolCard : boolean
«volatile»
- waitingForPanelChoice : boolean
- joinable : boolean
- lobbyTimerStartTime : long
- usernames : ArrayList<String>
- players : ArrayList<Player>

- assignPrivateObjectiveColors( ) : void
+ attachGameObserver(GameObserver, int) : void
+ attachLobbyObserver(LobbyObserver, int) : void
+ detachAllGameObservers(int) : void
+ detachLobbyObserver(int) : void + disableAFK(int) : void
+ disconnect(int) : boolean
- extractPanels( ) : HashMap<Integer, ArrayList<WindowPanel>>
- extractPublicObjCards( ) : void
- extractToolCards( ) : void
- gameHandler( ) : void
~ getActivePlayersNumber( ) : int
- getCurrentPlayerIndex( ) : int
- getDraftPool( ) : ArrayList<Dice>
+ getLegalPositions(int, Dice) : ArrayList<Integer>
- getPlayerByHashcode(int) : Player
+ getPlayerHashCode(String) : int - getPlayerScore(Player) : PlayerScore
+ getPlayerUsername(int) : String
+ getToolCardID(int) : int
- getToolCardParameters(ToolCardParameters, int) : ToolCardParameterContainer
~ init( ) : void
- isInMatch(String) : boolean
«synchronized»
+ isToolCardUsable(int, int) : boolean
+ joinGame(String, LobbyObserver) : int
+ leaveLobby(String) : void
- notifyDisconnection(Player, boolean) : void
- notifyEndGame(ArrayList<PlayerScore>) : void
- notifyEndTurn(Player, Player) : void
- notifyGameStart( ) : void «synchronized»
- notifyPanelChoice(int, ArrayList<WindowPanel>, HashMap<String, WindowPanel>, Color) : void
- notifyPlayerAFK(Player, boolean, Player) : void
- notifyPlayerJoin(String, ArrayList<String>, int) : void
- notifyPlayerLeave(String, ArrayList<String>, int) : void
- notifyReconnection(Player) : void
~ notifyTimerChanges(TimerStatus) : void
- notifyUsedToolCard(int, Player, ArrayList<Dice>, RoundTrack, int) : void
+ pairPanelToPlayer(int) : void
~ pingAllGameObservers( ) : void
~ pingAllLobbyObservers( ) : boolean
«synchronized»
+ placeDice(int, int, int, int) : PlaceDiceResult
+ putDiceInDraftPool(Dice) : void
+ reconnection(int, GameObserver) : ReconnectionResult
~ reorderPlayers( ) : void
~ setDiceBag(DiceBag) : void
~ setDicePlaced(boolean) : void
~ setDraftPool(ArrayList<Dice>) : void
+ setEndTurn(int) : void
- setPlayerAFK(Player) : void
~ setRound(int) : void
~ setTurn(int) : void
~ setUsedToolCard(boolean) : void
+ specialDicePlacement(int, int, Dice) : PlaceDiceResult
- toNextRound( ) : void
- toNextTurn( ) : void «synchronized»
+ useToolCard(int, ToolCardParameters) : UseToolCardResult

<<InnerClass>>
MyTimerTask

«volatile»
~ isValid : boolean

+run(): void

LobbyTimer

~ game : Game
- startTime : long

+ run( ) : void

PaneChoiceTimer

~ game : Game
- startTime : long

+ run( ) : void

WindowPanel

- loadedPanels : ArrayList<WindowPanel>
- cardID : int
- emptyCells : int
- favorTokens : int
- panelName : String
- cells : ArrayList<Cell>

+ addDice(int, Dice) : boolean
+ addDice(int, Dice, boolean, boolean, boolean) : boolean
- borderPosition(int) : boolean
- cellPairSimilarity(int, int, Dice) : boolean
+ diceOk(Dice, int) : boolean
+ diceOk(Dice, int, boolean, boolean, boolean) : boolean
- diceOkWithRestriction(Cell, Dice, boolean, boolean) : boolean
+ equals(Object) : boolean
+ getCell(int) : Cell
+ getCell(int, int) : Cell
~ getLegalPosition(Dice) : ArrayList<Integer>
~ getNumberOfPanels( ) : int
~ getPanel(int, int) : WindowPanel
~ getPrivateScore(Color) : int
- hasSimilarDiceAttached(Dice, int) : boolean
+ hashCode( ) : int «synthetic»
~ loadPanels( ) : void
+ noDiceNear(int) : boolean
+ removeDice(int) : Dice
+ toString( ) : String
~ unloadPanels( ) : void
- validPosition(int, int) : boolean
- windowIsEmpty( ) : boolean

<<Enumerazione>> GameStatus

INIT,
PANEL_CHOICE,
ACTIVE,
SCORE,
CLOSE

Player

- skipSecondTurn : boolean
- favorTokens : int
- hashCode : int
- privateColor : Color
- username : String
- panel : WindowPanel
- status : PlayerStatus

~ getPlayerStatus( ) : PlayerStatus
~ hasToSkipSecondTurn( ) : boolean
~ setPlayerStatus(PlayerStatus) : void
~ setSkipSecondTurn(boolean) : void

ToolCardFlags

«volatile»
+ colorDiceValueRequired : Color
«volatile»
+ isActionSignRequired : boolean
«volatile»
+ isDiceValueRequired : boolean
«volatile»
+ isDraftPoolDiceRequired : boolean
«volatile»
+ isPanelCellRequired : boolean
«volatile»
+ isPanelDiceRequired : boolean
«volatile»
+ isReRolledDiceActionRequired : boolean
«volatile»
+ isRoundTrackDiceRequired : boolean
«volatile»
+ isSecondPanelCellRequired : boolean
«volatile»
+ isSecondPanelDiceRequired : boolean
«volatile»
+ isTwoDiceActionRequired : boolean
«volatile»
+ reRolledDice : Dice

+ reset( ) : void

<<Interfaccia>> LobbyObserver

+ onPlayerJoined(String, ArrayList<String>, int) : void
+ onPlayerLeave(String, ArrayList<String>, int) : void
+ onTimerChanges(long, TimerStatus) : void
+ rmiPing( ) : void

<<Enumerazione>> PlayerStatus

ACTIVE,
INACTIVE

PlayerScore

- emptyCellsPoints : int
- favorTokenPoints : int
- playerHashCode : int
- privateColor : Color
- privateObjectiveCardPoints : int
- publicObjectiveCard1Points : int
- publicObjectiveCard2Points : int
- publicObjectiveCard3Points : int
- totalPoints : int
- username : String
- windowPanel : WindowPanel

~ calculateTotalPoints( ) : void

ToolCardParameters

«volatile»
+ actionSign : Integer
«volatile»
+ diceValue : Integer
«volatile»
+ draftPoolDiceIndex : Integer
«volatile»
+ panelCellIndex : Integer
«volatile»
+ panelDiceIndex : Integer
«volatile»
+ roundTrackDiceIndex : Integer
«volatile»
+ roundTrackRoundIndex : Integer
«volatile»
+ secondPanelCellIndex : Integer
«volatile»
+ secondPanelDiceIndex : Integer
«volatile»
+ toolCardID : Integer
«volatile»
+ twoDiceAction : Boolean

+ reset( ) : void