Mod on 08/05

ClientSide

AppClient
+login : boolean
+connnect : boolean

View
-nomeMembro
-nomeMembro

View rappresentata ad alto livello, più dettagli con la scrittura del codice

GuiView
-nomeMembro
-nomeMembro

CliView
-nomeMembro
-nomeMembro

SocketClient
-nomeMembro
-nomeMembro

ScoreView
-nomeMembro
-nomeMembro

GameView
-nomeMembro
-nomeMembro

LobbyView
-nomeMembro
-nomeMembro

ScoreTrackComponent
-nomeMembro
-nomeMembro

ActiveGameContainer
-nomeMembro
-nomeMembro

StatusGameContainer
-nomeMembro
-nomeMembro

CardComponent
-nomeMembro
-nomeMembro

RoundTrackComponent
-nomeMembro
-nomeMembro

WindowPanelComponent
-nomeMembro
-nomeMembro

CellComponent
-nomeMembro
-nomeMembro

DiceComponent
-nomeMembro
-nomeMembro

ConnectionHandler
-status : int
+getCurrentConnection : ConnectionMode
+setCurrentConnection

<<Interfaccia>> ConnectionMode

SocketConnectionMode
//interfaccia di rete TCP lato client, serializzazione e deserializzazione delle interazioni client/server

RmiConnectionMode
//interfaccia RMI lato client

Server Side

SocketThread
-nomeMembro
-nomeMembro

ServerSocket
//serializzazione e deserializzazione messaggi TCP

<<Interfaccia>> ControllerInterface
//tutti i metodi che saranno implementati nel controller per la gestione del gioco e delle interazioni server / client

ControllerImplementation

ApplicationServer
+init() : void

Server Entry Point

Service
-gameMode : Game
+createGame() : Game

Score Markers don't compare in the diagram as a class, because they don't have any behaviour

Game
-numberOfPlayers: int
-activePlayer: Player
+init(): void
+joinGame(String username): void
+isInMatch(String username): boolean
+getPrivateScore(Player player): int
-initWindowPanels(): ArrayList<WindowPanels>

myPanel: WindowPanel

Player
-username: String
-myPanel: WindowPanel
-privateColor: Color
+getUsername(): String
+setUsername(String username): void
+getPrivateColor(): Color
+getMyPanel(): WindowPanel

<<Enumeration>> Color
GREEN
YELLOW
BLUE
PURPLE
RED
+getRandomColor(): Color
+getColor(String colorName): Color

WindowPanel
-panelName: String
-favorTokens: int
-cardID: int
+getCellWithPosition(int row, int col): Cell
+getCellWithIndex(int i): Cell
+getPanelName(): String
+setPanelName(String panelName): void
+getFavorTokens(): int
+setFavorTokens(int favorTokens): void
+getCardID(): int
+setCardID(int cardID): void
+getCells(): ArrayList<Cell>
+addDiceOnCellWithIndex(int i, Dice dice): boolean
+diceOnCell cell, Dice dice, int i): boolean
+atLeastOneNeiarint (i): boolean
+validPosition(int row, int col): boolean
+borderPosition(int i): boolean
+hasSimilarDiceAttached(Dice dice, int i): boolean
+cellPairSimilarity(int row, int col, Dice dice): boolean
+addDiceOnCellWithPosition(int row, int col, Dice dice): boolean
+windowIsEmpty(): boolean

cells: ArrayList<Cell>

Cell
-color: Color
-value: Integer
+hasValueRestriction(): boolean
+hasColorRestriction(): boolean
+getColor(): Color
+setColor(Color color): void
+getValue(): Integer
+setValue(Integer value): void
+getDiceOn(): Dice
+setDiceOn(Dice diceOn): void
+hasDiceon(): boolean
+equals(Cell cell): boolean

diceOn: Dice

DiceBag
+size(): int
+getDice(int i): Dice
+getDiceBag(): ArrayList<Dice>
+numberOfValue(final int n): int
+numberOfColor(Color color): int
+numberOf(int n, Color color): int

diceBag: ArrayList<Dice>

Dice
-color: Color
-value: int
+getColor(): Color
+getValue(): int
+setValue(int value): void
+equals(Dice dice): boolean
+isSmaler(Dice dice): boolean

<<interface>> ToolCard
+useCard : void

12 tool cards

ToolCard"X"

<<interface>> CommandToolCard
+use : void

12 different actions

CommandToolCard"X"
-nomeMembro
-nomeMembro

SinglePlayerGame
+startGame(String): void

RoundTrack
-rounds: int
-currentRound: int
+getDicesOnRound(int round): ArrayList<Dice>
+setDicesOnTurn(int round , ArrayList<Dice> dices): void
+setDice(int round, int index, Dice dice): void
+addDice(int round, Dice dice): void
+getDice(int round, int index): Dice
+removeDice(int round,int index): boolean
+dicesOnRound(int turn): int
+hasDiceOnRound(int turn): boolean
+getRounds(): int
+getCurrentRound(): int
+setCurrentRound(int currentRound): void
+nextRound(): void

dicesOnTrack: ArrayList<ArrayList<Dice>>

<<Abstract>> PublicObjectiveCard
-name: String
-ID: int
+getName(): String
+getID(): int
+getScoreWindowPanel playerWindowPanel): int

PublicObjectiveCardID
+getScoreWindowPanel playerWindowPanel): int