PlayerController

- playerReal : Player
- playerBot1 : Player
- playerBot2 : Player
- playerBot3 : Player
- players : ArrayList<Player>
- deck : Deck
- cartaPescata : Carta
- playerView : PlayerView
- mainController : JTrash
- primoTurno : boolean
- audioManager : AudioManager
- instance : PlayerController

+ getInstance() : PlayerController
- PlayerController() : PlayerController
+ setImageCartaDaGiocare(Carta)
+ handleCartaDaGiocare(MouseEvent)
+ validateMove(Carta, int) : boolean
+ updateUIAfterCardDraw()
+ updateUIAfterCardExchange(Carta, Player)
+ terminaTurno()
+ checkHoVinto()
+ decidiAlertVittoria(Player)
+ svuotaEPopolaMani()
+ haiVinto()
+ showAlertVittoriaRound(Player)
+ showAlertNonVittoria(Player)
+ showDialogVittoriaPartita(Player)
+ caricaProfilo(ActionEvent)
+ tornaAMenu(ActionEvent)
+ caricaMenu(ActionEvent)
+ getPlayers() : ArrayList<Player>
+ getPlayer(int) : Player
+ getGridFromPlayer(Player) : GridPane
// getters and setters

<<Observer, Initializable>> PlayerView

- playerController : PlayerController
- gridPlayerReal : GridPane
- gridPlayerBot1 :GridPane
- gridPlayerBot2 :GridPane
- gridPlayerBot3 :GridPane
- gridDeck :GridPane
- gridCartaPescata :GridPane
- bottoneTerminaTurno: Button
- bottoneHoVinto: Button
- bottoneTornaAMenu: Button
- gridPanesPlayers: List<GridPane>
- numberOfPlayers: Int
- audioManager ; AudioManager

+ PlayerView(int) : PlayerView
+ update(Observable, Object)
+ initialize(URL, ResourceBundle)
+ setMatch()
+ muoviCarta(int, int, Carta)
+ impostaCarteCoperte()
+ terminaTurno()
+ hoVinto()
+ tornaAMenu()
+ sleepThread()
// getters and setters

JTrash

- players : ArrayList<Player>
- idxCurrentPlayer : int
- playerView : PlayerView
- profiloView : ProfiloView
- stagePlayer : Stage

+ start(Stage)
+ startGame(PlayerView, Stage)
+ playNextTurn(PlayerController, Deck)
+ nextBotMove(PlayerController, Player, Carta)
+ pescaEAggiornaCarta(PlayerController) : Carta
+ pescaCarta(PlayerController) : Carta
+ setPlayerView(int)
+ getPlayerView() : PlayerView
+ setIdxPlayer(int)
+ getIdxPlayer() : int
+ sleepThread()
+ getProfiloView(): ProfiloView
+ setProfiloView(ProfiloView)
+ getStagePlayer(); Stage

<<Observer, Initializable>> ProfiloView

- profiloController : ProfiloController
- nicknameTextField : TextField
- nicknameLabel : Label
- buttonConfirmName: Button
- buttonChangeName: Button
- progressBarLv: ProgressBar
- progress: BigDecimal
- gridAvatars: GridPane
- gridAvatar: GridPane
- buttonChangeAvatar: Button
- buttonConfirmAvatar: Button
- buttonStartGame: Button
- button2Players : Button
- button3Players : Button
- button4Players : Button
- buttonLogout; Button
- labelHowManyPlayer: Label
- partiteGiocate; Label
- partiteVinte: Label
- percentualeBarra: Label
- LV : Label

+ update(Observable, Object)
+ initialize(URL, ResourceBundle)
+ setProgress()
+ confirmNickname()
+ changeNickname()
+ changeAvatar()
+ confirmAvatar()
+ startGame()
+ choosePlayers()
+ logout()
// getters and setters

ProfiloController

- profilo : Profilo
- profiloView : ProfiloView
- main : JTrash
- avatars : List<Avatar>
- avatarTemp : Avatar
- instance: ProfiloController

+ getInstance() : ProfiloController
- ProfiloController() : ProfiloController
+ startGame(ActionEvent, int)
+ logout(ActionEvent)
+ choosePlayer()
+ scriviDati()
+ leggiDati()
+ confirmNickname()
+ setNickNameFrontEnd(String)
+ changeNickname()
+ changeAvatar()
+ confirmAvatar()
+ handleClickAvatar(MouseEvent, Avatar)
+ determinaPercorsoAvatar(String) ; Avatar
+ setPartiteVinte(String)
+ setPartiteGiocate(String)
// getters and setters

<<Observable>> Player

- mano: Mano
- isBot; boolean
- isPlaying: boolean
- indice : int
- playerController: PlayerController

+ Player(Mano, boolean, boolean, int): Player
+ notifyMove(Carta)
+ removeLastCard()
+ getIndiceFromCarta(Carta) : int
+ getIndice(): int
+ getSizeMano() : int
+ isPlaying(): boolean
+ isBot() : boolean
// getters and setters

Mano

- mano : ArrayList<Carta>

+ Mano(Deck) : Mano
+ getCarte(): ArrayList<Carta>
+ getCarta(int): Carta
+ setCarta(int, Carta)

<<Comparable>> Carta

- imageCarta : Image
- ImageViewCarta : ImageView
- isMoved : boolean
- coperta : boolean
- seme : Seme
- valore : Valore

+ Carta(Seme, Valore) : Carta
+ determinaPescorsoImg(Seme, Valore)
+ compareTo(Carta) : int
+ isEqual(Carta) : boolean
+ toString() : String
// getters and setters

<<Enum>> Seme

CLUBS,
DIAMONDS,
HEARTS,
SPADES

<<Enum>> Valore

- value

- Valore(int)
- getValue()

Deck

- mazzo : ArrayList<Carta>
- secondoMazzo : ArrayList<Carta>
- mazzoScartato : ArrayList<Carta>
- haSecondoMazzo : boolean

+ Deck() : Deck
+ distribuisciCarta(ArrayList<Carta>) : Carta
+ giocaConDueMazzi()
+ creaMazzo(ArrayList<Carta>)
+ shuffle(ArrayList<Carta>)
+ aggiungiAScartato(Carta)
+ rimescolaMazzoFinito()
// getters and setters

Avatar

- image: Image
- ImageView : ImageView
- path; String

+ Avatar(Avatar)
+ Avatar()
// getters and setters

<<Observable>> Profilo

- nickname : String
- avatar : Avatar
- partiteGiocate: int
- partiteVinte : int
- lv : int
- profiloController: ProfiloController

+ setNickname(String)
+ setAvatar(Avatar)
+ setPartiteVinte(int)
+ setPartiteGiocate(int)
// getters and setters

Initializable

Observer

Application

Observable

Use
Extends