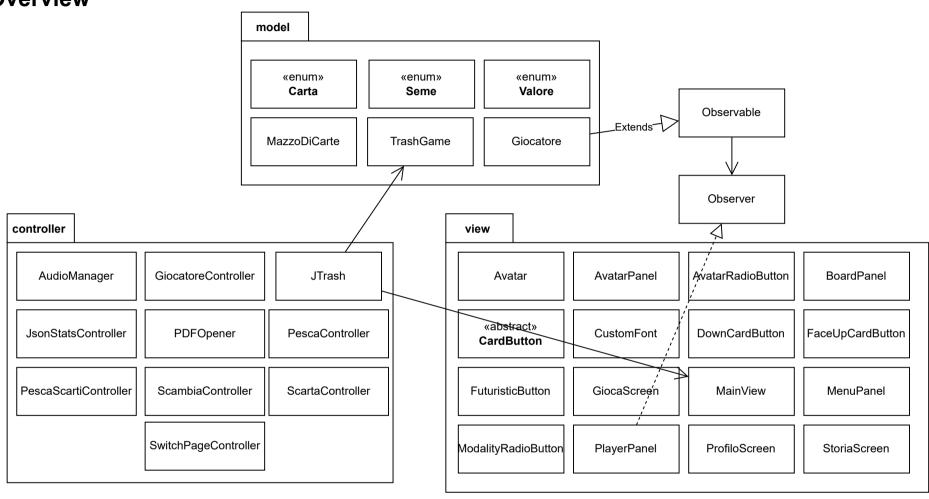
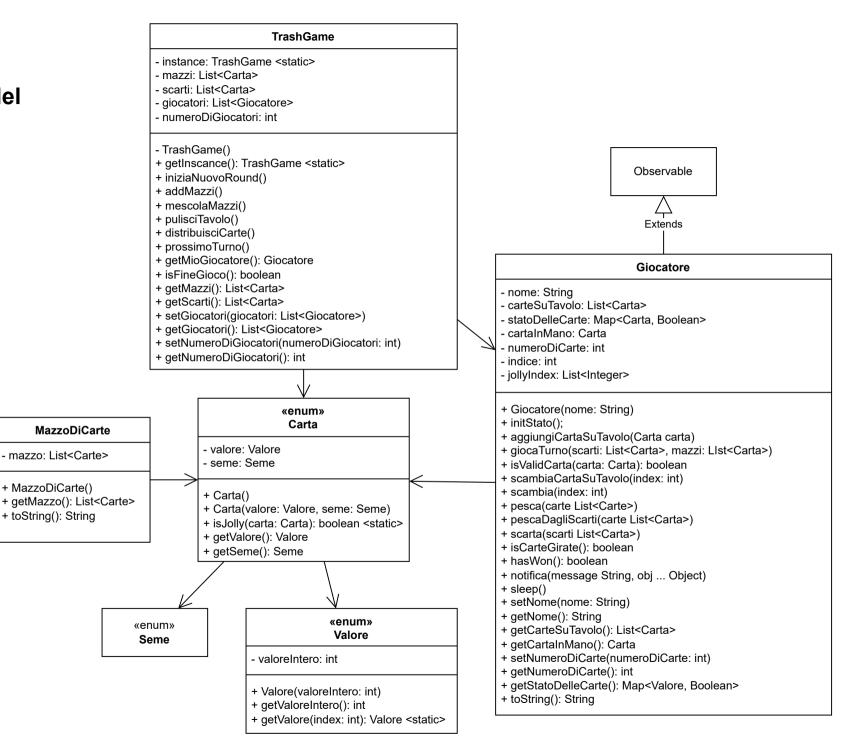
Overview



Model



AudioManager

- instance: AudioManager <static>

- bgClip: Clip - isMuted: boolean

- AudioManager()

+ getInstance(): AudioManager

+ play(filename: String)

+ setVolume(clip: Clip, volume: float)

+ isMuted(): boolean

+ setMuted(isMuted: boolean)

+ getBgClip(): Clip

JsonStatsController

- nikname: String

- avatar: int

partiteGiocate: intpartiteVinte: intpartitePerse: int

- livello: int

- nomeAvatar: String - filePath: String <final>

+ readJsonStats()

+ writeJsonStats()

+ getNickname(): String

+ setNickname(nickname: String)

+ getAvatar(): int

+ setAvatar(avatar: int)

+ getPartiteGiocare(): int

+ setPartiteGiocate(partiteGiocate: int)

+ getPartiteVinte(): int

+ setPartiteVinte(partiteVinte: int)

+ getPartitePerse(): int

+ setPartitePerse(partitePerse: int)

+ aetLivello(): int

+ setLivello(livello: int)

+ getNomeAvatar(): String

+ setNomeAvatar(nomeAvatar: String)

+ getFilePath(): String

+ updateStats()

Controller

GiocatoreController

- isPescaTime: boolean <static>
- isScambiaTime: boolean <static>

isScartaTime: boolean <static>isMioTurno: boolean <static>

- trashGame: TrashGame <static>

+ isScartaTime(): boolean <static>

+ setScartaTime(isScartaTime) <static>

+ isScambiaTime(): boolean <static>

+ setScambiaTime(isScambiaTime: boolean) <static>

+ isPescaTime(): boolean <static>

+ setPescaTime(isPescaTime: boolean) <static>

+ isMioTurno: boolean <static>

+ setMioTurno(isMioTurno) <static>

+ setGiocatori(numeroDIGiocatori) <static>

+ getGiocatori(numeroDiGiocatori: int): List<Giocatore> <static>

+ getNumeroDiCarte(giocatoreIndex: int): int <static>

+ addObserver(obs: Observer, index: int) <static>

JTrash

- mainView: MainView <static>

- trashGame: TrashGame <static>

- isGameRunning: boolean <static>

+ main(args: String[]) <static>

+ startGame() <static>

+ setGameRunning(inGameRunning: boolean) <static>

PescaController

+ actionPerformed(e: ActionEvent)

PescaScartiController

+ actionPerformed(e: ActionEvent)

ScambiaController

ActionListener

- index: int

+ ScambiaController(index: int)

+ actionPerformed(e: ActionEvent)

ScartaController

+ actionPerformed(e: ActionEvent)

SwitchPageController

- mainView: MainView <static>

+ showMenuScreeen() <static>

+ showGiocaScreeen() <static>

+ showProfiloScreeen() <static>

+ showStoriaScreeen() <static>

PDFOpener

+ openPDFFile(pdfFilePath: String) <static>

View

Avatar AvatarPanel **BoardPanel** - nome: String - screenWidth: int - avatar: Avatar - playerlmage: Bufferedlmage - imagePath: String - screenHeight: int - playerName: String - panelWidth: int - descrizione: String characterName: String - panelHeight: int - remainingCards: int - cardWidth: int + Avatar(nome: String, imagePath: String, descrizione: String) - isActive: boolean - rotationAngle: double + getNome(): String - panelColor: Color - playerINdex: int + getImagePath(): String - opacity: float - scartiPanel: JLayeredPane + getDescrizione(): String - imagePanel: JPanel - trashCard: JButton + getAvatar(): List<Avatar> <static> - trashButton: JButton + getAvatarBoardBvIndex(index: int): Avatar <static> + AvatarPanel(avatarIndex: int, remainingCards: int) - zIndex: int createAvatarPanel() - playerPanelList: List<PlayerPanel> + setActive(isActive: boolean) **AvatarRadioButton** updatePanelStyle() + BoardPanel(playerNumber: int) # paintComponent(g: Graphics) + setNewBoard() - icon: Imagelcon + getPlayerImage(): BufferedImage createBoardPanel() - isSelected: boolean **JRadioButton** + getCharacterName(): String Extends - createCenterPanel(): JPanel - setValue(playerInex: int, cardWidth: int, rotationAngle: double) + AvatarRadioButton(icon: ImageIcon) + addScarti(cardName: String) Extends # paintComponent(g: Graphics) + removeTrash() Extends + getPreferredSize(): Dimension + setActivePanel(index: int) **JButton** + createWinnerPopup(winnerName: String, winImage: BufferedImage) **JPanel** _____ Extends CusomFont <abstract> + loadFont(style: int, size: float): Font <static> CardButton Extends - image: Imagelcon rotationAngle: double **FuturisticButton** - cardWidth: int cardHeight: int - BUTTON COLOR: Color <static> <final> - HOVER COLOR: Color <static> <final> + CardButton(imagePath: String, cardWidth: int) - TEXT COLOR: Color <static> <final> + setRotationAngle(angle: double) - DISABLED COLOR: Font <static> <final> # paintComponent(g: Graphics) - BUTTON FONT: Font <static> <final> - isRed: boolean $\langle \rangle$ Extends Extends + FuturisticButton(text: String) + FuturisticButton(text: String, icon: ImageIcon) DownCardButton **FaceUpCardButton** - setStyle() + setRed(isRed: boolean) cardName: String + DownCardButton(cardWidth: int) + setEneabled(enabled: boolean) # painComponent(Graphics g) + FaceUpCardButton(cardName: String, cardWidth: int) + toString(): String

View

