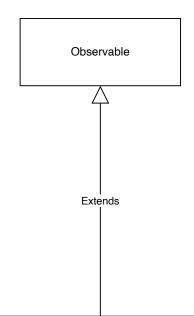
JBlackJack

- + BOARD_WIDTH: int
- + BOARD_HEIGHT: int
- + BACKGROUND_COLOR: Color
- start: Starthome: Home
- onePlayer: OnePlayer
- twoPlayers: TwoPlayers
- threePlayers: ThreePlayers
- betPanel: Bet
- avatar: int
- importo: int
- winCount: int
- defeatCount: int
- drawCount: int
- avatarBot1: int
- winCountBot1: int
- defeatCountBot1: int
- drawCountBot1: int
- avatarBot2: int
- winCountBot2: int
- defeatCountBot2: int
- drawCountBot2: int
- frame: JFrame
- + main(String[] args): void
- setupGUI(): void
- startGame(): void
- generateBotAvatars(): void
- switchAvatarLeft(): void
- switchAvatarRight(): void
- bet(String nickname, int avatar, int players): void
- $\hbox{-} startGameWithPlayers (String\ nickname,\ int\ avatar,\ int\ players):}\ void$
- onePlayerSelected(String nickname, int avatar): void
- twoPlayersSelected(String nickname, int avatar): void
- threePlayersSelected(String nickname, int avatar): void
- backHome(String nickname, int avatar, int result, int bot1result, int bot2result): void
- updateCounts(int result, int bot1result, int bot2result): void
- $\hbox{--updateBotCounts(int result, int botNumber): void}\\$
- switchToPanel(JPanel panel): void



Model

- deck: List<Card>

- random: Random

- hiddenCard: Card

- dealerHand: List<Card>

- playerHand: List<Card>

- bot1Hand: List<Card>

- bot2Hand: List<Card>

- dealerSum: int

- playerSum: int

- bot1Sum: int

- bot2Sum: int

- dealerAceCount: int

- playerAceCount: int

- bot1AceCount: int

- bot2AceCount: int

- players: int

+ Model(int players)

+ notifyObservers(): void

+ startNewGame(): void

- buildDeck(): void

- shuffleDeck(): void

+ drawCard(): Card

+ reduceAce(int sum, int aceCount): int

+ getHiddenCard(): Card

+ getDealerHand(): List<Card>

+ getPlayerHand(): List<Card>

+ getBot1Hand(): List<Card>

+ getBot2Hand(): List<Card>

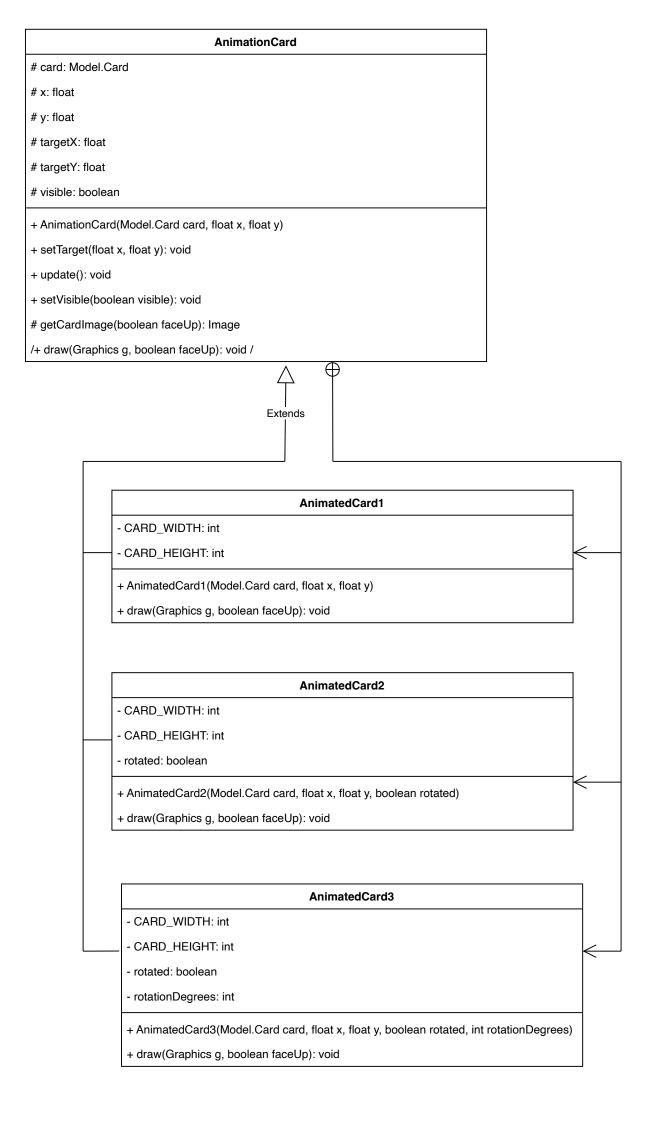
+ getDealerSum(): int

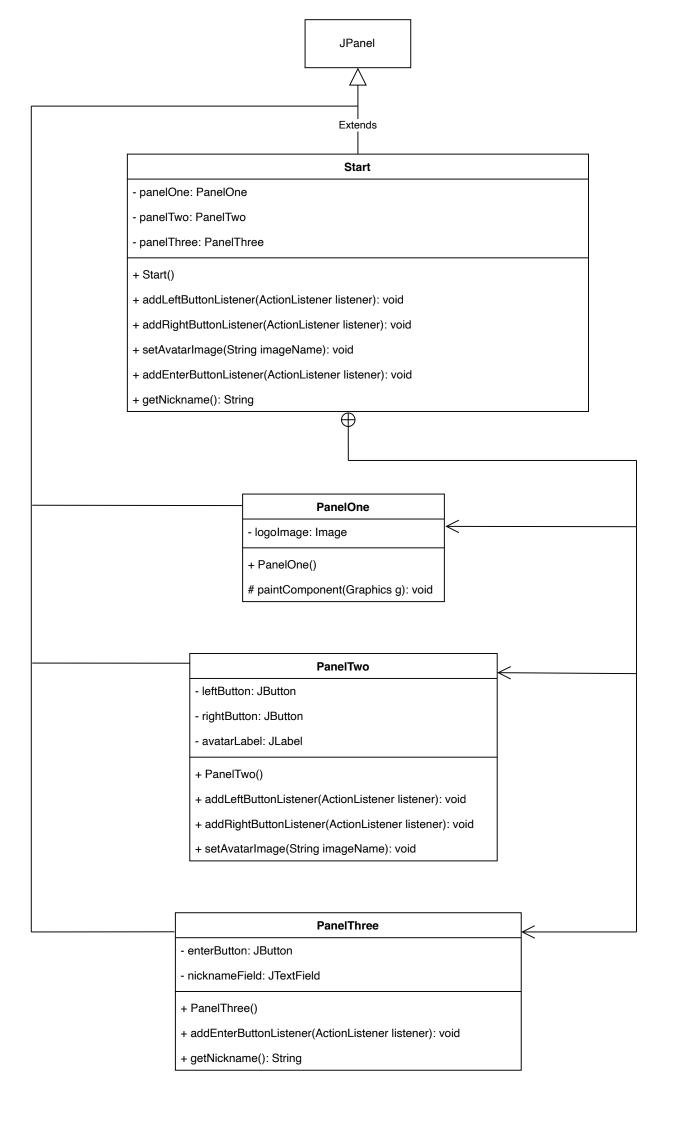
+ getPlayerSum(): int

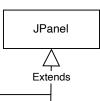
+ getBot1Sum(): int + getBot2Sum(): int + getDealerAceCount(): int + getPlayerAceCount(): int + getBot1AceCount(): int + getBot2AceCount(): int + getPlayers(): int + setDealerSum(int sum): void + setPlayerrSum(int sum): void + setBot1Sum(int sum): void + setBot2Sum(int sum): void + setDealerAceCount(int count): void + setPlayerAceCount(int count): void + setBot1AceCount(int count): void + setBot2AceCount(int count): void \oplus Card - value: String - type: String + Card(String value, String type) + getValue(): int + isAce(): boolean + getImagePath(): String + toString(): String

Controller - model: Model - view: GameView + Controller(Model model, GameView view) - hitAction(): void - stayAction(): void

AudioManager - instance: AudioManager - clip: Clip + getInstance(): AudioManager + play(String filename): void







Home

- nickname: String

- winCount: int

- defeatCount: int

- drawCount: int

- importo: int

- avatarBot1image: Image

- avatarBot2image: Image

- winCountBot1: int

- defeatCountBot1: int

- drawCountBot1: int

- winCountBot2: int

- defeatCountBot2: int

- drawCountBot2: int

- avatarlmage: Image

- panelOne: PanelOne

- panelTwo: PanelTwo

- + Home(String nickname, int avatar, int winCount, int defeatCount, int drawCount, int importo, int avatarBot1, int avatarBot2, int winCountBot1, int defeatCountBot1, int drawCountBot2, int drawCountBot2)
- loadAvatarImage(int avatar): Image
- + addOnePlayerButtonListener(ActionListener listener): void
- + addTwoPlayersButtonListener(ActionListener listener): void
- + addThreePlayersButtonListener(ActionListener listener): void

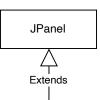


PanelOne

- + PanelOne()
- createAvatarPanel(String nickname, Image avatarImage, int winCount, int defeatCount, Font font, int avatarWidth, int avatarHeight): JPanel

PanelTwo

- onePlayerButton: JButton
- twoPlayerButton: JButton
- threePlayerButton: JButton
- + PanelTwo()
- addCard(JPanel panel, String imagePath, String labelText): JButton
- + addOnePlayerButtonListener(ActionListener listener): void
- + addTwoPlayerButtonListener(ActionListener listener): void
- + addThreePlayerButtonListener(ActionListener listener): void



Bet

- importo: int

+ puntata: int

- betButton: JButton

- chipButtons: Map<Integer, CircularButton>

- chipImages: Map<Integer, Image>

- chipValues: Map<Image, Integer>

- CHIP_WIDTH: int

- CHIP_HEIGHT: int

- SPACE_BETWEEN_CHIPS: int

- TOTAL_WIDTH: int

- START_X: int

- START_Y: int

- CIRCLE_DIAMETER: int

- ANIMATION_DURATION: int

- FRAMES_PER_SECOND: int

- FRAME_DELAY: int

- targetX: int

- targetY: int

- isAnimating: boolean

- animationTimer: Timer

- circles: List<MovingCircle>

+ Bet(int importo)

- initBetButton(): void

- loadChipImages(): void

- initChipButtons(): void

- createChipButton(Image chipImage): CircularButton

- createAndAddChipButtons(): void

- startAnimationForChip(int chipValue): void

- updatePuntata(int chipValue): void

- updateImporto(int chipValue): void

- getChipIndex(int chipValue): int

- disableButtonForChipValue(int chipValue): void

- enableButtonForChipValue(int chipValue): void

paintComponent(Graphics g): void

+ addBetButtonListener(ActionListener listener): void

+ removeBetButtonListener(): void

+ addChipButtonListener(int chipValue, ActionListener listener): void

MovingCircle

- startX: int

- startY: int

- targetX: int

- targetY: int

- x: int

- y: int

totalFrames: intcurrentFrame: int

- chipImage: Image

+ MovingCircle(int startX, int startY, int targetX, int targetY, int totalFrames, Image chipImage)

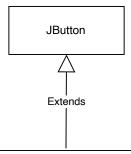
+ update(): void

+ isFinished(): boolean

+ getX(): int

+ getY(): int

+ getImage(): Image



CircularButton

+ CircularButton(Icon icon)

paintComponent(Graphics g): void

JPanel

<<Interface>> GameView

- + getHitButton(): JButton + getStayButton(): JButton + updateView(Model model): void + addHitButtonListener(ActionListener listener): void + addStayButtonListener(ActionListener listener): void
- + setHitButtonEnable(boolean enabled): void
- + setStayButtonEnable(boolean enabled): void + createButton(String text): JButton

Observer

Extends

	OnePlayer
	+ gamePanel: GamePanel
	- CARD_WIDTH: int
	- buttonPanel: JPanel
	- hitButton: JButton
	- stayButton: JButton
	- gameEnded: boolean
	- result: int
	+ OnePlayer()
	- initializePanel(): void
	- createButtonPanel(): JPanel
	+ createButton(String text): JButton
	+ getHitButton(): JButton
	+ getStayButton(): JButton
	+ getResult(): int
	+ getResult(): int
	+ updateView(Model model): void
	+ update(Observable o, Object arg): void
	
	√ GamePanel
-	- model: Model
	- homeButton: JButton
	- playerCards: List <animationcard.animatedcard1></animationcard.animatedcard1>
	- dealerCards: List <animationcard.animatedcard1></animationcard.animatedcard1>
	- animationTimer: Timer
	- dealerCardsInitialized: boolean
	+ GamePanel()
	- initalizePanel(): void
	- getHomeButtonBounds(): Rectangle
	- updateAnimations(): void
	+ setModel(Model model): void
	- initializeDealerCards(List <model.card> hand, Model.Card hiddenCard): void</model.card>
	- updateDealerCardPositions(): void
	- updatePlayerCardList(List <model.card> hand): void</model.card>
	 updateCardPositions(List<animationcard.animatedcard1> cards, int y): void</animationcard.animatedcard1>
	+ startDealerRevealAnimation(): void
	- updateDealerCardPositions(int newCards): void
	# paintComponent(Graphics g): void
	- drawDealerInfo(Graphics g): void
	- drawDealerCards(Graphics g): void
	- drawPlayerCards(Graphics g): void
	- showGameResult(Graphics g): void

TwoPlayers
+ gamePanel: GamePanel
+ CARD_WIDTH: int
+ CARD_HEIGHT: int
- buttonPanel: JPanel
- hitButton: JButton
- stayButton: JButton
- homeButton: JButton
- gameEnded: boolean
- result: int
- bot1Result: int
+ TwoPlayers()
- setupLayout(): void
- initializeComponents(): void
- createButtonPanel(): JPanel
+ createButton(String text): JButton
+ getHitButton(): JButton
+ getStayButton(): JButton
+ getResult(): int
+ getBot1Result(): int
+ updateView(Model model): void
+ update(Observable o, Object arg): void
T appears (essectivations), void
\downarrow
GamePanel
- model: Model
- playerCards: List <animationcard.animatedcard2></animationcard.animatedcard2>
- dealerCards: List <animationcard.animatedcard2></animationcard.animatedcard2>
- botCards: List <animationcard.animatedcard2></animationcard.animatedcard2>
- animationTimer: Timer
- dealerCardsInitialized: boolean
- botCardsInitialized: boolean
+ GamePanel()
- setupPanel(): void
- setupHomeButton(): void
- setupAnimationTimer(): void
+ setModel(Model model): void
- initializeDealerCards(List <model.card> hand, Model.Card hiddenCard): void</model.card>
- initializeBotCards(List <model.card> hand): void</model.card>

- createAnimatedCard(Model.Card card, float x, float y, boolean isBot): AnimationCard.AnimatedCard2

- updateDealerCardPositions(): void
- updateBotCardPositions(): void
- updatePlayerCardList(List<Model.Card> hand): void
- updatePlayerCardPosition(): void
- updateCardPositions(): void
- + startDealerRevealAnimation(): void
- + startBotRevealAnimation(): void
- # paintComponent(Graphics g): void
- drawCards(Graphics g): void
- drawLabels(Graphics g): void
- drawDealerLabel(Graphics g, Fontmetrics fm): void
- drawPlayer2Label(Graphics g, Fontmetrics fm): void
- showGameResult(Graphics g): void
- determineResults(int playerFinalSum, int dealerFinalSum, int bo1FinalSum): void
- calculateResult(int playerSum, int dealerSum): int
- getResultMessage(int playerSum, int dealerSum): String
- drawResultMessage(Graphics g, FontMetrics fm, String message): void
- + addHomeButtonListener(ActionListener listener): void

ThreePlayers + gamePanel: GamePanel - CARD_WIDTH: int - CARD_HEIGHT: int - buttonPanel: JPanel - hitButton: JButton - stayButton: JButton - homeButton: JButton - gameEnded: boolean - result: int - bot1Result: int - bot2Result: int + ThreePlayers() - initializeGamePanel(): void - initializeButtonPanel(): void + createButton(String text): JButton + getHitButton(): JButton + getStayButton(): JButton + getResult(): int + getBot1Result(): int + getBot2Result(): int + updateView(Model model): void + update(Observable o, Object arg): void **GamePanel** - model: Model - playerCards: List<AnimationCard.AnimatedCard2> - dealerCards: List<AnimationCard.AnimatedCard2> - bot1Cards: List<AnimationCard.AnimatedCard2> - bot2Cards: List<AnimationCard.AnimatedCard2> - animationTimer: Timer - dealerCardsInitialized: boolean - bot1CardsInitialized: boolean - bot2CardsInitialized: boolean

- TIMER_DELAY: int

- + GamePanel()
- initializeHomeButton(): void
- initializeAnimationTimer(): void
- + setModel(Model model): void
- initializeDealerCards(List<Model.Card> hand, Model.Card hiddenCard): void
- initializeBot1Cards(List<Model.Card> hand): void
- initializeBot2Cards(List<Model.Card> hand): void
- updatePlayerCardList(List<Model.Card> hand): void
- updateCardPositions(): void
- updatePlayerCardPosition(): void
- updateDealerCardPositions(): void
- updateBot1CardPositions(): void
- updateBot2CardPositions(): void
- + startDealerRevealAnimation(): void
- + startBot1RevealAnimation(): void
- + startBot2RevealAnimation(): void
- # paintComponent(Graphics g): void
- drawLabels(Graphics g): void
- drawLabel(Graphics g, String text, int x, int y, double angle): void
- drawRotatedLabel(Graphics g, String text, int x, int y, double angle): void
- drawCards(Graphics g): void
- showGameResult(Graphics g): void
- determineResult(int finalSum, int dealerFinalSum): void
- getResultMessage(int resultCode): String
- $+\ add Home Button Listener (Action Listener\ listener):\ void$