

Player
«get/set» + hands: ArrayList<Card> «get/set» + isTurn: boolean
+ Player(ArrayList<Card> hands, boolean newTurn) + updateHand(int index, Card card): void + startTurn(): void + endTurn(): void

Card
«get/set» + color: String «get/set» + number: String «get/set» + clickable: boolean «get/set» + selected: boolean «get/set» + hovering: boolean
+ Card(String newColor, String newNumber)

Button
«get/set» + location: Pair<int, int> «get/set» + clickable: boolean «get/set» + text: String «get/set» + selected: boolean «get/set» + hovering: boolean
+ Button(Pair<int, int> newLocation, String newText)

<<Interface>> ButtonListener
+ selected(): boolean + hovering(): boolean

<<Interface>> GameStateModelListener
+ modelChange(): void

GameStateModel
«get/set» + hintsRemain: int «get/set» + fusesRemain: int «get/set» + cardsRemain: int «get/set» + timeRemain: int «get/set» + score: int «get/set» + cardsOnTable: ArrayList<Card> «get/set» + discardPile: LinkedList<Card> «get/set» + players: ArrayList<Player> «get/set» + turn: int «get/set» + subscribers: ArrayList<GameStateModelListener>
+ GameStateModel(Array<Player> newPlayers) + appendDiscard (Card card): void + updateCardsOnTable(Card card): void + updatePlayerHand(Player player, Card card): void + incrementHints(): void + decrementHints(): void + decrementFuses(): void + decrementCards(): void + decrementTime(): void + addSubscriber(GameStateModelListener Sub): void + notifySubscribers(): void

ClientStateModel
«get/set» + playerSelected: int «get/set» + cardsSelected: ArrayList<int> «get/set» + menusOpen: Map<String, boolean> «get/set» + actionSelected: String «get/set» + notes: ArrayList<int> «get/set» + buttons: ArrayList<Button>
+ ClientStateModel(Player newPlayerSelected, Map<String, boolean> newMenusOpen, String newActionSelected, ArrayList<int> notes, ArrayList<Button> newButtons) + selectCard(int playerIndex, int cardIndex): void + toggleMenu(String name): void + selectAction(String action): void + getSelectedPlayer(): void + getSelectedCards(): void + getAction(): void