controller

Controller -game : Game -id:int -connectedPlayers : CurrentHashMap<Connection, Player> -turnHandler : TurnHandler -actionQueue : BlockingQueue < ActionMessage > -playerInTurn : String -setupFinished : boolean -endingCycle : boolean -processingAction : boolean +getId():int +addAction(ActionMessage): void +execAction() : void +pickQueue() : void +joinGame(Connection): boolean -startGame() : void +drawCard(Connection, String): void +drawCardFromBoard(Connection, int): void +placeCard(Connection, Card, int[]): void +placeStarterCard(Connection, Card): void +chooseObj(Connection, Achievement): void +chooseColor(Connection): void +setColor(Connection, Color): void +getPlayerInTurn(): String +sendChatMessage(Connection, Connection, String): void +sendChatMessage(Connection, String): void +getPlayerByClient(Connection): Player +disconnectedWhileInTurn(String): void +disconnectedWhileSetupping(Connection, Boolean): void +reconnectBackup(Connection): void +getGame() : Game +getTurnHandler(): TurnHandler +getConnectedPlayersMessage(): void +removeFromServer() : void TurnHandler -game : Game -controller : Controller -RECONNECTION_TIME : int -j : int ServerController -endingCycle : boolean -server : Server -endingByPlacingCard : boolean -gameCreationLock : Object -disconnectedWhileInTurn: boolean +createLobby(String, int, int): Controller -endCountDown : int +joinLobby(String, Controller): boolean +startEnd(Player): void +disconnectedWhileInTurn(Player): void +changePlayerState(Player) : void -declareWinnerByDisconnection(Player): void +changeSetupPlayer(): String