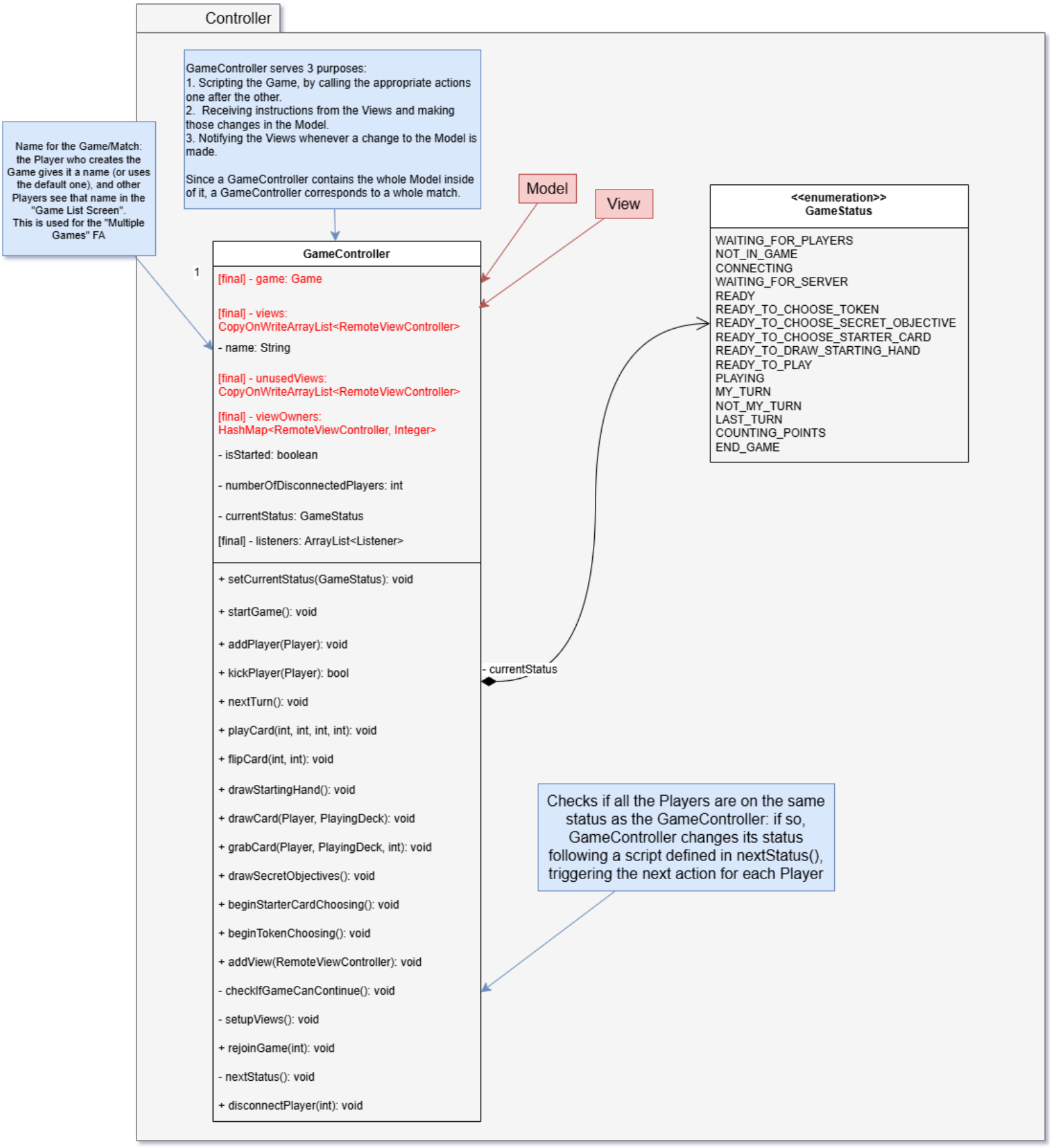
Controller
GameController serves 3 purposes:
1. Scripting the Game, by calling the appropriate actions one after the other.
2. Receiving instructions from the Views and making those changes in the Model.
3. Notifying the Views whenever a change to the Model is made.
Since a GameController contains the whole Model inside of it, a GameController corresponds to a whole match.
Name for the Game/Match: the Player who creates the Game gives it a name (or uses the default one), and other Players see that name in the "Game List Screen". This is used for the "Multiple Games" FA
Model
View
GameController
1
[final] - game: Game
[final] - views: CopyOnWriteArrayList<RemoteViewController>
- name: String
[final] - unusedViews: CopyOnWriteArrayList<RemoteViewController>
[final] - viewOwners: HashMap<RemoteViewController, Integer>
- isStarted: boolean
- numberOfDisconnectedPlayers: int
- currentStatus: GameStatus
[final] - listeners: ArrayList<Listener>
+ setCurrentStatus(GameStatus): void
+ startGame(): void
+ addPlayer(Player): void
+ kickPlayer(Player): bool
+ nextTurn(): void
+ playCard(int, int, int, int): void
+ flipCard(int, int): void
+ drawStartingHand(): void
+ drawCard(Player, PlayingDeck): void
+ grabCard(Player, PlayingDeck, int): void
+ drawSecretObjectives(): void
+ beginStarterCardChoosing(): void
+ beginTokenChoosing(): void
+ addView(RemoteViewController): void
- checkIfGameCanContinue(): void
- setupViews(): void
+ rejoinGame(int): void
- nextStatus(): void
+ disconnectPlayer(int): void
- currentStatus
<<enumeration>>
GameStatus
WAITING_FOR_PLAYERS
NOT_IN_GAME
CONNECTING
WAITING_FOR_SERVER
READY
READY_TO_CHOOSE_TOKEN
READY_TO_CHOOSE_SECRET_OBJECTIVE
READY_TO_CHOOSE_STARTER_CARD
READY_TO_DRAW_STARTING_HAND
READY_TO_PLAY
PLAYING
MY_TURN
NOT_MY_TURN
LAST_TURN
COUNTING_POINTS
END_GAME
Checks if all the Players are on the same status as the GameController: if so, GameController changes its status following a script defined in nextStatus(), triggering the next action for each Player