Network Protocol Documentation for group AM27

<Giuliano Crescimbeni, Mauro Leporace, Luca De Nicola, Cristian Dheskali>

The described implementation supports both network technologies (Socket and RMI) and the additional feature of Multiple Matches.

The server is composed of 5 components for managing connections and matches:

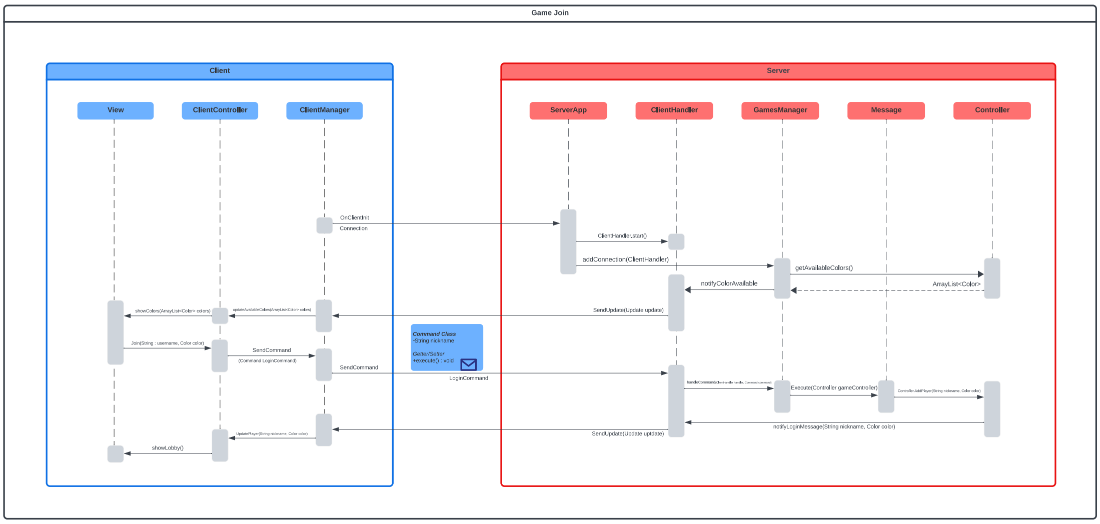
- **ServerApp**: Responsible for handling new incoming connections. For each connection, it generates a ClientHandler on a new thread.
- **ClientHandler**: Class responsible for exchanging messages over the network. It has two functions for message exchange: one for receiving and one for sending. Incoming messages are forwarded to the GamesManager.
- **GamesManager**: This is the main class for connection to match ID associations. It is instantiated only once by the ServerApp. For each incoming message, through the handleMessage() function, the GamesManager obtains the gameId of the client and the match controller, so it can execute the actions on the correct controller.
  - **Controller**
  - **Model**

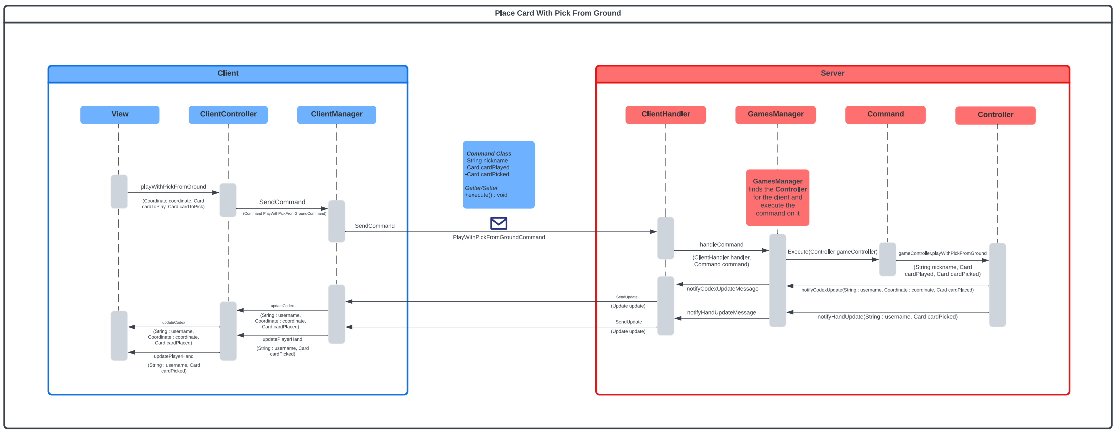
Messaging

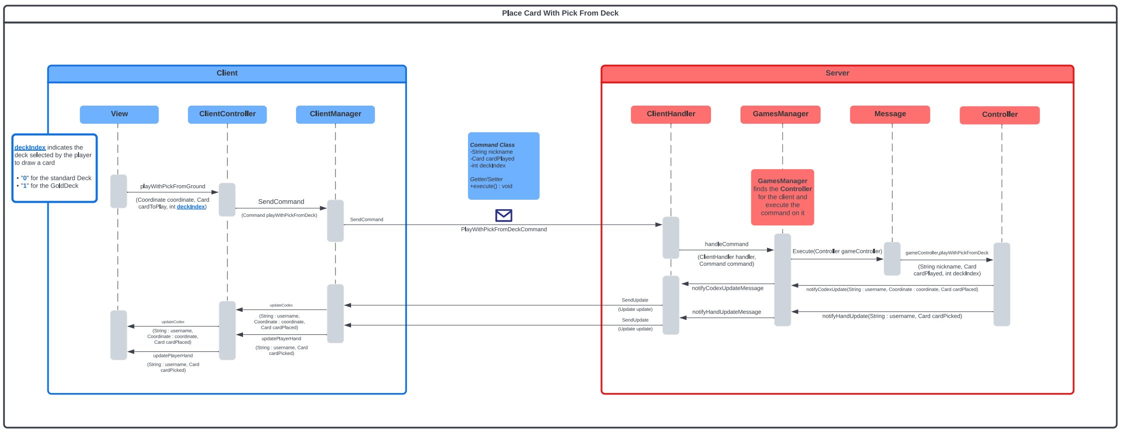
The exchange of messages occurs through the serialization of a class called Command. The subclasses of Command each represent a distinct client action. All subclasses override an execute() method for command execution. The function requires the passing of a Controller, which is provided by the GamesManager.

Updates to clients are carried out through a class called Update. Their functioning is similar to that of the Command class, so a detailed description is omitted.

Below are three scenarios of client-server interactions:


Game Join
Client
View
ClientController
ClientManager
OnClientInit
Connection
showColors(ArrayList<Color> colors)
updateAvailableColors(ArrayList<Color> colors)
Join(String : username, Color color)
SendCommand
(Command LoginCommand)
SendCommand
Command Class
-String nickname
Getter/Setter
+execute() : void
LoginCommand
UpdatePlayer(String nickname, Color color)
showLobby()
Server
ServerApp
ClientHandler
GamesManager
Message
Controller
ClientHandler.start()
addConnection(ClientHandler)
getAvailableColors()
notifyColorAvailable
ArrayList<Color>
SendUpdate(Update update)
handleCommand(ClientHandler handler, Command command)
Execute(Controller gameController)
Controller.AddPlayer(String nickname, Color color)
notifyLoginMessage(String nickname, Color color)
SendUpdate(Update uptdate)


Place Card With Pick From Ground
Client
View
ClientController
ClientManager
playWithPickFromGround
(Coordinate coordinate, Card cardToPlay, Card cardToPick)
SendCommand
(Command PlayWithPickFromGroundCommand)
SendCommand
Command Class
-String nickname
-Card cardPlayed
-Card cardPicked
Getter/Setter
+execute() : void
PlayWithPickFromGroundCommand
updateCodex
(String : username, Coordinate : coordinate, Card cardPlaced)
updatePlayerHand
(String : username, Card cardPicked)
updateCodex
(String : username, Coordinate : coordinate, Card cardPlaced)
updatePlayerHand
(String : username, Card cardPicked)
Server
ClientHandler
GamesManager
Command
Controller
GamesManager finds the Controller for the client and execute the command on it
handleCommand
(ClientHandler handler, Command command)
Execute(Controller gameController)
gameController.playWithPickFromGround
(String nickname, Card cardPlayed, Card cardPicked)
notifyCodexUpdateMessage
notifyCodexUpdate(String : username, Coordinate : coordinate, Card cardPlaced)
notifyHandUpdateMessage
notifyHandUpdate(String : username, Card cardPicked)
SendUpdate
(Update update)
SendUpdate
(Update update)


Place Card With Pick From Deck
Client
View
ClientController
ClientManager
deckIndex indicates the deck selected by the player to draw a card
• "0" for the standard Deck
• "1" for the GoldDeck
playWithPickFromGround
(Coordinate coordinate, Card cardToPlay, int deckIndex)
SendCommand
(Command playWithPickFromDeck)
SendCommand
Command Class
-String nickname
-Card cardPlayed
-int deckIndex
Getter/Setter
+execute() : void
PlayWithPickFromDeckCommand
updateCodex
(String : username, Coordinate : coordinate, Card cardPlaced)
updatePlayerHand
(String : username, Card cardPicked)
updateCodex
(String : username, Coordinate : coordinate, Card cardPlaced)
updatePlayerHand
(String : username, Card cardPicked)
Server
ClientHandler
GamesManager
Message
Controller
GamesManager finds the Controller for the client and execute the command on it
handleCommand
(ClientHandler handler, Command command)
Execute(Controller gameController)
gameController.playWithPickFromDeck
(String nickname, Card cardPlayed, int deckIndex)
notifyCodexUpdateMessage
notifyCodexUpdate(String : username, Coordinate : coordinate, Card cardPlaced)
notifyHandUpdateMessage
notifyHandUpdate(String : username, Card cardPicked)
SendUpdate
(Update update)
SendUpdate
(Update update)