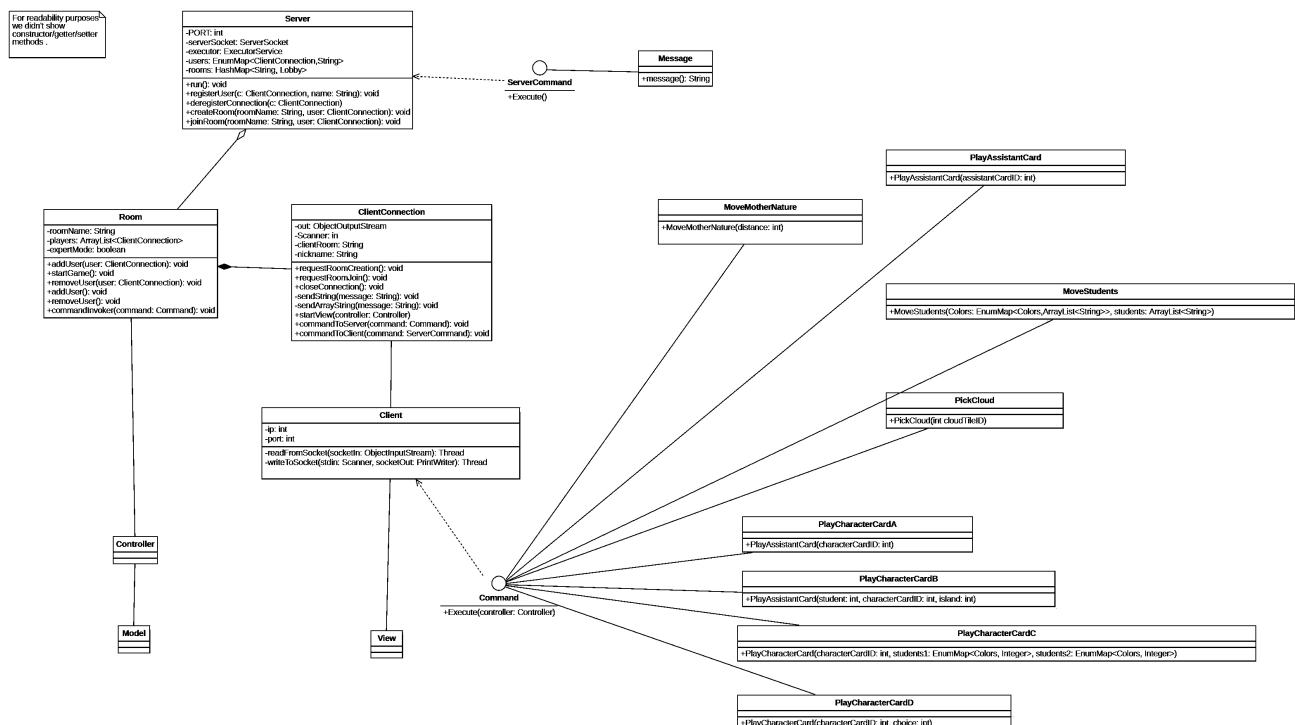


Network Uml Eryantis - Group GC23

The following UML class diagram represents the network for the board game Eryantis. The Server uses the ClientConnection to interact with the Client. The Server organizes the ClientConnections using Room instances. Each room has one controller and it's responsible for a single game. We use listeners between Model - Server, View - GUIapp or View-CLApp. We use Command Pattern to send actions from the clients to the server and to send messages from the Server to the clients.

The Client send a command, that represent a specific action, which is invoked by the Room (inside the Server). The room invokes the command, calling a method inside the Controller that immediately checks the parameters and calls the right method inside the Model.



The UML sequence diagram represents the typical communication between the client and server apps, stripped to its functional components to allow for readability.

The process starts with the View, that immediately instantiates the Client, which in turn immediately tries to connect to the Server: if a connection is secured, the Client is notified, and passes on the information to the View, which updates to the "lobby mode".

The View directly controls each Client action, and receives feedback to it, which is displayed.

A Client will create a Room, which then everyone can act upon by sending messages to the Server: this does not add much complexity to management of the rooms, while allowing for actions such as requesting the list of Rooms on the Server.

Eventually, the Client Leader will start the game: the Server will receive the input and pass it on to Room, which will change the View to "game mode" and instantiate the Controller, which creates the Game.

Now, every further communication will take place between the View and Controller, which will then act upon the Game: through the use of a Listener the View will be updated on every such action, and the game will play out to its end.

