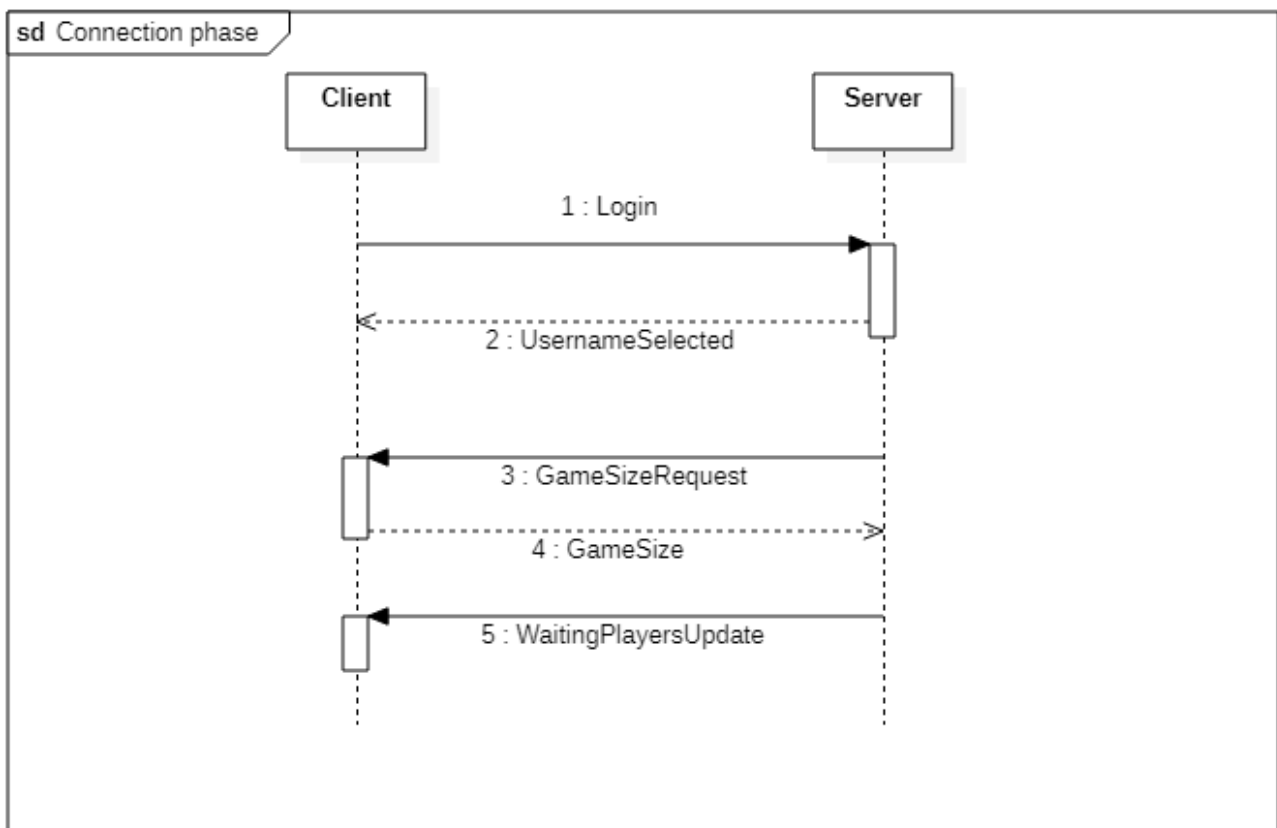


# COMMUNICATION PROTOCOL

## Connection phase

During the connection phase all the messages have the purpose of establish the connection with the server (if the game is online) and to setup all the information necessary to properly start a game.

In this phase, the exchange of messages is shown by the image below.



The **Login** message contain the username chose by the player. If the username of the player is not already used the server will respond with a **UsernameSelected** message, otherwise the server will send an **ErrorUpdate** message to notify the player to choose another username.

If the player is the first to connect to the server, the server will ask for the number of players who will be part of the game thanks to a **GameSizeRequest** message and the player will send back the number of players wanted in a **GameSize** message.

If the player is not the one who choose the size of the game or the player has already sent the game size, he will wait for **WaitingPlayersUpdate** which contain the current size of the lobby, in that way the player knows in every moment how many players remain to start the game.

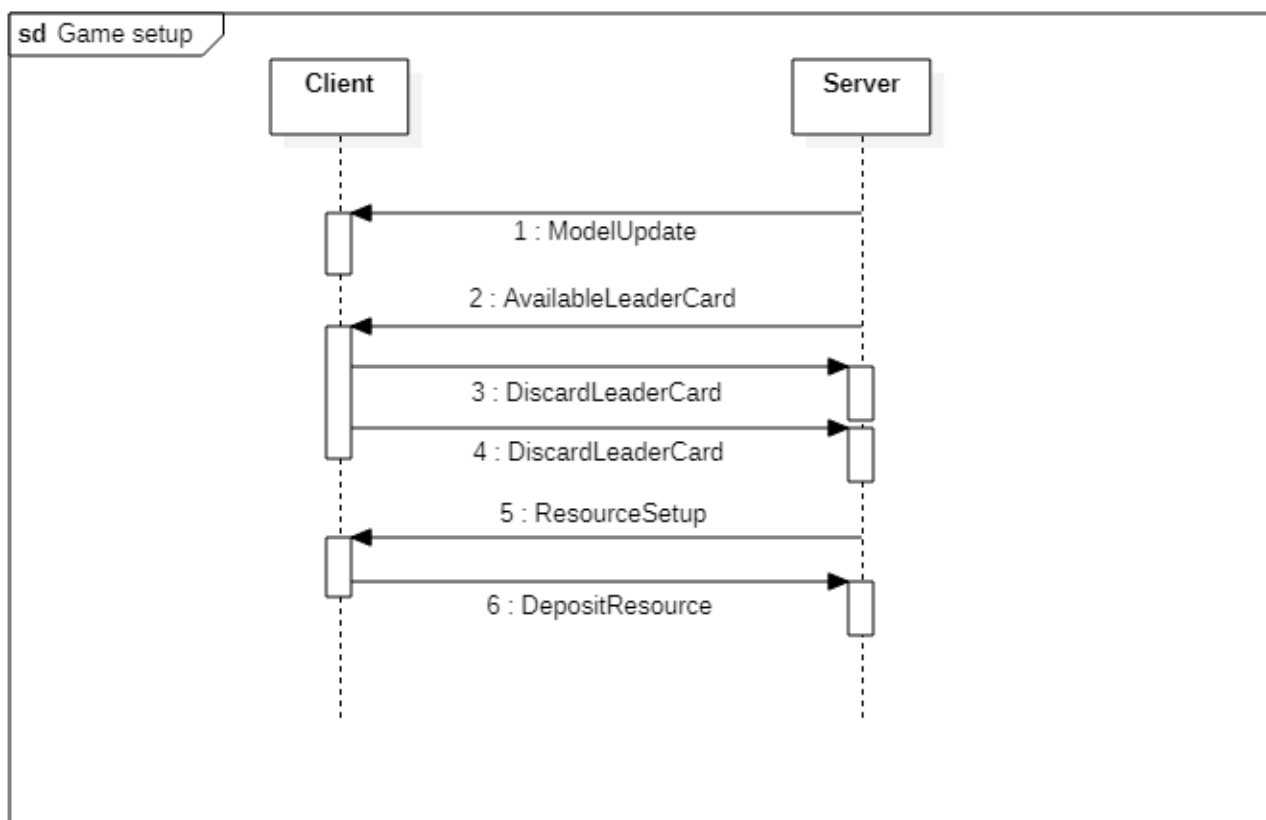
When the lobby is full the Game Phase could start.

## Game Phase

In this phase all the messages are the actual messages that are used to advance the state of the game. We choose to implement a client-side light model which is only a data container useful for read, in an easy way, all the information needed by user interface. So, all the messages which must update the light model we have decided to encapsulate them in a **ModelUpdate** message.

The Game Phase start at the first ModelUpdate from the model. This first ModelUpdate contain a **PlayerSetup** message which has the order of play of the player. After the first model update, to correctly setup the resource and cards of every player will arrive an **AvailableLeaderCard** message. This message has the list of leader cards to show to the player before the start of the game. The player must choose two of them and this produce 2 **DiscardLeaderCard** messages which contain only the id of the discarded leader card.

Eventually, some players have to choose 1 or 2 initial resources, and these are sent by the server through a **ResourceSetup** message with the number of resources to take. The client will choose its resource and will send 1 or 2 **DepositResource** messages which contain the color of the chosen resource and the number of shelf where to place it.



After this exchange of messages, the real game could start. During the game all the messages used to communicate some changes in the model are put in **ModelUpdates** and sent through these.

The messages that can be encapsulated in a ModelUpdate are the following:

- **ActionTokenPlayed**  
Message to notify the activation of an action token in single player.
- **ActivePlayer**

Message with the username of the player who is playing its turn.

- **AvailableLeadercard**  
Contain a list of leader card. It is used in the setup game but also to update the list of leader card in the light model after the discard of one of them.
- **ChestUpdate**  
Message with the update values of resources in the chest.
- **DevelopmentCardBuyUpdate**  
Message with the position where the development is placed and all the information of the card [color, id, victory points, require, result of production].
- **EndGame**  
Contain only the username of the winner and notify the end of the game.
- **LeaderCardActivation**  
Contain the leader card that has just been activated.
- **MarketCardUpdate**  
Message to notify the new card in the market after one is purchased. Contain the position in the market of the card to update [x, y] and the information of the card [id, victory points, price, require, production result, color]
- **MarketResourceAvailable**  
Message sent every time some resources are taken from the market. A message contains only one marble, then, every time a player take resources from the market multiple of these messages are sent. This message adds a marble to a resource buffer in the light model.
- **MarketResourceTaken**  
Message sent to remove a marble to the resource buffer in the light model.
- **ProductionBufferUpdate**  
Contains the list of resources generated by production. The list of resources is managed as a list of Marble.
- **ResourceMarketExtra**  
Notify the new bonus marble after the taking of resource from the marble.
- **ResourceMarketUpdate**  
Contain a list of marble and the position of the market where to updated it. Send to update the resource market after the taking of a row or column of it.
- **ShelfUpdate**  
Message to update a shelf. It contains the position of the shelf, its maximum size, its actual size and its color.
- **VaticanReport**  
Message to notify the activation of a Vatican report. It contains the index of the Vatican report and the list of usernames of the player who are in the area of activation.
- **VaticanRoutePosition**  
Contain the username of a player and its position in the Vatican route.
- **VictoryPointsUpdate**  
Contains the username of a player and its victory points

These were the messages that the model sends to the client.

The messages which could be sent by the client during the phases of the game are the following:

- **ActivateLeaderCard**  
Message to notify the activation of a leader card. It contains only the id of the card activated
- **BasicProduction**  
Message which contains the information of the activation of a basic production. This information are: first input resource, second input resource and output resource.
- **BuyDevelopmentCard**  
Message to notify the purchase of a development card. It contains the coordinate of the card in the market and the column of the production board where to place the card.
- **CardProduction**  
Message to notify a development card production and contains only the column of the production board where the card is placed.
- **DepositResource**  
Message used to deposit in the shelves the resources in the buffer of the light model. The fields are the color of the resource and the index of the shelf where to place it.
- **DiscardLeaderCard**  
Message to notify the discard of a leader card. It contains only the id of the card discarded.
- **EndTurn**  
Message sent to notify the end of the player turn, send only in the cases where the turn does not end automatically.
- **LeaderCardProduction**  
Contain the information of the leader card production chose by the player: the card id and the output resource wanted.
- **MoveResources**  
It contains the index of the shelf where to take the resources and the index of shelf where to move them.
- **TakeResources**  
Message for notify the taking of resources from the resource market. It has only a number which identify the row or column taken.

## Other messages

### ErrorUpdate

Every time an invalid action is tried by a player, the server notifies an ErrorUpdate which has a message inside that specified the type on invalid action was tried.

### Ping

Every client, since the start of the connection to the server, send every 1 second a Ping message to notify to the server that they are still connected. If the server does not receive a Ping message from a client for 5 seconds it will close the connection.