NETWORK PROTOCOL

Prova finale di Ingegneria del Software

AM19

Chizzola Andrea

Colombo Marco

Corigliano Davide

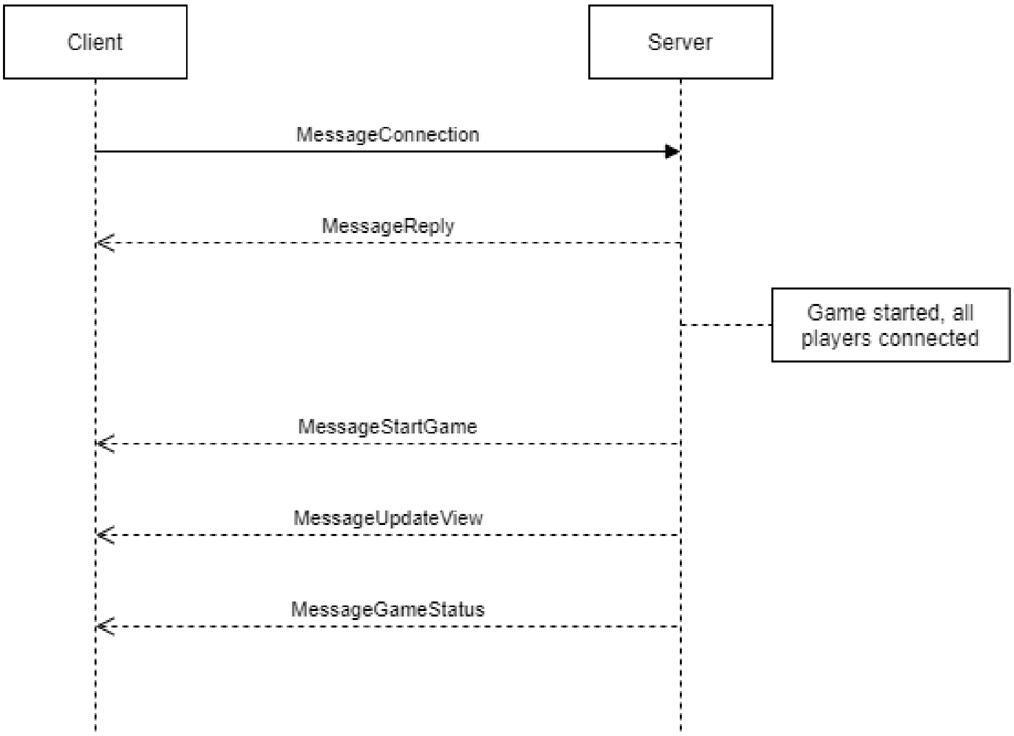
Summary

Introduction

This document illustrates the diagrams about the exchange of messages between client and server. In particular, those diagrams show the interaction sequences without negative reply messages from the server. If the client sends incorrect or unexpected messages, the server will reply with a negative response message. All the messages are in XML format and the structure of each message is reported in the final part of the document.
In the following diagrams MessageUpdateView represents one or more messages of view update.

1. Login phase

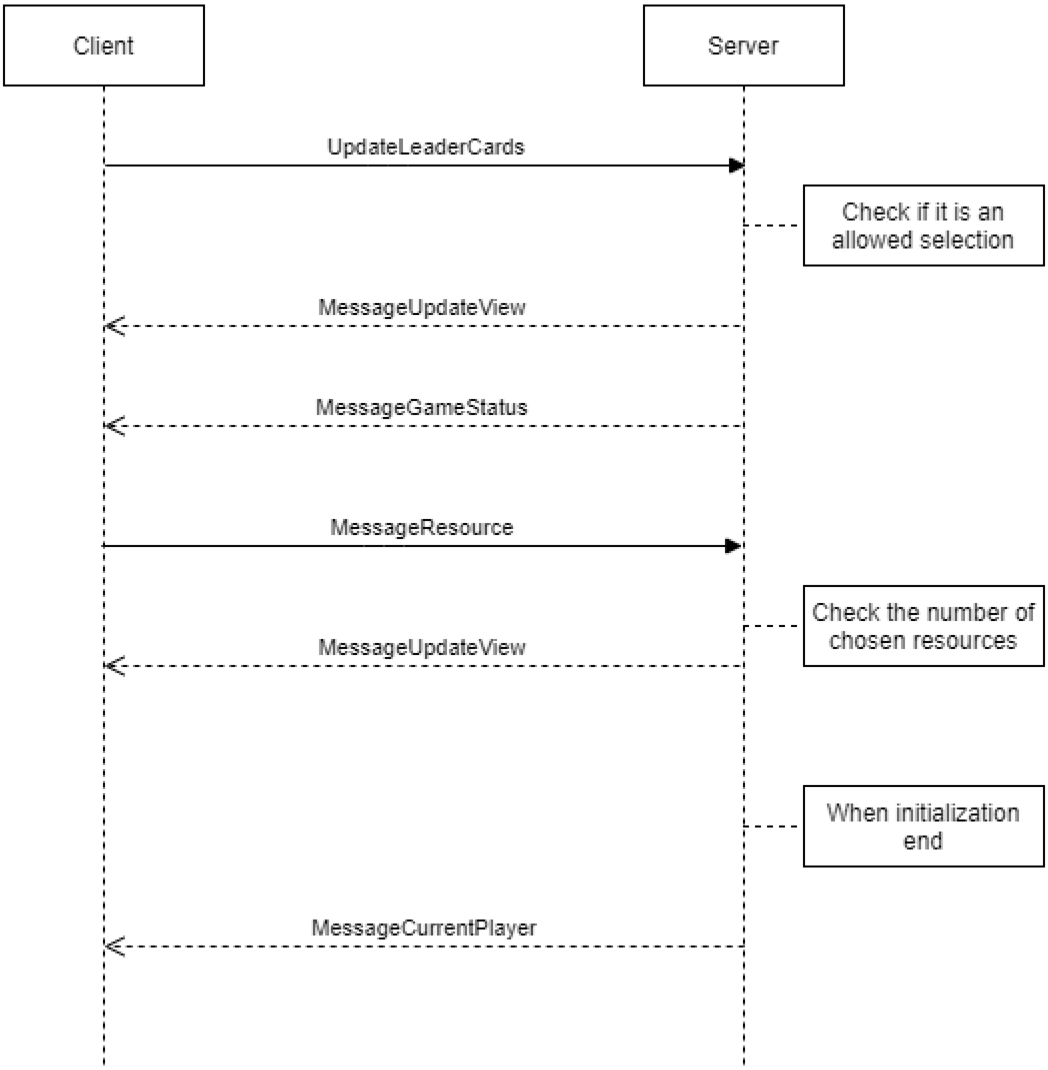
During this phase, players connect to the current game. The first player decides the number of players of the game, and all the Players give their own nickname.



2. Initialization

During this phase, the player initializes his game.

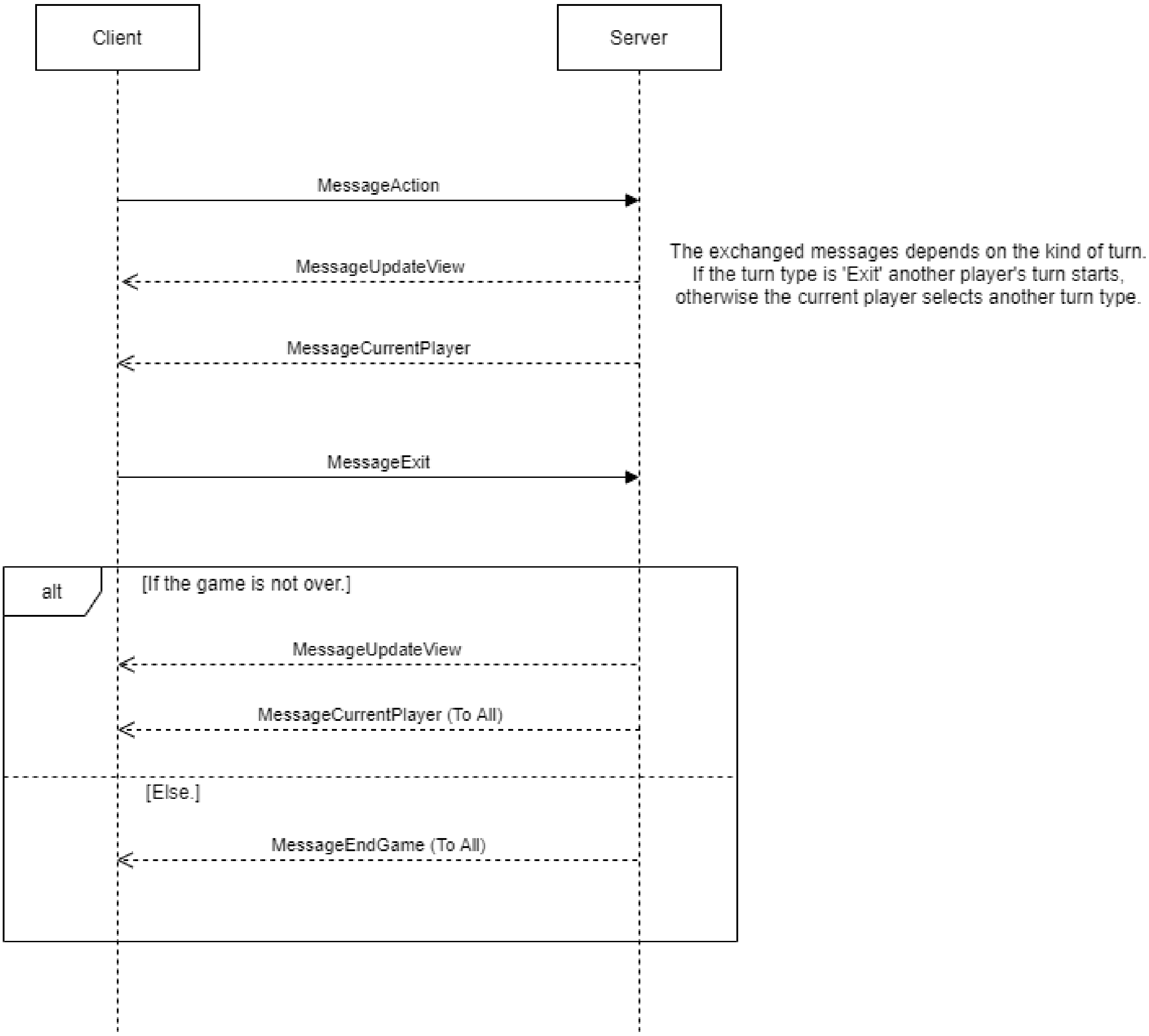
If a player disconnects himself during this phase, the game is over and all the players are disconnected.



3. Turn

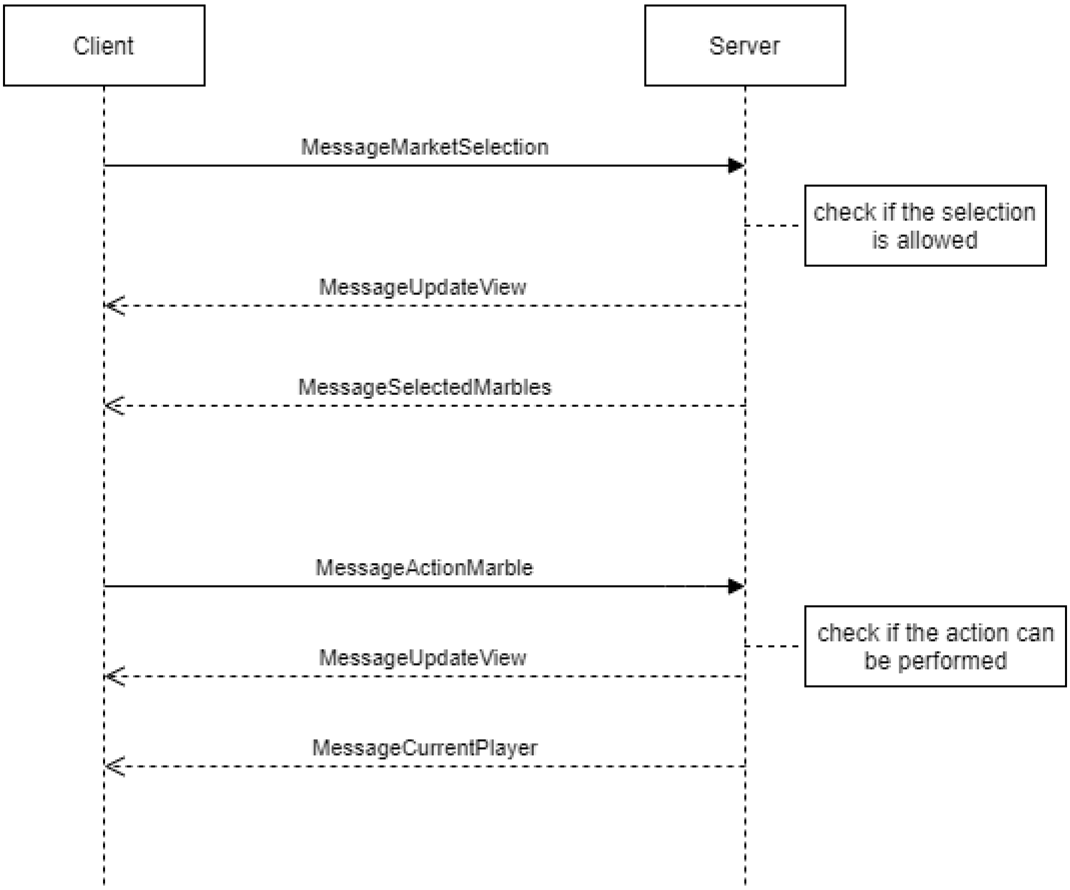
During this phase, the player manages his turn. In particular he can perform different type of actions until he decides to end his turn sending to the server the MessageExit.

The turn “Take Resources from Marketboard”, which corresponds to the MessageMarketSelection, involves the exchange of more than one message and therefore its description is on the next page.



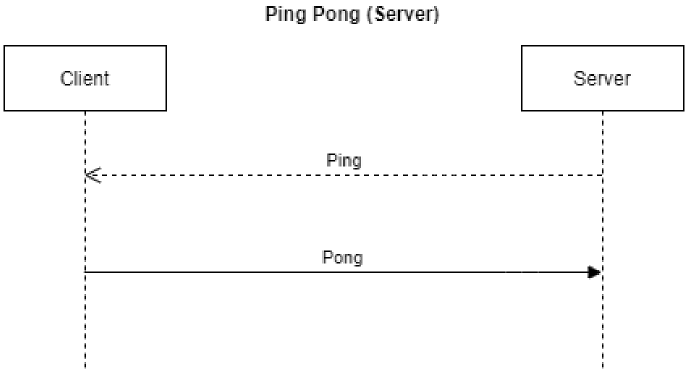
The exchanged messages depends on the kind of turn. If the turn type is 'Exit' another player's turn starts, otherwise the current player selects another turn type.

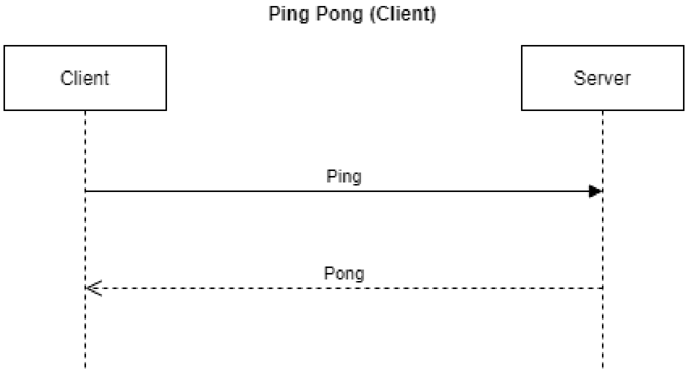
Take Resources from Marketboard



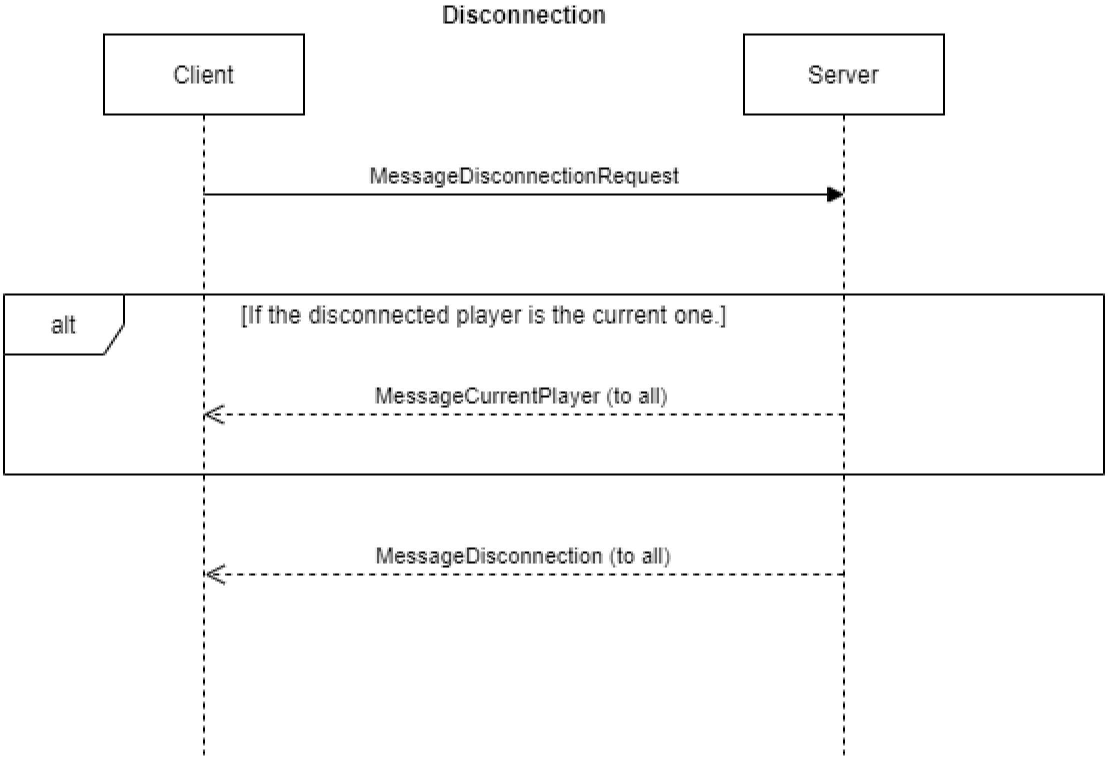
4. Ping Pong

A “Ping” message is sent each 15 seconds by the server. If the client does not reply to this message, it will be considered disconnected. The “Ping” message sent by the client follows the same rule.



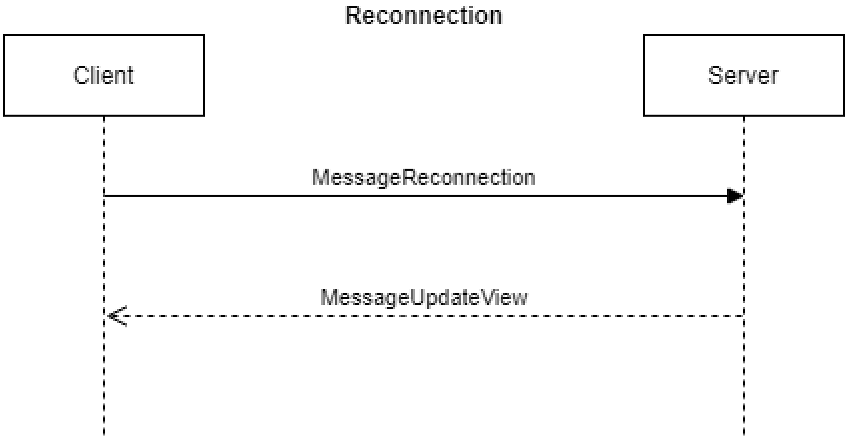


5. Disconnection


Disconnection
Client
Server
MessageDisconnectionRequest
alt
[If the disconnected player is the current one.]
MessageCurrentPlayer (to all)
MessageDisconnection (to all)

Note: when all the players are disconnected the game is over. If the current game is a single player match, the player may reconnect to the game within five minutes.

6. Reconnection



7. XML Message Structure

1.CONNECTION MESSAGE

MESSAGE_CONNECTION

```
<Message>
        <body>Connection request.</body>
        <messageType>CONNECTION</messageType>
        <player> nickname </player>
        <gameHost> true/false </gameHost>
        <playersNumber> number of players </playersNumber>
</Message>
```

MESSAGE_DISCONNECTION

```
<Message>
        <body>Disconnection request.</body>
        <messageType>DISCONNECTION</messageType>
        <player> nickname </player>
</Message>
```

MESSAGE_RECONNECTION

```
<Message>
        <body>Reconnection request.</body>
        <messageType>RECONNECTION</messageType>
        <player> nickname </player>
</Message>
```

2.ACTION MESSAGE

UPDATE_LEADER_CARDS

<**Message**>
<**body**>Leader cards initialization managing.</**body**>
<**messageType**>UPDATE_LEADER_CARDS</**messageType**>
<**leaderStatus**> *position(n):PlayerAction(n)* </**leaderStatus**>
<**leaderCards**> *position(1):ID(1):position(n):ID(n)* </**leaderCards**>
</**Message**>

RESOURCE

<**Message**>
<**body**>Selection of resources during initialization.</**body**>
<**messageType**>RESOURCE</messageType>
<**resources**> *shelf_number(1):resource(1):shelf_number(n):resource(n)*
</**resources**>
</**Message**>

BUY_CARD

<**Message**>
<**body**>Buy card managing.</**body**>
<**messageType**>BUY_CARD</**messageType**>
<**color**> *CardColor*</**color**>
<**level**> *level* </**level**>
<**slot**> *number of slot* </**slot**>
<ID> *ID of the card* </**ID>**
<**warehouse**> *shelf number(1):quantity(1):shelf number(n):quantity(n)*
</**warehouse**>
<**strongBox**> *resource(1):quantity(1):resource(n):quantity(n)* </**strongBox**>
</**Message**>

DO_PRODUCTION

```
<Message>
    <body>Production managing.</body>
    <messageType>DO_PRODUCTION</messageType>
    <personalProduction> true/false </personalProduction>
    <developmentCards> slot(1):ID(1):slot(n):ID(n) </developmentCards>
    <leaderCards> position(1):ID(1):position(n):ID(n) </leaderCards>
    <chosenProducts> resource(1):quantity(1):resource(n):quantity(n)
    </chosenProducts>
    <chosenMaterials> resource(1):quantity(1):resource(n):quantity(n)
    </chosenMaterials>
    <warehouse> shelf number(1):quantity(1):shelf number(n):quantity(n)
    </warehouse>
    <strongBox> resource(1):quantity(1):resource(n):quantity(n) </strongBox>
</Message>
```

LEADER_ACTION

```
<Message>
    <body>Leader card managing.</body>
    <messageType>LEADER_ACTION</messageType>
    <leaderCards> position(1):ID(1) </leaderCards>
    <action> PlayerAction </action>
</Message>
```

SWAP

```
<Message>
    <body>Swapping two shelves of the warehouse.</body>
    <messageType>SWAP</messageType>
    <source> shelf number </source>
    <target> shelf number </target>
</Message>
```

MARKET_SELECTION

```
<Message>
    <body>Selection of a row or a column from the market.</body>
    <messageType>MARKET_SELECTION</messageType>
    <tray> TraySelection </tray>
    <number> number </number>
</Message>
```

ACTION_MARBLE

```
<Message>
        <body>Marbles managing.</body>
        <messageType>ACTION_MARBLE</messageType>
        <marblesActions> Marble(n):PlayerAction(n):Shelf(n)</marblesActions>
</Message>
```

3.REPLY MESSAGE

MESSAGE_REPLY

```
<Message>
        <body>Reply message.</body>
        <messageType>REPLY</messageType>
        <player> nickname </player>
        <correct> true/false </correct>
</Message>
```

4.GAME STATUS MESSAGE

MESSAGE_CURRENT_PLAYER

```
<Message>
        <body>no body</body>
        <messageType>GAME_STATUS</messageType>
        <player> nickname </player>
        <correct> true/false </correct>
        <state>TURN_SELECTION</state>
        <turns> TurnType(1):TurnType(n) </turns>
</Message>
```

MESSAGE_END_GAME

```
<Message>
        <body>Winner message.</body>
        <messageType>END_GAME</messageType>
        <winner> nickname </winner>
        <points> nickname(1):points(1):nickname(n):points(n)  </points>
</Message>
```

5.UPDATE MESSAGE

MESSAGE_UPDATE_BOXES

```
<Message>
        <body>boxes update</body>
        <messageType>BOX_UPDATE</messageType>
        <player> nickname </player>
        <warehouse> shelf number(1):quantity(1):shelf number(n):quantity(n)
        </warehouse>
        <strongbox> resource(1):quantity(1):resource(n):quantity(n)  </strongbox>
</Message>
```

MESSAGE_UPDATE_CARD_SLOTS

```
<Message>
    <body>Update card slots</body>
    <messageType>SLOTS_UPDATE</messageType>
    <player> nickname </player>
    <devCards>> slot(1):ID(1):slot(n):ID(n) </devCards>
</Message>
```

MESSAGE_UPDATE_DECKS

```
<Message>
    <body>Update common decks</body>
    <messageType>DECKS_UPDATE</messageType>
    <devCards>> position(1):ID(1):position(n):ID(n)</devCards>
</Message>
```

MESSAGE_UPDATE_FAITHTRACK

```
<Message>
    <body>Update player’s FaithTrack</body>
    <messageType>FAITH_UPDATE</messageType>
    <faith> nickname(1):points(1):nickname(n):points(n)  </faith>
    <player1Sections> ItemStatus(1):ItemStatus(n) </player1Section>
    <player2Sections> ItemStatus(1):ItemStatus(n) </player2Sections>
    <LorenzoFaith> points </LorenzoFaith>
    <LorenzoSections> ItemStatus(1):ItemStatus(n) </LorenzoSections>
</Message>
```

MESSAGE_UPDATE_LORENZO

```
<Message>
    <body>Update top token</body>
    <messageType>TOKEN_UPDATE</messageType>
    <token> ID </token>
</Message>
```

MESSAGE_UPDATE_MARKET

```
<Message>
    <body>Update Market</body>
    <messageType>MARKET_UPDATE</messageType>
    <market> Marble(1):Marble(n) </market>
</Message>
```

MESSAGE_SELECTED_MARBLES

```
<Message>
    <body>Marble selected by the user</body>
    <messageType>SELECTED_MARBLES</messageType>
    <player> nickname </player>
    <marbles> Marble(1):Marble(n) </marbles>
    <candidates> Marble(1):Marble(n) </candidates>
</Message>
```