Communication protocol

Login:

```
+----------+                          +----------+
|  Client  |                          |  Server  |
+----------+                          +----------+
     |                                     |
     |            Login request            |
     |   ------------------------------->  |
     |                                     |Nickname verification OK
     |            Match started            |
     |   <------------------------------   |
     |                                     |
-------------------------------------------------- OR
     |                                     |Nickname verification KO
     |             Error text              |
     |   <------------------------------   |
     |                                     |
```
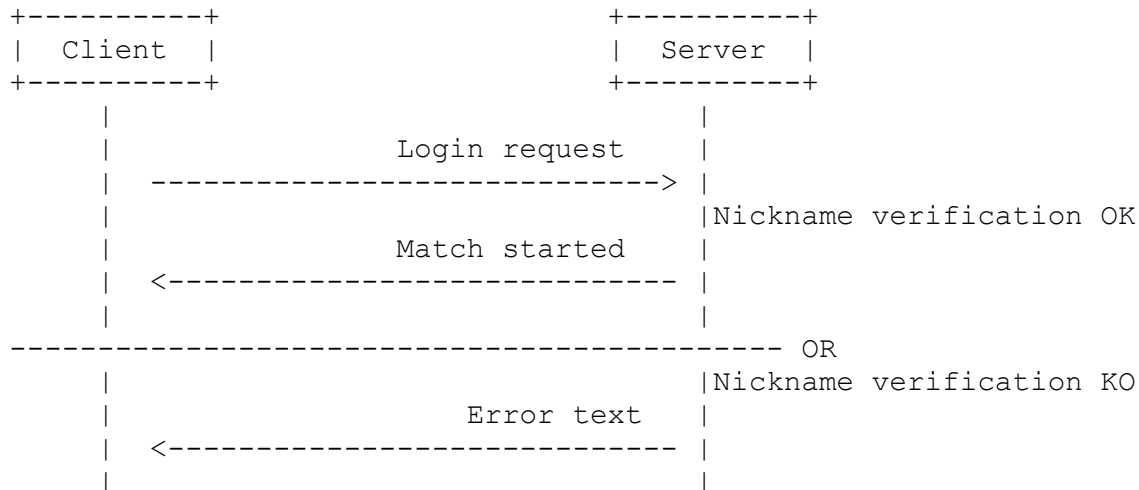
Match start:
Message: [board pre-set, client description VM, other players description VM]
where all "* VM" = [nickname, colour, wallet] (where VM stands for View Model and means it's a censored representation of the data of all the useful info for the views)
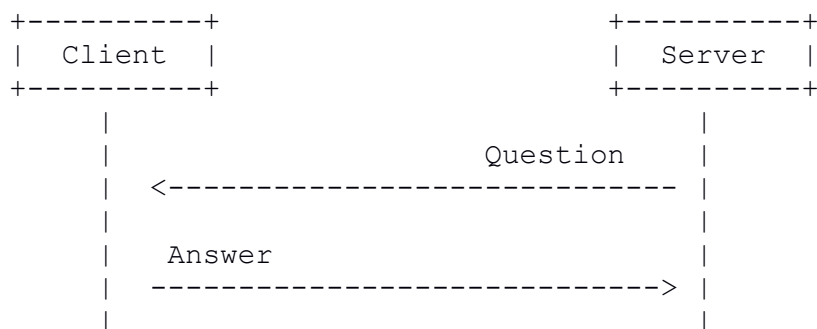
Error text:
Message: "Nickname not available, change it and try again"

A question Event happens when there is a player's turn, when a player can play a powerup, … Every time the player can do something, the server asks him if he wants and what he wants to do.

Question communication:

```
+----------+                          +----------+
|  Client  |                          |  Server  |
+----------+                          +----------+
     |                                     |
     |              Question               |
     |   <------------------------------   |
     |                                     |
     |      Answer                         |
     |   ------------------------------->  |
     |                                     |
```

Question:
Message: ["name", payload, "flow id", "type"]
where: payload = ["text", available options, skippable]
        flow id = string variable useful for parallel question/answer flows
        type = QUESTION

Answer:
Message: ["name", payload, "flow id", "type"]
where: payload = ["text", index of answer, skippable]
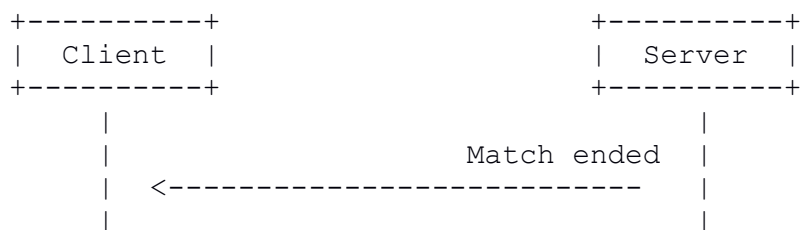   flow id = integer variable useful for parallel question/answer flows
        type = ANSWER

Every communication that is not a question or an answer, is an event. The server tells the clients all the info they need to know to represent the gameplay.

Match ended Event:

For every player we send:

```
+----------+                          +----------+
|  Client  |                          |  Server  |
+----------+                          +----------+
     |                                     |
     |                    Match ended      |
     |   <------------------------------   |
     |                                     |
```
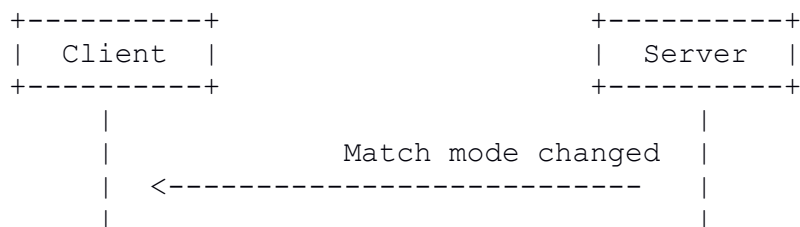
Match ended:
Message: [rankings]
where "rankings" is a map of position and list of players in that position

---

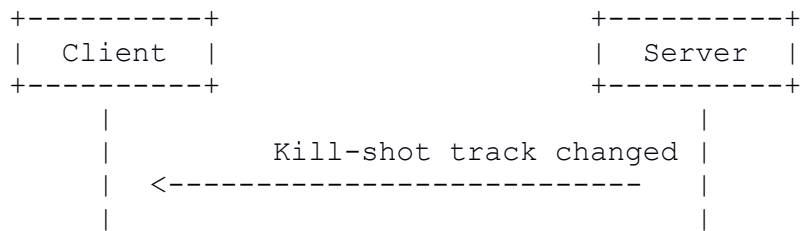Match mode changed Event:

For every player we send:

```
+----------+                          +----------+
|  Client  |                          |  Server  |
+----------+                          +----------+
     |                                     |
     |              Match mode changed     |
     |   <------------------------------   |
     |                                     |
```

Match mode changed:
Message: ["match mode"]

Kill-shot track changed Event:

For every player we send:

```
+----------+                    +----------+
|  Client  |                    |  Server  |
+----------+                    +----------+
     |                               |
     |           Kill-shot track changed |
     |   <--------------------------    |
     |                               |
```
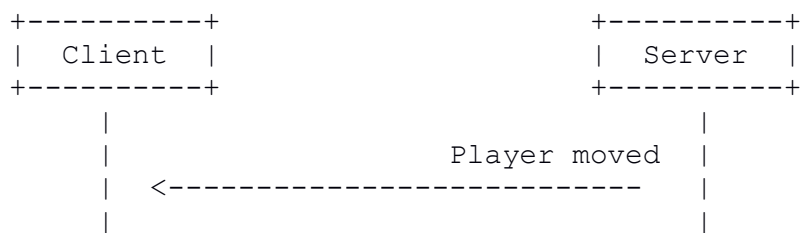
Kill-shot track changed:
Message: [kill-shots]
Where "kill-shots" is a list of tuples of player colours and booleans

---

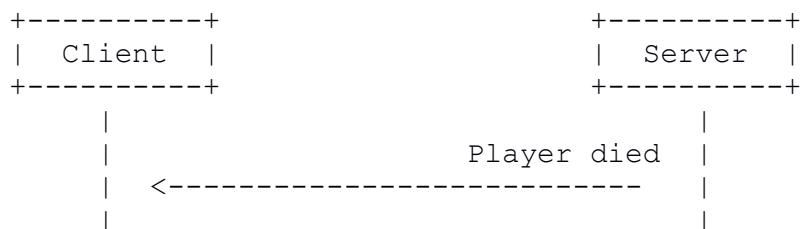## Player moved Event, Player teleported Event:

For every player we send:

```
+----------+                    +----------+
|  Client  |                    |  Server  |
+----------+                    +----------+
     |                               |
     |            Player moved  |
     |   <--------------------------    |
     |                               |
```

Player moved:
Message: [player, x, y]

Player died Event, Player reborn Event, Player board flipped Event:

For every player we send:

```
+----------+                    +----------+
|  Client  |                    |  Server  |
+----------+                    +----------+
     |                               |
     |             Player died  |
     |   <--------------------------    |
     |                               |
```

Player died:
Message: [player]

Player's wallet changed Event:

For every player we send:

```
+----------+                        +----------+
|  Client  |                        |  Server  |
+----------+                        +----------+
     |                                   |
     |              Wallet changed       |
     |  <----------------------------    |
     |                                   |
```
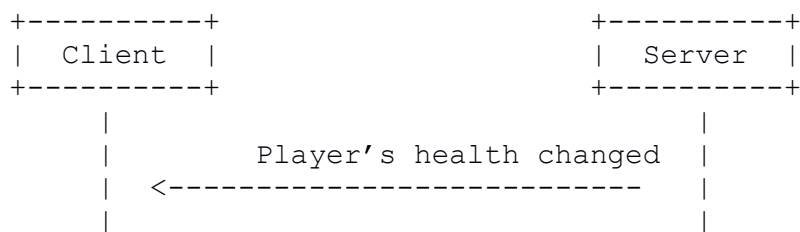
Wallet changed:
Message: [player, wallet]
where Wallet = [[loaded weapons, …], [unloaded weapons, …], [ammo cubes, …], [powerups, …]]

Player's health changed Event:

For every player we send:

```
+----------+                        +----------+
|  Client  |                        |  Server  |
+----------+                        +----------+
     |                                   |
     |         Player's health changed   |
     |  <----------------------------    |
     |                                   |
```

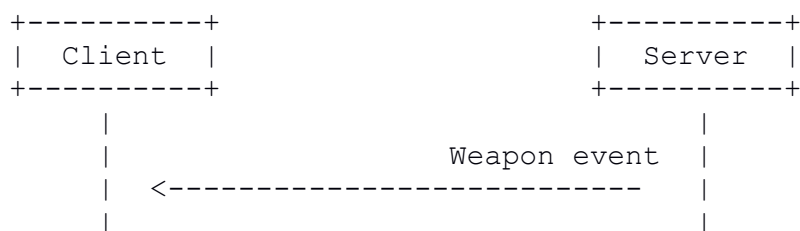Player's health changed:
Message: [player, damages, marks, skulls]

Weapon reloaded Event, Weapon unloaded Event:

For every player we send:

```
+----------+                        +----------+
|  Client  |                        |  Server  |
+----------+                        +----------+
     |                                   |
     |              Weapon event         |
     |  <----------------------------    |
     |                                   |
```
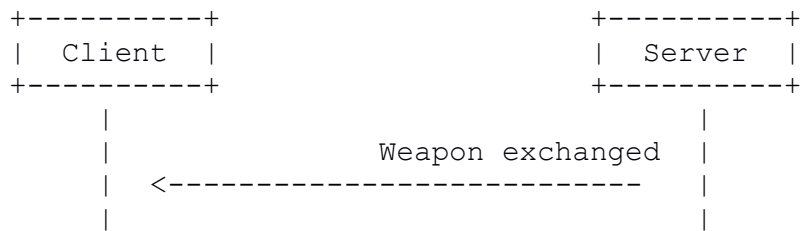
Weapon event:
Message: [player, weapon]
Where "Player" is the owner of the weapon

Weapon picked Event, Weapon dropped Event:

For every player we send:

```
+----------+                     +----------+
|  Client  |                     |  Server  |
+----------+                     +----------+
     |                                |
     |              Weapon exchanged  |
     |   <--------------------------  |
     |                                |
```
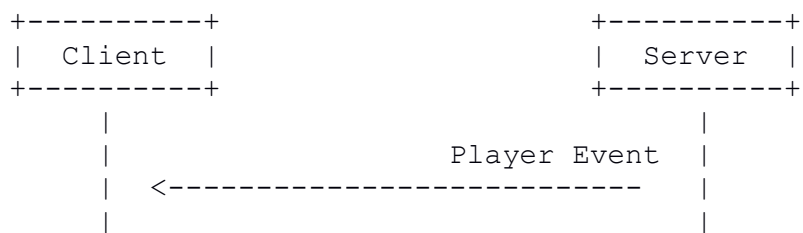
Weapon exchanged:
Message: [player, weapon, x, y]
Where "player" is the owner of the weapon

Player disconnected Event, Player reconnected Event:

For every player we send:

```
+----------+                     +----------+
|  Client  |                     |  Server  |
+----------+                     +----------+
     |                                |
     |                  Player Event  |
     |   <--------------------------  |
     |                                |
```

Weapon exchanged:
Message: [player]