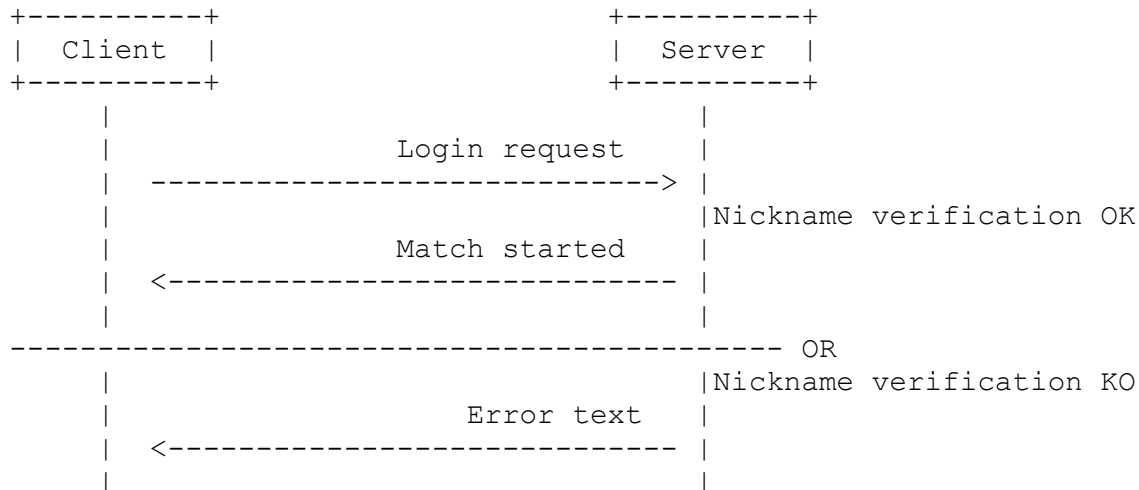


Communication protocol

Login:



Match start:

Message: [board pre-set, client description VM, other players description VM]

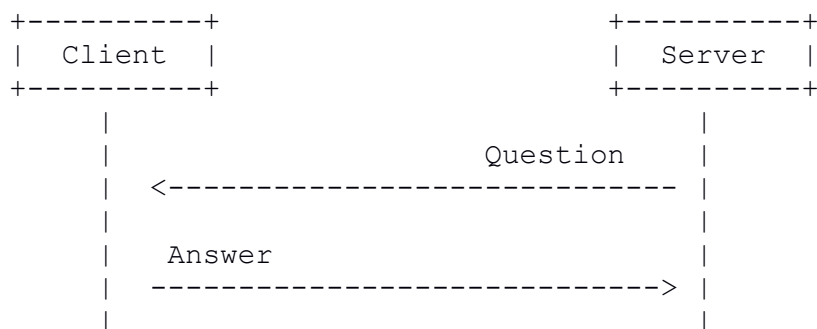
where all “* VM” = [nickname, colour, wallet] (where VM stands for View Model and means it’s a censored representation of the data of all the useful info for the views)

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:



Question:

Message: ["name", payload, "flow id", "type"]

where: payload = [question]

flow id = string variable useful for parallel question/answer flows

type = QUESTION

where: question = ["text", available options, skippable]

Answer:

Message: ["name", payload, "flow id", "type"]

where: payload = [answer]

flow id = string variable useful for parallel question/answer flows

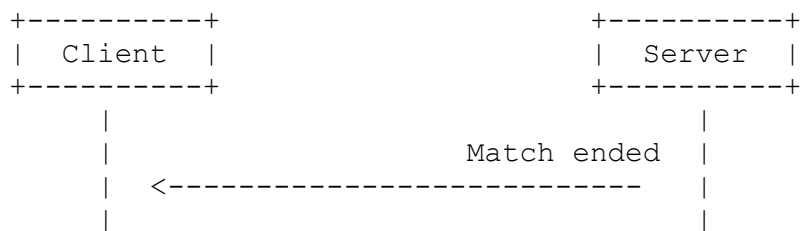
type = ANSWER

where: answer = [choice/null]

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:



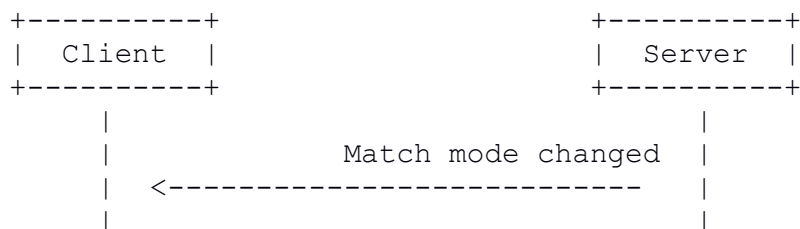
Match ended:

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:

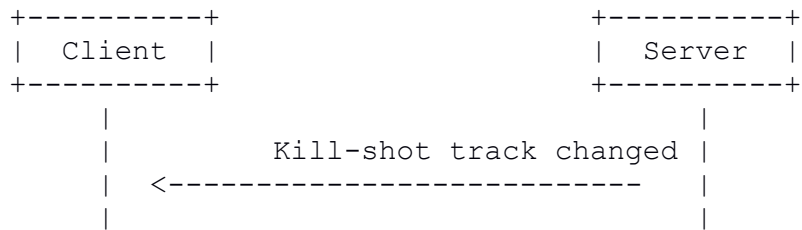


Match mode changed:

Message: ["match mode"]

Kill-shot track changed Event:

For every player we send:



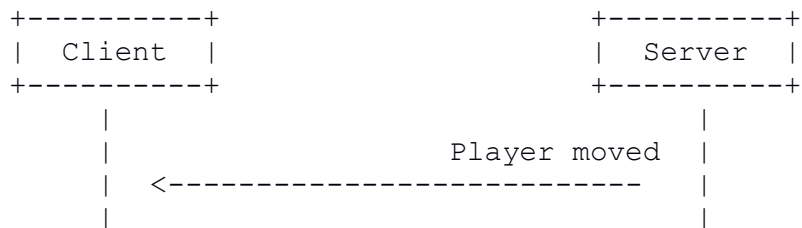
Kill-shot track changed:

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:

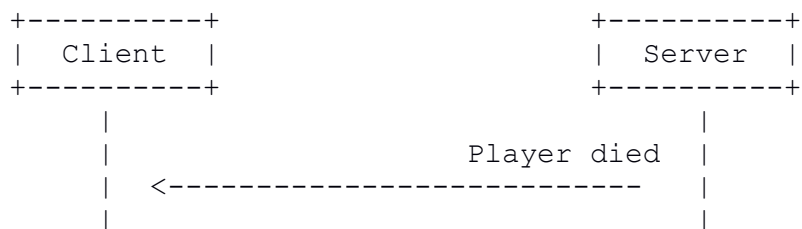


Player moved:

Message: [player, x, y]

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

For every player we send:

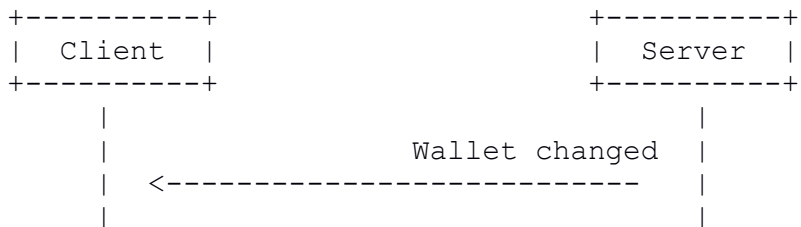


Player died:

Message: [player]

Player's wallet changed Event:

For every player we send:



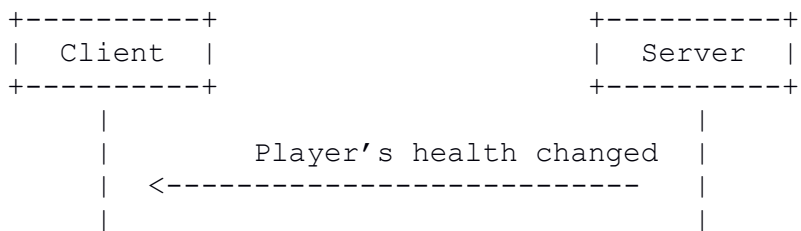
Wallet changed:

Message: [player, wallet]

where Wallet = [[loaded weapons, ...], [unloaded weapons, ...], [ammo cubes, ...], [powerups, ...]]

Player's health changed Event:

For every player we send:

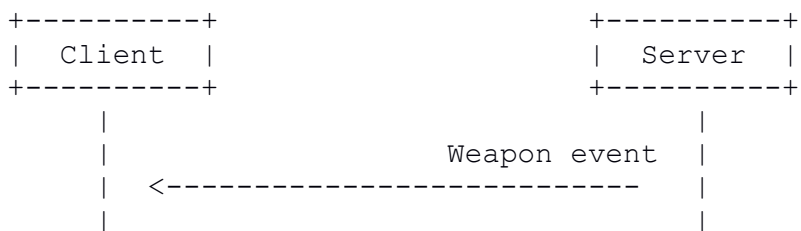


Player's health changed:

Message: [player, damages, marks, skulls]

Weapon reloaded Event, Weapon unloaded Event:

For every player we send:



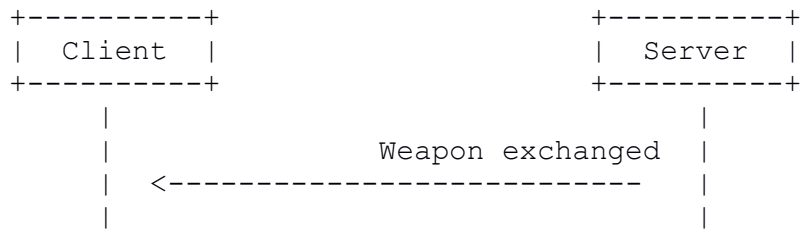
Weapon event:

Message: [player, weapon]

Where "Player" is the owner of the weapon

Weapon picked Event, Weapon dropped Event:

For every player we send:



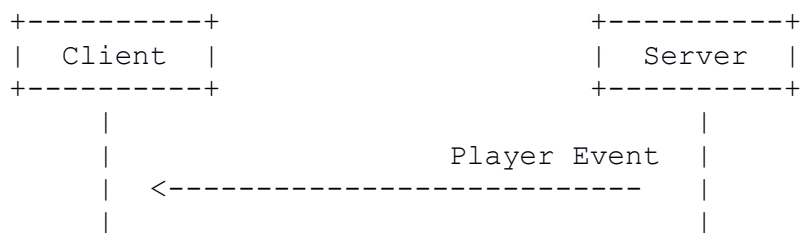
Weapon exchanged:

Message: [player, weapon, x, y]

Where “player” is the owner of the weapon

Player disconnected Event, Player reconnected Event:

For every player we send:



Weapon exchanged:

Message: [player]