PROVA FINALE DI INGEGNERIA DEL SOFTWARE

Documentation of the client-server communication layer

v. 1.0

Daniele Chiappalupi, Stefano Formicola, Elena Iannucci

Last Update – May 14th 2019

1. About this document

The following is a description of how the networking and communication layers cooperate in *Adrenaline*.

Even though it may seem a full description of the implemented and designed functionalities of the application, at the moment of writing of this document the communication between clients and the server is subject to changes that may not respect the messages described below.

We kindly ask the reader to check for the updated version on our GitHub repository.

2. Abstraction Layering

When designing the core of the application, the multiplayer functionality, we adopted the decision to abstract decomposing into different layers the communication between the interested parts in order to have cooperating protocols each one independent.

This decision allowed us to write two separate networking layers for RMI and Socket connections but working on the same type of data exchanged.

For example, when the server decides to notify a specific event to all of his listening clients, it will send the same message through different connections managed by different network handlers that will forward the message to the unique communication layer of the application.

3. Client-Server Communication

Now that an abstract introduction has been made, let's dive into the technical details of the communication between client and server: we will split the communication process into different pieces, in order to better focus on the distinct topics and make the document more readable.

Every communication phase starts with a `ping` signal sent by the Server followed by a pong reply signal transmitted by the Client to initialize the message exchange between the two parts.

a. Log-in

After the `ping-pong` messages, a `login` and `username` request will be received by the Client, that will answer with a message containing its desired username: the server will check if it is available, and if it isn't, another username will be asked. This conversation will be repeated until a valid username is provided.

Here is the list of the messages exchanged during this phase:

- Messages – server side:
  - `log-in`;
  - `insert-username`;
  - `username-ok`;
- Replies – client side:
  - `username`;

This communication is shown in the picture below.

```
+-----------+               +-----------+
|  Client   |               |   Server  |
+-----------+               +-----------+
      |            ping            |
      |<---------------------------|
      |                            |
      |            pong            |
      |--------------------------->|
      |                            |
      |           log-in           |
      |<---------------------------|
      |                            |
      |       insert-username      |
      |<---------------------------| <---     |-------------------------|
      |                            |     \--- |                         |
      |                            |          |          while          |
      |                            |          |  username-not-available |
      |          username          |     ---> |                         |
      |--------------------------->| ---/     |-------------------------|
      |                            |
      |        username-ok         |
      |<---------------------------|
```

b. *Character Choice*

In this phase, a list of available `characters` will be provided to the Client from the Server. The former will choose one and communicate its choice to the latter.

Here is the list of the messages exchanged during this phase:

- Messages – server side:
  - `pick-a-character`;
  - `available-characters-list`;
    example: `[["Banshee", "blue"], ["Dozer", "grey"]]`
  - `character-ok`;
- Replies – client side:
  - `character-choice`;

This communication is shown in the picture below.

```
+-----------+               +-----------+
|  Client   |               |   Server  |
+-----------+               +-----------+
      |            ping            |
      |<---------------------------|
      |                            |
      |            pong            |
      |--------------------------->|
      |                            |
      |      pick-a-character      |
      |<---------------------------|
      |                            |
      |  available-characters-list |
      |<---------------------------|
      |                            |
      |      character-choice      |
      |--------------------------->|
      |                            |
      |        character-ok        |
      |<---------------------------|
```

c. *Action-Hero Comment Choice*

In this phase, an `action hero comment` will be chosen from the Client and sent to the server.

Here is the list of the messages exchanged during this phase:

- Messages – server side:
  - `insert-hero-comment`;
  - `hero-comment-ok`;
- Replies – client side:
  - `hero-comment`;

This communication is shown in the picture below.

```
+-----------+                  +-----------+
|  Client   |                  |   Server  |
+-----------+                  +-----------+
      |            ping              |
      |<-----------------------------|
      |                              |
      |            pong              |
      |----------------------------->|
      |                              |
      |     insert-hero-comment      |
      |<-----------------------------|
      |                              |
      |        hero-comment          |
      |----------------------------->|
      |                              |
      |       hero-comment-ok        |
      |<-----------------------------|
```

d. *New Game Initialization*

In this stage, all of the game settings, such as `map_conf` or `skull_num` will be decided from the Client, so that a new Game can begin.

Here is the list of the messages exchanged during this phase:

- Messages – server side:
  - `select-the-map-type`;
  - `map-types-list`;
    example: [“conf_1”, “conf_2”, “conf_3”, “conf_4”]
  - `map-type-ok`;
  - `select-the-skull-num`;
  - `possible-skull-numbers`;
    example: [“5”, “6”, “7”, “8”]
  - `skull-num-ok`;
  - `select-the-first-player`;
  - `players-list`;
    example: [“username1”, “username2”, “username3”, “username4”]
  - `first-player-ok`;

- Replies – client side:
  - `map-type`;
  - `skull-num`;
  - `player`;

This communication is shown in the picture below.

```
+-----------+                 +-----------+
|  Client   |                 |   Server  |
+-----------+                 +-----------+
      |            ping             |
      |<----------------------------|
      |                             |
      |            pong             |
      |---------------------------->|
      |                             |
      |     select-the-map-type     |
      |<----------------------------|
      |                             |
      |       map-types-list        |
      |<----------------------------|
      |                             |
      |          map-type           |
      |---------------------------->|
      |                             |
      |         map-type-ok         |
      |<----------------------------|
      |                             |
      |    select-the-skull-num     |
      |<----------------------------|
      |                             |
      |   possible-skull-numbers    |
      |<----------------------------|
      |                             |
      |          skull-num          |
      |---------------------------->|
      |                             |
      |         skull-num-ok        |
      |<----------------------------|
      |                             |
      |   select-the-first-player   |
      |<----------------------------|
      |                             |
      |        players-list         |
      |<----------------------------|
      |                             |
      |           player            |
      |---------------------------->|
      |                             |
      |       first-player-ok       |
      |<----------------------------|
```

e. *Spawn*

In this phase, a Player chooses where to spawn.

Here is the list of the messages exchanged during this phase:

- Messages – server side:
  - `pick-a-card`;
  - `powerup-list`;
    example: *`[["Teleporter", "yellow"], ["Tagback Grenade", "blue"]]`*
  - `powerup-ok`;
- Replies – client side:
  - `powerup`;

This communication is shown in the picture below.

```
+-----------+               +-----------+
|  Client   |               |   Server  |
+-----------+               +-----------+
      |            ping            |
      |<---------------------------|
      |                            |
      |            pong            |
      |--------------------------->|
      |                            |
      |         pick-a-card        |
      |<---------------------------|     +--------------+
      |                            |-----| draw-powerup |
      |        powerup-list        |     +--------------+
      |<---------------------------|
      |                            |
      |          powerup           |
      |--------------------------->|
      |                            |     +------------------------+
      |         powerup-ok         |-----| update-player-position |
      |<---------------------------|     +------------------------+
```

f. *Choose an action*

In this stage, a Player chooses an action to do.

Here is the list of the messages exchanged during this phase:

- Messages – server side:
  - `choose-an-action`;
  - `actions-list`;
    example: *`["Run around", "Grab stuff", "Shoot people"]`*
  - `action-ok`;
- Replies – client side:
  - `action`;

This communication is shown in the picture below.

```
+-----------+               +-----------+
|  Client   |               |   Server  |
+-----------+               +-----------+
      |             ping             |
      |<-----------------------------|
      |                              |
      |             pong             |
      |----------------------------->|
      |                              |
      |       choose-an-action       |
      |<-----------------------------|
      |                              |
      |         actions-list         |
      |<-----------------------------|
      |                              |
      |            action            |
      |----------------------------->|
      |                              |
      |           action-ok          |
      |<-----------------------------|
```

g. Run Around

In this action, a Player chooses where to move.

Here is the list of the messages exchanged during this phase:

- Messages – server side:
  - `choose-a-square`;
  - `possible-movements-list`;
    example: `[["0, 1"], ["0, 2"], ["1, 0"], ["1, 1"]]`
  - `movement-ok`;
- Replies – client side:
  - `movement`;

This communication is shown in the picture below.

```
+-----------+               +-----------+
|  Client   |               |   Server  |
+-----------+               +-----------+
      |             ping             |
      |<-----------------------------|
      |                              |
      |             pong             |
      |----------------------------->|
      |                              |
      |        choose-a-square       |
      |<-----------------------------|      +--------------------+
      |                              |------| possible-movements |
      |    possible-movements-list   |      +--------------------+
      |<-----------------------------|
      |                              |
      |           movement           |
      |----------------------------->|
      |                              |
      |          movement-ok         |
      |<-----------------------------|
```

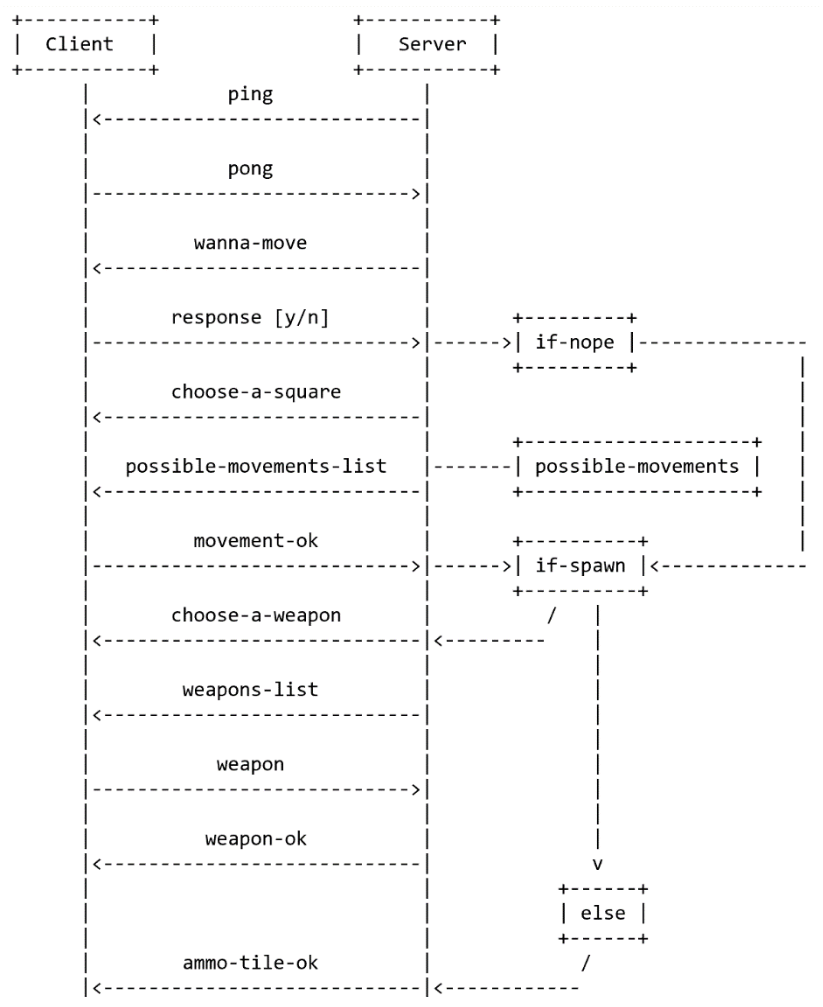
h. *Grab Stuff*

In this action, a Player chooses if he wants to move and what he wants to grab.

Here is the list of the messages exchanged during this phase:

- Messages – server side:
  - `wanna-move`;
  - `choose-a-square`;
  - `possible-movements-list`;
    example: *`[["0, 1"], ["0, 2"], ["1, 0"]]`*
  - `movement-ok`;
  - `choose-a-weapon`;
  - `weapons-list`;
    example: *`["Furnace", "Heatseeker", "Hellion"]`*
  - `weapon-ok`;
  - `ammo-tile-ok`;
- Replies – client side:
  - `response`;
  - `weapon`;

This communication is shown in the picture below.



```
+-----------+                +-----------+
|  Client   |                |   Server  |
+-----------+                +-----------+
      |            ping             |
      |<----------------------------|
      |                             |
      |            pong             |
      |---------------------------->|
      |                             |
      |         wanna-move          |
      |<----------------------------|
      |                             |
      |       response [y/n]        |       +---------+
      |---------------------------->|------>| if-nope |---------------
      |                             |       +---------+              |
      |       choose-a-square       |                                |
      |<----------------------------|                                |
      |                             |       +--------------------+   |
      |   possible-movements-list   |-------| possible-movements |   |
      |<----------------------------|       +--------------------+   |
      |                             |                                |
      |         movement-ok         |       +----------+             |
      |---------------------------->|------>| if-spawn |<-------------
      |                             |       +----------+
      |       choose-a-weapon       |           /    |
      |<----------------------------|<---------      |
      |                             |                |
      |        weapons-list         |                |
      |<----------------------------|                |
      |                             |                |
      |           weapon            |                |
      |---------------------------->|                |
      |                             |                |
      |          weapon-ok          |                |
      |<----------------------------|                v
      |                             |             +------+
      |                             |             | else |
      |                             |             +------+
      |        ammo-tile-ok         |               /
      |<----------------------------|<------------
```

i. Shoot People

In this action, a Player chooses the weapon to use and its effects.

Here is the list of the messages exchanged during this phase:

- Messages – server side:
  - `weapon-to-use`;
  - `weapons-list`;
    example: *`[“Furnace”, “Heatseeker”]`*
  - `choose-effects`;
  - `possible-effects-list`;
    example: *`[[“0, 1, 2”], [“0”], [“1”], [“2”], [“0, 1”], [“0, 2”]]`*
  - `effects-ok`;
- Replies – client side:
  - `weapon`;
  - `effects`;

This communication is shown in the picture below.

```
+-----------+                +-----------+
|  Client   |                |   Server  |
+-----------+                +-----------+
      |            ping            |
      |<---------------------------|
      |                            |
      |            pong            |
      |--------------------------->|
      |                            |
      |        weapon-to-use       |
      |<---------------------------|
      |                            |
      |        weapons-list        |
      |<---------------------------|
      |                            |
      |           weapon           |
      |--------------------------->|
      |                            |
      |       choose-effects       |
      |<---------------------------|
      |                            |
      |    possible-effects-list   |
      |<---------------------------|
      |                            |
      |          effects           |
      |--------------------------->|
      |                            |
      |         effects-ok         |
      |<---------------------------|
```

j. *Use Effect*

In this phase, a Player chooses the damages to make through the effects he is using.

Here is the list of the messages exchanged during this phase:

- Messages – server side:
  - `choose-the-damages`;
  - `damages-list`;
    example: *[[“target: username1”, “damages: 3”, “Cell: [“0, 0”]], [“target: username2”, “damages: 3”, “Cell: [“0, 1”]]]*
  - `damages-ok`;
- Replies – client side:
  - `damages`;

This communication is shown in the picture below.

```
+-----------+                  +-----------+
|  Client   |                  |   Server  |
+-----------+                  +-----------+
      |             ping             |
      |<-----------------------------|
      |                              |
      |             pong             |
      |----------------------------->|
      |                              |
      |       choose-the-damages     |
      |<-----------------------------|
      |                              |
      |         damages-list         |
      |<-----------------------------|
      |                              |
      |            damages           |
      |----------------------------->|
      |                              |
      |          damages-ok          |
      |<-----------------------------|
```