

# Relazione Progetto

Traccia 3 – CHATGAME

Cristian-Daniel Stratu

0000880139

July 2021

(it's in english)

# 1. Introduction

The project is a Python TCP Server-Client chat game where you have to pick questions and answer them, while avoiding the trap ones.

## 2. Description

It's a small chat game that only needs a modification of the JSON file included to diversify the game. It is ready for more ample questions but due to time constraints, it only has a few testing ones.

The two programs are **Server.py** and **Client.py**, either that can be launched normally using "python3 <filename>" or through the executables in the build folders.

It is recommended to launch multiple instances of Client to get the full experience of the project.

The Server will log everything and will handle all of the interactions while the client is a simple chat client that sends messages. It's an almost-black box relationship between the two.

Now, the server is waiting for more clients to enter and input their name. Once there are 2 or more (up to 4) clients connected and ready, the game will start a timer and get players ready. We are in a lobby.

Meanwhile the players can still chat between them.

If the player wants to quit at any time, they can type {quit} or just close the GUI window.

Once the game is now in session, players cannot talk to each other (they can only see their own message).

Now the server hands out the questions for each client and the client will now try to pick a question. If the question is a trap then they are defeated and have to wait until the round is over. Otherwise, now they are presented with multiple choices, of which the player must choose (player being the client). If they get it wrong then it's -1, if they get it right then it's +1.

This continues for 3 minutes. Then the server halts everything and goes to END OF MATCH session. Players can now talk to each other again and will see the results in a moment. After that, they have the choice to quit the game or continue playing.

If they want a rematch, they can do {rematch}.

Other players will get disconnect when game enters the lobby again.

If there aren't enough players (2 or more), then the server ends the game and disconnects everyone.

The client is a simple GUI that only sends and receives messages from the server.

### 3. Used libraries

- tkinter
- threading
- socket
- timer
- random
- json

### 4. Personal Notes

It was a fun project to code for but the time constraint for this was too much for me to do much more. There were a lot of ideas behind it that I wish I had more time to do. Plus, learning Python for these last 3 days was pretty fun. I have enjoyed making this project!