

# INF142 Computer Networks

## Mandatory assignment

Deadline: 11.03.2022



Hi there! Have you played our brand new game? It is called Team Local Tactics and it is available [here](#). So far, we have only managed to develop a local version, but users are asking for more features. They want to be able to play online or versus an AI and, of course, they also want more champions. Our user base is asking for more than we can handle at the moment. However, with your help, we believe we can transform TLT into a distributed application. In doing so, we can finally change its name to Team Network Tactics, which obviously is good branding. So, are you up for the job?

We think TNT should consist of three processes that communicate with each other over the Internet:

- A database for storing champions and stats.
- A server running the logic of the game.
- A client for the players.

This architecture would allow us to patch champions (nerf or buff) while players are still playing the game; and to place our servers close to our users, see Figure 1.

We thank you in advance for your help and politely ask you to send us a minimum viable product (MVP) of TNT no later than 11.03.2022.

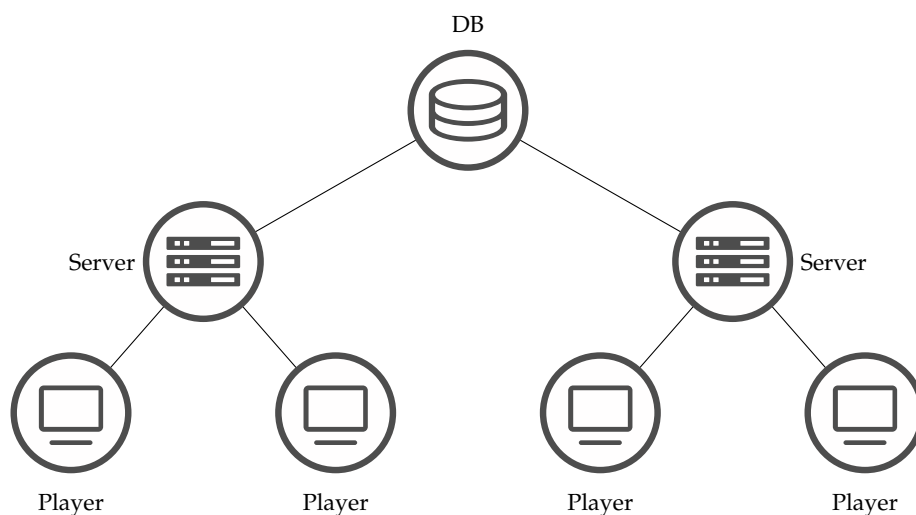


Figure 1: TLT becomes TNT.

You will count on the extremely valuable support of two of our tech leads (your mentor TAs) and, from time to time, you can also request the service of our Head of Silly Names (the course instructor).

## Tasks

As a preliminary step, note that the architecture in Figure 1 can be simplified as in Figure 2.

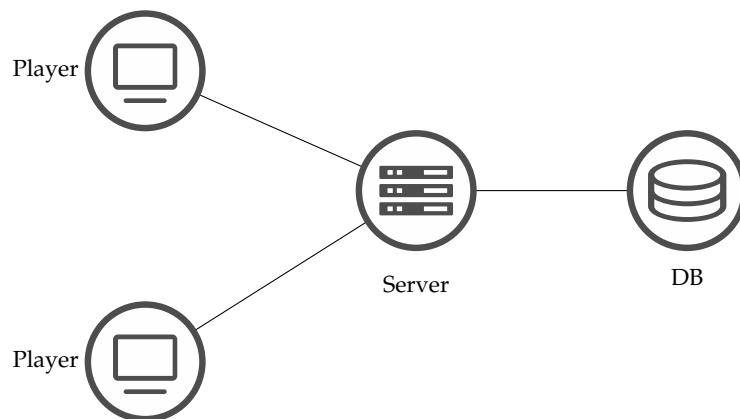


Figure 2: Simplified model.

Your application will be considered a MVP if it fulfills all of the following requirements:

- It consists of at least three Python scripts, one for each of the aforementioned processes.
- Socket programming is used.
- Data associated to champions, match history or other stats must persist in a database or in a file.

The three processes can run, without any isolation, in the same device by using local-host. Note that this is the minimum you have to accomplish. Of course, your tech leads encourage to do more.

## Adding some extras

If you wish, you can help us beyond what we are asking for by delivering more than just the MPV. However, it is of utmost importance that you meet the deadline and that

adding features to TNT will not demand too much from you. These extras should be added only if at least some of your teammates are comfortable with their usage and/or implementations. Do not add more than three. These are some features that you can add or tools that you can use:

- Additional champions.
- An AI so that a user can play against the computer.
- A GUI.
- A simple webpage showing the champions and/or the match history.
- The possibility of running TNT in different devices or user space instances instead of only running through localhost.

For features you can use some of the following tools:

- Requests
- Flask
- Docker (and Docker Compose)
- SQL
- MongoDB

Talk to your tech leads if you want to add something else.

*What follows is sensitive information. Therefore, the role-playing game is unfortunately over. We hope you have fun and learn something from this mandatory assignment.*

## Rules against cheating

- Cross-team cooperation is encouraged, however, rules regarding plagiarism still apply. This means that you should exchange ideas, suggestions, etc. but *not* code. Please note that if we do discover two teams with partially similar code, both teams will fail the assignment.
- If any member of a team cannot explain the code at all, the team will fail the assignment.
- If a teammate stops contributing, the team must report this to the mentors. If this is not reported, the team can risk failing the assignment. The way in which a team distributes the workload is up to the team. However, all its members must be able to explain every line of code. A couple of meetings, discussing who did what, should be enough for every teammate to understand what others did.

## Your grade

The grade for the mandatory assignment will be either pass/fail. If all goes according to plan, you will still receive a letter grade, but an unofficial one so that you may know more accurately how you did on the assignment.

You have to pass two checkpoints before the deadline:

**Week 8** Set up a git repository and invite your mentors.

**Week 9** Create a readme file and implement the communication between sockets in at least two of the scripts.

After you have turned in your assignment, we will schedule a meeting with each team where each teammate will be asked to explain parts of the code. The submission format for your assignment is up to the TAs mentoring your team. The actual grading of the assignment will be done by both your mentors and the lecturer. This assignment is designed with the time frame in mind, therefore extensions will only be granted in exceptional cases, or by reason of sickness, which would require a student to present a certificate of proof from a GP.