

Lab 5: Game Design—LOL 2D

Programming Fundamentals 2

4th May 2021



Goals

- ★ Understanding the architecture of a medium project (above 1500 lines of code).
 - ★ Adding a functionality in an existing project.
 - ★ Working in a large team.
-
- For this laboratory, the whole class is a team. You are encouraged to discuss.
 - The grade will be on your commits and pull-request, so make sure your commits are yours.
 - **Deadline: Thursday 20th May 08:00AM**

Deliverables

1. The code pushed as a pull-request to the following repository: <https://github.com/ptal/lol2D>.
2. A presentation video (one per student) of 3' minutes on FLIPGRID: <https://flipgrid.com/eeb7abc1>
 - Use your @student.uni.lu email to connect.
 - Share your screen and comment your work on this lab.

1 Preliminary

The explanations below are very minimal. We will study this software during the lectures, and you are welcome to ask questions about it on Discord as well. Among the new functionalities, LOL 2D is using JavaFX for the graphics, Java socket for the network, and the Visitor and Builder design patterns for the core functionalities. Also some more advanced (but not hard to understand) Java concepts are used: inner classes, anonymous classes, lambda functions and default interfaces.

2 LOL 2D

In the repository <https://github.com/ptal/lol2D>, there is a minimal working version of a 2D MOBA game¹. Your first task is to *fork* this repository by clicking on the top right fork button, see <https://docs.github.com/en/github/getting-started-with-github/fork-a-repo>. This will “copy” this repository to your Github account, and you can clone the forked repository afterwards.

You should not clone <https://github.com/ptal/lol2D> because you don't have the rights to push on this repository.

Once forked and cloned, you will need three terminal to execute the project:

```
# First compile the project
mvn compile
# In terminal 1: the server
mvn exec:java -Dexec.mainClass="lol.server.LOL2D"
# In terminal 2: the first client
mvn exec:java -Dexec.mainClass="lol.client.Client"
# In terminal 3: the second client
mvn exec:java -Dexec.mainClass="lol.client.Client"
```

A graphical user interface (GUI) should appear. For the same reasons than in laboratory 4, using WSL might not work. You can switch to Ubuntu (install it or use a virtual machine), or an IDE, but, unfortunately, I don't know IDEs well so I might not be able to help you!

Once everything is set up and running, you should either:

- Consult the issues list on <https://github.com/ptal/lol2D/issues>, and post a message to say you are interested by completing this issue. If other students are also interested, you are free to either group together, or do it separately on your respective fork. If you group, one of the student can add a collaborator to its forked repository, so the other student can push code.
- Propose your own issue on <https://github.com/ptal/lol2D/issues>, in this case, you should wait for a message from me to validate your issue.

Each time you finish an issue, you can do a pull request on my repository (see <https://docs.github.com/en/github/collaborating-with-issues-and-pull-requests/about-pull-requests>). Before doing the pull request, you should synchronize your forked repository with mine, and possibly fix the conflicts:

```
git remote add upstream https://github.com/ptal/lol2D.git # Only need to do it once.
git fetch upstream
git rebase upstream/main # replay the changes you made on top of the newest version, you might need to fix conflicts.
```

3 Competitive track

Those interested in the competitive track must register here (you can join anytime): <https://docs.google.com/spreadsheets/d/1KMZx58SoE08g-14usphLaLFnPhKzBDhTpa9PgixOok8/edit?usp=sharing>.

See Laboratory 1 for the submission instructions. This time, the exercises are:

- *514 - Rails* (easy, stack data structure (can use ArrayList)): https://onlinejudge.org/index.php?option=com_onlinejudge&Itemid=8&category=24&page=show_problem&problem=455
- *1595 - Symmetry* (geometry): https://onlinejudge.org/index.php?option=com_onlinejudge&Itemid=8&category=24&page=show_problem&problem=4470
- *612 - DNA Sorting*: https://onlinejudge.org/index.php?option=com_onlinejudge&Itemid=8&category=24&page=show_problem&problem=553

¹https://en.wikipedia.org/wiki/Multiplayer_online_battle_arena

- *11849 - CD (HashSet)*: https://onlinejudge.org/index.php?option=com_onlinejudge&Itemid=8&category=24&page=show_problem&problem=2949
- *10405 - Longest Common Subsequence (use dynamic programming)*: https://onlinejudge.org/index.php?option=com_onlinejudge&Itemid=8&category=24&page=show_problem&problem=1346

Good luck!