SciPy – Preliminary project description
Jakob Heller

Combination (and Visualization) of Logs uploaded to FFlogs.com[1]

Goal (and context)

There exists a tool to collect and upload extensive combat data in the online video game "Final Fantasy XIV". FFlogs.com, the site where these log files are uploaded then allows in-depth analysis of this data. It already offers sufficient tools to filter and visualize data by itself (here is an example log) but there is one shortcoming that I would like to address:

While the website is able to summarize combat data over the whole log file, analyzing multiple files at once is not possible.
I would like to create a tool that (1) lets me indicate what logs (usually separated into days) I want to have summarized, (2) accesses/ scrapes the data (only limited metrics), (3) summarizes it in an appropriate way and (4) visualizes it (preferably somewhat interactively).

Required Steps (4)

(Step 0:)

Generate and upload data. Since there are hundreds of thousands of logs already available (and I regularly upload some myself) this should not be an issue at all.

Step 1:

User input. The user has to provide the logs that should be analyzed in some kind of way – either by their direct links or by specifying dates. Since a fancy user interface is not a priority, this can stay in the console (no extra libraries needed).

Step 2:

Access and scrape data. This step will use scraping libraries like *Requests* and *Beautiful Soup*; *Selenium* might be necessary as well.
Here I need to decide what data I actually want to include in my final visualization – I will probably start with only including the basic metrics "Damage Done" and "Healing Done". This can always be adjusted to include more metrics when the project is at a further state.

Step 3:

Clean up and summarize data. This step will use the appropriate libraries introduced in the course, namely *numpy* and *pandas*.

---

[1] There exists an equivalent site for the game "World of Warcraft" which is probably better known – what I am planning should work for both, but I chose FFlogs since that's what I actually use myself.

Step 4:

Visualize data. This step will use *matplotlib*, but libraries like *plotly* to create a more interactive visualization also seem reasonable.