Part 1 – Analysis

Think about how these networks are different. Analyse the "dimensions" of these networks. What are the relevant attributes (e.g., commits, users, branches, commit size, etc.) of these representations? What other attributes could be relevant in this graph? Write a list of all the attributes your visualization could show.

The relevant attributes of the representations are commits, users, branches, the timeline and paths. Other possibly relevant attributes could be the type of the commits, a title and a legend for explaining the colours.

Are there different roles, i.e., different types of users who might want to achieve different things? Write a list of user roles.

Users who make changes in the projects.
Users who are interested in the project and want to see what has changed over time and how current these changes are.

Think about which tasks a user of your visualization might want to achieve. Write down a list of tasks.

See how many commits are made.
See which users participated to the project.
See who made what commits.
See what has changed and how big these commits are.
What functionality is added or changed.
When new commits are done.

Identify one role that you want to design your visualization for. Prioritize your task and attribute lists based on this role's needs.

We want to design a visualisation for the user who wants to get information about the commits and know who participated to the project. It is important to show the amount of commits and when these commits are made.

Part 2 – Sketching

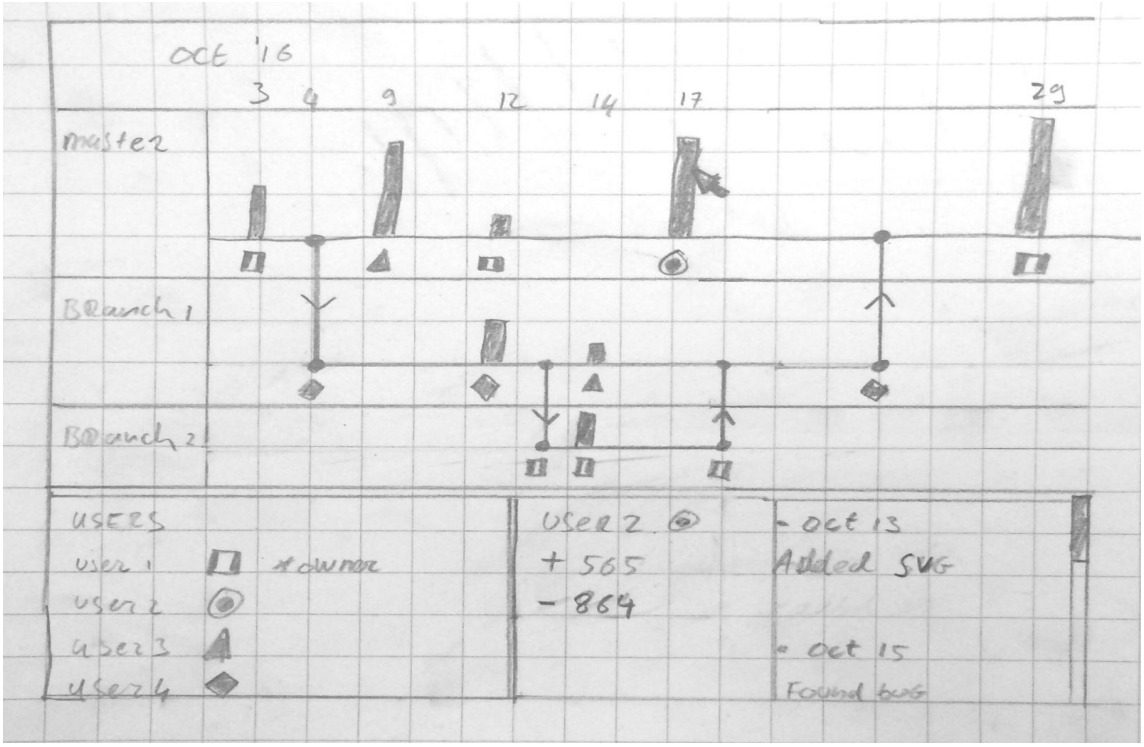

Image 1

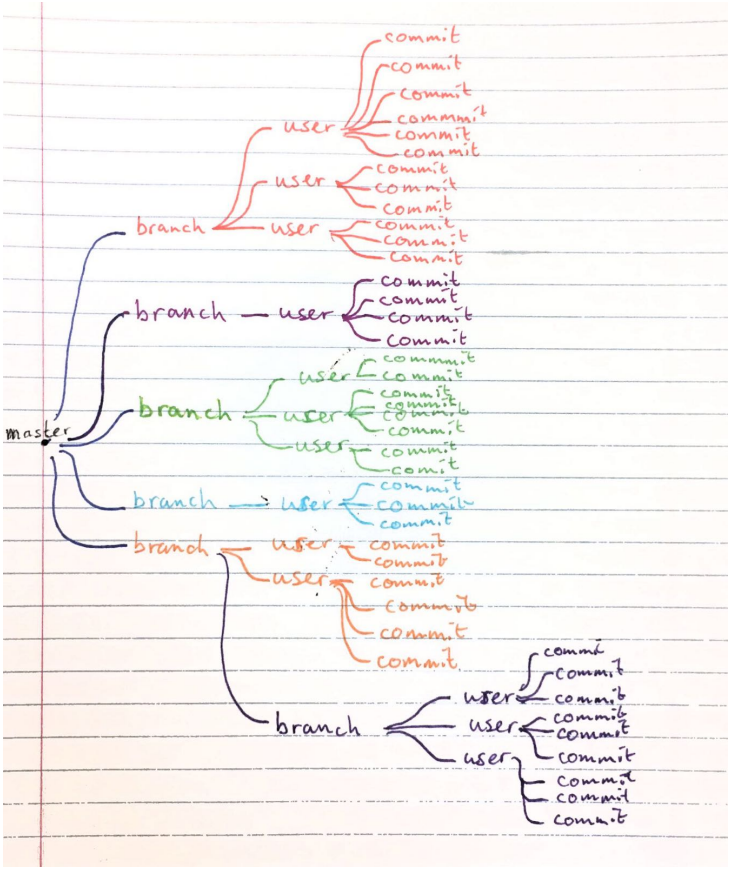

Image 2

Part 3 – Group reflection

Image 1 and image 2 show alternatives for the network graphs of Github. We wanted to remove the nodes of the different commits and didn't want to give the collaborators an own row. Image 1 shows a graph that offers a solution to these aspects. Instead of division by users, it shows a division by branches. The users are indicated with a visual variable. The bar that matches the symbol of the users shows the amount of commits the user has made on that day. The amount and type of commits will come up when clicked on the bar. The second image shows a totally different graph. Instead of hovering with the mouse over the nodes to see the commits, everything is shown immediately in one graph. This gives an overview of all the information that can be found in the network graph.

We considered image 1 to be the best design. This graph offers the necessary information to the user in an understandable way. It gives a better overview of the activity of the different users. Interactivity ensures structured information about the commits. Because of the big amount of information the graph provides, the graph of image 2 would be very big and therefore not very clear to the user anymore.