Yannick, Inge, Nizar, Laurens & Charlotte

Think about how these networks are different. Analyze 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.

Both representations show the existing branches over time. In these branches the commits are represented as nodes. The first representation mostly shows one branch, namely the master branch. In the second representation there are different branches, and you can see which user made which changes. When a commit is hover, the commit message is shown. It is possible to click on a commit to see exactly which changes were made.

Relevant attributes:
- Different colors for different users
- Commits (although sometimes there are a lot by one person in a short time, these might be grouped)
- Hovering the commits shows the commit message
- Being able able to click on the commits to see the changes

Additional attributes:
- Legends to show different branches + users that worked on branch
- Scroll bar to go through time easier
- Button to go to start/ and end
- Being able to zoom in/out
- Add time to commit message

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

User roles:
- Working alone/group
- Using multiple branches/only master
- Management
- Team members
- School project
- Development

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

Task:
- Overview who did what and who is working on which branch
- Being able to see which changes were made over time
- Overview structure of project, which branches exist
- Being able to see previous code

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

People who work in groups, and are interested to see an overview of the process.

Decide on which visual variable to use for which attributes of the visualizations. Consider the strengths and weaknesses of visual variables that were mentioned in Carpendale's article Reading 2 (also briefly discussed in this week's lecture: Process). Use the strongest visual variable for the most important attributes of the data.

Main branch: bold: main timeline, and in middle, never changes row
Long branches: underneath main
Short branches/changes: above main
Different branches: different colors
Different users: small label (initials above/in commit)

Do you think it is necessary to represent every single commit as a separate node? Could you think of ways to aggregate this?

No this is not necessary. You can represent one node per day per branch. When you hover this node, you see which changes are made on this branch on this specific day. When you zoom in you can see all commits on the line.

Do you think that every contributor needs a "row", as on the default network view on github? Could you think of a smarter way to summarize those?

Yes, labels linked to users.

Is a node-link diagram the appropriate representation? Or should you consider alternative graph representations?

The node-link diagram is an appropriate representation.