

## Design 4 – Github Commit Graph

### Part 1 - Analysis

#### Dimensions:

- Commits: when as user has changed the file
- Users: the users that can access the files
- Branches: file for each user
  - Merge: when information from one user is pushed into another
  - Pull: when information from one user is abstracted from another
- Commit size: the amount of changes that have been made to the file
- Date/time: when the certain events have been occurred

#### Roles:

- Those who write/change the files
- Those who can post commits on the files
- Those who can run and read the file but not edit them

#### Tasks of visualisation:

- When something occurred?
- Who made it occur?
- What did occur?
- What was the impact of the occurrence?

#### A clear overview of user participation:

- Made clear who did what, assign colours to the participants
- Kind of commits; remove, add, comment, script approval
- Size of commits

### Part 2 – Sketching

#### Visual variable:

- Circle for commit, in different colour layers
- Colour for kind of comment; the core black for adding, around the core white for removing, topped off with grey for comment. Encapsulated with the user colour
- Size for amount of changes

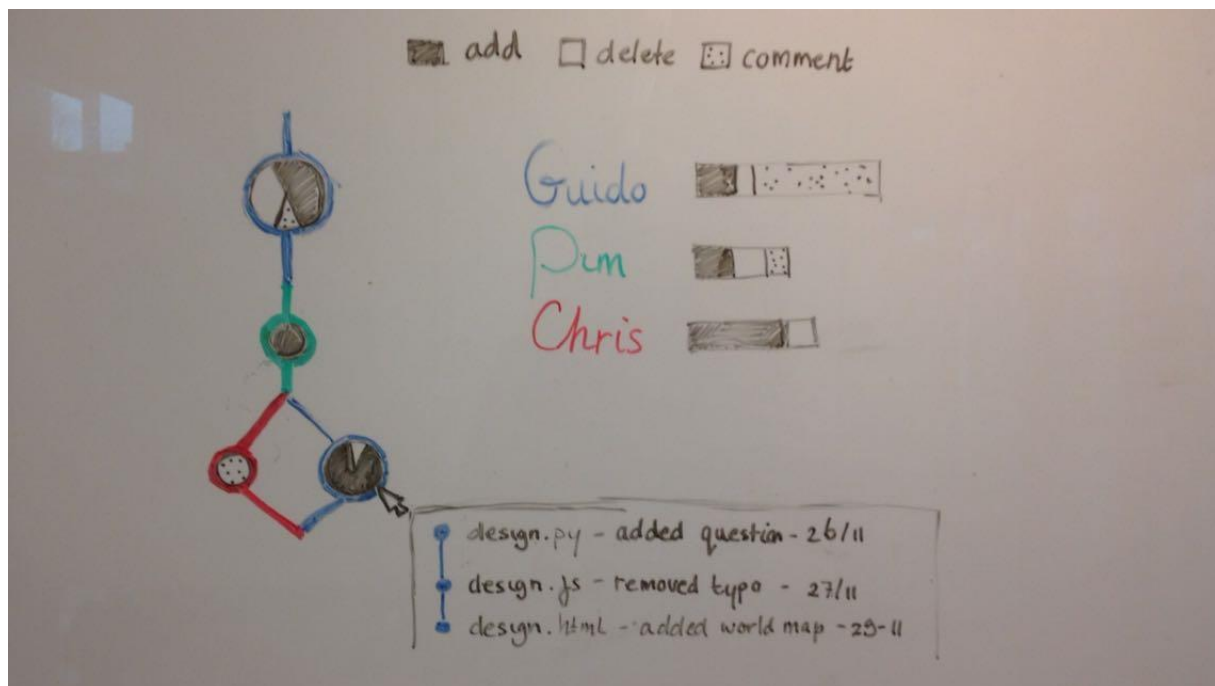
The dark/white/grey filling gives a good overview of what kind of content has been committed. The colours are assigned to different user, in this way the commit can be tracked to a specific user. The size of the circle determines the amount of committing the user has done.

Two styles can be used as a filling method. More traditionally, a pie chart filling can be used, cutting the circle in triangular pieces. The other method makes creates eye like figures, adding the layers over each other, determining with their thickness the amount of commit that has been made. The downside of the eye like figures is that it might look a bit more messy and it is harder to determine the amount of a commit sort, due to the surface properties of the circle.

When hovered: a circle commit expands and show a window with text, that elaborates on the date/time of the commit and in which file changes have been made.

- Each commit is bundled in a pie chartish node, until it is pushed again to other users.
- In our case, the contributors are assigned to the nodes with their specified colours, not to a row.
- A node-link diagram creates an event overview, which is very suitable for the variants at hand. We think this is the far best way to represent the data. An other possibility would be an ebb-flow like chart, flow lines that stack up on each other, differing in thickness, correlated to the amount of committing at a certain point of time.

### Part 3 – Group reflection



This is our visualisation of the github repository data, focusing on the content of the commits that have been made. Bundeling them into one node per branch.

We think this especially well shows the contribution of the users.