

A-Team: Kevin Shiah, Edward Law, Jacky Liao, Risa Newyear-Ramirez

A. Output

Our project will analyze the commit history and Java syntax and produce an animated real-time visualization of the code commit sequence and the syntax correctness of the code committed. Each contributor to the project will be represented by unique coloured dots randomly spread across a 1000 by 500 pixels black background. Under each dot is the contributor's Github ID. To make the animation happen, there will be a 1 pixel object called the "colour ring emitter". The emitter will continuously produce colour rings, and these rings will radically expand outward and eventually fade away. When a contributor commits a change, the colour ring emitter will move the contributor's dot and will continuously emit colour rings around the dot. The colour produced by the emitter is the same as the colour of the dot. If the next commit is done by a different contributor, the emitter will move from the dot of the last contributor to the dot of the new contributor. The emitter will also produce rings along the travel path. The colour of the rings produced along the path will be the gradual transitioning of colours between the two dots. The emitter will travel at a speed of 4 dots per second. We may produce some kind of music that changes intensity over time according to the code quality. When the code quality is good we may produce a more enjoyable sound.

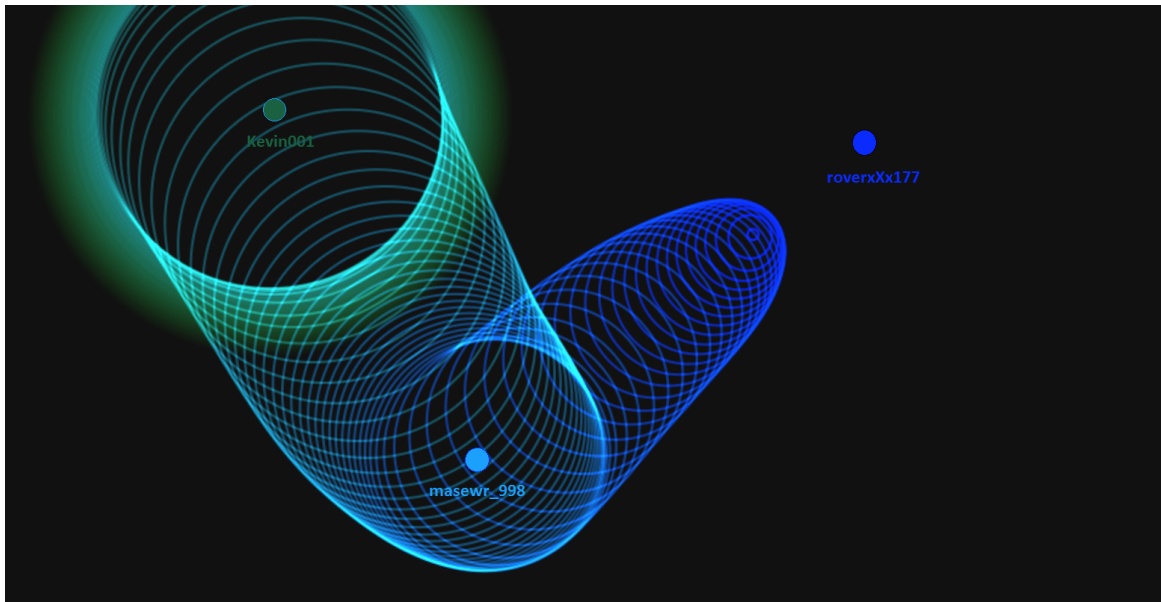
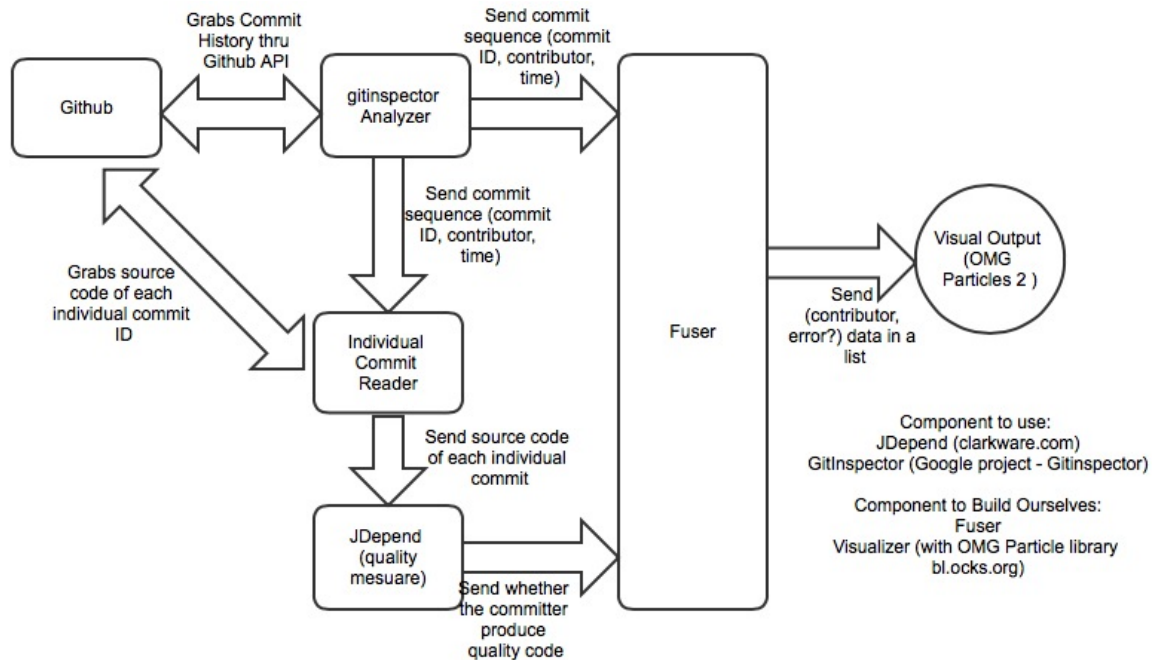


Fig 1: Three different GitHub contributors and their coloured IDs.

B. Architecture



C. Analyzer and Fusion

We will use “gitinspector” to fetch repository data from Github. First, the tool will grab the commit history, and store the information such as contributor, commit ID, and the date for each commit to a text or XML file. With this information, we will make a tool that will be able to locate the actual submitted code via Github API. We will use a static Java analysis tool (JDepend) to determine whether the code has good quality. This will be the consumable for the visualizer to produce the animation. Example of “gitinspector” analyzer result:

```
$ ./gitinspector.py -wTH1 /path/to/some/git/repository
The following historical commit information, by author, was found in the repository:
```

Author	Commits	Insertions	Deletions	% of changes
John Smith	288	7721	4617	39.19
James Johnson	135	8910	2422	35.99
Robert Brown	71	2564	1352	12.44
Michael Davids	134	2943	954	12.38

D. End Result:

From what is provided by the analyzers we can distinguish which committer has the most commits, the code quality, the project size, etc... We would also like to analyze the code quality over time.

E. Code bases

Essentially any Java based project on Github will be a valid input for our project. However, for the sake of making the output animation cool, we will limit the number of contributors to 30 and the number of commits to 400. This will allow enough dots to be spread across the black background. If the colour ring emitter travels 4 dots per second, the entire animation should not be longer than 100 seconds.

<https://github.com/SpongePowered/Sponge> with 20 Contributors and 77 Commits

<https://github.com/lucasr/twoway-view> with 6 Contributors and 311 Commits