

# Visualizing Github Contributions

a course project of Information Visualization

Davidson Zheng

# The data - Github

- web-based Git repository hosting service / source code management platform
- open source projects
- social network-like data of how developers work on repositories

# The data - Github

- Commits (additions and deletions)
- Contributors
- Star
- Fork
- Issues

# Insights of data - Individual

For developers, github is a great portfolio website

- **popularities** (followers, starred)
- **projects** getting involved (repositories)
- **amount** of contributions (total commits)
- **frequency** of contributions (daily/weekly/monthly commits)
- **quality** of code (stars, forks)
- **skill** set (number of languages used)
- etc.

# Insights of data - Organization

- trending **projects** (stars, forks)
- competent **developers** (number of commits, activity, etc.)
- problems (issues)
- etc.

# Getting the data via Github API

Getting the trending projects during the last month

```
curl -G https://api.github.com/search/repositories \
--data-urlencode "q=created:>`date -v-1m '+%Y-%m-%d'`" \
--data-urlencode "sort=stars" \
--data-urlencode "order=desc" \
-H "Accept: application/vnd.github.preview" \
| jq ".items" > trending_project_1m.json
```

# Getting the data via Github API

```
{
  "id": 50603846,
  "name": "parse-server",
  "full_name": "ParsePlatform/parse-server",
  "owner": {
    ...
    "type": "Organization",
  },
  ...
  "stargazers_count": 6432,
  "watchers_count": 6432,
  "language": "JavaScript",
  "has_issues": true,
  "has_downloads": true,
  "has_wiki": true,
  "has_pages": false,
  "forks_count": 1335,
  "mirror_url": null,
  "open_issues_count": 105,
  "forks": 1335,
  "open_issues": 105,
  "watchers": 6432,
  "default_branch": "master",
  "score": 1
},
```

# Filter the data

Format the data to be processed by d3 treemap with custom python script

```
{
  "name": "Top 10 Languages",
  "children": [
    {
      "name": "javascript",
      "children": [
        {
          "repo_id": 28457823,
          "name": "FreeCodeCamp",
          "html_url": "https://github.com/FreeCodeCamp/FreeCodeCamp",
          "children": [...],
          "owner_name": "FreeCodeCamp",
          "stars": 87392,
          "owner_type": "Organization",
          "forks": 3123
        },
        {...},
        {...},
        {...},
        {...},
        {...}
      ]
    },
    {...},
    {...},
    {...},
    {...},
    {...},
    {...},
    {...},
    {...},
    {
      "name": "c++",
      "children": [...],
    }
  ]
}
```

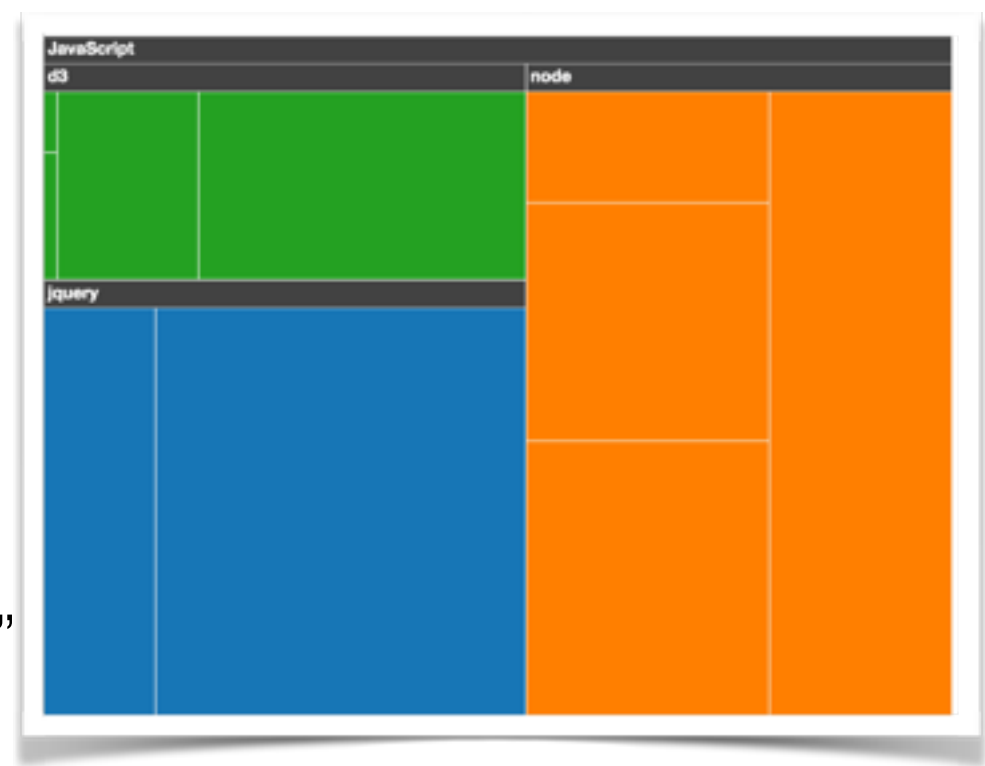


# Visualization using d3 with zoom effect



1st layer - languages

click label  
"Javascript"



2nd layer - projects

click any panel

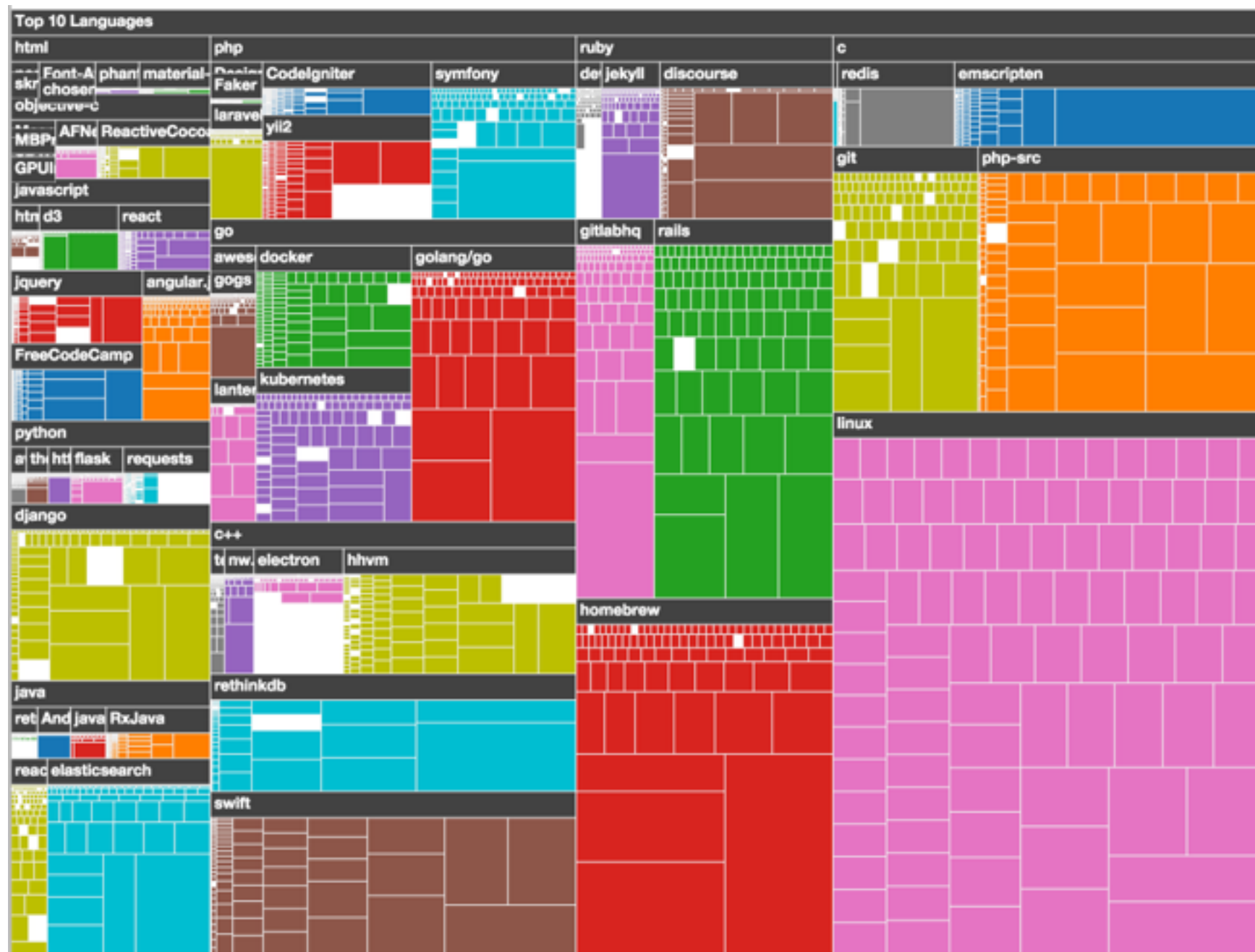


3rd layer - contributors

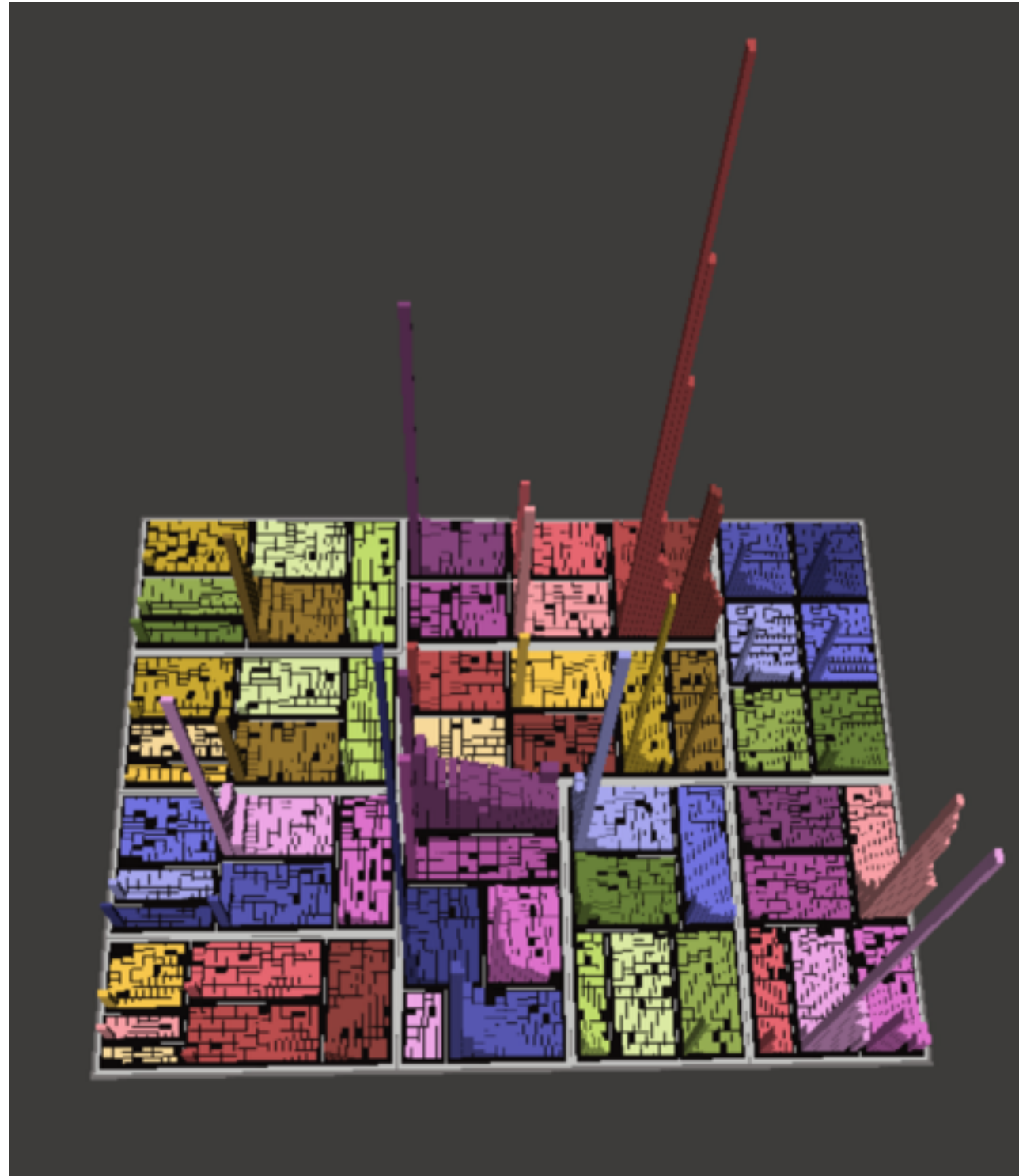
click label "d3" or  
any children panel

# Visualization using d3 with zoom effect

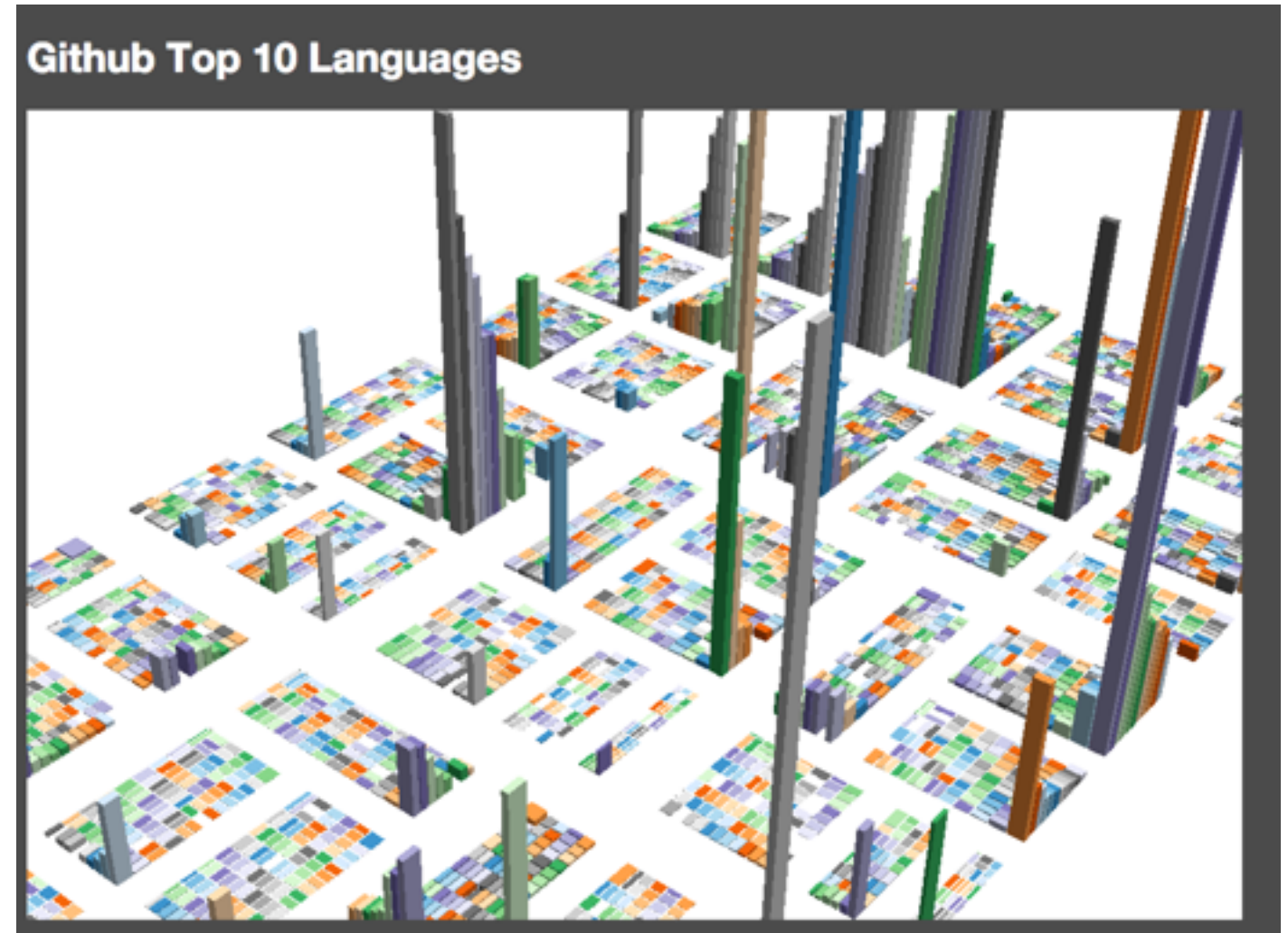
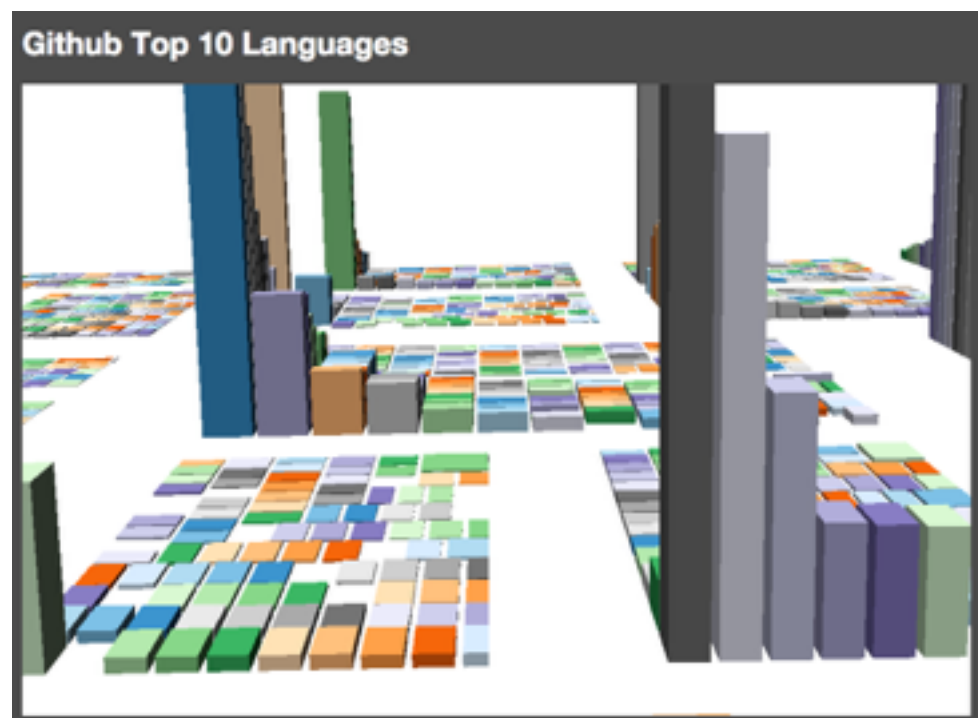
scale up to  $10 \times 6 = 60$  projects with up to  
100 contributors each  $\leq 6000$  child nodes



# Visualization using threejs

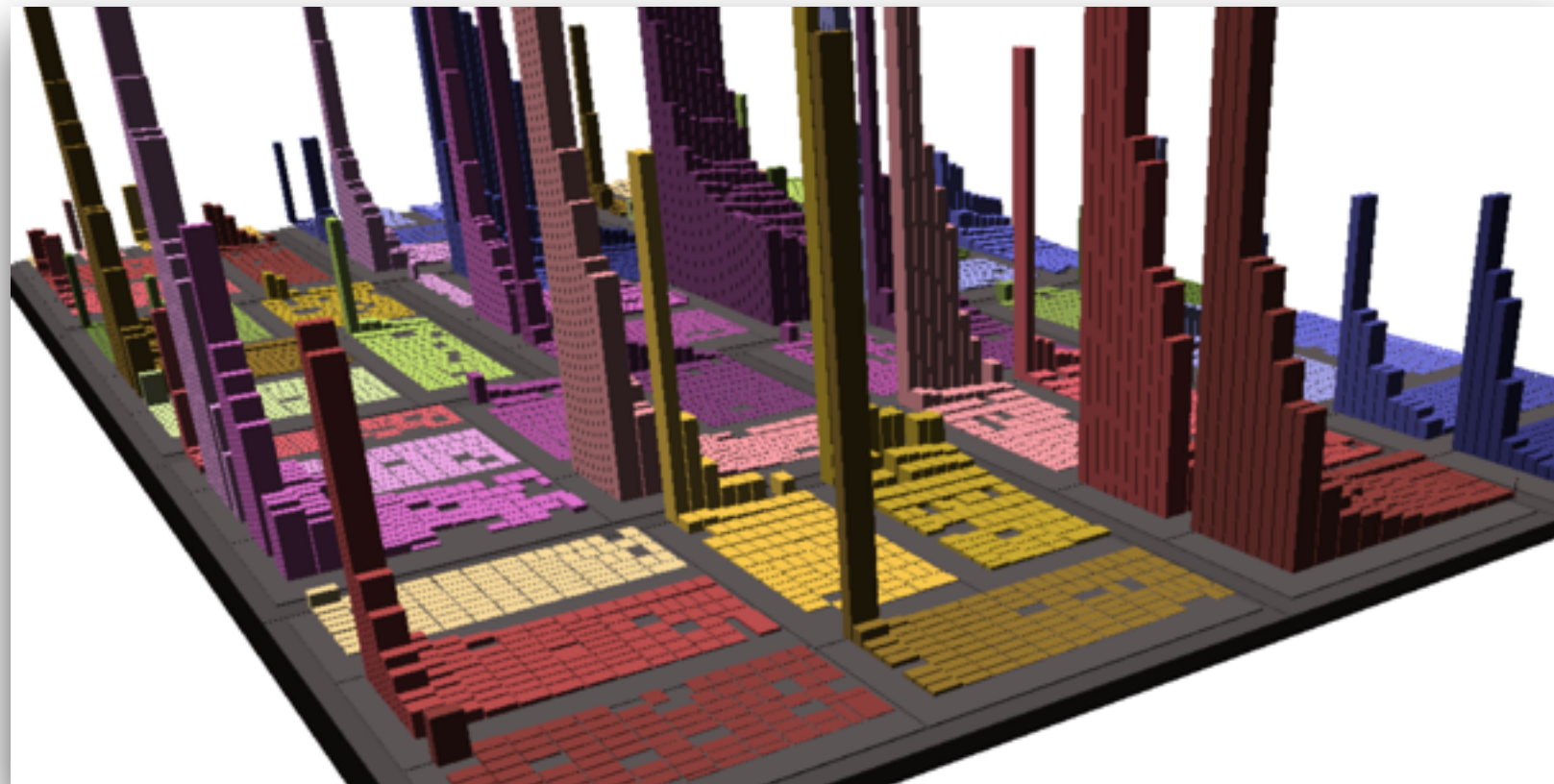
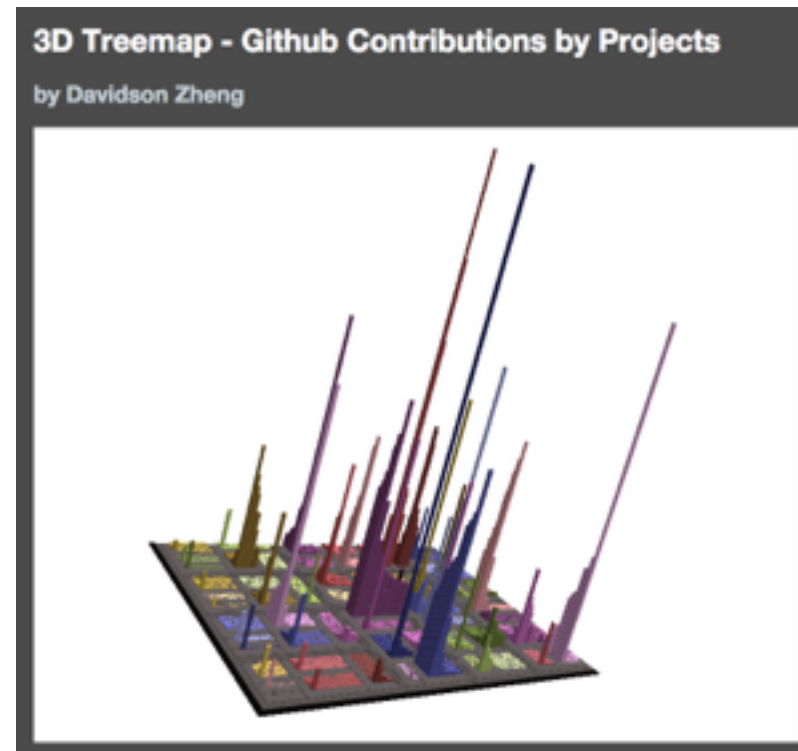
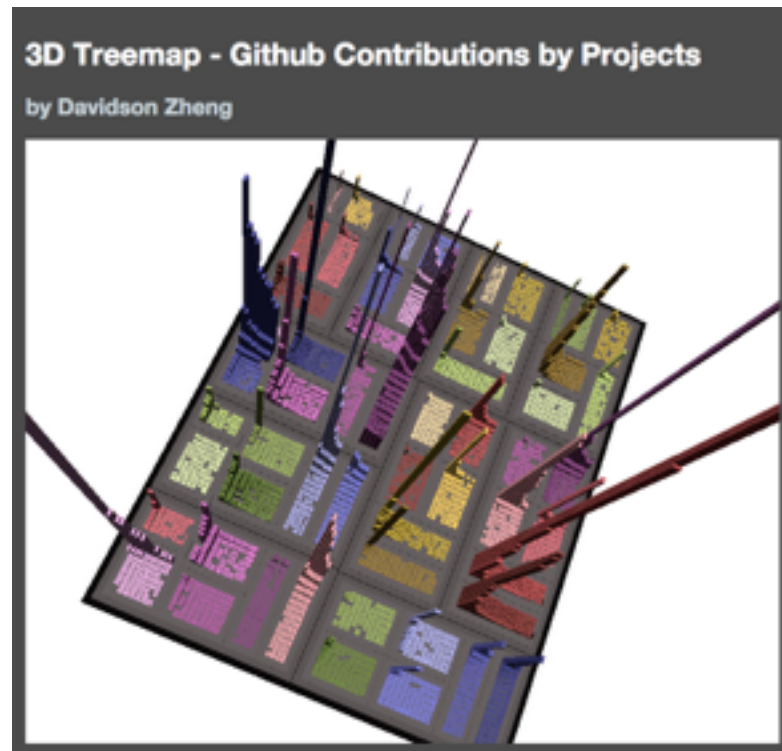


# Visualization using threejs iteration v1





# Visualization using threejs iteration v2



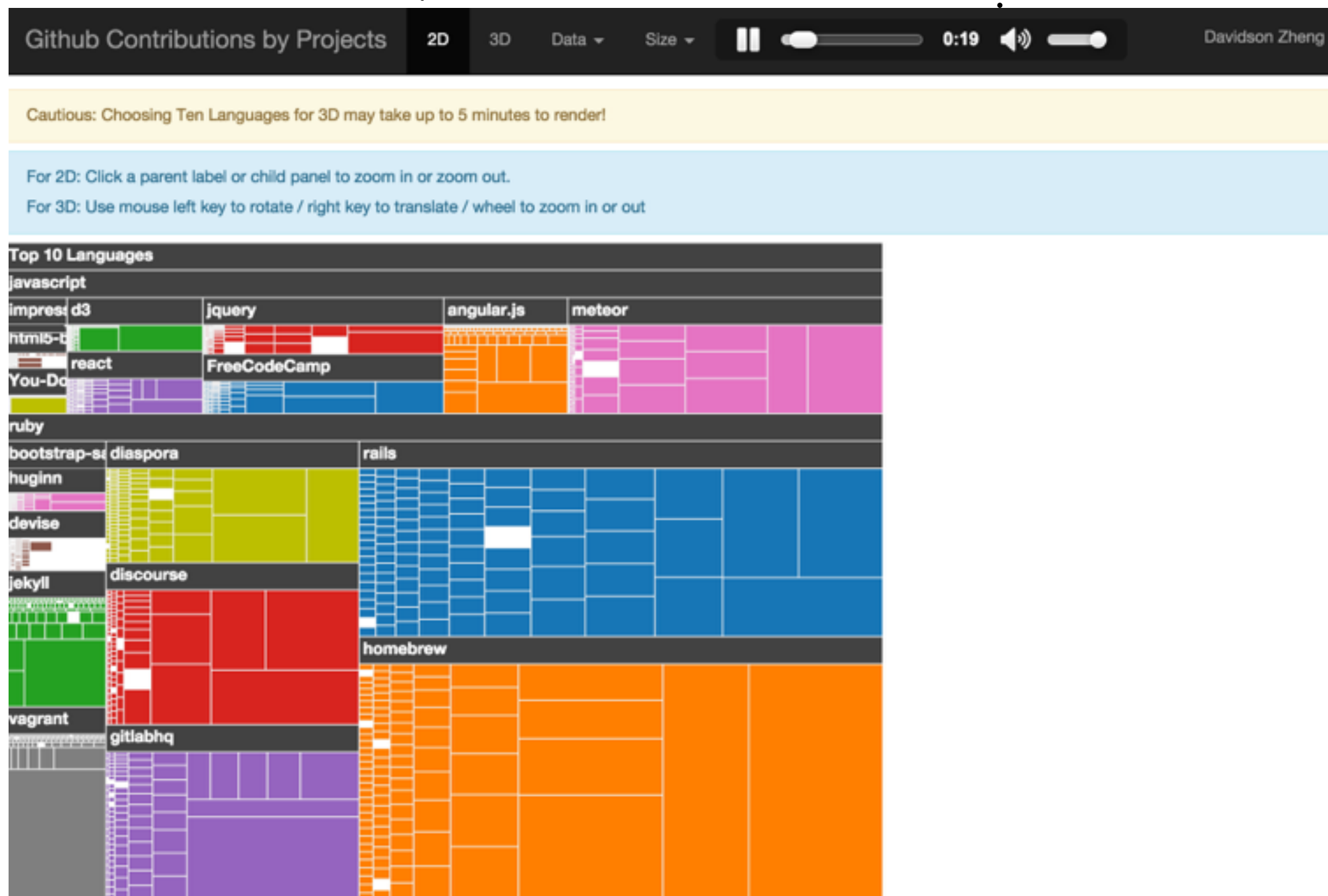
# Visualization with web interface using Bootstrap

change  
dimension to  
2D/3D

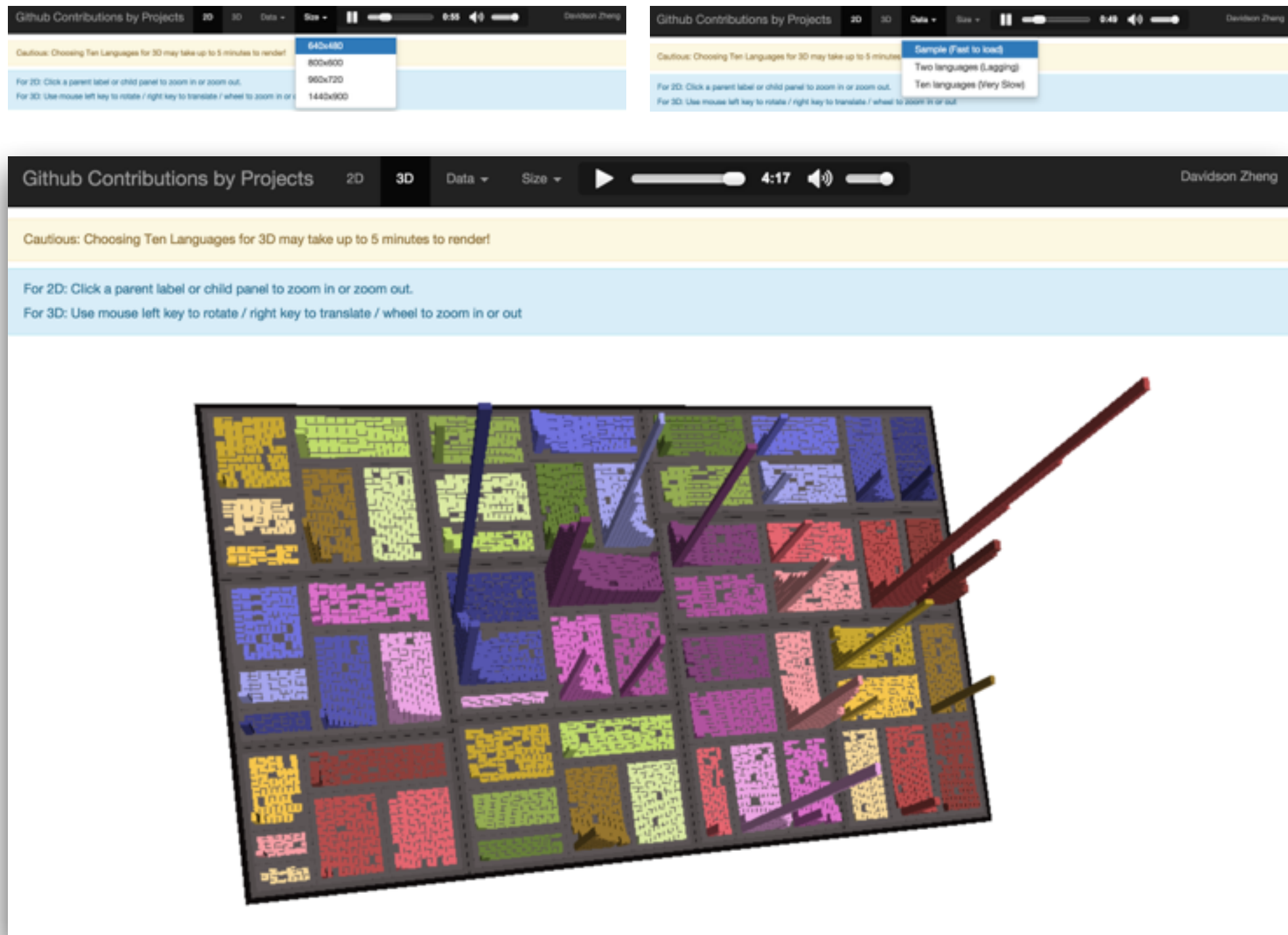
dynamically  
update data  
source

dynamically  
update canvas  
size

play music  
for fun!



# Visualization with web interface using Bootstrap



# Challenges

- jq (JSON processor) is good for retrieving and basic processing, but it is not flexible enough when manipulating data and organizing them into nested arrays
- d3 treemap + threejs, this approach by Bill White renders 3D scene using d3 treemap layout data. It is less intuitive to apply styles for different hierarchies of data and add 2D text labels on top of 3D scene
- huge performance overhead when rendering large dataset that contains > 10000 nodes in web browser
- with antialiasing off, rendering quality is not satisfying
- lagging animations (zoom effect) with > 1000 nodes
- missing chunks in 2D treemap (see last slide)
- 3D scene element messes up HTML layout



# Future Improvement

- Add axis and legend in 3D scene
- Add text labels (contributor names) to children nodes
- Fix missing chunk bugs in treemap layout
- Incorporate better color themes
- Change control to first-person
- Add minimap to highlight active zone
- Fadeout far objects with fogs?
- Improve initial rotation/translation
- Reduce initial performance overhead (how?)

# References

## **d3 Treemap**

<https://github.com/mbostock/d3/wiki/Treemap-Layout>

## **Counting Stars on Github** (treemap visualization of github)

<http://adereth.github.io/blog/2013/12/23/counting-stars-on-github/>

## D3 in 3D: **Combining d3.js and three.js**

<http://www.billdwhite.com/wordpress/2015/01/12/d3-in-3d-combining-d3-js-and-three-js/>

## **d3 Treemap** with Title **Headers**

<http://www.billdwhite.com/wordpress/2012/12/16/d3-treemap-with-title-headers/>

## 5 entertaining things you can find with the **GitHub Search API**

<https://gist.github.com/jasonrudolph/6065289>

## threejs examples - minecraft

[http://threejs.org/examples/#webgl\\_geometry\\_minecraft](http://threejs.org/examples/#webgl_geometry_minecraft)