

Visualize the NPM dependencies city ecosystem of your node project in VR

FOSDEM'23 - JavaScript devroom



Università
della
Svizzera
italiana

David Moreno Lumbreras - david.morenolu@urjc.es

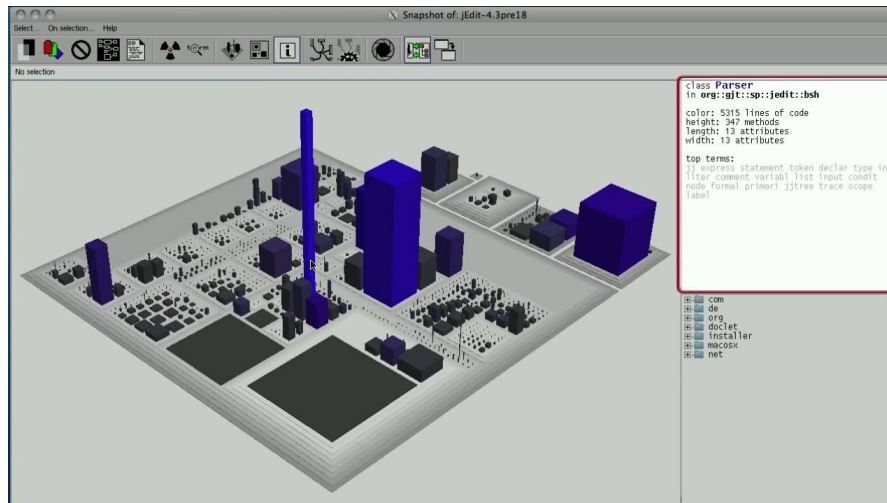


What's the city metaphor?

Software World...

CODECITY

- Software systems as cities
- Building features as metrics (height, area, color)
- Building location as position of the artifact in the system hierarchy
- Locality, orientation, habitability





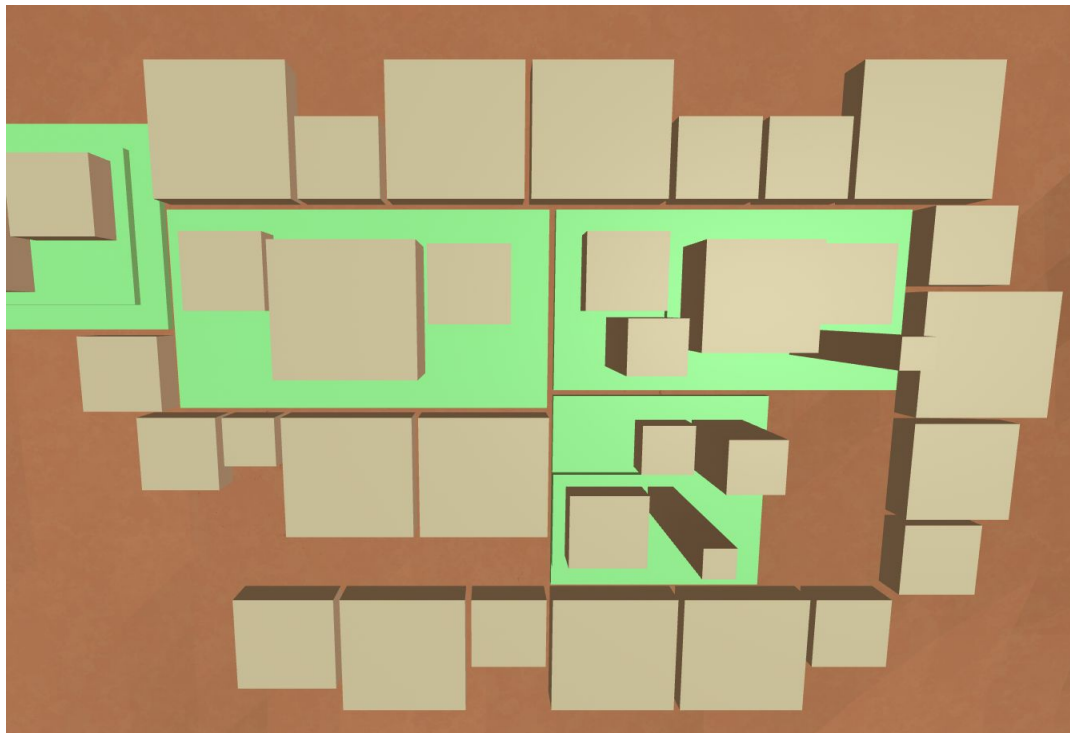
BabiaXR CodeCity visualization



Spiral algorithm - First element at the center of the spiral
Interactive

Building - Source code file (district as file hierarchy)

- **Base Area, Height and Color as metrics**
- **Elevated quarters**

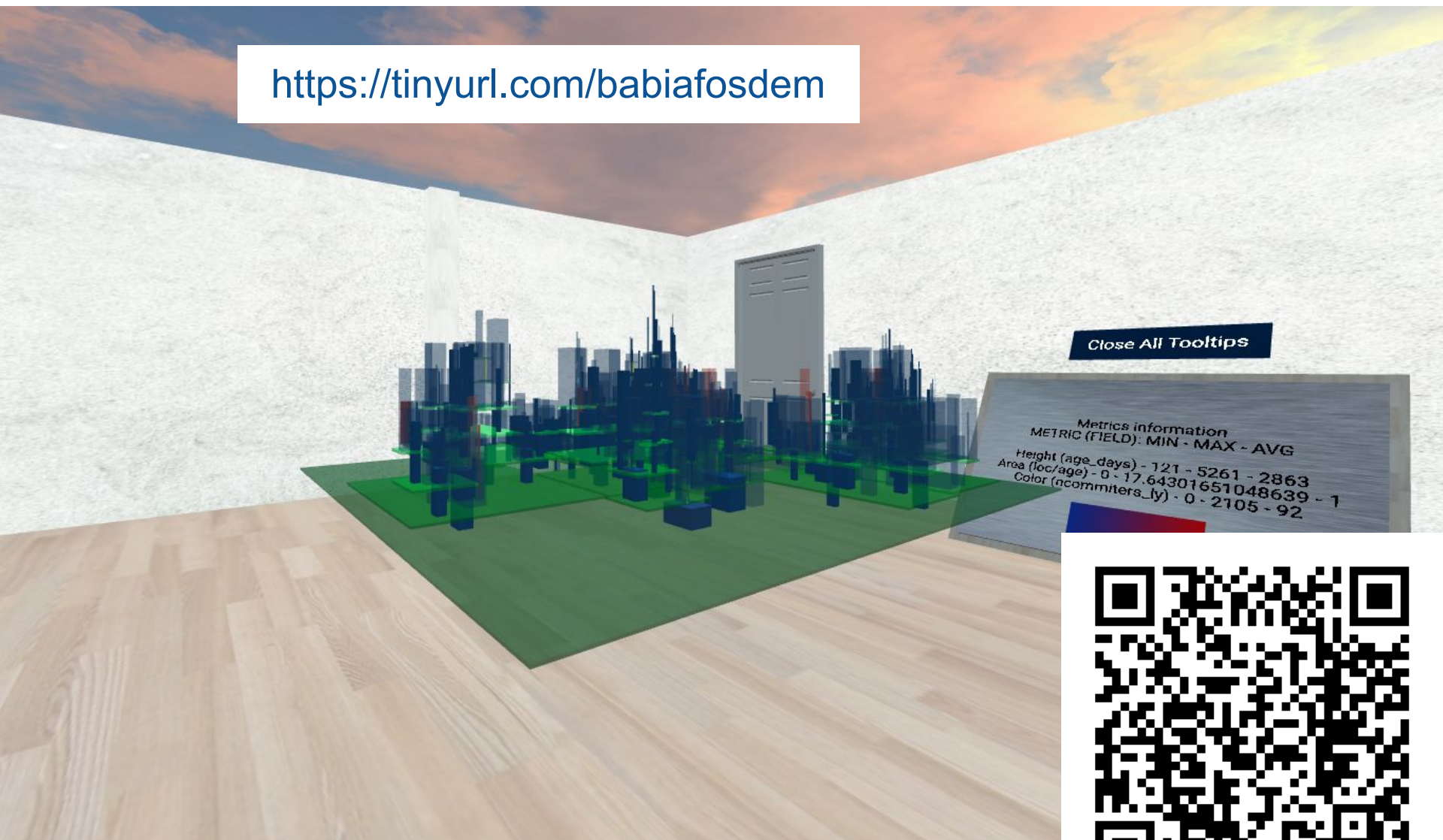


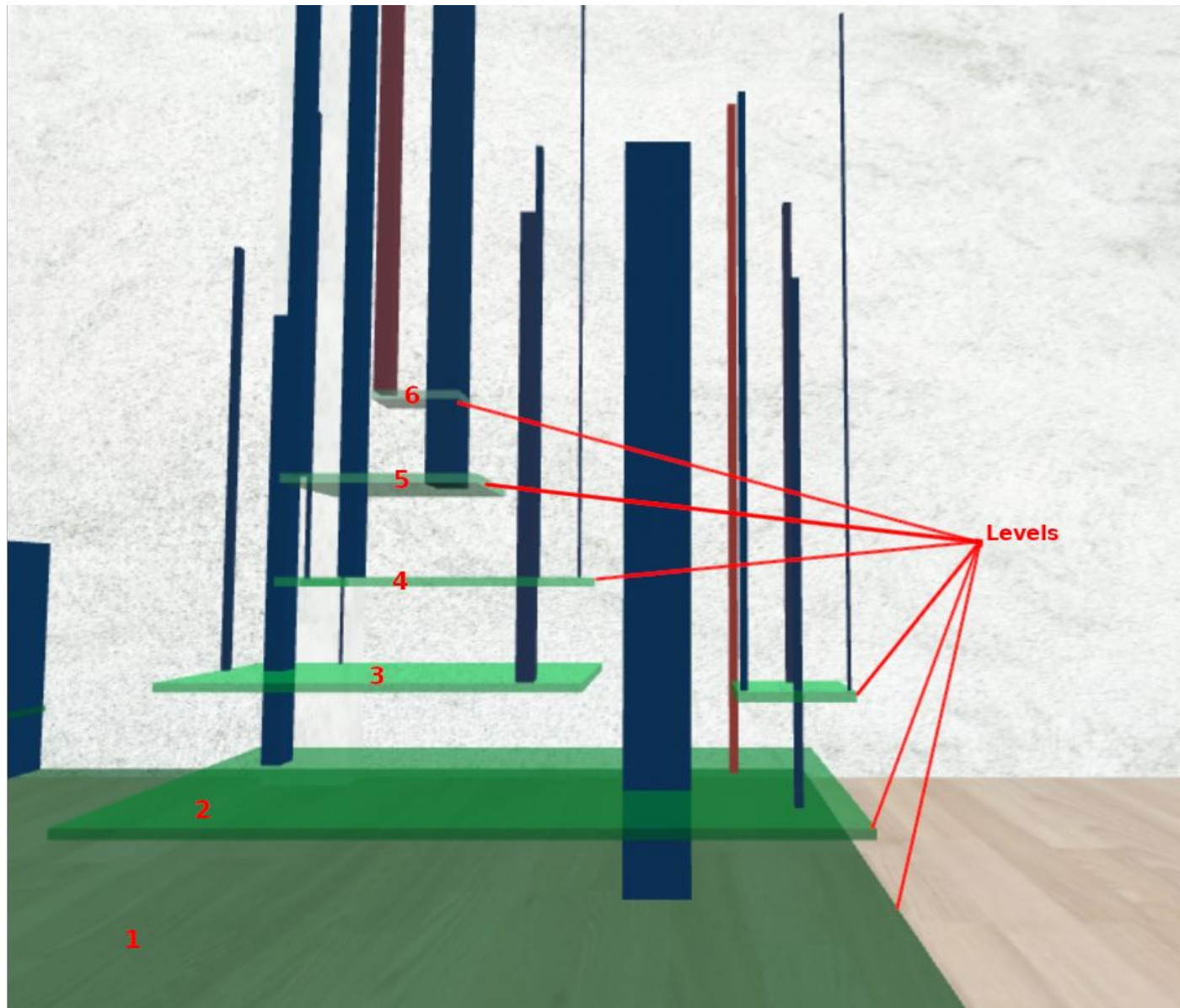


NPM dependencies ecosystem



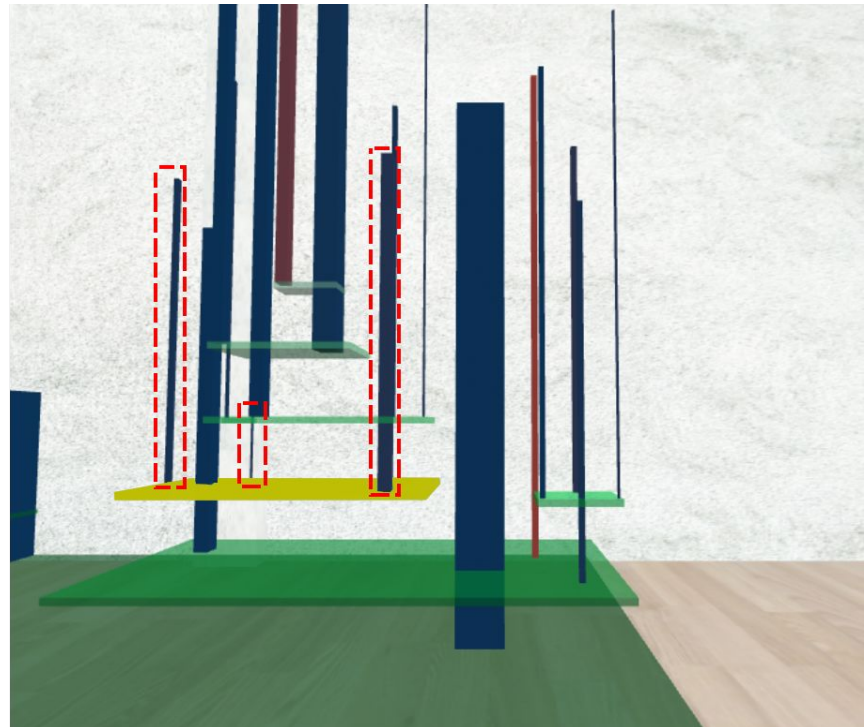
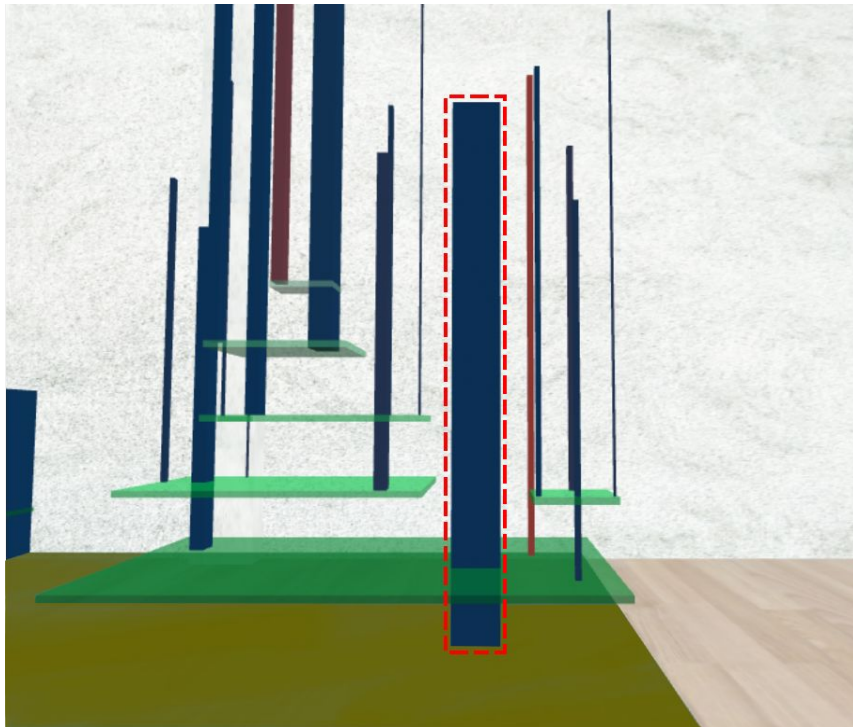
<https://tinyurl.com/babiafosdem>

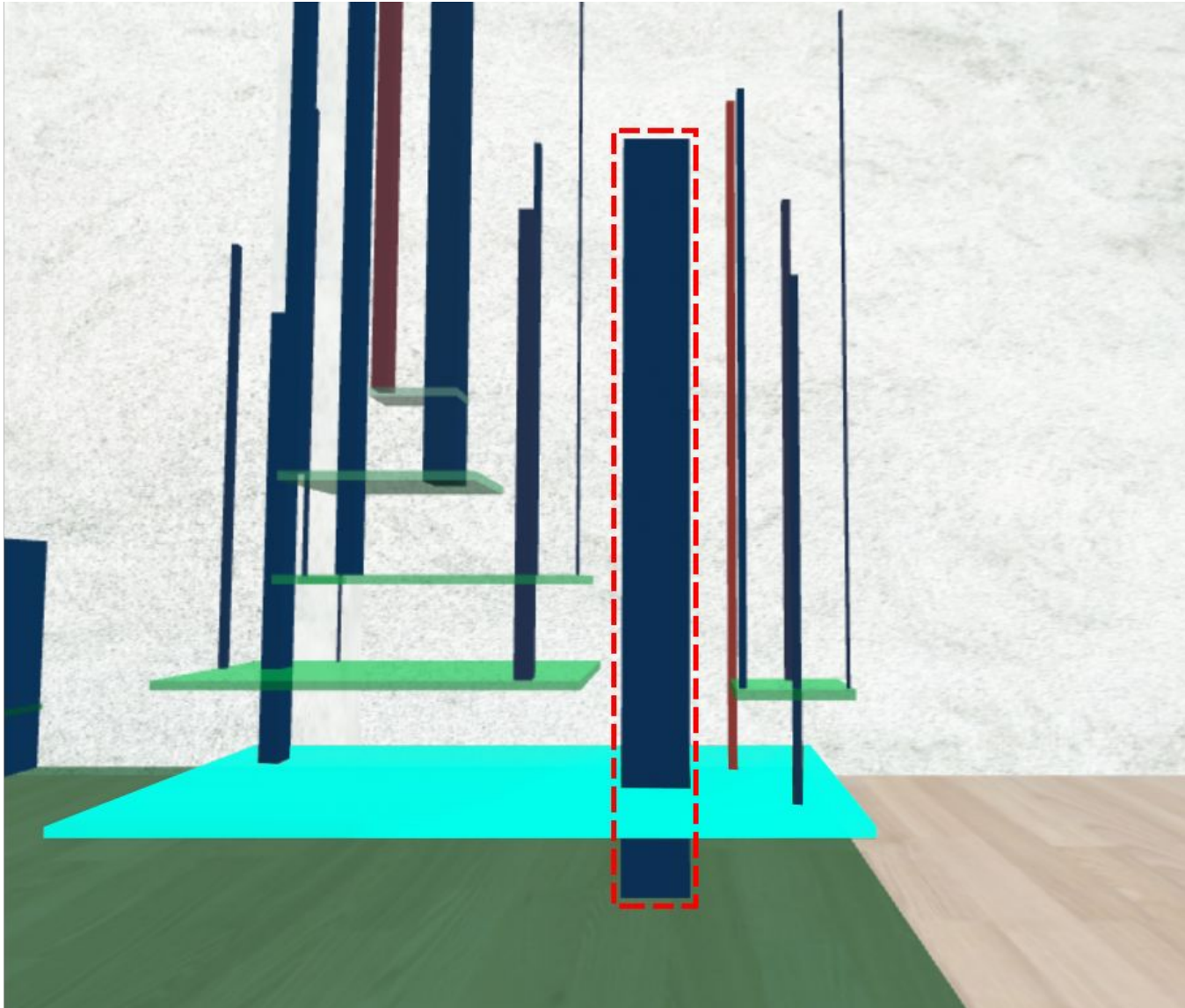






Building location



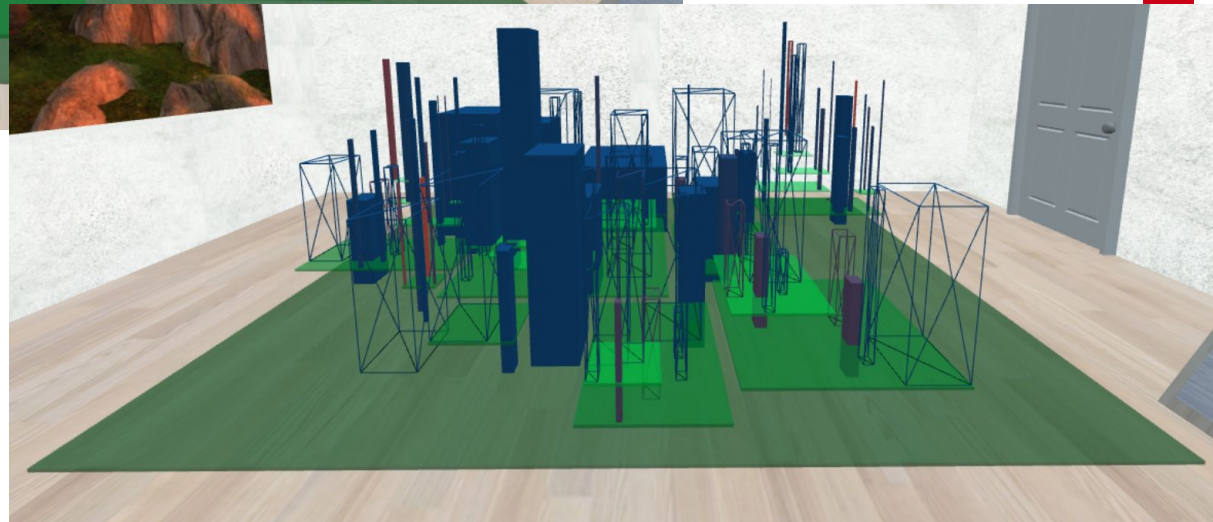
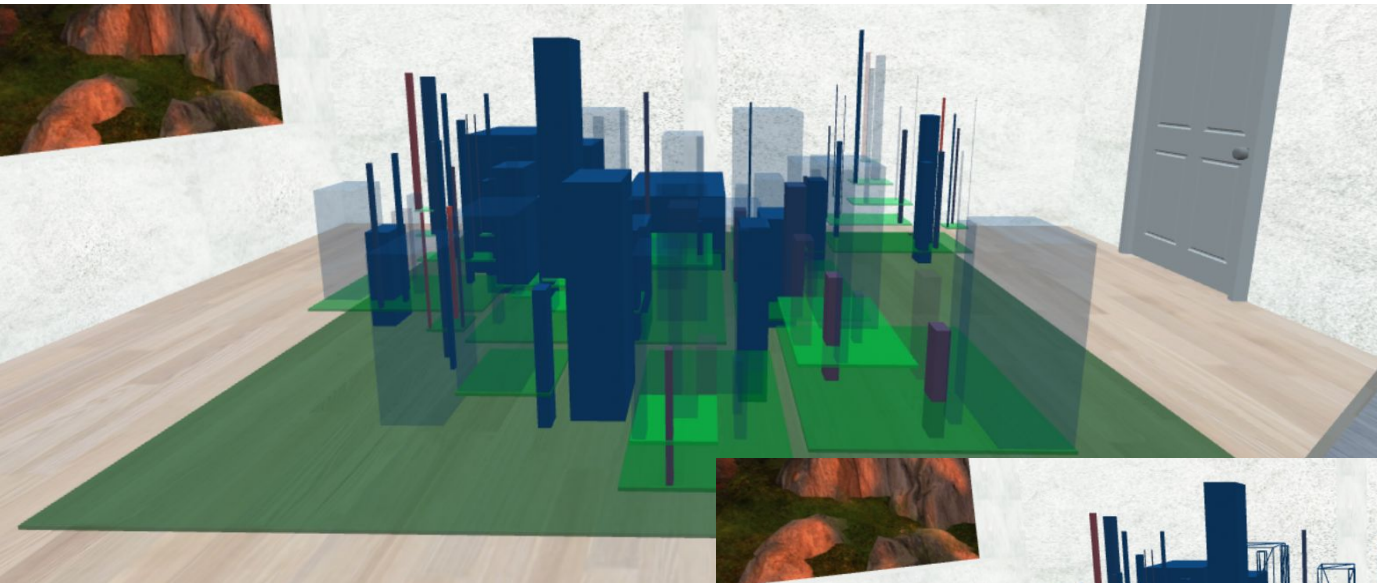
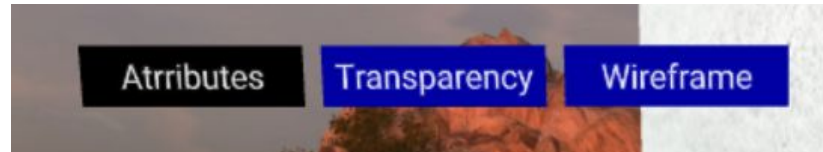


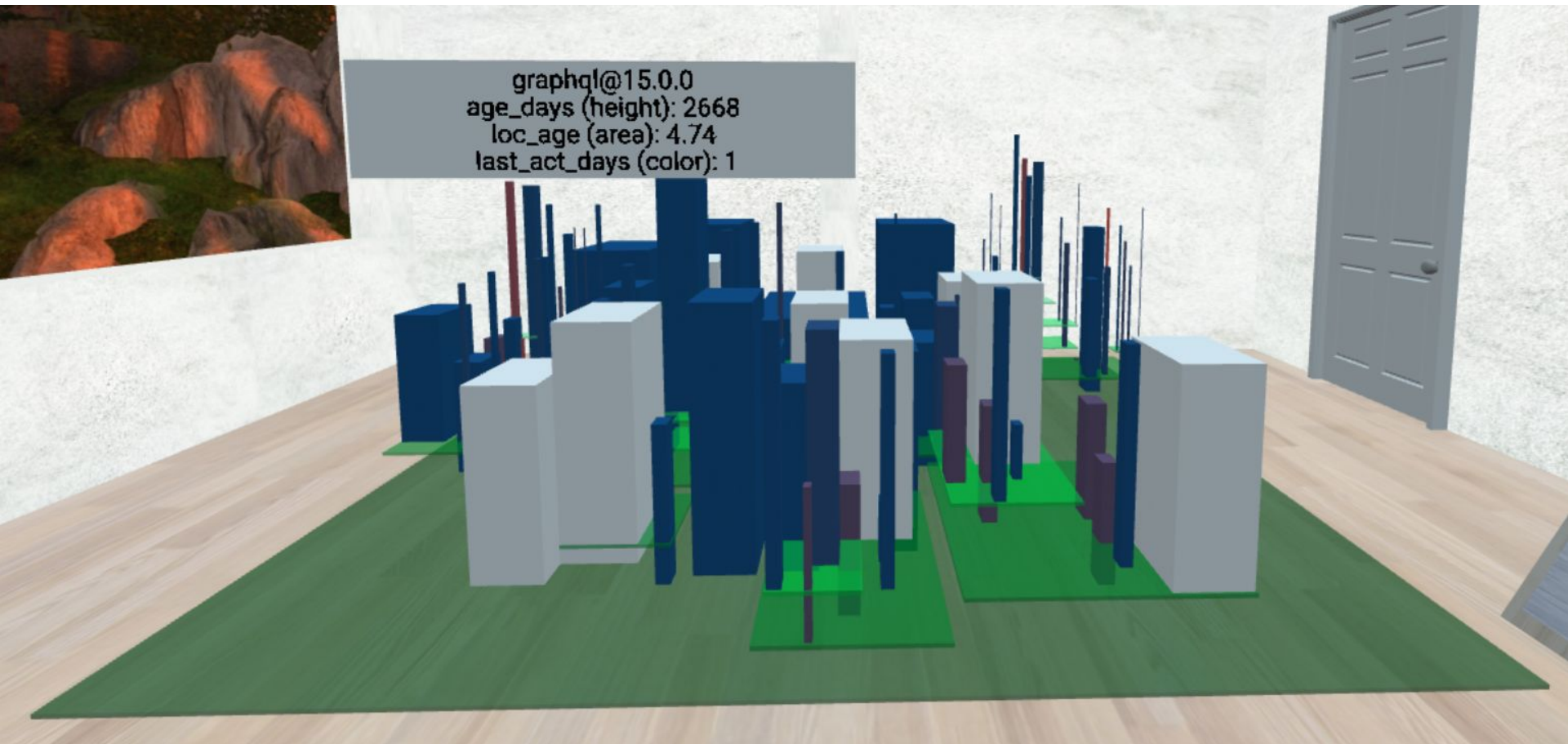


Changing metrics on the fly



Repeated dependencies







Building metrics



Height

The UI can change between these metrics:

- **age_days**: package age in total number of days.

Area

The UI can change between these metrics:

- **loc/age**: lines of code of the package divided by the age in total number of days.
- **size/age**: **size** in bytes of the package divided by the age in total number of days.
- **ncommits/age**: number of commits of the repository of the package divided by the age in total number of days.



Building metrics



Color

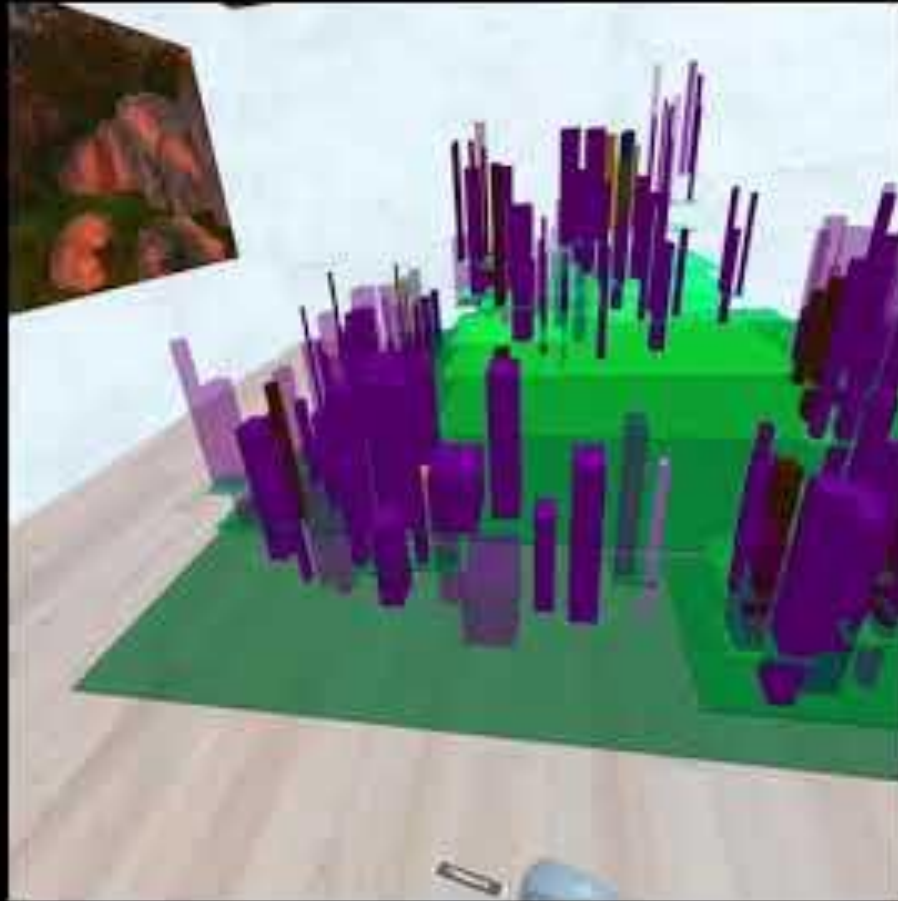
The color of the building is defined by a heatmap from blue to red.



UI can change between these metrics:

- **license**: type of license of the package (**categoric**).
- **timesInstalled**: how many times the package is installed, that means, in how many different versions the package is in the project.
- **timesAppear**: how many times the package appears in the project regardless the version.
- **last_act_days**: days since the last commit in the repository of the package.
- **ncommits_ly**: number of commits in the repository of the package in the last year.
- **ncommiters_ly**: number of unique commiters in the repository of the package in the last year.
- **nvuln**: number of vulnerabilities of the package, retrieved using the audit tool of npm.
- **nissues_ratio**: number of issues closed divided by total number of issues.

But this is for VR!





What we need (and we will have)



Dependencies tree/list of the node project/npm package

Data from the NPM registry (license, repo url...)

Vulnerabilities metrics

Community metrics from the repo

Other “physical” metrics (size, loc, etc...)



Fetching data



Visualizing NPM dependencies

1. Install your NPM package/application

```
npm install --omit=dev <package_name>
```





Fetching data



2. Get the dependencies list and the vulnerabilities

Dependencies list

```
npm ls --all --long --omit=dev --json > npm_list.json
```

Save the path of the file!

Vulnerabilities

```
npm audit --omit=dev --json > npm_audit.json
```

Save the path of the file!





Fetching data



3. Get the metrics from the NPM registry

In this step we are going to fetch all the information (size, license, repo url...) of the package in the NPM registry, if available.

1. Optional: create a python3 virtual env

```
python3 -m venv /tmp/vrdependencies  
source /tmp/vrdependencies/bin/activate
```

2. Install requirements

```
pip3 install -r requirements.txt
```

3. Execute the code

```
cd enrich_dependencies  
python3 get_npm_dep_from_installed.py -npmlist npm_list.json -depoutput <newlist.json> -pjson <projects.
```





Fetching data



Fix URLs that couldn't be found

If there is a repository that couldn't find its url, the script will ask for writing it manually (or leave it blank)

Example:

```
REPO URL NOT FOUND for pkgparent/pkg@version. Please, fill it manually or leave it blank: https://github
```

Returned files

This python code will return two important files for the next step:

- `newlist.json`: a simple 1D list with all the dependencies
- `projects.json`: project.json file for analyze with GrimoireLab





Fetching data



4. Get the community metrics from the package repository

Using GrimoreLab!

GrimoireLab docker images

1. Go to `grimoirelab-docker/default-grimoirelab-settings` and replace the content of `projects.json` file for the generated in the previous step.
2. Go to `grimoirelab-docker/compose` and launch the docker-compose, wait until all the git data is fetched, inside the `/tmp/` folder the logs can be visited.



5. Enrich the final dataset with community metrics, size metrics and github issues/prs metrics.

Important: container from GrimoireLab should be up!

This step is optional, to execute it, you have to go to the `enrich_dependencies` folder and:

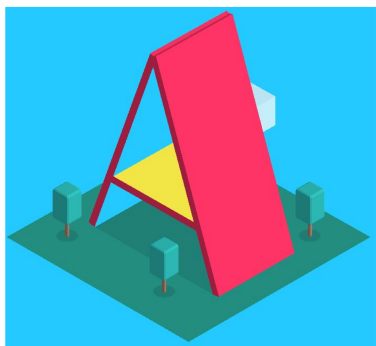
You can use the python venv created before

1. Run the file

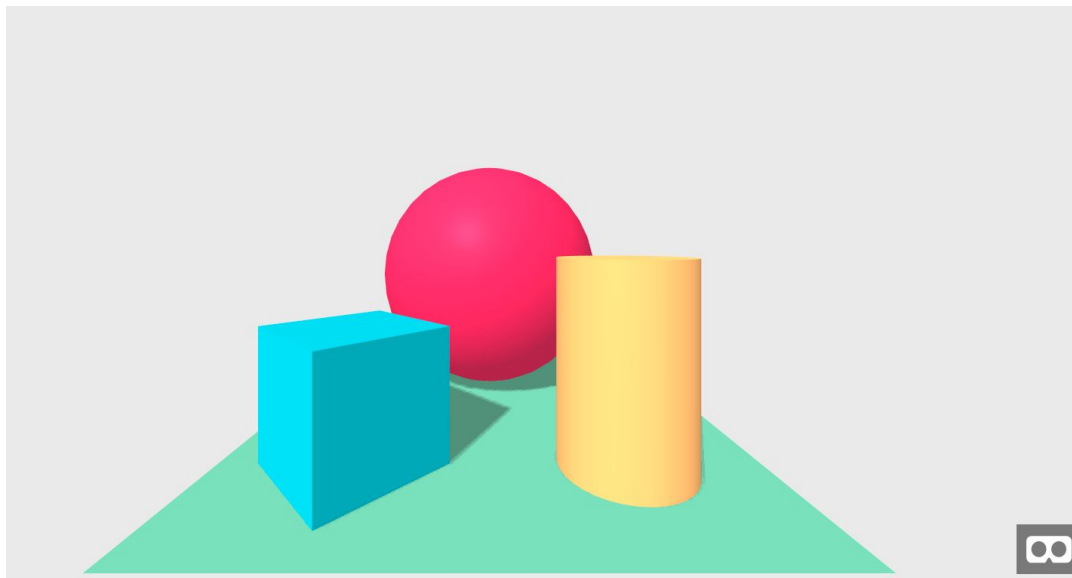
```
python3 enrich_dependencies.py
-deplist <dependencies_list_json>
-depout <dependencies_list_enriches_output_path>
-auditfile <npm_audit_file>
-githubusername <github_username>
-githubtoken <github_token>
```

2. It will take a bit due to the GitHub API rate limit, once finished, the returned file will be in the `<dependencies_list_enriches_output_path>` path ready to use in the city visualization.





```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Hello, WebVR! • A-Frame</title>
    <meta name="description" content="Hello, WebVR! • A-Frame">
    <script src="https://aframe.io/releases/0.8.2/aframe.min.js"></script>
  </head>
  <body>
    <a-scene background="color: #ECECEC">
      <a-box position="-1 0.5 -3" rotation="0 45 0" color="#4CC3D9" shadow></a-box>
      <a-sphere position="0 1.25 -5" radius="1.25" color="#EF2D5E" shadow></a-sphere>
      <a-cylinder position="1 0.75 -3" radius="0.5" height="1.5"
        color="#FFC65D" shadow></a-cylinder>
      <a-plane position="0 0 -4" rotation="-90 0 0" width="4" height="4"
        color="#7BC8A4" shadow></a-plane>
    </a-scene>
  </body>
</html>
```

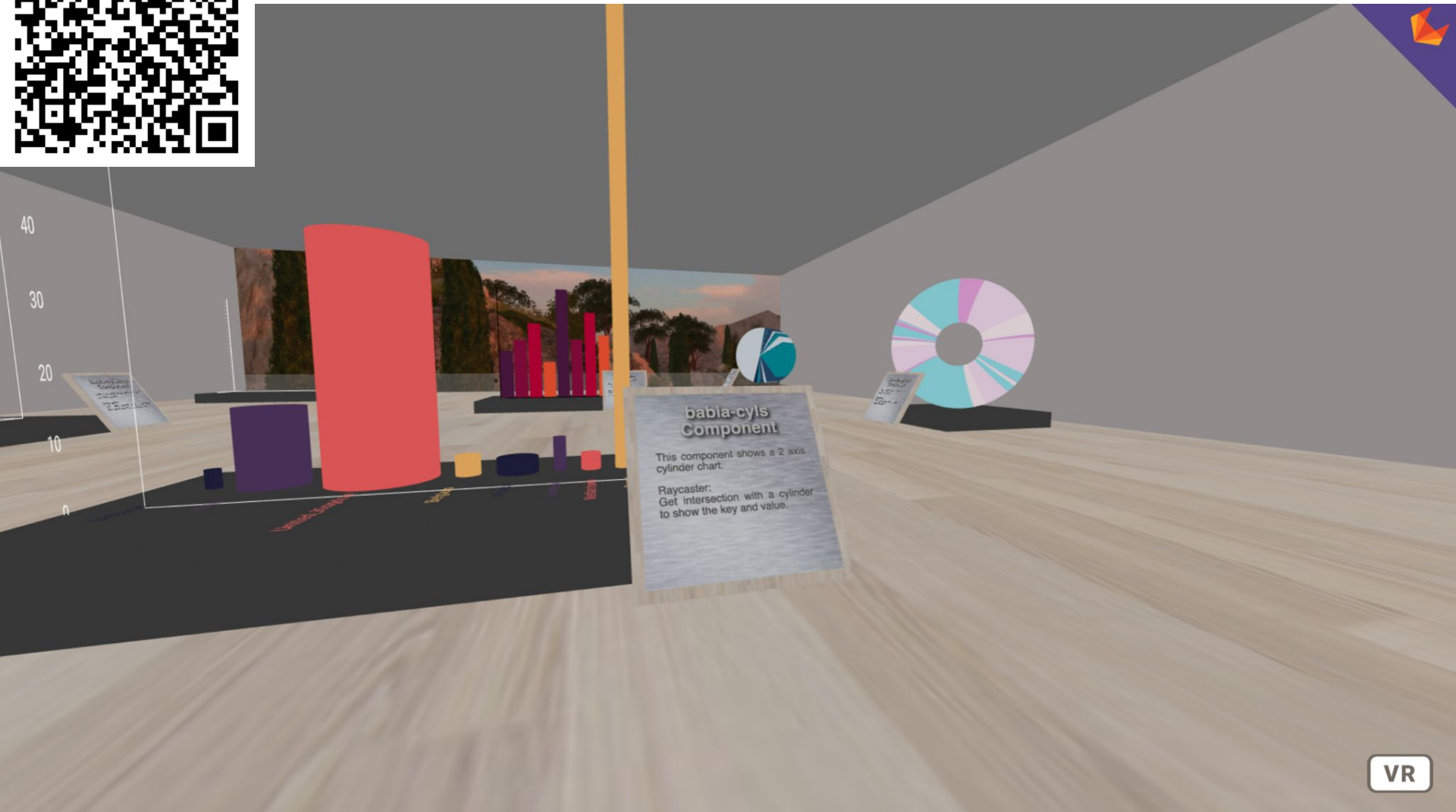




A bit of BabiaXR



<https://babiaxr.gitlab.io/aframe-babia-components/examples/demos/1.0.11/>





A bit of BabiaXR



babia-queryjson

```
<a-entity id="data"  
  babia-queryjson="url: ./data.json;">  
</a-entity>
```



babia-queryes

babia-querygithub





A bit of BabiaXR

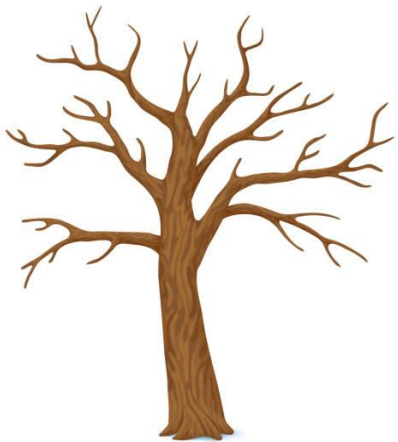


babia-filter

Work In Progress



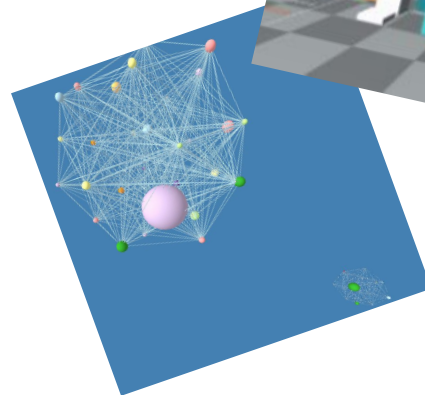
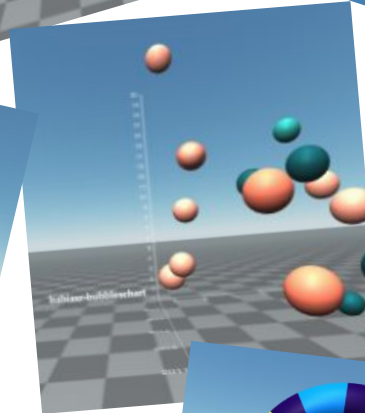
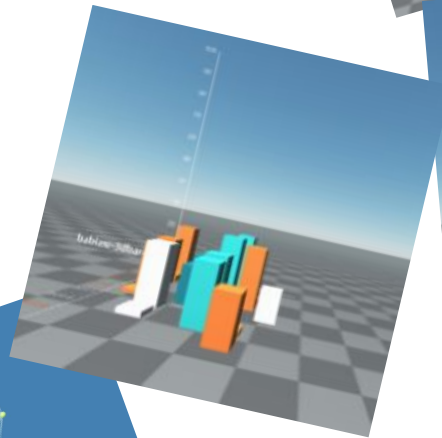
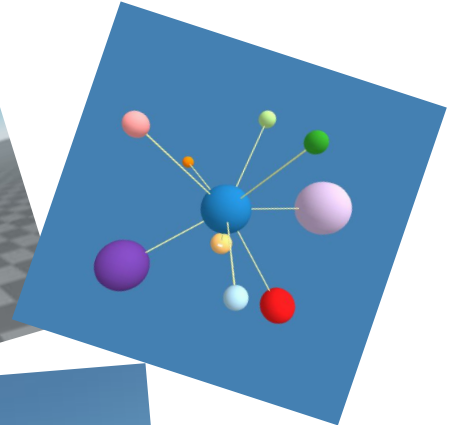
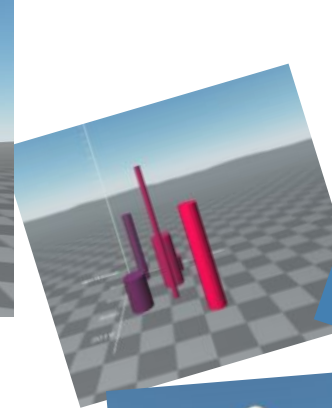
```
<a-entity babia-filter="from: data; filter:  
age=2017"></a-entity>
```



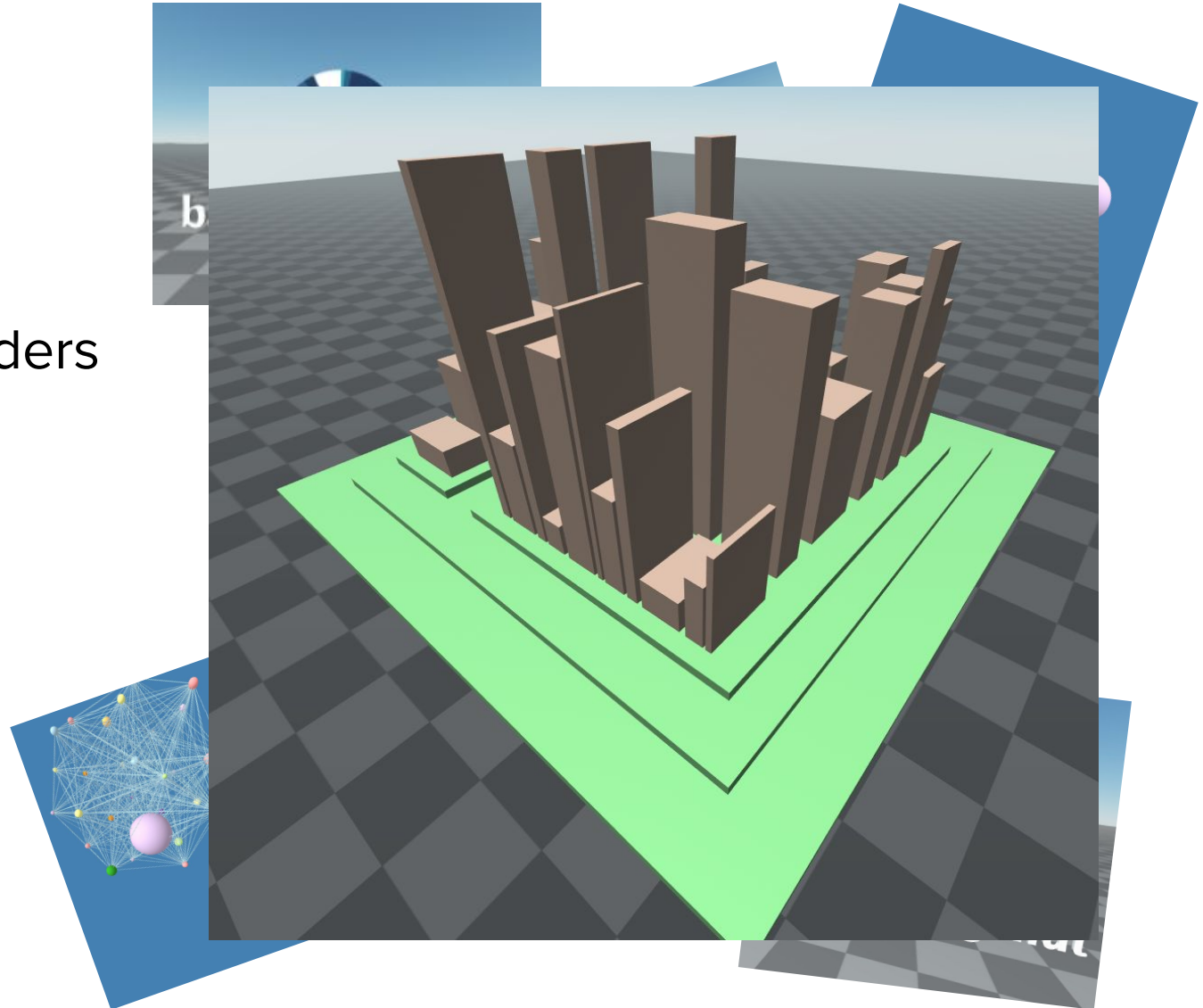
babia-treebuilder



1. Pie
2. Doughnut
3. 3D/2D bars
4. 3D/2D cylinders
5. Bubbles
6. Network
- .
- .



1. Pie
2. Doughnut
3. 3D/2D bars
4. 3D/2D cylinders
5. Bubbles
6. Network
- .
- .
- City**





How to define the scene



```
<a-entity id="ourdata" babia-queryjson="url:./ourdata.json;"></a-entity>
```

```
<a-entity id="ourtreedata" babia-treebuilder="field: parentsPath; split_by: /;  
from: ourdata">
```

```
<a-entity id="depcity"
```

```
babia-boats="from: ourtreedata;
```

```
area: loc/age; height: age_days; color: license;
```

```
treeLayout: true; gradientBaseColor: true; treeQuartersLevelHeight: 0.1; true;
```

```
treeFixQuarterHeight: true; treeHideOneSonQuarters: true;
```

```
></a-entity>
```





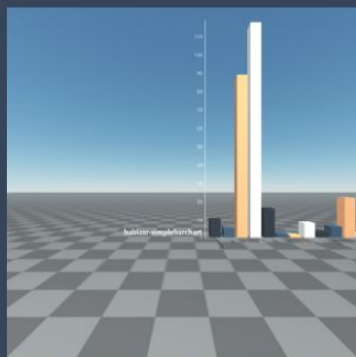
Full examples



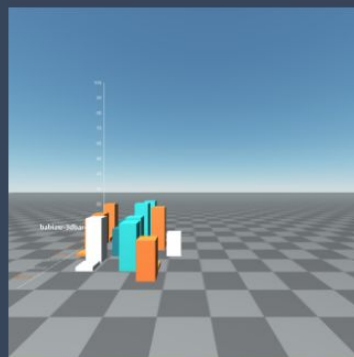
Charts



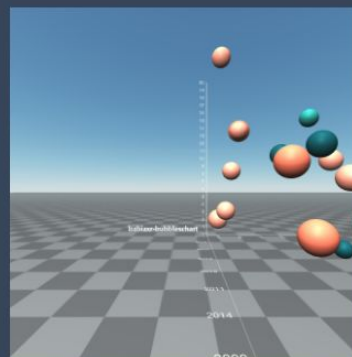
Pie chart



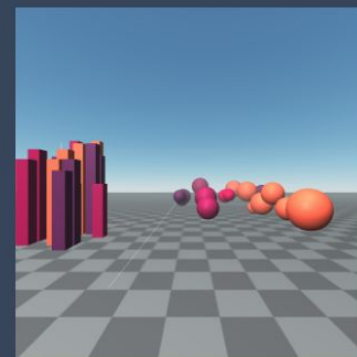
Bars chart



3D Bars chart

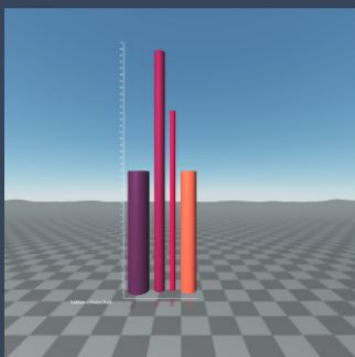


Bubbles chart

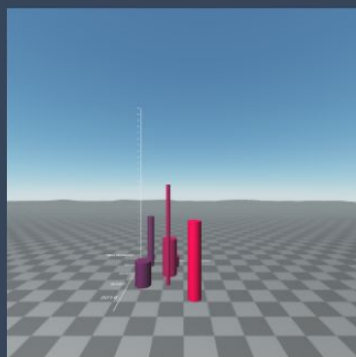


Multiple charts

Mixing of different type of charts



Cylinder charts



3D Cylinders chart



Doughnut chart



<https://babiaxr.gitlab.io/aframe-babia-components/>



Contribute!



GPLv3 License

Contributing guide

<https://gitlab.com/babiaxr/aframe-babia-components/-/blob/master/docs/CONTRIBUTING.md>



Visualize the NPM dependencies city ecosystem of your node project in VR

FOSDEM'23 - JavaScript devroom



Università
della
Svizzera
italiana

David Moreno Lumbreras - david.morenolu@urjc.es

