

# Gizual

## Repository Visualization for Git

Information Visualization, SS 2022  
2022-06-29

Group G2:  
Korduba, Schintler, Steinkellner



# Git Blame

```
P:\gizual > git blame .\package.json
0884c5ad (Stefan Schintler 2022-05-25 22:28:24 +0200 1) {
b0dc06c3 (Stefan Schintler 2022-05-26 15:36:50 +0200 2)     "name": "gizual",
a53149f0 (Stefan Schintler 2022-05-25 22:57:10 +0200 3)     "private": true,
a53149f0 (Stefan Schintler 2022-05-25 22:57:10 +0200 4)     "workspaces": [
a53149f0 (Stefan Schintler 2022-05-25 22:57:10 +0200 5)         "packages/*"
14105d78 (Stefan Schintler 2022-05-26 09:00:26 +0200 6)     ],
```

The diagram below the code block shows labels for the fields in the blame output:

- Short Commit ID**: Points to the first column of the blame output (e.g., 0884c5ad).
- Name**: Points to the author name (Stefan Schintler).
- Timestamp**: Points to the date and time (2022-05-25 22:28:24 +0200).
- Line Number**: Points to the line number in parentheses (1).
- File Content**: Points to the code content on the right side of the output.

# Initial Approach

CLI

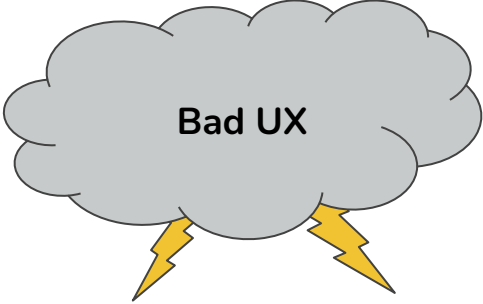
```
usage: ./gizual-cli <flags>

These flags are supported:

--repository=<folder-path> Path to the root repository folder (default: .)
--branch=<branch_name> The branch to explore (default: main)
--commit-id=<commit-id> The commit-id to explore (required)
--file=<file-path> The file which to explore (required)
--json-out=<file-path> Path to output JSON blame (optional)
```

```
1 {
2   "lines": [
3     {
4       "id": 1,
5       "commitHash": "0884c5ad23c4d8b38e9608140cc41c57c75ae2b7",
6       "authorName": "Stefan Schintler",
7       "authorEmail": "stefan.schintler@student.tugraz.at",
8       "updatedAt": "1653547271",
9       "startColumn": 0,
10      "endColumn": 1
11    },
12    {
13      "id": 2,
14      "commitHash": "b0dc06c38c90bdec1006effd1e82db949bfea8dc",
15      "authorName": "Stefan Schintler",
16      "authorEmail": "stefan.schintler@student.tugraz.at",
17      "updatedAt": "1653572210",
18      "startColumn": 1,
19      "endColumn": 18
20    },
21    {
22      "id": 3,
23      "commitHash": "a53149f07493f2f09ecd72de8cd0301ae30a424",
24      "authorName": "Stefan Schintler",
25      "authorEmail": "stefan.schintler@student.tugraz.at",
26      "updatedAt": "1653547271",
27      "startColumn": 1,
28      "endColumn": 17
29    }
30  ],
31}
```

JSON



Repository Visualisation - G2 - 2022

Select a Branch: master

Select a commit: 82064e1 13 days ago

packages/jest-react/index.js 103

packages/jest-react/package.json 25

react-gpt

packages

jest-react

src

index.js

package.json

README.md

react-art

react-cache

react-client

react-debug-tools

features

scripts

package.json

date: 2/15/22  
author: Stefan Schintler  
change: 2 lines

Browser

# libgit2

<https://libgit2.org/>

- C implementation of the Git core methods
- Minimal dependencies
- Cross-platform
- Permissive licensing (GPLv2)
- Used in production by many companies, including:
  - GitHub
  - GitLab

# WebAssembly (Wasm)

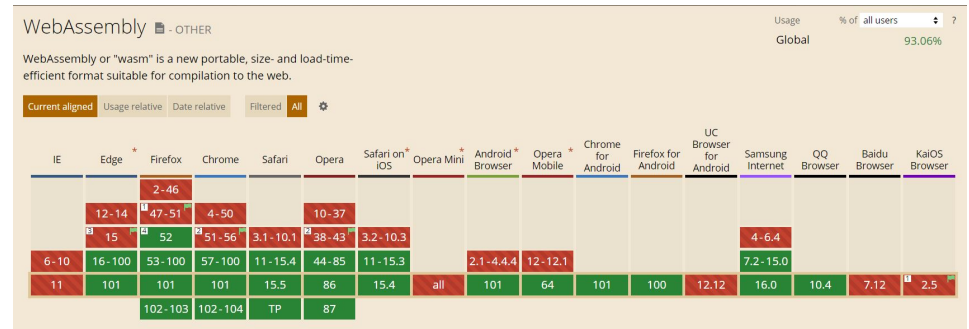
<https://webassembly.org/>

## Facts:

- Modern binary instruction format
- Load-time-efficient virtual stack machine
- Sandboxed and memory-safe

## Wasm allows to:

- Compile executables for the browser
- Use existing native libraries within the browser



Screenshot taken from <https://caniuse.com/wasm>

# Emscripten

<https://github.com/emscripten-core/emscripten>

- Compiler toolchain for WebAssembly
- Supports modern C and C++ (C++17 standard)
- Uses Clang<sup>1</sup> compiler and LLVM-13<sup>2</sup> toolchain
- Ships with a pre-compiled standard library
- Features different forms of FileSystems (e.g. memoryFS)

1: <https://clang.llvm.org/>

2: <https://llvm.org/>

# Emscripten + libgit2 (1)

- No native ability to execute **http requests** from Wasm
  - + Fork libgit2 and apply patches<sup>1</sup>
- No native **filesystem** within Wasm
  - + Use Emscripten's memoryFS implementation
- Libgit2's synchronous API **blocks** the **main thread** (unresponsive UI)
  - + Run wasm module within a Web Worker<sup>2</sup>

1: Inspired by <https://github.com/petersalomonsen/wasm-git>

2: Web Worker <https://web.dev/workers-overview/>

# Emscripten + libgit2 (2)

- Unable to clone into the browser because of **CORS**<sup>1</sup>
  - + Setup a proxy<sup>2</sup> on the server-side to relay requests
- Cloning a large repository takes **too long**
  - + Use of File System Access API as an alternative
- Timestamps in `int64` format, but Wasm is 32 bit
  - + Use of Emscripten's `BigInt` support for large numbers

1: Cross-Origin Resource Sharing <https://web.dev/cross-origin-resource-sharing/>

2: Inspired by <https://github.com/isomorphic-git/cors-proxy>



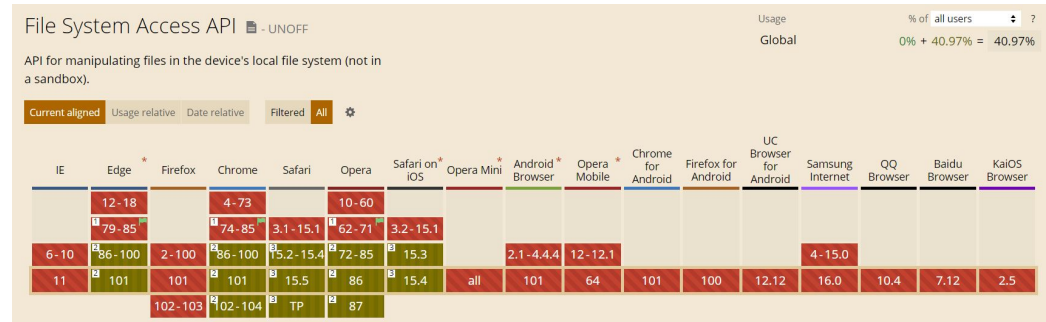
# File System Access API

<https://web.dev/file-system-access/>

- Enables interaction with local files and folders
- Users has to explicitly permit read and/or write access

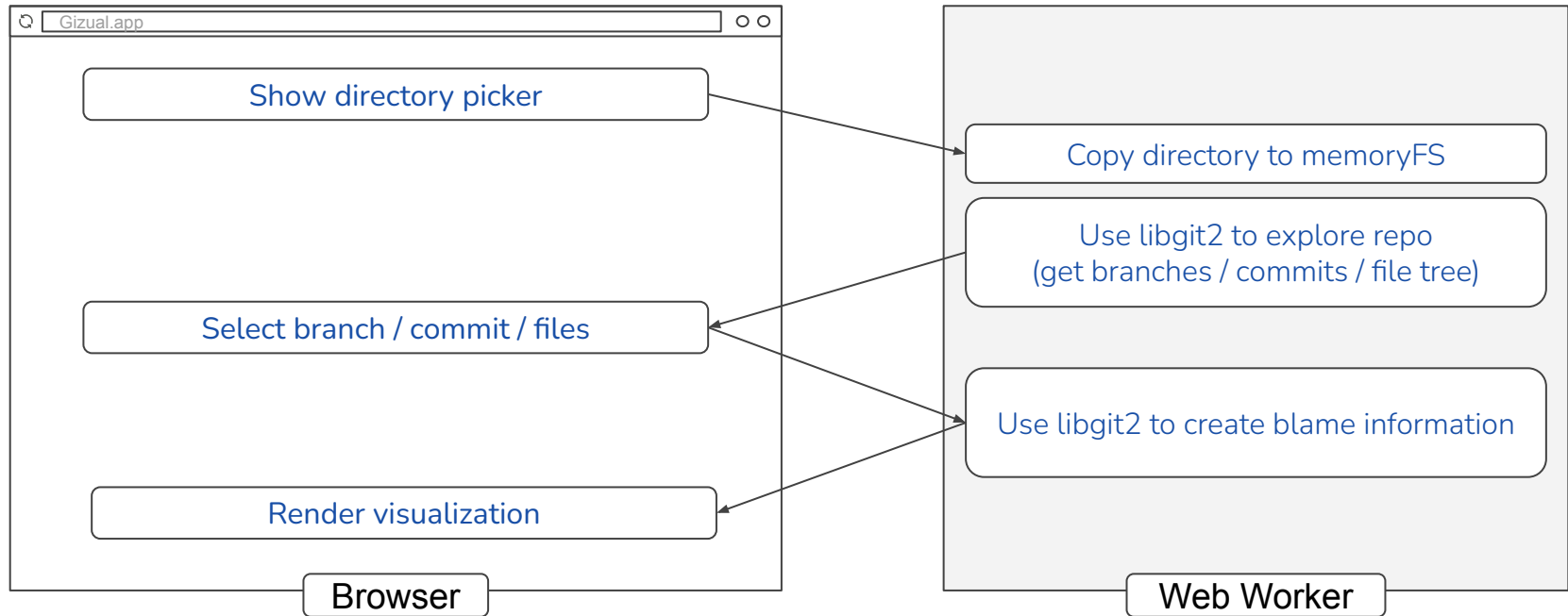
Allows us to:

- Copy local .git folder recursively into Emscripten's memoryFS



Screenshot taken from <https://caniuse.com/native-file-system-api>

# Current Approach



# Visualization: Toolset

- Frontend logic implementation: React<sup>1</sup>
- Component styling: Emotion<sup>2</sup>
- File tree and additional components: Mantine<sup>3</sup>
- Visual representation of the file map: Two.JS<sup>4</sup> and D3<sup>5</sup> (for color scaling)

1: React <https://reactjs.org/>

2: Emotion <https://emotion.sh/>

3: Mantine <https://mantine.dev/>

4: Two.JS <https://two.js.org/>

5: D3 <https://d3js.org/>

# Two.JS

```
const two = useMemo(  
  () =>  
    new Two({  
      type: Two.Types.svg,  
      autostart: true,  
      smoothing: true,  
    }  
  ),  
  []  
);
```

```
const two = useMemo(  
  () =>  
    new Two({  
      type: Two.Types.canvas,  
      autostart: true,  
      smoothing: true,  
    }  
  ),  
  []  
);
```

```
const two = useMemo(  
  () =>  
    new Two({  
      type: Two.Types.webgl,  
      autostart: true,  
      smoothing: true,  
    }  
  ),  
  []  
);
```

CMakeLists.txt  
[50 lines]



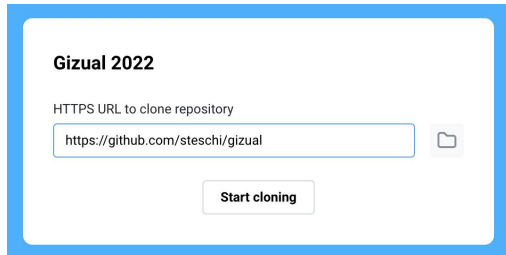
## Benefits:

- Quick change between renderers (SVG, Canvas, WebGL) is possible

## Shortcomings:

- Event handling needs additional adjustments, if the renderer is changed

# Visualization: Repository and File Selection



Branch

Commit

- Andreas Steinkellner  
Fix: Hide scrollbars when not needed  
**aed66d6** 2 minutes ago
- Yaryna Korduba  
Add color mapping by author.  
**0d5a8b0** 1 day ago
- Andreas Steinkellner  
Merge branch 'wip/repo-stats'  
**e2c6e23** 2 days ago
- Andreas Steinkellner  
Fix: Attach repo stats to specific commit id instead of branch head  
**626b0fe** 2 days ago
- Andreas Steinkellner  
Feat: Add time info to commit select box  
**58df9e7** 2 days ago



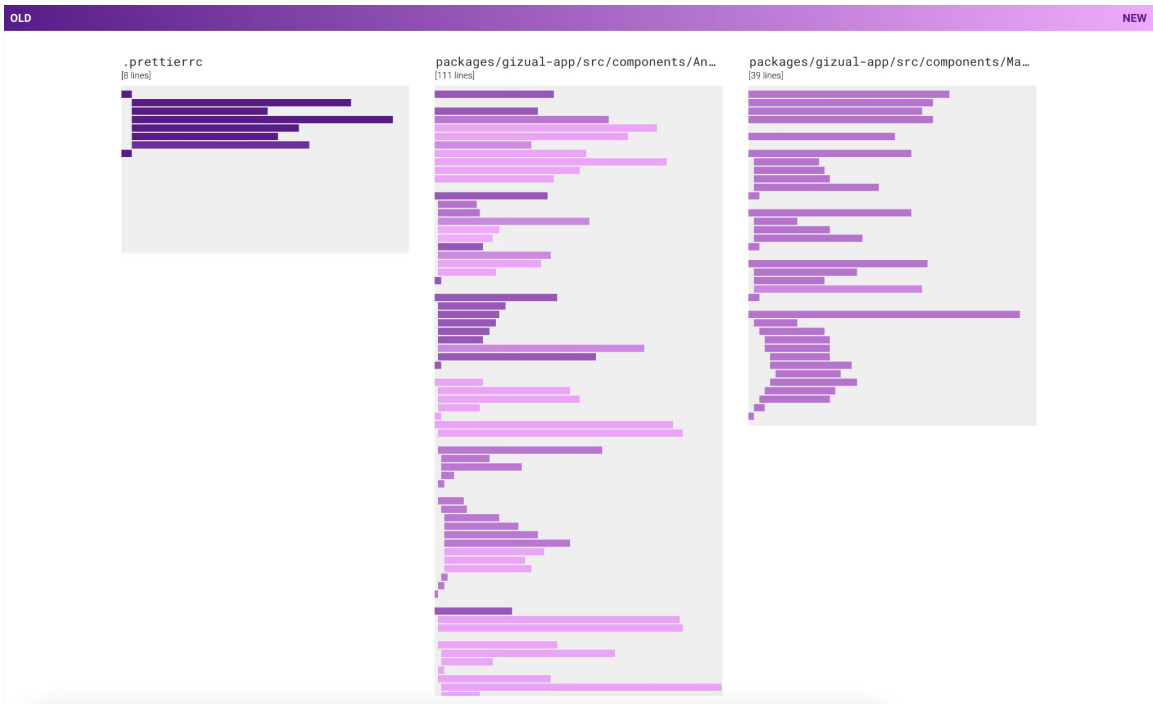
Branch

Commit

---

- public
- src
  - assets
  - commonStyles
  - components
  - helpers
  - hooks
    - index.js
    - memoryView.hks.js**
    - reactRedux.hks.js
  - icons
  - redux
  - index.js
  - serviceWorker.js
  - store.js
- .eslintrc
- .gitignore
- .prettierrc**
- README.md
- package-lock.json
- package.json**

# Visualization: File Map

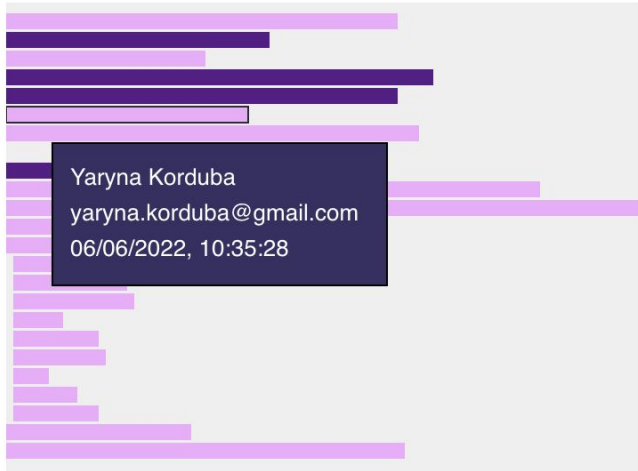


- The files are displayed in a row using CSS Grid Layout
- The layout is adjustable to a screen size
- The length and position of each line imitate its length and position in a real file
- Color of a line displays a particular feature (date of last edit or author)
- The legends are adjusted according to the feature shown

# Visualization: Interactivity

...gizual-app/src/components/FileMap.tsx

[225 lines]



- Hovered line is highlighted
- Line-specific tooltip is displayed
- Tooltip positioning takes into account container boundaries, so that the tooltip never overflows them

# Deployment

- Automated on each push to main using GitHub Actions<sup>1</sup>
- Bundled with Vite<sup>2</sup>

```
[build:client] dist/client/assets/gizual-worker.b76f3d03.wasm 666.99 KiB
[build:client] dist/client/assets/index.5aa27619.js          120.33 KiB
[build:client] dist/client/index.html                       0.42 KiB
[build:client] dist/client/assets/index.6baa7e99.css         0.36 KiB / gzip: 0.26 KiB
[build:client] dist/client/assets/index.87b88d9a.js          527.33 KiB / gzip: 156.15 KiB
[build:server] dist/server/main.js                          3.37 KiB
```

- Packaged in a Docker<sup>3</sup> Container
- Deployed on a server configured with Let's Encrypt<sup>4</sup> to support HTTPS
- Available at <https://gizual.xyz>

1: GitHub Actions <https://github.com/features/actions>

2: Vite Bundler <https://vitejs.dev/>

3: Docker <https://www.docker.com/>

4: Let's Encrypt <https://letsencrypt.org/>



# Cheers!

Thanks for your attention!