unconed / **mathbox** Public

Code

Issues 7

Pull requests

Actions

Projects

Wiki

Security

Insights

master

ChristopherChudzicki prepare release v2.2.0 (#35)
9e315b5 Apr 24, 2022
434 commits

.github/workflows
Separate linting and tests in Github Actions (#33)
Apr 18, 2022

config
Add typedefs
Mar 26, 2022

docs
Port work from 'dev' to master (#32)
Apr 19, 2022

examples
Port work from 'dev' to master (#32)
Apr 19, 2022

src
prepare release v2.2.0 (#35)
Apr 24, 2022

test
fix gridlines on filled surfaces (#27)
Apr 14, 2022

.eslintrc.js
Add typedefs
Mar 26, 2022

.gitignore
remove release/ directory (#28)
Apr 14, 2022

.prettierignore
Add shaders.js to the bundle, delete vendored gulp-jsify
Mar 4, 2022

CHANGES.md
prepare release v2.2.0 (#35)
Apr 24, 2022

FUNDING.yml
Upgrade project to modern ES6 modules, modern three.js
Feb 16, 2022

LICENSE.md
Upgrade project to modern ES6 modules, modern three.js
Feb 16, 2022

README.md
add developing.MD (#30)
Apr 17, 2022

RUNNING.md
Upgrade project to modern ES6 modules, modern three.js
Feb 16, 2022

developing.md
add developing.MD (#30)
Apr 17, 2022

gulpfile.js
remove gulp-eslint (#31)
Apr 17, 2022

karma.conf.js
Tidy config, update package.json fields
Mar 3, 2022

package-lock.json
Port work from 'dev' to master (#32)
Apr 19, 2022

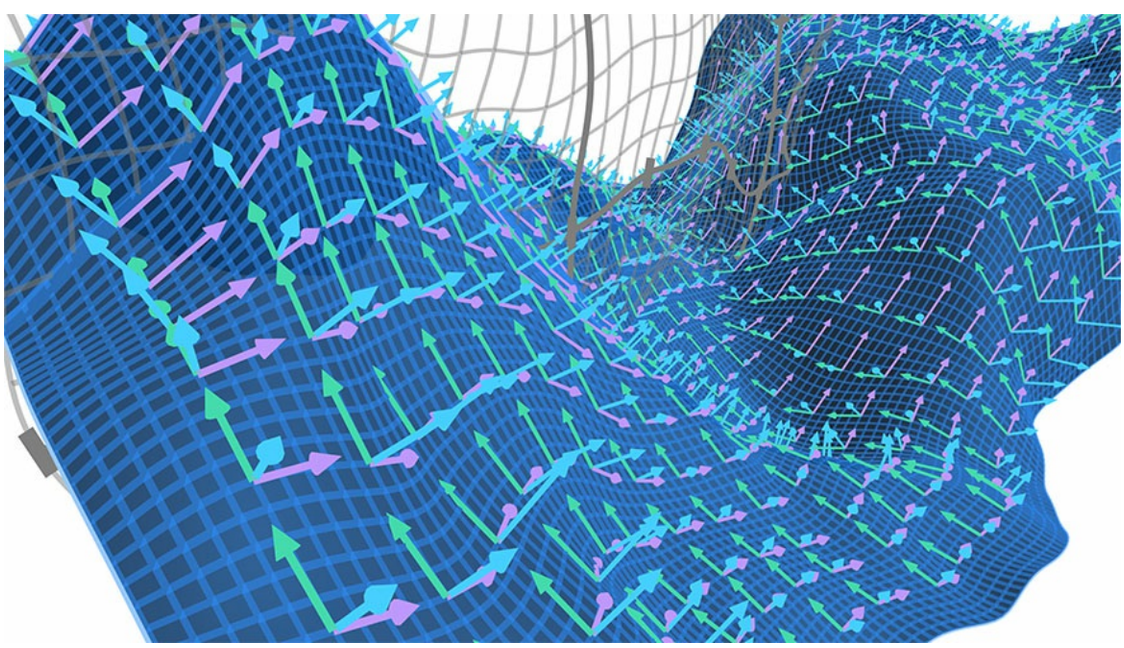
package.json
prepare release v2.2.0 (#35)
Apr 24, 2022

tsconfig.json
release version 2.1.3 (#24)
Apr 5, 2022

README.md

MathBox

Presentation-quality WebGL math graphing



MathBox is a library for rendering presentation-quality math diagrams in a browser using WebGL. Built on top of Three.js and ShaderGraph, it provides a clean API to visualize mathematical relationships and animate them declaratively.

MathBox is used through its JS-based DOM.

For background, see the article series on Acko.net.

Demos:

- Audio Visualizer
- Cylindrical Stream
- Data/Shape Mapping
- LaTeX/HTML/GL Labels
- Quaternion Hypersphere
- Render-to-Texture History
- Vertex Warping
- Volumetric Vectors

Presentations:

- The Pixel Factory

Note: this repo uses submodules, clone it with `--recursive` or do a `git submodule update --init` after cloning.

Installation

- via npm:

```
npm install mathbox
```

- or CDN:

```
<script
  type="text/javascript"
  src="https://cdn.jsdelivr.net/npm/mathbox@latest/build/bundle/mathbox.js"
></script>
<link
  rel="stylesheet"
  href="https://cdn.jsdelivr.net/npm/mathbox@latest/build/mathbox.css"
/>
```

Open the included `/examples` to see more demos.

Docs & Help

See:

- Quick Start for a hands on introduction.
- Glossary of terms to help get familiar with MathBox and WebGL.
- MathBox API for typical usage.
- List of Primitives for a full element reference.
- Writing Custom Shaders for info on custom shaders and GPU-side processing.
- Context API for advanced usage.

Join us in the MathBox Google Group or #mathbox on Freenode.

Basic Usage

Construct a MathBox instance using the provided `mathBox()` constructor:

```
import { mathBox } from "mathbox";
import "mathbox/mathbox.css";

const mathbox = mathBox(options);

const three = mathbox.three;
// three.renderer, three.scene, three.camera
```

See threestrap for all available `options`. e.g. To spawn inside a specific element, do:

```
const element = document.querySelector("#my-thing");
const mathbox = MathBox.mathBox({ element: element });
```

On initialization, it returns a MathBox API object, wrapping the MathBox `<root/>`. You can spawn new nodes:

```
<cartesian
  range={[
    [-2, 2],
    [-1, 1],
    [-1, 1],
  ]}
  scale={[2, 1, 1]}
>
  <axis axis={1} />
  <axis axis={2} />
</cartesian>
```

via

```
const view = mathbox
  .cartesian({
    range: [
      [-2, 2],
      [-1, 1],
      [-1, 1],
    ],
    scale: [2, 1, 1],
  })
  .axis({
    axis: 1,
  })
  .axis({
    axis: 2,
  });
```

You can select objects using `.select()` and a CSS-like selector to get a jQuery-like selection:

```
mathbox.select("cartesian > axis");
```

Use `.print()`, `.inspect()` and `.debug()` to show information about a selection.

Related Projects

- threestrap - Three.js bootstrapper
- shadergraph - Functional GLSL linker

MathBox and ShaderGraph (c) Steven Wittens 2013-2022. MIT License.

Libraries and 3rd party shaders (c) their respective authors.

About

Presentation-quality WebGL math graphing

Readme

MIT license

838 stars

31 watching

90 forks

Releases 8

v2.2.0 Latest
Apr 24, 2022

+ 7 releases

Sponsor this project

Sponsor

Learn more about GitHub Sponsors

Packages

No packages published

Used by 31

+ 23

Contributors 6

Languages

- JavaScript 83.0%
- TypeScript 16.3%
- CSS 0.7%