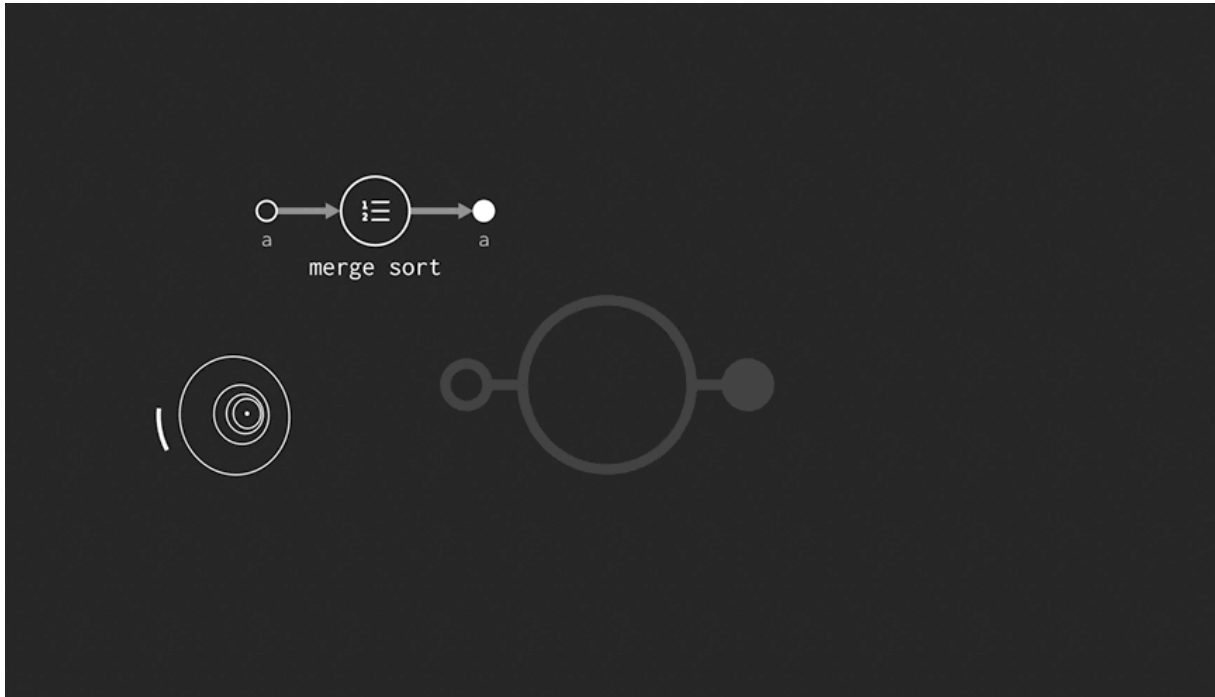

Unit

Next Generation Visual Programming System



Unit

Unit is a General Purpose Visual Programming Language and Environment built with a primary focus on Developer Experience.

It is heavily inspired by Live, Data Flow, Reactive, Functional and Object Oriented Programming paradigms. Formally, units are Multi Input Multi Output (MIMO) Finite State Machines (FSM). A program in Unit is represented as a Graph.

The Unit Programming Language was developed in close junction to the Unit Programming Environment, which is a Web application built for easy composition of new units. The environment is designed to feel visual and kinesthetic, giving the perception of Direct Manipulation of Live Virtual Objects. The Unit Programming experience is minimalistic, ergonomic, mobile, and can be performed through a variety of input devices, and editing can be partially expressed through Drawing, Gesture and Voice.

Unit is similar in concept to the Unix shell. For a seasoned programmer, Unit should feel like the 2D evolution of the Command Line Interface (CLI) where units' (commands) inputs and outputs (stdin/stdout/stderr) can be piped together into a graph (script). In fact, Unit can be thought of as a re-

exploration of the Unix Philosophy in the context of modern Software Engineering and Web Development.

The Unit system is set to grow into a simple and approachable Web Operating System, which is sandboxed by default, enabling a new era of Software Sharing and Collaboration, unifying the currently fragmented Computer Experience.

For an introduction to Unit Programming, visit [Getting Started](#).

The broader design philosophy behind Unit is discussed in [Concept](#).

To jump right into the official Unit Programming Environment, visit [unit.land](#) (beta).

Check out a collection of public open source units at [unit.tools](#) (beta).

Development

Install npm and git if you haven't already.

Clone this repository:

```
1 git clone git@github.com:samueltimbo/unit.git
```

Get into the unit folder:

```
1 cd unit
```

Install:

```
1 npm install
```

Setup:

```
1 npm run setup
```

Start a local unit server:

```
1 npm start
```

For development mode:

```
1 npm run watch
```

Library

Unit source code can be used as a general purpose library in any JavaScript project, both on Web and Node.js.

```
1 npm install --save @_unit/unit
```

To install unit globally:

```
1 npm install --global @_unit/unit
```

Then to start a local server:

```
1 unit
```

To render a unit JSON bundle:

```
1 import { renderBundle } from '@_unit/unit/lib/client/platform/web/
  render'
2
3 const root = document.getElementById('root')
4
5 const bundle = {
6   spec: {
7     units: {
8       helloworld: {
9         id: 'a9cbcd12-9a53-11eb-8c2e-f3146b36128d',
10       },
11     },
12   },
13 }
14
15 const [system, graph] = renderBundle(root, bundle)
16
17 const helloworld = graph.getUnit('helloworld')
18
19 helloworld.push('style', {
20   color: '#ffdd00',
21 })
```

To boot a system and instantiate a new unit on it:

```
1 import { boot } from '@_unit/unit/lib/client/platform/node/boot'
2 import { fromBundle } from '@_unit/unit/lib/spec/fromBundle'
3 import _specs from '@_unit/unit/lib/system/_specs'
4
5 export const [system] = boot()
6
7 const bundle = require('./MergeSort.json')
8
9 const MergeSort = fromBundle(bundle, _specs, {})
10
11 const mergeSort = new MergeSort(system)
12
13 mergeSort.play()
```

```
14
15 mergeSort.push('a', [2, 1])
16
17 console.log(mergeSort.take('a')) // [1, 2]
```

Test

```
1 npm test
```

Links

List of related web applications (beta):

unit.md

Redirect to the official unit source code.

unit.land

Fully local version of the Unit Environment.

Any *.unit.land subdomain represents a completely isolated instance of the Unit system.

unit.tools

Marketplace to explore, collect and play with live visual programs built in Unit.

unit.moe

Appending #url to https://unit.moe will attempt to encapsulate the unit served at that URL in a graph editor. This is helpful for debugging an open unit website, for instance:

https://unit.moe#https://unit.land will show the code for https://unit.land.

https://unit.moe#https://unit.tools will show the code for https://unit.tools.

Documentation

The unit Programming Environment was designed to be learnable by Experimentation and Exploration. Visual and written documentation about each unit can be accessed using Info Mode.

This codebase is a TypeScript implementation of the unit Programming Language Spec with an accompanying Web based Programming Environment. I tried to make it as clean and organized as I could while still maintaining a space for experimentation. Some folders contain a [README.md](#) file with information about that piece of the architecture.

Contributing

First of all, please feel free to clone or fork Unit's source code.

This is the best place to learn about Unit's design and implementation, and the official channel for Open Source collaboration.

This repository is open for pull requests from contributors.

If you want to collaborate with Unit's development. Please hit me up at [@io_sammt](#).

If you have a feature idea or hit a malfunction, this is a workflow I like to follow:

1. Iterate over a solution on top the latest main, testing it locally until it works and the code is clean;
2. Open a pull request explaining the problem and solution;
3. Iterate over it to make sure the change is positive and aligned with Unit's design and vision.

The Unit project has nearly infinite space for creative freedom, invention, and intimacy, which is what motivated it.

Ultimately, the goal is for everyone to have an easy to use, flexible and powerful visual system where one can build anything with absolute freedom, without ever having to leave it.

Backlog

Unit system API is composed of logical, platform (frontend and backend) and "operating system" units.

The direction of growth right now is ensuring a robust basic UI/UX and feature parity with Web and a subset of NodeJS (not native to the Web), like TCP and HTTP.

Visit [Evolution](#) to learn more about where Unit development is headed.

Community

If you have a question or a suggestion related to Unit, or if you just want to talk about the future of Programming, or if you wish to join Unit team, feel free to message me at [@io_sammt](#).

For weekly updates about Unit's development, including thoughts about vision and roadmap, follow us on Twitter [@io_unit](#).

We hope you enjoy Unit and invite you to be part of our community!

License

MIT License

Copyright 2021 UNIT IO, Inc