

July 2, 2018

# Data Authoring Environments

## An Overview

Hugh Dubberly, Cameron Burgess  
Dubberly Design Office

# Introduction

This presentation captures over 200 examples of computer software and hardware interfaces for authoring data and programs from the 1960s through the present day.

It is presented in eight sections. Additionally, some of the frequently occurring design patterns are cataloged in the last section.

- 1 Origins
- 2 Bits to Atoms
- 3 Design Machine
- 4 Visual Interfaces
- 5 IDE (Environments & Experiments)
- 6 Learning From/Within Cyberspace
- 7 Remaking Reality
- Design Patterns

# Table of Contents

## 1 Origins Starting with "Strings"

- 1.1 Text Editors
- 1.2 Notebooks
- 1.3 Card-Stack / Kit-of-Parts
- 1.4 'Bureaucracy Processing'

## 2 Bits to Atoms Computation meets Business & Industry

- 2.1 Spreadsheets
- 2.2 Data Analysis
- 2.3 Simulation
- 2.4 3D (Solid Geometry) Modeling

## 3 Design Machine Creative Tools for Everyone

- 3.1 Graphics
- 3.2 Presentation
- 3.3 Timeline-Based Applications
- 3.4 World Wide Web
- 3.5 'Creative Coding' Toolkits
- 3.6 Digital Drawing

## 4 Visual Interfaces Across the (Design-Development) Divide

- 4.1 Node-Based (Graph)
- 4.2 Interaction Prototyping
- 4.3 Prototyping & Development Tools
- 4.4 Design-Development Toolchain Integration
- 4.5 'Low Code' Environments
- 4.6 Dynamic Modeling Tools

## 5 IDE: Environments & Experiments What Is & What's Next

- 5.1 Traditional IDEs
- 5.2 Structure Editing
- 5.3 Work Flow
- 5.4 Visible State Editors
- 5.5 Dynamic Diagram Experiments

## 6 Learning From/Within Cyberspace How Game-Tech is enabling the Next Wave

- 6.1 3D (Mesh/NURBs) Creation Suites
- 6.2 Learnable Programming
- 6.3 Game Authoring
- 6.4 Games with Authoring
- 6.5 Games as Authoring

## 7 Remaking Reality Measure, Simulate, Predict & Act

- 7.1 Reality Design
- 7.2 Local Simulation
- 7.3 Global View

## – Design Patterns Preliminary Sketches

# Origins

## Starting with “Strings”

# Text Editors

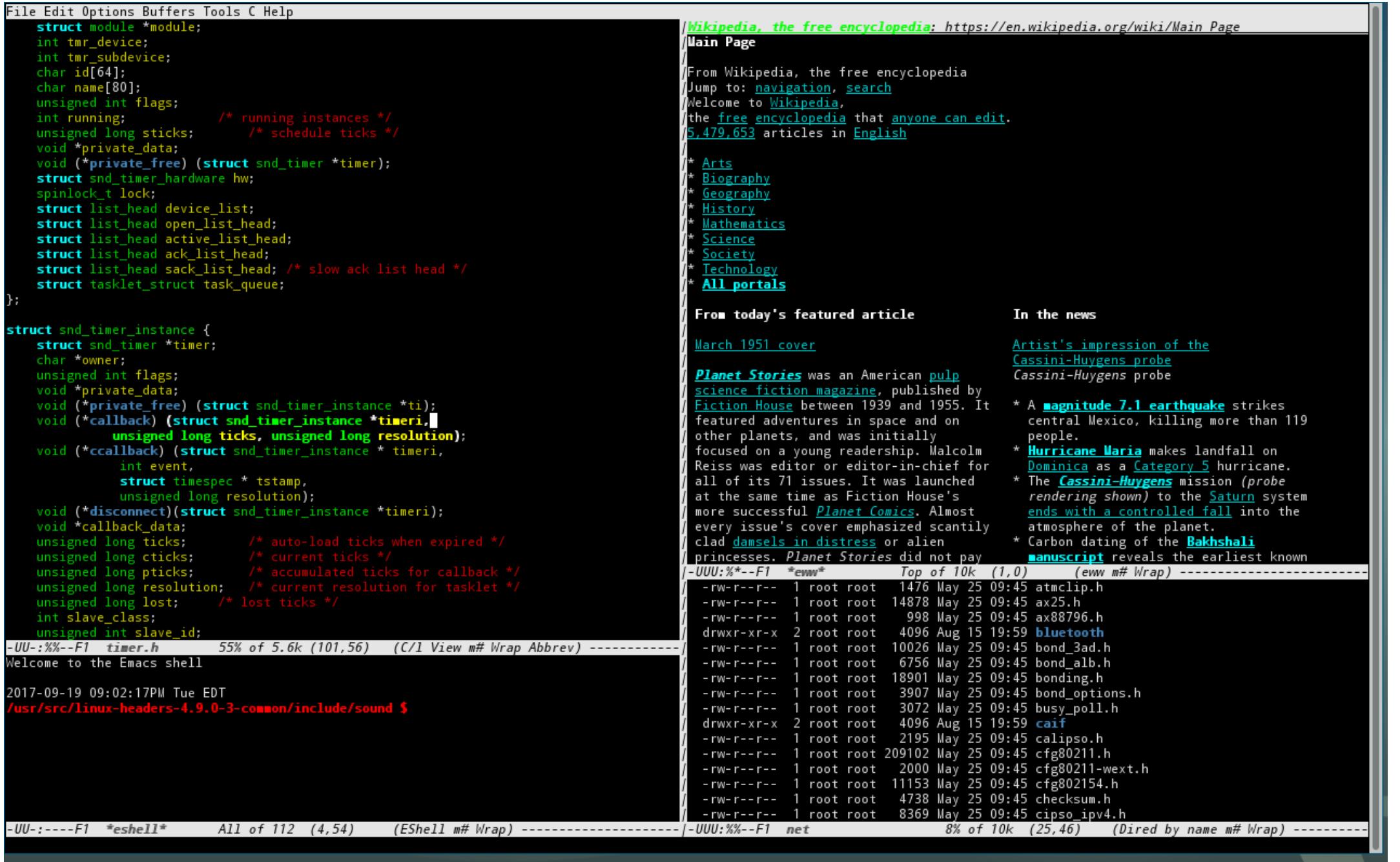
```
+8      +16     +24     +32     +40     +48     +56     +64     +72     +80
!
'x,.n
13
14 The principle difference between line editors and display editors
15 is that display editors provide instant feedback to user commands,
16 whereas line editors require sometimes lengthy input before any
17 effects are seen. The advantage of instant feedback, of course,
18 is that if a mistake is made, it can be corrected immediately,
19 before more damage is done. Editing in 'ed' requires more strategy
20 and forethought; but if you are up to the task, it can be quite
21 efficient.

14s/p
```

Ken Thompson

# Emacs

1976-Today



The screenshot shows an Emacs window with several buffers open. The leftmost buffer is a C source code file named `timer.h`, showing declarations for `struct module` and `struct snd_timer_instance`. The rightmost buffer is a web browser displaying the Wikipedia main page. Between these are two smaller buffers: one showing a file listing and another showing a directory listing.

**File Edit Options Buffers Tools C Help**

```
struct module *module;
int tmr_device;
int tmr_subdevice;
char id[64];
char name[80];
unsigned int flags;
int running; /* running instances */
unsigned long sticks; /* schedule ticks */
void *private_data;
void (*private_free) (struct snd_timer *timer);
struct snd_timer_hardware hw;
spinlock_t lock;
struct list_head device_list;
struct list_head open_list_head;
struct list_head active_list_head;
struct list_head ack_list_head;
struct list_head sack_list_head; /* slow ack list head */
struct tasklet_struct task_queue;
};

struct snd_timer_instance {
    struct snd_timer *timer;
    char *owner;
    unsigned int flags;
    void *private_data;
    void (*private_free) (struct snd_timer_instance *ti);
    void (*callback) (struct snd_timer_instance *timeri, unsigned long ticks, unsigned long resolution);
    void (*ccallback) (struct snd_timer_instance * timeri, int event, struct timespec * tstamp, unsigned long resolution);
    void (*disconnect)(struct snd_timer_instance *timeri);
    void *callback_data;
    unsigned long ticks; /* auto-load ticks when expired */
    unsigned long cticks; /* current ticks */
    unsigned long pticks; /* accumulated ticks for callback */
    unsigned long resolution; /* current resolution for tasklet */
    unsigned long lost; /* lost ticks */
    int slave_class;
    unsigned int slave_id;

```

-UU:%%--F1 timer.h 55% of 5.6k (101,56) (C/I View ## Wrap Abbrev) -----

Welcome to the Emacs shell

2017-09-19 09:02:17PM Tue EDT

/usr/src/linux-headers-4.9.0-3-common/include/sound \$

**Wikipedia, the free encyclopedia: [https://en.wikipedia.org/wiki/Main\\_Page](https://en.wikipedia.org/wiki/Main_Page)**

Main Page

From Wikipedia, the free encyclopedia

Jump to: [navigation](#), [search](#)

Welcome to [Wikipedia](#),  
the [free encyclopedia](#) that [anyone can edit](#).  
5,479,653 articles in [English](#)

\* [Arts](#)  
\* [Biography](#)  
\* [Geography](#)  
\* [History](#)  
\* [Mathematics](#)  
\* [Science](#)  
\* [Society](#)  
\* [Technology](#)  
\* [All portals](#)

**From today's featured article**

[March 1951 cover](#)

**Planet Stories** was an American [pulp](#) science fiction [magazine](#), published by [Fiction House](#) between 1939 and 1955. It featured adventures in space and on other planets, and was initially focused on a young readership. Malcolm Reiss was editor or editor-in-chief for all of its 71 issues. It was launched at the same time as Fiction House's more successful [Planet Comics](#). Almost every issue's cover emphasized scantily clad [damsels in distress](#) or alien princesses. *Planet Stories* did not pay

**In the news**

[Artist's impression of the Cassini-Huygens probe](#)  
Cassini-Huygens probe

\* A [magnitude 7.1 earthquake](#) strikes central Mexico, killing more than 119 people.  
\* [Hurricane Maria](#) makes landfall on [Dominica](#) as a [Category 5](#) hurricane.  
\* The [Cassini-Huygens](#) mission (probe rendering shown) to the [Saturn](#) system [ends with a controlled fall](#) into the atmosphere of the planet.  
\* Carbon dating of the [Bakhshali manuscript](#) reveals the earliest known

-UU:%%--F1 \*eww\* Top of 10k (1,0) (eww ## Wrap) -----

-rw-r--r--	1	root	1476 May 25 09:45 atmclip.h
-rw-r--r--	1	root	14878 May 25 09:45 ax25.h
-rw-r--r--	1	root	998 May 25 09:45 ax88796.h
drwxr-xr-x	2	root	4096 Aug 15 19:59 bluetooth
-rw-r--r--	1	root	10026 May 25 09:45 bond_3ad.h
-rw-r--r--	1	root	6756 May 25 09:45 bond_alb.h
-rw-r--r--	1	root	18901 May 25 09:45 bonding.h
-rw-r--r--	1	root	3907 May 25 09:45 bond_options.h
-rw-r--r--	1	root	3072 May 25 09:45 busy_poll.h
drwxr-xr-x	2	root	4096 Aug 15 19:59 caif
-rw-r--r--	1	root	2195 May 25 09:45 calipso.h
-rw-r--r--	1	root	209102 May 25 09:45 cfg80211.h
-rw-r--r--	1	root	2000 May 25 09:45 cfg80211-wext.h
-rw-r--r--	1	root	11153 May 25 09:45 cfg802154.h
-rw-r--r--	1	root	4738 May 25 09:45 checksum.h
-rw-r--r--	1	root	8369 May 25 09:45 cipso_ipv4.h

-UU:---F1 \*eshell\* All of 112 (4,54) (Eshell ## Wrap) -----

-UU:%%--F1 net 8% of 10k (25,46) (Dired by name ## Wrap) -----

David A. Moon and Guy L. Steele Jr.

```
// These two lines are required to initialize Express in Cloud Code.
var express = require('express');
var app = express();

// Global app configuration section
app.set('views', 'cloud/views'); // Specify the folder to find templates
app.set('view engine', 'ejs'); // Set the template engine
app.use(express.bodyParser()); // Middleware for reading request body

// This is an example of hooking up a request handler with a specific request
// path and HTTP verb using the Express routing API.
app.get('/hello', function(req, res) {
  res.render('hello', { message: 'Congrats, you just set up your app!' });
});

// // Example reading from the request query string of an HTTP get request.
// app.get('/test', function(req, res) {
//   // GET http://example.parseapp.com/test?message=hello
//   res.send(req.query.message);
// });

// // Example reading from the request body of an HTTP post request.
// app.post('/test', function(req, res) {
//   // POST http://example.parseapp.com/test (with request body "message=hello")
//   res.send(req.body.message);
// });

// Attach the Express app to Cloud Code.
app.listen();
~

~

~

~

~

~

'require' was used before it was defined.
```

Bram Moolenaar

# Sublime Text

# 2008-Today

GROUP 1

- main.cpp — Users/.../sa
- USBCore.cpp
- wiring\_digital.c

GROUP 2

- GDB Disassembly

GROUP 3

- GDB Callstack
- GDB Breakpoints

GROUP 4

- GDB Threads
- GDB Console
- GDB Session

FOLDERS

- sublime-text
- .pioenvs
- lib
- src
- main.cpp
- .gitignore
- platformio.ini
- platformio.sublime

main USBCore.cpp x wiring\_digital.c x

```
47     // Enable pull level (cf '22.6.3.2 I
48     PORT->Group[g_APinDescription[ulPin]
49     break ;
50
51     case INPUT_PULLDOWN:
52         // Set pin to input mode with pull-d
53         PORT->Group[g_APinDescription[ulPin]
54         PORT->Group[g_APinDescription[ulPin]
55
56         // Enable pull level (cf '22.6.3.2 I
57         PORT->Group[g_APinDescription[ulPin]
58         break ;
59
60     case OUTPUT:
61         // enable input, to support reading
62         PORT->Group[g_APinDescription[ulPin]
63
64         // Set pin to output mode
65         PORT->Group[g_APinDescription[ulPin]
66         break ;
67
68     default:
69         // do nothing
70         break ;
71     }
72 }
73
74 void digitalWrite( uint32_t ulPin, uint32_t
75 {
76     // Handle the case the pin isn't usable
77     if ( g_APinDescription[ulPin].ulPinType
78     {
79         return ;
80     }
81 }
```

GDB Disassembly x

```
1 /Users/ikravets/.platformio/packages/frame
2 0x00004c86: movs    r4, #1
3 0x00004c88: adds    r6, r4, #0
4 0x00004c8a: lsls    r6, r5
5 0x00004c8c: str r6, [r3, #4]
6 /Users/ikravets/.platformio/packages/frame
7 /Users/ikravets/.platformio/packages/frame
8 /Users/ikravets/.platformio/packages/frame
9 0x00004c8e: str r6, [r3, #20]
10 /Users/ikravets/.platformio/packages/frame
11 0x00004c90: b.n 0x4cc0 <pinMode+220>
12 /Users/ikravets/.platformio/packages/frame
13 /Users/ikravets/.platformio/packages/frame
14 /Users/ikravets/.platformio/packages/frame
15 /Users/ikravets/.platformio/packages/frame
16 0x00004c92: ldr r3, [pc, #48] ; (0x4cc4
17 0x00004c94: lsls    r2, r0, #1
18 0x00004c96: adds    r1, r2, r0
19 0x00004c98: lsls    r1, r1, #3
20 0x00004c9a: ldrrsb  r5, [r3, r1]
21 0x00004c9c: adds    r1, r2, r0
22 0x00004c9e: lsls    r1, r1, #3
23 0x00004ca0: adds    r1, r3, r1
24 0x00004ca2: ldr r4, [r1, #4]
25 0x00004ca4: lsls    r1, r5, #7
26 0x00004ca6: ldr r5, [pc, #32] ; (0x4cc8
27 0x00004ca8: adds    r1, r1, r5
28 0x00004caa: adds    r4, r1, r4
29 0x00004cac: adds    r4, #64 ; 0x40
30 0x00004cae: movs    r5, #2
31 0x00004cb0: strb    r5, [r4, #0]
32 /Users/ikravets/.platformio/packages/frame
33 /Users/ikravets/.platformio/packages/frame
34 /Users/ikravets/.platformio/packages/frame
35 0x00004c92: ldr r3, [pc, #48] ; (0x4cc4
```

GDB Callstack x GDB Breakpoints x

```
1 pinMode(ulPin = 31,ulMode = 1,);
2 USBDeviceClass::init(this = 0x200006b0 <
3     USBDevice>,this@entry = 0x200006b0 <
4     USBDevice>);
5 main();
```

GDB Threads x GDB Console x

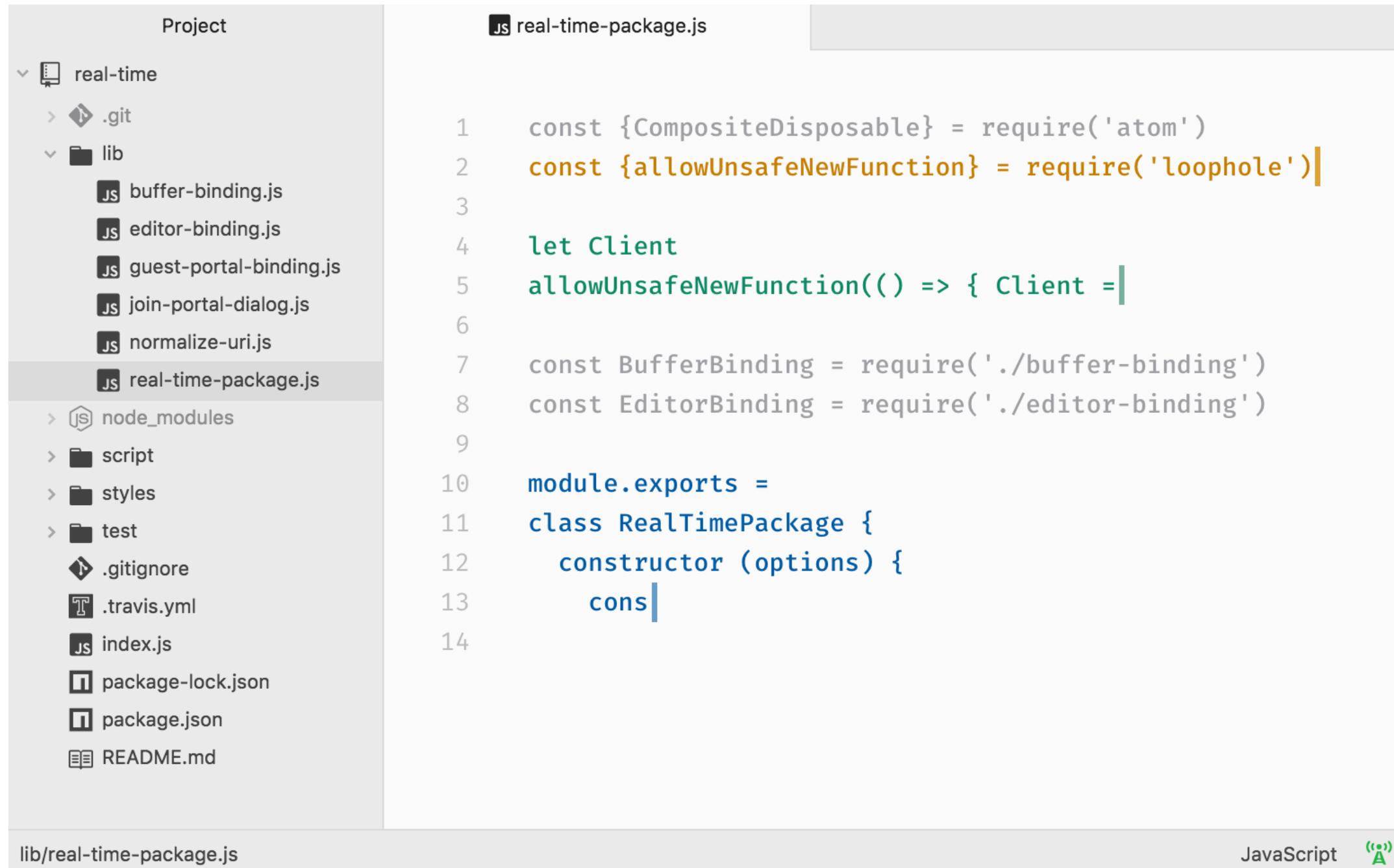
```
1 001 - stopped - pinMode(ulPin = 31,
2     ulMode = 1);
```

GDB

Jon Skinner, Will Bond

# Atom

2014-Today



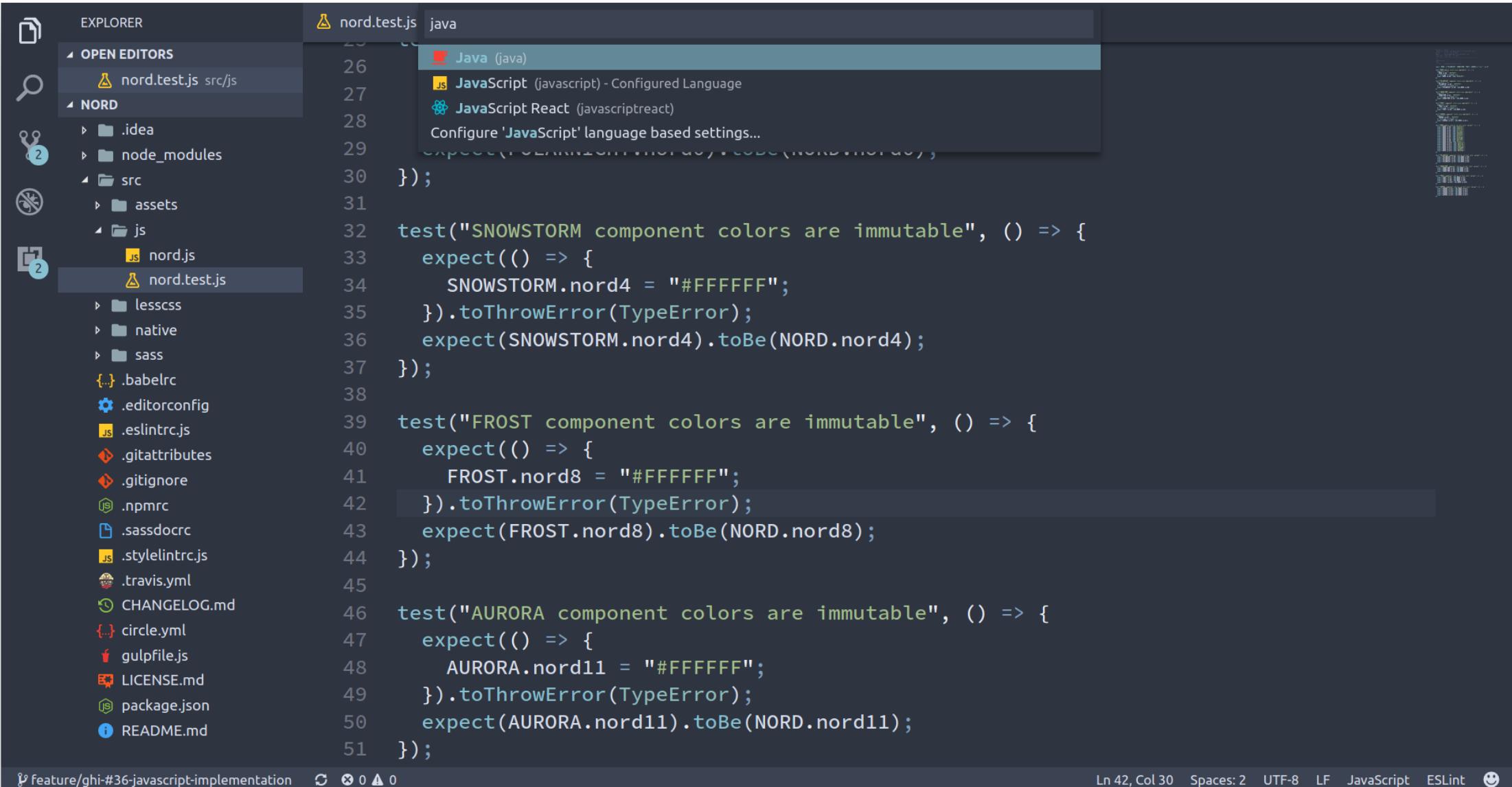
The screenshot shows the Atom code editor interface. On the left is the 'Project' sidebar with a tree view of the project structure. The 'real-time' folder is expanded, showing subfolders '.git', 'lib', and 'script'. The 'lib' folder contains several files: 'buffer-binding.js', 'editor-binding.js', 'guest-portal-binding.js', 'join-portal-dialog.js', 'normalize-uri.js', and 'real-time-package.js'. The 'real-time-package.js' file is selected and shown in the main editor area. The code is written in JavaScript and defines a 'RealTimePackage' class. The 'allowUnsafeNewFunction' line is highlighted in orange, and the 'cons' line is highlighted in blue, indicating code completion suggestions. The status bar at the bottom shows 'lib/real-time-package.js' and 'JavaScript'.

```
1  const {CompositeDisposable} = require('atom')
2  const {allowUnsafeNewFunction} = require('loophole')
3
4  let Client
5  allowUnsafeNewFunction(() => { Client =
6
7  const BufferBinding = require('./buffer-binding')
8  const EditorBinding = require('./editor-binding')
9
10 module.exports =
11 class RealTimePackage {
12   constructor (options) {
13     cons
14 }
```

GitHub

# Visual Studio Code

2015-Today



A screenshot of the Visual Studio Code interface. The Explorer sidebar on the left shows a file tree with a file named 'nord.test.js' selected. The Editor tab bar at the top has 'nord.test.js' and 'java' tabs, with 'java' currently active. The main editor area displays a block of JavaScript code for testing component colors. A language selector dropdown is open over the code, showing options: 'Java (java)', 'JavaScript (javascript) - Configured Language', and 'JavaScript React (javascriptreact)'. The status bar at the bottom shows the file path 'feature/ghi-#36-javascript-implementation', line and column numbers 'Ln 42, Col 30', and various code analysis tools like 'Spaces: 2', 'UTF-8', 'LF', 'JavaScript', and 'ESLint'.

```
26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51
```

```
Java (java)
JavaScript (javascript) - Configured Language
JavaScript React (javascriptreact)
Configure 'JavaScript' language based settings...
expect(FROST.nord8).not.toBe(NORD.nord8),
});

test("SNOWSTORM component colors are immutable", () => {
  expect(() => {
    SNOWSTORM.nord4 = "#FFFFFF";
  }).toThrowError(TypeError);
  expect(SNOWSTORM.nord4).toBe(NORD.nord4);
});

test("FROST component colors are immutable", () => {
  expect(() => {
    FROST.nord8 = "#FFFFFF";
  }).toThrowError(TypeError);
  expect(FROST.nord8).toBe(NORD.nord8);
});

test("AURORA component colors are immutable", () => {
  expect(() => {
    AURORA.nord11 = "#FFFFFF";
  }).toThrowError(TypeError);
  expect(AURORA.nord11).toBe(NORD.nord11);
});
```

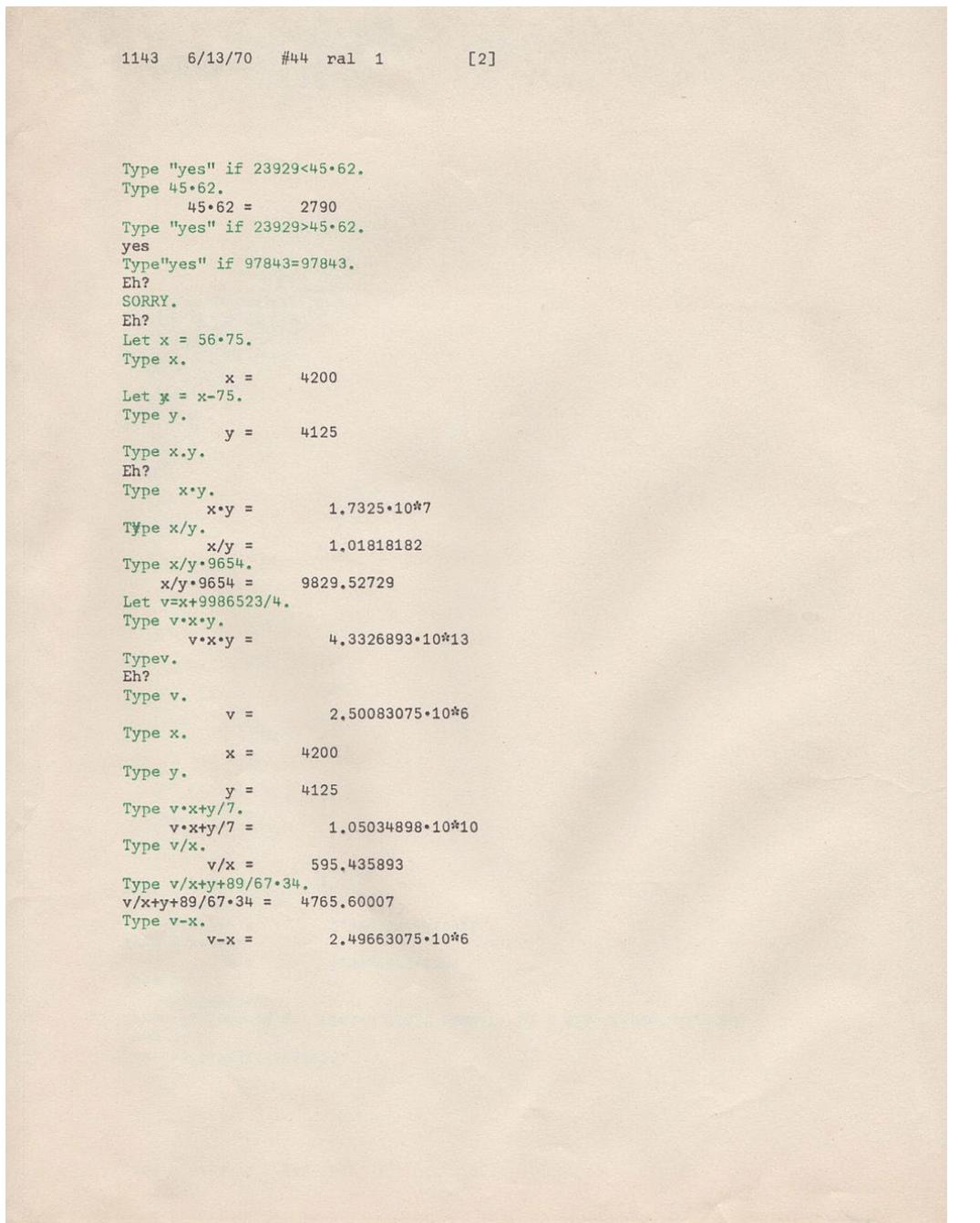
Microsoft

# Notebooks

## Query & Response Environments

# JOSS

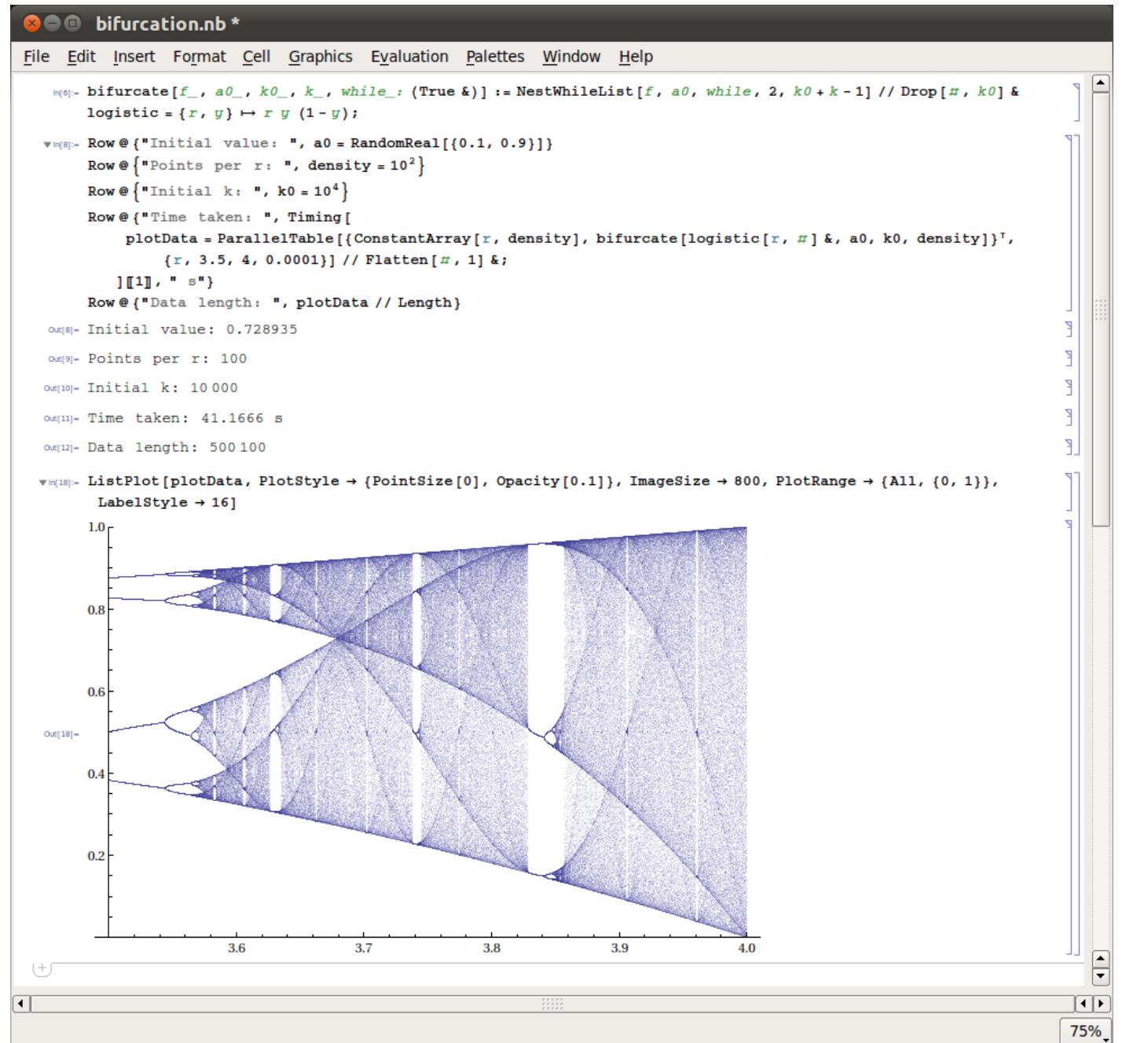
# 1966



RAND Corporation

# Mathematica

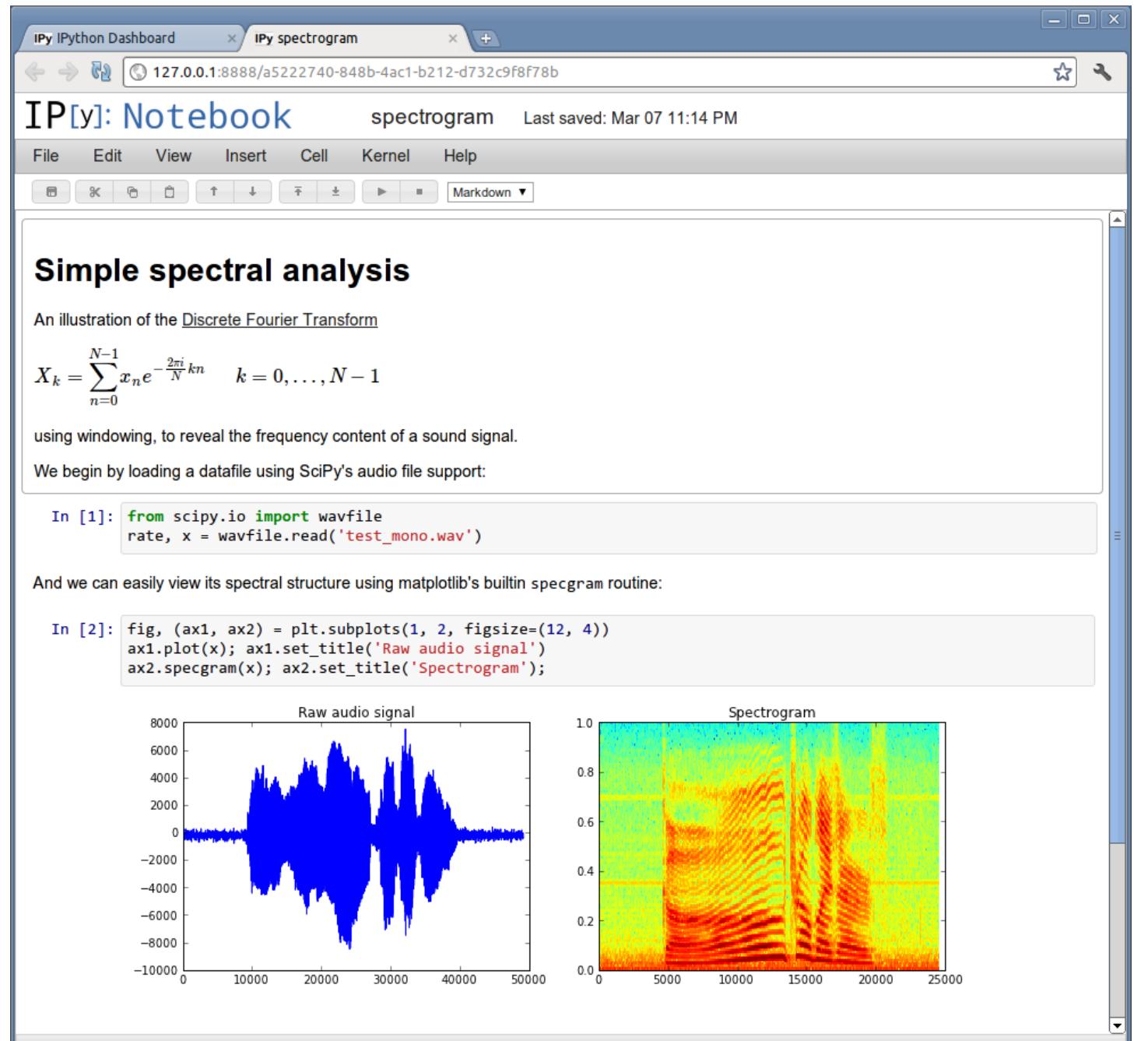
1988-Today



Stephen Wolfram

# iPython

2001-2014



Fernando Pérez

# Jupyter

2014-Today

jupyter Custom Display Logic (unsaved changes) [Logout](#)

File Edit View Insert Cell Kernel Widgets Help

Not Trusted Python 3

In [4]: `x = Gaussian(2.0, 1.0)`  
x

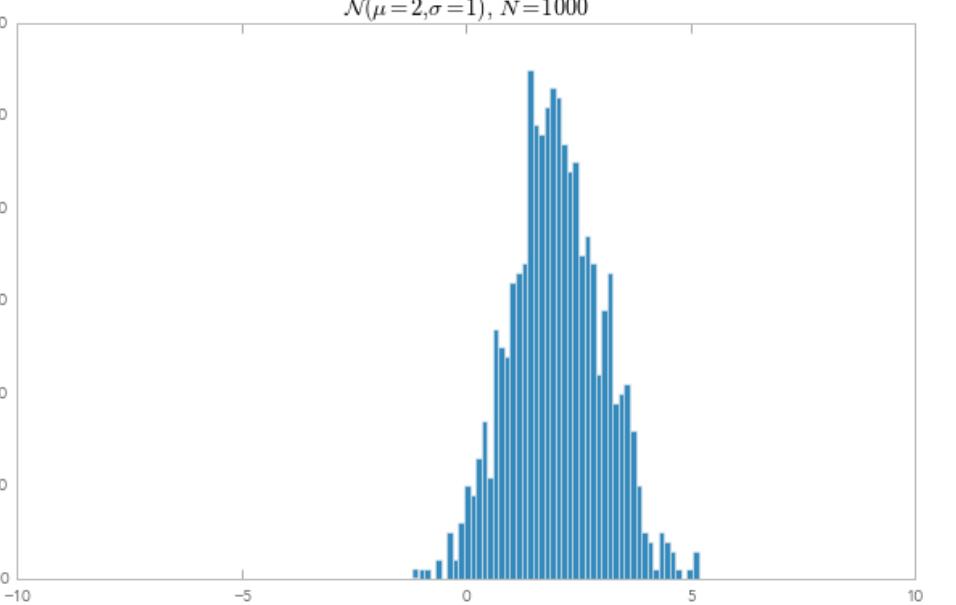
Out[4]:  $\mathcal{N}(\mu = 2, \sigma = 1), N = 1000$

You can also pass the object to the `display` function to display the default representation:

In [5]: `display(x)`  
 $\mathcal{N}(\mu = 2, \sigma = 1), N = 1000$

Use `display_png` to view the PNG representation:

In [6]: `display_png(x)`



Fernando Pérez and Others

# Tonic/RunKit

2015-Today

**Data Visualizing**

node 0.12 · version: master · publish

```
1 new Buffer("Hello!");
```

Buffer (6 bytes) Data Explorer

Offset	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	ASCII (7-bit)
0	48	65	6C	6C	6F	21	00	00	00	00	00	00	00	00	00	00	Hello!...
10	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
20	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
30	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
40	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
50	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
60	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
70	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
80	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
90	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
100	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....

2

saving

**Data Visualizing**

node 0.12 · version: master · publish

```
1 var coords = {latitude: 37.773972, longitude: -122.431297};
```

Object Map Coordinates

Leaflet | Map data © OpenStreetMap contributors, CC-BY-SA, Imagery © Mapbox

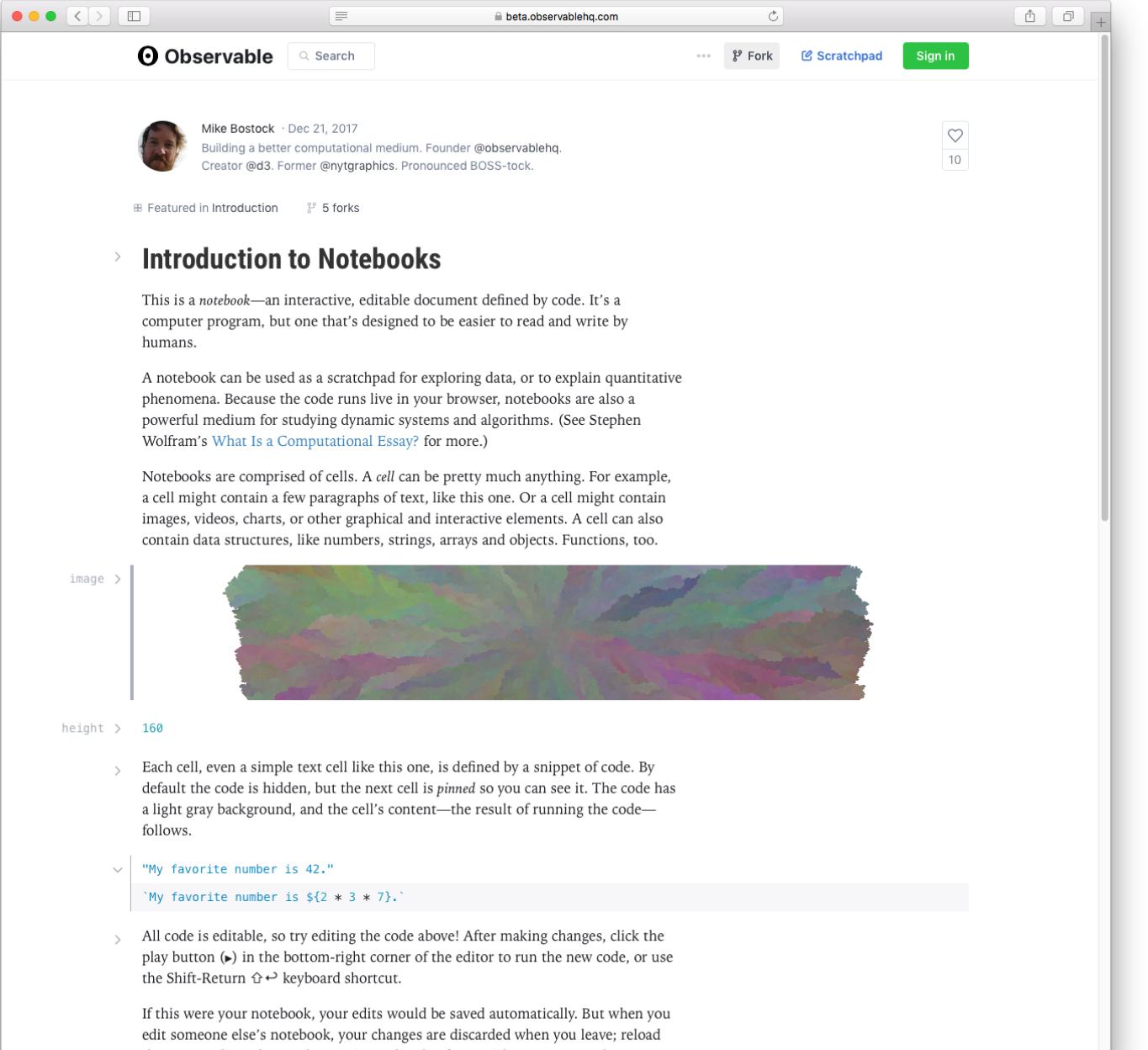
2

saving

Francisco Tolmasky

# Observable

2018-Today



A screenshot of the Observable web interface. The page title is "observablehq.com". The main content is a notebook by Mike Bostock, dated Dec 21, 2017. The notebook title is "Introduction to Notebooks". The content discusses what a notebook is, how it can be used as a scratchpad, and its components. It includes a section on cells, a code editor with a snippet of Python code, and a note about editing and running code. The interface has a clean, modern design with a sidebar on the left.

Mike Bostock · Dec 21, 2017  
Building a better computational medium. Founder @observablehq. Creator @d3. Former @nytgraphics. Pronounced BOSS-tock.

Featured in Introduction 5 forks

## Introduction to Notebooks

This is a *notebook*—an interactive, editable document defined by code. It's a computer program, but one that's designed to be easier to read and write by humans.

A notebook can be used as a scratchpad for exploring data, or to explain quantitative phenomena. Because the code runs live in your browser, notebooks are also a powerful medium for studying dynamic systems and algorithms. (See Stephen Wolfram's [What Is a Computational Essay?](#) for more.)

Notebooks are comprised of cells. A *cell* can be pretty much anything. For example, a cell might contain a few paragraphs of text, like this one. Or a cell might contain images, videos, charts, or other graphical and interactive elements. A cell can also contain data structures, like numbers, strings, arrays and objects. Functions, too.

image > 

height > 160

Each cell, even a simple text cell like this one, is defined by a snippet of code. By default the code is hidden, but the next cell is *pinned* so you can see it. The code has a light gray background, and the cell's content—the result of running the code—follows.

"My favorite number is 42."  
My favorite number is \${2 \* 3 \* 7},`

All code is editable, so try editing the code above! After making changes, click the play button (▶) in the bottom-right corner of the editor to run the new code, or use the Shift-Return ↵ keyboard shortcut.

If this were your notebook, your edits would be saved automatically. But when you edit someone else's notebook, your changes are discarded when you leave; reload the page to discard your changes immediately. If you wish to save your changes,

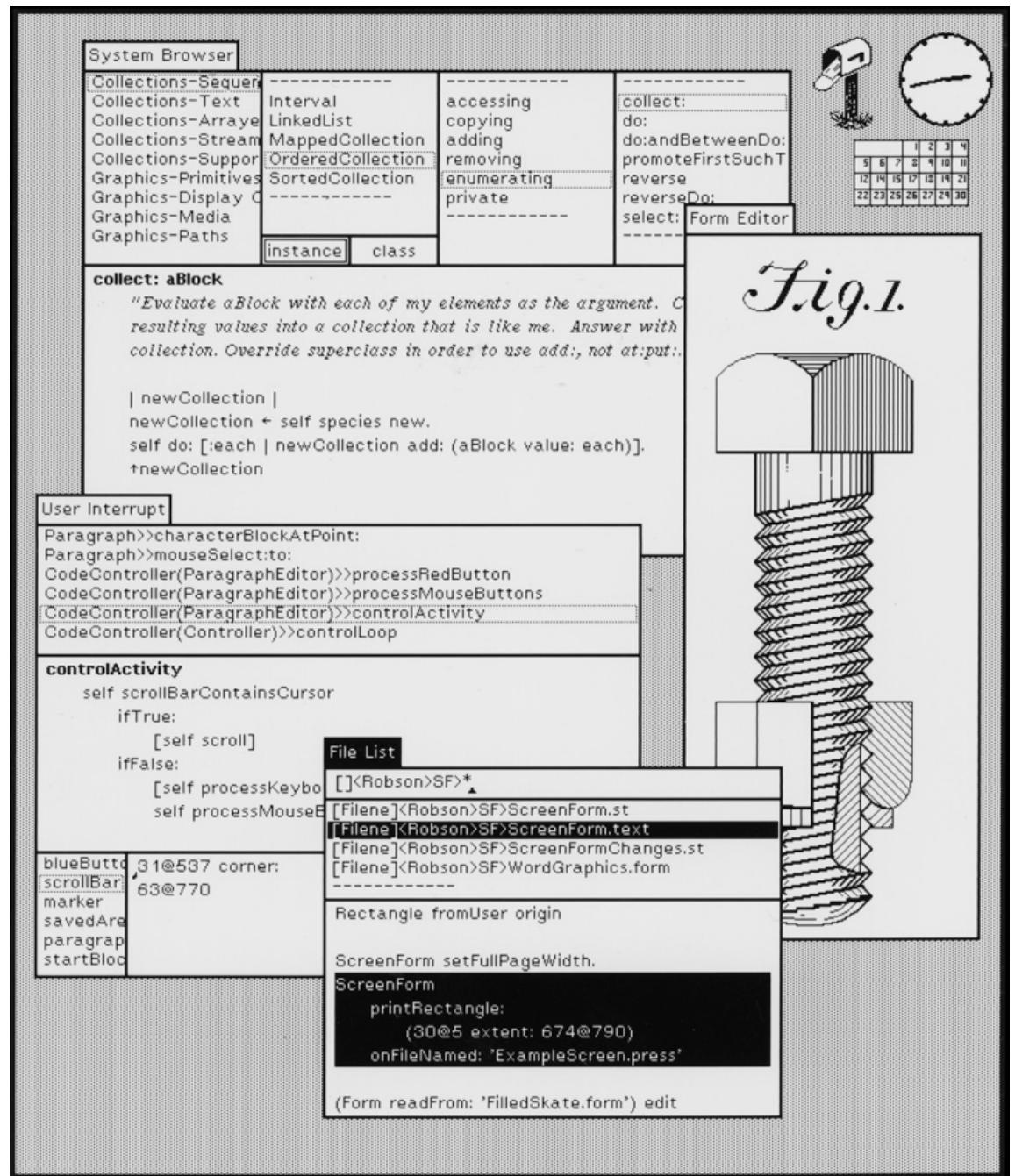
Mike Bostock, Tom MacWright, Jeremy Ashkenas for Observable, Inc.

# **Card-Stack / Kit-of-Parts**

## **'Construction Sets'**

# SmallTalk

1972-1980



Alan Kay, Dan Ingalls, Adele Goldberg, Ted Kaehler, Diana Merry, Scott Wallace, Peter Deutsch at XEROX PARC

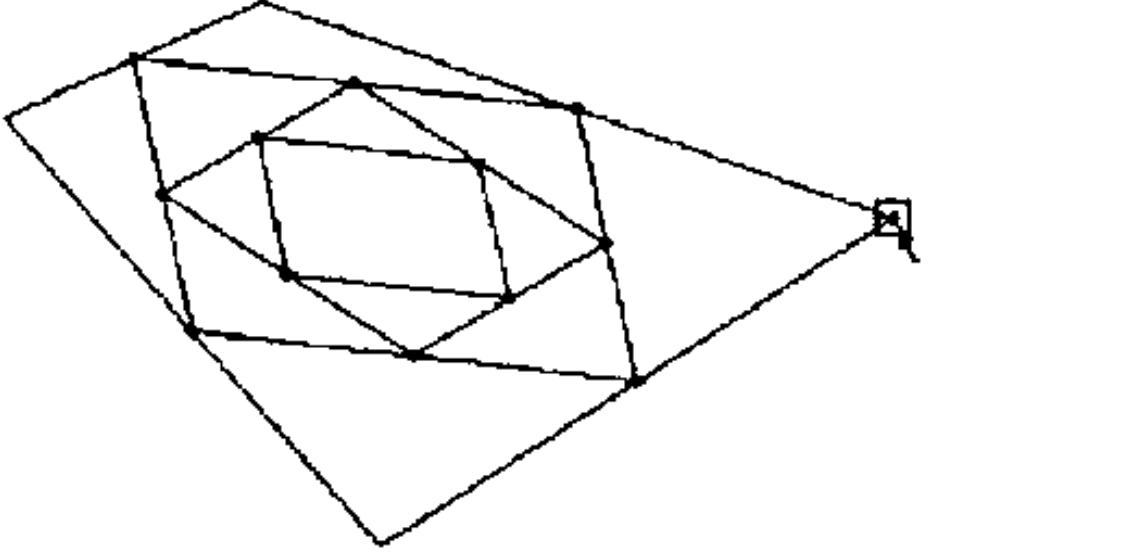
# ThingLab

1977

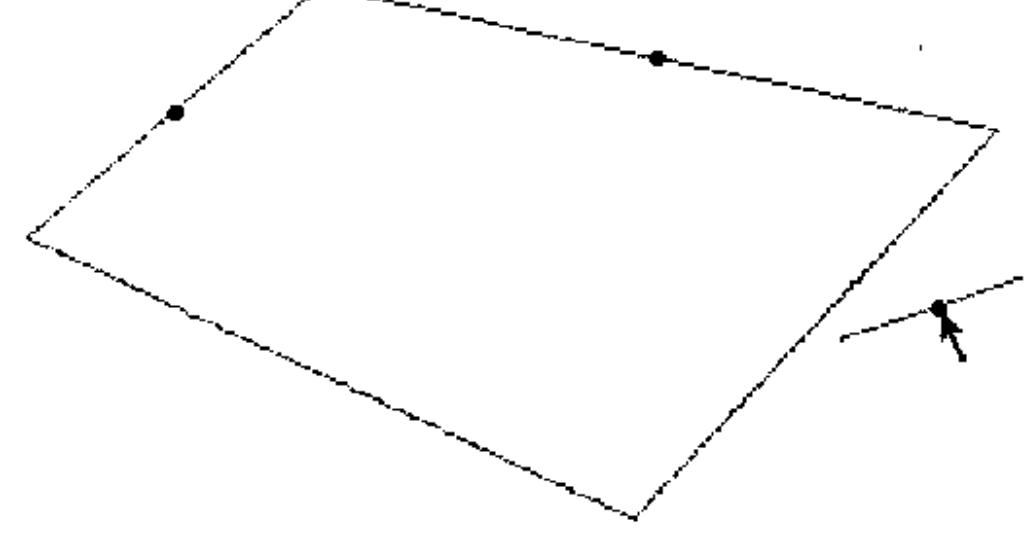
ThingLab Browser

Anchor	picture structure values	constrain merge move edit	LineSegment MidPointLine Node Plus Point Printer Rectangle
BitImage			
ConstantLengthL			
Ex2			
LineSegment			

**prefer** **require**



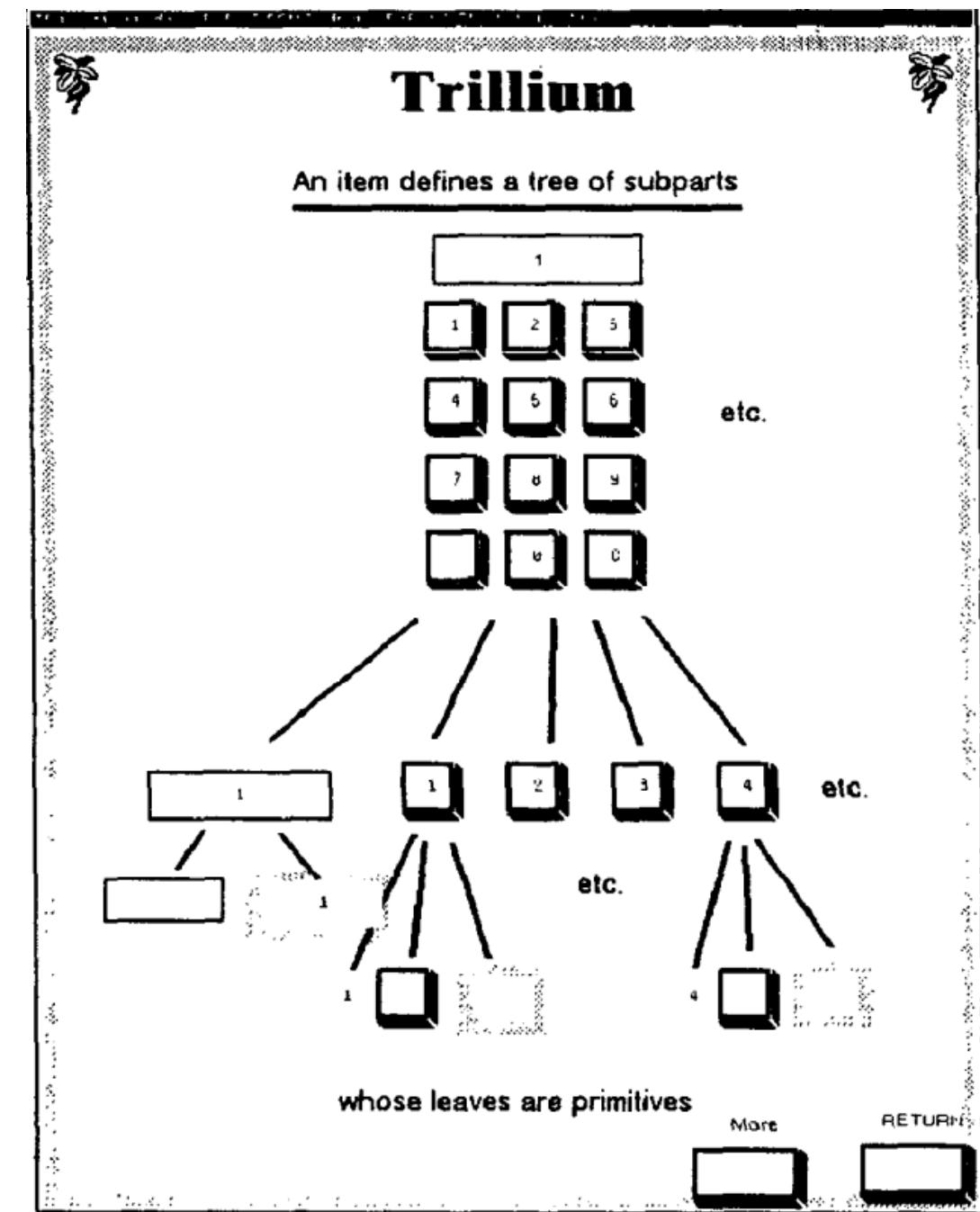
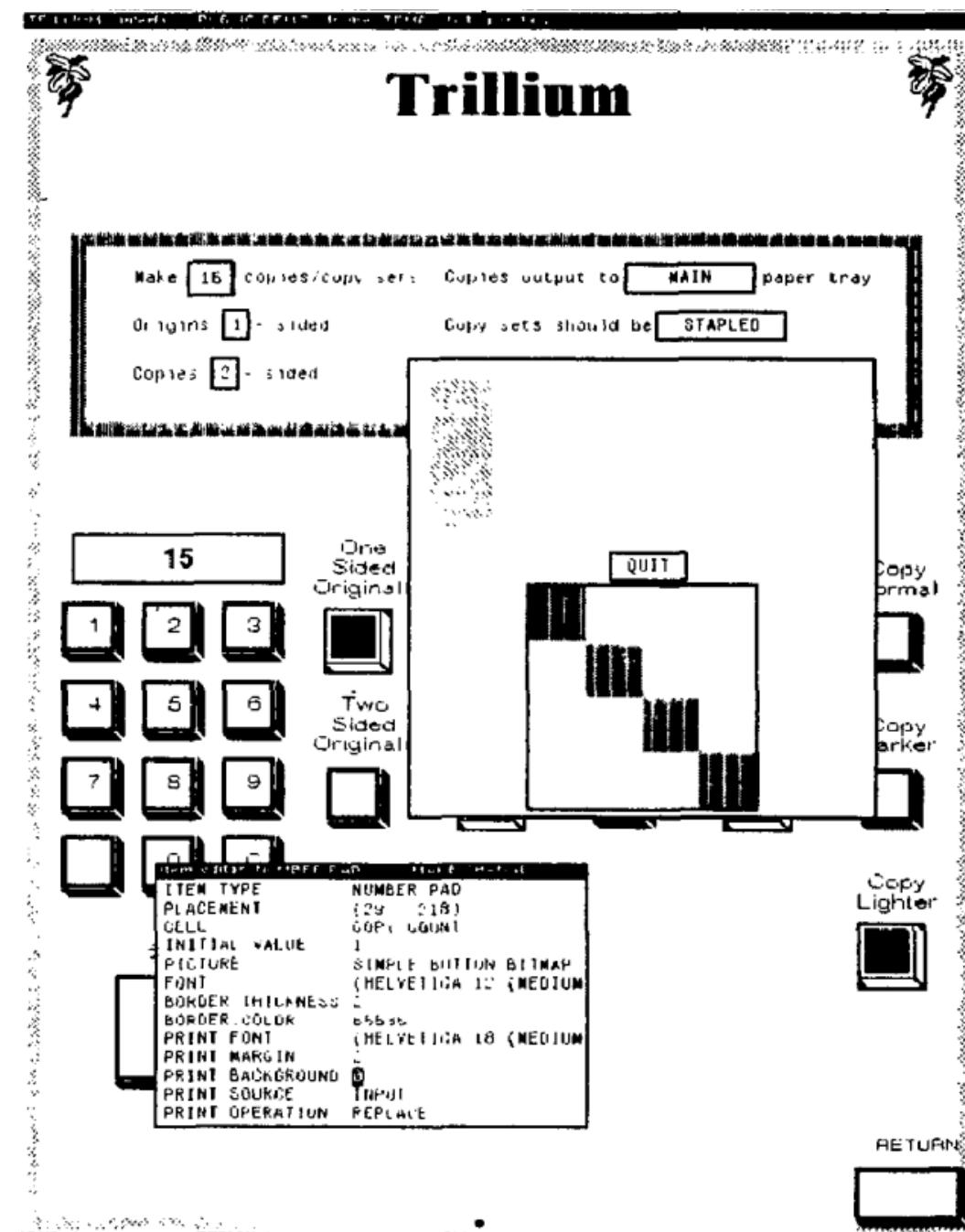
Object	structure prototype's picture prototype's values as save file subclass template	insert delete constrain merge move edit text	GeometricObject Line MidPointLine Point Quadrilateral Rectangle Triangle
Point			
Theorem			
Quadrilateral			
Rectangle			
TextThing			
Triangle			



Alan Borning

# Trillium

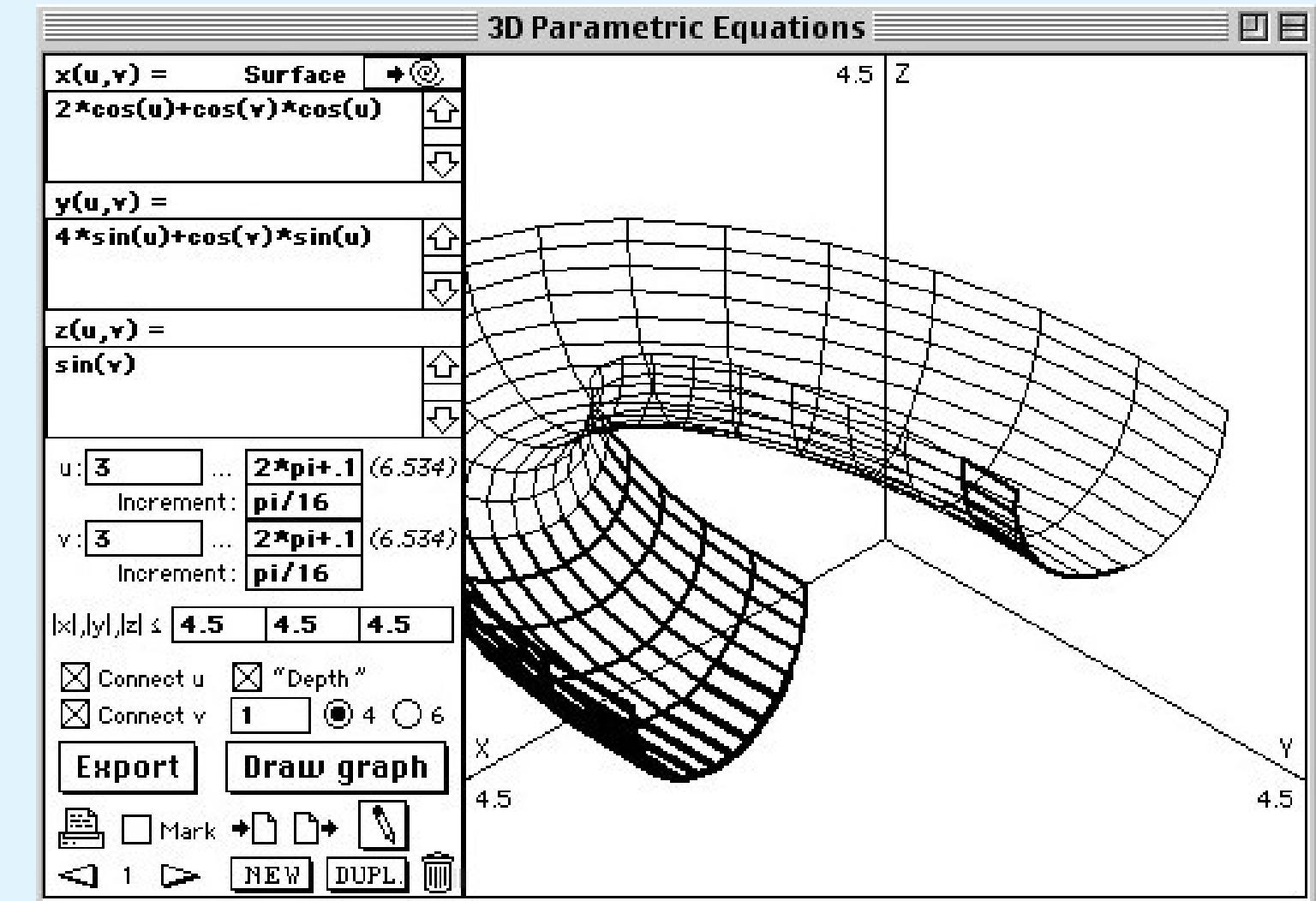
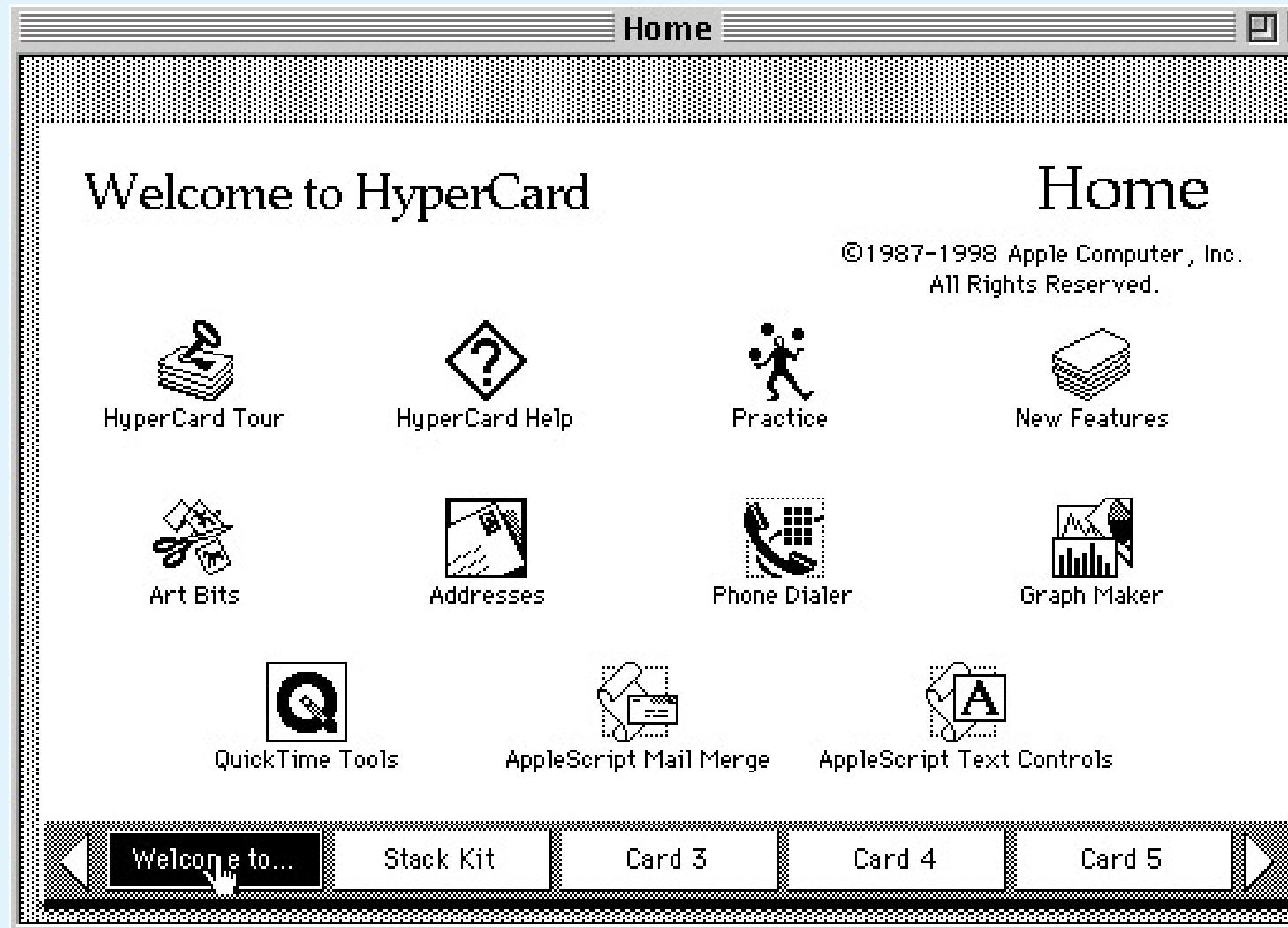
1986



Austin Henderson at Xerox PARC

# Hypercard

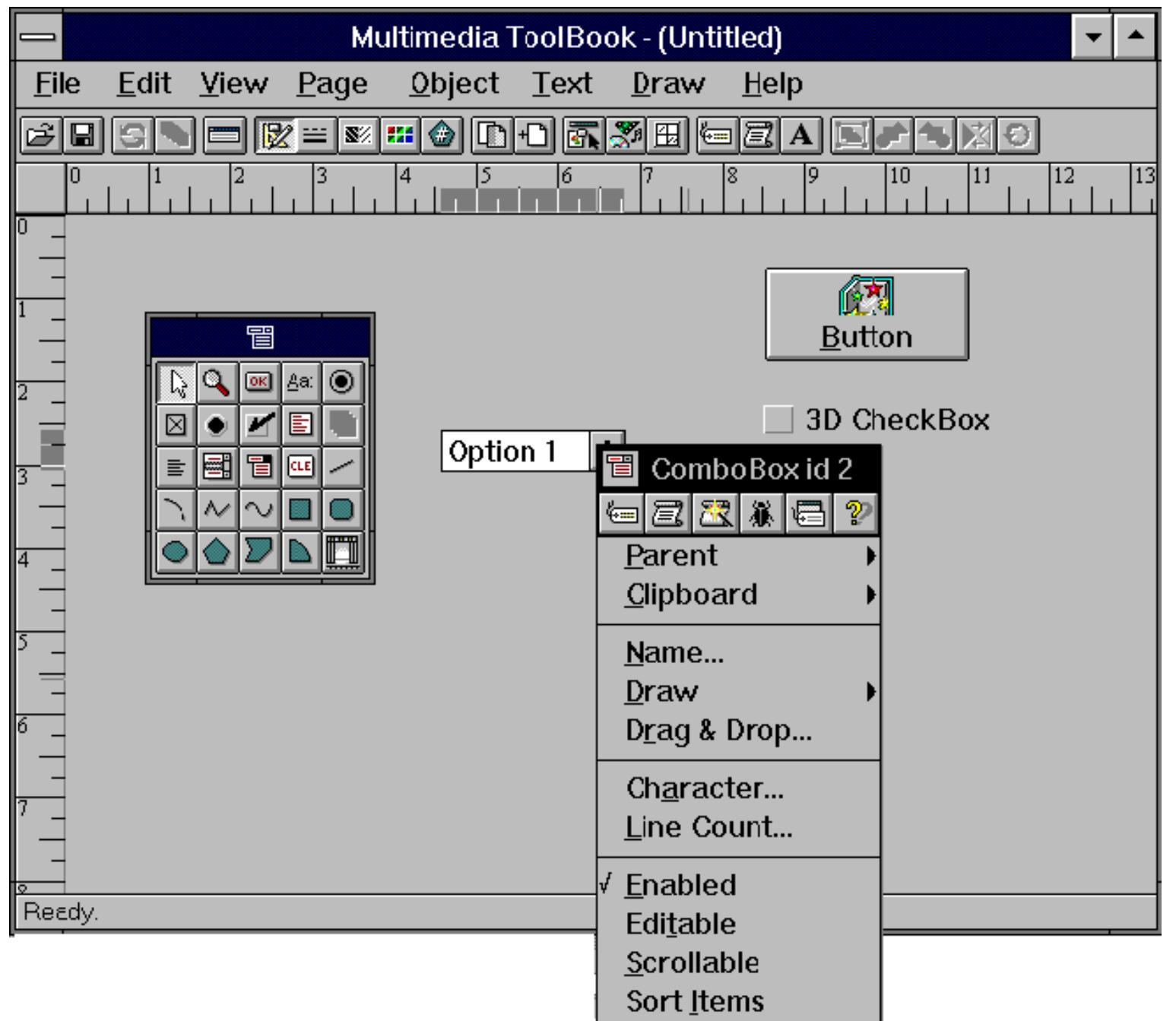
1987-1998



Bill Atkinson for Apple

# ToolBook

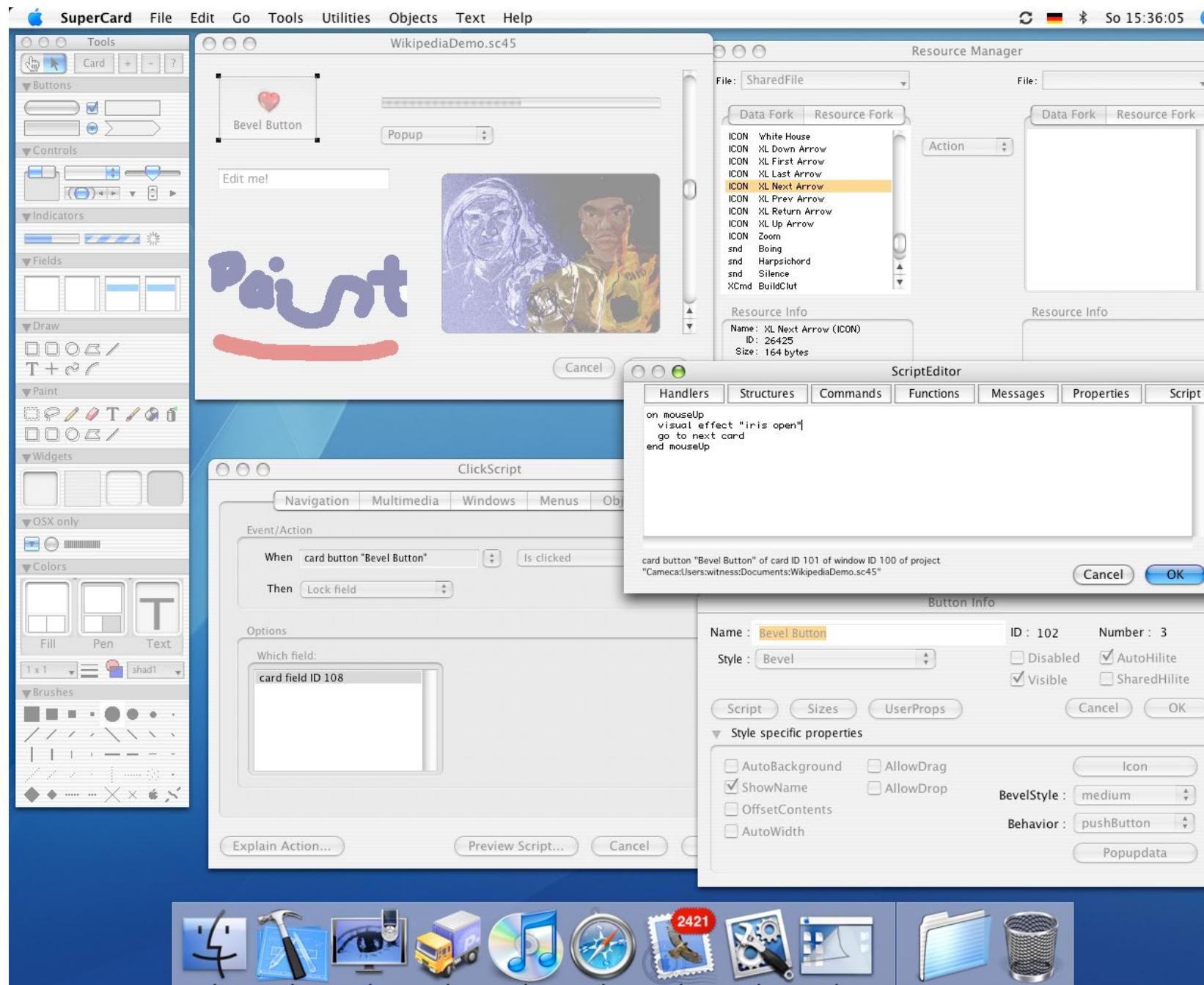
1990-2012



Asymetrix (Later SumTotal Systems)

# SuperCard

1989-2012



Bill Appleton for Silicon Beach Software (Later Aldus Corporation, Allegiant Technologies, Incwell DMG, now Solutions Etcetera)

# **'Bureaucracy Processing'**

From Static Documents  
To Living Environments

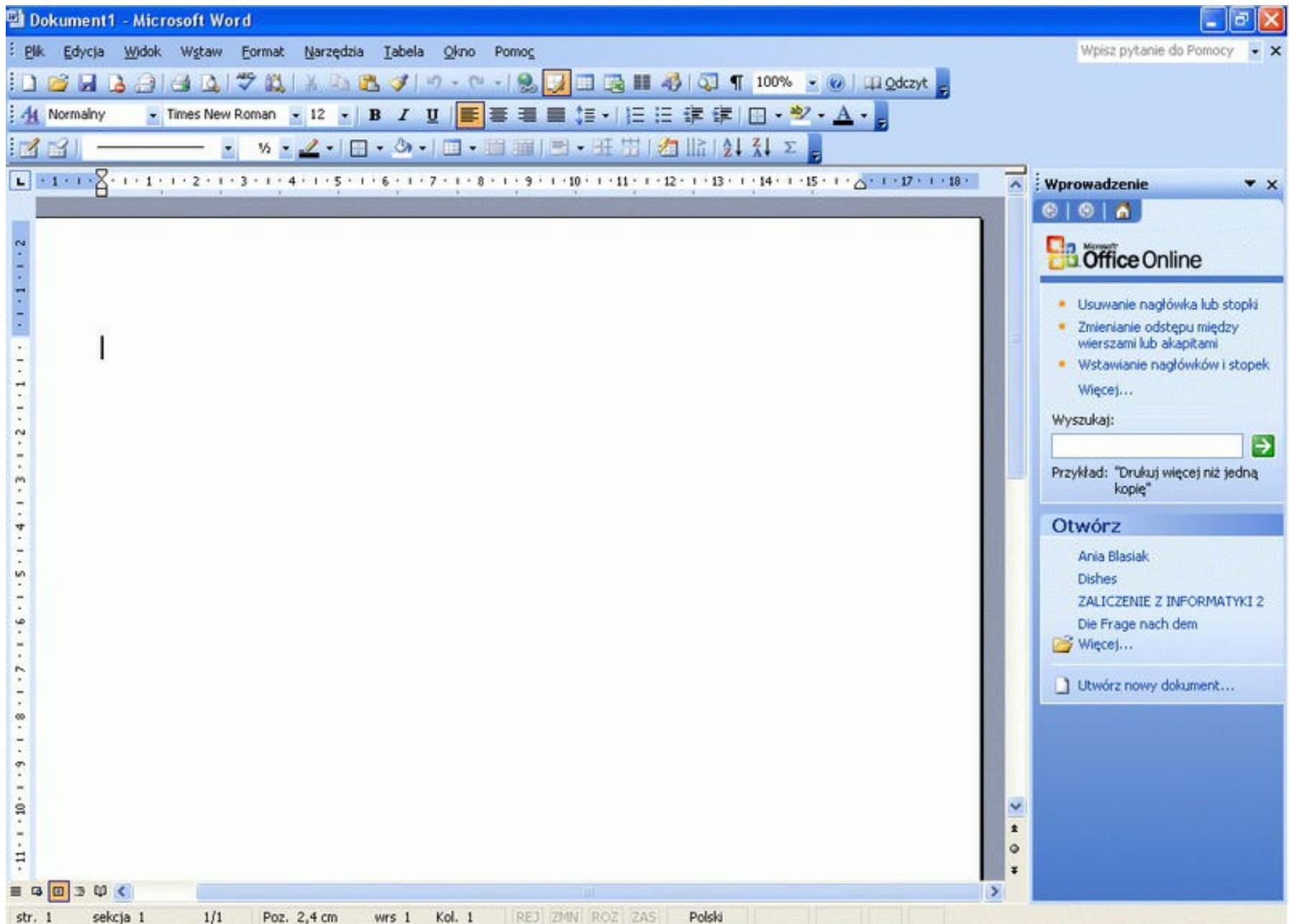
# WordStar

**1978-1999**

Rob Barnaby

# Word

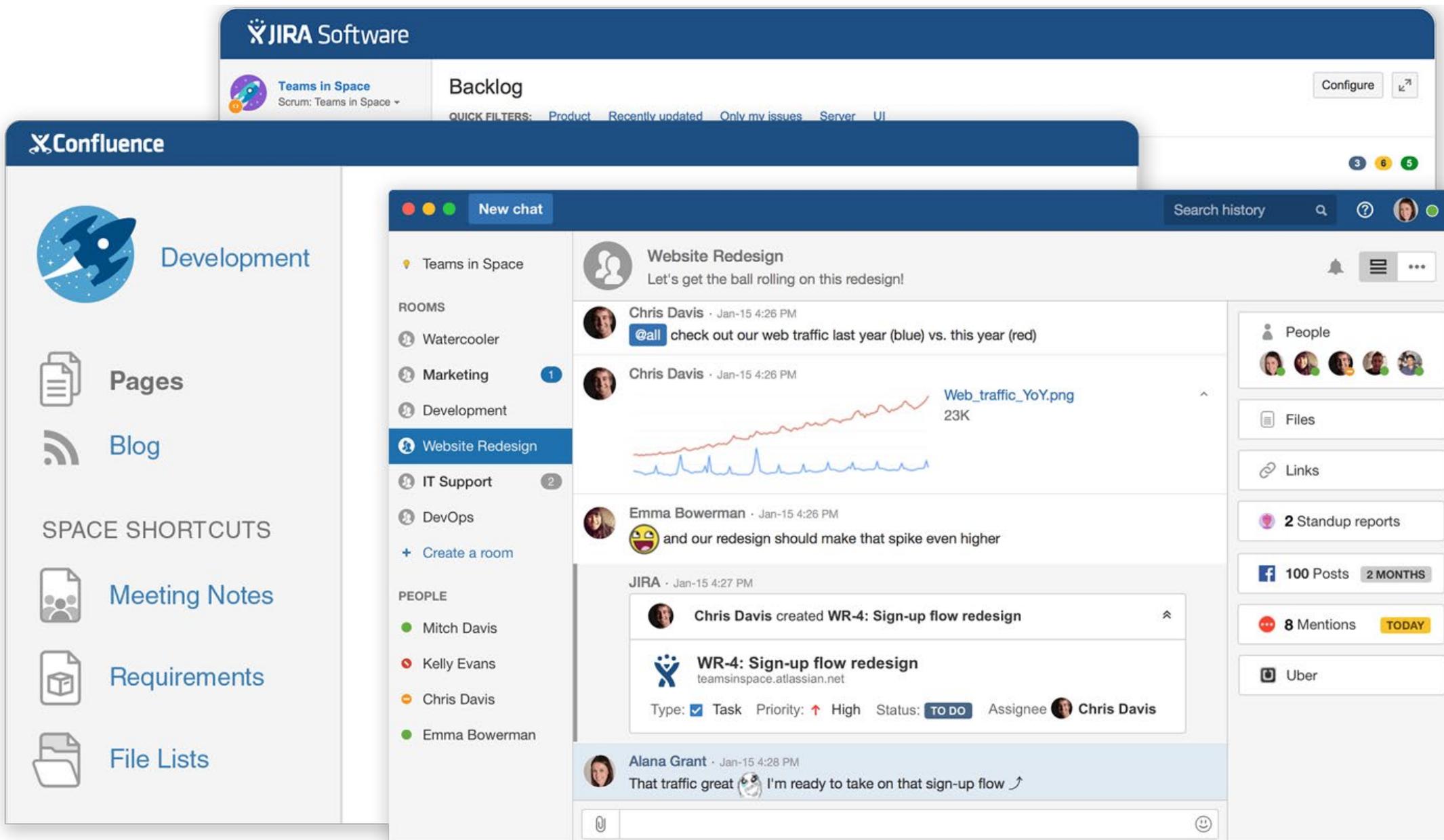
1983-Today



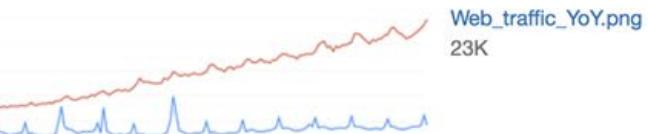
Microsoft

# Atlassian (Suite)

2002-Today



The screenshot shows the Atlassian Confluence interface. On the left, there is a sidebar with various links: Development (with a rocket icon), Pages (with a document icon), Blog (with a feed icon), SPACE SHORTCUTS (Meeting Notes, Requirements, File Lists), and PEOPLE (Mitch Davis, Kelly Evans, Chris Davis, Emma Bowerman). The main area shows a chat window titled 'New chat' with the following messages:

- Website Redesign: Let's get the ball rolling on this redesign!
- Chris Davis · Jan-15 4:26 PM: @all check out our web traffic last year (blue) vs. this year (red)  

- Chris Davis · Jan-15 4:26 PM: 23K
- Emma Bowerman · Jan-15 4:26 PM: 😊 and our redesign should make that spike even higher
- JIRA · Jan-15 4:27 PM: Chris Davis created WR-4: Sign-up flow redesign
- WR-4: Sign-up flow redesign  
teamsinspace.atlassian.net
- Type:  Task Priority: ↑ High Status: TO DO Assignee:  Chris Davis
- Alana Grant · Jan-15 4:28 PM: That traffic great 😊 I'm ready to take on that sign-up flow ↗

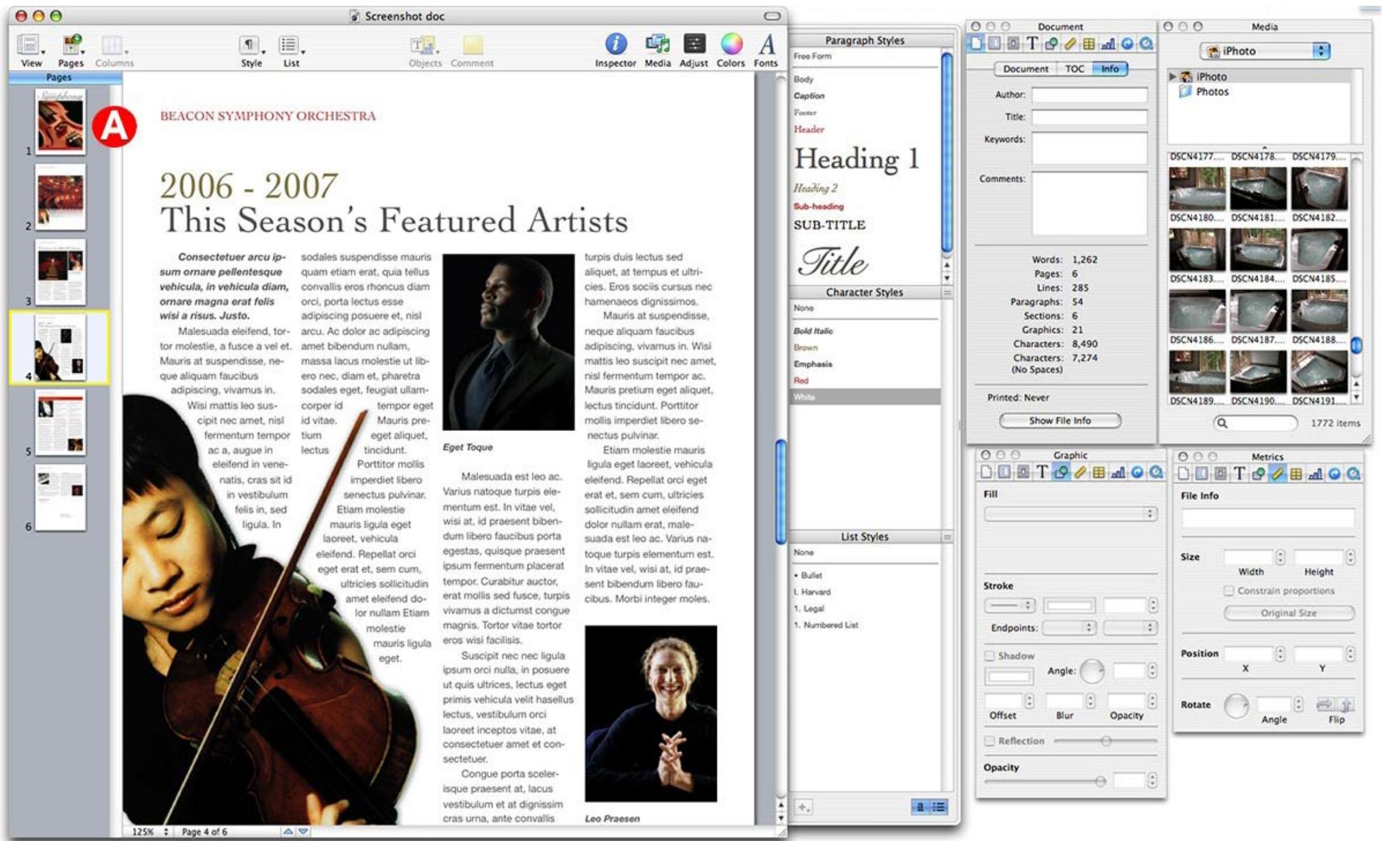
On the right, there is a sidebar with various metrics and links:

- People: 
- Files
- Links
- 2 Standup reports
- 100 Posts 2 MONTHS
- 8 Mentions TODAY
- Uber

Mike Cannon-Brookes and Scott Farquhar

# Pages

2005-Today

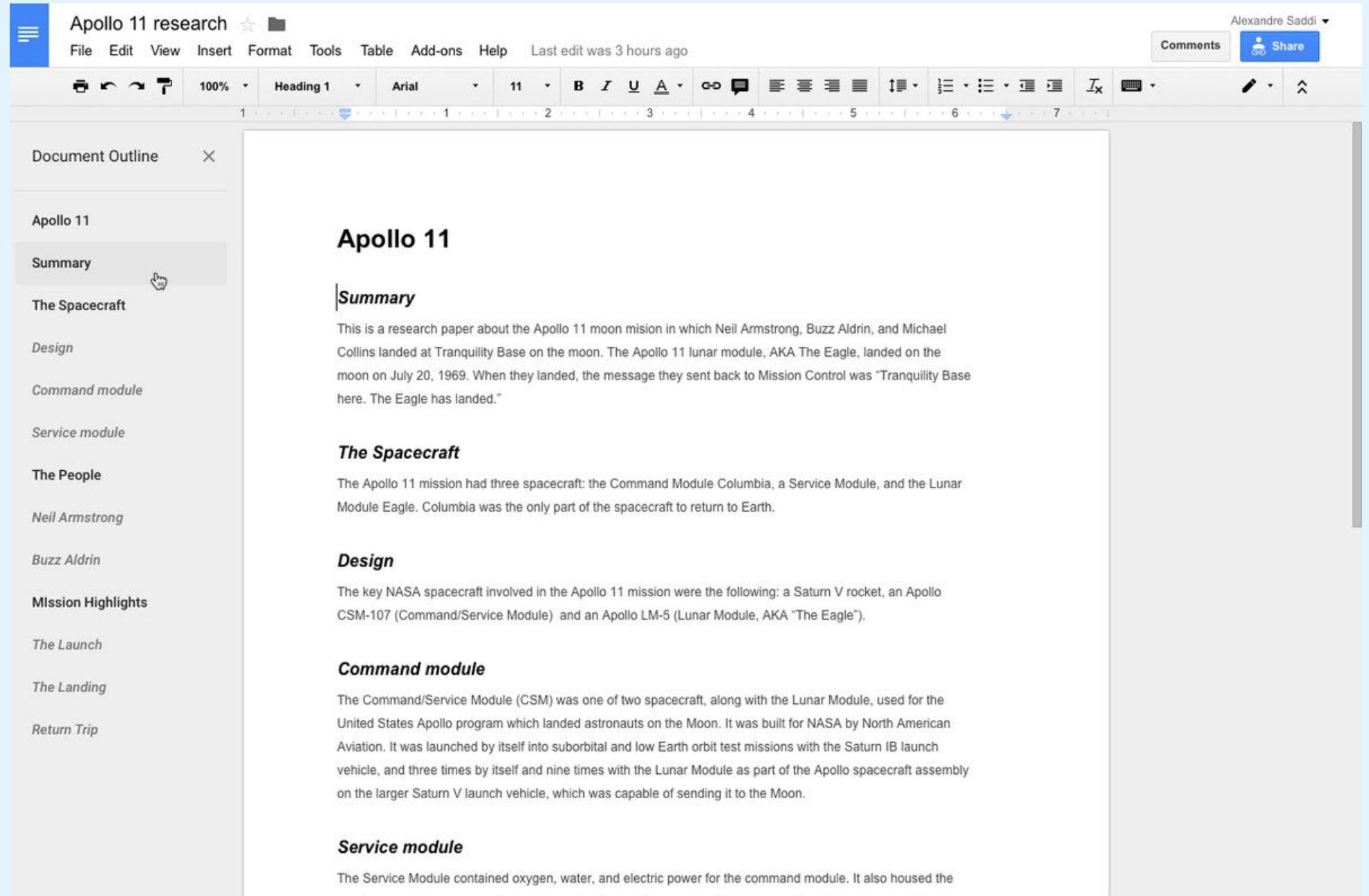


The screenshot shows the Apple Pages application interface. The main window displays a document titled "BEACON SYMPHONY ORCHESTRA" with the subtitle "2006 - 2007 This Season's Featured Artists". The document contains several images of musicians and blocks of placeholder text. The "Inspector" panel on the right is open, showing the "Document" tab with statistics: Words: 1,262, Pages: 6, Lines: 285, Paragraphs: 54, Sections: 6, Graphics: 21, Characters: 8,490, and Characters: 7,274. The "Media" tab shows a library of 1772 items from an "iPhoto" source, with thumbnail previews of various photos. Other tabs in the Inspector include "Graphic" and "Metrics". The "Pages" tab in the top left shows a preview of the document with page numbers 1 through 6. The "Style" tab shows a list of styles: "Heading 1", "Heading 2", "Sub-heading", "SUB-TITLE", "Title", and "Character Styles" (None, Bold Italic, Brown, Emphasis, Red, White). The "List Styles" tab shows "None", "• Bullet", "I. Harvard", "I. Legal", and "I. Numbered List". The "Paragraph Styles" tab shows "Free Form", "Body", "Caption", "Footer", and "Header". The "Colors" tab shows a color palette. The "Fonts" tab shows a list of fonts. The "Inspector" tab is also visible.

Apple

# Docs

2006-Today

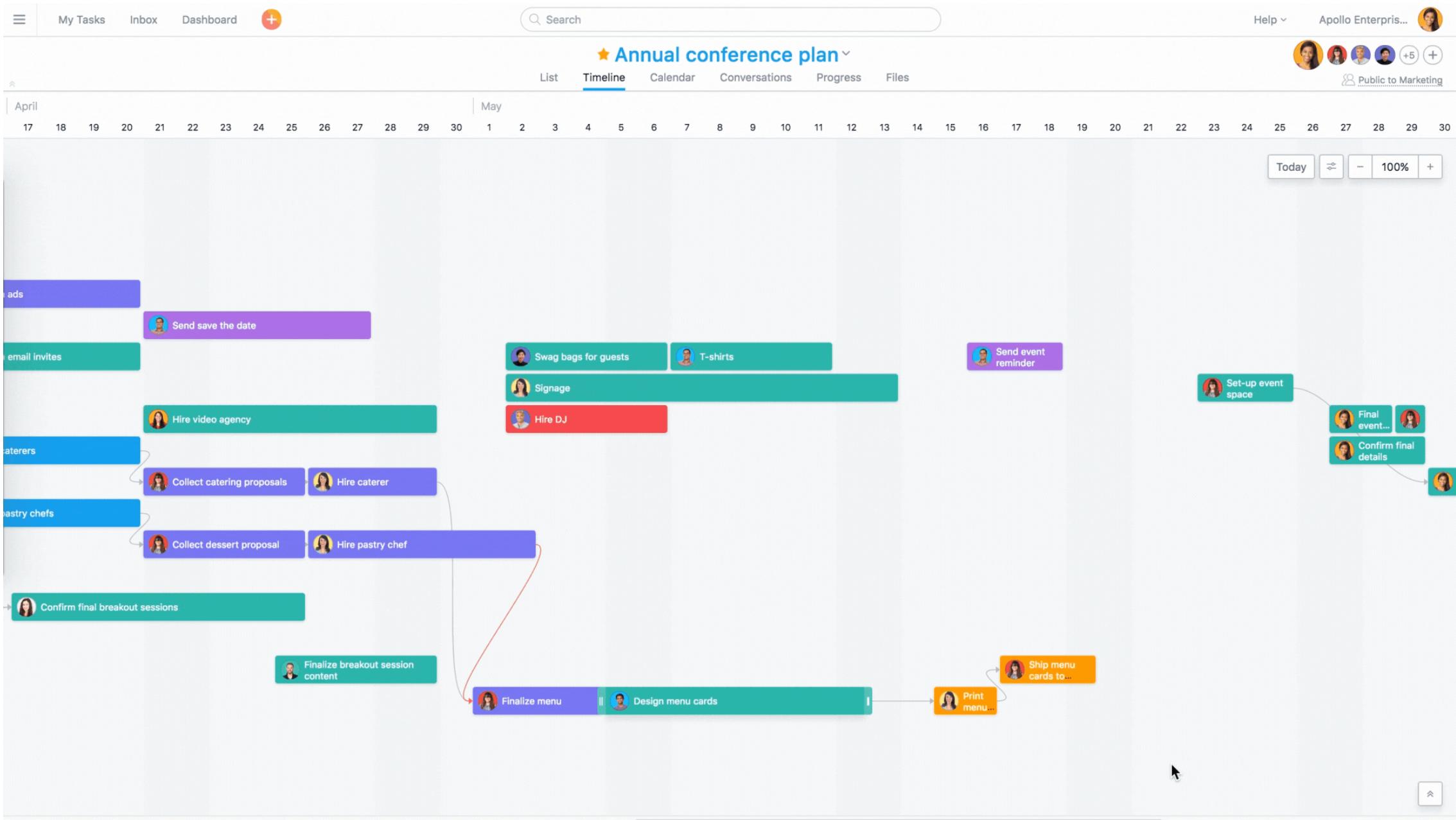


A screenshot of a Google Docs document titled "Apollo 11 research". The document is a research paper about the Apollo 11 moon mission. The main content is organized into sections: "Summary", "The Spacecraft", "Design", "Command module", "Service module", "The People", "Neil Armstrong", "Buzz Aldrin", "Mission Highlights", "The Launch", "The Landing", and "Return Trip". The "Summary" section contains a brief description of the mission. The "The Spacecraft" section discusses the three spacecraft: Command Module Columbia, Service Module, and Lunar Module Eagle. The "Design" section mentions the Saturn V rocket, CSM-107, and LM-5. The "Command module" section provides details about the CSM, its role in the mission, and its assembly with the other modules. The "Service module" section describes its function in providing oxygen, water, and power. The document is a collaborative effort, with "Alexandre Saddi" as the last editor and the document being last updated 3 hours ago.

Google

# Asana

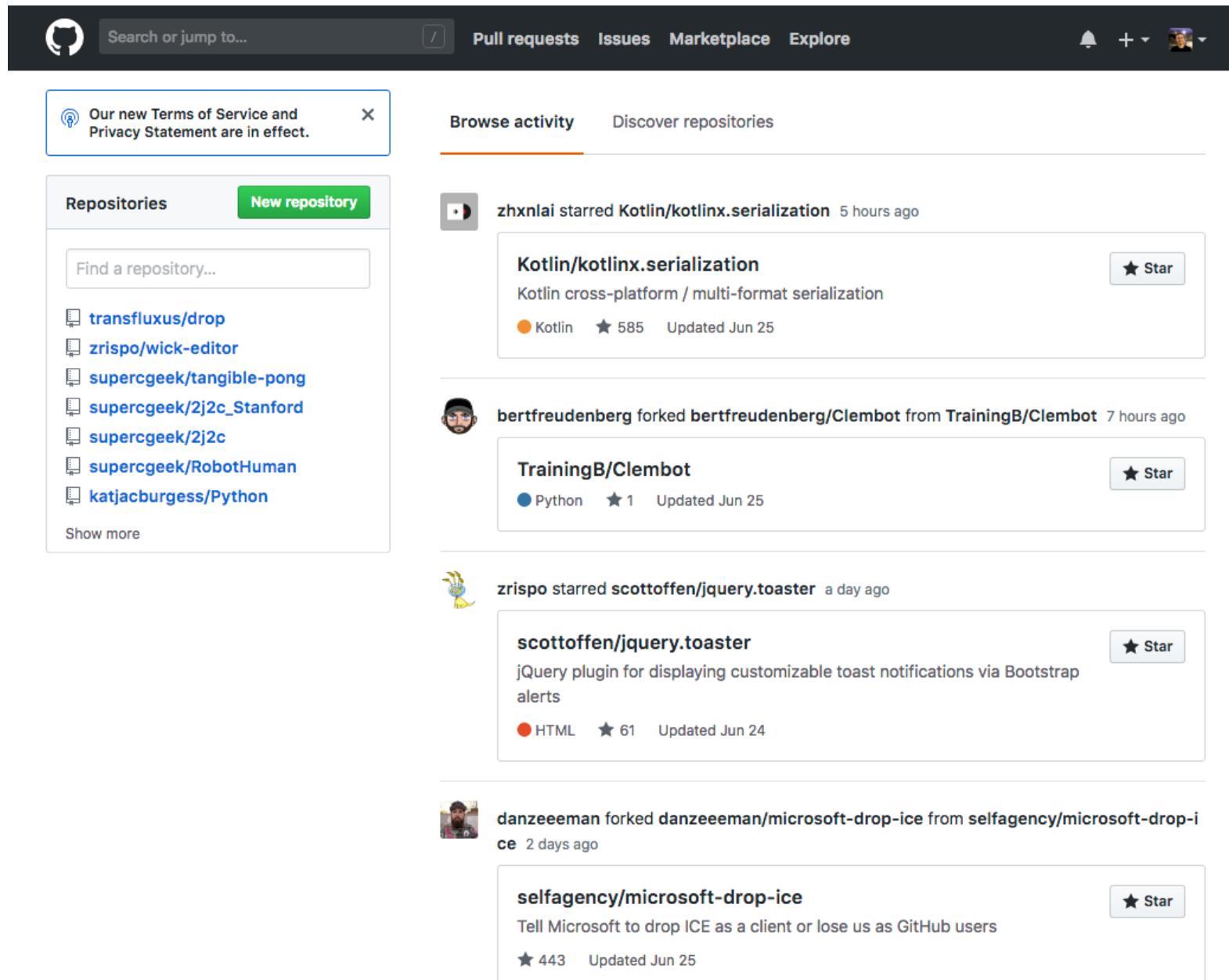
2008-Today



Dustin Moskovitz and Justin Rosenstein

# GitHub

2008-Today



The screenshot shows the GitHub homepage with a dark header. The header includes the GitHub logo, a search bar with placeholder text 'Search or jump to...', and navigation links for 'Pull requests', 'Issues', 'Marketplace', and 'Explore'. On the far right of the header are a notification bell, a plus sign for creating new repositories, and a user profile icon. Below the header, a blue-bordered box displays a message: 'Our new Terms of Service and Privacy Statement are in effect.' with a close 'X' button. The main content area is titled 'Browse activity' and 'Discover repositories'. It features a list of recent events:

- zhanlai starred Kotlin/kotlinx.serialization** 5 hours ago. Description: 'Kotlin cross-platform / multi-format serialization'. Languages: Kotlin. Stars: 585. Updated Jun 25. Includes a 'Star' button.
- berfreudenberg forked berfreudenberg/Clembot from TrainingB/Clembot** 7 hours ago. Description: 'TrainingB/Clembot'. Languages: Python. Stars: 1. Updated Jun 25. Includes a 'Star' button.
- zrispo starred scottffen/jquery.toaster** a day ago. Description: 'jQuery plugin for displaying customizable toast notifications via Bootstrap alerts'. Languages: HTML. Stars: 61. Updated Jun 24. Includes a 'Star' button.
- danzeeeman forked danzeeeman/microsoft-drop-ice from selfagency/microsoft-drop-ice** 2 days ago. Description: 'Tell Microsoft to drop ICE as a client or lose us as GitHub users'. Stars: 443. Updated Jun 25. Includes a 'Star' button.

On the left side, there is a sidebar titled 'Repositories' with a 'New repository' button. It includes a search bar and a list of repositories:

- transfluxus/drop
- zrispo/wick-editor
- supergeek/tangible-pong
- supergeek/2j2c\_Stanford
- supergeek/2j2c
- supergeek/RobotHuman
- katjacburgess/Python

Below this list is a 'Show more' link.

Tom Preston-Werner for GitHub Inc (Now Microsoft)

# Wave

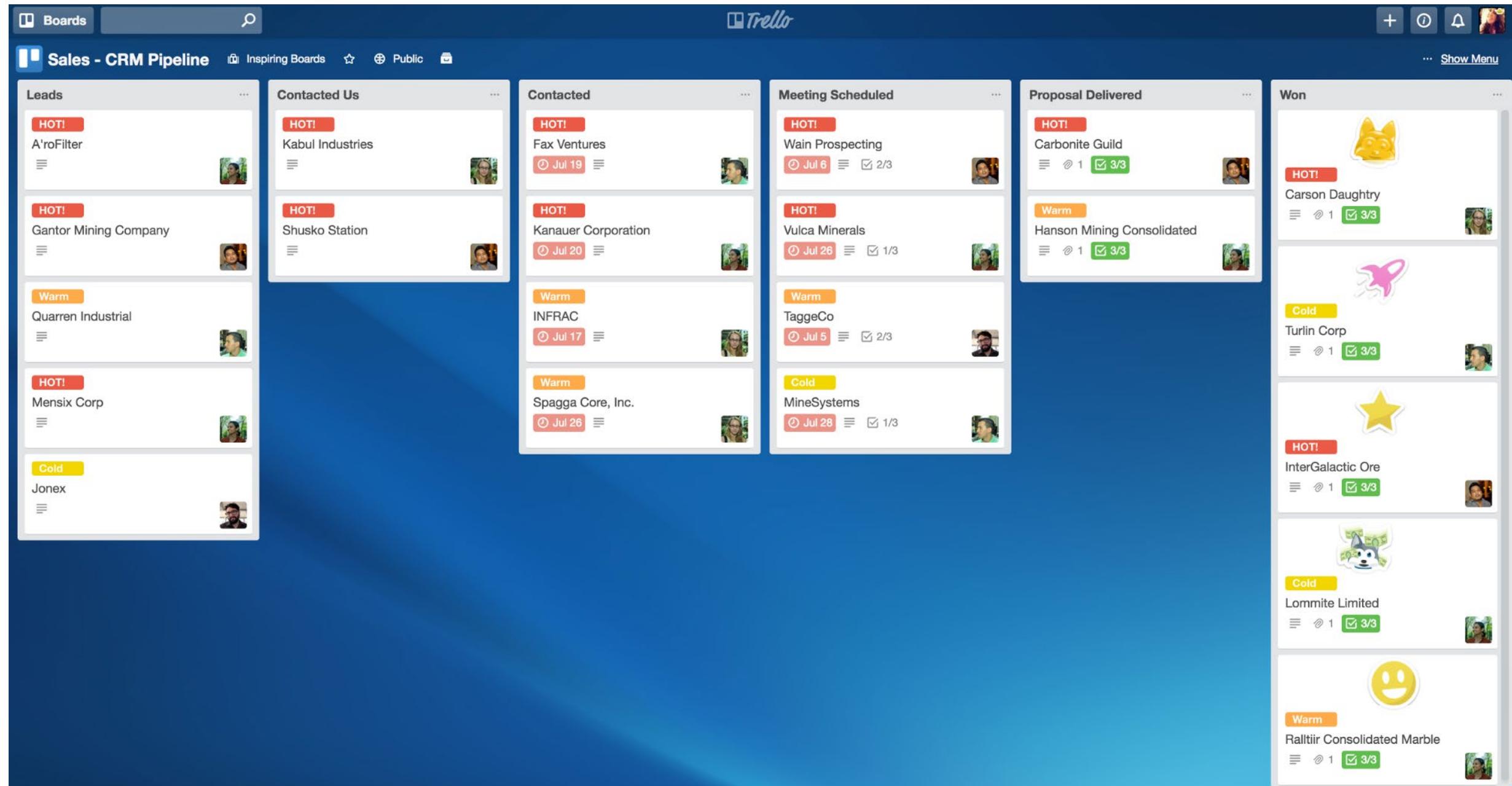
2009-2011



Google

# Trello

# 2011-Today



**Leads**

- HOT! A'roFilter
- HOT! Gantor Mining Company
- Warm Quarren Industrial
- HOT! Mensix Corp
- Cold Jonex

**Contacted Us**

- HOT! Kabul Industries
- HOT! Shusko Station
- INFRAC
- Warm Spagga Core, Inc.

**Contacted**

- HOT! Fax Ventures (Jul 19)
- HOT! Kanauer Corporation (Jul 20)
- Warm INFRAC (Jul 17)
- Warm Spagga Core, Inc. (Jul 26)

**Meeting Scheduled**

- HOT! Wain Prospecting (Jul 6)
- HOT! Vulca Minerals (Jul 26)
- Warm TaggeCo (Jul 5)
- Cold MineSystems (Jul 28)

**Proposal Delivered**

- HOT! Carbonite Guild (1)
- Warm Hanson Mining Consolidated (1)

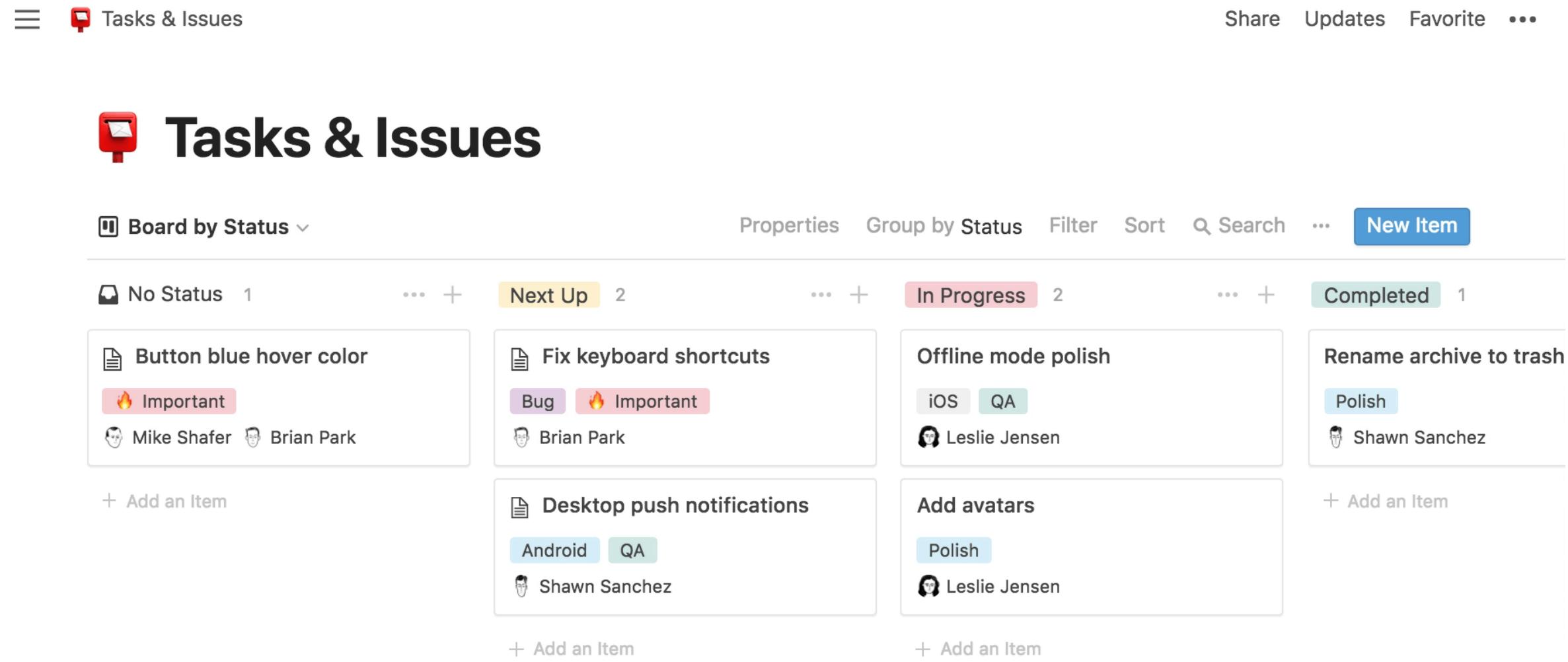
**Won**

- HOT! Carson Daughtry (1)
- Cold Turlin Corp (1)
- HOT! InterGalactic Ore (1)
- Cold Lommite Limited (1)
- Warm Ralltir Consolidated Marble (1)

Joel Spolsky at Fog Creek (Later Atlassian)

# Notion

2016-Today



Tasks & Issues

Board by Status

Properties Group by Status Filter Sort Search ... New Item

No Status 1

Next Up 2

In Progress 2

Completed 1

Button blue hover color (Important, Mike Shafer, Brian Park)

Fix keyboard shortcuts (Bug, Important, Brian Park)

Offline mode polish (iOS, QA, Leslie Jensen)

Rename archive to trash (Polish, Shawn Sanchez)

Desktop push notifications (Android, QA, Shawn Sanchez)

Add avatars (Polish, Leslie Jensen)

+ Add an Item

Notion Labs Inc

# Bits to Atoms

Computation meets Business & Industry

# Spreadsheets

Functional Programming – with a visual interface – became the PC's first 'Killer App'

# VisiCalc

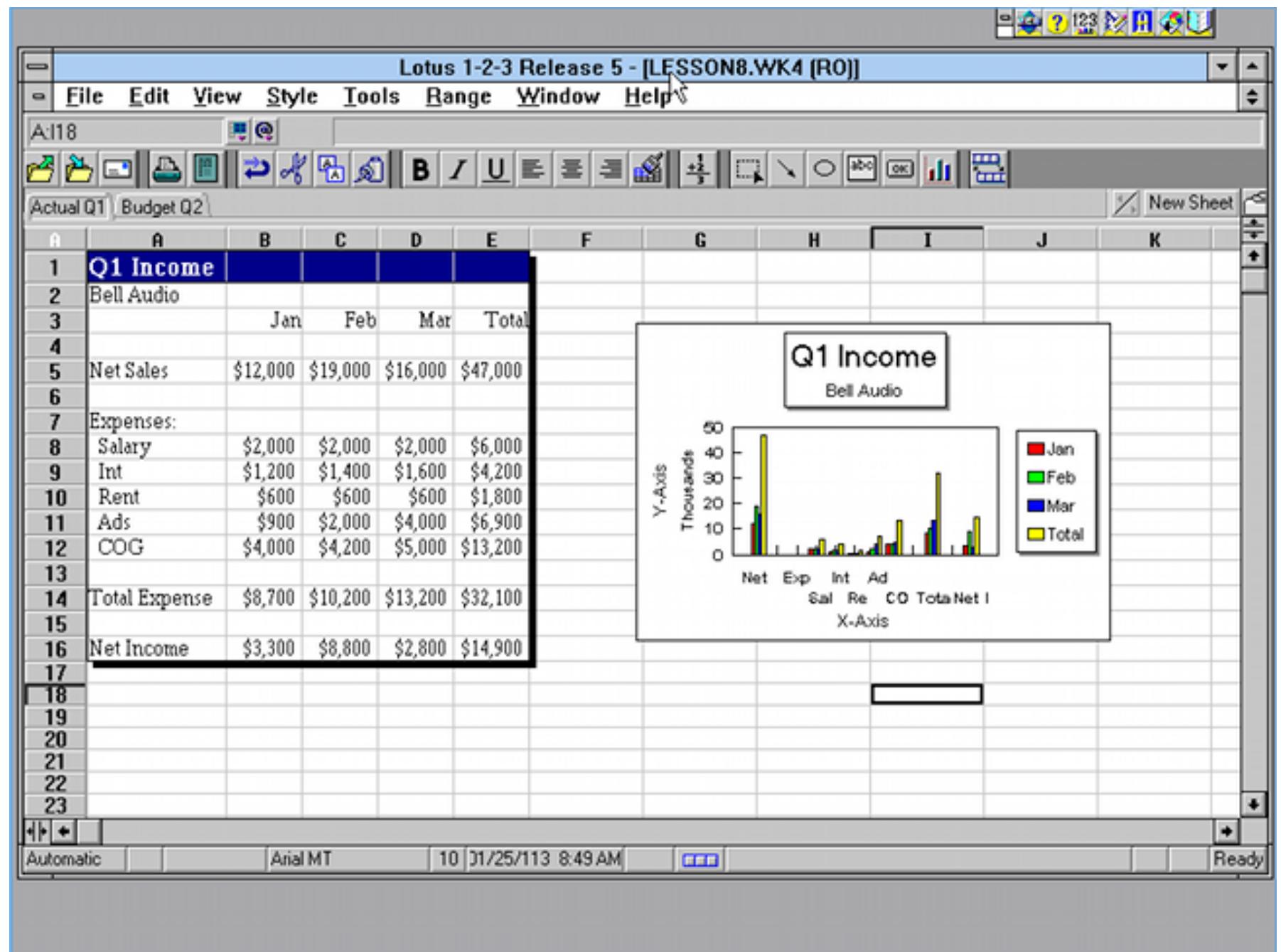
1979-1983

C11 (L) TOTAL				C125
	A	B	C	D
1	ITEM	NO.	UNIT	COST
2	MUCK RAKE	43	12.95	556.85
3	BUZZ CUT	15	6.75	101.25
4	TOE TONER	250	49.95	12487.50
5	EYE SNUFF	2	4.95	9.90
6			-----	-----
7			SUBTOTAL	13155.50
8			9.75% TAX	1282.66
9			-----	-----
10			TOTAL	14438.16
11				
12				
13				
14				
15				
16				
17				
18				
19				
20				

Software Arts

# 1-2-3

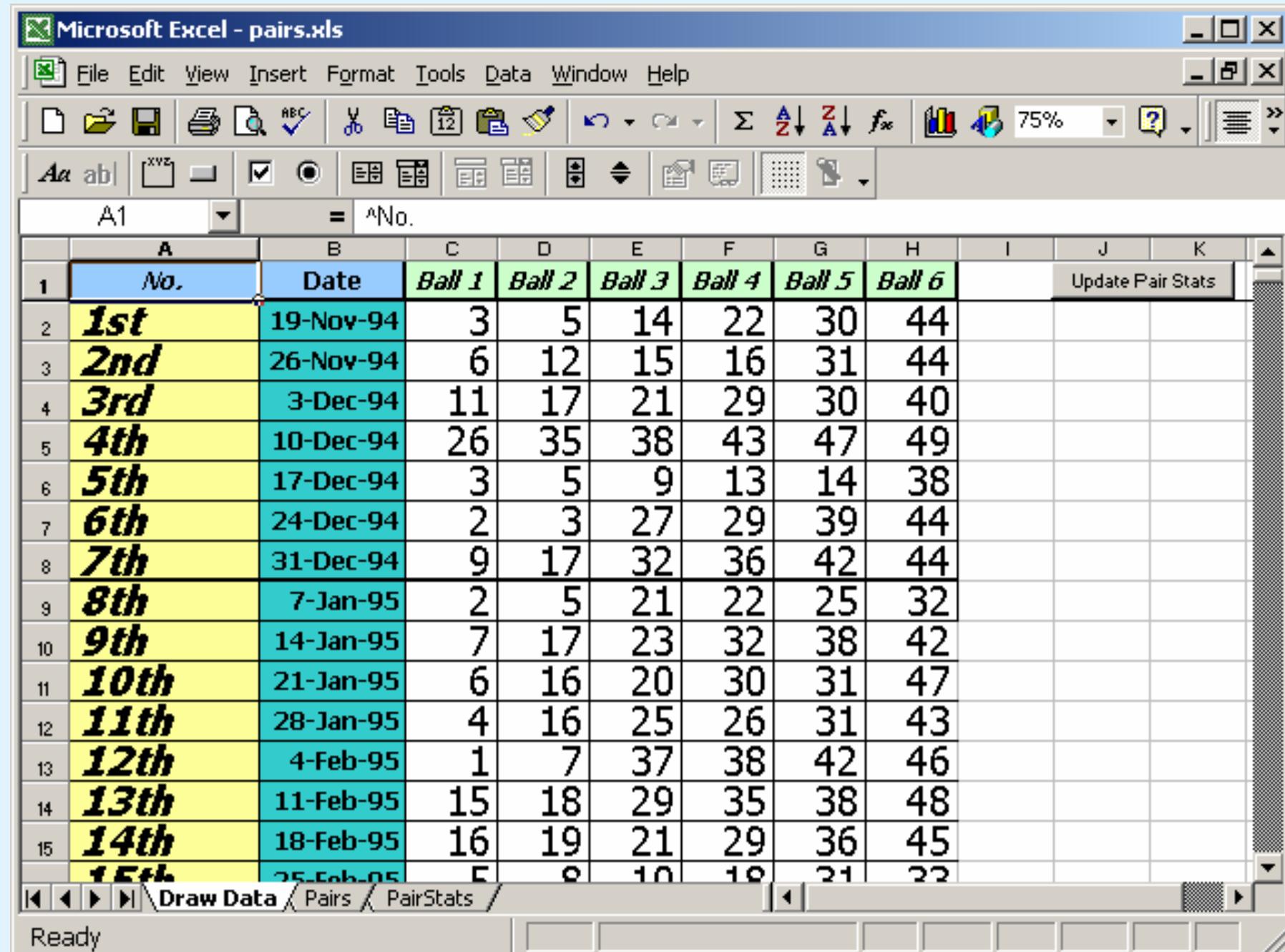
# 1983-2002



Lotus Software

# Excel

1987-Today



A screenshot of Microsoft Excel 97 showing a spreadsheet titled "Microsoft Excel - pairs.xls". The spreadsheet tracks lottery draws from November 1994 to February 1995. The columns represent the draw number (No.), date, and six lottery balls (Ball 1 to Ball 6). The first row contains the column headers. The data is organized into 15 rows, each representing a draw. The first draw is labeled "1st" and the last draw is labeled "15th". The dates range from "19-Nov-94" to "25-Feb-95". The balls are listed in columns C through H. The "No." column is bolded, and the "Date" column is also bolded. The "Ball 1" through "Ball 6" columns are green, and the "Update Pair Stats" button is in column I. The "Draw Data" tab is selected in the bottom navigation bar.

	A	B	C	D	E	F	G	H	I	J	K
1	No.	Date	Ball 1	Ball 2	Ball 3	Ball 4	Ball 5	Ball 6			Update Pair Stats
2	<b>1st</b>	19-Nov-94	3	5	14	22	30	44			
3	<b>2nd</b>	26-Nov-94	6	12	15	16	31	44			
4	<b>3rd</b>	3-Dec-94	11	17	21	29	30	40			
5	<b>4th</b>	10-Dec-94	26	35	38	43	47	49			
6	<b>5th</b>	17-Dec-94	3	5	9	13	14	38			
7	<b>6th</b>	24-Dec-94	2	3	27	29	39	44			
8	<b>7th</b>	31-Dec-94	9	17	32	36	42	44			
9	<b>8th</b>	7-Jan-95	2	5	21	22	25	32			
10	<b>9th</b>	14-Jan-95	7	17	23	32	38	42			
11	<b>10th</b>	21-Jan-95	6	16	20	30	31	47			
12	<b>11th</b>	28-Jan-95	4	16	25	26	31	43			
13	<b>12th</b>	4-Feb-95	1	7	37	38	42	46			
14	<b>13th</b>	11-Feb-95	15	18	29	35	38	48			
15	<b>14th</b>	18-Feb-95	16	19	21	29	36	45			
	<b>15th</b>	25-Feb-95	5	9	10	19	21	22			

Microsoft

# Numbers

2005-Today

River Rafting Budget

Participants & Budget

Participants

Budget

Expenses

Rafting

Camping

T-shirts

Participants

Participants & Budget

Participants

Budget

Expenses

Rafting

Camping

T-shirts

Styles

Basic

Basic (No Grid)

Gray

Gray Headers

Gray Fill

Beige

Ledger

Blue

Blue Headers

Blue Fill

Event Planner

Party Plan

White-water rafting on the Klamath River

To I-80

HWY 89

HWY 50

Let's meet at the campground.

River Rafting Budget.

Participants

Last name	First name	RSVP	Phone	Email address	Paid	Amount due
Chen	William	✓	(541) 555-8142	Chen_William152@postlive.net	\$0.00	\$365.00
Garcia	Sophia	✓	(707) 555-5824	Sophia245@postlive.net	\$365.00	\$0.00
Green	Mark	✓	(530) 555-4598	Green_Mark@postlive.net	\$365.00	\$0.00
Johnson	Gemma	✓	(707) 555-1673	GJohnson4170@postlive.net	\$100.00	\$265.00
Marshall	Orlando	■	(541) 555-2267	Marshall_Orlando@postlive.net	\$0.00	\$0.00
Mitchell	Jenny	✓	(530) 555-7030	Mitchell_Jenny@postlive.net	\$100.00	\$265.00
Nguyen	Nancy	✓	(530) 555-7634	Nancy_D_Nguyen@postlive.net	\$200.00	\$165.00
Parker	Seth	✓	(707) 555-7800	Parker_Seth@postlive.net	\$200.00	\$165.00
Parker	Judy	■	(707) 555-1149	Parker6379@postlive.net	\$0.00	\$0.00
Roberts	Greg	✓	(541) 555-1035	Roberts_Greg1834@postlive.net	\$0.00	\$365.00
Sanchez	Isabella	■	(530) 555-7540	Isabella_Sanchez@postlive.net	\$0.00	\$0.00
Wisman	Ronald	✓	(541) 555-3751	Ronald.Wisman@postlive.net	\$365.00	\$0.00
<b>Total</b>		9			<b>\$1,695.00</b>	<b>\$1,590.00</b>

Expenses

Item	Per person	Total cost
Rafting	\$165.00	\$1,485.00
Camping	\$60.00	\$540.00
Dinner	\$50.00	\$450.00
Beverages	\$36.00	\$324.00
Transportation	\$30.00	\$270.00
T-shirts	\$24.00	\$216.00
<b>Total</b>		<b>=SUM(Total Cost)</b>

Apple

# Sheets

2006-Today

Copy of Explore example

matthewguay@gmail.com

File Edit View Insert Format Data Tools Add-ons Help Last edit was 2 minutes ago

Comments Share

Explore

ANSWERS

Ask a question about your data

② Distribution of Winner

② Correlation between Teams and Matches

② Average of Teams

FORMATTING

Alternating colors for A1:J21 EDIT

ANALYSIS

Teams, Matches and Goals scored

Source: [https://en.wikipedia.org/wiki/FIFA\\_World\\_Cup](https://en.wikipedia.org/wiki/FIFA_World_Cup)

World Cup

Edition	Year	Host Country	Winner	Runner up	Average attendance	Teams	Matches	Goals scored	Average goals
1930 World Cup Uruguay	1930	Uruguay	Uruguay	Argentina	32,808	13	18	70	3.9
1934 World Cup Italy	1934	Italy	Italy	Czechoslovakia	21,353	16	17	70	4.1
1938 World Cup France	1938	France	Italy	Hungary	20,872	15	18	84	4.7
1950 World Cup Brazil	1950	Brazil	Uruguay	Brazil	47,511	13	22	88	4
1954 World Cup Switzerland	1954	Switzerland	Germany	Hungary	29,562	16	26	140	5.4
1958 World Cup Sweden	1958	Sweden	Brazil	Sweden	23,423	16	35	126	3.6
1962 World Cup Chile	1962	Chile	Brazil	Czechoslovakia	27,912	16	32	89	2.8
1966 World Cup England	1966	England	England	Germany	48,848	16	32	89	2.8
1970 World Cup Mexico	1970	Mexico	Brazil	Italy	50,124	16	32	95	3
1974 World Cup Germany	1974	Germany	Germany	Netherlands	49,099	16	38	97	2.6
1978 World Cup Argentina	1978	Argentina	Argentina	Netherlands	40,679	16	38	102	2.7
1982 World Cup Spain	1982	Spain	Italy	Germany	40,572	24	52	146	2.8
1986 World Cup Mexico	1986	Mexico	Argentina	Germany	46,039	24	52	132	2.5
1990 World Cup Italy	1990	Italy	Germany	Argentina	48,389	24	52	115	2.2
1994 World Cup United States	1994	United States	Brazil	Italy	68,991	24	52	141	2.7
1998 World Cup France	1998	France	France	Brazil	43,517	32	64	171	2.7
2002 World Cup Korea & Japan	2002	Korea & Japan	Brazil	Germany	42,269	32	64	161	2.5
2006 World Cup Germany	2006	Germany	Italy	France	52,491	32	64	147	2.3
2010 World Cup South Africa	2010	South Africa	Spain	Netherlands	49,670	32	64	145	2.3
2014 World Cup Brazil	2014	Brazil	Germany	Argentina	53,592	32	64	171	2.7

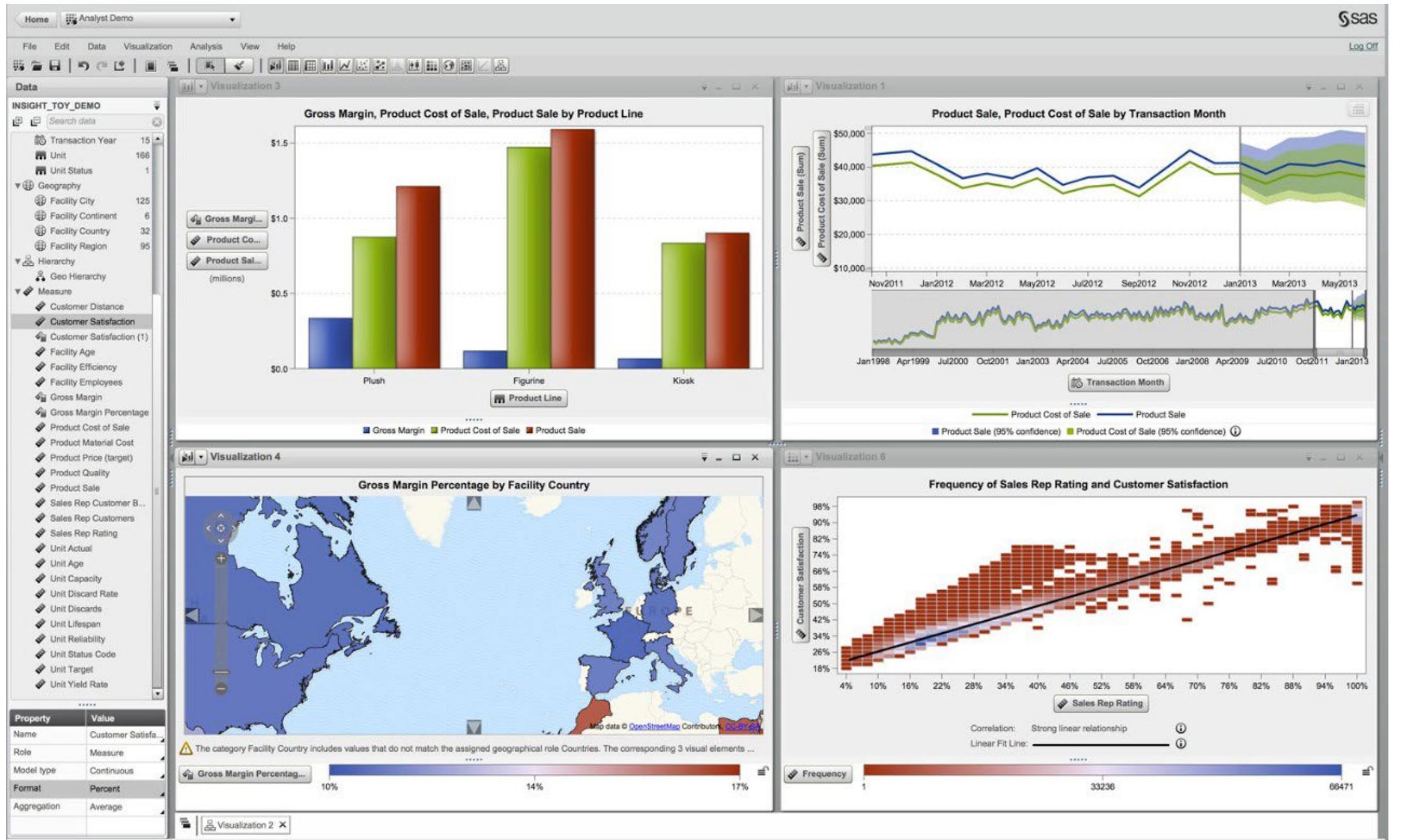
Google

# Data Analysis

From Simple Languages to  
Expansive Environments and  
Hosted Compute Platforms

# SAS

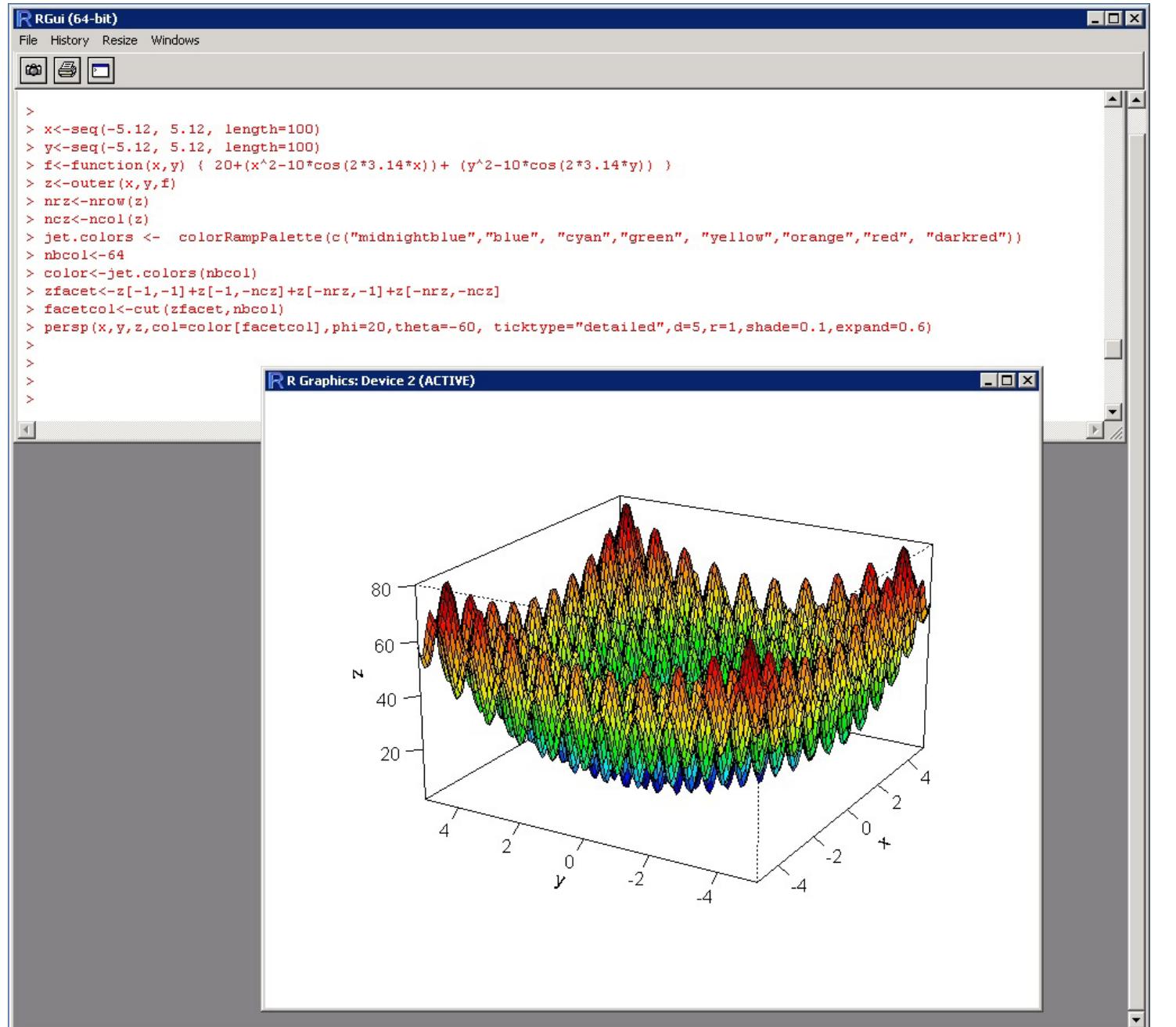
# 1976-2013



SAS Institute

# R

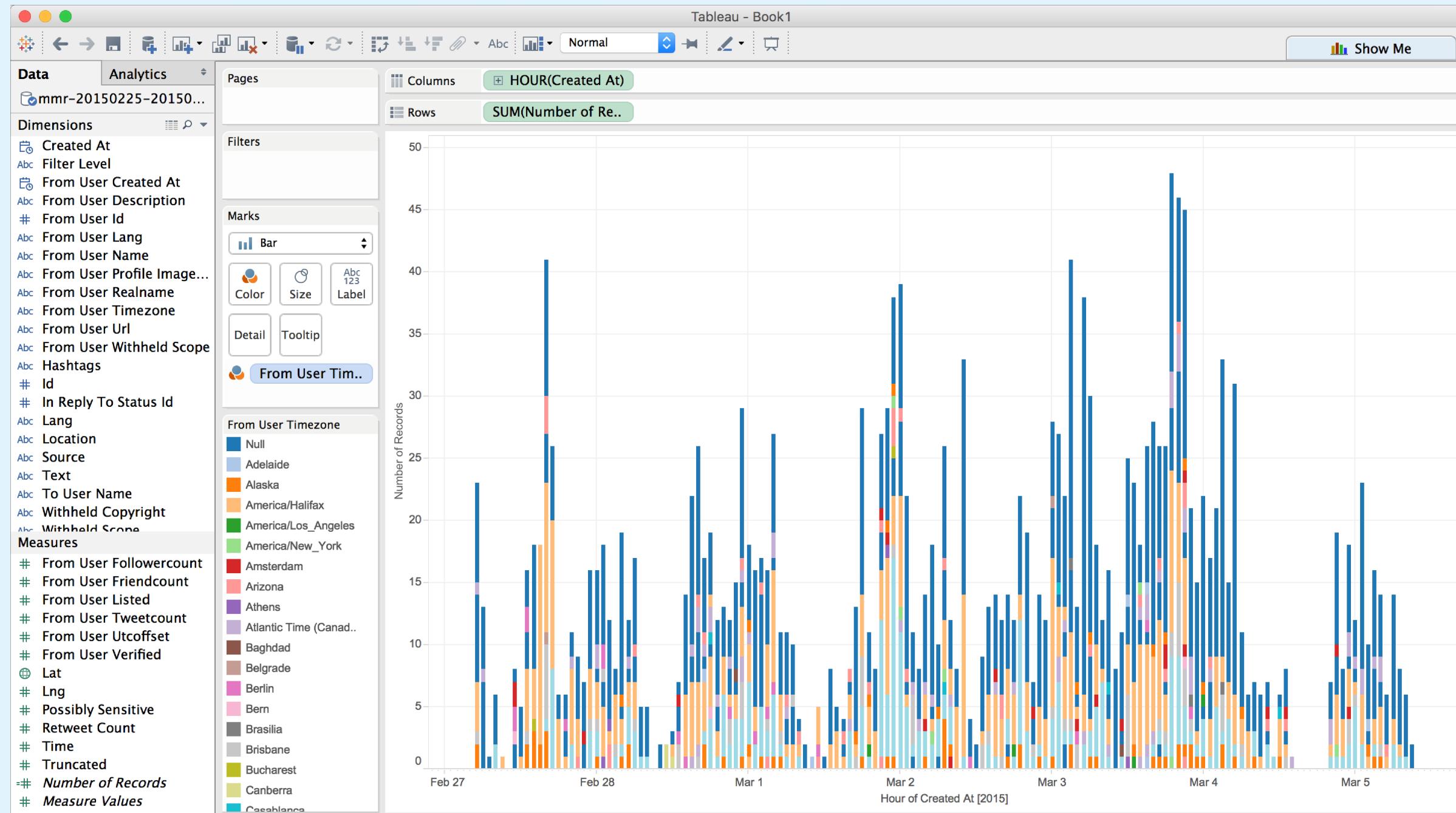
# 1993-Today



Ross Ihaka and Robert Gentleman (Now the R Core Team)

# Tableau

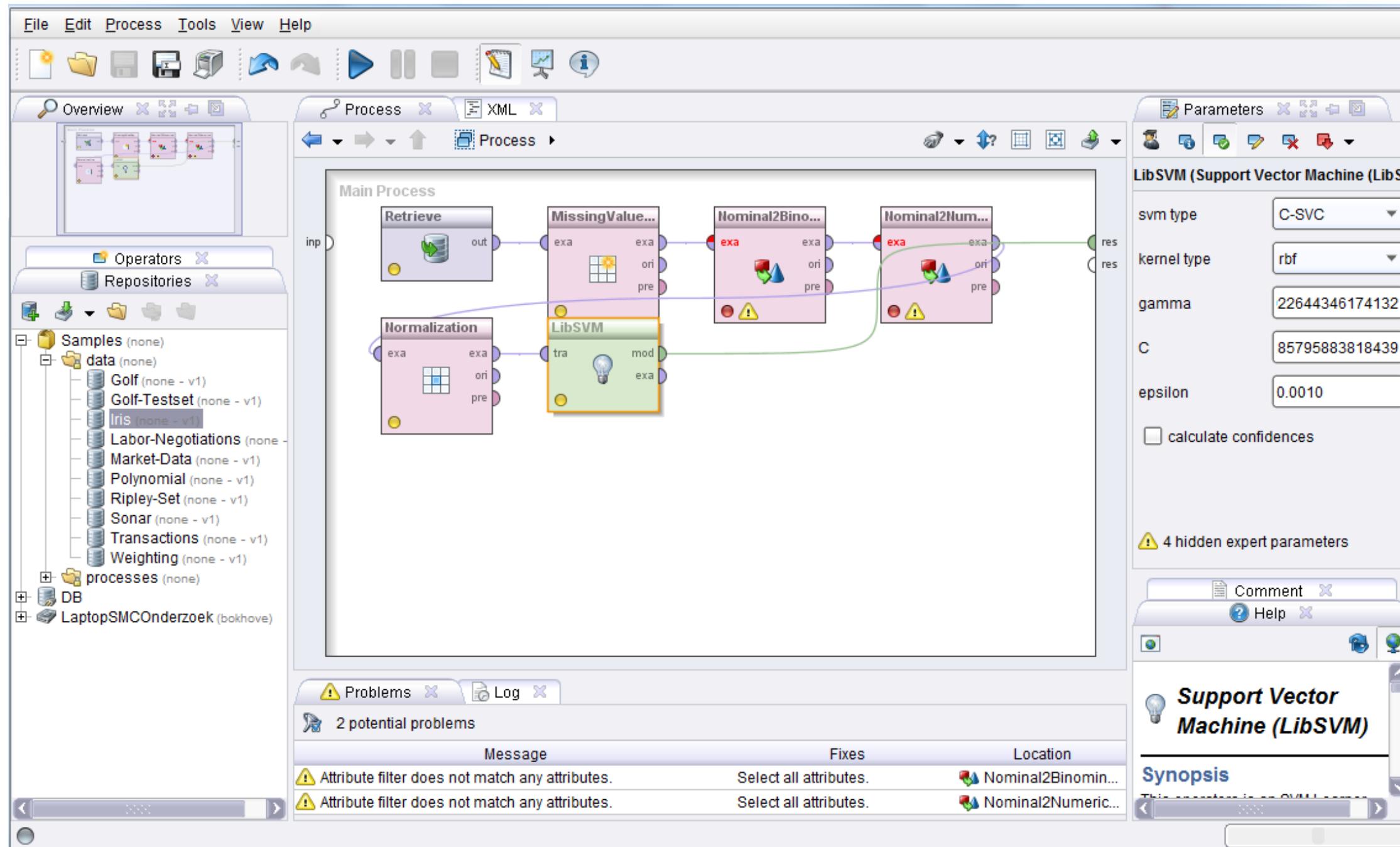
2003-Today



Pat Hanrahan, Christian Chabot, Chris Stolte for Tableau Software

# RapidMiner

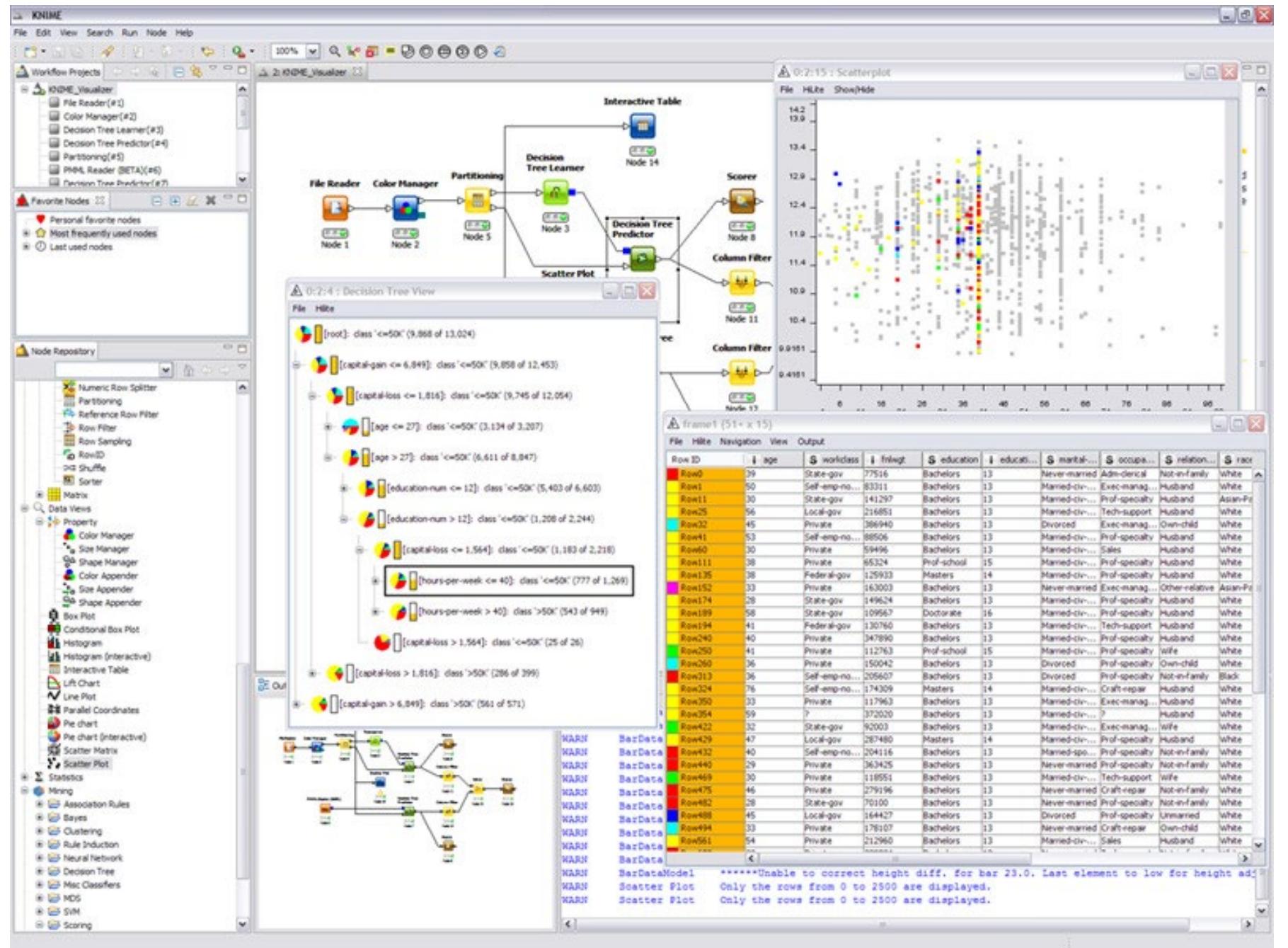
2006-Today



RapidMiner

# Knime

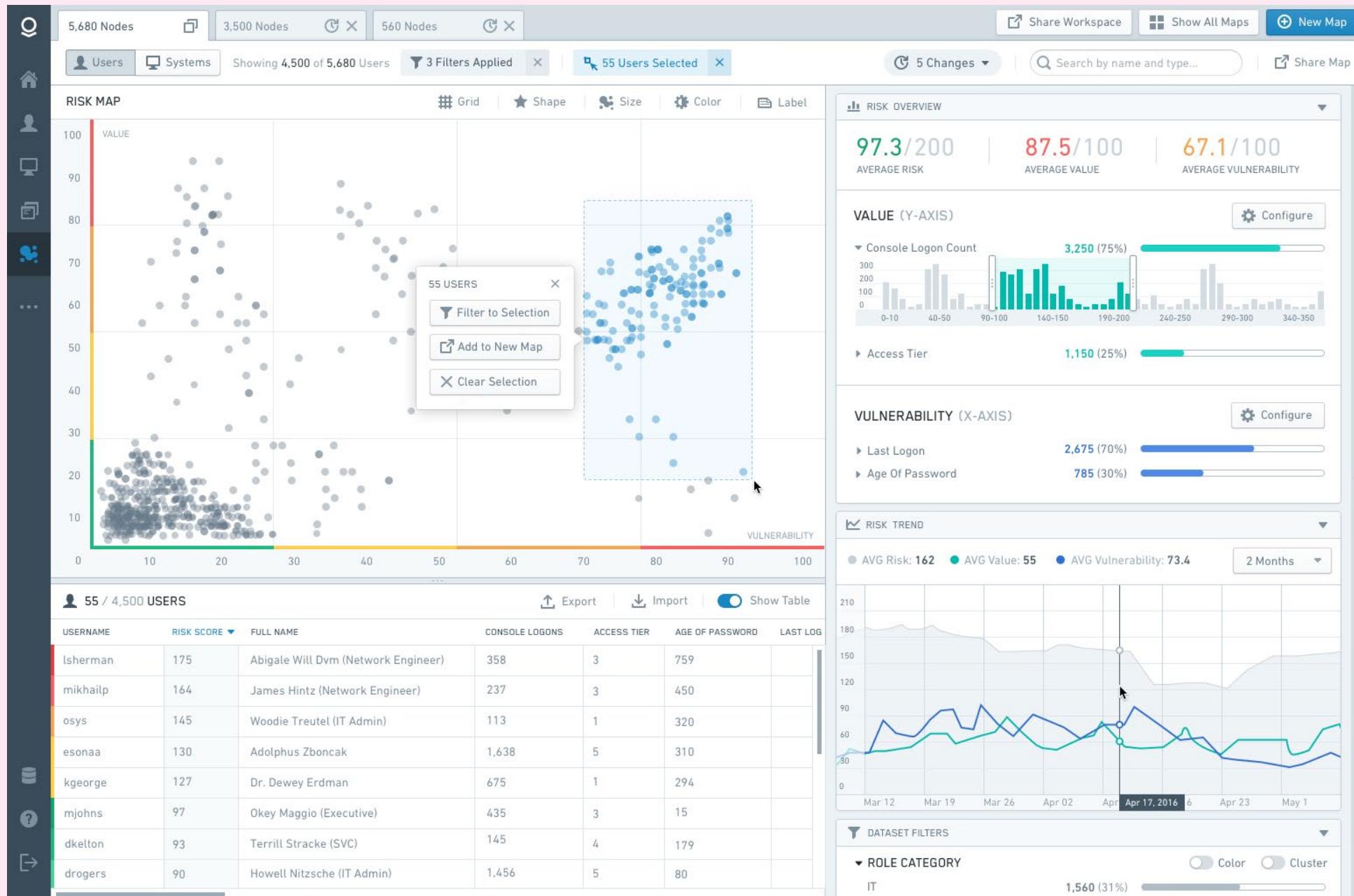
2006-Today



KNIME.com AG

# Foundry

# -Foundry

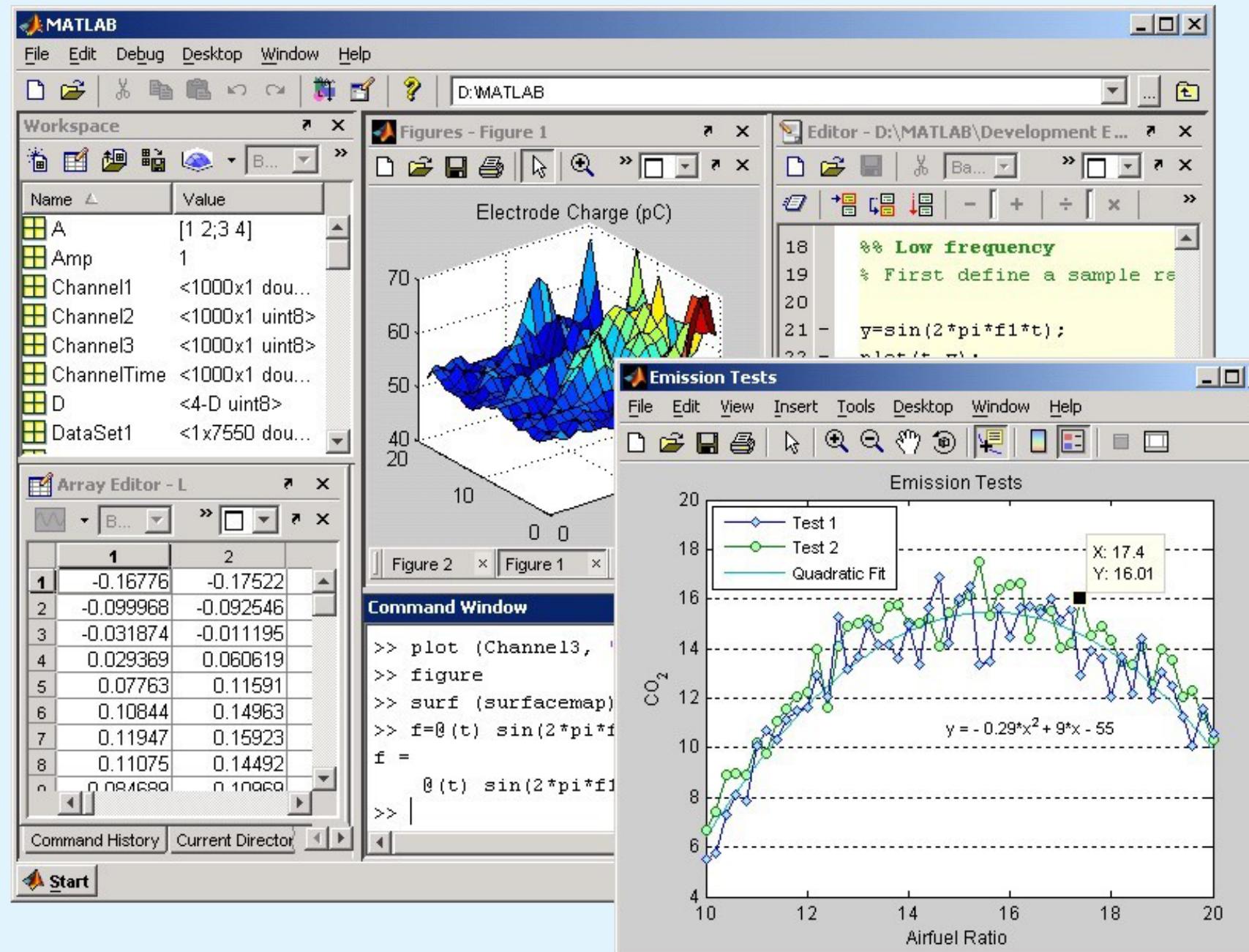


Palantir

# Simulation

# MATLAB

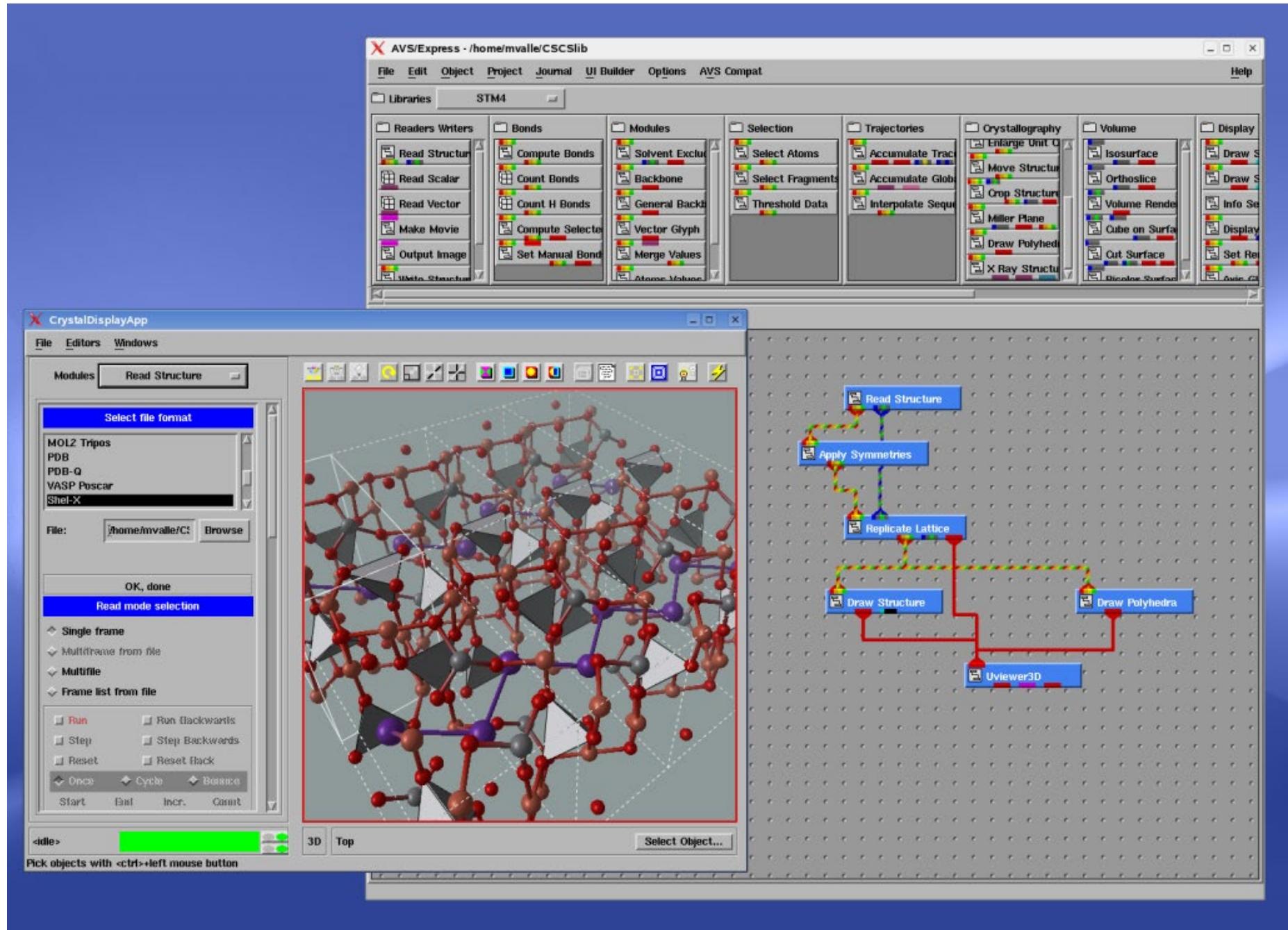
1984-Today



Cleve Moler, Steve Bangert and Jack Little for MathWorks

# AVS Express

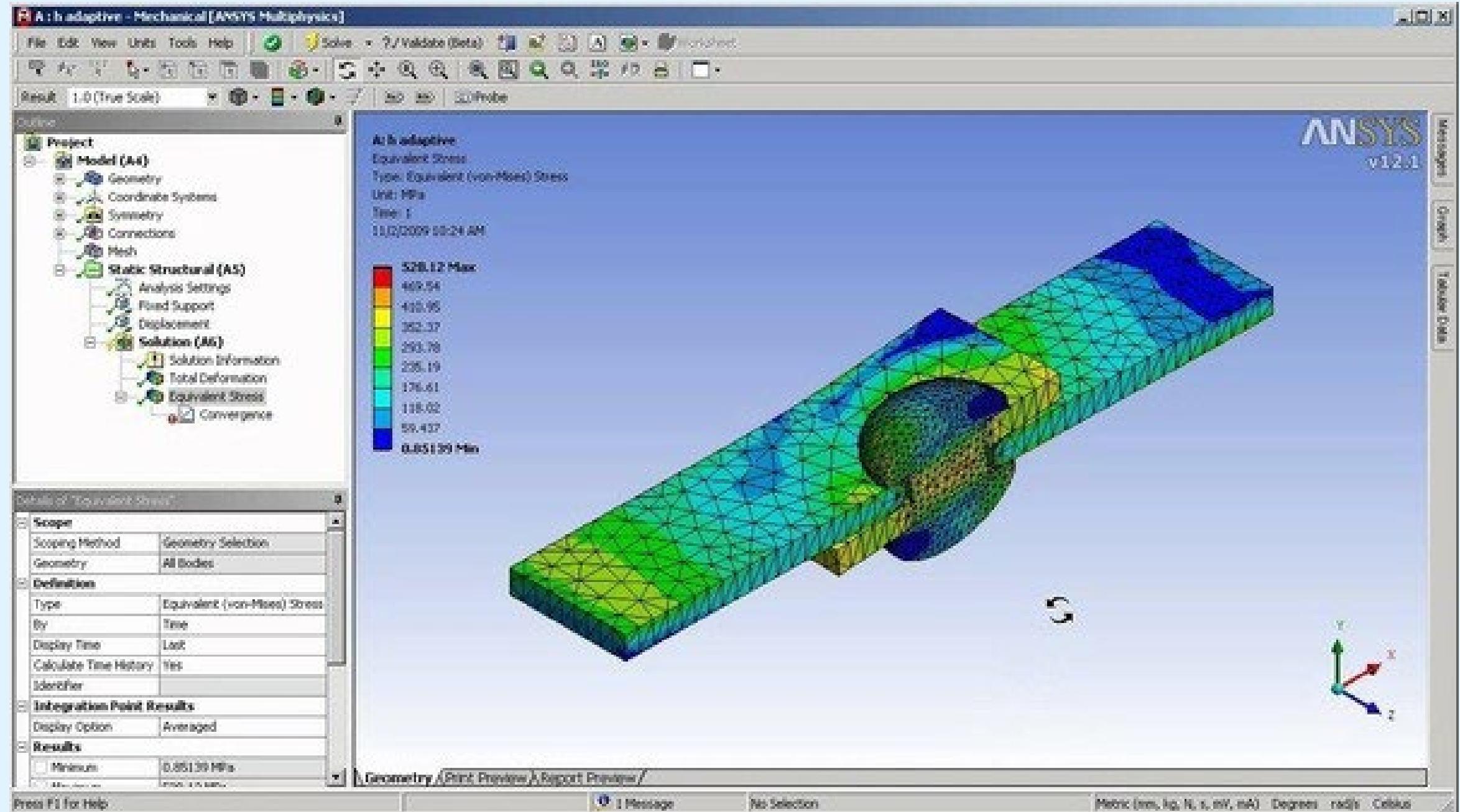
1991-Today



Advanced Visual Systems Inc

# Ansys

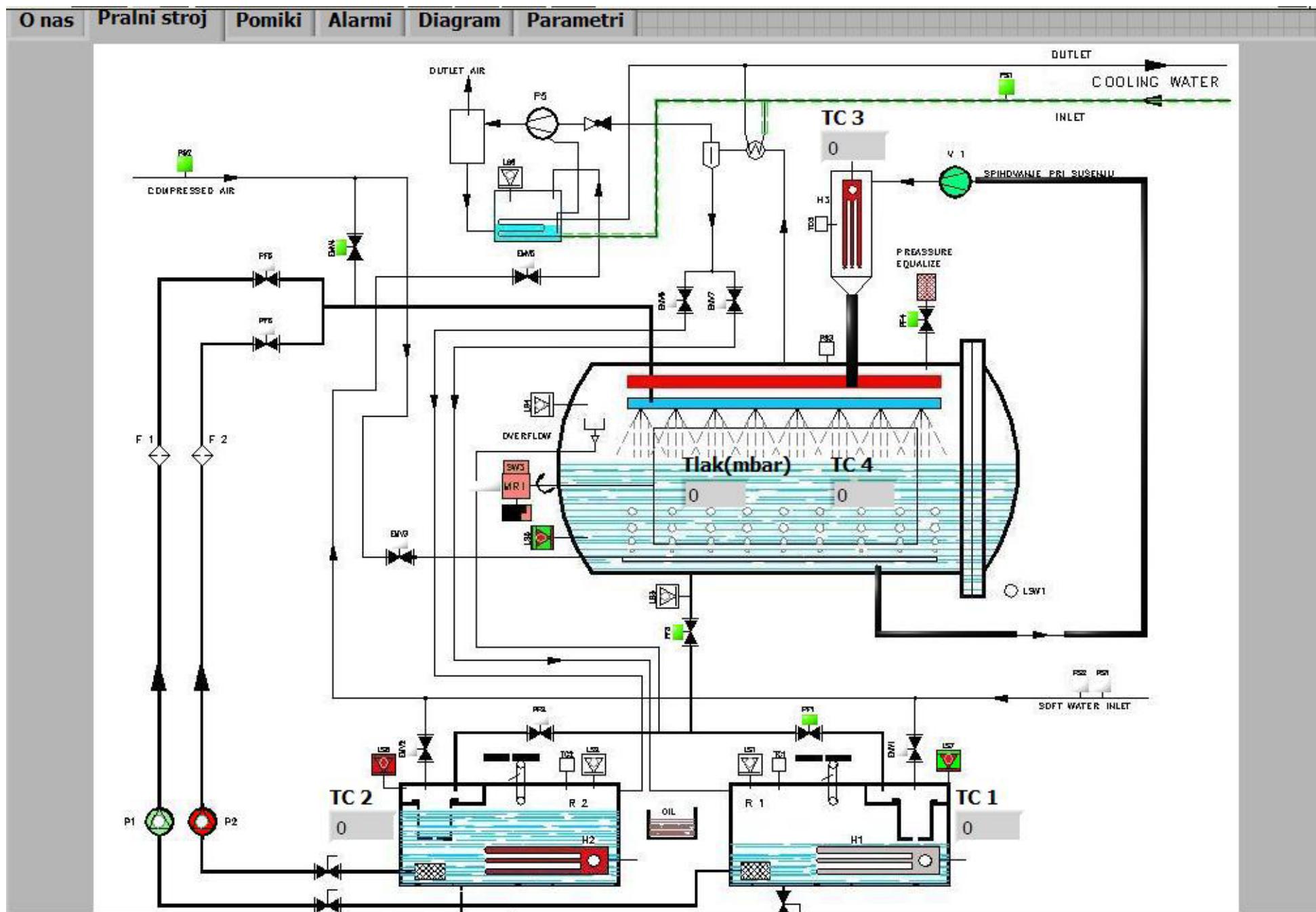
1970-Today



ANSYS Inc

# LabView

# 1986-Today

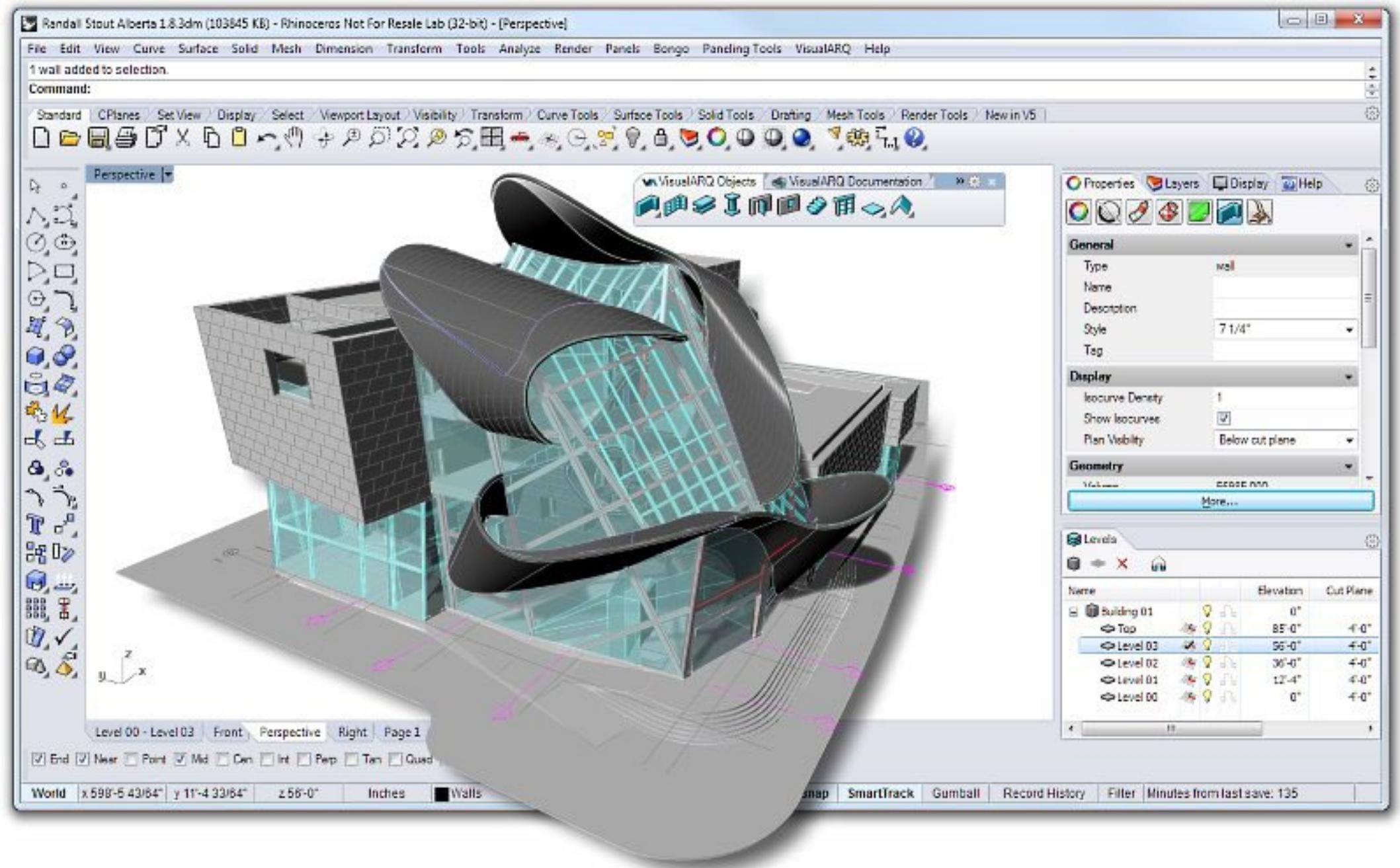


# National Instruments

# 3D (Solid Geometry) Modeling

# Rhino

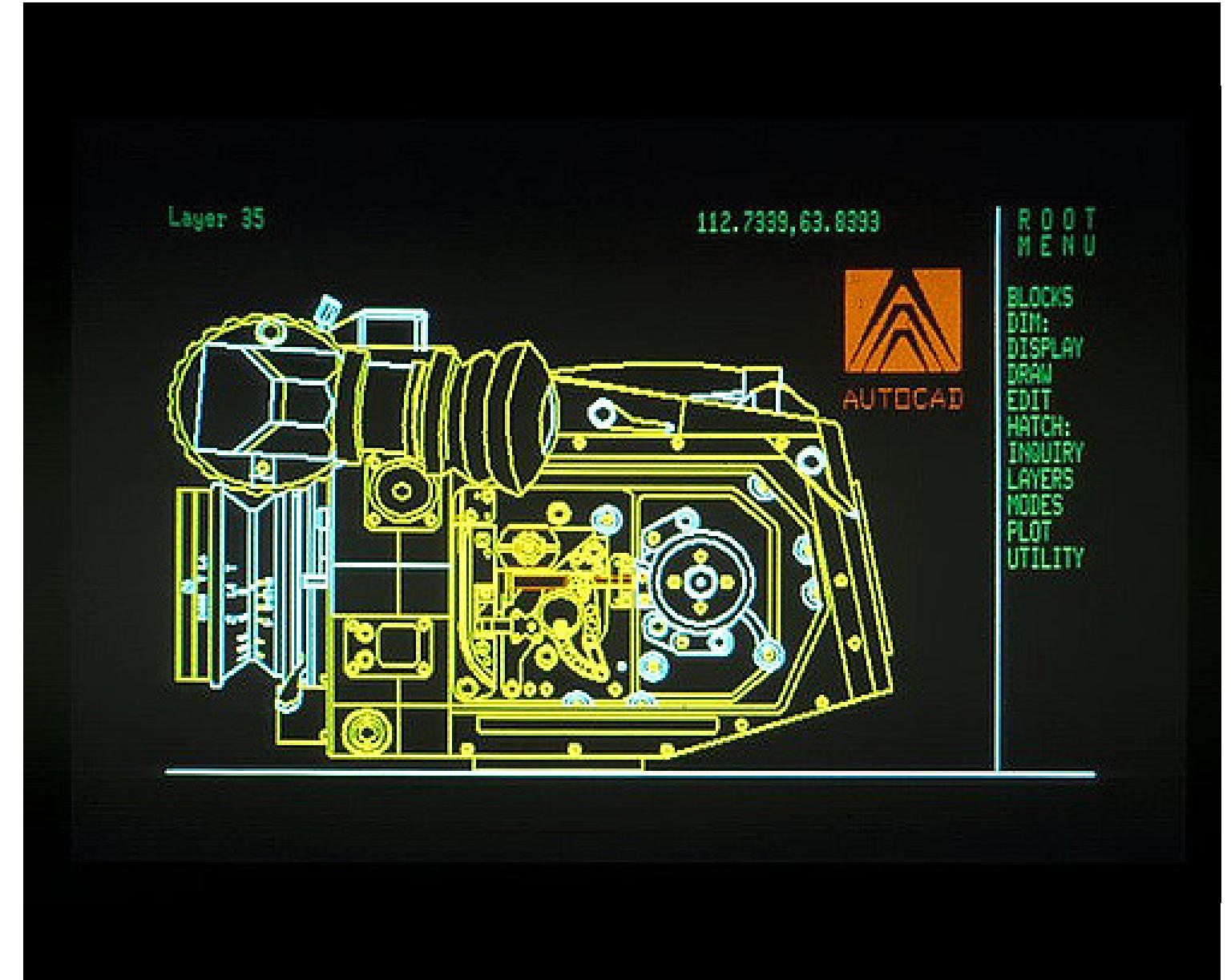
1980-Today



Robert McNeel & Associates

# AutoCAD

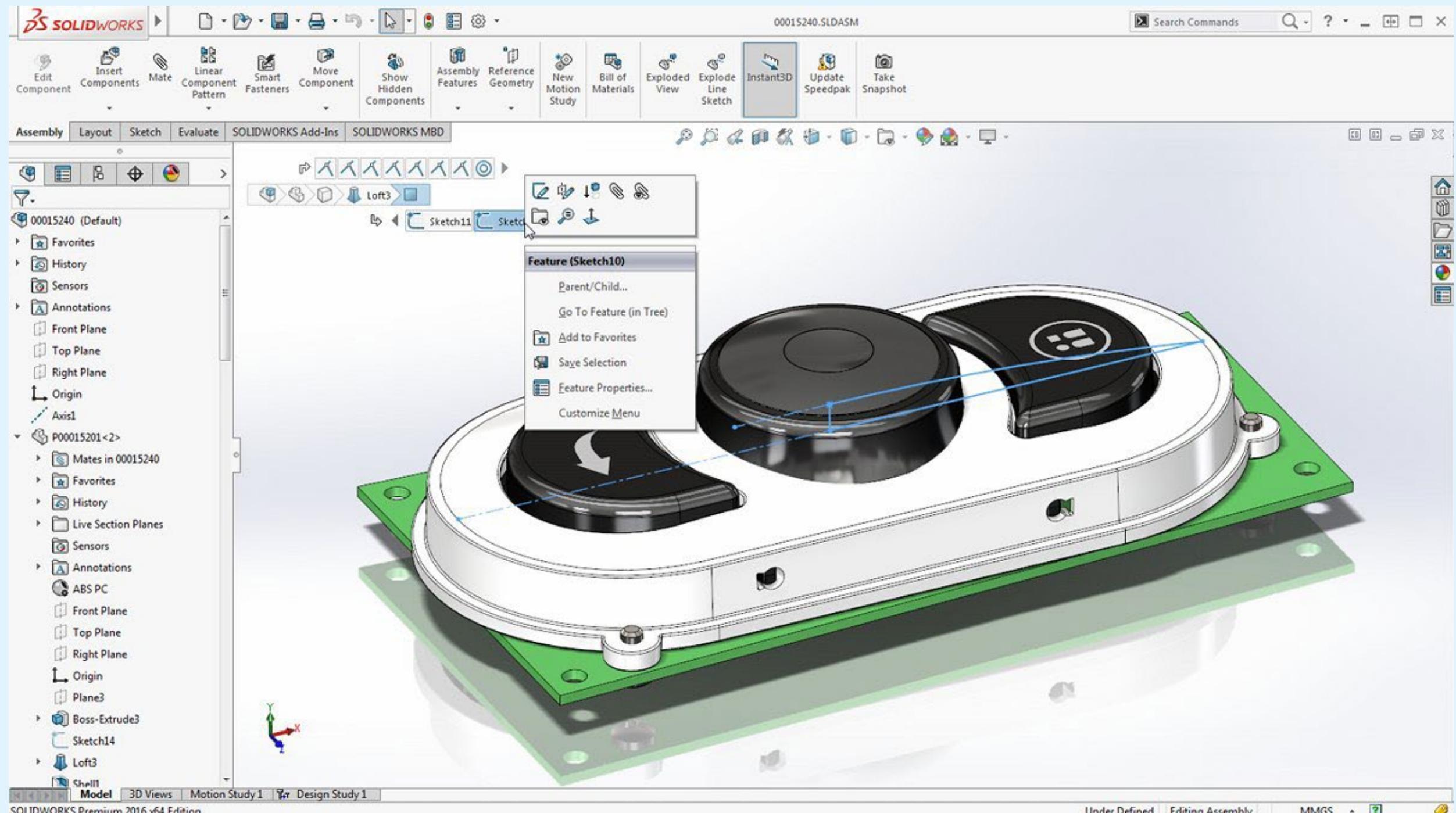
1982-Today



AutoDesk

# SolidWorks

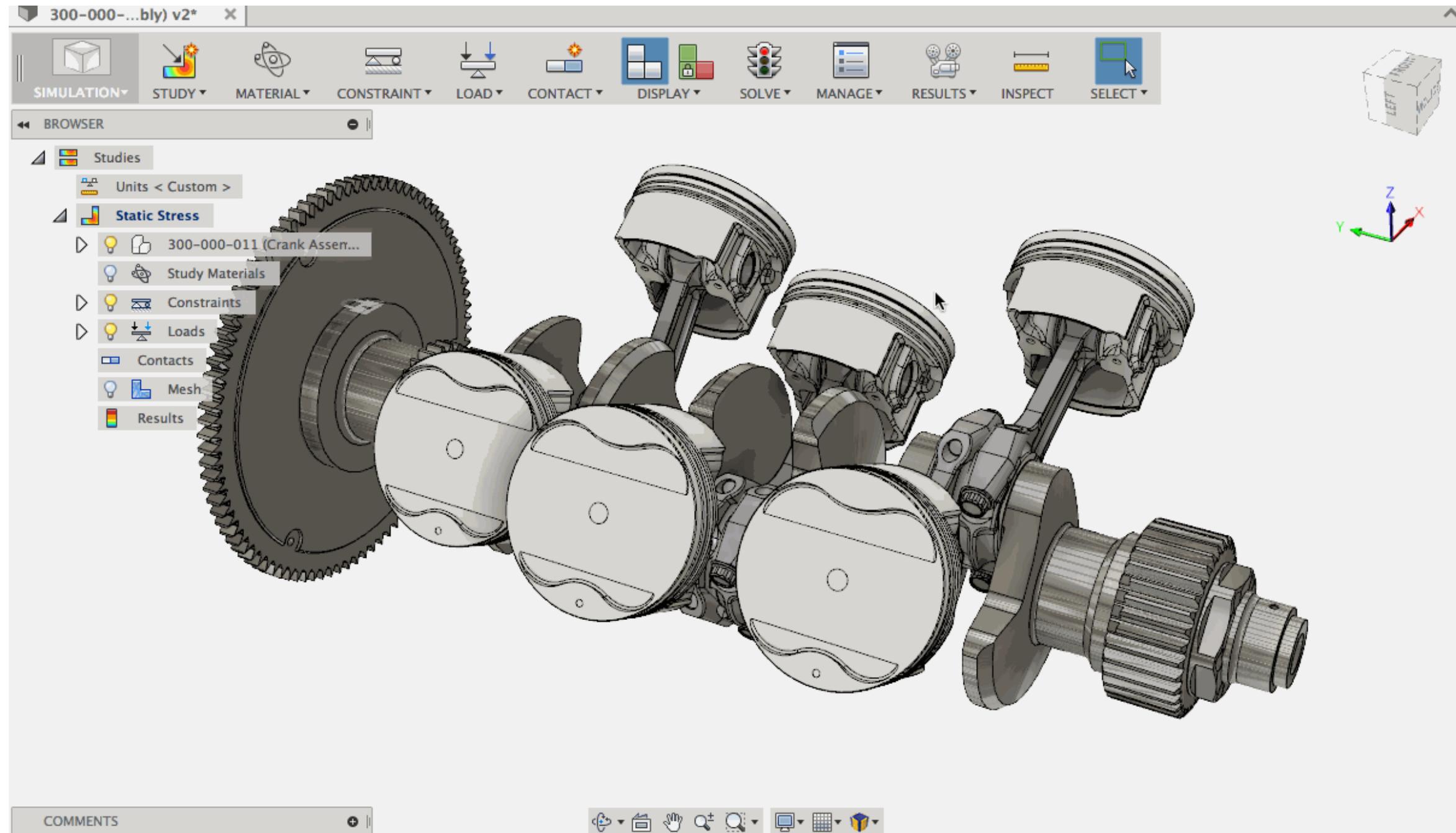
1995-Today



Dassault Systèmes

# Fusion 360

2013-Today



Autodesk

# Design Machine

## Creative Tools for Everyone

# Graphics

Direct-Manipulation +  
Computer Graphics

# Sketchpad

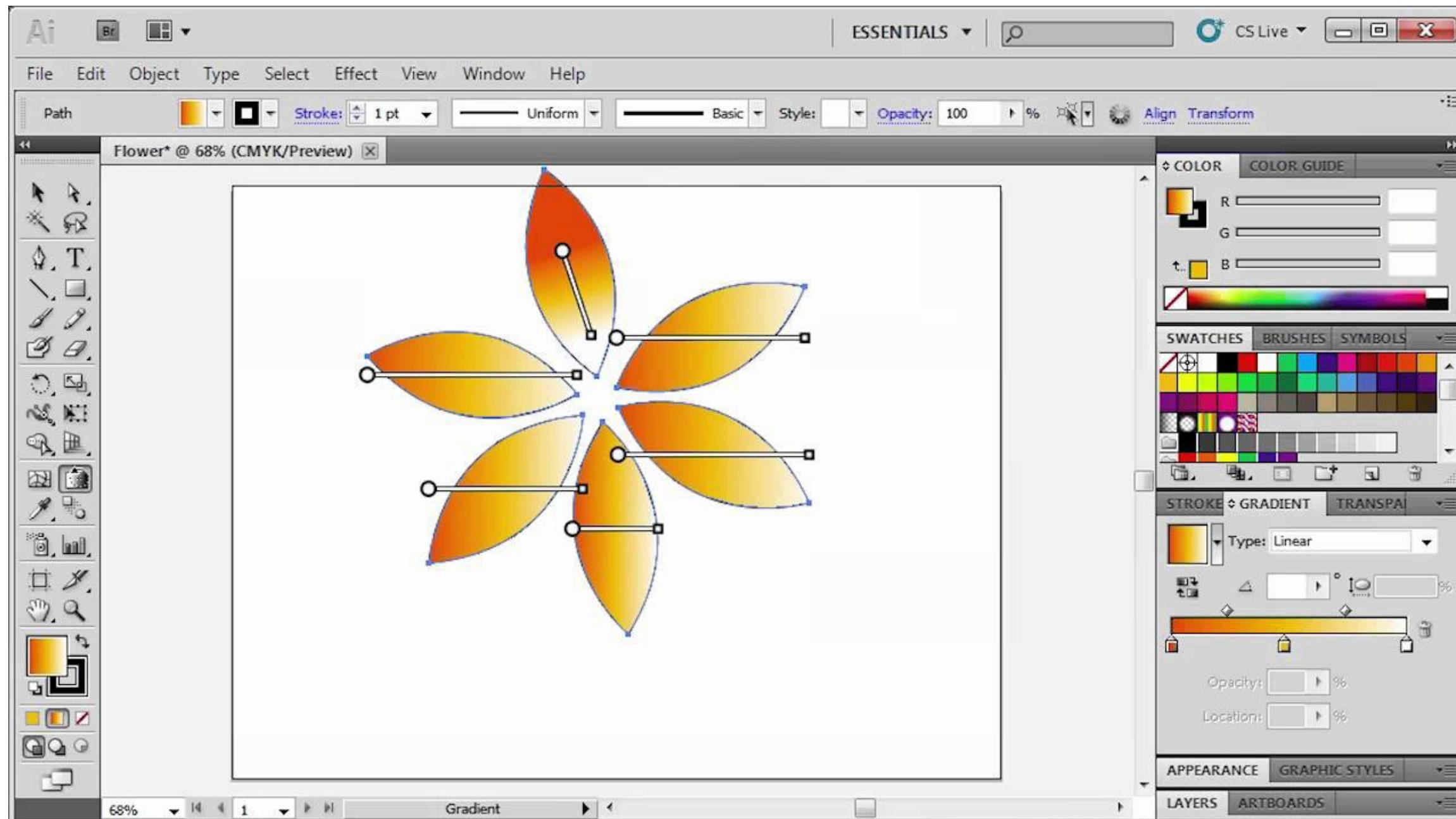
1963



Ivan Sutherland

# Illustrator

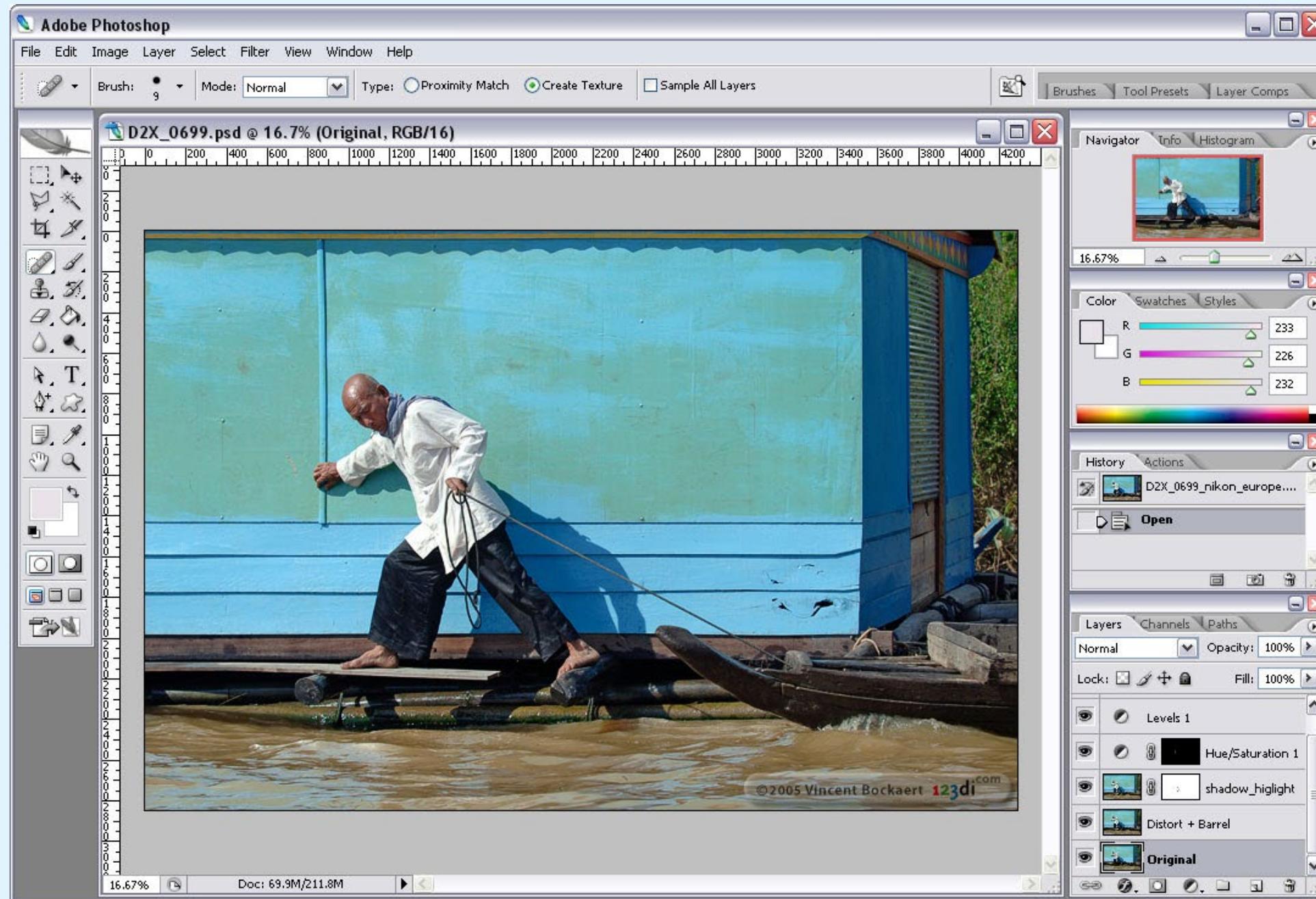
1987-Today



Adobe Systems

# Photoshop

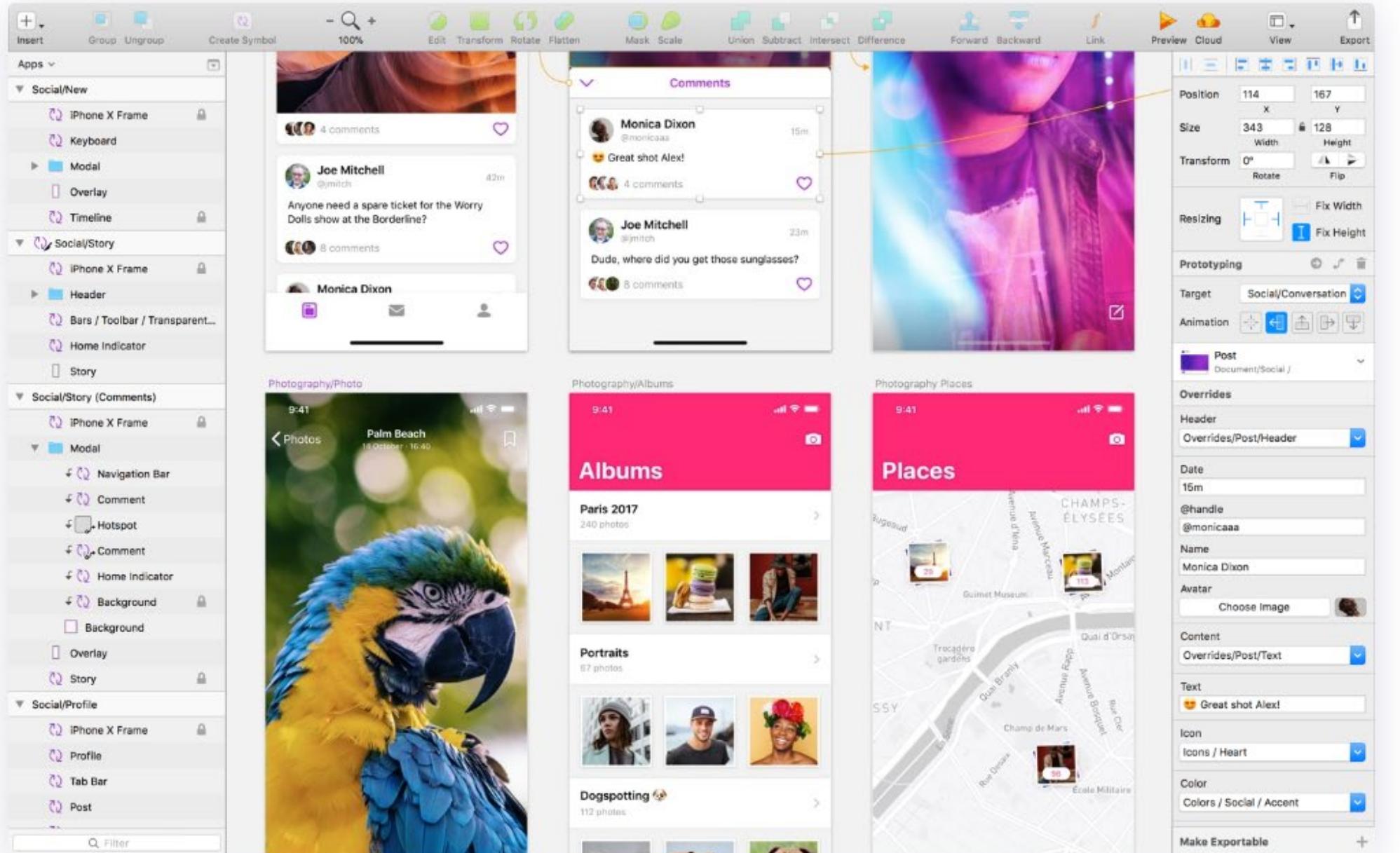
1990-Today



Adobe Systems

# Sketch

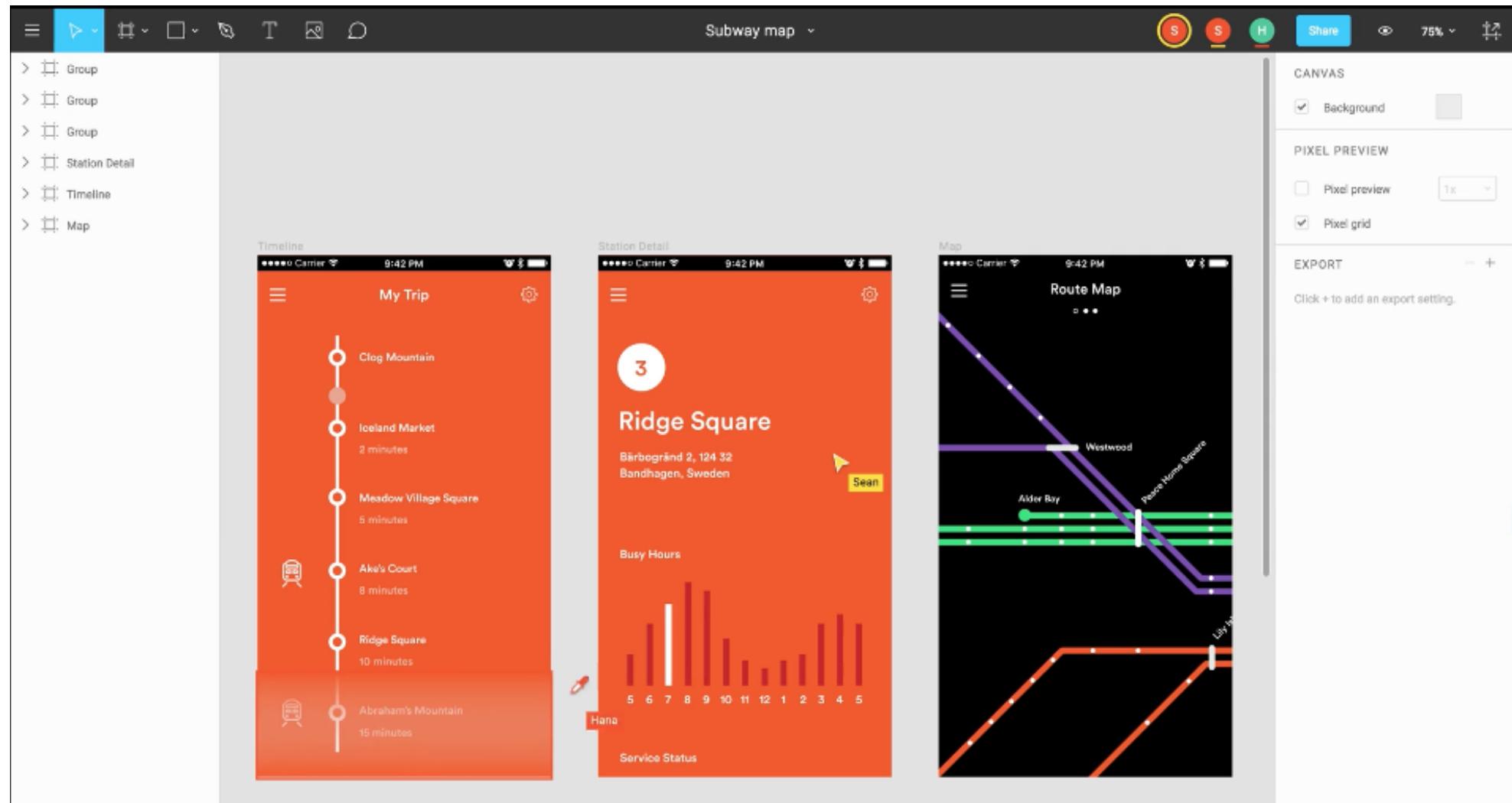
2010-Today



Bohemian Coding

# Figma

2016-Today



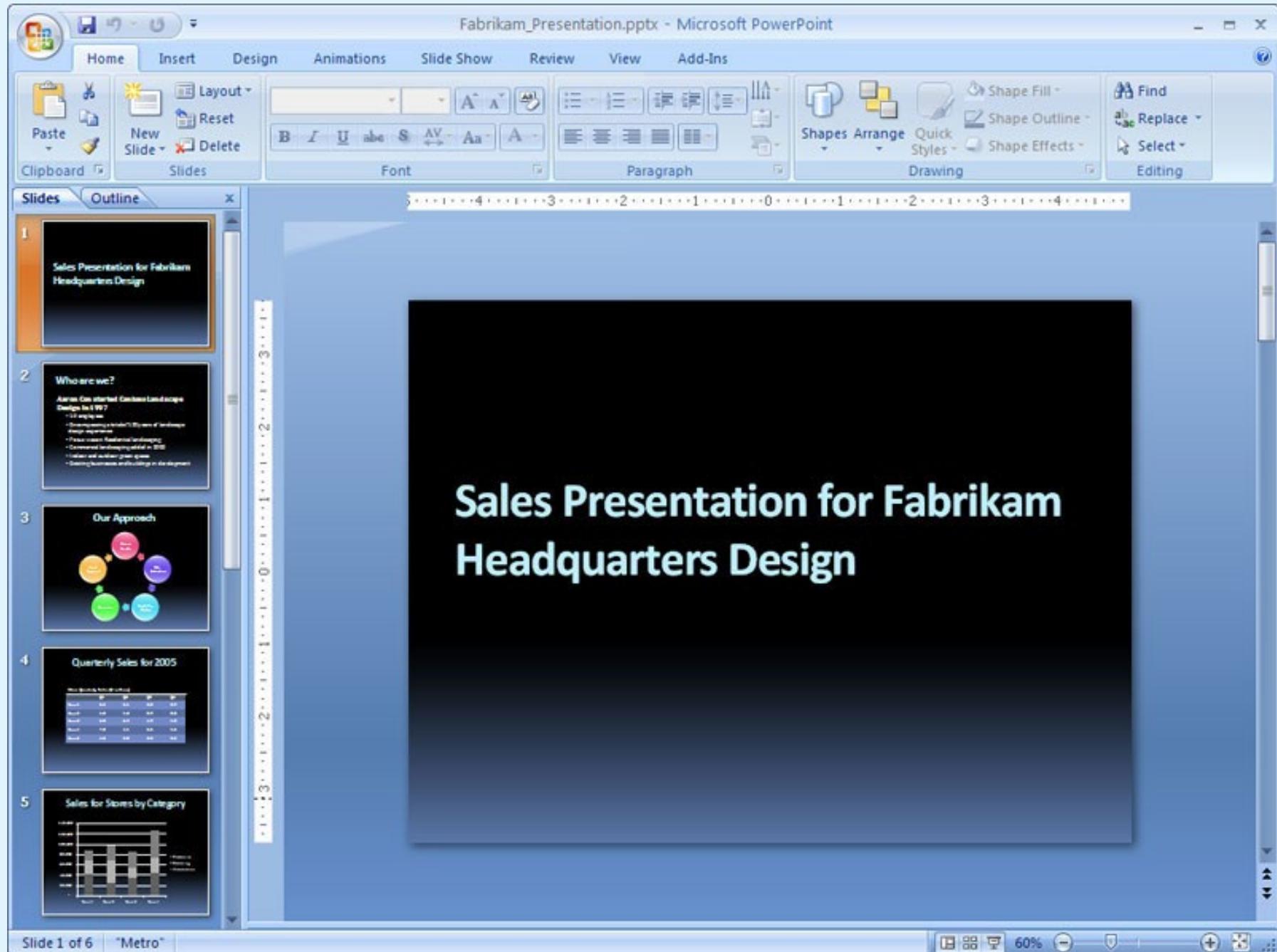
Figma

# Presentation

From Outline to Slides to Demo

# Powerpoint

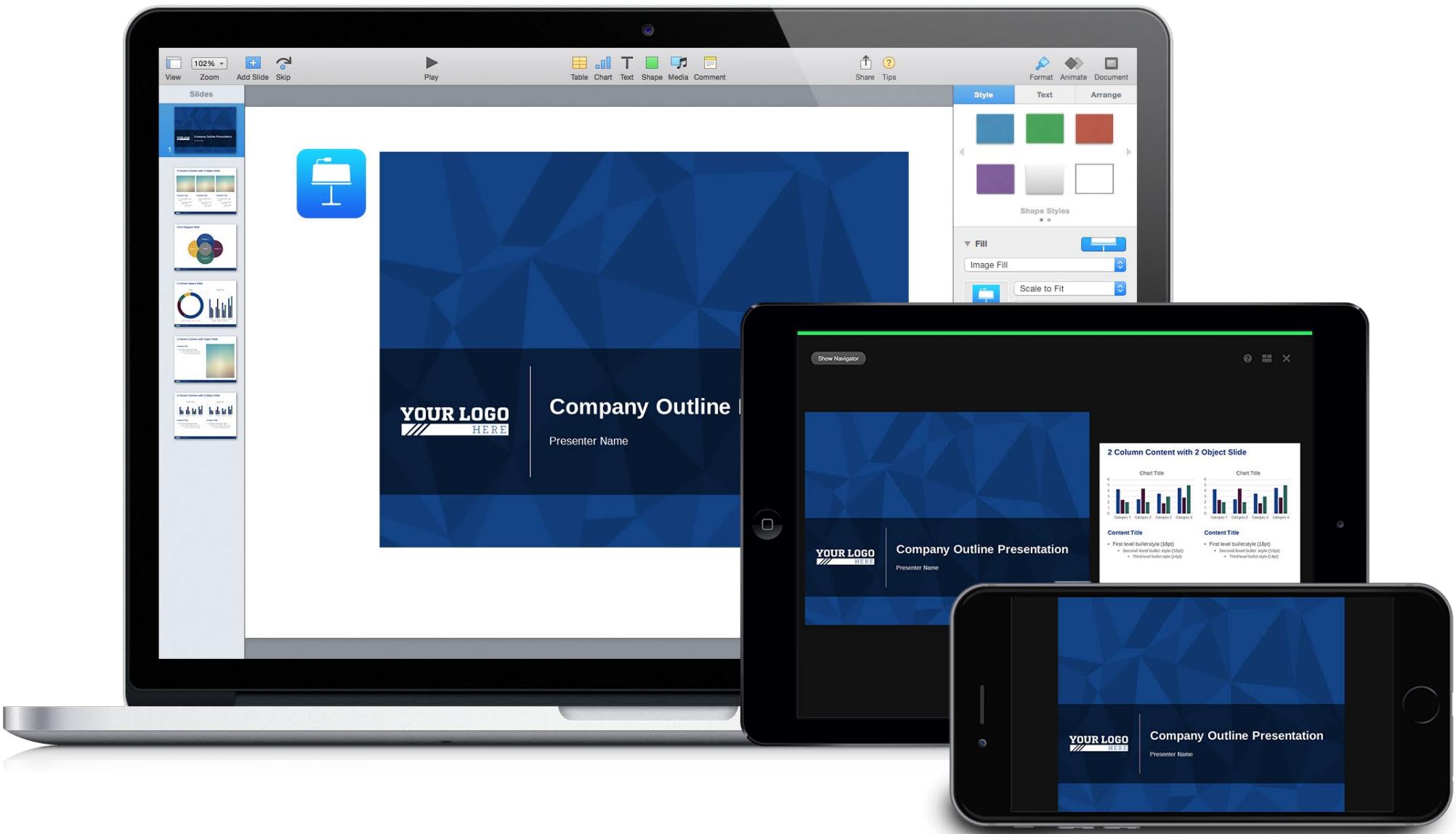
1987-Today



Microsoft

# Keynote

2003-Today



Apple

# Slides

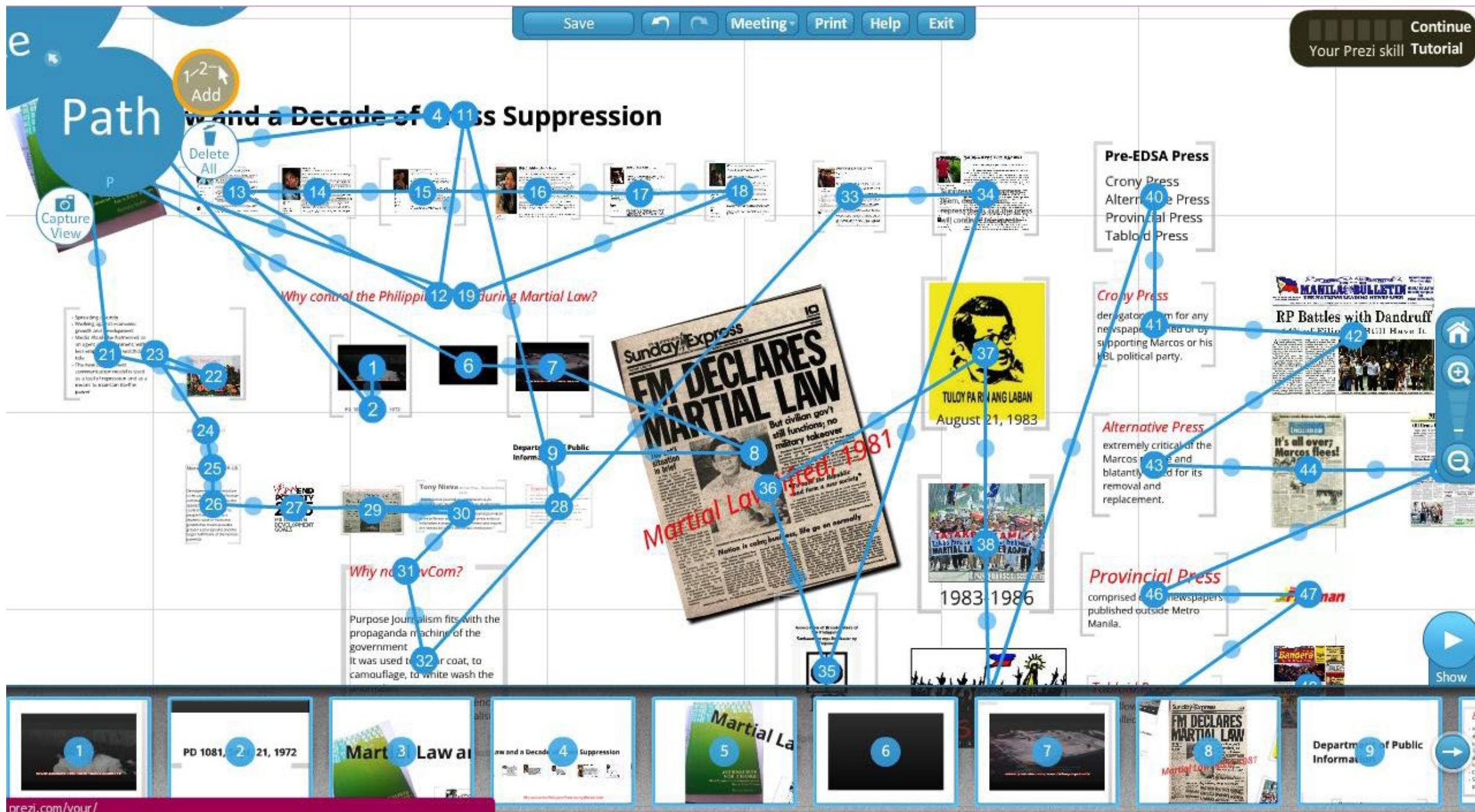
2006-Today

The screenshot shows the Google Slides interface. On the left, a slide titled "CLICK TO ADD TITLE" is displayed with a teal subtitle box containing "Click to add subtitle". Below the slide, a notes section says "Click to add notes". At the top, the menu bar includes "File", "Edit", "View", "Insert", "Slide", "Format", "Arrange", "Tools", "Table", "Help", and "All changes saved in Drive". The "Slide" menu is open, showing options like "Present", "Comments", and "Share". A "Themes" dialog box is open on the right, listing five themes: "Simple Light" (black background), "Simple Dark" (blue background), "Material" (orange background), "Swiss" (white background), and "Import theme". Each theme preview shows a "Click to add title" and "Click to add subtitle" placeholder.

Google

# Prezi

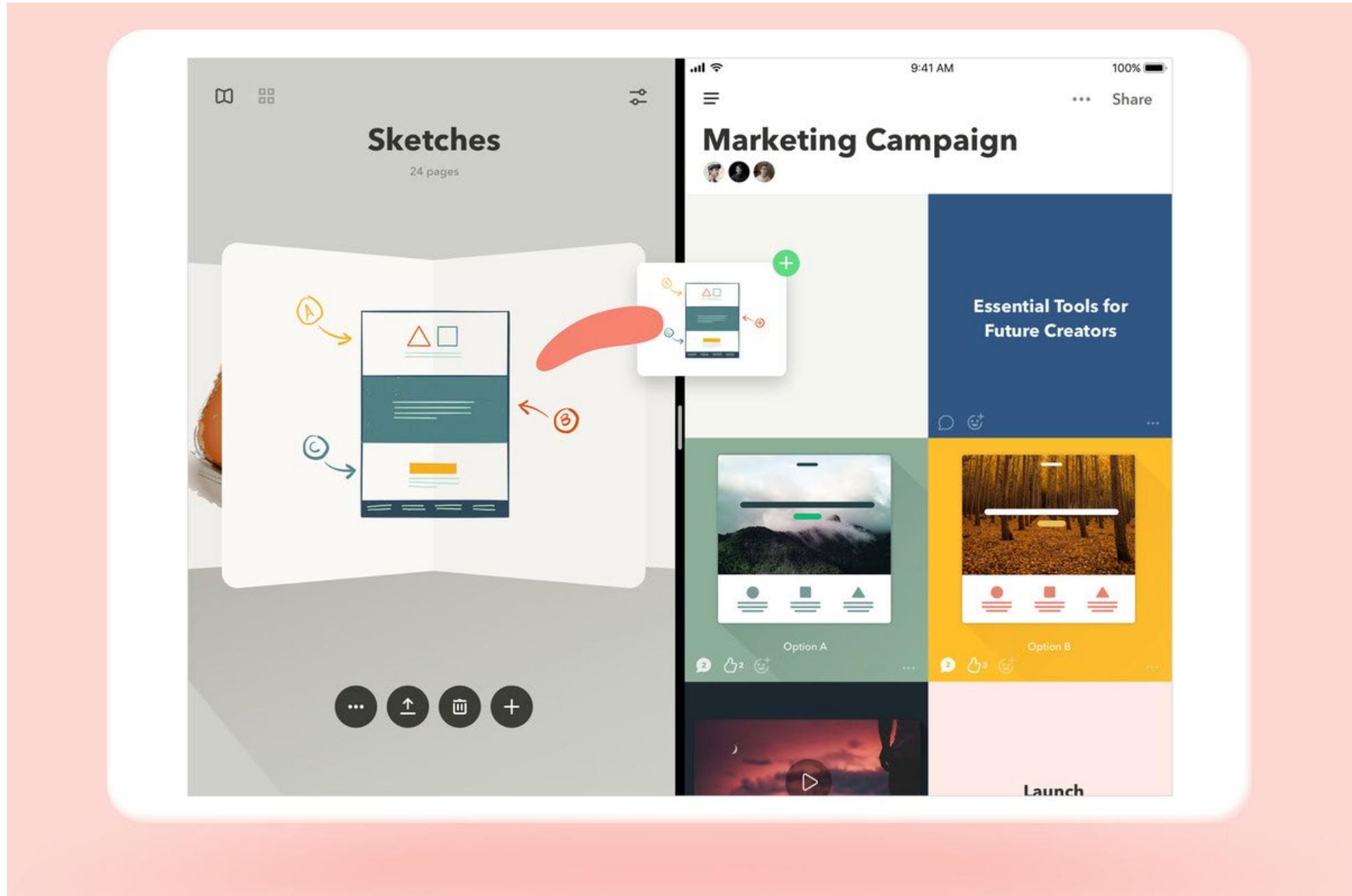
2009-Today



Adam Somlai-Fischer, Peter Halacsy and Peter Arvai

# Paste

2017-Today

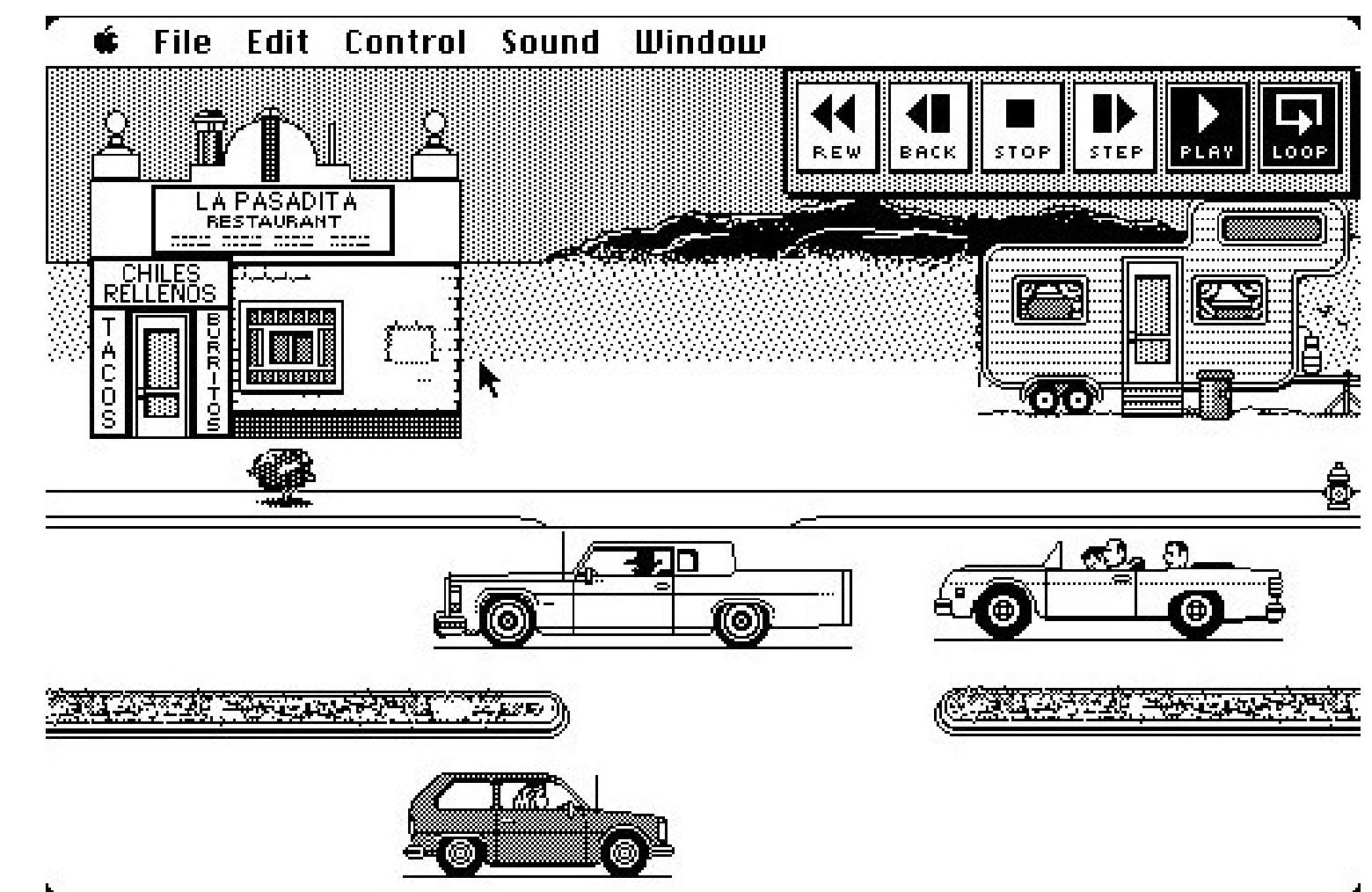
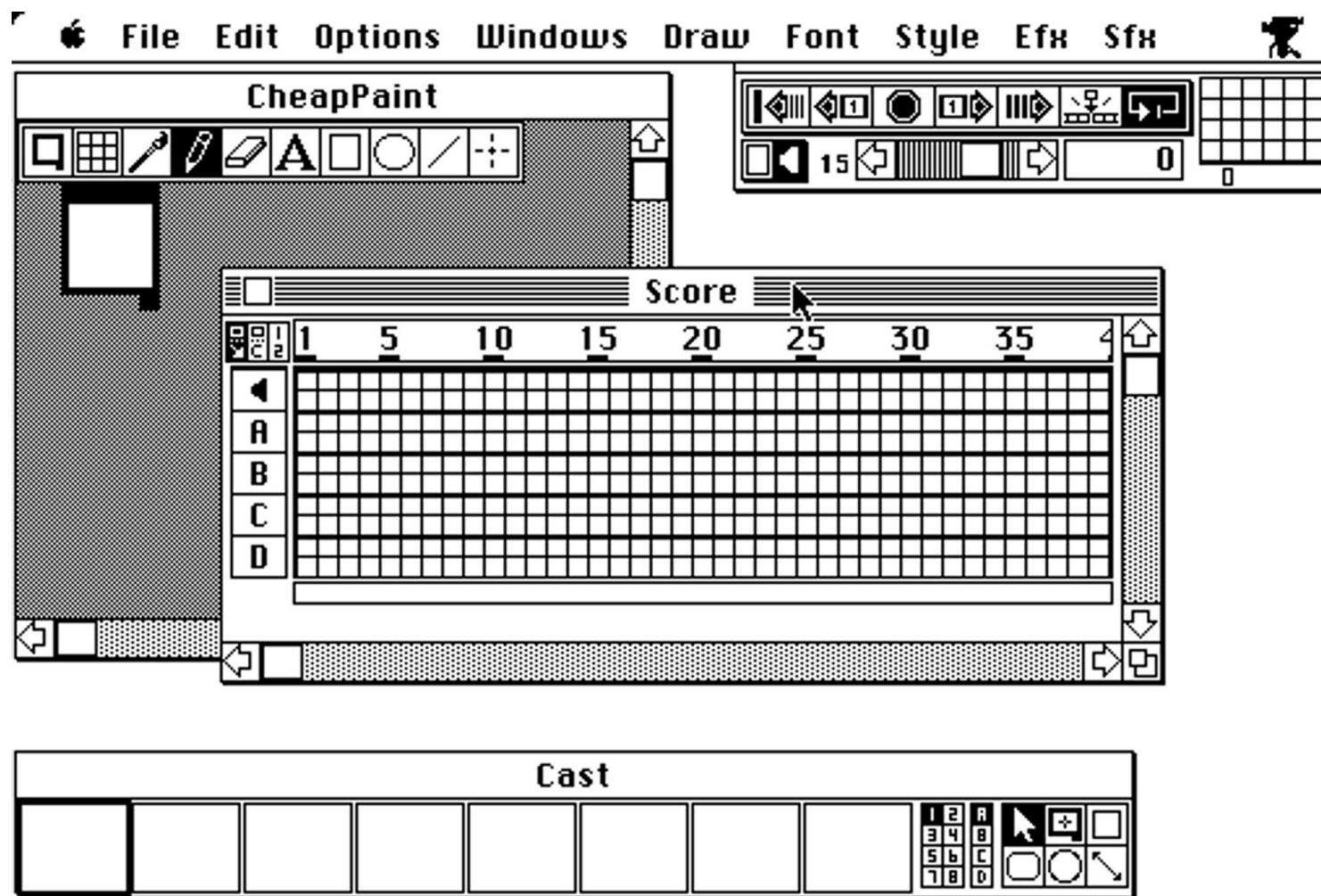


FiftyThree

# Timeline-Based Applications

# VideoWorks

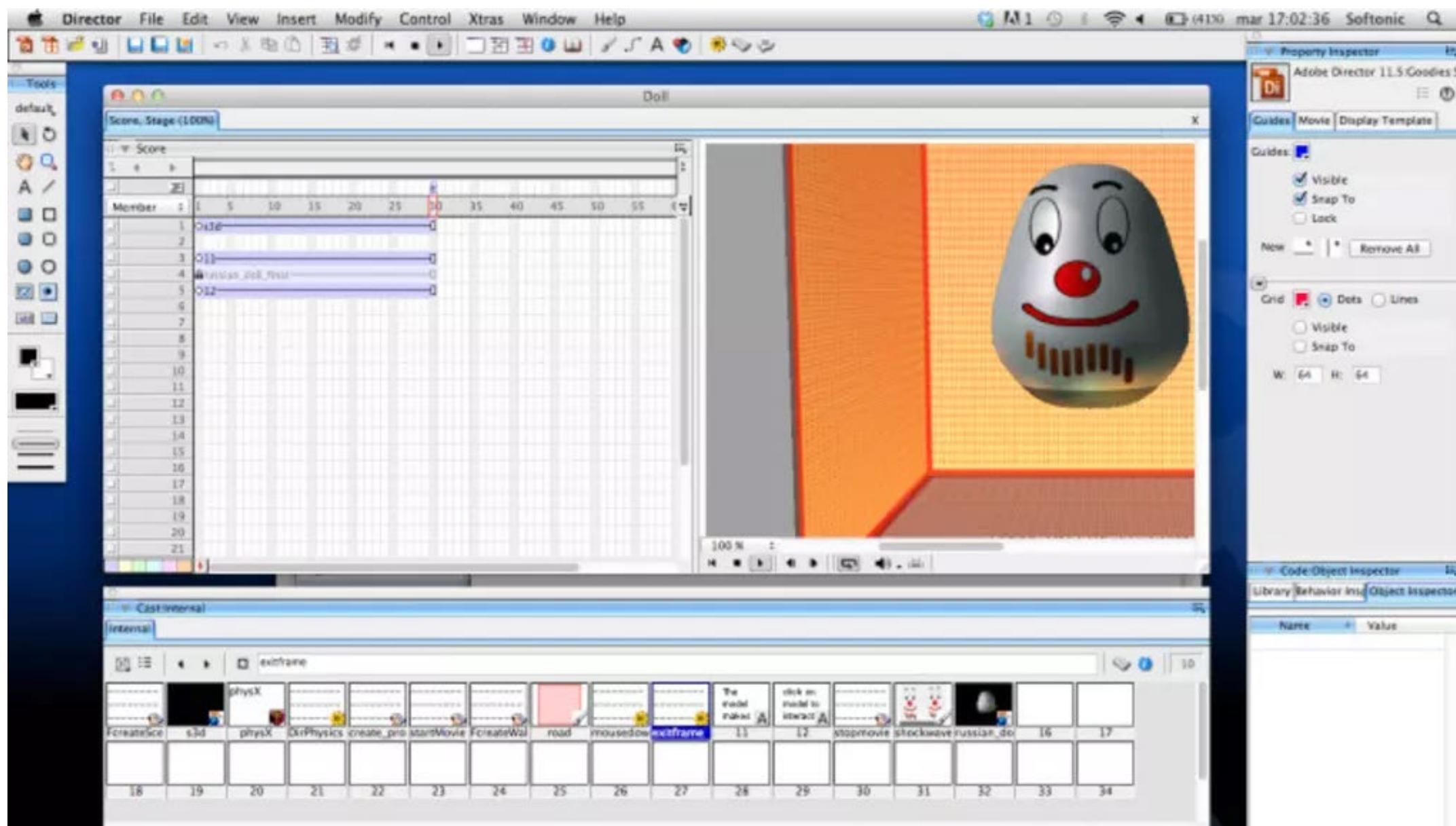
1985-1987



Marc Canter, Jay Fenton and Mark Pierce and Dan Sadowski for MacroMind

# Director

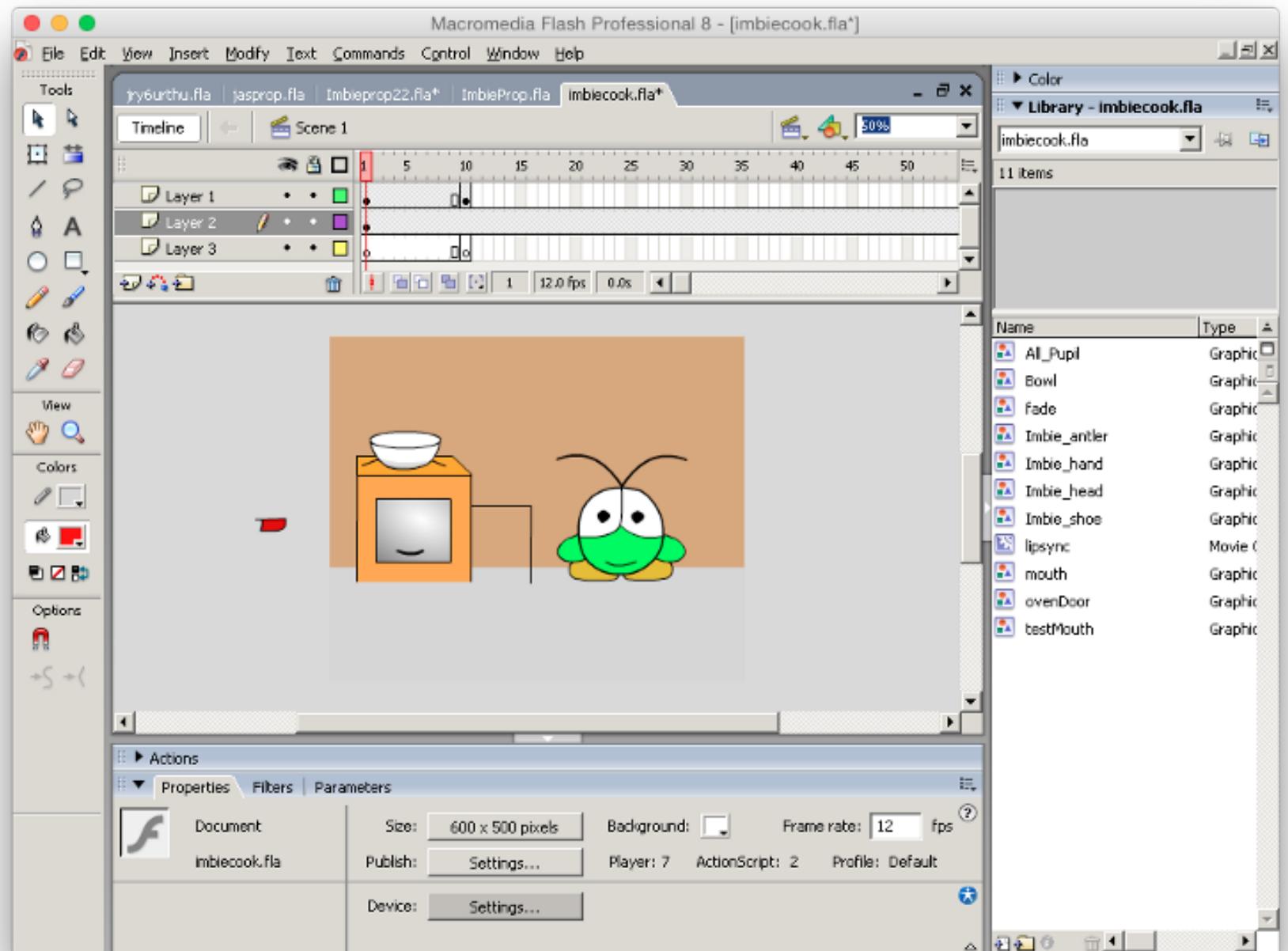
1988-2013



MacroMedia (Later Adobe)

# Flash

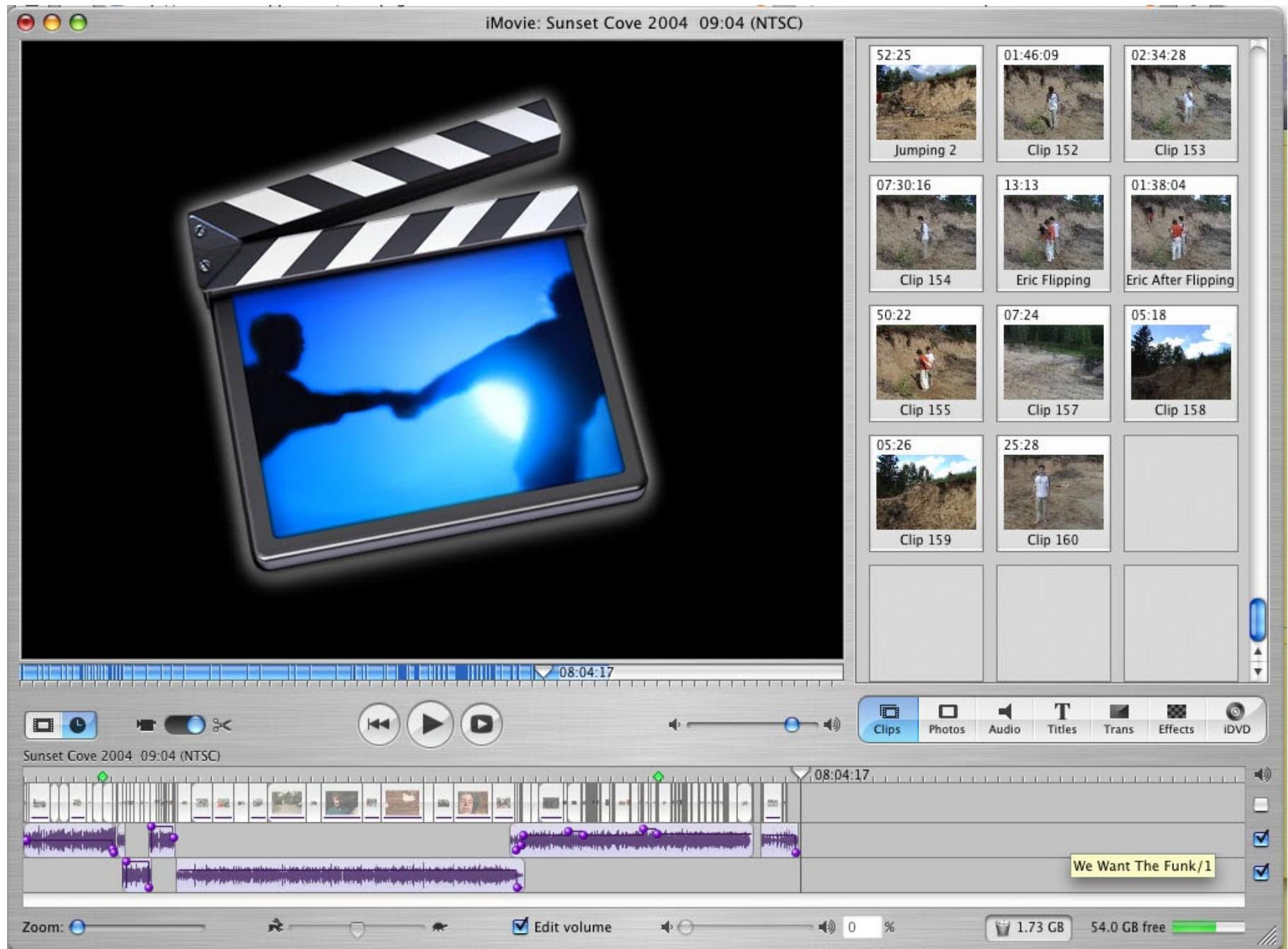
# 1996-2016



Adobe

# iMovie

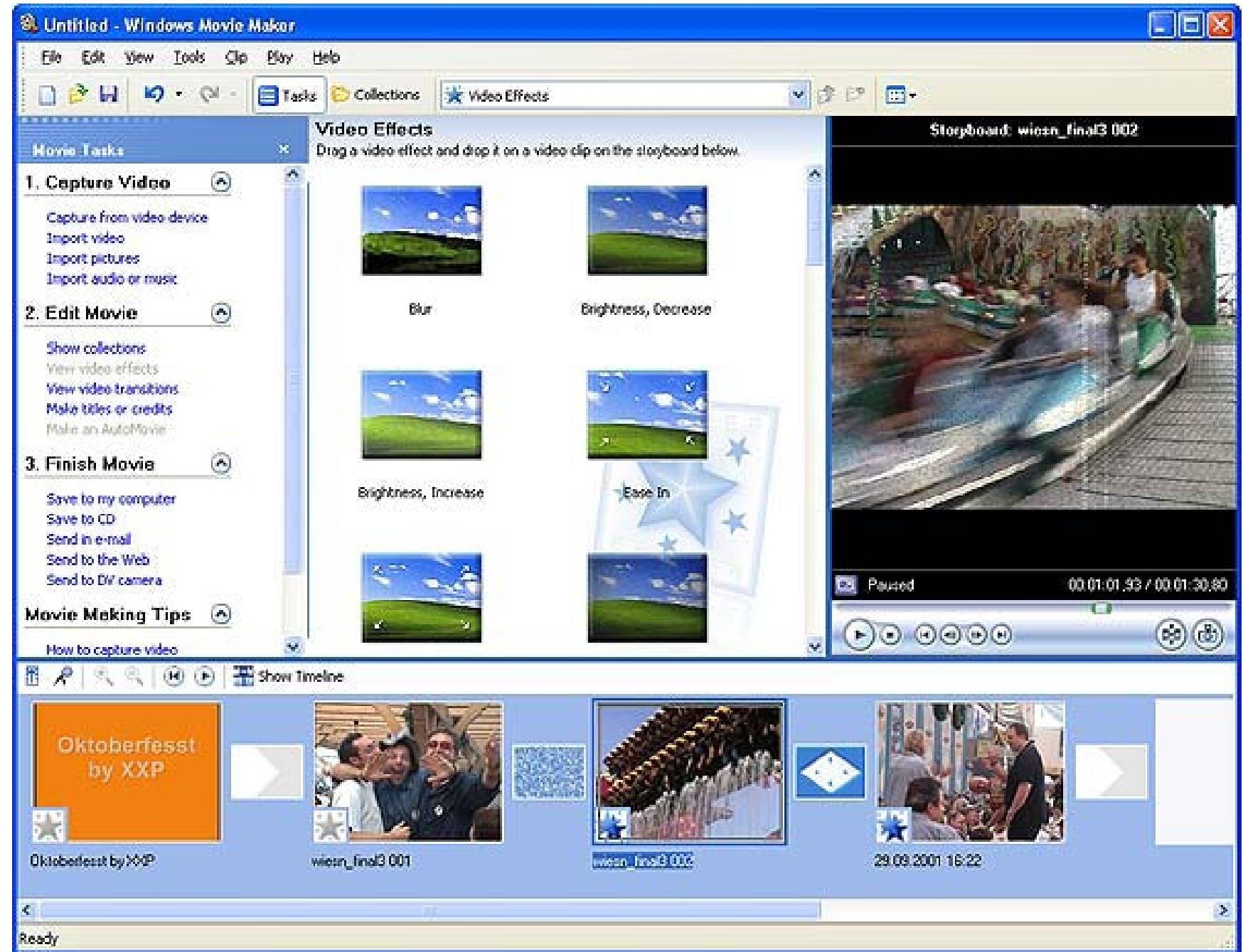
1999-Today



Apple

# Windows Movie Maker

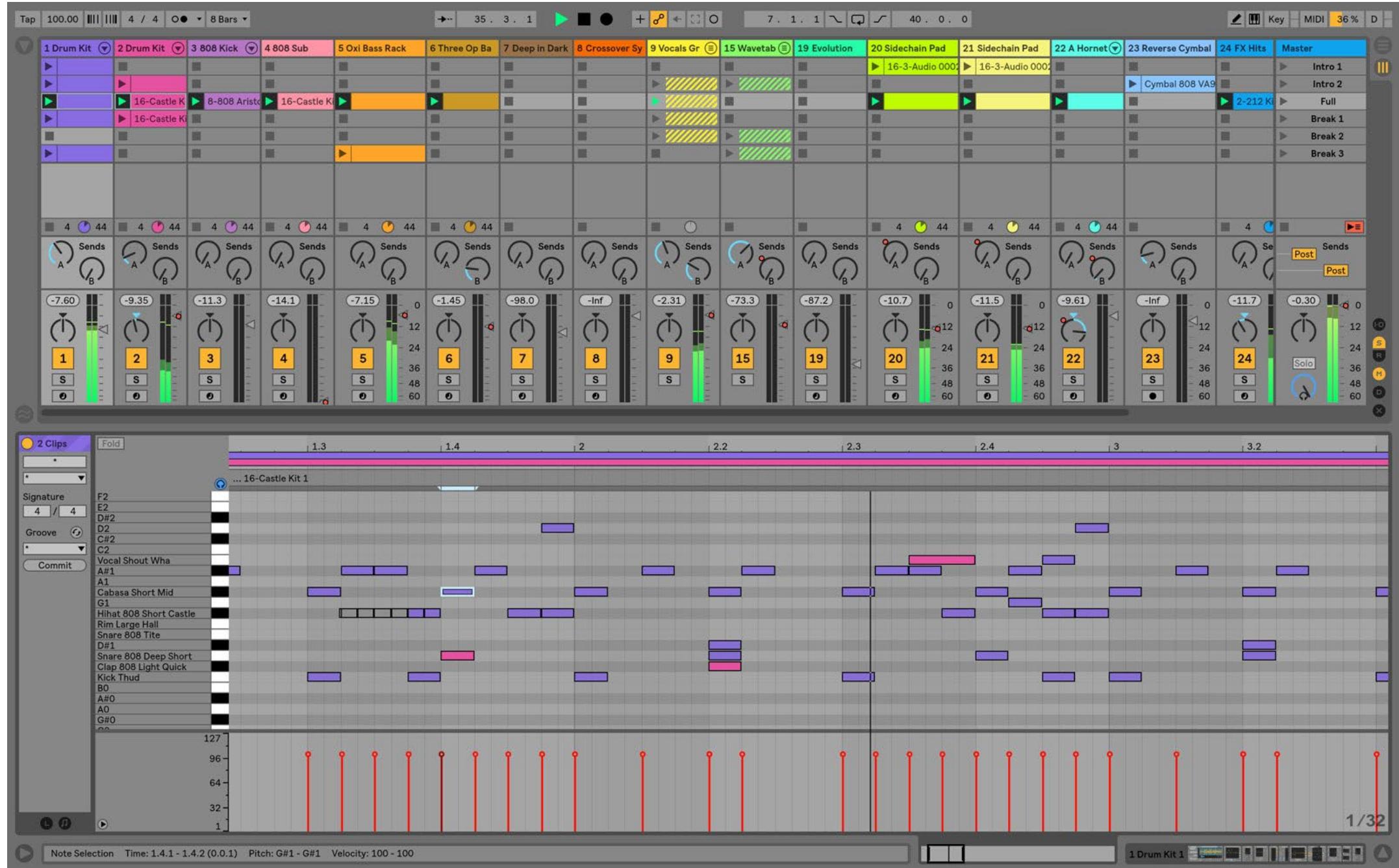
2000-2012



Microsoft

# Ableton Live

2001-Today



Ableton

# GarageBand

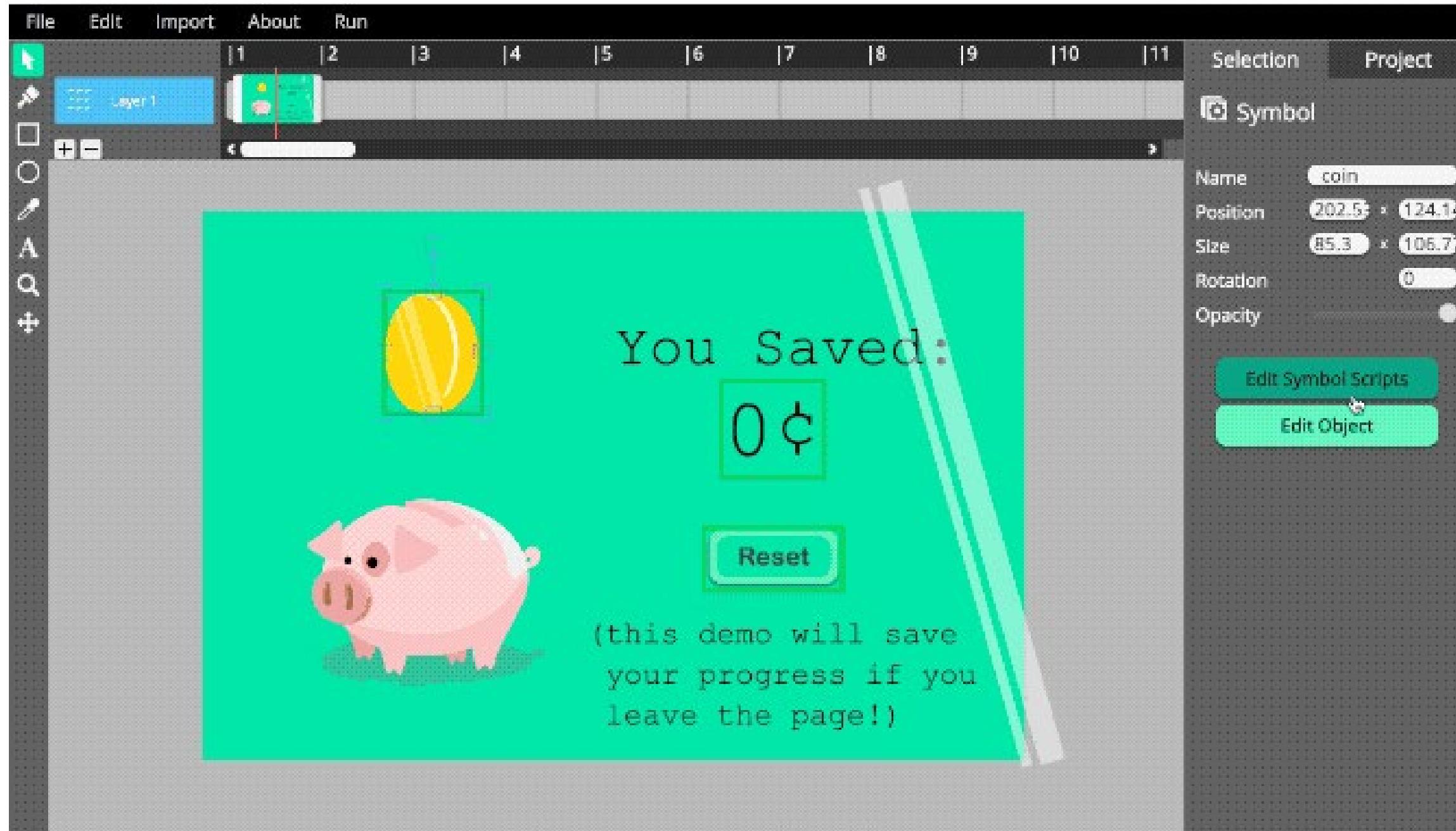
2004-Today



Apple

# Wick Editor

2017-Today



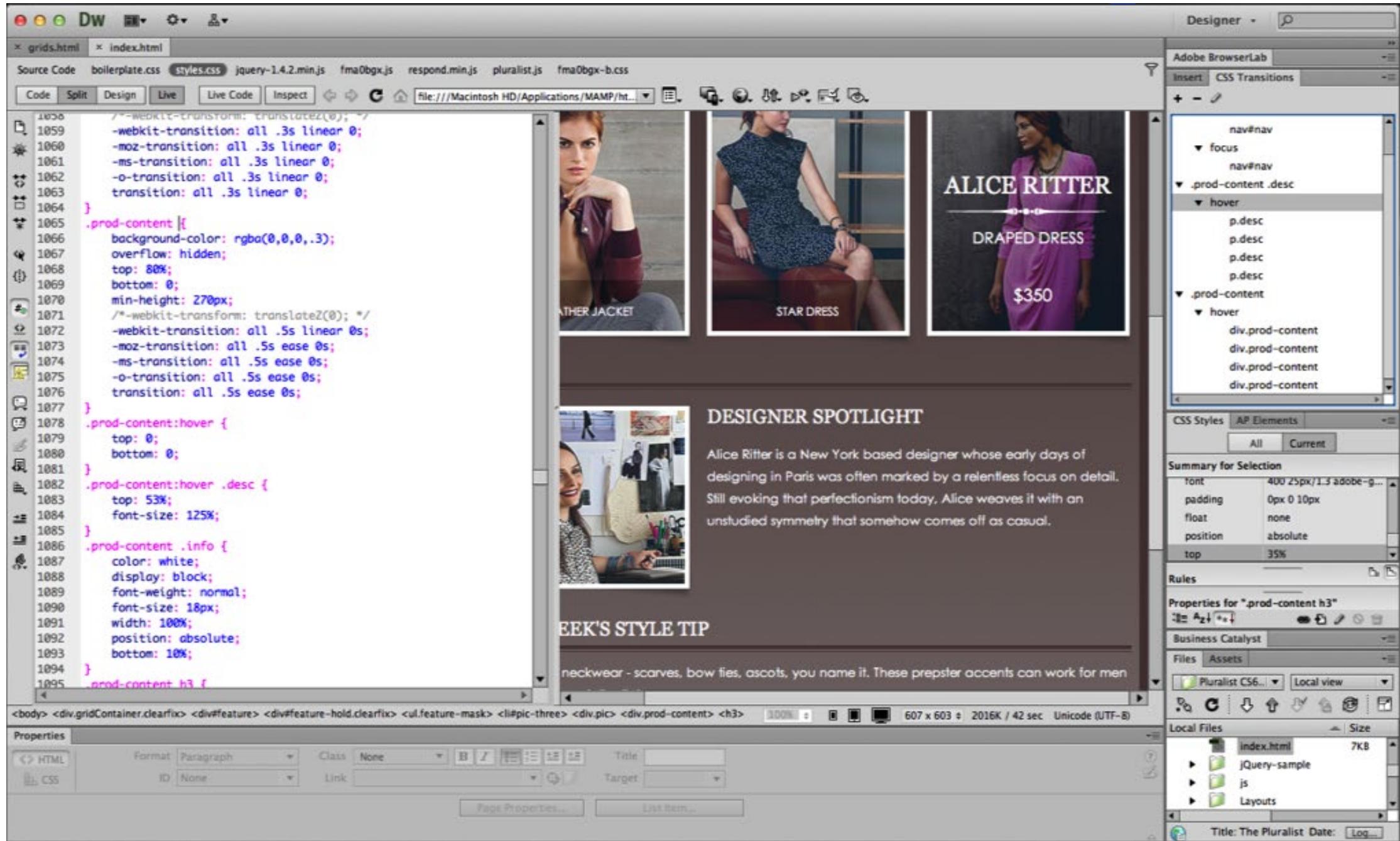
Luca Damasco and Zach Rispoli at the Frank-Ratchye STUDIO for Creative Inquiry at Carnegie Mellon University

# World Wide Web

## Design & Development Environments

# Dreamweaver

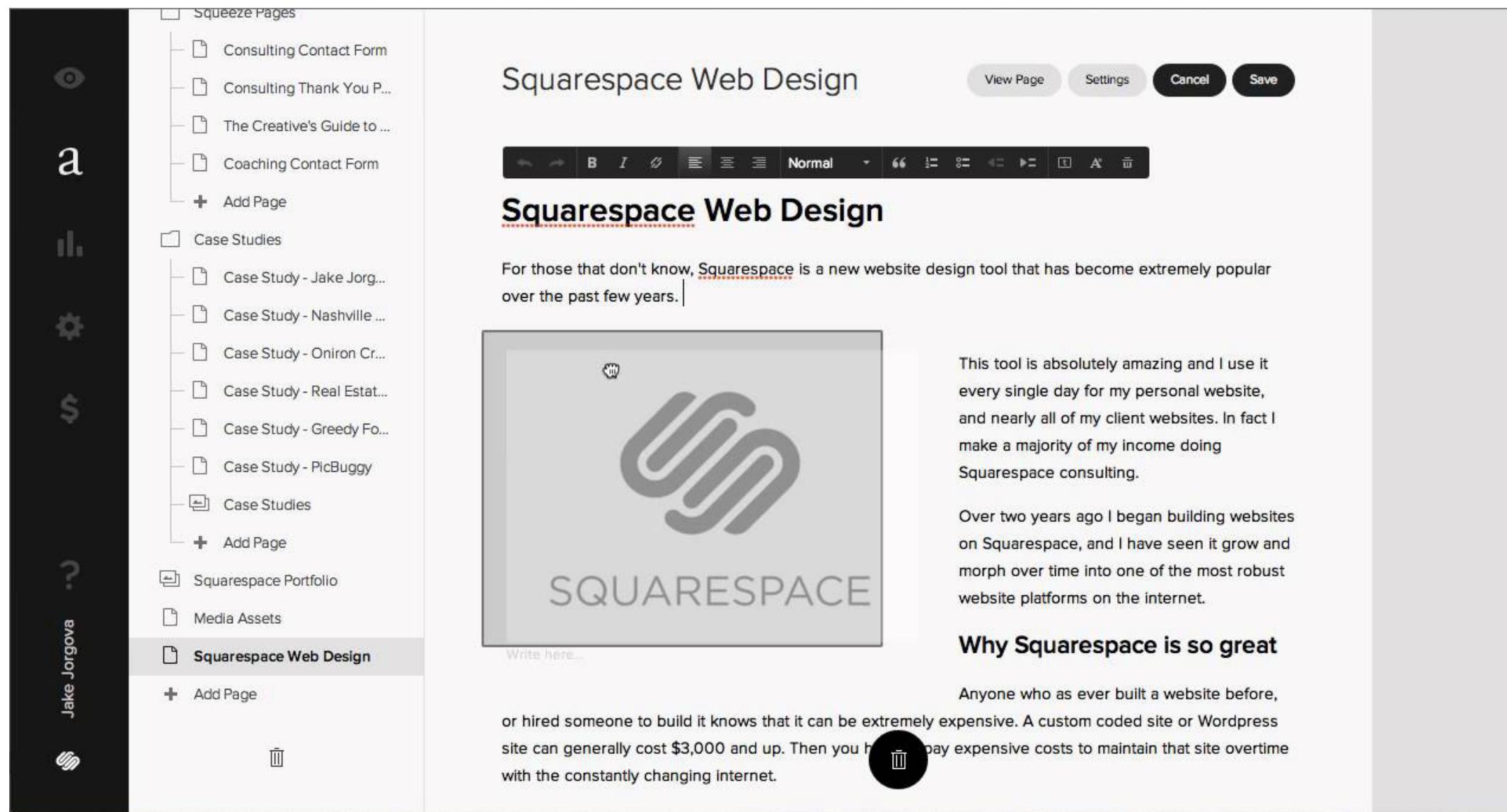
1997-Today



Macromedia (Later Adobe)

# Squarespace

2003-Today

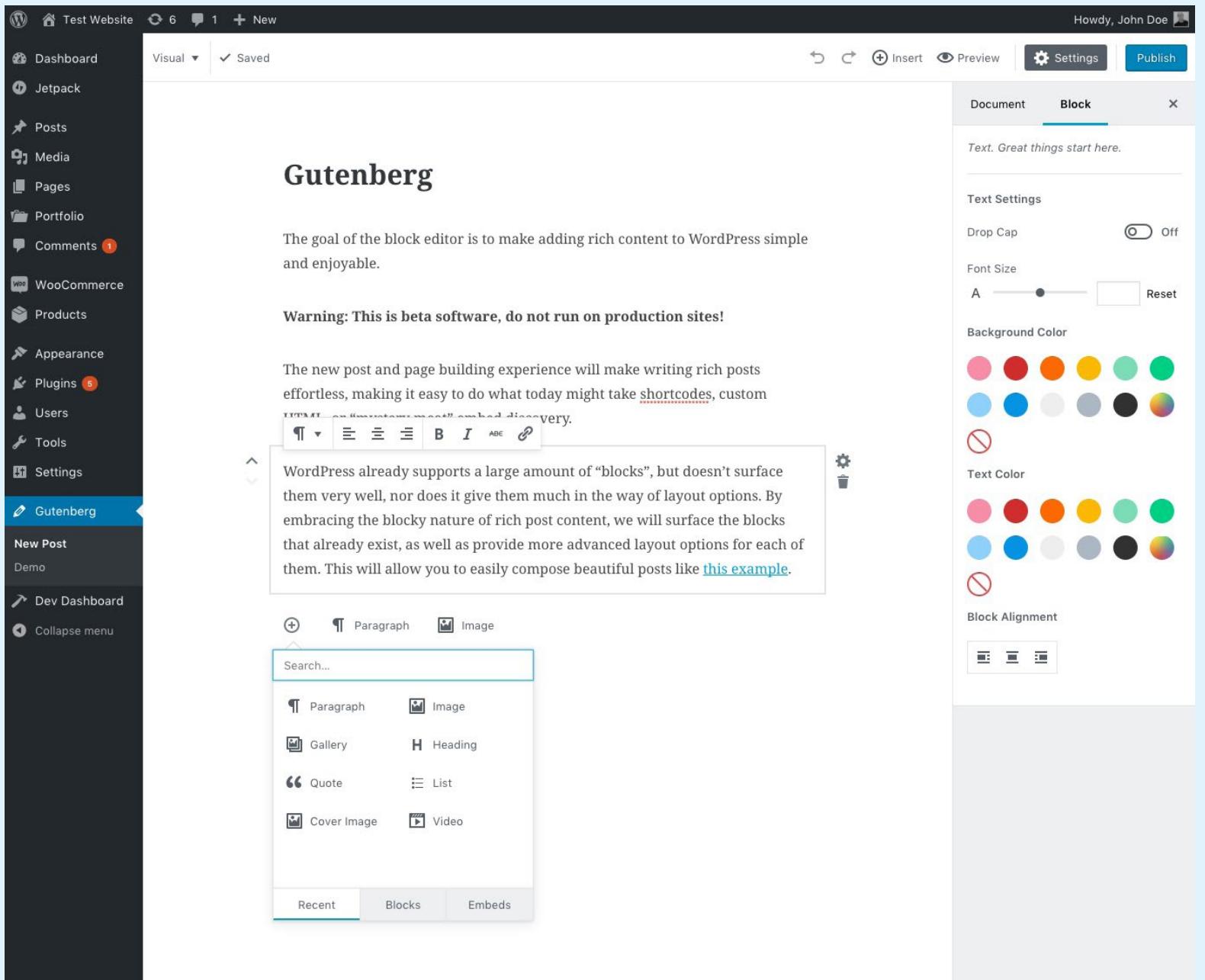


The screenshot shows the Squarespace website builder interface. On the left, a sidebar menu includes sections for 'Squeeze Pages', 'Case Studies', 'Squarespace Portfolio', 'Media Assets', and the current page, 'Squarespace Web Design', which is highlighted with a grey background. The main content area displays a page titled 'Squarespace Web Design'. The page content includes a heading 'Squarespace Web Design', a text block about Squarespace's popularity, a large image of the Squarespace logo, and a section titled 'Why Squarespace is so great' with a descriptive paragraph. The interface features a toolbar at the top with buttons for 'View Page', 'Settings', 'Cancel', and 'Save'.

Anthony Casalena and Others

# Wordpress

2005-Today



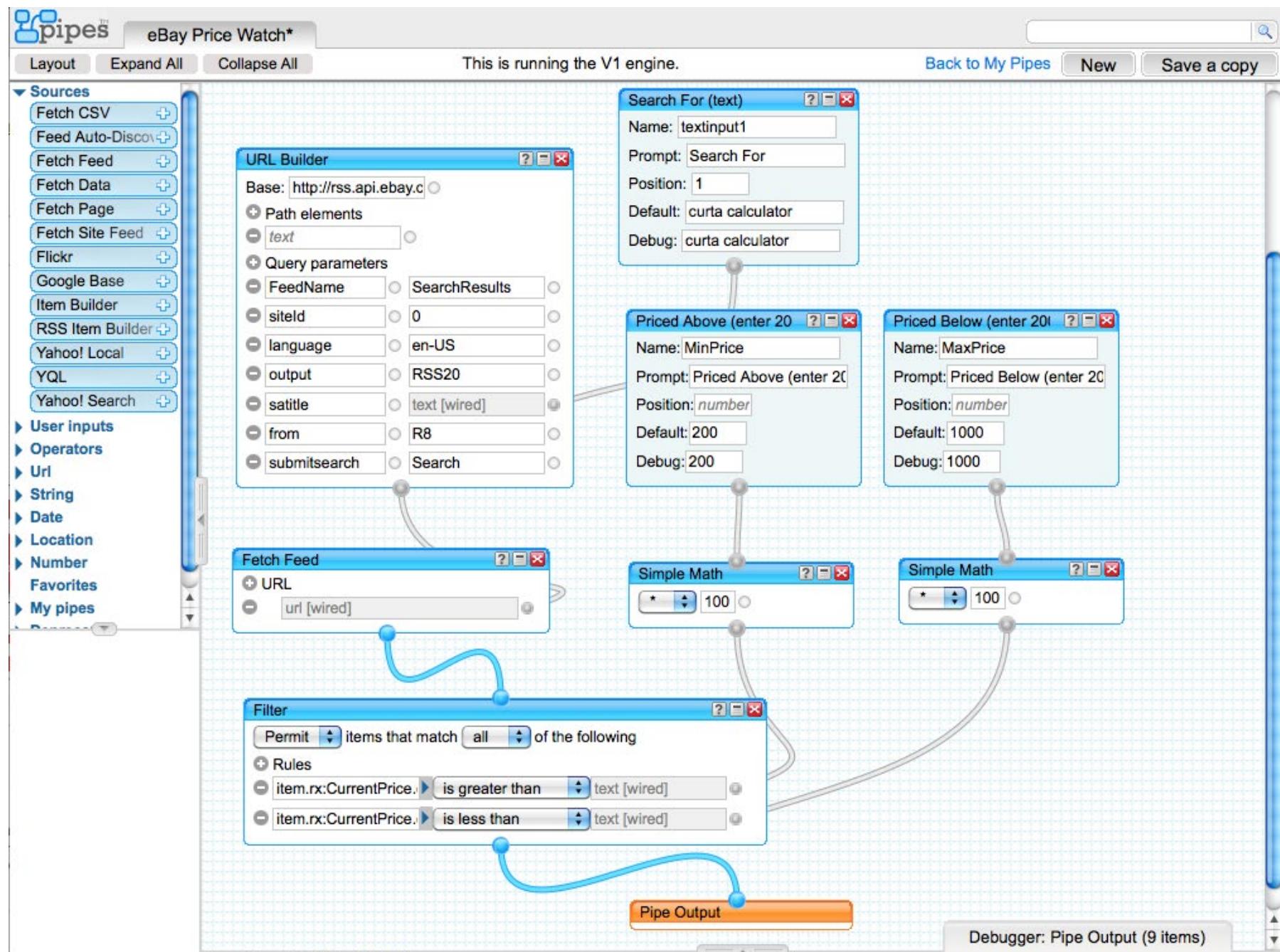
Automattic



Apple

# Yahoo! Pipes

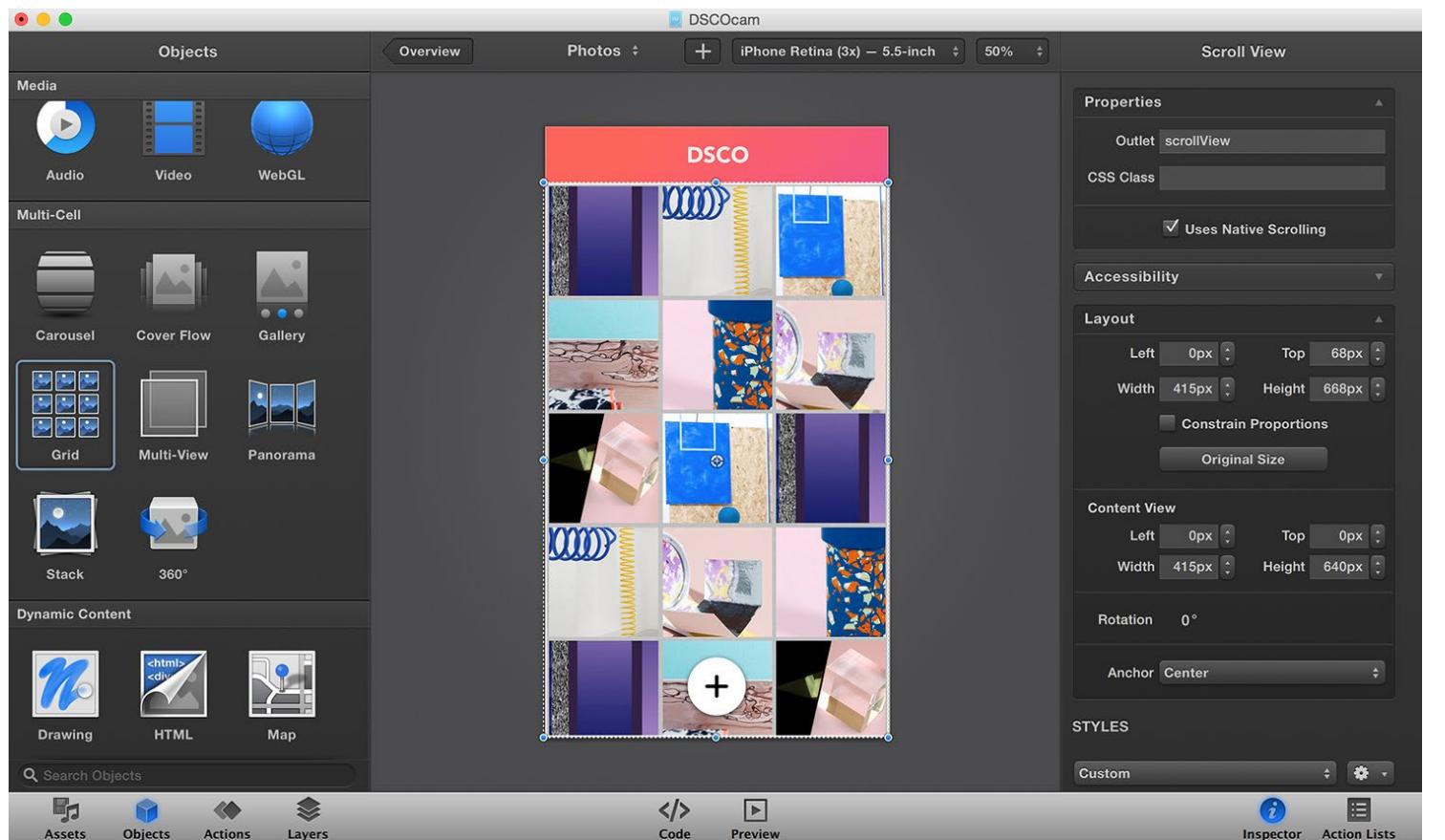
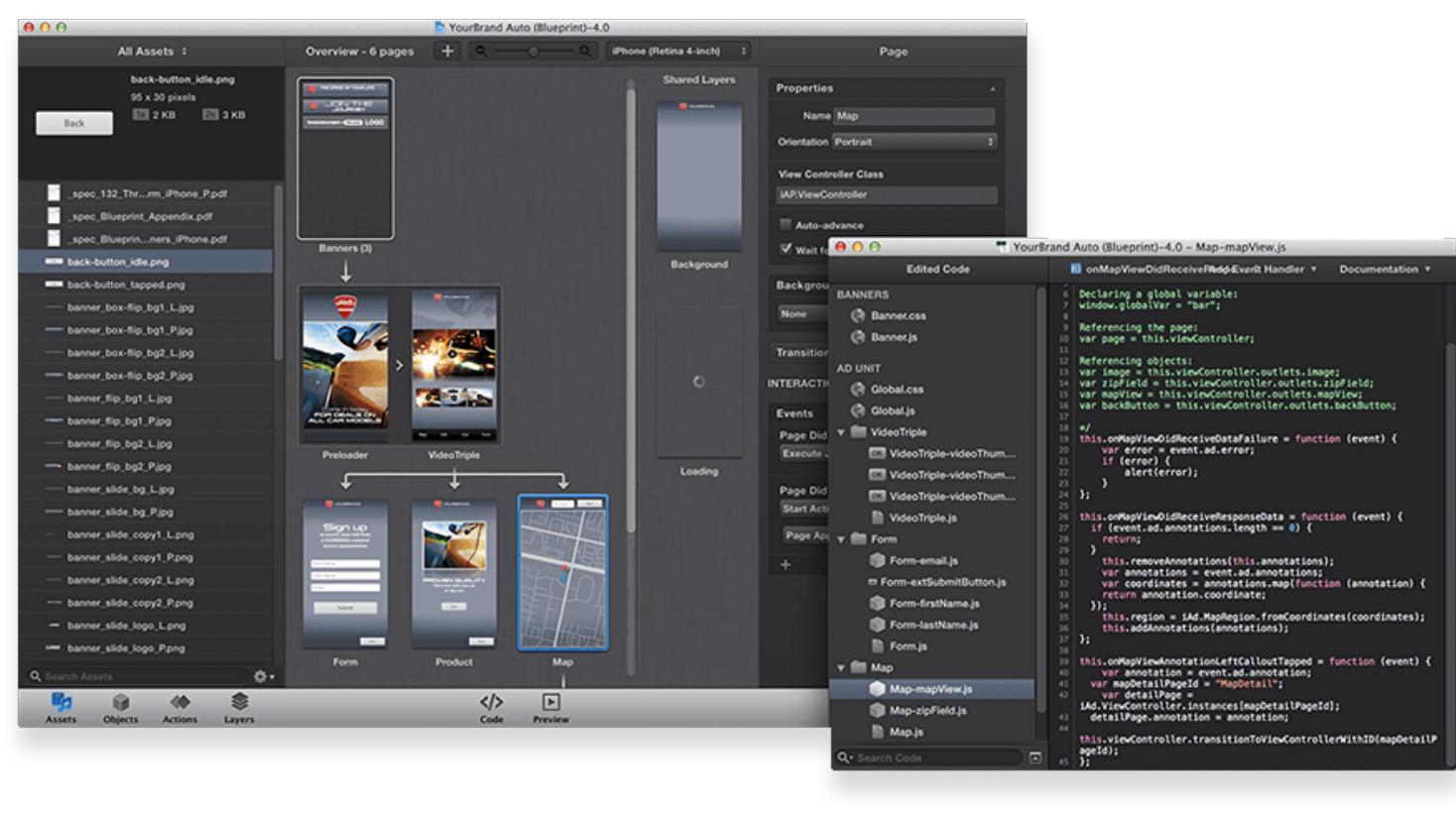
2007-2015



Pasha Sadri, Ed Ho, Jonathan Trevor, Kevin Cheng, Ido Green and Daniel Raffel at Yahoo!

# iAd Producer

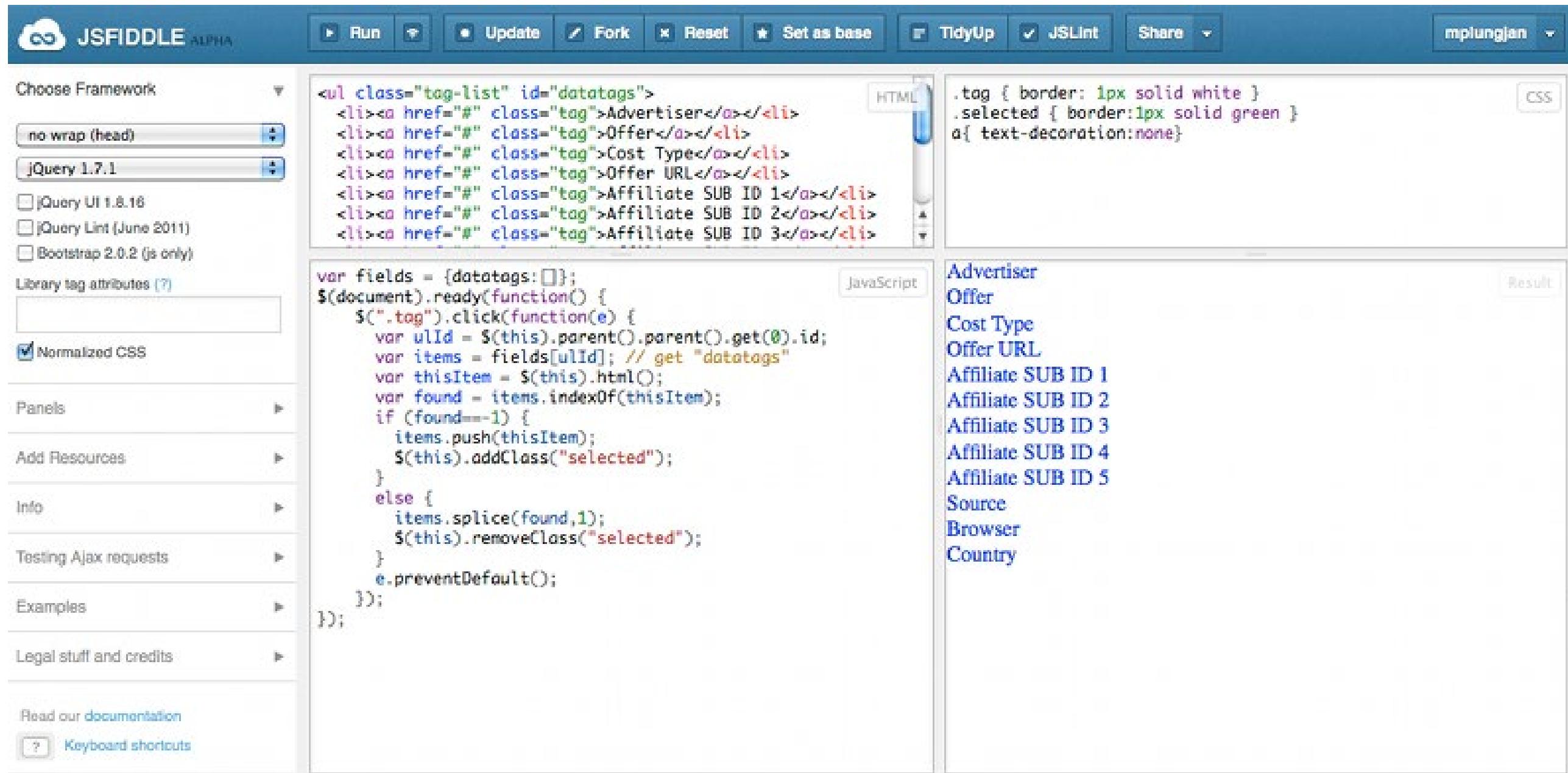
2010-2016



Apple

# JSFiddle

2010-Today



The screenshot shows the JSFiddle interface with the following components:

- Header:** JSFIDDLE ALPHA, Run, Update, Fork, Reset, Set as base, TidyUp, JSLint, Share, User: mplungjan
- Left Sidebar:** Choose Framework (no-wrap (head), jQuery 1.7.1, jQuery UI 1.8.16, jQuery Lint (June 2011), Bootstrap 2.0.2 (js only)), Library tag attributes, Normalized CSS, Panels, Add Resources, Info, Testing Ajax requests, Examples, Legal stuff and credits, Read our documentation, Keyboard shortcuts.
- HTML Panel:**

```
<ul class="tag-list" id="datatags">
  <li><a href="#" class="tag">Advertiser</a></li>
  <li><a href="#" class="tag">Offer</a></li>
  <li><a href="#" class="tag">Cost Type</a></li>
  <li><a href="#" class="tag">Offer URL</a></li>
  <li><a href="#" class="tag">Affiliate SUB ID 1</a></li>
  <li><a href="#" class="tag">Affiliate SUB ID 2</a></li>
  <li><a href="#" class="tag">Affiliate SUB ID 3</a></li>
```
- CSS Panel:**

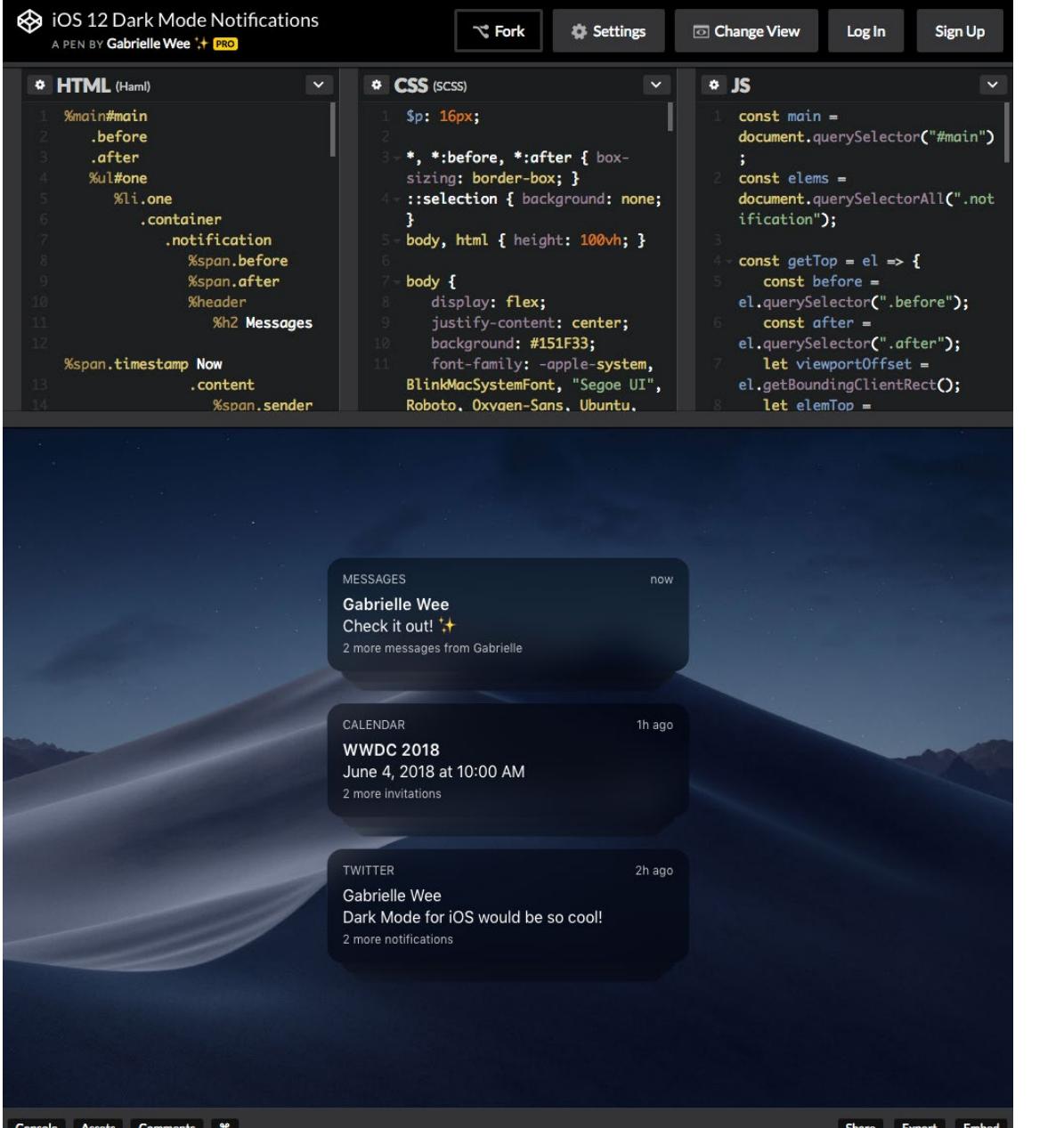
```
.tag { border: 1px solid white }
.selected { border: 1px solid green }
a { text-decoration: none}
```
- JavaScript Panel:**

```
var fields = {datatags:[]};
$(document).ready(function() {
  $(".tag").click(function(e) {
    var ulId = $(this).parent().parent().get(0).id;
    var items = fields[ulId]; // get "datatags"
    var thisItem = $(this).html();
    var found = items.indexOf(thisItem);
    if (found === -1) {
      items.push(thisItem);
      $(this).addClass("selected");
    }
    else {
      items.splice(found,1);
      $(this).removeClass("selected");
    }
    e.preventDefault();
  });
});
```
- Result Panel:** A list of items: Advertiser, Offer, Cost Type, Offer URL, Affiliate SUB ID 1, Affiliate SUB ID 2, Affiliate SUB ID 3, Affiliate SUB ID 4, Affiliate SUB ID 5, Source, Browser, Country. The first item, "Advertiser", is highlighted with a green border.

Oskar Krawczyk, Piotr Zalewa

# CodePen

2012-Today



The screenshot shows the CodePen interface with the following details:

- HTML (Ham)**

```
1 %main#main
2   .before
3   .after
4 %ul#one
5   %li.one
6     .container
7       .notification
8         %span.before
9         %span.after
10        %header
11        %h2 Messages
12
13 %span.timestamp Now
14 .content
15 %span.sender
```
- CSS (SCSS)**

```
1 $p: 16px;
2
3 *, *:before, *:after { box-sizing: border-box; }
4 ::selection { background: none; }
5
6 body, html { height: 100vh; }
7
8 body {
9   display: flex;
10  justify-content: center;
11  background: #151F33;
12  font-family: -apple-system,
13  BlinkMacSystemFont, "Segoe UI",
14  Roboto, Oxyvaen-Sans, Ubuntu,
```
- JS**

```
1 const main =
2   document.querySelector("#main")
3
4 const elems =
5   document.querySelectorAll(".notification");
6
7 const getTop = el => {
8   const before =
9     el.querySelector(".before");
10  const after =
11    el.querySelector(".after");
12  let viewportOffset =
13    el.getBoundingClientRect();
14  let elemTop =
```

**Preview:**

- MESSAGES**  
Gabrielle Wee  
Check it out! 🎉  
2 more messages from Gabrielle
- CALENDAR**  
WWDC 2018  
June 4, 2018 at 10:00 AM  
2 more invitations
- TWITTER**  
Gabrielle Wee  
Dark Mode for iOS would be so cool!  
2 more notifications

At the bottom, there are buttons for Console, Assets, Comments, Share, Export, and Embed.

Tim Sabat, Alex Vazquez, Chris Coyier

# Webflow

2013-Today

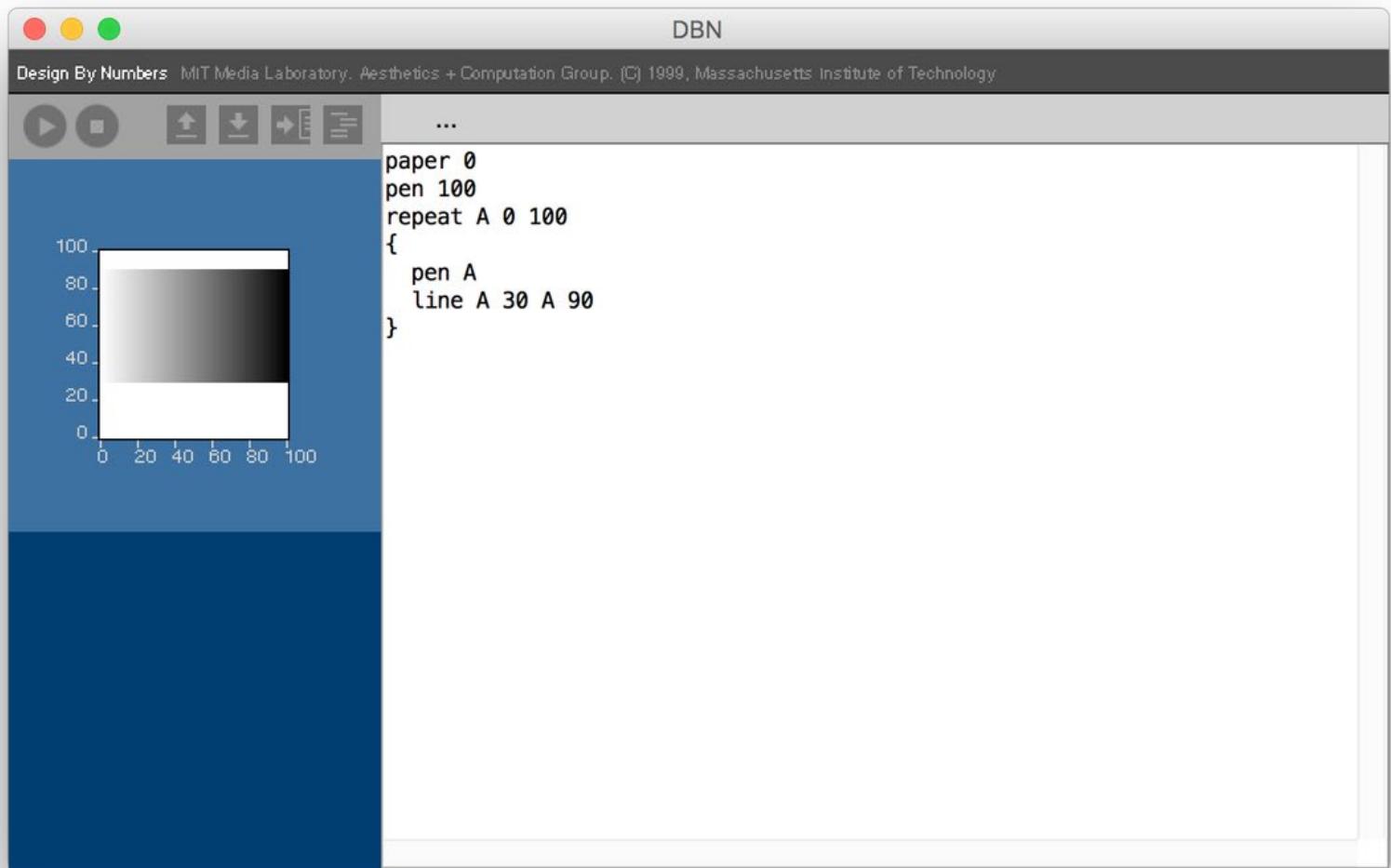
The screenshot shows the Webflow interface. On the left, a preview window displays a 'latest reviews' section with five cards. The cards contain album covers, titles, artists, ratings, and short descriptions. The bottom card has a detailed description and a 'Song Reviews' dropdown menu open, showing options like 'Get text from: Song Reviews', 'Select field...', 'Name (text)', 'Artist (text)', 'Review Summary (text)', 'Soundcloud Embed (text)', 'Author (text)', 'Post Date (date)', 'Star Rating Image (reference)', 'Name (text)', and 'Created On (date)'. On the right, the content editor shows a 'Paragraph' component with the selector 'post summary'. The properties panel shows settings for width, height, margin, float, clear, overflow, position, typography, and background. The bottom navigation bar shows the path: 'center center wrap' > 'Dynamic List Wrapper' > 'post list wrap' > 'post wrap' > 'post link' > 'post summary wrap' > 'post summary'.

Vlad Magdalin and Sergie Magdalin

# **'Creative Coding' Toolkits**

# Design by Numbers

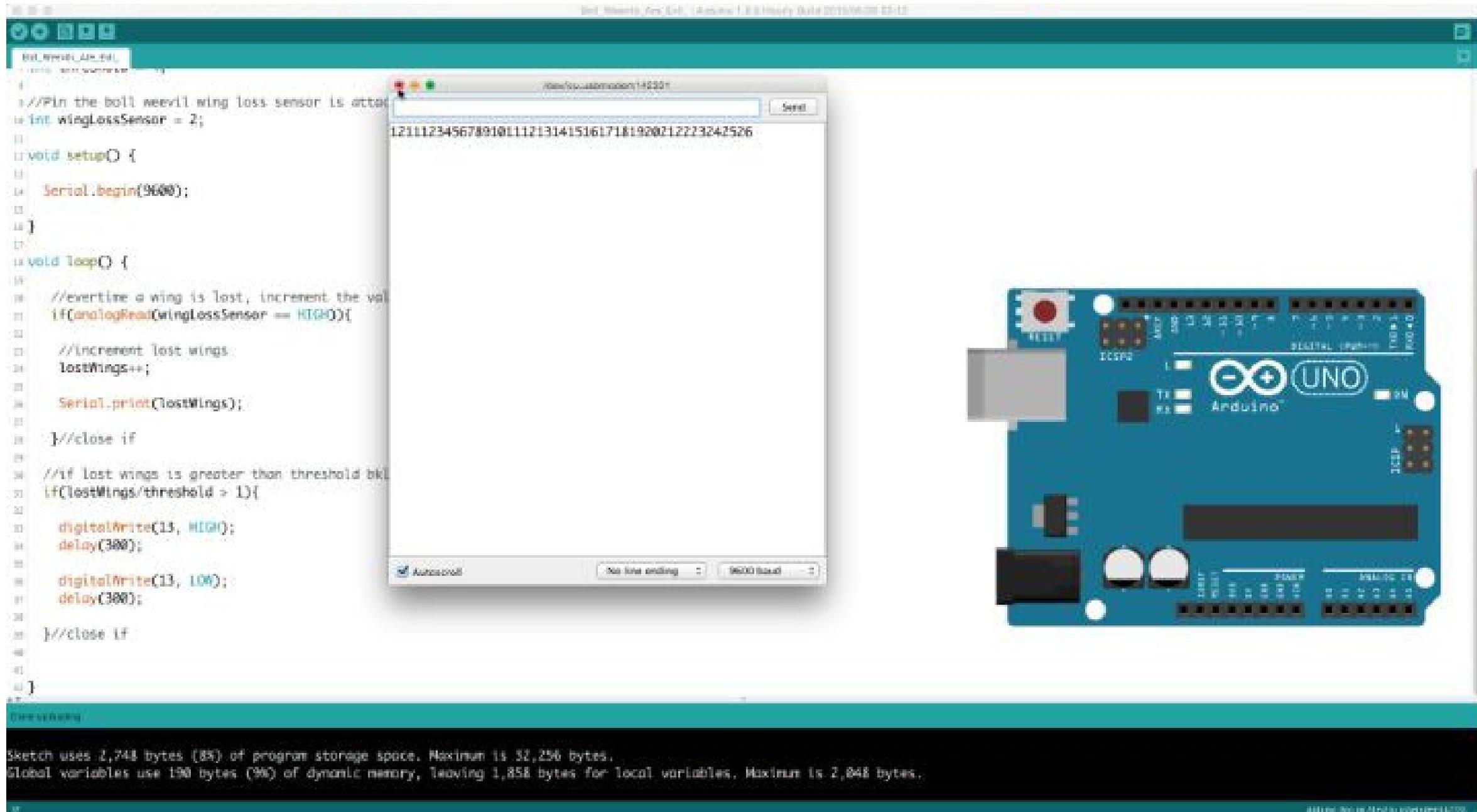
1999-2003



John Maeda at the MIT Media Lab (Aesthetics and Computation Group)

# Arduino

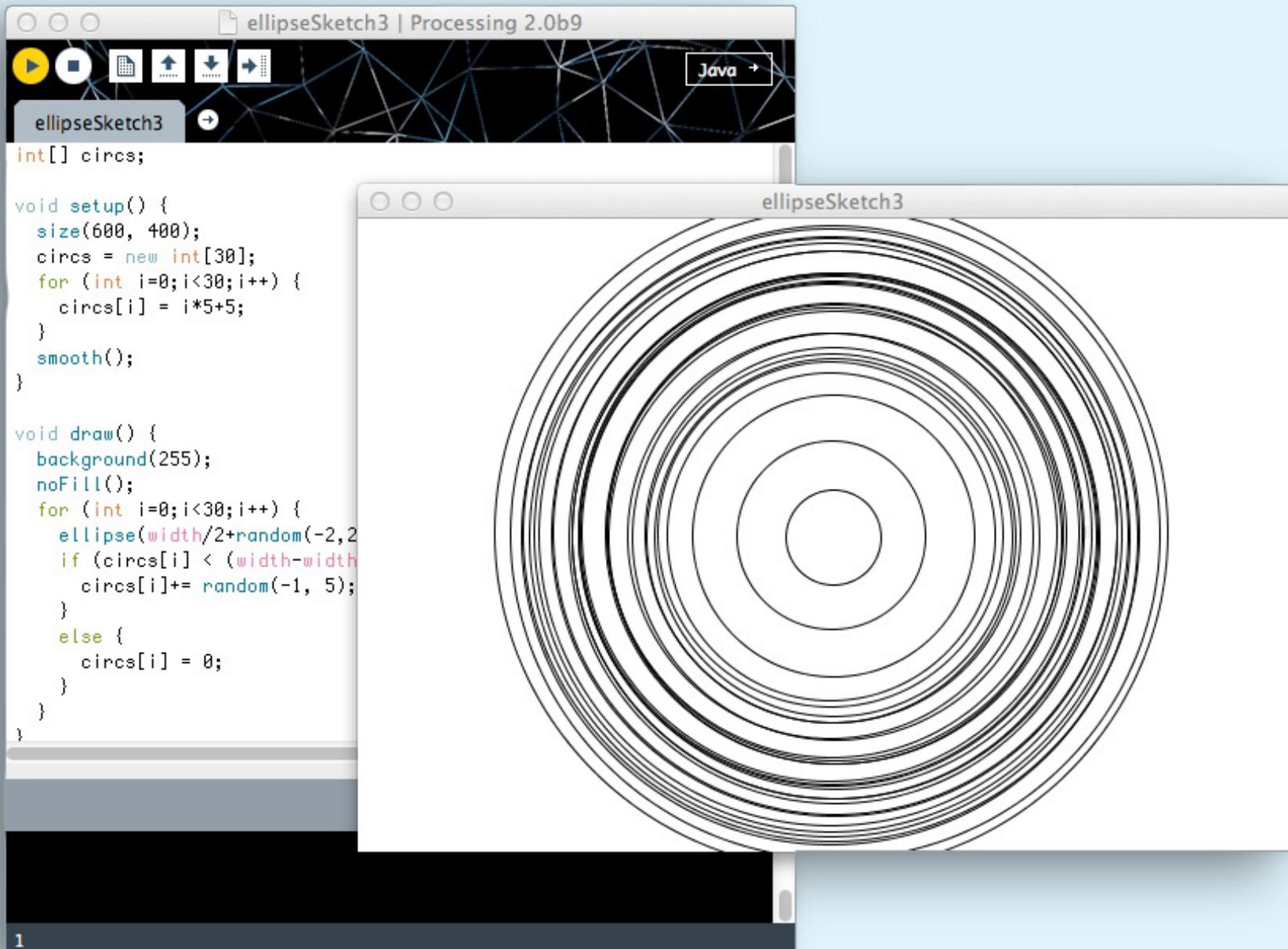
2003-Today



Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino, and David Mellis at Interaction Design Institute Ivrea

# Processing

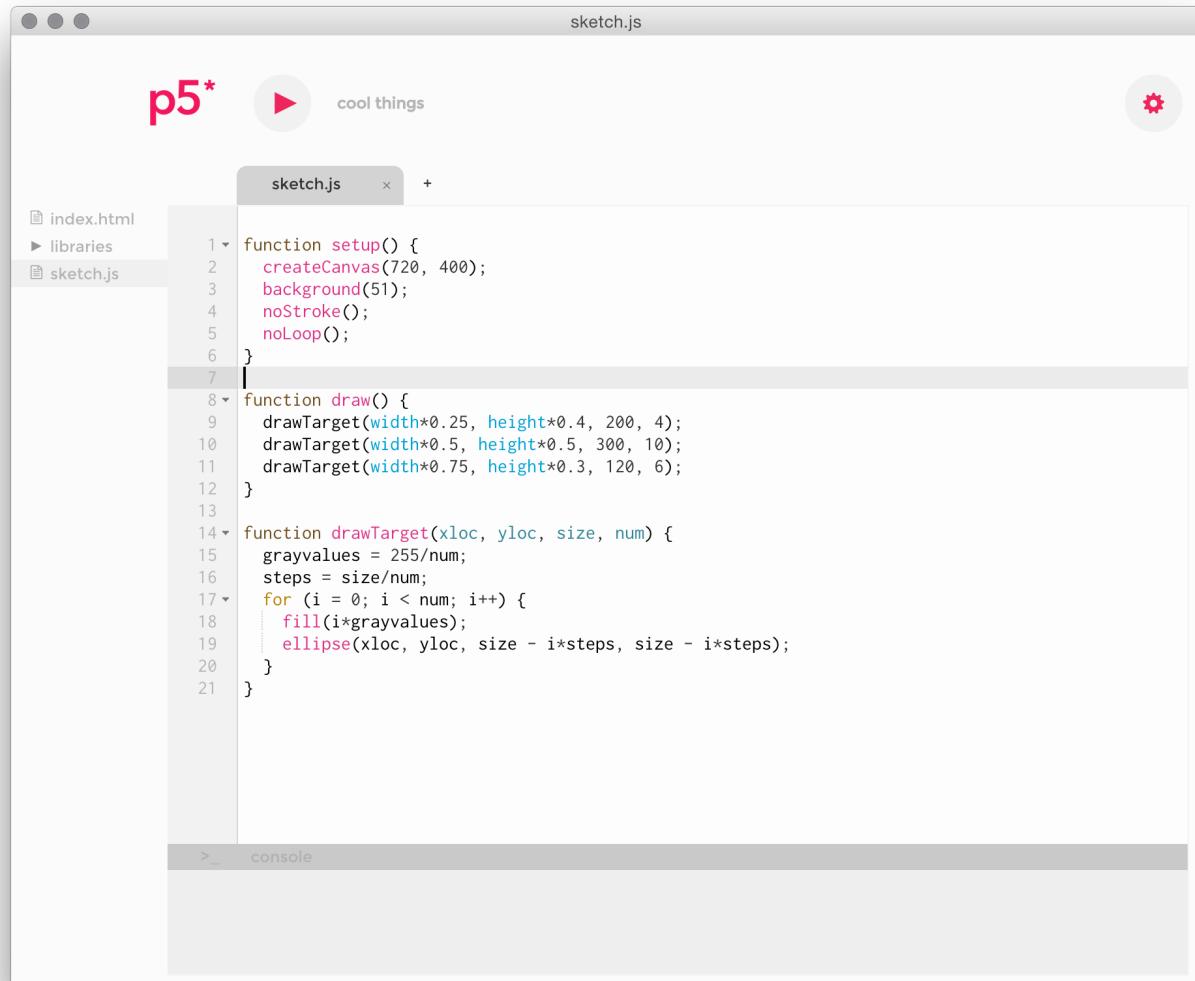
2001-Today



Casey Reas and Ben Fry at the MIT Media Lab (Now Processing Foundation)

# p5.js

2014-Today

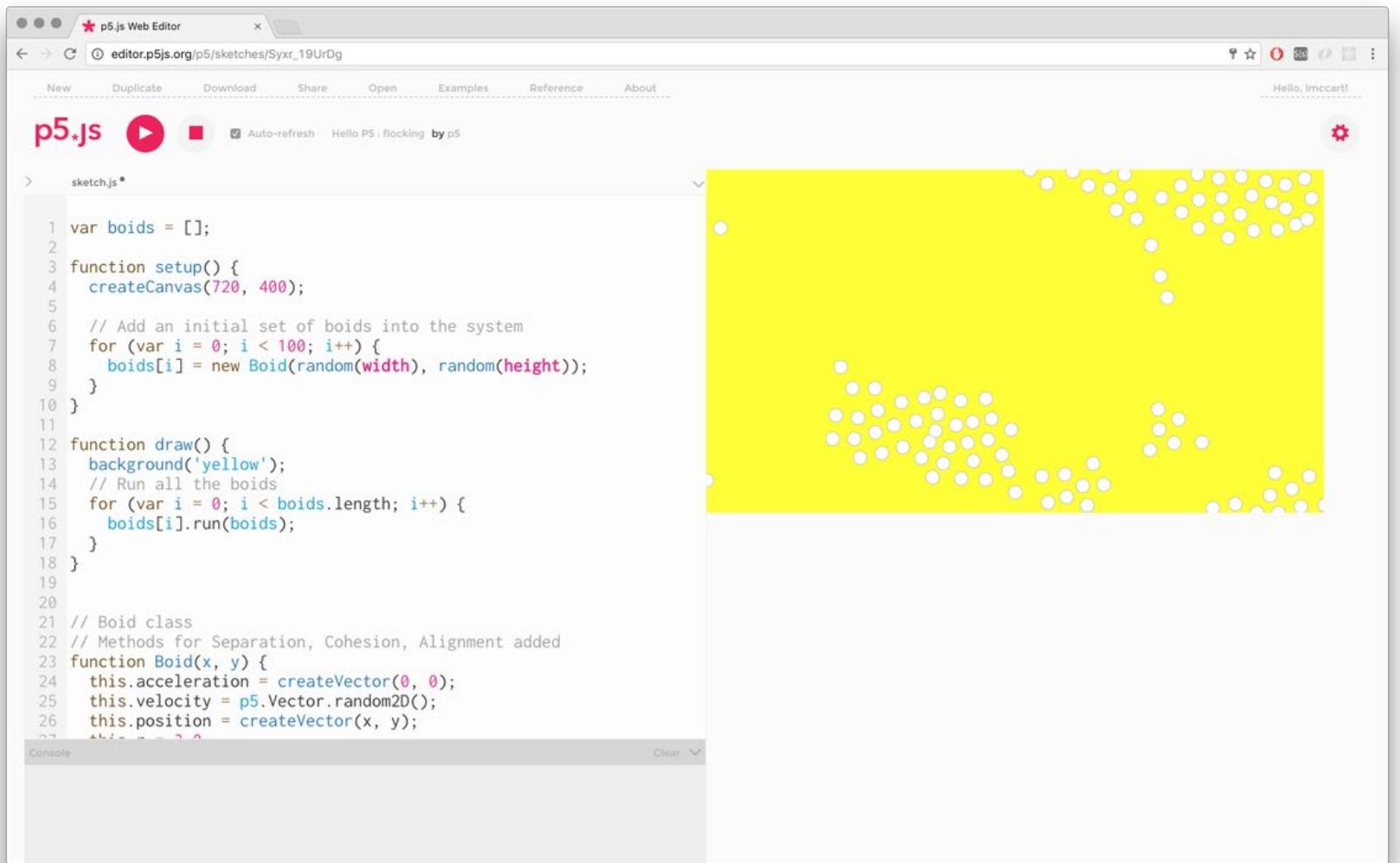


The screenshot shows the p5.js desktop application interface. The main window title is "sketch.js". The code editor contains the following JavaScript code:

```
function setup() {
  createCanvas(720, 400);
  background(51);
  noStroke();
  noLoop();
}

function draw() {
  drawTarget(width*0.25, height*0.4, 200, 4);
  drawTarget(width*0.5, height*0.5, 300, 10);
  drawTarget(width*0.75, height*0.3, 120, 6);
}

function drawTarget(xloc, yloc, size, num) {
  grayvalues = 255/num;
  steps = size/num;
  for (i = 0; i < num; i++) {
    fill(i*grayvalues);
    ellipse(xloc, yloc, size - i*steps, size - i*steps);
  }
}
```



The screenshot shows the p5.js Web Editor interface. The main window title is "sketch.js". The code editor contains the following JavaScript code:

```
var boids = [];

function setup() {
  createCanvas(720, 400);
  // Add an initial set of boids into the system
  for (var i = 0; i < 100; i++) {
    boids[i] = new Boid(random(width), random(height));
  }
}

function draw() {
  background('yellow');
  // Run all the boids
  for (var i = 0; i < boids.length; i++) {
    boids[i].run(boids);
  }
}

// Boid class
// Methods for Separation, Cohesion, Alignment added
function Boid(x, y) {
  this.acceleration = createVector(0, 0);
  this.velocity = p5.Vector.random2D();
  this.position = createVector(x, y);
  this.r = 20;
}
```

The preview window on the right shows a yellow canvas with a flock of small white dots representing boids, demonstrating a basic flocking algorithm.

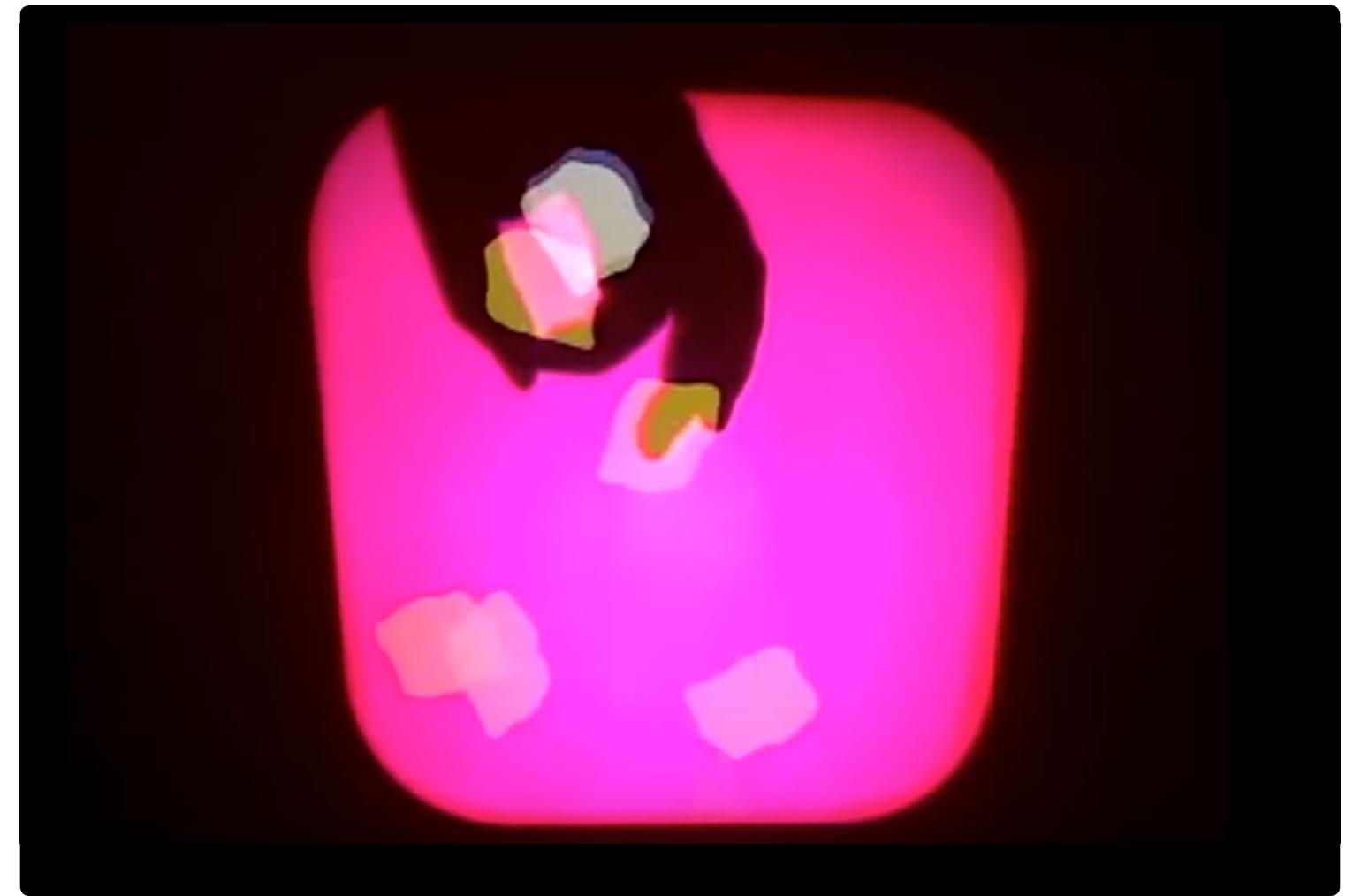
Lauren McCarthy at MIT (Now at UCLA, Dept. of Design Media Arts)

# Digital Drawing

## Painterly Approaches + Computational Methods

# The Manual Input Workstation

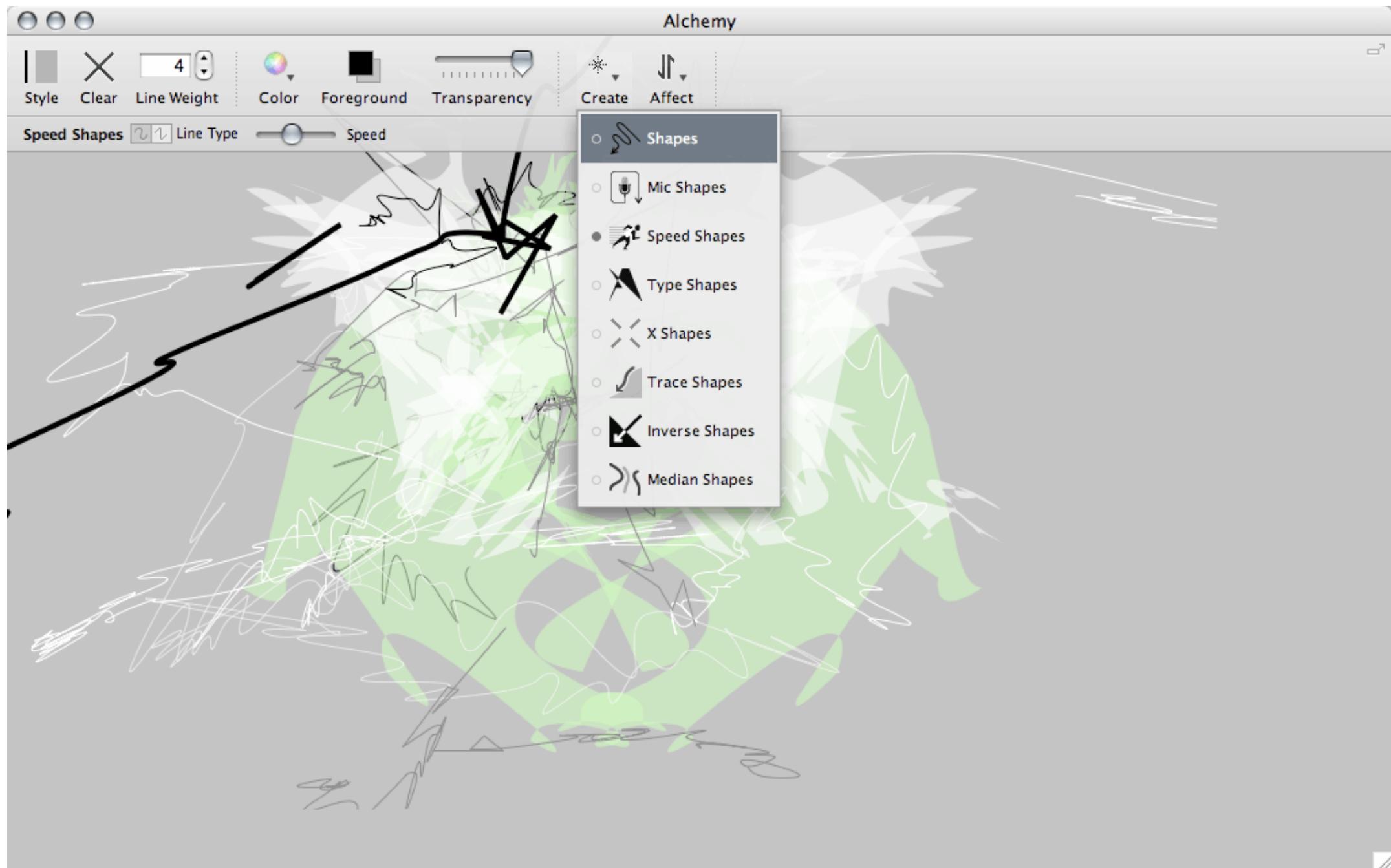
2004



Golan Levin and Zachary Lieberman

# Alchemy

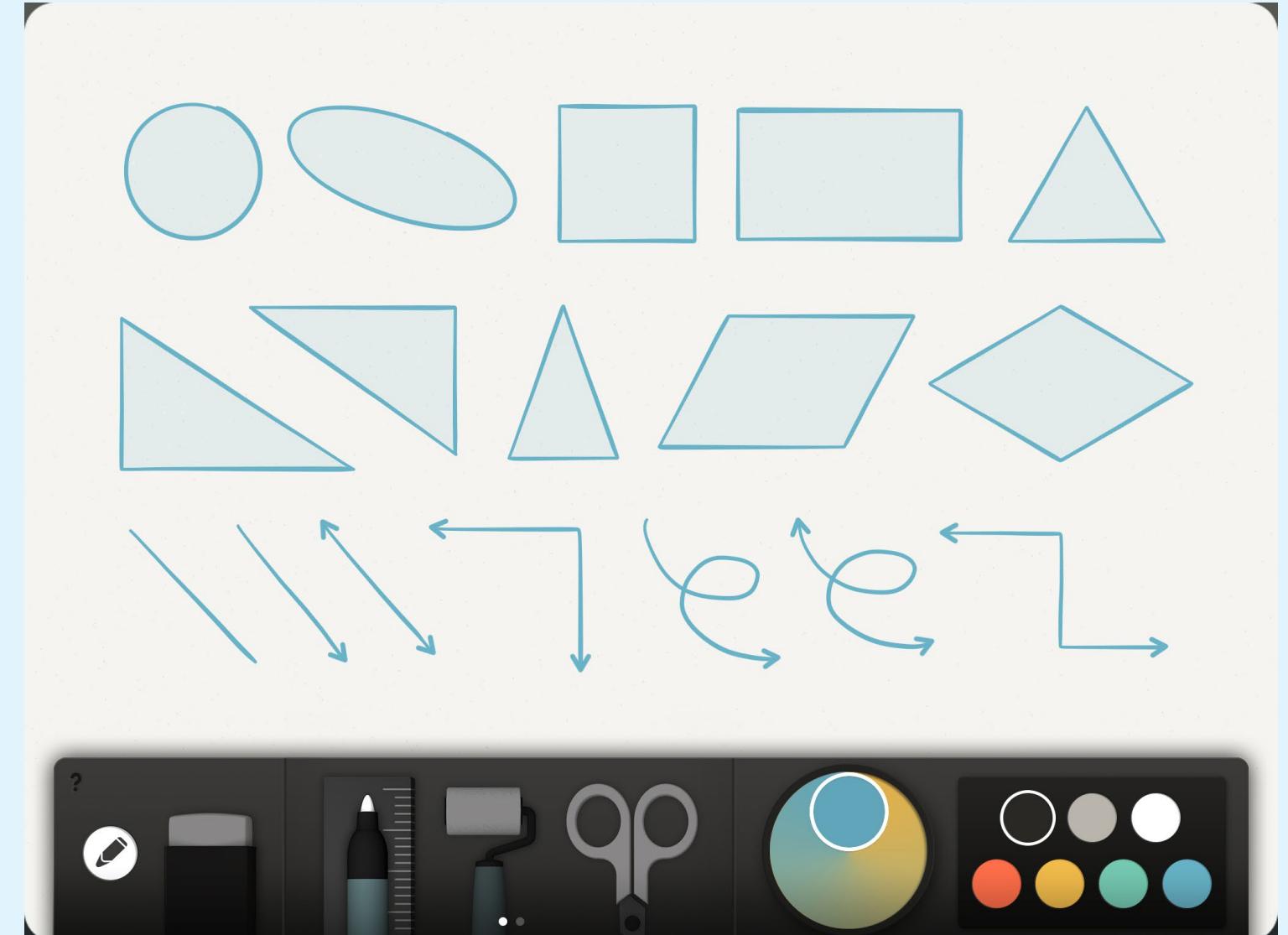
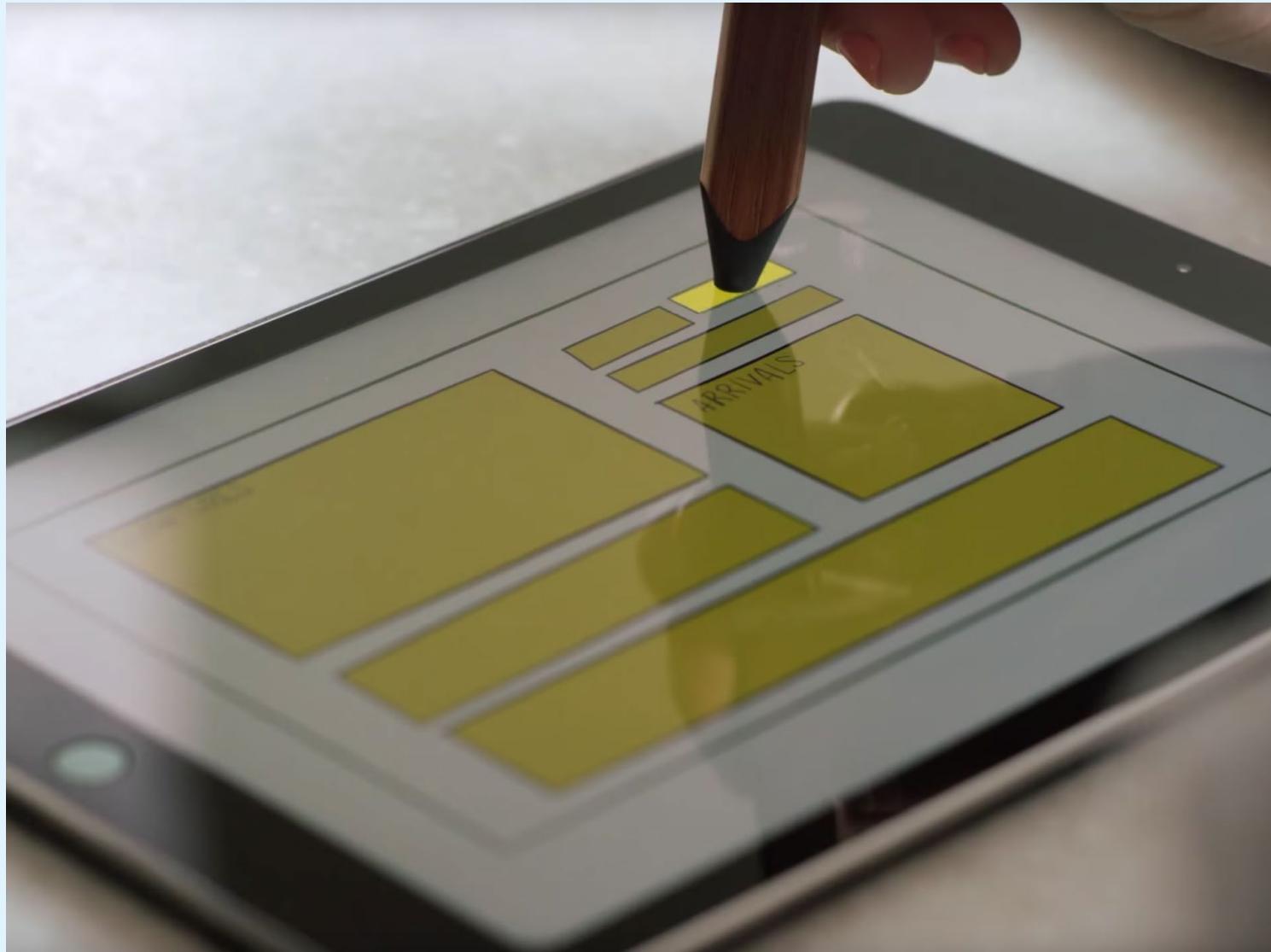
2008-



Karl D.D. Willis and Jacob Hina

# Paper

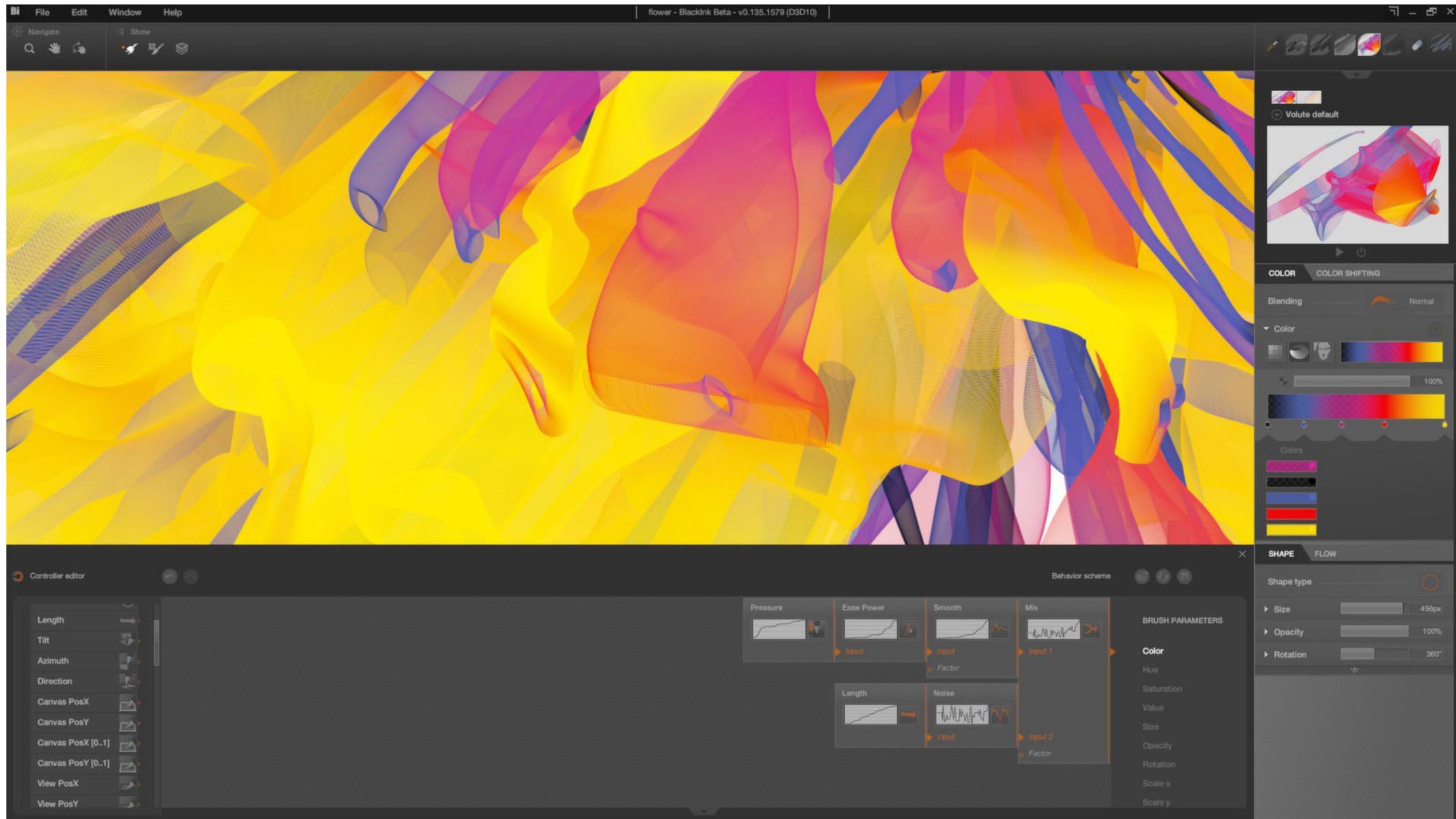
2011-Today



Georg Petschnigg, Andrew S. Allen, Julian Walker and Jonathan Harris for FiftyThree

# BlackInk

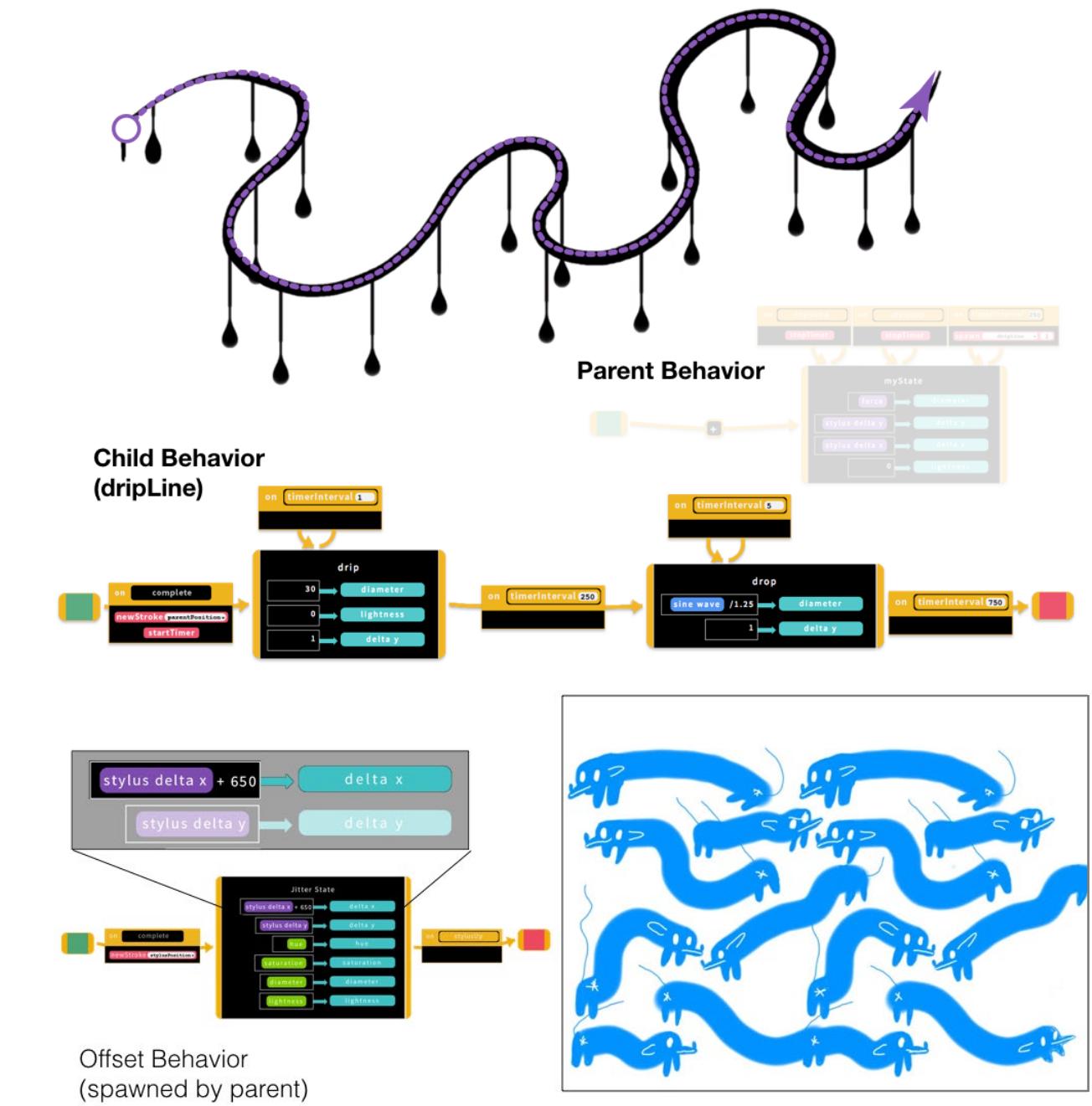
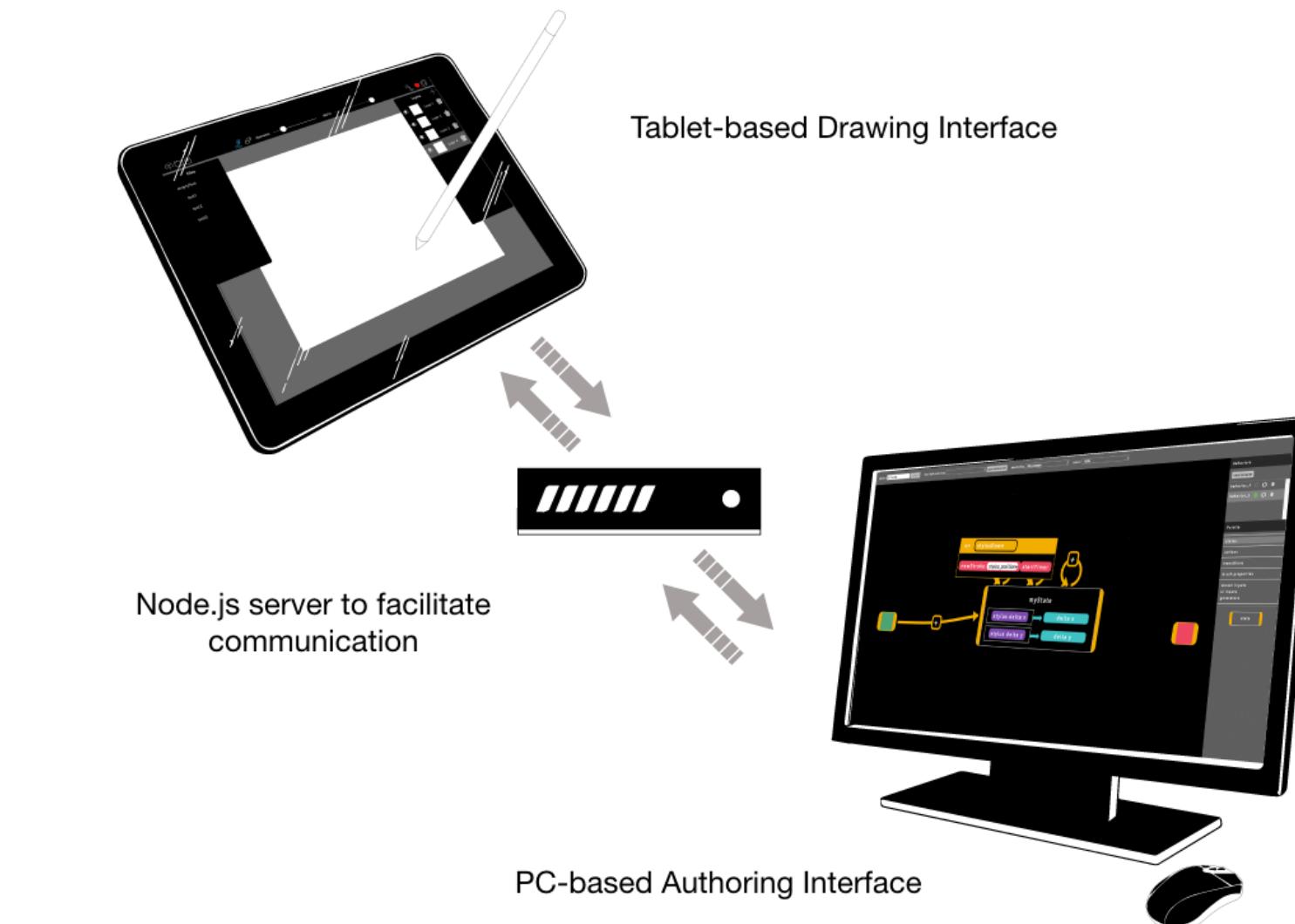
2013-Today



Bleank

# Dynamic Brushes

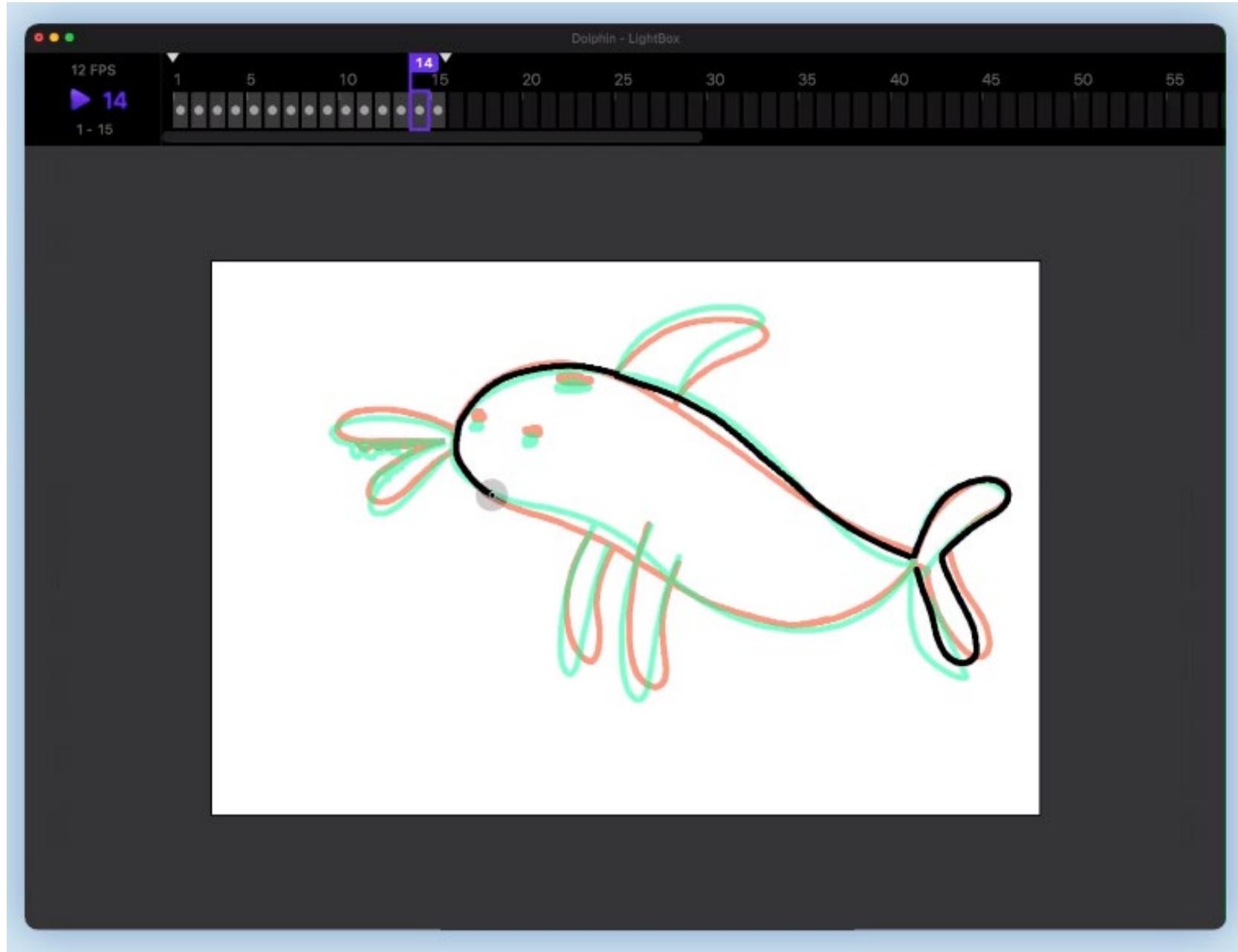
2017



Jennifer Jacobs

# Lightbox

2018-Today



Pasquale D'Silva, Jacob Bijani and Wojtek Witkowski

# Visual Interfaces

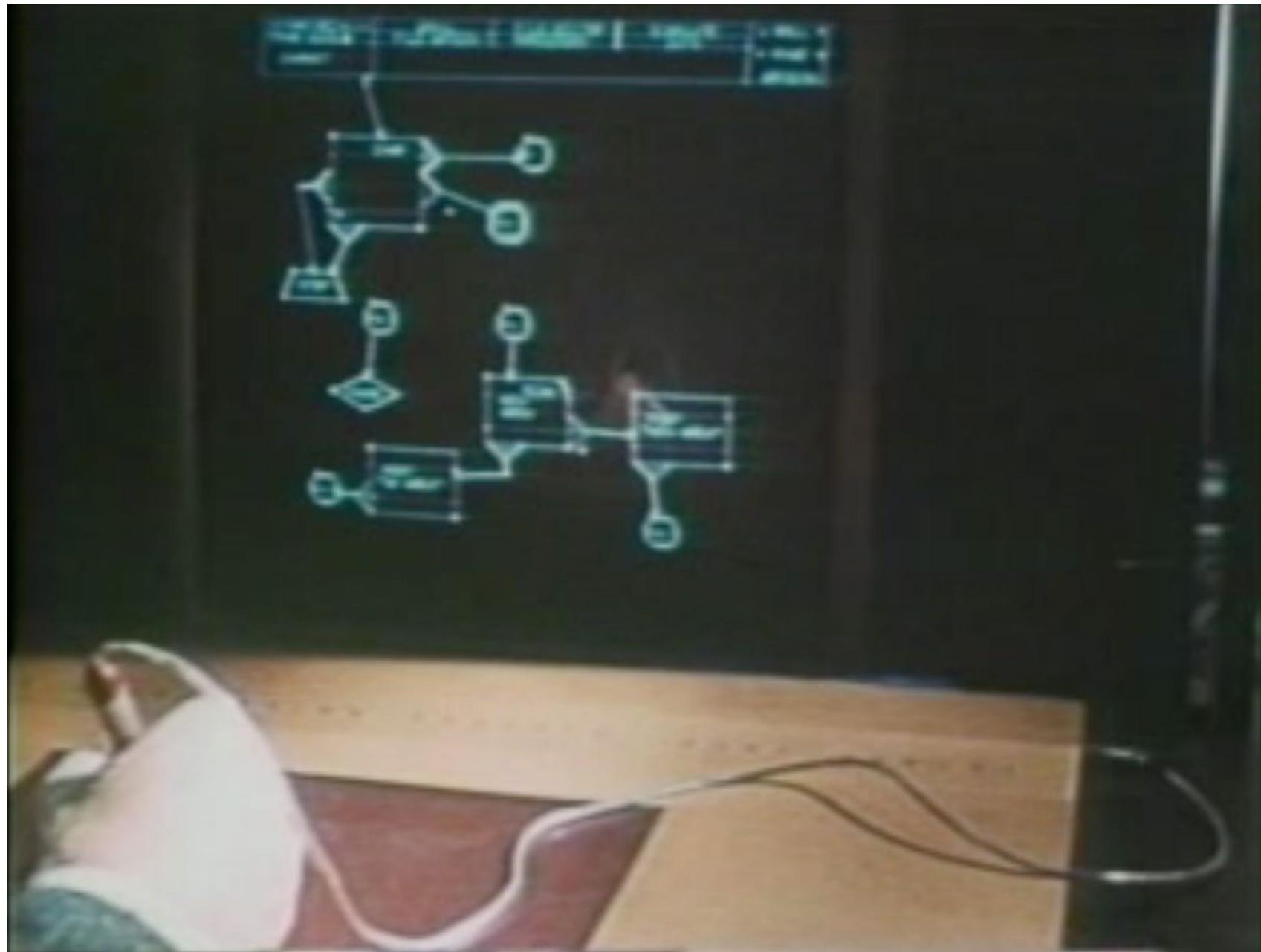
## Across the (Design-Development) Divide

# Node-Based (Graph)

Environments afford the  
Visual Authoring of Programs

# GRaIL

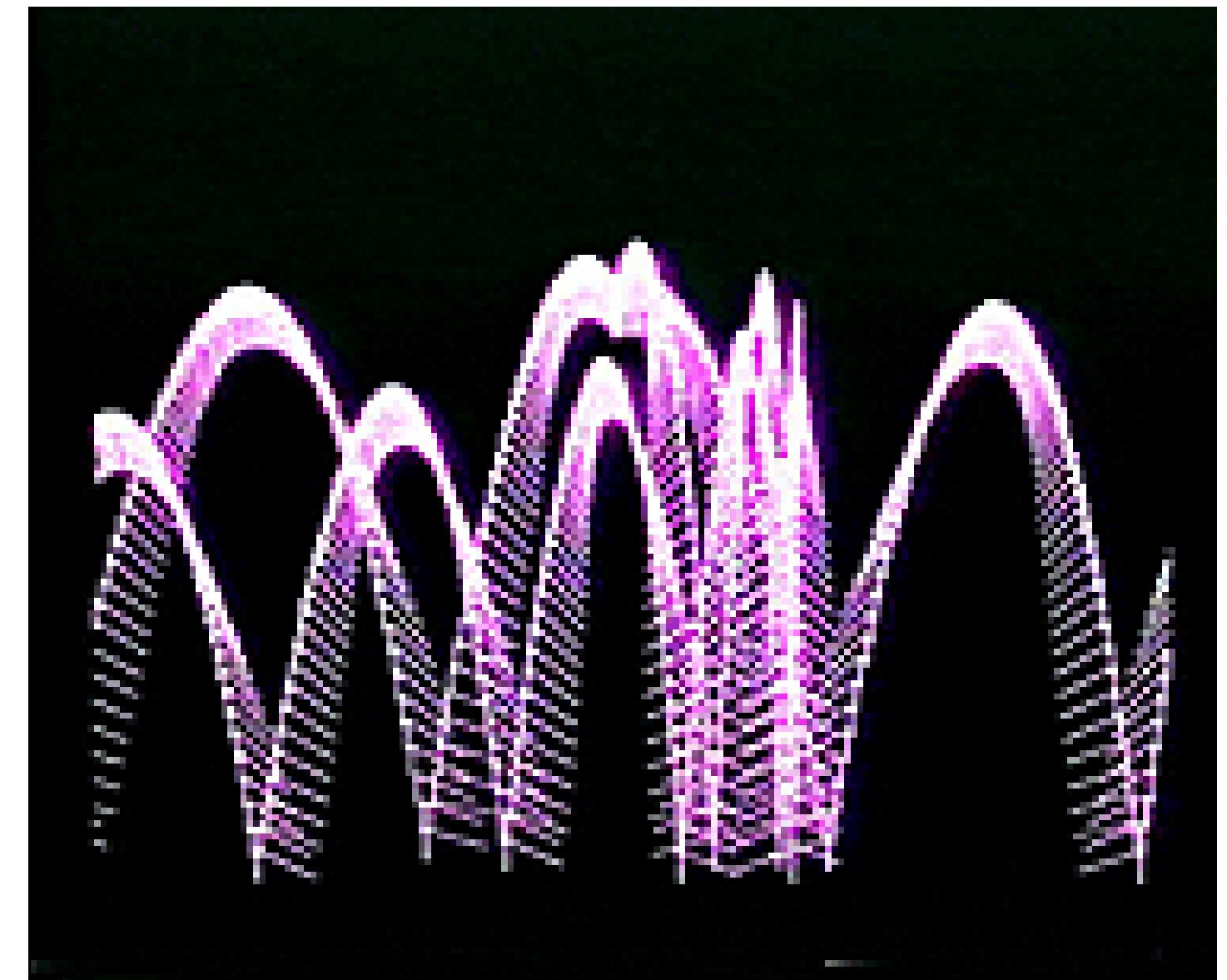
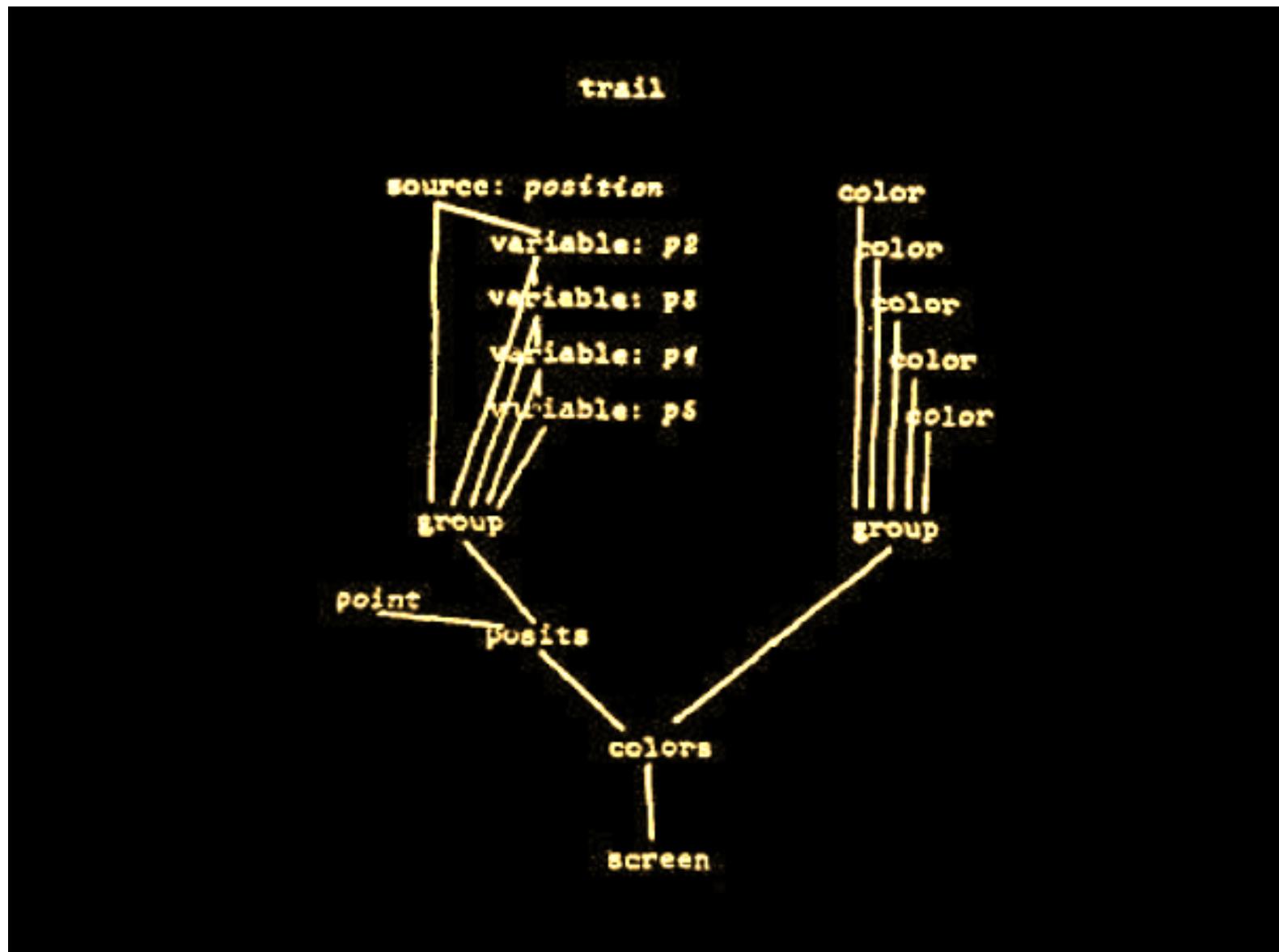
1968



RAND Corporation

# EOM (Graphical Simulation System)

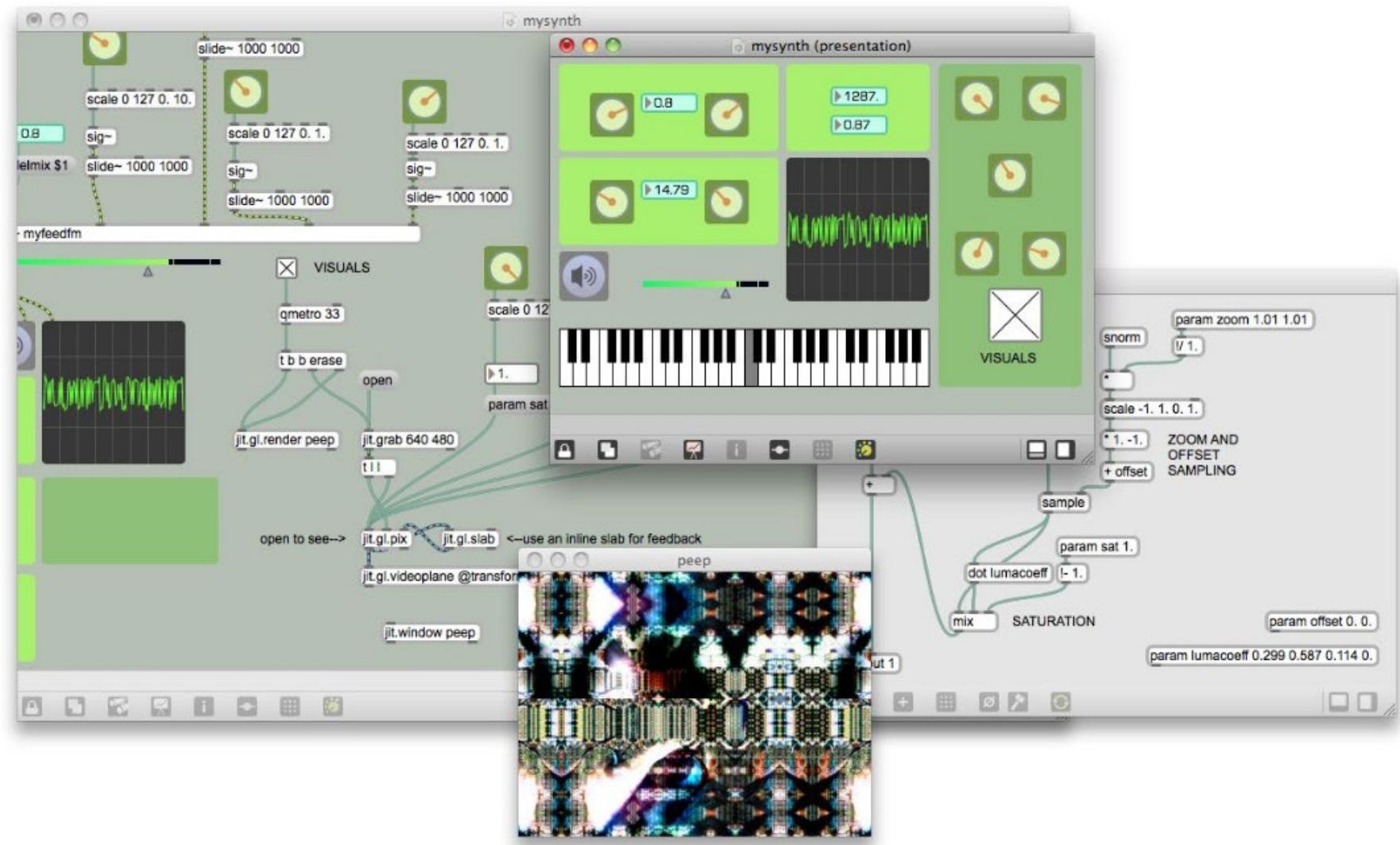
1976-1977



Paul Pangaro, Seth Steinberg, Jim Davis, and Ben McCann at the MIT Architecture Machine Group

# MaxMSP

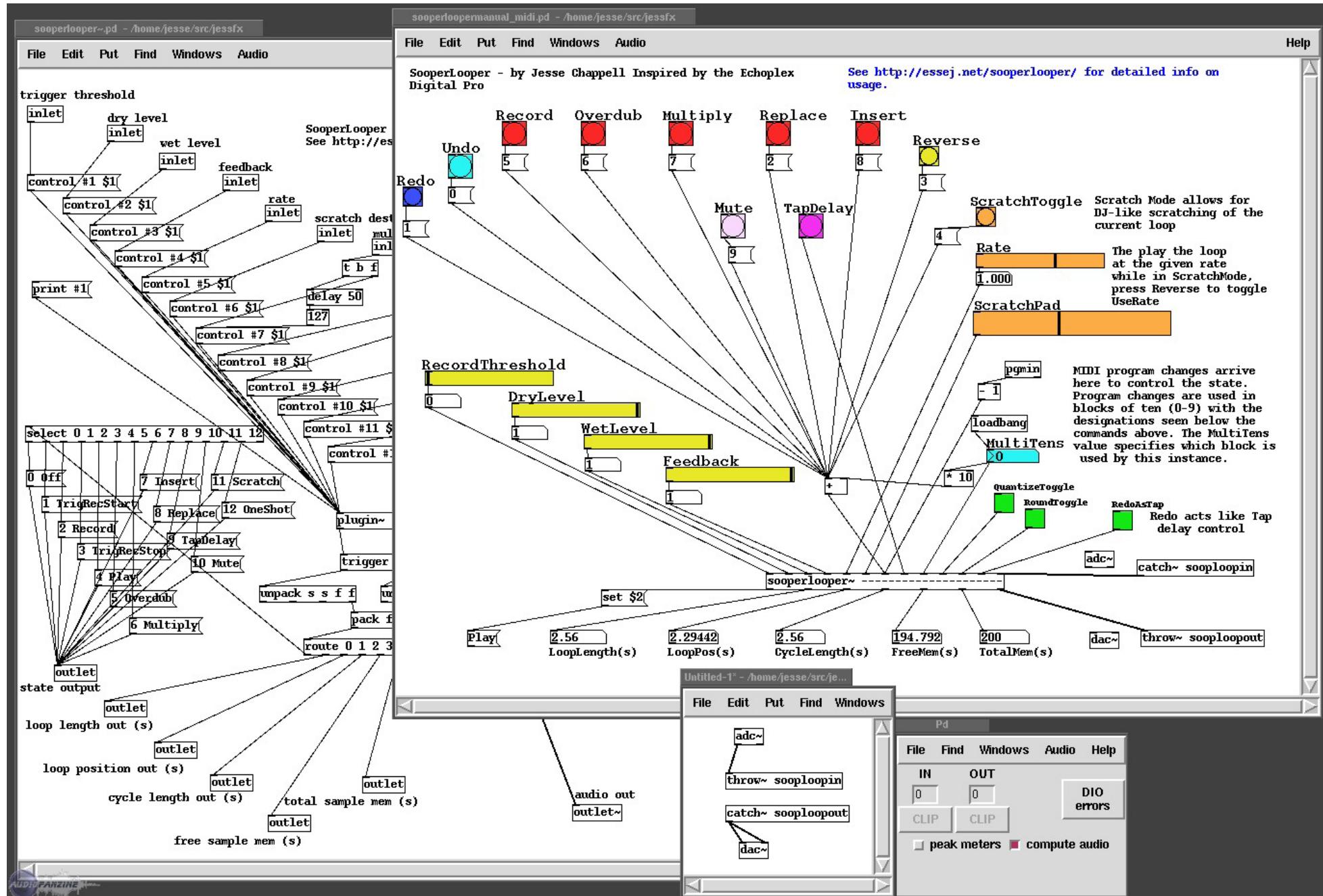
1990-Today



Miller Puckette at IRCAM (Now Cycling '74)

# Pure Data

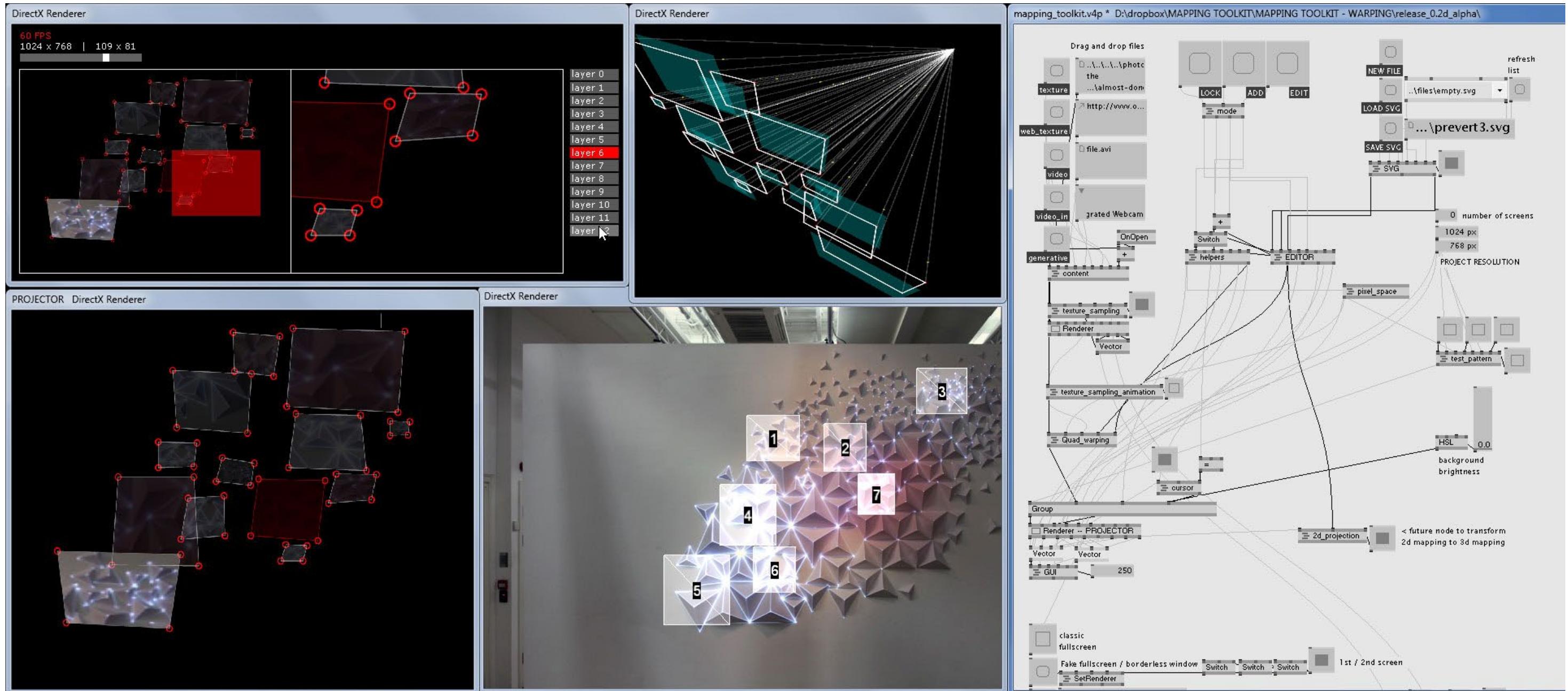
1996-Today



Miller Puckette

# vvvv

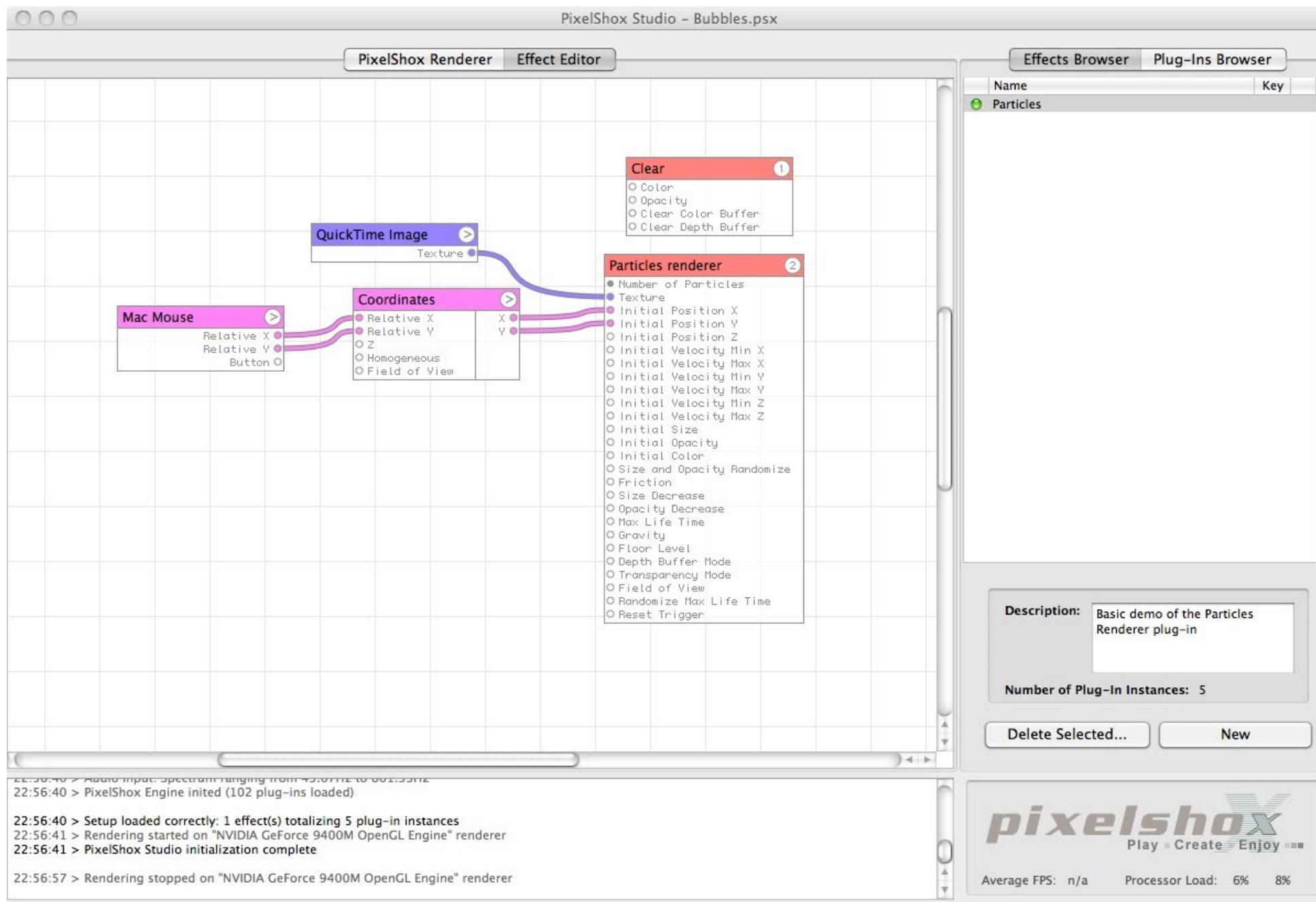
# 1998-Today



Joreg, Max Wolf, Sebastian Gregor, Sebastian Oschatz for 'vvvv Group'

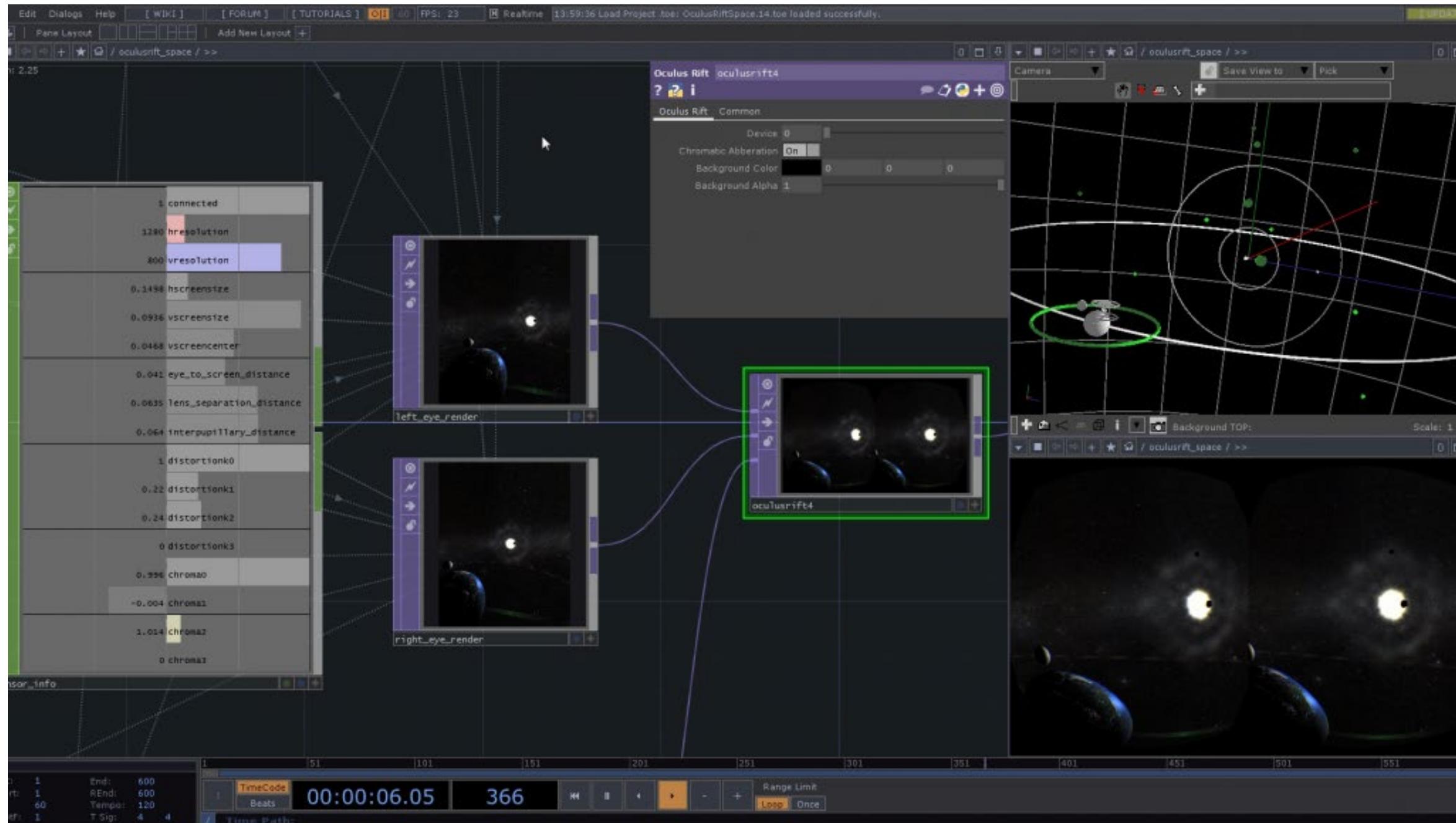
# PixelShox Studio

2002-2003



Pierre-Olivier Latour for PixelShox (Acquired by Apple)

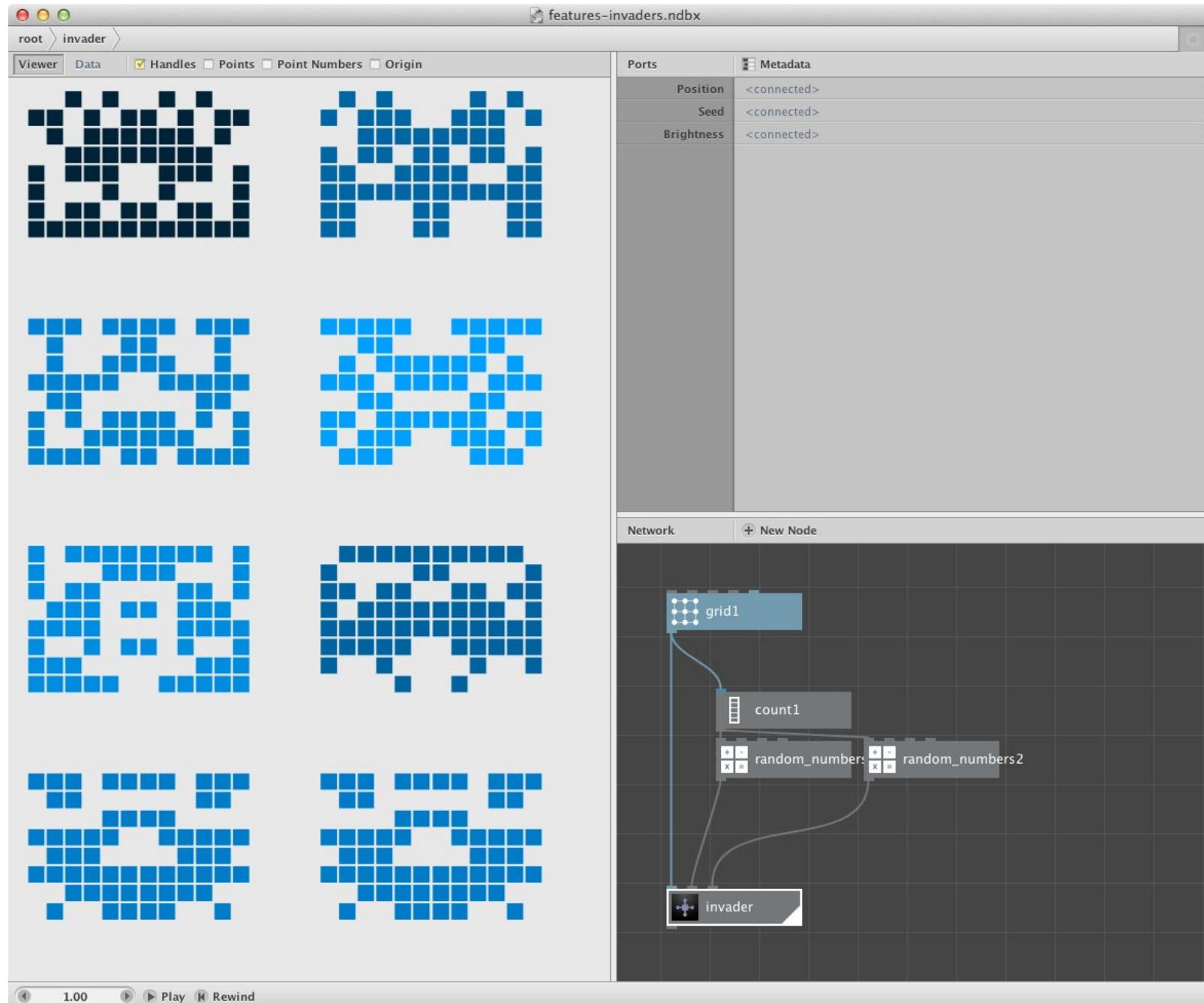
# TouchDesigner 2000-Today



Greg Hermanovic, Rob Bairos, and Jarrett Smith for Derivative Inc

# NodeBox

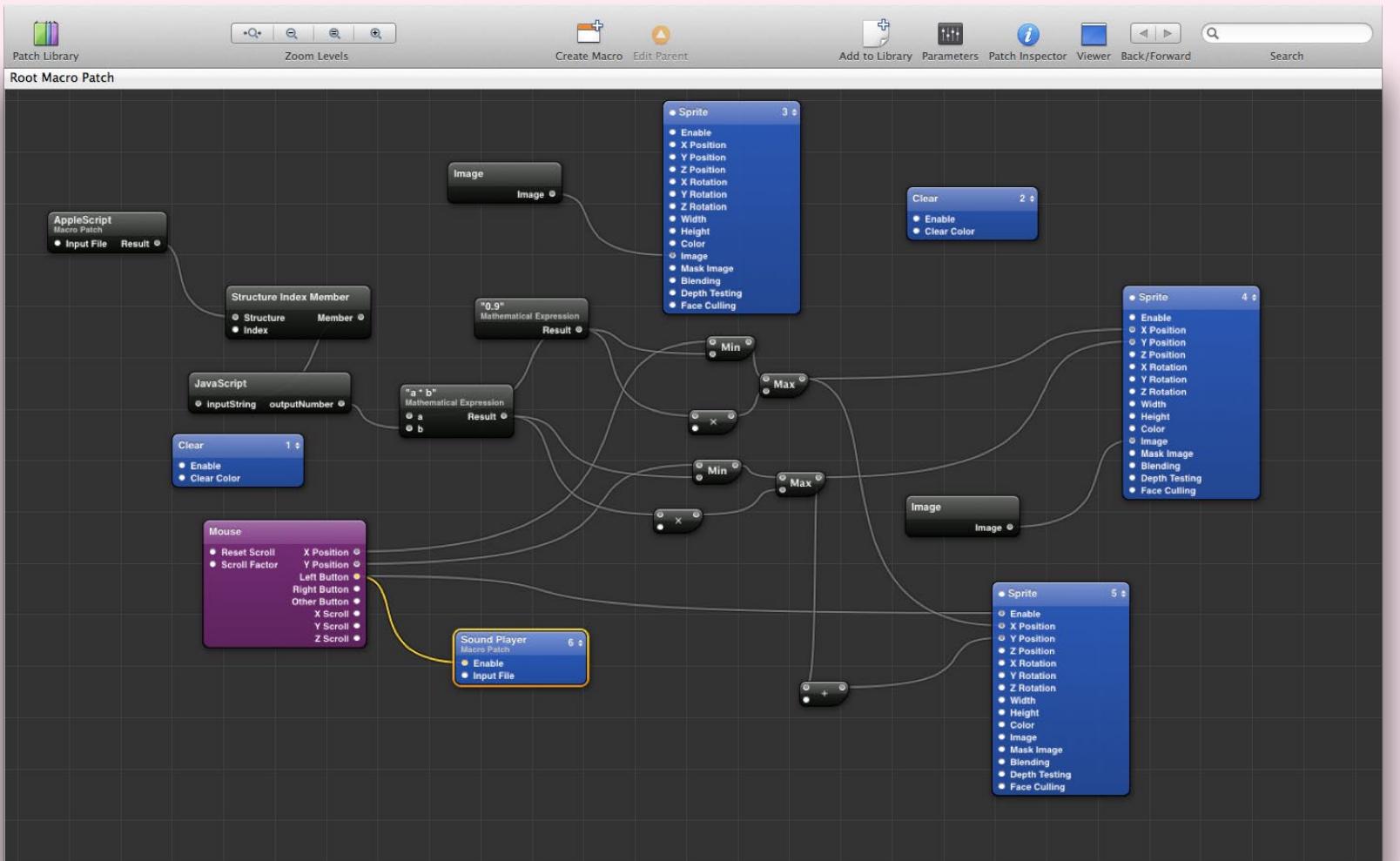
2004-Today



Experimental Media Research Group at Sint Lucas School of Arts of the Karel de Grote-Hogeschool

# Quartz Composer

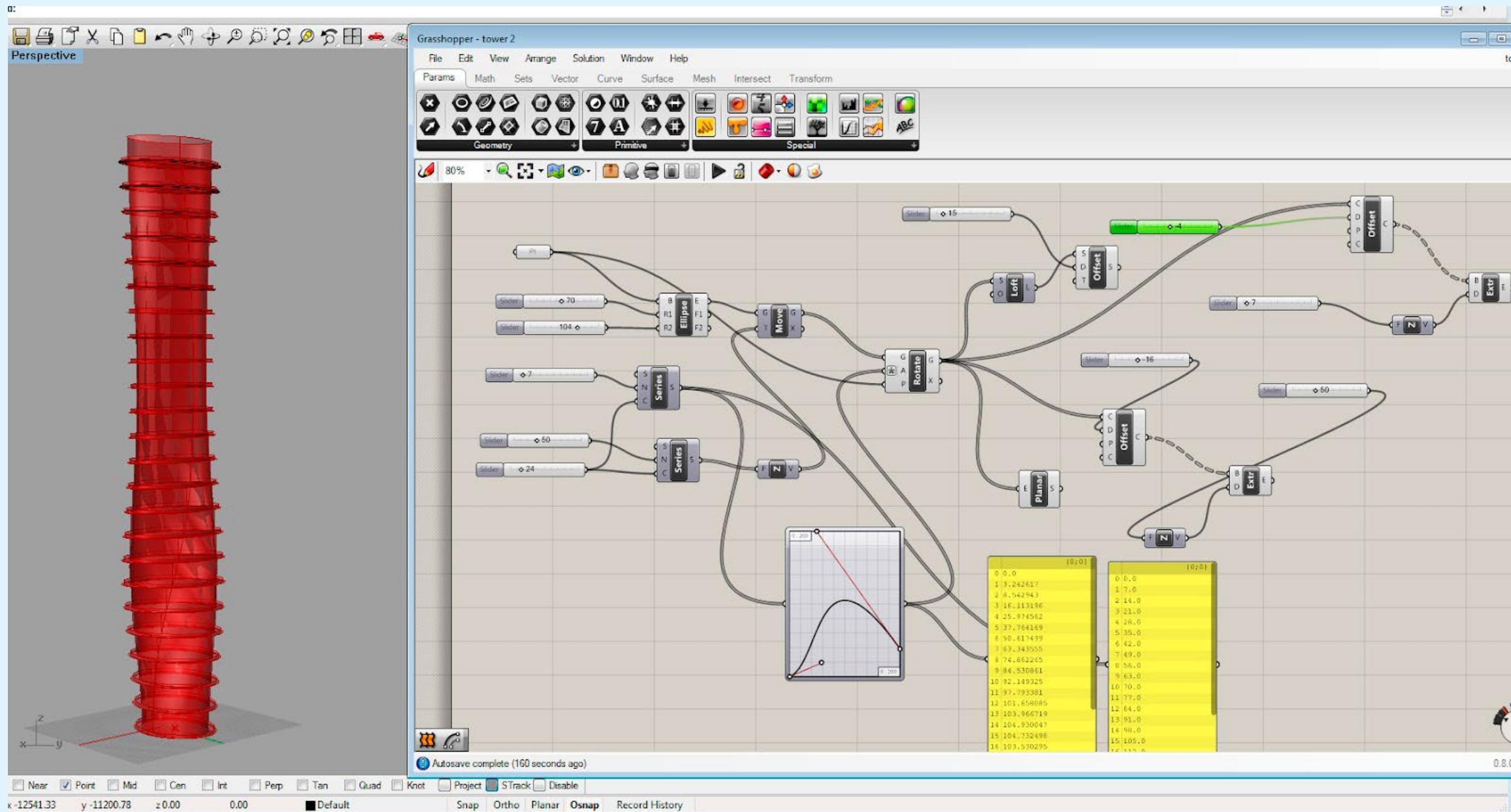
2005-2016



Apple

# Grasshopper for Rhino

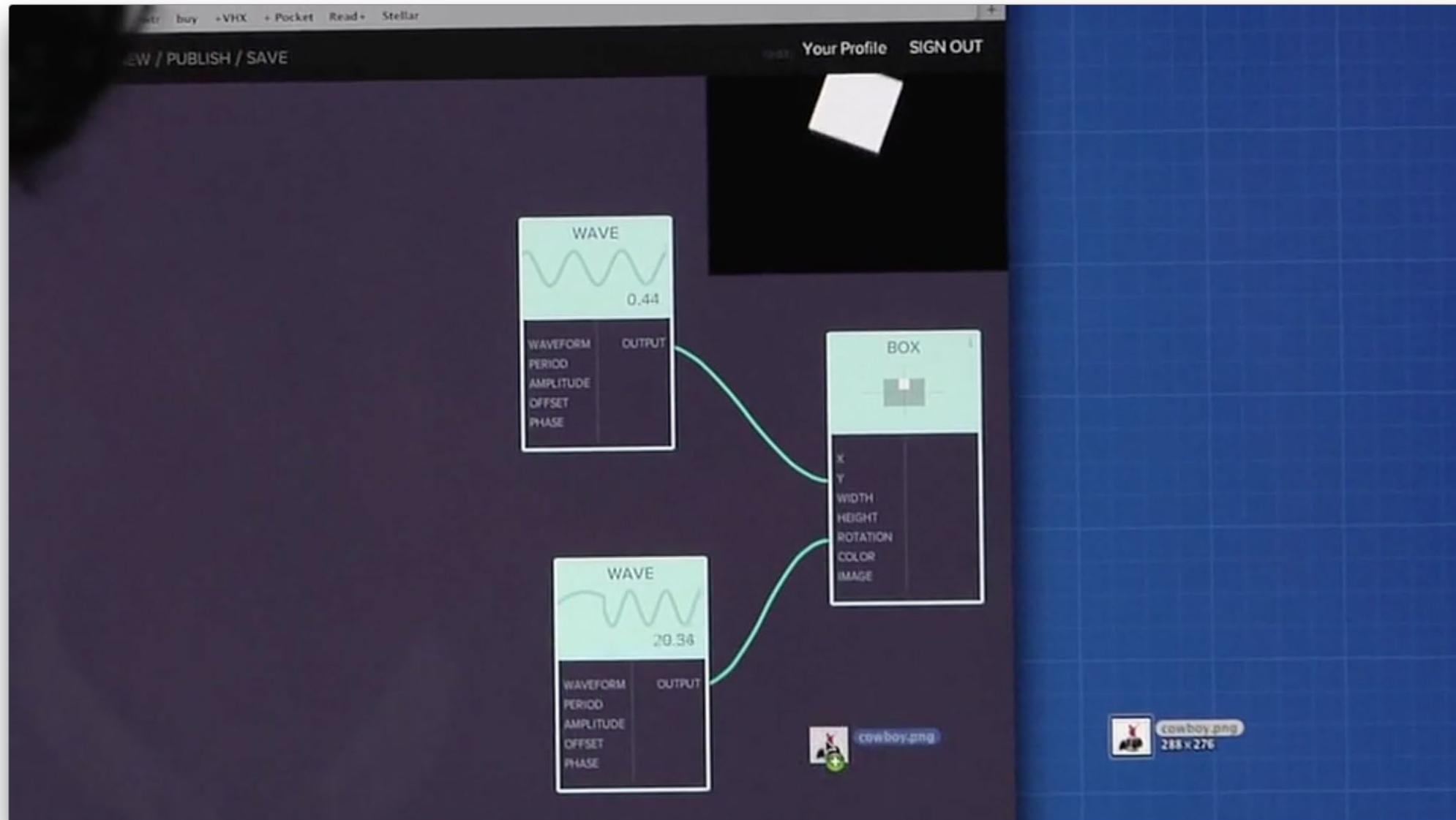
2007-Today



Robert McNeel and David Rutten

# Moonbase

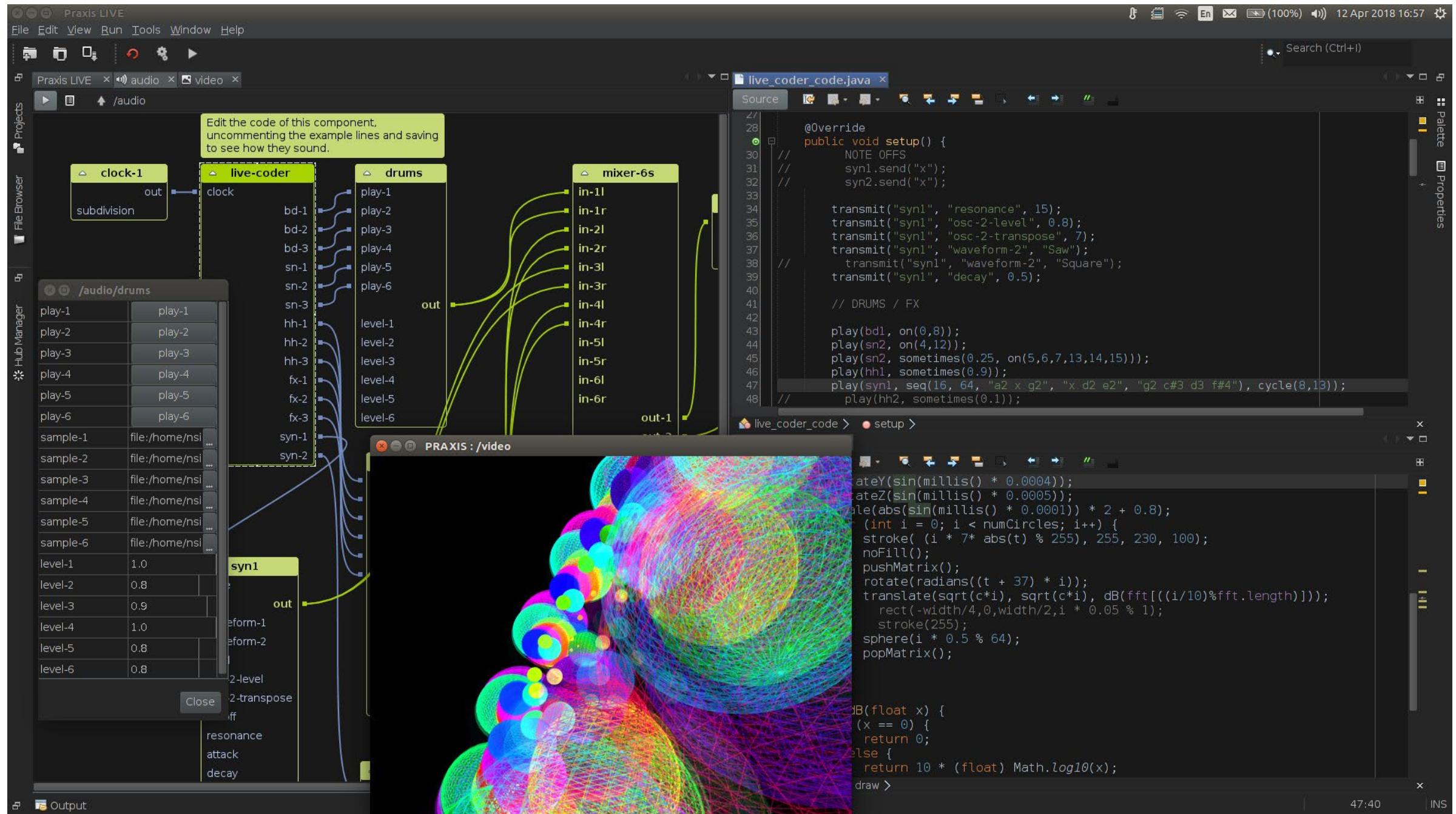
2012



William Cotton and Pasquale D'Silva

# Praxis Live

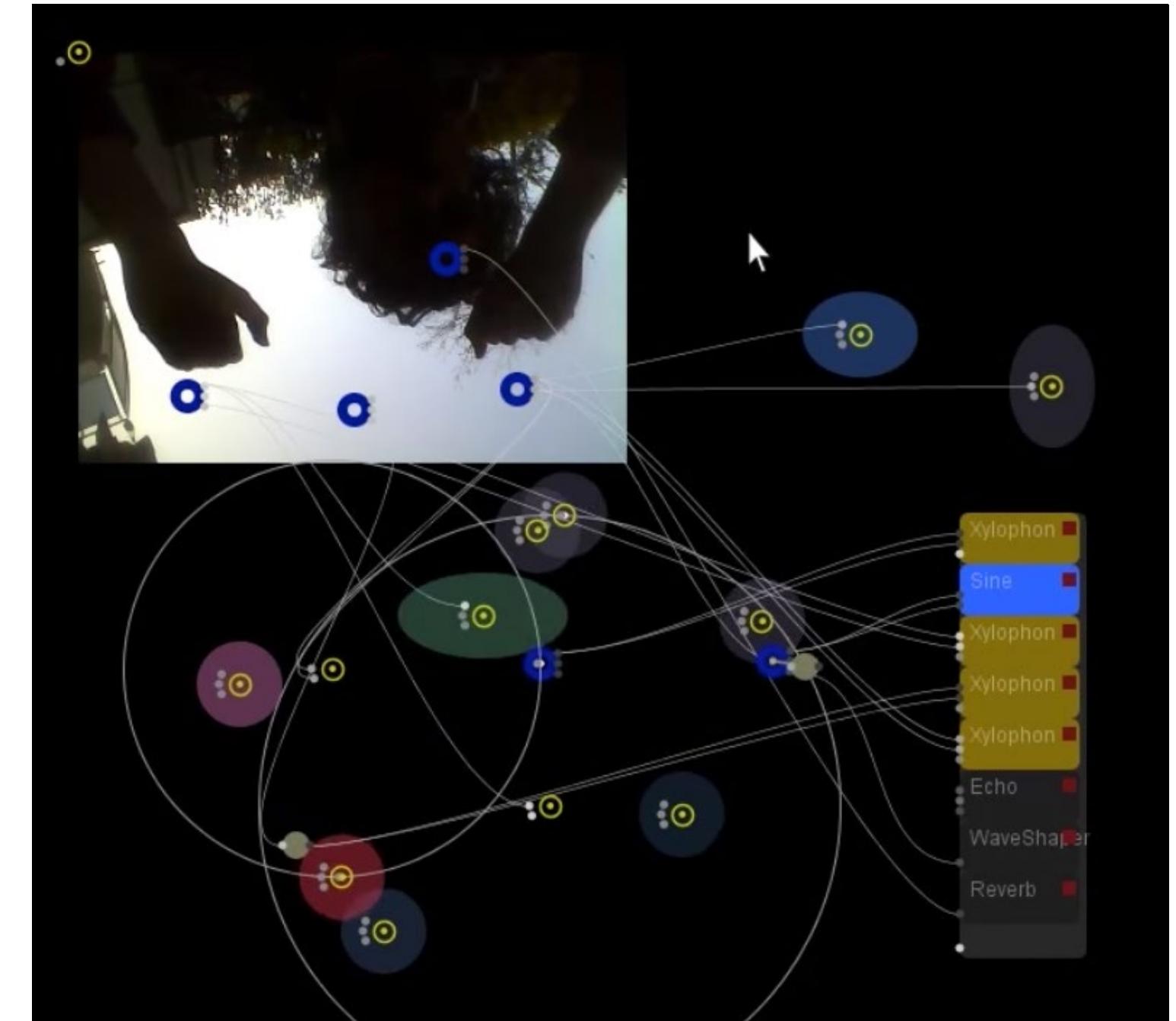
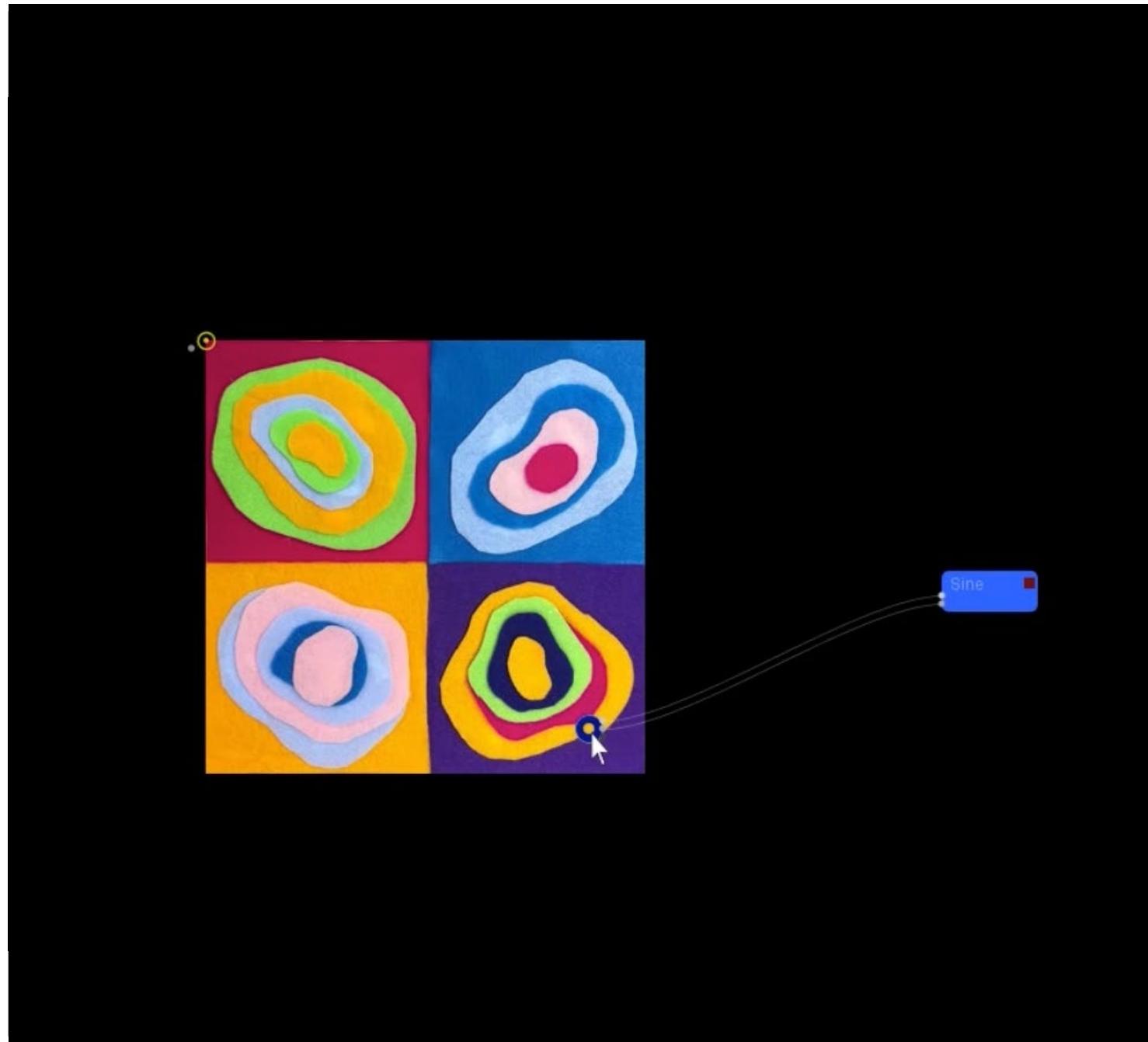
2012-Today



Neil C Smith

# Loligo

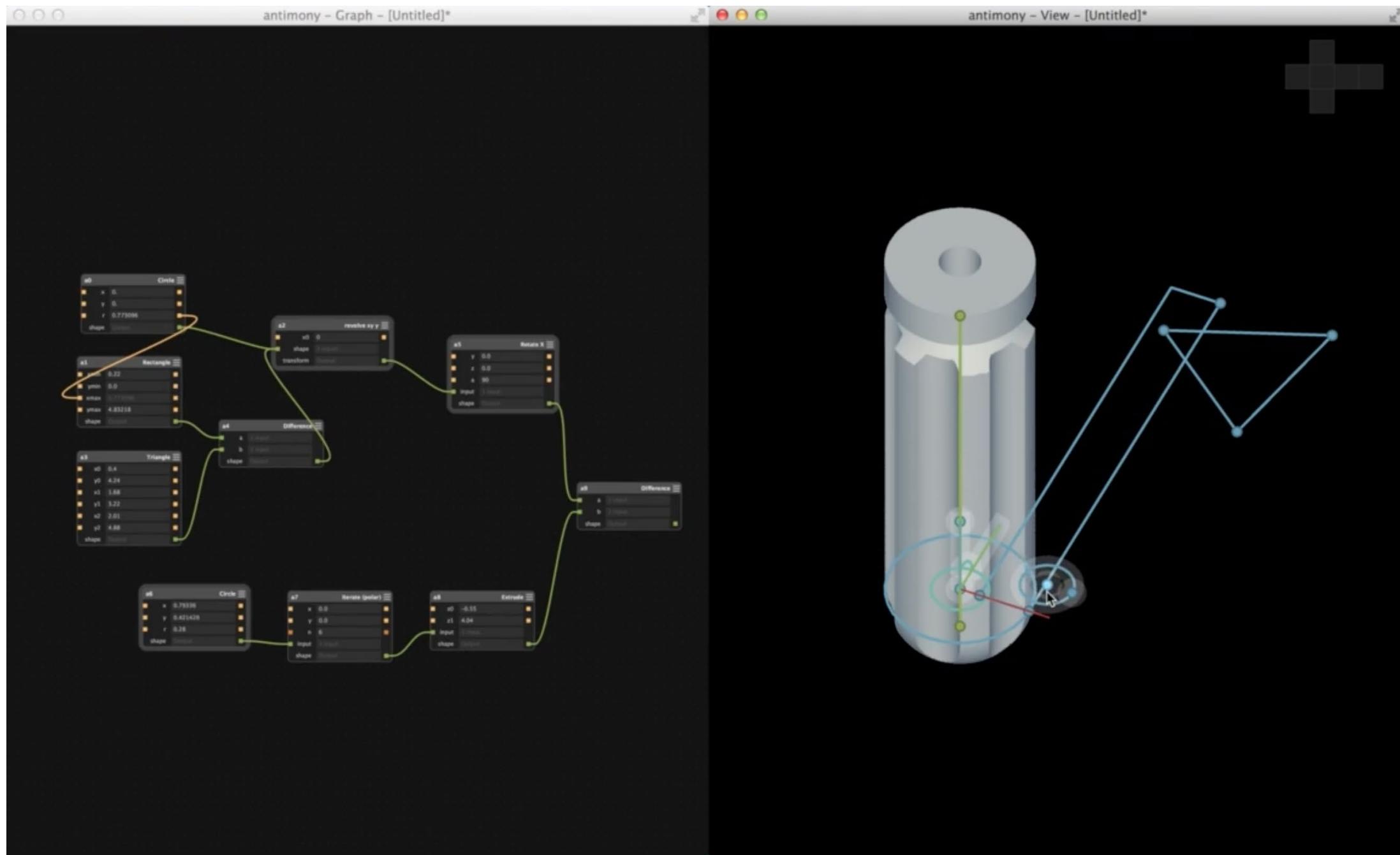
# 2014



Vanja Cuk

# Antimony

2015



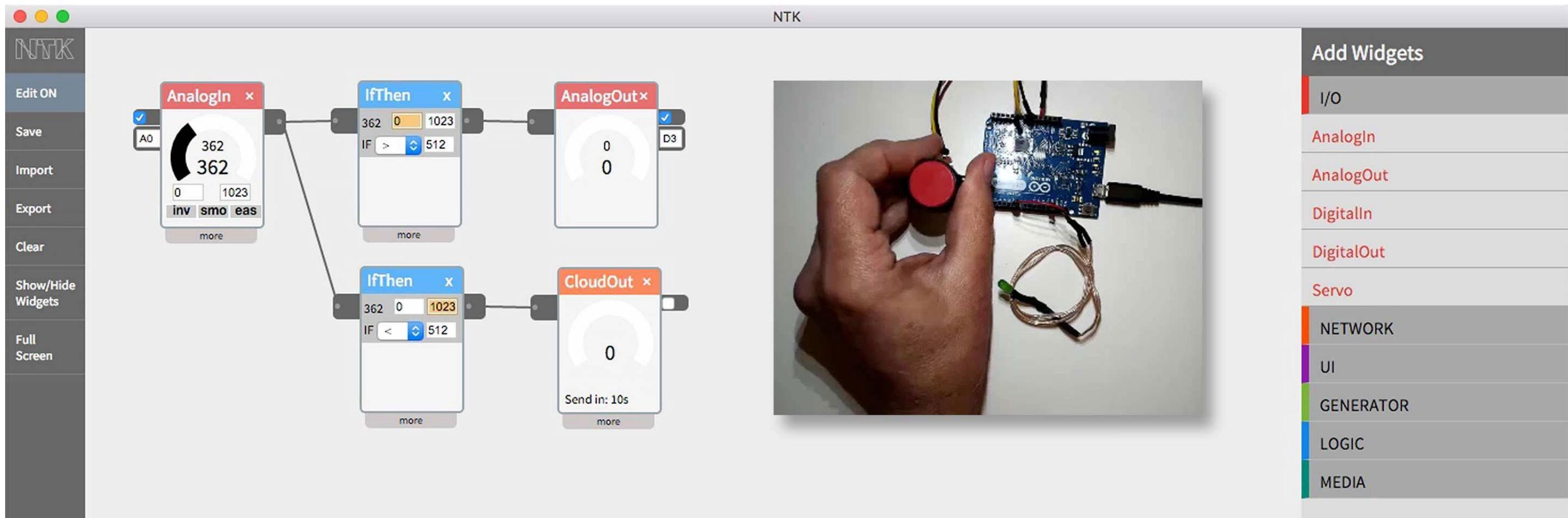
Matt Keeter

# Interaction Prototyping

Empower designers to Build  
'Working-Prototypes'

# Net Lab Toolkit (NTK)

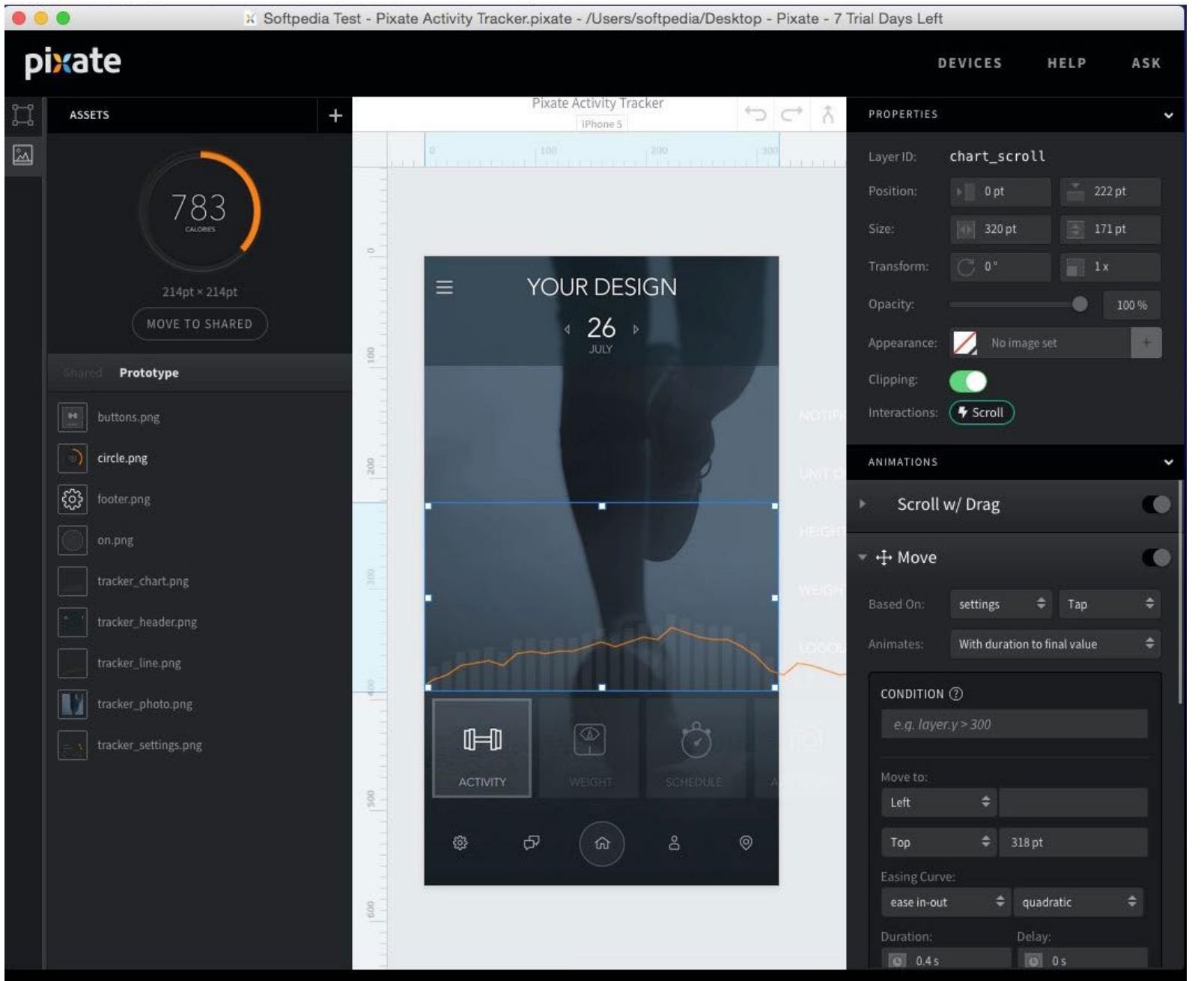
2003-Today



Philip van Allen at ArtCenter College of Design

# Pixate

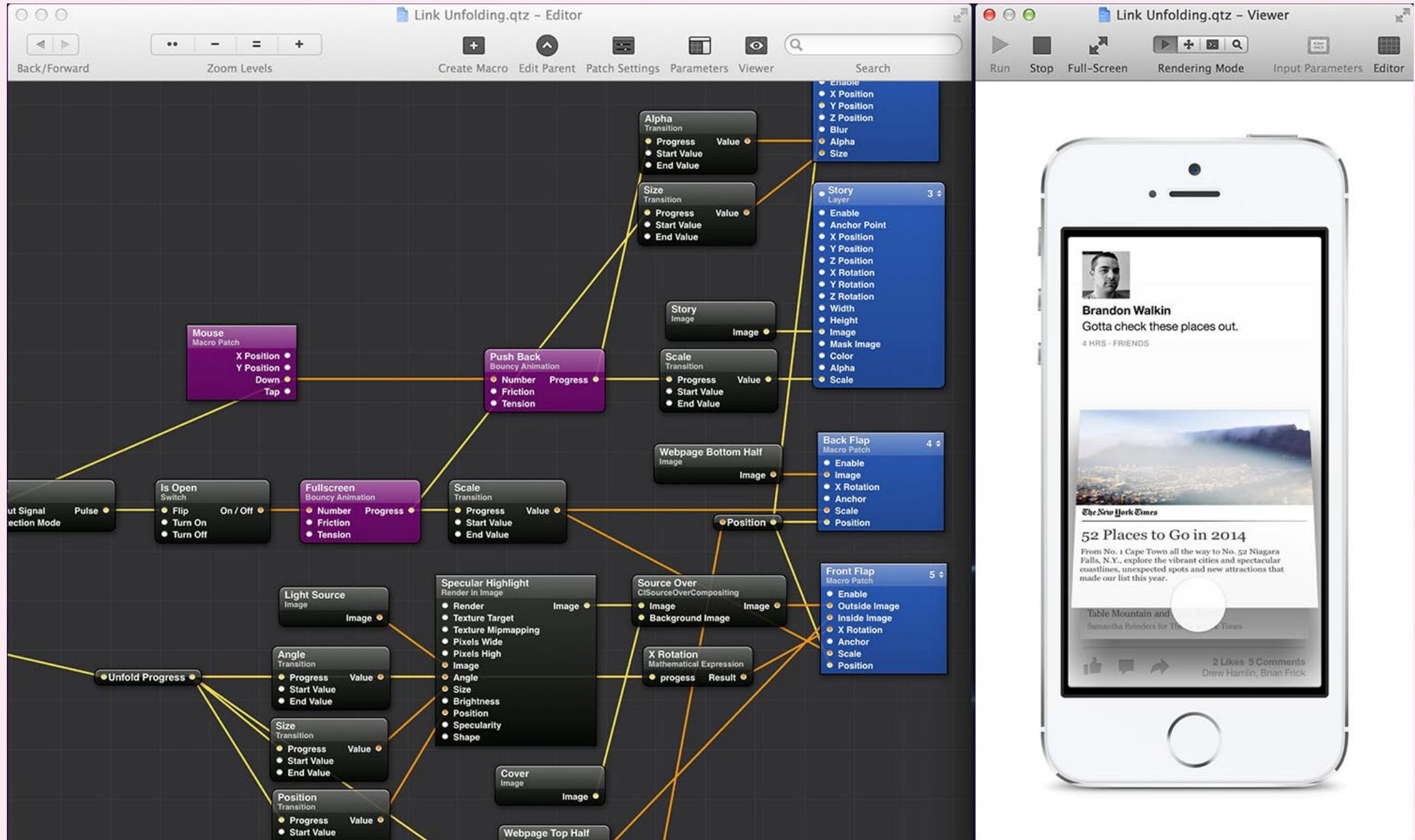
2012-2016



Kevin Lindsey and Paul Colton for Pixate (Acquired by Google)

# Origami for Quartz Composer

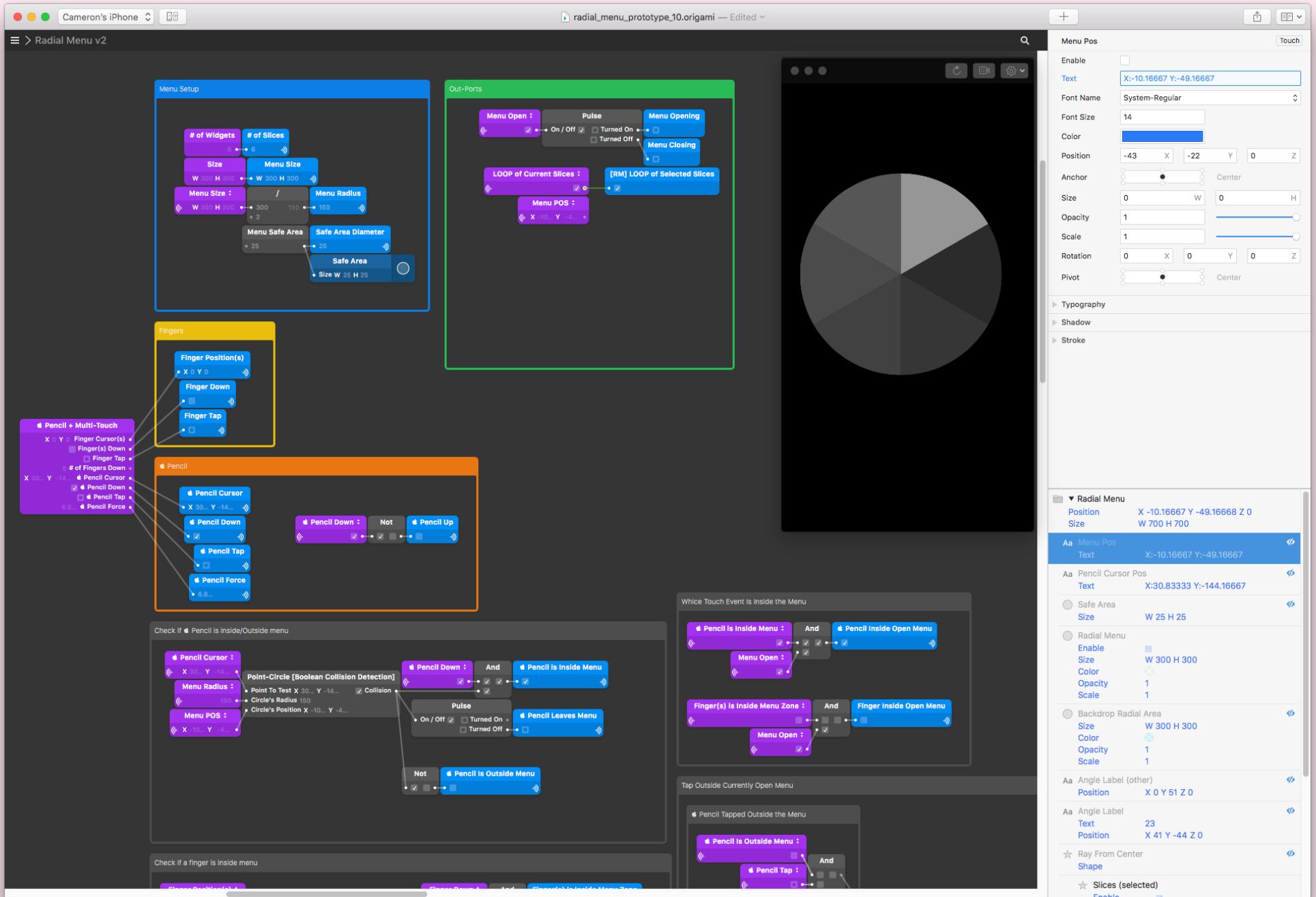
2013-2016



Facebook

# Origami Studio

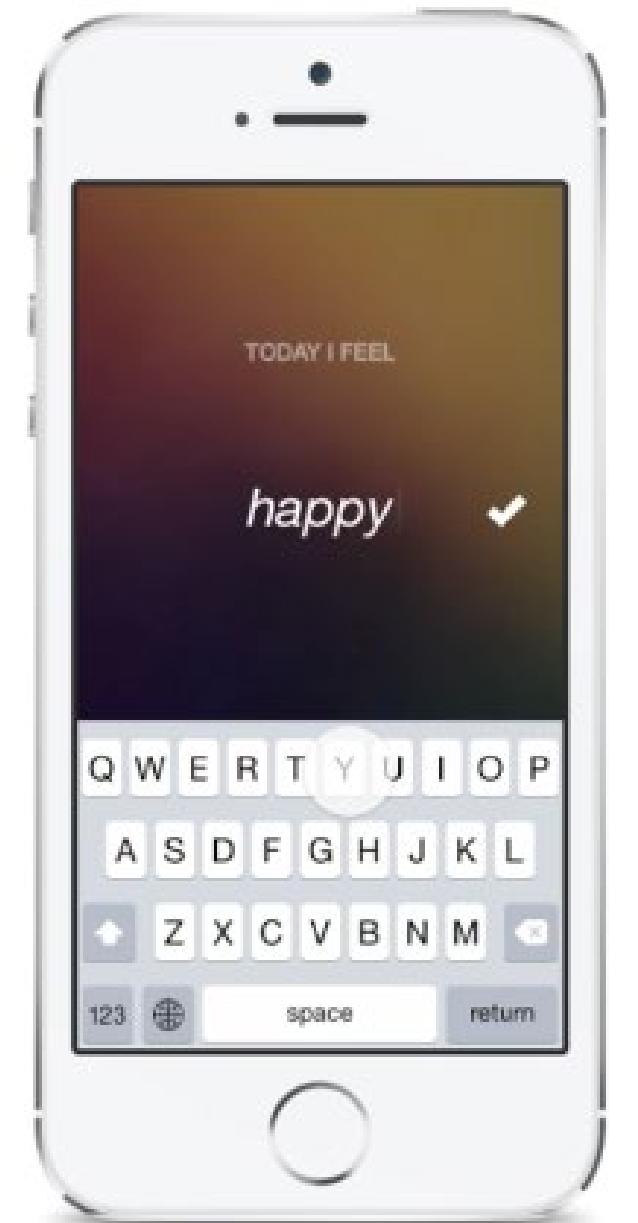
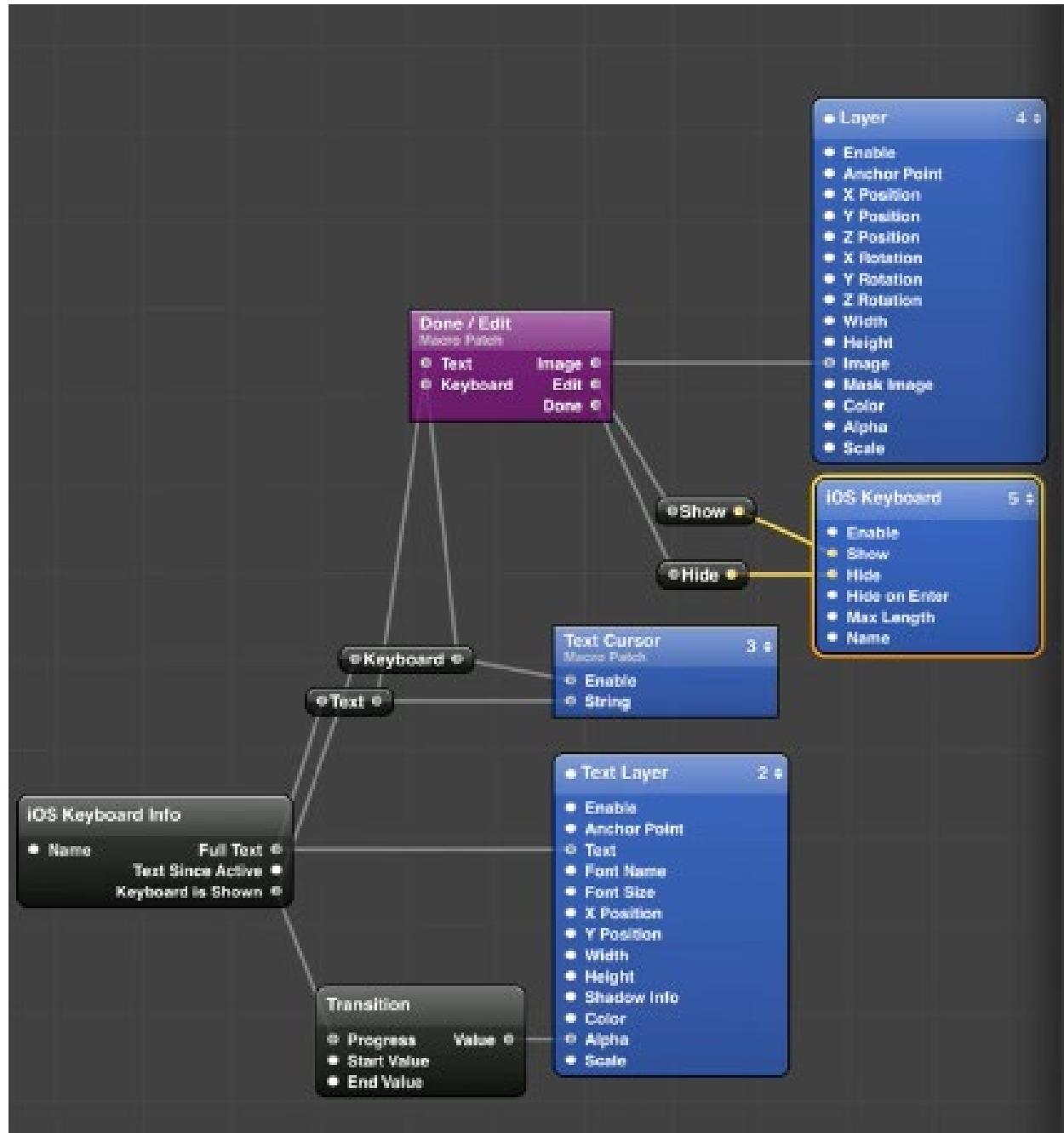
2016-Today



Facebook

# Avocado for Quartz Composer

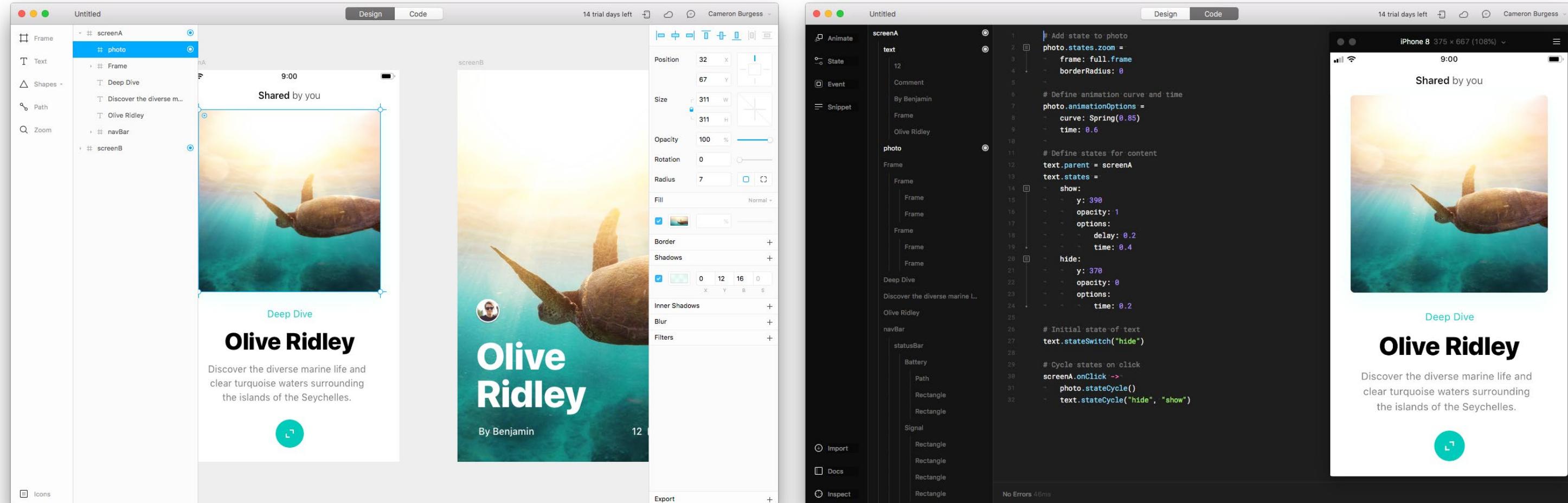
2014



Marco Triverio at IDEO LABS

# Framer Studio

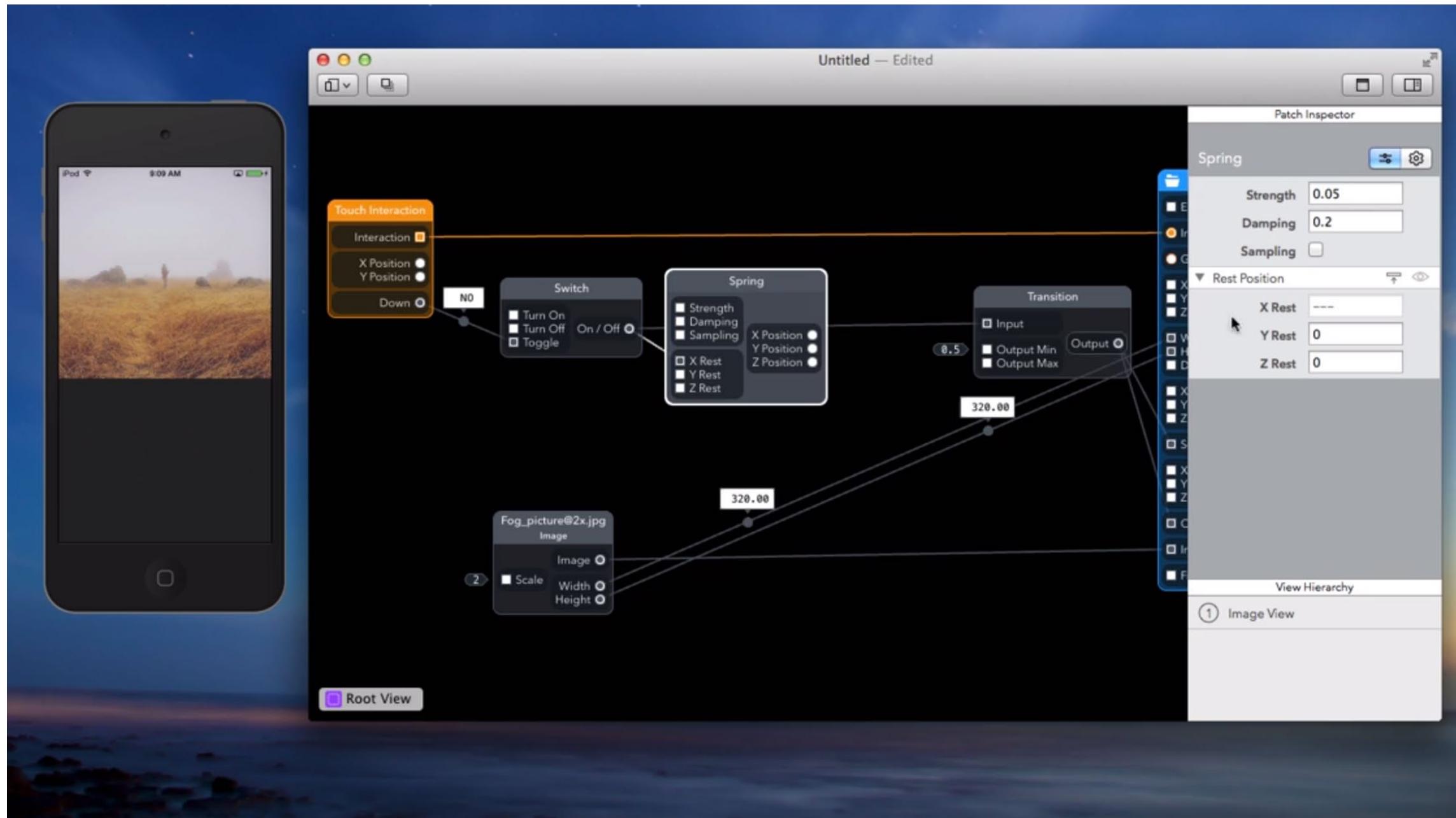
2013-Today



Koen Bok and Jorn van Dijk for Framer

# Form

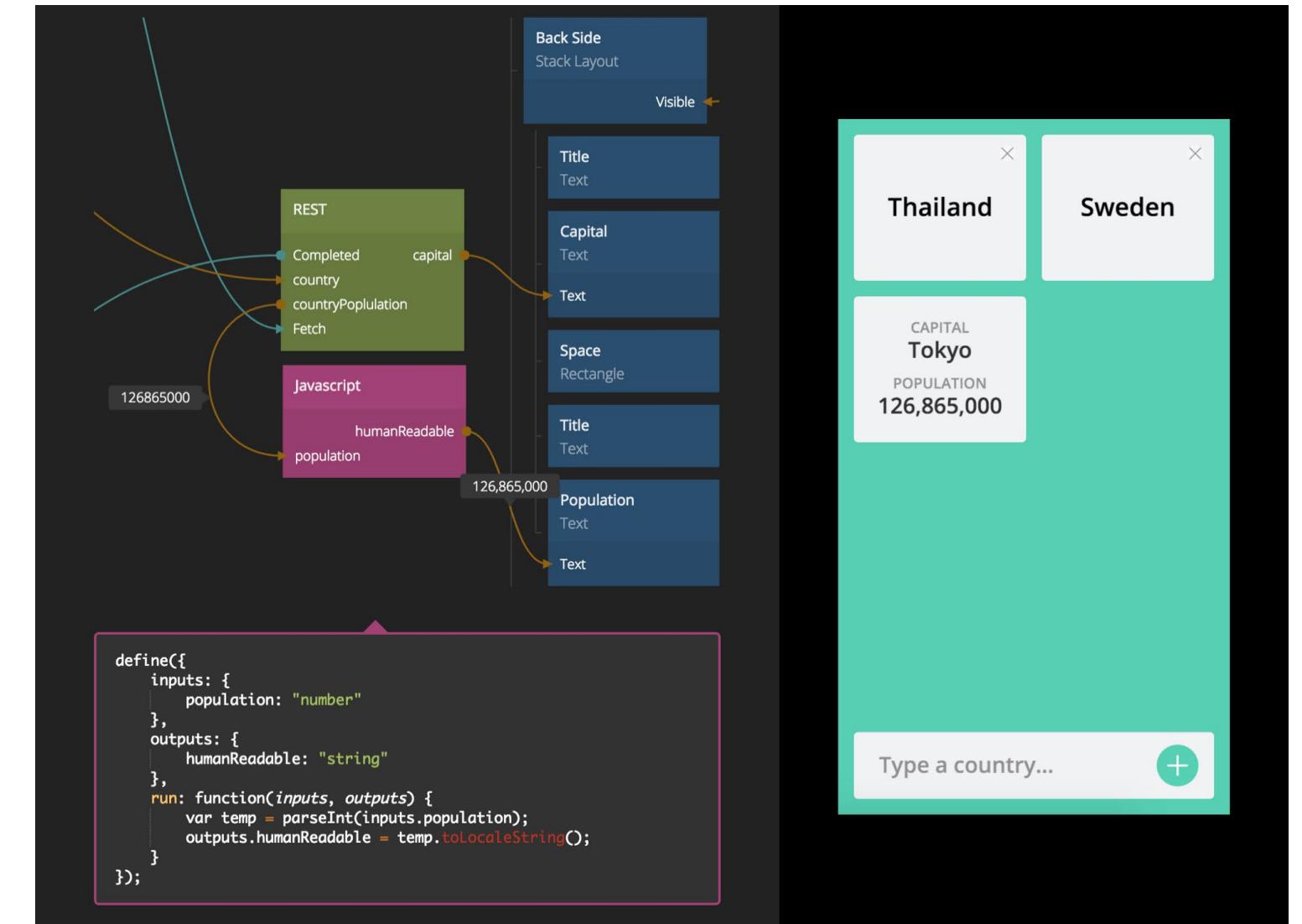
2012-2014



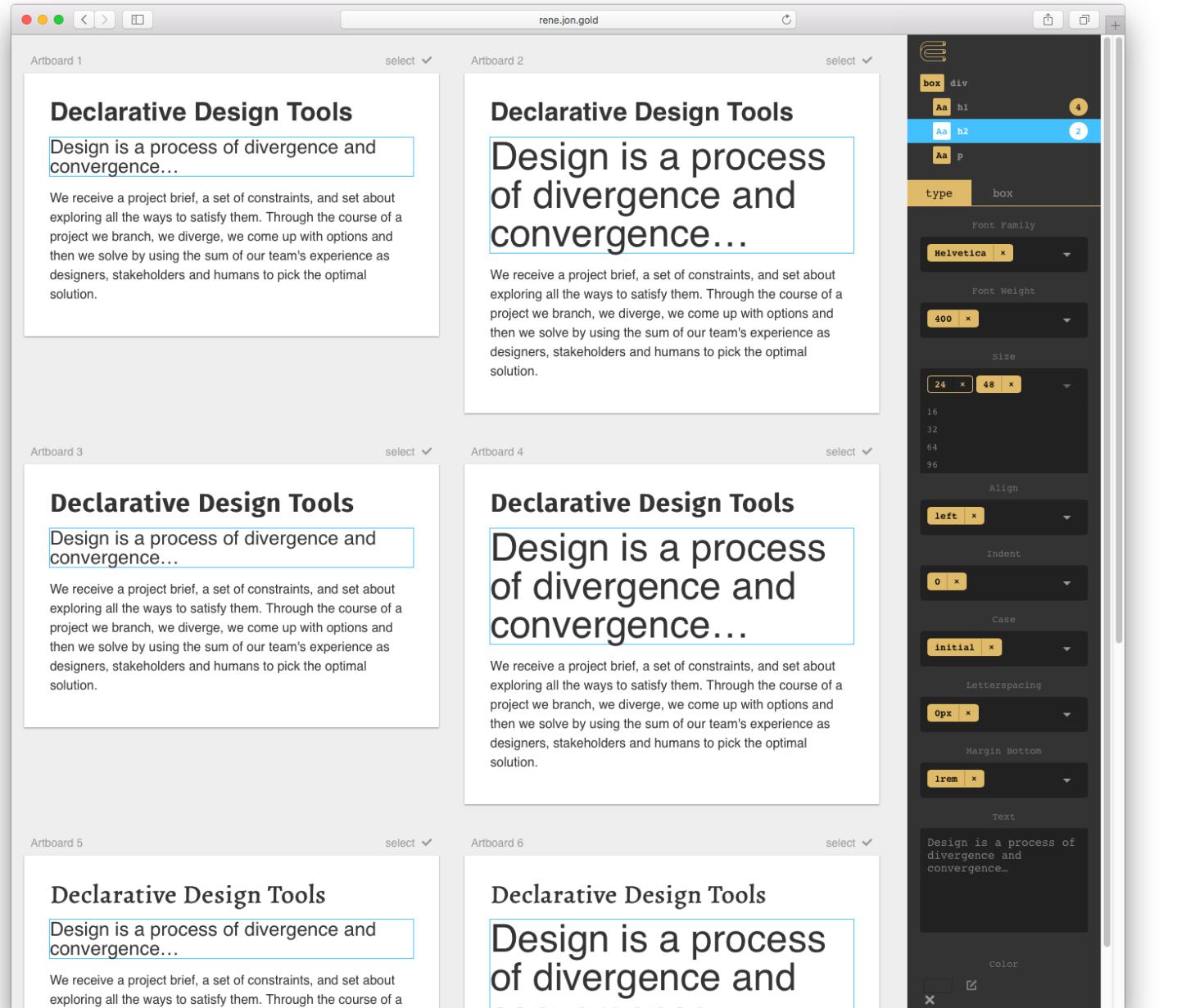
Max Weisel for RelativeWave (Acquired by Google)

# Noodl

2014-Today



Topp Design



Jon Gold

# Sketch.systems

2018-Today

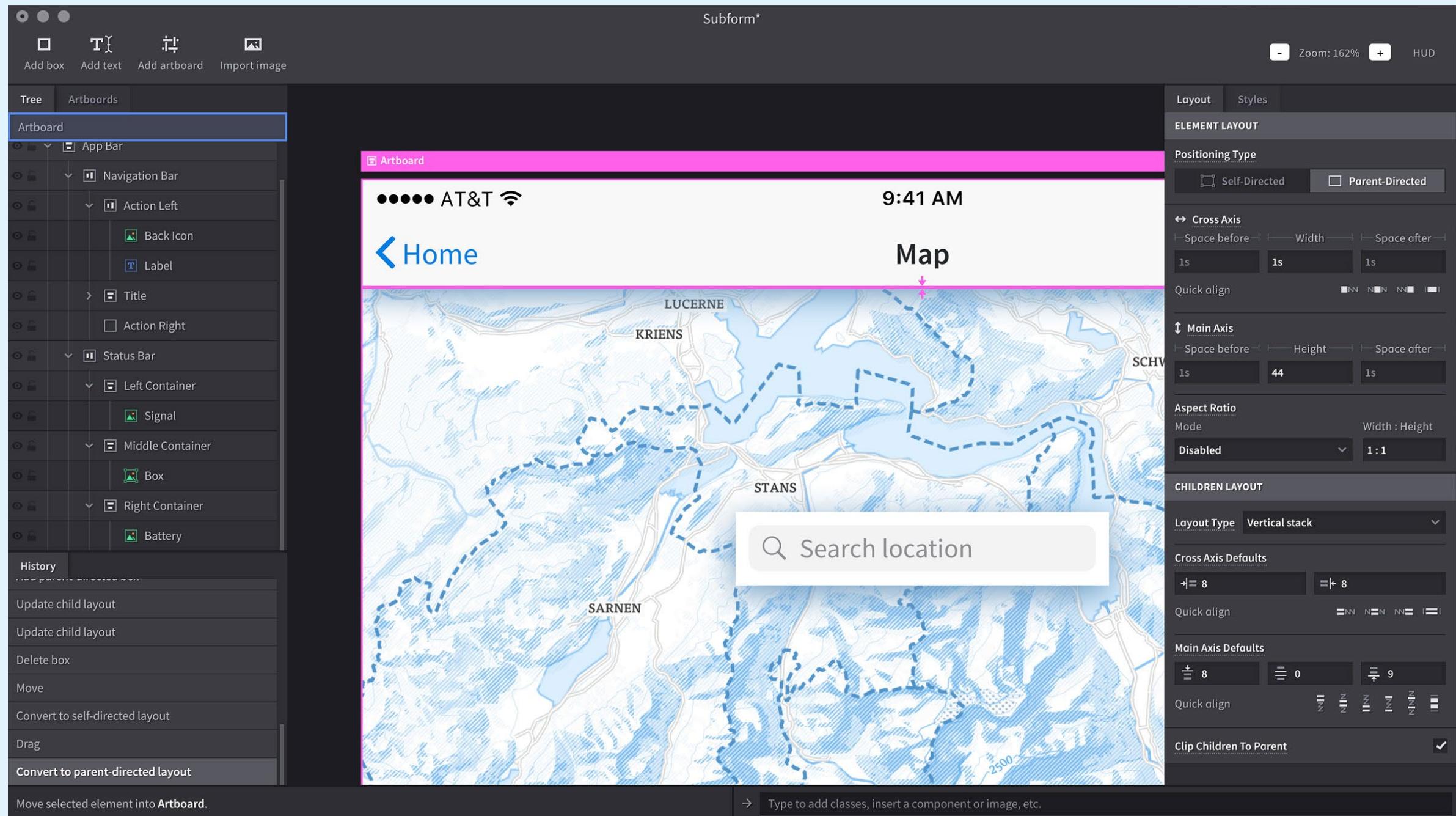
The screenshot shows the Sketch.systems interface for a search bar component. The interface is divided into several sections:

- Spec (See tutorial)**: A list of 12 items describing the search bar's states and transitions. The first few items are:
  - Search Bar\*
  - Inactive\*
  - focused → Active
- Vis**: A visualization of the search bar states. It shows a large blue box labeled "Inactive" with "focused" below it, and a smaller white box labeled "Active" with "canceled", "cleared", and "typed" below it. Below these are three smaller boxes: "Empty", "Text Entry", and "Results".
- Code (See tutorial)**: A block of JavaScript code for rendering the search bar. It includes functions for rendering the main search bar and the search bar itself, which includes an input field with placeholder text and a focus event handler.
- Prototype**: A live prototype of the search bar. It features a search input field with a placeholder "Enter a search term..." and a "Search" button.

Kevin Lynagh and Ryan Lucas for General Reactives L.L.C.

# Subform

2018



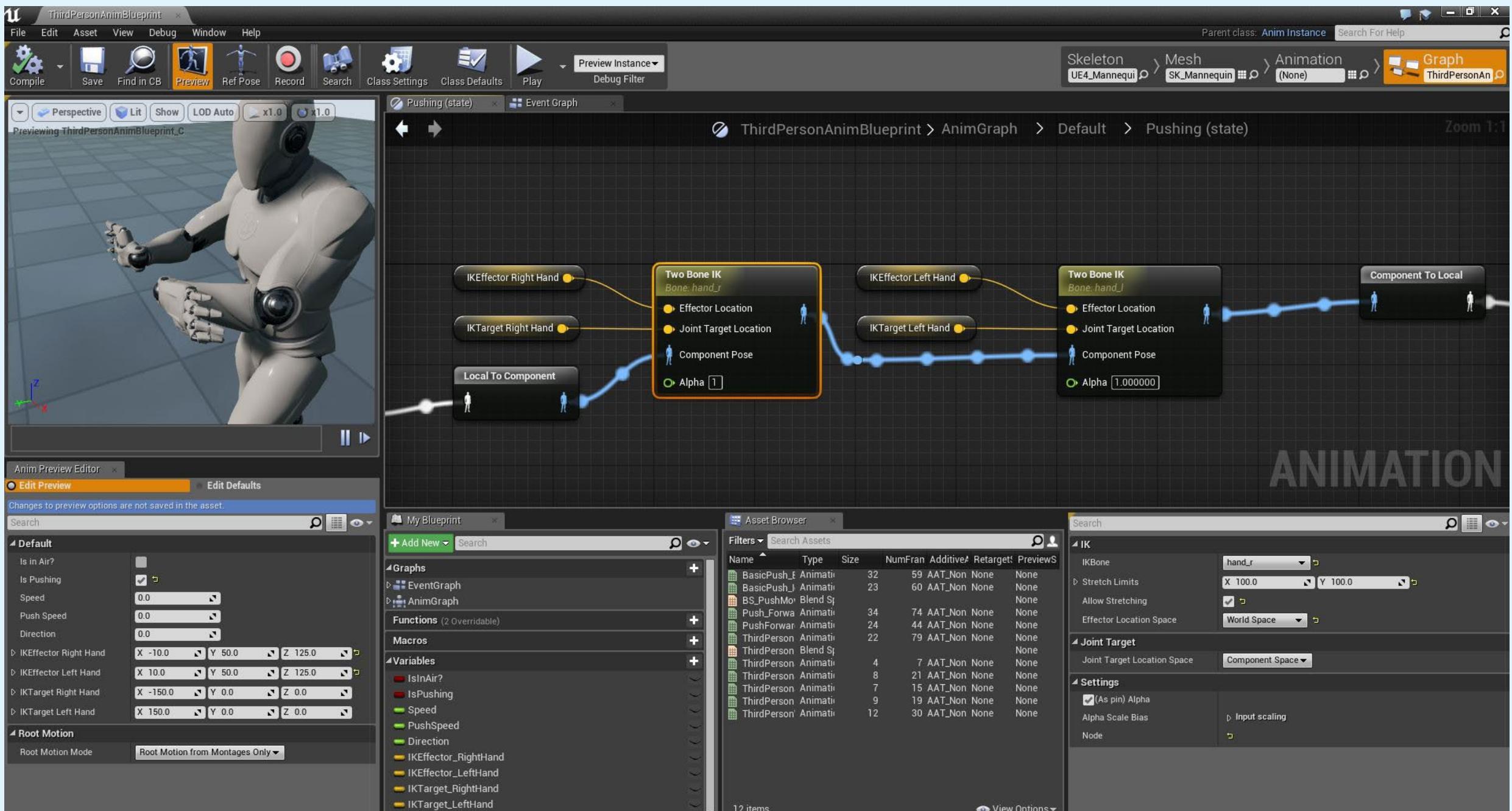
Kevin Lynagh and Ryan Lucas for General Reactives L.L.C.

# Prototyping & Development Tools

Enable more people to build  
'Working & Distributable' artifacts

# Blueprints in Unreal Engine

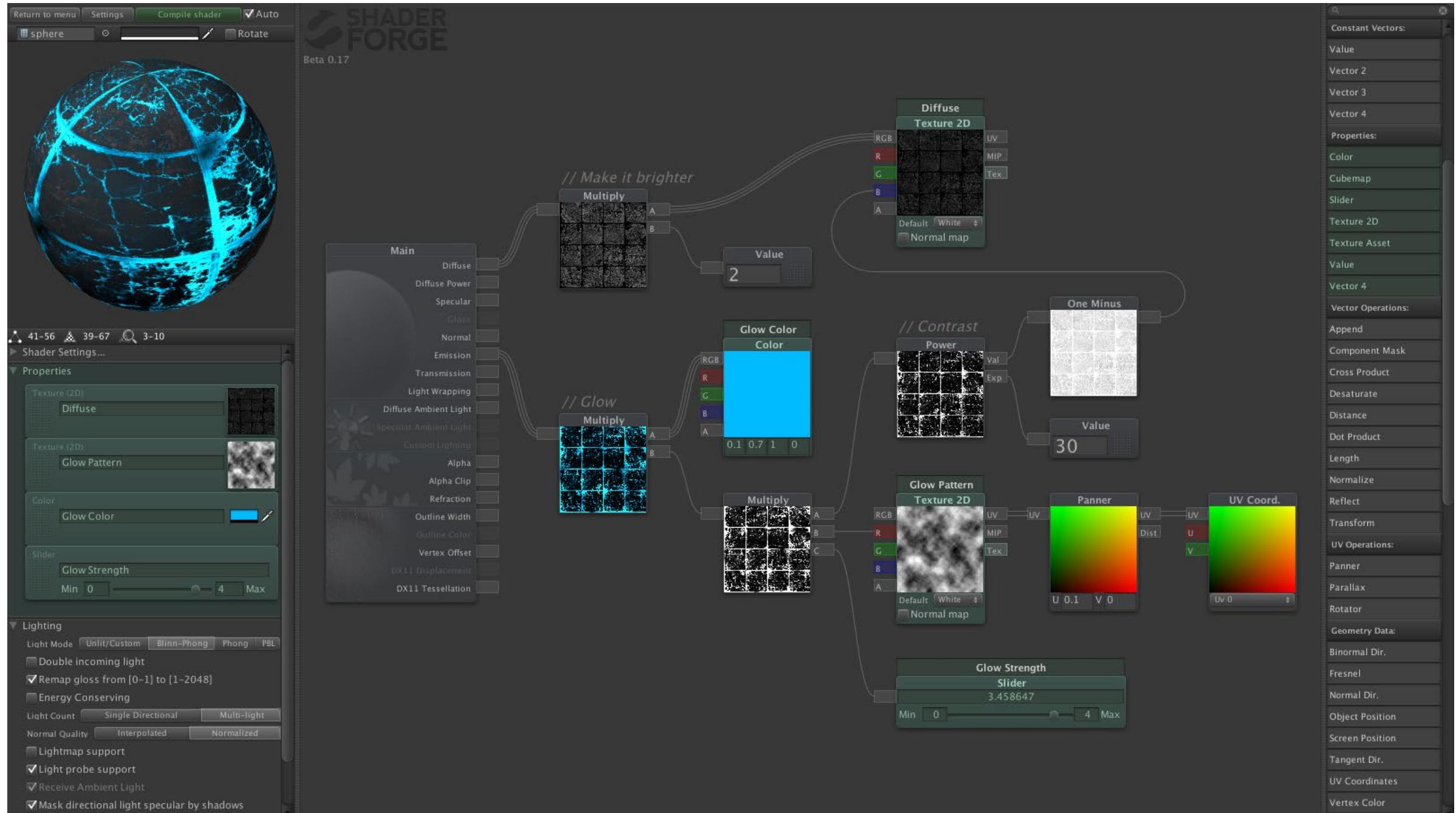
2012-Today



Epic Games

# ShaderForge for Unity

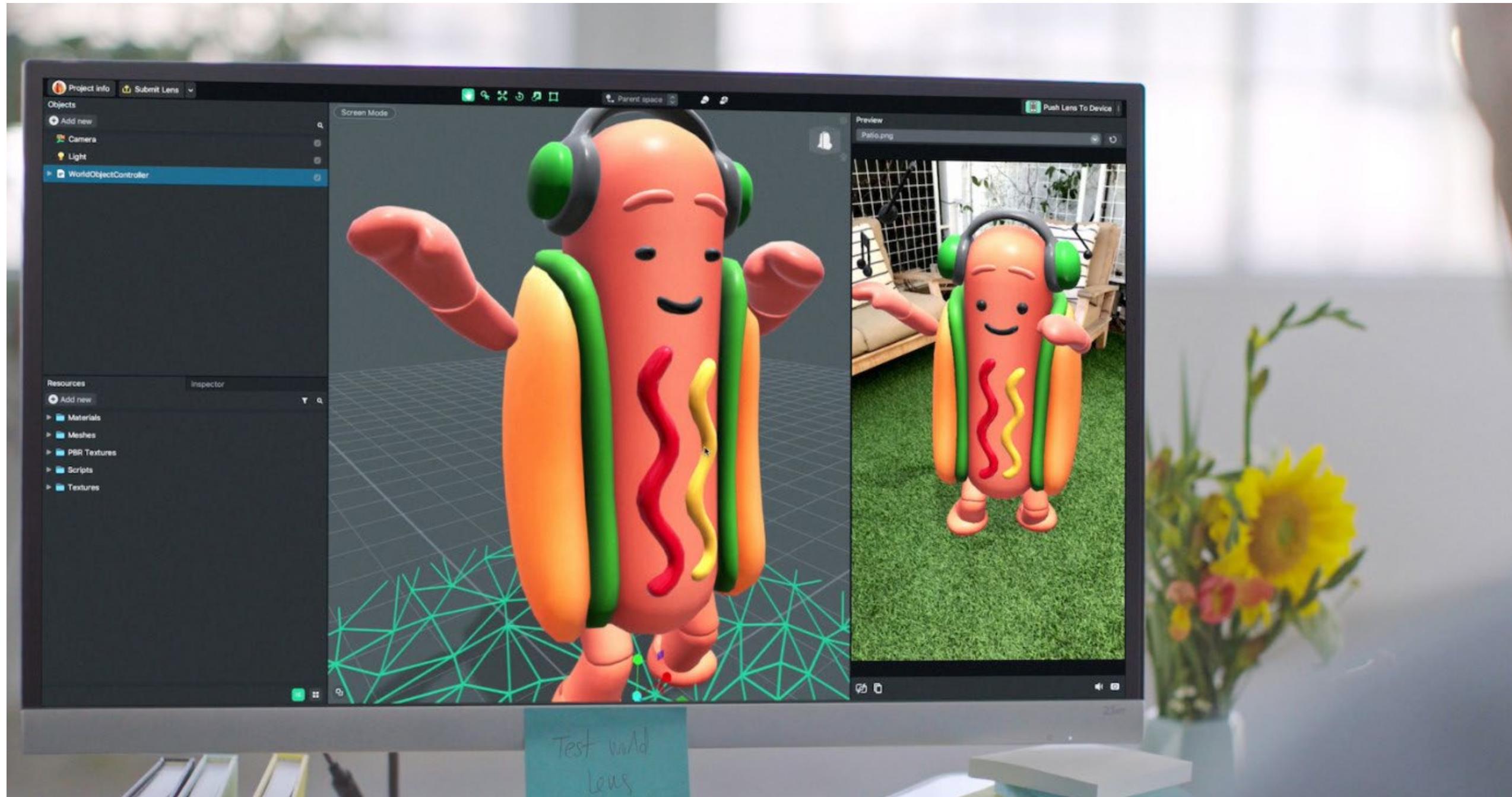
2013-2018



Freya Holmér

# Lens Studio

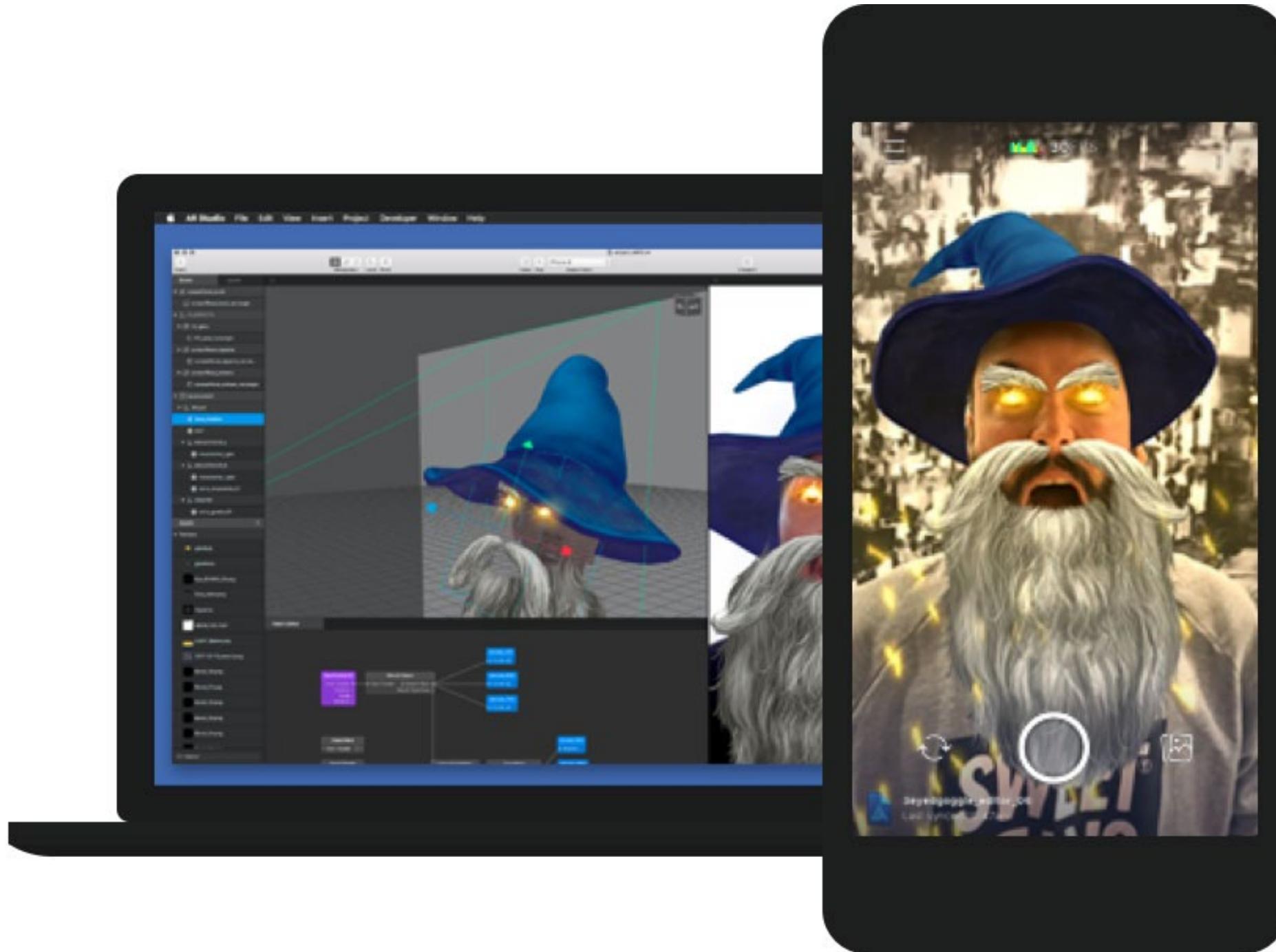
2017-Today



Snap Inc

# AR Studio

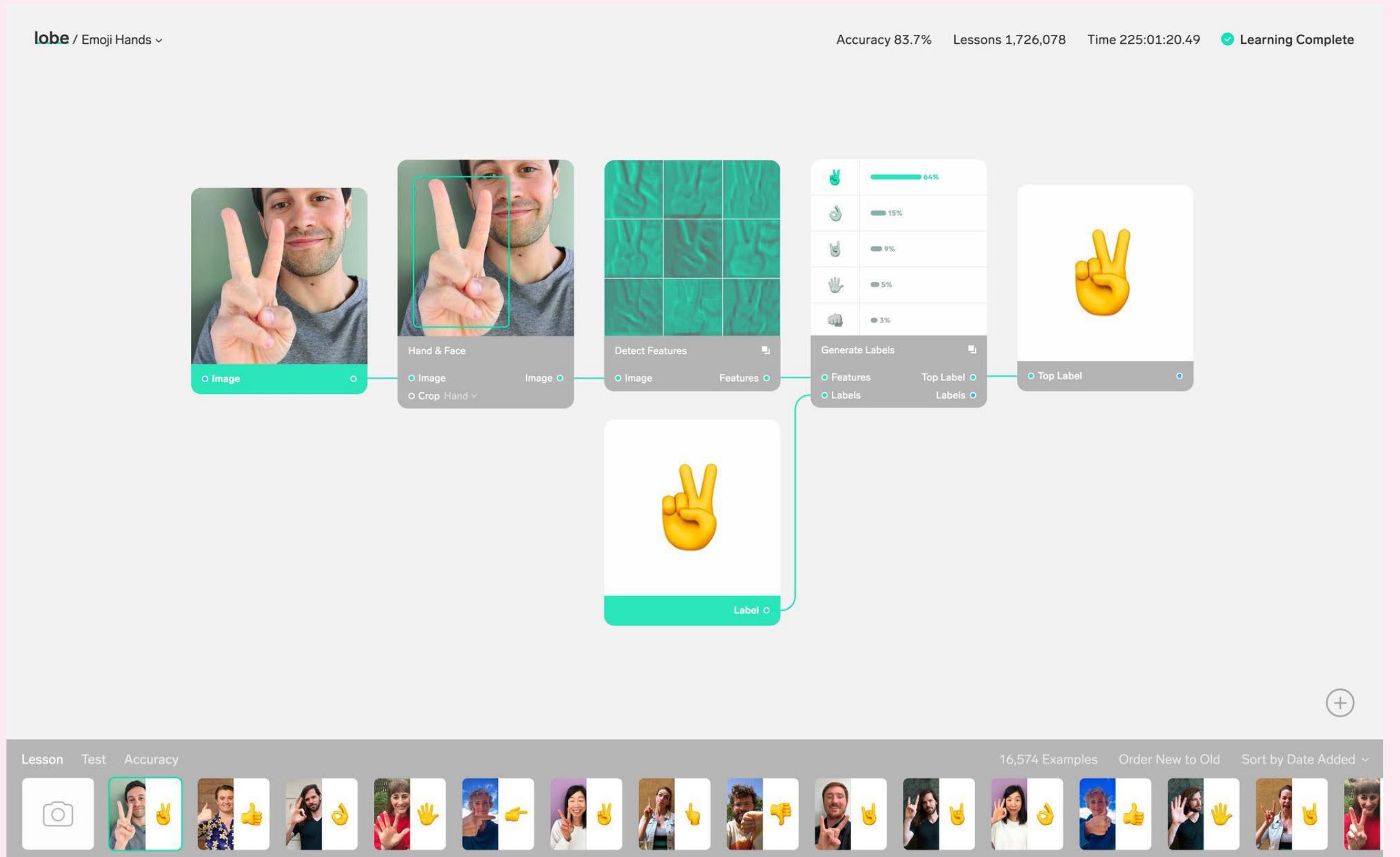
2017-Today



Facebook

# Lobe AI

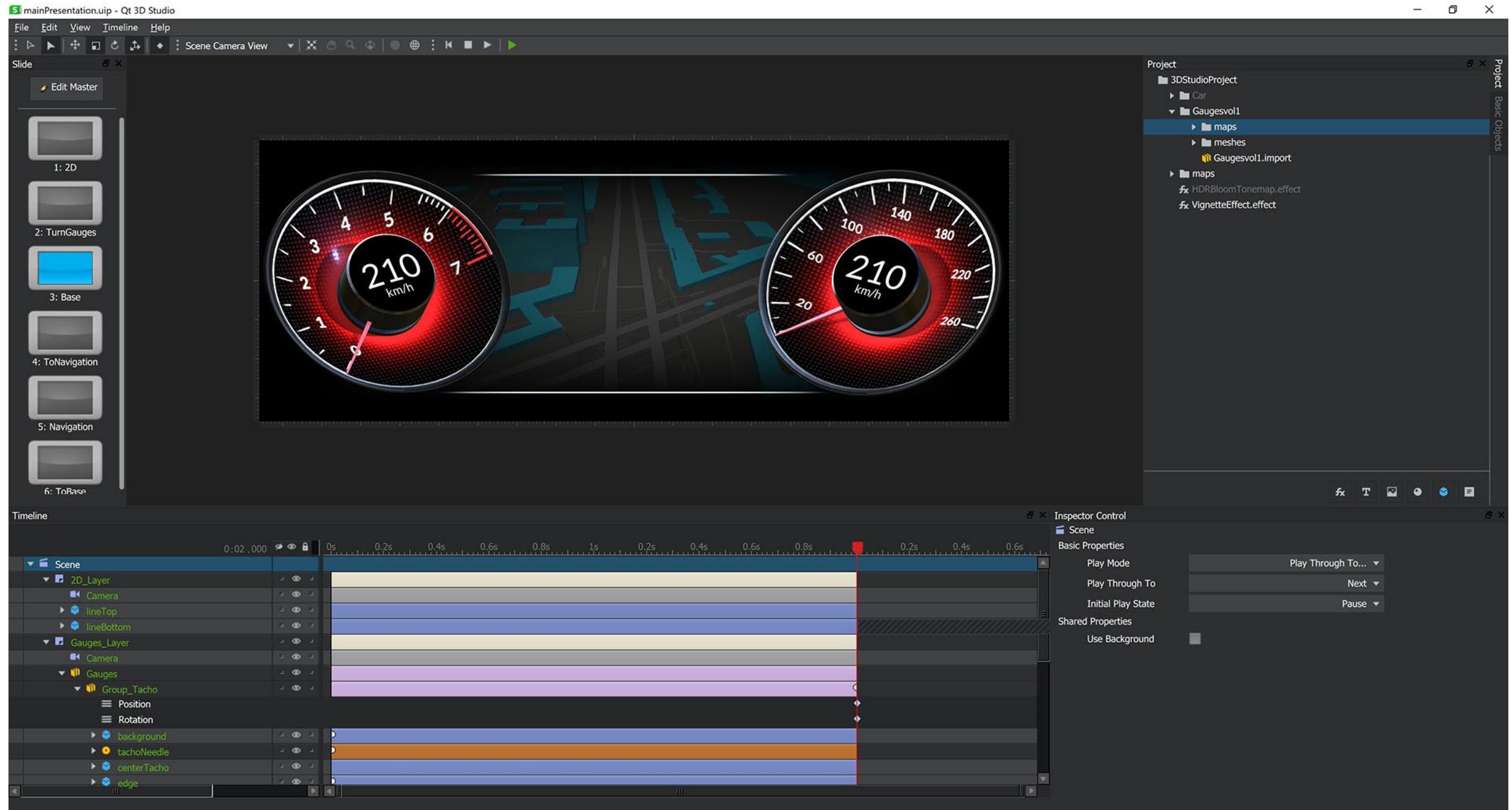
2018-Today



Mike Matas, Adam Menges and Markus Beissinger for Lobe Artificial Intelligence Inc

# Qt 3D Studio

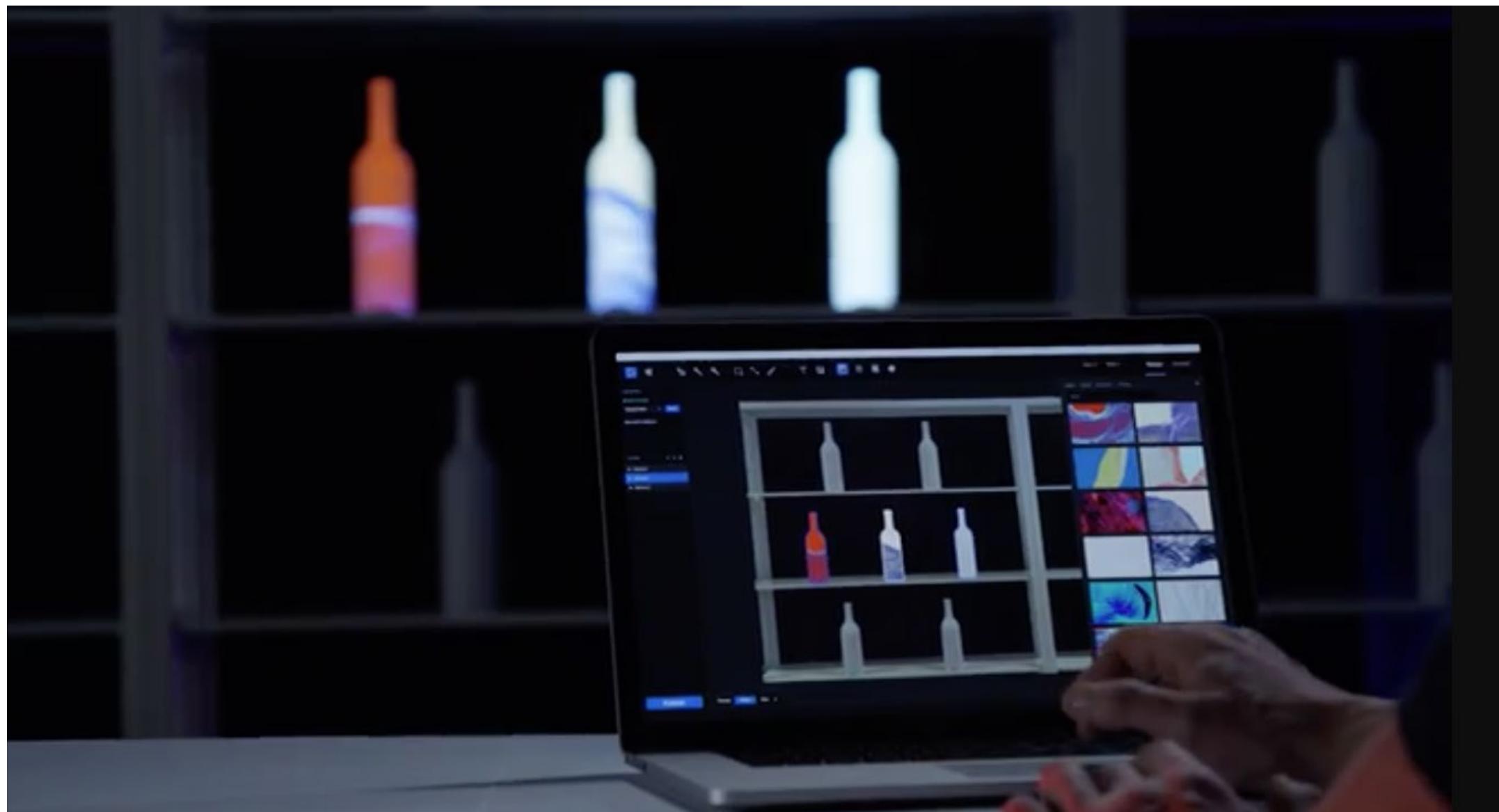
2017-Today



The Qt Company

# Lightform

2018-Today



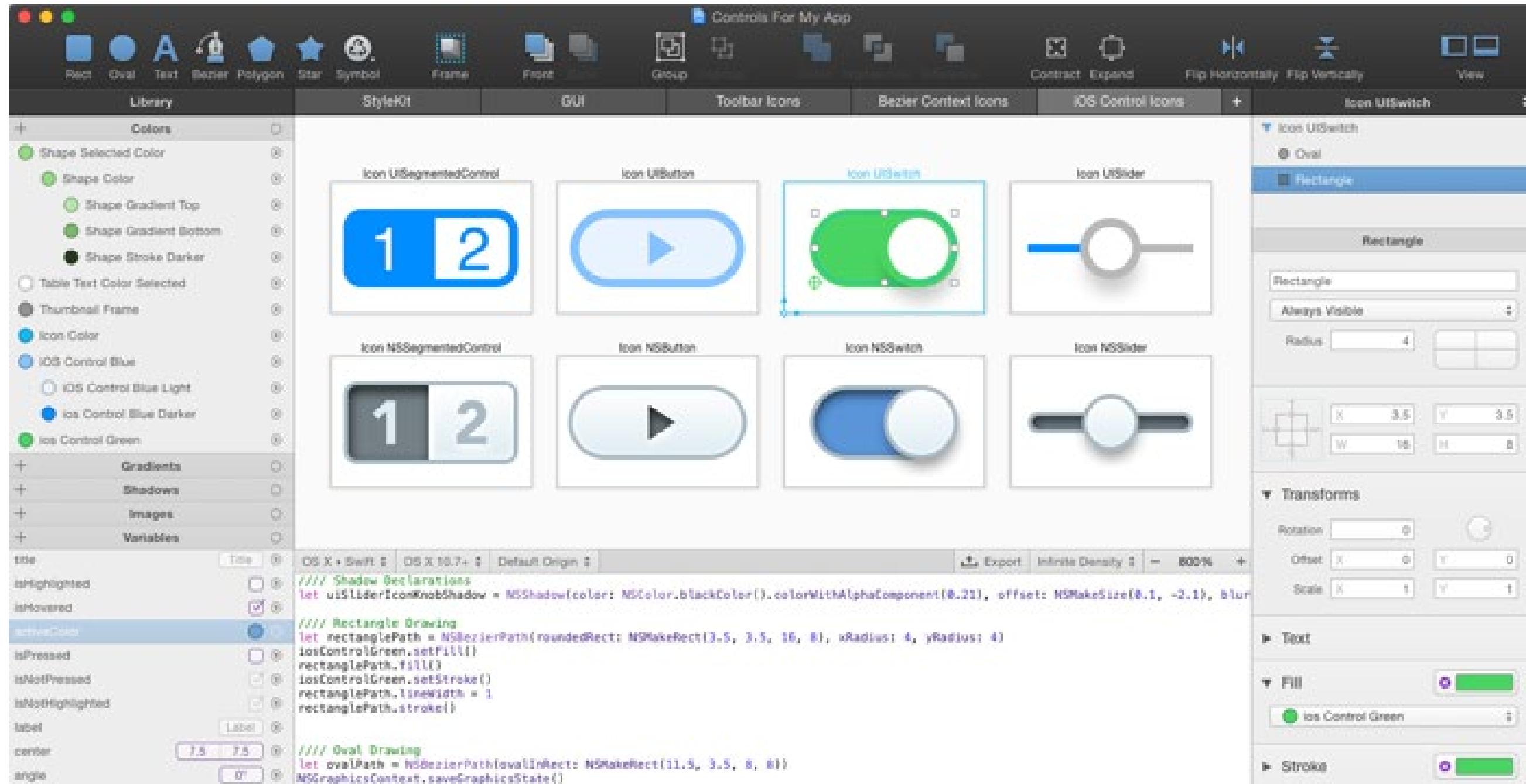
Brett Jones, Kevin Karsch and Rajinder Sodhi

# **Design-Development Toolchain Integration**

Has decreased the distance  
between Design & Engineering

# Paint Code

2012-Today



Peter Krajcik, Mike Antonic, Matt Dunik and Martin Kiss for PixelCut

# PageDraw

2016-Today

The screenshot shows the PageDraw interface with a todo list component. The component is a card with the title "todos" and a sub-section "What needs to be done?". Inside this section, there is a list of todos: "Hello world!" (completed) and "Type something" (default). The "Hello world!" item has a delete button. The "Type something" item has a placeholder "Type something". Below the list, it says "2 items left". A tooltip "Double-click to edit a todo" is visible. The "Draw" tab is selected in the top right, showing the component's structure. The "Code" tab is also visible. The "Component" tab is not selected. The component's properties are shown on the right, including:

- Block Type: Todolitem
- Name: Todolitem Instance
- X: 37, Y: 211, W: 491, H: 58
- Props:
  - Content: Hello world!
  - State: default
- EXPORT PARAMS AS JSON
- Flexibility options: Flexible Left Margin, Flexible Right Margin, Center Horizontally, Flexible Width, Flexible Top Margin, Flexible Bottom Margin

Below the component editor, the code editor shows the component's code:

```
/* import the MainScreen component drawn above in Pagedraw */
import MainScreen from './src/pagedraw/mainscreen'

class App extends Component {
  constructor() {
    super();
    this.state = {
      todos: [{content: "Meet with Yoda"}, {content: "Defeat Darth Vader"}]
    };
  }

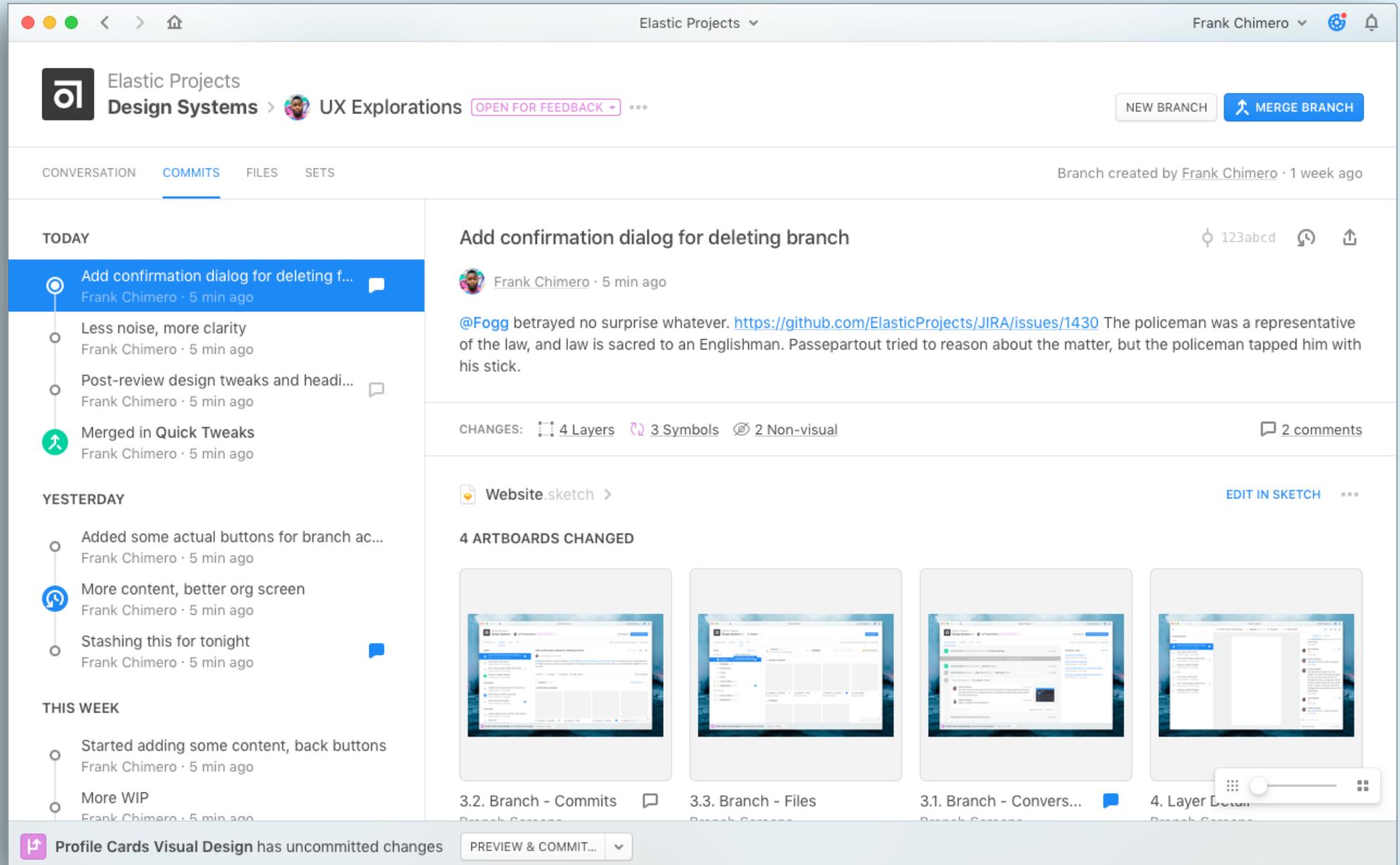
  render() {
    const todos = this.state.todos.map((todo, i) => {
      return {...todo, toggle: (() => this.toggleTodo(i)),
      delete: () => this.deleteTodo(i)}});
    return (
      <MainScreen
        list={todos}
        itemsLeft={this.state.todos.filter((elem) => !elem.completed).length}
        addTodo={this.addTodo}
    );
  }
}
```

The browser preview shows the rendered component with the todos "Meet with Yoda" and "Defeat Darth Vader".

Jared Pochtar

# Abstract

2016-Today

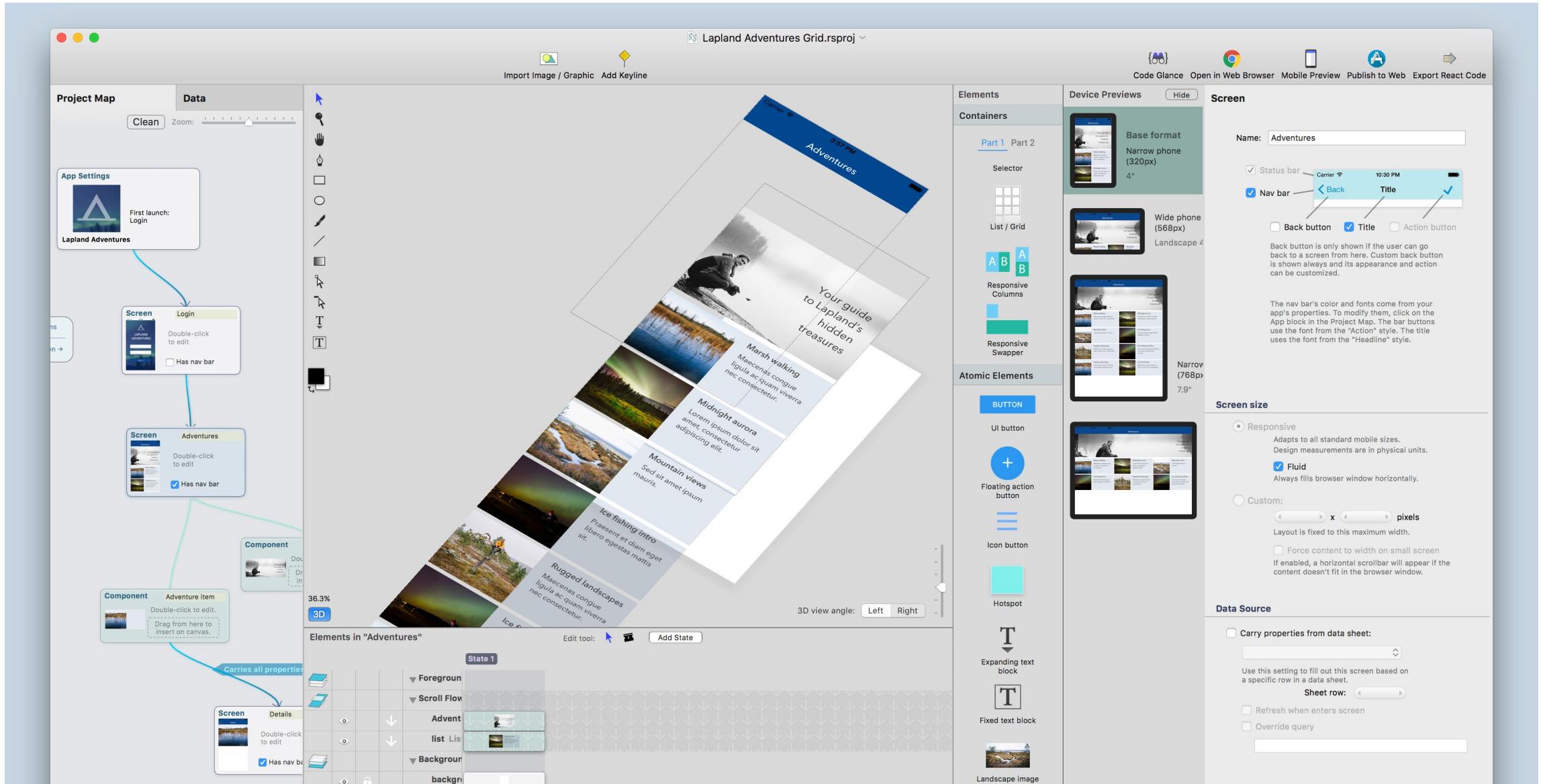


The screenshot shows the Elastic Projects interface. The top navigation bar includes the Elastic Projects logo, a dropdown for 'Elastic Projects', a user profile for 'Frank Chimero', and a notification bell. The main navigation bar shows 'Design Systems > UX Explorations' with an 'OPEN FOR FEEDBACK' button. Below this, a 'NEW BRANCH' and 'MERGE BRANCH' button are visible. The interface is divided into sections: 'CONVERSATION', 'COMMITS' (which is selected), 'FILES', and 'SETS'. The 'TODAY' section lists several commits by Frank Chimero, including adding a confirmation dialog for deleting a branch, making the interface less noisy, and post-review design tweaks. The 'YESTERDAY' and 'THIS WEEK' sections also list commits. The main content area shows a commit titled 'Add confirmation dialog for deleting branch' with a message from Frank Chimero. It includes a preview of a Sketch file titled 'Website.sketch' showing four artboards labeled 3.2. Branch - Commits, 3.3. Branch - Files, 3.1. Branch - Conversations, and 4. Layer Details. The preview shows a user interface with various components and a color palette. A 'PREVIEW & COMMIT...' button is at the bottom of the preview area.

Elastic Projects

# React Studio

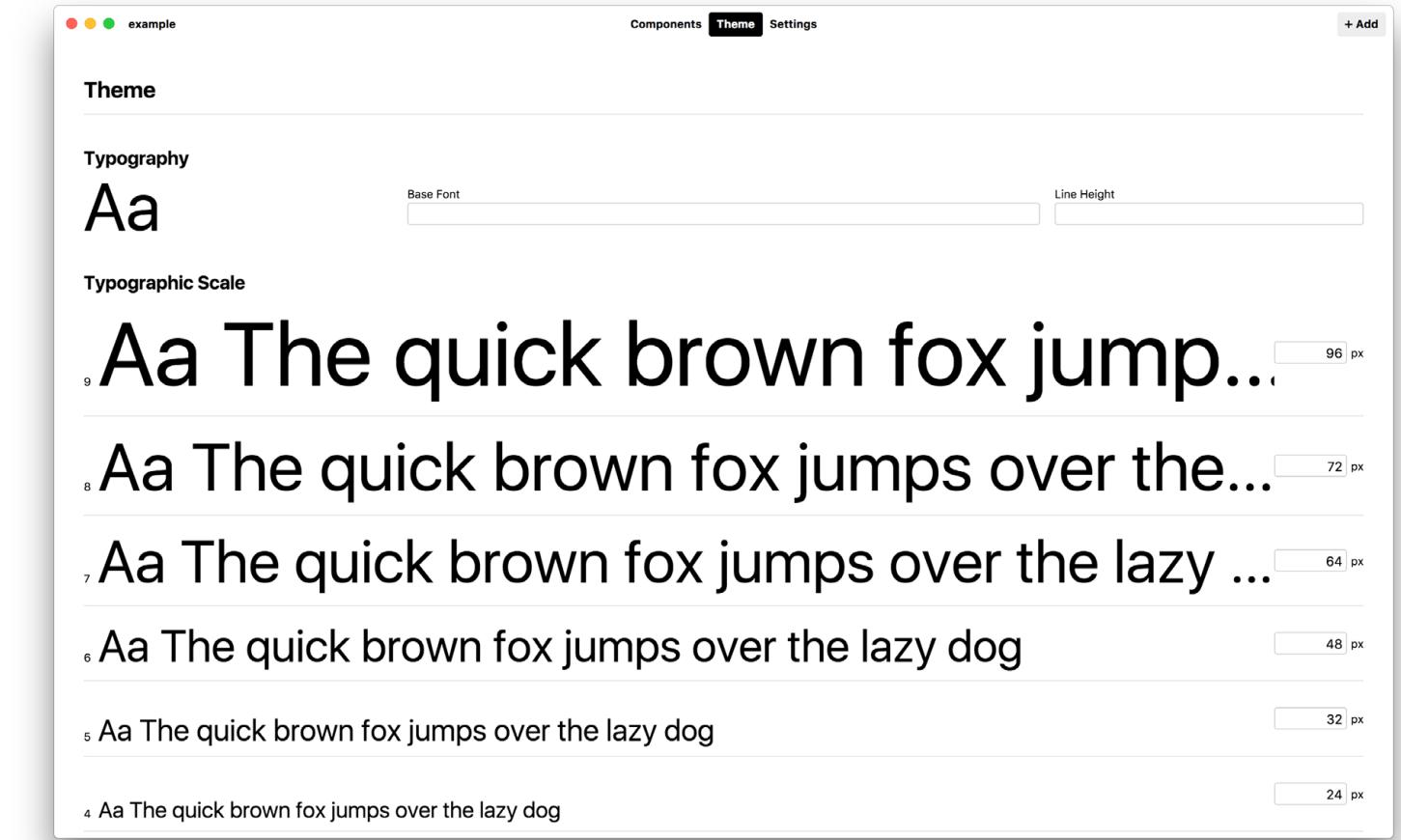
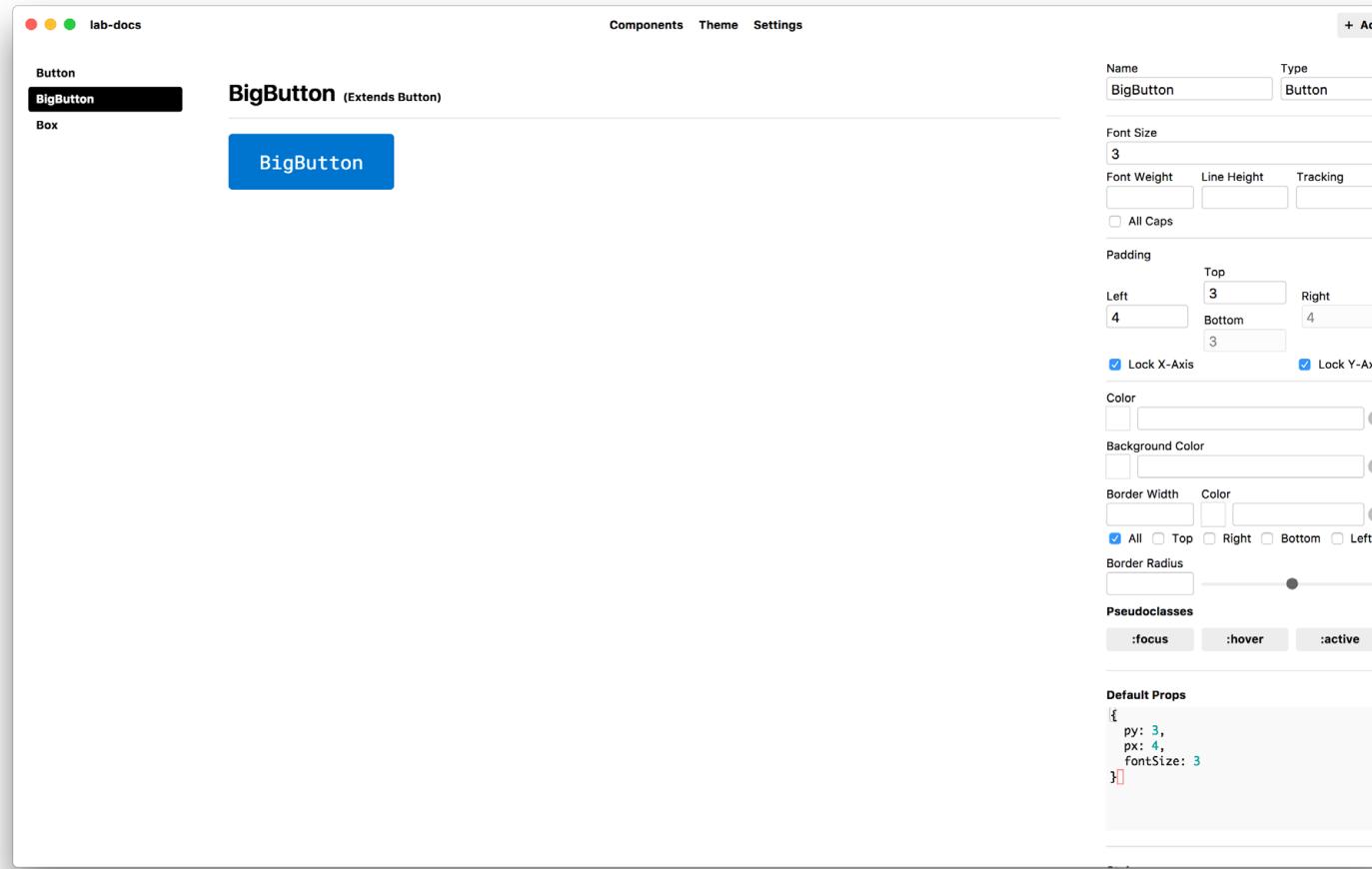
2017-Today



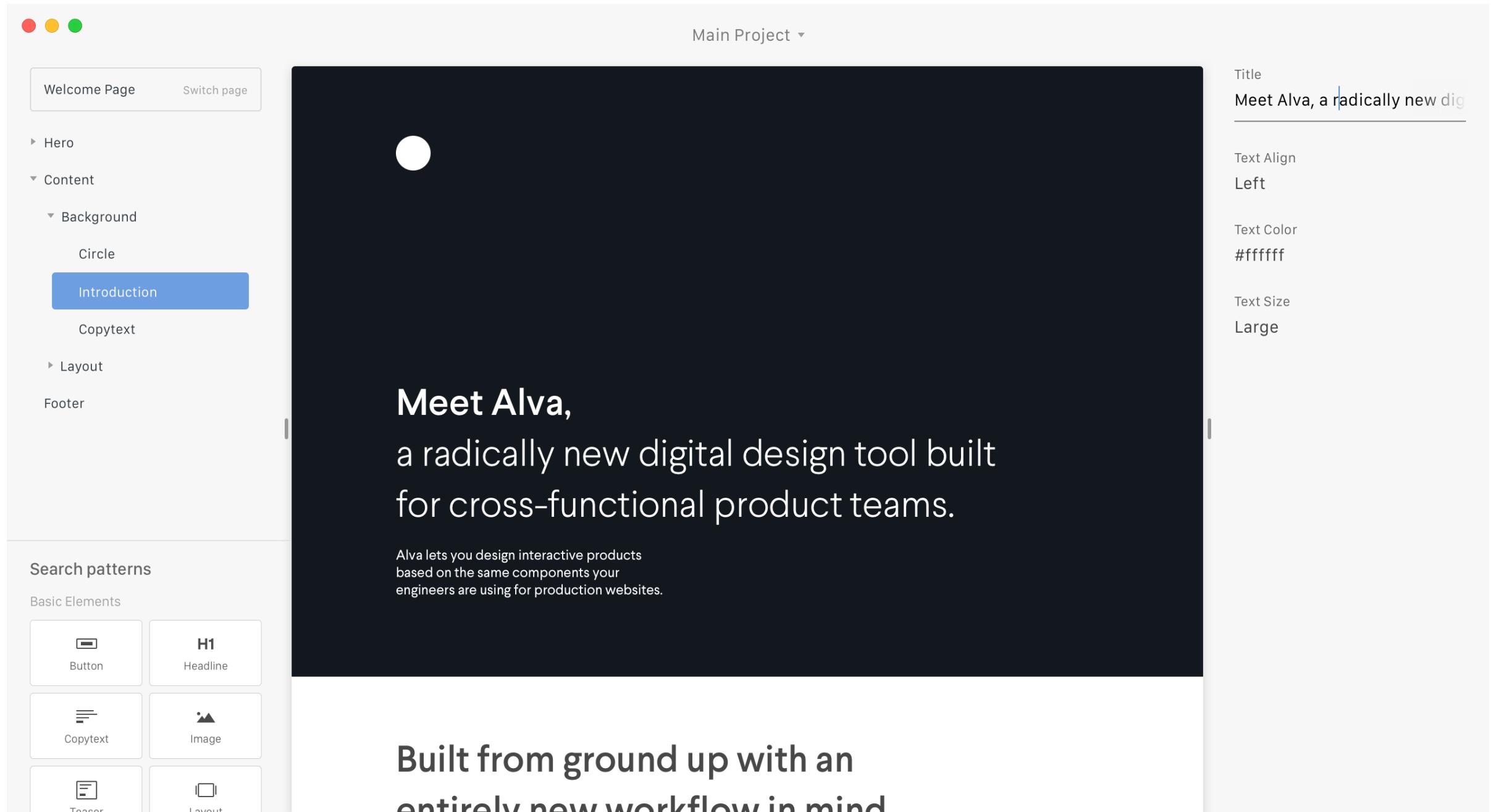
Neonto Ltd

# Compositor Lab

2017-Today



Compositor Inc



Main Project ▾

Welcome Page    Switch page

- ▶ Hero
- ▼ Content
  - ▼ Background
    - Circle
  - Introduction
  - Copytext
- ▶ Layout
- Footer

Search patterns

Basic Elements

Button	H1 Headline
Copytext	Image
Teaser	Layout

Main Project

Meet Alva, a radically new digital design tool built for cross-functional product teams.

Alva lets you design interactive products based on the same components your engineers are using for production websites.

Built from ground up with an entirely new workflow in mind

Title: Meet Alva, a radically new digital design tool built for cross-functional product teams.

Text Align: Left

Text Color: #ffffff

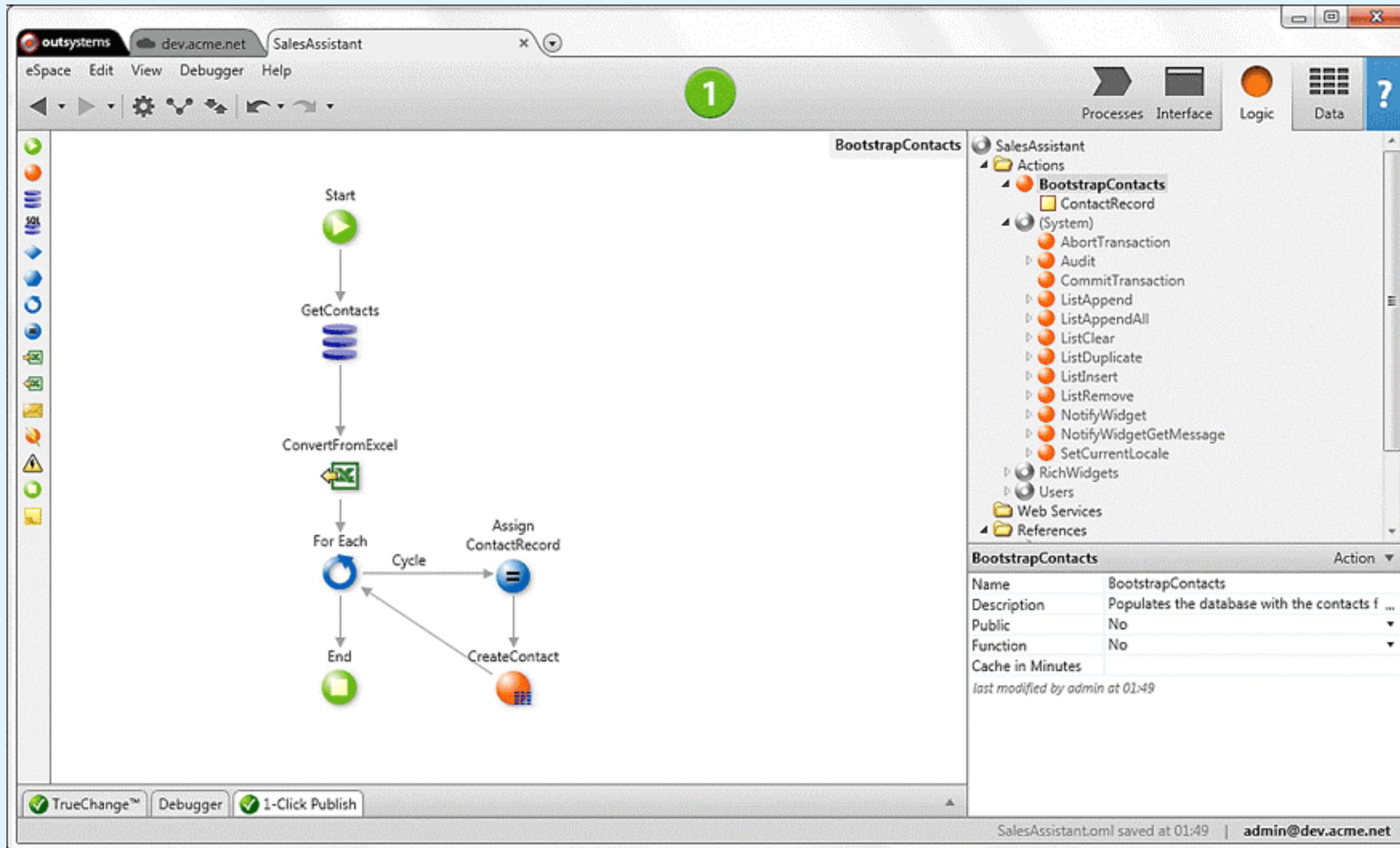
Text Size: Large

# ‘Low Code’ Environments

Enable the Development of Apps  
using modular Building-Blocks

# OutSystems Platform

2001-Today



OutSystems Inc

# App Inventor for Android

2010-Today

The image shows the MIT App Inventor 2 interface. On the left, the 'Blocks' view displays the project code for 'PresidentsQuizFinal'. The code initializes global lists for answers, questions, and pictures, and sets up an 'Initialize' event for the first screen. It then handles a 'Click' event for a 'NextButton' to move to the next question. On the right, a screenshot of the 'US Presidents' app running on an Android device shows a question about Franklin D. Roosevelt. The user has entered 'fghfdd' into an answer box, which is marked as incorrect.

```
initialize global [answerList] to [make a list "Roosevelt"]
initialize global [questionList] to [make a list "Nixon"]
initialize global [pictureList] to [make a list "roosChurch.gif" "carterChina.gif" "nixon.gif"]
initialize global [index] to [1]

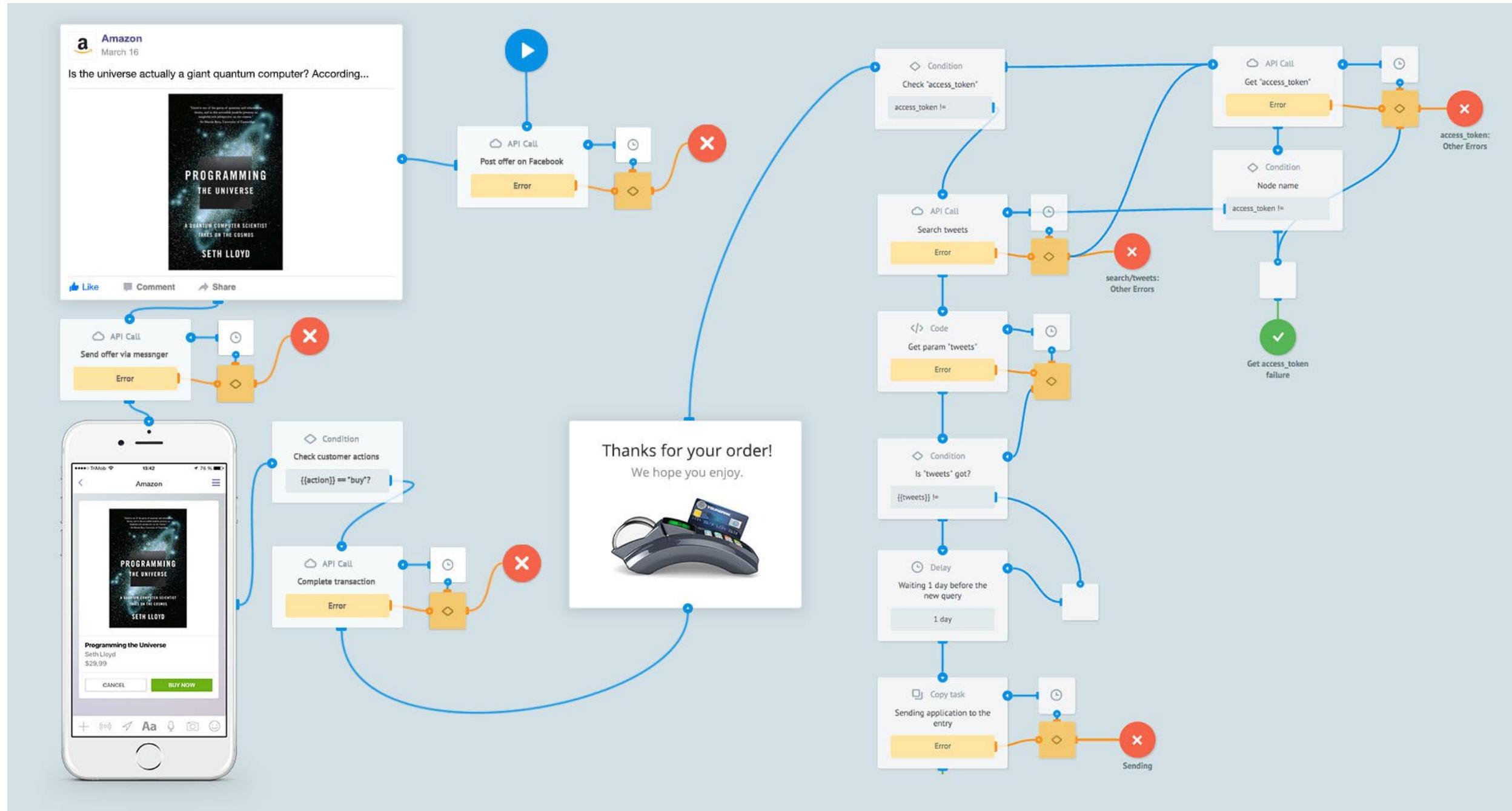
when [Screen1].Initialize
do [set [QuestionLabel].Text to [select list item list [get [global questionList]] index [1]]]

when [NextButton].Click
do [set [global index] to [get [global index] + 1]
if [get [global index] > [length of list list [get [global questionList]]] then [set [global index] to 1]
set [QuestionImage].Picture to [select list item list [get [global pictureList]] index [get [global index]]]
set [QuestionLabel].Text to [select list item list [get [global questionList]] index [get [global index]]]
set [CorrectLabel].Text to [""]
set [AnswerTextBox].Text to [""]]
```

Google and MIT Computer Science and Artificial Intelligence Lab

# Corezoid Process Modeler

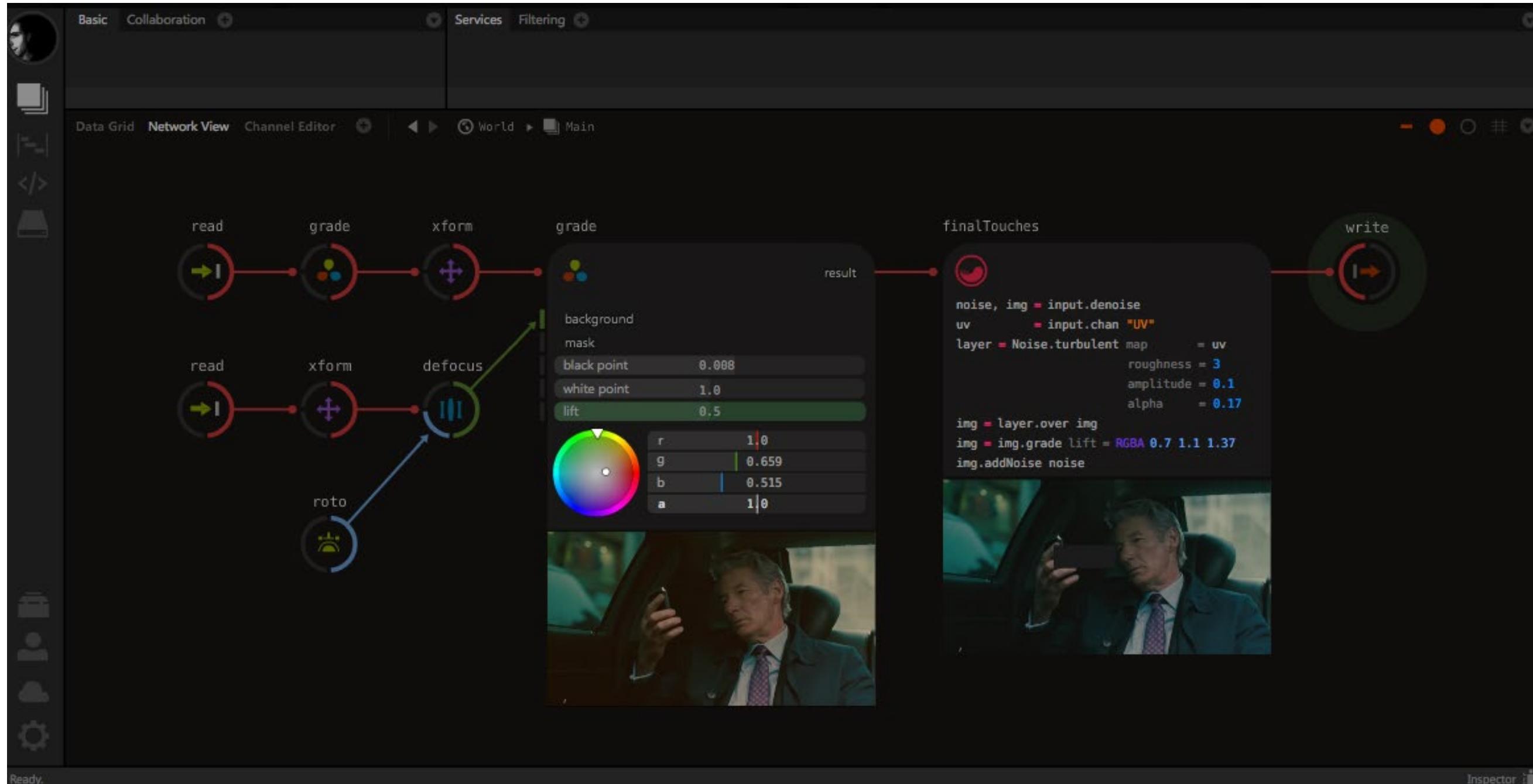
2013-Today



Corezoid.com Middleware Inc

# Luna Studio

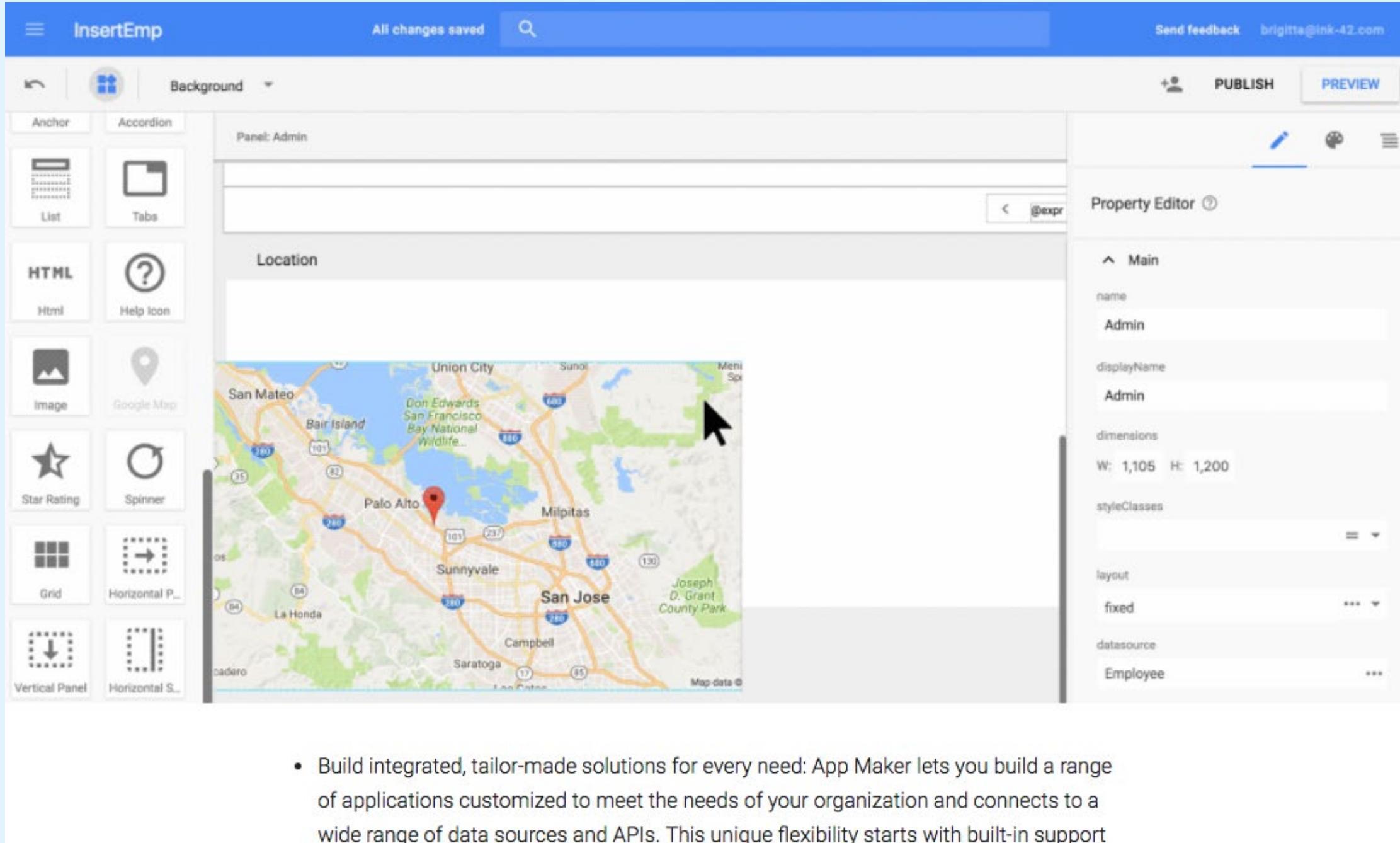
2017-Today



Wojciech Daniło, Marcin Kostrzewa

# AppMaker

2018-Today



The screenshot shows the Google AppMaker interface for a project named "InsertEmp". The main workspace displays a "Location" card with a map of the San Francisco Bay area, centered on Palo Alto. The map includes labels for Union City, San Mateo, Bair Island, Don Edwards San Francisco Bay National Wildlife, Milpitas, Sunnyvale, San Jose, Campbell, Saratoga, and Joseph D. Grant County Park. A red marker is placed on the map. The left sidebar contains a toolbar with icons for List, Tabs, HTML, Help icon, Image, Google Map, Star Rating, Spinner, Grid, Horizontal P..., Vertical Panel, and Horizontal S... components. The right sidebar shows the "Property Editor" for the "Main" component, with fields for name (Admin), displayName (Admin), dimensions (W: 1,105 H: 1,200), styleClasses, layout (fixed), and datasource (Employee).

- Build integrated, tailor-made solutions for every need: App Maker lets you build a range of applications customized to meet the needs of your organization and connects to a wide range of data sources and APIs. This unique flexibility starts with built-in support

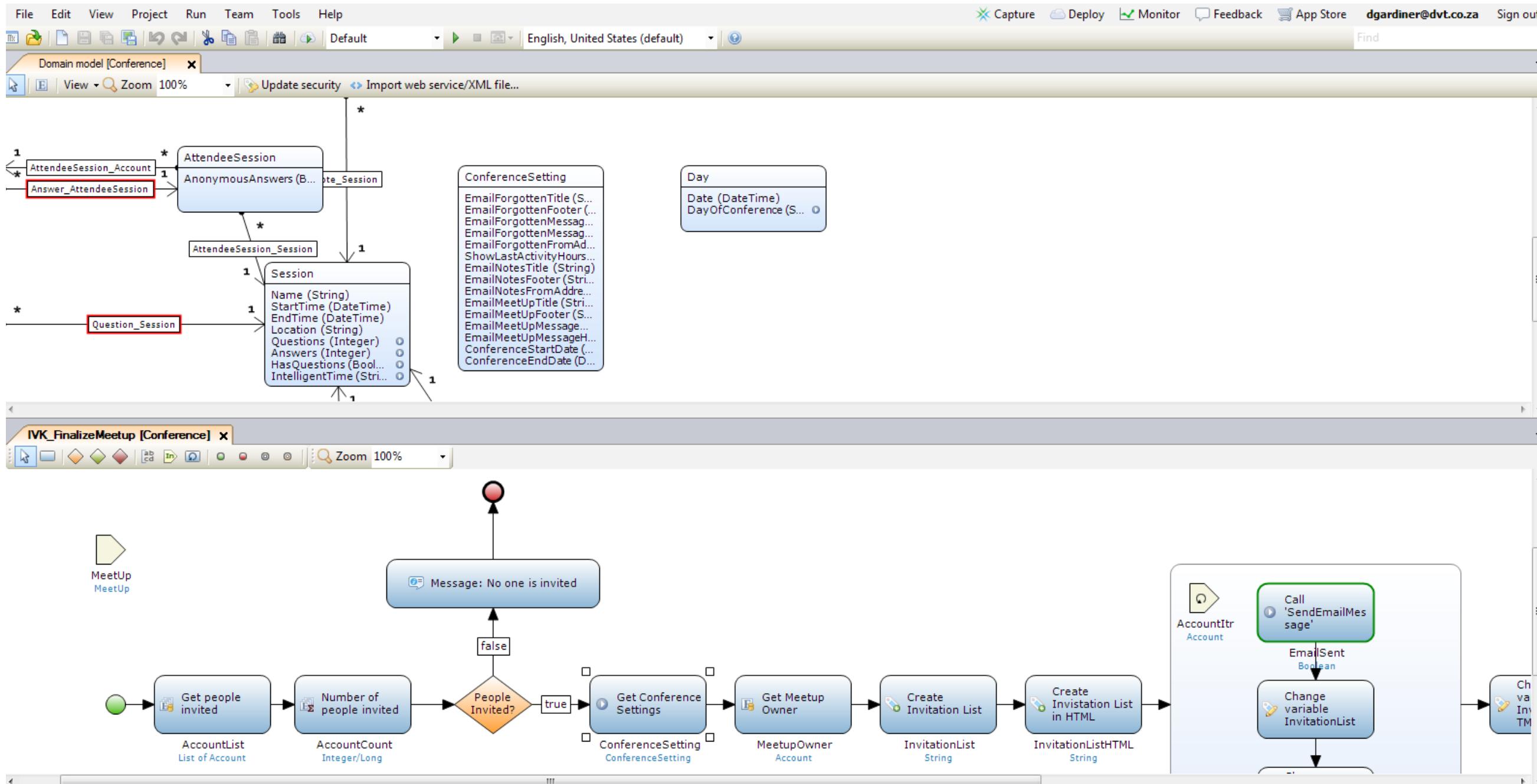
Google

# Dynamic Modeling Tools

## Support System Visualizing

# Mendix Business Modeler

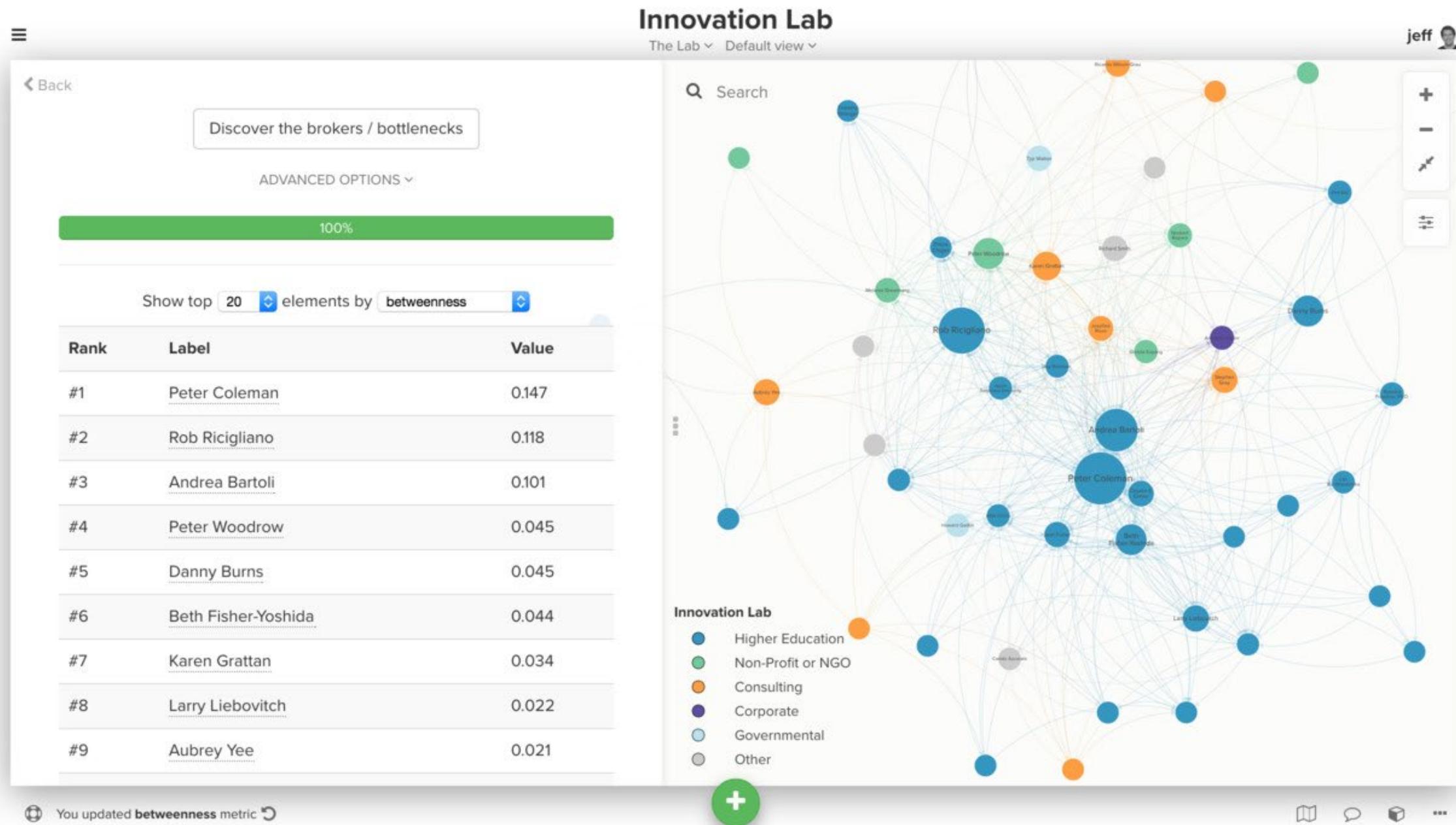
2005-Today



Mendix

# Kumu

2011-Today



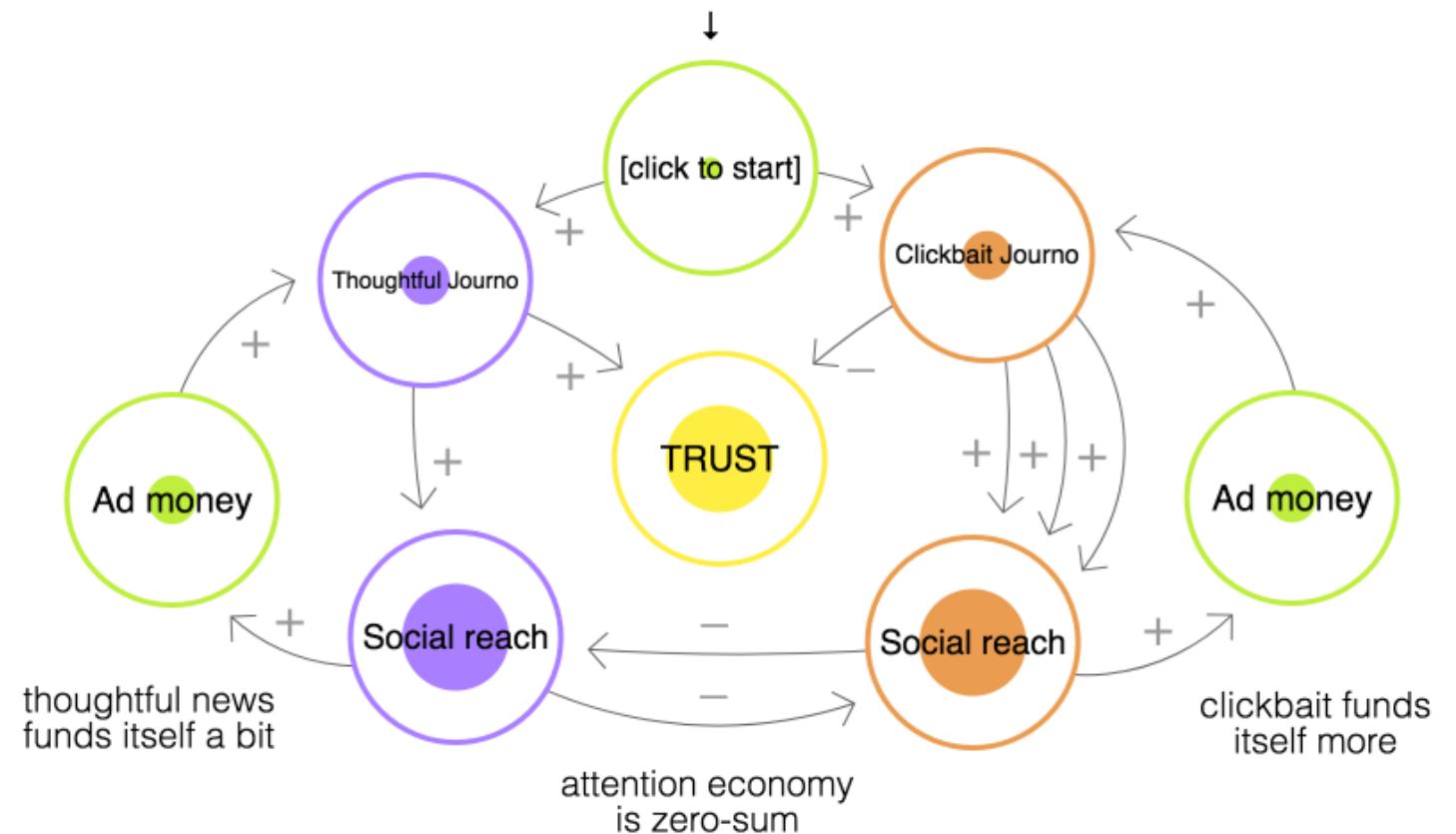
Jeff and Ryan Mohr for Kumu Inc

# Loopy

2017

Both thoughtful & clickbait journalism is supported by a positive feedback loop of ad money. But there's two differences: 1) thoughtful journo increases trust, clickbait hurts it. 2) clickbait gets more social reach... and this effect compounds.

result: ad-based journalism WILL skew towards clickbait, and WILL destroy trust.  
THE MEDIUM (of advertising) IS THE MESSAGE.



▶ Play

Nicky Case

# IDE: Environments & Experiments

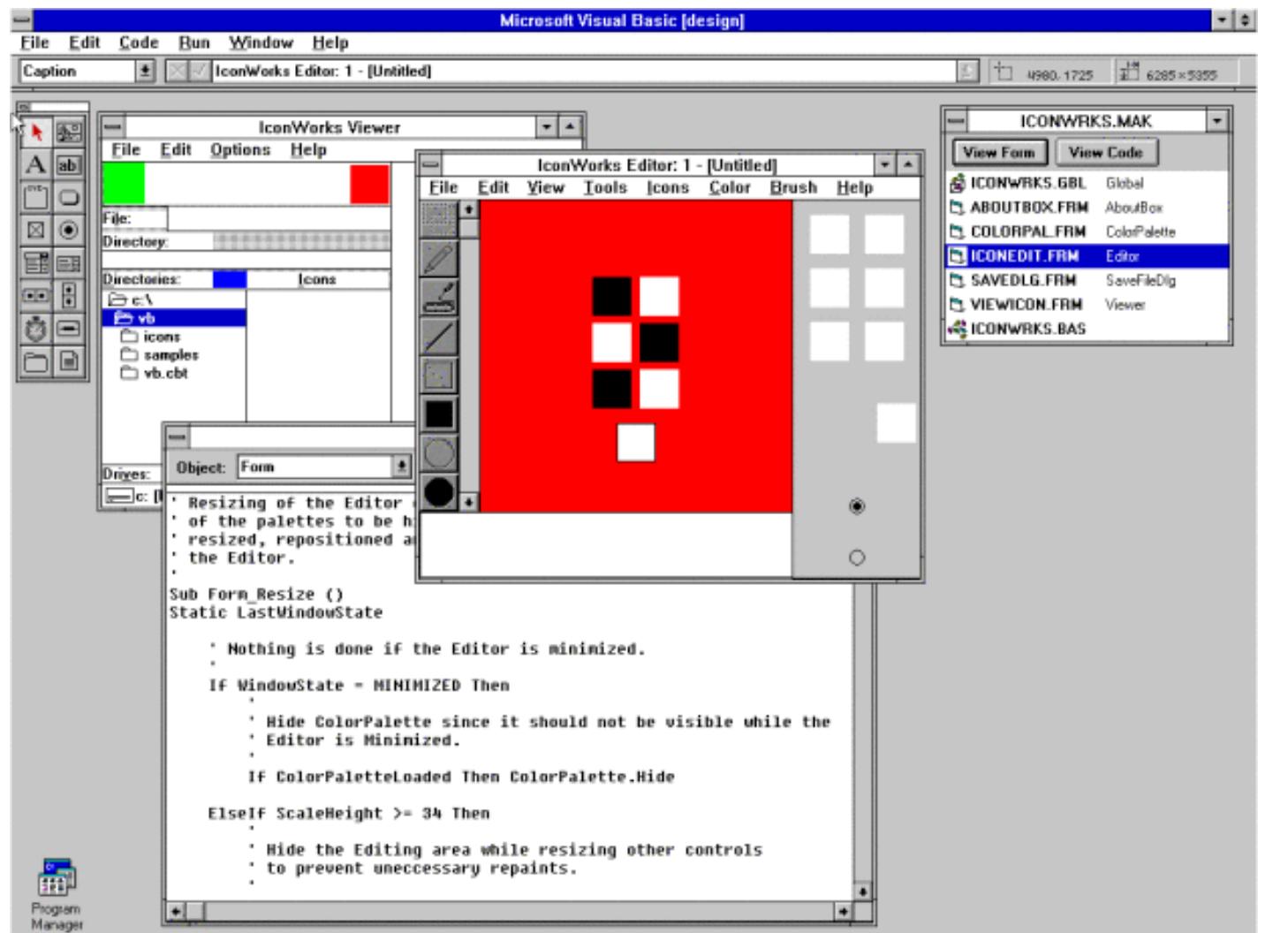
What Is & What's Next

# Traditional IDEs

Integrated Development Environments

# Visual Basic

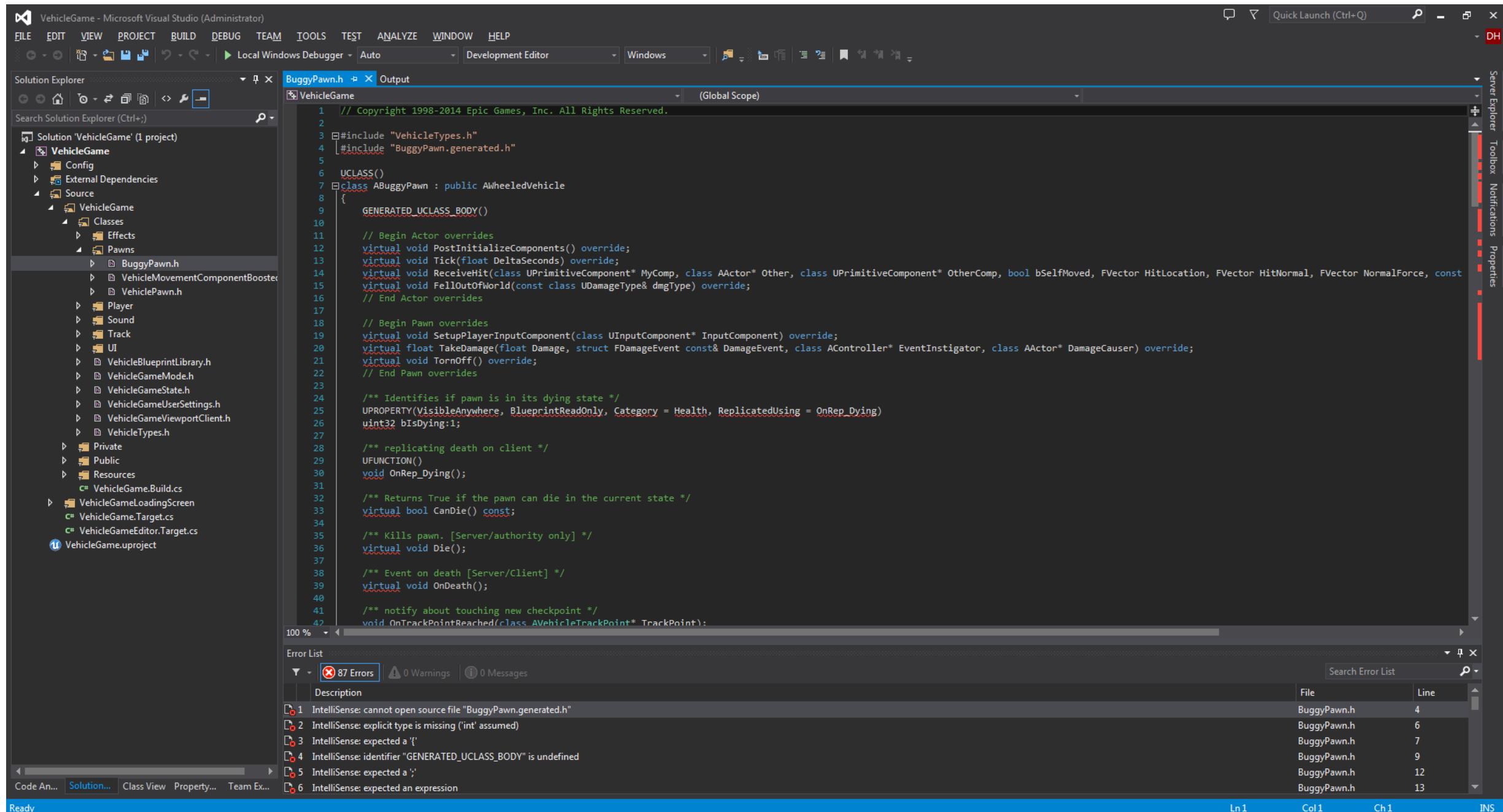
1991-1998



Cooper Software for Microsoft

# Visual Studio

1997-Today



The screenshot shows the Microsoft Visual Studio interface. The title bar reads "VehicleGame - Microsoft Visual Studio (Administrator)". The menu bar includes FILE, EDIT, VIEW, PROJECT, BUILD, DEBUG, TEAM, TOOLS, TEST, ANALYZE, WINDOW, and HELP. The toolbar has icons for file operations, search, and navigation. The "Local Windows Debugger" dropdown is set to "Auto". The "Development Editor" tab is selected, showing the code for "BuggyPawn.h" in "Global Scope". The code is a C++ header file for a vehicle pawn. The Solution Explorer on the left shows the "VehicleGame" project structure, including files like "VehicleGame.h", "VehicleMovementComponentBoosted.h", and "VehiclePawn.h". The Error List at the bottom shows 87 IntelliSense errors, all related to the "BuggyPawn.h" file, such as "cannot open source file 'BuggyPawn.generated.h'" and "expected an expression".

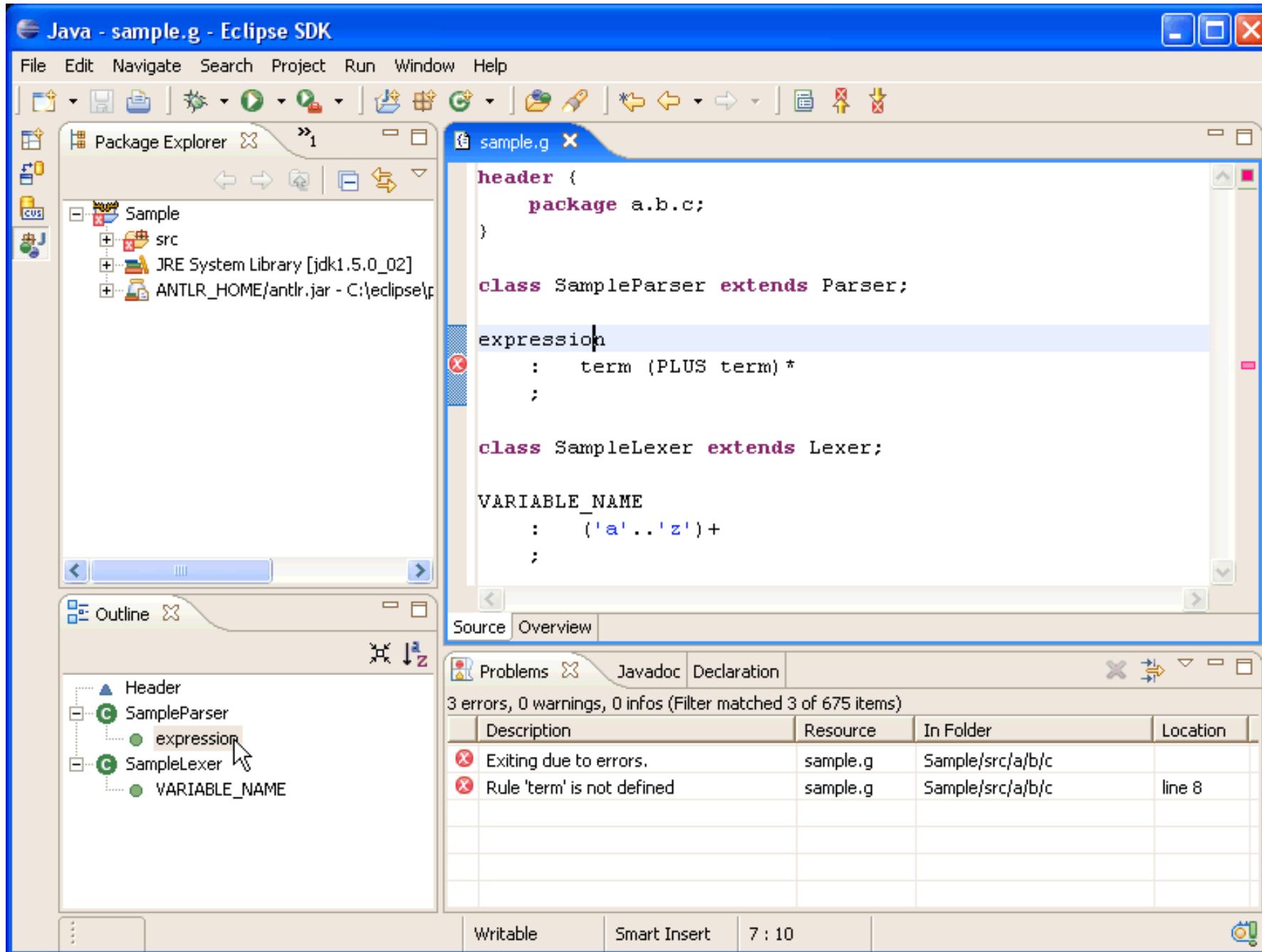
```
1 // Copyright 1998-2014 Epic Games, Inc. All Rights Reserved.
2
3 #include "VehicleTypes.h"
4 #include "BuggyPawn.generated.h"
5
6 UCLASS()
7 class ABuggyPawn : public AWheeledVehicle
8 {
9     GENERATED_UCLASS_BODY()
10
11     // Begin Actor overrides
12     virtual void PostInitializeComponents() override;
13     virtual void Tick(float DeltaSeconds) override;
14     virtual void ReceiveHit(class UPrimitiveComponent* MyComp, class AActor* Other, class UPrimitiveComponent* OtherComp, bool bSelfMoved, FVector HitLocation, FVector HitNormal, FVector NormalForce, const UDamageType& dmgType) override;
15     virtual void FellOutOfWorld(const class UDamageType& dmgType) override;
16     // End Actor overrides
17
18     // Begin Pawn overrides
19     virtual void SetupPlayerInputComponent(class UInputComponent* InputComponent) override;
20     virtual float TakeDamage(float Damage, struct FDamageEvent const& DamageEvent, class AController* EventInstigator, class AActor* DamageCauser) override;
21     virtual void TornOff() override;
22     // End Pawn overrides
23
24     /** Identifies if pawn is in its dying state */
25     UPROPERTY(VisibleAnywhere, BlueprintReadOnly, Category = Health, ReplicatedUsing = OnRep_Dying)
26     uint32 bIsDying;
27
28     /** replicating death on client */
29     UFUNCTION()
30     void OnRep_Dying();
31
32     /** Returns True if the pawn can die in the current state */
33     virtual bool CanDie() const;
34
35     /** Kills pawn. [Server/authority only] */
36     virtual void Die();
37
38     /** Event on death [Server/Client] */
39     virtual void OnDeath();
40
41     /** notify about touching new checkpoint */
42     void OnTrackPointReached(class AVehicleTrackPoint* TrackPoint);
```

Description	File	Line
1 IntelliSense: cannot open source file "BuggyPawn.generated.h"	BuggyPawn.h	4
2 IntelliSense: explicit type is missing ('int' assumed)	BuggyPawn.h	6
3 IntelliSense: expected a 'l'	BuggyPawn.h	7
4 IntelliSense: identifier "GENERATED_UCLASS_BODY" is undefined	BuggyPawn.h	9
5 IntelliSense: expected a ;	BuggyPawn.h	12
6 IntelliSense: expected an expression	BuggyPawn.h	13

Microsoft

# Eclipse

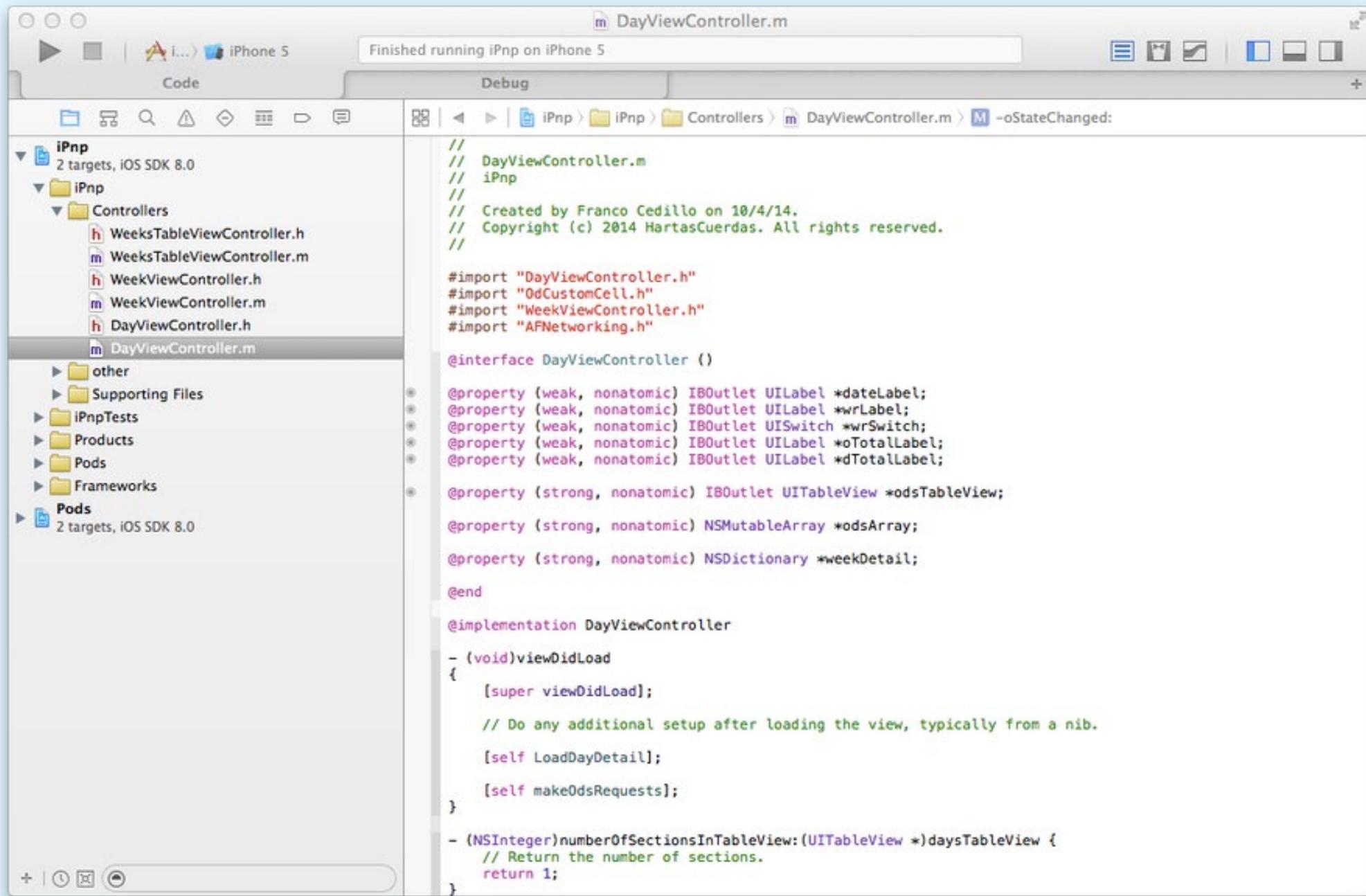
2001-Today



Eclipse Foundation

# Xcode

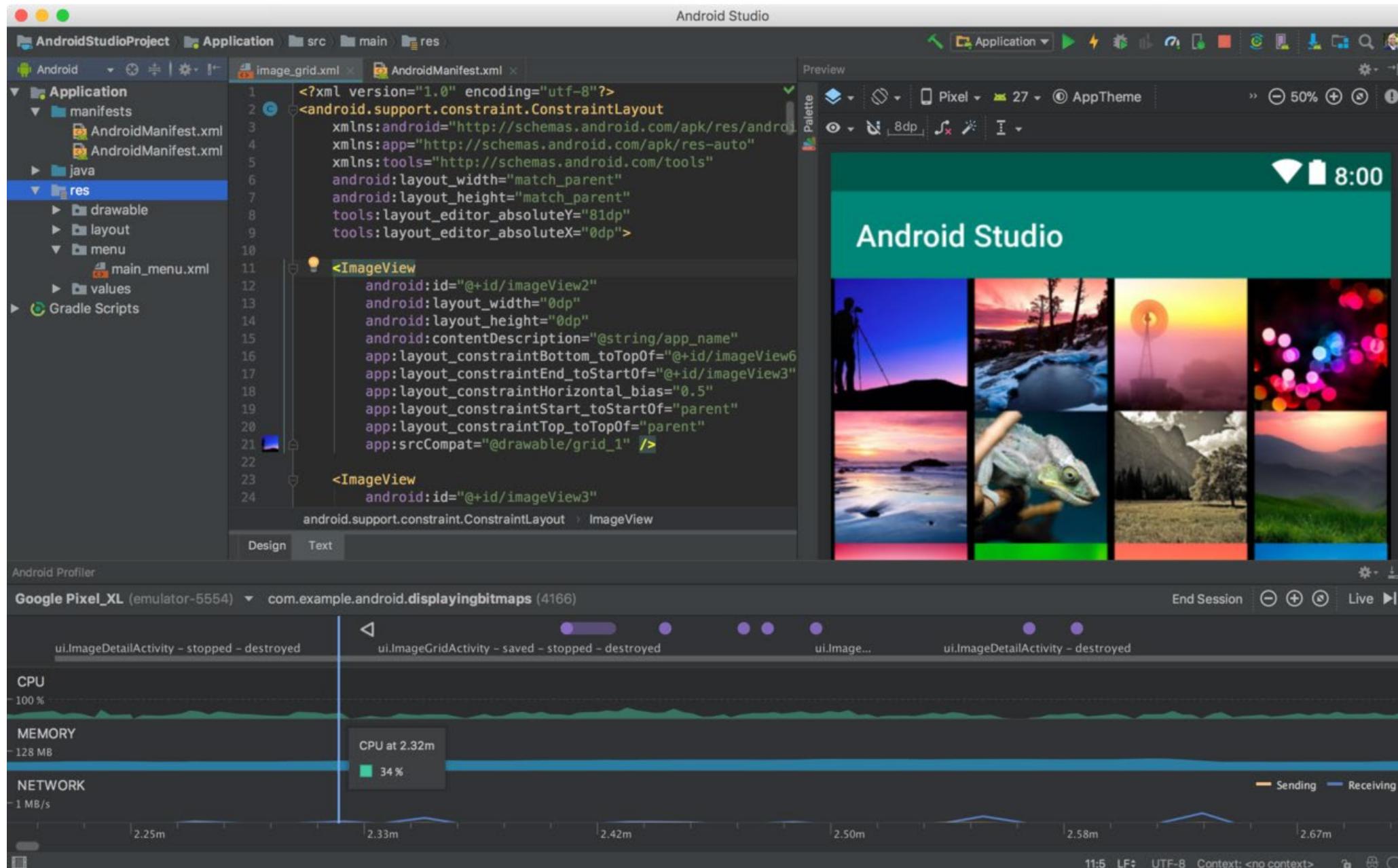
2003-Today



Apple

# Android Studio

2013-Today



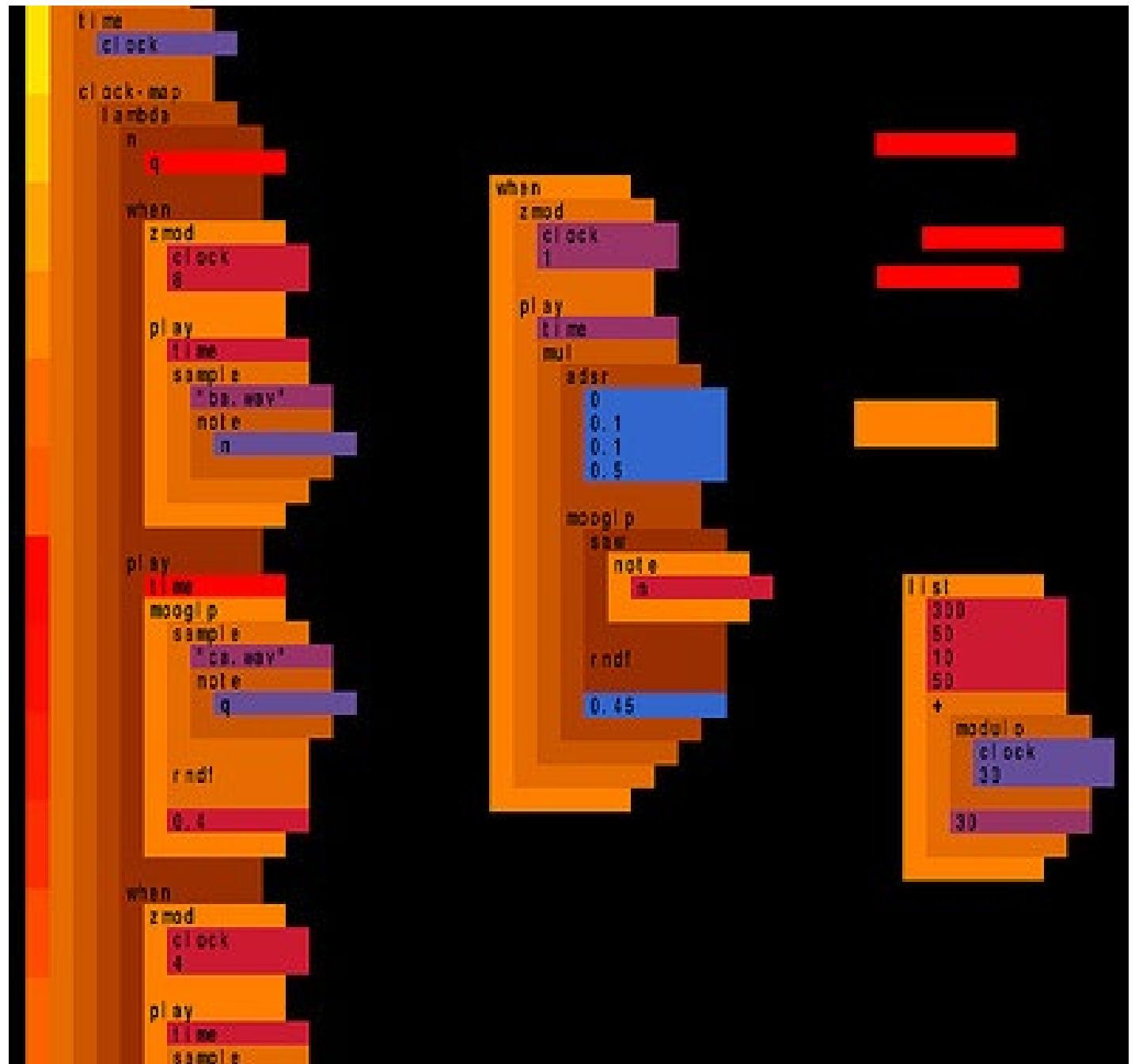
Google

# Structure Editing

At Nested Levels of Abstraction

# Scheme Bricks

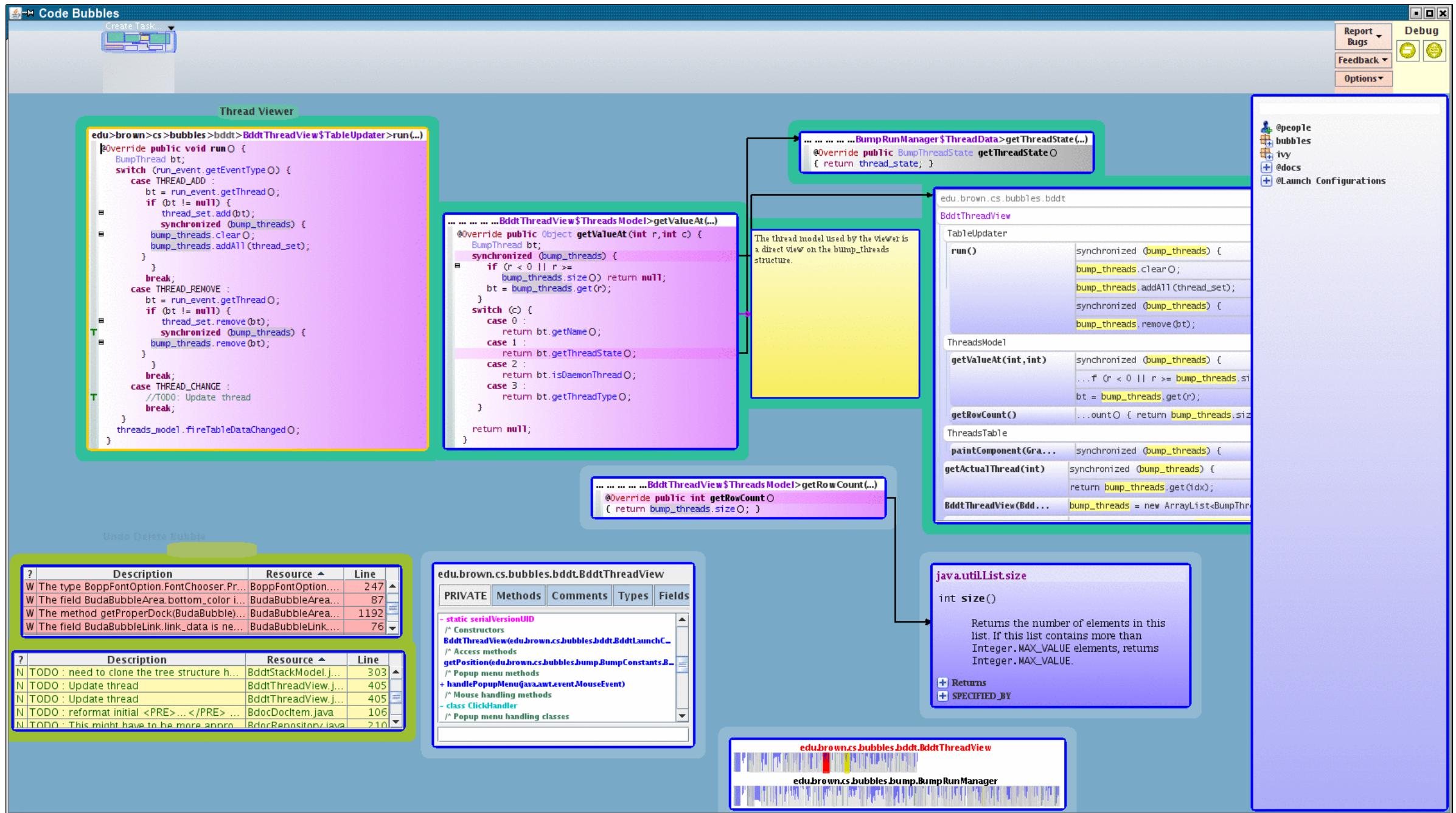
2008



Dave Griffiths

# Code Bubbles for Eclipse

2010



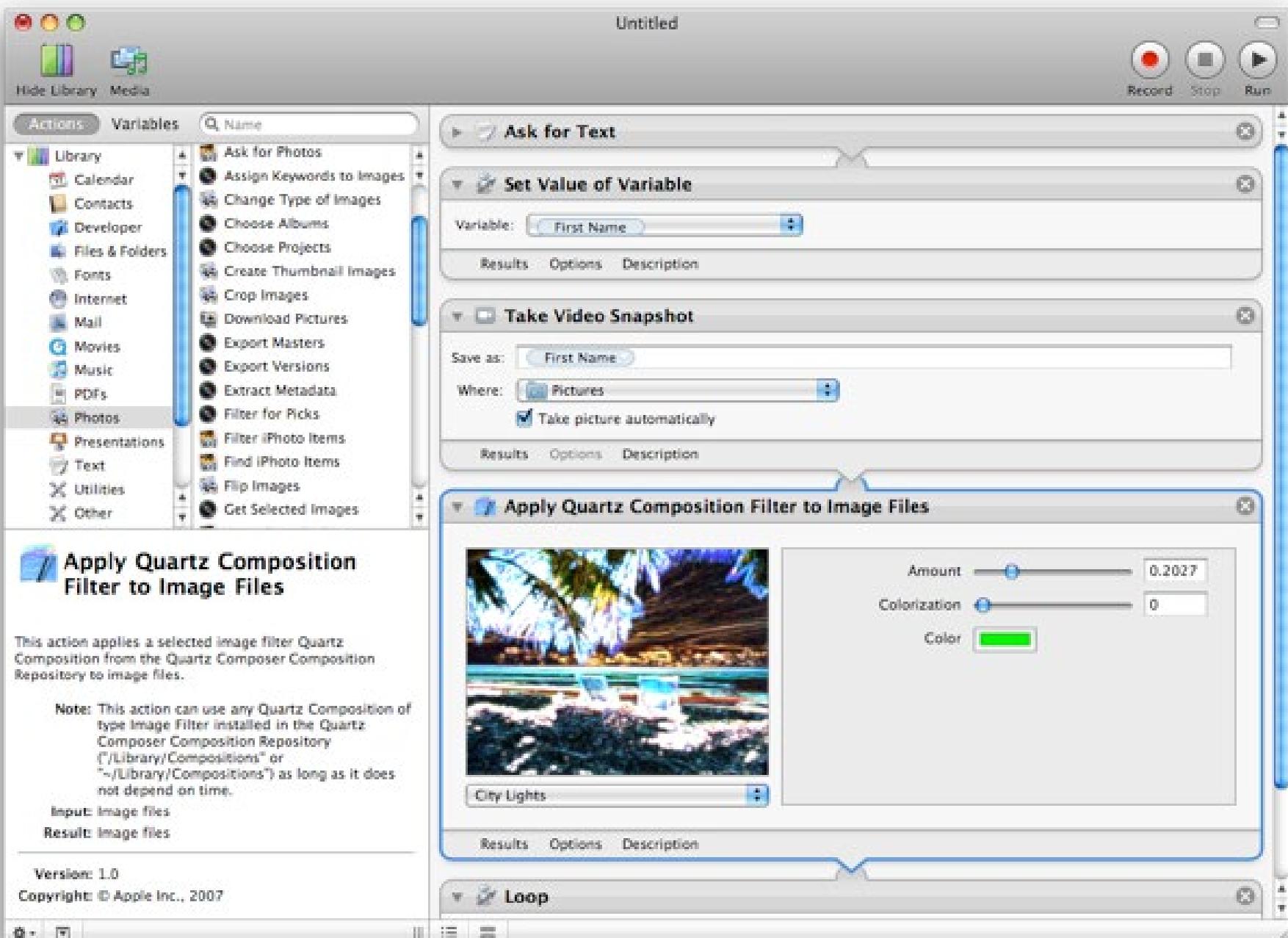
Computer Science at Brown University

# Work Flow

## Step-based Programming

# Automator

2005-Today



Apple



IFTTT Recipe Cards showing three automation examples:

- if YouTube then Blogger**  
When new video is uploaded, automatically creates new post with video embed on your blog
- if Twitter then Twitter**  
Twitter DM thanks for follow + newsletter promotion
- if RSS then Blogger**

Each card includes a set of four icons: power, trash, refresh, and edit.

Details for the first card:  
created about 3 hours ago  
never triggered

Details for the second card:  
created June 27, 2012  
last triggered 1 day ago  
triggered 48 times

Alexander Tibbets, Linden Tibbets

# WorkFlow

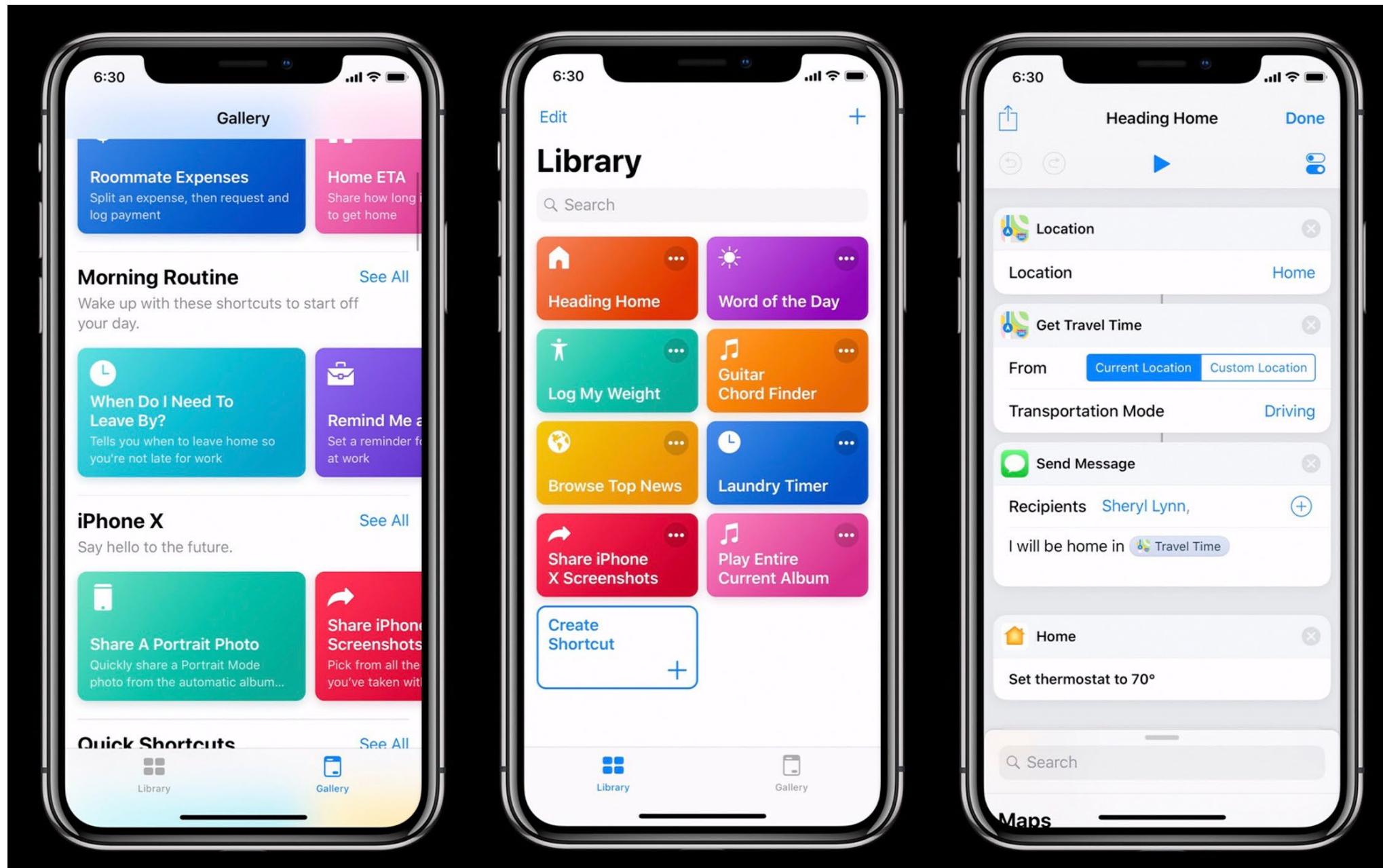
2014-



Ari Weinstein, Conrad Kramer, Ayaka Nonaka and Nick Frey for DeskConnect, Inc (Acquired by Apple)

# Siri Shortcuts

2018



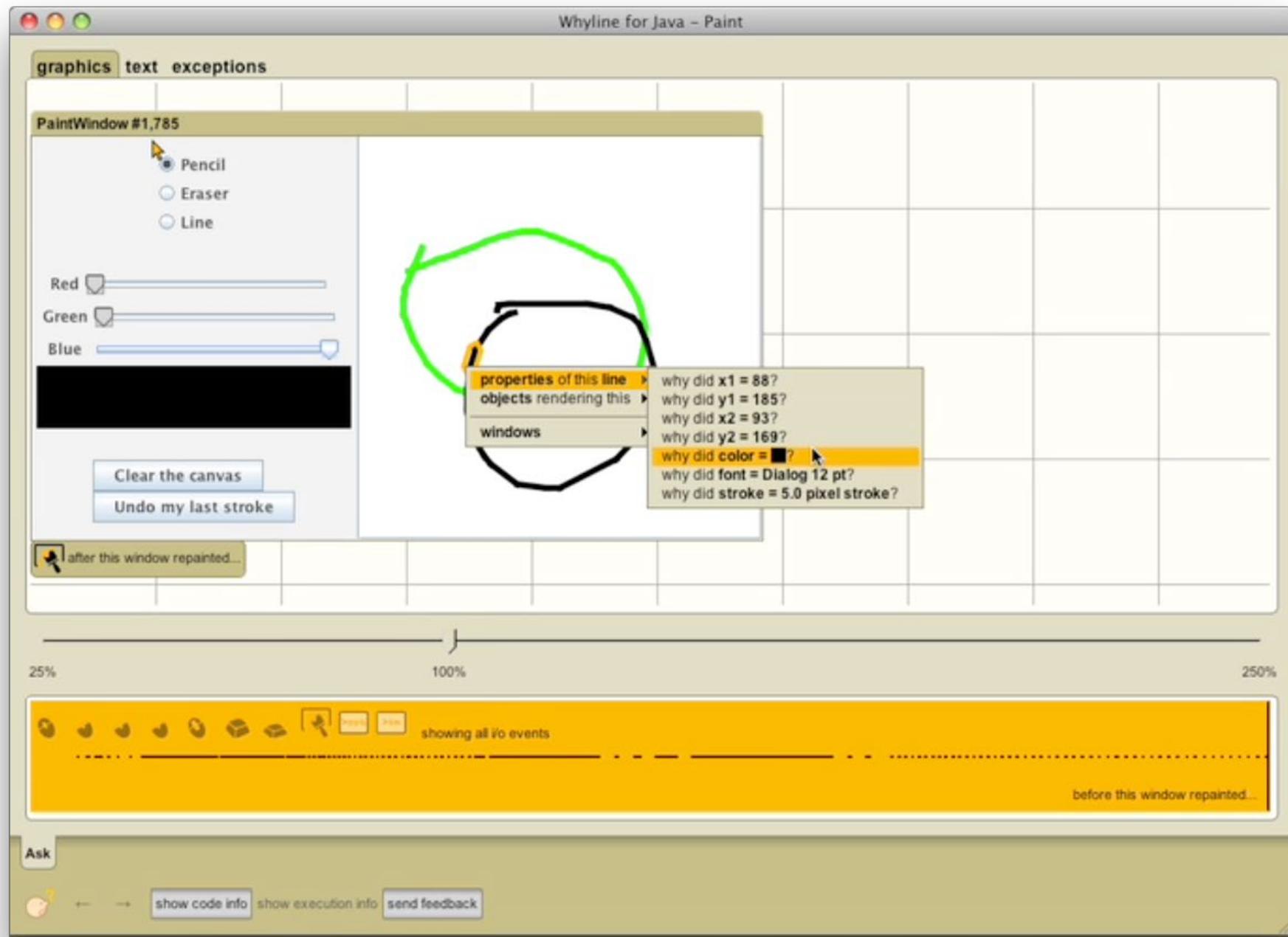
Apple

# Visible State Editors

Introduce Transparency to  
Traditional Programming

# Whyline

2008



Andrew Ko at Carnegie Mellon University (HCI Institute)

# Learnable Programming (Essay)

2012

step 24

```
var i = 0;
while (i < 20) {
  var scaleFactor = 1 + (20 - i)/20;
  resetMatrix();
  scale(scaleFactor);
  rotate(i * 6);
  fill(i * 30, i * 18, 0);
  triangle(0,0, 100,-20, 95,40);
  i += 1;           Set the "i" variable to 3.
}
```

0 10 20 30 40 50 60

0	t	t	t	t	t	t	t	t
2	1.95	1.9	1.85	1.8	1.75	1.7	1.65	
*	*	*	*	*	*	*	*	*
2	1.95	1.9	1.85	1.8	1.75	1.7	1.65	
↶	↶	↶	↶	↶	↶	↶	↶	
●	●	●	●	●	●	●	●	
◀	◀	◀	◀	◀	◀	◀	◀	
1	2	3	4	5	6	7		

fill(0,0,0);
drawRect(80,80, 40,25);
drawRect(60,20, 80,40);
drawLine(20,20, 180,160);
drawTriangle(80,60, 80,20, 140,60);
drawRect(60,20, 80,40);
drawEllipse(100,40, 80,40);
drawBezier(60,20, 180,20, 140,60, 60,60);
drawText("Hello!", 60,40);

fill(161, 219, 114);
for (var x = 40; x < 150; x += 50) {
 rect(x, 33, 20, 10);
 rect(x, 45, 20, 15);
 rect(x, 62, 47, 25); height of the rectangle
}

var x = 0, y = 50, dy = 0;
function draw () {
 x += 3;
 y += dy;
 if (y > 185) {
 dy = -dy;
 ellipse(x, 190, 36, 25);
 }
 else {
 dy = dy \* 1.07 + ↘;
 ellipse(x, y, 30, 30);
 }
}

frame 10

id

Bret Victor

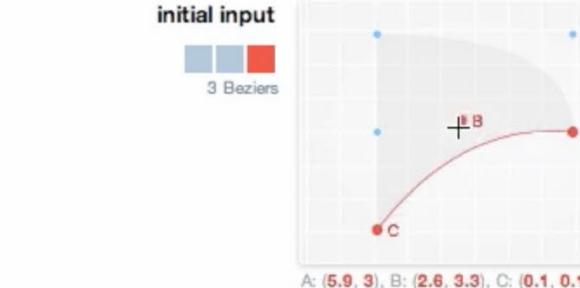
# Media for Thinking The Unthinkable (Nile Demo) 2013

**initial input**



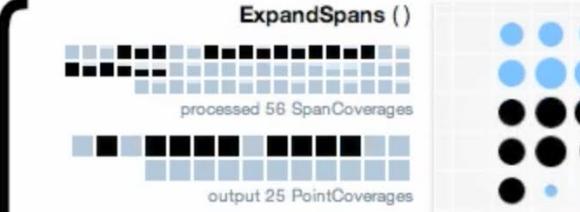
5 Points

**initial input**



3 Beziers

**ExpandSpans ()**

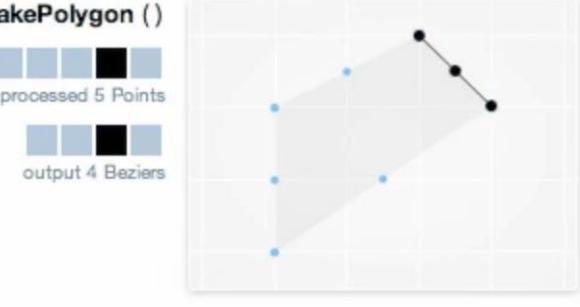


processed 56 SpanCovers



output 25 PointCovers

**MakePolygon ()**



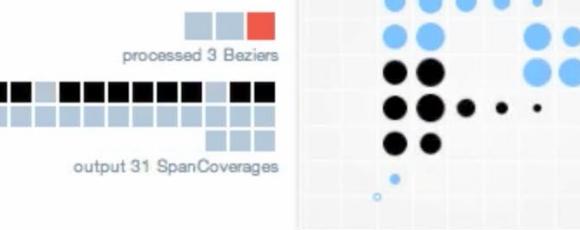
processed 5 Points

output 4 Beziers

**MakePolygon ()**

```
MakePolygon () : Point >> Bezier
p:Point = 0
first = true
& p'
  first' = false
  if !first
    >> (p, p ~ p', p')
```

**Rasterize ()**

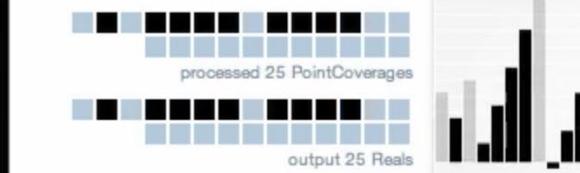


processed 3 Beziers



output 31 SpanCovers

**ProjectLinearGradient ()**

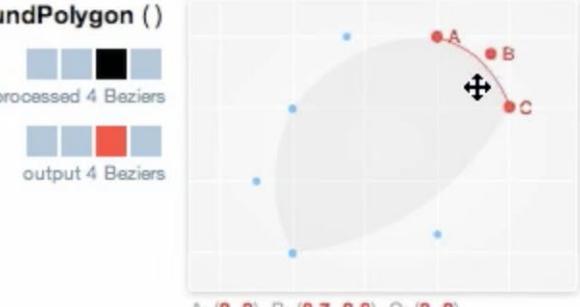


processed 25 PointCovers



output 25 Reals

**RoundPolygon ()**



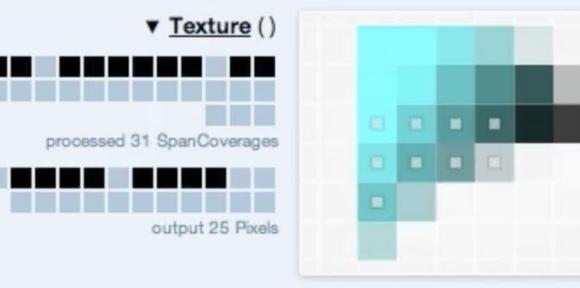
processed 4 Beziers

output 4 Beziers

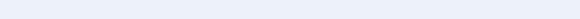
**RoundPolygon ()**

```
RoundPolygon () : Bezier >> Bezier
& (A, B, C)
  n = (A ⊥ C) / 4
  >> (A, B + n, C)
```

**Texture ()**

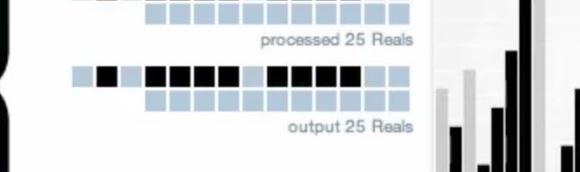


processed 31 SpanCovers



output 25 Pixels

**PadGradient ()**

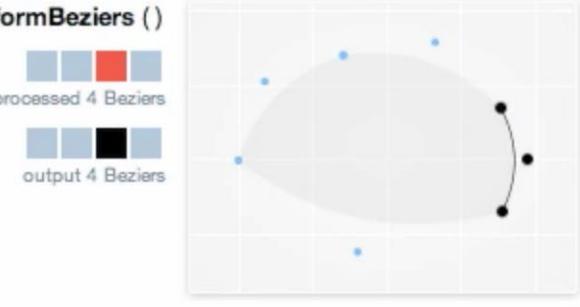


processed 25 Reals



output 25 Reals

**TransformBeziers ()**



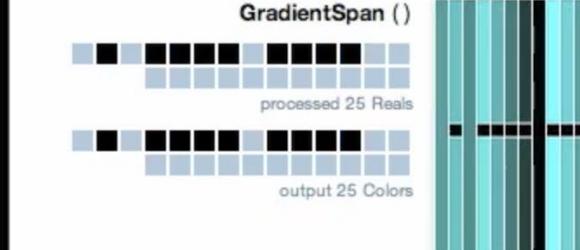
processed 4 Beziers

output 4 Beziers

**TransformBeziers (M:Matrix) : Bezier >> Bezier**

```
& (A, B, C)
  >> (MA, MB, MC)
```

**GradientSpan ()**



processed 25 Reals



output 25 Colors

**ExpandSpans ()**

```
& (x, y, c,
  if c >
    >>
    <<
```

**ProjectLinearGradient ()**

```
v = B - A
Δs = v / (B - A)
s00 = A + Δs
& (P, _)
  >> P + s00 * Δs
```

**PadGradient ()**

```
& s
  >> 0 ▷
```

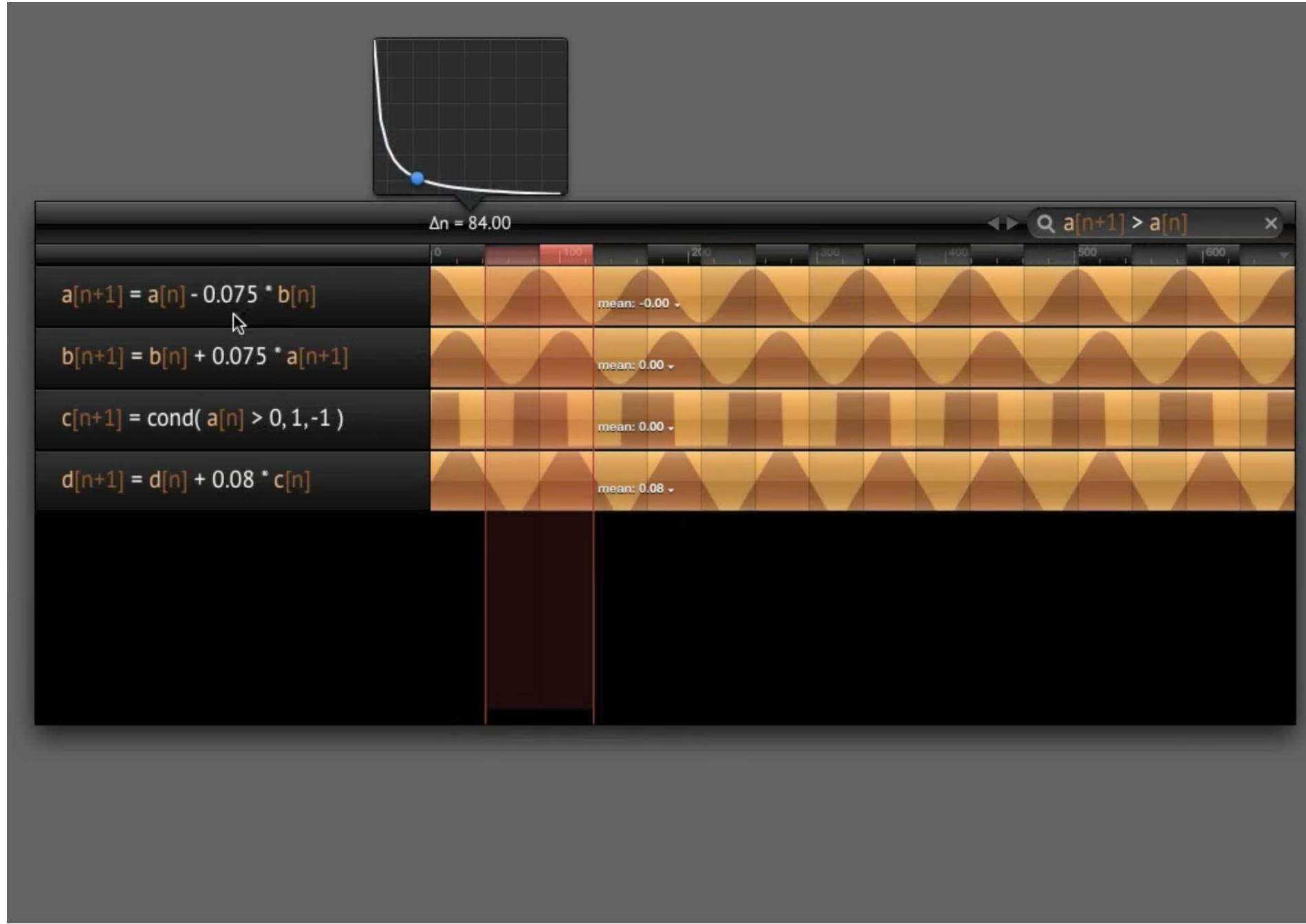
**GradientSpan (A, B, C)**

```
& s
  >> sA + sB + sC
```

Bret Victor

# Media for Thinking The Unthinkable (MSP)

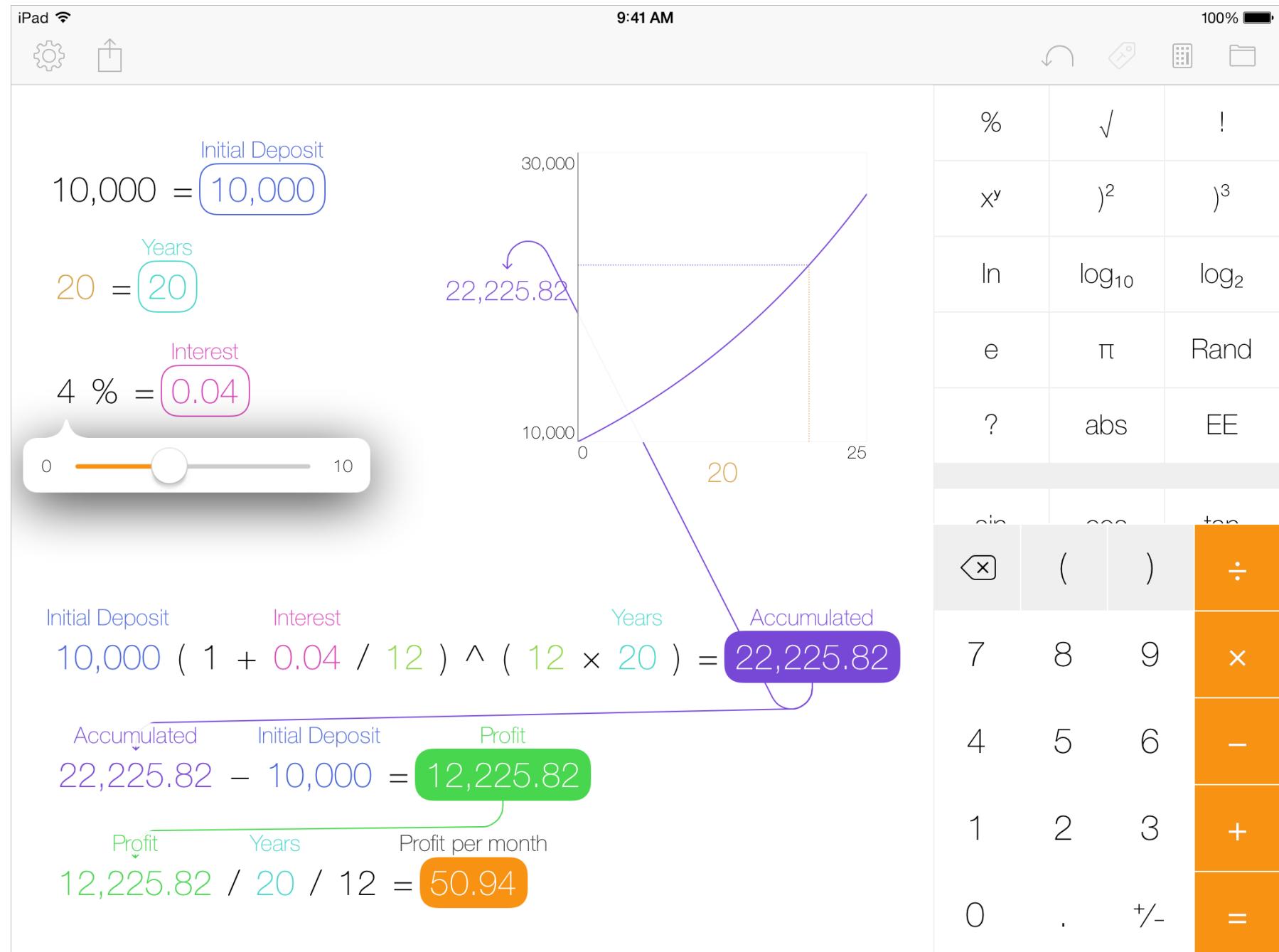
2013



Bret Victor

# Tydlig

2013



Andreas Karlsson for Tydlig Software AB

# Xcode Playgrounds

2015-Today

Running Newton's Cradle

## Newton's Cradle and UIKit Dynamics

This playground uses [UIKit Dynamics](#) to create a [Newton's Cradle](#). Commonly seen on desks around the world, Newton's Cradle is a device that illustrates conservation of momentum and energy.

Let's create an instance of our UIKit Dynamics based Newton's Cradle. Try adding more colors to the array to increase the number of balls in the device.

```
let newtonsCradle = NewtonsCradle(colors: [UIColor.pinkColor(), UIColor.blueColor(), UIColor.orangeColor(), UIColor.greenColor()])
```

Size and spacing

Try changing the size and spacing of the balls and see how that changes the device. What happens if you make ballPadding a negative number?

```
newtonsCradle.ballSize = CGSizeMake(60, height: 60)
newtonsCradle.ballPadding = 2.0
```

Behavior

Adjust elasticity and resistance to change how the balls react to each other.

```
newtonsCradle.itemBehavior.elasticity = 1.0
newtonsCradle.itemBehavior.resistance = 0.2
```

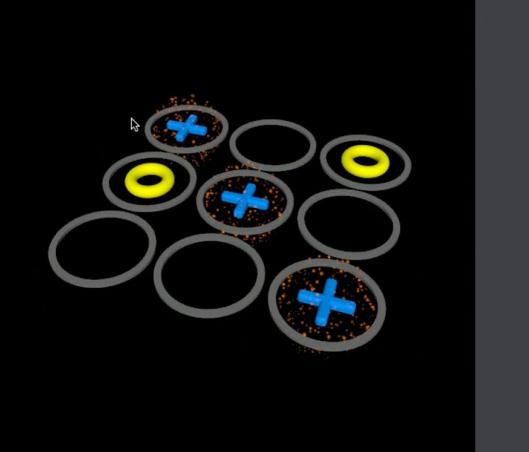
Shape and rotation



Ready to continue TicTacToe

## TicTacToe

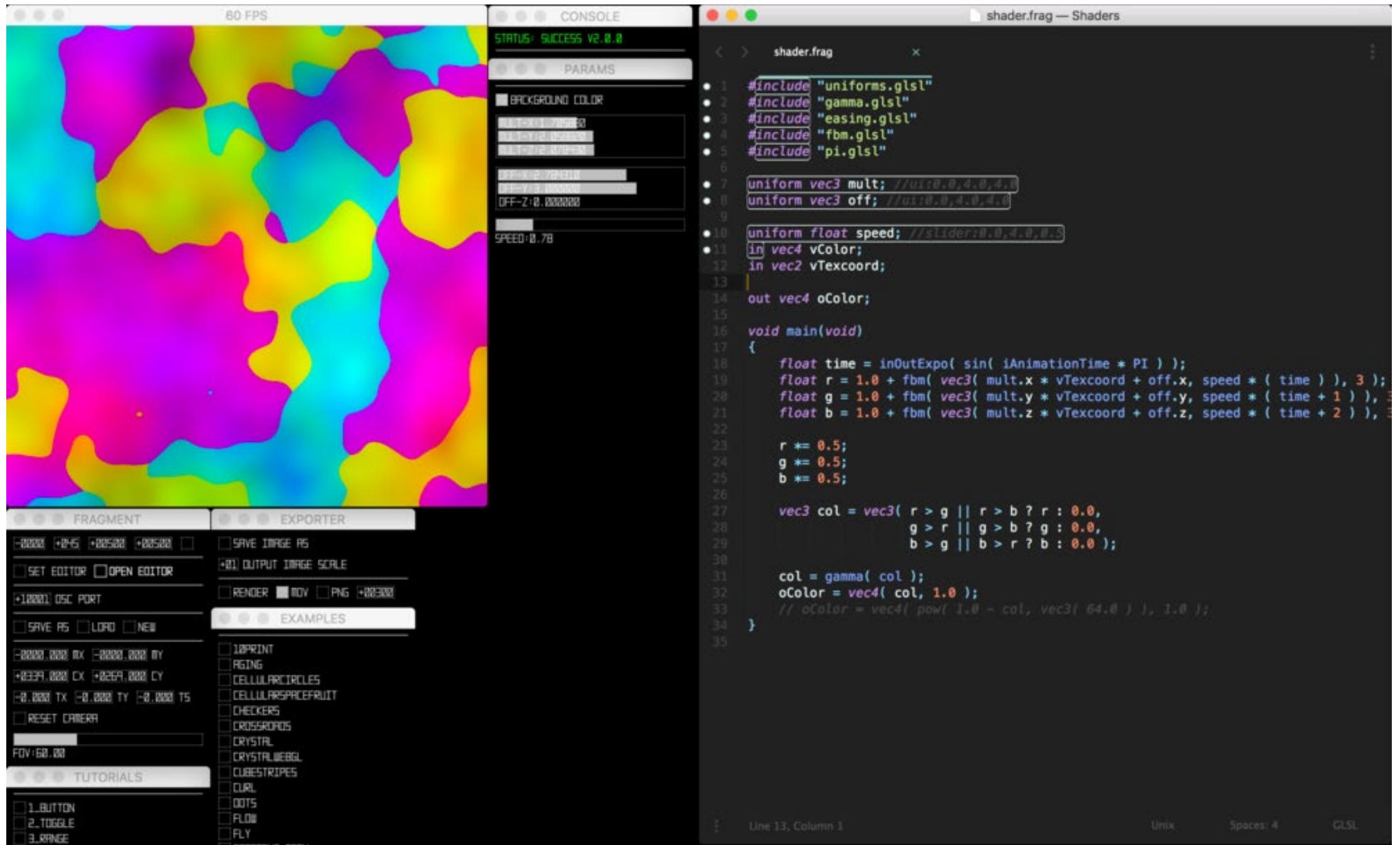
```
1 import PlaygroundSupport
2 import Foundation
3
4 let gameView = BoardView(frame: CGRect(x: 0, y: 0,
5                                         width: 600, height: 600))
6
7 PlaygroundPage.current.liveView = gameView
8
9 // Would you like to play a game of TicTacToe
10 var b = Board(view: gameView)
11
12 b.move(.top, .left)
13 b.autoMove()
14
15 b.move(.middle, .middle)
16 b.autoMove()
17
18 b.move(.bottom, .right)
19 gameView.newWinEffect()
```



Apple

# Fragment

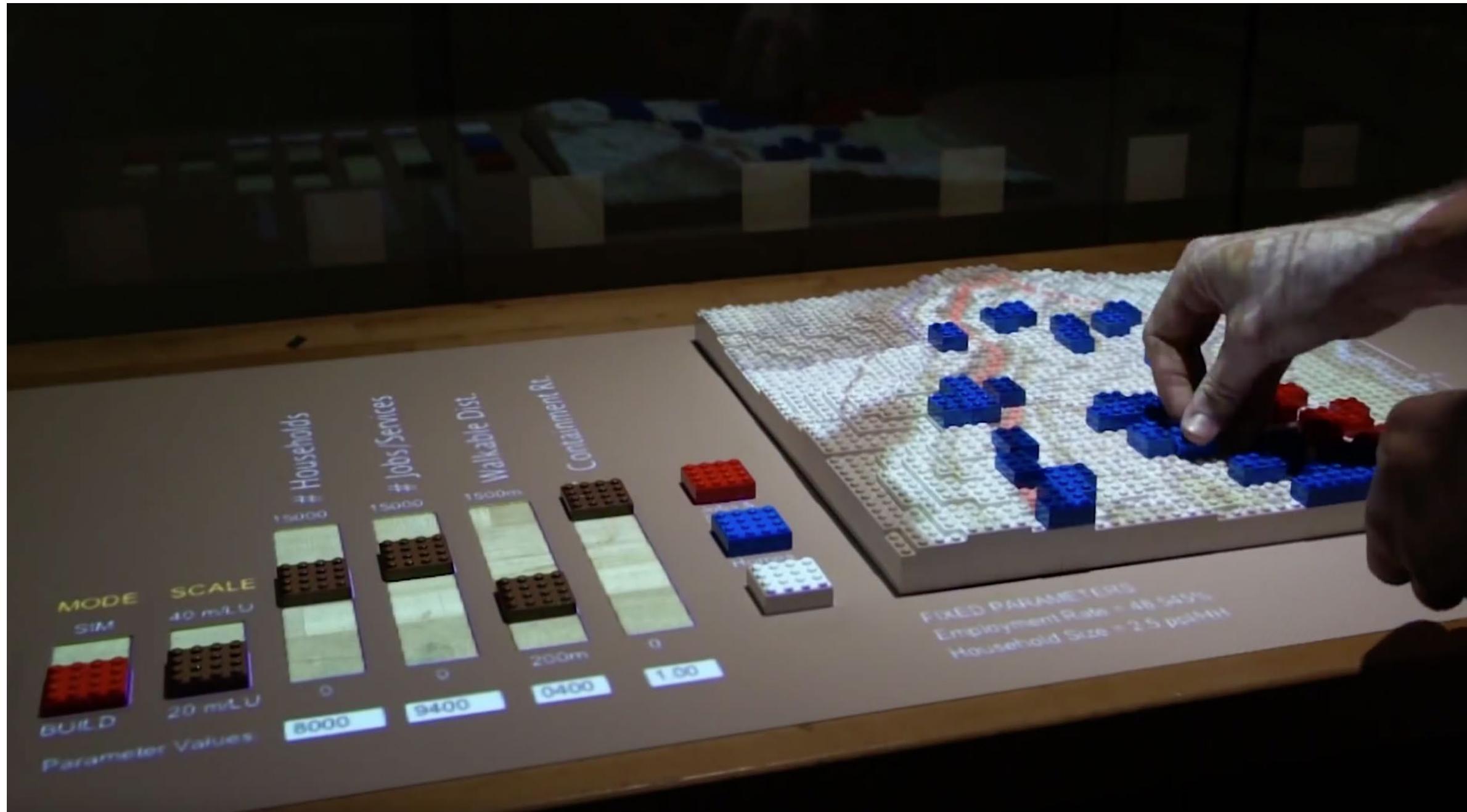
2017



Reza Ali

# CityScope

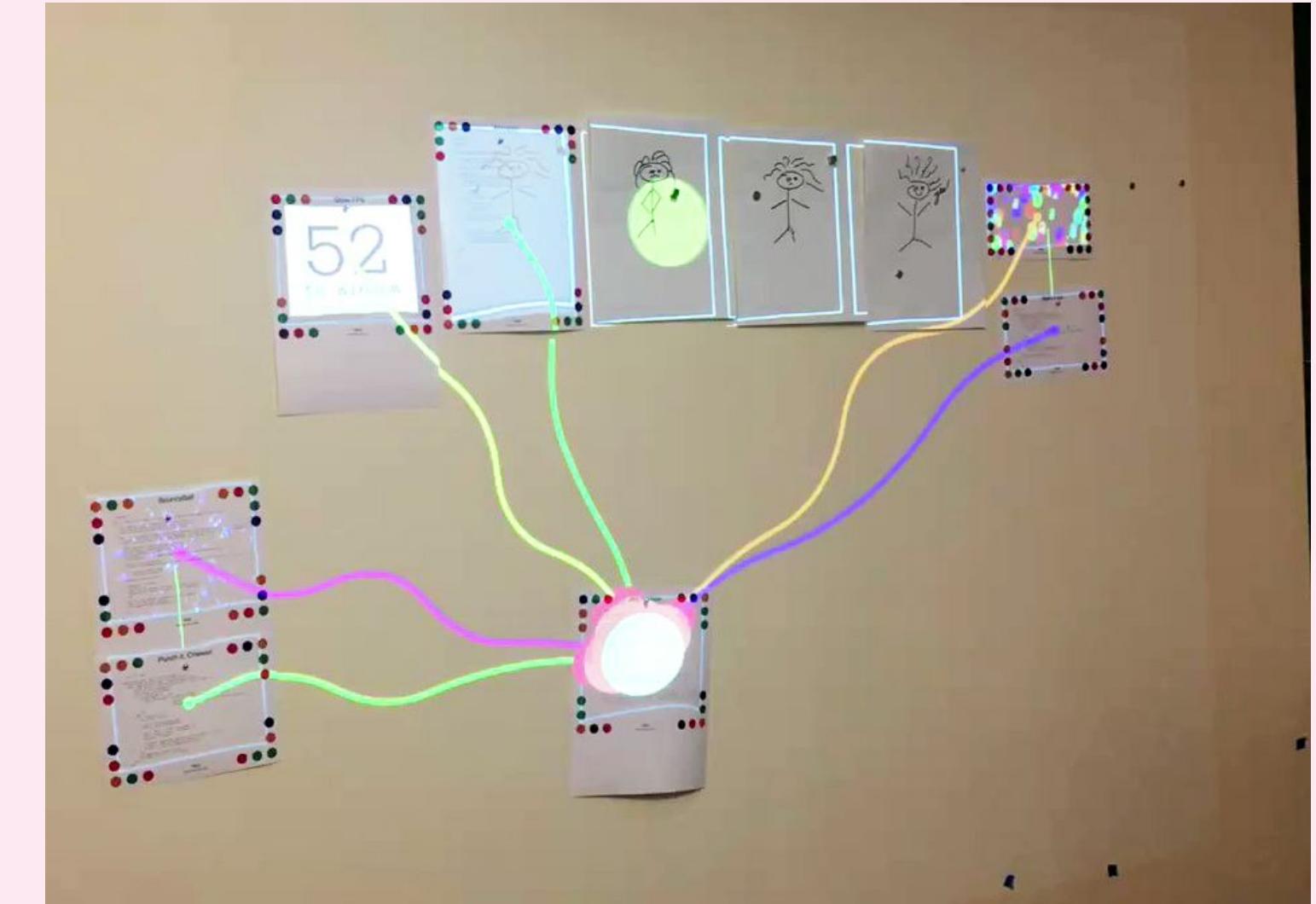
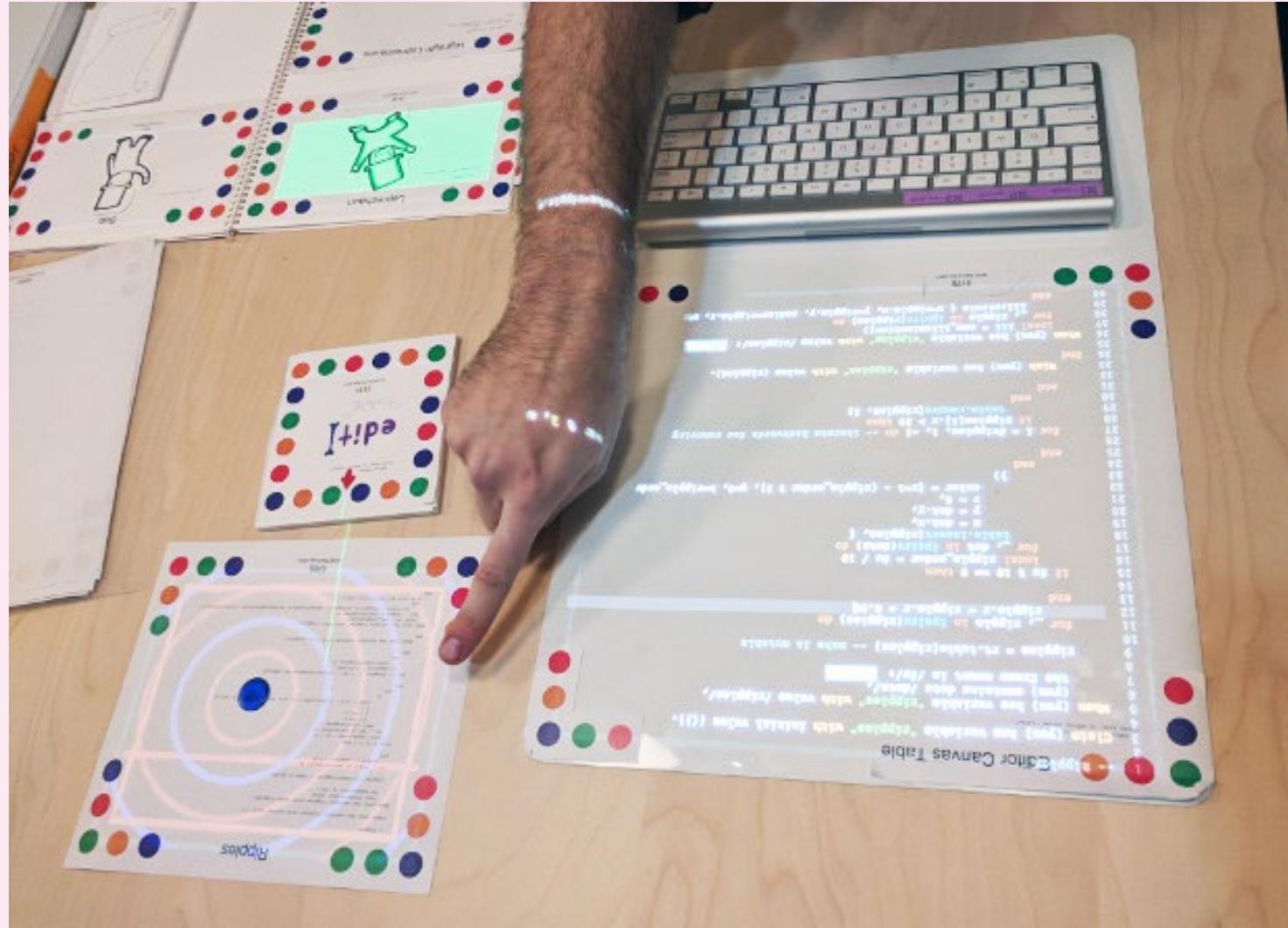
2015



Ira Winder and Joshua Fabian at the MIT Media Lab (Changing Places Group)

# Dynamicland

2017-Today



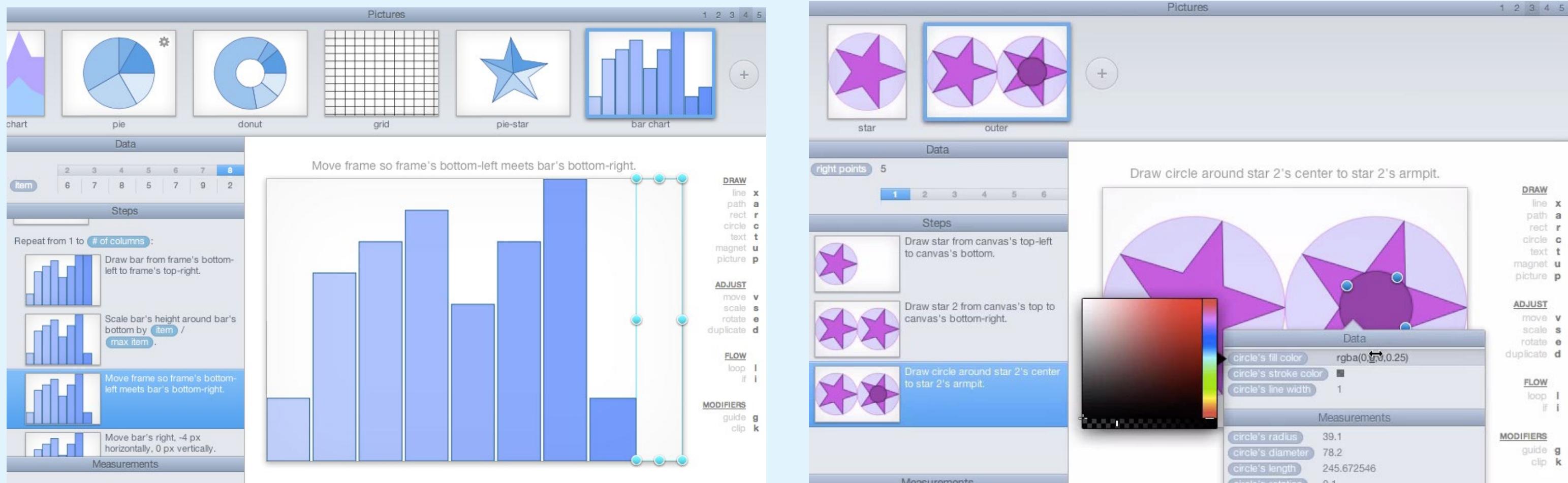
Bret Victor and Others at Y Combinator Research's Human Advancement Research Community

# Dynamic Diagram Experiments

Direct-Manipulation +  
Symbolic Relationships

# Drawing Dynamic Visualizations

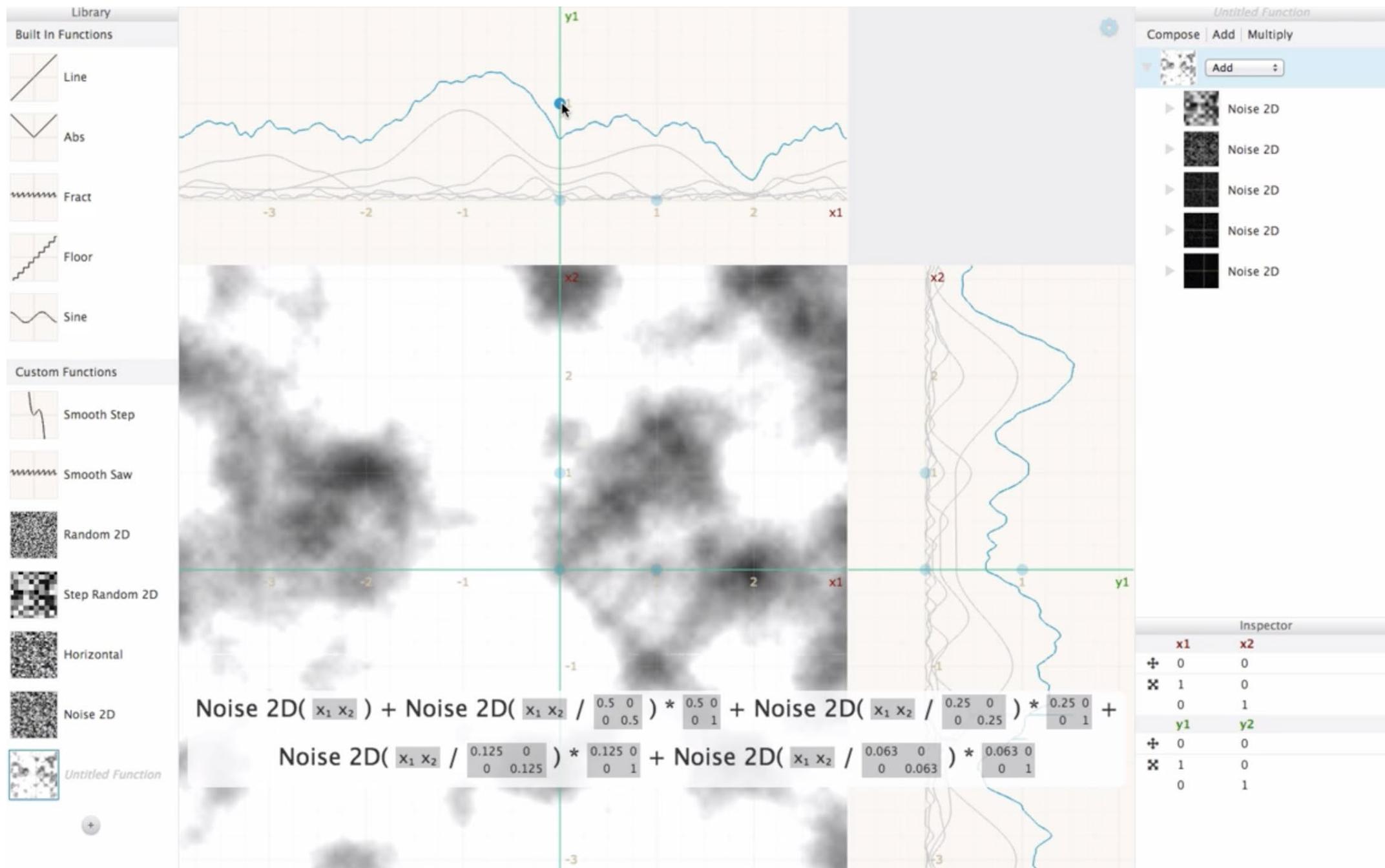
2013



Bret Victor

# ShaderShop

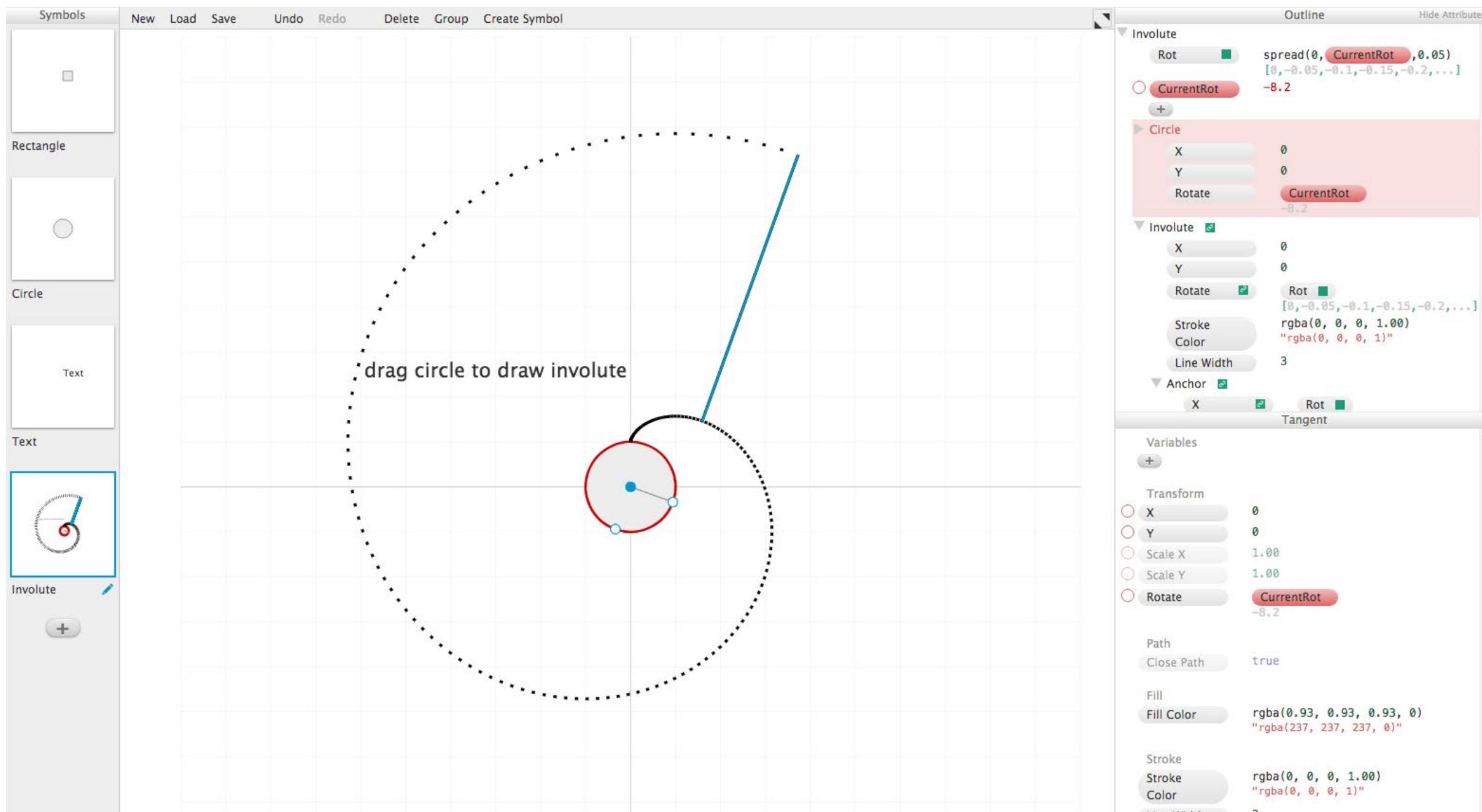
2014



Toby Schachman for Communication Design Group at SAP

# Appaaratus

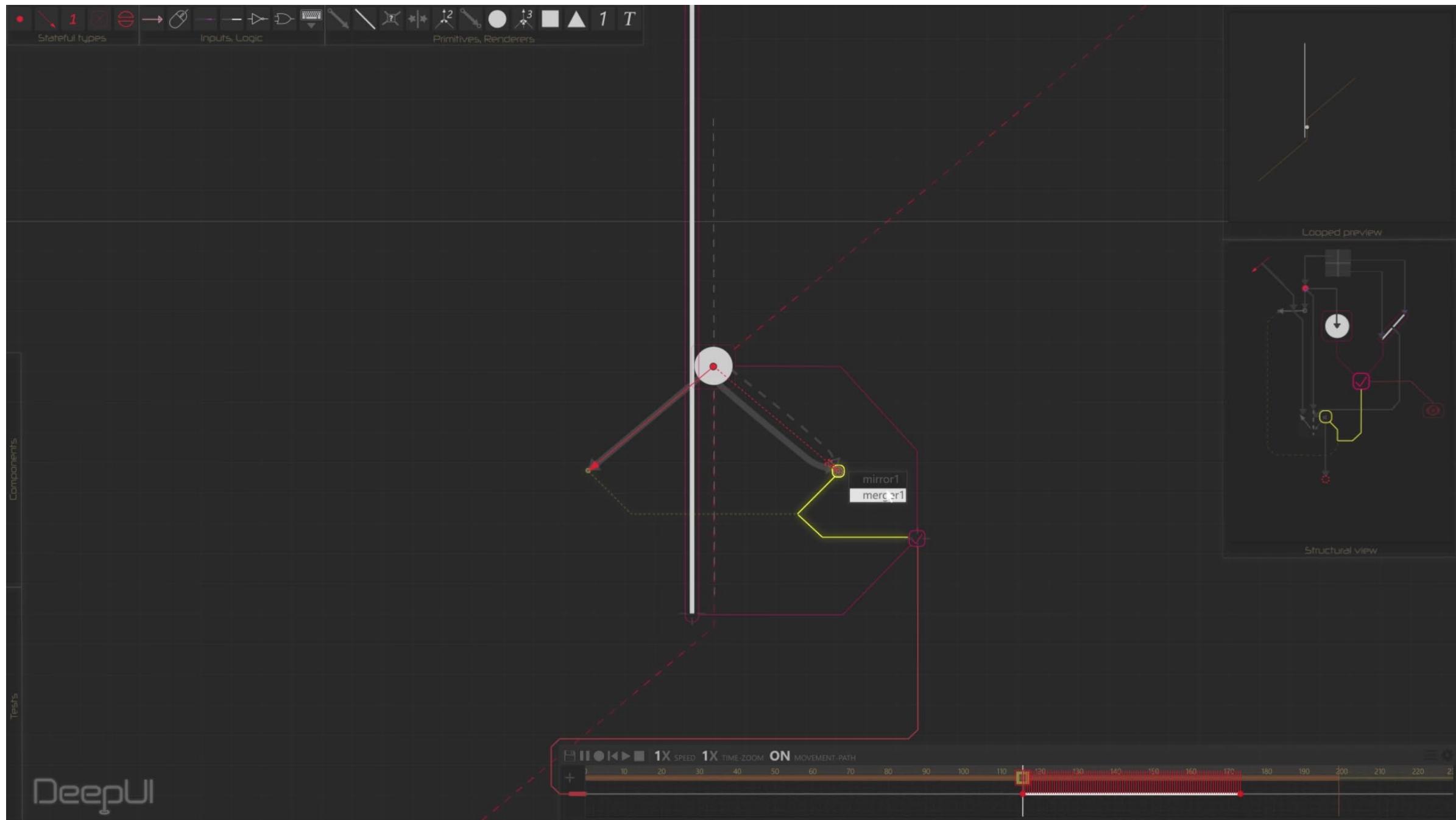
2015



Toby Schachman for Communication Design Group at SAP

# DeepUI

2017



Arnold Lagler

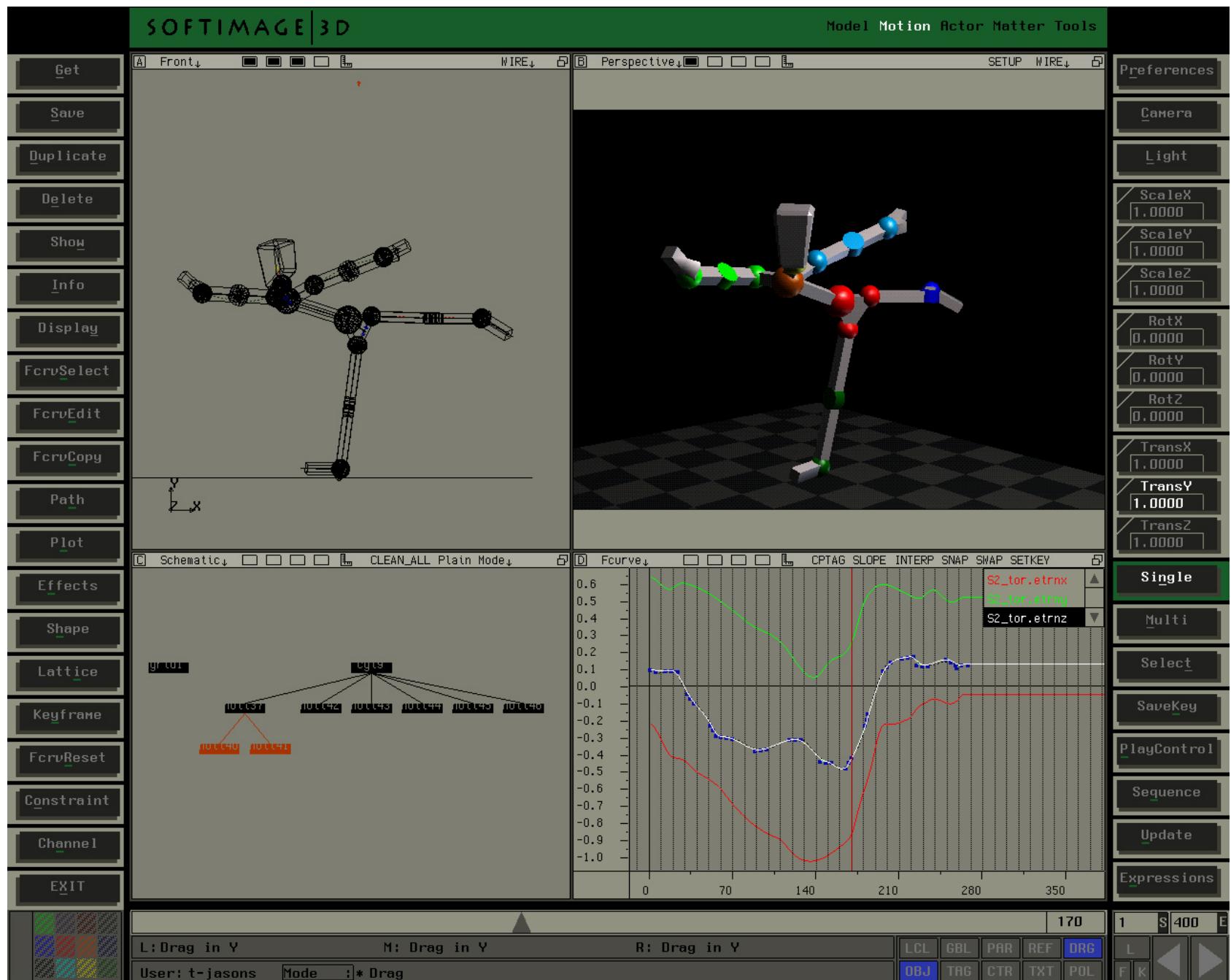
# Learning From/Within Cyberspace

## How Game-Tech is enabling the Next Wave

# 3D (Mesh/NURBs) Creation Suites

# Softimage 3D

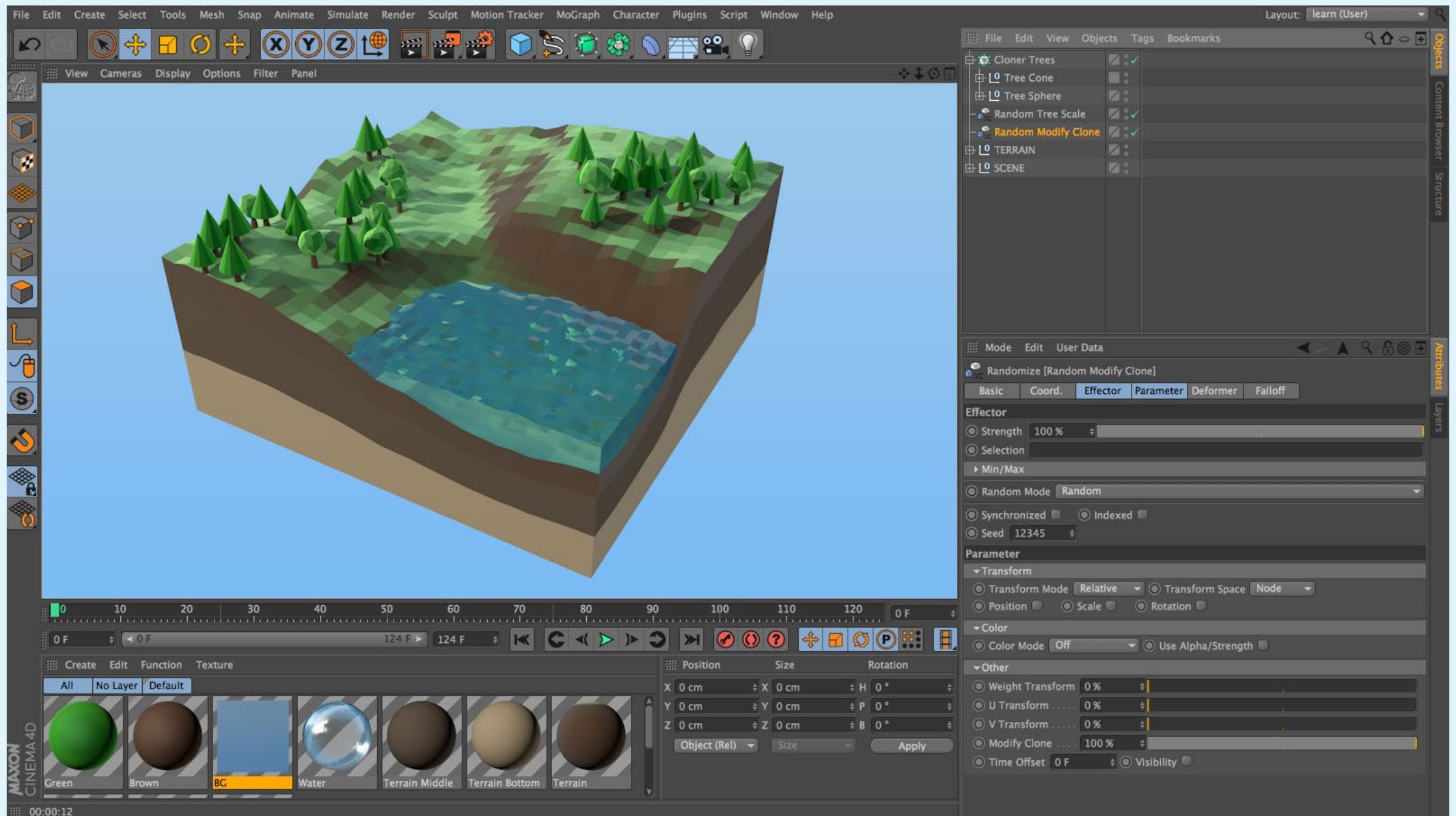
1988-2001



Softimage, Co.

# Cinema 4D

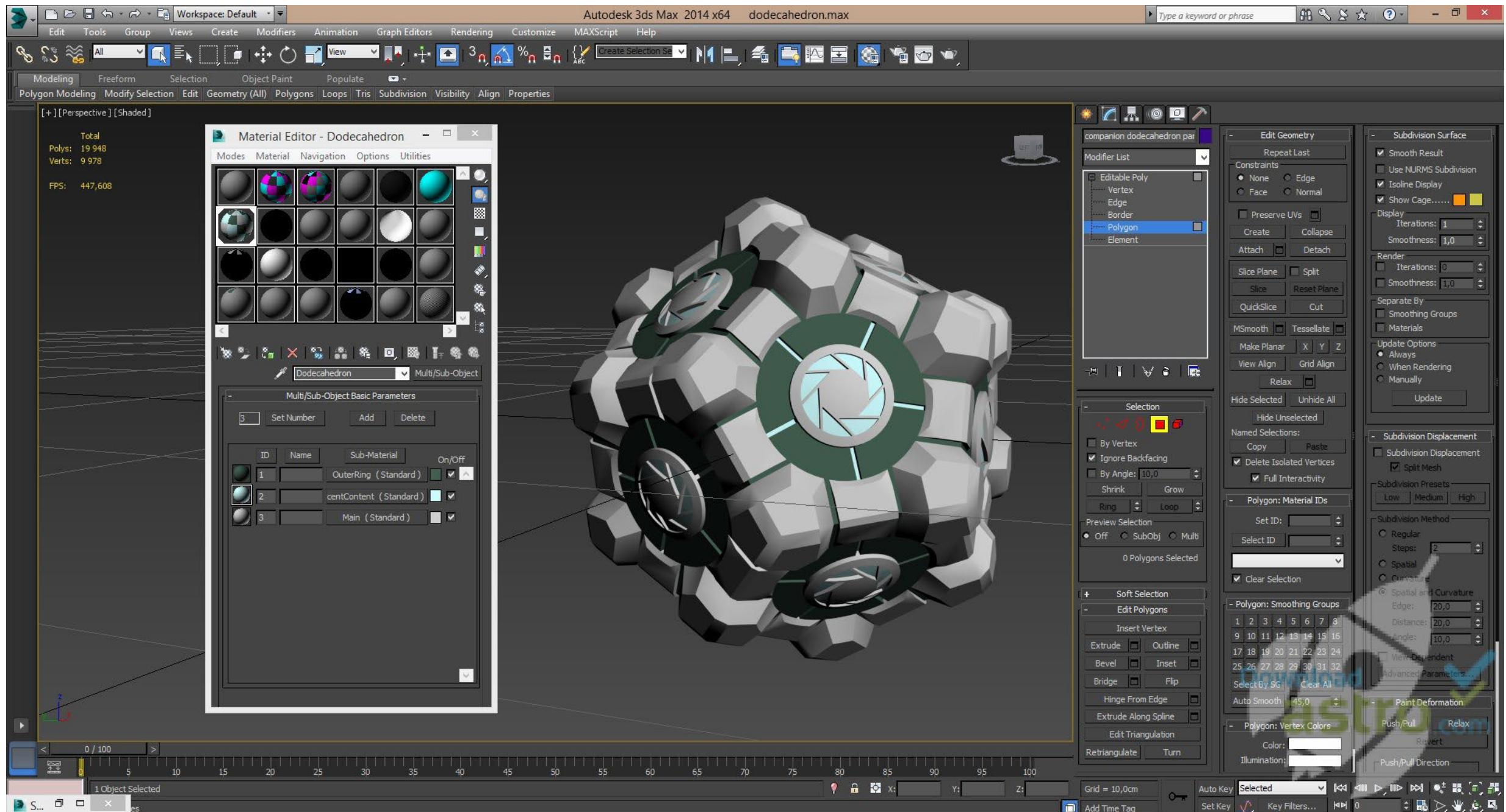
1990-2001



MAXON Computer GmbH

# 3DS Max

1996-Today



Autodesk

# Shake

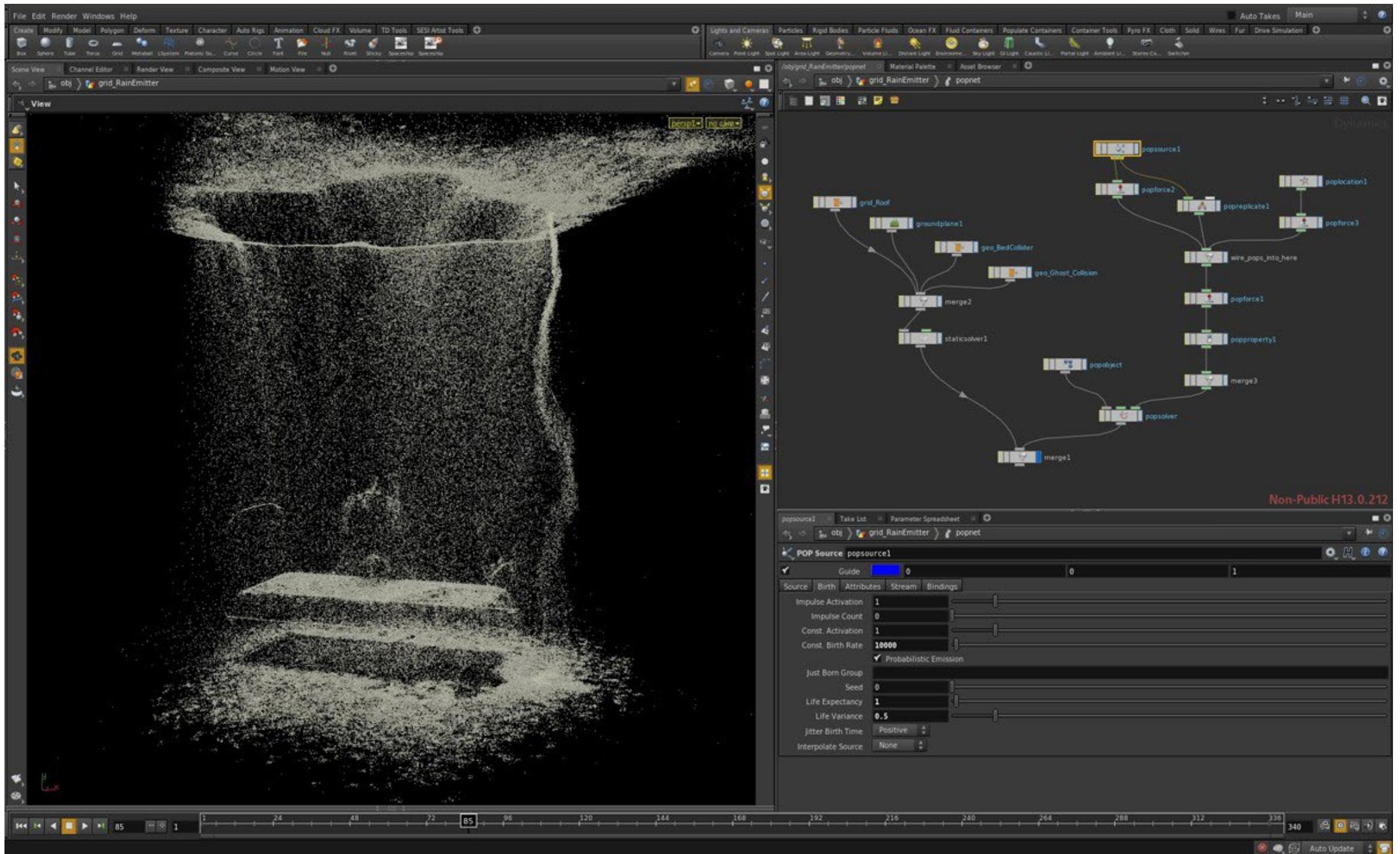
# 1997-2008



Arnaud Hervas and Allen Edwards for Nothing Real L.L.C. (Later Apple)

# Houdini

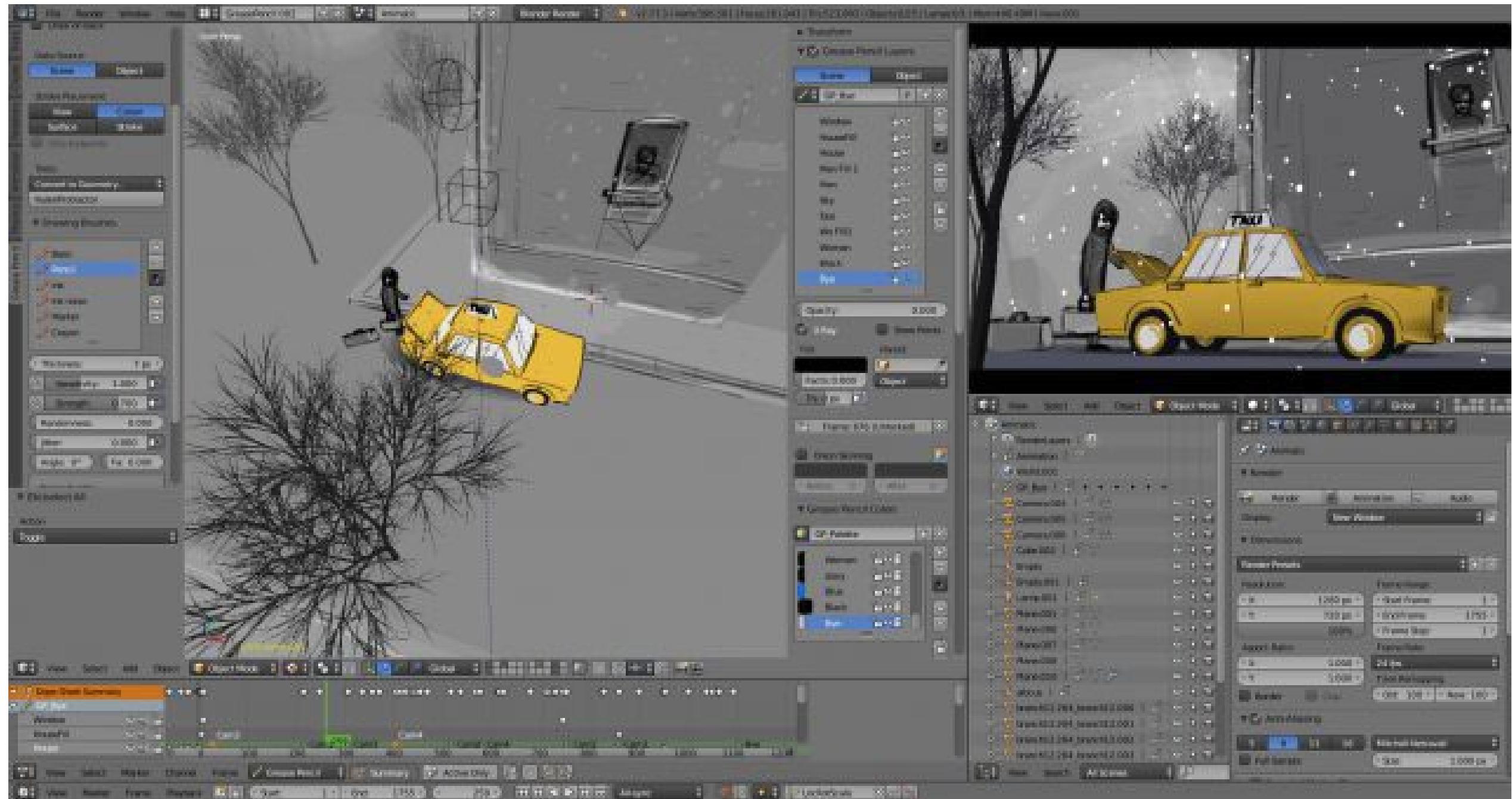
1996-Today



Side Effects Software Inc

# Blender

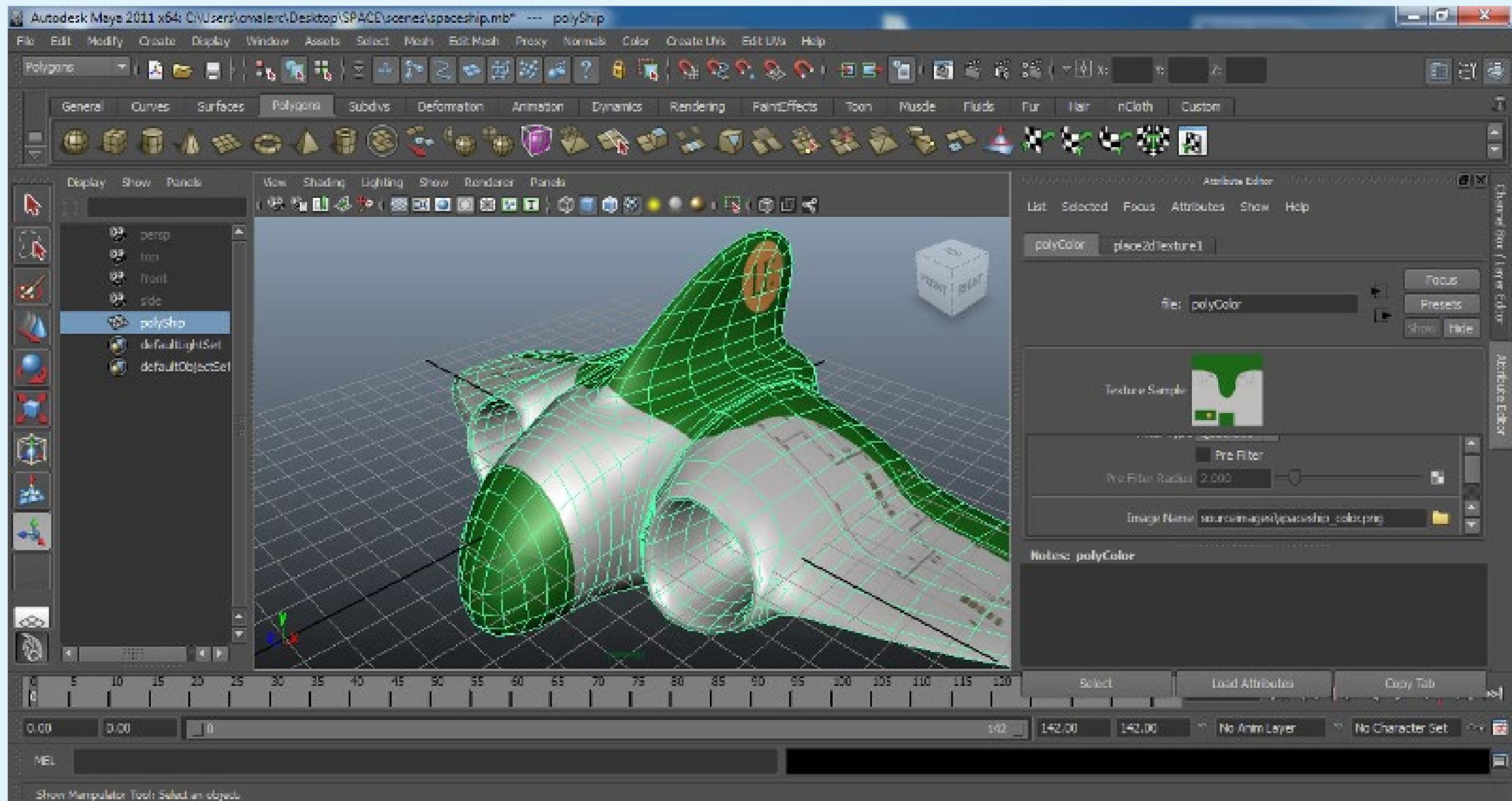
# 1998-Today



Blender Foundation

# Maya

# 1998-Today



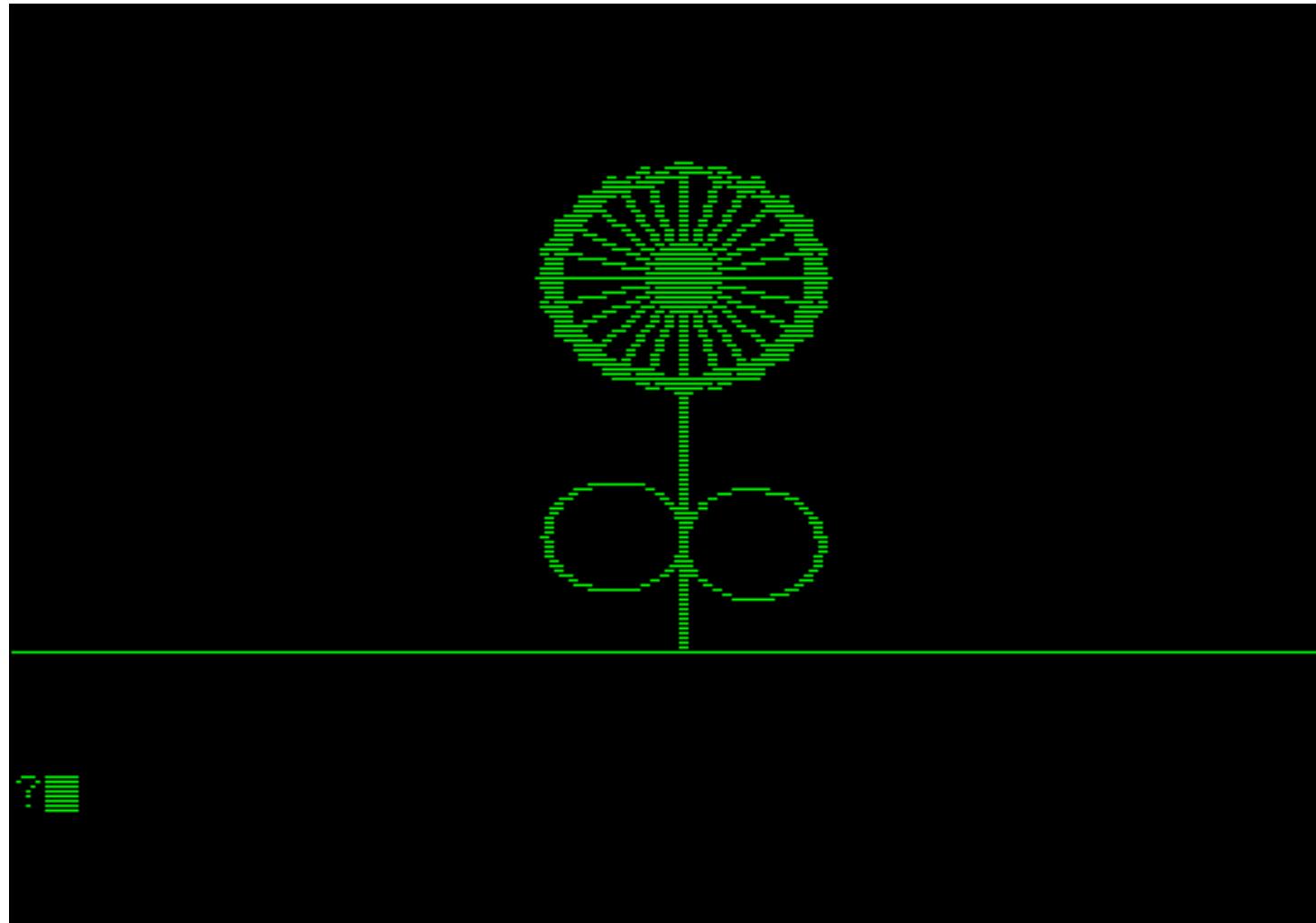
Alias Systems Corporation (Later Autodesk)

# Learnable Programming

Offers a 'Gentle Introduction'

# LOGO

1967-



```
BOAST WHO'S THE GREATEST?]
INT QUEST FOR STOP
RE COURSE TO STOP
PRINT TRY AGAIN
END

CHAT WHAT'S YOUR NAME?]
INT HELLO REQUEST
PRINT TYPE SOMETHING YOU LIKED
REQUEST USE IT I'M GLAD YOU LIKED REQUEST
END

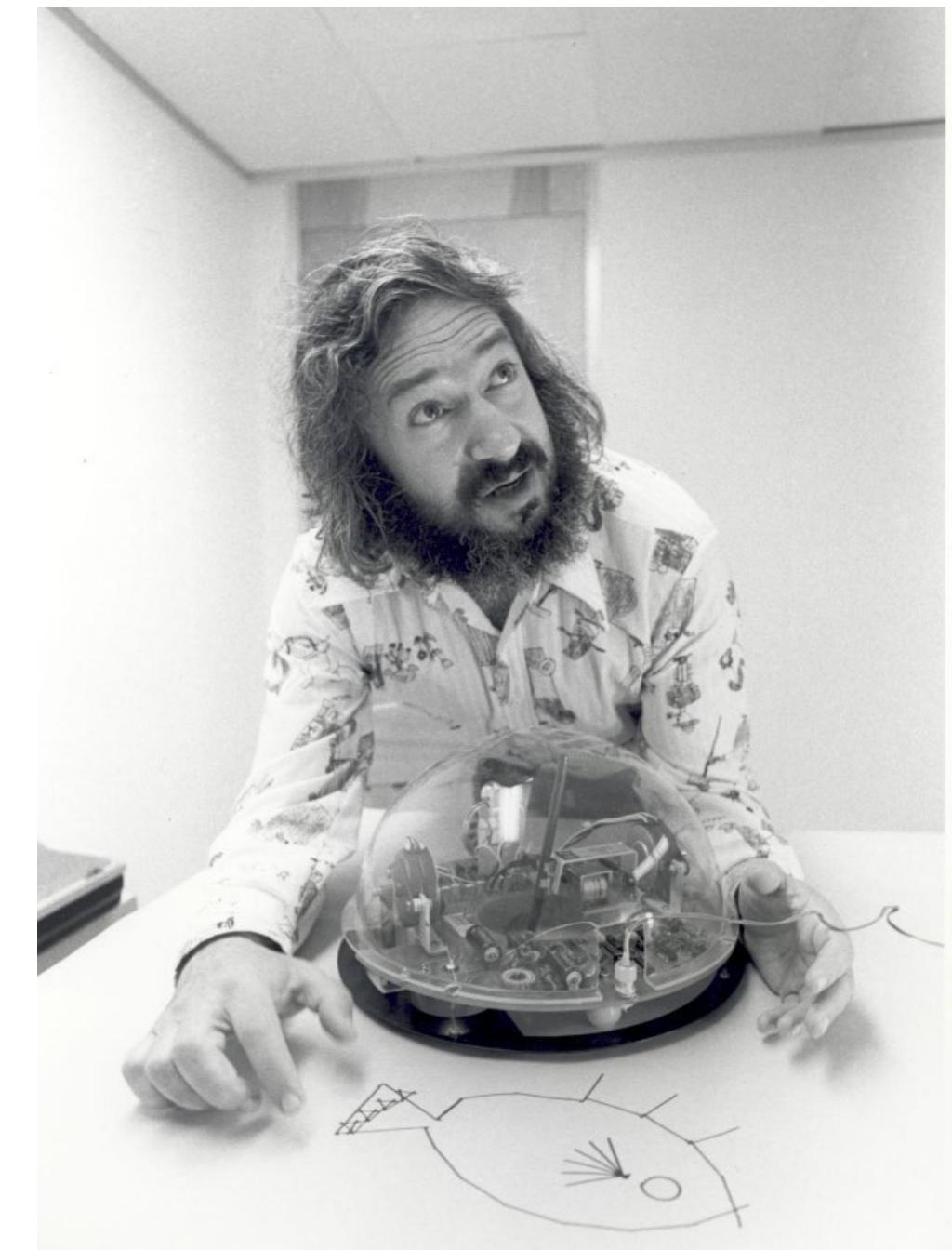
TIME EXCUSE ME, WHAT TIME IS IT?]
INT "TIME REQUEST
MAKE TIME IT'S LATE!]
PRINT SEE YOU LATER.]
END
```

QUESTION MARK

Wally Feurzeig, Seymour Papert, Cynthia Solomon at MIT

# Turtle

1969



Seymour Papert and Others at the MIT Logo Lab

# Q-Basic

1991

```
File Edit View Search Run Bar colors.bas Help  
  
SUB rainbow(x,y, c, radius)  
    ' Draws a circular rainbow. Our rainbow is a circle with thickness,  
    ' where color is defined by the angle (determined using arctangent).  
    ' In order to draw a thick circle, we simply draw a box and ignore  
    ' those pixels that are not part of the arc. The selection is done  
    ' by measuring the distance from the origin. Only pixels that fall  
    ' within the certain range are accerted.  
    minr = radius * 0.6  
    minr2 = minr*minr      ' minimum radius ^ 2  
    maxr2 = radius*radius ' maximum radius ^ 2  
    pi! = 3.14159!  
    xradius = radius*4/3  ' aspect ratio correction  
    FOR py=-radius TO radius  
        py2 = py*py  
        FOR px=-xradius TO xradius  
            pxr! = px*3/4  
            r = pxr!*pxr! + py2  
            IF r >= minr2 AND r <= maxr2 THEN  
                ' angle! = ATAN2(py, px) -- only QBasic does not have ATAN2.  
                IF px = 0 THEN angle! = SGN(py) * pi! * 0.5 ELSE angle! = ATN(py / pxr)  
                IF px < 0 THEN angle! = angle! + pi!  
                IF py < 0 THEN angle! = angle! + pi! + pi!  
                ' Convert angle into a color and place the pixel.  
                cc! = angle! * 12 / pi! + 6  
                cc = INT(cc! + RND) ' Quantize with random dithering  
                PSET(x+px, y+py), c + (cc + 24) MOD 24  
            END IF  
        NEXT  
    NEXT  
END SUB  
  
SUB Speak(x,y, e$, f$) STATIC  
    IF f=0 THEN f = EREFFILE: OPEN "VOX" AS f  
    IOCTL f, e$ + "~~" + f$ + "$"  
    ' Speak text. This is something I added to my copy of DOSBox.  
    ' Feel free to comment out those two lines if it does not work for you.  
    IF y>=200 THEN EXIT SUB  
END SUB  
END SUB  
  
QBasic masquerade mode engaged. 158 lines | 13:21:24 | *00154:001
```



Microsoft

# LEGO Mindstorms RCX

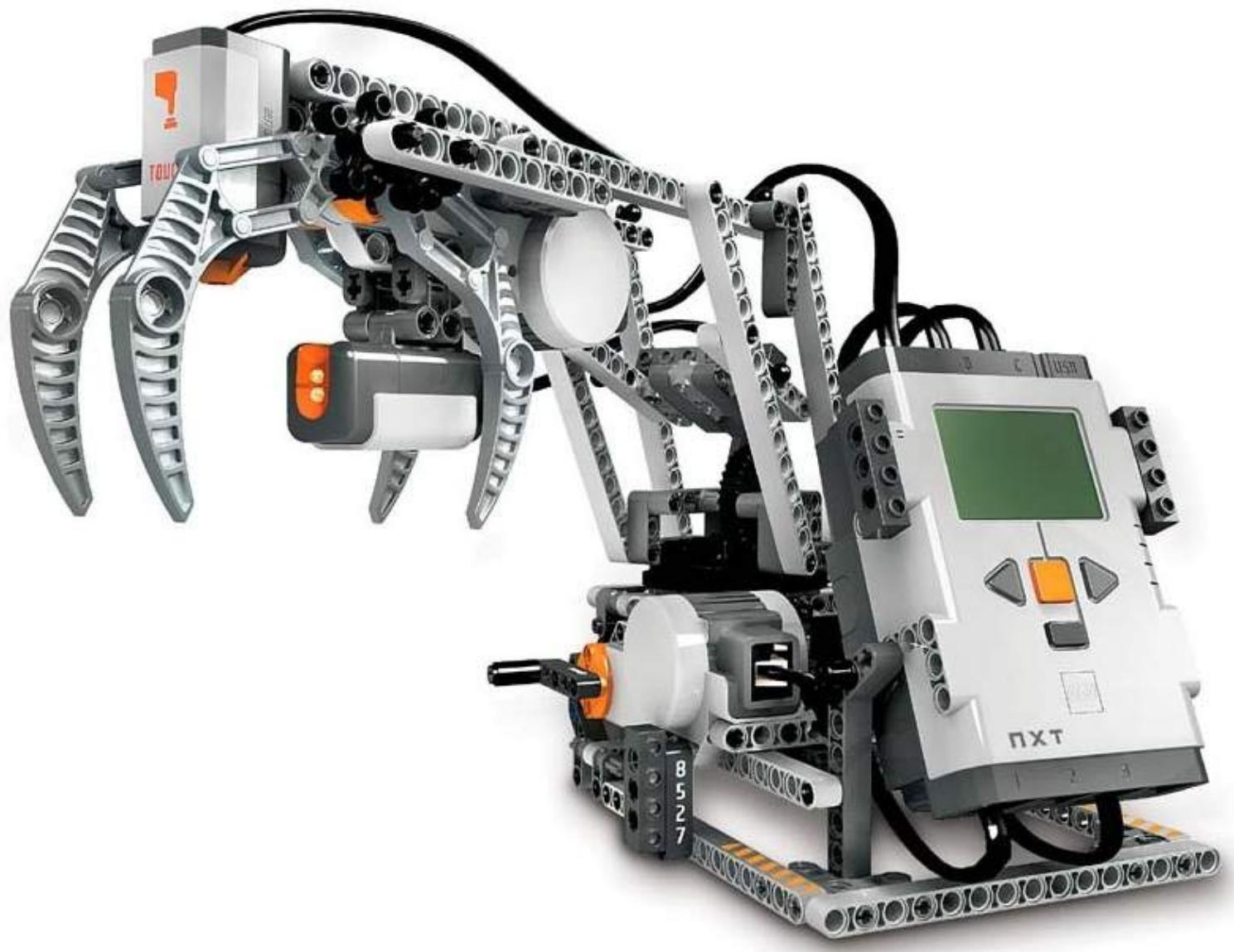
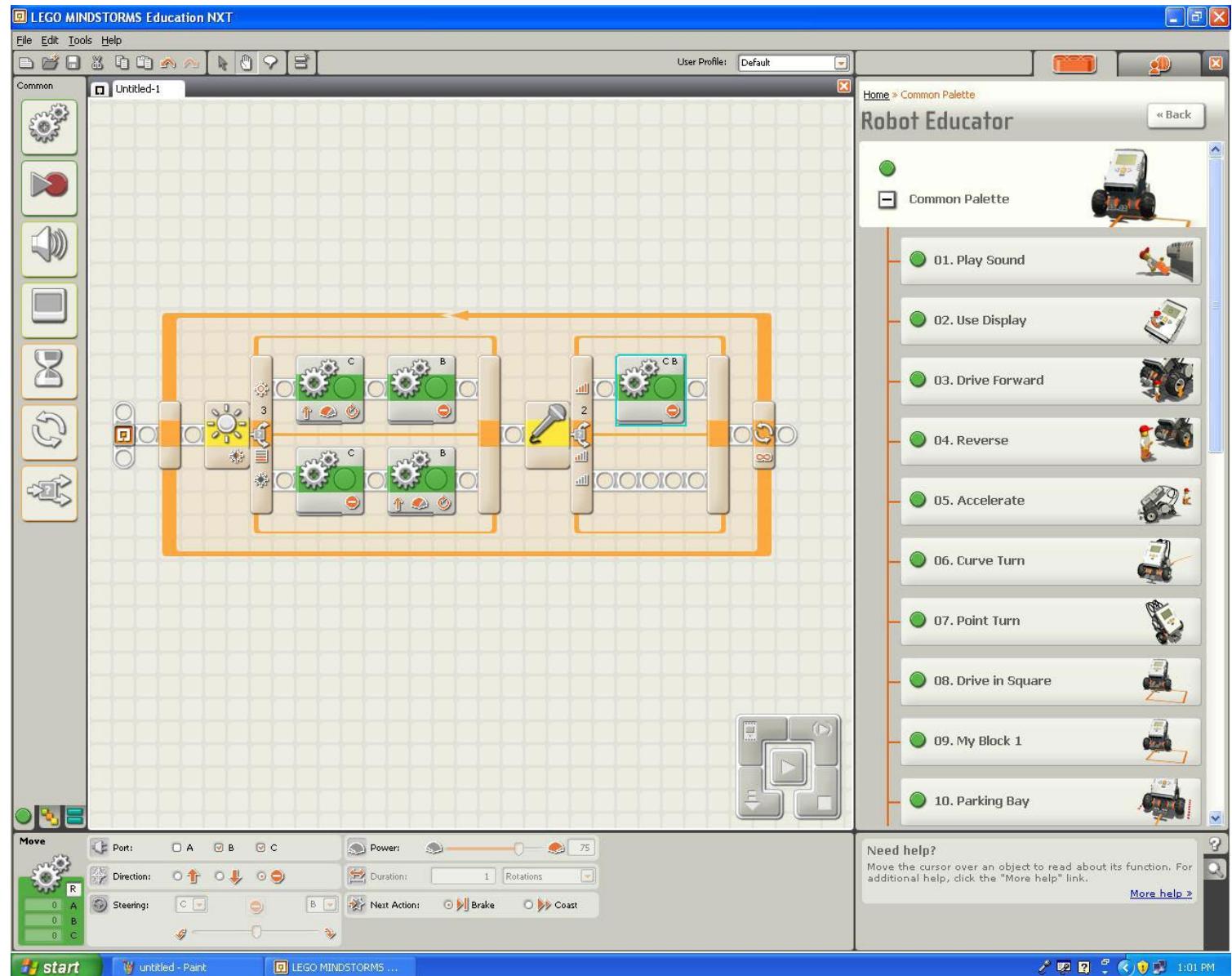
1998-2006



LEGO in Partnership with the MIT Media Lab (Lifelong Learning Group)

# LEGO Mindstorms NXT

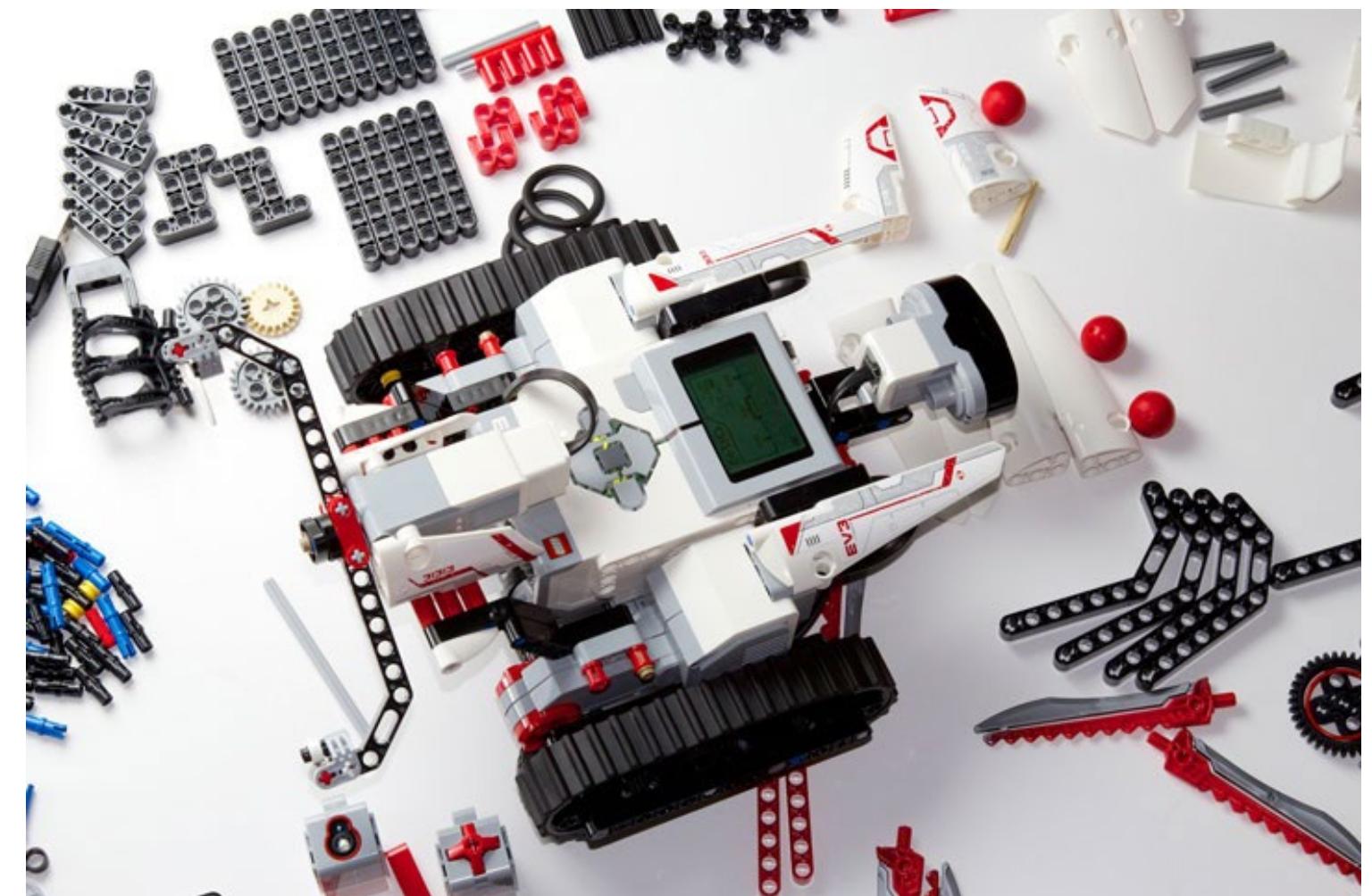
2006-2013



LEGO in Partnership with the MIT Media Lab (Lifelong Learning Group)

# LEGO Mindstorms EV3

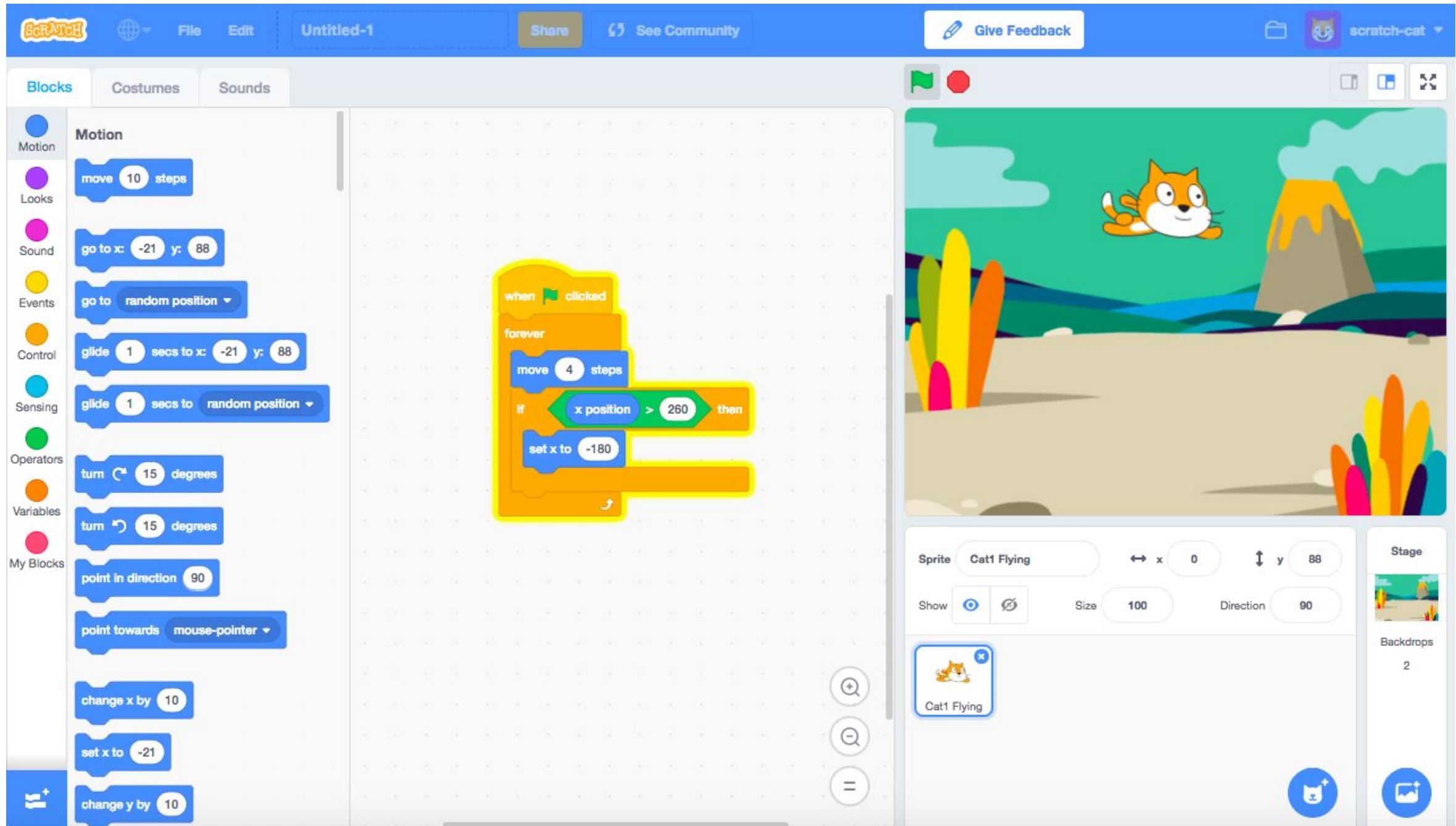
2013-Today



LEGO in Partnership with the MIT Media Lab (Lifelong Learning Group)

# Scratch

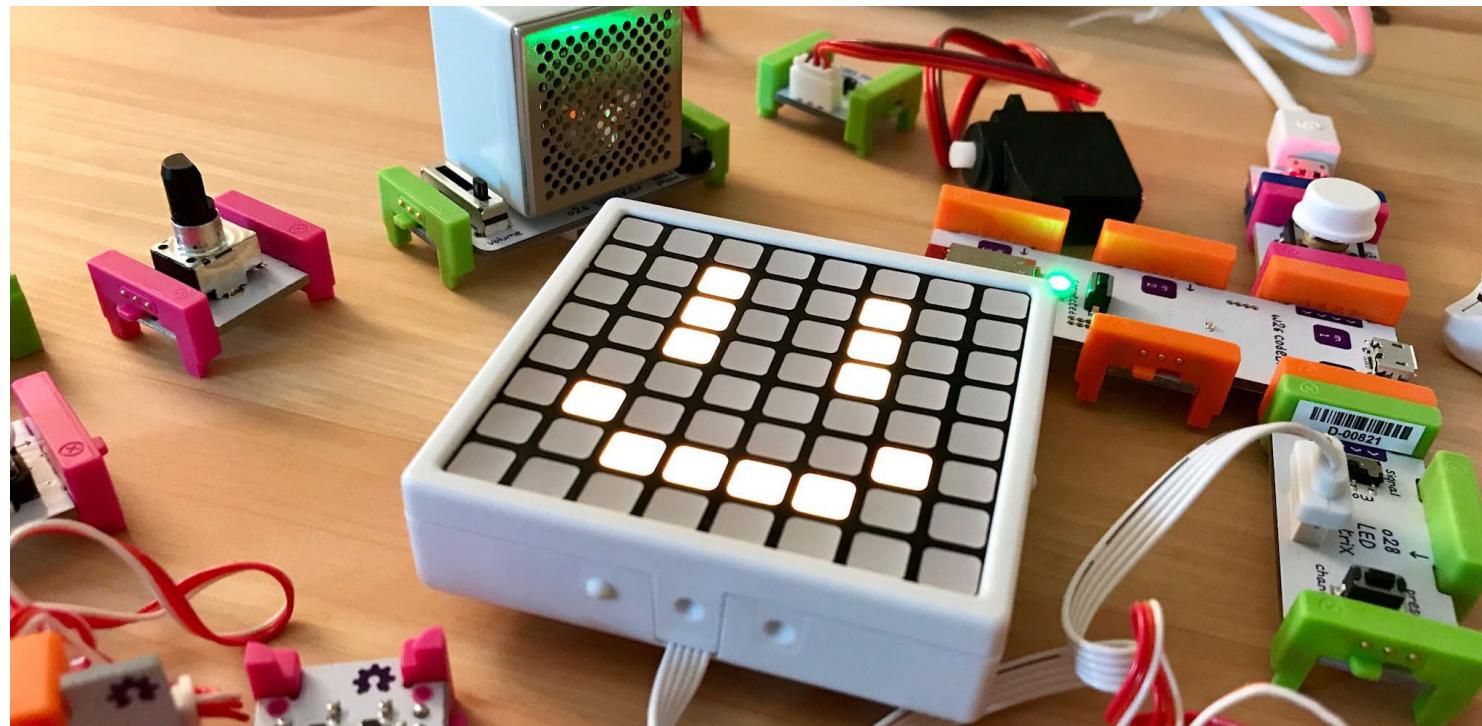
2002-Today



Mitchel Resnick and others at the MIT Media Lab (Lifelong Kindergarten Group)

# littleBits

# 2011-Today



Ayah Bdeir at the MIT Media Lab (Now littleBits Electronics Inc)

# Koov

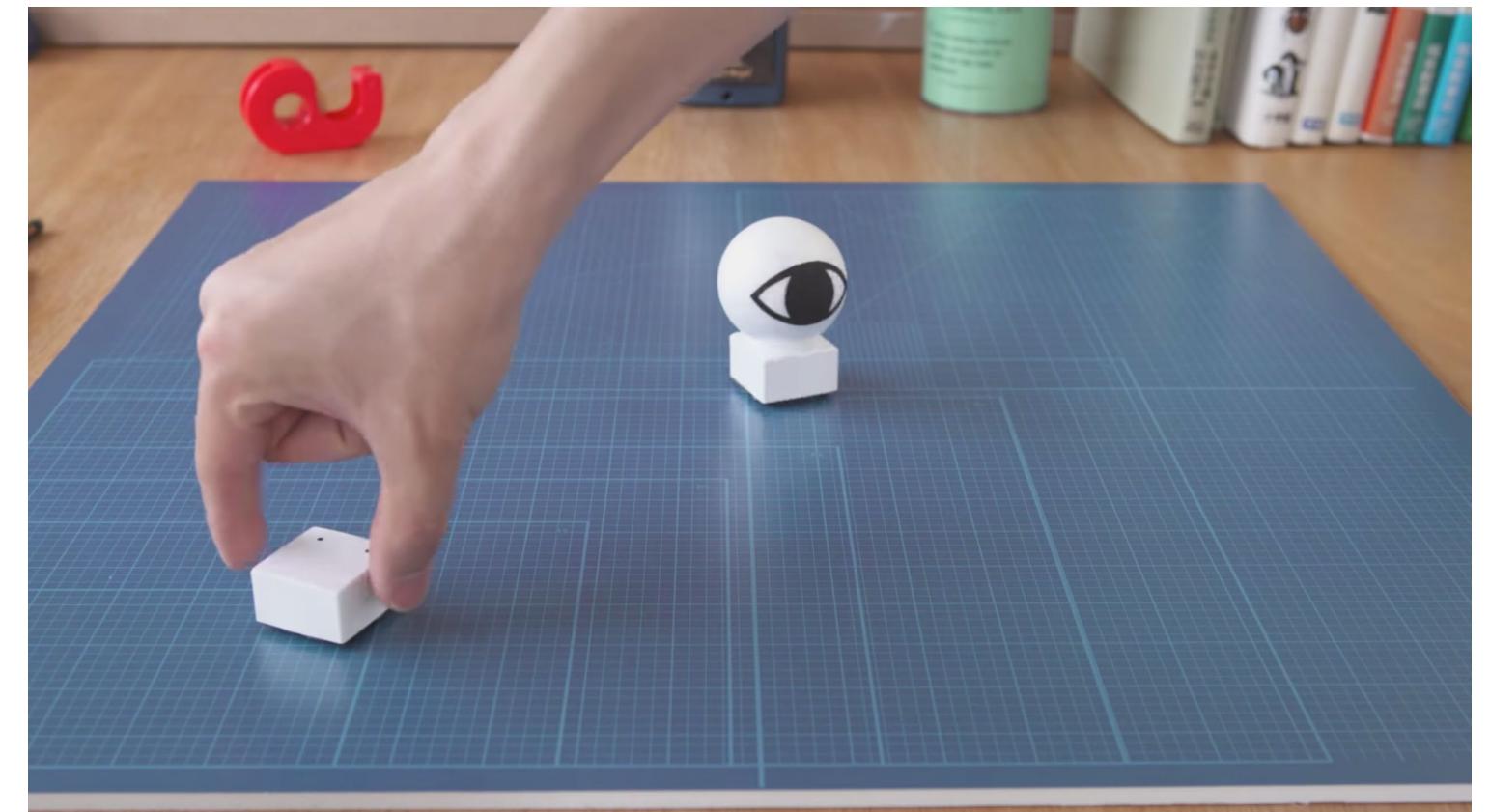
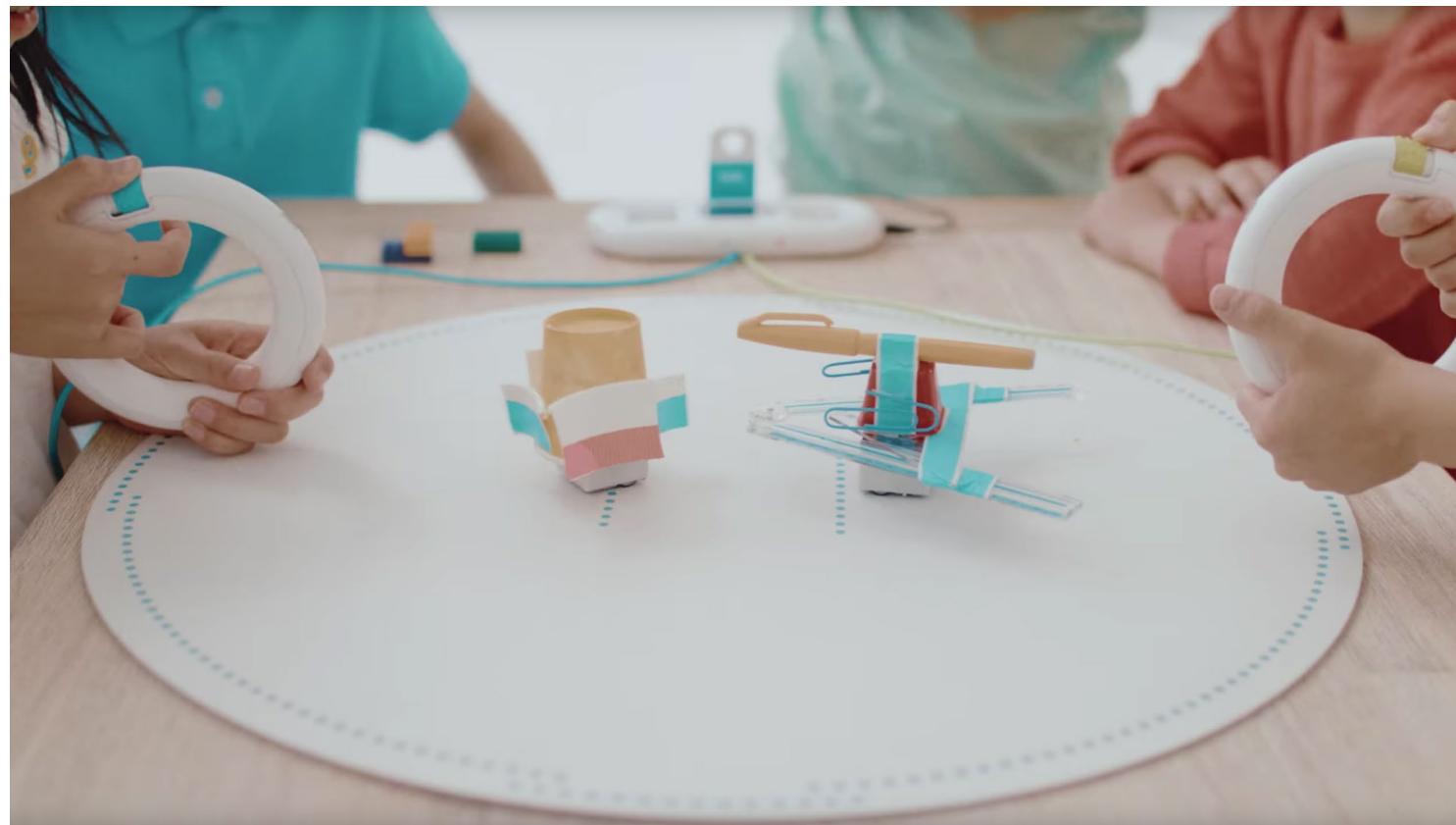
# 2017-Today



Sony

# Toio

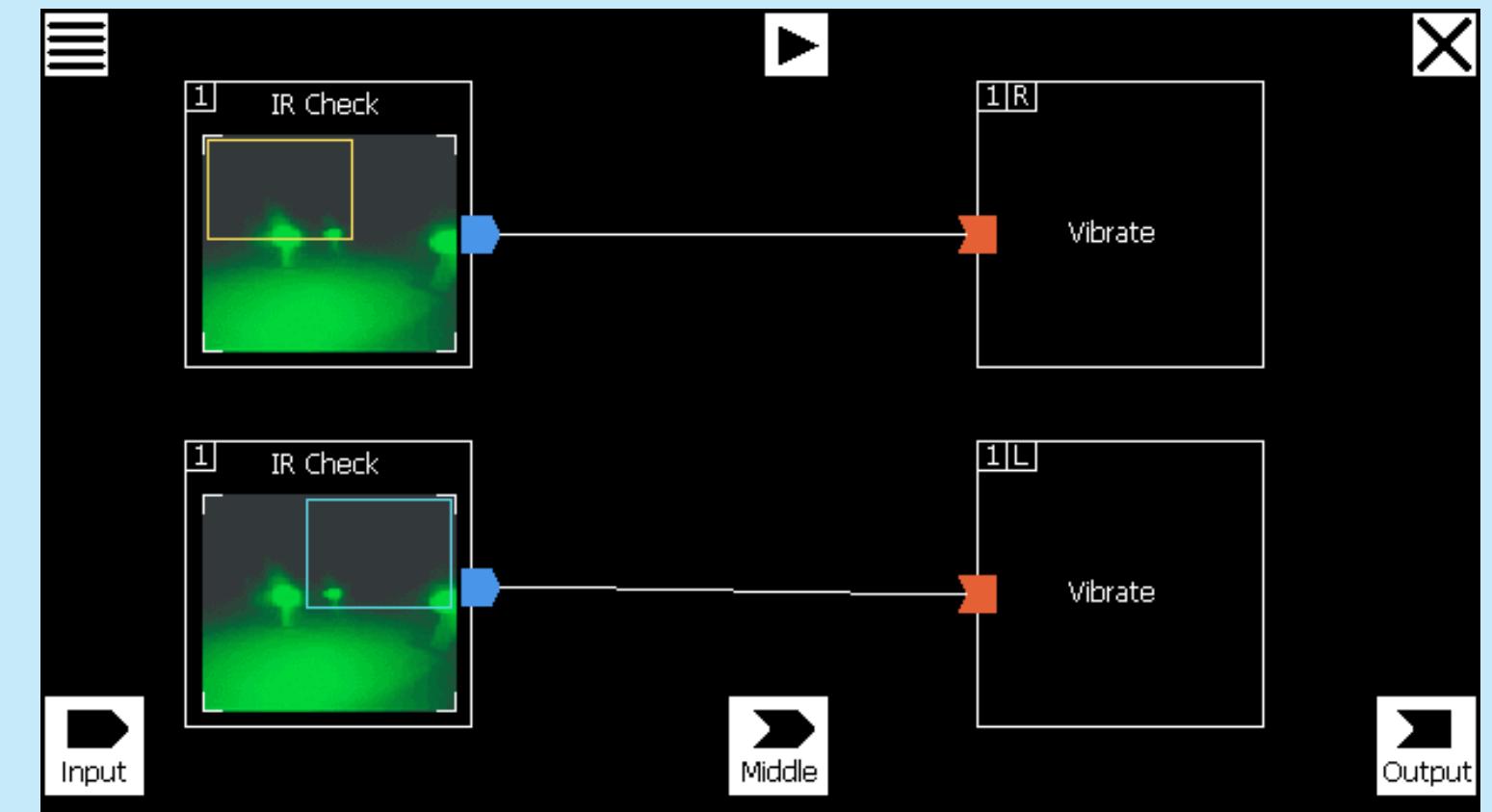
# 2017-Today



Sony

# Joy-Con Garage

2018-Today



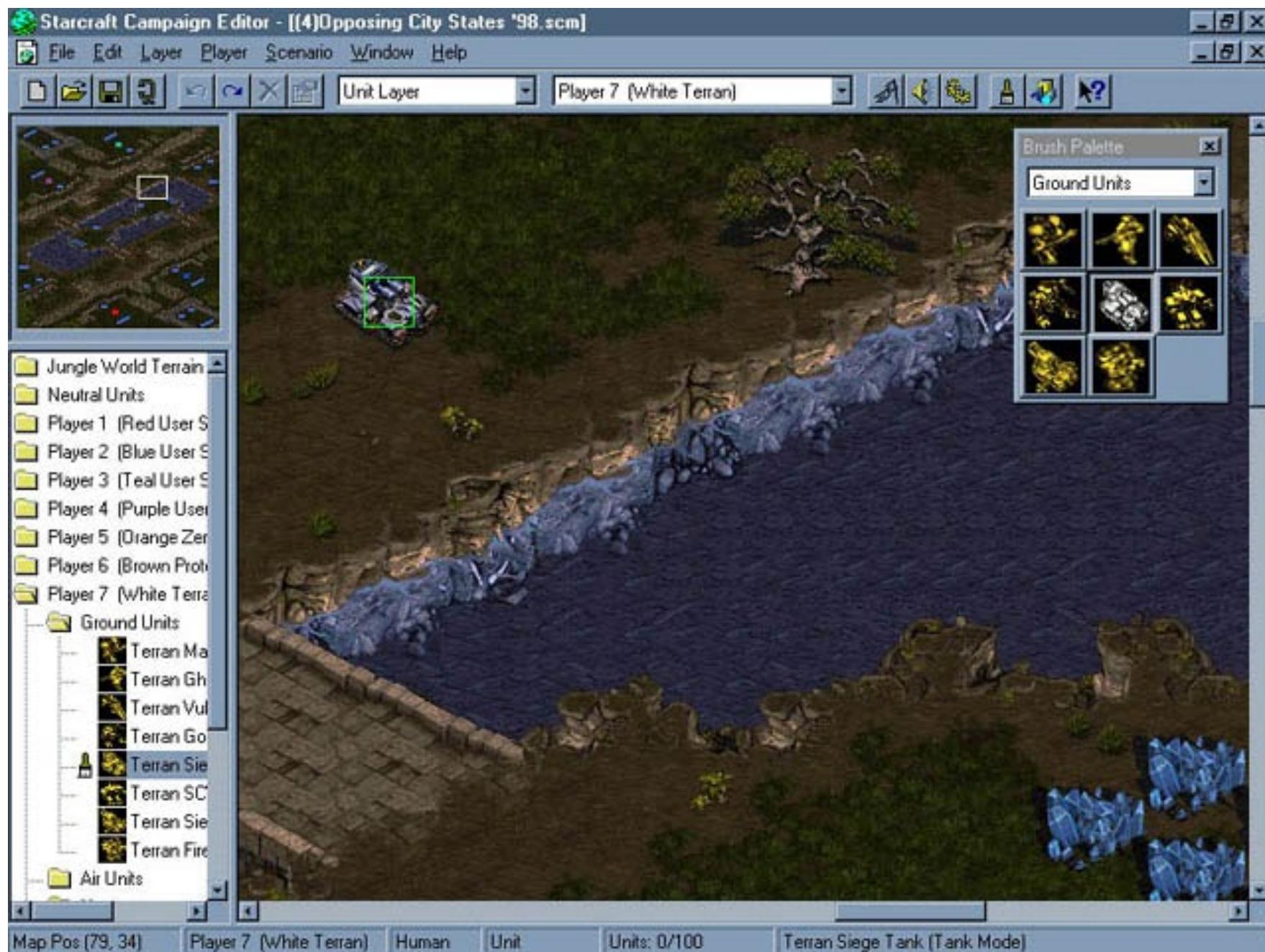
Nintendo

# Game Authoring

From video games to VR/AR/MR

# StarEdit

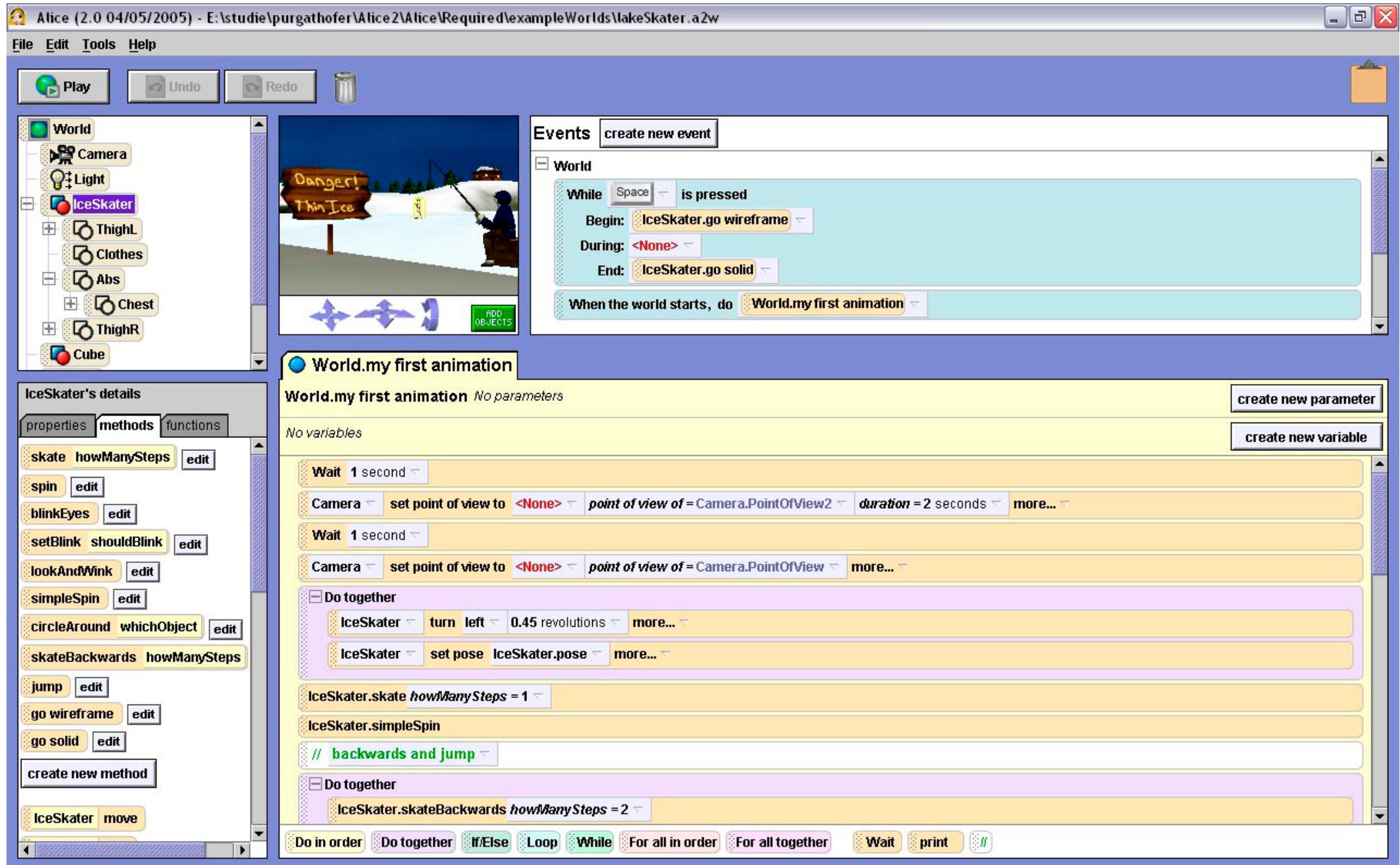
1998



Blizzard Entertainment

# Alice

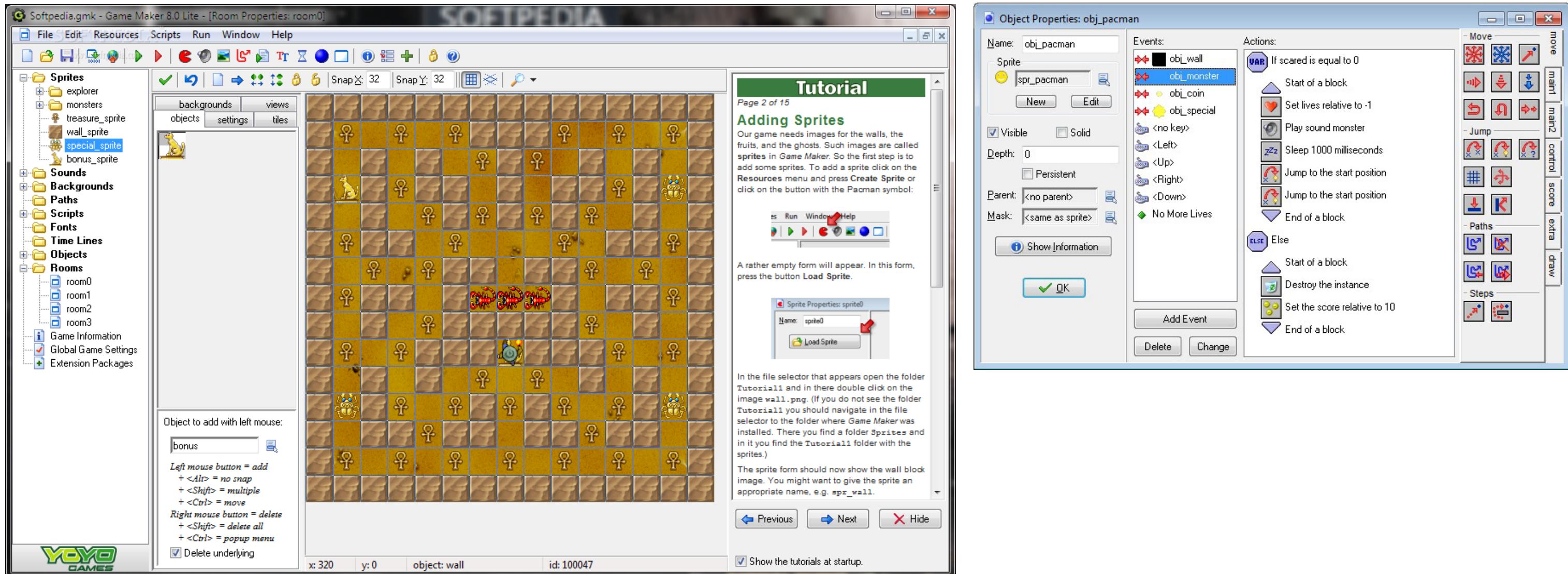
1998-Today



Randy Pausch and Others at Carnegie Mellon University

# GameMaker

1999-Today



Mark Overmars

# Unreal Engine

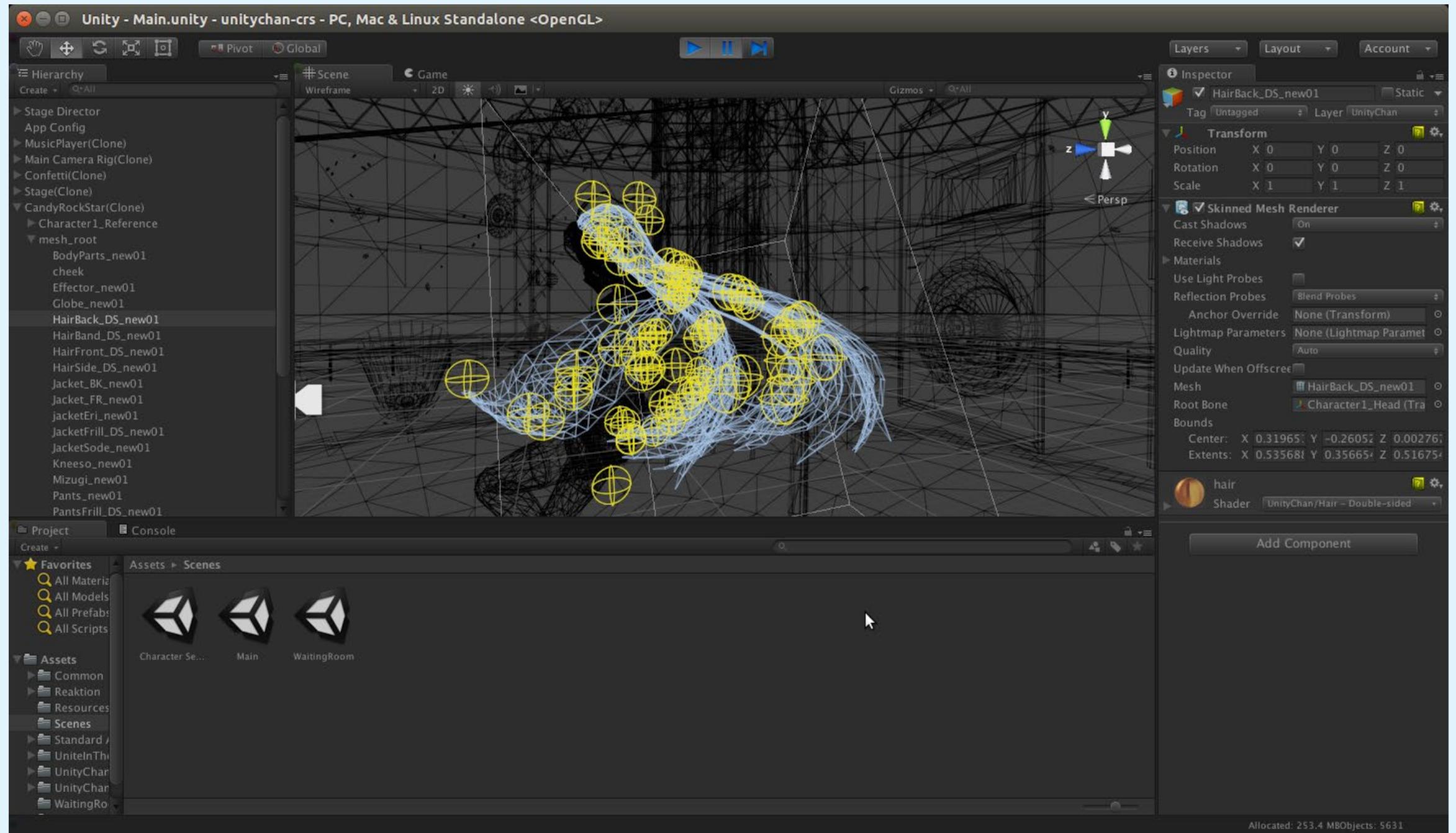
1998-Today



Epic Games

# Unity

2005-Today



Unity Technologies

# Source Filmmaker

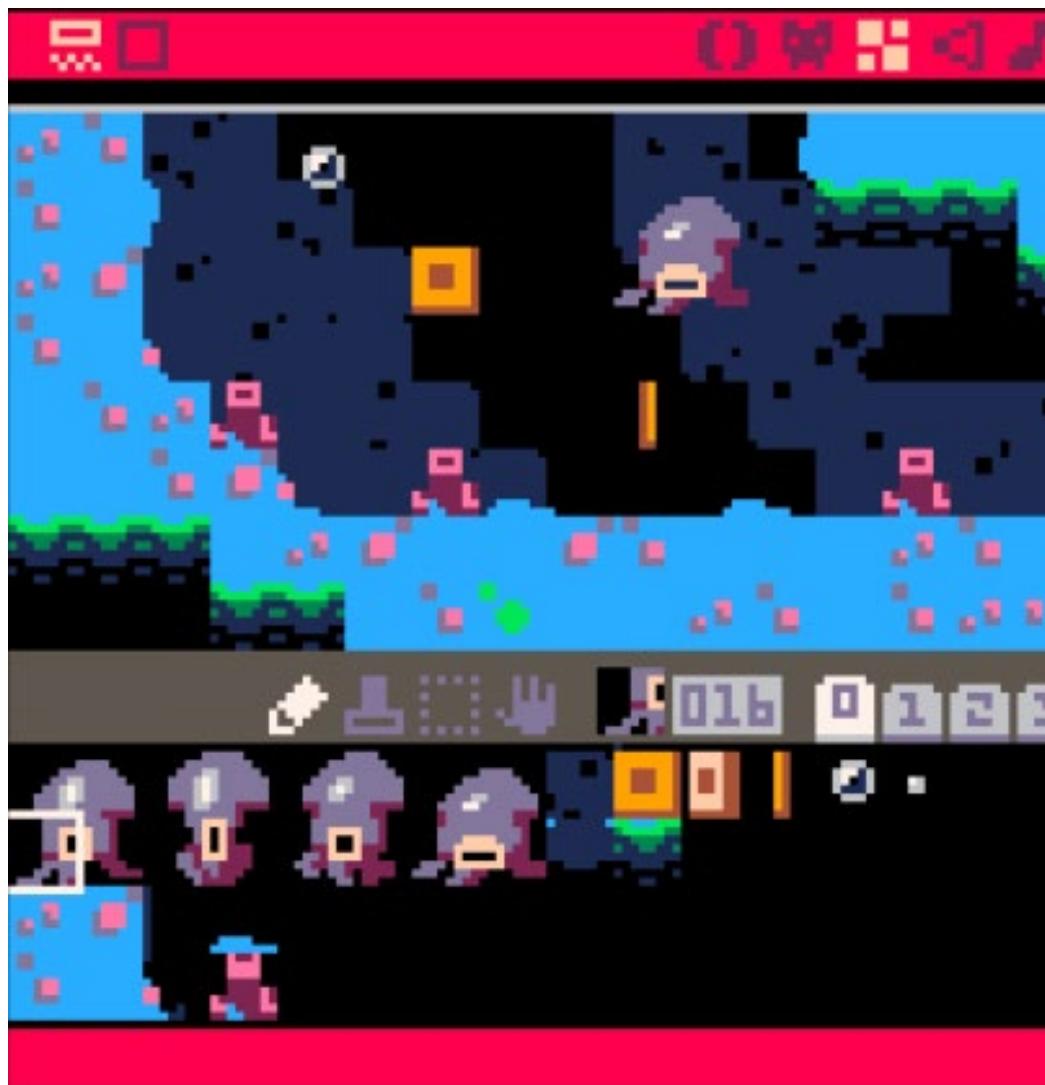
2012-Today



Valve Corporation

# Pocket Chip for PICO-8

2016-2018



```
15
= 64
= 64
IR = 1
TION _INIT()
()

TION _UPDATE()
P1.DIR == 1) THEN P1.Y=P1.Y-1 END
P1.DIR == 2) THEN P1.X=P1.X+1 END
P1.DIR == 3) THEN P1.Y=P1.Y+1 END
P1.DIR == 4) THEN P1.X=P1.X-1 END
BTn(0)) THEN P1.DIR = P1.DIR - 1
P1.DIR > 4) THEN P1.DIR = 1 END
P1.DIR < 1) THEN P1.DIR = 4 END
LINE 14/26 177/8192
```

Next Thing Co

# Games with Authoring

Decisions have Confined Effects  
on the Player Experience

# Sims (Series)

2000-Today



Electronic Arts and Others

# Second Life

2003-Today



Linden Lab

# Spore

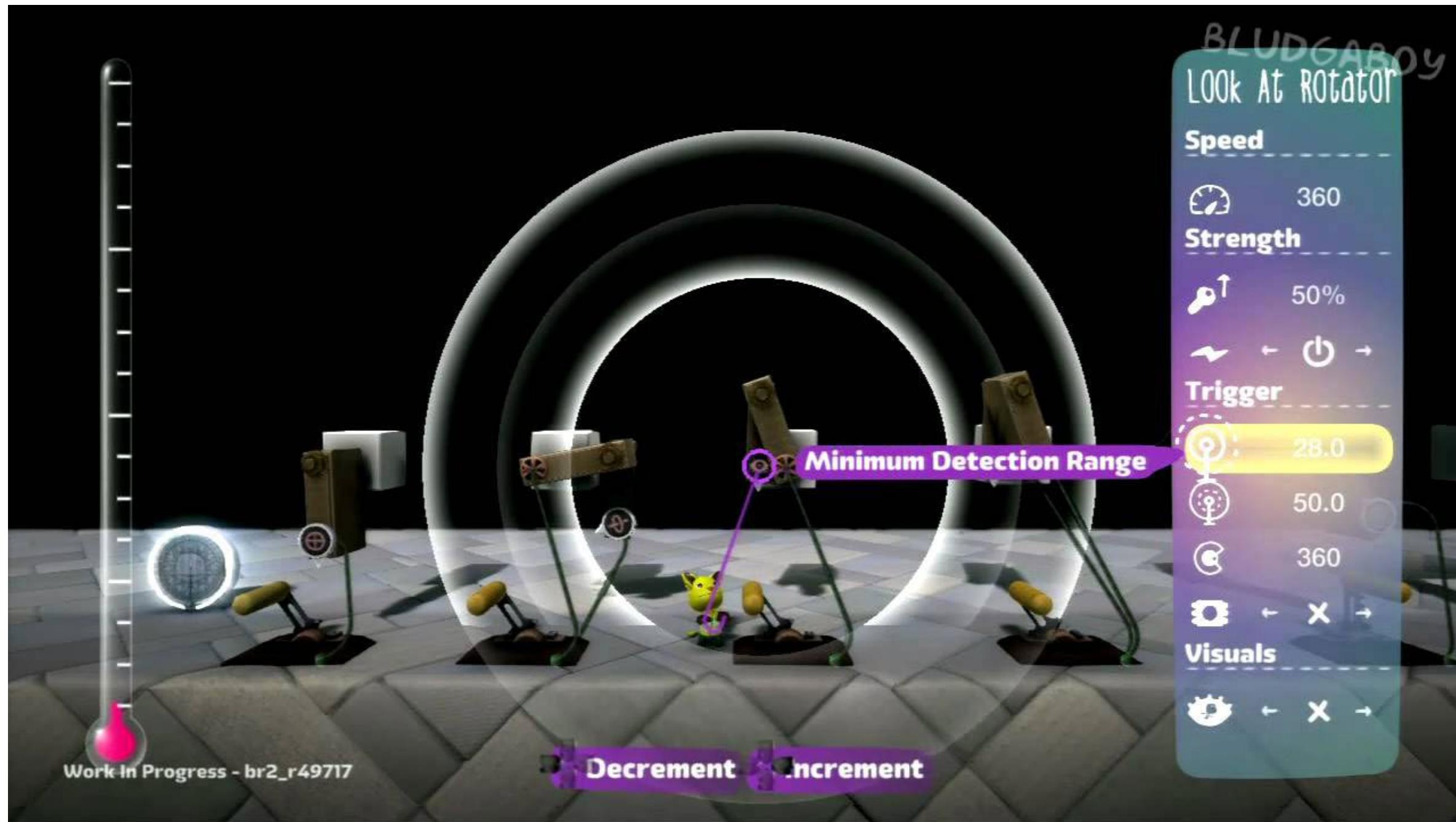
2008



Maxis

# Little Big Planet (Series)

2008-2014



Media Molecule and Sumo Digital

# ModNation Racers

2010



United Front Games and SIE San Diego Studio

# Grand Theft Auto V

2013



Rockstar Games

# Games as Authoring

Decisions & Their Side-Effects Are the  
Whole Player Experience

# The Incredible Machine (Series)

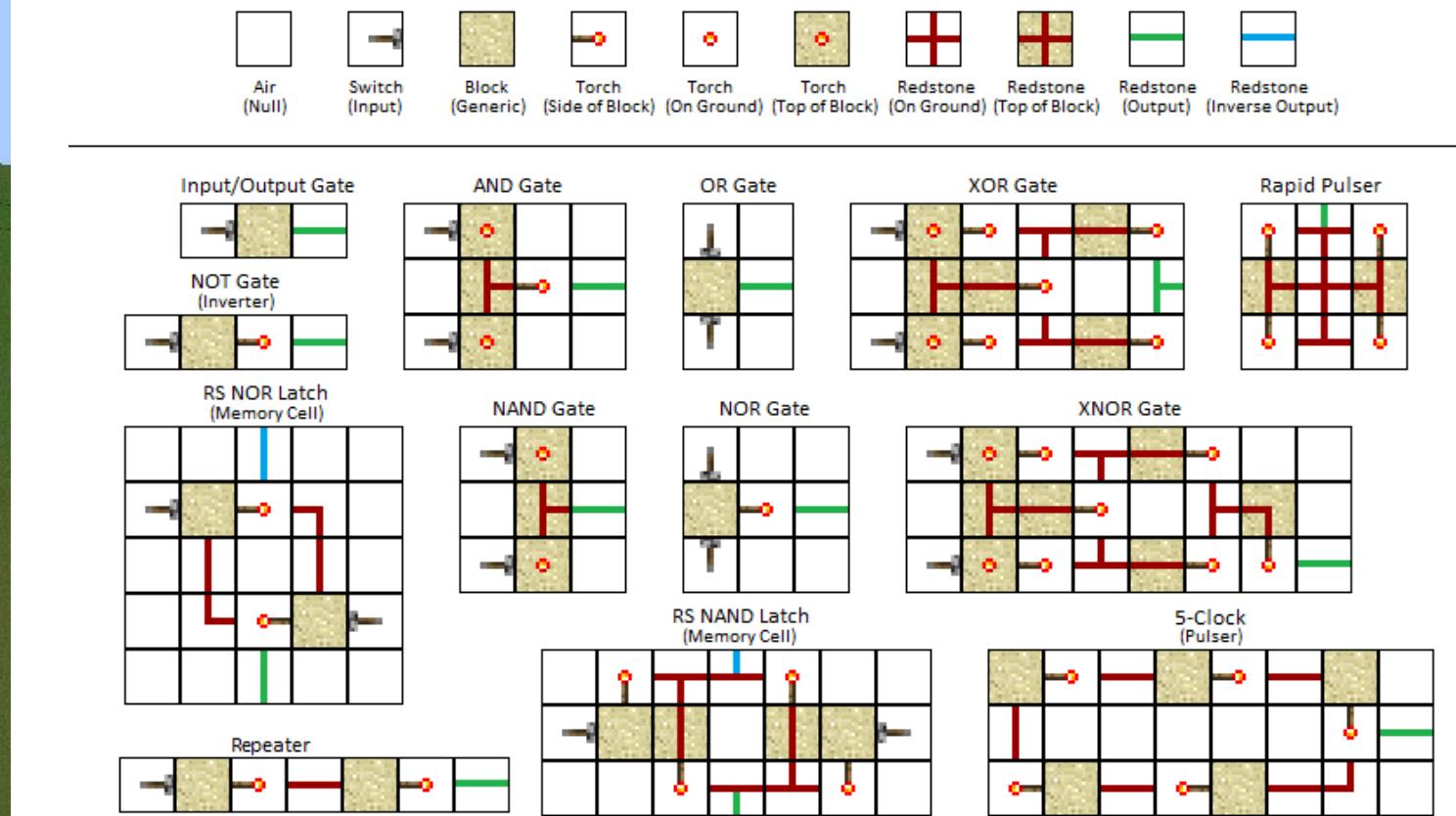
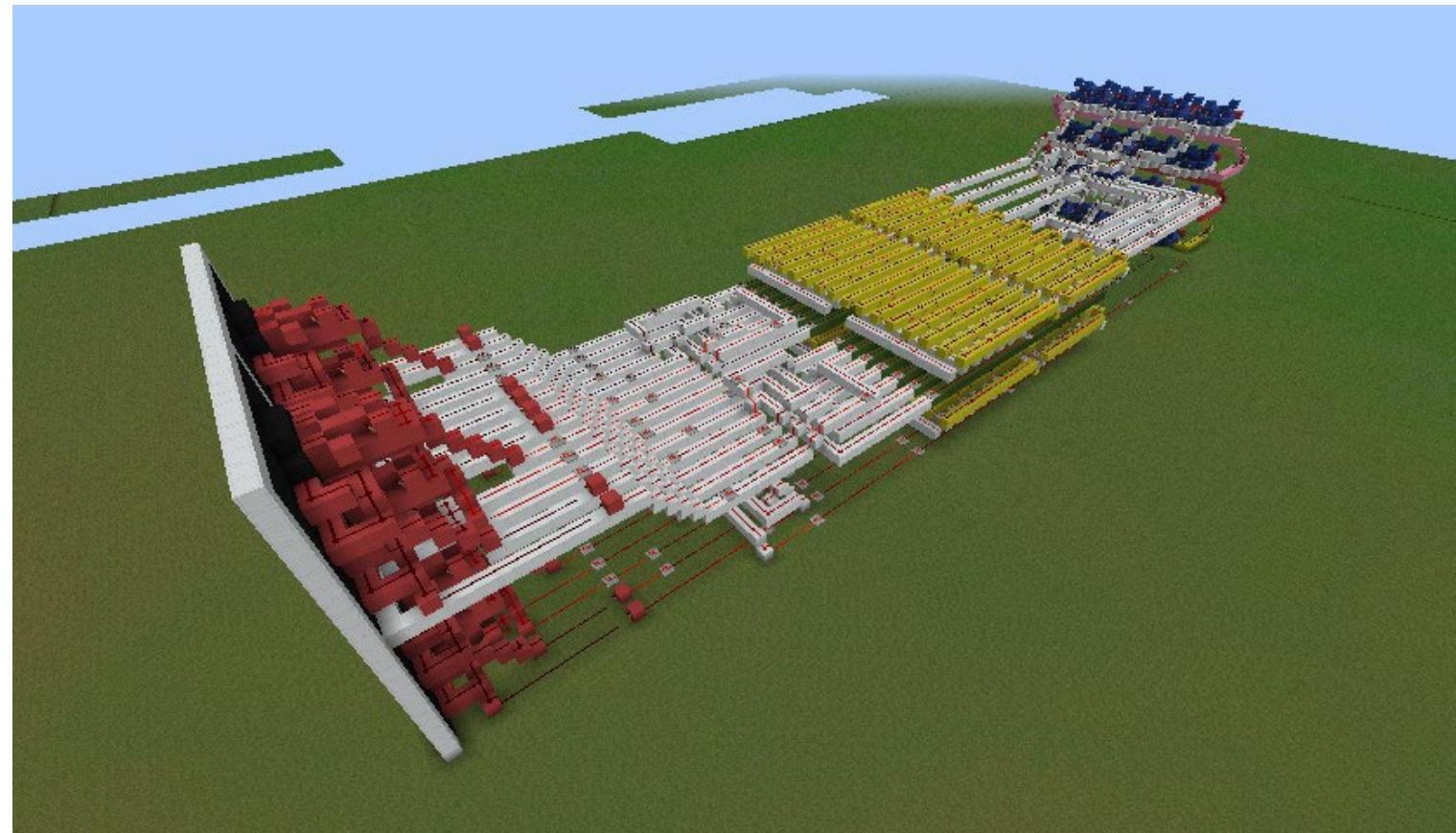
1993-2011



Jeff Tunnell Productions (Later PushButton Labs, Playdom, Disney Interactive)

# Minecraft

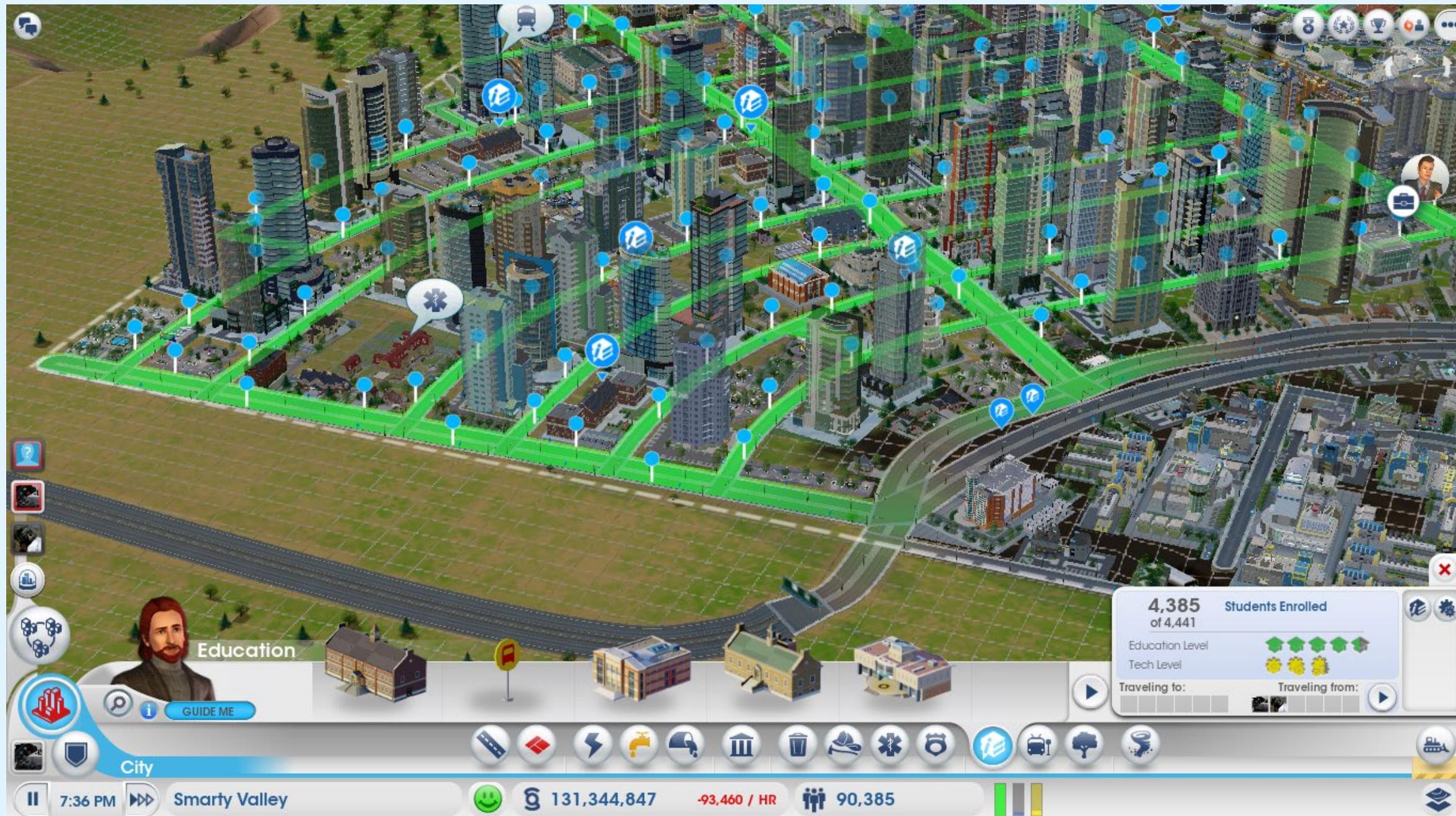
2009-Today



Markus Persson, Jens Bergensten at Mojang

# SimCity (Series)

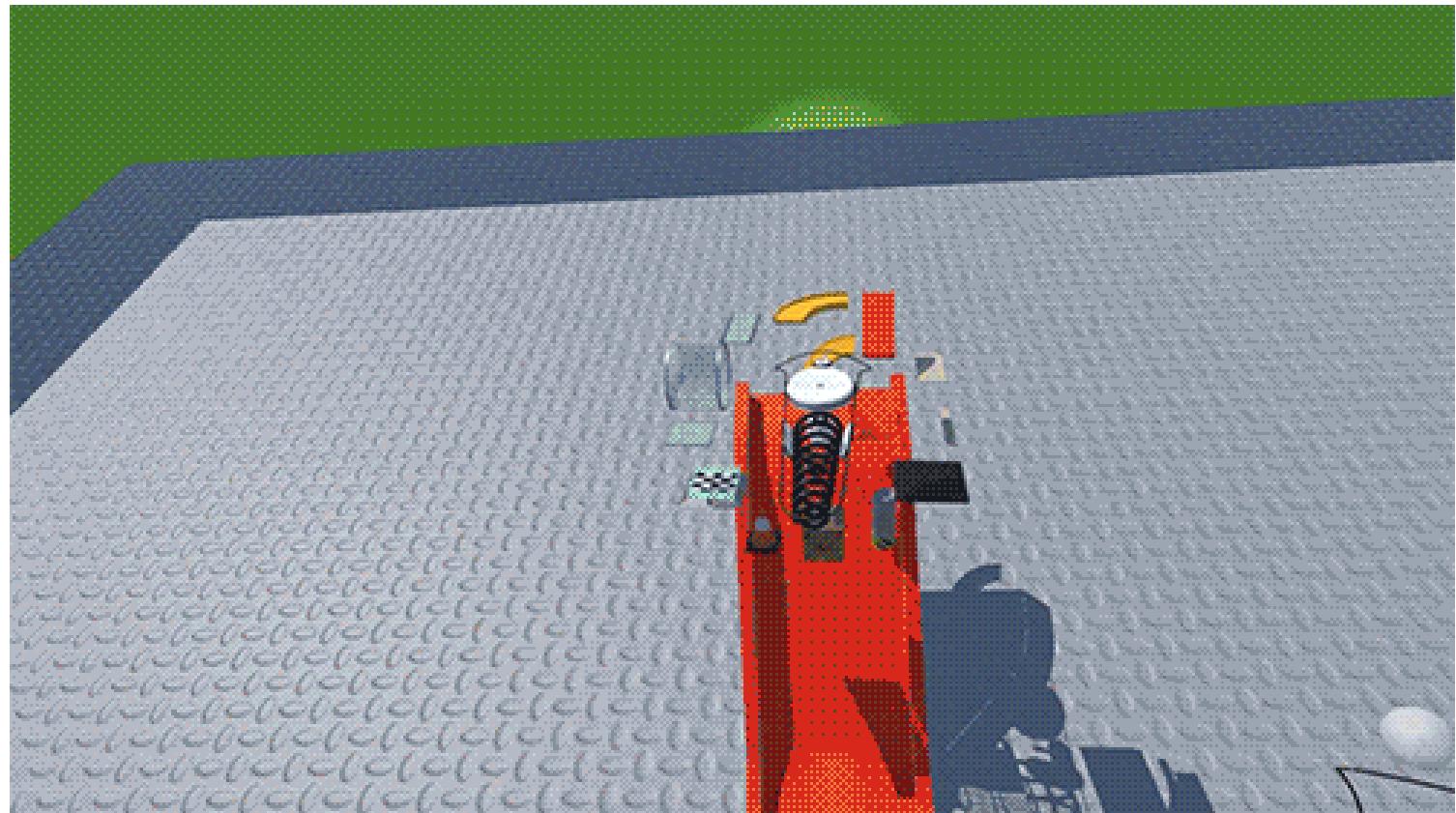
1989-2014



Various

# Tiny Wheels

2017



Robbie Tilton

# Dreams

2018



Media Molecule

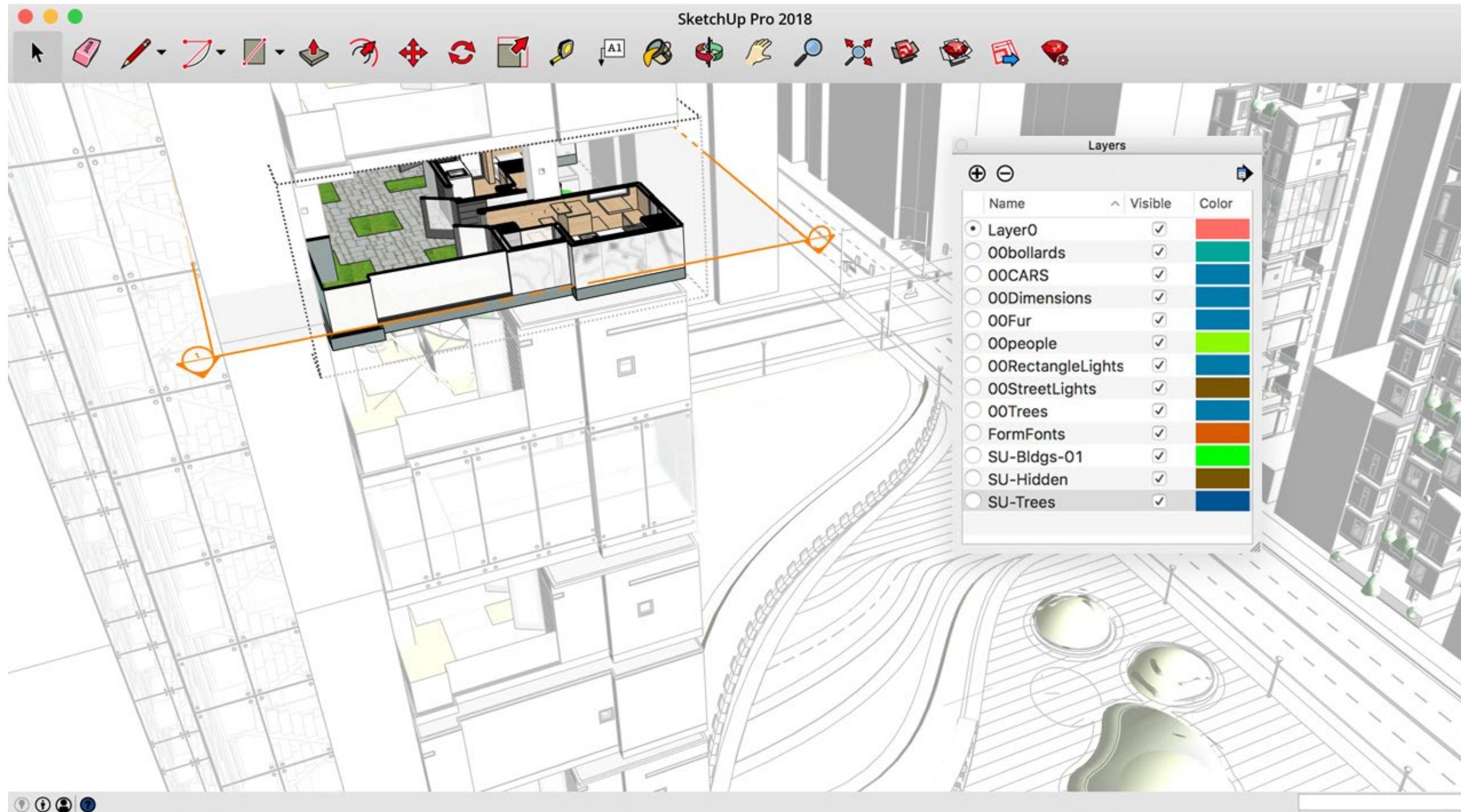
# Remaking Reality

## Measure, Simulate, Predict & Act

# Reality Design

# SketchUp

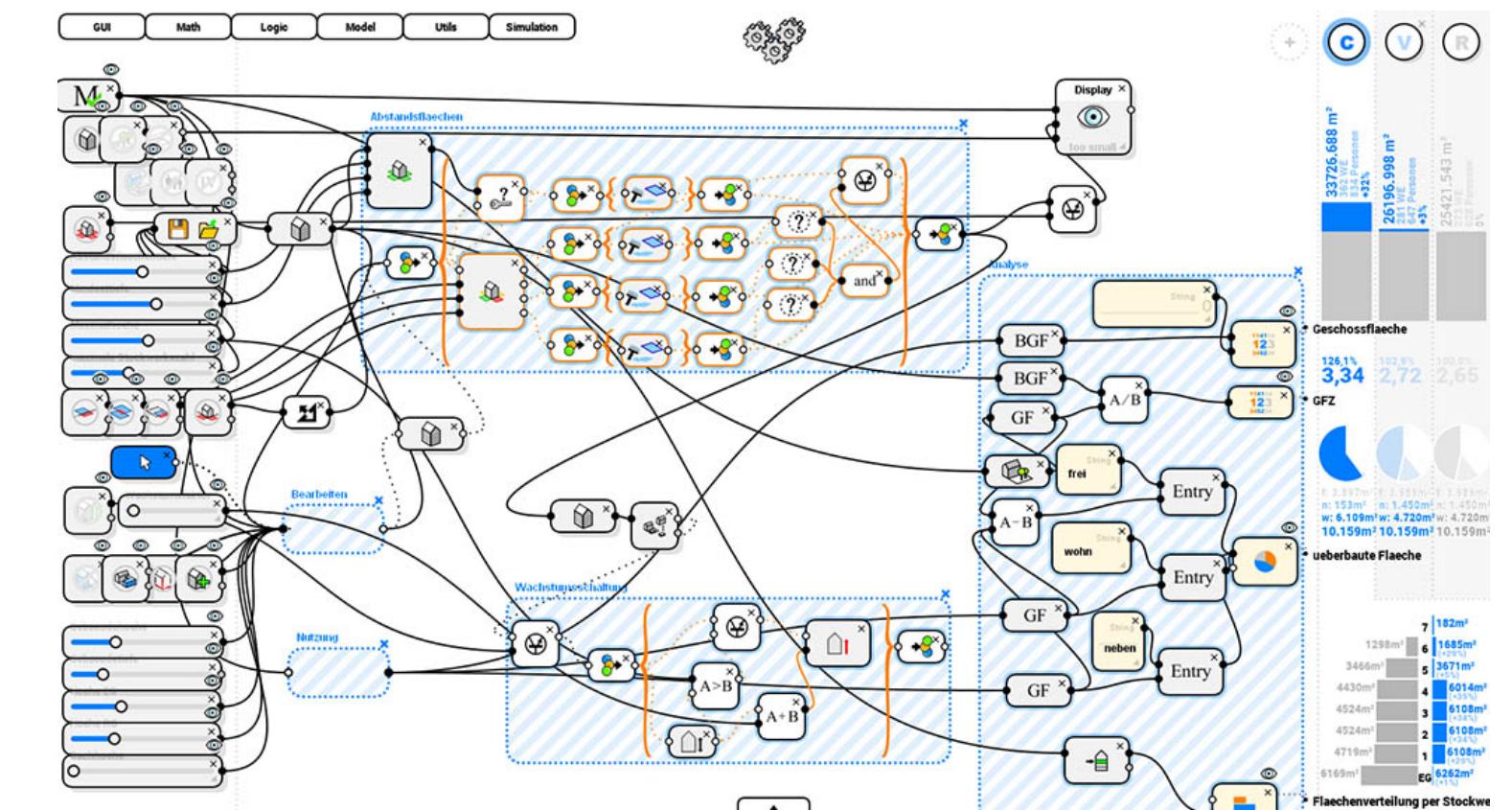
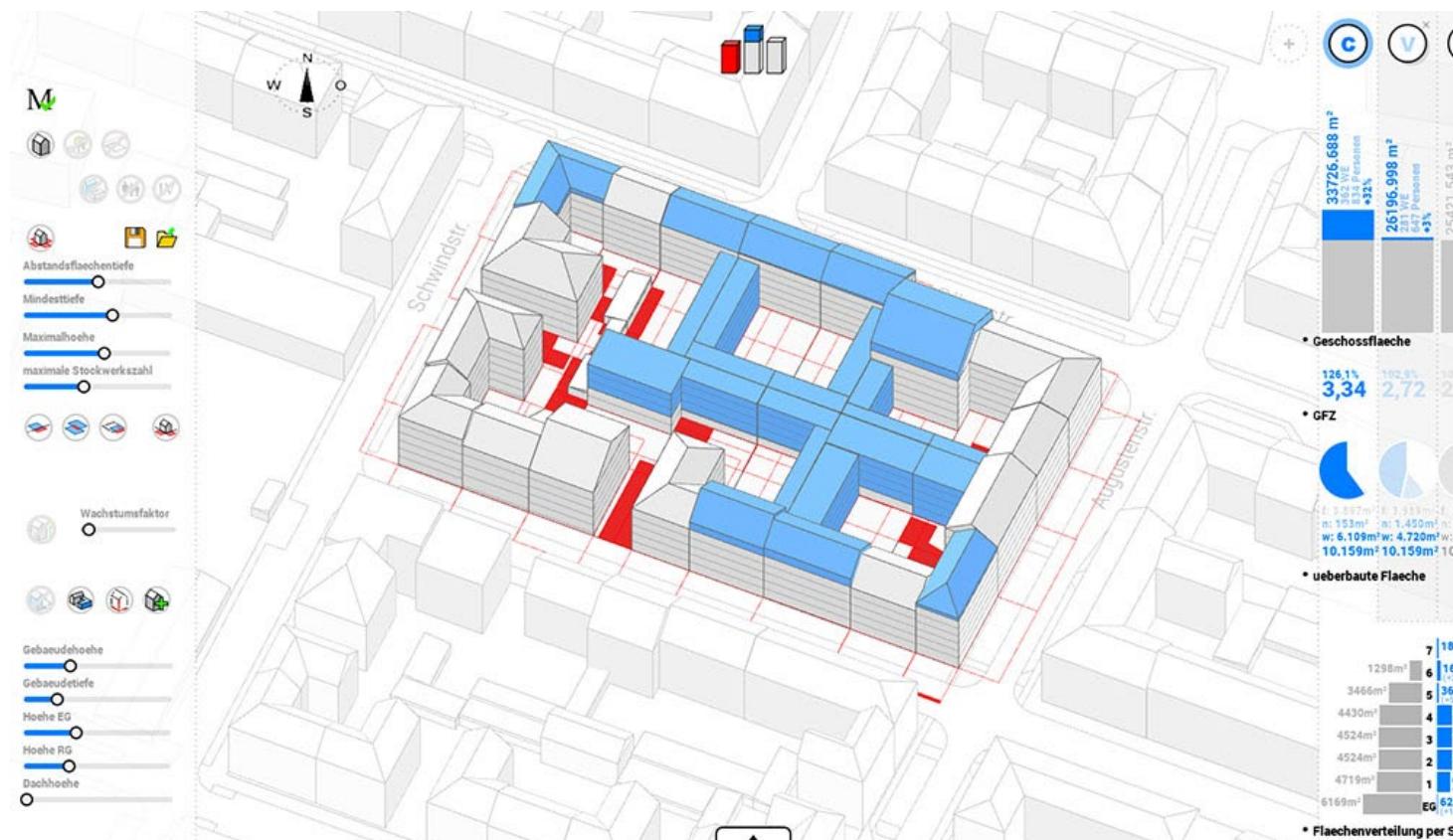
2000-Today



Brad Schell and Joe Esch for @Last Software (Later Google, Trimble Inc)

# Urban Strategy Playground (Suite)

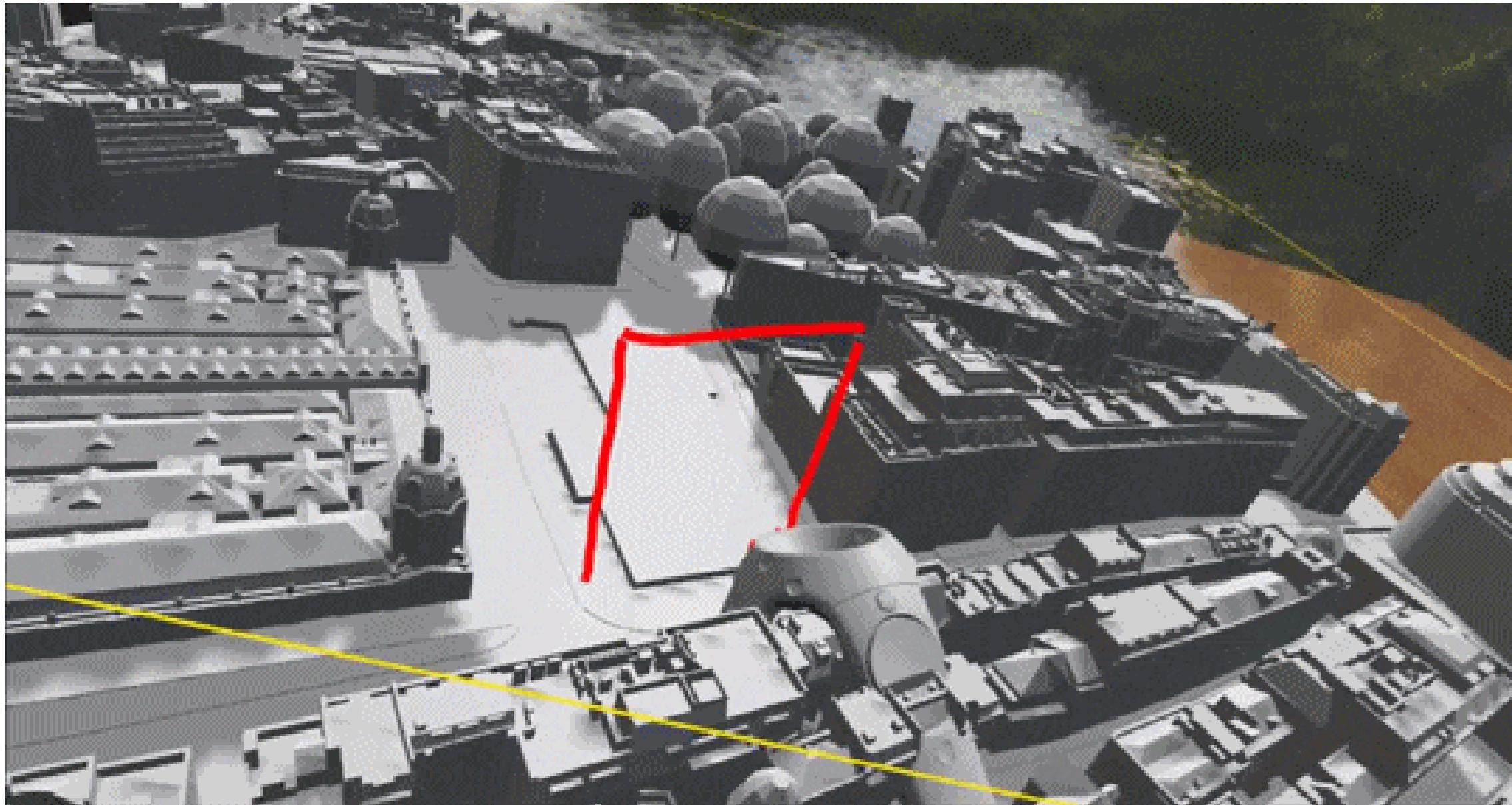
2013-Today



USP Research Group in the Department of Architektur at Technische Universität München

# DesignSpace

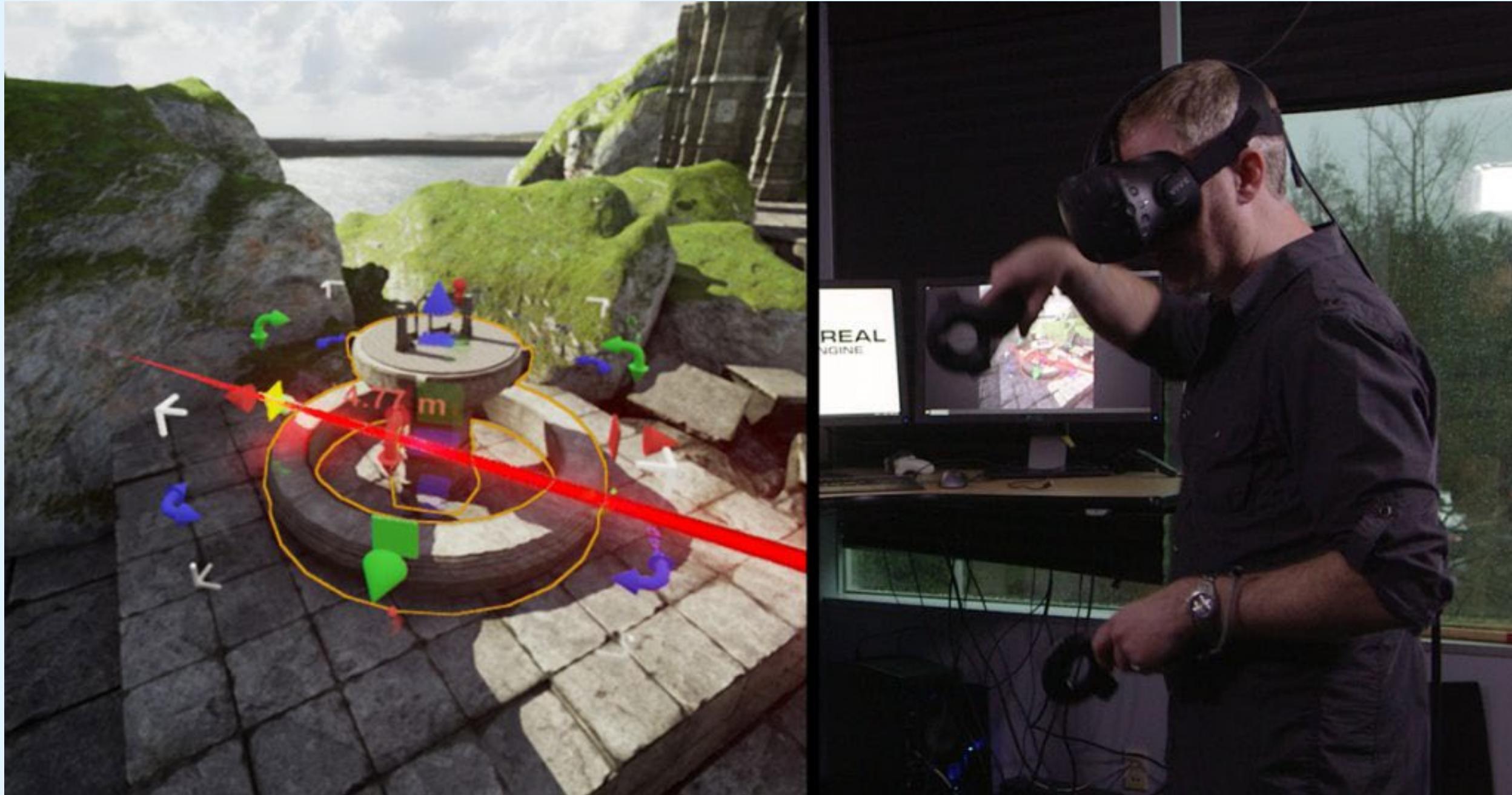
2016



Thomas Van Bouwe

# VR Editor for Unreal Engine

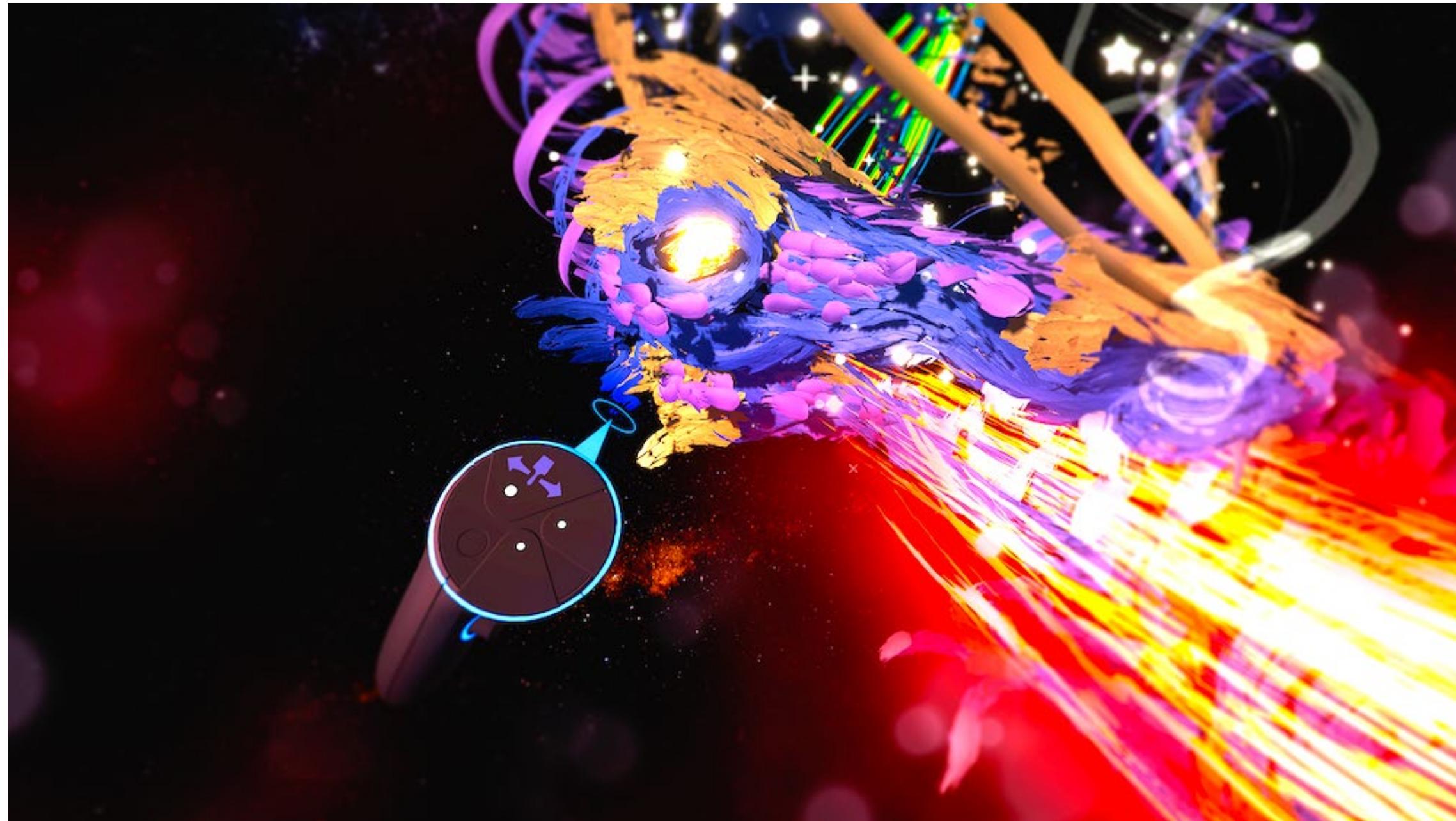
2016-Today



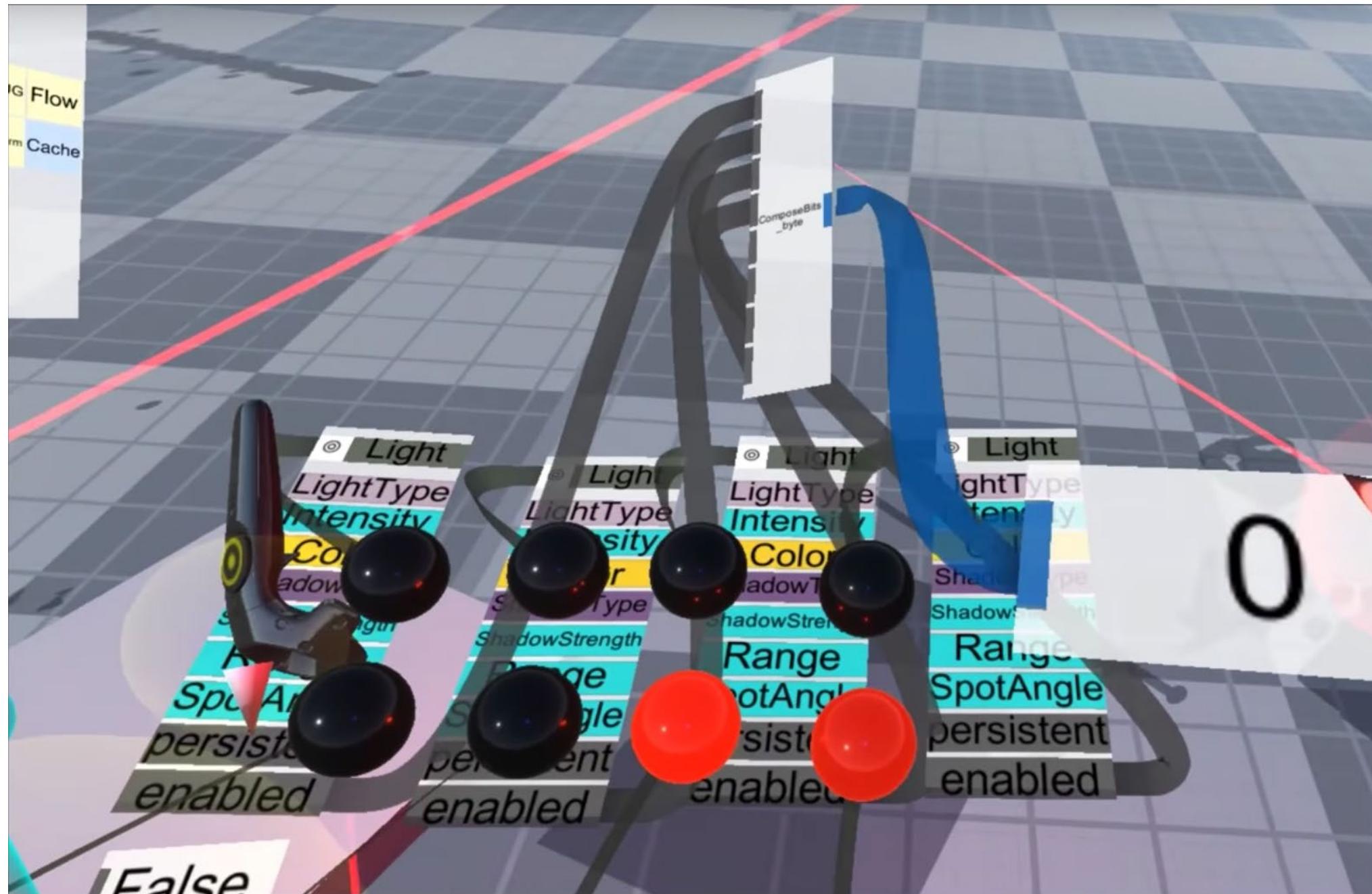
Epic Games

# TiltBrush

2016-Today



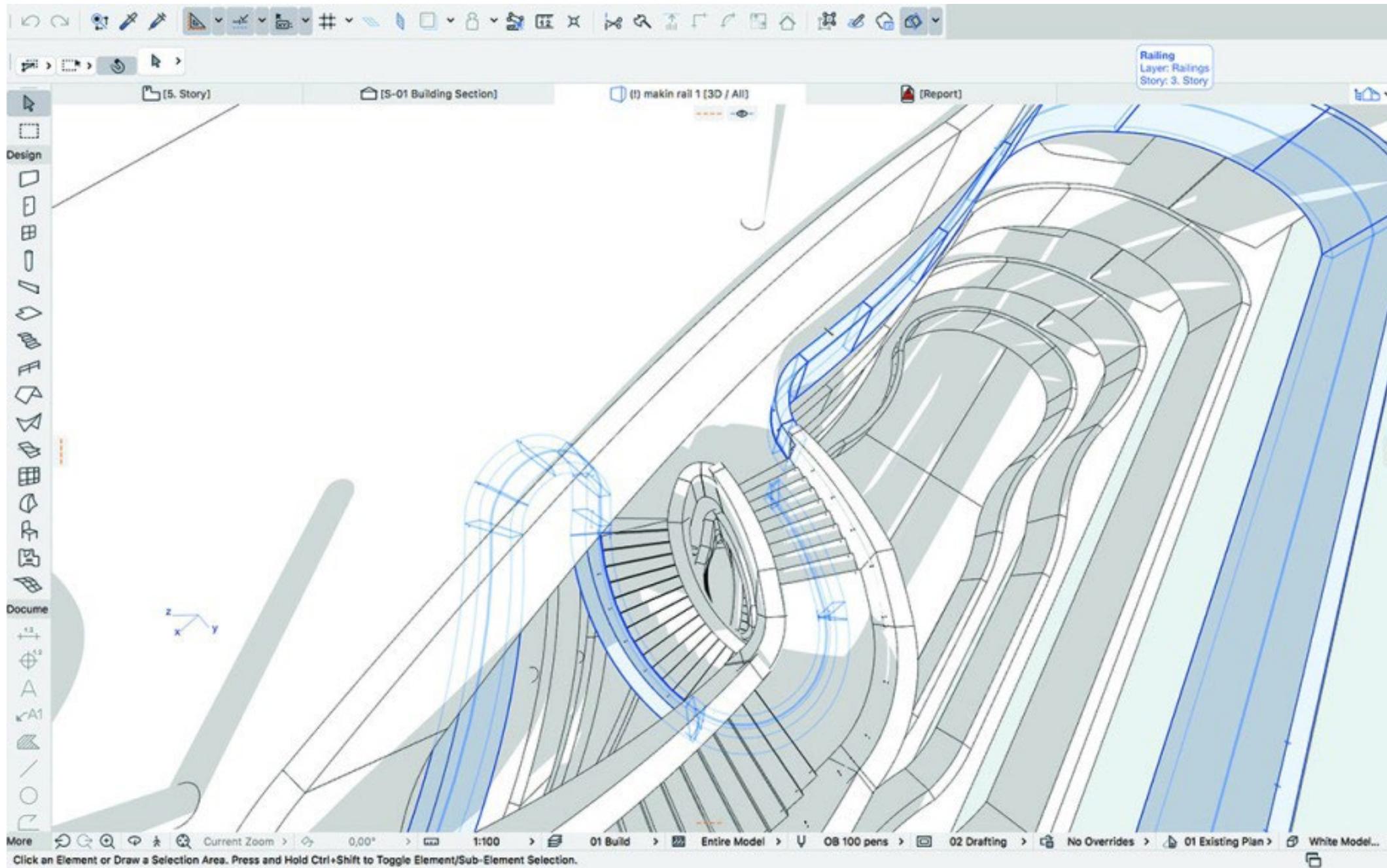
Google



Tomáš Mariancík

# ArchiCAD (Predictive Design)

2018-Today



Graphisoft SE (Part of the Nemetschek Group)

# Local Simulation

(i.e. Urban, Regional, etc)

# Gotham

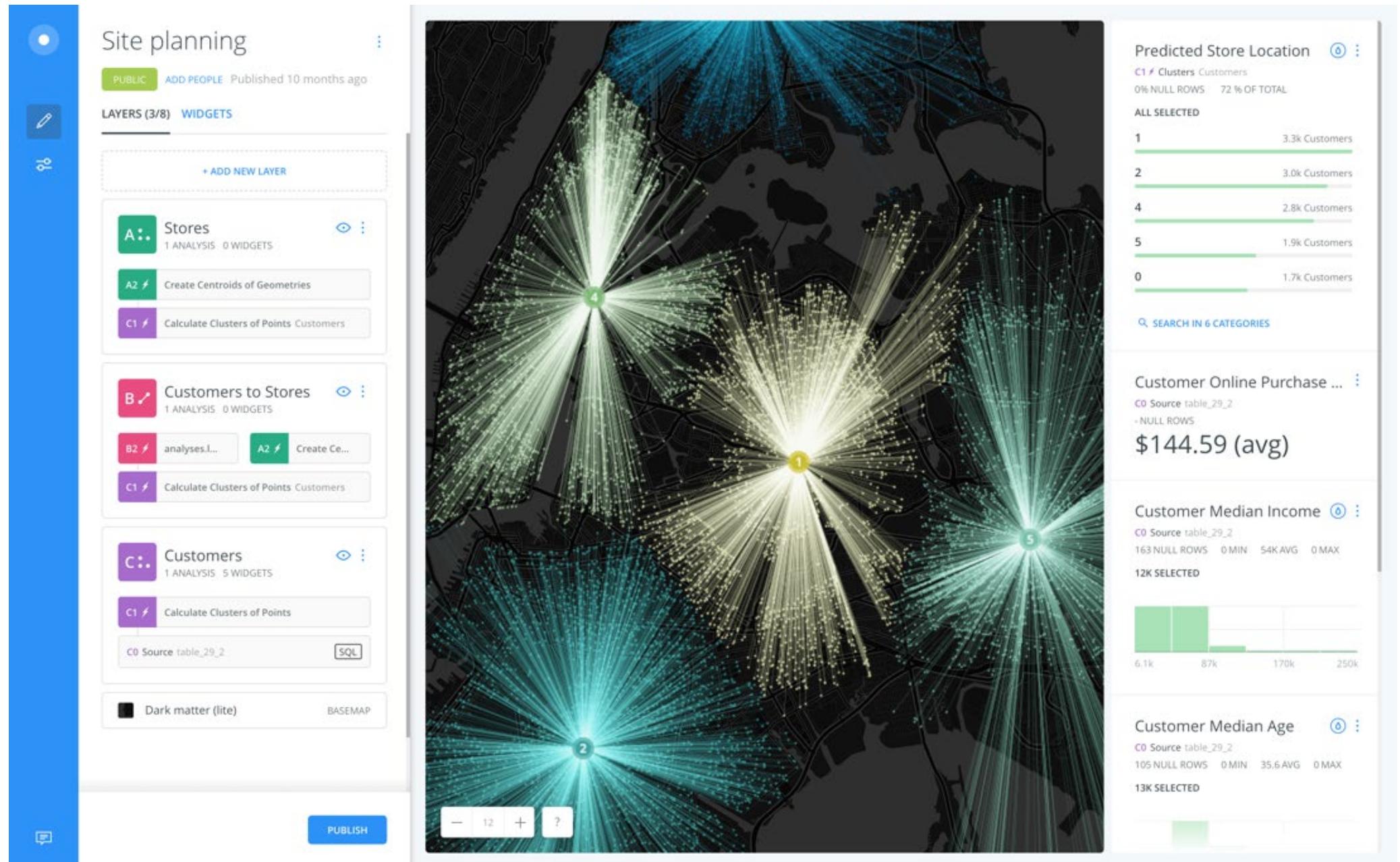
2004-Today



Palantir

# CartoDB

2011-Today



CARTO

# MARK43 Computer Aided Dispatch

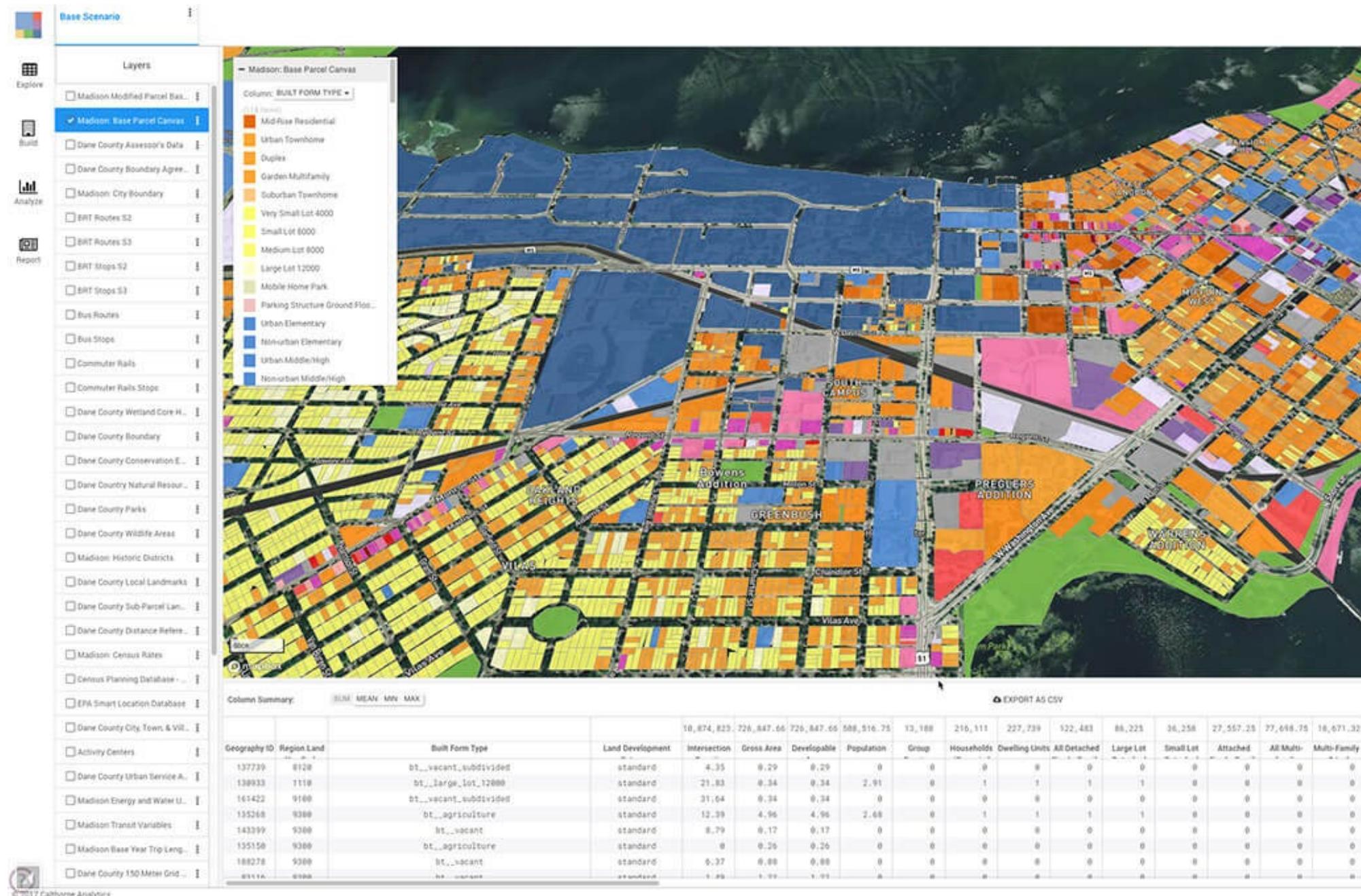
2012-Today



Mark43 Inc

# UrbanFootprint

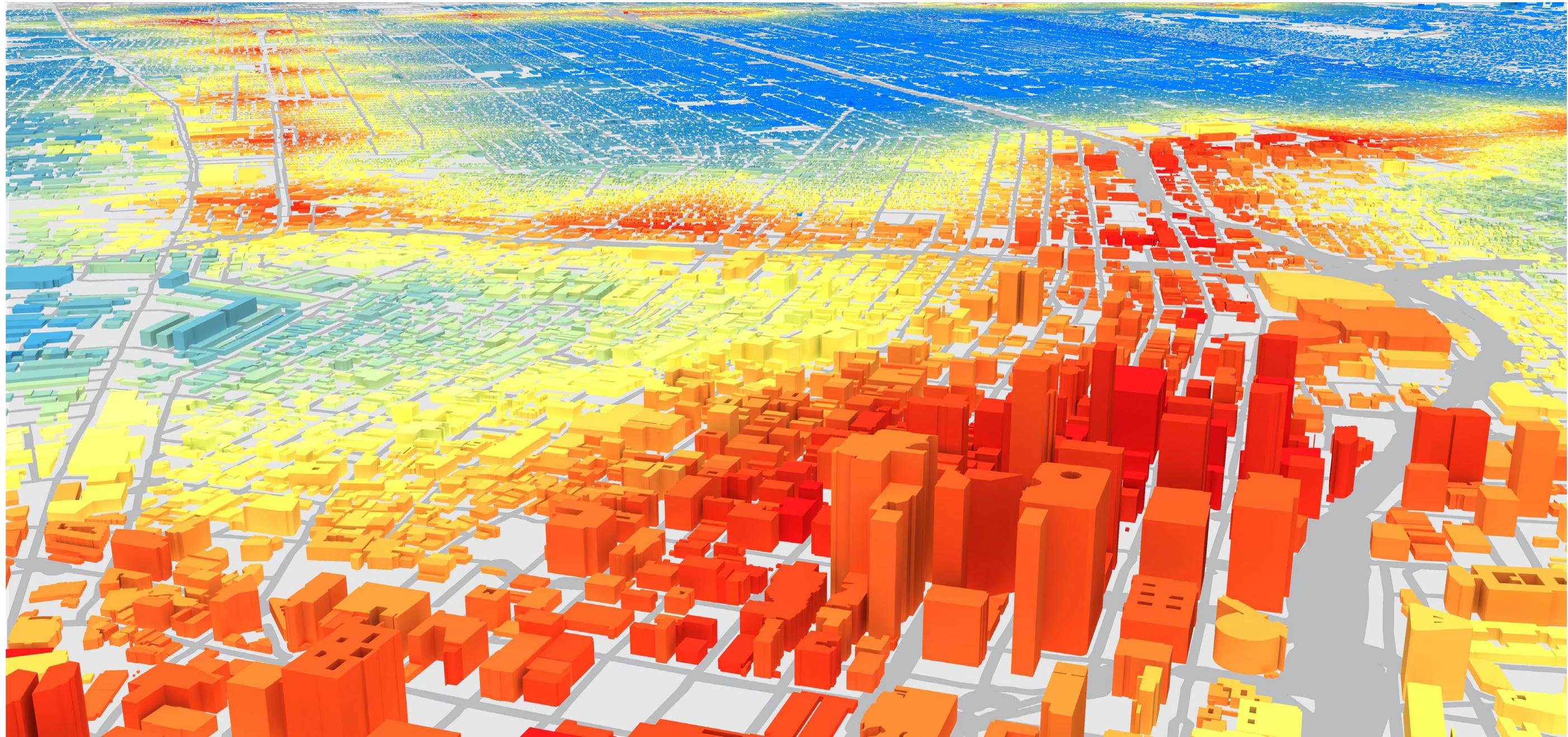
2014-Today



Joe Distefano, Peter Calthorpe for Calthorpe Analytics

# Urban Network Analysis Toolbox

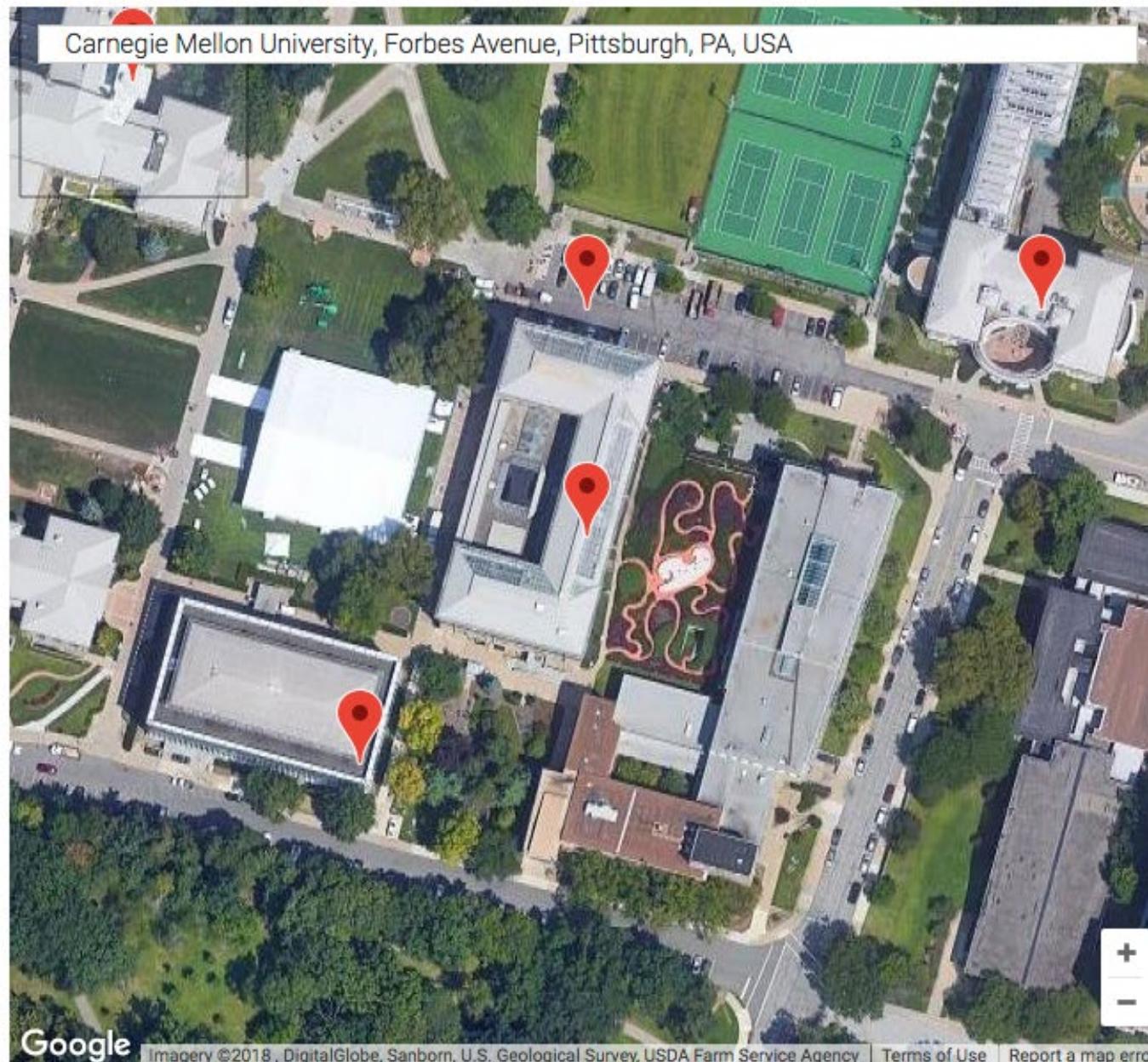
2015-Today



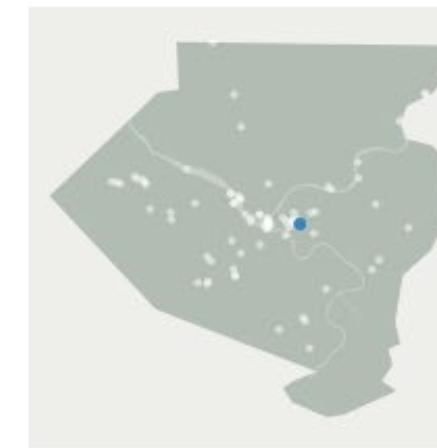
City Form Lab at the Harvard University Graduate School of Design

# TerraPattern

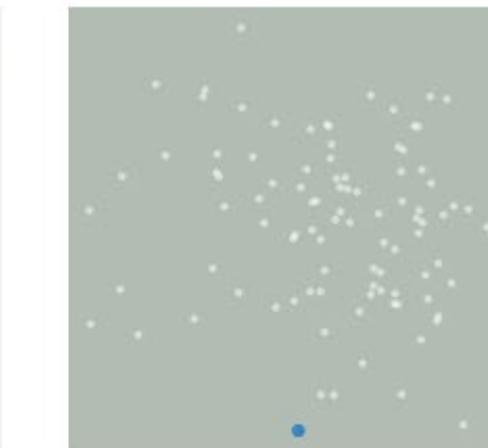
2016



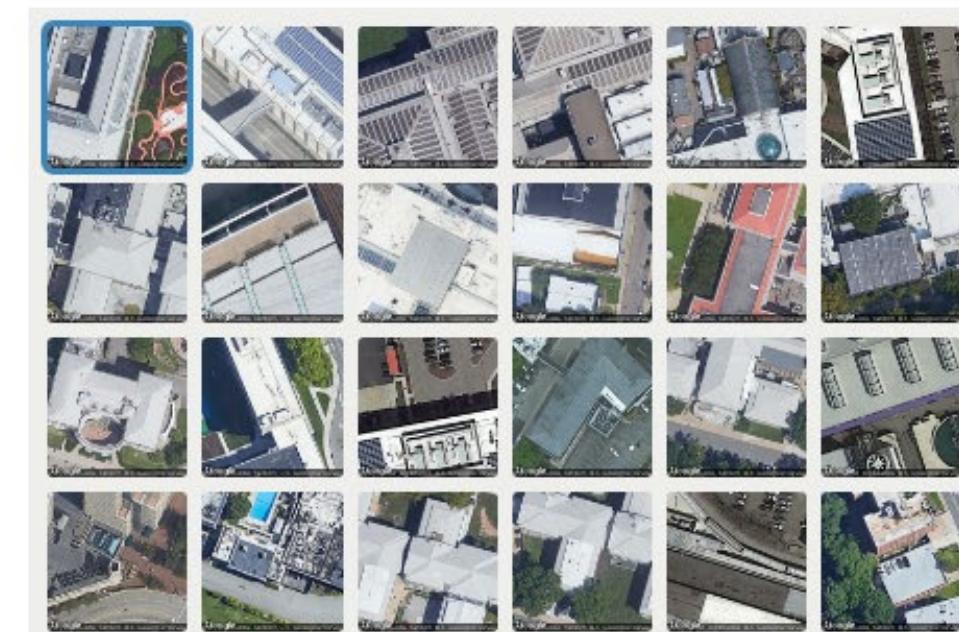
Geographical Plot



Similarity Plot



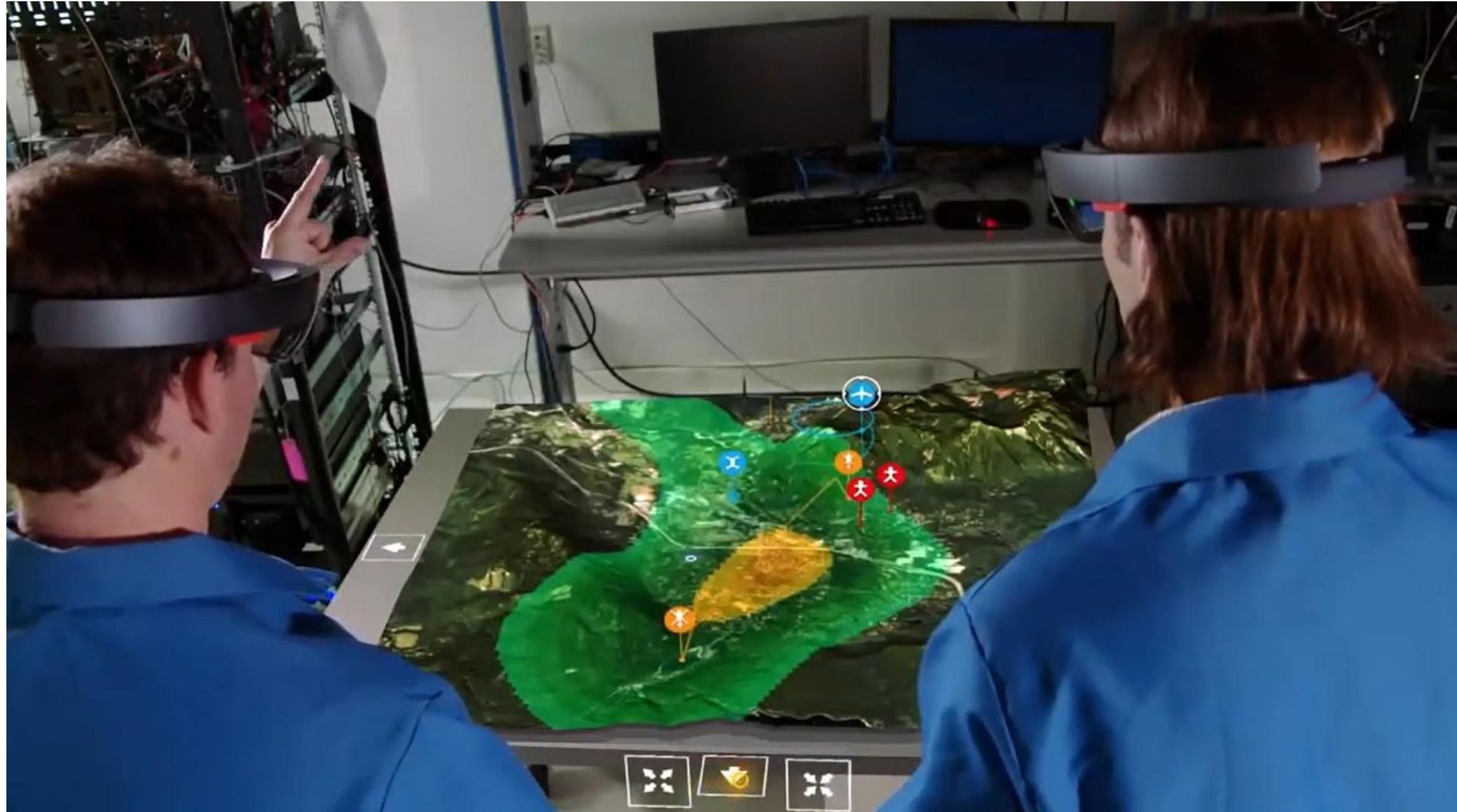
Search Results



Golan Levin, David Newbury, Kyle McDonald, Irene Alvarado, Aman Tiwari and Manzil Zaheer  
at the Frank-Ratchye STUDIO for Creative Inquiry at Carnegie Mellon University

# Hololens App: Forest Fire Fighting

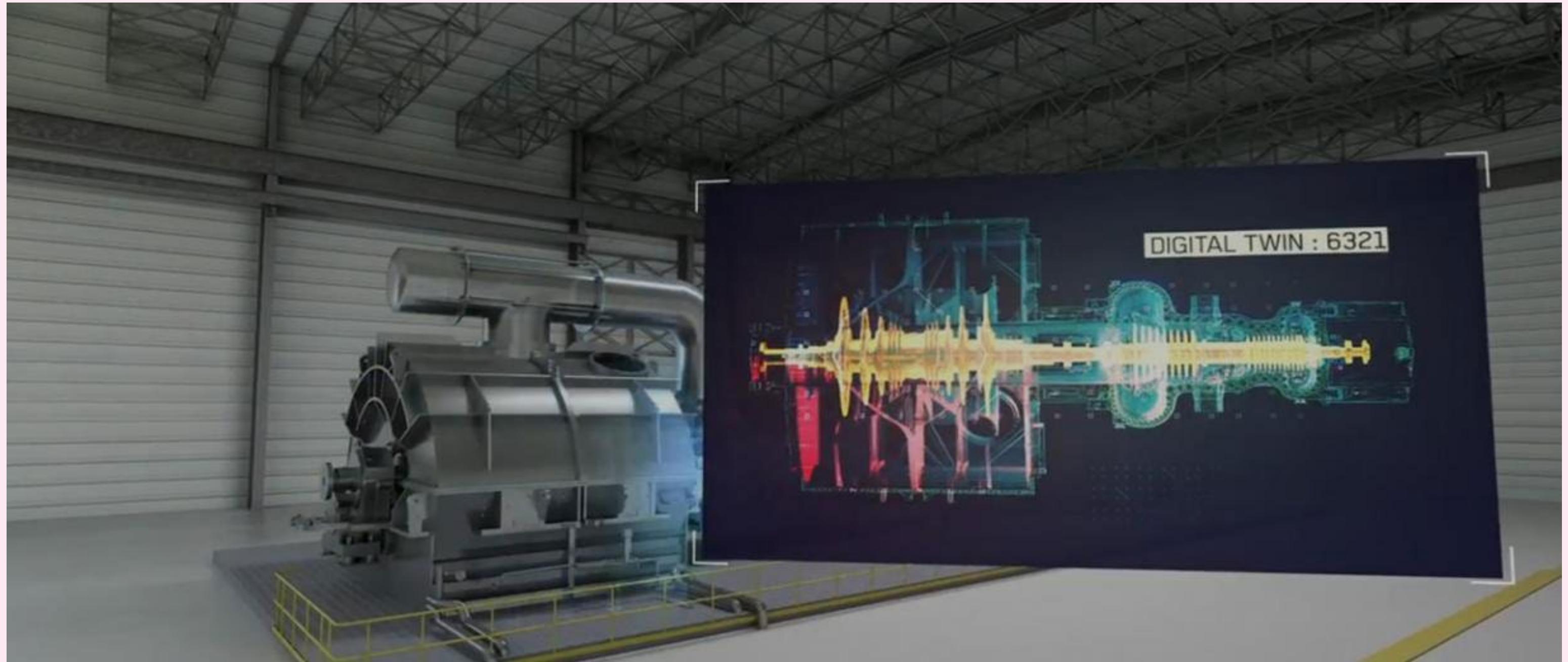
2016



Boeing

# GE Predix (Digital Twin)

2016-Today



General Electric and ANSYS

# Hololens App: Air Force Asset Planning

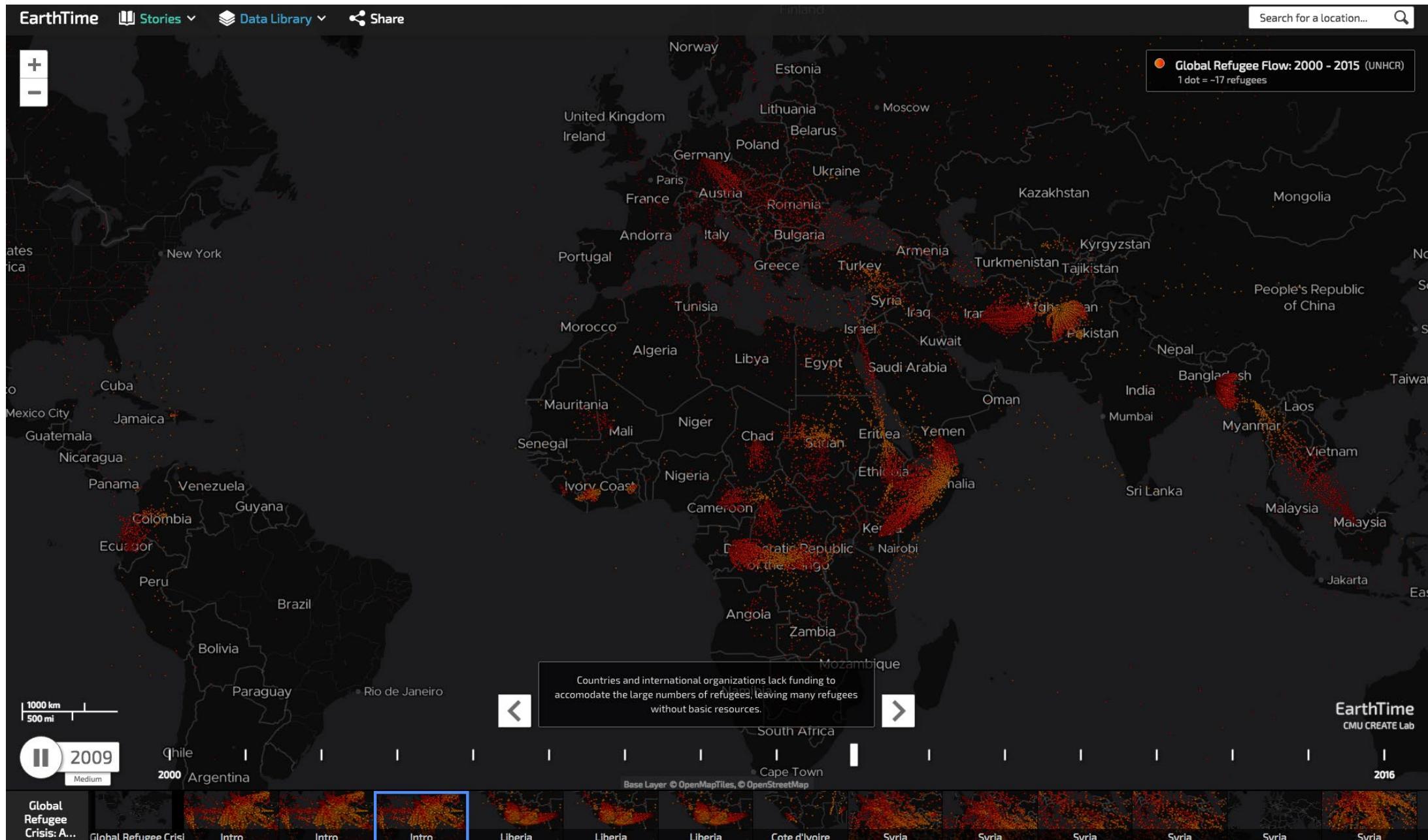
2017



Royal Australian Air Force

# EarthTime

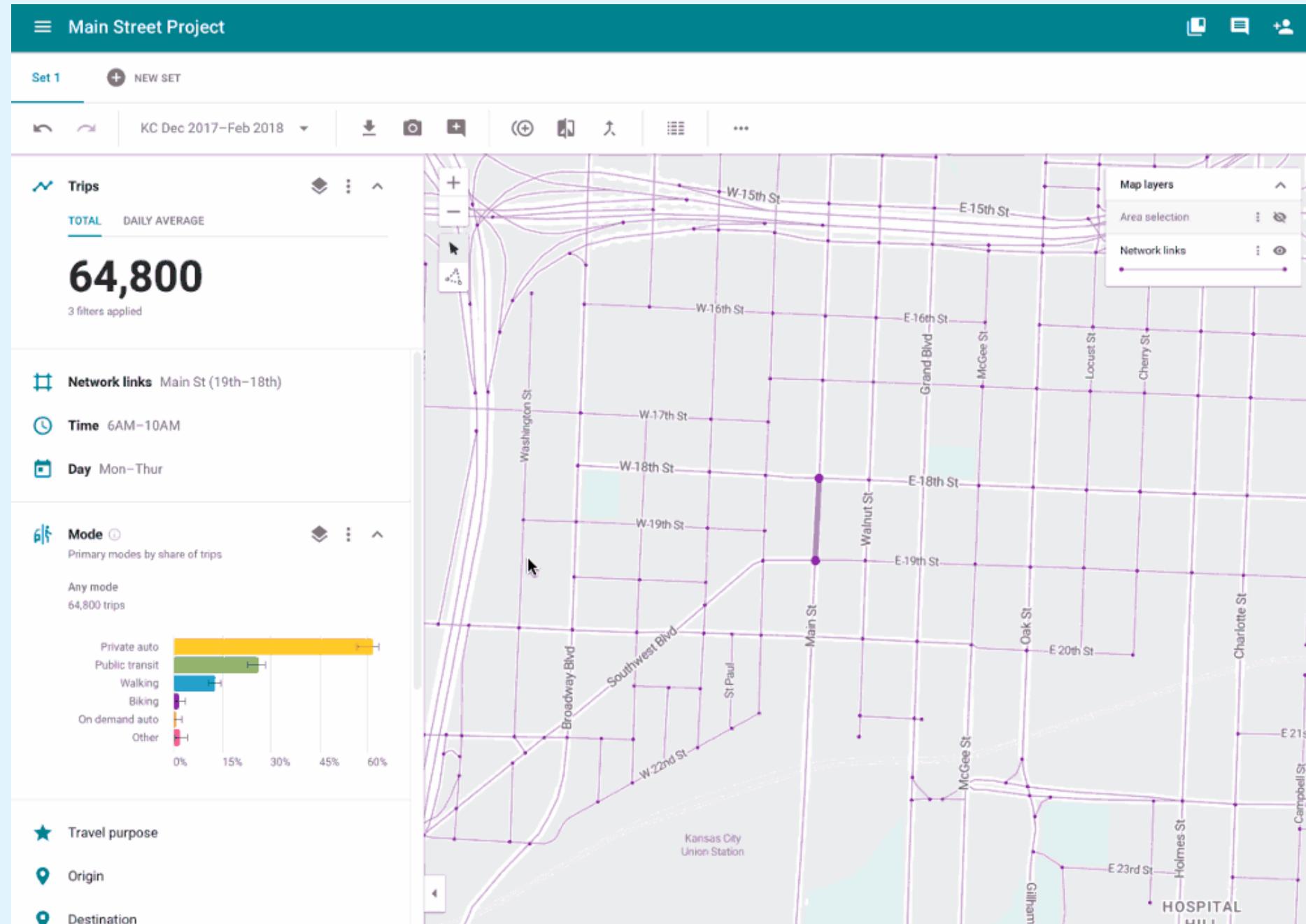
2018



CREATE (Community Robotics, Education and Technology Empowerment) Lab at Carnegie Mellon University

# Replica

2018-Today



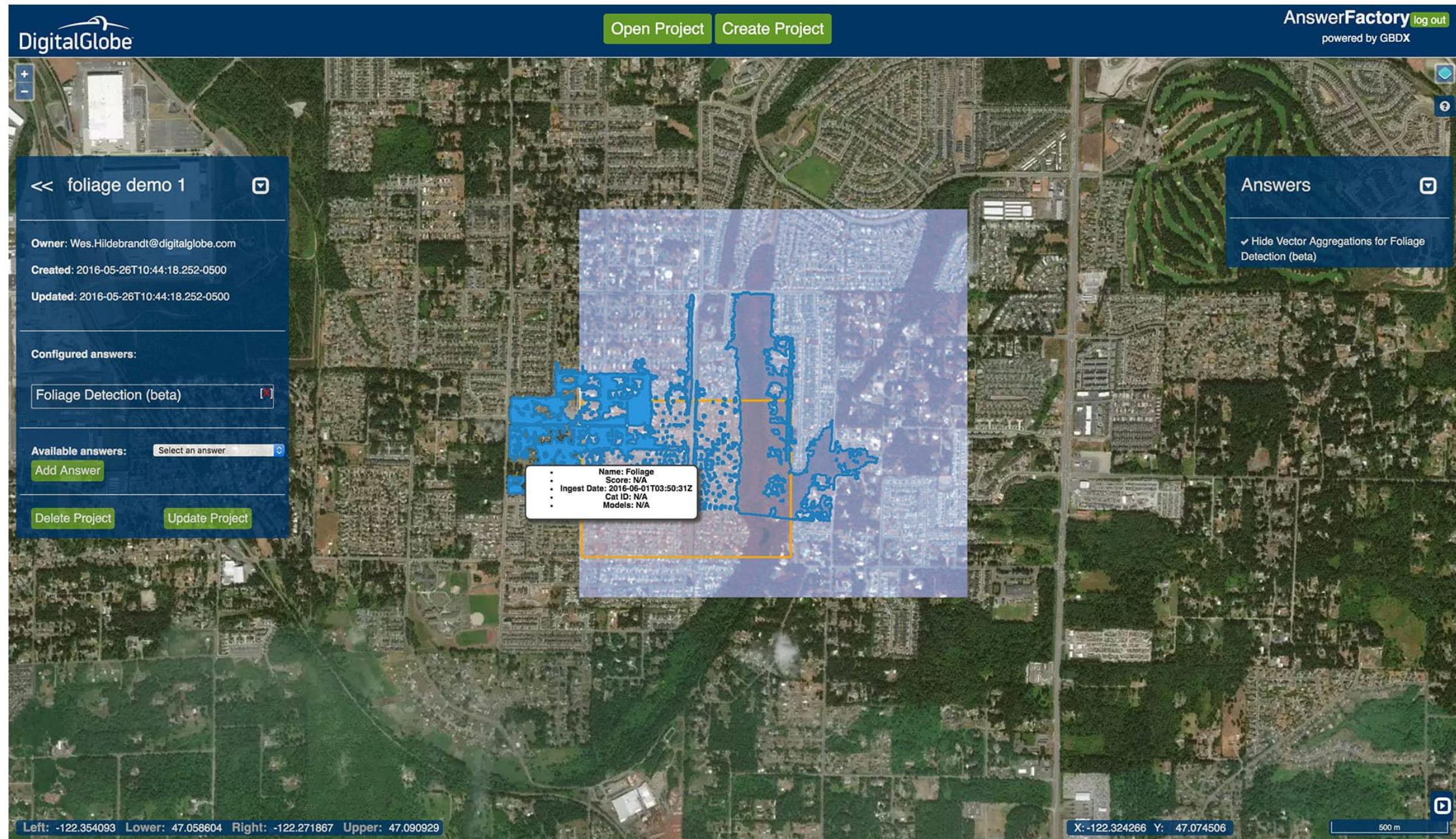
Sidewalk Labs (Part of Alphabet)

# Global View

‘Macrosopes’

# DigitalGlobe AnswerFactory

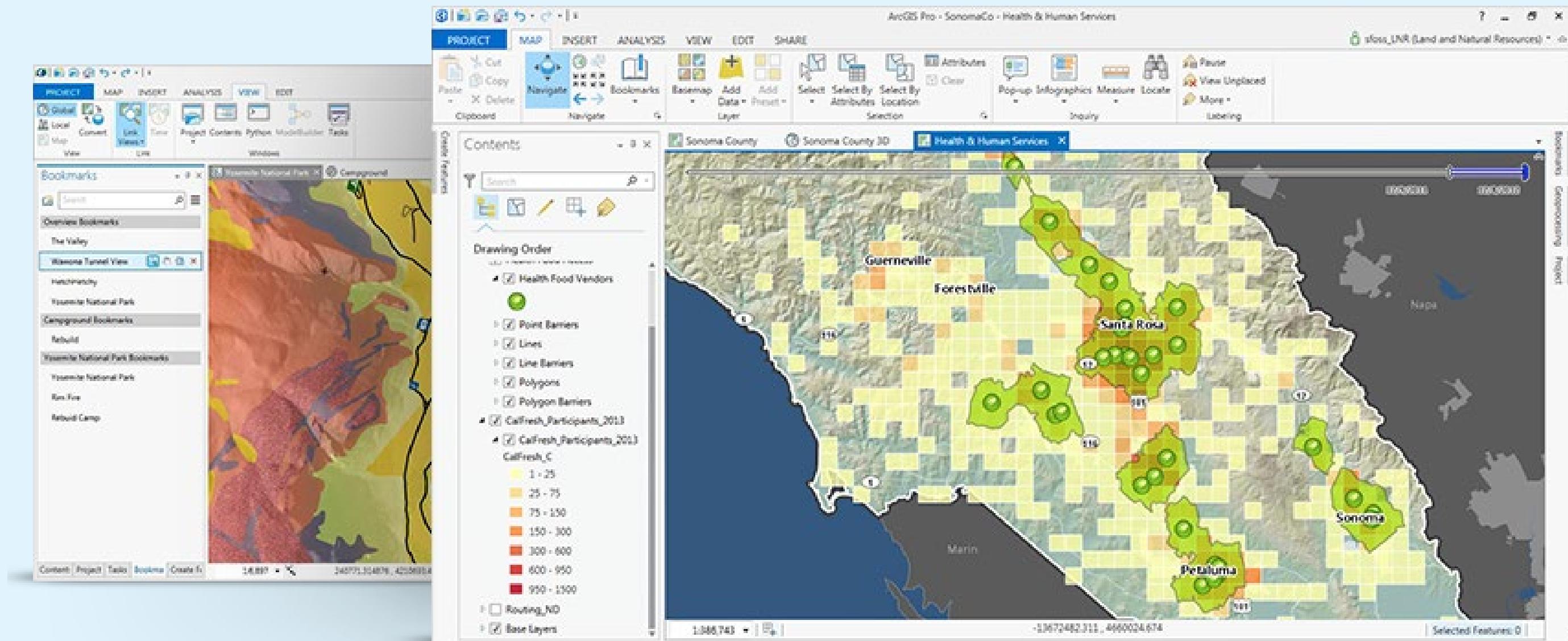
1992-Today



DigitalGlobe (Part of MAXAR)

# ArcGIS

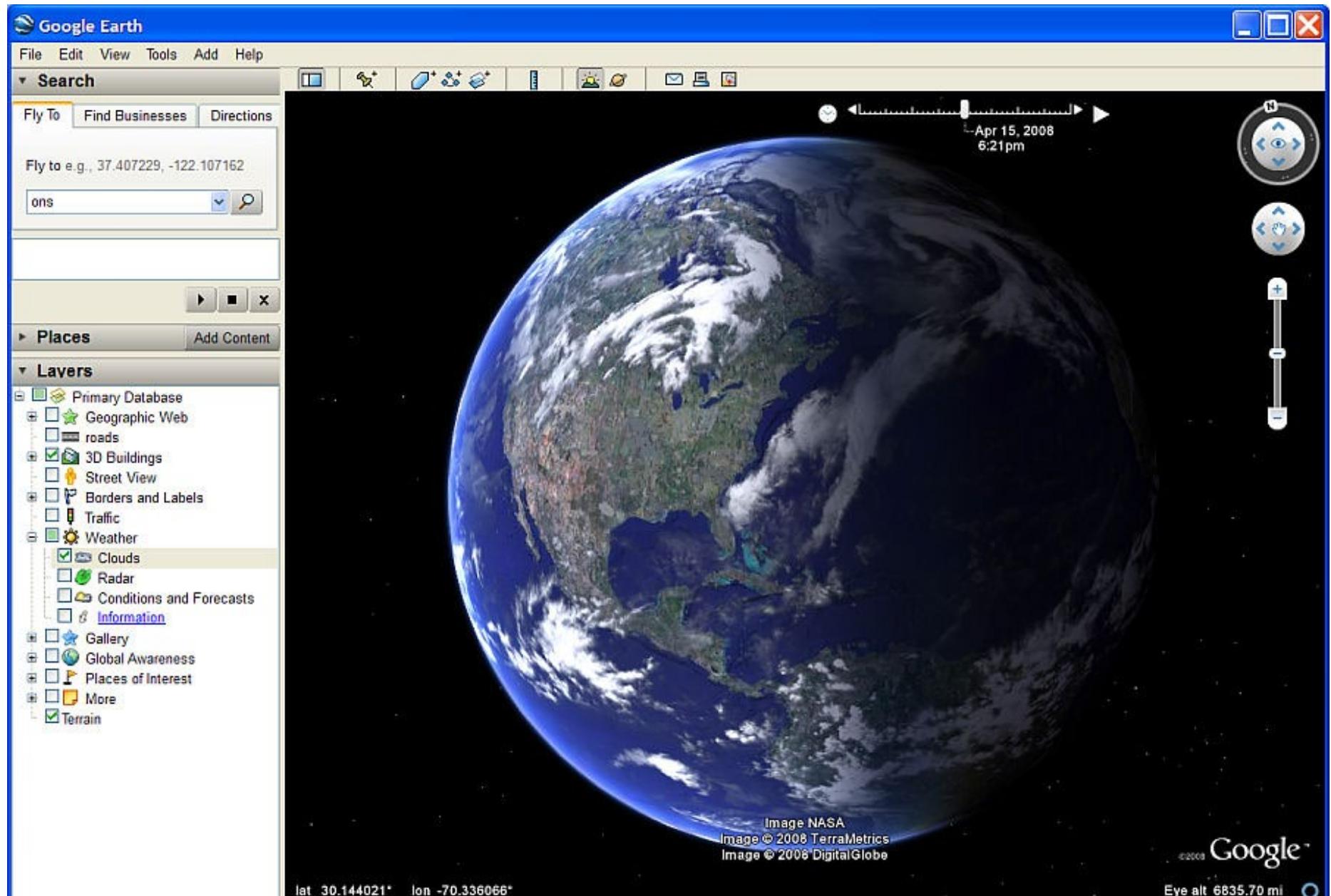
1999-Today



Esri

# Google Earth

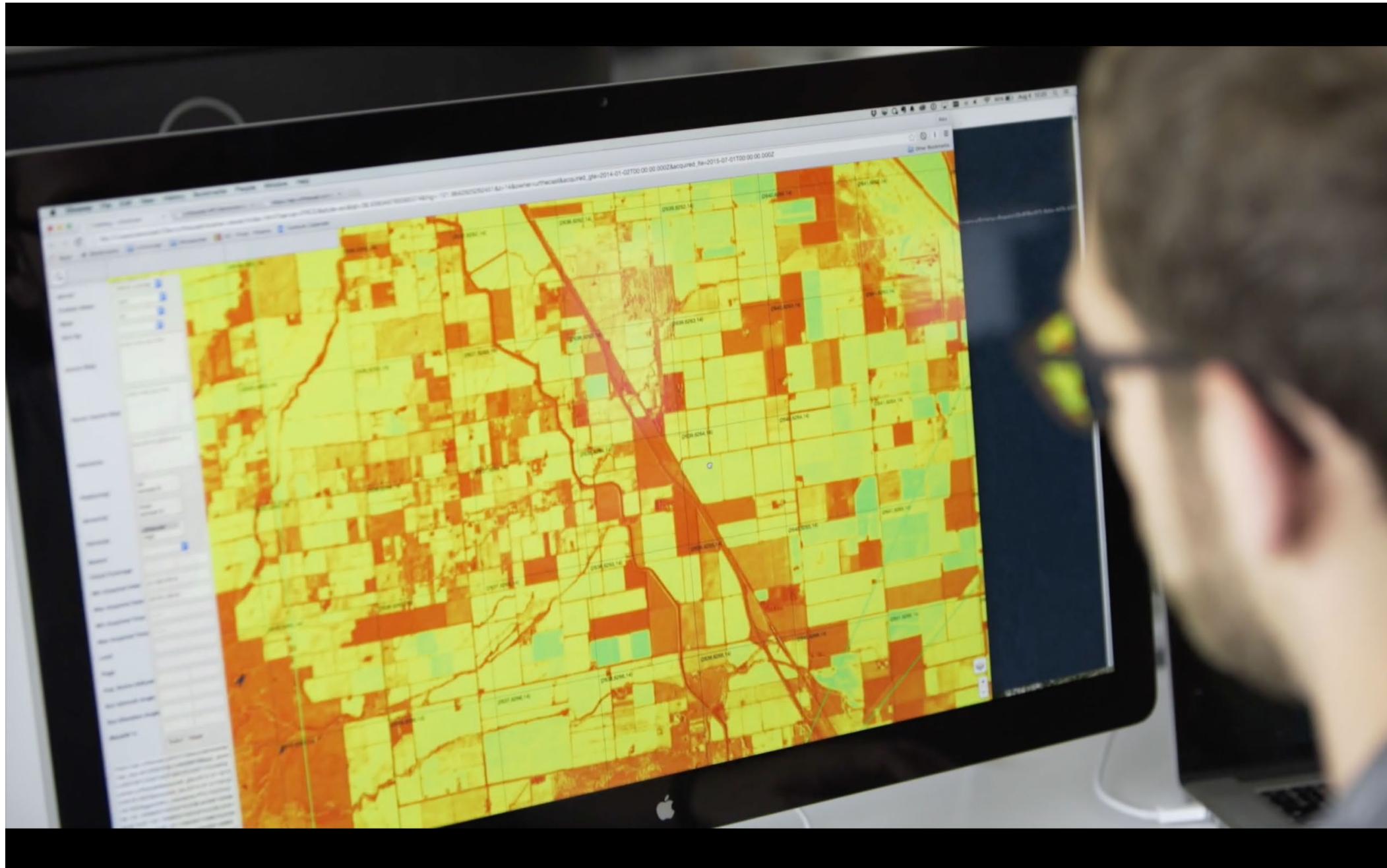
2001-Today



Intrinsic Graphics (Later Keyhole Inc, Google)

# UrtheCast

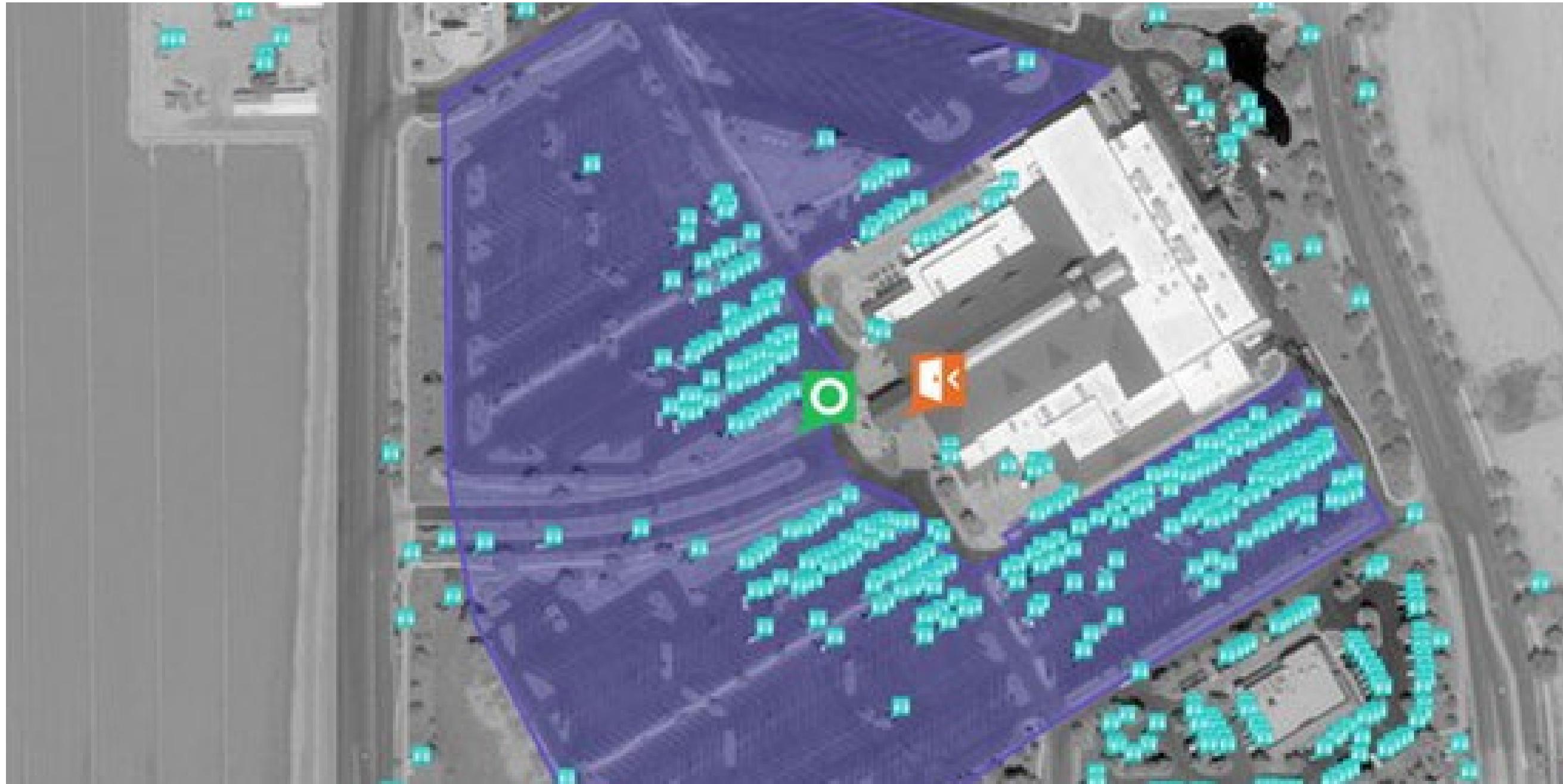
# 2004-Today



UrtheCast Corp

# Orbital Insight

2013-Today

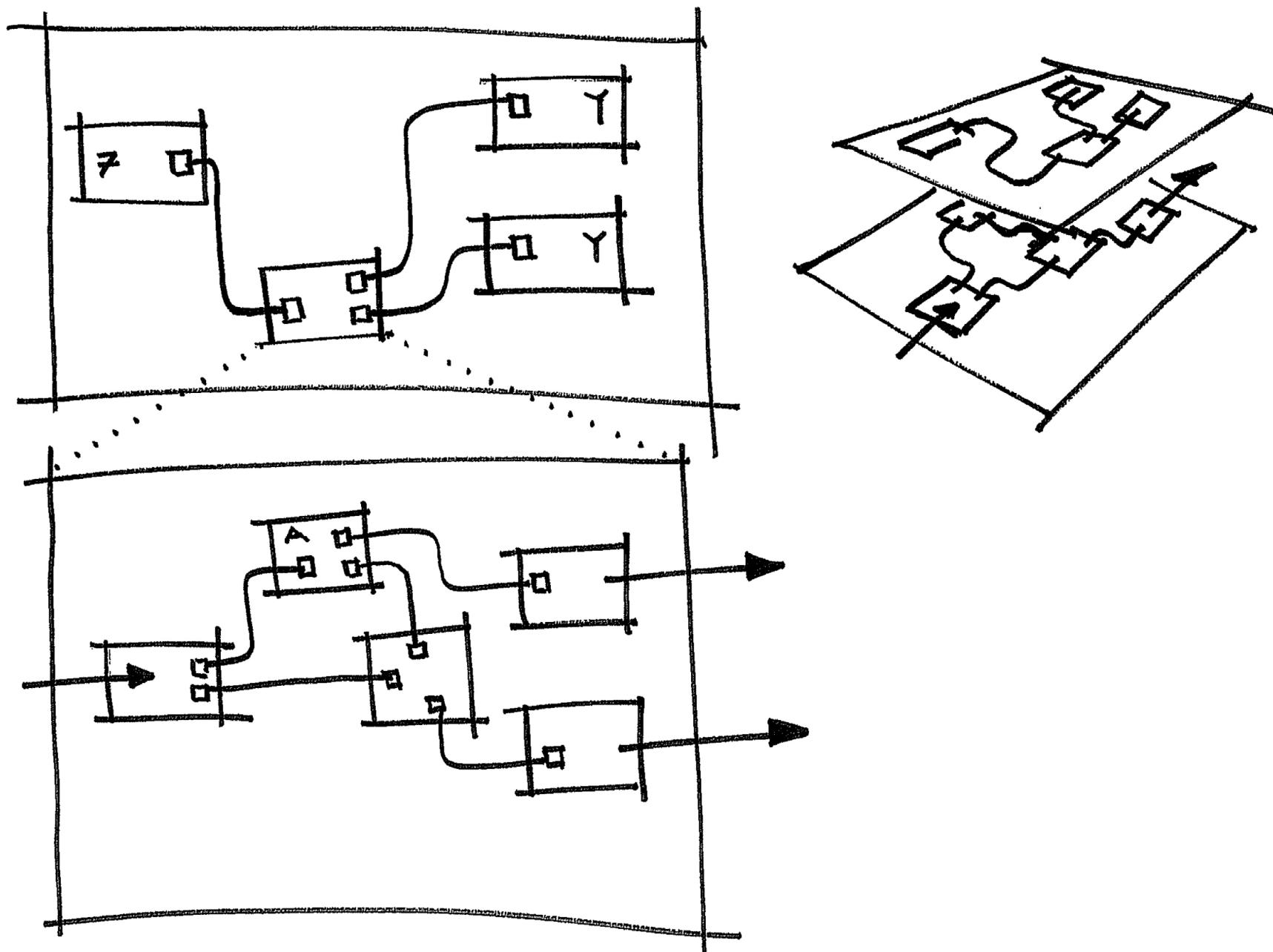


Orbital Insight Inc

# Design Patterns

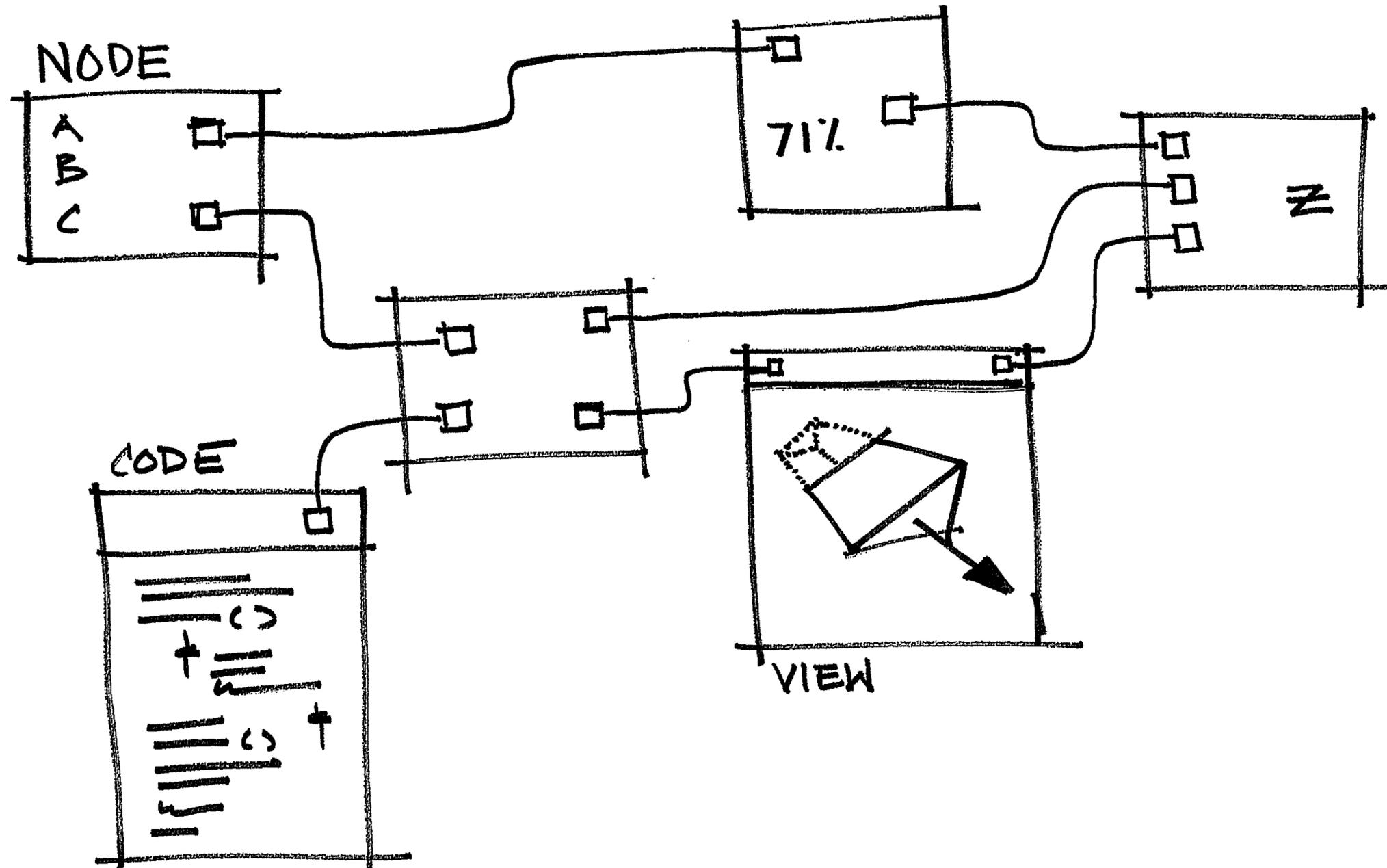
## Preliminary Sketches

# NESTED STRUCTURE



4.3.6, 4.1.6, 1.3.4

# GRAPH (NODE-LINK) ENVIRONMENT



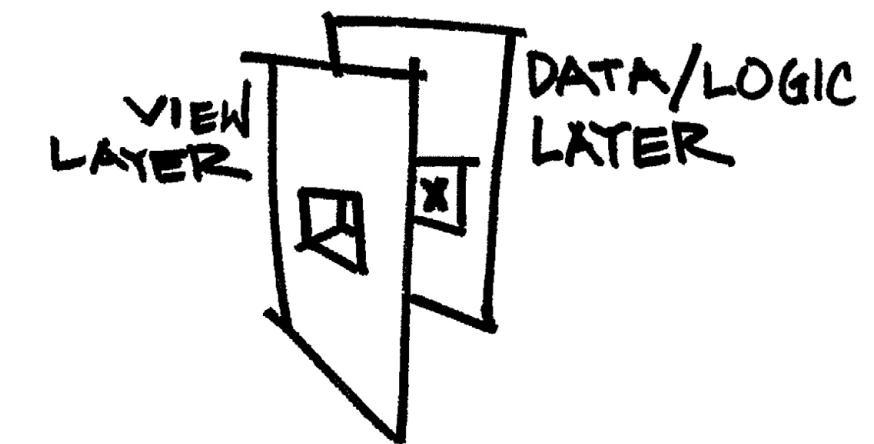
4.1.8, 4.3.1, 4.1.1

# KEYHOLE MODEL

DATA/LOGIC		
	7	
	2	
	1	
$f_x$		

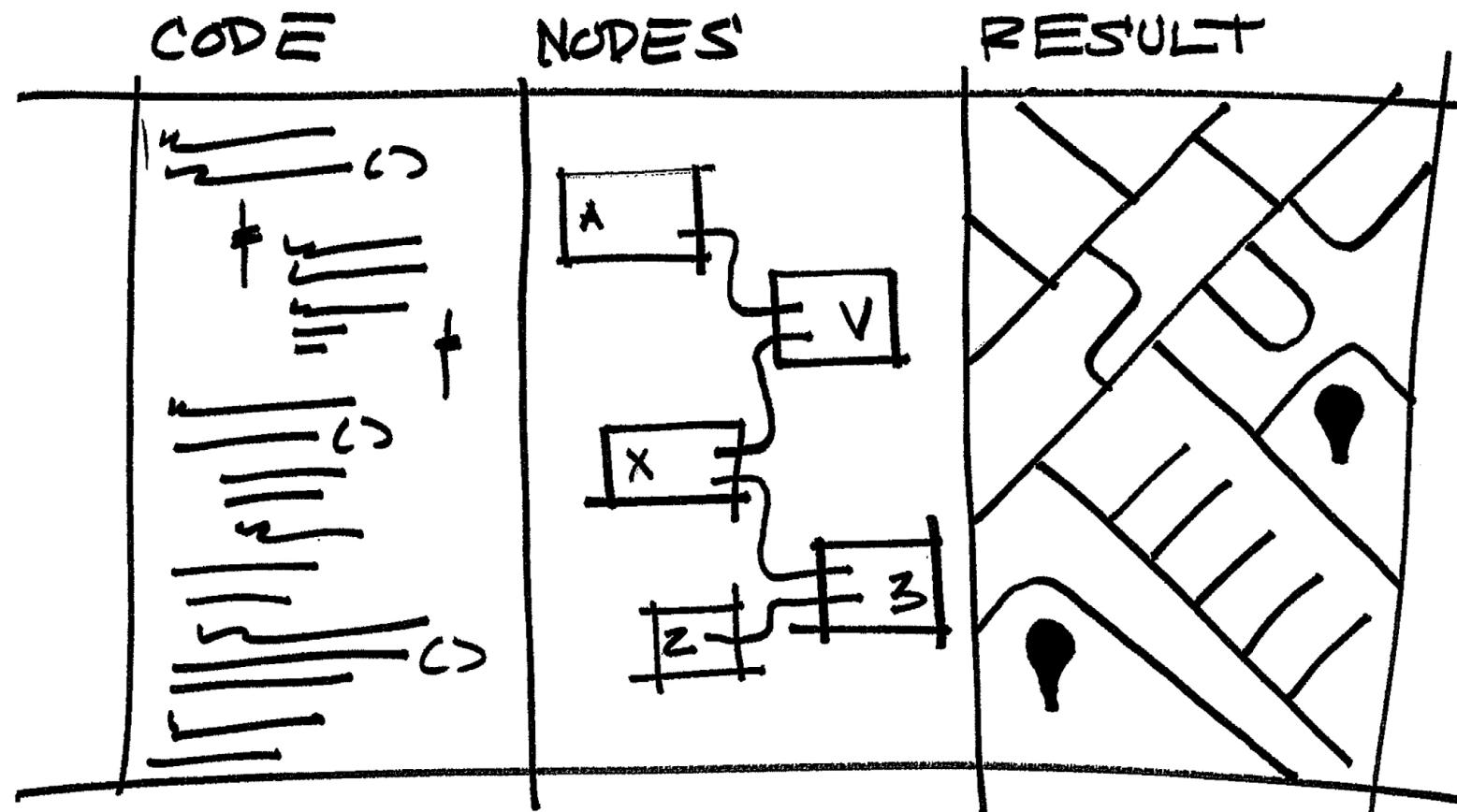
$f_x = \text{SUM}()$

VIEW		
	7	
	2	
	1	
	10	



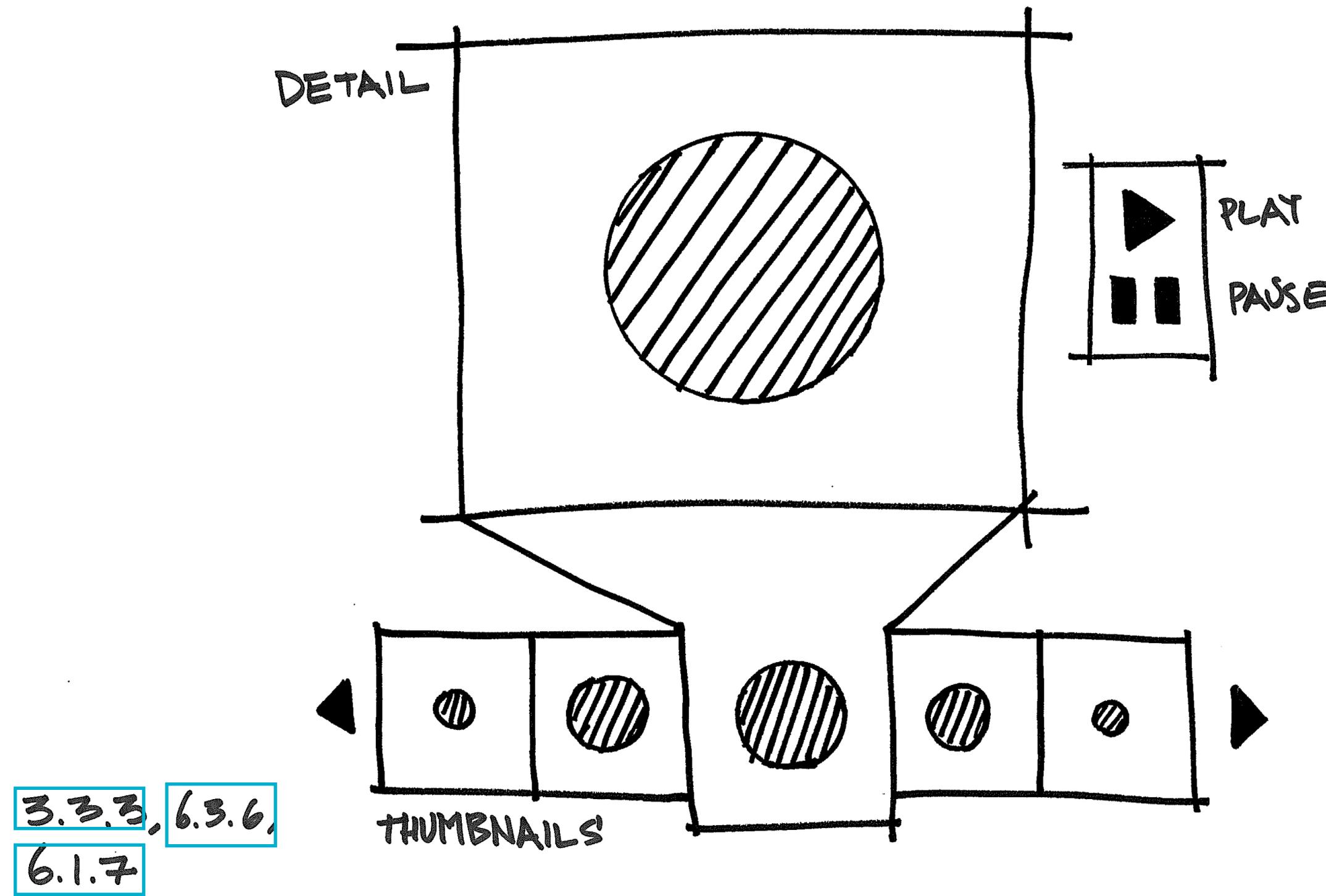
2.1.3, 2.2.3, 2.4.3

# MULTIPLE VIEWS



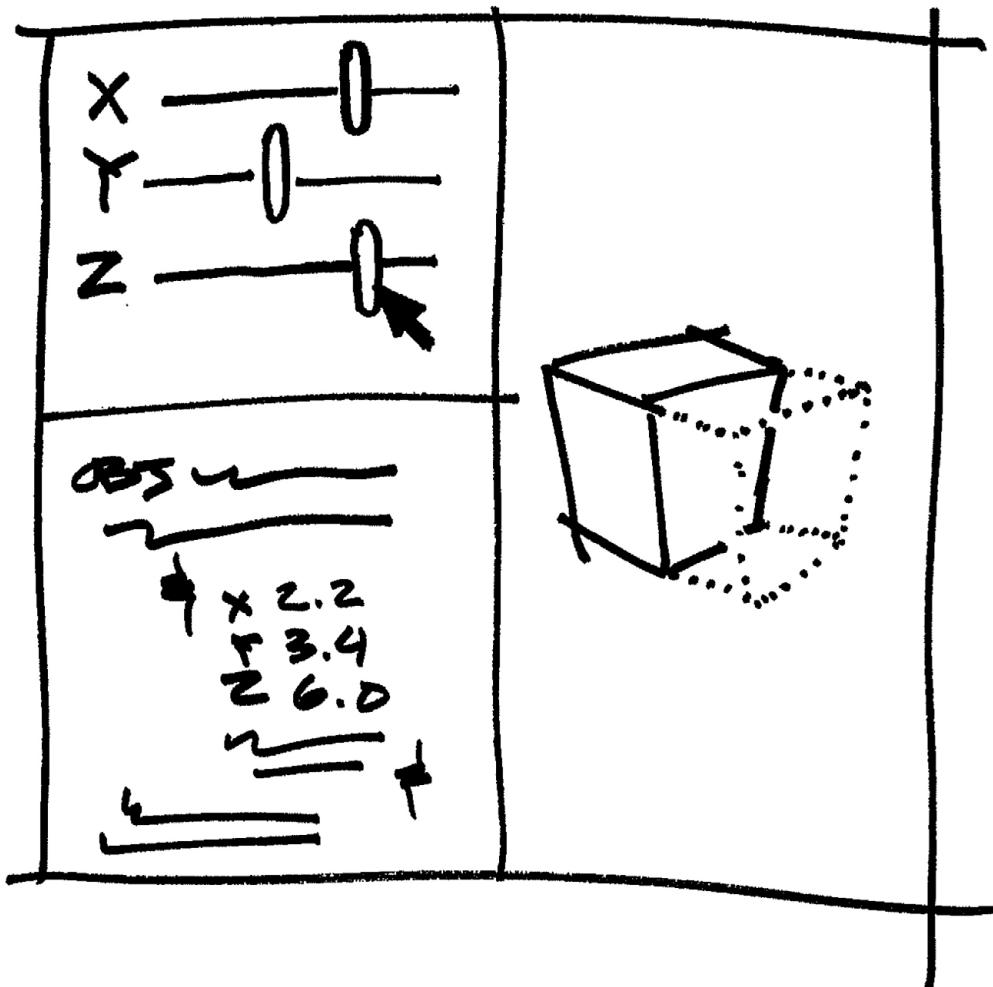
5.4.9, 5.4.3, 4.3.2, 7.2.1

## TIMELINE-BASED

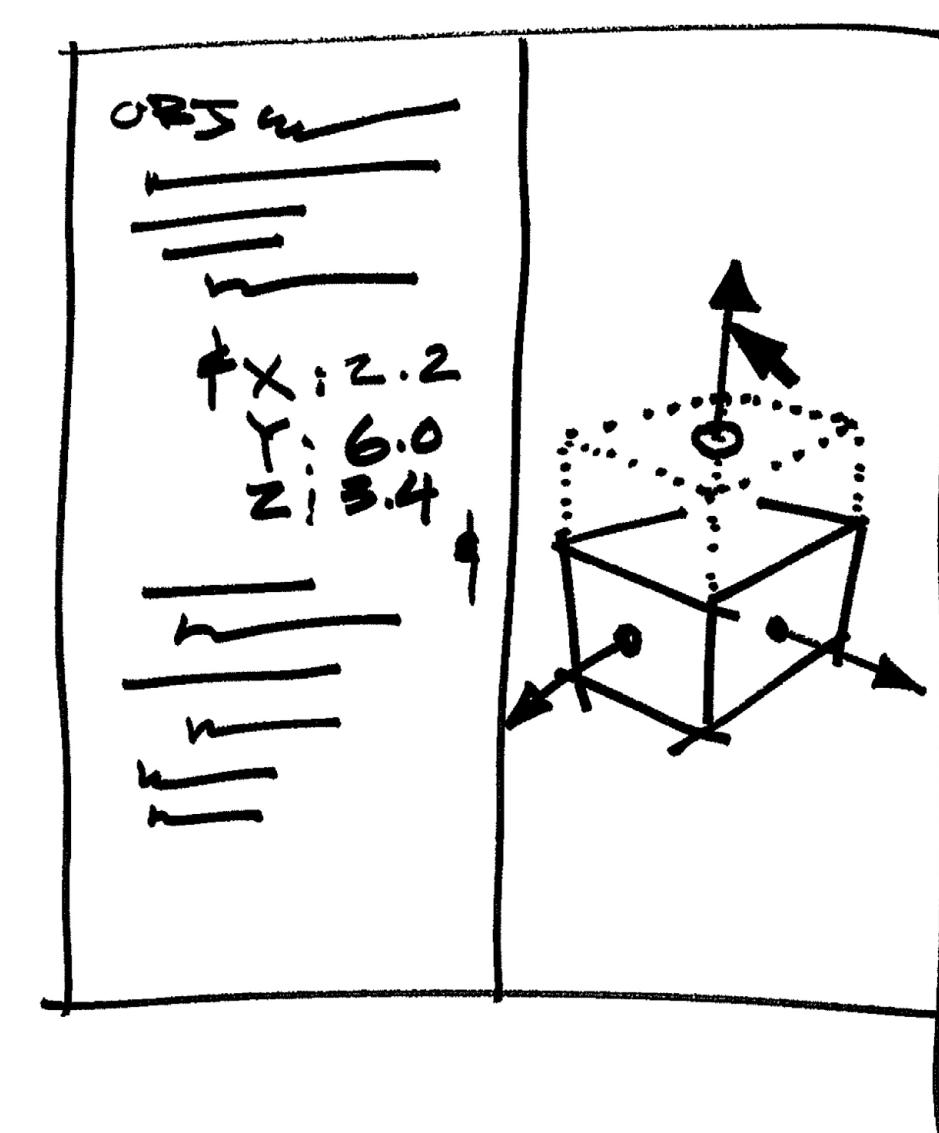


LIVE COMPILE

LIVE "FIDDLE"

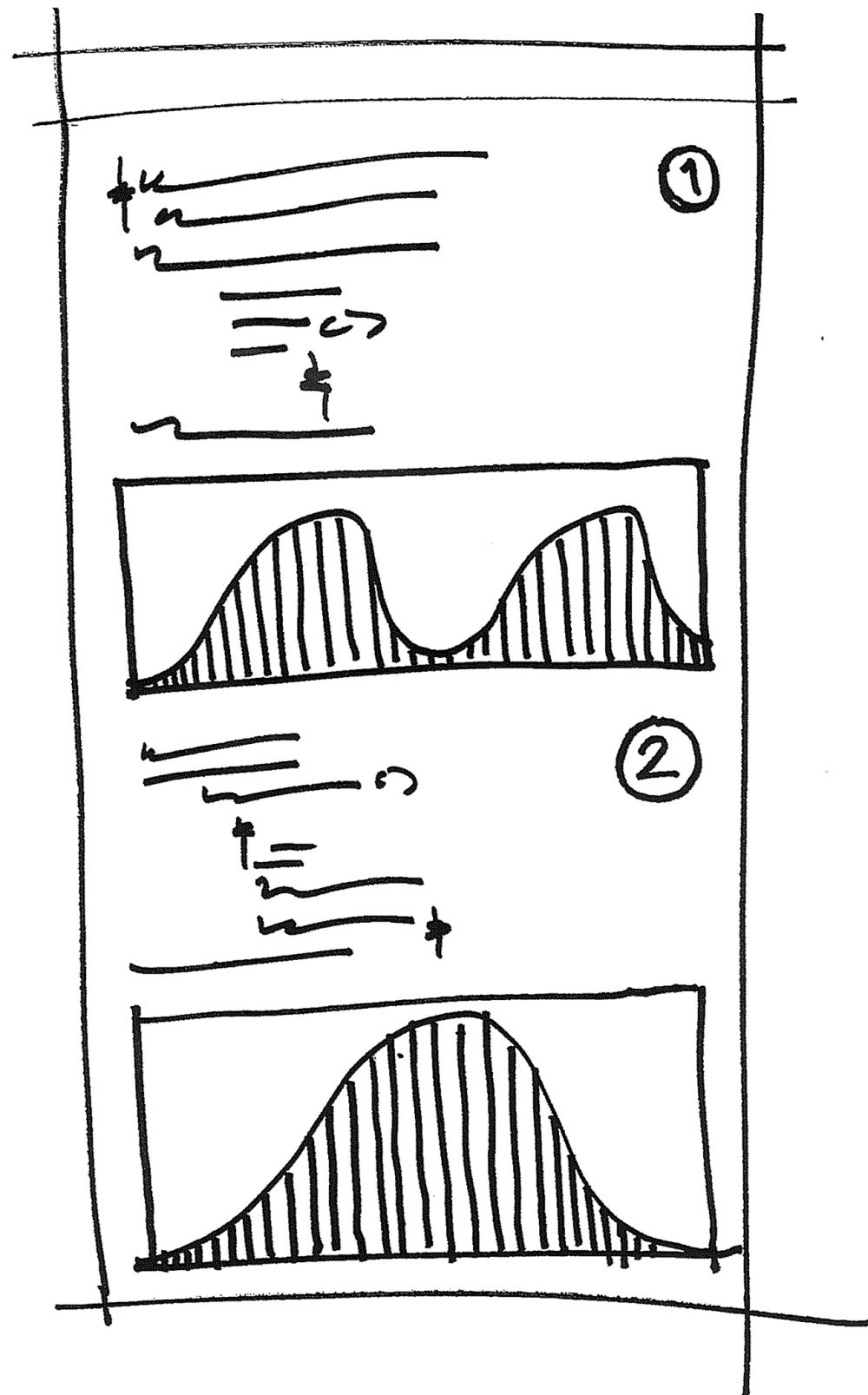


EDITABLE RESULT



4.1.9, 5.4.7, 4.2.4

QUERY  $\neq$  RESPONSE



1.2.4, 1.2.1, 1.2.6

hugh@dubberly.com  
@DubberlyDesign

cameron@dubberly.com  
@supercgeek