

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 principlce 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

```
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 m# 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
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

-UUU:%%-F1 *eww* Top of 10k (1,0) (eww m# Wrap) -----
-rw-r--r-- 1 root root 1476 May 25 09:45 atmclip.h
-rw-r--r-- 1 root root 14878 May 25 09:45 ax25.h
-rw-r--r-- 1 root root 998 May 25 09:45 ax88796.h
drwxr-xr-x 2 root root 4096 Aug 15 19:59 bluetooth
-rw-r--r-- 1 root root 10026 May 25 09:45 bond_3ad.h
-rw-r--r-- 1 root root 6756 May 25 09:45 bond_alb.h
-rw-r--r-- 1 root root 18901 May 25 09:45 bonding.h
-rw-r--r-- 1 root root 3907 May 25 09:45 bond_options.h
-rw-r--r-- 1 root root 3072 May 25 09:45 busy_poll.h
drwxr-xr-x 2 root root 4096 Aug 15 19:59 caif
-rw-r--r-- 1 root root 2195 May 25 09:45 calipso.h
-rw-r--r-- 1 root root 209102 May 25 09:45 cfg80211.h
-rw-r--r-- 1 root root 2000 May 25 09:45 cfg80211-wext.h
-rw-r--r-- 1 root root 11153 May 25 09:45 cfg802154.h
-rw-r--r-- 1 root root 4738 May 25 09:45 checksum.h
-rw-r--r-- 1 root root 8369 May 25 09:45 cipso_ipv4.h

-UU-:----F1 *eshell* All of 112 (4,54) (EShell m# Wrap) ----- -UUU:%%-F1 net 8% of 10k (25,46) (Dired by name m# Wrap) -----
```

David A. Moon and Guy L. Steele Jr.

# Vim

# 1991-Today

```
// 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);
// });

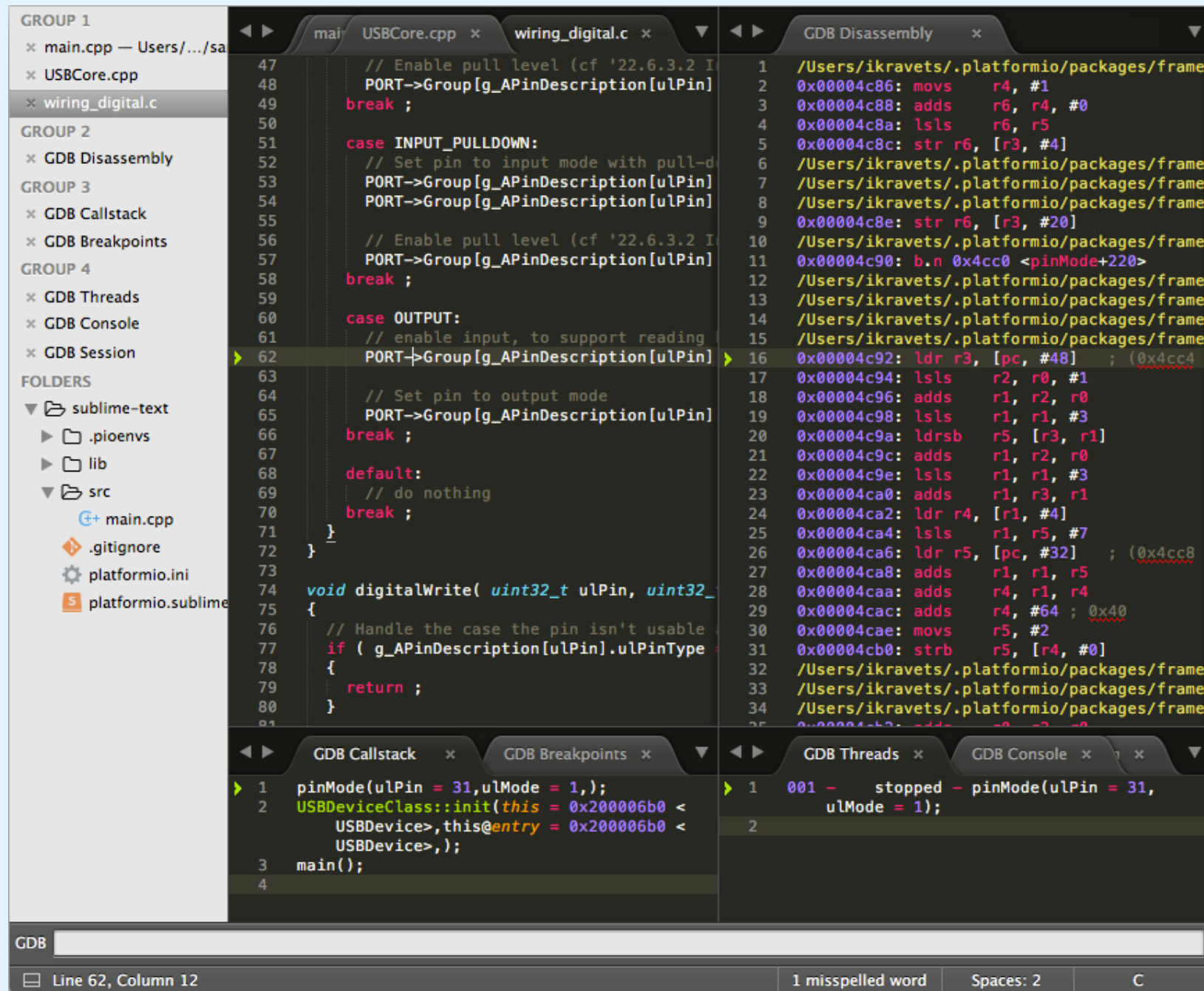
// Attach the Express app to Cloud Code.
app.listen();
~
~
~
~
~
~
~
'require' was used before it was defined.
```

Bram Moolenaar



# Sublime Text

# 2008-Today



Jon Skinner, Will Bond

# Atom

# 2014-Today

The screenshot shows the Atom text editor interface. On the left is the 'Project' sidebar with a file explorer for a 'real-time' project. The main editor area shows the file 'real-time-package.js' with the following code:

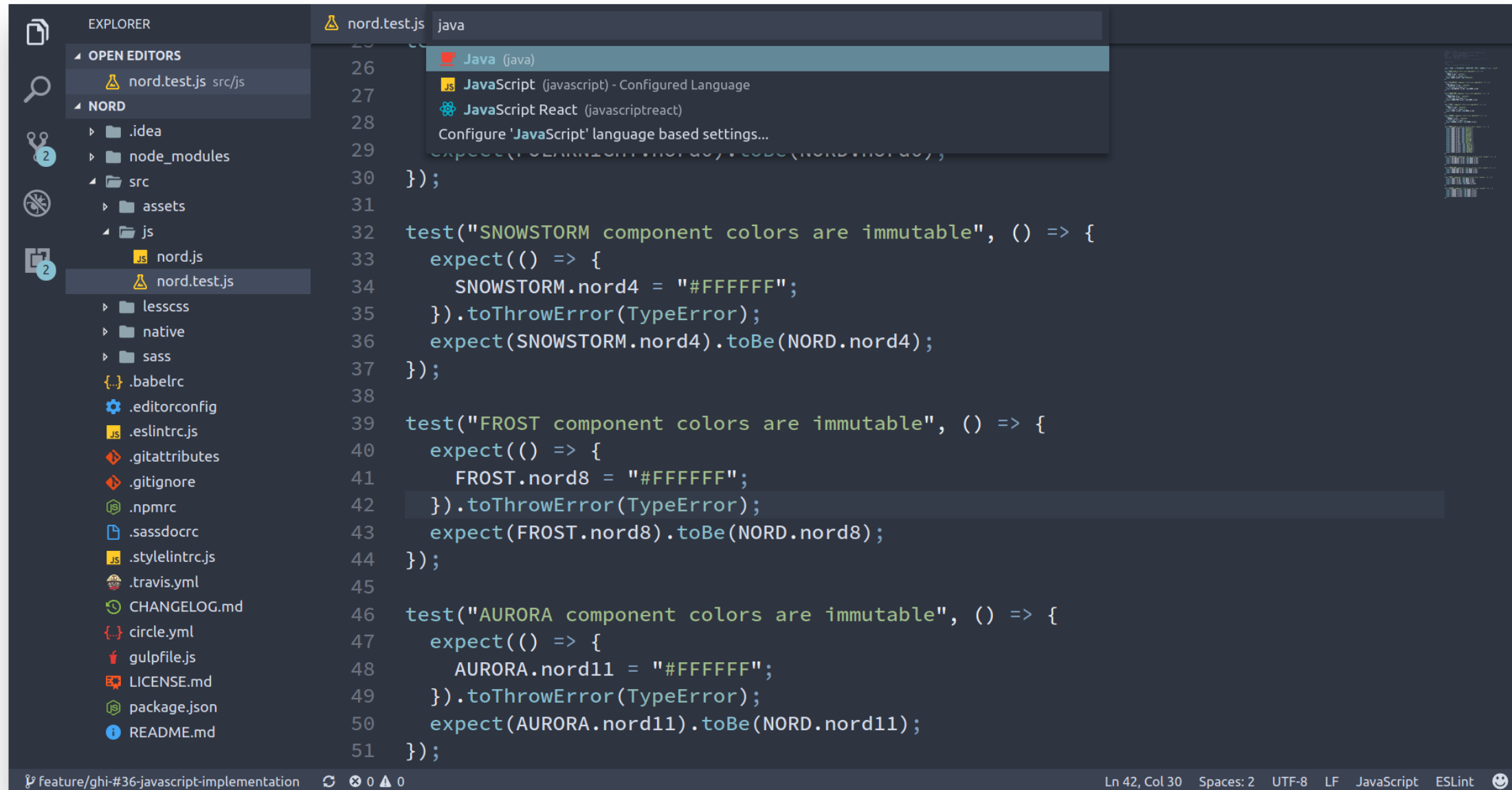
```
1  const {CompositeDisposable} = require('atom')
2  const {allowUnsafeNewFunction} = require('loophole')
3
4  let Client
5  allowUnsafeNewFunction(() => { Client =
6
7  const BufferBinding = require('./buffer-binding')
8  const EditorBinding = require('./editor-binding')
9
10 module.exports =
11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

At the bottom of the editor, the file path 'lib/real-time-package.js' is shown on the left, 'JavaScript' on the right, and a small Atom logo icon.

GitHub

# Visual Studio Code

# 2015-Today



Microsoft

# Notebooks

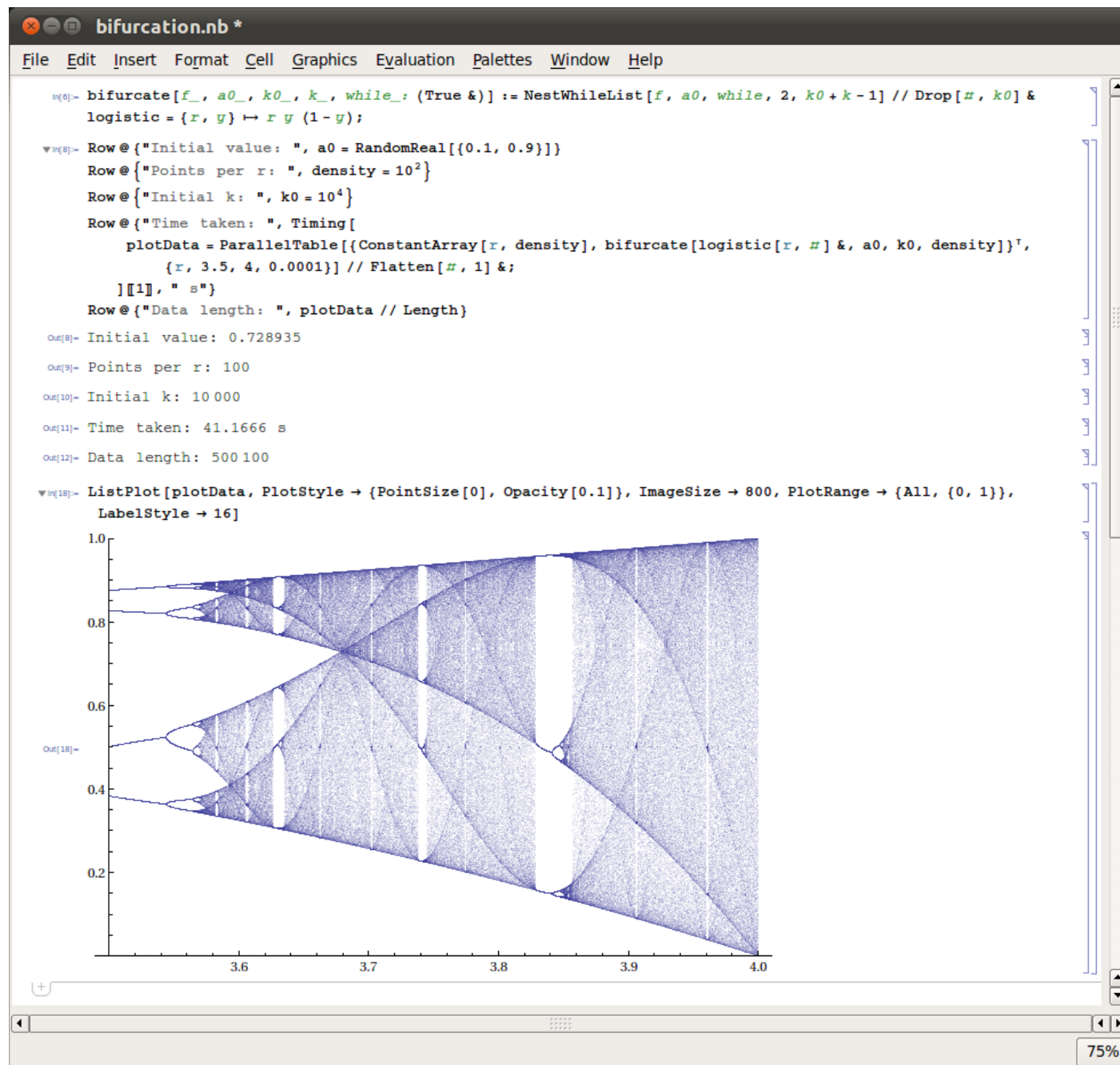
## Query & Response Environments

```
1143 6/13/70 #44 ral 1 [2]

Type "yes" if 23929<45*62.
Type 45*62.
      45*62 =      2790
Type "yes" if 23929>45*62.
yes
Type "yes" if 97843=97843.
Eh?
SORRY.
Eh?
Let x = 56*75.
Type x.
      x =      4200
Let y = x-75.
Type y.
      y =      4125
Type x*y.
Eh?
Type x*y.
      x*y =      1.7325*10*7
Type x/y.
      x/y =      1.01818182
Type x/y*9654.
      x/y*9654 =      9829.52729
Let v=x+9986523/4.
Type v*x*y.
      v*x*y =      4.3326893*10*13
Type v.
Eh?
Type v.
      v =      2.50083075*10*6
Type x.
      x =      4200
Type y.
      y =      4125
Type v*x+y/7.
      v*x+y/7 =      1.05034898*10*10
Type v/x.
      v/x =      595.435893
Type v/x+y+89/67*34.
v/x+y+89/67*34 =      4765.60007
Type v-x.
      v-x =      2.49663075*10*6
```

# Mathematica

# 1988-Today



Stephen Wolfram

The screenshot shows an IPython Notebook window with the following content:

- Browser tabs:** IPy IPython Dashboard, IPy spectrogram
- Address bar:** 127.0.0.1:8888/a5222740-848b-4ac1-b212-d732c9f8f78b
- Page title:** IP[y]: Notebook spectrogram Last saved: Mar 07 11:14 PM
- Menu bar:** File, Edit, View, Insert, Cell, Kernel, Help
- Toolbar:** Includes icons for undo, redo, copy, paste, and a dropdown menu set to 'Markdown'.
- Section Header:** Simple spectral analysis
- Text:** An illustration of the [Discrete Fourier Transform](#)
- Equation-Block:**
$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N} kn} \quad k = 0, \dots, N-1$$
- Text:** using windowing, to reveal the frequency content of a sound signal.
- Text:** We begin by loading a datafile using SciPy's audio file support:
- Code Cell [1]:**

```
from scipy.io import wavfile
rate, x = wavfile.read('test_mono.wav')
```
- Text:** And we can easily view its spectral structure using matplotlib's builtin specgram routine:
- Code Cell [2]:**

```
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(12, 4))
ax1.plot(x); ax1.set_title('Raw audio signal')
ax2.specgram(x); ax2.set_title('Spectrogram');
```
- Figure:** Two side-by-side plots. The left plot, titled 'Raw audio signal', shows a blue waveform with an amplitude range from -10000 to 8000 and a time range from 0 to 50000. The right plot, titled 'Spectrogram', shows a heatmap of frequency content with a vertical axis from 0.0 to 1.0 and a horizontal axis from 0 to 25000.

# Jupyter

# 2014-Today

The screenshot shows a Jupyter Notebook interface with the following content:

- Header: `jupyter Custom Display Logic (unsaved changes)` with a Python logo and a `Logout` button.
- Menu: `File Edit View Insert Cell Kernel Widgets Help`
- Trust status: `Not Trusted` and `Python 3`
- Toolbar: `Run`, `Stop`, `Refresh`, `Markdown`, and `Code` icons.
- Code cell 4: `In [4]: x = Gaussian(2.0, 1.0)`  
`x`
- Output 4: `Out[4]:  $\mathcal{N}(\mu = 2, \sigma = 1), N = 1000$`
- Text: `You can also pass the object to the display function to display the default representation:`
- Code cell 5: `In [5]: display(x)`
- Output 5:  `$\mathcal{N}(\mu = 2, \sigma = 1), N = 1000$`
- Text: `Use display_png to view the PNG representation:`
- Code cell 6: `In [6]: display_png(x)`
- Output 6: A histogram plot titled  `$\mathcal{N}(\mu = 2, \sigma = 1), N = 1000$` . The x-axis ranges from -10 to 10, and the y-axis ranges from 0 to 60. The plot shows a distribution of blue bars centered around 2.

Fernando Pérez and Others



# Tonic/RunKit

2015-Today

The screenshot shows the 'Data Visualizing' interface with the following content:

- Code editor: `1 new Buffer("Hello!");`
- Object type: **Buffer (6 bytes)** with a 'Data Explorer' dropdown.
- Table visualization:

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	.....	....

Code editor: `2`

saving

The screenshot shows the 'Data Visualizing' interface with the following content:

- Code editor: `1 var coords = {latitude: 37.773972, longitude: -122.431297};`
- Object type: **Object** with a 'Map Coordinates' dropdown.
- Map visualization: A map of the San Francisco Bay Area with a blue location pin centered on San Francisco. Labeled cities include San Rafael, San Pablo, Richmond, El Cerrito, Mill Valley, Berkeley, Walnut Creek, Oakland, Alameda, San Leandro, San Francisco, Daly City, South San Francisco, Hayward, and Union City. The map includes a zoom control (+/-) and a Leaflet attribution at the bottom.

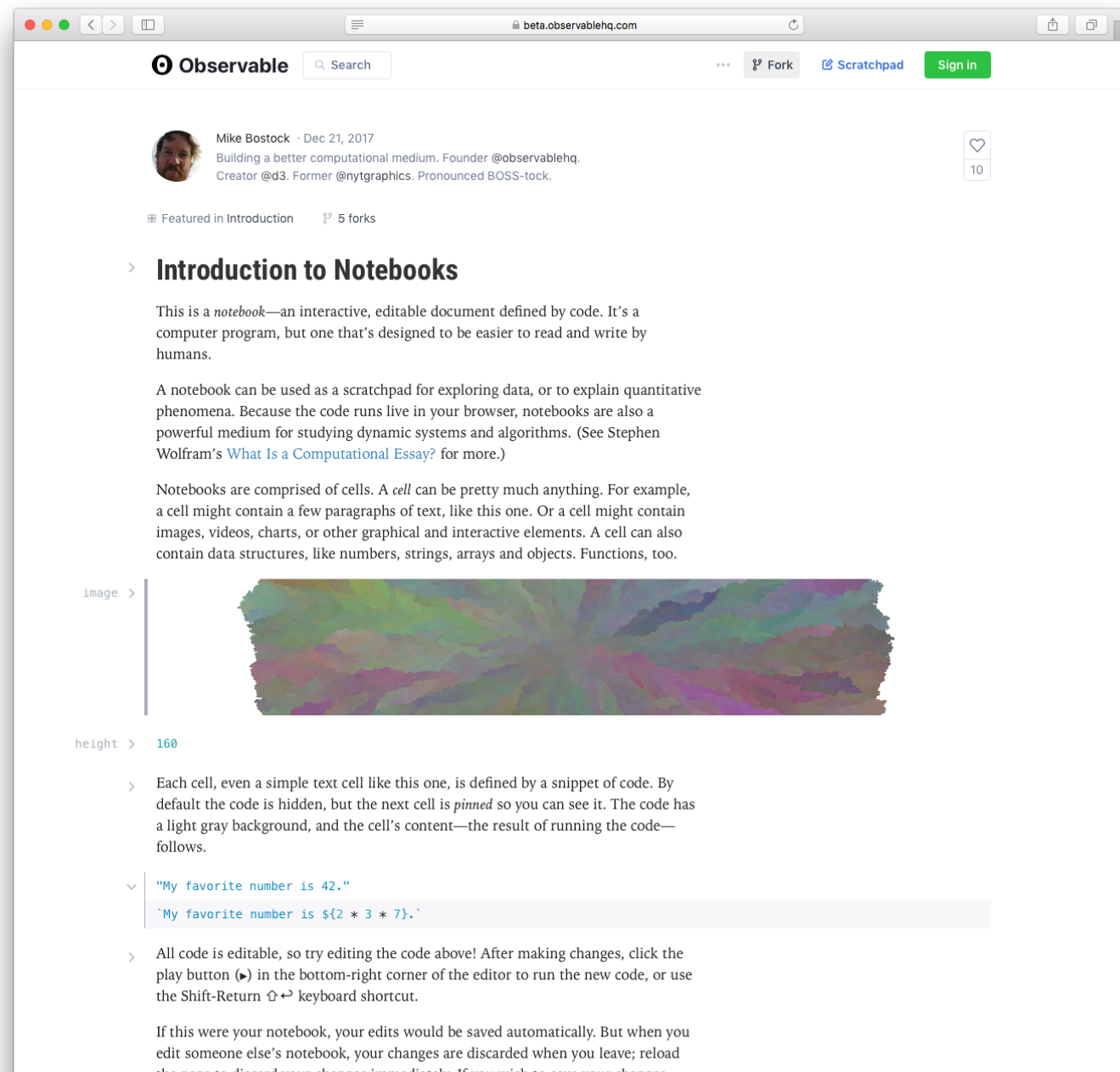
Code editor: `2`

saving

Francisco Tolmasky

# Observable

# 2018-Today



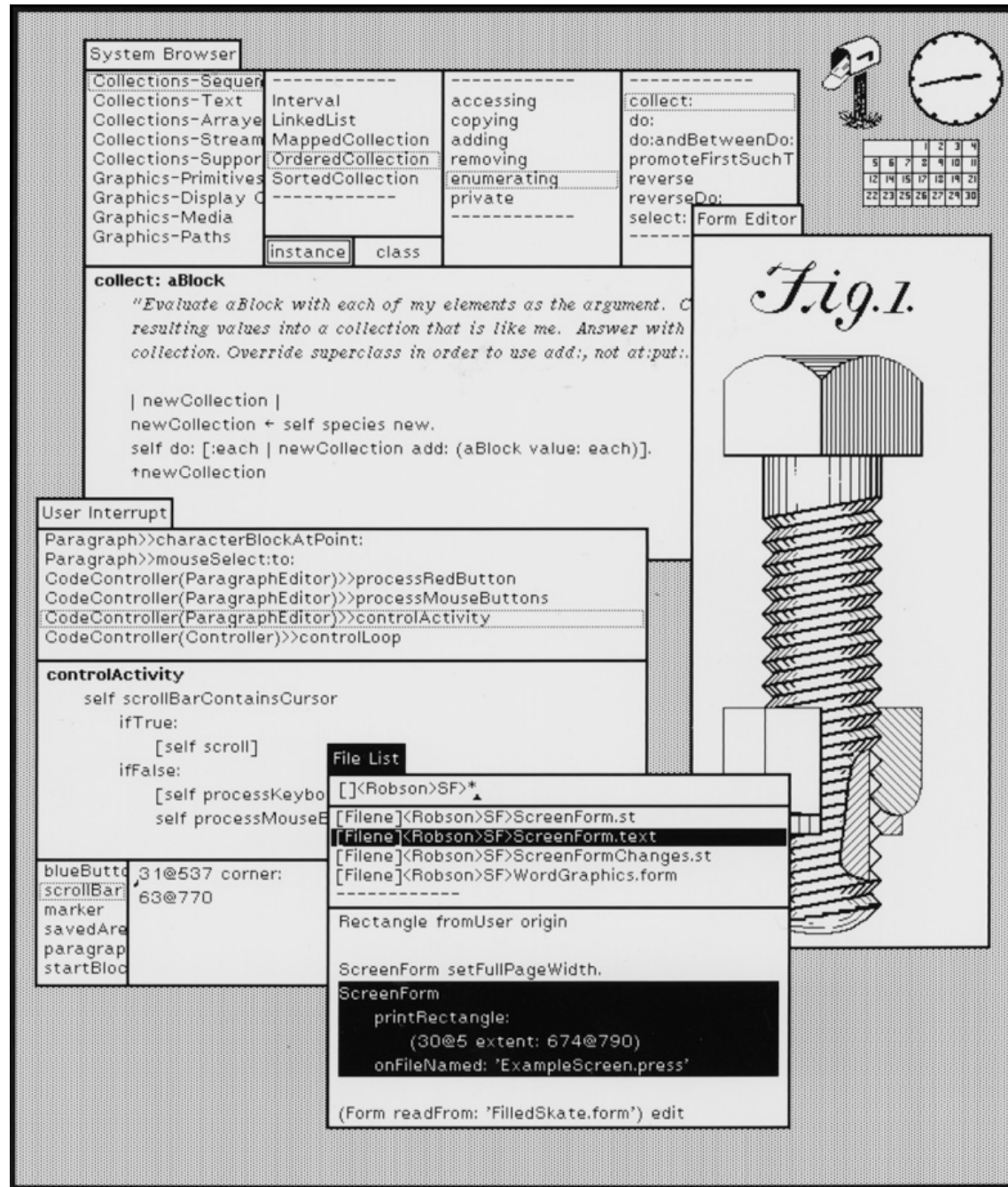
Mike Bostock, Tom MacWright, Jeremy Ashkenas for Observable, Inc.

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

'Construction Sets'

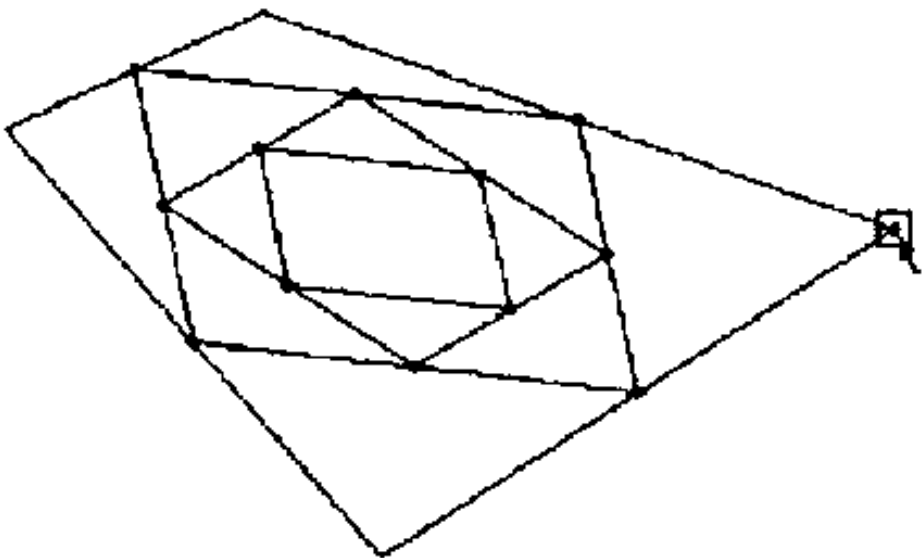
# SmallTalk

1972-1980

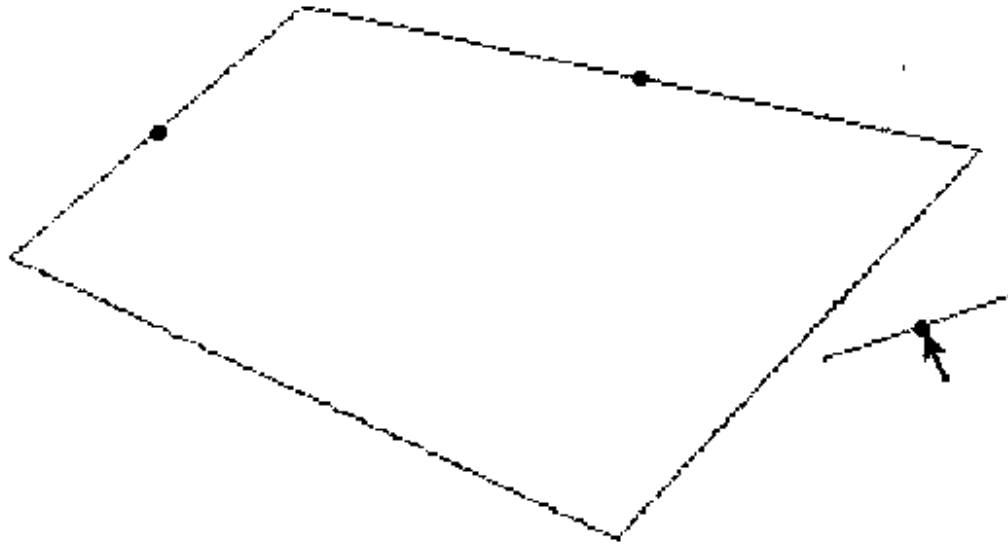


Alan Kay, Dan Ingalls, Adele Goldberg, Ted Kaehler, Diana Merry, Scott Wallace, Peter Deutsch at XEROX PARC

ThingLab Browser			
Anchor BitImage ConstantLengthL EX2 LineSegment	picture structure values	constrain merge move edit  prefer require	LineSegment MidPointLine Node Plus Point Printer Rectangle

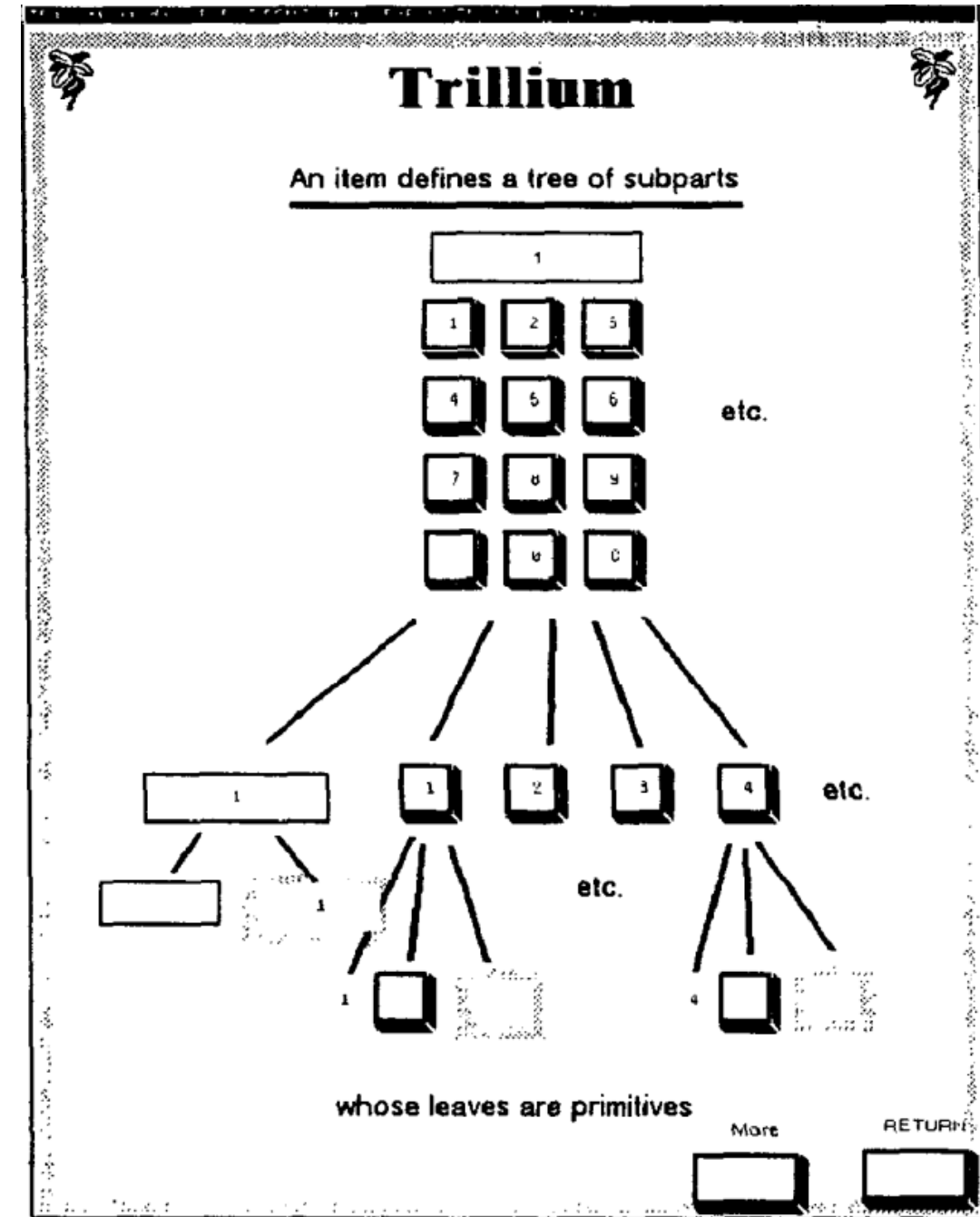
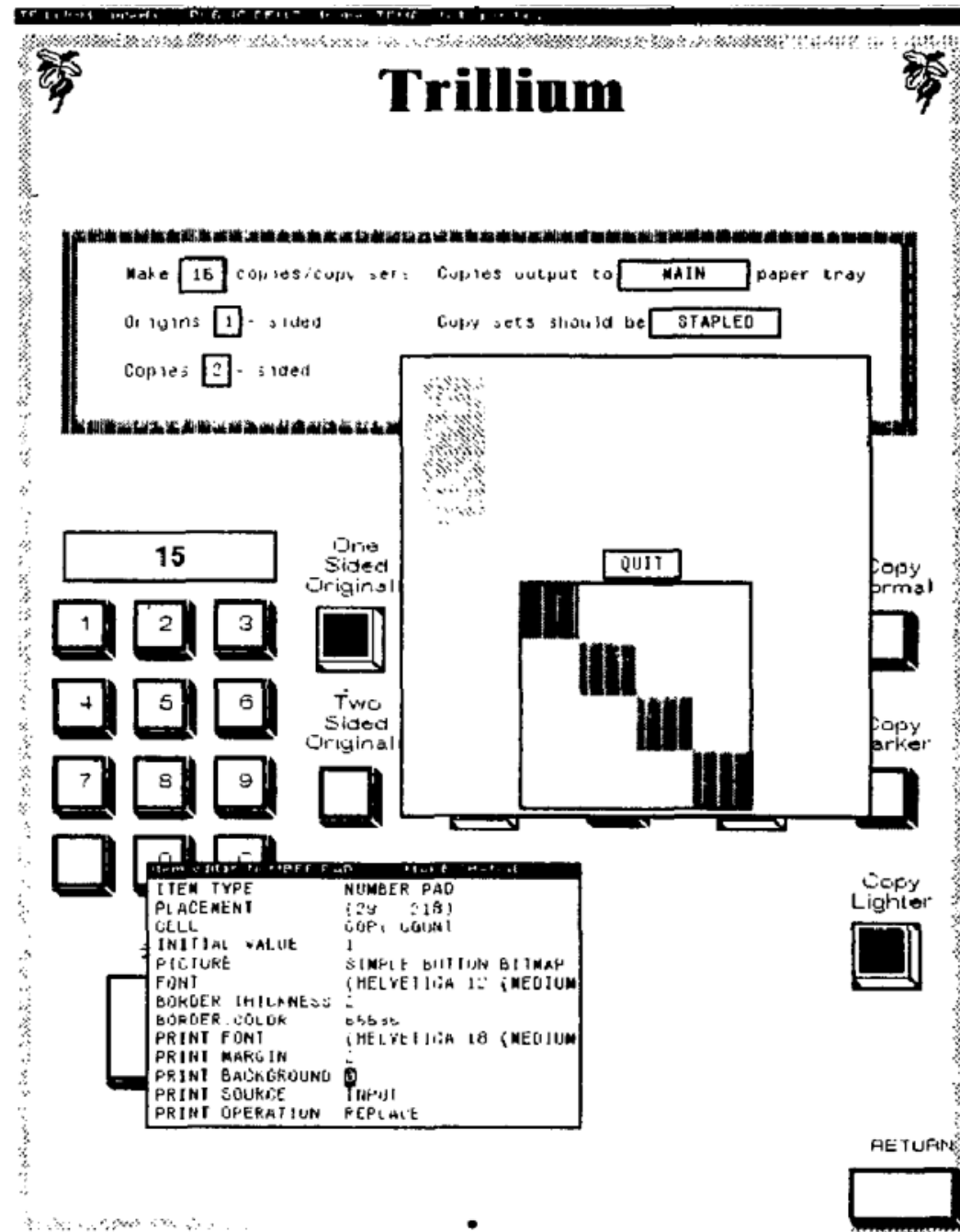


Object Point QTheorem Quadrilateral Rectangle TextThing Triangle	structure prototype's picture prototype's values as save file subclass template	insert delete constrain merge move edit text	GeometricObject Line MidPointLine Point Quadrilateral Rectangle Triangle
--	---	---	--



# Trillium

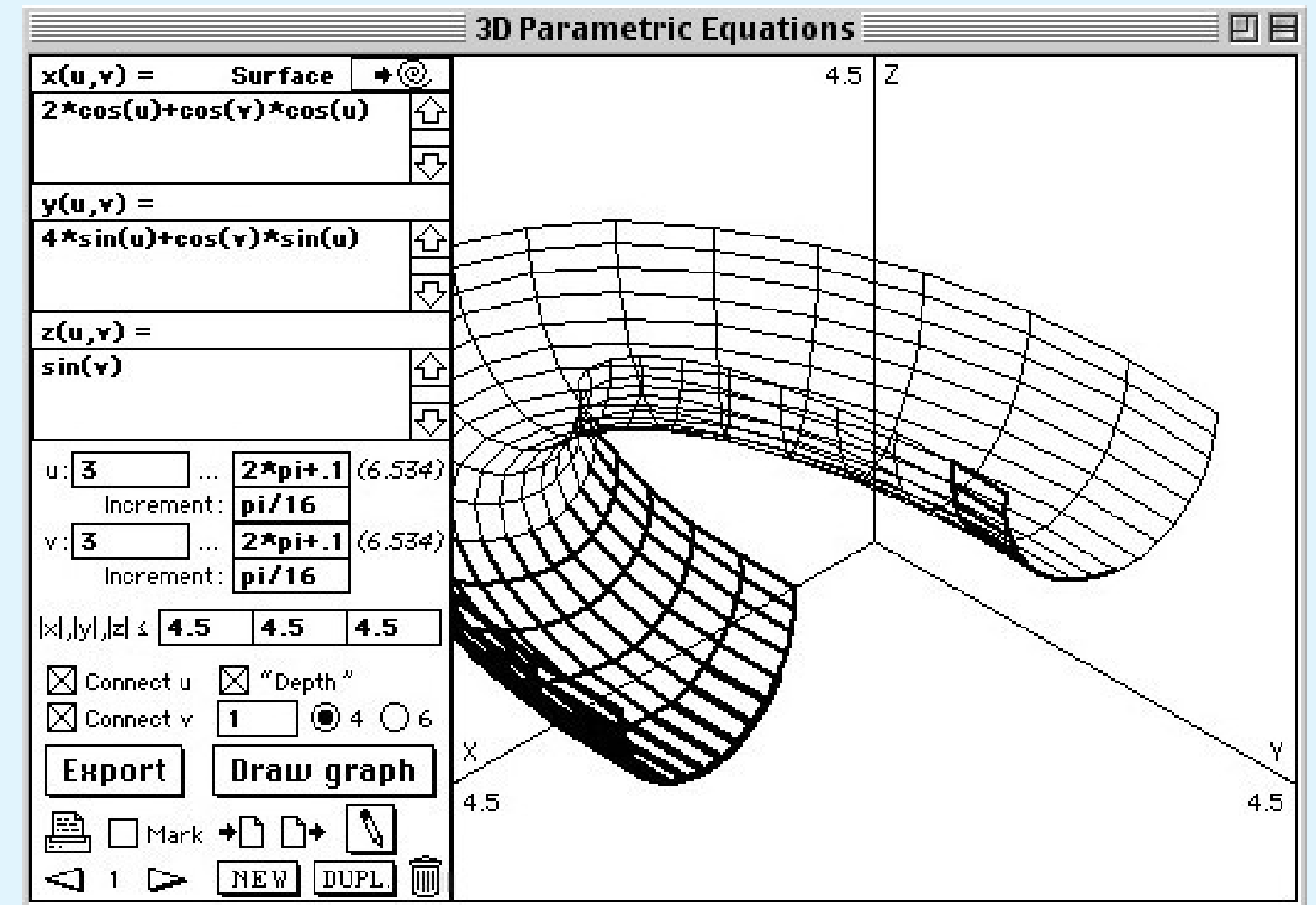
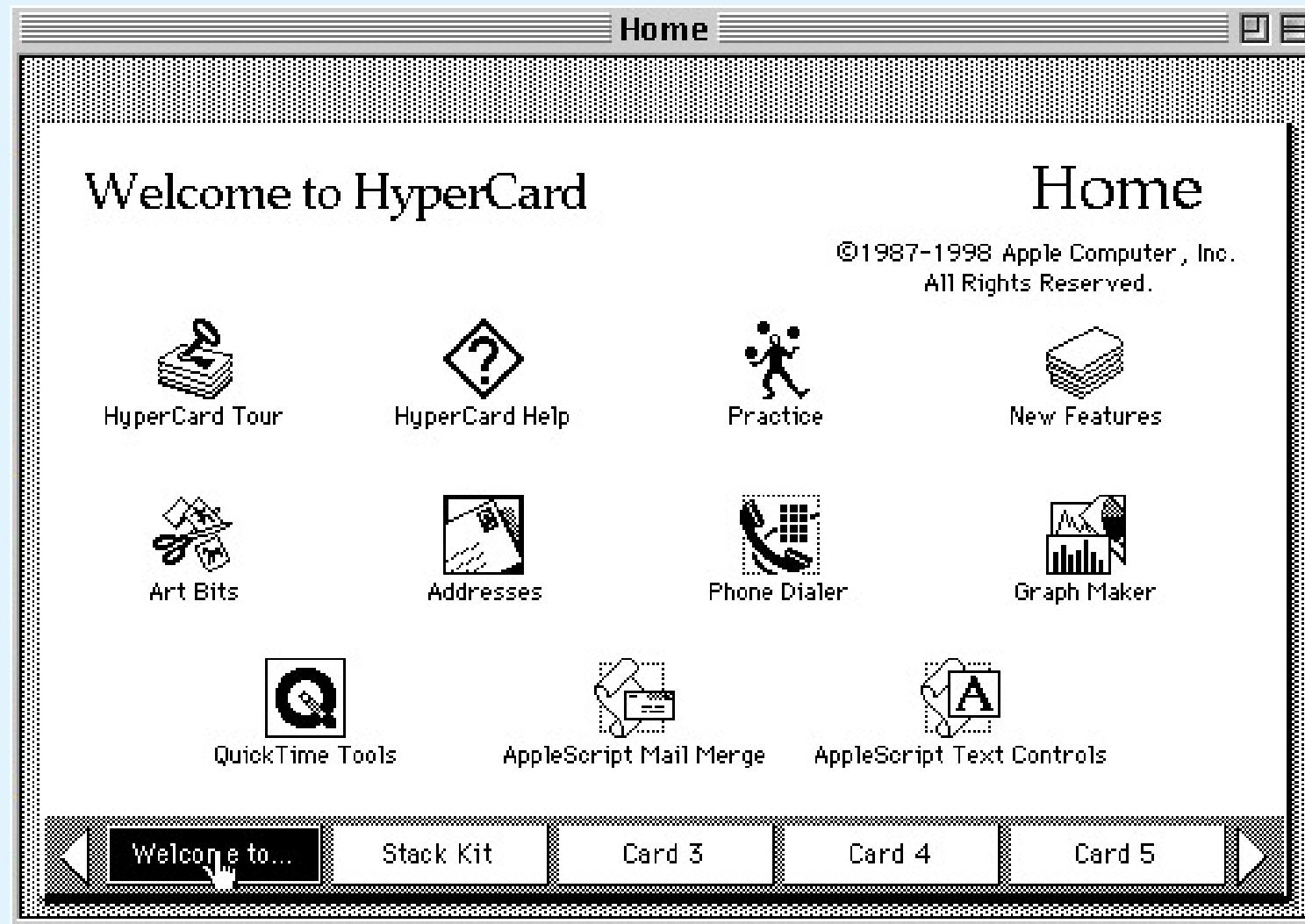
1986



Austin Henderson at Xerox PARC

# Hypercard

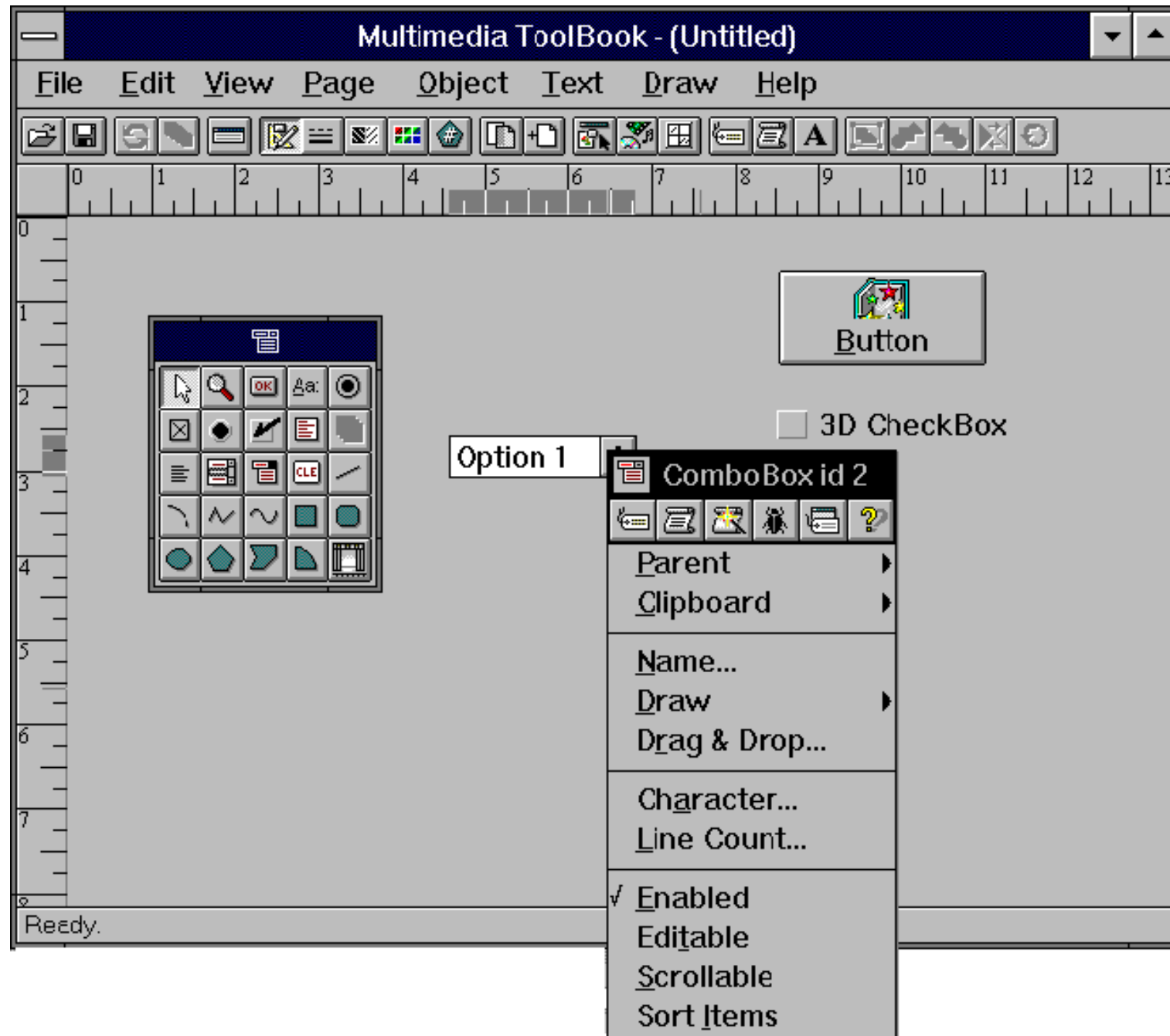
1987-1998



Bill Atkinson for Apple

# ToolBook

1990-2012

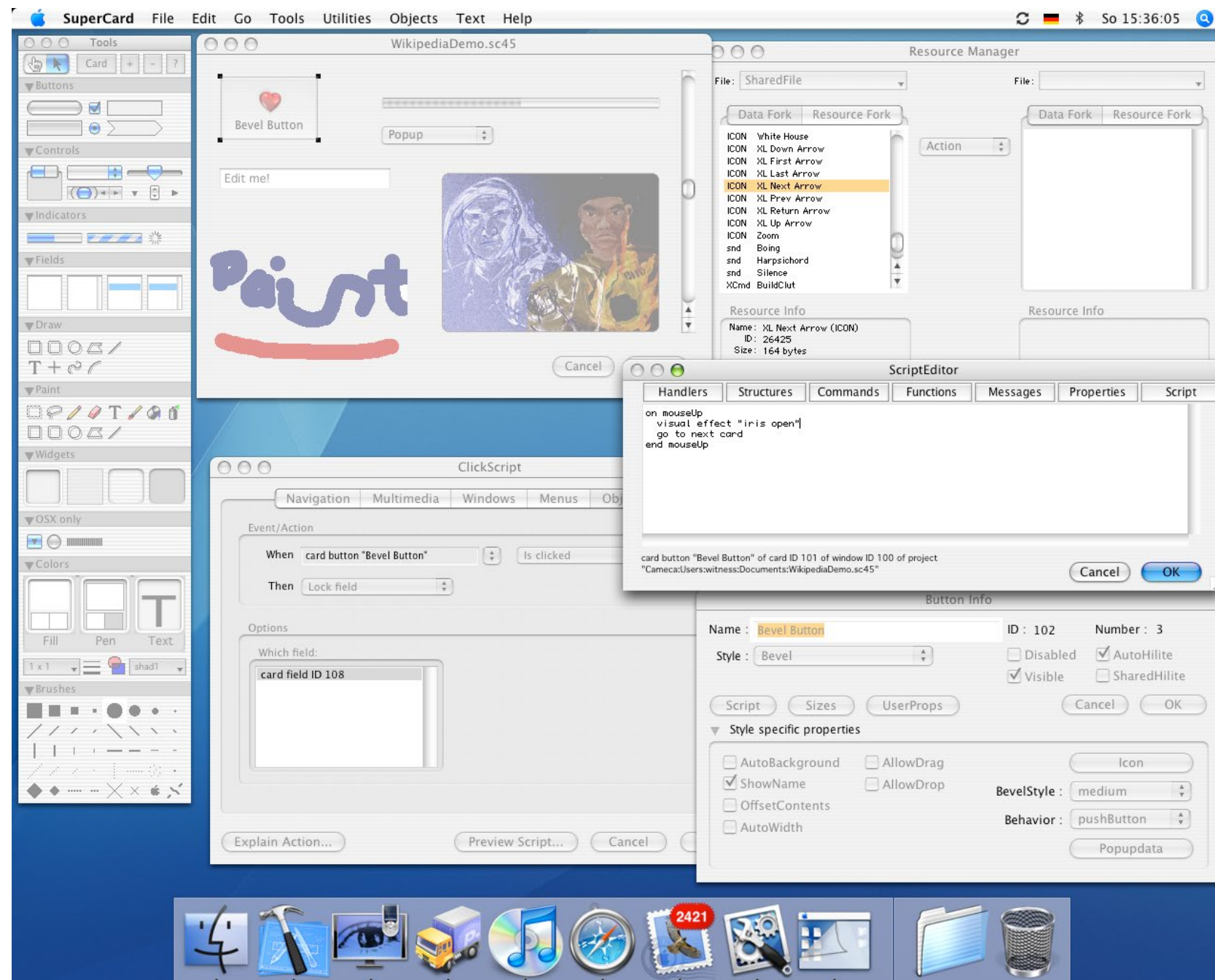


Asymetrix (Later SumTotal Systems)



# 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

```
H:INTRO PAGE 1 LINE 9 COL 11          INSERT ON
      <<<      M A I N   M E N U      >>>
--Cursor Movement-- | -Delete- | -Miscellaneous- | -Other Menus-
^S char left ^D char right | ^G char | ^I Tab  ^B Reform | (from Main only)
^A word left ^F word right | DEL chr lf| ^V INSERT ON/OFF | ^J Help  ^K Block
^E line up  ^X line down  | ^T word rt|^L Find/Replce again|^Q Quick ^P Print
      --Scrolling--      | ^Y line  | RETURN End paragraph|^O Onscreen
^Z line down ^W line up   |          | ^N Insert a RETURN  |
^C screen up ^R screen down|          | ^U Stop a command   |
!----!----!----!----!----!----!----!----!----!----!-----R

1. Introducing WordStar

WordStar is highly flexible and very visible. Watch the
screens as you give commands, and information in various
parts of the screen will guide you. You won't see all the
information all the time, but it will be there when you need
it.

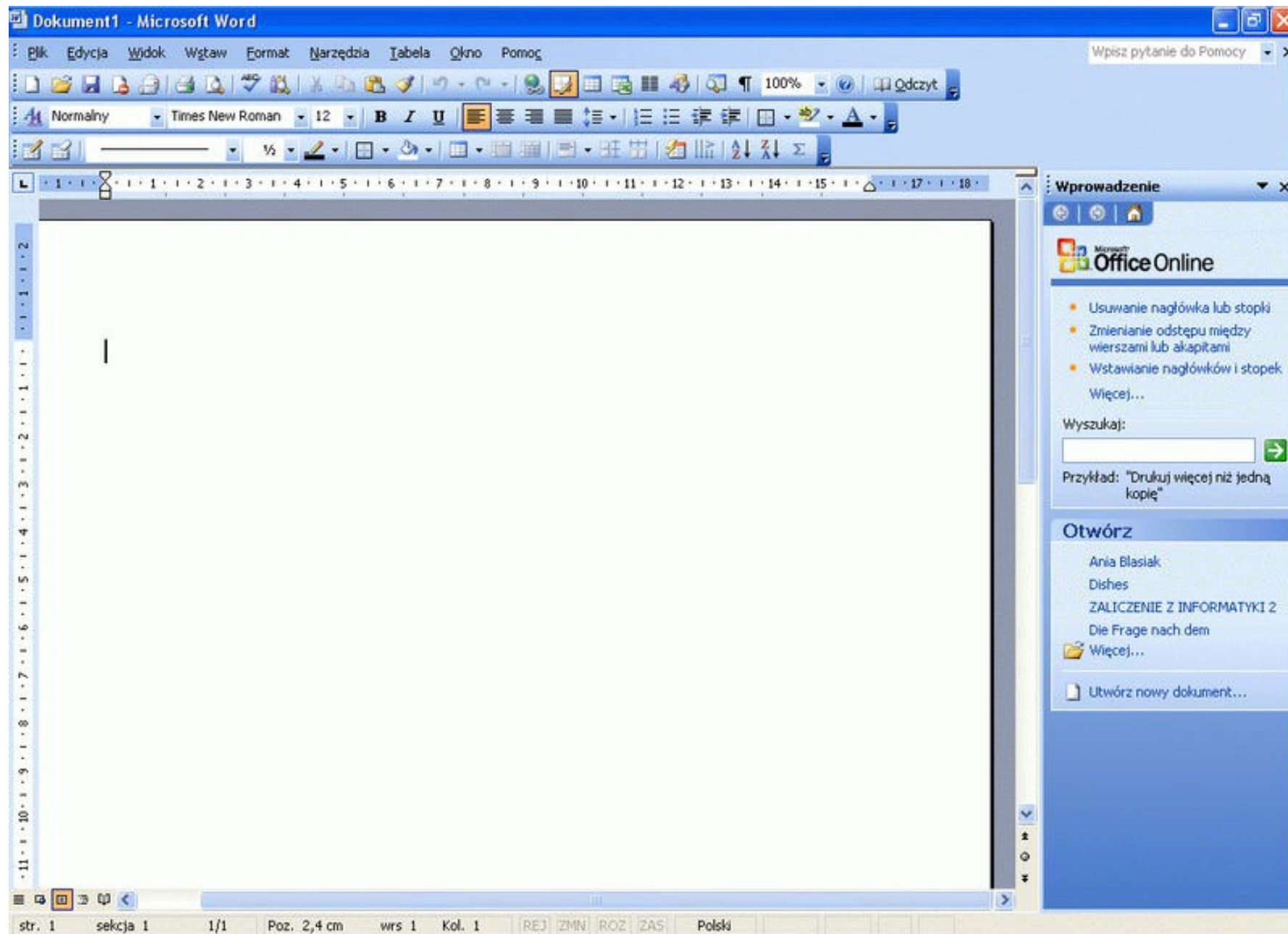
  WHERE YOU ARE

The seven WordStar menus are your greatest aids. They are
like signposts at the top of your screen, showing you where
you are.
1HELP 2INDENT 3SET LM 4SET RM 5UNDLIN 6BLDFCE 7BEGBLK 8ENDBLK 9BEGFIL 10ENDFIL
```

Rob Barnaby

# Word

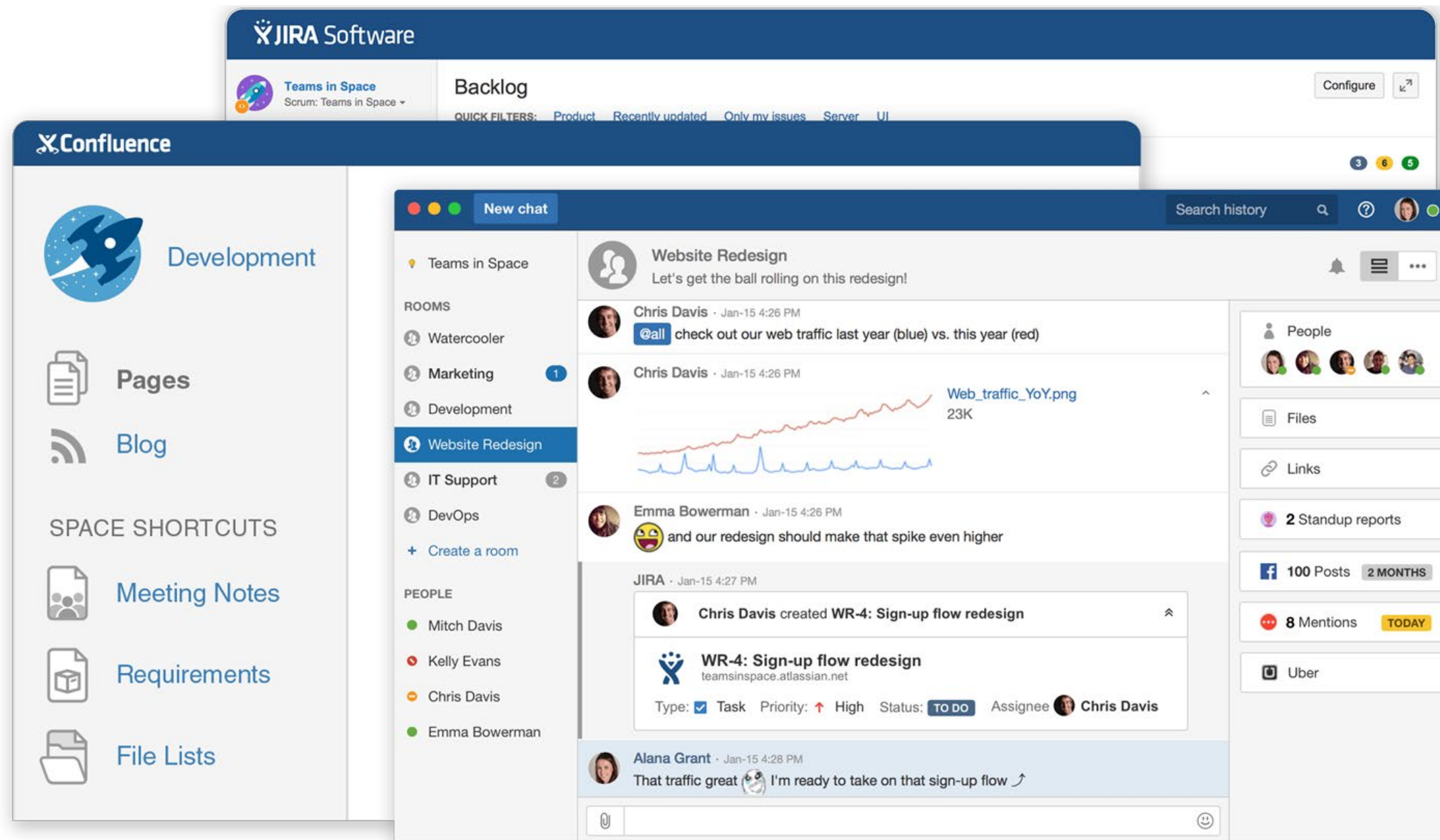
# 1983-Today



Microsoft

# Atlassian (Suite)

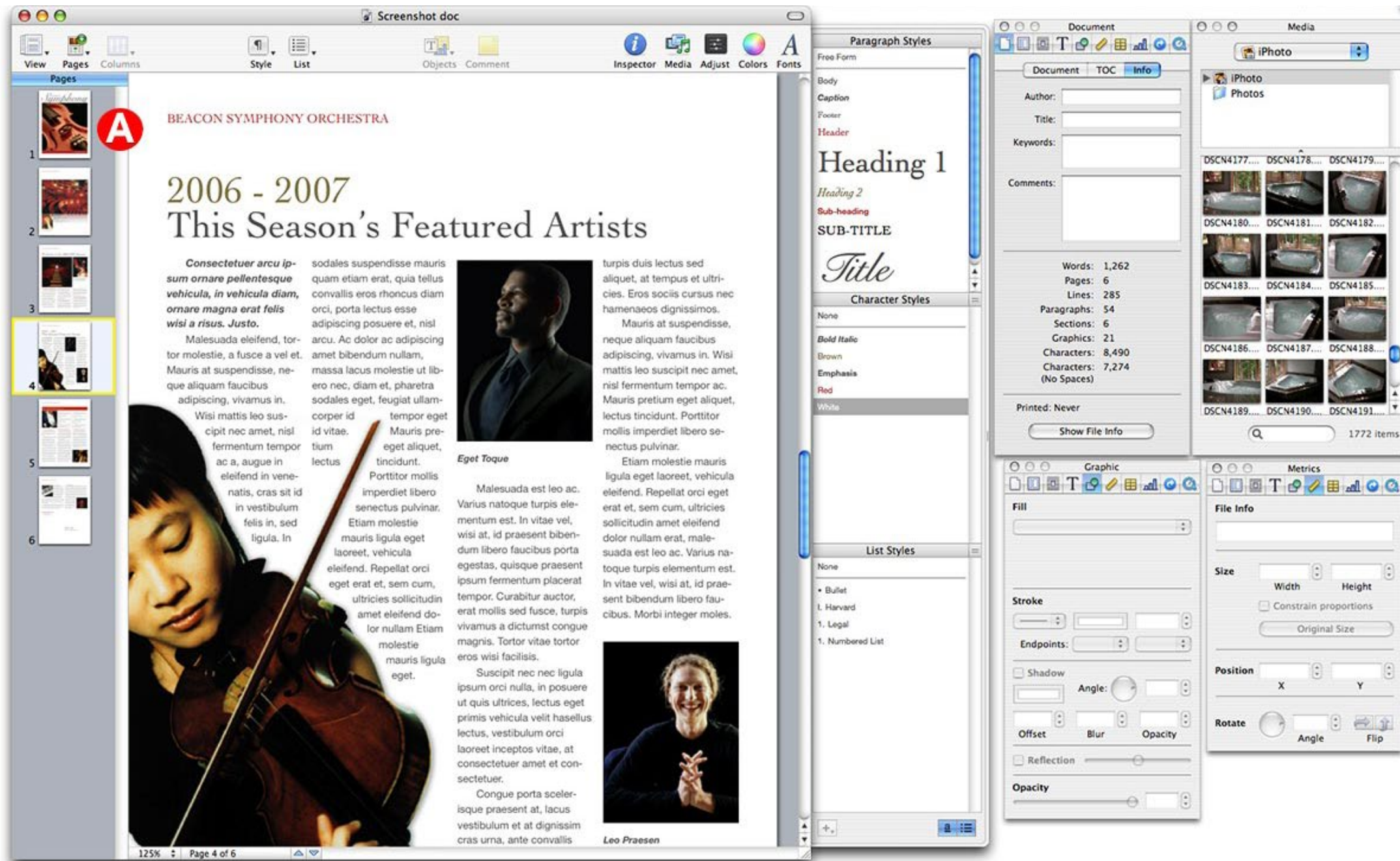
# 2002-Today



Mike Cannon-Brookes and Scott Farquhar

# Pages

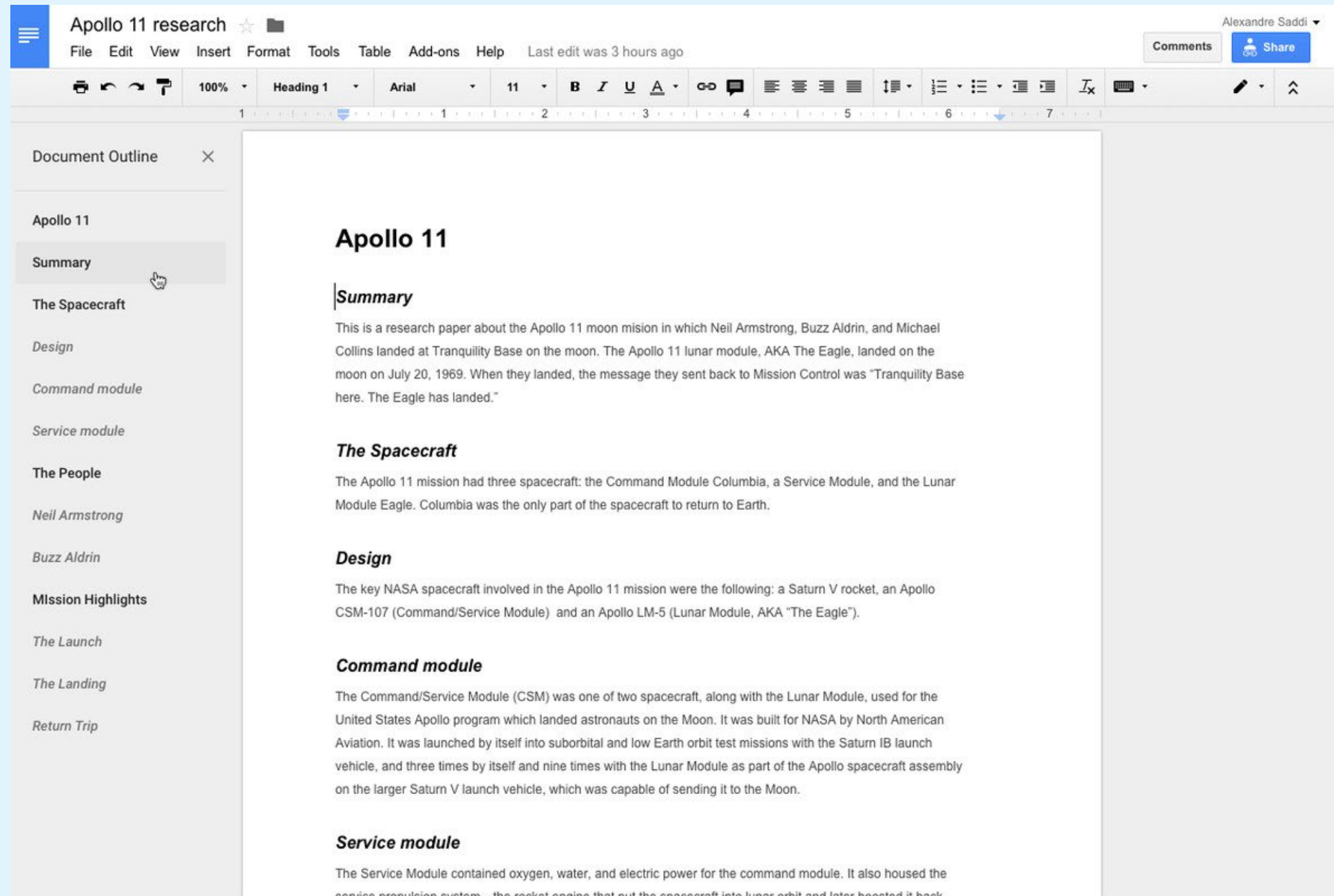
# 2005-Today



Apple

# Docs

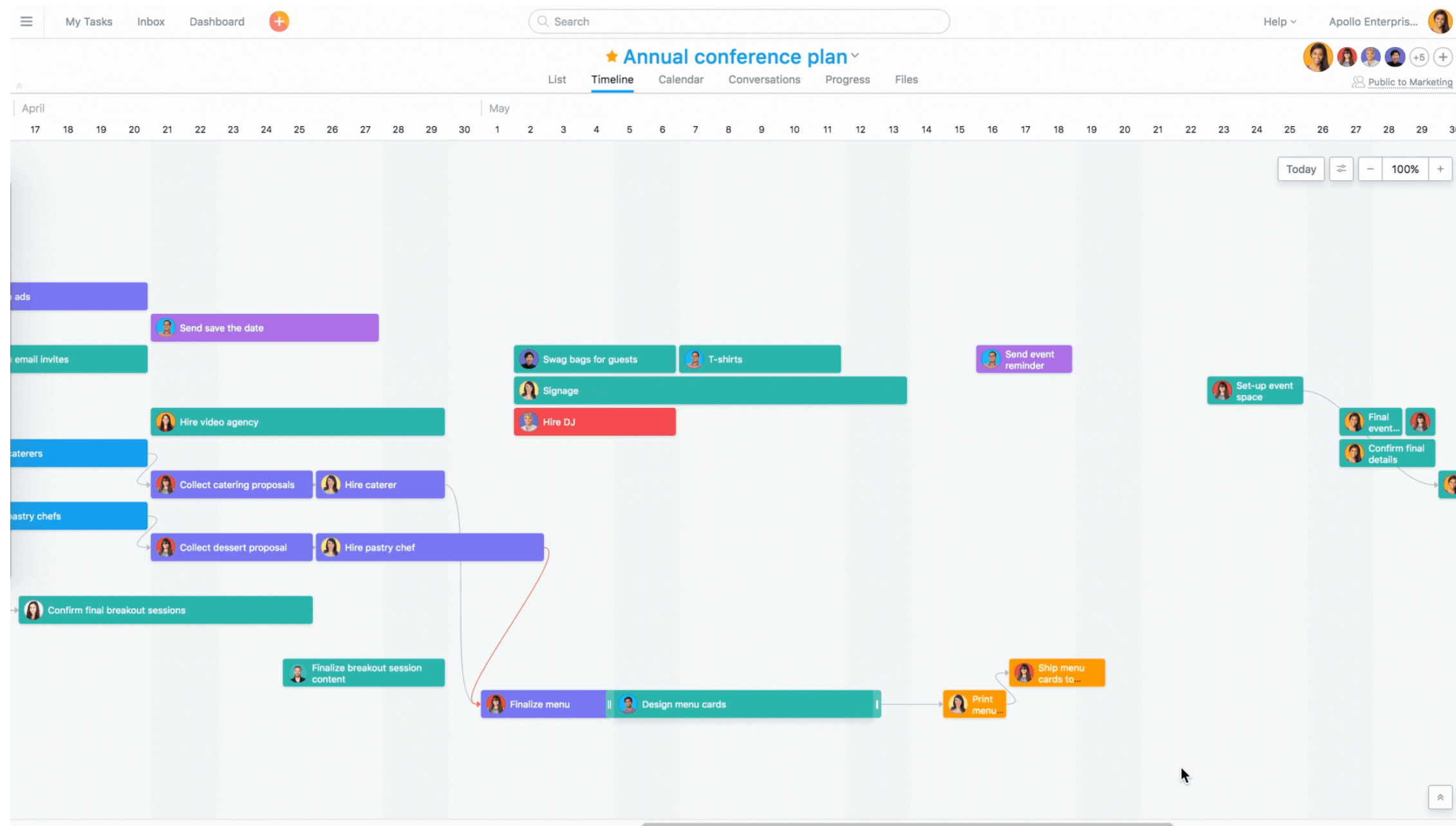
# 2006-Today



Google

# Asana

# 2008-Today



Dustin Moskovitz and Justin Rosenstein



The screenshot shows the GitHub homepage with a dark navigation bar at the top containing the GitHub logo, a search bar, and links for Pull requests, Issues, Marketplace, and Explore. A notification banner at the top left states: "Our new Terms of Service and Privacy Statement are in effect." Below this is a "Repositories" sidebar with a search bar and a list of repositories: transfluxus/drop, zrispo/wick-editor, supercgeek/tangible-pong, supercgeek/2j2c\_Stanford, supercgeek/2j2c, supercgeek/RobotHuman, and katjacburgess/Python. The main content area, titled "Browse activity", displays a list of repository events:

- zhxnlai** starred **Kotlin/kotlinx.serialization** 5 hours ago. The repository is for Kotlin cross-platform / multi-format serialization, has 585 stars, and was updated on Jun 25.
- bertfreudenberg** forked **bertfreudenberg/Clembot** from **TrainingB/Clembot** 7 hours ago. The repository is for Python, has 1 star, and was updated on Jun 25.
- zrispo** starred **scottoffen/jquery.toaster** a day ago. The repository is a jQuery plugin for displaying customizable toast notifications via Bootstrap alerts, has 61 stars, and was updated on Jun 24.
- danzeeman** forked **danzeeman/microsoft-drop-ice** from **selfagency/microsoft-drop-ice** 2 days ago. The repository is for telling Microsoft to drop ICE as a client or lose us as GitHub users, has 443 stars, and was updated on Jun 25.

Tom Preston-Werner for GitHub Inc (Now Microsoft)

# Wave

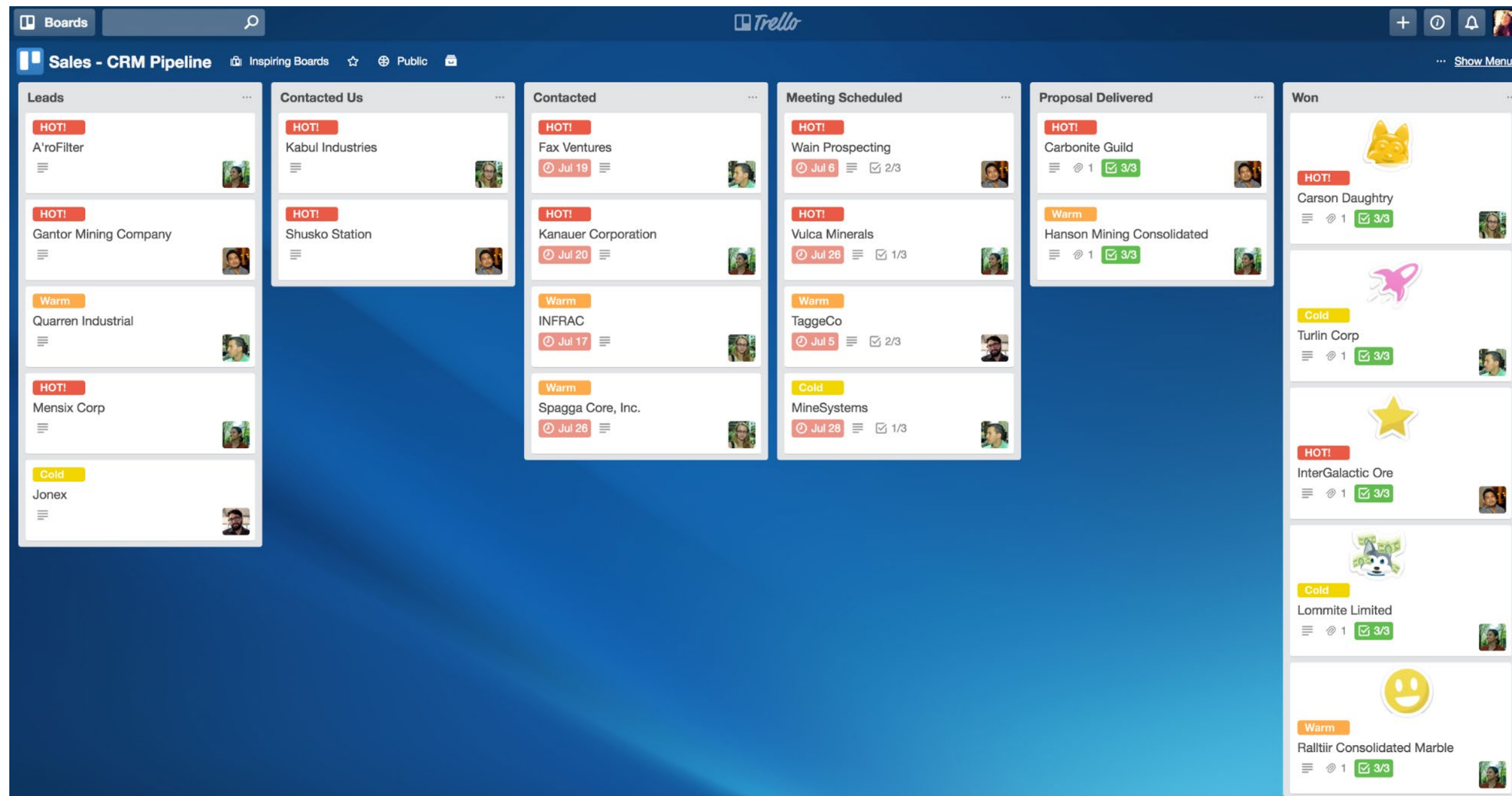
2009-2011



Google

# Trello

# 2011-Today



Joel Spolsky at Fog Creek (Later Atlassian)

# Notion

# 2016-Today

☰  Tasks & Issues

Share Updates Favorite ...

## 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



+ Add an Item

📄 Fix keyboard shortcuts

 Bug  Important

 Brian Park

📄 Desktop push notifications

 Android  QA

 Shawn Sanchez

+ Add an Item

Offline mode polish

 iOS  QA

 Leslie Jensen


Add avatars

 Polish

 Leslie Jensen

+ Add an Item

Rename archive to trash

 Polish

 Shawn Sanchez

+ Add an Item

# Bits to Atoms

Computation meets Business & Industry

# Spreadsheets

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

# VisiCalc

1979-1983

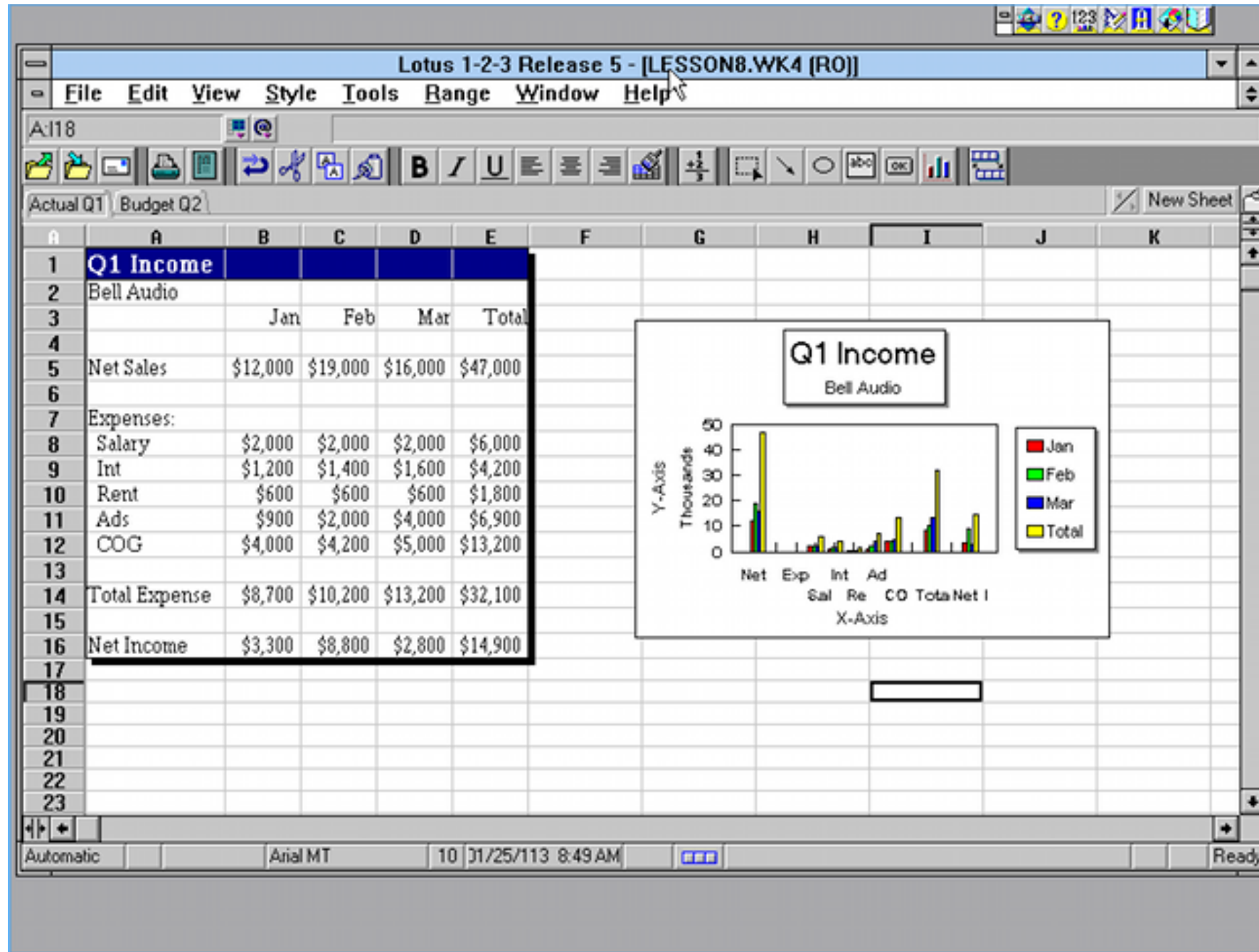
The screenshot shows a VisiCalc spreadsheet window titled "C11 (L) TOTAL" with a cursor at cell C1. The spreadsheet has four columns: A (ITEM), B (NO.), C (UNIT), and D (COST). The data is as follows:

A	B	C	D
ITEM	NO.	UNIT	COST
MUCK	4	12.50	50.00
HORN	1	1.00	1.00
TONER	25	4.90	122.50
SNUFF	2	4.90	9.80
SUBTOTAL			1315.50
9.75% TAX			1282.06
TOTAL			14438.16

Software Arts

# 1-2-3

# 1983-2002



Lotus Software



# Excel

# 1987-Today

Microsoft Excel - pairs.xls

File Edit View Insert Format Tools Data Window Help

A1 = ^No.

	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	8	10	18	21	22			

Ready

Microsoft

# Numbers

# 2005-Today

The screenshot shows an Apple Numbers spreadsheet titled "River Rafting Budget". The interface includes a menu bar with options like View, Sheet, Tables, Function, Sort & Filter, Charts, Text Box, Shapes, Comment, Inspector, Media, Colors, and Fonts. A sidebar on the left contains a "Sheets" pane with "Participants & Budget" (Participants, Budget, Expenses) and "Vendor Options" (Rafting, Camping, T-shirts), and a "Styles" pane with various fill and header styles. The main content area features a title "River Rafting Budget" and two tables. The "Participants" table lists 9 individuals with columns for Last name, First name, RSVP, Phone, Email address, Paid, and Amount due. The "Expenses" table is a pie chart showing the distribution of costs: Rafting (45%), Camping (16%), Dinner (14%), Beverages (10%), Transportation (8%), and T-shirts (7%). Below the tables are three images: a photo of white-water rafting on the Klamath River, a map showing the route from I-80 to HWY 89 and HWY 50 to a campground, and a small table summarizing the expenses. The small table has columns for Item, Per person, and Total cost, with a total row showing a sum formula.

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) 555-7834	Nancy_D_Nguyen@postlive.net	\$200.00	\$165.00
Parker	Seth	✓	(707) 555-7890	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>		<b>9</b>			<b>\$1,695.00</b>	<b>\$1,590.00</b>

Item	Percentage
Rafting	45%
Camping	16%
Dinner	14%
Beverages	10%
Transportation	8%
T-shirts	7%

	A	B	C
1	Item	Per person	Total cost
2	Rafting	\$165.00	\$1,485.00
3	Camping	\$60.00	\$540.00
4	Dinner	\$50.00	\$450.00
5	Beverages	\$36.00	\$324.00
6	Transportation	\$30.00	\$270.00
7	T-shirts	\$24.00	\$216.00
8	<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

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

ANSWERS

Ask a question about your data

- Distribution of Winner
- Correlation between Teams and Matches
- Average of Teams

FORMATTING

Alternating colors for A1:J21 [EDIT](#)

ANALYSIS

Teams, Matches and Goals scored

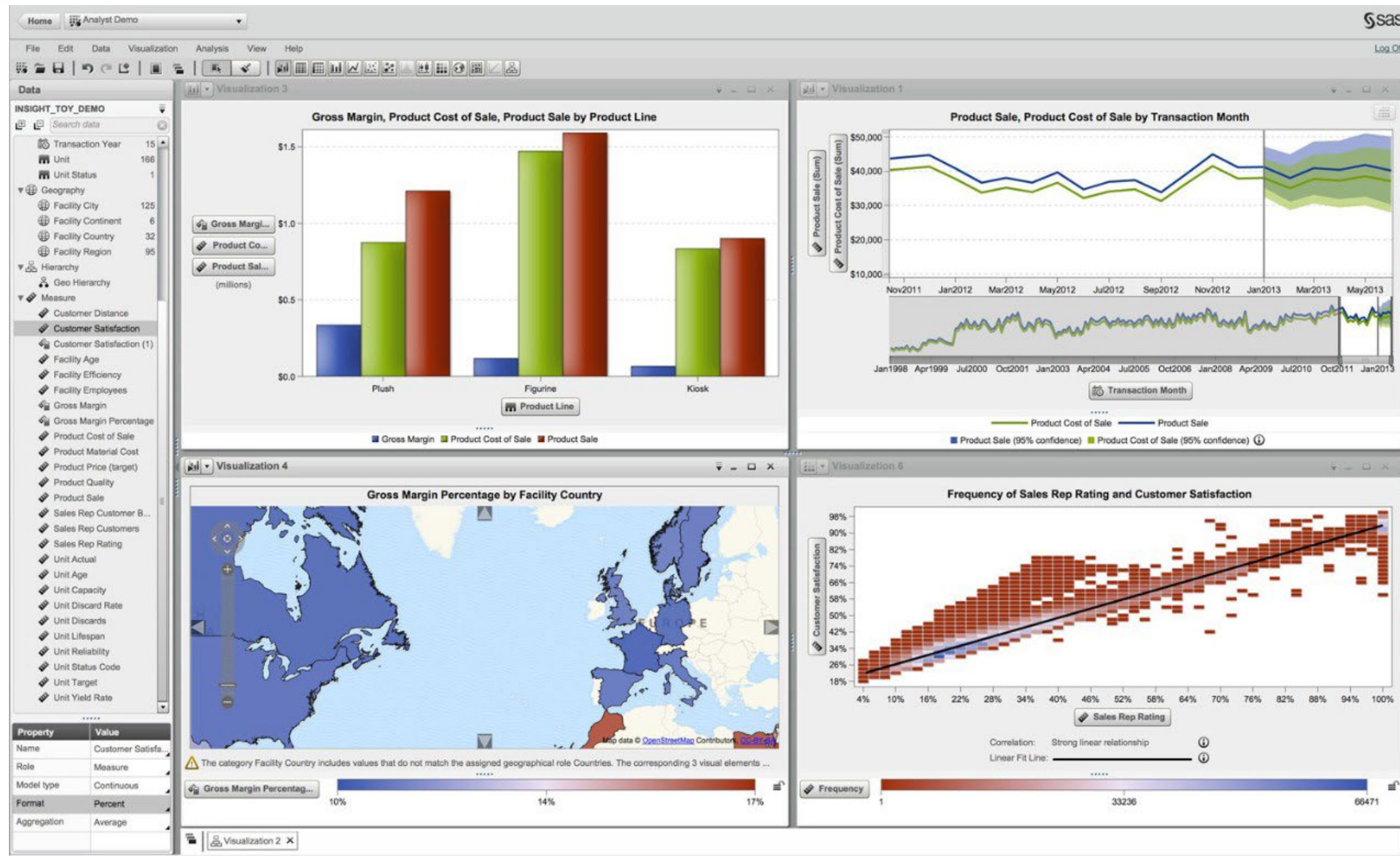
World Cup

Explore

Google

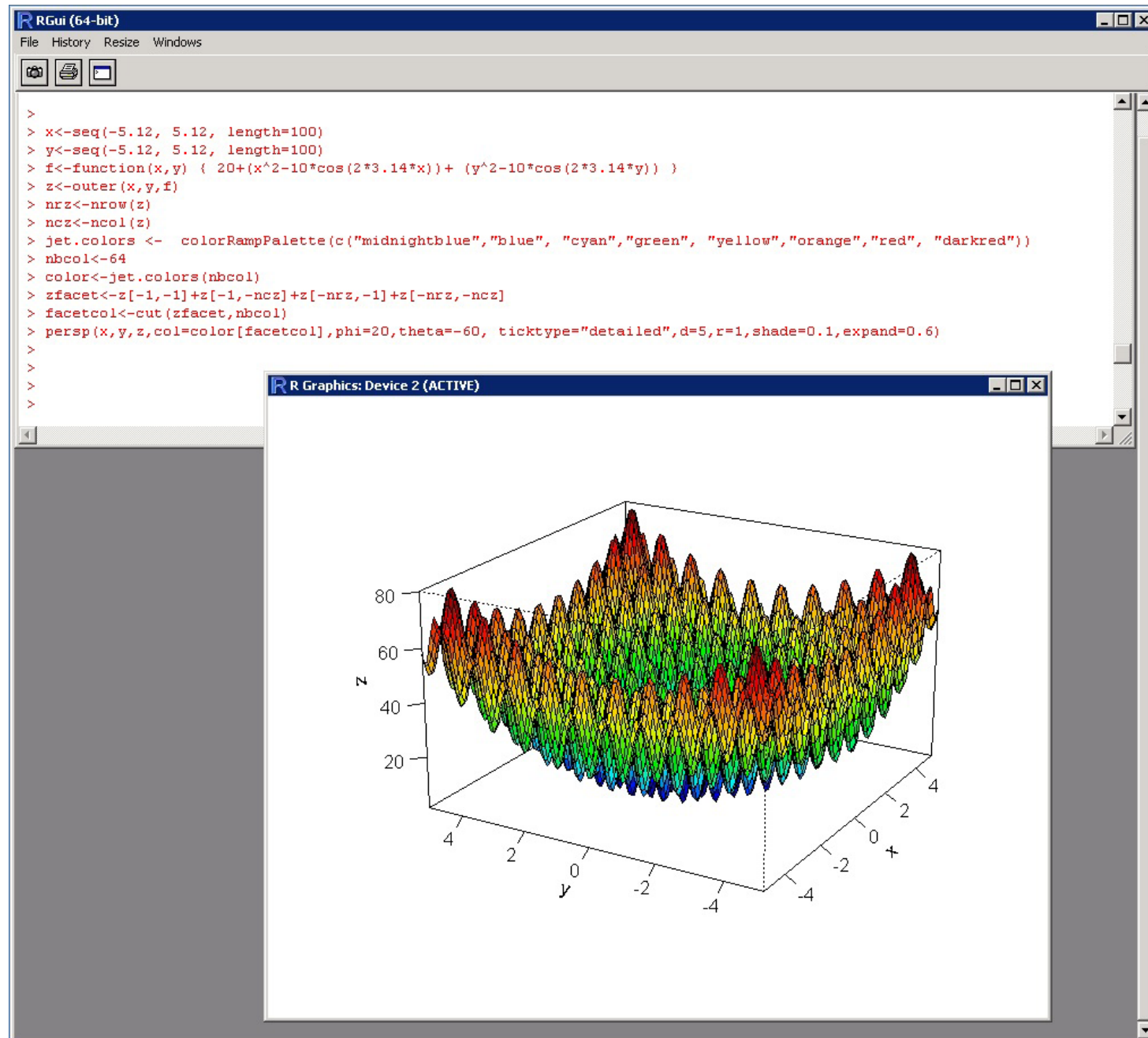
# **Data Analysis**

From Simple Languages to  
Expansive Environments and  
Hosted Compute Platforms



# R

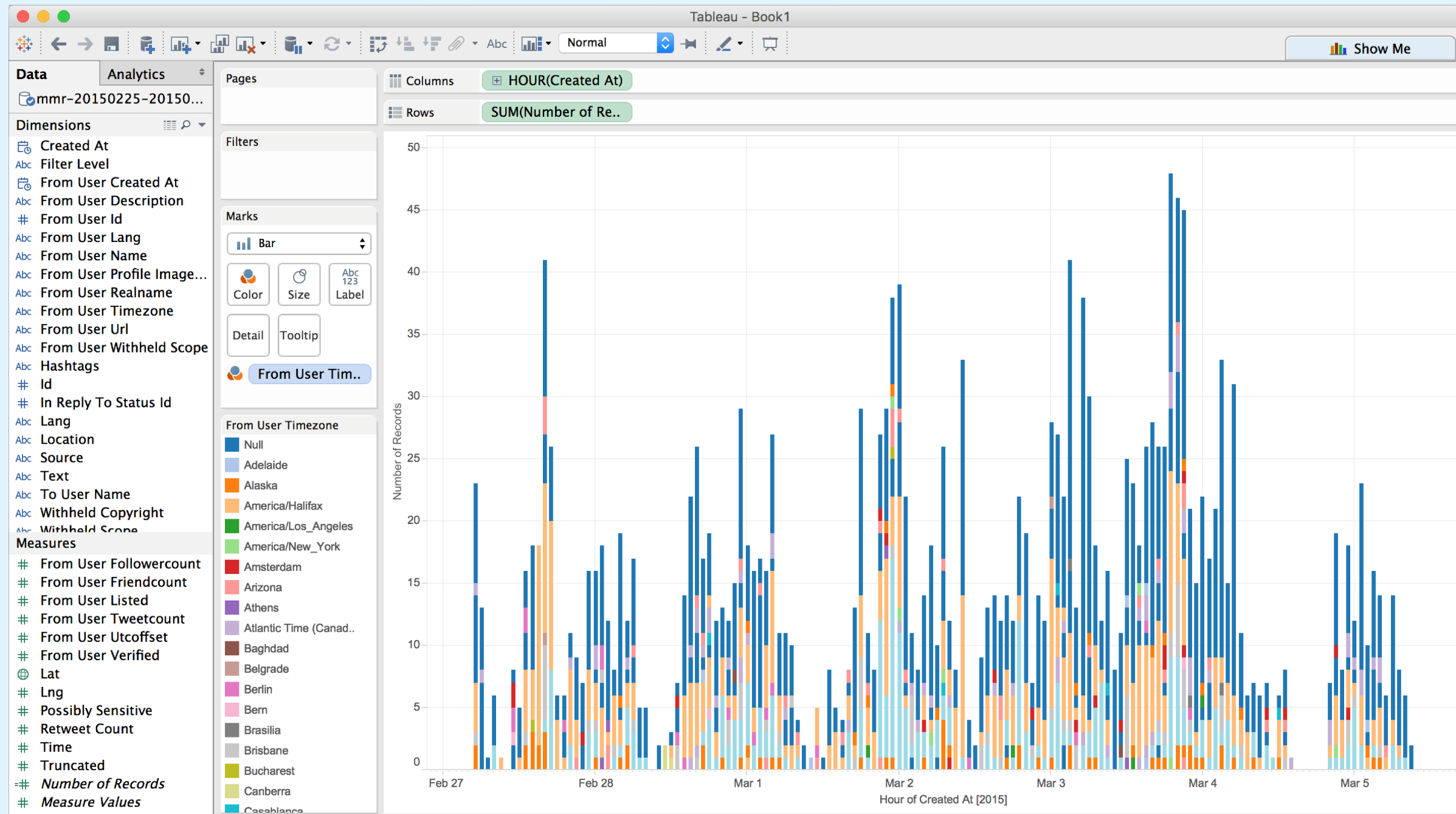
# 1993-Today



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

The screenshot displays the RapidMiner workflow editor. The main process area contains the following operators in sequence: Retrieve, Normalization, MissingValue, LibSVM, Nominal2Binominal, and Nominal2Numeric. The LibSVM operator is currently selected, and its configuration parameters are visible in the right-hand pane:

- svm type: C-SVC
- kernel type: rbf
- gamma: 22644346174132
- C: 85795883818439
- epsilon: 0.0010
- calculate confidences
- 4 hidden expert parameters

At the bottom of the interface, a 'Problems' pane indicates two potential issues:

Message	Fixes	Location
Attribute filter does not match any attributes.	Select all attributes.	Nominal2Binominal...
Attribute filter does not match any attributes.	Select all attributes.	Nominal2Numeric...

RapidMiner



# Knime

# 2006-Today

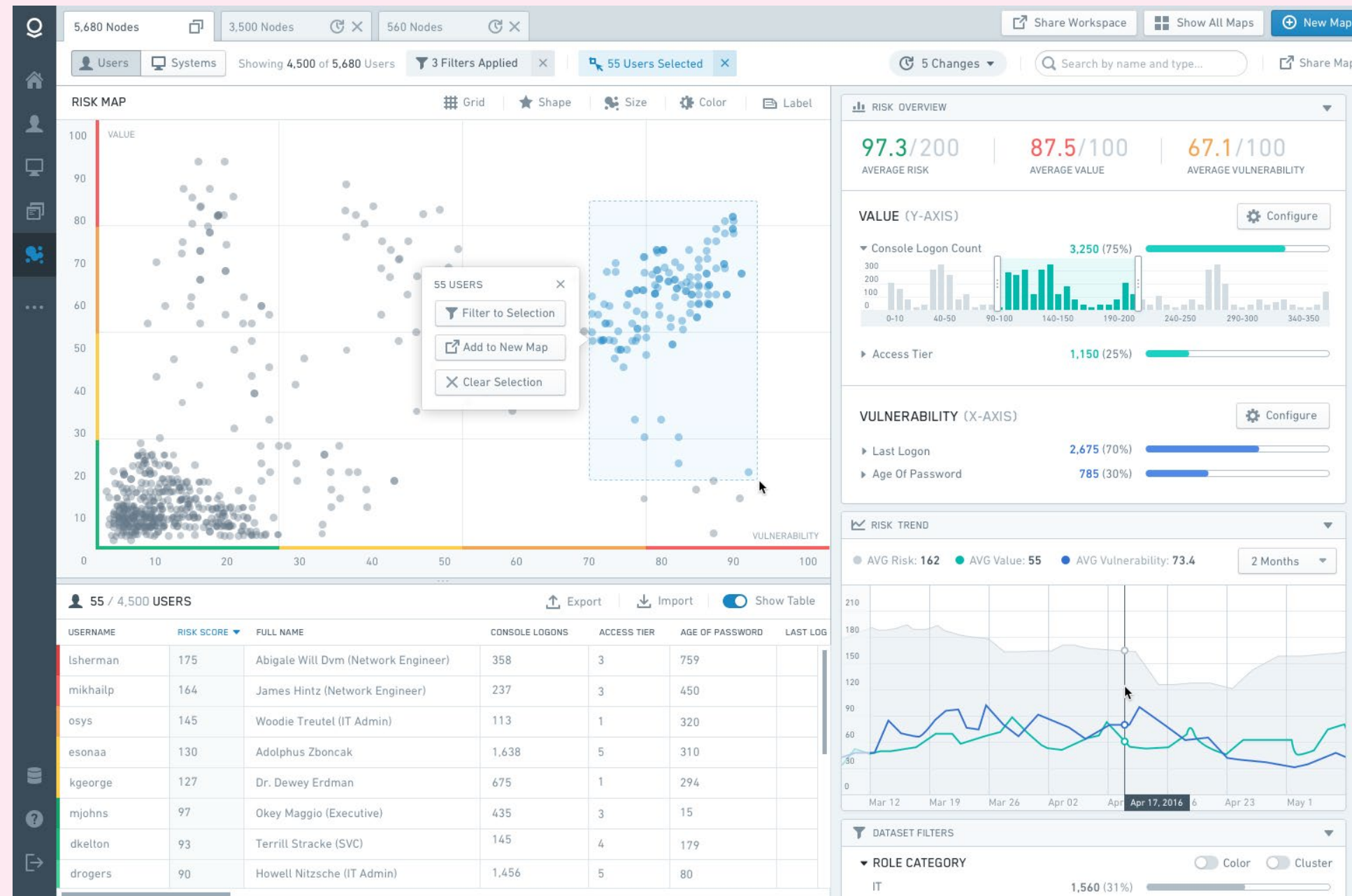
The screenshot displays the KNIME software interface with several key components:

- Workflow Editor:** A central workspace showing a workflow with nodes: File Reader (Node 1), Color Manager (Node 2), Partitioning (Node 5), Decision Tree Learner (Node 3), Interactive Table (Node 14), Decision Tree Predictor (Node 4), Scorer (Node 8), Column Filter (Node 11), and another Column Filter (Node 13).
- Decision Tree View:** A tree diagram showing classification rules. The root node splits on 'capital-gain <= 6.849'. Subsequent nodes split on 'age <= 27', 'education-num <= 12', and 'hours-per-week <= 40'. The final leaf nodes represent predicted classes: 'class <= 50K' and 'class > 50K'.
- Scatter Plot:** A plot titled '0:2:15 : Scatterplot' showing data points colored by class. The x-axis ranges from 0 to 90, and the y-axis ranges from 9.0 to 14.2.
- Data Table:** A table titled 'Frame1 [51 x 15]' showing a subset of data rows with columns: Row ID, age, workclass, fnlwgt, education, education-num, marital-status, occupation, relation, and race.
- Node Repository:** A sidebar on the left listing various data processing and analysis nodes such as File Reader, Color Manager, Partitioning, Decision Tree Learner, Interactive Table, Scorer, Column Filter, and various visualization nodes like Scatter Plot, Histogram, and Pie Chart.

KNIME.com AG

# Foundry

# -Foundry

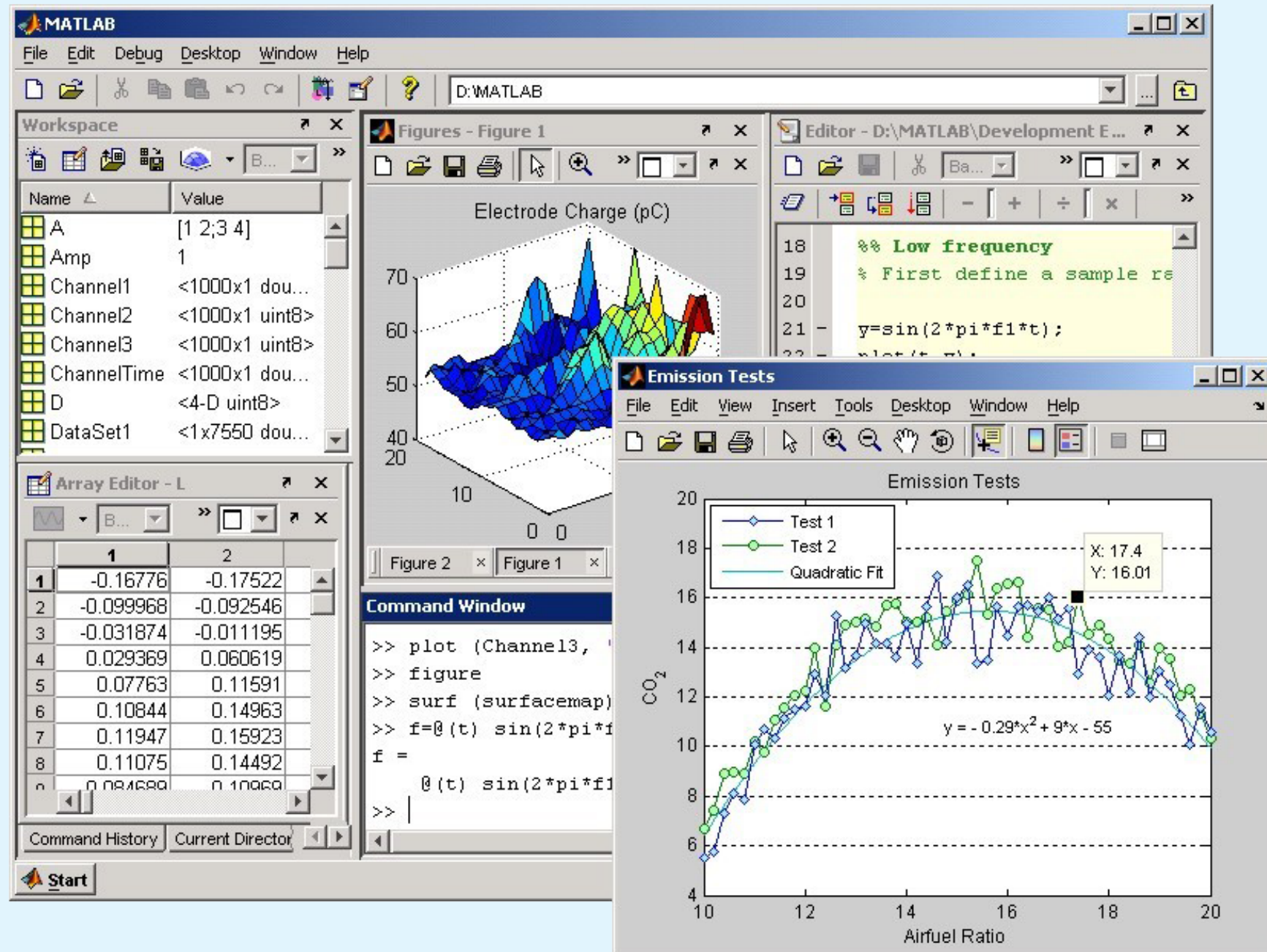


Palantir

# Simulation

# MATLAB

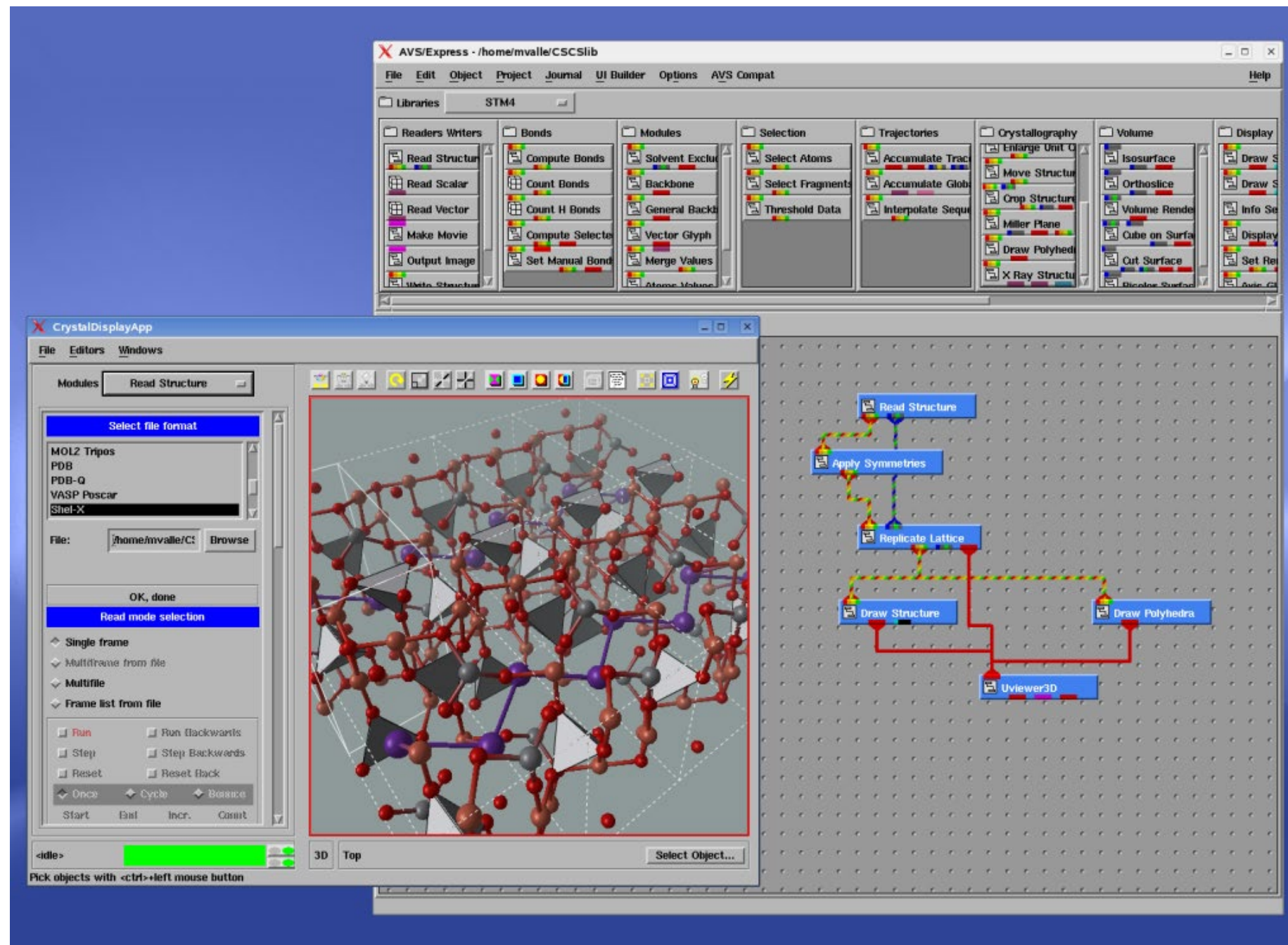
# 1984-Today



Cleve Moler, Steve Bangert and Jack Little for MathWorks

# AVS Express

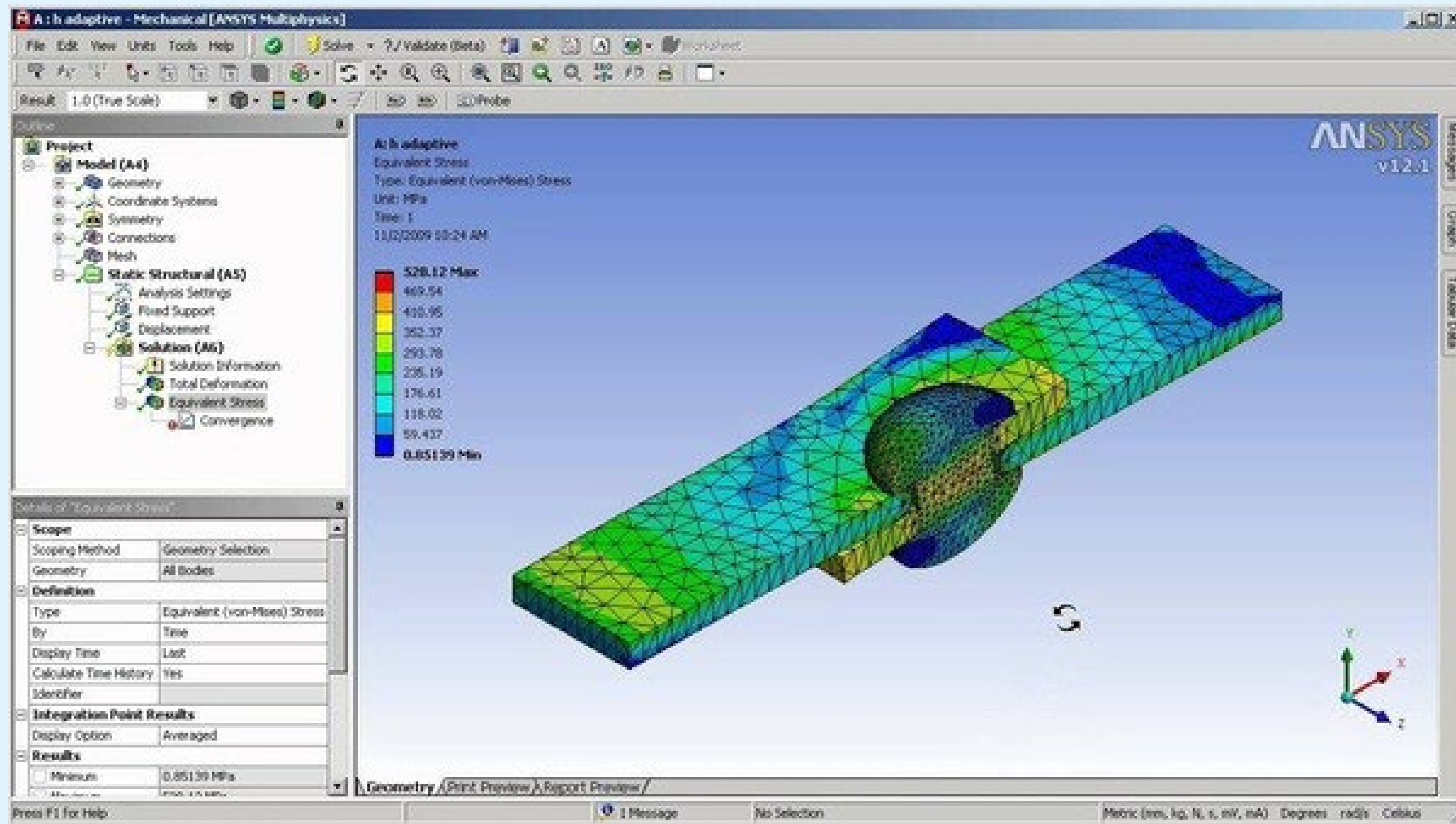
1991-Today



Advanced Visual Systems Inc

# Ansys

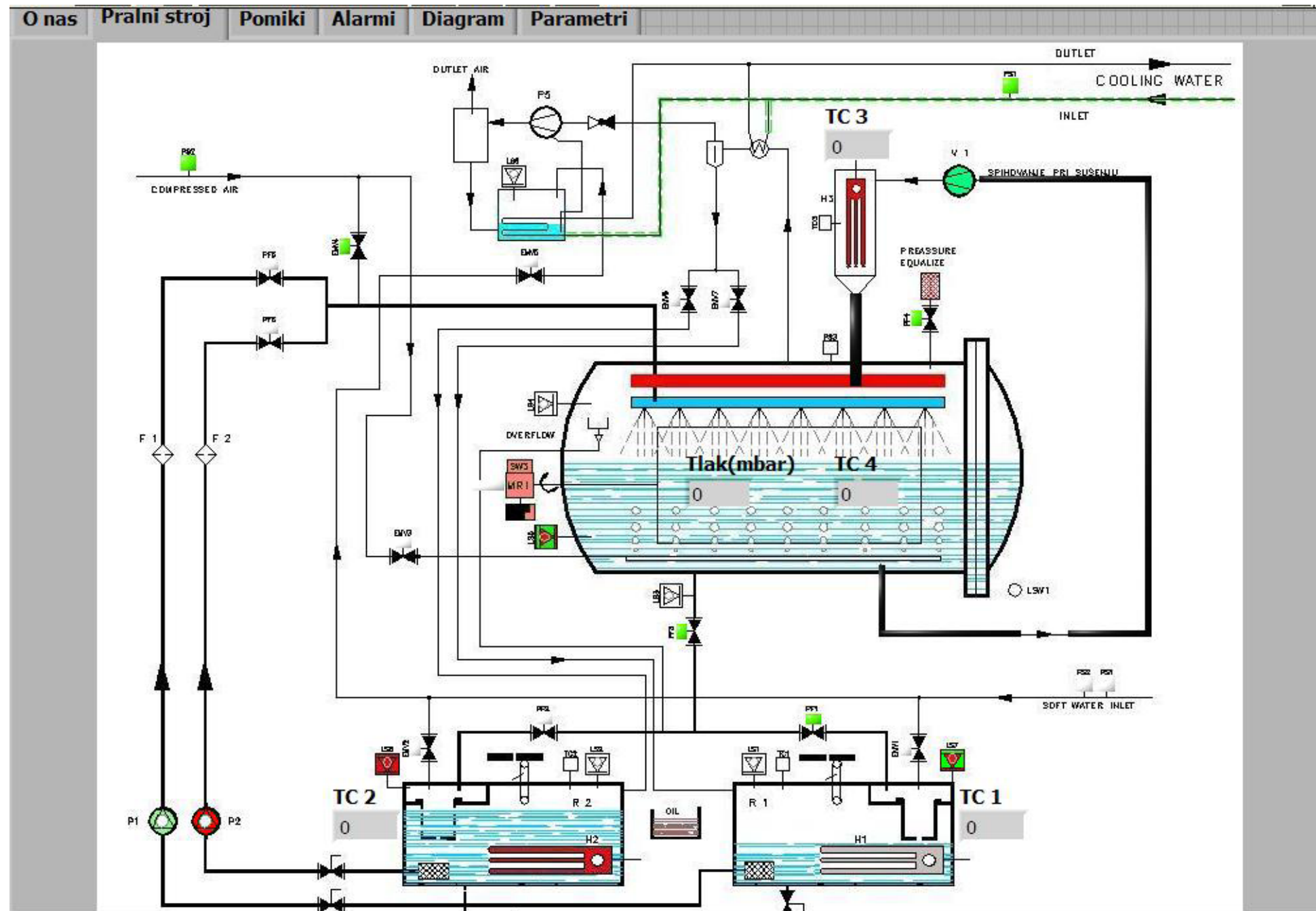
# 1970-Today



ANSYS Inc

# LabView

# 1986-Today



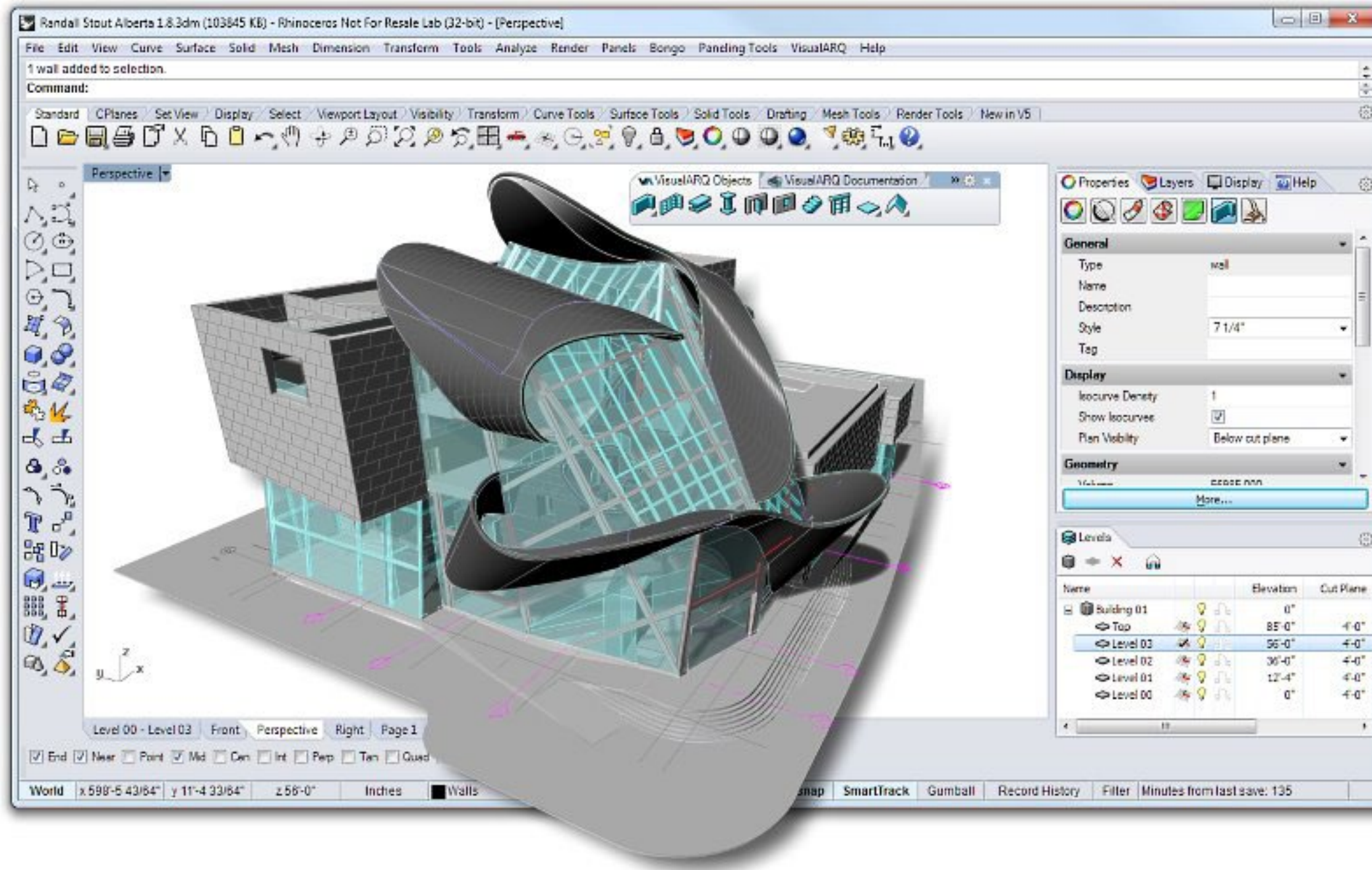
National Instruments

# 3D (Solid Geometry) Modeling



# Rhino

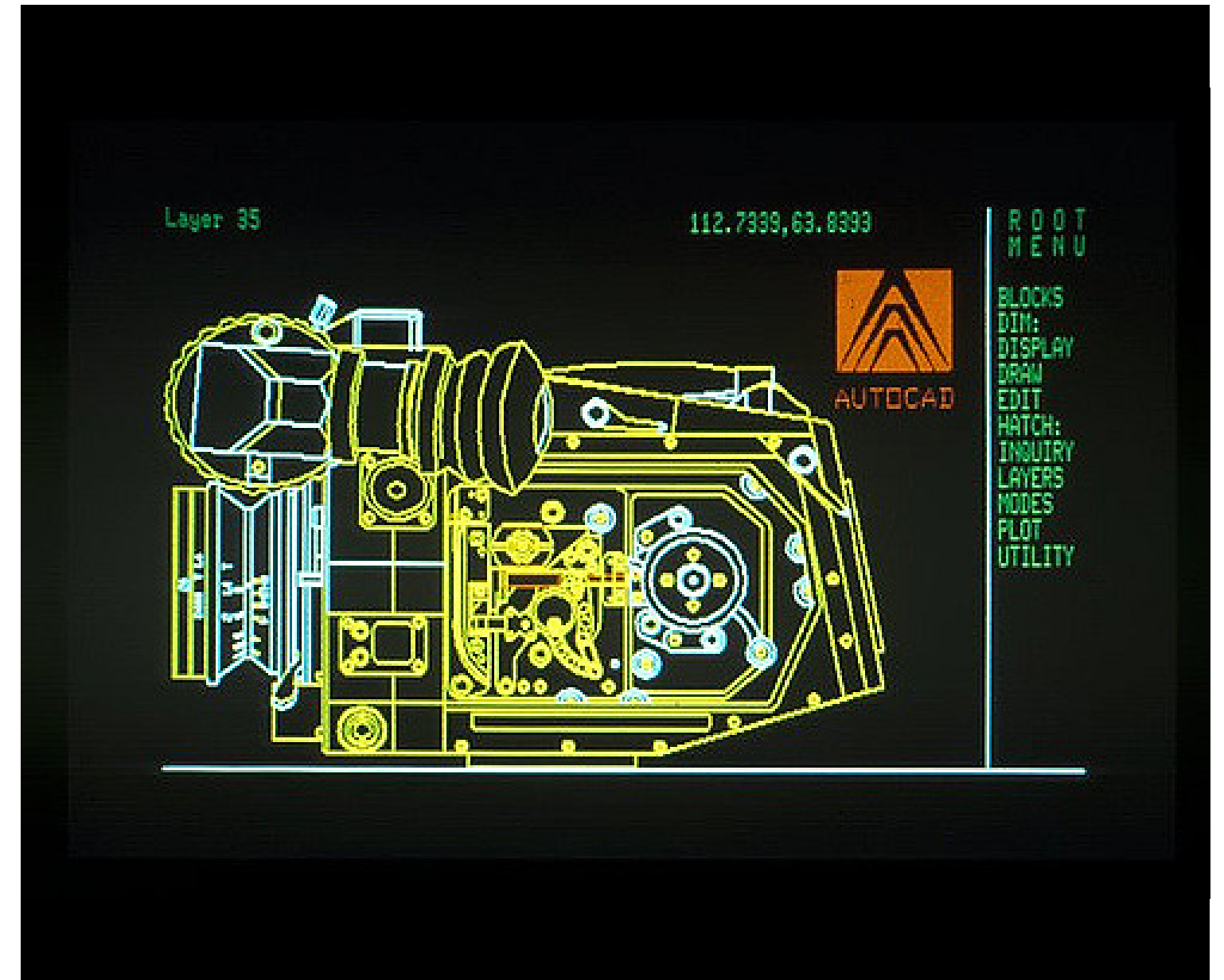
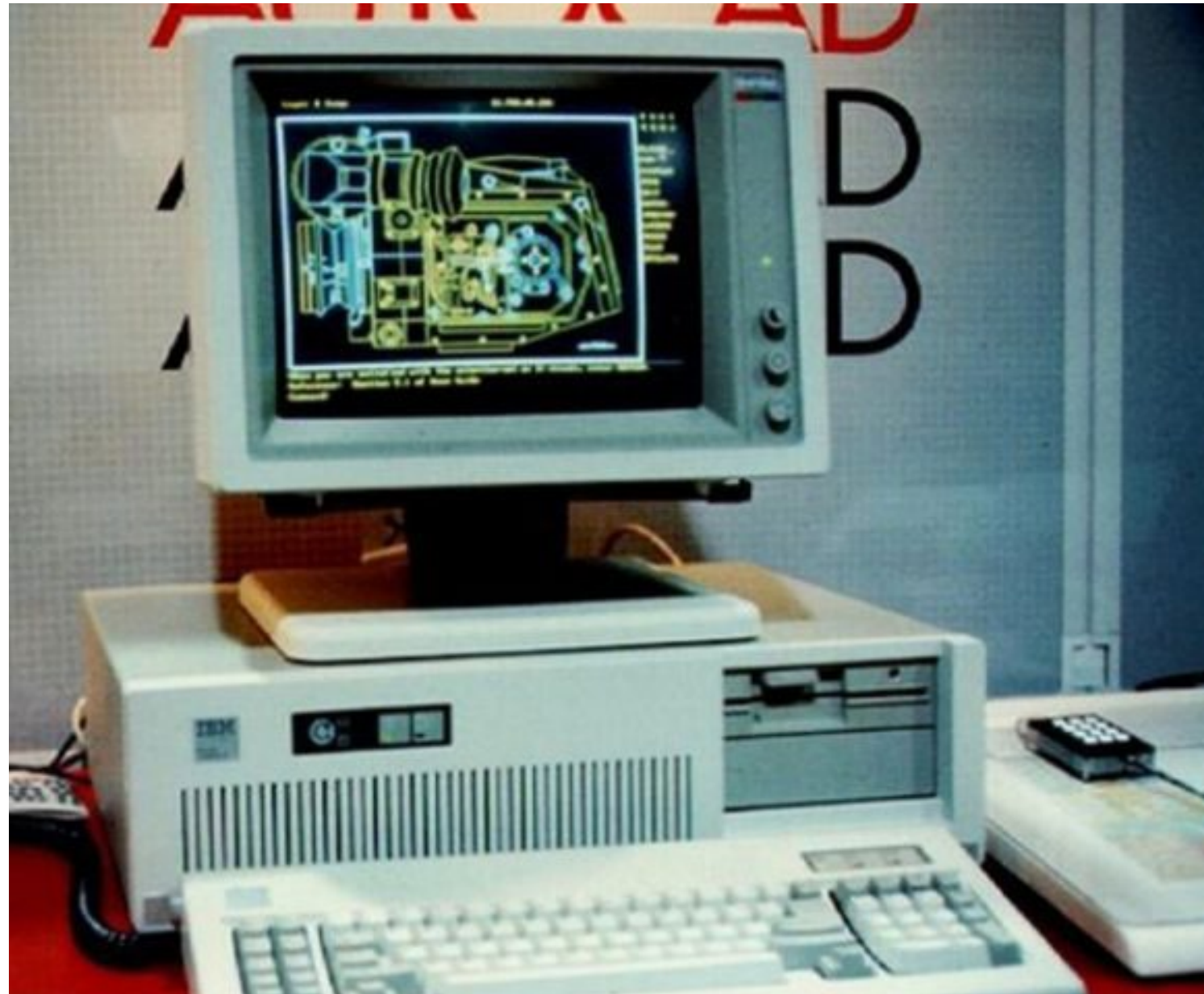
# 1980-Today



Robert McNeel & Associates

# AutoCAD

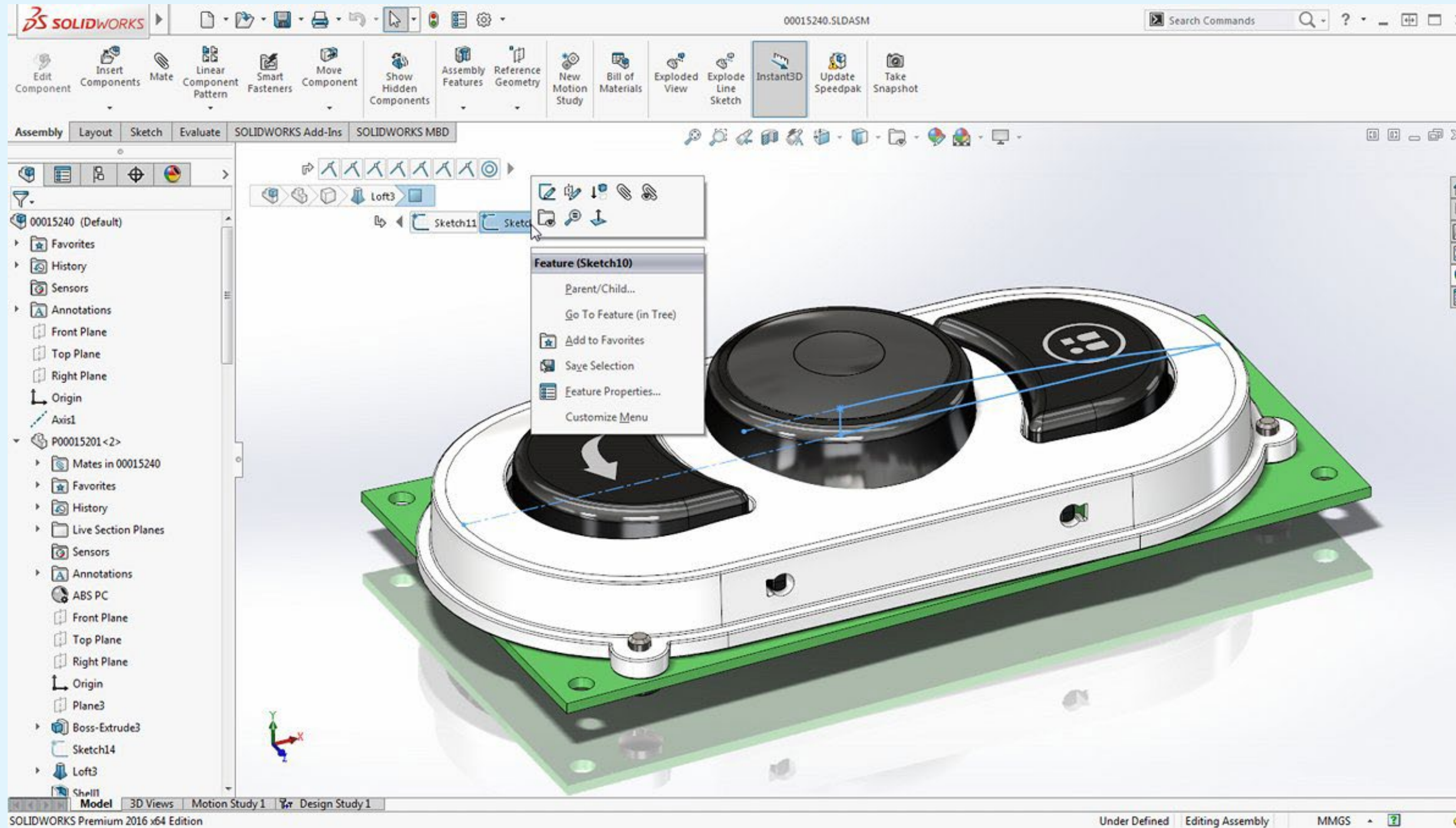
# 1982-Today



AutoDesk

# SolidWorks

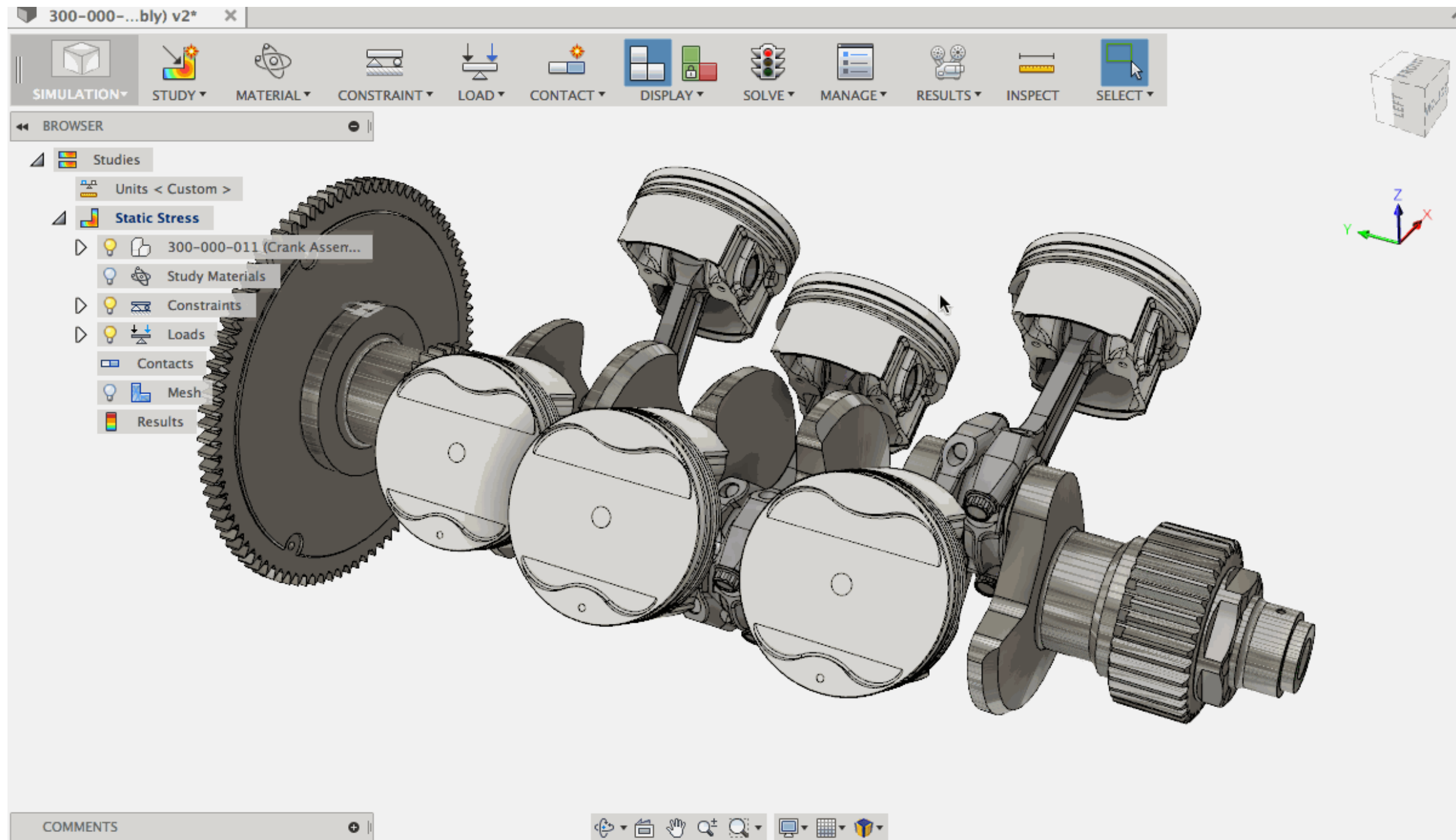
1995-Today



Dassault Systèmes

# Fusion 360

2013-Today



Autodesk

# Design Machine

## Creative Tools for Everyone

# Graphics

## Direct-Manipulation + Computer Graphics

# Sketchpad

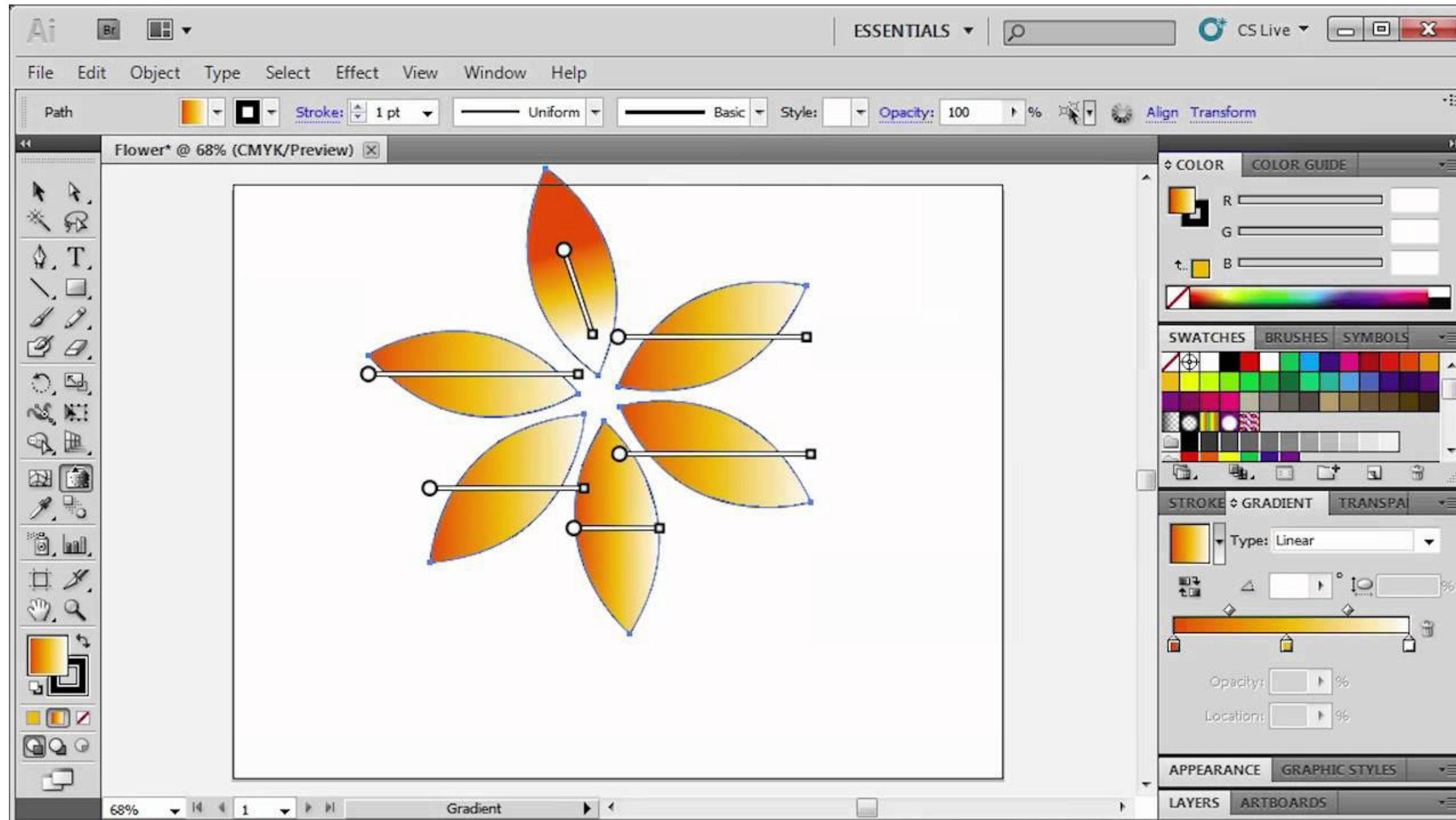
1963



Ivan Sutherland

# Illustrator

1987-Today

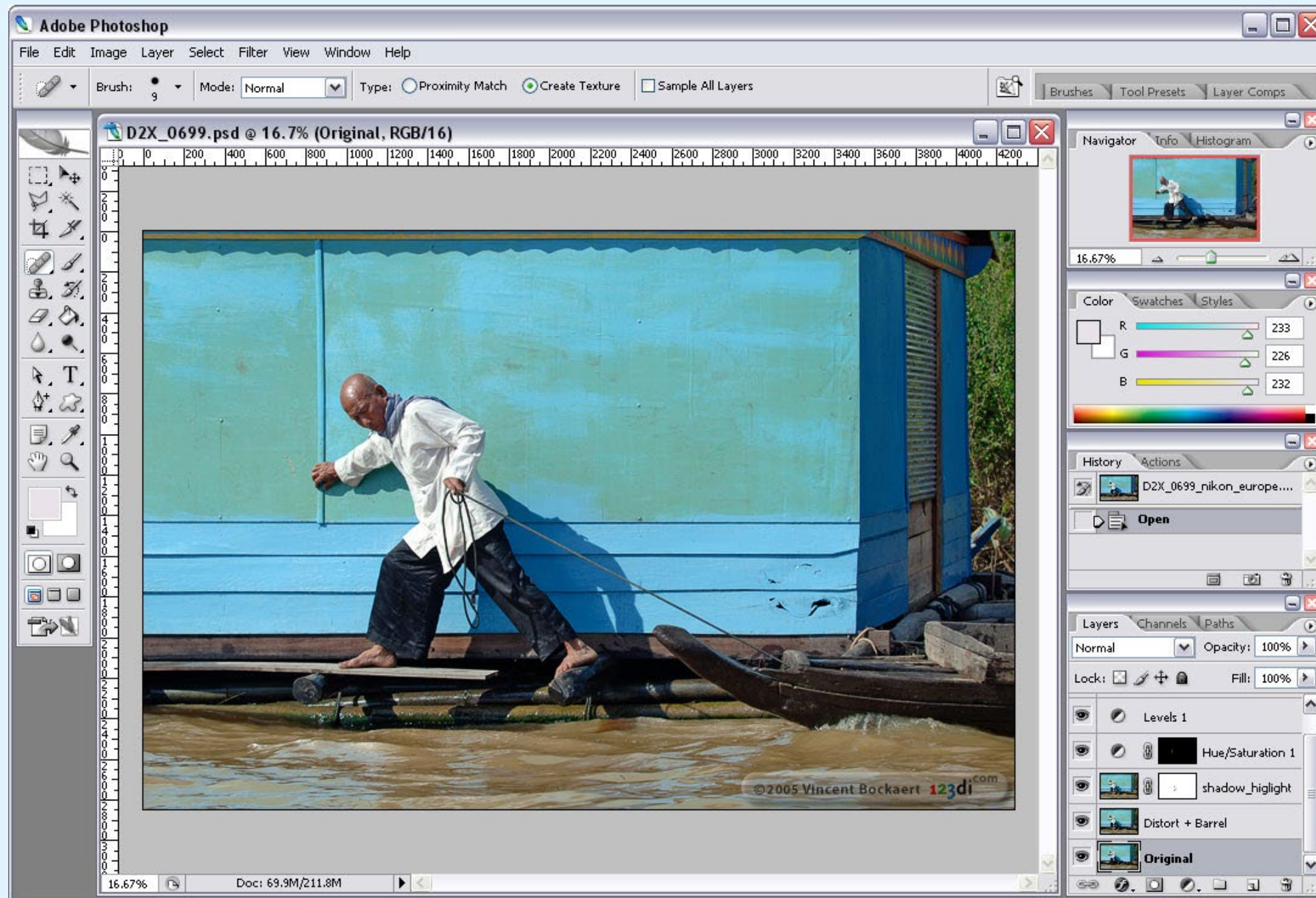


Adobe Systems



# Photoshop

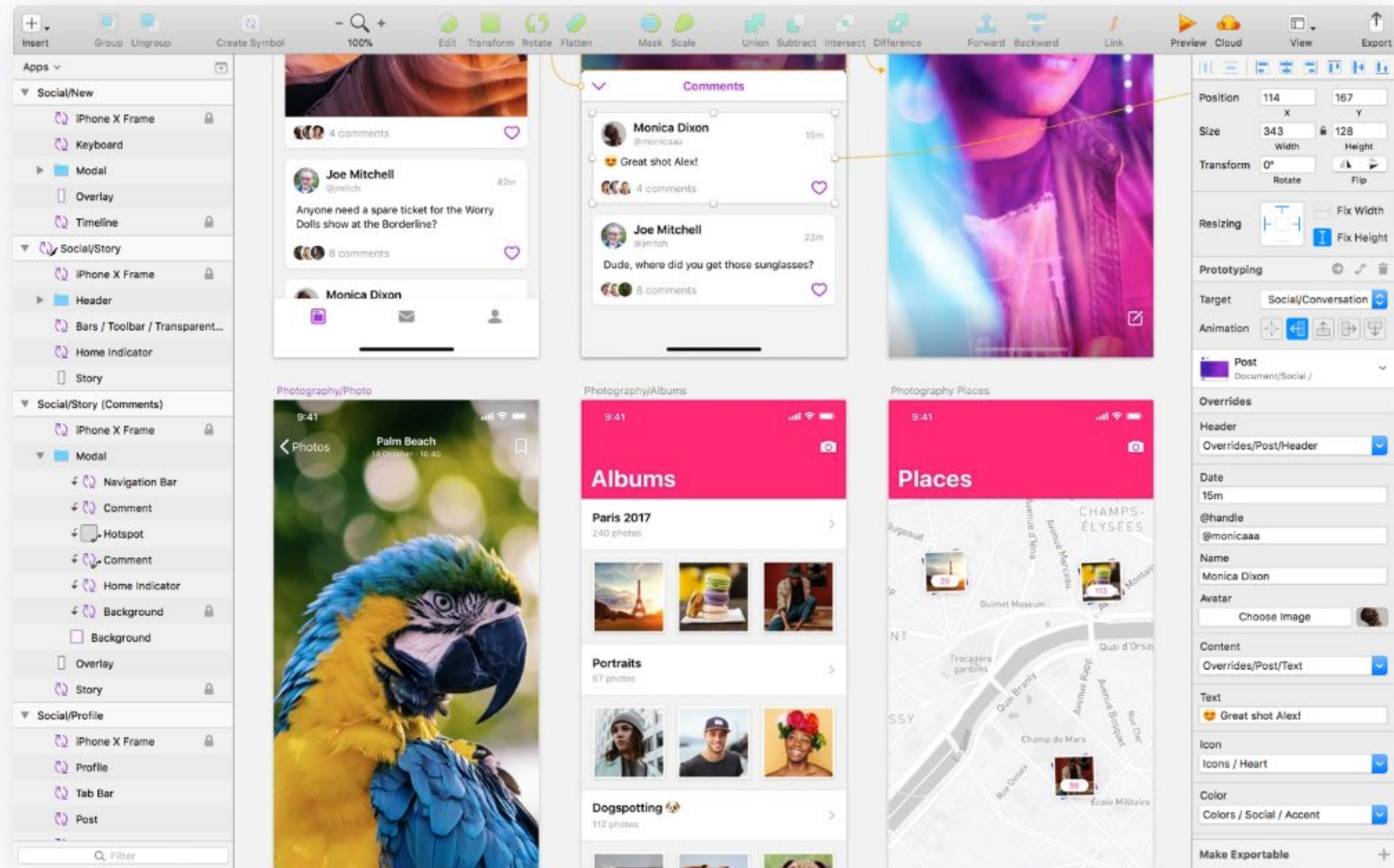
# 1990-Today



Adobe Systems

# Sketch

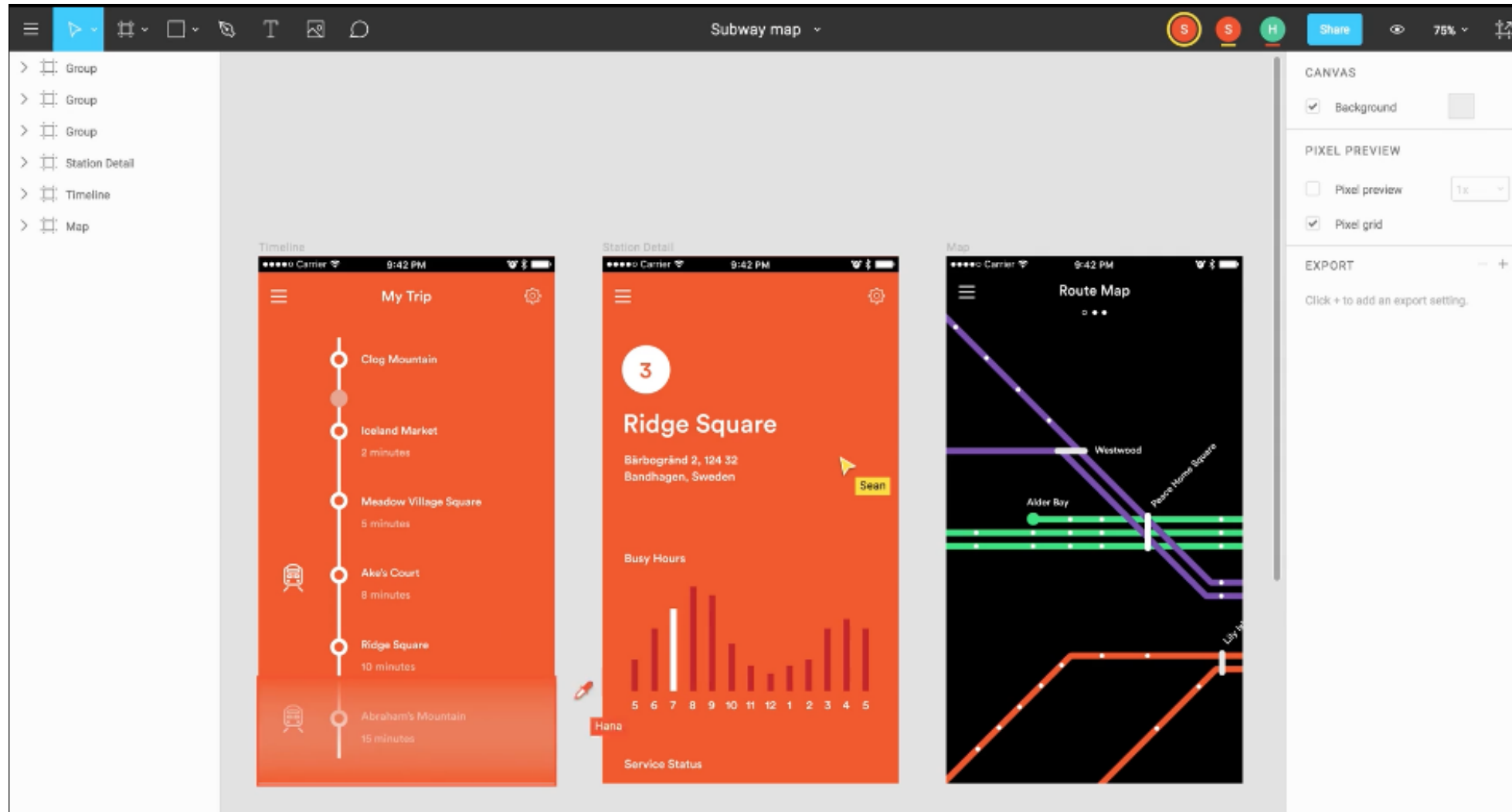
# 2010-Today



## Bohemian Coding

# Figma

# 2016-Today



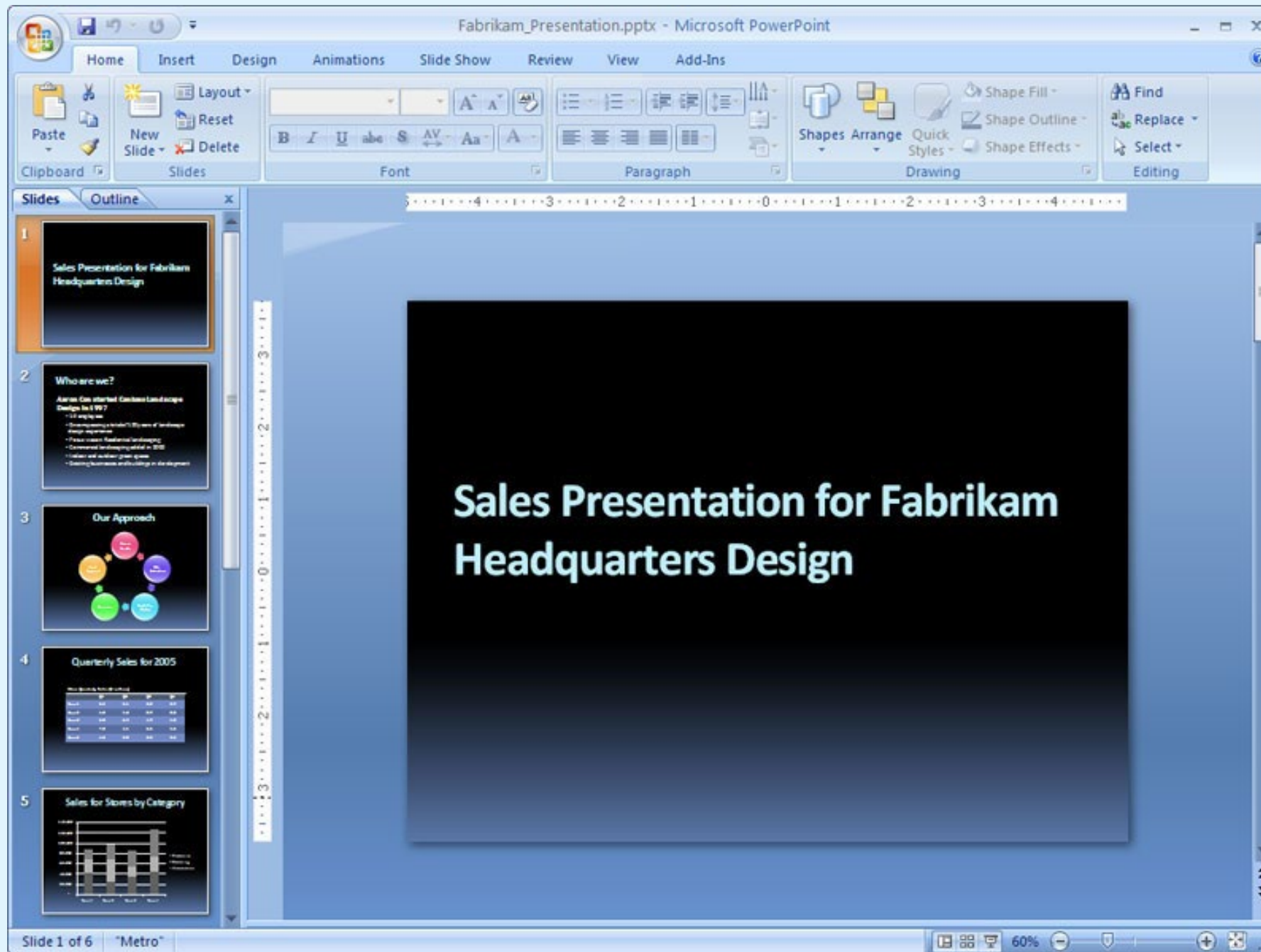
## Figma

# Presentation

From Outline to Slides to Demo

# Powerpoint

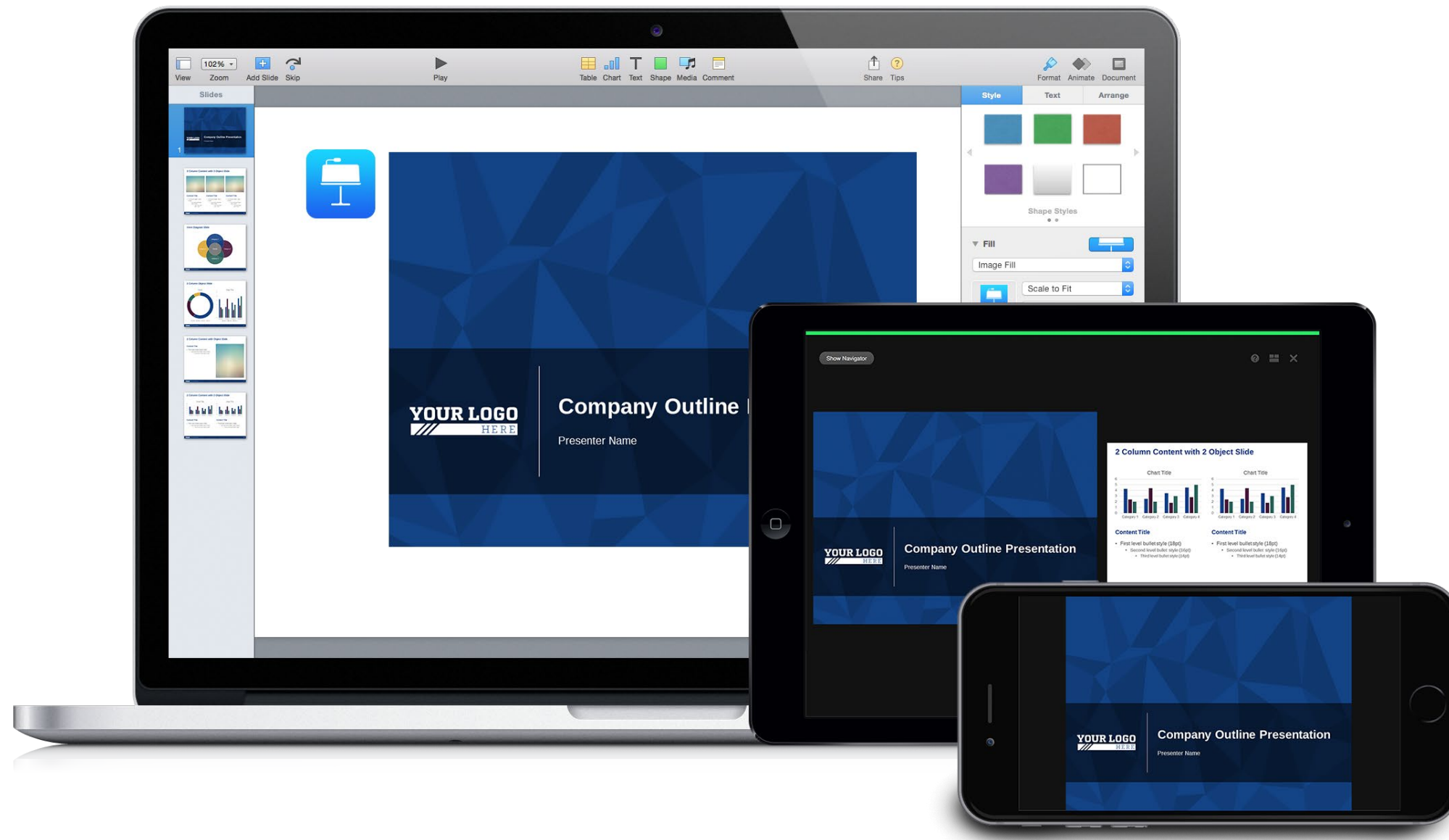
1987-Today



Microsoft

# Keynote

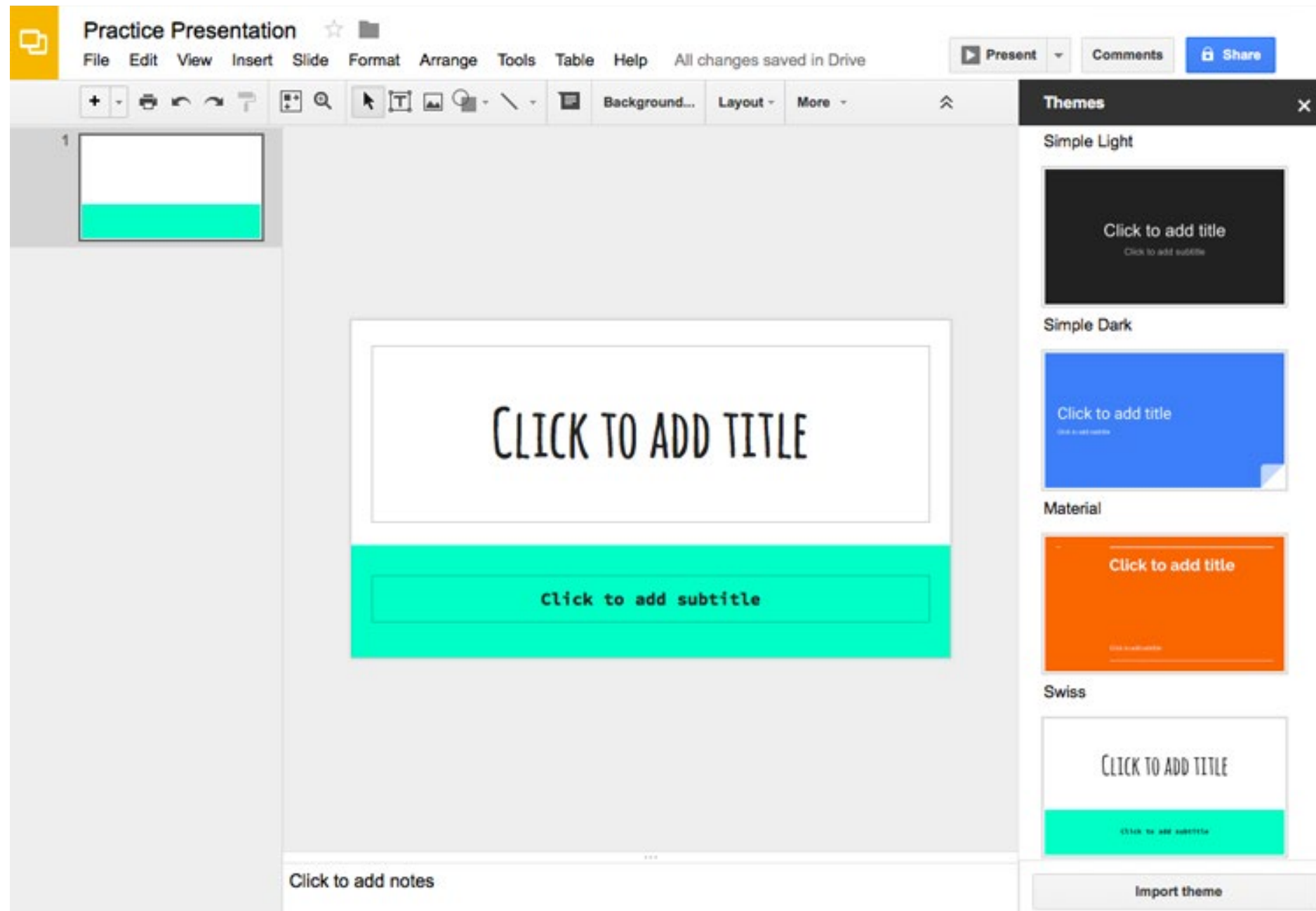
# 2003-Today



Apple

# Slides

# 2006-Today



Google

Prezi and a Decade of Press Suppression

Path

Save Meeting Print Help Exit

Continue Your Prezi skill Tutorial

Why control the Philippines during Martial Law?

Why no vCom?

Purpose journalism fits with the propaganda machine of the government. It was used to camouflage, to write wash the

Pre-EDSA Press

- Crony Press
- Alternative Press
- Provincial Press
- Tabloid Press

Crony Press

derogation for any newspaper led or by supporting Marcos or his PDP political party.

Alternative Press

extremely critical of the Marcos regime and blatantly called for its removal and replacement.

Provincial Press

comprised of newspapers published outside Metro Manila.

FM DECLARES MARTIAL LAW

But civilian gov't still functions; no military takeover

Martial Law declared, 1981

TULOY PA RANG LABAN August 21, 1983

1983-1986

RP Battles with Dandruff

It's all over; Marcos flees!

Department of Public Information

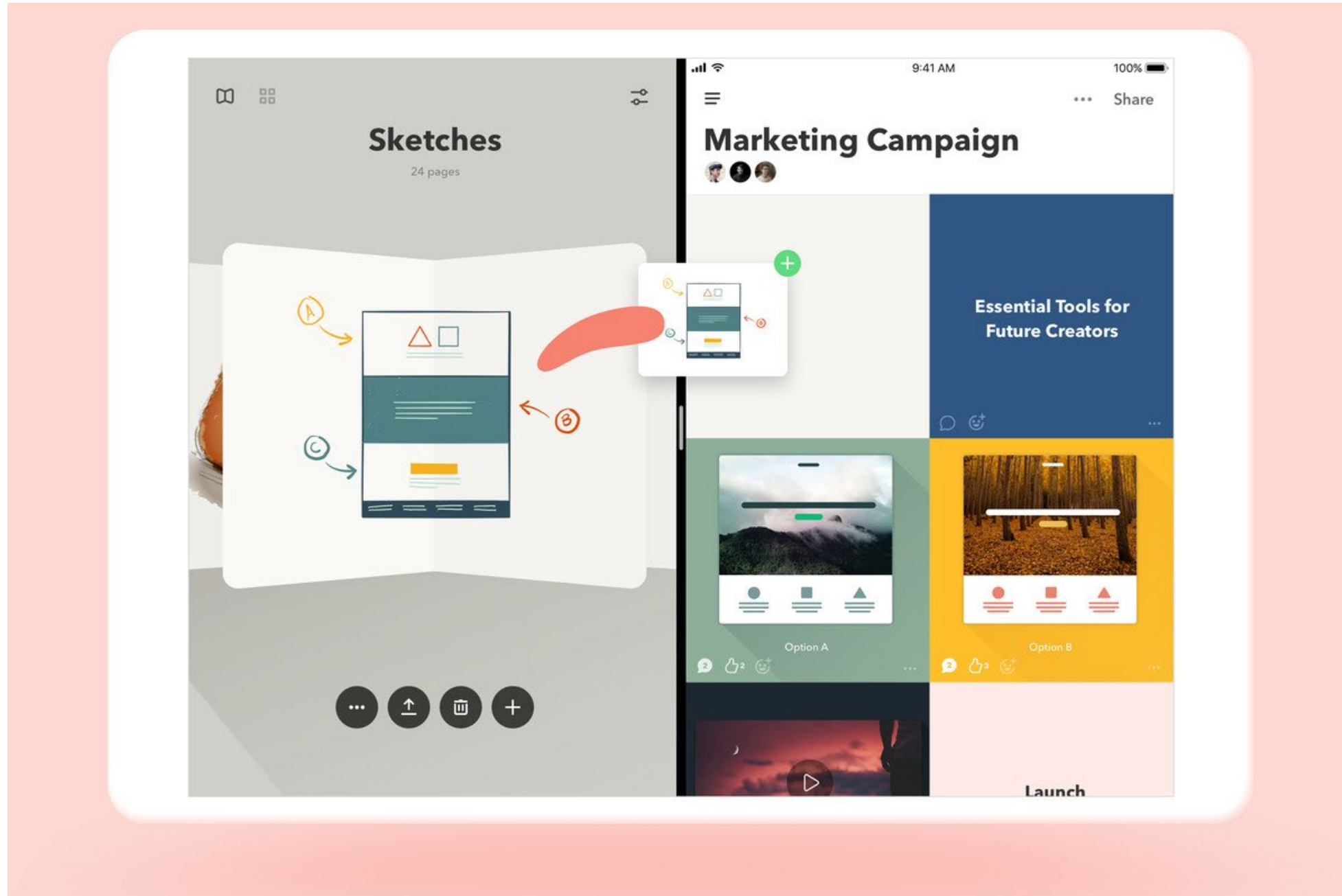
prezi.com/your/

Adam Somlai-Fischer, Peter Halacsy and Peter Arvai



# Paste

# 2017-Today

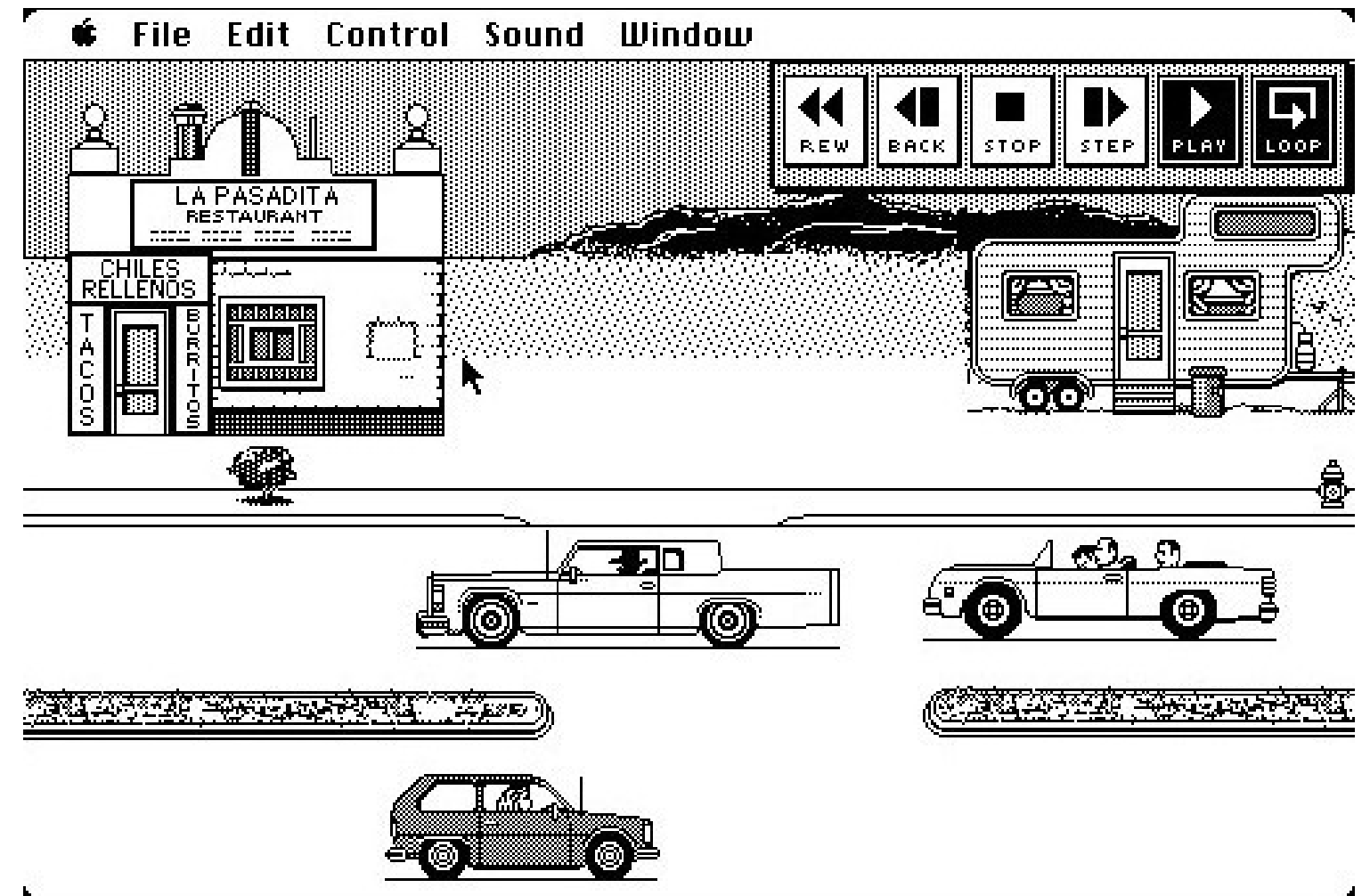
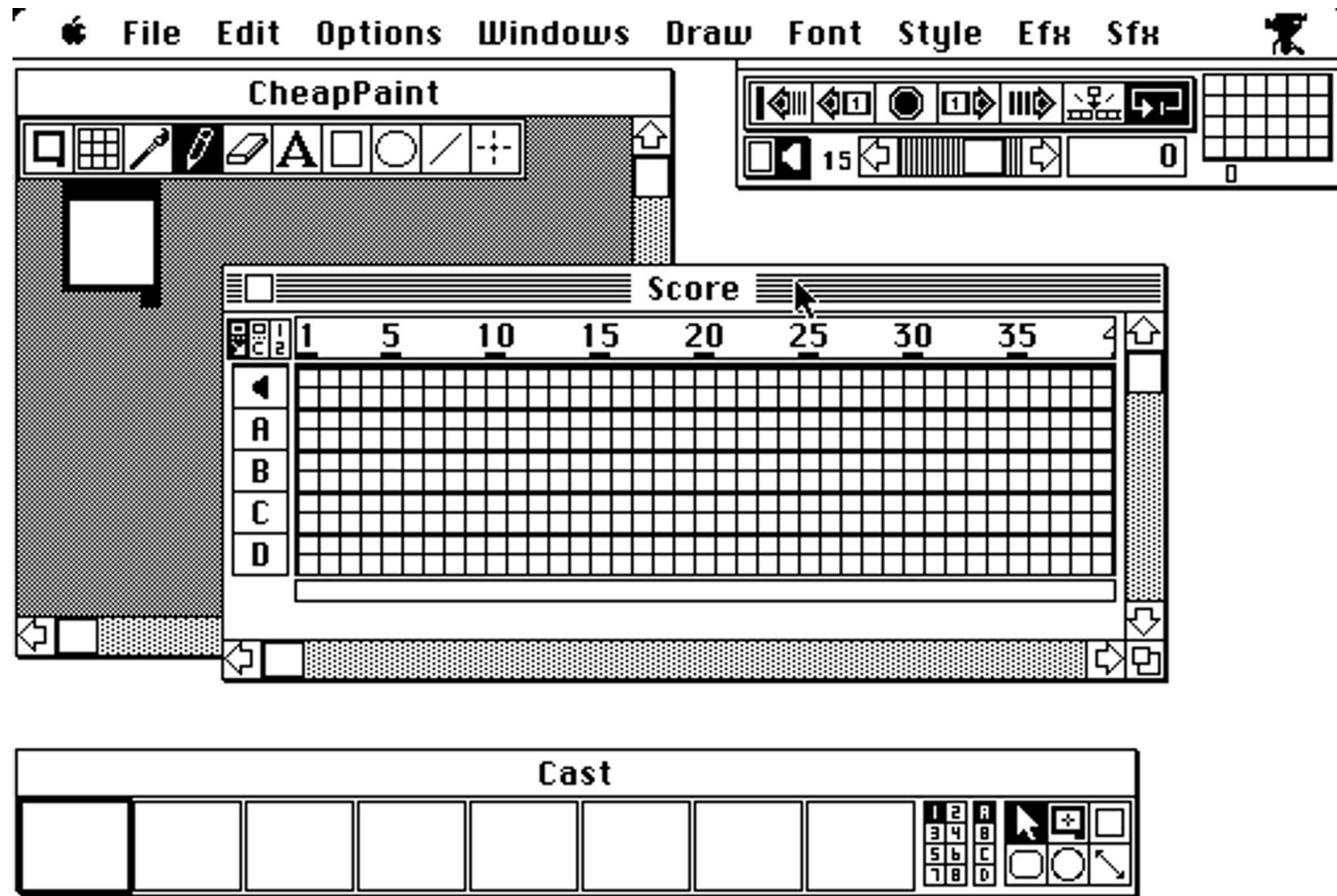


FiftyThree

# Timeline-Based Applications

# VideoWorks

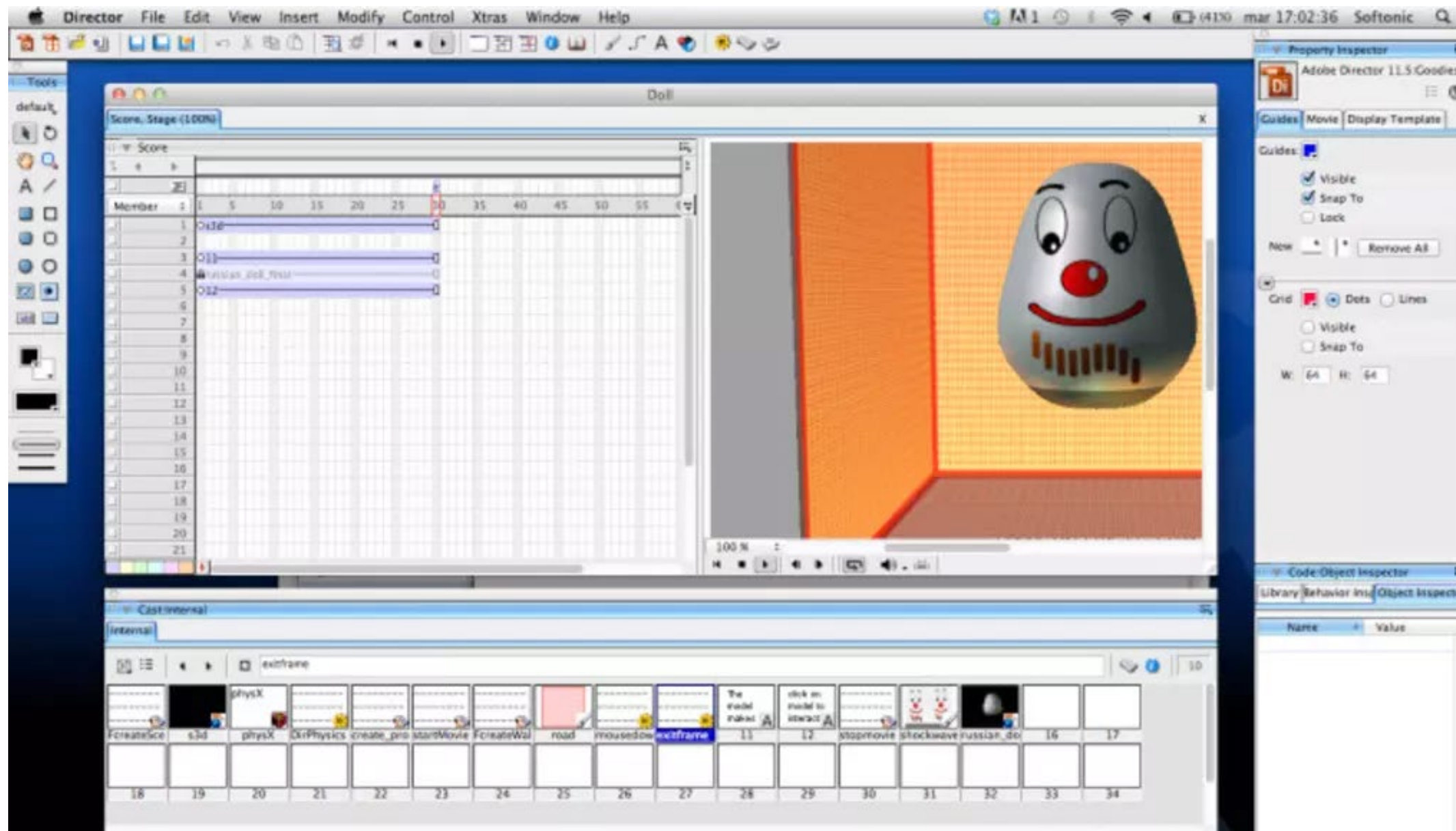
1985-1987



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

# Director

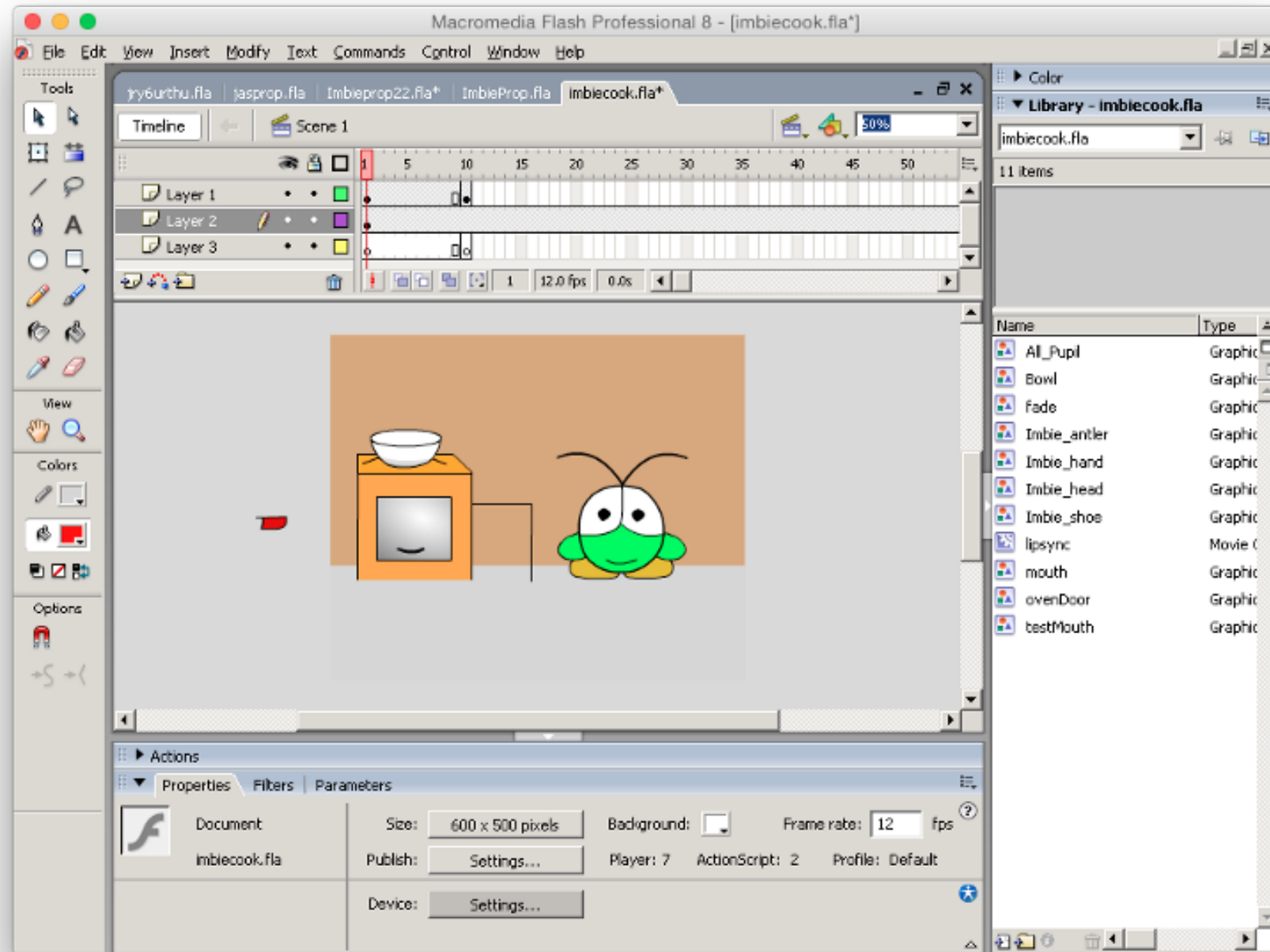
1988-2013



Macromedia (Later Adobe)

# Flash

1996-2016



Adobe

# iMovie

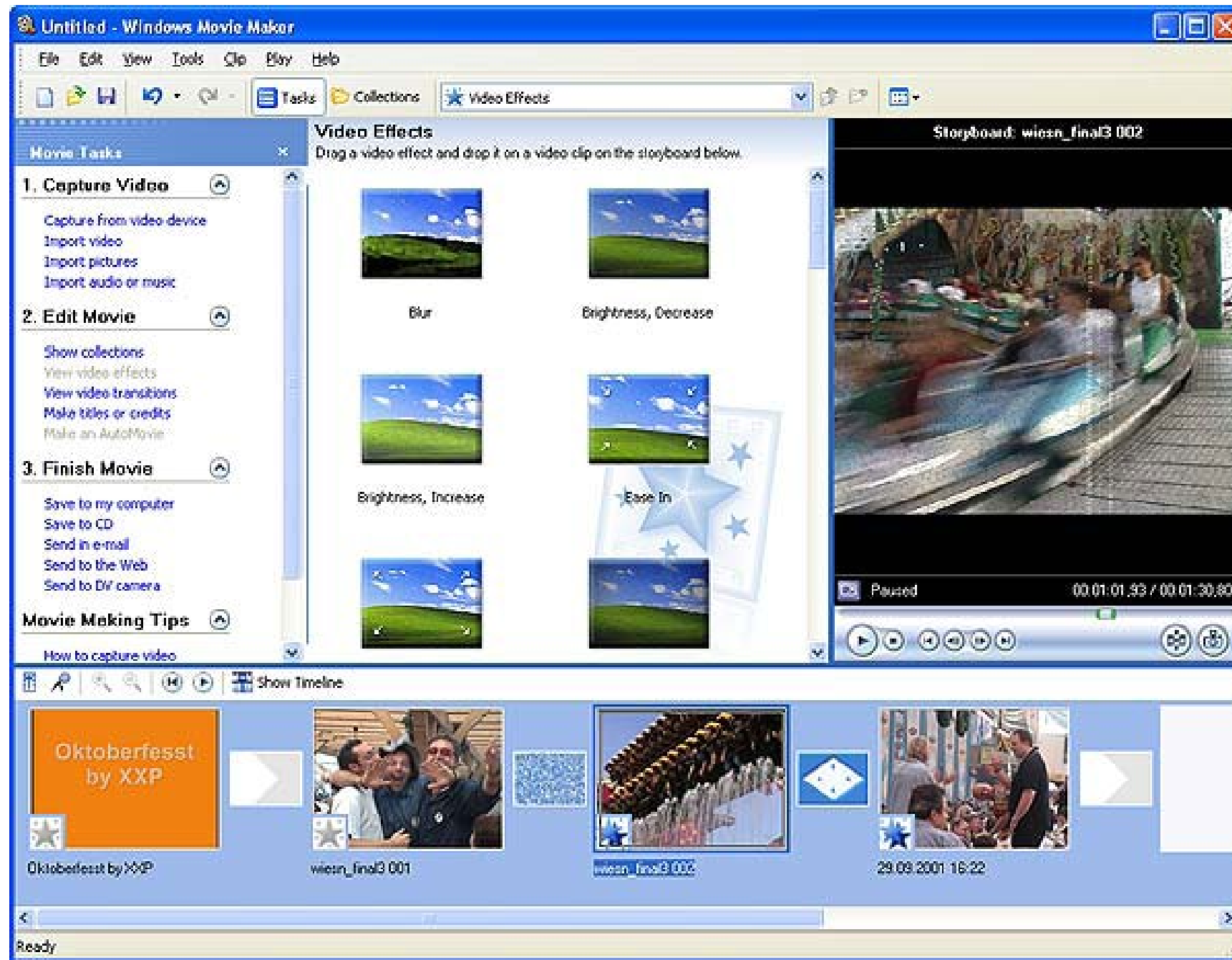
# 1999-Today



Apple

# Windows Movie Maker

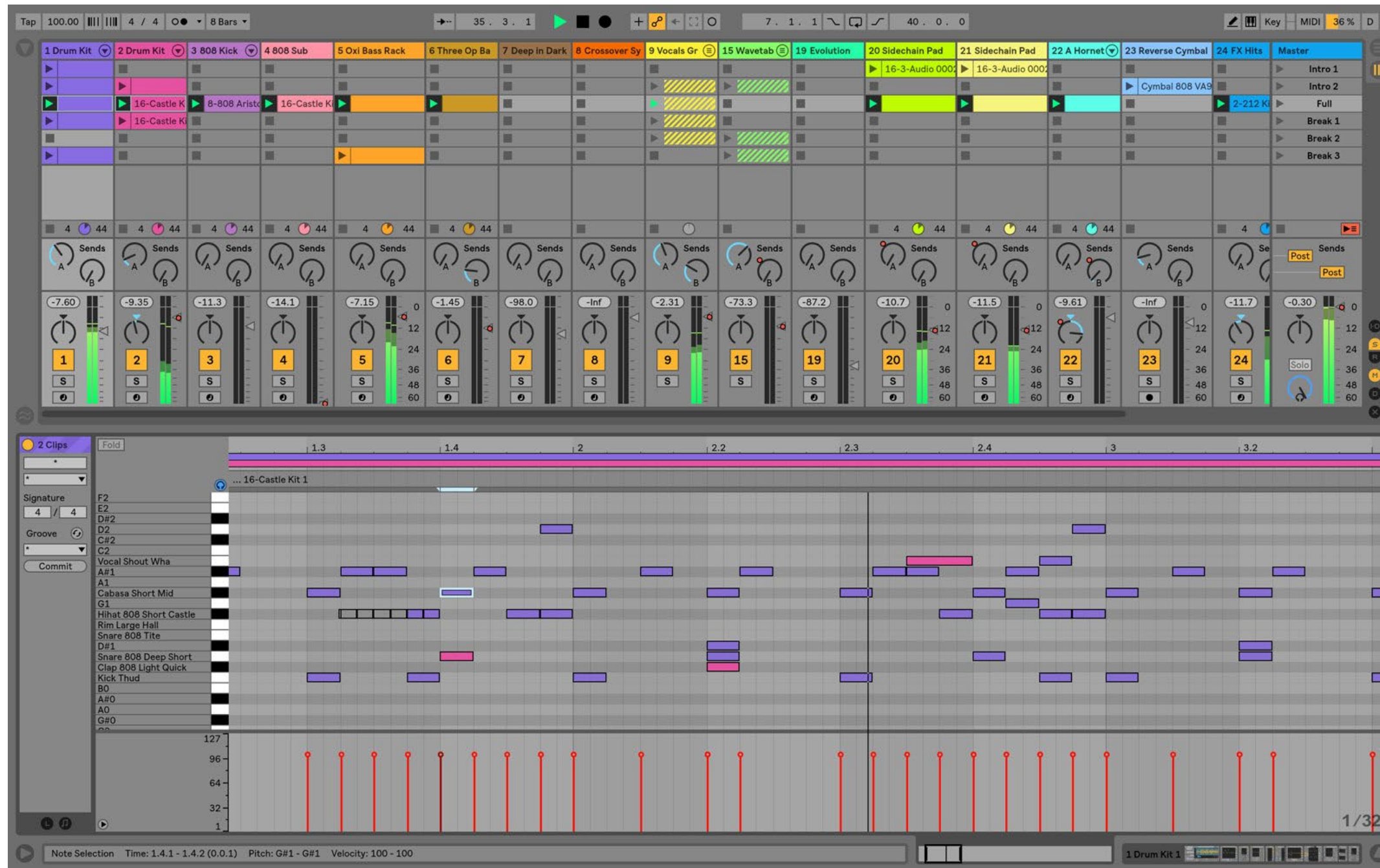
2000-2012



Microsoft

# Ableton Live

# 2001-Today



Ableton



# GarageBand

2004-Today



Apple

# Wick Editor

2017-Today



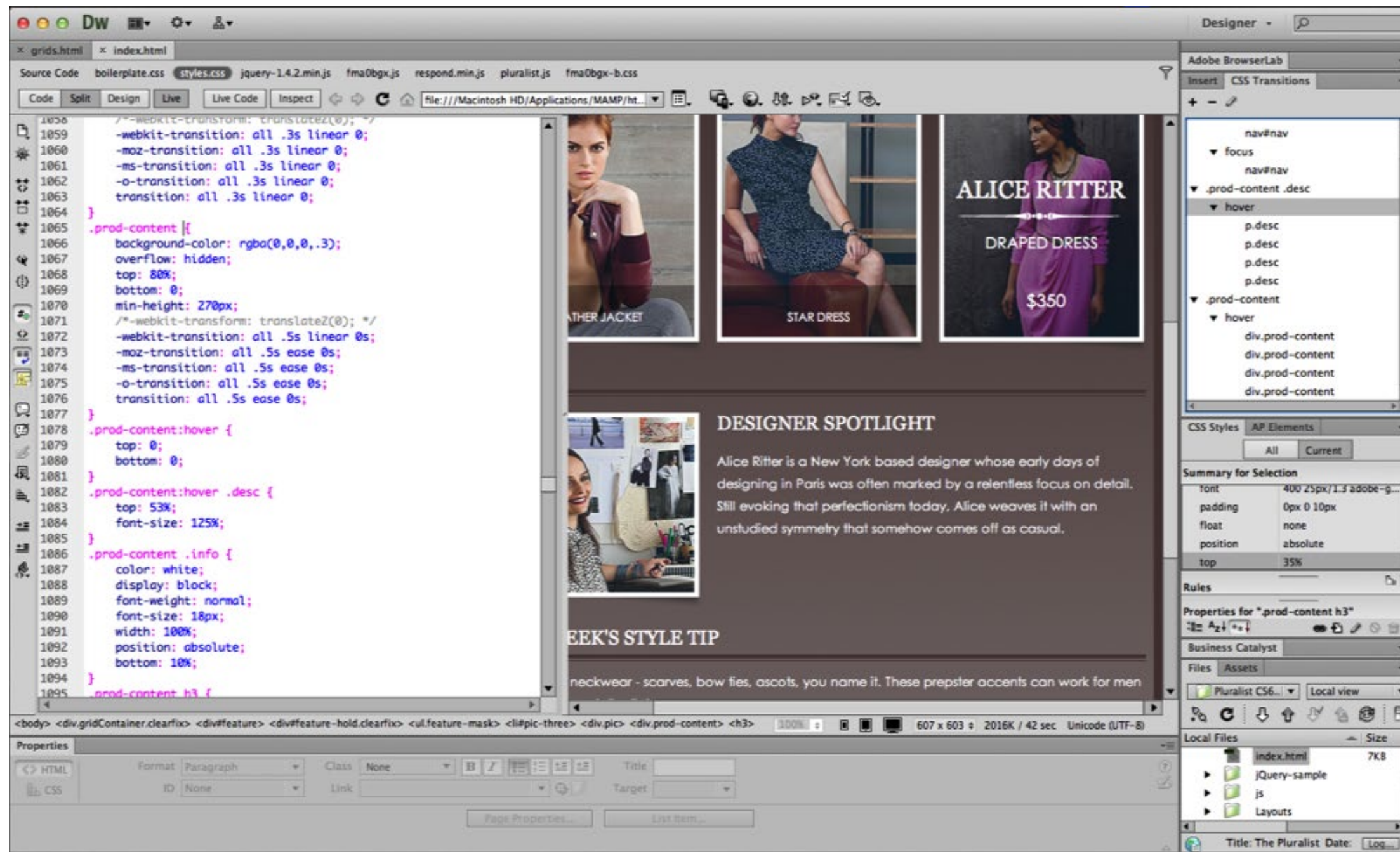
Luca Damasco and Zach Rispoli at the Frank-Ratchye STUDIO for Creative Inquiry at Carnegie Mellon University

# **World Wide Web**

## Design & Development Environments

# Dreamweaver

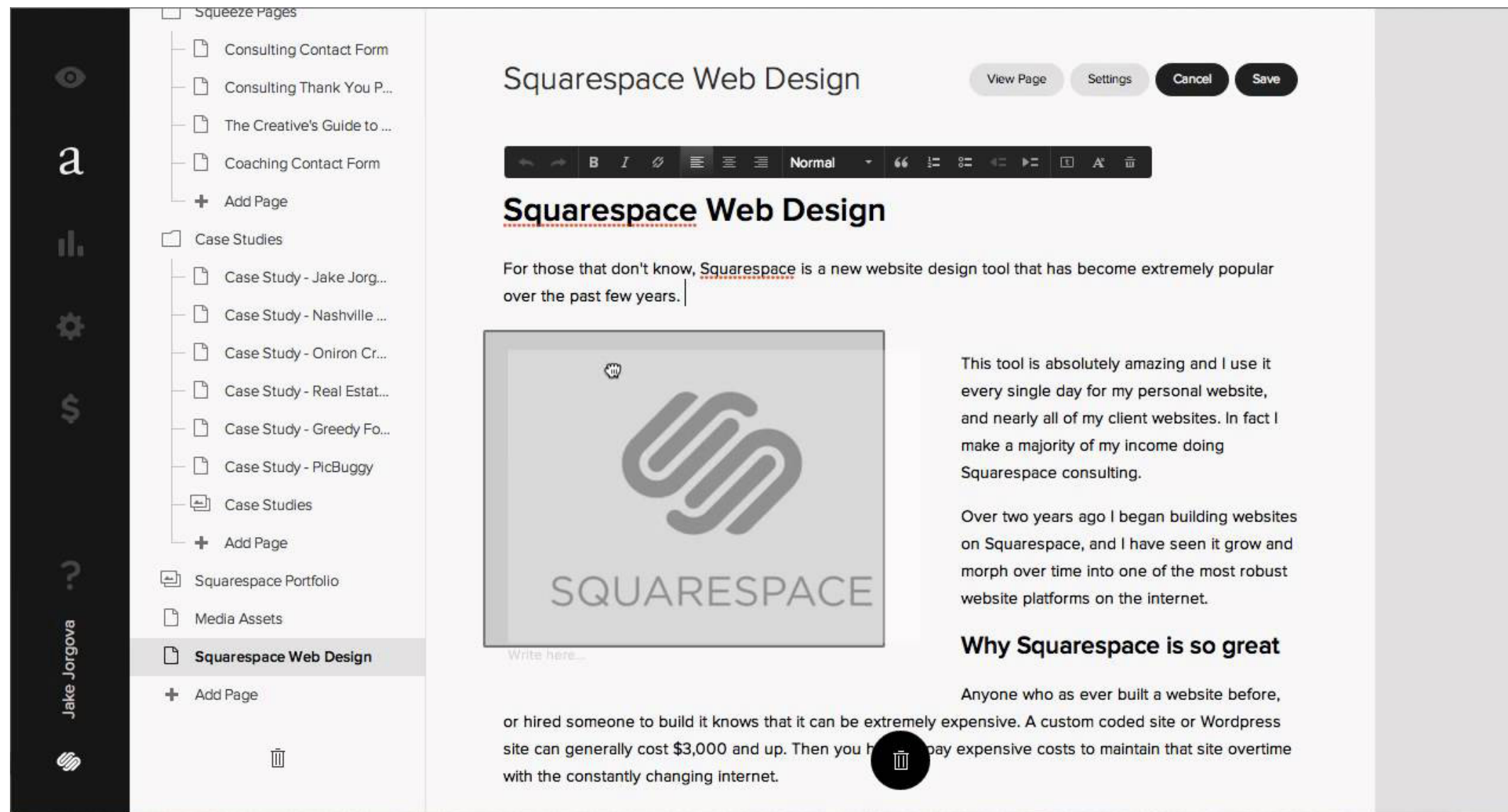
# 1997-Today



Macromedia (Later Adobe)

# Squarespace

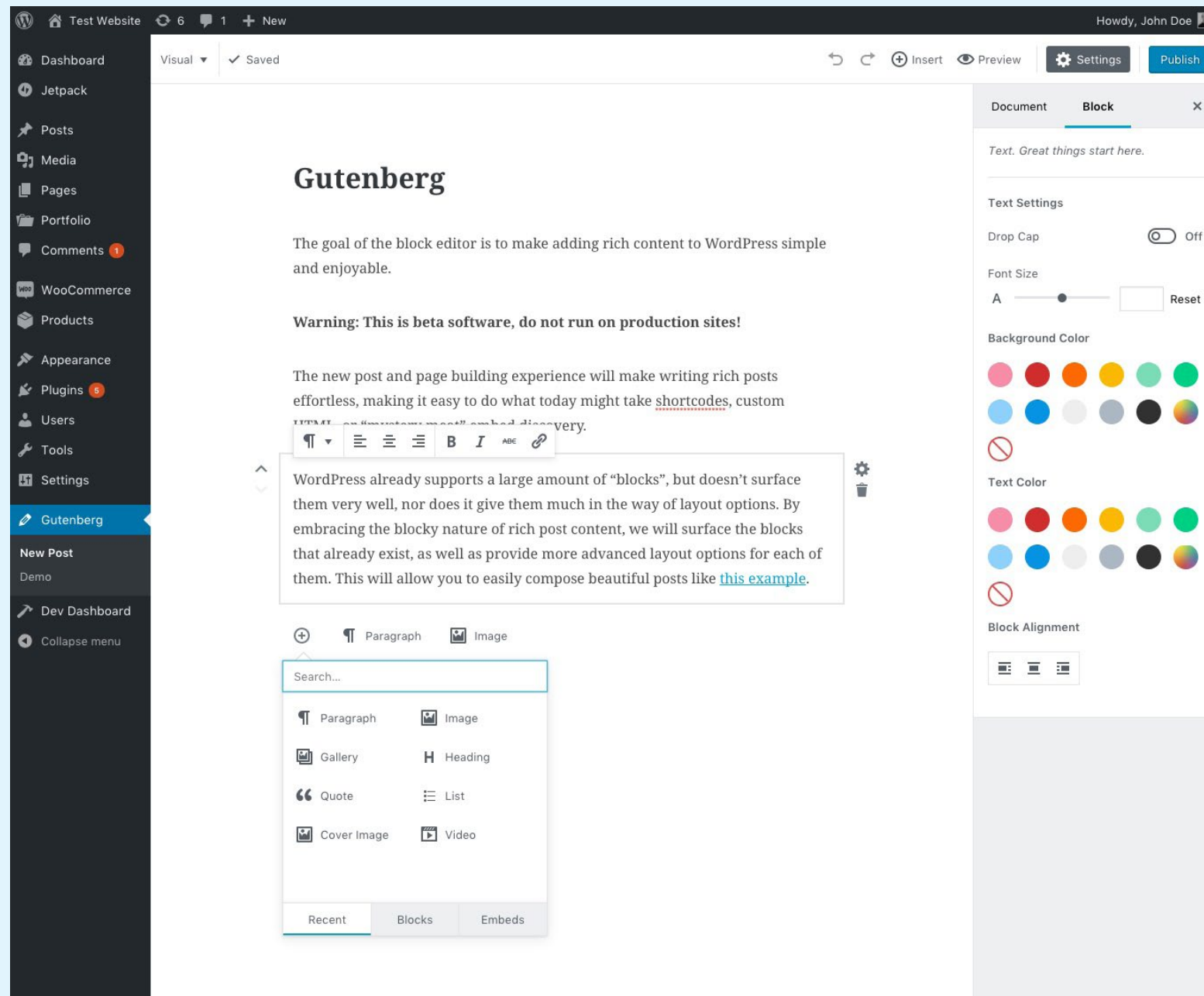
# 2003-Today



Anthony Casalena and Others

# Wordpress

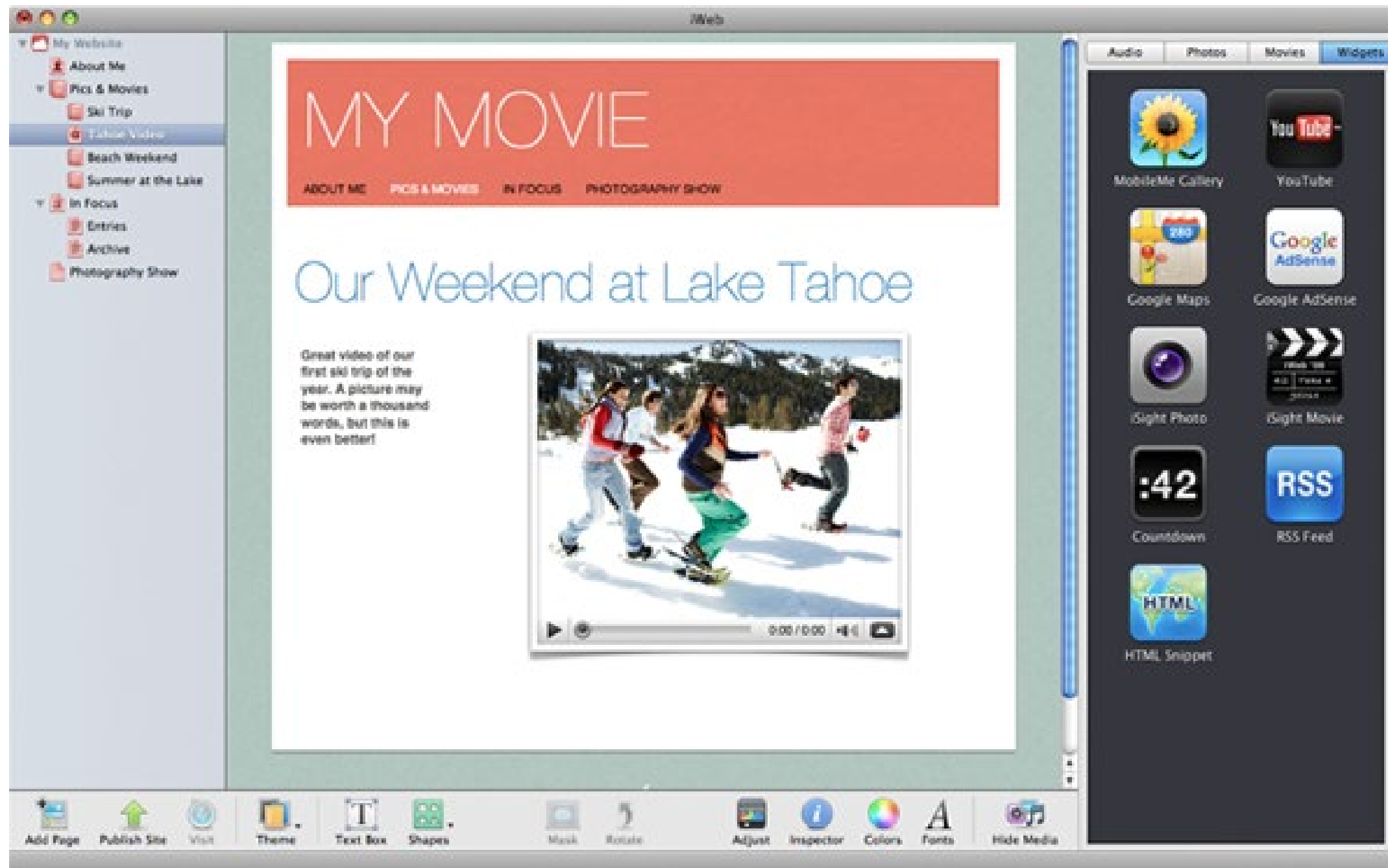
# 2005-Today



Automatic

# iWeb

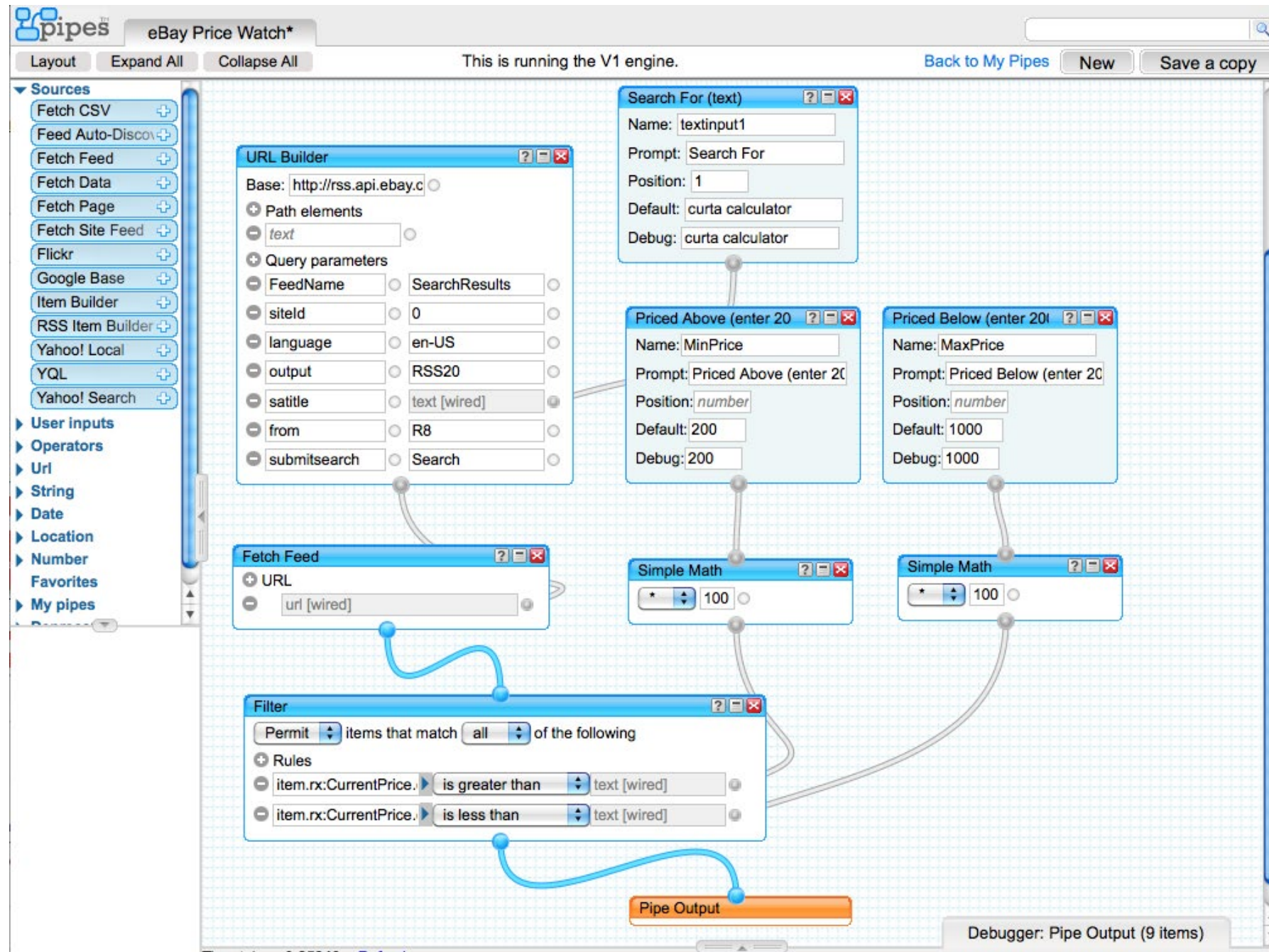
2006-2011



Apple

# Yahoo! Pipes

2007-2015

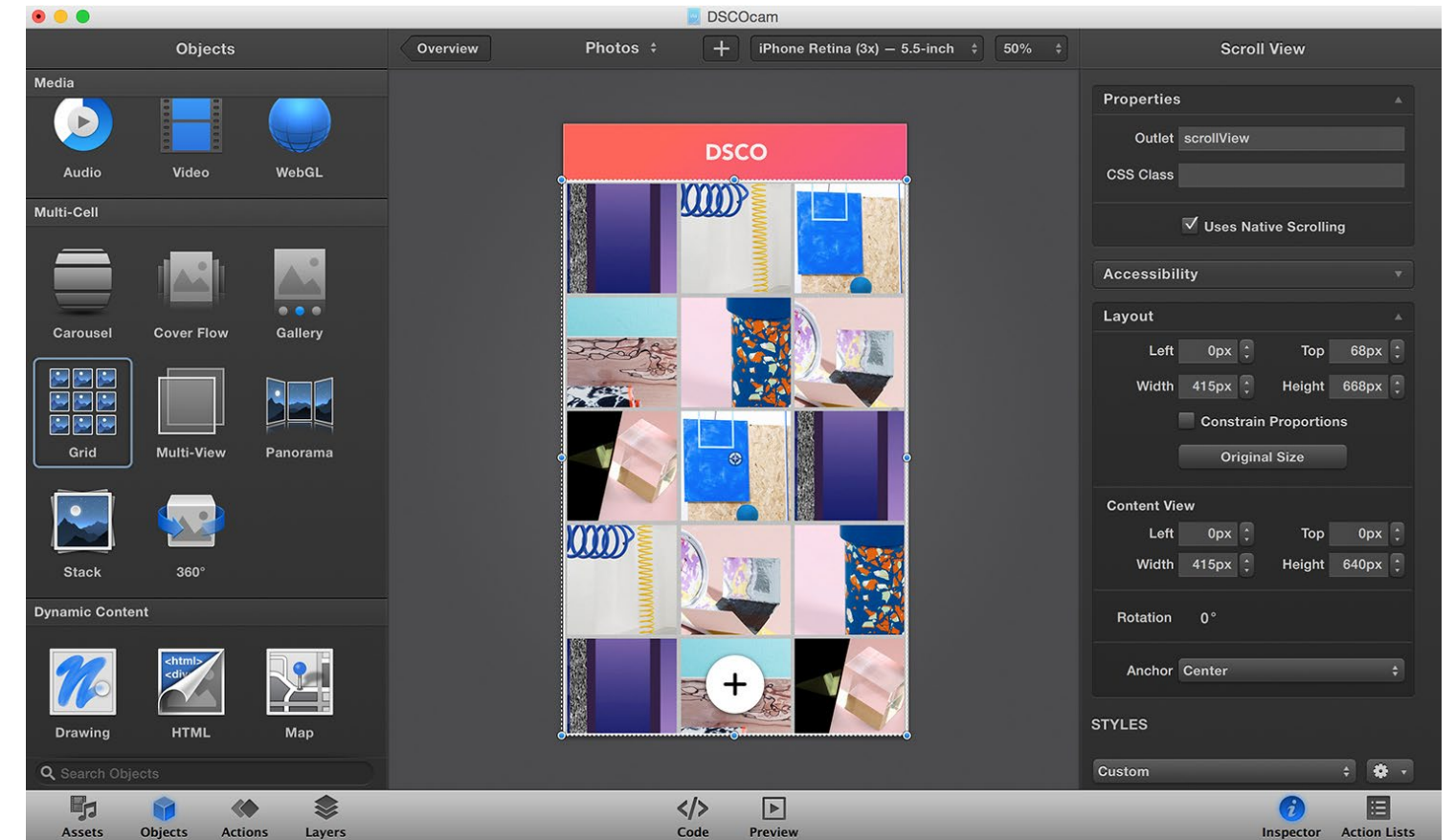
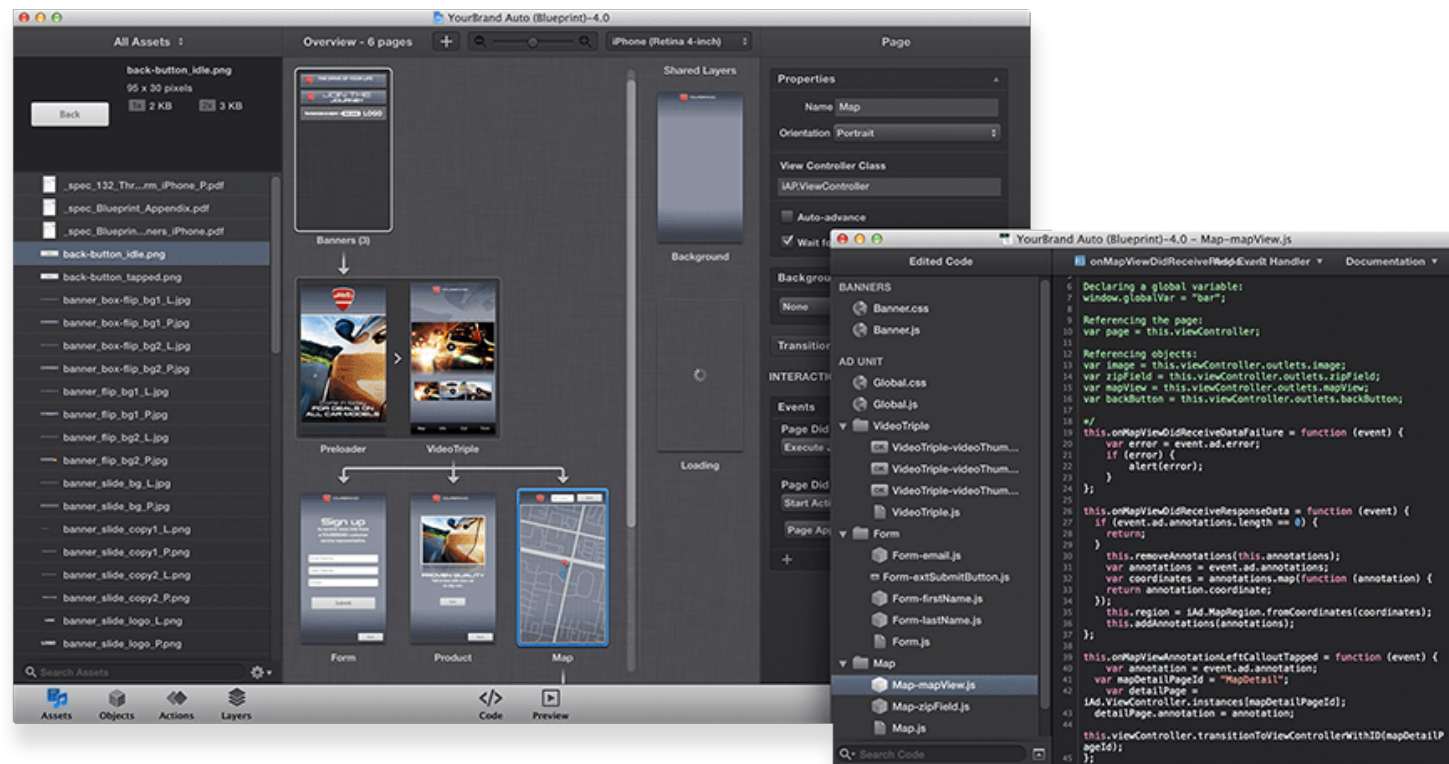


Pasha Sadri, Ed Ho, Jonathan Trevor, Kevin Cheng, Ido Green and Daniel Raffel at Yahoo!



# iAd Producer

2010-2016



Apple

The screenshot shows the JSFiddle web application interface. At the top, there is a navigation bar with buttons for Run, Update, Fork, Reset, Set as base, TidyUp, JSLint, and Share. The user's name 'mplungjan' is visible in the top right corner.

On the left side, there is a sidebar with the following sections:

- Choose Framework:** Includes a dropdown for 'no wrap (head)', 'jQuery 1.7.1', and checkboxes for 'jQuery UI 1.8.16', 'jQuery Lint (June 2011)', and 'Bootstrap 2.0.2 (js only)'.
- Library tag attributes (?):** A text input field.
- Normalized CSS:** A checked checkbox.
- Panels:** A dropdown menu.
- Add Resources:** A dropdown menu.
- Info:** A dropdown menu.
- Testing Ajax requests:** A dropdown menu.
- Examples:** A dropdown menu.
- Legal stuff and credits:** A dropdown menu.
- Read our documentation** and **Keyboard shortcuts** links.

The main content area is divided into three panels:

- HTML:** Contains the following code:

```
<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>
</ul>
```
- CSS:** Contains the following code:

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

```
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();
});
```

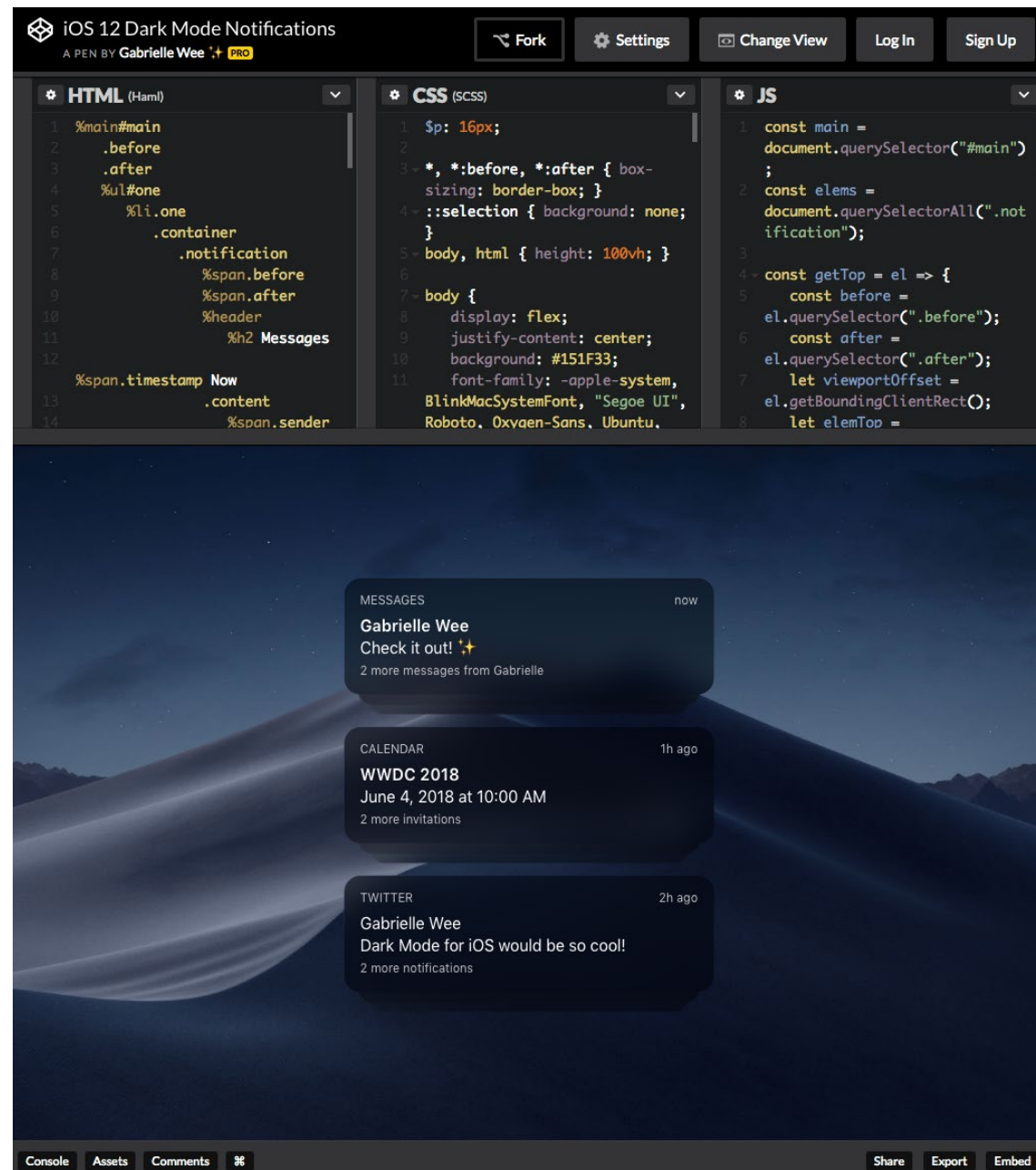
On the right side, there is a **Result** panel showing the output of the JavaScript code:

- 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

Oskar Krawczyk, Piotr Zalewa

# CodePen

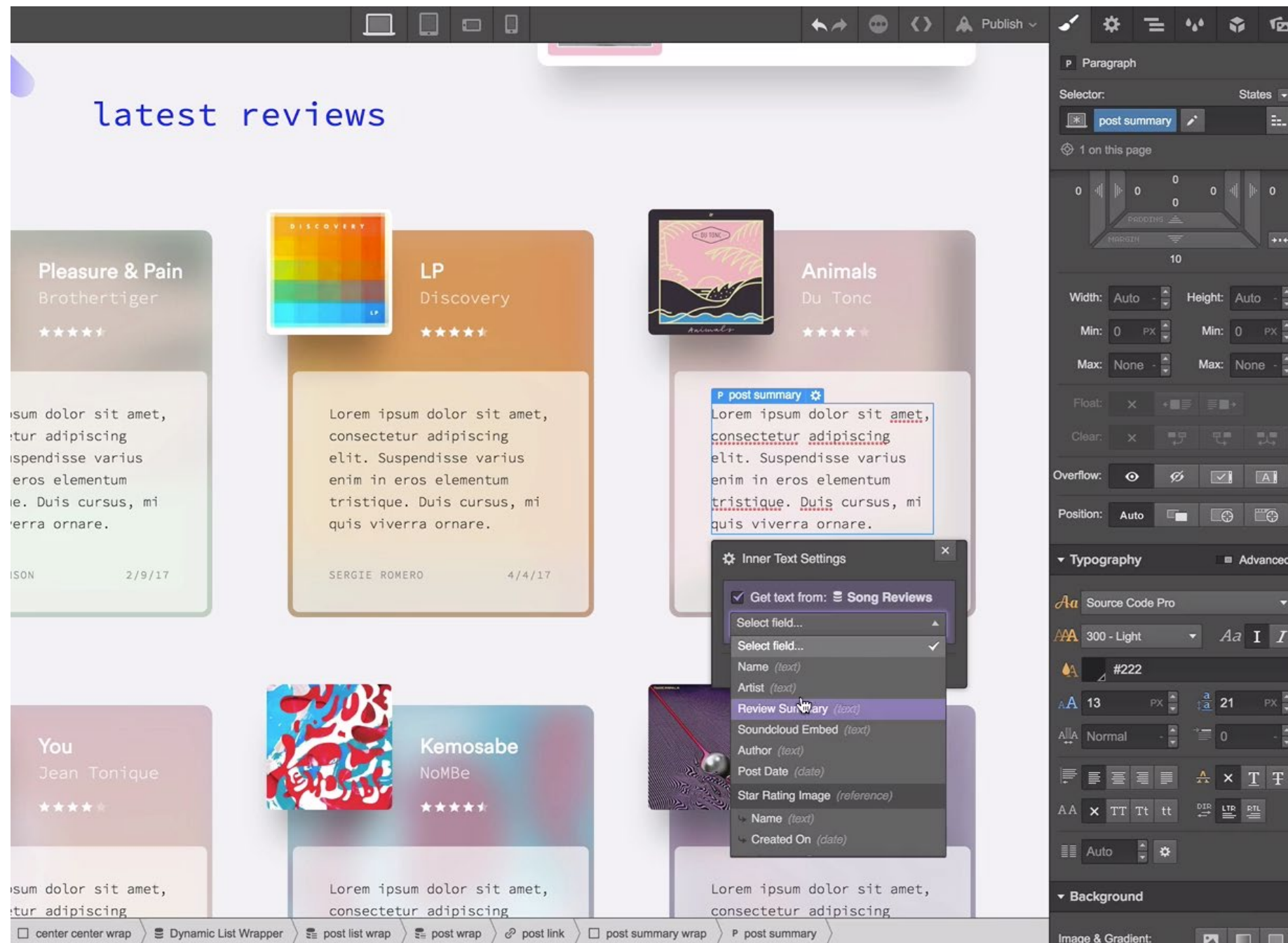
# 2012-Today



Tim Sabat, Alex Vazquez, Chris Coyier

# Webflow

# 2013-Today

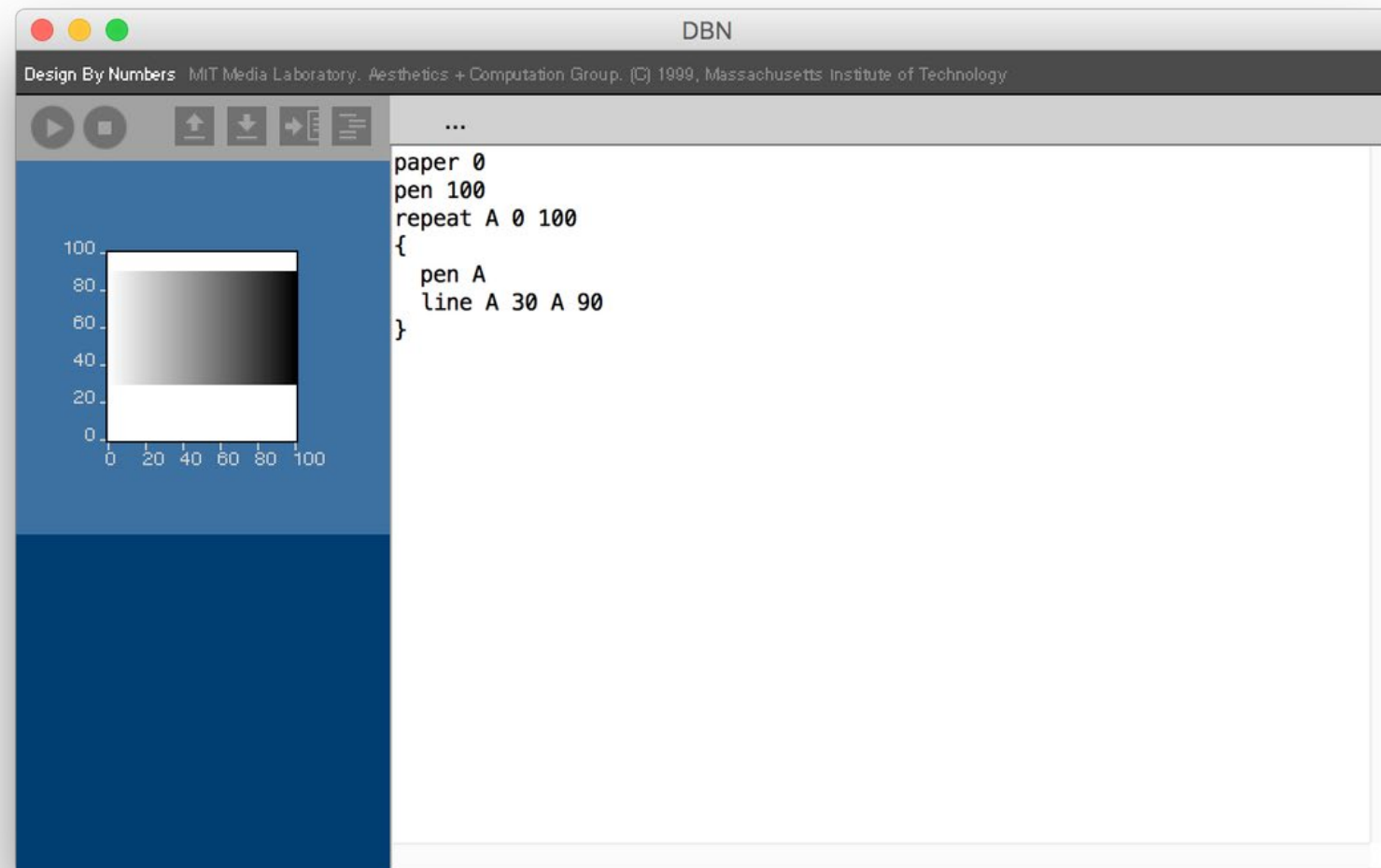


Vlad Magdalin and Sergie Magdalin

# 'Creative Coding' Toolkits

# Design by Numbers

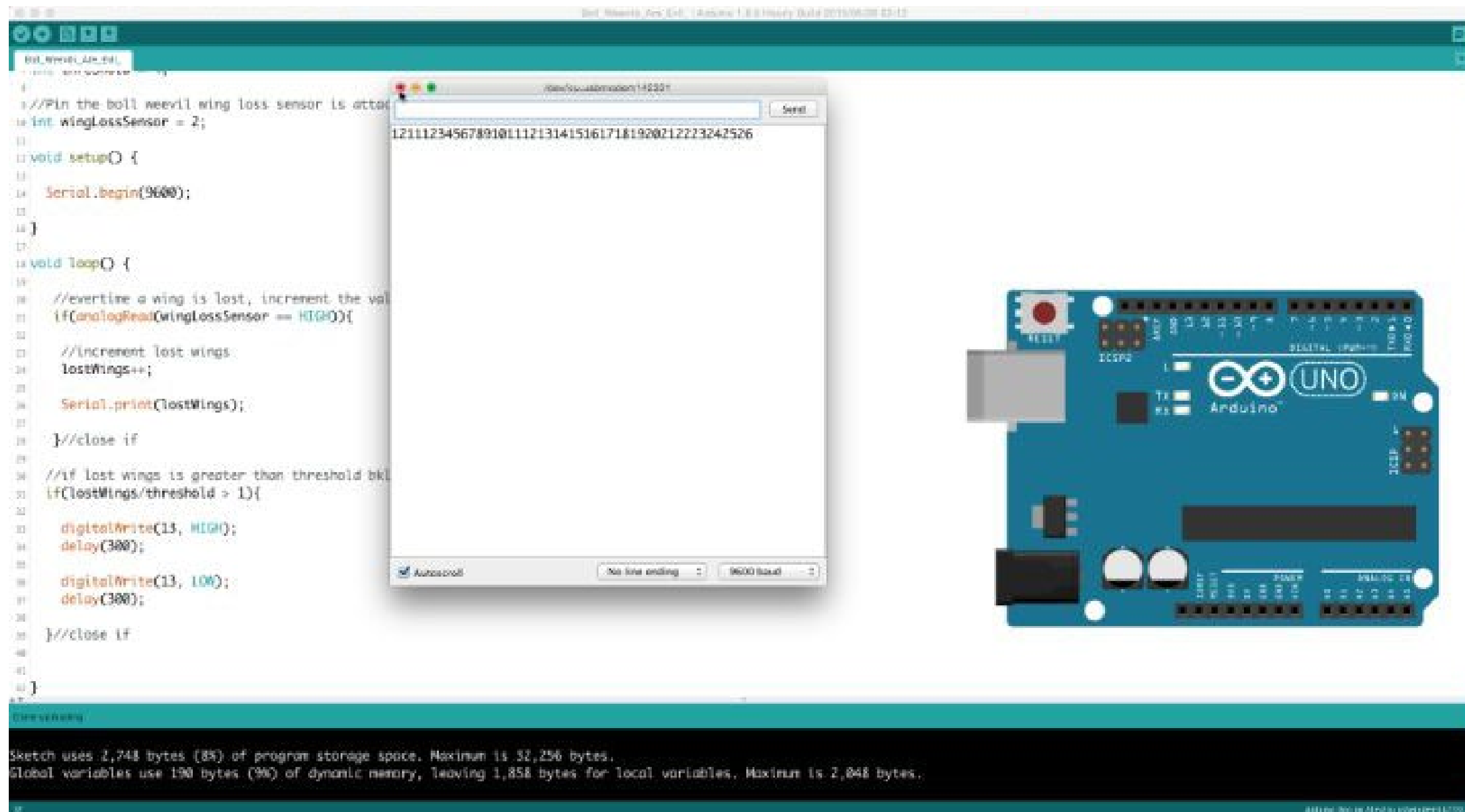
1999-2003



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

# Arduino

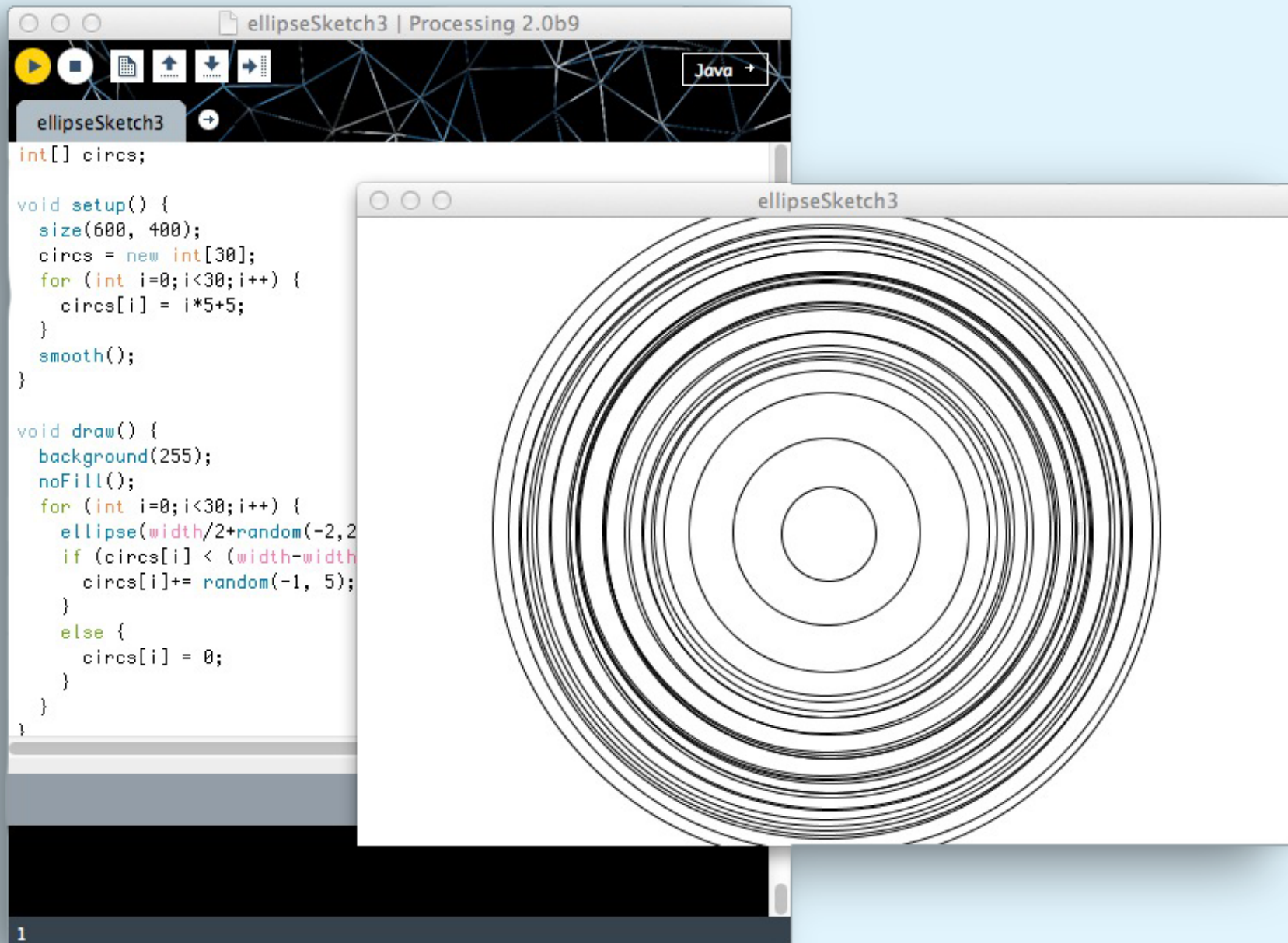
# 2003-Today



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

# Processing

2001-Today

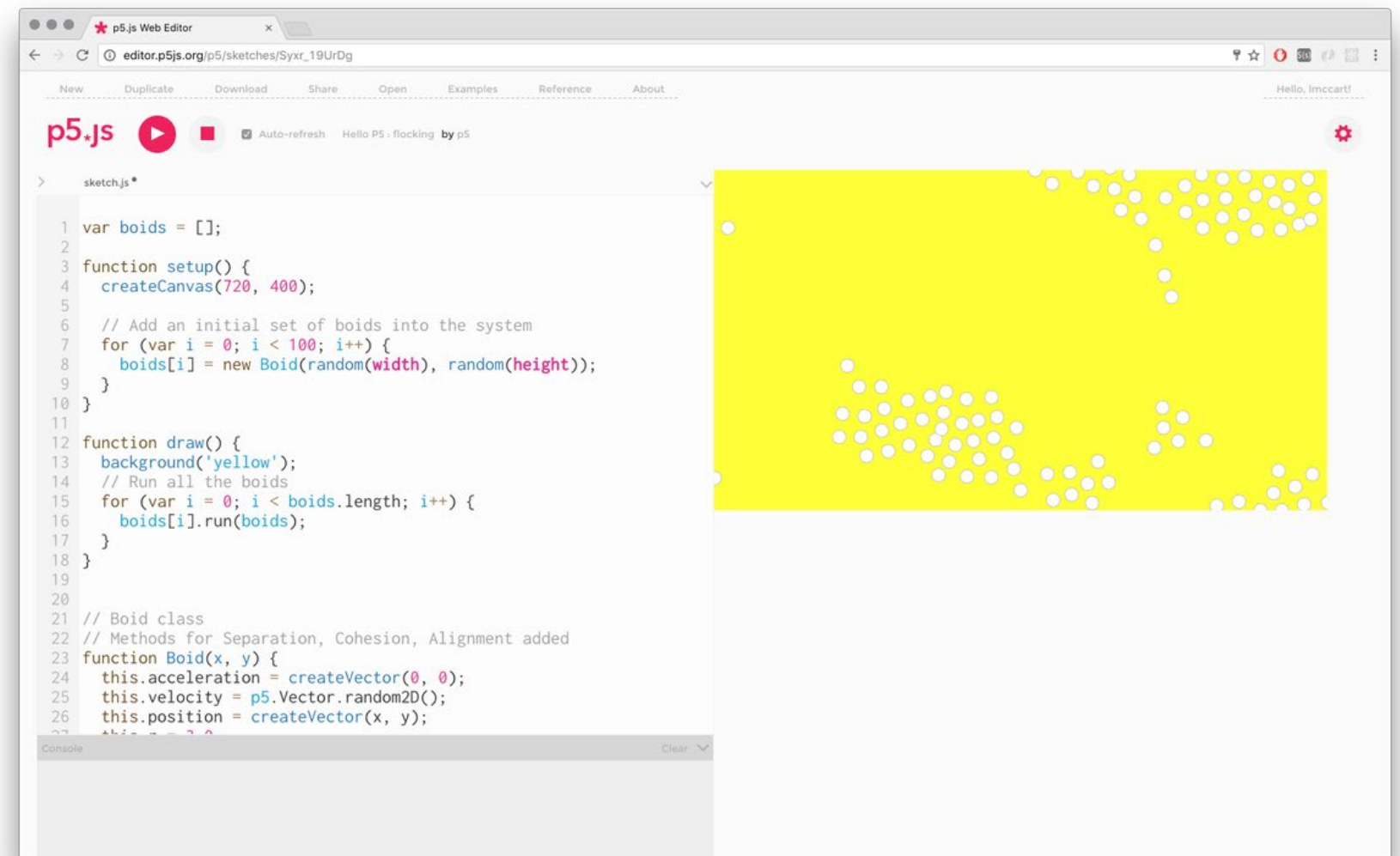
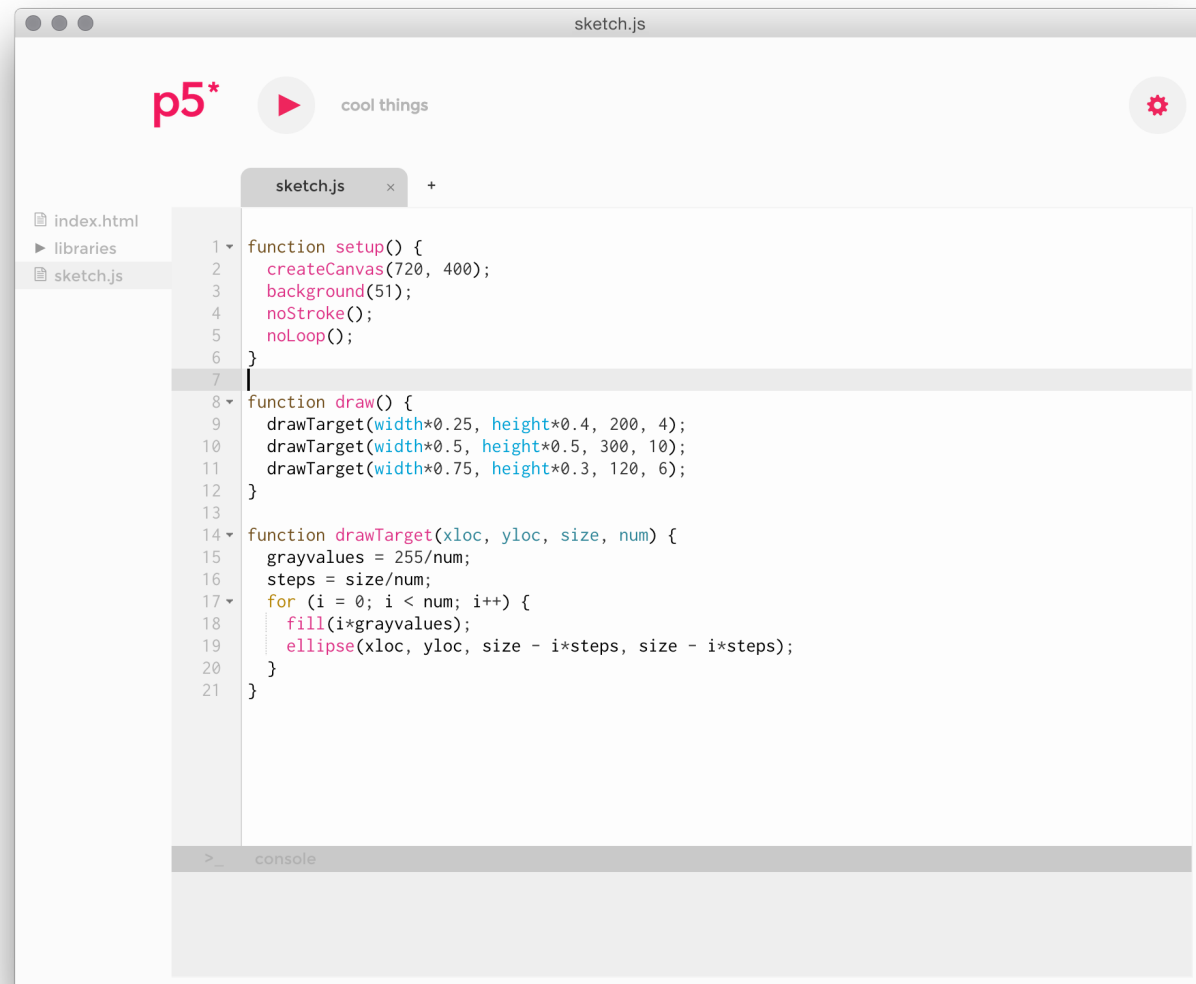


Casey Reas and Ben Fry at the MIT Media Lab (Now Processing Foundation)



# p5.js

# 2014-Today



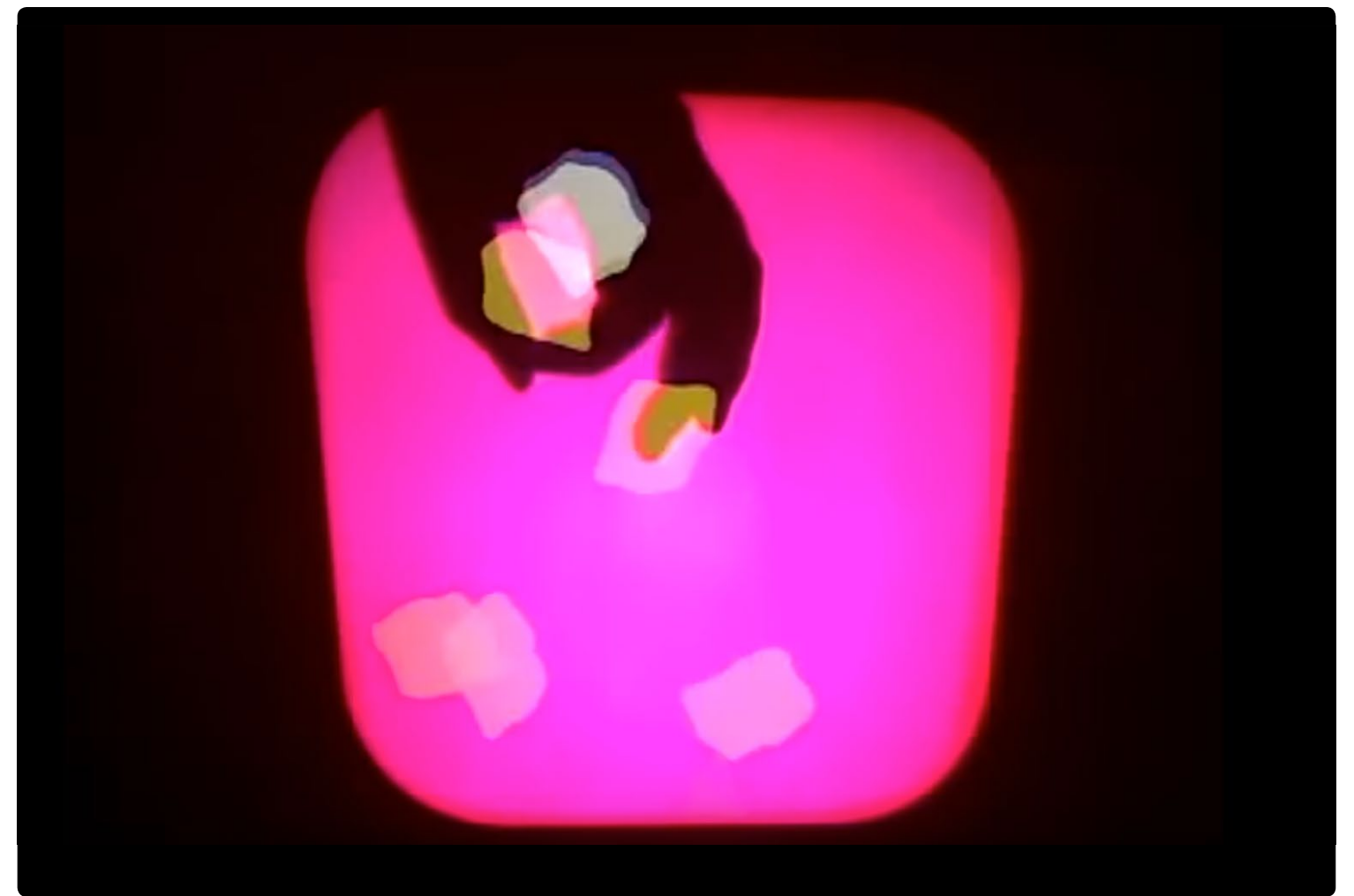
Lauren McCarthy at MIT (Now at UCLA, Dept. of Design Media Arts)

# Digital Drawing

## Painterly Approaches + Computational Methods

# The Manual Input Workstation

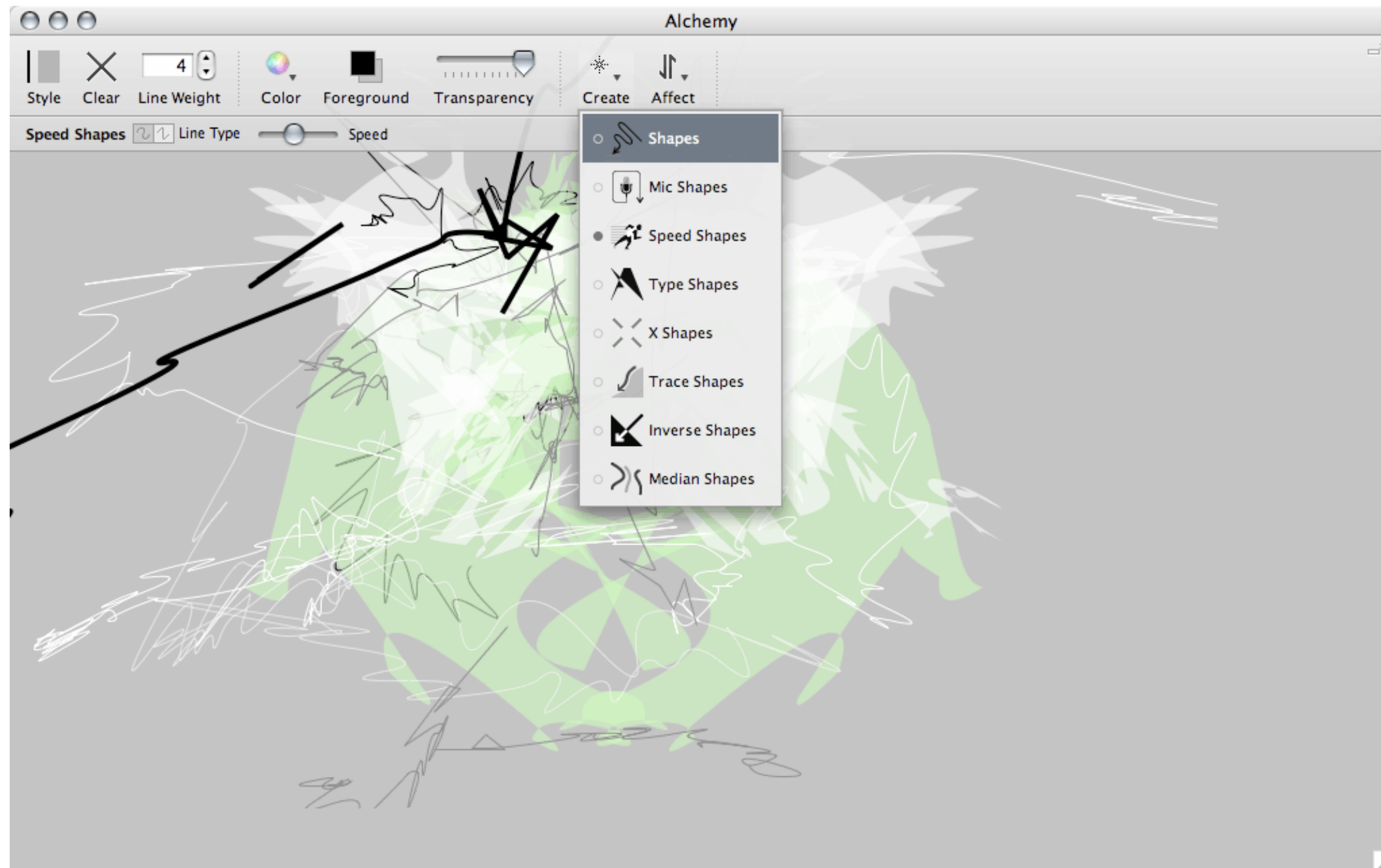
2004



Golan Levin and Zachary Lieberman

# Alchemy

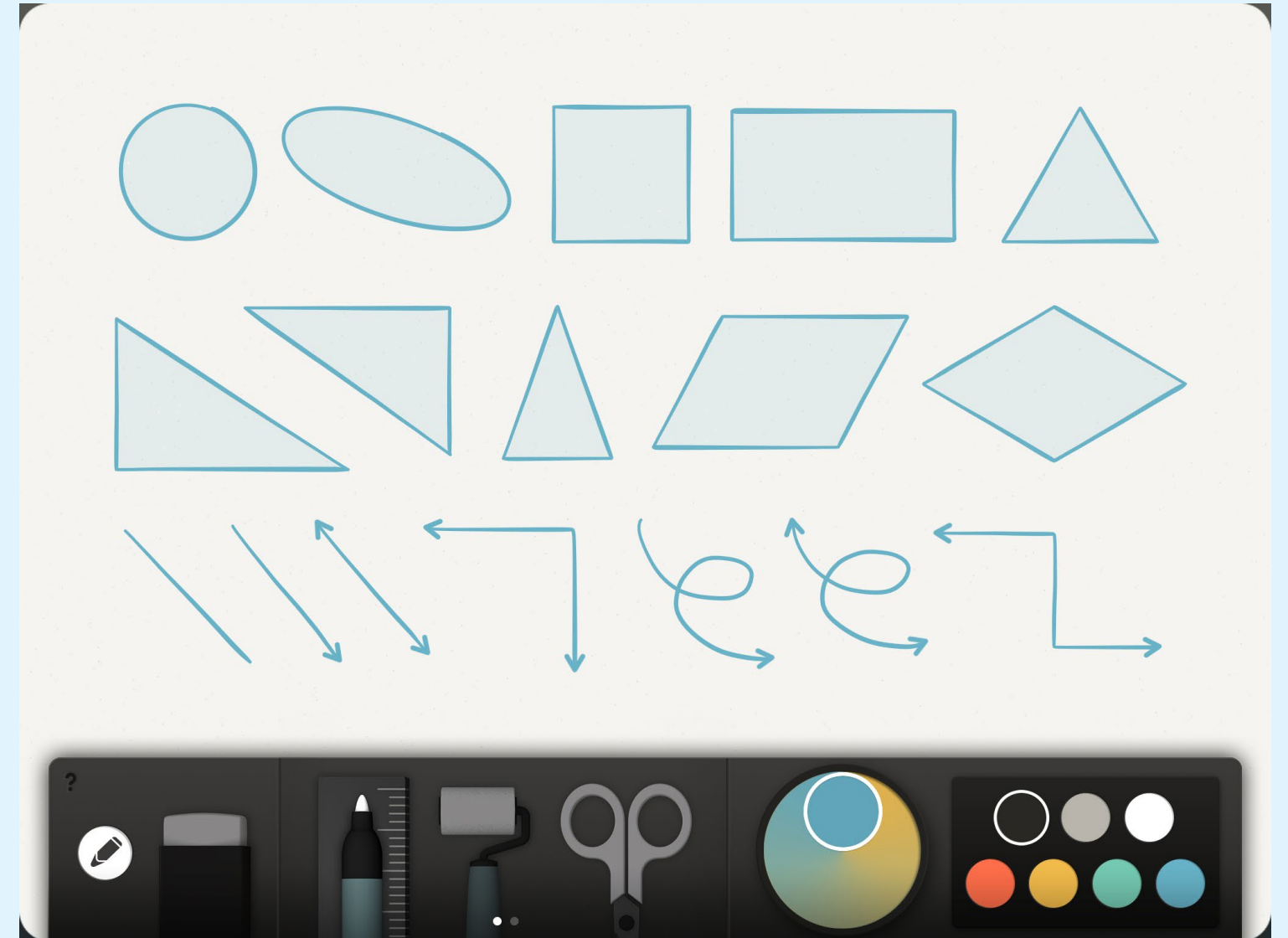
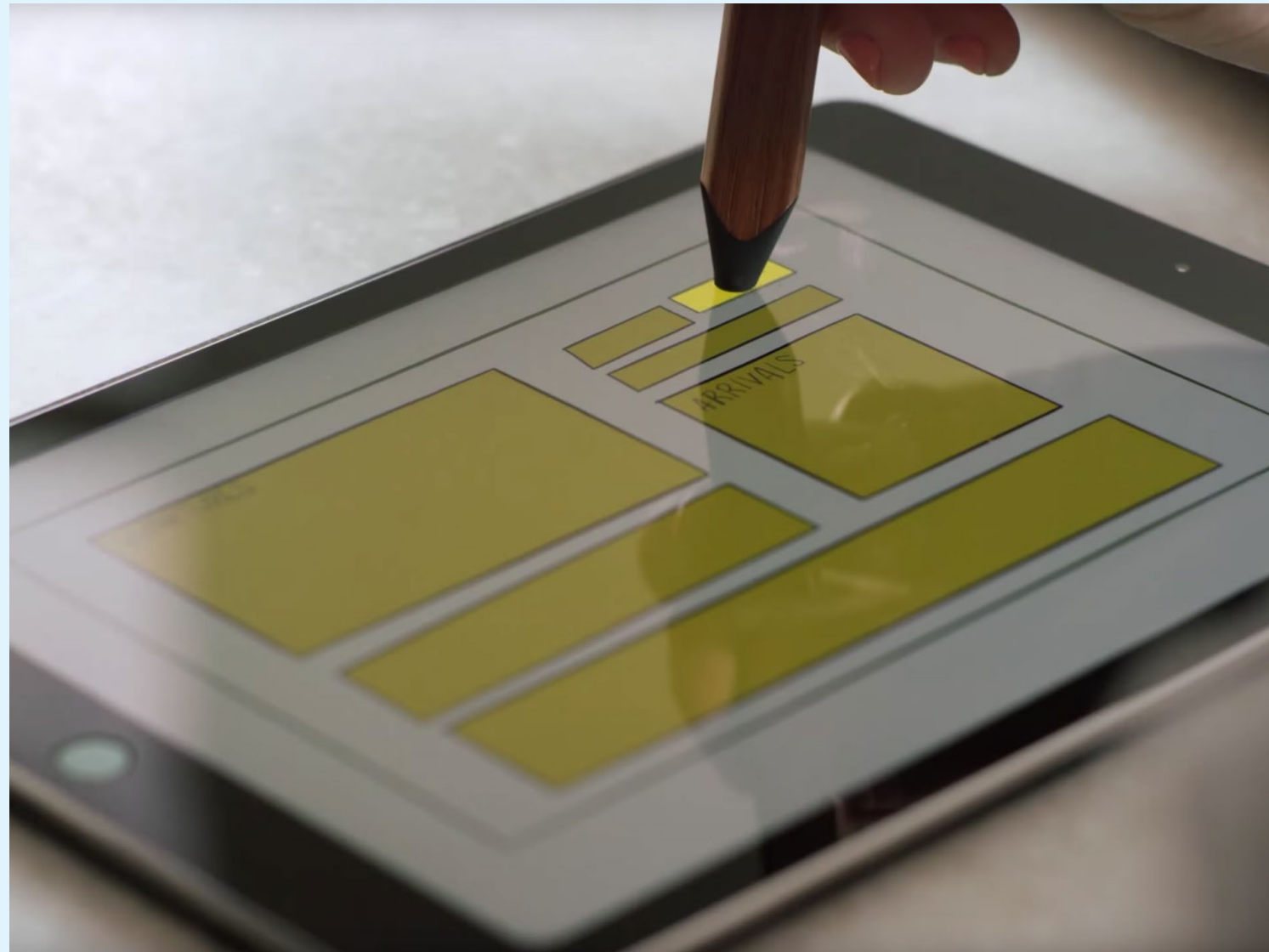
2008-



Karl D.D. Willis and Jacob Hina

# Paper

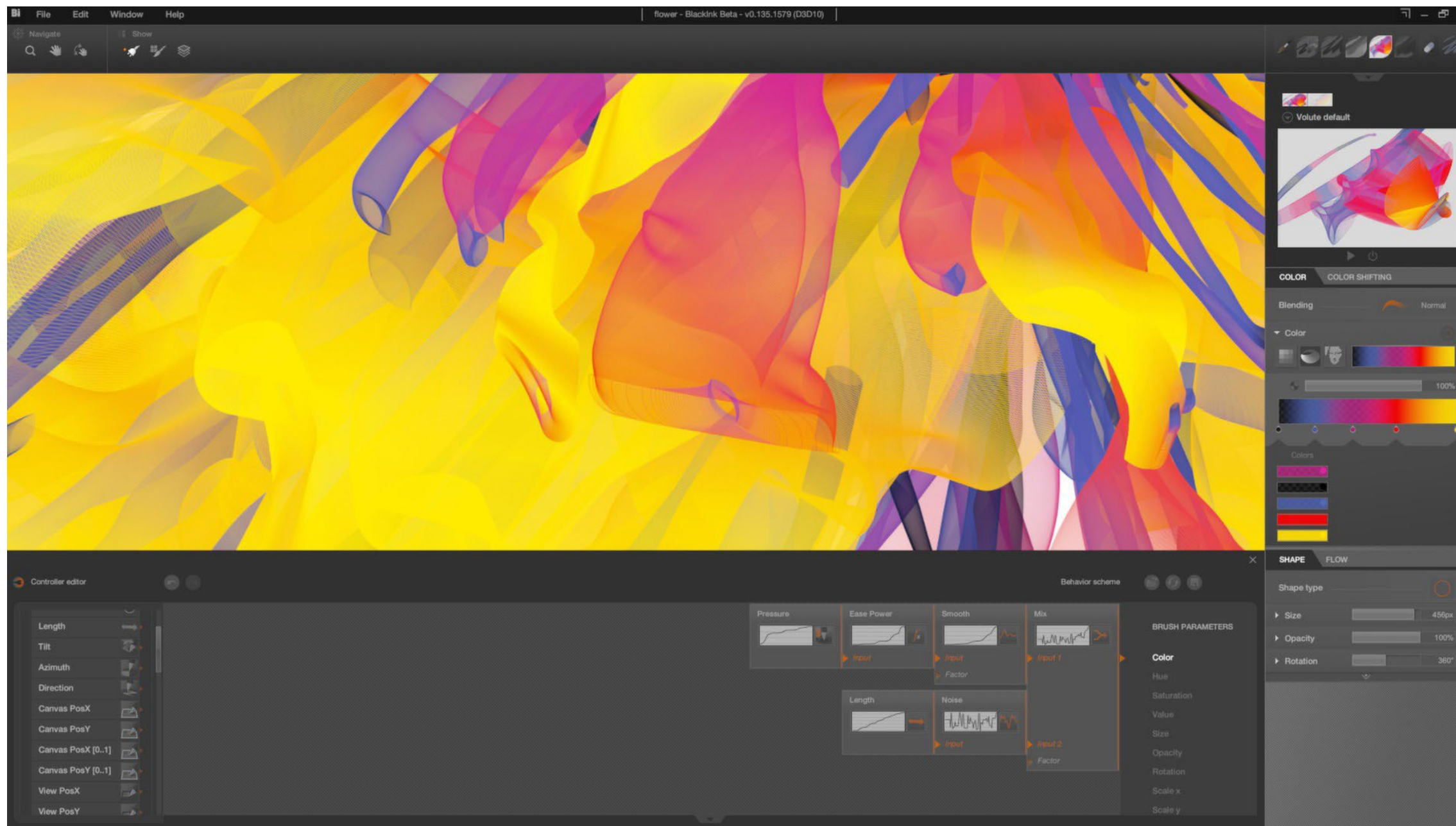
# 2011-Today



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

# BlackInk

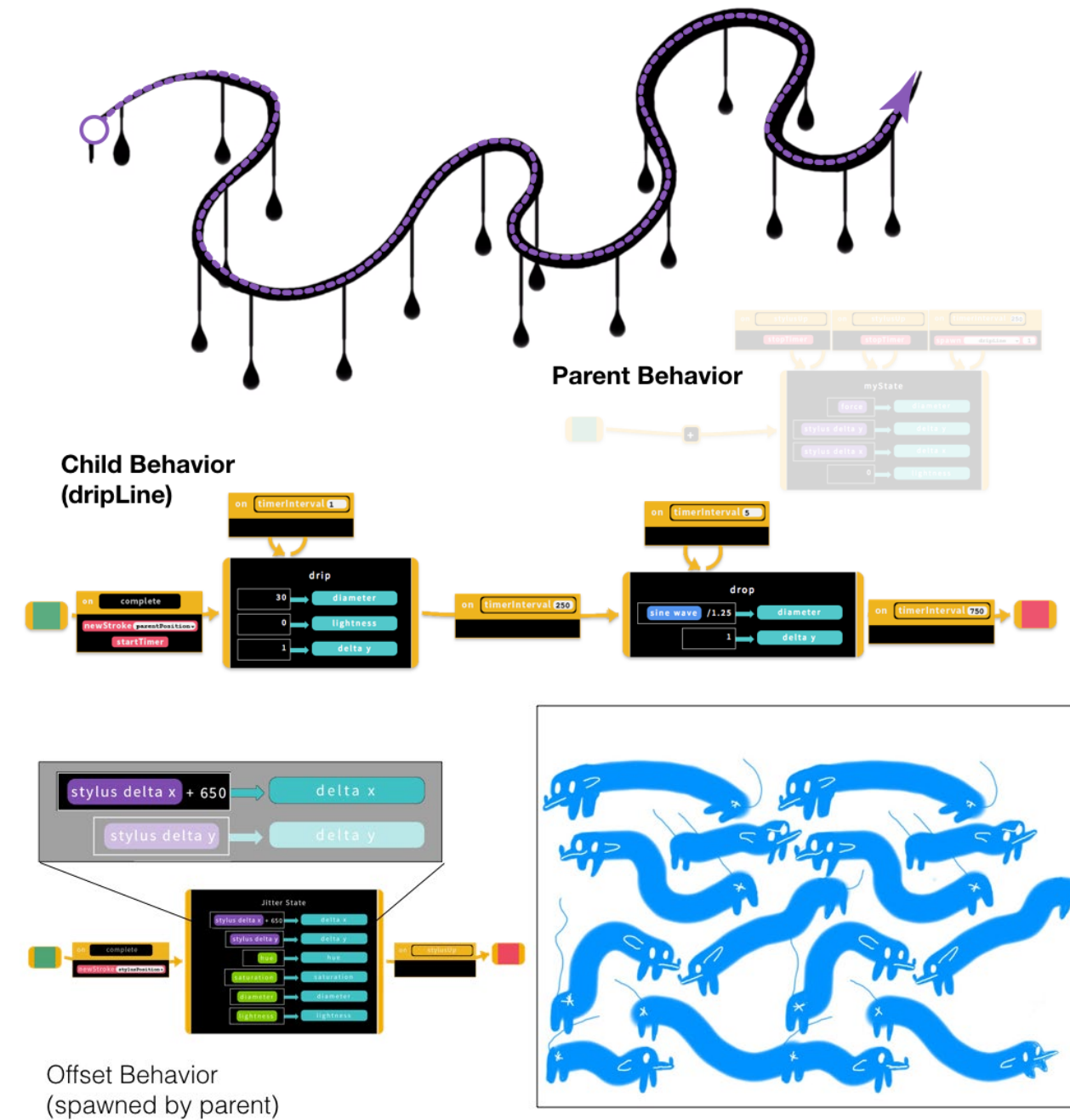
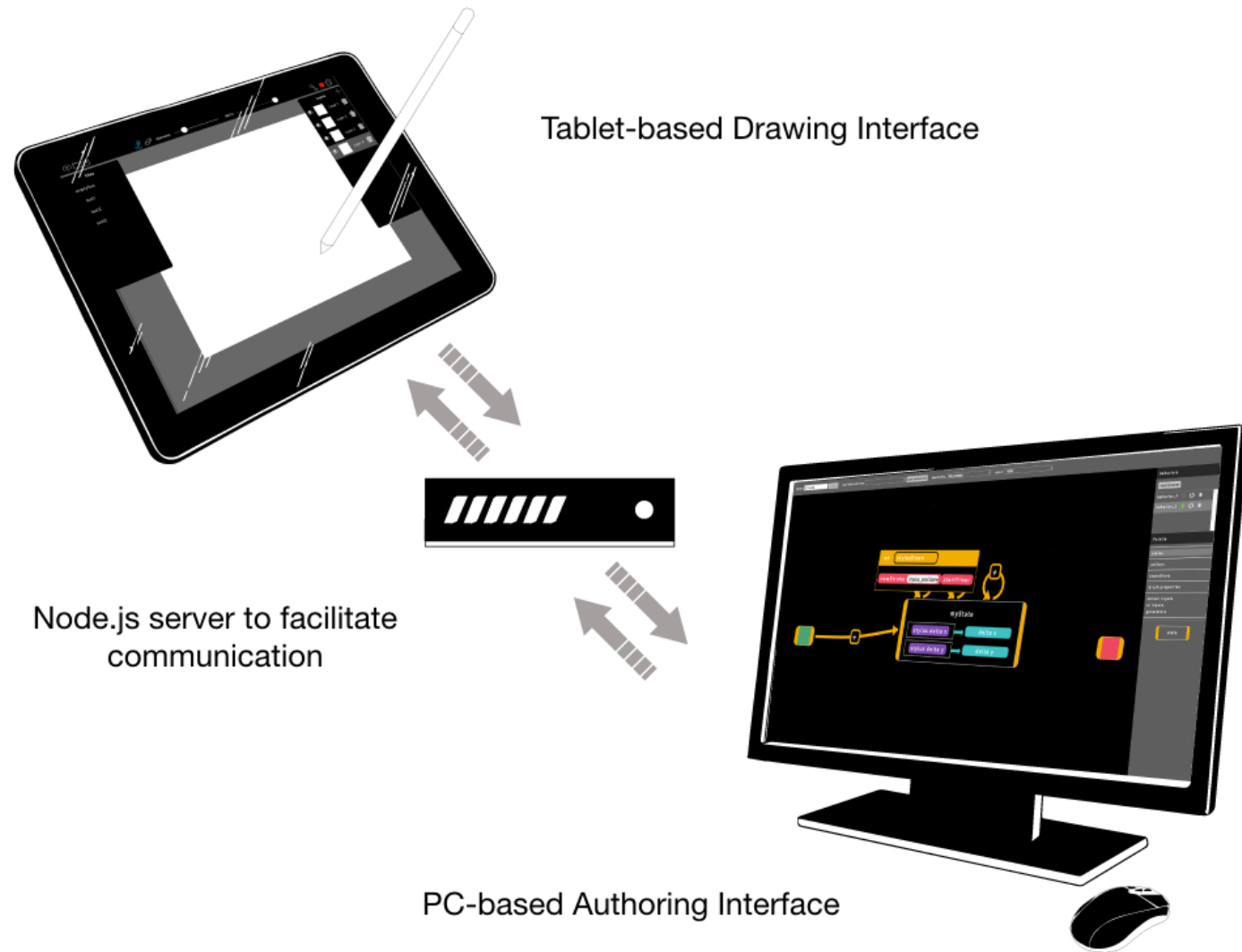
# 2013-Today



Blank

# Dynamic Brushes

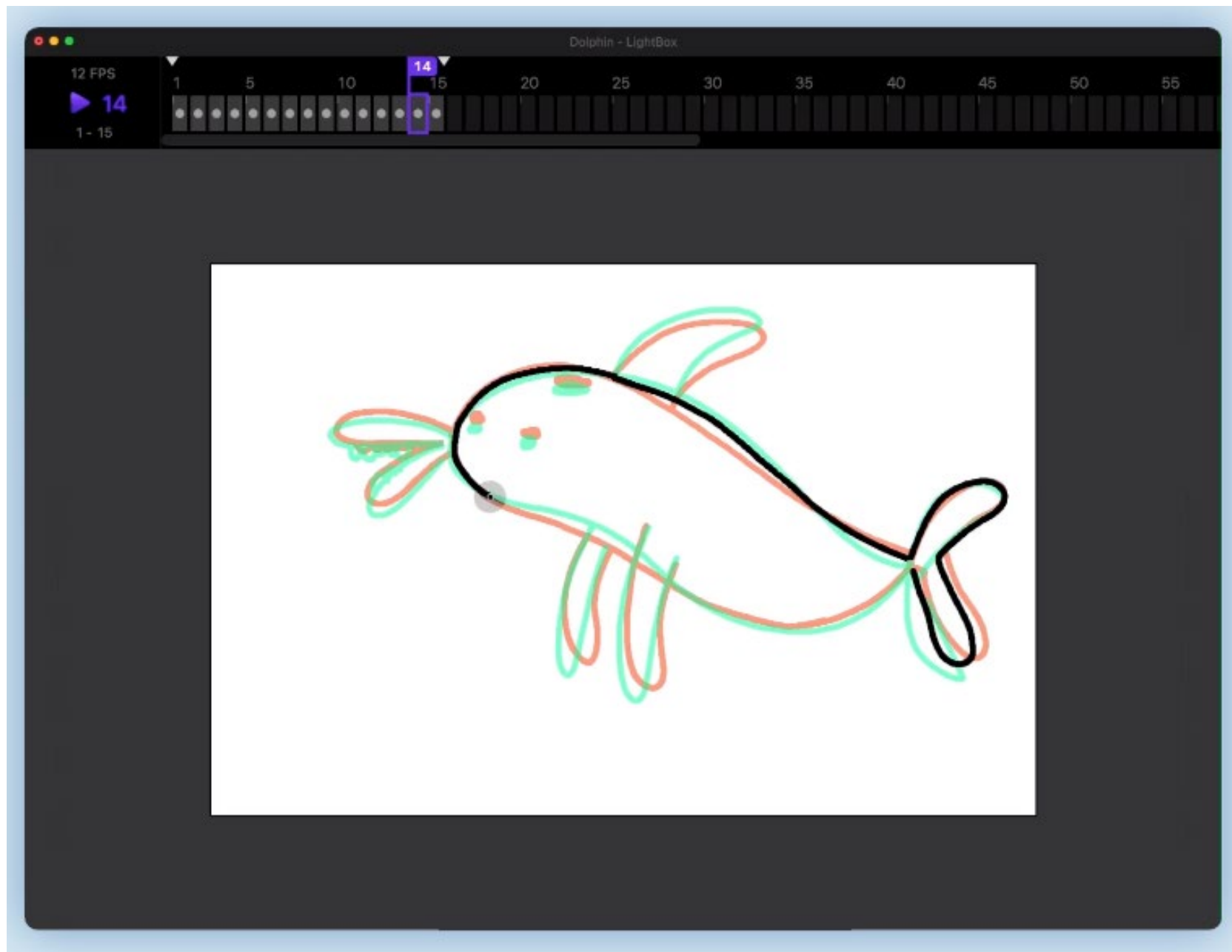
2017



Jennifer Jacobs

# Lightbox

2018-Today



Pasquale D'Silva, Jacob Bijani and Wojtek Witkowski



# Visual Interfaces

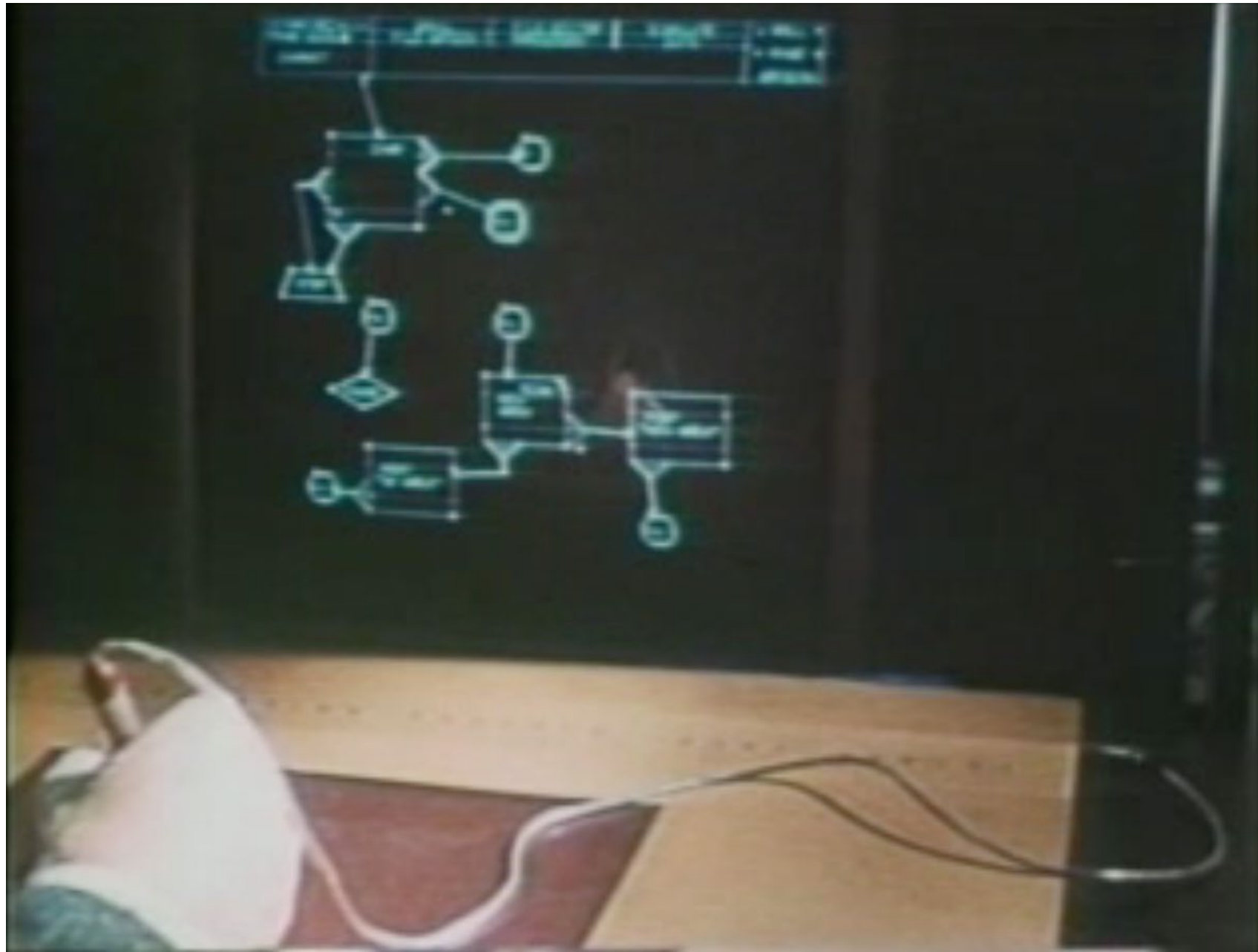
Across the (Design-Development) Divide

# **Node-Based (Graph)**

Environments afford the  
Visual Authoring of Programs

# GRail

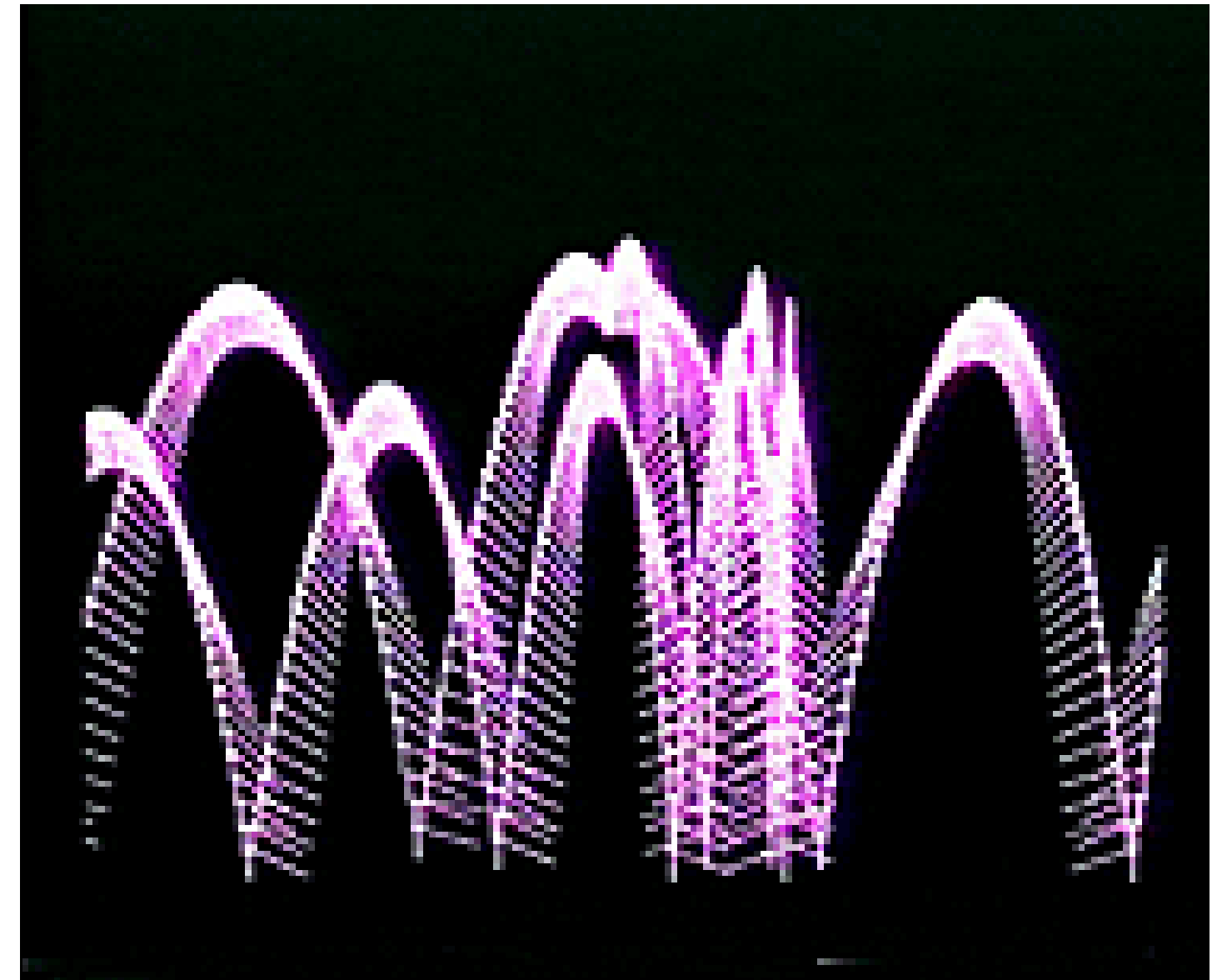
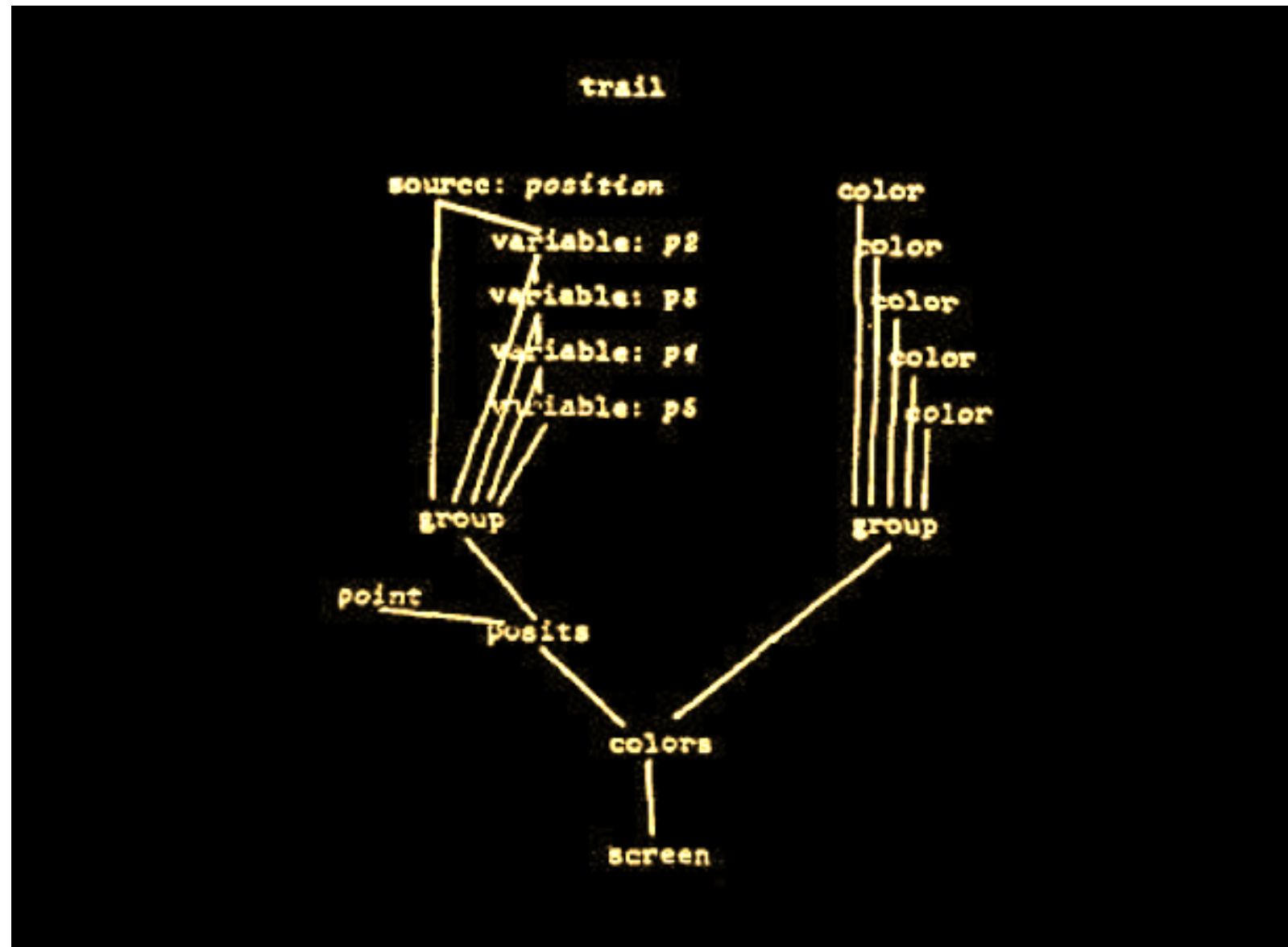
1968



RAND Corporation

# EOM (Graphical Simulation System)

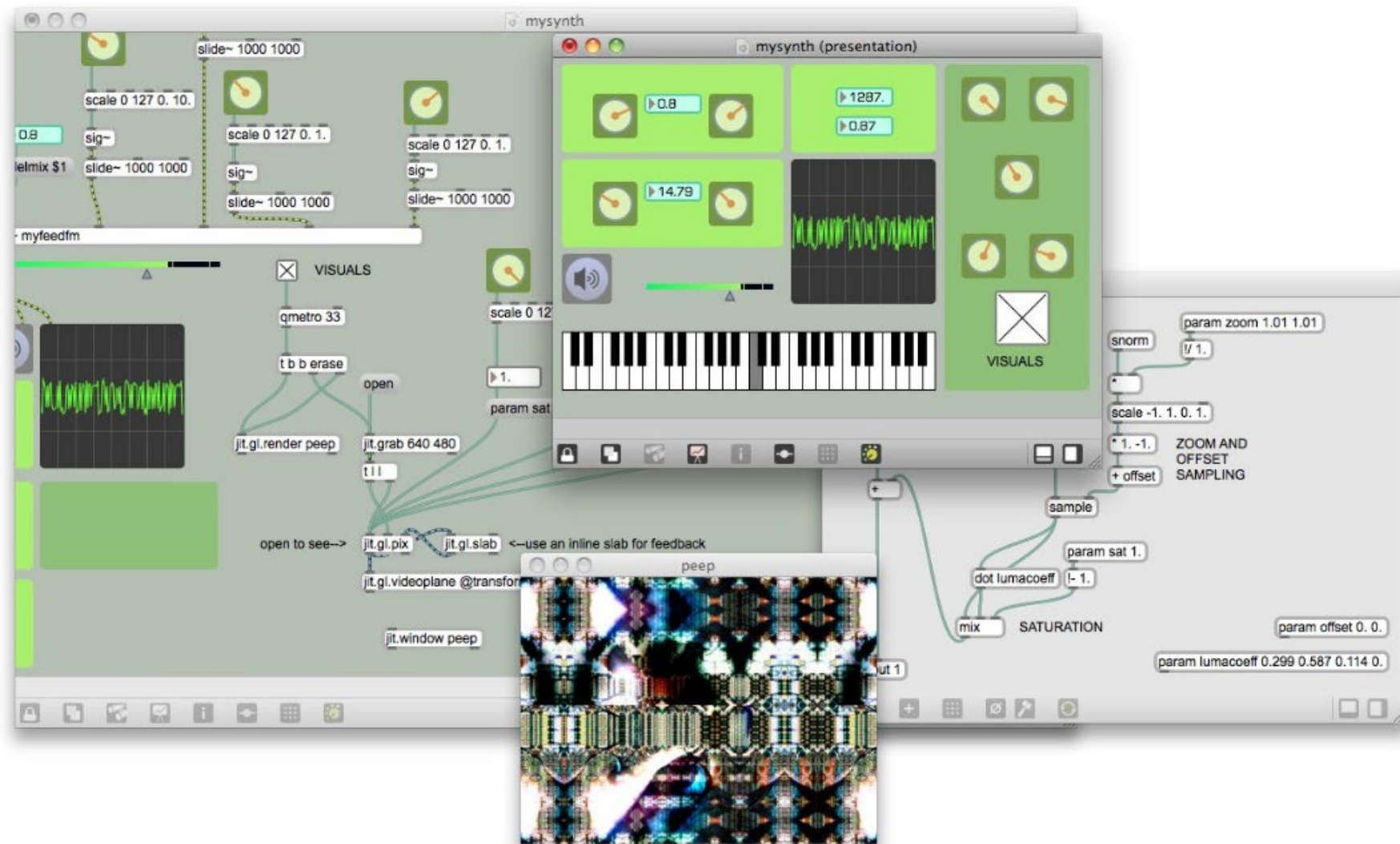
1976-1977



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

# MaxMSP

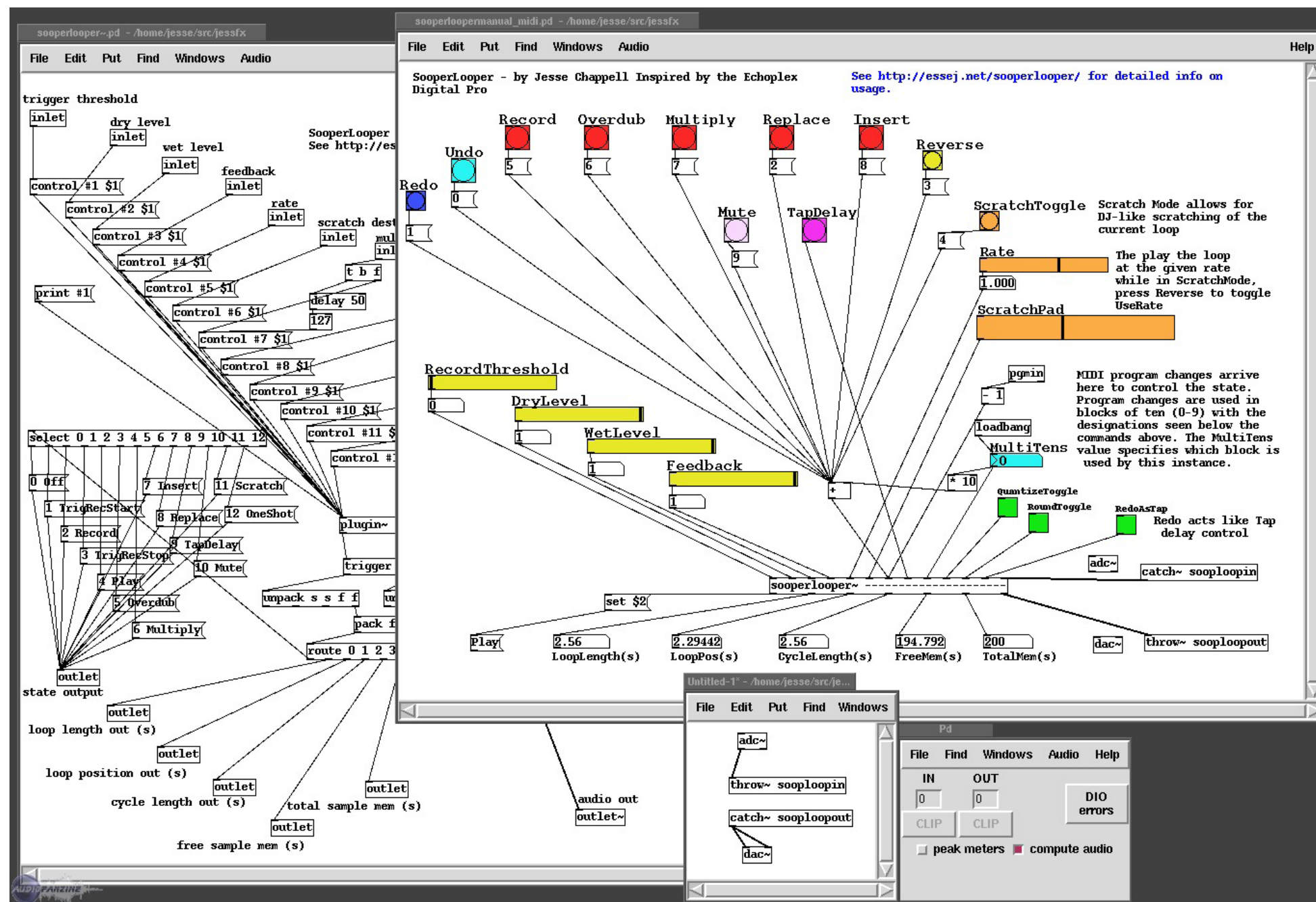
# 1990-Today



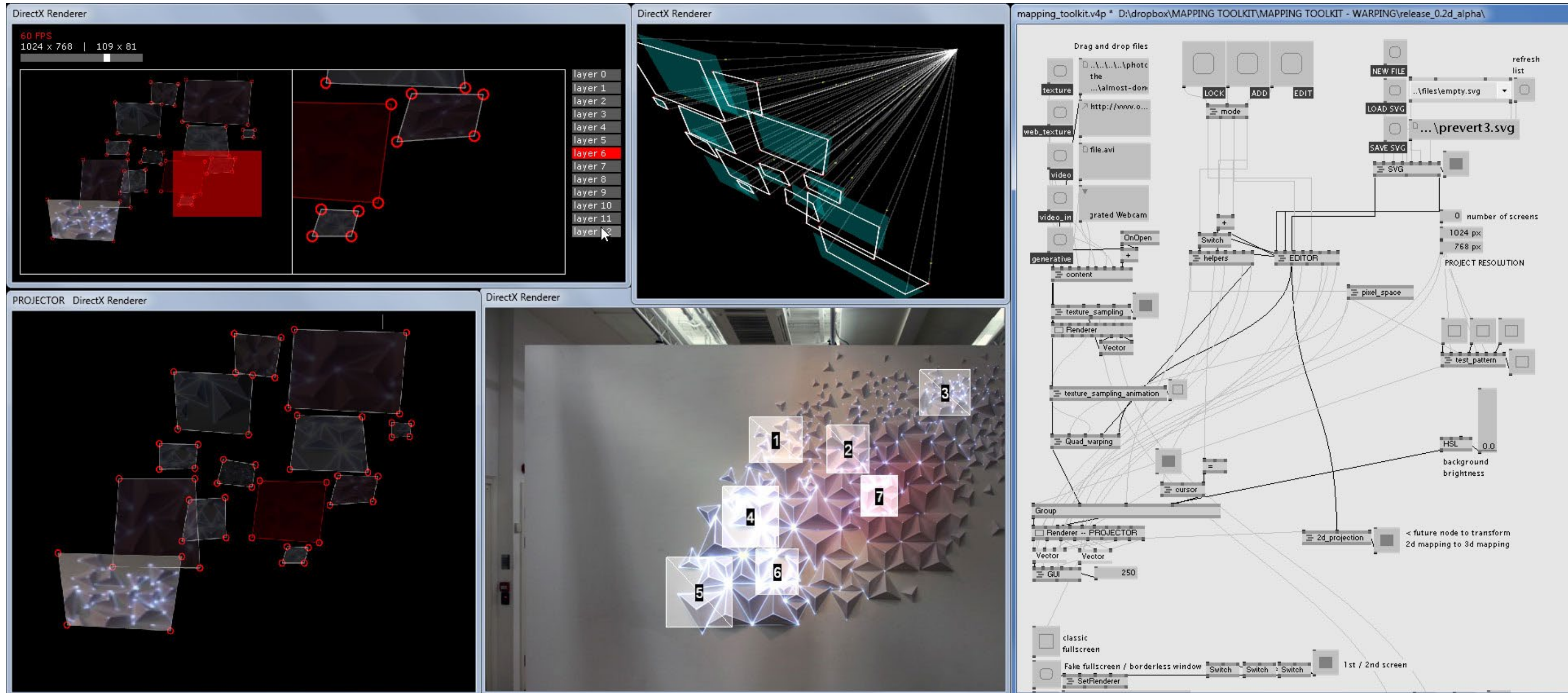
Miller Puckette at IRCAM (Now Cycling '74)

# Pure Data

# 1996-Today



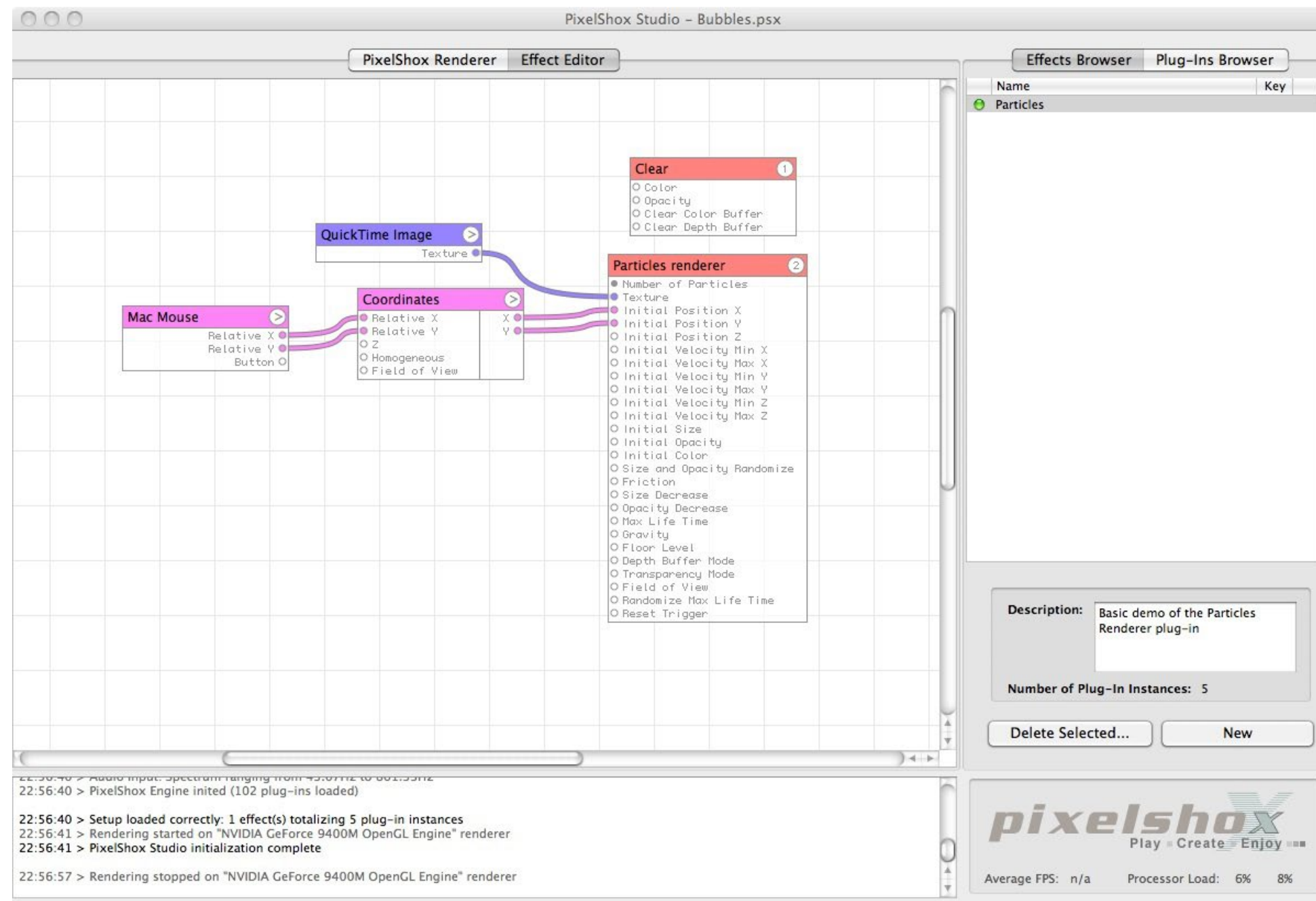
Miller Puckette



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

# PixelShox Studio

2002-2003

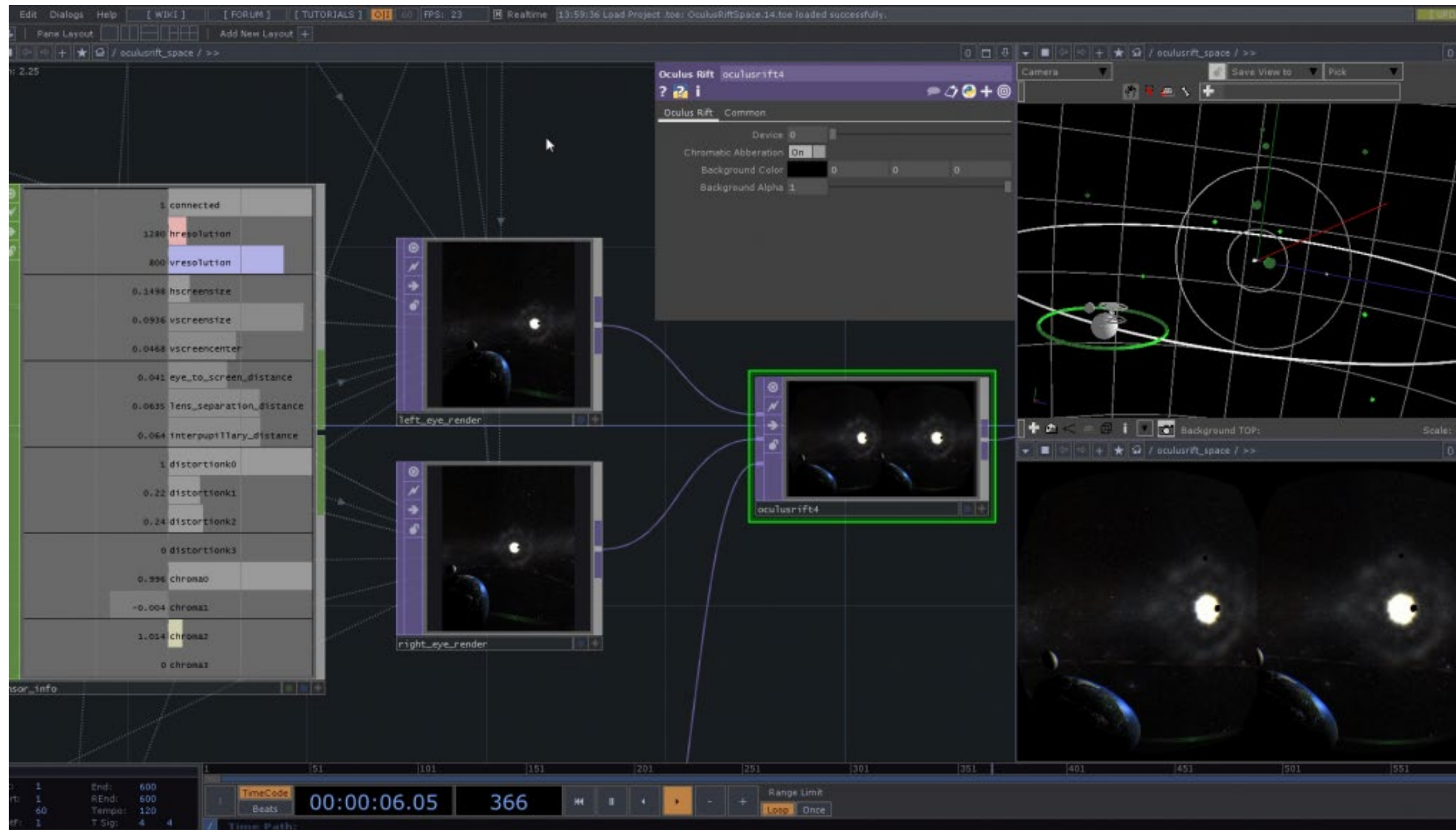


Pierre-Olivier Latour for PixelShox (Acquired by Apple)



# TouchDesigner

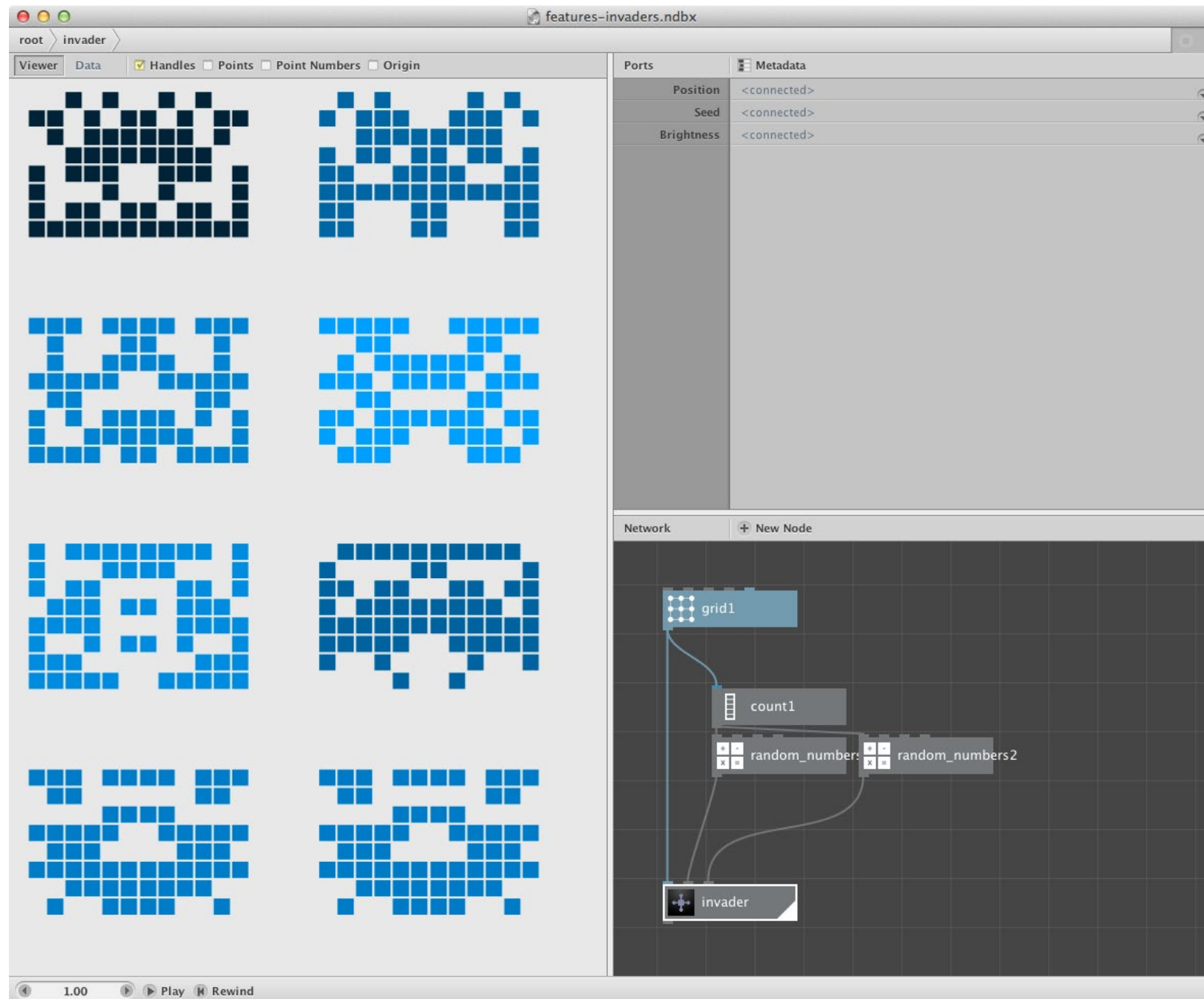
# 2000-Today



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

# NodeBox

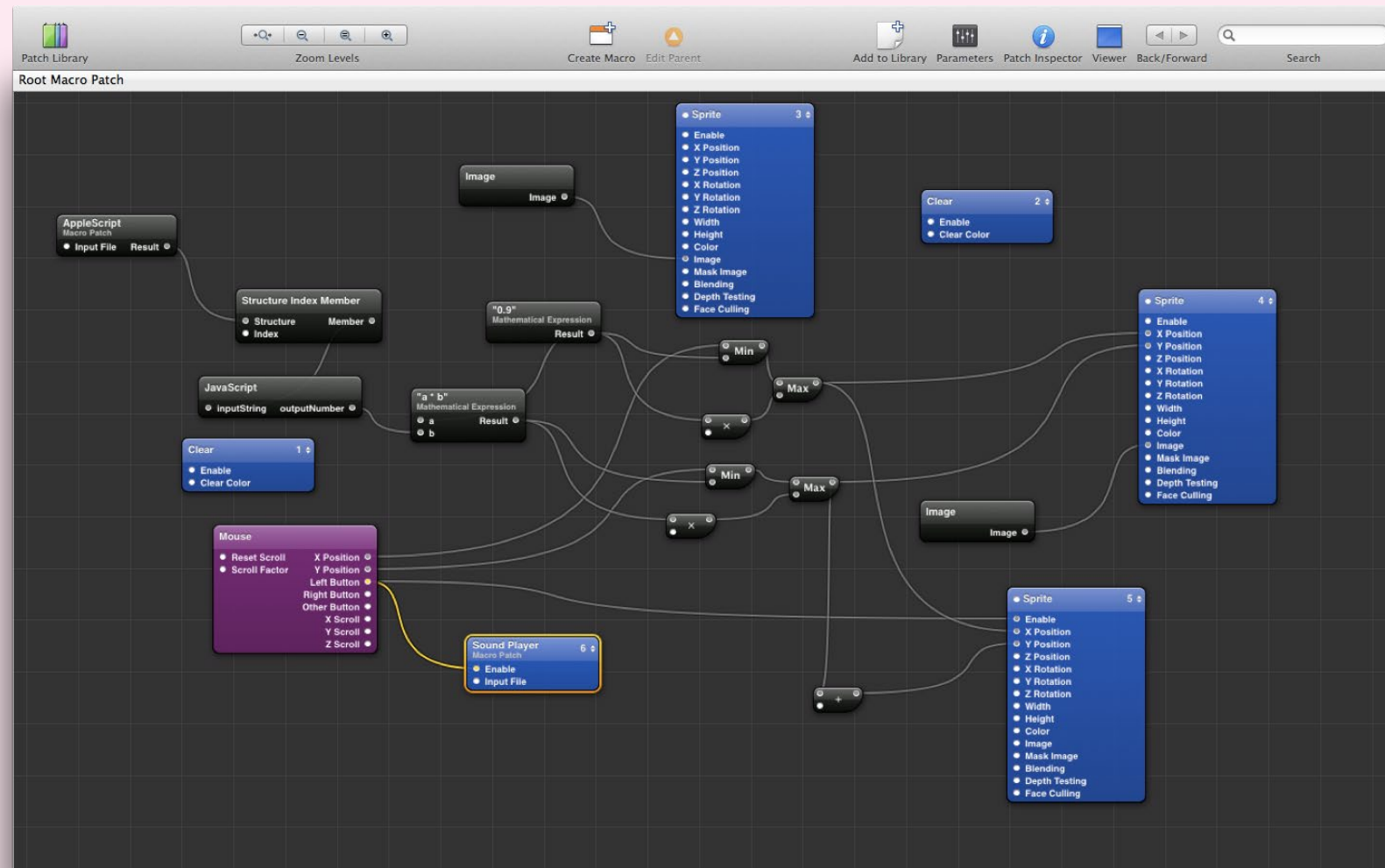
# 2004-Today



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

# Quartz Composer

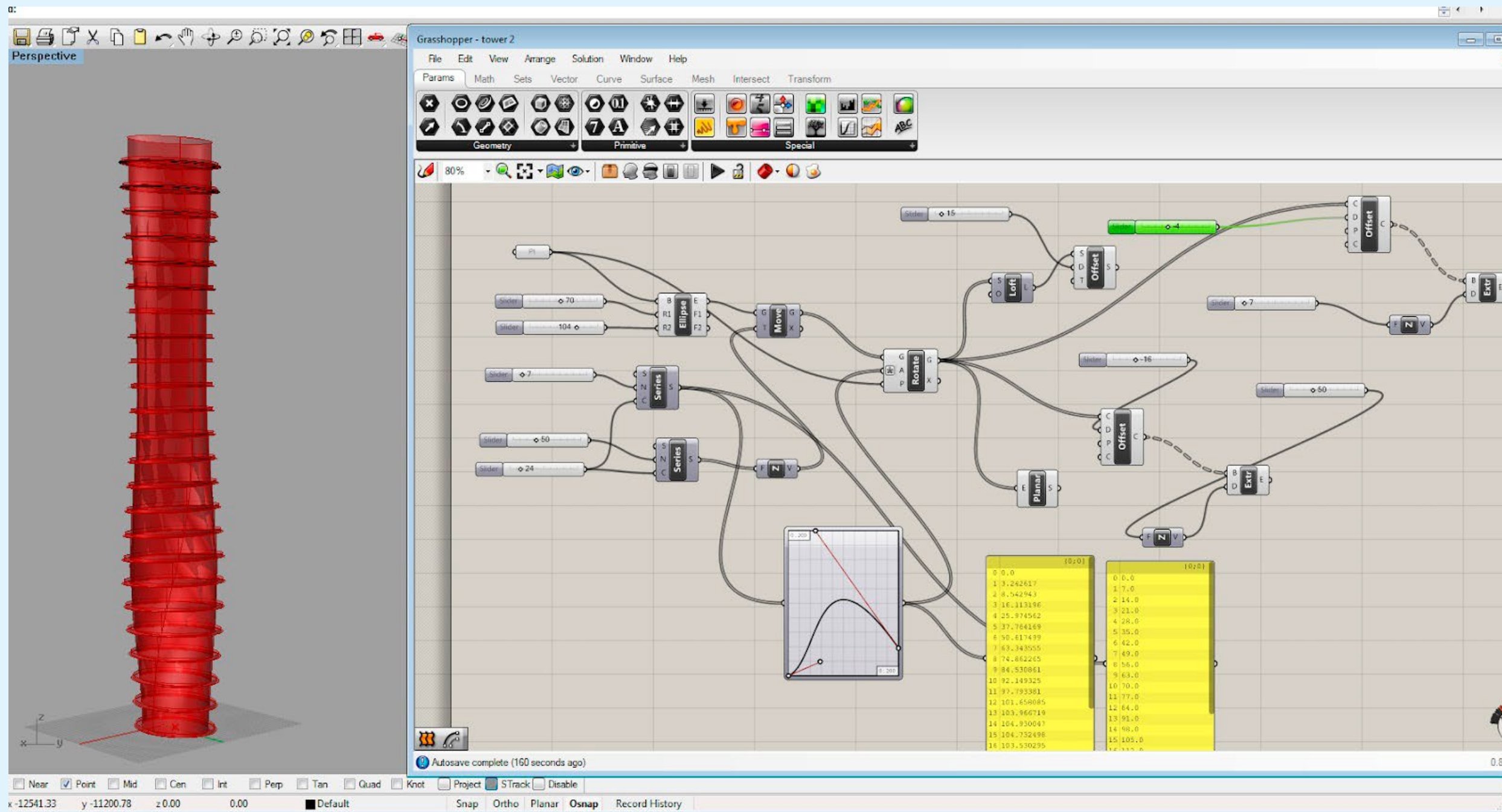
2005-2016



Apple

# Grasshopper for Rhino

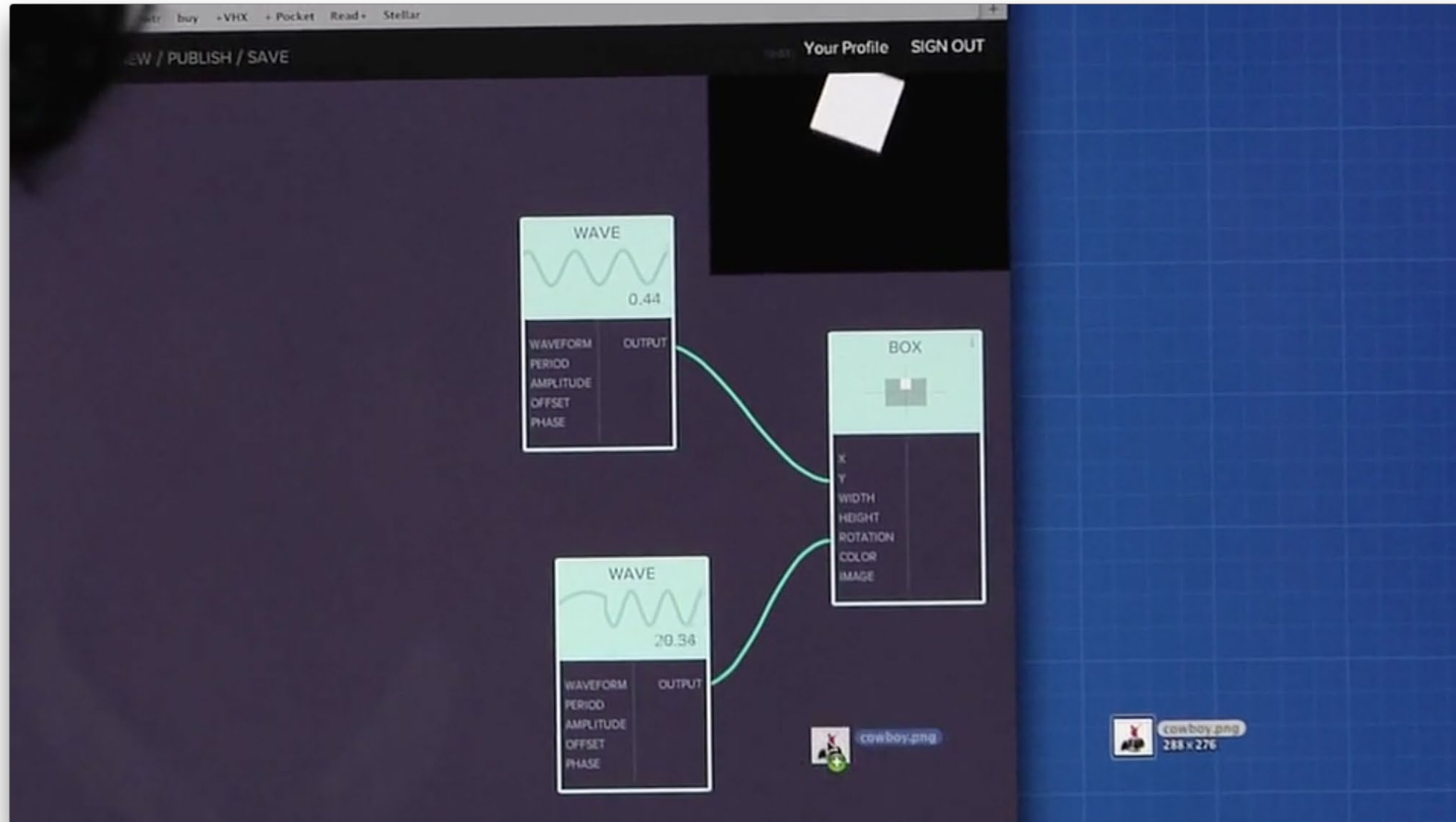
2007-Today



Robert McNeel and David Rutten

# Moonbase

2012



William Cotton and Pasquale D'Silva

# Praxis Live

# 2012-Today

The screenshot displays the Praxis LIVE software interface, which is a live coding environment. The main window is divided into several sections:

- Visual Patch:** A central area showing a network of interconnected components. On the left, there's a 'clock-1' component with a 'subdivision' parameter. Below it is a 'drums' component with a table of parameters: 

Parameter	Value
play-1	play-1
play-2	play-2
play-3	play-3
play-4	play-4
play-5	play-5
play-6	play-6
sample-1	file:/home/nsi/...
sample-2	file:/home/nsi/...
sample-3	file:/home/nsi/...
sample-4	file:/home/nsi/...
sample-5	file:/home/nsi/...
sample-6	file:/home/nsi/...
level-1	1.0
level-2	0.8
level-3	0.9
level-4	1.0
level-5	0.8
level-6	0.8

. The 'live-coder' component is highlighted with a yellow box and a tooltip that says: "Edit the code of this component, uncommenting the example lines and saving to see how they sound." The 'drums' component has a 'syn1' parameter with a 'resonance' value of 15. The 'mixer-6s' component has six input channels (in-1l to in-6r) and an 'out-1' output.
- Code Editor:** A window titled 'live\_coder\_code.java' showing the source code for the 'live-coder' component. The code includes a 'setup()' method with various parameters and a 'draw()' method with a complex visualization. The code is as follows:

```
@Override
public void setup() {
    NOTE OFFS
    syn1.send("x");
    syn2.send("x");

    transmit("syn1", "resonance", 15);
    transmit("syn1", "osc-2-level", 0.8);
    transmit("syn1", "osc-2-transpose", 7);
    transmit("syn1", "waveform-2", "Saw");
    // transmit("syn1", "waveform-2", "Square");
    transmit("syn1", "decay", 0.5);

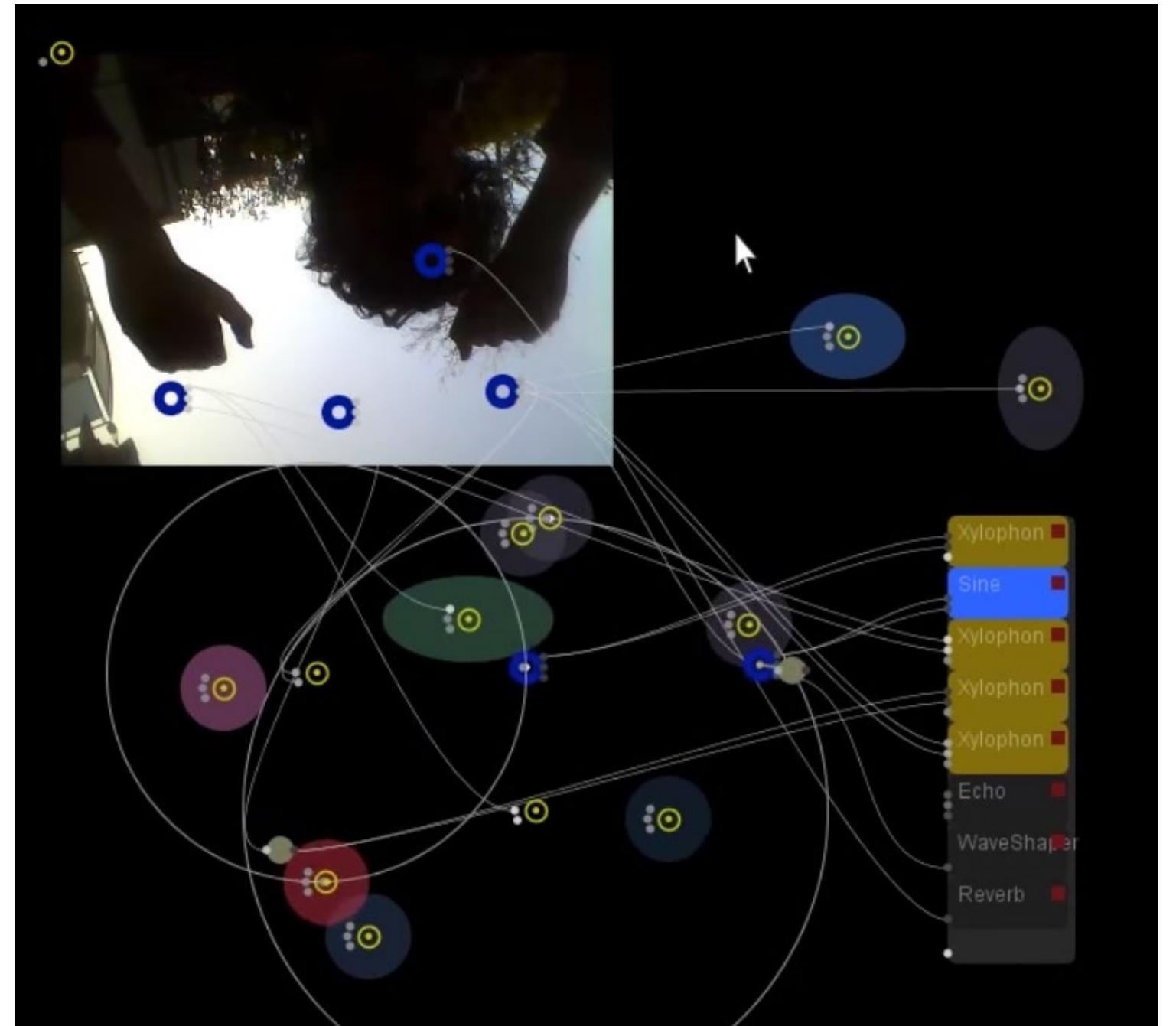
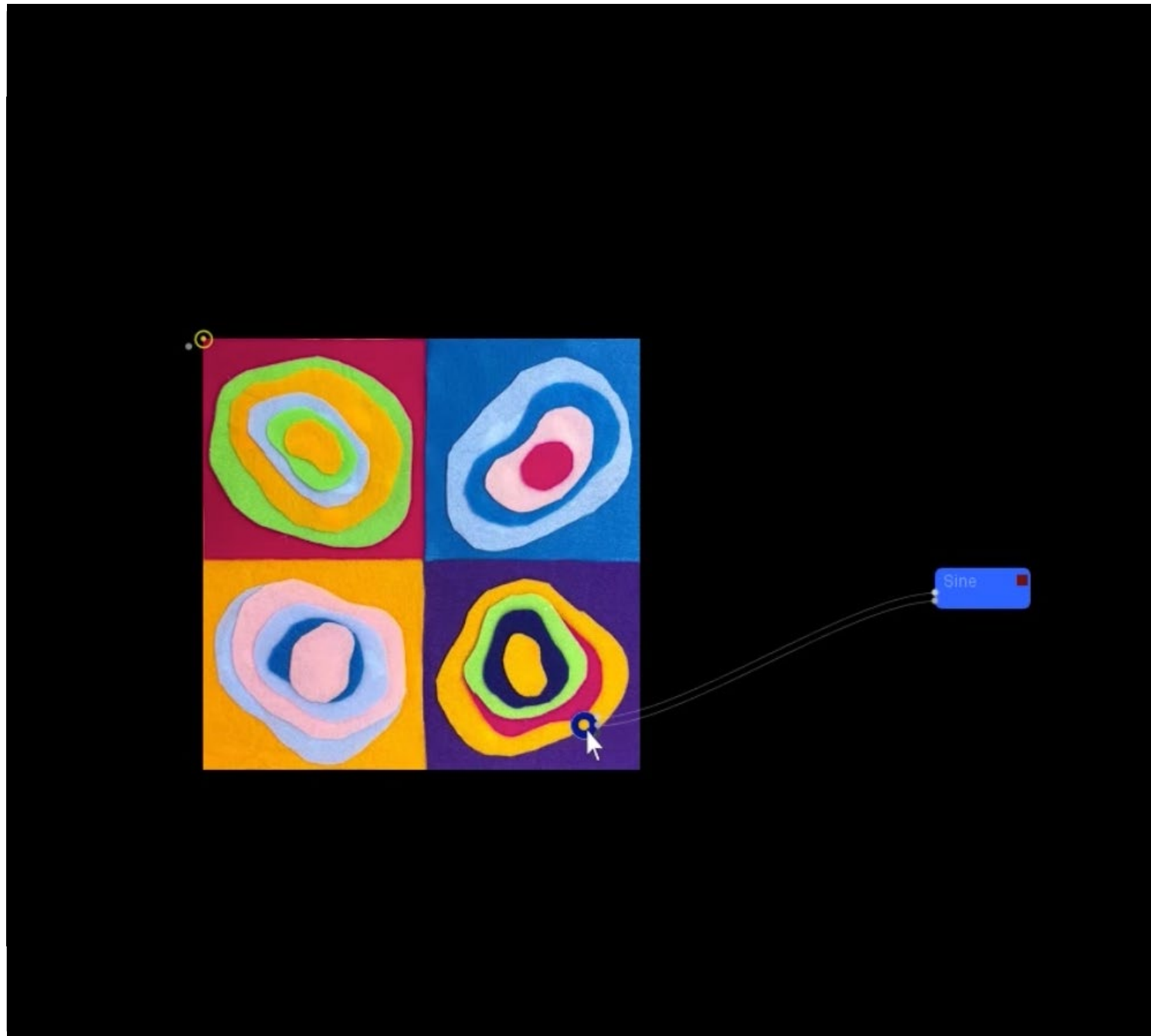
    // DRUMS / FX

    play(bd1, on(0,8));
    play(sn2, on(4,12));
    play(syn2, sometimes(0.25, on(5,6,7,13,14,15)));
    play(hh1, sometimes(0.9));
    play(syn1, seq(16, 64, "a2 x g2", "x d2 e2", "g2 c#3 d3 f#4"), cycle(8,13));
    play(hh2, sometimes(0.1));
}
```
- Video Window:** A window titled 'PRAXIS : /video' showing a colorful, abstract visualization of the code's output. It features a dense network of lines and spheres in various colors (red, green, blue, yellow, purple) against a dark background.
- Output Window:** A window titled 'Output' at the bottom left, which is currently empty.

Neil C Smith

# Loligo

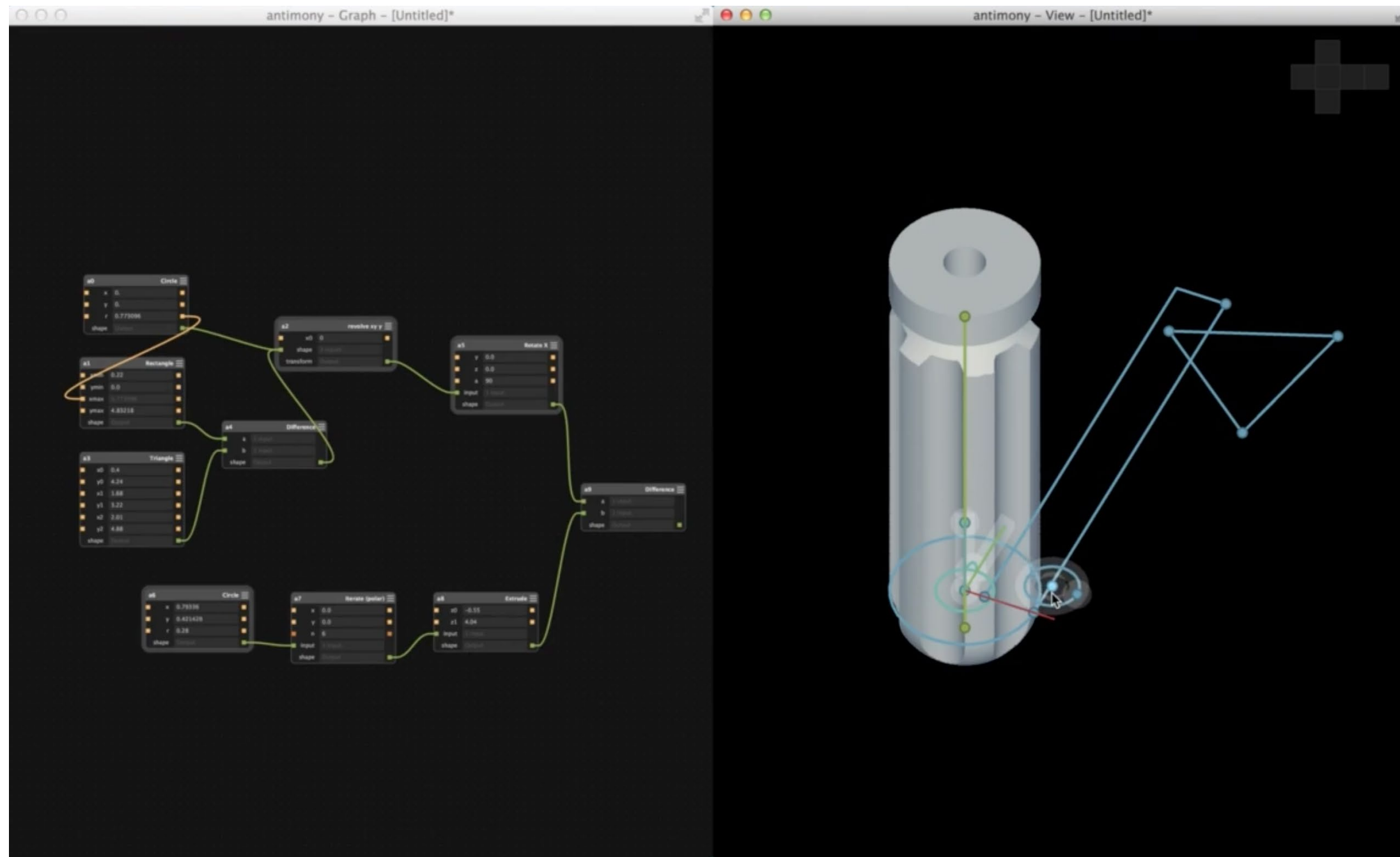
2014



Vanja Cuk

# Antimony

2015



Matt Keeter



# Interaction Prototyping

Empower designers to Build  
'Working-Prototypes'

# Net Lab Toolkit (NTK)

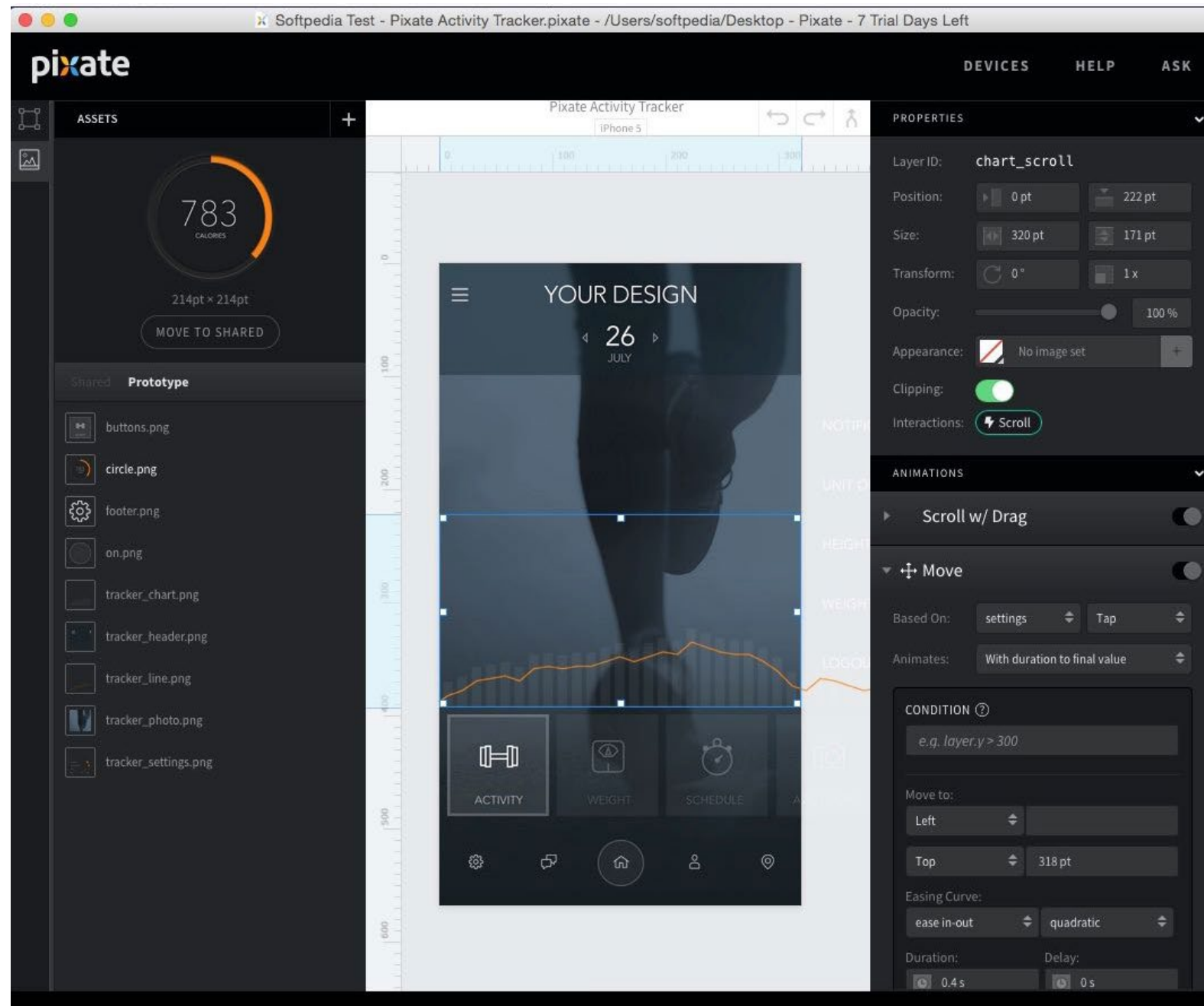
# 2003-Today

The screenshot displays the Net Lab Toolkit (NTK) software interface. On the left is a vertical toolbar with buttons for 'Edit ON', 'Save', 'Import', 'Export', 'Clear', 'Show/Hide Widgets', and 'Full Screen'. The main workspace shows a visual programming workflow with four widgets: 'AnalogIn' (displaying 362), 'IfThen' (IF > 512), 'AnalogOut' (displaying 0), and 'CloudOut' (displaying 0). A hand is shown holding a red potentiometer connected to a blue microcontroller board. On the right is an 'Add Widgets' panel with categories: I/O (AnalogIn, AnalogOut, DigitalIn, DigitalOut, Servo), NETWORK, UI, GENERATOR, LOGIC, and MEDIA.

Philip van Allen at ArtCenter College of Design

# Pixate

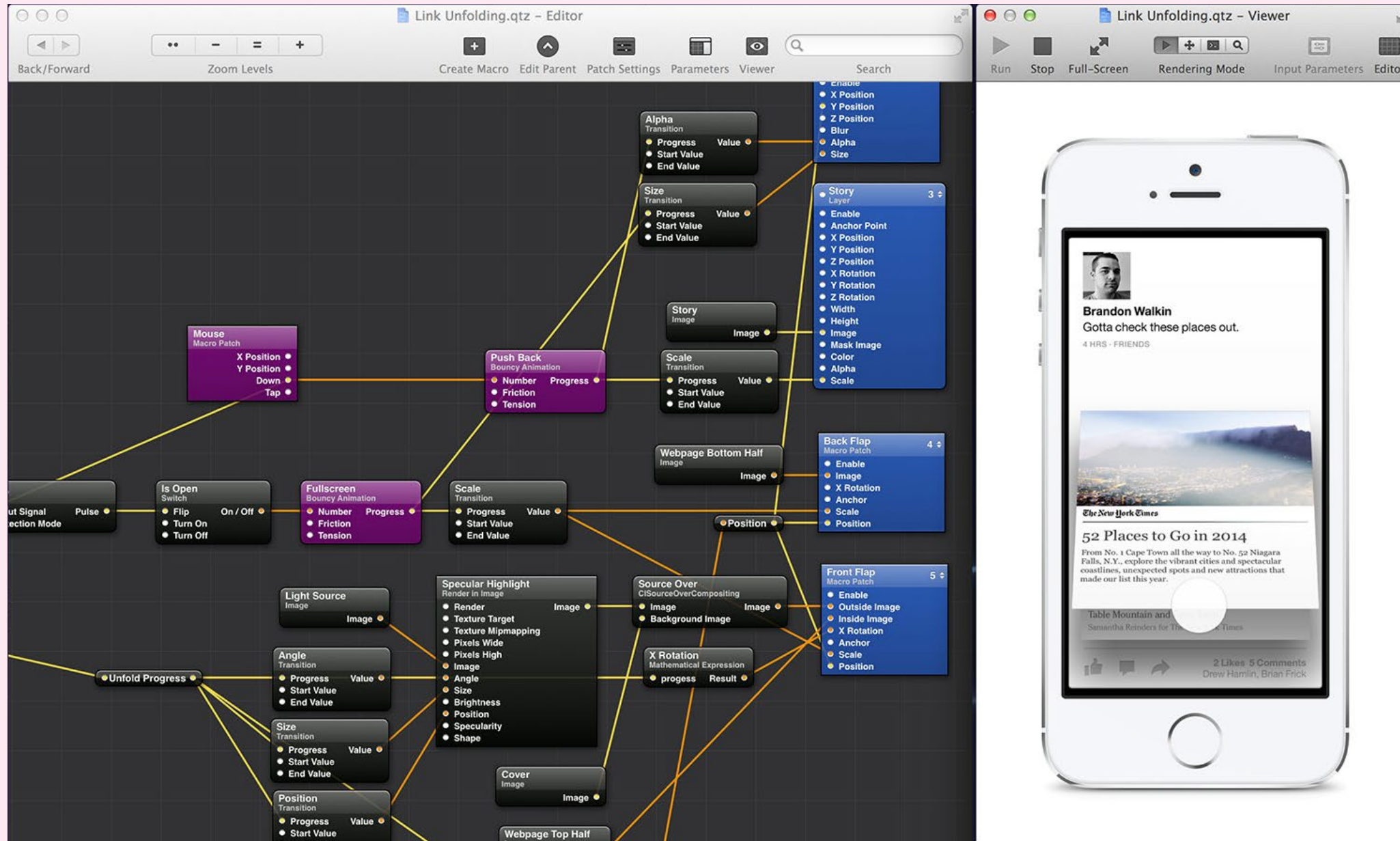
2012-2016



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

# Origami for Quartz Composer

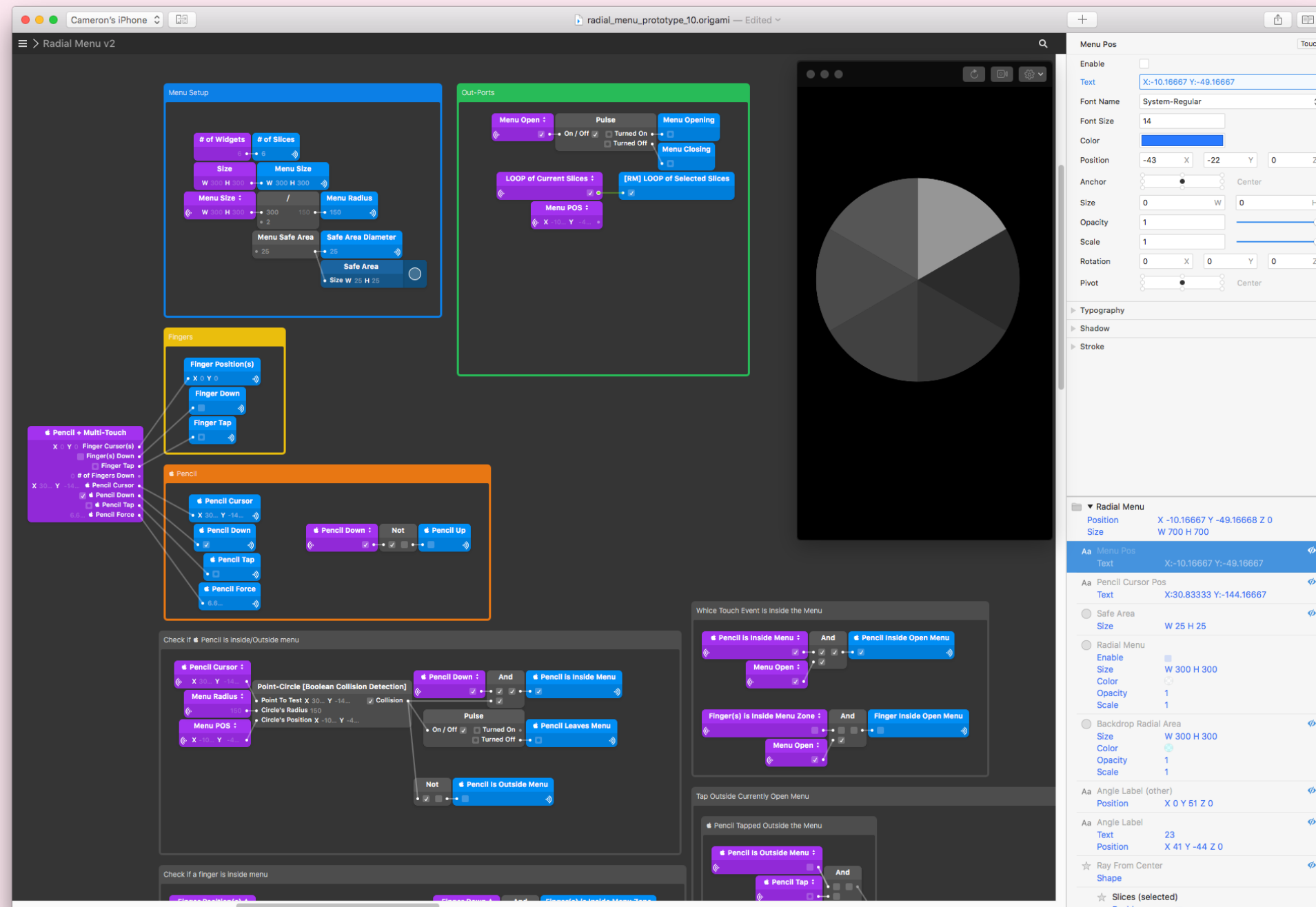
2013-2016



Facebook

# Origami Studio

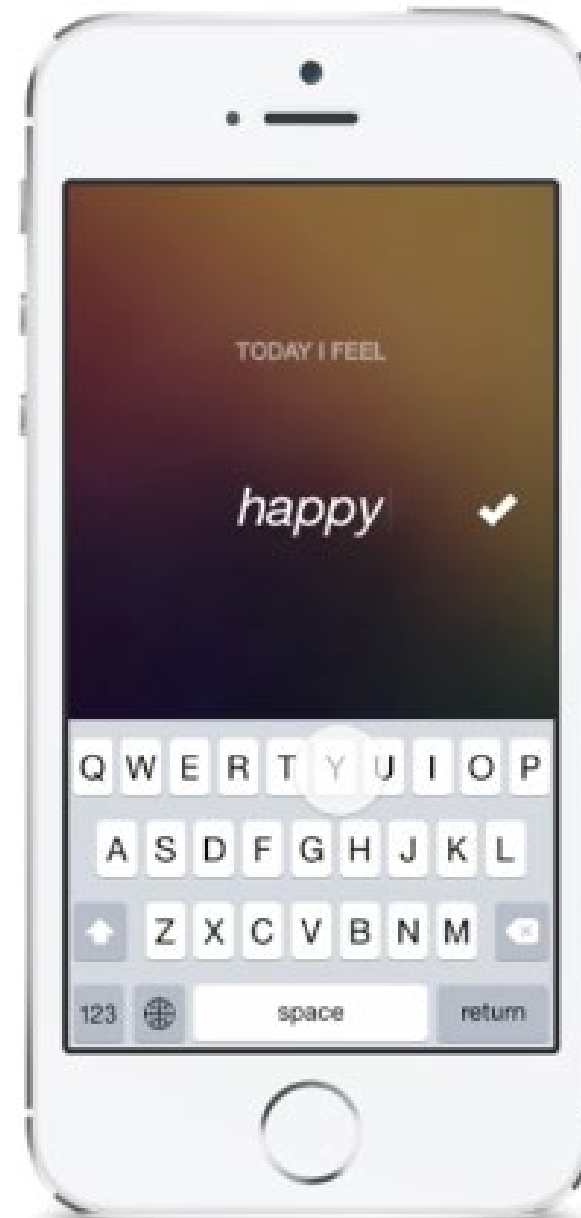
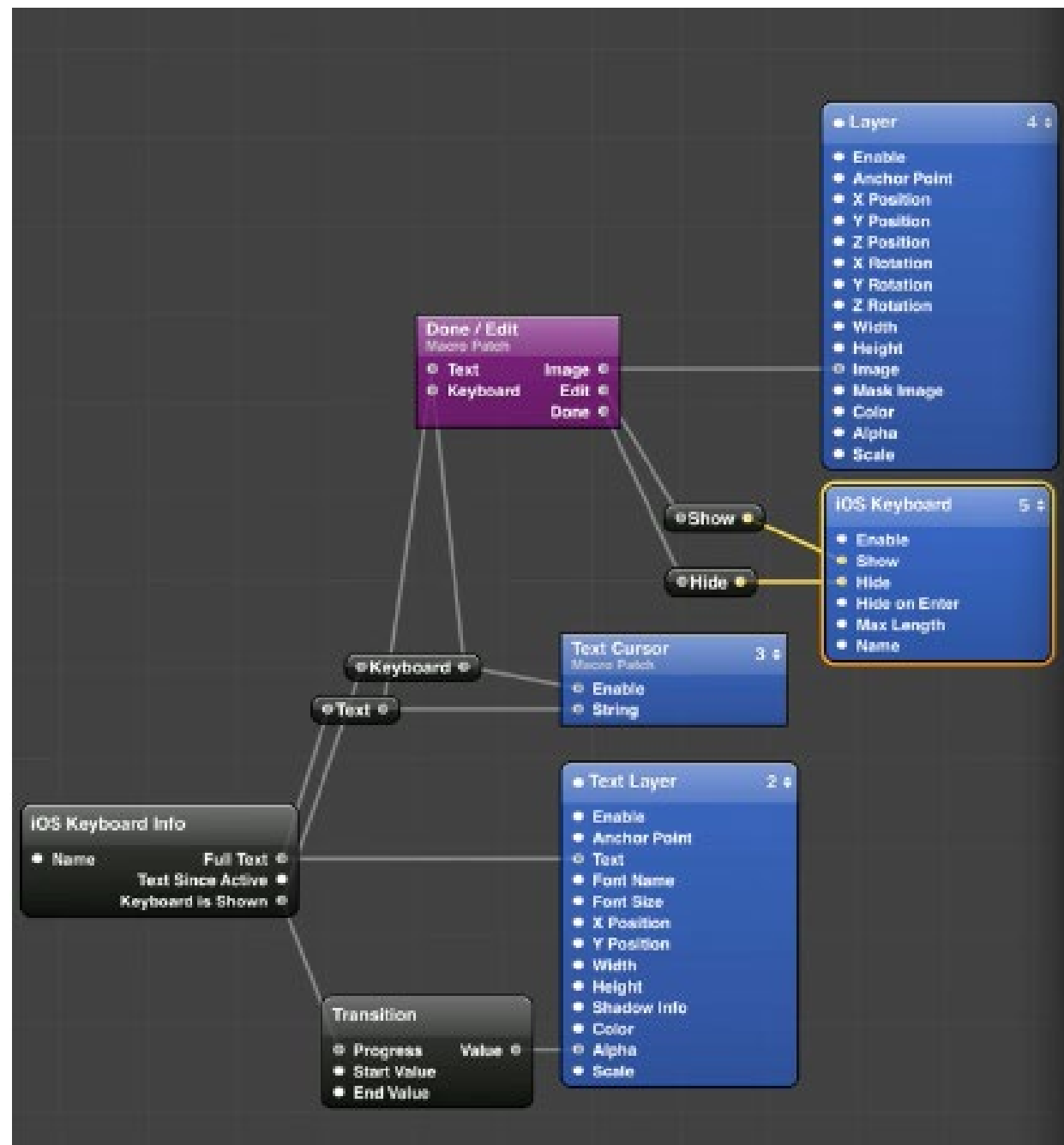
# 2016-Today



Facebook

# Avocado for Quartz Composer

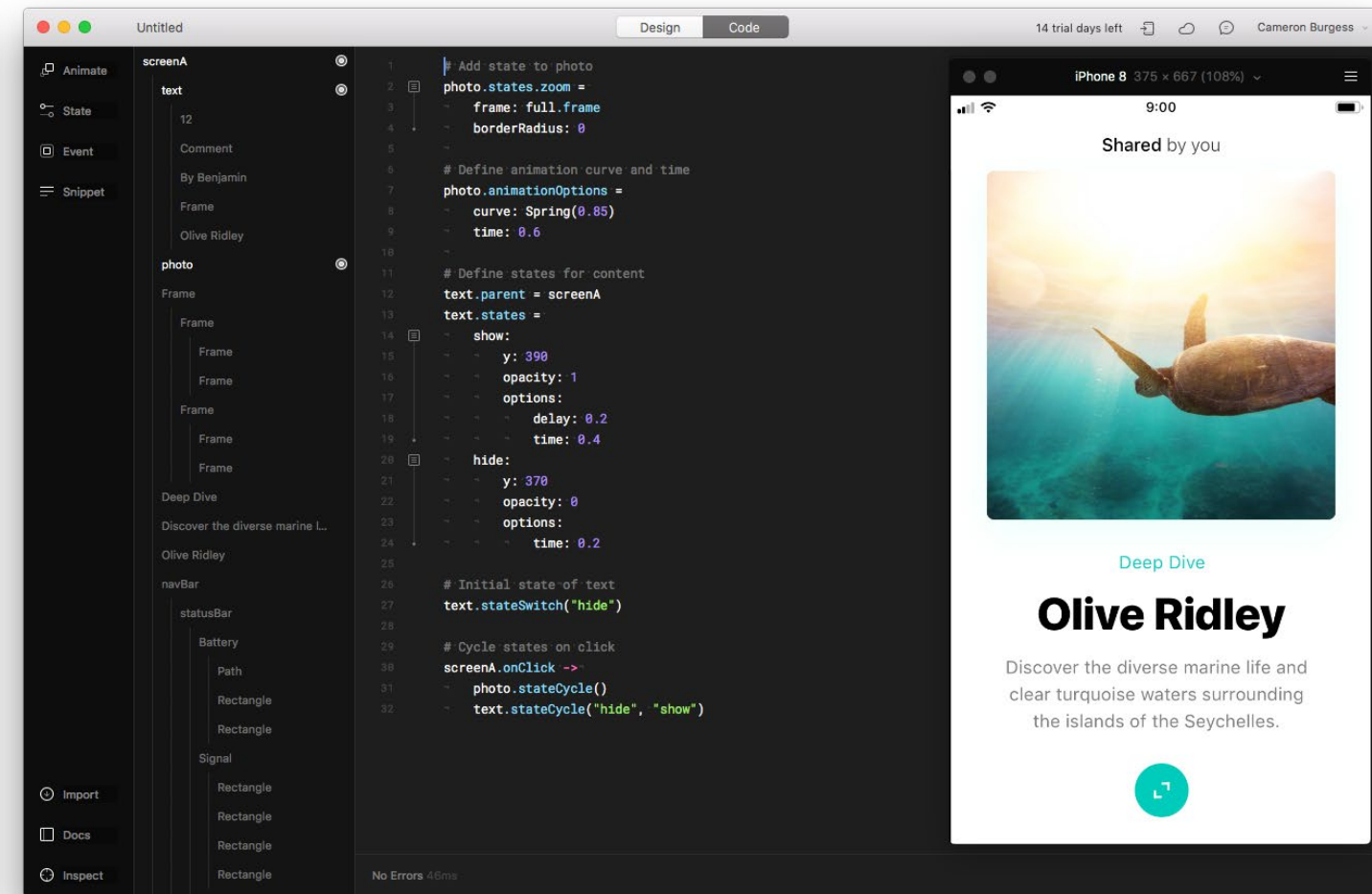
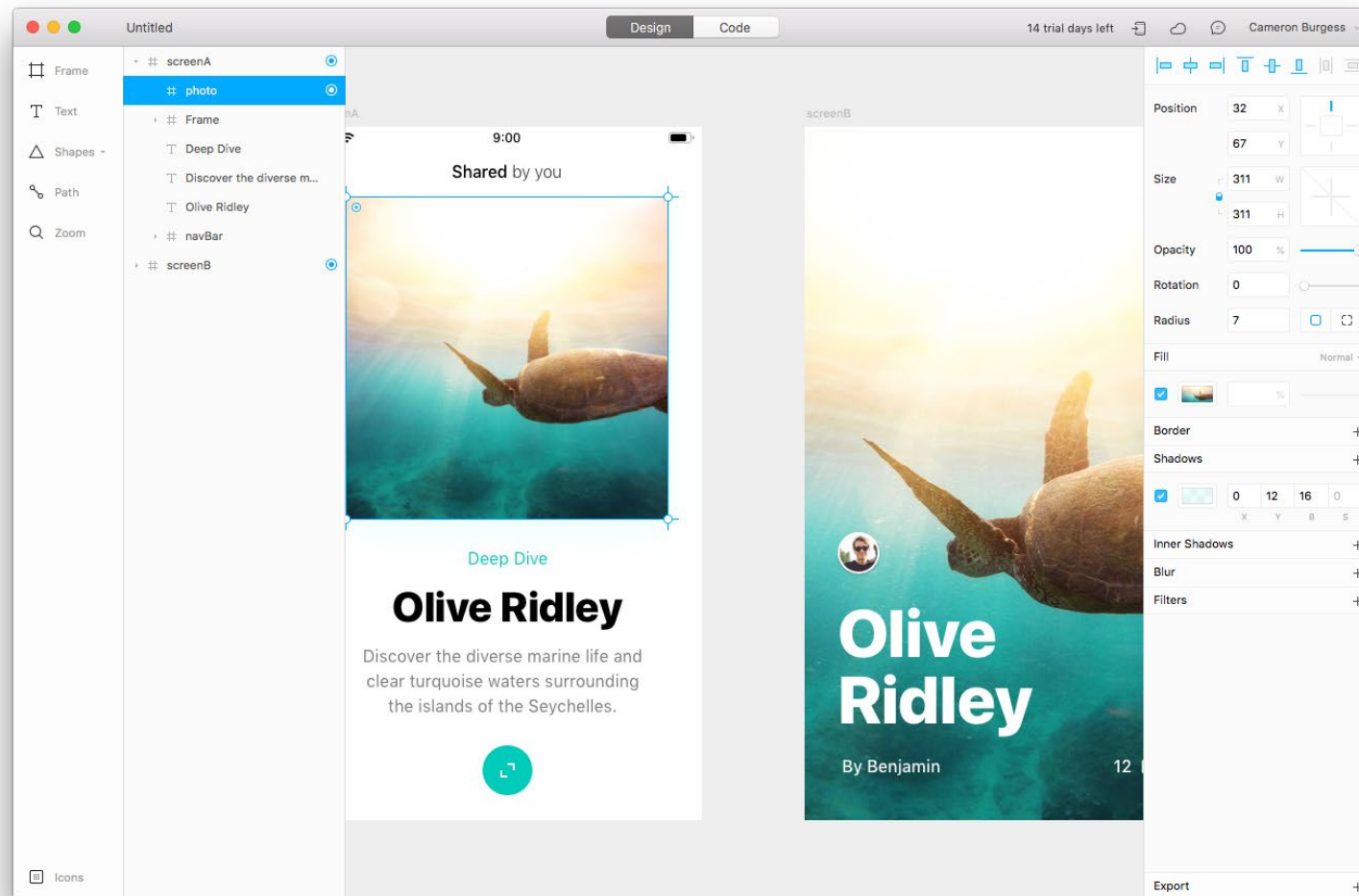
2014



Marco Triverio at IDEO LABS

# Framer Studio

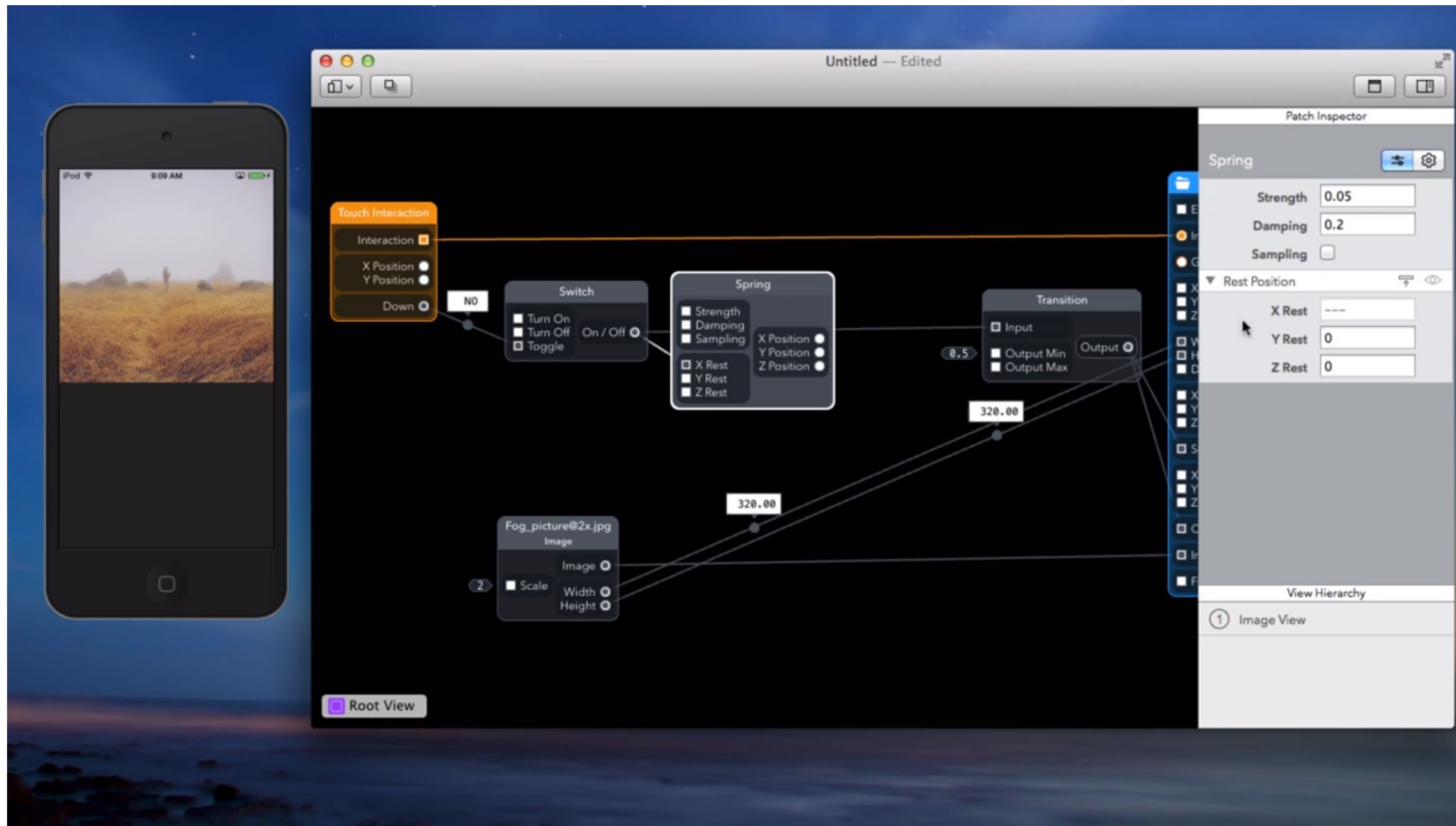
# 2013-Today



Koen Bok and Jorn van Dijk for Framer

# Form

2012-2014

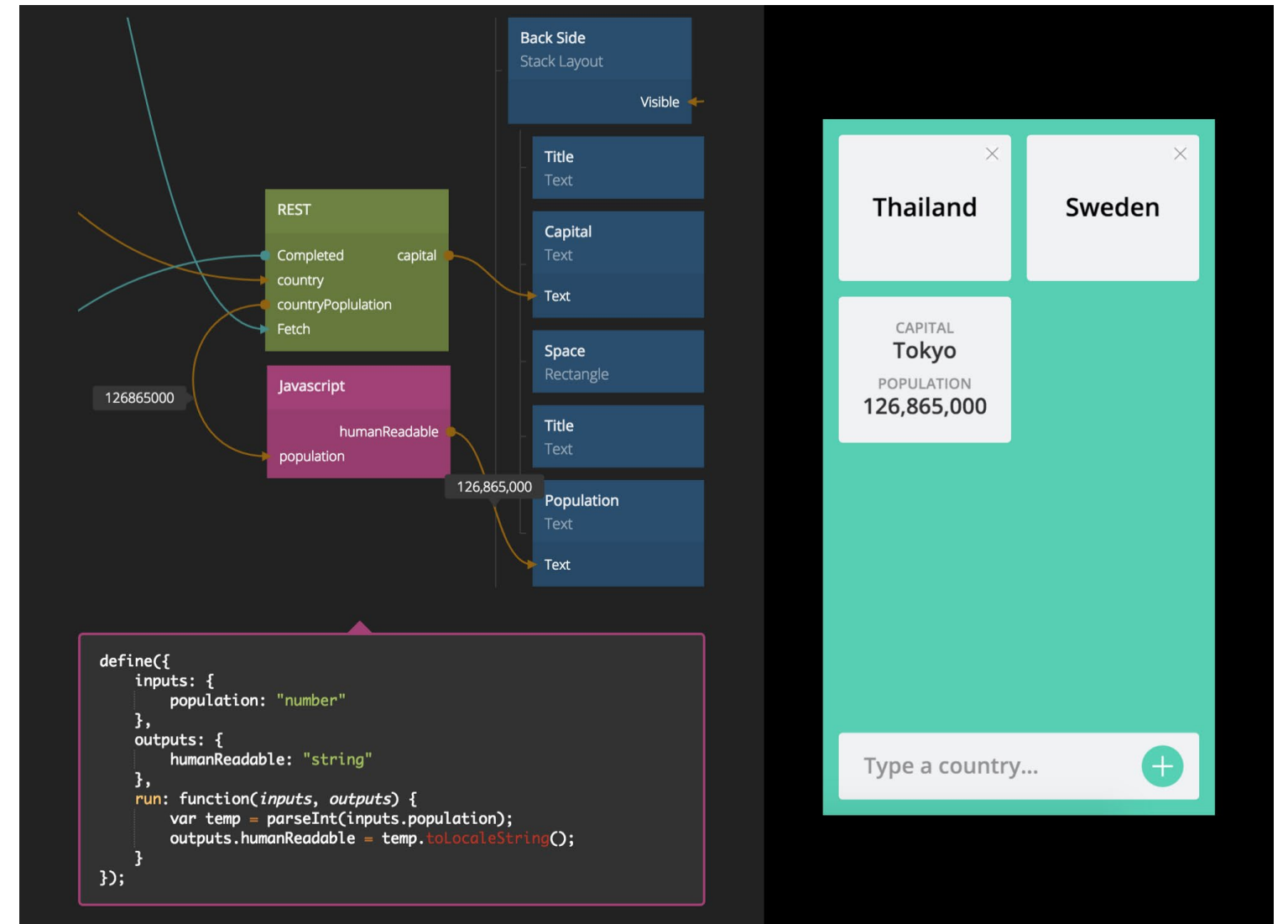


Max Weisel for RelativeWave (Acquired by Google)

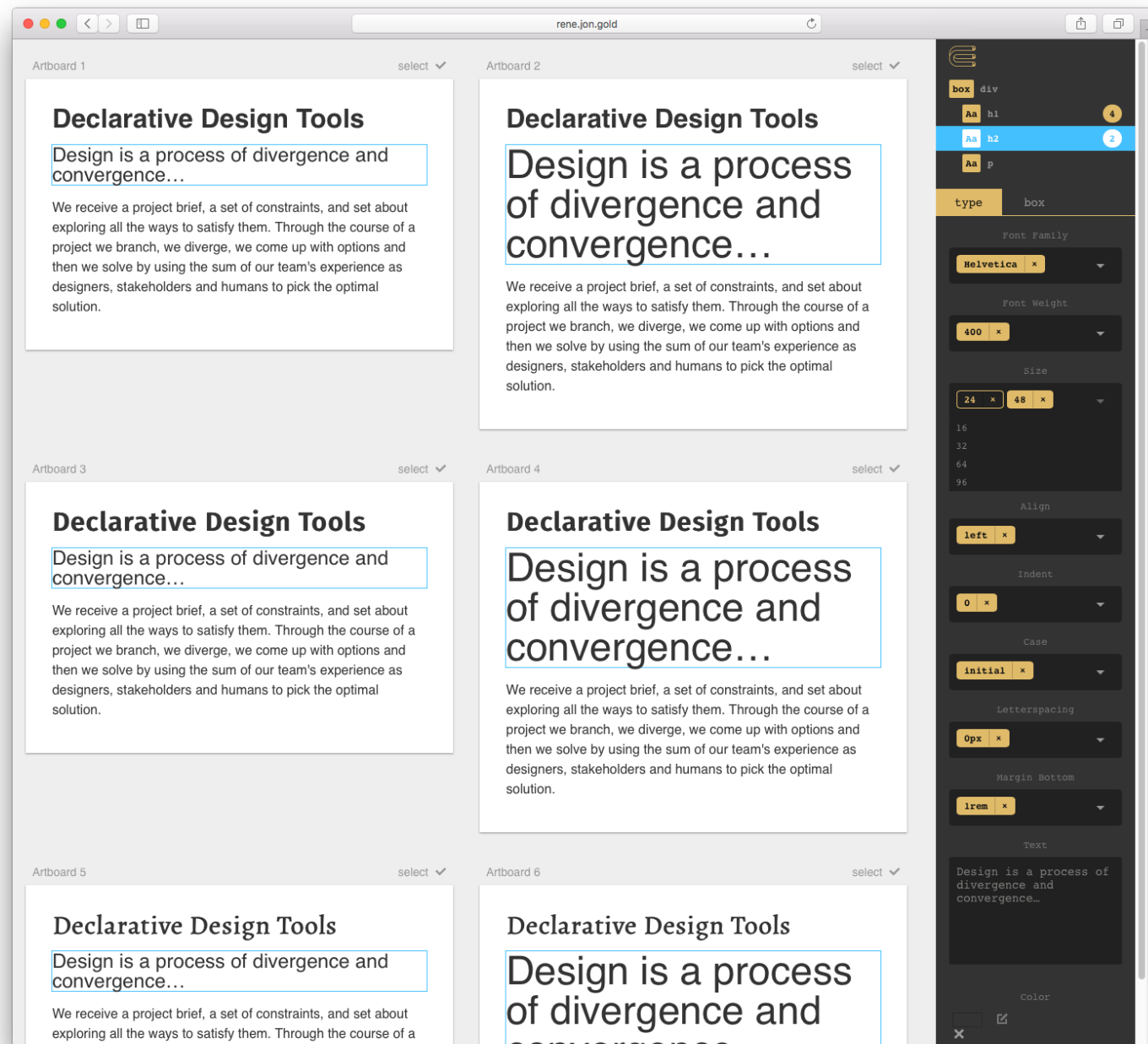


# Noodl

# 2014-Today



## Top Design



# Sketch.systems

2018-Today

The screenshot displays the Sketch.systems web application interface. The browser address bar shows the URL `https://sketch.systems/ryanluc`. The page title is "SketchSystems ALPHA" and it identifies the sketch as "A sketch by ryanlucas". A "Log in to fork this sketch" button is visible in the top right.

The interface is divided into four main sections:

- Spec (See tutorial):** Lists the component's states and transitions:
  - 1 Search Bar\*
  - 2
  - 3 Inactive\*
  - 4 `focused` -> Active
  - 5
  - 6 Active
  - 7 `typed` -> Text Entry
  - 8 `canceled` -> Inactive
  - 9 `cleared` -> Empty
  - 10
  - 11 Empty\*
  - 12
- Code (See tutorial):** Shows the React component code:

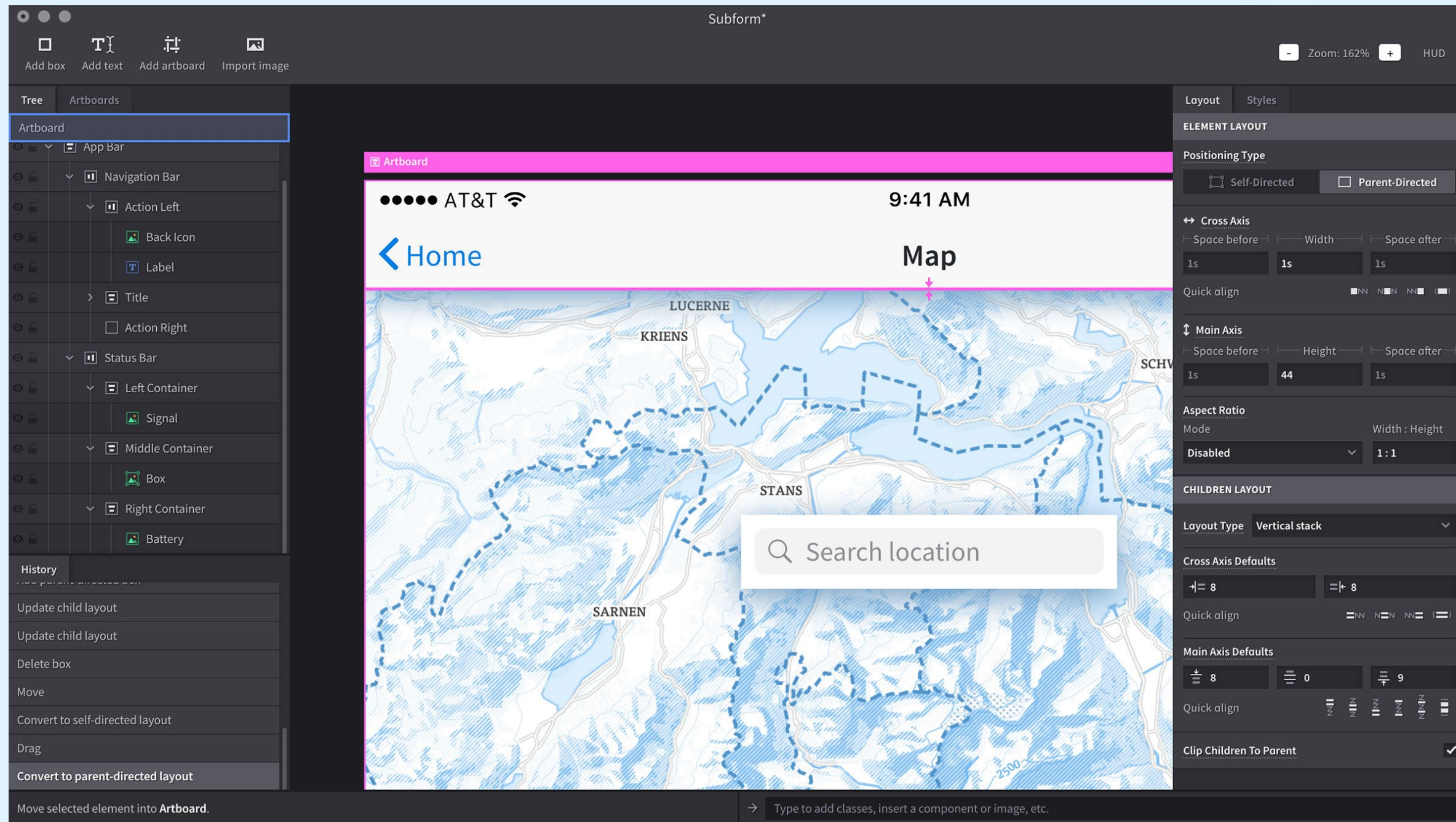
```
1 function render(model){
2   var active_state = model.active_states[0].name;
3   return $('div', {style: {display: 'flex', justifyContent: 'center', height:
4     render_search_bar(active_state, model)}});
5
6 function render_search_bar(active_state, model){
7   if (active_state == 'Inactive') {
8     return $('div',
9       $('input', {type: 'text',
10         placeholder: 'Enter a search term...',
11         style: {border: '1px solid grey', borderRadius: '8px
12         onFocus: function(){ model.emit("focused") }}),
```
- Vis:** Visual representation of the search bar in its "Inactive" state, showing a blue box with the text "Inactive" and a transition arrow pointing to "Active". Below it, the "Active" state is shown with three buttons: "Empty", "Text Entry" (with a transition arrow to "Results"), and "Results".
- Prototype:** A live prototype of the search bar with a text input field containing the placeholder "Enter a search term..." and a "Search" button.

At the bottom right, there is a link: "Have feedback or questions? [Contact us](#)".

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

# Subform

2018



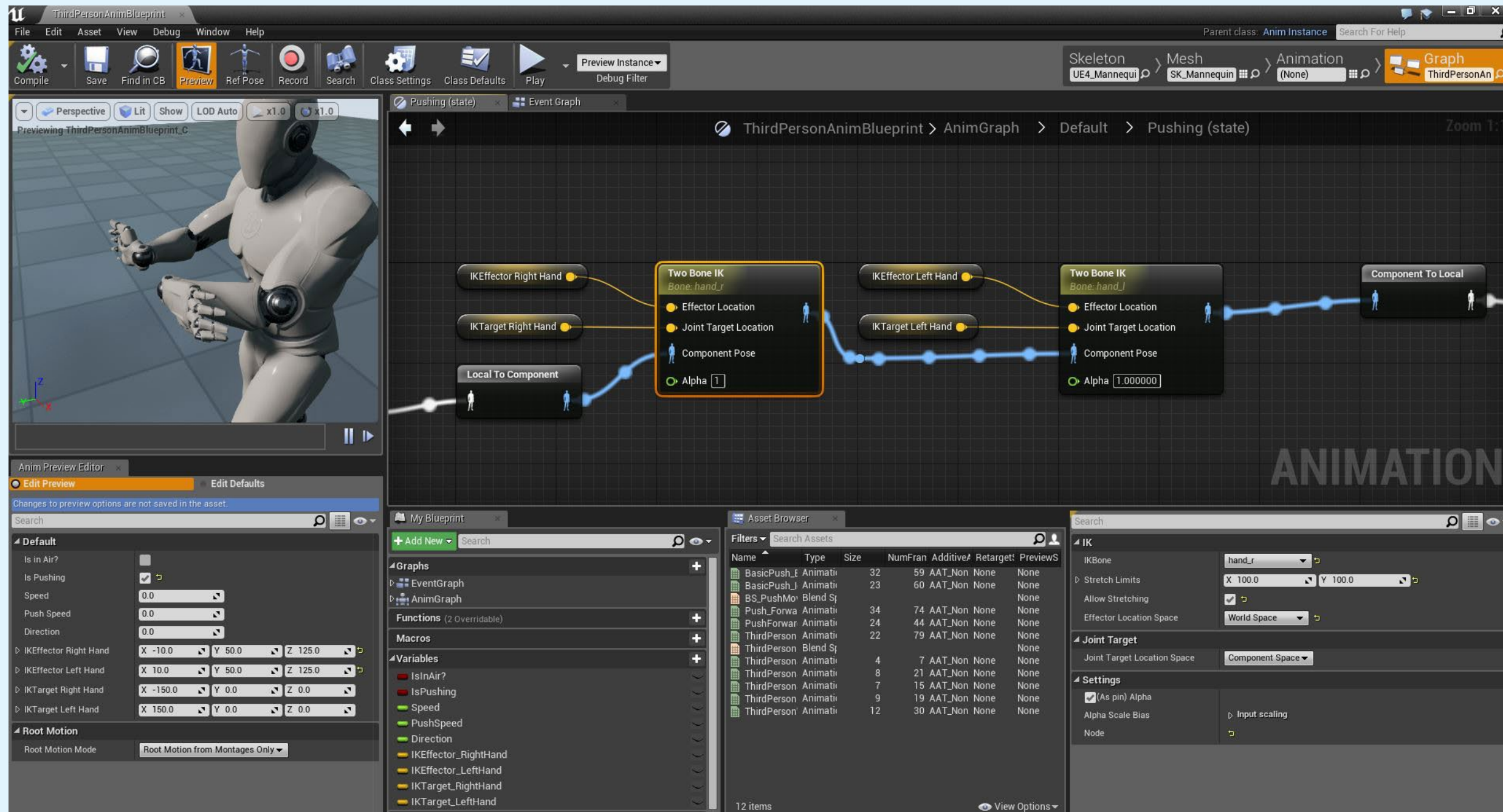
Kevin Lynagh and Ryan Lucas for General Reactives L.L.C.

# Prototyping & Development Tools

Enable more people to build  
'Working & Distributable' artifacts

# Blueprints in Unreal Engine

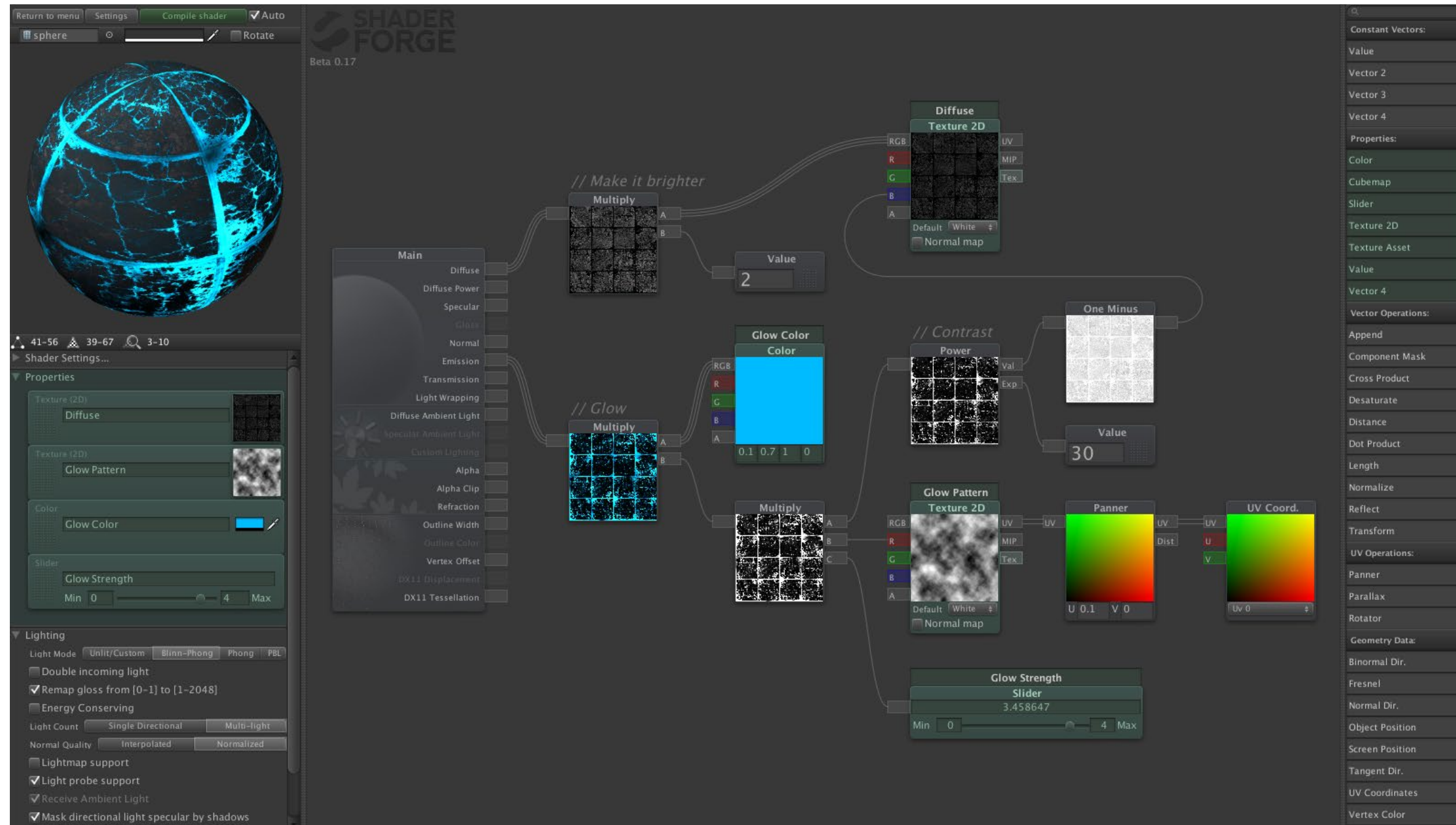
2012-Today



Epic Games

# ShaderForge for Unity

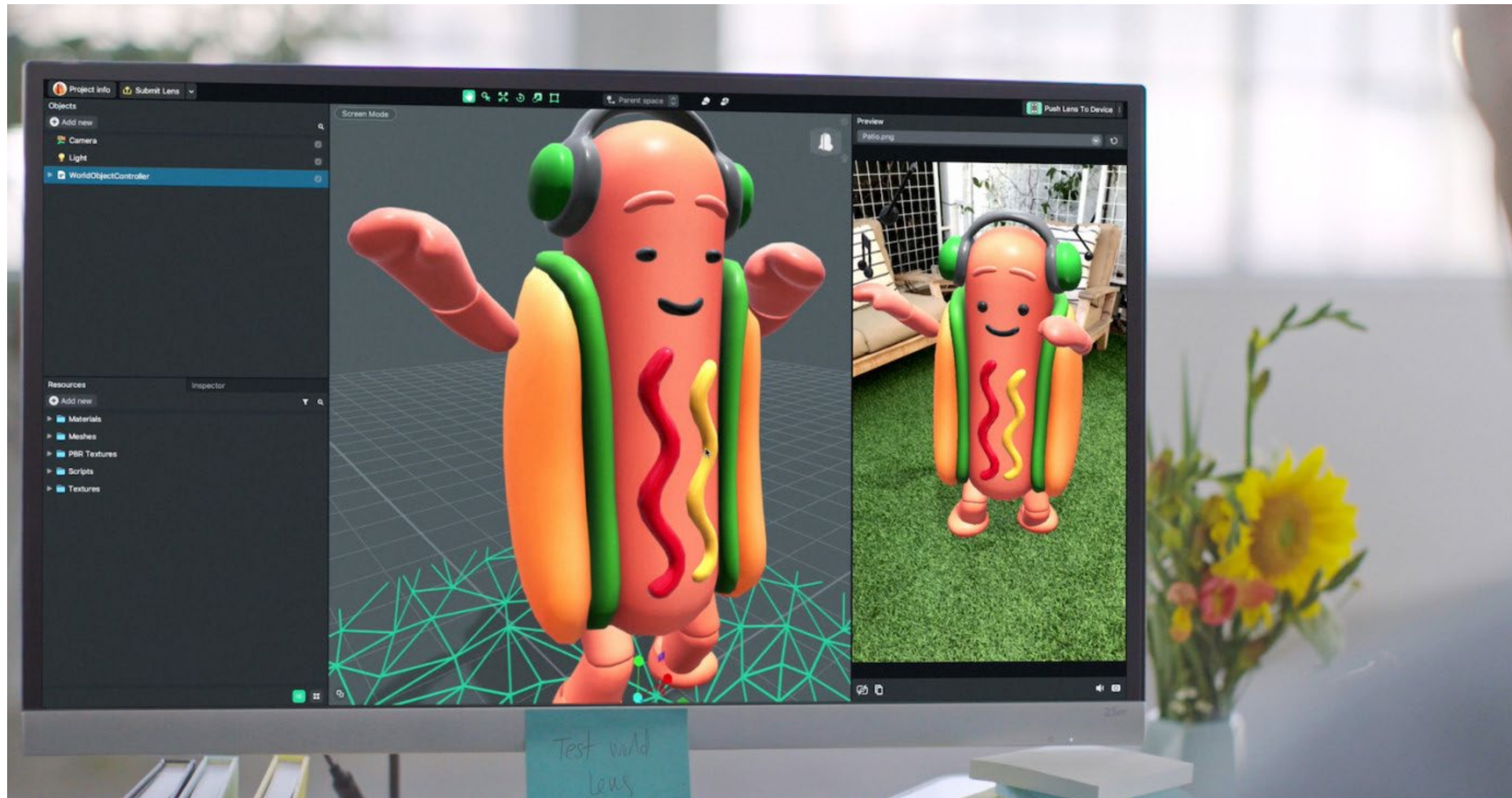
2013-2018



Freya Holmér

# Lens Studio

2017-Today

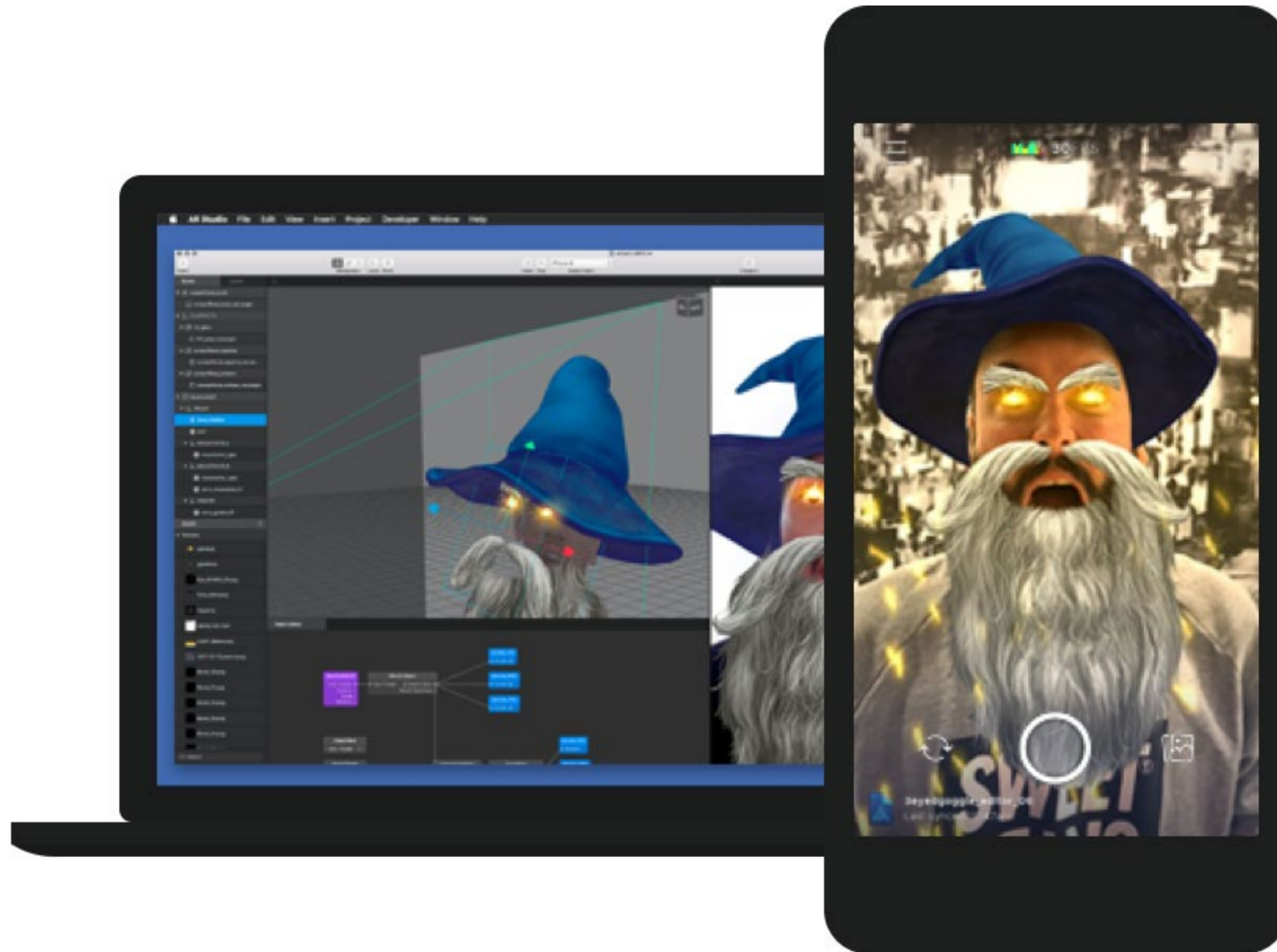


Snap Inc



# AR Studio

# 2017-Today



Facebook

# Lobe AI

# 2018-Today

lobes / Emoji Hands ▾

Accuracy 83.7% Lessons 1,726,078 Time 225:01:20.49 ✓ Learning Complete

The workflow consists of the following steps:

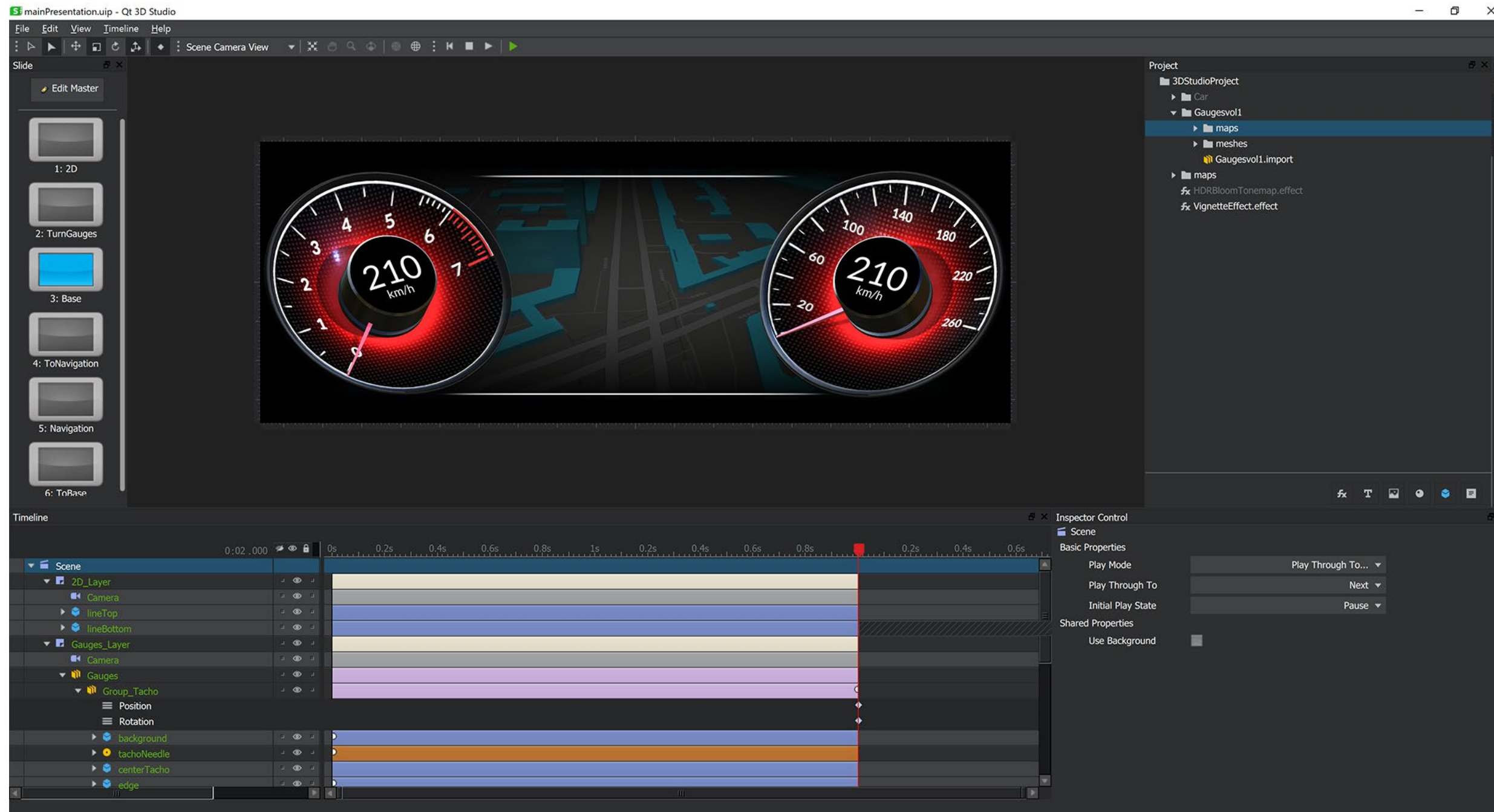
- Image**: Input image of a person making a peace sign.
- Hand & Face**: Detects the hand and face in the image.
- Detect Features**: Extracts features from the image.
- Generate Labels**: Generates labels for the features. The progress bar shows 64% completion for the top label.
- Top Label**: The final label for the image, which is the peace sign emoji.

16,574 Examples Order New to Old Sort by Date Added ▾

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

# Qt 3D Studio

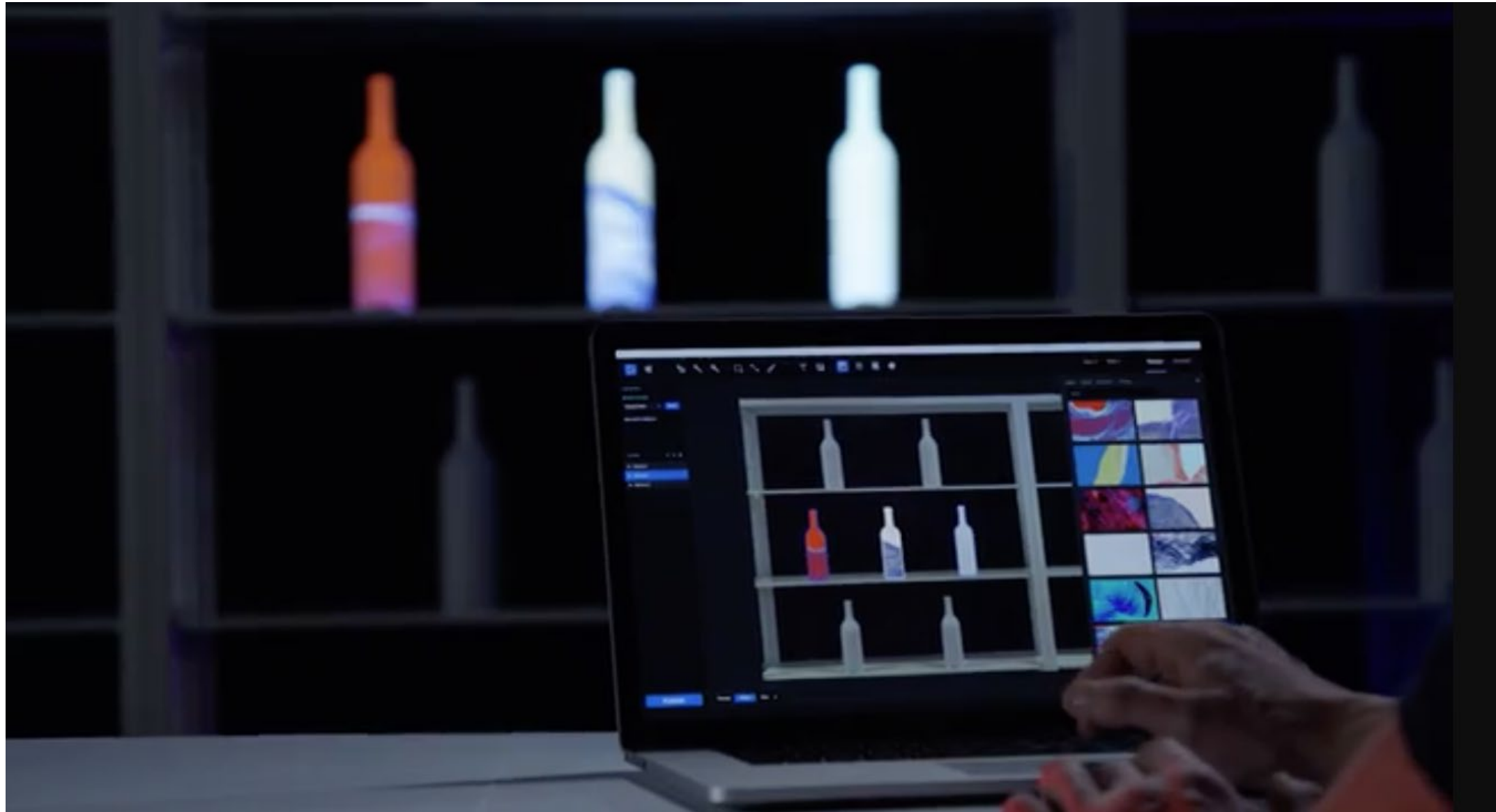
# 2017-Today



The Qt Company

# Lightform

2018-Today



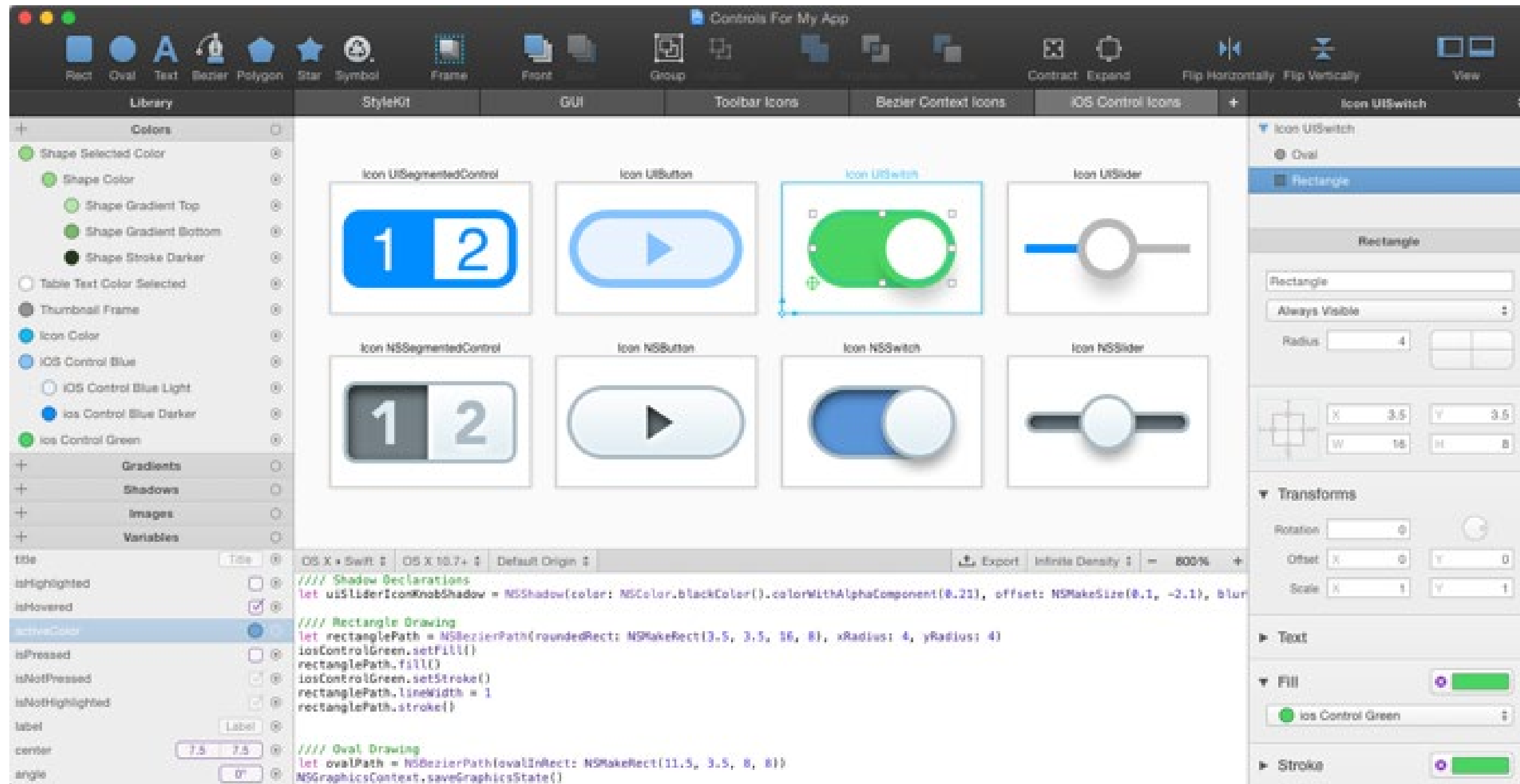
Brett Jones, Kevin Karsch and Rajinder Sodhi

# **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 image shows the PageDraw design tool interface. At the top, there's a toolbar with 'Add', 'Zoom In', and 'Zoom Out' buttons. The main workspace is divided into three sections:

- MainScreen:** A visual design of a 'todos' application. It features a title 'todos' in a light red font, a search input with a dropdown arrow and the text 'What needs to be done?', a list of items (one with a radio button and the text 'Hello world!'), and a footer that says '2 items left'. A tooltip at the bottom says 'Double-click to edit a todo'.
- TodoItem:** A detailed view of a single todo item. It shows two states: 'completed' (with a green checkmark) and 'default' (with a radio button). The text 'Type something' is visible in both states.
- TextInput:** A detailed view of the search input field, showing the text 'What needs to be done?'.

On the right side, there's a 'Component' panel with tabs for 'Draw', 'Code', and 'Component'. The 'Component' tab is active, showing the following details for the selected 'TodoItem' component:

- Block Type: TodoItem
- Name: TodoItem Instance
- X: 37, Y: 211, W: 491, H: 58
- Props: Content (Hello world!), State (default)
- Buttons: EXPORT PARAMS AS JSON
- Options: Flexible Left Margin, Flexible Right Margin, Center Horizontally, Flexible Width, Flexible Top Margin, Flexible Bottom Margin (all unchecked)

At the bottom, there's a code editor showing the React code for the application:

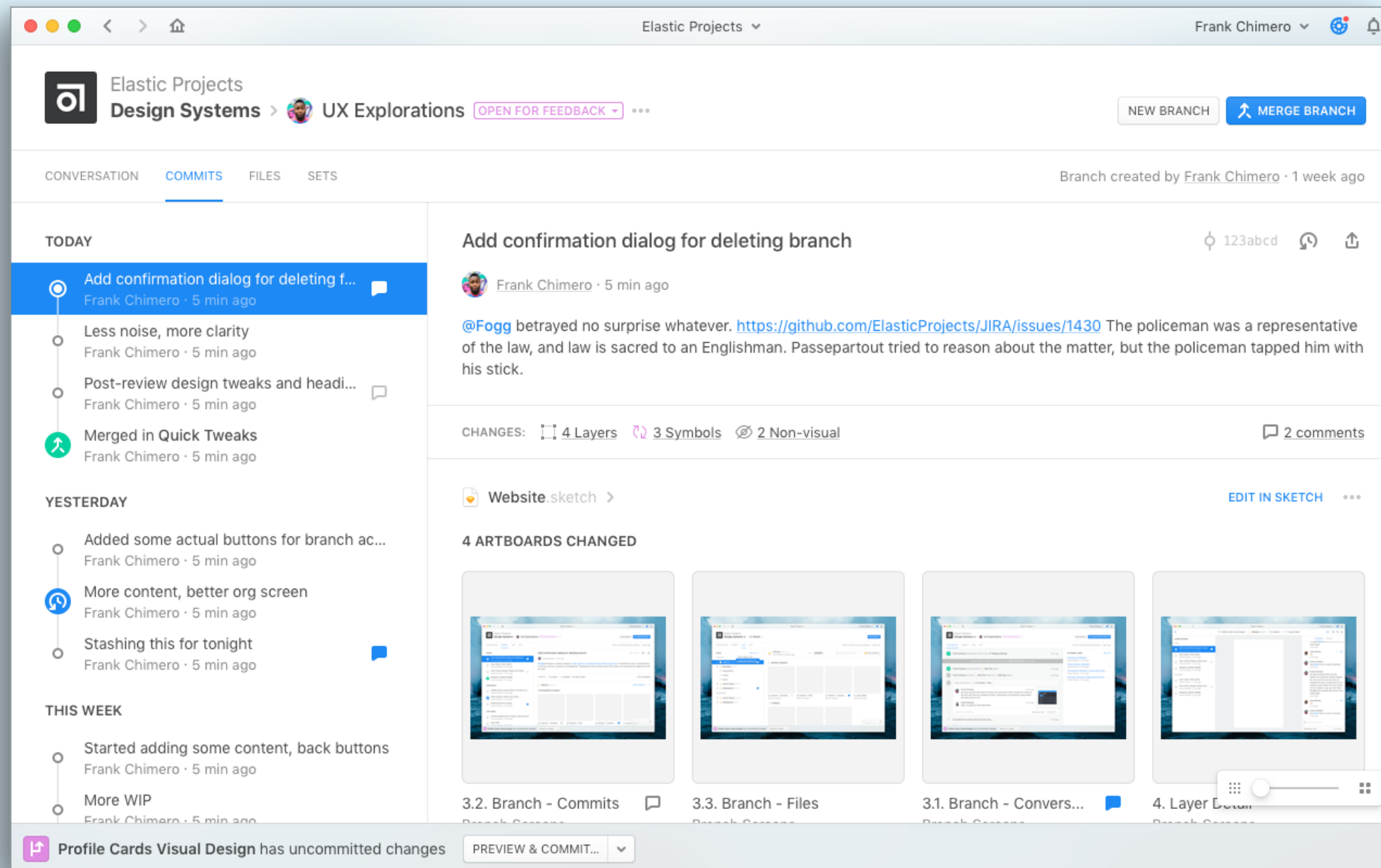
```
1 import React, { Component } from 'react';
2 import { render } from 'react-dom';
3
4 /* import the MainScreen component drawn above in Pagedraw */
5 import MainScreen from './src/pagedraw/mainscreen'
6
7 class App extends Component {
8   constructor() {
9     super();
10    this.state = {
11      todos: [{content: "Meet with Yoda"}, {content: "Defeat Darth Vader"}]
12    };
13  }
14
15  render() {
16    const todos = this.state.todos.map((todo, i) => {
17      return {...todo, toggle: (() => this.toggleTodo(i)),
18        delete: (() => this.deleteTodo(i))
19    });
20    return (
21      <MainScreen
22        list={todos}
23        itemsLeft={this.state.todos.filter((elem) => !elem.completed).length}
24        addTodo={this.addTodo}
25      />
26    );
27  }
28 }
```

Below the code editor is a live preview of the application running in a browser at <https://2yffz.run.stackblitz.io>. The preview shows the 'todos' app with the search input, a list of items ('Meet with Yoda' and 'Defeat Darth Vader'), and a footer that says '2 items left'. A blue chat bubble icon is visible in the bottom right corner of the preview.

Jared Pochtar

# Abstract

# 2016-Today

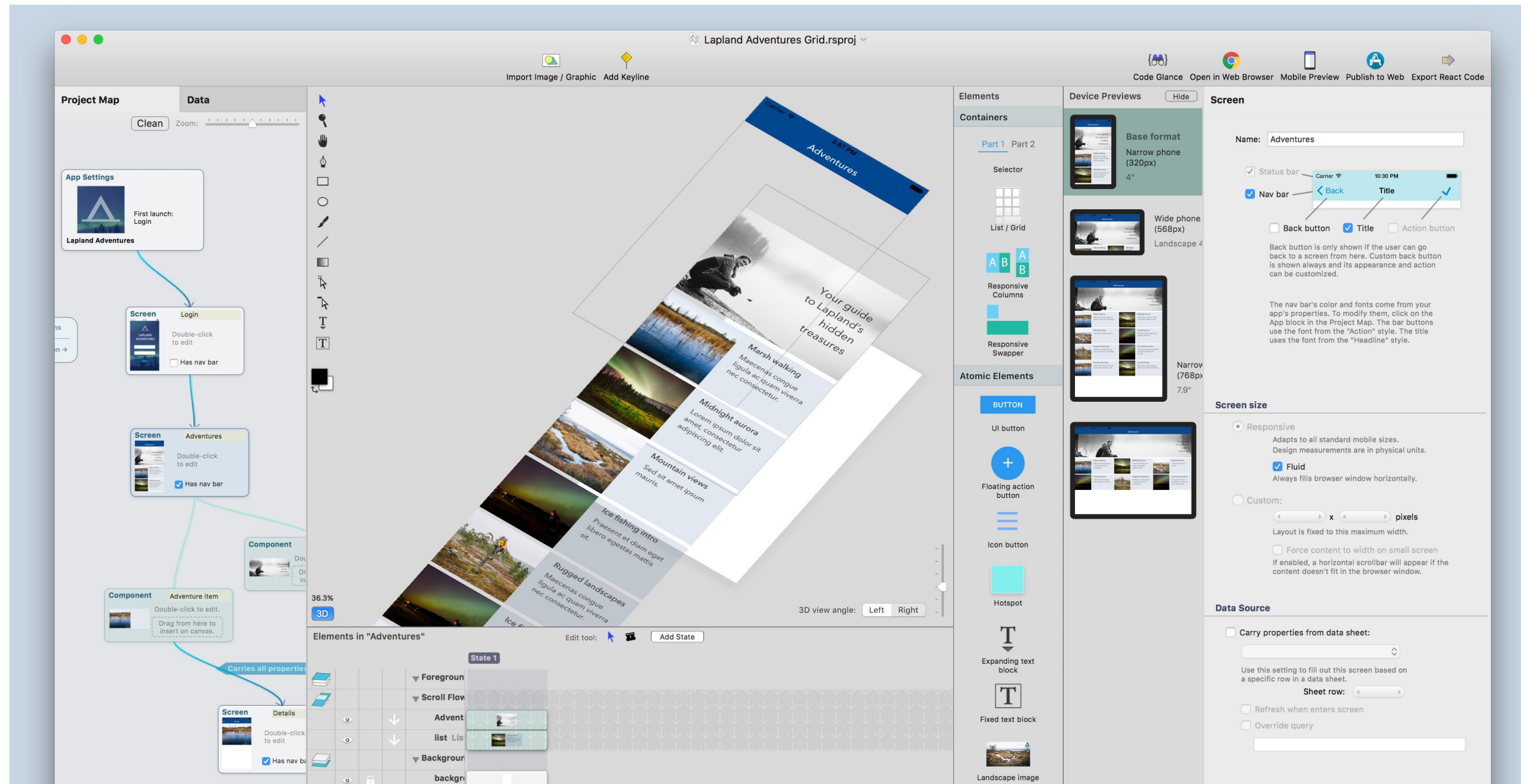


## Elastic Projects



# React Studio

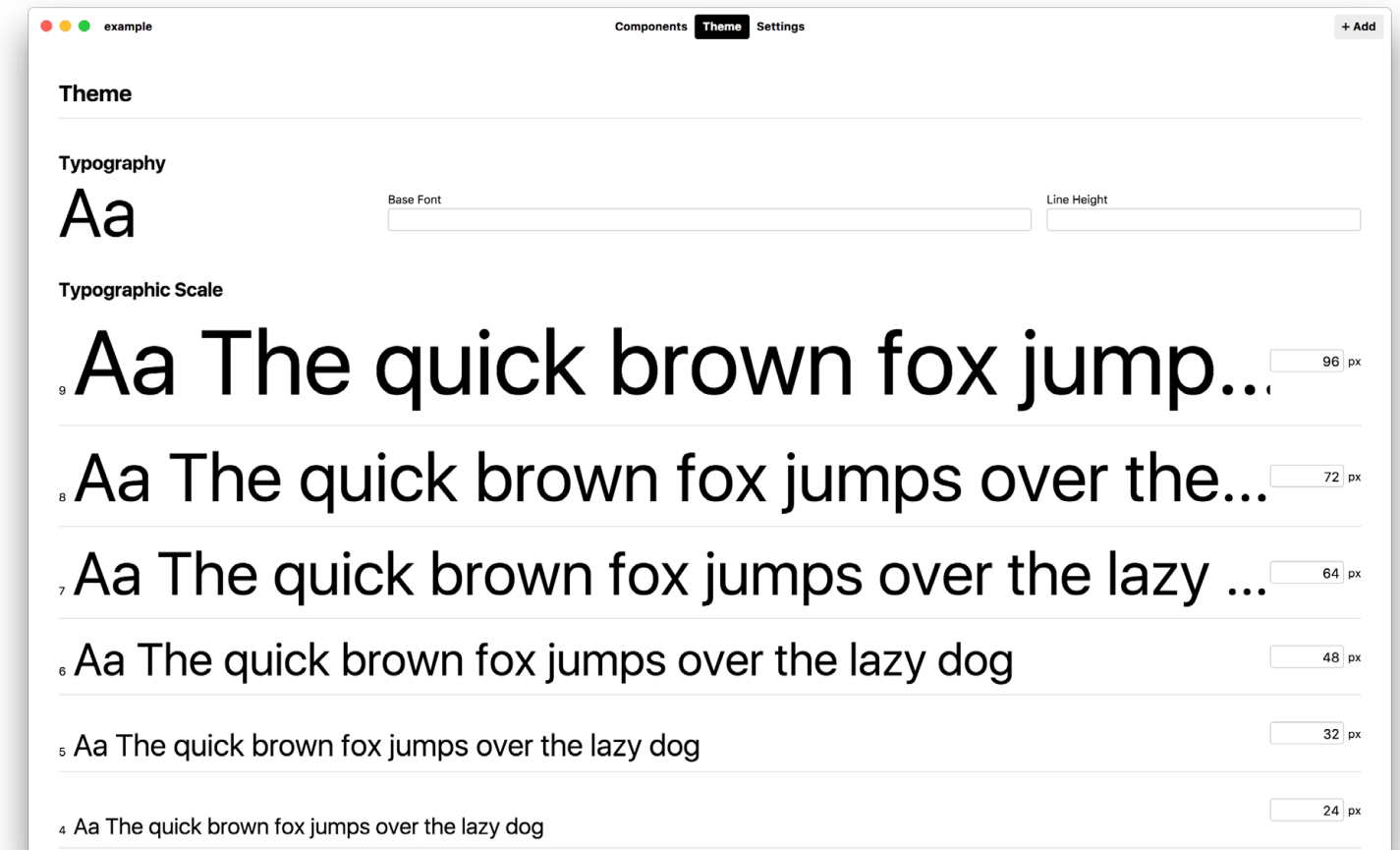
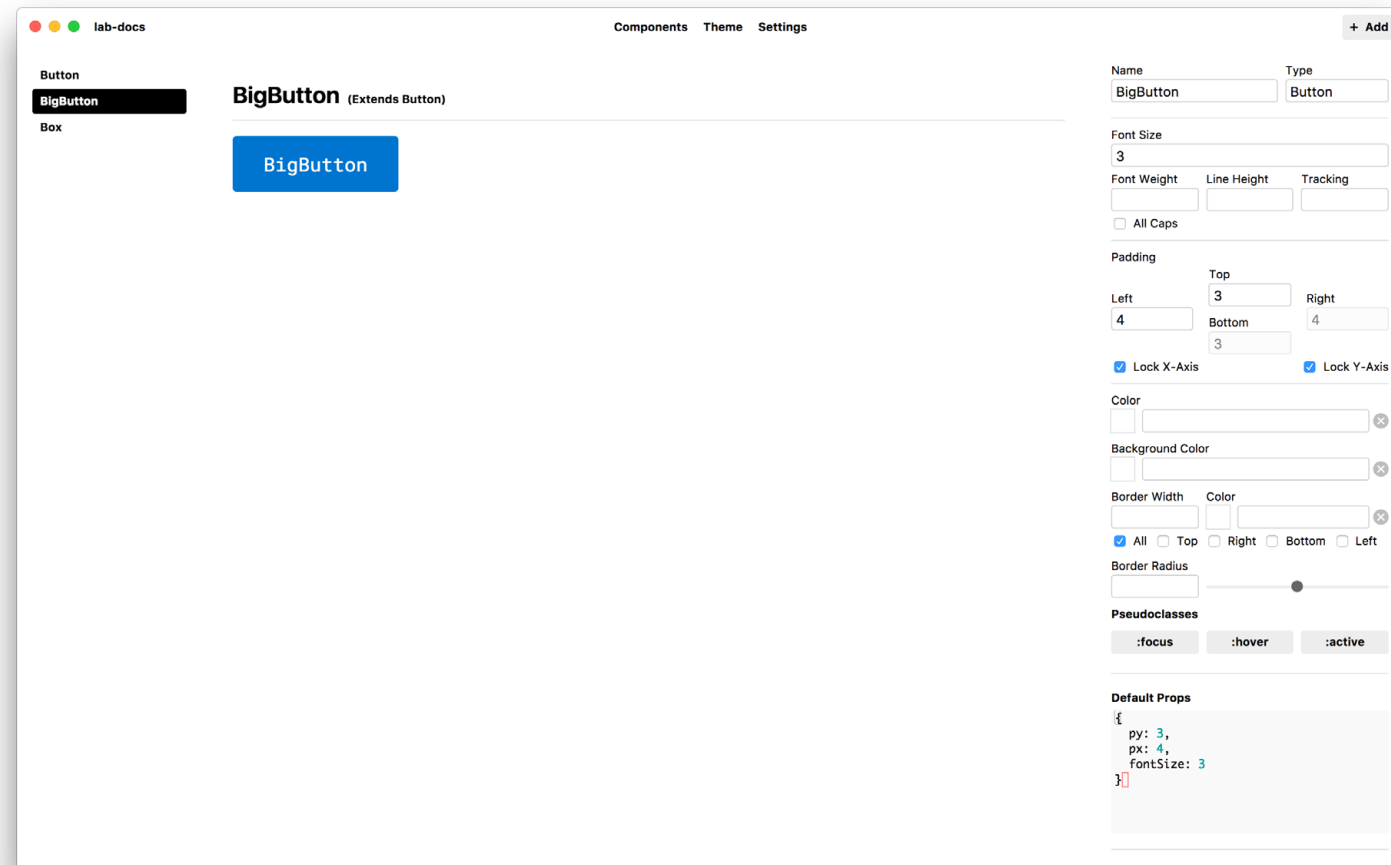
# 2017-Today

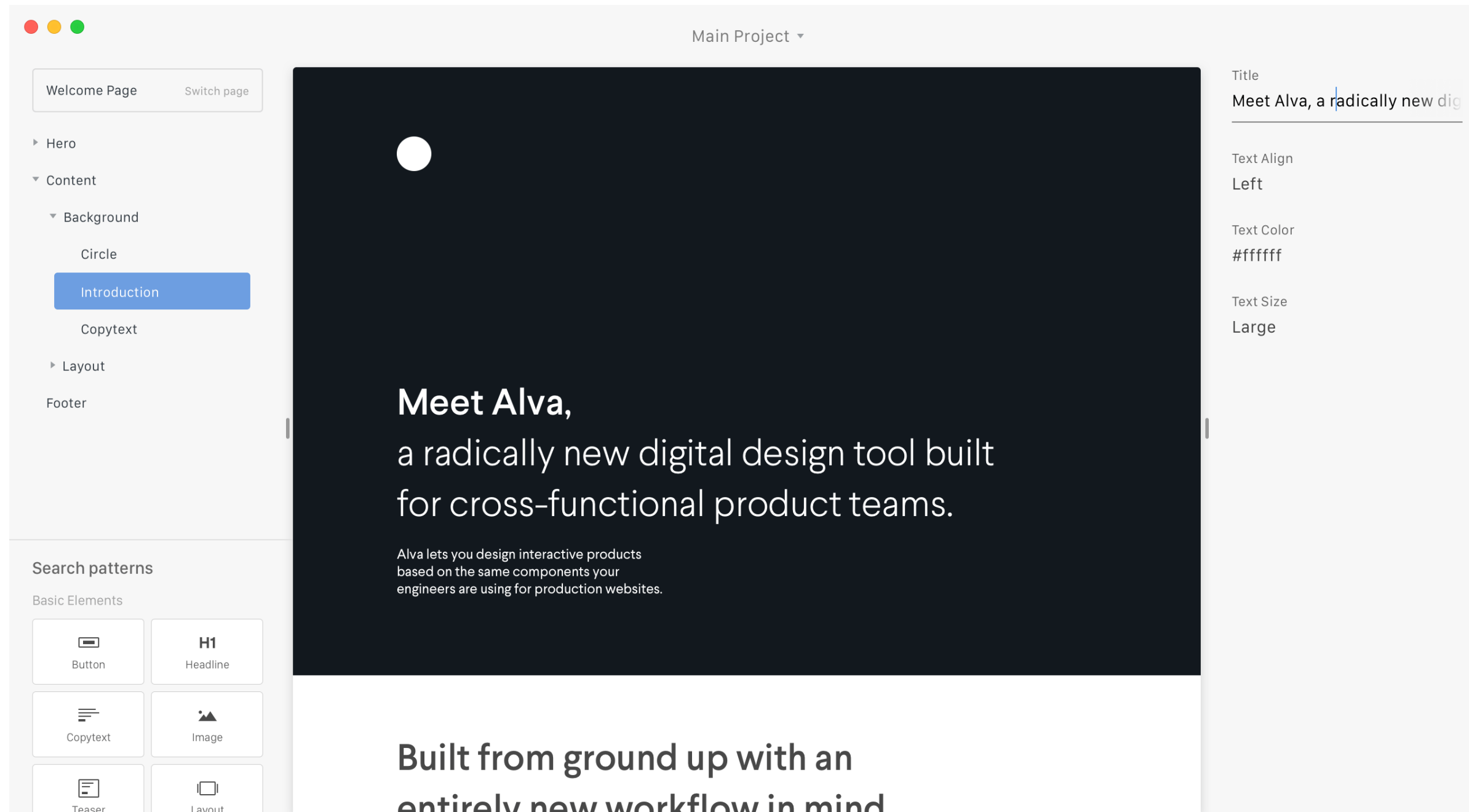


Neonto Ltd

# Compositor Lab

# 2017-Today





SinnerSchrader Deutschland GmbH (Part of Accenture Interactive)

# 'Low Code' Environments

Enable the Development of Apps  
using modular Building-Blocks

# OutSystems Platform

2001-Today

The screenshot displays the OutSystems development environment. The main workspace shows a workflow diagram for a process named 'BootstrapContacts'. The workflow starts with a 'Start' node, followed by 'GetContacts', 'ConvertFromExcel', and a 'For Each' loop. Inside the loop, there is an 'Assign ContactRecord' node and a 'CreateContact' node. The 'For Each' loop has a 'Cycle' arrow pointing to the 'Assign ContactRecord' node and an arrow pointing to the 'CreateContact' node. The 'End' node is at the bottom of the loop.

The right-hand pane shows the 'SalesAssistant' project structure. Under 'Actions', there is a folder 'BootstrapContacts' containing a 'ContactRecord' folder. Below that is a '(System)' folder with various actions like 'AbortTransaction', 'Audit', 'CommitTransaction', 'ListAppend', 'ListAppendAll', 'ListClear', 'ListDuplicate', 'ListInsert', 'ListRemove', 'NotifyWidget', 'NotifyWidgetGetMessage', and 'SetCurrentLocale'. Other folders include 'RichWidgets', 'Users', 'Web Services', and 'References'.

Below the project structure is a table for the 'BootstrapContacts' action:

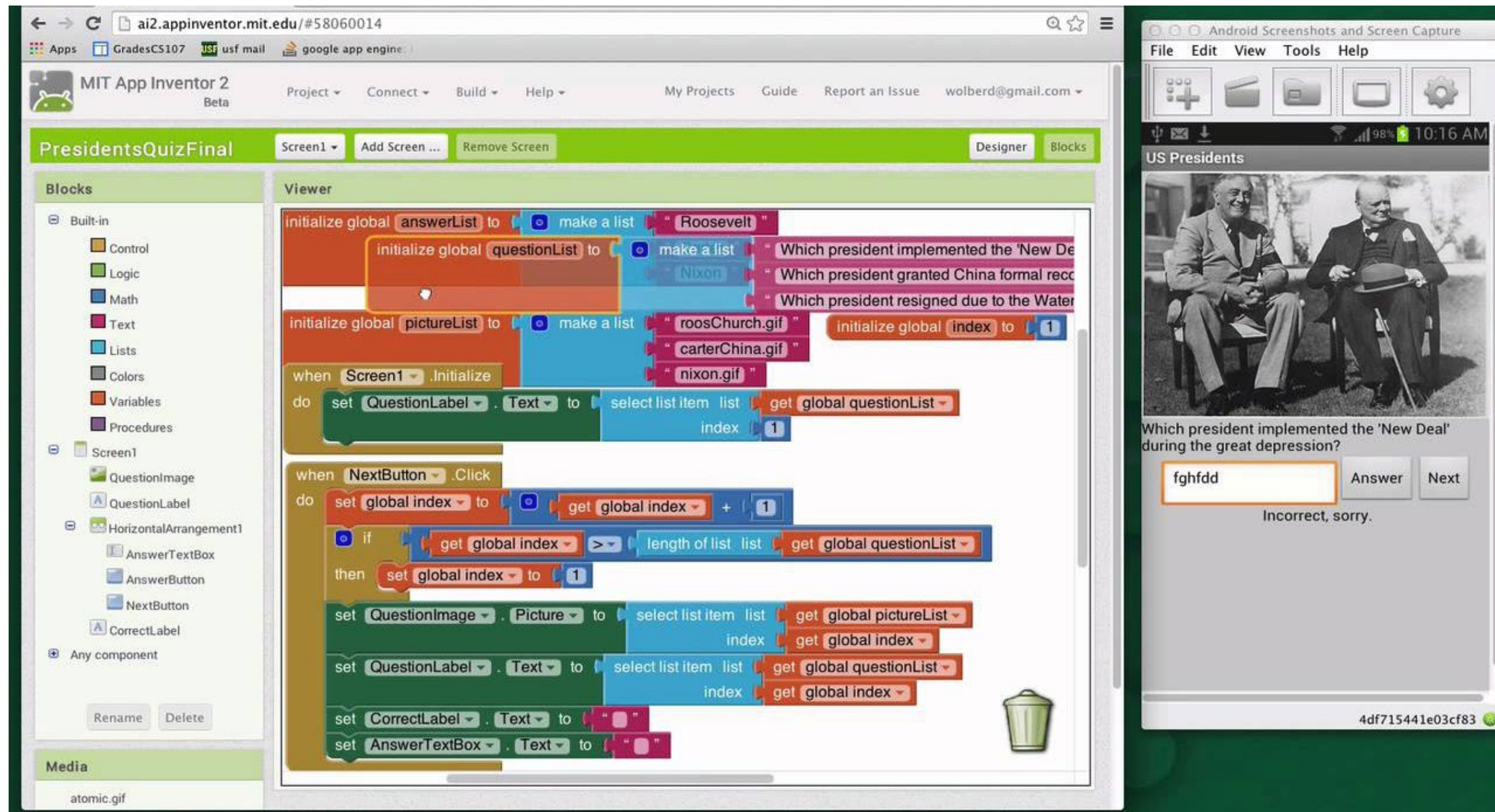
BootstrapContacts	
Name	BootstrapContacts
Description	Populates the database with the contacts f ...
Public	No
Function	No
Cache in Minutes	
<i>last modified by admin at 01:49</i>	

The bottom status bar shows 'SalesAssistant.oml saved at 01:49 | admin@dev.acme.net'. There are also buttons for 'TrueChange™', 'Debugger', and '1-Click Publish'.

OutSystems Inc

# App Inventor for Android

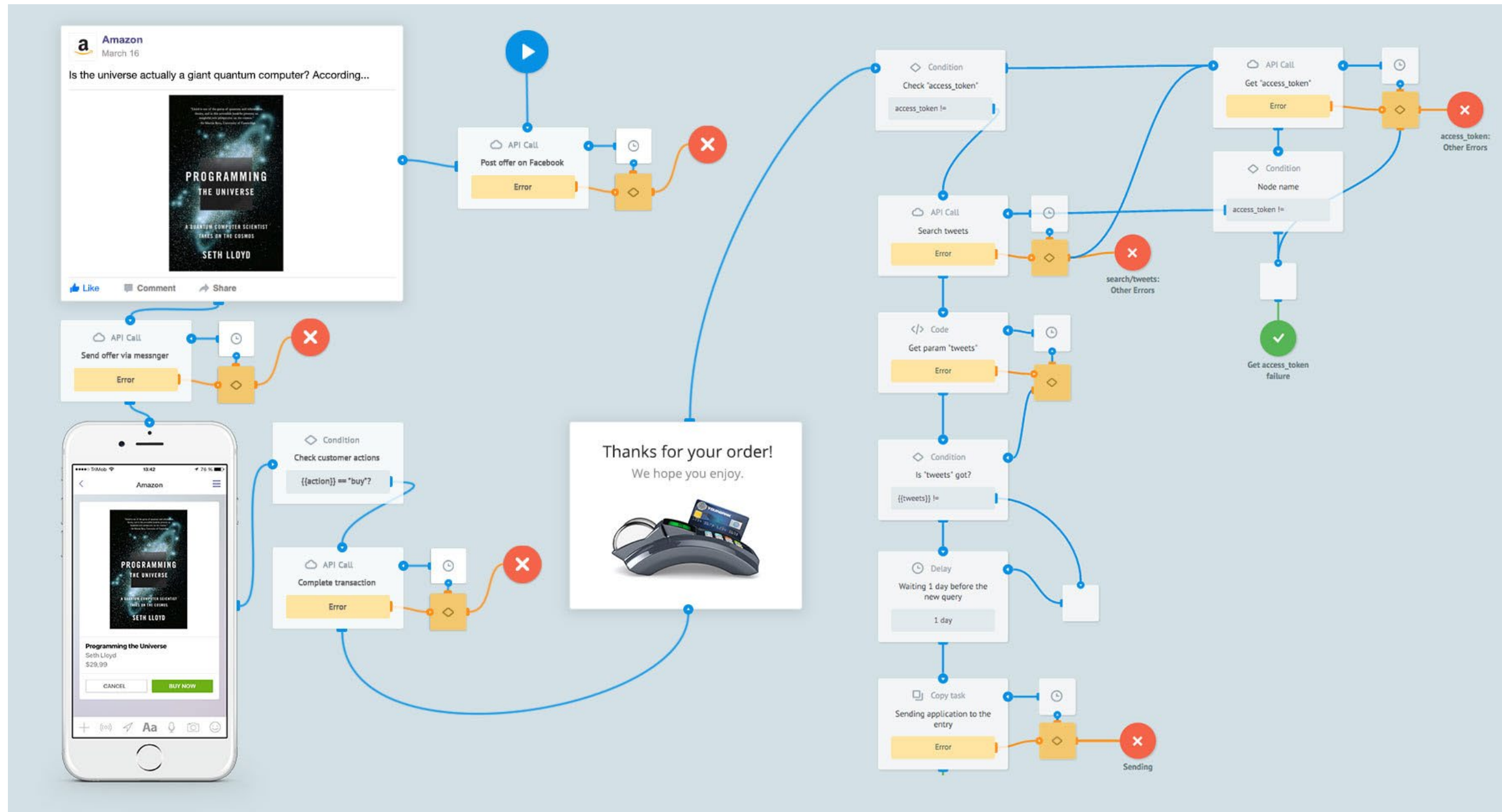
2010-Today



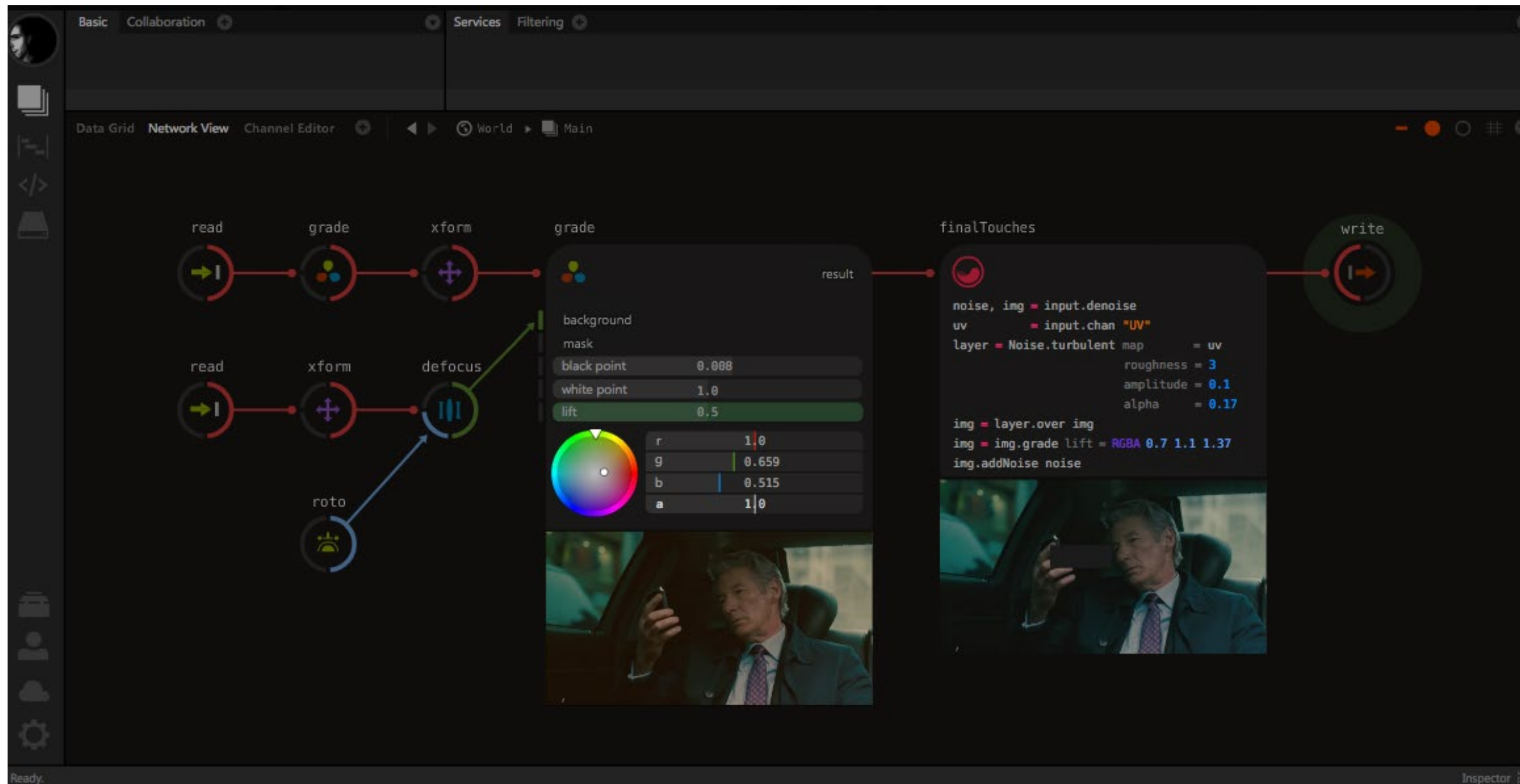
Google and MIT Computer Science and Artificial Intelligence Lab

# Corezoid Process Modeler

2013-Today



Corezoid.com Middleware Inc

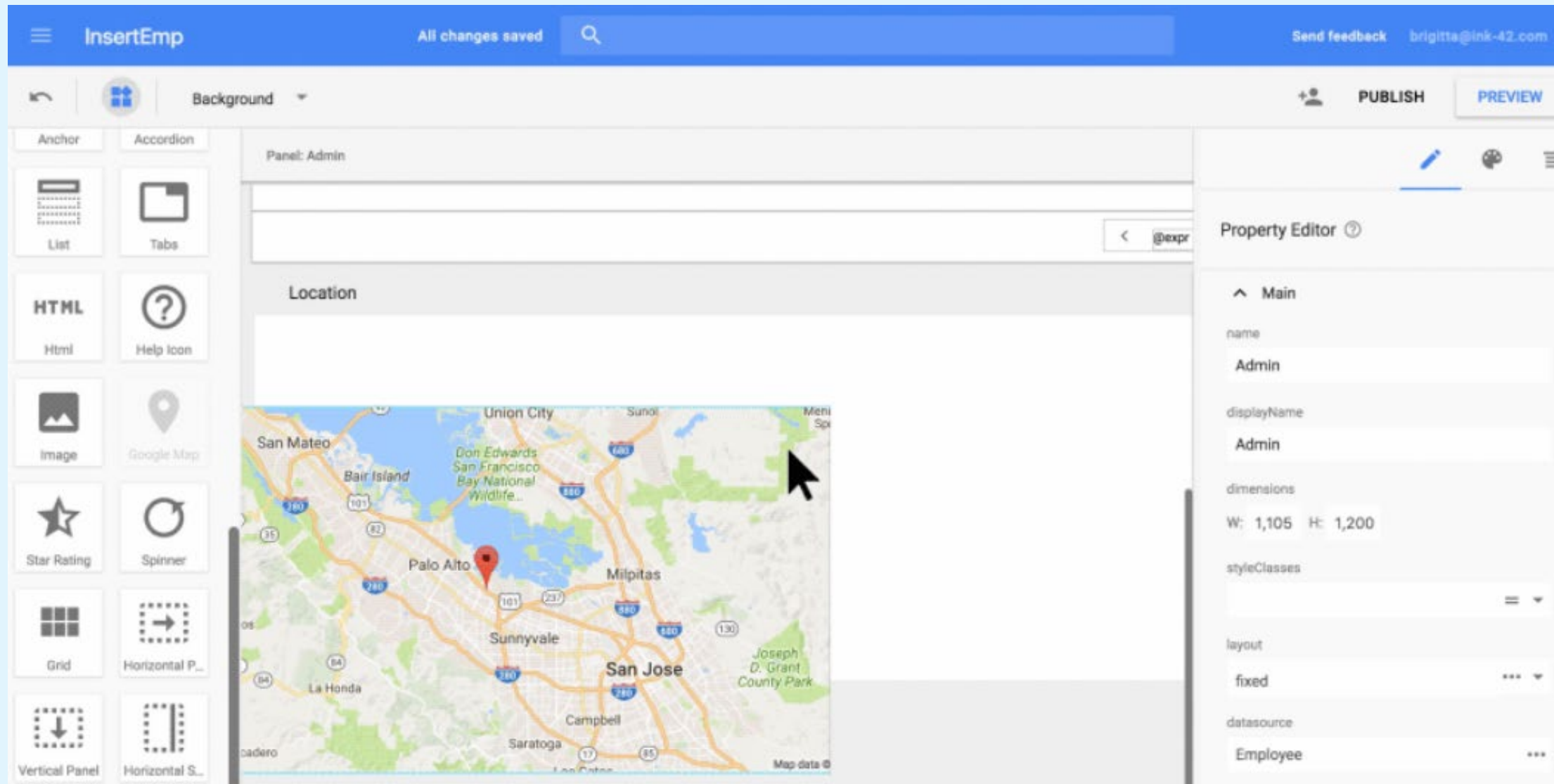


Wojciech Daniło, Marcin Kostrzewa



# AppMaker

# 2018-Today



- 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

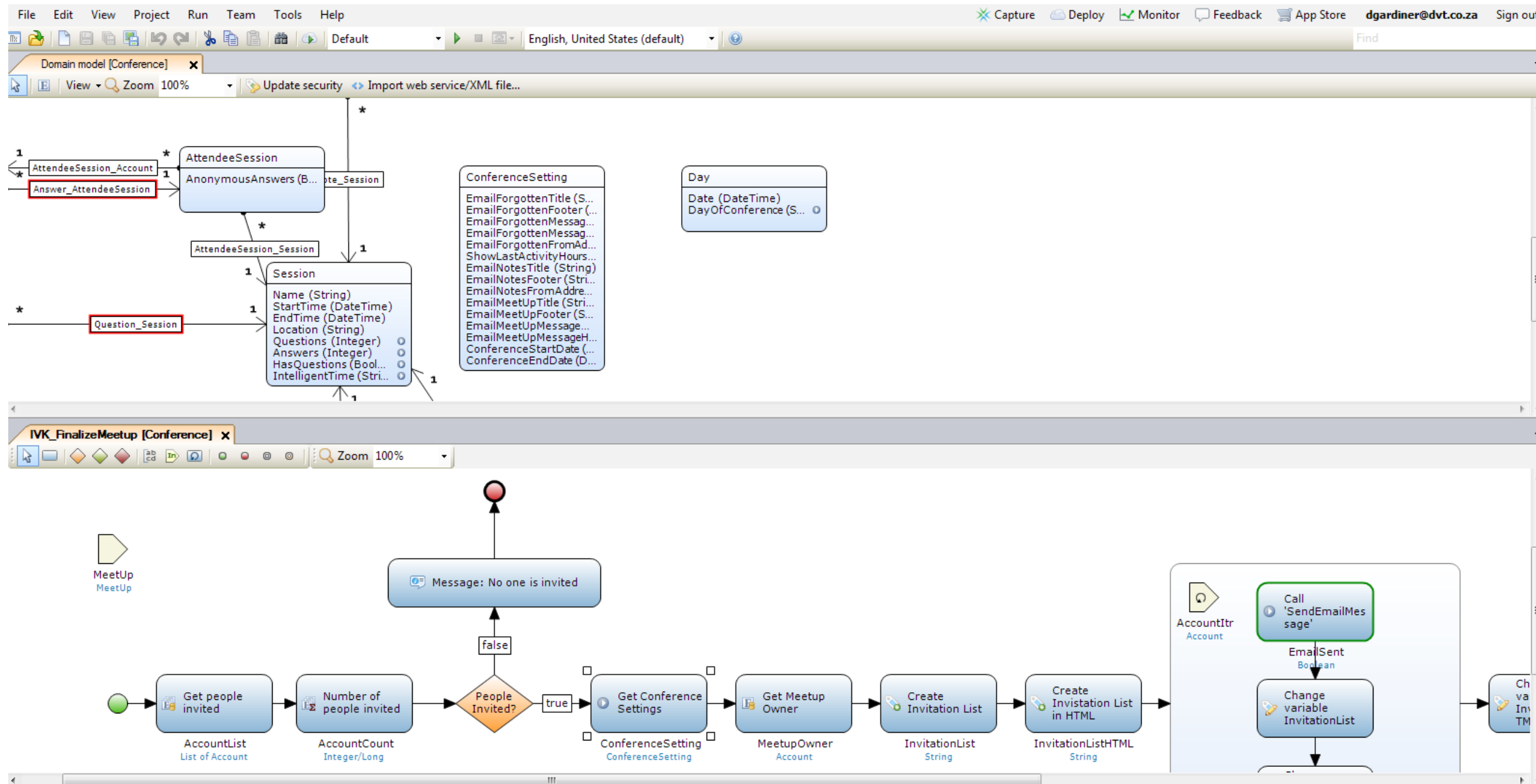
Google

# Dynamic Modeling Tools

Support System Visualizing

# Mendix Business Modeler

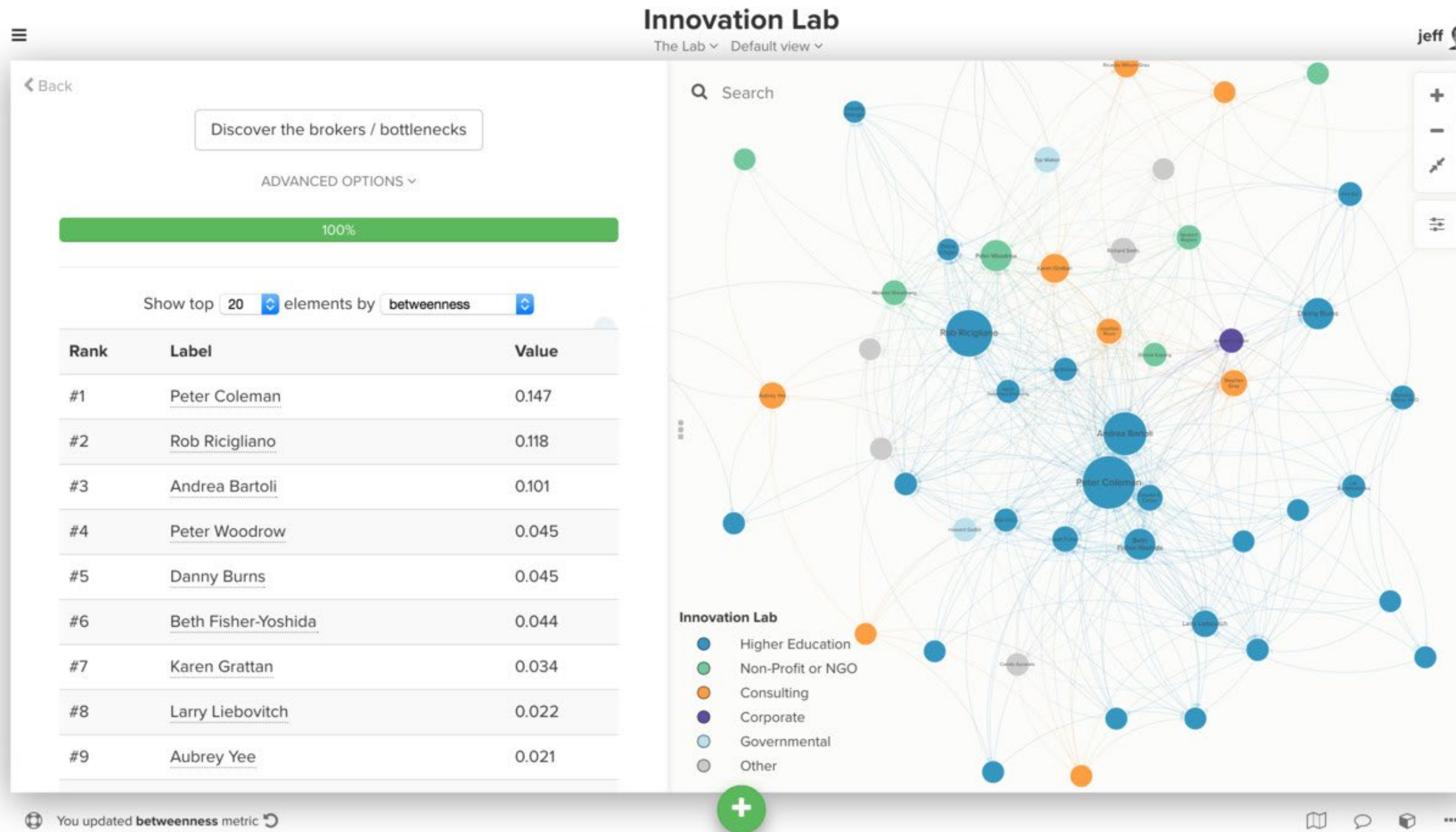
# 2005-Today



Mendix

# Kumu

# 2011-Today



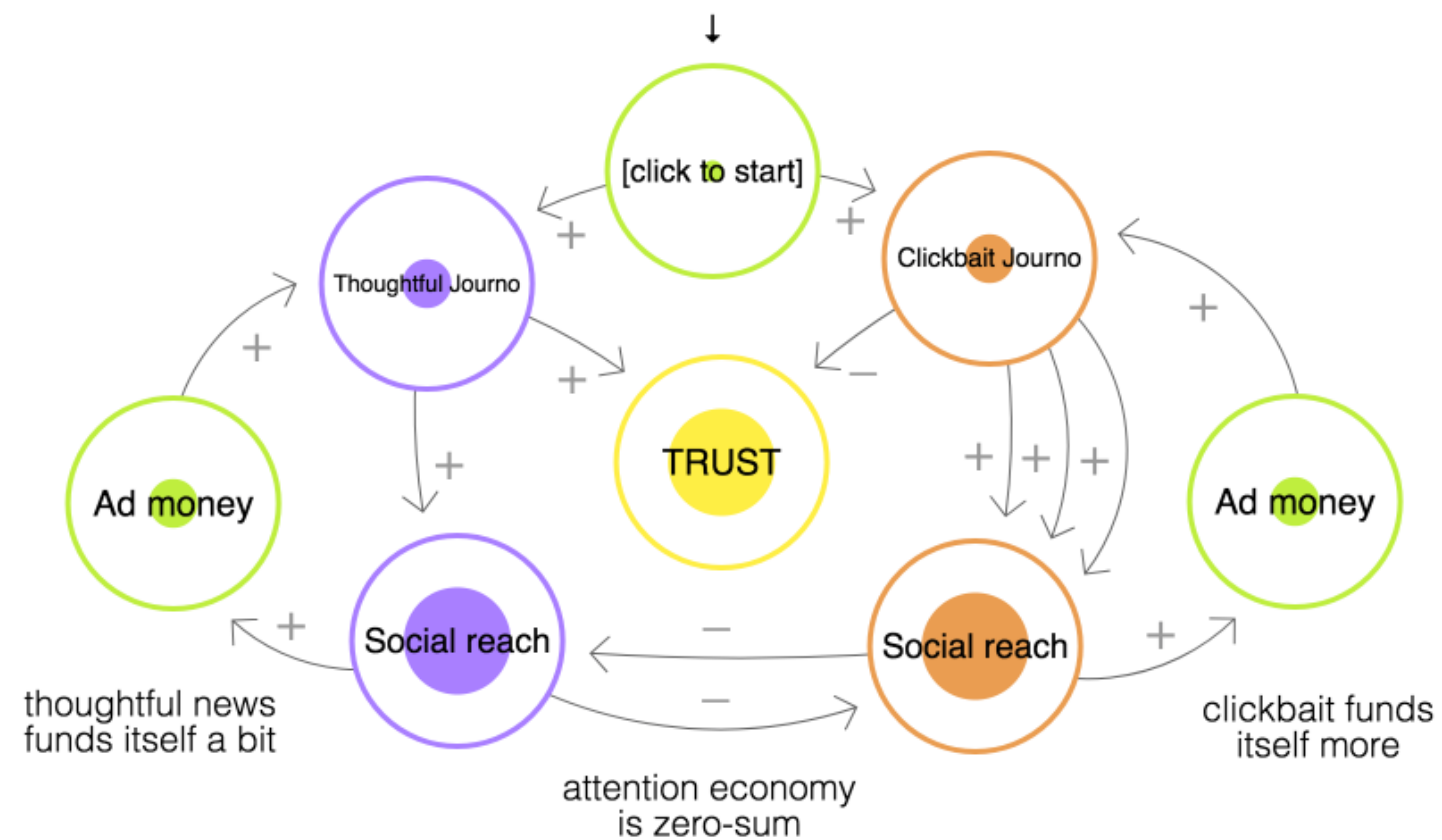
Jeff and Ryan Mohr for Kumu Inc

# 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.



Nicky Case

# IDE: Environments & Experiments

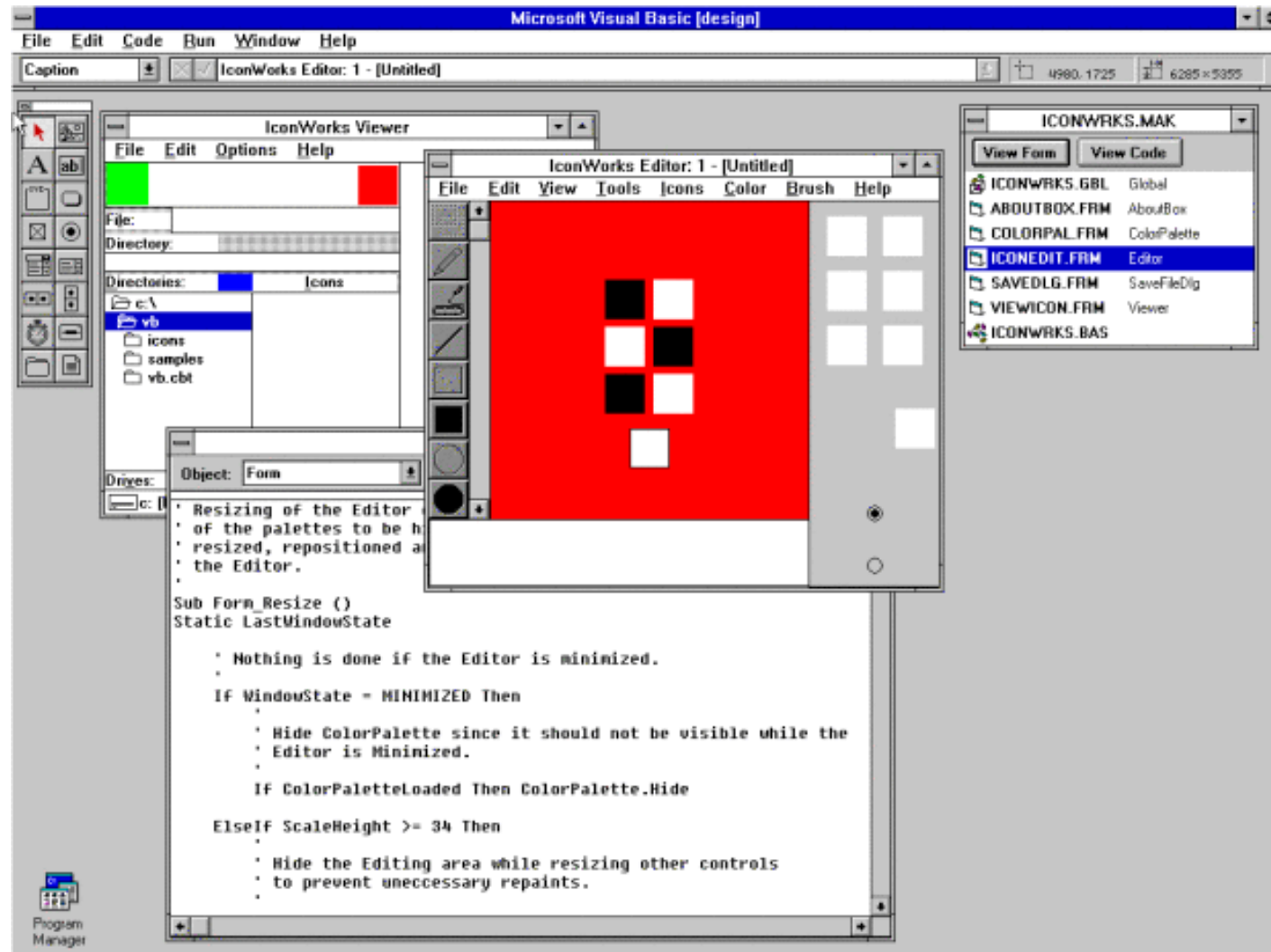
## What Is & What's Next

# **Traditional IDEs**

## Integrated Development Environments

# Visual Basic

1991-1998

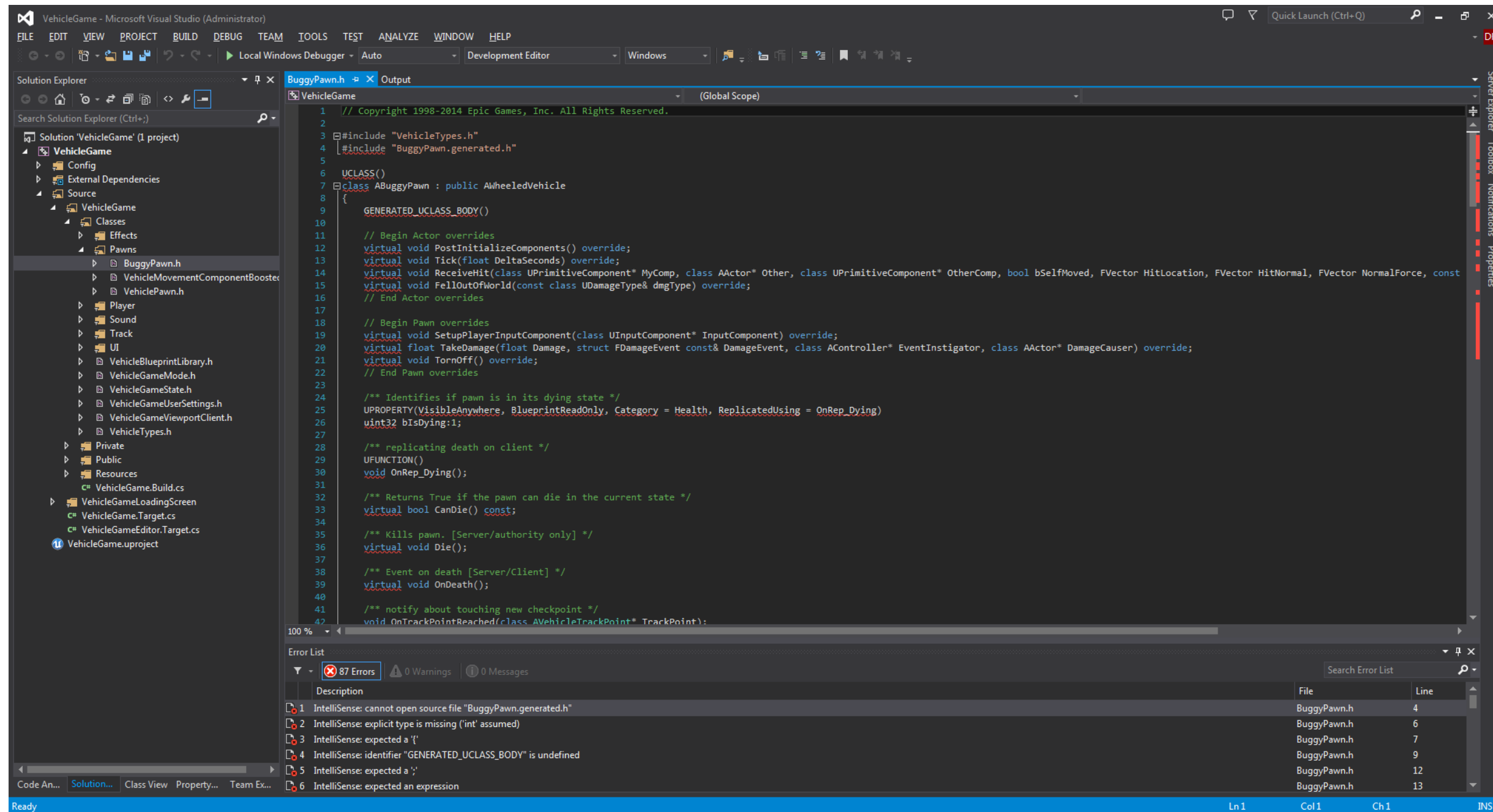


Cooper Software for Microsoft



# Visual Studio

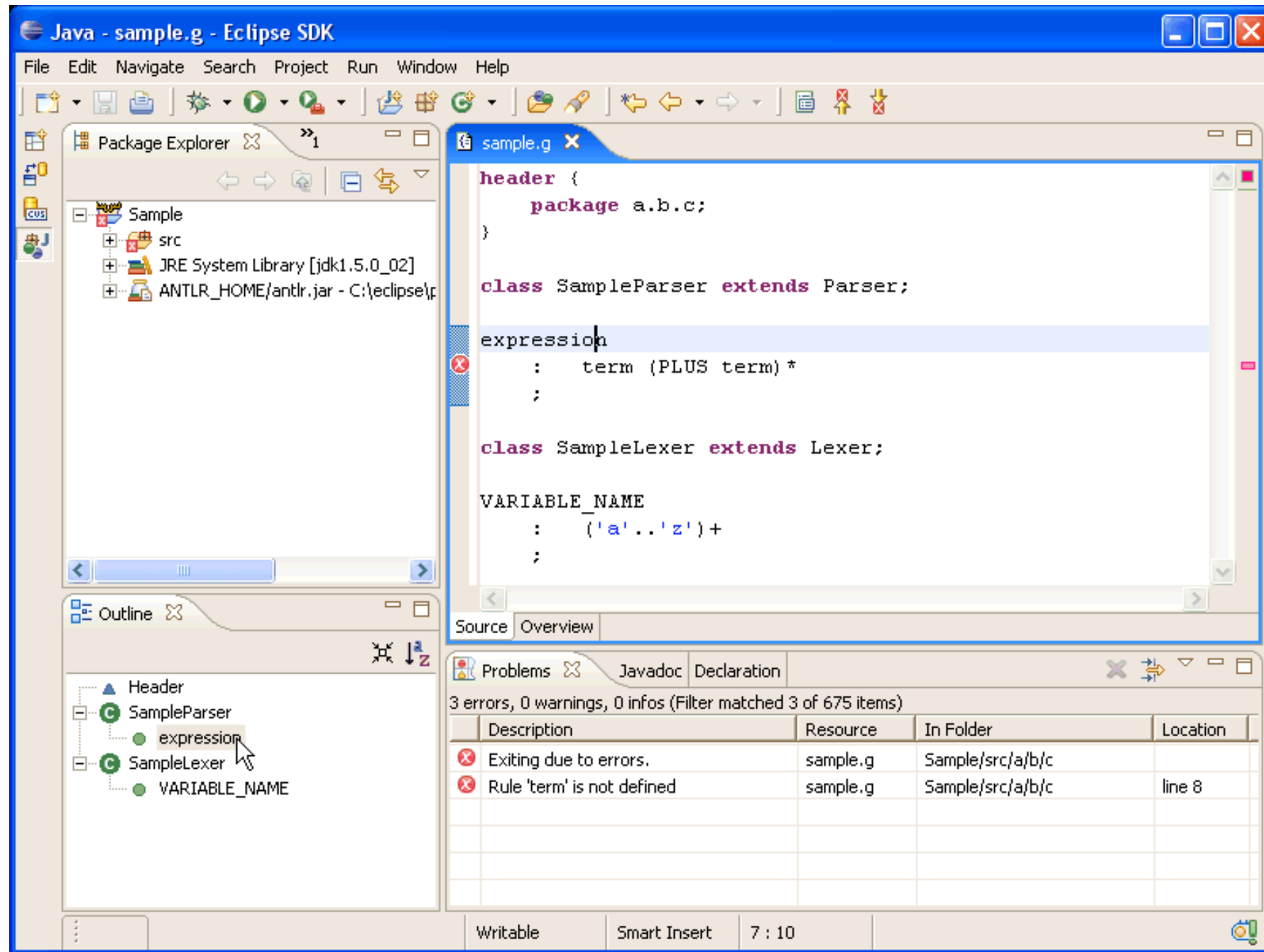
# 1997-Today



Microsoft

# Eclipse

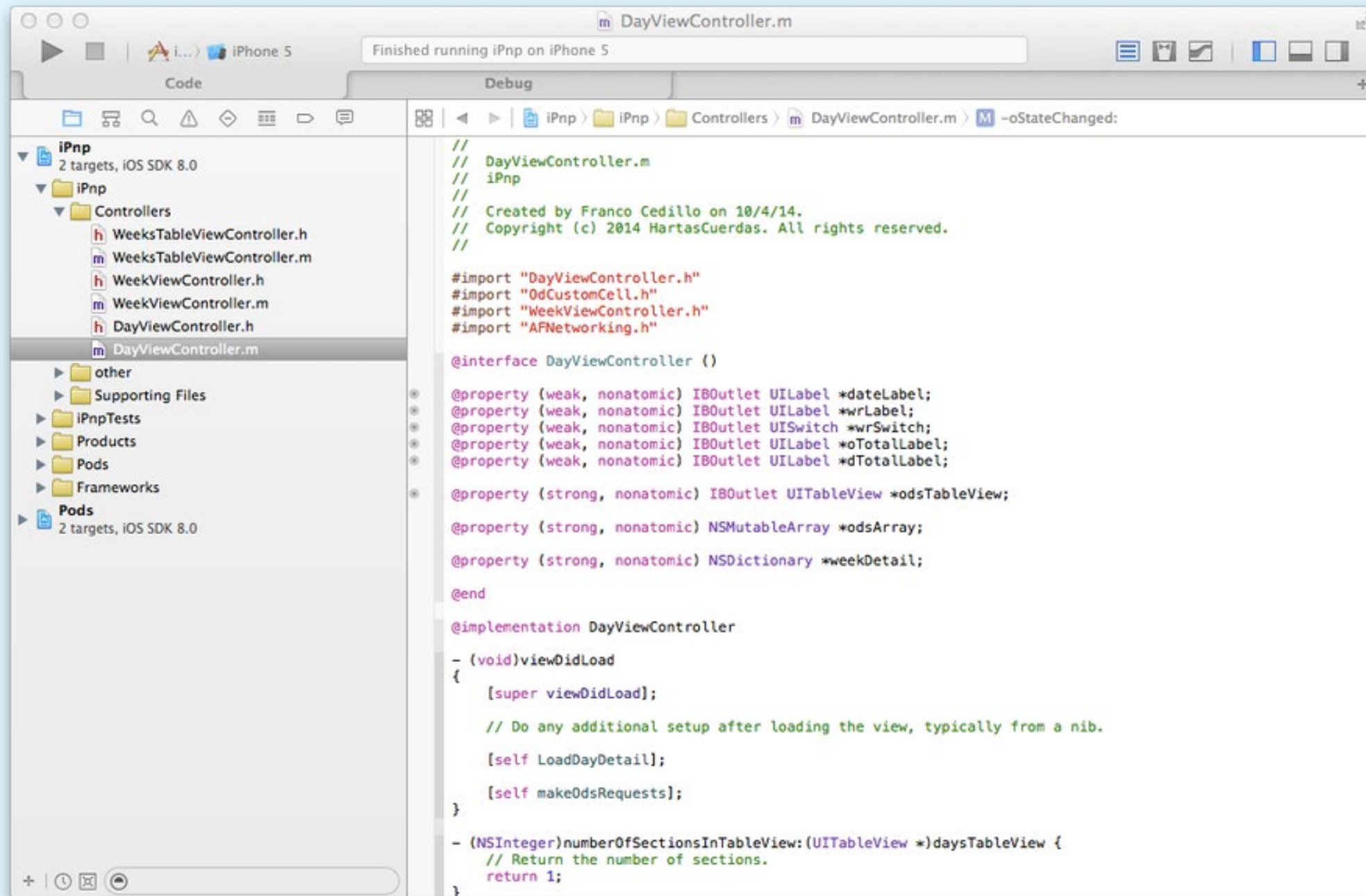
# 2001-Today



Eclipse Foundation

# Xcode

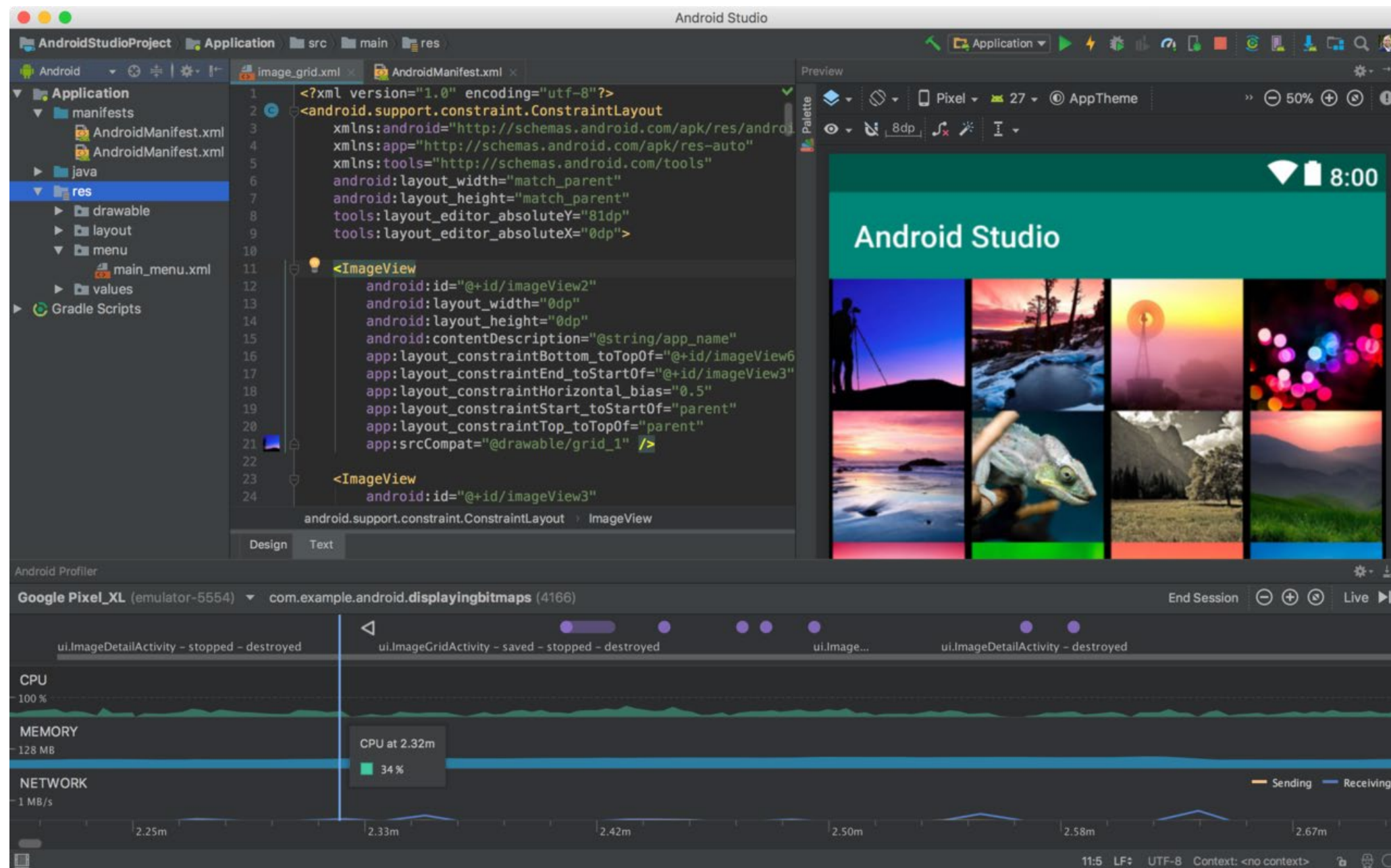
# 2003-Today



Apple

# Android Studio

2013-Today



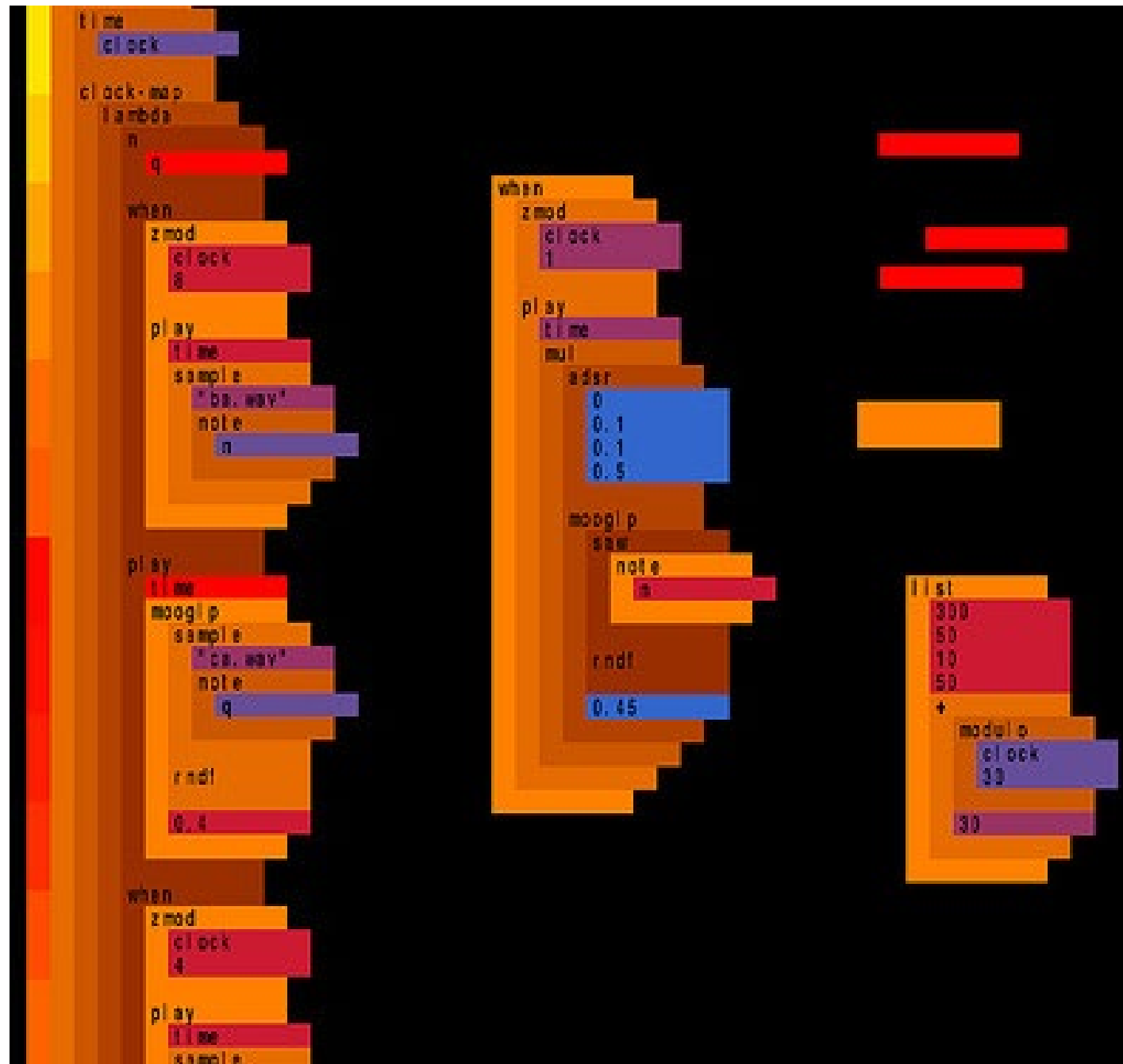
Google

# Structure Editing

## At Nested Levels of Abstraction

# Scheme Bricks

2008



Dave Griffiths

# Code Bubbles for Eclipse

2010

**Thread Viewer**

```
edu>brown>cs>bubbles>bddt>BddtThreadView$TableUpdater>run(...)
@Override public void run() {
    BumpThread bt;
    switch (run_event.getEventType()) {
        case THREAD_ADD :
            bt = run_event.getThread();
            if (bt != null) {
                thread_set.add(bt);
                synchronized (bump_threads) {
                    bump_threads.clear();
                    bump_threads.addAll(thread_set);
                }
            }
            break;
        case THREAD_REMOVE :
            bt = run_event.getThread();
            if (bt != null) {
                thread_set.remove(bt);
                synchronized (bump_threads) {
                    bump_threads.remove(bt);
                }
            }
            break;
        case THREAD_CHANGE :
            //TODO: Update thread
            break;
    }
    threads_model.fireTableDataChanged();
}
```

**Thread Model**

```
... .. BddtThreadView$ThreadsModel>getValueAt(...)
@Override public Object getValueAt(int r, int c) {
    BumpThread bt;
    synchronized (bump_threads) {
        if (r < 0 || r >= bump_threads.size()) return null;
        bt = bump_threads.get(r);
    }
    switch (c) {
        case 0 :
            return bt.getName();
        case 1 :
            return bt.getThreadState();
        case 2 :
            return bt.isDaemonThread();
        case 3 :
            return bt.getThreadType();
    }
    return null;
}
```

**Bump Run Manager**

```
... .. BumpRunManager$ThreadData>getThreadState(...)
@Override public BumpThreadState getThreadState() {
    return thread_state;
}
```

**Bump Thread View**

```
... .. BddtThreadView$ThreadsModel>getRowCount(...)
@Override public int getRowCount() {
    return bump_threads.size();
}
```

The thread model used by the viewer is a direct view on the bump\_threads structure.

**Table 1: Error/Warning Messages**

Description	Resource	Line
W The type BoppFontOption.FontChooser.Pr...	BoppFontOption...	247
W The field BudaBubbleArea.bottom_color i...	BudaBubbleArea...	87
W The method getProperDock(BudaBubble)...	BudaBubbleArea...	1192
W The field BudaBubbleLink.link_data is ne...	BudaBubbleLink...	76

**Table 2: TODO Messages**

Description	Resource	Line
N TODO : need to clone the tree structure h...	BddtStackModel.j...	303
N TODO : Update thread	BddtThreadView.j...	405
N TODO : Update thread	BddtThreadView.j...	405
N TODO : reformat initial <PRE>...</PRE> ...	BdocDocItem.java	106
N TODO : This might have to be more appro...	BdocRepository.java	210

**Table 3: Return Type of size()**

Return Type
int

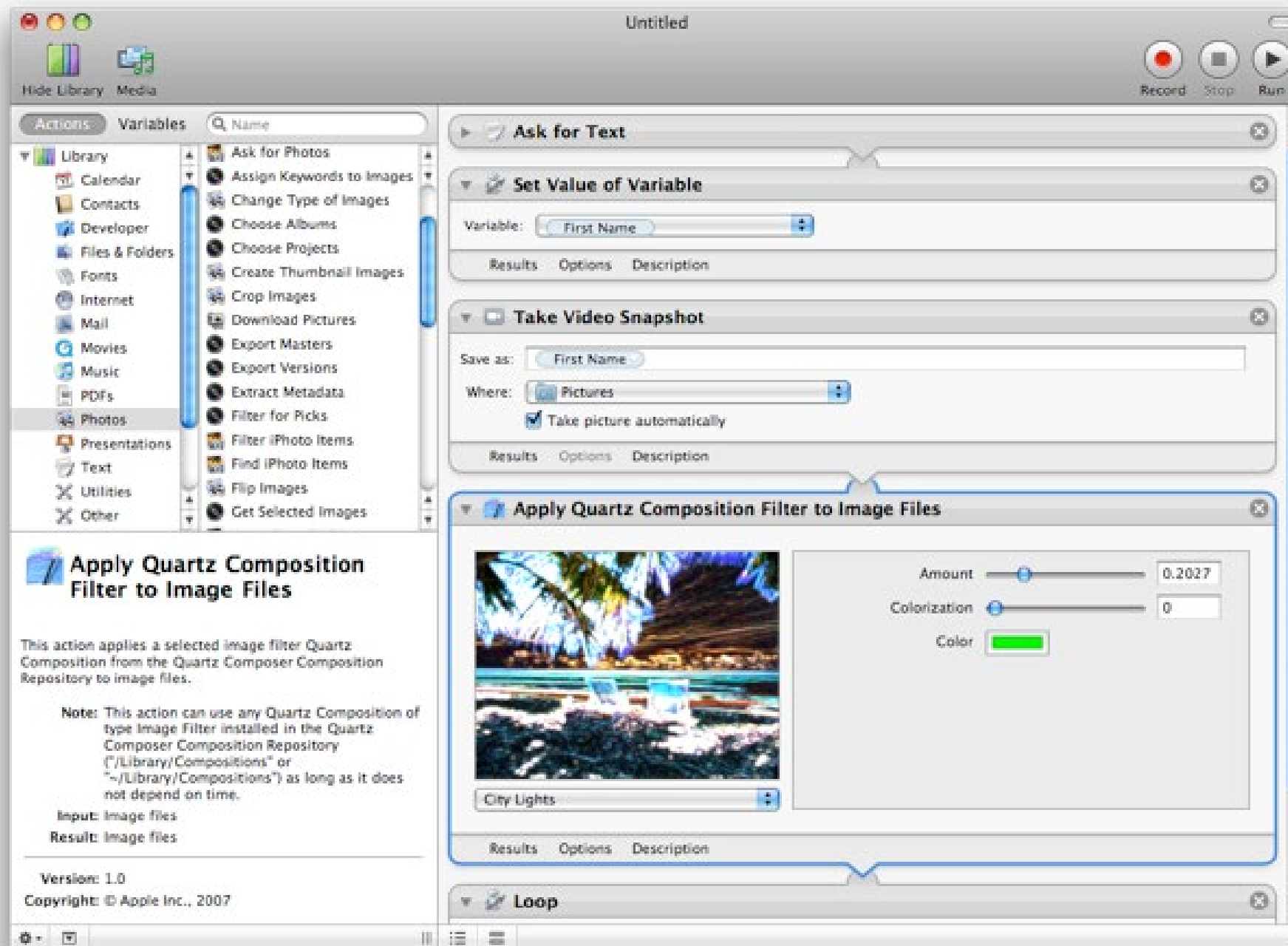
# Work Flow

## Step-based Programming



# Automator

# 2005-Today



Apple

# IFTTT

# 2010-Today

The screenshot displays three IFTTT automation recipes, each with a description, creation date, and status. Each recipe includes a set of control icons: a power button, a trash can, a share icon, and an edit icon.

- Recipe 1:** "if YouTube then [Blog icon]". Description: "When new video is uploaded, automatically creates new post with video embed on your blog". Status: "created about 3 hours ago never triggered".
- Recipe 2:** "if [Twitter icon] then [Twitter icon]". Description: "Twitter DM thanks for follow + newsletter promotion". Status: "created June 27, 2012 last triggered 1 day ago triggered 48 times".
- Recipe 3:** "if [RSS icon] then [Blog icon]". No description or status is visible for this recipe.

Alexander Tibbets, Linden Tibbets

# WorkFlow

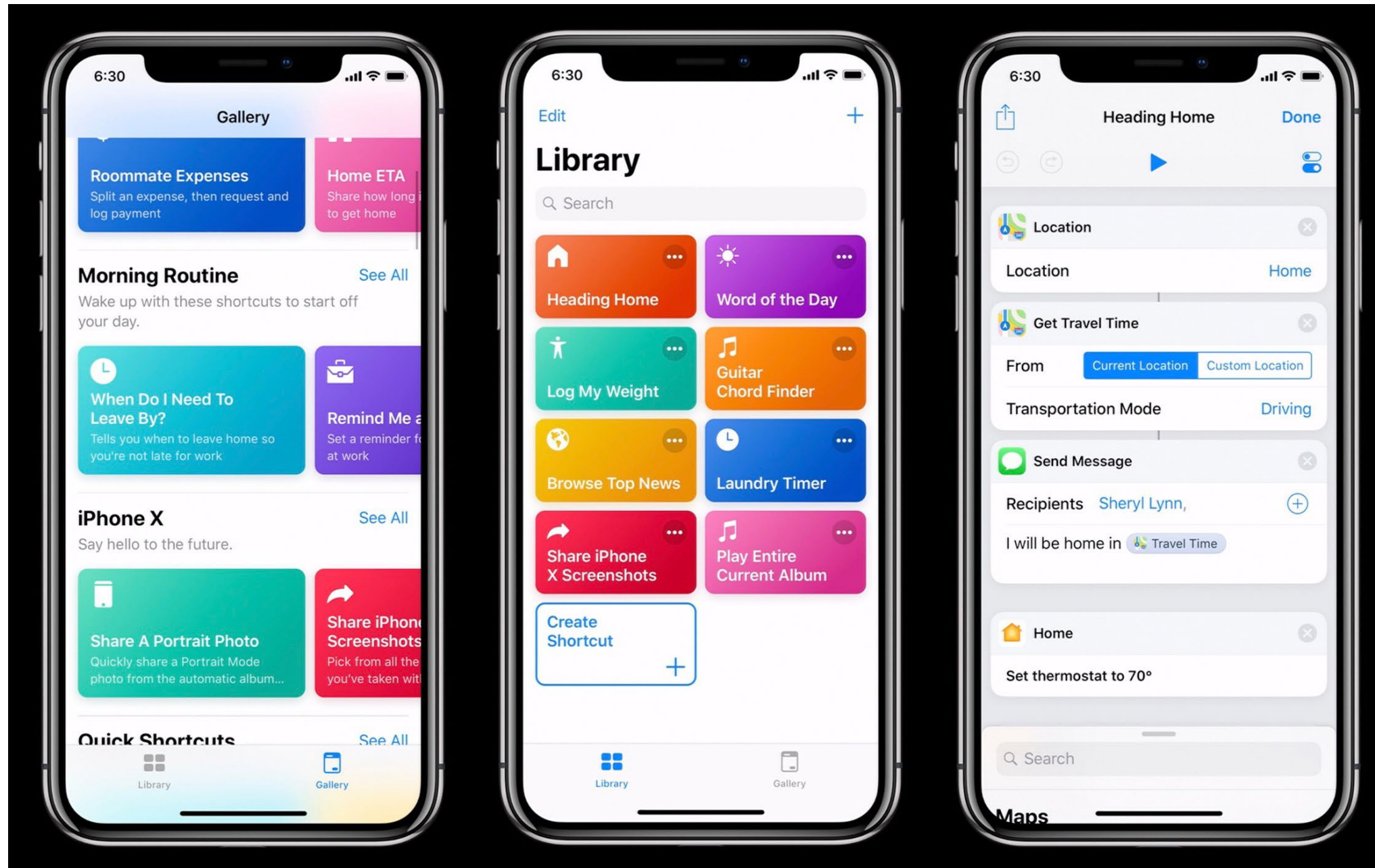
2014-



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

# Siri Shortcuts

2018



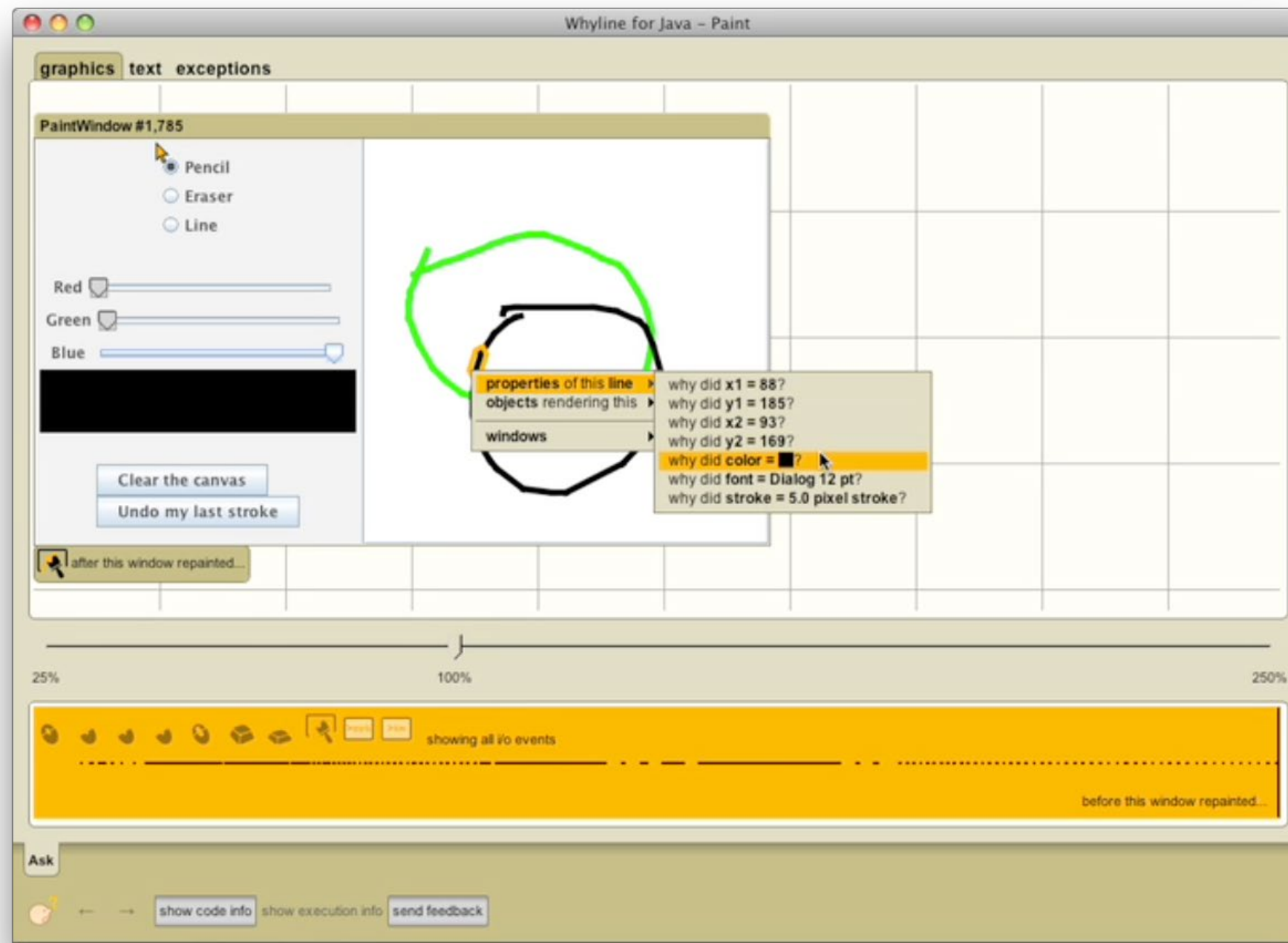
Apple

# Visible State Editors

Introduce Transparency to  
Traditional Programming

# Whyline

2008



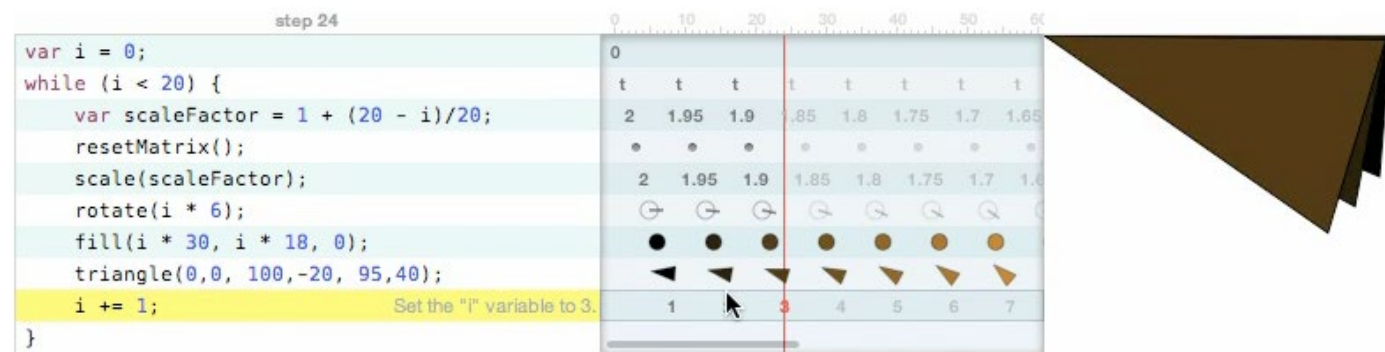
Andrew Ko at Carnegie Mellon University (HCI Institute)

# Learnable Programming (Essay)

2012

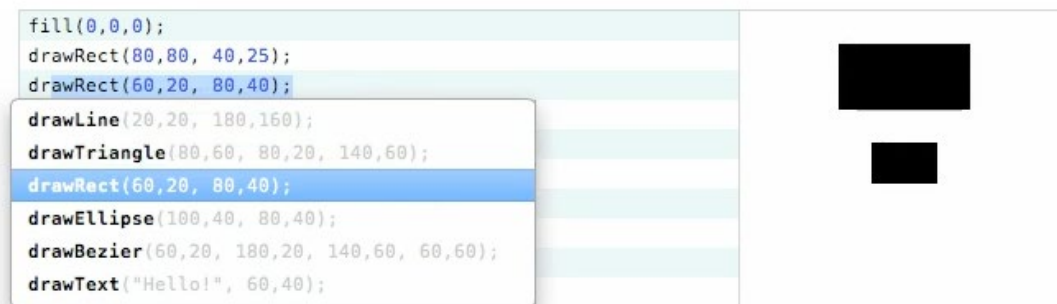
step 24

```
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;
}
```



The screenshot shows a code editor with a while loop that iterates 20 times. Each iteration scales the drawing by a factor of  $1 + (20 - i)/20$  and rotates it by  $i * 6$  degrees. The code fills a triangle with a color based on the iteration number  $i$ . The visual output shows a fan of triangles starting from a single point and spreading out as they are drawn.

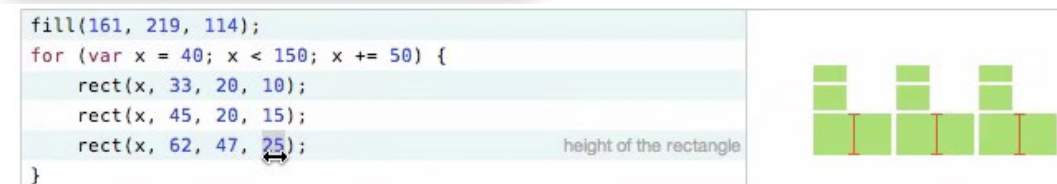
```
fill(0,0,0);
drawRect(80,80, 40,25);
drawRect(60,20, 80,40);
drawLine(20,20, 180,160);
drawTriangle(80,60, 80,20, 140,60);
drawRect(60,20, 80,40);
drawEllipse(100,40, 80,40);
drawBezier(60,20, 180,20, 140,60, 60,60);
drawText("Hello!", 60,40);
```



The screenshot shows a series of drawing commands including fill, drawRect, drawLine, drawTriangle, drawRect, drawEllipse, drawBezier, and drawText. The visual output shows a simple figure composed of black shapes: a rectangle, a line, a triangle, another rectangle, an ellipse, a Bezier curve, and the text "Hello!".

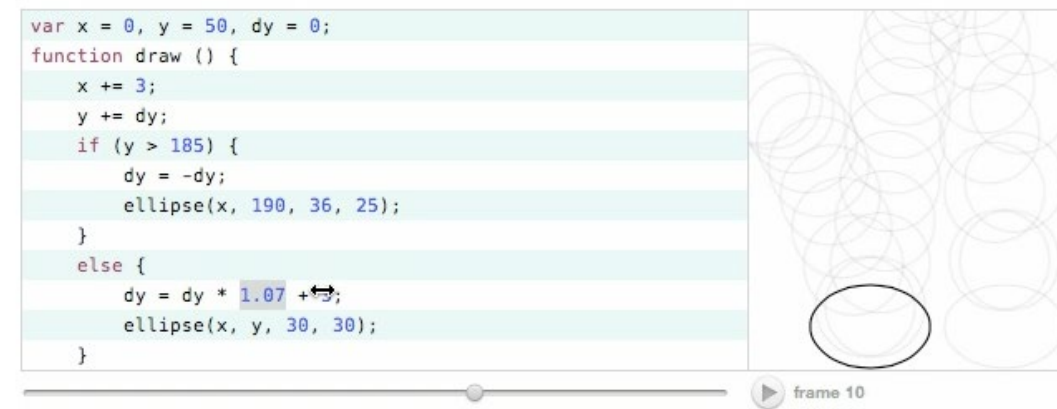
```
fill(161, 219, 114);
for (var x = 40; x < 150; x += 50) {
  rect(x, 33, 20, 10);
  rect(x, 45, 20, 15);
  rect(x, 62, 47, 25);
}
```

height of the rectangle



The screenshot shows a for loop that iterates over x values from 40 to 150 in increments of 50. The code draws three rectangles at each iteration with different heights. The visual output shows a bar chart with three bars of increasing height.

```
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);
  }
}
```



The screenshot shows a function draw() that iteratively draws ellipses. The x-coordinate increases by 3 units each time. The y-coordinate increases by dy, and dy is multiplied by 1.07 each time. The visual output shows a spiral of ellipses that grows in size and changes direction as it moves.

Bret Victor

# Media for Thinking The Unthinkable (Nile Demo) 2013

**initial input**  
5 Points

**MakePolygon ()**  
processed 5 Points  
output 4 Beziers

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

**RoundPolygon ()**  
processed 4 Beziers  
output 4 Beziers

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

**TransformBeziers ()**  
processed 4 Beziers  
output 4 Beziers

```
TransformBeziers (M:Matrix) : Bezier >> Bezi
∀ (A, B, C)
  >> (MA, MB, MC)
```

**initial input**  
3 Beziers

**Rasterize ()**  
processed 3 Beziers  
output 31 SpanCoverages

**Texture ()**  
processed 31 SpanCoverages  
output 25 Pixels

**ExpandSpans ()**  
processed 56 SpanCoverages  
output 25 PointCoverages

```
ExpandSpans ()
∀ (x, y, c,
  if c >
  >>
  <<
```

**ProjectLinearGradient ()**  
processed 25 PointCoverages  
output 25 Reals

```
ProjectLinearGr
v = B - A
Δs = v / (
s00 = A · Δ
∀ (P,_)
  >> P ·
```

**PadGradient ()**  
processed 25 Reals  
output 25 Reals

```
PadGradient ()
∀ s
  >> 0 ▷
```

**GradientSpan ()**  
processed 25 Reals  
output 25 Colors

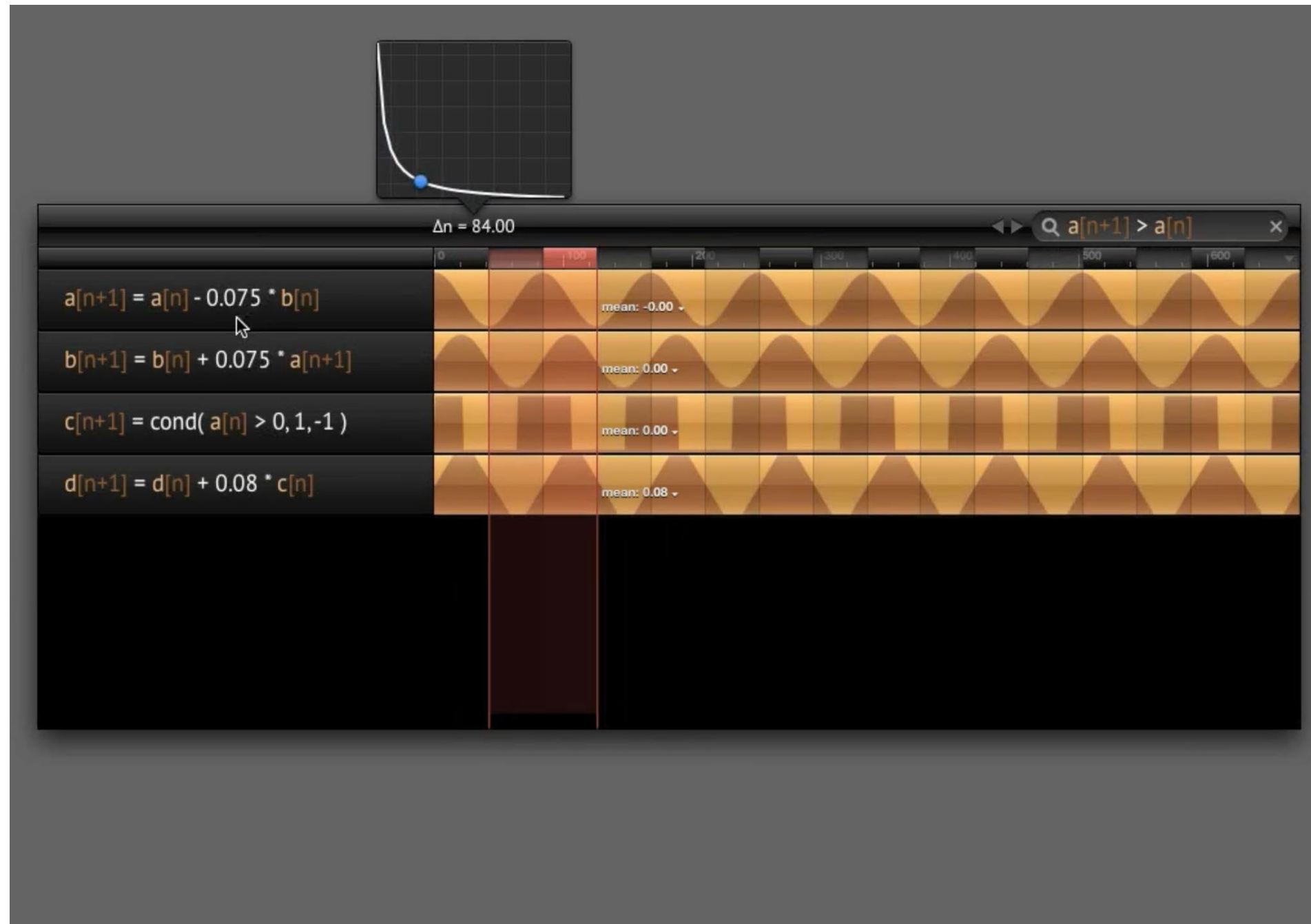
```
GradientSpan (A
∀ s
  >> sA +
```

Bret Victor



# Media for Thinking The Unthinkable (MSP)

2013



Bret Victor

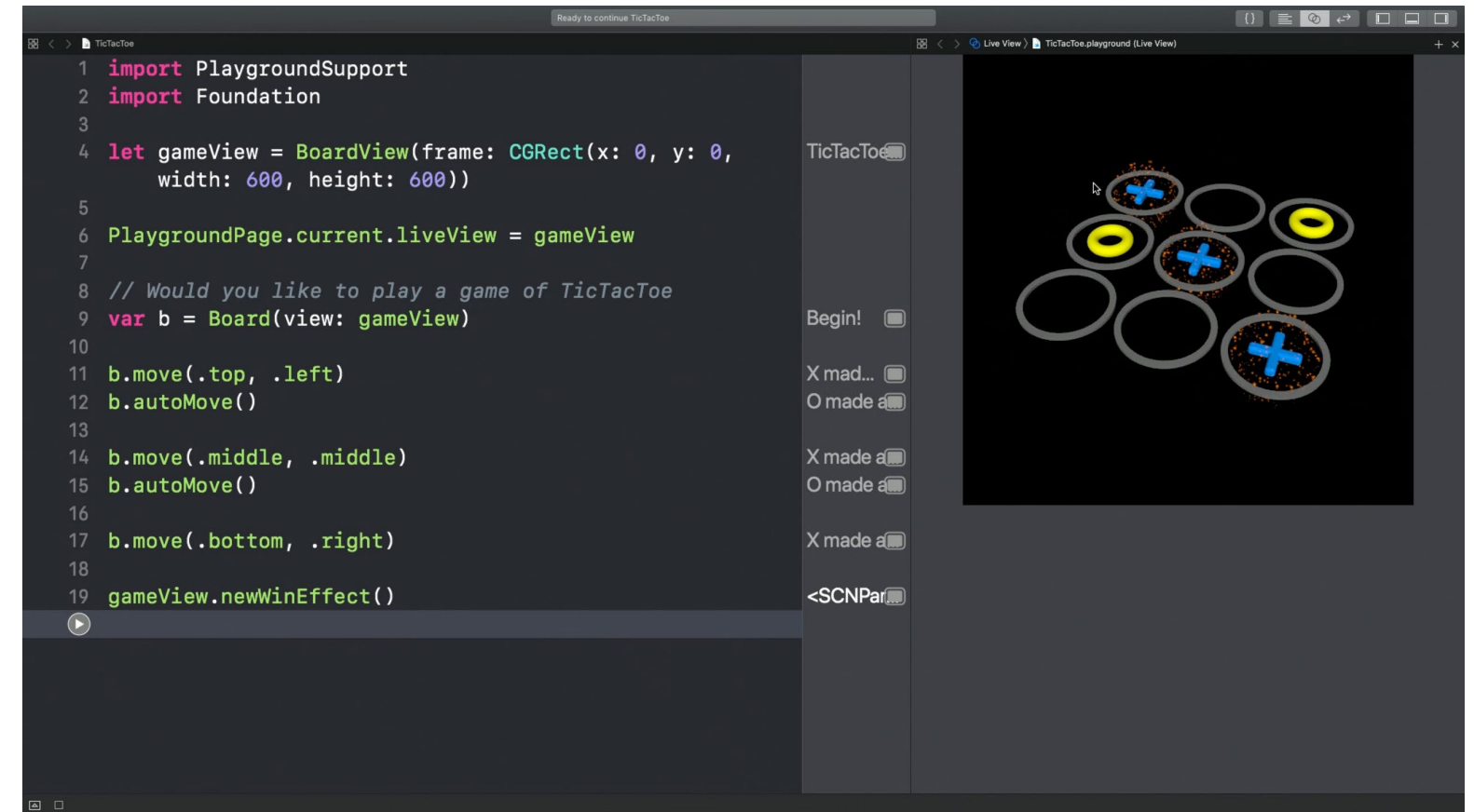
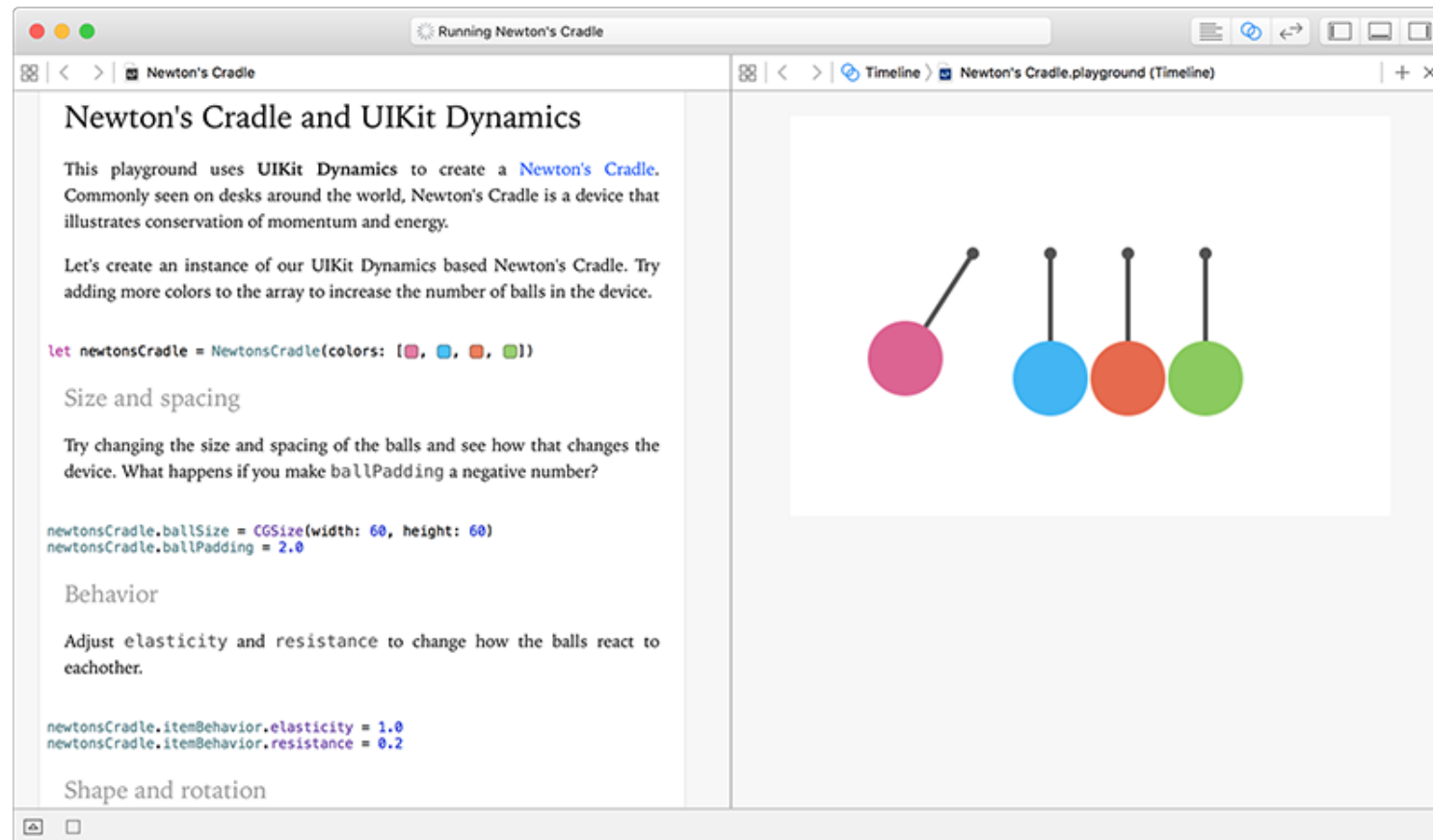
The screenshot shows an iPad app interface for a financial calculator. At the top, it displays 'iPad', '9:41 AM', and '100%' battery. The interface is divided into several sections:

- Input Section:** Three input fields are shown: 'Initial Deposit' with the value '10,000', 'Years' with the value '20', and 'Interest' with the value '0.04'. A slider below the interest field ranges from 0 to 10.
- Graph:** A line graph showing the growth of the deposit over time. The y-axis represents the amount, ranging from 10,000 to 30,000. The x-axis represents time in years, ranging from 0 to 25. A purple curve starts at (0, 10,000) and rises to approximately 22,225.82 at year 20. A horizontal dashed line at 22,225.82 intersects the curve at year 20.
- Equation Section:** The compound interest formula is displayed:  $10,000 (1 + 0.04 / 12)^{12 \times 20} = 22,225.82$ . The result '22,225.82' is highlighted in a purple box.
- Profit Calculation:** Below the formula, the profit is calculated:  $22,225.82 - 10,000 = 12,225.82$ . The result '12,225.82' is highlighted in a green box.
- Profit per Month:** The profit is then divided by the number of months:  $12,225.82 / 20 / 12 = 50.94$ . The result '50.94' is highlighted in an orange box.
- Keypad:** A standard calculator keypad is visible on the right side of the screen, with orange buttons for the equals sign, plus, minus, multiplication, and division.

Andreas Karlsson for Tydlig Software AB

# Xcode Playgrounds

# 2015-Today



Apple

# Fragment

2017

The screenshot displays a 3D software interface with several panels:

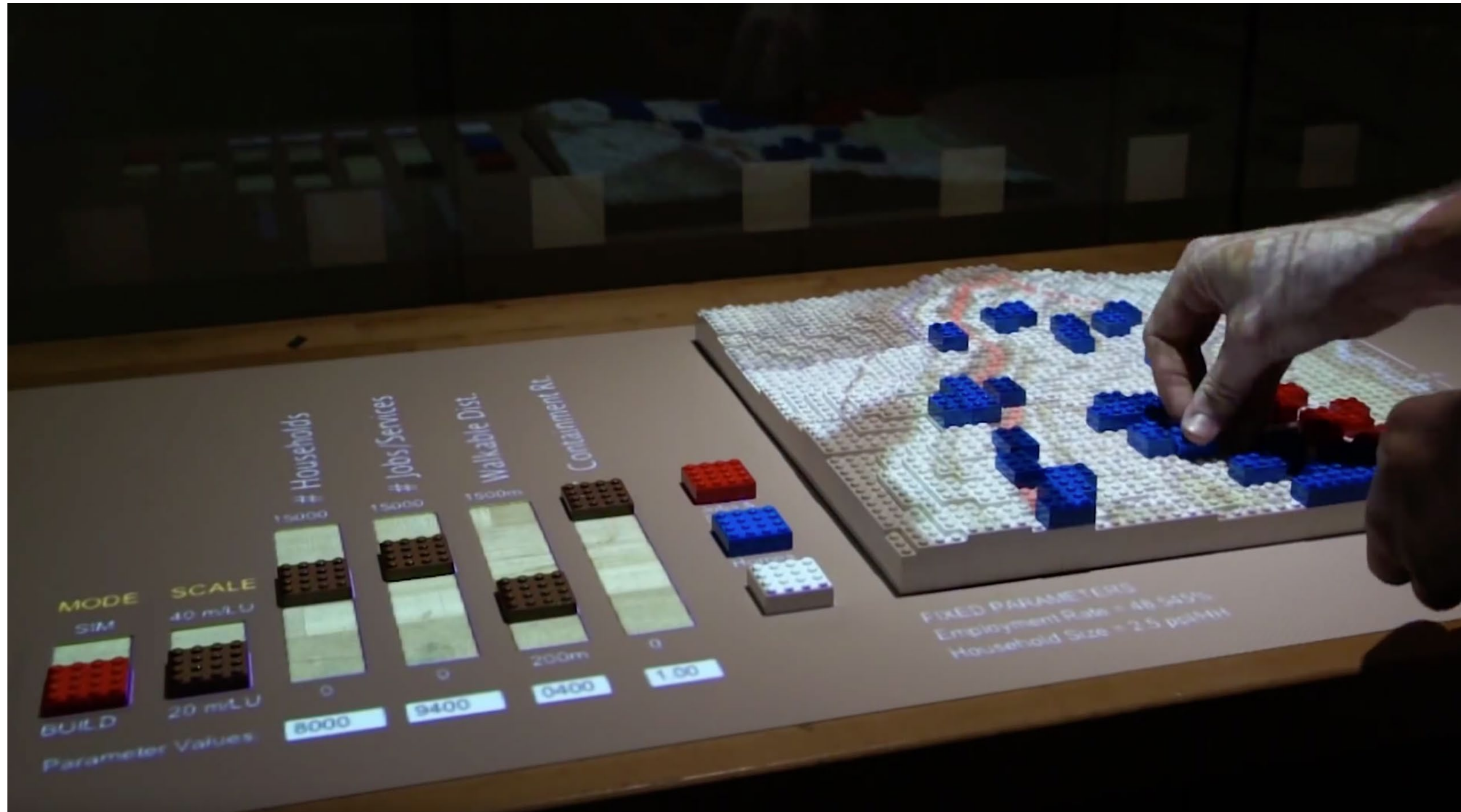
- FRAGMENT:** Shows a colorful, fractal-like texture rendered on a plane. The texture is composed of organic, flowing shapes in shades of cyan, magenta, yellow, and blue.
- EXPORTER:** Contains various settings for rendering and exporting the image, including options for save image as, output image scale, render format (ROV, PNG), and a list of examples like 1\_BUTTON, 2\_TOGGLE, 3\_RANGE, etc.
- CONSOLE:** Displays the status "SUCCESS V2.0.0" and a "PARAMS" section with various sliders and numerical values.
- shader.frag — Shaders:** Shows the GLSL code for the fragment shader. The code includes several include statements for uniforms, gamma, easing, fbm, and pi. It defines uniforms for mult, off, and speed, and uses the fbm function to generate the fractal texture. The final output is a color vector oColor.

```
1 #include "uniforms.glsl"
2 #include "gamma.glsl"
3 #include "easing.glsl"
4 #include "fbm.glsl"
5 #include "pi.glsl"
6
7 uniform vec3 mult; //ui:0.0,4.0,4.0
8 uniform vec3 off; //ui:0.0,4.0,4.0
9
10 uniform float speed; //slider:0.0,4.0,0.5
11 in vec4 vColor;
12 in vec2 vTexCoord;
13
14 out vec4 oColor;
15
16 void main(void)
17 {
18     float time = inOutExpo( sin( iAnimationTime * PI ) );
19     float r = 1.0 + fbm( vec3( mult.x * vTexCoord + off.x, speed * ( time ) ), 3 );
20     float g = 1.0 + fbm( vec3( mult.y * vTexCoord + off.y, speed * ( time + 1 ) ), 3 );
21     float b = 1.0 + fbm( vec3( mult.z * vTexCoord + off.z, speed * ( time + 2 ) ), 3 );
22
23     r *= 0.5;
24     g *= 0.5;
25     b *= 0.5;
26
27     vec3 col = vec3( r > g || r > b ? r : 0.0,
28                   g > r || g > b ? g : 0.0,
29                   b > g || b > r ? b : 0.0 );
30
31     col = gamma( col );
32     oColor = vec4( col, 1.0 );
33     // oColor = vec4( pow( 1.0 - col, vec3( 64.0 ) ), 1.0 );
34 }
35
```

Reza Ali

# CityScope

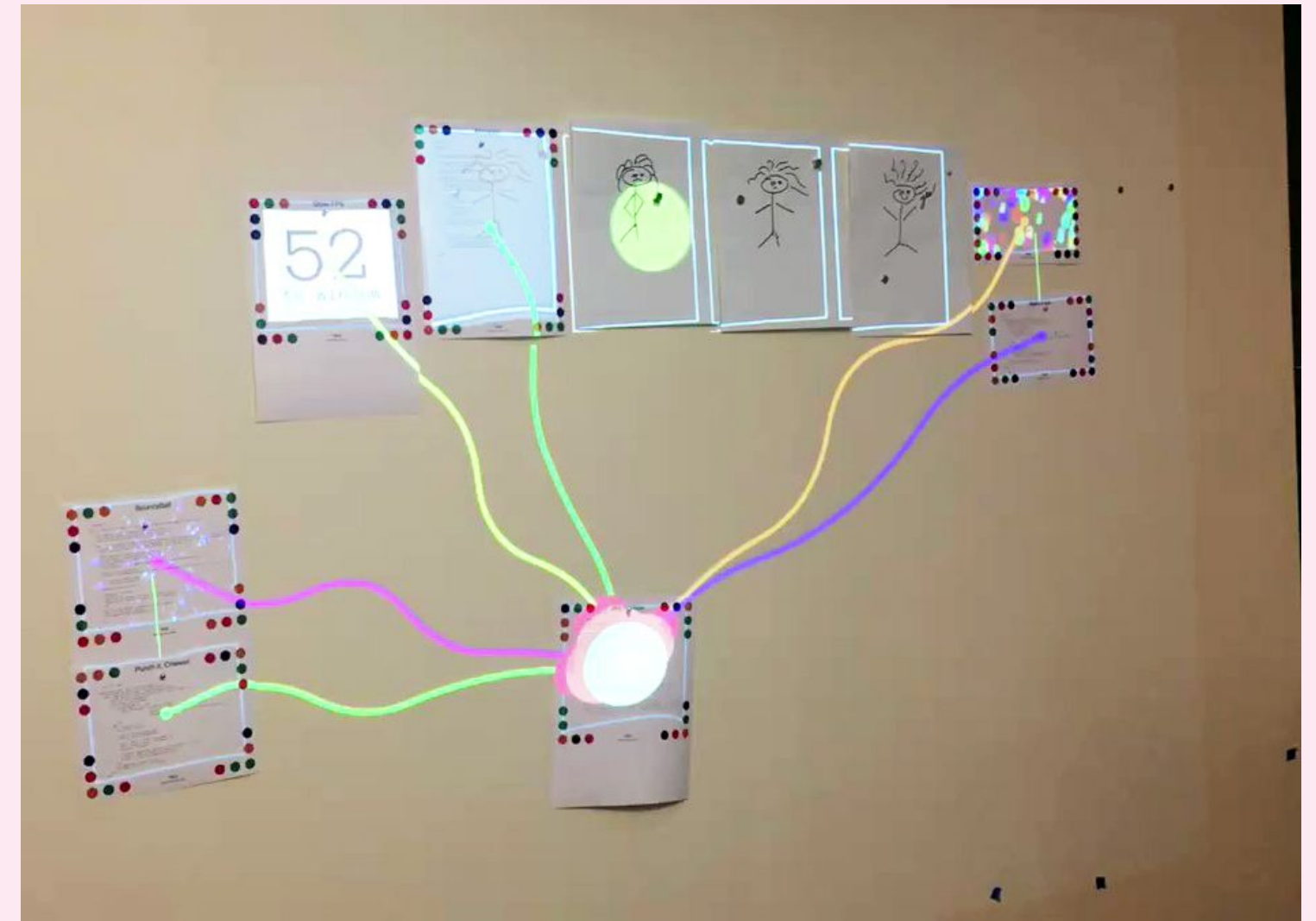
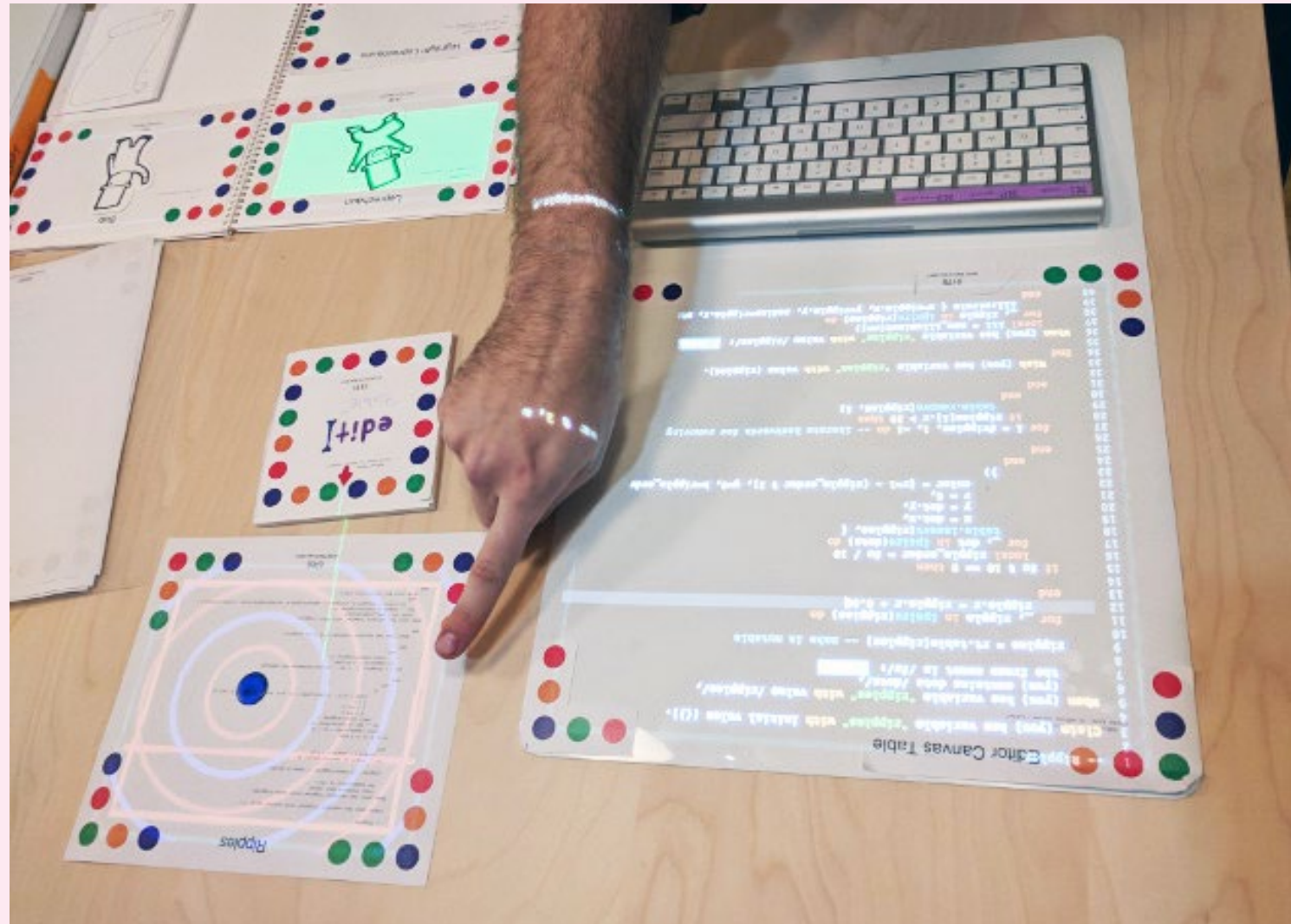
2015



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

# Dynamicland

2017-Today



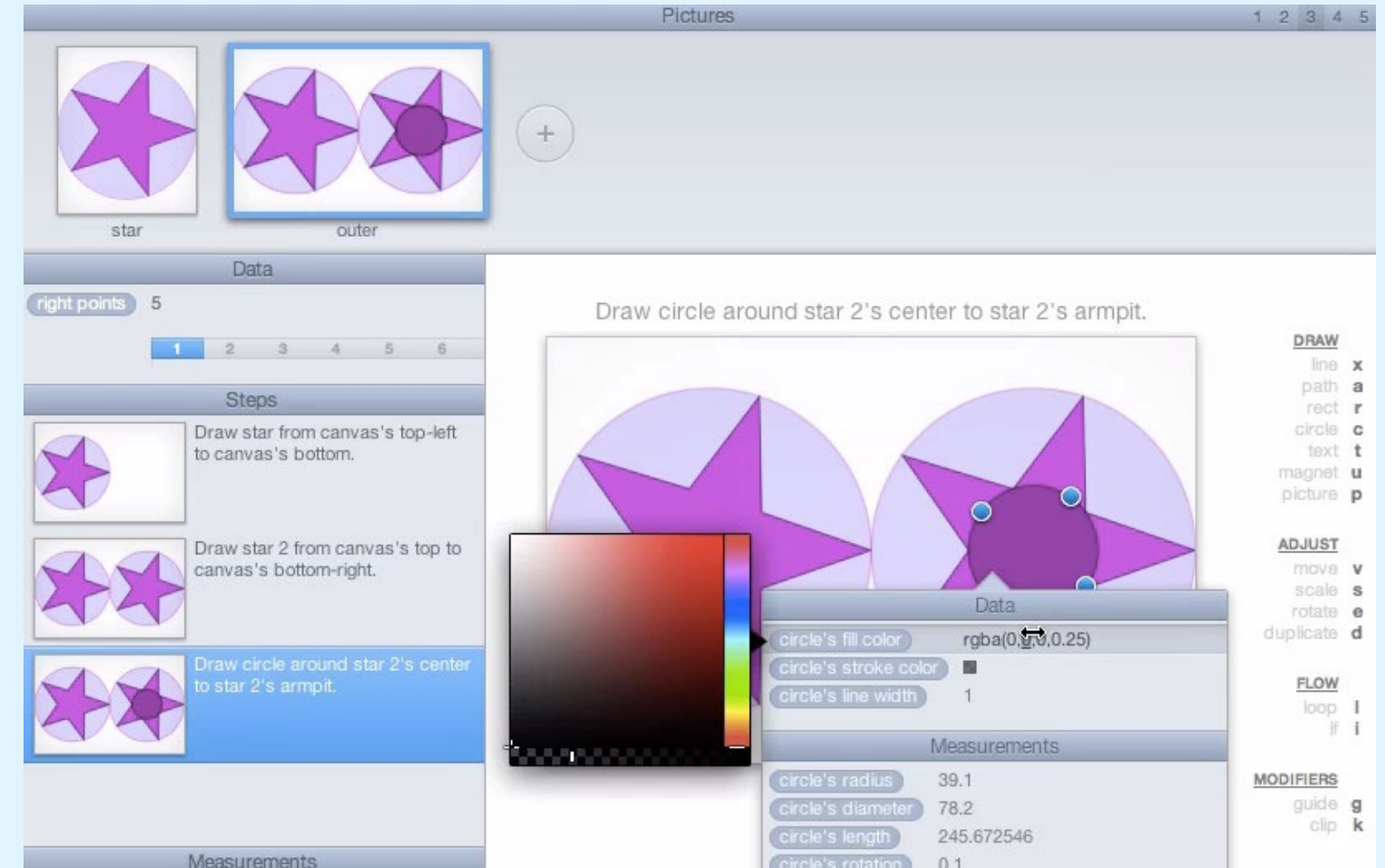
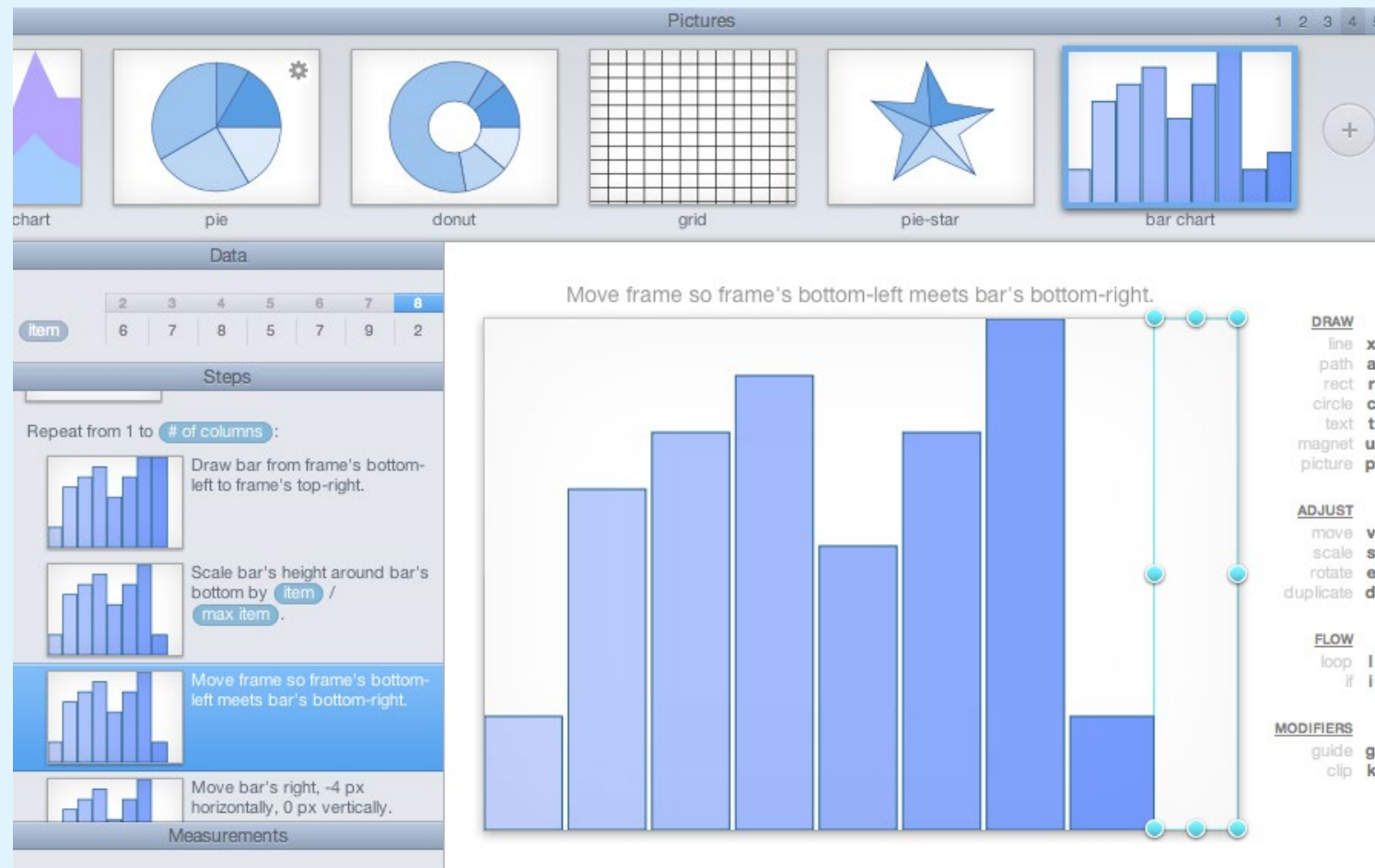
Bret Victor and Others at Y Combinator Research's Human Advancement Research Community

# Dynamic Diagram Experiments

Direct-Manipulation +  
Symbolic Relationships

# Drawing Dynamic Visualizations

2013



Bret Victor



# ShaderShop

2014

The screenshot displays the ShaderShop interface. On the left is a 'Library' with 'Built In Functions' (Line, Abs, Fract, Floor, Sine) and 'Custom Functions' (Smooth Step, Smooth Saw, Random 2D, Step Random 2D, Horizontal, Noise 2D, and an 'Untitled Function'). The central workspace features a 2D plot with axes x1 and x2, and a 1D graph with axis y1. The 2D plot shows a grayscale noise pattern, and the 1D graph shows a blue curve. At the bottom, a mathematical expression for the noise function is displayed:

$$\text{Noise 2D}(x_1, x_2) + \text{Noise 2D}\left(\frac{x_1, x_2}{\begin{pmatrix} 0.5 & 0 \\ 0 & 0.5 \end{pmatrix}}\right) * \begin{pmatrix} 0.5 & 0 \\ 0 & 1 \end{pmatrix} + \text{Noise 2D}\left(\frac{x_1, x_2}{\begin{pmatrix} 0.25 & 0 \\ 0 & 0.25 \end{pmatrix}}\right) * \begin{pmatrix} 0.25 & 0 \\ 0 & 1 \end{pmatrix} + \text{Noise 2D}\left(\frac{x_1, x_2}{\begin{pmatrix} 0.125 & 0 \\ 0 & 0.125 \end{pmatrix}}\right) * \begin{pmatrix} 0.125 & 0 \\ 0 & 1 \end{pmatrix} + \text{Noise 2D}\left(\frac{x_1, x_2}{\begin{pmatrix} 0.063 & 0 \\ 0 & 0.063 \end{pmatrix}}\right) * \begin{pmatrix} 0.063 & 0 \\ 0 & 1 \end{pmatrix}$$

On the right, the 'Compose' panel shows a stack of 'Noise 2D' nodes. Below it, the 'Inspector' panel shows a table of values:

Inspector	
x1	x2
+ 0	0
⊗ 1	0
0	1
y1	y2
+ 0	0
⊗ 1	0
0	1

Toby Schachman for Communication Design Group at SAP

# Apparatus

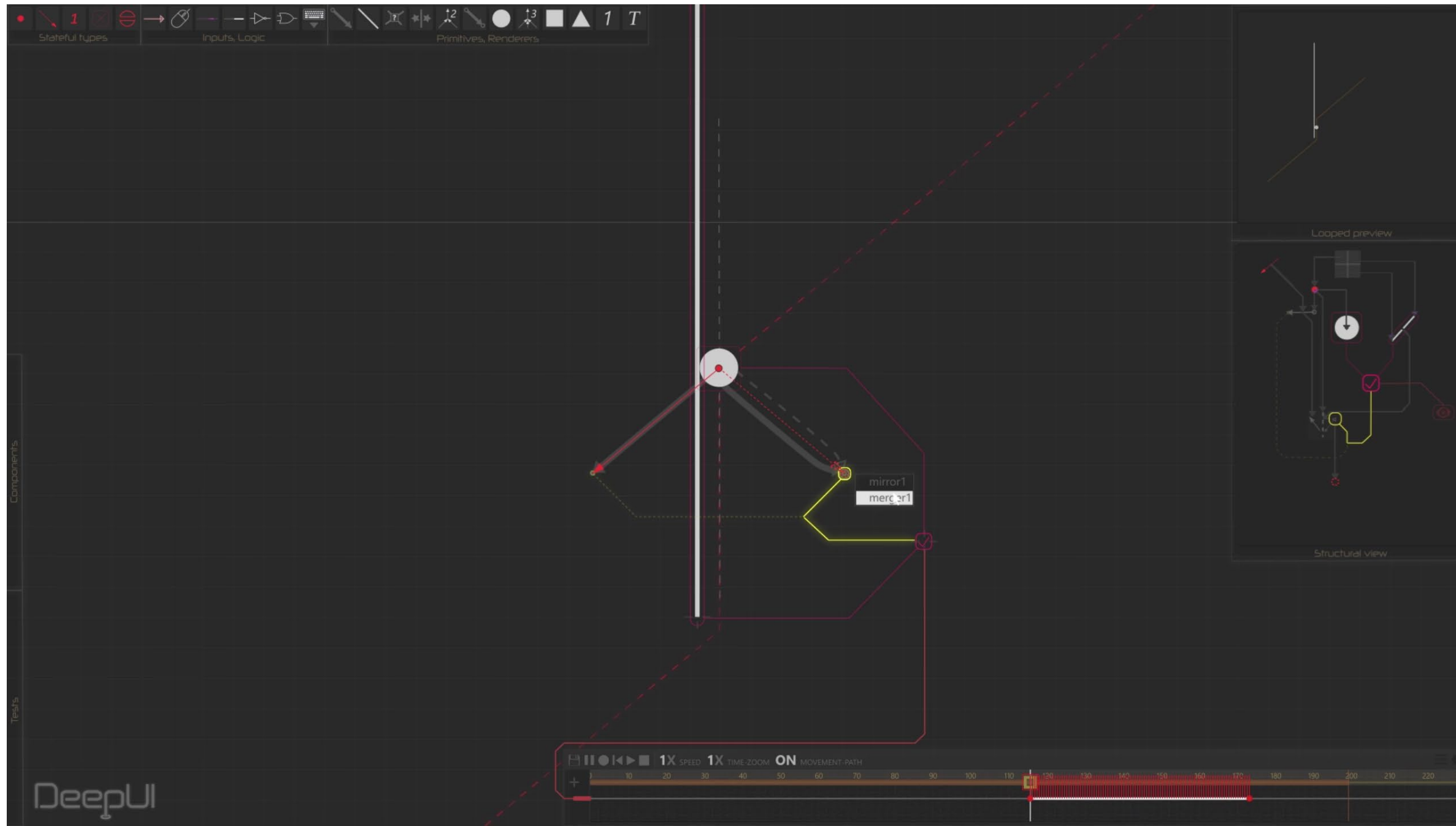
2015

The screenshot displays the Apparatus software interface. On the left is a 'Symbols' panel with icons for Rectangle, Circle, Text, and Involute. The central workspace shows a drawing of a circle with a red outline and a blue center point. A dotted line represents an involute curve starting from the circle. A blue line points from the text 'drag circle to draw involute' to the circle. On the right is the 'Outline' settings panel, which is currently expanded to show the 'Circle' settings. The 'Circle' settings include X and Y coordinates (both 0), a 'Rotate' property set to 'CurrentRot' (-8.2), and stroke properties (Color: 'rgba(0, 0, 0, 1)', Line Width: 3). Below the 'Circle' settings are the 'Involute' settings, which include X and Y coordinates (both 0), a 'Rotate' property set to 'Rot' with a value of -8.2, stroke properties (Color: 'rgba(0, 0, 0, 1)', Line Width: 3), and an 'Anchor' property with X and Rot checkboxes checked. The 'Variables' section at the bottom of the panel shows a 'Rotate' property set to 'CurrentRot' (-8.2). The 'Transform' section shows X and Y coordinates (both 0), Scale X and Y (both 1.00), and a 'Rotate' property set to 'CurrentRot' (-8.2). The 'Path' section shows 'Close Path' set to 'true'. The 'Fill' section shows 'Fill Color' set to 'rgba(0.93, 0.93, 0.93, 0)' and 'rgba(237, 237, 237, 0)'. The 'Stroke' section shows 'Stroke Color' set to 'rgba(0, 0, 0, 1)' and 'rgba(0, 0, 0, 1)'. The 'Tangent' section is also visible at the bottom of the panel.

Toby Schachman for Communication Design Group at SAP

# DeepUI

2017



Arnold Lagler

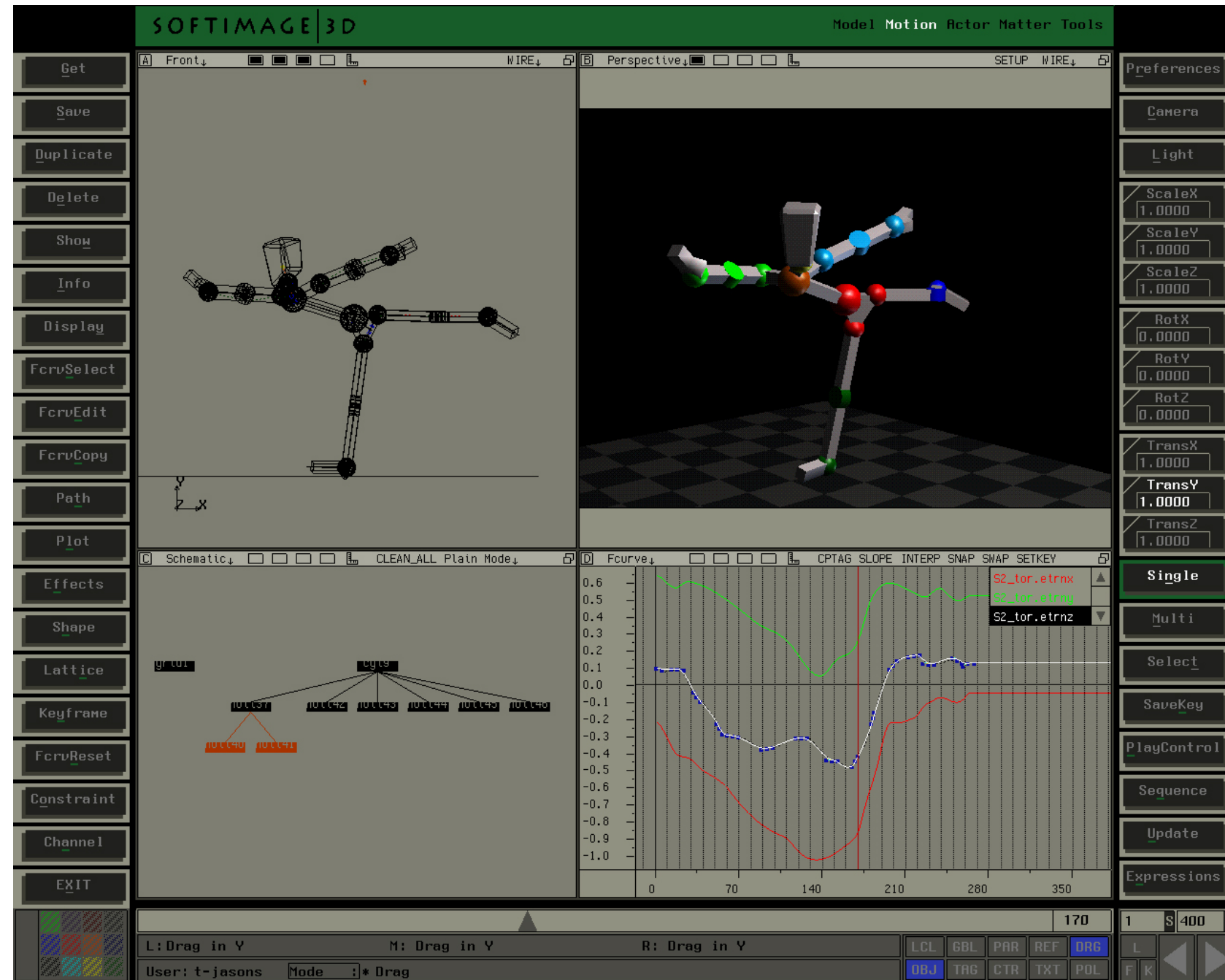
# Learning From/Within Cyberspace

How Game-Tech is enabling the Next Wave

# 3D (Mesh/NURBs) Creation Suites

# Softimage 3D

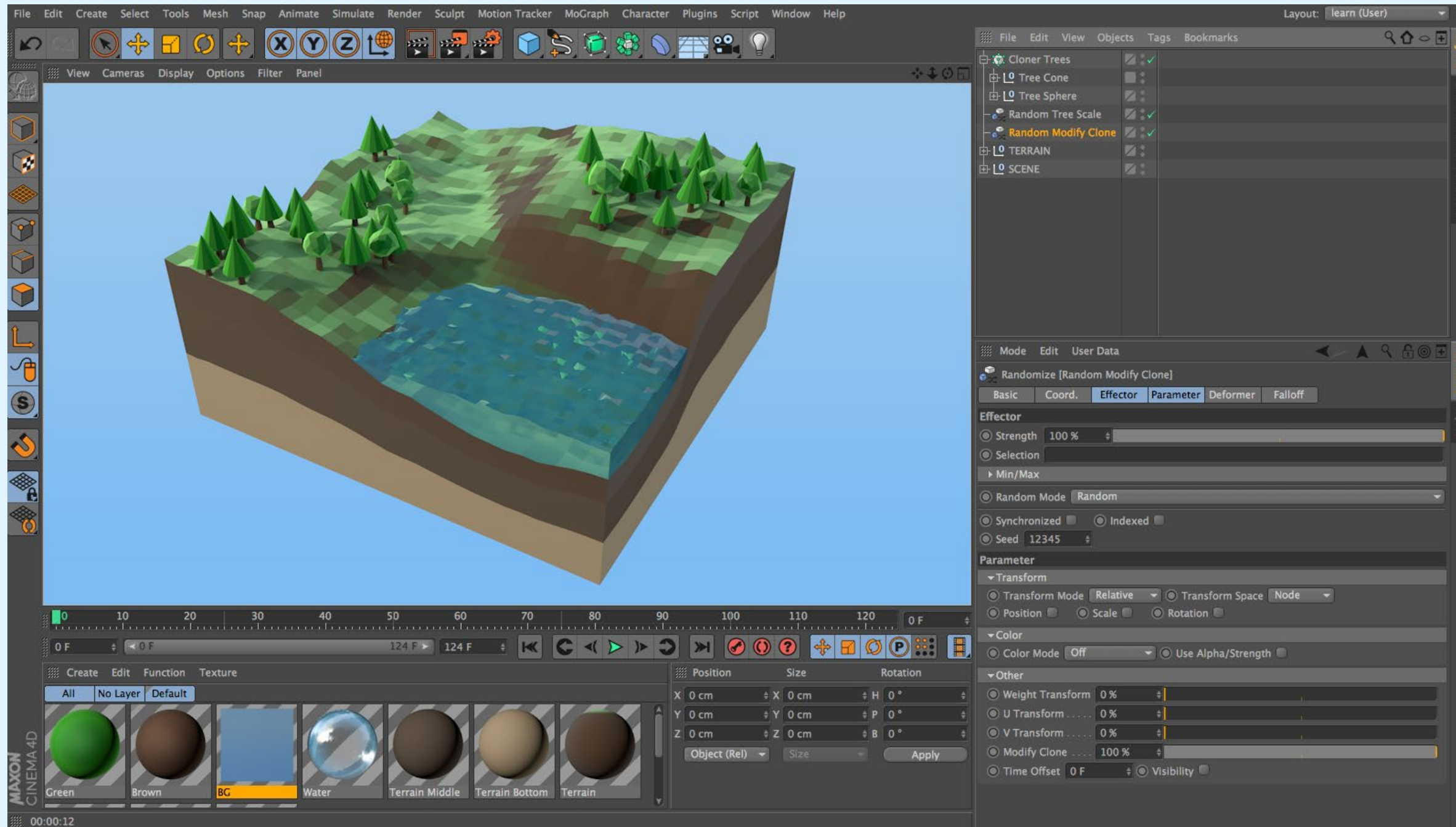
1988-2001



Softimage, Co.

# Cinema 4D

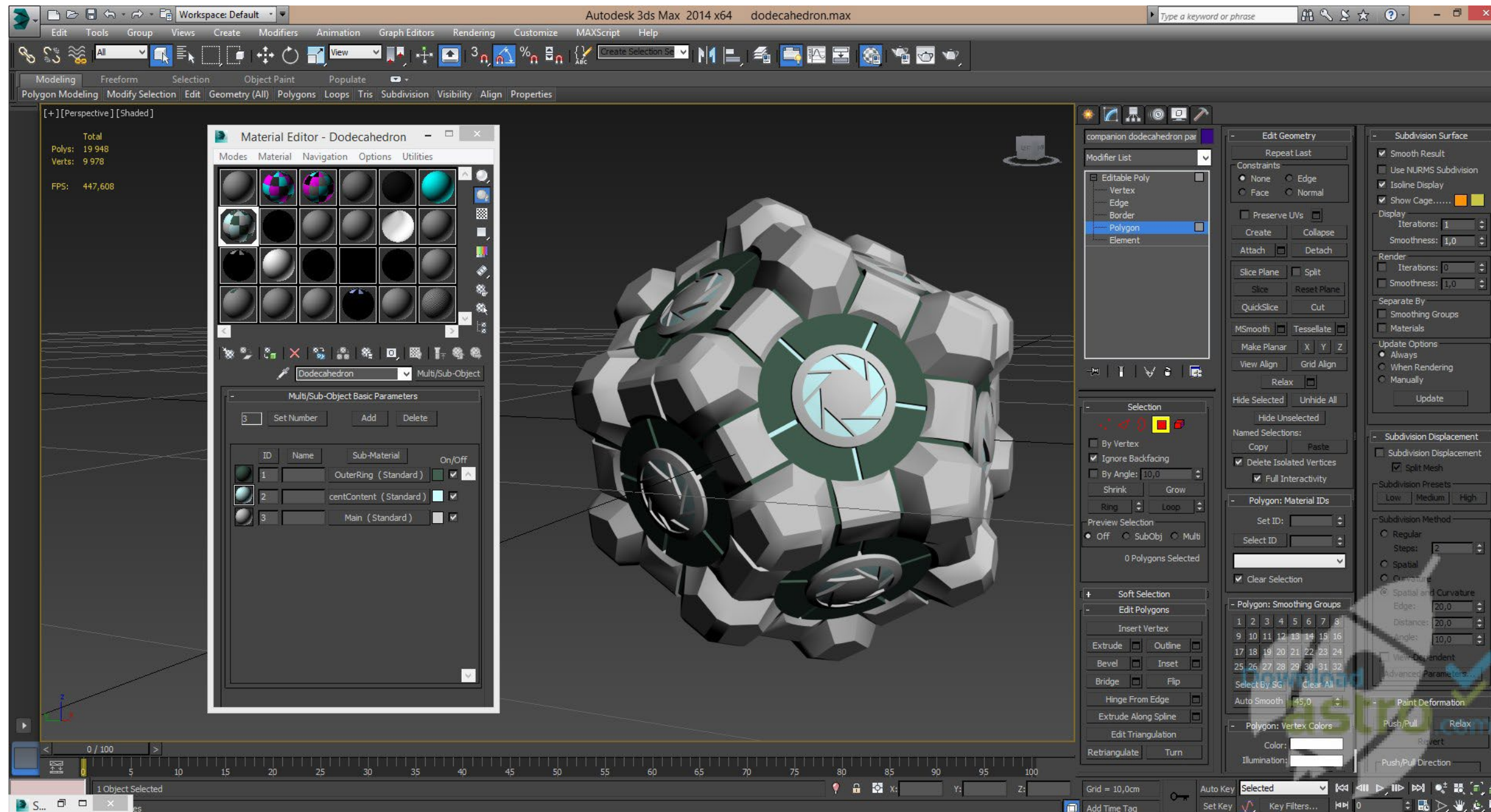
1990-2001



MAXON Computer GmbH

# 3DS Max

# 1996-Today



Autodesk



# Shake

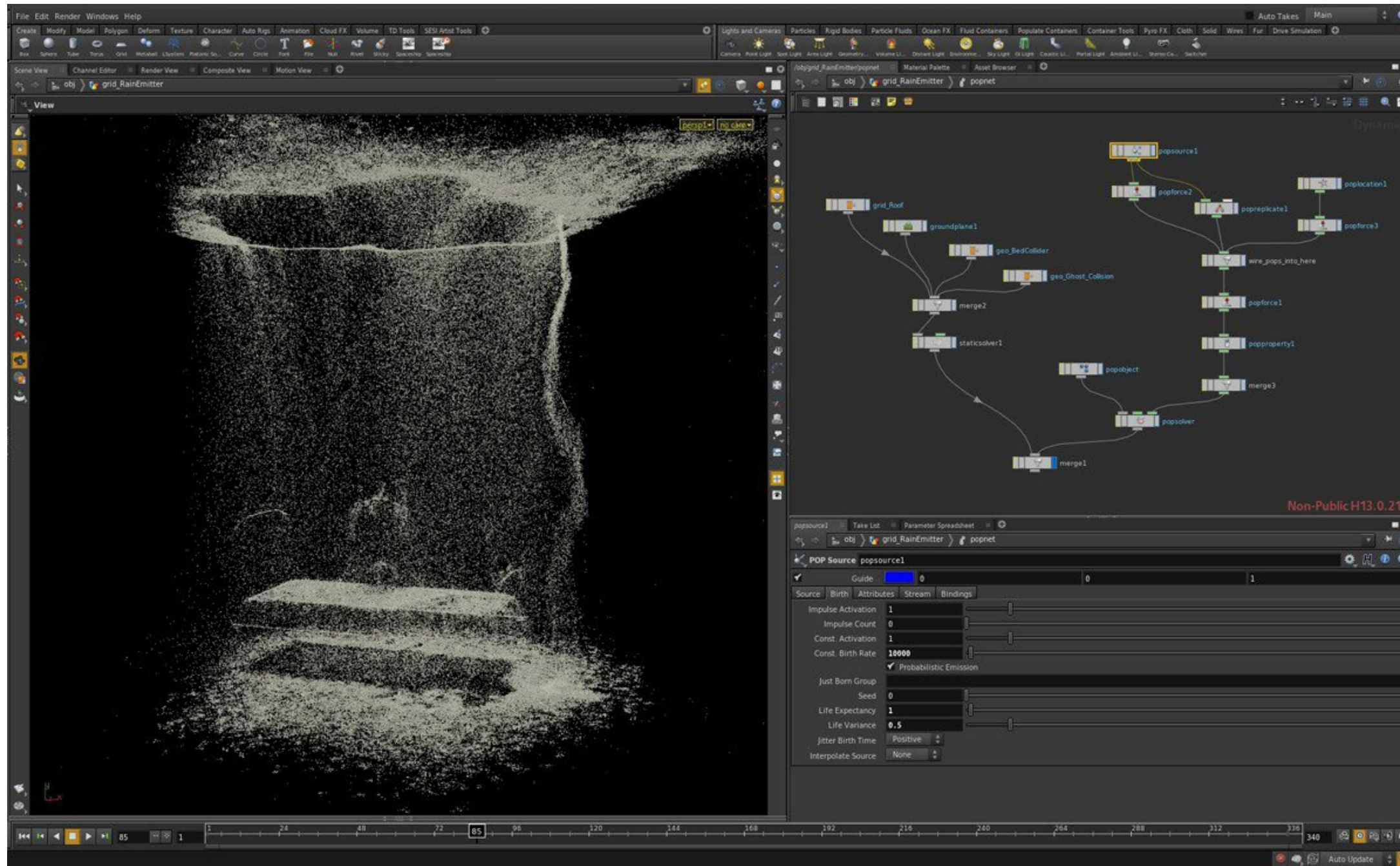
1997-2008



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

# Houdini

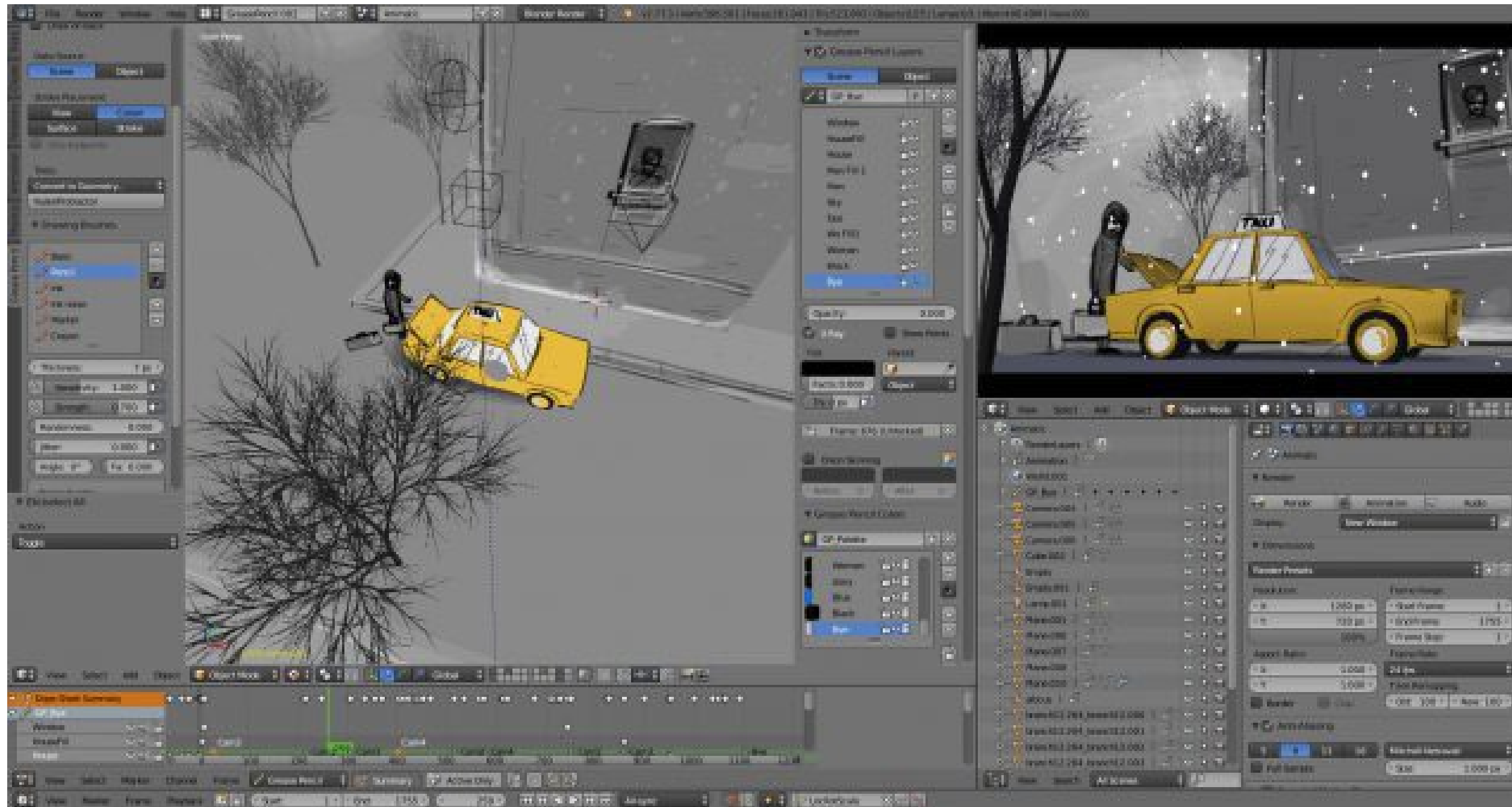
# 1996-Today



Side Effects Software Inc

# Blender

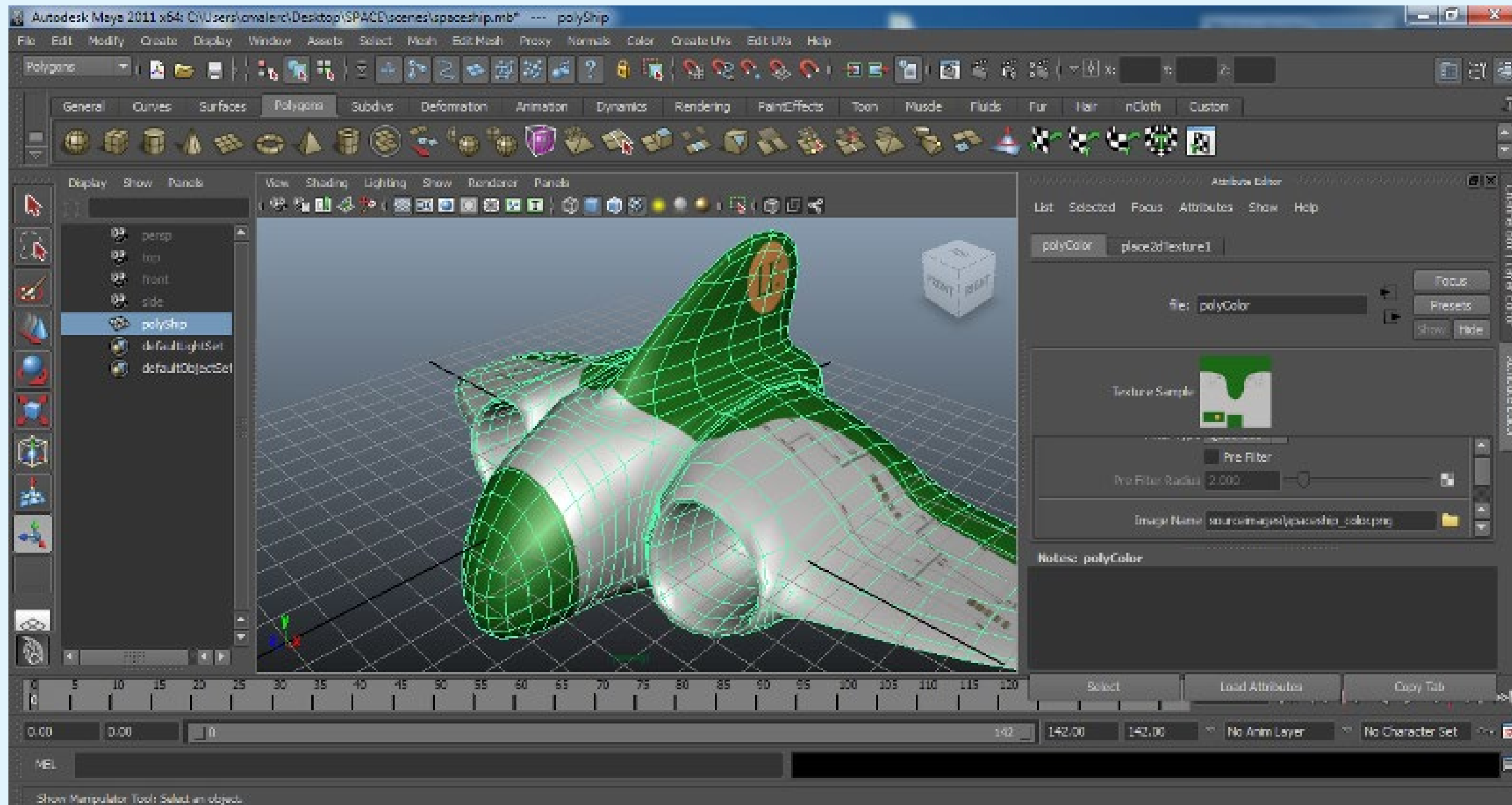
1998-Today



Blender Foundation

# Maya

1998-Today



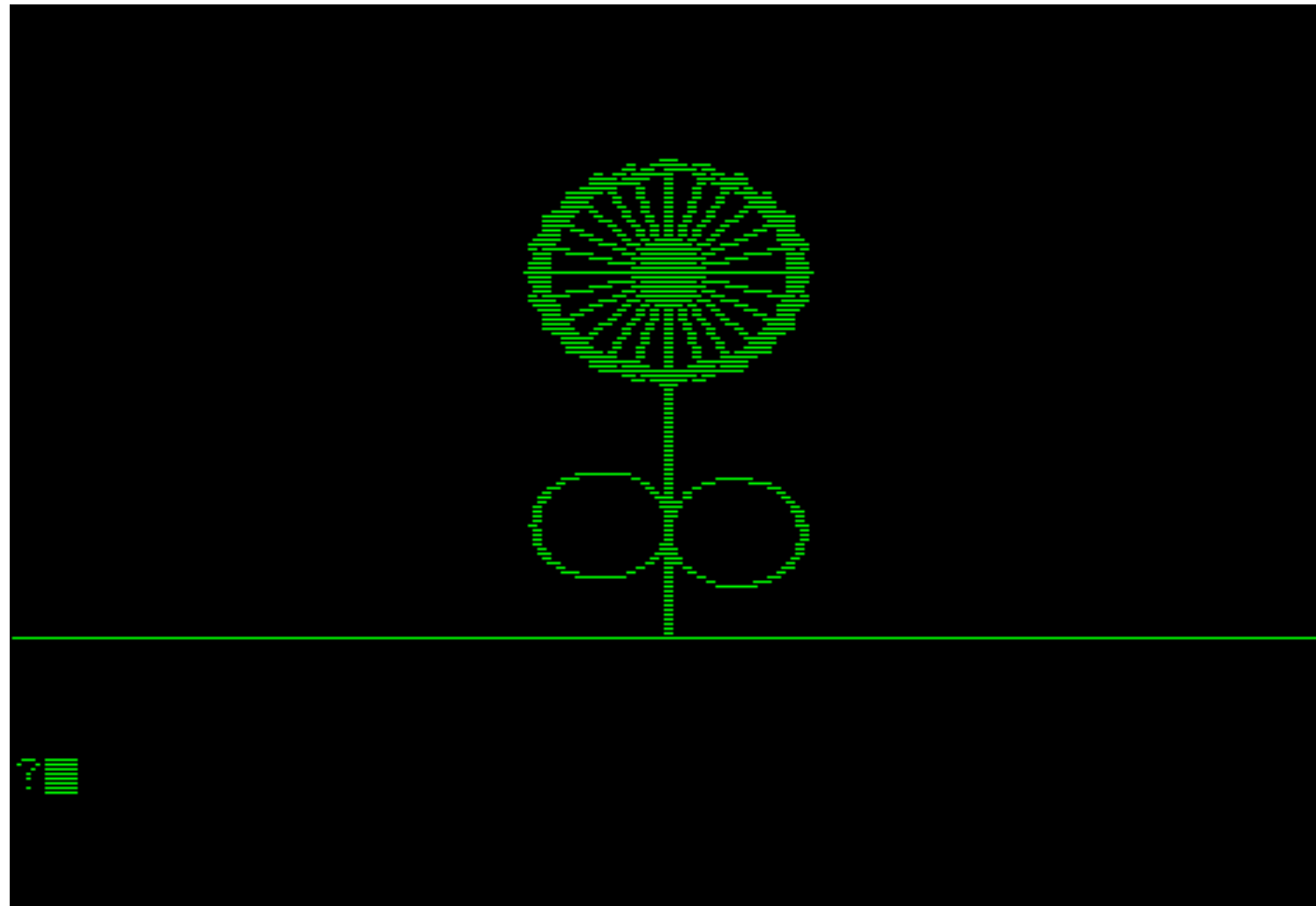
Alias Systems Corporation (Later Autodesk)

# **Learnable Programming**

Offers a 'Gentle Introduction'

# LOGO

1967-



```
TO GO
  CIRCLE 100
  FILL
  STOP
END

TO HELLO
  PRINT "HELLO"
  STOP
END

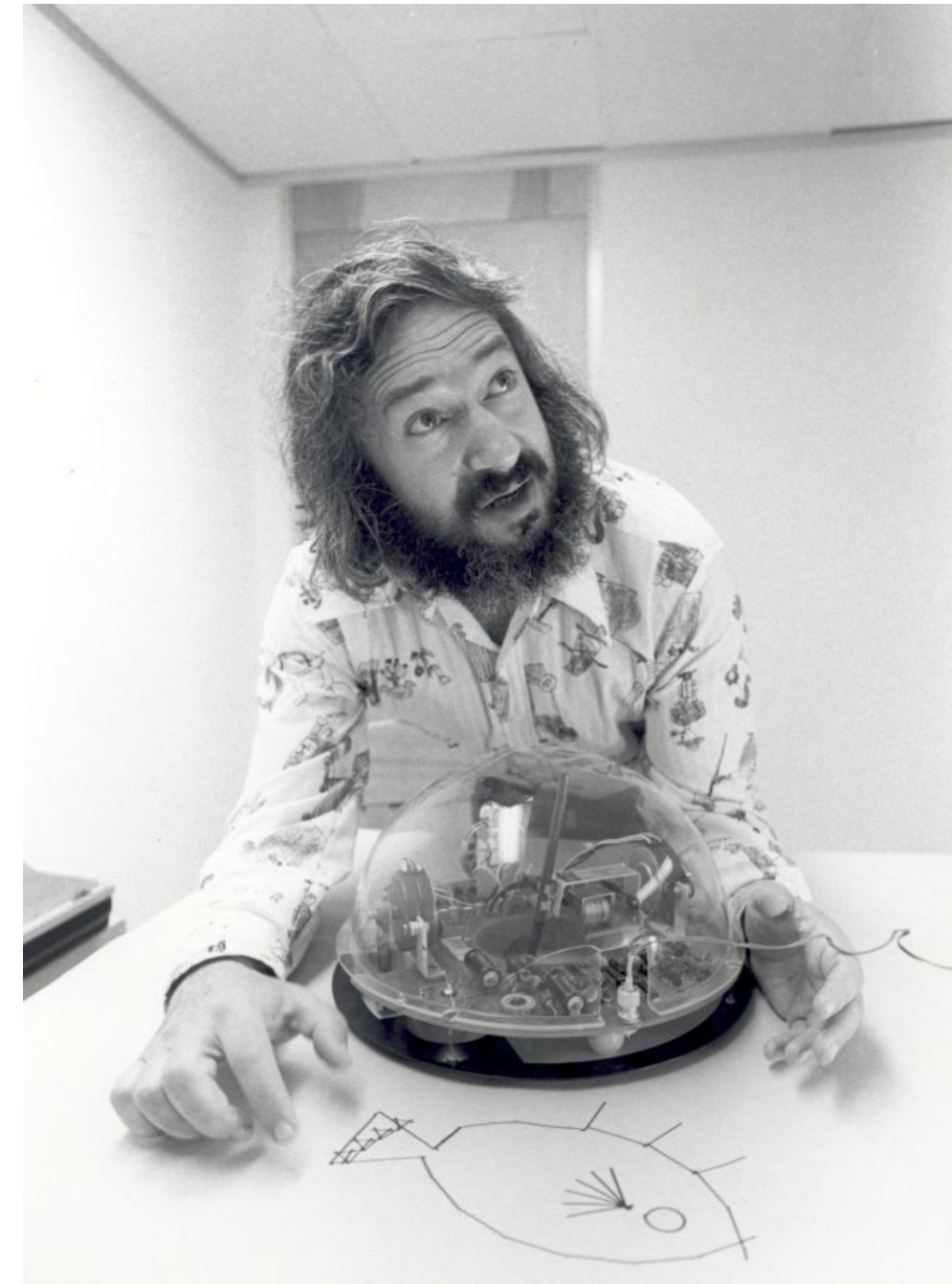
TO NAME
  PRINT "WHAT IS YOUR NAME?"
  READ
  PRINT "I'M GLAD YOU LIKED"
  PRINT "REQUEST"
  STOP
END

TO TIME
  PRINT "WHAT TIME IS IT?"
  READ
  PRINT "I'M LATE!"
  PRINT "SEE YOU LATER."
  STOP
END
```

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 Basic Menu Bar Help
colors.bas
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 accepted.
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 px
NEXT py
END SUB

SUB Speak(x,y, e$, f$) STATIC
IF f=0 THEN f = FREEFILE: OPEN "UOX" 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

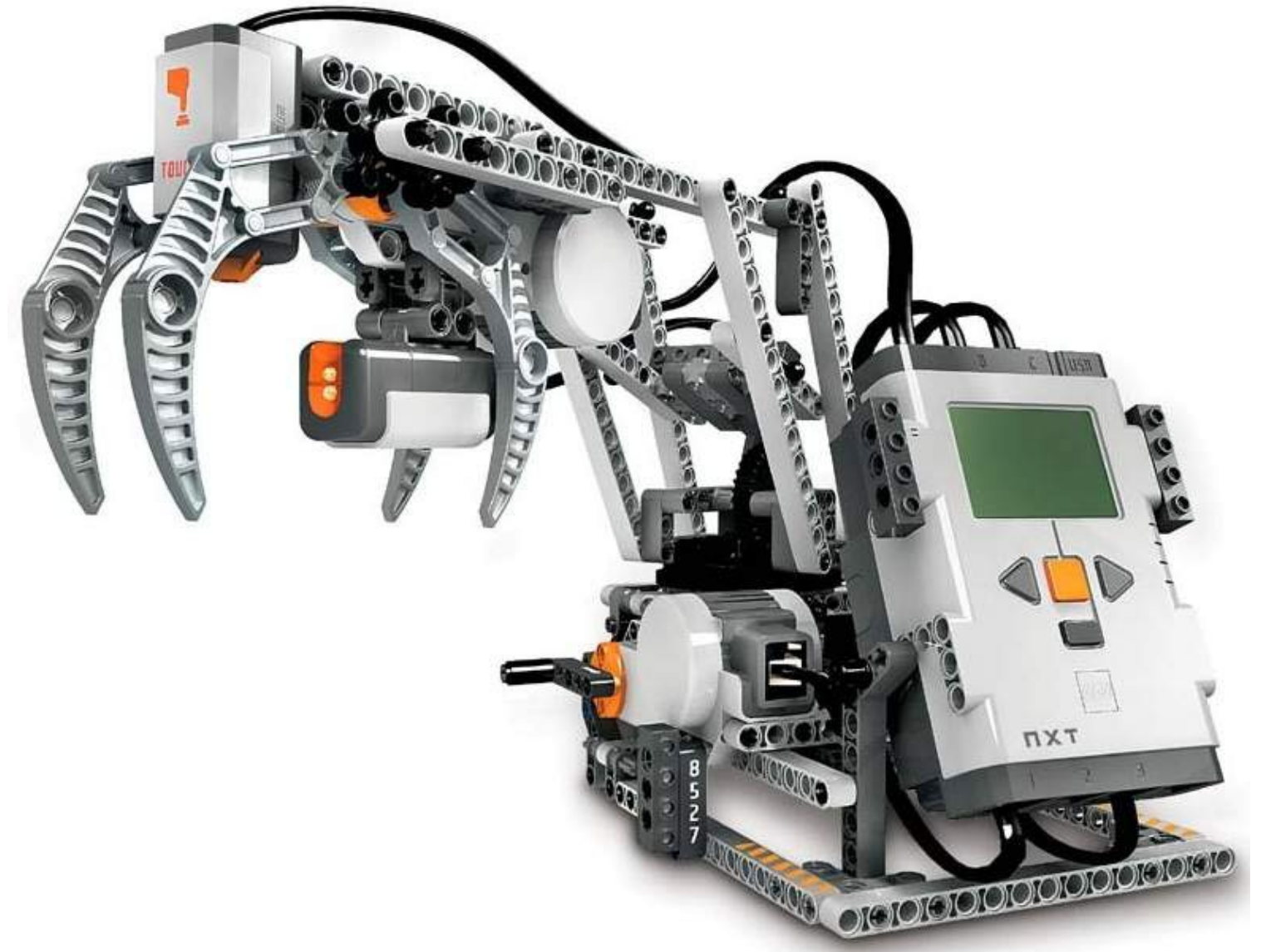
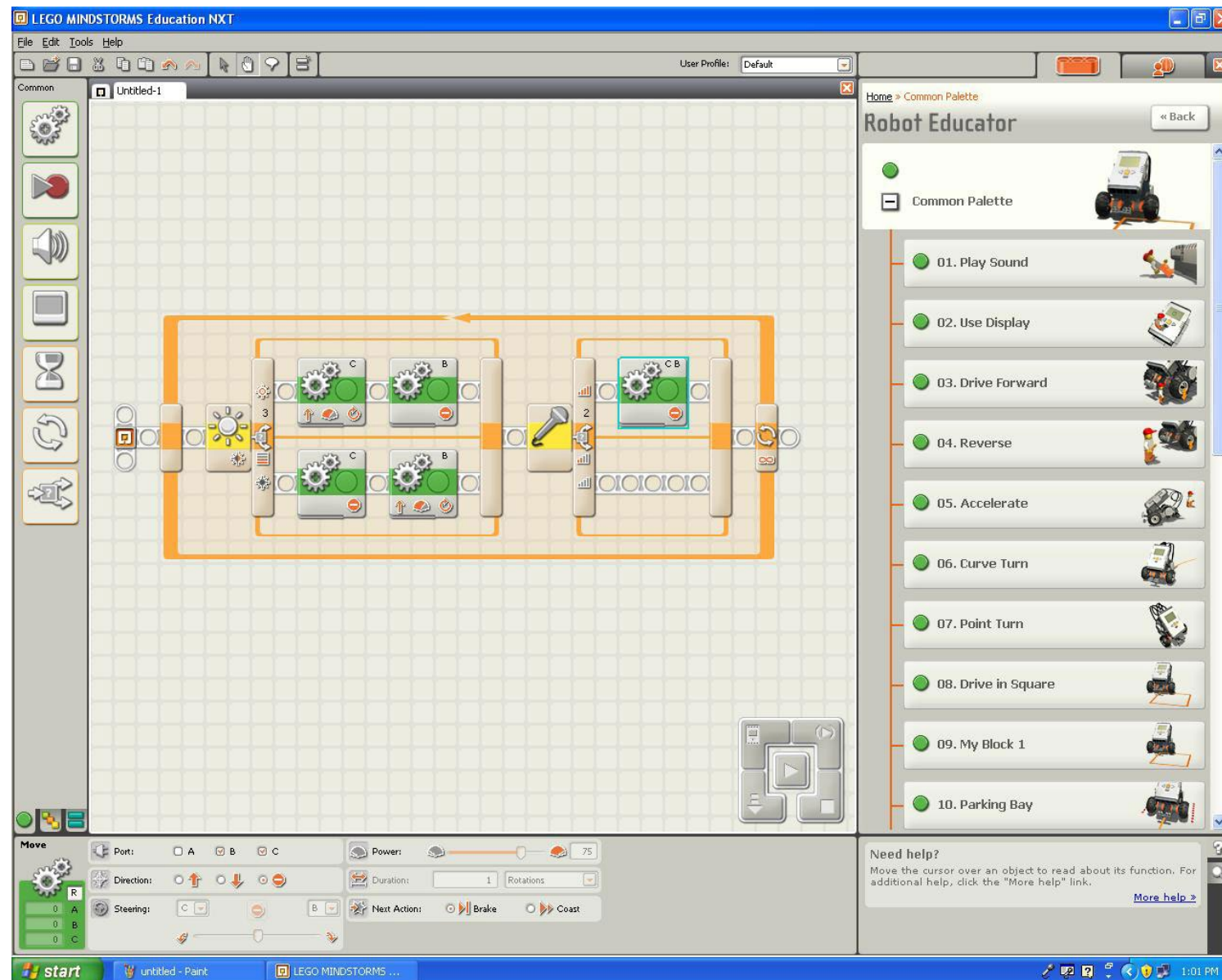
1998-2006



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

# LEGO Mindstorms NXT

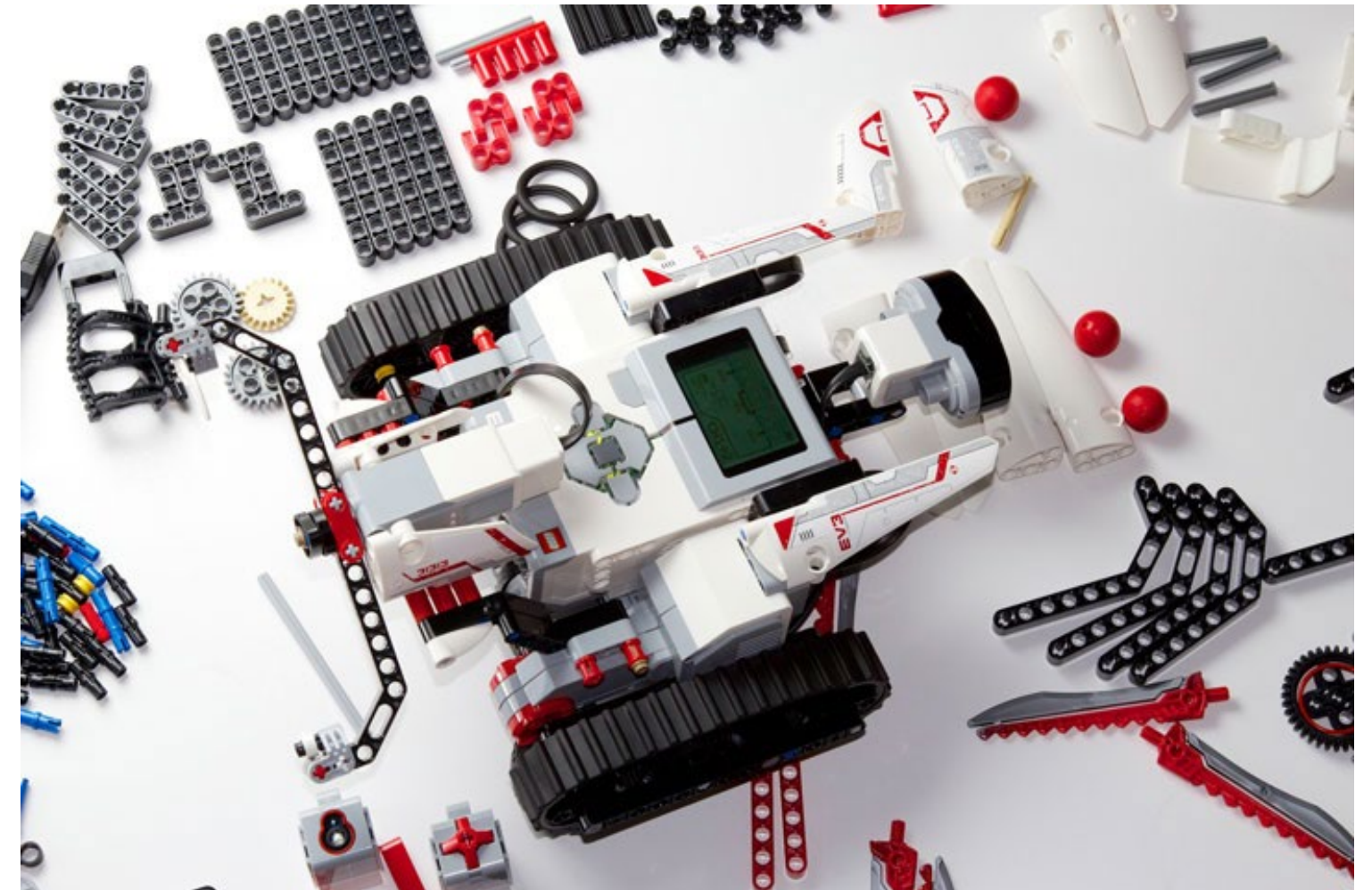
2006-2013



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

# LEGO Mindstorms EV3

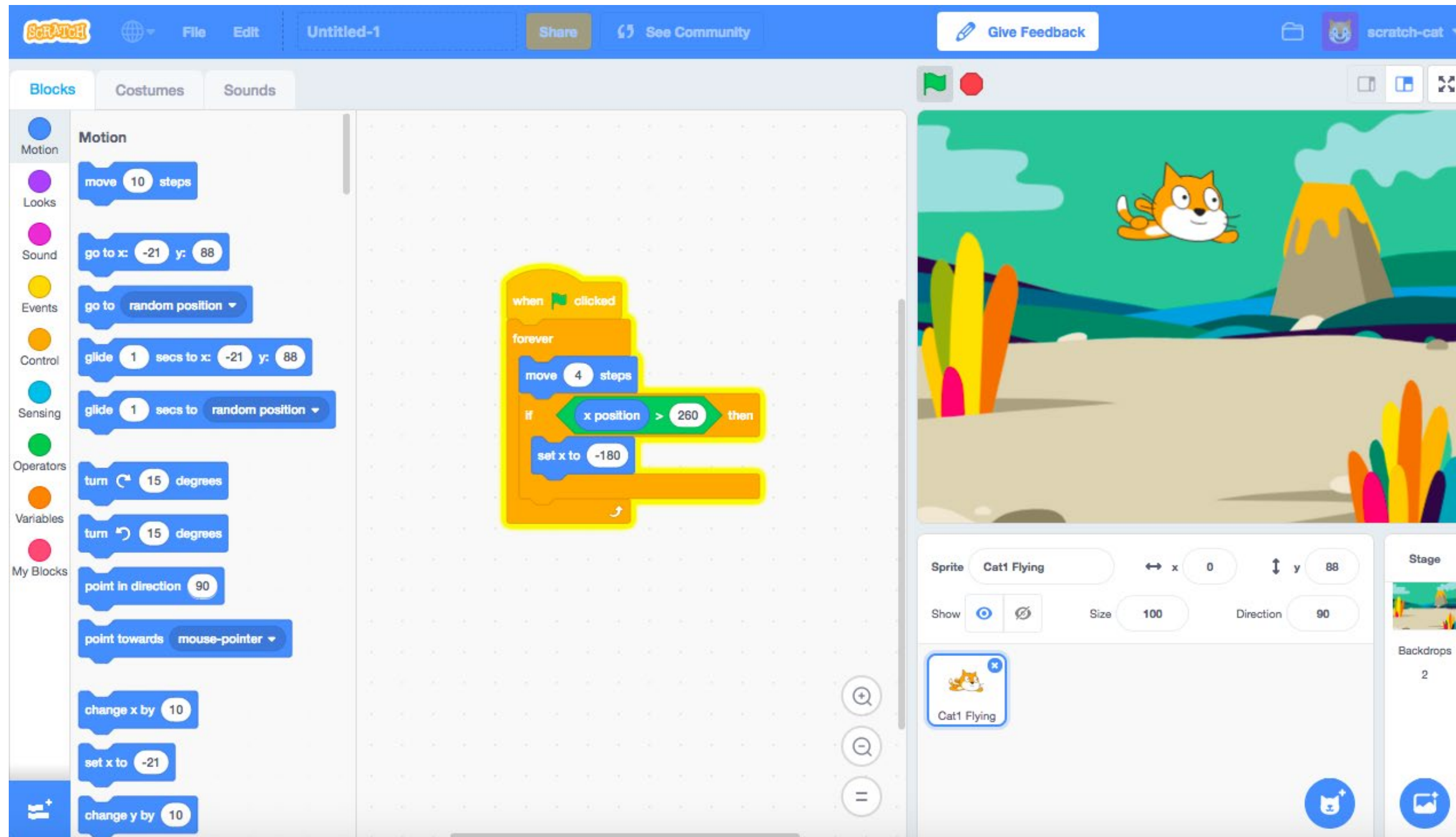
2013-Today



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

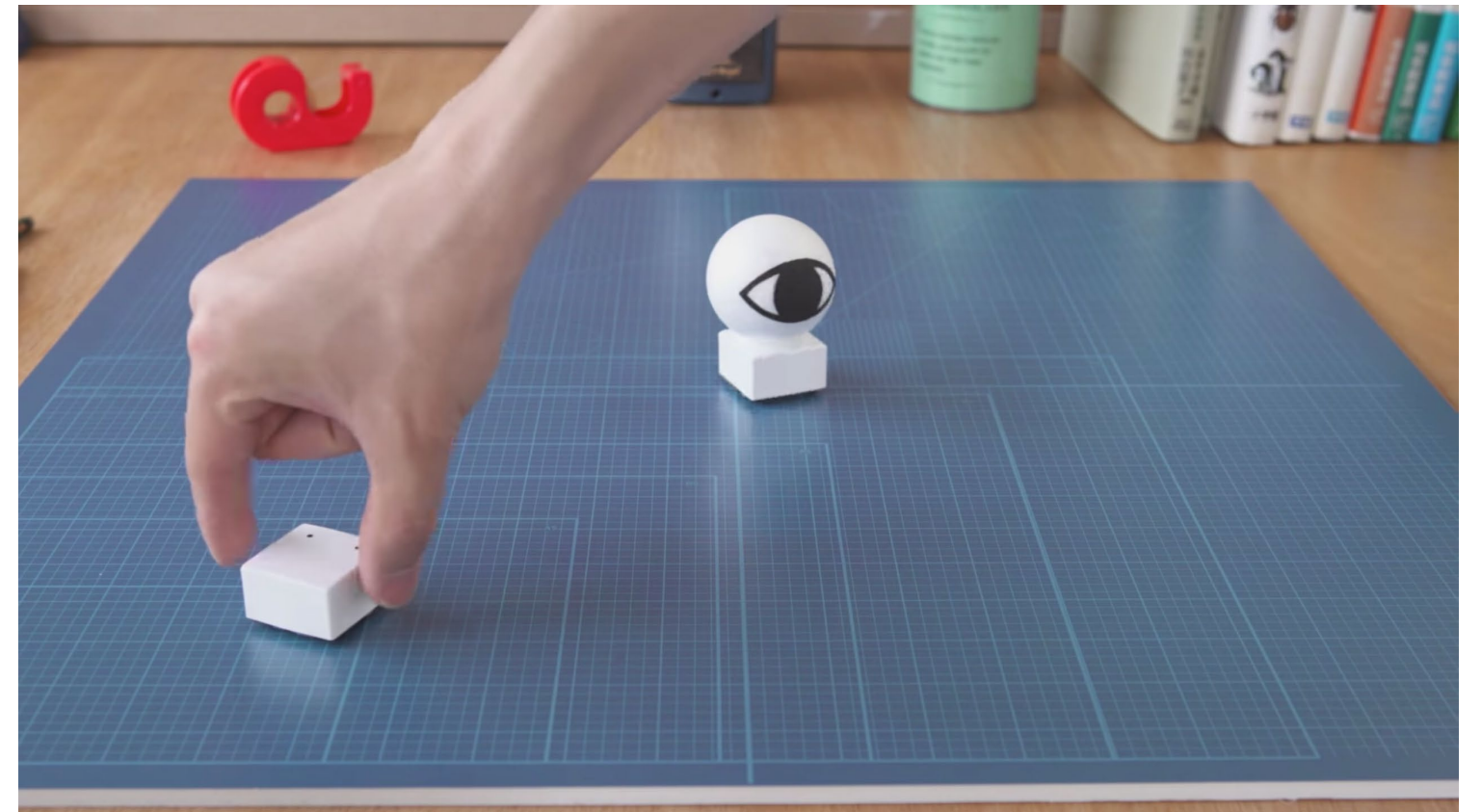
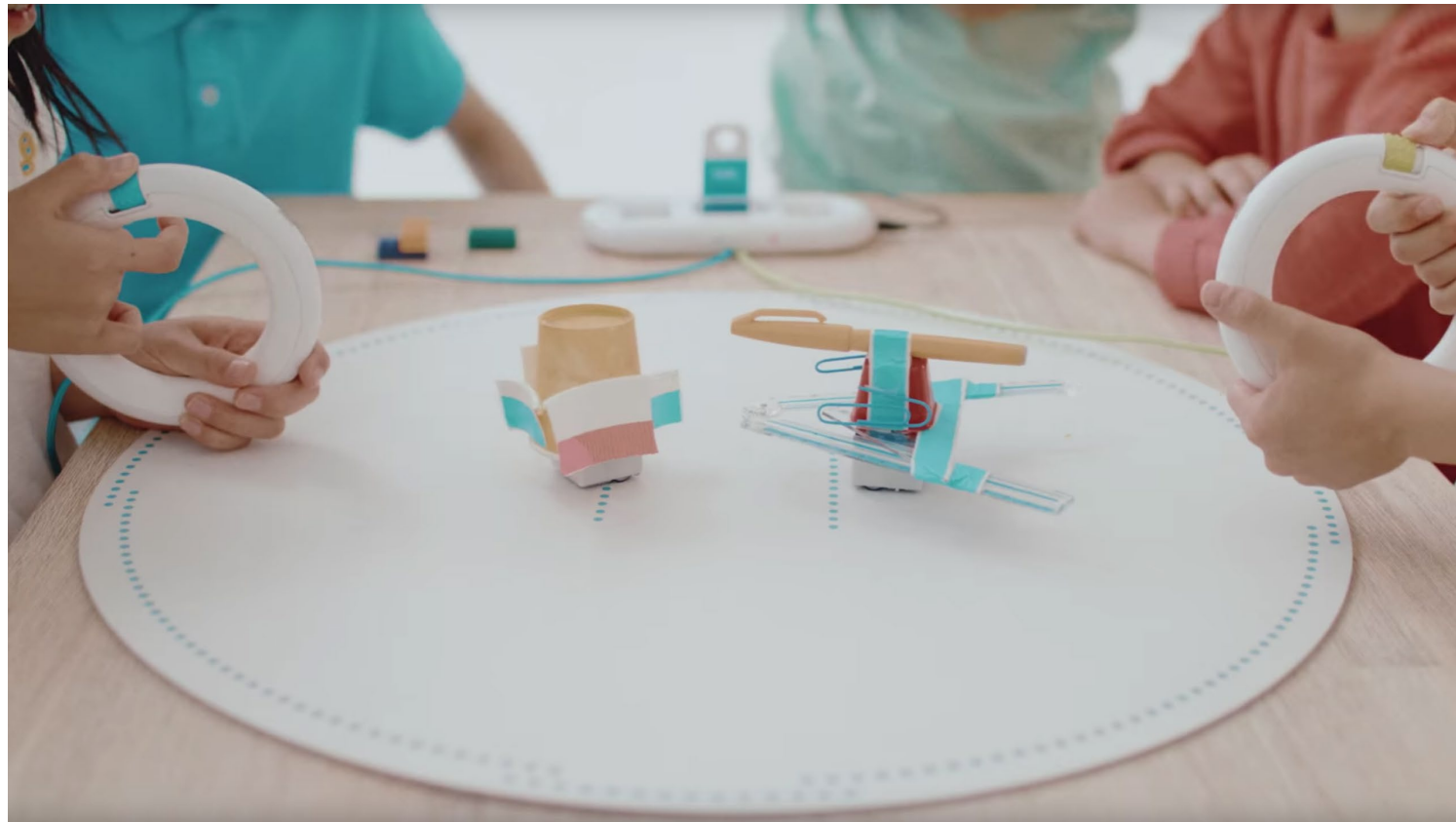
# 2017-Today



Sony

# Toio

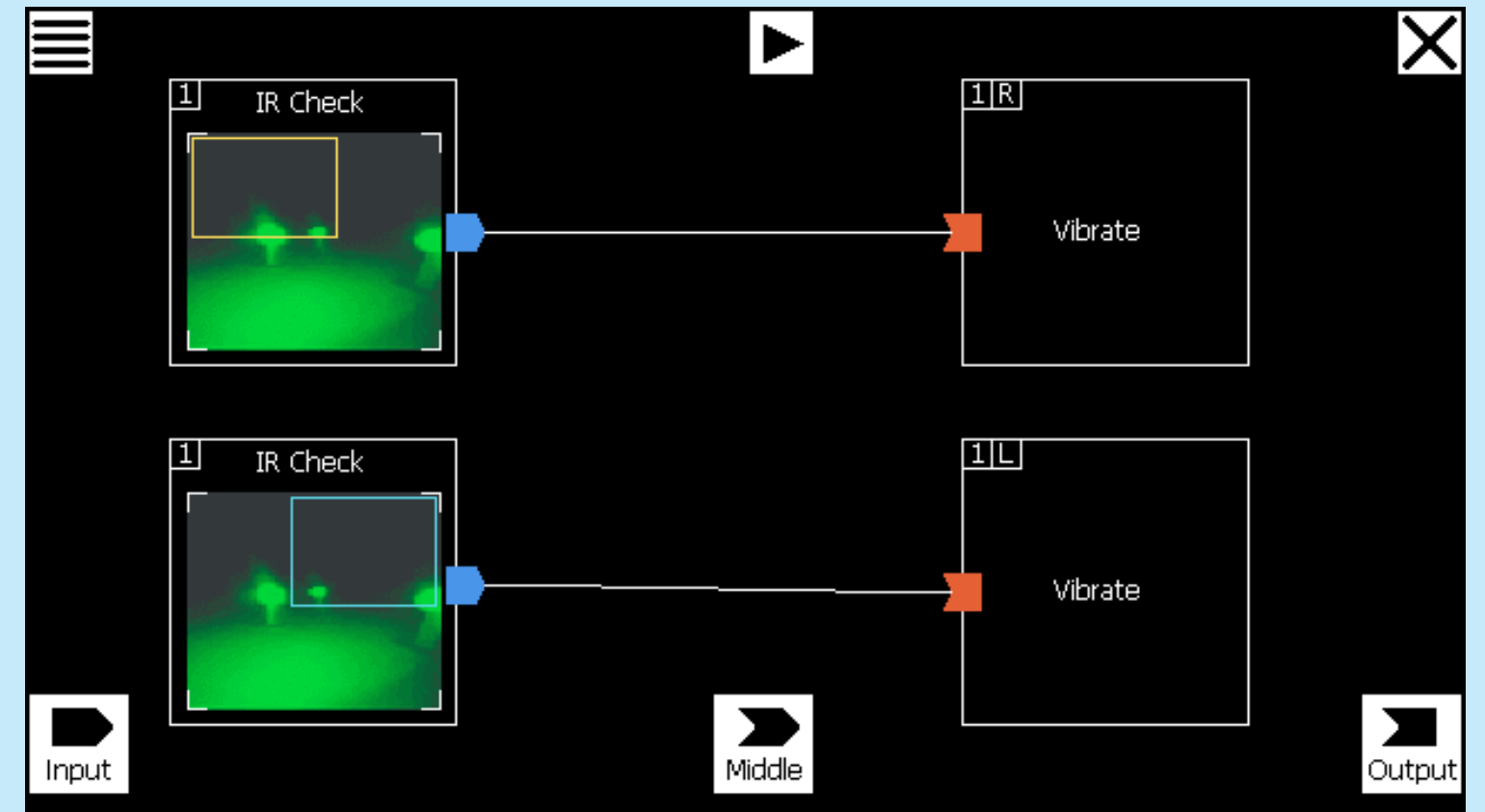
# 2017-Today



Sony

# Joy-Con Garage

2018-Today



Nintendo

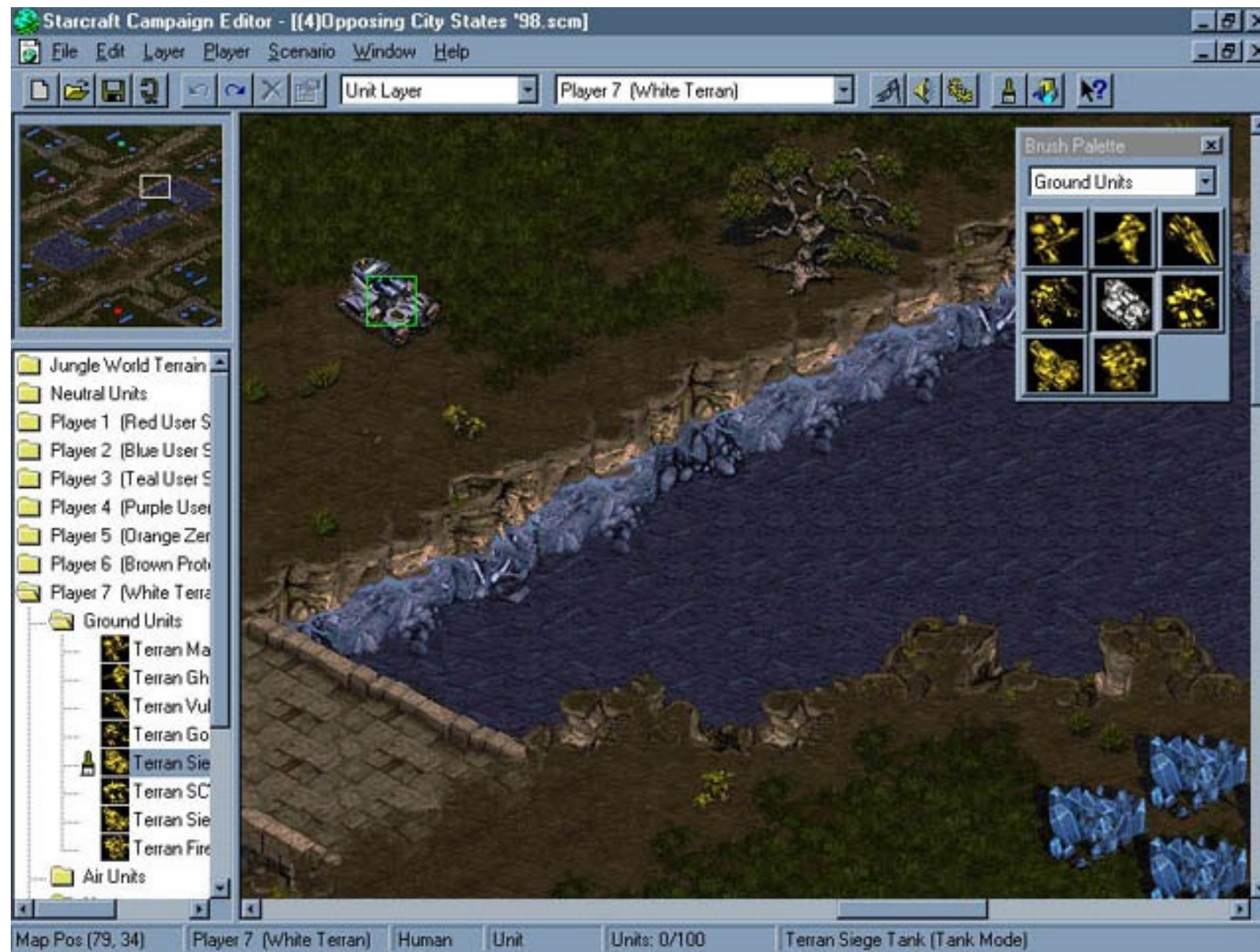


# Game Authoring

From video games to VR/AR/MR

# StarEdit

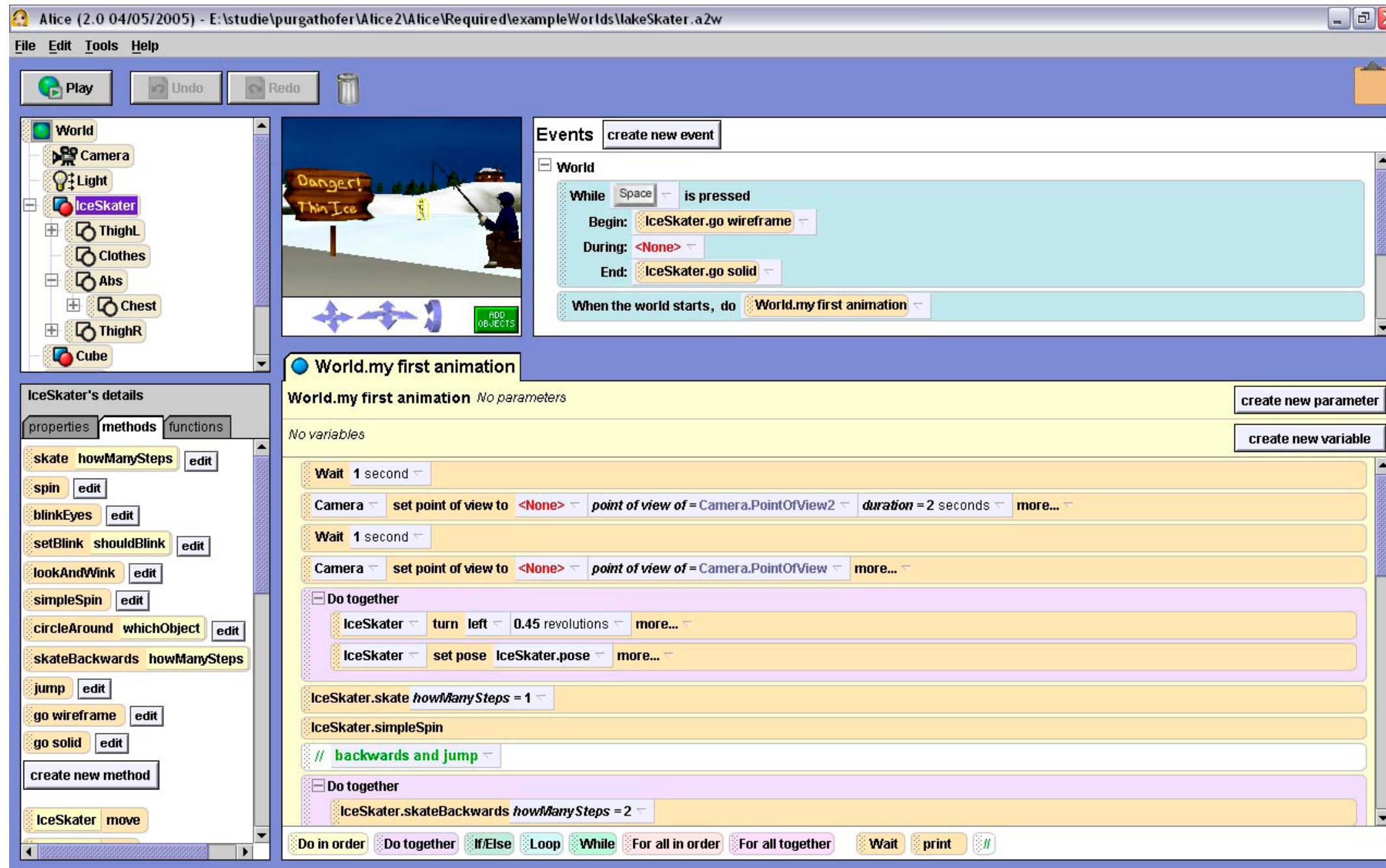
1998



Blizzard Entertainment

# Alice

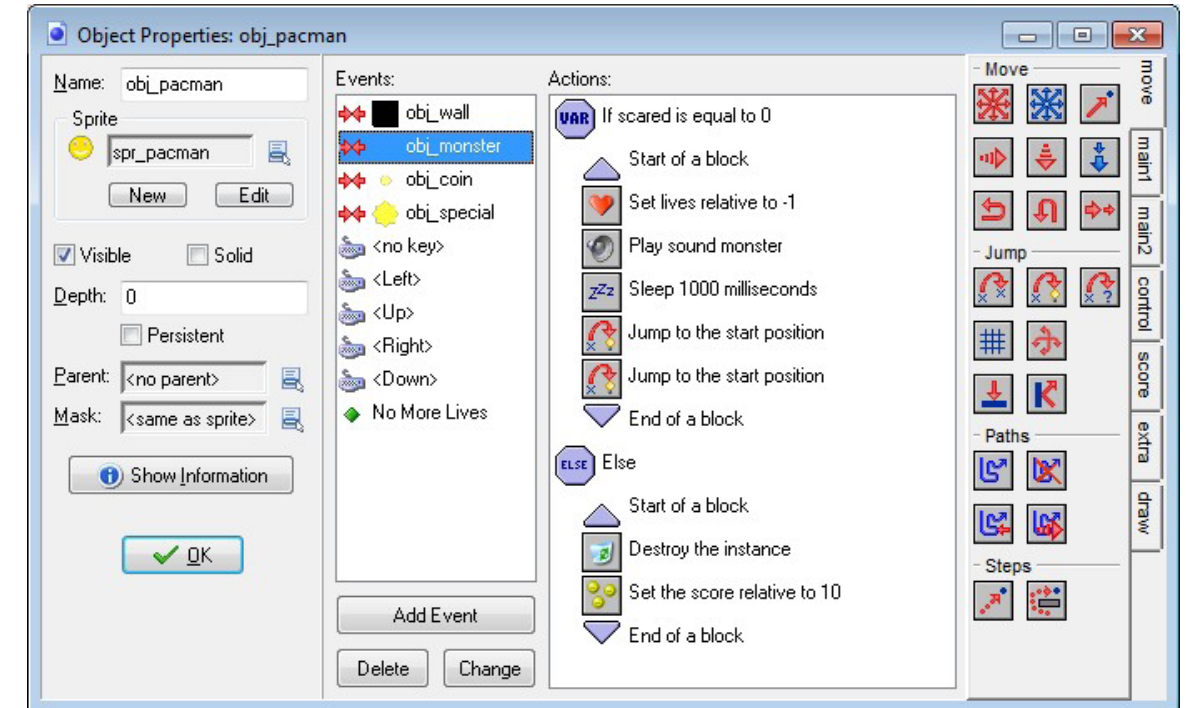
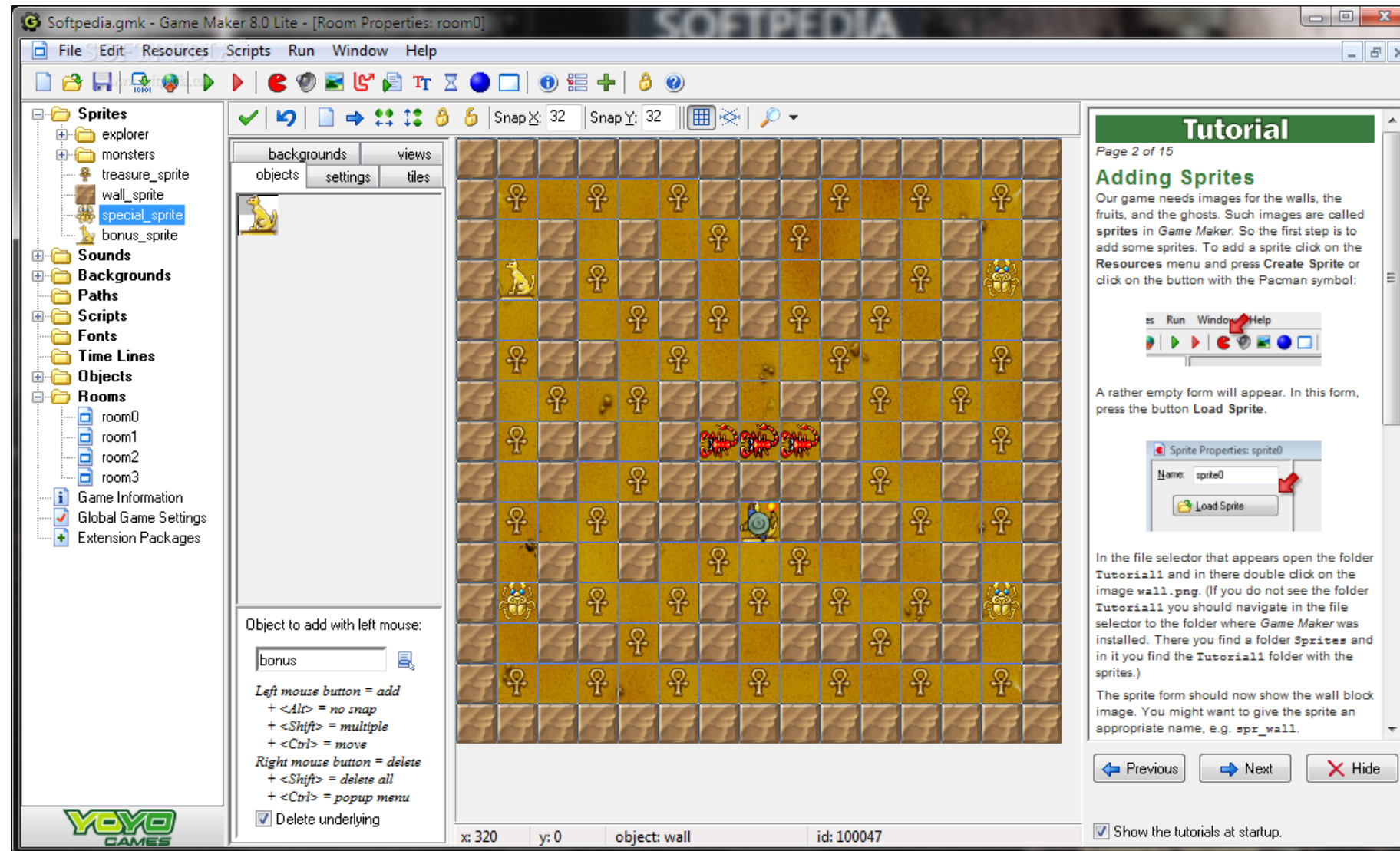
1998-Today



Randy Pausch and Others at Carnegie Mellon University

# GameMaker

# 1999-Today



Mark Overmars

# Unreal Engine

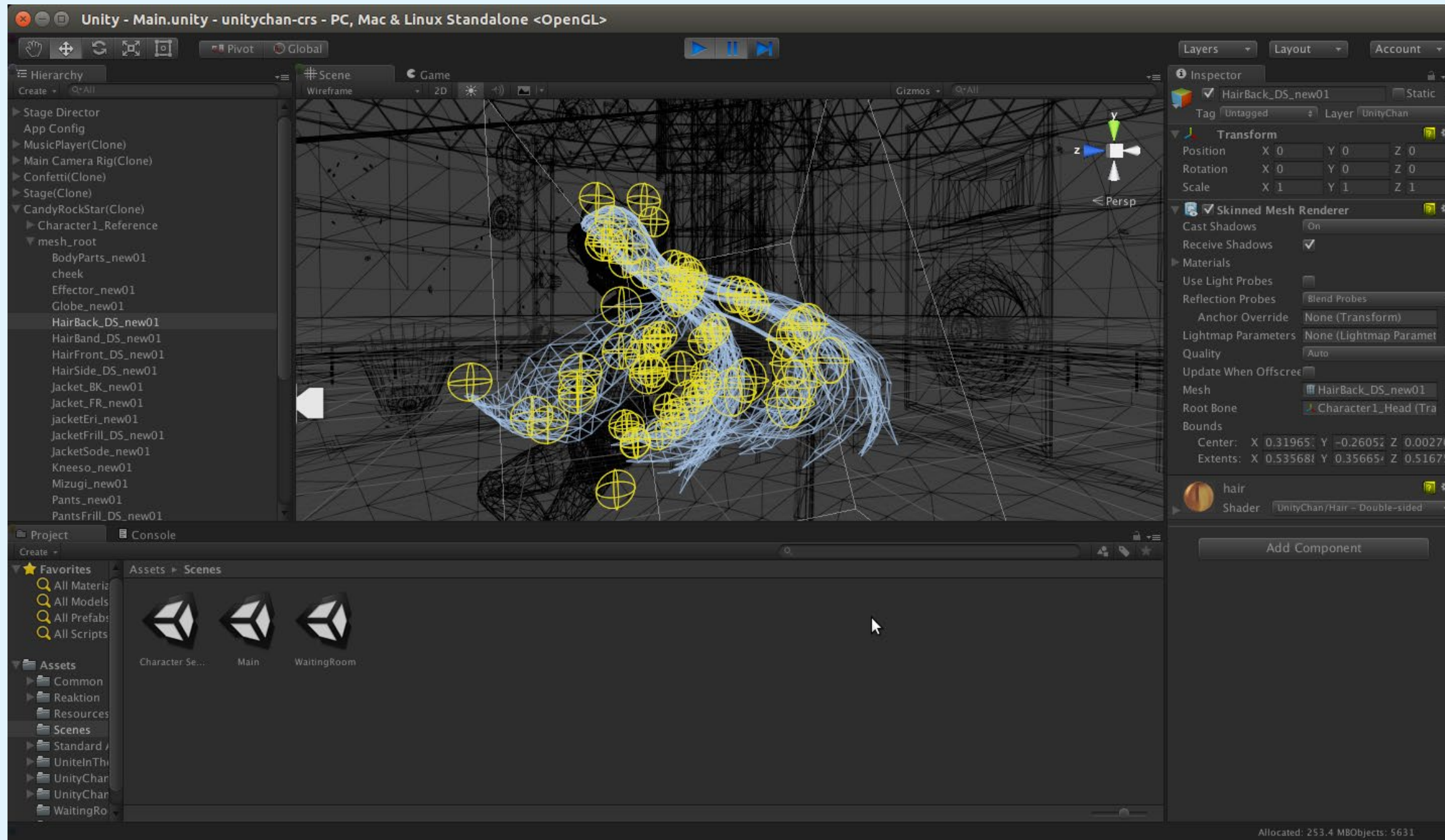
1998-Today



Epic Games

# Unity

# 2005-Today



## Unity Technologies

# Source Filmmaker

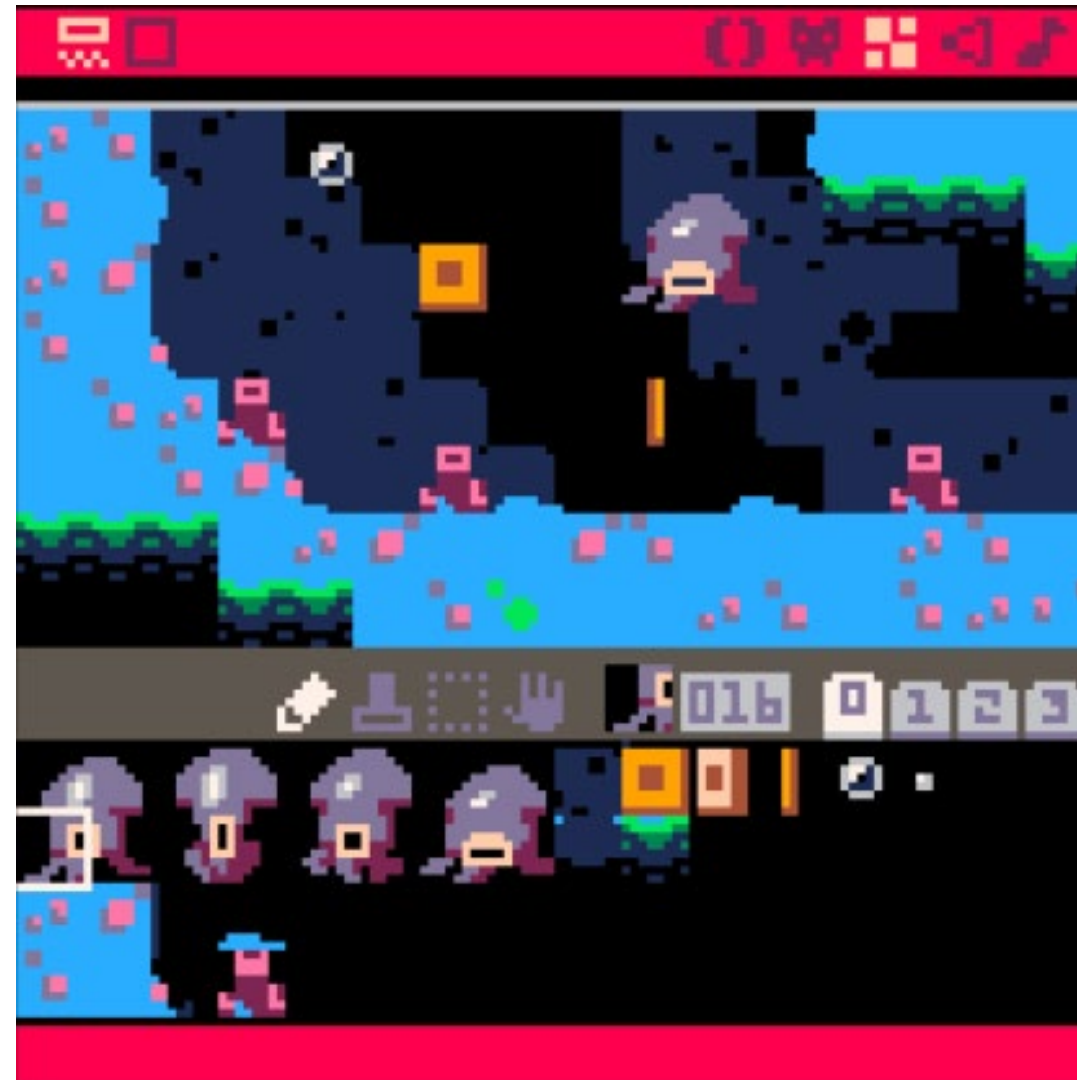
# 2012-Today



Valve Corporation

# Pocket Chip for PICO-8

2016-2018



```
1)
= 64
= 64
IR = 1

TICK _INIT()
()

TICK _UPDATE()
P1.DIR == 1) THEN P1.Y=P1.Y-1 EN
P1.DIR == 2) THEN P1.X=P1.X+1 EN
P1.DIR == 3) THEN P1.Y=P1.Y+1 EN
P1.DIR == 4) THEN P1.X=P1.X-1 EN

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 19/26 177/8192
```

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

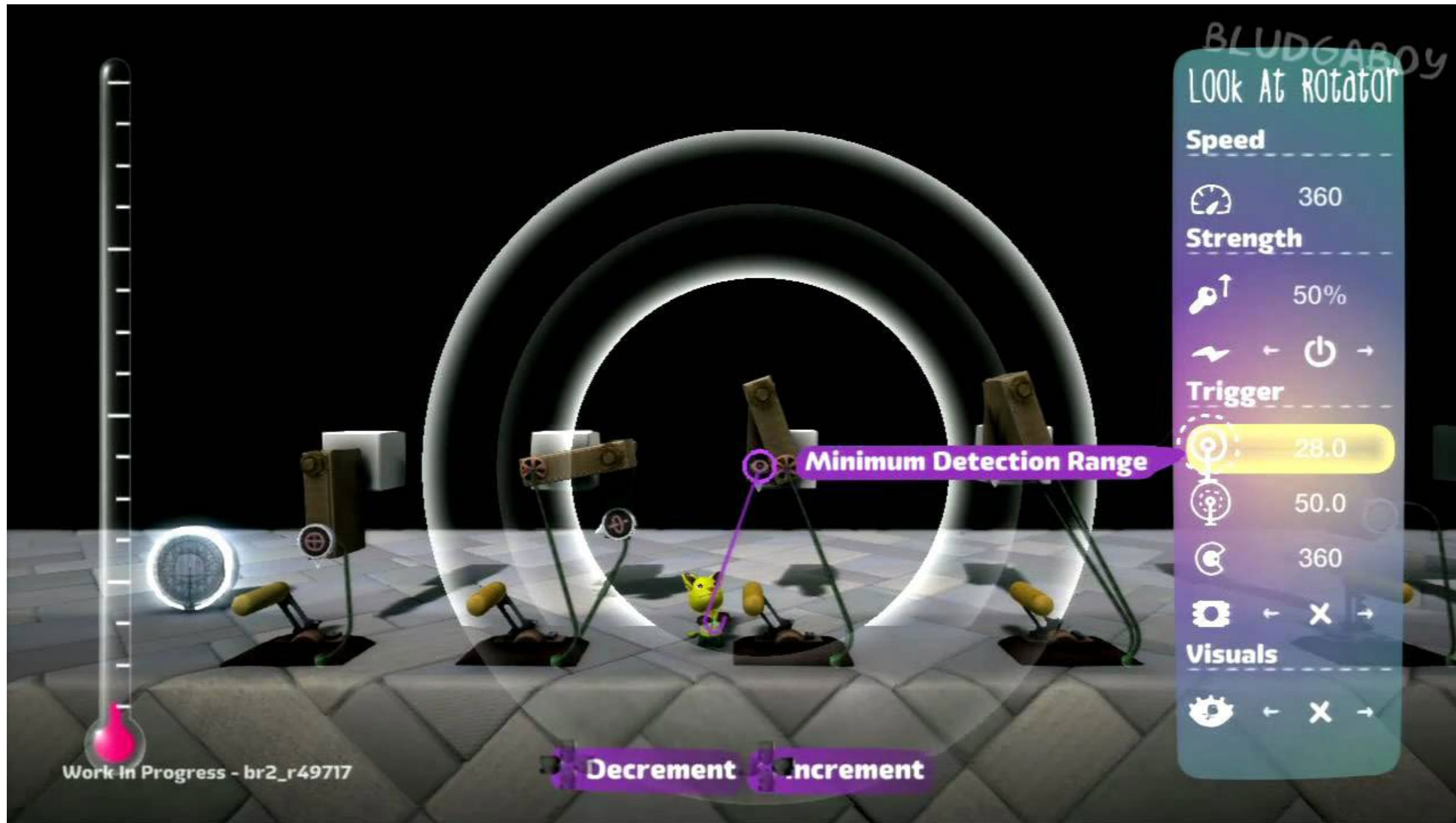
2008



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)

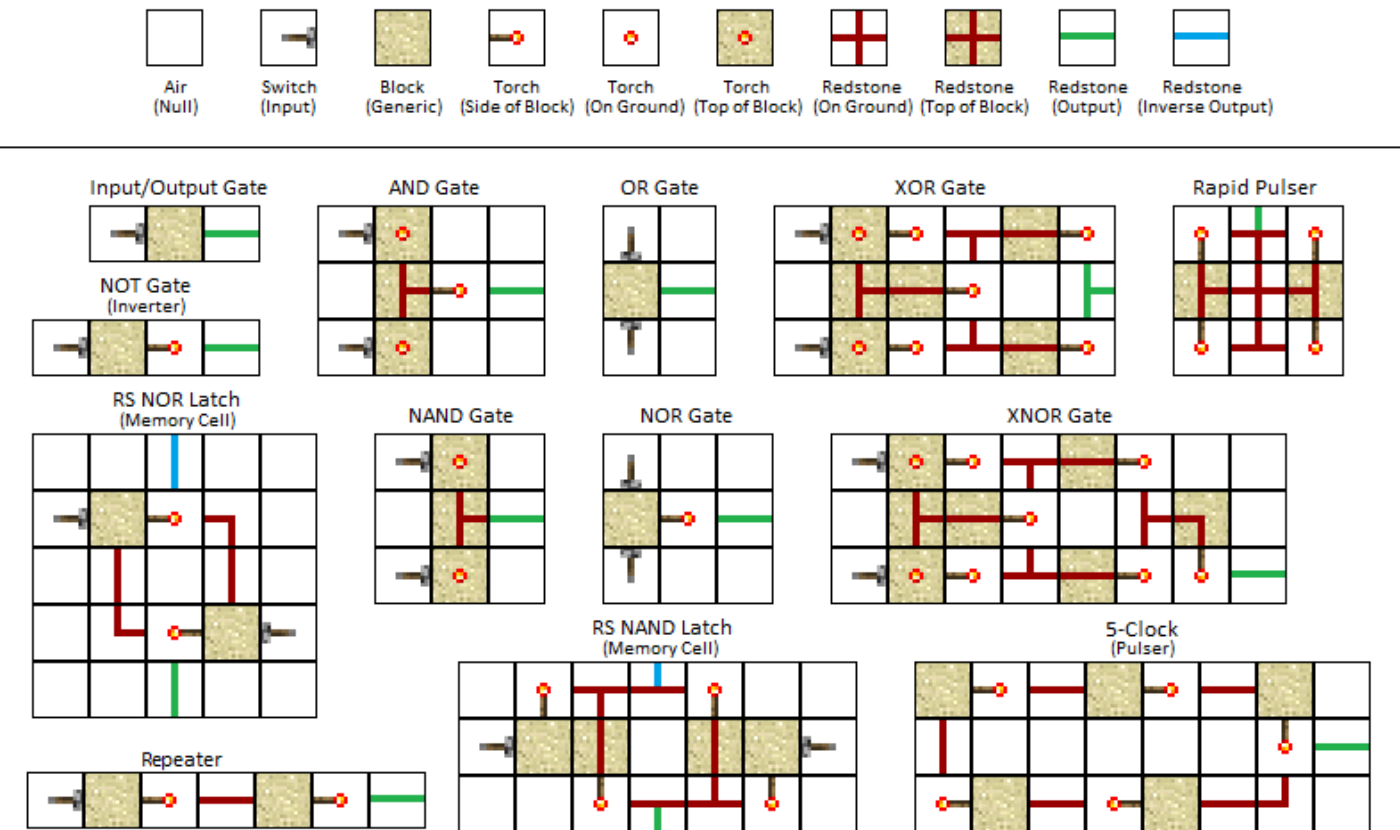
1993-2011



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

# Minecraft

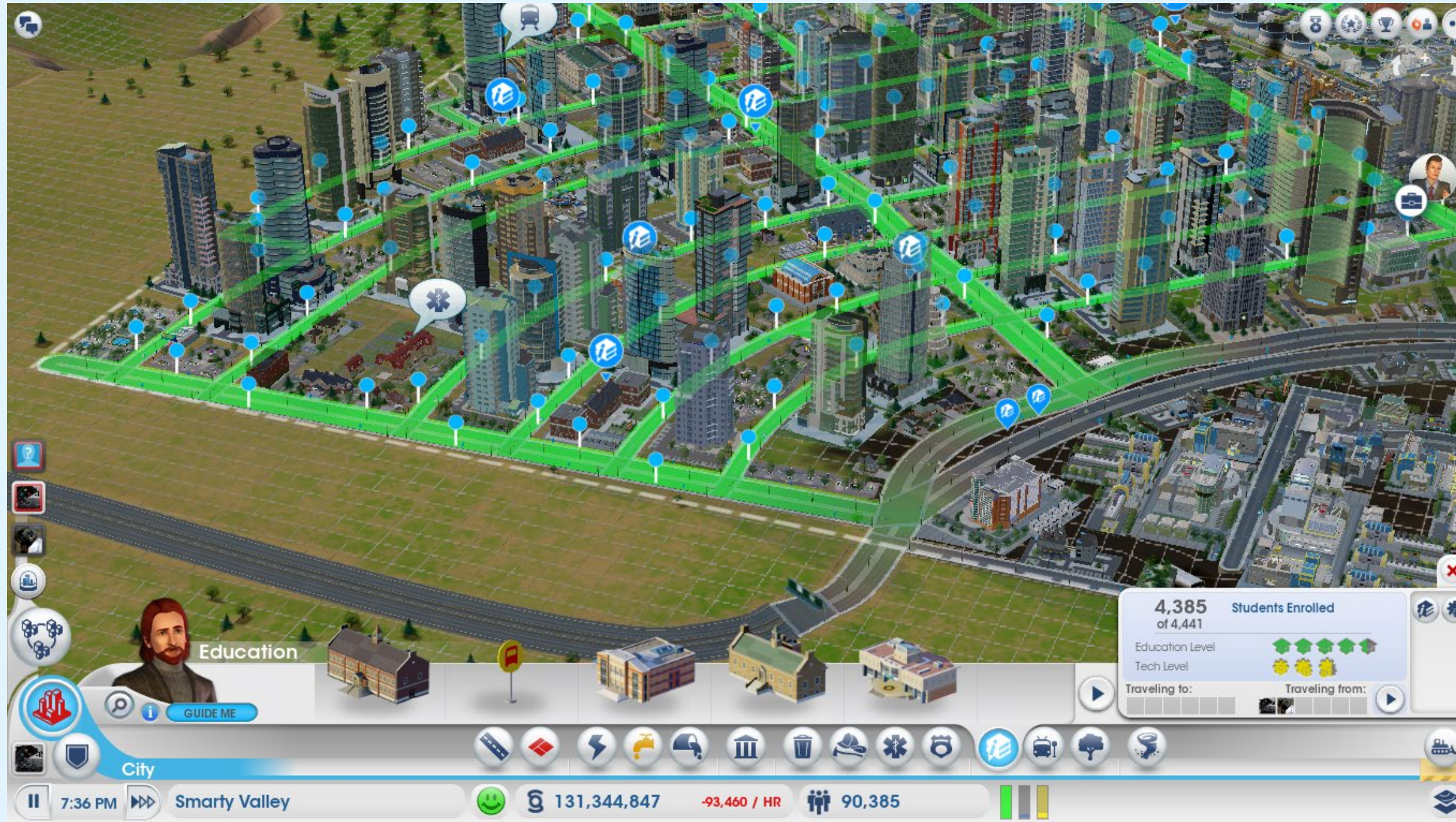
# 2009-Today



Markus Persson, Jens Bergensten at Mojang

# SimCity (Series)

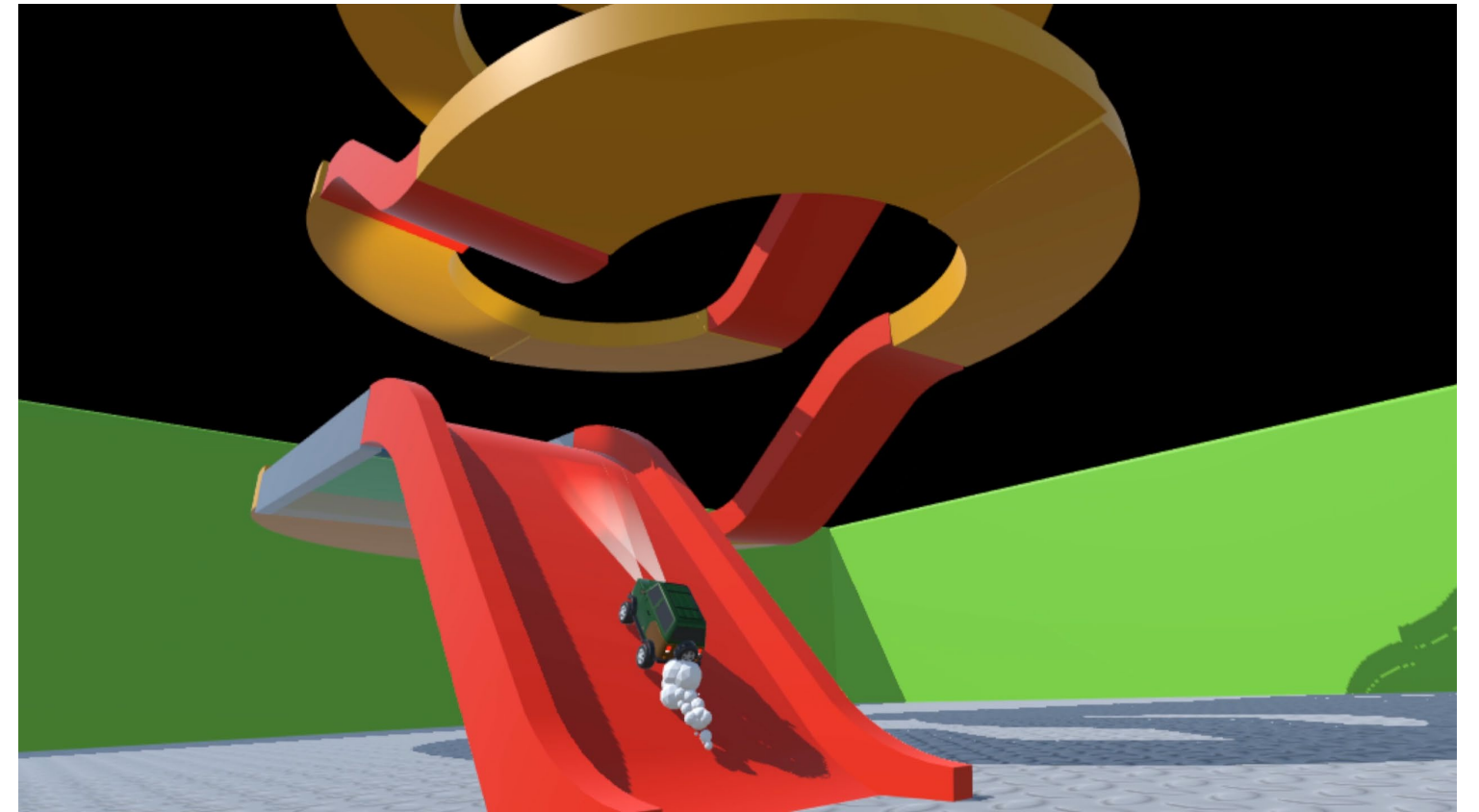
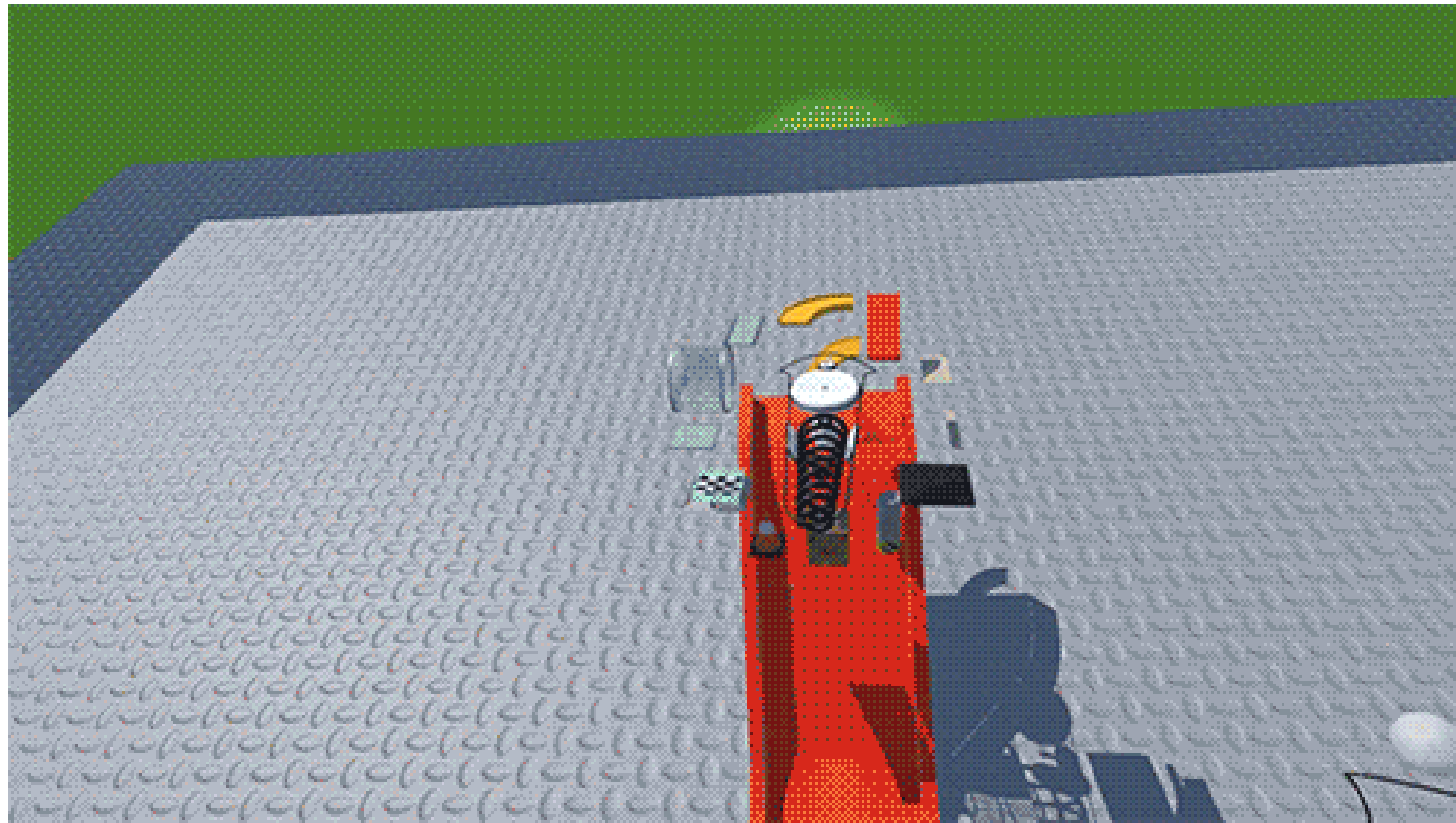
1989-2014



Various

# Tiny Wheels

2017



Robbie Tilton

# Dreams

2018



Media Molecule

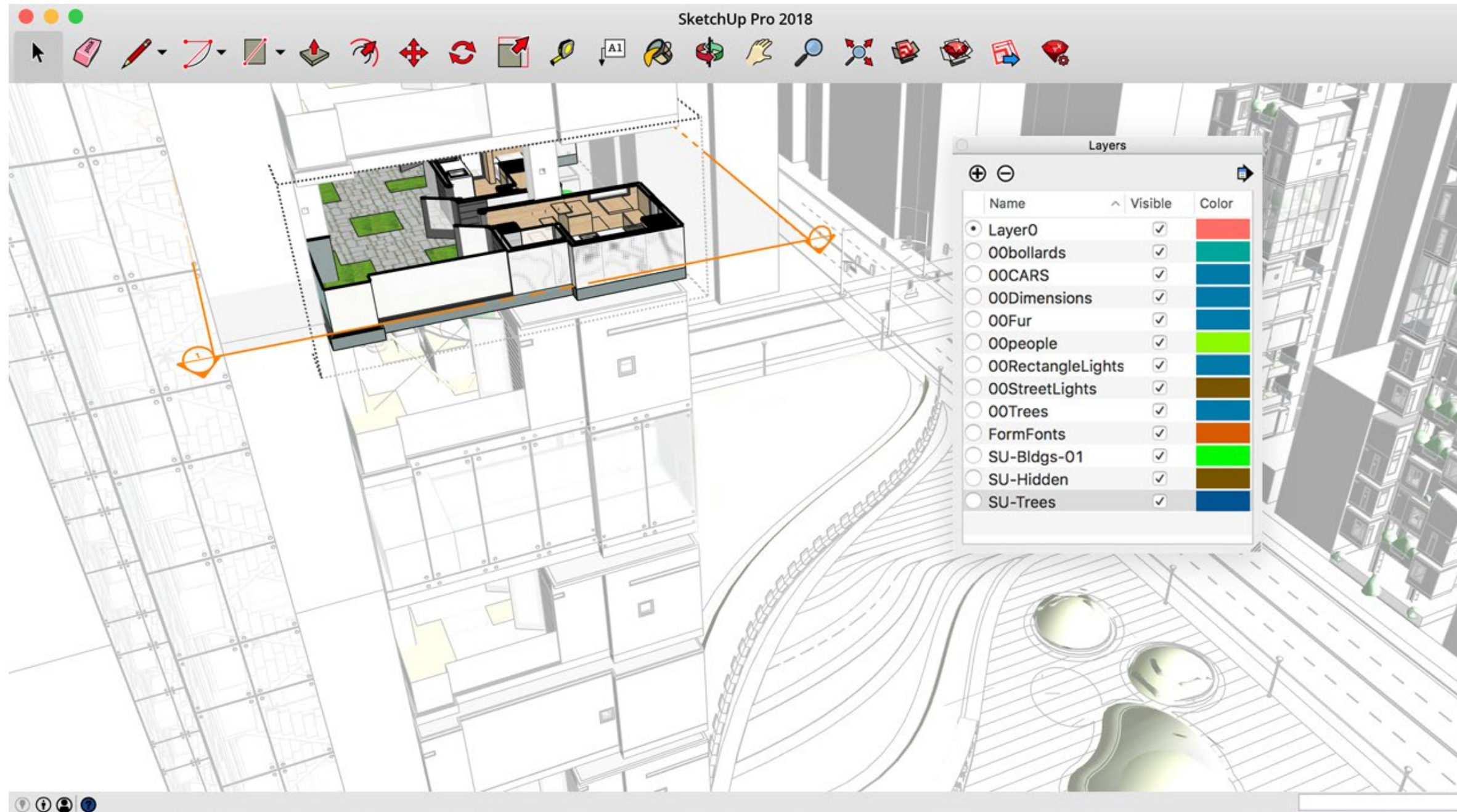
# Remaking Reality

Measure, Simulate, Predict & Act

# Reality Design

# SketchUp

# 2000-Today

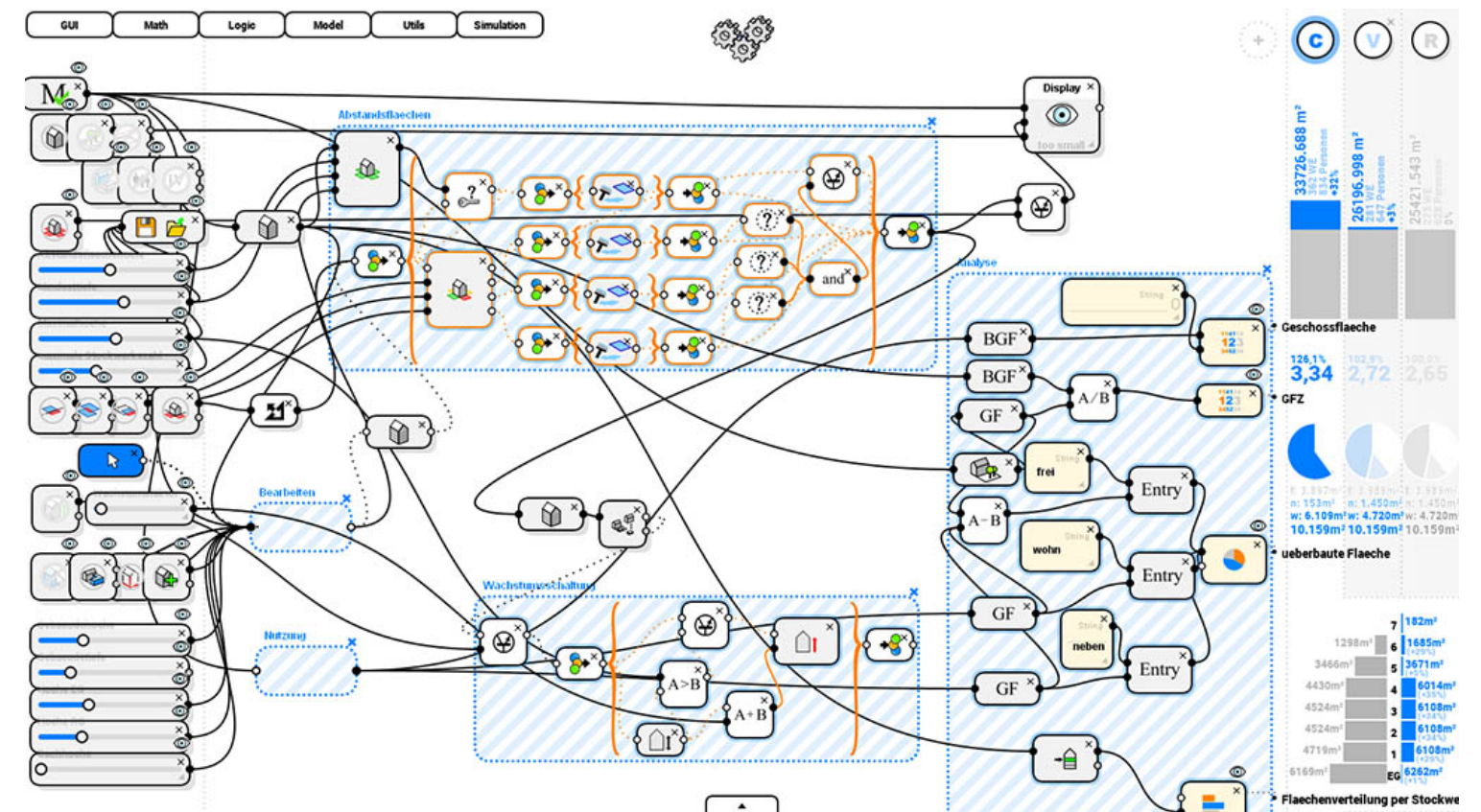
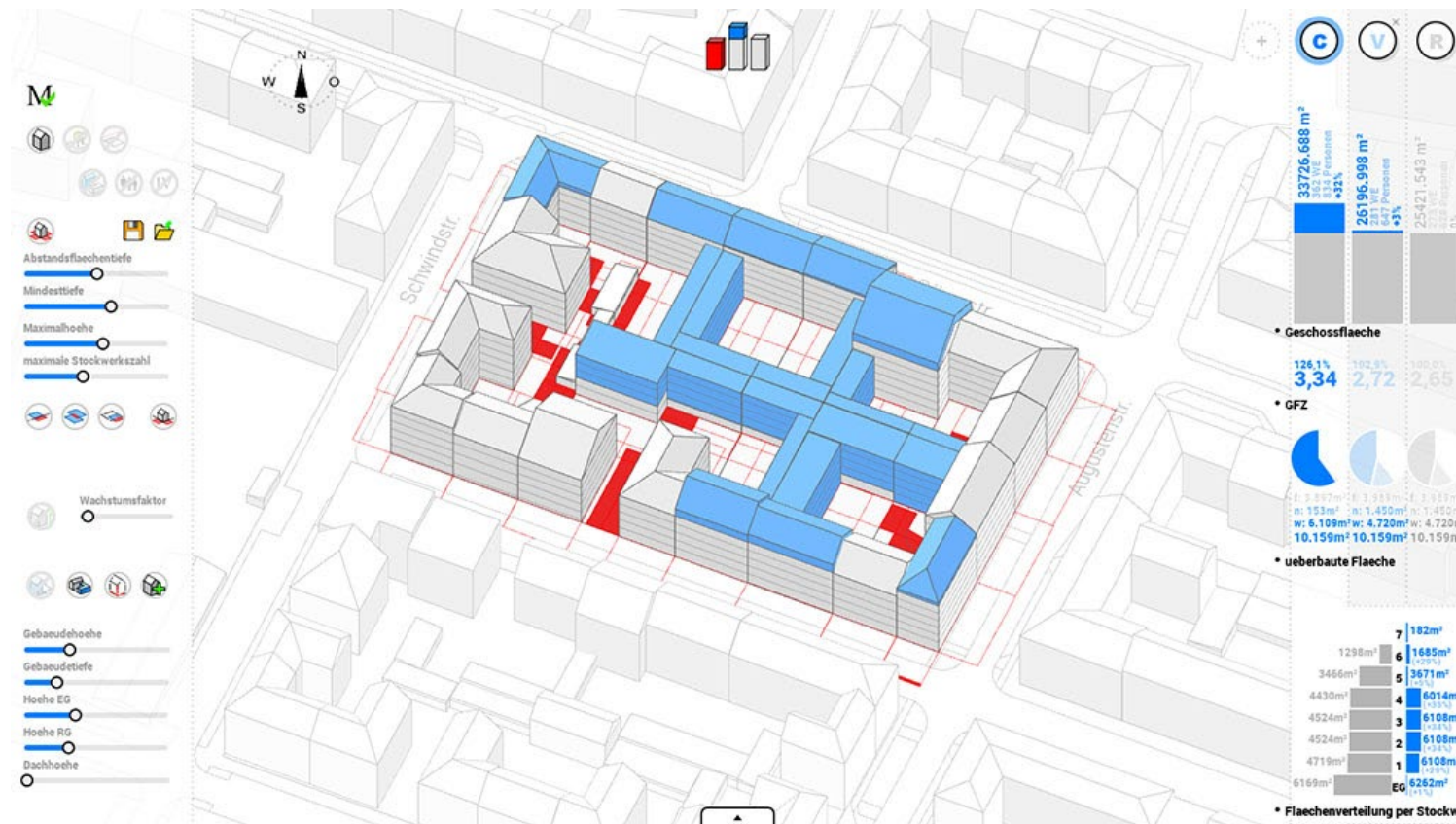


Brad Schell and Joe Esch for @Last Software (Later Google, Trimble Inc)



# Urban Strategy Playground (Suite)

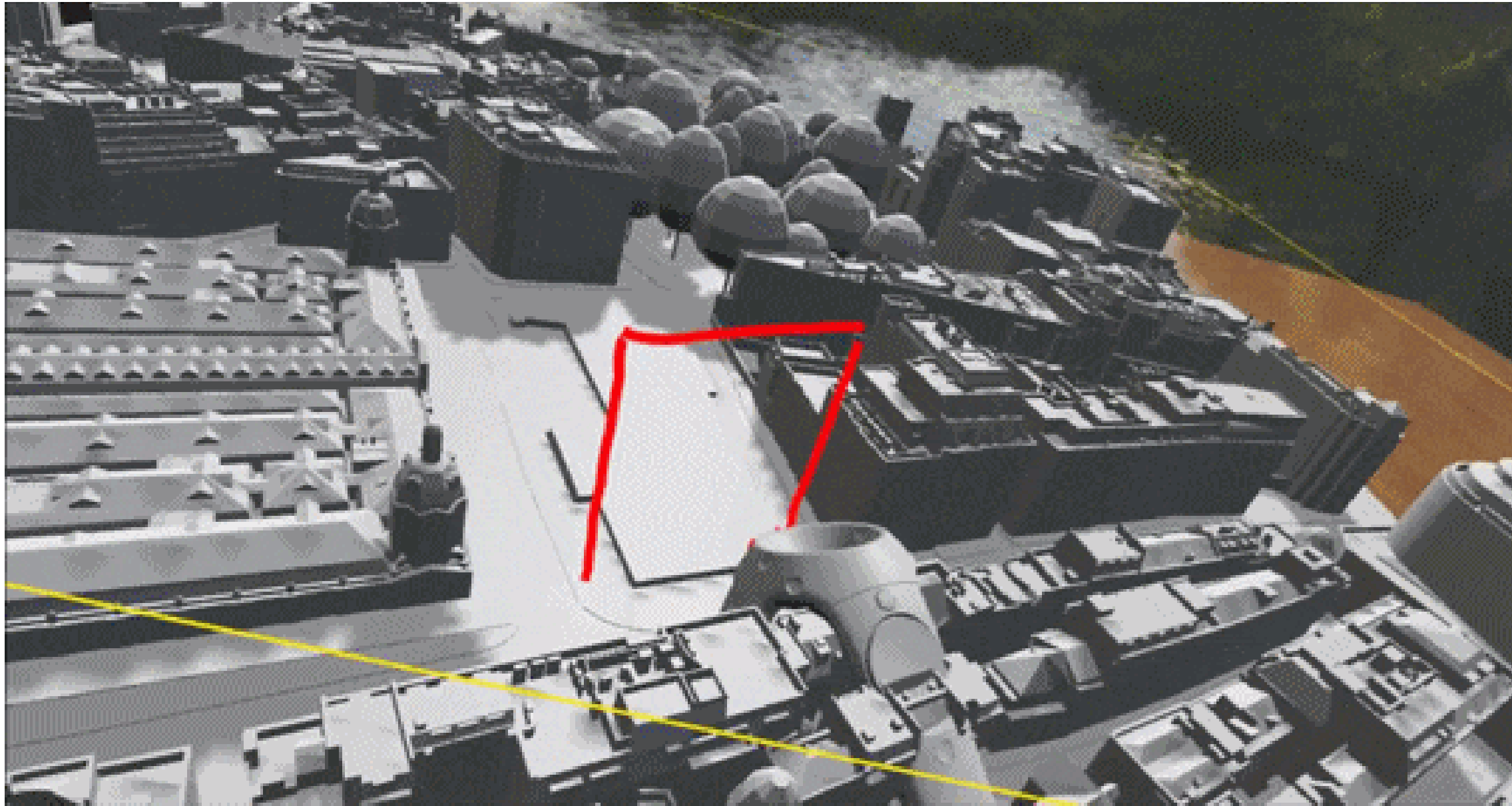
# 2013-Today



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

# DesignSpace

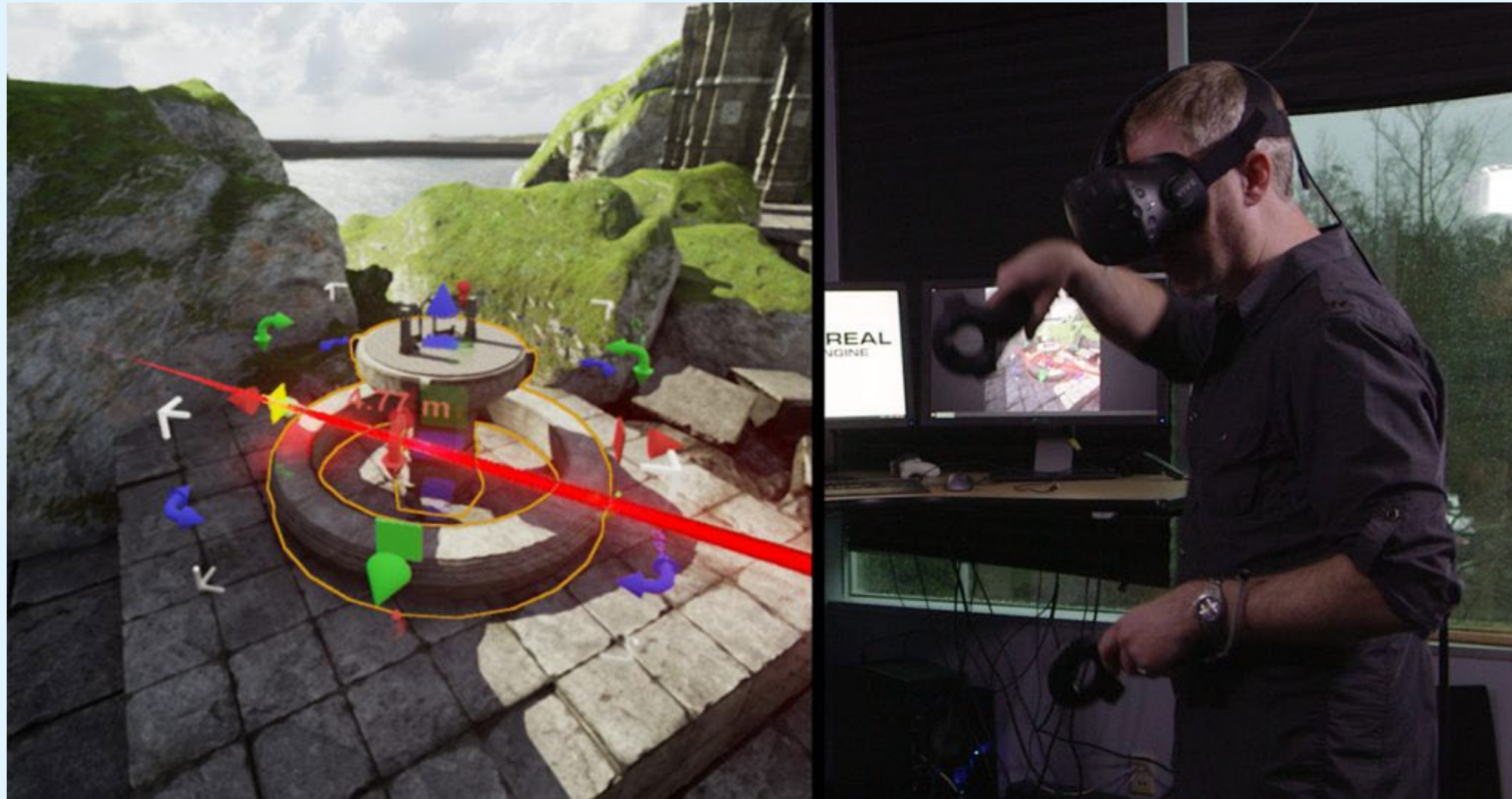
2016



Thomas Van Bouwe

# VR Editor for Unreal Engine

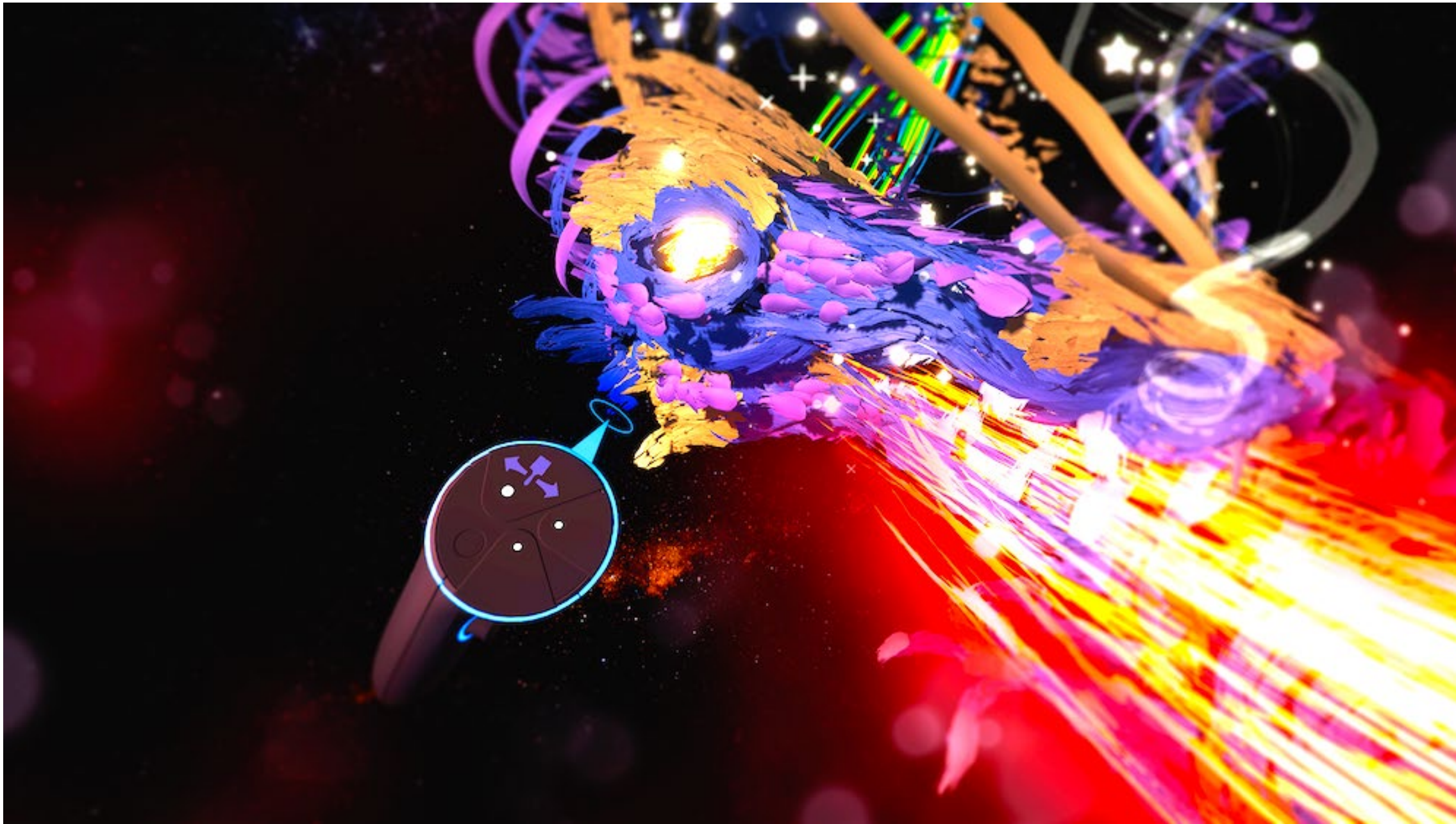
2016-Today



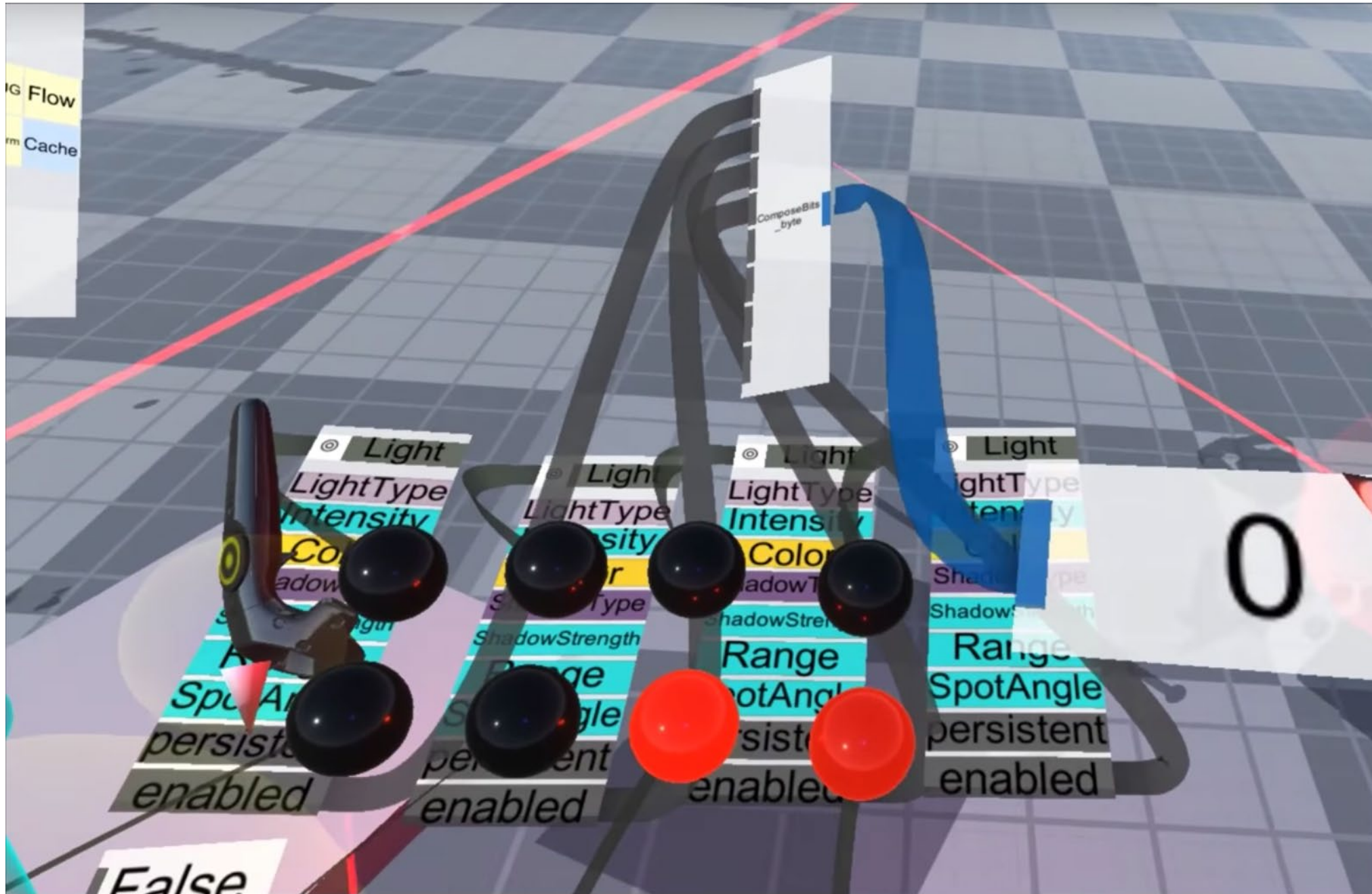
Epic Games

# TiltBrush

2016-Today



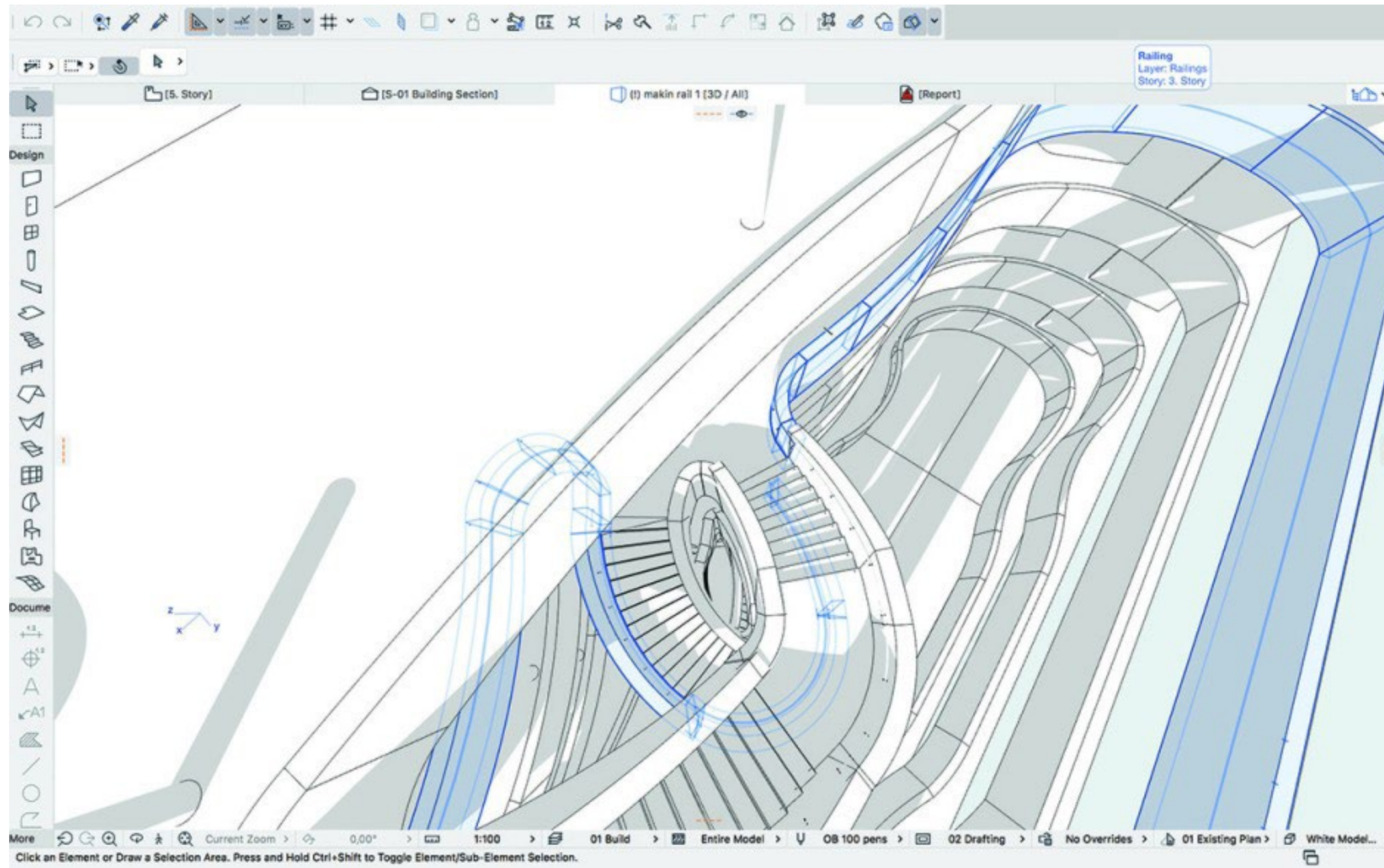
Google



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

The screenshot displays the CartoDB interface for a 'Site planning' project. The left sidebar contains a navigation menu with icons for home, edit, and share. The main content area is divided into a left-hand 'Layers' panel and a right-hand 'Widgets' panel. The map in the center shows a dark-themed map with several clusters of points, each represented by a central colored dot (green, yellow, blue) and radiating lines connecting to individual data points. The 'Layers' panel on the left lists three layers: 'Stores' (1 ANALYSIS, 0 WIDGETS), 'Customers to Stores' (1 ANALYSIS, 0 WIDGETS), and 'Customers' (1 ANALYSIS, 5 WIDGETS). The 'Customers' layer is currently selected. The 'Widgets' panel on the right displays four data widgets: 'Predicted Store Location' (C1 Clusters Customers, 0% NULL ROWS, 72% OF TOTAL, ALL SELECTED), 'Customer Online Purchase ...' (C0 Source table\_29\_2, - NULL ROWS, \$144.59 (avg)), 'Customer Median Income' (C0 Source table\_29\_2, 163 NULL ROWS, 0 MIN, 54K AVG, 0 MAX, 12K SELECTED), and 'Customer Median Age' (C0 Source table\_29\_2, 105 NULL ROWS, 0 MIN, 35.6 AVG, 0 MAX, 13K SELECTED). Each widget includes a small bar chart or table of data. The 'Predicted Store Location' widget shows a table with 6 rows and 2 columns. The 'Customer Median Income' widget shows a bar chart with 4 bars. The 'Customer Median Age' widget shows a bar chart with 1 bar.

Cluster	Customers
1	3.3k Customers
2	3.0k Customers
4	2.8k Customers
5	1.9k Customers
0	1.7k Customers

Income Range	Count
6.1k	~12k
87k	~12k
170k	~12k
250k	~12k

Age Range	Count
35.6	~13k

CARTO

# MARK43 Computer Aided Dispatch

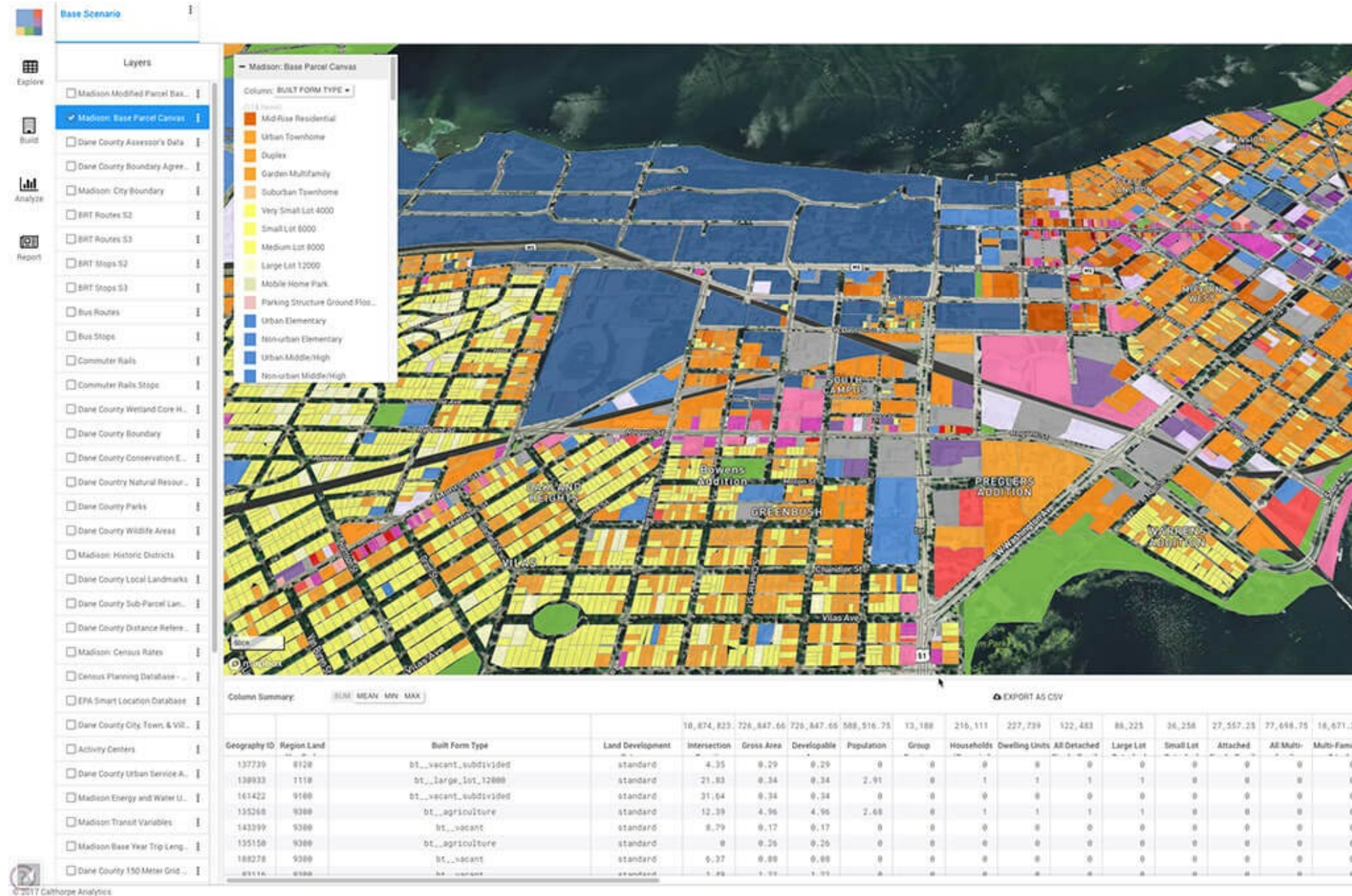
# 2012-Today



Mark43 Inc

# UrbanFootprint

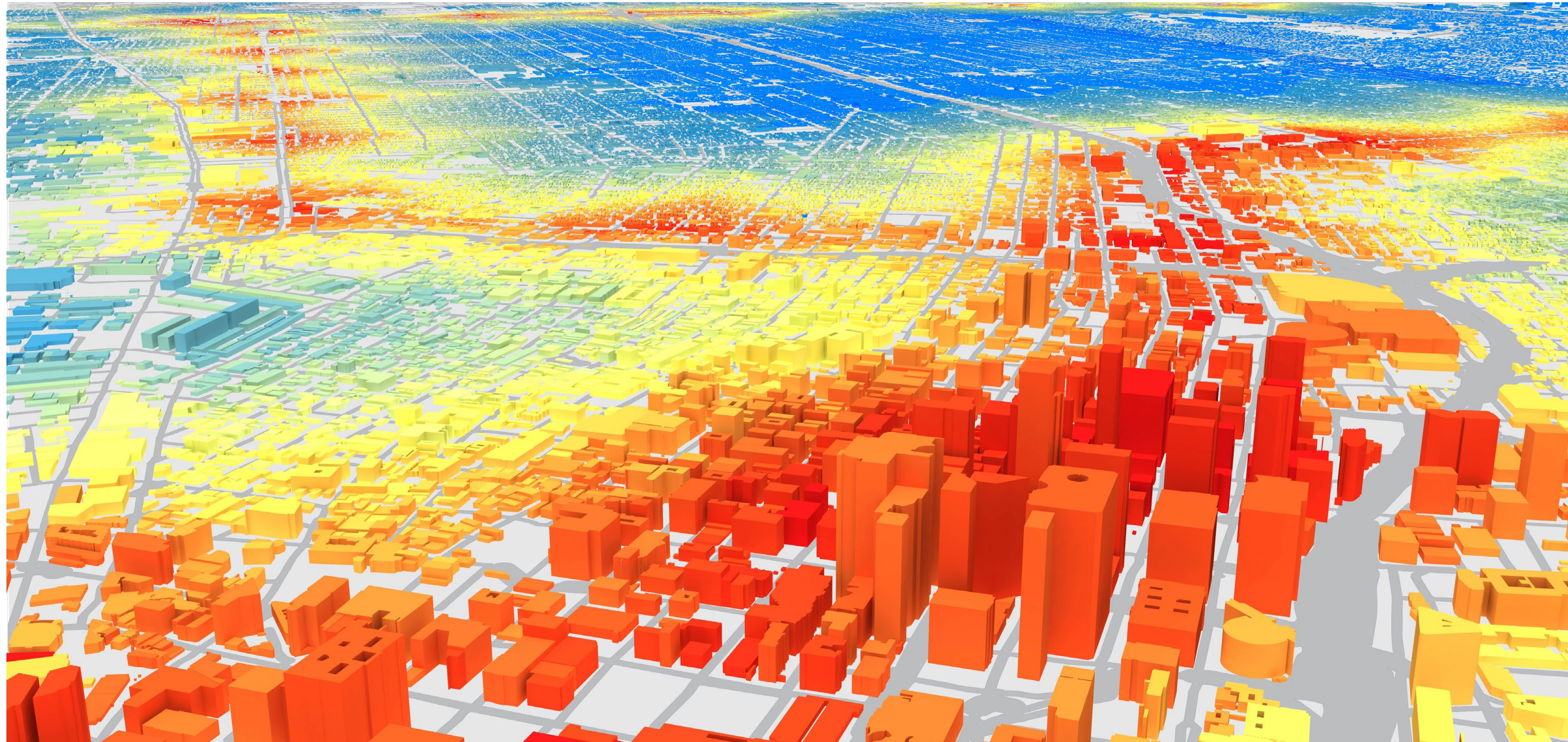
2014-Today



Joe Distefano, Peter Calthorpe for Calthorpe Analytics

# Urban Network Analysis Toolbox

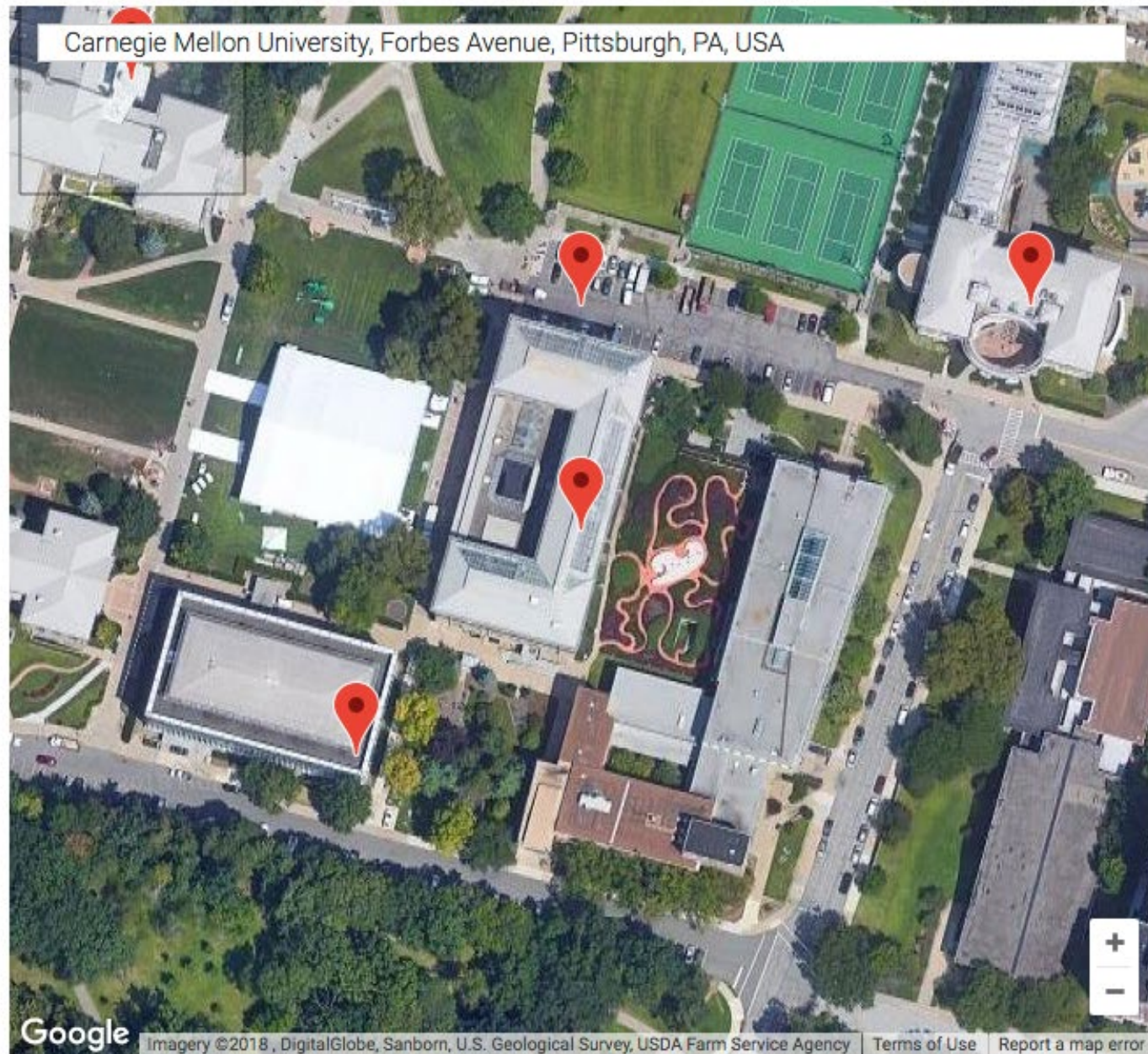
2015-Today



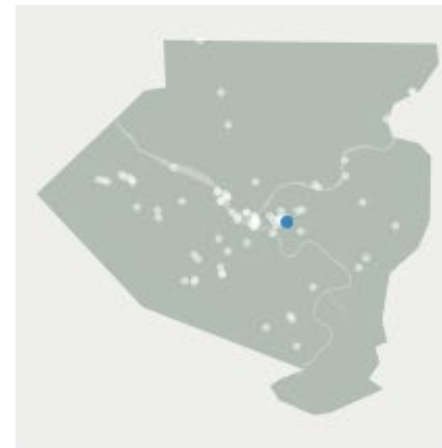
City Form Lab at the Harvard University Graduate School of Design

# TerraPattern

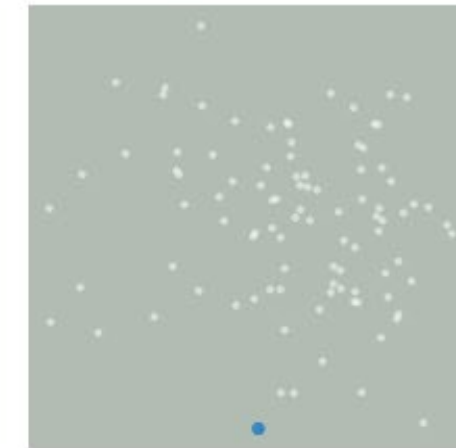
2016



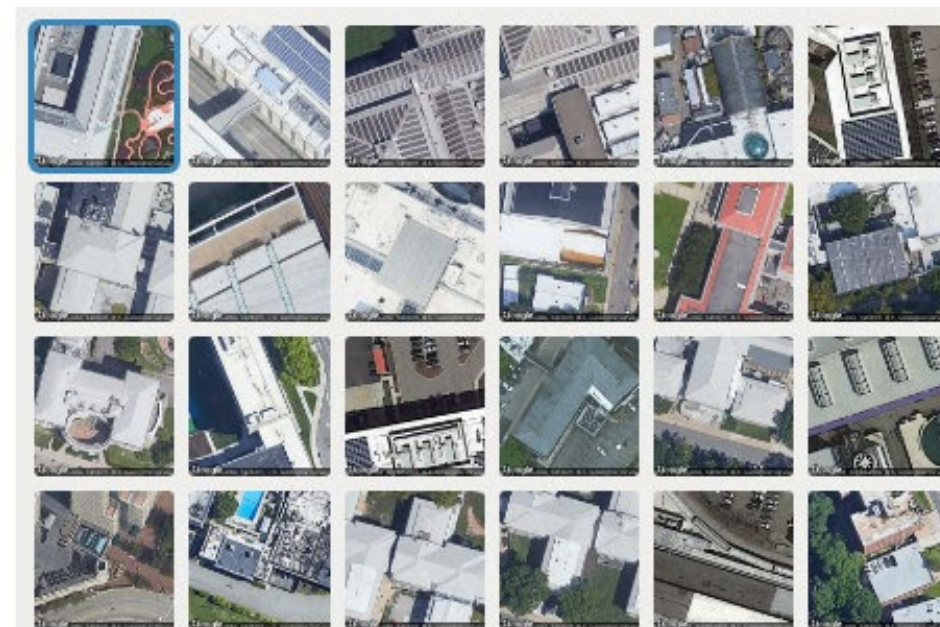
Geographical Plot



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

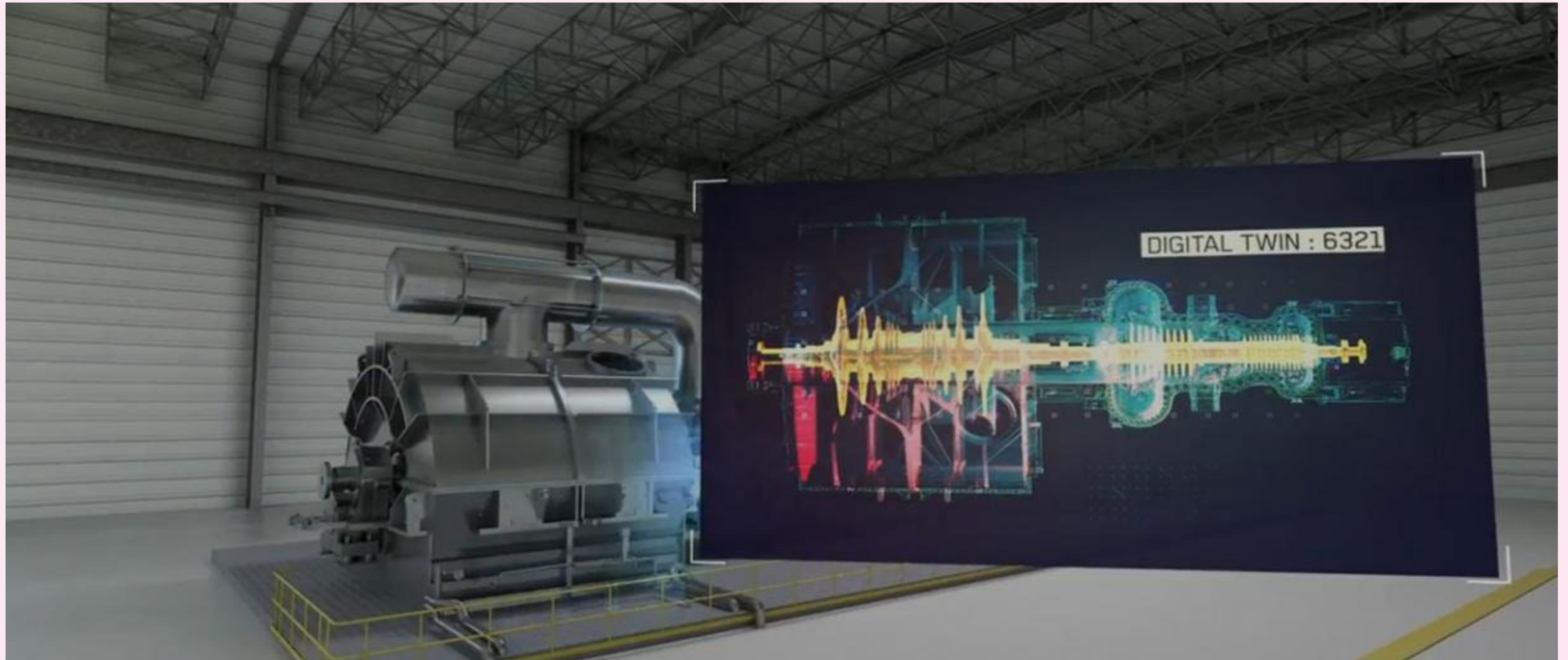
2016



Boeing

# GE Predix (Digital Twin)

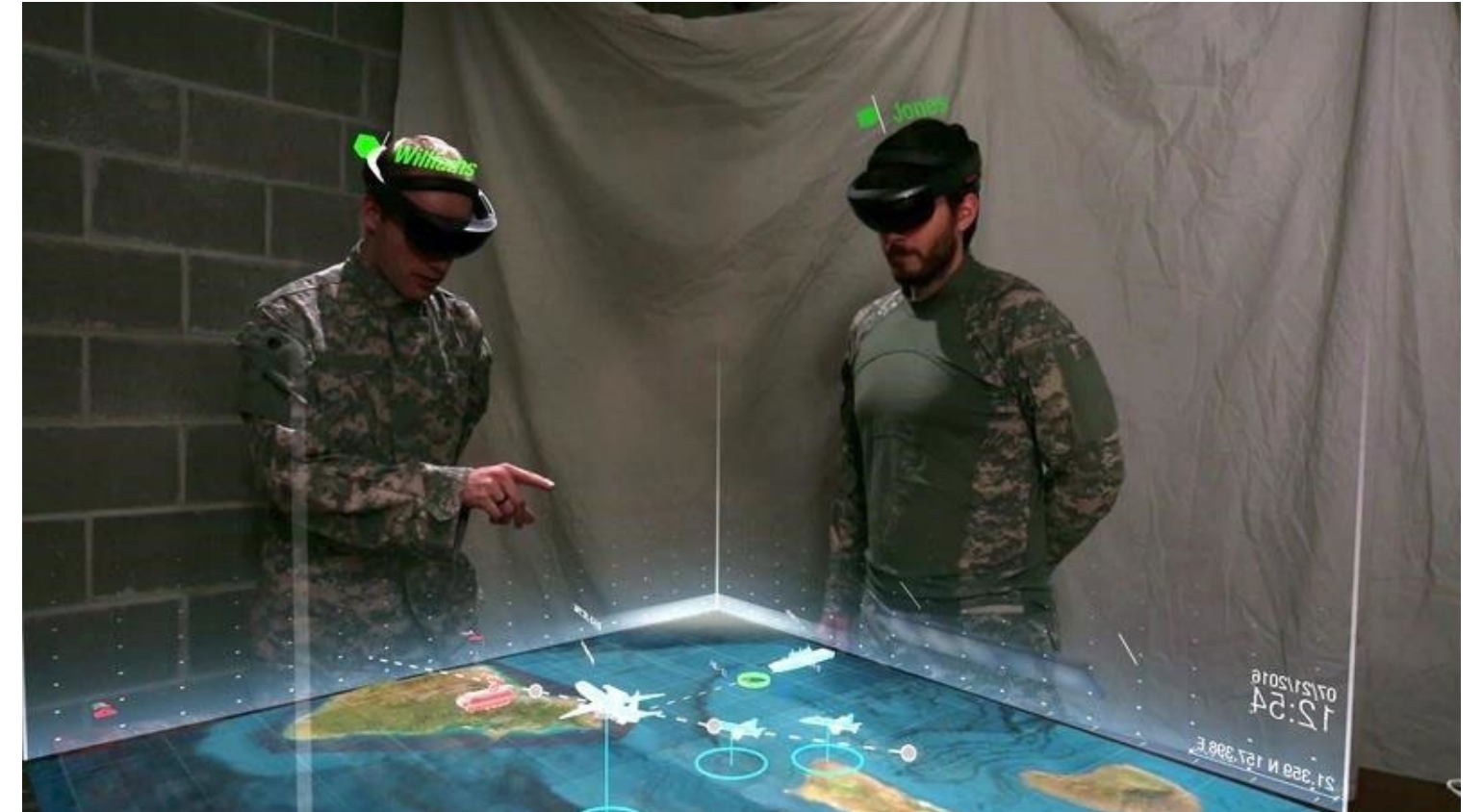
2016-Today



General Electric and ANSYS

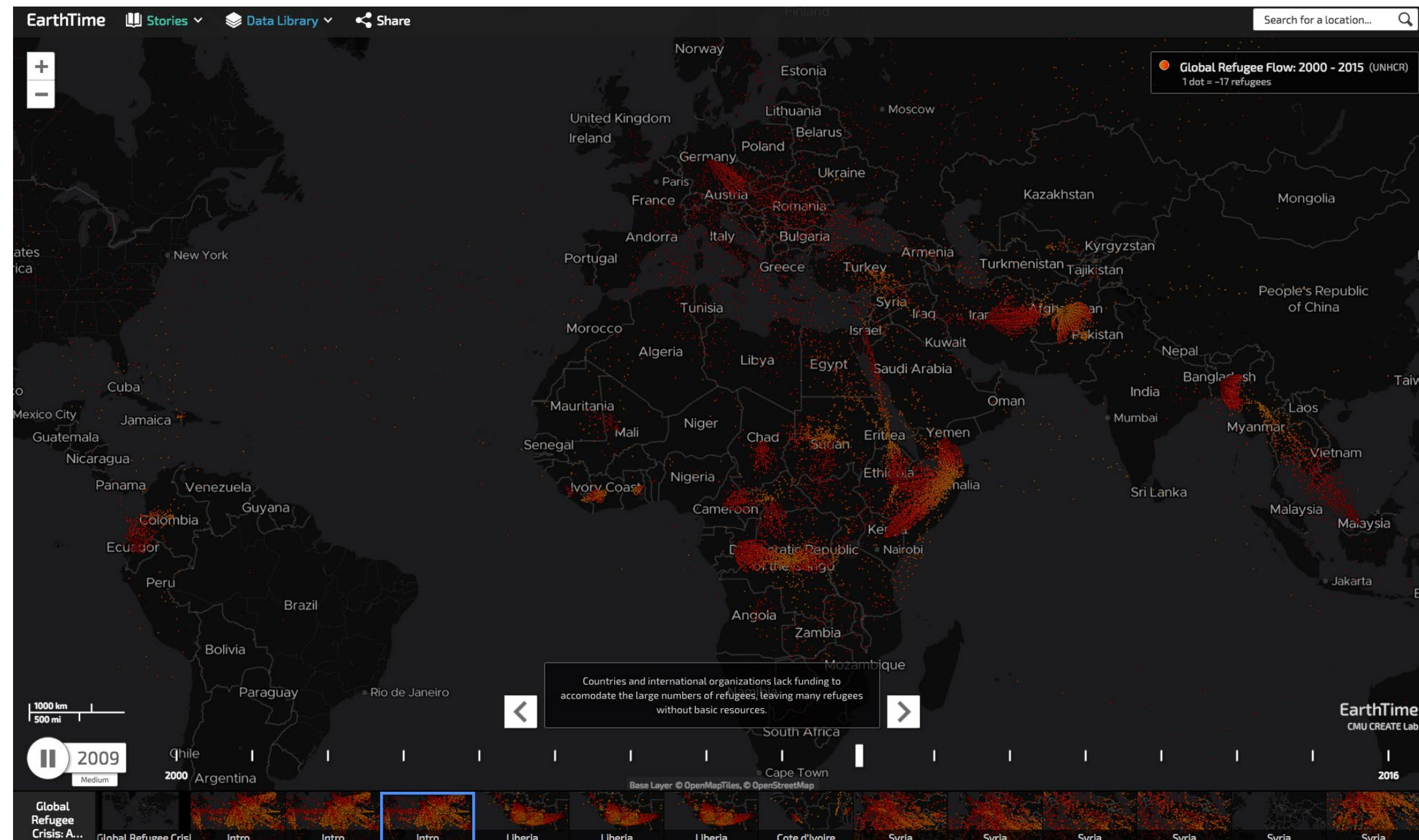
# Hololens App: Air Force Asset Planning

2017



Royal Australian Air Force

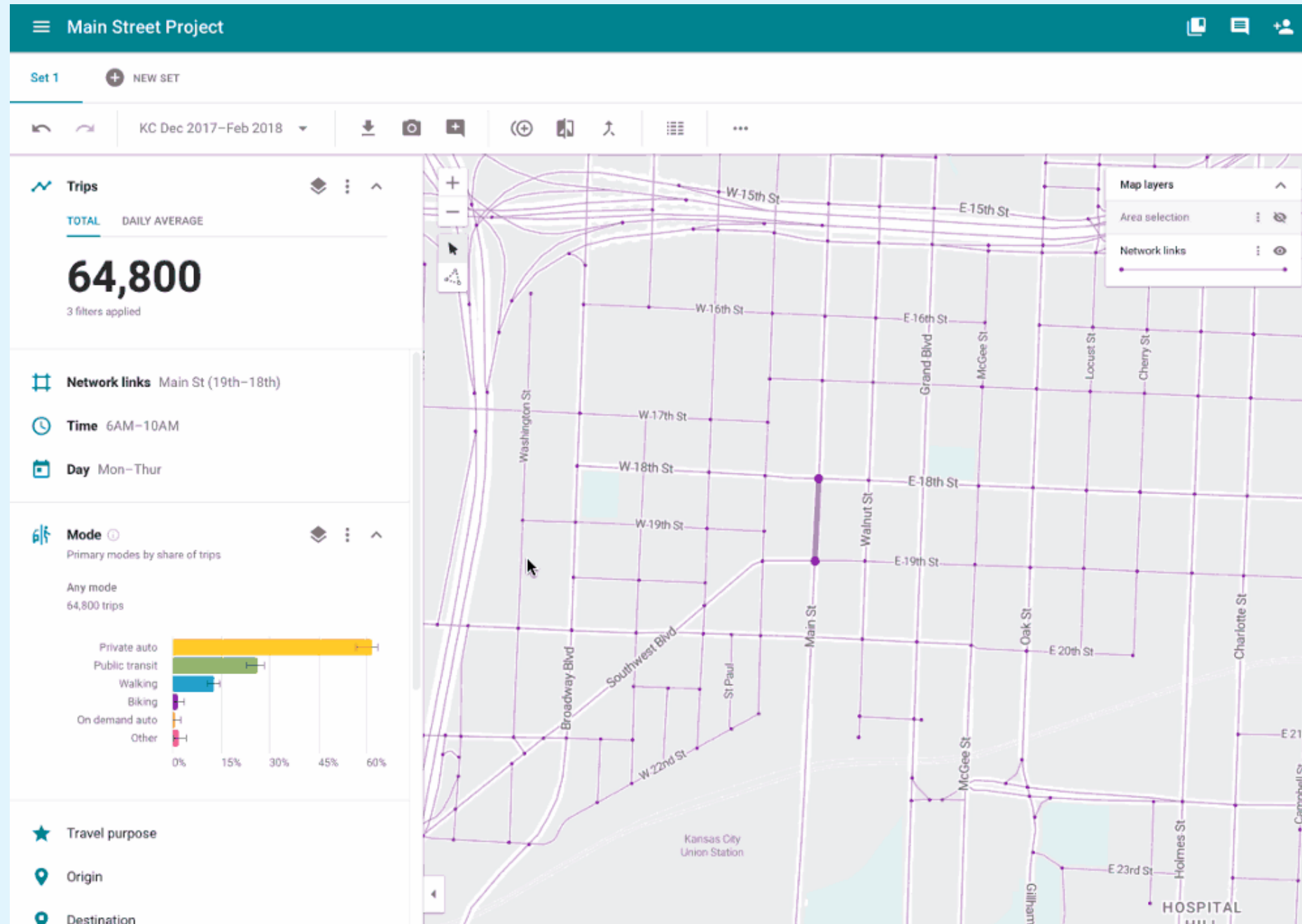




CREATE (Community Robotics, Education and Technology Empowerment) Lab at Carnegie Mellon University

# Replica

# 2018-Today



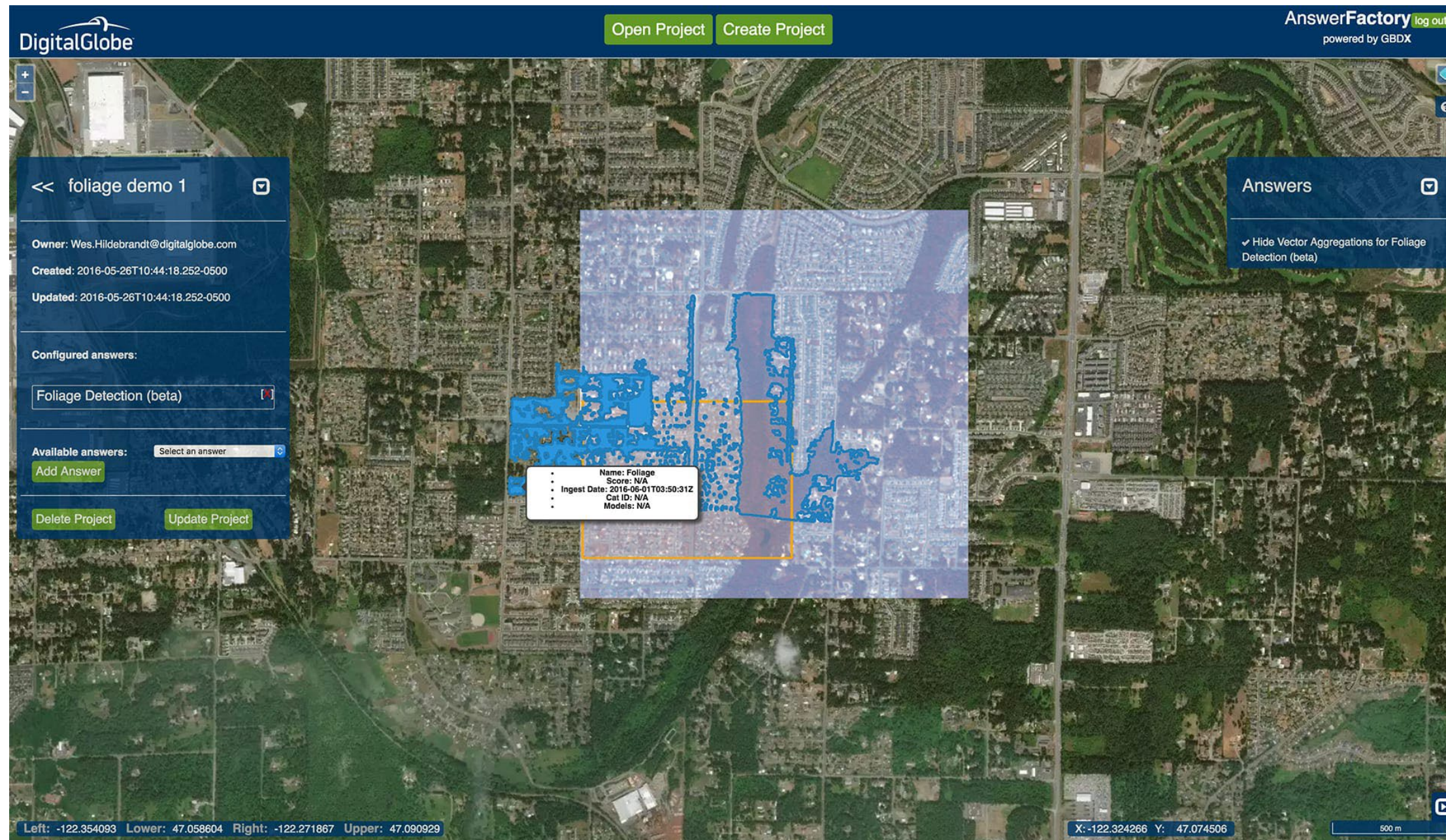
Sidewalk Labs (Part of Alphabet)

# Global View

'Macrosopes'

# DigitalGlobe AnswerFactory

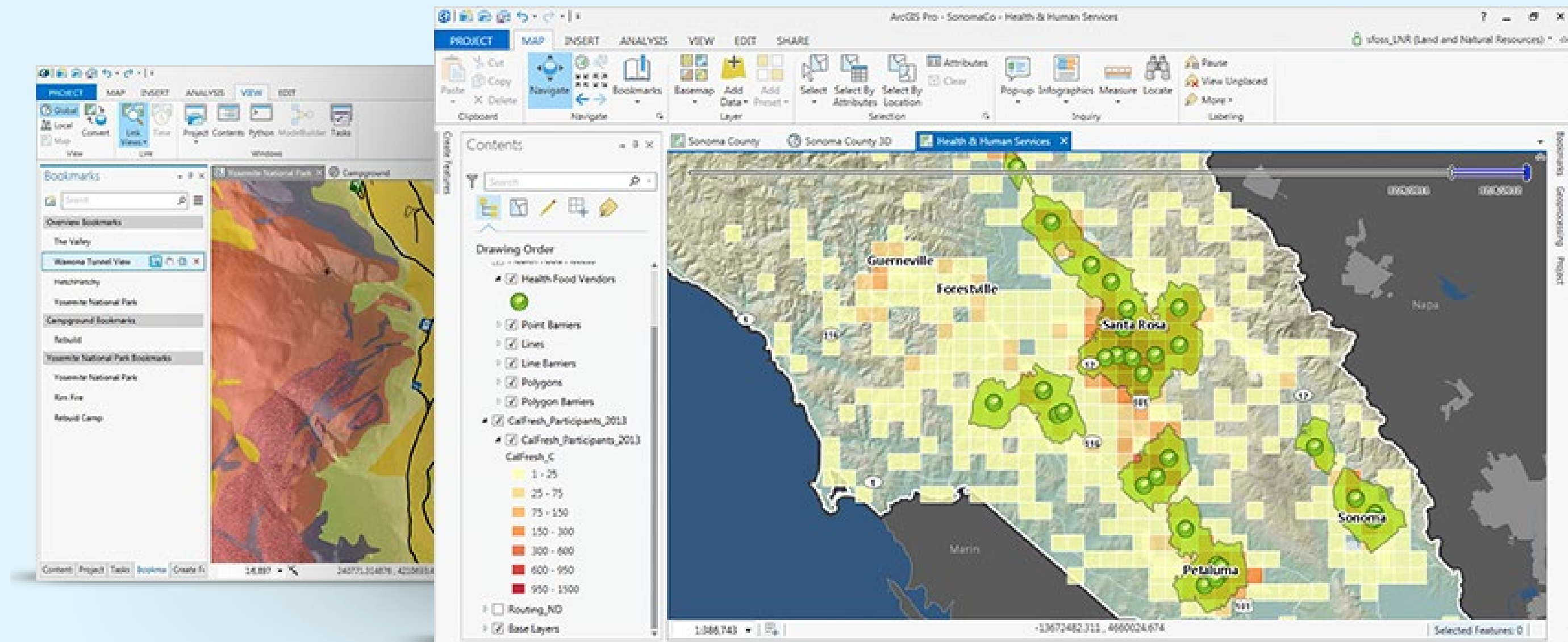
# 1992-Today



DigitalGlobe (Part of MAXAR)

# ArcGIS

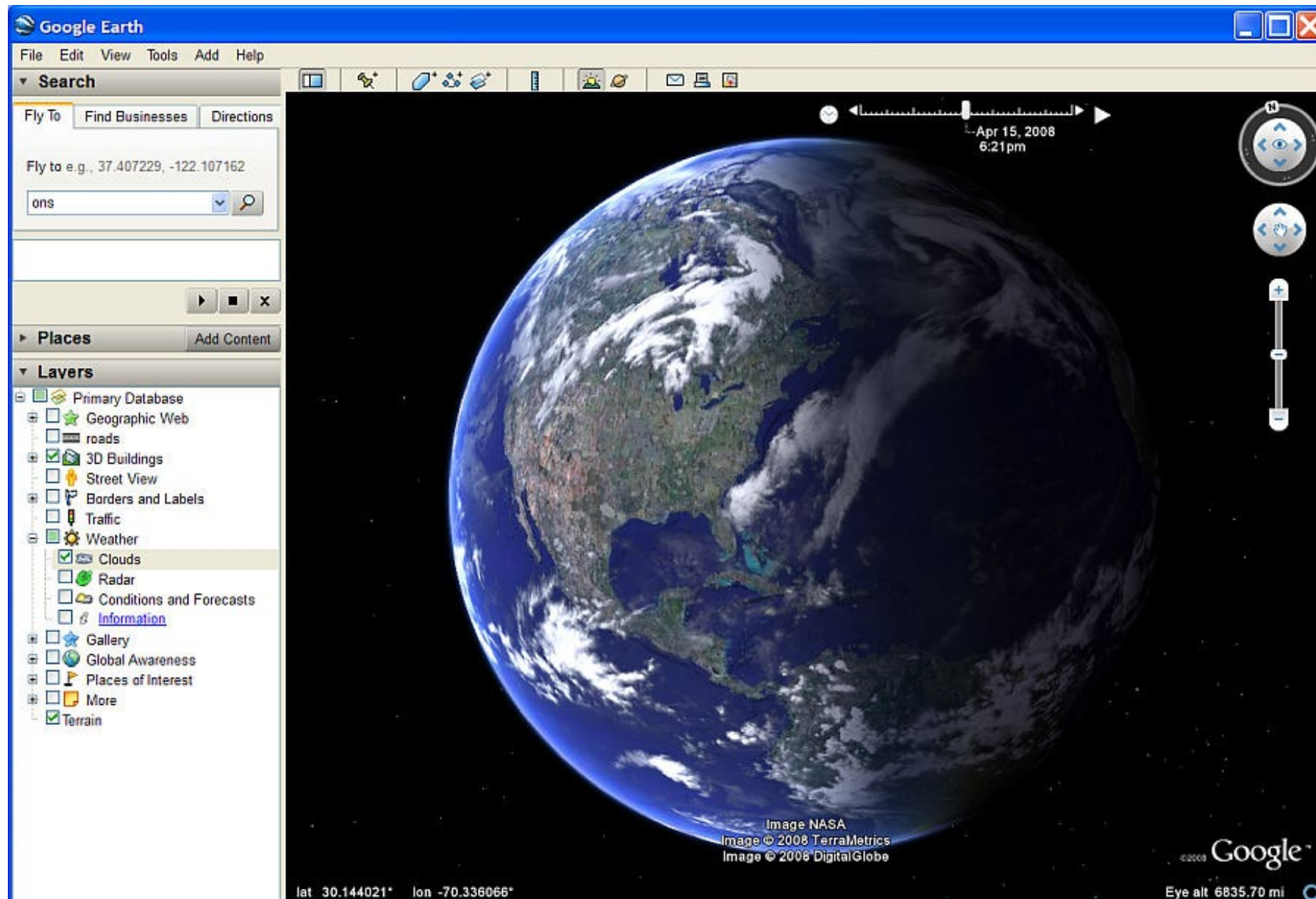
# 1999-Today



Esri

# Google Earth

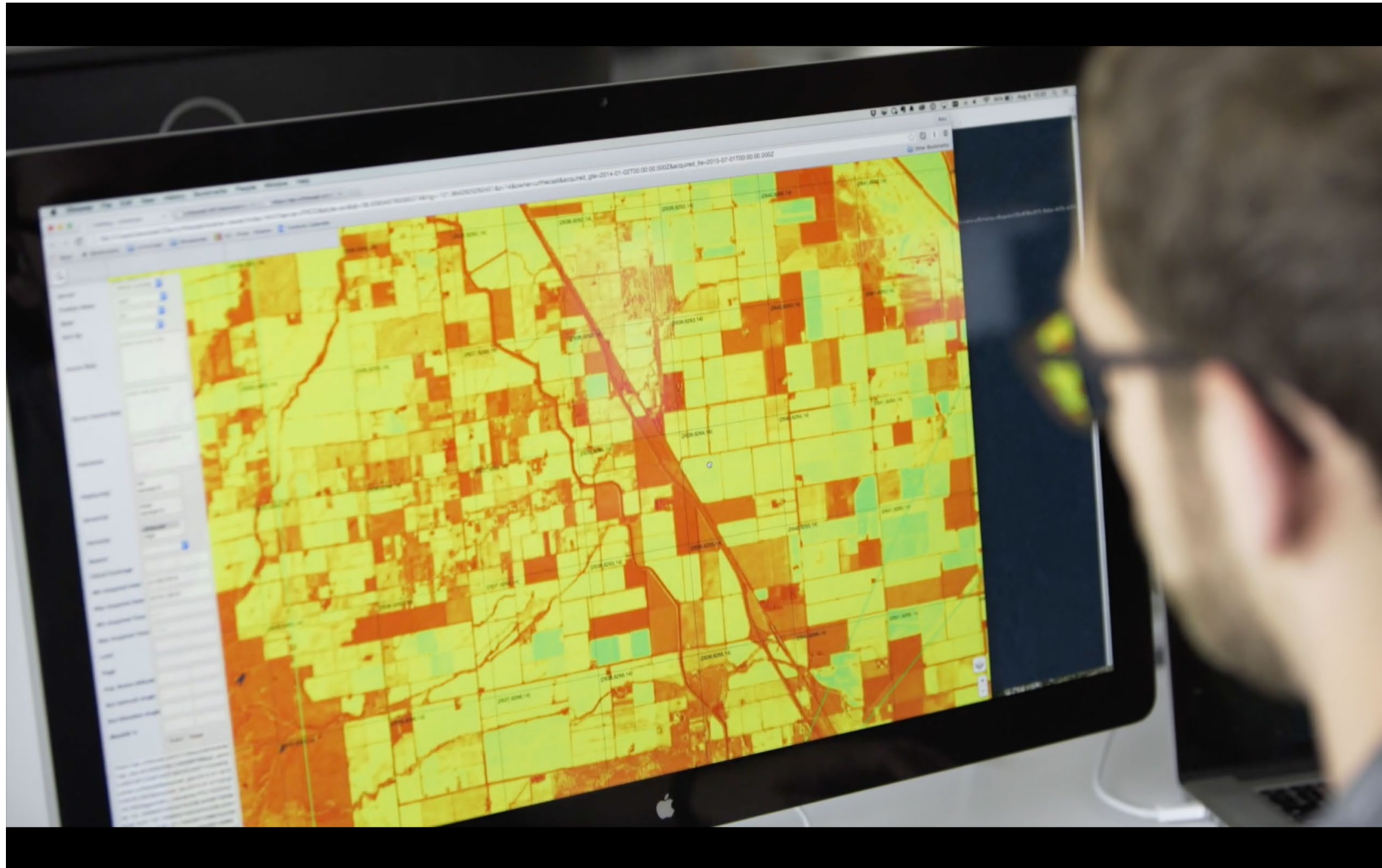
2001-Today



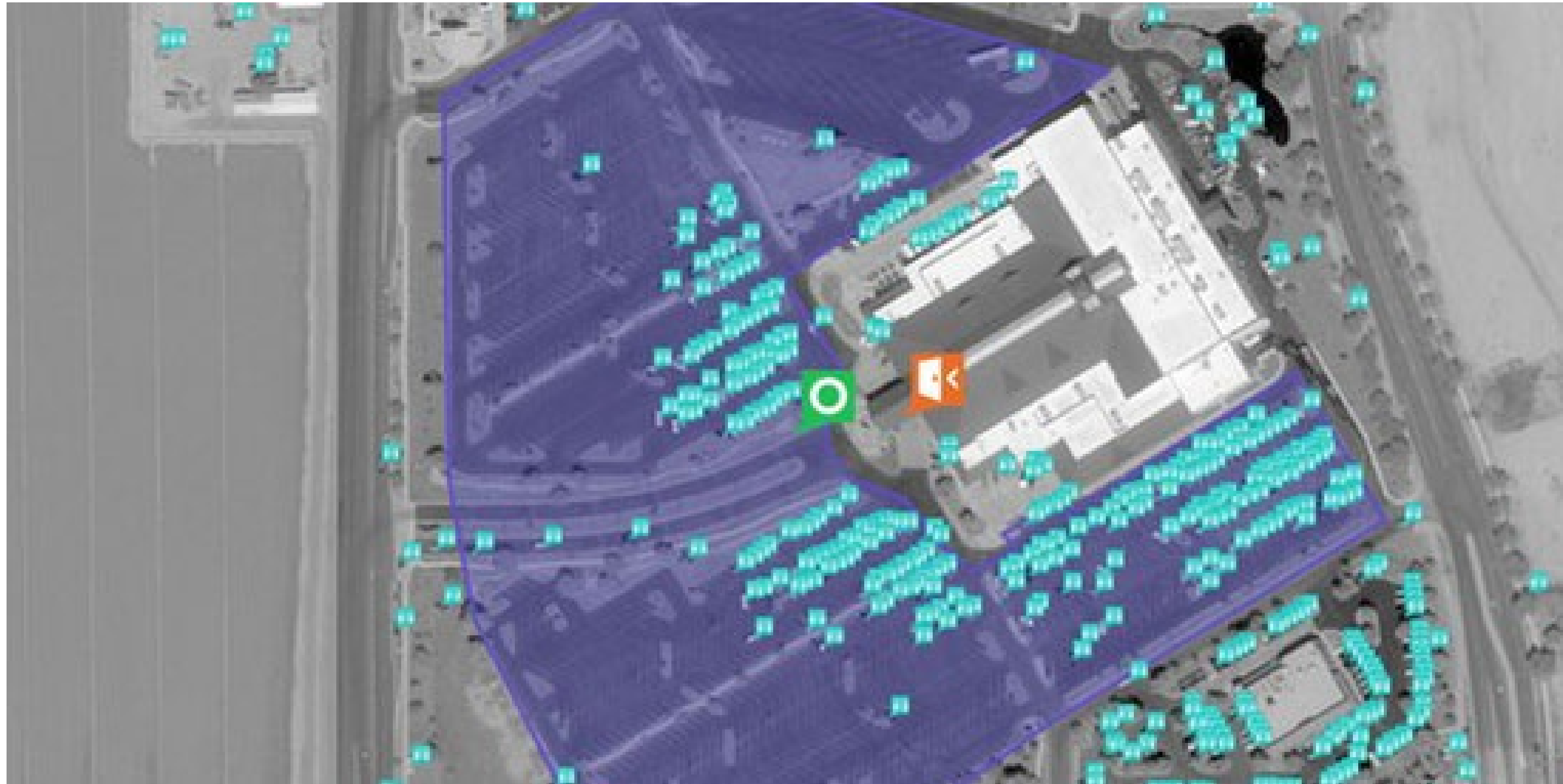
Intrinsic Graphics (Later Keyhole Inc, Google)

# UrtheCast

# 2004-Today



UrtheCast Corp



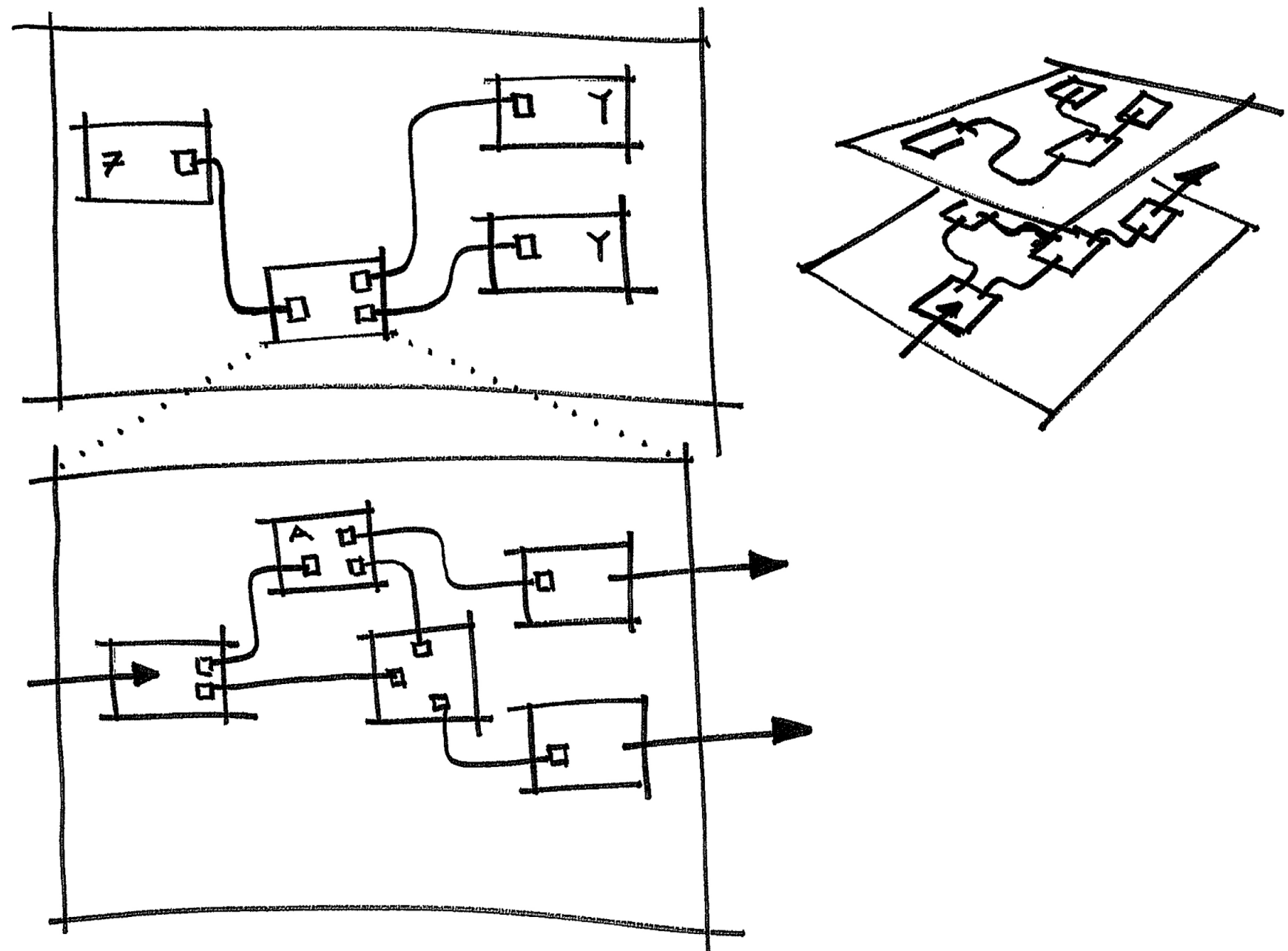
Orbital Insight Inc



# Design Patterns

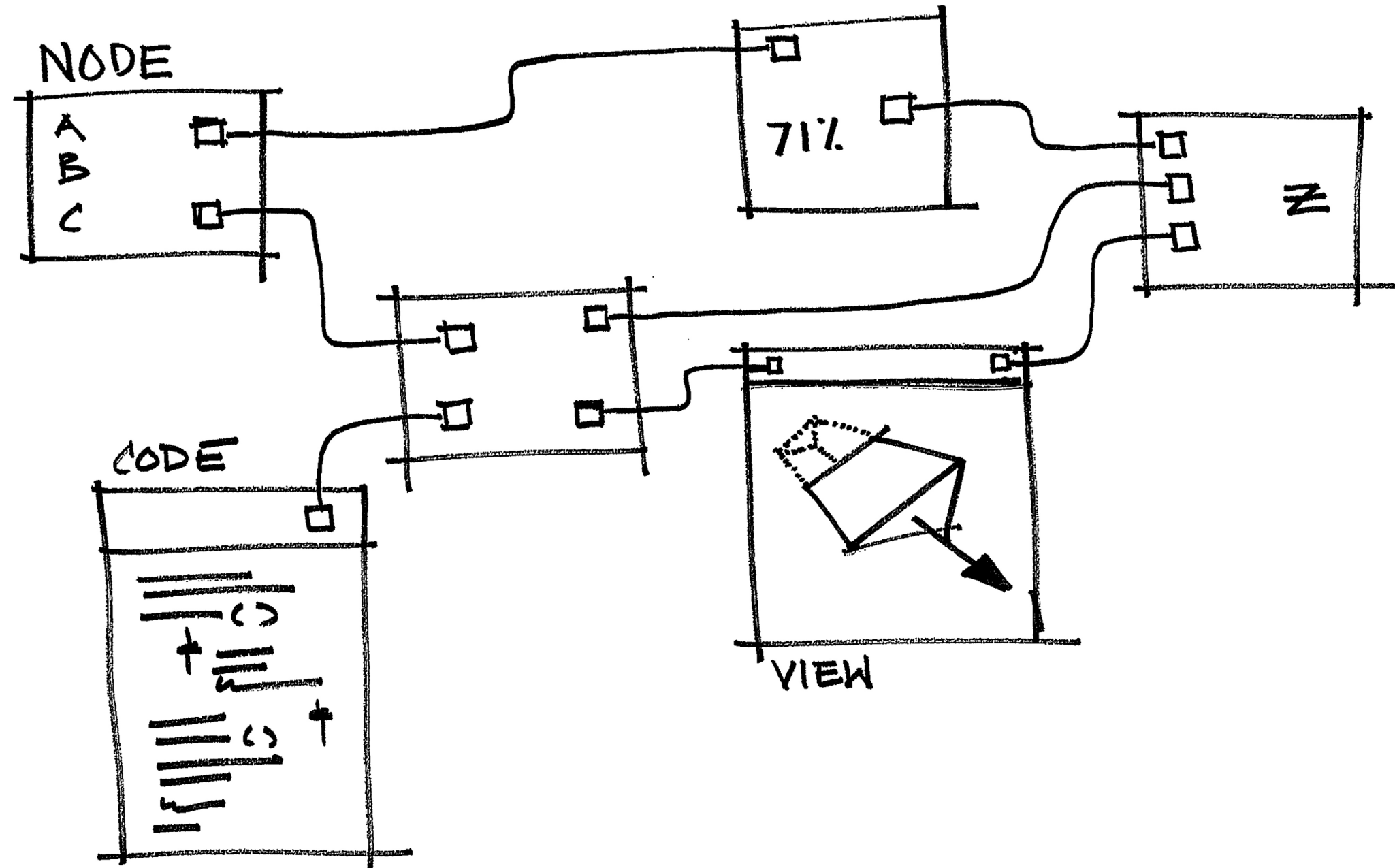
## Preliminary Sketches

# NESTED STRUCTURE



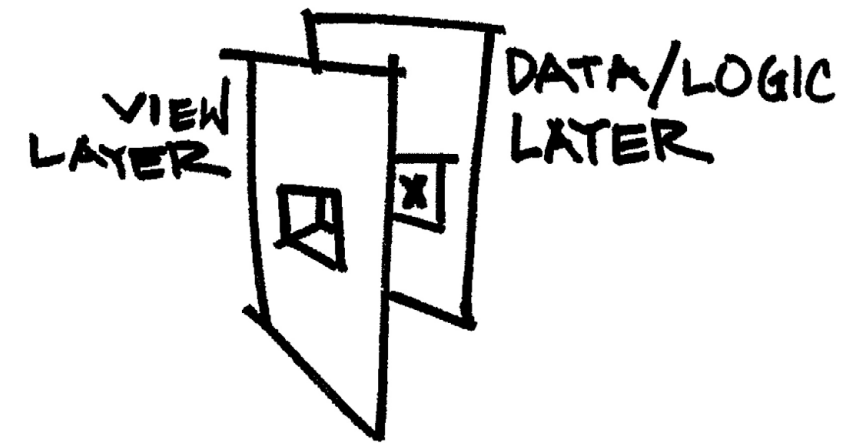
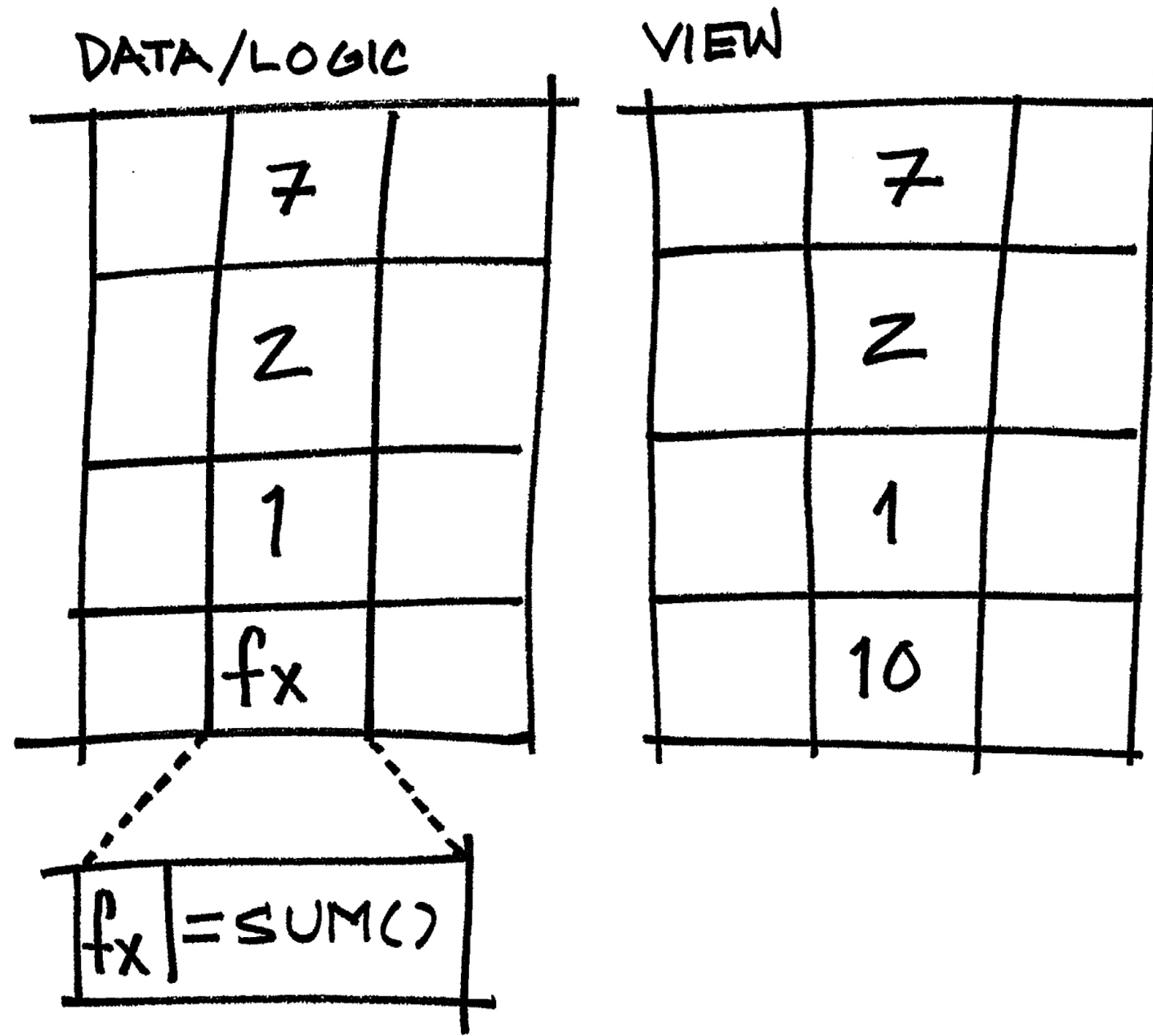
4.3.6, 4.1.6, 1.3.4

# GRAPH (NODE-LINK) ENVIRONMENT



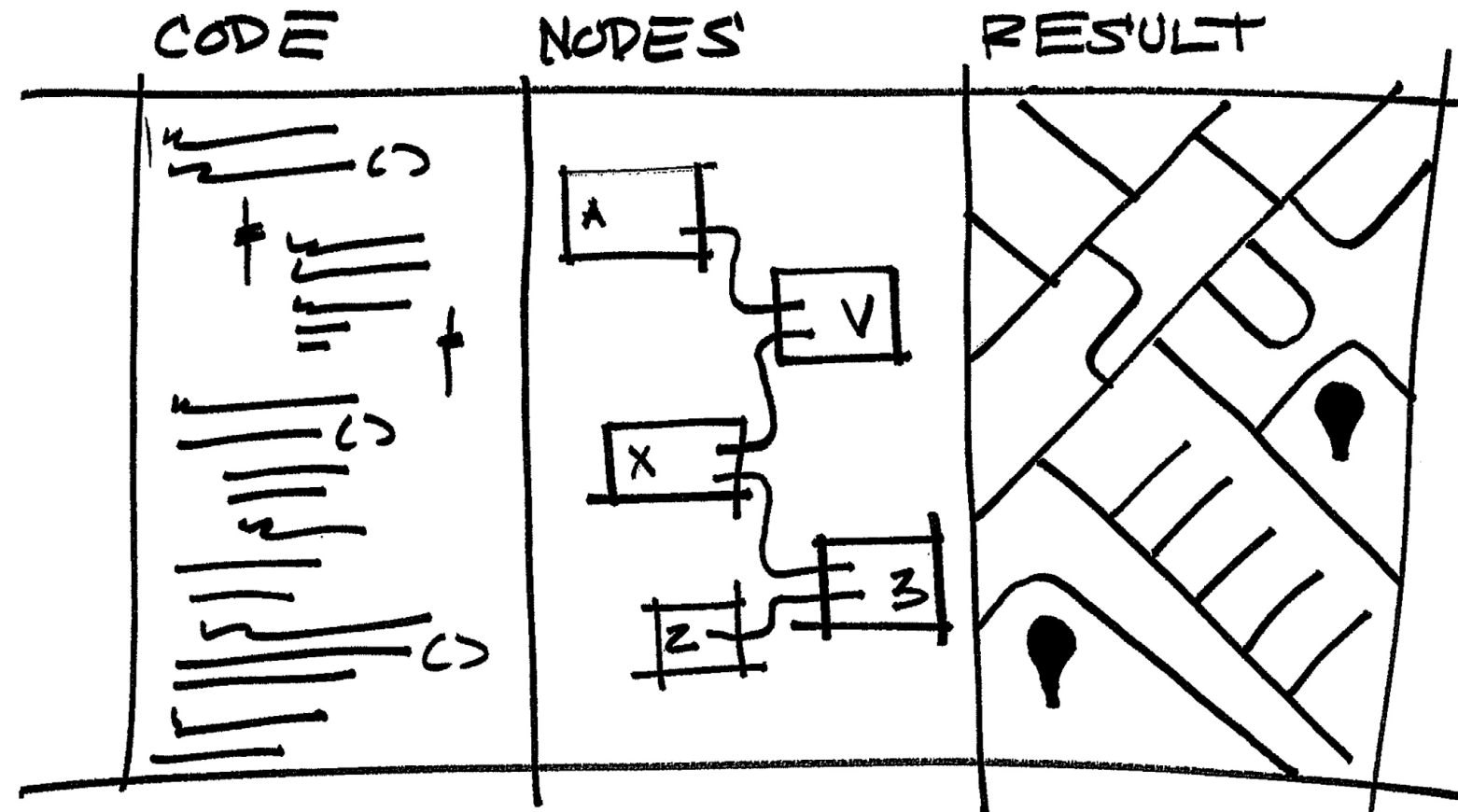
4.1.8, 4.3.1, 4.1.1

# KEYHOLE MODEL



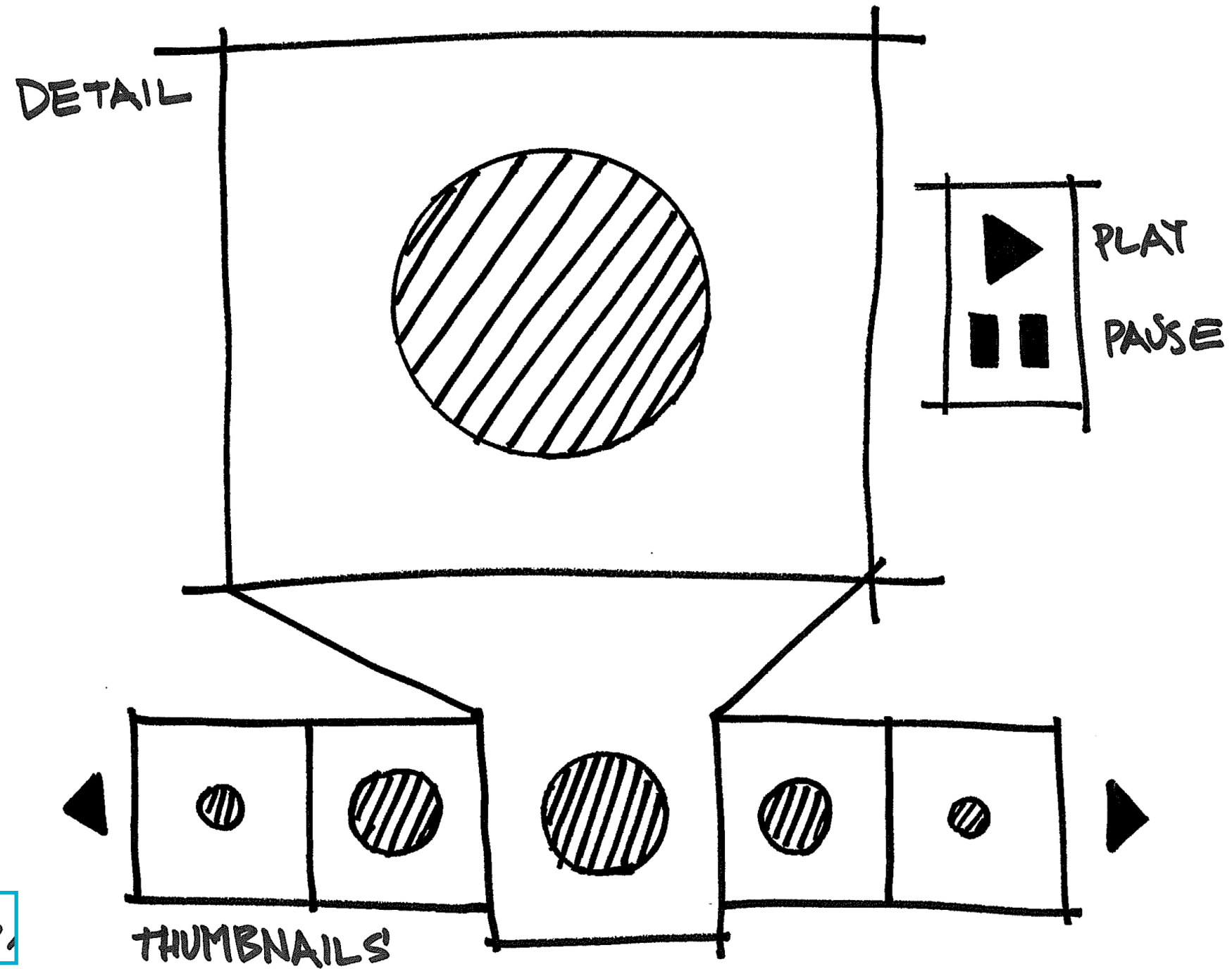
2.1.3, 2.2.3, 2.4.3

# MULTIPLE VIEWS



5.4.9, 5.4.3, 4.3.2, 7.2.1

# TIMELINE-BASED

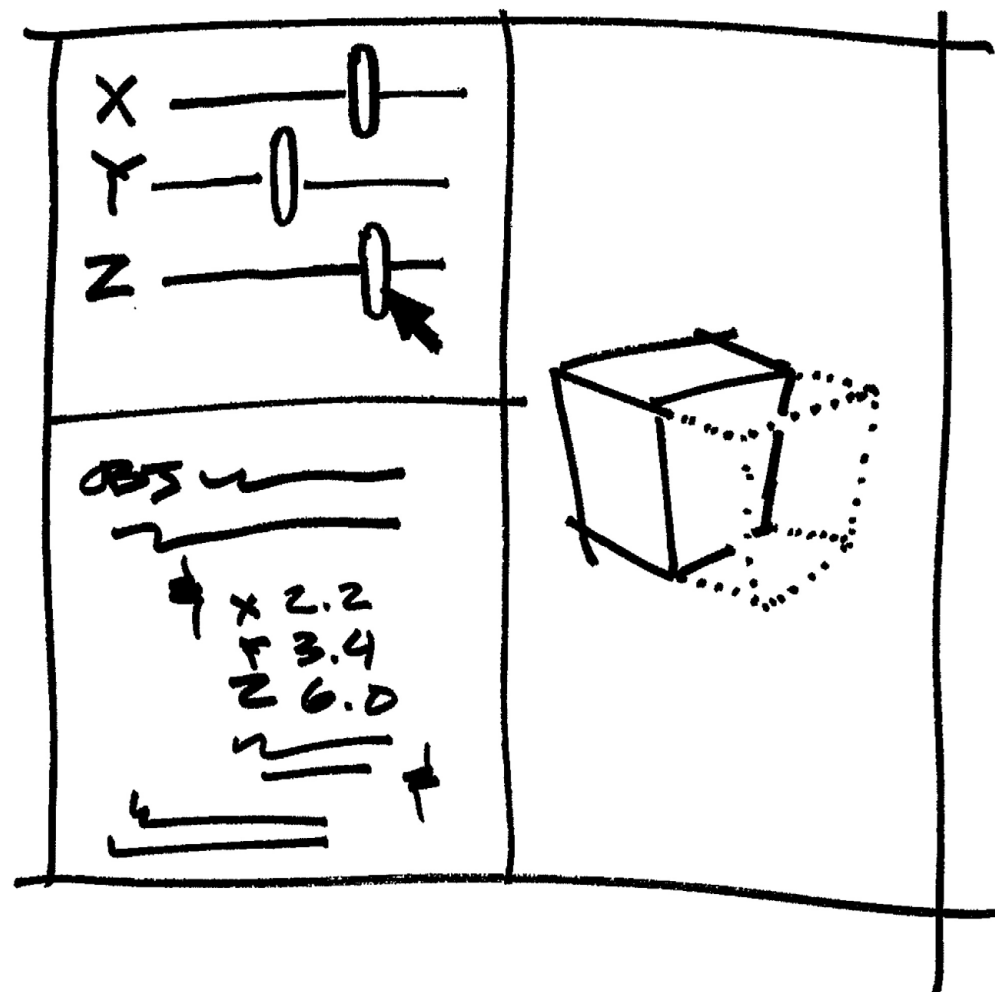


3.3.3, 6.3.6,

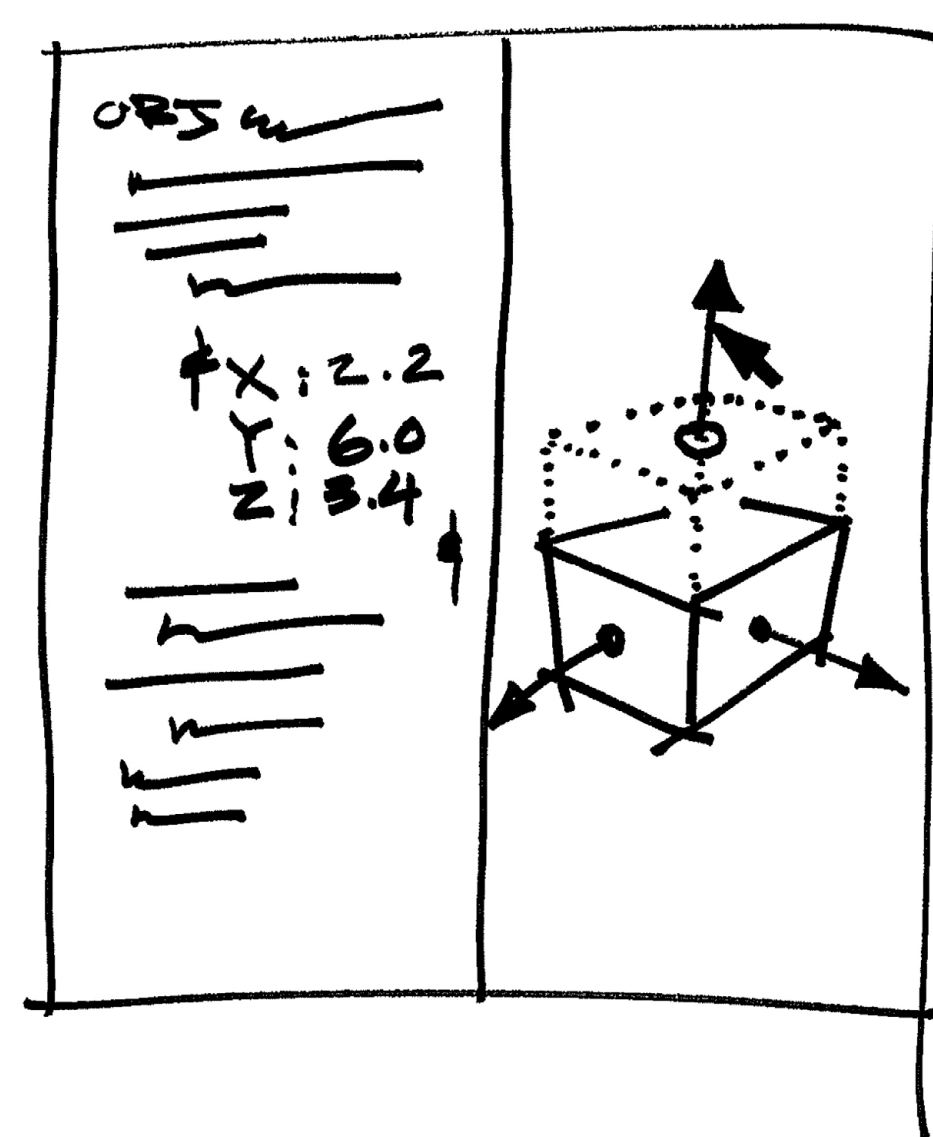
6.1.7

# LIVE COMPILER

## LIVE "FIDDLE"

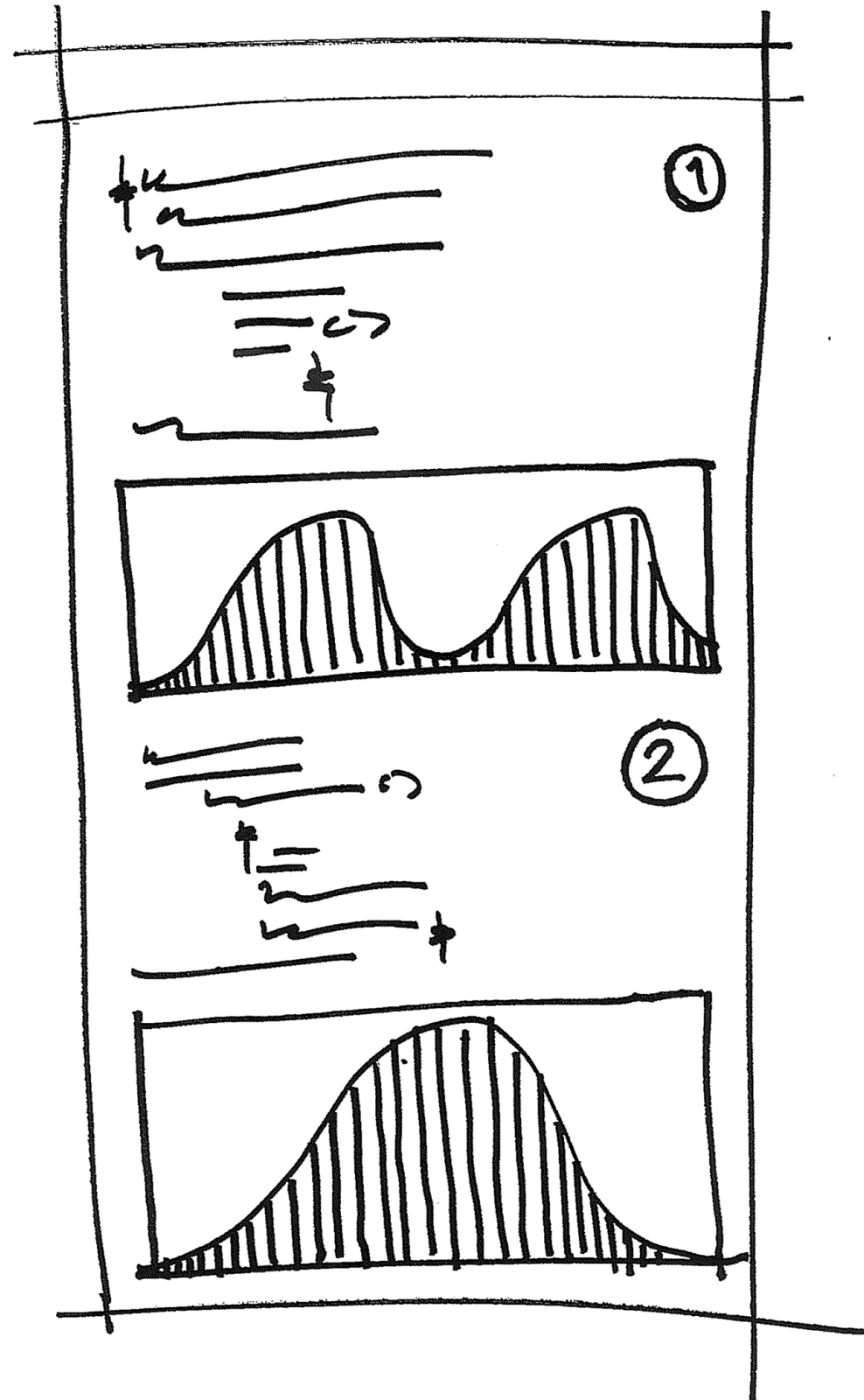


## EDITABLE RESULT



4.1.9, 5.4.7, 4.2.4

# QUERY & RESPONSE



1.2.4, 1.2.1, 1.2.6



hugh@dubberly.com  
@DubberlyDesign

cameron@dubberly.com  
@supercgeek