



Human-
Computer
Interaction
Institute

Carnegie Mellon University

RUBY: Reminiscing about **U**ser interfaces by **B**rad over the **Y**ears

Brad A. Myers

Human-Computer Interaction Institute

School of Computer Science

Carnegie Mellon University

<http://www.cs.cmu.edu/~bam>

bam@cs.cmu.edu

40th Anniversary!



Carnegie Mellon University

RUBY

Reminiscing about
User interfaces by
Brad over the
Years

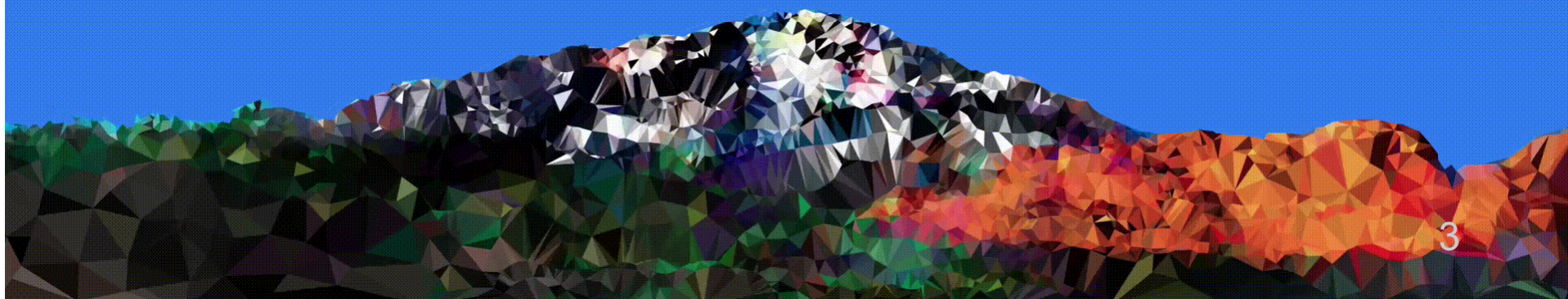




Carnegie Mellon University

CHI 2017

EXPLORE INNOVATE INSPIRE



3

35 CHI conferences: 1982 - 2017



Lots of Badges



Carnegie Mellon University





Carnegie Mellon University

35 CHI conferences: 1982 - 2017



Conference ribbons

1990

*(Thanks to Ben Shneiderman
for the picture)*



Human-Computer Interaction Institute
Carnegie Mellon University

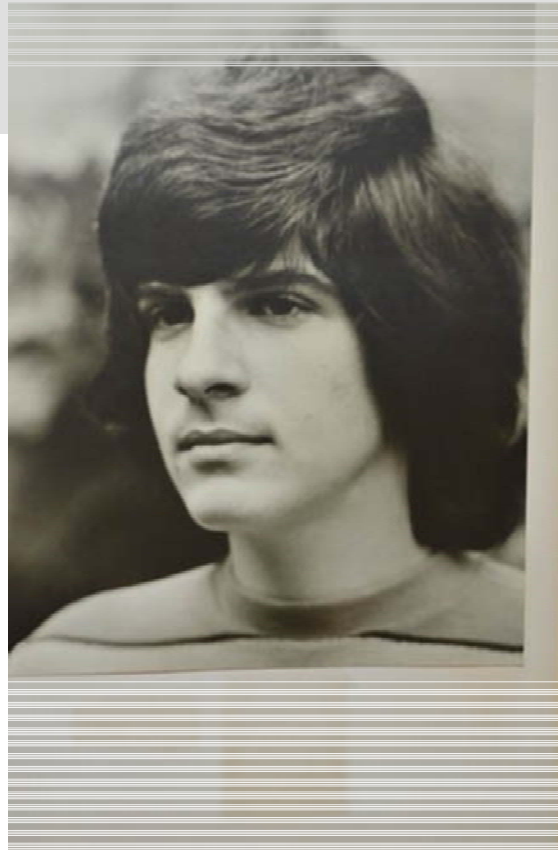
Baby me



Brad with Hair



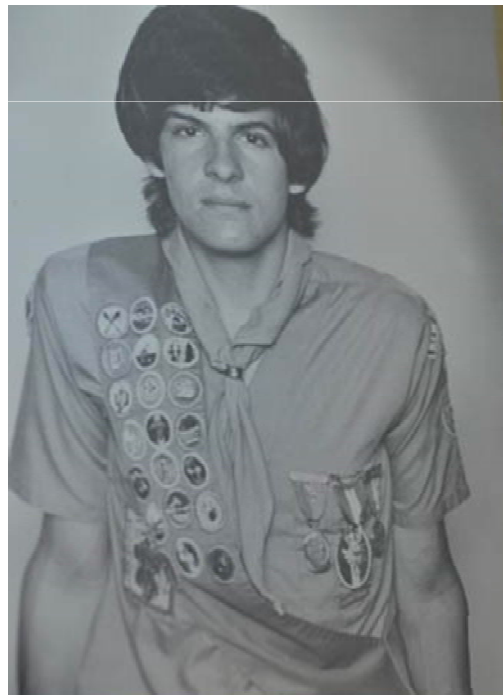
Carnegie Mellon University



Nerdy me



Carnegie Mellon University



~1974 - Teletype

```
5 LET S = 0
10 MAT INPUT V
20 LET N = NUM
30 IF N = 0 THEN 99
40 FOR I = 1 TO N
45 LET S = S + V(I)
50 NEXT I
60 PRINT S/N
70 GO TO 5
99 END
```

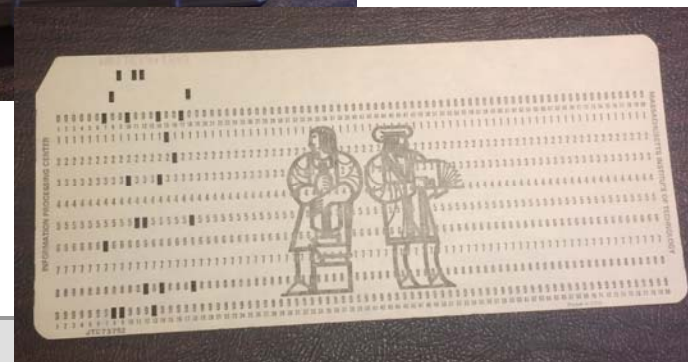
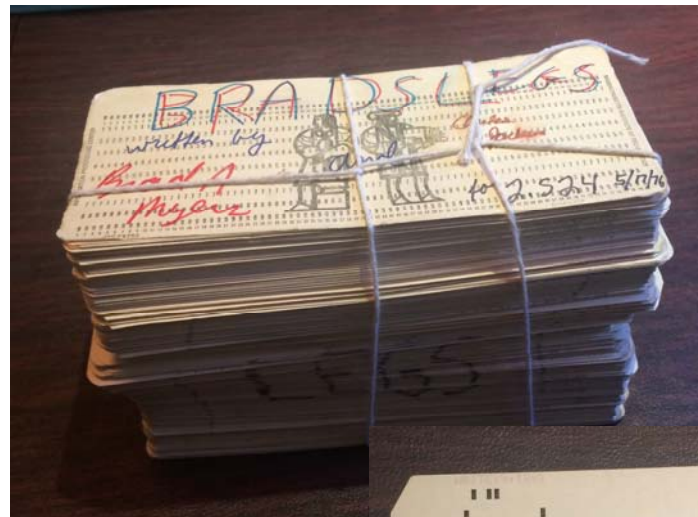
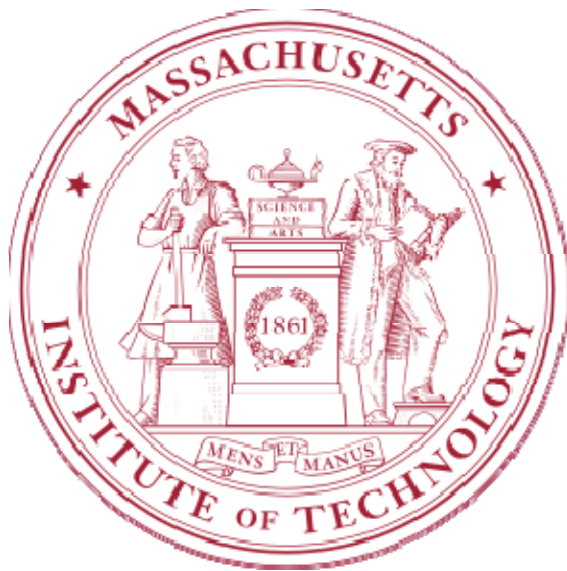


By Jamie - Flickr: Telex machine TTY, CC BY 2.0, <https://commons.wikimedia.org/w/index.php?curid=19282428>

MIT



Carnegie Mellon University



12



Human-
Computer
Interaction
Institute

Carnegie Mellon University

Observations

- Brad is really old...
- If you don't keep stuff, you won't have it 40 years later!



Spatial Data Management System (SDMS)

- ~ 1977
- MIT Architecture Machine Group
 - Predecessor of MIT Media Lab
- Undergraduate research



Christopher F. Herot, Richard Carling, Mark Friedell, and David Kramlich. 1980. A prototype Spatial Data Management System. In *Proceedings of the 7th annual conference on Computer graphics and interactive techniques (SIGGRAPH '80)*. ACM, New York, NY, USA, 63-70. DOI=<http://dx.doi.org/10.1145/800250.807470>

Xerox PARC

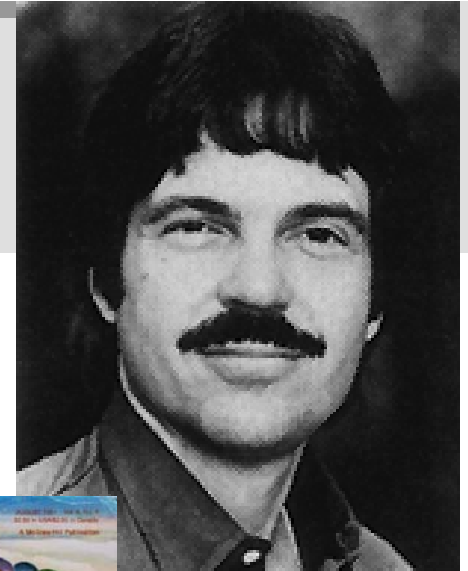
- Important research lab
- Summer intern



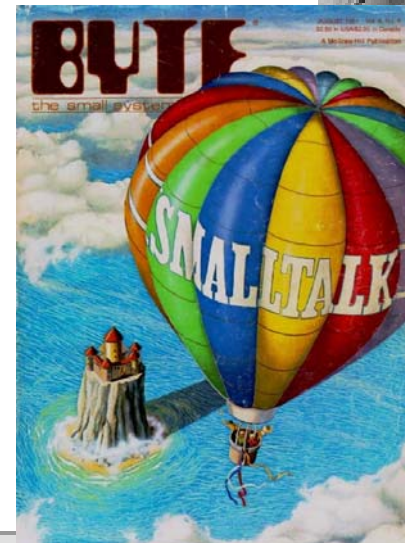
Picture credit: <http://www.digibarn.com/collections/locations/xerox-parc/>

Smalltalk group at PARC

- Summer 1977
- Systems Sciences Lab (SSL)
- Alan Kay
- Also: Dan Ingalls, Adele Goldberg, Ted Kaehler, Bruce Horn, etc.

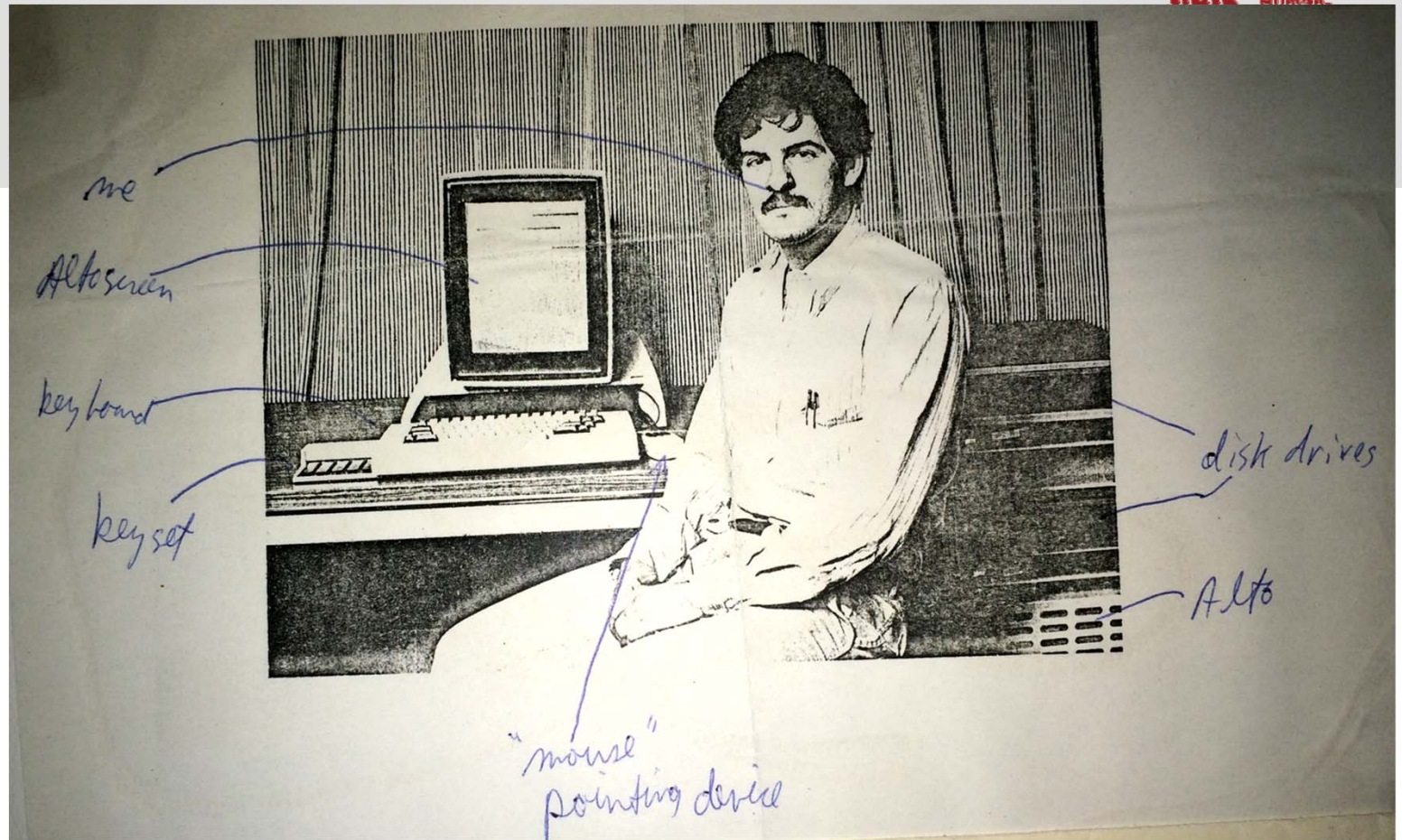


Alan Kay



Xerox Alto

• 1979



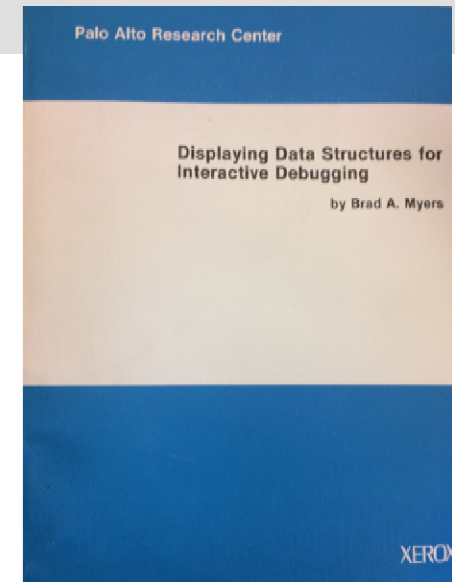
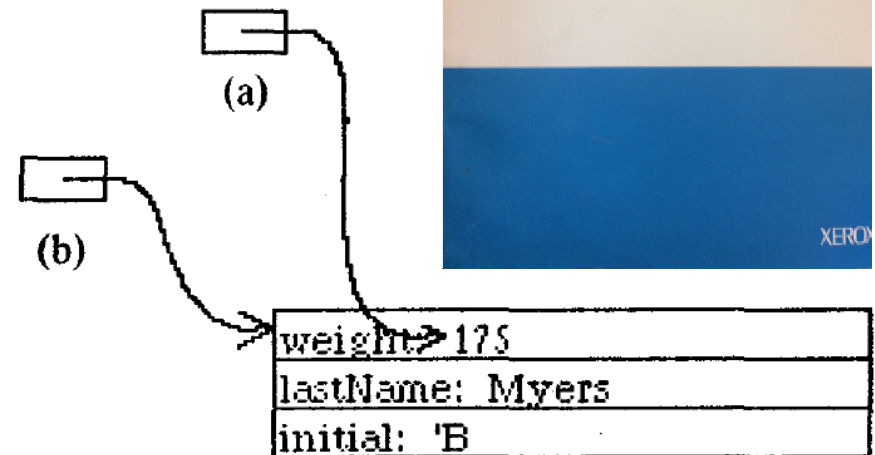
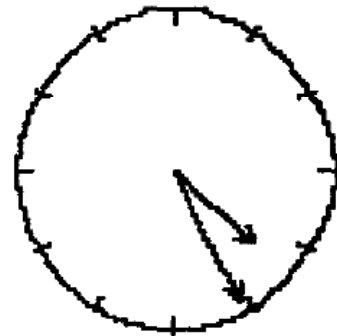
Computer Sciences Lab (CSL) at PARC

- Summer 1978
- Summer & Fall 1979 – Master’s Thesis
- Used the “Mesa” programming language
- Advised by: Dan Swinehart (PARC) and David Reed (MIT)
- Helped by: John Warnock, Ed Satterthwaite, Butler Lampson, Warren Teitelman, Bill Paxton and Paul Rovner

Incence

- Displays data structures similar to how they might be drawn on a blackboard
- Customizable

hours: 16
min: 25
seconds: 30



SIGGRAPH'83



Computer Graphics

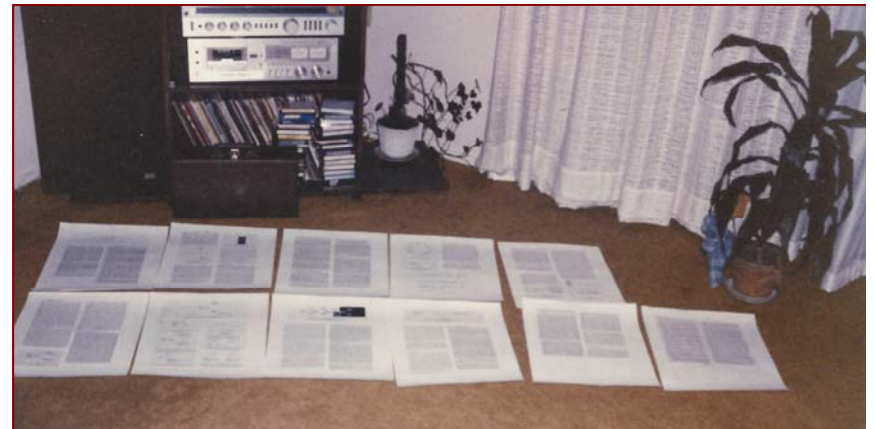
Volume 17, Number 3

July 1983

INCENSE: A SYSTEM FOR DISPLAYING DATA STRUCTURES

Brad A. Myers*

Xerox Palo Alto Research Center, California



20



Carnegie Mellon University

Observations

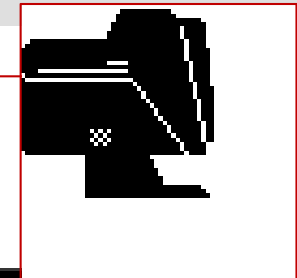
- It's better not to write out and read a talk.



Three Rivers Computer Corporation

- 1980-1983
- Pittsburgh, PA
- PERQ workstation





PERQ

A Complete Single-User Computer System

All the Processor, Display, Disk, and Keyboard are in one person needs, at his desk.

Ethernet local computer network connects up to 1024 PERQs in a distributed processing environment.

CursDesign, Version V6.1

PERQ Memory Description

PERQ MEMORY FEATURES

512 Kilobyte RAM

200 Megabit Aggregate Bandwidth

PERQ Processor Description

PERQ PROCESSOR

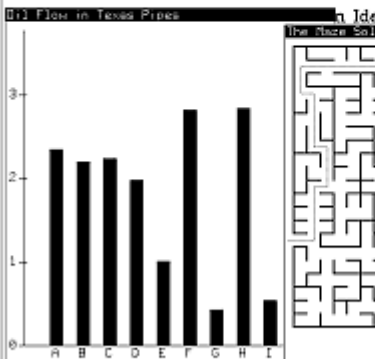
Micro-Programmed 18 Bit

Horizontal Micro-Architecture

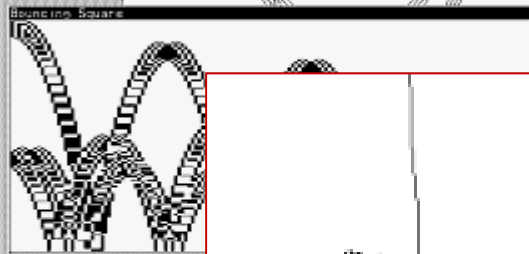
170 Nanosecond Micro-Instruction Cycle

Executes 1 Million P-Code Instructions per Second


Oil Flow in Texas Pipes

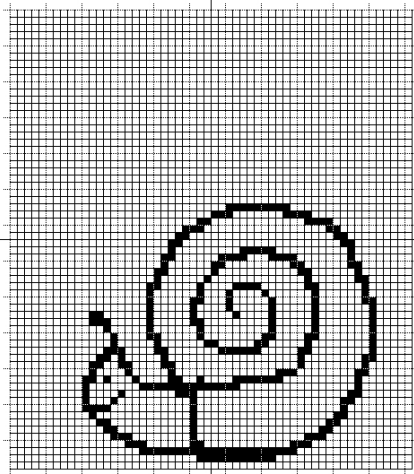


Bouncing Square



Maze





Cursor Origin
X = 0 Y = 0

System User > done >

Command	Type	? for Help
capl.cursor	opus.cursor	mouse.cursor
jungle.animate	ponder.cursor	write.cursor
bullseye.cursor	shoe.cursor	punct.cursor
join.cursor	tree.cursor	collection.S
nasty.cursor	cross_hairs.cursor	nuke.cursor
rotate.cursor	make_bee.animate	bee_make.animate
wpmo.cursor	menu.cursor	dirtree.cursor
grey.cursor	busy_bee.cursor	pm.cursor
		fourarrow.animate
		bad_bee.cursor

60 files found.

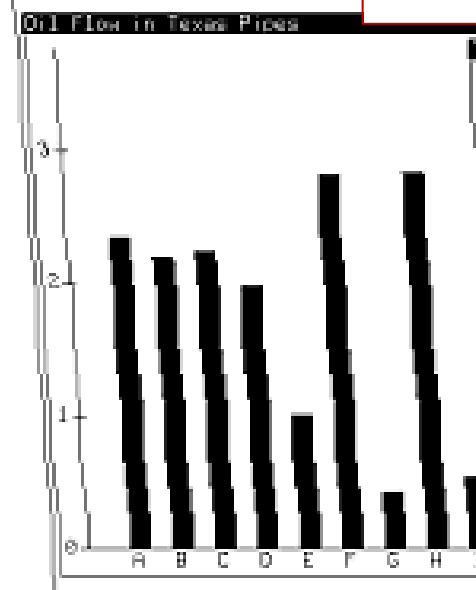
Get from solar or animate file? [a] s Filename: collection.SCcursor
collection.SCcursor has 46 cursors in it. How many do you want to read? [46]
Starting at [1]
Point at box where start reading.
** Could only fit 40 cursors.

System User > done >

tetra Z 0 4 200

just 8 0 new-slide 0 PERQ Memory

bouncingsquare 9 0 20 1500



Bernita

- 1982



Carnegie Mellon University

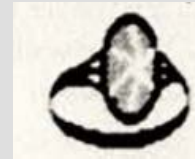
Bernita

- 1982
- 35 years!

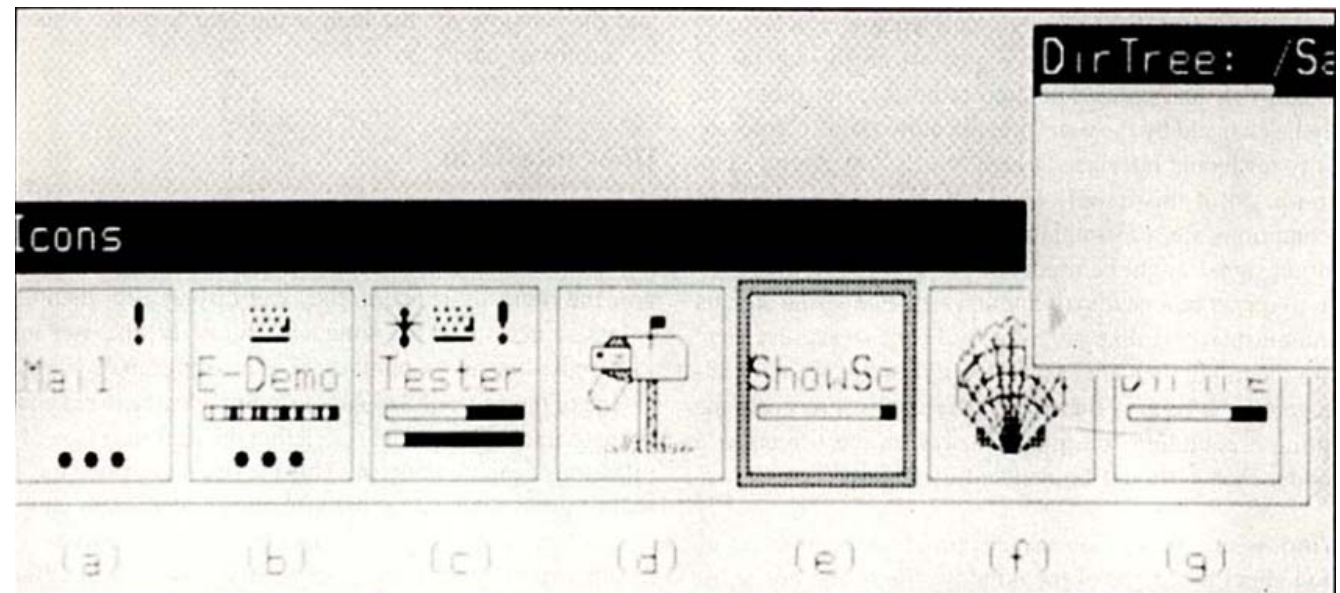


Carnegie Mellon University

Sapphire for PERQ



- 1983
- **SAPPHIRE**
Screen
Allocation
Package
Providing
Helpful
Icons and
Rectangular
Environments



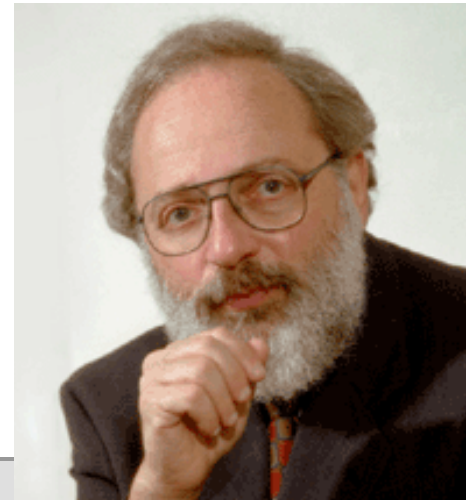
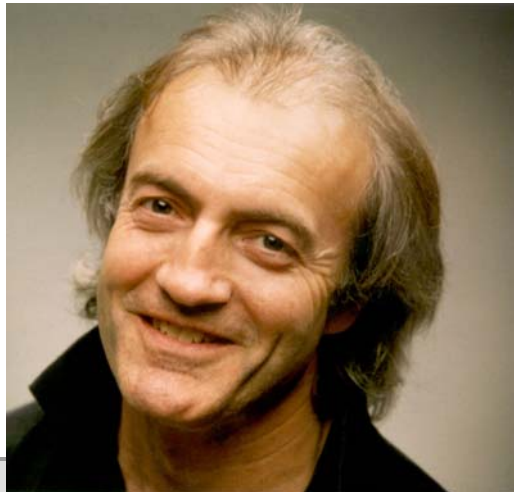
1983 - 1987



Carnegie Mellon University

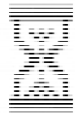


Computer Science UNIVERSITY OF TORONTO



Percent-Done Progress Indicators

- CHI'1985 paper



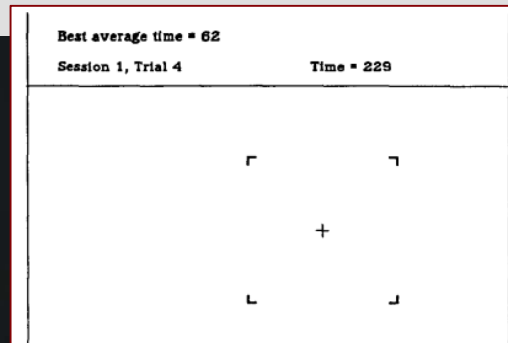
Static busy signals



Progress indicators are better

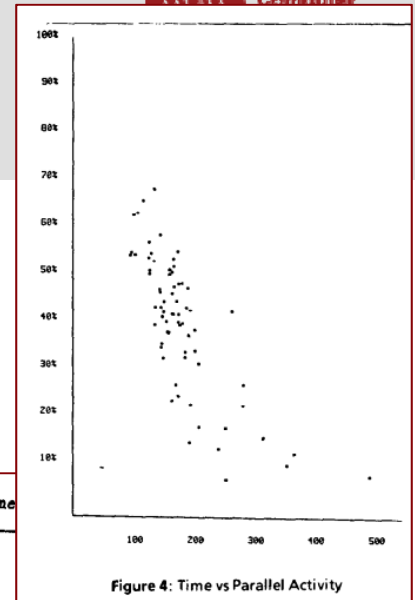
Two-Handed Input

[Buxton and Myers, CHI'1986]



Fastest Time = 78 secs. Your elapsed time

Screen dump	Middle	Right
6 Left	Middle	Right 10
7 Left	Middle	Right 11
8 Left	Middle	Right 12
9 Left	Middle	Right 13
10 Left	Middle	Right 14
11 Left	Middle	Right 15
12 Left	Middle	Right 16
13 Left	Middle	Right 17
14 Left	Middle	Right 18
15 Left	Middle	Right 19
16 Left	Middle	Right 20
17 Left	Middle	Right 21



--- CORRECT ---
 Select line 28, Left.

Taxonomy, 1986

- Programming by Example vs. with Example
- Visual Programming vs. Program Visualization

CHI'86 Proceedings

April 1986

**Visual Programming, Programming by
Example,
and
Program Visualization:
A Taxonomy.**

Brad A. Myers

Dynamic Graphics Project
Computer Systems Research Institute
University of Toronto
Toronto, Ontario, M5S 1A4
Canada

1. Introduction

As the distribution of personal computers and the more powerful personal workstations grows, the majority of computer users now do not know how to program. They

ABSTRACT



Human-
Computer
Interaction
Institute

Carnegie Mellon University

Observations

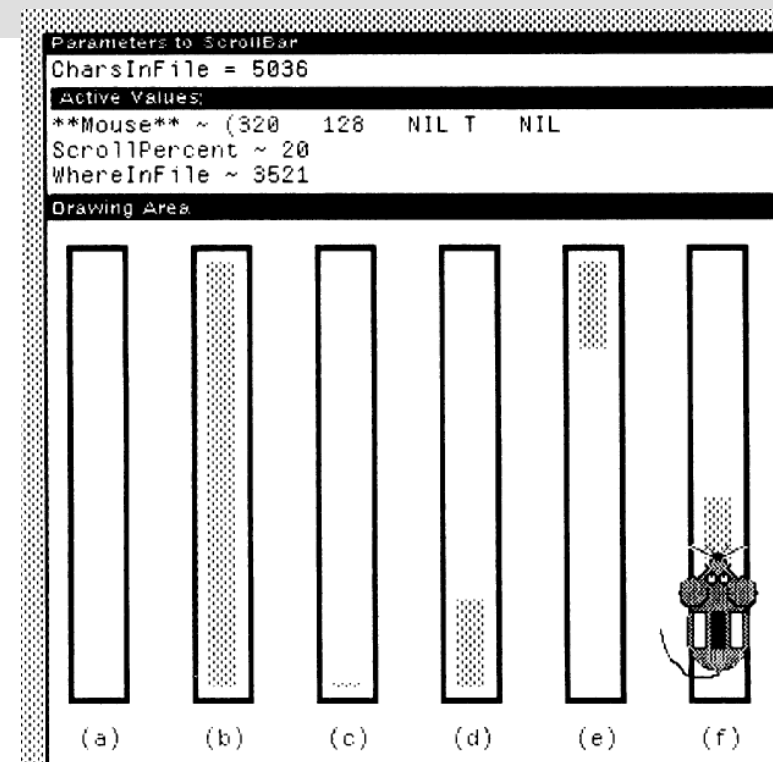
- A good taxonomy or survey can be a nice contribution

Peridot, 1986-87

- **PERIDOT**
Programming by
Example for
Real-time
Interface
Design
Obviating
Typing



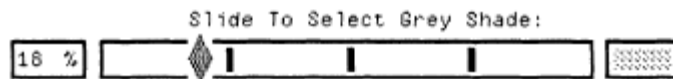
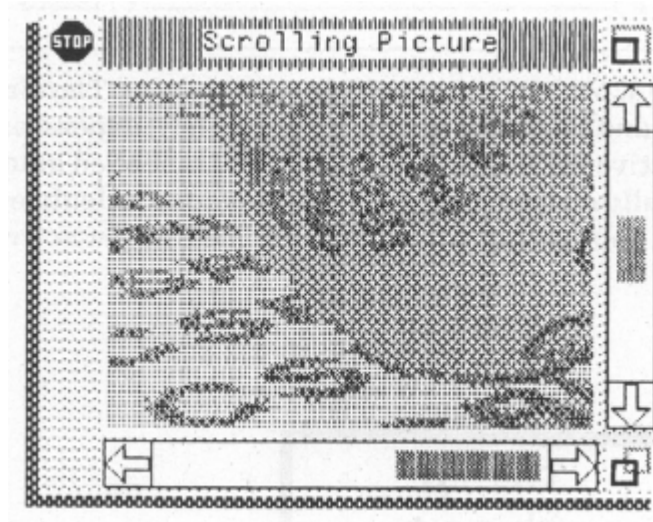
CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=996283>



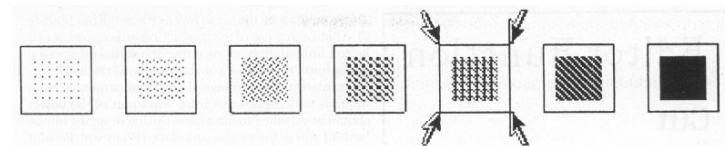
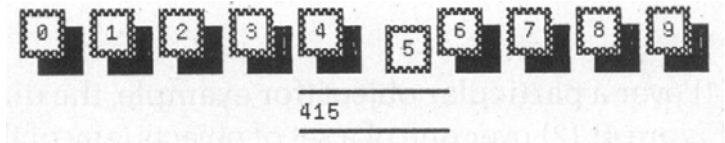
Peridot, 1986-87



- Bold
- Italic
- Underline
- Inverted
- Strike Through
- Shadow
- Outline



- Bold
- Italic
- StrikeThrough
- Underline
- Superscript
- Subscript
- Inverted





Human-
Computer
Interaction
Institute

Carnegie Mellon University

Observations

- If you are first, then you can just show “**it can be done**”
- To show “**better**” requires different evidence
 - User study or measurements
- Showing it can be used by **certain users**
- Depends on what you want to claim



Human-
Computer
Interaction
Institute

Carnegie Mellon University



Peridot

© 2017 – Brad A. Myers

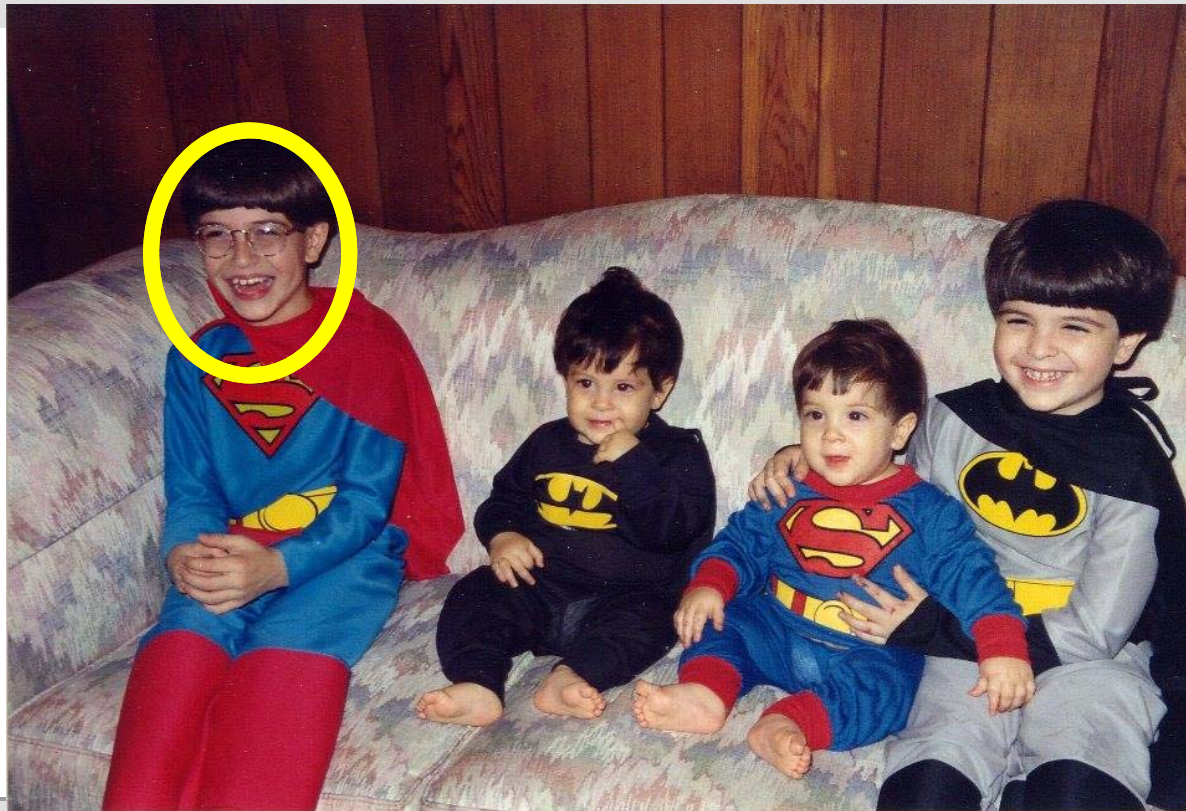
Observations

- Having kids while a graduate student can work out since you have more flexible time than when you are a professor.



Ryan Myers
was born in
Toronto,
Canada in
August, 1985

My Four Sons, 1993



Family Today



Carnegie Mellon University

- 2016



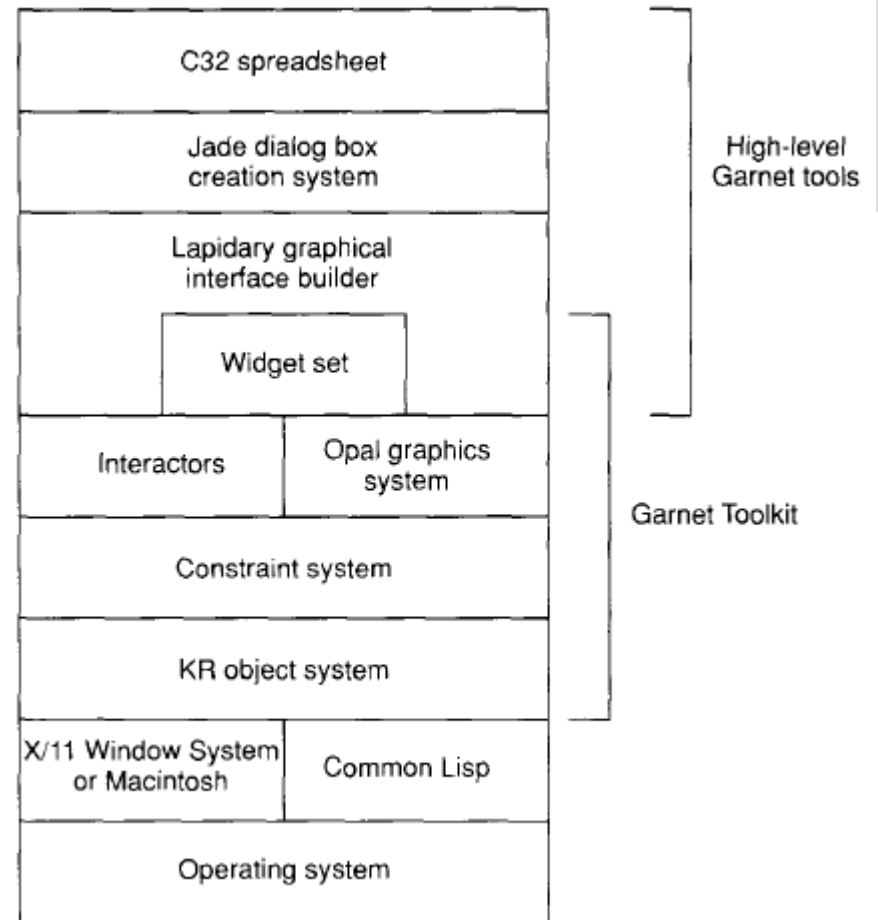
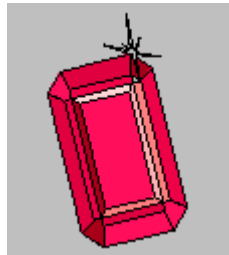
Carnegie Mellon University, 1987...

- 30 years this summer!



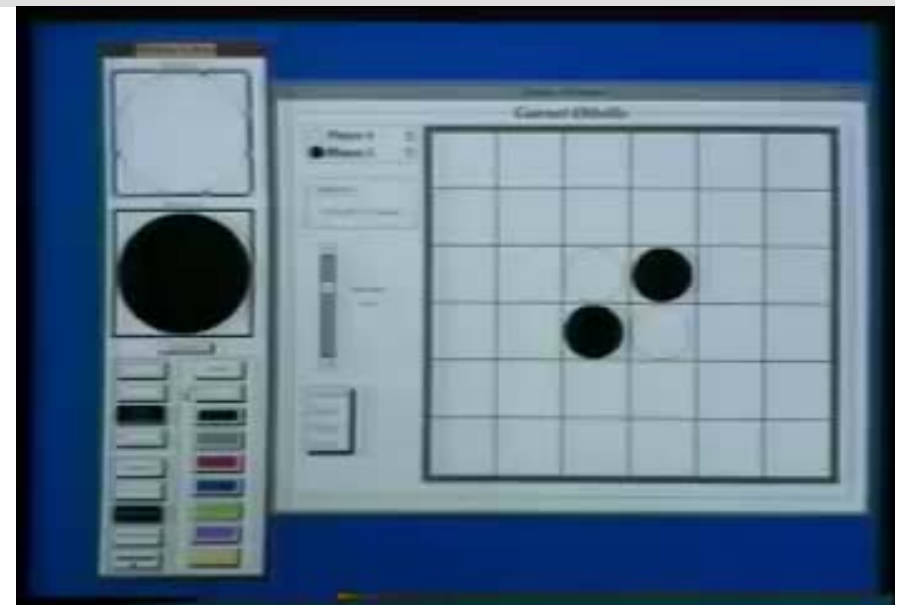
Garnet, 1988 - 1994

- **GARNET**
Generating an
Amalgam of
Real-time,
Novel
Editors and
Toolkits



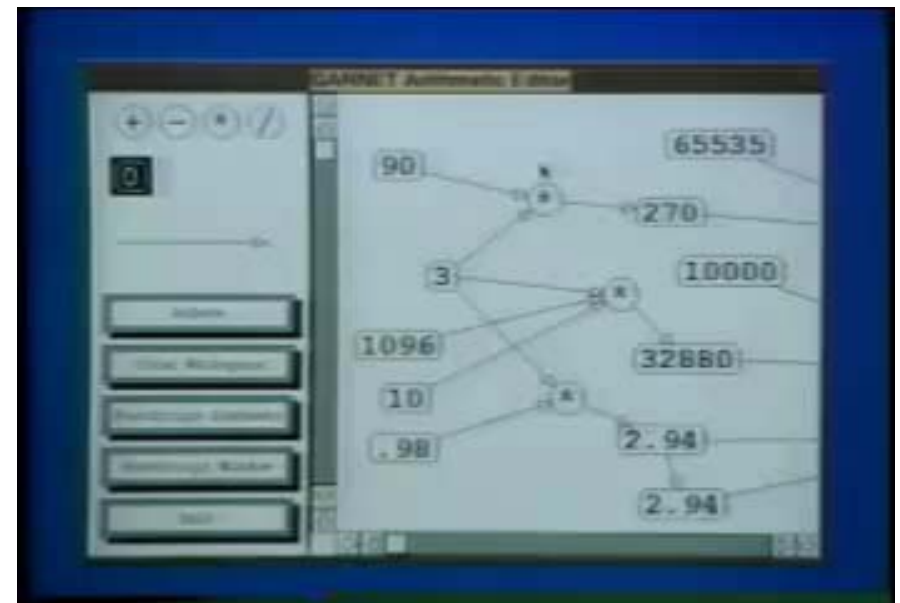
Prototype-Instance Object System

- Any object can be the *prototype* for other objects
- Now used by JavaScript
- Changes to prototype propagate
- **Structural inheritance**



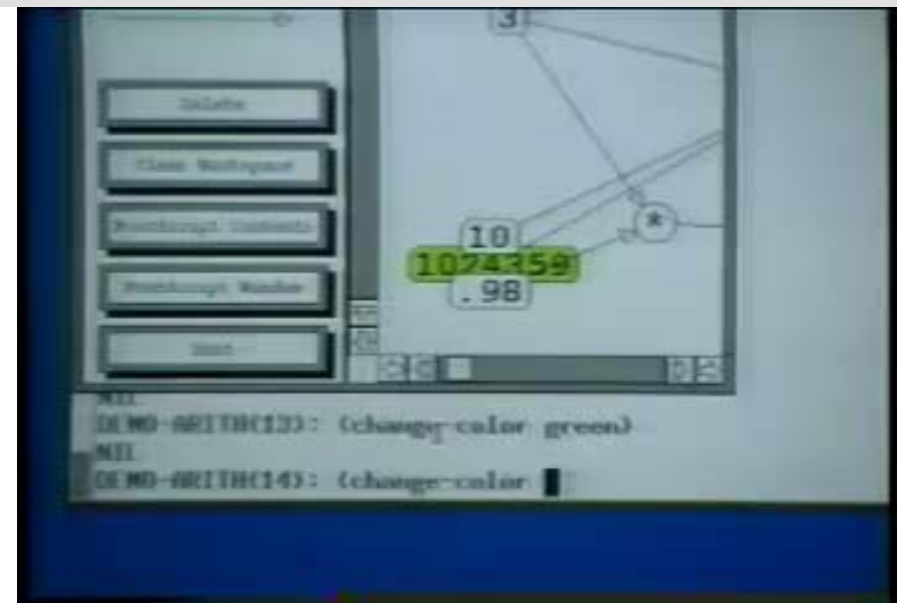
Constraints

- Values automatically computed based on dependencies
- Connect graphics
- Showed useful for **copying values**
 - “Data bindings”
- Introduced “**pointer variables**” for constraints
 - E.g., “same size as *whatever is selected*”



Retained Object Model

- “Display list” for graphics
- Automatically handles refresh
- Like DOM for web pages today



“Interactors” Input Model

- Parameterized objects for each kind of behavior
- Key behaviors used in GUIs
- Independent of the graphics
- Reusable for *all* interactions
- Never need to write event handlers
- Concept used by Adobe Flash Catalyst, etc.

- Menu-Interactor—select,
- Move-Grow-Interactor—position,
- New-Point-Interactor—position,
- Angle-Interactor—orient,
- Text-Interactor—text,
- Trace-Interactor—path.



Higher Level Tools in Garnet

LAPIDARY

Lisp-based
Assistant for
Prototyping
Interface
Designs
Allowing
Remarkable
Yield

AGATE

A
Gesture-recognizer
And
Trainer by
Example

JADE

Judgment-based
Automatic
Dialog
Editor

GILT

Graphical
Interface
Layout
Tool

MARQUISE

Mostly
Automated,
Remarkably
Quick
User
Interface
Software
Environment

C32
 CMU's
 Clever and
 Compelling
 Contribution to
 Computer Science in
 CommonLisp which is
 Customizable and
 Characterized by a
 Complete
 Coverage of
 Code and
 Contains a
 Cornucopia of
 Creative
 Constructs, because it
 Can
 Create
 Complex,
 Correct
 Constraints that are
 Constructed
 Clearly and
 Concretely, and
 Communicated using
 Columns of
 Cells, that are
 Constantly
 Calculated so they
 Change
 Continuously, and
 Cancel
 Confusion.

C32



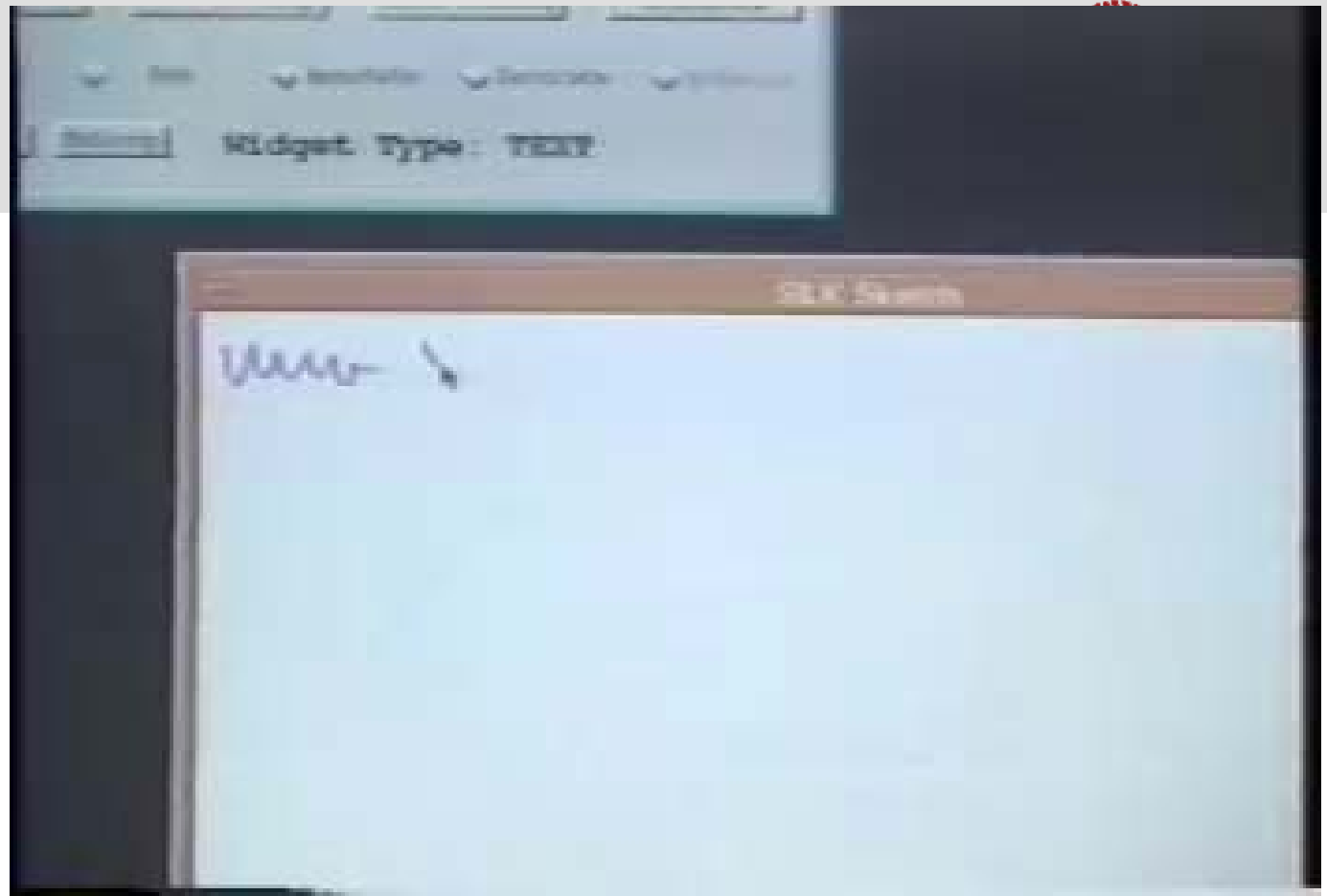
Carnegie Mellon University

- CHI'1991

↑	MY-RECTANGLE	↑	STRING1	↑	ARROW
↑	:Left 10	↑	:String "C32"	↑	:X1
↑	:Top 10	↑	:Font OPAL:DEFAULT-F... ⓘ	↑	:Y1
	:Width 50		:Left 25 ⓘ		:X2
	:Height 50		:Top 28 ⓘ		:Y2
	:Visible ⓘ ⓘ T		:Width 21 ⓘ ⓘ		:Left
	:Line-Style OPAL:DEFAULT-L... ⓘ		:Height 14 ⓘ ⓘ		:Top
	:Filling-Style NIL ⓘ		:Visible ⓘ ⓘ T		:Width
	:Draw-Function :COPY ⓘ		:Line-Style OPAL:DEFAULT-L... ⓘ		:Height
	:Window W		:Fill-Backgrou NIL ⓘ		:Visibl
↓	:Parent A	↓	:Actual-Height NIL ⓘ	↓	:Line-S
↓	:Is-A OPAL:RECTANGLE	↓	:Draw-Function :COPY ⓘ	↓	:Filling
		↓	:Window W	↓	:Draw-F

Silk

- James Landay
PhD, 1996
- **SILK**
Sketching
Interfaces
Like
Krazy



3 Generations of CHI Academy



- Baecker, Buxton, Myers, Landay



Carnegie Mellon University

Programming By Demonstration

TOURMALINE

Text-formatting

Ought to

Use and

Rely on

Macrostyles

And

Layout

Inferred

Nicely by

Example

GAMUT

Games

Are

Made

Using

This

Programming By Demonstration

TOURMALINE

Text-formatting

Ought to

Use and

Rely on

Macrostyles

And

Layout

Inferred

Nicely by

Example

GAMUT

Games

Are

Made

Using

This

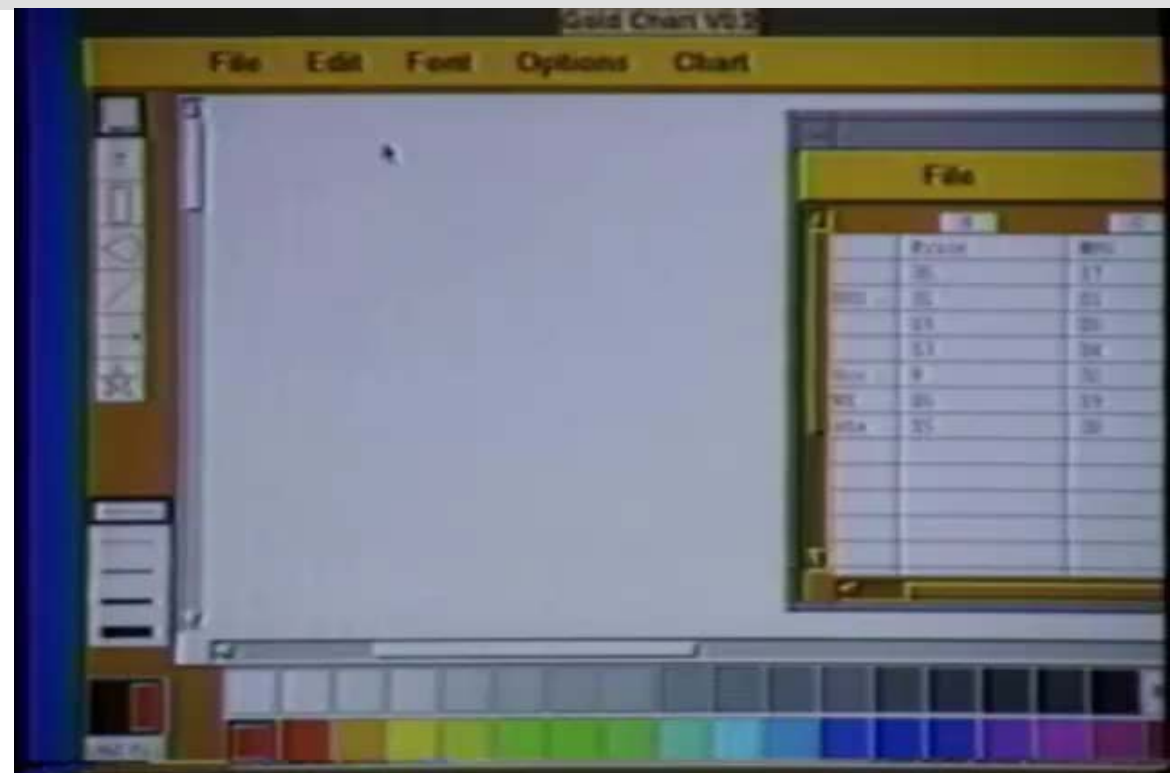
GOLD

Graphs and

Output

Laid-out by

Demonstration




All the Widgets, CHI'1990



Carnegie Mellon University

- Scroll Bars
- Menus
- Palettes
- Command Buttons
- Radio Buttons
- Check Boxes
- Text Selection
- Basic Text Editing
- Dialog Boxes
- Selecting Graphics
- Basic Editing of Graphics
- Desktops
- Window Manager
- Commands



Moving around:
"Scroll Bars"

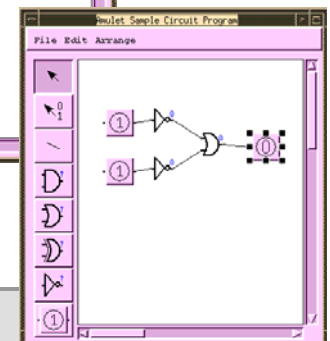
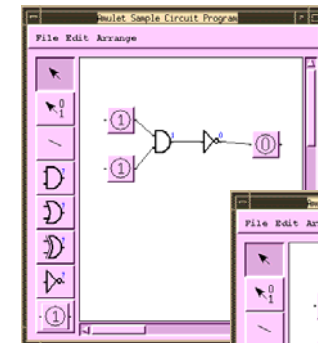
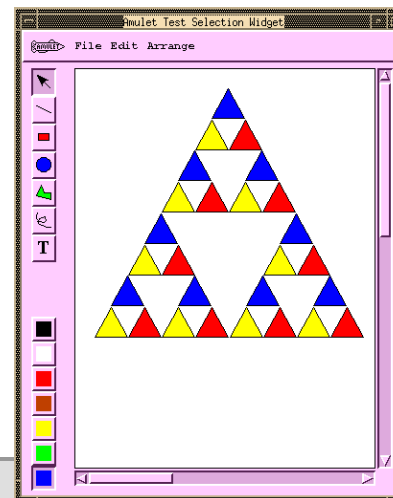
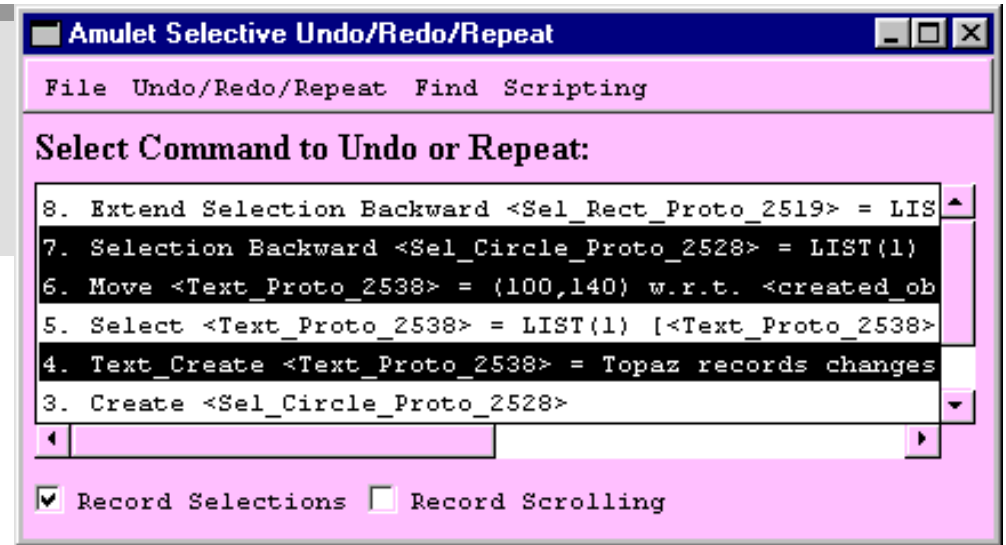
Amulet: 1994 – 1998

- **AMULET**
Automatic
Manufacture of
Usable and
Learnable
Editors and
Toolkits



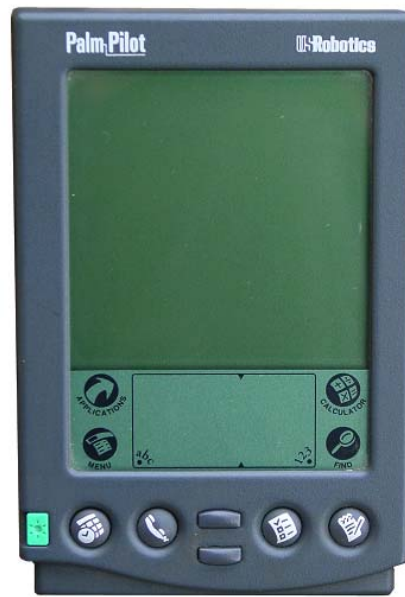
Command Objects

- TOPAZ
Transcripts
Of
Programs
Activated with
Zeal



Pebbles, 1998 -

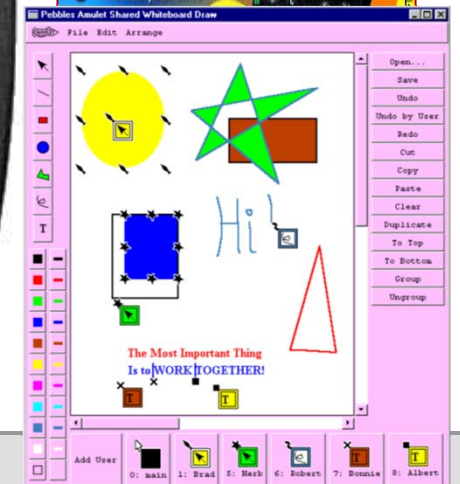
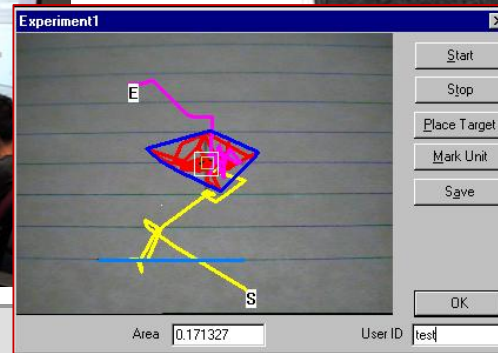
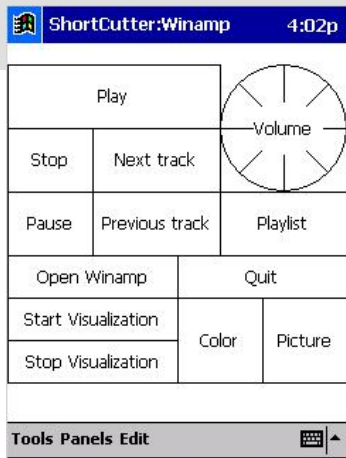
- **PEBBLES**
PDAs for
Entry of
Both
Bytes and
Locations from
External
Sources



Pebbles Projects



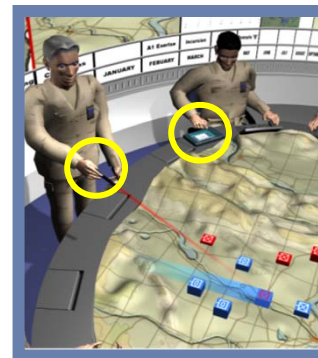
Carnegie Mellon University



Pebbles Projects

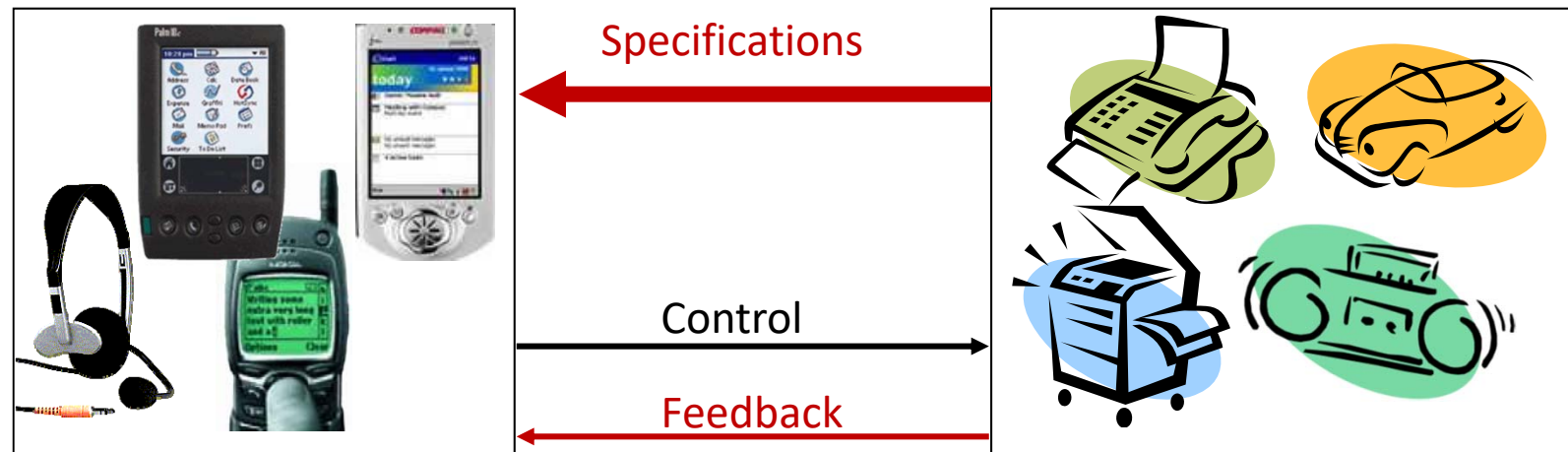


- Command Post of the Future
 - Successful DARPA program
 - Expected to be about speech/gestures
 - Actual issues were mainly situation awareness
 - Including connected mobile devices

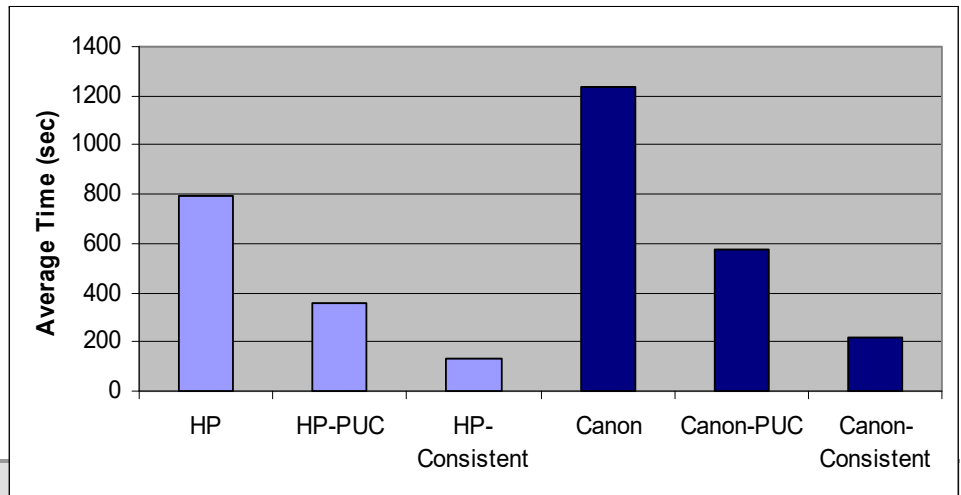


Personal Universal Controller (PUC)

- Jeff Nichols PhD, 2006

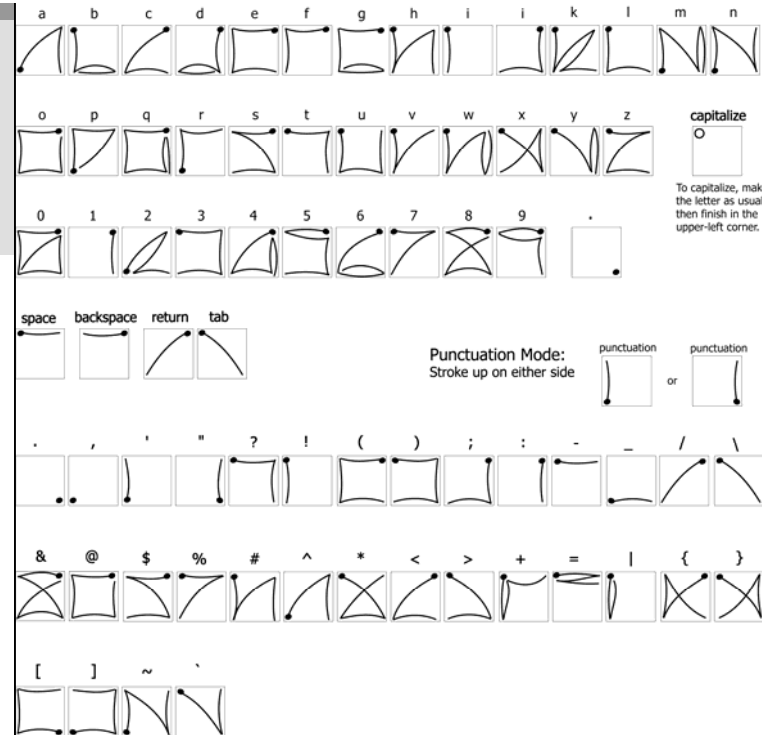


Personal Universal Controller (PUC)



EdgeWrite

- Jake Wobbrock, PhD, 2006



Sugilite

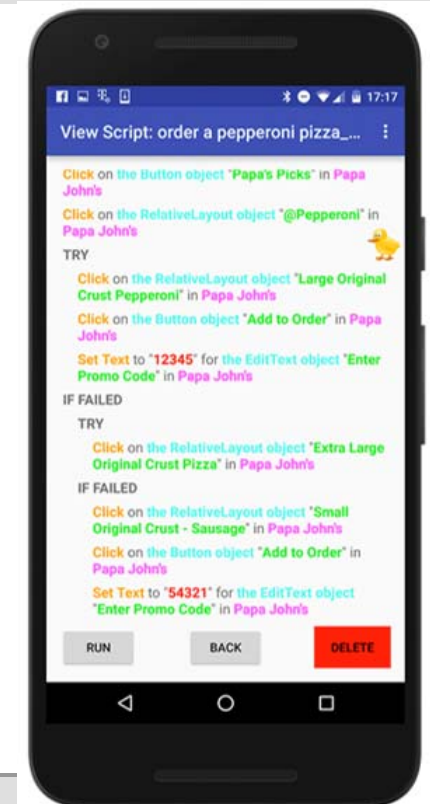
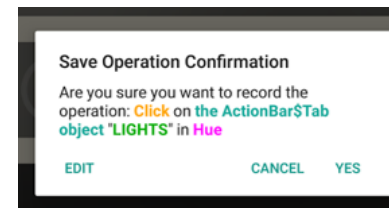
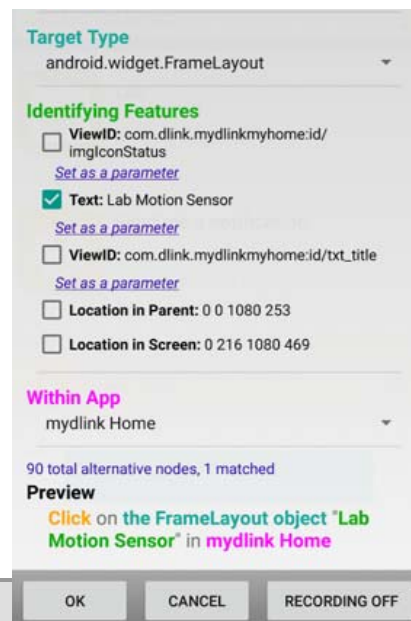


Carnegie Mellon University



- Current PhD work of Toby Li
 - See our paper, Thurs, 11:30 - 12:50 in Room 203

• **SUGILITE**
(come to the talk to see what it stands for!)



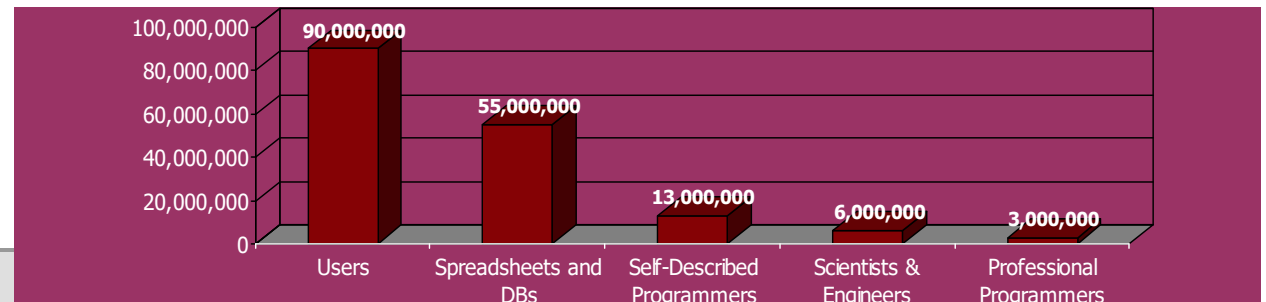
Natural Programming, 1995-



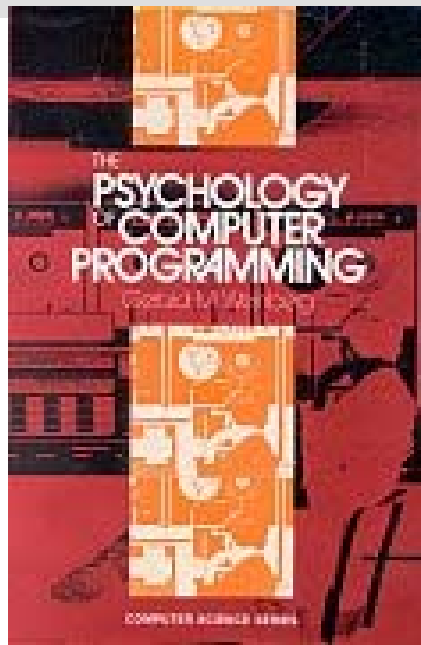
- Make programming easier and more correct by making it more *natural*
- Closer to the way that people think about algorithms and solving their tasks
- Novice, expert, and “end-user” programmers



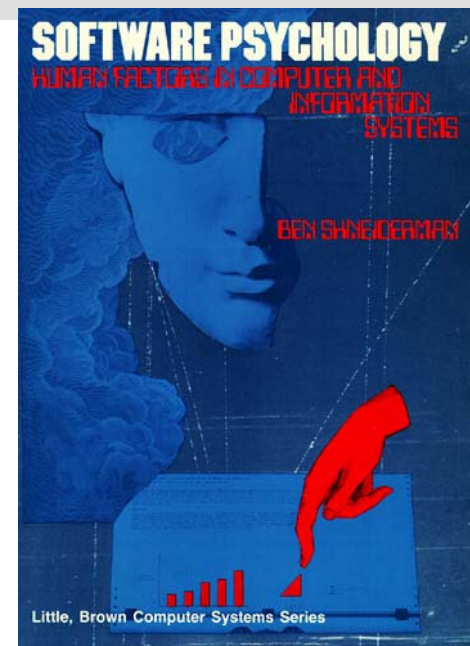
[Scaffidi, Shaw and Myers 2005]



HCI Study of Programmers



1973



1980

Allen Newell and Stuart Card, 1985

- “Millions for compilers but hardly a penny for understanding human programming language use. Now, programming languages are obviously symmetrical, the computer on one side, the programmer on the other. In an appropriate science of computer languages, one would expect that half the effort would be on the computer side, understanding how to translate the languages into executable form, and half on the human side, understanding how to design languages that are easy or productive to use.... The human and computer parts of programming languages have developed in radical asymmetry.”



Human-
Computer
Interaction
Institute

Carnegie Mellon University

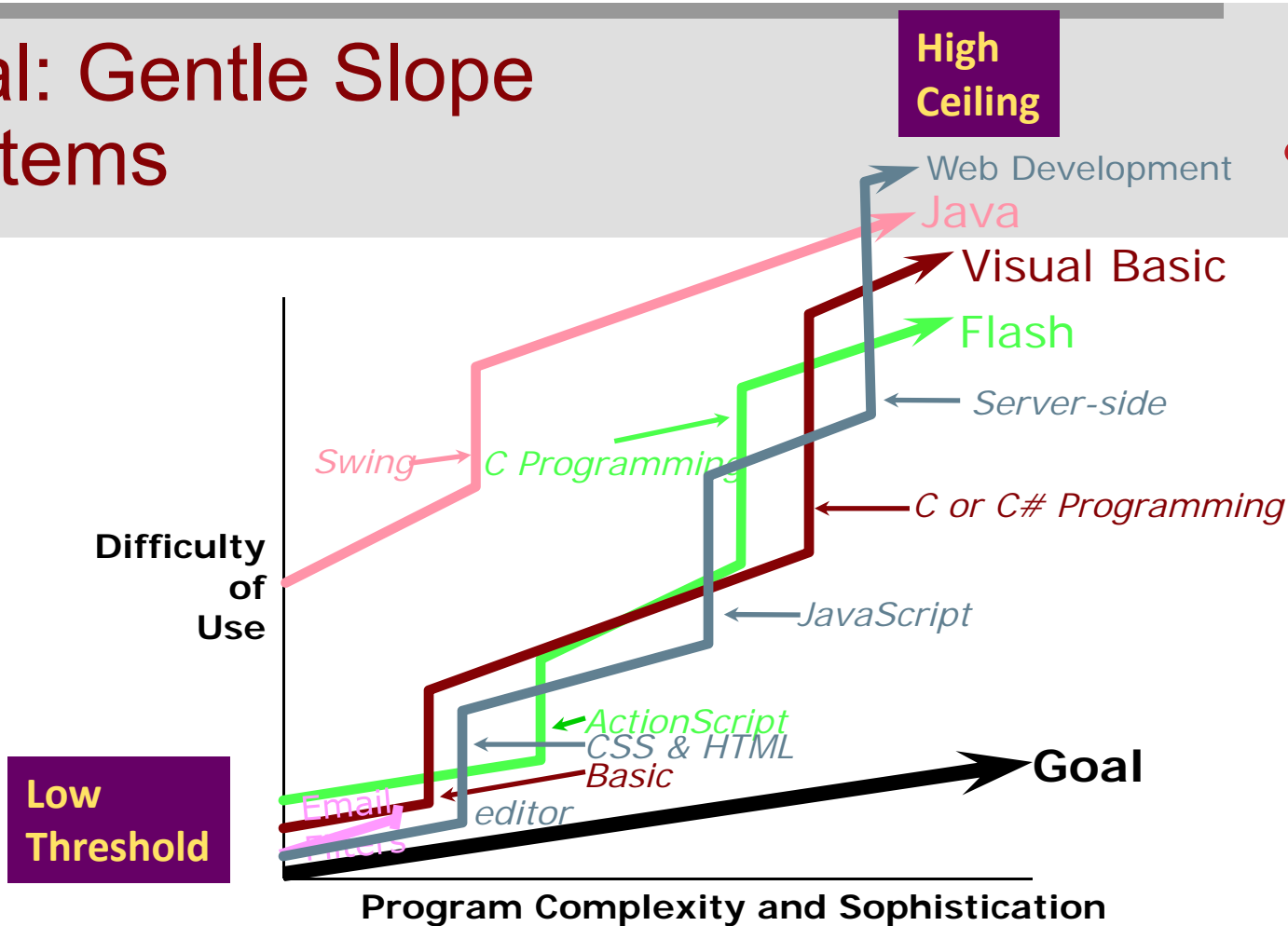
Observations:

- Intuitions of programmers about what would be helpful are often wrong
- Need to do studies of programmers

Goal: Gentle Slope Systems



Carnegie Mellon University





Human-
Computer
Interaction
Institute

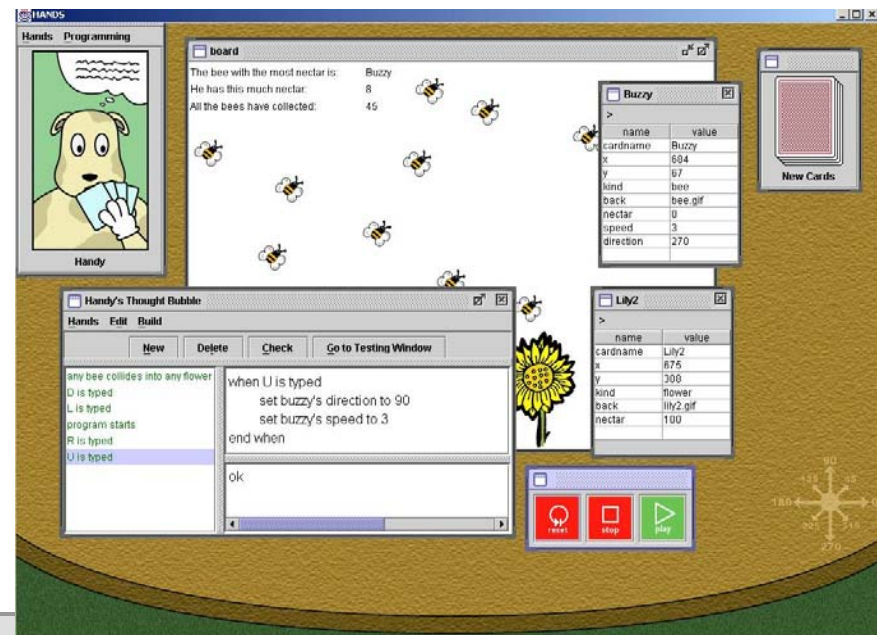
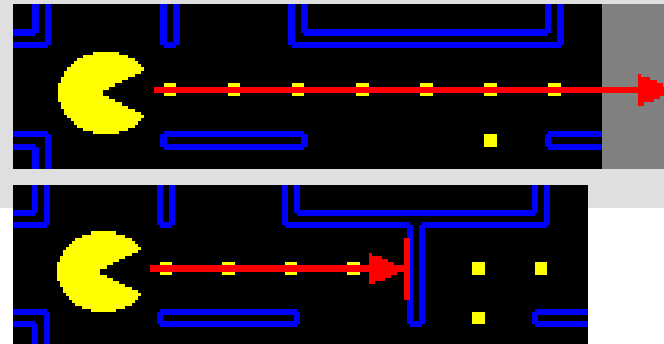
Carnegie Mellon University

Making Programming Easier

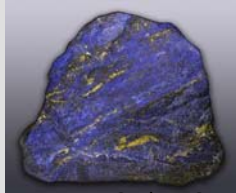
- Key focus of most of my research
- Life-long pursuit

HANDS

- John Pane, PhD 2002
- **HANDS**
Human-Centered
Advances for
Novice
Development of
Software



Lapis

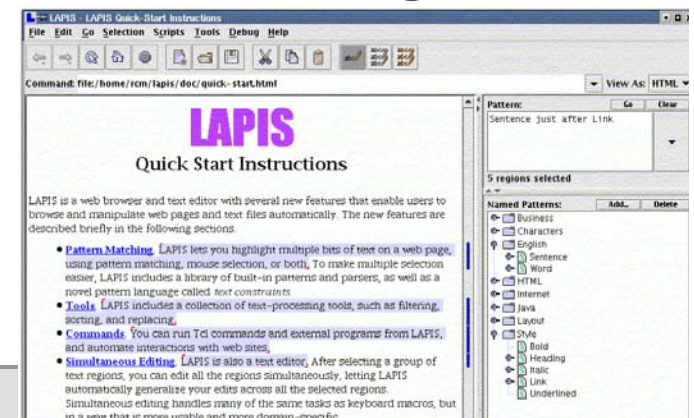


By Hannes Grobe - Own work, CC BY-SA 2.5, <https://commons.wikimedia.org/w/index.php?curid=3415430>



Carnegie Mellon University

- Rob Miller, PhD 2002
- **LAPIS**
Lightweight
Architecture for
Processing
Information
Structure
- Informal parsing
and reuse of semi-
structured text databases
- Simultaneous editing
- Outlier
finding



Whyline

- Andy Ko, PhD, 2008
- **WHYLINE**
Workspace that
Helps
You
Link
Instructions to
Numbers and
Events
- Causes of software defects
- Understanding software maintenance tasks
- Working sets when programming
- Navigation takes 35% of the time

Whyline



Carnegie Mellon University

The screenshot shows the Whyline interface for a Pac-Man simulation. At the top, there are controls for 'Resume', 'Stop', and 'Why...'. The main area displays a 3D scene of Pac-Man on a green field with a yellow Pac-Man character and a blue ghost. A tooltip over the Pac-Man character reads 'further questions can be asked'. Below the scene, a code editor shows a 'When' block with the condition 'Pac is within 1 meter of Big Dot becomes true'. A tooltip over this block says 'camera focuses on subject of question'. The code editor also shows a 'Do in order' block with actions like 'Big Dot set isShowing to false' and 'Big Dot.isEaten set value to true'. A tooltip over the code editor says 'code related to the selection is highlighted'. On the left, there are panels for 'Pac's details' and 'World.move Pac'. A tooltip over the 'World.move Pac' block says 'runtime actions'. At the bottom, a 'Question: Why didn't Pac resize 0.5?' is displayed, with an 'Answer' section. A flowchart below the question shows a sequence of events: 'Big Dot.isEaten set to true' (3.821010), 'isEaten true', 'Pac is within 2 of Ghost true', 'and', 'Doing else' (3.854011). A tooltip over the flowchart says 'causality arrows'. A tooltip over the 'Doing else' node says 'time cursor traverses execution history'. A tooltip over the 'Questions I've asked' panel says 'access to previous questions and answers'. A tooltip over the 'World.move Pac' block says 'tooltips show properties current values'.

The screenshot shows the Whyline interface for a Java Paint application. The main area displays a 'PaintWindow #1,785' with a canvas showing a green circle and a black line. A tooltip over the green circle reads 'properties of this line' and lists 'why did x1 = 88?', 'why did y1 = 185?', 'why did x2 = 93?', 'why did y2 = 169?'. A tooltip over the black line reads 'why did color = ...', 'why did font = Dialog 12 pt?', 'why did stroke = 5.0 pixel stroke?'. The interface includes a toolbar with 'Pencil', 'Eraser', and 'Line' tools, and a color palette with 'Red', 'Green', and 'Blue' options. A tooltip over the toolbar says 'access to previous questions and answers'. At the bottom, there is a 'showing all to events' section and an 'Ask' section with buttons for 'show code info', 'show execution info', and 'send feedback'.

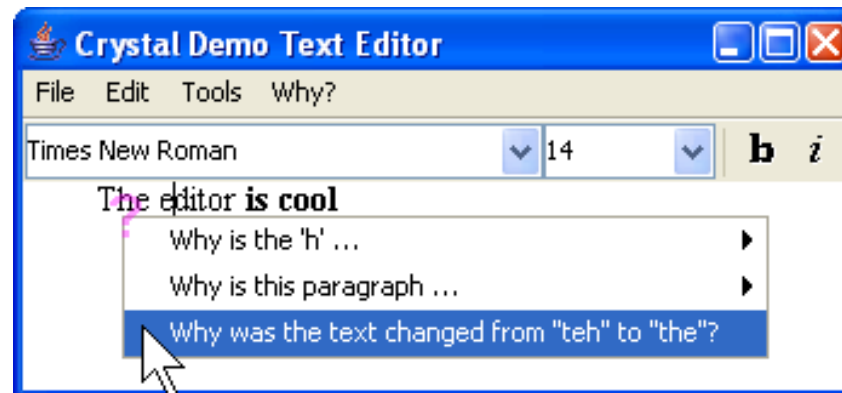
Crystal



Carnegie Mellon University



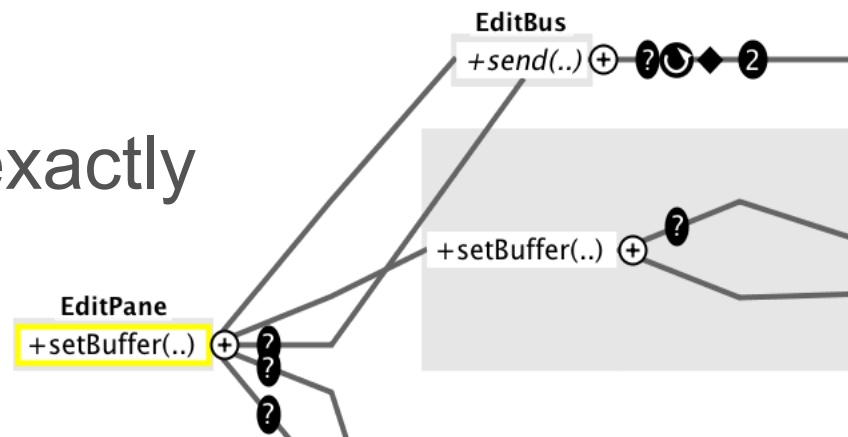
- Ask “why” for everything else!
- **CRYSTAL**
Clarifications
Regarding
Your
Software using a
Toolkit,
Architecture and
Language



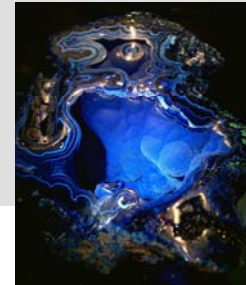
70

Understanding Unfamiliar Code

- Thomas LaToza, PhD 2012
- Hard to answer questions about code:
 - *Control flow* = “reachability”
 - 100s of others
- “Reacher” visualizes exactly the paths of interest



Backtracking During Editing



Carnegie Mellon University

- YoungSeok Yoon, PhD 2015
- **AZURITE**
Adding
Zest to
Undoing and
Restoring
Improves
Textual
Exploration

The screenshot displays two views from an IDE. The top view, titled 'Code History Diff', compares the current version of 'DrawingEditor.java' (lines 2-4) with a previous version from Feb 28, 2013. The current version has `int width = 1000;` and `int height = 600;`, while the previous version has `int width = 800;` and `int height = 600;`. The bottom view, titled 'Timeline View', shows a horizontal timeline for three files: 'DrawingEditor.java', 'Text.java', and 'Line.java'. A red vertical line marks a specific point in time at 10:48 AM on 02/28/2013, with a red triangle pointing to it. Other time markers on the timeline include 10:23 AM, 10:49 AM, 10:52 AM, and 10:57 AM, all on 02/28/2013.



Human-
Computer
Interaction
Institute

Carnegie Mellon University

Observations

- Viewing the code is central!
 - Visualizations of code are mostly useful as navigation aides
- Search is useful across many dimensions, e.g.:
 - Search along control flow
 - Search backwards in time



API Usability

- Jeff Stylos, PhD 2009
- **Application Programming Interface**
 - Libraries, frameworks, SDKs, ...
 - Now: web services, middleware, ...
 - **User interface** between programmer and functionality in code
- Patterns that decrease usability
- New documentation tools to compensate

API Usability Tools

MICA
Makes
Interfaces
Clear and
Accessible



JADEITE
Java
API
Documentation with
Extra
Information

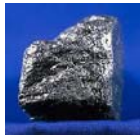


CALCITE
Construction
And
Language
Completion
Integrated
Throughout
Eclipse



EUKLAS
Eclipse
Users'
Keystrokes
Lessened by
Attaching
Samples

GRAPHITE
GRAphical
Palettes
Help
Instantiate
Types in the
Editor



Tacked-on for
Emphasis

APATITE
Associative
Perusing of
APIs
That
Identifies
Targets
Easily



DACITE
Design
Annotations for
Complementing
Interfaces
Targeting
Effectiveness

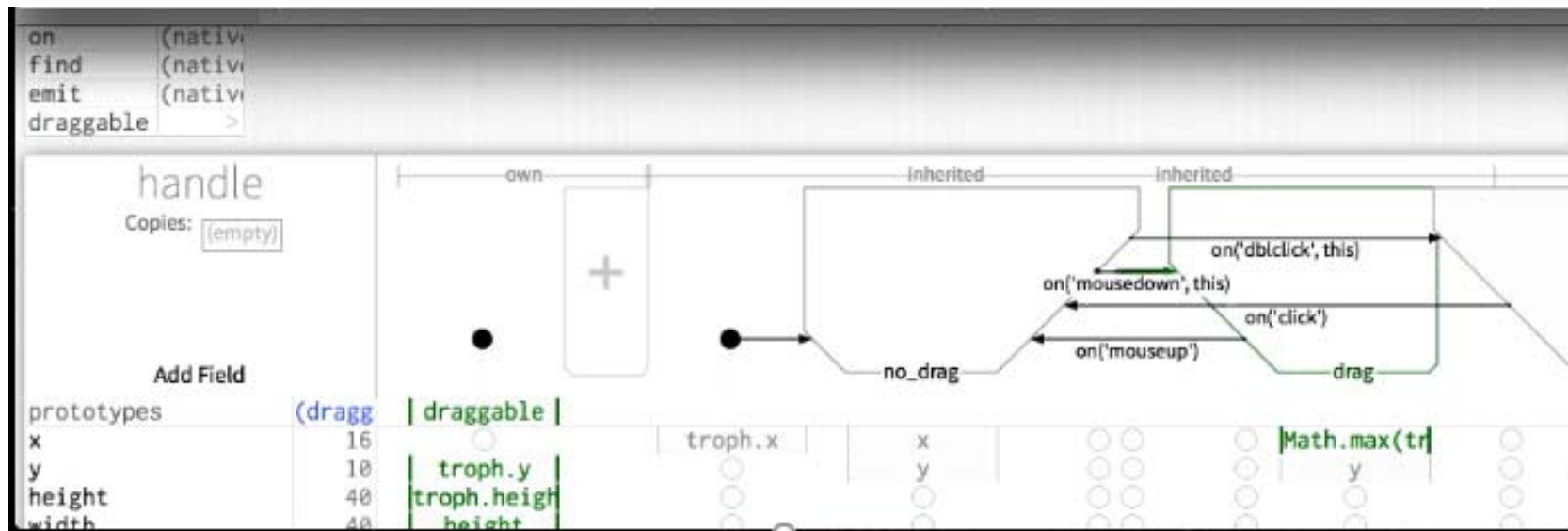
Glacier and Obsidian

- **GLACIER**
 - Great Languages
 - Allow Class Immutability
 - Easily and Readily
- **OBSIDIAN**
 - Object-oriented Blockchain State
 - Interaction and Development Implementation
 - And Notation
- Michael Coblenz's PhD work
- Key Goal: Design more usable programming languages
- Glacier:
 - Immutability – declare that instances cannot change after constructed
- Obsidian:
 - Easier blockchain programming

InterState



- Stephen Oney, PhD 2015



Gneiss



- Kerry Chang, PhD 2016
- **GNEISS**
Gathering
Novel
End-User
Internet
Services using
Spreadsheets

1	A (businesses.name)	B (businesses.categories.catd)
1	Coca Cafe	1.1 breakfast_brunch 1.2 newamerican
2	Waffles Incaffeinated	2.1 breakfast_brunch 2.2 newamerican 2.3 tradamerican
3	Point Brugge Café	3.1 belgian
4	The Dor-Stop Restaurant	4.1 breakfast_brunch 4.2 diners
5	Deluca's Diner	5.1 breakfast_brunch

2	A (businesses.name)	B (businesses.categories.catd)
1	Coca Cafe	1.1 breakfast_brunch 1.2 newamerican
2	Waffles Incaffeinated	2.1 breakfast_brunch 2.2 newamerican

3	A (businesses.categories)	B (businesses.name)
1	belgian	Point Brugge Café
2	belgian	Park Bruges
3	breakfast_brunch	Coca Cafe
4	breakfast_brunch	Waffles Incaffeinated

File Edit Setting

on?query={A1}&key=AlzaSyB6h57

Selected (click) 1 edit: results[B].name

When moving an item to the spreadsheet, populate the column with similar items in the array.

```

{
  "bar",
  "establishment"
}
[
  {
    formatted_address : "10 Colum
    geometry :
      location :
        lat : 40.768718,
        lng : -73.982489
      },
    icon : "http://maps.gstatic.c
    id : "94f9a538183c6a67c07e435
    name : "Dizzy's Club Coca Col
    opening_hours :
      open_now : false
    },
    photos :
      [
        {
          height : 960,
          html_attributions :
            photo_reference : "
          width : 1200
        }
      ],
    price_level : 4,
    rating : 4.4,
    reference : "CoQBeQAANsg3EEK
    types :
      "night_club",

```

2

A	B (name)	C (rating)	D (price_level)	E (formatted)	F	G
1	Jazz bar New York City	4.4	4	10 Columbus Cir #5, New York, NY, United States	false	
2	Little Branch	4.3	3	20 7th Ave S, New York, NY, United States	false	
3	Louis 649	4.2	2	649 E 9th St, New York, NY, United States	true	649 E 9th St, New York, NY, United States
4	Garage Restaurant & Cafe	4.1	2	99 7th Ave S, New York, NY, United States	false	
5	Jazz Standard	4.1	3	116 E 27th St, New York, NY, United States	true	116 E 27th St, New York, NY, United States
6	BIRDLAND	4	3	315 W 44th St, New York, NY, United States	true	315 W 44th St, New York, NY, United States
7	The Flatiron Room	4	3	37 W 28th St, New York, NY, United States	false	
8	La Lanterna di Vittorio	4	2	129 Macdougall St, New York, NY, United States	false	
9	Smoke Jazz & Supper Club	4	2	2751 Broadway, New York, NY, United States	false	
10	BLACK DUCK	4	3	122 E 28th St, New York, NY, United States	false	
11	Knickerbocker Bar & Grill	3.9	3	33 University Pl, New York, NY, United States	false	

3

Places to Go

Jazz bar New York City

Sort descending by rating | Sort descending by price

4

UI elements

- Vertical List
- Grid List
- Bar Chart
- Line Chart
- Pie Chart
- Scatter Chart
- Bubble Chart
- Map

5

Value	Jazz bar New York City
Search here!	
Width	200px
Live	false
Focused	false
State	true
Inline	false

Variolite



Carnegie Mellon University



- Current PhD work of Mary Beth Kery
 - See our paper, Tuesday, 9:30 - 10:50 in Room 111/113

- **VARIOLITE**
(come to the talk to see what it stands for!)

- **Exploratory Programming**

```
driverTest.py
▶ | variants

1 import matplotlib.pyplot as pyplot
2 import numpy as np
3 import math
4
5
6
7 def distance(x0, y0, x1, y1):
8     return math.sqrt((x1-x0)**2 + (y1-y0)**2)
9
10 def computeAngle (p1, p2):
11     dot = 0
12     if computeNorm(p2[0], p2[1]) == 0 or computeNorm(p1[0], p1[1]) == 0:
13         dot = 0
14     else:
15         dot = (p2[0]*p1[0]+p2[1]*p1[1])
16             /float(computeNorm(p1[0], p1[1])*computeNorm(p2[0], p2[1]))
17     if dot > 1:
18         dot = 1
```



Human-
Computer
Interaction
Institute

Carnegie Mellon University

Observations

- Always make a video – demos will stop working
- Having a group culture, fun "signature", e.g. for acronyms and gemstones, ribbons
- One good idea makes attending a conference or reading an article worthwhile

Thanks!

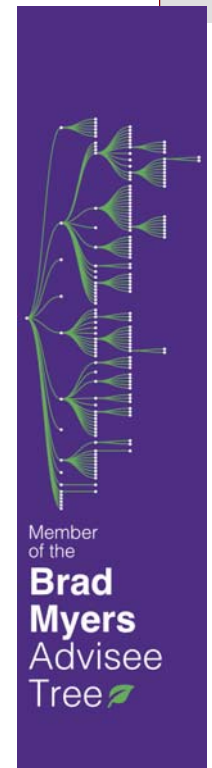
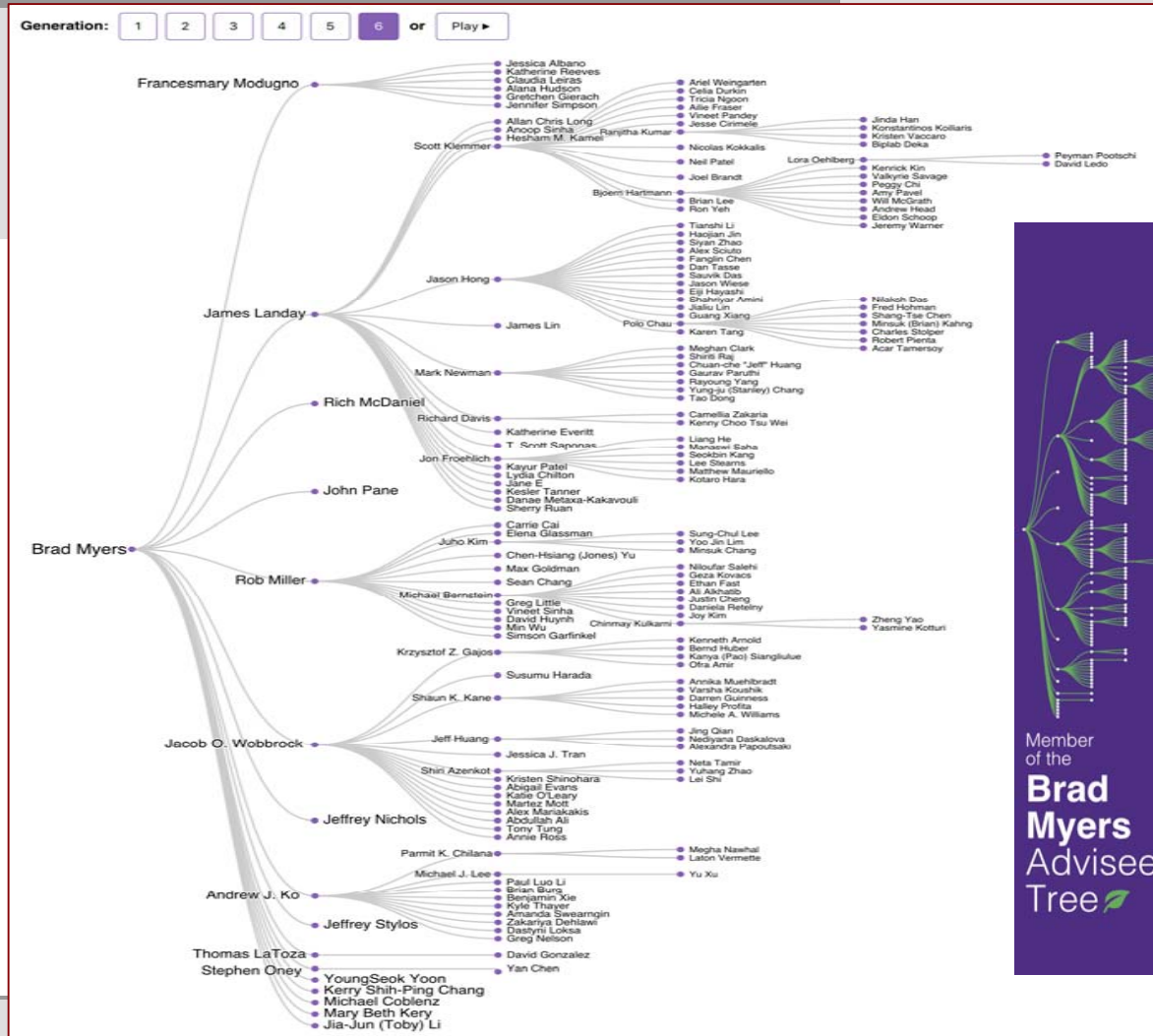
- Funding from Xerox PARC, UofT Fellowship, DARPA, NSF, NIH, Microsoft, SAP, Google, Adobe, IBM, Apple, and many others
- My wife and family
- My mentors, colleagues, and especially the hundreds of students
 - Bachelors, Masters, and PhD at CMU

Advisee tree

Generation	Count
1	Me
2	16
3	61
4	71
5	22
6	2
TOTAL	172

Thanks to: *Fred Hohman* (gen. 5), *Robert Pienta* (gen. 5), and *Polo Chau* (gen. 4) for animation and ribbon design

See: <http://tinyurl.com/myersadviseetree>





Thank You!

RUBY: Reminiscing about User interfaces by Brad over the Years

Brad A. Myers

For More information:

- Brad's home page: bradamyers.com
- Brad's 66 acronyms: <http://www.cs.cmu.edu/~bam/acronyms.html>
- Brad's 172 advisees tree: <http://tinyurl.com/myersadviseetree>