PROGRAMMING LANGUAGES ARE USER INTERFACES

Andrew J. Ko, Ph.D.

Associate Professor	The Information School
Adjunct Associate Professor	Computer Science and Engineering
Chief Scientist	AnswerDash, Inc.

dub

Information School
UNIVERSITY of WASHINGTON

CODE IS CHANGING THE WORLD

010100100100101001010010
100101001010010101001010
101010010101010010101010
100100010010111001010101
001010010101010010010100
101010100010101010010101
001010101001010101100100

BUT THE WORLD ISN'T CHANGING CODE

it's still difficult to learn, write, test, debug, design, deploy, fix etc.

headlines from the last month

Computer Error Costs Indiana Millions In Education Grants

United Continental CEO: Still fixing bugs in new computer system

Computer glitch hampers IMPD communications for 4 days

Computer Glitch Leads to $1 Gas (Sweet!)

ICANN Extends New Domain Deadline Because of Bug

Computer Glitch Means No Licenses, IDs

Computer Glitch Dashed High School Hopes for Five Queens Girls

Computer glitch causes hospital billing errors

Bats CEO Says Computer Glitch `Unfortunate'

State Panel Wants Answers about Prison Computer Glitch

Computer Glitch Delays NJ Jobless Claims developers use the wrong languages

teams lack effective methodologies

CS education fails to adequately prepare

tools fail to compensate for human fallibility developers use the wrong languages

teams lack effective methodologies

CS education fails to adequately prepare

tools fail to compensate for human fallibility

ALL OF THESE ARE **HUMAN** PROBLEMS

because

PROGRAMMING LANGUAGES ARE **USER INTERFACES**

PROGRAMMING LANGUAGES ARE **USER INTERFACES**

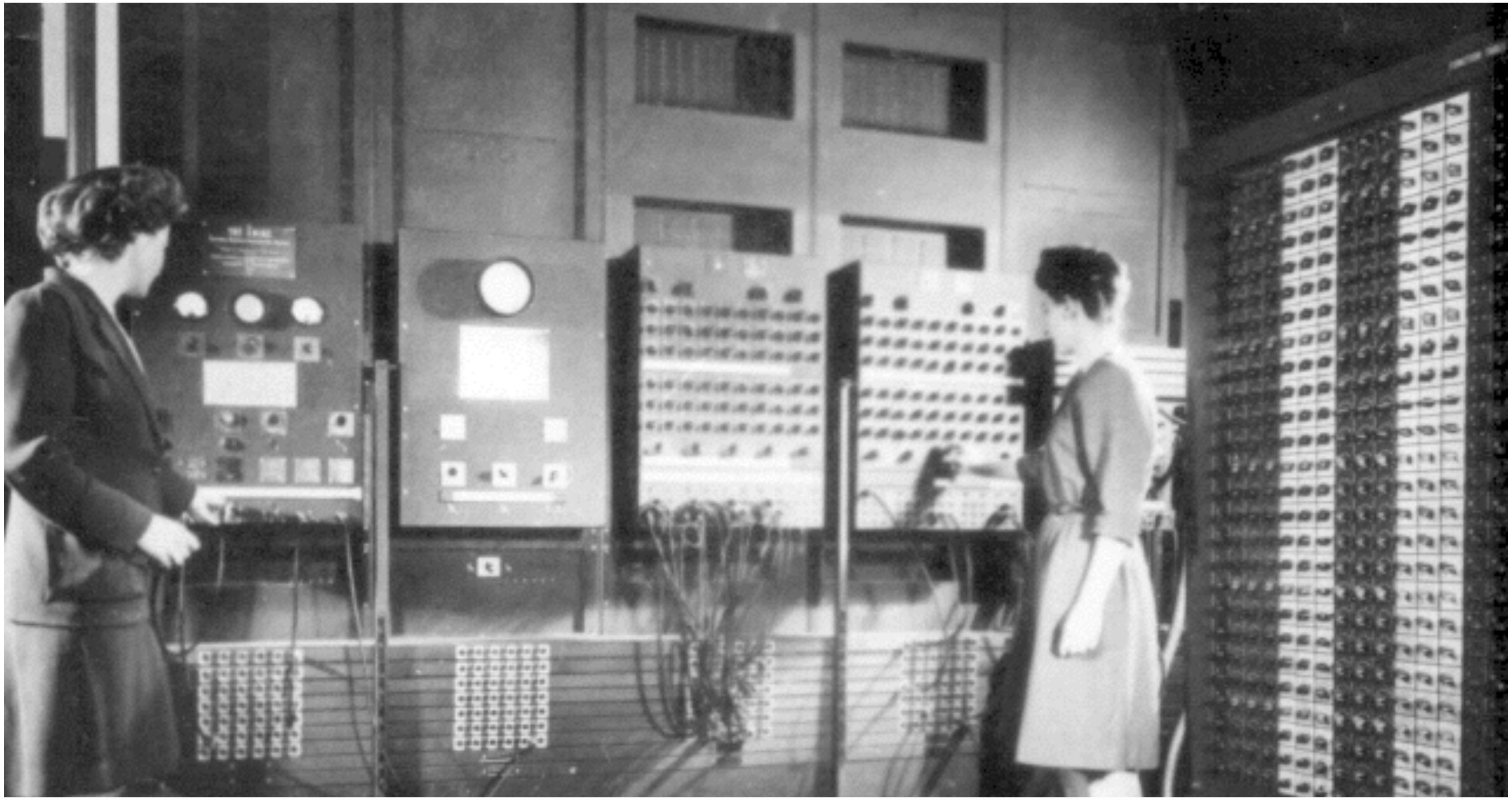
Some history on this viewpoint

Research on the topic

Open questions

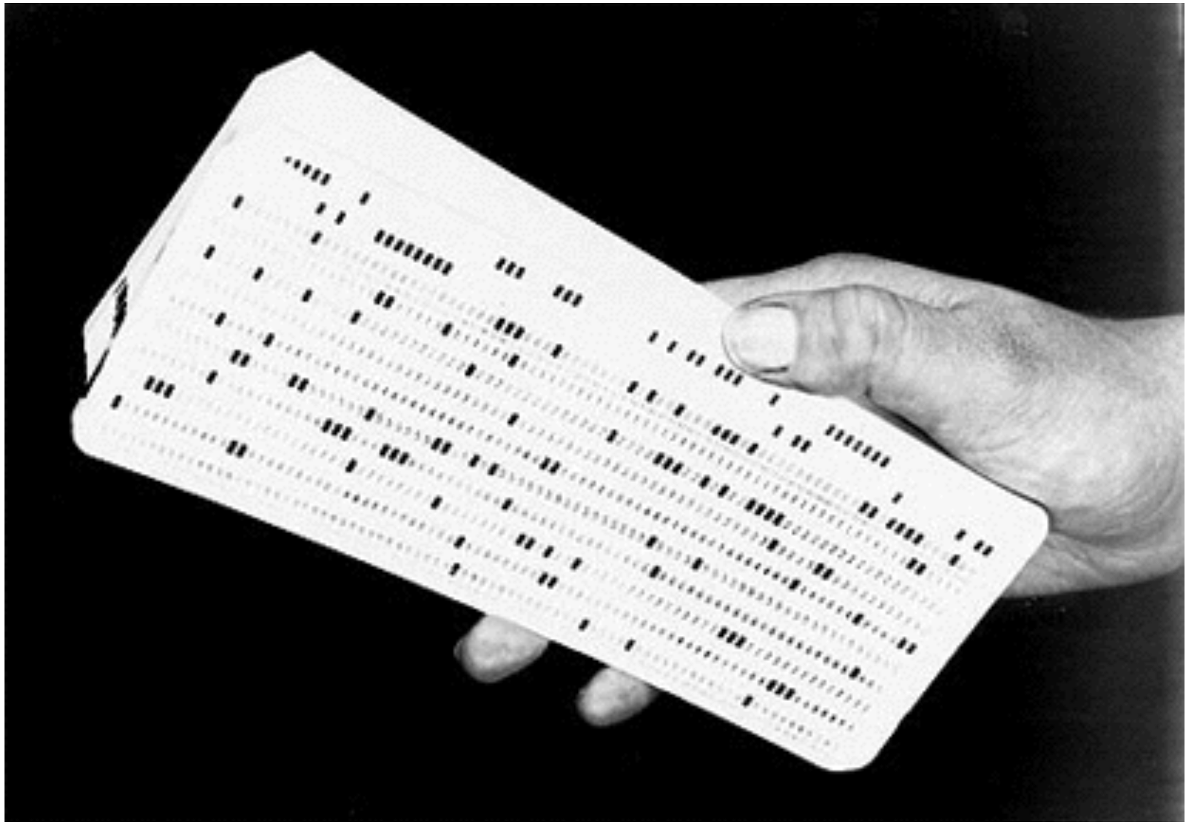
IN THE BEGINNING

(the early 1940's anyway)



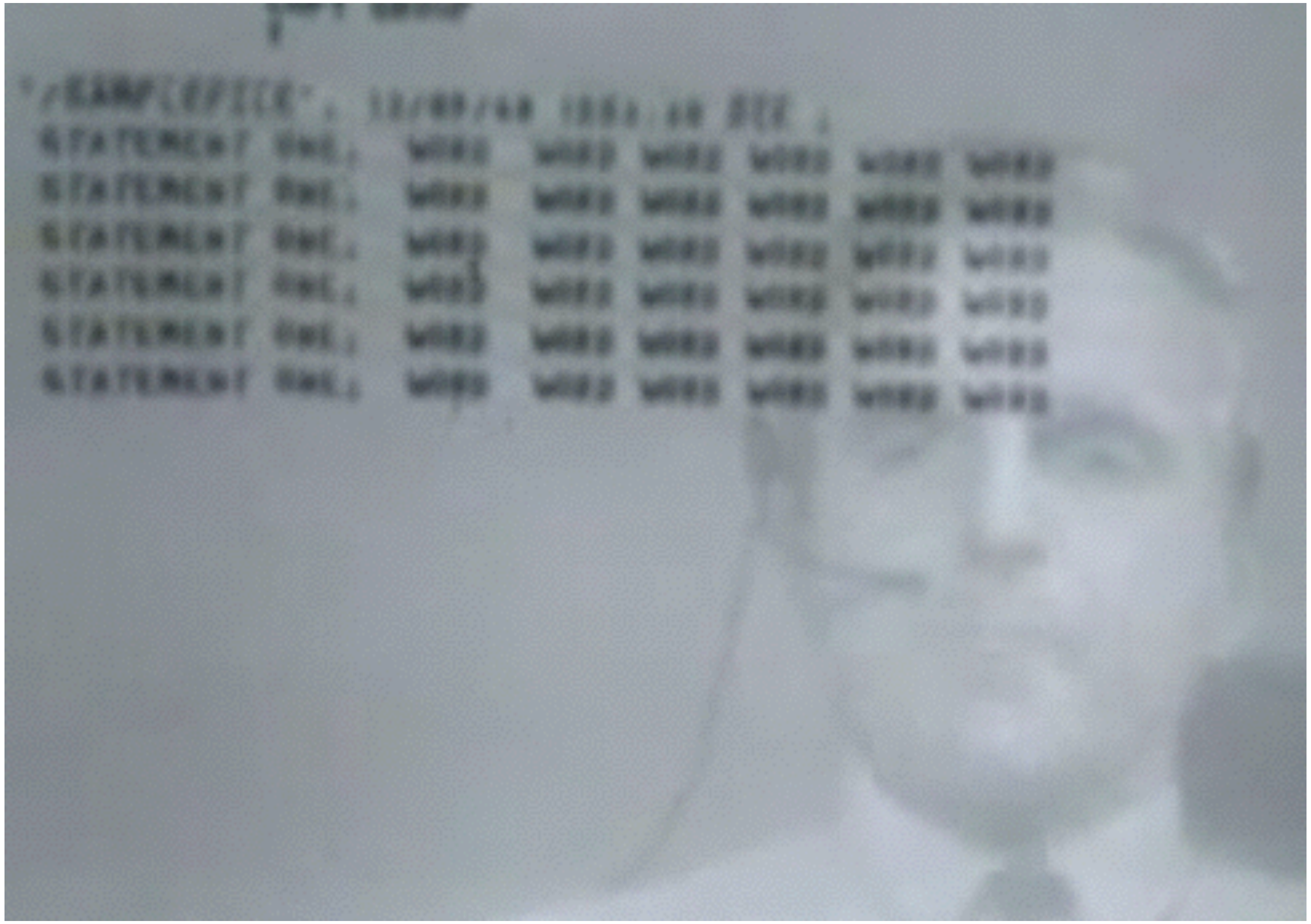
Programmers Betty Jean Jennings (left) and Fran Bilas (right) operate the ENIAC's main control panel at the Moore School of Electrical Engineering

SEPARATING HARDWARE AND SOFTWARE



the IBM punchcard

INTERACTIVE COMPUTING



```
'/SAMPLEFILE', 12/09/68 [illegible] ;
 STATEMENT ONE,  WORD  WORD WORD WORD WORD WORD
 STATEMENT ONE,  WORD  WORD WORD WORD WORD WORD
 STATEMENT ONE,  WORD  WORD WORD WORD WORD WORD
 STATEMENT ONE,  WORD  WORD WORD WORD WORD WORD
 STATEMENT ONE,  WORD  WORD WORD WORD WORD WORD
 STATEMENT ONE,  WORD  WORD WORD WORD WORD WORD
```

Douglas Engelbart, 1968

INTERACTIVE COMPUTING

what made this different was the **speed** with which the computer reacted to human input

- no longer necessary to write and wait
- feedback loops between people and computers were reduced to milliseconds
- the result of ones commands could be seen **immediately**, allowing people to engage in the rapid exchange of information

BATCH COMPUTING

programming

INTERACTIVE COMPUTING

GUIs

web sites

mobile apps

Kinect

....

BATCH COMPUTING

manipulate a computer’s **future** behavior through abstract notation

INTERACTIVE COMPUTING

manipulate the computer’s **present** behavior through concrete notations

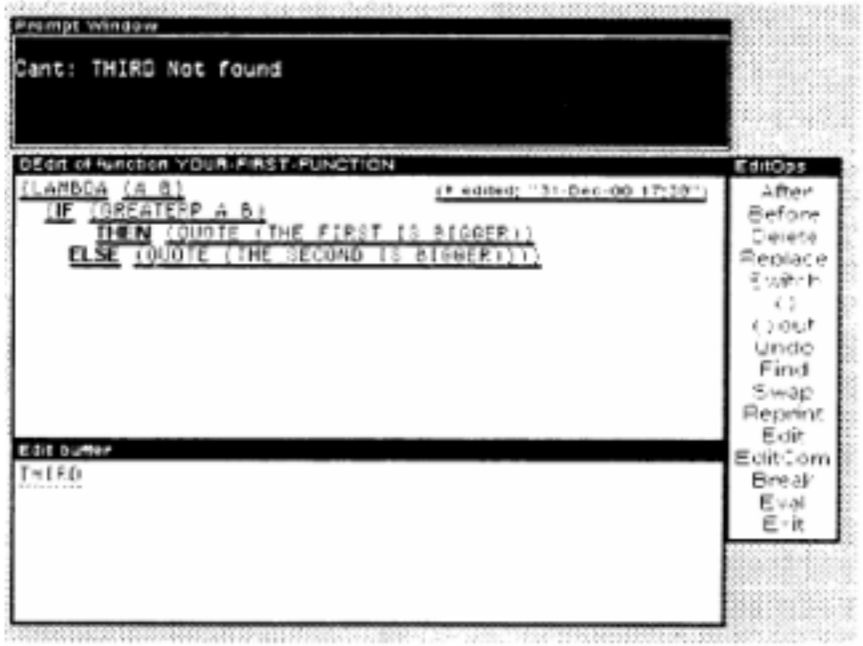
Blackwell, A.F. (2002). First steps in programming: A rationale for Attention Investment models. In Proceedings of the IEEE Symposia on Human-Centric Computing Languages and Environments, pp. 2-10.

BATCH COMPUTING

researchers started to ask...

> “why can’t code be interactive like every other kind of document?”

INTERACTIVE CODE 1980



Prompt Window

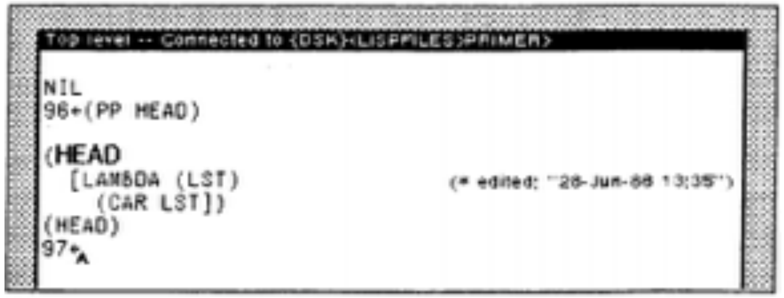
```
Cant: THIRD Not found
```

DEdit of function YOUR-FIRST-FUNCTION

```
(LAMBDA (A B)                          (* edited: "31-Dec-00 17:20")
  (IF (GREATERP A B)
      THEN (QUOTE (THE FIRST IS BIGGER))
    ELSE (QUOTE (THE SECOND IS BIGGER))))
```

Edit buffer

```
THIRD
```

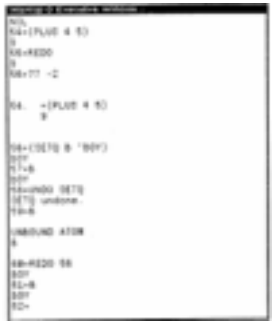


Top level -- Connected to {DSK}<LISPFILES>PRIMER>

```
NIL
96←(PP HEAD)

(HEAD
  [LAMBDA (LST)                          (* edited: "28-Jun-88 13:35")
    (CAR LST)])
(HEAD)
97←
```

FACTORIAL Frame

```
UNBOUND ATOM

(DUMMY broken)
```



```
UNBOUND ATOM
```

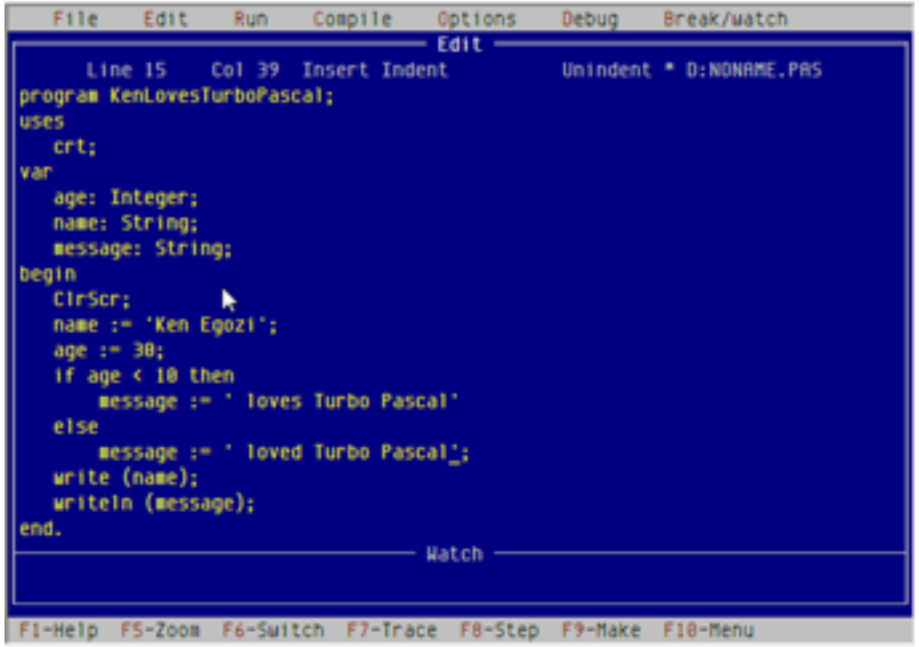
http://bitsavers.informatik.uni-stuttgart.de/pdf/xerox/interlisp/3102300_interlDprimer_Nov86.pdf

InterLisp: syntax highlighting, spell checking, auto-complete, version control, integrated debugger, etc.

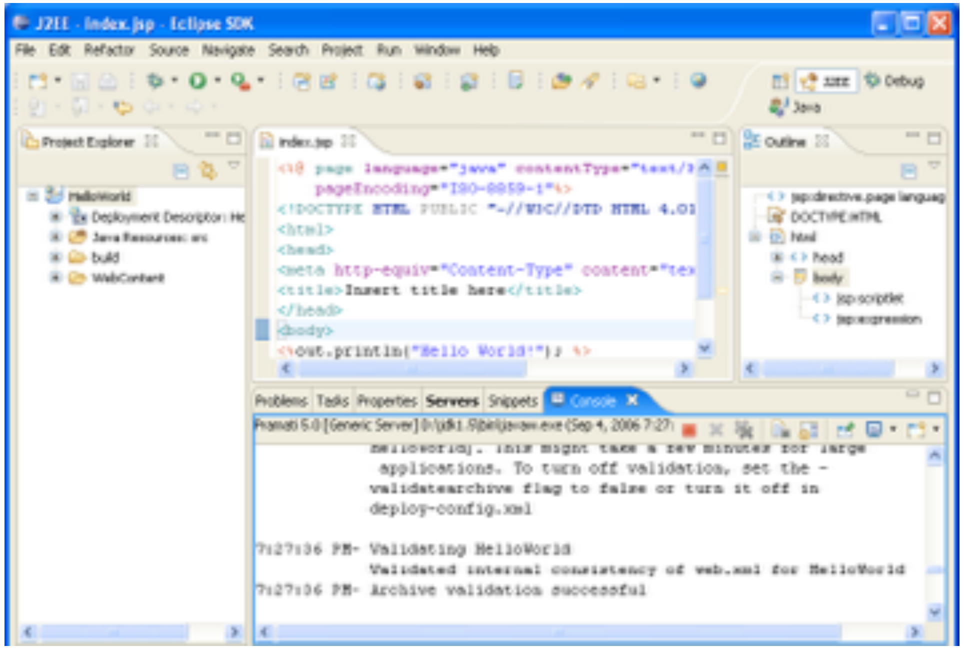
a vision for writing, executing, and understanding code interactively

INTERACTIVE CODE 1980–2000

these ideas go mainstream



```
      Line 15     Col 39  Insert Indent          Unindent * D:NONAME.PAS
program KenLovesTurboPascal;
uses
   crt;
var
   age: Integer;
   name: String;
   message: String;
begin
   ClrScr;
   name := 'Ken Egozi';
   age := 30;
   if age < 10 then
       message := ' loves Turbo Pascal'
   else
       message := ' loved Turbo Pascal';
   write (name);
   writeln (message);
end.
```



```
<%@ page language="java" contentType="text/h
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01
<html>
<head>
<meta http-equiv="Content-Type" content="tex
<title>Insert title here</title>
</head>
<body>
<%out.println("Hello World!"); %>
```

```
Pramati 5.0 [Generic Server] D:\jdk1.5\bin\javaw.exe (Sep 4, 2006 7:27)
            HelloWorld]. This might take a few minutes for large
             applications. To turn off validation, set the -
            validatearchive flag to false or turn it off in
            deploy-config.xml

7:27:36 PM- Validating HelloWorld
            Validated internal consistency of web.xml for HelloWorld
7:27:36 PM- Archive validation successful
```

Turbo Pascal 1983

Eclipse 2004

THE PRESENT AND FUTURE

What’s hard about making programming environments more usable?

What progress have we made?

SIX BARRIERS IN PROGRAMMING

Ko, A.J., Myers, B.A., and Aung, H. (2004). Six Learning Barriers in End-User Programming Systems (2004). IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC), 199-206.

Think of programming platforms as a collection of **programming interfaces**:

Language constructs, functions, classes, libraries, APIs, types, etc.

I claim that all barriers in programming arise from:

Problem solving challenges inherent to devising algorithms and data structures to solve a problem (which I called DESIGN barriers)

Usability problems with with the programming interfaces necessary to express these solutions

SIX BARRIERS IN PROGRAMMING

Discuss with your neighbor:

What was useful about the paper?

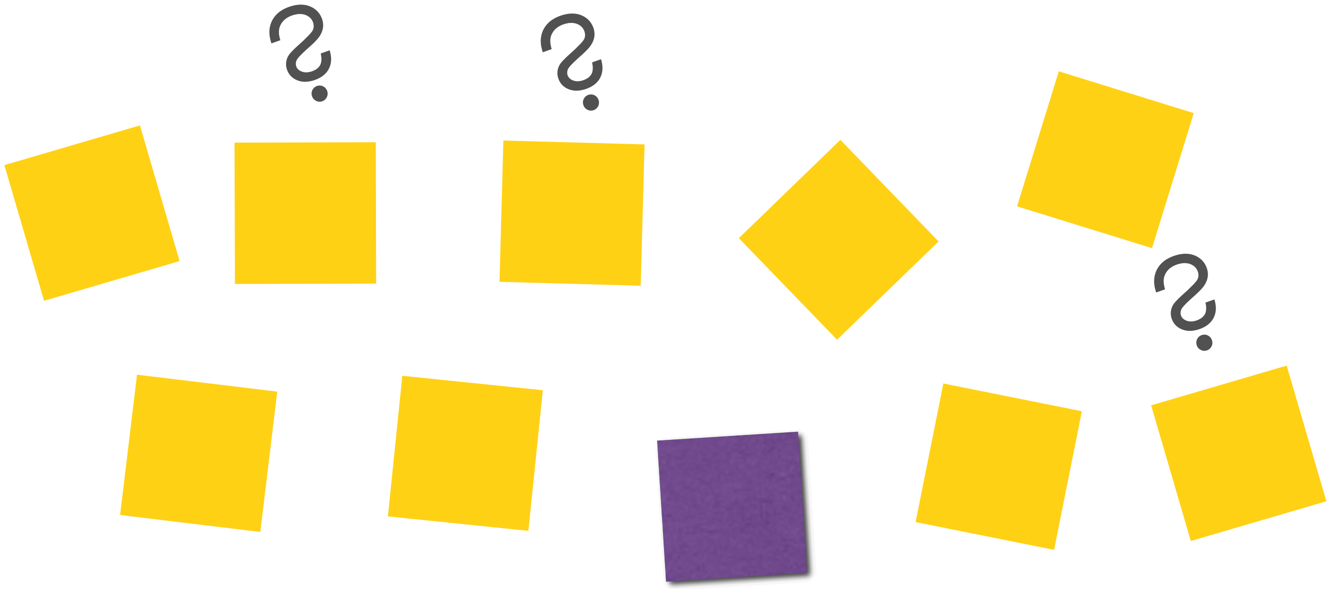
What was surprising?

What was less useful?

SELECTION barriers

Finding programming interfaces that implement a particular behavior

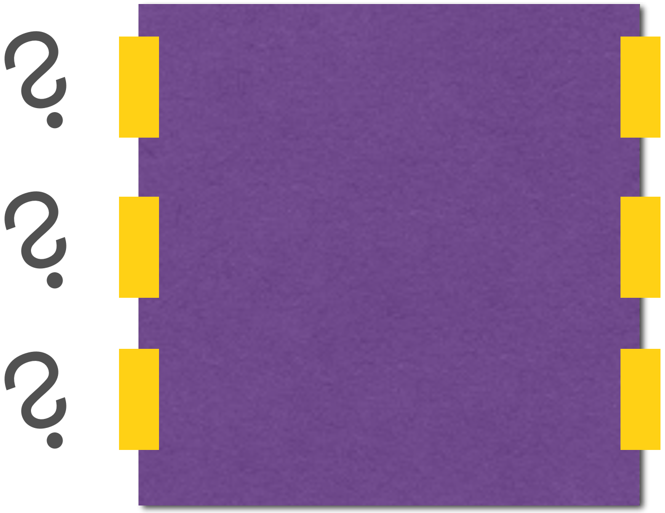
Reading API documentation, asking a friend, using a code search engine, searching Stack Overflow



USE barriers

Discovering the intended way to use a programming interface (syntax, inputs, outputs, side effects, preconditions, postconditions, etc.)

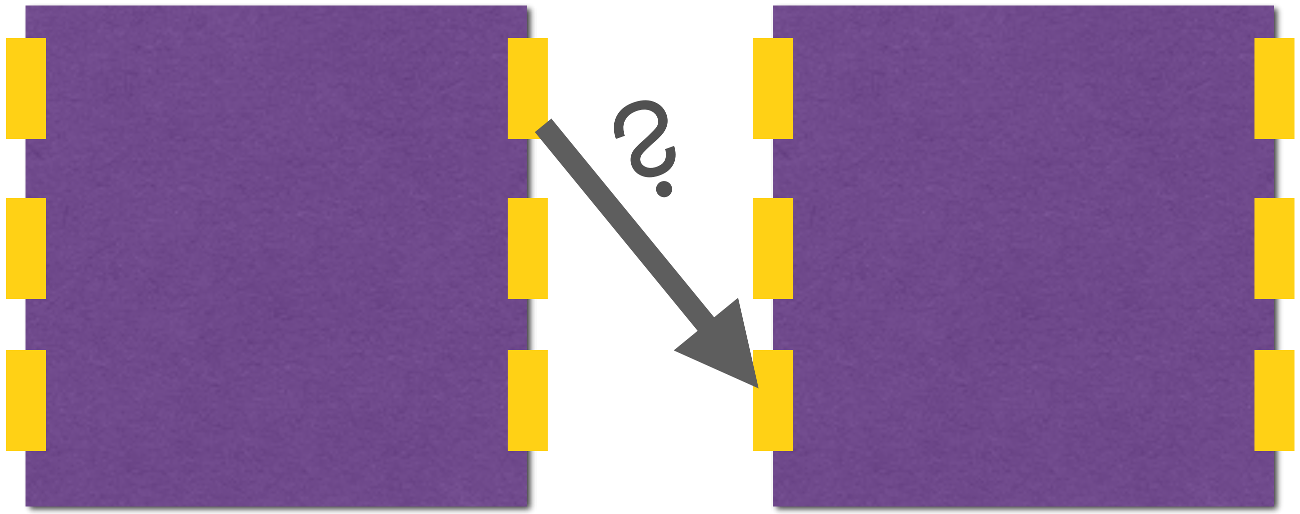
- Reading documentation about a function, class, or method, writing test cases



COORDINATION barriers

Discovering usage rules that govern how programming interfaces can be composed

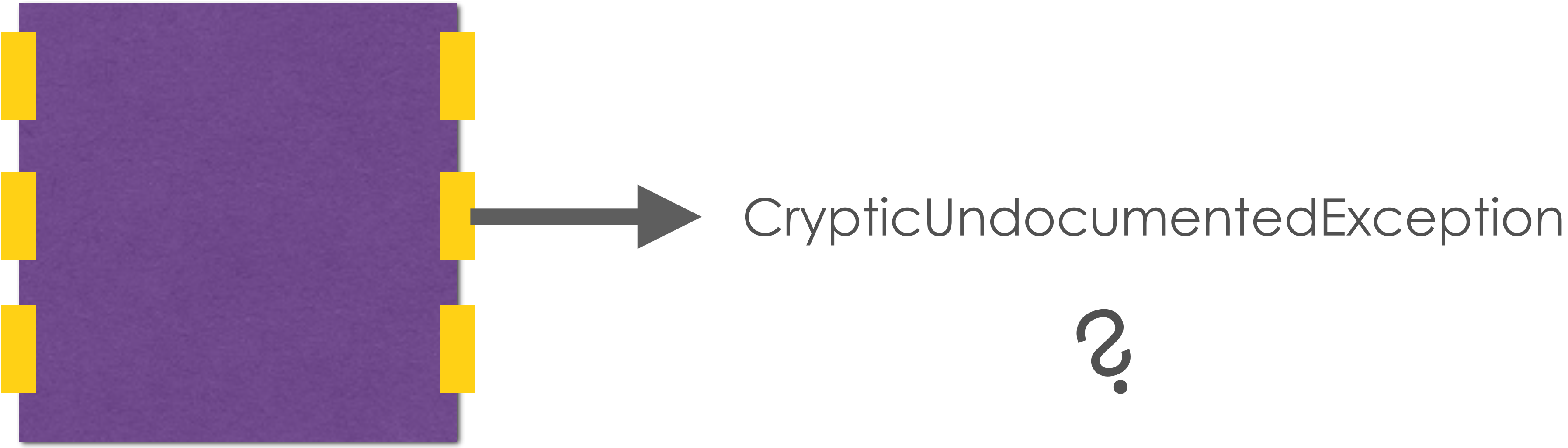
Reading Stack Overflow, searching for error messages on Google, reading documentation



UNDERSTANDING barriers

Difficulties interpreting the unexpected behavior of a programming interface

Searching Google for an error message, test case minimization, guessing



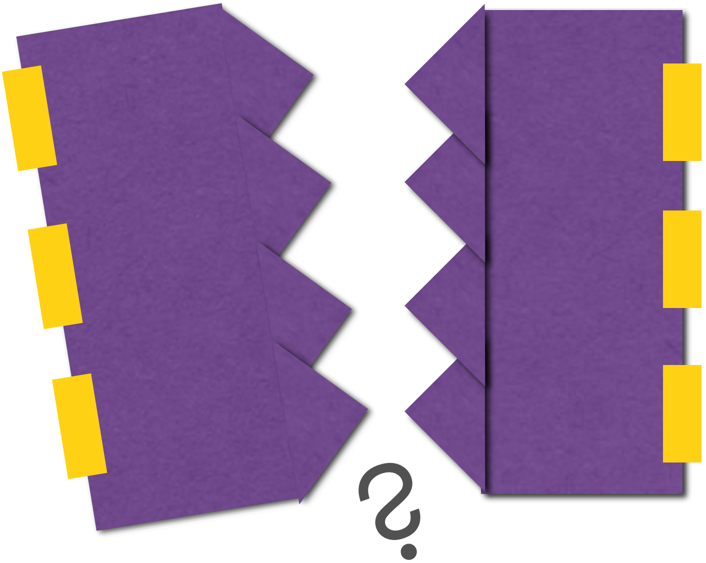
CrypticUndocumentedException

?

INFORMATION barriers

Difficulties observing the internal behavior of a programming interface

- Finding a better debugging tool, writing the perfect print statement, selecting the perfect breakpoint



PROGRESS

addressing these barriers

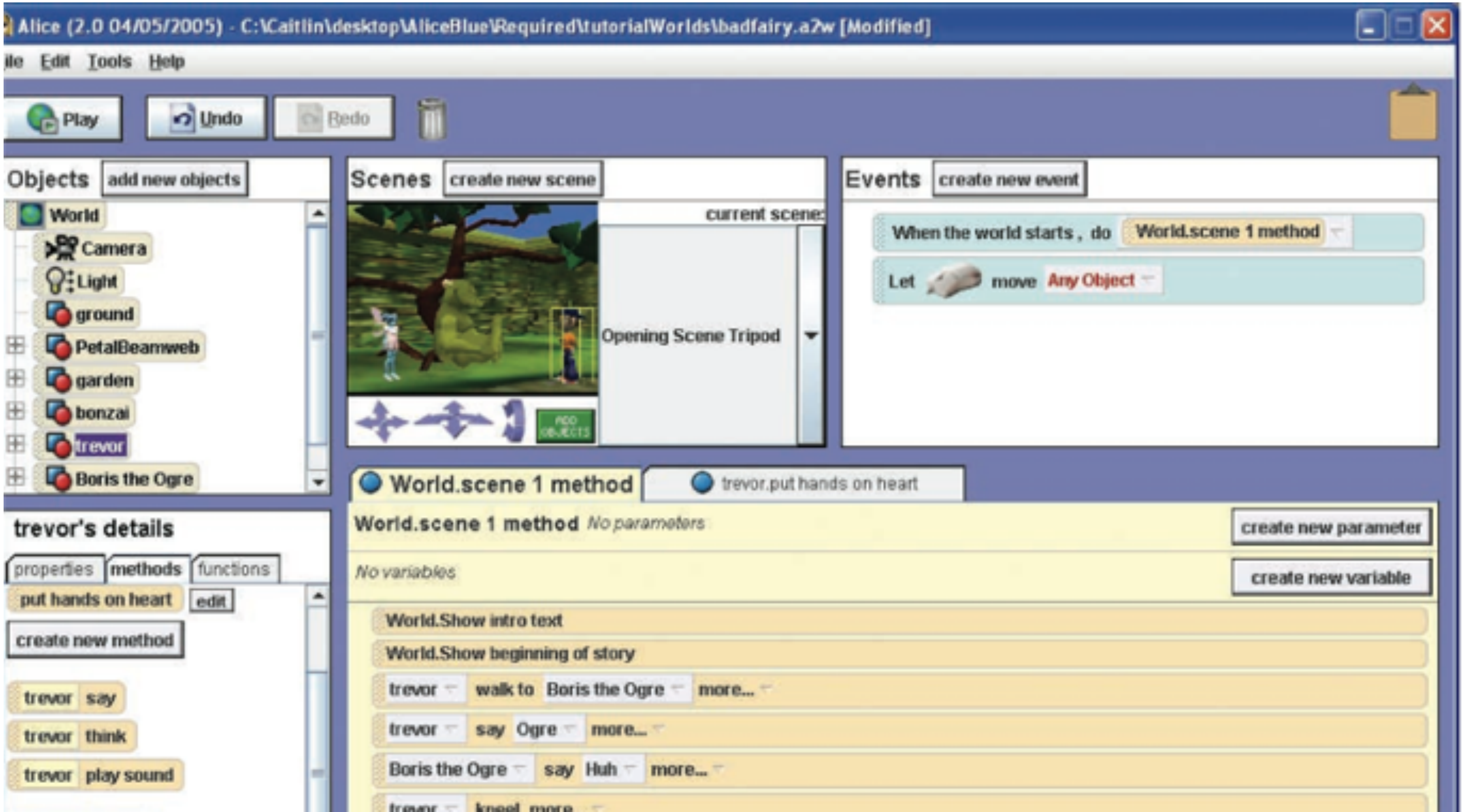
solutions to USE barriers

Alice (2007)

Kelleher, C. and R. Pausch. Using Storytelling to Motivate Programming. Communications of the ACM, vol. 50, no. 7, July 2007, pages 58-64.

what if syntax and type errors were impossible

(removes USE barriers)



When the world starts , do World.scene 1 method

Let move Any Object

World.scene 1 method *No parameters*

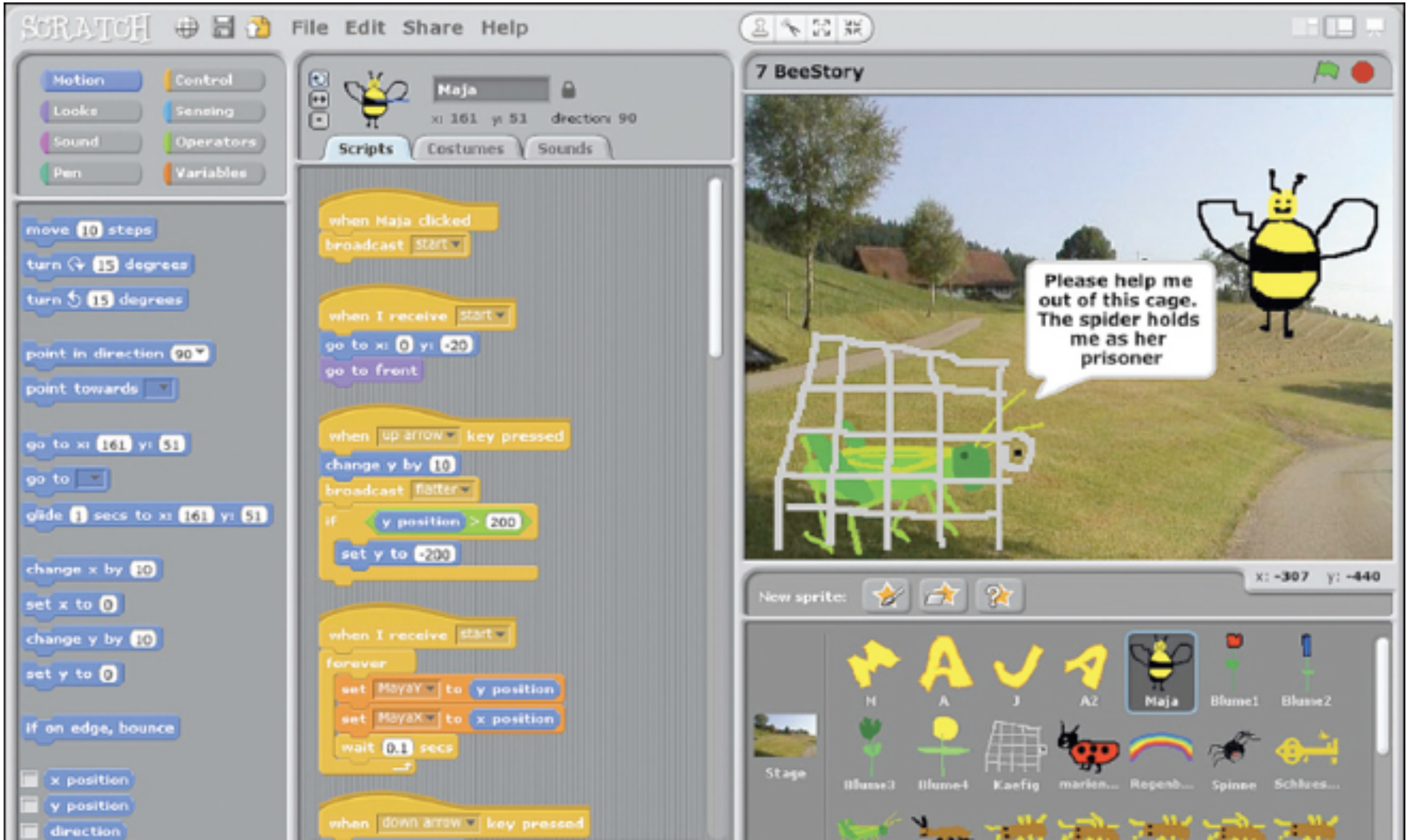
No variables

```
World.Show intro text
World.Show beginning of story
trevor walk to Boris the Ogre more...
trevor say Ogre more...
Boris the Ogre say Huh more...
trevor kneel more
```

Scratch (2008)

Mitchel Resnick, John Maloney, Andrés Monroy-Hernández, Natalie Rusk, Evelyn Eastmond, Karen Brennan, Amon Millner, Eric Rosenbaum, Jay Silver, Brian Silverman, Yasmin Kafai. Scratch: Programming for All. Communications of he ACM Vol. 52 No. 11, Pages 60-67

same idea as Alice: drag and drop prevents syntax and type errors *(removes USE barriers)*

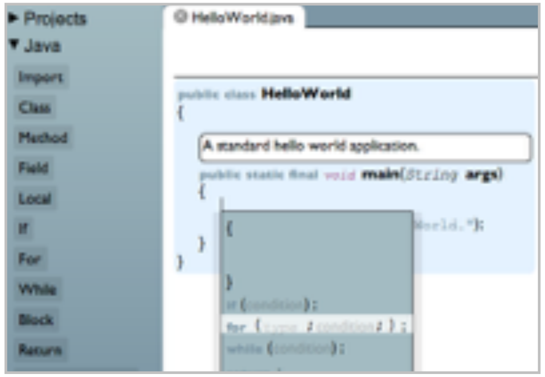


```
when Maja clicked
broadcast start

when I receive start
go to x: 0 y: -20
go to front

when up arrow key pressed
change y by 10
broadcast flatter
if y position > 200
    set y to -200

when I receive start
forever
    set MayaY to y position
    set MayaX to x position
    wait 0.1 secs

when down arrow key pressed
```

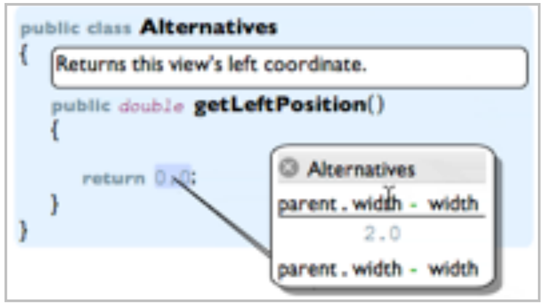
Barista (2006)

Ko, A. J. and Myers, B. A. (2006).Barista: An Implementation Framework for Enabling New Tools, Interaction Techniques and Views for Code Editors (2006). ACM Conference on Human Factors in Computing Systems (CHI), Montreal, Canada, April 24Ð27,

what if you could embed anything in a source file, in context? *(removes USE barriers)*



```
public class HelloWorld
{
    A standard hello world application.
    public static final void main(String args)
    {
        {
        }
        if (condition);
        for (type ; condition ; );
        while (condition);
            World.");
    }
}
```

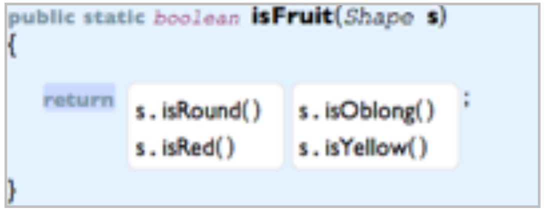


```
public class Alternatives
{
    Returns this view's left coordinate.
    public double getLeftPosition()
    {
        return 0.0;
    }
}
```

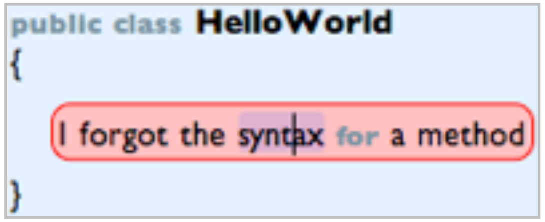
parent . width - width

2.0

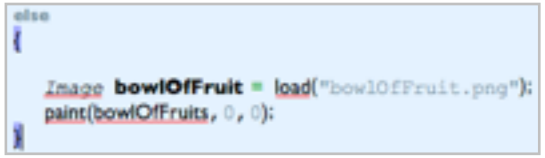
parent . width - width



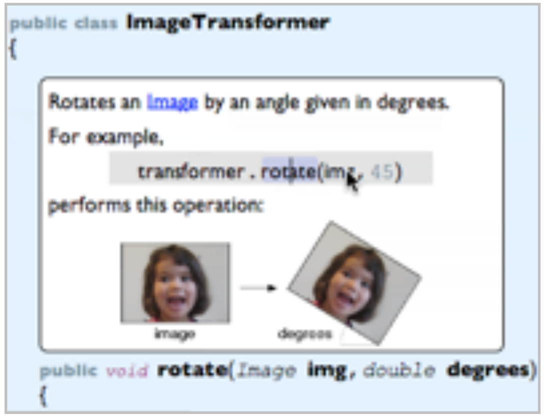
```
public static boolean isFruit(Shape s)
{
    return  s . isRound()   s . isOblong()  ;
            s . isRed()     s . isYellow()
}
```



```
public class HelloWorld
{
    I forgot the syntax for a method
}
```

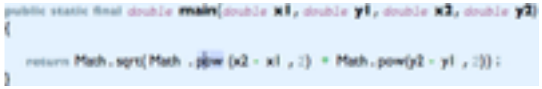


```
else
{
    Image bowlOfFruit = load("bowlOfFruit.png");
    paint(bowlOfFruits, 0, 0);
}
```

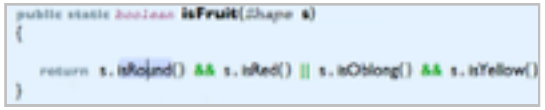


```
public class ImageTransformer
{
    Rotates an Image by an angle given in degrees.
    For example,
        transformer . rotate(img, 45)
    performs this operation:
```

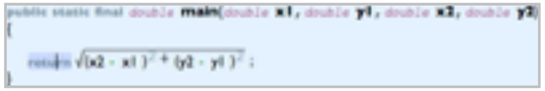
```
    public void rotate(Image img, double degrees)
    {
```



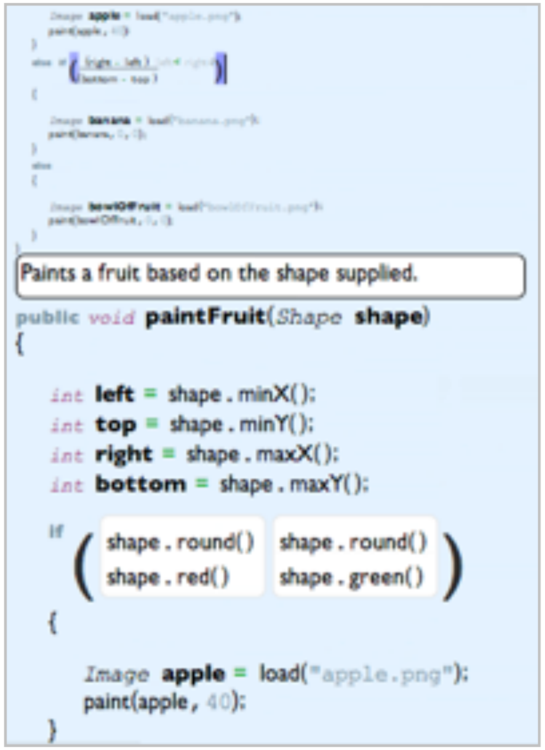
```
public static final double main(double x1, double y1, double x2, double y2)
{
    return Math.sqrt(Math.pow(x2 - x1, 2) + Math.pow(y2 - y1, 2));
}
```



```
public static boolean isFruit(Shape s)
{
    return s.isRound() && s.isRed() || s.isOblong() && s.isYellow()
}
```



```
public static final double main(double x1, double y1, double x2, double y2)
{
    return √(x2 - x1)² + (y2 - y1)² ;
}
```



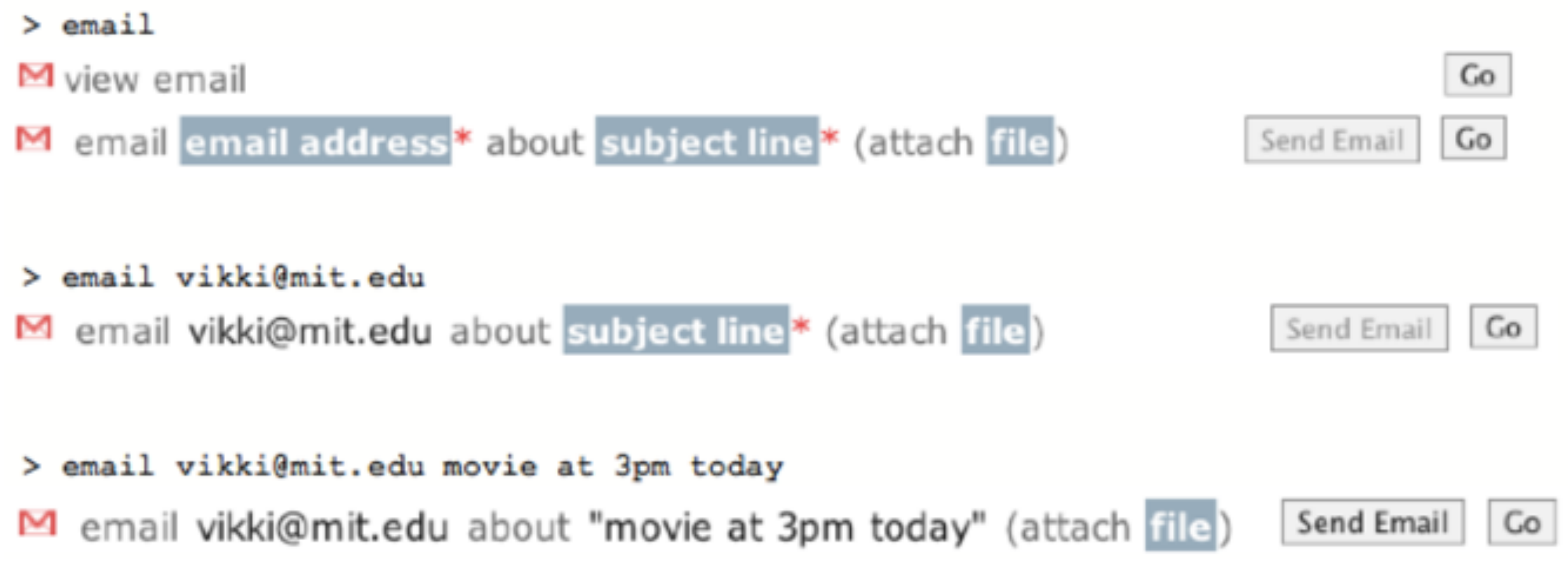
```
    Image apple = load("apple.png");
    paint(apple, 40);
}
else if (right - left ... bottom - top)
{
    Image banana = load("banana.png");
    paint(banana, 0, 0);
}
else
{
    Image bowlOfFruit = load("bowlOfFruit.png");
    paint(bowlOfFruit, 0, 0);
}
Paints a fruit based on the shape supplied.
public void paintFruit(Shape shape)
{
    int left = shape . minX();
    int top = shape . minY();
    int right = shape . maxX();
    int bottom = shape . maxY();

    if ( shape . round()   shape . round() )
         shape . red()     shape . green()
    {
        Image apple = load("apple.png");
        paint(apple, 40);
    }
```

solutions to
SELECTION barriers

keyword programming (2006)

Greg Little and Robert C. Miller. "Translating Keyword Commands into Executable Code." UIST 2006, pp. 135-144.

what if programs could be guessed from natural language? *(removes SELECTION barriers)*



```
> email
```

view email

email email address* about subject line* (attach file)

```
> email vikki@mit.edu
```

email vikki@mit.edu about subject line* (attach file)

```
> email vikki@mit.edu movie at 3pm today
```

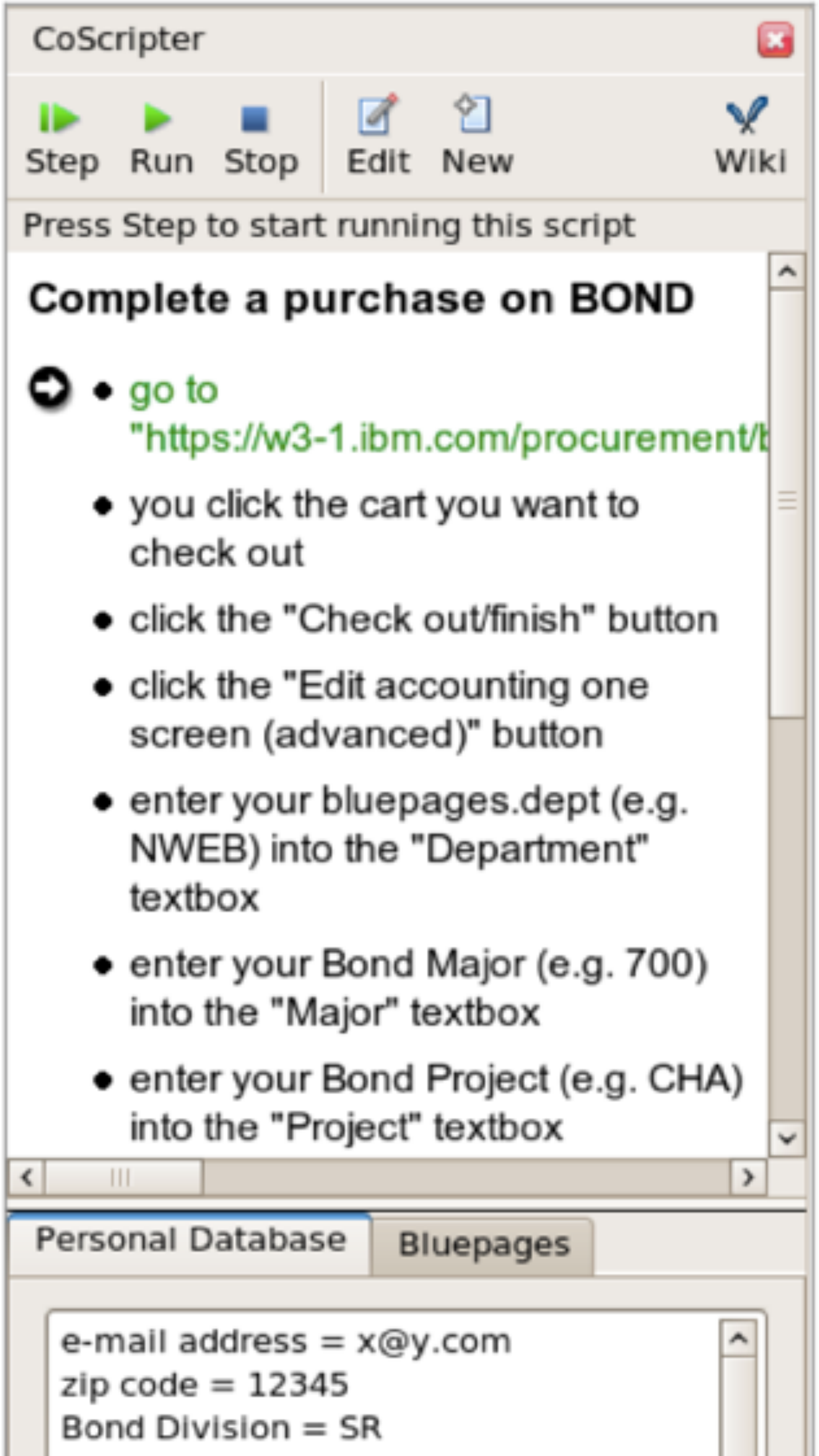
email vikki@mit.edu about "movie at 3pm today" (attach file)

discussion paper!

CoScripter (2008)

what if web interactions could be recorded and replayed? *(removes SELECTION barriers)*

Gilly Leshed, Eben M. Haber, Tara Matthews, and Tessa Lau. 2008. CoScripter: automating \& sharing how-to knowledge in the enterprise. In Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems (CHI '08). ACM, New York, NY, USA, 1719-1728.



Press Step to start running this script

Complete a purchase on BOND

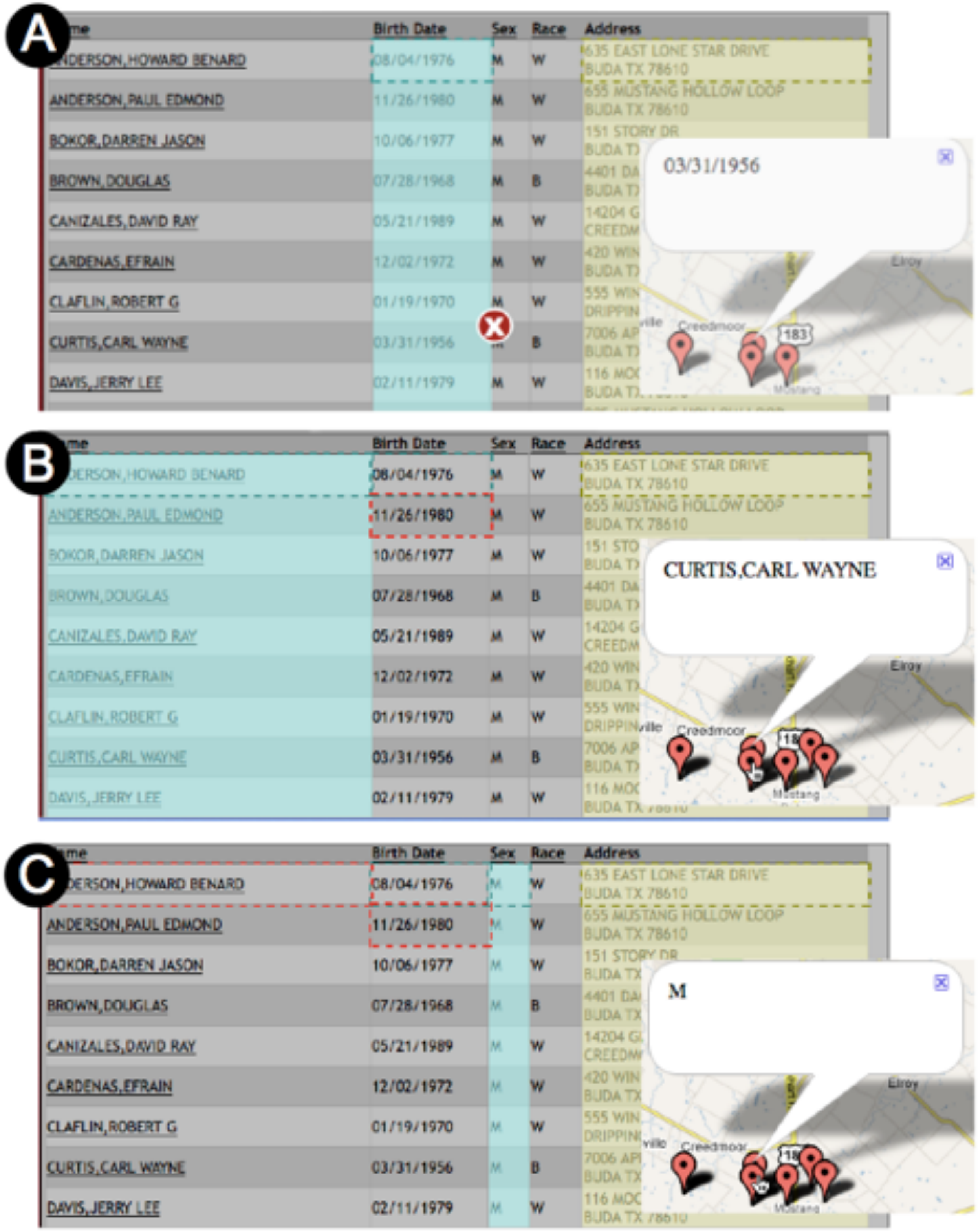
- go to "https://w3-1.ibm.com/procurement/b
- you click the cart you want to check out
- click the "Check out/finish" button
- click the "Edit accounting one screen (advanced)" button
- enter your bluepages.dept (e.g. NWEB) into the "Department" textbox
- enter your Bond Major (e.g. 700) into the "Major" textbox
- enter your Bond Project (e.g. CHA) into the "Project" textbox

e-mail address = x@y.com
zip code = 12345
Bond Division = SR

Reform (2009)

web mashups
through interactive
web scraping
(removes SELECTION barriers)

Michael Toomim, Steven M. Drucker, Mira Dontcheva, Ali Rahimi, Blake Thomson, and James A. Landay. 2009. Attaching UI enhancements to websites with end users. In Proceedings of the 27th international conference on Human factors in computing systems (CHI '09). ACM, New York, NY, USA, 1859-1868. DOI=10.1145/1518701.1518987 http://doi.acm.org/10.1145/1518701.1518987

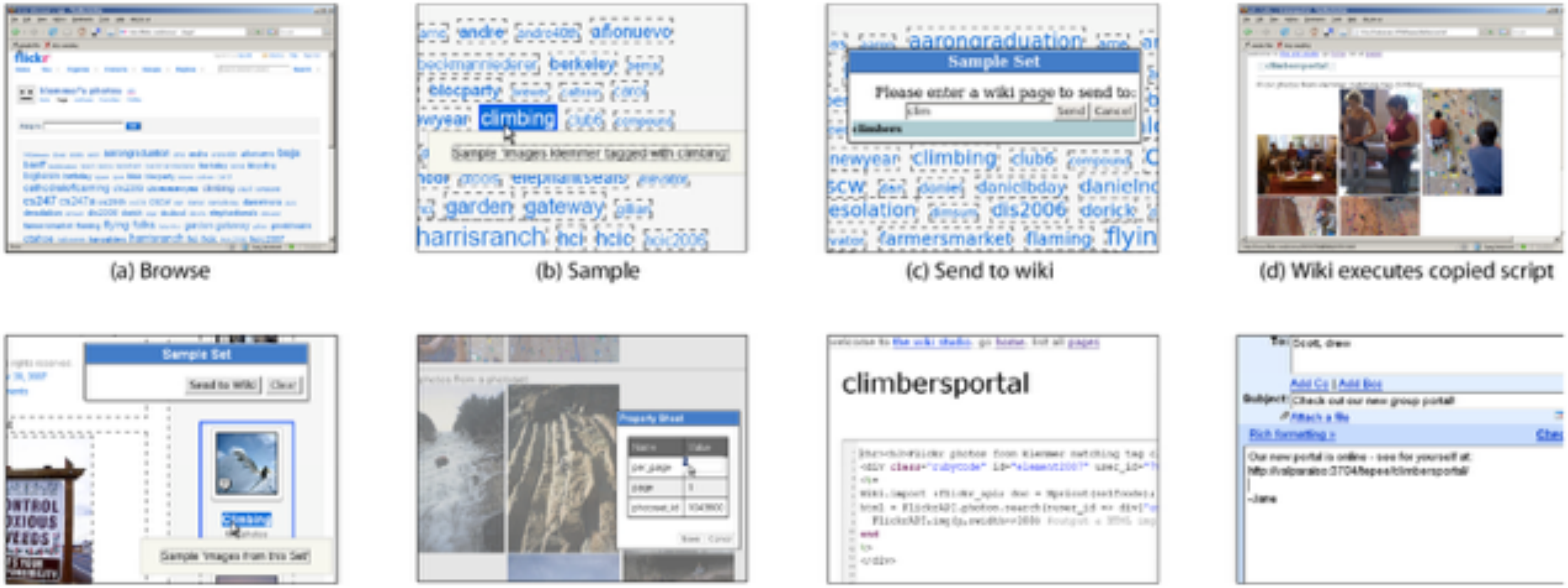


me	Birth Date	Sex	Race	Address
ANDERSON,HOWARD BENARD	08/04/1976	M	W	635 EAST LONE STAR DRIVE BUDA TX 78610
ANDERSON,PAUL EDMOND	11/26/1980	M	W	655 MUSTANG HOLLOW LOOP BUDA TX 78610
BOKOR,DARREN JASON	10/06/1977	M	W	151 STORY DR BUDA TX
BROWN,DOUGLAS	07/28/1968	M	B	4401 DA BUDA TX
CANIZALES,DAVID RAY	05/21/1989	M	W	14204 G CREEDM
CARDENAS,EFRAIN	12/02/1972	M	W	420 WIN BUDA TX
CLAFLIN,ROBERT G	01/19/1970	M	W	555 WIN DRIPPIN
CURTIS,CARL WAYNE	03/31/1956	M	B	7006 AP BUDA TX
DAVIS,JERRY LEE	02/11/1979	M	W	116 MOC BUDA TX

d.mix (2007)

Hartmann, Björn, Leslie Wu, Kevin Collins and Scott R. Klemmer. Programming by a Sample: Rapidly Creating Web Applications with d.mix. In Proceedings of uist 2007: ACM Symposium on User Interface Software and Technology. Newport, Rhode Island, USA, 2007.

what if web service mashups could be constructed by selecting examples? *(removes SELECTION barriers)*


Sample Set
Please enter a wiki page to send to:
Send
Cancel
Send to Wiki
Clear
Sample 'Images from this Set'
Sample 'Images Klemmer tagged with climbing'
climbing
climbersportal
Our new portal is online - see for yourself at:
Check out our new group portal

(a) Browse

(b) Sample

(c) Send to wiki

(d) Wiki executes copied script

(e) Browse & sample again

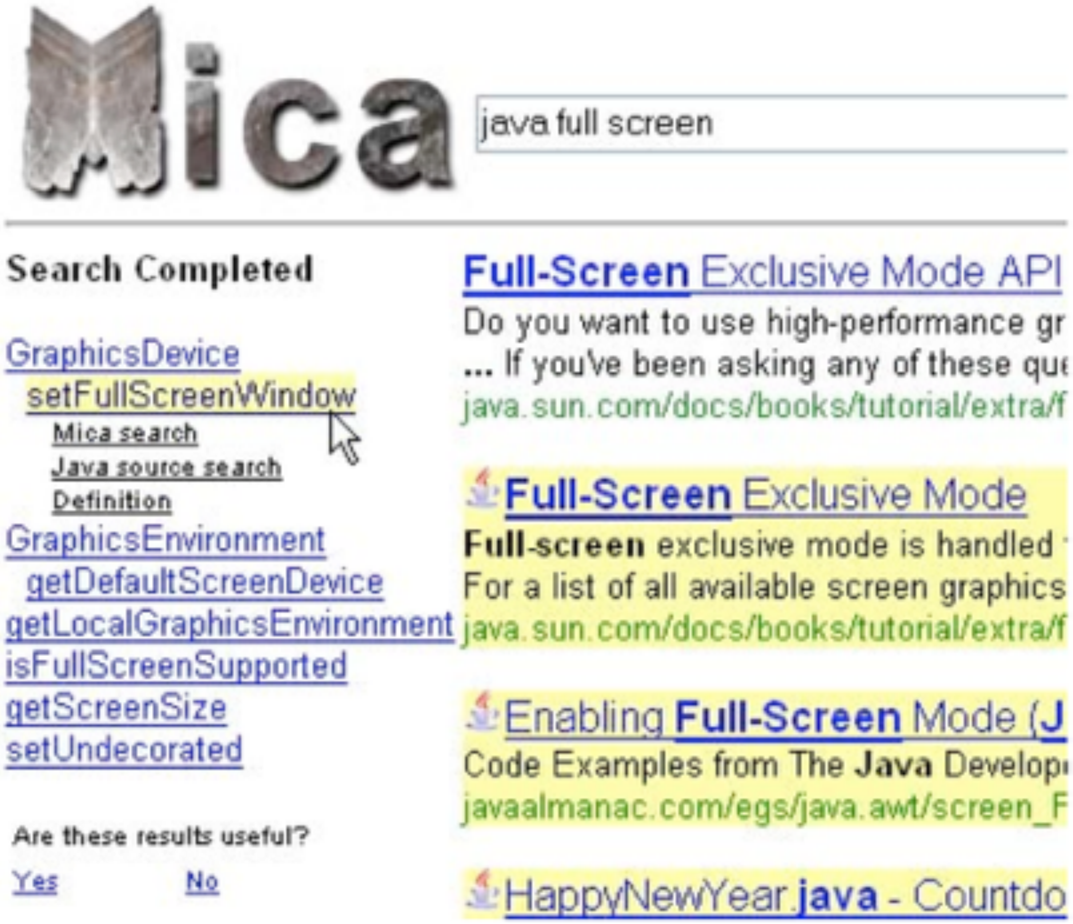
(f) Edit properties in wiki

(g) Edit source code in wiki

(h) Share URL

Mica (2006)

Mines an API to augment Google search results with classes and methods



Mica

java full screen

Search Completed

GraphicsDevice
- setFullScreenWindow
  - Mica search
  - Java source search
  - Definition

GraphicsEnvironment
- getDefaultScreenDevice

getLocalGraphicsEnvironment
isFullScreenSupported
getScreenSize
setUndecorated

Are these results useful?

Yes No

Full-Screen Exclusive Mode API
Do you want to use high-performance gr
... If you've been asking any of these que
java.sun.com/docs/books/tutorial/extra/f

Full-Screen Exclusive Mode
Full-screen exclusive mode is handled
For a list of all available screen graphics
java.sun.com/docs/books/tutorial/extra/f

Enabling **Full-Screen** Mode (**J**
Code Examples from The **Java** Develop
javaalmanac.com/egs/java.awt/screen_F

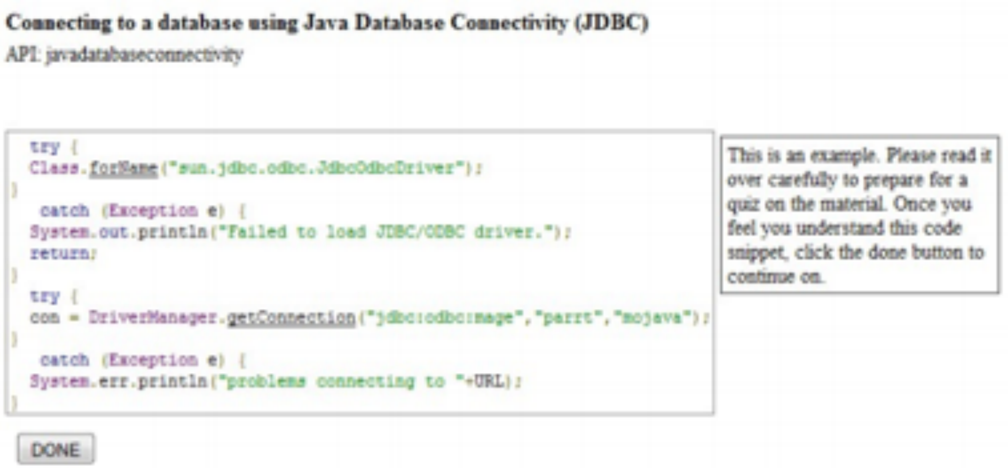
HappyNewYear.**java** - Countdo

solutions to COORDINATION barriers

Intelligent API tutors

Krishnamoorthy, V., Appasamy, B., and Scaffidi, C. (2013). Using intelligent tutors to teach students how APIs are used for software engineering in practice. IEEE Transactions on Education, 56, 3, 355-363.

Generates instructional tasks from online FAQs and open source code providing more explanation and context about API usage rules



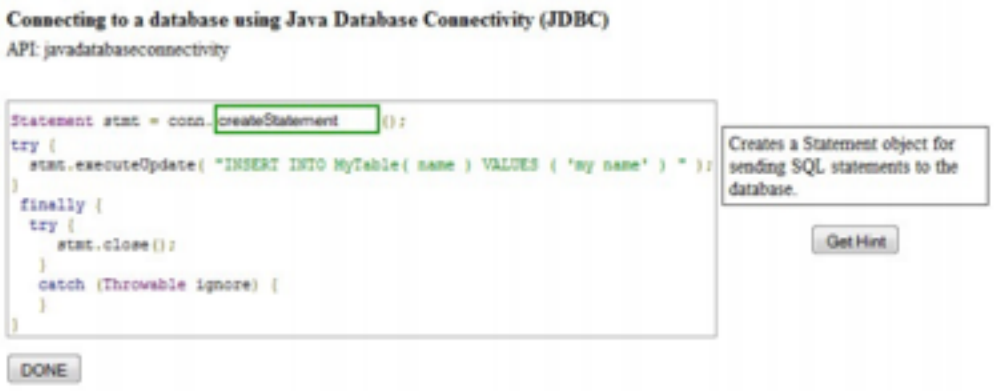
Connecting to a database using Java Database Connectivity (JDBC)

API: javadatabaseconnectivity

```
  try {
  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
}
   catch (Exception e) {
  System.out.println("Failed to load JDBC/ODBC driver.");
  return;
}
  try {
  con = DriverManager.getConnection("jdbc:odbc:mage","parrt","mojava");
}
   catch (Exception e) {
  System.err.println("problems connecting to "+URL);
}
```

This is an example. Please read it over carefully to prepare for a quiz on the material. Once you feel you understand this code snippet, click the done button to continue on.

Fig. 3. A quiz typically starts with a few examples showing how to use the API.



Connecting to a database using Java Database Connectivity (JDBC)

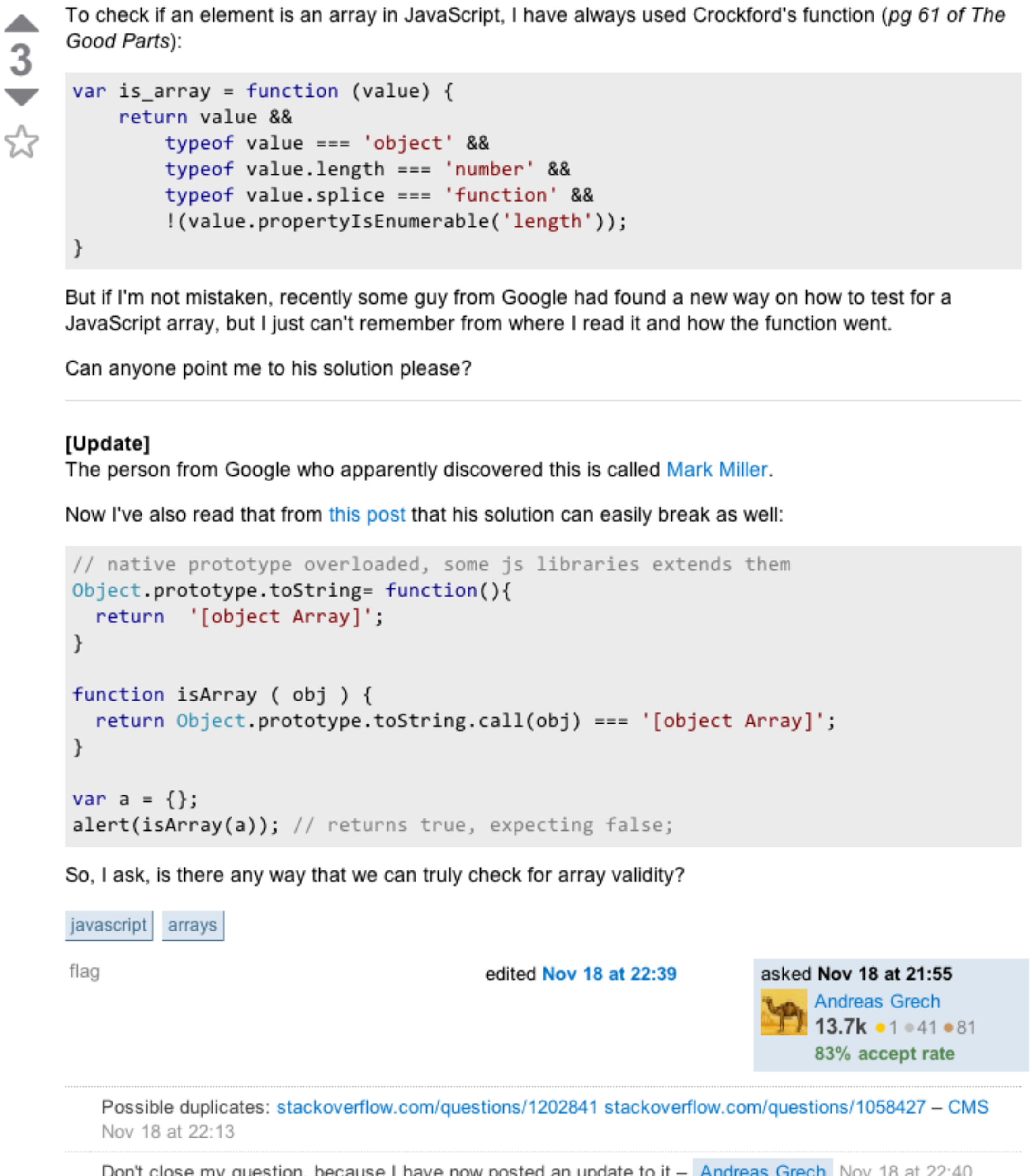
API: javadatabaseconnectivity

```
Statement stmt = conn.createStatement();
try {
  stmt.executeUpdate( "INSERT INTO MyTable( name ) VALUES ( 'my name' ) " );
}
 finally {
  try {
     stmt.close();
   }
   catch (Throwable ignore) {
   }
}
```

Creates a Statement object for sending SQL statements to the database.

Stack Overflow

A searchable repository of patterns and usage rules for composing programming interfaces



To check if an element is an array in JavaScript, I have always used Crockford's function (*pg 61 of The Good Parts*):

```
var is_array = function (value) {
    return value &&
        typeof value === 'object' &&
        typeof value.length === 'number' &&
        typeof value.splice === 'function' &&
        !(value.propertyIsEnumerable('length'));
}
```

But if I'm not mistaken, recently some guy from Google had found a new way on how to test for a JavaScript array, but I just can't remember from where I read it and how the function went.

Can anyone point me to his solution please?

[Update]
The person from Google who apparently discovered this is called Mark Miller.

Now I've also read that from this post that his solution can easily break as well:

```
// native prototype overloaded, some js libraries extends them
Object.prototype.toString= function(){
  return  '[object Array]';
}

function isArray ( obj ) {
  return Object.prototype.toString.call(obj) === '[object Array]';
}

var a = {};
alert(isArray(a)); // returns true, expecting false;
```

So, I ask, is there any way that we can truly check for array validity?

edited **Nov 18 at 22:39**

asked **Nov 18 at 21:55**
Andreas Grech
13.7k ●1 ●41 ●81
83% accept rate

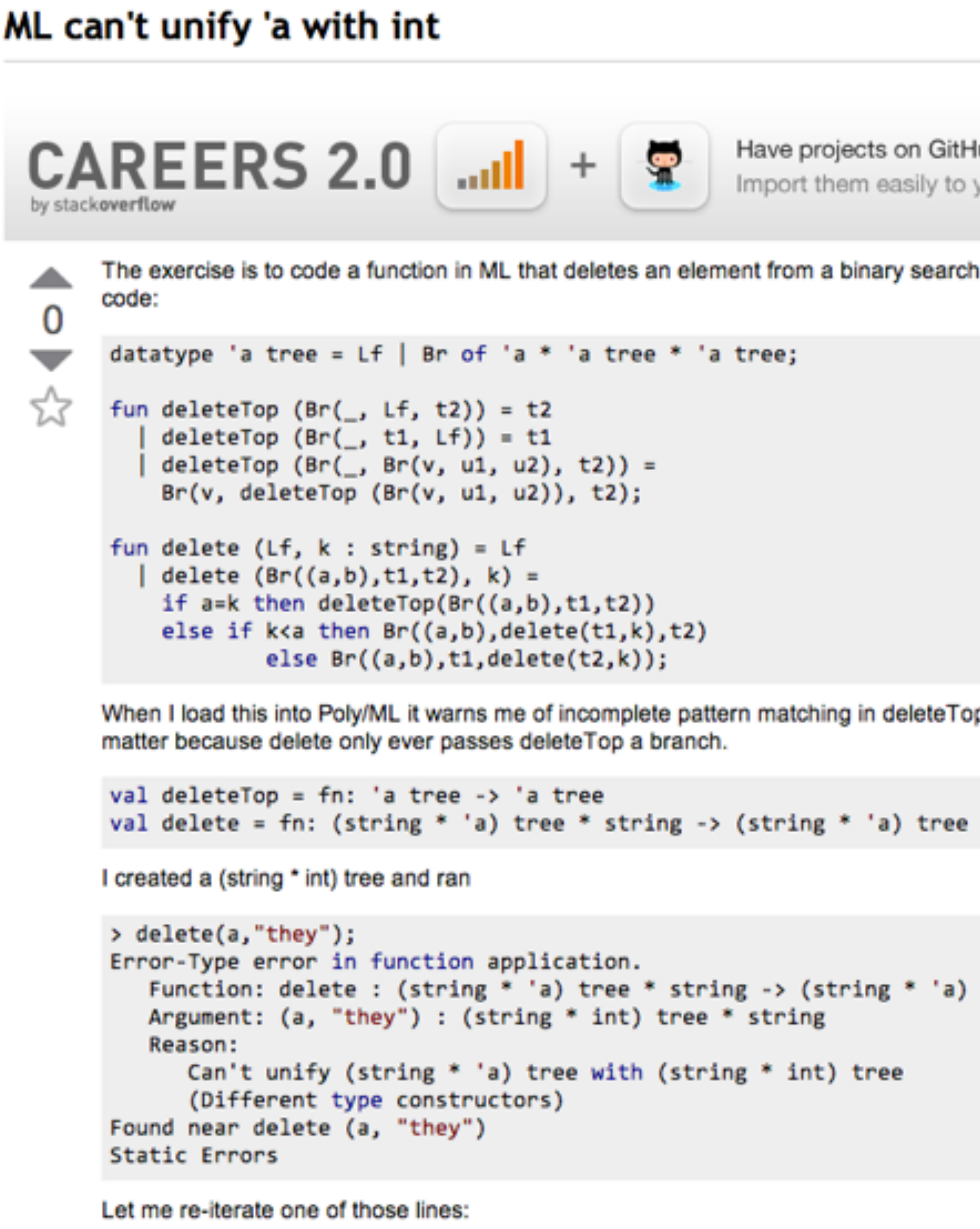
Possible duplicates: stackoverflow.com/questions/1202841 stackoverflow.com/questions/1058427 – CMS Nov 18 at 22:13

Don't close my question, because I have now posted an update to it – Andreas Grech Nov 18 at 22:40

solutions to UNDERSTANDING barriers

Stack Overflow

A searchable repository of human readable explanations of error messages and other strange behavior



ML can't unify 'a with int

The exercise is to code a function in ML that deletes an element from a binary search
code:

```
datatype 'a tree = Lf | Br of 'a * 'a tree * 'a tree;

fun deleteTop (Br(_, Lf, t2)) = t2
  | deleteTop (Br(_, t1, Lf)) = t1
  | deleteTop (Br(_, Br(v, u1, u2), t2)) =
    Br(v, deleteTop (Br(v, u1, u2)), t2);

fun delete (Lf, k : string) = Lf
  | delete (Br((a,b),t1,t2), k) =
    if a=k then deleteTop(Br((a,b),t1,t2))
    else if k<a then Br((a,b),delete(t1,k),t2)
            else Br((a,b),t1,delete(t2,k));
```

When I load this into Poly/ML it warns me of incomplete pattern matching in deleteTop
matter because delete only ever passes deleteTop a branch.

```
val deleteTop = fn: 'a tree -> 'a tree
val delete = fn: (string * 'a) tree * string -> (string * 'a) tree
```

I created a (string * int) tree and ran

```
> delete(a,"they");
Error-Type error in function application.
   Function: delete : (string * 'a) tree * string -> (string * 'a)
   Argument: (a, "they") : (string * int) tree * string
   Reason:
      Can't unify (string * 'a) tree with (string * int) tree
      (Different type constructors)
Found near delete (a, "they")
Static Errors
```

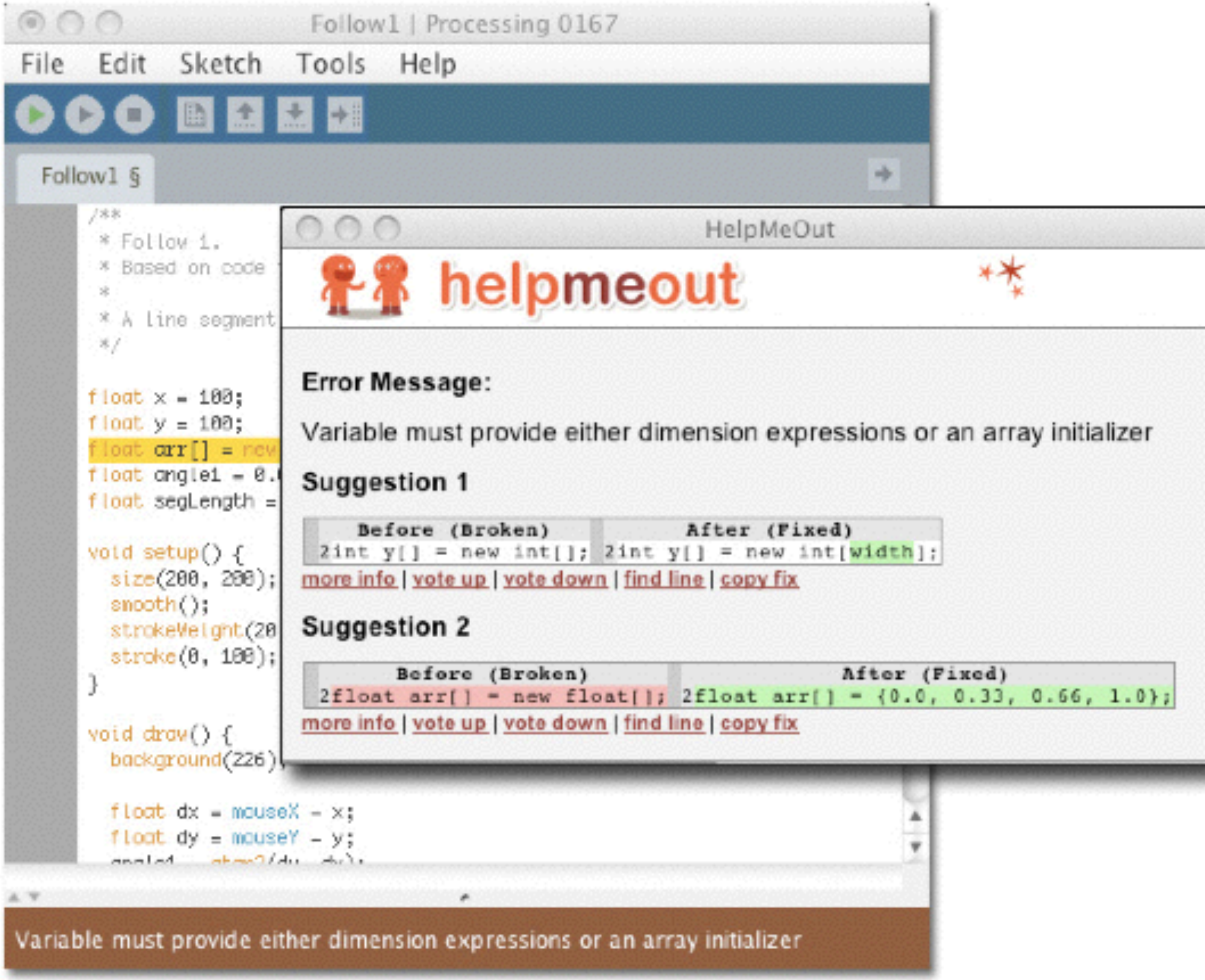
Let me re-iterate one of those lines:

HelpMeOut (2010)

Hartmann, Björn, MacDougall, D., Brandt, J., and Klemmer, S.R. What Would Other Programmers Do? Suggesting Solutions to Error Messages. Proceedings of CHI 2010: ACM Conference on Human Factors in Computing Systems. Atlanta, GA, 2010.

what if fixes to error messages could come from everyone who'd fixed the error before?

(removes UNDERSTANDING barriers)



```
/**
 * Follow 1.
 * Based on code
 *
 * A line segment
 */

float x = 100;
float y = 100;
float arr[] = new
float angle1 = 0.
float segLength =

void setup() {
  size(200, 200);
  smooth();
  strokeWeight(20
  stroke(0, 100);
}

void draw() {
  background(226)

  float dx = mouseX - x;
  float dy = mouseY - y;
```

Error Message:

Variable must provide either dimension expressions or an array initializer

Suggestion 1

Before (Broken)	After (Fixed)
`int y[] = new int[];`	`int y[] = new int[width];`

more info | vote up | vote down | find line | copy fix

Suggestion 2

Before (Broken)	After (Fixed)
`float arr[] = new float[];`	`float arr[] = {0.0, 0.33, 0.66, 1.0};`

more info | vote up | vote down | find line | copy fix

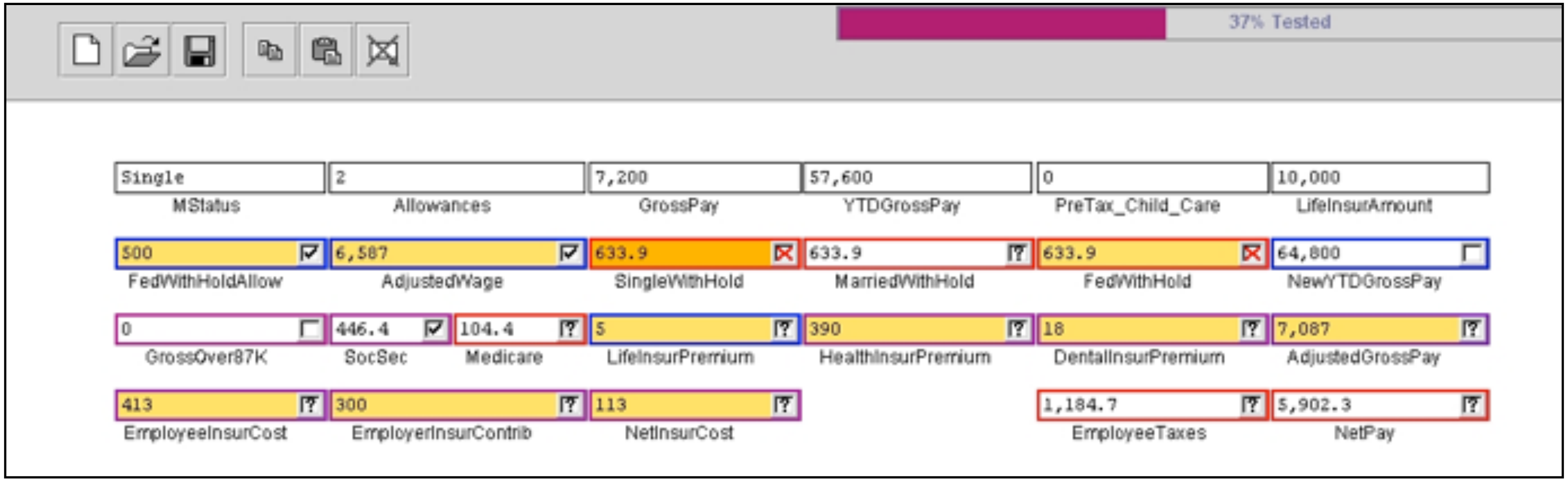
Variable must provide either dimension expressions or an array initializer

WYSIWYT (2000)

"WYSIWYT Testing in the Spreadsheet Paradigm: An Empirical Evaluation", K. Rothermel, C. Cook, M. Burnett, J. Schonfeld, T. R. G. Green, and G. Rothermel, International Conference on Software Engineering, Limerick, Ireland, June 2000, pp 230-239. PDF

what if you could test spreadsheets by simply marking which values are right and wrong?

(removes UNDERSTANDING barriers)



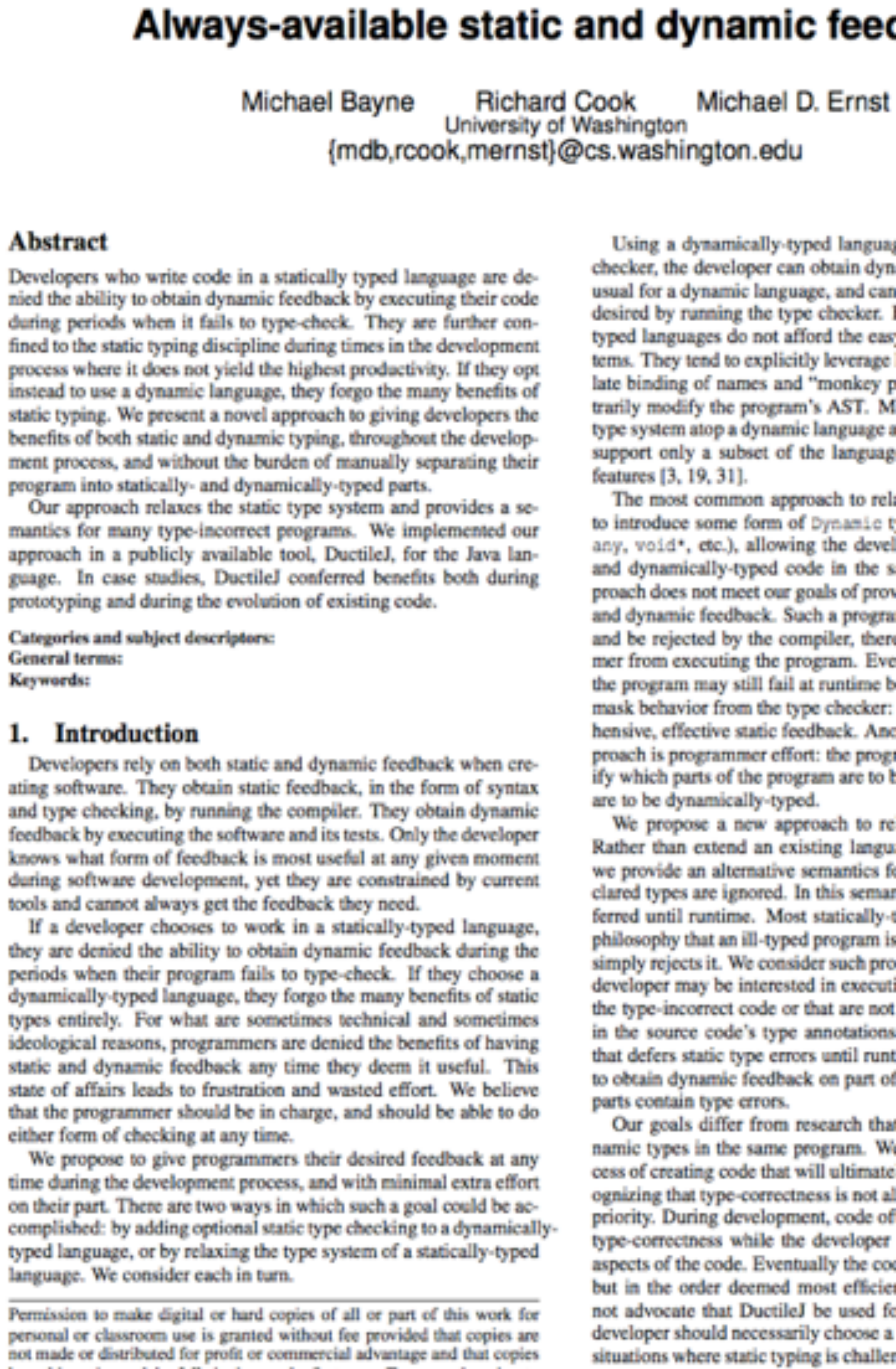
Single	MStatus
2	Allowances
7,200	GrossPay
57,600	YTDGrossPay
0	PreTax_Child_Care
10,000	LifeInsurAmount
500 ☑	FedWithHoldAllow
6,587 ☑	AdjustedWage
633.9 ☒	SingleWithHold
633.9 ?	MarriedWithHold
633.9 ☒	FedWithHold
64,800 □	NewYTDGrossPay
0 □	GrossOver87K
446.4 ☑	SocSec
104.4 ?	Medicare
5 ?	LifeInsurPremium
390 ?	HealthInsurPremium
18 ?	DentalInsurPremium
7,087 ?	AdjustedGrossPay
413 ?	EmployeeInsurCost
300 ?	EmployerInsurContrib
113 ?	NetInsurCost
1,184.7 ?	EmployeeTaxes
5,902.3 ?	NetPay

solutions to INFORMATION barriers

DuctileJ (2011)

what if programmers could run their programs whenever they wanted to, regardless of compiler errors? *(removes INFORMATION barriers)*

"Always-available static and dynamic feedback" by Michael Bayne, Richard Cook, and Michael D. Ernst. In ICSE'11, Proceedings of the 33rd International Conference on Software Engineering, (Waikiki, Hawaii, USA), May 25-27, 2011.



Always-available static and dynamic feed

Michael Bayne Richard Cook Michael D. Ernst
University of Washington
{mdb,rcook,mernst}@cs.washington.edu

Abstract

Developers who write code in a statically typed language are denied the ability to obtain dynamic feedback by executing their code during periods when it fails to type-check. They are further confined to the static typing discipline during times in the development process where it does not yield the highest productivity. If they opt instead to use a dynamic language, they forgo the many benefits of static typing. We present a novel approach to giving developers the benefits of both static and dynamic typing, throughout the development process, and without the burden of manually separating their program into statically- and dynamically-typed parts.

Our approach relaxes the static type system and provides a semantics for many type-incorrect programs. We implemented our approach in a publicly available tool, DuctileJ, for the Java language. In case studies, DuctileJ conferred benefits both during prototyping and during the evolution of existing code.

Categories and subject descriptors:
General terms:
Keywords:

1. Introduction

Developers rely on both static and dynamic feedback when creating software. They obtain static feedback, in the form of syntax and type checking, by running the compiler. They obtain dynamic feedback by executing the software and its tests. Only the developer knows what form of feedback is most useful at any given moment during software development, yet they are constrained by current tools and cannot always get the feedback they need.

If a developer chooses to work in a statically-typed language, they are denied the ability to obtain dynamic feedback during the periods when their program fails to type-check. If they choose a dynamically-typed language, they forgo the many benefits of static types entirely. For what are sometimes technical and sometimes ideological reasons, programmers are denied the benefits of having static and dynamic feedback any time they deem it useful. This state of affairs leads to frustration and wasted effort. We believe that the programmer should be in charge, and should be able to do either form of checking at any time.

We propose to give programmers their desired feedback at any time during the development process, and with minimal extra effort on their part. There are two ways in which such a goal could be accomplished: by adding optional static type checking to a dynamically-typed language, or by relaxing the type system of a statically-typed language. We consider each in turn.

Using a dynamically-typed languag
checker, the developer can obtain dyn
usual for a dynamic language, and can
desired by running the type checker.
typed languages do not afford the eas
tems. They tend to explicitly leverage
late binding of names and "monkey p
trarily modify the program's AST. M
type system atop a dynamic language a
support only a subset of the languag
features [3, 19, 31].

The most common approach to rela
to introduce some form of `Dynamic` t
`any`, `void*`, etc.), allowing the devel
and dynamically-typed code in the s
proach does not meet our goals of prov
and dynamic feedback. Such a progra
and be rejected by the compiler, ther
mer from executing the program. Eve
the program may still fail at runtime b
mask behavior from the type checker:
hensive, effective static feedback. And
proach is programmer effort: the prog
ify which parts of the program are to b
are to be dynamically-typed.

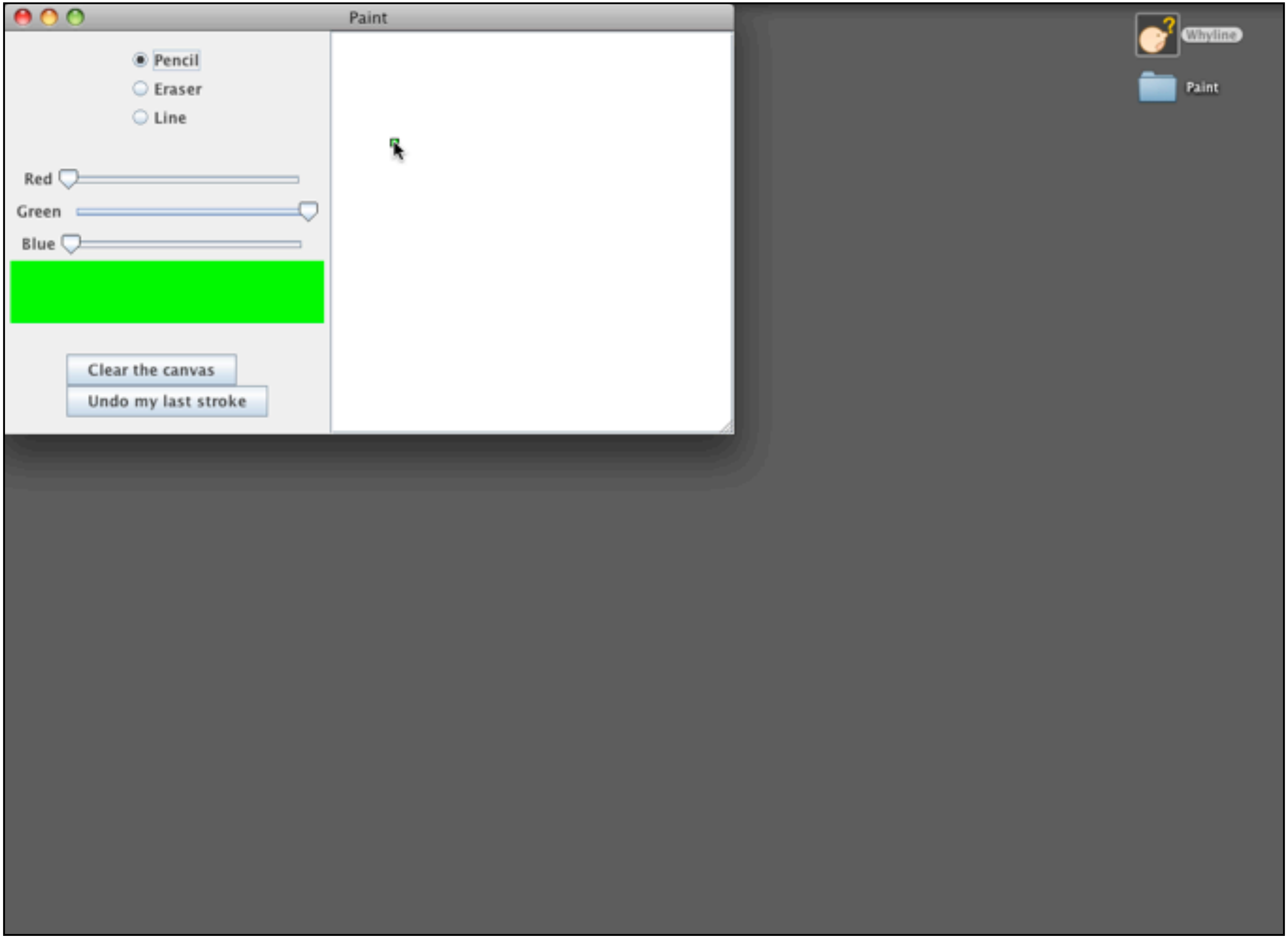
We propose a new approach to re
Rather than extend an existing langu
we provide an alternative semantics fo
clared types are ignored. In this seman
ferred until runtime. Most statically-t
philosophy that an ill-typed program is
simply rejects it. We consider such pro
developer may be interested in executi
the type-incorrect code or that are not
in the source code's type annotations
that defers static type errors until runt
to obtain dynamic feedback on part of
parts contain type errors.

Our goals differ from research that
namic types in the same program. We
cess of creating code that will ultimate
ognizing that type-correctness is not al
priority. During development, code of
type-correctness while the developer
aspects of the code. Eventually the cod
but in the order deemed most efficien
not advocate that DuctileJ be used fo
developer should necessarily choose a
situations where static typing is challen

Whyline

discussion paper!

Ko, A. J. and Myers B.A. (2010). Extracting and Answering Why and Why Not Questions about Java Program Output. ACM Transactions on Software Engineering and Methodology, 20(2), Article 4, August.

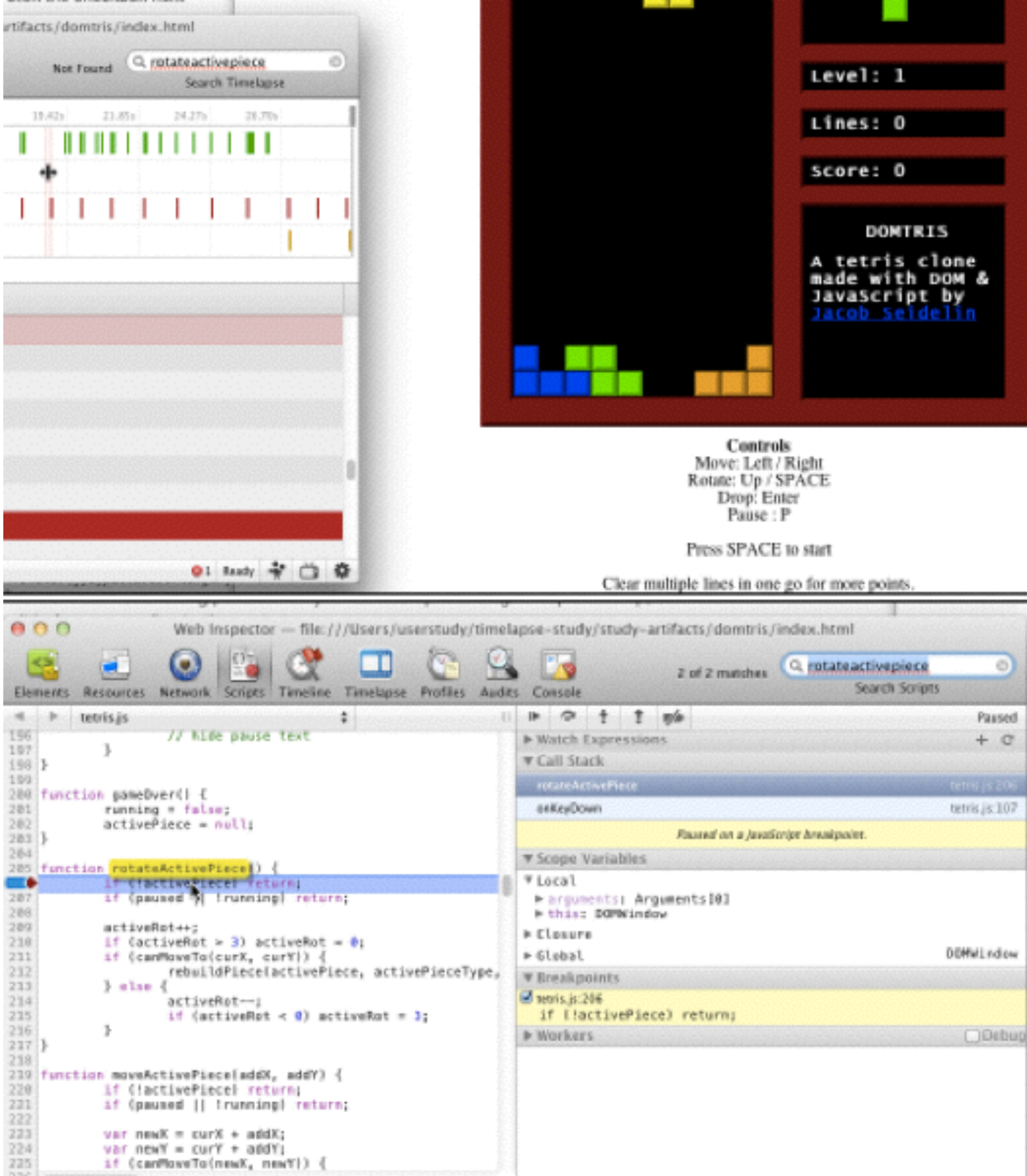


Brian Burg

TimeLapse

precise deterministic replay of web applications

(removes INFORMATION barriers)



Controls
Move: Left / Right
Rotate: Up / SPACE
Drop: Enter
Pause : P

Press SPACE to start

Clear multiple lines in one go for more points.

```
                // hide pause text
        }
}

function gameOver() {
        running = false;
        activePiece = null;
}

function rotateActivePiece() {
        if (!activePiece) return;
        if (paused || !running) return;

        activeRot++;
        if (activeRot > 3) activeRot = 0;
        if (canMoveTo(curX, curY)) {
                rebuildPiece(activePiece, activePieceType,
        } else {
                activeRot--;
                if (activeRot < 0) activeRot = 3;
        }
}

function moveActivePiece(addX, addY) {
        if (!activePiece) return;
        if (paused || !running) return;

        var newX = curX + addX;
        var newY = curY + addY;
        if (canMoveTo(newX, newY)) {
```

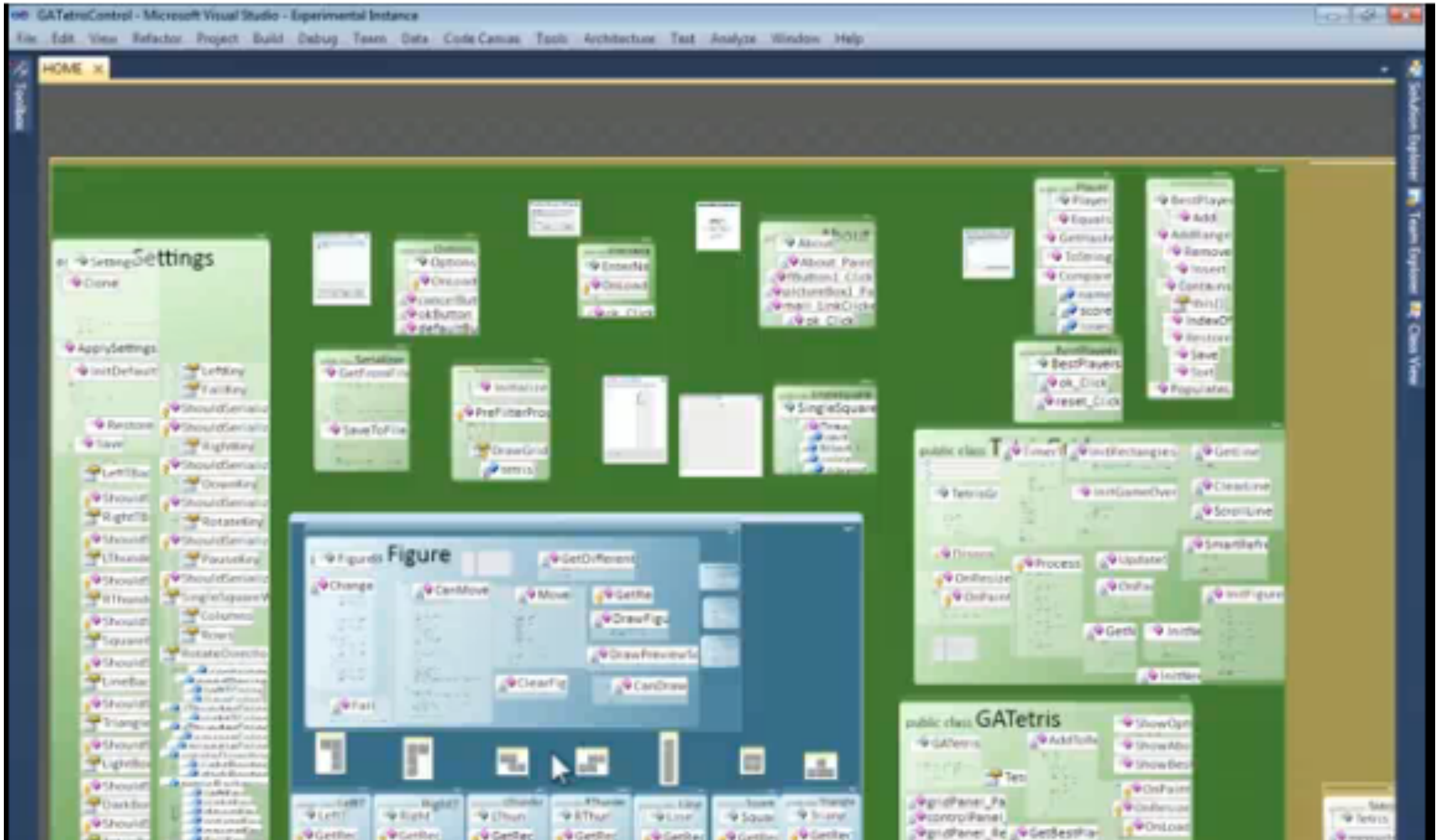
Paused on a JavaScript breakpoint.

```
if (!activePiece) return;
```

Code Canvas (2010)

Robert DeLine, Gina Venolia, and Kael Rowan, Software Development with Code Maps, in Communications of the ACM, vol. 53, no. 8, pp. 48-54, Association for Computing Machinery, Inc., 4 July 2010

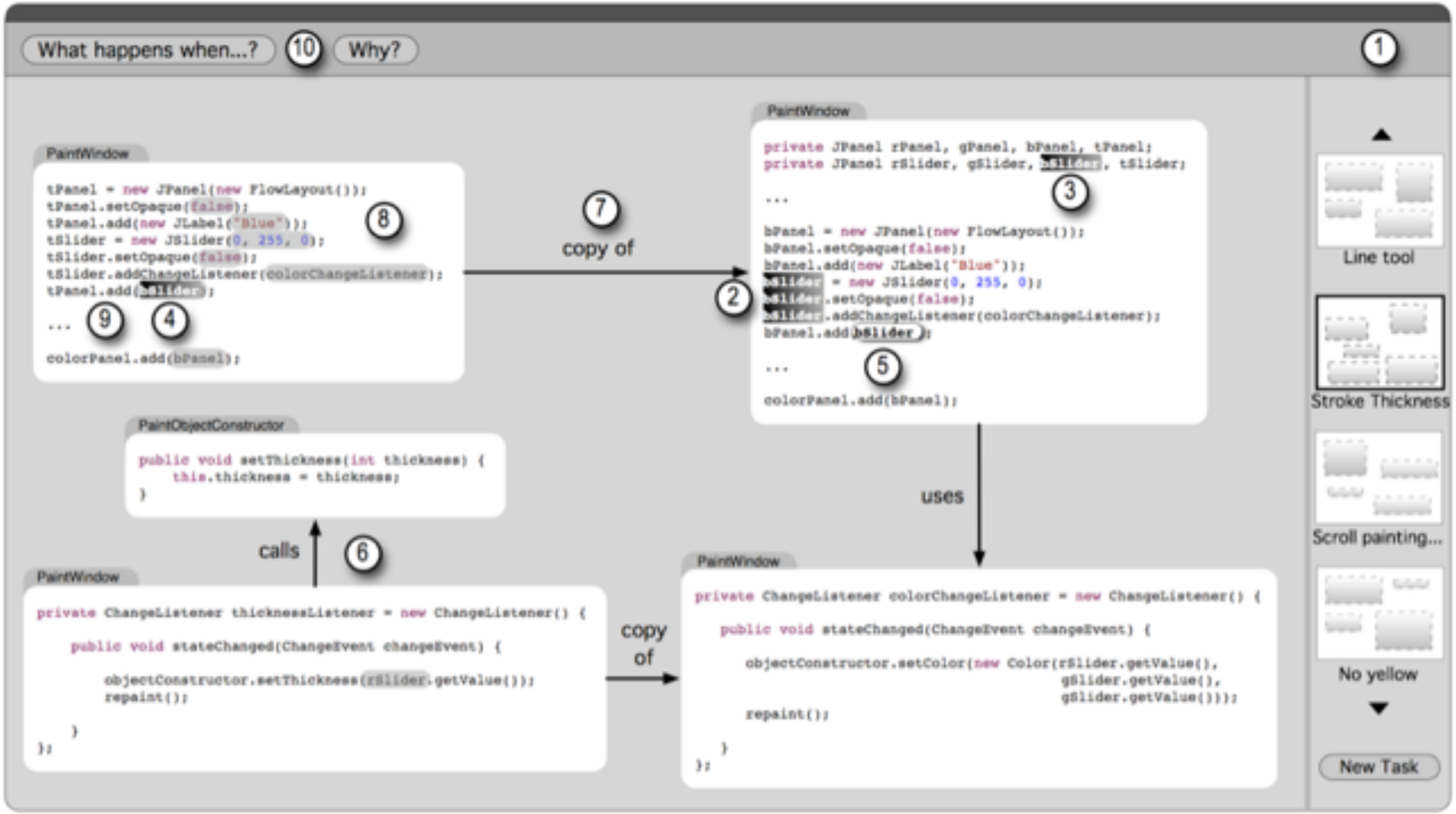
what if you could see all of your code and its dependencies on a single screen? *(removes INFORMATION barriers)*


GATetrisControl - Microsoft Visual Studio - Experimental Instance
File Edit View Refactor Project Build Debug Team Data Code Canvas Tools Architecture Test Analyze Window Help
HOME
Toolbox
Settings
Figure
public class GATetris
Solution Explorer
Team Explorer
Class View

A Working Set Interface (2006)

A design sketch I created

Ko, A. J., Myers B. A., Coblenz, M. J., and Aung, H. H. (2006). An Exploratory Study of How Developers Seek, Relate, and Collect Relevant Information during Software Maintenance Tasks. IEEE Transactions on Software Engineering, 33(12), December, 971-987.



PaintWindow

```
tPanel = new JPanel(new FlowLayout());
tPanel.setOpaque(false);
tPanel.add(new JLabel("Blue"));
tSlider = new JSlider(0, 255, 0);
tSlider.setOpaque(false);
tSlider.addChangeListener(colorChangeListener);
tPanel.add(bSlider);

...

colorPanel.add(bPanel);
```

PaintWindow

```
private JPanel rPanel, gPanel, bPanel, tPanel;
private JPanel rSlider, gSlider, bSlider, tSlider;

...

bPanel = new JPanel(new FlowLayout());
bPanel.setOpaque(false);
bPanel.add(new JLabel("Blue"));
bSlider = new JSlider(0, 255, 0);
bSlider.setOpaque(false);
bSlider.addChangeListener(colorChangeListener);
bPanel.add(bSlider);

...

colorPanel.add(bPanel);
```

PaintObjectConstructor

```
public void setThickness(int thickness) {
    this.thickness = thickness;
}
```

PaintWindow

```
private ChangeListener thicknessListener = new ChangeListener() {

    public void stateChanged(ChangeEvent changeEvent) {

        objectConstructor.setThickness(rSlider.getValue());
        repaint();

    }
};
```

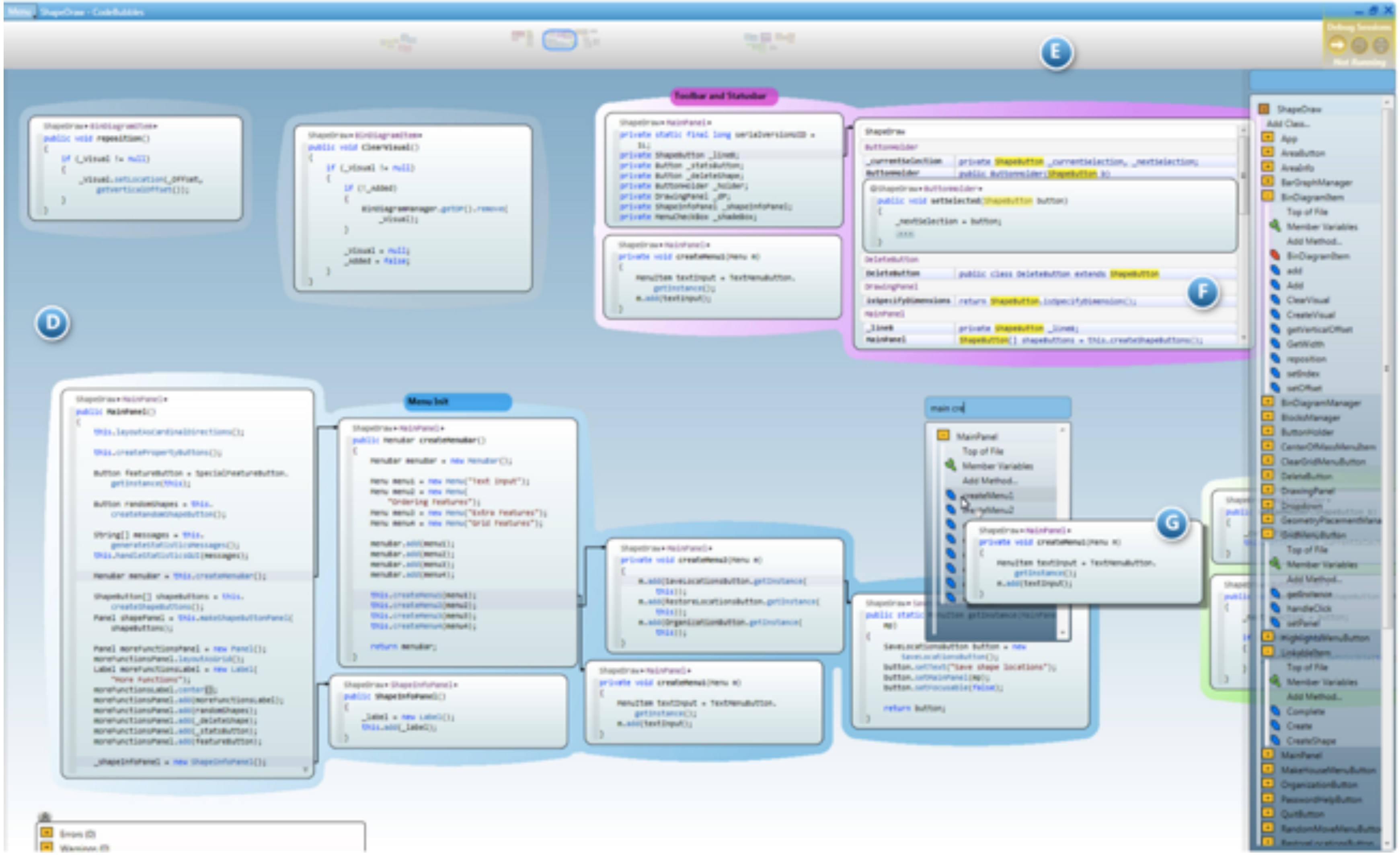
PaintWindow

```
private ChangeListener colorChangeListener = new ChangeListener() {

    public void stateChanged(ChangeEvent changeEvent) {

        objectConstructor.setColor(new Color(rSlider.getValue(),
                                             gSlider.getValue(),
                                             gSlider.getValue()));
        repaint();

    }
};
```

Code Bubbles (2010)

Code Bubbles: A Working Set-based Interface for Code Understanding and Maintanence. Andrew Bragdon, Robert Zeleznik, Steven P. Reiss, Suman Karumuri, William Cheung, Joshua Kaplan, Christopher Coleman, Ferdi Adeputra, and Joseph J. LaViola Jr. To appear in: Proceedings of the 28th International Conference on Human Factors in Computing Systems (2010).

what if IDEs sliced code up into snippets instead of files? *(removes INFORMATION barriers)*



```
ShapeDraw▸BinDiagramItem▸
public void reposition()
{
    if (_visual != null)
    {
        _visual.setLocation(_offset,
            getVerticalOffset());
    }
}
```

```
ShapeDraw▸BinDiagramItem▸
public void ClearVisual()
{
    if (_visual != null)
    {
        if (!_added)
        {
            BinDiagramManager.getDM().remove(
                _visual);
        }

        _visual = null;
        _added = false;
    }
}
```

```
ShapeDraw▸MainPanel▸
private static final long serialVersionUID =
    1L;
private ShapeButton _lineB;
private Button _statsButton;
private Button _deleteShape;
private ButtonHolder _holder;
private DrawingPanel _dP;
private ShapeInfoPanel _shapeInfoPanel;
private MenuCheckbox _shadeBox;
```

```
ShapeDraw▸MainPanel▸
private void createMenu1(Menu m)
{
    MenuItem textInput = TextMenuButton.
        getInstance();
    m.add(textInput);
}
```

```
ShapeDraw
ButtonHolder
_currentSelection    private ShapeButton _currentSelection, _nextSelection;
ButtonHolder         public ButtonHolder(ShapeButton b)
@ShapeDraw▸ButtonHolder▸
public void setSelected(ShapeButton button)
{
    _nextSelection = button;
    ...
}
DeleteButton
DeleteButton         public class DeleteButton extends ShapeButton
DrawingPanel
isSpecifyDimensions  return ShapeButton.isSpecifyDimension();
MainPanel
_lineB               private ShapeButton _lineB;
MainPanel            ShapeButton[] shapeButtons = this.createShapeButtons();
```

```
ShapeDraw▸MainPanel▸
public MainPanel()
{
    this.layoutAsCardinalDirections();

    this.createPropertyButtons();

    Button featureButton = SpecialFeatureButton.
        getInstance(this);

    Button randomShapes = this.
        createRandomShapeButton();

    String[] messages = this.
        generateStatisticsMessages();
    this.handleStatisticsGUI(messages);

    MenuBar menuBar = this.createMenuBar();

    ShapeButton[] shapeButtons = this.
        createShapeButtons();
    Panel shapePanel = this.makeShapeButtonPanel(
        shapeButtons);

    Panel moreFunctionsPanel = new Panel();
    moreFunctionsPanel.layoutAsGrid();
    Label moreFunctionsLabel = new Label(
        "More Functions");
    moreFunctionsLabel.center();
    moreFunctionsPanel.add(moreFunctionsLabel);
    moreFunctionsPanel.add(randomShapes);
    moreFunctionsPanel.add(_deleteShape);
    moreFunctionsPanel.add(_statsButton);
    moreFunctionsPanel.add(featureButton);

    _shapeInfoPanel = new ShapeInfoPanel();
```

```
ShapeDraw▸MainPanel▸
public MenuBar createMenuBar()
{
    MenuBar menuBar = new MenuBar();

    Menu menu1 = new Menu("Text Input");
    Menu menu2 = new Menu(
        "Ordering Features");
    Menu menu3 = new Menu("Extra Features");
    Menu menu4 = new Menu("Grid Features");

    menuBar.add(menu1);
    menuBar.add(menu2);
    menuBar.add(menu3);
    menuBar.add(menu4);

    this.createMenu1(menu1);
    this.createMenu2(menu2);
    this.createMenu3(menu3);
    this.createMenu4(menu4);

    return menuBar;
}
```

```
ShapeDraw▸ShapeInfoPanel▸
public ShapeInfoPanel()
{
    _label = new Label();
    this.add(_label);
}
```

```
ShapeDraw▸MainPanel▸
private void createMenu2(Menu m)
{
    m.add(SaveLocationsButton.getInstance(
        this));
    m.add(RestoreLocationsButton.getInstance(
        this));
    m.add(OrganizationButton.getInstance(
        this));
}
```

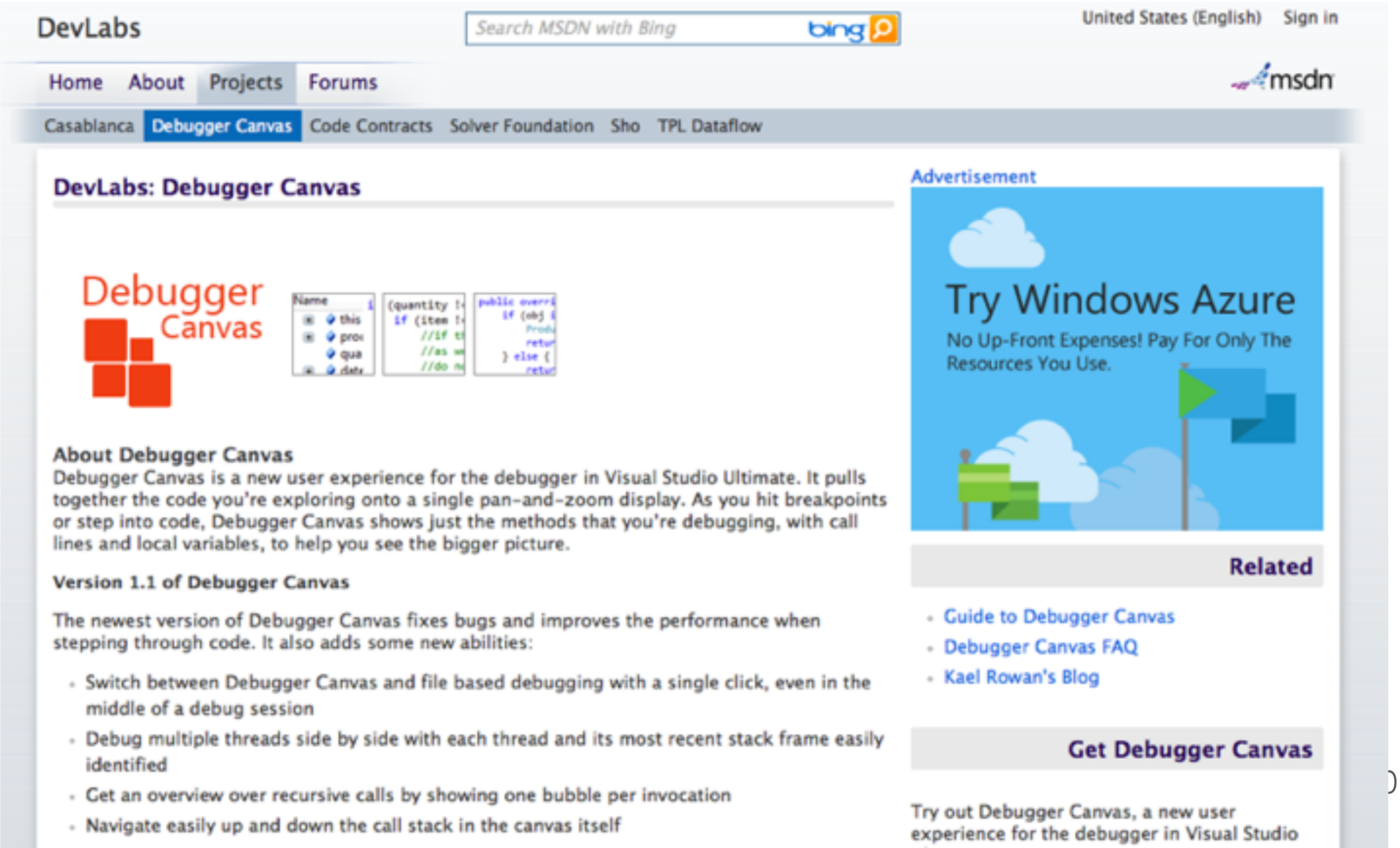
```
ShapeDraw▸MainPanel▸
private void createMenu1(Menu m)
{
    MenuItem textInput = TextMenuButton.
        getInstance();
    m.add(textInput);
}
```

```
ShapeDraw▸SaveLocationsButton▸
public static MenuItem getInstance(MainPanel
    mp)
{
    SaveLocationsButton button = new
        SaveLocationsButton();
    button.setText("Save shape locations");
    button.setMainPanel(mp);
    button.setFocusable(false);

    return button;
}
```

```
ShapeDraw▸MainPanel▸
private void createMenu1(Menu m)
{
    MenuItem textInput = TextMenuButton.
        getInstance();
    m.add(textInput);
}
```

Debugger Canvas

6 years from idea to Visual Studio plug-in



DevLabs: Debugger Canvas

About Debugger Canvas

Debugger Canvas is a new user experience for the debugger in Visual Studio Ultimate. It pulls together the code you're exploring onto a single pan-and-zoom display. As you hit breakpoints or step into code, Debugger Canvas shows just the methods that you're debugging, with call lines and local variables, to help you see the bigger picture.

Version 1.1 of Debugger Canvas

The newest version of Debugger Canvas fixes bugs and improves the performance when stepping through code. It also adds some new abilities:

- Switch between Debugger Canvas and file based debugging with a single click, even in the middle of a debug session
- Debug multiple threads side by side with each thread and its most recent stack frame easily identified
- Get an overview over recursive calls by showing one bubble per invocation
- Navigate easily up and down the call stack in the canvas itself

Related

- Guide to Debugger Canvas
- Debugger Canvas FAQ
- Kael Rowan's Blog

Get Debugger Canvas

Try out Debugger Canvas, a new user experience for the debugger in Visual Studio

WHAT'S NEXT?

010100100100101001010010
100101001010010101001010
101010010101010010101010
100100010010111001010101
001010010101010010010100
101010100010101010010101
001010101001010101100100

PRODUCTIVITY IS DONE

New dev tools are fine, but they're increasingly incremental, niche and irrelevant to industry

- Productivity is not the problem, it's learning, expertise, design, iteration, scale, domains

Look ahead 20 years…

- What will we be coding?
- Who will be coding it?
- Who will they coding it for?
- How should they be coding it?

NEW KINDS OF CODE

Machine-learned

How do we code against uncertainty?

Biological

How do we code against anatomy and physiology?

Crowd-powered

How do we code against human cognition?

Cloud-powered

How do we code against data centers, social networks, and massive data sets?

BETTER DEVELOPERS

Instead of making better tools, why not make better developers?

Training end-users

How can we insert education into end-user programming tools?

Facilitating experts

How can we help engineers make more effective decisions?

Teaching novices

How can we teach learners more efficiently and effectively?

Structuring teams

How can we help teams coordinate work more effectively?

Teaching Problem Solving (2016)

Loksa, D., Ko, A.J., Jernigan, W., Oleson, A., Mendez, C., Burnett, M.M. Programming, Problem Solving, and Self-Awareness: Effects of Explicit Guidance. CHI 2016.

What if we taught novice programmers how to structure and reflect on their programming efforts?

One hour of instruction on six stages:

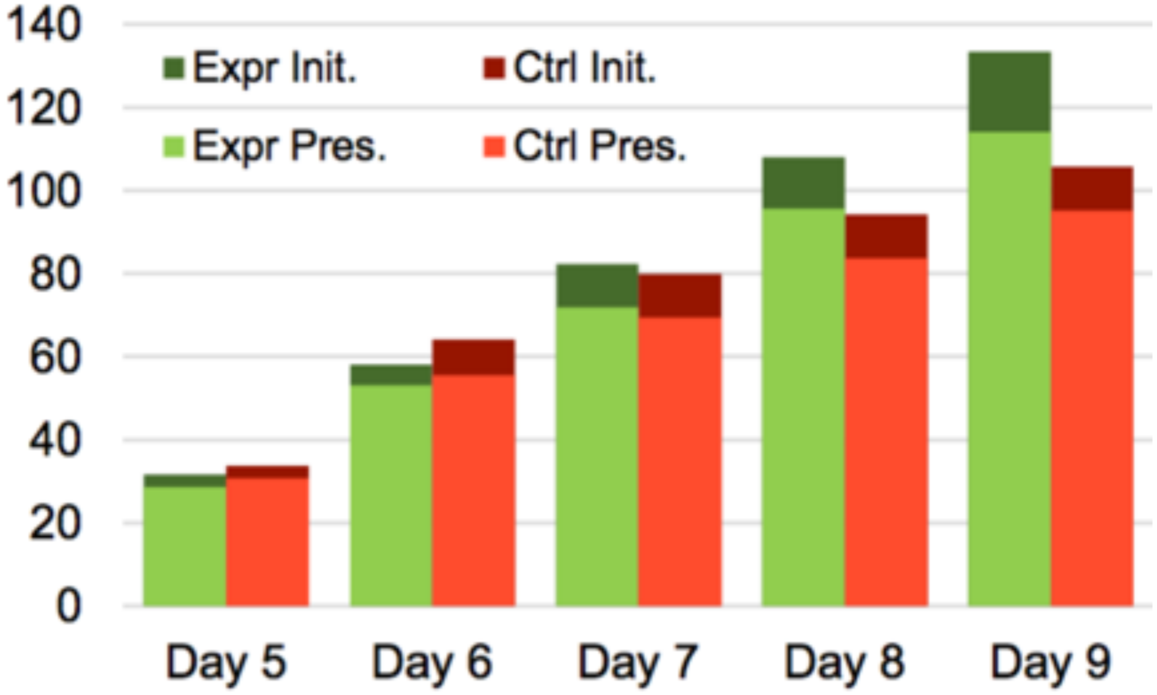
1) interpreting problem prompt,
2) search for analogous problems,
3) search for solutions,
4) evaluate solutions,
5) implement solution,
6) evaluate implementation

Upon help requests, prompt for reflection: “What are you doing, why are you doing it, and is it working”?

Teaching Problem Solving (2016), cont.

Two camps, two weeks, 25 students each

20 requirements to implement for a web application



Campers with the instruction were more productive, more creative, more independent, more confident in their ability to code and learn other non-coding skills

CS Ed for All

President Obama just announced a $4 billion initiative to:

- Prepare and place 10,000 CS teachers in U.S. public schools
- Fund $125 million in CS ed research **per year**, including NSF graduate fellowships, CAREER grants, basic research funding, faculty positions, etc.

The computing education research community will grow from ~50 researchers now to ~500 researchers in the next twenty years