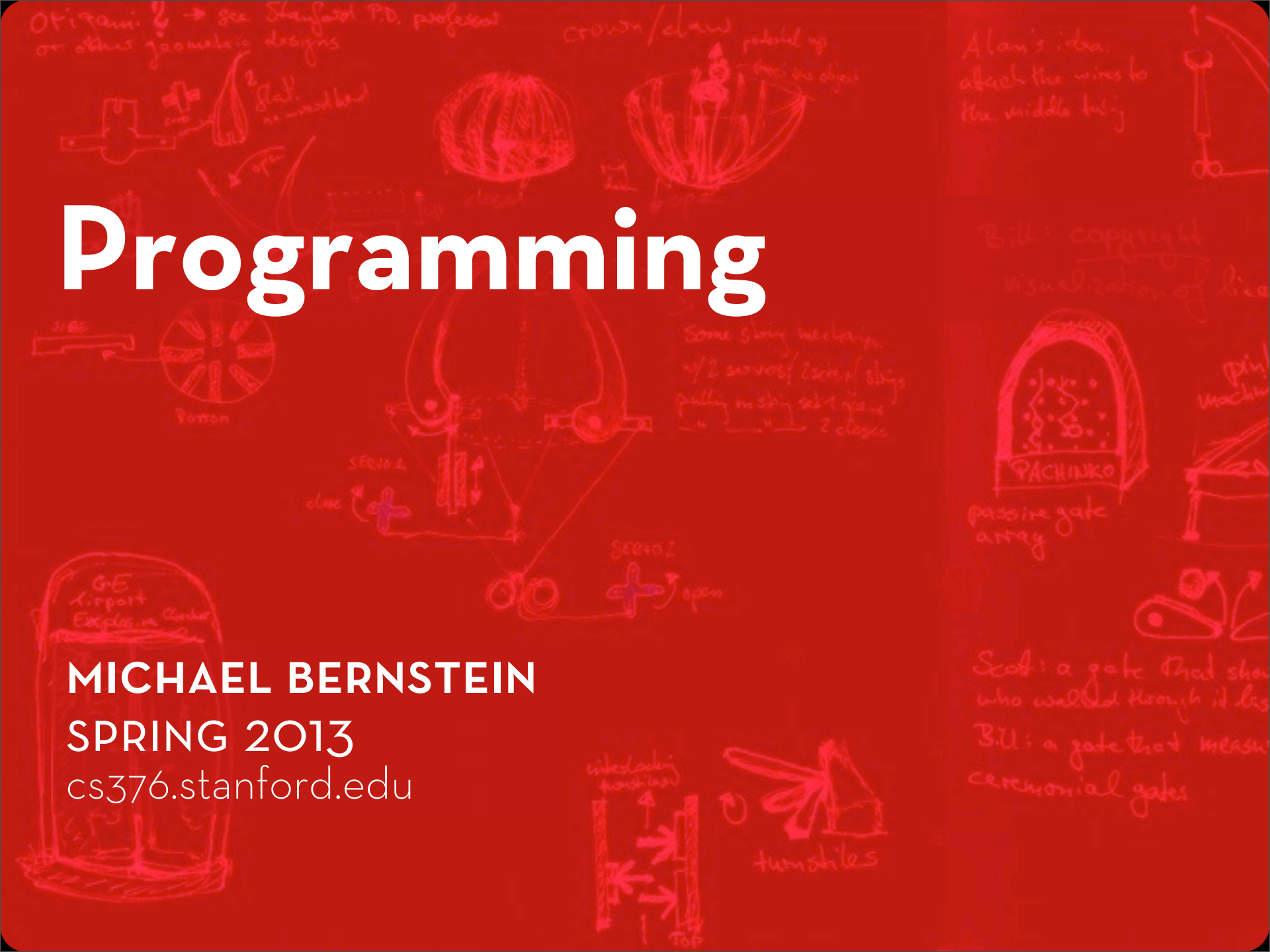

Programming
MICHAEL BERNSTEIN
SPRING 2013
cs376.stanford.edu

A Small Matter of Programming

- Software engineering is a highly complex task, a microcosm of many challenges in HCI
- Making software engineering more accessible could empower millions to customize applications and write programs

Research agenda: HCI and programming

- Understand the challenges in programming
- Design more effective software engineering interfaces
- Aid novices in learning to program or writing programs
- Abstract best practices into toolkits

Understanding programmers and programming

Information Needs in Programming [Ko, DeLine and Venolia , ICSE '07]

- Observed 17 developers in 90-minute sessions and transcribed all activities

Information Needs in Programming [Ko, DeLine and Venolia , ICSE '07]

- Observed 17 developers in 90-minute sessions and transcribed all activities
- Thematic coding of information needs
  - **Writing code** e.g., how do I use this method?
  - **Submitting a change** e.g., which files are included?
  - **Triaging bugs** e.g., is the problem worth fixing?
  - **Reproducing failure** e.g., what are failure conditions?
  - **Understanding execution** e.g., what caused this behavior?
  - **Design** e.g., why is the code implemented this way?
  - **Awareness** e.g., what are my collaborators working on?

Information Needs in Programming [Ko, DeLine and Venolia , ICSE '07]

- Observed 17 developers in 90-minute sessions and transcribed all activities
- Thematic coding of information needs
  - **Writing code** e.g., how do I use this method?
  - **Submitting a change** e.g., which files are included?
  - **Triaging bugs** e.g., is the problem worth fixing?
  - **Reproducing failure** e.g., what are failure conditions?
  - **Understanding execution** e.g., what caused this behavior?
  - **Design** e.g., why is the code implemented this way?
  - **Awareness** e.g., what are my collaborators working on?
- Most common need: collaborator awareness

Obstacles to learning APIs

[Robillard and DeLine, Empir. Software Engineering 2011]

- Survey and in-person interviews, combined reaching 440 professional software engineers

Obstacles to learning APIs

[Robillard and DeLine, Empir. Software Engineering 2011]

- Survey and in-person interviews, combined reaching 440 professional software engineers
- Biggest challenge: inadequate documentation

Obstacles to learning APIs

[Robillard and DeLine, Empir. Software Engineering 2011]

- Survey and in-person interviews, combined reaching 440 professional software engineers
- Biggest challenge: inadequate documentation
  - **API intent**: how it was intended to be used

    "Nowhere in there does it say, and we intended to be used for a few graphics of small size because the memory footprint is going to be this."

Obstacles to learning APIs

[Robillard and DeLine, Empir. Software Engineering 2011]

- Survey and in-person interviews, combined reaching 440 professional software engineers
- Biggest challenge: inadequate documentation
  - **API intent**: how it was intended to be used
    "Nowhere in there does it say, and we intended to be used for a few graphics of small size because the memory footprint is going to be this."
  - **Code examples**: snippets, tutorials, working apps

Obstacles to learning APIs

[Robillard and DeLine, Empir. Software Engineering 2011]

- Survey and in-person interviews, combined reaching 440 professional software engineers
- Biggest challenge: inadequate documentation
  - **API intent**: how it was intended to be used
    "Nowhere in there does it say, and we intended to be used for a few graphics of small size because the memory footprint is going to be this."
  - **Code examples**: snippets, tutorials, working apps
  - **Penetrability**: how much detail and implementation to expose?

Web foraging and programming

[Brandt et al., CHI '09]

- Laboratory study: ask programmers to implement a chat room in PHP
- Programmers make heavy use of the web
  - **JIT learning** of new skills
  - **Clarifying** existing skills
  - **Reminding** themselves of details
- Average participant spent **19%** of their programming time on the web

Software engineering interfaces

Goals of software engineering interface research

- Design a better toolbench, produce a better programmer
- Typically assume that the programming language is static, but the interface of the IDE can be molded

Example-centric programming

[Brandt et al., CHI '10]

- Close the loop between the development environment and web search
- Autocomplete code via web examples
- [video]

Asking 'why' questions of code

[Ko and Myers CHI '04, ICSE '09]

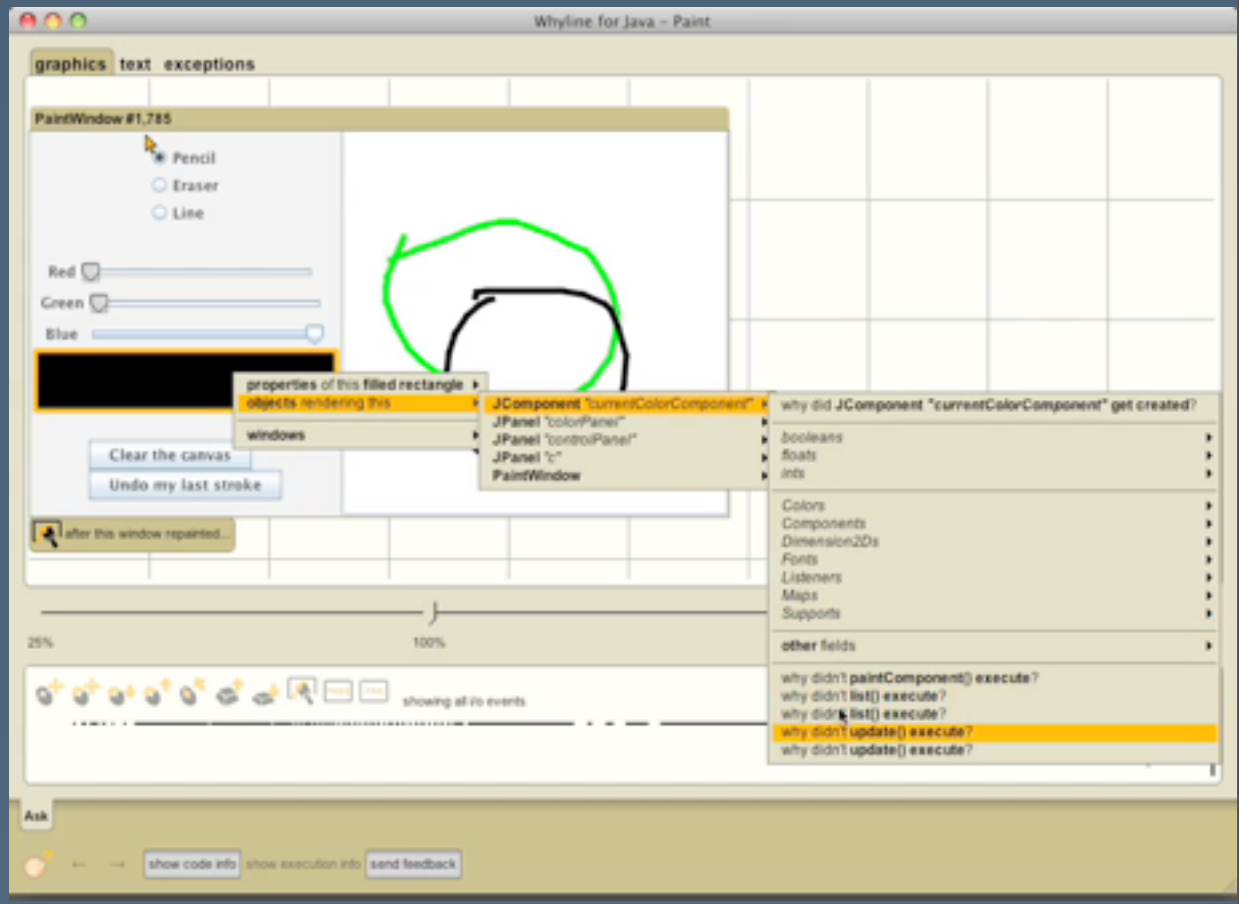
- Debugging problems often reduce to "why" questions
- Analyze program traces to answer them

Whyline for Java

Asking 'why' questions of code

[Ko and Myers CHI '04, ICSE '09]

- Debugging problems often reduce to "why" questions
- Analyze program traces to answer them

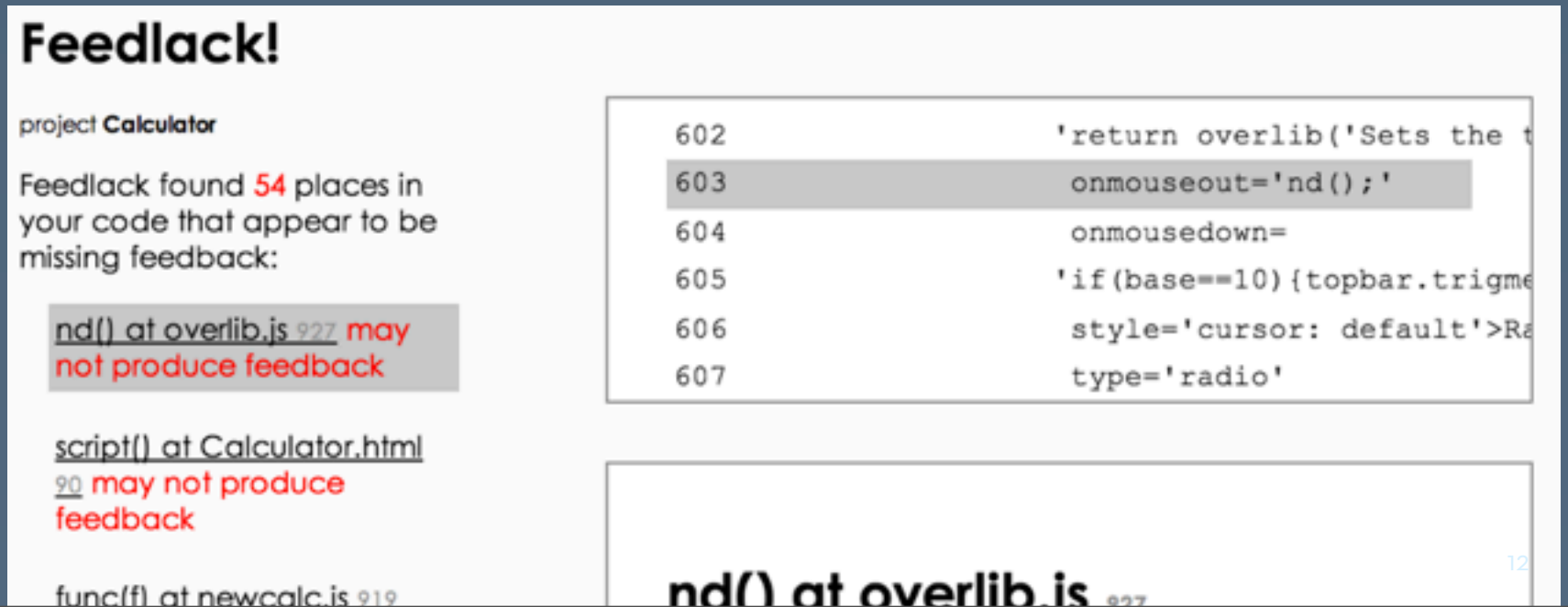


Whyline for Java

Missing user-facing feedback

[Ko and Zhang, CHI '11]

- Usability heuristic: all user inputs should produce some form of feedback
- Statically analyze code to identify user inputs that produce no feedback



Keyword programming

[Little and Miller, UIST '06, ASE '09]

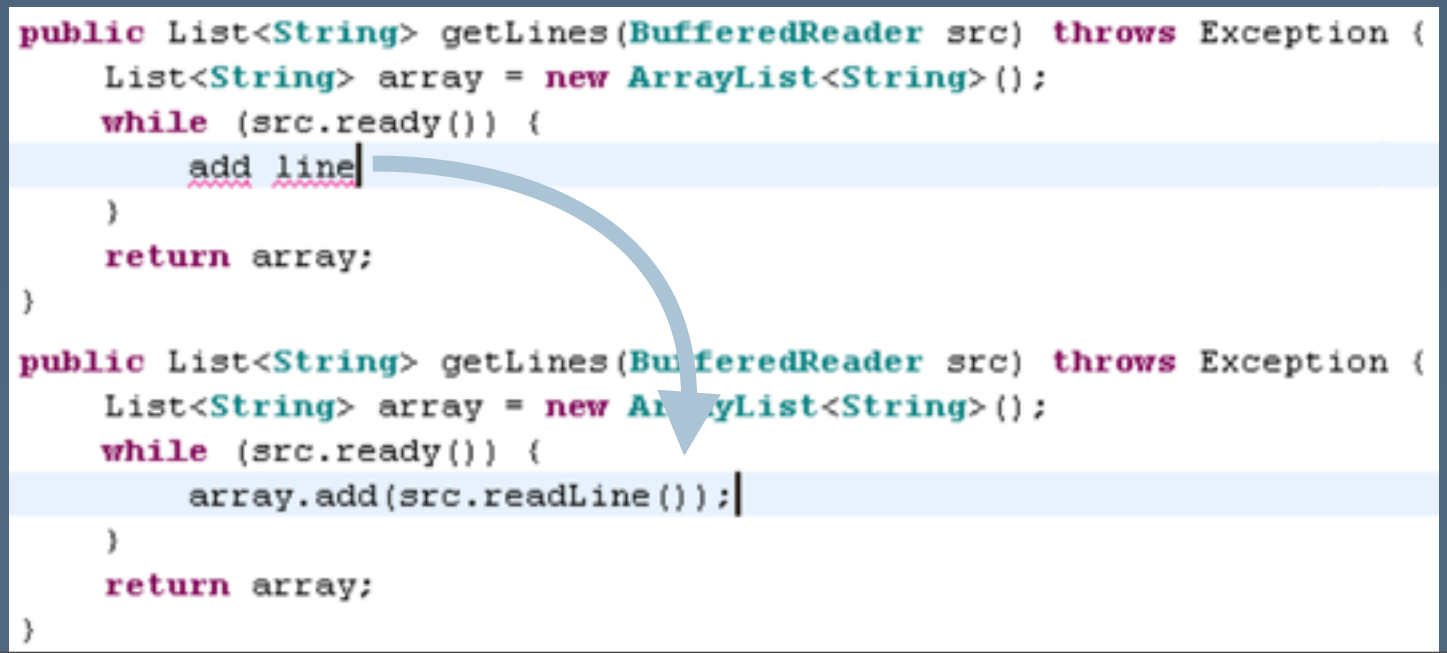
- Many applications have macro interfaces, but they are difficult to learn
- Allow keyword search over an API: e.g., `click search button` or `left margin 2 inches`

```
public List<String> getLines(BufferedReader src) throws Exception {
    List<String> array = new ArrayList<String>();
    while (src.ready()) {
        add line
    }
    return array;
}
```

Keyword programming

[Little and Miller, UIST '06, ASE '09]

- Many applications have macro interfaces, but they are difficult to learn
- Allow keyword search over an API: e.g., `click search button` or `left margin 2 inches`



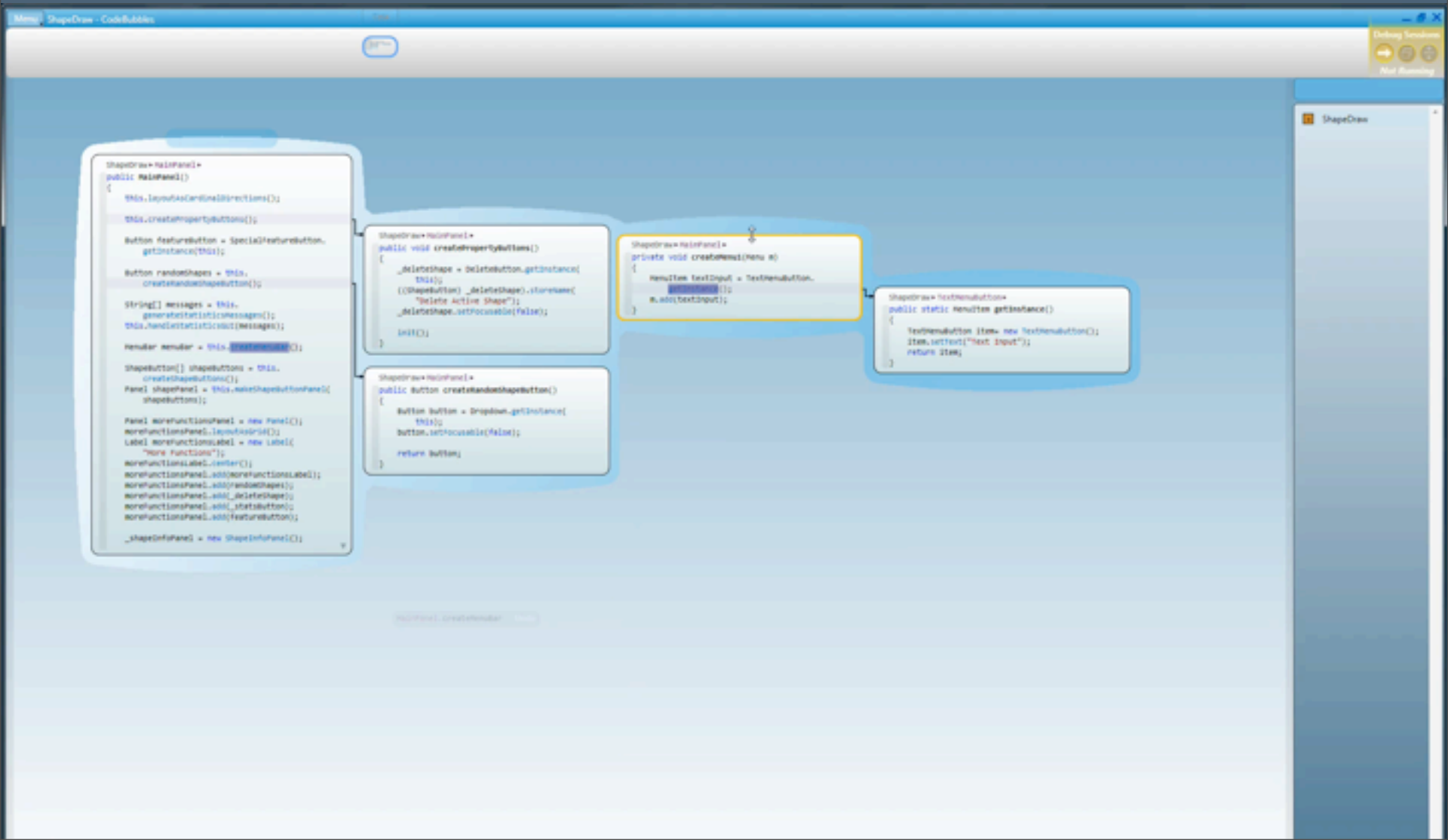
Visual layout of code snippets

[Bragdon et al., CHI '10]

- Most engineering time is spent navigating across multiple related code snippets
- So, design for many small windows into files

Visual layout of code snippets

[Bragdon et al., CHI '10]



Learning programming

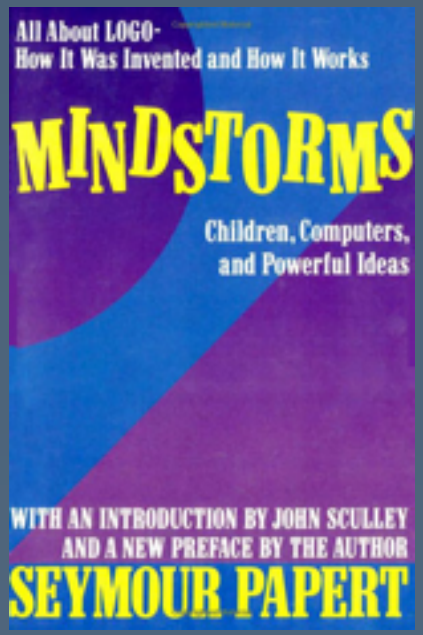
Goals of programming education

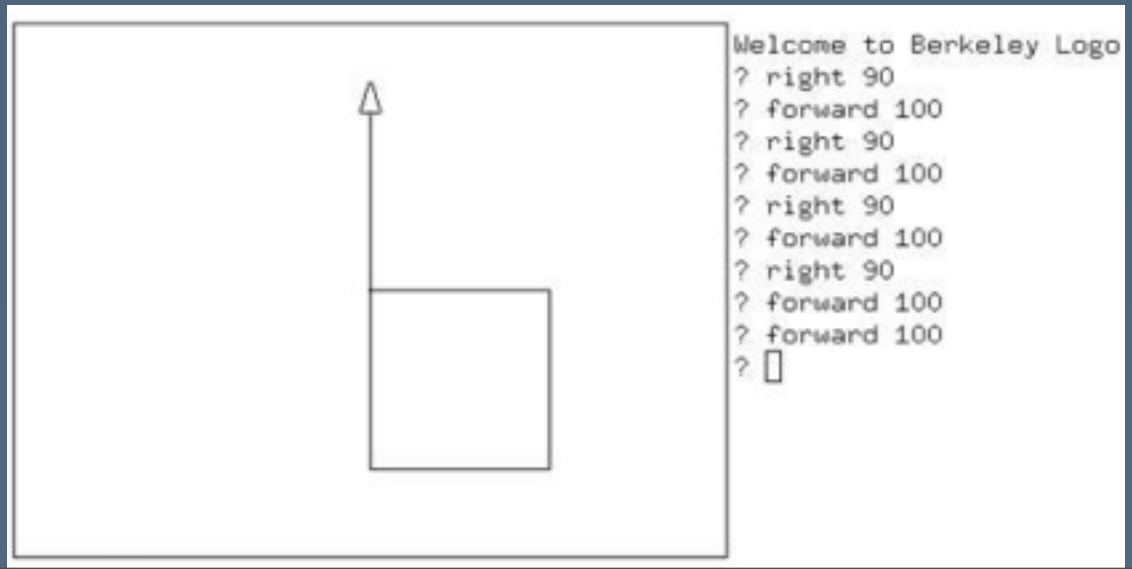
- Make programming accessible to new populations: children, scripters, interested amateurs
- Tools and innovations depend on the population

Logo: programming for children

[Papert '93]

- **Constructionist** learning: learning happens most effectively when people are making tangible objects
- Lego Mindstorms followed this mold and was named after it





Alice: 3D animations for intro programming [Cooper, Dann, and Pausch CCSC '00]

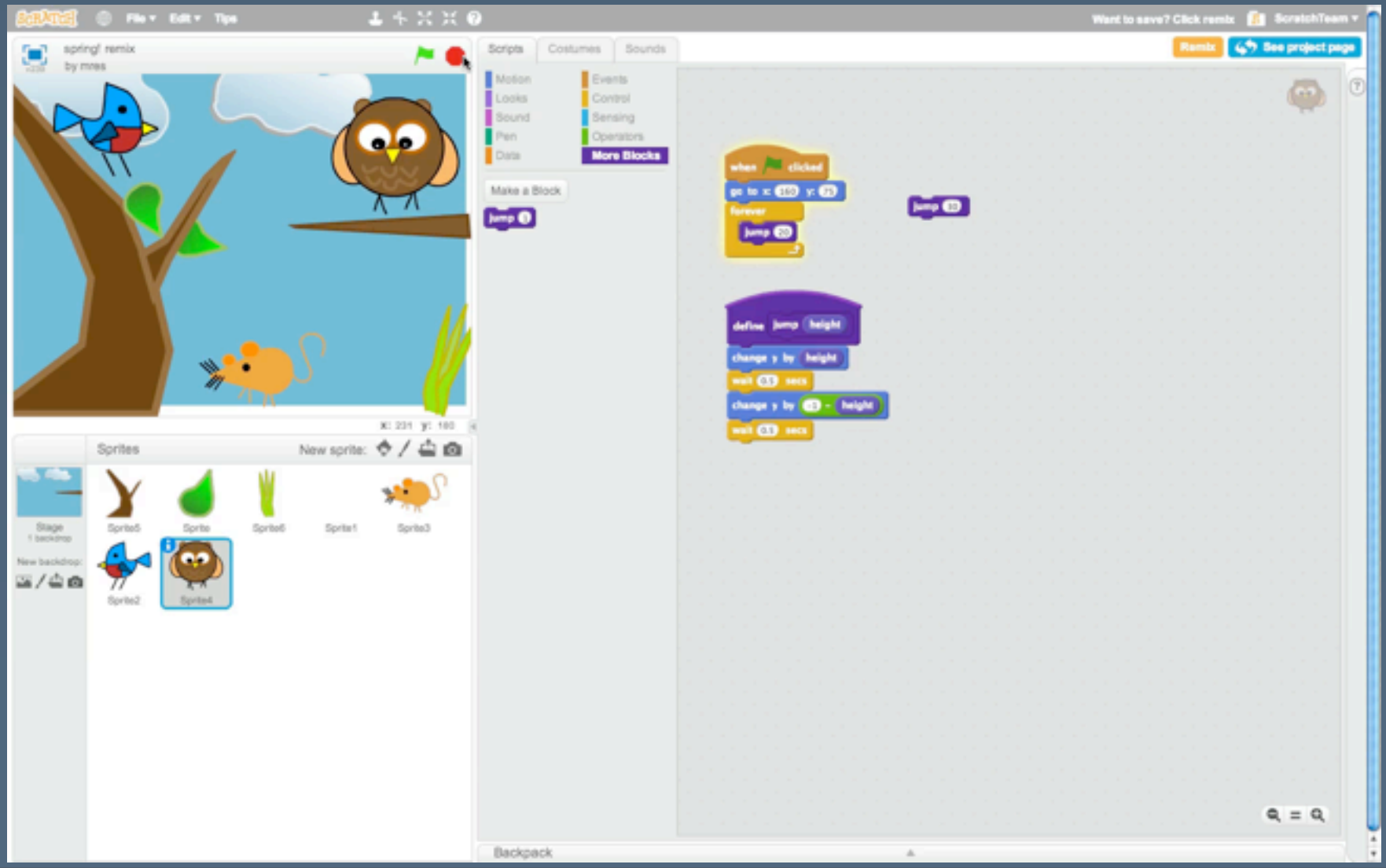
- Visualization and animation help new programmers understand program execution
- Evolution of Logo and Karel the Robot into a three-dimensional environment

Scratch: kids remix and create

[Resnick et al., CACM '09]

Scratch: kids remix and create

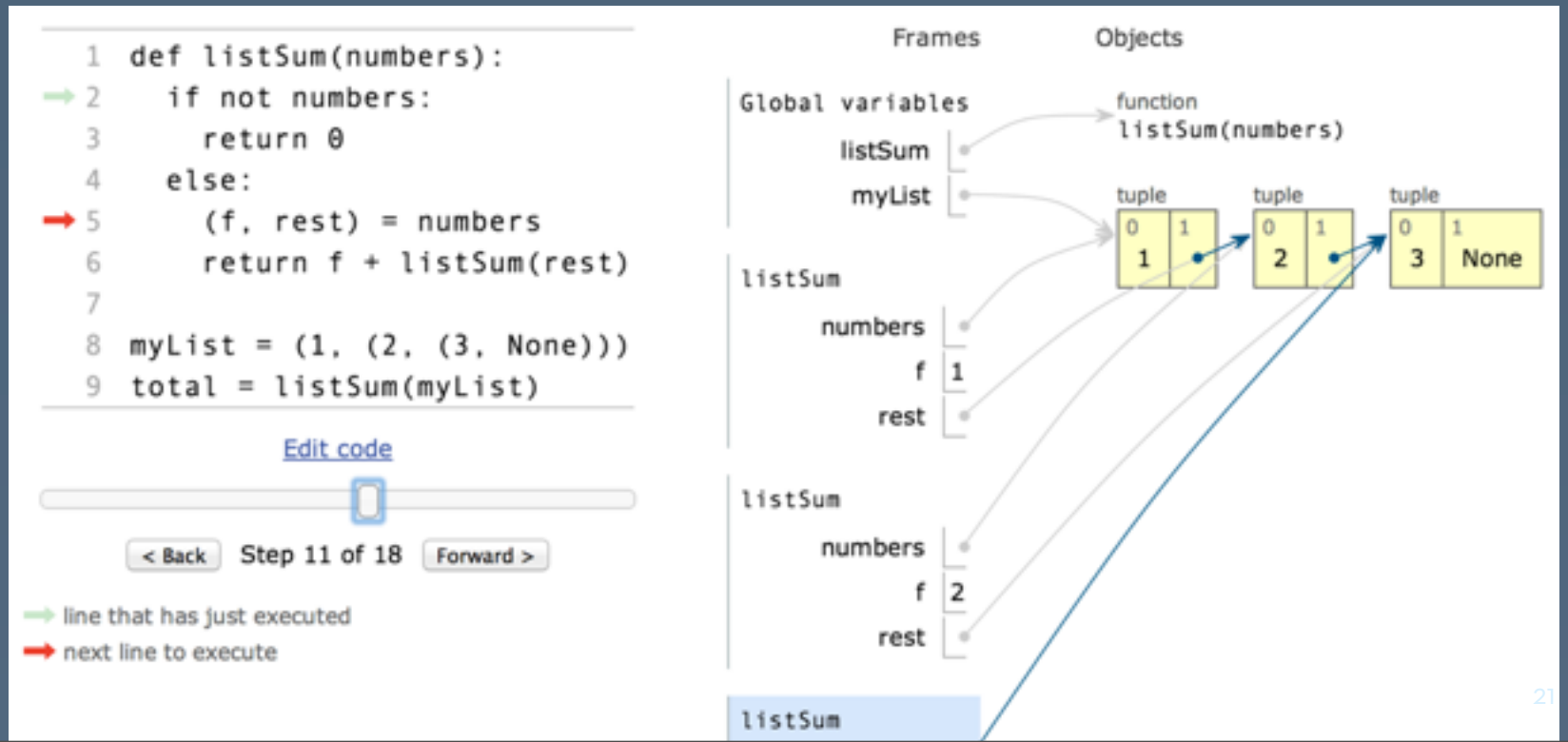
[Resnick et al., CACM '09]



Online python tutor

[Guo, SIGCSE '13]

- Embeddable Python data structure visualization
- Over 200,000 users and a dozen universities making use of the system



Programming by demonstration

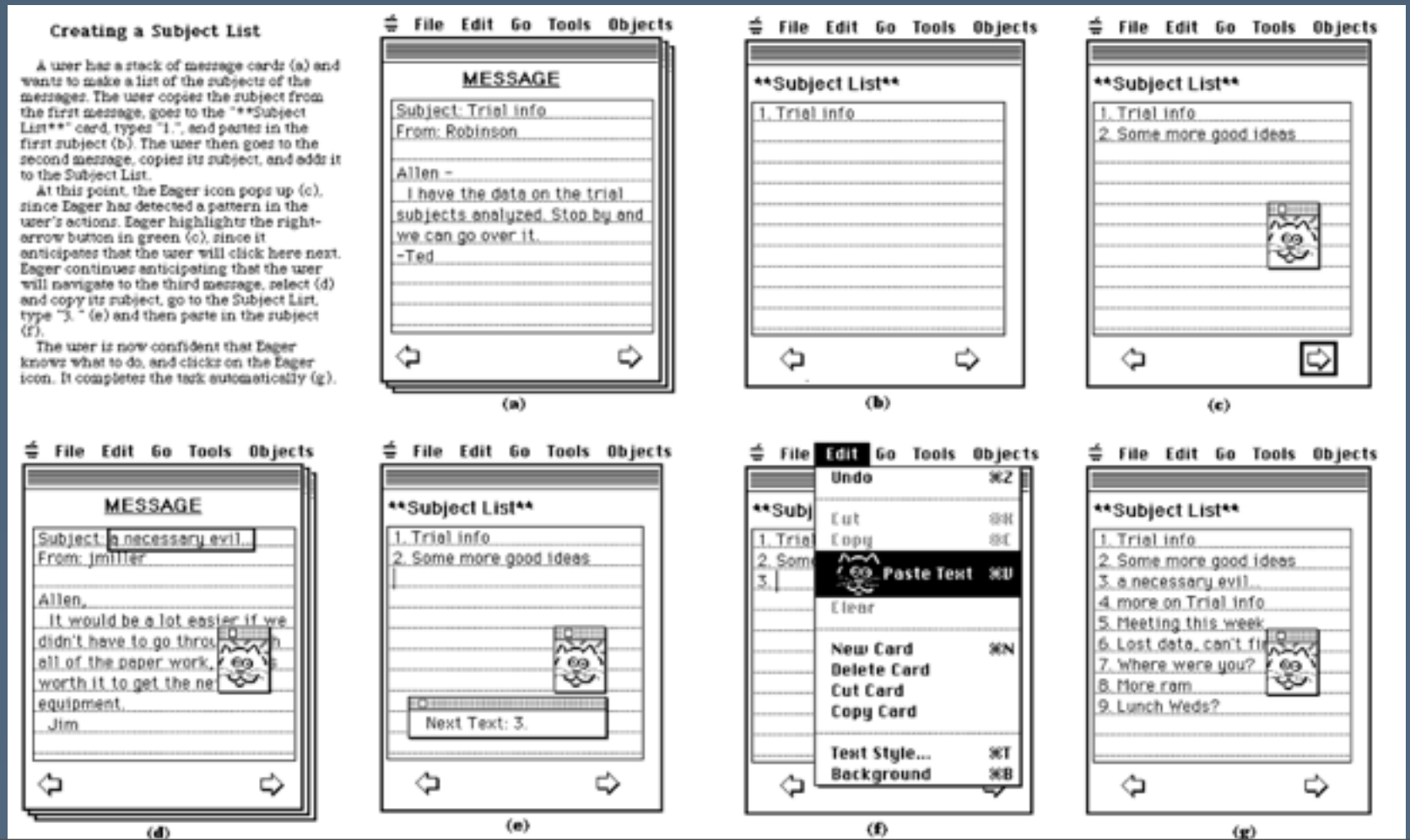
Goals of PBD

- Teach a computer to program simply by demonstrating what should be done
- Challenges
  - There is an infinite, and hugely branching, space of programs that might be inferred
  - Inferred macros can be extremely brittle

Recall: EAGER

[Cypher, CHI '91]

- Infer a macro by watching the user's behavior



Simultaneous structured editing

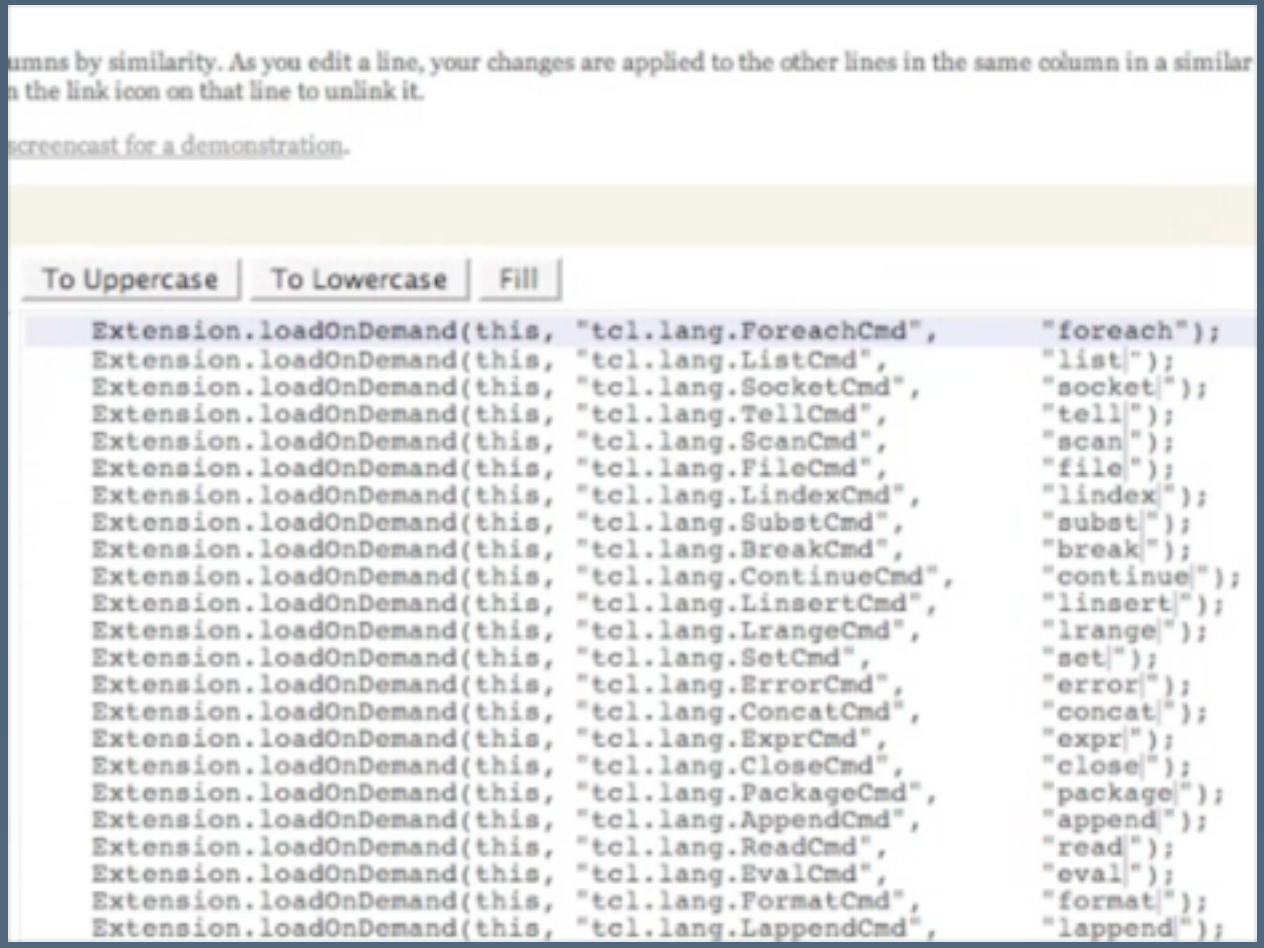
[Miller and Myers, USENIX '01]

- Utilize lightweight structure in text
- Today, versions of this exist in Sublime Text

Simultaneous structured editing

[Miller and Myers, USENIX '01]

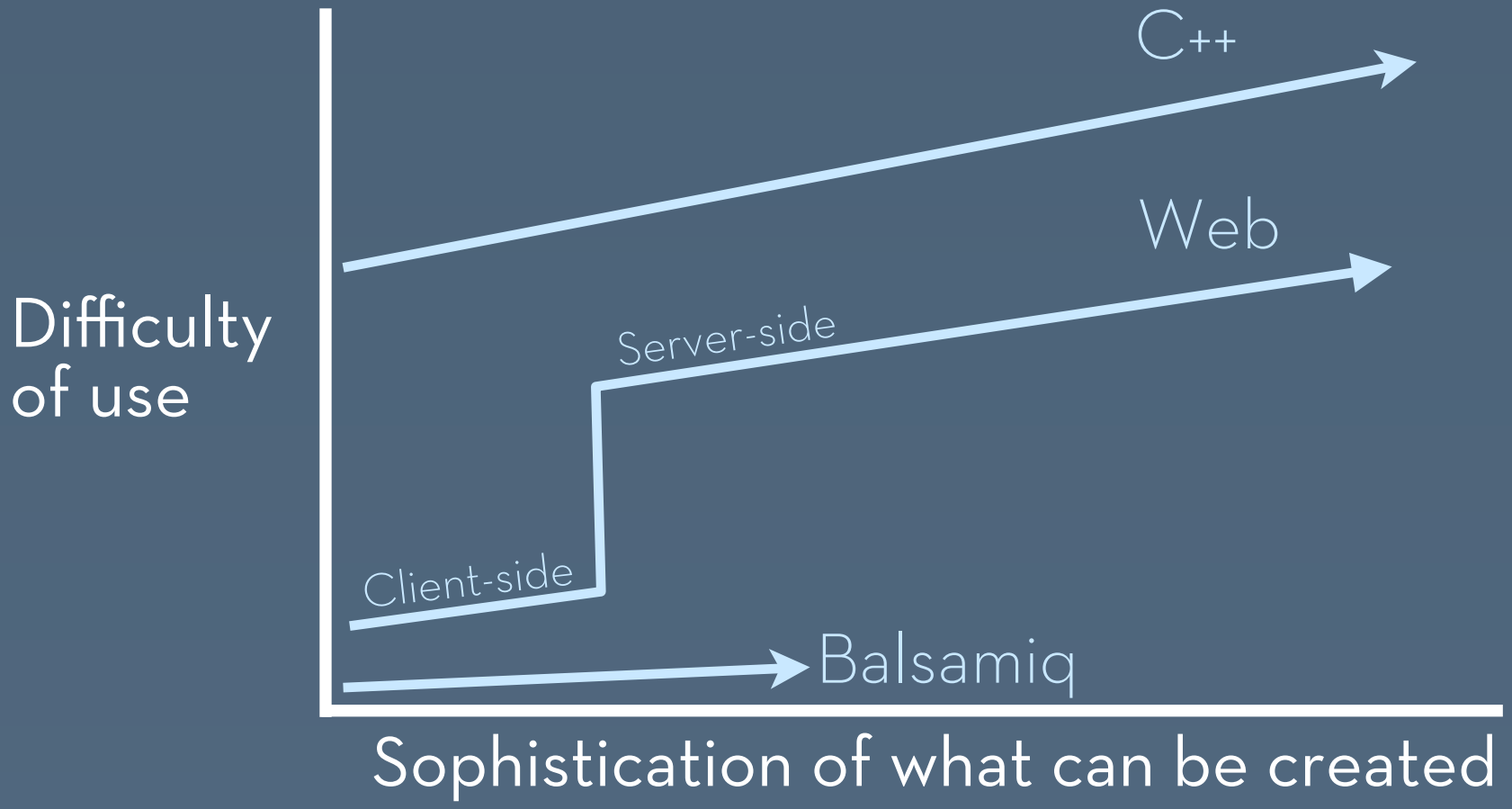
- Utilize lightweight structure in text
- Today, versions of this exist in Sublime Text



Toolkits

Recall: Threshold/Ceiling

[Myers, Hudson and Pausch, TOCHI 2000]



Research agenda: toolkits

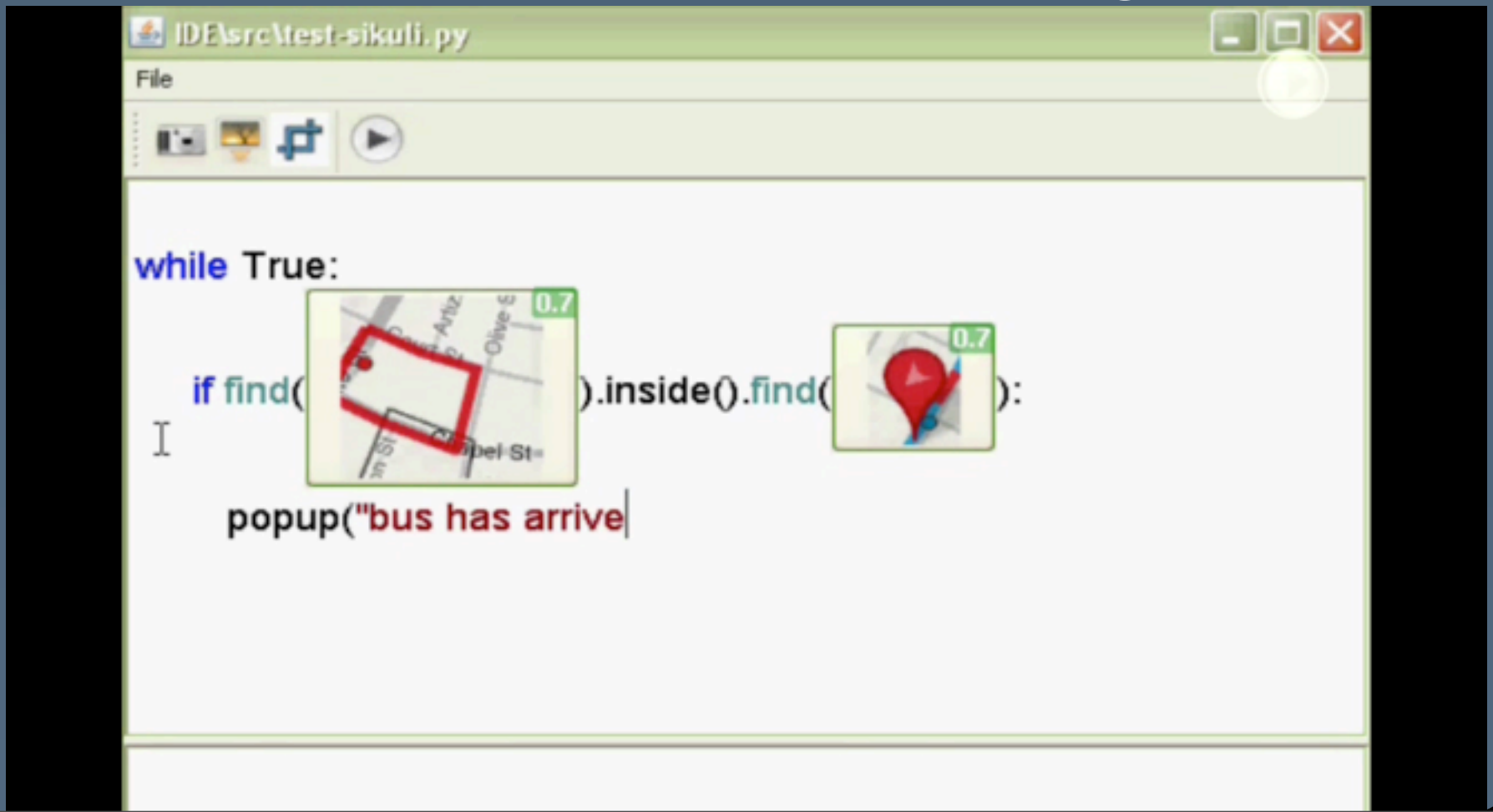
- Crystallize and formalize a perspective on a difficult engineering problem
- If successful, shift the entire programming practice for the area

Sikuli: programming with screenshots [Yeh, Chang, and Miller, UIST '09]

- Template search in desktop scripting

Sikuli: programming with screenshots [Yeh, Chang, and Miller, UIST '09]

- Template search in desktop scripting



Recall: Chickenfoot

[Bolin et al., UIST '05]

- Client-side scripting, automation and customization of rendered web pages

```
isbn = find('number just after isbn')
with (fetch('libraries.mit.edu')) {
   pick('Keywords');
   enter(isbn)
   click('Search')
   link=find('link just after Location')
}
// back to Amazon
if (link.hasMatch) {
   insert(before('first rule after "Buying"'),
   link.html)
}
```

Research agenda: HCI and programming

- Understand the challenges in programming
- Design more effective software engineering interfaces
- Aid novices in learning to program or writing programs
- Abstract best practices into toolkits

Final Presentations

- Wednesday, June 12, 8:30am - 11:30am
- Wallenberg 124
- 4 minutes to present, 2 minutes for questions
  - Next group plugs in laptop during questions

- Make sure to test your slides in the room before presentation day
- Presentation order TBD