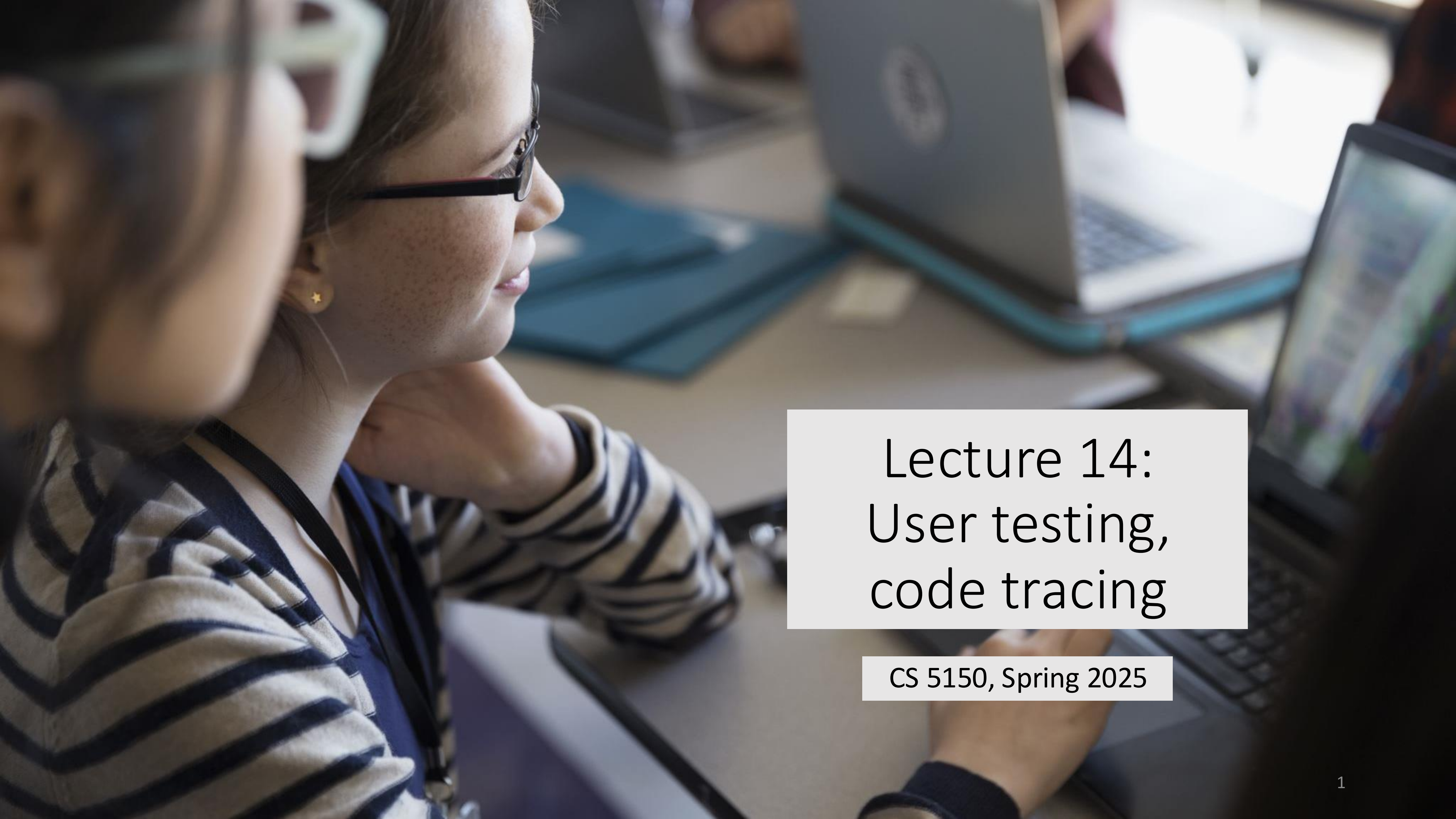


Lecture 14: User testing, code tracing

CS 5150, Spring 2025

Administrative Reminders

- Schedule midpoint presentation by Mar 25
  - Team w External Clients only: Contact Kimberly Budd (kj37@cornell.edu) to reserve a meeting room for midpoint presentation
  - Only the team leader should contact. Send Team ID, Date/time, and #attendees. CC me.
- Assignment A3 due Mar 26
- Report #3 due Mar 25
- Peer Eval: Submit today (if not done yet)
- In-class exam: Mar 27 (Mix of short answer qs, few descriptive ones, example format coming soon)

Presentation Rubrics (Midpoint): 50 points

- Content (28 points): 4 points each
  - A description of what the team has agreed to deliver to the client is included.
  - Team summarizes progress made so far this semester and discusses any unexpected events and risks.
  - An overview of the remaining plan to complete and deliver the project is provided.
  - Team conducts a demonstration of an operational prototype or delivered features.
  - Technical terms are well-defined in language appropriate for the target audience.
  - Material included is accurate and relevant to the overall purpose or goal of the presentation.
  - Team was able to answer audience questions with reasonable clarity.

Presentation Rubrics (Midpoint): 50 points

- Organization (9 points): 3 points each
  - Information is presented in a logical sequence.
  - A roadmap is provided at the beginning of the presentation that guides the audience.
  - There are clear closing remarks that provide a summary for the audience.

Presentation Rubrics (Midpoint): 50 points

- Presentation (13 points):
    - Speakers maintain good eye contact with the audience and are appropriately animated.
    - Speakers use a clear, audible voice.
    - Delivery is poised, controlled, and follows presentation etiquette.
    - Appropriate professional language is used during the presentation.
    - Visual aids and demonstrations are well-prepared, informative, and not distracting.
    - Information is communicated well.
    - Length of presentation is within the assigned time limits.

Lecture goals

- Give effective presentations to stakeholders
- Solicit feedback on user interface designs

User experience

Previously in 5150 ….

Terminology

- User Interface (UI)
    - Look and behavior of system’s controls
- User experience (UX)
    - All factors that contribute to usability of computing system
    - Encompass entire usage lifecycle, from discovery to accomplishing goals
    - Focus on user satisfaction
- Human-Computer Interaction (HCI)
    - Academic discipline studying how people interact with computers
    - Many courses and research programs in Information Science and Communications departments

Usability requirements and evaluation tools

	Initial	Mock-up	Prototype	Production
Client’s opinions	✓	✓	✓	
Competitive analysis	✓			
Expert opinion	✓	✓	✓	
Focus groups	✓	✓		
Observing users		✓	✓	✓
Measurements			✓	✓

Focus group

- Group interview – helps generate ideas that would not have occurred individually
- Participants: 5-12 potential users with similar viewpoints
- Interviewer
  - Ask a structured set of questions
  - Encourage group discussions
  - May show mock-ups
  - Summarize conclusions
- Recorder takes notes
- Repeat with contrasting user groups

Internal project users

- Internal projects are code review tools; users are software developers
- Your classmates are candidate users!
  - Not including your team members
  - Not including teams working on the same feature
- Recruit classmates for focus groups, user testing
  - Can coordinate on Ed Discussion
  - May be easiest to pair teams
- Documentation of user studies will be expected in future report

Accessibility

- Users have varying ability to interact with computer interfaces
  - Color blindness (1/12 men, 1/200 women)
  - Poor or no vision
  - Lack of hearing
  - Poor manual dexterity
  - Limited language skills, domain vocabulary
  - Sensitivity to flashing light, motion sickness

- Accessibility requirements constrain the user interface
  - Many systems have a legal requirement to support users with disabilities
  - Example: Compliance with Section 508 of US Rehabilitation Act https://www.section508.gov/
- Some technologies may not be suitable
  - Examples: Flash, untagged PDF, immature widget toolkits

Equipment requirements

- Software runs on wide variety of devices, with diverse configurations, in many environments
    - Screen size
    - Graphics performance
    - Network bandwidth, latency, stability
    - Peripheral hot plugging

- Be explicit about equipment assumptions/requirements
- Be explicit about failure handling
- Test on variety of equipment (including extremes)

- Example: Chat application

Dark patterns

- Many of our experiences with UI are in a marketing context
    - Goal is to maximize engagement and manipulate user decisions
    - Being commonplace and effective in marketing goals does not make a design pattern good
        - Avoid simply aping features of slick websites (even if libraries make it easy to do so)
- User-centric design
    - Interface should facilitate, not redirect, users’ objectives

- https://cacm.acm.org/magazines/2020/9/246937-dark-patterns/fulltext

Models

Relating user and system models

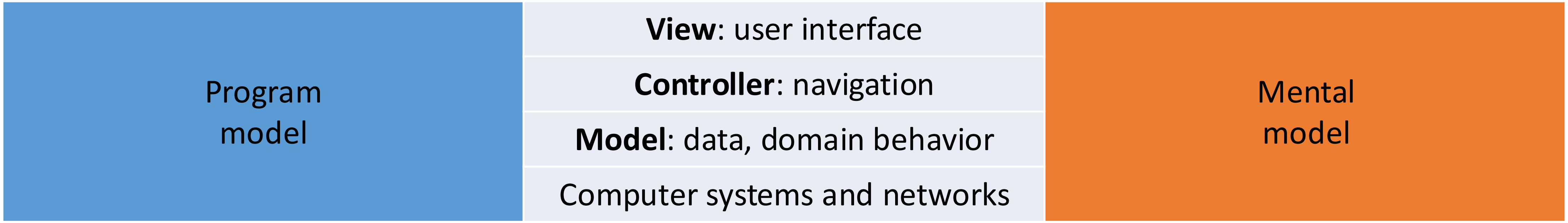
Mental model

- User's view of system and the UX it provides
- May include physical metaphors for digital interactions
- Examples:
  - Pieces on a game board
  - File folders and desks

Program model

- Data, relationships, and functions making up the system
- Examples:
  - Object identity & coordinates, rules constraining movement
  - Tree of data units with metadata

Model-view-controller (as a "model")



Program model

View: user interface

Controller: navigation

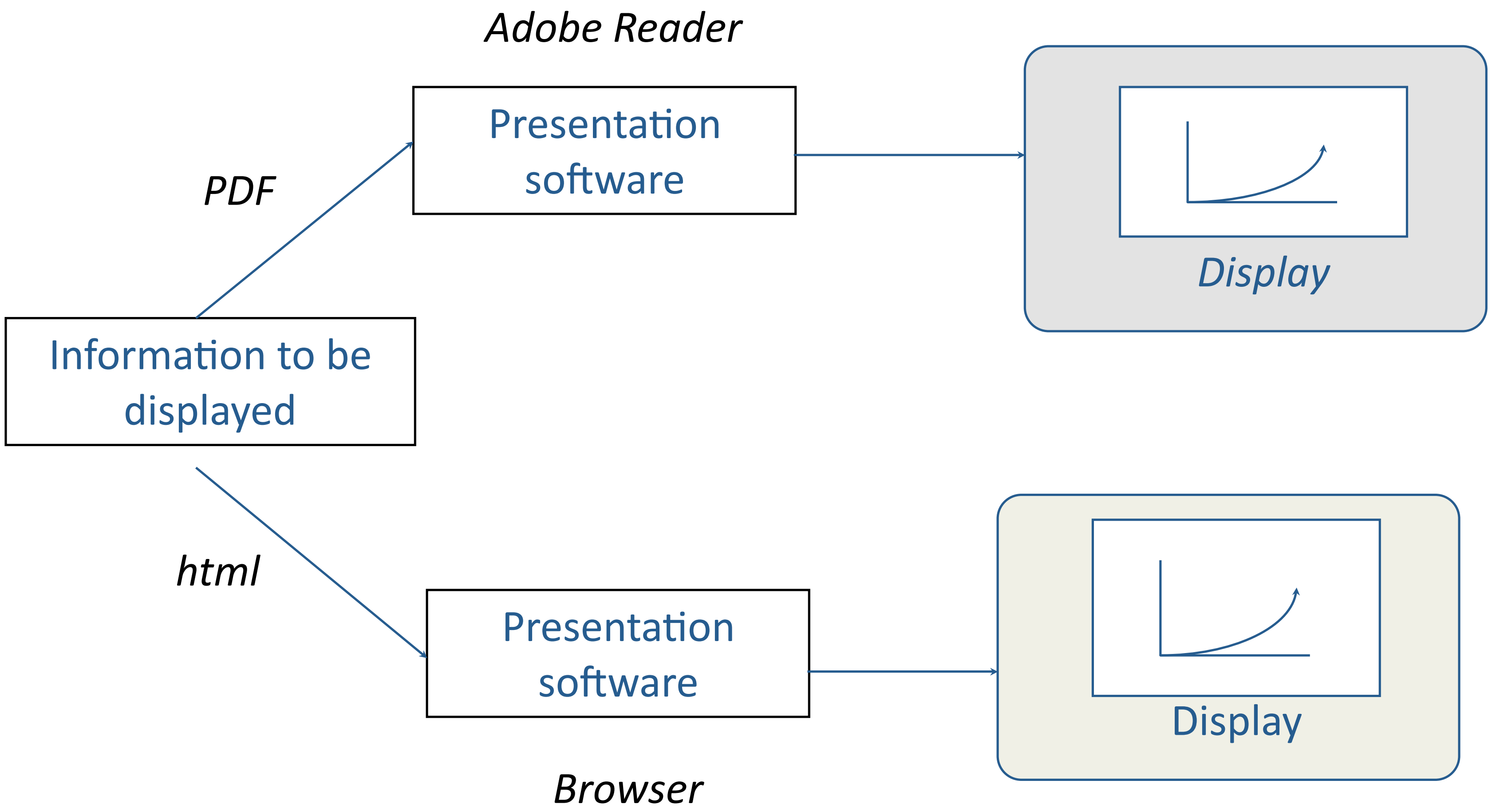
Model: data, domain behavior

Computer systems and networks

Mental model

Layers correspond with most users' mental models of computing systems.

Separation of content from view



Adobe Reader

PDF

Presentation software

Display

Information to be displayed

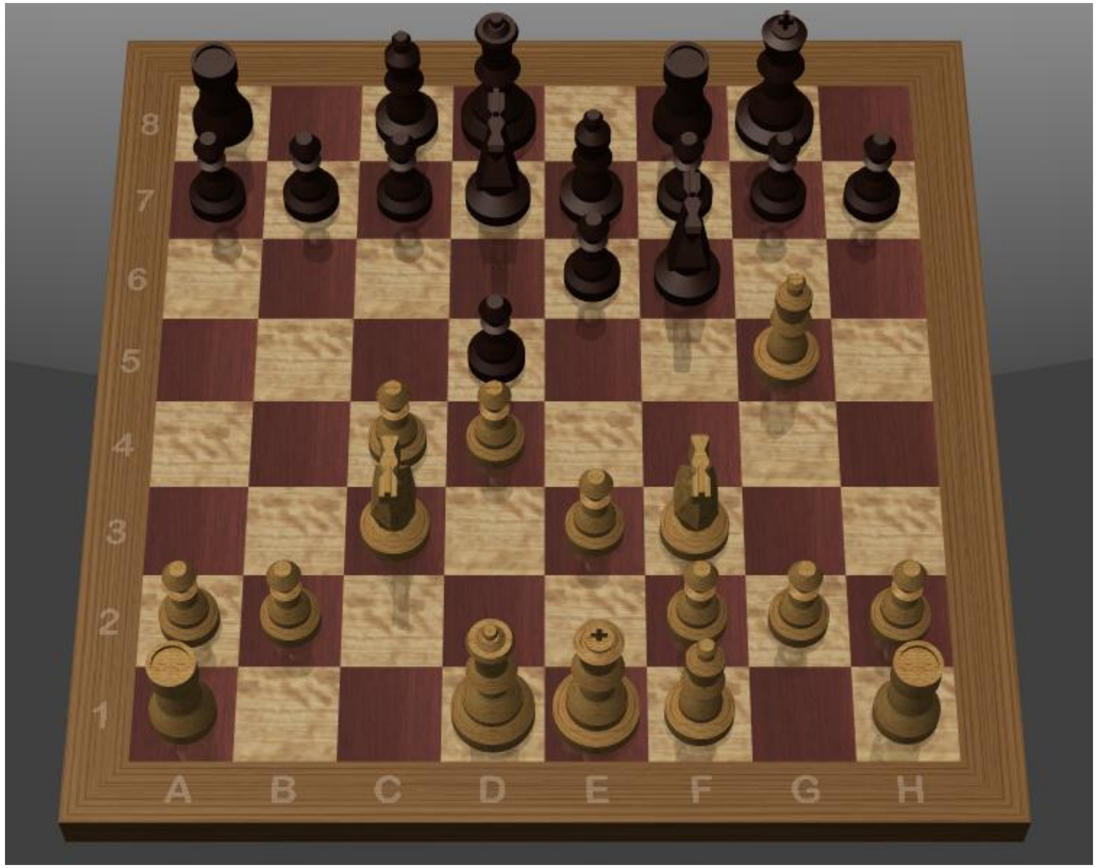
html

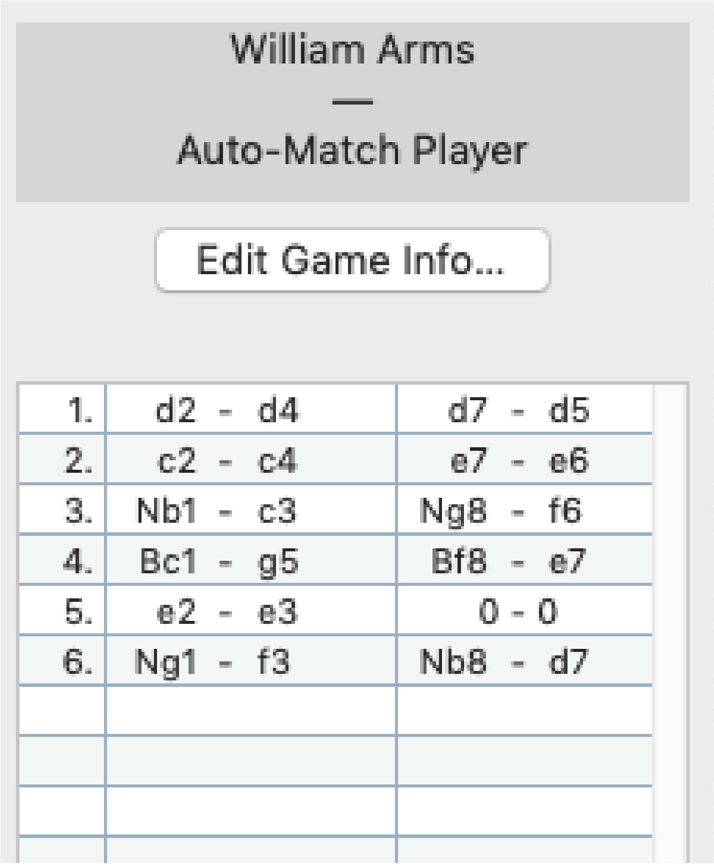
Presentation software

Display

Browser

Separation of content from view




William Arms

—

Auto-Match Player

1.	d2 - d4	d7 - d5
2.	c2 - c4	e7 - e6
3.	Nb1 - c3	Ng8 - f6
4.	Bc1 - g5	Bf8 - e7
5.	e2 - e3	0 - 0
6.	Ng1 - f3	Nb8 - d7

Design principles

UI design principles

- UI design is partly an art, but some general principles apply:
  - Consistency (in appearance, interaction, function)
  - Feedback (what is the system doing? why does the user see what they do? what is about to happen?)
  - Ability to interrupt or reverse actions
  - Comprehensible and non-destructive error handling
- The user should feel in control (not like they're being controlled)

Design choices: text vs. graphics

Text

- Precise, unambiguous (hopefully)
- Fast to compute, transmit

Graphics

- Quick to comprehend, learn
  - But icons may be difficult to recognize
- Variations can show different cases

Command line interfaces

- Limitations of GUIs
  - Only suitable for human users (difficult to automate)
  - Awkward to control complex interactions (difficult to compose)

Internal projects

- Gerrit: Use Git CLI to create, update reviews
- Review Board: Use `rbt` CLI to create, update reviews

- Command line interfaces (CLI)
  - User interacts with system by typing commands
  - Composable
  - Scriptable
  - Can be adapted for users with disabilities
  - Amenable to formal specification
  - Usually requires learning or training

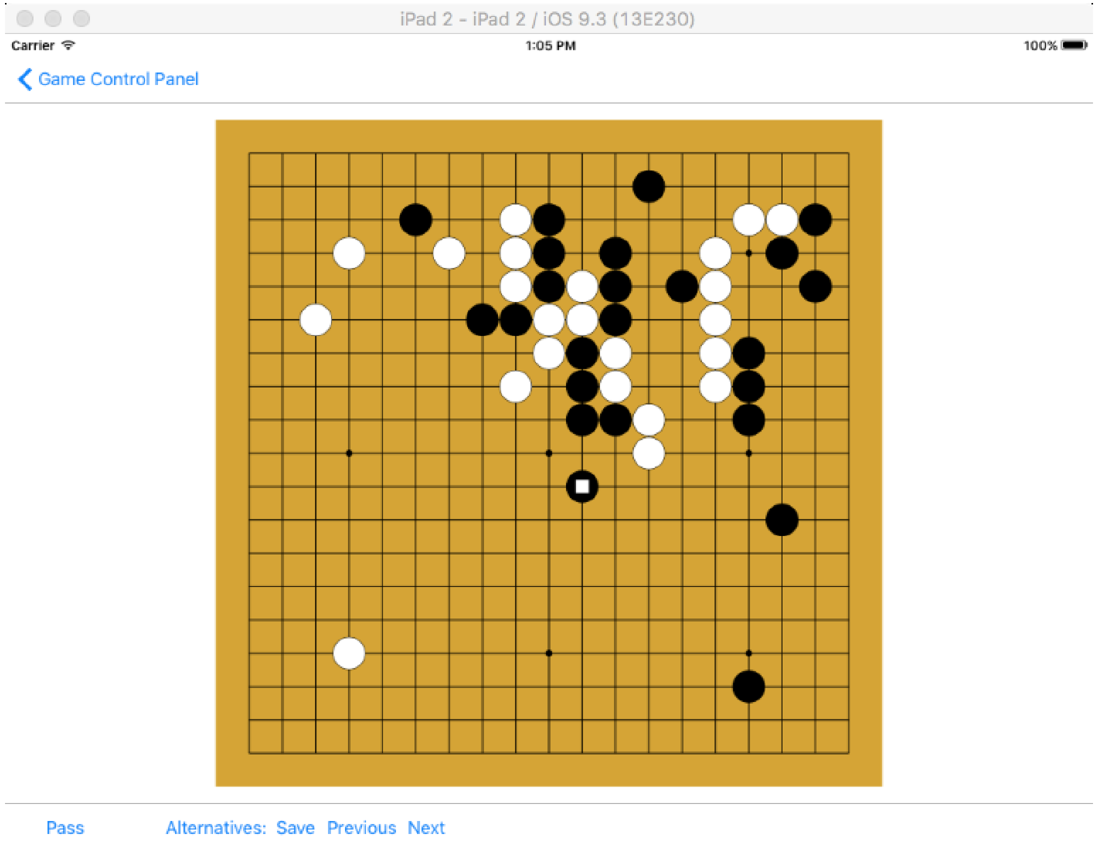
Web and mobile interfaces

Web and mobile apps

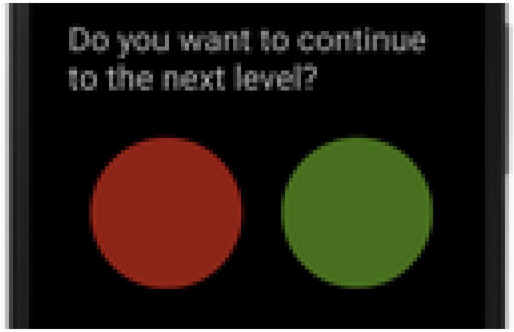
- Must consider network
  - Transfers may need to be asynchronous to hide latency
    - Need visual feedback that operation is in progress
    - Should support cancellation
  - Connections may be unreliable
    - Should be robust to duplication

Leverage simulation

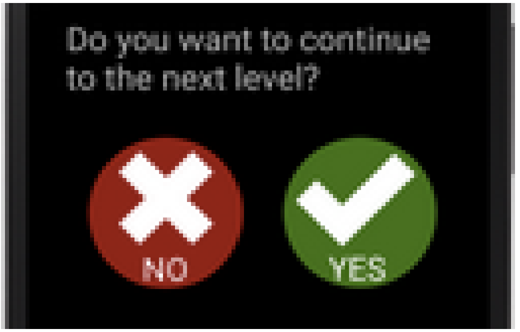
- App development environments (e.g. Xcode, Android Studio) allow you to simulate screen sizes, touch events
- Web browser developer tools allow you to simulate screen sizes, network speed


iPad 2 - iPad 2 / iOS 9.3 (13E230)
Carrier
1:05 PM
100%
Game Control Panel
Pass
Alternatives: Save Previous Next

Test for accessibility



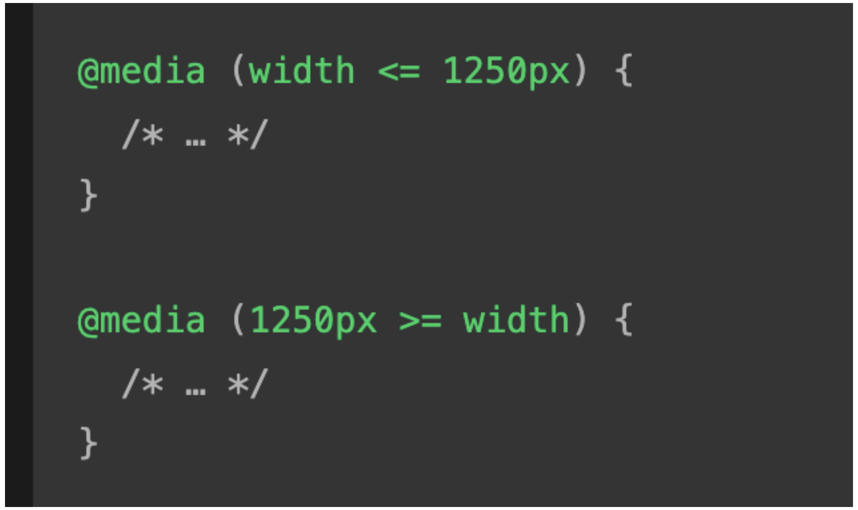
Do you want to continue to the next level?



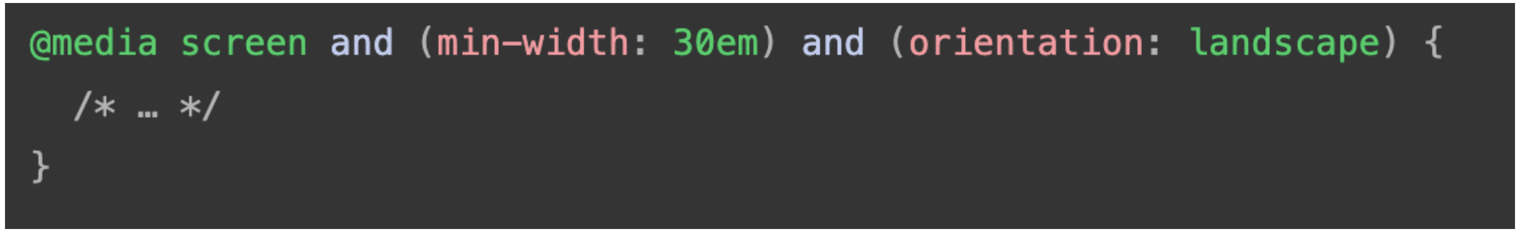
Do you want to continue to the next level?

Responsive design

- Automatically adjust user interface based on size of screen (or other device properties)
  - Beyond simple layout scaling – can completely change layout to accommodate device
  - Use CSS media queries to select different style rules in different situations



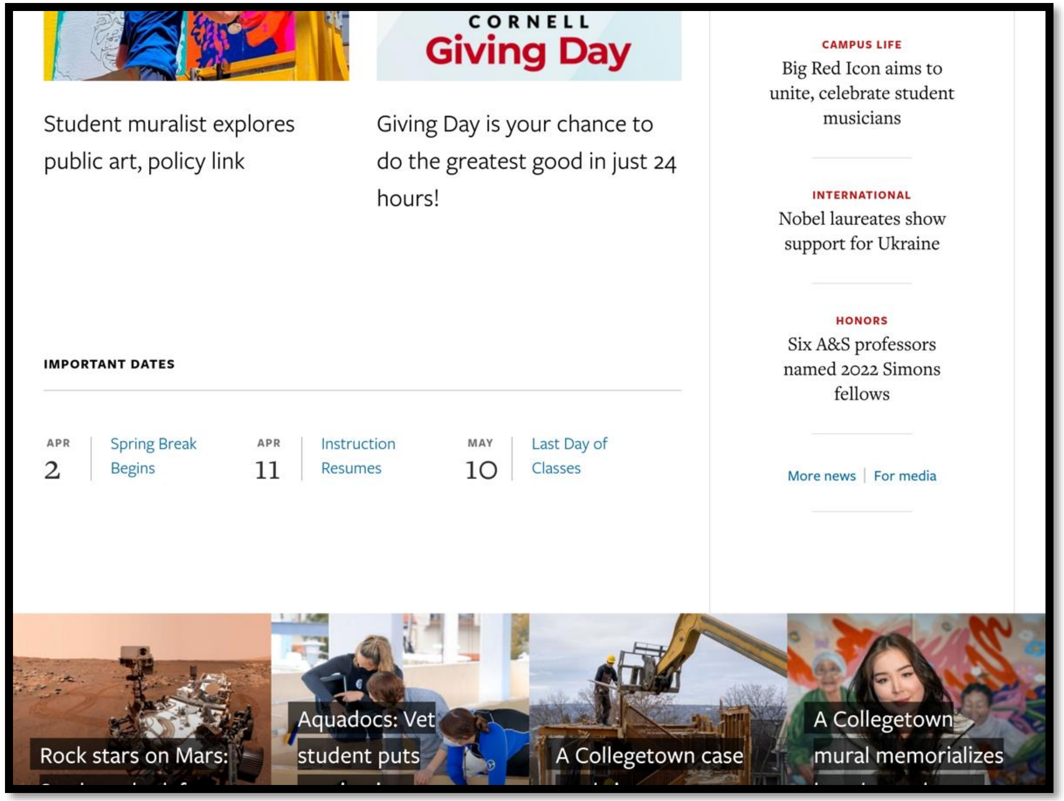
```
@media (width <= 1250px) {
  /* … */
}

@media (1250px >= width) {
  /* … */
}
```



```
@media screen and (min-width: 30em) and (orientation: landscape) {
  /* … */
}
```

Responsive design

Tablet



CORNELL Giving Day

Student muralist explores public art, policy link

Giving Day is your chance to do the greatest good in just 24 hours!

CAMPUS LIFE
Big Red Icon aims to unite, celebrate student musicians

INTERNATIONAL
Nobel laureates show support for Ukraine

HONORS
Six A&S professors named 2022 Simons fellows

More news | For media

IMPORTANT DATES

APR 2 | Spring Break Begins
APR 11 | Instruction Resumes
MAY 10 | Last Day of Classes

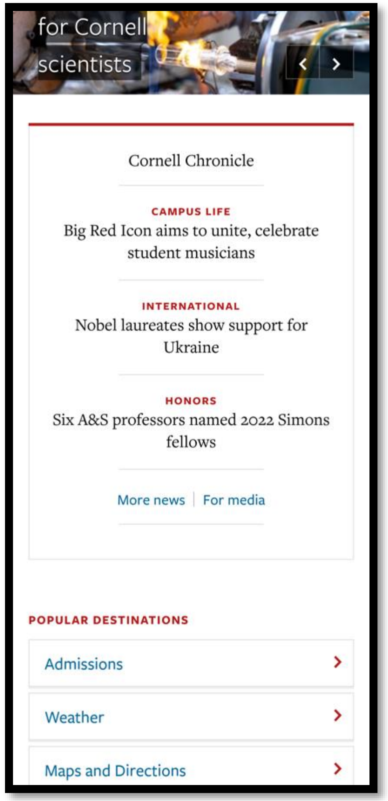
Rock stars on Mars:

Aquadocs: Vet student puts

A Collegetown case

A Collegetown mural memorializes

Smartphone



for Cornell scientists

‹ ›

Cornell Chronicle

CAMPUS LIFE
Big Red Icon aims to unite, celebrate student musicians

INTERNATIONAL
Nobel laureates show support for Ukraine

HONORS
Six A&S professors named 2022 Simons fellows

More news | For media

POPULAR DESTINATIONS

Admissions ›

Weather ›

Maps and Directions ›

Flexible grids

- Divide screen into columns
- Declare how many columns each element occupies at each breakpoint
  - Use more columns for narrower screens
- Example: Bootstrap
- Demo

Aside: semantic markup

- Many attempts to make content, style separate concerns
  - HTML+CSS, LaTeX, DocBook XML, Content Management Systems
  - Allows content to be delivered in multiple media (web, print, ebooks)
- Tension with designing around content
  - Separating tightly-coupled info is more work, hard to maintain
  - Style rules tend to leak into content

Progressive enhancement

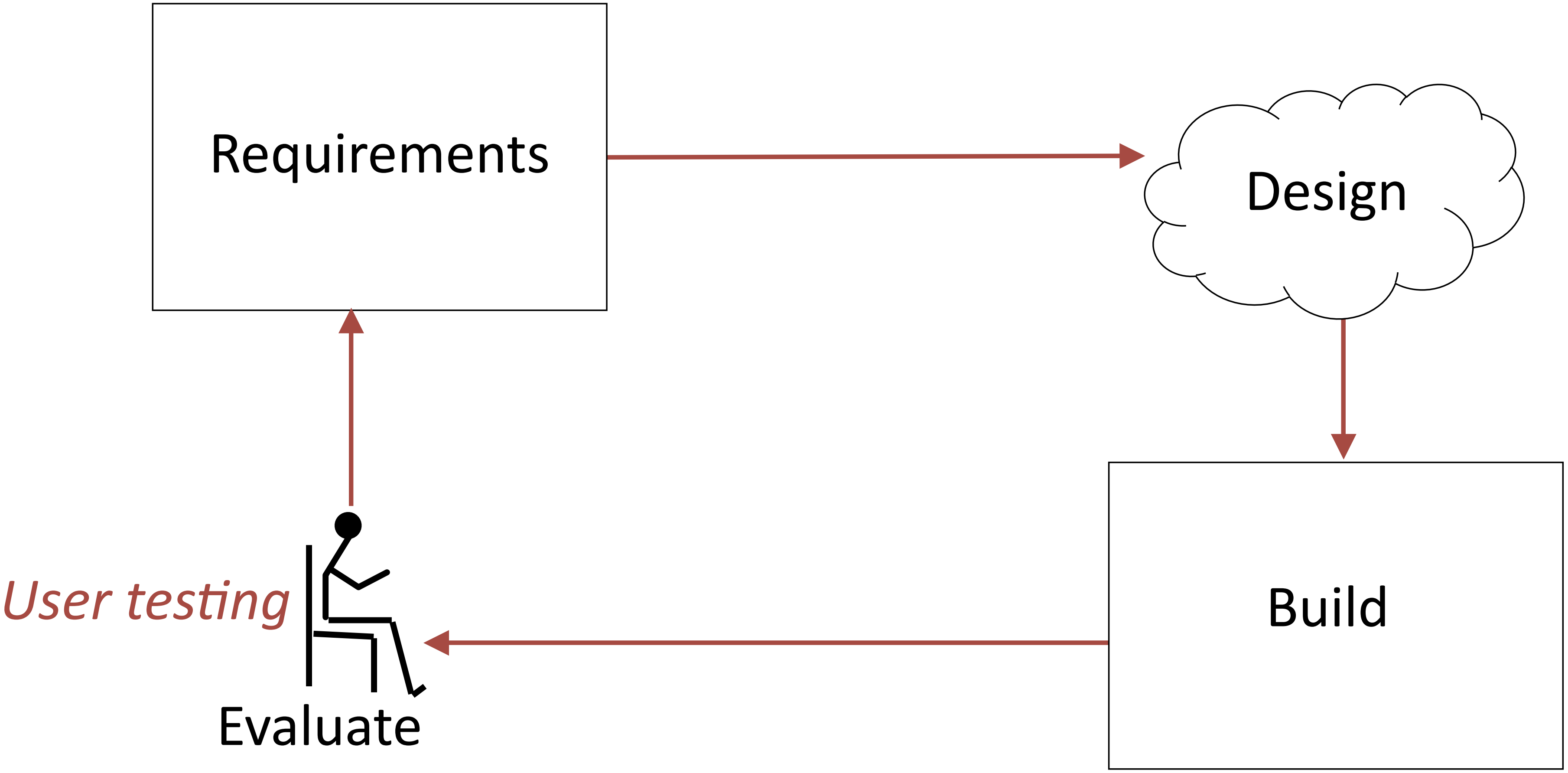
- Beware the fancy
  - Modern browsers are "evergreen" - they keep themselves up-to-date and support many of the latest web standards
  - But compatibility is still a concern
    - Support for standards is uneven (e.g., Edge vs. CMSX)
    - Mobile devices often stop receiving updates
    - User preferences, browser extensions, firewalls make browsers heterogeneous
- Progressive enhancement
  - Leverage fancier features to improve UX, but ensure that core functions are still available without them
  - Use fallbacks, polyfills to maximize compatibility

Poll: Progressive enhancement

PollEv.com/cs5150sp25

Evaluation and user testing

Analyze/design/build/evaluate loop



Evaluation

- Design and evaluation should be done by different people
- Schedule must include time to conduct tests *and* make changes
- Evaluation should be ongoing
  - Iterative refinements during development
  - Quality assurance before deployment
  - Improvements after launch
- Methods of evaluation
  - Empirical (user testing)
  - Quantitative (measurements on operational systems)
  - Analytical (sans users; not in CS 5150)

Standards for usability: ISO 9241:11

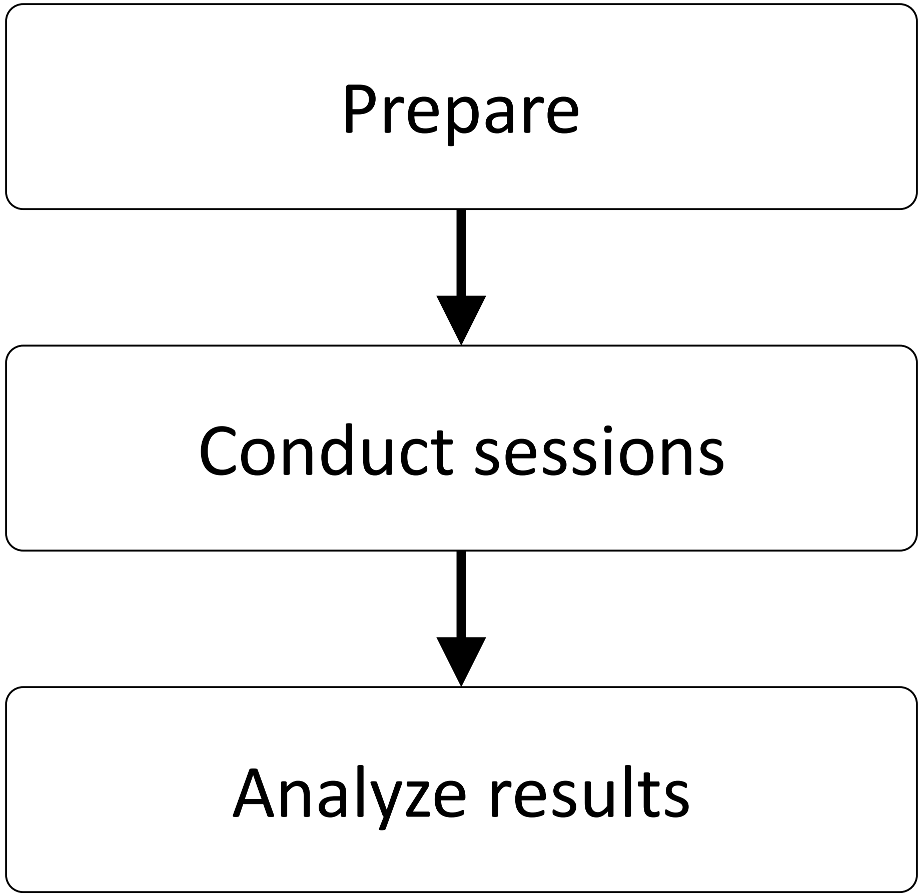
- Effectiveness
  - The accuracy and completeness with which users achieve certain goals
  - Measures: quality of solution, error rates
- Efficiency
  - The relationship between the effectiveness and the resources expended in achieving them
  - Measures: task completion time, learning time, number of clicks
- Satisfaction
  - The users' comfort with and positive attitudes towards the use of the system
  - Measures: attitude rating scales

https://www.iso.org/standard/63500.html

Poll: Measuring usability

PollEv.com/cs5150sp25

User testing stages



- User testing is time-consuming, expensive, and *critical*

Preparation

- Determine goals of usability testing
  - *"Can a user find the required information in no more than two minutes?"*

- Write the user tasks
  - *"Given a new customer application form, add a new customer to the customer database"*

- Recruit participants
  - Use the descriptions of users from the requirements phase to determine categories of potential users and user tasks

Participants

- Don't need many (per feature)
  - Diminishing returns after 5-6 users
  - Look for diversity (age, experience, ability)
- Combine structured tests with free-form interviews
- Have at least two evaluators per test
  - Should *not* include designers

- Advice: it's not a race!
  - Example: user testing for arXiv

Conducting sessions

- Environment
  - Informal
  - Simulated work environment
  - Usability lab
- Give the user their task
- Observe the user
  - Human observer(s)
  - Recording (with permission)
- Query satisfaction

Analyzing results

- Test the system, not the users
  - Respect the data and the user's responses
  - Do not make excuses for designs that failed
  - If possible, use statistical summaries
- Pay close attention to instances where users:
  - Were frustrated
  - Took a long time
  - Could not complete tasks

- Also note aspects of the design that *did* work
  - Ensures they are maintained / do not regress in final product

Example: Past CS 5150 methodology

How we're user testing:

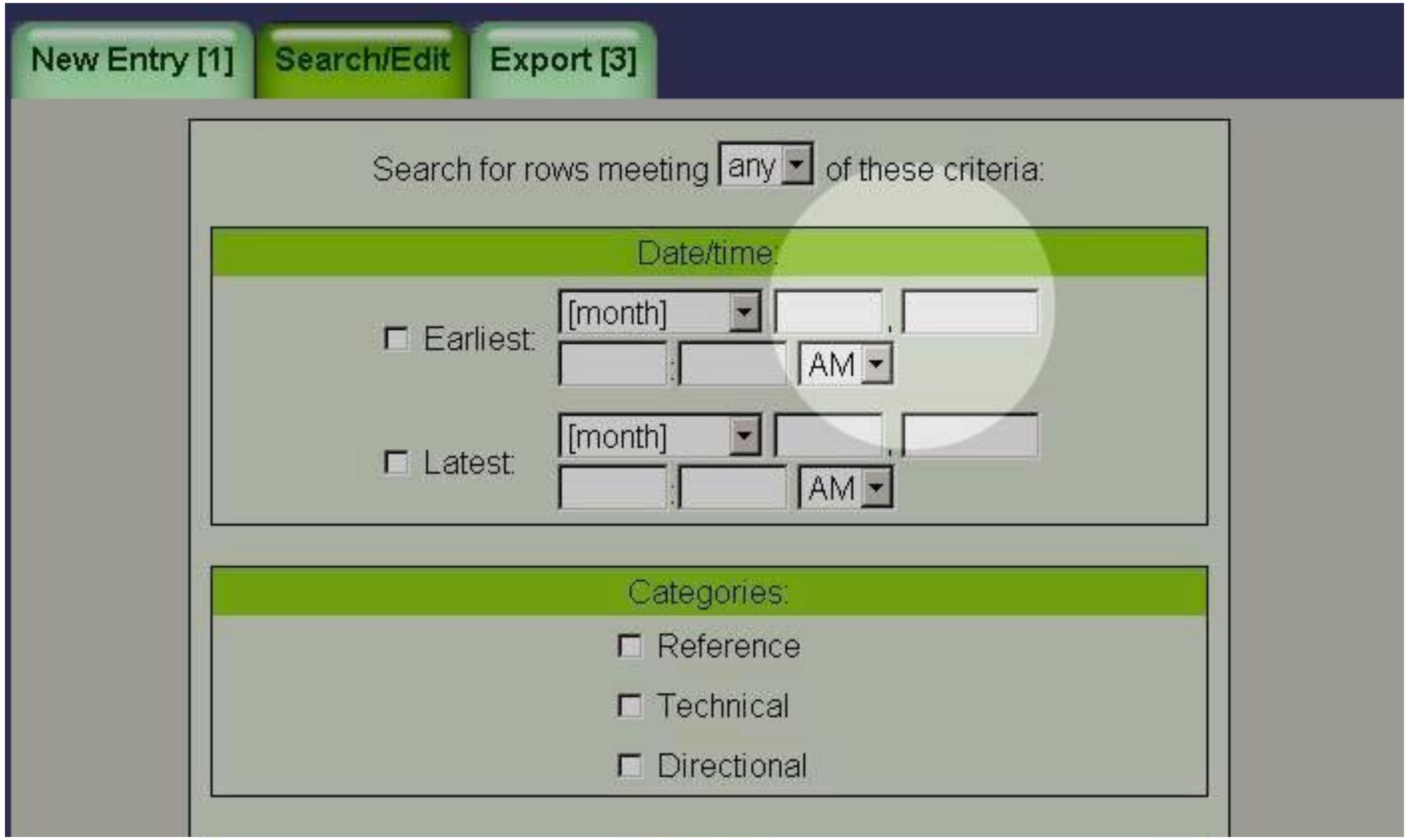
- One-on-one, 30-45 min user tests with staff levels
- Specific tasks to complete
- No prior demonstration or training
- Pre-planned questions designed to stimulate feedback
- Emphasis on testing system, not the stakeholder!
- Standardized tasks / questions among all testers

Example

Types of questions we asked:

- Which labels, keywords were confusing?
- What was the hardest task?
- What did you like, that should not be changed?
- If you were us, what would you change?
- How does this system compare to your paper based system
- How useful do you find the new report layout? (admin)
- Do you have any other comments or questions about the system? (open ended)

What we've found:
Issue #1, Search Form Confusion!



Search for rows meeting any of these criteria:

Date/time:

□ Earliest: [month] , : AM

□ Latest: [month] , : AM

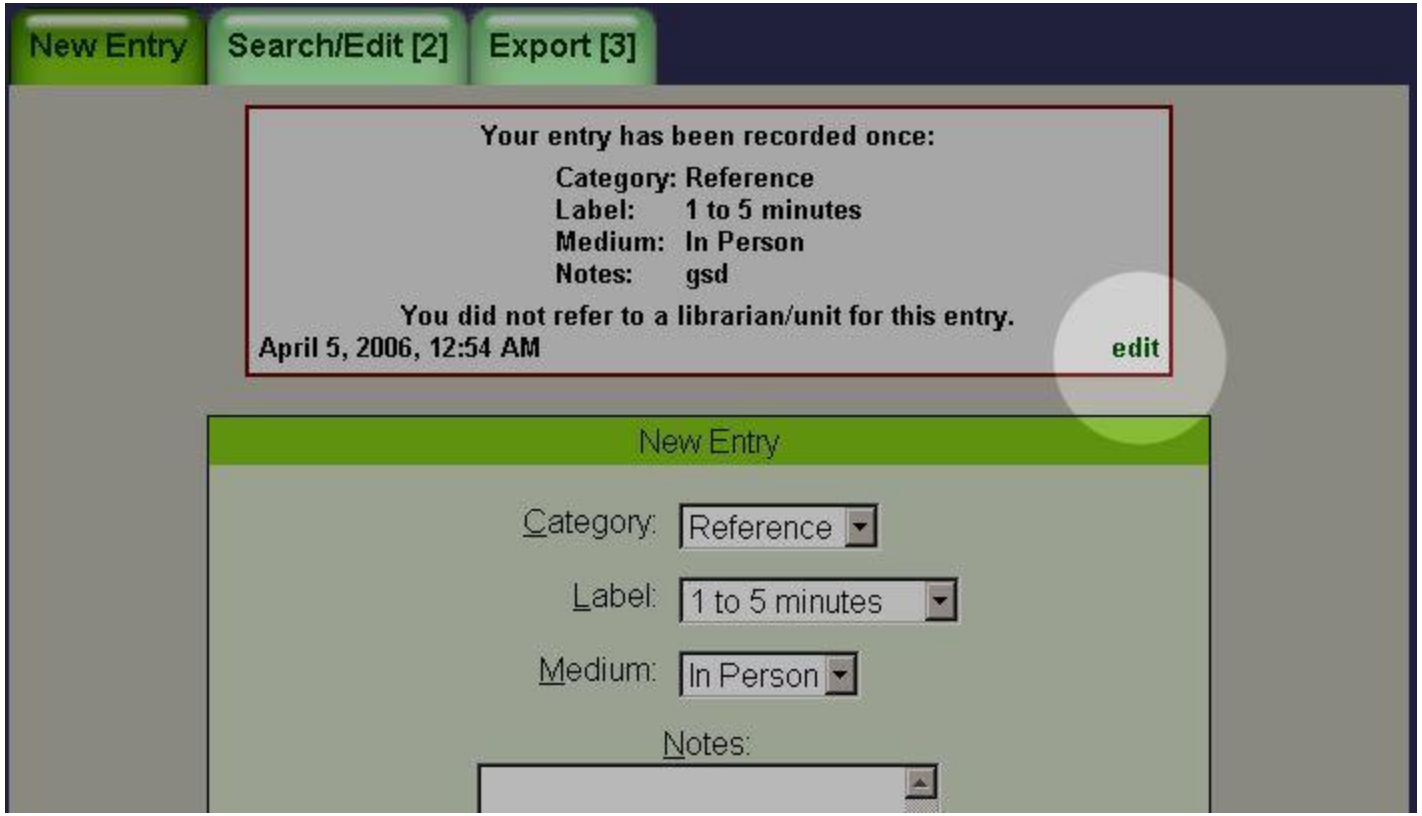
Categories:

□ Reference

□ Technical

□ Directional

What we've found:
Issue #2, Inconspicuous Edit/Confirmations!



Your entry has been recorded once:

Category: Reference
Label: 1 to 5 minutes
Medium: In Person
Notes: gsd

You did not refer to a librarian/unit for this entry.

April 5, 2006, 12:54 AM

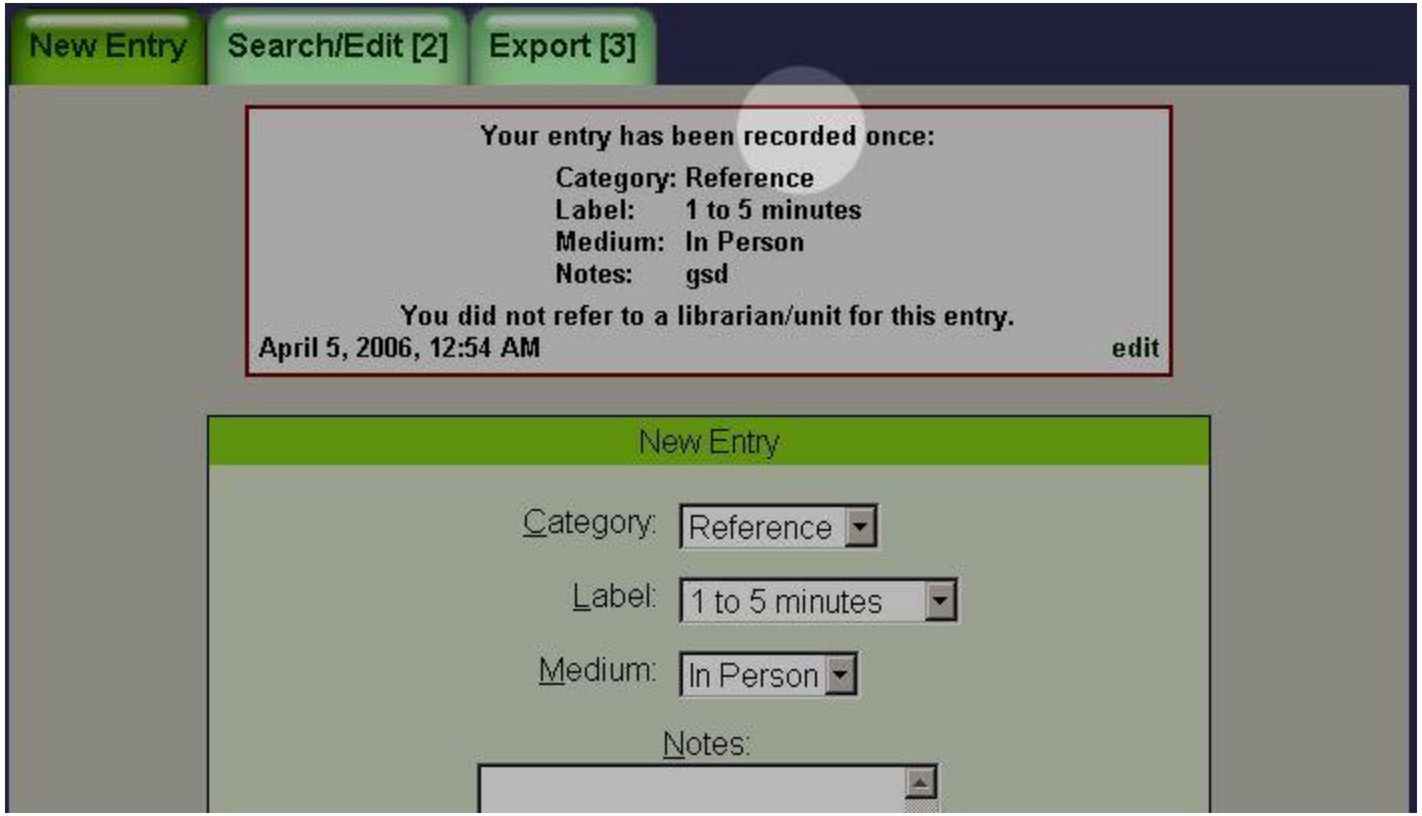
Category: Reference

Label: 1 to 5 minutes

Medium: In Person

Notes:

What we've found:
Issue #3, Confirmation Terms



Your entry has been recorded once:

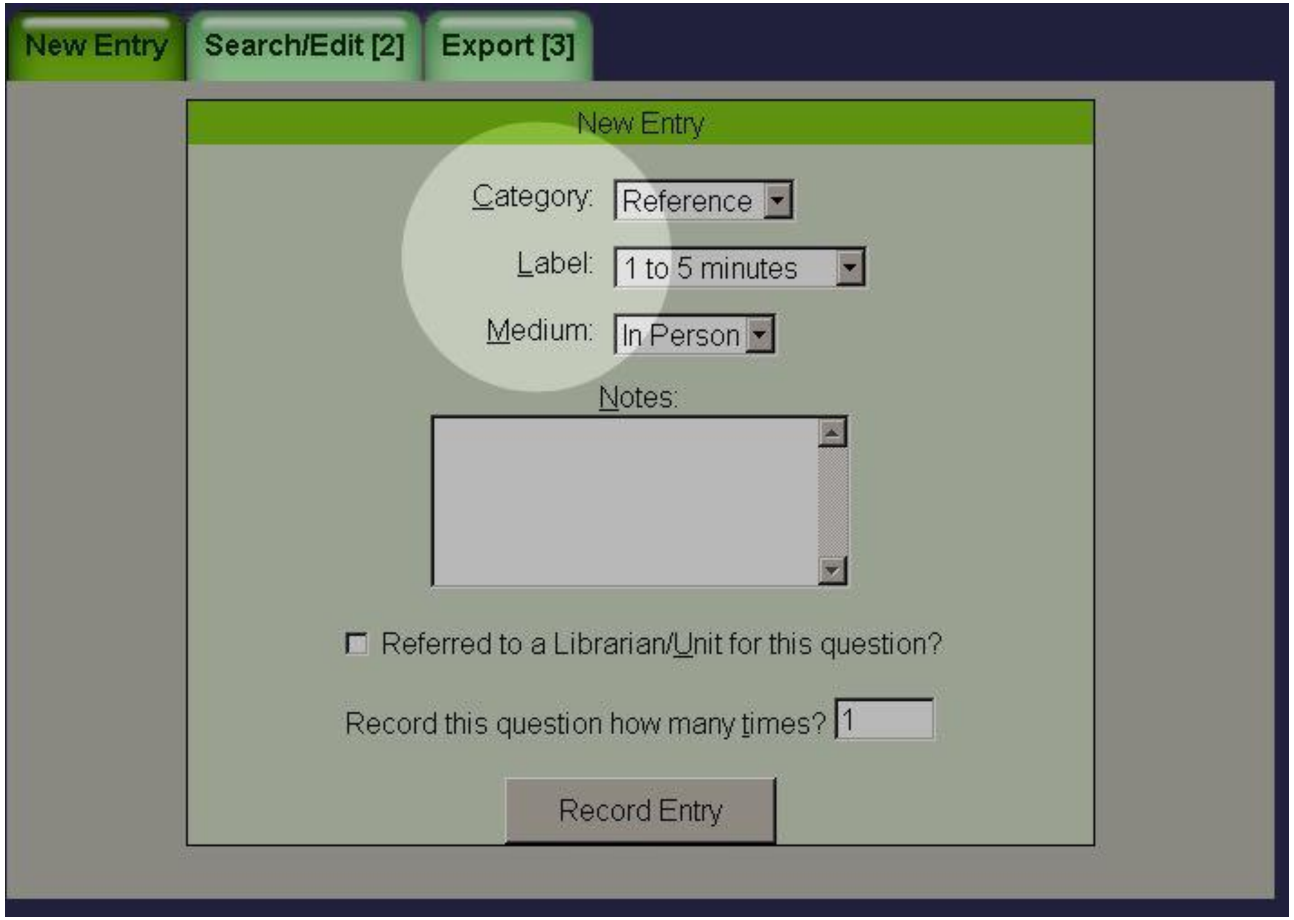
Category: Reference
Label: 1 to 5 minutes
Medium: In Person
Notes: gsd

You did not refer to a librarian/unit for this entry.
April 5, 2006, 12:54 AM

New Entry

Category: Reference
Label: 1 to 5 minutes
Medium: In Person
Notes:

What we've found:
Issue #4, Entry Semantics



New Entry

Category: Reference

Label: 1 to 5 minutes

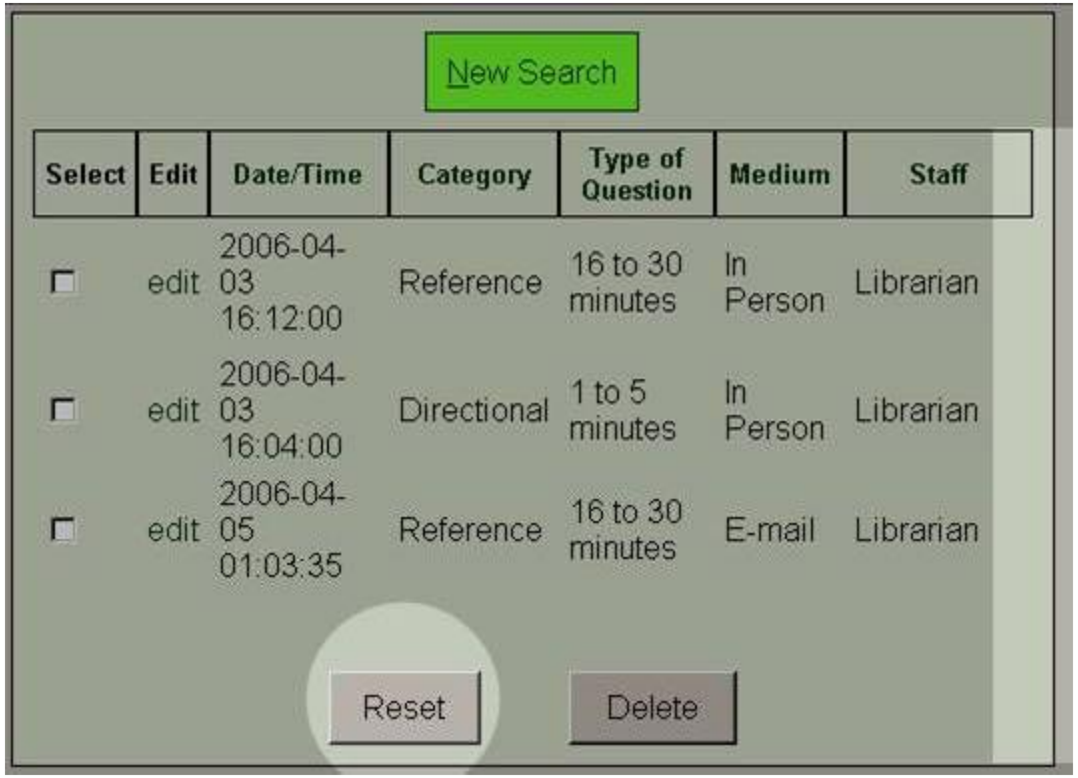
Medium: In Person

Notes:

□ Referred to a Librarian/Unit for this question?

Record this question how many times? 1

What we've found: Issue #5,
Search Results Disambiguation & Semantics



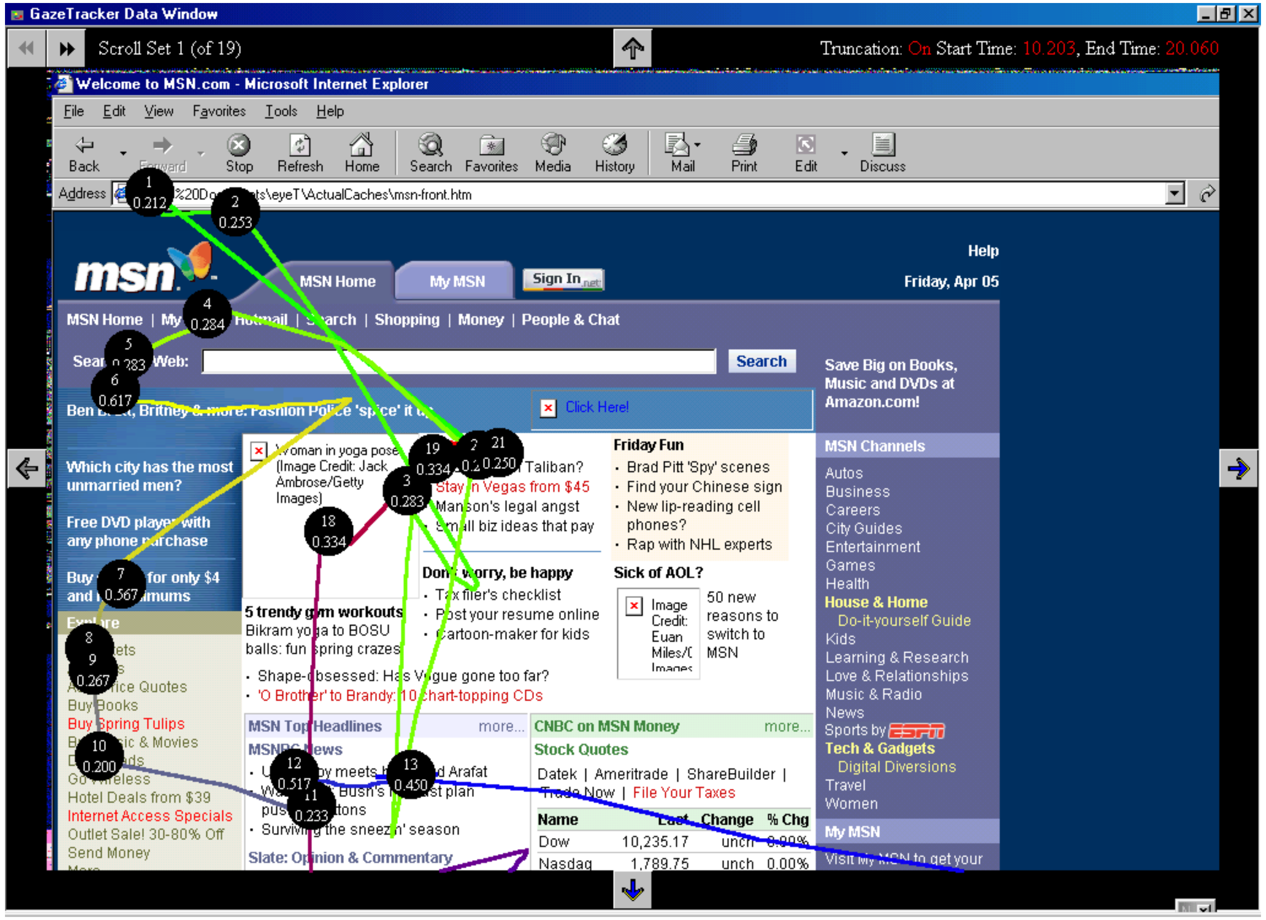
Select	Edit	Date/Time	Category	Type of Question	Medium	Staff
□	edit	2006-04-03 16:12:00	Reference	16 to 30 minutes	In Person	Librarian
□	edit	2006-04-03 16:04:00	Directional	1 to 5 minutes	In Person	Librarian
□	edit	2006-04-05 01:03:35	Reference	16 to 30 minutes	E-mail	Librarian

Measurement-based evaluations

- User testing can be done with (non-functional) prototypes
    - Requires more interaction with evaluator (risk of bias)
- Measurements require an operational system

- Log events in users' interactions with system
    - Clicks (when, where)
    - Navigation (from page to page)
    - Keystrokes
    - Use of help system
    - Errors encountered
    - Eye tracking
- May be used for statistical analysis or for detailed study of an individual user

Eye tracking



Save Big on Books, Music and DVDs at Amazon.com!

Ben Affleck, Britney & more: Fashion Police 'spice' it up

Which city has the most unmarried men?

Free DVD player with any phone purchase

Buy ... for only $4 and ... mums

Woman in yoga pose (Image Credit: Jack Ambrose/Getty Images)

Friday Fun

- Brad Pitt 'Spy' scenes
- Find your Chinese sign
- New lip-reading cell phones?
- Rap with NHL experts

Taliban?

Stay in Vegas from $45

Manson's legal angst

Small biz ideas that pay

Don't worry, be happy

- Tax filer's checklist
- Post your resume online
- Cartoon-maker for kids

Sick of AOL?

Image Credit: Euan Miles/C Images

50 new reasons to switch to MSN

5 trendy gym workouts

Bikram yoga to BOSU balls: fun spring crazes

- Shape-obsessed: Has Vogue gone too far?
- 'O Brother' to Brandy: 10 chart-topping CDs

Explore

Price Quotes

Buy Books

Buy Spring Tulips

Music & Movies

Go Wireless

Hotel Deals from $39

Internet Access Specials

Outlet Sale! 30-80% Off

Send Money

MSNBC News

- ... meets ... Arafat
- ... Bush's ... plan
- Surviving the sneezin' season

Slate: Opinion & Commentary

Stock Quotes

Datek | Ameritrade | ShareBuilder | Trade Now | File Your Taxes

Name	Last	Change	% Chg
Dow	10,235.17	unch	0.00%
Nasdaq	1,789.75	unch	0.00%

Visit My MSN to get your

Analyzing measurements

- Which interface options were used?
- When was the help system consulted?
- What errors occurred? From where and how often?
- Which links were followed? (clickthrough data)

- Human feedback (less structured)
  - Complaints and praise in feedback forms
  - Bug reports
  - Calls to customer service

Refining designs

- Do not allow test evaluators to become designers
  - Designers are poor evaluators of their own work,
  - But designers know requirements, constraints, context of design
    - Know which problems might be addressed with small changes
    - Know which problems require major changes that should be escalated
    - Know which user requests are mutually incompatible
      - Balance between configurability and simplicity (designer's job)
- Designers and evaluators must work as a team
  - But not try to do each other's work

User testing in CS 5150

- All projects must conduct user testing of user interfaces you design
  - Internal projects: recruit classmates from other teams
  - Decide how much training users should have
    - They should probably be familiar with existing system
    - You can provide training (but don't "teach to the test"), or a user manual
  - Design tasks & metrics
    - "Which files has your reviewer read so far?"
    - "Which, if any, of your commit messages has your reviewer left a comment on?"
    - "Add a reviewer comment to this file that was not modified"
  - Design survey

Code tracing

Techniques

- Monitor application logs
  - Enable logging for your project
- Developer tools network view
  - Look for mutating methods (POST, PUT, DELETE, vs. GET); ignore static resources
  - Look at initiator stack trace
    - Ignore framework methods (jQuery, etc.)
    - Look for promising files, then read them
- Search source code
  - Filter results (ignore static, tests, docs)