Final Exam Review

CS160 Spring 2005
Tara Matthews

Format

• Questions
  – Lots of short answer
  – 1 or 2 long answer
• Topics
  – 1 question on pre-midterm material
  – rest of questions will cover material since midterm
• Date
  – May 18, 8 am

Topics

• Review each lecture: there will be questions from all of them
  – Testing without users
  – Testing with users
  – Interface tools and platforms
  – Model-view controller
  – Handling user errors
  – Help and documentation
  – Design patterns
  – Information design and visualization
  – Speech and multimodal interfaces
  – Big Ideas from this course

Testing Without Users

• What is it?
• How do you conduct it?
• Advantages?
• Disadvantages?

Testing Without Users

• What is it?
  – Heuristic Evaluation
  – Cognitive Walkthrough
  – GOMS
• How do you conduct it?
• Advantages?
• Disadvantages?
Testing Without Users

- What is it?
- How do you conduct it?
- Advantages?
  - cheap: costs less than user studies
  - fast: takes days, not weeks
  - effective: can find problems
  - without users: user time may be difficult to get
- Disadvantages?

User Testing

- How do you conduct?
- Advantages?
- Disadvantages?

User Testing

- How do you conduct?
  - recruit representative users
  - ethical considerations
  - user test plan / report
  - select tasks
  - observe & record: process, bottom-line, qualitative data
  - analyze data: statistics, overall user opinions
- Advantages?
- Disadvantages?
Interface Tools & Platforms

• Sequential v. event-driven programming
  – sequential (or flow-driven)
  – event-driven

Model-View Controller

• What does MVC mean?
• Why separate M, V, & C?

• What does MVC mean?
  – Architecture for interactive apps
  – Model: represent & maintain data
  – View: display data (output)
  – Controller: handle events that affect the M or V (input)
• Why separate M, V, & C?

• What does MVC mean?
  – Can interchange input, output mechanisms, or the data being displayed (e.g., change view from 2D to 3D w/o changing M or C)
  • scalable
  • easy to maintain

Handling User Error

• Types of errors?
• Typical errors?
• Recovery strategies?

• Types of errors?
  – mistakes
  – slips
• Typical errors?
• Recovery strategies?
Handling User Error

• Types of errors?
  • Typical errors?
    – mode / modal errors: UI elements mean different things depending on mode
    – description error: action insufficiently specified by user
    – capture error: entering the wrong input due to similarities in commands/actions
  • Recovery strategies?

Help & Documentation

• Types of help?
  • quick reference: reminders, common commands
  • task-specific: how-to
  • full explanation: why, how
  • tutorial: lead user through task
  • Other types…
    – context-sensitive help (e.g., tooltips)
    – adaptive help systems
    – online documentation and tutorials

Design Patterns

• What is a design pattern?
• How is it formulated?
Design Patterns

• What is a design pattern?
  – A reusable formulation of a common solution to a problem
  – E.g., lots of buildings have alcoves; how have they been successfully designed in the past?

• How is it formulated?
  – Pattern title:
  – Context: collaborative and common areas in buildings
  – Forces: open spaces are inviting, but people want a sense of enclosure for private discussions
  – Problem statement: create a space that invites collaboration but also supports private discussion
  – Solution: make small places at the corners of common room, ~6 feet wide, 3-6 feet deep…
  – Solution sketch
  – Other patterns to consider: meeting room, courtyard

Questions?