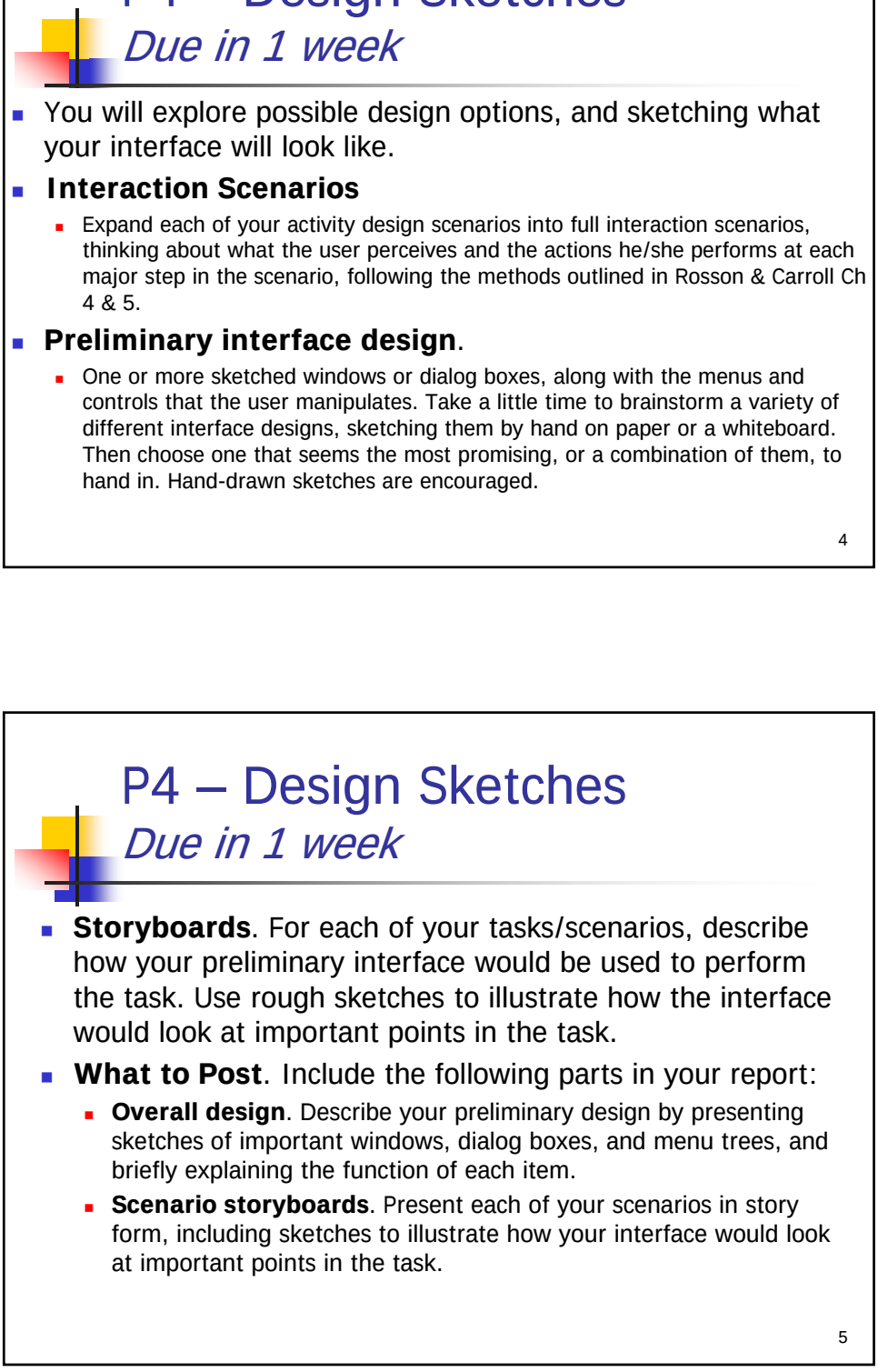
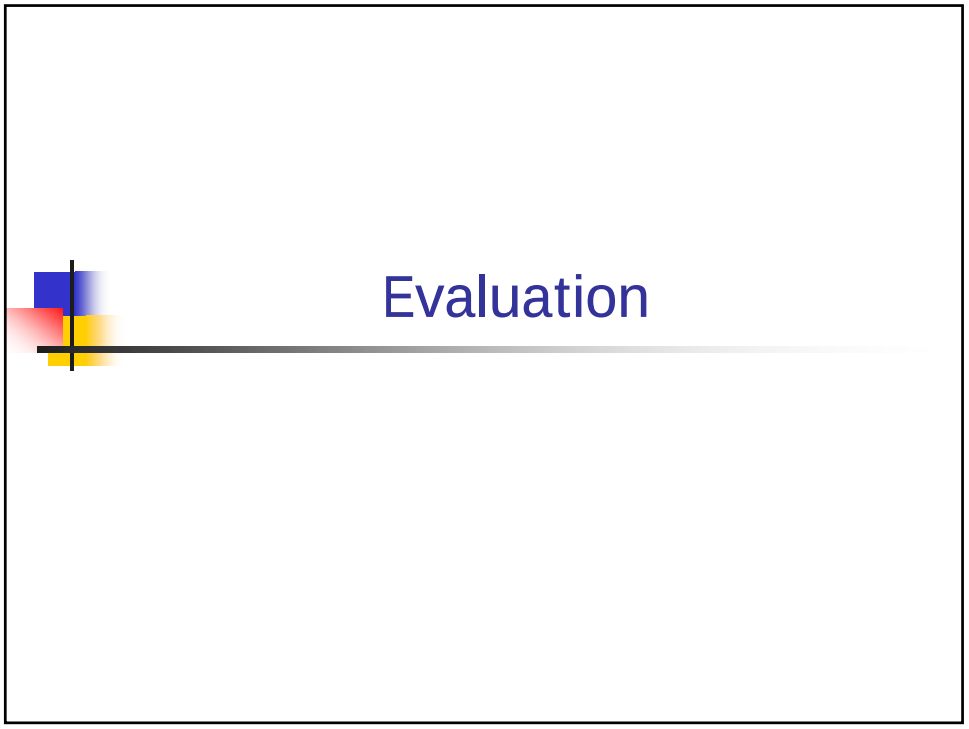
# Human-Computer Interaction IS4300

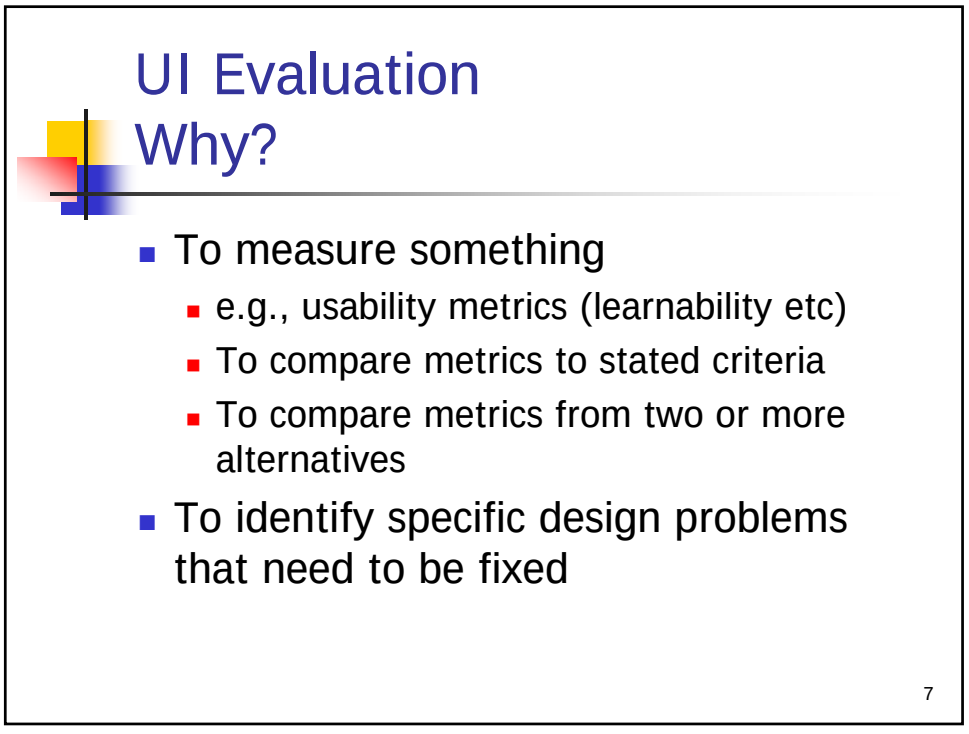


## I5 *due*

- Your mission in this exercise is to implement a very simple Java painting applet. The applet must support the following functions:
- Draw curves, specified by a mouse drag.
- Draw filled rectangles or ovals, specified by a mouse drag (don't worry about dynamically drawing the shape during the drag - just draw the final shape indicated).
- Shape selection (line, rectangle or oval) selected by radio buttons.
- Color selection using a combo box.
- Line thickness using a group of radio buttons.
- A CLEAR button.

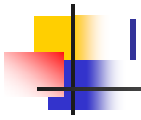

P4 – Design Sketches
Due in 1 week
You will explore possible design options, and sketching what your interface will look like.
Interaction Scenarios
Expand each of your activity design scenarios into full interaction scenarios, thinking about what the user perceives and the actions he/she performs at each major step in the scenario, following the methods outlined in Rosson & Carroll Ch 4 & 5.
Preliminary interface design.
One or more sketched windows or dialog boxes, along with the menus and controls that the user manipulates. Take a little time to brainstorm a variety of different interface designs, sketching them by hand on paper or a whiteboard. Then choose one that seems the most promising, or a combination of them, to hand in. Hand-drawn sketches are encouraged.
4
P4 – Design Sketches
Due in 1 week
Storyboards. For each of your tasks/scenarios, describe how your preliminary interface would be used to perform the task. Use rough sketches to illustrate how the interface would look at important points in the task.
What to Post. Include the following parts in your report:
Overall design. Describe your preliminary design by presenting sketches of important windows, dialog boxes, and menu trees, and briefly explaining the function of each item.
Scenario storyboards. Present each of your scenarios in story form, including sketches to illustrate how your interface would look at important points in the task.
5


Evaluation


UI Evaluation
Why?
To measure something
e.g., usability metrics (learnability etc)
To compare metrics to stated criteria
To compare metrics from two or more alternatives
To identify specific design problems that need to be fixed
7

# UI Evaluation Methods

- Expert/Inspection methods
  - Heuristic evaluation
  - Cognitive walk-through
  - Modeling
- User Testing
  - Qualitative methods (interviews, questionnaires, think aloud)
    - observation in the field
  - Quantitative methods
    - Descriptive studies
    - Experiments (same environment & task with 2 or more alternative designs)
- Generally: Qualitative used for debugging; Quantitative for measures (summative; testing hypotheses; comparisons)



# Inspection methods

- Heuristic evaluation
  - Checklist approach
  - Independent evaluators (3-5)
  - Severity rating for problems

    0. No problem
    1. Cosmetic problem
    2. Minor – low priority
    3. Major problem – high priority
    4. Catastrophe – must fix

## Inspection methods

- Cognitive walkthrough
  - Walk through each step in the task and evaluate:
    1. Is the effect of the action the same as the user's goal at that point?
    2. Will users see that the action is available?
    3. Once users have found the action, will they know it is the one they need?
    4. After the action is taken, will users understand the feedback they get?

## Model-based evaluation

- Build and evaluate a formal model
- e.g. GOMS
  - Goals, Operators, Menus, Selection

## Problem

- Mary has just designed a web site for her daughter's girl scout troup, allowing the public to order cookies. She'd like the site to be as usable as possible. She has no budget, and no access to users, but does have a copy of Nielsen's usability book. What is the most appropriate evaluation method for her situation? Why?

12

## Problem

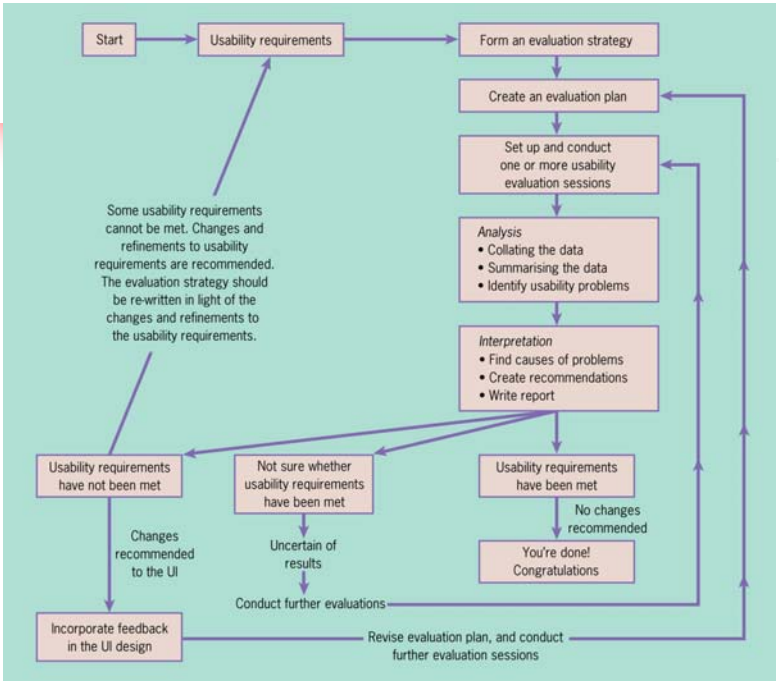
- Megabuck Inc's R&D department has just designed a new replacement for the desktop mouse that they say will revolutionize computing by cutting time-to-target in half. "Prove it" says the CEO. What is the most appropriate evaluation method? Why?

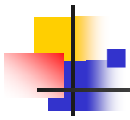
13

# Problem

- Startup Industries is thinking of developing a new web portal linking office gossip blogs, and have developed an early prototype, but they're not sure if anyone will want to use it. What kind of evaluation method should they use? Why?

14



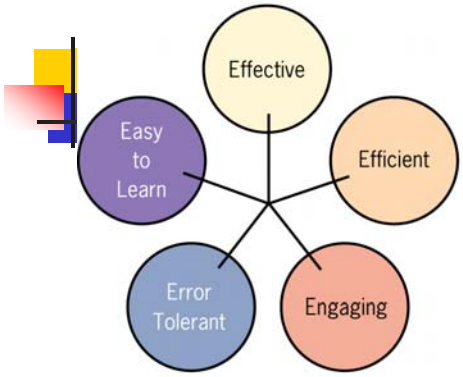


## Creating an Evaluation Strategy

- What Is the Purpose of This Evaluation?
  - Does system meet usability requirements/concerns
    - "Railway clerks work in extremely noisy environments, so any warning messages to them should be visually distinct and highlighted on the screens."

## Creating an Evaluation Strategy

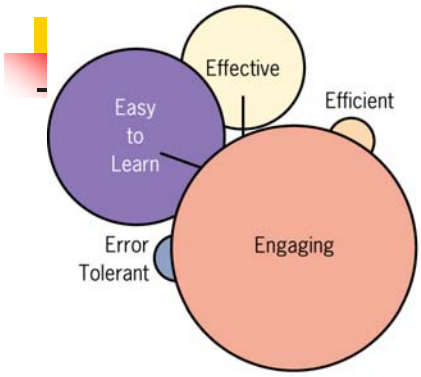
- What is the purpose of the evaluation?
- Usability Requirements & Metrics
  - Quantitative (per Nielsen)
  - Qualitative Usability Requirements
    - "The users on an e-shopping site should be able to order an item easily and without assistance."
- Prioritizing requirements

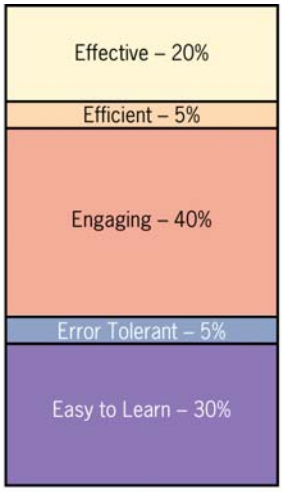




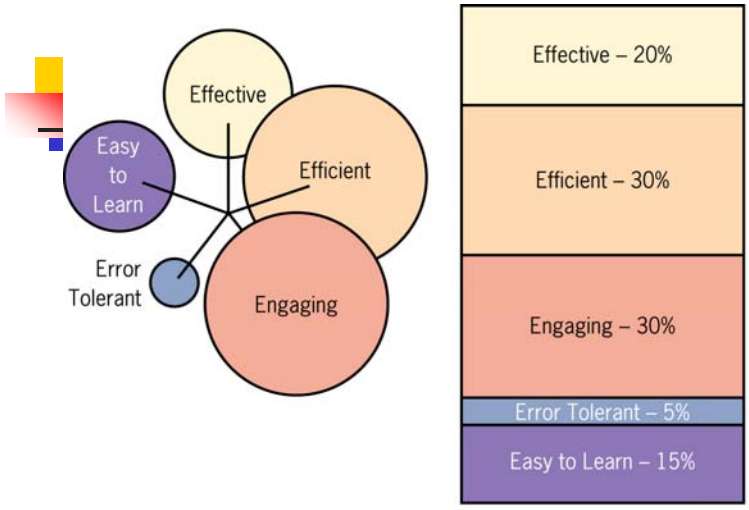
Different products may weight usability requirements differently.

## Online Museum Exhibition



- General Museum Site




# Evaluation Strategy

- What Am I Evaluating? (prototype or ?)
- What Constraints Do I Have?
  - Money
  - Timescales
  - Availability of usability equipment
  - Availability of participants and the costs of recruiting them
  - Availability of evaluators
- Documenting the Evaluation Strategy

## Usability Test Research Plan

- Experiments can take a huge amount of time to plan and prepare.
- Extreme example: medical clinical trials
- **What kinds of things do you need to worry about when planning a study?**

26

## Sample Research Plan

Embodied Conversational Agents to Promote Health Literacy for Older Adults

## Sample Research Plan

A. SPECIFIC AIMS

B. BACKGROUND AND SIGNIFICANCE

C. PRELIMINARY STUDIES

D. RESEARCH DESIGN AND METHODS

E. HUMAN SUBJECTS

28

## Sample Research Plan

### Hypotheses

- H1. Immediate and distal knowledge gains and glycemic control will be significantly improved when the current standard of care is augmented with a brief “virtual consultation” with an embodied conversational agent, compared to the current standard of care alone.
- H2. Patient satisfaction will be greater when the current standard of care is augmented with an embodied conversational agent, compared to the current standard of care alone.

29

# Sample Plan
# Research Model

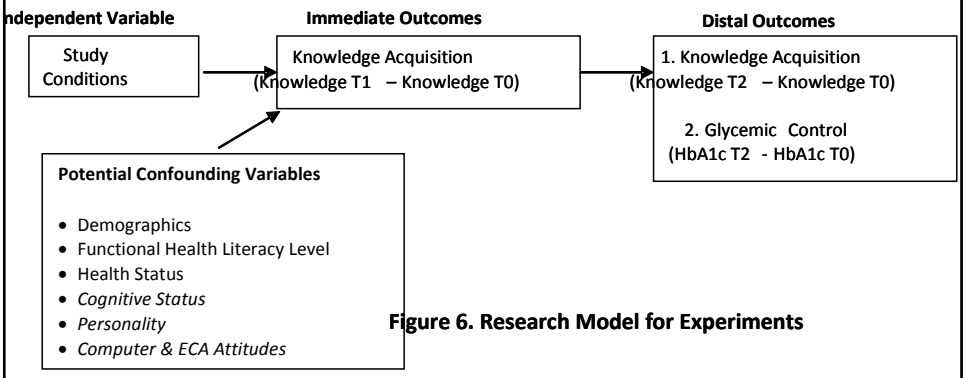



**Figure 6. Research Model for Experiments**

30

# Measures

- D.3.1.1 Independent Variables
  - Agent vs. Standard of Care
- D.3.1.2 Primary Dependent Variables
  - Diabetes Knowledge
  - HA1c
  - Satisfaction
- D.3.1.3 Potential Confounding Variables
  - Sociodemographics
  - Health literacy
  - Cognitive status
  - Etc.

31

## Sample Measure

- Diabetes Knowledge.
  - Diabetes knowledge will be assessed using the Diabetes Knowledge (DKN) Scales, three separate 15-item multiple choice questionnaires that measure general diabetes knowledge. Reliability for the items in the scales (Cronbach's alpha) was 0.92, indicating high internal consistency. Validity was assessed by determining that 219 participants who participated in a 1-1/2 day class on diabetes scored significantly higher posttest on the measures compared to pretest (11.27 vs. 7.61, $p<.001$).
  - We will administer the DKN immediately before the educational intervention (T0), immediately following the intervention (T1), and at three months follow up (T2).

32

## Study Population

- D.3.2 Study Population
- D.3.2.1 Study Setting: The Geriatric Ambulatory Practice
- D.3.2.3 Eligibility and Exclusion Criteria
  - Eligibility criteria include:
    - Age 60 years or greater,
    - Have Type 2 diabetes mellitus, with or without complications (ICD-9 codes 250.00-250.90)
  - Exclusionary criteria include:
    - Patients with significant cognitive disability …

33

## Sample Plan

- D.3.3 Sample Size and Power Considerations

34

## Sample Plan

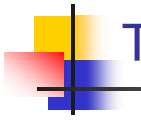
- D.3.4 Recruitment and Data Collection Procedures
  - D.3.4.1 Study Subjects
  - D.3.4.2 Recruitment and Initial Telephone Interview
  - D.3.4.3 Initial Clinic Visit (T0, T1)
  - D.3.4.4 Follow-up Clinic Visit (T2)

35

# Sample Plan

- D.3.5 Analysis
  - D.3.5.1 The Analysis Plan

36



# Types of Study Designs

- Quantitative
  - Descriptive
  - Correlational
  - Demonstrative
  - Experimental
    - Between-subjects
      - Single factor, two-level
      - Single factor, N-level (for N>2)
      - Two factor, N-level (for N>=2)
    - Within-subjects
      - Single factor, two-level

37

## Kinds of measures

- Nominal
- Ordinal
- Scale
- Ratio

Nominal, Ordinal: Categorical

Scale, Ratio: Numeric

38

## Types of study design

| | Number of Variables | Number of IV Levels | Manipulation |
|---|---|---|---|
| Descriptive | 1 | NA | NA |
| Demonstration | $\geq 2$ | 1 | √ |
| Correlational | $\geq 2$ | NA | NA |
| Experimental | $\geq 2$ | $\geq 2$ | √ |

39

# The logic of hypothesis testing

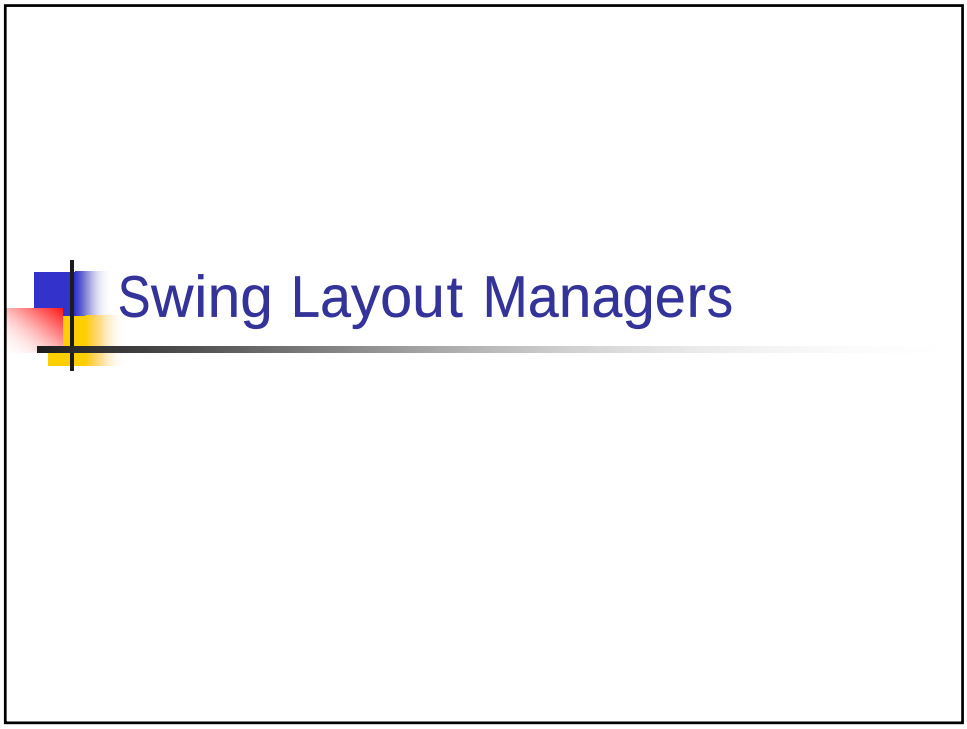
| | "The Truth" H1 True | "The Truth" H1 False |
|---|---|---|
| **Decide to Reject H0 & accept H1** | Correct p = power | Type I err p = $\alpha$ |
| **Do not Reject H0 & do not accept H1** | Type II err p = $\beta$ | Correct p = 1-$\alpha$ |

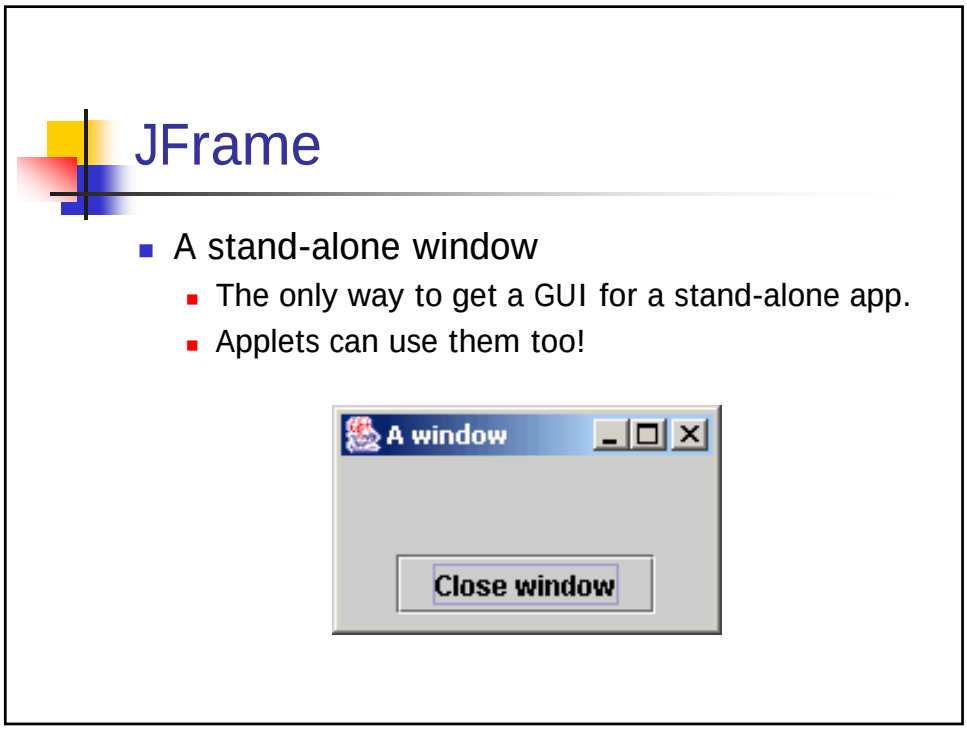
40

# Some Statistical Tests

- X^2 tests
  - For nominal measures
  - Can apply to a single measure
- Correlation tests
  - For two numeric measures
- t-test for independent means
  - For categorical IV, numeric DV
- t-test for dependent means
- 1-way ANOVA
- Factorial ANOVA
- Non-parametric tests
- Regression

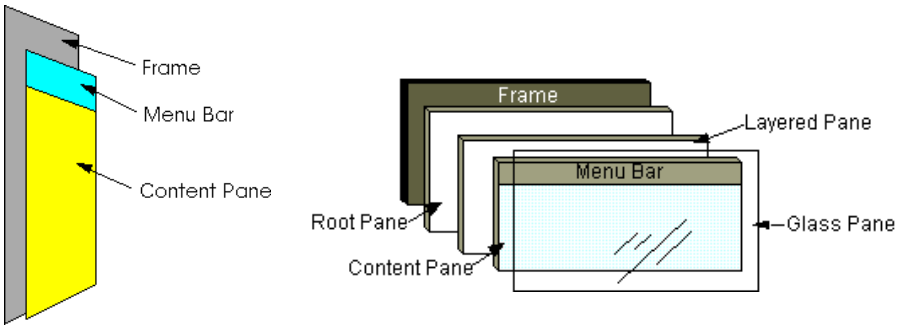
41


Swing Layout Managers


JFrame
A stand-alone window
The only way to get a GUI for a stand-alone app.
Applets can use them too!
A window
Close window

# JFrame guts

- We're just going to focus on the Content Pane




# Creating a JFrame

```
class MyFrame extends JFrame {
    public MyFrame() {
      super("My Example");
       setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      //populate the content pane: getContentPane() …
      pack();
    }
    public static void main(String args[]) {
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new MyFrame().setVisible(true);
            }
        });
    }
}
```

## JDialog

- Just like a JFrame except you can make it *modal*
- *Note:* Use JOptionPane for simple, standard alert & informational message dialogs.
- JColorChooser, JFileChooser – built in, special-purpose dialogs.

## Layout Managers

- Decide how to display the Components within a Container.
- To use a layout manager:
  - Construct an instance of the manager.
  - Assign the instance to the container using:
    setLayout(LayoutManager)
  - Each Container can only have one layout manager.
- Or in NetBeans:
  R-click on component and choose “Set Layout…”

# FlowLayout

- The default for JPanel
- Strategy:
  - Keeps components at their preferred size. Place components in rows, left-to-right. When a row fills up, a new row is started.
  - Rows can be centered, left or right justified.

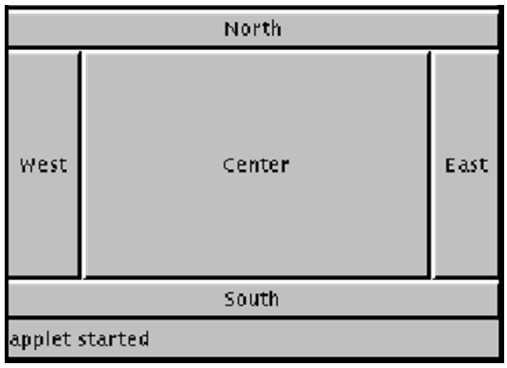
# Example FlowLayout

```
class FlowLayoutExample extends JFrame {
    public FlowLayoutExample() {
       super("Flow Layout Example");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        Container frame=getContentPane();
       frame.setLayout(new FlowLayout(FlowLayout.LEFT,10,10));
       frame.add(new Button("Button 1"));
       frame.add(new Button("Button 2"));
       frame.add(new Button("Button 3"));
       frame.add(new Button("Button 4"));
       frame.add(new Button("Button 5"));
       pack();
}
```

# BorderLayout

- Partitions the layout space into regions
  - You specify which region you want to place Components into by name
  - At most one component can go into each region

`add(Component,<where>)`




# BorderLayout Example

```
class BorderLayoutExample extends JFrame {
    public BorderLayoutExample() {
      super("Border Layout Example");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        Container frame=getContentPane();
      frame.setLayout(new BorderLayout());
      frame.add(new Button("Button 1"),BorderLayout.NORTH);
      frame.add(new Button("Button 2"),BorderLayout.SOUTH);
      frame.add(new Button("Button 3"),BorderLayout.EAST);
      frame.add(new Button("Button 4"),BorderLayout.WEST);
      frame.add(new Button("Button 5"),BorderLayout.CENTER);
      pack();
    }
```

# GridLayout

- Forms a rectangular grid of rows and columns
  - You specify the number of rows, columns, or both
  - Components are forced into the same shape for **every** cell.
  - Grid is filled left-to-right, top-down
- Constructor

  `GridLayout(int rows,int cols)`
  - Value of zero denotes undefined

# GridLayout Example

```
class GridLayoutExample extends JFrame {
    public GridLayoutExample() {
      super("Grid Layout Example");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        Container frame=getContentPane();
      frame.setLayout(new GridLayout(0,2));
      frame.add(new Button("Button 1"));
      frame.add(new Button("Button 2"));
      frame.add(new Button("Button 3"));
      frame.add(new Button("Button 4"));
      frame.add(new Button("Button 5"));
      pack();
    }
```

## CardLayout

- Swaps among each of its components
- Each component can be named:
  `add("name",Component)`
- First component displayed initially
- To swap among components

```
CardLayout.next(Container parent)
CardLayout.first(Container parent)
CardLayout.last(Container parent)
CardLayout.show(Container parent,"name")
```

## JTabbedPane

- Acts like a JPanel with a CardLayout

# Hierarchical Example

```
class HierarchyExample extends JFrame {
    public HierarchyExample() {
        super("Hierarchy Layout Example");
         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
         Container frame=getContentPane();
        JPanel P1=new JPanel();
        JPanel P2=new JPanel();
        JPanel P3=new JPanel();
        P1.setLayout(new BorderLayout());
        P1.add(new TextArea(5,15),BorderLayout.CENTER);
        P1.add(new Button("Clear"),BorderLayout.SOUTH);
        P2.setLayout(new GridLayout(0,1));
        P2.add(new Button("Option 1"));
        P2.add(new Button("Option 2"));
        P2.add(new Button("Option 3"));
        P3.setLayout(new FlowLayout());
        P3.add(new Button("OK"));
        P3.add(new Button("Cancel"));
        frame.setLayout(new BorderLayout());
        frame.add(P1,BorderLayout.EAST);
        frame.add(P2,BorderLayout.WEST);
        frame.add(P3,BorderLayout.SOUTH);
        pack();
    }
```

# Exercise – what will this look like?

```
P1.setLayout(new BorderLayout());
P1.add(new TextArea(5,15),BorderLayout.CENTER);
P1.add(new Button("Clear"),BorderLayout.SOUTH);
P2.setLayout(new GridLayout(0,1));
P2.add(new Button("Option 1"));
P2.add(new Button("Option 2"));
P2.add(new Button("Option 3"));
P3.setLayout(new FlowLayout());
P3.add(new Button("OK"));
P3.add(new Button("Cancel"));
frame.setLayout(new BorderLayout());
frame.add(P1,BorderLayout.EAST);
frame.add(P2,BorderLayout.WEST);
frame.add(P3,BorderLayout.SOUTH);
```


## To Do

- No class Monday
- Read
  - Rettig article on paper prototyping
  - 3 Research articles on paper prototyping
  - Quiz: Come prepared with one research question about each of the articles
- Finish by next class
  - P4 – design sketches
- Finish in two weeks
  - I6 – Swing Layout Managers

59

Pers Ubiquit Comput (2008) 12:269–277
DOI 10.1007/s00779-007-0147-2

ORIGINAL ARTICLE

# Adapting paper prototyping for designing user interfaces for multiple display environments

Brian P. Bailey · Jacob T. Biehl · Damon J. Cook · Heather E. Metcalf

Received: 13 February 2006 / Accepted: 2


**Abstract** A multiple display en networks personal and shared devic workspace, and designers are just b with the challenges of developing


CHI 2010: End-User Programming I April 10–15, 2010, Atlanta, GA, USA

# FrameWire: A Tool for Automatically Extracting Interaction Logic from Paper Prototyping Tests

Yang Li [1*] Xiang Cao [2] Katherine Everitt [1] Morgan Dixon [1] James A. Landay [1]
[1] DUB Institute, University of Washington, Seattle, WA [2] Microsoft Research Cambridge, UK
{yangli, xiangcao}@acm.org, katherine.everitt@gmail.com, {mdixon, landay}@cs.washington.edu

**ABSTRACT**
Paper prototyping offers unique affordances for interface design. However, due to its spontaneous nature and the limitations of paper, it is difficult to distill and communicate a paper prototype design and its user test findings to a wide audience. To address these issues, we created FrameWire, a computer vision-based system that automatically extracts interaction flows from the video recording of paper prototype user tests. Based on the extracted logic, FrameWire offers two distinct benefits for designers: a structural view of the video recording that allows a designer or a stakeholder to easily distill and understand the design concept and user interaction

playing the role of the "computer", presents an interface screen (e.g., drawn on a piece of paper) to a user according to the user's actions. The user interacts with the interface

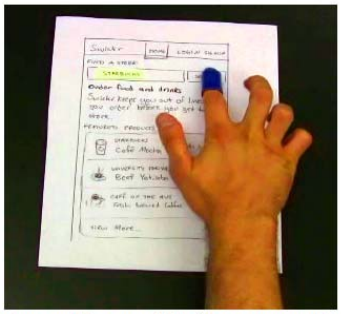


Figure 2. A user clicks on an interface component by tapping

# DisplayObjects: Prototyping Functional Physical Interfaces on 3D Styrofoam, Paper or Cardboard Models

**Eric Akaoka, Tim Ginn and Roel Vertegaal**

Human Media Lab
Queen's University
Kingston, ON K7L 3N6

info

**ABSTRACT**

This paper introduces DisplayObjects, a rapid protot
workbench that allows functional interfaces to be proj
onto real 3D physical prototypes. DisplayObjects u
Vicon motion capture system to track the locatic
physical models. 3D software renditions of the 3D ph
model are then texture-mapped with interactive bel
and projected back onto the physical model to allow
time interactions with the object. We discuss
implementation of the system, as well as a selection o
and two-handed interaction techniques for DisplayOb
We conclude with a design case that comments on sor
the early design experiences with the system.


Figure 3. Brick model with 5 retroreflective markers