

Midterm Review

Administration

Midterm

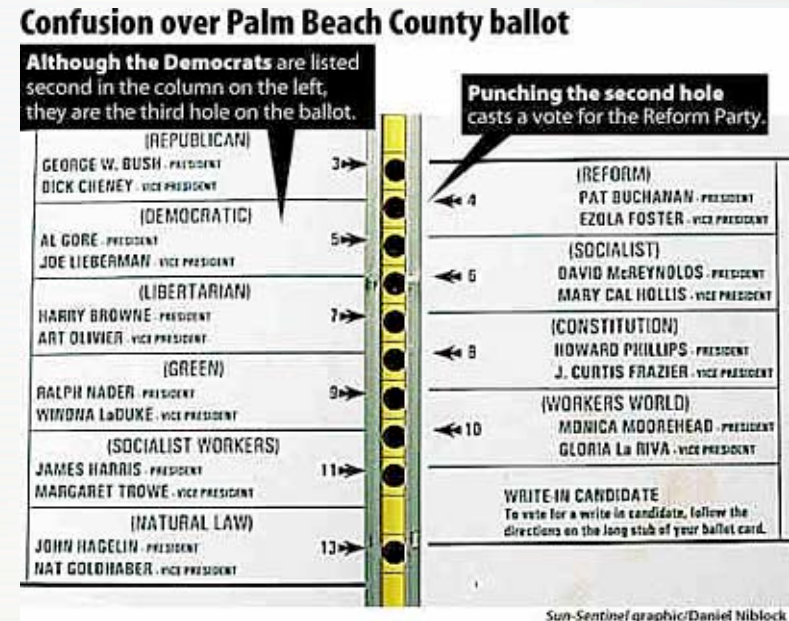
- **80 minutes, start promptly at 14:05**
- **Style:**
 - **Short answer**
 - **Essay: apply a concept**
 - **Problem solving: programming (limited)**

Learning Goals

- **Express yourself in executable form**
- **Basics of what is hard and easy to rapidly prototype**
- **Terminology and approaches used by programmers, so you can work with them**
- **Experience pain of programming**
- **Design and conduct informal user tests**

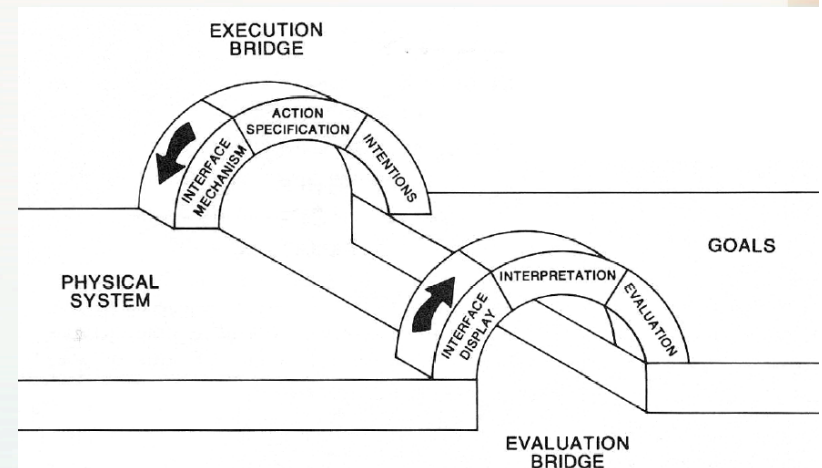
User Interfaces Introduction

- Terms: **Users, user interfaces, usability**
- Why are interfaces important?
- Why are interfaces hard to design?
- Why are interfaces hard to implement?



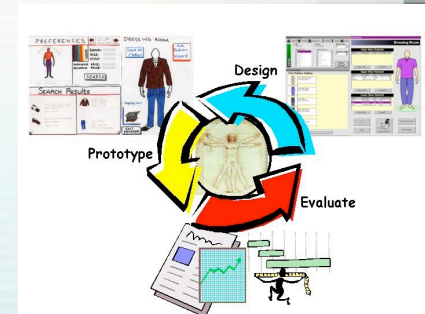
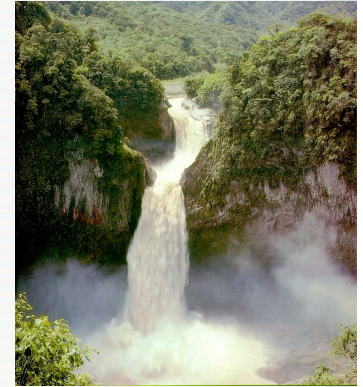
What is Design?

- **Terms: design, affordances, user conceptual model, constraints, natural mappings, feedback**
- **Good and bad examples**
- **Norman's 7 stages**
- **Gulf of evaluation and gulf of execution**
- **Tradeoffs/issues**
- **Design support**
- **How designers work**



Usability Engineering Design Process

- Terms: **waterfall model, iterative process**
- **10 steps of process, issues**
 1. Study the users and their tasks
 2. Study the competition
 3. Set usability goals
 4. Participatory Design
 5. Coordinating the Total Interface for Consistency
 - Include documentation, help, etc.
 6. Guidelines and Heuristic Evaluation
 - Evaluate your interface according to the guidelines.
 7. Make prototypes of the system early and quickly
 - Actually is faster to prototype first
 8. Empirical testing
 9. Iterative design
 10. Collect feedback from field use
- **Obstacles/warnings**



Prototyping

- Terms: **prototype, lo-fi, medium-fi, high-fi**
- **What, why, who, when, how**
- **Types of prototypes**
- **Types of prototyping**
- **Trade-offs**
- **Testing**
- **Support tools**
- **Paper prototyping exercise/lessons**



UI Software Organization

- **Terms: separation of concerns, windows system, windows manager, toolkit, UIDE**
- **UI flow**
- **Models**
 - **Model-View-Controller**
 - **Object-oriented**
- **Layers of UI software**
- **Window System: input and output model**
- **Window Manager**
- **Toolkit and High-Level Tools**

Debugging

- **Terms: bug, debugging**
- **Why debug?**
- **Why is it hard?**
- **Types of bugs, how to fix**
- **Debugging steps and approach**
- **Debugging strategies**
- **Tools**

Output Styles

- **Terms: metaphors, styles**
- **Issues with interaction styles**
 - How do you choose?
- **Interaction styles: pros/cons**
 1. Question and answer,
 2. Single character commands and/or function keys,
 3. Command Language,
 4. Menus
 5. Forms/Dialogue Boxes
 6. Direct Manipulation
 7. WYSIWYG
 - really is a subclass of DM, not another style
 8. Gestures
 9. Natural Language
 10. Natural Behavior

Output Graphics

- **Terms: anti-aliasing**
- **Models: stroke, pixel, region, color**
- **Coordinate systems**
- **Drawing Objects: Lines, Bezier Curves, Fonts, FontMetrics, Images,**
- **Transformations**

Input Devices

- **Why harder than output?**
- **Devices: keyboard, buttons, valuator, locators,**
- **Absolute, relative, clutched absolute locators**

Input Models

- **Terms: events**
- **Logical devices, events, sampling**
- **Unified model of events**
- **What does an event consist of?**
- **Extending events**
- **Synchronizing problem**
- **Dispatching and handling events**

Questions?