Review for the Final
Final

- **Style**
  - Short answer
  - Essay: apply a concept
- **Length**
  - 90 minutes
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, FRAME BUFFER
• Coordinate systems
• Drawing Objects: Lines, Bezier Curves, Fonts, FontMetrics, Images,
• Transformations
Input Devices

• Why harder than output?
• Devices: keyboard, buttons, valuators, locators,
• Absolute, relative, clutched absolute locators
Input Models

• Terms: \textit{events}
• Logical devices, \textit{events}, sampling
• Unified model of \textit{events}
• What does an \textit{event} consist of?
• Extending \textit{events}
• \textbf{Synchronizing} \textit{problem}
• Dispatching and handling \textit{events}
Interaction Techniques

• Relation to interaction styles and widget libraries
• Macintosh 7
• Features for design/selection
  – Affordance, feedback, performance (feel, Fitt’s Law)
  – Guidelines
• Advantages/disadvantages of widget libraries
• Choosing a technique
Finite State Machines

- What are they, what do they do?
- Relation to interaction techniques
- Why do we need them?
- Notation: enough to draw a very simple FSM, or explain FSM
- Relation of FSM to event loops
- Why don’t they scale?
- General technique on how to combine 2 independent finite state machines
Properties of People

• What are **mental models**: difference between system designer’s and end user’s

• Good UI: convenient access to functionality to complete task efficiently & user’s mental model accurately predicts interface action
  – Only one is a property of people
  – Affect with feedback, affordances
Properties of People

• UI Guidelines
  – System designer can’t pretend to be a user
  – Explicitly design conceptual model and use feedback and affordance to reinforce
  – Premature optimization is bad
  – Errors are not exceptional events -- > help form mental model
Properties of People

• Performance:
  – How long physical motion takes: Fitt’s Law
  – How much can people remember: short term, long term memory, recognition vs. recall
  – How fast do people perceive: STM decay, bad response time, expectations, consistency
Animation

• What value do they serve?
• Challenges in prototyping: animation
• **Animation**: visual continuity enhancing perception (change); draws attention
• 3 principles:
  – **Solidity**: objects appear solid
  – **Exaggeration**: exaggerate physical actions to enhance perception
  – **Reinforcements**: effects to drive home feeling of reality
Animation

• Solidity:
  – Motion *blur*
  – *Squash and stretch* (mass and shape)
  – *Follow through*: objects don’t stop

• Exaggeration: tweak perception
  – *Anticipation*, squash and stretch, follow through

• Reinforcement:
  – *Slow-in/slow-out*, move in arcs

• 3 parts of motion: anticipation, motion, follow through
Internationalization

- What is it, why important, how support
- Interface designed for different cultures
- **Internationalization vs. localization**
- How icons come to have meaning
  - Arbitrary, reference, resemblance
  - What to avoid
- Care in wording, numbers
- Implications for design:
  - Space, layout, content, decide what to translate, pictures vs. text
Context awareness

• Give examples of smart spaces
• Give brief summary of issues relating to
  – Privacy
  – Feedback
  – Affordances
• Questions?