

CS 520

Theory and Practice of Software Engineering
Fall 2019

User Interfaces

October 31, 2019

1

coming up

- No class on Nov 5 (but in-class 3 is due that day)
 - Work on final-projects
 - Mid-point presentations due Nov 7
<https://people.cs.umass.edu/~brynj/cses/2019Fall/CS520/midProjectReport.pdf>
 - Attend guest lecture on Nov 6, 12 PM, room CS 151
 - Talia Ringer, University of Washington
 - “Proof Engineering Tools for a New Era”
 - Pizza, if you attend, while supplies last
(video of talk will be posted after)



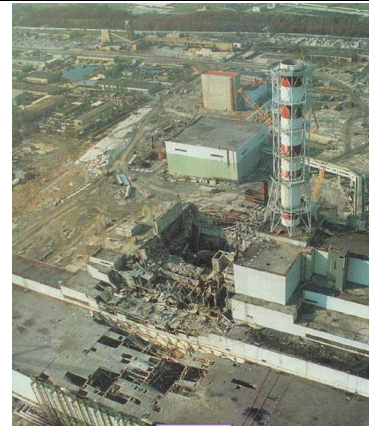
2

User Interface



Three Mile Island

3



Chernobyl

4

How do we avoid bad UI?

- Learn from past mistakes
- Build prototypes

5

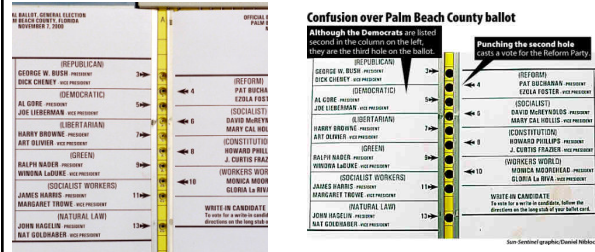
Big questions

- What's the point of prototyping? Should I do it?
 - If so, when should I?
- Should I make my prototype on paper or digitally?
- How do I know whether my UI is good or bad?
 - What are the ways in which a UI quality can be quantified?
 - What are some examples of software you use that have an especially good/bad UI?
 - What do you think makes them good/bad?

6

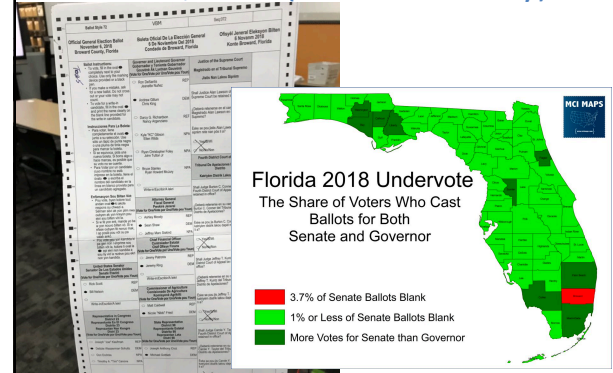
Usability and software design

- **usability**: the effectiveness of users achieving tasks
 - Human-Computer Interaction (HCI).
 - Usability and good UI design are closely related.
 - A bad UI can have serious results...



7

Florida, 2018 (Broward County)



8

Achieving usability

- User testing and field studies
 - having users use the product and gathering data
- Evaluations and reviews by UI experts
- Prototyping
 - Paper prototyping
 - Code prototyping
- Good UI design focuses on the *user*
 - not on the developer, not on the system environment

9

Prototyping

- **prototyping**: Creating a scaled-down or incomplete version of a system to demonstrate or test its aspects.
- Reasons to do prototyping:
 - aids UI design
 - provides basis for testing
 - team-building
 - allows interaction with user to ensure satisfaction

10

Some prototyping methods

1. UI builders (Visual Studio, ...)
 - draw a GUI visually by dragging/dropping UI controls on screen
2. implementation by hand
 - writing a quick version of your code
3. **paper prototyping**: a paper version of a UI



11

Why do paper prototypes?

- much faster to create than code
- can change faster than code
- more visual bandwidth (can see more at once)
- more conducive to working in teams
- can be done by non-technical people
- feels less permanent or final

12

Where does paper prototyping fit?

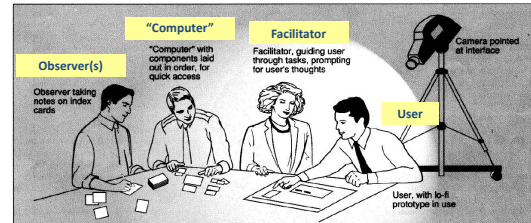
When in the software lifecycle is it most useful to do (paper) prototyping?

- Requirements are the **what** and design is the **how**. Which is paper prototyping?
- Prototyping
 - helps uncover requirements and upcoming design issues
 - during or after requirements but before design
 - shows us **what** is in the UI, but also shows us details of **how** the user can achieve goals in the UI

13

Paper prototyping usability session

- user gets tasks to perform on a paper prototype
- observed by people and/or recorded
- a developer can "play computer"



14

Schneiderman's 8 Golden Rules

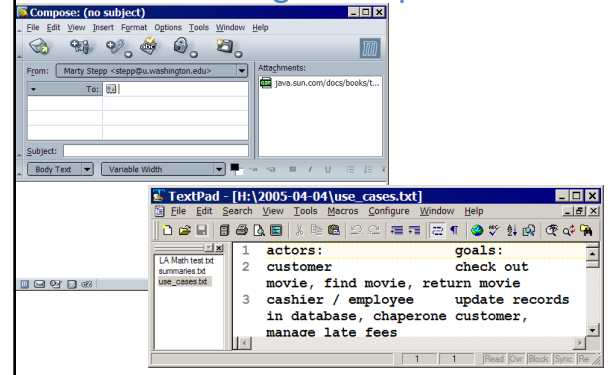
1. Strive for consistency.
2. Give shortcuts to the user.
3. Offer informative feedback.
4. Make each interaction with the user yield a result.
5. Offer simple error handling.
6. Permit easy undo of actions.
7. Let the user be in control.
8. Reduce short-term memory load on the user.



(from *Designing the User Interface*, by Ben Schneiderman of UMD, noted HCI and UI design expert)

15

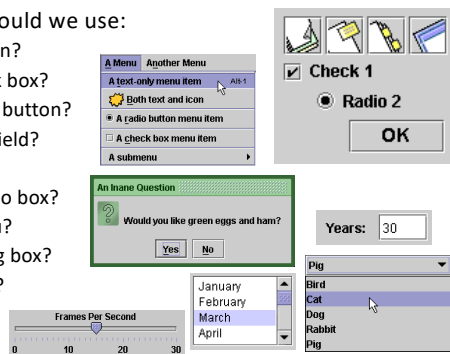
UI design examples



16

UI design, components

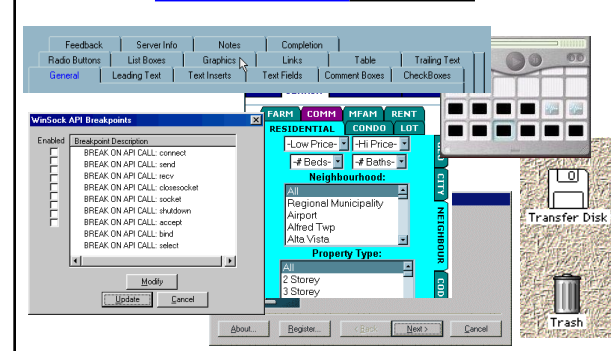
- When should we use:
 - A button?
 - A check box?
 - A radio button?
 - A text field?
 - A list?
 - A combo box?
 - A menu?
 - A dialog box?
 - Other..?



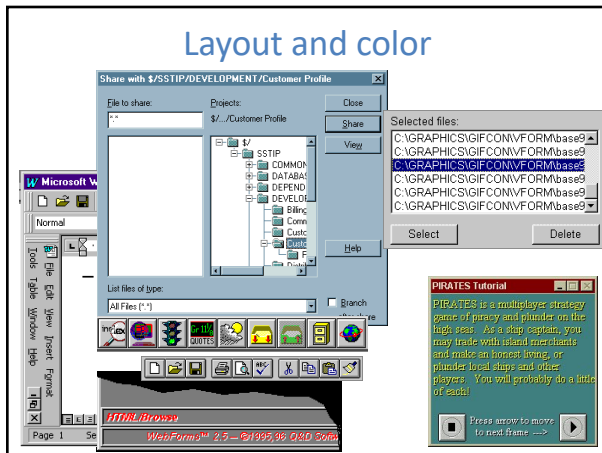
17

UI Hall of Shame

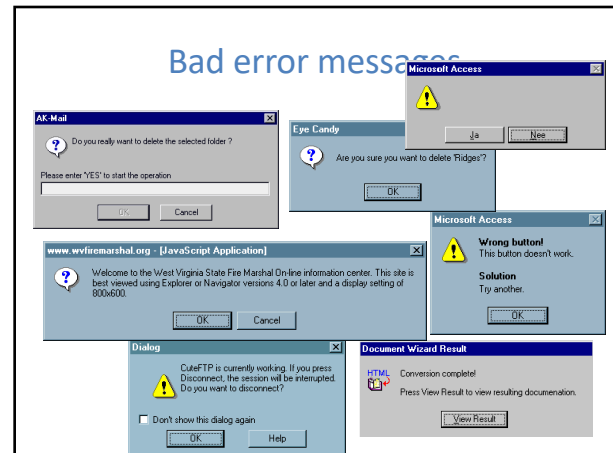
<http://interfaceallofshame.eu> (sadly now defunct)



18



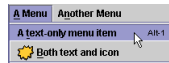
19



20

UI design – buttons, menus

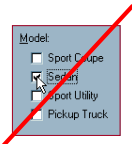
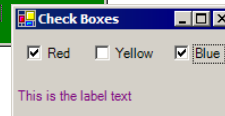
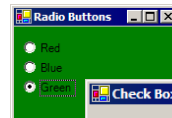
- Use **buttons** for single independent actions that are relevant to the current screen.
 - Try to use button text with verb phrases such as "Save" or "Cancel", not generic: "OK", "Yes", "No"
 - use **Mnemonics** or **Accelerators** (Ctrl-S)
- Use **toolbars** for common actions.
- Use **menus** for infrequent actions that may be applicable to many or all screens.
 - Users hate menus!* Try not to rely too much on menus. Provide another way to access the same functionality (toolbar, hotkey, etc.)



21

UI design – checkboxes, radio buttons

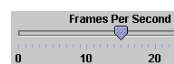
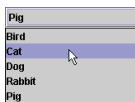
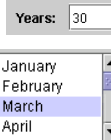
- Use **check boxes** for independent on/off switches
- Use **radio buttons** for related choices, when only one choice can be activated at a time



22

UI design – lists, combo boxes

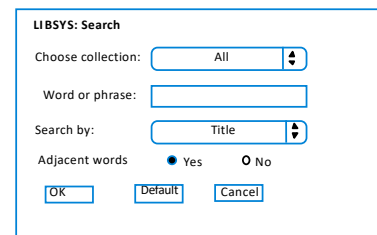
- use **text fields** (usually with a label) when the user may type in anything they want
- use **lists** when there are many fixed choices (too many for radio buttons); *all* choices visible on screen at once
- use **combo boxes** when there are many fixed choices; don't take up screen real estate by showing them all at once
- use a **slider** or **spinner** for a numeric value



23

An example UI

- Good UI dialog? Did the designer choose the right components? assume there are 20 collections and 3 ways to search



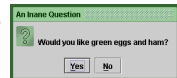
24

UI design – multiple screens

- use a **tabbed pane** when there are many screens that the user may want to switch between at any moment



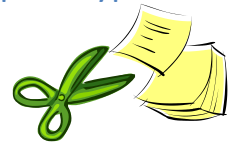
- use **dialog boxes** or **option panes** to present temporary screens or options



25

Creating a paper prototype

- gather materials
 - paper, pencils/pens
 - tape, scissors
 - highlighters, transparencies

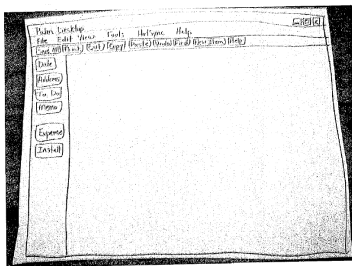


- identify the screens in your UI
 - consider use cases, inputs and outputs to user
- think about how to get from one screen to next
 - this will help choose between tabs, dialogs, etc.

26

Application backgrounds

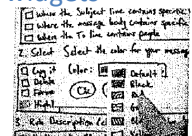
- draw the app background (parts that matter for the prototyping) on its own, then lay the various subscreens on top of it



27

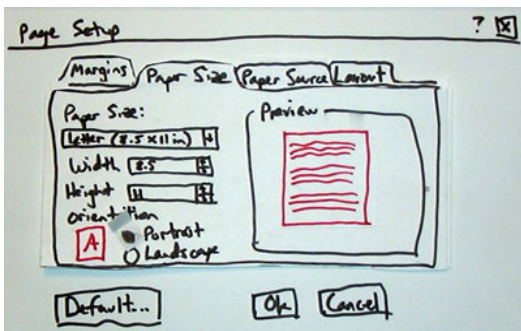
Representing interactive widgets

- buttons / check boxes: tape
- tabs, dialog boxes: index cards
- text fields: removable tape
- combo boxes: put the choices on a separate piece of paper that pops up when they click
- selections: a highlighted piece of tape or transparency
- disabled widgets: make a gray version that can sit on top of the normal enabled version
- computer beeps: say "beep"



28

Example paper prototype screen



29

Prototyping exercise

- In your project groups, draw a rough prototype for a music player (e.g., WinAmp or iTunes).
 - Assume that the program lets you store, organize, and play songs and music videos.
 - Draw the main player UI and whatever widgets are required to do a **search for a song or video**.
 - After the prototypes are done, we'll try walking through each UI together.
- Things to think about:
 - How many clicks are needed? What controls to use?
 - Could your parents figure it out without guidance?

30

30