

Interaction Styles

Interaction

We consider:

- Manipulation styles
- Interface Components
- Examples

Aim is to deepen understanding of the issues involved, and to provide guidance on interface design.

- Interaction Devices (later) is a brief tour of the physical aspects of interaction between humans and computer

Shneiderman, Chapter 5-9

Preece, Chapter 2.5, 6



Type vs Style

Interaction types - what is the aim of the interaction?

- Instructing
- Conversing
- Manipulating
- Exploring

Interaction styles - what mechanism is to be used? E.g.

- Typing
- Speech
- Gesture
- Touch
- Menu

Historical Perspective

First (after the paper tape days), there were command-driven interfaces - and command-driven games...

You Are In Cave: 16

Tunnels Lead To: (18 3 4)

Shoot Or Move ? (S - M) m

Where To ? 3

The very first of these was adventure

(see http://rickadams.org/adventure/a_history.html)

Historical Perspective

An early development was the *form-fill style*

- Clerical workers were used to filling in paper forms

Marina Registration Fill-In Form

The purpose of this form is to collect information about your marina without relying on the post office will still have to be mailed the old fashioned way, but, by using this form we can get your site

Personal Information

Contact Name.....
Street.....
City.....
Prov/State....
PCode/Zip....
Country.....
Phone.....
Fax.....

What is Name of your Marina?

Interface Development: WIMP

Design of games, development of word processing systems, use of computers by non-specialist users, all were driving forces in developing other forms of interaction

Now we have a huge range of interface features

- menus, command menus, question/answer sequences, buttons, sliders, scrollbars, floating toolbars...

Direct vs Indirect Manipulation

Direct manipulation:

- Extension of the real world
- Objects and actions continuously visible
- Actions rapid and incremental (and visible)
- Actions are easily reversible

Indirect manipulation:

- Words and text replace objects and actions
- Typing rather than pointing
- Difficulties with suitable metaphors

Direct Manipulation

Advantages:

- Novices can learn the basic functionality quickly
- Experienced users can perform actions quickly
- Intermittent users are quickly reminded of functionality
- Immediate feedback (is the action helping toward desired goal?)
- More comprehensible, so users feel more secure
- Users feel in control (can predict system responses)

Direct Manipulation

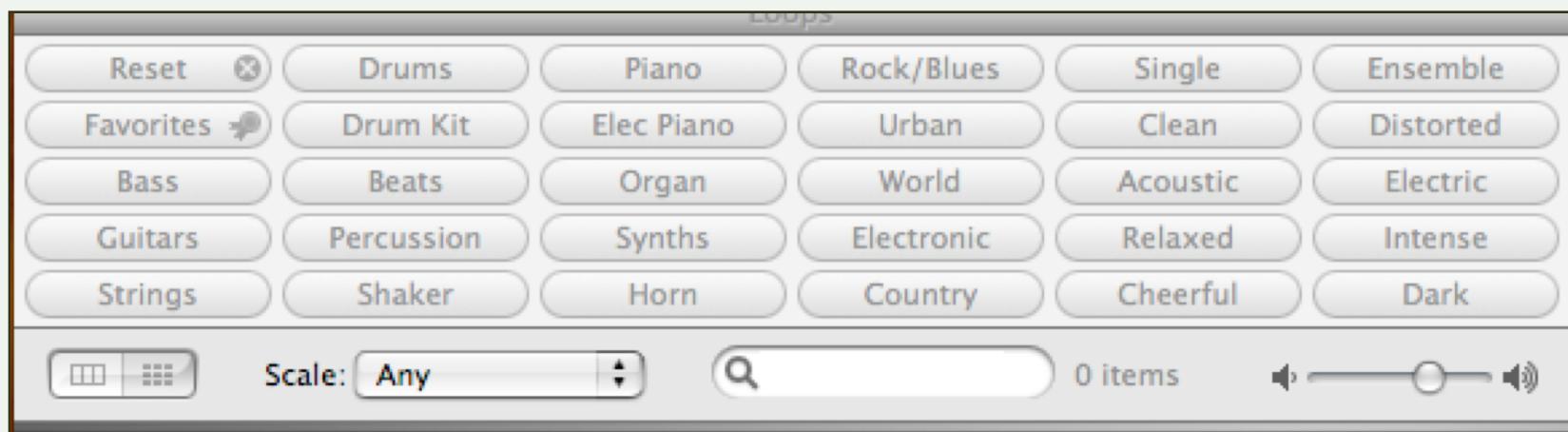
Disadvantages:

- Not all tasks can be described using concrete objects; metaphors will only stretch so far
 - And user needs to understand the metaphor
- Not all actions can be performed directly
- Range of possible actions may be limited by the need for visibility
 - Can be difficult to provide all the options permissible
- Often has issues with handling large numbers of objects

On balance?

- Different users may prefer different ways of doing things.
- Often familiarity is a factor.

Direct Manipulation



...can get a little clumsy when there are many choices!

Command Languages

They provide ways to express precisely what is required

- An example of indirect manipulation
- Commands can be function keys, single characters, short abbreviations, or long words

Examples:

- operating systems languages (Unix, DOS)
- query languages
 - search engines
 - other types of databases

Awkward, unless you know what's going on

Examples: Command Languages

From MSDOS:

```
copy filename1 filename2
```

From UNIX:

```
grep -iR hci *.htm
```

(Search all files with extension ‘htm’ in the current folder and below for the text string ‘hci’, ignoring case)

```
diff -b filename1 filename2
```

(compare filename1 and filename2 contents)

```
diff dir1 dir2
```

(compare directory contents)

From Google's query language

Computer OR Computing OR Computation

Command Languages

Advantages:

- Very powerful! (Wide range of commands possible)
- Experienced knowledgeable users can use short commands fast

Disadvantages:

- Memory required
- Need to know the exact commands
- Novice users feel intimidated

Command Menus

Cross between a command language and menu system

Example from the old Stirling Library System:

```
Enter AUTHOR NAME :  
SO=Start Over, B=Back, ?=Help
```

Example from Lynx:

```
H)elp O)ptions P)rint G)o M)ain screen Q)uit  
/=search [d)elete]=history list
```

Menus

For selecting an item from a specific fixed list

Useful for

- structuring the decision of which command to perform
- reminding the user of terminology
- reducing the possibility of keystroke errors

For other requirements menus are not appropriate, e.g.

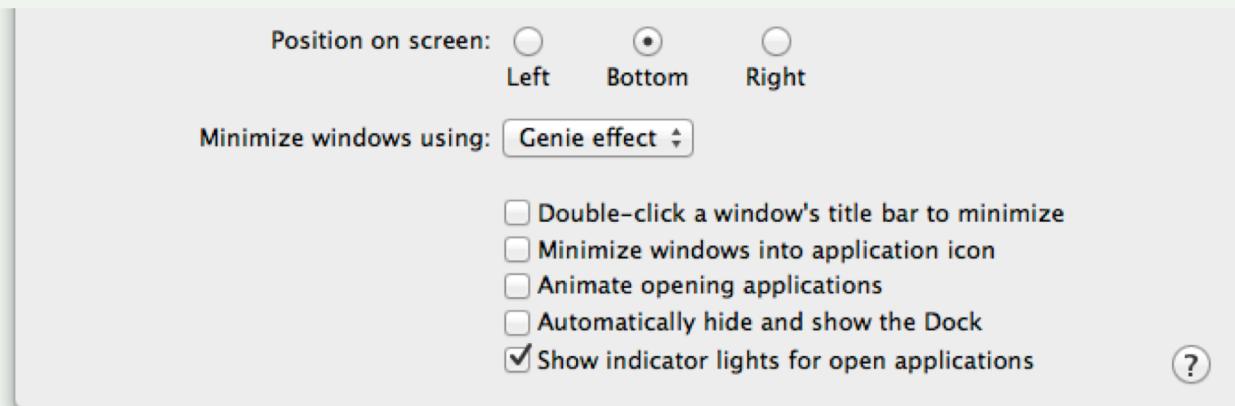
- when the list of options is subject to change
- When the list of options is huge

There are various forms of menu:

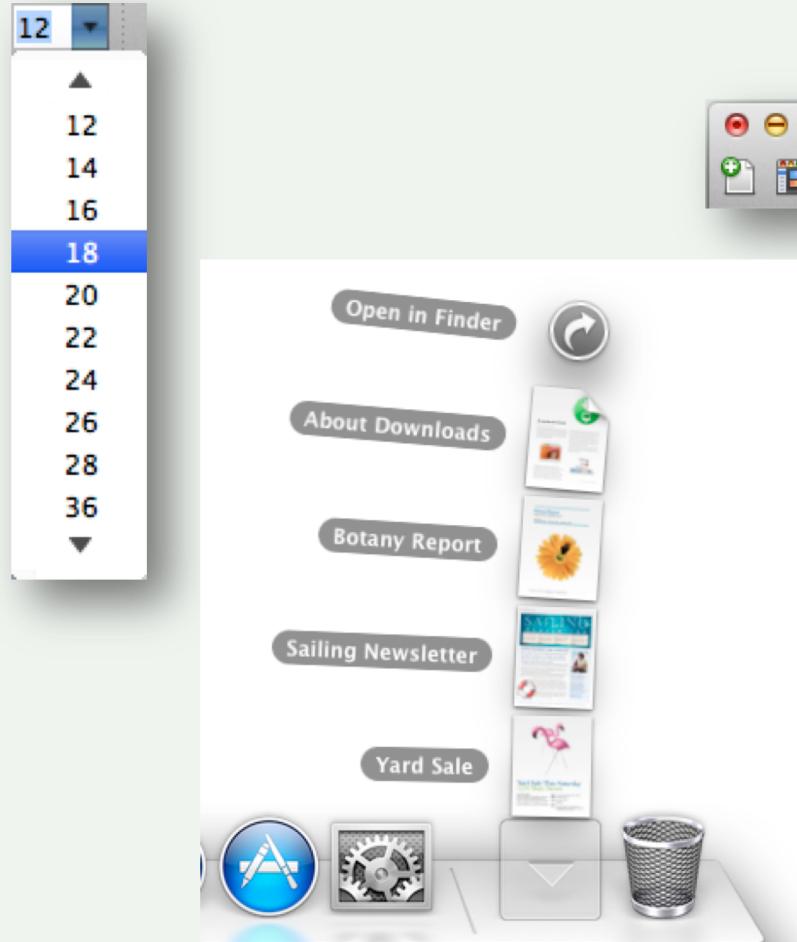
- single, scrolling, iconic, hierarchical

Single Menus

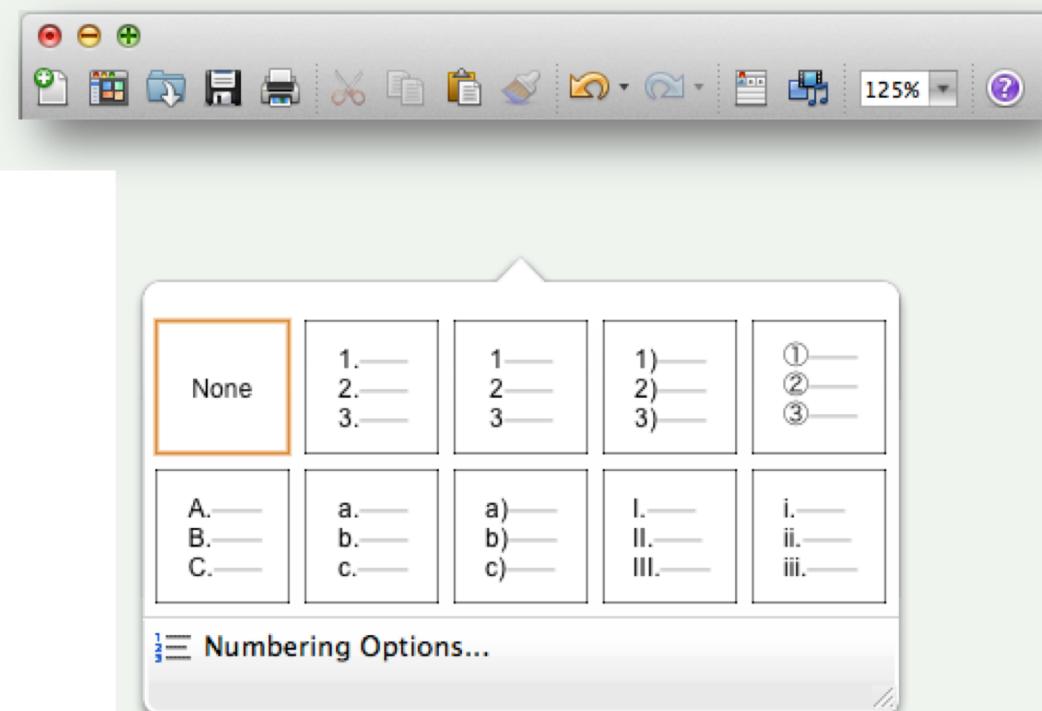
Should have two or more items. May allow just one, or multiple selections (radio buttons versus checkbox).



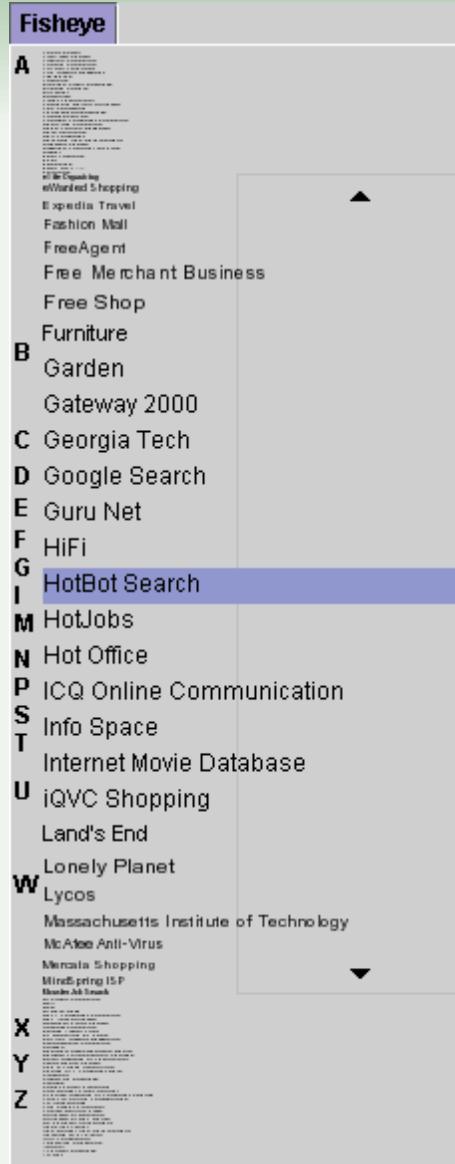
Scrolling Menus



Iconic Menus

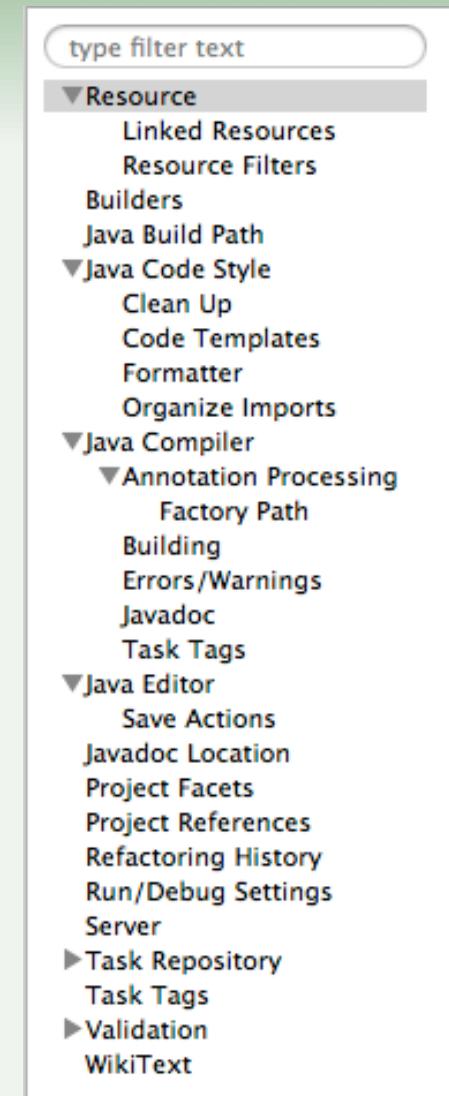






< Fisheye Menus

Hierarchical > Menus



Hierarchical Menus

Structure:

- Depth of a menu is the number of levels
- Breadth of a menu is the number of items per level

There is a trade-off between depth and breadth for maximum ease-of-use

Breadth is generally preferred over depth - users get too confused if they get too deep in a menu

- but too much breadth is not good if it can't be displayed all at once

Hierarchical Menus

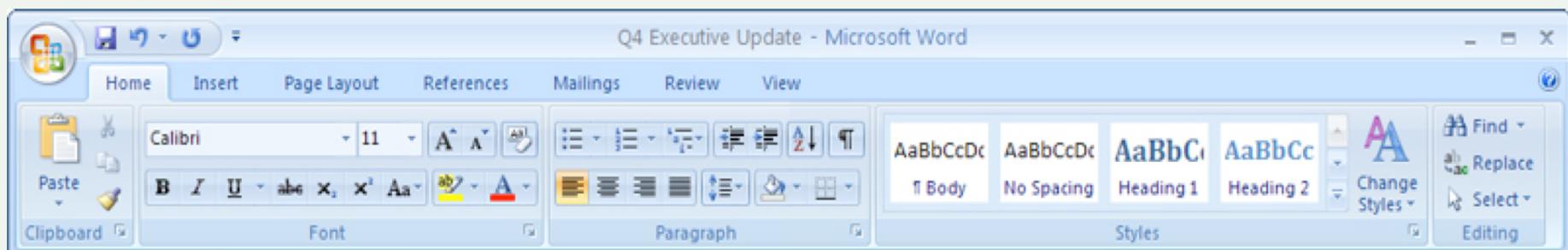
Other considerations:

- Create groups of logically similar items
- Form groups that cover all possibilities (where applicable), e.g. 0-9, 10-19, 20-29 and 30+
- Make sure groups are non-overlapping
 - e.g. **Concerts and Sports** are better labels than **Entertainment and Events**
- Use familiar terminology, but ensure items are clear
 - e.g. Day and Night is less clear than Before 6pm and After 6pm

Hierarchical Menus

The Microsoft Office Ribbon

- Tabs give the first level of functionality
- Panels group functionality within the ribbon



Form Filling

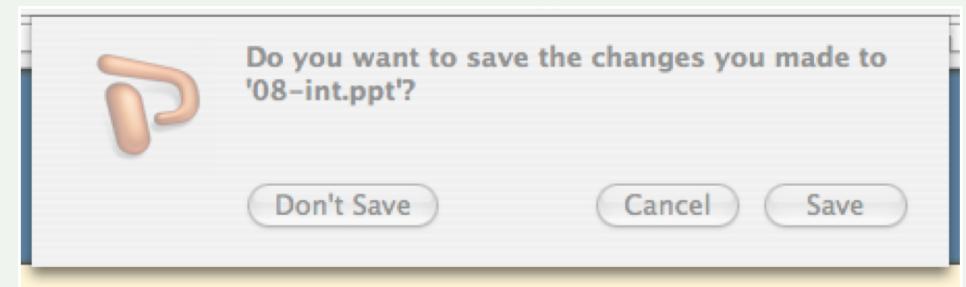
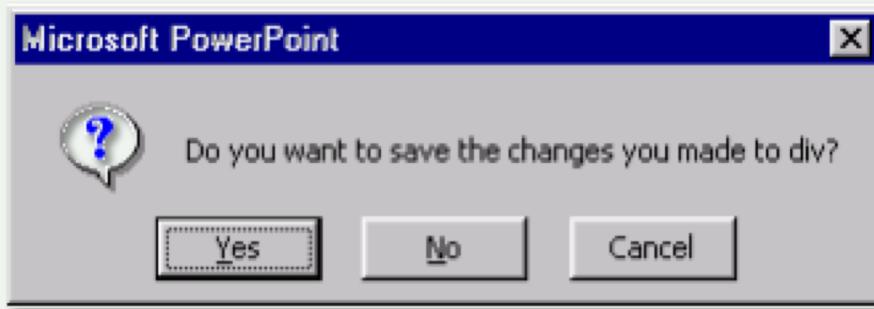
Form-filling is useful for more general data entry, for a specific task, with many pieces of information required. Easy to use - all information is visible, user feels in control of what they type in. Design guidelines include:

- Meaningful title
- Understandable instructions
- Logical arrangement of fields
- Neat layout
- Easy movement between fields
- Error correction/prevention where possible
 - (some fields may be choice boxes)
- Clear indication of compulsory/optional fields

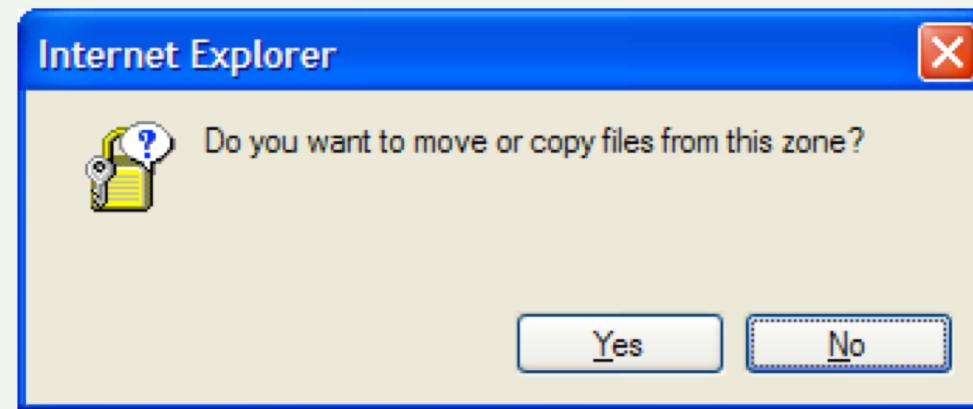
Dialog Boxes

Guidelines include:

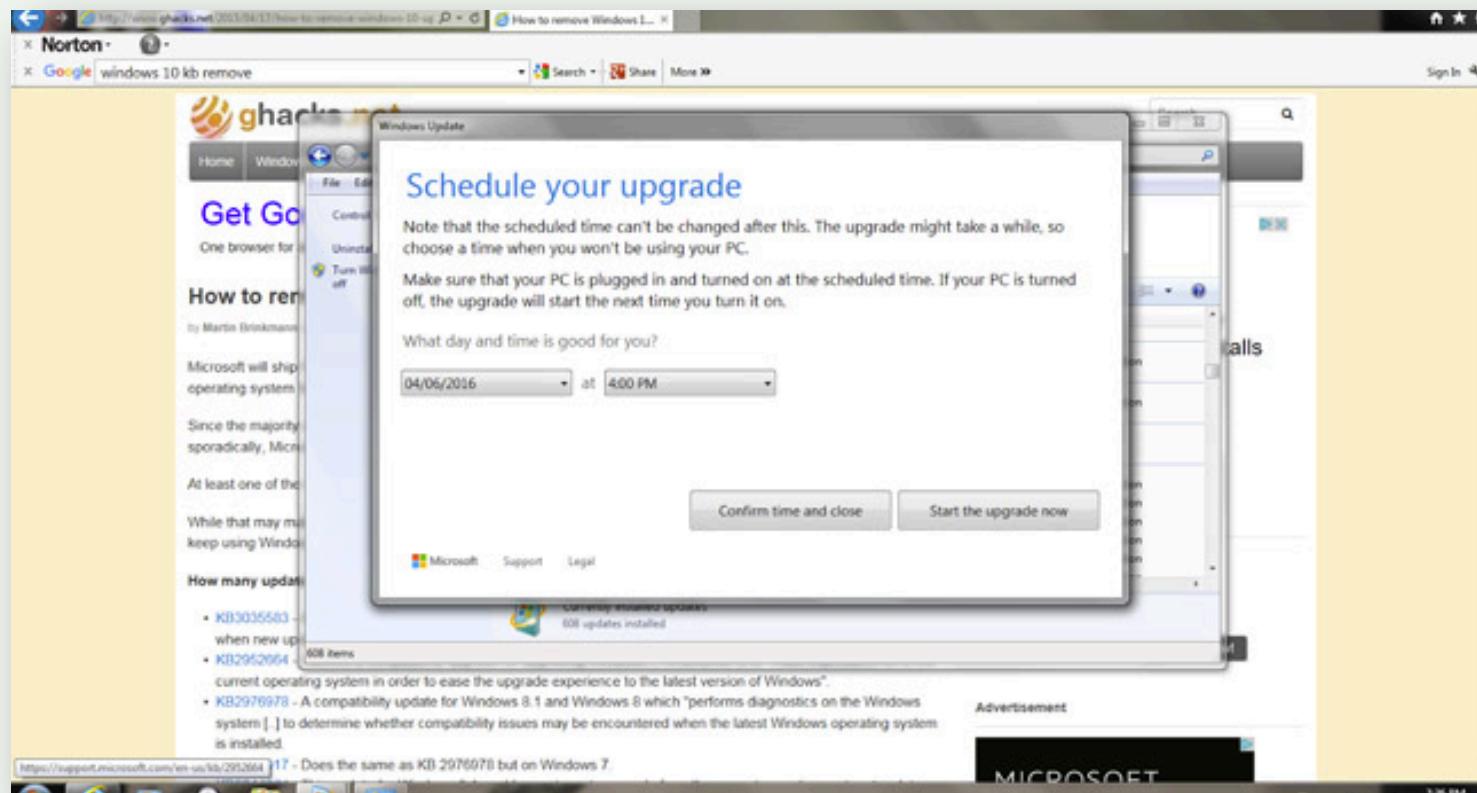
- Meaningful title
- Consistent layout
- Standard buttons (e.g. Ok, Cancel)
- Easy to make disappear
- Clear how to complete or cancel



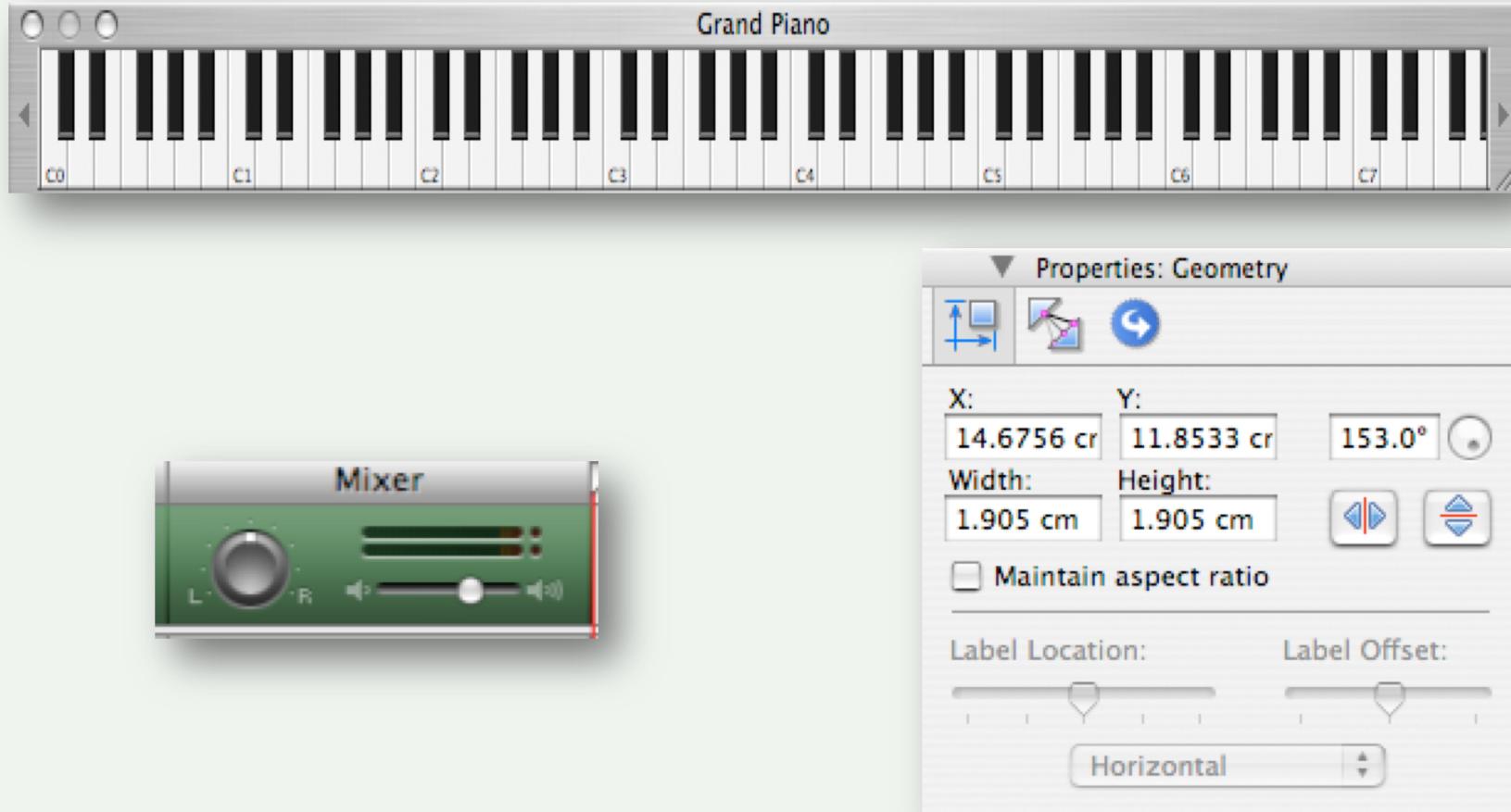
Dialog Boxes



Dialog Boxes



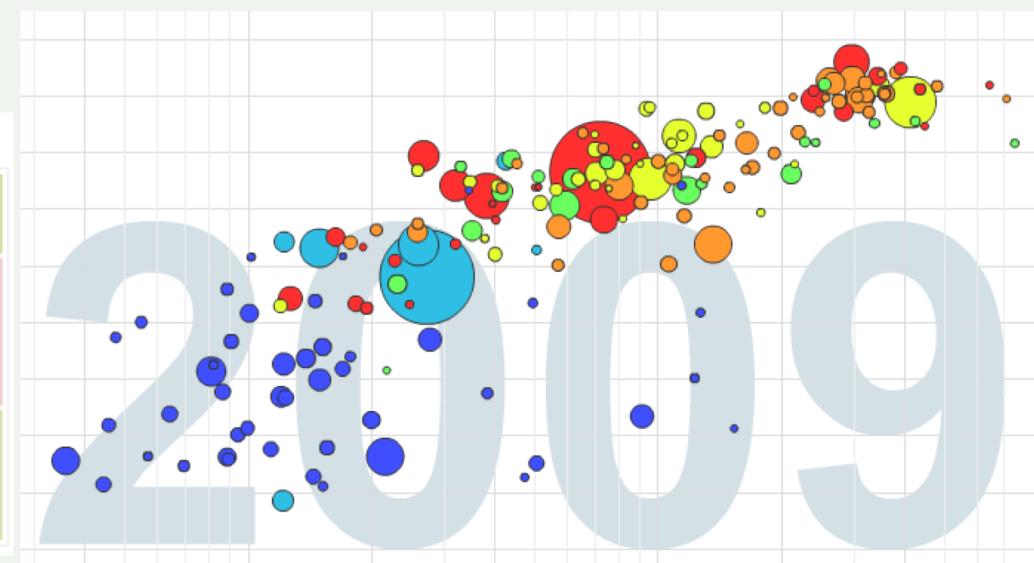
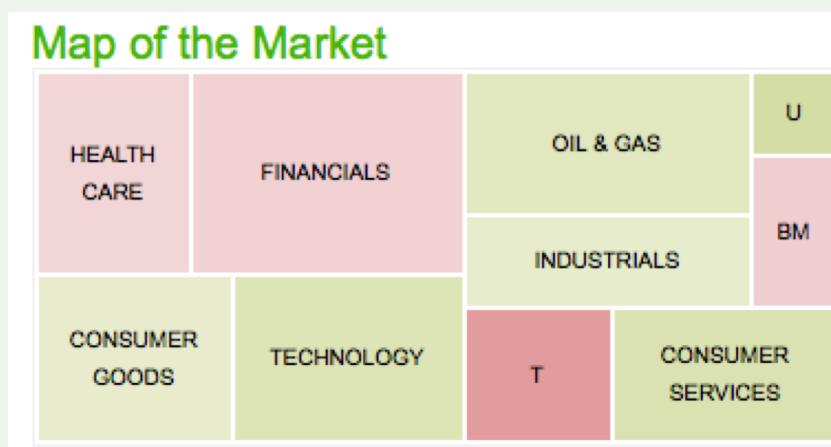
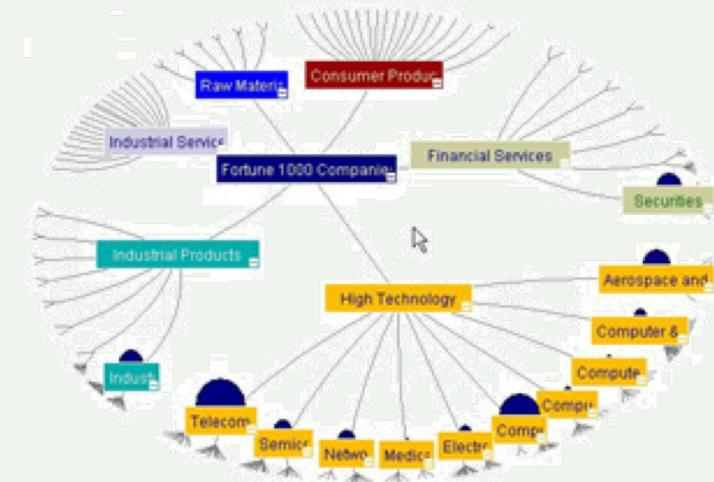
There are other interaction styles as well...



The Future (1)

Visual interfaces for managing vast amounts of information

- Star trees
- <http://www.inxightfedsys.com/products/vizserver/>
- <http://www.gapminder.org/world>
- Google maps



The Future (2)

- Wearable technology
 - huggy top, smart hoodie, kinetic dress, Apple Watch
 - <http://cutecircuit.com/wearables-workshop-at-aho/>
- Other Haptics
 - every phone!
 - cow <http://www.live.ac.uk/our-work/haptic-cow>
- Audio interfaces
 - multiple assistants in this field (Alexa, Siri, Google Home)
- QR codes

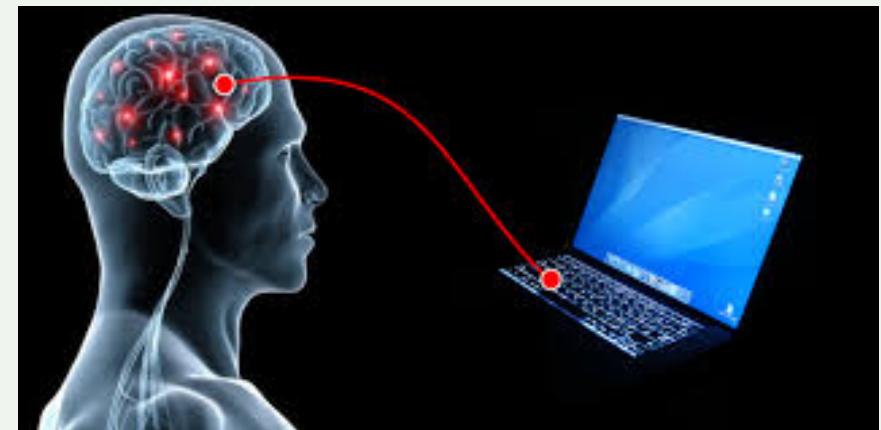


The Future (3)

Google Glass =? Augmented Reality - aimed more at businesses

RFID tags
in cards
in people

Brain Computer Interfaces
becoming more common in prosthetics
- cochlear implants,
thought controlled interfaces?



End of Lecture

Homework: think about interaction styles next time
you sit down at a computer / use your phone