

*Object-Oriented Systems
Development:
Using the Unified Modeling
Language*

**Chapter 12:
View Layer: Designing
Interface Objects**



Goals

- **Identifying View Classes**
- **Designing Interface Objects**
- **Guidelines to Graphical User Interface (GUI)**

... The design of your software's interface, more than anything else, affects how a user interacts and therefore experiences your application.

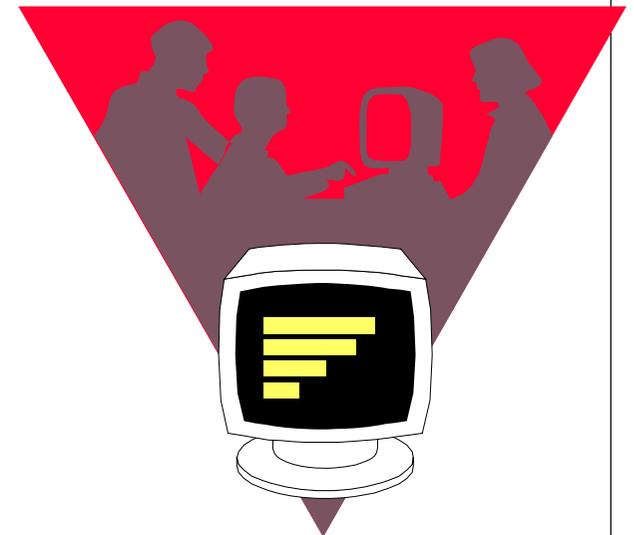
Tandy Trower



Designing View Layer Classes

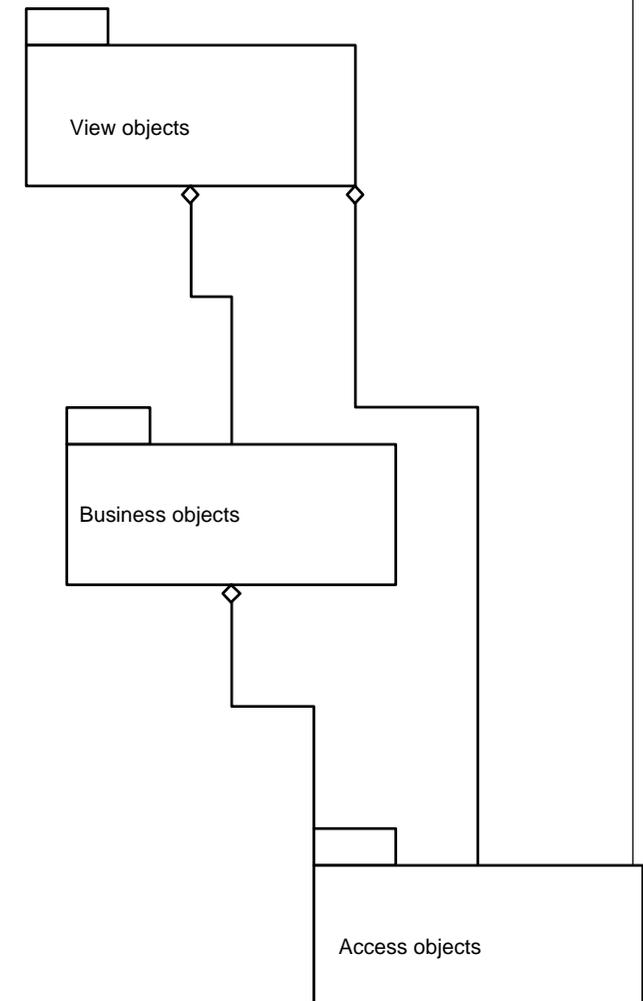
The view layer classes are responsible for two major aspects of the applications:

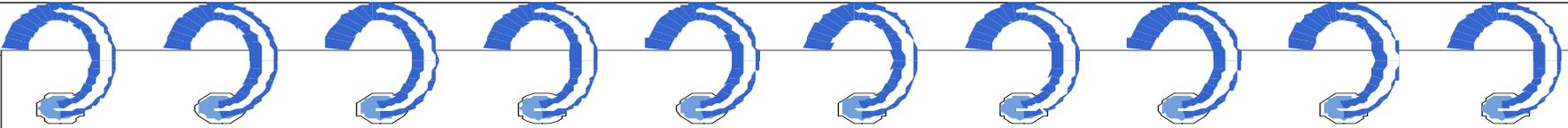
- **Input-Responding to user interaction**
- **Output-Displaying business objects**



Relationships Among Business, Access and View Classes

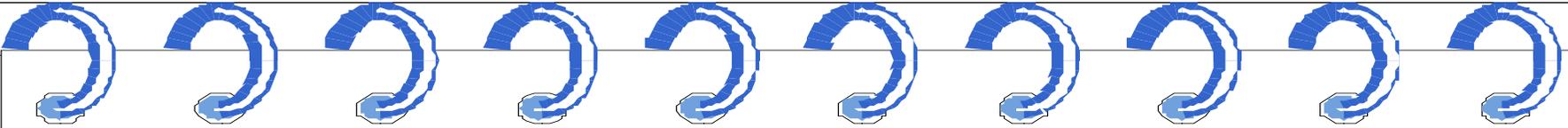
- **In some situations the view class can become a direct aggregate of the access object, as when designing a web interface that must communicate with application/Web server through access objects.**





Designing View Layer Classes (Con't)

- **Design of the view layer classes are divided into the following activities:**
 - **I. Macro Level UI Design Process-
Identifying View Layer Objects.**
 - **II. Micro Level UI Design Activities.**
 - **III. Usability and User Satisfaction
Testing.**
 - **IV. Refine and Iterate.**



View Layer Macro Level

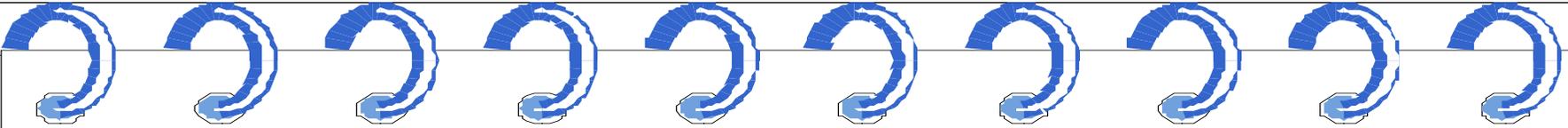
1. For Every Class Identified

1.1 Determine If the Class Interacts With Human Actor: If **yes, do next step **otherwise** move to next class.**

1.1.1 Identified the View (Interface) Objects for The Class.

1.1.2 Define Relationships Among the View (Interface) Objects.

2. Iterate and refine.

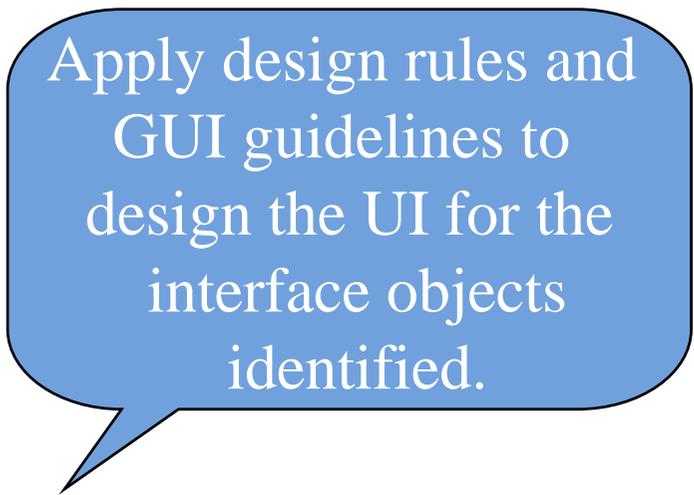


View Layer Micro Level

1. For Every Interface Object Identified in the Macro UI Design Process.

1.1 Apply Micro Level UI Design Rules and Corollaries to Develop the UI.

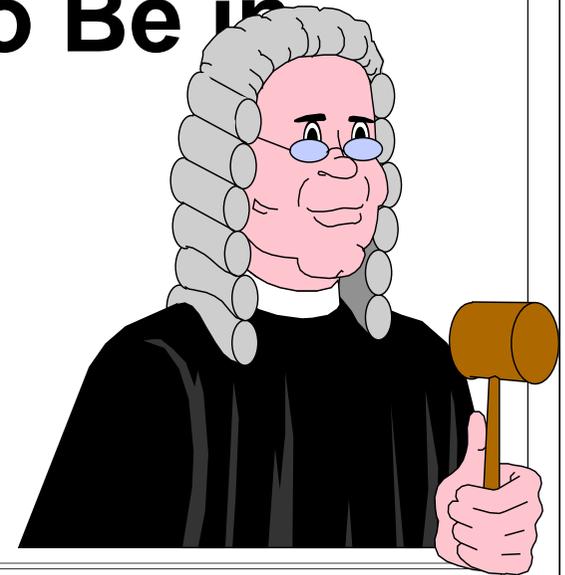
2. Iterate and refine.



Apply design rules and GUI guidelines to design the UI for the interface objects identified.

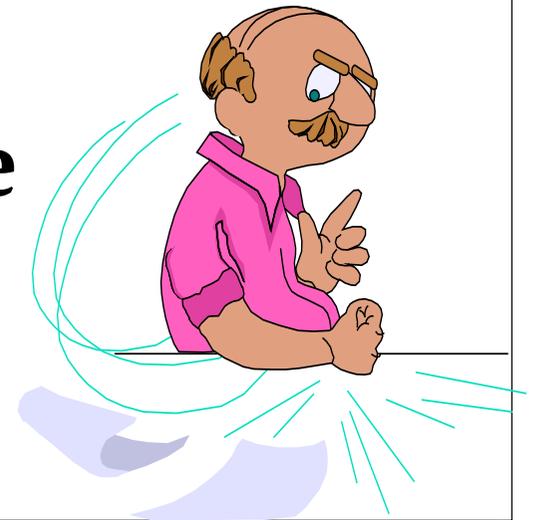
UI Design Rules

- **Rule 1- Making the Interface Simple**
- **Rule 2- Making the Interface Transparent and Natural**
- **Rule 3- Allowing Users to Be in Control of the Software**



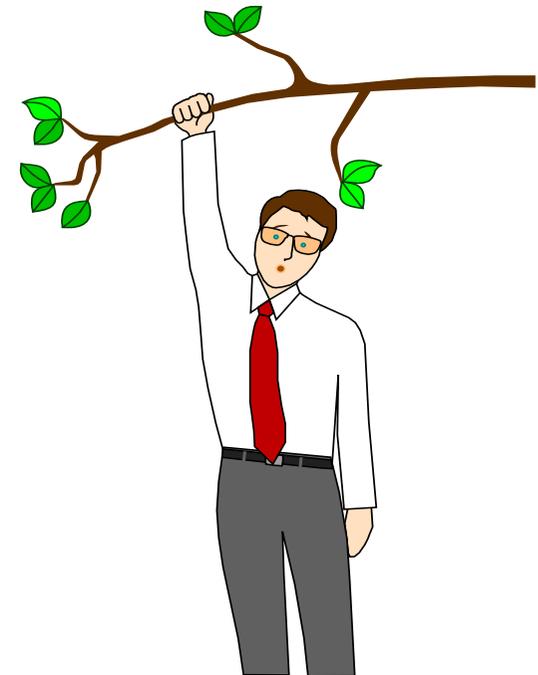
UI Design Rule 1

- Making the interface simple: application of **corollary 2**.
- KISS, Keep It Simple, Stupid.
- Simplicity is different from being simplistic.
- Making something simple requires a good deal of work and code.



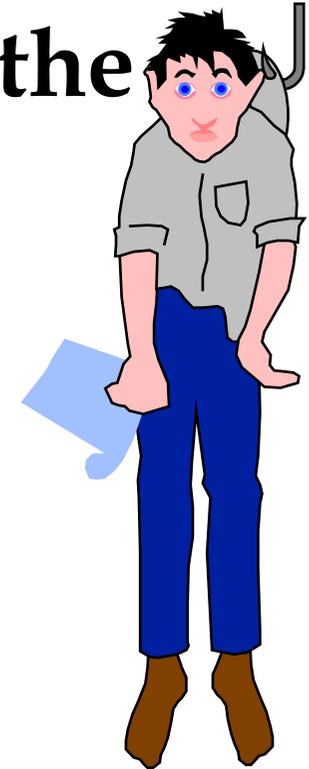
Making The Interface Simple(Con't)

- **Every additional feature potentially affects performance, complexity, stability, maintenance, and support costs of an application.**



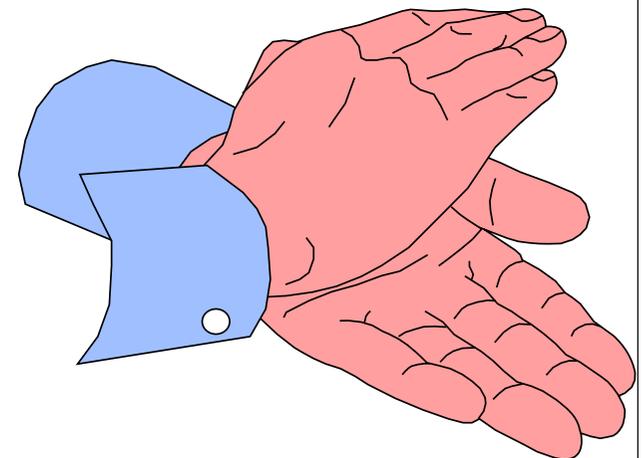
UI Design Rule 1 (Con't)

- A design problem is harder to fix after the release of a product because users may adapt, or even become dependent on, a peculiarity in the design.



UI Design Rule 2

- Making the interface transparent and Natural: application of **corollary 4**.
- Corollary 4 implies that there should be strong mapping between the user's view of doing things and UI classes.



Making The Interface Natural

- **The user interface should be intuitive so users can anticipate what to do next by applying their previous knowledge of doing tasks without a computer.**



Using Metaphors

- **Metaphors can assist the users to transfer their previous knowledge from their work environment to your application interface.**
- **For example, forms that users are accustomed to seeing.**

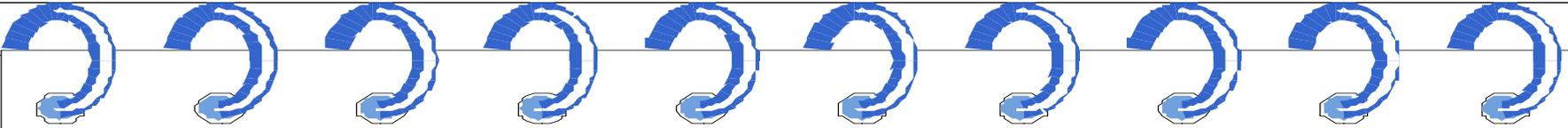


UI Design Rule 3

- **Allowing users to be in control of the software: application of **corollary 1.****

Users should always feel in control of the software, rather than feeling controlled by the software.



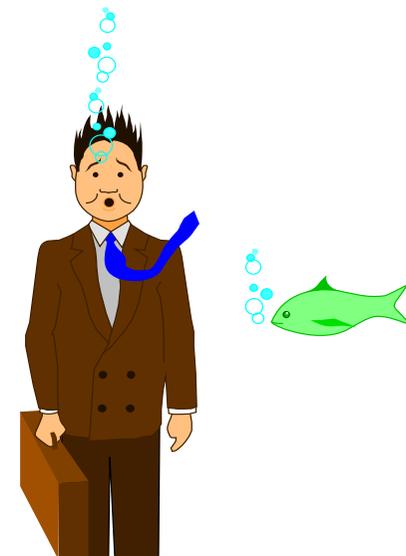


Allowing Users Control of the Software

- **Some of the ways to put users in control are:**
 - Making the interface forgiving.
 - Making the interface visual.
 - Providing immediate feedback.
 - Avoiding Modes.
 - Making the interface consistent.

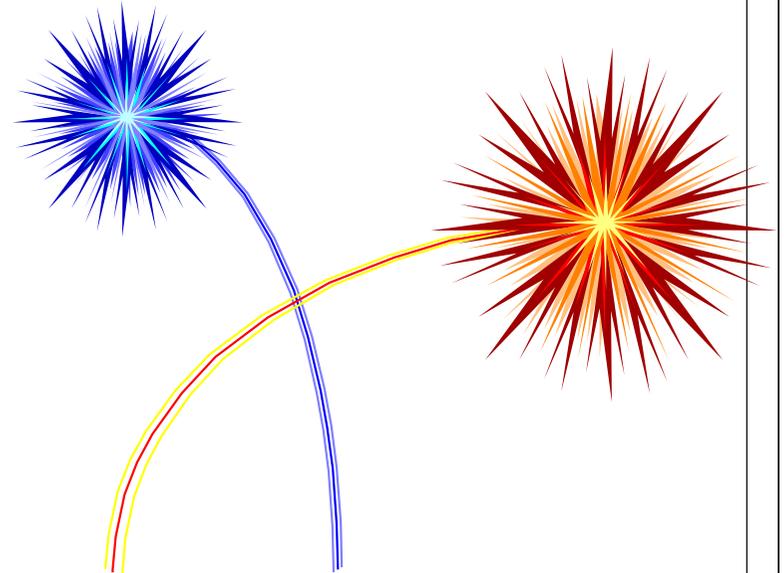
Making the Interface Forgiving

- **Users should be able to back up or undo their previous action.**
- **They should be able to explore without fear of causing an irreversible mistake.**



Making the Interface Visual

- You should make your interface highly visual so users can see, rather than recall, how to proceed.
- Whenever possible, provide users with a list of items from which they can choose.



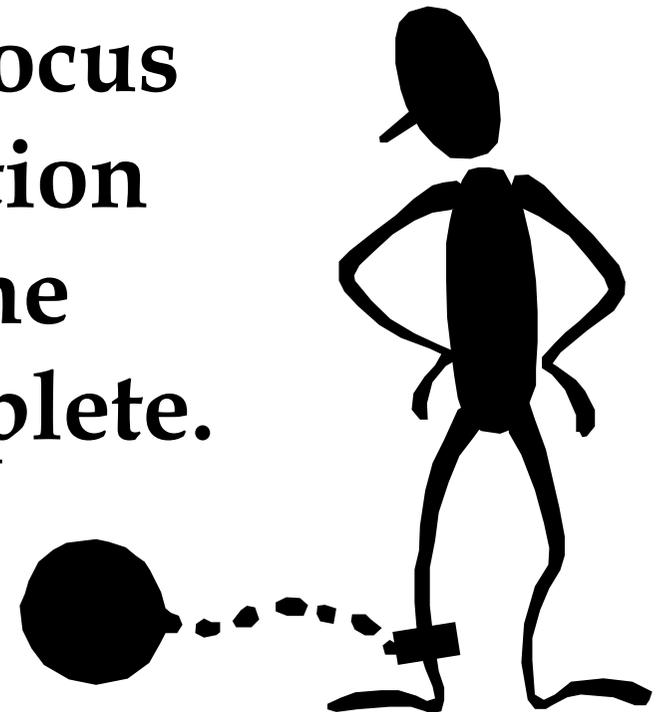
Providing Immediate Feedback

- **Users should never press a key or select an action without receiving immediate visual feedback, audible feedback, or both.**



Avoiding Modes

- Users are in a mode whenever they must cancel what they are doing before they can do something else.
- Modes force users to focus on the way an application works, instead of on the task they want to complete.



Can Modes be useful?

Yes, however:

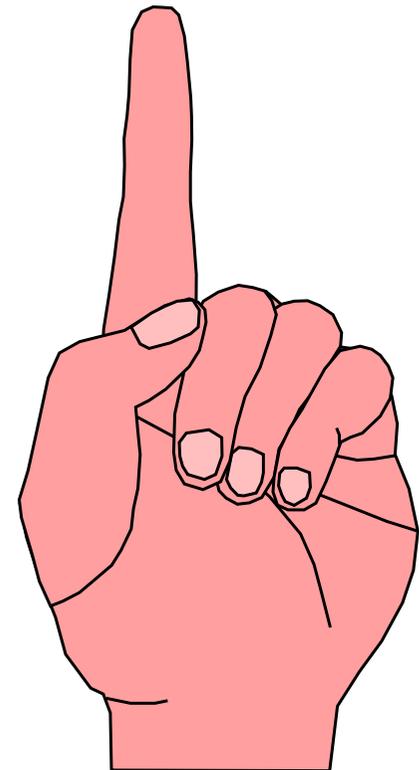
- **You should make modes an exception and limit their use.**
- **Whenever users are in a mode, you should make it obvious by providing good visual cues.**
- **The method for ending the mode should be easy to learn and remember.**



Modes can be useful (Con't)

These are some of the modes that can be used in the user interface.

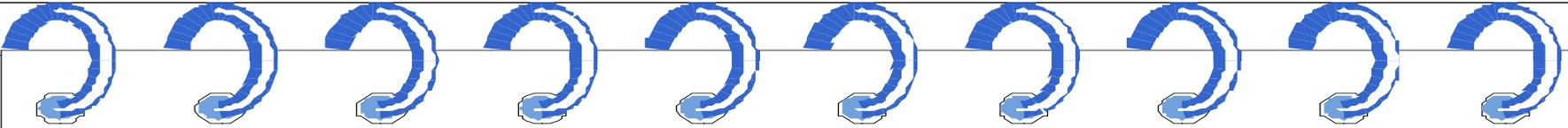
- **Modal Dialog**
- **Spring-Loaded Modes**
- **Tool-Driven Modes**



Making the Interface Consistent

- **User Interfaces should be consistent throughout the applications.**
- **For example, keeping button locations consistent make users feel in control.**



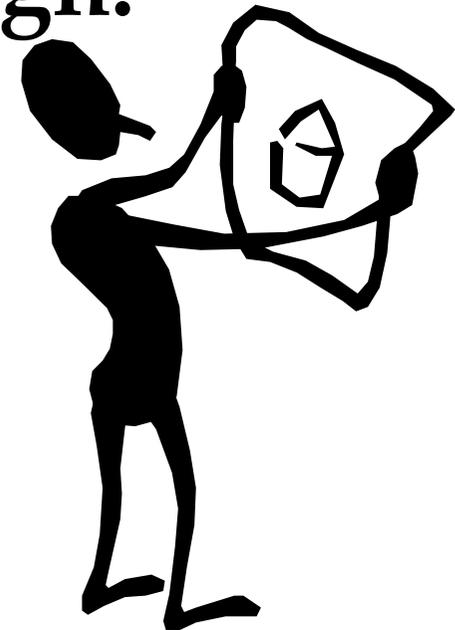


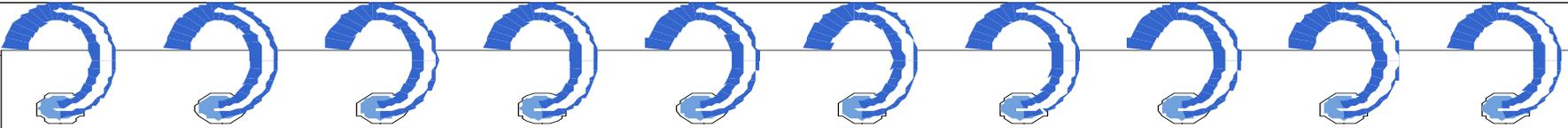
Purpose of a User Interface

- **Data Entry Windows:** Provide access to data that users can retrieve, display, and change in the application.
- **Dialog Boxes:** Display status information or ask users to supply information.
- **Application Windows (Main Windows):** Contain an entire application that users can launch.

Guidelines For Designing Data Entry Windows

- **You can use an existing paper form such as a printed invoice form as the starting point for your design.**





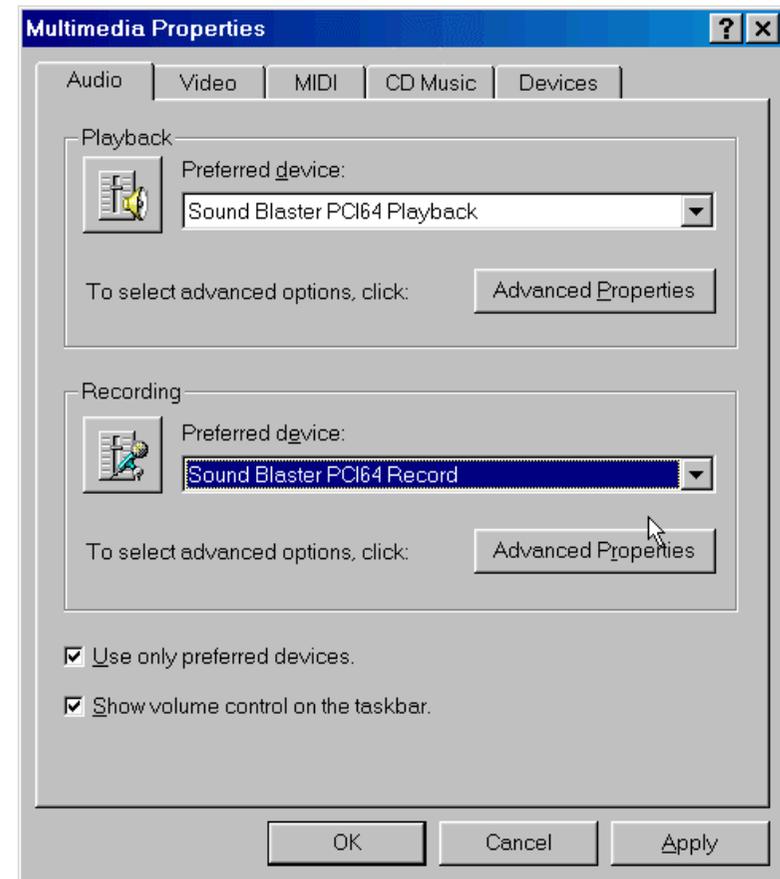
Guidelines For Designing Data Entry Windows (Con't)

If the printed form contains too much information to fit on a screen:

- **Use main window with optional smaller Windows that users can display on demand, or**
- **Use a window with multiple pages.**

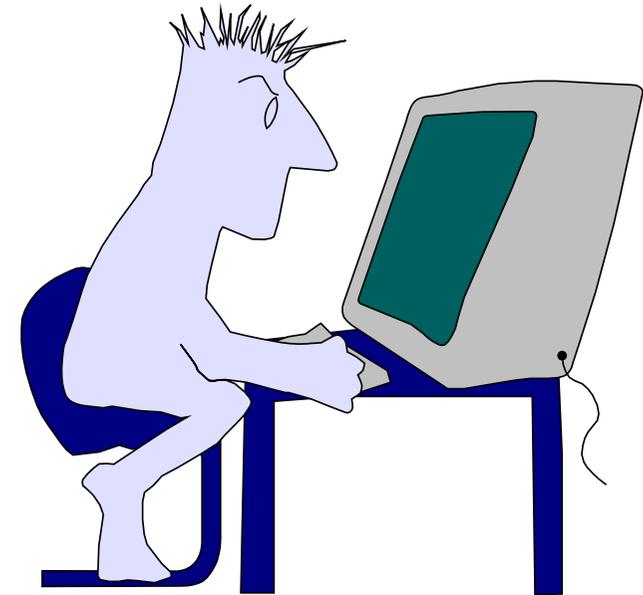
Guidelines For Designing Data Entry Windows (Con't)

- **An example of a dialog box with multiple pages in the Microsoft multimedia setup.**



Guidelines For Designing Data Entry Windows (Con't)

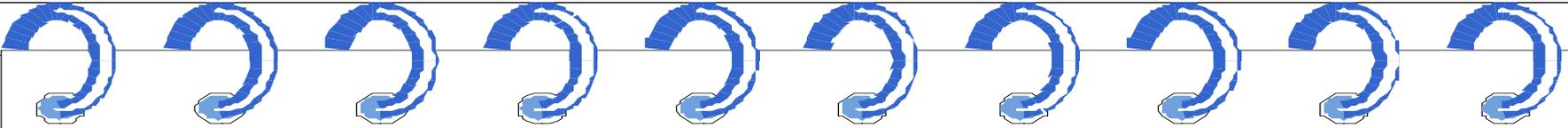
Users scan a screen in the same way they read a page of a book, from left to right, and top to bottom.



Guidelines For Designing Data Entry Windows (Con't)

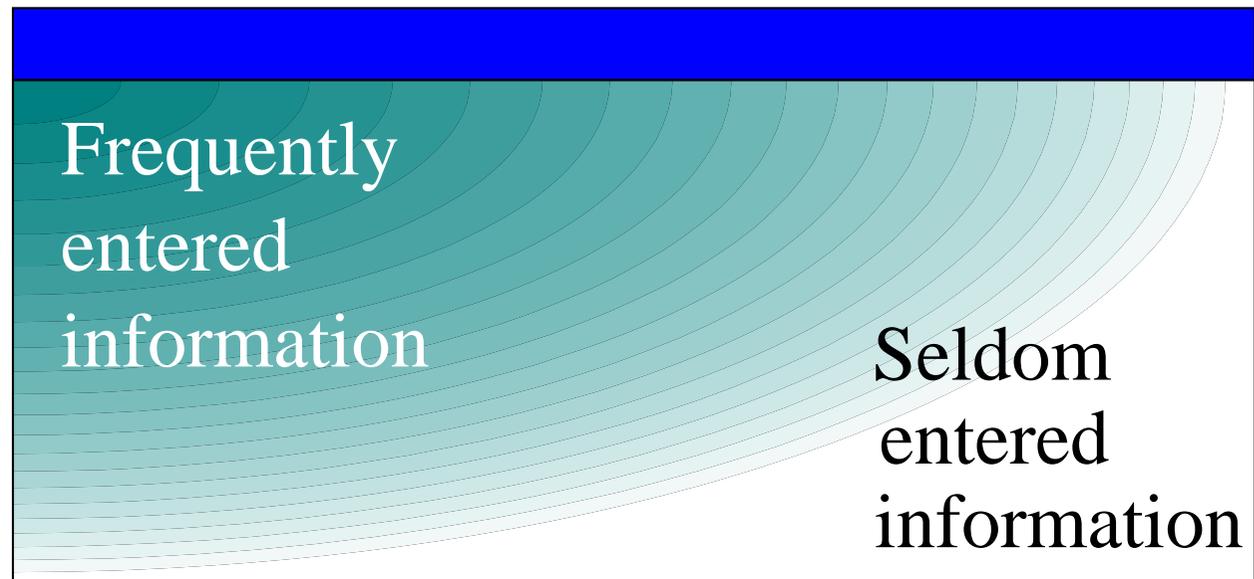
- **Orient the controls in the dialog box in the direction people read.**
- **In the Western world this usually means left to right, top to bottom.**

The image shows a dialog box with a grey background and a dark blue title bar. It contains several text input fields arranged vertically. A yellow line traces a path through the fields from top to bottom, illustrating the reading order: First Name, Address, City, State, Zip Code, OK, and Cancel. The fields are labeled as follows: First Name, Address, City, State, and Zip Code. The OK and Cancel buttons are located at the bottom right of the dialog box.



Guidelines For Designing Data Entry Windows (Con't)

- **Required information should be put toward the top and left side of the form, entering optional or seldom entered information toward the bottom.**

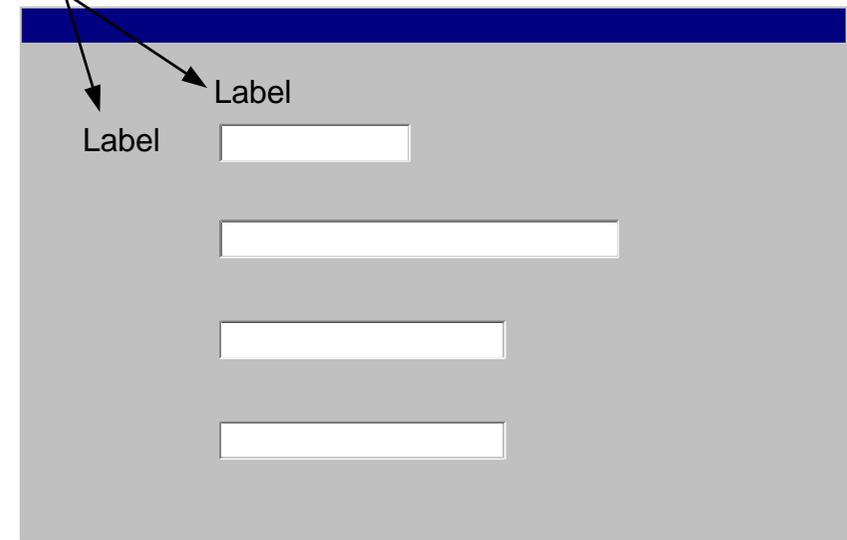


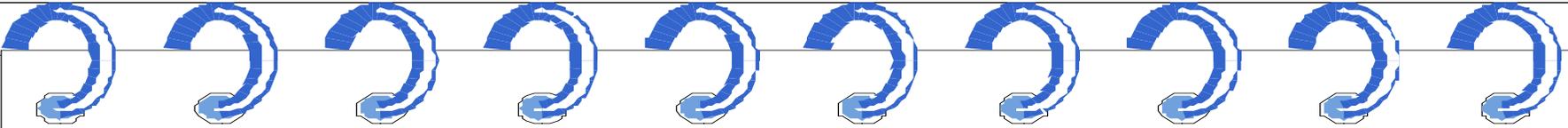
Guidelines For Designing Data Entry Windows (Con't)

- **Place text labels to the left of text box controls, align the height of the text with text displayed in the text box.**

Possible locations for text

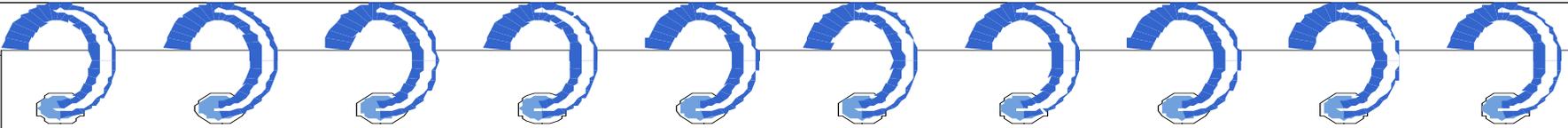
Labels





Guidelines For Designing Dialog Boxes

- If the dialog box is for an error message, use the following guidelines:
- Your error message should be positive.
- For example instead of displaying **“You have typed an illegal date format,”** display this message **“Enter date format mm/dd/yyyy.”**



Guidelines For Designing Dialog Boxes(Con't)

- Your error message should be constructive, brief and meaningful.
- For example, avoid messages such as
“You should know better! Use the OK button”
- instead display
“Press the Undo button and try again.”

Guidelines For The Command Buttons Layout

- Arrange the command buttons either along the upper-right border of the form or dialog box or lined up across the bottom.

The diagram shows a dialog box with a dark blue title bar. The main area is light gray and contains several text input fields: "First Name:", "Last Name:", "Address:", "City:", "State:", and "Zip Code:". On the right side, there is a vertical stack of three buttons: "OK", "Cancel", and "Help". An arrow labeled "Default Button" points to the "OK" button in this stack. At the bottom of the dialog box, there is a horizontal row of three buttons: "OK", "Cancel", and "Help". An arrow labeled "Or" points to the "OK" button in this row.

Buttons Layout (Con't)

- Positioning buttons on the left or center is popular in Web interfaces.

The screenshot displays a web interface for The McGraw-Hill Companies. On the left, a vertical navigation menu lists: home, what's new, corporate news and information, our businesses, news bureau, career opportunities, and FAQ. The main content area features a central grid of buttons for various industries: aviation, BROADCASTING, business & economics, COMMUNICATION, COMPUTERS, CONSTRUCTION, Consumer Products, Education, FINANCIAL MARKETS, Government Solutions, industry, and Medical and Professional. A vertical text label 'Our Businesses' is positioned to the left of this grid. At the bottom, there are two lines of navigation links and a copyright notice.

home

what's new

corporate news and information

our businesses

news bureau

career opportunities

FAQ

Our Businesses

aviation

BROADCASTING

business & economics

COMMUNICATION

COMPUTERS

CONSTRUCTION

Consumer Products

Education

FINANCIAL MARKETS

Government Solutions

industry

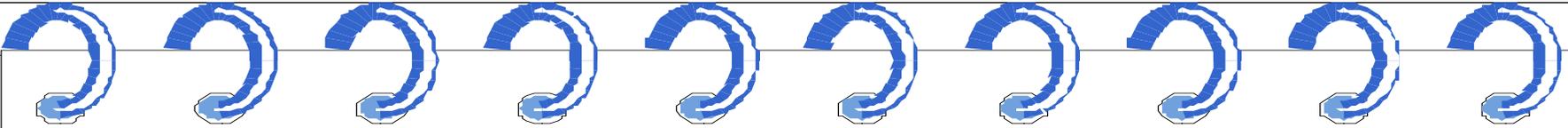
Medical and Professional

The McGraw-Hill Companies

[Aviation](#) | [Broadcasting](#) | [Business & Economics](#) | [Computers/Communication](#) | [Construction](#) | [Consumer Products](#) | [Education](#) | [Financial Markets](#) | [Government Solutions](#) | [Industry](#) | [Medical & Professional](#)

[Home](#) | [What's New](#) | [Corporate News and Information](#) | [Our Businesses](#) | [Privacy Policy](#) | [Books](#) | [Career Opportunities](#) | [Directory](#) | [News Bureau](#) | [Contact Us](#) | [Terms of Use](#)

Copyright © 1998 The McGraw-Hill Companies, Inc.

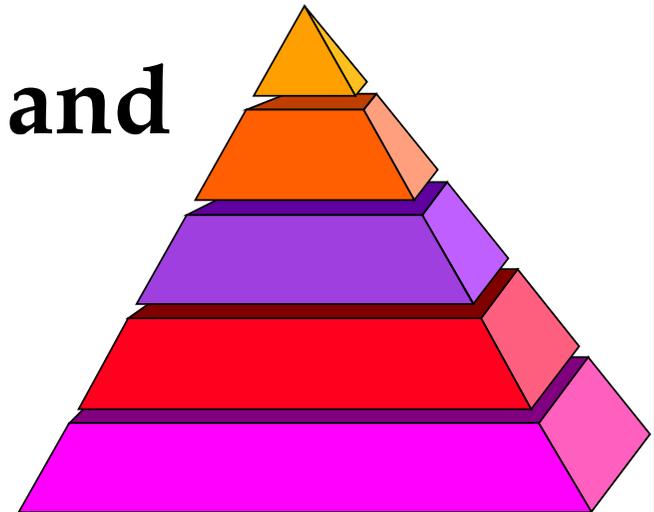


Guidelines For Designing Application Windows

- **A typical application window consists of a frame (or border) which defines its extent:**
- **title bar**
- **scroll bars**
- **menu bars,**
- **toolbars, and**
- **status bars.**

Guidelines For Using Colors

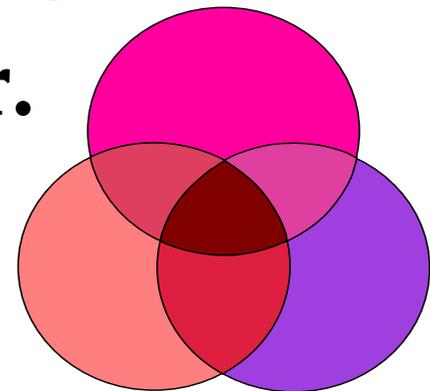
- Use identical or similar colors to indicate related information.
- Use different colors to distinguish groups of information from each other.
- For example, checkout and in-stock tapes could appear in different colors.

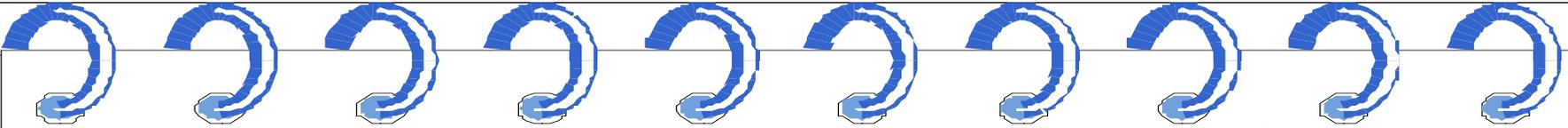


Guidelines For Using Colors

(Con't)

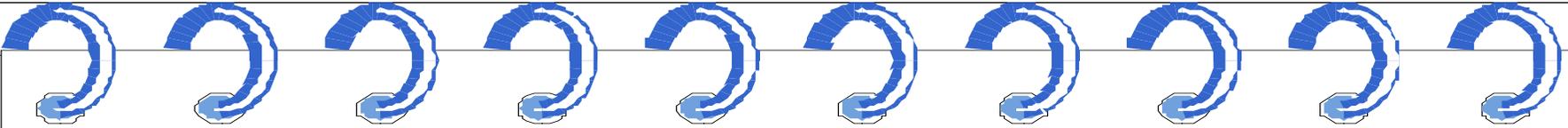
- For an object background, use a contrasting but complementary color.
- For example, in an entry field, make sure that the background color contrasts with the data color.





Guidelines For Using Colors *(Con't)*

- **Use bright colors to call attention to certain elements on the screen.**
- **Use dim colors to make other elements less noticeable.**
- **For example, you might want to display the required field in a brighter color than optional fields.**



Guidelines For Using Colors (Con't)

- **Use colors consistently within each window and among all Windows in your application.**
- **For example the colors for Pushbuttons should be the same throughout.**

Guidelines For Using Colors

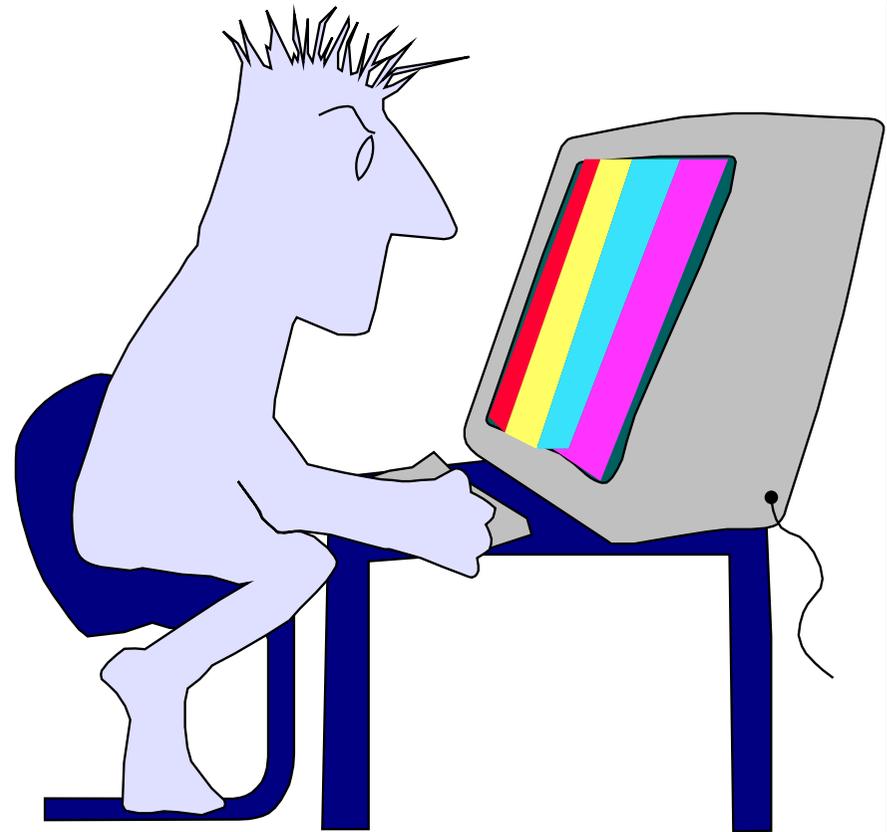
(Con't)

- Using too many colors can be visually distracting, and will make your application less interesting.



Guidelines For Using Colors (Con't)

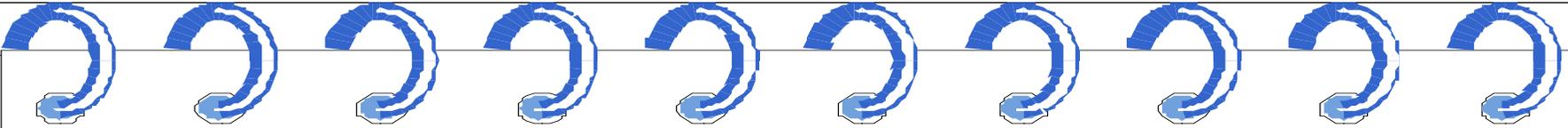
- **Allow the user to modify the color configuration of your application.**



Guidelines For Using Fonts

- **Use commonly installed fonts, not specialized fonts that users might not have on their machines.**
- **Use bold for control labels so they will remain legible when the object is dimmed.**

A B C



Guidelines For Using Fonts (Con't)

- **Use fonts consistently within each form and among all forms in your application.**
- **For example, the fonts for check box controls should be the same throughout.**
- **Consistency is reassuring to users, and psychologically makes users feel in control.**

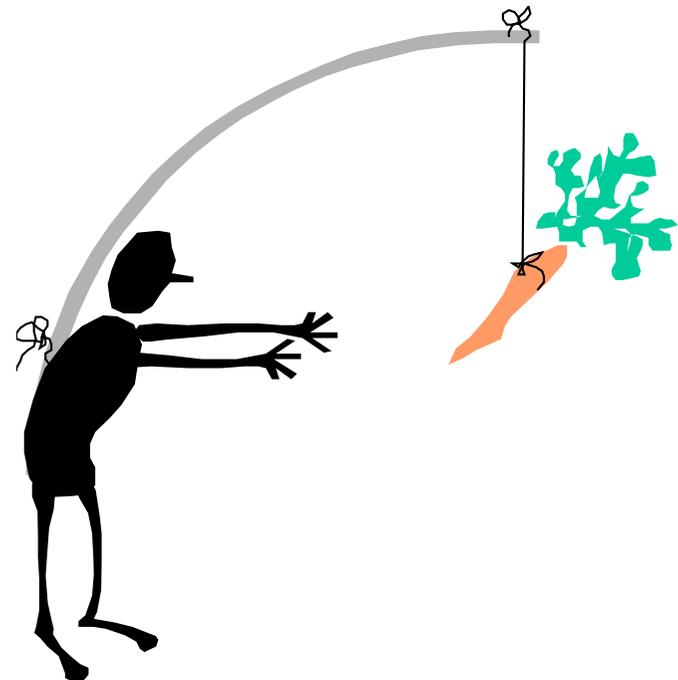
Guidelines For Using Fonts (Con't)

- **Using too many font styles, sizes and colors can be visually distracting and should be avoided.**



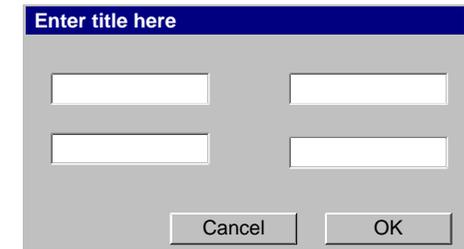
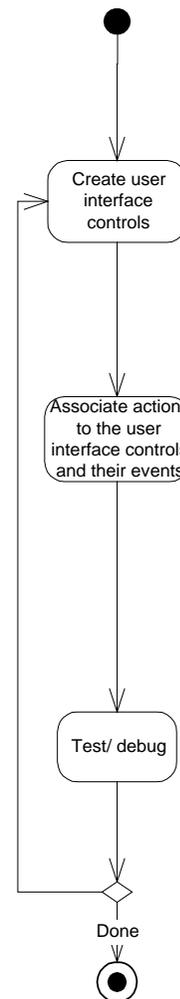
Prototyping the User Interface

- Rapid prototyping encourages the incremental development approach, **“grow, don’t build.”**

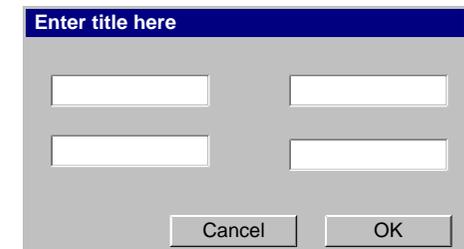


Three General Steps

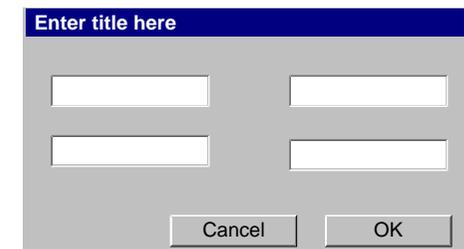
- 1. Create the user interface objects visually.
- 2. Link or assign the appropriate behaviors or actions to these user interface objects and their events.
- 3. Test, debug, then add more by going back to step 1.



Create the forms and controls



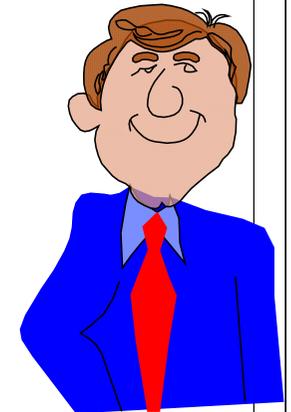
Add actions



Test the UI

Make Users Feel in Charge

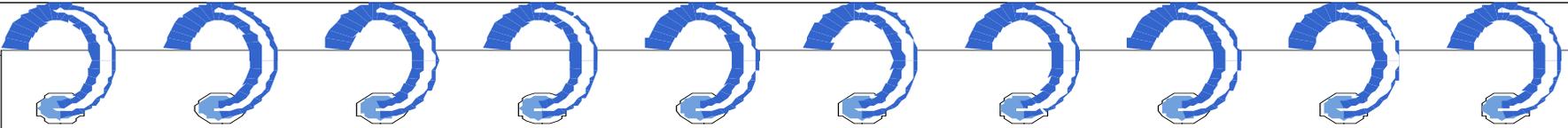
- Instead of using leading phrases like, "we could do this ..." or "It would be easier if we ..."
- Choose phrases that give the user the feeling that he/she is in charge:
 - “Do you think that if we did ... it would make it easier for the users?”
 - “Do users ever complain about ...? We could add .. to make it easier.”



Summary

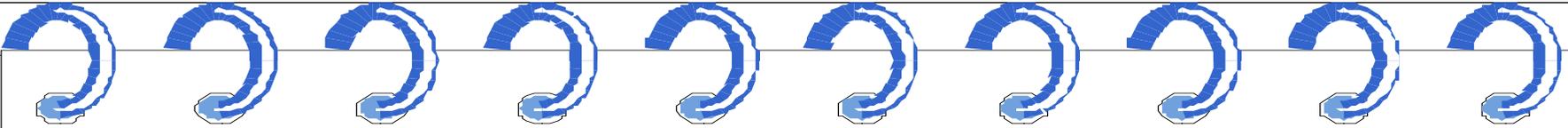
- **The main goal of UI is to display and obtain information you need in an accessible, efficient manner.**
- **The design of your software's interface, more than anything else, affects how a user interacts and therefore experiences your application.**





Summary (Con't)

- **UI must provide users with the information they need and clearly tell them what they need to successfully complete a task.**
- **A well-designed UI has visual appeal that motivates users to use your application.**
- **UI should use limited screen space efficiently.**



Summary (Con't)

- **Designing View layer classes consists of the following steps:**

I. Macro Level UI Design Process- Identifying View Layer Objects

II. Micro Level UI Design Activities

II.1 Designing the View Layer Objects by applying Design Axioms and corollaries .

II. 2 Prototyping the View Layer Interface.

III. Usability and User Satisfaction Testing

IV. Refine and Iterate

Summary (Con't)

- **Guidelines are not a standalone tool, and they cannot substitute for effective evaluation and iterative refinement within a design.**
- **However, they can provide helpful advice during the design process.**

