Prototyping & UI Software

John Stasko

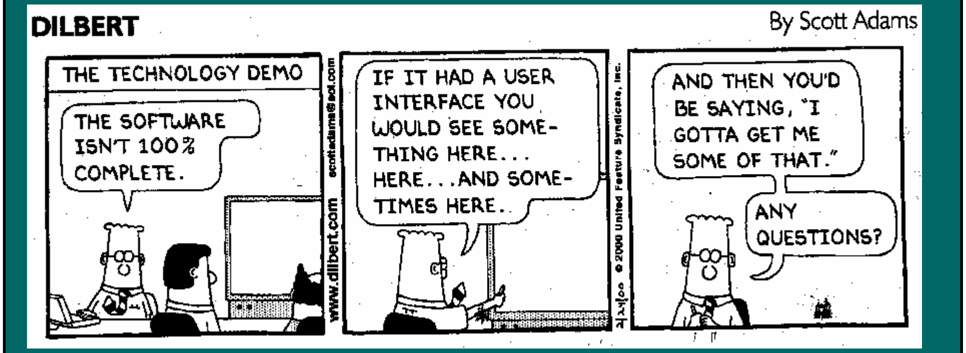
Spring 2007

This material has been developed by Georgia Tech HCI faculty, and continues to evolve. Contributors include Gregory Abowd, Al Badre, Jim Foley, Elizabeth Mynatt, Jeff Pierce, Colin Potts, Chris Shaw, John Stasko, and Bruce Walker. Permission is granted to use with acknowledgement for non-profit purposes. Last revision: January 2007.

Agenda

- Prototyping
  - Dimensions and terminology
  - Non-computer methods
  - Computer methods
- UI Software
  - Design tools
  - UI toolkits
  - GUI builder tools
- Poster session preview
- Exam preview

Your Project Group


DILBERT
By Scott Adams
THE TECHNOLOGY DEMO
THE SOFTWARE ISN'T 100% COMPLETE.
scottadams@aol.com
www.dilbert.com
IF IT HAD A USER INTERFACE YOU WOULD SEE SOMETHING HERE... HERE...AND SOMETIMES HERE.
© 2000 United Feature Syndicate, Inc.
AND THEN YOU'D BE SAYING, "I GOTTA GET ME SOME OF THAT."
ANY QUESTIONS?

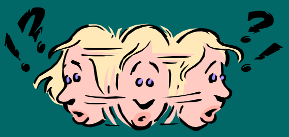
Design Artifacts

- How do we express early design ideas?
  - No software coding at this stage
- Key notions
  - Make it fast!!!
  - Allow lots of flexibility for radically different designs
  - Make it cheap
  - Promote valuable feedback

***** Facilitate iterative design and evaluation *****

Dilemma

- You can’t evaluate design until it’s built
  - But…
- After building, changes to the design are difficult



- Simulate the design, in low-cost manner

Prototyping Dimensions

- 1. Representation
  - How is the design depicted or represented?
  - Can be just textual description or can be visuals and diagrams
- 2. Scope
  - Is is just the interface (mock-up) or does it include some computational component?

Dimensions (contd)

- 3. Executability
  - Can the prototype be “run”?
  - If coding, there will be periods when it can’t
- 4. Maturation
  - What are the stages of the product as it comes along?

Revolutionary - Throw out old one
Evolutionary - Keep changing previous design

Terminology (1)

- Early prototyping

- Late prototyping

Terminology (2)

- Low-fidelity prototype

  Far from final form of system, such as paper, drawings, etc.

- High-fidelity prototype

  Close to final form of system, much more realistic to actual application

Terminology

- Horizontal prototype

  Very broad, does or shows much of the interface, but does this in a shallow manner

- Vertical prototype

  Fewer features or aspects of the interface simulated, but done in great detail

Rapid Prototyping Methods

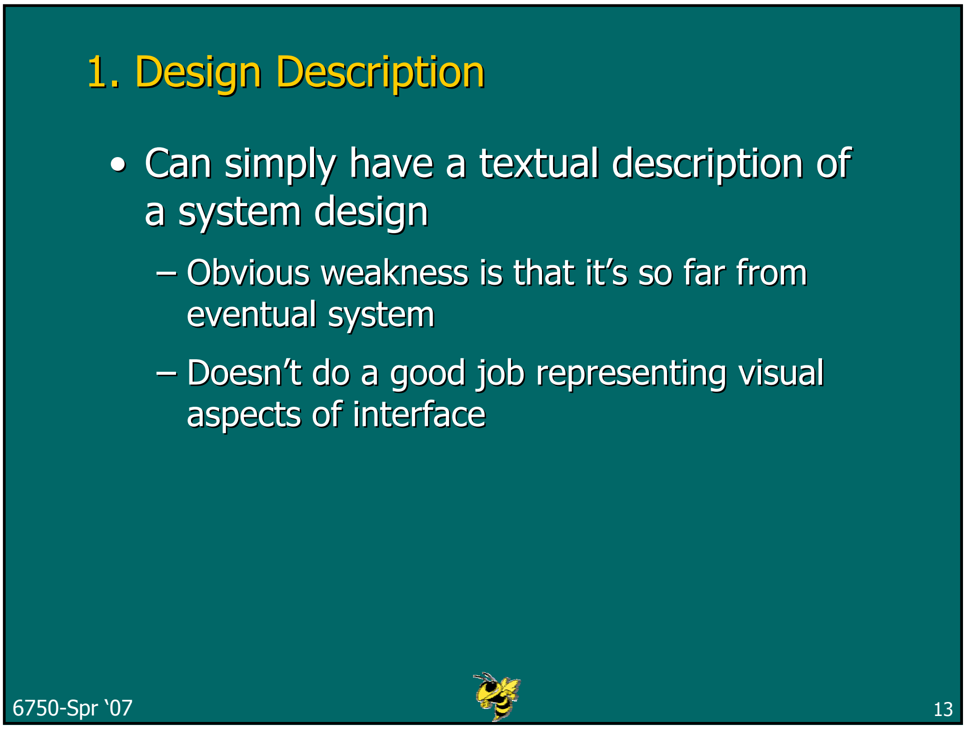
- Non-computer vs. computer-based

Typically earlier in process

Typically later in process

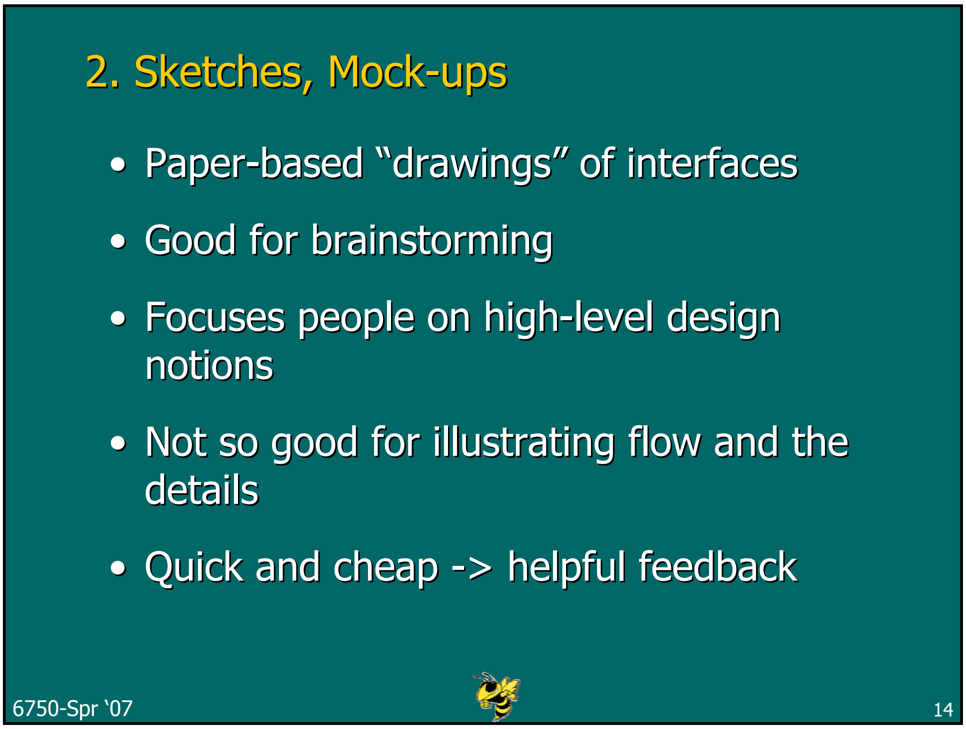
Non-Computer Methods

- Goal: Want to express design ideas and get quick & cheap opinions on system
- Methods?



1. Design Description

- Can simply have a textual description of a system design
  - Obvious weakness is that it's so far from eventual system
  - Doesn't do a good job representing visual aspects of interface

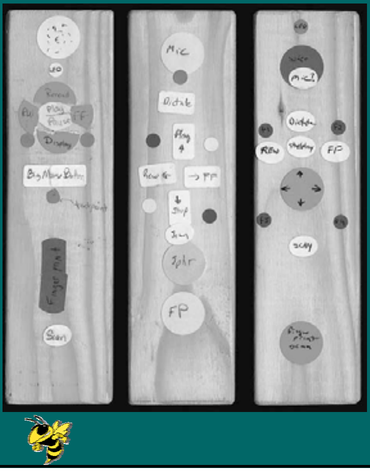


2. Sketches, Mock-ups

- Paper-based "drawings" of interfaces
- Good for brainstorming
- Focuses people on high-level design notions
- Not so good for illustrating flow and the details
- Quick and cheap -> helpful feedback

Physical Mock-Ups

- Wooden blocks and labels - device control

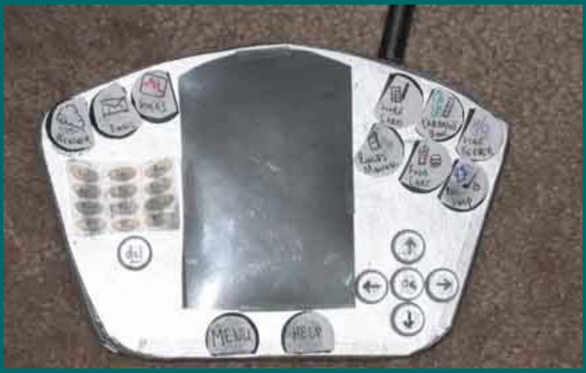


(Three versions of

a hand-held controller)

Physical Mock-Up

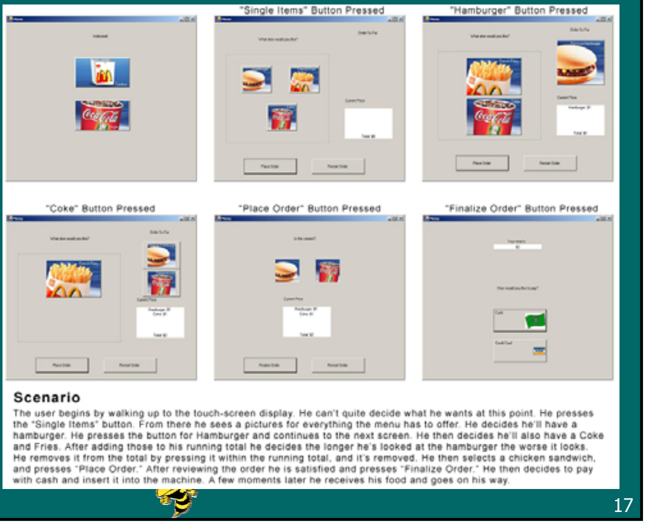
- Styrofoam and Buttons



Spring 2004 CS 4750 project “Golf Caddy” by:
Chris Hamilton
Linda Kang
Luigi Montanez
Ben Tomassetti

3. Storyboard

- What is it?


"Single Items" Button Pressed
"Hamburger" Button Pressed
"Coke" Button Pressed
"Place Order" Button Pressed
"Finalize Order" Button Pressed

Scenario

The user begins by walking up to the touch-screen display. He can't quite decide what he wants at this point. He presses the "Single Items" button. From there he sees a pictures for everything the menu has to offer. He decides he'll have a hamburger. He presses the button for Hamburger and continues to the next screen. He then decides he'll also have a Coke and Fries. After adding those to his running total he decides the longer he's looked at the hamburger the worse it looks. He removes it from the total by pressing it within the running total, and it's removed. He then selects a chicken sandwich, and presses "Place Order." After reviewing the order he is satisfied and presses "Finalize Order." He then decides to pay with cash and insert it into the machine. A few moments later he receives his food and goes on his way.

Some slides taken
from lecture
by Khai Truong

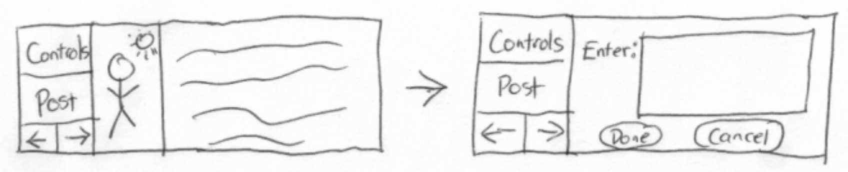
Storyboarding

- Pencil and paper simulation or walkthrough of system look and functionality
  - Use sequence of diagrams/drawings
  - Show key snap shots
  - Quick & easy

Uses / background

- Very similar in nature to:
  - Comic art / cartoons
- Used in:
  - Movie / multimedia design
  - Product / software development

Example



Sketches solve two problems with use of more fully-developed prototypes

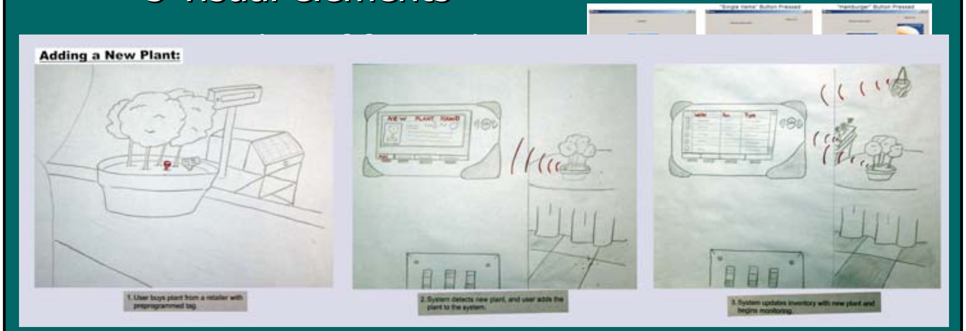
User reluctance to suggest changes to what might look like a finished product

User focus too much on details (graphic design, etc) of UI rather than big picture

Elements of storyboard

- Graphical depiction of scenarios

- 5 visual elements


"Single Items" Button Pressed
"Hamburger" Button Pressed
Adding a New Plant:
NEW PLANT FOUND
1. User buys plant from a retailer with preprogrammed tag.
2. System detects new plant, and user adds the plant to the system.
3. System updates inventory with new plant and begins monitoring.

How is it done?

- **Novice designers' process**
  - Individual brainstorming about ideas
  - May do some quick initial sketches
  - Team meeting to discuss ideas / drawings
  - Decision on what to draw
  - Spend next ~8 hours together drawing
    - Co-location allows quick feedback
    - Can also glance at what others are drawing for inspiration

How is it done?

- **Expert designers' process**
  - Get assignment
  - Individual brainstorming about ideas
    - Determine the story
    - Includes a lot of sketches using pencil + paper
    - A very iterative process through a lot of initial drafts
  - Team meeting to discuss ideas / drawings
    - Share copies of drawings
    - Discuss what stories should be told
  - Repeat
  - Generate more polished art for presentation
  - Develop

Experts' advice on storyboarding

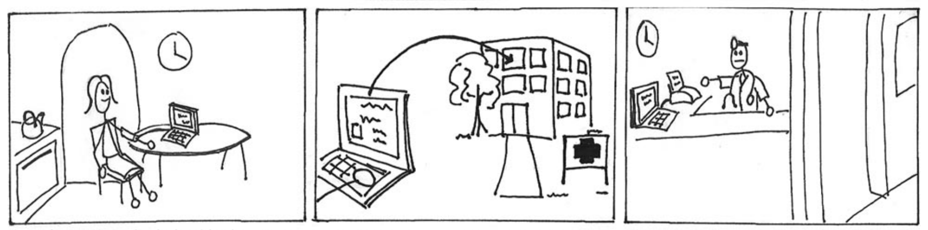
- Keep it short: 1 interaction/activity per storyboard
- More is not always better. Why?
  - May lose focus of story
  - May lose reader's attention
- Biggest challenge? Experts say:
  - Must be able to succinctly tell story

Keep the drawing short

- Drawing more is not always needed…

Use taglines / captions

- Keep it short



1. At home, Mary checks her blood pressure.

2. After a few simple key presses, her blood pressure readings get sent to a clinic.

3. The information is made available to her doctor.

Inclusion of actors and objects helps to create empathy

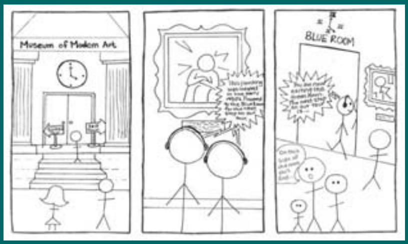
- *"The first thing users will want to know is why do I even care about this application?"*

I hate eating lunch alone with Sam
Oh good Frances came on line. Maybe she'll join us.
BANG

- Can show how the user interacts with the system and how the system affects the user

When to show time passing

- Time passing is implicit
- Only needed when gross changes or minute changes need to be explicit
- Readers bring own expectations of how much time passes into the storyboard



Some advice

- Figure out your story
- Identify main points in the story
- Draw 3-5 frames/panes (to match the main points)
- Keep it simple…
- Add taglines / text to enhance understanding
- Pilot storyboards & iterate

4. Scenarios (aka Use Cases)

- Hypothetical or fictional situations of use
  - Typically involving some person, event, situation and environment
  - Provide context of operation
  - Often in narrative form, but can also be sketches or even videos

Scenario Utility

- Engaging and interesting
- Allows designer to look at problem from another person’s point of view
- Facilitates feedback and opinions
- Can be very futuristic and creative
- Can involve social and interpersonal aspects of the task

5. Other Techniques

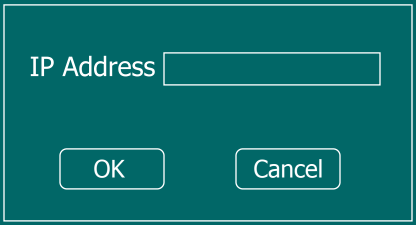
- Tutorials & Manuals
  - Maybe write them out ahead of time to flesh out functionality
  - Forces designer to be explicit about decisions
  - Putting it on paper is valuable

Computer-Supported Methods

- Simulate more of system functionality
  - Usually just some features or aspects
  - Can focus on more of details
  - Typically engaging
  - Can lead to “stale” design, can focus user (or customer) too much on the details of the interface, too early in the design process
  - Danger: Users are more reluctant to suggest changes once they see more realistic prototype

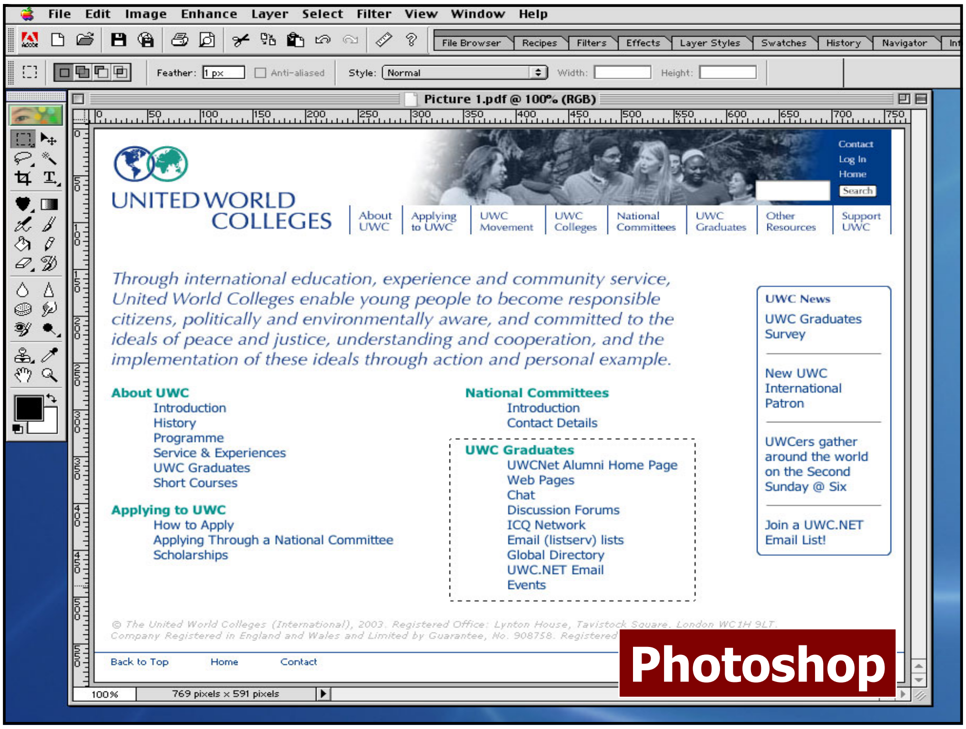
Prototyping Tools

- 1.Draw/Paint programs
  - Draw each screen, good for look



Thin, horizontal prototype

PhotoShop, Corel Draw,...



UNITED WORLD COLLEGES

Contact
Log In
Home
Search

About UWC | Applying to UWC | UWC Movement | UWC Colleges | National Committees | UWC Graduates | Other Resources | Support UWC

Through international education, experience and community service, United World Colleges enable young people to become responsible citizens, politically and environmentally aware, and committed to the ideals of peace and justice, understanding and cooperation, and the implementation of these ideals through action and personal example.

About UWC
- Introduction
- History
- Programme
- Service & Experiences
- UWC Graduates
- Short Courses

Applying to UWC
- How to Apply
- Applying Through a National Committee
- Scholarships

National Committees
- Introduction
- Contact Details

UWC Graduates
- UWCNet Alumni Home Page
- Web Pages
- Chat
- Discussion Forums
- ICQ Network
- Email (listserv) lists
- Global Directory
- UWC.NET Email
- Events

UWC News

UWC Graduates Survey

New UWC International Patron

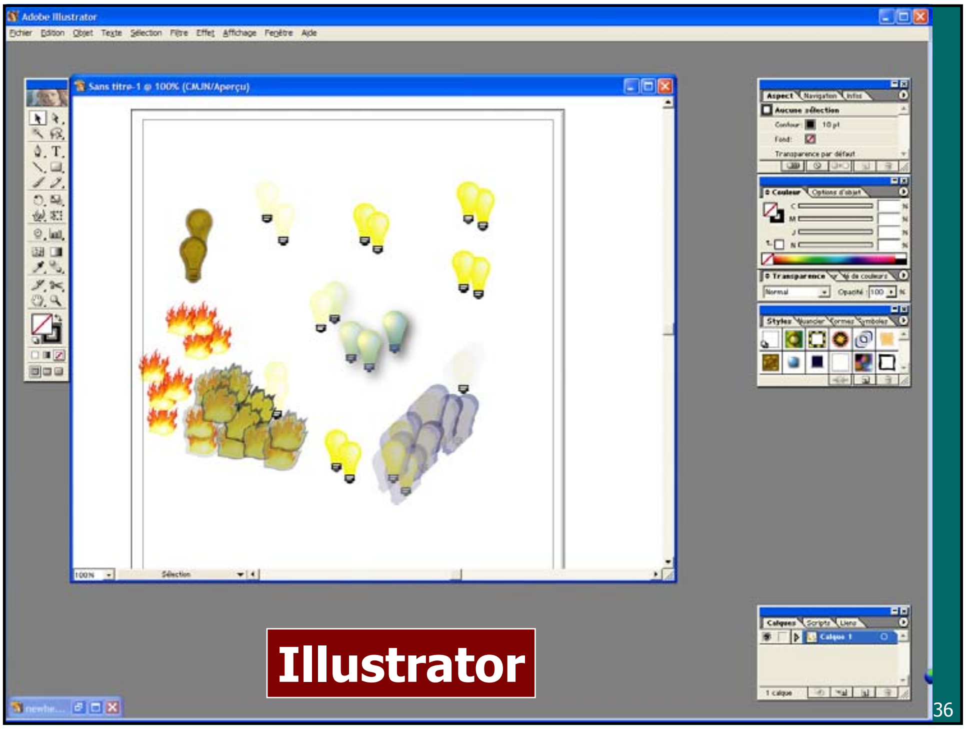
UWCers gather around the world on the Second Sunday @ Six

Join a UWC.NET Email List!

© The United World Colleges (International), 2003. Registered Office: Lynton House, Tavistock Square, London WC1H 9LT. Company Registered in England and Wales and Limited by Guarantee, No. 908758. Registered

Back to Top | Home | Contact

Photoshop



Illustrator

Prototyping Tools

- 2. Scripted simulations/slide shows
  - Put storyboard-like views down with (animated) transitions between them
  - Can give user very specific script to follow
  - Often called *chauffeured prototyping*
  - Examples: PowerPoint, Hypercard, Macromedia Director, HTML

Powerpoint Transition Controls

Mouse click actions:

Next slide

Previous slide

First slide

Last slide

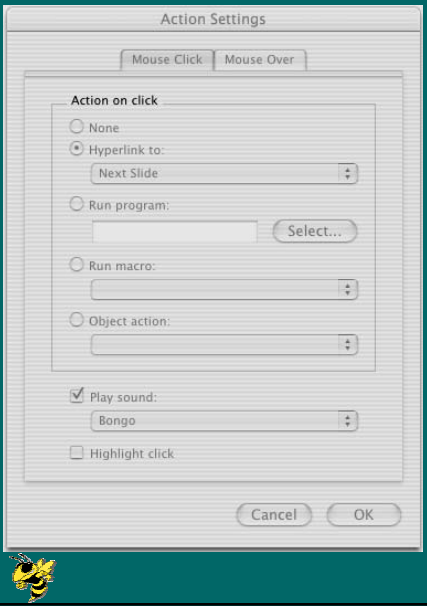
Last slide viewed

End show

Custom show

URL

File



Action Settings

Action on click

○ None

● Hyperlink to:

Next Slide

○ Run program:

○ Run macro:

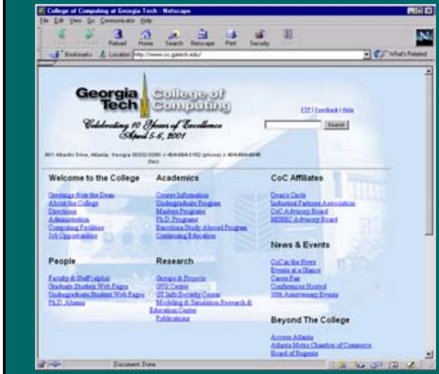
○ Object action:

☑ Play sound:

Bongo

□ Highlight click

Scripting Example



College of Computing at Georgia Tech - Netscape

http://www.cc.gatech.edu/

Georgia Tech College of Computing

Celebrating 10 Years of Excellence
April 5-6, 2001

Welcome to the College
- Greetings from the Dean
- About the College
- Directions
- Administration
- Computing Facilities
- Job Opportunities

Academics
- Course Information
- Undergraduate Program
- Masters Programs
- Ph.D. Programs
- Barcelona Study Abroad Program
- Continuing Education

CoC Affiliates
- Dean's Circle
- Industrial Partners Association
- CoC Advisory Board
- MSBIC Advisory Board

News & Events
- CoC in the News
- Events at a Glance
- Career Fair
- Conferences Hosted
- 10th Anniversary Events

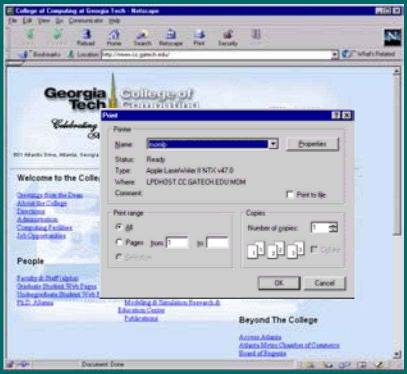
People
- Faculty & Staff (alpha)
- Graduate Student Web Pages
- Undergraduate Student Web Pages
- Ph.D. Alumni

Research
- Groups & Projects
- GVU Center
- GT Info Security Center
- Modeling & Simulation Research & Education Center
- Publications

Beyond The College
- Access Atlanta
- Atlanta Metro Chamber of Commerce
- Board of Regents

Document: Done

Ctrl-p



Print

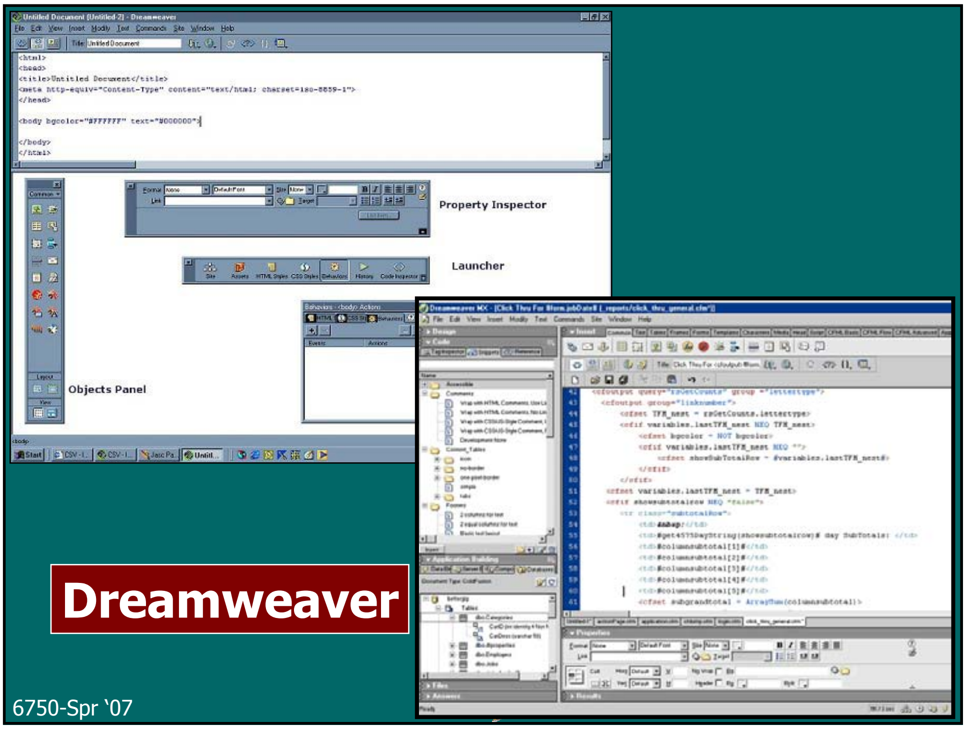
Printer
Name: momlp
Status: Ready
Type: Apple LaserWriter II NTX v47.0
Where: LPDHOST.CC.GATECH.EDU:MOM
Comment:
□ Print to file

Print range
● All
○ Pages from: 1 to:
○ Selection

Copies
Number of copies: 1
□ Collate

OK Cancel

But Beware!

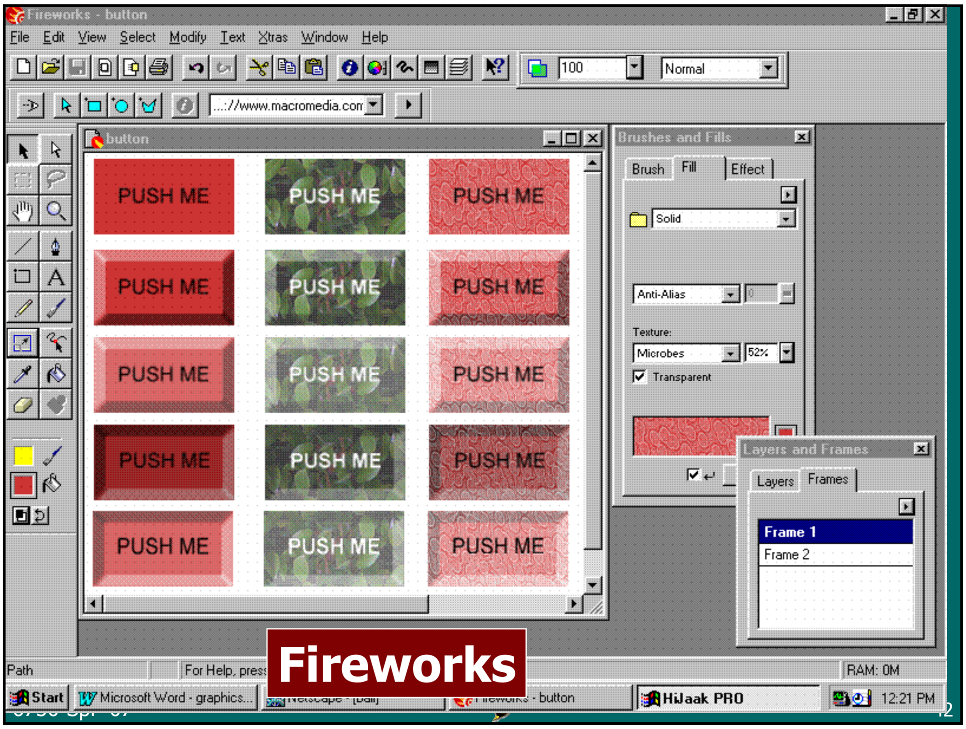


```
<html>
<head>
<title>Untitled Document</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>

<body bgcolor="#FFFFFF" text="#000000">

</body>
</html>
```

```
<cfoutput query="rsGetCounts" group ="lettertype">
  <cfoutput group="linknumber">
      <cfset TFR_nest = rsGetCounts.lettertype>
      <cfif variables.lastTFR_nest NEQ TFR_nest>
          <cfset bgcolor = NOT bgcolor>
          <cfif variables.lastTFR_nest NEQ "">
              <cfset showSubTotalRow = #variables.lastTFR_nest#>
          </cfif>
      </cfif>
   <cfset variables.lastTFR_nest = TFR_nest>
   <cfif showsubtotalrow NEQ "false">
      <tr class="subtotalRow">
          <td>&nbsp;</td>
          <td>#get4575DayString(showsubtotalrow)# day SubTotals: </td>
          <td>#columnsubtotal[1]#</td>
          <td>#columnsubtotal[2]#</td>
          <td>#columnsubtotal[3]#</td>
          <td>#columnsubtotal[4]#</td>
          <td>#columnsubtotal[5]#</td>
          <cfset subgrandtotal = ArraySum(columnsubtotal)>
```

Untitled Document (Untitled-2) - Dreamweaver
File Edit View Insert Modify Text Commands Site Window Help
Title: Untitled Document
Property Inspector
Launcher
Objects Panel
Dreamweaver MX - [Click Thru For #form.jobDate# (_reports/click_thru_general.cfm*)]

Dreamweaver



Fireworks

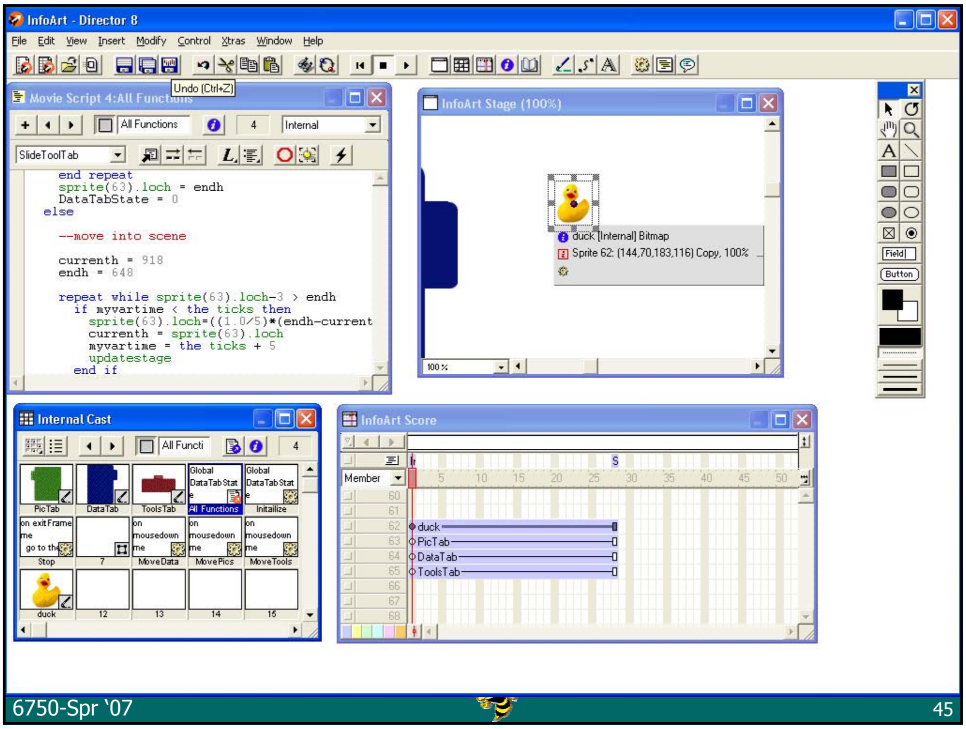
Apple Hypercard

- Once a very popular prototyping tool for simulating UI
- Allows control of simple card transitions
- More complex behaviors

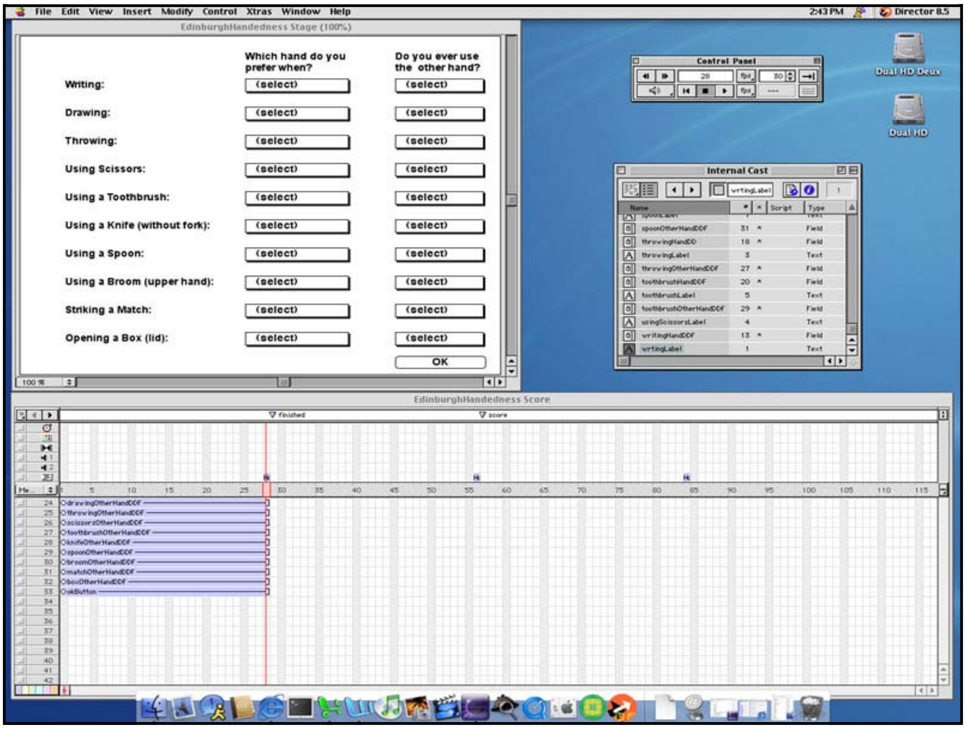
```
on mouseUp
   play “boing”
   wait for 3 seconds
   visual effect wipe left very fast to black
   click at 150,100
   type “goodbye”
end mouseUp
```

Macromedia Director

- Combines various media with script written in Lingo language
- Concerned with place and time
  - Objects positioned in space on “stage”
  - Objects positioned in time on “score”
- Easy to transition between screens
- Can export as executable or as Web Shockwave file



```
    end repeat
    sprite(63).loch = endh
    DataTabState = 0
  else

    --move into scene

    currenth = 918
    endh = 648

    repeat while sprite(63).loch-3 > endh
      if myvartime < the ticks then
        sprite(63).loch=((1.0/5)*(endh-current
        currenth = sprite(63).loch
        myvartime = the ticks + 5
        updatestage
      end if
```



	Which hand do you prefer when?	Do you ever use the other hand?
Writing:	(select)	(select)
Drawing:	(select)	(select)
Throwing:	(select)	(select)
Using Scissors:	(select)	(select)
Using a Toothbrush:	(select)	(select)
Using a Knife (without fork):	(select)	(select)
Using a Spoon:	(select)	(select)
Using a Broom (upper hand):	(select)	(select)
Striking a Match:	(select)	(select)
Opening a Box (lid):	(select)	(select)

OK

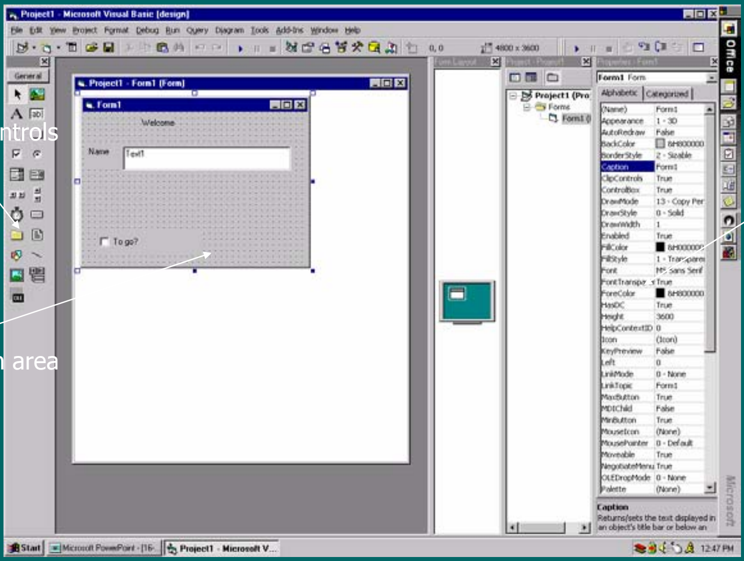
Name	#	*	Script	Type
spoonOtherHandDDF	31	*		Field
throwingHandDD	18	*		Field
throwingLabel	3			Text
throwingOtherHandDDF	27	*		Field
toothbrushHandDDF	20	*		Field
toothbrushLabel	5			Text
toothbrushOtherHandDDF	29	*		Field
usingScissorsLabel	4			Text
writingHandDDF	13	*		Field
wrtingLabel	1			Text

Prototyping Tools

- 3. Interface Builders
  - Tools for laying out windows, controls, etc. of interface
    - Have build and test modes that are good for exhibiting look and feel
    - Generate code to which back-end functionality can be added through programming
  - Examples: Visual Basic, Delphi, UIMX, ...

Visual Basic

More to come later today


Project1 - Microsoft Visual Basic [design]
File Edit View Project Format Debug Run Query Diagram Tools Add-Ins Window Help
0, 0
4800 x 3600
General
Project1 - Form1 (Form)
Form1
Welcome
Name
Text1
To go?
Form Layout
Project - Project1
Project1 (Pro
Forms
Form1 (
Properties - Form1
Office
Microsoft
Start
Microsoft PowerPoint - [16-
Project1 - Microsoft V...
12:47 PM

Form1 Form

Alphabetic | Categorized

(Name)	Form1
Appearance	1 - 3D
AutoRedraw	False
BackColor	&H800000
BorderStyle	2 - Sizable
Caption	Form1
ClipControls	True
ControlBox	True
DrawMode	13 - Copy Per
DrawStyle	0 - Solid
DrawWidth	1
Enabled	True
FillColor	&H000000
FillStyle	1 - Transparen
Font	MS Sans Serif
FontTranspa	True
ForeColor	&H800000
HasDC	True
Height	3600
HelpContextID	0
Icon	(Icon)
KeyPreview	False
Left	0
LinkMode	0 - None
LinkTopic	Form1
MaxButton	True
MDIChild	False
MinButton	True
MouseIcon	(None)
MousePointer	0 - Default
Moveable	True
NegotiateMenu	True
OLEDropMode	0 - None
Palette	(None)

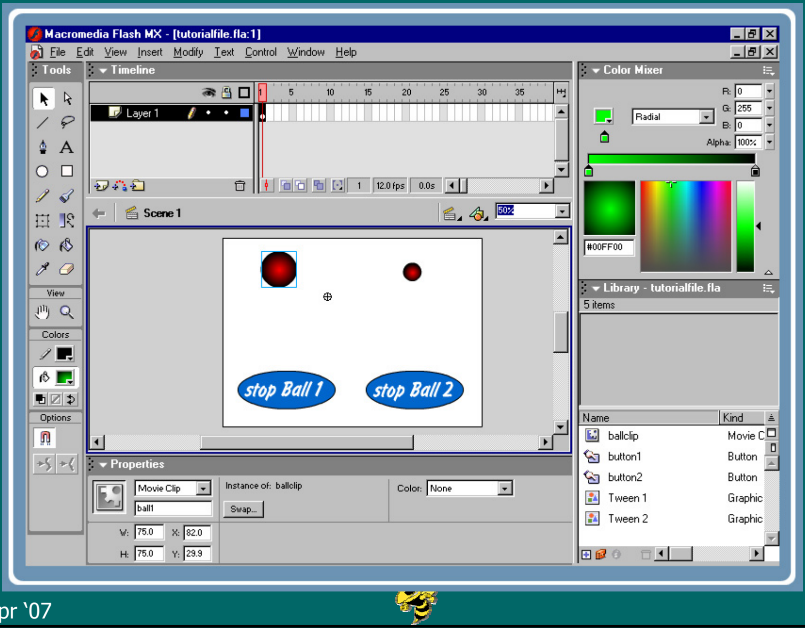
Caption
Returns/sets the text displayed in an object's title bar or below an

UI Controls

Control properties

Design area

Flash - A category of its own


Macromedia Flash MX - [tutorialfile.fla:1]
File Edit View Insert Modify Text Control Window Help
Tools
Timeline
Layer 1
12.0 fps
0.0s
Scene 1
50%
View
Colors
Options
stop Ball 1
stop Ball 2
Properties
Swap...
Color Mixer
Library - tutorialfile.fla
5 items

Movie Clip

ball1

Instance of: ballclip

Color: None

W: 75.0 X: 82.0

H: 75.0 Y: 29.9

R: 0 G: 255 B: 0 Alpha: 100%

Radial

#00FF00

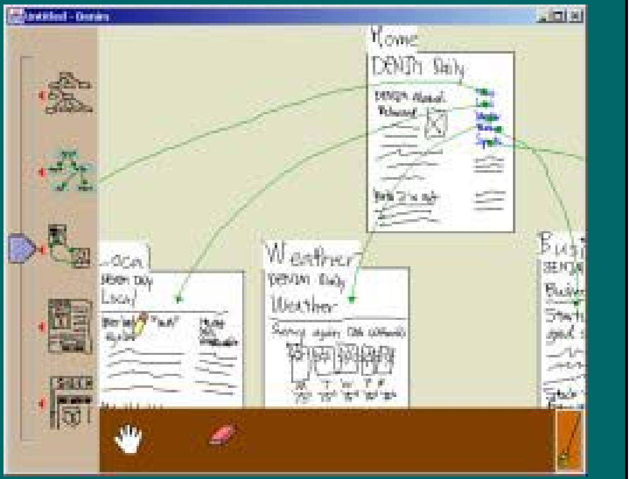
Name	Kind
ballclip	Movie C
button1	Button
button2	Button
Tween 1	Graphic
Tween 2	Graphic

True Programming

- Less useful for rapid prototyping, but can save re-coding time down the road
- More constrained in look and feel
- Constrained to traditional interaction styles and methods
  - Hard to think outside the box
- More to come in a few minutes…

Other Prototyping Tools

- Denim

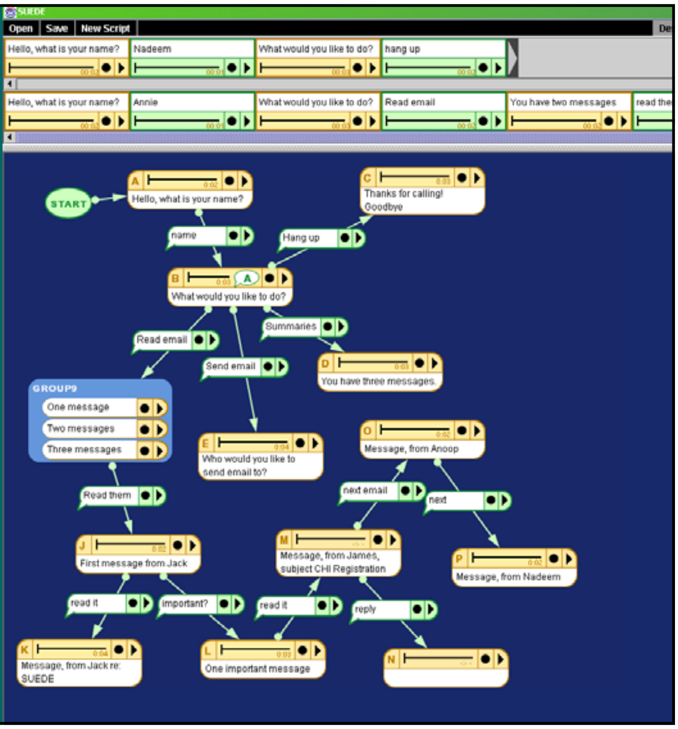


http://guir.berkeley.edu

Audio Interface (Telephony) Builder Tools

- SUEDE - Flow-chart for speech interface
  - Landay et al, UC Berkeley (now U Washington)
- Used for wizard-of-Oz studies
- Could be used to drive real system
- guir.berkeley.edu/projects/suede/index.shtml

Suede



Prototyping Technique

- Wizard of Oz - Person simulates and controls system from "behind the scenes"
  - Use mock interface and interact with users
  - Good for simulating system that would be difficult to build

Can be either computer-based or not

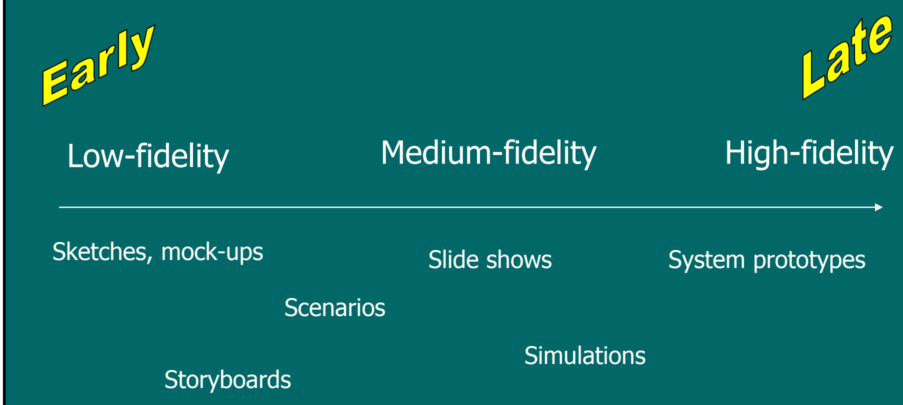
Wizard of Oz

- Method:
  - Behavior should be algorithmic
  - Good for voice recognition systems
- Advantages:
  - Allows designer to immerse oneself in situation
  - See how people respond, how specify tasks

Prototyping Tools

- Good features
  - Easy to develop & modify screens
  - Supports type of interface you are developing
  - Supports variety I/O devices
  - Easy to link screens and modify links
  - Allows calling external procedures & program
  - Allows importing text, graphics, other media
  - Easy to learn and use
  - Good support from vendor

Prototyping



Prototyping Summary

- Tradeoffs of simplicity, manageability
- Veracity
- Interactiveness
- Up-front costs vs. down the road costs

- Key: Don't let the prototyping environment drive or constrain your creativity!!

Tutorials

Photoshop/Illustrator:

http://www.absolutecross.com/tutorials/photoshop/

http://www.planetphotoshop.com/tutorials.html

http://thetechnozone.com/bbyc/Illustrator.htm

http://studio.pinnacle-elite.com/tutorials/aitut01.html

Dreamweaver/HTML:

http://www.cbtcafe.com/dreamweaver/

http://www.sitebuilder.ws/dreamweaver/tutorials/

Fireworks:

http://www.cbtcafe.com/fireworks/index.html

VB:

http://www.vbtutor.net/vbtutor.html

http://juicystudio.com/tutorial/vb/

http://webspace.dialnet.com/paul_pbcoms/vb/tutor.html

Flash:

http://www.uic.edu/depts/accc/seminars/flashintro/index.html

http://www.absolutecross.com/tutorials/flash/

Director:

http://www.herts.ac.uk/lis/mmedia/directortutorial/

http://www.tutorialfind.com/tutorials/macromedia/director/

http://www.fbe.unsw.edu.au/learning/director/

UI Software & Programming

- OK, let's return to what we were talking about earlier
- The final level up…building the actual application

User Interface Software

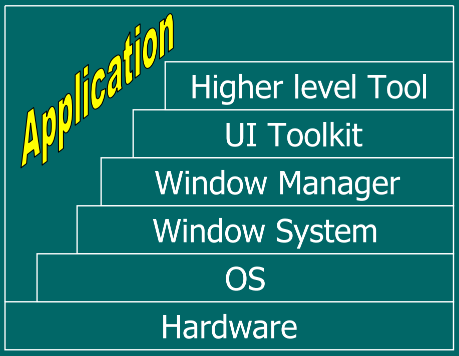
- What support is provided for building graphical user interfaces?
  - UI toolkits
  - GUI builder tools

- Let’s examine some background…

GUI System Architecture

What does it look like?

Layered Architecture



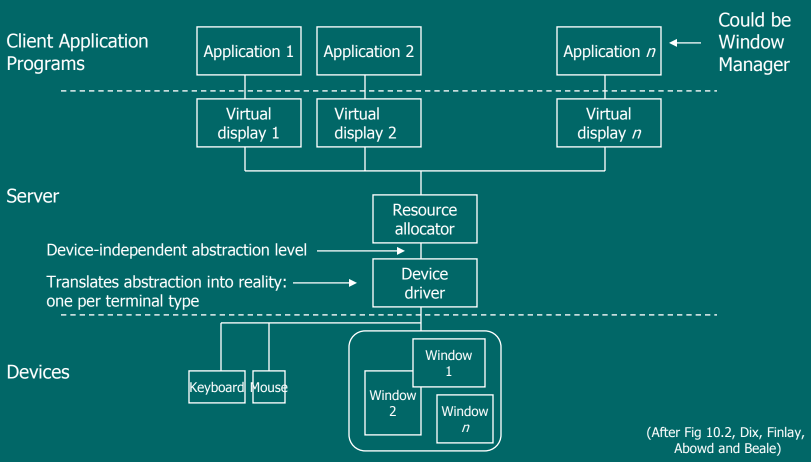
Window System

- Allocates and manages
  - Regions of display to application programs, keeps programs out of each other's way
  - Input devices (keyboard, mouse) to application, routes input events
- Called by application program to
  - Create/delete windows (ie, allocate resource)
  - Operate on windows (move, resize, bring to top, hide, give name, clear)
  - Enable input devices
  - Get input from user via interaction devices

Key Windowing Concepts

- Server/client
- Window manager ≠ window system
- Virtual display/input device abstraction
- Window hierarchy
- Event notification: queue vs. callback
- User actions
- Input focus
- Consequences of server actions (redraw)

Client-Server Window System



Client Application Programs

Application 1

Application 2

Application *n*

Could be Window Manager

Virtual display 1

Virtual display 2

Virtual display *n*

Server

Resource allocator

Device-independent abstraction level

Device driver

Translates abstraction into reality: one per terminal type

Devices

Keyboard

Mouse

Window 1

Window 2

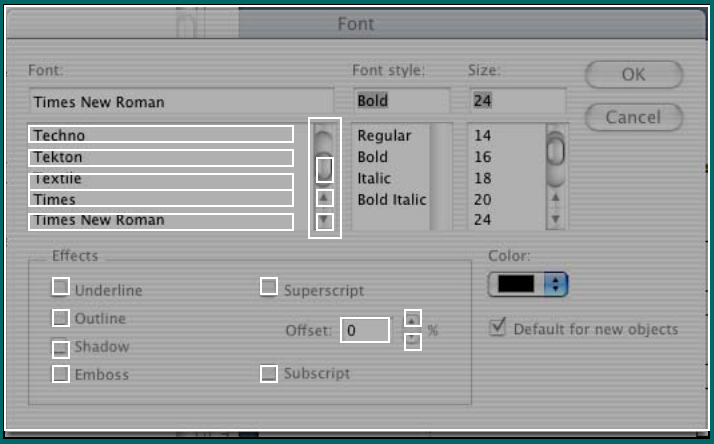
Window *n*

(After Fig 10.2, Dix, Finlay, Abowd and Beale)

Client- Server

- "Policy-free" server, not seen by user
- A *Window manager* client gives the "look and feel", not the *window system* (server)
- Multiple clients, each seen by user on terminal
- Linux/Unix WMs built on the X Window System include
  - LVWM - Open LookVvirtual Window Manager
  - FVWM - Feeble Virtual Window Manager
  - MWM - Motif Window Manager
  - AfterStep
  - GWM - Generic Window Manager

Window Hierarchy



Font

Font: Times New Roman

Techno
Tekton
Textile
Times
Times New Roman

Font style: Bold

Regular
Bold
Italic
Bold Italic

Size: 24

14
16
18
20
24

Effects

□ Underline
□ Outline
□ Shadow
□ Emboss
□ Superscript
Offset: 0 %
□ Subscript

Color:

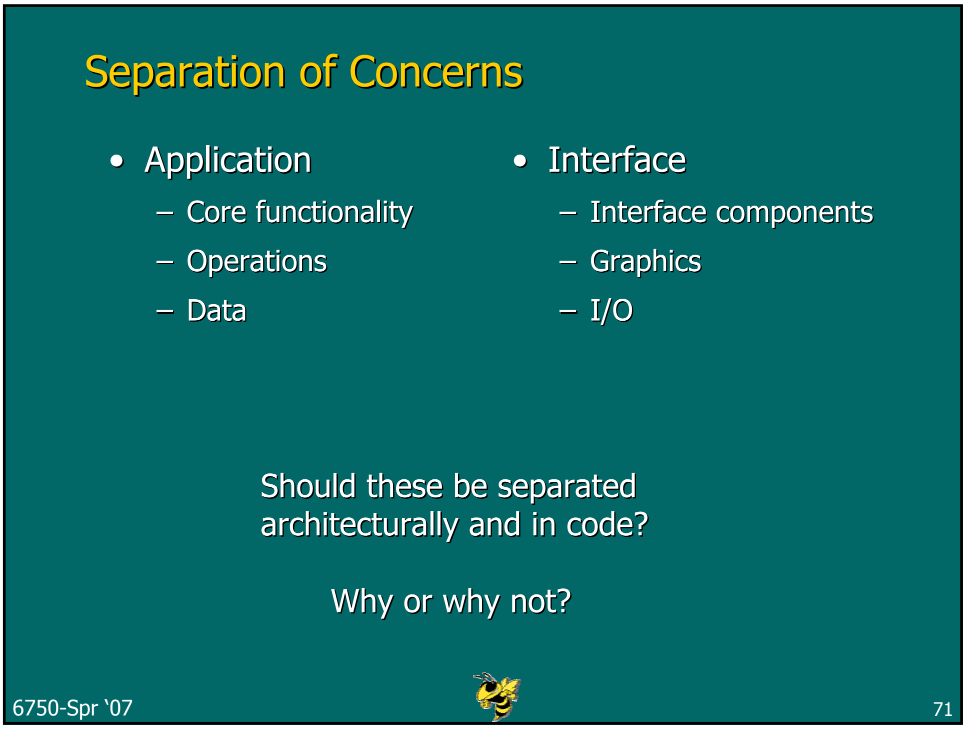
☑ Default for new objects

Event Notification

- Event types
  - User actions - mouse movements/button clicks, keyboarding, enter or leave window
  - Server actions, such as making window visible - client may need to redraw window
- Event queue – program examines the queue, calls an action routines, determined by event type and screen object under cursor
- Callback – the window system *notifier* is told which action routine to call for each type of event for each object on screen

Input Focus

- To which client do events go?
- Not always where the cursor is located
- Dynamic dragging outside of window
- Type into one window while "mouse" in another

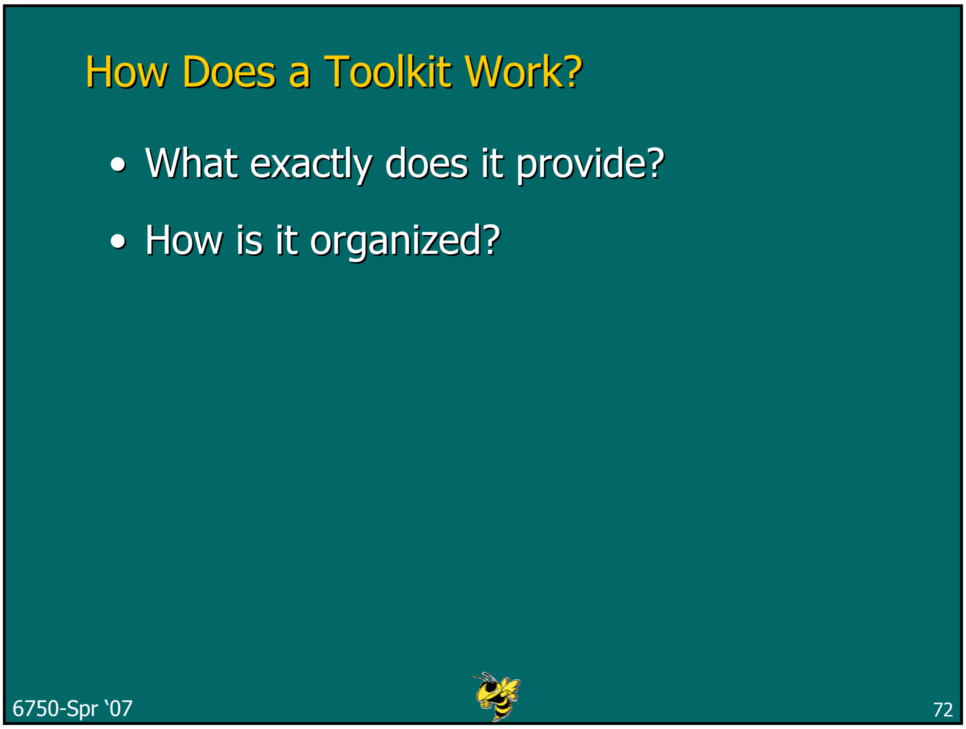


Separation of Concerns

- Application
  - Core functionality
  - Operations
  - Data
- Interface
  - Interface components
  - Graphics
  - I/O

Should these be separated architecturally and in code?

Why or why not?



How Does a Toolkit Work?

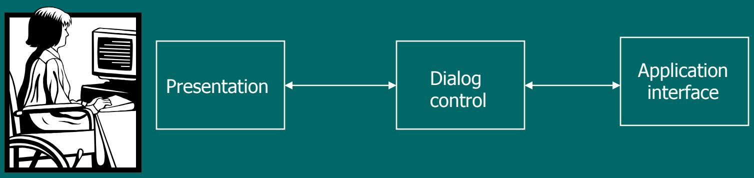
- What exactly does it provide?
- How is it organized?

Toolkit Workings

- User takes actions, interacts with interface
- Those actions must be delivered to application in meaningful ways
- Application takes appropriate actions, perhaps updating display

Seeheim Model

Conversational model

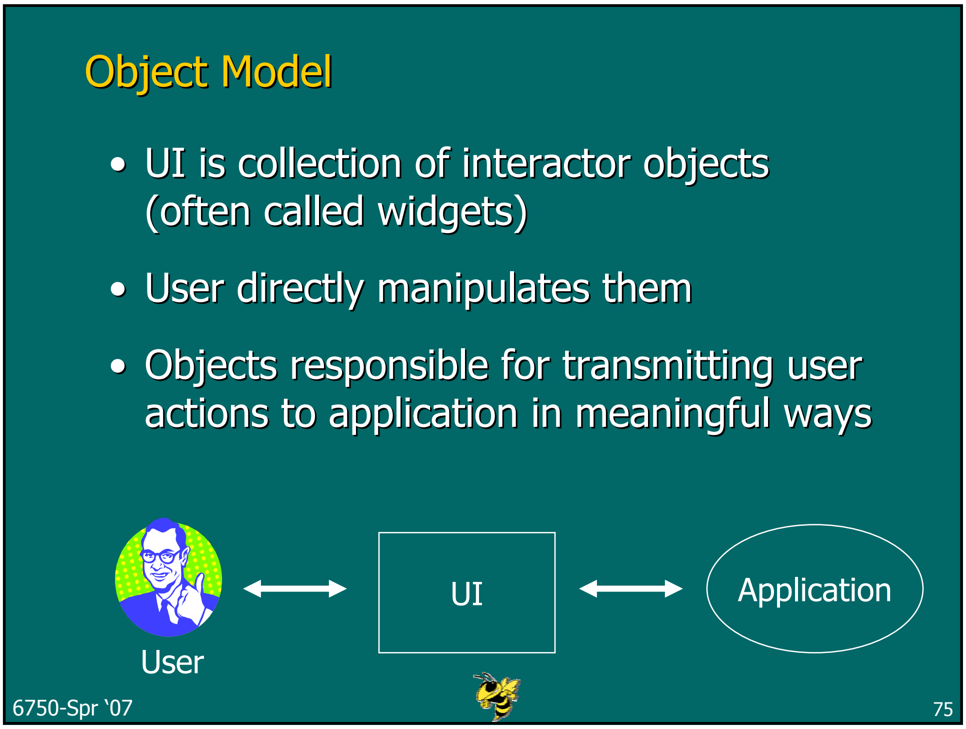


Presentation

Dialog control

Application interface

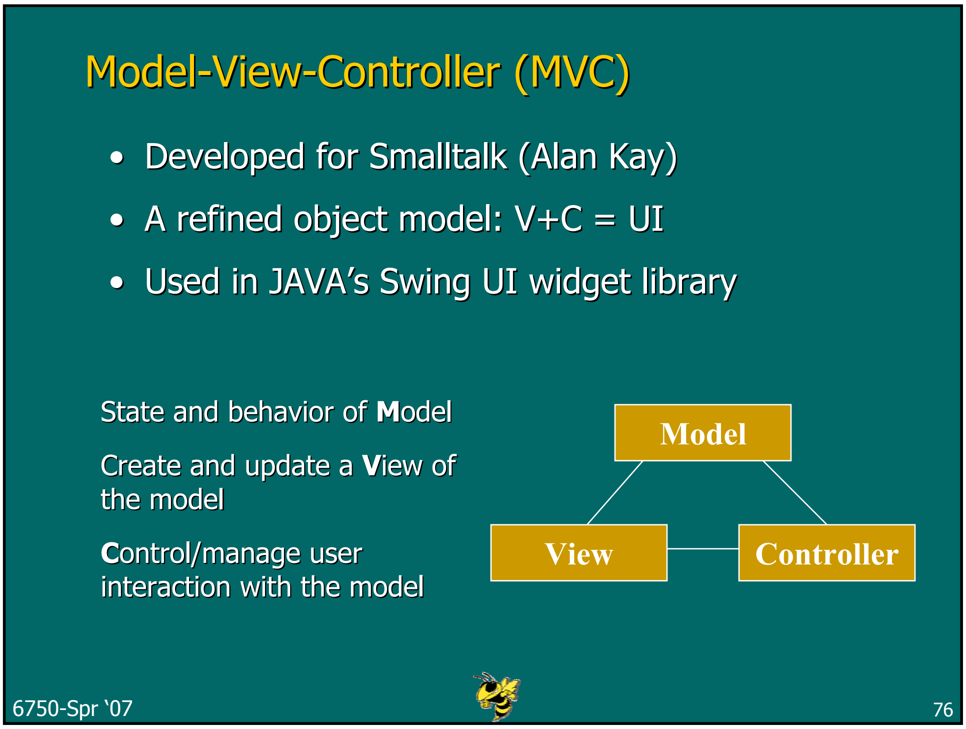
Dominant model for long time



Object Model

- UI is collection of interactor objects (often called widgets)
- User directly manipulates them
- Objects responsible for transmitting user actions to application in meaningful ways

User
UI
Application



Model-View-Controller (MVC)

- Developed for Smalltalk (Alan Kay)
- A refined object model: V+C = UI
- Used in JAVA's Swing UI widget library

State and behavior of **M**odel

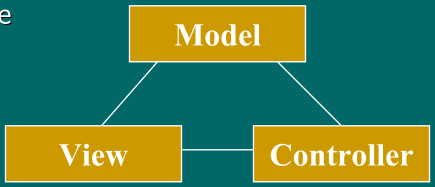
Create and update a **V**iew of the model

Control/manage user interaction with the model

Model
View
Controller

MVC Example - a Push Button

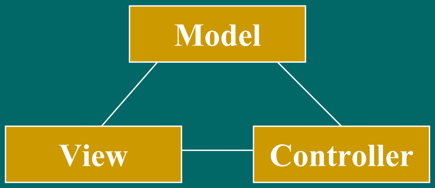
- Model - a boolean - on or off
- View - a drawing - in each possible state
- Controller - tell model to change state, and view to change view
- Note - most buttons have more complex behavior than this

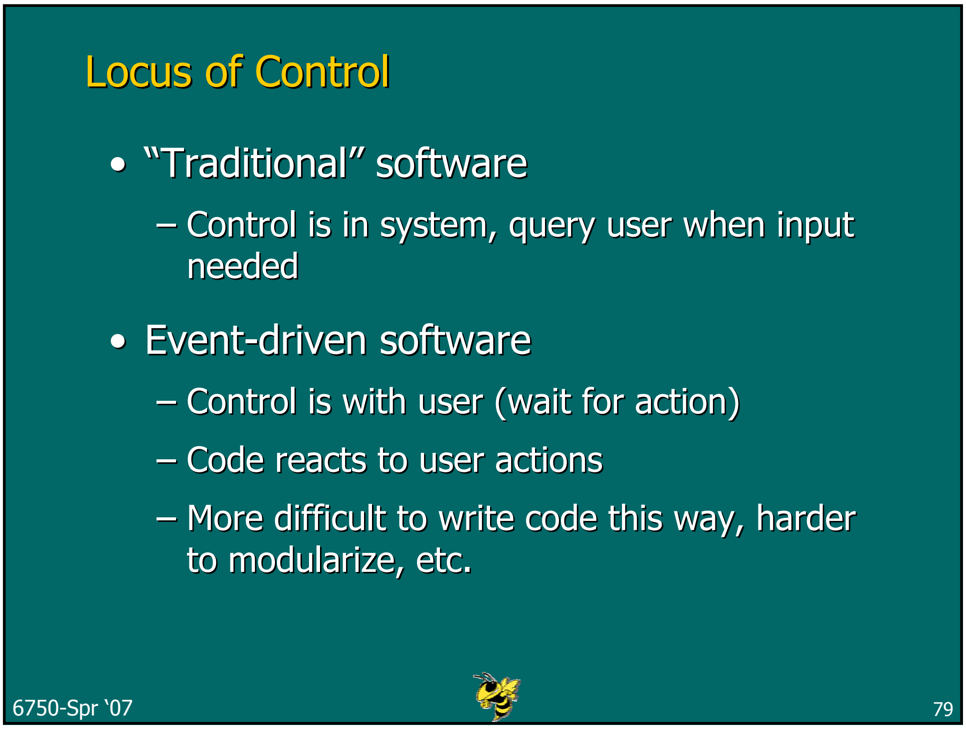


See http://www.javaworld.com/javaworld/jw-04-1998/jw-04-howto.html

MVC Flexibility

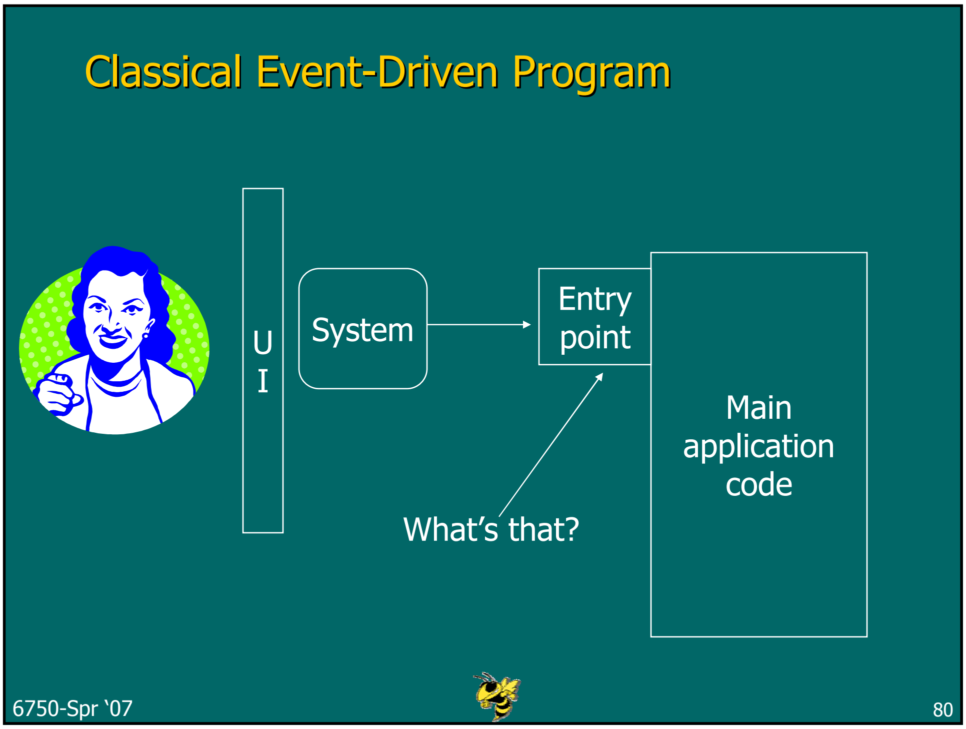
- Clear separation of concerns makes changes easy
  - Want a different button appearance? Change the view, nothing else.
  - Want mouse_over rather than mouse_down to change state?





Locus of Control

- “Traditional” software
  - Control is in system, query user when input needed
- Event-driven software
  - Control is with user (wait for action)
  - Code reacts to user actions
  - More difficult to write code this way, harder to modularize, etc.



Classical Event-Driven Program

U I

System

Entry point

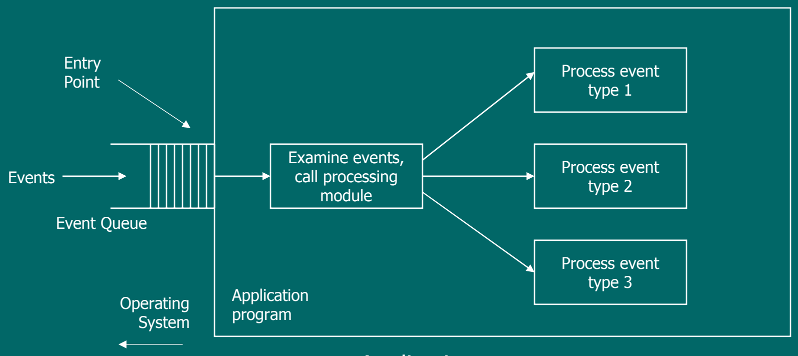
Main application code

What’s that?

UI Toolkit

- What application programmer typically programs with
- Combination of interface objects and management behaviors
- Usually object-oriented now
- Library of software components and routines that programmer puts together
  - X Windows: X Toolkit & Motif
  - Macintosh: Mac Toolbox, MacApp
  - Windows: Windows Developers' Toolkit
  - Java: Swing

Classical Approach to Input Event Dispatching – Not so Good



Application program polls the event queue

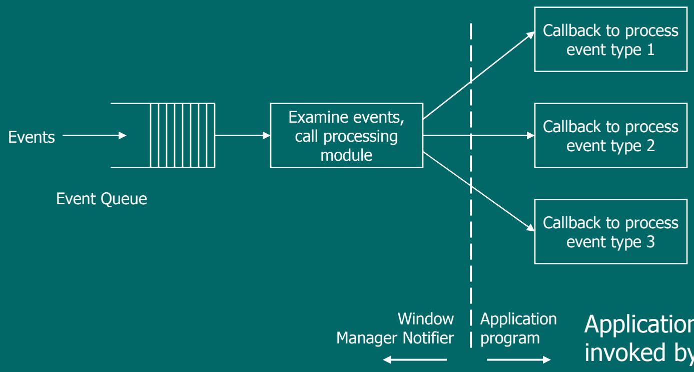
Classical Event-Driven Program

- Initialize display & system
- Repeat
  - Wait for and get next user action
  - Decipher action
  - Take appropriate action
  - Update display
- Until Done

Callback Routine – Way to go

- Software procedure, part of application
- Invoked when particular action occurs to UI component, such as pressing a PushButton
- Procedure is invoked with event parameters

Window Approach to Input Event Dispatching - Good


Events
Event Queue
Examine events, call processing module
Callback to process event type 1
Callback to process event type 2
Callback to process event type 3
Window Manager Notifier
Application program

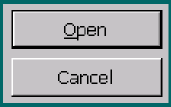
Application program callbacks invoked by window manager

Widgets are Source of Events

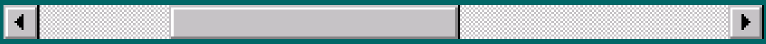
- Widgets are typically structured as objects
- JAVA's Swing uses MVC model
  - Each widget has three parts

Windows Widgets

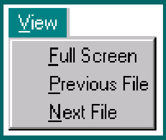
- Buttons (several types)



- Scroll bars and sliders

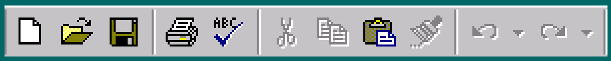


- Pulldown menus

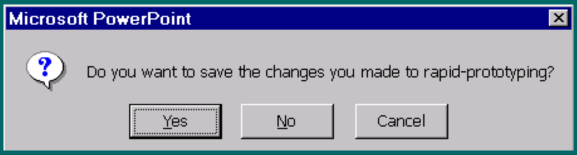


More Windows Widgets

- Palettes



- Dialog boxes



Do you want to save the changes you made to rapid-prototyping?

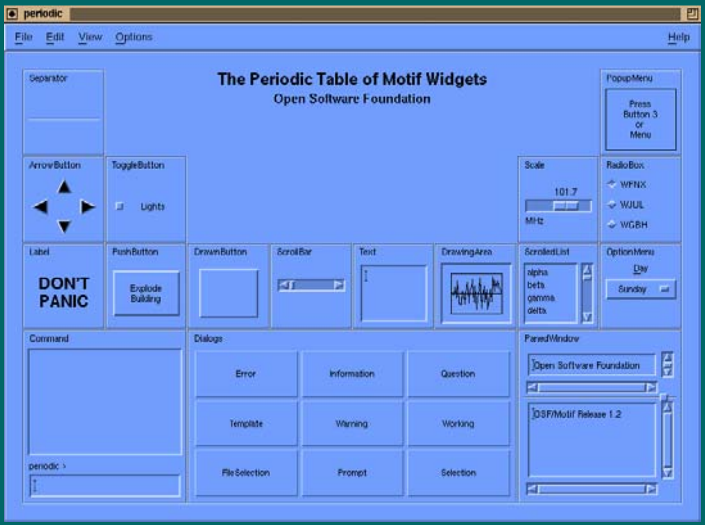
- Windows and many more...

Example – X & Motif

- Object-oriented hierarchy of UI interactors called widgets
  - – Associate callback routines in your code with them
- Interface is built up by putting child widgets “onto” parent widgets

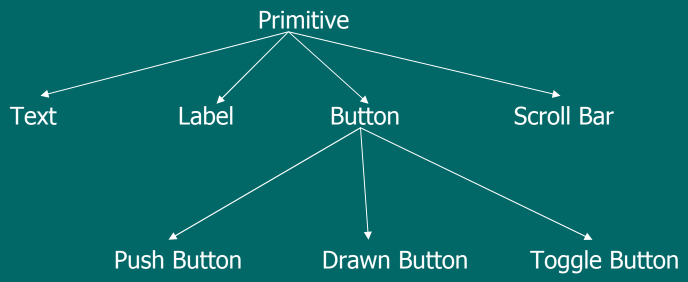
Widget

Graphical user interface interactor object



Widget Hierarchy

- Widgets organized into inheritance hierarchy



Widget

- Visual appearance
- Set of tailorable attributes

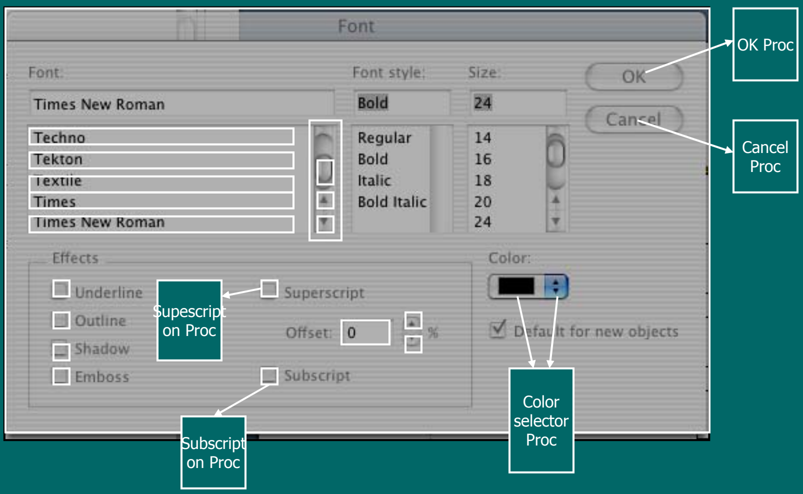
```
PushButton  {
   Color  BackGround;
   int MarginLeft;
   int MarginRight;
   int BorderWidth;
   Pixmap ArmPixmap;
   Boolean FillOnArm;
   CallbackList ActivateCallback;
}
```

- Interactive behavior

Widget Use

- Set up widget attributes
- Create widget object (as child of parent widget)
- Define callback or event procedure for widget

Callbacks associated with objects and events



Font

Font: Times New Roman

Font style: Bold

Size: 24

Techno
Tekton
Textile
Times
Times New Roman

Regular
Bold
Italic
Bold Italic

14
16
18
20
24

OK

Cancel

Effects

□ Underline
□ Outline
□ Shadow
□ Emboss
□ Superscript
Offset: 0 %
□ Subscript

Color:

☑ Default for new objects

OK Proc

Cancel Proc

Supescript on Proc

Subscript on Proc

Color selector Proc

Multiple Callbacks per Object

- Example - button object with 5 callbacks
  - Mouse enter - (1) highlight
  - Mouse button down - (2) additional highlighting
  - Mouse leave while button down - (3) remove highlight
  - Mouse button up - (4) remove highlight, (5)perform action

Widget and Callback

```
   n = 0;
   xmstr = XmStringCreate("Color",XmSTRING_DEFAULT_CHARSET);
   XtSetArg(args[n], XmNlabelString, xmstr); n++;
   XtSetArg(args[n], XmNbackground, red); n++;
colorbut = XtCreateManagedWidget("colorbutton",
               xmPushButtonWidgetClass,focusrowcol, args, n);
   XtAddCallback(colorbut, XmNactivateCallback,
                 colorChangeCB, id);


void
colorChangeCB(Widget w, XtPointer userdata, XtPointer evtdata)
{
   // Actions
}
```

Main Program Event Loop

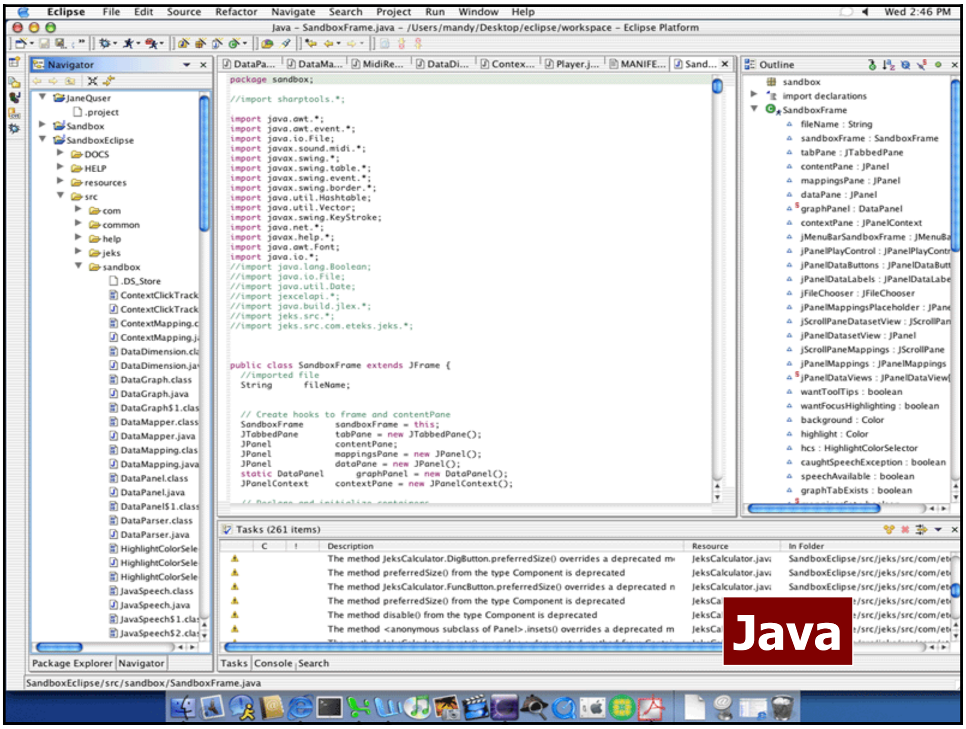
```
voidCheckXEvents()
{
   XEvent xev;

   while (XtAppPending(_context)) {
      XtAppNextEvent(_context, &xev);
      XtDispatchEvent(&xev);
      }
}
```

OO Systems

- Java's GUI programming done with AWT and Swing
- More distributed model (separate threads)
- Primary action here is dispatching events to objects (widgets) as messages
- Delegation important
  - Can make particular objects responsible for event handling

Example - Java AWT

```
public void mouseReleased(MouseEvent e){
      System.out.println ("Changing color");
      if (bHighlight)
           frame.setBackground(highlight);
      else
           frame.setBackground(normal);
      bHighlight = !bHighlight;
     }
```


Eclipse File Edit Source Refactor Navigate Search Project Run Window Help
Wed 2:46 PM
Java – SandboxFrame.java – /Users/mandy/Desktop/eclipse/workspace – Eclipse Platform
Navigator
JaneQuser
.project
Sandbox
SandboxEclipse
DOCS
HELP
resources
src
com
common
help
jeks
sandbox
.DS_Store
Package Explorer Navigator
DataPa... DataMa... MidiRe... DataDi... Contex... Player.j... MANIFE... Sand...
Outline
sandbox
import declarations
SandboxFrame
fileName : String
sandboxFrame : SandboxFrame
tabPane : JTabbedPane
contentPane : JPanel
mappingsPane : JPanel
dataPane : JPanel
graphPanel : DataPanel
contextPane : JPanelContext
highlight : Color
hcs : HighlightColorSelector
caughtSpeechException : boolean
speechAvailable : boolean
graphTabExists : boolean
Tasks Console Search
SandboxEclipse/src/sandbox/SandboxFrame.java

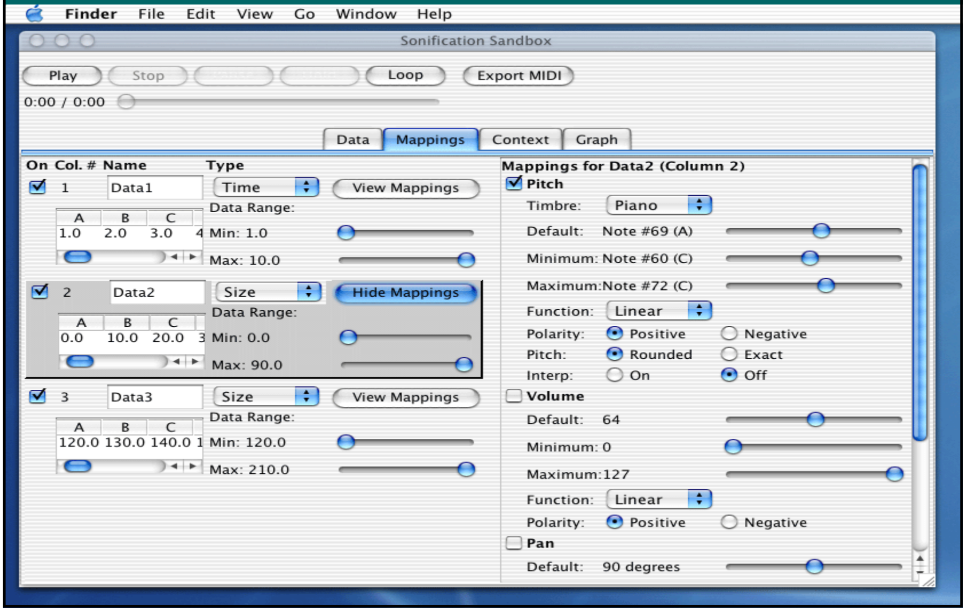
```
package sandbox;

//import sharptools.*;

import java.awt.*;
import java.awt.event.*;
import java.io.File;
import javax.sound.midi.*;
import javax.swing.*;
import javax.swing.table.*;
import javax.swing.event.*;
import javax.swing.border.*;
import java.util.Hashtable;
import java.util.Vector;
import javax.swing.KeyStroke;
import java.net.*;
import javax.help.*;
import java.awt.Font;
import java.io.*;
//import java.lang.Boolean;
//import java.io.File;
//import java.util.Date;
//import jexcelapi.*;
//import java.build.jlex.*;
//import jeks.src.*;
//import jeks.src.com.eteks.jeks.*;



public class SandboxFrame extends JFrame {
  //imported file
  String      fileName;


  // Create hooks to frame and contentPane
  SandboxFrame      sandboxFrame = this;
  JTabbedPane       tabPane = new JTabbedPane();
  JPanel            contentPane;
  JPanel            mappingsPane = new JPanel();
  JPanel            dataPane = new JPanel();
  static DataPanel      graphPanel = new DataPanel();
  JPanelContext     contextPane = new JPanelContext();
```

Tasks (261 items)

C	!	Description	Resource	In Folder
		The method JeksCalculator.DigButton.preferredSize() overrides a deprecated m	JeksCalculator.jav	SandboxEclipse/src/jeks/src/com/et
		The method preferredSize() from the type Component is deprecated	JeksCalculator.jav	SandboxEclipse/src/jeks/src/com/et
		The method JeksCalculator.FuncButton.preferredSize() overrides a deprecated n	JeksCalculator.jav	SandboxEclipse/src/jeks/src/com/et
		The method preferredSize() from the type Component is deprecated	JeksCal	e/src/jeks/src/com/et
		The method disable() from the type Component is deprecated	JeksCal	e/src/jeks/src/com/et
		The method <anonymous subclass of Panel>.insets() overrides a deprecated m	JeksCal	e/src/jeks/src/com/et

Java

Java Output



On	Col. #	Name	Type	
☑	1	Data1	Time	View Mappings
☑	2	Data2	Size	Hide Mappings
☑	3	Data3	Size	View Mappings

Data1: A 1.0, B 2.0, C 3.0, 4 — Data Range: Min: 1.0, Max: 10.0

Data2: A 0.0, B 10.0, C 20.0, 3 — Data Range: Min: 0.0, Max: 90.0

Data3: A 120.0, B 130.0, C 140.0, 1 — Data Range: Min: 120.0, Max: 210.0

Mappings for Data2 (Column 2)

☑ Pitch

Timbre: Piano

Default: Note #69 (A)

Minimum: Note #60 (C)

Maximum:Note #72 (C)

Function: Linear

Polarity: ● Positive ○ Negative

Pitch: ● Rounded ○ Exact

Interp: ○ On ● Off

□ Volume

Default: 64

Minimum: 0

Maximum:127

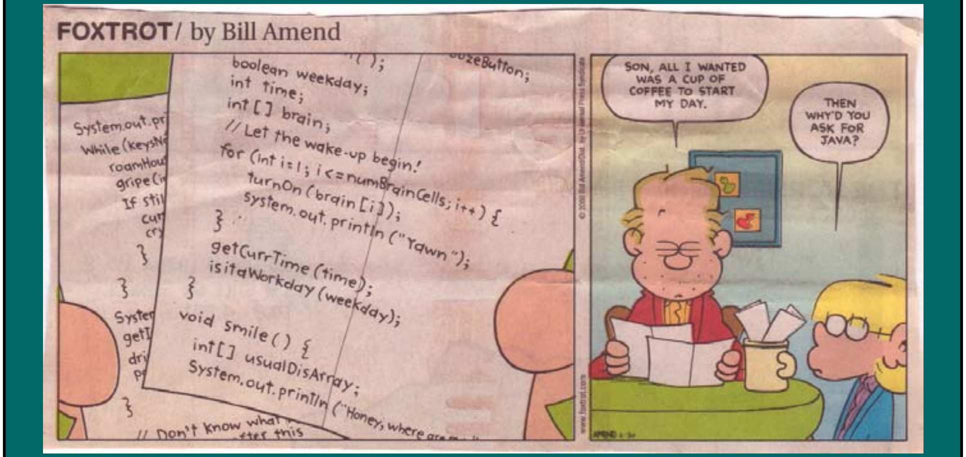
Function: Linear

Polarity: ● Positive ○ Negative

□ Pan

Default: 90 degrees

Get What You Ask For



Higher Level Tools

- Provide assistance or some automation in developing UIs
- Four types
  - Language - high-level programming language
  - Application framework - for specific application domain
  - Model-based systems - driven by UML or DB Schema
  - Interactive GUI Builders - by far the most prevalent and accessable

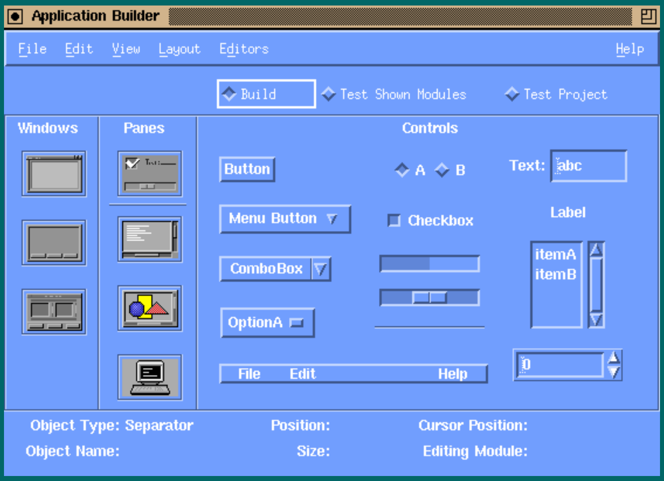
GUI Builder Tools

- Why build graphical (visual) interface with textual commands?
- Why not show what you want it to look like?
- Visual builder tools: Visual Basic, Visual C++, Borland Delphi, Symantec Café, NetBeans

Tool Methods

- Work area (interface being built)
- Drag and drop interactors/widgets onto work area
- Specify position, color, look, etc.
- Often provide Build/Test modes

Example: dtbuilder (Motif)



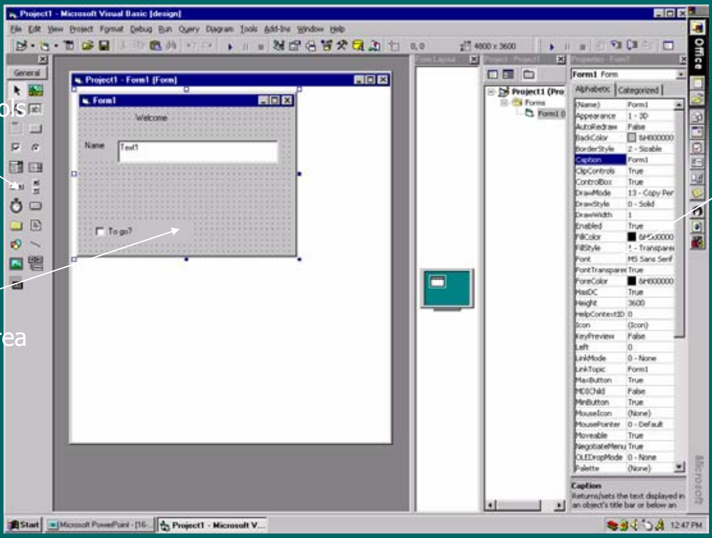
Object Type: Separator
Position:
Cursor Position:
Object Name:
Size:
Editing Module:

Example: Visual Basic

UI Controls

Design area

Control properties


Project1 - Microsoft Visual Basic [design]
File Edit View Project Format Debug Run Query Diagram Tools Add-Ins Window Help
General
Project1 - Form1 (Form)
Form1
Welcome
Name
Text1
To go?
Form Layout
Project - Project1
Project1 (Pro
Forms
Form1
Properties - Form1
Office
Microsoft
Start
Microsoft PowerPoint - [16...
Project1 - Microsoft V...
12:47 PM

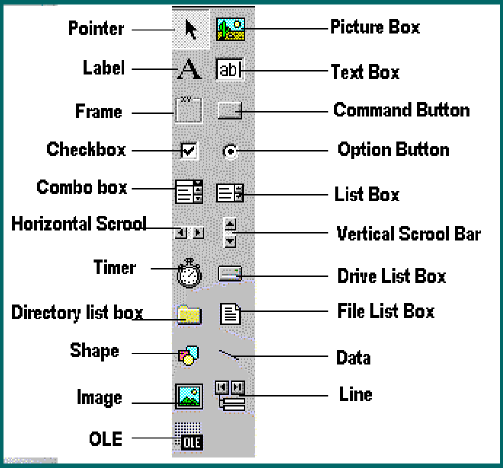
Form1 Form

Alphabetic | Categorized

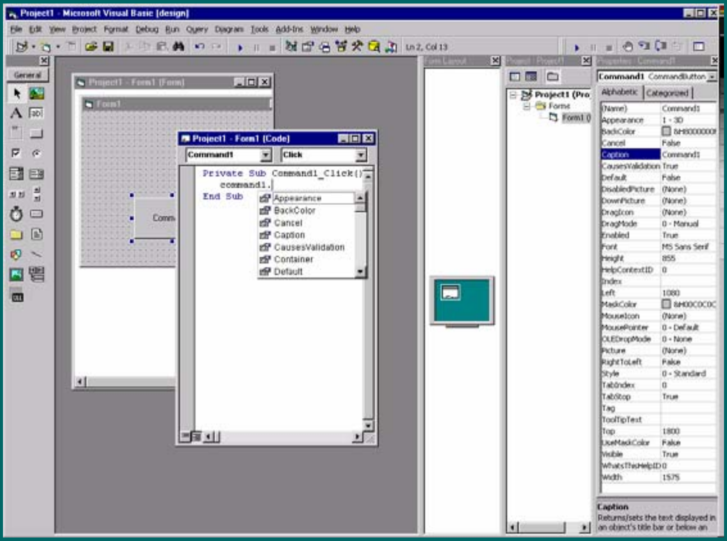
(Name)	Form1
Appearance	1 - 3D
AutoRedraw	False
BackColor	&H800000
BorderStyle	2 - Sizable
Caption	Form1
ClipControls	True
ControlBox	True
DrawMode	13 - Copy Per
DrawStyle	0 - Solid
DrawWidth	1
Enabled	True
FillColor	&H000000
FillStyle	1 - Transpare
Font	MS Sans Serif
FontTransparen	True
ForeColor	&H800000
HasDC	True
Height	3600
HelpContextID	0
Icon	(Icon)
KeyPreview	False
Left	0
LinkMode	0 - None
LinkTopic	Form1
MaxButton	True
MDIChild	False
MinButton	True
MouseIcon	(None)
MousePointer	0 - Default
Moveable	True
NegotiateMenu	True
OLEDropMode	0 - None
Palette	(None)

Caption
Returns/sets the text displayed in an object's title bar or below an

Widgets in VB Toolbox



Connecting Code to Widgets



```
Private Sub Command1_Click()
    command1.
End Sub
```

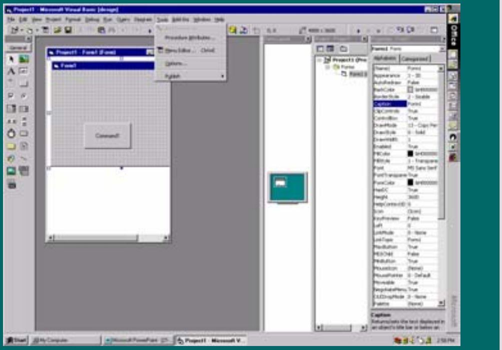
Command1 CommandButton

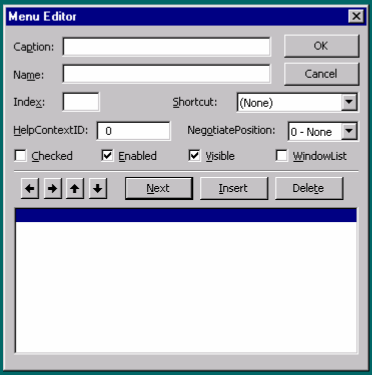
Alphabetic | Categorized

(Name)	Command1
Appearance	1 - 3D
BackColor	&H8000000F
Cancel	False
Caption	Command1
CausesValidation	True
Default	False
DisabledPicture	(None)
DownPicture	(None)
DragIcon	(None)
DragMode	0 - Manual
Enabled	True
Font	MS Sans Serif
Height	855
HelpContextID	0
Index	
Left	1080
MaskColor	&H00C0C0C
MouseIcon	(None)
MousePointer	0 - Default
OLEDropMode	0 - None
Picture	(None)
RightToLeft	False
Style	0 - Standard
TabIndex	0
TabStop	True
Tag	
ToolTipText	
Top	1800
UseMaskColor	False
Visible	True
WhatsThisHelpID	0
Width	1575

Caption
Returns/sets the text displayed in an object's title bar or below an

Making Menus





Caption:

Name:

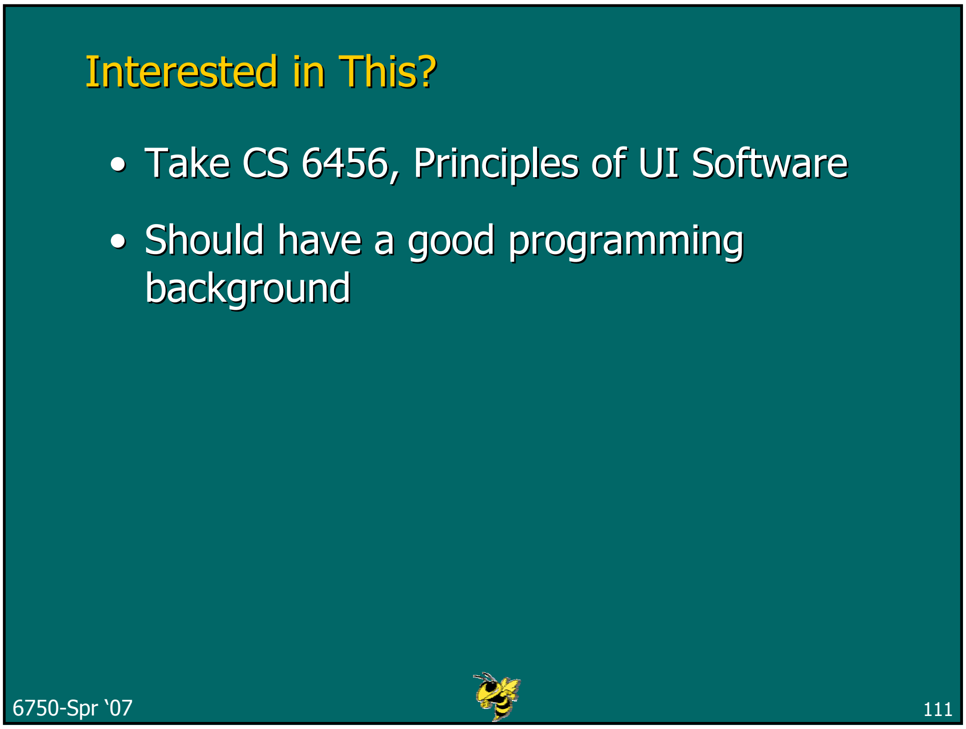
Index:

Shortcut: (None)

HelpContextID: 0

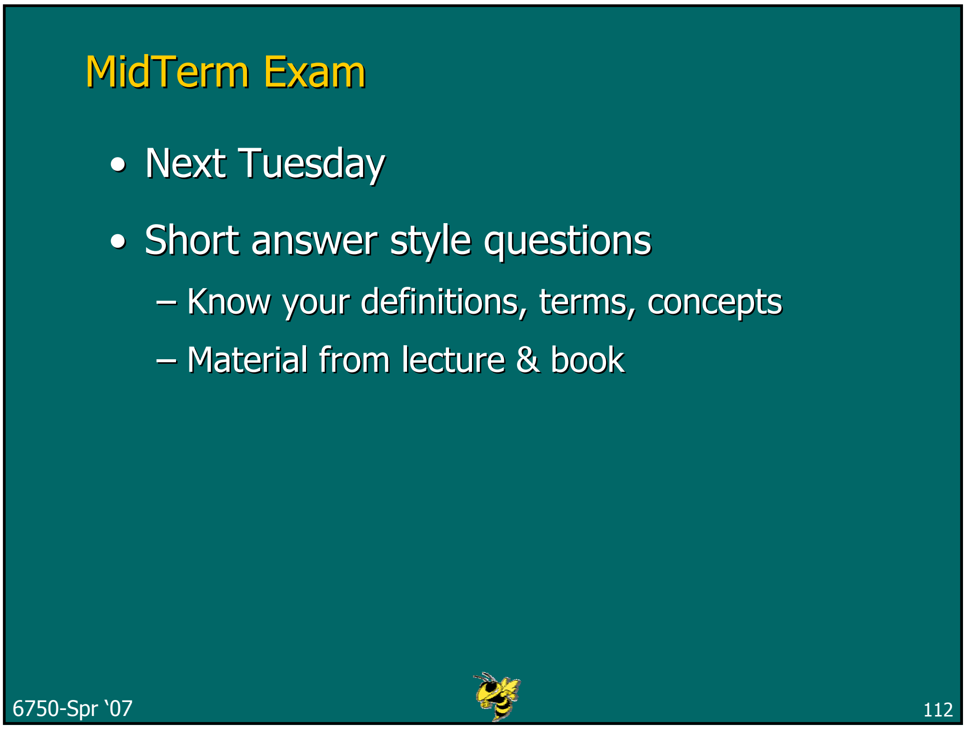
NegotiatePosition: 0 - None

□ Checked ☑ Enabled ☑ Visible □ WindowList



Interested in This?

- Take CS 6456, Principles of UI Software
- Should have a good programming background



MidTerm Exam

- Next Tuesday
- Short answer style questions
  - Know your definitions, terms, concepts
  - Material from lecture & book

Poster Session

- Thursday during class
- Buy a poster board (bookstore)
- Display your design ideas
  - Lots of pictures
  - Explanatory text
- Get some good feedback

Upcoming

- Mid-term Exam
- Poster Session
- Dialog styles