Lecture 21:

Interactive Tools: Prototypers (HyperCard, Director, Visual Basic, Balsamiq), Interface Builders, and Sketching Tools

05-431/631 Software Structures for User Interfaces (SSUI)

Fall, 2021

Logistics

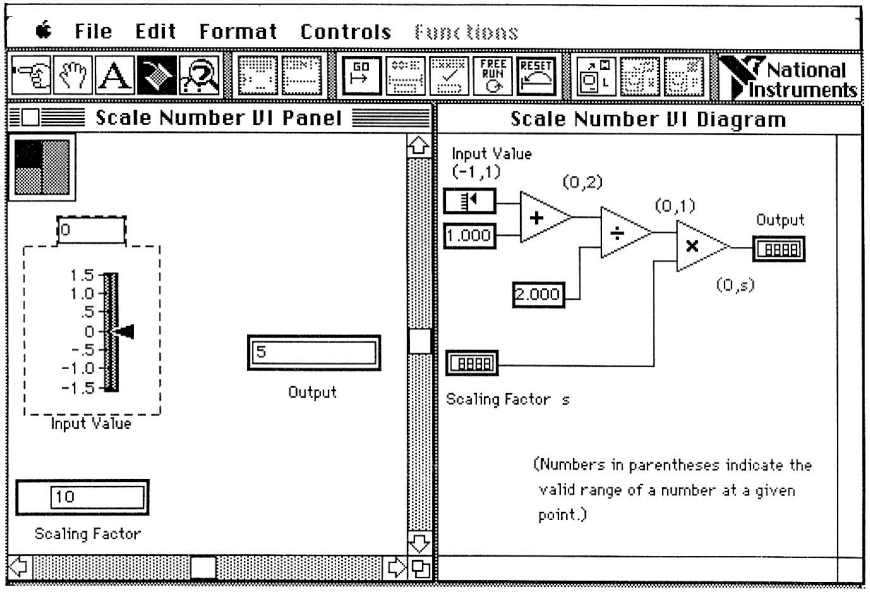
- HW6 due next Thursday, 11/18/2021, but need to work on arranging the final projects in parallel
- Initial project ideas are due TOMORROW
  - Lots of ideas here: https://www.cs.cmu.edu/~bam/uicourse/05631fall2021/FinalProject/index.html
  - Enter proposals into Piazza before Friday, 11/12/2021 at 3:05pm (in parallel with working on HW6)
  - Try to form groups around the projects on Piazza
  - We will finalize groups on Thursday, 11/16 in class, so you can get started right away (when finished with HW6)

Interactive Tools (review from Lecture 8)

- *Not* a programming interface
- Supports designers who might not be programmers
- Select widgets and place them
  - Layout, possibly with constraints
  - Specify properties of widgets
- Two categories:
  - GUI Tools – create representations used by the real code
    - Often built into IDEs
  - Prototypers – just to work out look and feel, and must be re-implemented
- Examples:
  - Adobe Dreamweaver for web pages
  - Resource editors & builders: Eclipse, Xcode IB, Android studio, Microsoft Visual Basic IDE
  - Prototypers: Balsamiq, Axure, etc.

Definition, cont.

- Tools that *use* graphical techniques to *specify* UI
- Usually focus on graphical parts of UI
- *Not* same as “visual” or “graphical programming”
  - Use graphics for the *code*



Interface Builders (IB)

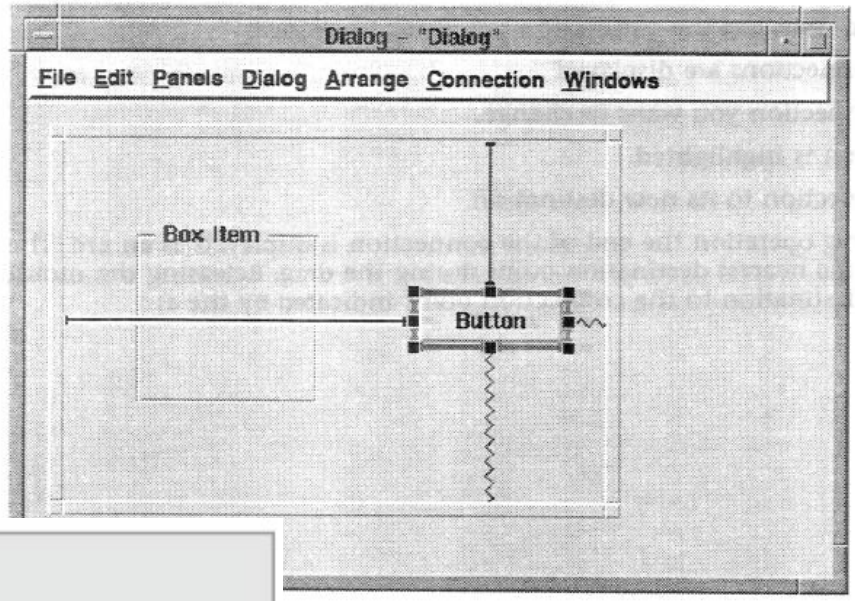
- Also called Interface Development Tools (IDTs) or GUI Builders or “Resource Editors” or “Form Editors”
- First = ResEdit on original Macintosh (1984)
- Lay out widgets to make dialog boxes, menus.
- Have a palette or menu of kinds of widgets
- Select widget, place with mouse in a window
- Set some properties
- Design menus, palettes, dialog boxes, controls
- Put in “graphics” pane for main application window
- Easy to use, but limited
- Connect call-backs with each widget
- Generates C code directly or intermediate language
- Sometimes connected to an interpreter so can execute call-backs.
  - If not, some call-backs can be simulated, e.g., transition to another window; pop-up error

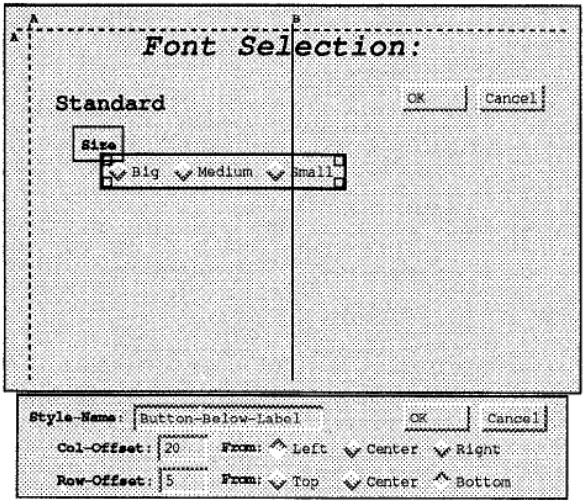
Interface Builders, cont.

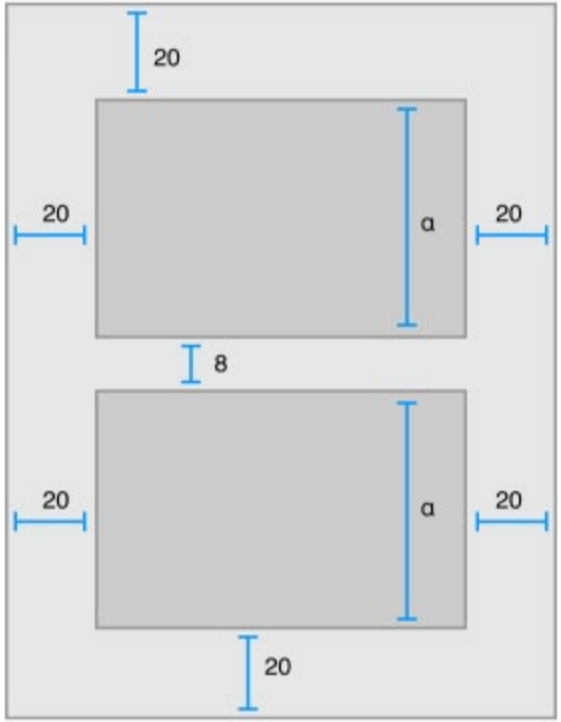
- Layout mechanisms
  - See lectures 10 (Geometry Management) and 17 (Constraints)
  - Usually a complication
    - X's row and columns stuff
    - Galaxy's struts and springs
    - Java's Layout Managers
- "Resources" (lecture 10)
  - store information in special files rather than in source code
  - positions, colors, text labels, etc.
  - allow for easier modification for users, internationalization, etc.
- IBs Usually don't support:
  - Error checking of values, e.g. for text input fields
  - Graying of widgets depending on values and other widgets
  - Default values of widgets
  - Dynamic changing of widgets (e.g., add more items)
  - Dynamic changing layers (groups) of widgets (visibility) depending on values and other widgets
  - Any dynamically created graphical objects.

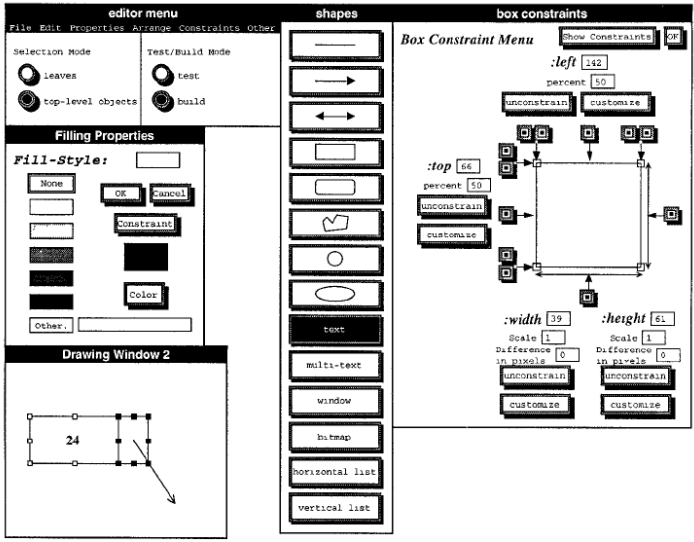
Examples from previous lectures

- Struts and springs
- Gilt's graphical tabs
- iOS Auto Layout
- Lapidary constraints





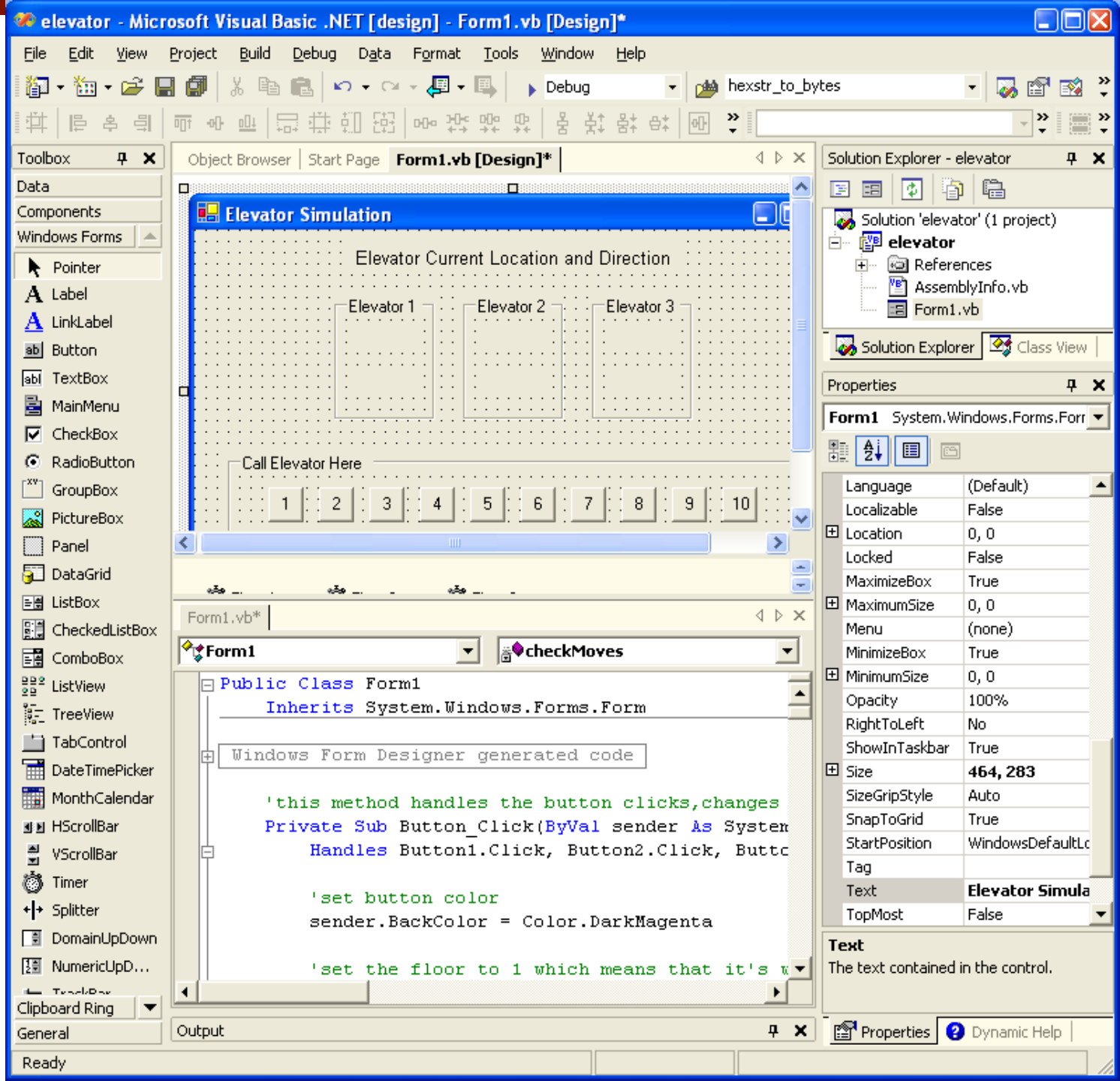




Interface Builders, cont.

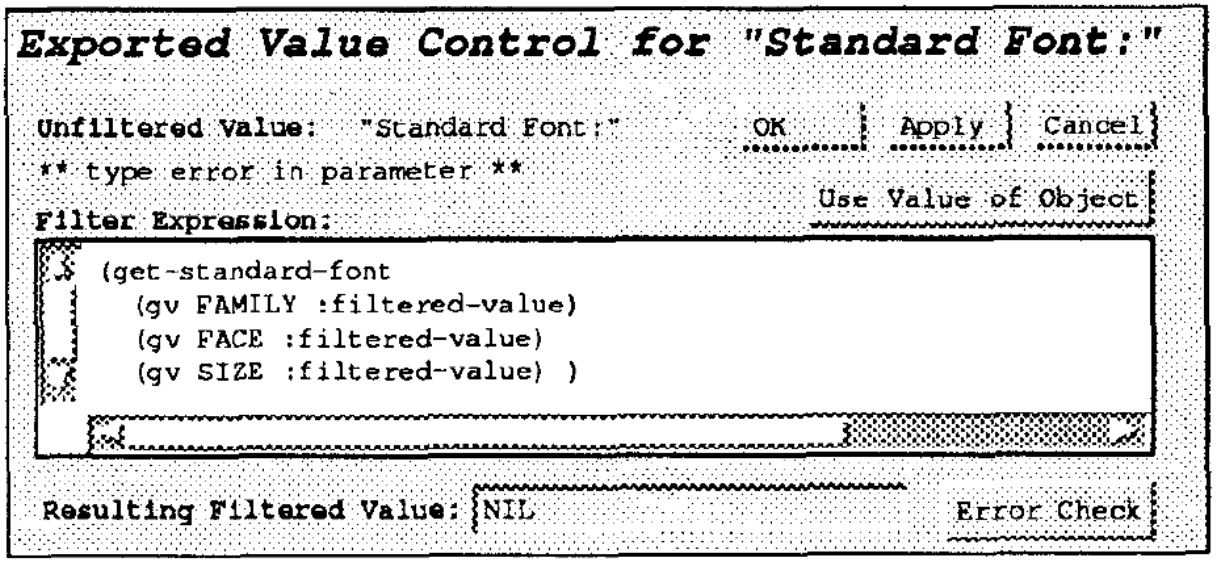
- Examples:
  - See “Card” systems, e.g., Menulay (1983-research system)
  - NeXT Interface Builder (NeXT) - 1988 popularized the name
    - By Jean-Marie Hullot who had an IB in Lisp at INRIA in France
      - Started in 1984, finished in 1986, used a Macintosh
      - Key innovation – binding between UI and source code
  - Visual Basic
    - First released in 1991 on Windows 3.0
    - Originally for End-User Development (lecture 23) but gave up in 2002
  - Resource editors in programming environments
- Used to be lots of IB products
  - Used to be many commercial tools are in this category; over 100
    - See my old list (1997): http://www.cs.cmu.edu/~bam/toolnames.html
  - Most went out of business
  - Microsoft, MetroWorks, etc. include “resource editors” for “free”

VB Screen


elevator - Microsoft Visual Basic .NET [design] - Form1.vb [Design]*
File Edit View Project Build Debug Data Format Tools Window Help
Debug
hexstr_to_bytes
Toolbox
Data
Components
Windows Forms
Pointer
Label
LinkLabel
Button
TextBox
MainMenu
CheckBox
RadioButton
GroupBox
PictureBox
Panel
DataGrid
ListBox
CheckedListBox
ComboBox
ListView
TreeView
TabControl
DateTimePicker
MonthCalendar
HScrollBar
VScrollBar
Timer
Splitter
DomainUpDown
NumericUpD...
Clipboard Ring
General
Object Browser
Start Page
Form1.vb [Design]*
Elevator Simulation
Elevator Current Location and Direction
Elevator 1
Elevator 2
Elevator 3
Call Elevator Here
1 2 3 4 5 6 7 8 9 10
Form1.vb*
Form1
checkMoves
Public Class Form1
Inherits System.Windows.Forms.Form
Windows Form Designer generated code
'this method handles the button clicks,changes
Private Sub Button_Click(ByVal sender As System
Handles Button1.Click, Button2.Click, Buttc
'set button color
sender.BackColor = Color.DarkMagenta
'set the floor to 1 which means that it's w
Output
Solution Explorer - elevator
Solution 'elevator' (1 project)
elevator
References
AssemblyInfo.vb
Form1.vb
Solution Explorer
Class View
Properties
Form1 System.Windows.Forms.Forr
Language (Default)
Localizable False
Location 0, 0
Locked False
MaximizeBox True
MaximumSize 0, 0
Menu (none)
MinimizeBox True
MinimumSize 0, 0
Opacity 100%
RightToLeft No
ShowInTaskbar True
Size 464, 283
SizeGripStyle Auto
SnapToGrid True
StartPosition WindowsDefaultLc
Tag
Text Elevator Simula
TopMost False
Text
The text contained in the control.
Properties
Dynamic Help
Ready

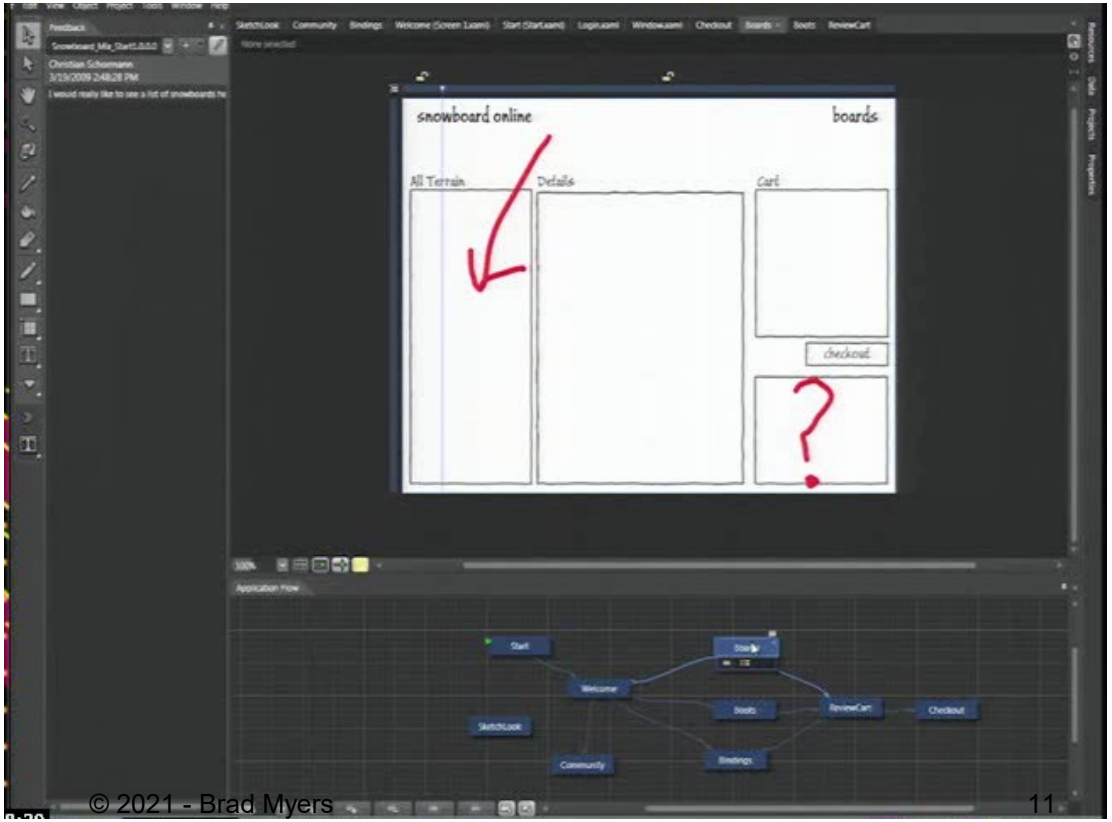
Some Research in IB

- Garnet's GILT interface builder:
  - Eliminating Call-backs (UIST'91)
    - http://doi.acm.org/10.1145/120782.120805 or video (5min) or **video of both** (9 min)
    - Handles error checking, data transformations, connections of widgets to each other



Microsoft's Expression Blend

- Microsoft Silverlight Blend's SketchFlow
  - http://channel9.msdn.com/Events/MIX/MIX09/C01F (1 hour video)
- 2006-2012
- Behaviors, etc. as well
- Landday says this has "sketching" (see 3/19/09 blog)
- Now discontinued
  - Some features put into Visual Studio

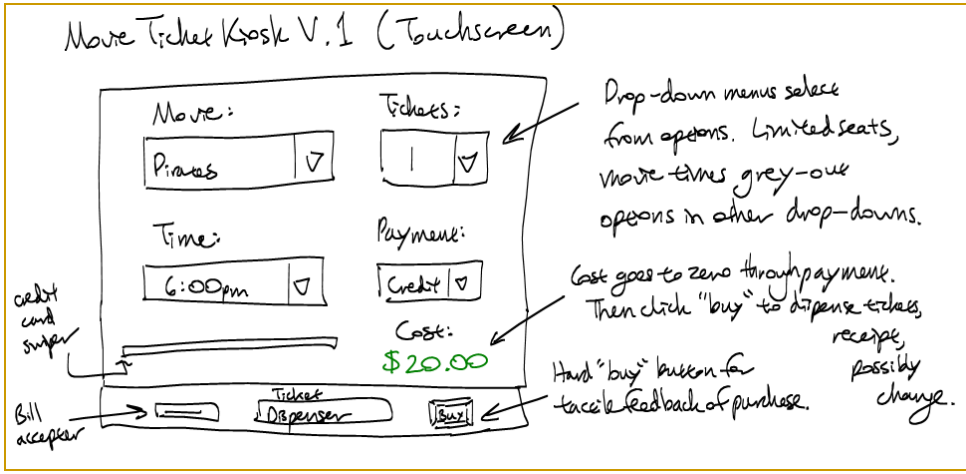


Prototyping Tools

- Just show what looks like
  - Storyboard of screens
- “Wireframing tools”, “Click-through prototypes”
- Note: differentiate from term "rapid prototyping“
- Some support for behavior: typically changing screens
- Like a movie of the interaction
- Goal: see some of interface very quickly (hours)
- Often no possibility of migrating to real application
- May not use "real" widgets
- "Low Fidelity" Techniques

Low Fidelity Prototyping

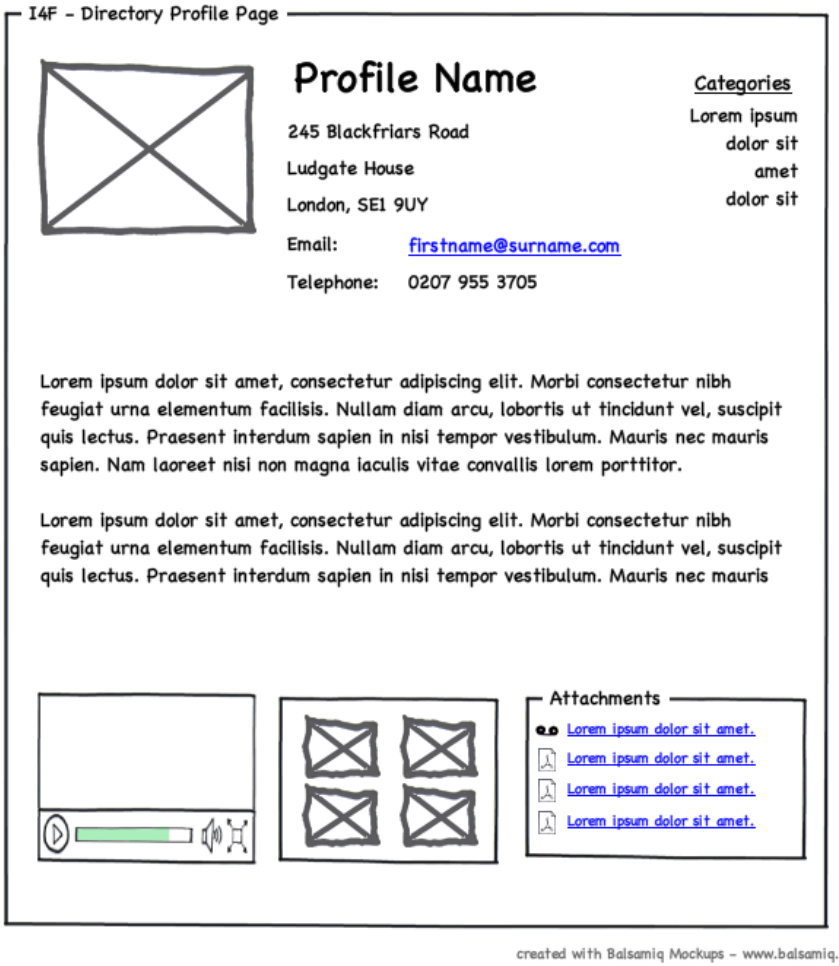
- Just use paper and/or overheads
  - No tools
  - Experimenter "plays computer"
  - Ask the user "what would you do now“
  - Experimenter shows the computer's expected response
  - Very cheap and easy and gets surprisingly good results



  - Find out if users understand organization, how to find desired operations, if understand menu names, etc.
  - Easy to change between sessions
- Can make a movie of the paper using a regular video camera
  - To demonstrate/explain the interface

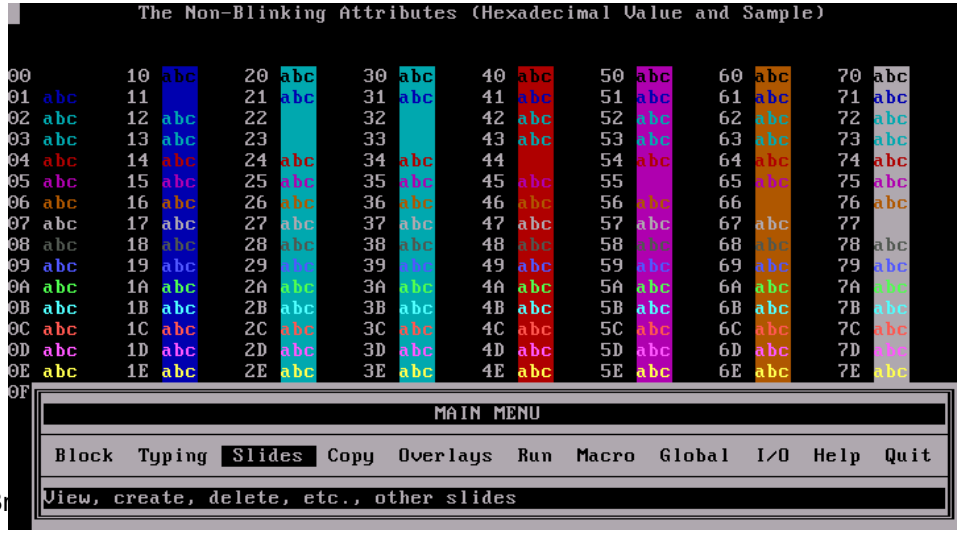
Low Fidelity Examples

- "Wireframes" since often just draw the outlines



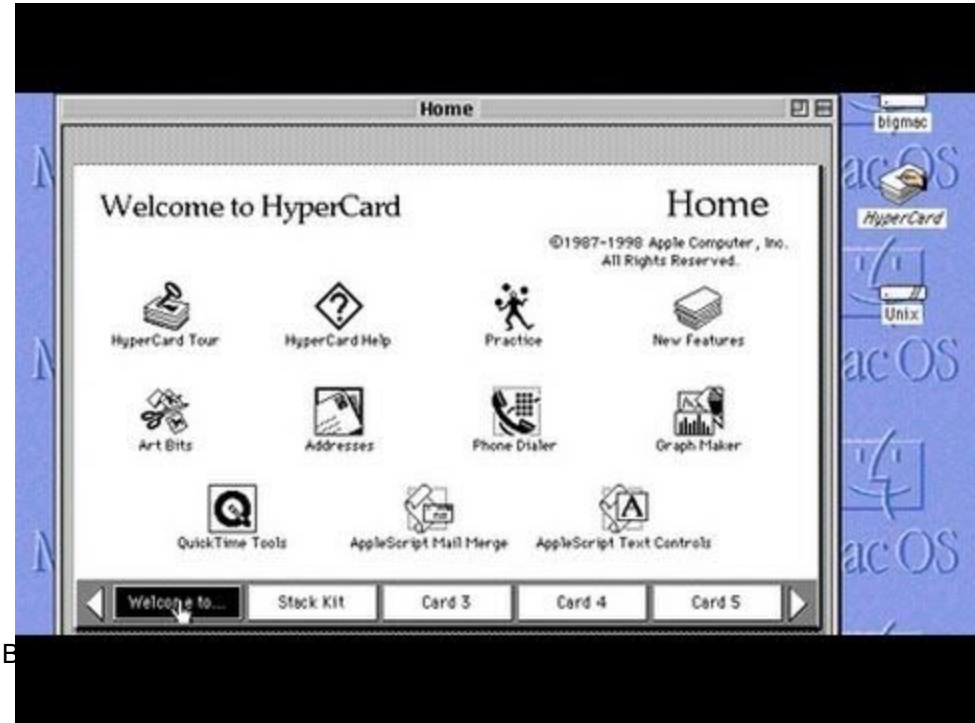
Early Prototypers

- For Character Screens
  - 24x80 DOS, often no mouse (like terminal / console)
  - Especially for forms-based applications
  - Examples: Dan Bricklin's Demo-It (Windows v2.0 ~1987), Protoscreens for PCs from Bailey&Bailey (~1990)
  - Specify characters for each position of screen, or a "character graphics"
  - Can specify fields that are editable text
  - Can specify that clicking on an area cause changing to a new screen.
  - Also menus



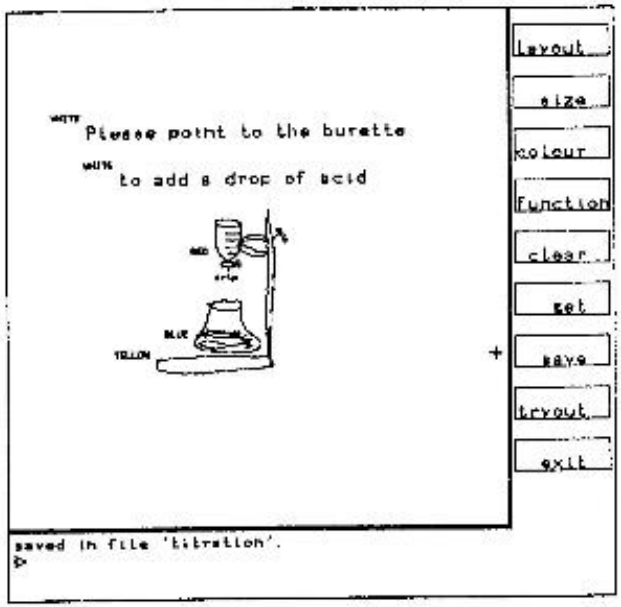
Card Programs as Prototypers

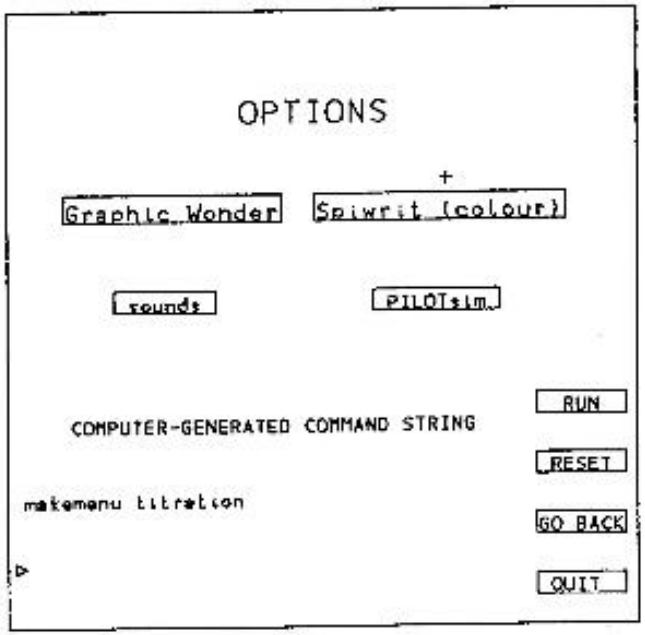
- Card Programs
  - Examples:
    - HyperCard (1987) and SuperCard for Mac
    - OWL's GUIDE for PCs (gone?)
    - Toolbook (formerly from Asymetrix then Click2Learn, then SumTotal Systems, Inc. http://www.toolbook.com , now EOL)
  - Sequence of cards
  - Click-through prototypes
  - Paint program (not "draw")
  - Draw pictures on each card
  - May be multiple layers



Early Research Card Systems

- Menulay
  - Buxton, Siggraph'83 pp. 31-38
    - http://www.billbuxton.com/menulay.pdf
    - http://www.youtube.com/watch?v=Kt0oAg0haU0
  - vector screens, widgets, sounds, text, output C code and tables
  - All actions (including transitions) required C programming





Early Research Card Systems, cont.

- Trillium
  - Henderson, CHI'86
  - http://doi.acm.org/10.1145/22339.22375
  - Xerox copier interfaces
  - Interpreted Lisp
  - Transitions defined using the interface

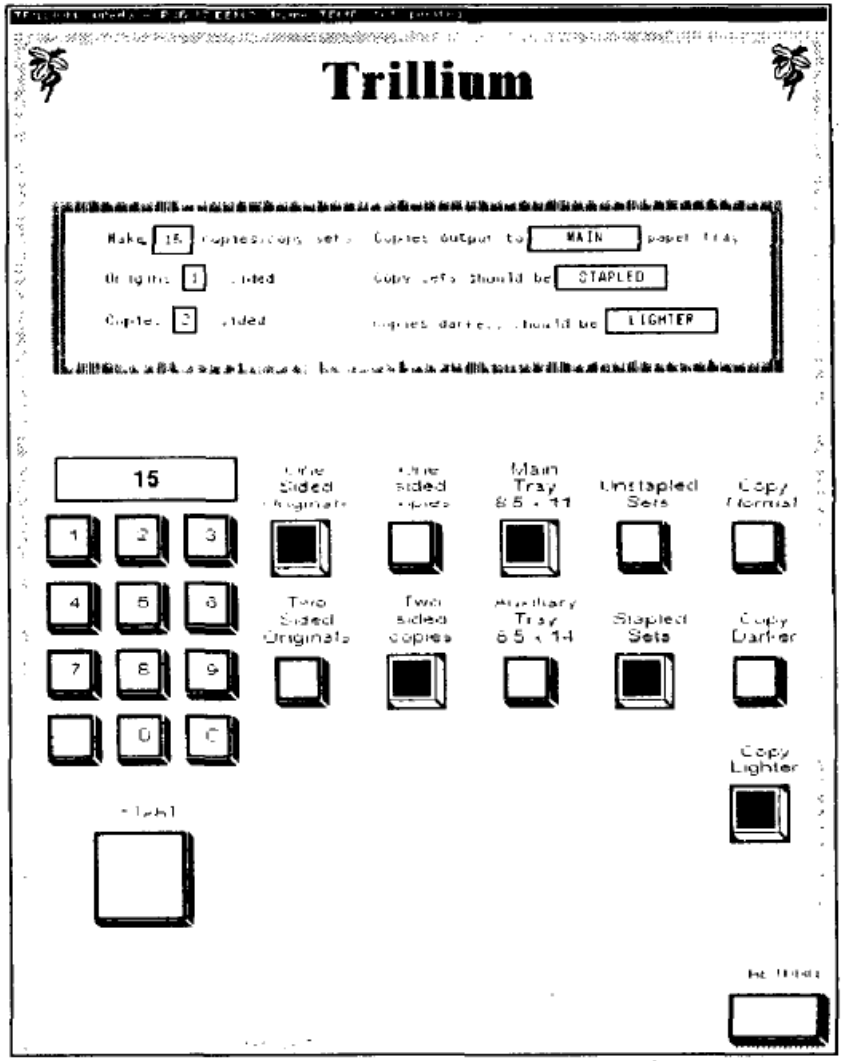
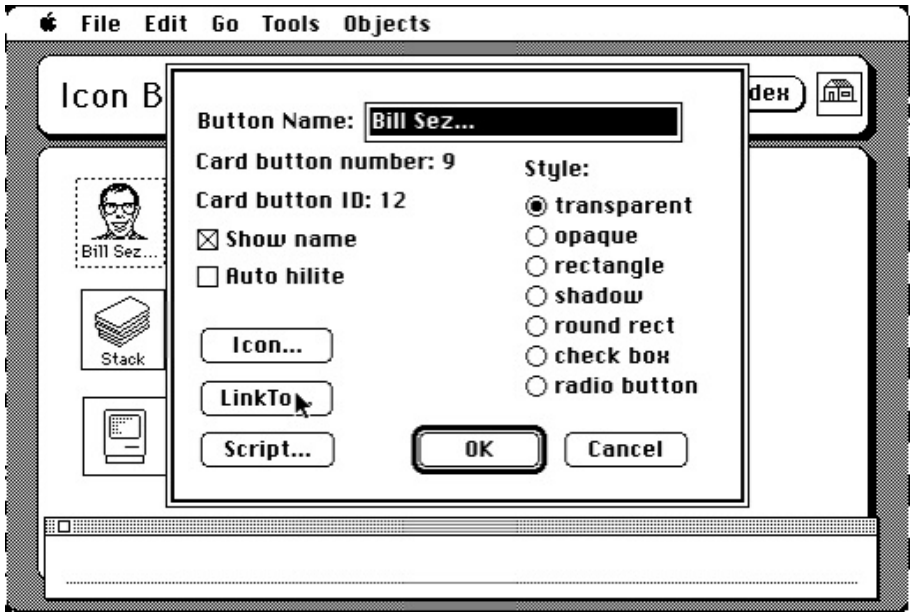
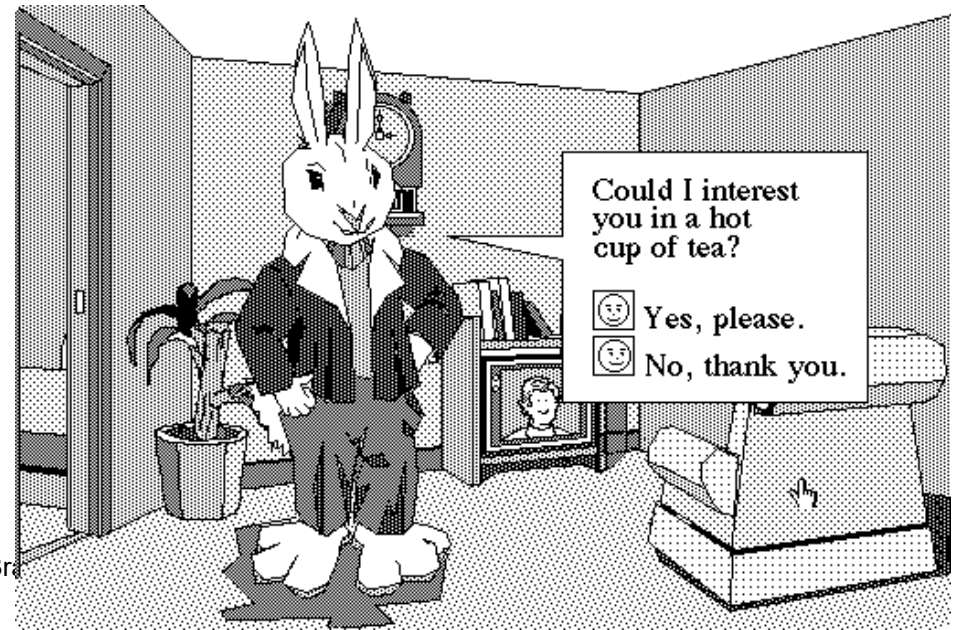


Figure 1: A single frame from an interface for a multi-functional office machine.

HyperCard

- 1987 – 2004



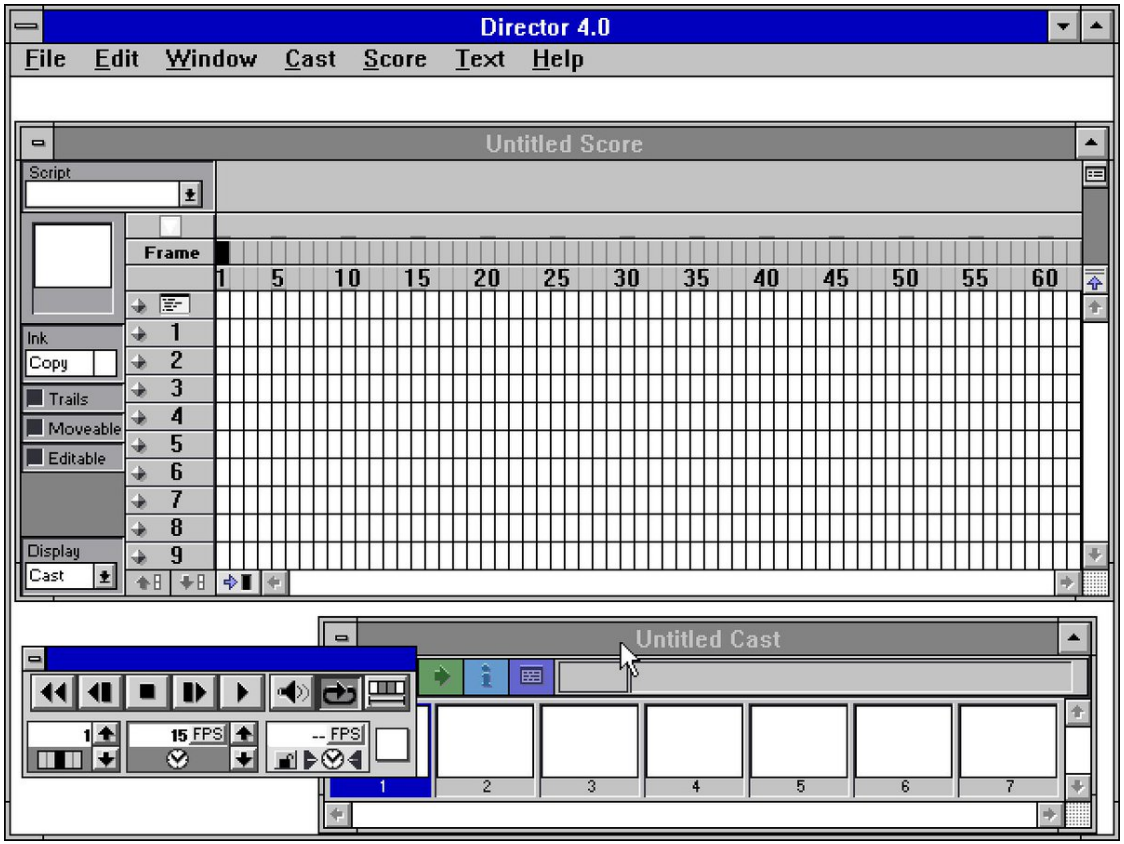


HyperCard, details

- Goal: programming for everyone
- Buttons can transition to another card
  - Fancy transitions
- Single window
- Buttons can start running a script ("HyperTalk")
  - Script can move objects, change cards, animate, compute, etc.
  - Code management: who changes what; finding the script
  - Not good for dynamically created graphics
- Complete control of individual pixels
  - Graphic designers have complete control
  - Design new widgets
- Can be "real" application if sufficient power/speed
  - Used for original Myst game, etc.
- See also Lecture 19 on EUP

Animation Programs

- Example: MacroMedia's Director (1987) – now Adobe
  - Replaced (pretty much) by Adobe Flash, now html
  - Discontinued January 27, 2017
- Also control individual pixels
- Individual paintings can be specified as animation element
  - E.g., characters
  - Each can be instantiated, moved, etc.
- Good control over timing, synchronization
- Scripting language
  - Can program that when a mouse button is clicked in an area, start an animation or transition
  - Scripting language even more primitive than HyperTalk
- Good for "Future Scenarios" when want good fidelity with real look
- Not for final (real) interface unless Multi-media



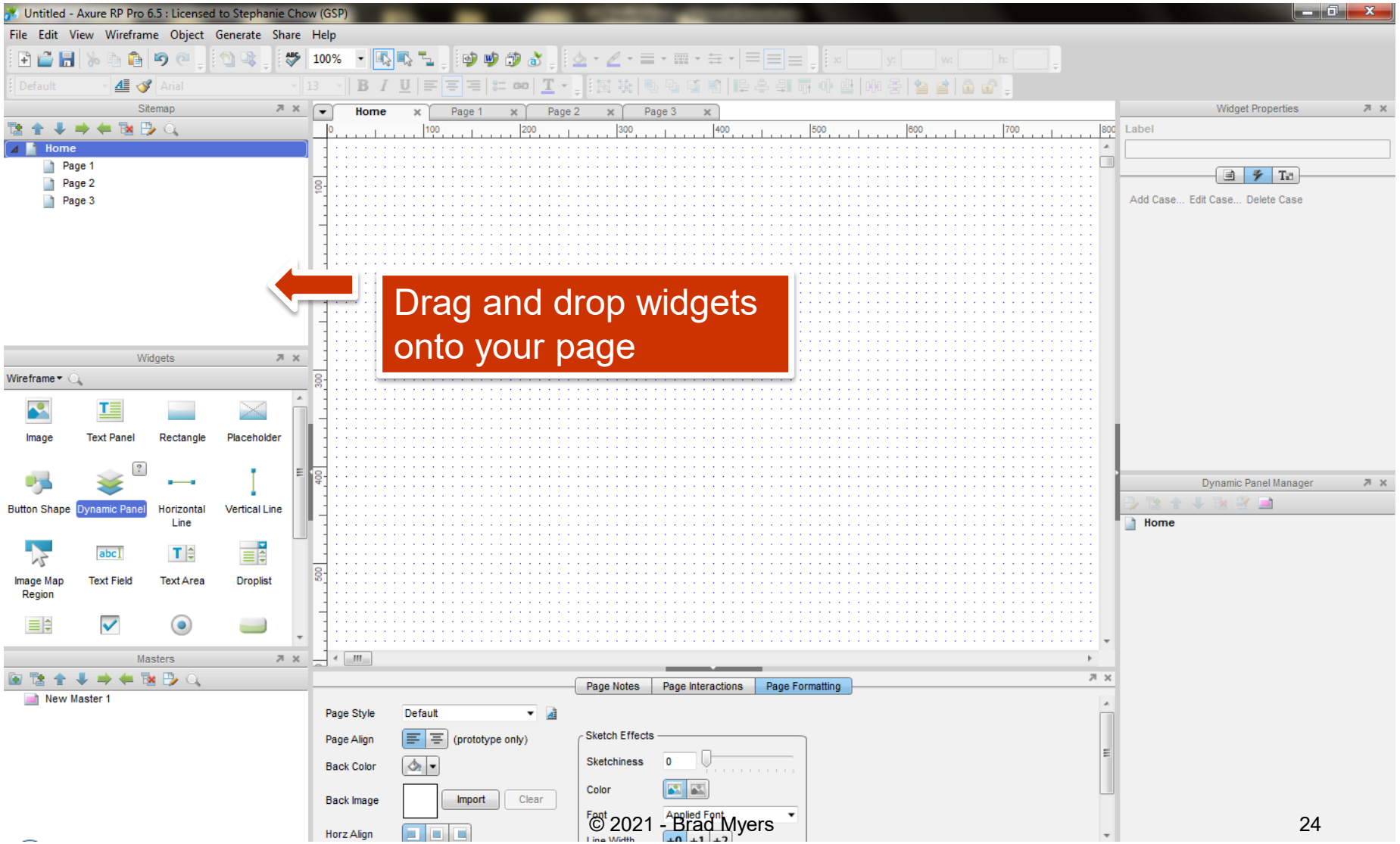
Commercial Prototypers

- Search for “Prototyping tools” or “Wireframing Tools”
- Here are some lists:
  - 2021: https://www.uxbooth.com/articles/10-best-prototyping-tools-for-ux-designers-in-2021/
  - 2018: https://medium.theuxblog.com/11-best-prototyping-tools-for-ui-ux-designers-how-to-choose-the-right-one-c5dc69720c47

Examples:

- Adobe XD
  - New, free and quite powerful
- Axure (downloaded)
- InVision
- Sketch (Mac only)
- On-line tools
  - Figma – good collaboration features (also downloadable)
  - Balsamiq (http://www.balsamiq.com/)
  - Just in Mind
  - Protopie https://www.protopie.io/
  - ….many others!

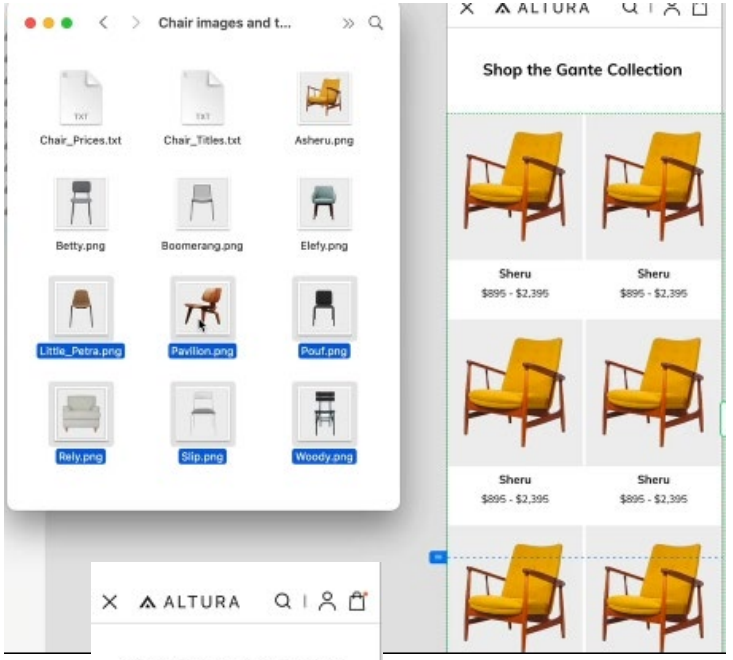
Axure

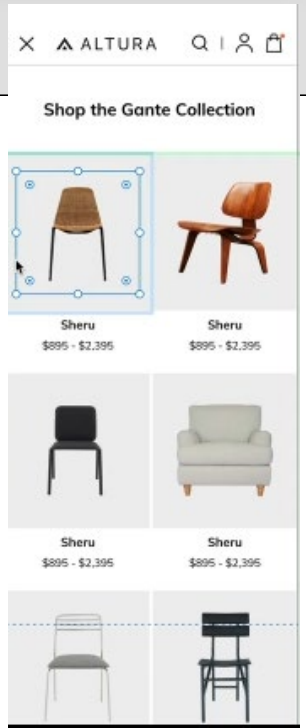


Adobe XD

- Pull in components from palette
- Lots of provided elements
  - Mimic any screen
  - Supports “design systems” if have company-specific requirements
  - Create “art boards” for each screen
  - Can keep track of previous versions, or options
- “Repeat Grid”
  - Very clever feature for lists, etc.
  - Can pull out as many as desired
  - Drag-and-drop lists of text/images, etc. onto grid

Adobe XD Repeat Grid

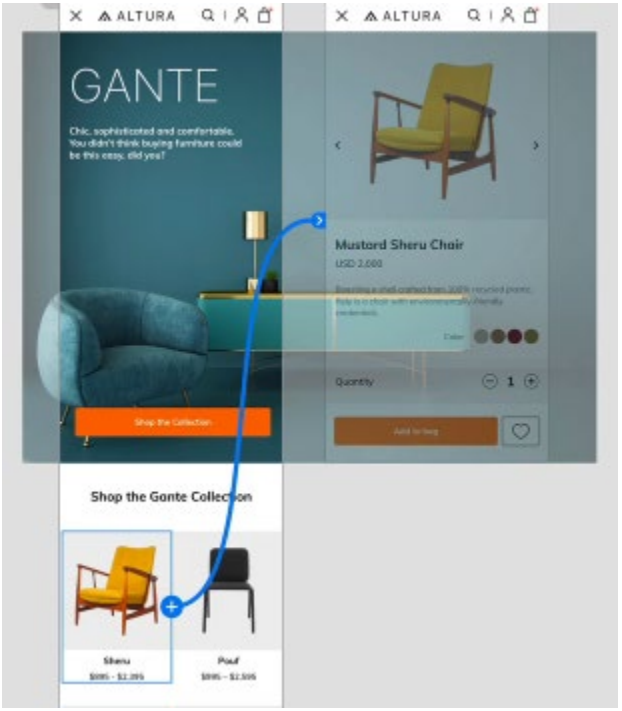


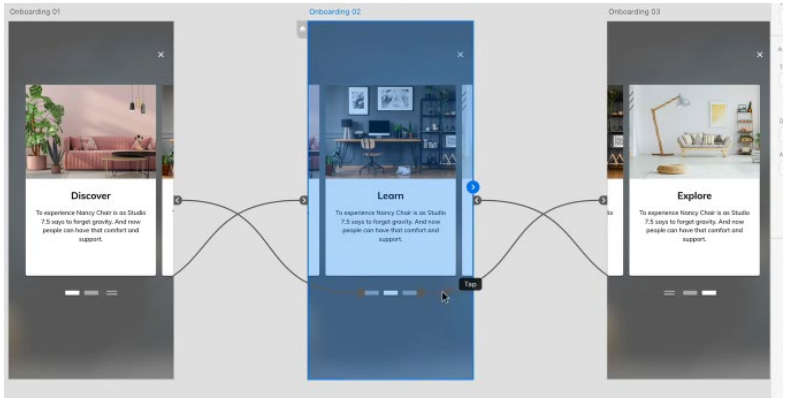


Source: https://www.adobe.com/products/xd/learn/get-started.html

Adobe XD Interactions

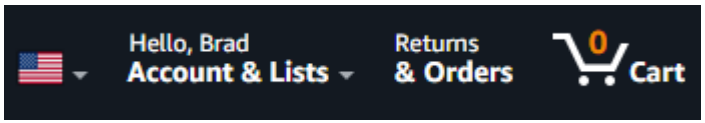
- Click-throughs by wiring click points to other “art boards”
  - Can trigger on tap or other events
  - Transition effects including animations, timing
  - Or wire to “previous artboard” or other
- “Auto-animate” – copy and paste, edit new one
  - E.g., position, size, opacity, color…
  - Trigger on click, etc.

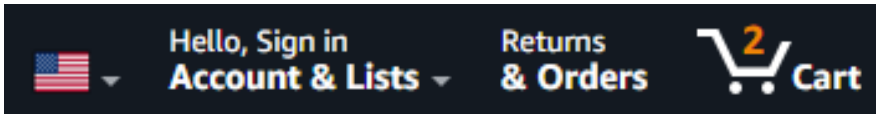




Adobe XD "Components"

- (Also available in Axure, but not simpler tools like Balsamiq)
- Create elements with internal behaviors that can reuse in multiple places
  - E.g., login/logout vs. cart on multiple web pages
  - E.g., can create custom button – change color, etc.
- Prototype ("main") and instances
  - Edit main and others change accordingly
  - Can override properties in instances will be retained

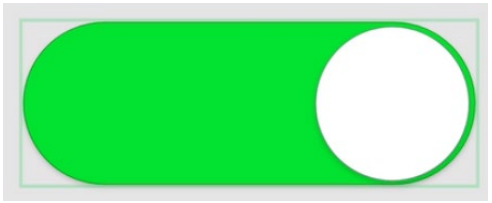


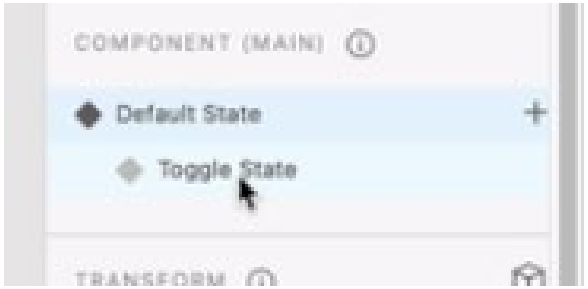


Adobe XD Component "states"

- Can define different states for a component
  - E.g., hover state, toggle state, user-defined
  - Different property values in different states
- Triggers can cause state change
- Can animate between states
- Reusable in all instances

Source:
https://www.adobe.com/products/xd/learn/prototype/component-states/component-states-common-use-cases.html



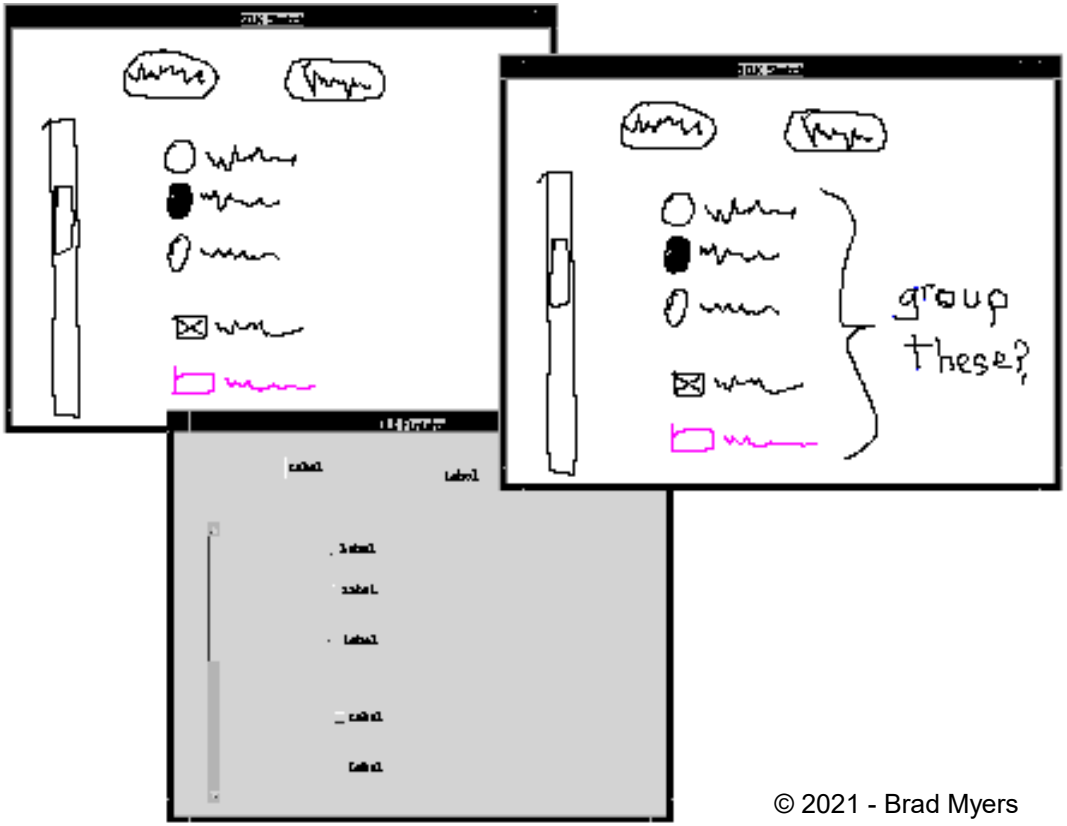


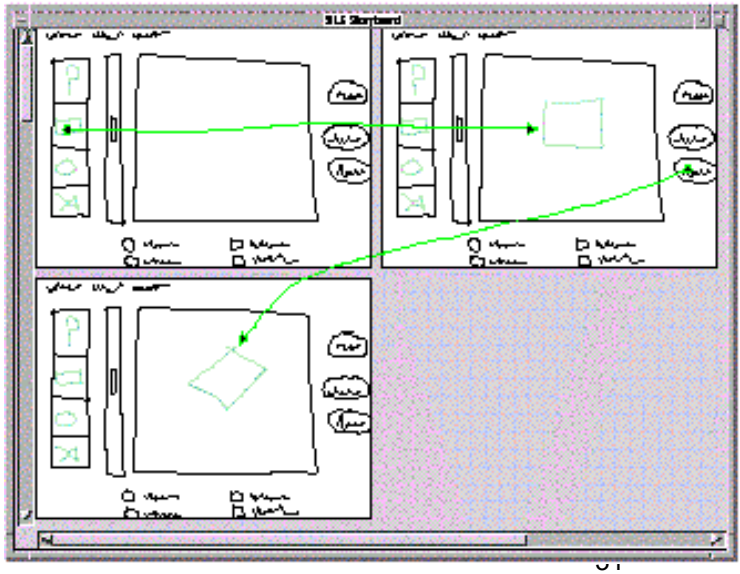
Research in Informal Prototyping Tools

- Sketching tools
- Use before interface builders
- Designed to help support the ideation phase
- Menulay (saw earlier)
- James Landay's SILK tool
  - Infer formal widgets and widget groupings from sketches
  - Convert to real widgets
  - Sketch storyboards for transitions

Research in Informal Tools

- Silk main paper: http://doi.acm.org/10.1145/223904.223910
- Video from CHI'96 (8:22 min)





Landay's later tool: Denim

- Denim and its video