

CS 338: Graphical User Interfaces

Lecture 3-1: Rapid Prototyping

Some materials adapted from James Landay's materials at UC Berkeley

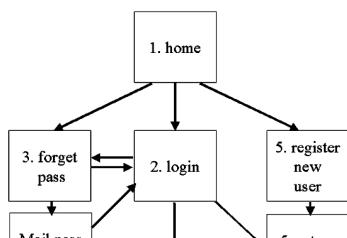
Rapid Prototyping

- What is prototyping?
 - rapid initial development, sketching & testing many designs to determine the best (few?) to continue w/ further design & implementation
- Why do we prototype?
 - get feedback on our design faster
 - saves time! saves money!
 - experiment with alternative designs
 - fix problems before code is written
 - keep the design centered on the user (who are we building it for)

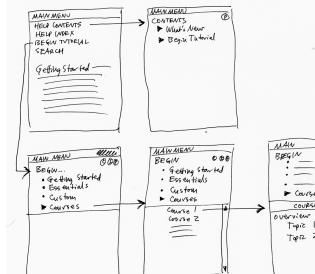
Progressive refinement

- Applications are iteratively refined
 - from less to more detail
 - from coarse to fine granularity
- Designers create representations of app, web sites, etc. at multiple levels of detail

Info Map



Storyboards



Schematics

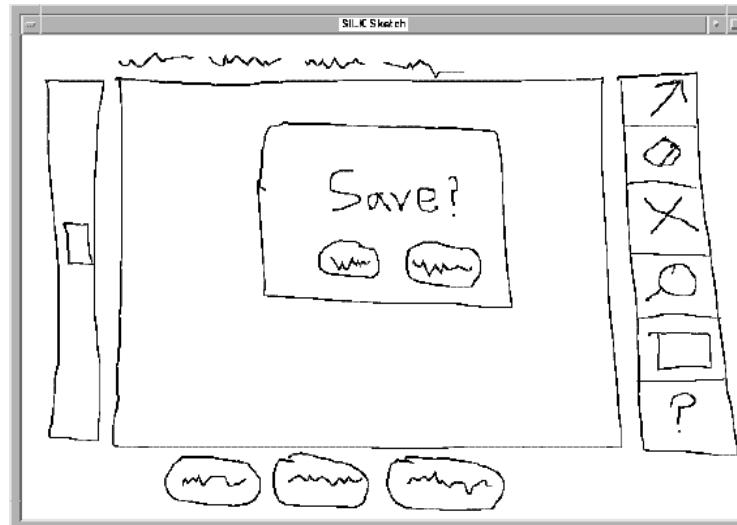
(Site Branding)	
Acme, Inc.	Contact email: [redacted] Search: [redacted]
Kids	(What this site is about. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.)
Outdoors	News Topic: <ul style="list-style-type: none">This month's news release (date)
Catalogue	News Topic: <ul style="list-style-type: none">This month's news release (date)
Travel	News Topic: <ul style="list-style-type: none">This month's news release (date)
Features	News Topic: <ul style="list-style-type: none">This month's news release (date)
About This Site	News Topic: <ul style="list-style-type: none">This month's news release (date)
(global nav bar)	News Topic: <ul style="list-style-type: none">This month's news release (date)

Mock-ups



Levels of prototyping

- Fidelity refers to the level of detail
- High fidelity
 - prototypes look like the final product
- Low fidelity
 - artists renditions with many details missing

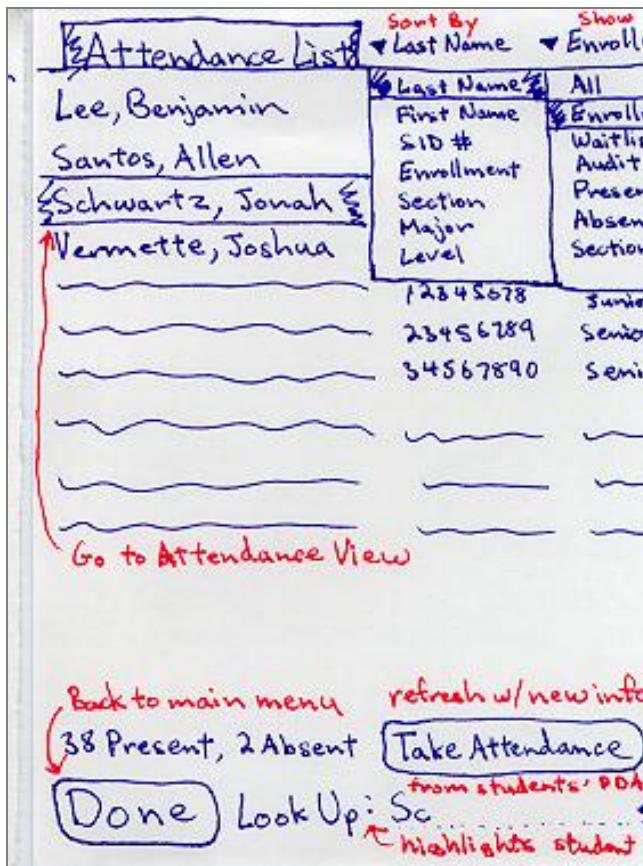


Why create low-fidelity prototypes?

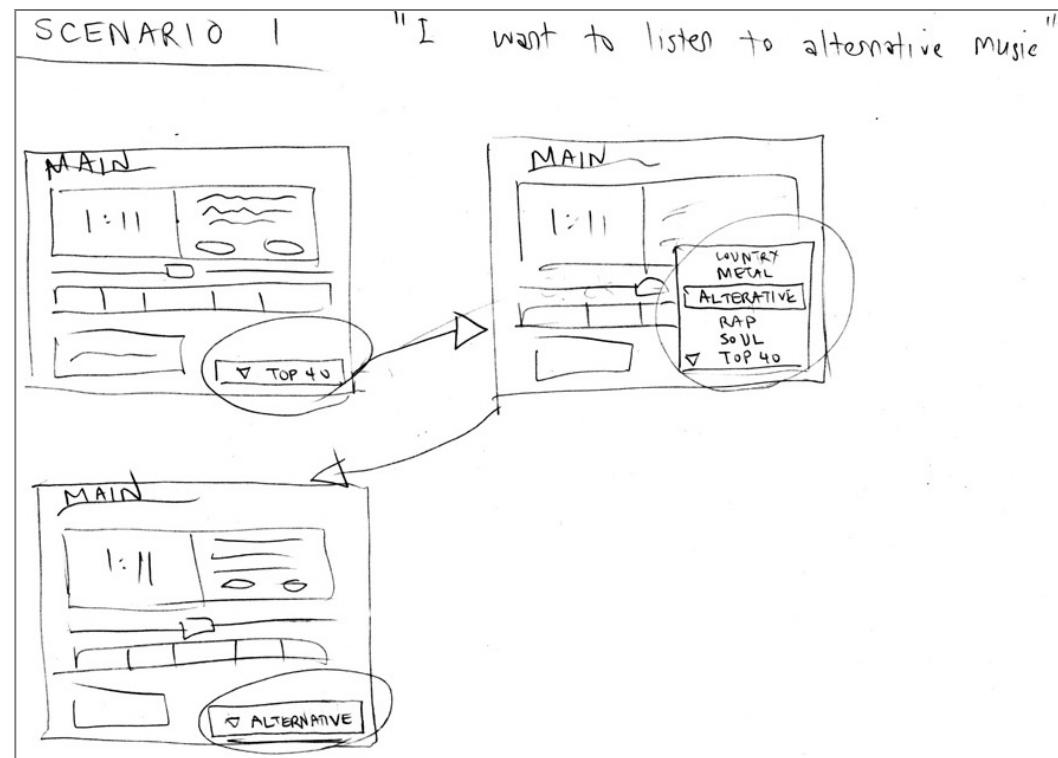
- Traditional methods take too long
 - sketches -> implementation -> evaluate -> iterate
- Can simulate the prototype
 - sketches -> evaluate -> iterate
 - sketches act as prototypes
 - designer “plays computer”
 - other design team members observe & record
- Kindergarten implementation skills (?!)
 - allows non-programmers to participate

Paper sketches

Course manager



Music player

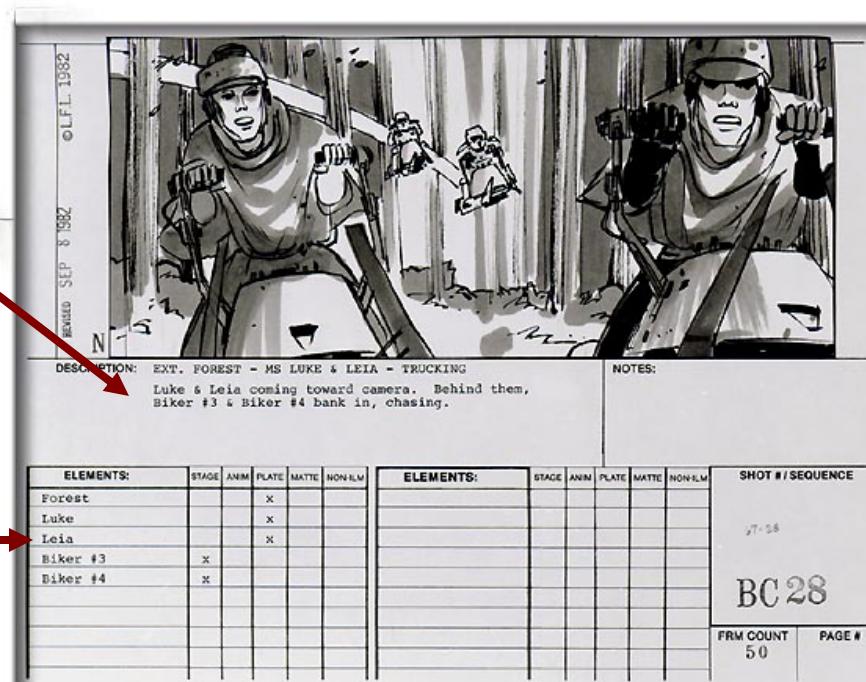


Storyboards

- What are “storyboards”?
 - used in film & animation
 - provide a “script” of important events
 - leave out the details, concentrate on key interactions

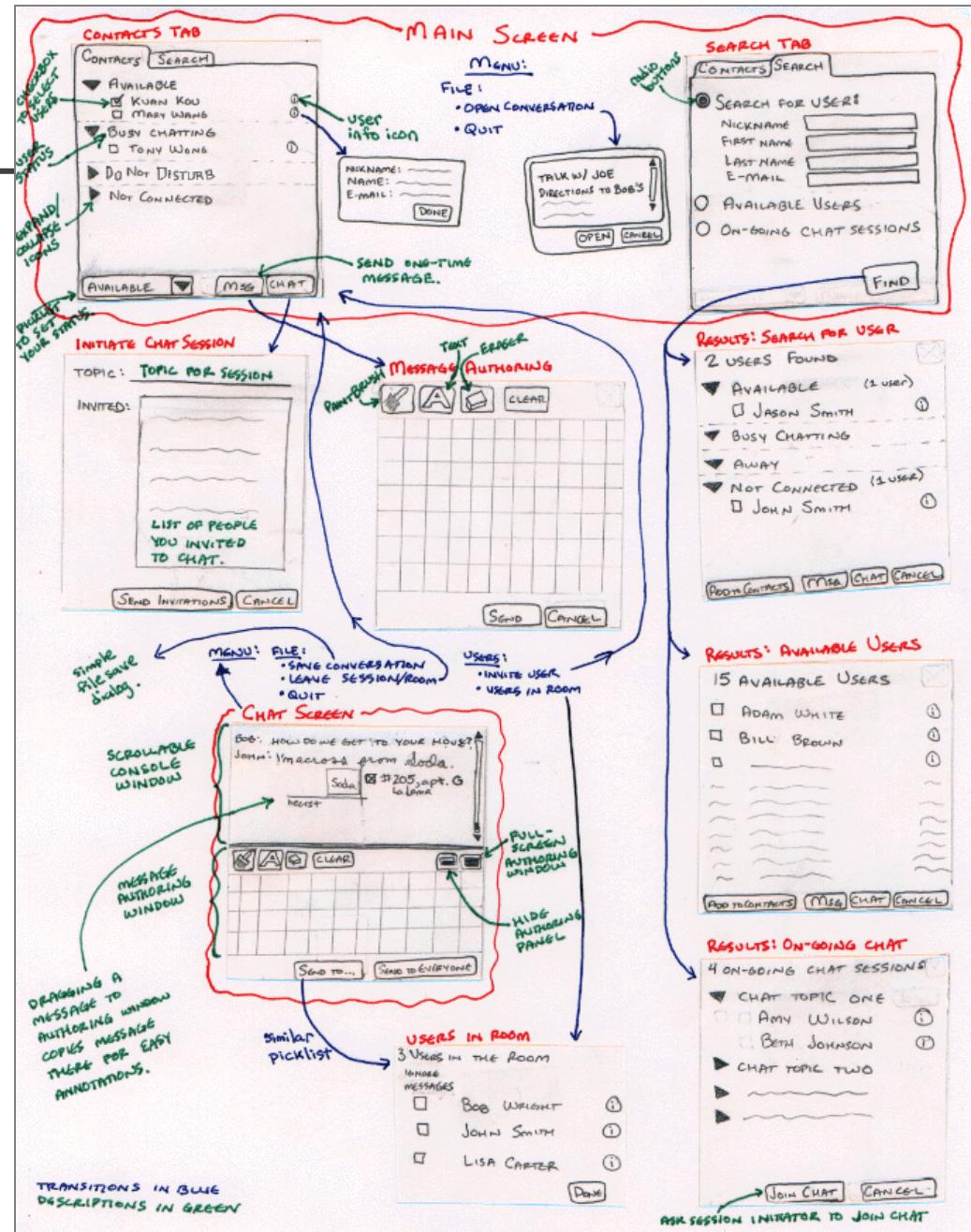
“Luke & Leia coming toward camera. Behind them, Biker #3...”

“Elements:
Luke
Leia
”
...”



Storyboards

- Storyboards for designing GUIs
 - create / pin-up lots of screens
 - specify user interaction by associating screens



CONTACTS TAB

CONTACTS **SEARCH**

- ▼ AVAILABLE
 - KUAN KOU
 - MARY WANG
- ▼ BUSY CHATTING
 - TONY WONG
- Do Not Disturb
- Not Connected

AVAILABLE **MSG** **CHAT**

MAIN SCREEN

menu:

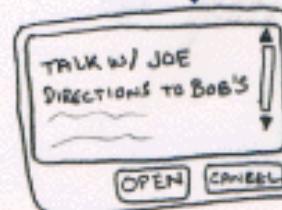
FILE:

- OPEN CONVERSATION
- QUIT

user info icon

NICKNAME: _____
NAME: _____
E-MAIL: _____
DONE

radio button



SEARCH TAB

CONTACTS SEARCH

SEARCH FOR USER:

NICKNAME	_____
FIRST NAME	_____
LAST NAME	_____
E-MAIL	_____

- AVAILABLE USERS
- ON-GOING CHAT SESSIONS

FIND

INITIATE CHAT SESSION

TOPIC: TOPIC FOR SESSION

INVITED:

LIST OF PEOPLE YOU INVITED TO CHAT.

SEND INVITATIONS **CANCEL**

PRINT PREVIEW **MESSAGE** **ERASER** **AUTHORING**

TEXT

GRID **CANCEL**

simple file save dialog.

menu:

FILE:

- SAVE CONVERSATION
- LEAVE SESSION/ROOM
- QUIT

USERS:

- INVITE USER
- USERS IN ROOM

RESULTS: SEARCH FOR USER

2 USERS FOUND

- ▼ AVAILABLE (1 user)
 - JASON SMITH
- ▼ BUSY CHATTING
- ▼ AWAY
- ▼ NOT CONNECTED (1 user)
 - JOHN SMITH

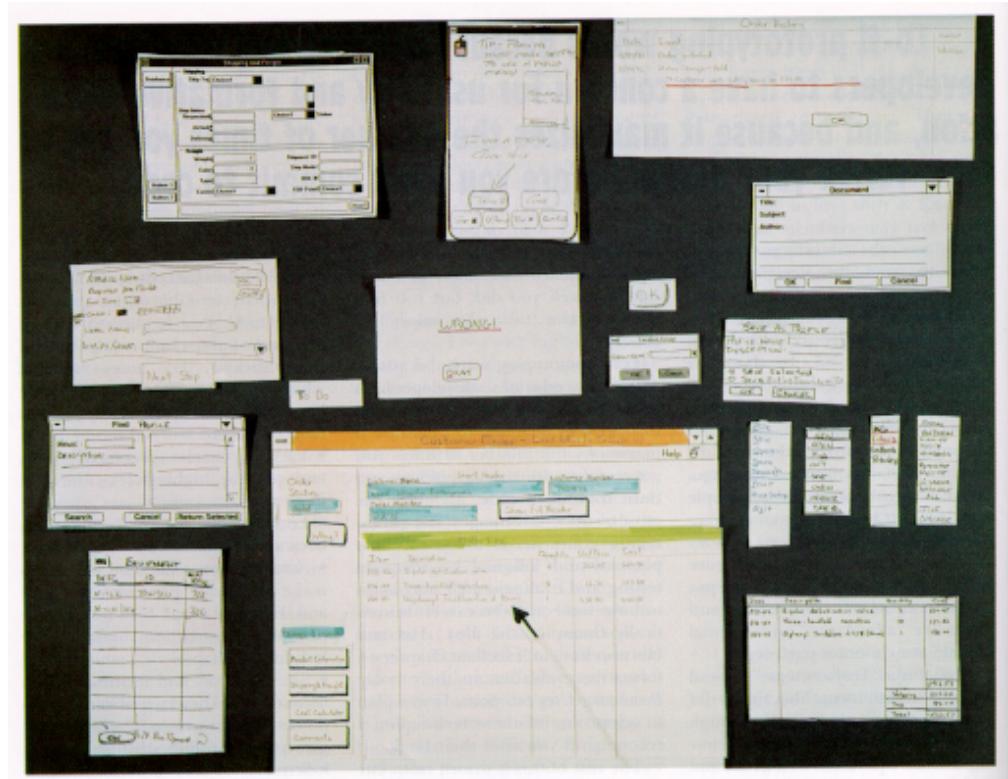
ADD TO CONTACTS **MSG** **CHAT** **CANCEL**

RESULTS: AVAILABLE USERS

15 AVAILABLE USERS

Basic low-fidelity materials

- Large, heavy, white paper (11 x 17)
- 5x8 in. index cards
- Tape, stick glue, correction tape
- Pens & markers (many colors/sizes)
- Overhead transparencies
- Scissors, knives, ...



Low-fidelity examples

Housing search

Search

Search

Address	Type in here
City	Type in here
Zip Code	Spin
Maximum Bid	Type in here

Features

Beds	<input type="checkbox"/>
Baths	<input type="checkbox"/>
Parking Spaces	<input type="checkbox"/>
Balcony	<input type="checkbox"/> yes <input type="checkbox"/> no
Backyard	<input type="checkbox"/> yes <input type="checkbox"/> no
Pets	<input type="checkbox"/> yes <input type="checkbox"/> no

Garbage	<input type="checkbox"/> yes <input type="checkbox"/> no
Water	<input type="checkbox"/> yes <input type="checkbox"/> no
Cable	<input type="checkbox"/> yes <input type="checkbox"/> no
Electric	<input type="checkbox"/> yes <input type="checkbox"/> no
Spa	<input type="checkbox"/> yes <input type="checkbox"/> no
Pool	<input type="checkbox"/> yes <input type="checkbox"/> no
Security Guard	<input type="checkbox"/> yes <input type="checkbox"/> no
Furnished	<input type="checkbox"/> yes <input type="checkbox"/> no

Listing

Listing



123 Beverly Way Los Angeles Ca 91706

Type in here

Features

Beds	3	Garbage	Y
Baths	2	Water	N
Parking Spaces	1	Cable	N
Balcony	Y	Electric	Y
Backyard	N	Spa	Y
Pets	Y	Pool	Y
		Security Guard	Y
		Furnished	N

Search Results

- Search Results
- | | | |
|---|--|------------------------------------|
|  | 1. 123 Beverly Way, Los Angeles Ca 91706 | <input type="button" value="Bid"/> |
|  | 2. 666 Burning Lane, Greenwich CT 06830 | <input type="button" value="Bid"/> |
|  | 3. 1899 Angel Ave, Bayport CA 97021 | <input type="button" value="Bid"/> |
|  | 4. 777 Luckey Ave, Carroll MD 20832 | <input type="button" value="Bid"/> |
|  | 5. Apt 20 50 Bell Air, Carroll MD 20832 | <input type="button" value="Bid"/> |
|  | 6. Apt 2115 1000 Melrose, Calistoga CA 94521 | <input type="button" value="Bid"/> |
|  | 7. Apt 15 2315 Channing, Capon NJ 9172 | <input type="button" value="Bid"/> |
- More Results: Pg 1 Pg 2 Pg 3

Search Results

Search

- Housing
- Roommate(s)

Listing



123 Beverly Way
Los Angeles Ca 91706

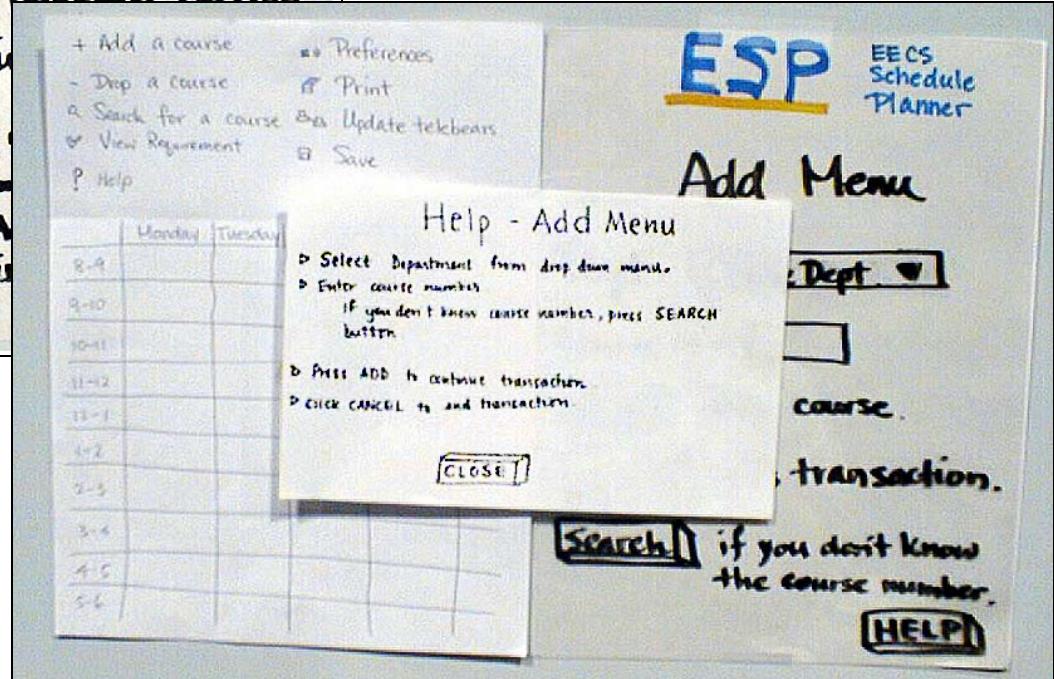
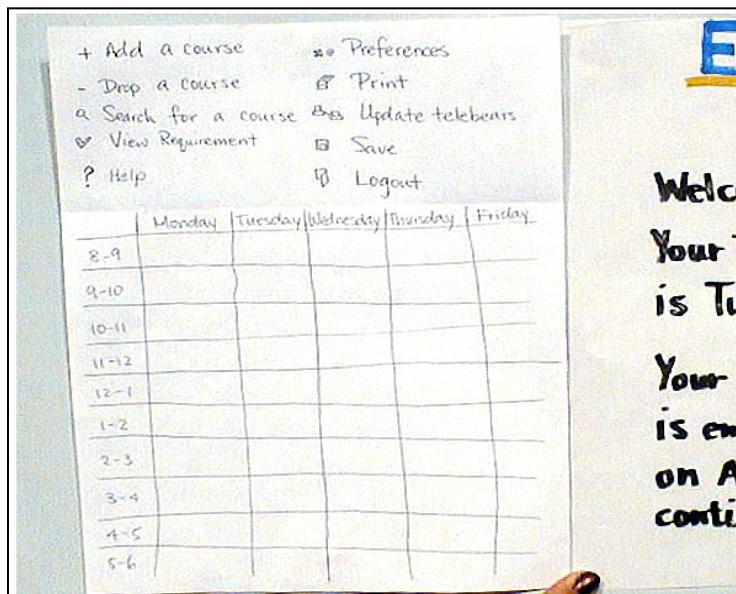
Features

Beds	3	Garbage	*yes *no
Baths	2	Water	*yes *no
Parking Spaces	1	Cable	*yes *no
Balcony	*yes *no	Electric	*yes *no
Backyard	*yes *no	Spa	*yes *no
Pets	*yes *no	Pool	*yes *no
		Security Guard	*yes *no
		Furnished	*yes *no

Listing

Low-fidelity examples

■ Course registration



Testing with a low-fi prototype

- Construct your model
 - Prepare the actual test
 - Conduct the test
 - Evaluate results
 - ... and typically, iterate
-
- The details...

Constructing the model

- Set a deadline
 - don't think too long - build it!
- Draw a window frame on large paper
- Put different screen regions on cards
 - anything that moves, changes, appears/disappears
- Ready response for any customer action
 - e.g., have those pull-down menus already made
- Use photocopier to make many versions

Preparing for a test

- Select your customers
 - understand background of intended customers
 - use a questionnaire to get the people you need
 - don't use friends or family
 - current customers OK? (maybe)
- Prepare scenarios that are...
 - typical of the product during actual use
 - make prototype support these (small, yet broad)
- Practice to avoid “bugs”

Conducting a test

- Four testers (minimum)
 - greeter - puts participants at ease & gets data
 - facilitator - only team member who speaks
 - gives instructions & encourages thoughts, opinions
 - computer - knows application logic & controls it
 - always simulates the response, w/o explanation
 - observers - take notes & recommendations
- Typical session is 1 hour
 - preparation, the test, debriefing

Conducting a test

- Greet
 - get forms filled, assure confidentiality, etc.
- Test
 - facilitator hands written tasks to participant
 - must be clear & detailed
 - facilitator keeps getting “output” from participant
 - “What are you thinking right now?”, “Think aloud”
 - observe -> “a-ha”, laugh, gape, etc.

Conducting a test

- Debrief
 - fill out post-evaluation questionnaire
 - ask questions about parts you saw problems on
 - gather impressions
 - thank them

Evaluating results

- Sort & prioritize observations
 - what was important?
 - lots of problems in the same area?
- Create a written report on findings
 - gives agenda for meeting on design changes
- Make changes & iterate

"Wizard of Oz" technique

- Faking the interaction
 - name from (surprise) “The Wizard of Oz”
 - “the man behind the curtain”
- Much more important for hard-to-implement features
 - e.g., speech & handwriting recognition
- Example: How might you evaluate a natural language flight-reservation system using a Wizard-of-Oz technique?
 - materials/setup? input/output?

Advantages of low-fi prototyping

- Takes only a few hours
 - no expensive equipment needed
- Can test multiple alternatives
 - fast iterations
 - number of iterations is tied to final quality
- A lot of interaction can be faked without a computer

Issues with low-fi prototyping (?)

- Slow compared to real computer
- End users can't do it themselves
- Sometimes hard for participants to recognize widgets
- Hard to implement certain interactive functionality, like pulldowns, or highlighting
- Dynamic elements hard to simulate
 - changing widgets: pop-ups, etc.
 - changing elements: animations, movies, etc.
 - interactive controls: dragging, scrolling, etc.

Issues with low-fi prototyping (?)

- Doesn't map well to what will actually fit on the screen
- Timing in real-time hard to do (slooooow computer)
- Couldn't hold in your hand -- different ergonomics from target device
- Writing on paper not the same as writing on target device
- Won't look like the final product
- Not in context of user's work environment

Issues with low-fi prototyping (?)

- Can't measure realistic I/O
 - mouse (can't sketch the same way)
 - slow response
- "Computer" has to keep track of a lot of paper
- Hard to draw well (recognition of elements)
- Users wouldn't criticize UI
- Couldn't give proper affordance that something wasn't selectable
- Can't get accurate time measurement

Why create high-fidelity prototypes?

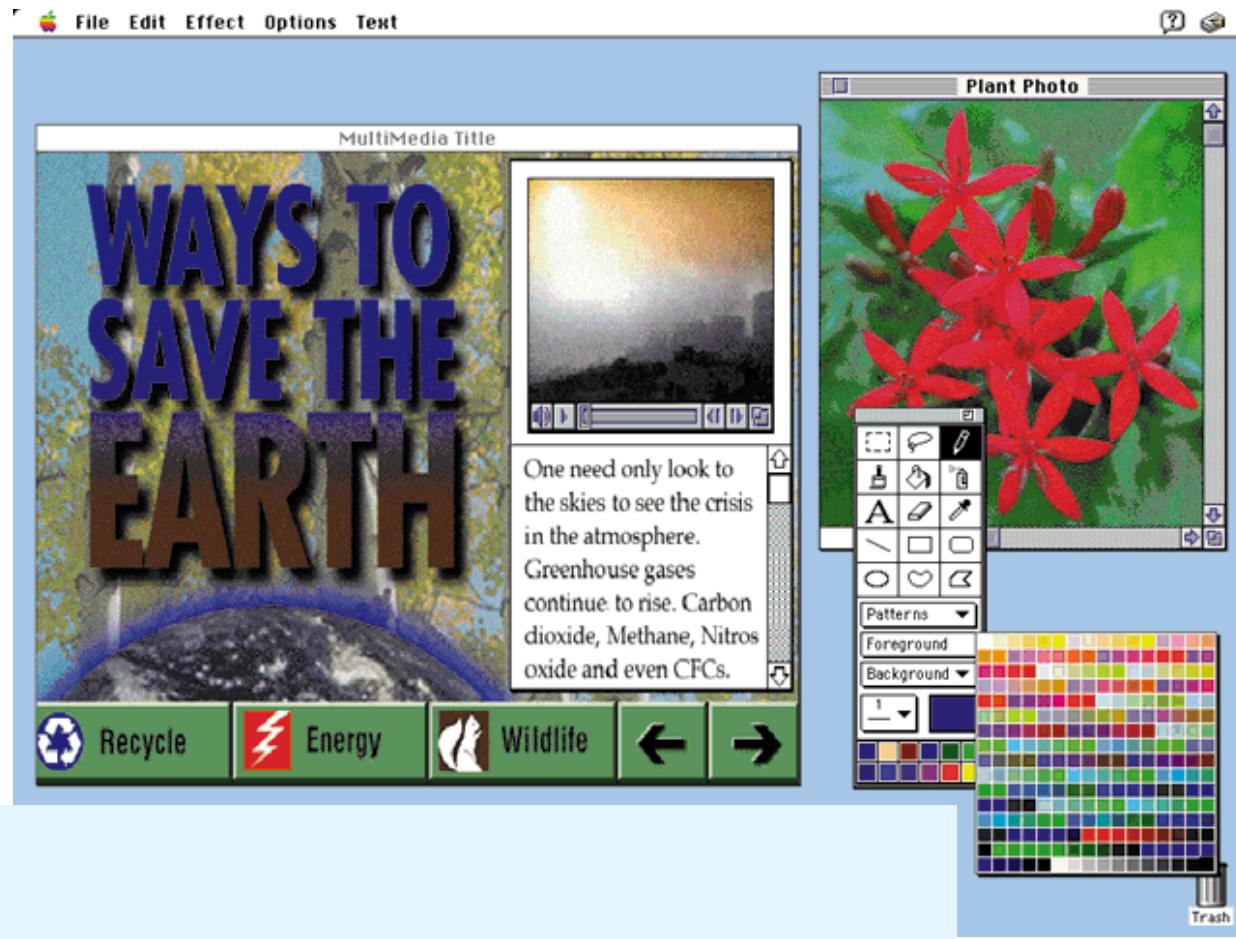
- Must test & observe ideas with customers
- Paper mock-ups don't go far enough
 - how would you show a drag operation?
 - not realistic of how interface will be used
- Building final app now might be a mistake...
 - changes in the UI can cause huge code changes
 - need to convince programmer & hope they get it right
 - takes too much time
- ... but a drag & drop prototyping tool might be just the thing

History of Hi-Fi Prototyping

- HyperCard
 - for Macintosh – built by Bill Atkinson
 - metaphor: card transitions on button clicks
 - comes with widget set
 - drawing & animation limited
- Adobe Director
 - still commonly used by designers
 - intended for multimedia — for a long time, it lacked interface widgets or controls
 - good for non-widget UIs or the “insides” of app
- Both have scripting languages

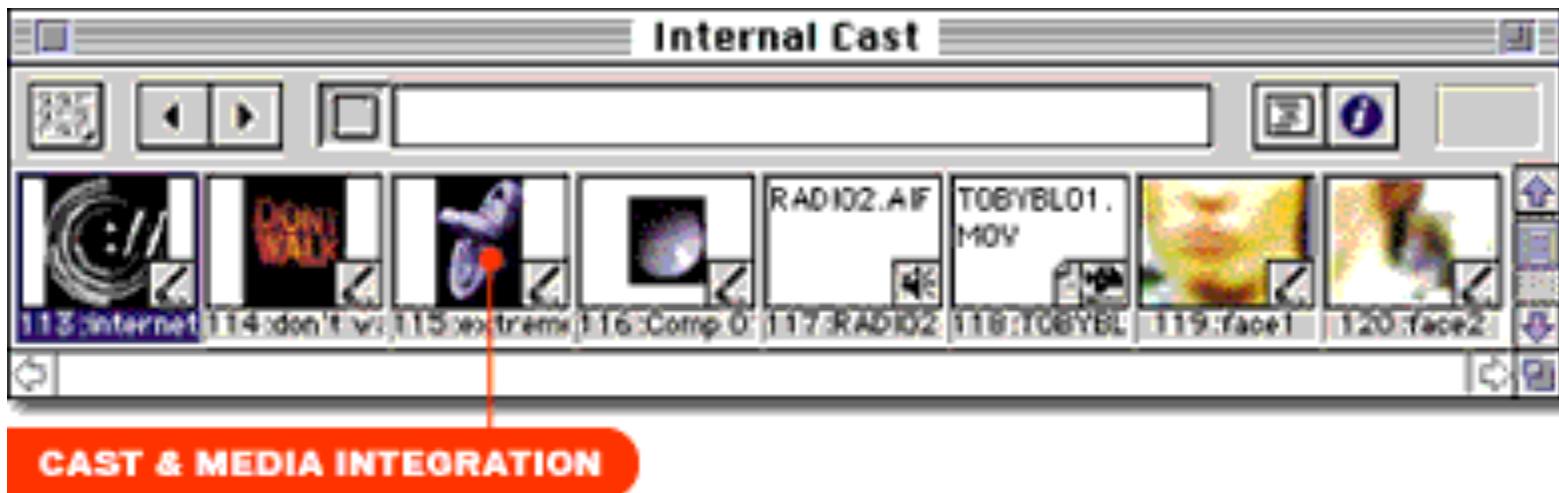
HyperCard

- Tool palettes



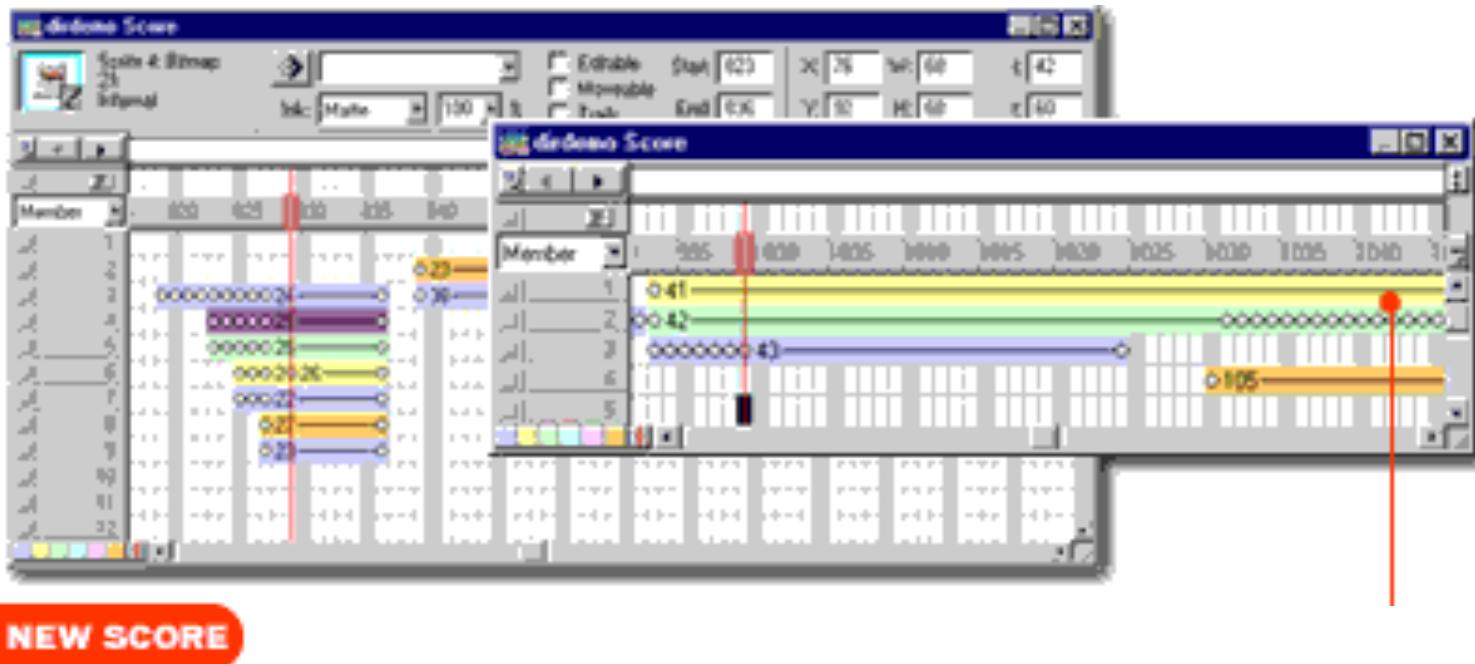
Director Cast

- Basic objects used in interface
- Mainly multimedia in nature
 - images, audio, video, etc.
 - synchronize with cue points



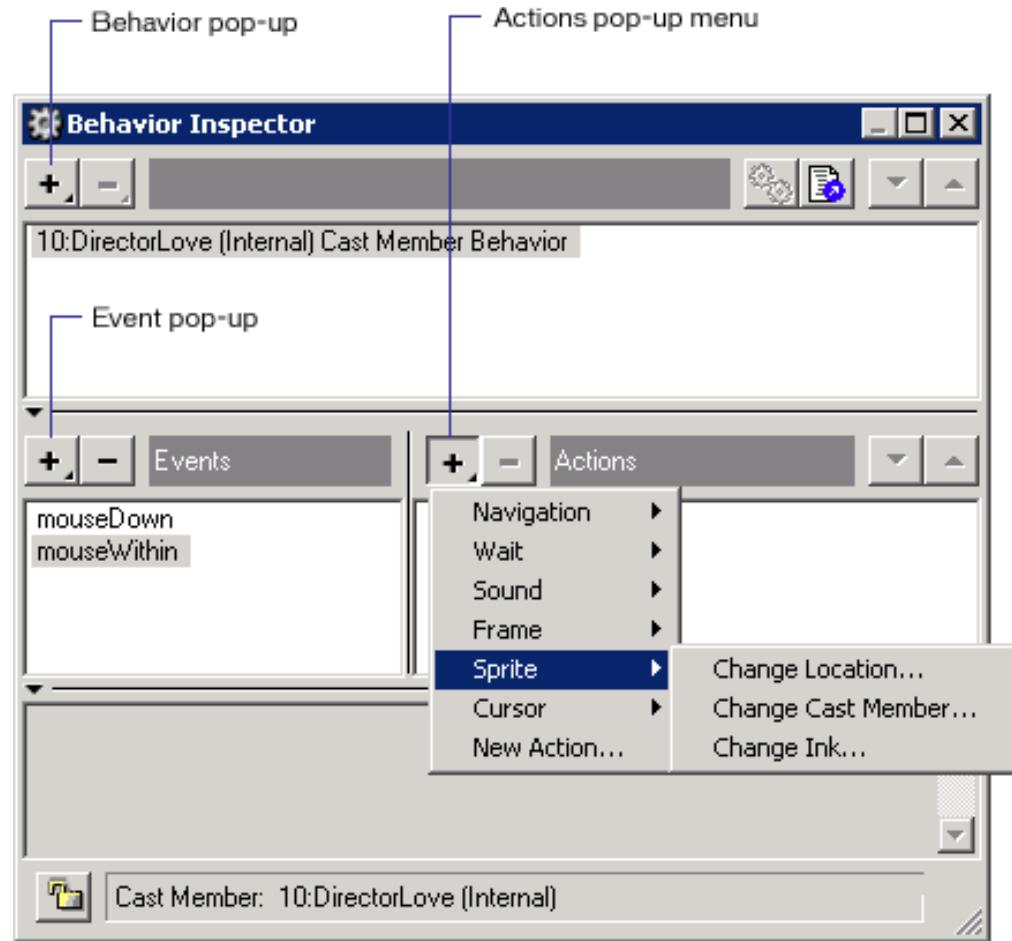
Director Score

- Overview of events in time



Director Behavior Inspector

- Connect events to actions, drag & drop

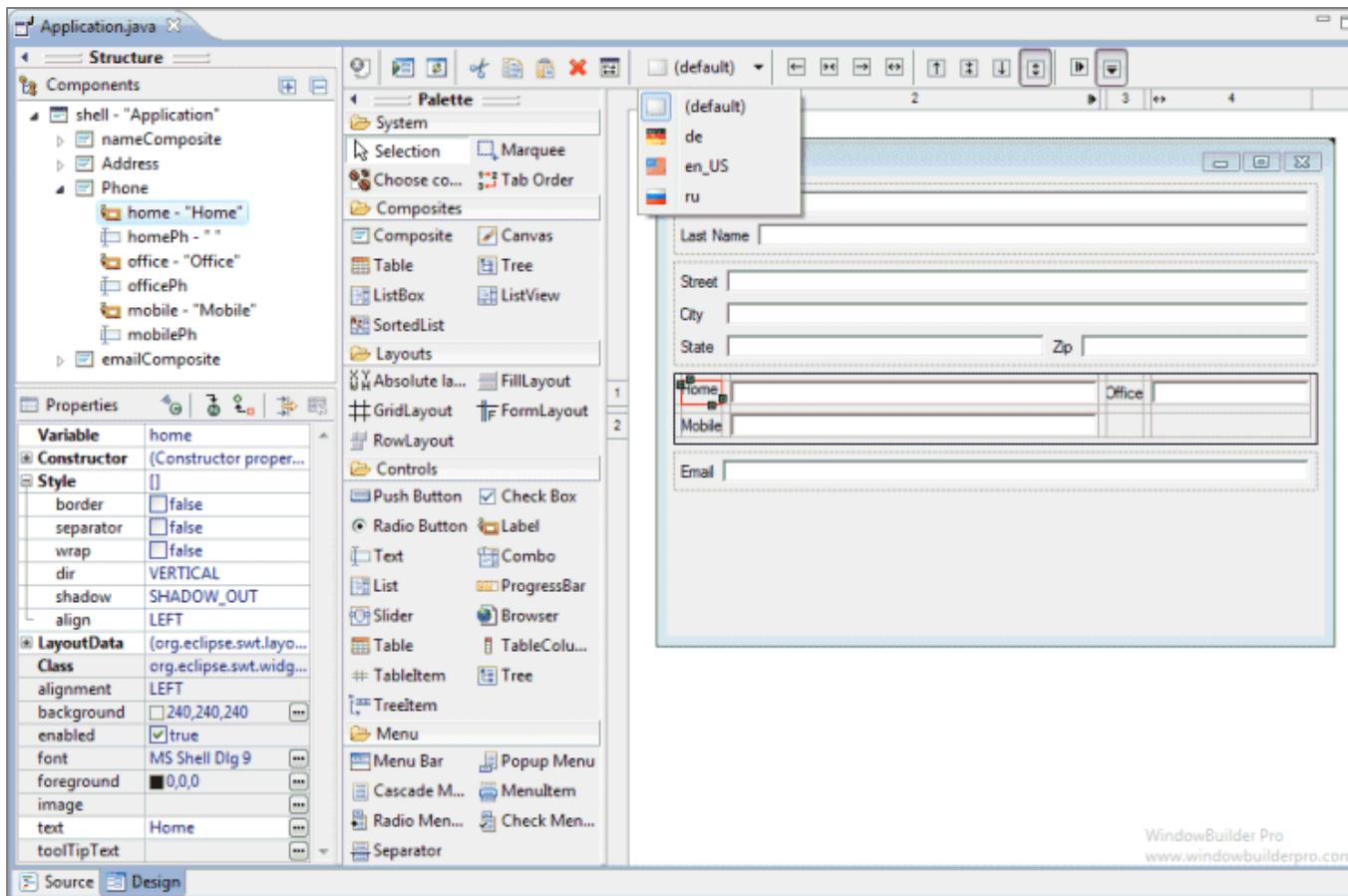


Older Interface Builders

- Visual Basic
 - lots of widgets (AKA controls)
 - simple language
 - slower than other UI builders
 - out of use now for ~10 years
- NeXT UIB, SpecTCL, PowerBuilder...
 - widget sets
 - easily connect to code via callbacks, etc.
 - some of the first to use "commercial-strength" languages

Today's Interface Builders

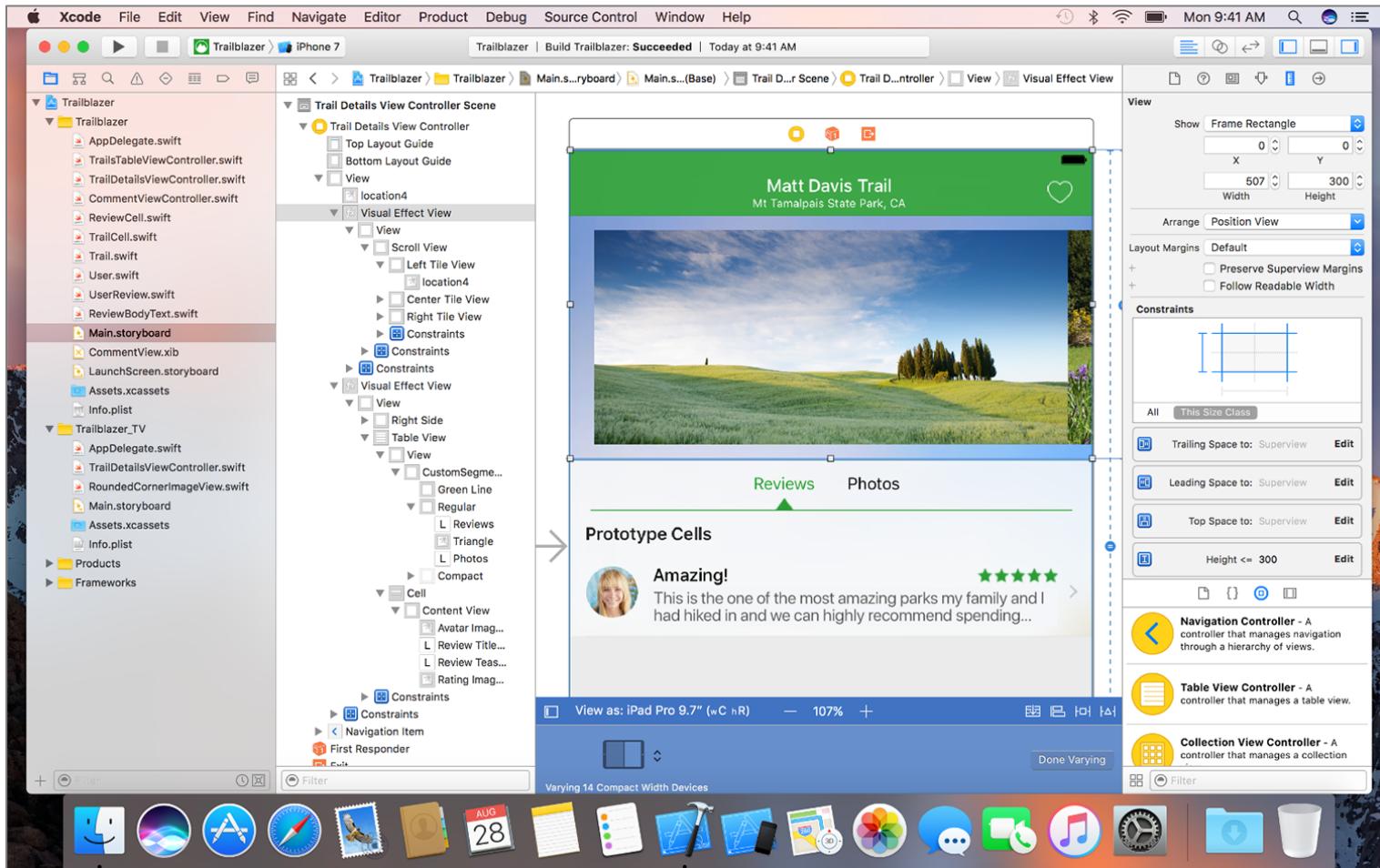
- WindowBuilder for Java Swing



WindowBuilder Pro
www.windowbuilderpro.com

Today's Interface Builders

- Interface Builder & Xcode for Apple's Swift



Today's Interface Builders

- Similar systems also exist for web apps
- For example, ones targeted for Bootstrap, e.g.:

<https://www.layoutit.com/build>

Hi-fi prototyping tools & UI builders

- Performance
 - prototyping tools produce slow programs
 - UI builders depend on underlying language
- Widgets
 - prototyping tools may not have complete set
 - UI builders have widget set common to platform
- Final product
 - generally use UI builders, though occasionally see things created in a prototyping tool (multimedia)

Summary: Low-fi prototyping

- Advantages
 - quick & cheap
 - support brainstorming
 - designers feel comfortable sketching
 - facilitates fast iterations
 - interaction can be faked
- Drawbacks
 - do not evolve easily
 - lack support for “design memory”
 - force manual translation to electronic format
 - do not allow end-user interaction

Summary: Hi-fi prototyping

- Advantages
 - don't need to produce layouts from scratch
 - initializes all the basic component code,
you fill in the functionality later
- Disadvantages
 - must give specific instance of a general idea
 - e.g., exact widgets, fonts, alignments, colors
 - takes longer!
 - designers, evaluators focus on details
instead of overall structure & functionality

Some other prototyping tools

- Balsamiq: <https://balsamiq.com/>
- InVision: <http://www.invisionapp.com/>
- Proto.io: <http://proto.io>
- Fluid UI: <http://www.fluidui.com>
- Adobe Muse: <http://muse.adobe.com/>