#### CS3205 – HCI IN SOFTWARE DEVELOPMENT

#### **MORE PROTOTYPING STRATEGIES**

**Tom Horton** 

\* Material from: Floryan (UVa) Klemmer (UCSD, was at Stanford)

# WHAT ARE WE DOING TODAY?

#### • High-fidelity, Mock-ups

#### • Wizard of Oz Prototyping

## WHAT IS A HIGH-FIDELITY PROTOTYPE?

• "A prototype that mimics the design very closely, but remains functionally incomplete."

• This might be actual code, or actual physical devices that don't work fully.

• For things on screens, how might this be created?

- Using the final, target implementation language
  - Qt, Java Swing, Tkinter, HTML, Android,...
  - How's that different than version 1?
- Using a tool to create a wire-frame or mock-up
  How's that different than low-fidelity prototyping?

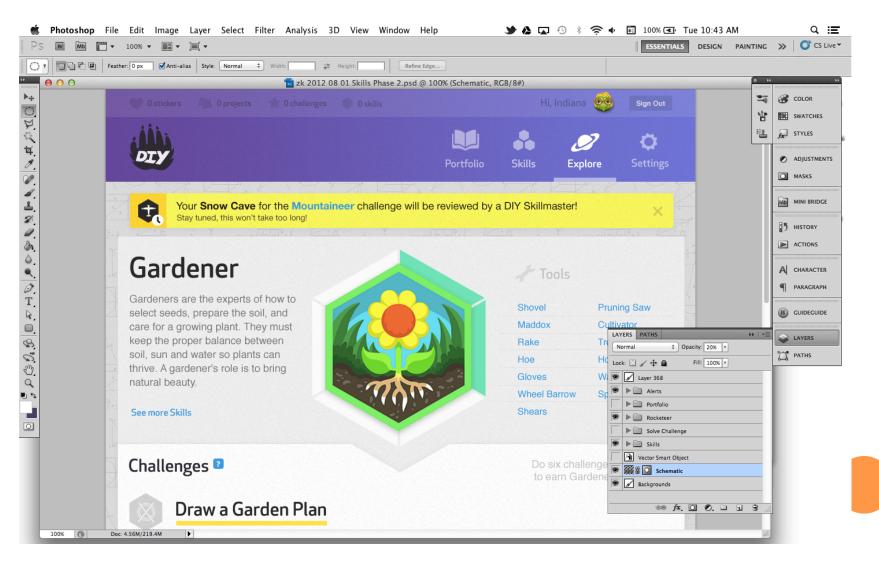
# WIRE-FRAMES, DIGITAL MOCKUPS

Dashboard $ ightarrow$ Demonstration Space $ ightarrow$ Home $ ightarrow$ My Great Feature Spec		Browse -	Admin -	Search
			<u> </u>	🕂 Add 🗸 🔅 Tools 🗸
2 Added by <u>Admin</u> , last edited by <u>Admin</u> on Oct 11, 2010 (view change) show comment				
This is a feature specification for my great new feature. I know you won't read this text, so I designed the key screens for you, enjoy!				
	A Web Page		$\bigcirc$	
	Features			
	<u>Home</u> > <u>Products</u> > <u>Xyz</u> > Features			
	lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt			
			~	
	edit this mockup			
This is the main feature page.				

Add Labels

Add Comment

#### DIGITAL MOCKUPS



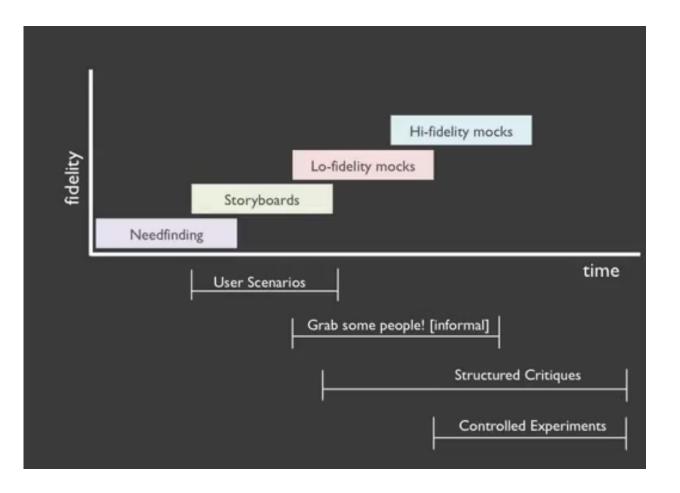
#### **OBSERVATIONS ABOUT EXAMPLES**

- Both clearly are not "functional"
- Balsamiq example:
  - Looks low-fi. Does it really mimic the final design closely? What's missing?
- Photoshop
  - What's there?
    - Layout, organizations, size of things.
    - Color, fonts
- Which took more time to create?

## TOOLS

- Free software exists for doing this!
  - <u>http://mashable.com/2012/06/07/mockup-tools/</u>
- Balsamiq Mockups (desktop application)
- Mockingbird (online in-browser mockups)
- Mockup Builder
- POP (Prototyping on paper) app
- Can also use tools like Powerpoint or Photoshop to simulate a design relatively effectively.

#### PROTOTYPING OVER TIME



# WHEN DOES IT BECOME HI-FIDELITY?

• When you:

- take the time to put in more and more detail,...
- that's closer to the final design, ...
- that includes "fit and finish"
- When the goal is
  - More about refinement about a designed you've chosen,
  - And less about exploring alternatives.
- You might be using the same tool to create the representation, so that's not necessarily the difference.

### WIZARD OF OZ PROTOTYPES

# MOTIVATION: PROTOTYPING PROBLEM

• Need feedback from real people in order to improve your design. However,

• Can't get feedback unless you have something built and working!

• What if your system is very complicated?

#### MOTIVATION: WHAT IF WE COULD...

• Make an *interactive* application without (much) code, but still:

• Get feedback from real users.

# WIZARD OF OZ PROTOTYPING!



#### WIZARD OF OZ PROTOTYPING

- Human operator simulates the functionality of the system behind the scenes.
- Don't need to build the whole system! Just have your "wizard" simulate it for the user.



# WIZARD OF OZ TECHNIQUE

- Make an interactive application without much code.
  - Front end interface is coded.
  - A remote wizard controls the interface's characteristics based on user input directly.
  - Makes sense to do this if it's faster/cheaper/easier than making a real thing.
- Get feedback from real users.
  - Users will think the system is more real if done well!

# ANOTHER EXAMPLE: SPEECH RECOGNITION

#### • There was a time when:

- Speech recognition software was NOT good at all (still not great).
- We knew it would get better.
- We wanted to know what user's interactions with this technology should be like.
- How to study this without building a speech recognition system ourselves?



# ANOTHER EXAMPLE: SPEECH RECOGNITION

• Wizard of Oz the interaction!

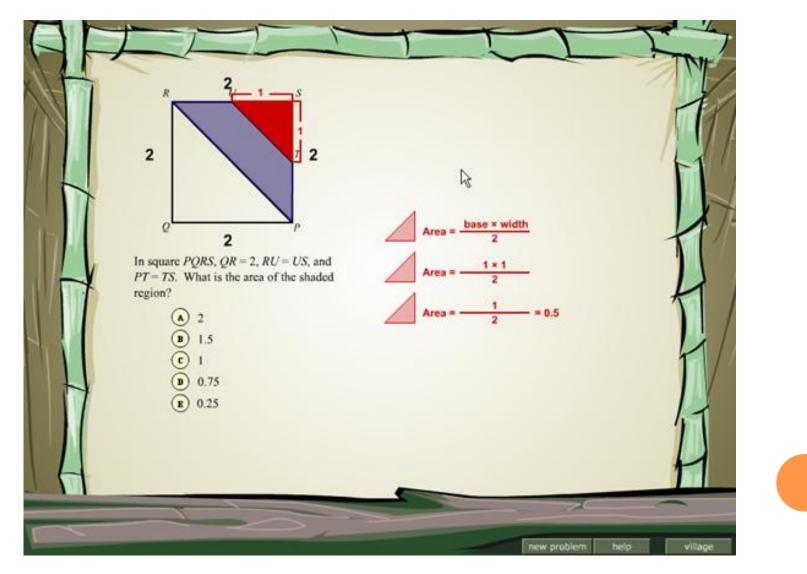
- Human listens to the voice commands of the user.
- Adjusts the interface accordingly.



# FUNNY EXAMPLE

o https://www.youtube.com/watch?v=AlyvYLLtQOg

# ANOTHER EXAMPLE: INTELLIGENT TUTORS!



## WHEN TO USE WIZARD OF OZ?

#### • Useful when:

- There is an advanced technology in your system that you don't have time to build / incorporate into prototype.
  - Speech recognition, artificial intelligence, etc.
- You haven't determined how best to implement a feature (e.g., personalized feedback) and want to test it first.

## WIZARD OF OZ CAN BE LOW-FIDELITY!

- In low-fidelity evaluations, "playing the computer" is a form of WOO
  - You are essentially the "Wizard" controlling interactions with a paper prototype while users use it.
- However, remember that the more 'real' the wizard's mirage is, the more realistic the user's reactions will be.

# MAKING A WOO PROTOTYPE

- 1. Map out scenarios and application flow.
  - Enumerate ALL scenarios if possible.
  - If not, provide guidelines in as specific a format as possible.
- 2. Put together interface "skeletons"
- 3. Develop "hooks" for wizard input.
  - If paper, hooks will be very manual.
  - If done in software, some code will need to be developed.

# MAKING A WOO PROTOTYPE

#### • 4. Put it all together.

- Where and how will the wizard provide input?
- How will the wizard receive input from the user?
- \* Remember that you'll need to build actual software for wizard's role eventually, so it must be possible!
- 5. Rehearse wizard role with a colleague.
  - Being the wizard is surprisingly difficult.
  - Work out easy bugs in interaction before using real users.

# COLLECTING DATA FOR WOO PROTOTYPES

- Practice with friends first.
- Once comfortable, recruit "users"
- Two roles: facilitator and wizard
  - Facilitator: Provides tasks (paper) and takes notes.
  - Wizard: Operates the interface.

# COLLECTING DATA FOR WOO PROTOTYPES

- User feedback can be:
  - Think aloud (speak freely as performing tasks)
  - Retrospective (discuss task afterwards)
  - Heuristic Evaluation (experts are watching interaction unfold)
- Interaction is normally video taped so designers can review later.
- Always debrief users, reveal the wizard if necessary / appropriate.

## WIZARDS THROUGHOUT DEVELOPMENT

• Functionality vs. Time graph

#### ADVANTAGES OF WIZARDS

- Faster to make / cheaper, thus more iterative prototypes possible.
- Creating multiple variations is very easy (no code to rewrite).
- More "real" than pure paper prototyping or mockups.
- Identifies bugs and problems with current design.
- Places user at center of development.
- Can envision challenging to build application.
- Designers learn by playing wizard.

#### DISADVANTAGES OF WIZARDS

- Can you really "hide behind the curtain" in a non-distracting way?
- Simulations may represent otherwise imperfect (or impossible) tech.
- Wizards require training and can be inconsistent.
- $\circ$  Playing the wizard can be exhausting  $\otimes$
- Some features are difficult (or impossible) to simulate perfectly.
- May be inappropriate in some venues.



## PROTOTYPES: QUANTITY VS. QUALITY

• Is it better to produce a large quantity of designs, or to focus on creating the best one design?

# QUANTITY VS. QUALITY

• Bayles and Orland put this to the test.





#### QUANTITY VS. QUALITY

• Well, come grading time and a curious fact emerged: the works of highest quality were all produced by the group being graded for quantity. It seems that while the "quantity" group was busily churning out piles of work - and learning from their mistakes -- the "quality" group had sat theorizing about perfection, and in the end had little more to show for their efforts than grandiose theories and a pile of dead clay. [Bayles, Orland]

# FUNCTIONAL FIXATION

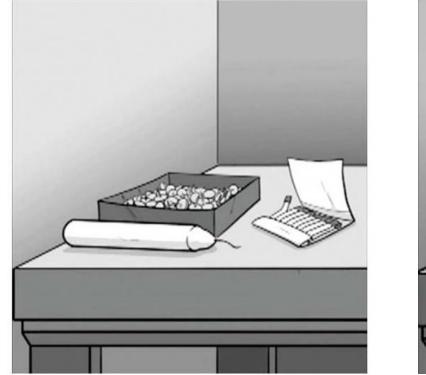
- You make a set of decisions that produce one prototype (or anything)
- You get feedback or (re)evaluate it
- But the improvement you make to this are minimal
  - You stick with the same essential design approach
  - You are reluctant to break the mold and try something very different
- You get an idea and stick with it! (It's human nature! Search for psychological concept of "functional fixedness.")
  - <u>https://en.wikipedia.org/wiki/Functional\_fixedness</u>
  - <u>https://www.mendix.com/blog/functional-fixation-and-the-power-of-parallel-prototyping/</u>

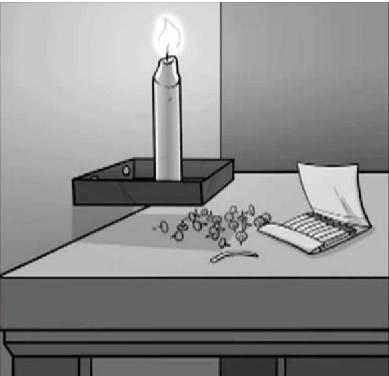
### FUNCTIONAL FIXATION



#### [Duncker, 1945]

#### FUNCTIONAL FIXATION





#### [Duncker, 1945]

# BETTER APPROACH?

#### • Parallel Prototyping:

- Making multiple prototypes in parallel
- Studies show that this leads to better designs!
  - i.e., designs led to statistically higher values in quantifiable variables of interest (more on this later).
  - \*Klemmer, Gentner, Loewenstein, Thomson, etc.

#### • Seperates Ego from Artifact

- i.e., a criticism of one design is NOT a criticism towards the designer.
- Supports TRANSFER of positive attributes across designs.

# OK FINE...SO HOW DO WE COMPARE PROTOTYPES?

- We perform an evaluation!
- An *evaluation* is an experiment (or set of experiments) meant to provide answers to at least one design question.

• The next topic!

o <u>https://www.youtube.com/watch?v=kCSzjExvbTQ</u>

# QUESTIONS?