Tape your P2 milestones to the wall
Imbalance in Interaction Design

MICHAEL BERNSTEIN
CS 247

THANKS TO SCOTT KLEMMER AND MANEESH AGRAWALA
Balance is good; uniformity is not

- Balance: all design elements average each other out to a sustainable center of mass
  - In visual attention
  - In task importance
  - In features

- Uniformity: even and symmetric placement of importance and attention
High-level info scent, location, neg. space
Medium-level grids, responsive
Low-level colors, typography, components
80/20 rule

We are the 80 percent.
80/20: What’s important?

- The 80/20 rule: 80% of the effects of a system are caused by 20% of the variables in that system.

- In other words: 80% of peoples’ use will be focused on 20% of the features.
  - Be clear with yourself: what is in the 20%?
  - How do you amplify those elements and de-emphasize the others?
Too uniform in emphasis
More 80/20-aligned
Another perspective on 80/20

- Make that central 20% functionality easy, then make the other 80% possible
Main goal: front and center

Do: one clear main action

Google Material Design spec

Don’t: multiple main actions
Violations of 80/20

• **Edge casers:** every team has one member like this
  • “But what if the user wants to make the text small caps efficiently? It’s so inefficient.”
  • Be open to a discussion of how common the case is across all your users.

• Power users often live in the 80%. Unless you’re making a power user tool, stick to accelerators, customizations, and advanced menus to support them.
Activity: What’s the 20%?

This is what’s in the home ribbon tab for Microsoft Word. (Don’t look!)
Design the layout. What would you put where? With what emphasis? What would you remove?

- Cut
- Copy
- Paste
- Format paint
- Font
- Point size
- Bigger font
- Smaller font
- Upper/lowercase
- Format erase
- Bold
- Italic
- Underline
- Strikethrough
- Subscript
- Superscript
- Text effects
- Text background color
- Text color
- Bullets
- Numbers
- Multilevel list
- Left indent margin
- Right indent margin
- Sort A–Z
- Show paragraph marks
- Left align
- Center align
- Right align
- Full justify
- Paragraph spacing
- Shading
- Border
Energy and tension
Uniform weights: little tension

• Some things must be small, so that others may be large

Example from Sean Adams, Stanford Lynda
Simple grids: little tension

- Tension requires elements that are not entirely predictable or repeated.

- However, they must still balance.
Play with the grid

Inspired by Sean Adams, Stanford Lynda
Play with the grid

Inspired by Sean Adams, Stanford Lynda
Balance

- Create tension by breaking the grid, which puts extra weight in some parts of the design.

- Counter that tension to maintain overall balance in the design.
De novo vs. patterns

“I’ll just write it myse—”
“NO STAHP”
What you already know

- Utilize reusable design patterns, (almost) never reinvent them

Autocomplete

ui-patterns.com
Resist the temptation

- It may seem like a good idea to develop your own icons for your design
- It’s typically not a good idea
- Good icon design is hard
Easy to find well-designed icons

<table>
<thead>
<tr>
<th>The Noun Project</th>
<th>Font Awesome</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amazon</td>
<td>🌐 amazon</td>
</tr>
<tr>
<td>Battery-1</td>
<td>🌋 battery-1 (alias)</td>
</tr>
<tr>
<td>Battery-2</td>
<td>🌋 battery-2 (alias)</td>
</tr>
<tr>
<td>Battery-empty</td>
<td>🌋 battery-empty</td>
</tr>
<tr>
<td>Battery-full</td>
<td>🌋 battery-full</td>
</tr>
<tr>
<td>Battery-three-quarters</td>
<td>🌋 battery-three-quarters</td>
</tr>
<tr>
<td>Black-tie</td>
<td>🔥 black-tie</td>
</tr>
<tr>
<td>Calendar-plus-o</td>
<td>📅 calendar-plus-o</td>
</tr>
<tr>
<td>Calendar-times-o</td>
<td>📅 calendar-times-o</td>
</tr>
<tr>
<td>Chrome</td>
<td>🌋 chrome</td>
</tr>
<tr>
<td>Contao</td>
<td>🔍 contao</td>
</tr>
<tr>
<td>Fonticons</td>
<td>🔍 fonticons</td>
</tr>
<tr>
<td>Gg-circle</td>
<td>🌋 gg-circle</td>
</tr>
</tbody>
</table>
Utilizing years of refinement

- The native HTML user interface components are bland, incomplete, and do not carry modern affordances.

![HTML User Interface Components](http://www.cs.tut.fi/~jkorpela/www/testel.html)

- Button: A cool button
- Reset button: Reset
- Single-line text input field: Default text.
- Multi-line text input field (textarea):
  Default text.

The following two radio buttons are inside a fieldset element with a legend:

**Legend**

- Radio button 1
- Radio button 2 (initially checked)

Check those that apply
UI toolkits

- Capture today’s best practice

<table>
<thead>
<tr>
<th>Name</th>
<th>Double-Line Header</th>
<th>Rating</th>
<th>Signed Up</th>
</tr>
</thead>
<tbody>
<tr>
<td>John Boo</td>
<td>ok</td>
<td>3</td>
<td>15 Sep, 8:56 AM (2013)</td>
</tr>
<tr>
<td>Michael Robinson</td>
<td>ok</td>
<td>5</td>
<td>15 Sep, 7:12 AM (2013)</td>
</tr>
<tr>
<td>Alexander Robson</td>
<td>Blocked</td>
<td></td>
<td>15 Sep, 4:34 AM (2013)</td>
</tr>
<tr>
<td>Jannifer Pinsker</td>
<td>Blocked 24h</td>
<td></td>
<td>15 Sep, 2:08 AM (2013)</td>
</tr>
<tr>
<td>Bob Robson</td>
<td>ok</td>
<td>5</td>
<td>15 Sep, 8:56 AM (2013)</td>
</tr>
<tr>
<td>Michael Robinson</td>
<td>Suspect</td>
<td></td>
<td>15 Sep, 7:12 AM (2013)</td>
</tr>
</tbody>
</table>
<button type="button" class="btn btn-default">Default</button>

<button type="button" class="btn btn-primary">Primary</button>

<button type="button" class="btn btn-success">Success</button>

<button type="button" class="btn btn-info">Info</button>

<button type="button" class="btn btn-warning">Warning</button>

<button type="button" class="btn btn-danger">Danger</button>

<button type="button" class="btn btn-link">Link</button>

<--- Standard button -->

<--- Provides extra visual weight and identifies the primary action in a set of buttons -->

<--- Indicates a successful or positive action -->