Interaction Techniques

SWE 632, Spring 2018

GEORGE MASON UNIVERSITY

Today

- How do users determine which interface elements map to which actions?
- How can we design interactions that minimize physical effort?
- What's important in designing for mobile devices?
- What's universal design, and why does it matter?

Identifying actions

goals → action sequence

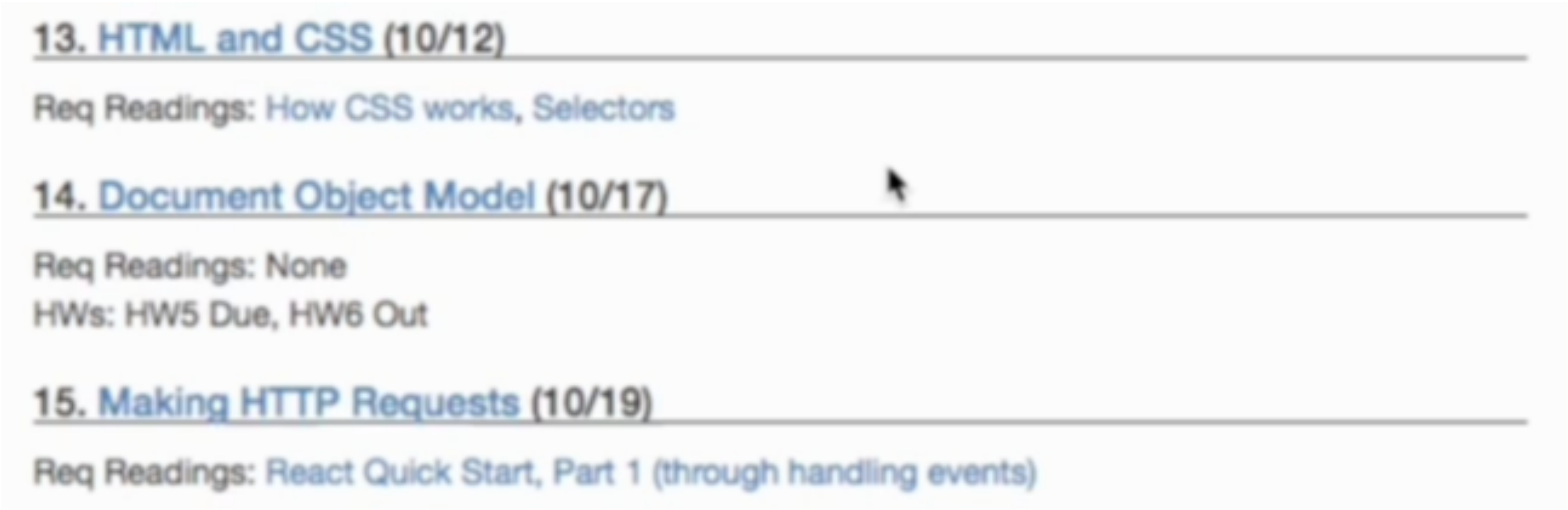
Signifiers

Is this a button? Or a link?

- Goals
  - Show which UI elements can be manipulated
  - Show how they can be manipulated
  - Help users get started
  - Guide data entry
  - Suggest default choices
  - Support error recovery

Hinting

- Indicate which UI elements can be interacted with
- Possible visual indicators
  - Static hinting - distinctive look & feel
  - Dynamic hinting - rollover highlights
  - Response hinting - change visual design with click
  - Cursor hinting - change cursor display



13. HTML and CSS (10/12)

Req Readings: How CSS works, Selectors

14. Document Object Model (10/17)

Req Readings: None
HWs: HW5 Due, HW6 Out

15. Making HTTP Requests (10/19)

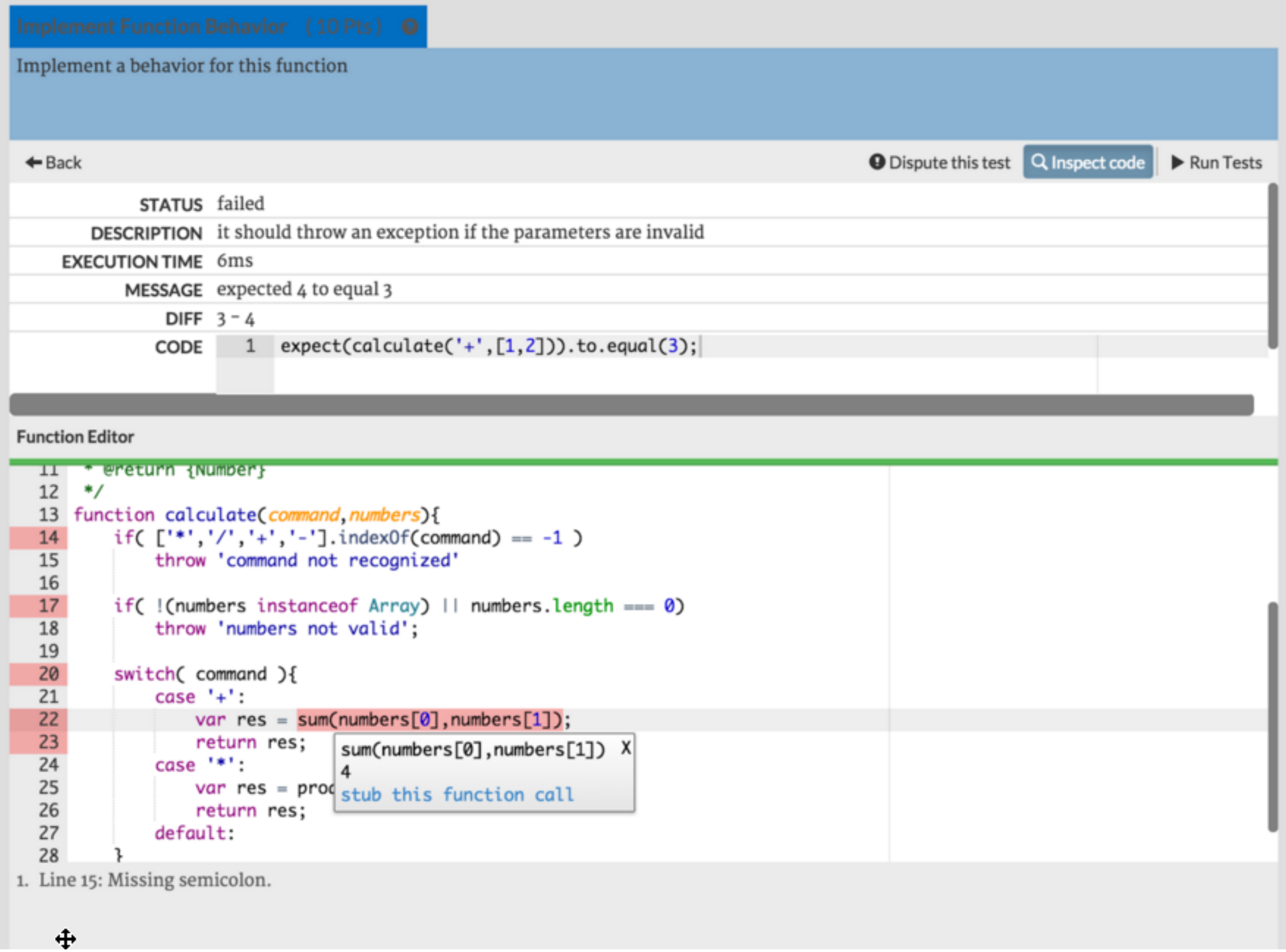
Req Readings: React Quick Start, Part 1 (through handling events)

Help users predict outcome of actions

- What does this do?
- Should I click it?

Clarity of wording (Example)

- Design for clarity & precision



Implement a behavior for this function

STATUS failed

DESCRIPTION it should throw an exception if the parameters are invalid

EXECUTION TIME 6ms

MESSAGE expected 4 to equal 3

DIFF 3 - 4

CODE

```
1   expect(calculate('+',[1,2])).to.equal(3);
```

```
11    * @return {Number}
12    */
13   function calculate(command,numbers){
14       if( ['*','/','+','-'].indexOf(command) == -1 )
15           throw 'command not recognized'
16
17       if( !(numbers instanceof Array) || numbers.length === 0)
18           throw 'numbers not valid';
19
20       switch( command ){
21           case '+':
22               var res = sum(numbers[0],numbers[1]);
23               return res;
24           case '*':
25               var res = proc
26               return res;
27           default:
28       }
```

```
sum(numbers[0],numbers[1])  X
4
stub this function call
```

1. Line 15: Missing semicolon.

Clarity of wording

- Choose words carefully
- Speak the user’s language
- Avoid vague, ambiguous terms
- Be as specific as possible
- Clearly represent domain concepts

Likely & useful defaults

- Default text, if relevant (e.g., date)
- Default cursor position
- Avoid requirements to retype & re-enter data

Modes

- Vary the effect of a command based on state of system
- Examples
  - caps lock
  - insert / overtype mode
  - vi / emacs command modes
  - keyboard entry used for controlling game and chatting

Challenges with modes

- Modes create inconsistent mapping
  - E.g., control S sometimes saves, sometimes sends email
  - Especially dangerous for frequent interactions that become highly automatic System 1 actions

- Avoid when possible
- Clearly distinguish if necessary
  - Make clear to user which mode they are in and how to change

Command interactions

- How can a user invoke a command?

- Common examples
  - Menus
  - Buttons
  - Toolbar
  - Dialog box
  - Keyboard shortcut
  - Gesture

- What are some advantages and disadvantages of each approach?

Physical actions

Avoid physical awkwardness

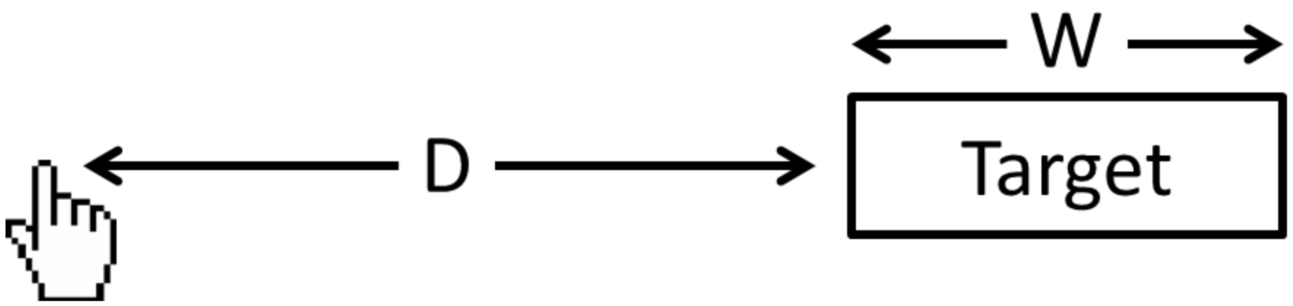
- Switching between input devices takes time
- Avoid forcing user to constantly switch between input devices (e.g., keyboard & mouse)
  - e.g., Effective tab order between fields
- Avoid awkward keyboard combinations

Moving the mouse

- After a user has (1) realized that a region is interactable, (2) decided that it will cause the desired action to be invoked
- How long does it take for a user to move the cursor to click on it?

- What factors might influence this time?

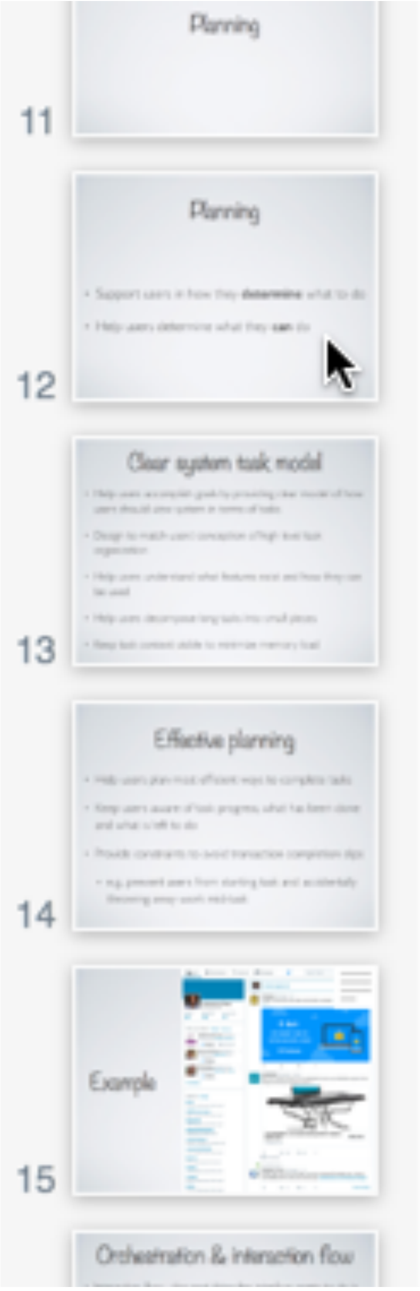
Fitt’s law



- Time required to move to a target **decreases** with target **size** & **increases** with **distance** to the target
- Movements typical consist of
  - one large quick movement to target (**ballistic** movement)
  - fine-adjustment movement (**homing** movements)
- Homing movements generally responsible for most of movement time & errors
- Applies to rapid pointing movements, not slow continuous movements

Design implications of Fitt’s law

- **Constraining** movement to one dimension dramatically increases speed of actions
  - e.g., scroll bars are 1D



11 Planning

12 Planning

- Support users in how they **determine** what to do
- Help users determine what they **can** do

13 Clear system task model

- Help users accomplish goals by providing clear model of how users should view system in terms of tasks
- Design to match users' conception of high level task organization
- Help users understand what features exist and how they can be used
- Help users decompose long tasks into small pieces
- Keep task context visible to minimize memory load

14 Effective planning

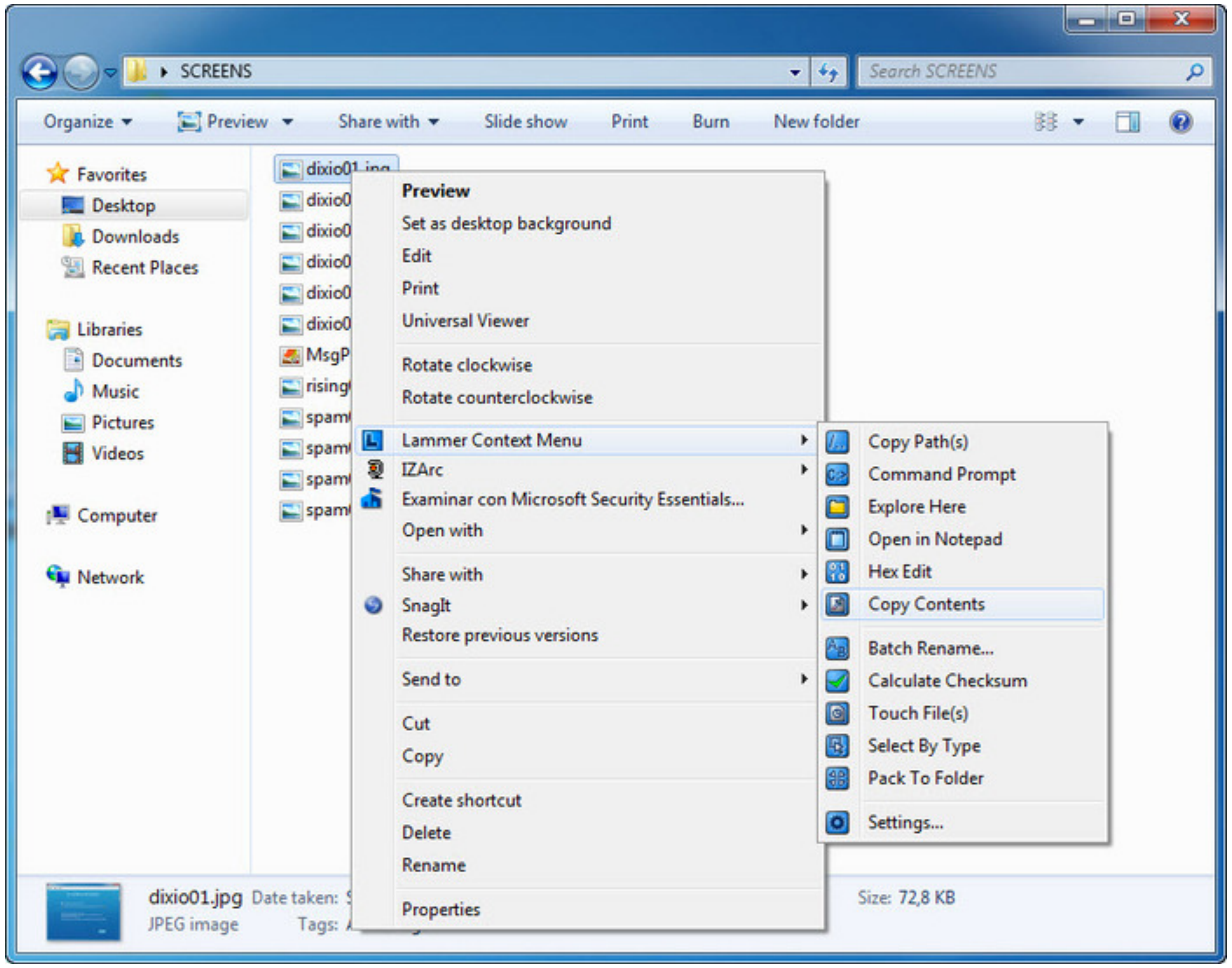
- Help users plan most efficient ways to complete tasks
- Keep users aware of task progress, what has been done and what is left to do
- Provide constraints to avoid transaction completion slips
  - e.g., prevent users from starting task and accidentally throwing away work mid-task

15 Example

Orchestration & interaction flow

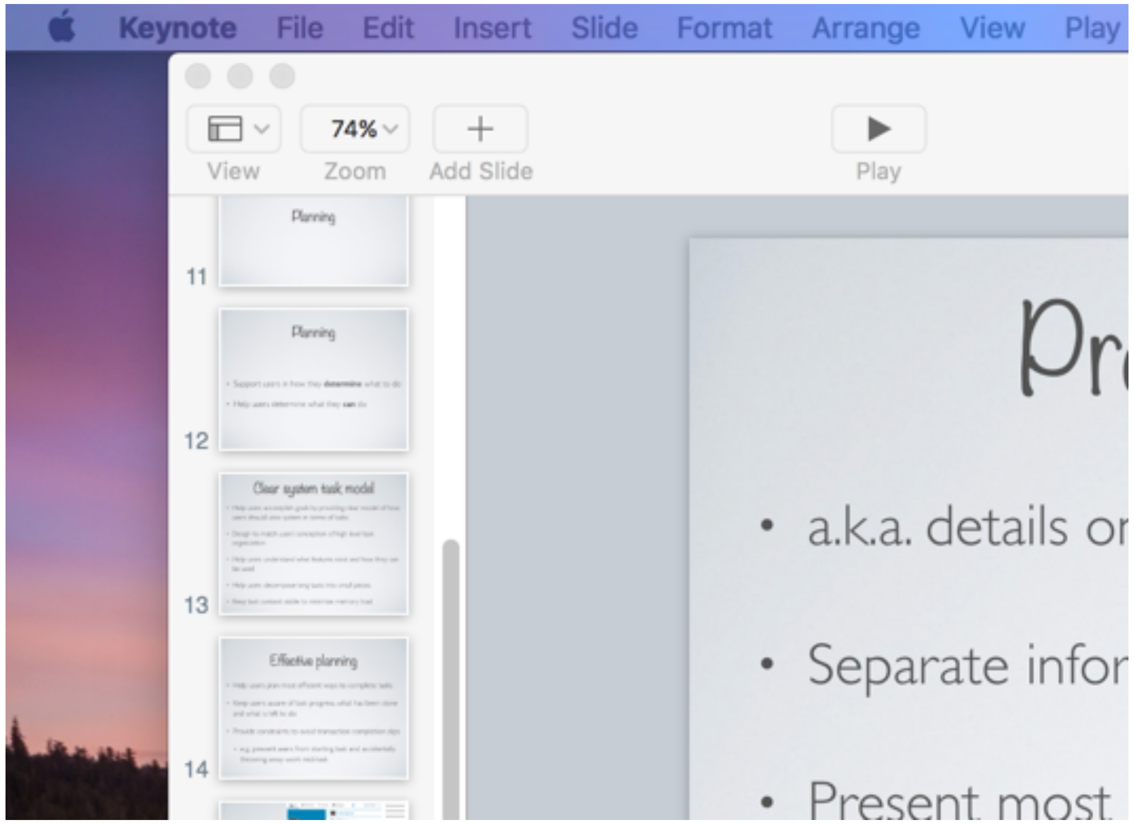
Design implications of Fitt's law

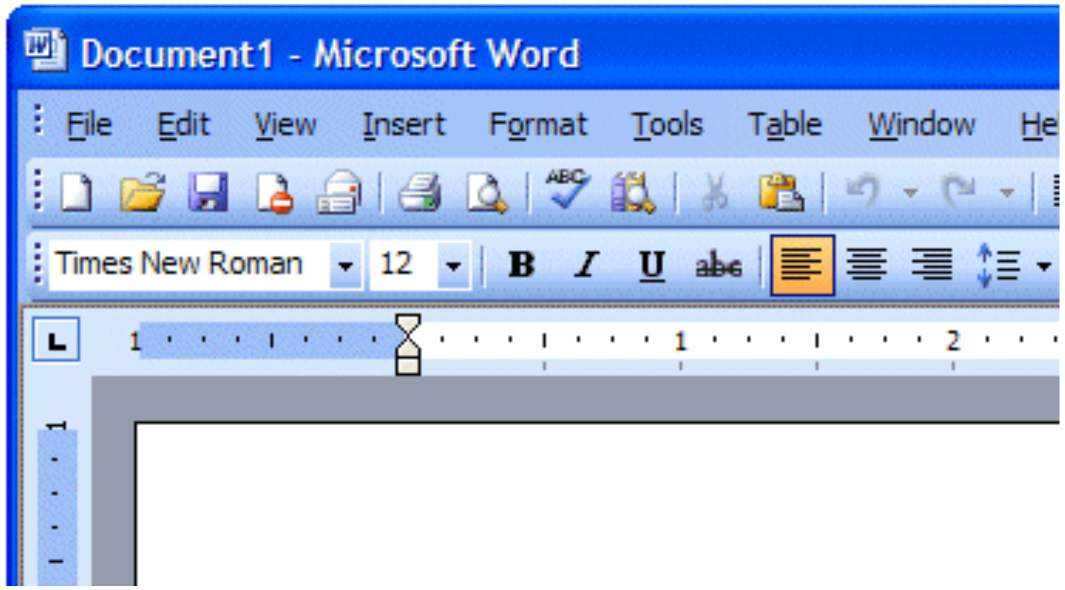
- Making controls **larger** reduces time to invoke actions
- Locating controls closer to user **cursor** reduces time
    - e.g., context menus



Design implications of Fitt’s law

- Positioning button or control along **edge** of screen acts as barrier to movement, substantially reducing homing time & errors

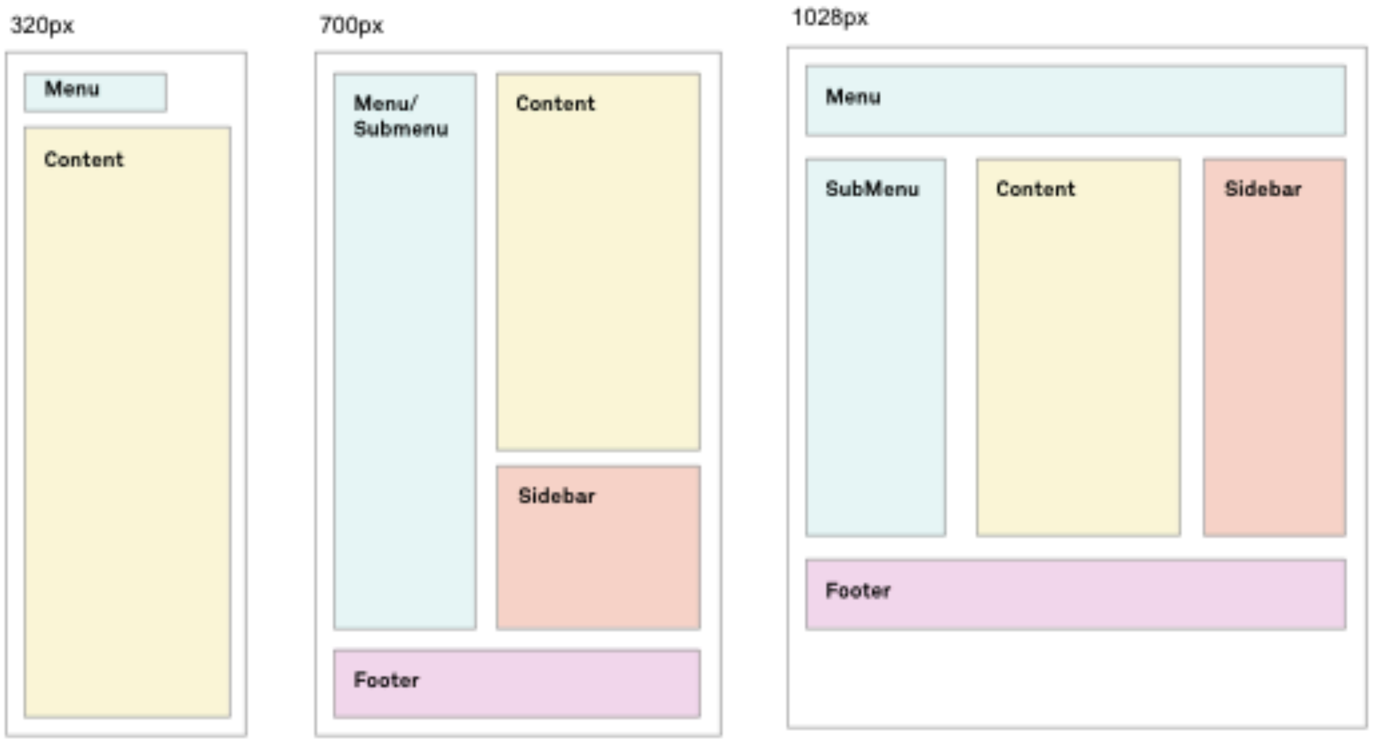




Mobile design

Responsive design

- Mobile devices often have smaller form factor than desktop / laptop OS
- Can design a separate UI
- Or may build a **fluid** UI that rescales for different display sizes



Where's the cursor?

- No cursor on many mobile devices
- Cannot use dynamic hinting to determine which elements can be interacted with
  - May require more use of static hinting
- Fitt's law still applies
  - Fingers are less sensitive, hard to select small buttons, occlude elements

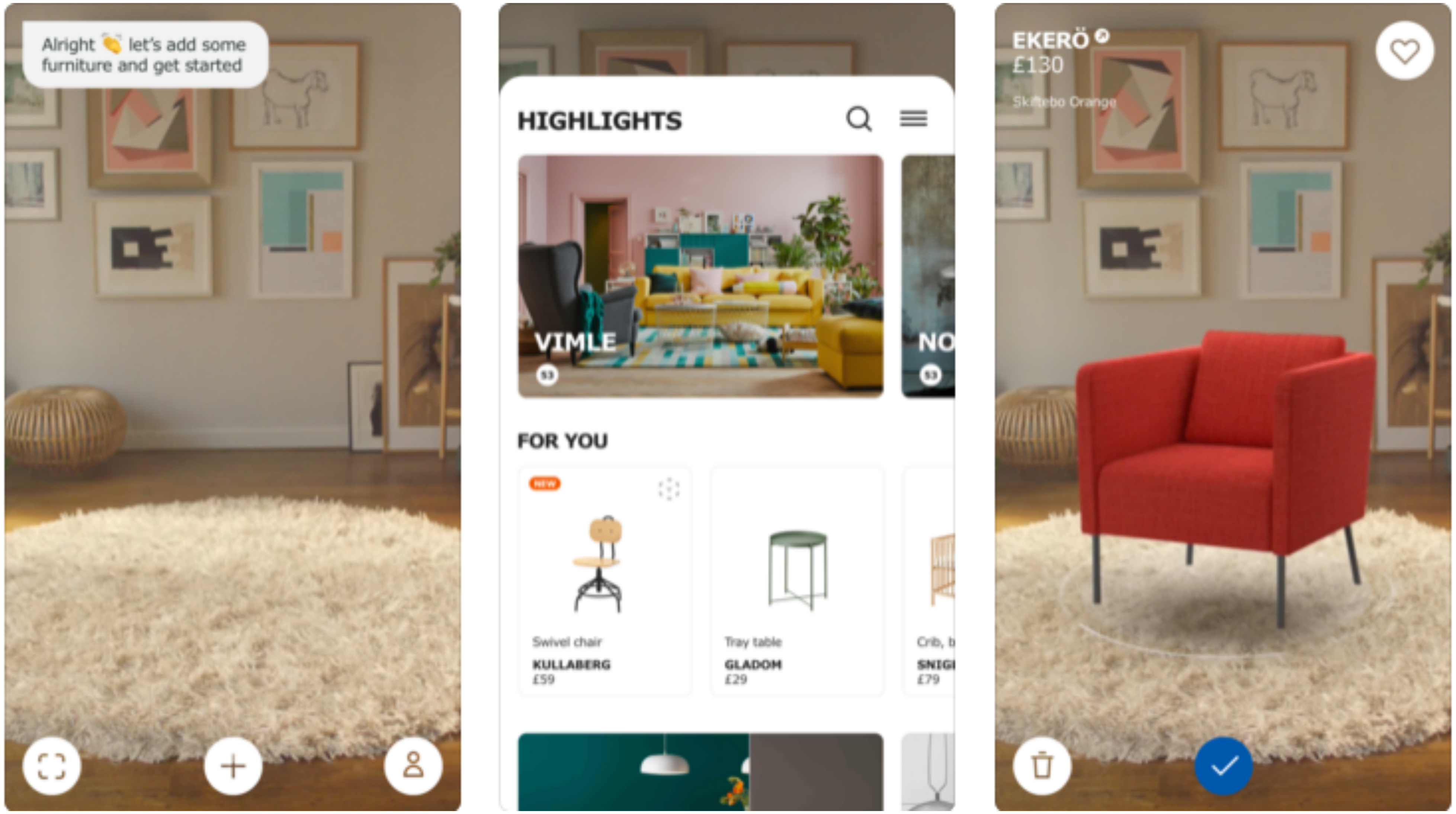
Alternative inputs

- Modern mobile devices often have a wide range of sensors which can be used for input
  - Camera
  - Microphone
  - Accelerometer
  - Three-axis gyro
  - GPS
  - Barometer
  - Proximity sensor
  - Ambient light sensor

- Enables new interaction techniques

Augmented reality

- Overlaying generated content on top of view of the real world



Alright 👏 let's add some furniture and get started

HIGHLIGHTS

VIMLE

NO

FOR YOU

NEW

Swivel chair
KULLABERG
£59

Tray table
GLADOM
£29

Crib, b
SNIGl
£79

EKERÖ
£130
Skiftebo Orange

Universal design

Supporting users with disabilities

- **Perception** - visual & auditory impairments
  - Blindness or visual impairments
  - Color blindness
  - Deafness & hearing limitations
- **Motion** - muscle control impairments
  - Difficulties with fine muscle control
  - Weakness & fatigue
- **Cognition** - difficulties with mental processes
  - Difficulties remembering
  - Difficulties with conceptualizing, planning, sequencing actions

Blindness and visual impairments

- Users use screenreader to listen to screen elements
- Reads all of the text on the page
  - Through practice, learn to listen to text at 400+ words per minute

- Important to have **alt-text**
  - Images should have labels that explain them
- Important to have **hierarchy**
  - Rather than visually skimming page, skims page by listening to section heads to determine which level to navigate to next

Universal design

- How can users with physical disabilities be supported in user interactions?
- Good: **assistive design** - offering equivalent actions for disabled users that cannot take normal actions
- Better: **universal design** - designing interactions so broadest set of users across age, ability, status in life can use normal actions

Example - Curb cut

- Initially designed for **accessibility** - support for disabled & wheel chairs
- But potentially benefits **all users** of public spaces - people w/ suitcases, hand carts, roller blades, bikes, …

7 Principles of Universal Design

- **Equitable use**: The design is useful and marketable to people with diverse abilities
- **Flexibility in use**: The design accommodates a wide range of individual preferences and abilities
- **Simple and intuitive**: Use of the design is easy to understand, regardless of the user's experience, knowledge, language skills, or current concentration level
- **Perceptible information**: The design communicates necessary information effectively to the user, regardless of ambient conditions or the user's sensory abilities
- **Tolerance for error:** The design minimizes hazards and the adverse consequences of accidental or unintended actions
- **Low physical effort**: The design can be used efficiently and comfortably and with a minimum of fatigue
- **Size and space for approach and use**: Appropriate size and space is provided for approach, reach, manipulation, and use regardless of user's body size, posture, or mobility

http://universaldesign.ie/What-is-Universal-Design/The-7-Principles/

In-Class Activity

In-Class Activity: Interaction Design Guidelines

- Take a design problem and build design recommendations
- List a set of alternative interaction techniques
  - Identify examples from desktop / web / mobile apps
- Offer guidance on pros and cons
- Identify mobile and universal design considerations

- (1) Navigating lists of items
  - Examples: grids, lists, pages of results, infinite scrolling, filtering
- (2) Invoking commands on content
  - Examples: toolbar, floating toolbar, cards, context menu, sidebar pane
- (3) Invoking top level commands
  - Examples: drawers, toolbar, menus, dialog
- (4) Entering formatted text
  - Examples: toolbar commands, Markdown, HTML
- (5) Panning and zooming
  - Example: zoom slider, scrollbars, pinch to zoom, drag to pan
- (6) Accelerometer-based control
  - Examples: shake to undo, rotate to pan, roll / pitch / yaw game control
- (7) Chat bots