Accessibility and the Web

Input, Interaction, and Accessibility
Spring 2019

Today

- Briefly:
  - Other laws of human perf
  - Beating Fitts' Law
- Web accessibility prep
- Intro to Project 1 (due in 3 weeks)

Related laws of human performance

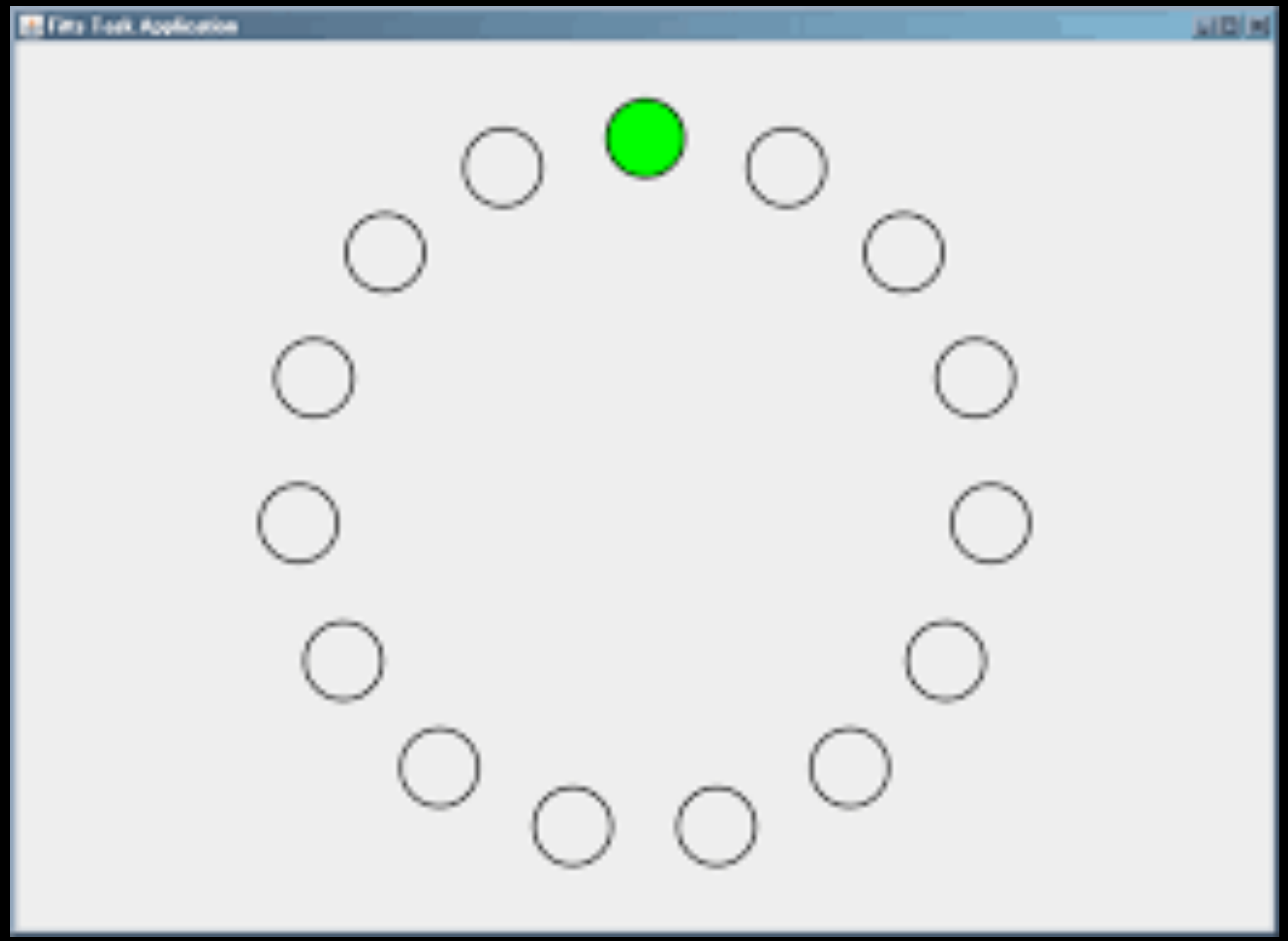
Choice reaction time

- Fitts' Law assumes that the target location is known
- What if the user needs to find the target first?

Hick-Hyman Law (1952-3)

- Given n equally probable choices, time to choose among them is approx.
  - $RT = a + b \log_2 (n+1)$
- 1/b is often 5-7 bits for adults
- This assumes all options are equally probable (but variation exists for unequal probabilities)

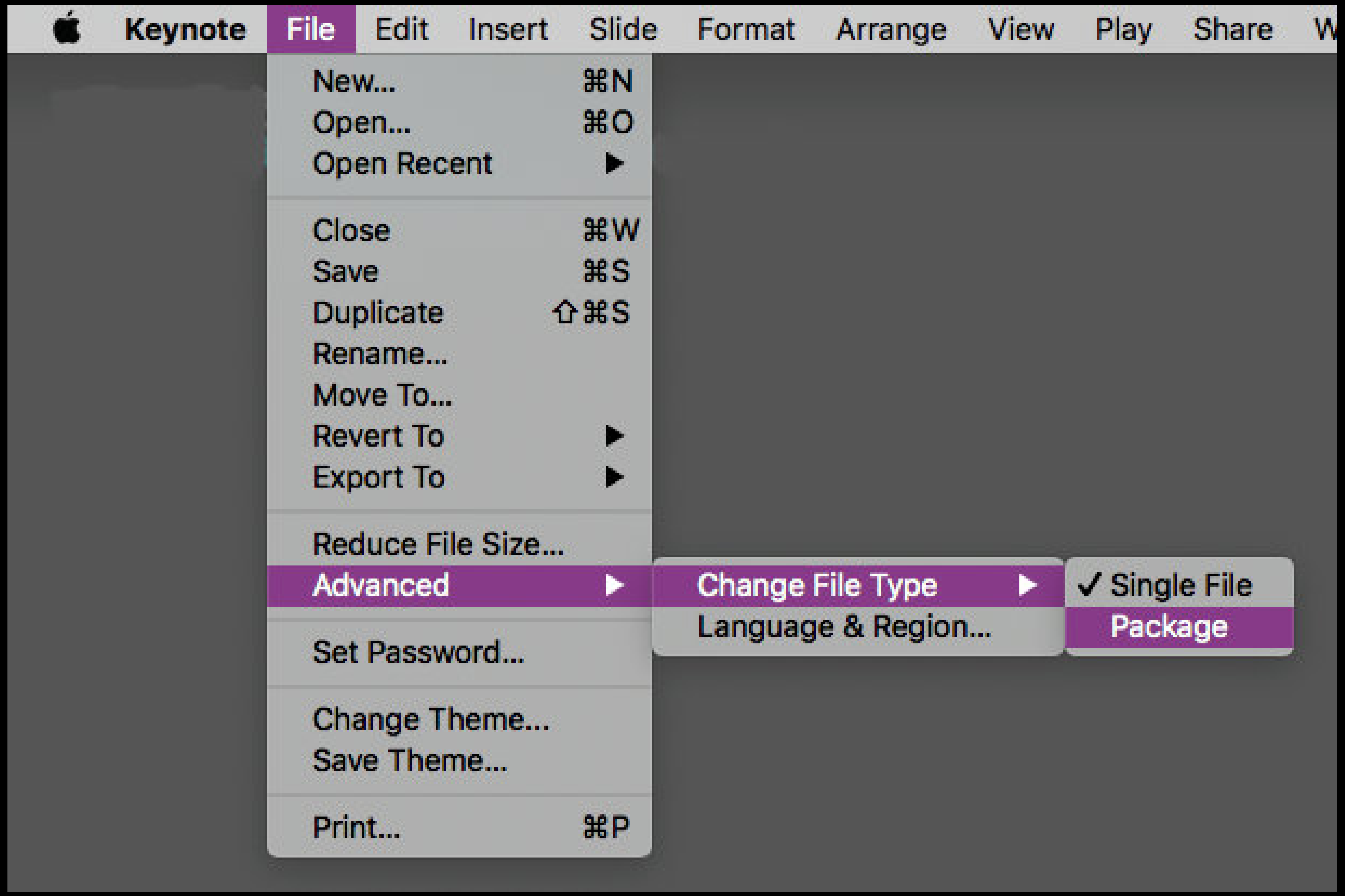
Using Hick-Hyman



- N buttons; one becomes highlighted
- Can combine with Fitts' to calculate *time to locate and then click*
- Used by Soukoreff and MacKenzie to calculate [theoretical upper bound for stylus and soft keyboard typing]()

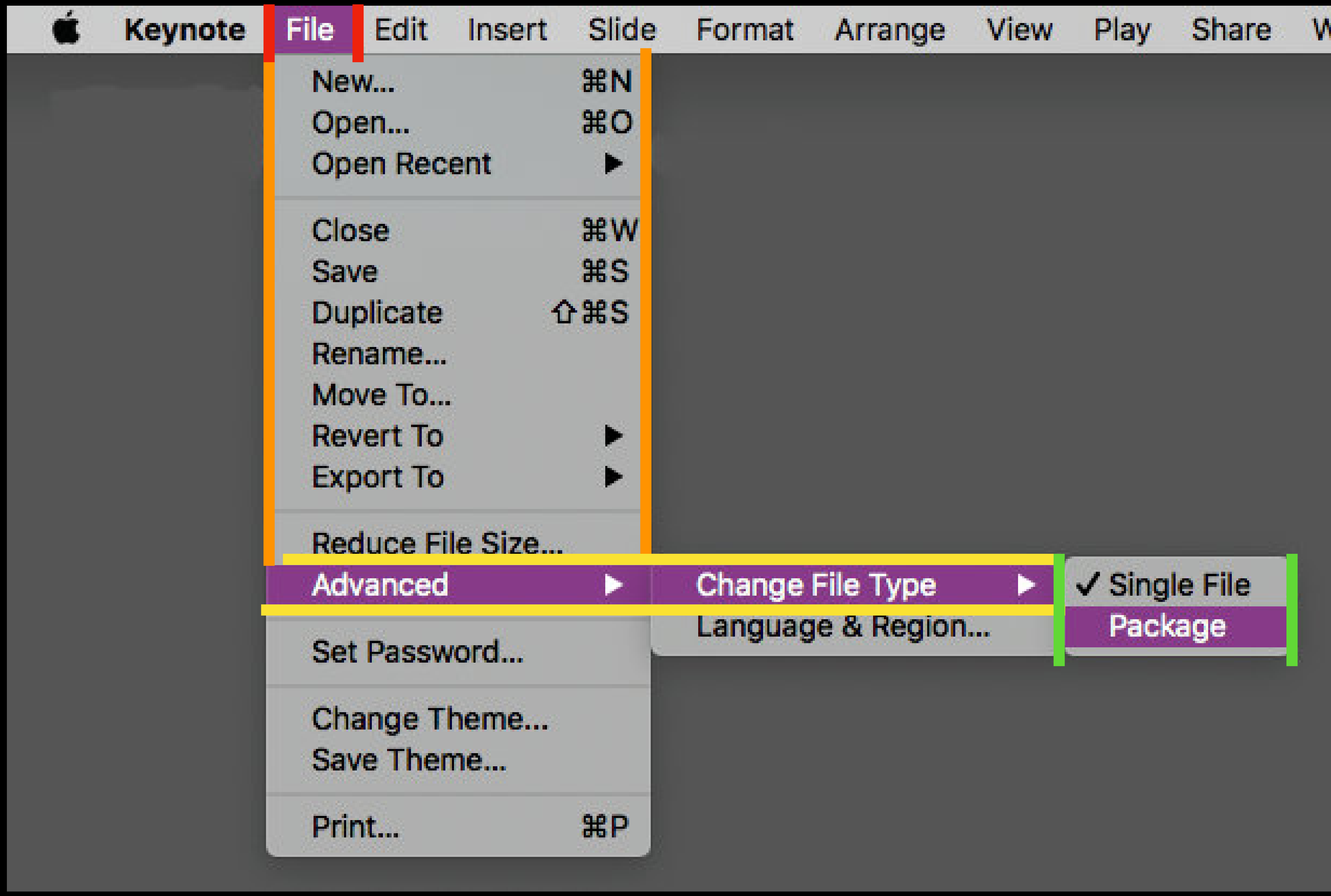
Steering law

- Fitts' Law only models a single target (essentially a 2D line)
- What about more complicated pointing tasks?
- These are *steering tasks*



Steering law

- User is navigating a 2D tunnel
- We essentially integrate along the curve of movement



Steering law (Accot and Zhai 1997)

$$T = a + b \int_C \frac{ds}{W(s)}$$

- T - time to navigate the path
- C - path
- W(s) - width of path at s
- a and b - index of performance (as in Fitts' Law)

Error Law (Wobbrock et al. 2008)

- Fix movement time (in studies, use a metronome)
- Derived the error law: $$P(E) = 1 - erf\left(\frac{2.066\frac{W}{A}\left(2^{\frac{MT_e - a}{b}} - 1\right)}{\sqrt{2}}\right)$$
- For more details, see the paper

Other laws of performance

- Power law of practice
- Guiard's model for bimanual tasks
- Schmidt's law for open-loop movement

Beating Fitts’ Law

Can we beat Fitts?

- What would it mean to beat Fitts Law?

Can we beat Fitts's Law?

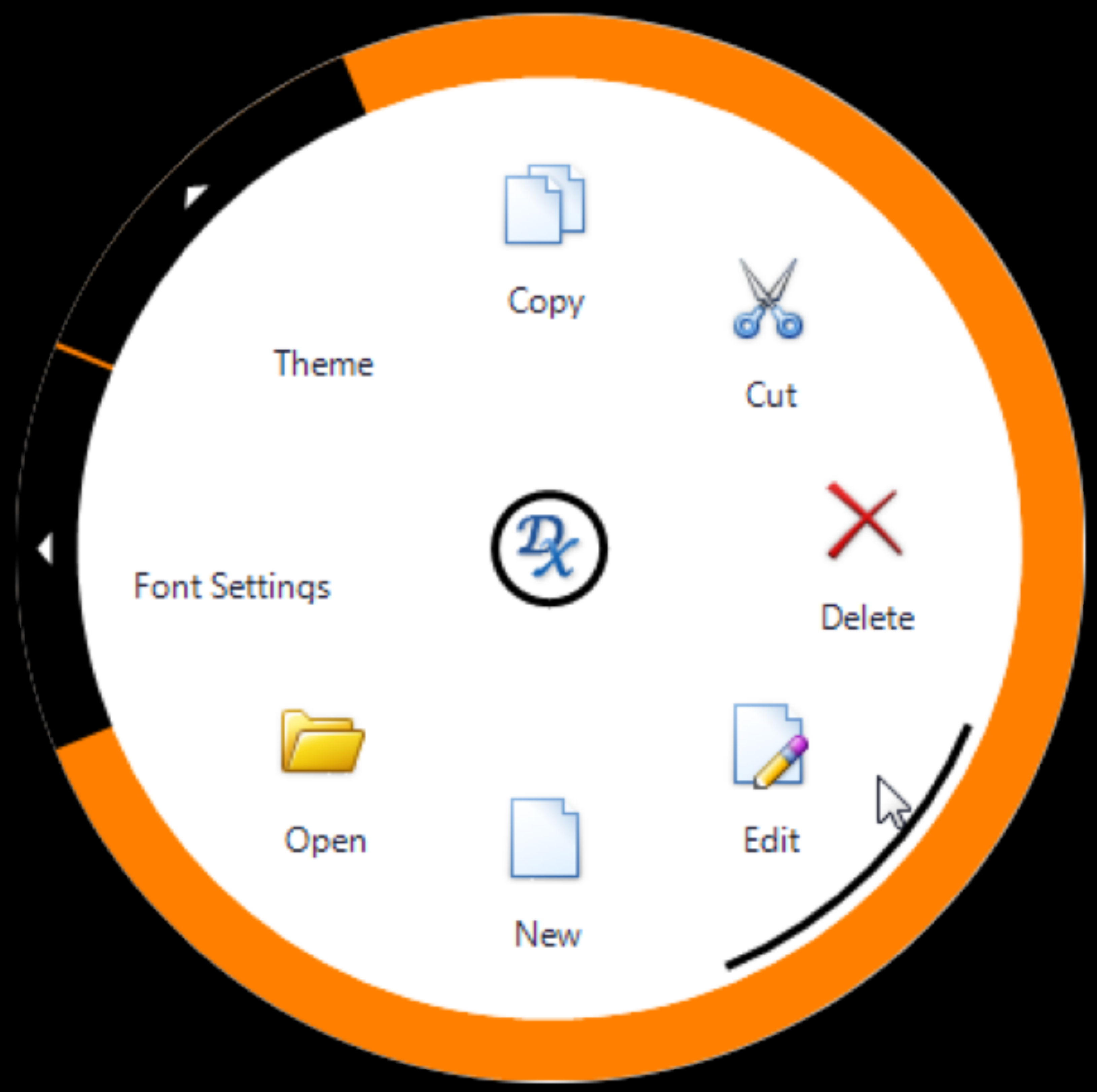
- How does Fitts's Law teach us how to improve performance?
  - Reduce (effective) distance to target
  - Increase (effective) size of target

Improving performance

- Pointing performance will be improved when targets are **nearby** and **large**
- So, design UIs such that important functions are nearby and large

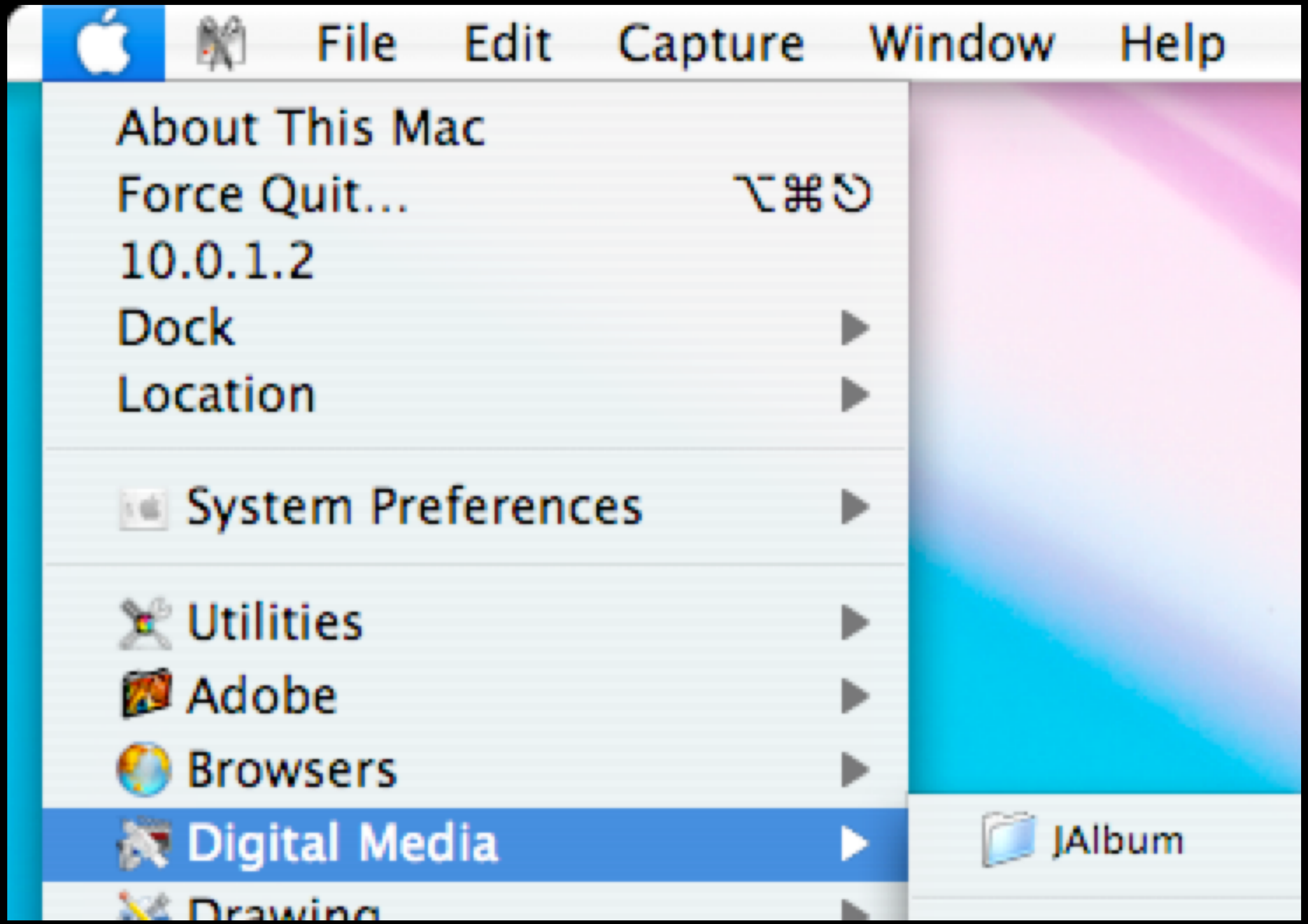
Radial menus

- Place menus around the cursor
- Often used in games and artistic software
- Limitations?



Edges and corners

- Edges and corners typically have infinite size
- No penalty for overshooting the target

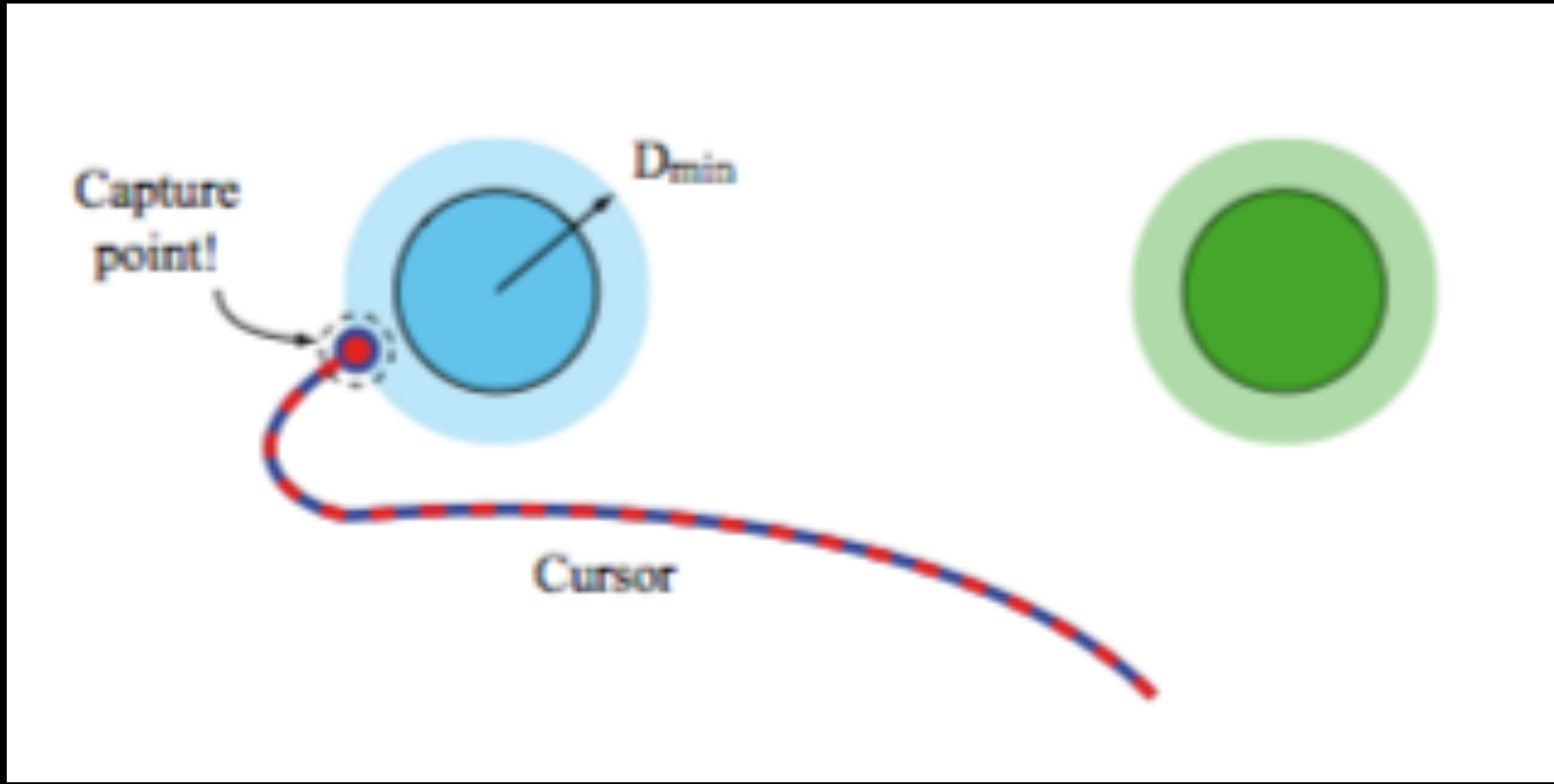


Pointing facilitation

- Two fundamental approaches
- **Target aware** - the system knows where targets are on the screen
- **Target agnostic** - the system does not know where targets are on the screen

Target aware: Facilitating movement

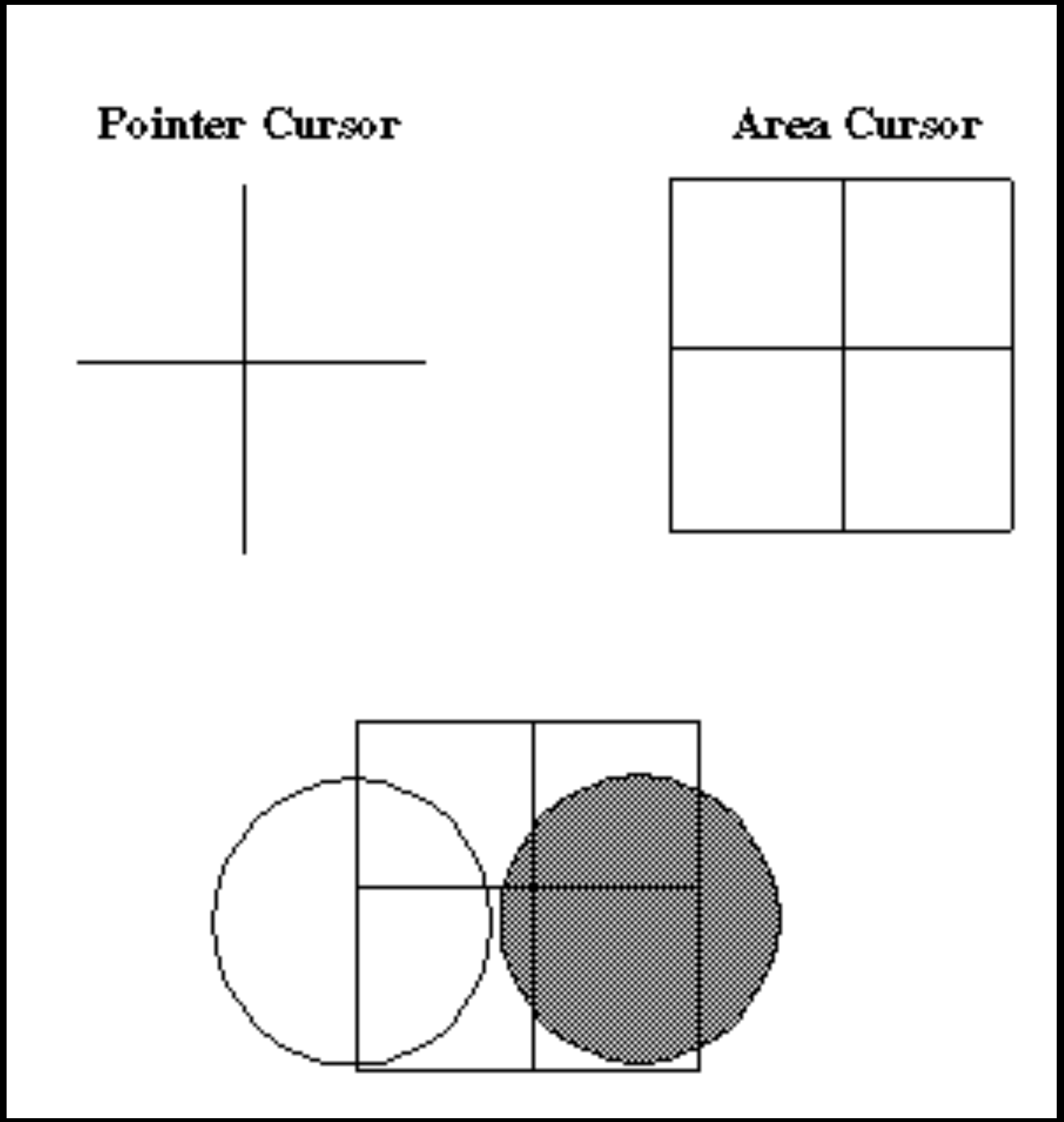
- Jump between icons or make “sticky” icons
- “Gravity fields”



Area Cursors

- One way to improve pointing is to increase the size of the cursor: easier to hit targets with a bigger cursor

- Downside: this method can break down when targets are close to each other



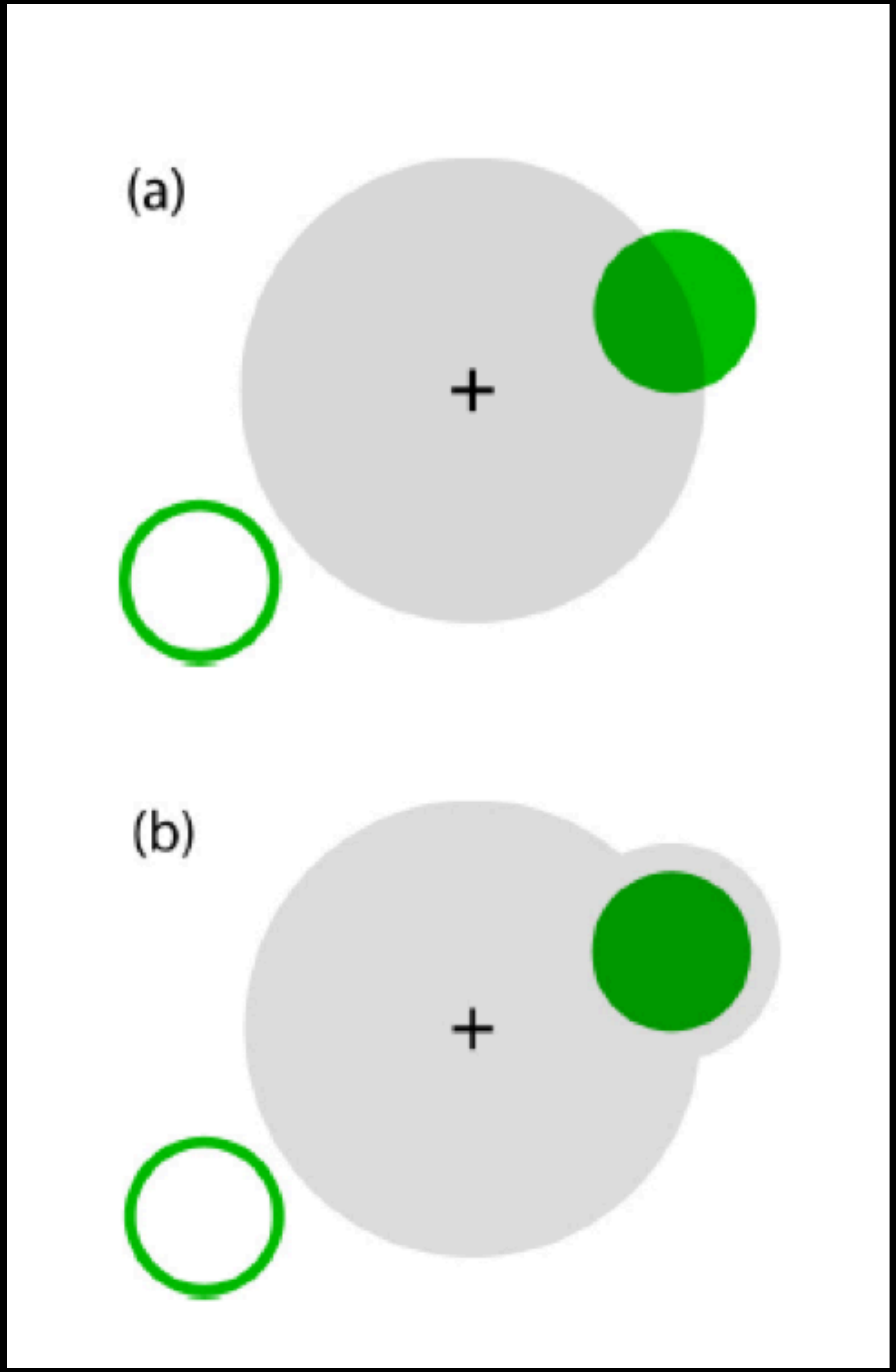
Limitation of area cursor

- What to do with densely packed targets?

Bubble Cursor

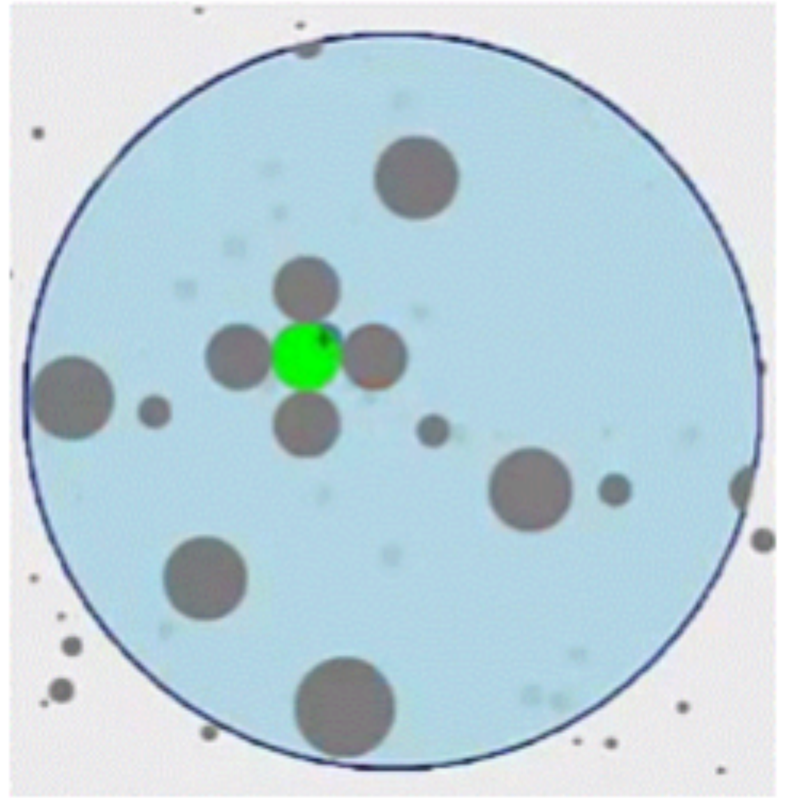
(Grossman and Balakrishnan 2005)

- Dynamically resize area cursor to contain the closest target, but not the second closest
- Considered by many to be the best pointer facilitation method



Beating the bubble cursor

The Bubble Lens



Martez E. Mott, University of Washington [contact]
Jacob O. Wobbrock, University of Washington

Download

Current Version 1.0.0

Windows executable: BubbleLensExe.zip
Source code: BubbleLens.zip
Pseudocode: Pseudocode (PDF)

The Bubble Lens requires the Microsoft .NET 2.0 Framework or later. Download it here.
This software is distributed under the New BSD License agreement.

About

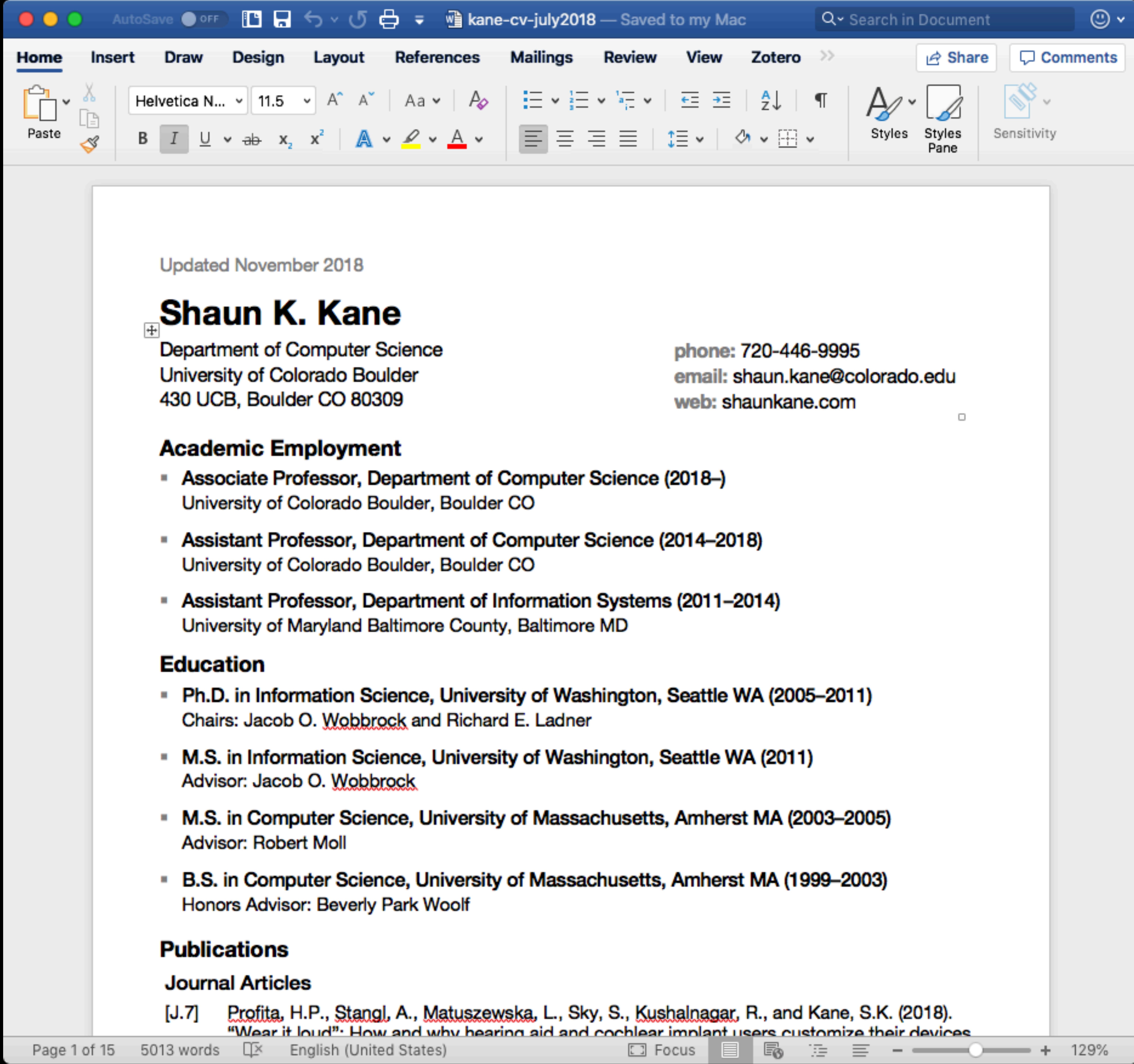
The Bubble Lens is a pointing facilitation technique that builds upon the Bubble Cursor by remedying the Bubble Cursor's poor performance around small, dense targets. Specifically, the Bubble Lens detects when the user is approaching a small, densely packed target and, as the user attempts to acquire this target, the cursor automatically magnifies into a circular lens, speeding the user's final stage of target acquisition. The automatic magnification is based on a technique called *kinematic triggering* that continually monitors a movement's velocity profile for the presence of certain events or conditions. In the case of the Bubble Lens, kinematic triggering fires the lens on the downslope of the first corrective movement in the velocity profile. In laboratory studies of random target-field pointing, the Bubble Lens was 10.2% faster and 37.9% more accurate than the Bubble Cursor.

Why target agnostic?

- Sometimes the programmer does not have the ability to programmatically identify UI layout (e.g. old UI toolkits)
- Sometimes the number of targets is too large

Where are the targets?

- Buttons, tabs, etc.
- In between every character



Target-agnostic methods

- Angle Mouse (Wobbrock et al.)
  - Detect when corrective movements start (based on speed and angular deviation)
  - Slow the cursor during correction phase

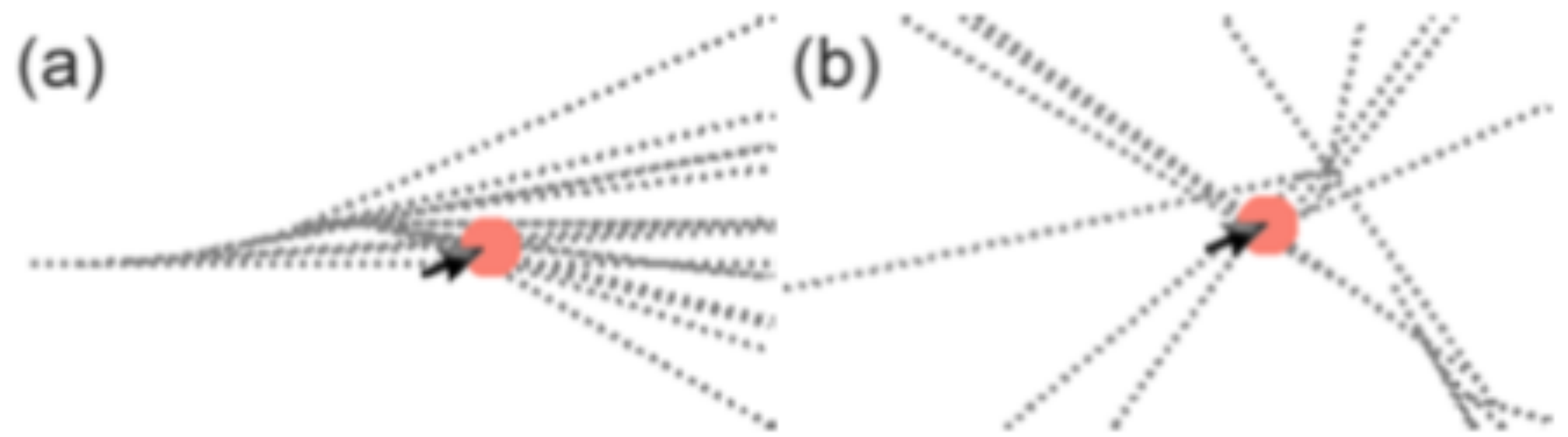


Figure 2. (a) Coherent movement produces low angular deviation (σ_θ), 13.7° in this example, and retains most of the maximum gain (89%). **(b)** Divergent movement produces high angular deviation (94.8°) and keeps less of the maximum gain (21%).

More target-agnostic methods

- Steady Clicks (Trewin et al.)
  - For pen-based interfaces
  - Some users, especially older users, accidentally “slip” the cursor while clicking
  - Detect this movement (based on characteristic shape) and find the target from a few seconds before

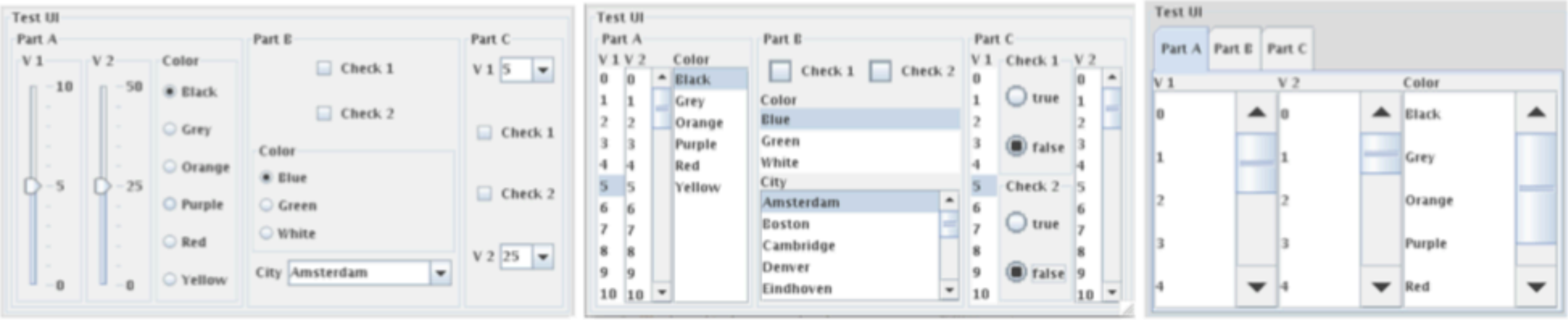
Implementing target aware techniques

- Incredibly difficult unless you control the OS
- Some research has used computer vision to reverse engineer the UI (see Morgan Dixon's Prefab system)
- Depends on there being an appropriate number of targets

User-specific models

- Some individuals may have additional, atypical challenges related to pointing
- Examples
  - Parkinson’s Disease may cause tremor when using a pointing device
  - Muscular Dystrophy may impair large movements, but doesn’t affect precision
- Can we build user-specific models of pointing?

Supple++ (Gajos et al.)



(a) (b) (c)

Figure 5: Three renderings of a synthetic interface used in the preliminary study and automatically generated under the same size constraint: (a) base line preference-optimized GUI; (b) personalized for the mouse user with muscular dystrophy (M03); (c) personalized for the eye tracker user (ET01). GUIs (b) and (c) were generated by SUPPLE++.

Roots of SUPPLE

- Originally developed to adapt remote controls to different devices

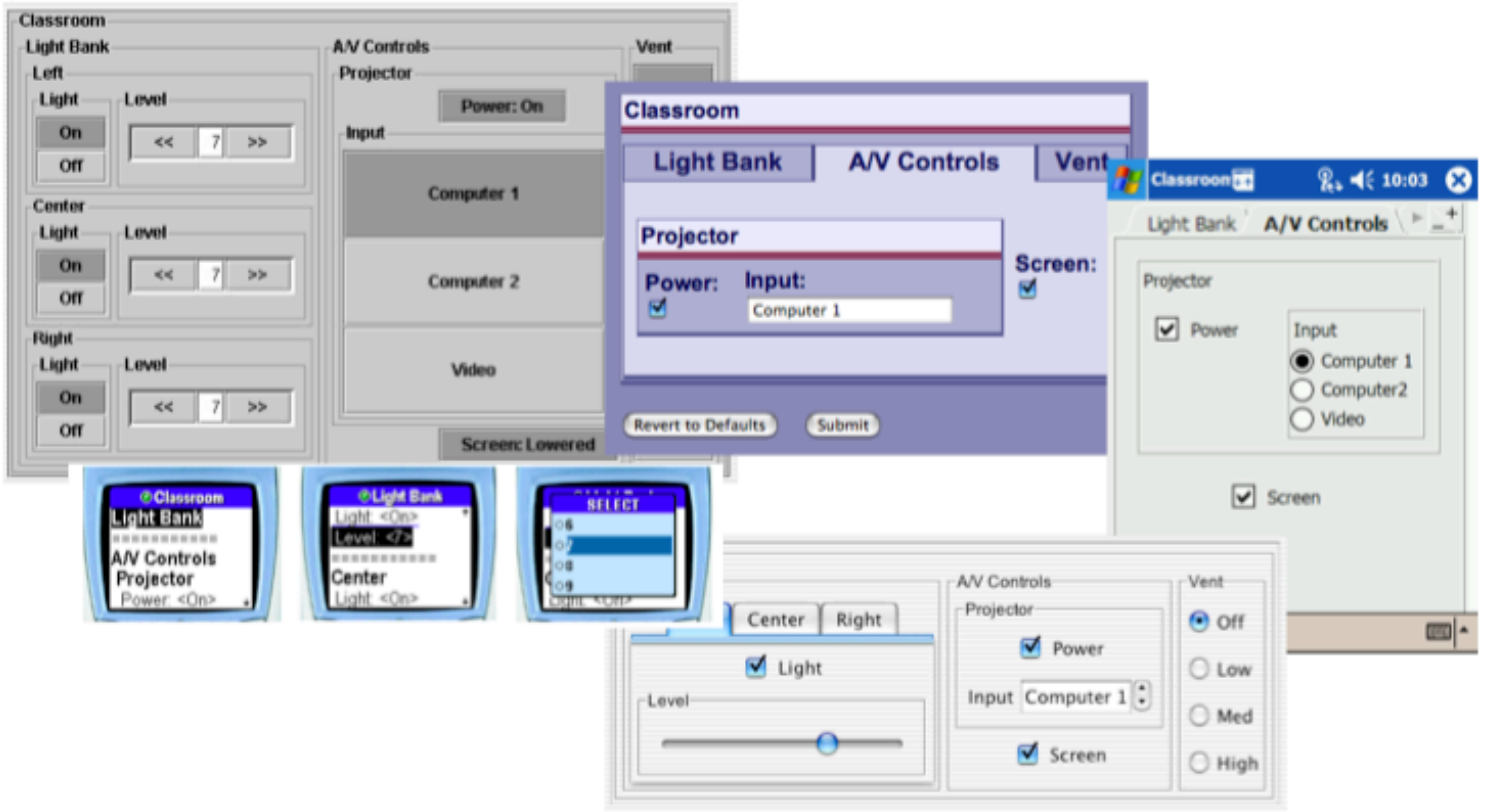


Figure 1: An interface for a simple application rendered automatically by SUPPLE for a touch panel, an HTML browser, a PDA, a desktop computer and a WAP phone.

Heuristic vs. data-driven

- Most of these approaches rely on some heuristic rules, identified and specified by a human programmer
- More modern techniques are often based on modeling a user (e.g. SUPPLE++)
- However, we still need to have an understanding of the problem space to understand possible solutions

Mini-design activity

- Consider our phone dialer user interface
- Sketch out a variant for
  - User with tremor
  - User with hand fatigue
  - User with low vision (but fine motor ability)

Accessibility as a design perspective

10,000 foot view

- Beginning our exploration of interaction challenges and solutions
- Exploring different kinds of interactions
- How to engineer accessible user interfaces

Considering diverse users

- How do we think about the individual?
- How do we think about the design process?

A quick note on terminology

- Class follows the **reasonable person principle:** assume everyone is reasonable, acting in good faith
- Disability isn't a bad word
- Some terms have gone out of fashion (e.g., wheelchair bound, mental retardation)
- Some terms raise questions: what does it mean to call something **normal?**

Models of disability

- **Medical model:** people with disabilities (PWD) are patients; our goal is to restore or replace function
- **Social model:** What counts as “disability” is defined socially; people are naturally different, but society sets the bounds for what is considered normal and what is abnormal. Maybe we don't need to fix people.

The social model in action

- [The Shared Sign Language of Martha's Vineyard]
- In the 18th century, deafness was common on the island of Martha’s Vineyard
- Sign language was commonly used by everyone

By the early 18th century, it was not uncommon for people in Martha's Vineyard to be deaf from birth. This had a profound effect on the culture of Martha's Vineyard - and one that went on to influence Deaf culture in the United States as a whole.

Definitions change over time

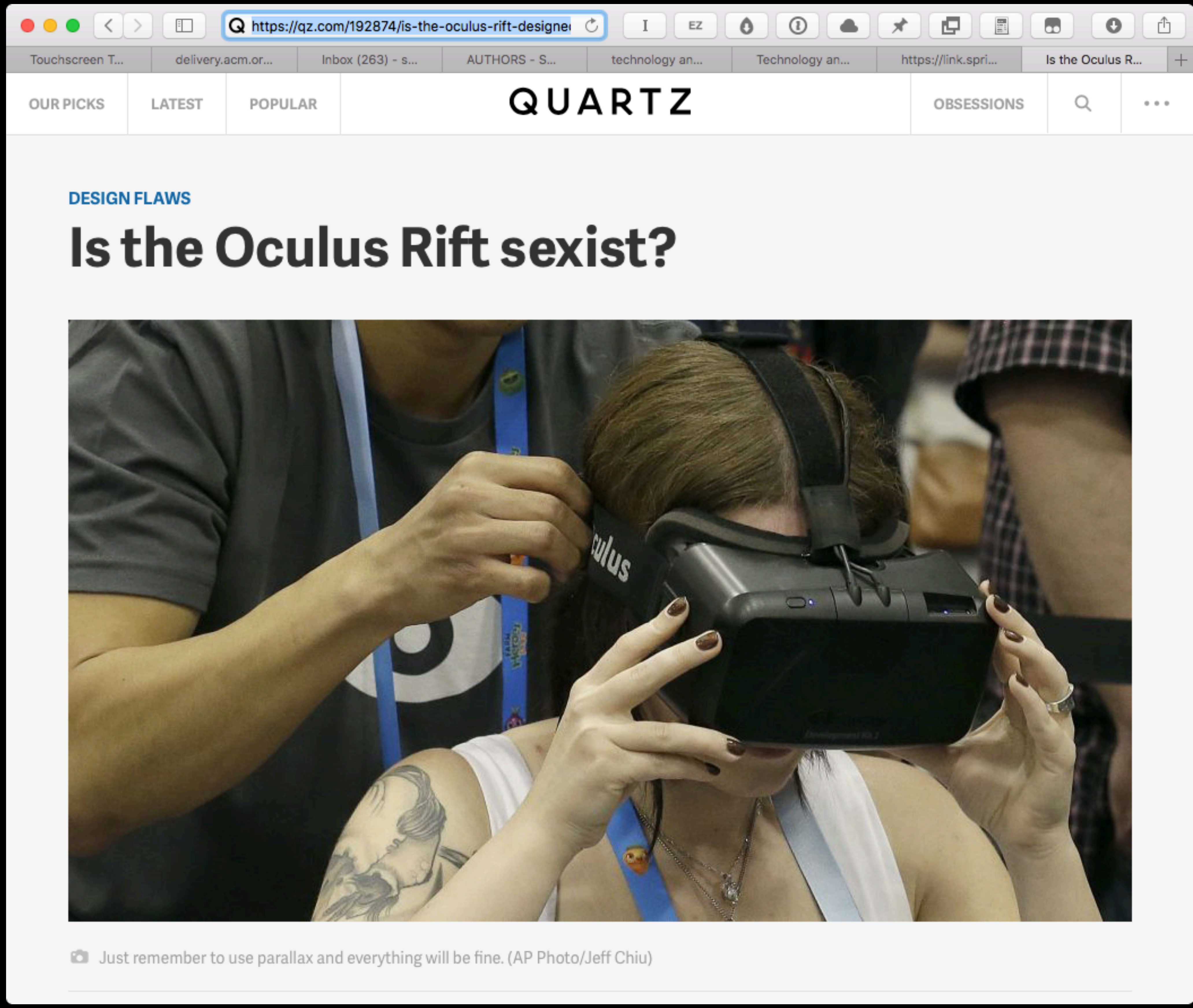
- Being nearsighted was not a significant disability until literacy become common
- Changing norms around texting may benefit some (e.g., deaf people) and impair others (e.g., people with dyslexia)
- Other examples from history? Possible examples in the future?

WHO model of disability (2012)

- **Body functions** and **body structures** describes how a person's body functions
- **Impairments** refer to functional limitations of a person's body
- **Activity limitations** refer to difficulties performing specific activities
- **Participation restrictions** refer to difficulties participating in activities of daily life
- (and all of this is affected by environmental factors)

Is VR sexist?

- Two ways that humans detect depth: motion parallax, shape-from-shading
- danah boyd found that women favor shape-from-shading; men the opposite
- But most current VR systems use motion parallax (it's also easier to compute)



But really, is VR sexist?

- Let’s say you work at Oculus and verify the results of this study
- Let’s discuss arguments on both sides
  - 1: Oculus is responsible for addressing this
  - 2: Oculus is not responsible for addressing this
  - Either way, what do we do about it?

Comments on sex and VR

In-groups, out-groups, group size

- One thing that is tricky with all of this work is dealing with in-group and out-group effects, and our own ability to empathize outside of our own identity.
- Often, looking at a larger group (e.g. women) feels different than a smaller group (e.g. people with disabilities). But why is it that way?
- For context, about 32 million PWD in US; about 5 million Chinese Americans

Perspectives and solutions

“What should we do about it?”

- **Rehabilitation technology** - enable people to restore natural function (usually temporary)
- **Assistive technology** - provide technology to solve an accessibility problem (usually permanent)
- **Special education** - focus on access to education and empowerment
- **Legal perspective** - focus on rights and societal supports
    - Whose responsibility is this problem?

Design approaches

- **Assistive technology** - developing technology to “repair” a disability or address an activity limitation
- **Universal design** - build it from the ground up to accommodate the widest user group
- **Inclusive design** - often more pragmatic than universal design (include more, but not everyone), emphasizes participation by excluded users in the design process

More design approaches

- **Ability-based design** - develop systems that model and adapt to a user's ability
- **Design for empowerment** - emphasize empowering and increasing representation from marginalized groups
- **Value Sensitive Design** - consider other values in design other than usability (e.g., independence, self-determination)
  - Example: if we could, should we replace ASL interpreters with ASL-translating smart gloves? Why or why not?

But who is right?

- ¯_(ツ)_/¯
- All of these perspectives may be useful at certain times
- Not necessarily mutually exclusive
- Useful to be aware of any implicit assumptions we are carrying with us

Setting norms

- An eternal challenge for designers of more accessible technology is setting norms - what should we consider a “normal” level of caring about this stuff?
- Not taking a stance isn’t very different than taking a stance; it has effects either way

Considering impact

- Often we are considering trade-offs between the size of the group we are including and the magnitude of the impact
- Consider:
  - Helping 1 million people do their jobs 1% faster
  - Helping 100k people gain access to education
  - Helping 500 people live independently
- In reality, we are often dealing with external constraints. But it’s worth taking a moment to consider this in the abstract case.

Questions? Thoughts?

OK, so what do we actually do about it?

Why make things accessible?

- Laws or regulations
- Because we want to do the right thing
- Because accessible design is good design

Laws and regulations in the US

- Section 508 of Rehabilitation Act - guidelines for access to electronic resources
- Americans with Disabilities Act - guidelines for “reasonable accommodations” to work, education, public life
- Individuals with Disabilities Education Act (IDEA) - guidelines for access to education

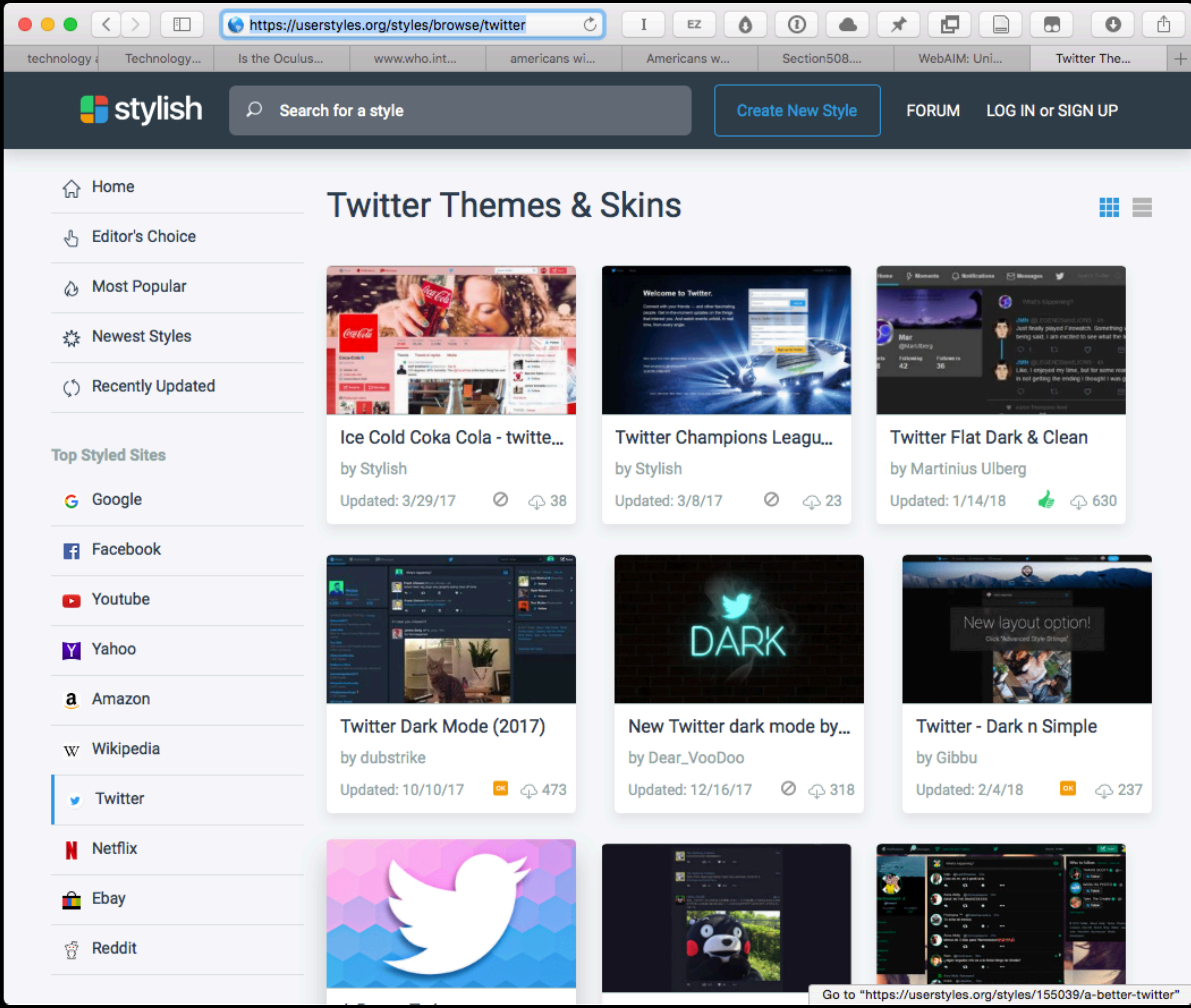
How to know what to do?

- In the old days, sometimes it was difficult or impossible to create accessible versions of our programs
- Fortunately, we now have better tools on our side:
  - Programming frameworks that support accessibility (if you do the right thing)
  - Guidelines for accessible design (and sometimes automated testing tools)

HTML is great for this

- Well defined standards for creating accessible content
- Web Content Accessibility Guidelines (WCAG)
- By design, the web separates content (HTML) from presentation (CSS)
- Everything is rendered as HTML, can be scraped or altered by user’s own device

Remixing the Web



Making HTML accessible

- We’ll get into the guts of how to do it in code on Wednesday
- But first, let’s think about what it actually means
- So: what does it mean for a web application to be accessible?

What IS accessibility on the web?

Ways to think about accessibility

- Following standards - this helps our design work on a variety of current and future devices
- Presenting content in multiple ways - since people may consume media in different ways, have we included the appropriate content for doing so?
- Supporting various input methods - including mouse, touch, keyboard

More ways to think about accessibility

- Using appropriate accessibility features of HTML (e.g., alt tags for images, ARIA labels for dynamic user interfaces)
- Semantic UI code - should represent the logical structure of the interface, not just visual
- Support end-user customization - e.g. font size, color contrast

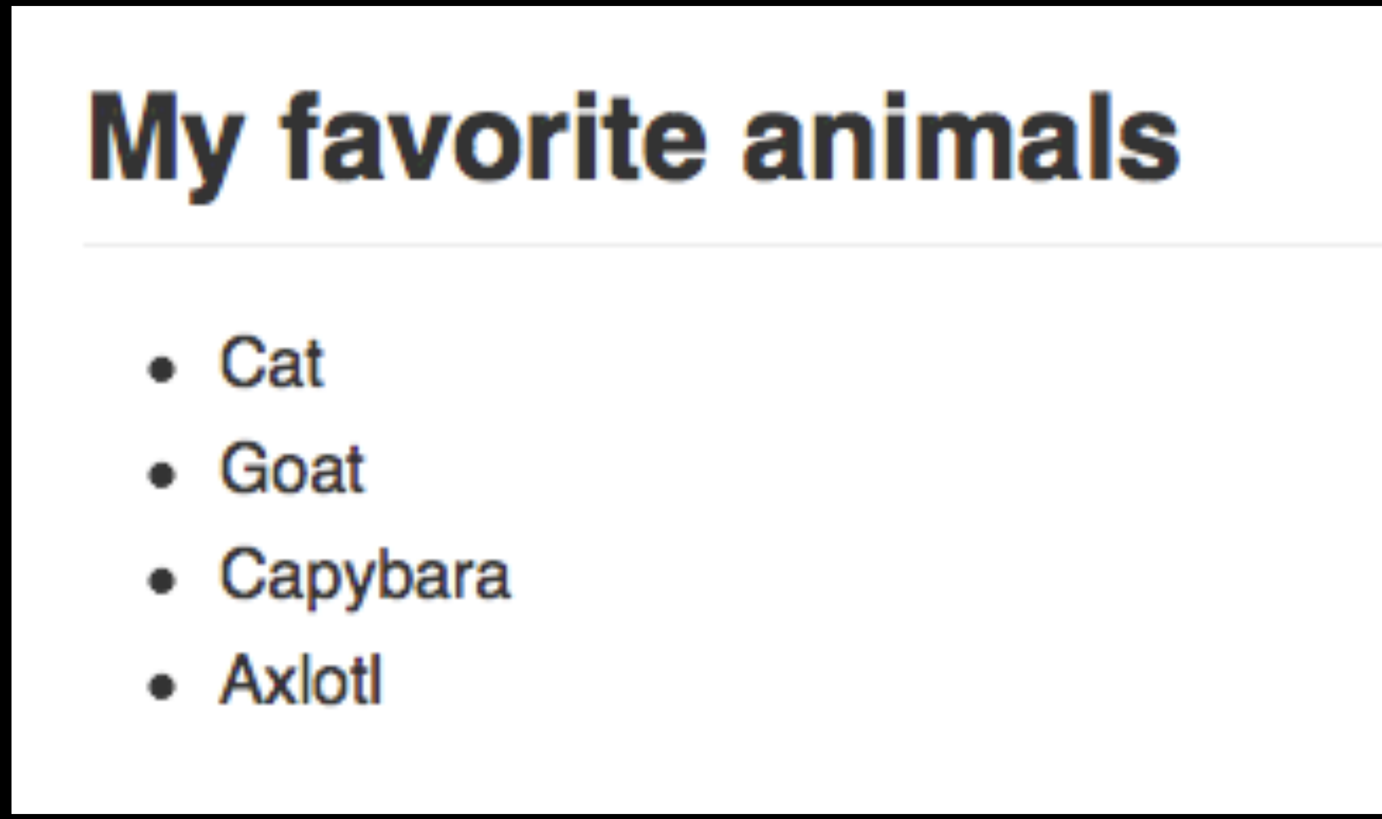
Where we go wrong

- Writing buggy UI code
- Neglecting to include alternative representations (image descriptions, video captions) - or including lousy versions of these
- Getting the visual presentation right but messing up the semantic representation

Semantic UI?

```
<h1>My favorite animals</h1>
<ul>
<li>Cat</li>
<li>Goat</li>
<li>Capybara</li>
<li>Axlotl</li>
</ul>
```

```
<p><strong>My favorite animals
</strong></p>
<p>• Cat</p>
<p>• Goat</p>
<p>• Capybara</p>
<p>• Axlotl</p>
```



Wednesday

- We’ll explore some concrete fixes we can make to our web code

Project 1

- Due in 3 weeks (Feb 25th)
- Build a small interactive app in HTML/CSS/JavaScript