CHAPTER

Our Hand–Eye Coordination Follows Laws

13

Have you ever had trouble hitting a tiny button or link on a computer screen or smartphone, or trouble keeping a screen-pointer inside a narrow path required to reach a menu item or link?

Perhaps your hand was jittery because you had recently consumed too much coffee or taken a medication. Maybe it was shaking due to high anxiety, anger, or fear. Maybe you have Parkinson's disease, which makes your hands shake, or arthritis in your hands and arms that restricts your hand movement. Maybe your arm was restricted because it was temporarily in a cast or sling. Maybe you were trying to text someone while riding on a bumpy bus, train, or horse. Maybe you were using an unfamiliar pointing device. Or maybe the target was simply too small or the allowed path too narrow.

It turns out that pointing at objects on a display and moving pointers along constrained paths follow consistent, quantitative laws.

FITTS' LAW: POINTING AT DISPLAYED TARGETS

The law for pointing at targets is called Fitts' law, named after the man who discovered it, Paul Fitts[1] (Fitts, 1954; Card et al., 1983). This law says that the larger your target on a screen and the nearer it is to your starting point, the faster you can point to it, the less mental effort required, and the more likely you are to hit the target. The formula for Fitts' law allows you to predict the time for a person to move a pointer—including a finger—to a target a specified size and distance from the pointer's starting point.

$$T = a + b \log_2 (1 + D / W)$$

T is the time to move to the target, D is the distance to target, and W is the width of the target along the direction of movement of the pointer. From the formula, you can see that as the distance (D) increases, the time (T) to reach the target increases, and as the target width (W) increases, the time to reach it decreases (see Fig. 13.1).

Fitts' law is quite general. It applies to *any* type of pointing: mouse, trackball, track pad, joystick, even fingers on touch screens. It also applies to everyone regardless of

[1] The "s" is in his name. Some researchers prefer "Fitts's", but I prefer "Fitts'".

Designing with the Mind in Mind. https://doi.org/10.1016/B978-0-12-818202-4.00013-1

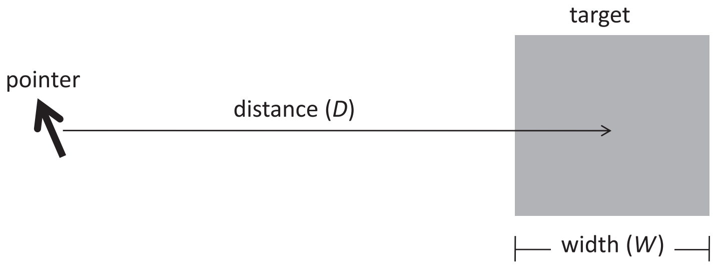


FIGURE 13.1

Fitts' law: pointing time depends on distance (*D*) and the width (*W*) of the target.

age, physical abilities, or mental state. But people vary in how fast they can move, and devices vary in how fast they can be moved, so the formula includes the parameters *a* and *b* to adjust it for those sorts of differences: *a* is the ease of starting and stopping the movement, and *b* is a measure of the average difficulty of moving the hand and pointing the device (if any).

The dependency of pointing time on target size and distance codified in Fitts' law can be understood by considering how screen-pointers move. A person sees a target on a screen and decides to hit it. Hands and pointing devices have inertia, so movement toward the target starts slowly but rapidly accelerates until reaching some maximum speed. This initial movement is fairly gross; it is essentially a shot in the general direction of the target without much control. We call the initial shot *ballistic*, like a shell fired from a cannon. As the pointer nears the target, the movement speed slows as the person's hand–eye feedback loops take control. The movement ends slowly, with finer and finer corrections, until the pointer is on the target (see Fig. 13.2).

Although the basic prediction of Fitts' law—people hit on-screen targets faster the closer and larger they are—seems pretty intuitive, the law also predicts something less intuitive: the *more* the distance decreases or the target grows, the *less* the decrease in pointing time. If a target is tiny and you double its size, the time people take to hit it decreases, but if you double the size *again*, the pointing time doesn't improve quite as much. So beyond a certain size, making a target even larger provides little added benefit (see Fig. 13.3), and below a certain distance, moving a target even closer doesn't help much.

A final noteworthy prediction of Fitts' law is that if the pointer or finger is blocked from moving past the edge of the screen, targets at the edge will be very easy to hit. People can just yank the pointer toward the target until the edge stops it with no need for slow, fine adjustments at the end of the movement. Thus, from the point of view of Fitts' law, targets at the screen's edge behave as if they were much larger than they actually are. However, this edge-pointing detail of the law applies mainly to desktop and laptop computers, because modern smartphones and tablet computers don't have raised edges that physically stop fingers.

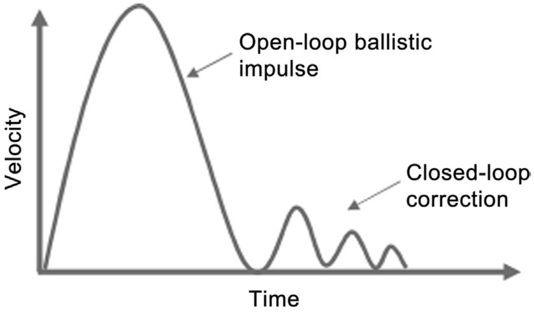


FIGURE 13.2

Graph of velocity over time as pointer moves to target. *Courtesy of Andy Cockburn.*

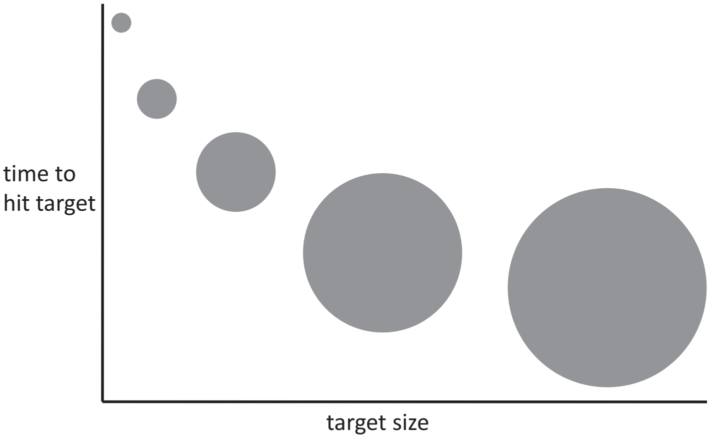


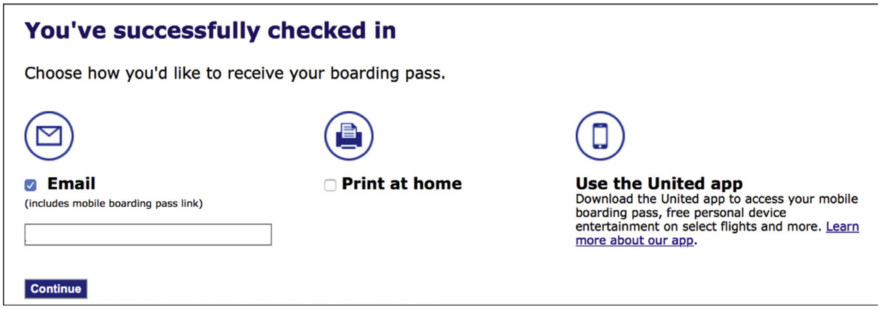
FIGURE 13.3

Diminishing marginal benefit of increasing click-target size (if *D* is constant).

Design implications of Fitts' law

Fitts' law is the basis of several common user-interface design guidelines:

- Make click-targets—graphical buttons, menu items, links—big enough that they are quick and easy for people to hit. Don't make people tap or click on tiny targets. That will slow everyone down and cause some to miss their target. An example of click-targets that are too small is provided by United.com's boarding pass delivery option page (see Fig. 13.4). To choose an option, users must click directly on the tiny checkboxes in United.com's boarding pass delivery option



You've successfully checked in

Choose how you'd like to receive your boarding pass.

☑ Email
(includes mobile boarding pass link)

□ Print at home

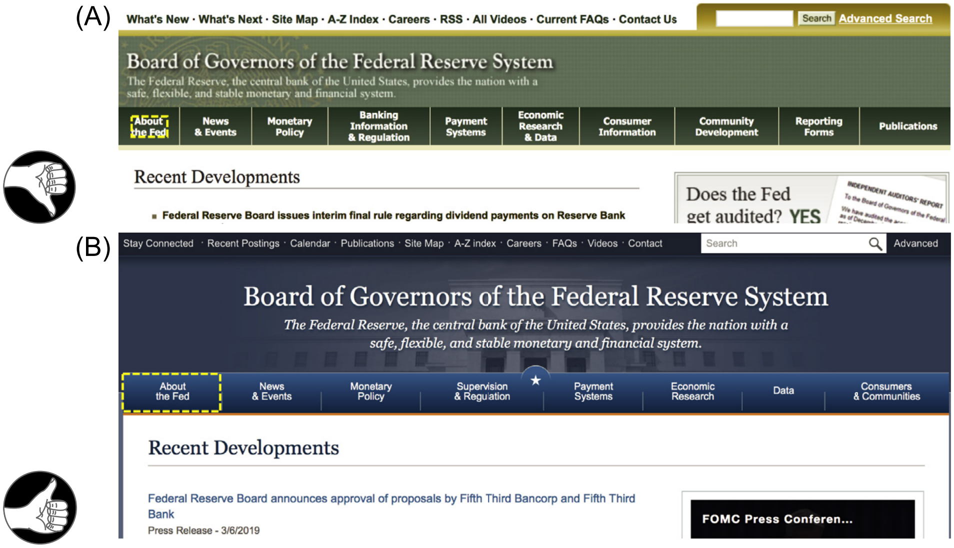
Use the United app
Download the United app to access your mobile boarding pass, free personal device entertainment on select flights and more. Learn more about our app.

FIGURE 13.4

Users must click on tiny checkboxes on United.com's boarding pass delivery option page; the symbols and text labels near the checkboxes do not accept clicks.

page. The circled symbols and text labels near the checkboxes do not accept clicks. Making them accept clicks would not be difficult and would greatly increase the effective size of the click-targets. See Johnson and Finn (2017) for guidelines on how large a click-target or tap-target must be to be reliably clickable by people of all ages and in all situations.

- Make the *actual* click-target at *least* as large as the *visible* click-target. Above all, don't frustrate users by presenting big buttons that only accept clicks on a small area (e.g., the text label), as the Federal Reserve Bank did in the navigation buttons on its 2016 website (see Fig. 13.5A). Accept clicks over at least the entire area of the visible click-target, as the improved 2019 Federal Reserve Bank website did (see Fig. 13.5B). If the visible target must be small (e.g., a small word embedded in text), design the user interface to treat clicks *near* the link as if they were *on* it.
- Checkboxes, radio buttons, and toggle switches should accept clicks on their labels as well as on the buttons, thereby increasing the clickable area.
- Leave plenty of space between buttons and links so people don't have trouble hitting the intended one.
- Place important targets near the edge of the screen to make them very easy to hit.
- Display choices in pop-up and pie menus if possible (see Fig. 13.6). They are faster to use than pull-down menus, on average, because they open "around" the screen pointer rather than below it, so users need to move the pointer less to reach most of the items. However, even pull-down menus are faster than pull-right ("walking") menus.
- For smartphone apps, consider using menus that people can reach easily with a thumb when holding the phone normally (see Fig. 13.7).


(A)
What's New · What's Next · Site Map · A-Z Index · Careers · RSS · All Videos · Current FAQs · Contact Us
Search
Advanced Search
Board of Governors of the Federal Reserve System
About the Fed
News & Events
Monetary Policy
Banking Information & Regulation
Payment Systems
Economic Research & Data
Consumer Information
Community Development
Reporting Forms
Publications
Recent Developments
Does the Fed get audited? YES
INDEPENDENT AUDITORS' REPORT
(B)
Stay Connected · Recent Postings · Calendar · Publications · Site Map · A-Z index · Careers · FAQs · Videos · Contact
Search
Advanced
Board of Governors of the Federal Reserve System
About the Fed
News & Events
Monetary Policy
Supervision & Regulation
Payment Systems
Economic Research
Data
Consumers & Communities
Recent Developments
FOMC Press Conferen...

The Federal Reserve, the central bank of the United States, provides the nation with a safe, flexible, and stable monetary and financial system.

Federal Reserve Board issues interim final rule regarding dividend payments on Reserve Bank

The Federal Reserve, the central bank of the United States, provides the nation with a safe, flexible, and stable monetary and financial system.

Federal Reserve Board announces approval of proposals by Fifth Third Bancorp and Fifth Third Bank

Press Release - 3/6/2019

FIGURE 13.5

(A) In 2016, the FederalReserve.gov website's navigation-bar "buttons" accepted clicks only on the label. (B) In 2019, its updated website accepted clicks anywhere in each link's rectangle.

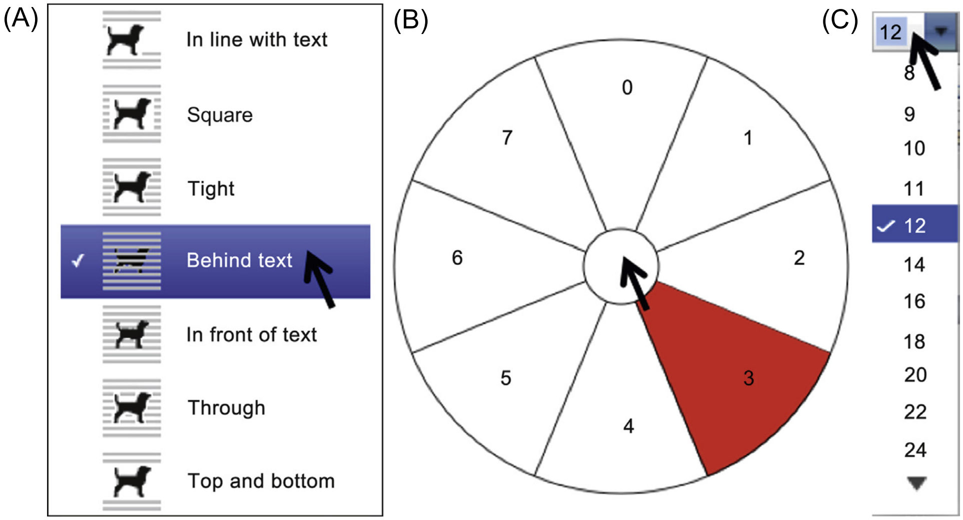


FIGURE 13.6

Types of menus for desktop/laptop computers: (A) pop-up menu, (B) pie menu, and (C) pull-down menu.

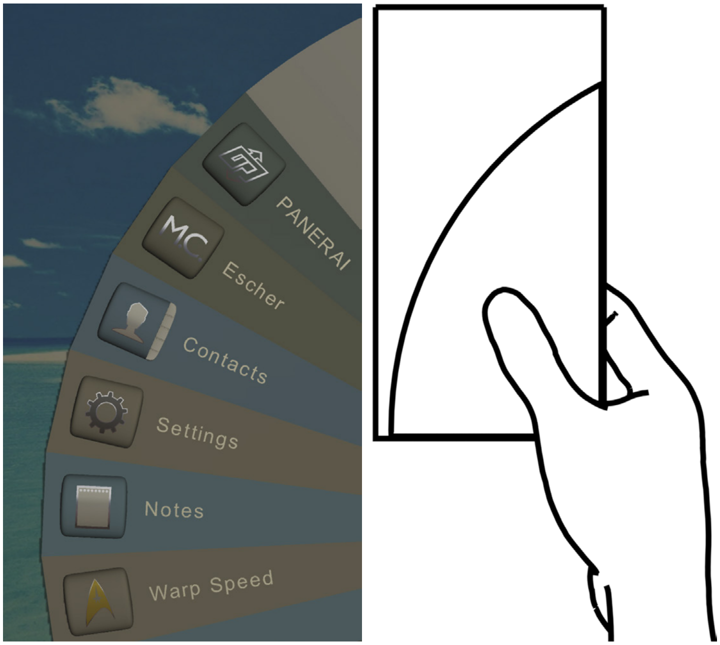


FIGURE 13.7

For mobile phones, menus can be designed so the options are easily reachable with the thumb. (Source: Bob Burrough; used with permission.)

STEERING LAW: MOVING POINTERS ALONG CONSTRAINED PATHS

A law derived from Fitts' law governs the time it takes to steer a screen-pointer along a constrained path to a target. Appropriately, it is named the *steering law* (Accot and Zhai, 1997). It says that if you must keep a pointer within a certain confined path while moving it to a target, then the wider the path, the faster you can move the pointer to the target (see Fig. 13.8). Its formula is simpler than that of Fitts' law:

$$T = a + b\ (D/W)$$

Like Fitts' law, the steering law seems like common sense: a wider path means you need not move the pointer carefully; you can move it ballistically—that is, fast.

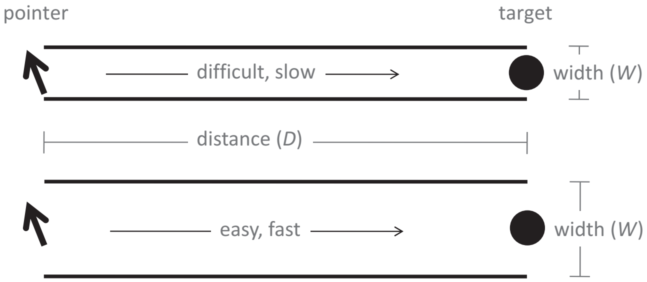


FIGURE 13.8

Steering law: pointing time depends on distance (D) and width (W) of path.

Design implications of the steering law

Anyone who has used devices that have point-and-click or touch-screen user interfaces can probably think of situations where they had to steer the screen-pointer or their finger along a constrained path. Those are situations in which the steering law applies. The following are two examples:

- Pull-right menus (also called "walking" menus) where you must keep the pointer within a menu item as you move it sideways to a submenu; otherwise, the menu switches to the item above or below. The narrower each menu item is, the slower the menu will be to use.
- Page rulers (e.g., for setting tabs) where you must keep the pointer within the ruler as you drag a tab to a new location; otherwise, the tab doesn't move (as in recent versions of Microsoft Word). The narrower the ruler, the slower your movement.

Pull-right menus are fairly common in software applications. For example, Apple's Safari browser uses them (see Fig. 13.9). Some software applications, such as DataTaker's DataLogger product (see Fig. 13.10), have pull-right menus going out to several levels.

To see how widening pointer-movement paths can speed the use of pull-right menus, compare the former and current websites of RoadScholar.org, a travel website aimed at older adult travelers. In mid-2012, a usability test of the website showed that people in the site's target age group had trouble choosing travel destinations using the site's pull-right menus (Finn and Johnson, 2013). By early 2013, the site's designers had widened the menu items significantly, making it easier and faster for users to choose travel destinations that interested them (see Fig. 13.11). That's the steering law at work.

Once upon a time, scrollbars in graphical user interfaces (GUIs) were constrained paths—you had to keep the pointer within the narrow vertical or horizontal scrollbar as you dragged the "elevator" along; otherwise, you lost control of the scrollbar. GUI designers soon realized that this made scrollbars slow, tedious, and error-prone

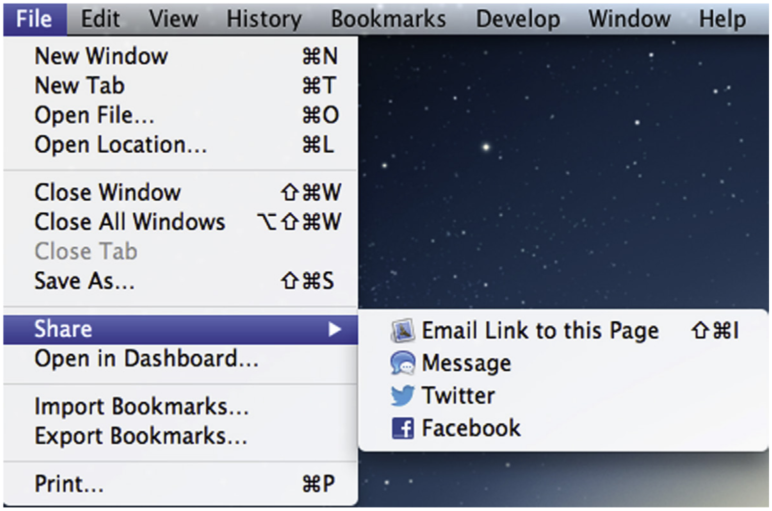


FIGURE 13.9

Pull-right menus in Apple's Safari browser.

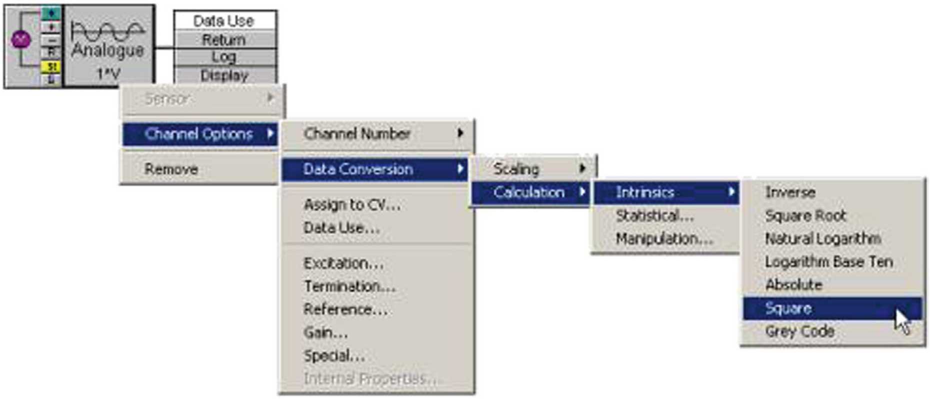


FIGURE 13.10

Pull-right menus in DataTaker's DataLogger application.

to use, so they eliminated the constraint. Modern scrollbars allow you to move the pointer outside of the bar while you drag the "elevator." They track motion in the direction of the scrollbar only and ignore any perpendicular motion. This change effectively made the constrained path as wide as the entire screen, greatly speeding scrollbar operation and eliminating a source of errors. Another case of the steering law at work.

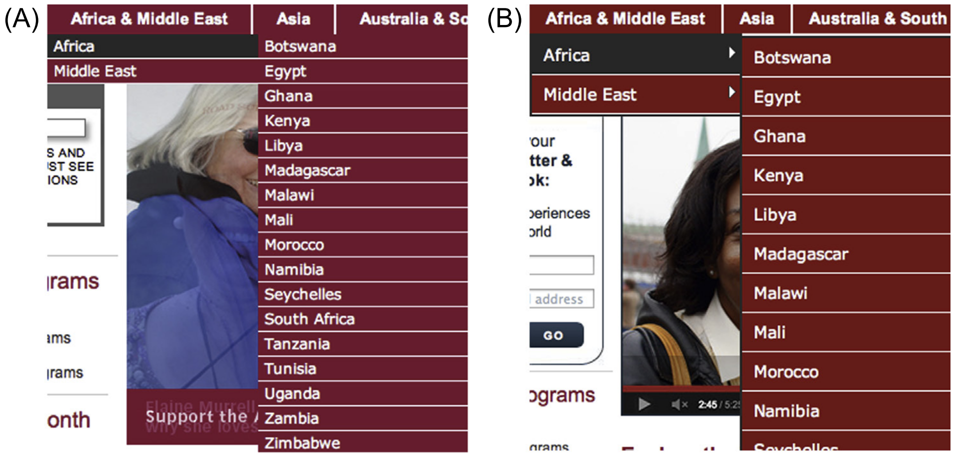


FIGURE 13.11

RoadScholar.org travel website: (A) in 2012, narrow menu items, and (B) in 2013, wider items.

IMPORTANT TAKEAWAYS

- The time it takes for a person to move any type of pointer to a specified target can be predicted by a formula called Fitts' law: $T = a + b \log_2 (1 + D/W)$. The law includes several components that influence hit-time:
  - D = distance to target: the greater, the longer the time.
  - W = width of target (in the direction of motion): the wider, the shorter the time. In plain language, bigger targets are faster to hit.
  - a and b are parameters for the start-lag and inertia of the pointer, respectively.
- Design implications of Fitts' law:
  - Make click-targets big.
  - Make the actual click-target at least as large as the visible click-target.
  - UI controls should accept clicks on their labels as well as on the visible controls.
  - Leave plenty of space between controls so people don't hit the wrong one.
  - Place important targets near the edge of the screen.
  - On smartphones, design menus so people can easily choose items with their thumb.
- The time it takes to move a pointer along a constrained path can be predicted by a formula called the steering law: $T = a + b\ (D/W)$. The main thing this law adds to Fitts' law is that the time to hit a target also depends on the ***width*** of the path in which a pointer has to stay while moving: the wider the path, the faster a user can move it to the target.
- Design implications of the steering law:
  - UI controls that require the user to keep the pointer in a narrow path while moving it will be slow and error-prone to use. Don't require users to move a pointer within a narrow area toward a target.