Lecture 3: Human Capabilities

UI Hall of Fame or Shame?

Source: Interface Hall of Shame

This message used to appear when you tried to delete the contents of your Internet Explorer cache from within the Windows Explorer.

Put aside the fact that the message is almost tautological ("Cookie… is a Cookie") and overexcited ("!!"). Does it give the user enough information to make a decision? What's a Cookie? What will happen if I delete it? Don't ask questions the user can't answer.

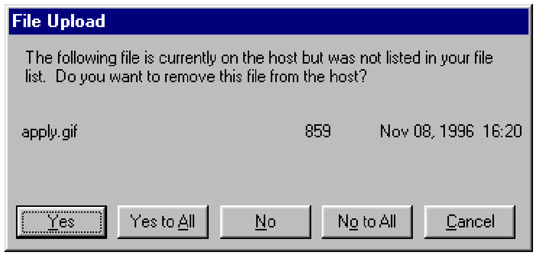
Hall of Shame

Source: Interface Hall of Shame

And definitely don't ask more than once. There may be hundreds of cookies cached in the browser; this dialog box appears for each one!

There's something missing from the dialog, whose absence becomes acute once the dialog appears a few times: a Cancel button. **Always give users a way to escape.**

Hall of Fame or Shame?



Source: Interface Hall of Shame

One way to fix the too-many-questions problem is Yes To All and No To All buttons, which short-circuit the rest of the questions by giving a blanket answer. That's a helpful **shortcut**, but this example shows that it's not a panacea.

This dialog is from Microsoft's Web Publishing Wizard, which uploads local files to a remote web site. Since the usual mode of operation in web publishing is to develop a complete copy of the web site locally, and then upload it to the web server all at once, the wizard suggests deleting files on the host that don't appear in the local files, since they may be orphans in the new version of the web site.

But what if you know there's a file on the host that you **don't** want to delete? You'd have to say No to every dialog until you found that file.

If your interface has a potentially large number of related questions to ask the user, it's much better to aggregate them into a single dialog. Provide a list of the files, and ask the user to select which ones should be deleted. Select All and Unselect All buttons would serve the role of Yes to All and No to All.

Today's Topics

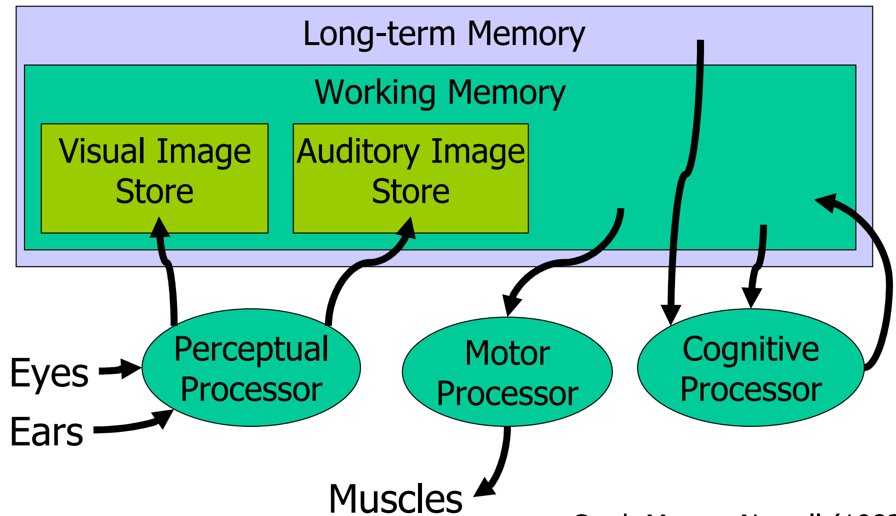
- Model human processor
- Perception
- Motor skills
- Memory
- Vision & color

This course is about building effective human-computer interfaces. Just as it helps to understand the properties of the computer system you're programming for – its processor speed, memory size, hard disk, operating system, and the interaction between these components – it's going to be important for us to understand some of the properties of the human that we're designing for.

We talked last time about user analysis, which collects information about high-level properties of our target users, particularly ways in which the target users are different from ourselves.

In today's lecture, we're going to look at low-level details: the processors, memories, and properties of the human cognitive apparatus. And we will largely concentrate on properties that most of us have in common (with some important exceptions when we look at color).

The Model Human Processor (MHP)



Card, Moran, Newell (1983)

The Model Human Processor was developed by Card, Moran, and Newell as a way to summarize decades of psychology research in an **engineering model**. It's a high-level look at the cognitive abilities of a human being -- really high level, like 30,000 feet. MHP is an abstraction, of course. But it's an abstraction that actually gives us *numerical parameters* describing how we behave. (Card, Moran, Newell, *The Psychology of Human-Computer Interaction*, Lawrence Erlbaum Associates, 1983)

Just as a computer has memory and processor, so does our model of a human. Actually, the MHP has several different kinds of memory, and several different processors.

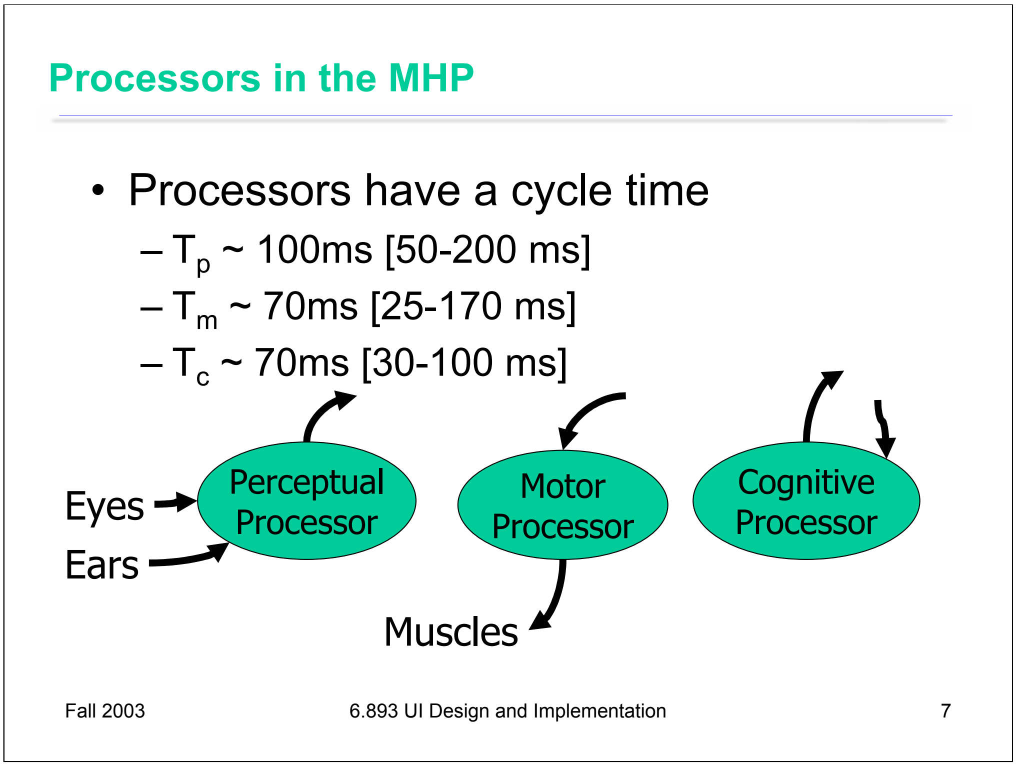
The **perceptual processor** takes input from the eyes and ears and drops it into two temporary memories, the **visual image store** and the **auditory image store**. As a computer hardware analogy, these memories are like frame buffers, storing a single frame of perception.

The **motor processor** takes instructions from the **working memory** (which you might think of as RAM, although it's pretty small), and runs those instructions on the muscles.

The **cognitive processor** operates on data from all the memories, including **long-term memory**, and puts its results back in the working memory.

Note that this model doesn't reflect the anatomy of your nervous system. There probably isn't an area in your brain corresponding to the perceptual processor. But it's a useful abstraction nevertheless. As an analogy, a computer's memory isn't physically an array of bytes, even though that's the abstraction we use to program it. Each byte is actually divided among several chips, 1 bit per chip, and different parts of the memory are on different DIMMs, and a couple of caches sit between the CPU and the RAM.

We'll look at each of these parts in more detail, starting with the processors.



The cycle time of an MHP processor is analogous to the cycle time of a computer processor. It's the time to accept one input and produce one output.

Like all parameters in the MHP, the cycle times shown above are derived from a survey of psychological studies. Each parameter is specified with a typical value and a range of reported values. For example, the typical cycle time for perceptual processor, T_p, is 100 milliseconds, but studies have reported between 50 and 200 milliseconds. The reason for the range is not only variance in individual humans; it is also varies with conditions. For example, the perceptual processor is faster (shorter cycle time) for more intense stimuli, and slower for weak stimuli. You can't read as fast in the dark. Similarly, your cognitive processor actually works faster under load! Consider how fast your mind works when you're driving or playing a video game, relative to sitting quietly and reading. The cognitive processor is also faster on practiced tasks.

It's reasonable, when we're making engineering decisions, to deal with this uncertainty by using all three numbers, not only the nominal value but also the range. Card, Moran, & Newell gave names to these three imaginary humans: Fastman, whose times are all as fast as possible; Slowman, whose times are all slow; and Middleman, whose times are all typical.

Perceptual Fusion

- Two stimuli within the same PP cycle (T_p ~ 100ms) appear **fused**
- Consequences
  - 1/ T_p frames/sec is enough to perceive a moving picture (10 fps OK, 20 fps smooth)
  - Computer response < T_p feels instantaneous
  - Causality is strongly influenced by fusion

One interesting effect of the perceptual processor is **perceptual fusion**. Here's an intuition for how fusion works. Every cycle, the perceptual processor grabs a frame (snaps a picture). Two events occurring less than the cycle time apart are likely to appear in the same frame. If the events are similar – e.g., Mickey Mouse appearing in one position, and then a short time later in another position – then the events tend to *fuse* into a single perceived event – a single Mickey Mouse, in motion.

Perceptual fusion is responsible for the way we perceive a sequence of movie frames as a moving picture, so the parameters of the perceptual processor give us a lower bound on the frame rate for believable animation. 10 frames per second is good for Middleman, but 20 frames per second is better for Fastman (remember that Fastman's T_p = 50 ms represents not just the quickest humans, but also the most favorable conditions).

Perceptual fusion also gives an upper bound on good computer response time. If a computer responds to a user's action within T_p time, its response feels instantaneous with the action itself. Systems with that kind of response time tend to feel like extensions of the user's body. If you used a text editor that took longer than T_p response time to display each keystroke, you would notice.

Fusion also strongly affects our perception of causality. If one event is closely followed by another – e.g., pressing a key and seeing a change in the screen – and the interval separating the events is less than T_p, then we are more inclined to believe that the first event caused the second.

Fitts's Law

- Open-loop control: T_m ~ 70 ms
- Closed-loop control: $T_p + T_c + T_m$ ~ 240 ms
- Moving your hand to a target is a sequence of corrections
  - Each cycle covers remaining distance D with error εD
- Fitt's Law
  - Time T to move your hand to a target of size S at distance D away is:
    - T = a + b log (D/S + 0.5)
  - Depends only on relative precision required D/S

Fitts's Law specifies how fast you can move your hand to a target of a certain size at a certain distance away (within arm's length, of course). It's a fundamental law of the human sensory-motor system, which has been replicated by numerous studies. Fitts's Law applies equally well to using a mouse to point at a target on a screen. We can explain Fitts's Law by appealing to the Model Human Processor.

The motor processor can operate in two ways. It can run autonomously, repeatedly issuing the same instructions to the muscles. This is "open-loop" control; the motor processor receives no feedback from the perceptual system about whether its instructions are correct. With open loop control, the maximum rate of operation is just T_m.

The other way is "closed-loop" control, which has a complete feedback loop. The perceptual system looks at what the motor processor did, and the cognitive system makes a decision about how to correct the movement, and then the motor system issues a new instruction. At best, the feedback loop needs one cycle of each processor to run, or $T_p + T_c + T_m$ ~ 240 ms. (Here's a simple but interesting experiment that you can try: take a sheet of lined paper and scribble a sawtooth wave back and forth between two lines, going as fast as you can but trying to hit the lines exactly on every peak and trough. Do it for 5 seconds. The frequency of the sawtooth carrier wave is dictated by open-loop control, so you can use it to derive your T_m. The frequency of the wave's **envelope**, the corrections you had to make to get your scribble back to the lines, is closed-loop control.)

Fitt's Law relies on closed-loop control. In each cycle, your motor system instructs your hand to move the entire remaining distance D. The accuracy of that motion is proportional to the distance moved, so your hand gets within some error εD of the target (possibly undershooting, possibly overshooting). Your perceptual and cognitive processors perceive where your hand arrived and compare it to the target, and then your motor system issues a correction to move the remaining distance εD – which it does, but again with proportional

Implications of Fitts's Law

- Targets at screen edge are easy to hit
  - Mac menubar beats Windows menubar
  - Unclickable margins are foolish
- Hierarchical menus are hard to hit
  - Gimp/GTK: instantly closes menu
  - Windows: .5 s timeout destroys causality
  - Mac does it right: triangular zone
- Linear popup menus vs. pie menus

Fitts's Law has some easy implications:

•The edge of the screen stops the mouse pointer, so you don't need a correcting cycle to hit it. Essentially, the edge of the screen acts like a target with *infinite* size. So edge-of-screen real estate is precious. The Macintosh menu bar, positioned at the top of the screen, is faster to use than a Windows menu bar (which, even when a window is maximized, is displaced by the title bar). Similarly, if you put controls at the edges of the screen, they should be active all the way to the edge to take advantage of this effect. Don't put an unclickable margin beside them.

•As we discussed last week, hierarchical submenus are hard to use, because of the correction cycles the user is forced to spend getting the mouse pointer carefully over into the submenu. Windows tries to solve this problem with a 500 ms timeout, and now we know another reason that this solution isn't ideal: it exceeds T_p (even for Slowman), so it destroys perceptual fusion and our sense of causality. Intentionally moving the mouse down to the next menu results in a noticeable delay. The Mac gets a Hall of Fame nod here, for doing it right with a triangular zone of activation for the submenu. The user can point straight to the submenu without unusual corrections, and without even noticing that there might be a problem. Hall of Fame interfaces are often invisible!

•Fitts's Law also explains why pie menus are faster to use than linear popup menus. With a pie menu, every menu item is a slice of a pie centered on the mouse pointer. As a result, each menu item is the same distance D away from the mouse pointer, and its size S is infinite (in the radial direction). Contrast that with a linear menu, where items further down the menu have larger D, and all items have a small S (height).

Power Law of Practice

- Time T_n to do a task the *n*th time is:

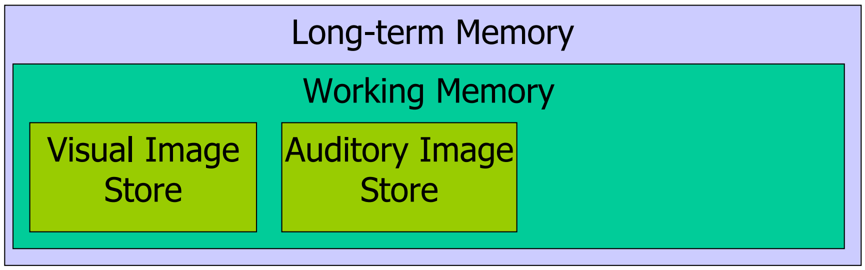
$$T_n = T_1 n^{-\alpha}$$

- α is typically 0.2-0.6

Another important feature of the perceptual-motor system is that the time to do a task decreases with practice. In particular, the time decreases according to the power law, shown above. The power law describes a linear curve on a log-log scale of time and number of trials.

In practice, the power law means that novices get rapidly better at a task with practice, but then their performance levels off to nearly flat (although still slowly improving).

Memories in the MHP



- Memory properties
  - Encoding: type of things stored
  - Size: number of things stored
  - Decay time: how long memory lasts

Now we turn to the other part of the Model Human Processor: the memories. MHP memories are characterized by three properties: encoding, size, and decay time.

Perceptual Stores

- Visual Image Store
  - encoded as physical image
  - size ~ 17 [7-17] letters
  - decay ~ 200 ms [70-1000 ms]
- Auditory Image Store
  - encoded as physical sound
  - size ~ 5 [4.4-6.2] letters
  - decay ~ 1500 ms [900-3500 ms]

The **visual image store** is basically an image frame from the eyes. It isn't encoded as pixels, but as physical features of the image, such as curvature, length, edges. It retains physical features like intensity that may be discarded in higher-level memories (like the working memory). We measure its size in letters because psych studies have used letters as a convenient stimulus for measuring the properties of the VIS; this doesn't mean that letters are represented symbolically in the VIS. The VIS memory is fleeting, decaying in a few hundred milliseconds.

The **auditory image store** is a buffer for physical sound. Its size is much smaller than the VIS (in terms of letters), but lasts longer – seconds, rather than tenths of a second.

Working Memory (WM)

- Small capacity: 7 ± 2 "chunks"
- Chunking depends on presentation and on what you already know
  - MWR CAA OLI BMF BIA TMD ODP HDI SPV CRB
  - BMW RCA AOL IBM FBI ATM DOD PHD ISP VCR
- Fast decay (7 [5-226] sec)
- **Maintenance rehearsal** fends off decay
- **Interference** causes faster decay

Working memory is where you do your conscious thinking. In terms of the MHP, working memory is where the cognitive processor gets its operands and drops its results. The currently favored model in cognitive science holds that working memory is not actually a separate place in the brain, but rather a pattern of **activation** of elements in the long-term memory.

The elements of working memory and long-term memory are called **chunks**. In one sense, chunks are defined symbols; in another sense, a chunk represents the activation of past experience. A famous result, due to George Miller (unrelated), is that the capacity of working memory is roughly 7 ± 2 chunks.

Our ability to form chunks in working memory depends strongly on how the information is presented – a sequence of individual letters tend to be chunked as letters, but a sequence of three-letter groups tend to be chunked as groups. It also depends on what we already know. If the three letter groups are well-known TLAs (three-letter acronyms) with well-established chunks in long-term memory, we are better able to retain them in working memory.

Working memory decays in tens of seconds. Maintenance rehearsal – repeating the items to yourself – fends off this decay, much as DRAM must refresh itself. But distraction destroys working memory. A particularly strong kind of distraction is **interference** – stimuli that activate several conflicting chunks are much harder to retain.

Say the Colors of These Words Aloud

- Book
- Pencil
- Slide
- Window
- Car
- Hat

Here's a little demonstration of interference. Say the **colors** of each of these words aloud, and time yourself.

Now Do It Again

- Green
- Orange
- Red
- Black
- Pink
- Blue

Now do it again – say the **colors** of each word aloud. It's harder, and most people do it much slower than the previous task. Why? Because the word, which names a different color, interferes with the color we're trying to say.

Long-term Memory (LTM)

- Huge capacity
- Little decay
- **Elaborative rehearsal** moves chunks from WM to LTM by making connections with other chunks

Long-term memory is probably the least understood part of the MHP. It contains the mass of our memories. Its capacity is huge, and it exhibits little decay. Long-term memories are apparently not intentionally erased; they just become inaccessible.

Maintenance rehearsal (repetition) appears to be useless for moving information into into long-term memory. Instead, the mechanism seems to be **elaborative rehearsal**, which seeks to make connections with existing chunks. Elaborative rehearsal lies behind the power of mnemonic techniques like associating things you need to remember with familiar places, like rooms in your childhood home.

The Eye

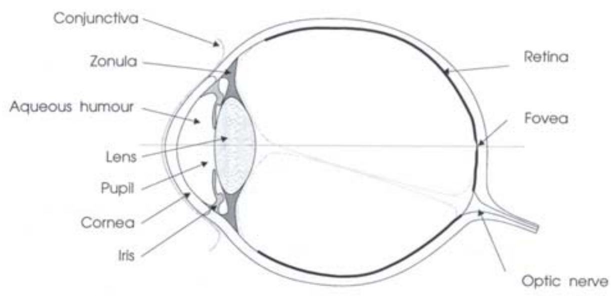


Figure 3: The human eye.

OK, we've looked at the perceptual system, the motor system, and memory. We'll conclude our discussion of the human machine today by considering the vision system in a little more detail, since vision is the primary way that a graphical user interface communicates to the user.

Here are key parts of the anatomy of the eye:

•The **cornea** is the transparent, curved membrane on the front of the eye.

•The **aqueous humor** fills the cavity between the cornea and the lens, and provides most of the optical power of the eye because of the large difference between its refractive index and the refractive index of the air outside the cornea.

•The **iris** is the colored part of the eye, which covers the lens. It is an opaque muscle, with a hole in the center called the **pupil** that lets light through to fall on the lens. The iris opens and closes the pupil depending on the intensity of light; it opens in dim light, and closes in bright light.

•The **lens** focuses light. Under muscle control, it can move forward and backward, and also get thinner or fatter to change its focal length.

•The **retina** is the surface of the inside of the eye, which is covered with light-sensitive receptor cells.

•The **fovea** is the spot where the optical axis (center of the lens) impinges on the retina. The highest density of photoreceptors can be found in the fovea; the fovea is the center of your visual field.

Figure taken from Lilley, Lin, Hewitt, & Howard, "Colour in Computer Graphics", University of Manchester.

Photoreceptors

- Rods
  - Only one kind (peak response in green wavelengths)
  - Sensitive to low light (“scotopic vision”)
    - Multiple nearby rods aggregated into a single nerve signal
  - Saturated at moderate light intensity (“photopic vision”)
    - Cones do most of the vision under photopic conditions
- Cones
  - Operate in brighter light
  - Three kinds: S(hort), M(edium), L(ong)
  - S cones are very weak, centered in blue wavelengths
  - M and L cones are more powerful, overlapping
  - M centered in green, L in yellow (but called “red”)

There are two kinds of photoreceptor cells in the retina. **Rods** operate under low-light conditions – night vision. There is only one kind of rod, with one frequency response curve centered in green wavelengths, so rods don’t provide color vision. Rods saturate at moderate intensities of light, so they contribute little to daytime vision. **Cones** respond only in brighter light. There are three kinds of cones, called S, M, and L after the centers of their wavelength peaks. S cones have very weak frequency response centered in blue. M and L cones are two orders of magnitude stronger, and their frequency response curves nearly overlap.

Signals from Photoreceptors

- Brightness

  M + L + rods
- Red-green difference

  L - M
- Blue-yellow difference

  weighted sum of S, M, L

The rods and cones do not send their signals directly to the visual cortex; instead, the signals are recombined into three channels. One channel is **brightness**, produced by the M and L cones and the rods. This is the only channel really active at night. The other two channels convey color **differences**. High responses mean red, and low responses indicate green.

These difference channels drive the theory of **opponent colors**: red and green are good contrasting colors because they drive the red-green channel to opposite extremes. Similarly, black/white and blue/yellow are good contrasting pairs.

Color Blindness

- Red-green color blindness (protonopia & deuteranopia)
  - 8% of males
  - 0.4% of females
- Blue-yellow color blindness (tritanopia)
  - Far more rare
- Guideline: don't depend solely on color distinctions
  - use redundant signals: brightness, location, shape

Color deficiency ("color blindness") affects a significant fraction of human beings. An overwhelming number of them are male.

There are three kinds of color deficiency, which we can understand better now that we understand a little about the eye's anatomy:

•**Protonopia** is missing or bad L cones. The consequence is reduced sensitivity to red-green differences (the L-M channel is weaker), and reds are perceived as darker than normal.

•**Deuteranopia** is caused by missing or malfunctioning M cones. Red-green difference sensitivity is reduced, but reds do not appear darker.

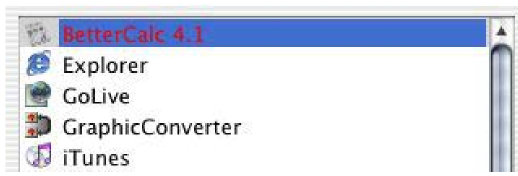
•**Tritanopia** is caused by missing or malfunctioning S cones, and results in blue-yellow insensitivity.

Since color blindness affects so many people, it is essential to take it into account when you are deciding how to use color in a user interface. Don't depend solely on color distinctions, particularly red-green distinctions, for conveying information. Microsoft Office applications fail in this respect: red wavy underlines indicate spelling errors, while identical green wavy underlines indicate grammar errors.

Traffic lights are another source of problems. How do red-green color-blind people know whether the light is green or red? Fortunately, there's a spatial cue: red is always above (or to the right of) green. Protonopia sufferers (as opposed to deuteranopians) have an additional advantage: the red light looks darker than the green light.

Chromatic Aberration

- Different wavelengths focus differently
  - Highly separated wavelengths (red & blue) can't be focused simultaneously
- Guideline: don't use red-on-blue text
  - It looks fuzzy and hurts to read



The refractive index of the lens varies with the wavelength of the light passing through it; just like a prism, different wavelengths are bent at different angles. So your eye needs to focus differently on red features than it does on blue features.

As a result, an edge between widely-separated wavelengths – like blue and red – simply can't be focused. It always looks a little fuzzy. So blue-on-red or red-on-blue text is painful to read, and should be avoided at all costs.

Apple's ForceQuit tool in Mac OS X, which allows users to shut down misbehaving applications, unfortunately falls into this trap. In its dialog, unresponding applications are helpfully displayed in red. But the selection is a blue highlight. The result is incredibly hard to read.

Blue Details Are Hard to Resolve

- Fovea has no S cones
  - Can't resolve small blue features (unless they have high contrast)
- Lens and aqueous humor turn yellow with age
  - Blue wavelengths are filtered out
- Lens weakens with age
  - Blue is harder to focus
- Guideline: don't use blue where small details matter (text!)

A number of anatomical details conspire to make blue a bad color choice when small details matter.

First, the fovea has very few S cones, so you can't easily see blue features in the center of your vision (unless they have high contrast with the background, activating the M and L cones).

Second, older eyes are far less sensitive to blue, because the lens and aqueous humor slowly grow yellower, filtering out the blue wavelengths.

Finally, the lens gets weaker with age. Blue is at one extreme of its focusing range, so older eyes can't focus blue features as well.

As a result, avoid blue text, particularly small blue text.

Fovea Has No Rods

- Rods are more sensitive to dim light
- In scotopic conditions, peripheral vision (rod-rich) is better than foveal vision
  - Easier to see a dim star if you don't look directly at it

Incidentally, the fovea has no rods, either. That explains why it's easier to see a dim star if you don't look beside it, rather than directly at it.

Reading for Next Time

- Two Norman papers about errors