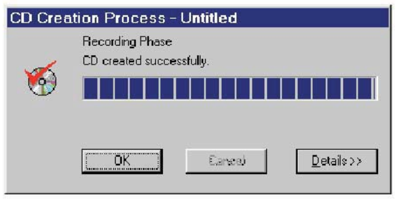
Lecture 20: Color Design and Typography

PS3, RS3 due Sunday

GR4 due one week from Sunday

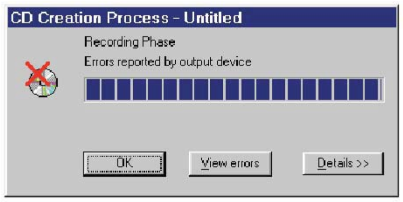
UI Hall of Fame or Shame?


CD Creation Process - Untitled

Recording Phase

CD created successfully.

Source: UI Hall of Shame


CD Creation Process - Untitled

Recording Phase

Errors reported by output device

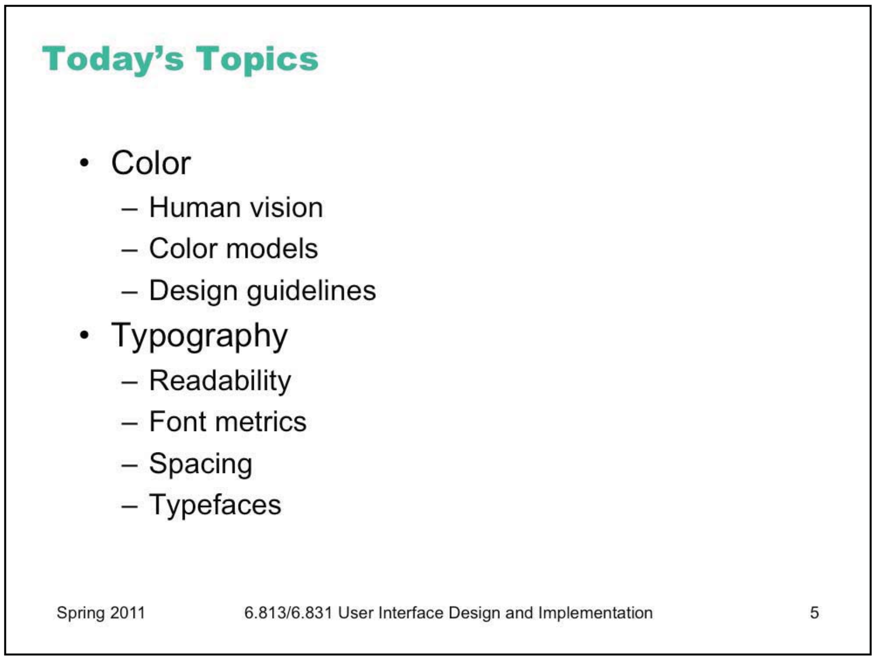
Our Hall of Shame candidate for the day is this dialog box from Adaptec Easy CD Creator, which appears at the end of burning a CD. The top image shows the dialog when the CD was burned successfully; the bottom image shows what it looks like when there was an error.

What does the squint test tell you about these dialogs?

The key problem is the **lack of contrast** between these two states. Success or failure of CD burning is important enough to the user that it should be obvious at a glance. But these two dialogs look identical at a glance. How can we tell? Use the **squint test**, which we talked about in the graphic design lecture. When you're squinting, you see some labels, a big filled progress bar, a roundish icon with a blob of red, and three buttons. All the details, particularly the text of the messages and the exact shapes of the red blob, are fuzzed out. This simulates what a user would see at a quick glance, and it shows that the graphic design doesn't convey the contrast.

One improvement would change the check mark to another color, say green or black. Using red for OK seems **inconsistent** with the real world, anyway. But designs that differ only in red and green wouldn't pass the squint test for color-blind users.

Another improvement might remove the completed progress bar from the error dialog, perhaps replacing it with a big white text box containing a more detailed description of the problem. That would clearly pass the squint test, and make errors much more noticeable.



Today's Topics

- Color
  - Human vision
  - Color models
  - Design guidelines
- Typography
  - Readability
  - Font metrics
  - Spacing
  - Typefaces

Spring 2011 6.813/6.831 User Interface Design and Implementation 5

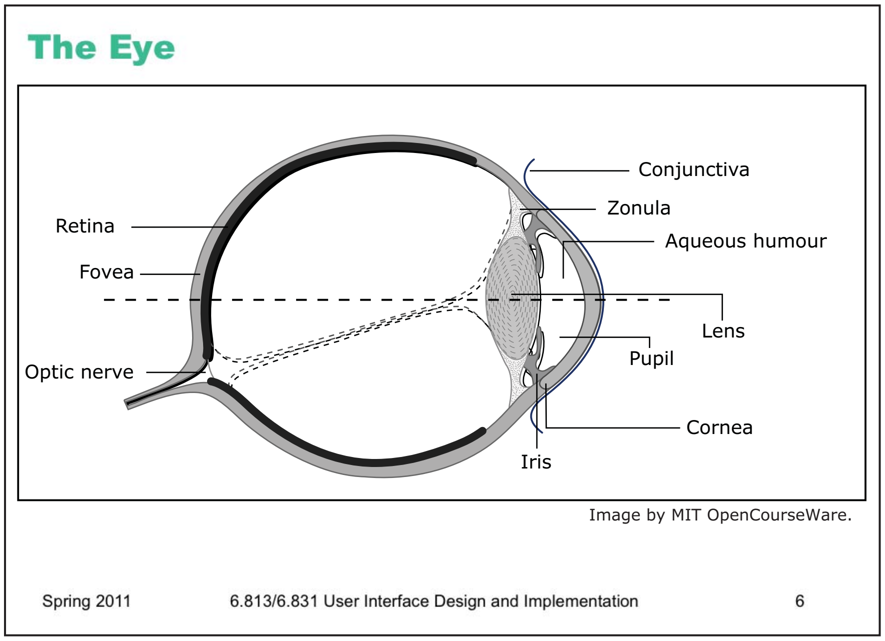
Today's lecture is about choosing colors for a user interface. We'll discuss some of the properties of **human vision** that affect this decision, particularly the limitations of color vision. We'll go over some **models** for representing colors, not just the familiar RGB model. And we'll discuss some guidelines for choosing colors. The most important guidelines will be applications of rules we already discussed in graphic design: **simplicity** as much as possible, **contrast** where important.

A good reference about color is Colin Ware, *Information Visualization: Perception for Design*, Morgan Kaufmann, 2000 (which we also mentioned in the graphic design lecture). For typography, an outstanding book is Robert Bringhurst, *The Elements of Typographic Style*, Hartley & Marks, 2002. Also useful is "Principles of Typography for User Interface Design" by Paul Kahn and Krzysztof Lenk, *interactions*, which you can find online.

An aside about why we're learning this. This lecture is effectively part 2 of the graphic design lecture, and many of the guidelines in both lectures are more about aesthetics than pure usability. Serious *mistakes* in graphic design certainly affect usability, however, so we're trying to help you avoid those pitfalls. But there's a larger question here: in practice, should software engineers have to learn this stuff at all? Shouldn't you just hire a graphic designer and let them do it? Some people who teach UI to CS students think that the most important lesson a software engineer can learn from a course like this is "UI design is hard; leave it to the experts." They argue that a little knowledge can be a dangerous thing, and that a programmer with a little experience in UI design but too much self-confidence can be just as dangerous as an artist who's learned a little bit of HTML and thinks they now know how to program. But I prefer to believe that a little knowledge is a step on the road to greater knowledge. Some of you may decide to *become* UI designers, and this course is a step along that road.

In a commercial environment, you *should* hire experienced graphic designers, just as you should hire an architect for building your corporation's headquarters and you should contract with a licensed building firm. Big jobs for big bucks require experts who have focused their education and job experience on those problem. One reason this course is useful is that you can appreciate what UI experts do and evaluate their work, which will help you work on a team with them (or supervise them).

But it's also worth learning these principles because you can apply them yourself on smaller-scale problems. Are you going to hire a graphic designer for every PowerPoint presentation you make, every chart you draw, every web page you create? Those are all user interfaces. Many problems in life have a user interface, and many of them are up to you to do-it-yourself. So you should know when to leave it to the experts, but you should be able to do a creditable job yourself too, when the job is yours to do.



The Eye

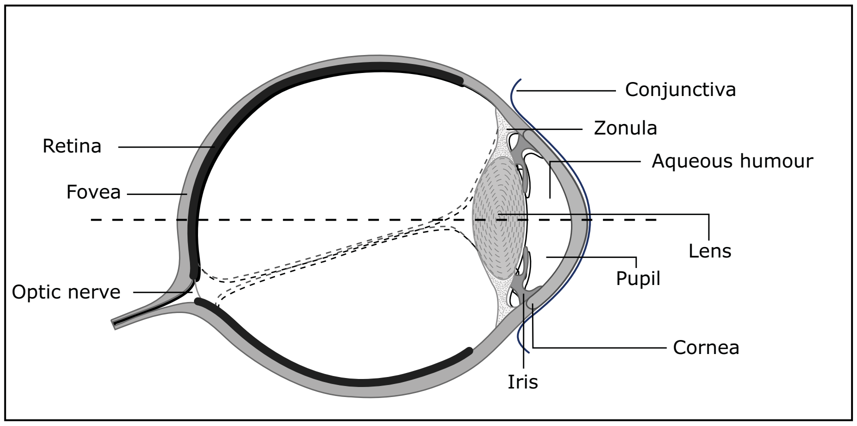


Image by MIT OpenCourseWare.

Spring 2011 6.813/6.831 User Interface Design and Implementation 6

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.

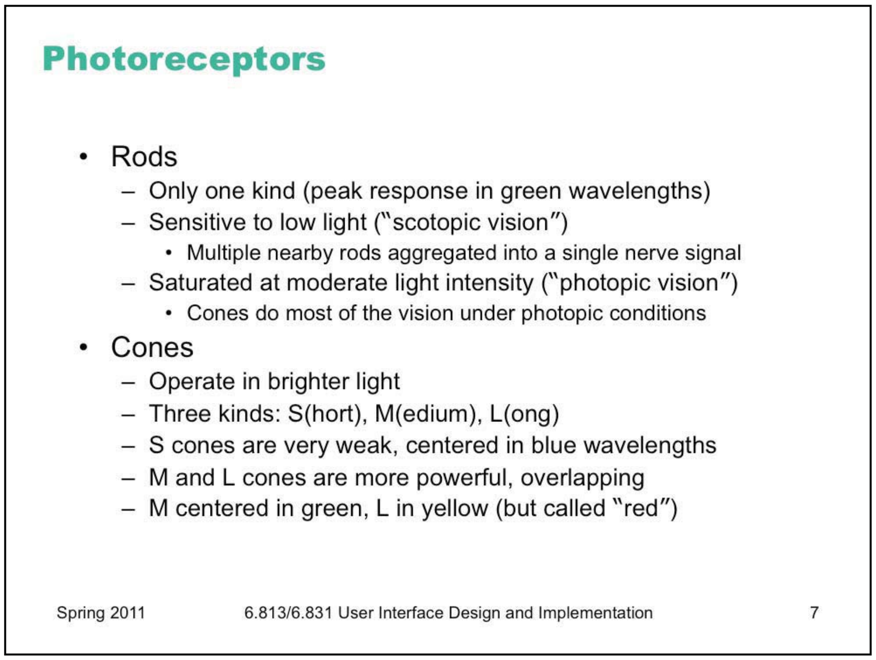
•The **iris** is the colored part of the eye, which covers the lens. It is an opaque muscle, with a hole in the center called the **pupil** that lets light through to fall on the lens. The iris opens and closes the pupil depending on the intensity of light; it opens in dim light, and closes in bright 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 from Lilley, Lin, Hewitt, & Howard, "Colour in Computer Graphics", University of Manchester.

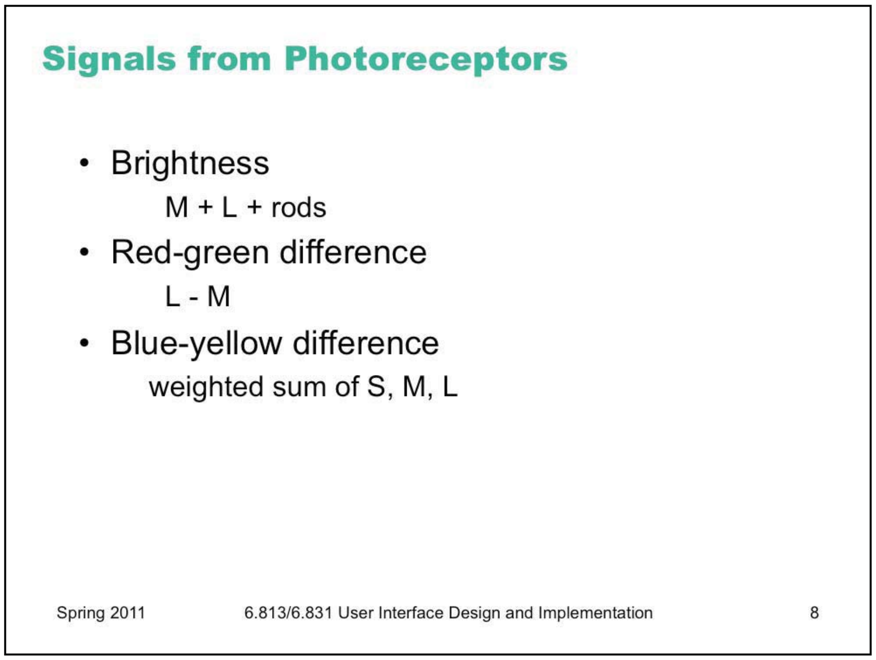


Photoreceptors

- 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")

Spring 2011 6.813/6.831 User Interface Design and Implementation 7

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.



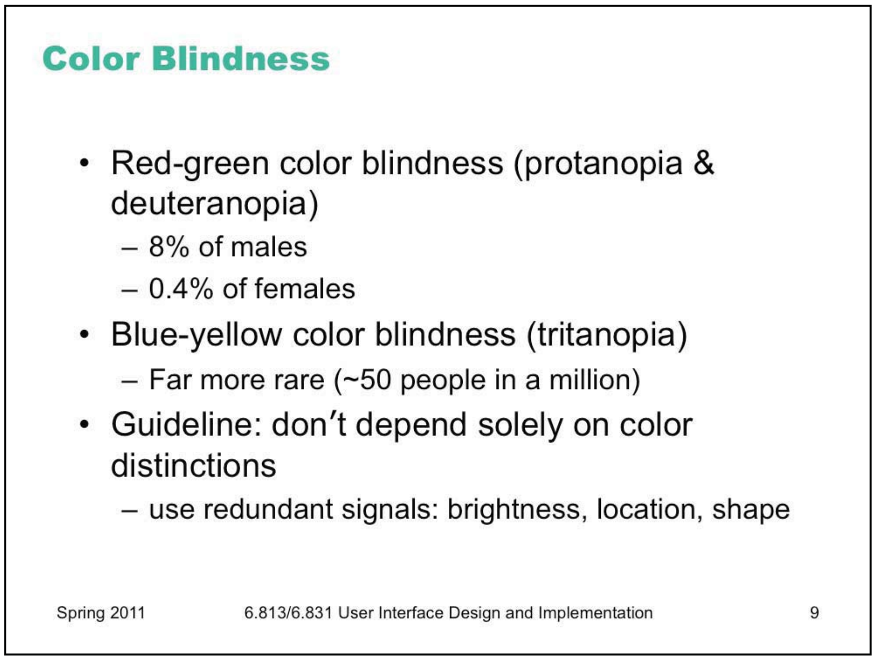
Signals from Photoreceptors

- Brightness
  - M + L + rods
- Red-green difference
  - L - M
- Blue-yellow difference
  - weighted sum of S, M, L

Spring 2011 6.813/6.831 User Interface Design and Implementation 8

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**, red-green and blue-yellow. For the red-green channel, for example, 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 (protanopia & deuteranopia)
  - 8% of males
  - 0.4% of females
- Blue-yellow color blindness (tritanopia)
  - Far more rare (~50 people in a million)
- Guideline: don't depend solely on color distinctions
  - use redundant signals: brightness, location, shape

Spring 2011 6.813/6.831 User Interface Design and Implementation 9

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:

•**Protanopia** 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.

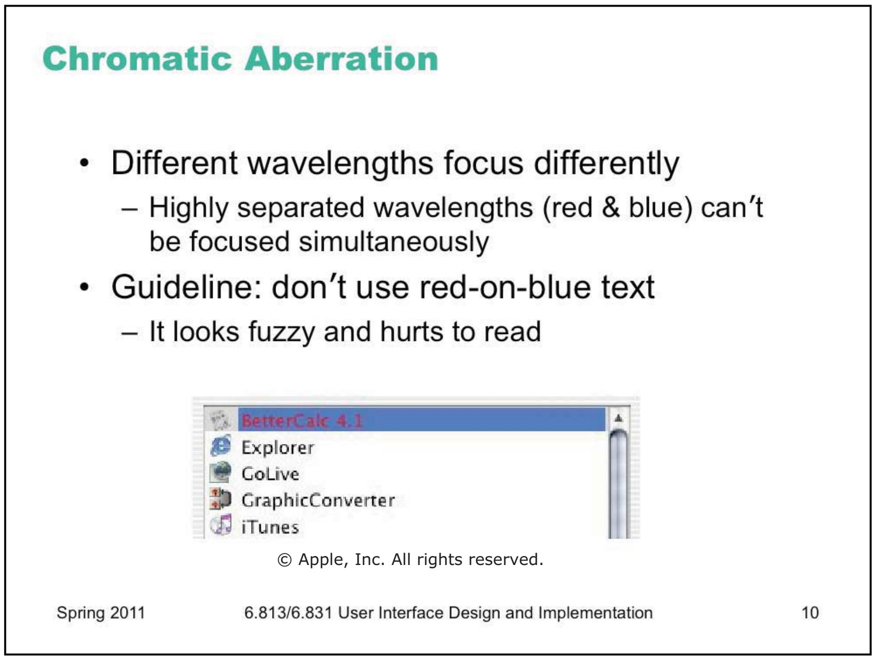
Red/green color blindness affects about 8% of males and 0.4% of females; blue/yellow color blindness is much much rarer.

But 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. Protanopia sufferers (as opposed to deuteranopians) have an additional advantage: the red light looks darker than the green light.

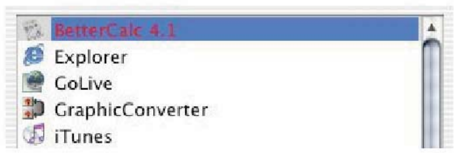
There are online tools for checking your interface against various kinds of color blindness; one good one is Vischeck (http://www.vischeck.com/vischeck/).

Henry Sturman is a red-green colorblind software developer who has written a good article about what it's like (http://henrysturman.com/english/articles/colorvision.html).



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

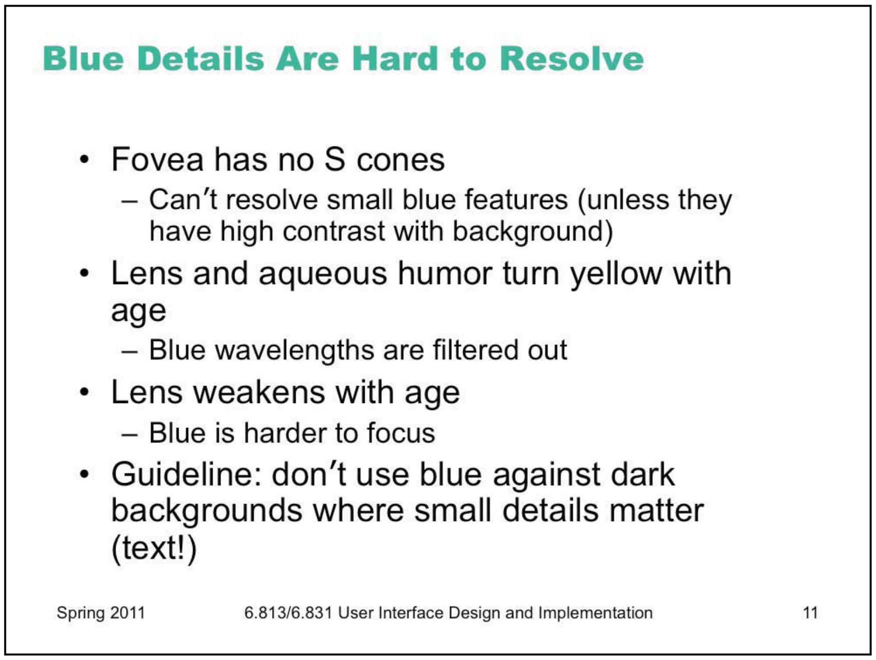


The refractive index of the lens varies with the wavelength of the light passing through it; just like a prism, different wavelengths are bent at different angles. So your eye needs to focus differently on red features than it does on blue features.

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 fell into this trap. In the dialog, unresponding applications are helpfully displayed in red. But the selection is a blue highlight. The result is incredibly hard to read.

Here's an experiment you can try to demonstrate chromatic aberration. Put a small purple dot on a piece of paper. Hold it close to your eye; it should look blue in the center, surrounded by a red halo. Then move it farther away; the colors should switch places, so that red is in the center and blue is the halo.



Blue Details Are Hard to Resolve

- Fovea has no S cones
  - Can′t resolve small blue features (unless they have high contrast with background)
- 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 against dark backgrounds where small details matter (text!)

Spring 2011 6.813/6.831 User Interface Design and Implementation 11

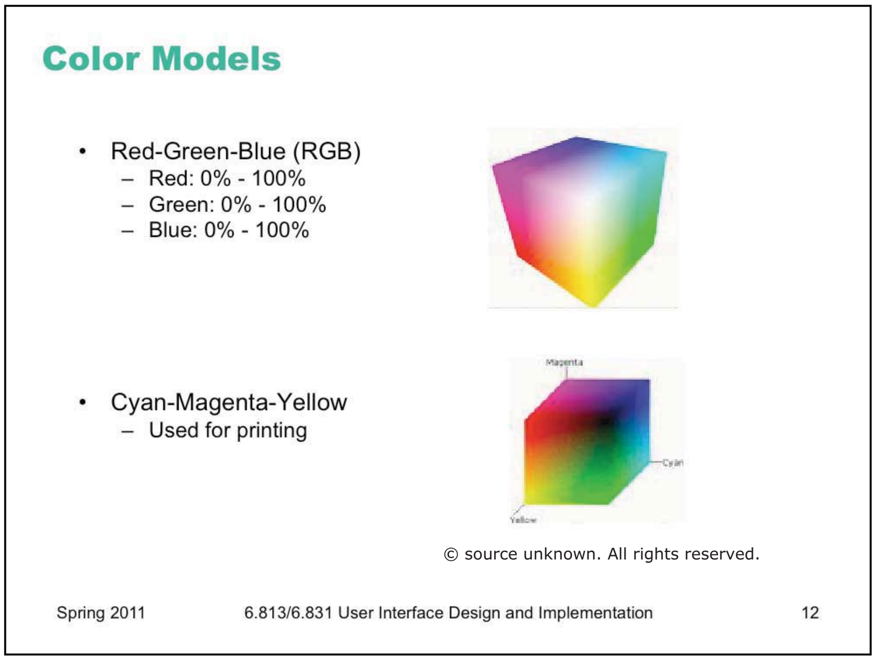
A number of anatomical details conspire to make blue a bad color choice when small details matter.

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.

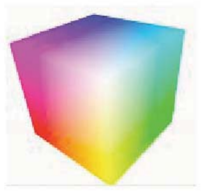
Finally, the lens gets weaker with age. Blue is at one extreme of its focusing range, so older eyes can't focus blue features as well.

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

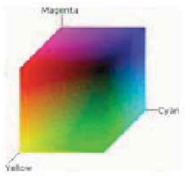


Color Models

- Red-Green-Blue (RGB)
  - Red: 0% - 100%
  - Green: 0% - 100%
  - Blue: 0% - 100%



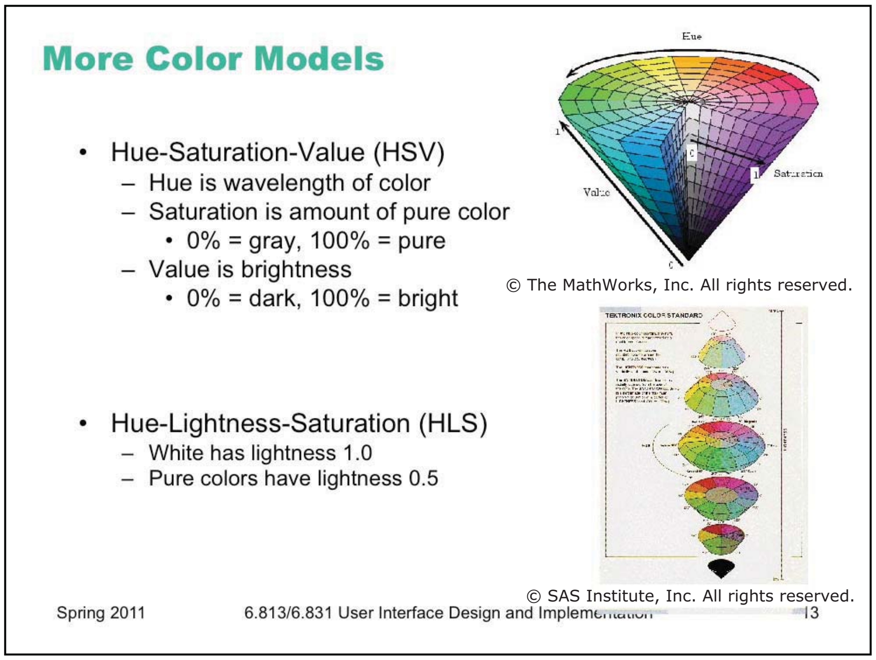
- Cyan-Magenta-Yellow
  - Used for printing



Spring 2011 6.813/6.831 User Interface Design and Implementation 12

Now let's look at how colors are represented in GUI software. At the lowest level, the RGB model rules. The RGB model is a unit cube, with (0,0,0) corresponding to black, (1, 1, 1) corresponding to white, and the three dimensions measuring levels of red, green, and blue. The RGB model is used directly by CRT and LCD monitors for display, since each pixel in a monitor has separate red, green, and blue components.

The CMYK (cyan, magenta, yellow, and sometimes black) is similar to the RGB model, but used for print colors, where pigments absorb wavelengths instead of generating them.


Hue
Saturation
Value
TEKTRONIX COLOR STANDARD

More Color Models

- Hue-Saturation-Value (HSV)
  - Hue is wavelength of color
  - Saturation is amount of pure color
    - 0% = gray, 100% = pure
  - Value is brightness
    - 0% = dark, 100% = bright
- Hue-Lightness-Saturation (HLS)
  - White has lightness 1.0
  - Pure colors have lightness 0.5

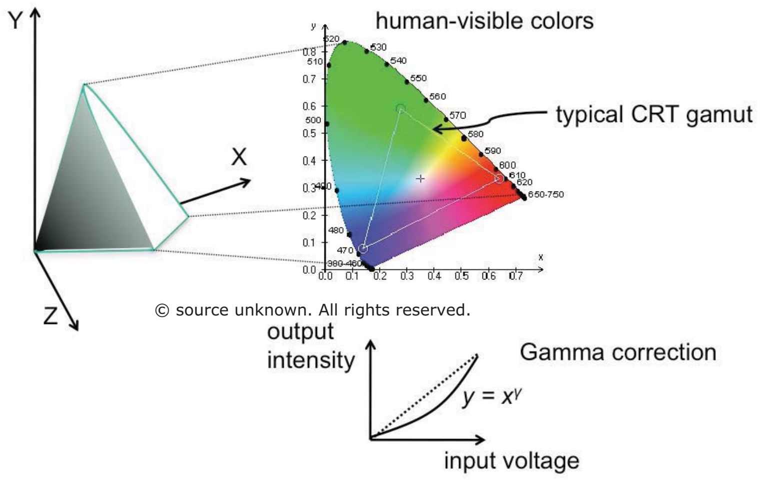
Spring 2011 6.813/6.831 User Interface Design and Implementation 13

HSV (hue, saturation value) is a better model for how humans perceive color, and more useful for choosing colors in user interface design. HSV is a cone. We've already encountered hue and value in our discussion of visual variables. Saturation is the degree of color, as opposed to grayness. Colors with zero saturation are shades of gray; colors with 100% saturation are pure colors.

HLS (hue, lightness, saturation) is a symmetrical relative of the HSV model, which is elegant. It basically pulls up the center of the HSV cone to make a double-ended cone.

Many applications have a color picker that lets you pick HSV values as an alternative to RGB.

Obtaining Accurate Color



Although RGB and HSV are commonly used for implementing graphical user interfaces, they are not in fact *standardized*. The way that a color like pure red (RGB=1,0,0) actually looks depends strongly on the display's characteristics, so your application can't be sure it will get exactly the right color.

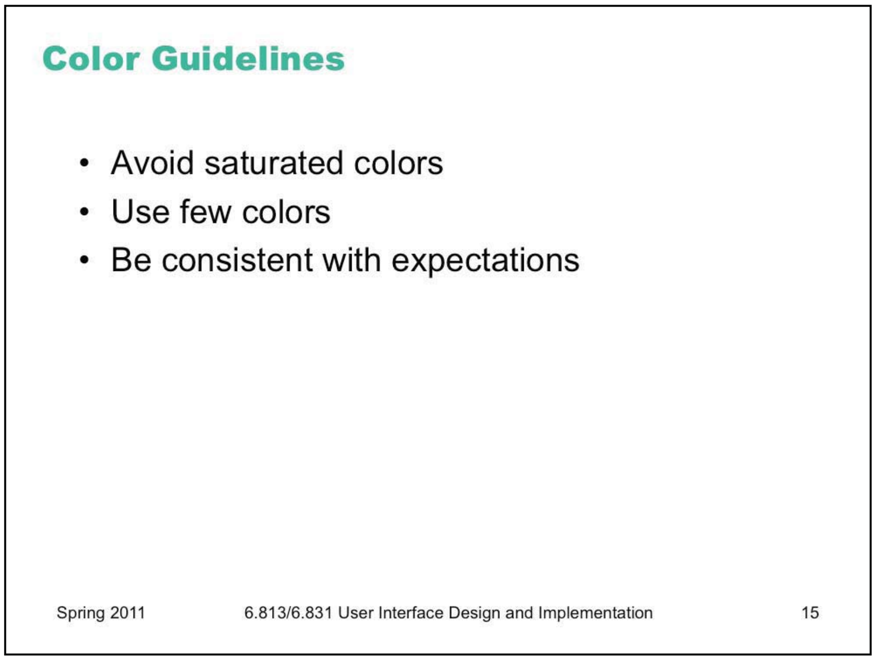
The science of colorimetry is concerned with accurate measurement and reproduction of color. Most of it is outside the scope of this course, but here are a few things you should know. Colorimetry starts with a 3D space based on three primary colors, called XYZ, chosen so that all human-visible colors are bounded within the positive octant -- so that any visible color can be made as a mix of positive amounts of X, Y, and Z. The solid area on the left shows the visible colors perceivable to the human eye; black is of course at the origin. Note that X, Y, and Z are *imaginary* colors in the sense that they cannot be produced by a physical light source or perceived by the human eye; but they're useful bases for the space, much like imaginary numbers are useful. To consider hue and saturation in isolation, we look at a plane of constant intensity, shown on the right.

The wedge-shaped figure shows the whole space of human-perceivable colors (with fully-saturated, pure-wavelength colors around its perimeter). Any given display device can produce some triangle of colors on this plane (called the device's **gamut**), where the corners are the three colors used by the device as its primary colors – e.g., the exact colors of red, green, and blue in a CRT's phosphors. The triangle here shows a typical cathode-ray television's gamut. Devices with different gamuts will produce different colors from the same RGB value. This problem can be addressed by calibrating the display device, producing an ICC profile that specifies how the device's RGB space maps into a standardized space like XYZ.

Other standardized color spaces exist. One drawback of XYZ is that it's not perceptually uniform – green occupies a huge chunk at the top of the wedge, while yellow is a narrow little line; it would be preferable if the distance in color space produced the same difference in perception in both areas. LUV is an alternative model that addresses that by distorting the projection. But neither XYZ or LUV is particularly useful for programming because they're imaginary colors; sRGB aims to fix that by standardizing the RGB color space instead (http://www.w3.org/Graphics/Color/sRGB.html).

Another issue in accurate color reproduction is the intensity (value) of the color. Different display devices have different response curves. When the red component of an RGB value (0-1) is mapped directly to a voltage applied to an electron gun, the intensity of light produced does not vary linearly from 0 to 1, but typically follows a power curve (y=x^gamma, for some gamma > 1). **Gamma correction** is the process of standardizing the intensity so that a linear response is obtained.

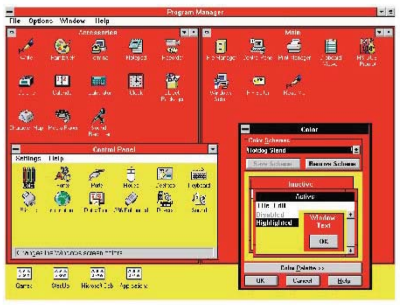
All these issues also apply to cameras as well as displays, of course. Cameras have gamuts (for the hues and saturations they can record) and response curves (to intensity) that require calibration and correction. To learn more about human perception and color, take 6.098/6.882 Computational Photography.



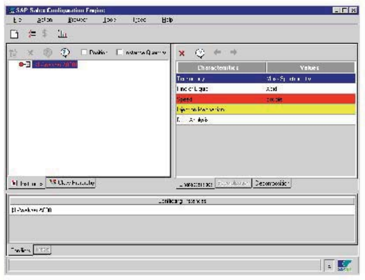
Color Guidelines

- Avoid saturated colors
- Use few colors
- Be consistent with expectations

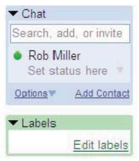
Avoid Saturated Colors



© source unknown. All rights reserved.



© SAP AG. All rights reserved.



© Google. All rights reserved.

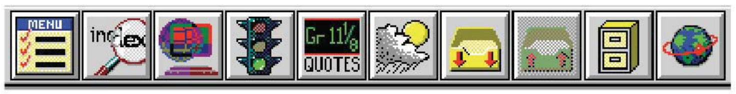
Spring 2011 6.813/6.831 User Interface Design and Implementation 16

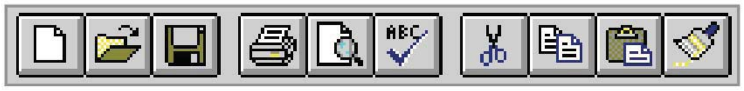
In general, avoid strongly saturated colors – i.e., the colors around the outside edge of the HSV cone. Saturated colors can cause visual fatigue because the eye must keep refocusing on different wavelengths. They also tend to saturate the viewer's receptors (hence the name). One study found that air traffic controllers who viewed strongly saturated green text on their ATC interfaces for many hours had trouble seeing pink or red (the other end of the red/green color channel) for up to 15 minutes after their shift was over.

Use less saturated, "pastel" colors instead, which mix gray or white into the pure color.

The examples on top use colors with high saturation; on the bottom, low saturation. Shades of gray have minimum saturation.

Use Few Colors



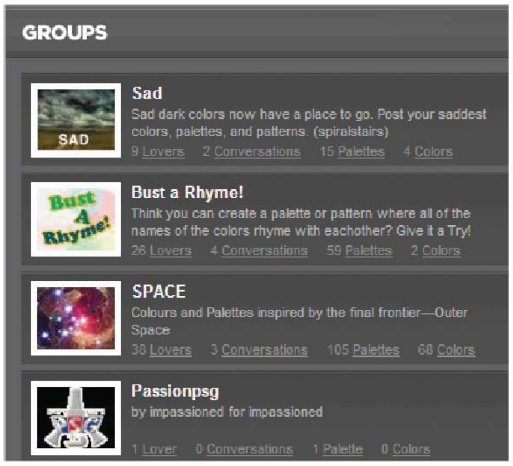


In general, colors should be used sparingly. An interface with many colors appears more complex, more cluttered, and more distracting. Use only a small number of different hues.

The toolbar on top uses too many colors (many of them highly saturated), so none of the buttons stand out, and the toolbar feels hard to scan. In contrast, the toolbar at the bottom uses only a handful of colors. It's more restful to look at, and the buttons that actually use color (like the Open File button) really pop out.

A simple and very effective color scheme uses just one hue (like blue or green, weakly saturated and in various values), combined with black, white, and shades of gray. On top of a scheme like that, a bit of red in an icon will pop out wonderfully.

Background Colors



GROUPS

Sad
Sad dark colors now have a place to go. Post your saddest colors, palettes, and patterns. (spiralstairs)
9 Lovers 2 Conversations 15 Palettes 4 Colors

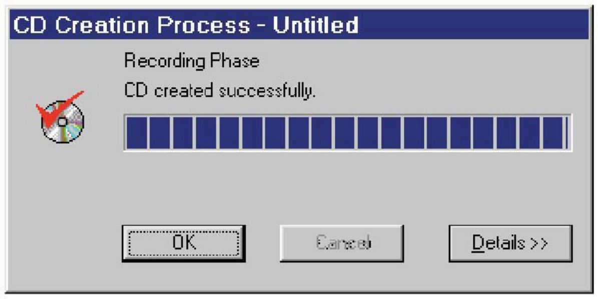
Bust a Rhyme!
Think you can create a palette or pattern where all of the names of the colors rhyme with eachother? Give it a Try!
26 Lovers 4 Conversations 59 Palettes 2 Colors

SPACE
Colours and Palettes inspired by the final frontier—Outer Space
38 Lovers 3 Conversations 105 Palettes 68 Colors

Passionpsg
by impassioned for impassioned
1 Lover 0 Conversations 1 Palette 0 Colors

Background colors should establish a good contrast with the foreground. White is a good choice, since it provides the most contrast; but it also produces bright displays, since our computer displays emit light rather than reflecting it. Pale (desaturated) yellow and very light gray are also good background colors. Dark backgrounds are tricky; it's too easy to mess up the contrast and make text less legible, as shown in this example.

Be Consistent With Expectations

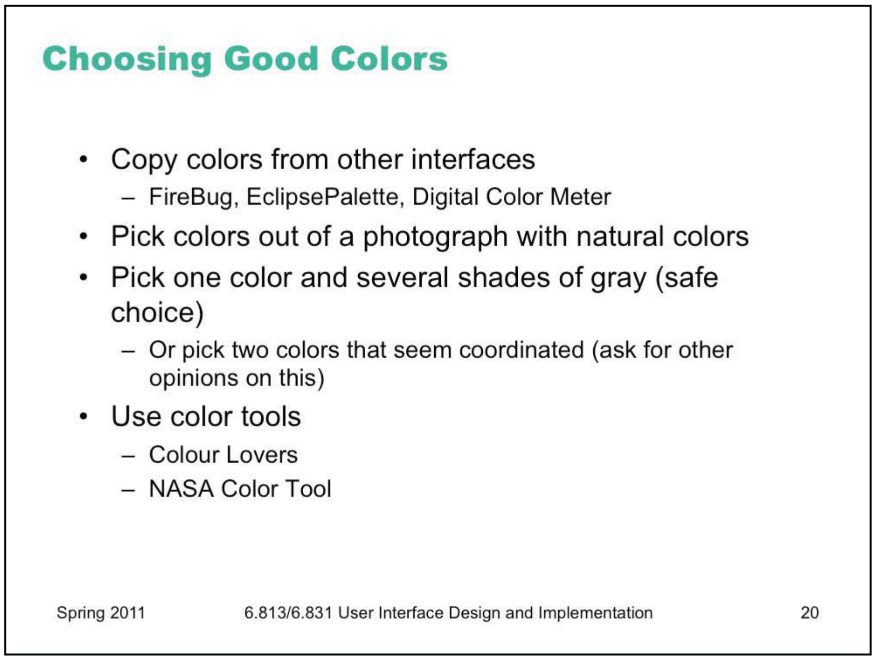

CD Creation Process - Untitled

Recording Phase

CD created successfully.

Finally, match expectations. One of the problems with the Adaptec dialogs at the beginning of this lecture was the use of red for OK. Red generally means stop, warning, error, or hot. Green conventionally means go, or OK. Yellow means caution, or slow.

(But note that these conventional meanings for colors are culturally dependent, and what works in Western cultures may not work for all users.)



Choosing Good Colors

- Copy colors from other interfaces
  - FireBug, EclipsePalette, Digital Color Meter
- Pick colors out of a photograph with natural colors
- Pick one color and several shades of gray (safe choice)
  - Or pick two colors that seem coordinated (ask for other opinions on this)
- Use color tools
  - Colour Lovers
  - NASA Color Tool

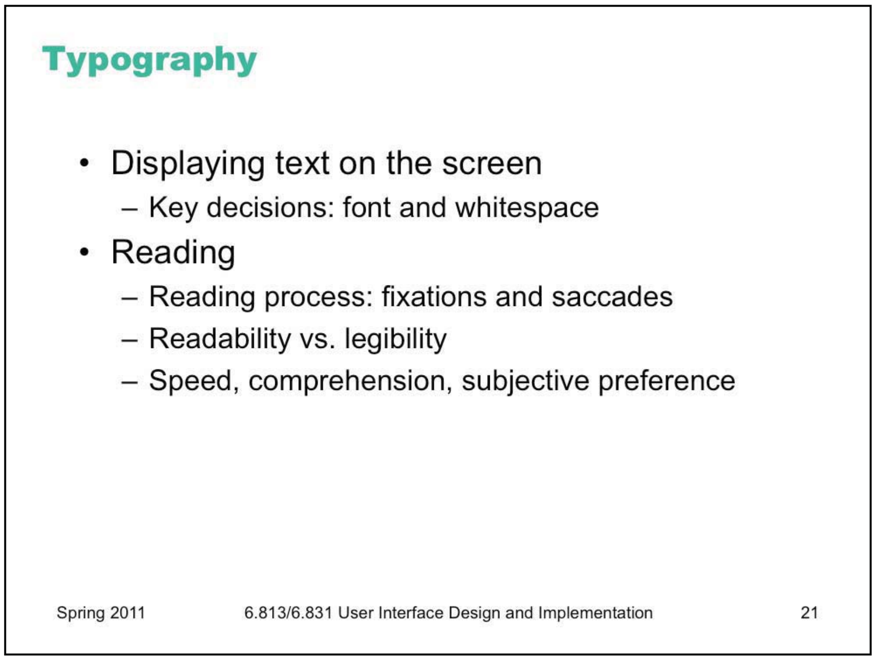
Spring 2011 6.813/6.831 User Interface Design and Implementation 20

Given all these rules about what colors *not* to choose, what colors *should* you choose? There are no hard-and-fast rules here, but there are a few heuristics. The first heuristic is an old standby – use color schemes that seem to work well for other interfaces on the desktop or the web. There are several tools you can use to probe your web browser (Firebug for Firefox) or desktop screen (EclipsePalette for Windows, Digital Color Meter for Mac) to determine what color is being used by a particular display element.

Another effective heuristic is to find a photograph of a natural scene that looks appealing to you, and extract colors from it (using the same tools, or using the eyedropper tool in a paint program). The intuitive basis for this heuristic is that our visual systems evolved to easily perceive and appreciate the natural world.

Keep your choices simple. You can't go far wrong by choosing one weakly saturated color and a few shades of gray. As soon as you choose two colors, however, you run the risks of an aesthetic clash between them; it's good to get some other opinions on your choice, particularly if you might be somewhat colorblind yourself.

There are also some sites out there that help you choose colors. Colour Lovers (http://www.colourlovers.com/) is a large collection of user-contributed color schemes, with ratings and votes. The NASA Color Tool helps select a palette of colors using HLS and view them side-by-side on sample data.



Typography

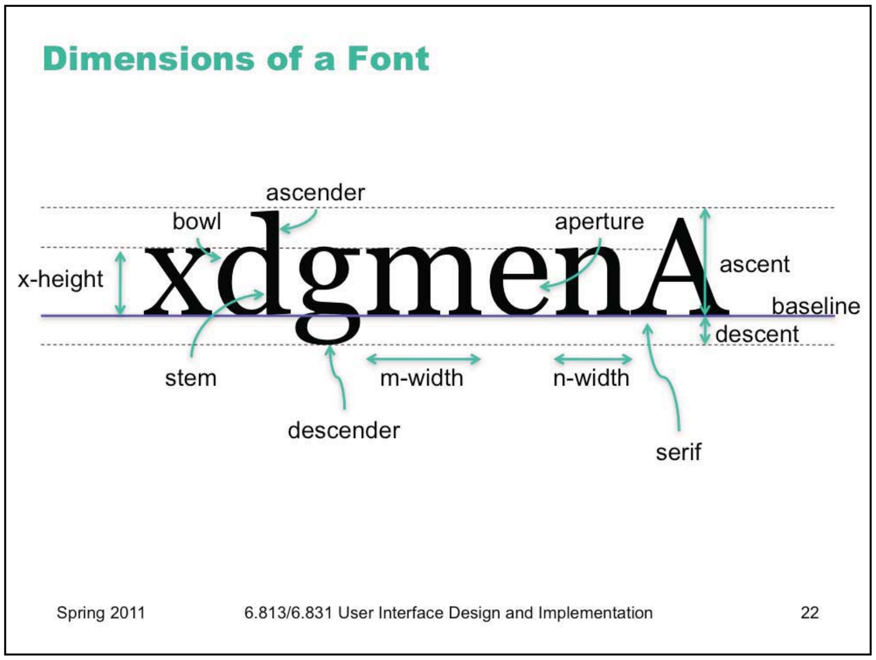
- Displaying text on the screen
  - Key decisions: font and whitespace
- Reading
  - Reading process: fixations and saccades
  - Readability vs. legibility
  - Speed, comprehension, subjective preference

Spring 2011 6.813/6.831 User Interface Design and Implementation 21

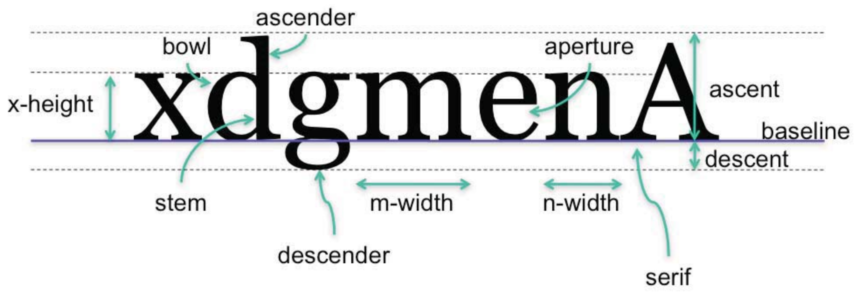
Now we turn from color design to **typography**, the art and science of displaying text (or "setting type" as print designers call it). The key decisions of typography concern font (the shapes of letters and other characters) and spacing (the white space around letters, words, lines, and paragraphs). Both are important to successful text display; without adequate spacing, the shape of the text is much harder for the eye to discriminate.

In keeping with our brief tours of cognitive science for each design topic, let's say a bit about what's known about reading. (Note that these high-level comments are applicable to most written languages, not just English. Most of the rest of the lecture will be specific to languages that use the Latin alphabet and its corresponding fonts, however.) First, reading is not a smoothly linear process, even though it may feel that way to introspection. The eye does not move steadily along a line of text; it proceeds in fits and starts, called **fixations** (stopping and focusing on one place to recognize a word or several words at a time) and **saccades** (an abrupt jump to the next fixation point). At the end of a line, the eye must saccade back to the beginning of the next line.

Researchers studying reading and typography often make a distinction between *legibility*, which is low-level and concerns how easy it is to recognize and distinguish individual letter shapes, and *readability*, which concerns the effectiveness of the whole reading process. A single fixation can consume whole words or multiple words, so fluent readers recognize the shape of an entire word, not necessarily its individual letters. Readability can be measured by several metrics, including speed, comprehension, error rate, and subjective preference. Readability is essentially the usability of a display of text.



Dimensions of a Font

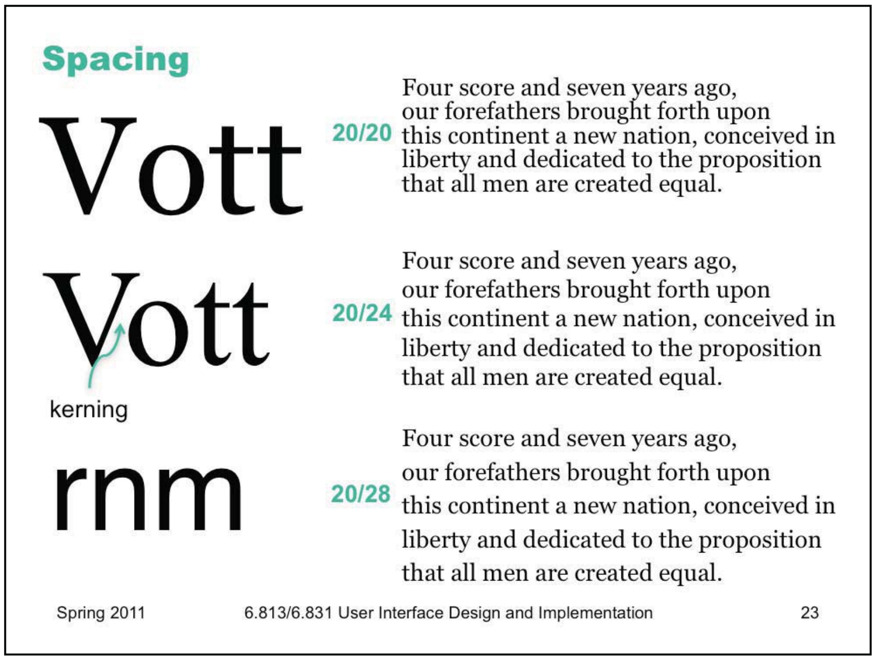


Now a few definitions. Characters in the Latin alphabet all sit on a common baseline. Some characters have descenders dipping below the baseline; others have ascenders rising above the typical height of a lowercase character (the **x-height**). Capital letters also ascend above the x-height. The typical height of ascenders above the baseline is called the **ascent**, and the typical height of descenders below it is called the **descent** of the font.

The **font size** is typically ascent + descent (but not always, alas, so two fonts with the same numerical size but using different typefaces may not line up in height!). Font size is denoted in points; a point is 1/72 inch, so a 12-point font occupies 1/6 of an inch vertically.

X-height, m-width, and n-width are useful font-dependent length metrics. You can find them used in CSS, for example. They allow specifying lengths in a way that will automatically adapt when the font is changed.

More font terminology can be found at http://www.davidairey.com/images/design/letterform.gif.



Spacing

Vott
Vott
kerning
rnm

20/20 Four score and seven years ago, our forefathers brought forth upon this continent a new nation, conceived in liberty and dedicated to the proposition that all men are created equal.

20/24 Four score and seven years ago, our forefathers brought forth upon this continent a new nation, conceived in liberty and dedicated to the proposition that all men are created equal.

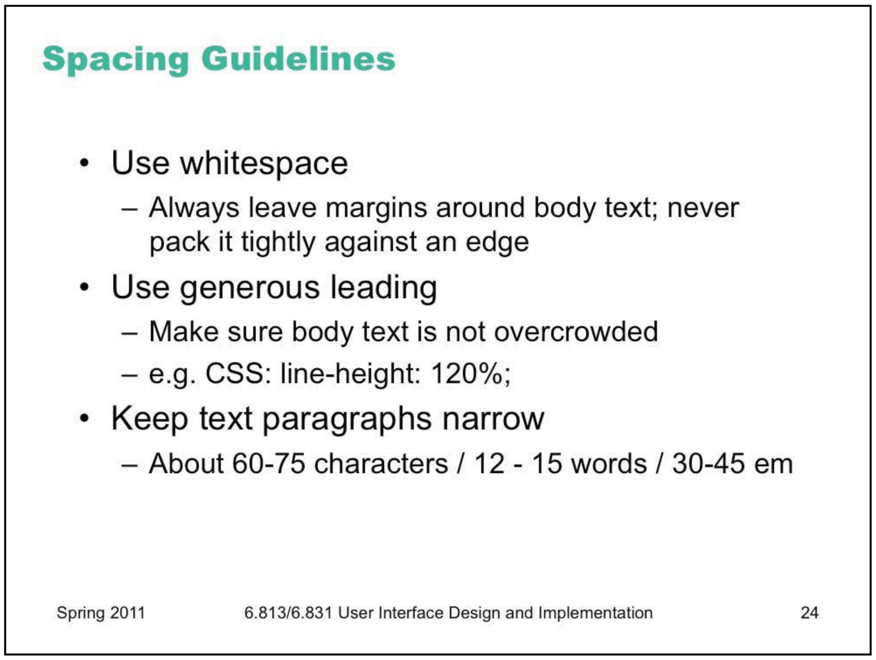
20/28 Four score and seven years ago, our forefathers brought forth upon this continent a new nation, conceived in liberty and dedicated to the proposition that all men are created equal.

Spring 2011 6.813/6.831 User Interface Design and Implementation 23

Several kinds of spacing matter in typography. At the lowest level is character spacing. The gaps between characters must be managed to prevent uneven gaps or characters appearing to run into each other. **Kerning** is the process of adjusting character spacing for particular pairs of characters; sometimes it needs to be narrowed (as V and o shown here), and sometimes widened (e.g. to keep "rn" from looking too much like "m"). A good font has kerning rules built into it, and a good GUI toolkit uses them automatically, so this is rarely something GUI programmers need to worry about, except when choosing a font. Note that the top Vott is displayed in Georgia, which at least on my system appears *not* to have any kerning for V and o. The second Vott is displayed in Times New Roman.

Spacing between words and lines matters too. Words must have adequate space around them to allow the word shape to be easily recognized, but too much space interferes with the regular rhythm of reading. Similarly, adequate line spacing is necessary to make word shapes recognizable in a vertical dimension, but too much line spacing makes it harder for the eye to track back to the start of the next line. Line spacing is also called **leading**; technically speaking, the leading is the distance between baselines of adjacent lines. Both font size and leading are important. Print designers say, for example, "12 point type on 14 points of leading" (or "12/14") to indicate that the font size is 12 points (typically ascent + descent) with 2 points of space between the descent of one line and the ascent of the next. Using the same line height as font size (like 20/20) is almost always a mistake; characters from adjacent lines touch each other, and the paragraph is much too crowded. Note that leading also strongly affects the overall **value** of the body text (which type designers somewhat confusingly call the "color" of the text; historically print is mainly black-and-white, of course, but it's confusing when talking about modern printing and modern computer displays). Tight spacing looks much darker than loose spacing.

In most toolkits, choosing a font size implicitly chooses a leading, but the default leading may not always be the best choice. Look at it and adjust if necessary. CSS makes this possible using the line-height property.



Spacing Guidelines

- Use whitespace
  - Always leave margins around body text; never pack it tightly against an edge
- Use generous leading
  - Make sure body text is not overcrowded
  - e.g. CSS: line-height: 120%;
- Keep text paragraphs narrow
  - About 60-75 characters / 12 - 15 words / 30-45 em

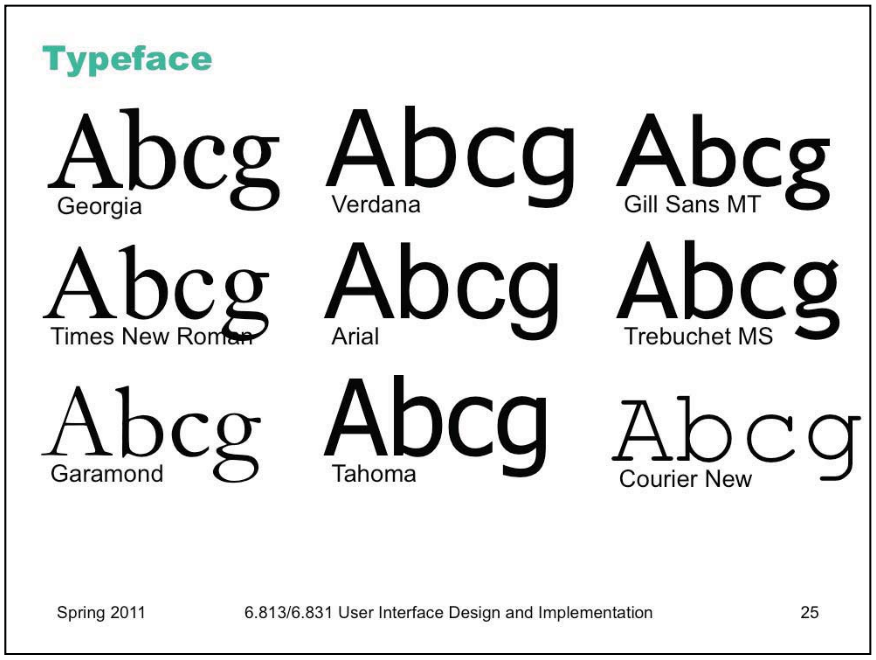
Spring 2011 6.813/6.831 User Interface Design and Implementation 24

Here is some advice for choosing spacing for body text (text set in paragraphs). Always leave margins around body text; never pack it tightly against a rule, an edge, or a window boundary. The margin helps frame the text and also helps the reader find the ends of the lines, which is essential for the saccade back to the beginning of the next line.

Use generous leading, but not too generous. 120% of the font size is a good rule of thumb; this would correspond to the 20/24 leading shown on the previous slide.

Line length (or equivalently, paragraph width) is another important consideration. Hundreds of years of experience from print typography suggest that fairly short lines (3-4 inches) are both faster to read and preferred by users. Unfortunately the studies of onscreen reading yield mixed answers; apparently, on screen, longer lines (about twice the ideal length for print) help users read faster, but users still prefer the short lines (perhaps because their consistent feel with print). These same studies show that onscreen reading is slower than print reading, however, and recent studies have shown less and less effect of line length on speed, possibly because display and font technology is improving rapidly. So it's possible that making the lines longer merely offsets the poorer resolution and legibility of computer displays relative to print, and as the displays approach print in quality, this distinction will go away. (Bailey, "Optimal Line Length: Research Supporting How Line Length Affects Usability", December 2002, http://webusability.com/article_line_length_12_2002.htm)

Translating the 3-4 inch rule into characters or m-widths for typical 10-point to 12-point type gives 60-75 characters or 30-45 em widths.


Typeface

Abcg
Georgia
Abcg
Verdana
Abcg
Gill Sans MT
Abcg
Times New Roman
Abcg
Arial
Abcg
Trebuchet MS
Abcg
Garamond
Abcg
Tahoma
Abcg
Courier New

Spring 2011 6.813/6.831 User Interface Design and Implementation 25

After spacing, a key decision is what **typeface** to use. Typeface refers to a family of fonts sharing the same name, like "Arial" or "Georgia." A **font** is a choice of typeface *and* size *and* style, like roman, italic, oblique, boldface, etc.

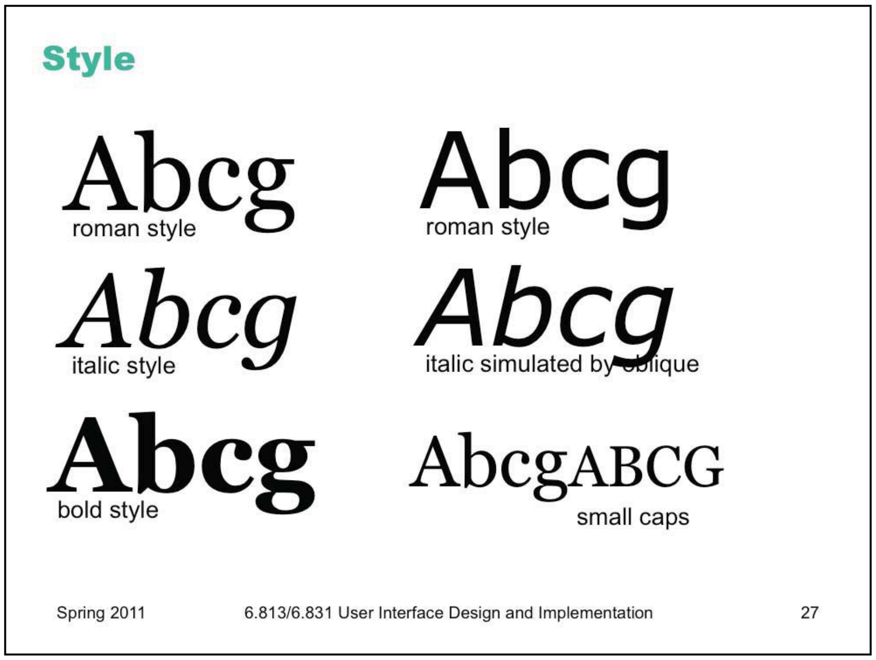
Typefaces can be classified in many ways, and can convey strong associations that influence how the user perceives the text. One important classification you should know is between **serif** fonts, like Georgia and Times, and **sans serif** fonts, like Verdana and Arial. Historically, in print typography, serif fonts have been used for **body text** (text set in paragraphs), because they offer stronger cues to word shape that allow measurably faster reading. Sans serif fonts were generally used for **display text** (text that stands alone, like headings and labels), for which reading speed is less important and contrast from body text is useful.

In the early days of computer typography, sans serif fonts were often preferred for all uses, because their simpler letter shapes were far more legible on low-resolution displays. As displays become higher resolution, however, serif text may once again assert itself; even now, there is evidence that serif fonts are faster to read on screen (Bernard et al, "A Comparison of Popular Online Fonts: Which is Best and When?", Usability News, 2001, http://www.surl.org/usabilitynews/32/font.asp).

Another key distinction is between proportional fonts (in which each character has a different width) and monospace fonts (in which all characters have the same width, like Courier New shown here). Monospace fonts waste screen space and generally look worse than well-designed proportional fonts, so avoid them unless you have a good reason.

Your choice of font family conveys a tone. Some designers think Times and Arial look cheap because they're so widely used; using Georgia or Garamond will give your UI a more "designed" look (i.e., you actually made an informed choice, rather than choosing a default). Another consideration is whether the font was designed for screen use. On this slide, Verdana, Georgia, Tahoma, and Trebuchet were commissioned by Microsoft primarily for onscreen use. Most of the other fonts shown here are digital updates of old fonts originally designed for print. Note some distinct features of Georgia/Verdana/etc. relative to the others – larger x-height (as a fraction of total ascent) and generous bowls and apertures, intended to make the fonts more legible at small sizes on lower-resolution displays.

All the fonts shown here are appropriate and useful for body text (though they aren't the only possibilities, of course). Fonts for body text are designed to have evenly-distributed "color" (the value of the text in a squint test) so that they look good in bulk. You will find many other fonts installed on your computer, some of them very wacky. Some of these may be useful for occasional display text, but certainly not body text. Unfortunately today's word processors give users many fonts but very little help selecting the right one for the right use. Instead we get a long undifferentiated list of installed fonts that hides gems like Garamond and Georgia in a sea of Comic Sans, Goudy Stout, and Old English Text MT -- some of which probably should not be used in any imaginable circumstance.



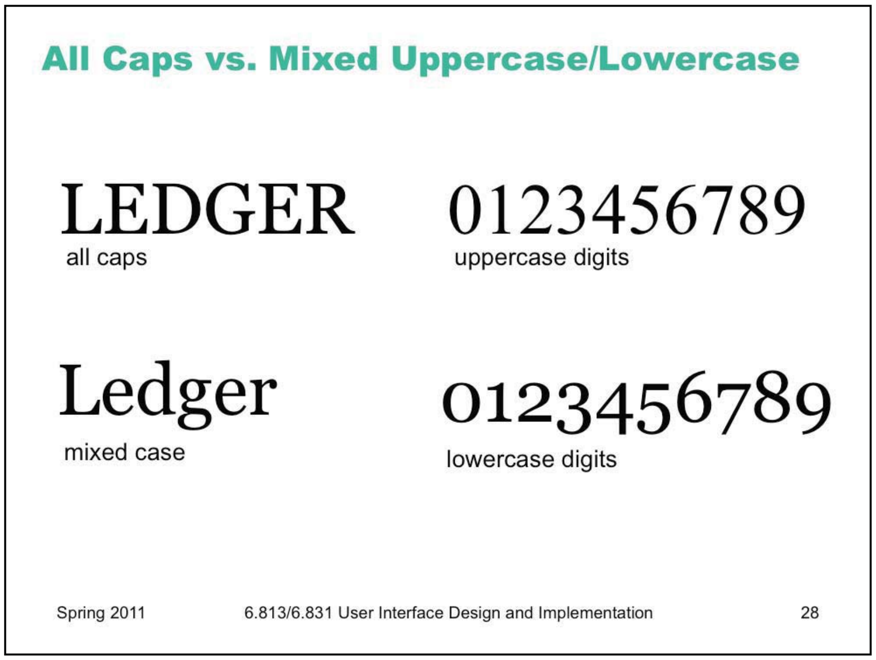
Style

Spring 2011 6.813/6.831 User Interface Design and Implementation 27

We said that a font consists of typeface, size, and **style**. Here are a few common styles you can use to establish contrast.

Italic and boldface create contrast in orientation and value, respectively, without substantially changing the shape of the typeface. Some typefaces lack a true italic, and instead substitute an *oblique* font which is just a slanted version of the normal roman style (sometimes even automatically-transformed from the roman font, not hand-designed). Georgia, shown on the left, has a true italic – notice that the b loses its lower serif, and the g actually changes shape. Sans serif fonts have an oblique rather than italic; look at Arial for an example.

Small caps is another useful style. Small caps are uppercase letters that are as tall as the x-height, rather than the full ascent of the font. Like italic, small caps are sometimes a hand-designed font included with the typeface family (often slightly wider and lighter than capital letters), and sometimes simply automatically generated by shrinking the font.

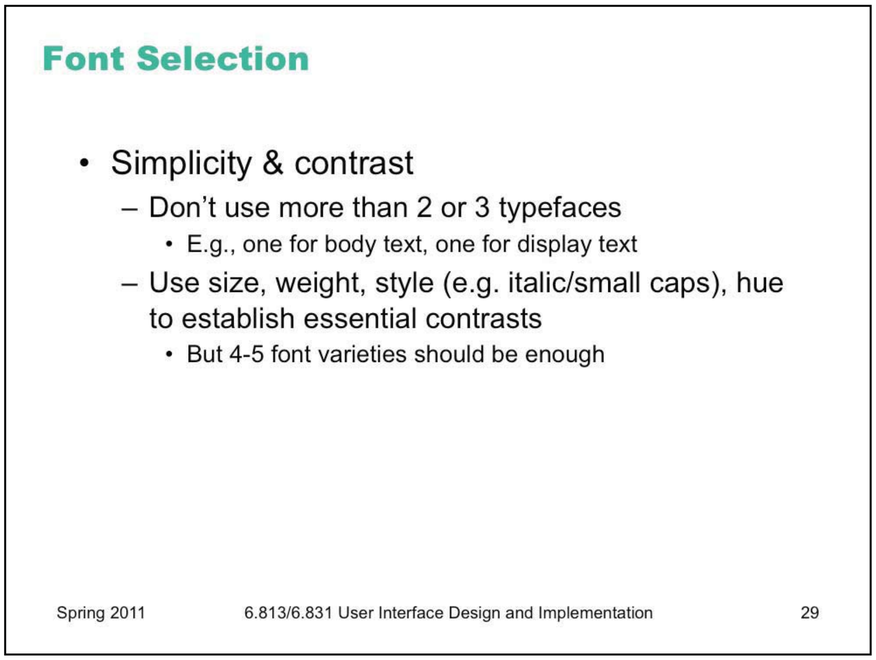


All Caps vs. Mixed Uppercase/Lowercase

LEDGER
all caps
0123456789
uppercase digits
Ledger
mixed case
0123456789
lowercase digits

While we're talking about capital letters, it's worth discussing when it's appropriate to set text in all capitals. All-caps has very little variation in word shape, because all the letters have the same top (the full ascent of the font) and the same bottom (the baseline, with almost no descenders). For this reason, it's both slow and unsatisfying to read body text set in all-caps. All-caps should be reserved only for display text (headings, labels, etc), and even then used very sparingly.

Older print typography actually had lowercase *digits*, not just letters. Notice that the lowercase digits predominantly follow the x-height, with ascenders and descenders for certain digits, just like lowercase letters. You may have seen typesetting like this in older books, published in the first half of the 20th century or earlier. Lowercase digits fell out of fashion in print in favor of more uniform uppercase digits, which may be monospaced horizontally as well, so that columns of digits line up easily; all the digits in Times New Roman have equal width, for example, even though the rest of the typeface is proportional. But lowercase digits are worth some consideration. They are more readable in body text than uppercase digits, for the same reasons as lowercase letters, and they convey a feel that is simultaneously retro and "designed." Unfortunately the character sets we use (ASCII and Unicode) make no distinction between lowercase 5 and uppercase 5 (unlike a and A), so when you choose a typeface, you either get *only* lowercase digits (like Georgia on the bottom) or *only* uppercase digits (like Times on top).



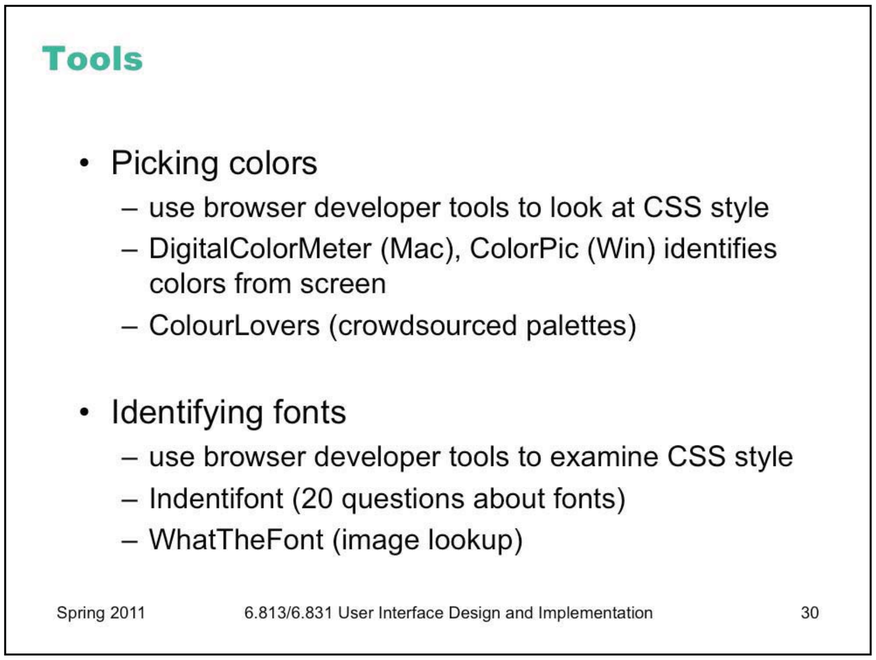
Font Selection

- Simplicity & contrast
  - Don’t use more than 2 or 3 typefaces
    - E.g., one for body text, one for display text
  - Use size, weight, style (e.g. italic/small caps), hue to establish essential contrasts
    - But 4-5 font varieties should be enough

Spring 2011 6.813/6.831 User Interface Design and Implementation 29

In general, decisions about typography are like other decisions in graphic design: use font selection to make important contrasts, and otherwise keep your font choices simple. Don’t use more than 2 or 3 typefaces (if that many). You might use a serif face for body text, and a sans serif face for display text. Many interfaces have no real need for body text at all, in which case you can easily get away with a single typeface.

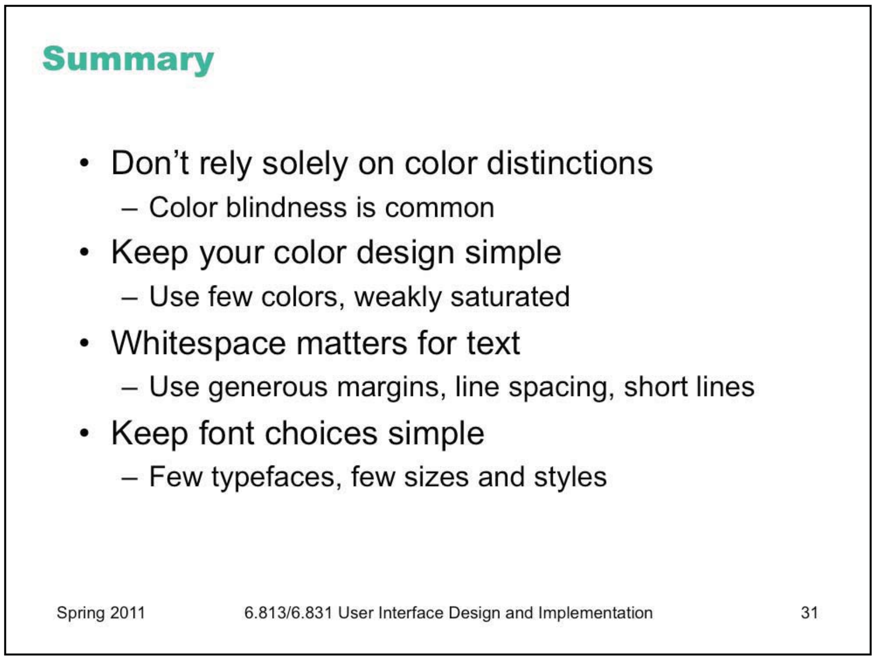
Within the typefaces you chose, use variation of size and style (and color) to establish the necessary contrasts. Size, in particular, makes it easy to establish a hierarchy, such as headings and subheadings. Even so, 4-5 fonts in all should be all you need.



Tools

- Picking colors
  - use browser developer tools to look at CSS style
  - DigitalColorMeter (Mac), ColorPic (Win) identifies colors from screen
  - ColourLovers (crowdsourced palettes)
- Identifying fonts
  - use browser developer tools to examine CSS style
  - Indentifont (20 questions about fonts)
  - WhatTheFont (image lookup)

Spring 2011 6.813/6.831 User Interface Design and Implementation 30



Summary

- Don’t rely solely on color distinctions
  - Color blindness is common
- Keep your color design simple
  - Use few colors, weakly saturated
- Whitespace matters for text
  - Use generous margins, line spacing, short lines
- Keep font choices simple
  - Few typefaces, few sizes and styles

Spring 2011 6.813/6.831 User Interface Design and Implementation 31