painting and drawing programs

(continued)

colour intensities in binary

• in the computer memory, colours are represented in binary notation (0's and 1's)
• 8 bits (= 0/1 variables) are needed to represent 256 possibilities

more on binary notation

• 1 bit can represent 2 possibilities: 0 or 1
• 2 bits can represent 4 possibilities: 00, 01, 10, 11
• How many possibilities can be represented with 3 bits? (Can you list them?)
• How many possibilities can be represented with 8 bits?

specifying colour intensities

• you can specify a colour precisely, using three numbers between 0 and 255
  – e.g.: 0, 5, 255
  – human friendly
• or, you can use three 8-bit binary numbers
  – e.g.: 00000000, 00000101, 11111111
  – computer friendly
• or, you can use hexadecimal notation
  – a compromise between human and computer
  – see pages 105-106 of the text

hexadecimal notation

• decimal notation uses 10 digits
• binary notation uses 2 digits (0 and 1)
• hexadecimal notation uses 16 digits:
  – '0' through '9' (10 of the digits)
  – 'A' through 'F' (6 more "digits" representing 10 through 15)

table of first 16 numbers

... fill in missing spaces!

Note: leftmost 0's optional

decimal | binary | hexadecimal
---|---|---
00 | 0000 | 00
01 | 0001 | 01
02 | 0010 | 02
03 | 0011 | 03
04 | 0100 | 04
05 | 0101 | 05
06 | 0110 | 06
07 | 0111 | 07
08 | 1000 | 08
09 | 1001 | 09
10 | 1010 | 0A
11 | 1011 | 0B
12 | 1100 | 0C
13 | 1101 | 0D
14 | 1110 | 0E
15 | 1111 | 0F
from binary to decimal
- see page 292 of the text
- example:

```
1 1 0 0 1
```

1s place multiply by 1: 1
2s place multiply by 2: 0
4s place multiply by 4: 0
8s place multiply by 8: 8
16s place multiply by 16: 16
Total (add them up): 25

from hexadecimal to binary and back
- easy: replace each hexadecimal "digit" with the corresponding four binary digits
- examples:

<table>
<thead>
<tr>
<th>hexadecimal</th>
<th>binary</th>
</tr>
</thead>
<tbody>
<tr>
<td>AC</td>
<td>10101100</td>
</tr>
<tr>
<td>11</td>
<td>01101001</td>
</tr>
<tr>
<td>FF</td>
<td>11111111</td>
</tr>
</tbody>
</table>

why hexadecimal
- less cumbersome than binary for humans
  - e.g., easier to specify an HTML colour
- easy to convert from hexadecimal to binary (computer representation) and back

exercises
- Can you convert the binary number 10010110 to decimal? To hexadecimal?
- Try other 8-bit binary numbers.
- Use binary to decimal converters on the web to check your answer
  - e.g.: http://www.aquariussoft.com/ce-binary-converter
- (With a bit of effort, you could probably write your own converter in javascript...)

concepts: representing images digitally
- “Text and music have had abstract symbolic notational systems for thousands of years: the visual arts have just achieved such a system for the first time” - Anne Morgan Spalter
- How might you write down an accurate representation of an image, from which someone else could recreate the image?

bit-mapped image representation
- also called raster image representation
- image is partitioned into tiny squares
- a sample of the colour in each square is obtained
- the colour of the square is set to that of the sample (represented as RGB colour intensities)
- a pixel (picture element) is the colour of the point sample
bit-mapped images, continued

- resolution-dependent: scaling up diminishes quality

applying the concepts: how filters work

- filters *globally transform* an image, by changing color intensity values of pixels

filters for blurring

- to blur an image, the colour value of each pixel is averaged with the values of its neighbours
- to avoid unexpected colours at interfaces between objects, UP darkens blurred areas
- the custom filter can also be used to control the averaging weights when blurring

concepts: compressing bit-mapped images

- bit-mapped image files can be big
- but, bit-mapped image can often be compressed without significant loss in quality
- What might be reasonable approaches to image compression?

some compression approaches

1. areas with similar hue can be modified to have the same hue (*lossy compression*)

2. "runs" of identical intensities are identified as such, rather than repeating the intensities

3. recurrent patterns in the sequence data is replaced by short "codes", and a dictionary of codes is maintained

image file types

- GIF (graphic image format) files use
  - 8 bits per pixel to represent colour intensities
  - compression approach 3

- JPEG (joint photographic experts group) files use
  - 24 bits per pixel to represent colour intensities
  - compression approaches 1 and 2 (lossy)
  - user can choose size/quality trade-off