

## Class 34: Storing Images as Simple Pixel Maps

**Held:** Wednesday, 2 April 2008

**Summary:** We consider techniques for storing images in files so that these images can later be retrieved.

### Related Pages:

- EBoard.
- Lab: Pixel Maps.
- Reading: Pixel Maps: A Technique for Storing Images.
- Due: Assignment 7: Spirograph-Like Drawings.

### Notes:

- I will continue to reserve time at the start of class for a discussion of campus issues.
- Preregistration is approaching soon. I hope many of you will think about taking CSC161, the next CS course.
- Happy April Fools' Day! (or not).

### Overview:

- Files, revisited.
- Storing images.
- Storing colors.
- Storing images, revisited.

## Files, Revisited

- Yesterday, we considered files.
- Files are a heterogeneous compound data type.
  - “data type”: Mechanism for organizing information
  - “compound”: That group together lots of values (similarly to how lists group values)
  - “heterogeneous”: And allow multiple kinds of values in the group.
- We use files to ensure that data persist between invocations of programs.
- We use files to share data between programs.
- Common file formats (such as text files) make it easier to share data between programs.

## Saving Images in Files

- Since we've been focusing on images for much of this semester, it makes sense to think about how you would store an image in a file.
- There are dozens of standard file formats for images. Most are relatively complex.

- Hence, we will start by designing our own image file format.
- The easiest model: An image file contains a sequence of color values, one for each pixel in the image.
  - We write and read the pixels row by row from top to bottom
  - We write the pixels in each row from left to right
- The problem of writing images then becomes a problem of writing colors.

## Writing Colors

- So, how do we store a color in a file?
- Three obvious options:
  - Write the color name
  - Write the RGB number (whatever that number means)
  - Write the three components
- Color names
  - Easy for humans to read and write
  - Slow to compute with
- RGB numbers
  - Fast to compute with
  - Hard for humans to read and write
- RGB components as numbers
  - Relatively fast to compute with
  - Not too hard for humans to read and write

## Looking Ahead

- Over the next few days, we'll consider ways to make the image files we create smaller.
- Two basic strategies:
  - Write less per color
  - Write fewer colors per image

---

Copyright © 2007-8 Janet Davis, Matthew Kluber, and Samuel A. Rebelsky. (Selected materials copyright by John David Stone and Henry Walker and used by permission.) This material is based upon work partially supported by the National Science Foundation under Grant No. CCLI-0633090. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. This work is licensed under a Creative Commons Attribution-NonCommercial 2.5 License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc/2.5/> or send a letter to Creative Commons, 543 Howard Street, 5th Floor, San Francisco, California, 94105, USA.