Manipulating Images

Shiffman Chpt. 15

Image Representation

- Pixel grid
  - Images are represented as a grid of color and alpha values
- Color is RGB
- Alpha defines transparency
  - [0..255], 255 is opaque

PImage Class

- Processing class for storing and manipulating images
- Loading and displaying an image:

```
PImage img;  //declare instance
img = loadImage(<fname>); //load from file
image(img, 0, 0); //display image at (0,0)
```

- Generic form of the *image* function:

```
image(<imageObject>, <x>, <y>, <width>, <height>);
```

File Locations

- To add an image to the sketch:
  - Sketch->Add File…

    OR
  - Sketch->Show Sketch Folders and manually drag files to where you need them
- Do not use absolute file names in sketches
  - e.g. avoid: "C:/user/images/gorilla.jpg"
  - Why?

More Image Functions

- Create a blank image
  - createImage(<width>, <height>, RGB);
- Set image as background
  - background(<imageObject>);

State Transformations

- Current rendering state will affect image
  - e.g. translate() or rotate() can be called before calling image() to adjust the image

Example of Simple Image Manipulation and Display

tint()

- tint() is an image (or video) specific rendering state
- Call it *before* displaying an image
- Adjust the brightness
  tint(<intensity>);//intensity in [0..255]
- Adjust the brightness and transparency
  tint(<intensity>, <alpha>);//alpha in [0..255]
- Adjust the brightness of each color channel

tint(<R_intensity>, <G_intensity>, B_intensity>);

- Also with alpha

tint(<R_intensity>, <G_intensity>, <B_intensity>, <alpha>);

Interactive Tint Control

- Map tint values to mouse movement
- Can map two color values to mouseX and mouseY
  - Set the other to 0 or 255

Direct Access to Pixels

- For more complicated effects, you may need to directly manipulate individual pixels
- Pixels stored in a 1 dimensional array called *pixels*

0	1	2	3
4	5	6	7
8	9	…	

- img.loadPixels();//copy image values to pixels array
- Access using img.pixels[location]
- location = x + y * width;
  - Number_of_rows * width + number_of_columns

Direct Access to Pixels

- img.updatePixels(); //set image values from pixels array
- To get color components:
  - red(<color>), green(<color>), blue(<color>)
  - e.g. float r = red(img.pixels[27]);

General Manipulation Work Flow

- Initialize image:
  - PImage img;
  - img = loadImage("image.jpg");

    OR
  - img = createImage(<width>, <height>, RGB);
- Transfer pixel values to pixel array
  - img.loadPixels()

General Manipulation Work Flow

- Manipulate Pixels
  - e.g. img.pixels[index] = color(0,255,0);
- Transfer pixel values back to image
  - img.updatePixels()
- Display image
  - image(img, 0, 0);

Can Work With Multiple Images at Once

- Write/copy to other image, instead of screen:

PImage source1;

Pimage source2;

PImage dest; //destination

- Example

Accessing the "Displayed" Image

- loadPixels(), pixels and updatePixels() can be used directly i.e. with no explicit Pimage
- They will act on the current image
  - This is an image associated with the window
- Flashlight example from Shiffman