Exploring Images

Subject Area(s)	Science & technology
Associated Unit	None
Associated Lesson	None
Activity Title	Exploring Images
Header	Insert Image 1 here, right justified to wrap

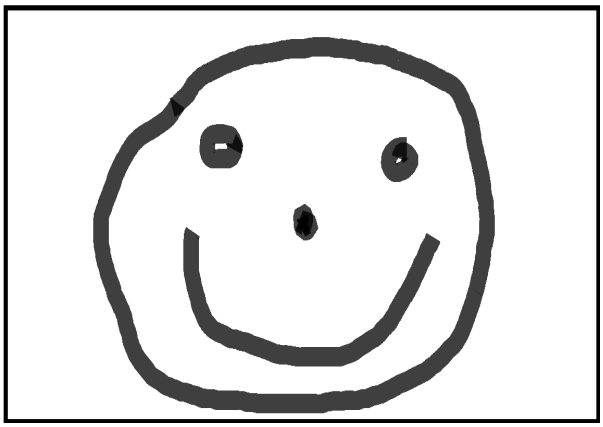
Image 1

ADA Description: Students using programming to process images in different ways

Caption: Students performing image processing

Image file name: exploring_images1.jpg

Source/Rights: Copyright 2010 Pavel Khazron. Used with permission.



Grade Level	6-8
Activity Dependency	Image Scanner!
Time Required	50 minutes
Group Size	3
Expendable Cost per Group	US$0

Summary
Students learn about different ways in which images can be manipulated using computer programs. Using the Processing programming environment, students modify example programs to produce different effects on given images: brightening or darkening an image, cropping sections from images, image rotation, and zooming.

Engineering Connection
After an image has been read or scanned into a computer, a digital artist or an image processing engineer often has to find ways of improving or modifying images to suit an application. For example, some images taken in dark conditions, such as under water, need to be made brighter to make the subjects appear more visible. At times, only a small part of an image is of interest, such as occurs in photographs of crowds. Image zooming (or magnification) is used to make small or far-away features appear larger. Finally, image rotation may be needed to level images taken under an angle, as for example in space photography.

Engineering Category

Category 3: Engineering analysis or partial design

Keywords
Image, sense of sight, image editing, programming

Educational Standards
- New York state intermediate-level science: Standard 1: T1.4b, T1.5a
- New York state technology: engineering design, information systems, computer technology

Pre-Requisite Knowledge
Ability to open, modify, and run a programming sketch written in Processing.

Learning Objectives
After this activity, students will be able to:
- Explain what kind of processing a given image may require
- Understand how image editing software can be used to improve image appearance
- Know how to modify and run programs written in a computer programming language

Materials List
Each group needs:
- One laptop or desktop computer with Processing 1.0.8 programming environment and attached programs and images

Introduction/Motivation

Editing images is often needed for various purposes. Have you even taken an image with a digital camera to find that it appears too dark? How about action photographs as in sports (conduct a discussion)?

We will experiment with the following ways of processing images: brightness adjustment, cropping, rotation, and zooming. These are perhaps the most important and useful image editing techniques. Consider the following types of images acquired by a digital camera and think about which one of the four named techniques can improve how the image appears.

- A photograph of a deep sea wildlife taken by a submersible craft
- Image captured while on a roller coaster
- Surveillance photograph of a subject taken on a street in a business section of the city
- Photograph of the space shuttle taking off

Vocabulary/Definitions

Word	Definition
Image editing	Altering the way an image appears.
Brightness adjustment	Altering an image by adding or subtracting a fixed number from all image pixels.
Cropping	Selecting and removing a small area of interest in an image.
Rotation	Rotating an image about some point to change the orientation.
Zooming	Magnification of an image so distant objects appear larger.
Sketch	A name for a computer program written in the language Processing

Procedure

Before the Activity

- Unzip and copy the contents of the **Programs** folder to student computers.

With the Students

1. The four programs in the provided folder each contain one modifiable constant. All constants can be modified following the first example below.
2. **Brightness adjustment** Open the folder **Programs/brightness**. Then open the programming sketch **brightness.pde**.
    a. The program adds or subtracts a number called **constant** from every pixel in the image. Run the program with **constant** set to 0 by assigning zero as **int constant = 0;**. What do you observe? Now change **constant** to a small positive number, for example 5. Run and notice whether the image appears brighter. Try different numbers as well as images. To try a different image, for example **garden**, replace the line **Image = loadImage("../Images/boat.png");** with **Image = loadImage("../Images/garden.png");**. Different sample images are located in the **Programs/Images** folder.

b. Notice how some pixels change color unexpectedly when **constant** is set to either a sufficiently large or small number. This is because pixels have values 0 to 255, and larger numbers are "wrapped" around. For example, if **constant** is 15, 245 + **constant** = 260. However, the displayed value will actually be 4 (Why?).

3. **Cropping** Open the folder **Programs/cropping**. Then open the programming sketch **cropping.pde**.
    a. The program selects and removes (crops) a rectangular area of an image. The removed piece is displayed under the original image.
    b. Select an area to crop. To do so, modify the numbers **x**, **y**, **dx**, and **dy**. Numbers **x** and **y** are the x and y coordinates of the upper left hand corner of the cropped piece, and numbers **dx** and **dy** are its width and height. Run the sketch and observe results.
    c. Run the sketch with the image **tabletennis**. Challenge: try to crop the white tennis ball!
4. **Rotation** Open the folder **Programs/rotating**. Then open the sketch **rotating.pde**.
    a. This sketch rotates an image by an angle of 0 to 90 degrees. To modify the angle, assign a number between 0 and 90 to the constant **angle**.
    b. Try rotating **boat** in increments of 5 degrees.
    c. Pick an angle to make the box held by the salesman in the **salesman** image appear horizontal.
5. **Zooming** Open the folder **Programs/zooming**. Then open the sketch **zooming.pde**.
    a. In **zooming.pde**, the constant **zoomfactor** controls the amount of zooming. With **zoomfactor** greater than 1, the zoomed image appears larger, while having **zoomfactor** less than 1 shrinks the image.
    b. Try **zoomfactor** equal to 0.5. This will make the image twice smaller. Now try setting **zoomfactor** to 2. When the number is greater than 1, can you better discern the image details?

Attachments
Programs.zip

Troubleshooting Tips

Investigating Questions

Assessment

Pre-Activity Assessment
Sharing Experiences: Lead a class discussion about student experiences with photography in general.

Activity Embedded Assessment
Analysis: Encourage collaboration.

Post-Activity Assessment

What Have We Learned?: Conduct a discussion with the same theme as in the pre-activity assessment.

Activity Extensions
Have students look at the examples provided with Processing, and think of creative ways to modify the examples using the four image processing techniques they have experimented with.

Activity Scaling None

Additional Multimedia Support None

References
Processing programming environment. Accessed February 10, 2010. http://processing.org/.

Free image editing software. Accessed February 10, 2010. http://www.getpaint.net/.

A collection of royalty free images. Accessed February 10, 2010. http://www.images.com/.

Open source 3D content creation suite. Accessed February 10, 2010. http://www.blender.org/.

Redirect URL http://gk12.poly.edu/amps-cbri/

Owner Pavel Khazron

Contributors Pavel Khazron

Copyright The development of this activity was supported by Project AMPS under a GK-12 Fellows grant 0807286 from the National Science Foundation.

Version: January, 2010