Lecture Notes

Overview

Chapter 7 introduces simple graphics and image processing with Python. Some basic concepts, including the RGB system, analog and digital information, and sampling, are introduced. Students learn to work with graphics through many examples using two non-standard, open-source modules: turtlegraphics and images.

Objectives

After completing this chapter, students will be able to:

- Use the concepts of object-based programming—classes, objects, and methods—to solve a problem
- Develop algorithms that use simple graphics operations to draw two-dimensional shapes
- Use the RGB system to create colors in graphics applications and modify pixels in images
- Develop recursive algorithms to draw recursive shapes
- Write a nested loop to process a two-dimensional grid
- Develop algorithms to perform simple transformations of images, such as conversion of color to grayscale

Simple Graphics

Graphics is a discipline that underlies the representation and display of geometric shapes in two- and three-dimensional space.

A *Turtle graphics* toolkit provides a simple and enjoyable way to draw pictures in a window. The turtlegraphics module used in this chapter is a non-standard, open-source Python module. In addition to turtlegraphics, the turtle module provides turtle graphics primitives, in both object-oriented and procedure-oriented ways. For more information, visit: www.python.org/doc/2.3/lib/module-turtle.html.

Overview of Turtle Graphics

Tturtlegraphics uses a Cartesian *coordinate system* with the origin (0, 0) located at the center of a window. Students may find it useful to experiment with Logo before using the turtlegraphics module. If possible, allocate some time for them to experiment with this programming language. For an online Turtle Logo Applet, visit: www.mathsnet.net/logo/turtlelogo/index.html.

Use Table 7.1 to illustrate the most important attributes of a turtle. Together, these attributes make up a turtle's state.

Turtle Operations

Types of objects are called classes and each class includes methods relating to the specific object type.

Use Table 7.2 for a brief description of some of the most important Turtle class methods.

An *interface*, which is the set of methods of a given class, is used to interact with an object. Use the docstring mechanism to view an interface.

It is usually sufficient to know the geometry of the desired graphic and the interface of the appropriate object to create graphics in Python.

The Screen and Canvas classes are also important when working with the Turtle graphics system, and outline the uses of each of these two objects.

Object Instantiation and the turtlegraphics Module

Before you apply any methods to an object, you must create an *instance* of a class. Introduce the terms *instantiation* and *constructor*.

Note that an attempt to manipulate a turtle whose window has been closed raises an error.

Drawing Two-Dimensional Shapes

Many graphics applications use *vector graphics*, which is the drawing of simple two-dimensional shapes such as rectangles, triangles, and circles.

Use the example provided in the book to see how to create a function that draws a polygon using the turtlegraphics module.

Taking a Random Walk

Use the example provided in the book to see how to simulate an animal taking a random walk.

Colors and the RGB System

Review how the *RGB system* encodes color information, as well as the terms *pixel* and *true color*.

Use Table 7.3 to help explain how RGB encodes pixel color information. Note that the total number of bits needed to represent a color value in this system is 24= (8 × 3).

Example: Drawing with Random Colors

Use the example provided in the book to see how to use the penColor method.

Examining an Object's Attributes

Review the difference between mutator methods and *accessor methods*, which return the value of an object's attributes without altering its state.

Manipulating a Turtle's Screen

Keep in mind the Screen and Canvas classes and their uses with respect to a Turtle object.

Research a few examples to illustrate the characteristics of the Screen and Canvas classes and how their methods can be used to manipulate the background of a Turtle graphic.

Setting up a cfg File and Running IDLE

A configuration file and a different way of launching IDLE are required to run a program with the turtle module.

Research an example of the turtle.cfg file, and note some of the initial settings for attributes of Turtle, Screen, and Canvas objects.

Review how an IDLE window should be opened in order to use the turtle module. What happens when the IDLE window is opened the regular way and when you attempt to use the turtle module?

Case Study: Recursive Patterns in Fractals

Fractals are highly repetitive or recursive patterns. Remember, even though a *fractal object* appears geometric, it cannot be described with ordinary Euclidean geometry.

Use Figure 7.6 to see that one example of a fractal curve is the *c-curve*. "The geometric characterization of the simplest fractals is self-similarity: the shape is made of smaller copies of itself. The copies are similar to the whole: same shape but different size." *Reference:* http://classes.yale.edu/fractals/

Use Figure 7.8 and the pseudocode provided in the book to explain how to implement a c-curve function in Python. Note that because fractals are repetitive, their implementation lends itself to recursive programming.

Implementation (Coding)

The program includes the three function definitions of cCurve, drawLine, and main. Because drawLine is an auxiliary function, its definition is nested within the definition of cCurve.

Image Processing

Review the uses of digital image processing. For more information on digital image processing, visit: http://www.ciesin.org/docs/005-477/005-477.html.

Analog and Digital Information

Research the difference between *analog* and *digital information*, and the terms: *discrete values*, *continuous range*, and *sampling*.

Sampling and Digitizing Images

Research how sampling and digitizing of images is performed.

Image File Formats

Once an image has been sampled, it can be stored in one of many file formats. Review the terms: *raw image*, *lossless compression*, *lossy scheme*, and *color palette*.

Note the difference between the JPEG and GIF image formats.

Image-Manipulation Operations

Image-manipulation programs either transform the information in the pixels or alter the arrangement of the pixels in the image.

The Properties of Images

The coordinates of pixels in the two-dimensional grid of an image range from (0, 0) at the upper-left corner to (width-1, height-1) at lower-right corner. This *screen coordinate system* differs from the standard Cartesian coordinate system used by turtle graphics.

Colors of images are typically represented in RGB.

The images Module

Note that the images module is a non-standard, open-source Python tool.

Use the example provided in the book to see that the Image class represents an image as a two-dimensional grid of RGB values.

Using Table 7.4, review some of the most useful Image methods. Use the example provided in the book to see how to use some of these methods.

A Loop Pattern for Traversing a Grid

Research the difference between a *linear loop structure* and a *nested loop structure*. The latter is commonly used to traverse a two-dimensional grid of pixels (see Figure 7.11).

Use the pseudocode provided in the book to see that *row-major traversal* is used to develop many of the algorithms that follow.

Converting an Image to Black and White

Use the example provided in the book to see how to convert an image to black and white.

Converting an Image to Grayscale

Use the example provided in the book to see how to convert an image to *grayscale*. Introduce the term *luminance*.

Copying an Image

The method clone builds and returns a new image with the same attributes as the original one, although with an empty string as the filename.

Blurring an Image

Pixilation can be mitigated by *blurring* the affected areas of an image.

Review the code for blur provided in the book, noting the design considerations listed in bullet points on Pages 280-281.

Edge Detection

Edge detection removes the full colors to uncover the outlines of the objects represented in the image.

Use Figure 7.14 to see how setting different luminance threshold values affects the results of the edge detection algorithm.

Reducing the Image Size

The size and the quality of an image on a display medium depend on two factors: the image's width and height in pixels and the display medium's resolution (measured in pixels or dots per inch).

Note that a higher DPI causes a sampling device to take more samples (pixels) through the two-dimensional grid.

Use the example provided in the book to see how to shrink an image. Note that a size reduction usually preserves an image's *aspect ratio*.Reducing the size of an image throws away some pixel information.

Additional Resources

1. Online Logo Turtle Applet:
   www.mathsnet.net/logo/turtlelogo/index.html

2. Fractal:
   http://classes.yale.edu/fractals/

3. Digital Image Processing:
   http://www.ciesin.org/docs/005-477/005-477.html

4. Image Processing with Python SIG:
   www.python.org/community/sigs/current/image-sig/

Key Terms

- **accessor method(s):** A method used to examine an attribute of an object without changing it.
- **analog information:** Information containing a continuous range of values.
- **aspect ratio:** The ratio between the width and height of an image.
- **blurring:** making jagged edges in an image appear softer. Involves loss of information.
- **c-curve:** A fractal shape that resembles the letter C.
- **constructor:** A method that is run when an object is instantiated, usually to initialize that object's instance variables. This method is named _init_ in Python.
- **coordinate system:** A grid that allows a programmer to specify positions of points in a plane or of pixels on a computer screen.
- **discrete value(s):** Individual information values, such as individual integers, characters of text or bits in a bit string.
- **edge detection:** Removing the full colors from an image to uncover the outlines of the objects represented in the image.
- **fractal geometry:** A theory of shapes that are reflected in various phenomena, such as coastlines, water flow, and price fluctuations.
- **fractal object:** A type of mathematical object that maintains self-sameness when viewed at greater levels of detail.
- **graphics:** A discipline that underlies the representation and display of geometric shapes in two- and three-dimensional space.
- **grayscale:** A color scheme that contains various shades of gray in addition to black and white.
- **instance:** A computational object bearing the attributes and behavior specified by a class.
- **instantiation:** The process of creating a new object or instance of a class.
- **interface:** A formal statement of how communication occurs between the user of a module (class or method) and its implementer.
- **lossless compression:** A compression scheme in which no information is lost.
- **lossy scheme:** A compression scheme in which some of the original color information is lost.
- **luminance:** The amount to which the human eye is sensitive to specific colors.
- **mutator method(s):** A method used to change the value of an attribute of an object.
- **nested loop:** A loop as one of the statements in the body of another loop.
- **object-based programming:** The construction of software systems that use objects.
- **origin:** The point (0,0) in a coordinate system.
- **pixel(s):** A picture element or dot of color used to display images on a computer screen.
- **pixilation:** The condition in which an image appears to contain rough, jagged edges.
- **raw image file:** A file that saves all the information sampled from an image.
- **resolution:** A measure for a display medium's accuracy of display. Measured in pixels or DPI.
- **RGB system:** A system for representing colors which represents the intensity of each of the colors red, green, and blue as integers in the range 0-255.
- **row-major traversal:** A grid traversal scheme in which the rows are traversed in the outer loop and the columns in each row are traversed using an inner loop.
- **screen coordinate system:** A coordinate system used by most programming languages in which the origin is in the upper-left corner of the screen, window, or panel, and the y values increase toward the bottom of the drawing area.
- **Turtle graphics:** A set of methods that manipulate a pen in a graphics window.
- **vector:** A one-dimensional array that supports resizing, insertions, and removals.