

03 Python Graphics

Introduction to Graphics

- Graphics involve the representation and display of geometric shapes in two and three dimensions, as well as image processing

Turtle Graphics

- It provides a visual representation by moving a "turtle" around the screen to create drawings
- **Turtle State** : The position, heading (direction), pen color, and pen state (up or down)
- `setHeading(degrees)` 0(E), 90(N), 180(W), 270(S)

Basic Turtle Operations

1. Moving the Turtle

- `forward(distance)` : Moves the turtle forward by the specified distance
- `backward(distance)` : Moves the turtle backward by the specified distance
- `left(angle)` : Turns the turtle left by the specified angle
- `right(angle)` : Turns the turtle right by the specified angle

2. Pen Control

- `penup()` : Lifts the pen so that moving the turtle does not draw a line
- `pendown()` : Lowers the pen so that moving the turtle draws a line
- `pencolor(color)` : Sets the color of the pen
- `pensize(size)` : Sets the width of the pen

3. Setting the Turtle's Attributes

- `shape(shape)` : Sets the shape of the turtle (e.g., "turtle", "arrow", "circle")
- `speed(speed)` : Sets the speed of the turtle (1 to 10)

4. Filling Shapes

- `begin_fill()` : Starts filling the shape
- `end_fill()` : Ends filling the shape

Drawing Two-Dimensional Shapes

- eg : triangle.py

```
import turtle

def draw_triangle(t, length):
    for _ in range(3):
        t.forward(length)
        t.left(120)

# Setup turtle
t = turtle.Turtle()
draw_triangle(t, 100)
turtle.done()
```

Colors and the RGB System

- Colors in graphics are often represented using the RGB (Red, Green, Blue) system, where each color component can range from 0 to 255

```
import turtle

t = turtle.Turtle()
t.pencolor('red') # Red
t.fillcolor('green') # Green
t.begin_fill()
for _ in range(4):
    t.forward(100)
    t.left(90)
t.end_fill()
turtle.done()
```

Image Processing

- Involves capturing, representing, and manipulating images
- Rotate an image
- Convert an image from color to grayscale
- Apply color filtering to an image
- Highlight a particular area in an image
- Blur all or part of an image
- Sharpen all or part of an image
- Control the brightness of an image
- Perform edge detection on an image
- Enlarge or reduce an image's size
- Apply color inversion to an image

- Morph an image

Convert an image from color to grayscale

```
def grayscale(image):  
    """Converts the argument image to grayscale."""  
    for y in range(image.getHeight()):  
        for x in range(image.getWidth()):  
            (r, g, b) = image.getPixel(x, y)  
            r = int(r * 0.299)  
            g = int(g * 0.587)  
            b = int(b * 0.114)  
            lum = r + g + b  
            image.setPixel(x, y, (lum, lum, lum))  
  
# Test the function  
from images import Image  
image = Image("example.gif")  
grayscale(image)  
image.draw()
```

Image Methods

open()

```
from PIL import Image  
# from images import Image  
  
# Open an image file  
img = Image.open('example.jpg')  
img.show()
```

Image(filename)

```
from PIL import Image  
  
# Open an image file  
img = Image.open('example.jpg')  
img.show()
```

getWidth() & getHeight()

```
height = image.getHeight()
print(f"Height: {height}")

height = image.getHeight()
print(f"Height: {height}")
```

getPixel(x, y)

- Returns the RGB color value of the pixel at the specified (x, y) position

```
pixel = image.getPixel(0, 0)
print(f"Pixel at (0, 0): {pixel}")
```

setPixel(x, y, (r, g, b))

- Sets the color of the pixel at the specified (x, y) position to the given RGB value

```
image.setPixel(0, 0, (255, 0, 0))
# Set the pixel at (0, 0) to red
image.draw()
```

draw()

- Displays the image

```
image.draw()
```

save(filename)

```
image.save("output.gif")
```

copy()

```
image_copy = image.copy()
image_copy.draw()
```

Graphical User Interfaces (GUIs)

- GUIs allow users to interact with software through graphical elements like buttons, icons, and menus, rather than text commands

```
import tkinter as tk

def on_click():
    print("Button clicked!")

root = tk.Tk()
button = tk.Button(root, text="Click Me", command=on_click)
button.pack()
root.mainloop()
```