МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ

по лабораторной работе №2

по дисциплине «Информатика»

Тема: Введение в архитектуру компьютера

Студент гр. 3342 _____ Галеев А.Д.

Преподаватель _____ Иванов Д.В.

Санкт-Петербург

2023

Цель работы

Научиться работать с библиотекой Pillow, а так же изучить ее основные функции и возможности

Задание

Вариант №3

Рисование пентаграммы в круге:

Необходимо написать функцию, которая рисует на изображении пентаграмму в окружности.

Функция принимает на вход:

Изображение (img), координаты центра окружность (x,y), радиус окружности, толщину линий и окружности (thickness), Цвет линий и окружности (color) - представляет собой список (list) из 3-х целых чисел

Функция должна изменить исходное изображение и вернуть его изображение.

Вершины пентаграммы высчитывать по формуле:

phi = (pi/5)*(2*i+3/2)

node_i = (int(x0+r*cos(phi)),int(y0+r*sin(phi)))x0,y0 - координаты центра окружности, в который вписана пентаграммar - радиус окружностиi - номер вершины от 0 до 4

Поменять местами участки изображения и поворот:

Необходимо реализовать функцию, которая меняет местами два квадратных, одинаковых по размеру, участка изображений и поворачивает эти участки на 90 градусов по часовой стрелке, а затем поворачивает изображение на 90 градусов по часовой стрелке.

Функция принимает на вход:

Квадратное изображение (img), координаты левого верхнего угла первого квадратного участка(x0,y0), Координаты левого верхнего угла второго квадратного участка(x1,y1), Длину стороны квадратных участков (width)

Функция должна вернуть обработанное изображение, не изменяя исходное.

Средний цвет:

Необходимо реализовать функцию, которая заменяет цвет каждого пикселя в области на средний цвет пикселей вокруг (не считая сам этот пиксель).

Функция принимает на вход:

Изображение (img), Координаты левого верхнего угла области (x0,y0), координаты правого нижнего угла области (x1,y1)

Функция должна заменить цвета каждого пикселя в этой области на средний цвет пикселей вокруг.

Функция должна вернуть обработанное изображение, не изменяя исходное.

Средний цвет - берется целая часть от среднего каждой компоненты из rgb. (int(sum(r)/count),int(sum(g)/count),int(sum(b)/count))

Основные теоретические положения

Для решения задач в программе использовались библиотеки:

math - библиотека в которой находятся математические функции и числа (например: cos, sin, pi)

PIL - библиотека для работы с изображениями, в нее входят множество функций (например: Image.new, img.copy и другие)

Выполнение работы

Функция swap:

Функция принимает на вход изображение, координаты первого и второго участка квадратной формы с которыми нужно проделать действие и длину стороны каждого из этих участков. Сначала исходя из условий, создается копия изображения, далее в переменные box0 и box1 записываются координаты квадратных участков. В переменные region0 и region1 копируются участки по координатам box0 и box1, с помощью функции .crop. Далее в скопированное изображение помещаются эти участки, после чего с помощью функции rotate они разворачиваются на 90 градусов по часовой стрелке. В конце готовое изображение полностью разворачивается на 90 градусов по часовой стрелке и выводится на экран.

Функция avg_color:

Функция принимает на вход изображение и координаты его краев. Сначала исходя из условий, создается копия изображения, после этого с помощью циклов for проходимся по каждому пикселю, далее с помощью функции get.pixel забираем цвет каждого пикселя в переменную pixel_color. После этого отмечаем границы соседних пикселей для каждого пикселя на изображении, и вносим все соседние пиксели в список neighbors. Далее по данной условием задачи формуле находим средний цвет и заменяем цвет рассматриваемого пикселя на него. В конце выводится скопированное изображение с замененными на соседние цвета пикселями.

Функция pentagram:

Функция принимает на вход изображение, координаты центра окружности, радиус окружности, толщину линий окружности (thickness) и цвет окружности и линий.

Сначала создается список vertices в который добавляются координаты, вычисленные с помощью формул данных в условии задачи и библиотеки math, точек вершин пентограмы которую требуется нарисовать. Далее рисуется окружность в которую вписана пентограма, с помощью радиуса и координат

центра окружности находим края квадрата в который вписана эта окружность. В конце проводится линии которые соединяют вершины пентограмы через одну, после чего выводится результат.

Тестирование

Один из результатов тестирования представлен на рисунке 1

Рисунок 1 - Результат выповления функции swap

Выводы

Была изучена библиотека Pillow.

Разработана программа состоящая из 3 частей, которые используют различные функции и команды из библиотеки Pillow.

ПРИЛОЖЕНИЕ А
ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main_lb2

```
from PIL import Image, ImageDraw
import math


def swap(img, x0,y0,x1,y1,width):
    new_img = img.copy()
    box0 = (x0, y0, x0 + width, y0 + width)
    box1 = (x1, y1, x1 + width, y1 + width)
    region0 = new_img.crop(box0)
    region1 = new_img.crop(box1)
    new_img.paste(region1, box0)
    new_img.paste(region0, box1)
    region0 = new_img.crop(box0).rotate(-90)
    region1 = new_img.crop(box1).rotate(-90)
    new_img.paste(region0, box0)
    new_img.paste(region1, box1)
    new_img = new_img.rotate(-90)

    return new_img

def avg_color(img, x0, y0, x1, y1):
    new_img = img.copy()
    for y in range(y0, y1 + 1):
        for x in range(x0, x1 + 1):
            pixel_color = img.getpixel((x, y))
            x_min = max(x - 1, 0)
            x_max = min(x + 1, img.width - 1)
            y_min = max(y - 1, 0)
            y_max = min(y + 1, img.height - 1)
            neighbors = []
            for ny in range(y_min, y_max + 1):
                for nx in range(x_min, x_max + 1):
                    if nx != x or ny != y:
                        neighbors.append(img.getpixel((nx, ny)))
            count = len(neighbors)
            avg_color = (
                int(sum([c[0] for c in neighbors]) / count),
                int(sum([c[1] for c in neighbors]) / count),
                int(sum([c[2] for c in neighbors]) / count))
            new_img.putpixel((x, y), avg_color)

    return new_img

def pentagram(img, x, y, r, thickness, color):
    draw = ImageDraw.Draw(img)
    vertices = []
    for i in range(5):
        phi = (math.pi/5)*(2*i + 3/2)
        vertex = (int(x + r * math.cos(phi)), int(y + r * math.sin(phi)))
```

```
        vertices.append(vertex)
    draw.ellipse((x - r, y - r, x + r, y + r), outline=tuple(color),
width=thickness)
    for i in range(5):
        draw.line([vertices[i], vertices[(i + 2) % 5]], fill=tuple(color),
width=thickness)

    return img
```