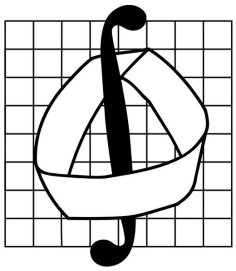
Московский государственный университет имени М.В.Ломоносова
Механико-математический факультет



Отчёт по работе с BMP-изображениями в Python
Изменение первого изображения по черным пикселам второго

Студент: Конобеев Г.В.
Преподаватель: Почеревин Р.В.

Группа: 224

12 апреля 2024г.

Отчёт по работе с BMP-изображениями в Python

1 Условие:

Программа должна загрузить изображения из графических файлов InputFile1, InputFile2, обнулить значения всех пикселов в первом изображении, для которых соответствующий пиксел во втором изображении нулевой и вывести получившееся изображение в графический файл OutputFile. Под нулевыми пикселами подразумеваются пикселы, для которых все три компоненты равны 0.

2 Решение:

Для решения задачи булем использовать библиотеку PIL(импортируем методы Image и ImageDraw):

```
from PIL import Image, ImageDraw
```

А также будем использовать библиотеку sys для загрузки файлов из командной строки:

```
import sys
```

Загружаем изображения, ищем их размеры, чтобы случайно не выйти за границы при перекрашивании пикселов. Для этого финальное изображение будет минимальным по ширине и длине.

```
image = Image.open(sys.argv[1])
image2 = Image.open(sys.argv[2])
draw = ImageDraw.Draw(image)
width = image.size[0]
height = image.size[1]
width2 = image2.size[0]
height2 = image2.size[1]
w=min(width, width2)
h=min(height, height2)
pix = image.load()
pix2 = image2.load()
```

Далее начинаем работать с этими изображениями. Наша цель - найти все нулевые пикселы второго изображения. Будем перебирать их с помощью двойного цикла совместно с пикселами первого изображения

```
for i in range(w):
    for j in range(h):
        a = pix2[i, j][0]
        a = pix[i, j][0]
        b = pix[i, j][1]
        c = pix[i, j][2]
        a1 = pix2[i, j][0]
        b1 = pix2[i, j][1]
        c1 = pix2[i, j][2]
```

В том же цикле тут же будем менять цвет пиксела первого изображения на цвет пиксела второго, если условие (0,0,0) для цвета соблюдено. Для этого будем заново рисовать на нашем первом изображении с помощью команды *point*.

```
if a1==0 and b1==0 and c1==0:
    draw.point((i, j), (a1, b1, c1))
else:
    draw.point((i, j), (a, b, c))
```

Сохраним изображение и очистим память от элемента *draw*:

```
image.save(sys.argv[3])
del draw
```

Для запуска программы необходимо передать следующие параметры через командную строку:

- **sys.argv[0]**: Имя программы
- **sys.argv[1]**: Путь к изображению для редактирования
- **sys.argv[2]**: Путь к изображению, используемому для редактирования
- **sys.argv[3]**: Путь для сохранения полученного изображения

3 Использование библиотек

Для работы программы необходимо установить библиотеку Python Imaging Library (сокращенно **PIL**). Она предназначена для работы с растровой графикой.

Для ее установки следует прописать в терминале следующую команду: **pip install pillow**

4 Результат работы программы

Рис. 1: Изображение для редактирования

Рис. 2: Изображение, используемое для редактирования

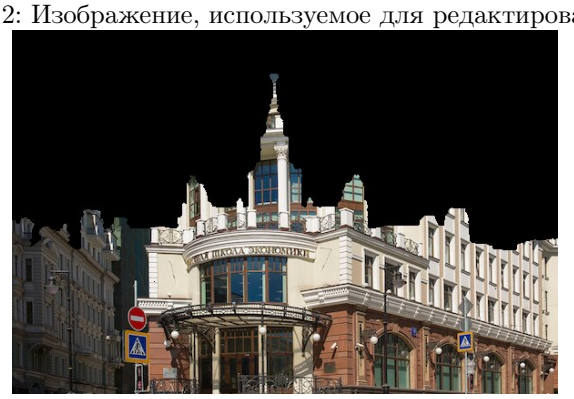


Рис. 3: Итоговое изображение