Государственное бюджетное общеобразовательное учреждение города Москвы "Школа № 1532"

**Решение задач раскроя программным методом**

Скородина Майя Алексеевна, 9 класс,

ГБОУ школа 1532  
Руководитель: Сергиенко Антон Борисович, учитель информатики,

ГБОУ Школа №1532

Москва, 2022

**Оглавление**

Введение……………………………………………………………….….………3

Цели и задачи работы…………………………………………………….………3

Методика выполнения и результаты…………………………………….………3

Вывод…………………………………………………………………….…..….…8

Список используемой литературы………………………………………..……...9

**Введение**

Выбранная задача состояла в написании приложения с помощью языка Python, способного смоделировать на заданной поверхности наиболее компактное расположение данных объектов в формате PNG с возможностью регулировки количества размещаемых объектов и размеров печатаемой поверхности.

В наши дни многие сферы не обходятся без того или иного вида печати: от распечатки наклеек на листах бумаги, до печати заготовок и деталей на листах металла. И зачастую после печати и вырезания объектов остается большое количество отходного материала, или «обрезков». Одна из задач проекта – сокращение количества расходов, остающихся после печати.

**Цели, задачи работы и актуальность**

Цель работы — создание приложения, решающего задачи раскроя с произвольными фигурами разными методами. Такое приложение будет

Задачей работы является создание программы на Python с использованием библиотеки Pillow, решающей задачи раскроя с произвольным изображением, заданным пользователем через интерфейс. И как следствие использование программы – сокращение числа остаточного материала, появившегося после обработки напечатанных изделий.

Задача актуальна, так как стандартные решения задач раскроя не предусматривают использование произвольных фигур.

**Методика выполнения и результаты**

Этапы:

1) Создание интерфейса с помощью Qt creator, подключение к нему обработки событий

2) Загрузка изображения. Подсчет площади фигуры на изображении в пикселях

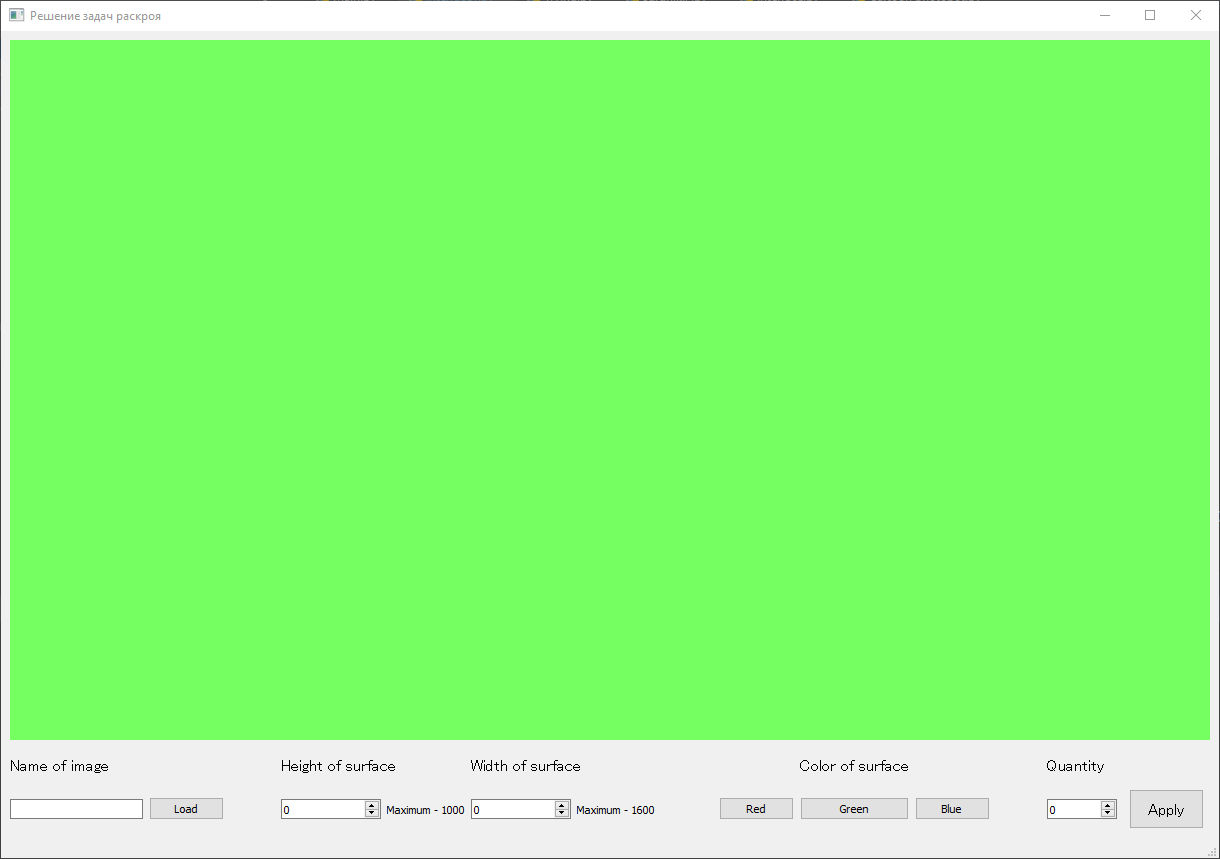
3) Подсчет максимального количества фигур, которые можно разместить на основной поверхности.

4) Создание матриц для основной поверхности и для фигуры. Это нужно, чтобы контролировать ситуации наложения фигур при размещении. (Подробнее - далее)

5) Размещение фигуры на основной поверхности с использованием циклов.

Методика:

1. Создание интерфейса с помощью Qt creator:

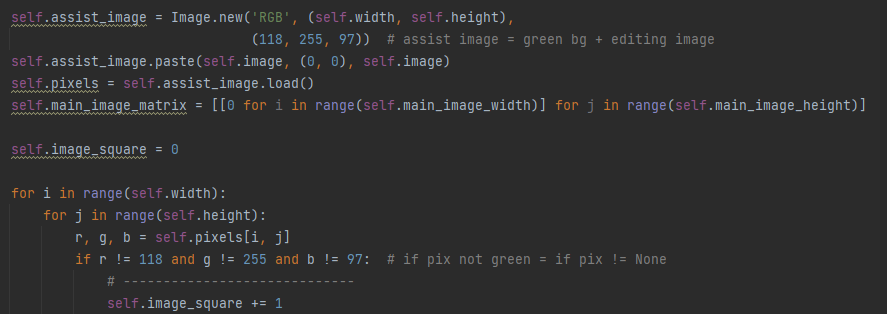


Перед вами стартовое окно. При открытии программы пользователь видит следующие виджеты:

* Name of image: поле для ввода и кнопка Load. Сюда пользователь вводит имя файла и путь к нему и нажимает кнопку для загрузки изображения.
* High of surface и Width of surface: два поля для ввода размеров основной поверхности
* Color of surface: Кнопки, меняющие цвет основной поверхности (изначально – «Green»)
* Quantity: поле для ввода числа: количества дублирований заданного изображения на основной поверхности

1. Загрузка изображения. Подсчет площади фигуры на изображении в пикселях:

В представленном фрагменте кода идет наложение фигуры на фон определенного цвета с помощью библиотеки Pillow. Далее идет подсчет количества пикселей, чей цвет не совпадает с фоновым. Количество пикселей и есть площадь фигуры.



1. Подсчет максимального количества фигур:

Для расчёта используется формула:

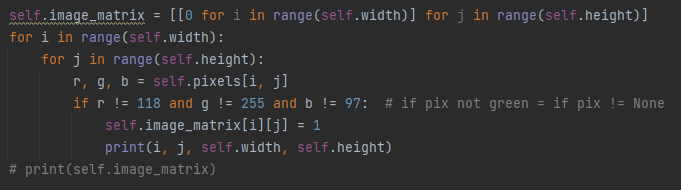
(Площадь основной поверхности) // (площадь фигуры)

(где // - целочисленное деление)

1. Создание матриц для основной поверхности и для фигуры:

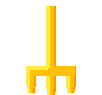
Матрицы создаются по принципу:

* 1. Инициализация списка, включающего в себя вложенные списки, количество которых = высоте поверхности или фигуры. В каждом вложенном списке содержатся «0», количество которых = ширине поверхности или фигуры. Так получается пока пустая матрица.
  2. Замена «0» на «1» в местах, где альфа канал пикселя > 0. По принципу, схожему с тем, что использовался для подсчета площади.

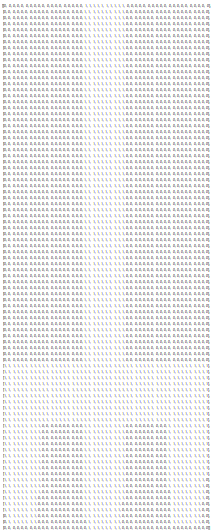


Перед вами фрагмент кода, отвечающий за заполнение единицами матрицы фигуры.

Если мы загрузим в программу следующее изображение:



То его матрица будет выглядеть так:



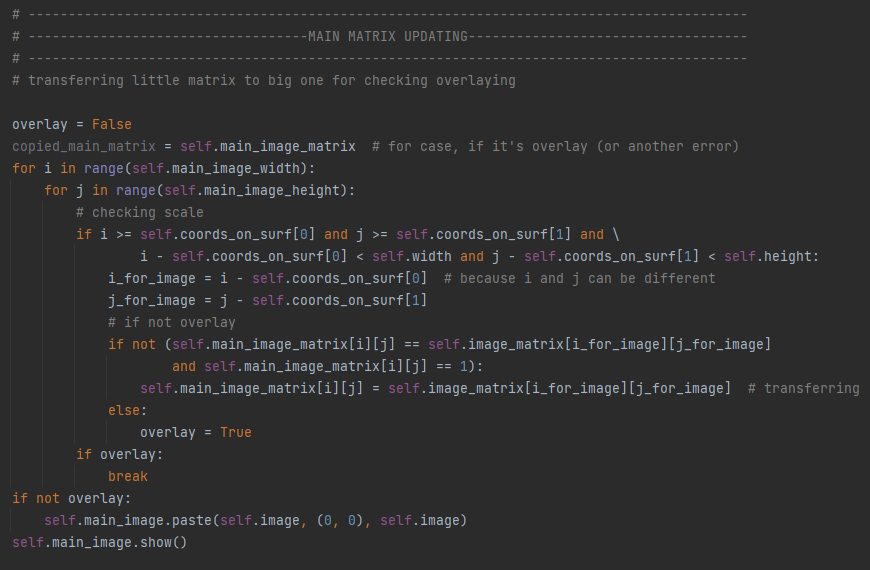
1. Размещение фигуры на основной поверхности:

В цикле для размещения каждый раз заданная фигура должна получать две характеристики: угол поворота и кортеж с координатами.

Наиболее простым в реализации способом будет задание этих параметров с помощью методов рандома.

После получения параметров в цикле, матрица фигуры меняется (если угол поворота не равен 0) и «накладывается» на основну. В процессе наложения «1» из матрицы фигуры копируются в матрицу основной поверхности с учетом заданных координат.

Ниже представлен фрагмент кода, в котором идет процесс наложения. Стоит также отметить, что автор предусмотрел возможность наложения фигур, поэтому при обнаружении таковых цикл переноса матрицы останавливается и начинается заново уже с другими параметрами поворота и координат.



По завершению работы цикла программа выводит получившееся изображение, в котором количество фигур = максимально допустимому, определенном в начале:

#картинка

**Выводы**

В работе автором была реализована программа, решающая задачу раскроя с произвольными фигурами, а так же реализована возможность пользователя самостоятельно задавать основные параметры для создания необходимого раскроя.

**Список импользуемой литературы**

* 1. Документация библиотеки Pillow [Internet] <https://pillow.readthedocs.io/en/stable/index.html>
  2. PyQt6 — полное руководство для новичков / Хабр [Internet]

<https://habr.com/ru/company/skillfactory/blog/599599/>

* 1. PyQt5 Tutorial Documentation, Release [Internet] https://pyqt5.files.wordpress.com/2017/06/pyqt5tutorial.pdf