Растровый графический редактор

«Qt Paint»

Идея проекта состоит в том, чтобы впервые поучаствовать в создании оконного приложения. Ну а поскольку на повестке дня было изучение PyQt5, его функционала, классов и дизайнера, было решено написать программу с пользовательским интерфейсом, являющимся аналогом к MS Paint.

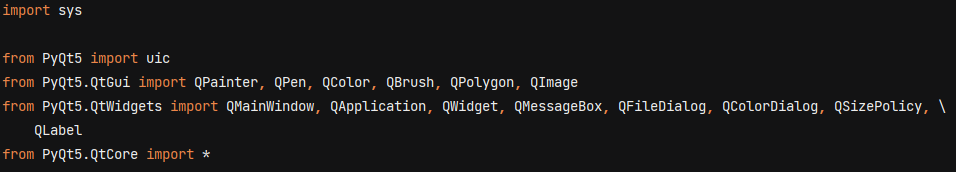
Основные классы - Window и Canvas. Window подключает .ui файл, содержащй сам интерфейс и все его активных элементы, к основному коду и отвечает за то, чтобы все элементы были на своем месте и могли исполнять свои предполагаемые действия. Canvas – перегруженный встроенный класс QWidget, который является основой для любого виджета в Qt, будь то кнопка или диалоговое окно. Поскольку в Qt нет встроенного виджета-холста для рисования, но движок предусматривает рисование поверх виджета, можно сделать холст одним большим «самодельным» виджетом, чем и является объект класса Canvas.

Ну а чтобы Canvas не был чересчур забитым параметрами, основные инструменты также имеют свои классы, где они определяются вместе с переданными параметрами и имеют общий метод draw, где создается инструкция по рисованию того или иного примитива, собирается в единый объект и передается в Canvas. paintEvent в Canvas просто берет список всех объектов, принятых от классов-инструментов, и постоянно отрисовывает их в порядке появления в списке.

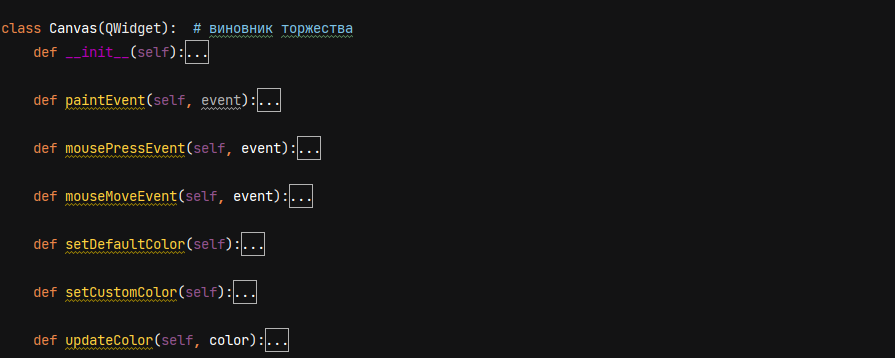
Немного о самом устройстве классов – инструментов для рисования. Brush, Pencil и Eraser – схожие по реализации классы, рисующие линию с постоянно обновляющимися точками начала и конца. Line обновляет только конечную точку, из-за чего реализация похожа на аналог из MS Paint. Классы многоугольников тоже являются аналогами «пейнтовских» (пришла идея считать координаты по прямоугольнику, выделяющему фигуру), однако есть и изюминка. При разрешении на заливку MS Paint рисует контур по основному цвету и заливку по дополнительному, но проект разрешает менять цвета при заливке (если выбранный цвет – дополнительный, то контур дополнительного цвета, заливка – основного, и наоборот). Классы Fill и Image рисуют поверх всего холста, но Fill – заливку активным цветом, Image – выбранную пользователем картинку. И класс Save – сохранение всего нарисованного на холсте. По сути, создает пустую картинку (объект класса QImage), соразмерную холсту, и после выбора пользователем места и формата файла для сохранения переносит все ранее нарисованные объекты на эту самую картинку и сохраняет её.

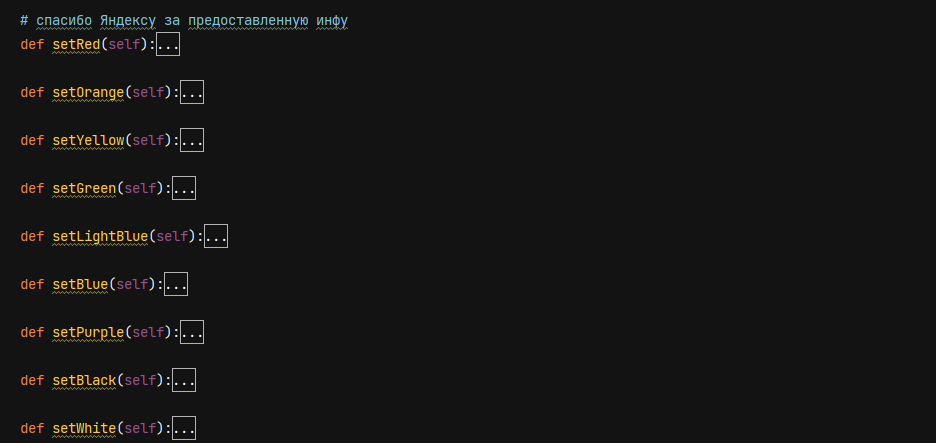
Для работы требуется Python версии 3 и выше и библиотеки PyQt5 – PyQt5, PyQt5-Qt5, PyQt5-stubs и PyQt5-sip. Чтобы не иметь проблем с версиями, в файле проекта **requirements.txt** уже прописаны все необходимые невстроенные библиотеки, которые можно установить по следующему алгоритму:

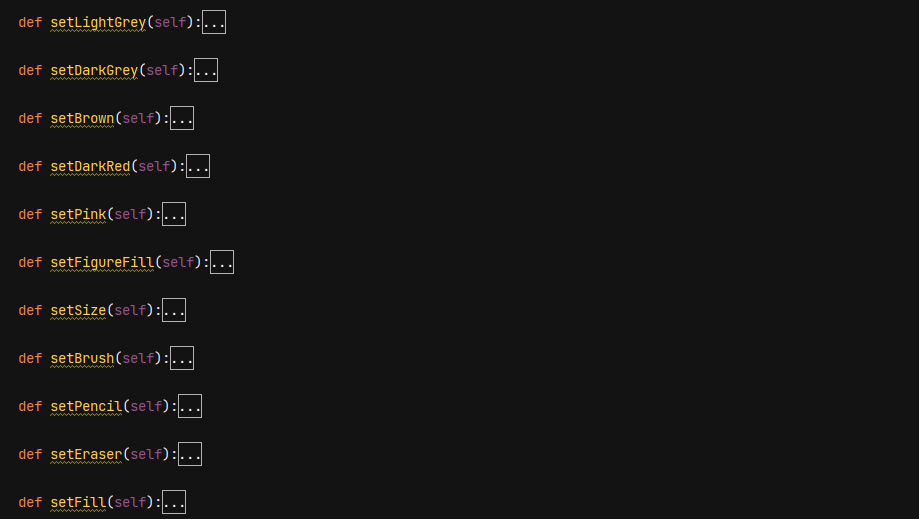
1. Открыть командную строку или терминал
2. Дойти до папки с проектом
3. Прописать команду **pip install –r requirements.txt**
4. Дождаться установки



Импортированные библиотеки



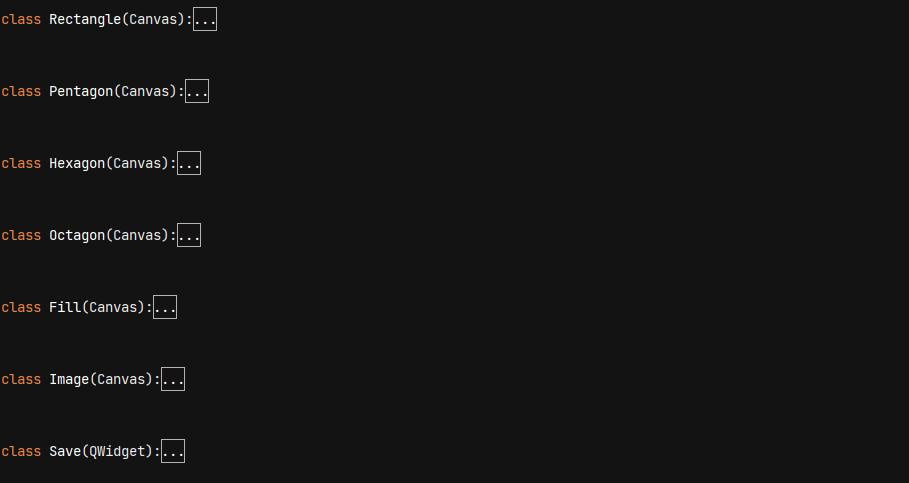




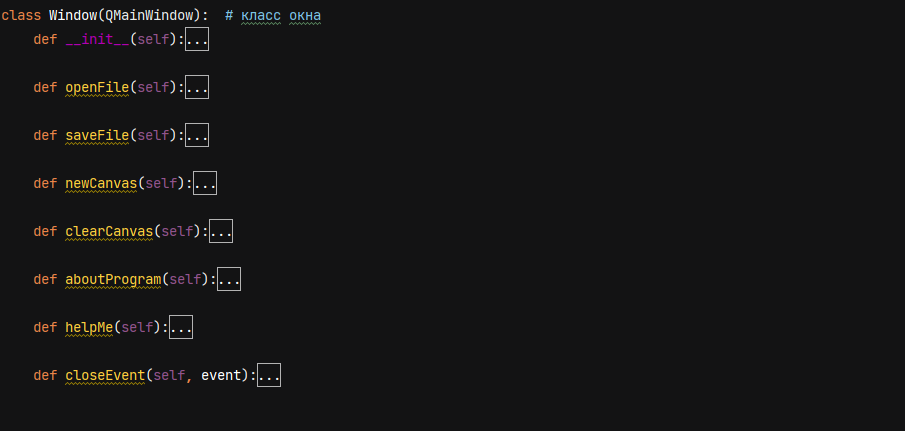


Класс Canvas и его методы

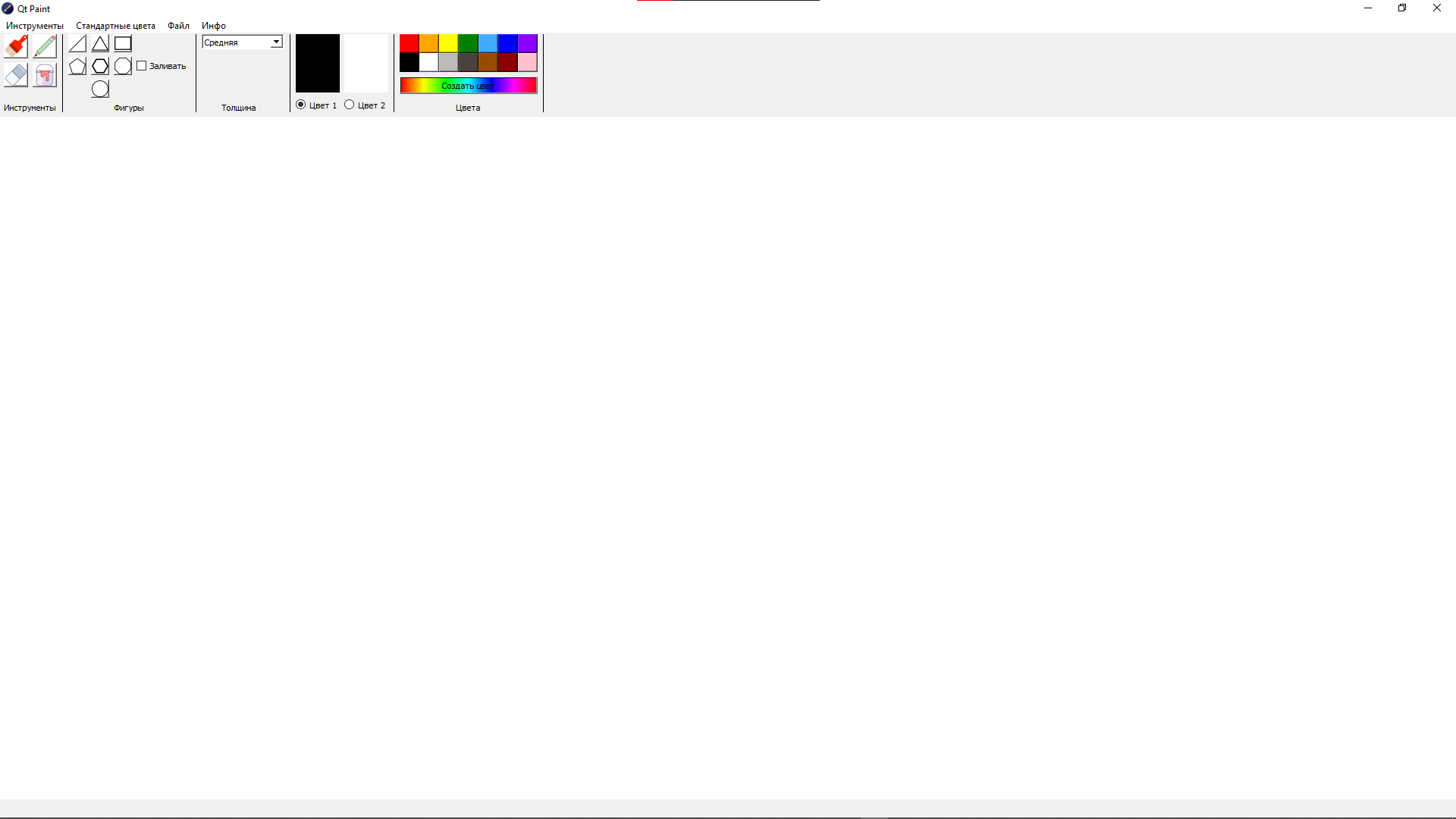




Классы – инструменты



Класс окна и его методы



Окно приложения в готовности