

«Национальный исследовательский университет  
«Высшая школа экономики»  
Лицей

Индивидуальная выпускная работа

**ИТ-ПРОЕКТ: Простой в освоении растровый графический редактор с  
функцией работы со слоями**

[https://github.com/maratbekzenteev/ivr\\_project](https://github.com/maratbekzenteev/ivr_project)

*Выполнил: Бекзентеев Марат Рустамович*

*Группа: 11И2*

Консультант:

Палеев Даниил Алексеевич

Москва – 2023

## **1. Введение**

Выбор предметной области проекта был обусловлен, во-первых, интересом к графическим редакторам с точки зрения пользователя и большим опытом работы как в MS Paint, так и в Adobe Photoshop и GIMP, во-вторых, предыдущим опытом разработки (в учебных целях) веб- и desktop-приложений с использованием flask и PyQt 5 соответственно. Интерес к созданию графического редактора объясняется желанием дополнить и улучшить вышеперечисленные имеющиеся на рынке решения.

## **2. Проблемное поле**

Проблема, решаемая проектом, заключается, во-первых, в отсутствии в функциональности имеющихся на рынке графических редакторов одной или нескольких из следующих возможностей: работа с многослойными изображениями (в том числе работа с каждым из слоёв по отдельности), обработка прозрачных и полупрозрачных изображений (такими считаются изображения, содержащие пиксели со значением альфа-канала, не равным 255), выравнивание содержимого слоёв по сетке, во-вторых, в недостаточной простоте и интуитивности интерфейса профессиональных графических редакторов, который, как правило, включает в себя несколько подвижных панелей инструментов и множество элементов, с которыми пользователь не может работать одновременно), что делает их недоступными для множества пользователей.

## **3. Образ продукта**

Финальным продуктом являются два приложения, отвечающие за две составляющие проекта: клиентскую, которая запускается на компьютере пользователя, и серверную, работающую на сервере либо локально и отвечающая на запросы пользователя по получению пользовательских проектов из базы данных и их сохранению в неё.

К клиентской части проекта предъявлены следующие функциональные требования:

- Создание, удаление слоёв, их перемещение друг относительно друга, перемещение их содержимого по плоскости (последнее в финальной версии проекта не было реализовано для растровых слоёв, но было реализовано для остальных типов слоёв)),
- Выделение части слоя (в финальной версии проекта реализовано не было),
- Рисование кистью на слоях (реализовано в рамках функциональности растровых слоёв),
- Заливка выделенных участков слоёв (реализована в рамках функциональности растровых слоёв),
- Создание текстовых слоёв и работа с ними,
- Создание фигурных слоёв, то есть слоёв, предназначенных для размещения на них фигур и выравнивания фигур по сетке, и работа с ними (требование изначально не предъявлялось),
- Создание пользовательской сетки и выравнивание содержимого слоёв по ней (было реализовано для слоёв-картинок, фигурных и текстовых слоёв, изначально не предполагалось к реализации для растровых слоёв),
- Экспорт изображений в форматах PNG, BMP, GIF (в финальной версии проекта реализован экспорт в форматах PNG, BMP, JPG),
- Сохранение файлов на компьютер пользователя и открытие файлов с него.

Вышеуказанные требования позволяют проекту быть удобным в использовании, вкупе с интуитивным интерфейсом, автоматически скрывающим ненужные пользователю элементы (например, если пользователь работает с растровым слоем, на панели инструментов показываются только инструменты для работы с

растровым слоем, сеткой и всем проектом в целом), позволяют пользователю быстро и эффективно использовать программу, не тратя времени на её освоение.

К серверной части проекта предъявлены следующие требования:

- Регистрация и авторизация пользователя из программы-клиента,
- Смена пароля пользователя (требование изначально не предъявлялось),
- Сохранение проекта в аккаунт пользователя после авторизации,
- Получение списка проектов пользователя из аккаунта после авторизации,
- Открытие проекта из списка после авторизации.

Вышеуказанные требования позволяют пользователю работать со своими проектами, переключаясь между несколькими компьютерами, что увеличивает скорость и эффективность работы в программе.

#### **4. Сделанный продукт**

Сделанный продукт рассчитан на следующие сценарии использования:

- Работа с проектом целиком. Включает в себя создание проекта с нуля с указанием целевого разрешения, открытие проекта из файла с компьютера пользователя, открытие проекта из аккаунта пользователя, сохранение проекта в файле на компьютер пользователя, сохранение проекта в аккаунте пользователя, изменение размера выходного файла проекта с указанием нового размера, экспорт проекта в файлы изображений форматов PNG, BMP, JPG.
- Работа с растровыми слоями (в терминологии программы они носят название «холсты» по своей возможности рисовать). Растровые слои – слои, на которых пользователь может редактировать каждый пиксель в отдельности, рисуя на слое кистями различных начертаний, чертя отрезки, прямоугольники, окружности, заливая области. Растровые слои поддерживают прозрачность.
- Работа с сеткой. На сетку допускается добавлять вертикальные и горизонтальные прямые, деля тем самым плоскость проекта на прямоугольники, в которые можно

помещать различные типы объекты: текст, слои, картинки. Прямая сетки задаётся либо абсолютно (в пикселях), либо относительно (в процентах от ширины/высоты изображения проекта в зависимости от направления прямой сетки), в таком случае прямая меняет своё положение при смене разрешения изображения проекта.

- Работа со слоями-картинками. Слойми-картинками в терминологии программы именуются слои, на которые можно вставлять какое-то изображение целиком без возможности редактирования, а затем перемещать его содержимое по плоскости проекта, масштабировать, помещать в какой-либо прямоугольник сетки и выравнивать внутри него (выравнивать по краям / по центру / заполнять изображением весь прямоугольник сетки).
- Работа с фигурными слоями. Фигурный слой – слой, содержащий векторную фигуру – ей может быть отрезок, прямоугольник, эллипс – которую пользователь может привязывать к узлам сетки, тем самым выравнивая её, или перемещать по плоскости проекта.
- Работа с текстовыми слоями. Текстовый слой – слой, на котором пользователь может печатать форматированный текст (пользователь может менять цвет, шрифт и размер отдельных его символов, делать их полужирными, курсивными и/или подчёркнутыми, выравнивать их по левому краю / по центру / по правому краю / по ширине). Текст помещается пользователем в любой прямоугольник сетки.
- Взаимодействие пользователя с сервером. Включает в себя регистрацию, авторизацию, смену пароля, выбор проекта из списка для последующего открытия, открытие и сохранение проекта на сервер.

## **5. База данных**

В качестве базы данных backend-части проекта (фактически его серверной части) выступает нереляционная база данных `mongoDB`. Выбор пал на неё в силу её структуры, а именно записей («документов»), представляющих собой JSON-

подобные файлы. Эта особенность позволяет хранить файлы проектов графического редактора без внесения каких-либо изменений, так как они сами по себе имеют JSON-подобную структуру. Сама база данных имеет две таблицы («коллекции»), users – коллекция пользователей, каждая запись содержит в себе имя пользователя и его пароль, зашифрованный по алгоритму SHA256, projects – коллекция проектов, каждая запись – проект, содержит информацию о пользователе (имя, пароль) и данные проекта (имя, разрешение, данные о сетке и слоях).

## **6. Реализация проекта**

И клиентская, и серверная часть проекта были написаны на Python версии 3.9. Для реализации клиентской части были задействованы библиотеки PyQt5 (графический интерфейс программы), json (сохранение, открытие и обработка JSON-файлов, полученных в результате работы программы либо из файловой системы), requests (HTTP-запросы к серверной части проекта, в том числе с JSON-аргументами). PyQt 5 была выбрана в качестве GUI-библиотеки как единственная, известная мне, и как лучшая из подобных библиотек для Python; json и request обладают всем необходимым для проекта функционалом и минималистичны в синтаксисе.

Для реализации серверной части использованы библиотеки flask (создание самого веб-приложения, обработка запросов клиентской части), pymongo (взаимодействие с mongoDB-базой данных, в которой хранятся данные о пользователях и их проектах), hashlib (для хеширования паролей пользователей по алгоритму SHA256). flask был выбран, так как серверная часть проекта не настолько велика, чтобы реализовывать её на django или .NET Framework, вдобавок к этому, обладает минималистичным синтаксисом. pymongo был выбран как единственная официальная реализация mongoDB-клиента для Python.

Из инструментов для разработки / сервисов были использованы PyCharm Community Edition (написание проекта, взаимодействие с системой контроля версий, вёрстка Markdown-файлов), Qt Designer (набросок интерфейса программы на раннем этапе проектирования), reqbin.com (отладка серверной части проекта), Microsoft To Do (организация невыполненных задач по проекту), документация Qt.

## **7. Этапы работы**

Хронологическое разбиение проекта на этапы примерно соответствует финальному разбиению проекта на пользовательские сценарии: сначала реализовывалась функциональность растровых слоёв, затем работа с несколькими слоями, слои-картинки, фигурные слои, текстовые слои, серверная функциональность, затем было проведено комментирование кода и его рефакторинг, после чего была улучшена графическая составляющая программы (нарисованы иконки). На каждом этапе так или иначе дорабатывался функционал предыдущих этапов: исправлялись ошибки, добавлялись новые методы классов, совместимость с только что написанными элементами программы. Один из самых ярких примеров – виджет списка слоёв, дорабатывавшийся на протяжении всего времени работы над клиентской частью проекта.

## **8. Рефлексия**

В процессе работы над проектом возникла проблема правильного планирования времени, так как проект оказался больше и тяжелее в реализации, чем предполагалось изначально. Проблему решило увеличение темпов работы над проектом и отслеживание прогресса в Microsoft To Do. Вторая возникшая проблема – неудобство при работе с PyQt в силу громоздкости библиотеки и её синтаксиса, а также неочевидности реализации некоторых функций. Проблема была решена прочтением нужных статей из документации библиотеки.

Продукт в перспективе дальнейшего развития следует перенести на другую библиотеку, а значит, на другой язык (предположительно, C# или Kotlin). Также стоит отметить, что в текущей версии лишь частично реализованы некоторые функции, которые можно было бы дополнить до изначально заявленного состояния. Помимо этого, проект мог бы пройти косметическую доработку, став ещё более удобным для использования.

В процессе работы над проектом были приобретены навыки планирования времени, работы с документацией, планирования структуры проекта (а значит, и кода) на будущее, создания пользовательских интерфейсов, работы с нереляционными базами данных.

Риски, обозначенные в проектной заявке, частично оправдали себя: программа работает несколько медленнее, чем аналоги, написанные на C++ (особенно можно отметить получение ответов на запросы от сервера и заливку участков слоёв), при этом качество выходного изображения, вопреки названному риску, получается на уровне программ-конкурентов.

## **9. Заключение**

Работа над проектом, несмотря на сложности при разработке и большую его величину, позволила мне приобрести множество полезных навыков и создать действительно применимый на практике продукт, имеющий ценность и дальнейшие перспективы.