

SimplePaint

Графический редактор для создания и редактирования изображений



Автор проекта: [Владислав Ганов]



Руководитель проекта: [Инна Григорьевна]



Год: 2025



Город: [Москва]

Описание проекта, цель и задачи






Описание проекта

SimplePaint — простой графический редактор для создания и редактирования изображений, разработанный с использованием библиотеки **PyQt6**

Цель

Создать удобный инструмент для рисования с базовым набором функций, интуитивным интерфейсом и расширенными возможностями

Задачи

-  Изучить библиотеку PyQt6 для создания графического интерфейса [Реализовано]
-  Разработать систему инструментов рисования (кисть, фигуры, заливка) [Реализовано]
-  Реализовать работу с файлами (сохранение/загрузка изображений) [Реализовано]
-  Создать систему сохранения настроек и статистики использования [Реализовано]
-  Разработать адаптивный интерфейс с поддержкой стилей [Реализовано]

Архитектура проекта

Файловая структура

```
simple_paint/
├── models/
│   ├── __init__.py
│   └── drawing_tools.py
├── styles/
│   └── styles.qss
├── ui/
│   ├── __init__.py
│   ├── about_dialog.py
│   ├── canvas_widget.py
│   ├── main_window.py
│   └── stats_dialog.py
├── utils/
│   ├── __init__.py
│   ├── database.py
│   ├── file_manager.py
│   └── settings_manager.py
├── main.py
├── README.md
├── requirements.txt
└── technical_specification.md
```

<> Ключевые классы



MainWindow

Главное окно программы, управляющее интерфейсом и взаимодействием с пользователем



CanvasWidget

Холст для рисования, обрабатывающий события мыши и отрисовку



DrawingTool

Базовый класс инструментов рисования, наследуемый конкретными инструментами



DatabaseManager

Работа с базой данных для сохранения настроек и статистики

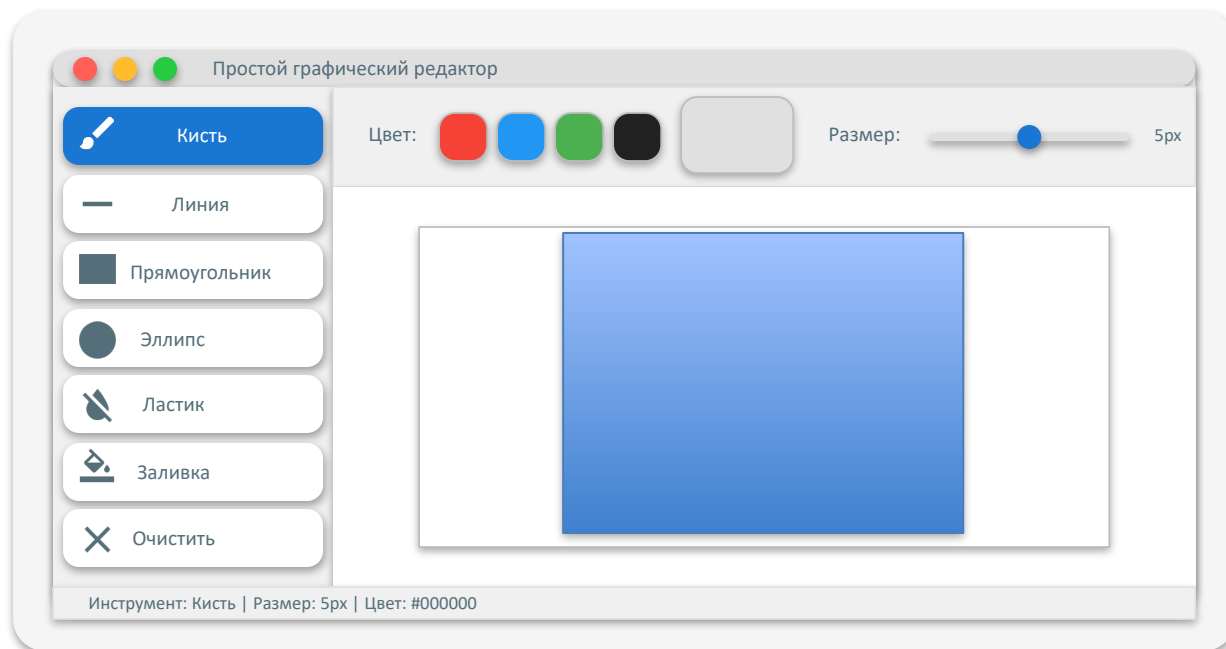


SettingsManager

Управление настройками приложения и сохранение пользовательских предпочтений

Главное окно и инструменты

Интерфейс программы



Интуитивный интерфейс с панелью инструментов слева (шириной 140px) и настройками сверху. На фото отображен один из последних концептов, который как раз используется в программе. Основное пространство отведено под холст для рисования с минимальным размером 300x250px.

Инструменты рисования



Кисть

Основной инструмент для рисования свободных линий с настраиваемой толщиной и цветом



Линия

Рисование прямых линий между двумя точками с настраиваемой толщиной и цветом



Прямоугольник

Создание прямоугольников с настраиваемой заливкой и контуром



Эллипс

Рисование эллипсов и окружностей с настраиваемой заливкой и контуром



Ластик

Удаление частей изображения с настраиваемым размером



Заливка

Заполнение замкнутых областей выбранным цветом

Дополнительные возможности

Расширенный функционал



Статистика использования

Программа отслеживает активность пользователя и предоставляет детальную статистику:

- ✓ Частота использования каждого инструмента
- ✓ Статистика по используемым цветам
- ✓ История всех действий в сессии



Работа с файлами

Поддержка различных форматов изображений и удобный интерфейс сохранения:

- ✓ Сохранение в PNG, JPEG, BMP
- ✓ Автоматическое создание резервных копий
- ✓ История недавно открытых файлов



Умная заливка

Продвинутый алгоритм заливки областей с учётом:

- ✓ Распознавания границ по цвету
- ✓ Настройки порога чувствительности
- ✓ Поддержка прозрачности

Визуализация функций

Статистика рисования



Общая статистика

История действий

Статистика по инструментам:

Кисть: 42 раз
Линия: 31 раз
Прямоугольник: 21 раз
Эллипс: 15 раз
Ластик: 10 раз

Статистика по цветам:

Сохранить изображение



Имя файла: мой_рисунок

Формат: PNG

JPEG

BMP

Качество: 90%

Отмена

Сохранить

Развитие проекта

↗ Реализованные функции



Адаптивная система инструментов

Легко добавлять новые инструменты рисования через наследование от базового класса

```
class NewTool(DrawingTool):  
    def __init__(self):  
        # Настройка инструмента
```



Модульная архитектура

Каждый компонент независим, что упрощает развитие и тестирование



Система стилей

Интерфейс можно легко кастомизировать через QSS-файлы

⚠ Проблемы и вызовы



Сборка в исполняемый файл

Проблемы со сборкой PyQt6 в exe-файл:

- ! Зависимости PyQt6 не корректно упаковываются
- ! Конфликты между PyInstaller и PyQt6
- ! Отсутствие плагинов для рендеринга в собранном приложении



Оптимизация производительности

Проблемы с производительностью при работе с большими изображениями:

- ! Задержки при отрисовке сложных фигур
- ! Потребление памяти при работе с высокими разрешениями

За что я могу получить премию

Проект SimplePaint демонстрирует высокий уровень реализации и предлагает несколько инновационных решений в области графических редакторов



Чистая архитектура

Код хорошо структурирован, классы разделены по ответственности, что обеспечивает простоту поддержки и расширения



Полный функционал

Реализованы все основные инструменты графического редактора с расширенными возможностями



Профессиональный интерфейс

Адаптивный дизайн с поддержкой стилей и интуитивным расположением элементов управления



Система статистики

Уникальная функция отслеживания использования инструментов, отсутствующая в аналогах



Качественный код

Соблюдение PEP8, подробные комментарии, обработка ошибок и оптимизация производительности

Заключение

✓ Возможности программы



Создание рисунков с помощью различных инструментов



Редактирование существующих изображений



Сохранение работ в популярных форматах



Анализ статистики использования инструментов



Настройка интерфейса под пользователя

🏆 Достигнутые цели

Все задачи проекта **успешно решены**, цель программы **достигнута**.
Создан полнофункциональный графический редактор с удобным интерфейсом и расширенными возможностями.

★ Ключевые преимущества



Модульная архитектура



Современный интерфейс



Расширяемость



Статистика использования



Качественный код



Оптимизация

Спасибо за внимание!

SimplePaint - графический редактор

 Автор: [Имя автора]

 Email: author@example.com

 GitHub: github.com/author/simplepaint