Министерство образования и науки Тамбовской области

Комитет образования администрации города Тамбова Тамбовской области

Муниципальное автономное общеобразовательное учреждение «Лицей №14

имени Заслуженного учителя Российской Федерации А.М. Кузьмина»

**Индивидуальный проект**

**Интерактивное приложение для работы с графами.**

Автор работы:

*ученик 10 класса К*

*Булгаков Дмитрий*

Руководитель:

*Вязовов С.М.,*

*Заведующий кафедрой информатики,*

*учитель информатики*

**ТАМБОВ 2024**

Оглавление

[Введение. Страница - 2](#_Toc194705503)

[Постановка задачи. Страница - 3](#_Toc194705504)

[Основа приложения. Страница - 3](#_Toc194705505)

[Задачи разработки. Страница - 3](#_Toc194705506)

[Проектирование интерфейса. Страница - 4](#_Toc194705507)

[Реализация функционала. Страница - 5](#_Toc194705508)

[Ход работы. Страница - 6](#_Toc194705509)

[Тестирование и отладка. Страница - 15](#_Toc194705510)

[Заключение. Страница - 15](#_Toc194705511)

[Используемые источники. Страница - 16](#_Toc194705512)

[Приложения. Страница - 16](#_Toc194705513)

**Введение**

В условиях современного развития информационных технологий и широкого применения алгоритмических методов умение работать с графами приобретает особую значимость. Графы используются в сетевых технологиях, социальном анализе, логистике и многих других областях. Целью данного проекта является разработка графического редактора «Graf++», предоставляющего пользователю возможность визуального создания, редактирования и анализа графов.

Проект реализован на языке Python с использованием библиотеки PyQt6, что обеспечивает кроссплатформенность и гибкость интерфейса. Разработанный инструмент может использоваться как в образовательных целях, так и в качестве основы для дальнейших исследовательских и практических задач.

**Постановка задачи**

**Основа приложения**

На начальном этапе были определены следующие основные требования:

* **Интерактивное создание графа.** Возможность добавления вершин (точек) и соединения их рёбрами посредством щелчков мыши.
* **Редактирование элементов.** Реализация функций изменения цвета, размера, меток вершин и рёбер, а также параметров, таких как вес ребра.
* **Алгоритмы обхода графа.** Интеграция алгоритмов обхода графа: поиск в ширину (BFS) и поиск в глубину (DFS) с пошаговой демонстрацией.
* **Сохранение и загрузка.** Возможность экспорта текущего состояния графа в файл (формат JSON) и последующего восстановления для продолжения работы.
* **Удобство использования.** Создание интуитивно понятного и эстетически привлекательного интерфейса, позволяющего легко переключаться между режимами работы.

**Задачи разработки**

* Ознакомится и подробнее изучить PyQt6 [Выполнена]
* Создать интерфейс для работы с графами [Выполнена]
* Реализовать функционал добавления и удаления вершин и рёбер [Выполнена]
* Реализовать представление графа в виде: матрицы смежности/списка рёбер/списка смежности [Выполнена]
* Создать систему сохранения и загрузки графов [Выполнена]
* Добавить алгоритмы для графов
* Внедрить визуализацию шагов выполнения алгоритмов
* Провести тестирование всех функций приложения

### 

### Проектирование интерфейса

При проектировании интерфейса особое внимание уделялось удобству и функциональности:

* **Главное окно приложения.** Содержит рабочую область для рисования графа, панель инструментов для выбора режимов и боковую панель с подсказками.
* **Панель инструментов.** Расположена в верхней части окна и включает кнопки для добавления вершин, рисования рёбер, удаления объектов, изменения цвета и размера элементов.
* **Боковая панель.** Предоставляет пользователю информацию о текущем режиме, а также кнопки для запуска алгоритмов обхода (BFS, DFS) и управления настройками.
* **Контекстное меню.** Позволяет изменять параметры выбранных объектов (цвет, размер, вес рёбер) посредством удобного меню, вызываемого правым кликом.

Такой дизайн позволяет быстро и без предварительного обучения приступить к работе с приложением.

**Реализация функционала**

***1. Алгоритмы обхода графа***

В проекте реализованы два основных алгоритма обхода:

* **Поиск в ширину (BFS):**  
  Используя очередь, алгоритм последовательно посещает все смежные вершины, начиная с выбранной пользователем. Каждый шаг обхода сопровождается выводом информации, позволяющей визуально отследить процесс.
* **Поиск в глубину (DFS):**  
  Реализован через рекурсивную функцию, позволяющую глубоко обходить структуру графа до тех пор, пока не будут посещены все достижимые вершины. На каждом этапе выполнения происходит обновление интерфейса и вывод текущего состояния.0

***2. Рисование и редактирование графа***

Основной функционал редактора включает в себя:

* **Создание вершин:**  
  При клике по рабочей области создаётся вершина, представленная в виде круга с уникальной меткой. Параметры вершины (цвет, размер) можно изменять через контекстное меню.
* **Создание рёбер:**  
  Соединение вершин осуществляется путём выбора двух точек. Ребро автоматически создается между ними. Пользователь может настраивать вес и цвет рёбра для улучшения визуального представления связей.
* **Контекстное меню и настройка параметров:**  
  Для вершин и рёбер предусмотрено контекстное меню, позволяющее изменять их внешние параметры. Это обеспечивает гибкость настройки и повышает удобство работы с графом.

***3. Сохранение и загрузка графа***

Функционал сохранения и загрузки реализован с использованием формата JSON:

* **Сохранение:**  
  Текущее состояние графа, включая позиции вершин, их параметры и связи, экспортируется в JSON-файл.
* **Загрузка:**  
  При загрузке из файла приложение считывает данные и восстанавливает граф с сохранёнными параметрами.

**Ход работы**

Разработка базового интерфейса с главным окном и базовыми элементами управления. (См. Рис.1.)

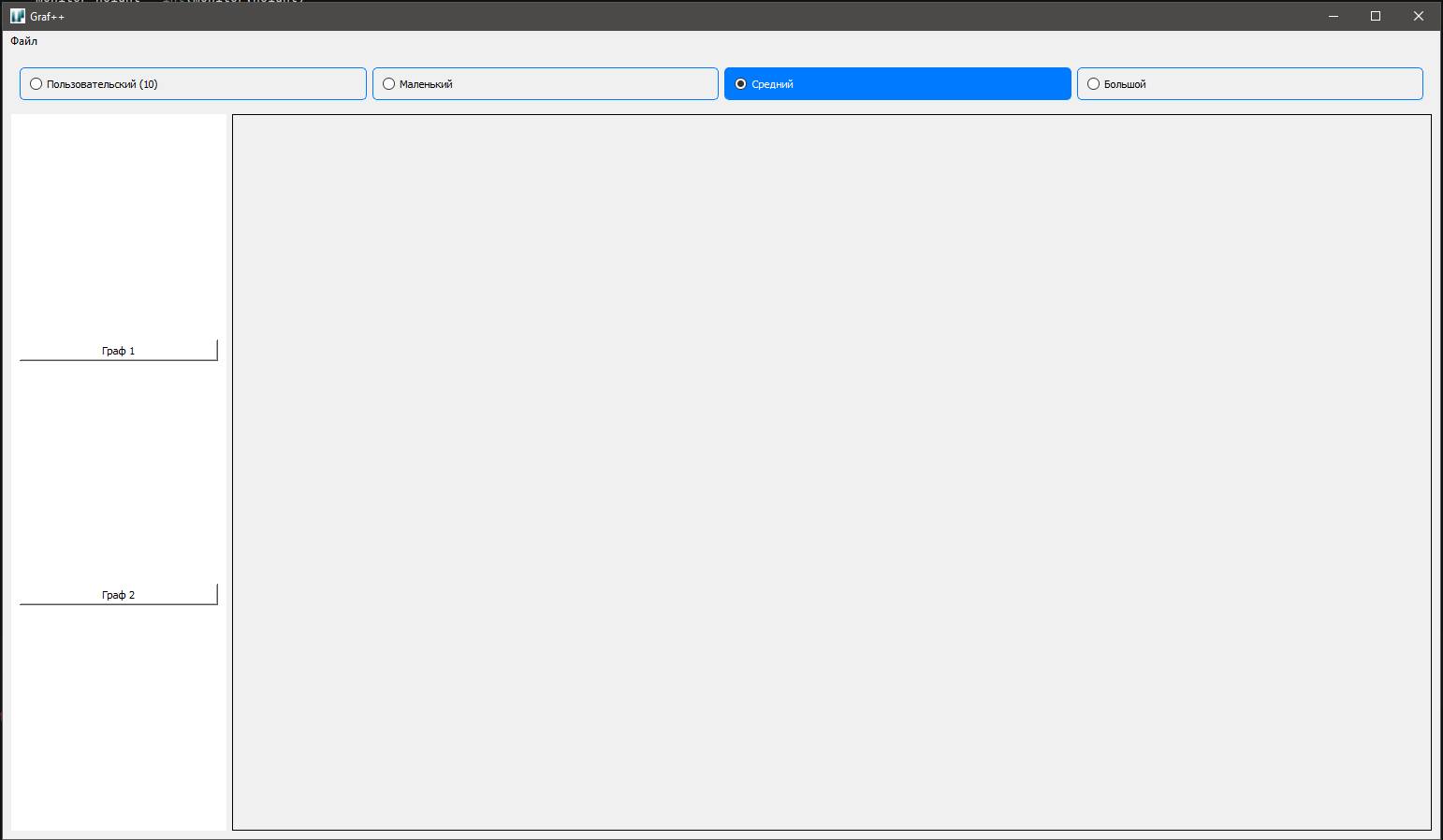


Рис.1.

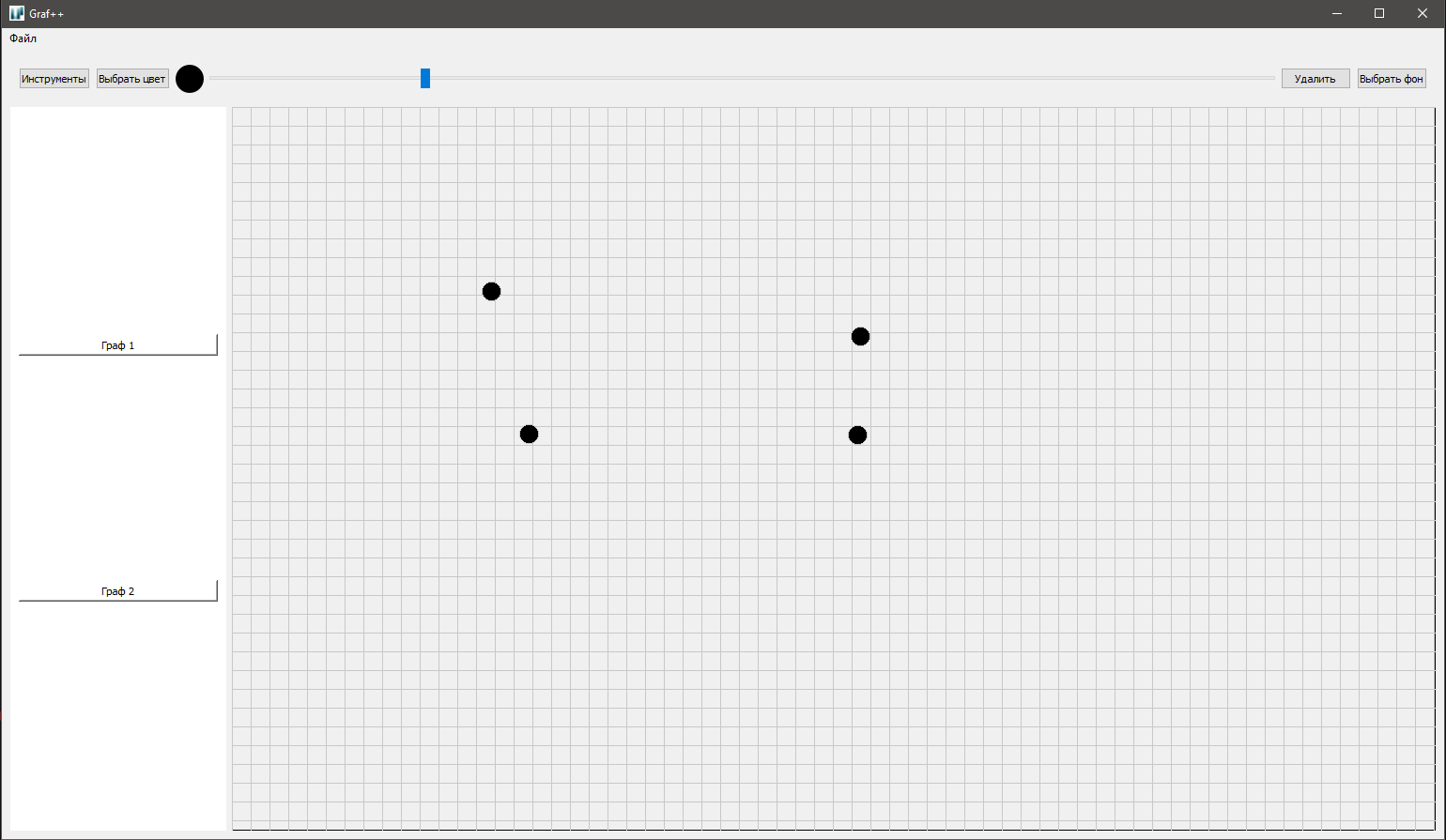
Добавлено: отрисовка кругов(вершин); размер кругов; сетка; улучшен интерфейс. (См. Рис.2.)  


Рис.2.

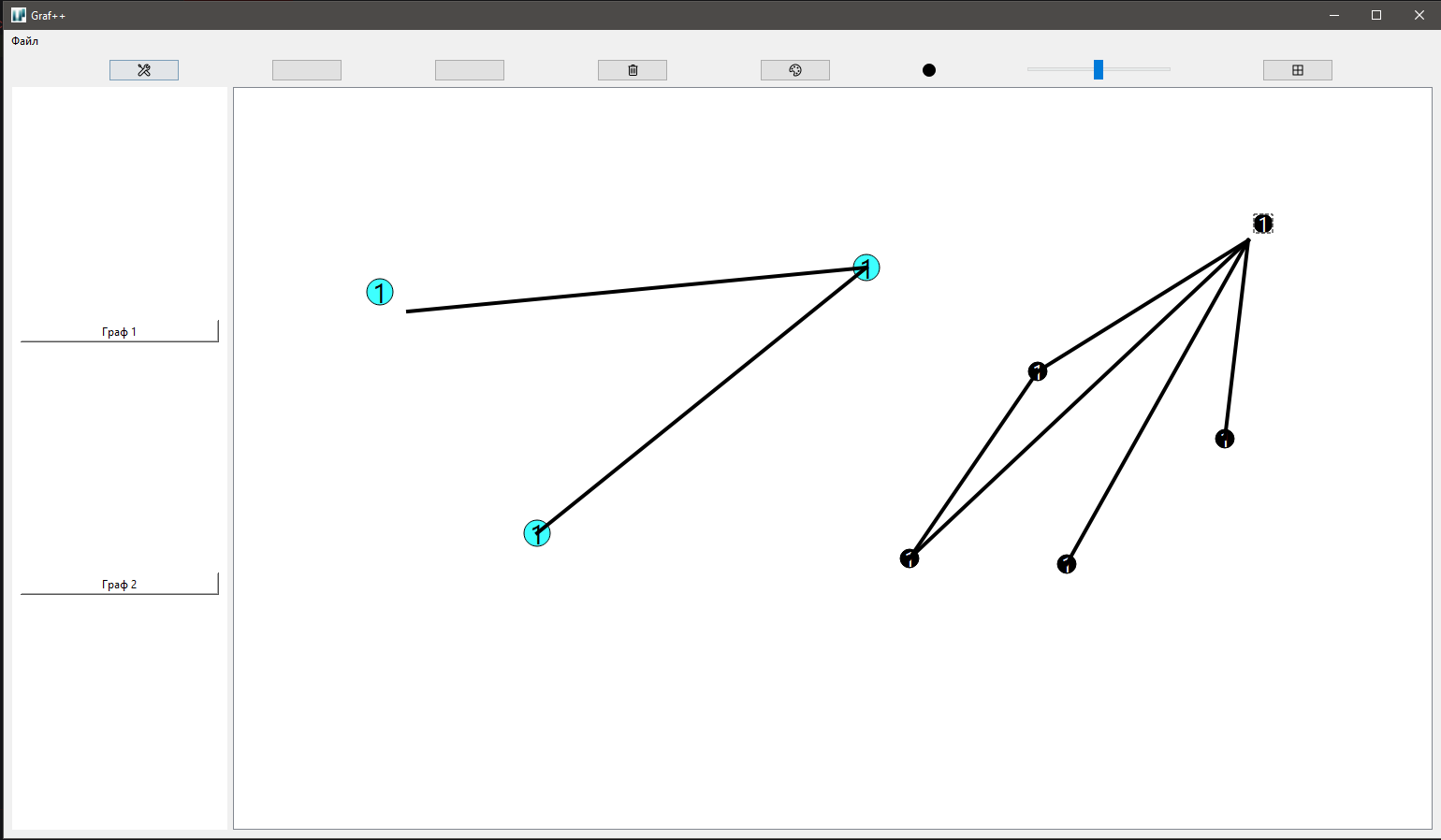
Добавлено: переход на PyQt6; отрисовка линий(рёбер); цвет вершин; убрана сетка из-за большой нагрузки на компьютер; кнопка с svg иконкой; определён вариант конечного интерфейса. (См. Рис.3.)

Рис.3.

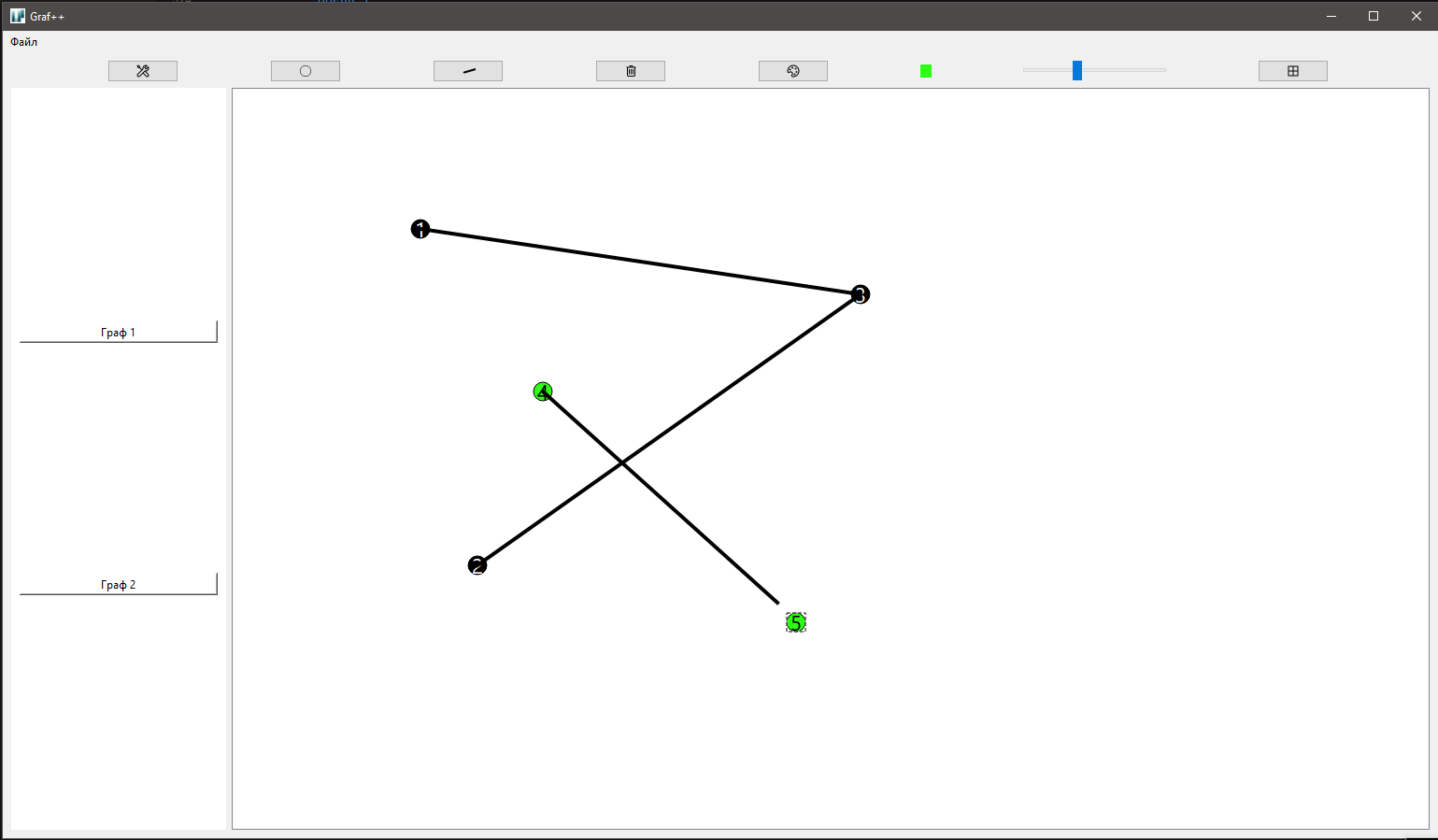
Добавлено: интуитивно понятные иконки; собственный словарь с особой сортировкой для вершин; исправление наложения вершин и рёбер друг на друга(дубликатов). (См. Рис.4.)  


Рис.4.

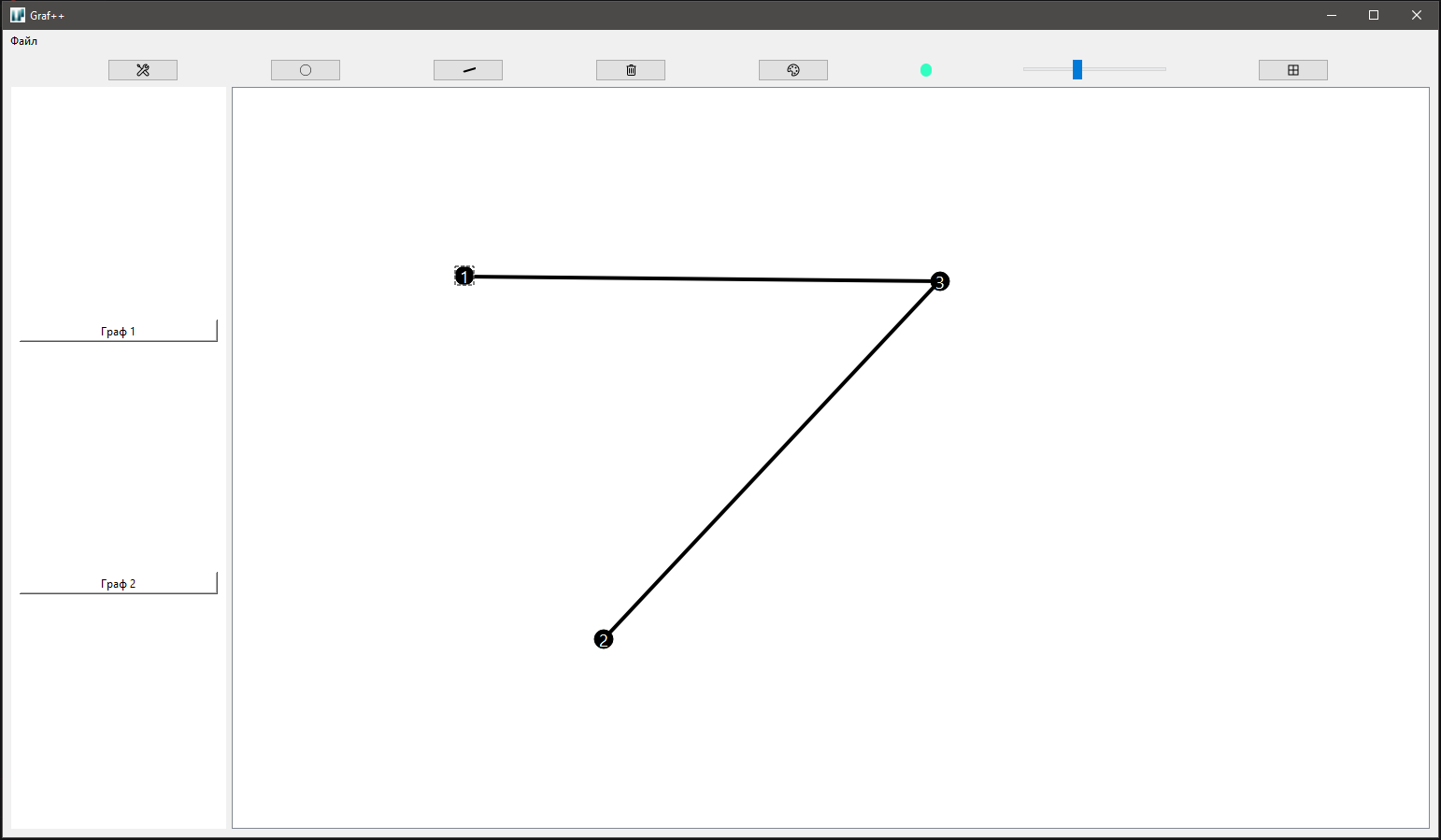
Добавлено: собственные сигналы для вершин и привязка с их помощью положения вершин к положению рёбер; Z-value(”высота”) для объектов на сцене; исправление бага с квадратом цвета. (См. Рис.5.)  


Рис.5.

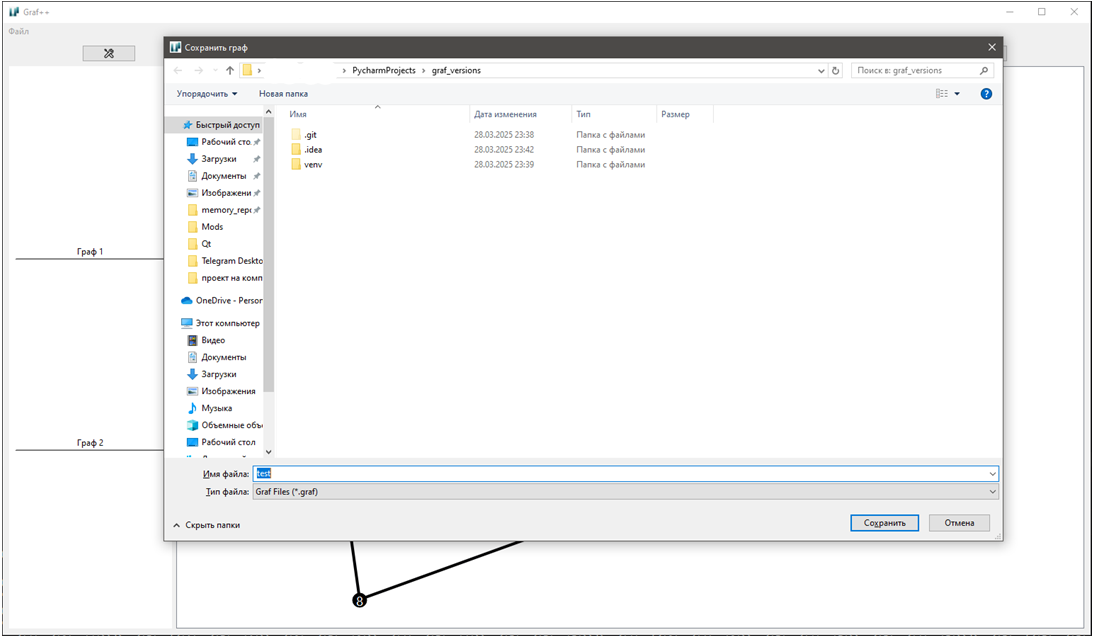
Добавлено: Сохранение и загрузка графов в формат .graf (.JSON); исправление бага с неправильной работой диалоговых окон. (См. Рис.6.)

Рис.6.

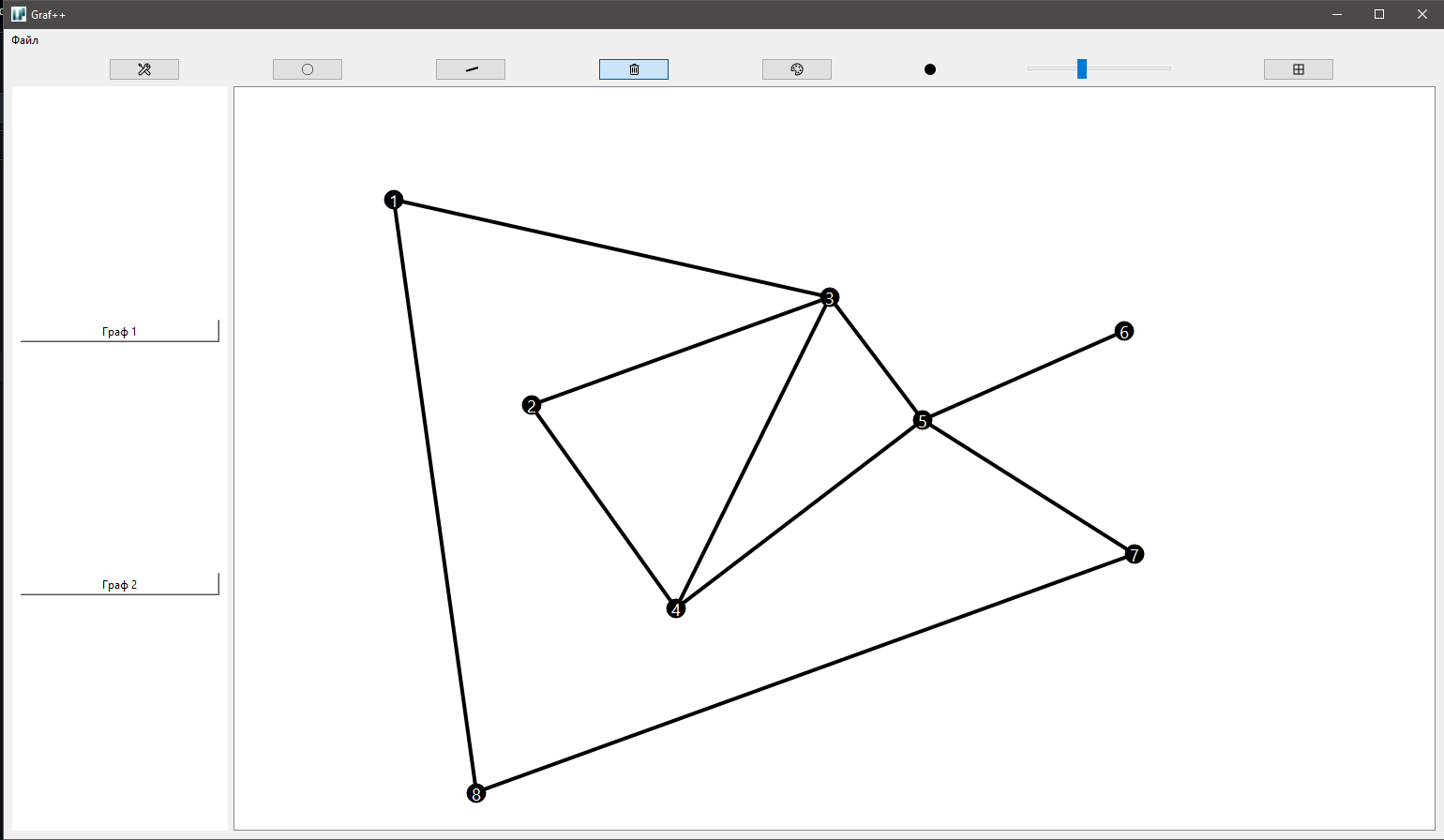
Добавлено: удаление объекта через расчётную область(если нажать в 4px от линии она удалится, не распространяется на клики очень близко с вершинами, так как там происходит наложение областей, приоритет у вершин). (См. Рис.7.)  


Рис.7.

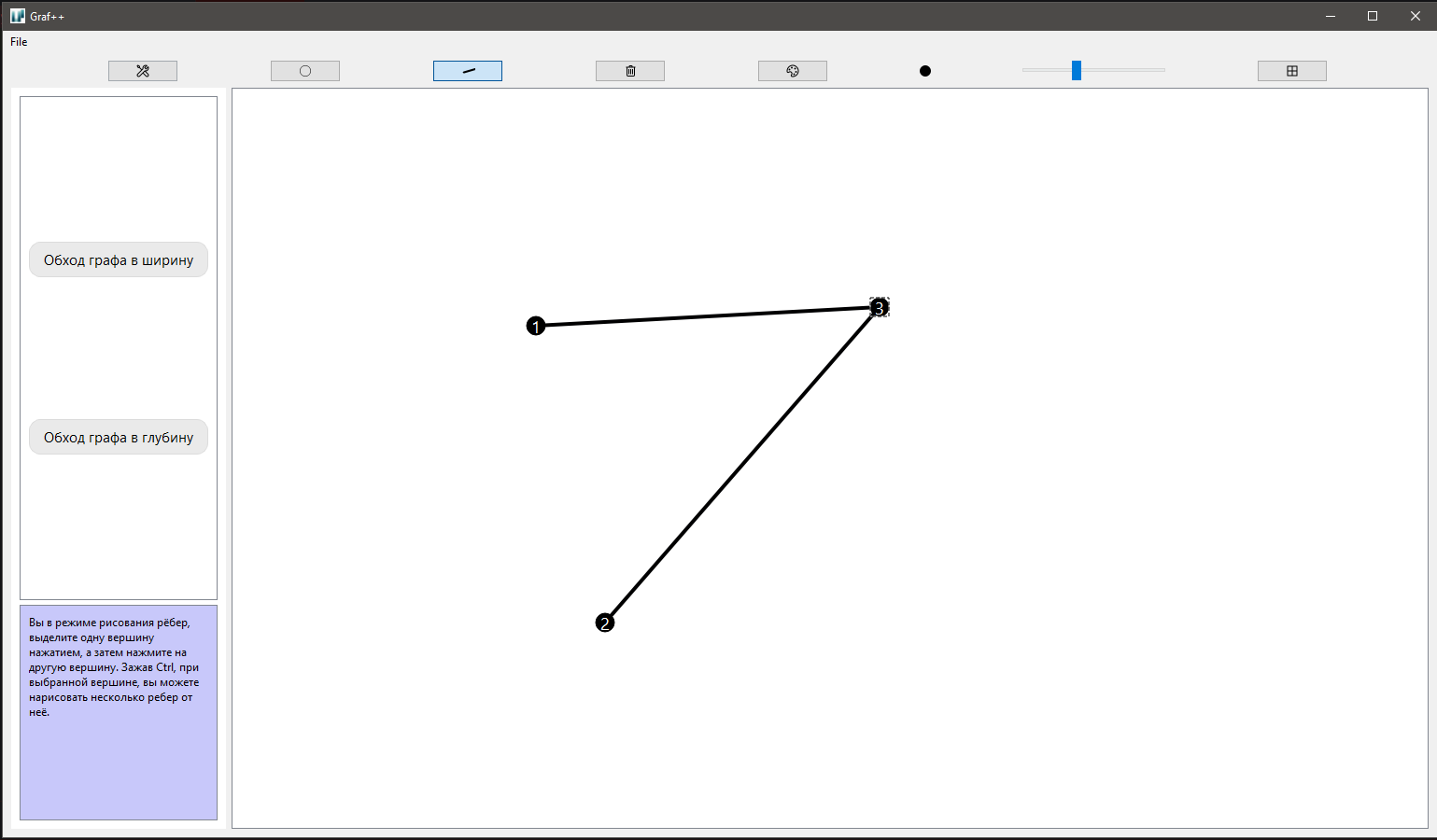
Добавлено: стилизация боковой панели; подсказки в информационном поле; обработка и вывод ошибок в информационном поле. (См. Рис.8.)  


Рис.8.

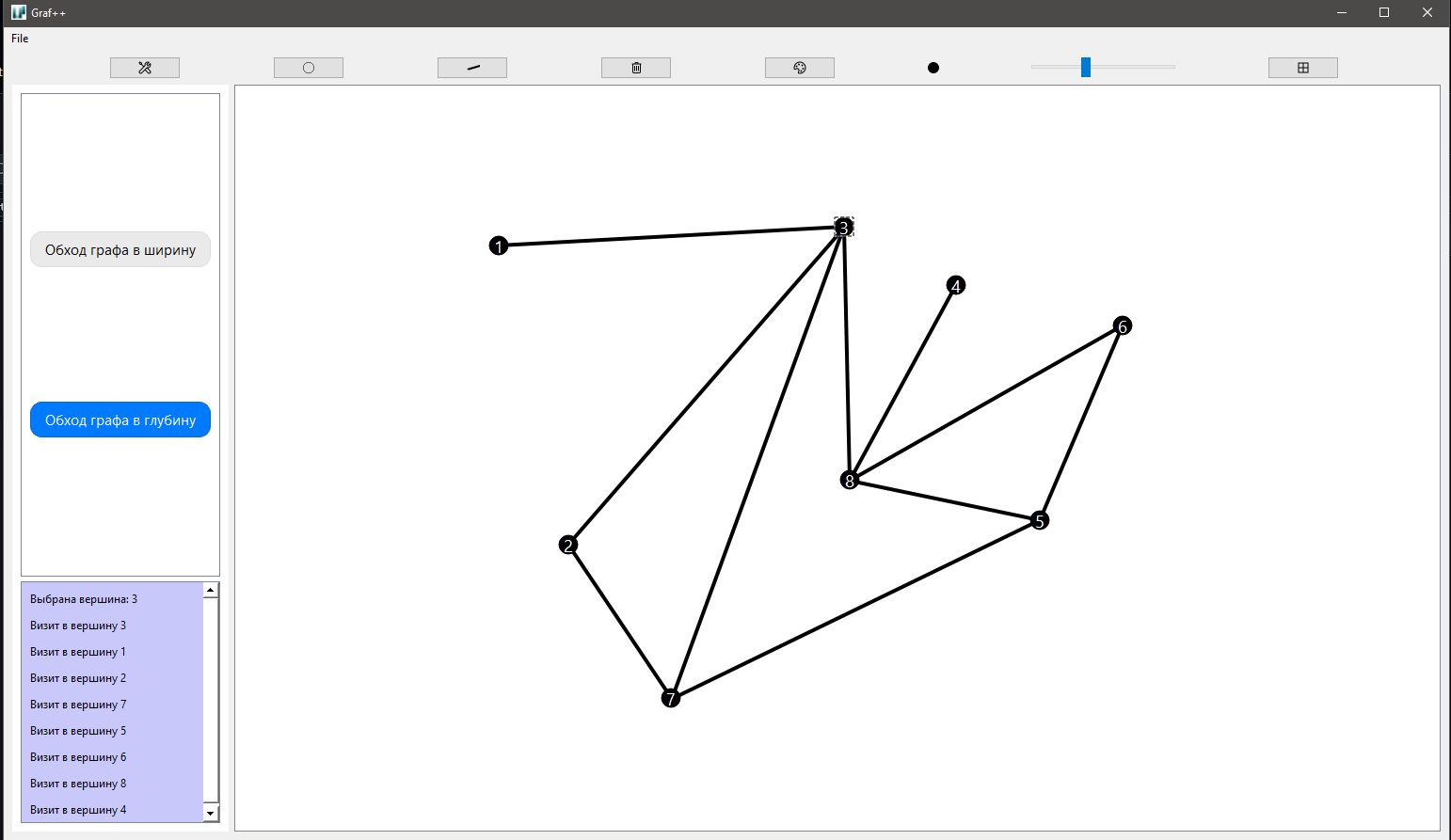
Добавлено: перевод графа во все виды представления для работы алгоритмов; алгоритмы базового обхода. (См. Рис.9.)  


Рис.9.

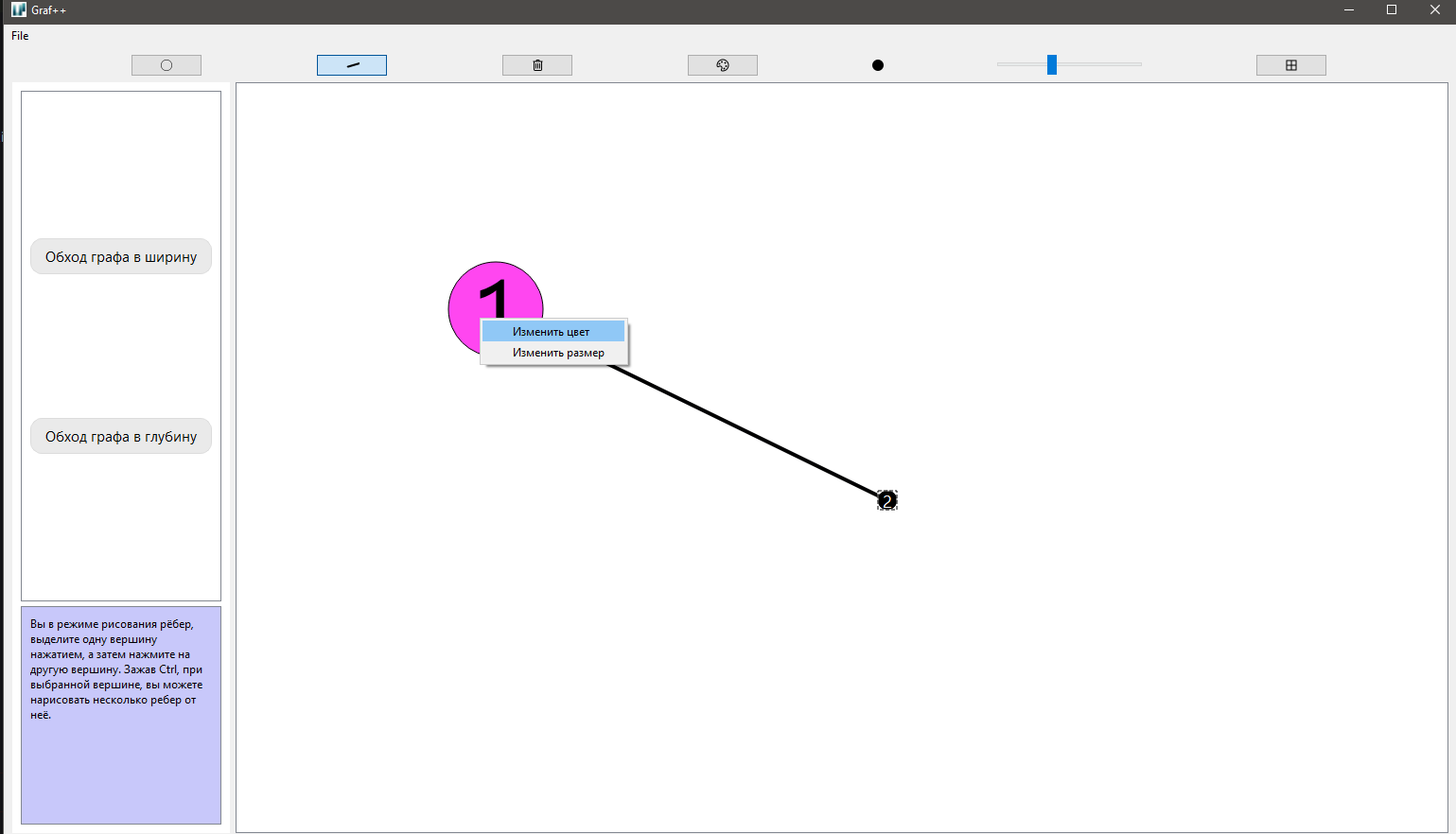
Добавлено: меню конфигурации объектов на правую кнопку мыши; исправление бага выбора объекта. (См. Рис.10.)  


Рис.10.

**Тестирование и отладка**

(Описание тестирования и устранения ошибок. С какими трудностями столкнулся)

* Проблема хранения графа
* Баг с формой, размером и накладыванием вершин:  
  Когда вершины достаточно различаются по размеру, то можно поставить рядом и они наложатся друг на друга, а если поставить рядом с краем, то вершина обрежется.  
  Решение:  
  Пересмотр логики расчёта постановки вершин, создание бесконечного поля.
* Баг с нумерацией:  
  При удалении вершины, иногда возникал пробел в порядке нумерации и цифры повторялись. Также при светлом цвете вершины текст на ней не видно.  
  Решение:  
  Пересмотр логики нумерации и использование

**Заключение**

Проект «Graf++» успешно разработан и реализует основные задачи: создание, редактирование и анализ графов в визуальном интерфейсе. Приложение поддерживает добавление и удаление вершин и рёбер, сохранение и загрузку графов, а также выполнение базовых алгоритмов обхода.

* Создан удобный и интуитивно понятный интерфейс.
* Реализованы основные алгоритмы обхода графа (BFS, DFS).
* Обеспечена возможность сохранения и загрузки данных в формате JSON.
* Улучшена интерактивность за счёт контекстного меню и гибких настроек параметров объектов.
* Проведено тестирование, исправлены ошибки и оптимизирована работа приложения.

#### Возможности для улучшения

* Поддержку ориентированных графов
* Улучшенные визуальные эффекты для наглядной демонстрации работы алгоритмов
* Возможность экспорта графов в другие форматы

**Используемые источники**

* Python 3 – https://docs.python.org/3/
* PyQt6 – https://www.riverbankcomputing.com/static/Docs/PyQt6/
* JSON – Официальная документация.

**Приложения**

(Фрагменты кода, примеры работы программы, скриншоты интерфейса.)