**Документация к проекту 'Навигатор' по PyQt5.**

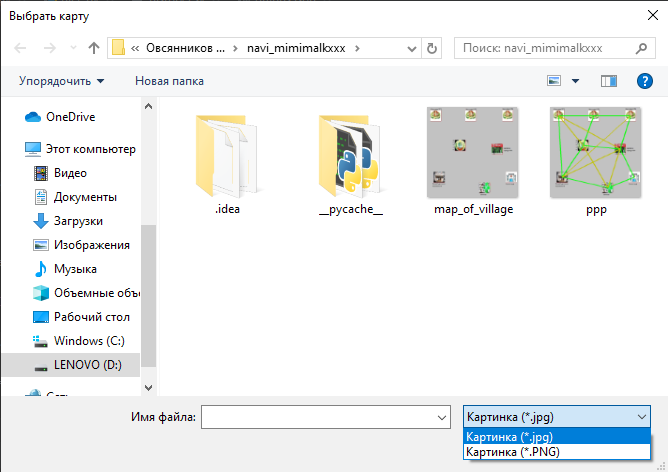
1. [*вступление*](#_Вступление.)
2. [*входные данные*](#_Входные_данные.)
3. [*описание работы приложения*](#_Описание_работы_приложения.)
4. [*используемые библиотеки и функции*](#_Используемые_библиотеки_и)

# **Вступление.**

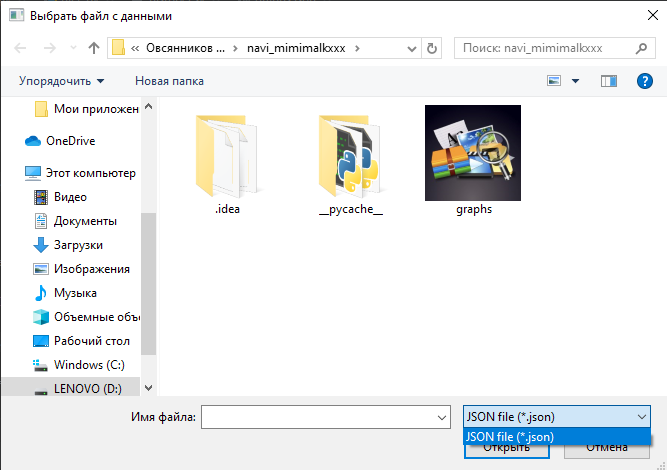
Продукт предназначен для удобства навигации в различных местах, где навигаторы рода Яндекс или Гугл карт не применимы. Речь идет, например, о больших складах, или торговых центрах, где очень часто сложно сразу найти правильный и оптимальный маршрут до места назначения. В торговых компаниях и не только часто важна каждая минута, ведь иначе может сильно упасть производительность и соответственно прибыль компании. Но как часто рабочие, особенно новые, теряются в больших складах при этом не доставляя что-либо по адресу в нужное время? Я сам как-то раз случайно попал на склад в Leomax, и ориентироваться там без каких-либо инструментов было очень сложно, я потратил достаточно много времени, пока нашёл выход. А ведь как я уже сказал, такое опоздание от рабочего может быть эквивалентно достаточно большим суммам денег. Задумавшись над этой проблемой, я пришел к выводу, что очень было бы удобно иметь при себе навигатор, показывающий оптимальный путь до пункта назначения. Именно такой навигатор я и разработал. Базовый функционал приложения позволяет строить маршруты используя различные способы передвижения и рассчитывать оптимальные пути, исходя из заданных пользователем параметров.

# **Входные данные**.

При запуске программы вам выпадет на экран окно выбора файла с картинкой, представляющей карту:



Здесь вы должны выбрать файл с картой, на которой будут обозначены и подписаны различные объекты, карт может быть несколько.



После этого вам предложат выбрать json-файл с внутренними данными:

В этом файле должен в формате json храниться список со следующими параметрами программы:

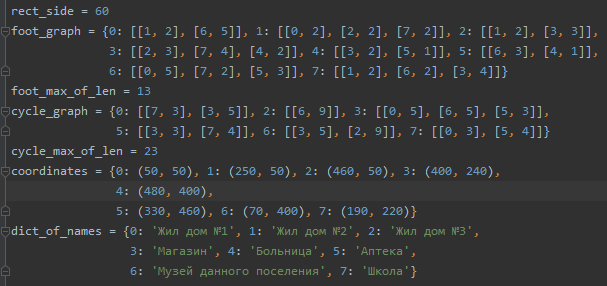
1. сторона квадрата;
2. граф транспорта №1(в примере это будут пешеходные дорожки);
3. максимальный путь между любыми 2мя точками в первом графе;
4. граф транспорта №2(в примере это личный транспорт: велосипеды и тд)
5. максимальный путь между любыми 2мя точками во втором графе;

* хочу заметить, что ключами в графах являются некоторые id, которые определяются из последнего в списке словаря

1. словарь координат (ключ - id, значение - кортеж координат в пикселях на карте введенной ранее);
2. словарь дешифратор (id: значение).

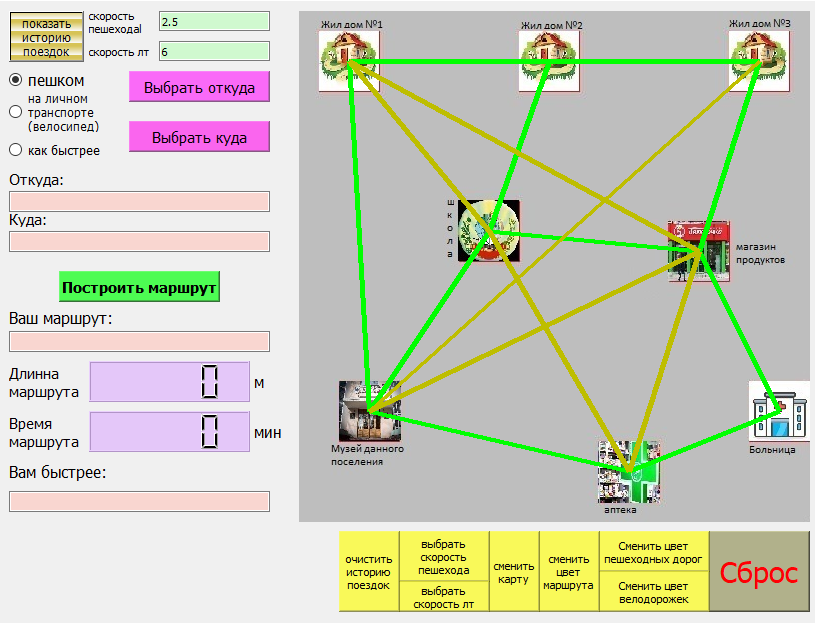
Для правильной записи в json-файл в комплекте идет специальная программа.

Больше на вход программе ничего подавать не надо. Для удобства приведу пример правильной подачи данных:



# **Описание работы приложения.**

Внешний вид при старте (зависит от поданных данных):



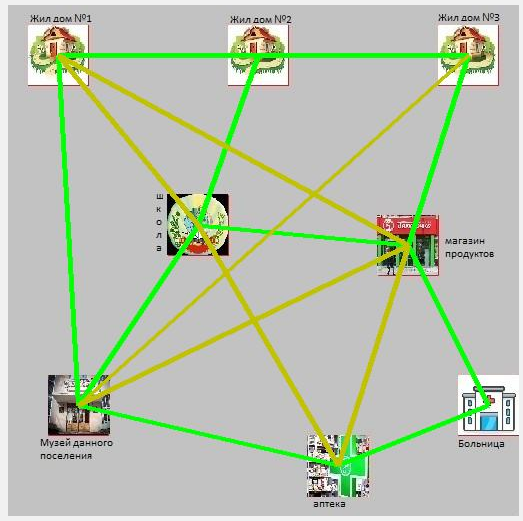
Все белые поля предназначены для вывода информации, соответствующей подписям к этим полям. Стоит заметить, что скорости приведены в системе СИ.



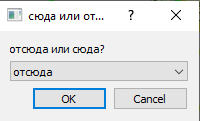
В левом углу сверху можно увидеть такую группу кнопок:

здесь можно выбрать как вы хотите добираться до пункта назначения или же вовсе определить, как добраться быстрее.

Справа вы видите большую карту:

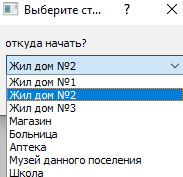


на ней прорисованы все пути, обозначенные в графах, для каждого графа свой собственный цвет дорог (по началу он задается автоматически). При двойном нажатии на любой объект на карте появится диалог:

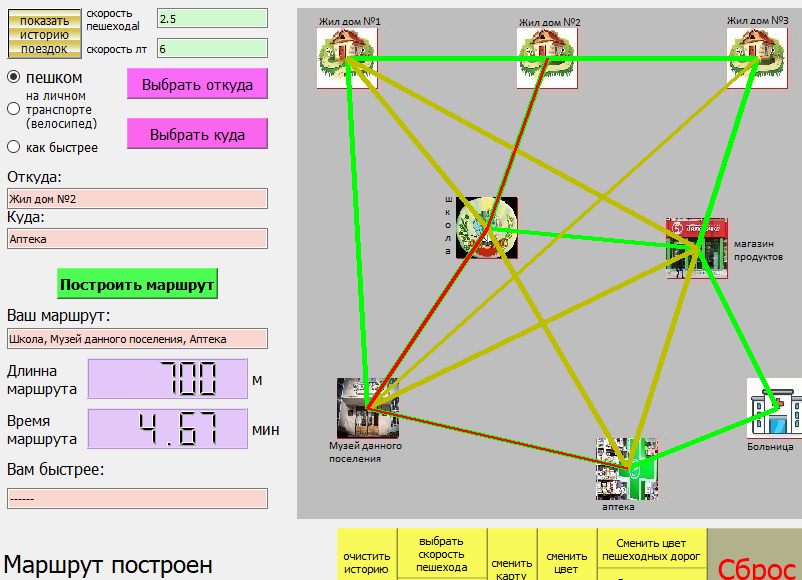


При выборе соответствующего варианта в соответствующие поля попадает этот объект, например, если дважды кликнуть на «Жил дом №1» и в диалоге ответить «отсюда», то в поле «Откуда» попадет «Жил дом №1».

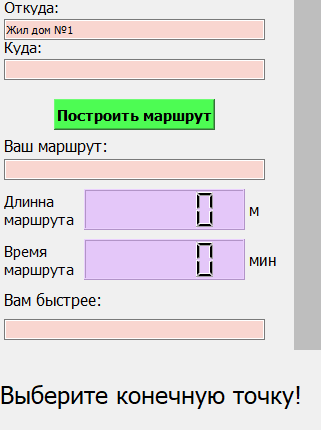
Так же выбрать откуда можно комбинацией клавиш «cotrol + f» или кнопкой «Выбрать откуда», при активации этих вещей выпадет диалог:

здесь надо выбрать, откуда вы хотите начать. Аналогично выбрать куда вы хотите попасть можно комбинацией «control + t» или кнопкой «Выбрать куда».

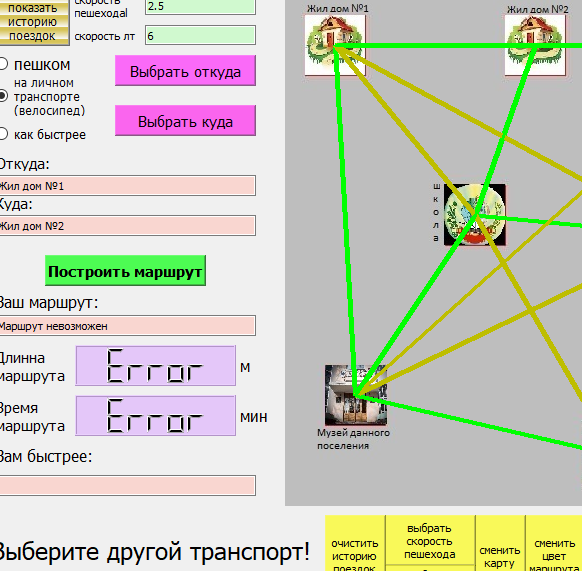
Теперь для примера выберем, что мы хотим попасть в «Музей данного поселения» и нажмем «Построить маршрут». Тогда на карте маршрут выделится соответствующим цветом (как его менять будет сказано позже), а в поля ниже попадет информация о маршруте:



Для случая, когда пользователь не точки маршрута или одну из точек, программа выдаст ошибку следующего вида:



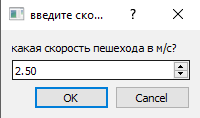
При отсутствии дороги из одного пункта в другой на выбранном транспорте или любых других пользовательских ошибках внизу появляется сообщение об этой ошибке, для примера укажем маршрут на велосипеде между пунктами «Жил дом №1» и «Жил дом №2». К пункту назначения не идут велодорожки, поэтому программа сообщит и об этом:



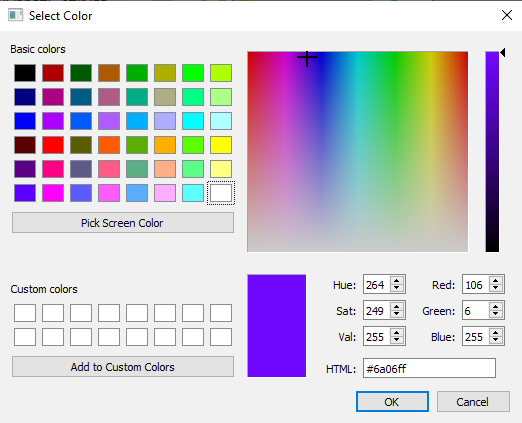
Теперь рассмотрим пользовательскую настройку интерфейса приложения.

В правом нижнем углу находятся целый ряд кнопок за это отвечающих:

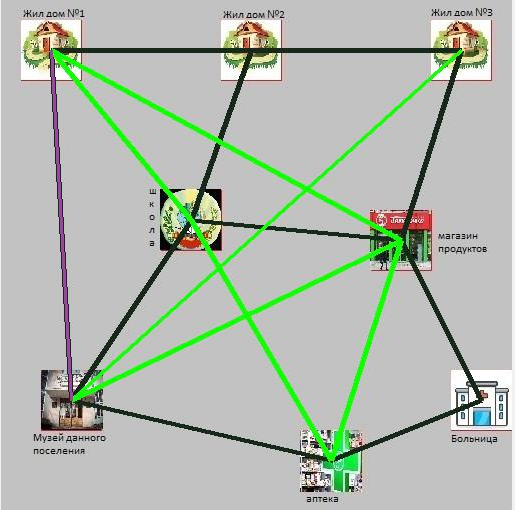


При нажатии на «выбрать скорость пешехода» или «выбрать скорость лт» выпадет диалог взятия числа с плавающей точкой:

При нажатии любой из кнопок, начинающихся на «Сменить цвет» выпадет диалог выбора цвета:



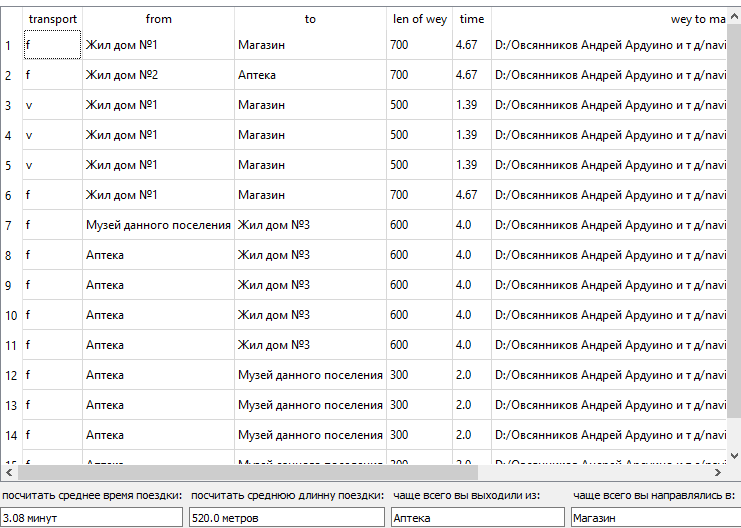
Так можно перерисовать карту под удобные цвета:



Нажав кнопку «Сброс», вы просто отчистите приложение от текущего маршрута и всей выведенной информации о нём. Настройки, заданные ранее сохранятся.

Кнопка «сменить карту» заново запросит у вас информацию из [пункта 2 оглавления](#_Входные_данные.).

Так же после каждого нового построенного маршрута он будет записываться в историю поездок ('history.csv'), нажав на кнопку «показать историю поездок» вам выпадет окно с таблицей истории. В низу этого окна показаны среднее время, длина поездки, а также наиболее часто встречаемые в истории начальная и конечная точки маршрута. При желании историю поездок можно очистить.



# **Используемые библиотеки и функции.**

****

