

Документация к проекту 'Навигатор' по PyQt5.

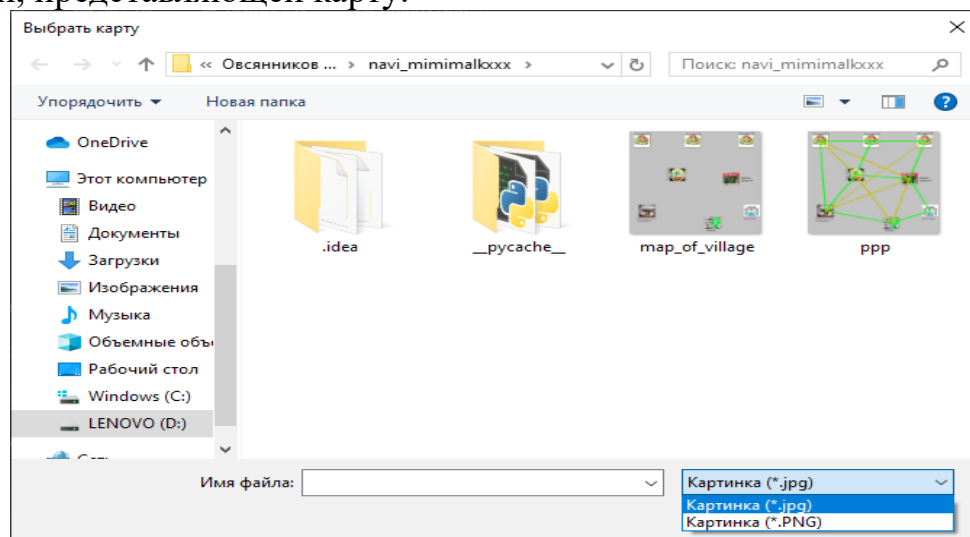
- 1) [вступление](#)
- 2) [входные данные](#)
- 3) [описание работы приложения](#)
- 4) [используемые библиотеки и функции](#)

Вступление.

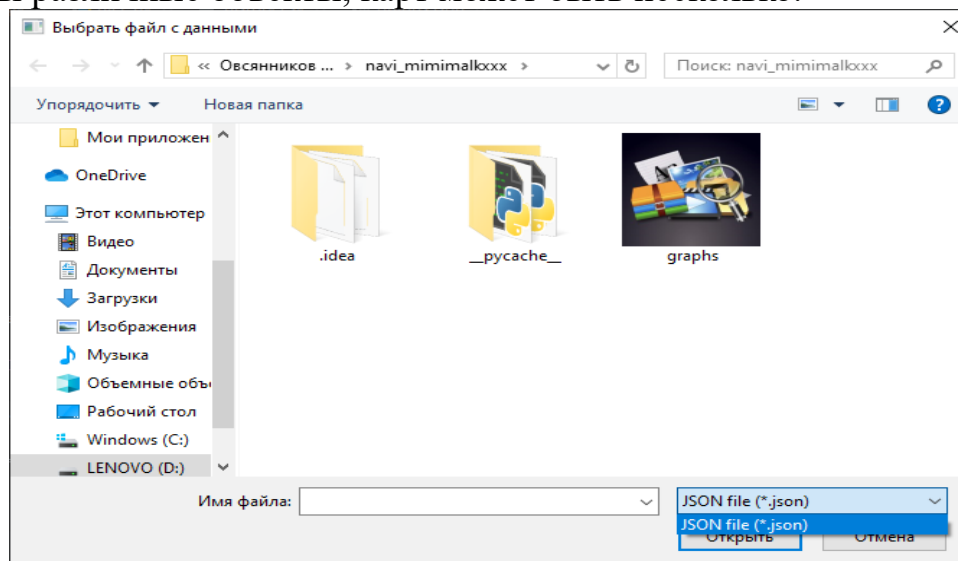
Продукт предназначен для удобства навигации в различных местах, где навигаторы рода Яндекс или Гугл карт не применимы. Речь идет, например, о больших складах, или торговых центрах, где очень часто сложно сразу найти правильный и оптимальный маршрут до места назначения. В торговых компаниях и не только 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 Леомах, 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
- 6) словарь координат (ключ - id, значение - кортеж координат в пикселях на карте введенной ранее);
- 7) 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:

```
rect_side = 60
foot_graph = {0: [[1, 2], [6, 5]], 1: [[0, 2], [2, 2], [7, 2]], 2: [[1, 2], [3, 3]],
              3: [[2, 3], [7, 4], [4, 2]], 4: [[3, 2], [5, 1]], 5: [[6, 3], [4, 1]],
              6: [[0, 5], [7, 2], [5, 3]], 7: [[1, 2], [6, 2], [3, 4]]}
foot_max_of_len = 13
cycle_graph = {0: [[7, 3], [3, 5]], 2: [[6, 9]], 3: [[0, 5], [6, 5], [5, 3]],
               5: [[3, 3], [7, 4]], 6: [[3, 5], [2, 9]], 7: [[0, 3], [5, 4]]}
cycle_max_of_len = 23
coordinates = {0: (50, 50), 1: (250, 50), 2: (460, 50), 3: (400, 240),
              4: (480, 400),
              5: (330, 460), 6: (70, 400), 7: (190, 220)}
dict_of_names = {0: 'Жил дом №1', 1: 'Жил дом №2', 2: 'Жил дом №3',
                 3: 'Магазин', 4: 'Больница', 5: 'Аптека',
                 6: 'Музей данного поселения', 7: 'Школа'}
```

Описание работы приложения.

Внешний вид при старте (зависит от поданных данных):

показать историю поездок

скорость пешехода: 2.5

скорость лт: 6

☒ пешком
☐ на личном транспорте (велосипед)
☐ как быстрее

Выбрать откуда

Выбрать куда

Откуда:

Куда:

Построить маршрут

Ваш маршрут:

Длина маршрута: 0 м

Время маршрута: 0 мин

Вам быстрее:

очистить историю поездок

выбрать скорость пешехода

выбрать скорость лт

сменить карту

сменить цвет маршрута

сменить цвет пешеходных дорог

сменить цвет велодорожек

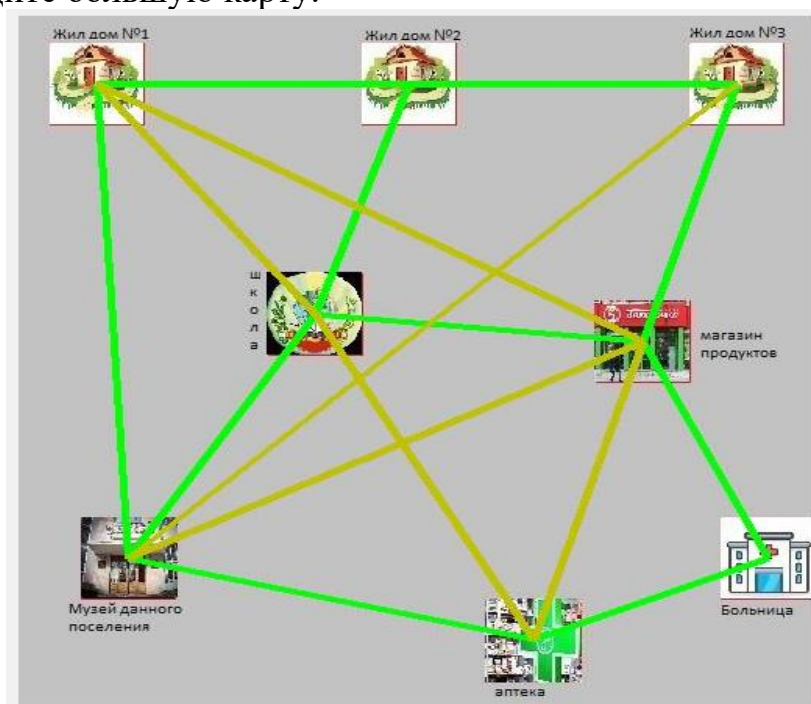
Сброс

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

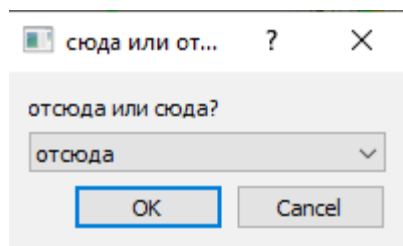
- ☒ пешком
☐ на личном транспорте (велосипед)
☐ как быстрее

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.

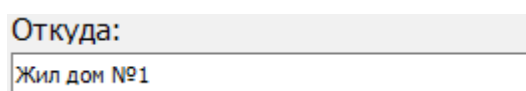
Справа вы видите большую карту:



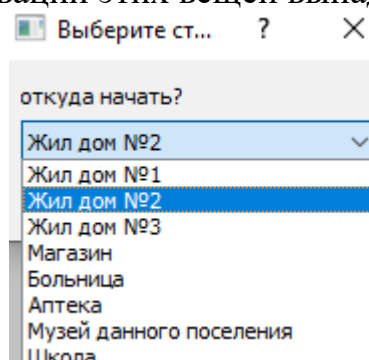
на ней прорисованы все пути, обозначенные в графах, для каждого графа свой собственный цвет дорог (по началу он задается автоматически). При двойном нажатии на любой объект на карте появится диалог:



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 «control + 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:

показать историю поездок

пешком

на личном транспорте (велосипед)

как быстрее

Выбрать откуда

Выбрать куда

Откуда:

Жил дом №2

Куда:

Аптека

Построить маршрут

Ваш маршрут:

Школа, Музей данного поселения, Аптека

Длина маршрута

700

М

Время маршрута

4.67

МИН

Вам быстрее:

Жил дом №1

Жил дом №2

Жил дом №3

Школа

Музей данного поселения

Аптека

Магазин продуктов

Больница

очистить историю

выбрать скорость пешехода

сменить карту

сменить цвет

Сменить цвет пешеходных дорог

Сброс

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

Откуда:

Жил дом №1

Куда:

Построить маршрут

Ваш маршрут:

Длина маршрута

0

М

Время маршрута

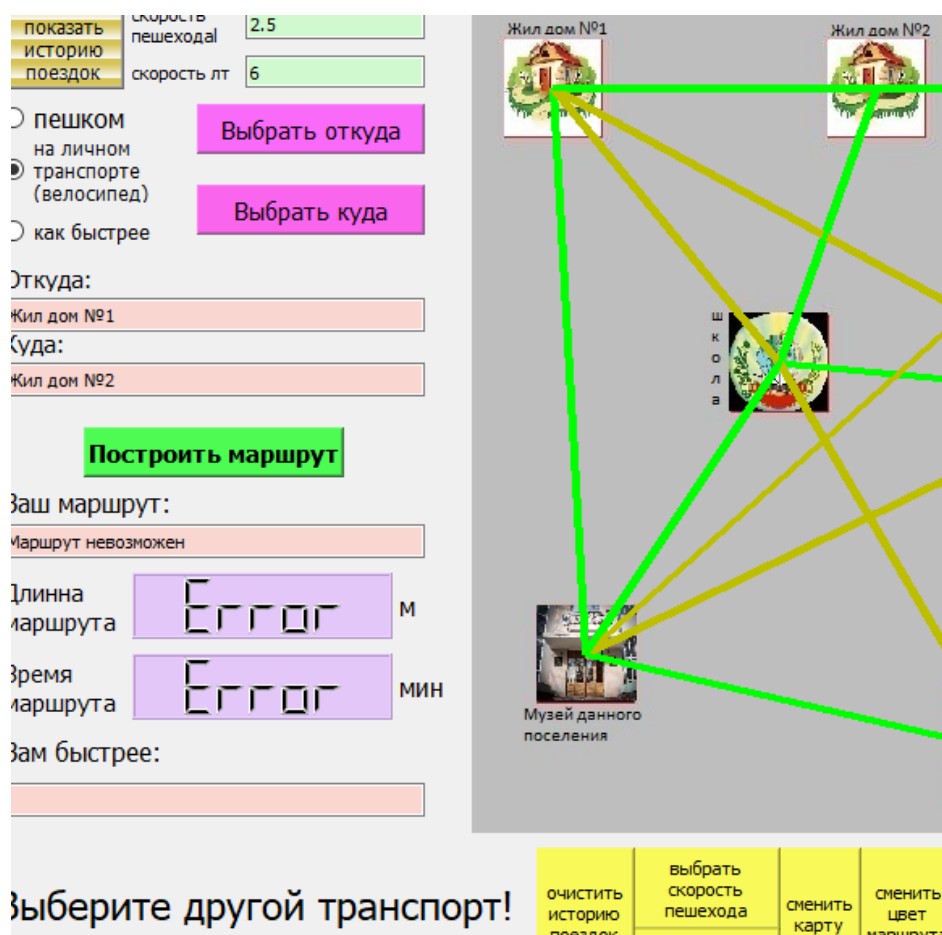
0

МИН

Вам быстрее:

Выберите конечную точку!

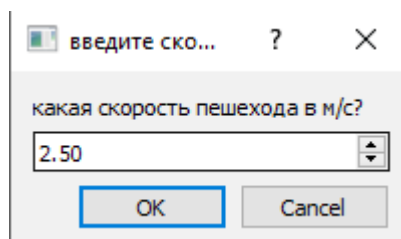
При отсутствии дороги из одного пункта в другой на выбранном транспорте или любых других пользовательских ошибках внизу появляется сообщение об этой ошибке, для примера укажем маршрут на велосипеде между пунктами «Жил дом №1» и «Жил дом №2». К пункту назначения не идут велодорожки, поэтому программа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



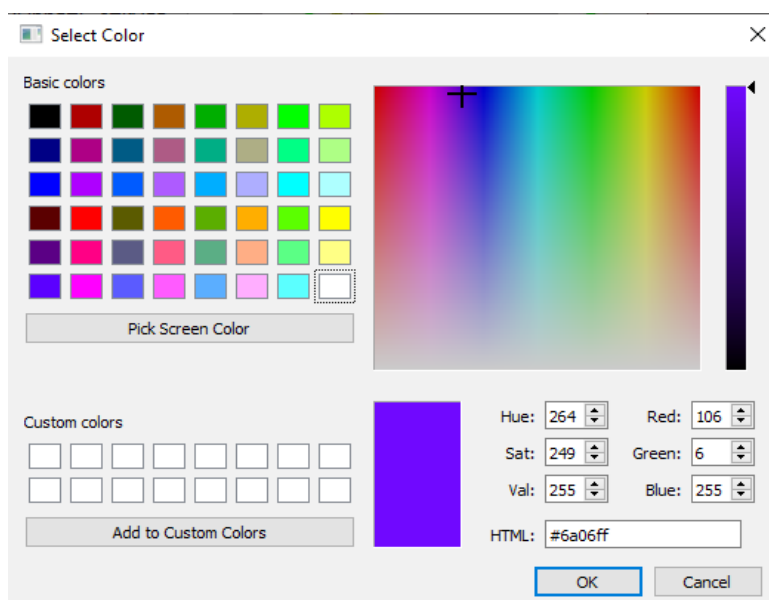
Теперь рассмотрим пользовательскую настройку интерфейса приложения. В правом нижнем углу находятся целый ряд кнопок за это отвечающих:



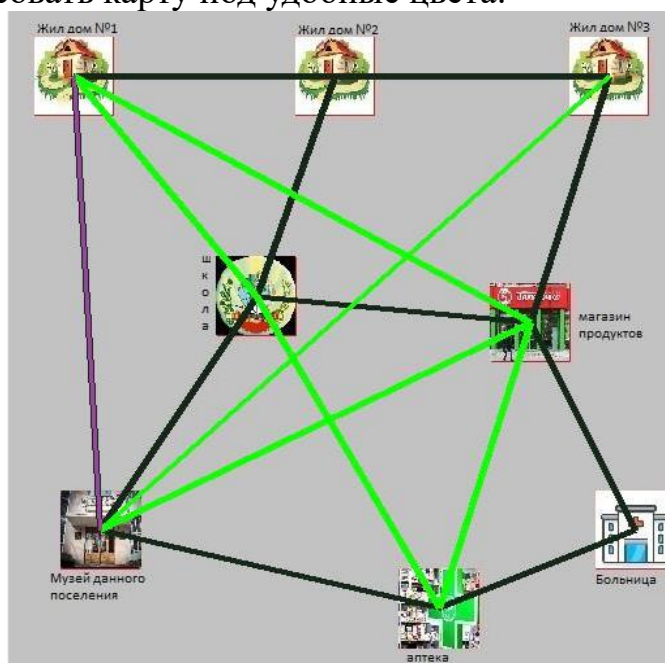
При нажатии на «выбрать скорость пешехода» или «выбрать скорость лт» выпадет диалог взятия числа с плавающей точкой:



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 сохраняются.

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

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.

	transport	from	to	len of wey	time	wey to ma
1	f	Жил дом №1	Магазин	700	4.67	D:/Овсянников Андрей Ардуино и т д/navi
2	f	Жил дом №2	Аптека	700	4.67	D:/Овсянников Андрей Ардуино и т д/navi
3	v	Жил дом №1	Магазин	500	1.39	D:/Овсянников Андрей Ардуино и т д/navi
4	v	Жил дом №1	Магазин	500	1.39	D:/Овсянников Андрей Ардуино и т д/navi
5	v	Жил дом №1	Магазин	500	1.39	D:/Овсянников Андрей Ардуино и т д/navi
6	f	Жил дом №1	Магазин	700	4.67	D:/Овсянников Андрей Ардуино и т д/navi
7	f	Музей данного поселения	Жил дом №3	600	4.0	D:/Овсянников Андрей Ардуино и т д/navi
8	f	Аптека	Жил дом №3	600	4.0	D:/Овсянников Андрей Ардуино и т д/navi
9	f	Аптека	Жил дом №3	600	4.0	D:/Овсянников Андрей Ардуино и т д/navi
10	f	Аптека	Жил дом №3	600	4.0	D:/Овсянников Андрей Ардуино и т д/navi
11	f	Аптека	Жил дом №3	600	4.0	D:/Овсянников Андрей Ардуино и т д/navi
12	f	Аптека	Музей данного поселения	300	2.0	D:/Овсянников Андрей Ардуино и т д/navi
13	f	Аптека	Музей данного поселения	300	2.0	D:/Овсянников Андрей Ардуино и т д/navi
14	f	Аптека	Музей данного поселения	300	2.0	D:/Овсянников Андрей Ардуино и т д/navi
15	f	Аптека	Музей данного поселения	300	2.0	D:/Овсянников Андрей Ардуино и т д/navi
<						>

посчитать среднее время поездки: 3.08 минут
 посчитать среднюю длину поездки: 520.0 метров
 чаще всего вы выходили из: Аптека
 чаще всего вы направлялись в: Магазин

Используемые библиотеки и функции.

Pillow	6.0.0	▲ 8.0.1
PyQt5	5.15.1	5.15.1
PyQt5-sip	12.8.1	12.8.1

```

from functions_for_navigator import min_way_func, StartPointError, EndPointError, WeyError
from PyQt5.QtGui import QPixmap
from interfeis import Ui_MainWindow
from PyQt5.QtGui import QPainter, QColor
from PyQt5.QtWidgets import QApplication, QMainWindow, QFileDialog, QColorDialog, QInputDialog, \
    QTableWidgetItem
import sys
import csv
from random import randint
from PyQt5.QtCore import Qt
from PIL import ImageQt
import json
from PIL import Image, ImageDraw
from PyQt5 import uic

```