Qt+OSM地图软件开发

软件功能

1. 支持对地图进行刷新
2. 支持通过菜单导入导出OSM地图
3. 支持通过菜单查看软件的帮助界面
4. 支持使用鼠标拖动进行地图的平移
5. 支持使用鼠标滚轮或侧边滑动条进行地图缩放
6. 支持显示当前地图区域的经纬度
7. 支持切换不同的最短路径算法
8. 支持使用鼠标来选中道路起点与终点
9. 使用队列来管理选中的起点与终点
10. 支持求解起点与终点间的最短路径
11. 支持对地图中的最短路径进行红色高亮

技术亮点

1. 通过巧妙的窗口坐标变换实现地图的动态平移与缩放
2. 通过 `Radio Button` 作为顶点，使其可以被选中
3. 通过 `QButtonGroup` 来管理顶点最多只能选中两个
4. 使用 `Qt Designer` 来设计软件的界面
5. 使用 `Haversine` 公式来计算两点间的距离
6. 尝试使用SFINAE来支持多种最短路径算法
7. 使用队列来刷新选中的顶点
8. 使用CSS来美化Qt的组件

代码整体架构

```
class MainWindow : public QMainWindow {
    Q_OBJECT

    struct Node {
        ll id;
        double lon, lat;
    };
    struct Way {
        ll id;
        vector<Node> nodes;
        map<string, string> tags;
    };

   public:
    MainWindow(QWidget *parent = nullptr);
    ~MainWindow();

   protected:
```

```
    // mapPaint.cpp
    void paintEvent(QPaintEvent *event) override;
    // mapEvent.cpp
    void updateVertexPos();
    void updateLocationLabel();
    void mousePressEvent(QMouseEvent *event) override;
    void mouseMoveEvent(QMouseEvent *event) override;
    void wheelEvent(QWheelEvent *event) override;
    void resizeEvent(QResizeEvent *event) override;

private slots:
    // mapSlot.cpp
    void updateDistLabel(double minDist);
    void onVertexToggled(bool checked, ll id);
    void on_scaleSlider_valueChanged(int value);
    void on_algoButton_released();
    void on_flushButton_released();

public:
    double minLon, maxLon;      // 经度范围
    double minLat, maxLat;      // 纬度范围
    double scaleFactor = 1.0;   // 缩放因子

private:
    Ui::MainWindow *ui;         // ui界面
    QButtonGroup vertexGroup;   // 顶点组
    QSlider *scaleSlider;       // 缩放滚动条
    QLabel *distanceLabel;      // 文本显示器
    QLabel *locationLabel;      // 地图区域坐标

    map<ll, list<ll>> adj;              // 顶点邻接表
    list<ll> selectedVertex;            // 起点终点队列
    map<ll, QRadioButton *> vertex;     // 顶点按钮组件
    unordered_set<ll> visitedVertex;    // 最短路径途经的顶点

    QPoint lastMousePos;    // 鼠标位置

    map<ll, Node> nodes;    // 顶点
    map<ll, Way> ways;      // 道路

    // mapLoad.cpp
    void loadOSM();     // 加载OSM文件
    void importFile();  // 导入文件
    void exportFile();  // 导出文件
    void helpShow();    // 显示帮助窗口

    // mapSPath.cpp
    struct SPath_tag {};
    struct dijkstra_tag : public SPath_tag {};
    struct AStar_tag : public SPath_tag {};
    enum { Dijkstra, AStar } algo_tag = Dijkstra;

    double toRadians(double degrees);
    double calcuDist(double lat1, double lon1, double lat2, double lon2);
    double updateShortestPath(dijkstra_tag);
    double updateShortestPath(AStar_tag);
};
```