

《创新训练课B》结题报告

学院： **机电工程与自动化学院**

题目： **模拟深圳地铁自动售票系统**

班级： **自动化2班**

姓名： **常伯符**

学号： **220320221**

教师： **吴晓军**

上交日期： **2023.12.28**

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| 2023.8.1 | V0.1 | 1. 基于C++初步实现了换乘站查询、换乘方案打印等一系列控制台功能 | 常伯符 |
| 2023.12.24 | V1.0 | 1. 基于Qt6.2搭建程序前端框架 2. 修复了控制台程序的bug | 常伯符 |
| 2023.12.25 | V1.1 | 1、实现地图缩放、点击选站、输入选站等功能 | 常伯符 |
| 2023.12.26 | V1.2 | 1、实现换乘方案图形化呈现 | 常伯符 |
| 2023.12.27 | V1.3 | 1. 实现售票，程序目标基本完成并打包 | 常伯符 |

**目 录**

[1 引言 1](#_Toc519846913)

[1.1编写目的 1](#_Toc519846914)

[1.2背景 1](#_Toc519846915)

[1.3定义 1](#_Toc519846916)

[1.4参考资料 1](#_Toc519846917)

[2 任务概述 1](#_Toc519846918)

[3 需求分析 1](#_Toc519846919)

[3.1 用户需求分析 1](#_Toc519846920)

[3.2 运行环境 2](#_Toc519846921)

[4 功能及操作介绍 2](#_Toc519846922)

[4.1 操作 2](#_Toc519846923)

[4.2 功能 2](#_Toc519846924)

[5 系统设计 3](#_Toc519846925)

[5.1 总体架构设计 3](#_Toc519846926)

[5.2 模块分析与设计 3](#_Toc519846927)

[5.3软件结构（流程图） 6](#_Toc519846928)

[6 调试与测试 8](#_Toc519846929)

[6.1调试过程 8](#_Toc519846930)

[6.2测试结果 8](#_Toc519846931)

[7 编程中遇到的问题 13](#_Toc519846932)

[7.1 问题1 13](#_Toc519846933)

[7.2 问题2 13](#_Toc519846934)

[7.3 问题3 13](#_Toc519846935)

[8分析总结与心得体会 13](#_Toc519846936)

# 1 引言

## 1.1编写目的

本文详细介绍了售票系统的各个功能的实现及其代码逻辑。

## 1.2背景

该项目为哈尔滨工业大学(深圳)自动化专业 2023 年暑季小学期创新训练课 B 的实践作业，旨在通过编写一个图形化应用程序掌握对C++及Qt的理解及运用。

## 1.3定义

“换乘方案”或“方案”除非特殊说明，一般指用户在始发站和终点站不变的情况下，可以选择的乘坐地铁线路以及中间的换乘站的组合。不同的方案往往有着不同的价格。例如，用户想要从西丽站出发去泰安站，可以选择5号线和7号线两种方案，但乘坐7号线比5号线贵3元。当然，本程序所使用的价格计算公式较为简化，与实际有所不符。

## 1.4参考资料

《C++ primer（第5版）》Stanley B. Lippman 等合著。

《Qt 6 C++开发指南》王维波等合著.

# 2 任务概述

本项目基于2024年深圳地铁线路规划图，立足用户实际情况分析，能够根据用户输入的起始站和终点站给出地铁乘坐及换乘方案（仅考虑换乘次数最少的方案），并实现售票功能。

具体目标有：

1. 显示基本的窗口、按钮等控件框架
2. 实现最基本的窗口转移操作
3. 显示地图，并支持缩放、平移等操作
4. 显示帮助文档，程序说明和版权声明
5. 在地图上的所有站点显示可供点击的按钮，实现点击选站操作
6. 实现输入站名选站的操作
7. 显示站点的信息
8. 自动计算出地铁乘坐及换乘方案，并利用qpaintEvent绘制图形化的方案显示
9. 自动显示方案的价格
10. 根据用户输入的人数和金额判断是否成功支付
11. 若成功则显示找零，否则显示缺少的金额

# 3 需求分析

## 3.1 用户需求分析

本程序旨在实现和改良真实的售票系统。用户在使用售票系统时，可能有查询站点、查看地图、选择站点、选择人数、进行支付等一系列需求。

本程序除了实现上述功能外，还着重强调了“换乘方案”的选择，正如在上文的定义中所述，不同的换乘方案往往有不同的价格、线路选择和经过站点。本程序并不仅仅计算价格最便宜的方案，而是在换乘次数最少的前提下，把所有可能的方案都计算出来，并交给用户选择。考虑到用户可能有独特的需求，这种想法是非常合理的。

除此之外，本程序还基于笔者的亲身体验做出一些改良。例如，传统的地铁系统并不能直观地展现用户应该选择地铁的哪个方向（例如，5号线有“往 赤湾 方向”和“往 黄贝岭 方向”两个方向），用户必须去找到地图并仔细确认才能选对方向。假如准备乘坐地铁时选错了方向，尤其是在一些有特殊设计或客流量较大的站点（深圳北站就是一个典型的例子），再走到另一个方向的等待处会非常费时，以至于错过地铁；若是上了地铁才发现选错方向，则更是费财费力，令人扼腕。本程序在展示方案时，会直接将地铁的方向标在起始站和每个换乘站的右上方，方便用户直观地选择方向。笔者认为，现代的地铁售票系统和支付小程序等都应该实现这个功能。

## 3.2 运行环境

操作系统：Windows 11

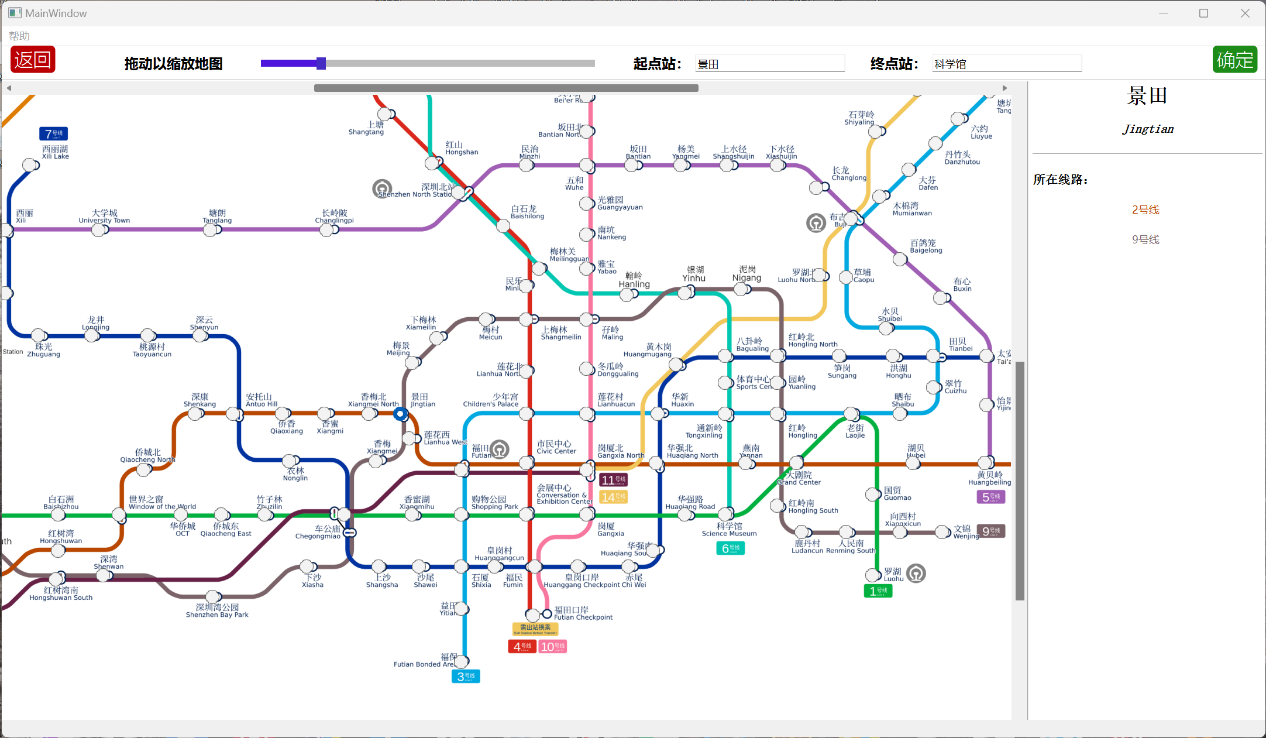
UI设计：C++ Qt 6.2

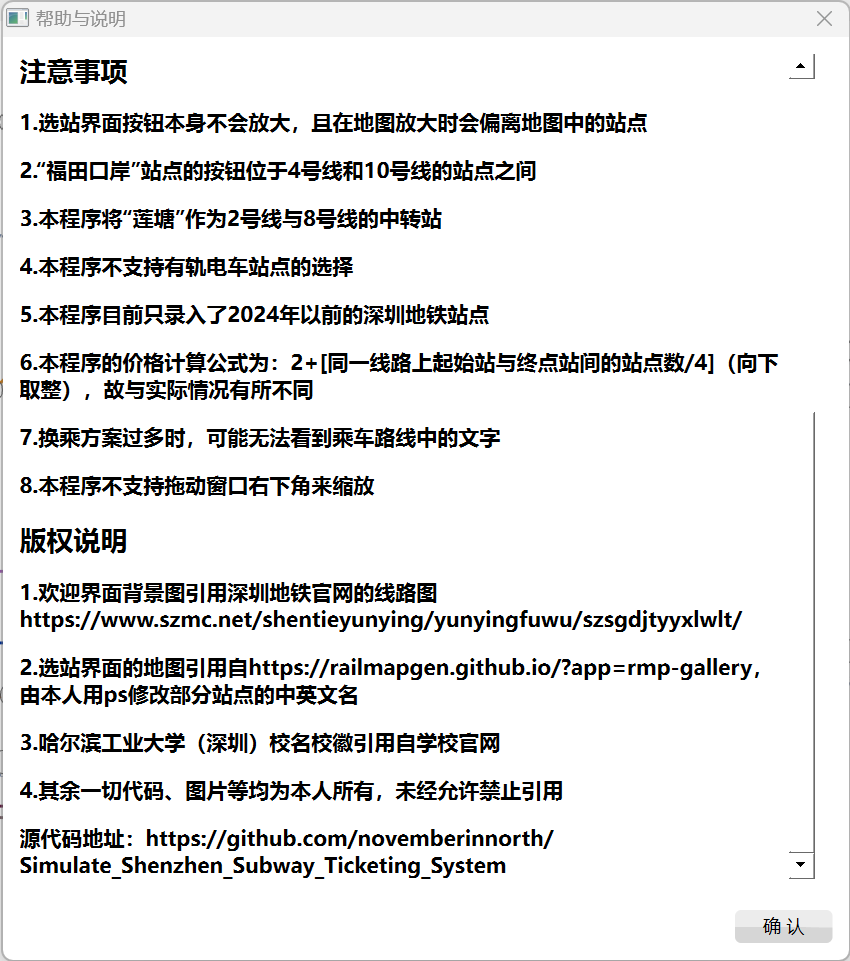
# 4 功能及操作介绍

## 4.1 操作



1.欢迎界面：点击开始购票进入选站界面，点击退出可以退出程序





2.选站界面： 点击返回可以返回到欢迎界面

拖动紫色的滑动条可以缩放地图

拖动地图上方和右方的滚动条可以移动地图

鼠标点击起点站和终点站右边的文本框可以设置输入为起点站或终点站

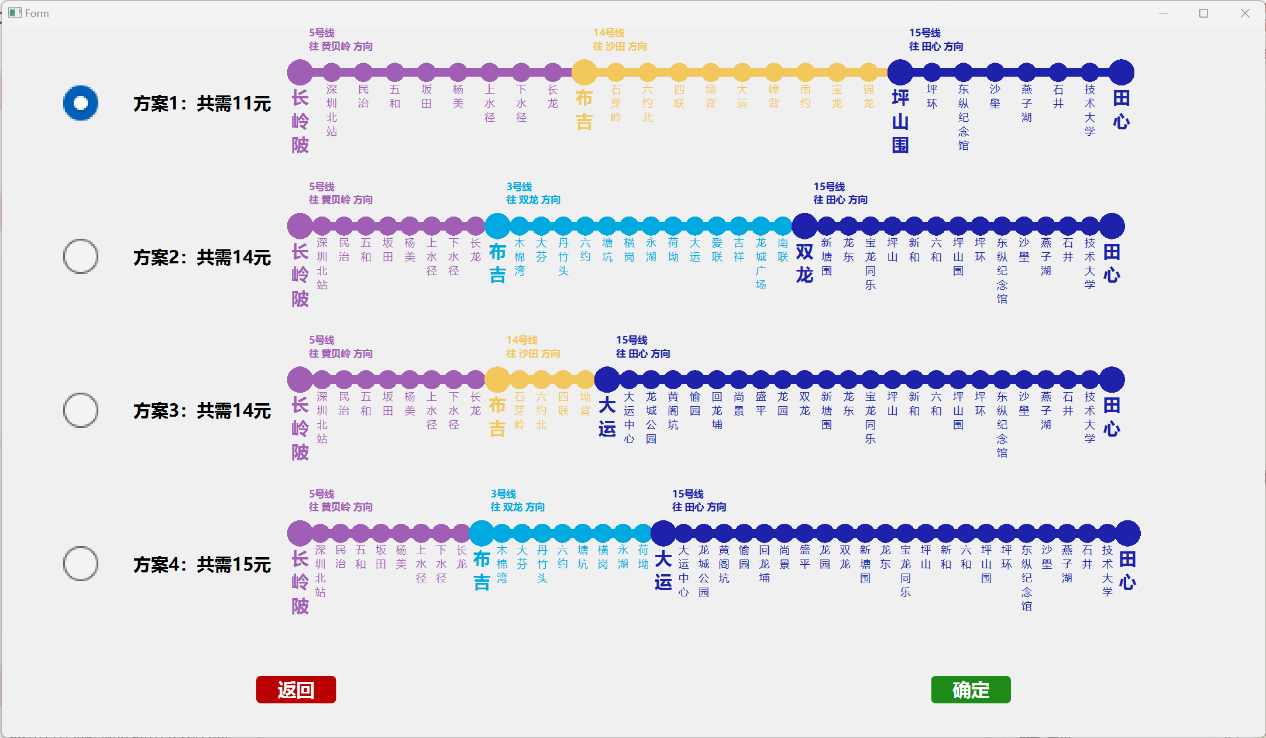
点击按钮可以选择站点

在文本框内输入站点名可以选择站点，并使对应按钮被选中

当选中站点时，右边会显示站点的中英文名和所在线路，线路用其对应的颜色显示

点击确定按钮进入选线界面（仅当起始站和终点站都被设置、站点名都正确且不为同一个站点时，确定键才会变绿并可以被点击）

点击帮助->说明文档可以打开帮助文档，点击下面的确认按钮可以关闭文档

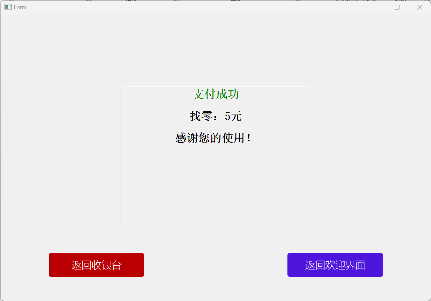


3.选线界面

点击返回可以返回到选站界面

点击方案左边的按钮可以选择该方案

点击确定进入支付界面



4.支付界面

在选择人数处输入人数并预览总价

在输入金额处输入金额

二者均支持直接键入、点箭头输入、鼠标滚轮输入，其中人数为整数，金额可以为小数

若支付金额大于等于总价，则显示支付成功并找零，可以返回收银台或返回欢迎界面

若支付进而小于总价，则显示支付失败和差价，只能返回收银台

## 4.2 功能

（1）支持地图缩放、平移、直接点击站点选站

（2）支持在输入框内选站

（3）支持图形化显示方案

（4）人数和金额支持多种输入方式

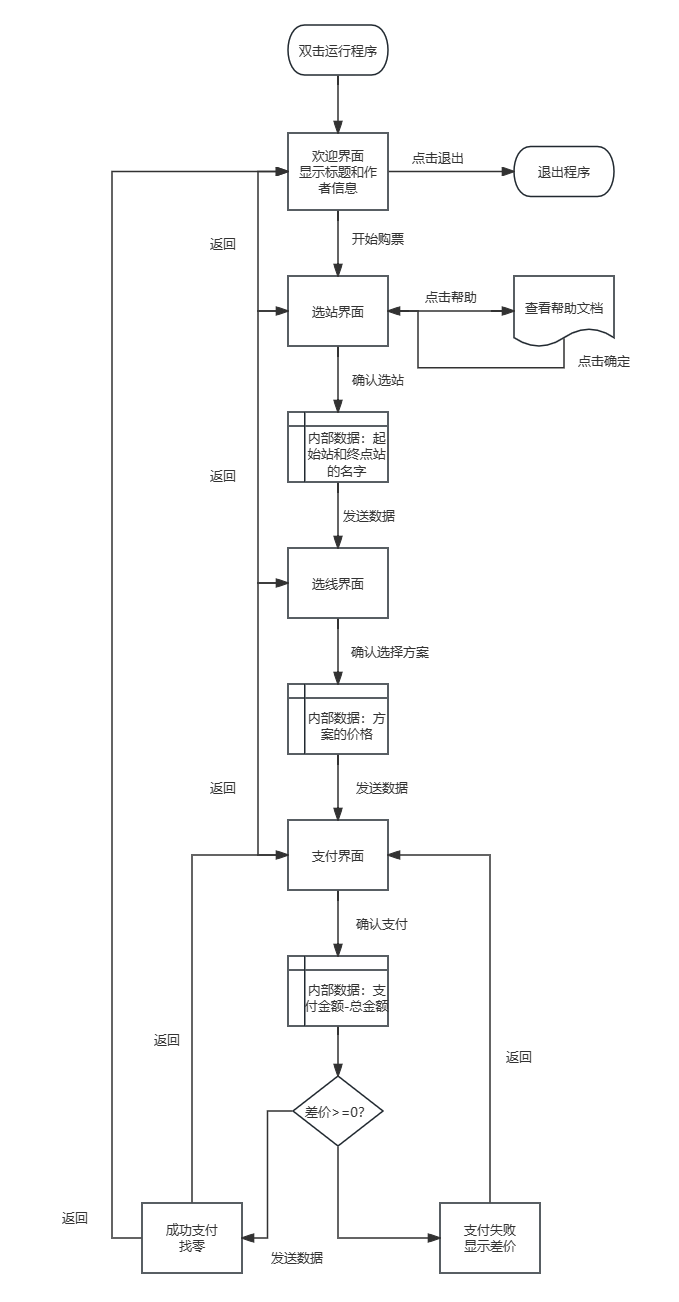
（5）支持差价自动显示与判别

（6）所有场景均设有返回和退出键，不会有死胡同

（7）内置帮助与说明文档

# 5 系统设计

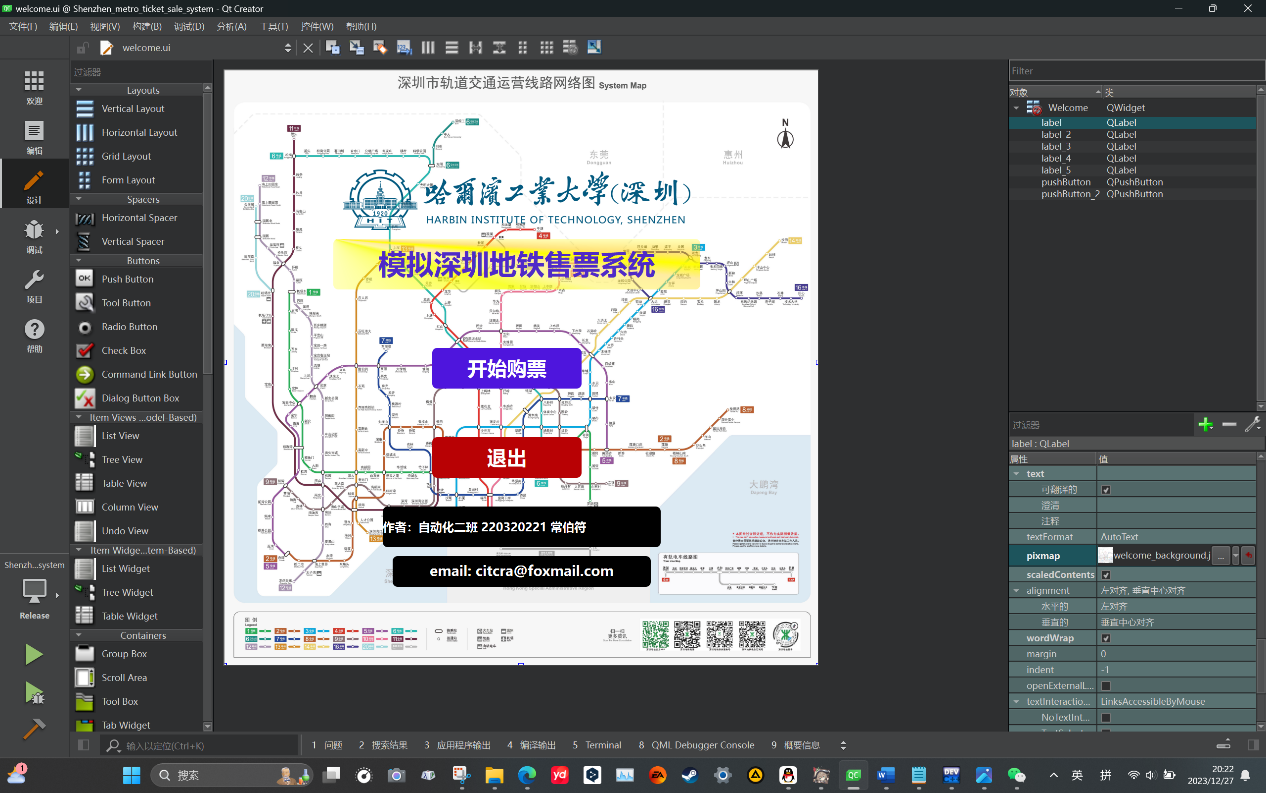
## 5.1 总体架构设计



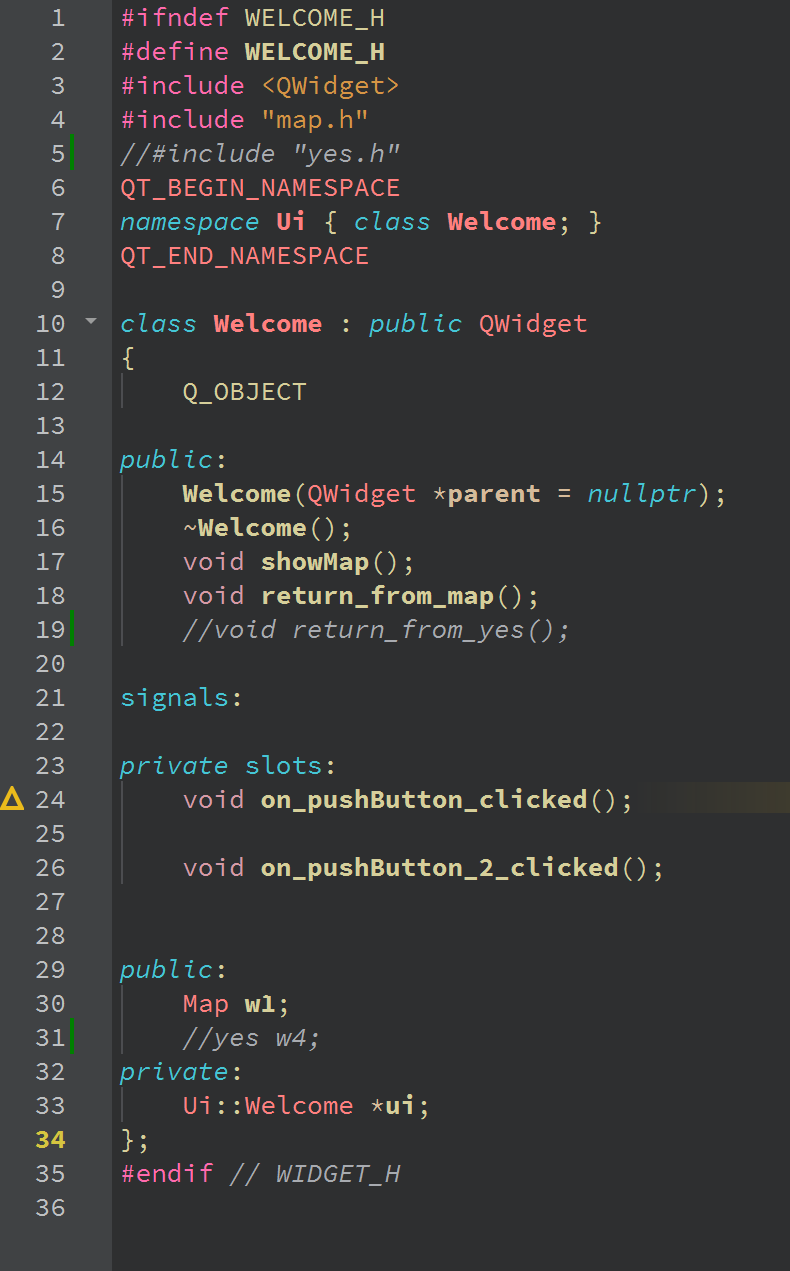
## 5.2 模块分析与设计

下面将分开详细介绍各个窗口的代码实现和算法逻辑。

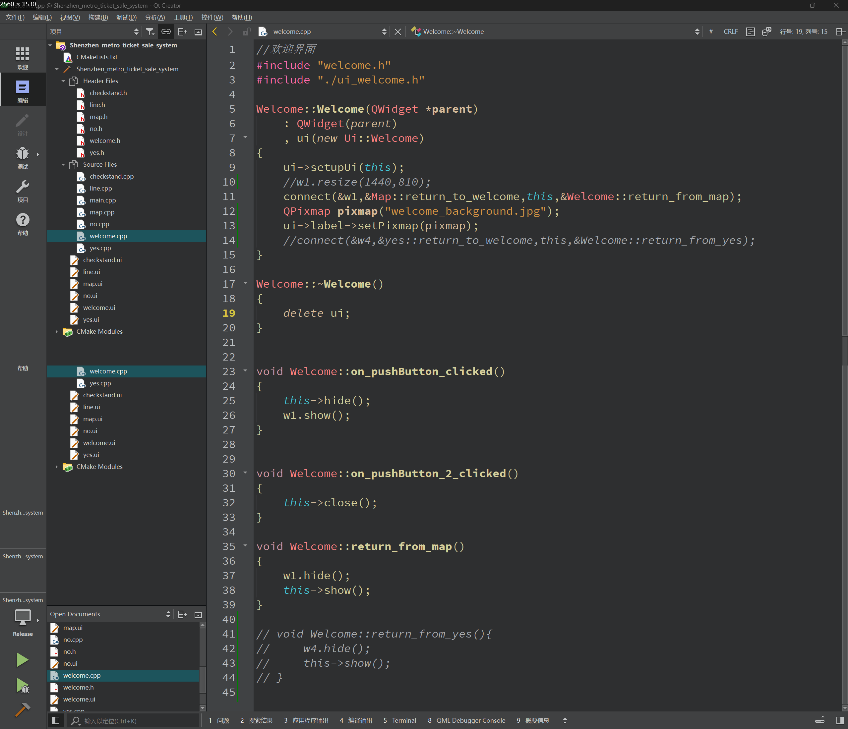
1. **欢迎界面 (welcome)**

欢迎界面是程序的主界面，主要用来显示程序标题、作者等信息。背景图采用QLabel的setPixmap放置图片，这是为了利用其相对路径，方便后期打包可以脱离资源文件。

头文件(welcome.h)代码：



源文件(welcome.cpp)代码：



主要实现过程：

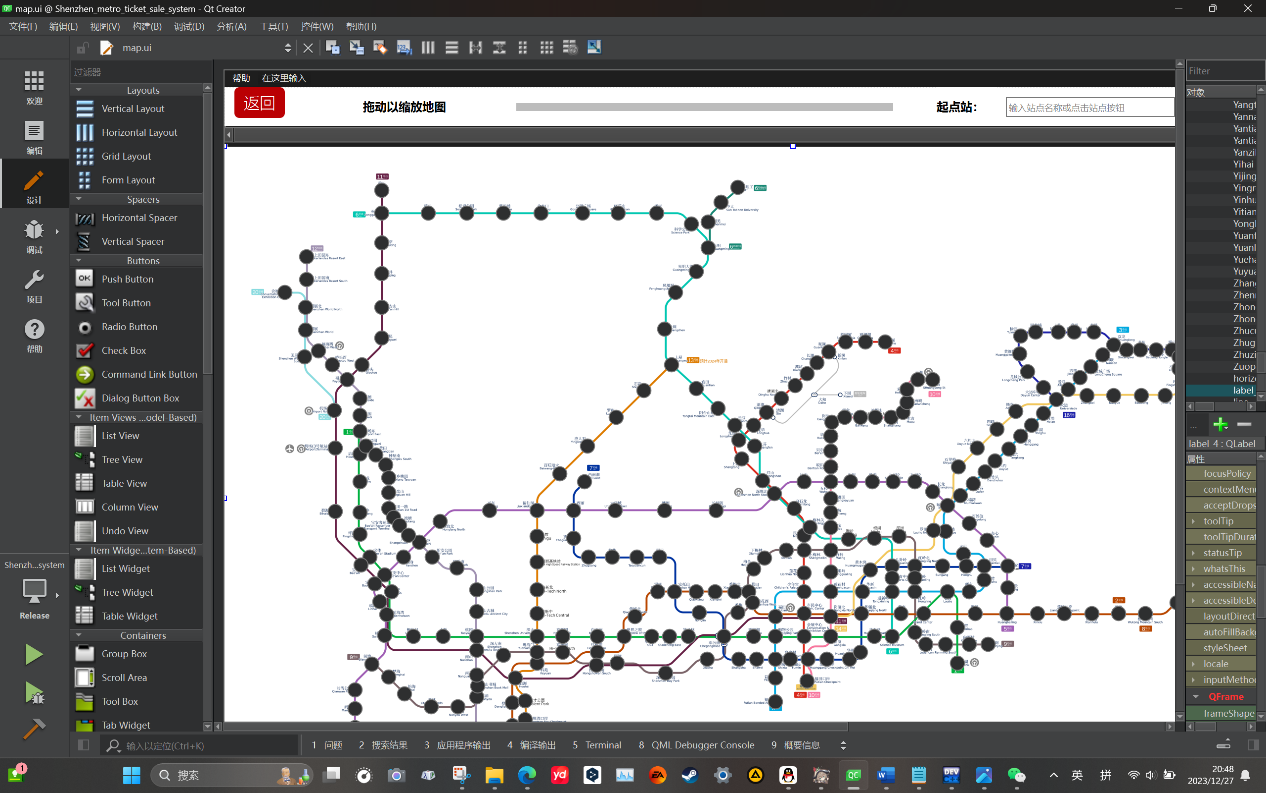
点击购票按钮，本窗口隐藏(this->hide())，选站界面出现(w2.show())；

点击退出按钮，本窗口隐藏，程序将自动退出；

使用connect接受来自选站界面的信号，触发槽函数，隐藏选站界面，显示本窗口，以实现返回的效果

后面的确定（打开下一个界面）与返回（返回上一个界面）的实现原理大多类似，不再赘述。

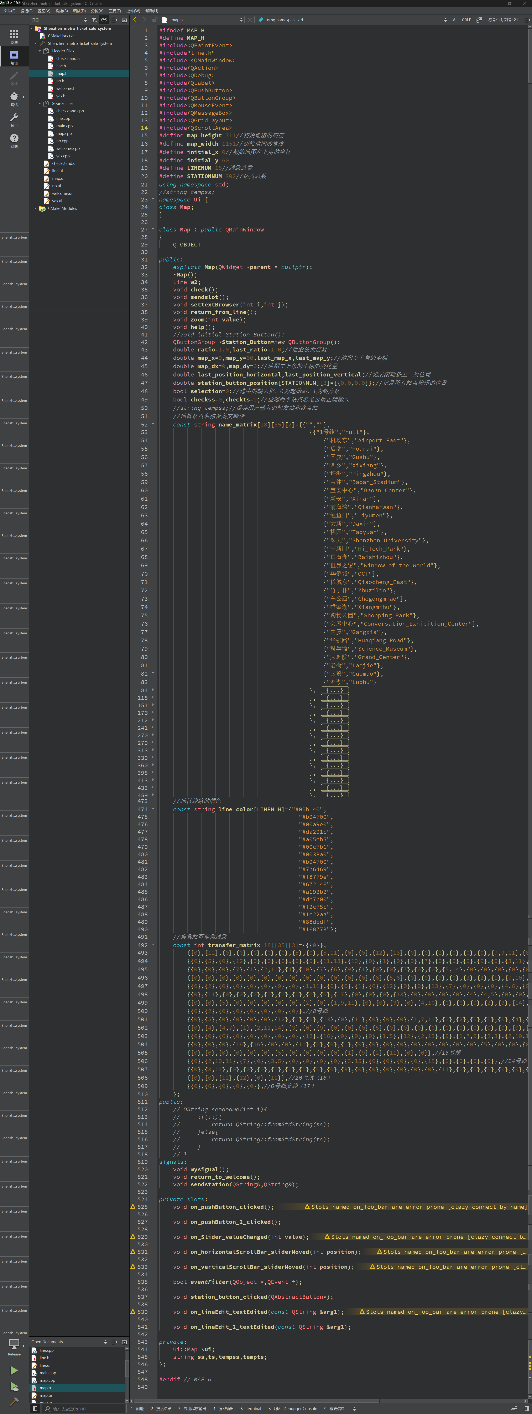
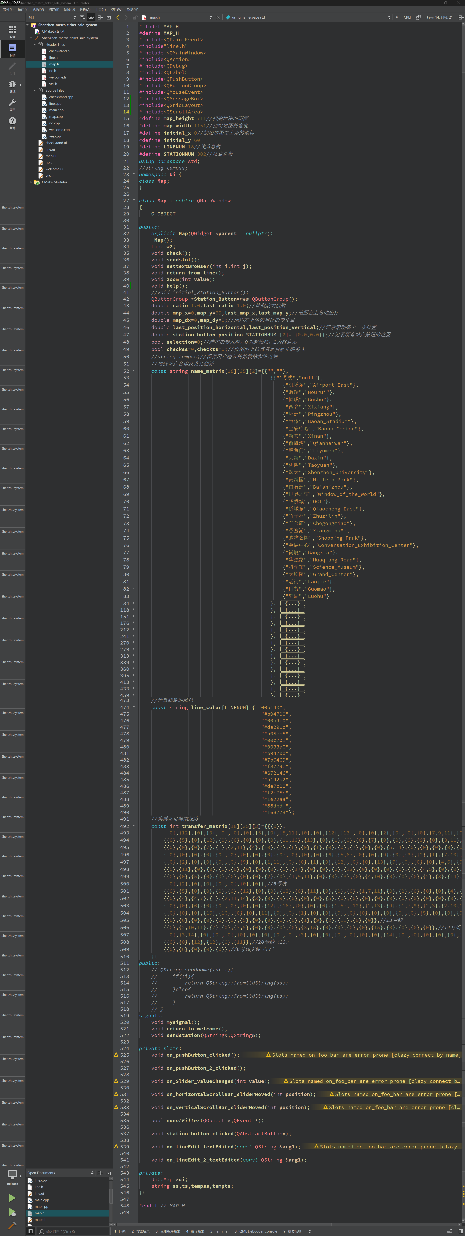
1. **选站界面 (map)**

****

选站界面的ui设计工作量主要集中在需要手动放置三百多个单选按钮并试图对齐地图上对应的位置。由于qt designer不支持放大界面，对齐的难度相当大，即使肉眼感觉已经对齐，实际缩放时仍会有所偏移；同时，由于按钮的名称只支持英文，而且qButtonGroup必须手动导入。为了后面方便确定选项等考量，笔者决定自行写一个小程序，辅助将原来的地铁名称矩阵扩充为中英文对照表，并生成三百多行的qButtonGroup的导入函数。



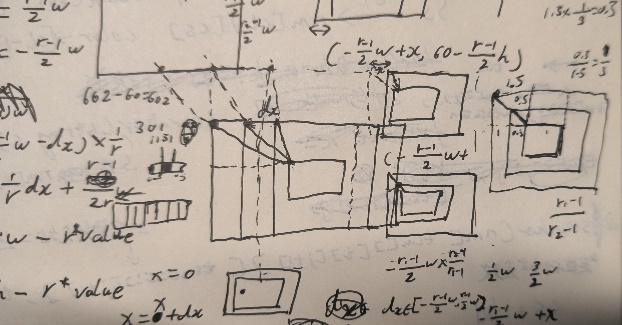
map.h:主要是常量与变量的定义



map.cpp:

控制地图与按钮的缩放与平移：

问题与解决算法：



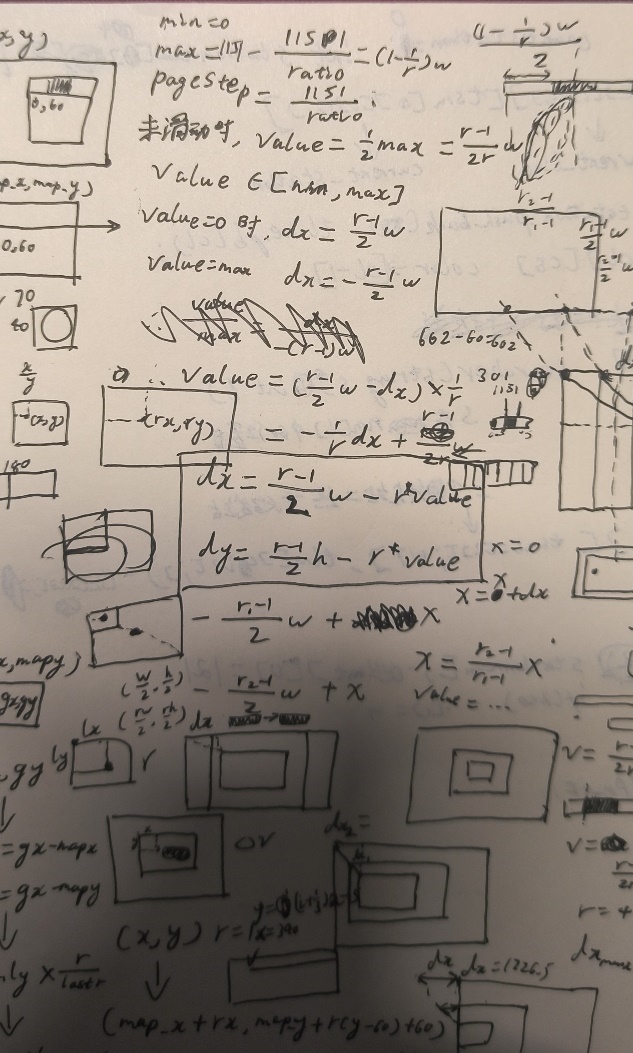
1. 当滑动条代表的值改变时，地图的左上角的位置及其长宽的改变量与上一次的值有关，这个值同时也受图片平移的影响。如上图所示，为了保证经过平移后的缩放不至于将图片带离窗口，左上角坐标在缩放时必须沿着不同的“轴”移动：例如，地图没有位移时，左上角沿着图片的对角线移动，而当窗口左侧与地图左侧重合时，左上角沿着y轴移动。

当然，本程序并没有计算左上角沿着哪条轴移动，而是利用相似三角形来分别计算正常缩放的比例和平移带来的改变量dx,dy乘以相似比，最后求和即可。其公式描述为：

map\_x=map\_dx+initial\_x-(map\_width\*ratio-map\_width)/2;

map\_y=map\_dy+initial\_y-(map\_height\*ratio-map\_height)/2;

其中ratio为用户控制的缩放比例。

1. 图片在缩放时，滚动槽中的滚动条的左（上）端位置、滚动条最大值、滚动条步长（这些参数将自动决定滚动条的长度）均会发生改变，且滚动条的左（上）端位置（下记为value）与其上一次位置有关。例如，当图片没有位移并放大到两倍时，横向滚动条的max为0.5\*width，步长为0.5\*width，value=0.5\*max；其它倍率及位移后的公式如下图所示：

有了以上公式后，我们即可掌握value与dx（dy）的转换关系，进而正确设置滚动条和滑动条对地图的影响。

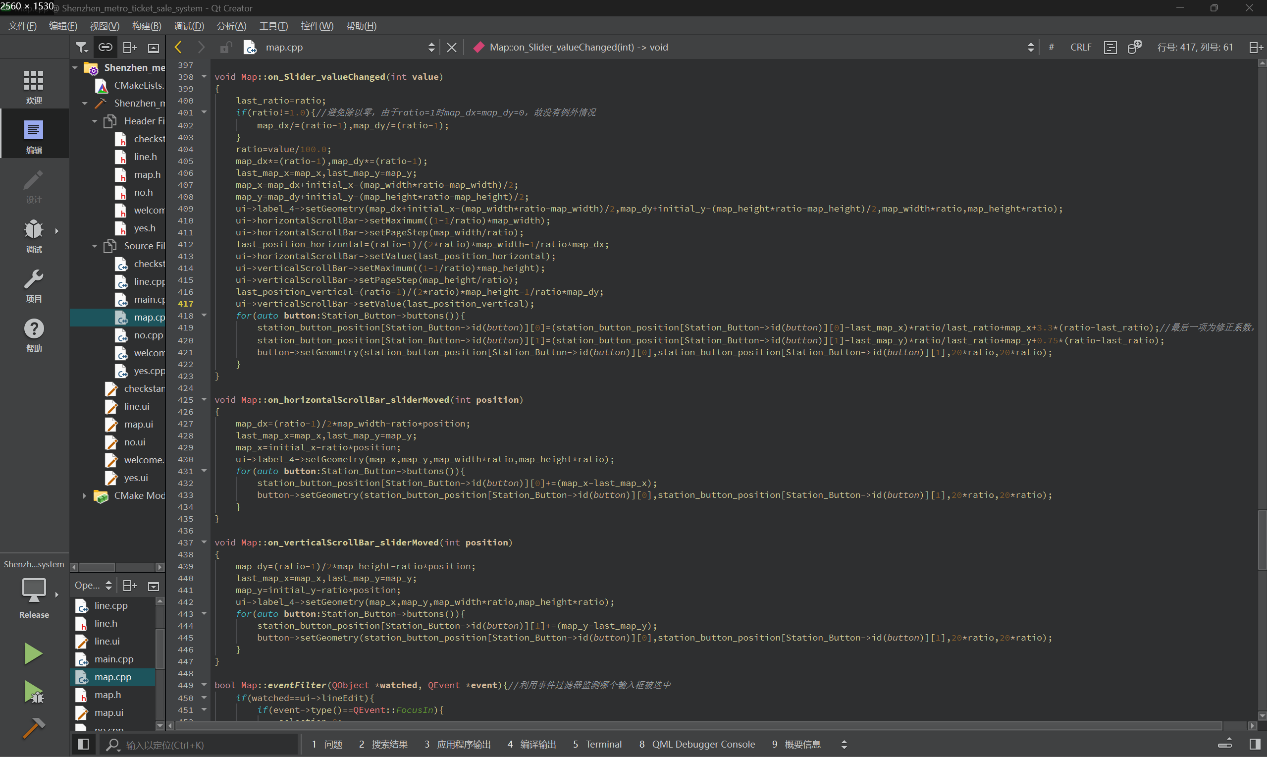
1. 正确缩放及位移地图后，所有站点按钮也要跟随移动。具体而言，站点按钮左上角相对于地图左上角的坐标（记为(X,Y)）的比值Y/X应始终保持不变，这将保证按钮大致保持在地图对应的站点处。对于每一个按钮，其坐标值(gx,gy)与地图左上角坐标(mapx,mapy)的关系为：

gx=X+mapx,gy=Y+mapy

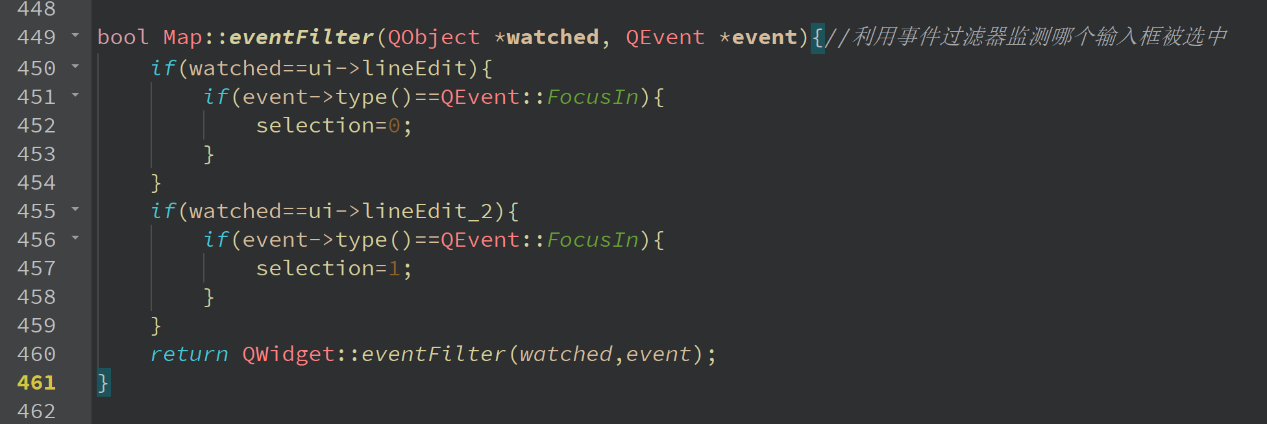
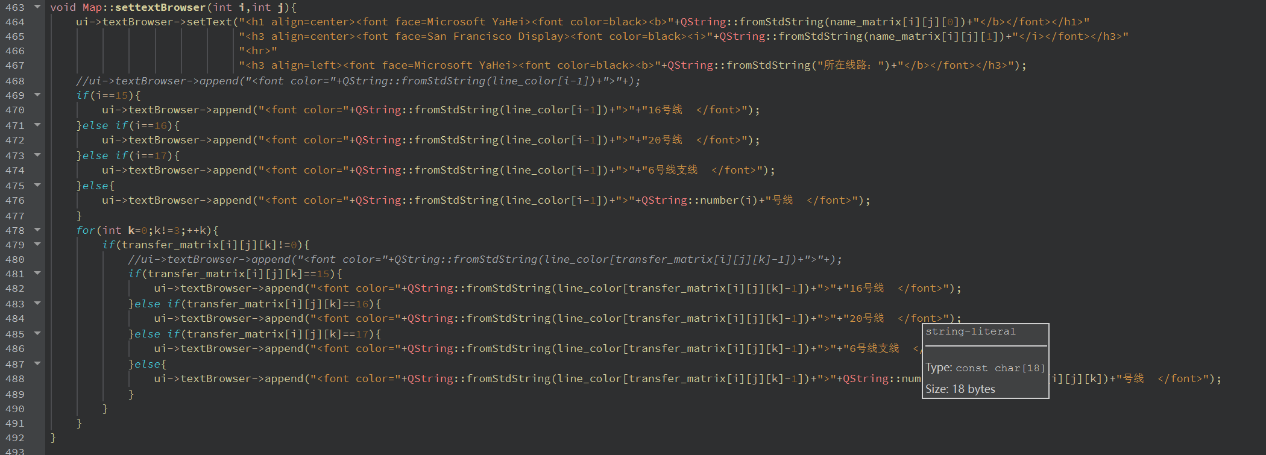
在缩放时，X,Y均乘以ratio/last\_ratio，再用上述公式换算出新的(gx,gy)；在平移时，只需让gx加上map\_x的增量即可，gy同理。

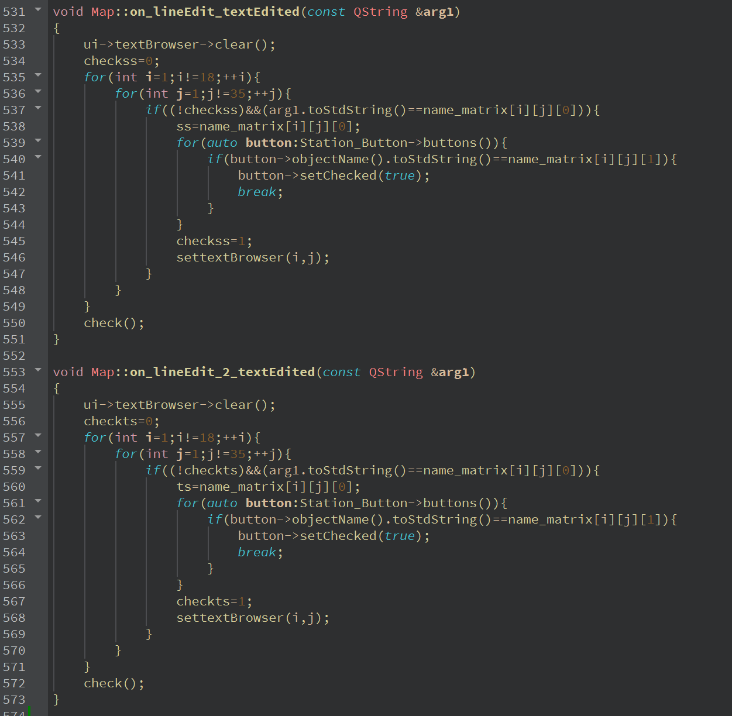
然而，在实际操作时，发现按钮在缩放时仍会偏离站点，虽然是过程无关的（即相同缩放比例下的偏移度相同）。因此，考虑在按钮坐标上加上一修正系数f(ratio)-f(last\_ratio)来保持过程无关，同时尽量减少偏移度。

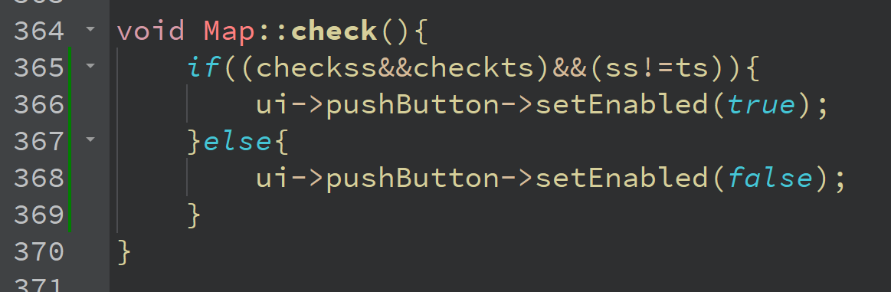
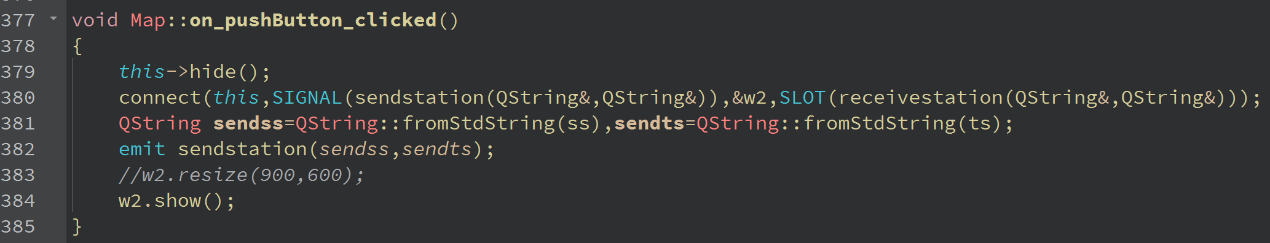
实现代码如下所示：



选站与显示信息：

1. 通过拦截事件过滤器来检测哪个输入框被选中：
2. 接受线路及站点值，在文本框中打印信息：
3. 点击按钮时，将站点名输出到输入框里，并调用文本框打印：
4. 输入站点名时，若存在该站点，则使对应按钮被选中，并调用文本框打印；否则不做处理：

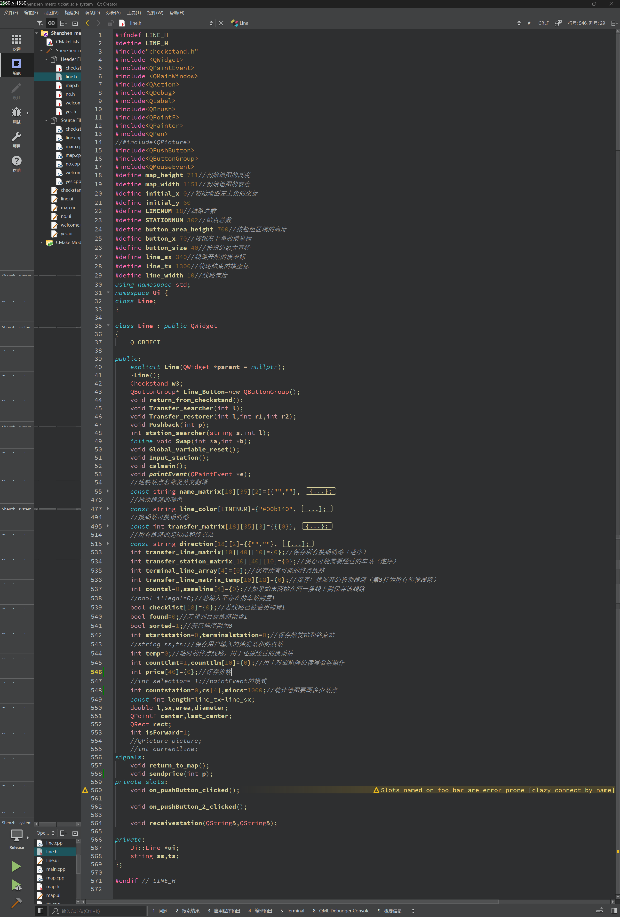
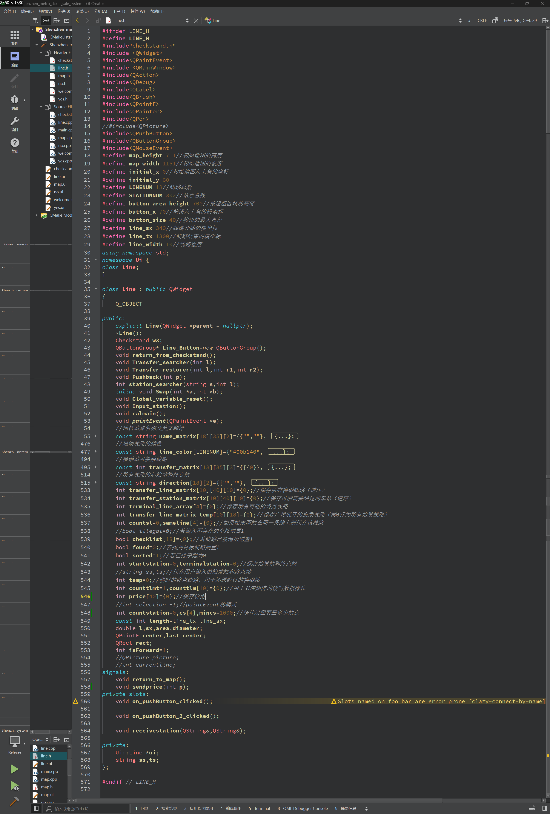


1. 仅当起始站和终点站都被正确输入且不相同时才能点击确定：
2. 显示帮助文档：
3. 利用信号与槽将起始站与终点站的名字发送给选线界面：
4. **选线界面 (line)**

本界面ui较为简单，主要ui显示集中在paintEvent的绘制中，故不作展示。

本界面主要根据接收到的起始站与终点站的名字，计算出所有的方案，然后利用全局变量在最后调用重写的paintEvent进行方案的图形化展示。

line.h:常量及变量定义



line.cpp:

方案的计算：

首先调用Input\_station，将两站相同的线路都填入terminal\_line\_array中；

若tla[0]!=0，即起始站和终点站在同一条线路上，则直接按tla[i]里的所有线路转换两站的坐标后，计算并填写price[i]，再冒泡排序后开始打印方案。

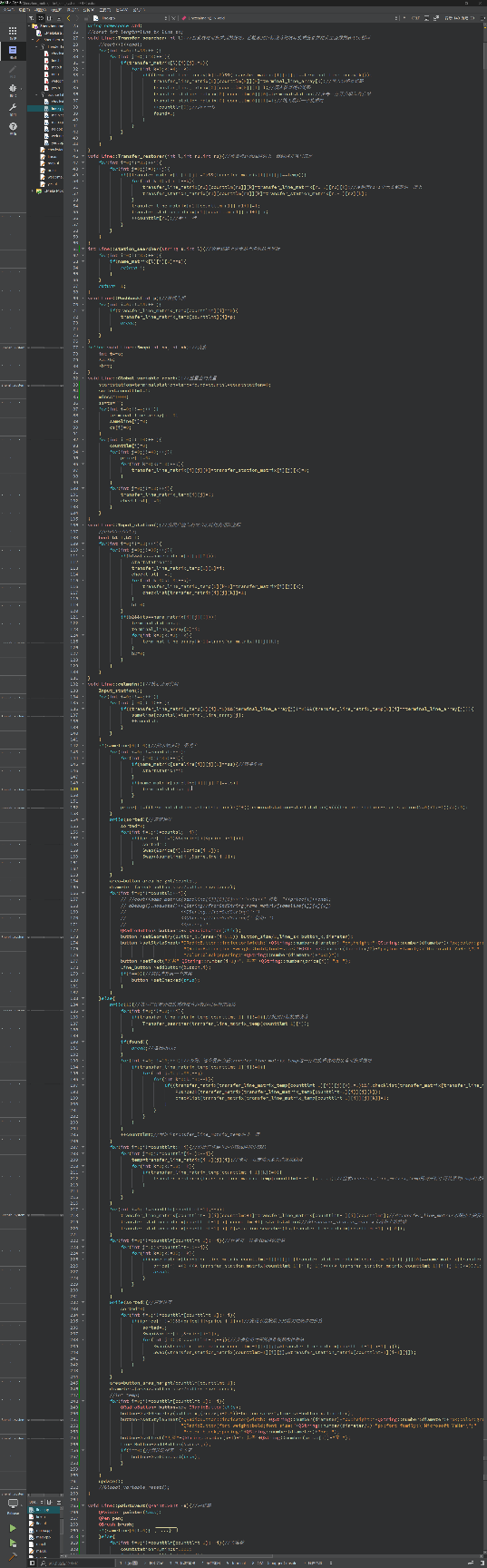
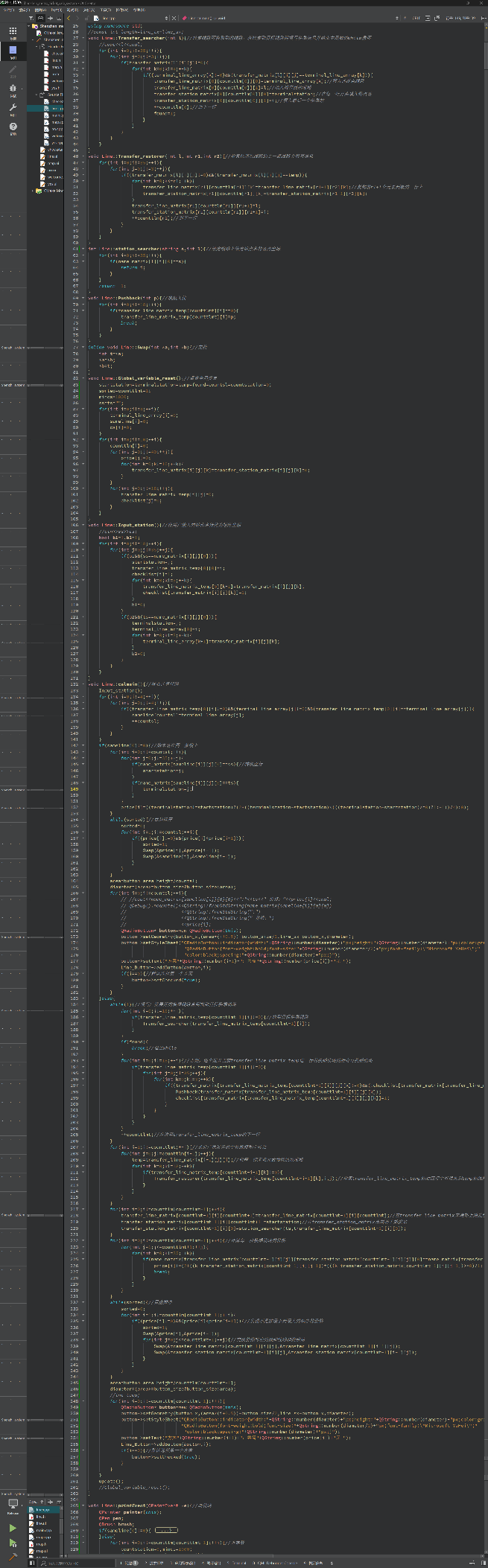
否则，不断调用Transfer\_searcher函数寻找可能的换乘线路：若找到换乘线路，将终点线路、被查线路、终点站、最后一个换乘站分别填入transfer\_line\_matrix和transfer\_station\_matrix，并跳出循环；否则，应用广度优先搜索的思想，逐个展开当前transfer\_line\_matrix\_temp这一行的换乘线路的所有可换乘线路（transfer\_line\_matrix\_temp已在Input\_station中初始化），调用Pushback对transfer\_matrix模拟入栈操作，填写transfer\_line\_matrix\_temp中的线路，同时所有被填入的线路都会在checklist中置1，以避免重复搜索造成死循环。

广优搜索结束后，开始还原中转线路和中转站，取transfer\_line\_matrix的每一行末端的元素作为待还原线路，检索transfer\_line\_matrix\_temp的对应行中可换乘到待还原线路的线路，并调用Transfer\_restorer填写transfer\_line\_matrix和transfer\_station\_matrix的信息。

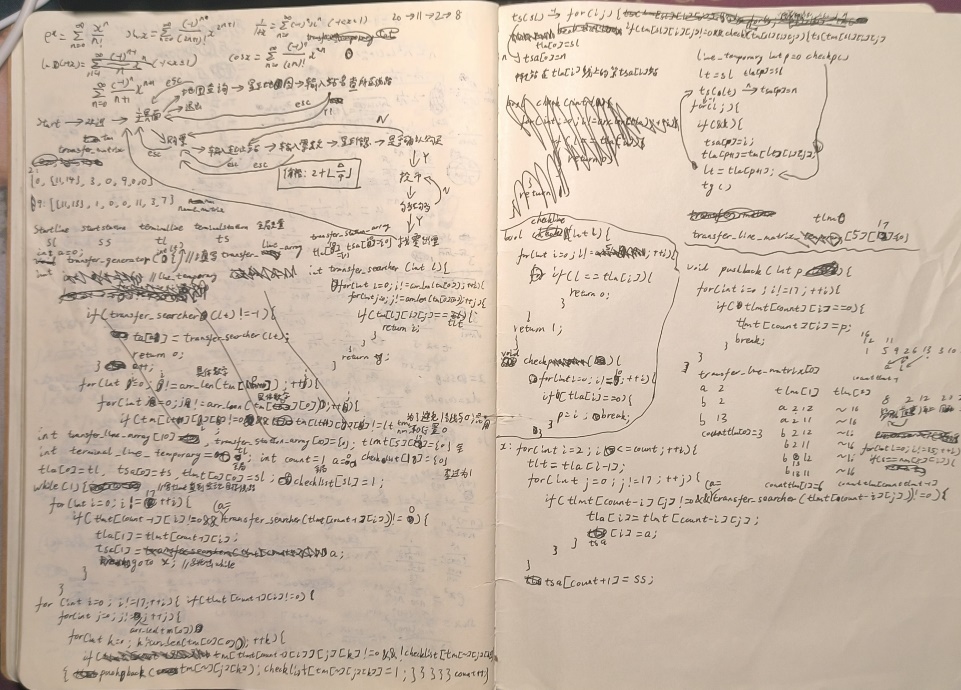
还原结束时，在transfer\_line\_matrix和transfer\_station\_matrix末端补上始发线路和始发站，同时调用station\_searcher修正终点站的站点坐标。之后，对每一个方案计算并填写price[i]，再冒泡排序后开始打印方案。

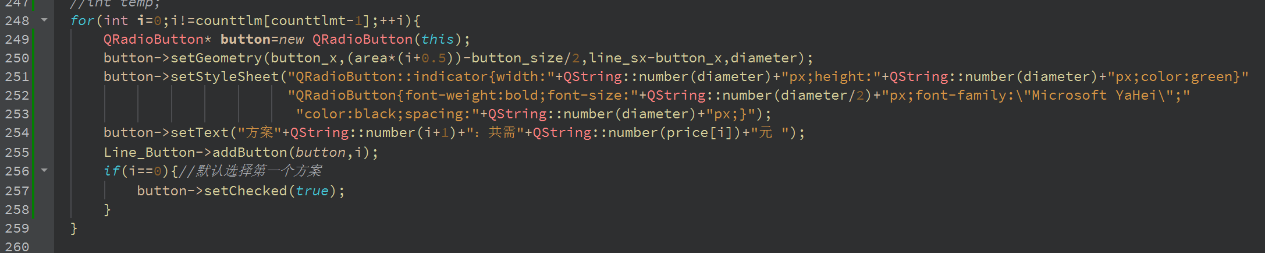
总体来说，由于应用了广度优先搜索的思想，这个算法能保证所有换乘方案的换乘次数最少，并且不会有遗漏的方案。

以上过程的实现代码如下图所示：



思路及草稿展示：

  
根据方案设置单选按钮的位置和对应文字（价格）：

将y=0-700的区域按方案数分割，将按钮放在分割域的中线，再打印文字。按钮直径diameter有最大值，若方案过多导致区域过小，可以利用diameter调整按钮及后面图形化的各项参数。下面仅以起始站和终点站不在同一条线路上为例，起始站和终点站在同一条线路上时大同小异。

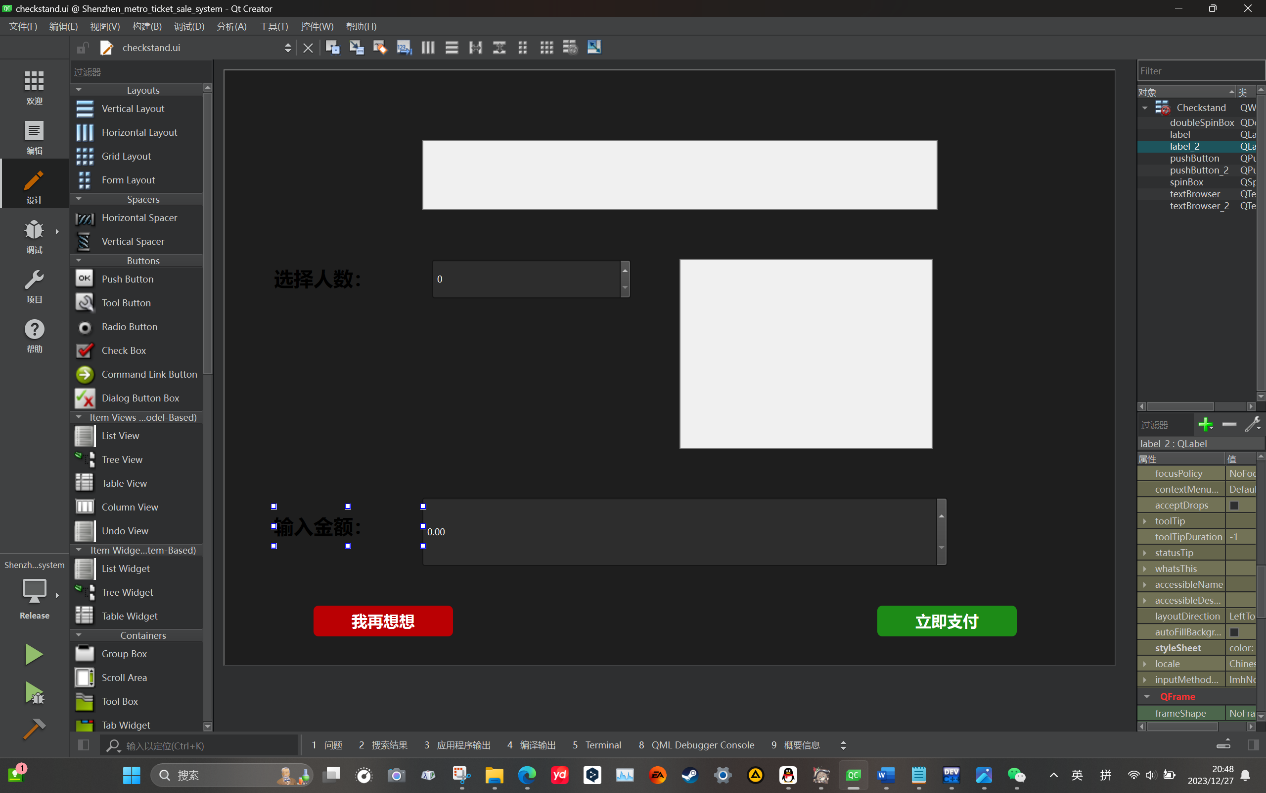
方案的图形化显示：

由于qPainter等绘图工具只能在paintEvent中使用，而paintEvent无法传参，无法反复被update()唤起，故只能在最后调用全局变量来绘制。

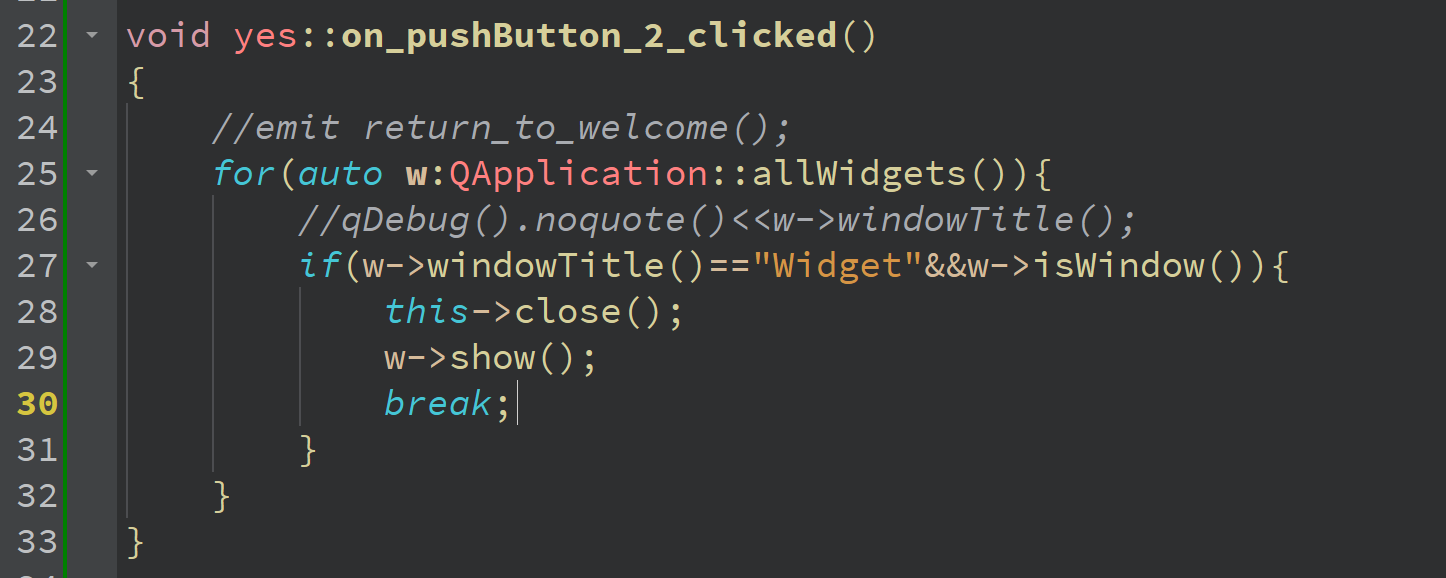
主要逻辑：统计每个方案中的经过站点总数及每条线路的站点数。绘图区域length除以站点总数即为两个站点间线段的长度。对于每个站点画一个实心圆，在下面竖排显示站点名；若站点为起始站、终点站、中转站，则圆的半径和站点名都会变大，对于起始站和中转站还会在右上角打印当前线路的乘坐方向。下面仅以起始站和终点站不在同一条线路上为例，起始站和终点站在同一条线路上时大同小异。

用户选完方案点击确定时，将方案的价格发送给支付界面，原理与map相同。

1. **支付及判定界面 (checkstand, yes, no)**



该部分逻辑较为简单，用户输入人数和金额后，点击支付按钮会进行判断，若金额足够则调用yes，否则调用no，调用时将差价的绝对值发送给对应窗口。

yes中返回欢迎界面的逻辑较为特殊，由于传统的返回只能在相邻两个窗口中实现，不适用于yes，因此，采用暴力搜索所有已打开的窗口，若搜到欢迎界面的窗口名即可获取其窗口指针，进而调用close和show。代码如下：

# 6 调试与测试

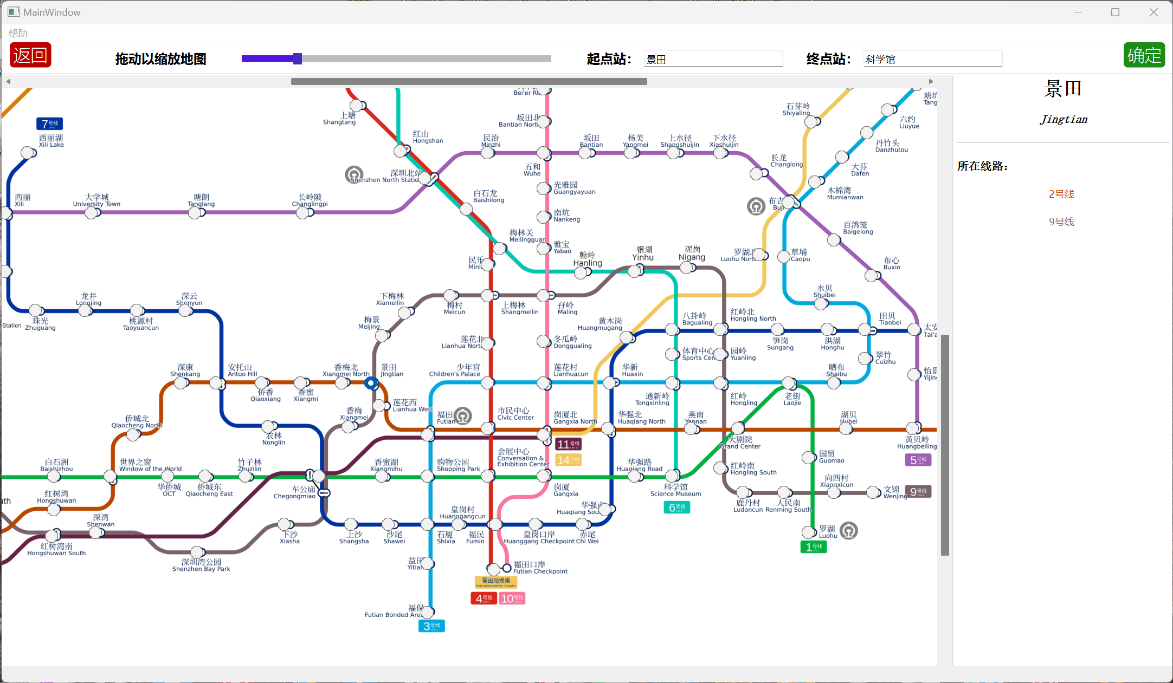
## 6.1调试过程

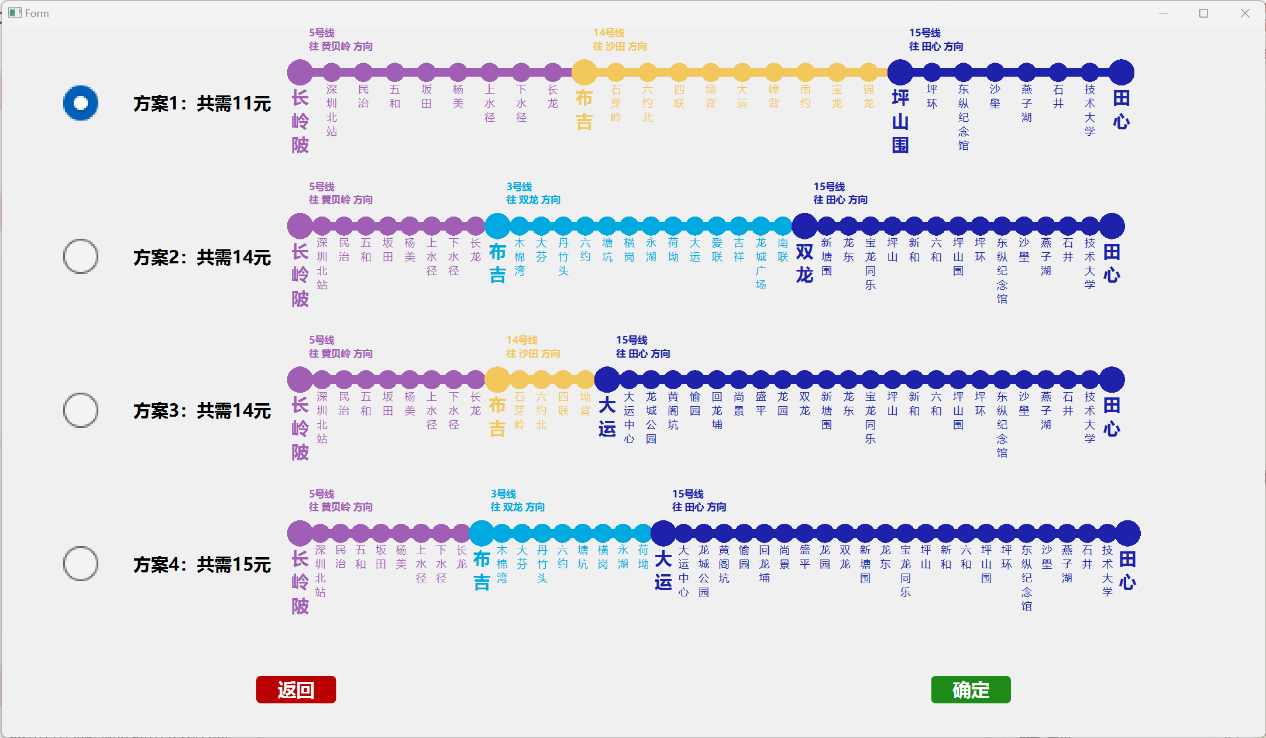
主要采用qDebug()调试，在此过程中学到了很多它的用法，例如.noquote()关闭双引号等。

## 6.2测试结果（包括界面，数据，结果）

测试结果与之前展示的大体一致。







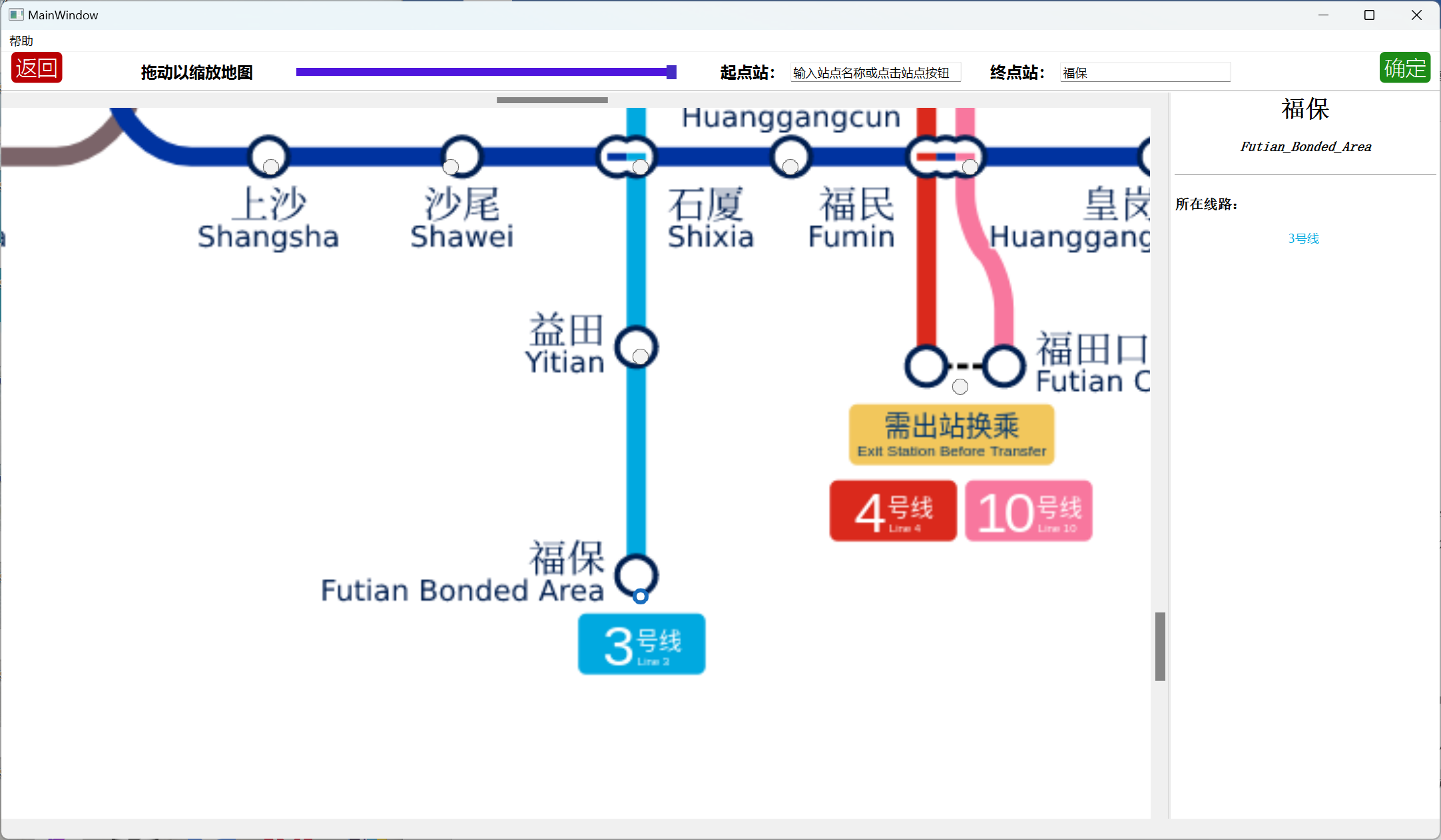




# 7 编程中遇到的问题

编程中遇到的问题已在前面有所提及，下面讲讲目前程序中存在的问题。

## 问题1

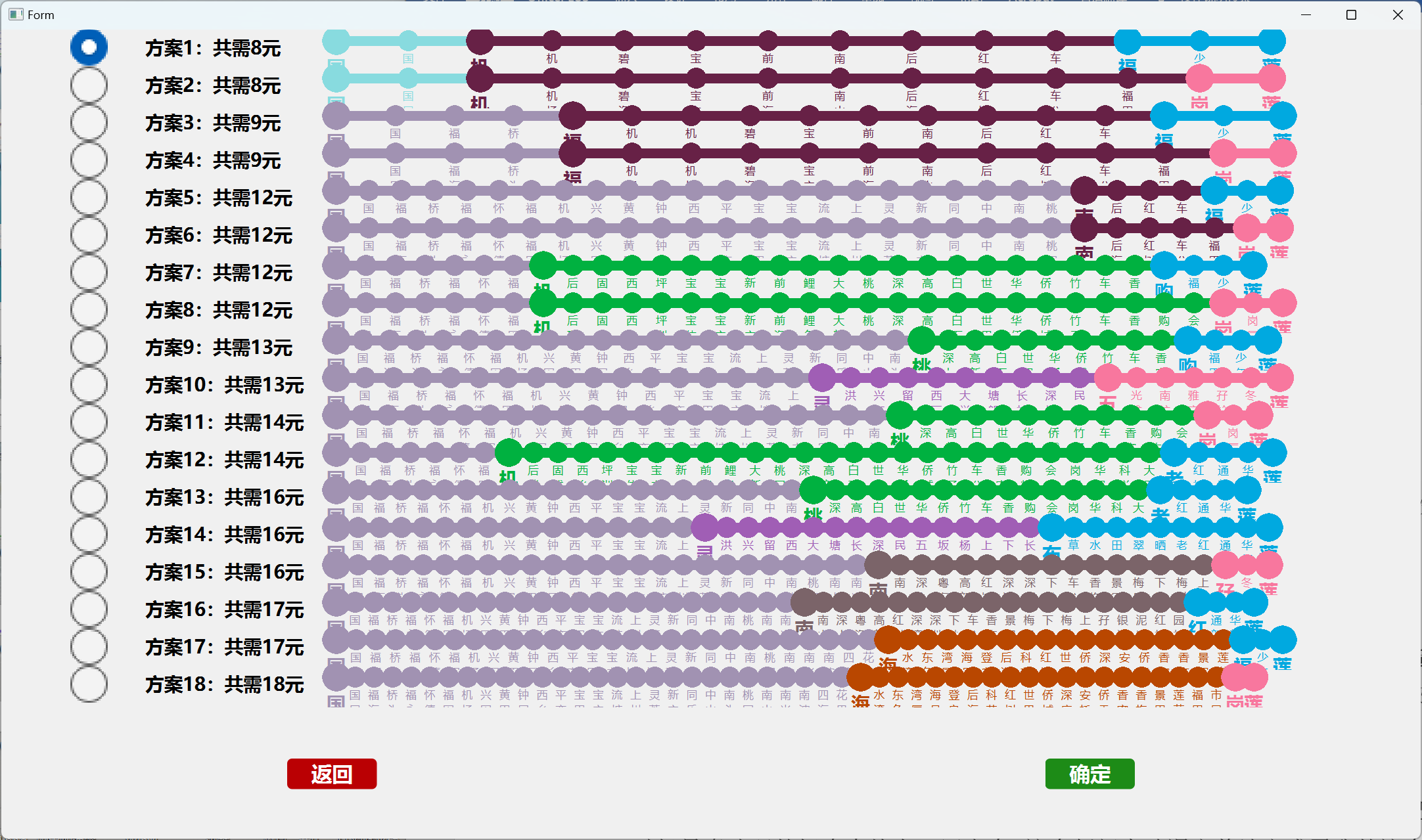


部分站点的按钮偏移过多，需要之后慢慢修正。

## 7.2 问题2

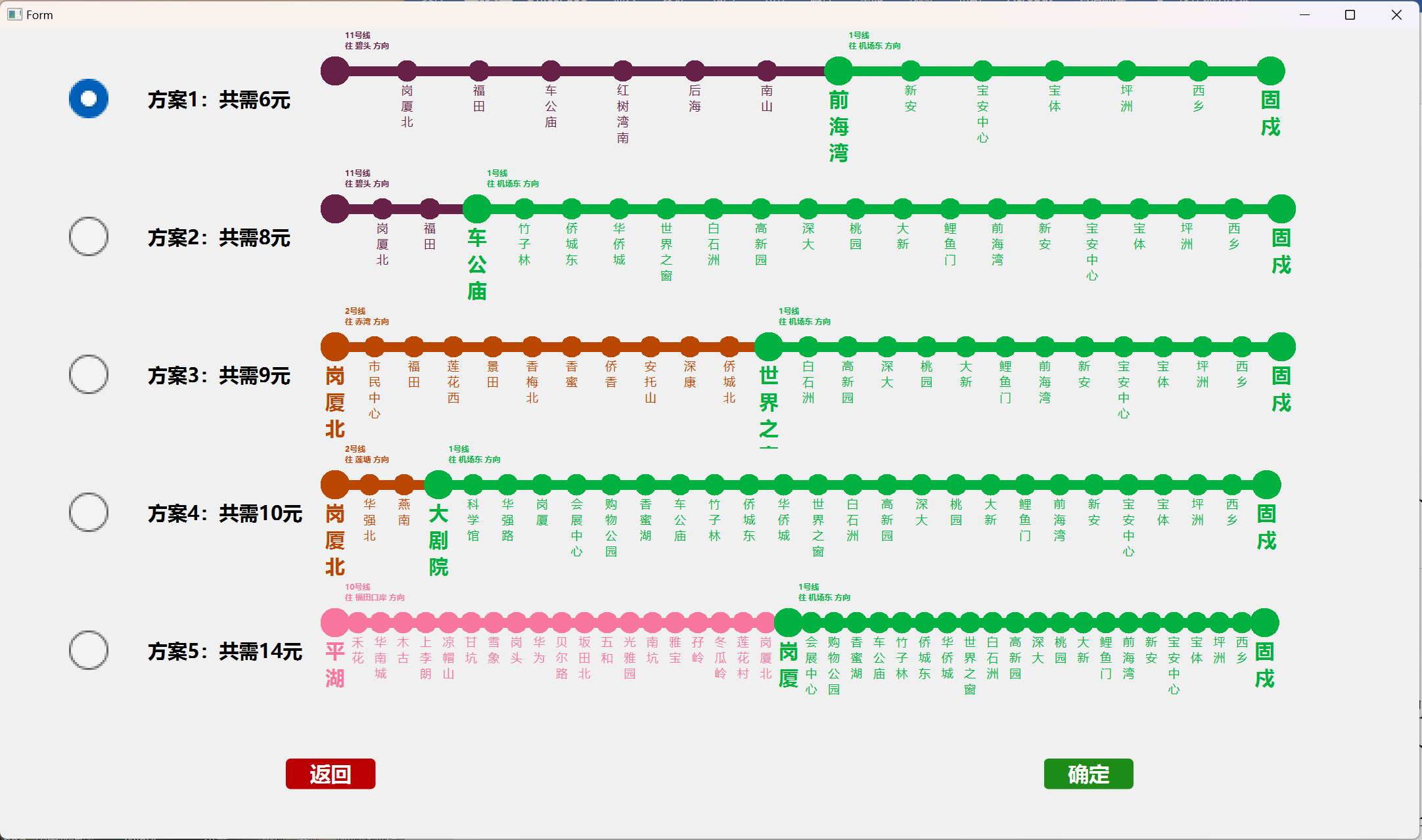
选站界面的水平滑动条不够灵敏，经常无法拖动，增大宽度或许可以解决。

## 7.3 问题3



方案过多时，显示效果不理想，可以在右边添加一排按钮，点击按钮弹出一个小窗口，在该窗口中单独绘制对应的方案。

## 7.4 问题4



部分方案的起始站点名称没有绘制，且一些换乘线路出现错误，推测为transfer\_station\_matrix的起始站点需要修正。

# 8分析总结与心得体会

我从初一开始便接触c语言，初二开始接触c++，在这期间参加过不少信竞，写过不少程序交作业，但最令我印象深刻的，是我在草稿纸上写完的一元二次方程步骤模拟器，那是我第一个纯粹凭兴趣写出来的程序。彼时我还没有电脑，每周只有两个下午可以进机房实现程序，所以我废寝忘食地在草稿纸上设计框架、推演算法、脑测程序，甚至在德克士快餐店里边吃边写，不亦乐乎。现在想来，正是那段靠笔和纸写程序的时光，极大地锻炼和丰富了我的逻辑思维和推演能力，也让我深刻感受到了c++的魅力与能力。

今天早上，当我敲完模拟深圳地铁自动售票系统的最后一行代码，成功将整个项目封装成一个可执行文件后，我又想起了那个在德克士里边啃汉堡边写代码的身影。一缕汉堡的香味传来，我不禁感叹，是啊，到头来，兴趣才是最好的老师。

本项目所有源代码地址：https://github.com/novemberinnorth/Simulate\_Shenzhen\_Subway\_Ticketing\_System