

# 目录

## CONTENTS

01

项目概述

02

系统设计

03

关键点介绍

04

视频演示

05

调试与总结



Part.01

# 项目概述

## 项目概述

### 背景

随着城市地铁线路的不断扩展和客流量的增加，传统的人工售票方式已经无法满足需求。自动售票系统可以实现快速购票、减少排队时间，提高地铁运营效率和乘客满意度。

### 编写目的

提高购票效率  
减少人为错误  
便于数据管理和分析

### 任务概述

研发一个模拟地铁自动售票系统，模拟实现现实中地铁自动售票系统的一些功能。  
本项目中，模拟地铁自动售票系统包括三个子项目，分别是“地图选点”、“查看方案”、“找零”。



Part.02

# 系统设计

## 软件模块功能流程图

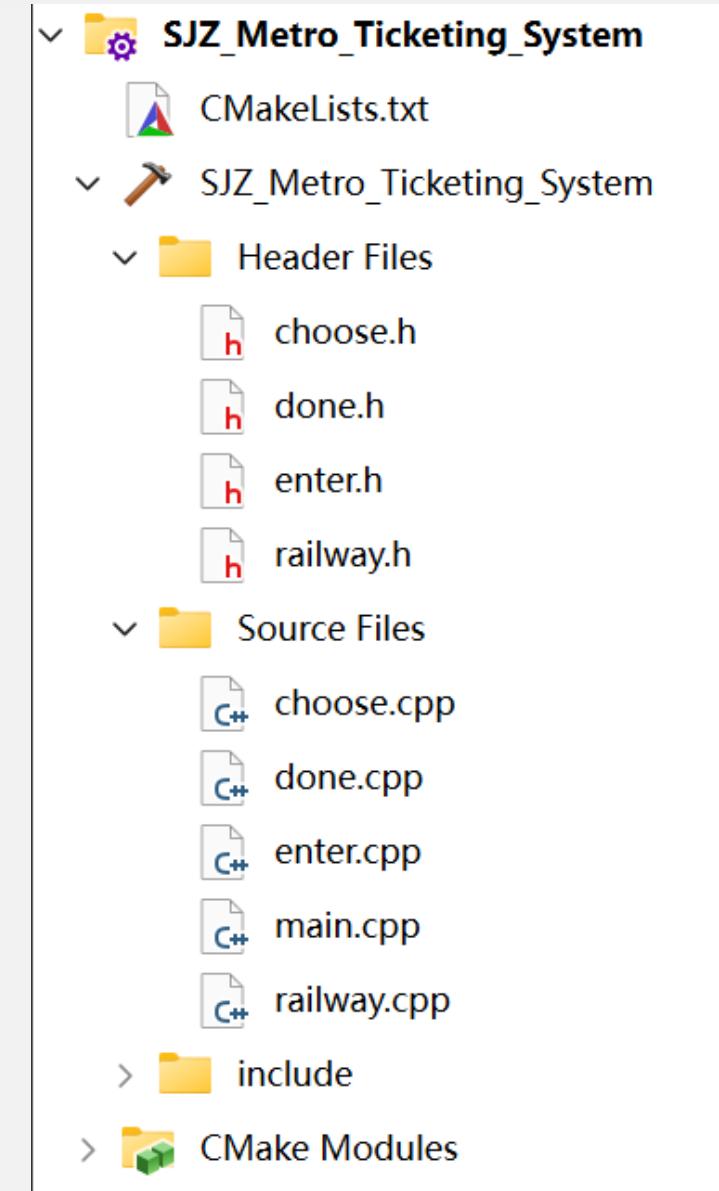


欢迎页面: **enter.h/enter.cpp**

地图选站界面: **railway.h/railway.cpp**

支付界面: **choose.h/choose.cpp**

找零界面: **done.h/done.cpp**





Part.03

## 关键点介绍

(1) 地图的缩放和移动可以通过**鼠标的滚轮和左键拖拽**来实现，也可以通过右下角的**按钮控制**地图的放大、缩小以及重置。

```
38  ↗ void railway::iniUI2() { ... }
270
271  ↗ void railway::do_return() //返回开始界面 { ... }
275
276  ↗ void railway::reset_para(void) //初始化地图 { ... }
302
303  ↗ void railway::update_para(void) //更新地图 { ... }
409
410  ↗ void railway::do_fd() //放大 { ... }
416
417  ↗ void railway::do_sx() //缩小 { ... }
423
424  ↗ void railway::do_cz() //重置 { ... }
430
431  ↗ void railway::do_show_line() //单独显示线路 { ... }
436
437  ↗ void railway::paintEvent(QPaintEvent *event) //绘制地图 { ... }
445
446  ↗ bool railway::event(QEvent *event) //鼠标拖动地图 { ... }
491
492  ↗ void railway::wheelEvent(QWheelEvent *event) //滚轮放大地图 { ... }
506
```

相关函数

(2) 选择站点时先点击起点站或终点站的输入框，然后可以通过直接在地图上**选点或输入**的方式选站，输入时可以**自动填充**。界面右下方可以预览所选站点的**中英文名和所在线路**。

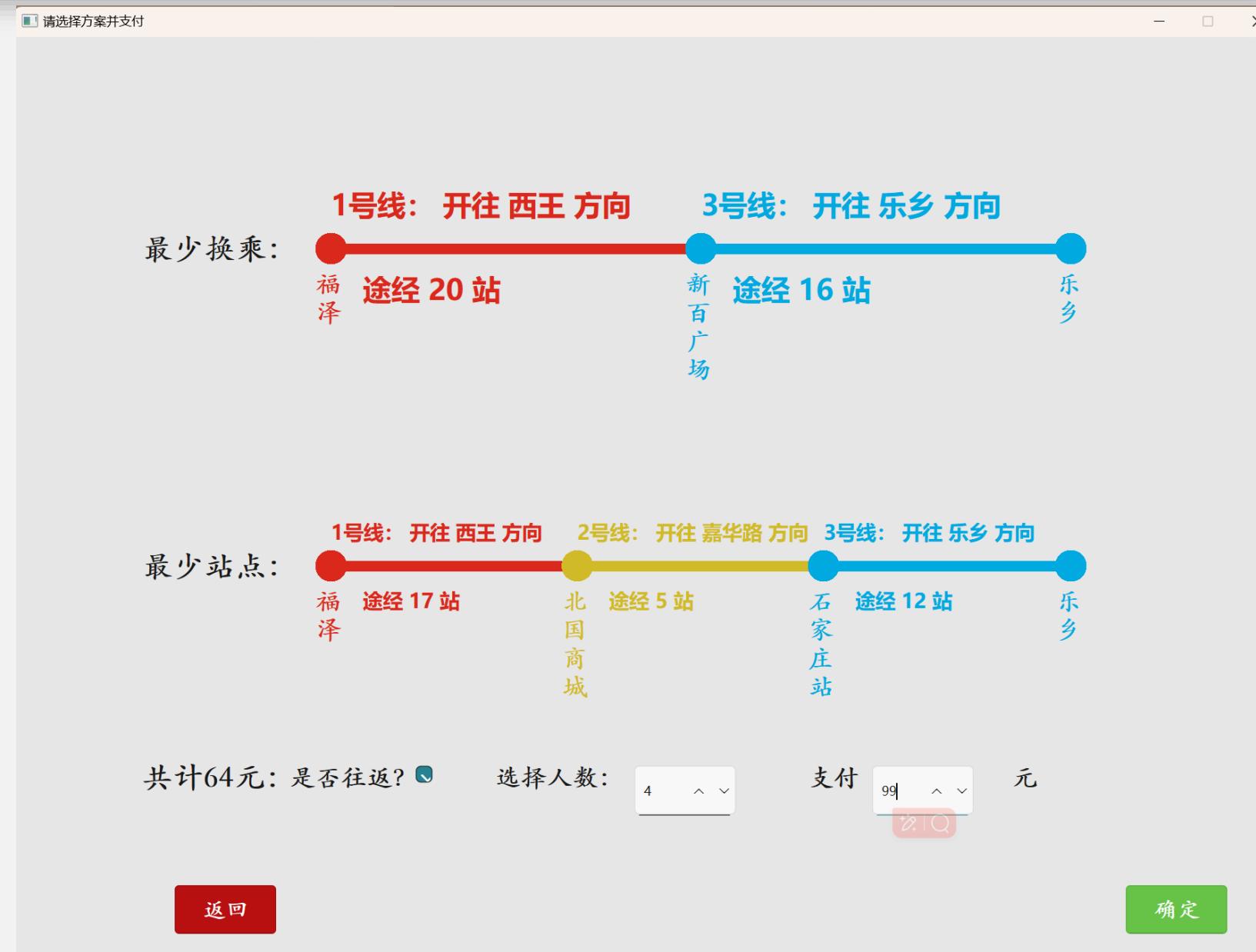
```
507 ▶ void railway::search() //站点自动填充 [...]  
523  
524 ▶ void railway::do_station_btn(QAbstractButton* button) //地图选站: 填入 [...]  
550  
551 ▶ void railway::check() //检查两个站是否正确 [...]  
563  
564 ▶ void railway::do_edit_qd(const QString &string) //直接编辑起点站时 [...]  
591  
592 ▶ void railway::do_edit_zd(const QString &string) //直接编辑终点站时 [...]  
619  
620 ▶ bool railway::eventFilter(QObject *watched, QEvent *event) //地图选站: 确定起点或终点 [...]  
642  
643 ▶ void railway::filltext(int i,int j) //显示所选站点详细信息 [...]  
662  
663 ▶ void railway::do_send_station() //点击确定传输两个站点 [...]  
673
```

## 相关函数

(3) 系统展示**最小换乘线路**（可能不止一条）和**最少站点线路**，帮助用户更高效地规划行程。

其中，线路中会显示起点、终点站、换乘站、途经站点数、每条线对应的方向。

**只显示途经站点数而不显示途经站点名称**，在途经站点数较大时的界面更清晰。



(4) 支付界面只会给出一个价格，这个价格是按照经过的站点数最少的情况来计算的。

【注：本程序的价格计算公式为： $2 + [\text{同一线路上起始站与终点站间的站点数} / 5]$ （向下取整），而现实中为按最短里程分段计价，二者有所不同】

```
153 void choose::all_reset() //重置所有 {...}  
176  
177 void choose::sameline() //检查是否在同一条线路，在的话直接得出价格 {...}  
212  
213 /*  
214     现实票制方案：采取里程分段计价的票制方式，按照“分级递进，递远递减”的原则，发行单程票和储值票。  
215     单程票价方案：起步价 6公里 2 元， 6~20公里每递增7公里加1元， 20~36公里每递增8公里加1元， 36~54公里每递增9公里加1元， 54公里以上每递增10公里加1元。  
216  
217     采用：起步2元，每5站加1元 给出一个价格，1或2个最少换乘方案，1个最少站点方案  
218 */  
220 void choose::calculate() {...}  
293  
294 void choose::paintEvent(QPaintEvent *event) //画线路图 {...}  
946  
947 void choose::do_return() //返回选站界面
```

## 相关函数

(5) 支付界面可以选择是否往返。输入的人数限制为1-50之间的整数，输入支付金额限制为1-1500之间的整数，且只有付款金额不低于所需金额时方可结算。

```
93  ▶ void choose::receive_station(QString &qdz, QString &zdz) //接收站点 { ... }  
103  
104  ▶ void choose::do_double(bool checked) //往返价格加倍 { ... }  
116  
117  ▶ void choose::do_change(int people) //价格乘人数 { ... }  
126  
127  ▶ void choose::do_check() //进行检测 { ... }  
131  
132  ▶ void choose::check_enable() //检测付的钱是否足够 { ... }  
139  
140  ▶ void choose::do_send_money() //传输找零 { ... }  
152
```

---

## 相关函数



Part.04

## 视频演示



Part.05

# 调试与总结

## 调试过程

(1) 问题：发现有时换乘两次比换乘一次便宜，如从“博物院”到“孙村”。也即“广度优先搜索”（BFS）的方法找到的是最少换乘的方案，不是最少站的方案。

解决：采用计算最少站即最小金额的算法。

(2) 问题：人数为0时也能购票，且人数上限和购票金额上限都为99，若金额超过99元则无法支付。

解决：输入的人数限制为1-50之间的整数，输入支付金额限制为1-1500之间的整数，且只有付款金额不低于所需金额时方可结算。

(3) 问题：第一次是有找零，第二次刚好不找零的时候，而且两次之间不退出系统时，显示的找零钱数有时候是第一次的。

解决：优化了找零部分的代码，不找零时不会出现文字提示。

## 编程中遇到的问题

### 问题1

开题报告中的有些部分难以实现。因此在实际过程中采取代替方案。

### 问题2

采用各站点间实际的金额较为繁琐。因此本程序的价格计算公式为： $2 + [\text{同一线路上起始站与终点站间的站点数}/5]$ （向下取整），而现实中为按最短里程分段计价，二者有所不同。

### 问题3

地图选站界面按钮本身不会缩放，因此在地图缩放时会偏离地图中的站点。

### 问题4

有些组件的堆叠可能出现次序上的问题。并且在系统深色模式下有些组件的显示会出现问题。

## 分析总结与心得体会

在本项目中，我们选用QT框架，通过C++来模拟一个地铁自动售票系统。这个过程既考验了我们的编程技能，又是一次团队协作与问题解决的实战演练。

在合作过程中，我们深刻体会到了团队协作的重要性。每当遇到问题时，我们会一起讨论，共同寻找解决方案。这种团队精神提高了我们的工作效率。

此外，这次实践也让我们对QT框架有了更深入的了解。通过实际操作，我们掌握了QT的基本使用方法，学会了如何设计并开发一个功能完善的图形用户界面。这不仅巩固了我们的编程知识，更拓宽了我们的编程视野。

未来的程序还可以向以下几个地方优化：可以优化为支持多种支付方式：无现金交易，用手机通过微信或支付宝扫码支付，迎合乘客的支付习惯；可以加上背景音乐，且用户可以控制播放与暂停；可以加入音效或语音播报。

总之，这次双人合作编程项目是一次充满挑战与收获的经历。它不仅提升了我们的编程技能，更让我们学会了如何在团队中更好工作。



# 请各位老师指正！



答辩人：zzk (ui设计与框架构建)  
xlh (调试改进与报告撰写)



时间：2024.12.15