“铁路行程分析系统”项目报告

PKU Trail Blazers 组 组长：刘宬泽 组员：张嘉宸、张桐嘉

一、程序功能

本程序是一个基于 Qt 框架开发的铁路出行数据管理与分析工具，旨在帮助用户便捷地管理个人铁路行程并生成可视化的年度报告。系统提供全国铁路地图的构建、行程导入与可视化、个人出行网络建模、以及智能出行建议生成等功能。用户可以通过图形界面导入出行记录，系统将自动构建用户的行程图谱，统计访问城市、出行频率与途径铁路，绘制个性化的用户画像，最终生成铁路行程分析报告，并基于数据提出合理化的下一年度出行建议。本程序适合作为 C++ 与 Qt 技术结合应用的学习项目，功能结构清晰，易于扩展和优化。

在运行该程序时，开场首先播放一段62帧的动画，动画契合“铁路行程分析系统”的主题。主界面共有三个按钮：“添加行程”支持用户导入行程，自主选择出发地与终到地，经过的铁路由系统根据现实常识与BFS（宽度优先搜索）结合计算，并实现即时的点亮行程功能。点亮行程功能利用图层覆盖原理，点亮用户经过的铁路名称，增加了图形界面的吸引力。同时，该程序即时地将点亮行程后的界面图片予以导出并保存，使用户在关闭程序后仍然可以看到界面图片。

“生成报告”则支持自动生成用户的行程分析报告。报告以弹窗形式弹出，统计了用户“过去30天总出行次数”和“共涉及城市数量”，并给出用户“常见出发城市TOP3”和“常见到达城市TOP3”，顺序按出行次数排列，出行次数相同时则按照出发或到达的时间先后次序。系统根据已经到访城市的城市类型，推荐不超过5个未访问城市，为用户后续的行程规

划提供参考。系统还将通过行程分析得出相应的用户画像，为程序增加趣味性。

“清除行程”则实现了全部行程的一次性清除。为防止误删，系统增加一确认弹窗，用户点击“确认”后才会进行删除。

（程序运行中主界面的截图如下）

二、项目各模块与类设计细节

TripPlanner.pro文件如截图所示：

```
QT += core gui widgets

TARGET = TripPlanner
TEMPLATE = app

SOURCES += \
    main.cpp \
    mainwindow.cpp \
    core/Time.cpp \
    core/City.cpp \
    core/Railway.cpp \
    core/Trip.cpp \
    core/Statistics.cpp\
    reportdialog.cpp\
    addtripdialog.cpp

HEADERS += \
    mainwindow.h \
    core/Time.h \
    core/City.h \
    core/Railway.h \
    core/Trip.h \
    core/Statistics.h\
    reportdialog.h\
    addtripdialog.h

FORMS += \
    mainwindow.ui\
    addtripdialog.ui\
    reportdialog.ui

RESOURCES += resources.qrc
```

本部分将逐文件对模块与类设计细节进行介绍。

main.cpp:主程序，无特殊内容

Mainwindow.h和Mainwindow.cpp:主界面程序，构建了Mainwindow类。

Mainwindow::Mainwindow()是构造函数，主要与动画播放与计时功能有关。首先初始化与动画播放有关的参数，并初始化定时器，随后调用loadAnimationFrames()函数以加载动画帧，加载成功后开始播放。

MainWindow::~MainWindow()是析构函数，关闭界面并停止动画播放定时器。

void MainWindow::loadAnimationFrames()是加载动画帧函数，通过绝对路径获取动画帧并进行排序，将每帧统一缩放为 800×600后进行播放。

void MainWindow::startAnimation() 是动画播放函数，将动画播放设置为10fps。

void MainWindow::onAnimationFrameTimeout()处理每帧动画播放完毕后的情况。若动画未结束，则显示下一帧并适应窗口大小；若动画已结束，那么显示主界面。

void MainWindow::startMainInterface()是主界面显示函数，初始化UI控件并显示主界面内容。

void MainWindow::onGenerateReport()是点击“生成报告”后调用的函数，通过调用stats对象中的runAnalysis()函数以实现报告生成。

void MainWindow::on_addTripButton_clicked()是点击“添加行程”后调用的函数，弹出弹窗供用户选择出发与到达城市，在添加成功后弹窗提示“行程已成功添加”。

void MainWindow::on_clearTripsButton_clicked()是点击“清除行程”后调用的函数，弹出弹窗要求用户确认是否进行删除，若确认删除，在删除完成后提示“所有行程已清除”。

void MainWindow::resizeEvent(QResizeEvent* event)是调整界面大小的函数，加载背景图并将其缩放至800×600。

void MainWindow::initializeData()是数据初始化函数，完成城市名的初始化后创建对应的城市对象，并进行铁路对象的初始化。

void MainWindow::loadRailwayOverlays()是加载图层以实现点亮功能的函数。根据铁路名称加载相应的png图片，为通过图层覆盖实现点亮功能作准备。

void MainWindow::updateRailwayMap()是对铁路地图进行更新，在初始化函数、添加行程与清除行程后的逻辑中均有调用。首先加载背景图并缩放以适应控件，其次遍历所有行程，查找所经铁路，并绘制所有访问过的铁路图层，以此实现点亮功能。随后将叠加后的背景图（即点亮功能实现后的结果）进行导出与保存。

core/City.h和core/City.cpp是城市类的属性与成员函数。

City类有以下属性：名字、访问次数，并通过std::string getName() const和int getVisitCount() const来访问这两个私有属性。

Core/Railway.h和core/Railway.cpp是铁路类的属性与成员函数。

为实现点亮功能，本系统将相邻两个城市间的铁路（不经过任何其它城市）视作一个对象。故铁路有三个私有属性：名字、出发城市与到达城市。通过std::string Railway::getName() const来访问名字，通过std::vector<City*> Railway::getFullPath()来访问出发城市与到达城市，通过bool Railway::includes(City* a, City* b) const来判断两个城市是否在这条铁路的两端。

Core/Trip.h和core/Trip.cpp是行程类的属性和成员函数。

Trip类有出发城市、到达城市两个私有属性，并通过City* getStart() const和City* getEnd() const进行访问。

Core/Statistics.h和core/Statistics.cpp构建了系统类，它负责维护行程分析系统的日常工作，是整个程序的内在逻辑。系统类以所有的City、Trip和Railway分别组成的vector为私有属性。

void Statistics::addTrip(const Trip& trip)、void Statistics::addCity(City* city)和void Statistics::addRailway(const Railway& railway)用来在各自的vector中添加新的对象。

void Statistics::clearall()清除所有的行程。

void Statistics::getTopCities则是对城市的被访问次数进行排序，得出用户的“常见出发城市TOP3”和“常见到达城市TOP3”，顺序按出行次数排列，出行次数相同时则按照出发或到达的时间先后次序，以此为生成报告做准备。

std::vector<City*> Statistics::getUnvisitedViaCities() const是找到未访问的经由城市，以为生成报告时推荐城市做准备。首先获取所有访问过的城市，其次获取所有经由城市，最后找出所有未访问的经由城市，并借此选择推荐的未访问城市。

std::vector<Railway*> Statistics::findRailwayPathBFS(const Trip& trip)是为了解决“怎样根据起始城市与终到城市，找到经由的铁路以及经过的城市”这一核心问题，这对于生成报告和实现点亮行程功能都具有重要意义。这是一个经典的BFS算法，要注意的是建图是无向图，因此两个方向都要考虑。

std::string Statistics::evaluateUserType() const是实现用户画像的逻辑，通过出行次数等指标进行用户画像。

void Statistics::runAnalysis()是生成报告的函数，具体内容在介绍程序功能时已经介绍，不再赘述。

三、小组成员分工情况

1. 刘宬泽：QT界面的构建，设计主界面，实现弹窗界面，集成分析内容，实现点亮功能。代码的主要编写者，负责大部分类与函数的编写。报告的主要编写者。演示视频的录制者。

2. 张嘉宸：完成了行程导入部分的代码编写，并对项目代码作出了有益的修改。报告大纲的编写者。

3. 张桐嘉：完成了开场动画、背景铁路图的绘画制作。

四、项目总结与反思

作为《程序设计实习》的大作业，这是信科同学第一个需要通过小组协作来完成的项目。回首QT项目开发历程，我收获了许多经验，也得到了一些教训。

本QT项目开发过程中获得的经验主要有：

其一，增加了对QT的熟练度。由于初次接触，我对QT的使用方式并不熟悉，特别在开场动画与背景加载方面。在实践中，我明白了背景与其它资源图片应该放置的路径，学会写resources.qrc文件，并通过调整背景与动画图片的大小，更进一步了解了QT相关知识。

其二，增加了对计算机中图片存储等原理的掌握。在实现图层叠加时，最初增加的图片会覆盖背景图片，使得背景图片不再显示。通过求助ChatGPT，我发现这是因为增加的图片的背景是白色的而非透明的；于是我写了一个Python程序（pig.py）将图片背景颜色从白色改为透明，成功解决了这一问题。

其三，通过github的使用，增加了对git知识的了解，掌握了这一重要工具，收获颇丰。

此外，在这次项目开发过程中，我也得到了一些教训。

本次小组合作过程说不上愉快，在小组成员间任务分配方面存在问题。张桐嘉在完成绘图相关任务后，因后续任务问题与小组成员发生争吵，在项目尚未完成的情况下事实上退出小组（未再参与任何工作，未在小组微信群内发言）。这是项目工作的一个损失。原定张嘉宸撰写本次项目的报告，但他所写报告仅有不到两页，对于细节问题更是一笔带过，只能作为大纲，由我进行报告撰写。于是，事实上我完成了本次项目的大多数工作。除缺乏沟通外，部分组员工作态度不够积极也是导致这一后果的重要原因，恳请助教相应扣除相关组员分数。

通过本次小组作业活动，我在提升面向对象编程熟练度、熟悉QT使用与git仓库使用方面收获颇丰，在小组协作中得到了教训，在选择队友、增加沟通等方面有一些自己的心得。总体而言，本次项目的完成过程是我成长的过程，是努力与汗水的结晶。

报告撰写人：刘宸泽

2025年6月30日