

实验报告

一、任务说明：

本项目旨在创建一个基于高德地图 API 的地图应用程序，具备以下功能：

1. 使用高德地图 API 获取步行、骑行和驾车路线。
2. 提供菜单界面，路线查询界面和静态地图界面
3. 实现不同界面之间的跳转功能。

项目地址：[cxy000000/dl-course \(github.com\)](https://github.com/cxy000000/dl-course)

Demo 地址：[dl-course/expl/demo.mp4 at main · cxy000000/dl-course \(github.com\)](https://github.com/cxy000000/dl-course/blob/main/dl-course/expl/demo.mp4)

二、实验细节：

1. API

本应用程序使用高德地图 API：[概述-Web 服务 API | 高德地图 API \(amap.com\)](https://lbs.amap.com/)，程序中使用的全 API 如下（以下参数说明仅标明程序中用到的参数，其余参数详见链接文档）：

(1) 地理编码（[地理/逆地理编码-API 文档-开发指南-Web 服务 API | 高德地图 API \(amap.com\)](https://lbs.amap.com/api/web-service-api/geocoding/)）

请求参数说明：

参数名	含义	规则说明	是否必须	缺省值
key	高德 Key	用户在高德地图官网申请 Web 服务 API 类型 Key	必填	无
address	结构化地址信息	规则遵循：国家、省份、城市、区县、城镇、乡村、街道、门牌号码、屋邨、大厦	必填	无
city	指定查询的城市	可选输入内容包括：指定城市的中文（如北京）、指定城市的中文全拼，不支持县级市。	可选	无，会进行全国范围内搜索
output	返回数据格式类型	可选输入内容包括：JSON，XML。	可选	JSON

返回参数说明：

名称	含义	规则说明
status	返回结果状态值	返回值为 0 或 1，0 表示请求失败；1 表示请求成功。
geocodes	地理编码信息列表	结果对象列表，包括下述字段：
location	坐标点	经度，纬度

(2) 路径规划（[路径规划-API 文档-开发指南-Web 服务 API | 高德地图 API \(amap.com\)](https://lbs.amap.com/api/web-service-api/path-planning/)）

1) 步行路径规划

请求参数说明：

参数名	含义	规则说明	是否必须	缺省值
key	请求服务权限标识	用户在高德地图官网申请 Web 服务 API 类型 KEY	必填	无
origin	出发点	规则：lon, lat（经度，纬度），“,” 分割，如 117.500244, 40.417801 经纬度小数点不超过 6 位	必填	无
destination	目的地	规则：lon, lat（经度，纬度），“,” 分割，如	必填	无

		117.500244, 40.417801 经纬度小数点不超过 6 位		
output	返回数据格式类型	可选输入内容包括：JSON，XML。	可选	JSON

返回参数说明：

status	返回结果状态值	返回值为 0 或 1，0 表示请求失败；1 表示请求成功。
route	路线信息列表	
origin	起点坐标	
destination	终点坐标	
paths	步行方案	
distance	起点和终点的步行距离	单位：米
duration	步行时间预计	单位：秒
steps	返回步行结果列表	
step	每段步行方案	
step	每段步行方案	

2) 驾车路径规划（与步行路径规划类似，详见链接文档）

3) 骑行路径规划（与步行路径规划类似，详见链接文档）

(3) 静态地图（[静态地图-API 文档-开发指南-Web 服务 API | 高德地图 API \(amap.com\)](#)）由于篇幅限制，详见链接文档

2. 函数与类（具体函数与类的功能和参数详见代码注释）

main.py:

closeAndShowRouteWindow(w_1, w_2):实现页面跳联，即关闭原页面，打开新页面

utils.py:

class Map_API:调用五个高德地图 API

get_walkingroute(self, origin, destination):步行路径规划 API

get_ridingroute(self, origin, destination):骑行路径规划 API

get_drivingroute(self, origin, destination):驾车路径规划 API

get_geocode(self, address, city=None):地理编码 API

get_static_map(self, location, zoom, size):静态地图 API

class RouteGUI(QMainWindow):路径规划 GUI 界面

def setup_ui_style(self):美化 GUI 界面

def display_message(self, message):在 textEdit widget 中打印一行信息

def get_walking_route(self):调用 get_geocode 和 get_walkingroute 函数，获得步行路径信息

def get_riding_route(self):与 get_walking_route 同理

def get_driving_route(self):与 get_walking_route 同理

class MenuGUI(QMainWindow):菜单 GUI 界面

def setup_ui_style(self):美化 GUI 界面

```
class SMapGUI(QMainWindow):菜单 GUI 界面
    def setup_ui_style(self):美化 GUI 界面
    def get_image(self):调用 get_static_map, 获得静态地图
```

3. 实验中的难点与解决方案

(1) 问题: 高德地图 GUI 输入输出参数多, 难以规划结果呈现

解决方案: 使用 step、instruction 等描述型参数, 描述路径的规划。定义用于打印的函数, 一行行打印路径描述。

(2) 问题: 静态地图 API 调用时一直出现 bug

解决方案: 静态地图 API 参数过多, 因此在代码中考虑到每一种错误情况, 并在命令行打印出来。采用不同参数调试, 最后发现是 local 参数出现问题。

(3) 问题: 路径规划的 API 中, 定位出现偏差, 比如中科大定位到北京

解决方案: 查阅地理编码的请求参数, 发现加上城市之后得到的定位结果更加准确, 因此在路径规划的功能中加上了城市的输入框。

(4) 问题: 一开始使用的 gui 界面是 tthinker (图 1), 感觉太丑了

解决方案: 查找资料发现 tthinker 很难美化, 于是决定使用 PyQt5。使用了 QtDesigner 之后, 发现还是只能进行比较简洁的布局和字体设置, 没有找到带有颜色设置。因此在查阅资料后, 在 utils.py 文件中 RouteGUI、SMapGUI、RouteGUI 这三个关于 GUI 的类里面新增 setup_ui_style 函数, 设置 label、layout 等组件的颜色与格式、文字的字体和背景页面。当然, QtDesigner 使用过程中也碰到了一些问题, 比如图 2, layout 中无法和整个背景一起渐变。看起来效果不佳。最后, 经过调整, 达到了想要的效果 (见: 四、页面布局)

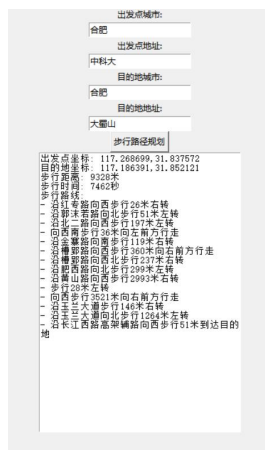


图 1

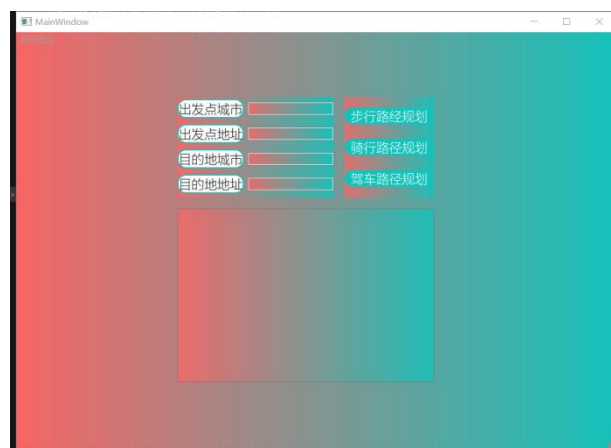


图 2

(5) 问题: 实现页面之间点击按钮的跳转

解决方案: 在查阅资料后, 找到了一些例子, 可以将 pushbutton 组件的点击和页面打开连接起来, 但是无法关闭原页面。因此补充了 closeAndShowRouteWindow 函数, 以实现在打开新页面的同时关闭原页面。

三、实验总结

1. 结果分析

基本实现了想要的结果。路径规划功能中，在输入地址后点击骑行/步行/驾车路径规划，会出现目的地和出发点的距离分析以及详细路径描述。但是如果输入地址不够详细，仍会出现定位错误导致路径错误的情况，这是由于 API 的限制。因此在输入时需要详细描述地点。静态地图功能中，也会有同样的问题，但是为了页面的简洁，没有添加城市的输入框，因此在输入详细地址时最好加上城市。静态地图的地图页面效果良好。

2. 心得体会

这是我第一次尝试写前端页面，在入门的过程中遇到了很多难点，也获得了很多收获：前端和后端总体要进行分离，否则在修改时会非常麻烦，互相影响；代码的框架最好在一开始就有一定的规划，在写的时候沿着框架写，这样就不会出现一个类里有许多函数，或者有太多类的情况；代码的可读性非常重要，在代码长了之后，良好的可读性使其便于理解；为提高代码的鲁棒性，可以进行防御性编程，减少程序出现莫名其妙的错误的可能。

不过，本项目还有一些改进空间，比如为路径规划的结果做出多样化的展示，尽量贴近高德地图 app 的展示效果；高德地图 app 中还有很多 API 功能待开发；GUI 界面还可以进一步美化，使其更清晰且赏心悦目……期待之后对这些地方进一步改进。

四、界面布局



图 3：菜单

图 4：步行路径规划

图 5：骑行路径规划

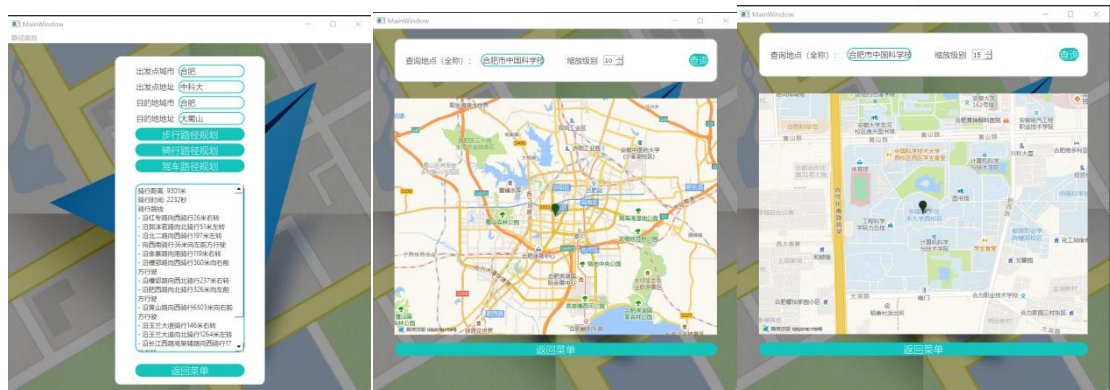


图 6：驾车路径规划

图 7：静态地图（缩放级别 10）

图 8：静态地图（缩放级别 15）