47 组程设作业报告

第一部分 功能简介

主要实现了天气信息的检索和获取功能，本项目是基于 Qt6 开发的天气信息系统，可基本实现天气信息的获取，检索，显示与一定的个性化设置，具有较高实用性.

本项目通过调用 API，实现天气信息的获取和检索. 天气信息包括当下天气和未来七天的天气状况，其地理范围涉及国内外. 天气信息均精确至县,区(国外地点可能有例外).

本项目允许用户通过输入地点汉字名称，汉语拼音，英文名称来检索天气信息，并允许用户实时刷新. 用户可以查看当下的天气情况，也可以查看未来几天的天气预报.

本项目有丰富多样的背景，用户可凭借喜好更换界面背景.

第二部分 程序与组件说明

ui 部分
一、主窗口：mywidget
Ui：

MainCityBtn（“当前位置”）：显示当前搜索的城市
RefreshBtn（“刷新”）：刷新按钮，刷新实时天气
CityBtn（“切换城市”）：选择新的城市
changeBack（“切换背景”）：一共四个背景图片，在此之间切换
ShowTimeButton：显示当前时间，精确到秒，通过应用<QTimer>头文件，调用 showtime()函数实现。
ShowTemperatureButton：以大字号显示当前温度

（minimaxT）：显示最高温与最低温
ShowTemperatureButton 右侧有一个 png 图片，是由 svg 库转换而来，与当前天气状况相对应。
主页面力求简洁，仅仅显示最基本的信息：位置、温度、天气状况，同时这些信息主次分明，温度和天气作为最主要呈现的信息以最显眼的白色加粗字体显示。
通过重写 paintEvent()函数实现北京的设置。
RefreshBtn（“刷新”）和 CityBtn（“切换城市”）均设置为圆角半透明。
并且，MainCityBtn（“当前位置”）、RefreshBtn（“刷新”）、CityBtn（“切换城市”）均通过 installEventFilter 的方式，实现鼠标按下和取消时变化不同的灰度和字体大小。
点击 CityBtn（“切换城市”）按钮后，弹出“search_box”的上半部分。
左上角通过 NowRegion（QLabel）显示地址。

一．search_box

search_box 是一个自定义的 widget，其上半部分包含一个 QLineEdit 和一个 QPushButton，QLineEdit 提示在此处“输入你关心的城市”。

输入完成后，点击“搜索”按键，弹出 search_box 的下半部分。下半部分是一个 QListView，里面包含相关的联想词，选择自己想了解的地区后点击，search_box 即消失，再点击刷新按钮即可刷新为新地址当前的天气状况。

二．FollowMouseWidget

实现了自定义的 widget：FollowMouseWidget，通过 installEventFilter 的方式实现了鼠标跟随，即在不按鼠标的情况下让其时刻跟随鼠标移动。通过槽函数实现在 0.5 秒内按下两次空格键或双击，即可使 FollowMouseWidget 隐藏。通过 setWindowOpacity()函数设置其透明度为 0.8。

在 mywidget 下实例化两个 FollowMouseWidget：

第一个是 MyFollow1，在点击 ShowTimeButton 时显示，上面显示近 7 天的详细信息，包括体感温度、UV、风力等级、风向、大气压等。

第二个是 MyFollow2，在点击 ShowTemperatureButton 时显示，上面显示当天的详细信息，包括体感温度、UV、风力等级、风向、大气压。

api 相关说明：主要实现了以下几个类和函数

一、 文件 decompressGzip.h decompressGzip.cpp

实现了 QByteArray decompressGzip(const QByteArray &data)函数：利用 zlib.h 库中的相关

类，实现对 api 获得的 gzip 压缩后的 json 文件的解压缩，返回解压缩后的 json 文件（若失败返回空数组），以方便后续进一步的处理。

二、文件 cityinfo.h cityinfo.cpp

a) 实现类 cityinfo

i. 成员变量

1. QString country;
2. QString adm1; //一级行政区划，如“江西省” “北京市”
3. QString adm2; //二级行政区划，如“上饶” “抚州”（地级市） “北京”
4. QString lat; //纬度
5. QString lon; //经度
6. QString name; //地名，如“鄱阳”(县） “海淀”
7. QString id;//城市 id

ii. 构造函数 cityInfo(const QJsonObject& cityObject)//根据 json 文件实现 cityinfo 的更新

iii. 成员函数 print()//用于调试

b) 声明全局变量，储存搜索所得的城市信息

i. int numCity; //备选城市的数量,最大取 10 可能为 0

ii. cityInfo cityToSelect[12];//cityToSelect[0]为第一个备选城市

iii. QString nowID;//目前显示的城市的 id

iv. QNetworkAccessManager manager;//用于发送网络请求的对象

v. QString nowCity[4];//当前显示的城市的行政区划等信息

c) 实现与天气信息更新相关的函数

i. void updateCity(const QJsonDocument& data); 基于 json 文件更新 cityToSelect[]，在 sendCityRequest 函数中被调用

ii. void sendCityRequest(QNetworkAccessManager *manager,const QString& name//利用输入的城市名发送城市请求，此时更新 cityToSelect[]);

三、文件 weathereachday.h weathereachday.cpp

a) 实现类 weatherEachDay

i. 成员变量

1. QString date;
2. QString weatherDay;//白天天气
3. QString weatherNight;//晚上天气
4. QString tempMin;//最低温度
5. QString tempMax;//最高温度
6. QString windDirDay;//白天风向，“东南”
7. QString windScaleDay;//白天风力等级
8. QString windSpeedDay;//白天风速，公里/小时
9. QString windDirNight;//晚上风向
10. QString windScaleNight;//晚上风力等级
11. QString windSpeedNight;//晚上风速
12. QString sunRise;//日出时间
13. QString sunSet;//日落时间
14. int vis;//能见度，公里
15. int cloud;//云量，百分比数值，可能为空

16. int uvIndex;//紫外线强度指数

ii. weatherEachDay(const QJsonObject& dayObject)// 根据 json 文件实现 weatherEachDay 的更新

iii. 成员函数 print()//用于调试

b) 声明全局变量，储存搜索所得到的七天天气信息

i. QString updateTime//七天预报信息更新的时间

ii. weatherEachDay weatherForSevenDay[8];;//weatherForSevenDay[0]为第一个备选城市

c) 实现与天气信息更新相关的函数

i. void updateWeather(const QJsonDocument& data); 基于 json 文件更新 weatherForSevenDay[],在 sendWeatherRequest 函数中被调用

ii. sendWeatherRequest(QNetworkAccessManager *manager,const QString& locId="101010100") 利用位置的 id 发送天气请求，此时更新 weatherForSevenDay[]

四、文件 weatherrealtime.h weatherrealtime.cpp

a) 实现类 weatherRealTime

i. 成员变量

1. QString obsTime;//数据观测事件
2. QString temp;//温度，单位为摄氏度
3. QString feelsLike;//体感温度，单位为摄氏度
4. QString text;//天气的文字描述，如晴、多云
5. QString windDir;//风向，如东南风
6. QString wind360;//风向 360°角，如 123
7. QString windScale;//风力等级
8. QString windSpeed;//风速，单位为公里/小时
9. QString humudity;//相对湿度，百分比数值，如 72
10. QString precip;//当前小时累计降水量，单位为毫米，
11. QString pressure;//大气压强，单位为百帕
12. QString vis;//能见度，单位为公里
13. QString icon;//天气状态的图标代码

ii. 构造函数 weatherRealTime(const QJsonObject& weatherObject);//根据 json 文件实现 weatherRealTime 更新

iii. 成员函数 print()//用于调试

b) 声明全局变量，储存搜索所得的城市信息

i. weatherRealTime weatherRealTimeObj;//当前显示城市的实时天气信息

c) 实现与天气信息更新相关的函数

i. void updateWeatherRealTime(const QJsonDocument& data);基于 json 文件更新 weatherRealTimeObj，在 sendWeatherRealTimeRequest 函数中被调用

ii. void sendWeatherRealTimeRequest(QNetworkAccessManager *manager,const QString& locId="101010100");//利用输入的城市 id 发送实时天气请求，此时更新 weatherRealTimeObj

第三部分 功劳簿

余卓林主要负责 api 相关的程序，包括天气信息的抓取和检索功能的实现。

龚正宁主要负责 ui 设计，实现了搜索框和信息窗口的设计。

杨知非主要负责信息显示和代码整合，具体实现了天气信息的展示和搜索。

三人共同完成了演示和报告。