

Lab1_Python & API & GUI

2019.3.30 by 张永停 PB17111585

实验要求

1. 使用与网络有关的API 或者自己使用工具从网站上获取数据。
2. 设计至少一个类，并且至少使用两个文件。
3. 代码的逻辑要比较清晰，同时具有比较高的可读性和精简性，要求有一定的注释。
4. 使用GUI界面或者网页将所完成的功能展示出来。

实验设计

API接口

通过聚合api获得信息

weather类

用于获得天气相关的信息

该类的私有变量是appkey。

```
RequestCityLoc(self,cityname,m="GET")
```

通过输入城市名字查询天气信息，返回数据。

由于某种神奇的原因，当输入为中文时，`parse.urlencode`并不能正确的得到结果，我花了很长时间搜索，但没有找到原因（它单独输出的字符串是对的，但就是请求失败）如果助教可以告诉我原因，学生感激不尽。

```
RequestGPS(self,lon,lat,m="GET")
```

通过输入的经度纬度查询天气信息，但此接口只能查询国内天气信息，返回数据。

使用urlencode使请求符合标准。

airquality类

用于获取空气质量的相关信息(该类是后来补充加上的)

同weather类，该类的私有变量是appkey。

```
RequestAir(self,cityname,"m=GET")
```

通过城市名来获取天气信息。当输入为gps信息时，将 `RequestGPS` 中的城市信息作为参数。

数据处理

在gui中

城市查询时的数据处理

`push_butn_city()` 处理鼠标点击城市查询

- 首先判断请求是否正常 `if res_city and res_air:`, 如果任何一个请求不正常, 通过 `QMessageBox` 输出 `request_error`。这里需要指定输出字体颜色, 因为之前在通过QSS优化UI时设定了字体为白色。
- 然后判断 `error_code`, 如果为0, 则说明输入数据合法, 进入数据图形化函数, 若否, 则通过 `QWssageBox` 输出错误原因。

位置查询时候的数据处理

`push_butn_gps()`处理鼠标点击GPS查询

- 首先判断请求是否正常与输入是否合法, 同城市查询。
- 本函数较之城市查询函数, 多了一步通过天气查询api得到城市名字再传递给空气质量查询api接口的步骤 (因为空气质量查询的接口不提供GPS查询模式)。

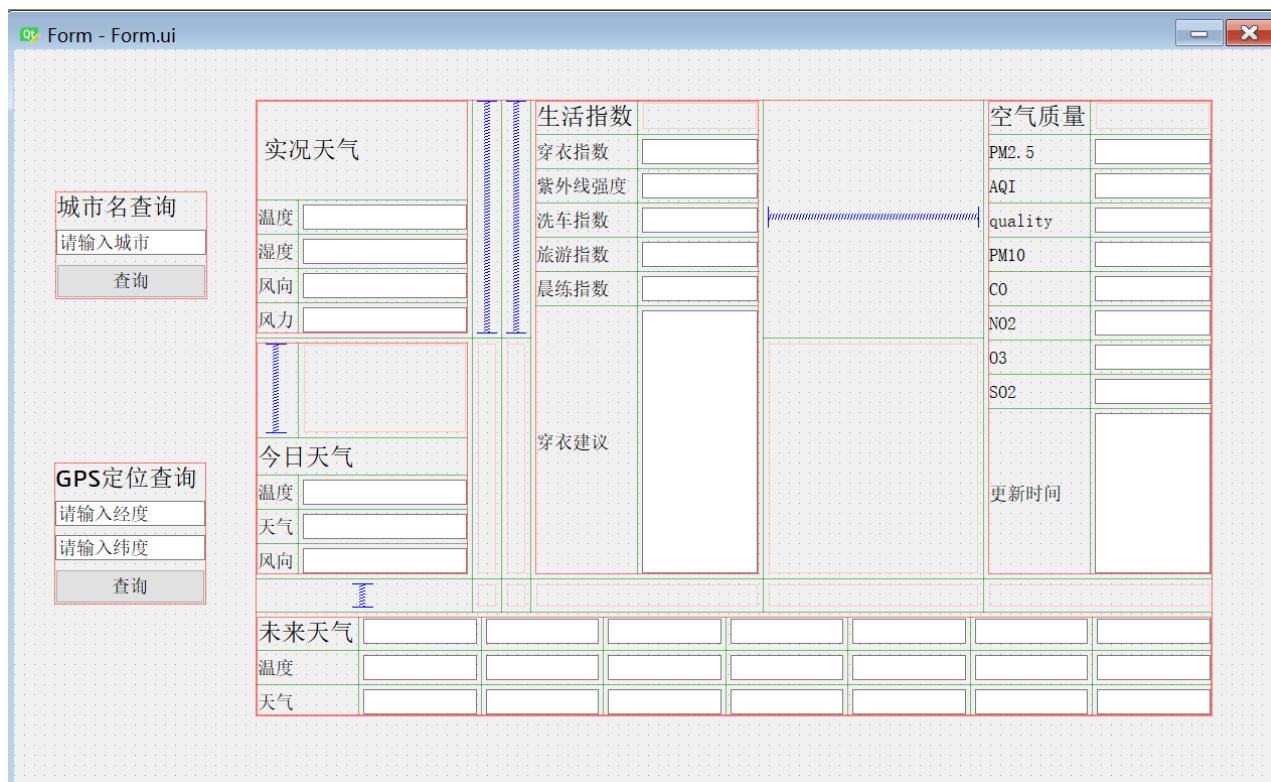
数据显示

- `gui`模块继承自 `Ui_Weather`, 它是通过Qt designer设计的。
- 通过私有变量 `self.__weather = weather()` 与 `__ai_rqu=ai_rqual_i ty()` 来获得数据。
- 如果数据正常获取, 用 `printfnormal()` 向控件 `TextEdit` 输出结果。这里的一个小技巧是, 输出未来天气时, 数据给键是日期, 而由于日期是变化的, 不能提前知道, 因而通过 `list(res["result"]["future"].keys())` 来获得键, 然后用迭代器来访问未来天气。如 `self.date1.setText(res["result"]["future"][it[0]]["date"])`。
- 如果数据获取异常, 用 `printferror()` 向控件输出“N/A”。

图形化

采用PyCharm+Qt designer+PyUic+QSS来设计UI

- 首先使用Qt Designer绘制基本框架, 通过栅格化优化界面



这是通过Qt designer设计的界面

- 通过 `setWindow` 来改变窗口名字、透明度、图标
- 然后通过QSS进一步优化
 - 设计QLineEdit、QPushButton、QLabel、QTextEdit的颜色，将边角设置为圆弧，改变字体。
 - 将特殊的控件如显示“城市名查询”等的Label重新设定，增大字号与宽度。
- 最后通过Qbrush增加背景图



优化后的界面

功能与使用说明

- 该代码通过两个api接口实现了查询天气与空气质量。通过用户输入，程序会告知用户当前天气与空气质量的信息。比如，实况天气，实况空气质量，今日天气，今日各项生活指数(旅游、锻炼、洗车、穿衣建议)以及未来七天的天气。
- 通过城市名查询天气，输入中文城市名即可。如不合法，程序会报错。
- 通过GPS查询天气，输入经纬度，只可以查询国内。查询国外或其它输入不合法程序报错。

总结

跳坑经历

- 未加urlencode,一直无法请求成功。
- 对于请求中有中文需要转成UTF8,最开始未注意到，以及后来用urlencode一样无法成功，无奈之下使用 `Request(("s?ci tyname=%s&key=%s") % (url, parse.quote(ci tyname), self.__apikey))` 来发送请求。
- 没有仔细看api返回参数，在处理空气质量时，自以为是字典类型，浪费了很多时间调试，结果是列表形式。

收获

- 虽然没有用到，但学会了QT的中文处理
- 通过设计，终于理解了信号和槽
- 之前一直不会继承通过本次大作业被强制性使用继承，终于学会了。也学到了 `__name__` 和 `super()` 的用法。
- 复习预习了QT的基本用法
- 由于初版UI太丑，而我又比较闲，就花了很多时间在UI上，**然后学会了QSS**，终于让UI看得过去了。QSS万岁！
(破音)

感受与心路历程

最开始一直觉得难迟迟未下手，最后在ddl前四天终于开始写，~~是第一生产力~~没错了。

写的时候发现，~~我没学过Python~~。果然语言这个东西，还是要真正自己去写一写。通过自己上网查，发现了许多有意思的东西，也差不多算是入了门。

写完发现，**Python太强大了！**其实主控代码很短，但实现的功能非常强大。~~不应该是Qt designer强大~~。由于受不了UI的丑陋，又去学了QSS，其实QSS学的蛮快，它和CSS几乎一样。主要的时间都花在各种调文字、大小、样式上了。幸好我没去学设计

最后看着成品其实蛮有成就感的！

最大的感触就是，有的时候，真的是自己吓自己，真的下手了，才会发现其实不难。