2019소프트웨어 project 보고서

작성 : 20171703정태원, 20191647이종엽

1. 프로그램의 구조

프로그램명 : 서울의 역 지명과 지하철 길 찾기

일시 : 2019년 12월 14일

장소 : 국민대학교

자원 : Pycharm, 공공데이터포털, 서울 열린데이터 광장(open api), Beatifulsoup

투입 인력 : 정태원, 이종엽

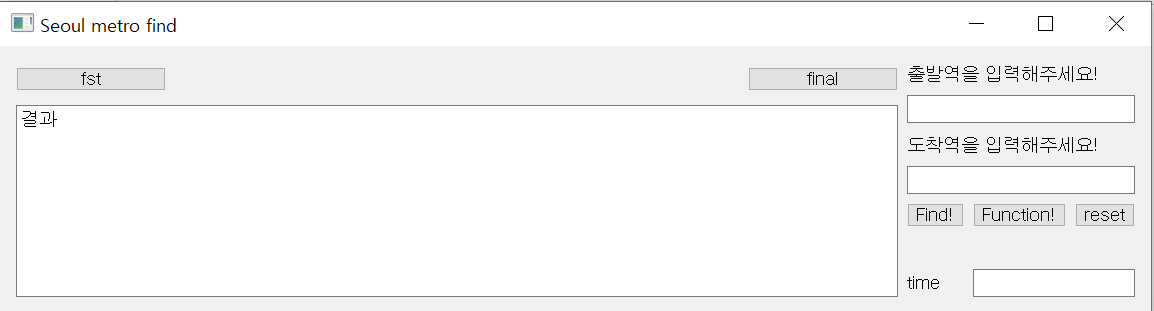
1. 프로그램의 목적과 목표

목적 : Open api를 활용해 봄으로서 데이터의 활용도에 인지해보며 pYQT5를 활용한 GUI인터페이스의 프로그램을 개발을 해보자

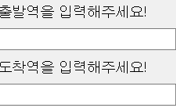
목표 : Open Api 활용 방법에 대해서 알아보기, Gui 인터페이스를 원하는 대로 작성하여

원하는 버튼이나 텍스트 세팅하기

1. 예제 프로젝트 구현 결과물



 : 출발역이 들어갈 자리  : 도착역이 들어갈 자리

각각 역을 입력  : find를 누르면 open api 에서 각각의 정보를 가져옴 ex) 환승역, 역의 호선, 역사내용

 : 가장 알고 싶은 총 지하철 이용 시간이 출력된다.

1. 요구사항 분석

Open api의 서비스 키를 받아와야 함

api에서 원하는 정보만 출력하도록 프로그램을 설정

- api안에는 같은 태그를 가진 정보가 무수히 많이 있음

5. 요구 사항 명세서(SRS)

기능적 요구사항 : 정확하게 필요한 api정보를 읽어 들일 것.

객체들이 각각 제기능을 하도록 객체지향적으로 프로그램을 설계할 것

인터페이스 요구사항 : 지하철 호선에 맞는 색으로 선을 출력해낼 것

비기능적 요구사항 : 소프트웨어 구현에 open api, pycharm, pyqt5를 이용할 것

작업을 손쉽게 하기위해서 github를 적극적으로 활용할 것

1. 소프트웨어 구조 설계

|  |  |  |
| --- | --- | --- |
| 모듈 | 클래스 | 역활 |
| adproject | Gettime\_station\_history | 지하철 이용시간  지하철 환승역  지하철의 역사정보  를 가져온다. |
| Ad\_dict | - | 역의 wgs84 좌푯값정보를 딕셔너리 형태로 저장 |
| adtool | SeoulMetro | Gui로 구현  각각의 함수를 버튼을 누르면 호출하도록 하고(call back 함수를 구현)  지하철 역 사이를 제대로 호선에 맞는 색으로 구현 |

1. 클래스 인터페이스 설계

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 클래스 | 메서드 | 입력 | 출력 | 기능 |
| Seoul  Metro | setup\_ui |  |  | 말그대로 버튼의 위치  Text와 버튼을 connect |
| findEvent | 출발역  도착역 | 역사내용  걸리는 시간  환승역 | FindClicked가 눌렸을 때 발생하는 이벤트 |
| drawEvent | 환승역의  갯수 | QLINE의  좌표 | 역에서 역 사이 구간의 좌표를 Qline으로 받음 |
| paintevent |  |  | FunctionEvent가 눌렸을 때 Qline에 호선에 맞는 선을 그림 |
| draw\_middleLine | Qline의  실존 여부 | Qline에  맞는 선  그리기 | paintEvent에서 활용 |
| resetEvent |  |  | 모든 값을 null값으로 변경 |

1. 소프트웨어 구조 설계서

Gettime\_station\_history

(Open Api, BeautifulSoup)

Seoul\_Metro

(Pycharm)

Ad\_dict

(wsg84 좌표계)

1. 구현 상세 설계서

Class Gettime\_station\_history

Attribute

depart : 출발역, arrive : 도착역, shortindex : 열차를 탑승하는 경우의 시간을 담은 리스트

bs\_obj2 : beautifulSoup을 통해 url를 통해 받아온 값을 파싱하여 담은 값

history : 도착역의 역사를 담은 정보,

wantst\_line : 받아온 모든 역의 호선들

url\_3 = **"http://ws.bus.go.kr/api/rest/pathinfo/getPathInfoBySubway?ServiceKey=%2BurVX0NS%2FR0gnSJAPplvPe4AKfCzVhZG9UY6LUnSFTQwKmHcbtAsetokdvddXzVVQJ9ThRvwBzGXifojUxm9Ag%3D%3D&startX="** + str(  
 st[1]) + **"&startY="** + str(st[0]) + **"&endX="** + str(ar[1]) + **"&endY="** + str(ar[0])

url\_2 = **"http://openapi.seoul.go.kr:8088/"** + service\_key + **"/xml/StationNmfpcOrgnThemaNm/1/5/"** + self.top\_list4[-1]+ **"/"** + self.arrive

1번째는 ad\_dict에서 넣어 논 wsg84좌표계를 받아와 경로, 시간, 호선을 받아오는 api 서비스

2번째는 1호선에서 4호선까지 도착역에 대해 역사를 가져와주는 api

method

Gethistory : bs\_obj2에서 파싱한 값들 중에 역사에 관한 내용만 출력

GetTransit\_station : 똑같은 값을 enumerate함수를 응용하여 파싱했을 때 나오는 모든 경우 중에 시간이 적게 걸리는 경우의 역들만 list slice를 통해 추려낸다.<태그>를 했을 때 나오는 값이 매우 다양함으로 인덱스나 슬라이스를 잘 활용해야 함

Gettime : 가장 적게 걸리는 시간을 파싱한 값들을 리스트에 담은 후 비교하여 출력

GetStation\_line : Transit\_station과 원리는 비슷하고 원하는 호선들만 가져와서 시간이 적게 걸리는 경우의 호선들만 리턴한다.

Class SeoulMetro

GUI를 구성하고 Gettime\_station\_history로부터 받아온 값을 이용해 GUI에 표시한다.

Attribute

Dict : 1호선부터 분당선까지의 호선 별 색상정보를 담아두었다.

Line1 ~ 7 : 역과 역 사이를 선 긋기 위해 만든 빈 라인

Method

findEvent : 입력한 출발지와 도착지를 이용해 Gettime\_station\_line 함수에 넣어 걸리는 시간, 환승역, 환승 노선 등을 저장 및 GUI에 표현한다.

drawEvent : findEvent에서 저장한 값을 기반으로 line에 역과 역 사이의 거리를 저장한다.

paintEvent : GUI에 그림을 표현할 준비를 하는 method로 매 순간 불러오기를 반복한다.

draw\_middleLine : paintEvent에 의해 지속적으로 호출되며 line에 역과 역 사이의 좌표 값이 들어올 경우 노선에 맞는 색을 이용해 역 사이를 선 긋는다.

resetEvent : findEvent, drawEvent 등에 의해 저장된 값과 변경된 GUI들을 초기값으로 되돌린다.

|  |  |
| --- | --- |
| MainLayout | |
| ShowLayout 그래픽을 나타내기위한 레이아웃 | StatusLayout 역 정보를 입력하고 출력하는 레이아웃 |
| MetroLayout 역 구간을 표시하는 레이아웃 | strStart : 출발지를 입력하는 곳  strEnd : 도착지를 입력하는 곳  findBtn : 출발지와 도착지를 기반으로 환승역 및 소요 시간을 알려준다.  drawBtn : 역 사이를 역을 상징하는 색으로 선을 그린다.  resetBtn : 입력했던 정보를 초기화한다.  strTime : findBtn 클릭 시 소요 시간을 표시함 |
| fstBtn, finalBtn : findBtn 클릭시 출발역과 도착역이 다음 버튼에 기입됨  spaceNull : 역과 역 사이 빈공간을 표시하기 위해 임의로 삽입 및 환승역을 표시하기 위해 숨겨둔 버튼의 역할을 한다.  recentStatus : findBtn 클릭 시 도착역의 지명 유래를 알려주는 역할을 한다. |

10. 실행 결과

