프로젝트 기술서

Git Address : https://github.com/merignet

2021.07 ~ 2021.08　　빅데이터 분석가 교육과정 파이널 팀 프로젝트

▶ 프로젝트 명 : 군집화를 통한 서울 도심권 요식업 상권분석

▶ 인　　원 : 5명

▶ 사 용 언 어 : SQL, Python3.6

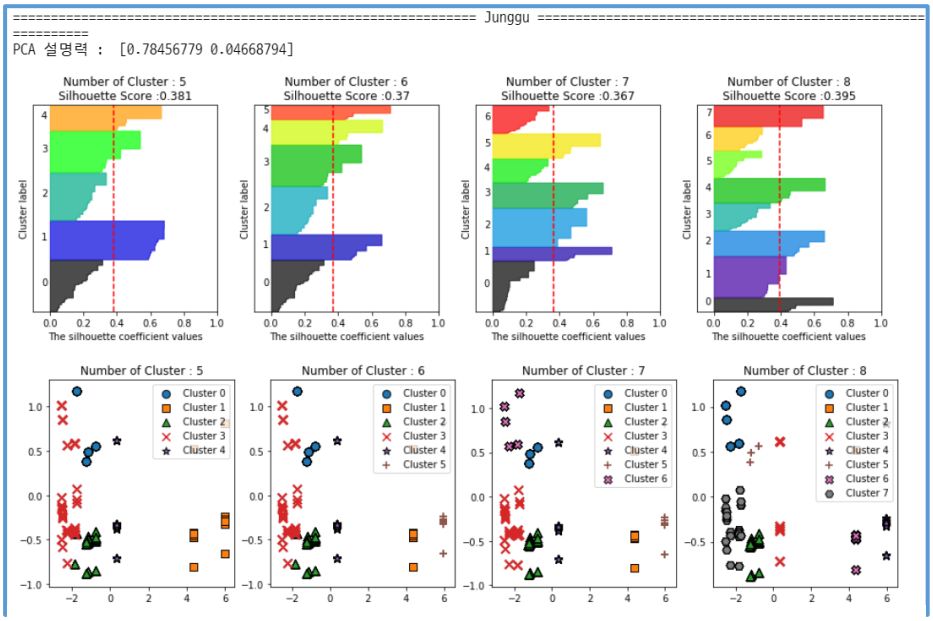
▶ 개 발 환 경 : Jupyter Notebook. Filezilla, QGis, Toad for MySQL, MS Office Excel

▶ 프로젝트소개: 코로나 팬데믹으로 인한 요식업 상권이 붕괴되고 있다. 특히 인구 밀집 지역인 수도권 지역의 코로나 확산세 및 고강도 사회적 거리두기의 지속으로 역세권 상권마저 힘을 잃었다. 이에 본 프로젝트는 경제 침체 속에서 지역별 상권분석을 군집화를 통해 실시한다. 이를 통해 구, 동 별 상권의 분포 등을 분석한다. 지역 범위는 서울시 25개 구 중 (폐업률-개업률)이 높은 도심권 3개 구(용산구, 종로구, 중구)로 설정한다.

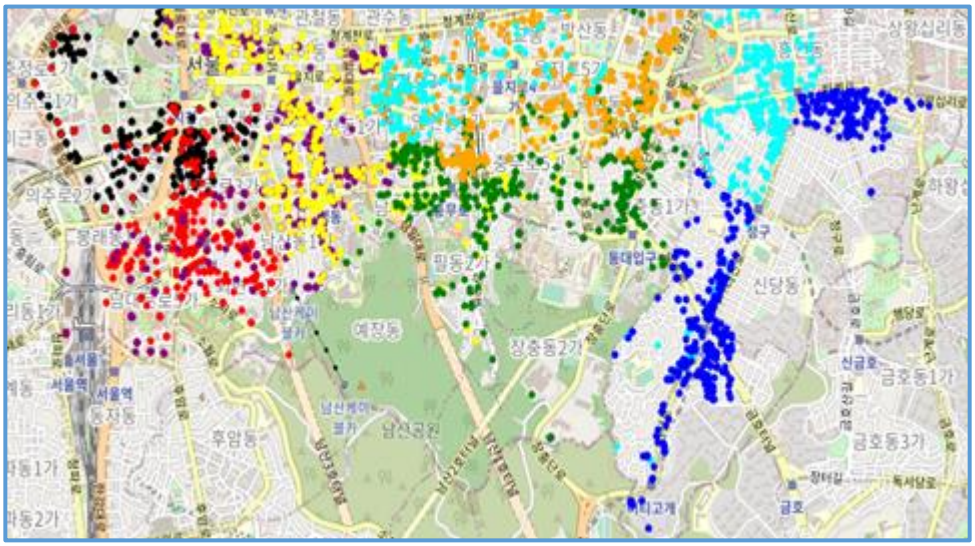
▶ 수집 데이터 : 서울시 총 인구수, 지하철 유동인구수, 상권 업종 및 상권 데이터, 임대료 데이터, 서울시 카드소비패턴 데이터(신한카드), 상권 업종코드 데이터, 서울시 블록코드 데이터

☞ 기준년도 : 2021년

▶ 프로젝트 진행 화면



[그림 Ⅰ-1] 실루엣 계수 별 및 군집화 결과 확인(중구)



[그림 I-2] 음식점 위치 군집별 지도 시각화(중구)

cluster	rent_21_total	total_pop	Dong_code_140520.0	Dong_code_140550.0	Dong_code_140590.0	com_sort_code_1	com_sort_code_6	com_sort_code_8
0	7.797000	-1.714156	0.000000	0.0	0.0	0.536585	0.146341	0.097561
1	1.080691	-2.281456	0.000000	1.0	0.0	0.431507	0.237443	0.083333
2	6.133835	-2.069881	0.000000	0.0	0.0	0.431776	0.213084	0.095327
3	-0.461801	-1.192538	0.000000	0.0	0.0	0.520619	0.149485	0.134021
4	2.169214	-1.988555	0.000000	0.0	1.0	0.513410	0.166667	0.120690
5	0.648279	-2.389202	0.635253	0.0	0.0	0.000000	0.361257	0.228621
6	-0.138638	-2.160771	0.000000	0.0	0.0	0.434524	0.190476	0.130952
7	0.644186	-2.394248	0.593607	0.0	0.0	1.000000	0.000000	0.000000

[그림 Ⅰ-3] 군집별 변수 영향 수치(중구)

▶ 담 당 업 무 : 프로젝트 기획 및 데이터 수집, 전처리

ⅰ. 프로젝트 기획 : 프로젝트 주제 선정, 가설 설정, 데이터 수집 내용 및 범위 지정, 데이터 전처리 방법론 구현, 분석 방법론 선정

ⅱ. 데이터 수집

① 웹 스크래핑 및 Excel을 활용하여 각 CSV 파일로 저장.

② 서울시 빅데이터 캠퍼스의 데이터 사용.

③ 서울시 블록코드 데이터(.shp)파일을 QGis를 통해 CSV 파일로 변환

ⅲ. 데이터 전처리

① 수집 데이터에서 이상치·결측치 제거 후 필요 컬럼만을 추출

② 추출한 컬럼을 Python mergy 함수를 통해 통합

③ 필요 데이터를 데이터프레임으로 변환

④ 피처 스케일링, 원-핫 인코딩, PCA 차원축소 진행.

ⅳ. 분석 데이터를 통한 인사이트 도출

① 군집의 특성을 파악하여 인사이트를 도출한다.

▶ 담당업무 작성 코드

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.preprocessing import scale
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import PowerTransformer
from sklearn.compose import make_column_transformer
from sklearn.metrics import silhouette_score, silhouette_samples
%matplotlib inline
```

[그림 Ⅰ-4] 데이터 전처리 및 분석을 위한 Python 라이브러리

```
['Dong_code', 'rent_21_total', 'total_pop', 'total_station', 'com_sort_code', 'profit_21']
Gu_code = list(df['시군구코드'])
Dong_code = list(df['행정동코드'])
rent_21_total = list(df['21_전체'])
total_pop = list(df['총인구'])
total_station = list(df['1월승차총승객수']+df['1월하차총승객수']+df['2월승차총승객수']+df['2월하차총승객수']
                     +df['3월승차총승객수'] +df['3월하차총승객수'])
station_name = list(df['역명'])
com_sort_code = list(df['상권업종중분류코드'])
profit_21 = list(df['1월_매출']+df['2월_매출']+df['3월_매출'])
```

[그림 Ⅰ-5] 군집화 전, 필요 변수 리스트화

```
df = pd.DataFrame({'Gu_code':Gu_code,'Dong_code':Dong_code,'rent_21_total':rent_21_total,'total_pop':total_pop,\
                   'total_station':total_station,'com_sort_code':com_sort_code, 'profit_21':profit_21})
df2 = pd.DataFrame({'Gu_code':Gu_code,'Dong_code':Dong_code,'rent_21_total':rent_21_total,'total_pop':total_pop,\
                    'total_station':total_station,'com_sort_code':com_sort_code,'profit_21':profit_21,
                    'dong_name':dong_name,'lat':lat,'lng':lng})
```

[그림 Ⅰ-6] 리스트를 데이터 프레임으로 변환

```
dum_df = pd.get_dummies(df_zero)
```

[그림 Ⅰ-7] 상권업종중분류코드 원-핫 인코딩

```
val = sel_one.values
val_scaled = StandardScaler().fit_transform(val)
val_scaled_df = pd.DataFrame(data = val_scaled, columns = ['Gu_code','Dong_code', 'Rent_21_total', 'Total_pop',/
                                                           'Total_station', 'com_sort_code', 'profit_21'])
```

[그림 Ⅰ-8] 분석을 위한 컬럼값 피처 스케일링

```
pca = PCA(n_components=2)
pca_transformed = pca.fit_transform(X_features)
dataframe = pd.DataFrame(pca_transformed, columns=['PCA1','PCA2'])
```

[그림 Ⅰ-9] 피처 수를 줄이기 위한 PCA 차원축소 코드

2021.05 ~ 2021.07 빅데이터 분석 개발자과정 개인 및 미니 팀 프로젝트(2) 및 제1회 K-Digital Hackaton 예선 참가

▶ 프로젝트 명 : 차원축소와 군집화를 이용한 수소차 충전소 입지선정

▶ 주 최 기 관 : 고용노동부 K-Digital Hackaton(해커톤)

▶ 인 원 : 5명

▶ 사 용 언 어 : Python, HTML, Java Script

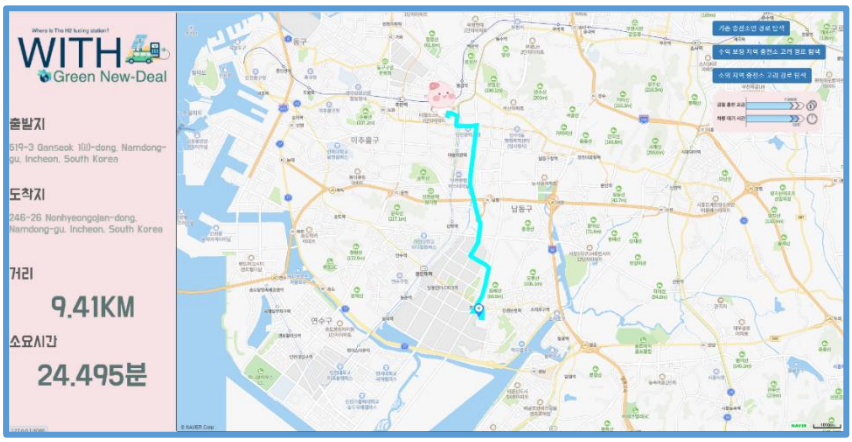
▶ 개 발 환 경 : MS Office Excel, Jupyter Notebook, Visual Code

▶ 프로젝트소개: 코로나로 인한 불안정한 일자리 및 경제 침체가 심화되었다. 이에 정부는 한국판 뉴딜 사업을 통해 경제 대공황 사태를 대비책을 발표했다. 본 프로젝트는 '그린 뉴딜' 중 '탄소 중립 추진'을 위하여 수소 차량 보급을 활성화 할 수 있는 수소가 충전소 입지선정을 주제로 하였다. 또한 사용자 지정 위치에서 가장 가까운 거리의 수소충전소까지의 길 찾기, 거리, 소요 시간 등의 정보를 제공하는 웹 페이지를 개발을 통해 결과를 시각화, 사용자에게 편의성을 제공한다. 지역은 경기 남부로 한정하였다.

▶ 수집 데이터 : 표준 공시지가 데이터, 자동차 주행거리 데이터, 총 인구수 데이터,
수소차 지원대수 데이터, 경기도 수소 생산지로부터의 거리데이터

▶ 담 당 업 무 : 프로젝트 기획, 데이터 수집, 결과 도출을 위한 공식 제작,
웹 개발

- 군집화를 통해 입지 선정된 수소 충전소의 위치를 지도 표시
  (기존 수소 충전소 마커, 수익 보장 지역 입지선정 마커, 소외계층 입지선정 마커)
- 네이버 API를 이용하여 길 찾기, 거리, 소요시간 출력 기능 개발
- 구글 API를 이용하여 사용자 지정 위치 위치와 도착 위치의 주소를 출력
- 사용자 지정 위치의 주소명, 위도, 경도 데이터를 정보창으로 제공
- 사용자 지정 위치와 직선거리 상 가장 가까운 수소충전소 자동 지정
- 길 찾기 루트 폴리 라인(poly-line) 그리기 기능 및 출발 지역 지도 표시(핵심 마커)



[그림 Ⅱ-1] 수소충전소 길 찾기 웹 서비스 구현 화면

▶ 담당업무 작성 코드

```python
from flask import Flask, render_template
from flask.globals import request
import latlon
import route
import startpoint_list
import euclidean
import find_addr

app = Flask(__name__)

@app.route('/', methods=["POST", "GET"])
def index():
```

[그림 Ⅱ-2] Python과 HTML 연결을 위한 Flask 라이브러리 사용

```python
import googlemaps
def find_addr(point):
    #point = (37.65433, 127.65499)
    gmaps = googlemaps.Client(key='AIzaSyCRv0gpIO1D
    g = gmaps.reverse_geocode((point[0],point[1]))
    # print(g[0]['formatted_address'])
    return g[0]['formatted_address']

print(find_addr((37.129833,127.812922)))
```

[그림 Ⅱ-3] Google API 연동 및 위치 정보 주소 출력

```python
import geopy
import pandas as pd
import csv

def load_station():
    station = pd.read_csv('C:/Users/ktb58/OneDrive/바탕 화면/해커톤/Web_p
    # print(station['주소'])
    service = geopy.Nominatim(user_agent = "myGeocoder")
    # Nominatim 서비스 객체에 대한 핸들러 가져 오기
    d_station = {}
    f = open('station_latlong.csv','w', newline='')
    wr = csv.writer(f)
    for stat,loca in zip(list(station['충전소']),list(station['주소'])):
        service = geopy.Nominatim(user_agent = "myGeocoder")
        locationObj = service.geocode(loca)
        s_lat = locationObj.latitude
        s_long = locationObj.longitude
        d_station[stat]=  (s_lat,s_long)
        print(loca)

        wr.writerow([loca,s_lat,s_long])

    f.close()
    return d_station
```

[그림 Ⅱ-4] 위도, 경도 정보를 이용하여 충전소 위치 찾기

```python
import latlon
from haversine import haversine

# print(latlon.list_station())

def euclidian(startpoint, d_station):
    distance = []

    for i in d_station:
        start_lat = startpoint[0]
        start_lng = startpoint[1]
        dist = haversine((start_lat,start_lng),(i[1],i[2]))
        distance.append([dist,i[0],i[1],i[2]])

    return min(distance)
```

[그림 Ⅱ-5] haversine 라이브러리를 통한 최소 직선 거리 찾기

2021.05 ~ 2021.06 빅데이터 분석가 과정 미니 프로젝트(1)

▶ 프로젝트명 : 세이버 스탯을 이용한 한국 프로야구 선수의 연봉 데이터 분석

▶ 인 원 : 5명

▶ 사 용 언 어 : R, Python

▶ 개 발 환 경 : MS Office Excel, Jupyter Notebook, R Studio

▶ 프로젝트소개: 프로야구에 대한 국민의 관심이 뜨겁다. 스포츠 중 야구는 한 번의 경기에서 대량의 데이터가 발생한다. 본 프로젝트는 선수 개인의 역량(독립변수)이 연봉(종속변수)에 영향을 미치는지 회귀분석 방법을 통해 확인한다.

▶ 수집 데이터 : 2018년도 ~ 2020년도 프로야구 선수 연봉 데이터, 선수 개인 역량 데이터

▶ 담 당 업 무 : 프로젝트 기획, 데이터 수집 및 전처리, 분석 시각화를 통한 인사이트 도출

- 프로젝트 가설 설정 및 진행 방안 기획(머신러닝 방법론, 데이터 전처리 방법론)
- KBO 공식 사이트와 STATIZ 사이트에서 제공하는 데이터 수집
- R을 이용하여 데이터의 이상치·결측치 제거, 연관관계 변수 정제, 데이터 범주화

변수		설명
Batter (타자)	WAR	**Wins Above Replacement** 대체선수와 비교했을 때, 얼마나 많은 승리에 기여했는가를 나타내는 수치
	wRC	**Weighted Runs Created** wOBA에 기반을 둔 타격으로 얻어낸 득점기여도
		$(\frac{wOBA - 리그wOBA}{wOBA스케일} + \frac{리그득점}{타석}) * 타석$
	wRAA	**Weighted Runs Above Average** 평균적인 선수와 비교해서 타격으로 얻어낸 득점기여도
		$\frac{wOBA - 리그wOBA}{wOBA스케일} * 타석$
	wOBA	**weighted On Base Average** 출루 이벤트별 실제 득점 가치에 비례한 가중치를 부여한 출루율
Pitcher (투수)	FIP	**Fielding Independent Pitching** ERA의 단점을 보완한 스탯으로, 전적으로 투수에게 책임이 있다고 생각되는 기록들만을 추린 평균자책점의 형태
	LOB%	**Left On Base Percentage** 출루 된 주자 중 득점하지 않은 비율을 나타내는 수치
		$\frac{안타 + 볼넷 + 사구 - 실점}{안타 + 볼넷 + 사구 - (1.4 * 피홈런)}$
	BABIP	**Batting Average on Balls in Play** 인플레이 타구의 안타 비율 혹은 피안타 비율

[그림 Ⅲ-1] 머신러닝을 위해 사용된 변수 리스트

```r
# 이상치를 결측치로 변환
Data_Pitcher <-  ifelse(Data_Pitcher$SALARY(2018) == 900000, NA, Data_Pitcher$SALARY(2018))
Data_Pitcher <-  ifelse(Data_Pitcher$SALARY(2019) == 900000, NA, Data_Pitcher$SALARY(2019))
Data_Pitcher <-  ifelse(Data_Pitcher$SALARY(2020) == 900000, NA, Data_Pitcher$SALARY(2020))

# LOB 변수 생성
Pitcher_record$LOB <- round((Pitcher_record$H + Pitcher_record$BB + Pitcher_record$HBP - Pitcher_record$R) / (Pitcher_record$H +
Pitcher_record$HBP - (1.4 * Pitcher_record$HR)), 2)

# 데이터 확인
head(Pitcher_record)

# 필요 데이터 추출
Data_Pitcher <- Pitcher_record %>%
  filter(!is.na(WAR)) %>%
  filter(!is.na(FIP)) %>%
  filter(!is.na(LOB)) %>%
  filter(!is.na(BABIP)) %>%
  filter(!is.na(SALARY(2018))) %>%
  filter(!is.na(SALARY(2019))) %>%
  filter(!is.na(SALARY(2020))) %>%
  group_by(NAME) %>%
```

[그림 Ⅲ-2] 데이터 내 이상치·결측치 제거 코드

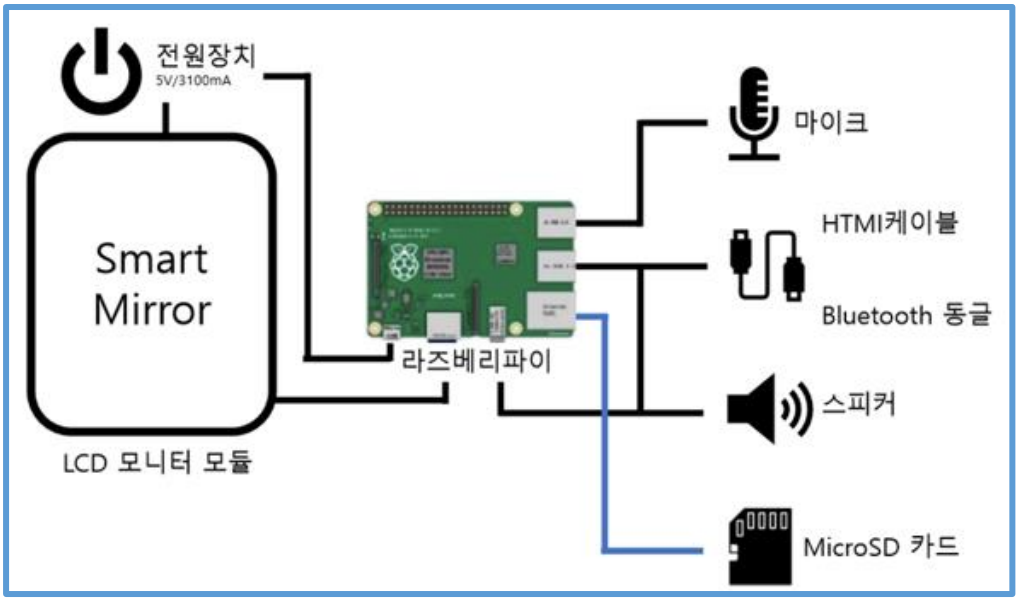
2019.03 ~ 2019.06　　성공회대학교 정보통신공학과 졸업 캡스톤 프로젝트(2)

▶ 프로젝트명 : 미세먼지 정보를 제공하는 스마트 AI 미러

▶ 인　　원 : 3명

▶ 사용언어 및 환경, H/W : Shell Script, Ubuntu Linux, Raspberry Pi, Google API, Java Script

▶ 프로젝트소개: 미세먼지에 대한 국가적 관심이 뜨겁다. 본 프로젝트는 이를 상업적 용도로 활용할 수 있는 아이템을 개발을 목표하였다. 이를 위해 당시 유행하던 AI 스피커를 활용하였고 생활 필수품인 거울을 이용해 프로젝트를 진행했다.



[그림 Ⅳ-1] 제품 구현을 위한 회로도

▶ 담 당 업 무 : AirKorea API 및 Google API(Assistant) 담당

- AirKorea API를 사용하여 대한민국의 정확한 미세먼지 정보를 제공한다,
- Google API 기능 중 Google Assistant를 활용하여 AI 기능을 활용한다.

2018.09 ~ 2018.12 성공회대학교 정보통신공학과 졸업 캡스톤 프로젝트(1)

▶ 프로젝트명 : VR 안전교육 완전 정복

▶ 인 원 : 3명

▶ 게 임 엔 진 : Unreal Engine

▶ 구 현 환 경 : BluePrint, Adobe Photoshop, 3D Max

▶ 프로젝트소개: 어린이 교통 사고는 해마다 증가하고 있다. 이는 기존의 실시하는 어린이 교통안전 교육에 문제가 있다고 판단할 수 있다. 이에 본 프로젝트는 VR 게임을 이용하여 실제 상황과 같은 화면을 통해 어린이가 즐겁게 교통 교육을 받을 수 있도록 한다.

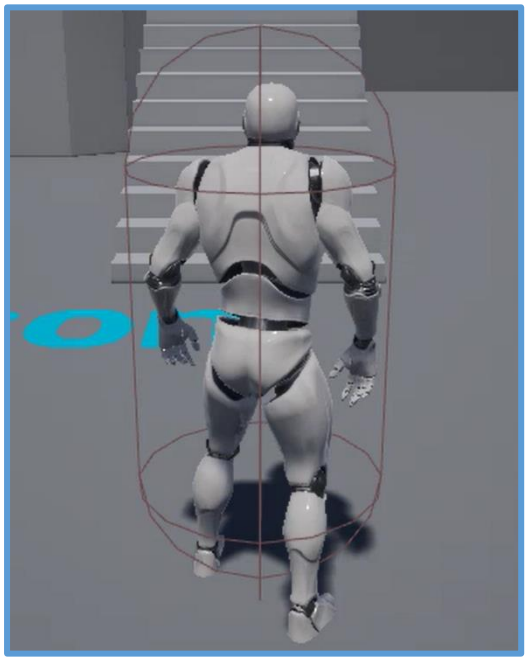
▶ 대 상 연 령 : 5~7세

▶ 담 당 업 무

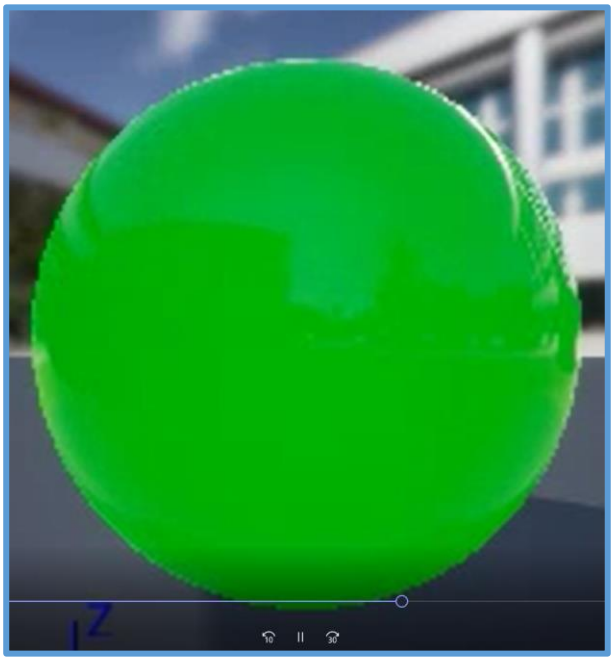
- BluePrint를 이용한 Unreal Engine 구현

→ 사람의 걷는 속도 조절, 신호등 불빛 조정, 움직이는 각도조정, 전체 루트 조정

- Adobe Photoshop을 이용하여 스토리 라인 구성, 컬러 리스트 정리

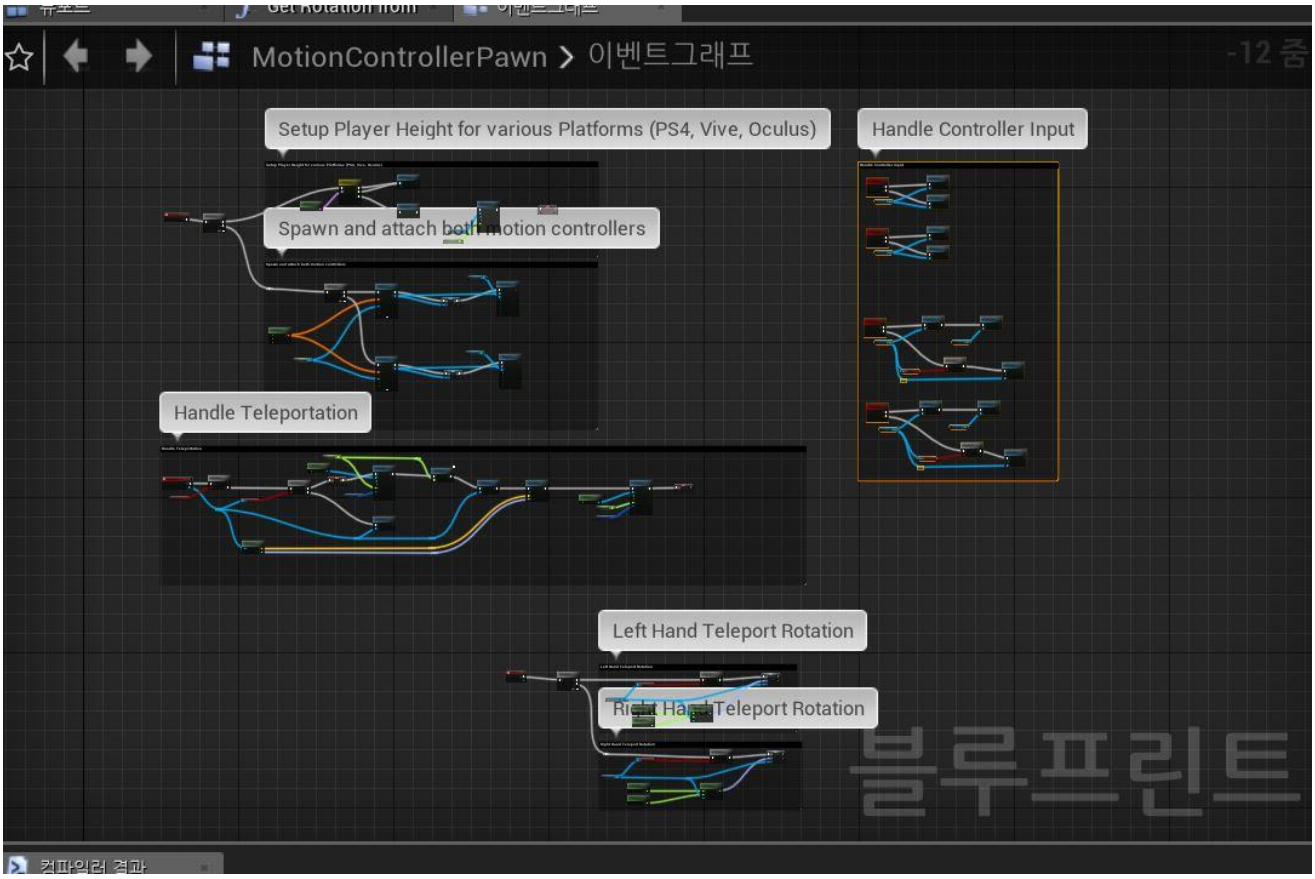


[그림 V-1] 인물 움직임 속도 조절

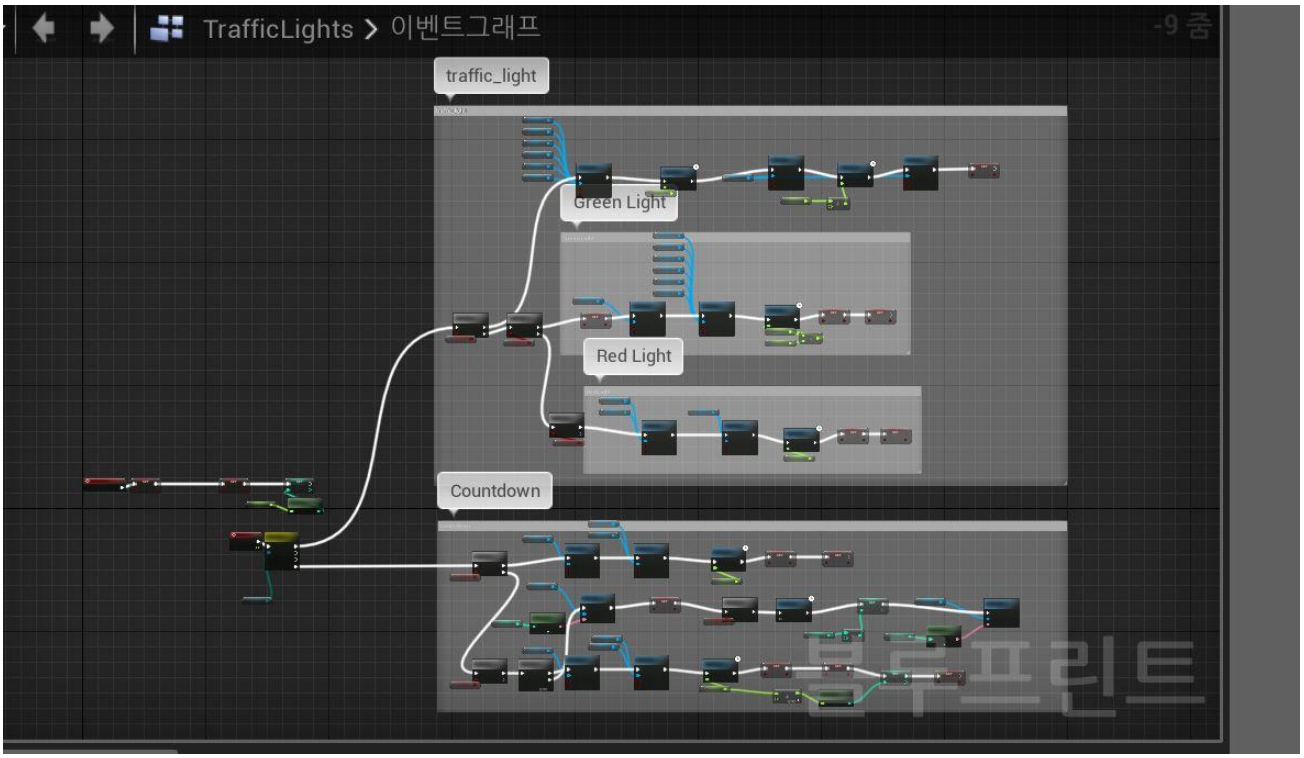


[그림 V-2] 신호등 불빛 조절

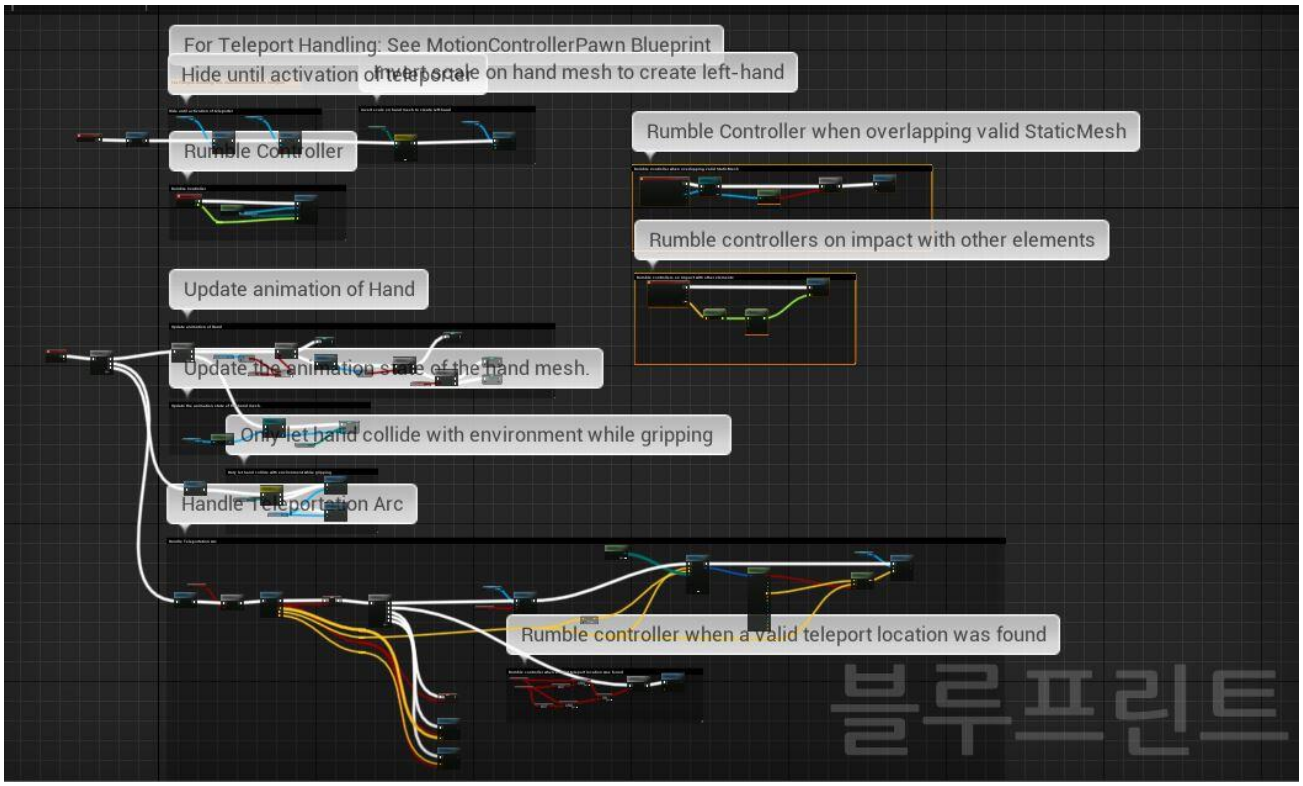
[그림 V-3] 3D 이미지 매핑 화면



[그림 V-4] 순간이동 카메라 초점 조절



[그림 V-5] 신호등 초록불, 빨간불 조정



[그림 V-6] 순간이동 핸들 조절(기어 핸들 조절)