2. Pokemon Go 프로그램 설명

**4조 김상규, 손민우, 최희영**

**실행 과정 및 코드 설명**

1. DB import

- 주어진 dataset에서 os mudule에 있는 os.system 명령어를 이용하여 python 상에서 pokemon.json 파일을 import함. 이 때, database는 ds2, collection 이름은 pokedex을 사용함

- pymongo를 통해 database 연결함

A)

- 오바람이 가지고 있는 포켓몬 3가지의 id, name, spawn\_time, type, weakness를 알아내기 위해find 함수를 사용했고, 해당 결과를 cursor로 받아 python으로 공통 약점을 추출

- 해당 코드는 아래와 같으며 주석을 달아 설명하였음

**wind\_weak = []**

**wind\_pokemon = ['Scyther', 'Vileplume', 'Butterfree']**

**# 약점 찾기**

**# 오바람이 가지고 있는 3가지 포켓몬의 정보 찾기**

**weaknesses = pokedex.find({'name': {'$in': wind\_pokemon}}, {'id':1, 'name':1, 'spawn\_time':1, 'type':1, 'weaknesses':1, '\_id':0}).sort([('name', 1)])**

**for item in weaknesses:**

**wind\_weak.append(item['weaknesses']) # 각 포켓몬의 약점을 리스트로 받음**

**temp = wind\_weak[0] # 첫 번째 포켓몬의 약점**

**for i in range(1, len(wind\_weak)):**

**temp = set(temp)&set(wind\_weak[i]) # 모든 포켓몬이 가지고 있는 공통 약점 추출**

**wind\_weak = list(temp) # 최종 결과를 list로 변환**

- 공통 약점은 wind\_pokemon에 저장했으며 공통 약점은 ['Ice', 'Flying', 'Fire']

- 공통 약점과 출몰시간 조건을 넣어 조건에 맞는 db를 이름으로 오름차순 정리했음

- 해당 코드와 결과는 아래와 같음

**strong = pokedex.find({'$and': [{'type': {'$in': wind\_weak}},**

**{'spawn\_time': {'$gte': '20:00', '$lte': '24:00'}}]},**

**{'id':1, 'name':1, 'spawn\_time':1, 'type':1, '\_id':0}).sort([('name', 1)])**

**for item in strong:**

**print('{ ', end='')**

**for (k, v) in sorted(item.items()):**

**print('{}:{}'.format(k, v), end=', ')**

**print('\b\b }')**

{ id:142, name:Aerodactyl, spawn\_time:23:40, type:['Rock', 'Flying'] }

{ id:149, name:Dragonite, spawn\_time:23:38, type:['Dragon', 'Flying'] }

{ id:126, name:Magmar, spawn\_time:20:36, type:['Fire'] }

B)

- 이 문제를 풀기 위해서는 제일 먼저 최종 진화 포켓몬을 찾아야 함

- 최종 진화 포켓몬의 공통점은 ‘next\_evolution’ 필드가 없고, ‘next\_evolution’ 필드를 가지고 있음

- 최종 진화 포켓몬을 찾는 명령어는 아래와 같음(id로 오름차순 sorting)

final\_pokemons = pokedex.find({'next\_evolution':{'$exists':False},

'prev\_evolution':{'$exists':True}}).sort([('id', 1)])

- 최종 진화 포켓몬의 prev\_evolution num를 통해 진화 전 단계의 포켓몬 db를 가져와 진화에 필요한 candy\_count 합계를 구하고 최종 포켓몬의 이름과 진화 이전 이름, 진화에 필요한 candy\_count를 출력함

**n\_prev = len(pokemon['prev\_evolution']) # 최종진화 포켓몬의 전단계 수**

**for i in range(n\_prev):**

**pokemon\_num = pokemon['prev\_evolution'][i]['num'] # 전단계 포켓몬 num**

**prev\_pokemon = pokedex.find({'num': pokemon\_num}, {'\_id':0, 'candy\_count':1})**

**# 해당 포켓몬 db**

**for i in prev\_pokemon:**

**count = count+i['candy\_count'] # 진화를 위한 candy 수 합계**

**print(pokemon['name'], end=' => ')**

**print('{}: {} '.format(candy.encode('ascii', 'ignore').decode('ascii'), count))**