

OSSP 4차 보고서

<리그오브레전드 승/패 상관관계분석>

DHL

2016112190 한종호

2016112204 임정우

2016112231 정현성

2016112194 최재원

1. 4주차 프로젝트 진행상황

- 판별함수 구현 및 수정

탑, 미드, 캐리, 서폿은 각각 1개씩 -> '정글'은 조합 라인 두개(미드, 탑)를 조합으로 가진다.

```
if my_position == 0: # my_position is top
    comb_position = 1 # 탑 => 정글
    position_dict = positions[0]

elif my_position == 1: # my_position is jungle
    # 정글은 두 포지션 유무를 확인
    comb_position = 0 # 정글 => 탑
    comb_position2 = 2 # 정글 => 미드
    position_dict = positions[1]

elif my_position == 2: # my_position is mid
    comb_position = 1 # 미드 + 정글
    position_dict = positions[2]

elif my_position == 3: # my_position is carry
    comb_position = 4 # 원딜 => 서폿
    position_dict = positions[3]

elif my_position == 4: # my_position is sup
    comb_position = 3 # 서폿 => 원딜
    position_dict = positions[4]
```

Top, Mid, Carry, Sup은 같은 포지션의 상대 존재유무, 아군조합 1개의 존재유무 -> 총 4가지 케이스를 고려한다.

Jungle은 같은 포지션의 **상대 존재유무**, 아군 **조합1(Top)의 존재유무**, 아군 **조합2(Mid)의 존재유무** -> 총 8가지 케이스를 고려한다.

266	
1	0.5000
2	0.4545
3	0.5556
4	0.4130
5	0.4333
6	0.5000
7	0.4205
8	0.5424
9	0.6207

<사진>

total_win_rate DF

왼쪽의 사진은 total_win_rate 데이터 프레임의 일부분이다.
column과 row의 index는 챔피언의 고유 key값이다.
왼쪽 사진의 266은 '아트록스'라는 챔피언이고, 각각 9만게임에서 챔피언 key 1, 2, 3, 4, 5, 6 ... 총 151개의 챔피언들을 적으로 만났을 경우 승률을 나타냄

266	
1	0.5000
2	0.5000
3	0.4545
4	0.4935
5	0.5000
6	0.5000
7	0.4876
8	0.4857
9	0.6296

<사진>

comb_win_rate DF

왼쪽의 사진은 comb_win_rate 데이터 프레임의 일부분이다.
column과 row의 index는 챔피언의 고유 key값이다.
왼쪽 사진의 266은 '아트록스'라는 챔피언이고, 각각 9만게임에서 챔피언 key 1, 2, 3, 4, 5, 6 ... 총 151개의 챔피언들을 아군으로 만났을 경우 승률을 나타냄

상대(맞라인)가 있는 경우 total_win_rate를 참조,
아군 조합이 있는 경우 comb_win_rate를 참조

<아군조합과 상대(맞라인) 둘다 존재하는 경우>

```
if 아군존재 and 상대존재: # 아군o 상대o
    아군승률 = dict(comb_win_rate.loc[아군[0]][:])
    상대승률 = dict(total_win_rate.loc[상대[0]][:])
```

```
for k, v in 아군승률.items():
    추천픽리스트1.append([k, v])

for k, v in 상대승률.items():
    추천픽리스트2.append([k, round(1 - v, 4)])

for 픽 in zip(추천픽리스트1, 추천픽리스트2):
    최종추천.append(픽[0][0], 픽[0][1] * 픽[1][1])
```

* $\text{round}(1-v)$: v = 적👹가 X상대로 승리 확률
 $1 - v$ 는 X가 적👹 상대로 승리 확률이다.

```
최종추천.sort(key=lambda x: x[1], reverse=True)
```

추천받고 싶은 포지션이 Top이라고 가정을 해보자.

😊아군조합(Jungle) 과 👹적팀Top챔프 둘다 나온 경우
추천 받을 Top챔피언을 X 라고 하면

A = X와 😊가 함께 싸워서 승리확률

B = X가 👹를 상대로 승리확률

A와 B가 동시에 일어날 확률 $A*B$ 를 모든 X에 대해서 구하고, Top챔피언 X를 최종추천에서 ban_list에 포함x인 5개를 큰 순서대로 뽑아낸다.

(Mid, Carry, Sup)들도 이에 해당함

ex) 현재 나는 미드챔피언을 추천받고 싶은 상황이고, 아군에 리신이라는 정글 라인과,
적 미드라이너로 카타리나가 나왔다고 가정을 해보자.

리신 + 미드챔피언 X1 의 A1, 미드챔피언 X1 vs 카타리나 의 B1 -> $A1*B1$

리신 + 미드챔피언 X2 의 A2, 미드챔피언 X2 vs 카타리나 의 B2 -> $A2*B2$

리신 + 미드챔피언 X3 의 A3, 미드챔피언 X3 vs 카타리나 의 B3 -> $A3*B3$

.

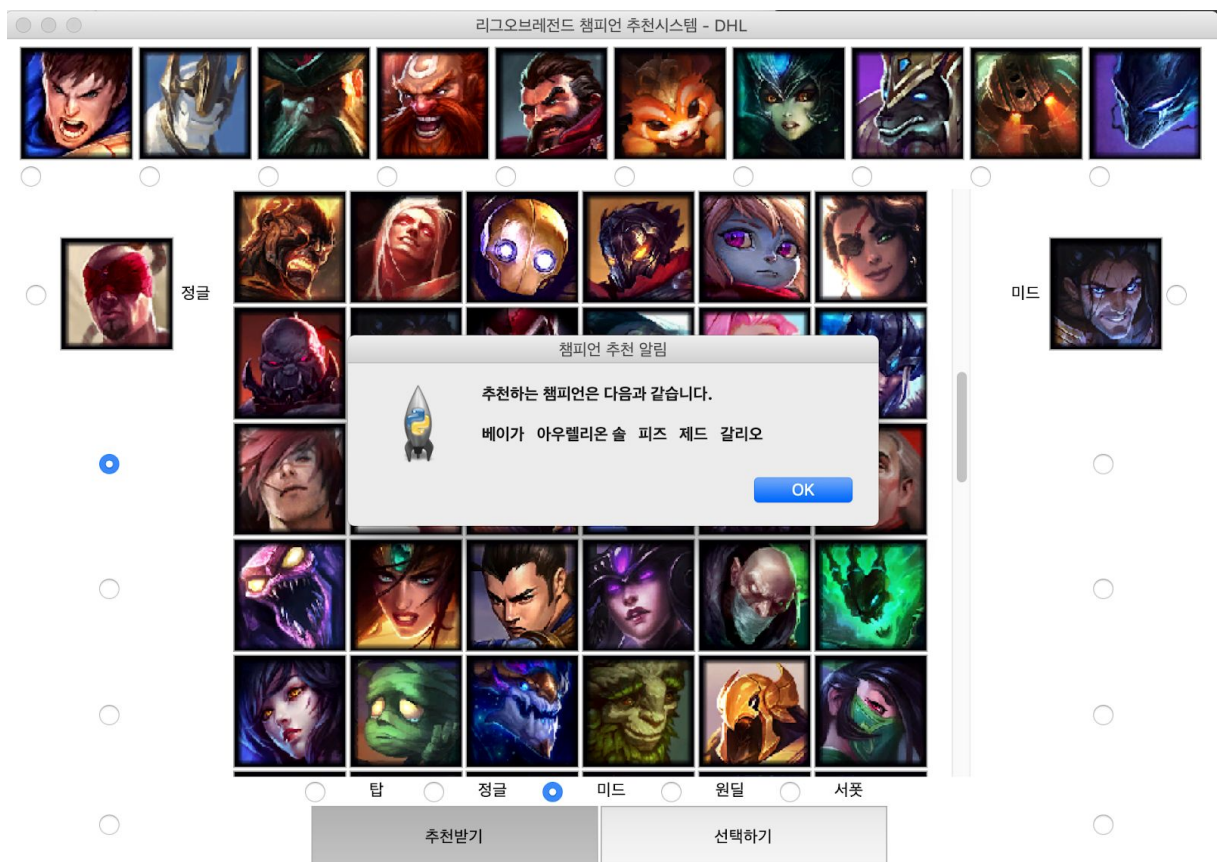
리신 + 미드챔피언 X_n 의 A, 미드챔피언 X_n vs 카타리나 의 B $\rightarrow A_n * B_n$ (n = 미드 챔피언의 개수)

최종추천 = $([X_1, A_1 * B_1], [X_2, A_2 * B_2], [X_3, A_3 * B_3], \dots, [X_n, A_n * B_n])$ 을 **Sorting**
출력 : **Sorting**된 최종추천에서 Ban_list에 없는 챔피언 X 5개를 출력한다.

추천받고 싶은 포지션이 **Jungle** 이면 😊아군조합1(Top)
😊아군조합2(Mid) 😡적팀Jungle챔프 세가지를 모두 고려한다.

- GUI수정된 부분

1) 추천 챔피언 출력 콘솔 -> 알림창 생성



- 2) 밴이나 챔피언 선택했을 시 이후 선택안되게 챔피언 버튼 비활성화
- 3) 상대 팀에서 여러개 챔피언을 넣었을 때 가장 마지막에 넣은 챔피언 기준으로 라인 추천 되는 버그 수정
- 4) 정글 라인을 추천할 때 기존 미드 라인과의 조합만 계산하던 부분에서 탑 라인과의 조합도 계산하게 변경

3. 최종까지 수정해볼만한 사항

- 1) 챔피언 선택했을 때 식별이 용이하게 변경
- 2) 아군 또는 적 챔피언만 나왔을 때 밴리스트에 있는 챔피언도 추천하던 버그 수정
- 3) 반복되는 코드 수정
- 4) .py파일 .exe파일로 변환 (pyinstaller 사용)