

Portfolio

SeonHong Cho

Introduction

- 성장하는 개발자, 조선흥입니다
- 도전정신을 토대로 다양하고 지속적인 경험, 그에 따른 성장을 추구합니다.
- 협업에 가장 중요한 의사소통 능력과 프레젠테이션 능력을 바탕으로 더 나은 미래를 꿈꾸고, 내가 가진 것을 나눌 줄 아는 사람을 지향합니다.

Summary

- 아주대학교 국방디지털융합학과 졸업 (2019.03 ~ 2023.02) - 심화트랙 : UAV제어
- Dev3B - Information Security Research(보안 팀스터디- 2019.03~2019.06)
- 졸업논문 - 다중 셀 기법을 이용한 UAV 군집 내 충돌회피(2023.01)
- 맛집을 찾아라! (데이터베이스) - 프로그램 설계 및 최종 코드리뷰(2021.05)
- ATM 시스템 만들기(객체지향프로그래밍) - atm 구동부 코드담당(2020.05)
- LOL_MDC loldamagesimulator(팀플젝) - 프론트엔드 담당(2023.07)
- 정보처리기사(2022.09), SQLD(2022.12), TOEIC(770, 2022.02)
- 한민캠프 강사 2회(2019.07, 2020.01)

Project.1

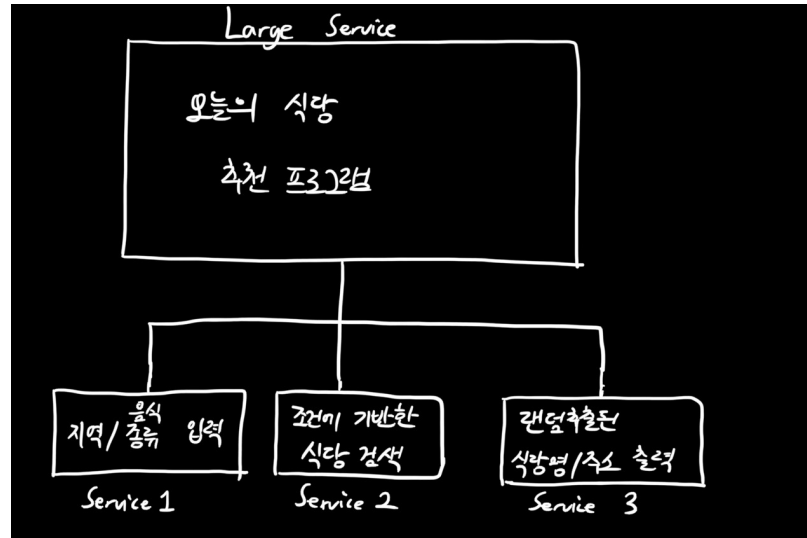
- 맛집을 찾아라!
 - 프로그램 설계 및 최종 코드리뷰(2021.05)

소속 : Meta View

수행기간 : 2021.05

- 3인 개발 팀 내에서 프로그램 설계 및 최종 코드리뷰를 담당
- 시스템 구조 설계, 사용자 인터랙션 시나리오 작성
- 공공데이터포털에 있는 안심식당 목록([API](#)) 사용
- DBMS : [Postgresql](#), 개발 언어 : [JAVA](#)

Project.1



```
if(s.equals("1") || s.equals("고양시")) // 숫자 1 혹은 "고양시" 입력 시
{
    System.out.println("\n고양시의 음식을 불러오는 중입니다...");
    Thread.sleep(3000); // Loading

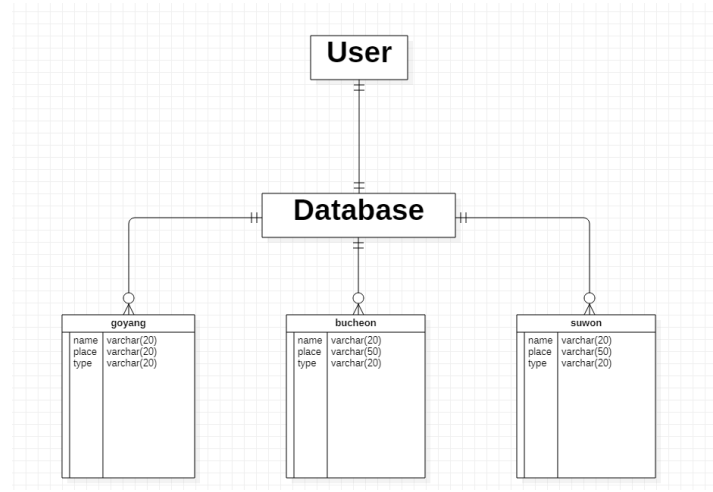
    System.out.println("\n음식 종류를 골라주세요.\n");
    Statement st = connection.createStatement();
    ResultSet rs = st.executeQuery("select distinct nType from goyang"); //고양시의 식당 중 음식 종류
    ArrayList<String> arr = new ArrayList<>();
    while(rs.next())
    {
        arr.add(rs.getString("nType")); // String 배열(정확히는 String을 포함한 ArrayList)에 음식 type 저장
    }

    Collections.sort(arr);
    Collections.reverse(arr); // 음식 type 내림차순 정렬 (기타를 마지막 번호로 보내기 위함)

    int arrSize = arr.size(); // type 개수

    for(int i = 0; i < arrSize ; i++)
    {
        System.out.printf("%1d.%s ", i+1 , arr.get(i));
    }
    System.out.println();
    System.out.printf("입력 : ");
    String ss = scan.next(); // 음식 타입 정하기
}
```

그림 4 사용자로부터 입력받은 '지역'과 '음식 종류'를 바탕으로 해당 지역의 음식점을 불러오는 SQL



DB의 구성

```
for(int i = 0; i < arrSize ; i++)
{
    if(ss.equals(Integer.toString(i+1)) || ss.equals(arr.get(i))) // 1~arraySize번까지 일치하는게 있는 경우 그 type 에 해당하는 랜덤 1개 출력
    {
        System.out.printf("\n%s을 고르셨군요!\n",arr.get(i));

        String que = "select nName , nAddress "
            + "from goyang "
            + "where nType ="
            + arr.get(i)
            + "' "
            + "order by random() limit 1;"; // order by random()을 사용하여 랜덤 추출
        Statement state = connection.createStatement();
        ResultSet rss = st.executeQuery(que);
        while(rss.next())
        {
            System.out.printf("오늘의 추천 식당입니다!\n\n이름 : %s\n주소 : %s\n\n",
                rss.getString("nName"),rss.getString("nAddress")); //식당 이름, 주소 출력
        }
        rss.close();
        state.close();
        break;
    }
}

if(i == arrSize-1)
{
    System.out.println("\n입력이 잘못되었습니다. 프로그램을 종료합니다."); // 입력 오류 시 프로그램 종료.
}
}
```

그림 5 불러온 Tuples 중에 무작위로 하나의 '식당명'과 '주소'를 출력하는 SQL

SQL 코드리뷰

Project.2

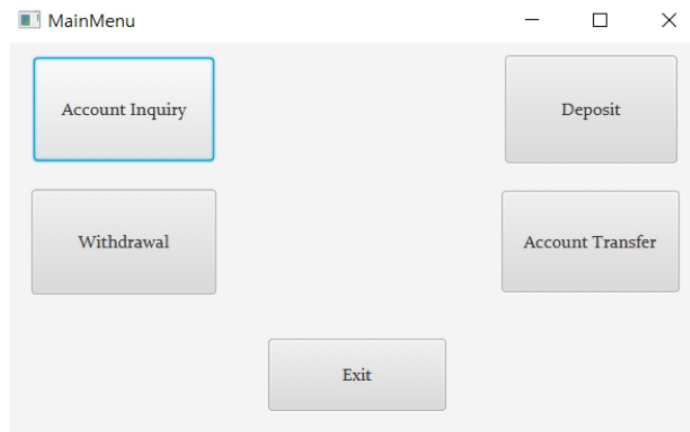
- ATM 시스템 만들기
 - atm 구동부분 담당(2020.05)

소속 :

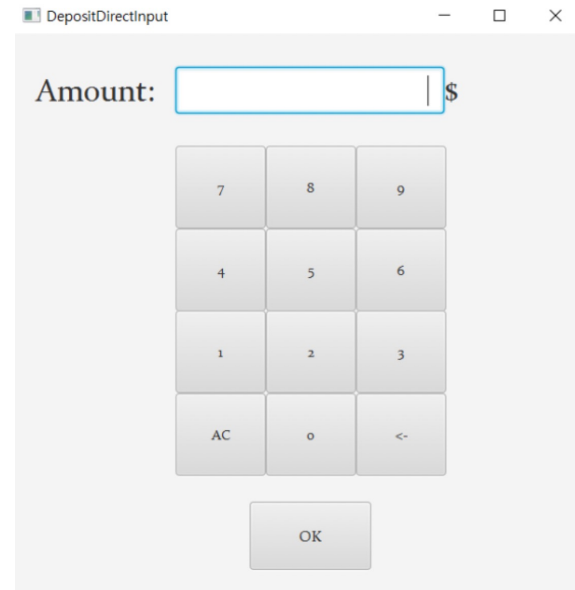
수행기간 : 2020.05 ~ 2020.06

- 3인 개발 팀 내에서 [GUI](#) 기본구조 구현
- ATM 시스템이 작동하는 데 필요한 Controller부 코드 작성
- DBMS : [Mysql](#), 개발 언어 : [JAVA](#)

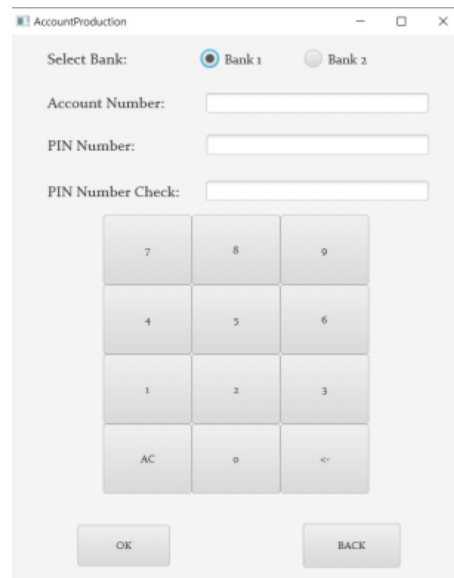
Project.2



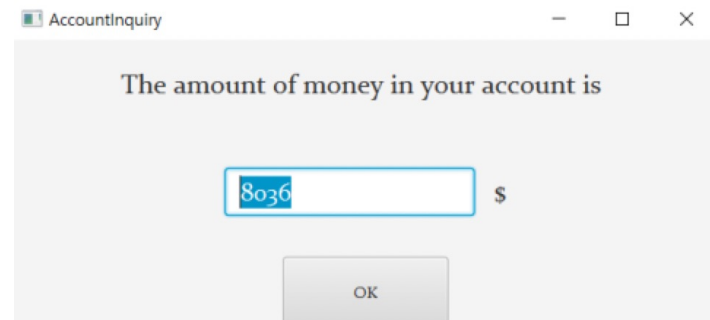
초기화면



입금액 입력



계좌생성



계좌 잔액확인

Project.3

- LOL_MDC loldamagesimulator
 - Frontend 담당(2023.07)

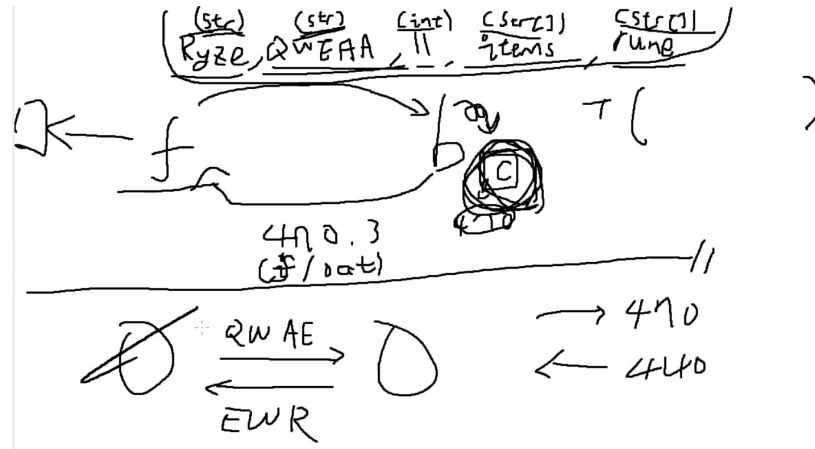
소속 : BAD_MDC

수행기간 : 2023.04 ~ 2023.08

- 3인 개발 팀 내에서 Frontend 담당
- League of Legend에서 챔피언이 특정 조건일 때의 값과 콤보를 입력받아 그 데미지를 계산하는 홈페이지 작성 및 기능구현
- [Django, React, Node.js, CSS, Git](#) 사용

Link : <https://bad-mdc.github.io/loldamagesimulator/>

Project.3



초기구상

```

10 ap = 0
11 total_dmg = [0.0, 0.0, 0.0]
12 u_dmg = getRealSkillDamage(champion, "Q", level[0], [ad_bonused_ap, atkspd])
13 w_dmg = getRealSkillDamage(champion, "W", level[1], [ad_bonused_ap, atkspd])
14 e_dmg = getRealSkillDamage(champion, "E", level[2], [ad_bonused_ap, atkspd])
15 r_dmg = getRealSkillDamage(champion, "R", level[3], [ad_bonused_ap, atkspd])
16
17 for skill in combo:
18     for i in range(0, 3):
19         if skill == 'q' or skill == 'Q':
20             total_dmg[i] = total_dmg[i] + u_dmg[i]
21         elif skill == 'w' or skill == 'W':
22             total_dmg[i] = total_dmg[i] + w_dmg[i]
23         elif skill == 'e' or skill == 'E':
24             total_dmg[i] = total_dmg[i] + e_dmg[i]
25         elif skill == 'r' or skill == 'R':
26             total_dmg[i] = total_dmg[i] + r_dmg[i]
27
28 print(total_dmg)
29
30 damageSimulator("Akali", "qwer", 13, "none", "5152")

```

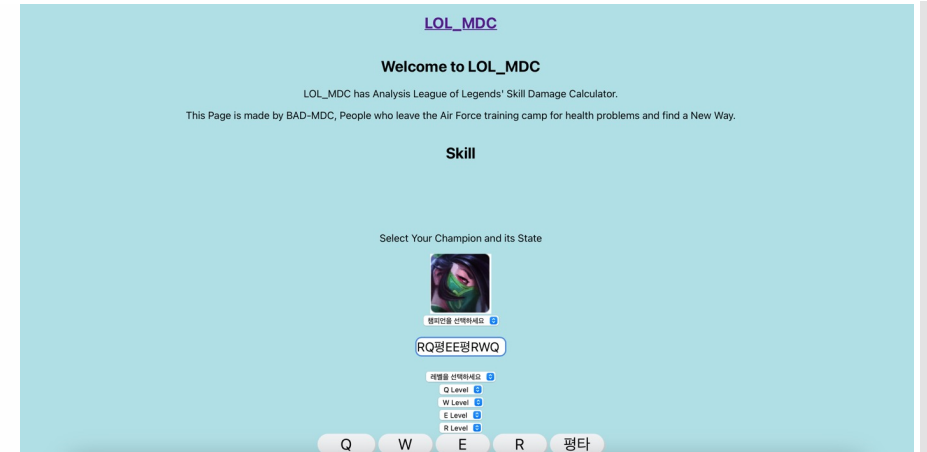
Run

```

C:\Users\cray\AppData\Local\Microsoft\WindowsApps\python3.10.exe C:\Users\cray\PycharmProjects\lolDamageSimulator\backend\loldata\damageSimulator.py
[0.0, 1248.44, 0.0]
Process finished with exit code 0

```

스킬 데미지 계산

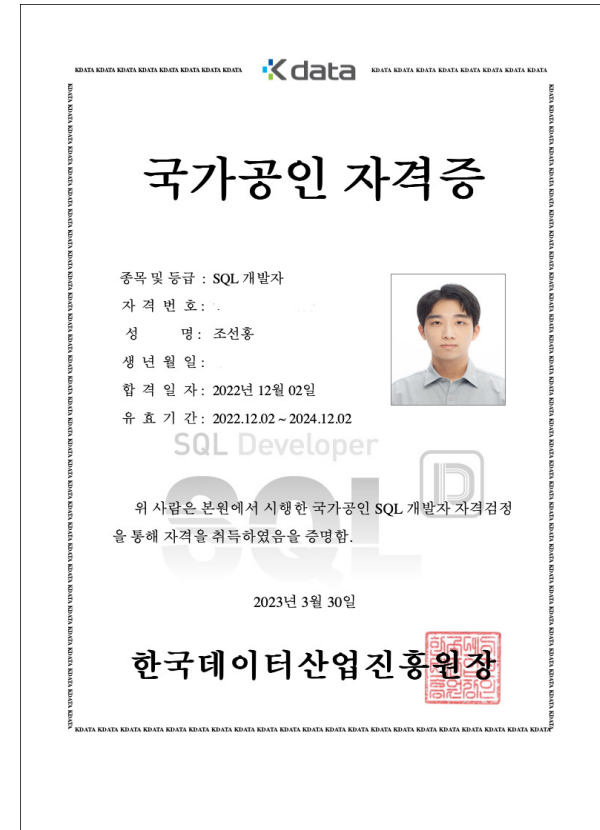


Ver 1.0 완성본

License



- 정보처리기사(2022.09)



- SQLD(2022.12)

Bachelor's thesis

Collision Avoidance of UAVs in Swarm Using Multi-Cell Strategy

- 기존 Leader - Follower 기법의 군집 UAV는 Leader의 움직임에 따라서 Follower의 움직임이 결정. Leader에 문제 발생시 Follower의 움직임 예상불가, 군집 전체 제어불능 문제 + 신뢰성을 항상 보장할 수 없음
- 논문에서 제안한 다중 셀 기반 방식은 각 UAV에게 충돌 회피 알고리즘을 부여. 또한 군집 내 각 UAV가 다중 셀 방식을 이용하고 특정한 Leader를 사용하지 않기 때문에 기존의 방식에서 보장할 수 없었던 신뢰성 문제 해결
- MATLAB 시뮬레이터로 성능분석 결과 제안하는 기법이 기존 기법에 비해 우월한 성능이라는 점은 보장하지 못하지만 다중 셀 방식이 군집 내 UAV의 제어에 사용될 때에도 유효함을 확인

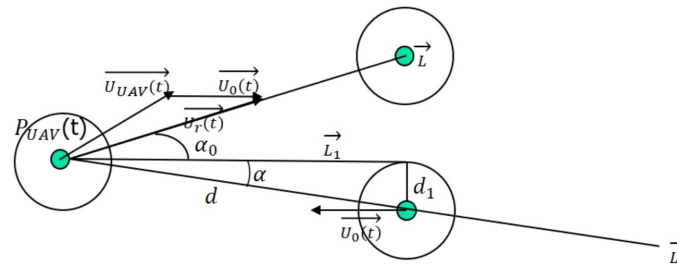


그림 3.2. 군집 내 UAV들의 충돌 회피 상황 도식화

Fig. 3.2. Schematic diagram of collision avoidance situations of UAVs Swarm

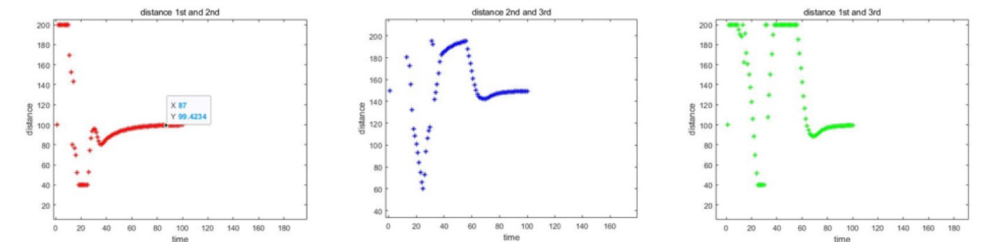


그림 4.2. 시간 흐름에 따른 각 UAV간 거리 변화

Fig. 4.2. Variation of Distance Between Each UAV over Time

Tutoring

- 한민캠프 강사 2회(2019.07, 2020.01)
 - 중학생 군자녀들을 각각 5일간 인솔하고, 본인의 학창시절 경험 등을 공유



The background features two large teal geometric shapes. On the left is a parallelogram, and on the right is a triangle, both pointing towards the center where the text is located.

Thank you for Watching !