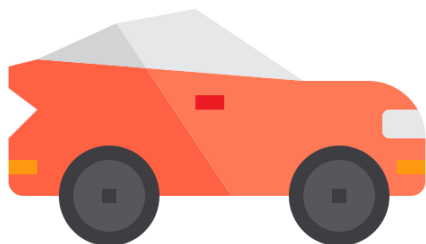


5팀

Racing Matrix

김현우, 김보윤, 이자연



Contents

1 프로젝트 소개

- 프로젝트 설명
- 팀원 별 역할

2 구현 방법

- 사용 하드웨어
- 사용 모듈
- 코드 설명

3 시연 영상

- 감상

4 마치며...

- 시행착오
- 배운 점

01 프로젝트 소개

프로젝트 설명

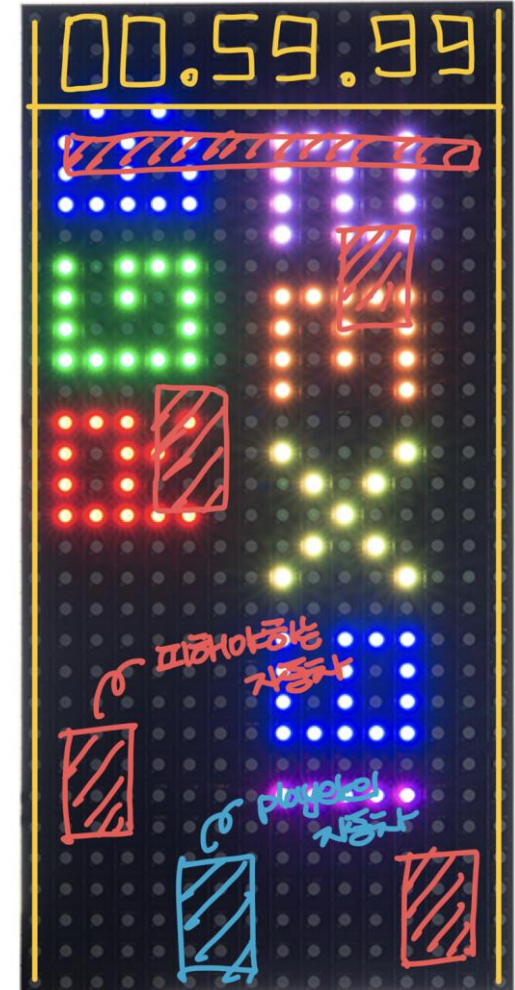
단순한 조작과 음악, 16x32의 크기에 맞춘 UI가 특징인 레이싱 게임
시작화면, 플레이화면, 종료화면의 3페이지로 구성된 프로젝트입니다.

시작화면에서 소리를 on off 설정할 수 있으며 사용자의 이름을 입력할 수 있습니다.

키보드(a, d)를 사용하여 플레이어의 자동차가 다른 자동차에 부딪히지 않게 조종합니다.
다른 차량과 부딪히게 되면 처음 160 hp에서 60만큼 감소하게 됩니다.
랜덤으로 발생하는 아이템 두 가지중 하나를 획득하면 40만큼의 hp가 회복되거나
화면에 있는 방해 차량이 화면에서 지워집니다.

hp가 0이 되어 게임이 종료하게 되면 사용자의 이름과 순위, 점수를 보여줍니다.
사용자가 3위안으로 점수를 기록하게 된다면 시작화면에 사용자가 입력한 이름이 나타납니다.

처음 구상한 matrix racing 화면



01 프로젝트 소개

팀원 별 역할

김현우

시작/플레이/종료 화면 UI 제작

장애물과 아이템 랜덤 생성

아이템 획득 시 아이템 효과 구현

Hp에 따른 플레이어 차량과
게이지 색상 동기화

Timer 구현, 점수로 변환

플레이어 이름을 문자 배열로 변환

김보운

플레이 기본 흐름 코드 작성

문자 알파벳 배열 생성

플레이어의 입력과 점수를
dictionary로 만들고
.pickle 파일에 저장

.pickle 파일을 읽고
플레이어의 순위 반환
& 랭킹 1-3위 이름 출력

Timer 숫자를 천, 백,
십, 일의 자리에 맞춰 parsing

이자연

Sound 자료 조사, 음원 파일 가공

차량과 장애물의 충돌 판정

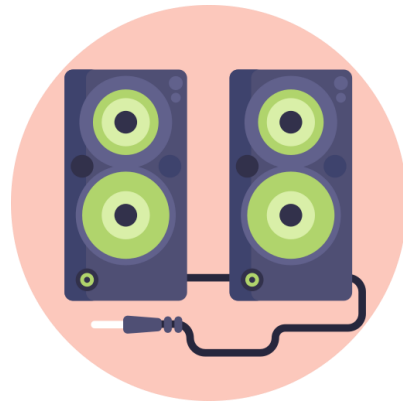
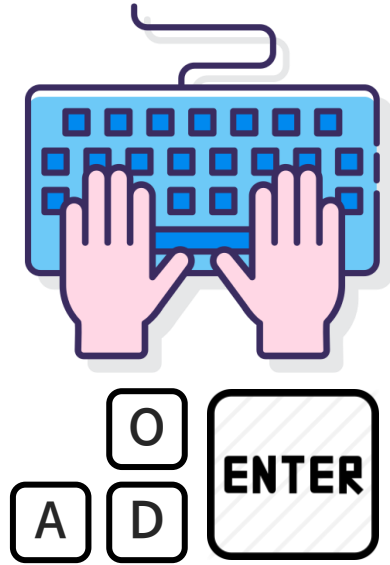
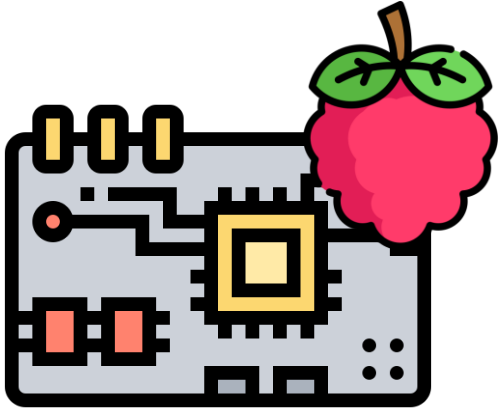
충돌에 따른 효과 구현
(hp감소/효과음)

게임 상황 별 sound 설정
(hp감소/효과음)

플레이화면 출력 전 카운트다운 구현

02 구현 방법

사용 하드웨어 & 모듈



pygame 모듈에서 mixer라이브러리를 이용
(배경음악, 효과음 삽입)



pickle 모듈을 이용해 랭킹 목록을 파일형태로 저장



time

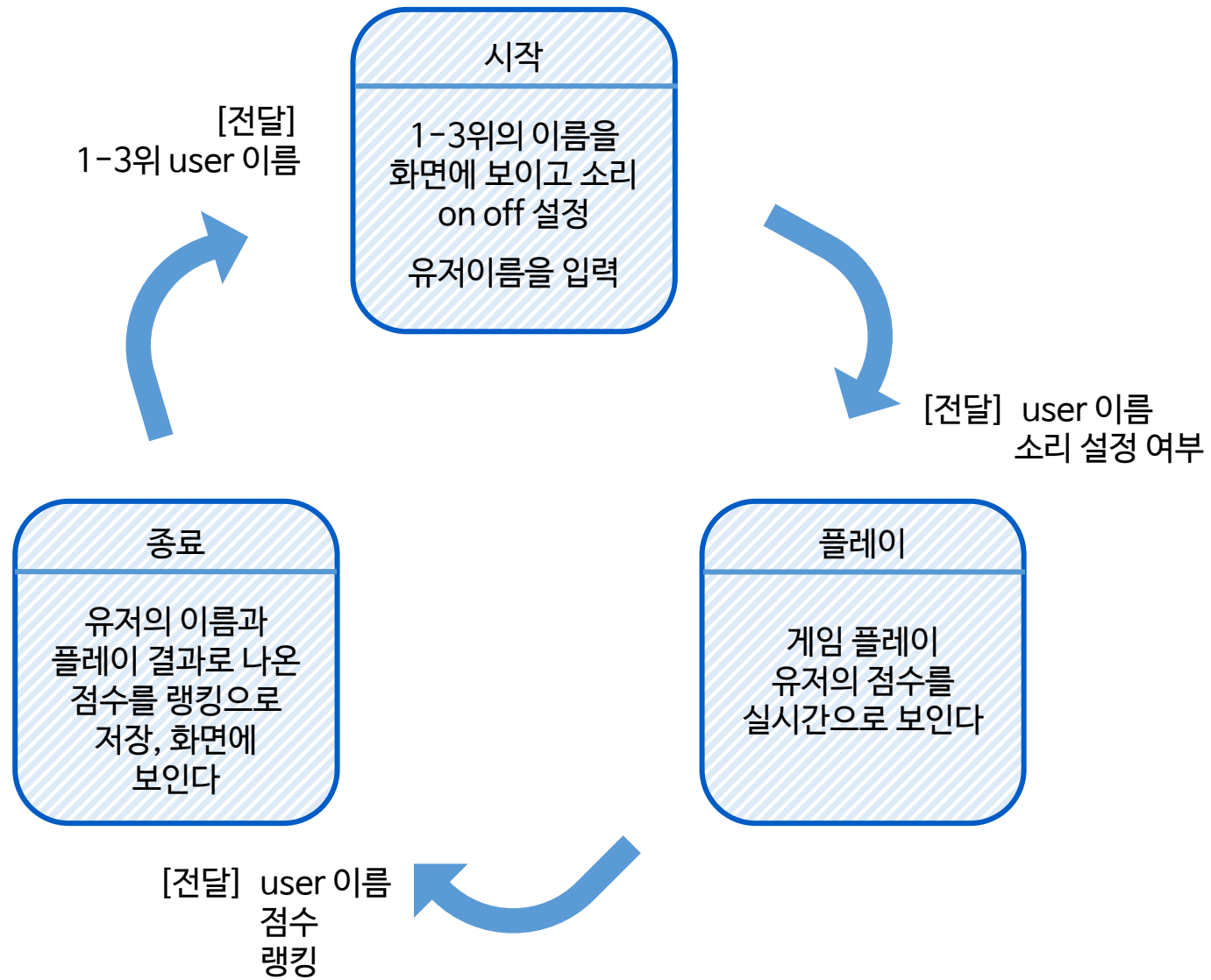
전반적으로 게임 진행속도를
늦추기 위해 time.sleep을 사용

timeit

게임에 필요한 타이머를 구현하기 위해
현재시각을 timeit.default_timer()로 불러옴

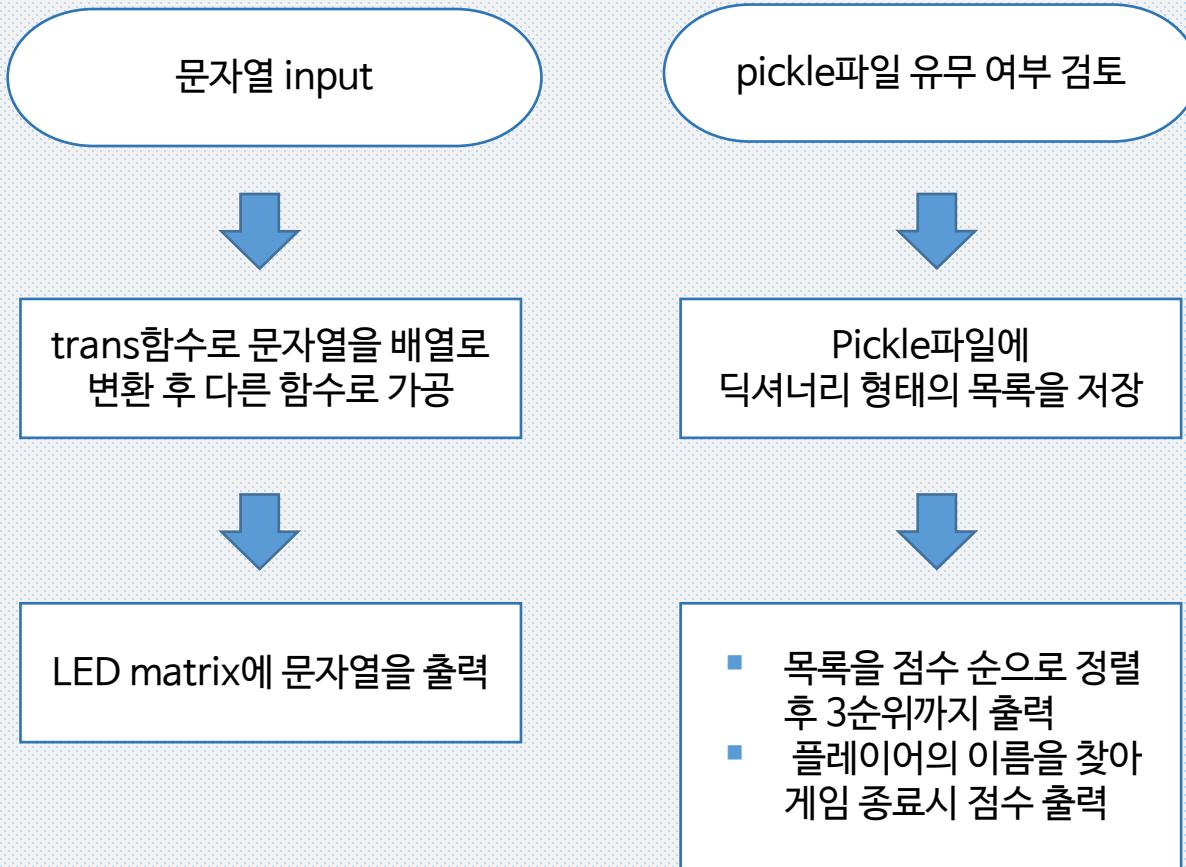
02 구현 방법

전체 흐름

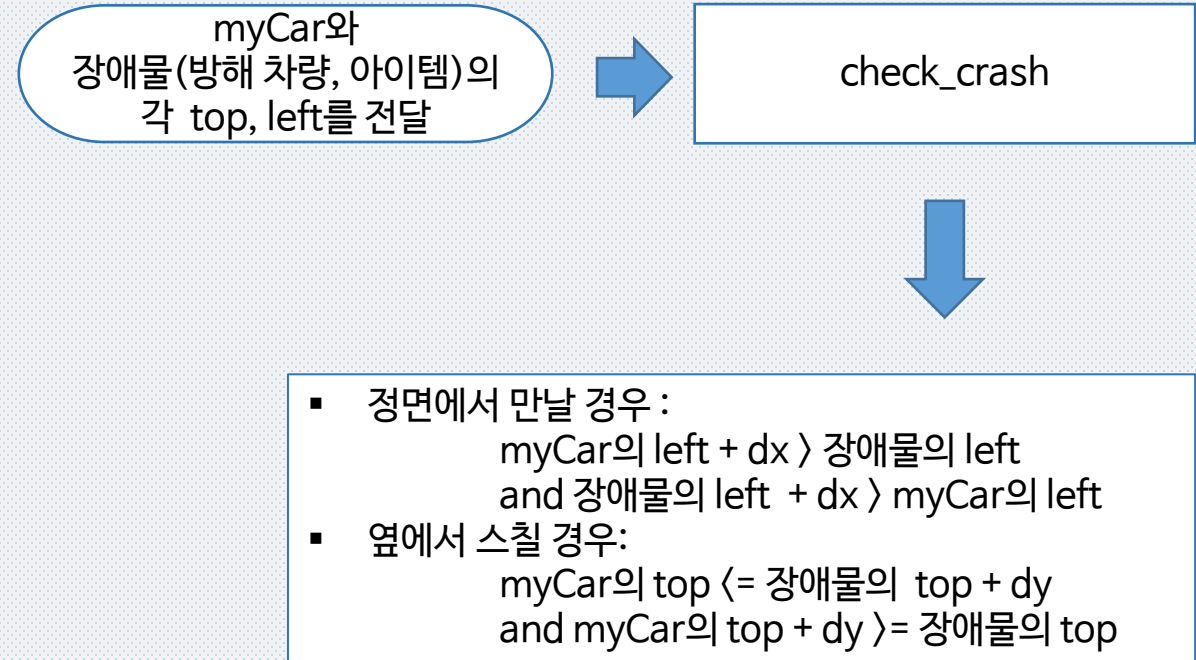


02 구현 방법

문자 배열화 및 랭킹시스템

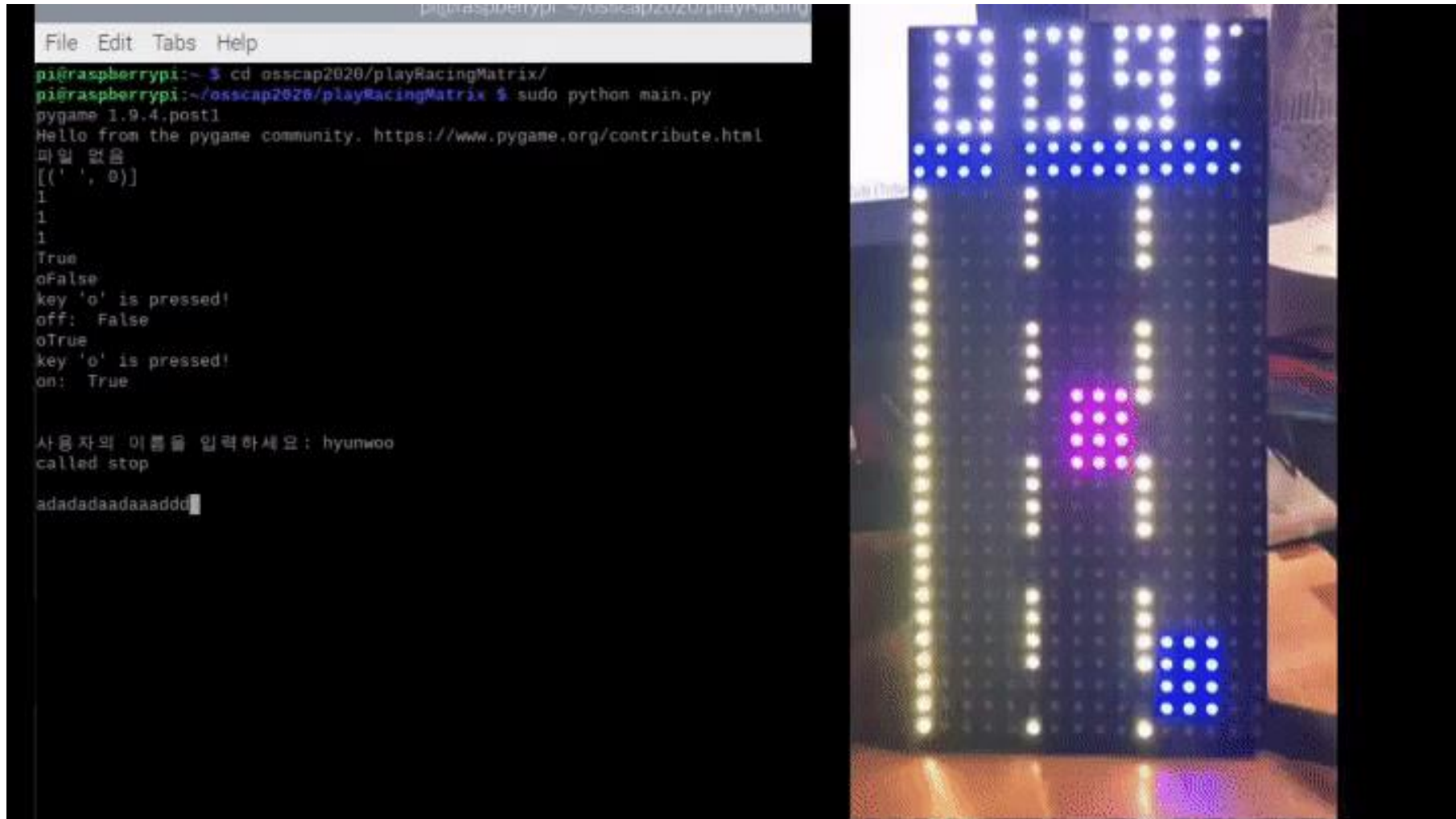


차량 충돌 처리



03 시연 영상

<https://youtu.be/LtkLSI5j5KM>



04 프로젝트를 마치며...

시행 착오 1

프로젝트 마치기까지 겪었던 문제점들

프로젝트가 중간 단계쯤 되었을 때, 파이게임 스크린을 LED matrix에 띄우는 것은 우리 팀이 하기에는 지식적인 면에서 무리가 있다고 판단했다.

따라서 파이게임 모듈을 통한 게임 만들기에서 파이썬 배열을 통해 게임을 만들게 되었다. 게임 모듈없이 배열로 만들었기 때문에 조금 더 복잡해지고 어려워졌다. 또, 그때까지 작성한 200줄 가량의 코드 또한 폐기하게 되었다.

팀장으로서 팀원들 역할 분배에 신경을 써야 한다는 게 어려웠다. 텍스트 스크롤을 검색해봐도 우리 프로젝트에 바로 사용할 수 있는 게 없어서 직접 만들어야 했다. pygame으로 그래픽을 쉽게 만들고 화면에 출력하려는 계획이었는데, led matrix와 pygame 실행 창을 동기화하는 방법을 찾지 못해 배열로 작성하게 되었다.

충돌 시 HP 게이지가 감소하는 것을 구현해내기 위해 다양한 방법을 시도하다보니 시간이 지체되는 경향이 있었다. gauge 배열을 4개로 쪼개 장애물 차량과 같은 방식으로 hp 값에 따라 크기가 다른 배열을 할당하고 이를 oscreen에 paste하였는데 코드의 길이나 동작 측면에서 최적화되지 않은 것 같았다. 코드의 길이가 길다보니 어느 방식으로 코드를 짜는 것이 직관적으로 눈에 들어오지 않아서 어렵게 느껴졌던 것 같다.

배운 점 2

프로젝트를 통해 느낀 점

프로젝트를 진행할 때 큰 문제점이 발생했을 때 이를 타개할 방안으로 지금까지 해온 결과물을 버릴 수 있는 과감함도 때론 필요하다는 것을 느꼈다. 또, 코드를 함수형태로 하는 것의 편의성 또한 알게 되었다

언택트로 프로젝트 대부분을 진행하고 팀원들의 진행사항을 자주 확인하는 것이 생각만큼 쉽지 않다고 느꼈다. 팀원들의 진행사항을 확인하고 그에 맞춰서 역할을 분배하는 방법을 배웠다. 어떻게든 작동되는 코드를 작성하다보니, 코드리뷰와 클린코드 작성의 필요성을 느꼈다. 중복되는 부분을 나중에라도 꼭 함수로 작성해야겠다고 다짐했다.

프로젝트를 진행하다보니 자신이 코딩하는 방식에 어떠한 문제점이 있는지 바로 알 수 있게 되는 것 같다. 코드의 구조화를 먼저 하지 않고 바로 코딩을 시작하곤 했는데, 프로젝트를 수행하면서 중간에 막히거나 긴 코드로 인해 수정도 쉽지 않음을 느꼈다. 따라서 코딩을 하는 습관을 새롭게 기를 수 있었던 것 같다.

감사합니다
