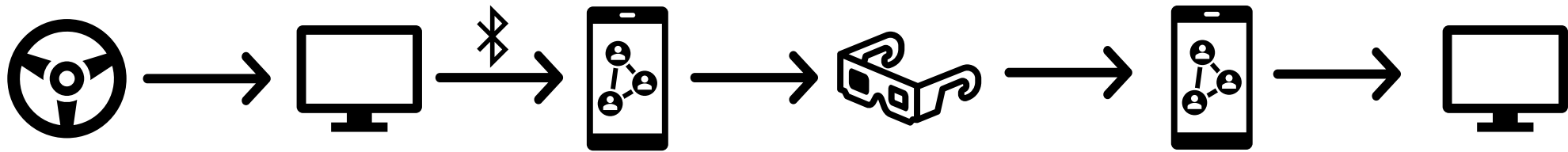


# 유저 데이터 분석 팀

유니티 & 웹 유저 데이터 분석 및 시각화



# Flow



1. 센서 - 노트북 간 통신을 통해 유저의 주행 데이터를 취득한다.
2. 노트북 - 모바일 간 Bluetooth serial 통신을 통해 input을 시뮬레이션에 적용한다.
3. 유니티를 통해 시뮬레이션 환경을 구축하고 유저의 주행 능력을 평가한다.
4. VR 기기를 통해 모바일의 주행 환경을 VR로 재생한다.
5. 웹을 통해 취득한 주행 데이터 및 유저 평가를 DB에 저장 & 시각화하여 보여준다.

# 운전자 유형별 임계값 지표

Table 2. 위험운전유형별 논리적 임계값(9개 유형)

위험운전유형	판단변수	논리적 임계값
직진구간 과속	속도	법규에 준함
선회구간 과속	횡방향가속도	0.2 g
	요 각속도	8.6 deg/sec
급격한 좌(우)회전	횡방향가속도	0.2 g
	요 각속도	8.6 deg/sec
급출발	종방향가속도	0.2 g
급가속	종방향가속도	0.1 g
급정지	종방향가속도	-0.4 g
급감속	종방향가속도	-0.4 g
급차선 변경	주기 및 주파수	4초
	횡방향가속도	0.2 g
	요 각속도	5.0 deg/sec
연속적인급차선 변경	주기 및 주파수	8초
	횡방향가속도	0.2 g
	요 각속도	7.0 deg/sec

주: 오주택 외(2008)

논리적 임계값

인간공학적 임계값

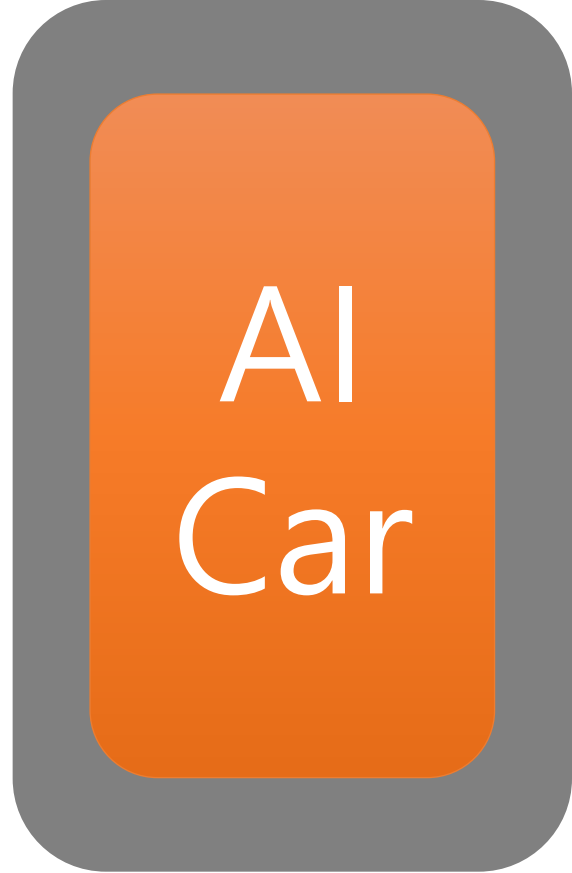
두가지 기준 존재

# 유저 평가

- 유니티 이벤트 함수(using UnityEngine.Events) 사용
- 이벤트 발생 시 체크
- 맵 내의 Object와의 충돌
- 운전자 유형별 임계값 지표 기준과 비교 (자차)
- 신호위반, 정지선 위반, 중앙선 침범

# 오브젝트 처리

- 프리팹(Prefab) 이용
- 하나의 Game Object를 여러 개 사용할 때 프리팹 사용



AI차량을 실제 크기보다 큰 범위로 구현 (도로의 크기 내)

운전자가 AI과 충돌 시 충돌 발생을 알리고 감점을 주어 종합 점수에 영향을 끼친다.

\*다른 이벤트에도 자차를 기준으로 설계할 예정

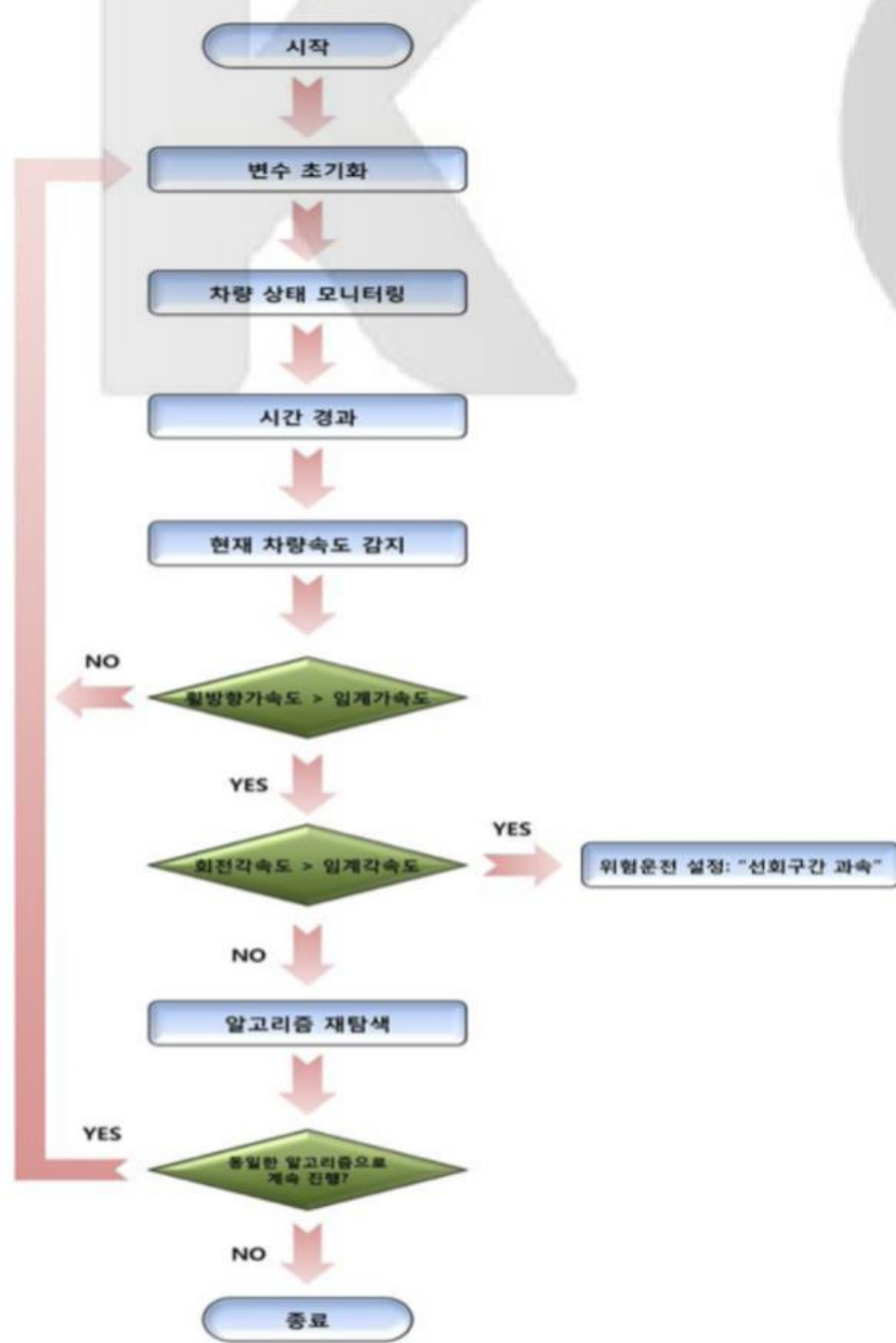


Figure 9. 선회구간 과속 알고리즘

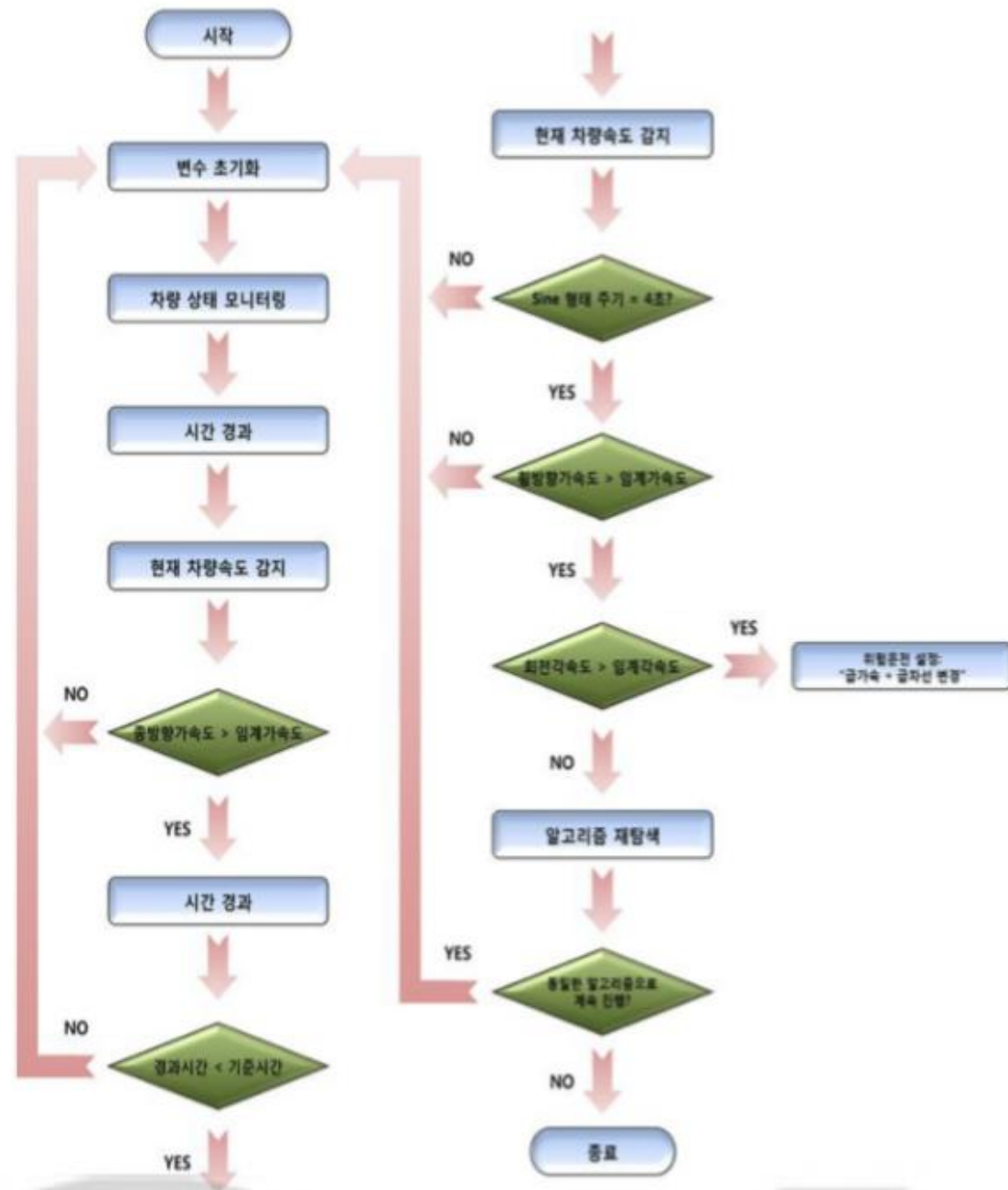
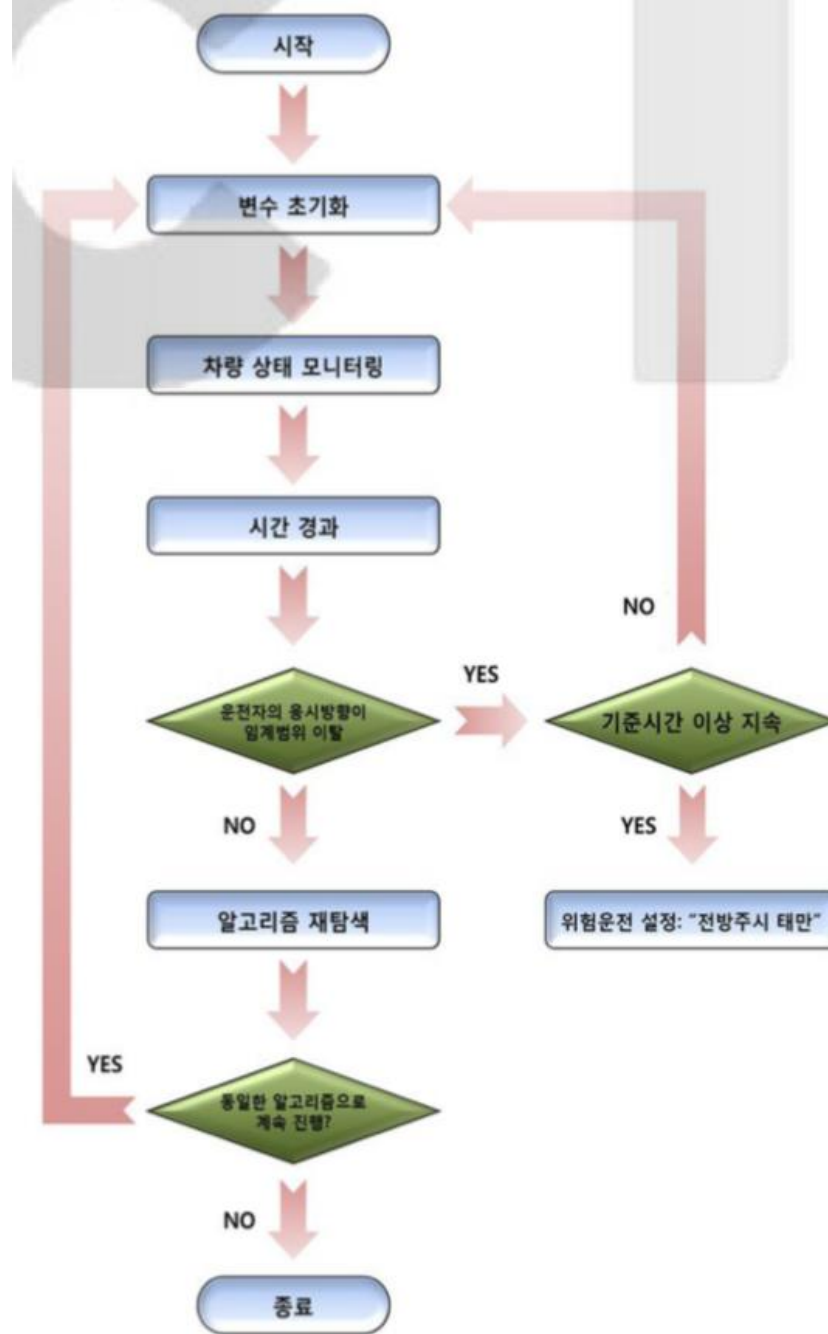


Figure 10. 급가속 + 급차선 변경 알고리즘





HMD에서 시야 포인팅 활용

시스템 통합 및 디버깅 이후  
추가 기능 구현 단계에서 구현

Figure 11. 전방주시 태만 알고리즘

# 주행 후 결과

- 내 주행 평가 정보를 구체적으로(시각화) 볼 수 있도록 한다.
- 내 정보를 DB상의 다른 유저들과 비교할 수 있도록 한다.
- 내 정보를 **위험운전유형별 임계값**과 비교할 수 있도록 한다.
- DB에서 원하는 조건의 운전자를 필터링하여 정보를 볼 수 있도록 한다.

유니티에서 유저 데이터 분석(성향 등)

운전 종료 후 분석한 데이터를 보여줌

결과 비교

앱 -> 웹 유니티에서 유저 데이터를 JSON으로 전송

DB에 등록 (유저 특징과 함께 등록)

유저와 기존 데이터간 비교 등을 시각화해 보여줌

\* 웹에서 따로 DB에 저장된 데이터를 필터링하여 시각화

# 데이터의 이동





유니티 -> 웹 : json 으로 전송

웹 어플리케이션 서버에서 데이터를 받아 연동된 DB로 처리한 결과를 앱으로 전송

데이터 시각화 이미지 전송 및 표시 (아직 미정, 그래프 시각화 방법에 따라 변동)

WEB 및 DB는 Python으로 구현할 예정

# DB 구성

나이 / 성별 / 지역 / 운전 성향 / etc

아직 미정

# 공부할 것

- 애리

- ✓ Web server
- ✓ Data Base
- ✓ AWS 클라우드 서버 구축
- ✓ 데이터 시각화

- 종욱

- ✓ Unity
- ✓ 데이터 처리(csv, JSON)

- 공통

- ✓ C#, python