

종합설계 프로젝트 수행 보고서

프로젝트명	고령 운전자들의 위험군 판별을 위한 운전 시뮬레이션 콘텐츠 개발
팀번호	S4-10
문서제목	수행계획서() 2차발표 중간보고서() 3차발표 중간보고서() 최종결과보고서(O)

2020.06.27

팀원 : 이 강 우 (팀장)
박 준 형
권 지 영

지도교수 : 정 성 택 교수 (인)

문서 수정 내역

작성일	대표작성자	버전(Revision)	수정내용	
2020.01.14.	이강우(팀장)	1.0	수행계획서	최초작성 팀 역할 분담 개발 일정 개발 환경 개발 내용 작 성
2020.01.15.	권지영(팀원)	1.1	수행계획서 수정	작품선정 배경 및 필요성 추가 개발 목표 추가
2020.01.16	박준형(팀원)	1.2	수행계획서 수정	기존 연구 분석 기술 동향 분석 개발 내용 문제 및 해결 방안 시험 시나리오
2020.03.02	박준형(팀원)	2.0	상세 설계	소프트웨어 설계 시나리오 맵 설계 이벤트 설계 AI 주행차량 설계 자동차 연동 설계
2020.04.06	권지영(팀원)	2.1	상세 설계	이벤트 설계 시선추적 설계
2020.04.07.	이강우(팀장)	2.2	상세 설계	DB 상세 설계 로그인, 메뉴 설계
2020.04.28.	이강우(팀장)	2.3	상세 설계 수정	수정된 UI 적용 Record 씬 추가
2020.04.30.	이강우(팀장)	2.4	Prototype 구현	로그인 및 메뉴 시나리오 진행 및 결과화면 확인
2020.05.01.	권지영(팀원)	2.5	Prototype 구현	하드웨어 (VR) 하드웨어(TOBII) 하드웨어(Logitech)
2020.05.01.	이강우(팀장)	2.6	최종 점검	-
2020.06.24.	이강우(팀장)	3.1	6, 7 목록 추가	-
2020.06.24	권지영(팀원)	3.2	시험/테스트 결과	시나리오 3 변경 사항 추가
2020.06.24	박준형(팀원)	3.3	시험/테스트 결과	시나리오 1 변경사항 추가

2020.06.26	권지영(팀원)	3.4	시험/테스트 결과	프로젝트 변경사항 추가
2020.06.27	박준형(팀원)	3.5	시험/테스트 결과 Coding & DEMO	시나리오 진행(VR) 이벤트 실행
2020.06.27	권지영(팀원)	3.6	시험/테스트 결과 최종 점검	시나리오 진행(Tobii) -
2020.11.10.	이강우(팀장)	3.7	결론 초안 작성	-

문서 구성

진행단계	프로젝트 계획서 발표	중간발표1 (2월)	중간발표2 (4월)	학기말발표 (6월)	최종발표 (10월)
기본양식	계획서 양식	계획서 양식	계획서 양식	계획서 양식	계획서 양식
포함되는 내용	I. 서론 (1~6)	I. 서론 (1~6)	I. 서론 (1~6)	I. 서론 (1~6)	I
	II. 본론 (1~3)	II. 본론 (1~4)	II. 본론 (1~5)	II. 본론 (1~7)	II
	참고자료	참고자료	참고자료	참고자료	III

이 문서는 한국산업기술대학교 컴퓨터공학부의
“종합설계” 교과목에서 프로젝트
“고령 운전자들의 위험군 판별을 위한 운전 시뮬레이션 콘텐츠 개발”
을 수행하는 S4-10 이강우, 박준형, 권지영 등이 작성한 것으로
사용하기 위해서는 팀원들의 허락이 필요합니다.

목 차

I . 서론

1. 작품선정 배경 및 필요성	5
2. 기존 연구/기술동향 분석	6
3. 개발 목표	7
4. 팀 역할 분담	7
5. 개발 일정	7
6. 개발 환경	8

II . 본론

1. 개발 내용	9
2. 문제 및 해결방안	9
3. 시험 시나리오	11
4. 상세 설계	13
5. Prototype 구현	30
6. 시험 / 테스트 결과	35
7. Coding & DEMO	48

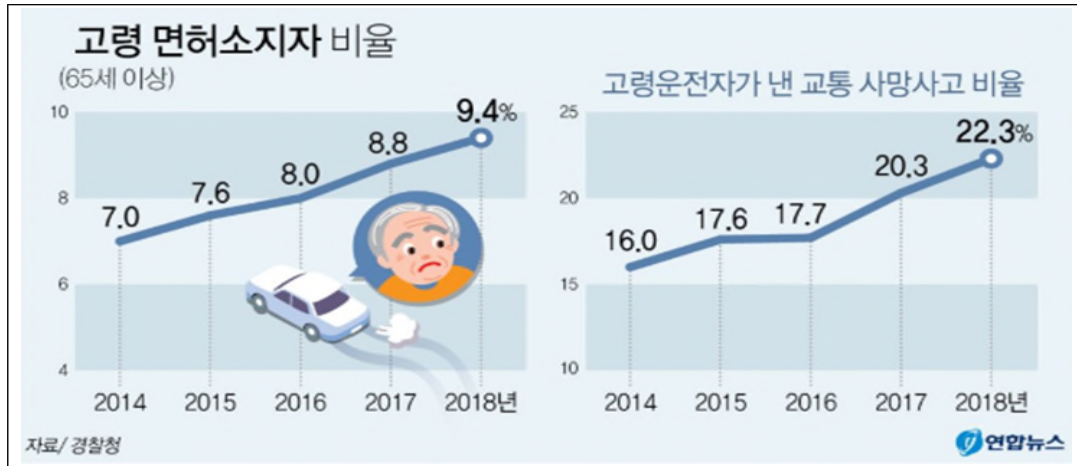
III . 결론

1. 연구 결과	53
2. 작품제작 소요재료 목록	54

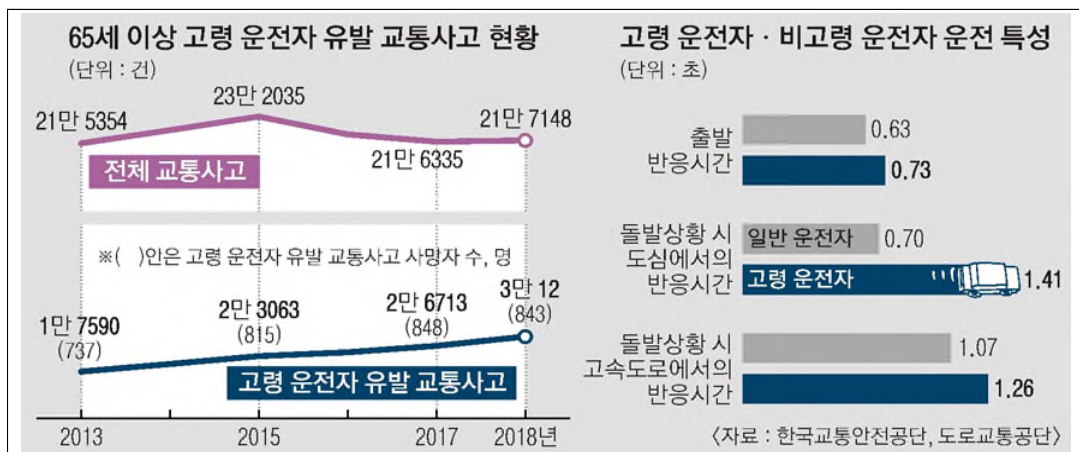
참고자료	55
------------	----

I. 서론

1. 작품선정 배경 및 필요성



- 우리나라는 급속한 고령화로 인하여 2018년 기준 전체 인구 중 65세 이상 고령자의 비율은 14.3%를 기록했고, 고령 면허소지자 비율이 2016은 8%에서 2017년 8.8%, 2018년 9.4%로 매년 증가하고 있음
- 고령 운전자로 인한 교통사고 사망자 비율 또한 매년 증가세를 보이는데, 경찰 통계에 따르면 고령 운전자가 낸 교통 사망사고 비율은 지난 2016년 17.7%에서 20.3%, 2018년에는 22.3%를 기록했고, 고령자 교통사고 사망 비중도 계속해서 증가하고 있음



- 노화가 진행됨에 따라 여러 기능의 감소로 인하여 고령 운전자의 경우 반응시간, 인지능력 및 상황 발생 시 대처 능력이 비고령운전자에 비해 취약하고, 고령 운전자의 경우 사고 발생 시 치명적 사망사고와도 관련됨
- 사고 예방을 위해 고령 운전자가 실제 도로 상황에서 사고 상황을 훈련하기에는 안전상의 이유로 어려움이 있음
- 가상현실에서의 사고 상황 훈련 및 주행 훈련은 운전 관련 자극의 체계적인 제시와 실시간 반응 행동 데이터 수집이 가능함
- 가상현실에서의 상황에 따른 인지 반응 및 반응 속도를 가지고 위험군을 판별하는 운전 시뮬레이션 콘텐츠 필요

2. 기존 연구/기술동향 분석

2.1 한국 도로 교통공사 VR 교통사고 체험



- 도로교통공단(이사장 윤종기)은 서울, 부산, 경기(용인), 대전, 부산, 광주 5곳의 첨단교육센터에서 가상현실(VR)을 활용한 교통안전교육을 시행
- 교통안전 첨단체험교육 프로그램은 HMD(Head Mounted Display·가상현실 헤드셋)와 모션 시뮬레이터 등을 활용해 진행되는데, 이를 통해 VR로 교통안전 상황을 생생하게 체험할 수 있어 사고에 대한 경각심을 가질 수 있음
- 도로교통공단의 전문 교수진이 알기 쉽게 교통안전에 대한 설명을 병행하여 체험자의 교통안전의식 함양에 도움을 줌

2.2 운전면허 시험장 운전 시뮬레이터



- 도로주행연습시뮬레이터로는 1종 보통, 2종 보통(자동, 수동)으로 기능시험 및 도로주행시험이 가능하고, 주로 운전면허 취득을 위해 이용하는 사람이 많음
- 1종과 2종 모두 연습이 가능하게 만들어짐
- 클러치와 두 개의 기어 변속기가 있는 점, 사이드미러 조종 버튼이 터치인 것은 연습 시 주의해야 할 부분
- 실제 차에서 체감할 수 있는 부분들이 미흡하다는 점은 꼭 염두에 둬야 함
- 운동신경이 안 좋다면 차이가 더 크게 와 닿을 것으로 예상됨

3. 개발 목표

- 가상현실에서 예측할 수 없는 돌발 상황들을 통해 인지 반응 및 반응속도를 측정
- 평가 결과로 고령 운전자들의 위험군 판별이 가능한 운전 시뮬레이션 콘텐츠 개발

4. 팀 역할 분담

이 강 우	박 준 형	권 지 영
<ul style="list-style-type: none"> - UGUI 설계 및 구현 - 게임 전체 씬 흐름도 설계 및 구현 - MySQL을 이용한 데이터 처리 설계 및 구현 	<ul style="list-style-type: none"> - VR을 이용한 설계 및 구현 - VR을 이용한 콘텐츠 제작 - 결과 시각화 구현 	<ul style="list-style-type: none"> - TOBII를 이용한 설계 및 구현 - TOBII를 이용한 콘텐츠 제작 - 결과 시각화 구현

5. 개발 일정


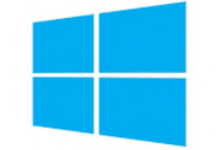

사 항	12월	1월	2월	3월	4월	5월	6월	7월	8월	9월	10월
사전 조사 및 계획서 발표											
평가척도 조사 및 콘텐츠 기획											
하드웨어와 유니티 통합											
Unity 콘텐츠 제작 및 DB 구축											
테스트 및 유지보수											
최종 보고서 작성 및 논문 작성											

6. 개발 환경



- PC 개발 환경

CPU	Intel Core™ i7-7700HQ CPU
Graphic	GeForce GTX 1070
RAM	16G RAM
O/S	Window10 x64bit



- 개발 환경

		
Unity3D	Windows 10	Visual Studio C#

- 개발 툴

		
VR SDK	Tobii Unity SDK	Logitech steering Wheel SDK

- 하드웨어

		
FOVE / VIVE VR	Tobii Eye Tracker 4C	Logitech steering Wheel

II. 본론

1. 개발 내용

- 고령 운전자들의 위험군 판별을 위한 운전 시뮬레이션 콘텐츠 개발
- 시뮬레이션 콘텐츠를 개발하기 위해 Unity3D를 사용하여 개발을 진행
- 실제 운전하는 경험을 주기 위해 VR과 Logitech steering Wheel을 사용
- 예측할 수 없는 돌발 상황들을 통해 위험군을 판별하여 운전자에게 위험군 판별
- 여부를 전달하는 콘텐츠를 개발

2. 문제 및 해결방안

2.1 운전자 경험의 문제

- 콘텐츠를 이용하는 운전자가 실제로 운전을 하는 것이 아니기 때문에 실제 운전과 다른 결과를 도출
- 실제 운전과 다른 결과가 나타나는 것은 위험군을 판별하기에 문제가 됨
- 이러한 문제를 해결하기 위해서 콘텐츠를 이용하는 운전자에게 실제 운전을 하는 것과 같은 경험을 주어야 함

2.1.1 해결 방안

- 콘텐츠를 이용하는 운전자에게 실제 운전을 하는 것과 같은 경험을 주기 위해 VR을 선택
- 실제 운전하는 경험을 주기 위해서 핸들과 엑셀 브레이크가 필요하며, 이러한 문제는 Logitech Steering Wheel을 사용하여 문제를 해결
- VR과 Logitech Steering Wheel을 이용하여 운전자 경험의 문제를 해결

2.2 사이버멀미의 문제

- VR 사용자에게 사이버멀미(Cyber sickness) 증상이 나타날 가능성 존재
- 사이버멀미 증상은 현기증, 어지러움, 두통, 구토감 등 기존의 멀미 증상과 유사함
- 사이버멀미 증상이 심한 사용자의 경우, 다른 장치를 이용한 시뮬레이션이 필요함

2.2.1 해결 방안

- 사이버멀미 증상이 심한 사용자를 고려하여, 추가로 Tobii Eye Tracker 장치 선택
- Tobii Eye Tracker를 사용한 3D 콘텐츠를 추가하여 사이버멀미 증상이 심한 사용자 문제 해결

2.3 평가 척도의 문제

- 위험군을 판별하기 위해서는 어떤 상황에서 반응속도 측정을 해야 위험군이 판별되는지 평가척도의 부재
- 이를 해결하기 위해서 논문들을 읽고 정리할 필요가 있다고 판단되어 고령 운전자에 대한 논문, 반응속도에 대한 논문들을 검색하여 문서화함

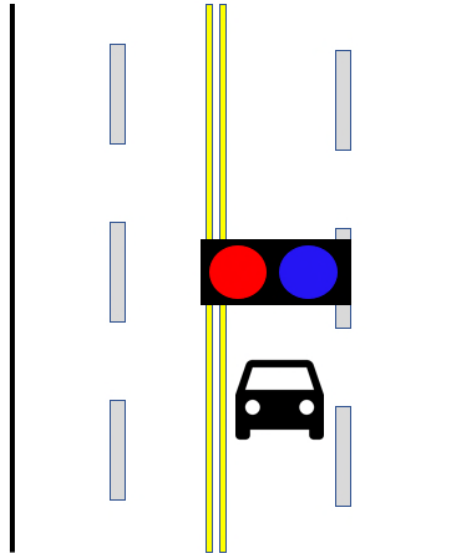
2.3.1 해결 방안

(단위:초, s)

시나리오 상황	위험군
신호등 변화	2.0 ~
앞 차량의 급정거	1.9 ~
다른 차량의 끼어들기	1.9 ~
교차로 내 충돌 위험	1.6 ~

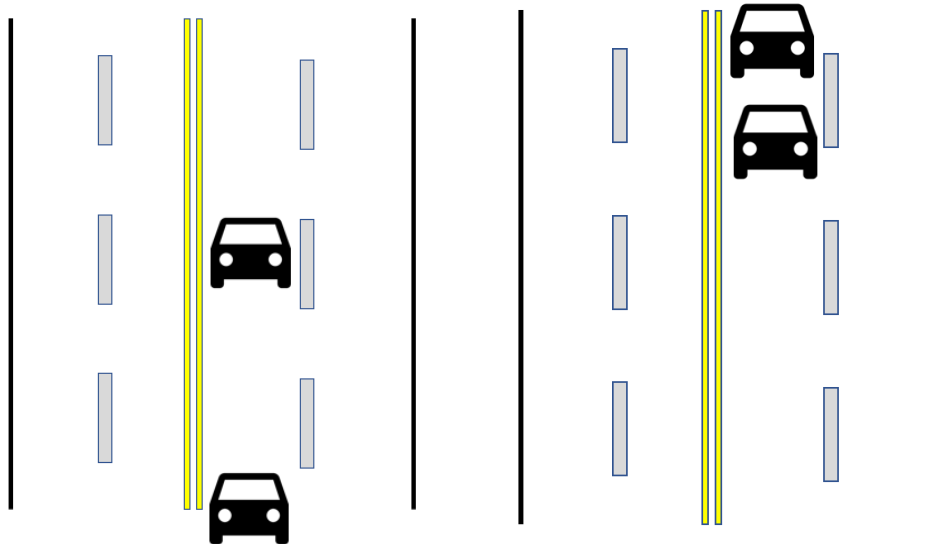
3. 시험 시나리오

3.1 시나리오



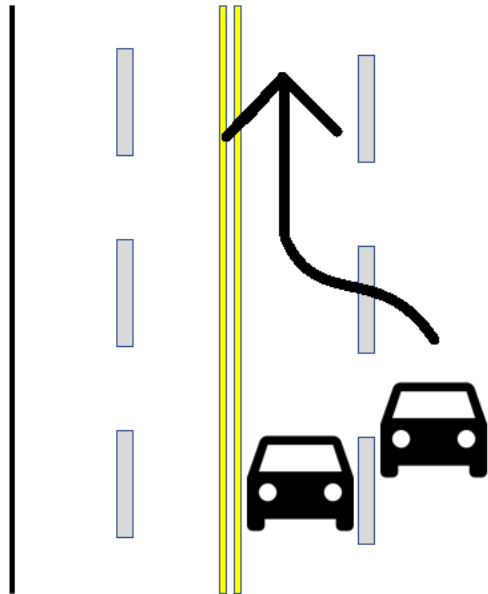
- 신호등 색의 전환에 대한 반응
- 파란색에서 빨간불 전환 시, 보는 데 걸리는 시간 측정, 브레이크 밟는 반응속도 측정

3.2 시나리오



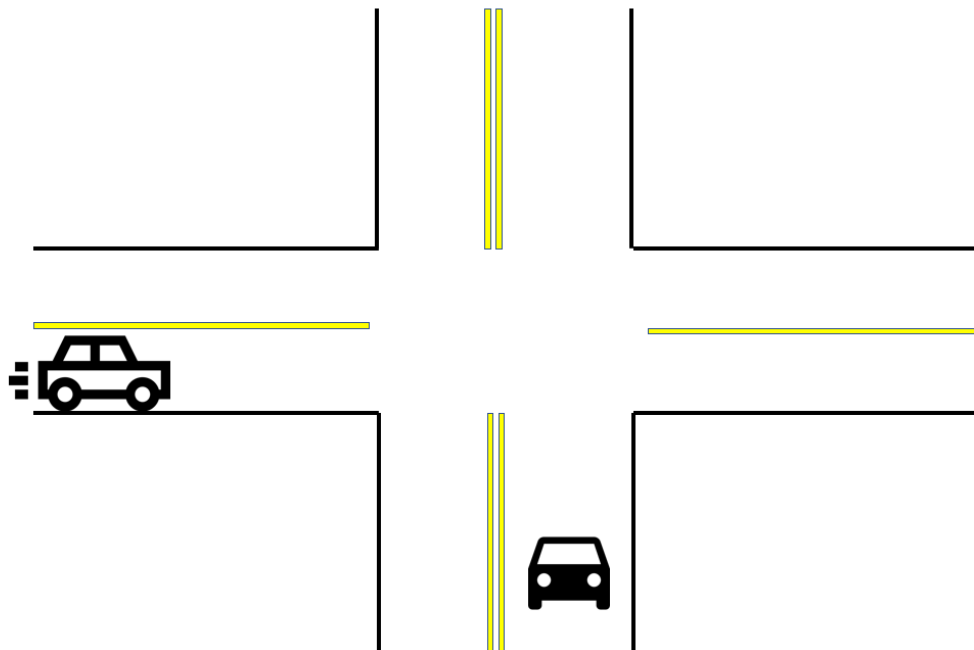
- 차량 주행 시, 앞 차량 급정거에 대한 반응
- 주행 중 앞 차량 급정거 시, 보는 데 걸리는 시간 측정, 브레이크 밟는 반응속도 측정

3.3 시나리오



- 도로 주행 시, 옆 차선에서 끼어드는 차량에 대한 반응
- 특정 지점에서 차량이 끼어들 시, 보는 데 걸리는 시간 측정, 브레이크 밟는 반응 속도 측정

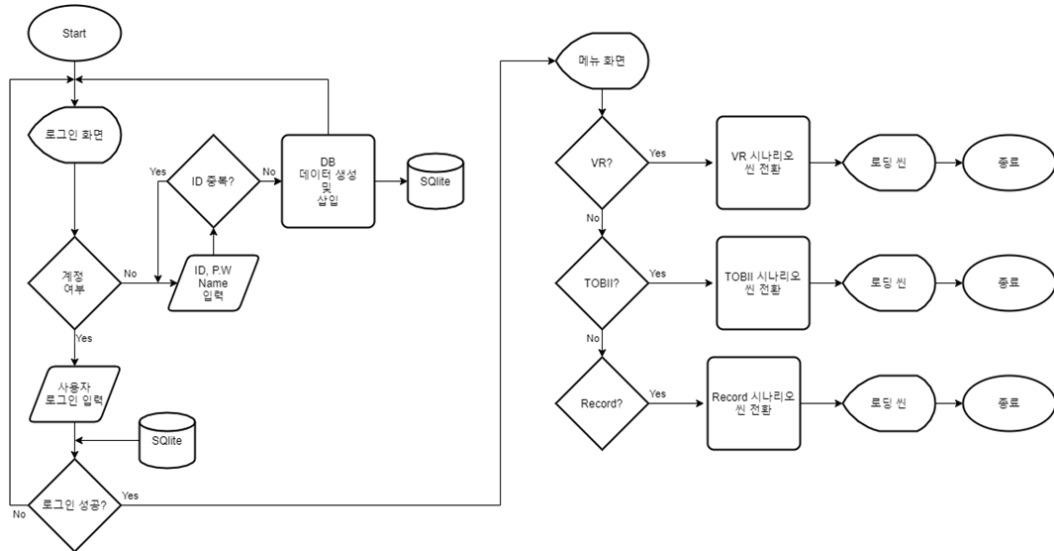
3.4 시나리오



- 교차로 주행 시, 우측도로에서 진입하는 차량에 대한 반응
- 골목길 교차로 우측도로에서 차량 진입 시, 보는 데 걸리는 시간 측정, 브레이크 밟는 반응속도 측정

4. 상세 설계

4.1 플로우 차트 (로그인)



로그인 시 데이터베이스에 접근하여 회원 정보 취득.

정보가 없을 시 로그인 실패.

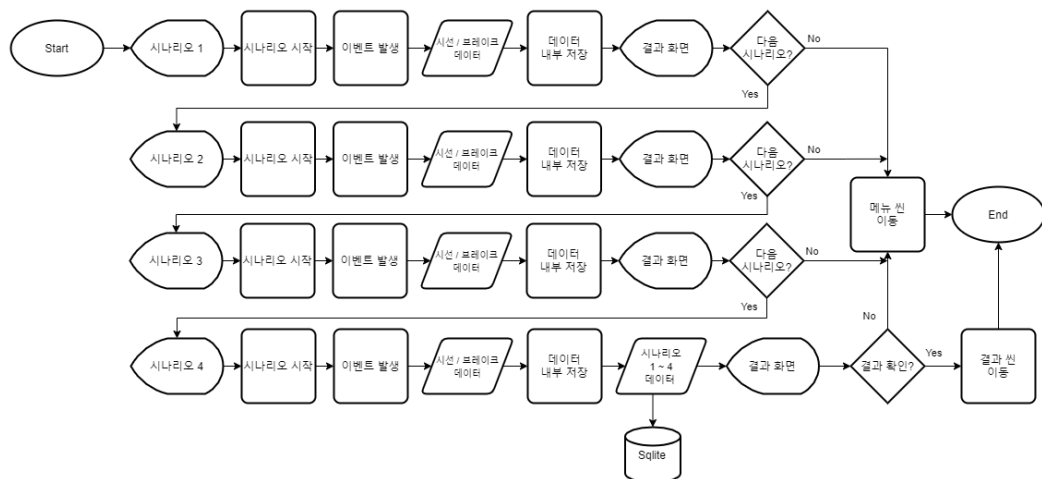
계정 없을 시 회원 가입을 통해 데이터베이스에 데이터 저장 (ID가 기본키).

로그인 성공 시 메인 메뉴로 이동.

유저의 선택에 맞추어 이벤트 발생

4.2 플로우 차트 (시나리오)

VR / TOBII 시나리오



유저가 시나리오 실행 시 이벤트 발생으로 통해 시선/ 브레이크 반응 속도를 측정.

스크립트를 통해 측정 데이터를 시나리오 별로 구분하여 내부에 저장.

각 시나리오 종료 화면을 통해 다음 시나리오 진행 여부 확인.

모든 시나리오 종료 시 결과 화면 출력.

4.3 소프트웨어 설계 (로그인 화면)

The image shows a login screen with a dark background. It features two input fields: one for '아이디' (ID) with a placeholder 'ID 입력 (12글자 이내)' and one for '비밀번호' (Password) with a placeholder '비밀번호 입력 (12글자 이내)'. Below the fields is a prompt '입력해주세요.' (Please enter). There are two blue buttons: '로그인' (Login) and '회원가입' (Sign Up).

login_btn_Manager모듈	
클래스	
기능	
1.UGUI InputField를 이용한 사용자 입력	
2.ID / PW 입력 후 DB 조회	
3.DB 내 데이터 유무 확인	
상세기능	
<ul style="list-style-type: none"> • DB를 통해 ID / PW를 확인하고 • ID /PW 데이터 없을 시, TEXT로 없는 데이터 표시 • 데이터 있을 시, 메뉴로 이동 	

Check_have_ID(string ID, string PW)	
형식	public string Check_have_ID(string ID, string PW)
리턴 값	string
설명	<ul style="list-style-type: none"> • ID 와 PW를 받고, DB에 ID 와 PW가 있는지 확인 • ID 와 PW가 있으면, string.Empty반환 • ID 가 없으면, “아이디가 존재하지 않습니다.” 반환 • PW 가 없으면, “비밀번호가 잘못되었습니다.” 반환

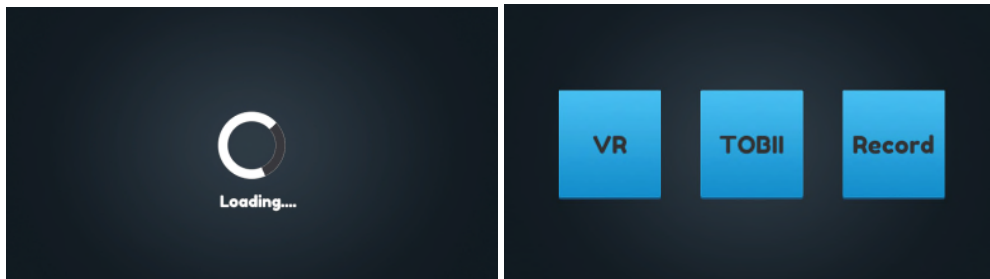
check_login()	
형식	public void check_login()
리턴 값	void
설명	<ul style="list-style-type: none"> • “로그인” UGUI Button 클릭 시, 활성화 • UGUI ID / PW 입력 데이터가 있는지 확인, 없으면 throw new NullReferenceException() 전달 후, 데이터 입력 요청 • 입력 데이터를 Check_have_ID 함수에 전달, 결과 값에 따라 처리

4.4 소프트웨어 설계 (계정 생성)

CreateAccount_Manager모듈	
클래스	
기능	<ol style="list-style-type: none"> 1.UGUI InputField를 이용한 사용자 입력 2.“가입하기” 버튼 클릭 3.입력된 ID/PW/ 이름 DB에 삽입 4.버튼 클릭 후 로그인 화면으로 전환
상세기능	<ul style="list-style-type: none"> • ID / PW / 이름 을 입력받고, “가입하기”버튼 클릭 시, 이를 DB에 삽입 • DB 삽입 후 로그인 화면으로 전환

Btn_Add_Account()	
형식	public void Btn_Add_Account()
리턴 값	void
설명	<ul style="list-style-type: none"> • “가입하기” UGUI Button 클릭 시, 활성화 • UGUI ID / PW / 이름입력 데이터가 있는지 확인, 없으면 throw new NullPointerException() 전달 후, 데이터 입력 요청 • ID가 DB 내 기본키이기 때문에 중복 검사. 중복 시, 중복됨을 알리고 return • query문을 작성, DB에 데이터를 삽입

4.5 소프트웨어 설계 (메뉴)

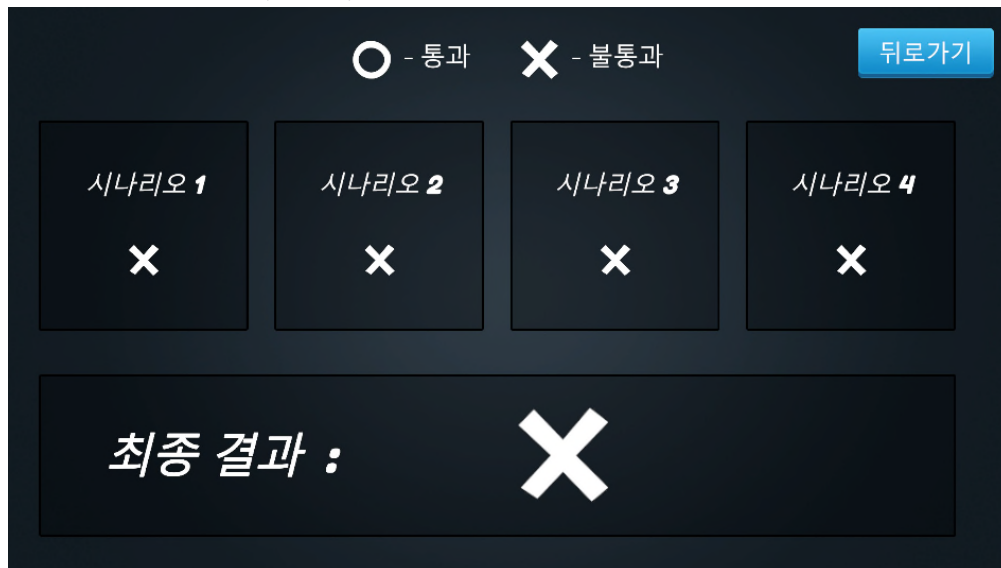


MainScene_Manager 모듈	
클래스	
기능	
1.메뉴 씬을 활성화 2.UGUI로 구성된 버튼메뉴 활성화 3.Async 을 이용 로딩을 구현 4.로딩 완료 시, 선택 씬으로 이동	
상세기능	
<ul style="list-style-type: none"> • 버튼에 해당하는 씬으로 이동을 구현 • 버튼 클릭 시, Async을 이용한 씬 전환 기법을 • 사용해 다음 씬 로딩을 구현 • 로딩 진행 정도를 ProgressBar을 통해 시각화 	

GO_Scene()	
형식	public void GO_Scene(string btn_name)
리턴 값	void
설명	<ul style="list-style-type: none"> • UGUI 메뉴 버튼을 클릭시, 버튼 이름에 받는 이름을 변수로 받음 • 이름에 맞는 씬으로Async을 이용한 로딩을 활성화 • StartCoroutine>Loading_SceneChanged(async)) 이용 로딩 활성화

loading_SceneChanged(AsyncOperation async)	
형식	IEnumerator loading_SceneChanged(AsyncOperation async)
리턴 값	yield
설명	<ul style="list-style-type: none"> • GO_Scene을 통해 받은 async을 통해 해당 씬을 로딩 • 로딩 진행도를 ProgressBar을 통해 시각화 • Mathf.Lerp(선형보간법) 이용해 자연스러운 ProgressBar을 생성 • 로딩이 끝날 시, 해당 씬으로 이동

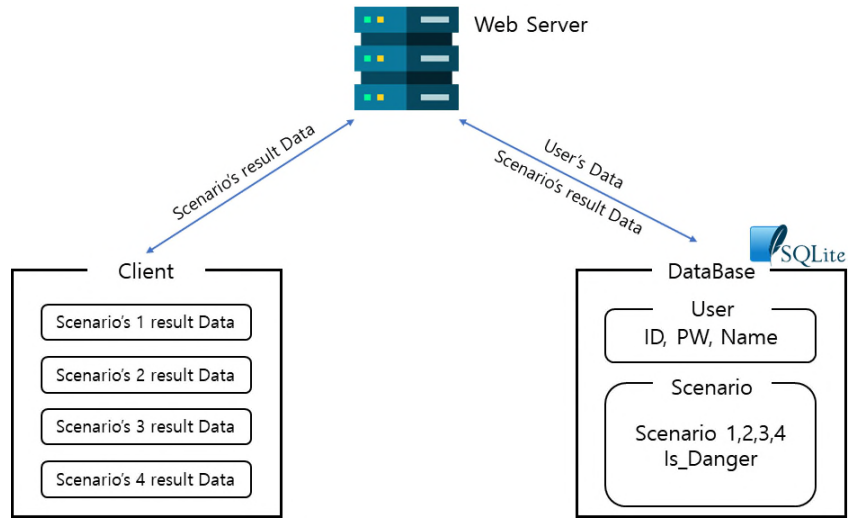
4.6 소프트웨어 설계 (레코드)



Record_btn_Manager 모듈	
클래스	
기능	
<ol style="list-style-type: none"> 1. 레코드 썸 활성화 2. 로그인한 유저의 데이터 접근 3. 접근한 데이터 시각화 	
상세기능	
<ul style="list-style-type: none"> • Recrod 버튼 클릭 시, 레코드 썸 활성화 • 유저의 각 시나리오 결과 값과 최종 결과 값을 시각화 • O : 통과 X : 불통과 	

Show_result()	
형식	public void Show_result()
리턴 값	void
설명	<ul style="list-style-type: none"> • user 클래스 객체 리스트를 참조 • 리스트 내 로그인한 유저를 검색 • 검색된 유저의 결과 데이터 시각화

4.7 DB 설계 (Sqlite)



- 클라이언트를 통해 각 시나리오의 결과 데이터를 서버에 전송
- 서버를 통해 Database에 접속, 해당 클라이언트의 시나리오 결과 값에 데이터 삽입
- 학교 서버가 외부 서버를 사용한 데이터 관리가 불가능하다는 것을 인지
서버를 이용하지 않고, 프로그램 자체 로컬 DB를 사용하여 Database 구현
- 프로그램 자체 내부에 DB 활성화

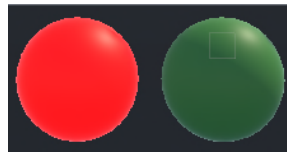
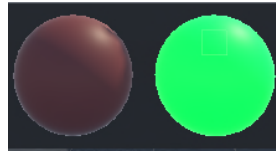
	ID	Password	Name	Scenario1	Scenario2	Scenario3	Scenario4	Is_Danger
필터	필터	필터	필터	필터	필터	필터	필터	필터
1	2015152025	1q2w3e4r	LeeKangWoo	0	0	0	0	0
2	2017152015	Park	ParkJoonHy...	0	0	0	0	0
3	2017150005	Kwon	KwonJiYoung	0	0	0	0	0

- ID, Password, Name 은 회원 가입 시, DB를 통해 데이터 삽입
- 각 Scenario 결과 및 최종 결과인 Is_Danger는 시나리오 종료 시 데이터 삽입

4.7 DB 설계 (Sqlite)

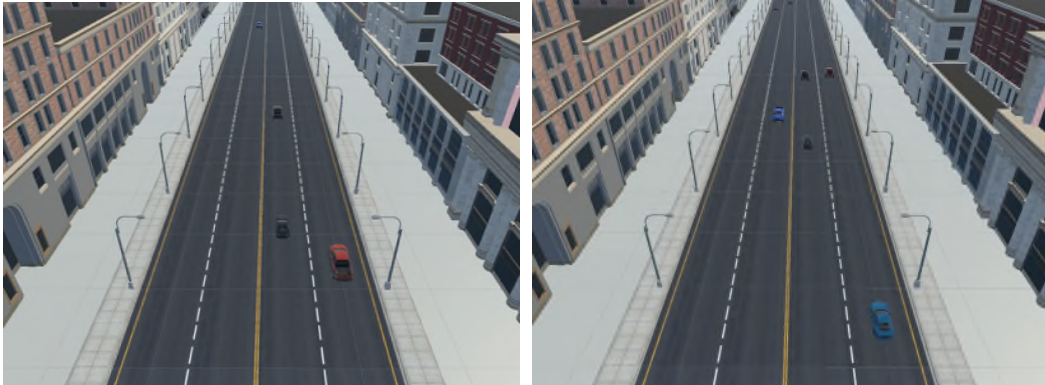
DB_sqlite_Manager 모듈	
클래스	
기능	
1. Database 연결 2. Database 읽기 3. Database 쿼리문 작성	
상세기능	
<ul style="list-style-type: none"> • Database 에 접근해야 하는 이벤트 발생 시 호출 • 각 상황에 맞춰서 쿼리문을 작성 • 쿼리문을 토대로 Database 접근 및 작업 • 작업 결과를 반환 	
DB_Read(string query)	
형식	private void DB_Read(string query)
리턴 값	void
설명	<ul style="list-style-type: none"> • 매개변수를 통해 쿼리문을 받음 • 쿼리문을 통해 데이터 읽기 • 데이터가 올바르게 들어갔는지 디버깅 가능 • 결과 조회 시, 쿼리문을 통해 데이터 검색 및 시각화
DB_Query(string query)	
형식	public void DB_Query(string query)
리턴 값	void
설명	<ul style="list-style-type: none"> • 매개변수를 통해 쿼리문을 받음 • 쿼리문을 통해 데이터 삽입 • 회원 가입 시, User's Data 삽입 • 시나리오 종료 시, 각 Scenario's Data 삽입

4.8 이벤트 설계 (시나리오 1 신호등 색의 전환)



Traffic_Light_Manager 모듈	
클래스	
기능	
1. 타이머 작동 2. 신호등의 색 전환	
상세기능	
<ul style="list-style-type: none"> • 게임 시작 시 타이머 작동 • 타이머 작동 후 7~10초 사이에 초록색 불이 빨간색으로 전환 	
Change_Traffic_Light()	
형식	public void Change_Traffic_Light()
리턴 값	void
설명	<ul style="list-style-type: none"> • Time.deltaTime을 이용해 게임 시작과 동시에 타이머를 활성화 • 7~10초의 랜덤 시간 후 신호등의 색 전환 • 초록색 -> 노란색 -> 빨간색

4.9 이벤트 설계 (시나리오 2)



AI_NavigationControl 모듈	
클래스	
기능	
<ol style="list-style-type: none"> 1. 목표지점을 설정 2. AI가 최단거리를 발견 3. 자율주행 진행 	
상세기능	
<ul style="list-style-type: none"> • List<>를 이용하여 각 차량들에게 목표지점을 설정 • SetDestination을 이용하여 목표지점까지의 최단거리를 오브젝트가 스스로 발견 • 오브젝트가 목표지점을 향해 이동 • 목표지점에 도달하면 오브젝트 삭제 	

MoveWayPoint()	
형식	public void MoveWayPoint()
리턴 값	void
설명	<ul style="list-style-type: none"> • List<>에 목표지점을 설정 후 NavMeshAgent 컴포넌트를 사용 • SetDestination을 이용하여 최단거리를 AI가 발견 후 이동 • 최종 목적지에 도착 후 자동차 소멸

CarStop()	
형식	public void CarStop()
리턴 값	void
설명	<ul style="list-style-type: none"> • 운전자의 차량이 Trigger와 충돌 • 앞차량의 움직임은 Vector3.zero를 이용하여 제자리에 급정거 활성화

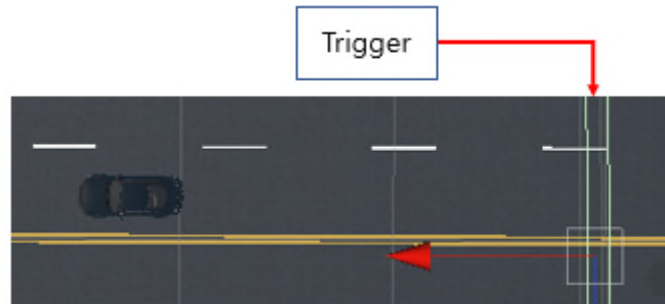
4.10 이벤트 설계 (시나리오 2 앞차량의 급정거)



Sudden_Stop_Car_Manager 모듈	
클래스	
기능	<ol style="list-style-type: none"> 1. 운전자의 차량이 지정 위치에 도달 2. 앞차량의 급정거로 후미등 점등
상세기능	<ul style="list-style-type: none"> • 운전자의 차량이 주행 중 지정 위치의 Trigger와 충돌 • 앞차량의 GameObject 후미등을 On으로 활성화

Light_On()	
형식	public void Light_On()
리턴 값	void
설명	<ul style="list-style-type: none"> • 운전자 차량의 Tag를 "Player"로 설정 • 운전자의 차량이 Trigger와 충돌 • 앞 차량의 후미등 점등 활성화

4.11 이벤트 설계 (시나리오 랜덤 이벤트 발생)



Random_Trigger 모듈	
클래스	
기능	<ol style="list-style-type: none"> 1. Bool 함수를 이용하여 X, Y, Z 축의 Random 지정하고 싶은 축을 지정 2. 정해진 축으로 Random 위치에 Trigger 생성
상세기능	<ul style="list-style-type: none"> • 사용자가 이벤트 발생에 익숙해지는 현상을 방지하기 위해 Trigger의 위치를 Random으로 생성

Random_Trigger()	
형식	public void Random_Trigger()
리턴 값	void
설명	<ul style="list-style-type: none"> • Bool 함수를 이용하여 X, Y, Z 축의 Random 지정하고 싶은 축을 지정 • 정해진 축으로 Random 위치에 Trigger 생성

4.12 이벤트 설계 (시나리오 3 차량 끼어들기)



Interfering_Car 모듈	
클래스	
기능	
1. 끼어들기 차량 - 운전 차량 사이의 거리가 특정 정도에 도달 2. 차량 끼어들기 이벤트 구현	
상세기능	
<ul style="list-style-type: none"> • 끼어들기 차량에 달려있는 Trigger와 운전 차량이 충돌 • 운전 차량 앞에 있는 destination으로 끼어들기 차량 이동 • 끼어들기 차량이 destination에 도착한 이후, 속도 감소 	

OnTriggerEnter(Collider other)	
형식	public void OnTriggerEnter(Collider other)
리턴 값	void
설명	<ul style="list-style-type: none"> • 운전자 차량의 Tag를 "Player"로 설정 • 운전자의 차량이 Trigger와 충돌 • 이벤트 조건 활성화

Interfering()	
형식	public void Interfering()
리턴 값	void
설명	<ul style="list-style-type: none"> • 끼어들기 차량과 운전 차량의 거리가 특정 정도에 도달할 시 IsArrive = true; • 이벤트 조건 활성화 및 IsArrive가 true일 때, 운전 차량 앞에 설정되어 있는 Destination으로 끼어들기 차량 이동 및 속도 감소

4.13 이벤트 설계 (시나리오 4 교차로로 주행)

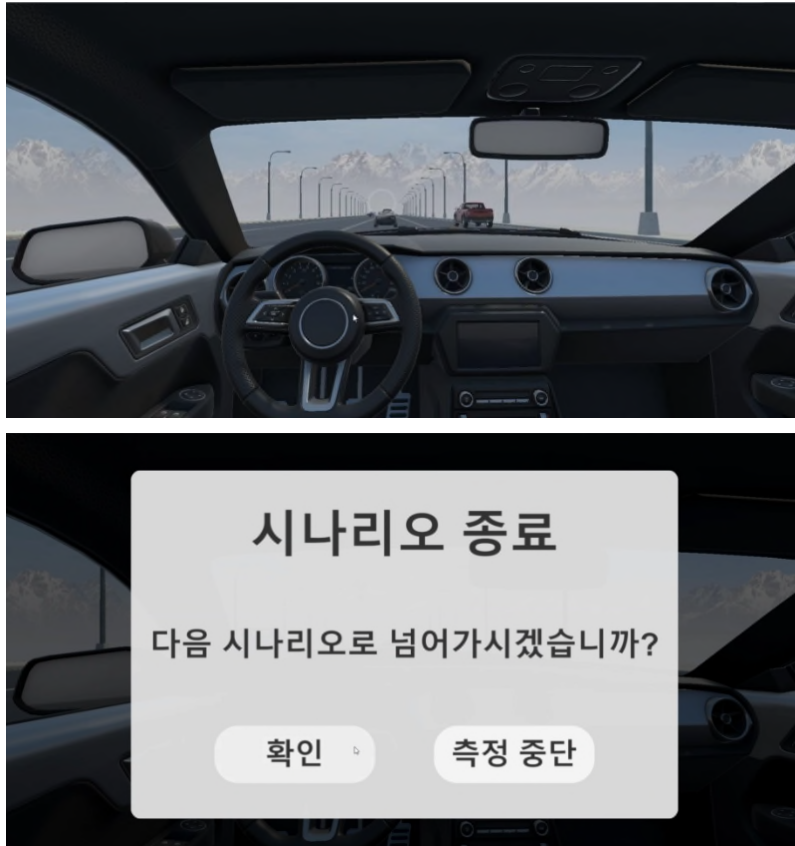


SuddenCar 모듈	
클래스	
기능	
<ol style="list-style-type: none"> 1. 운전자의 차량이 지정 위치에 도달 2. 교차로에서 지나가는 차량 구현 	
상세기능	
<ul style="list-style-type: none"> • 운전자의 차량이 주행 중 지정 위치의 Trigger와 충돌 • 정차되어 있던 차량이 주행을 시작 • 운전자의 앞을 지나가는 차량 출현 	

OnTriggerEnter(Collider other)	
형식	public void OnTriggerEnter(Collider other)
리턴 값	void
설명	<ul style="list-style-type: none"> • 운전자 차량의 Tag를 “Player”로 설정 • 운전자의 차량이 Trigger와 충돌 • StartCoroutine(Go()) 이용 다른 차량이동 활성화

Go()	
형식	IEnumerator Go()
리턴 값	yield
설명	<ul style="list-style-type: none"> • Vector3.MoveTowards을 이용해 외부 차량의 자연스러운 직진 구현 • Target position의 위치와 외부 차량의 위치가 동일 시 될 때까지 동작

4.14 이벤트 설계 (Tobii 시선 추적)



GazeEvent 모듈	
클래스	
기능	
	<ol style="list-style-type: none"> 1. 이벤트가 발생하면 시간 측정 시작 2. 시선 및 브레이크 반응 시 각 반응 시간 기록
상세기능	
	<ul style="list-style-type: none"> • 이벤트가 발생하면 타이머 측정을 시작 • 사용자의 시선이 특정 물체에 닿으면 시선 반응 시간을 기록하고, • 그 후 브레이크를 밟으면 브레이크 반응 시간 기록 • 각 반응 시간이 모두 기록되면 시나리오 종료 및 결과 화면 출력

4.15 이벤트 설계 (Tobii 시선 추적)

OnTriggerEnter(Collider other)	
형식	public void OnTriggerEnter(Collider other)
리턴 값	void
설명	<ul style="list-style-type: none"> • 이벤트 발생 시 시선과 특정 물체 충돌 확인 • 충돌 발생 시, EyesTime 변수에 시선 반응 시간 기록

Check_Scenario(int n)	
형식	public void Check_Scenario(int n)
리턴 값	void
설명	<ul style="list-style-type: none"> • 시선 반응 여부 + 이벤트 발생 여부 확인 후 만족 시 실행 • 브레이크 반응 시, BrakeTime 변수에 브레이크 반응 시간 기록 • Manager 스크립트에 각 시나리오, 시선 반응 시간, 브레이크 반응 시간 전송

4.16 이벤트 설계 (Logitech Steering Wheel)



Logitech_Manager 모듈	
클래스	
기능	
<ol style="list-style-type: none"> 1. Accel을 통한 전진 2. Break를 통한 정지 3. Handle을 통한 방향전환 	
상세기능	
<ul style="list-style-type: none"> • Accel의 밝기 세기에 따라 Rpm계기판의 계수가 증가, 감소 • 현재 속도에 따른 Speed계기판의 계수 증가, 감소 • Break의 밝기 세기에 따라 감속의 차이 구현 • Handle의 회전에 따른 자동차 바퀴 방향전환 	

GetInput()	
형식	public void GetInput()
리턴 값	void
설명	<ul style="list-style-type: none"> • Accel, Break, Handle의 입력 값을 받아 사용하기 편한 정수로 변환

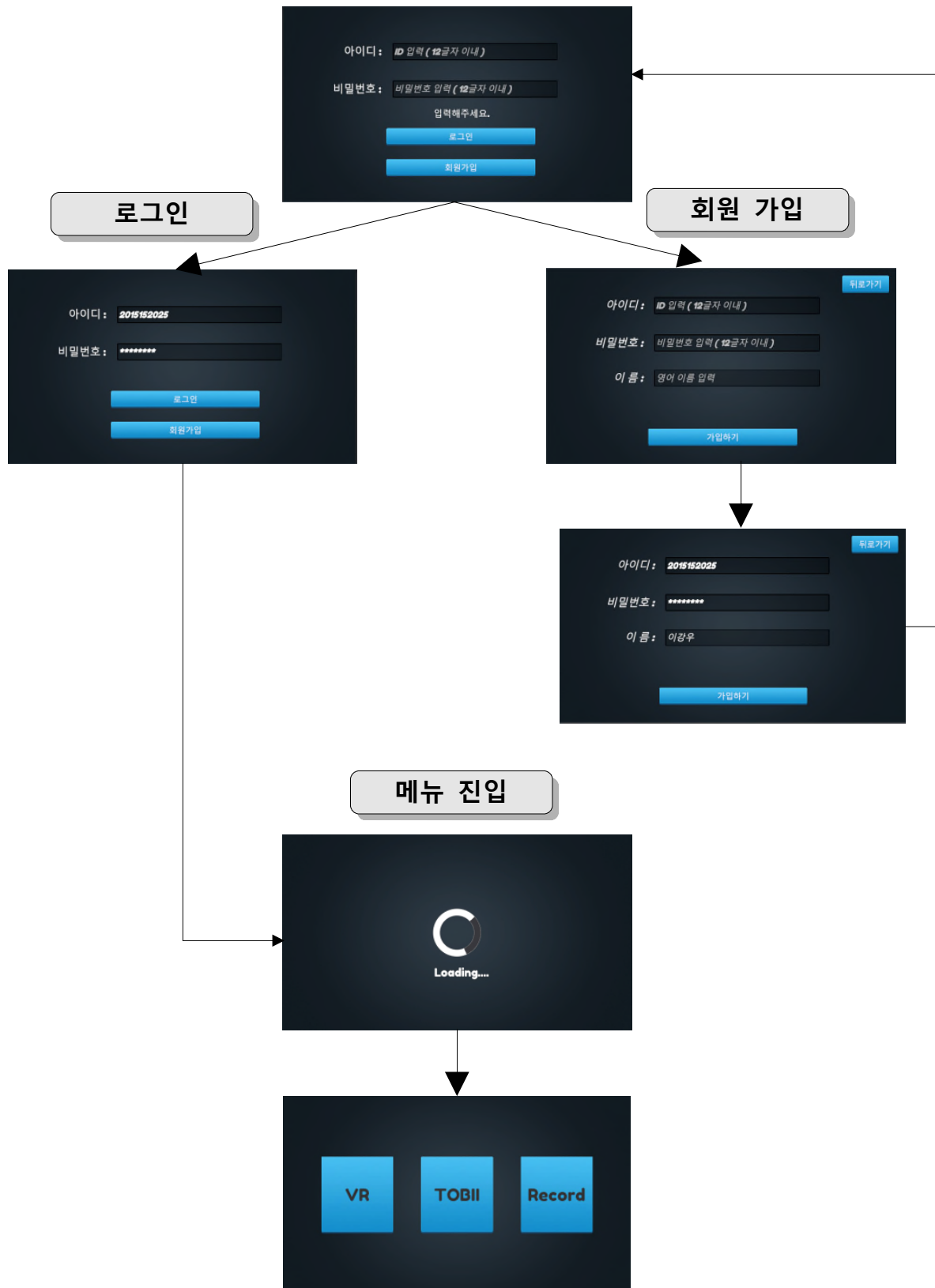
Steer()	
형식	public void Steer()
리턴 값	void
설명	<ul style="list-style-type: none"> • Handle 회전 값에 따라 게임에서 육안으로 보이는 Handle의 회전과 바퀴의 방향을 회전

4.17 이벤트 설계 (시나리오 4)

Accel()	
형식	public void Accel()
리턴 값	void
설명	<ul style="list-style-type: none"> • Accel 밟기의 세기에 따라 모터의 회전 값, Rpm에 변화를 주어 가속에 변화

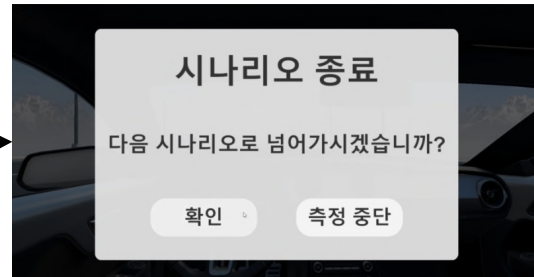
Break()	
형식	public void Break()
리턴 값	void
설명	<ul style="list-style-type: none"> • Break 밟기의 세기에 따라 정지에 부여하는 힘 증가

5. Prototype 구현
5.1 로그인 및 메뉴

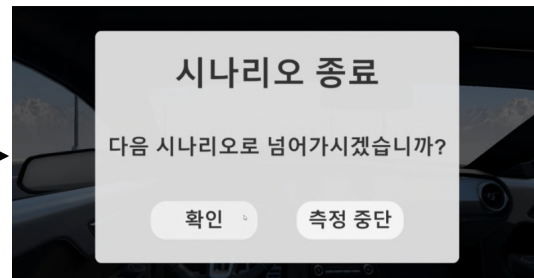


5.2 시나리오 진행 및 결과화면 확인

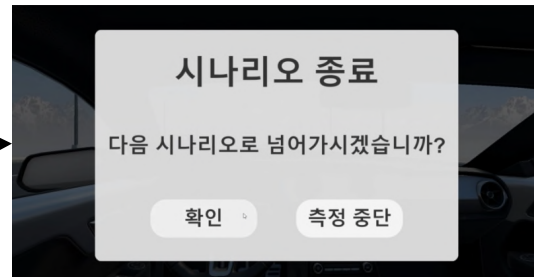
시나리오 1



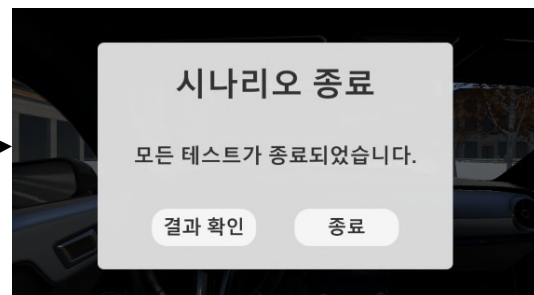
시나리오 2



시나리오 3



시나리오 4



5.3 결과(Record) 확인

메뉴 씬에서 선택 / 시나리오4 수행 후 결과화면에서 선택

결과 확인



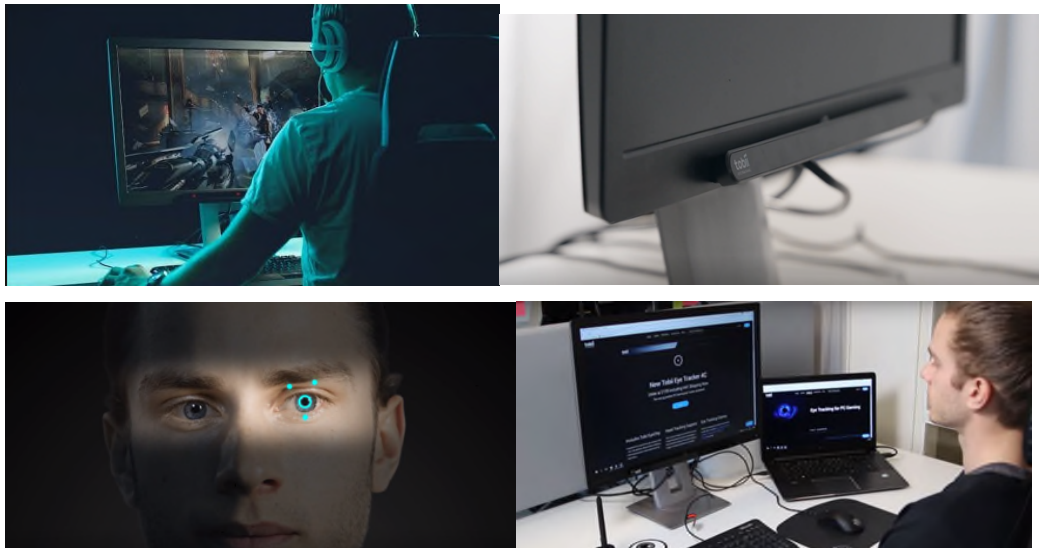
5.4 하드웨어 (VR/TOBII)

VR - Vive Pro Eye



- 기존 FOVE 기기에서 이상 현상 발생
- 기존 기기로는 프로젝트를 진행하기에 한계가 있어 Vive Pro Eye로 변경하였음
- Unity에 연동하여 시선 추적 기능 사용

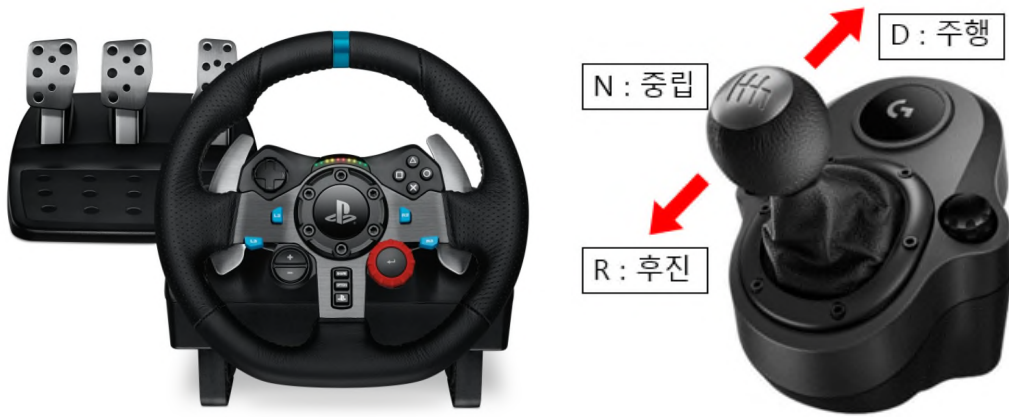
TOBII - Tobii Eye Tracking 4C



- 모니터 하단에 Tobii 장치 부착
- Unity에 연동하여 시선 추적 기능 사용

5.5 하드웨어 (LOGITECH)

LOGITECH - LOGITECH Driving Force Racing Wheel G29



- Unity에 연동하여 각 운전 기능 수행
- 휠 : 조향 값을 통해 Unity 내부에서 차량 조향기능을 구현
- 바닥 페달 : 가속 페달, 브레이크 페달 값을 통해 Unity 내부에서 가감속 기능 구현
- 시프터 : 시프터 값을 통해 Unity 내부 D: 주행, N: 중립, R: 후진 기능 구현

6. 시험 / 테스트 결과

6.1 로그인

기존 데이터베이스 정보

	ID	Password	Name	Scenario1	Scenario2	Scenario3	Scenario4
	필터	필터	필터	필터	필터	필터	필터
1	2015152025	1q2w3e4r	LeeKangWoo	1	1	1	1
2	2017152015	Park	ParkJoonHy...	0	0	0	0
3	2017150005	Kwon	KwonJiYoung	1	0	0	0

ID/PW 입력 오류(4가지) 테스트

아이디 :
비밀번호 :

아이디가 존재하지 않습니다.

아이디 :
비밀번호 :

비밀번호가 잘못되었습니다.

<아이디가 존재하지 않을 경우>

<비밀번호가 존재하지 않을 경우>

아이디 :
비밀번호 :

아이디를 입력해 주세요.

아이디 :
비밀번호 :

비밀번호를 입력해 주세요.

<아이디 입력이 되지 않았을 경우>

<비밀번호 입력이 되지 않았을 경우>

로그인 테스트

Save_user_data (Script)
Script
User_id

- 현재 로그인된 계정 정보

6.2 회원가입

기존 데이터베이스 정보

	ID	Password	Name	Scenario1	Scenario2	Scenario3	Scenario4
	필터	필터	필터	필터	필터	필터	필터
1	2015152025	1q2w3e4r	LeeKangWoo	1	1	1	1
2	2017152015	Park	ParkJoonHy...	0	0	0	0
3	2017150005	Kwon	KwonJiYoung	1	0	0	0

회원가입 테스트

아이디: 2017150005

비밀번호: ****

이름: asdf

아이디가 중복됩니다.

가입하기

- 중복되는 ID가 존재할 경우

아이디: abc123

비밀번호: ****

이름: Test

가입하기

- 회원가입 진행

	ID	Password	Name	Scenario1	Scenario2	Scenario3	Scenario4
	필터	필터	필터	필터	필터	필터	필터
1	2015152025	1q2w3e4r	LeeKangWoo	1	1	1	1
2	2017152015	Park	ParkJoonHy...	0	0	0	0
3	2017150005	Kwon	KwonJiYoung	1	0	0	0
4	abc123	test123	Test	0	0	0	0

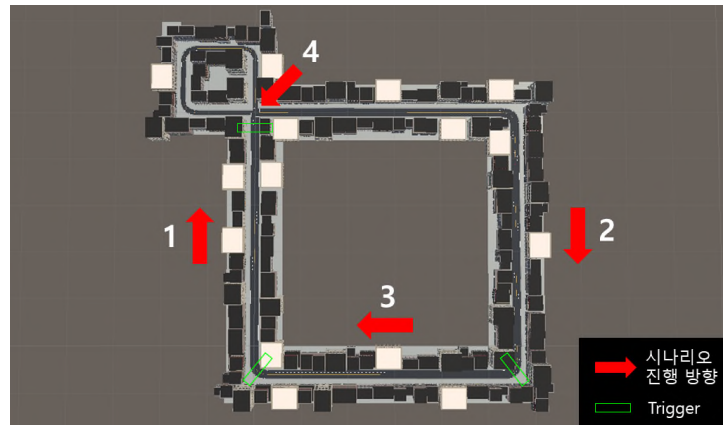
- DB에 계정 정보가 추가된 것을 확인할 수 있음

6.3 3차 수행보고서 평가

- 시나리오의 Play 길이 확대
- 시나리오의 조합을 만들어 볼 것

프로젝트 변경 사항

- 시나리오 1,2,3,4를 한 Scene에서 플레이 할 수 있도록 변경하였음
- 플레이 순서 : 2 - 3 - 1 - 4
- 각 시나리오가 시작되는 커브에 Trigger 설치
- 변경된 맵



시나리오 1 변경 사항

- 신호등이 이벤트가 시작되면 한 번만 빨간불로 변환된 후 지속적으로 파란불

시나리오 3 변경 사항

- 급정거 시나리오 후 자차가 2차선에서 주행할 가능성이 존재
- 1차선으로 주행 차선을 변경시키기 위해 오른쪽 차선에 불법 주차 차량 추가



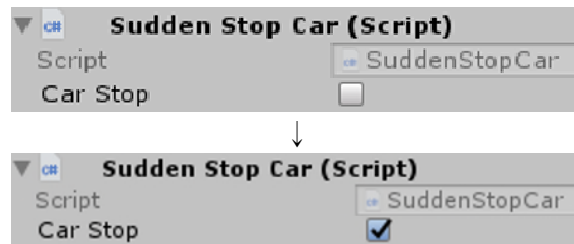
6.4 시나리오 진행 및 결과화면 확인 (VR)

- 이벤트 함수들 변화 과정(시나리오 실행 전, 후)

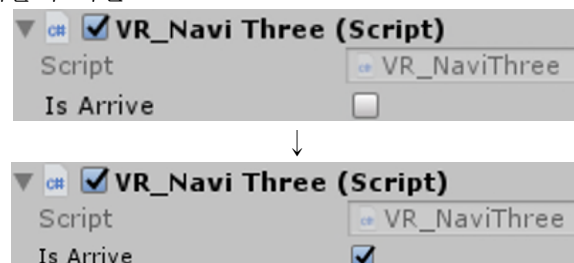
- 시나리오 1 신호등 이벤트



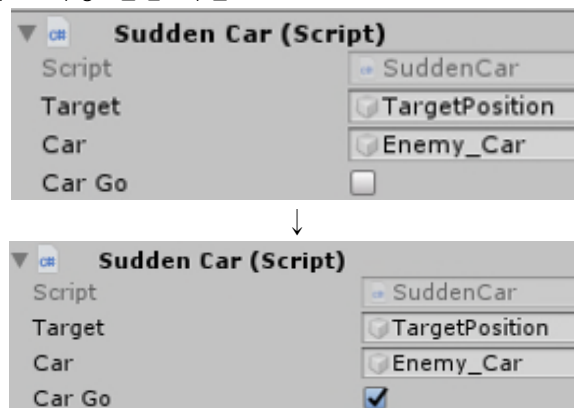
- 시나리오 2 급정지 이벤트




- 시나리오 3 끼어들기 이벤트



- 시나리오 4 교차로 차량 진입 이벤트



6.5 시나리오 진행 후 시나리오마다 데이터 내부 List 저장 확인 (VR)

▼  VR_Manager (Script)	
Script	VR_Manager
▼ L_VR	
Size	4
▼ VR_Scenario2	
Stage_name	VR_Scenario2
Eyes_time	0.2788354
Brake_time	0.4517677
▼ VR_Scenario3	
Stage_name	VR_Scenario3
Eyes_time	0.0135119
Brake_time	0.0298635
▼ VR_Scenario1	
Stage_name	VR_Scenario1
Eyes_time	0.012423
Brake_time	0.047826
▼ VR_Scenario4	
Stage_name	VR_Scenario4
Eyes_time	0.0193486
Brake_time	0.0326684

6.6 시나리오 진행 (VR)

- VR 급정거 이벤트



- VR 차량 끼어들기 이벤트



6.7 시나리오 진행 (VR)

- VR 신호등 이벤트



- VR 교차로 차량 출현 이벤트



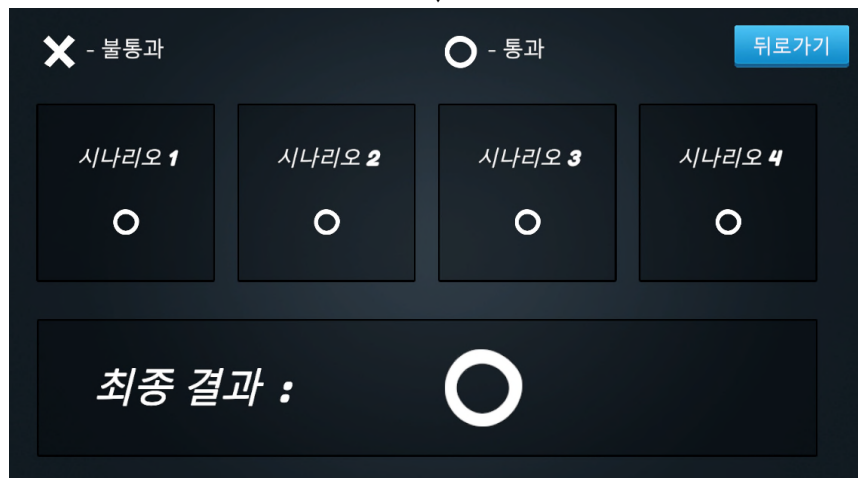
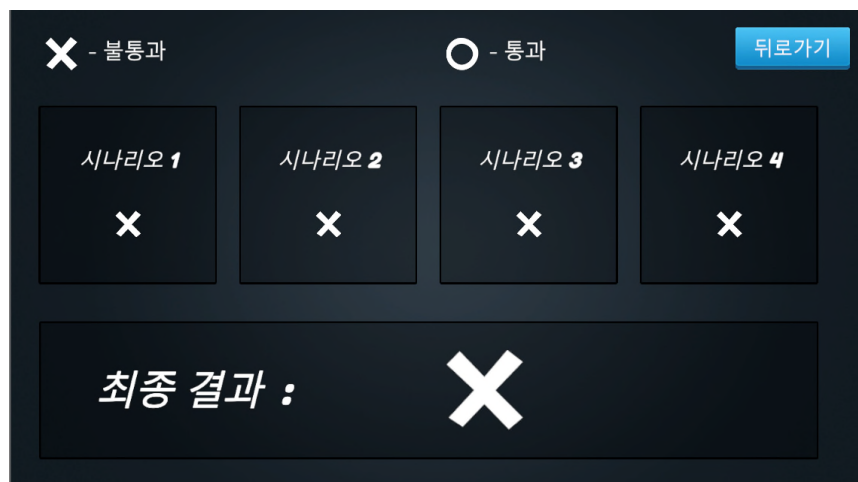
6.8 DB 결과(Record) 확인 (VR)

	ID	Password	Name	Scenario1	Scenario2	Scenario3	Scenario4	Is_Danger
	필터	필터	필터	필터	필터	필터	필터	필터
1	2015152025	1q2w3e4r	LeeKangWoo	1	1	1	1	1
2	2017152015	Park	ParkJoonHy...	0	0	0	0	0
3	2017150005	Kwon	KwonJiYoung	0	1	1	0	0



	ID	Password	Name	Scenario1	Scenario2	Scenario3	Scenario4	Is_Danger
	필터	필터	필터	필터	필터	필터	필터	필터
1	2015152025	1q2w3e4r	LeeKangWoo	1	1	1	1	1
2	2017152015	Park	ParkJoonHy...	1	1	1	1	1
3	2017150005	Kwon	KwonJiYoung	0	1	1	0	0

6.9 결과화면(Record) 확인 (VR)



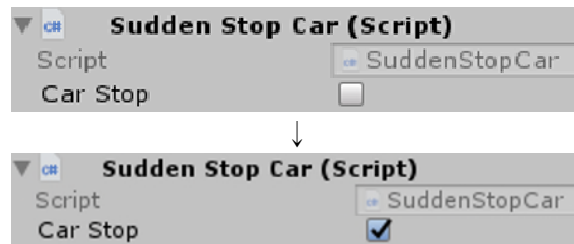
6.10 시나리오 진행 및 결과화면 확인 (Tobii)

- 이벤트 함수들 변화 과정(시나리오 실행 전, 후)

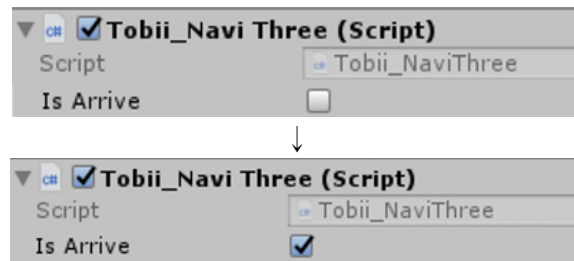
- 시나리오 1 신호등 이벤트



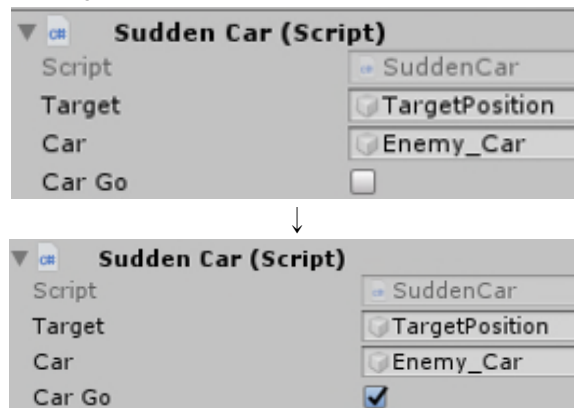
- 시나리오 2 급정지 이벤트



- 시나리오 3 끼어들기 이벤트



- 시나리오 4 교차로 차량 진입 이벤트



6.11 시나리오 진행 후 시나리오마다 데이터 내부 List 저장 확인 (Tobii)

TOBII_Manager (Script)	
Script	TOBII_Manager
▼ L_TOBII	
Size	4
▼ TOBII_Scenario2	
Stage_name	TOBII_Scenario2
Eyes_time	0.1527891
Brake_time	0.4452197
▼ TOBII_Scenario3	
Stage_name	TOBII_Scenario3
Eyes_time	0.1456711
Brake_time	0.3441542
▼ TOBII_Scenario1	
Stage_name	TOBII_Scenario1
Eyes_time	0.0115478
Brake_time	0.4211357
▼ TOBII_Scenario4	
Stage_name	TOBII_Scenario4
Eyes_time	0.172345
Brake_time	0.198457
Scenario 1 Danger	0
Scenario 2 Danger	0
Scenario 3 Danger	0
Scenario 4 Danger	0
Total Danger	0

6.12 시나리오 진행 (Tobii)

- Tobii 급정거 이벤트



- Tobii 차량 끼어들기 이벤트



6.13 시나리오 진행 (Tobii)

- Tobii 신호등 이벤트



- Tobii 교차로 차량 출현 이벤트



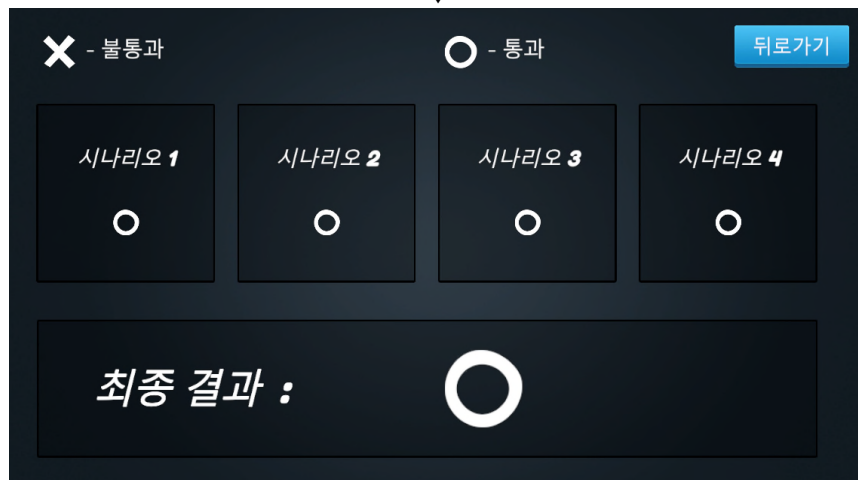
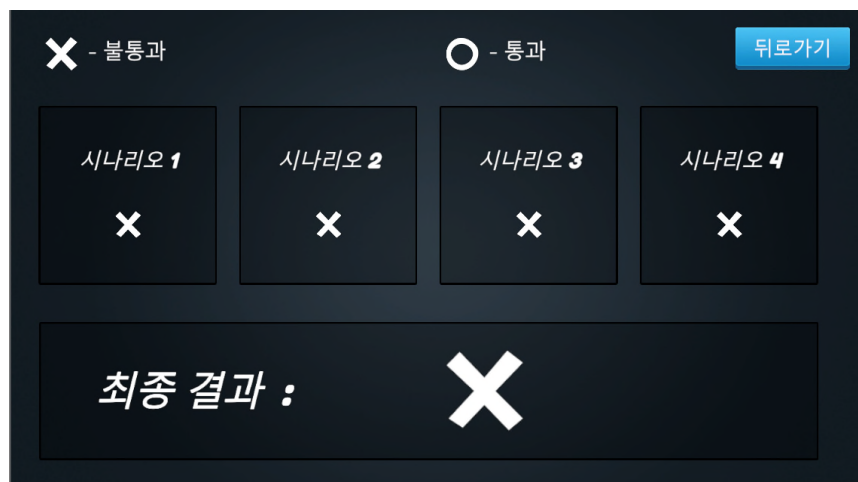
6.14 DB 결과(Record) 확인 (Tobii)

	ID	Password	Name	Scenario1	Scenario2	Scenario3	Scenario4	Is_Danger
	필터	필터	필터	필터	필터	필터	필터	필터
1	2015152025	1q2w3e4r	LeeKangWoo	1	1	1	1	1
2	2017152015	Park	ParkJoonHy...	1	1	1	1	1
3	2017150005	Kwon	KwonJiYoung	0	0	0	0	0



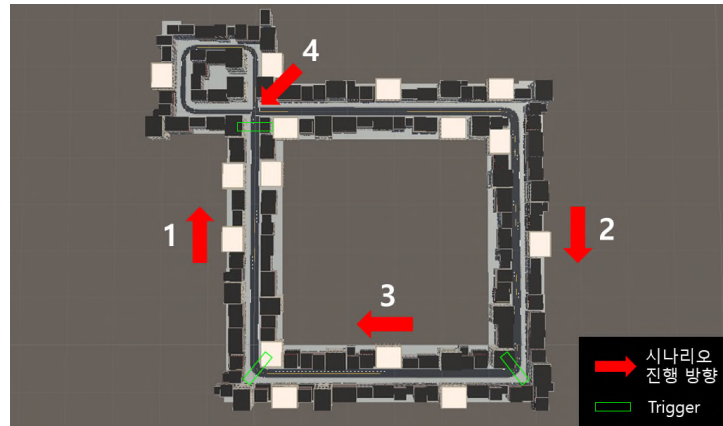
	ID	Password	Name	Scenario1	Scenario2	Scenario3	Scenario4	Is_Danger
	필터	필터	필터	필터	필터	필터	필터	필터
1	2015152025	1q2w3e4r	LeeKangWoo	1	1	1	1	1
2	2017152015	Park	ParkJoonHy...	1	1	1	1	1
3	2017150005	Kwon	KwonJiYoung	1	1	1	1	1

6.15 결과화면(Record) 확인



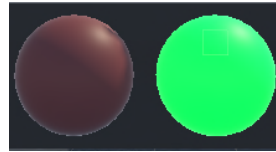
7. Coding & DEMO

7.1 시나리오 이벤트 실행 (수정된 부분만 추가 작성)



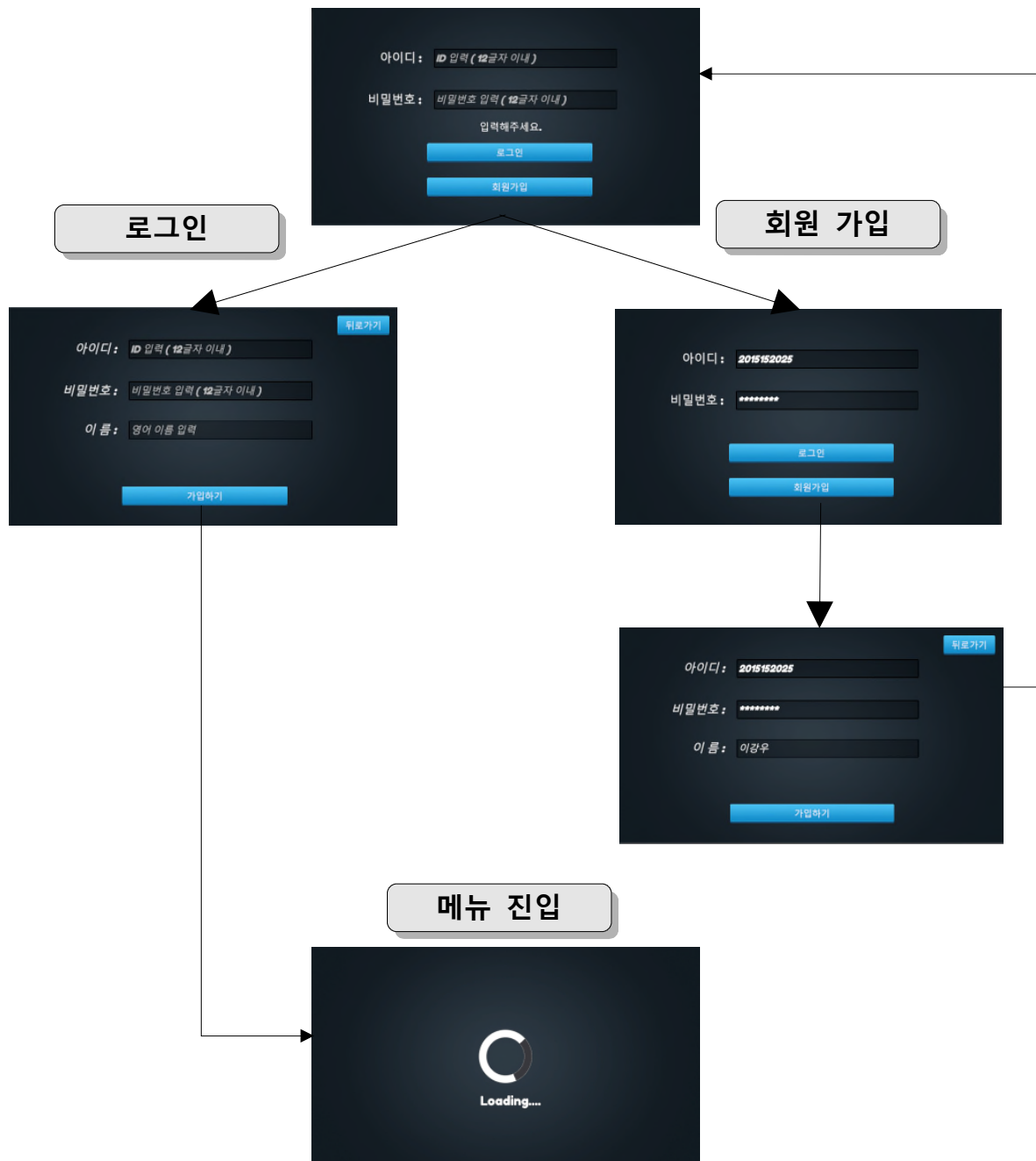
Event_Trigger_Manager 모듈	
클래스	
기능	
1. 각각의 위치에 설정된 Trigger를 지나감 2. 각 위치에서 시나리오에 실행	
상세기능	
<ul style="list-style-type: none"> • 처음 시작을 제외한 총 3개의 Trigger 존재 • 코너를 돌며 도로에 진입할 시 Trigger와 자동차의 충돌 • 각각 위치에서 시나리오에 해당하는 이벤트 실행 	
Event_Trigger_On()	
형식	public void Event_Trigger_On()
리턴 값	void
설명	<ul style="list-style-type: none"> • 운전자(Player)와 Trigger 충돌 • 시나리오에 해당하는 이벤트 실행

7.2 시나리오 1 신호등 색의 전환

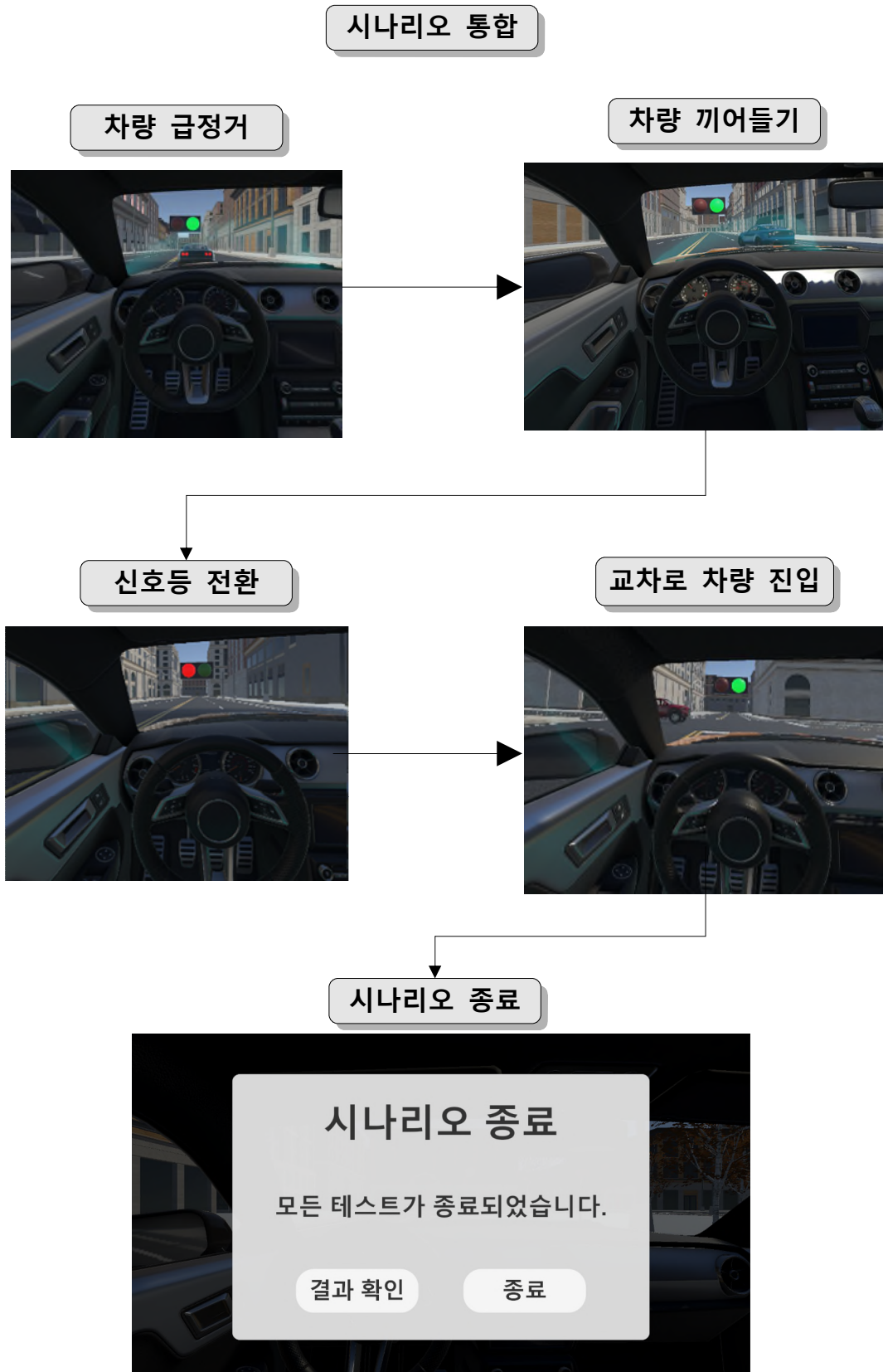


Traffic_Light_Manager 모듈	
클래스	
기능	
<ol style="list-style-type: none"> 1. 타이머 작동 2. 신호등의 색 전환 	
상세기능	
<ul style="list-style-type: none"> • 게임 시작 시 타이머 작동 • 타이머 작동 후 10~12초 사이에 초록색 불이 빨간색으로 전환 • 전환 후 4초 뒤 초록불로 전환 • 이후에 다시 빨간불로 전환되지 않음 	
Change_Traffic_Light()	
형식	public void Change_Traffic_Light()
리턴 값	void
설명	<ul style="list-style-type: none"> • Time.deltaTime을 이용해 게임 시작과 동시에 타이머를 활성화 • 10~12초의 랜덤 시간 후 신호등의 색 전환 • 초록색 -> 빨간색 -> 초록색

7.3 최종 로그인 및 회원가입

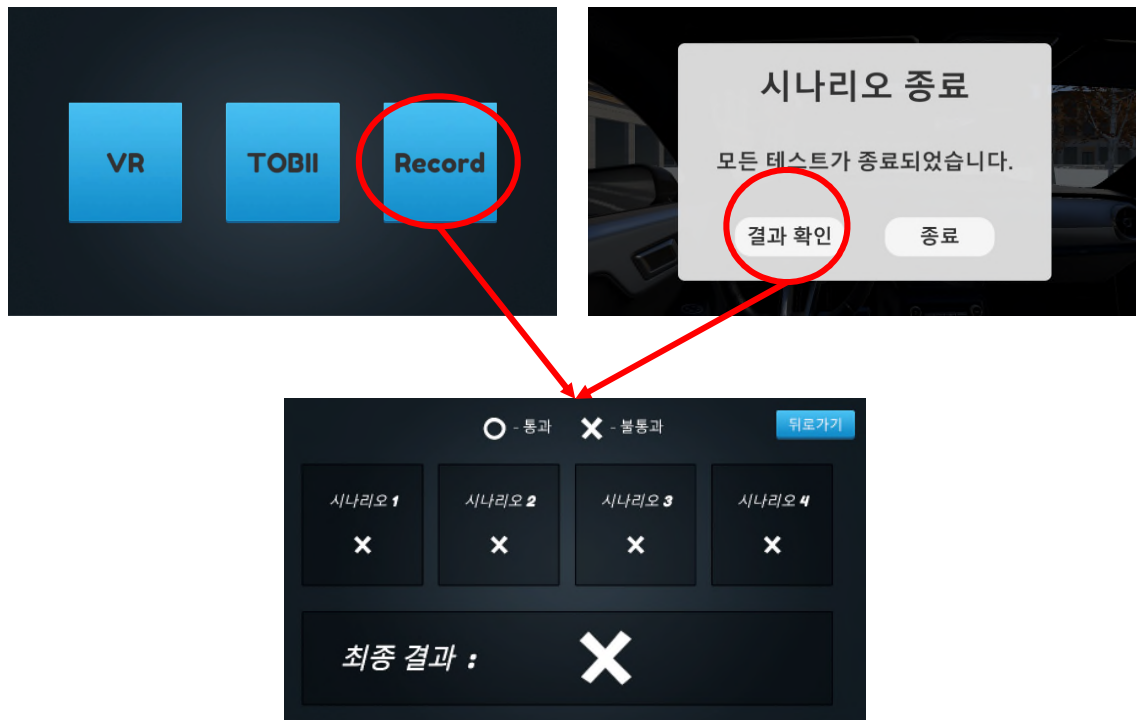


7.4 시나리오 통합 후 진행 순서



7.5 결과 확인

결과 확인



Ⅲ. 결론

1. 연구결과

- 시뮬레이션을 통해 고령 운전자들의 위험군 판별을 안전하게 진행할 수 있음
- VR을 이용하여 실제 운전과 가까운 경험을 제공할 수 있음
- 3D 멀미를 대비하여 TOBII를 이용해 시뮬레이션 진행 가능
- 각자 자신의 ID를 가지고 있으므로, 자신의 최신 결과를 확인할 수 있음
- 총 4개의 시나리오로 구성되어 있으며, 시나리오마다 끊임없이 연속적으로 진행이 되어 사용자의 연속적인 경험을 제공
- 4개의 시나리오를 모두 마무리하게 되면, 자신이 위험군인지에 대한 결과를 확인할 수 있으며, 어떤 시나리오에서 통과하지 못하였는지도 확인할 수 있음

Ⅲ. 결론

1. 작품제작 소요재료 목록

재료명	수량	금액 (수량 x 단가)	비고
-	-	-	-

참고자료

- [1] 전진우, 이수범, 박준태, 엄대룡. (2008). 차량시뮬레이터를 이용한 연령별 운전자 인지반응시간 분석. 대한교통학회 학술대회지, 58(): 547-554
- [2] 주재홍, 최경임, 김주영. (2015). 운전자 인지반응시간에 관한 연구. 대한교통학회 학술대회지, 72(0): 102-106
- [3] 주재홍, 하승우, 정연호, 이춘식. (2010). 운전시뮬레이터를 활용한 운전자 인지반응 시간 연구. 대한교통학회 학술대회지, 63(): 452-457
- [4] 양지웅(YANG Jiwoong), 김진태(KIM Jin-Tae), 김주영(KIM Jooyoung). (2018). 교통 신호등 형태별 운전자 인지반응 시간 비교 연구: 등화 색상 및 조합에 따른 비서술적/서술적 기억 활용. 대한교통학회지, 36(4): 240-250
- [5] 박세진, 오승빈. (2011). 드라이빙 시뮬레이터 환경에서 고령 운전자의 운전 행동 및 인지 특성. 한국감성과학회 추계학술대회, 2011, 65-67
- [6] 김명수, 박상진. (2012). VR 시나리오를 이용한 음주운전자 운전 특성의 유의성 및 상관분석에 관한 연구. 한국도로학회논문집, 14(5), 165-177.
- [7] 한상우. (2016). 가상 운전 시뮬레이터를 이용한 고령자의 운전수행능력 평가. 한국엔터테인먼트산업학회논문지, 10(1), 95-101.
- [8] 고령운전자 운전능력 따라 야간운전 제한
<http://www.kyongbuk.co.kr/news/articleView.html?idxno=1055843>
- [9] 경찰통계자료
<https://www.police.go.kr/www/open/public/public0205.jsp>
- [10] 고령운전자 유발 교통사고
<https://www.seoul.co.kr/news/newsView.php?id=20190909011006>
- [11] 운전면허 시험장 운전 시뮬레이터
<https://news.joins.com/article/23587444>
- [12] 한국 도로 교통공사 VR 교통사고 체험
<http://m-line.tv/project/20180715/>