

2019년 2학기

# HCI 프로젝트 보고서

프로젝트명 : Red Light  
(무단횡단 방지 시뮬레이터)

담당교수 : 황경순교수님

팀명 (조) : TTFF (3조)

제출일자 : 2019. 12. 16.

제 출 자 :

역 할	성명	학번	비고
조 장	이한주	2015041075	●
조 원	안한길	2016039050	●
조 원	정채은	2018038001	
조 원	홍영선	2018038009	
조 원	황승혜	2018038024	

< 변경 기록표 >

버전	변경일자	변경 내용	변경자
v1.0	2019.10.30	자동차 이동 구현, 맵 구조 제작, 유니티-키넥트 연동	안한길
v1.1	2019.11.06	자동차 이동 구현(prefab), 맵 구조 확장, 키넥트 동작 인식 성공	홍영선
v1.2	2019.11.17	카메라 시점 이동 구현, 신호등 구현, 키넥트를 이용한 캐릭터 이동, 회전	정채은
v1.3	2019.11.20	맵 3개 구현, 캐릭터와 차 충돌 구현, 자동차 트리거 구현, 캐릭터 트리거 구현	황승혜
v1.4	2019.11.27	경찰 추적기능 구현, 캐릭터와 차 충돌 효과 구현, 차량 효과음 설정, 메인 메뉴 제작, 엔딩 화면 제작, 튜토리얼 화면 제작,	이한주
v1.5	2019.12.04	게임 UI 변경, 키넥트 인식 개선, 게임 오버 화면 및 재시작 화면 제작, 맵 수정	정채은
v1.6	2019.12.14	로딩화면 제작, 경찰 피격시 3인칭 시점 전환 구현, 키넥트 인식 개선, 난이도 조정	황승혜

## 1. 프로젝트 팀원 소개

	<p>역할 설명</p> <ul style="list-style-type: none"> <li>- Player 제작</li> <li>- Unity Kinect 연동</li> <li>- 게임 기획</li> </ul>		<p>역할 설명</p> <ul style="list-style-type: none"> <li>- 시점 변환</li> <li>- 전반적인 애니메이션 제작</li> <li>- UI 디자인</li> </ul>
<p>조장 : 이한주</p>		<p>팀원 : 안한길</p>	
	<p>역할 설명</p> <ul style="list-style-type: none"> <li>- 신호등 제작</li> <li>- 경찰 제작</li> <li>- 자동차 제작</li> </ul>		<p>역할 설명</p> <ul style="list-style-type: none"> <li>- 맵 디자인 및 제작</li> <li>- 음향 효과</li> <li>- 추가 게임 오브젝트 제작 (Timer, Score Manager 등)</li> </ul>
<p>팀원 : 홍영선</p>		<p>팀원 : 황승혜</p>	
	<p>역할 설명</p> <ul style="list-style-type: none"> <li>- 신호등 제작</li> <li>- 경찰 제작</li> <li>- 자동차 제작</li> <li>- 발표</li> </ul>	<p>사진</p>	<p>역할 설명</p>
<p>팀원 : 정채은</p>		<p>팀원 :</p>	

## 2. 프로젝트 추진 일정

프로젝트 일정	2019.10.01. ~ 2019.12.13								
단계	10				11				12월
	첫째 주 ~11일	둘째 주 ~18일	셋째 주 ~25일	넷째 주 ~31일	첫째 주 ~8일	둘째 주 ~15일	셋째 주 ~22일	넷째 주 ~29일	첫째 주 ~7일
프로젝트 명									
설계(구성)도									
화면 설계									
세부기능 설명서									
알고리즘 설계									
구현									
발표자료 작성									
발표									
보고서 작성									

### 3. 프로젝트 추진을 위한 팀원의 의견정리

#### - 9월 18일 팀원 회의

1. 회의 주제 : 개발 주제 선정
2. 회의 결정 사항
  - 가) 팀 이름 TTFF
  - 나) 지적 장애인을 위한 집을 찾아가는 교육 게임을 만들기로 결정

#### - 9월 25일 팀원 회의

1. 회의 주제 : 게임 주제 선정
2. 회의 결정 사항
  - 가) 키넥트를 이용하고 시각 장애인과 어린 아동을 대상으로 하는 게임 결정
  - 나) 무단횡단을 방지하는 위한 게임을 만들기로 결정

#### - 10월 2일 팀원 회의

1. 회의 주제 : 게임 시나리오 및 스토리 선정
2. 회의 결정 사항
  - 가) 게임 시나리오 및 스토리 선정
  - 나) 스토리 모드와 무한 모드 두 가지 모드 선정
  - 다) 게임의 기본적인 규칙 선정

#### - 10월 17일 팀원 회의

1. 회의 주제 : 게임 시나리오 구체화, 구성도 및 세부기능 제작
2. 회의 결정 사항
  - 가) 목적지의 도착 시간순으로 순위를 매긴다.
  - 나) 횡단 보도에 도착해서 초록 불로 바뀌기까지 시간이 걸린다.
  - 다) 횡단 보도의 신호등이 빨간불일 때 캐릭터의 시야가 제한된다.
  - 라) 주먹을 쥐면 앞으로 간다. (키넥트)
3. 개인 의견
  - 가) 이한주 : 무단횡단 방지 게임에 더해서 음주 운전 방지, 흡연 예방 등 공익 게임을 만들자. 무단횡단게임의 타임어택 모드를 추가하자.
  - 나) 안한길 : 무단횡단 게임의 맵이 무한하도록 설정하고 이동한 거리를 점수로 하는 간단한 게임으로 하자.
  - 다) 정채은 : 무단횡단 게임의 맵을 상세히 정하고 NPC 또는 아이템을 추가하여 만들자.
  - 라) 황승혜 : 무단횡단 방지를 위해 횡단보도의 신호등이 빨간색일 때 게임화면의 시야를 제한하자.
  - 마) 홍영선 : 무단횡단 방지 게임을 자세하게 정하자.

#### - 10월 30일 팀원 회의

1. 회의 주제 : 게임 구성도 수정 및 게임 화면설계도 작성
2. 회의 결정 사항
  - 가) 캐릭터의 진행에 따라 카메라의 시점 변환 효과를 준다.
  - 나) 게임 시작 버튼을 누르면 튜토리얼 진행 확인 메시지를 띄우도록 한다.
3. 개인 의견
  - 가) 안한길 : 게임 튜토리얼 스킵 버튼을 만들자
  - 나) 정채은 : 튜토리얼을 게임 첫 스테이지에 넣어 진행하도록 하자
  - 다) 홍영선 : 게임 메뉴에 튜토리얼 버튼을 추가 하자
  - 라) 황승혜 : 캐릭터가 횡단보도를 진행할 때 카메라의 위치를 캐릭터에 가까이 두어 시야를 좁게 하자
  - 마) 이한주 : 튜토리얼 때 npc가 나와 설명하도록 만들자.

#### - 11월 06일 팀원 회의

1. 회의 주제 : 맵의 구성 및 자동차의 이동방식 및 캐릭터의 이동방식
2. 회의 결정 사항
  - 가) 맵 상의 도로를 다양하게 구현 (2차선, 4차선 등)
  - 나) 캐릭터가 앞으로 가는 것뿐만 아니라 좌우로 방향 이동이 가능

### 3. 프로젝트 추진을 위한 팀원의 의견정리

- 다) 자동차는 앞으로만 이동, 신호등 신호에 맞춰감
- 라) 캐릭터가 자동 경로를 통해 이동

#### - 11월 17일 팀원 회의

1. -회의 주제 : 게임 콘텐츠 추가 및 구현 방향(알고리즘) 결정
2. -회의 결정 사항
  - 가) 오른손을 머리 위로 들면 캐릭터가 오른쪽으로 회전하고, 왼손을 머리 위로 들면 왼쪽으로 회전하게 구현한다.
  - 나) 카메라 전환에 트리거를 사용한다.
  - 다) 캐릭터가 이동하면서 아이템을 먹는다.
  - 라) 미니맵을 표시한다.

#### - 11월 20일 팀원 회의

1. -회의 주제 : 게임 콘텐츠 추가 및 구현 방향(알고리즘) 결정
2. -회의 결정 사항
  - 가) 플레이어의 캐릭터의 HP 상태를 추가한다
  - 나) 양손을 쥐면 한 손을 질 때 보다 속도가 빨라진다
  - 다) 캐릭터를 쫓아오는 경찰관(적)을 구현한다

#### - 11월 27일 팀원 회의

1. -회의 주제 : 게임 콘텐츠 추가 및 구현 방향(알고리즘) 결정
2. -회의 결정 사항
  - 가) 캐릭터가 무단횡단을 하면 경찰관(적)이 추적한다.
  - 나) 캐릭터와 차가 충돌하면 붉은 화면이 나타난다.
  - 다) 자동차에 효과음을 넣는다.
  - 라) 메인 메뉴와 엔딩 화면을 결정한다.

#### - 12월 4일 팀원 회의

1. -회의 주제 : 게임 제작 마무리 및 발표자료 제작
2. -회의 결정 사항
  - 가) 로딩 화면 제작
  - 나) 경찰 충돌 시 3인칭으로 바뀜

#### - 12월 14일 팀원 회의

1. -회의 주제 : 게임 제작 마무리 및 발표자료 제작
2. -회의 결정 사항
  - 가) 게임 실행 동영상 촬영을 위해 게임화면에 사용자 모습이 출력되게 한다.
  - 나) 게임 난이도 조절
  - 다) 팔꿈치가 어깨 위로 올라가면 캐릭터가 회전하게 한다. (인식 방법 수정)
  - 라) 튜토리얼 그림 설명 추가

## 4. 프로젝트 주제 및 목적

### 1. 주제

게임 사용자가 키넥트 동작 인식을 이용하여 각종 장애물로부터 방해 받지 않고 무사히 목적지에 빨리 도착하는 게임

### 2. 목적

가. C# Script와 kinect, unity를 연동하고 이를 활용할 수 있는 능력 향상

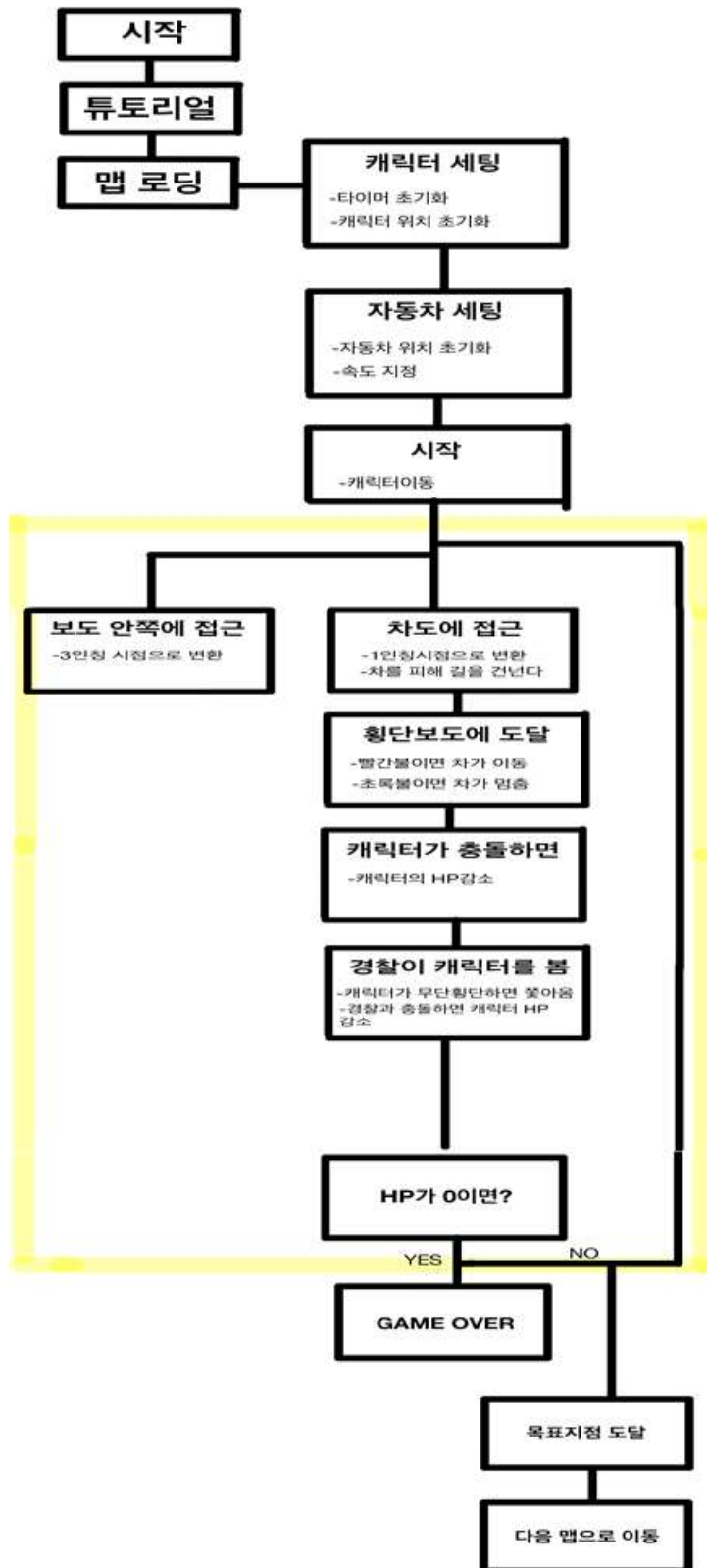
나. 사람들에게 무단횡단의 위험성을 다시 각인 시키는 게임

## 5. 프로젝트 시나리오

1. 게임 시작 화면이 나온다.
2. 캐릭터가 등장하고 주먹을 쥐면 앞으로 가고 손을 피면 멈춘다. (키넥트)
3. 장애물로 횡단보도 및 자동차, 경찰이 나타난다.
4. 차가 다니며 캐릭터 가까이 오면 차 소리가 커지고 마다 멀어지면 작아진다.
5. 횡단보도 이벤트
  - 1) 초록불에 횡단보도를 건너면 차가 다니지 않고 경찰이 쫓아오지 않는다.
  - 2) 빨간불에 횡단보도를 건너면 차가 지나 다니고 경찰이 쫓아오게 된다.
6. 무작위로 오는 차에 캐릭터가 부딪히면 체력이 줄어들고, 체력이 모두 소모되면 게임 오버 화면이 표시된다.
7. 만약 차에 부딪히지 않고 목적지에 도착하면, 다음 스테이지로 가고 모두 성공하면 게임 기록이 나온다.



## 6. 프로그램 구성도 및 세부기능 설명



## 6. 프로그램 구성도 및 세부기능 설명

### \*“Red Light” 게임의 세부기능

1. 캐릭터 : 맵에서 길을 건너서 목표지점으로 간다.
  - 속도 ( ex) 양손을 모두 쥐면 인식률이 좋아짐)
  - 방향 ( ex) 오른손을 머리 위로 들면 오른쪽으로 회전)
  - 효과 ( ex) 캐릭터와 차량 충돌 시 화면이 붉어짐)
2. 자동차 : 캐릭터의 이동을 방해하는 역할을 한다.
  - 속도 ( ex) 차선에 따라 속도가 다름)
  - 모양 ( ex) 택시, 버스, 승용차가 랜덤)
  - 이동 ( ex) 신호등 불빛에 따라 움직임 제어)
  - 소리 ( ex) 캐릭터와 자동차 거리에 따라 소리 크기 제어)
3. 신호등 : 불빛이 달라지면서 자동차를 제어하는 역할을 한다.
  - 신호의 종류 ( ex) 초록불일 경우 차가 멈춤)
4. 맵 : 스테이지마다 게임의 난이도와 특징이 다르다.
  - 화면 ( ex) 도로, 도보, 자동차와 같은 정보를 담고 있음)
  - 효과 ( ex) 안개 효과로 시야를 흐리게 함)
5. 카메라 : 캐릭터의 시점을 바꿔주는 역할을 한다.
  - 차도에 접근 ( ex) 1인칭의 시점으로 바꿈)
6. 경찰: 캐릭터의 무단횡단을 감시하는 역할을 한다.
  - 역할( ex) 캐릭터를 감시하며 무단횡단을 하면 쫓아가서 충돌-->캐릭터의 이동 정제)

## 7. 프로그램 화면 설계

### 1. 메뉴 화면

Crossing on the Street  
Red Light



• START  
• EXIT

### 2. 튜토리얼 화면

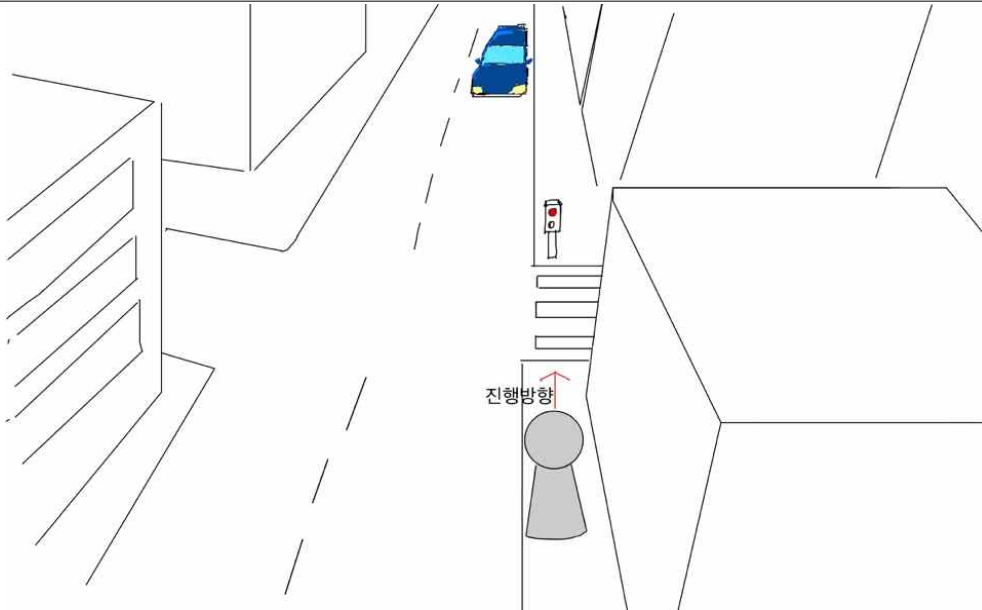
skip

npc

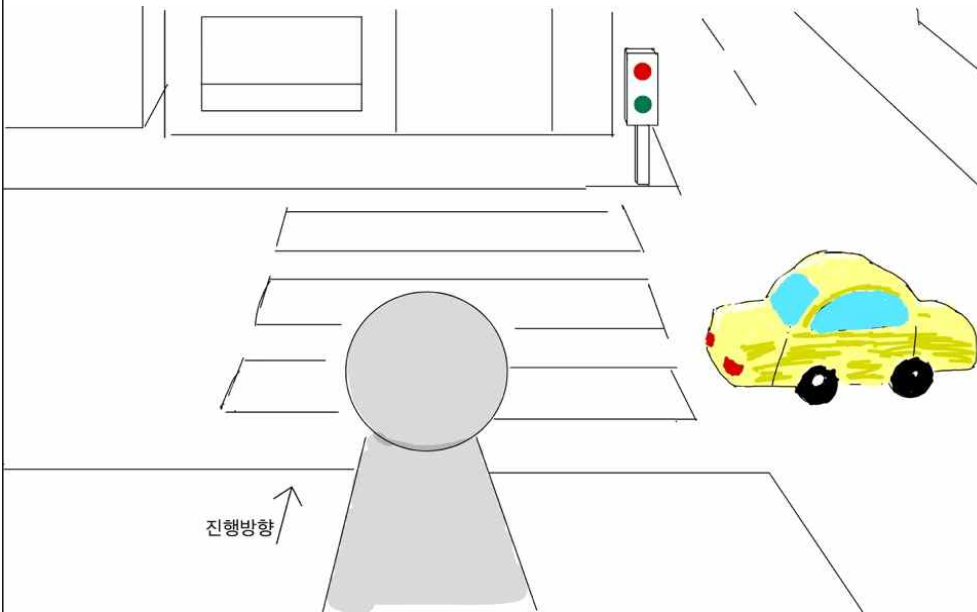
튜토리얼 내용  
조작방법 게임 목표 등이 설명 된다.

### 3. 게임 시작 화면

## 7. 프로그램 화면 설계



## 4. 횡단보도 앞 시점 (파란불일 경우 원래 시야처럼 넓어짐)



## 7. 프로그램 화면 설계

### 5. 게임오버 화면

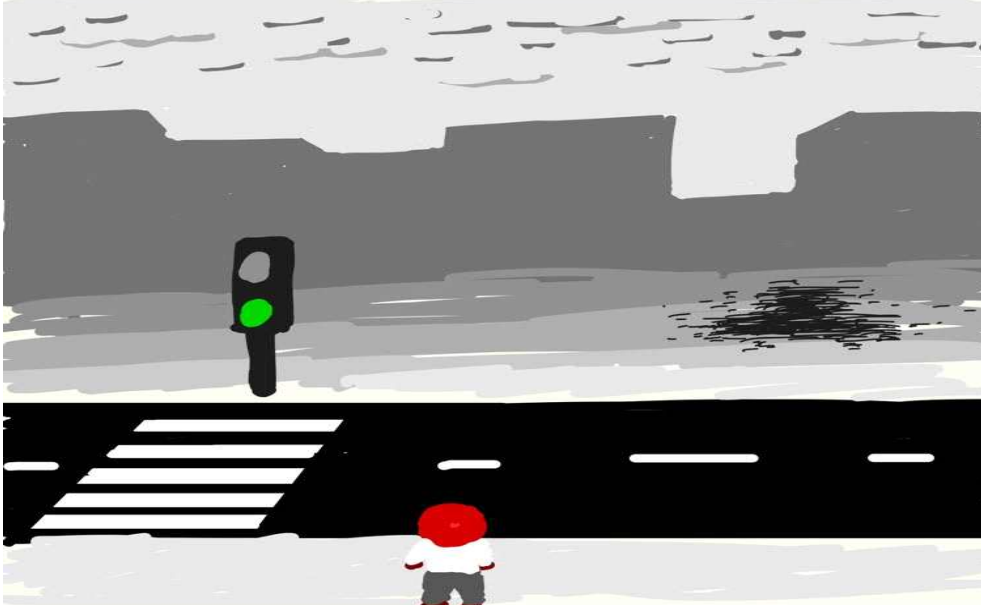


### 6. Stage 2 화면

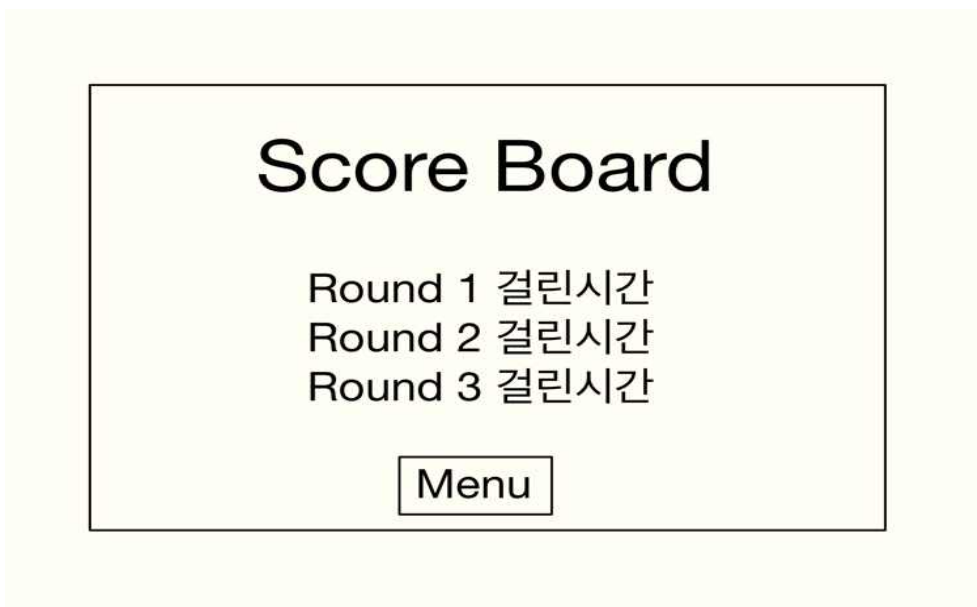


## 7. 프로그램 화면 설계

### 7. Stage 3 화면



### 7. Score Board 화면



## 7. 프로그램 화면 설계

### 8. Loading 화면

맵 마다 로딩화면의 색이 다르다.

# Now Loading...



tip. 화면하단에 팁을 출력한다.

## 8. 자료구조 설계

### 1. 자동차

- string CarName // 자동차 이름 (자동차 id)
- Vector3 CarPosition // 자동차 위치
- float CarMoveSpeed // 자동차 속도 (변하는 값)
- enum CarState { MOVE, STOP, STAY } // 자동차 상태

### 2. 자동차 클론

- float posX, posY, posZ // 자동차 클론의 생성 좌표
- float angX, angY, angZ // 자동차 클론의 회전 각도
- Random CarChoice // 생성할 자동차 클론을 랜덤으로 정하는 변수

### 3. 플레이어

- string CharName // 캐릭터 이름 (캐릭터 id)
- float CharSpeed // 캐릭터의 이동 속도
- int CharRotation // 캐릭터의 회전 방향 (왼쪽, 오른쪽, 정면)
- Vector3 CharAngle // 캐릭터의 회전 값
- int CharState // 캐릭터의 상태 (충돌 없음, 자동차 충돌, 경찰관 충돌)

### 4. 경찰

- string PoliceName // 경찰의 id
- float PoliceSpeed // 경찰의 속도
- Vector3 PoliceAngle // 경찰의 회전 값

### 5. 목적지

- Vector3 GoalPosition; // 목적지의 좌표
- Vector3 skipPosition; //튜토리얼 스킵 버튼

### 6. 횡단보도

- Vector3 CrossingPosition // 횡단보도의 좌표

### 7. 신호등

- string TrafficName // 신호등의 이름 (신호등 id)
- float RedLightTime // 빨간불의 지속 시간 (변하지 않음)
- float GreenLightTime // 초록불의 지속 시간 (변하지 않음)
- float TrafficLightTimer // 신호의 지속 시간 측정 (빨간불, 초록불과 비교)

### 8. 카메라

- string CamName // 카메라의 이름 (카메라 id)
- Vector3 CamPosition // 카메라의 위치
- Vector3 CamRotation // 카메라의 회전 값

### 9. 타이머

- Text TimeText// 들어갈 텍스트 값
- string TimeStr // TimeCnt를 String으로 바꿈
- float TimeCnt // 경과 시간

### 10. 맵

- Vector3[] StreetPosition // 길의 위치 정보
- Vector3 PlayerPosition // 플레이어 위치
- Vector3 GoalPosition // 목표 지점 위치
- Vector3[] CarPosition // 자동차 위치



## 9. 알고리즘

기능	캐릭터 이동, 회전
원손 또는 오른손을 쥐면 캐릭터가 앞으로 이동한다	
왼팔을 머리 위로 들면 캐릭터가 왼쪽으로 회전한다	
오른팔을 머리 위로 들면 캐릭터가 오른쪽으로 회전한다	

기능	캐릭터 충돌
만약 캐릭터가 자동차와 충돌한다면 화면을 붉게 변화한다	
만약 캐릭터가 경찰관과 충돌한다면 캐릭터의 상태를 경찰관 충돌로 바꾼다	

기능	목표 지점 도착
만약 캐릭터가 목표 지정에 도착한다면 다음 맵으로 넘어간다	

기능	게임 오버
만약 캐릭터의 목숨이 0 보다 작아지면 게임 오버 화면으로 넘어간다	

기능	카메라 이동
만약 캐릭터가 1인칭 트리거와 충돌한다면 1인칭 카메라를 활성화 한다 3인칭 카메라를 비활성화 한다	
만약 캐릭터가 3인칭 트리거와 충돌한다면 1인칭 카메라를 비활성화 한다 3인칭 카메라를 활성화 한다	

## 9. 알고리즘

기능	자동차 충돌
만약 자동차 상태가 STAY면 자동차 상태를 MOVE로 바꾼다	
레이캐스트를 안전거리 만큼 발사한다면 만약 충돌체의 태그가 "Car"면 자동차의 상태를 STAY로 바꾼다	
만약 자동차가 Destroy 트리거를 만난다면 자동차 오브젝트를 삭제한다	
만약 자동차가 Stop 트리거를 만나다면 자동차의 속도를 0으로 한다	
만약 자동차가 Go 트리거를 만난다면 자동차의 속도를 복구한다	

기능	자동차 이동
자동차의 상태를 받아온다.	
자동차 상태가 MOVE이면 자동차를 속도만큼 앞으로 이동한다.	

기능	자동차 복제
Car1부터 Car3까지 랜덤하게 자동차를 선택한다	
지정된 위치에 랜덤으로 선택된 자동차를 복제한다	

기능	키넥트 인식
키넥트로 몸 데이터를 읽어서 m_Body 객체에 저장한다	
foreach(m_Body 객체에서 몸 정보를 받아온다) 만약 왼손이 닫혀 있으면 왼손의 grap을 true로 한다 만약 오른손이 닫혀 있으면 오른손의 grap을 true로 한다	
만약 왼손을 머리 위로 들면 왼손의 rotation을 true로 한다 만약 오른손을 머리 위로 들면 오른손의 rotation을 true로 한다	

## 9. 알고리즘

기능	타이머
타이머를 00:00으로 초기화하고 동작	
만약 타이머가 동작된다면 시간을 1초 단위로 증가 시간을 초 단위로 출력	

기능	신호등 신호 변경
만약 현재 신호 시간이 빨간불의 지정 시간을 넘으면 초록불을 켜준다 빨간불을 꺼준다 현재 신호 시간을 0으로 만들어 준다 Stop 트리거를 활성화한다 Go 트리거를 비활성화 한다	
만약 현재 신호 시간이 초록불의 지정 시간을 넘으면 빨간불을 켜준다 초록불을 꺼준다 현재 신호 시간을 0으로 만들어 준다 Stop 트리거를 비활성화한다 Go 트리거를 활성화 한다	

기능	경찰 이동
만약 Player의 Jaywalking이 True이면 경찰이 Player를 쫓아온다. 그렇지 않으면 제자리에 있다	

## 10. 실행결과 화면



### Main Menu

- START 부분에서 주먹을 쥐면 게임이 시작된다.
- EXIT 부분에서 주먹을 쥐면 게임이 종료된다.



### Tutorial

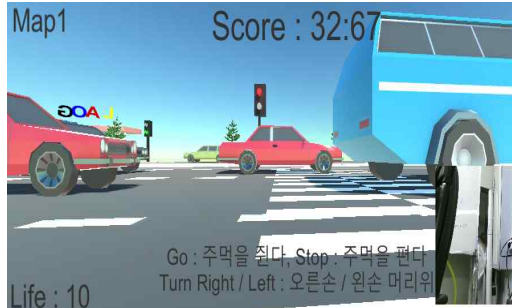
- 튜토리얼 메시지를 보여주면서 사용법을 알려준다.
- 빨간 큐브에 닿으면 Tutorial이 skip된다.



### Map1 3인칭 시점 화면

- 3인칭 시점으로 화면을 넓게 본다.
- 손동작으로 캐릭터를 조종하면서 목표 지점으로 이동한다.
- Player가 길을 건너려고 하면 시점이 바뀌게 된다.

## 10. 실행결과 화면



### Map1 1인칭 시점 화면

- 1인칭 시점으로 화면을 좁게 본다.
- 무단횡단 하기가 어려워 지며 신호등을 통해 길을 건너기를 유도한다.



### Map2

- Map1을 클리어하면 Map2가 나온다
- Map2는 안개가 가득한 맵으로 맵마다 테마가 다르다



### Map3

- Map2를 클리어하면 Map3로 이동한다.
- Map3는 어두운 맵이어서 게임의 난이도를 변경했다.

## 10. 실행결과 화면



### Score Board 화면

- 게임을 전부 클리어하면 기록을 볼 수 있다.



### 로딩화면

- 맵을 불러오는 동안 로딩 화면이 나타난다.

## 11. 구현 중 발생한 오류 및 해결방법

오류	신호등이 2개 이상일 때 신호등의 초록불, 빨간불 색깔이 변하지 않는다.
내용	<p>- 신호등 오브젝트에 적용된 스크립트는 신호등이 한 개만 있을 때 동작하는 코드였다. 신호등을 제어하는 코드는 전체 오브젝트에서 빨간불 오브젝트와 초록불 오브젝트를 찾는데, 이것은 신호등이 두 개 이상 있을 때는 한 개의 빨간불과 초록불만 찾게 되는 결과가 나오게 된다. 따라서 신호등이 두 개 이상일 때는 신호등이 동작하지 않게 된다.</p> <p>해결책</p> <p>- 코드에서 빨간불 오브젝트와 초록불 오브젝트를 찾을 때 게임 오브젝트 전체에서 찾는 것이 아니라 스크립트가 연결된 신호등 오브젝트 내에서 빨간불 오브젝트와 초록불 오브젝트를 찾도록 수정을 했다.</p>

오류	캐릭터와 경찰이 충돌했을 때 해당하는 애니메이션이 실행되지 않았다.
내용	<p>- 캐릭터와 경찰이 충돌하면 캐릭터의 상태는 HIT이 된다. 캐릭터의 상태가 HIT이면 캐릭터 상태를 DIZZY로 바꾸고 일정 시간 동안 머리를 흔드는 애니메이션이 실행되고 움직일 수 없어야 한다. 하지만 두 기능 모두 실행되지 않았고 여전히 움직일 수 있었다.</p> <p>해결책</p> <p>- 캐릭터 상태가 DIZZY 일 때만 움직일 수 없게 만들었는데, 캐릭터 상태가 HIT일 때도 움직일 수 없도록 코드를 수정했다.</p>

오류	복제된 자동차 오브젝트들끼리 사고가 났다.
내용	<p>- 랜덤으로 복제된 자동차들이 회전값과 위치값이 고정되지 않아서 자동차들이 이동해야 하는 지정된 차선을 벗어나서 옆 차선의 차들과 사고가 났다.</p> <p>해결책</p> <p>- 원본 자동차의 코드 초기화 부분에서 회전축(X, Y, Z)을 고정하고 위치값(Y)을 고정함으로써 복제된 차들이 지정된 차선에서만 움직일 수 있게 수정 했다.</p>

## 12. 참고문헌(사이트)

1. 신호등 제작 (유튜브)  
<https://www.youtube.com/watch?v=BR1RmGNo3HM>
2. 게임 컨셉 (길건너 친구들)  
<https://play.google.com/store/apps/details?id=com.yodo1.crossyroad&hl=ko>
3. 키넥트 연결 구현 (마이크로소프트 공식 문서, 유튜브)  
[https://docs.microsoft.com/ko-kr/previous-versions/windows/kinect/dn799273\(v=ieeb.10\)](https://docs.microsoft.com/ko-kr/previous-versions/windows/kinect/dn799273(v=ieeb.10))  
<https://www.youtube.com/watch?v=aHGILxh6a88>
4. 경찰 움직임 구현 (유니티 공식 문서, 티스토리 블로그)  
<https://docs.unity3d.com/kr/530/ScriptReference/NavMeshAgent.html>  
<https://solution94.tistory.com/19>
5. 캐릭터 애니메이션 구현 (티스토리 블로그, 유니티 공식 문서)  
<https://treeofimaginary.tistory.com/66?category=726133>  
<https://docs.unity3d.com/kr/530/ScriptReference/Animation.html>
6. 레이캐스트 (유니티 공식 문서)  
<https://docs.unity3d.com/kr/530/ScriptReference/Collider.Raycast.html>
7. 효과음 및 배경음악 (유튜브)  
<https://youtu.be/AubSRuzYNXQ>  
<https://www.youtube.com/audiolibrary/music?nv=1>  
<https://www.youtube.com/channel/UCHae4C99XJORB7log62wqvw>



## 13. 고찰

이한주

- 프로젝트의 팀장이 되어서 팀원들을 모아서 진행한다는 것이 쉽지 않은데, HCI 과목에서는 팀원들이 다들 적극적으로 참여를 해주어서 진행이 빠르고 쉽게 되었습니다. 게임을 전체적으로 기획하고 팀원들과 매주 회의를 통해 세부 사항을 결정하면서 게임이 구체화 되는 과정을 하게 되고, 장기 프로젝트를 진행하는 방법을 배운 것 같습니다. 또한 매주 보고서를 통해 다른 사람에게 우리의 프로그램 진행 상황을 알려주었는데, 이 과정을 통해 현재 프로그램의 진척도를 정리할 수 있어서 프로그램에 대한 이해도를 높일 수 있게 되었습니다. 유니티와 C#은 전혀 사용하지 않아서 게임 개발을 하는데 많은 어려움을 예상했는데, 게임 내 오브젝트를 하나씩 만들면서 반복되는 기능들과 코드들을 사용하다 보니 게임을 완성할 수 있었습니다. 이를 통해 내가 모르는 새로운 언어를 통한 개발 경험도 중요하고 생각보다 어렵지 않다는 것을 알 수 있게 되었습니다. 그리고 스크립트의 수가 많이 나오게 되었는데 클래스의 관계를 설계하는 방법을 알아야 되는 필요성을 느끼게 되었습니다.

안한길

- 인간상호를 하며 처음 게임을 만든다는 생각에 들떠 많은 것을 해보고 싶었습니다. 하지만 생각 외로 신경써야 할 부분이나 구현하기에 어려움이 많아 아쉬웠던 점이 많았습니다. 팀 프로젝트를 하며 저에게 부족함을 많이 느꼈고 부족함을 느낀 만큼 프로젝트에 기여하기 위해 노력도 많이 하였습니다. 그렇게 프로젝트를 진행하면서 의견이 맞지 않아 의견도 조율해보고 같이 상의도 하면서 게임을 조금씩 만들어져 가는 것을 볼 수 있어 좋은 경험이었습니다.

정채은

- 게임을 플레이만 했을 때는 게임을 만들고 싶다는 생각만 했었지만, 막상 진짜 구현을 해보니 오브젝트 하나하나 세부적인 설정을 해줘야 한다는 것을 알게 되었다. 또한, 오류가 발생했을 때 모든 방법을 다 동원하여 문제해결을 위한 시도를 해야 한다는 점을 배웠다.

홍영선

- 게임을 만들며 어떤 기능을 구현하기 위해서는 오브젝트에 세부적인 성질을 하나씩 적용해 줘야 한다는 것을 알게 되었다. 사소한 기능 하나를 구현하기 위해서 많은 노력이 필요하다는 것을 깨닫게 되었다.

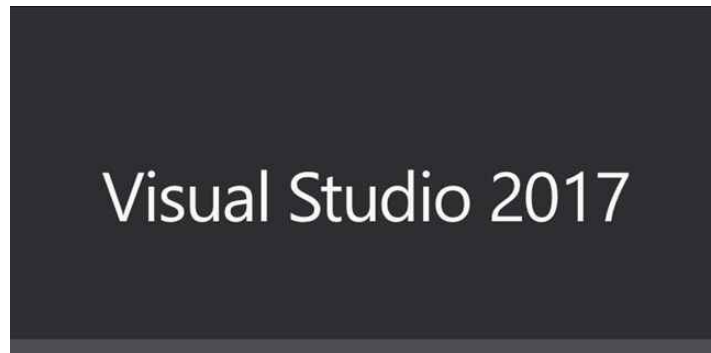
황승혜

- 게임이 길을 건너서 목적지에 도착하기까지의 시간을 측정하기 때문에 직접 플레이 해보면서 빨리 도착하기 위해 무단횡단을 해야 하는 것에 아이러니를 느꼈다. 게임을 제작하면서 현실에서도 무단횡단을 신경쓰게 되었다. 또한, 게임 제작도 다른 프로그래밍처럼 많은 이들의 시간과 노력을 요구한다는 사실을 새삼 깨닫게 되었다.

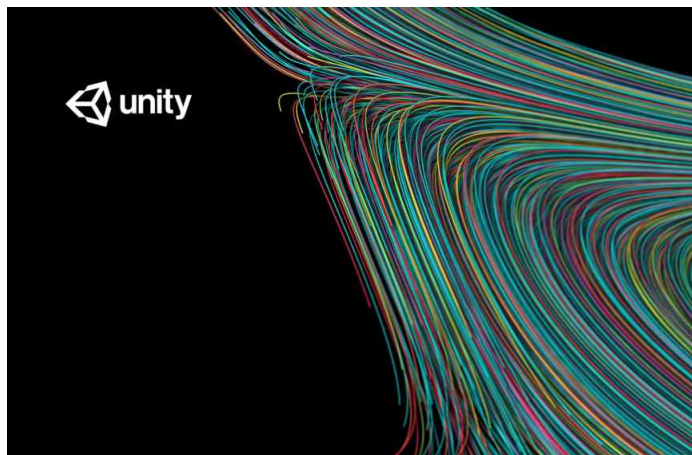
## 14. 부록

### 개발 환경

- OS: Windows 10
- 사용 언어 : C#
- Tool
  - \* Visual Studio 2017



- \* Unity (ver.2018.4.12.f1)



- \* Microsoft Kinect V2



## 14. 부록

### 깃허브

코드 주소

[https://github.com/Yanjoo/TTFF\\_CODE](https://github.com/Yanjoo/TTFF_CODE)

The screenshot shows the GitHub repository page for 'Yanjoo / TTFF\_CODE'. At the top, there's a header with the repository name and navigation links: Unwatch (1), Star (0), and Fork (0). Below this is a secondary navigation bar with links for Code, Issues (0), Pull requests (0), Actions, Projects (0), Wiki, Security, Insights, and Settings. The main content area starts with the repository name '팀 TTFF 코드' and an 'Edit' button. Below this is a 'Manage topics' section. A green progress bar shows repository statistics: 1 commit, 1 branch, 0 packages, 0 releases, and 1 contributor. Below the progress bar are buttons for 'Branch: master', 'New pull request', 'Create new file', 'Upload files', 'Find file', and 'Clone or download'. A list of files is shown, including 'Yanjoo 작성자 추가' (Latest commit 1ca23a5 12 hours ago) and 'Scripts' (작성자가 추가 12 hours ago). At the bottom, there's a light blue box with the text 'Help people interested in this repository understand your project by adding a README.' and a green 'Add a README' button.