

## [캡스톤디자인 결과보고서]

### ■ 과제명

과제명	urban traffic/pedestrian simulation 내 신호 무시 agent 구현 및 대차 알고리즘 개발	참여학기	2022 년 1 학기
-----	---	------	-------------


### ■ 강좌정보

과목명	소프트웨어융합캡스톤디자인	학수번호	SWCON40100
과제기간	2022 년 3 월 2 일 ~ 2022 년 6 월 17 일	학점	3

### ■ 팀구성

팀명	OMUS		팀구성 총인원	1 명
구분	성명	학번	학부(과)	학년
대표학생	이륜하	2019102114	소프트웨어융합	4
참여학생				

### ■ 지도교수 확인

지도교수	성명	강형엽	직급	전임교원	
	소속	소프트웨어융합학과		성명 :	강형엽 

### ■ 붙임

[양식] 과제 요약보고서

[결과물] 최종결과물 (최종작품 사진/도면/발표자료 등)

본 팀은 과제를 성실히 이행하고 이에 따른 결과보고서를 제출합니다.

2022 년 6 월 16 일

팀 대표 : \_\_\_\_\_ 이륜하

  
(인)

## [캡스톤디자인 과제 요약보고서]

과제명	urban traffic/pedestrian simulation 내 신호 무시 agent 구현 및 대처 알고리즘 개발
<p>1. 과제 개요</p> <p>가. 과제 설계 배경 및 필요성</p> <p>현재 일반적인 urban traffic simulation 은 그럴 듯 하기는 보이지만 현실 세계를 완벽하게 시뮬레이션하고 있다고 보기는 힘들다. 특히 일정한 라인과 속도, 완벽한 교통 법규를 지키면서 운전하는 자동차들은 분명 이상적인 모습이지만 현실과 괴리가 있다는 한계를 지니고 있다. 이번 과제에서는 그중에서 교통 신호를 지키지 않는 운전자, 보행자와 그로 인해 발생하는 사고 상황을 시뮬레이션해 보는 데에 중점을 두고자 한다.</p> <p>나. 과제 주요내용</p> <p>SUMO 에서 사용되는 개념을 참고하여 traffic simulation 을 구축한 뒤, 일반적인 차량들과는 달리, 신호등의 정지 신호를 무시하고 달리는 차량을 시뮬레이션 내에 적용시키고자 한다. 또한 이러한 차량으로 인해 발생하는 사고 상황을 대처하고 정상화하는 알고리즘을 개발한다. 이를 인도에서도 적용해, Pedestrian simulation 을 구축한 뒤, 신호를 무시하고 무단횡단하는 보행자와 이에 따라 발생하는 사고 상황을 테스트 한다. 마지막으로 두 시뮬레이션을 동시에 발생시켜 차량과 사람 간의 사고에도 적용해 본다.</p> <p>다. 최종결과물의 목표</p> <p>교통 신호, 보행자 움직임과 인구 유동량, 그리고 돌발적인 운전자와 보행자 움직임을 모두 실시간에 고려하는 urban traffic/pedestrian simulation 구현</p> <p>최소 세 종류 이상의 도로 형태가 포함된, urban traffic/pedestrian simulation 이 구현된 게임 파일 완성본, 최종 과제가 업로드된 github 페이지, 약 3~5 분 정도의 시연영상</p>	
<p>2. 과제 수행방법</p> <p>SUMO Based Platform for Cooperative Intelligent Automotive Agents 의 논문 내용을 참고하여, ue4 를 이용해 시뮬레이션 구축, 에셋은 언리얼 엔진 마켓을 참고하고자 한다.</p>	

#### 나. 과제수행 계획(실습비 사용계획 포함)

먼저 기본적인 urban traffic simulator 를 구현한 뒤, 신호 무시 차량과 이에 대한 사고 정상화 알고리즘을 개발한다. 이후 이를 인도에도 적용시켜, pedestrian simulator 를 먼저 구축한 뒤, 신호 무시 보행자를 구현한다. 마지막으로 가능하다면 이 둘을 혼합하여 모든 사고 상황에 대한 정상화 시스템을 개발한다.

### 3. 수행결과

#### 가. 각 object 의 구성

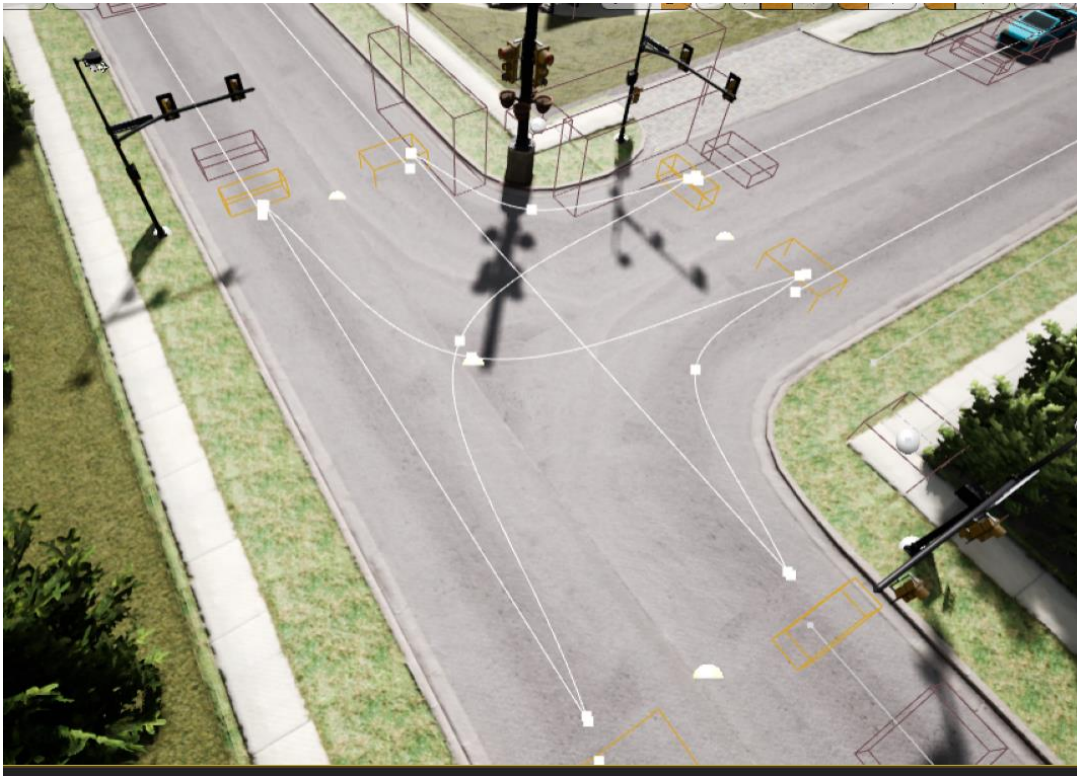
##### 1) 도로

도로의 구성에 spline 을 사용하여, 각 도로는 spline 을 따라 일정한 방향성을 가지고 있다. 각각의 agent 는 도로의 collision box 또는 신호등을 통해서 이 spline 을 인식 및 저장하고, spline 의 방향을 따라 이동한다.

##### a) 차량용 도로

도로의 입구(spline 의 시작점)에 Collision box 를 설치하여, 차량이 spline 을 따라 이동하도록 통제한다. 교차로는 인식한 차량이 가진 dir 변수에 따라 차량의 이동 방향을 결정한다.





#### b) 보행자용 도로

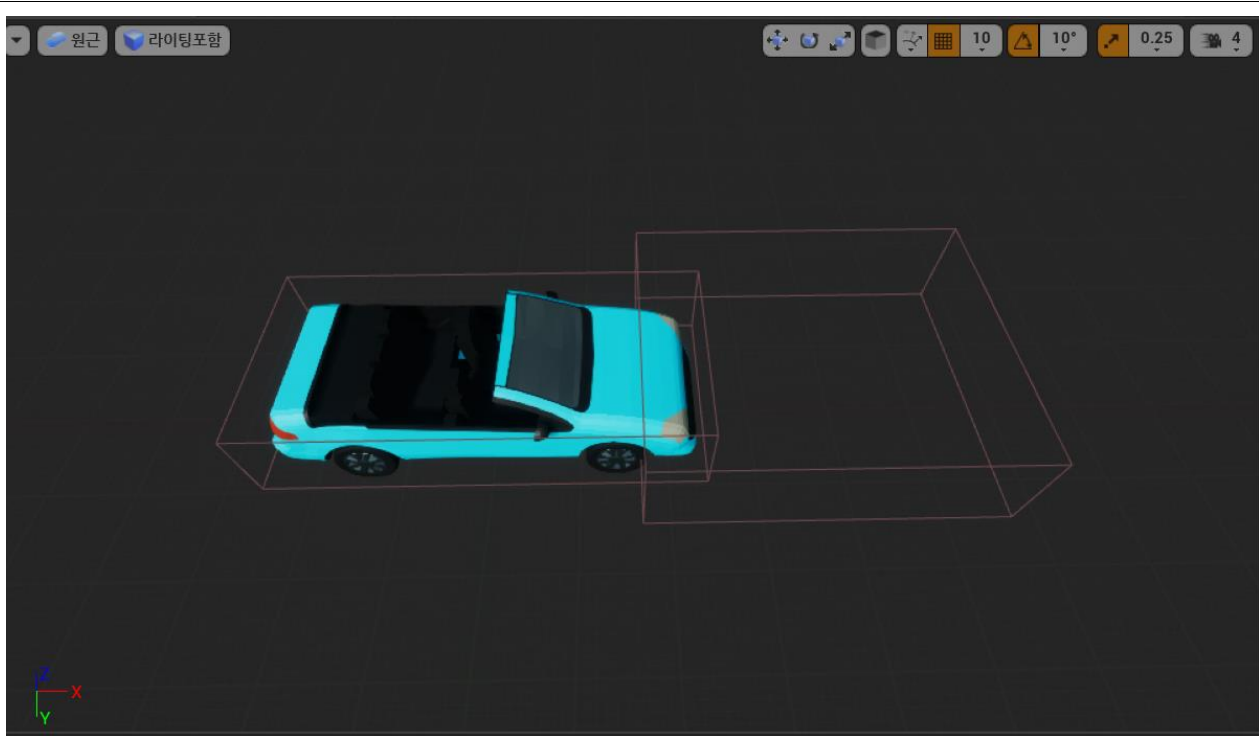
각 횡단보도용 신호등에 종속되어 있으며, 보행자는 신호등을 통해 spline 을 인식한다.  
또한 spline 과 별개로 NavMesh 를 필요로 한다.

#### 2) Agent

다른 오브젝트를 감지하는 detect 와 충돌을 인식하는 hit 두 개의 collision 을 갖는다. 도로의 spline 을 변수로 받아 이를 따라 연속적으로 이동한다.

#### a) 차량

Ue4 의 vehicle blueprint 를 참조하여 구성하였다.



Legality 변수에 따라 Stop 함수, ChangeSpeed 함수를 확률적으로 무시할 수 있다. 이는 불법운전자들이 신호등이나 교통표지판의 정지/감속 신호를 종종 무시한다는 점에서 착안하였다.

Spawn 함수에서 랜덤으로 정해진 Legality 를 따라 차량 색깔과 속도를 결정한다.

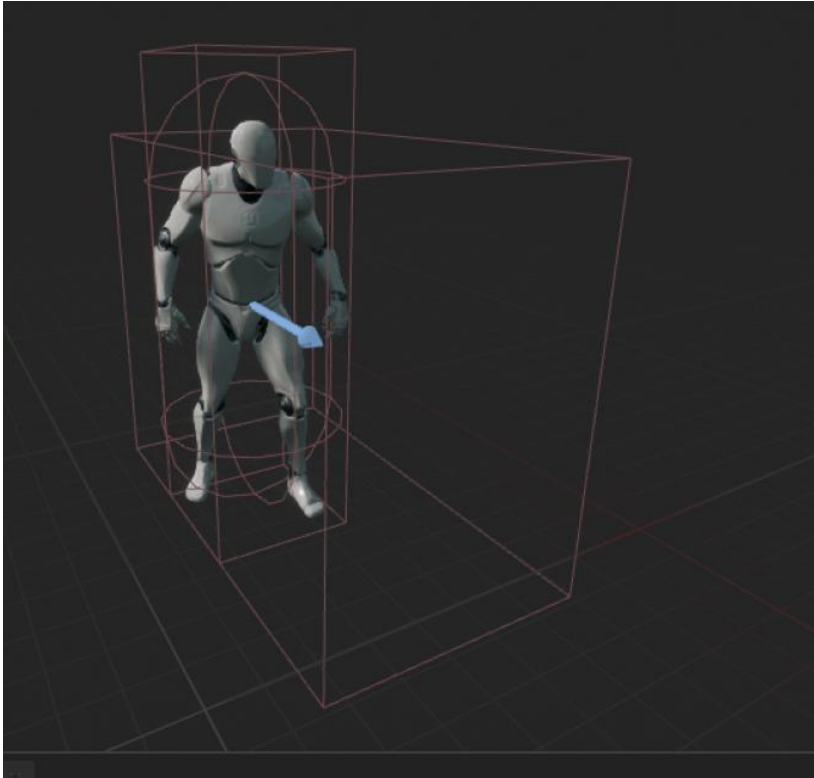
legality > 0.3 : 파랑/흰색 차량, 기본 속도의 -0.2~0.2 배 빨라진 속도

legality < 0.3 : 빨강/검정 차량, 기본 속도의 0~0.3 배 빨라진 속도

#### b) 보행자

Ue4 의 Third Person blueprint 를 참조하여 구성하였다.

신호등을 통해 도로의 spline 을 변수로 받는다. AIController 의 AIMoveTo 와 NavMesh 를 통해 변수에 저장된 spline 의 각 지점을 따라가며 연속적으로 이동한다.



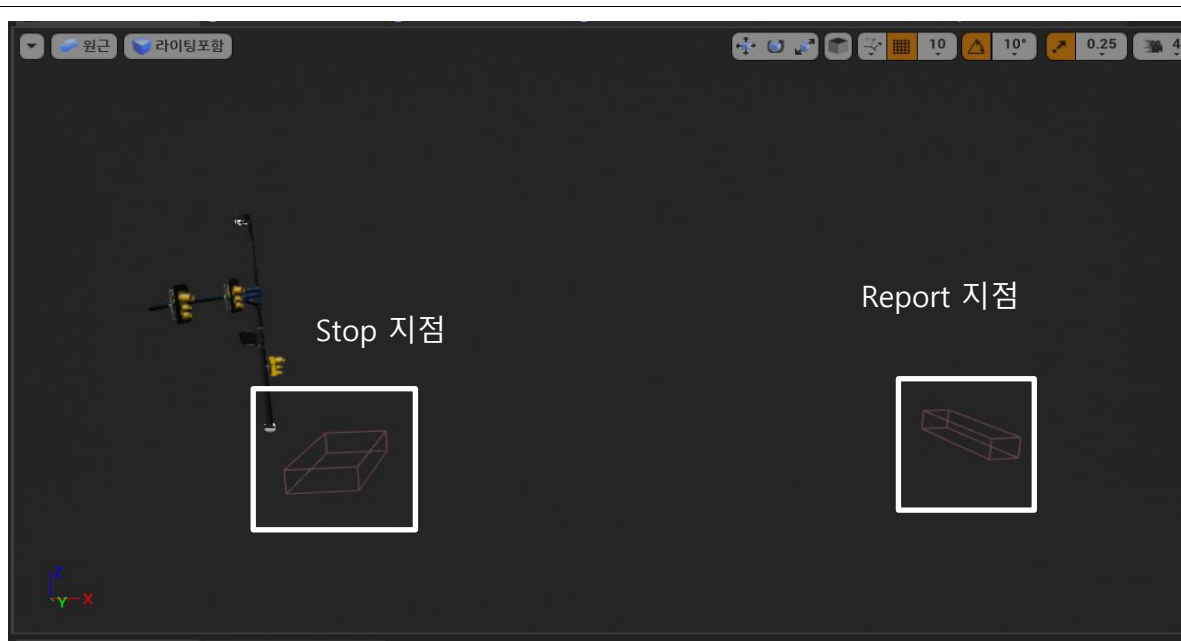
### 3) 신호등 (judge)

두 개의 Collision box 를 이용해 agent 를 인식한다. 인식한 agent 에게 랜덤한 번호를 부여하여 다음 교차로에서의 방향을 결정한다.

빨간 불일 때 가장 최근에 인식한 agent 를 leader 로 설정한 뒤 stop 명령을 내리고, 다음 초록 불에 leader 에게 이동 명령을 내린다.

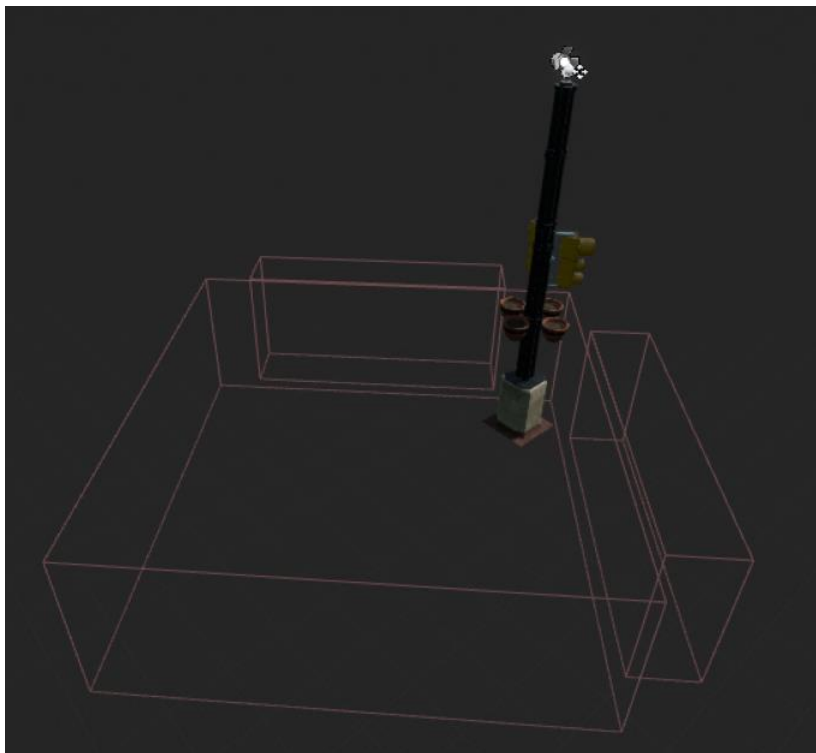
#### a) 차량용 신호등

의도적으로 두 개의 box 를 멀리 분리시켜 속도 감속 구간을 만들고자 하였다. Report 지점에서 차량에게 랜덤한 번호를 부여함과 동시에, changeSpeed 함수를 호출해 감속을 명령한다. 이는 신호등 앞에서 빠른 속도로 주행하는 차량이 급하게 브레이크를 걸었을 때 접촉 사고가 자주 생겼기 때문이다. Stop 지점에서는 빨간불 방향의 차량이 완전히 정지하며, leader 를 설정한다.



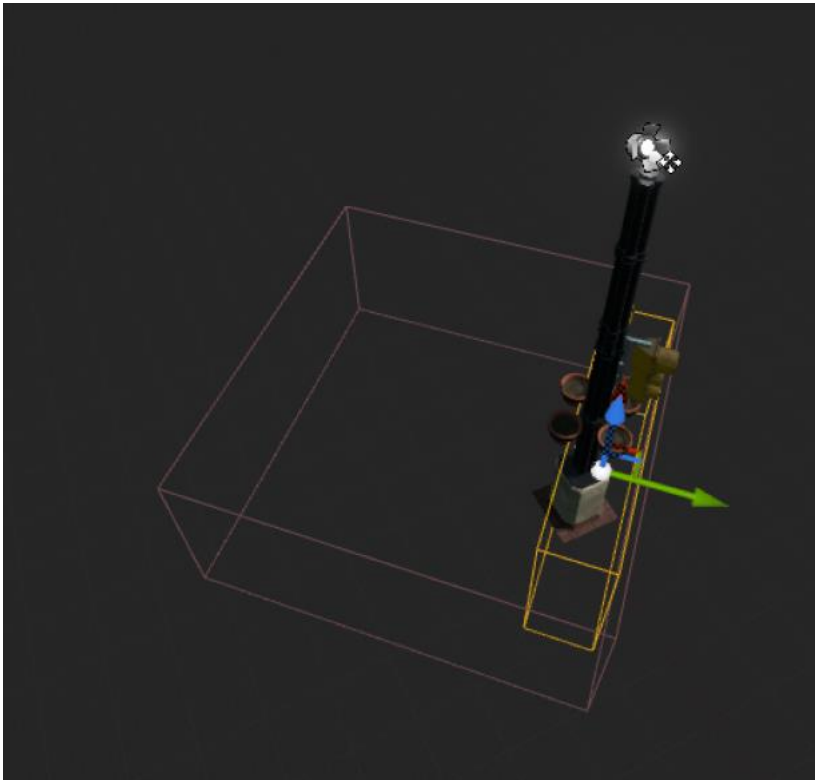
#### b) 보행자용 신호등

큰 collision box 를 통해 랜덤 번호를 부여하고, 작은 box 를 이용해 정지 신호를 보낸다.  
통제하는 횡단보도의 개수에 따라 두 가지 종류로 분류하였다.



횡단보도가 2 개인 신호등





횡단보도가 1 개인 신호등

#### 4) spawn point



일정 시간마다 새로운 차량을 생성한다. 차량은 spawn point 에 변수로 저장된 spline 을 따라 이동하다, 도로와 접촉하며 교통 시스템의 일부로 편입된다. 차량 간의 사고 때문에 줄어드는 차량의 개수를 맞춰주는 역할을 한다.

#### 나. 교차로 위에서의 차량 통제

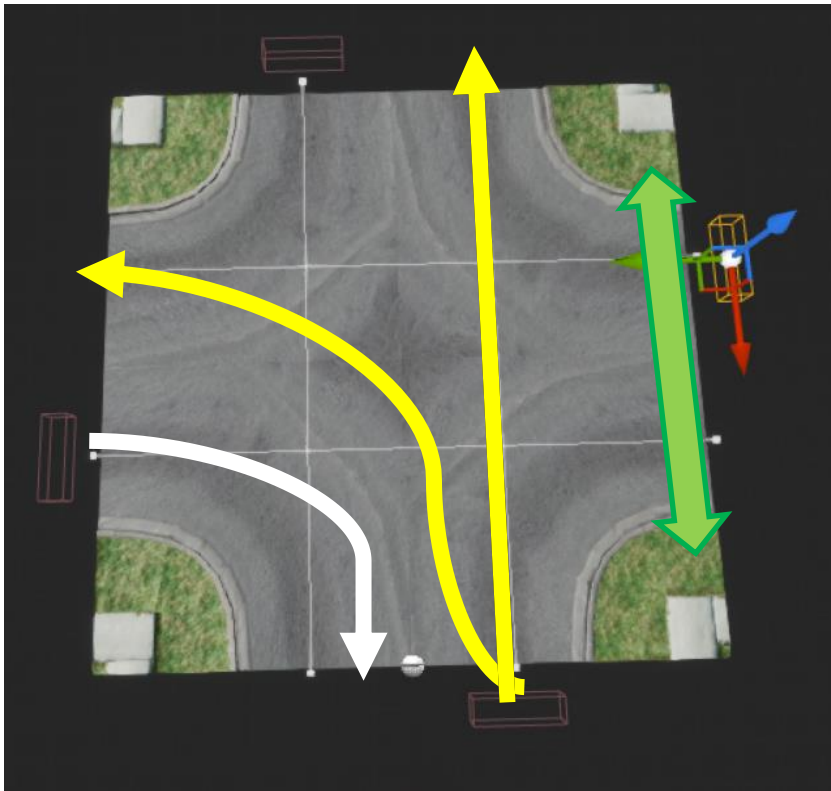


교차로는 신호등을 제어하는 일종의 시퀀스를 가진다. 일정 시간마다 이를 반복하며 각 신호등이 현재 어떤 신호를 보내야 하는지를 제어한다. 신호등은 교차로의 신호를 받아 자신의 상태를 변경하고, 정지하거나 통과할 agent 를 판단한다.

#### 1) 사거리

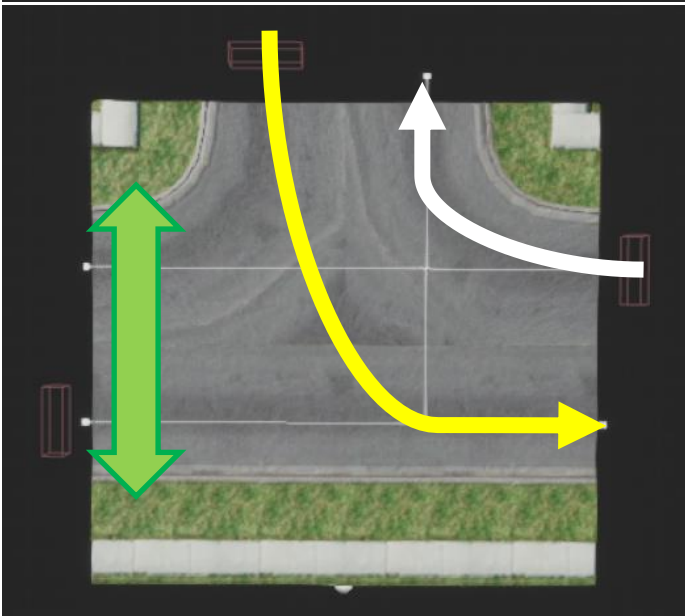
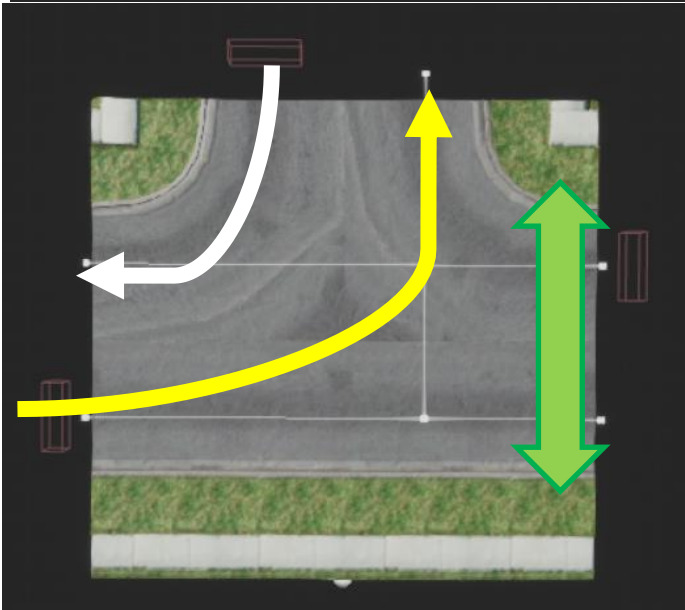
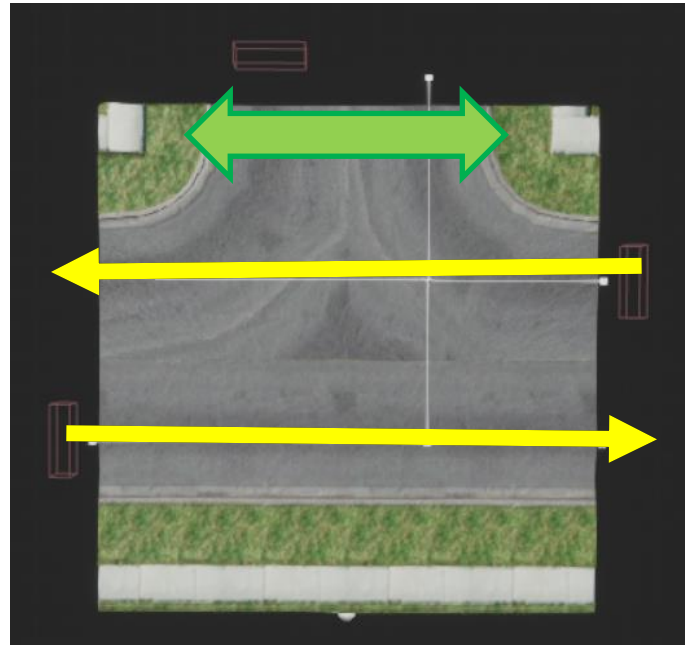
시계방향으로 회전하며 각 차선의 신호등이 한 번씩 직진+좌회전 -> 우회전을 할 수 있도록 배치했다. 즉, 각 신호등은 [직진+좌회전-우회전-정지-정지]의 시퀀스를 번갈아가며 갖게 된다.

현재 직좌 신호등의 왼쪽 신호등이 우회전이 가능하도록 함으로써, 오른쪽 신호등에 차량이 진입하지 않도록 하여 보행자용 신호등(녹색 화살표)을 고려하였다.



#### 2) 삼거리

삼거리의 시퀀스는 다음 3 개의 과정을 가진다. . 마찬가지로 녹색은 보행자가 건널 수 있는 횡단보도를 표시한 것이다.



#### 다. agent 간의 충돌 처리 및 사고구간 통제

모든 도로 class 는 충돌 사고를 처리하는 여러 함수를 공통 부모로부터 상속받는다. 충돌한 agent 는 자신이 위치한 도로에서 이 함수를 호출한다. 함수는 도로의 입구에 StopSign 오브젝트를 생성하여 agent 의 접근을 막는다. 일정 기간마다 충돌한 agent 와 StopSign 오브젝트는 자동적으로 게임에서 제거된다.



#### 라. 최종결과물 주요특징 및 설명

에픽스토어 무료 에셋 [Modular Neighborhood Pack]의 기본 맵을 일부 수정하여 사용하였다.

Spline 과 NavMesh 를 이용해 agent 의 도로를 구성하였다. 또한, 각 삼거리/사거리마다 차량/보행자용 신호등을 배치하여 교통 시스템을 구축하였다.





#### 4. 기대효과 및 활용방안

##### 가. 기대효과

교통 시뮬레이션 내에서 자동적으로 사고 처리가 가능하다. 사고 처리로 사라진 agent 를 계속해서 생성 하면서, 이론적으로 계속해서 버그 없이 작동 가능하다. 또한 이번 연구에서 개발한 에셋은 어떤 scene 위

에서도 작동 가능하므로, 최종 결과물에 사용된 scene 뿐만 아니라 다양한 게임 내에서 유용하게 사용 가능하리라 생각한다.

#### 나. 활용방안

교통 시뮬레이션 측면에서, Agent 에 추가적으로 특징을 부여하여, 다양한 운전자/보행자를 구현하고 교통 제어 상황과 충돌 사고를 시뮬레이션해 볼 수 있다. 도로 구현이 간단하여 여러 종류의 도로를 간단하게 만들 수 있는 점이 장점이 될 것이다.

게임 내에서 적용한다면, npc 들이 일으키는 사고 상황이나 도로 막힘 등을 자동적으로 시뮬레이션하고, 플레이어가 이와 자연스럽게 상호작용할 수 있도록 활용 가능할 것이다.

### 5. 결론 및 제언

이번 연구를 진행하면서 차량과 캐릭터에 대한 이해가 부족함을 깨달을 수 있었다. 차량 시뮬레이션은 만족할 만큼 잘 개발되었지만 보행자 시뮬레이션은 충분치 않았던 것 같다는 아쉬움이 든다. 또한 일차선의 폭이 좁은 도로에서 개발하다 보니 여러 가지 어려움이 있었다. 후속 연구가 가능하다면 넓은 도로를 가진 새로운 scene 위에서 부족한 부분을 보완하고 싶다.

#### 가. 추가 보완/개발점

##### 보행자 시뮬레이션

- 군중 단위의 보행자의 자연스러운 이동
- 신호등 시퀀스 수정
- 변수가 없는 상황에서 보행자 시뮬레이션 통제

##### 2 차선 이상의 도로에서의 라인 변경

- 2 차선 이상의 도로에서 각 차량이 직진, 좌회전, 우회전을 하기 위해 라인을 변경
- 강화학습을 통한 자연스러운 라인 변경
- 불법 운전자들의 차선 끼어들기, 보복운전 등의 특징 개발

#### [참고자료]

Levente Alekszejenkó<sup>1</sup> and Tadeusz Dobrowiecki, "SUMO Based Platform for Cooperative Intelligent Automotive Agents", *EPiC Series in Computing Volume 62*, 2019, Pages 107–123

Seredias, *Unreal Engine AI Car Tutorial*

<https://www.youtube.com/playlist?list=PLSc29g4OjKUNImPwDA4S2B8lr-xX3zpax>

※ 본 양식은 요약보고서이며, 최종결과물을 추가제출 필수

팀 대표 : \_\_\_\_\_ 이륜하

 (인)