

자율주행 상황 속 환경 객체와 상호작용하는 모빌리티 AR 콘텐츠 개발 방법 연구

박유천, 경병표
국립공주대학교 게임디자인학과
e-mail : 6000@yucheon.io, kyungbp@kongju.ac.kr

A Study on the Development Method of Mobility AR Content Interacting with Environmental Objects in Autonomous Driving Situations

Park Yucheon, Kyung Byungpyo
Dept. of Game Design, Kongju National University

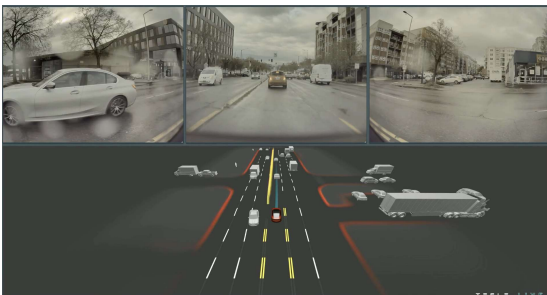
요 약

본 연구는 레벨 4 자율주행 상황에서 외부의 다양한 환경 객체와 상호작용하는 모빌리티 AR 콘텐츠를 제작할 수 있는 개발 환경의 구조와 개발 방법을 제안한다. Unity 엔진을 활용하여 자율주행 시뮬레이터를 제작하고, 구분된 Scene과 API를 기반으로 모빌리티 AR 콘텐츠 개발 환경을 구축하였다. 이를 통해 콘텐츠 제작자는 Unity 엔진으로 AR 콘텐츠를 쉽게 구현할 수 있으며, 콘텐츠 소비자에게 몰입감 있는 AR 경험을 제공할 것으로 기대한다.

1. 서 론

자율주행 기술의 발전으로 많은 완성차 기업에서는 차 내에서 어떻게 콘텐츠를 제공할지에 대한 다양한 콘셉트를 제시하고 있다. 특히 레벨 4 자율주행에서는 시스템이 운영설계영역(ODD) 내의 모든 동적주행작업을 수행하고, 모든 위험 상황에 대비하여 모든 결정을 내리기에[1], 운전자에게 어떠한 제어도 요구되지 않는다. 이로 인해 자동차 내에서 영화, 게임 등의 콘텐츠 소비도 가능하다.

Tesla는 자율주행을 위해 주행환경을 3D 월드로 변환하는 기술을 개발하였다. 해당 기술과 발전된 AR Heads up display(이하 AR HUD)의 융합으로 자동차 밖의 요소들과 상호작용을 하는 AR 콘텐츠가 전면 유리를 통해 제공될 수 있고, 이러한 콘텐츠를 ‘모빌리티 AR 콘텐츠’라 정의한다.



(그림 1) 주행환경을 인식하여 3D 월드 생성 (Tesla, 2022) [2]

본 논문에서는 미래의 모빌리티 AR 콘텐츠(특히 게임) 제작자가 자율주행 자동차 시스템이 제공하는 정보를 가지고 어떻게 콘텐츠를 개발할 수 있는지 구조를 설계, 개발하고 이를 설명한다.

2. 모빌리티 AR 콘텐츠 제작 환경 구축

2.1 용어 정의

용어	의 미
실제 환경	본 연구의 Unity 시뮬레이션 환경이 아닌 자율주행이 이뤄지는(또는 이뤄질) 실제 환경
환경 객체	주행 환경에서 존재하는 자동차, 건물 등
사용자	자율주행 자동차에서 AR 콘텐츠를 소비하는 사람
사용자 자동차	사용자가 탑승한 자율주행 자동차
콘텐츠 제작자	Unity 엔진을 활용하여 모빌리티 AR 콘텐츠를 제작하는 사람

2.1 시뮬레이션 환경

실제 환경의 자율주행 자동차는 카메라, LiDAR 등의 센서를 활용하여 환경 객체를 검출한다. 본 연구에서는 Unity 엔진을 사용하여 자율주행 환경 시뮬레이터를 제작하였고, 이는 *Environment Scene*에서 처리된다. 자동차, 건물과 같은 환경 객체는 *EnvironmentalObject*(이하 *EnvObj*)로 구현되며, 각 객체의 크기에 맞는 *BoxCollider*를 가진다. 사용자 자동차에는 *SphereCollider*를 추가하여, 충돌한 *EnvObj*를 통해 외부의 환경 객체를 검출한다.

*SphereCollider*의 크기는 실제 환경에서 센서를 통한 환경 객체 검출 가능 범위를 의미한다.

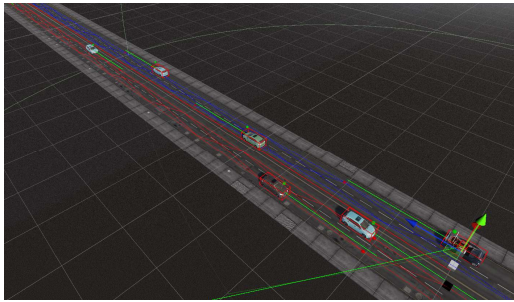
2.2 검출된 세상

검출된 환경 객체는 *Detected World Scene*에서 *DetectedObject*(이하 *DetObj*)로 생성되어 관리되며, 각 *DetObj*는 대응하는 *EnvObj*의 크기와 위치 정보를 포함하고 계속 업데이트한다. 만약 사용자 자동차의 검출 범위를 벗어났을 경우, 해당되는 *DetObj*가 삭제된다.

추가로 사용자 자동차의 위치, 자동차 내 사용자의 머리

및 손 위치 등도 *DetObj*로 제공할 수 있다.

콘텐츠 제작자는 *Detected World Scene* 안에서 자유롭게 Unity를 활용한 개발을 진행할 수 있다.



(그림 2) 검출된 Detected Object (빨간색 박스 테두리 시각화)

2.3 이벤트 API 등을 통한 데이터 획득

사용자 자동차의 객체 검출 시스템(*DetectionManager*)은 관찰자 패턴을 사용하여, *EnvObj* 검출 시작 및 종료 시, *DetObj* 정보를 각 관찰자에게 이벤트로 전달한다. 콘텐츠 제작자는 이를 활용하여 AR 객체를 배치할 수 있다.

```
// 객체 검출 시스템(DetectionManager)에서 새로운 검출된 객체가 생성되었을 때 이벤트를 받는다.
DetectionManager.OnDetectedObjectCreated.AddListener((detectedObject, _) =>
{
    // 몬스터를 생성 위치
    // (자동차 크기의 y만큼 더해서 차 루프 위에 몬스터가 생성될 수 있도록 한다.)
    Vector3 pos = detectedObject.Position; // Detected Object의 pivot은 자동차 최하단 가운데
    pos.y += detectedObject.Scale.y;

    // 몬스터 게임 오브젝트 생성
    GameObject clone = Instantiate(enemyPrefab, pos, Quaternion.identity);

    // 차가 이동해도 몬스터가 차 루프 위에 계속 서 있도록
    // 몬스터의 부모로 Detected Object를 지정한다.
    clone.transform.SetParent(detectedObject.transform);
});
```

(그림 3) 새로 검출된 자동차 위에 AR 몬스터를 스폰하는 스크립트

추가로 사용자 자동차의 속도(*DetectionManager.UserCarSpeed*) 등 변수 형태의 정보도 활용할 수 있다.

2.4 실제 환경에서의 구현

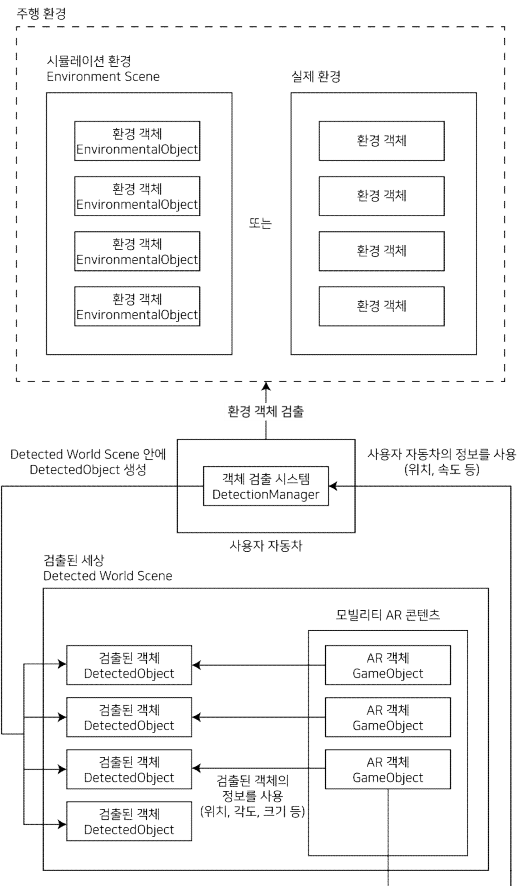
AR HUD로 콘텐츠를 구동할 때, 사용자의 시선과 일치하는 위치에 정확히 AR 그래픽을 출력해야 한다. 이를 위해 Head/Eye Tracking으로 렌더 카메라의 위치를 보정하고, 이후 *Detected World Scene*에서 AR 객체만 렌더링해 AR HUD에 출력한다.

3. 결 과



(그림 4) 자동차 위에 몬스터가 생성된 결과

2.3 과정으로 몬스터가 주행 중인 자동차의 *DetObj* 위에 생성되어, 자동차 위에 올라타 있는 것처럼 보인다.



(그림 5) 도식화한 전체 제작 환경 구조

4. 결 론

본 연구는 모빌리티 AR 콘텐츠의 제작을 위한 개발 환경과 방법을 제안하였다. 이를 통해 손발이 자유로운 자율주행 환경에서 지속적 운동을 돕는 체감형 게임[3] 등을 AR로 제작하여 이동 중에 플레이할 수 있을 것으로 기대한다.

본 연구는 모빌리티 AR 콘텐츠를 위한 개발 환경의 설계 및 구축에 중점을 두었으며, 향후 연구에서는 이 개발 환경을 활용하여 실제 콘텐츠(게임)를 직접 구현하고, 다양한 응용 가능성에 대해 탐색할 계획이다.

참고 문헌

- [1] 이광구, 용부중, 우현구. “레벨 4 자율주행자동차의 기능과 특성 연구”, 『자동차안전학회지』 제12권 제4호, 2020. pp.61-69.
- [2] Tesla. (2022, Oct 1). Tesla AI Day 2022 [Video]. https://www.youtube.com/watch?v=ODSJsviD_SU. (2024년 9월 12일 기준)
- [3] 정석호, 고정운, 허설화, 경병표. “사용자의 헬스케어를 위한 AR기반 체감형 게임 콘텐츠 연구”. 『한국게임학회 논문지』 제18권 제4호, 2018. pp.53-63.