

VR 기반의 급발진 대응 시뮬레이션

VR/AR의 개론 및 실습 프로젝트

Team 4 급발진
건설환경공학부 최지웅
재료공학부 최현식
전기정보공학부 홍선우



서울대학교
SEOUL NATIONAL UNIVERSITY

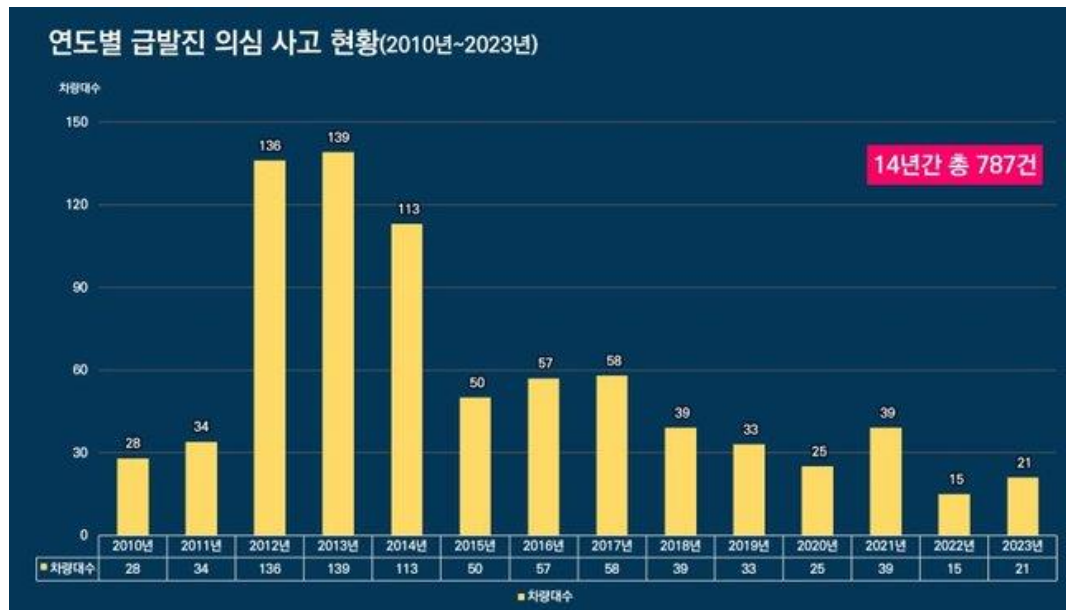
Table of Contents

- I. 프로젝트 목표
- II. 수행 내용
- III. 수행 결과
- IV. 기대 효과 및 수행 후기

Goal of the Project

Background

- 급발진 사고 대응 교육의 필요성
 - 자동차의 주요 기능들이 컴퓨터와 결합, 전자화 → 급발진 사고 꾸준히 발생



멈추지 않는 급발진 의심사고, '제자리걸음'인 국토부, 하목형 기자, 교통일보, 2023-10-10

Background

차량에 대한
제어 상실

큰 인명사고
위험

신속한 조치
매우 중요

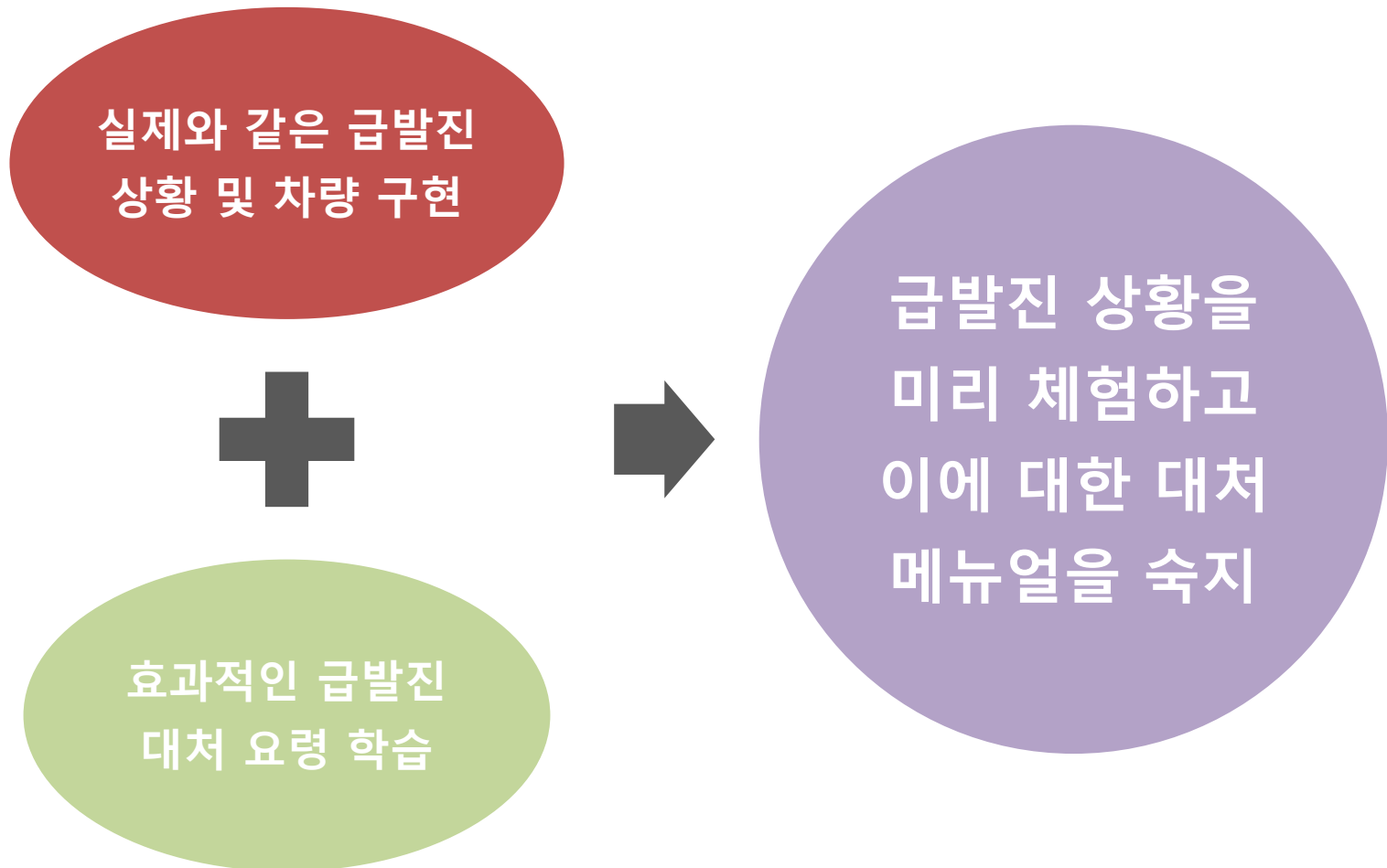
급발진 발생 시
대응 조치

실제적인 교육
및 훈련이 절실

왜 VR 인가?

- 급발진
 - 차량에 탑승한 상황에서 미리 체험 불가
 - 대처 요령 또한 평상시에 연습하기 어려움
- VR
 - 실제와 같은 몰입감 있는 상황을 구현
 - 경각심을 통해 상황에 대한 위험 인지
 - 급발진 상황 체험 가능
 - 대응 메뉴얼을 효과적으로 학습 가능

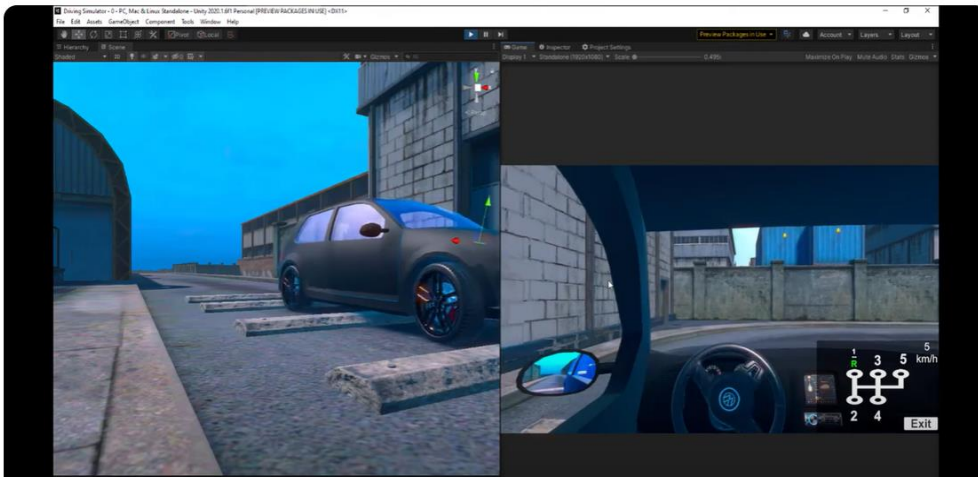
Project Goals



Implemented Contents

'Unity Asset' Exploration

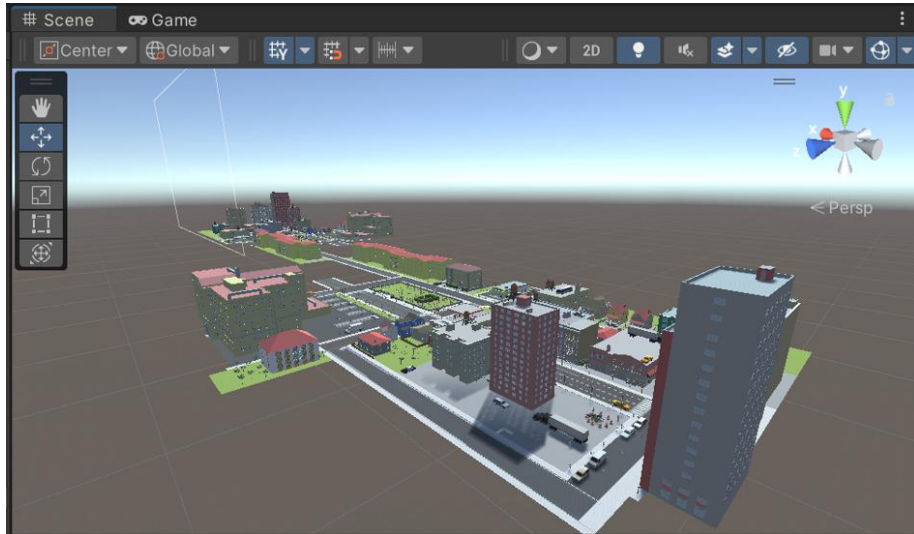
- 자동차 에셋
 - 1인칭 운전용으로 사용 가능
 - 차량 내부 실제와 같은 구현에 주안점
 - "Driving_Simulator_URP_PS4Controller" 채택
 - 자동차의 physics가 실제와 같이 구현
 - Oculus 기반이 아닌, PS4 기반



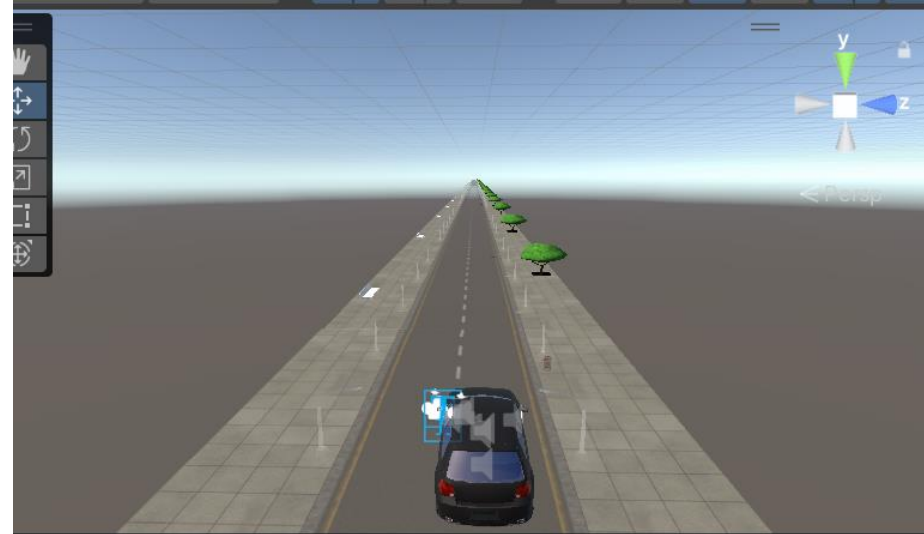
https://github.com/Viktoreznov/Driving_Simulator_URP_PS4Controller

'Unity Asset' Exploration

- 도시 배경 에셋
 - 직선 구간이 길고 많게
 - 무겁지 않은 그래픽



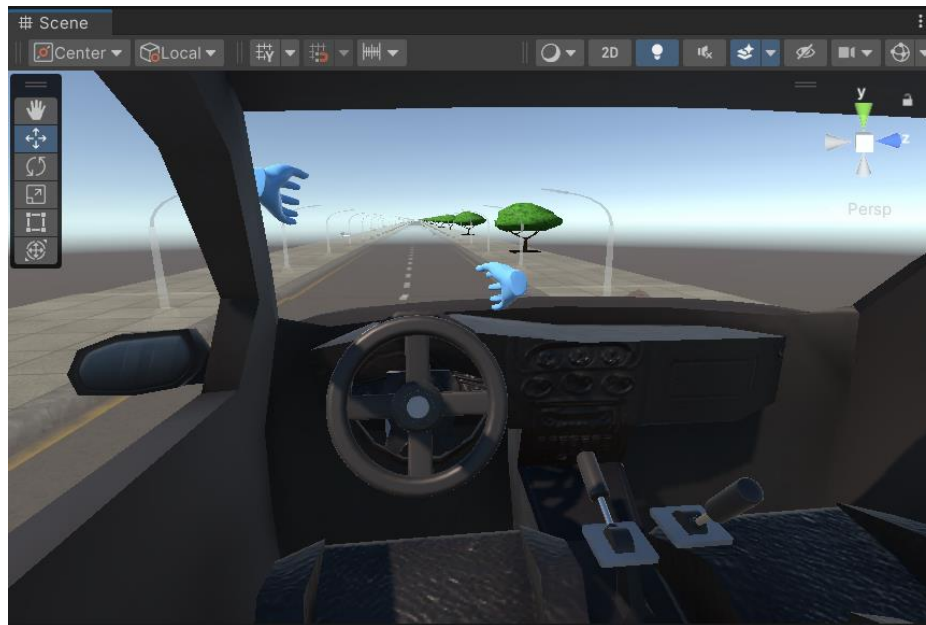
< 도시 구현 >



< 도로 구현 >

'Unity Asset' Exploration

- 핸들, 레버 추가
 - XR Interaction Toolkit에서 제공하는 핸들과 레버 (기어 변속기, 사이드 브레이크) 사용



< 핸들 화면 >

Car Implementation

- 핸들, 레버 추가
 - 기존의 자동차 에셋에 핸들과 레버를 부착한 결과, 구현되어 있는 차량의 수많은 Collider와 추가된 핸들, 레버의 Collider가 충돌
 - 자동차가 평면상에서 움직이지 못하고, 회전하거나 불규칙한 상하 움직임을 보였다

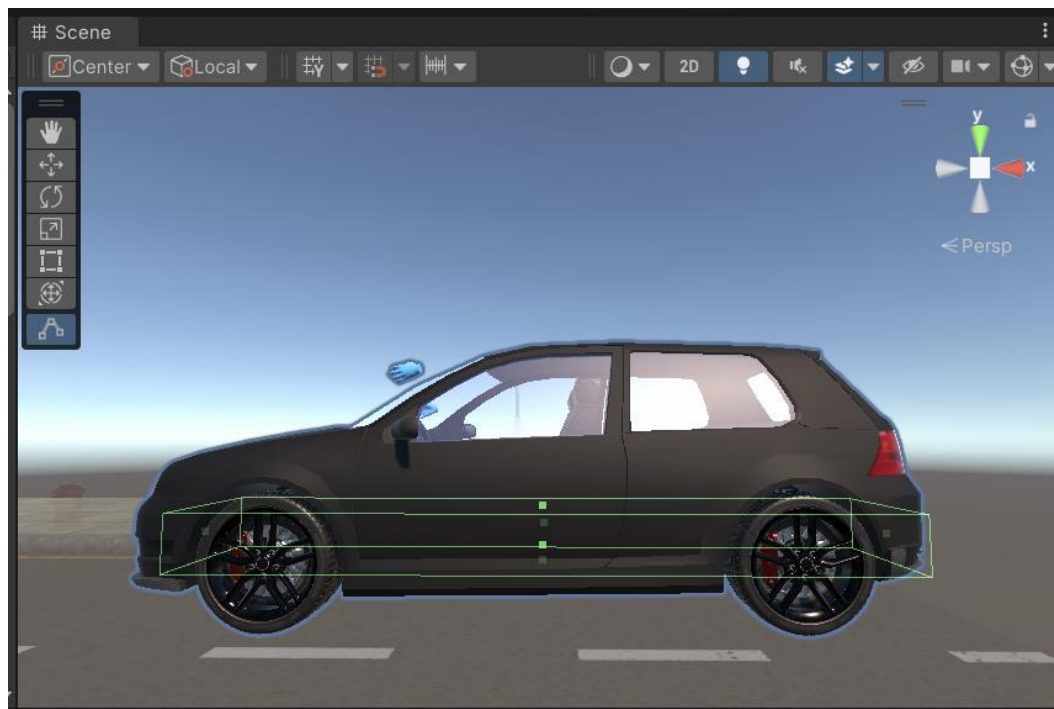
Car Implementation



< 자동차 개발 과정 >

Car Implementation

- 핸들, 레버 추가
 - 바퀴를 제외한 차체의 mesh collider를 모두 제거
 - 추가할 핸들과 레버에 충돌하지 않도록 차의 아래쪽에 높이가 작은 직육면체의 collider 하나로 대체



Implementation of Car Control

- 핸들의 구현
 - 핸들의 값을 통해 회전 값을 설정하고, 현재 속도만큼 그 방향으로 자동차의 위치를 translation
 - XRKnob로 정의된 핸들의 값은 0에서 1 사이의 float 값으로 반환 → Mathf.Lerp를 통해 -1과 1 사이의 값으로 변환
 - turnSpeed와 시간의 곱으로 차의 회전 각도를 구현

```
// Update is called once per frame
// Unity 메시지 참조 0개
void Update()
{
    float turnInput = knob.value;
    float turnAngle = Mathf.Lerp(-1.0f, 1.0f, turnInput) * turnSpeed * Time.deltaTime;

    turnAngle = Mathf.Clamp(turnAngle, -90.0f, 90.0f);

    // 새로운 회전을 적용
    transform.Rotate(Vector3.up, turnAngle);

    // Check if both A and X buttons are pressed
```

< Unity Script >

Implementation of Car Control

- 기어 변속기와 사이드 브레이크 구현
 - 기어는 D와 N만 구현
 - N으로 기어를 변속 → 차의 가속이 stop
 - lever의 value에 따라 가속도 값의 추가 여부 결정하는 스크립트 작성
 - 사이드 브레이크의 경우, 비슷한 logic으로 lever의 value에 따라 속도를 감속시키는 스크립트 작성

Implementation of Car Control

```
else // 속도가 0 이하이면 멈추도록 설정
{
    currentSpeed = 0.0f;
    if (audioSource.isPlaying)
    {
        audioSource.Stop();
    }
}

// 가속도 적용
currentSpeed += acceleration * Time.deltaTime * (lever.value ? 0 : 1);
// 사이드브레이크 적용
currentSpeed -= sidebrakeForce * (lever2.value ? 0 : 1) * Time.deltaTime;

// 전진 이동
transform.Translate(Vector3.forward * currentSpeed * Time.deltaTime);
//Debug.Log("Moving Forward");

// 핸들 돌린 각도만큼 차량을 회전시킴
float rotationAmount = turnInput * turnSpeed * Time.deltaTime;
transform.Rotate(Vector3.up, rotationAmount);
}
```

< 자동차의 Unity Script >

Implementation of Car Control

- 브레이크 구현
 - Oculus 2의 한계점은 손, 팔, 얼굴을 제외한 신체 부위에서는 값을 불러오지 못한다는 점
 - 브레이크 → 양손 컨트롤러의 primary button (A & X button) hold!



< Oculus quest2의 Controller >

Implementation of Car Control

- 브레이크 구현
 - primary button이 hold 될 때 true의 value를 반환하게끔 새로운 XR Interaction을 정의하고, 스크립트 작성

```
// Start is called before the first frame update
// Unity 메시지 | 참조 0개
void Start()
{
    // Brake_R 스크립트를 찾아서 할당
    brakeScript1 = FindObjectOfType<Brake_R>();
    brakeScript2 = FindObjectOfType<brake>();

    audioSource = GetComponent<AudioSource>();
    currentSpeed = forwardSpeed;
    Debug.Log("void Start finished");
    audioSource.clip = accelerationSound;
}
```

```
if (brakeScript1.val && brakeScript2.val) //브레이크를 받았을 때
{
    if (audioSource.clip != brakeSound)
    {
        audioSource.Stop();

        audioSource.clip = brakeSound;

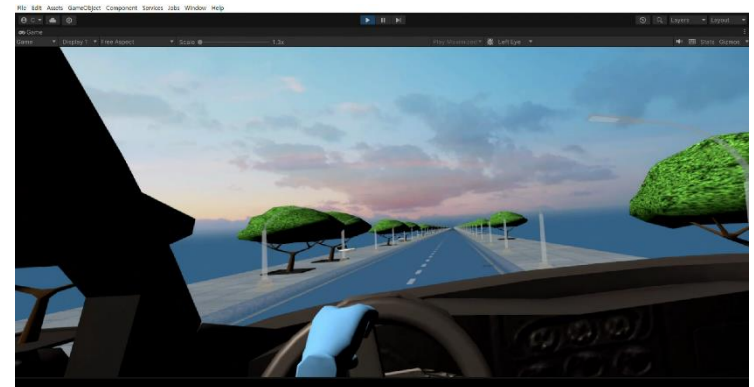
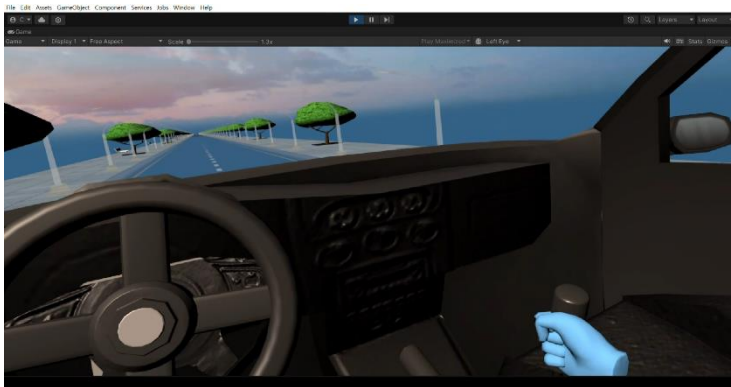
        //audioSource.PlayOneShot(accelerationSound, desiredVolume);
        //audioSource.PlayOneShot(brakeSound, desiredBrakeVolume);
    }

    currentSpeed -= brakeForce * Time.deltaTime;
}
```

< Brake의 Unity Script >

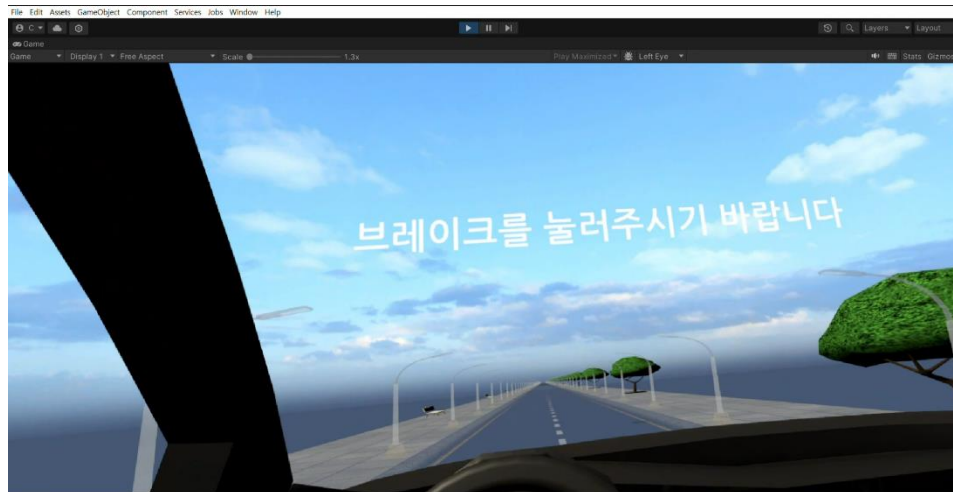
Implementation of Car Control

- 소리 효과 추가
 - 실제와 같은 몰입감 제공 → 다양한 음원 추가
 - 자동차 주행 중 발생하는 ambient sound를 계속 재생
 - 급발진 시작 → 기어가 D일 때는 가속하는 엔진의 소리 지속적으로 재생
 - 브레이크 작동 → 차량의 속도에 비례하는 음량으로 브레이크 소리 추가
 - 차의 속도가 0 미만으로 계산 → 차량의 속도 0으로 초기화하여 정지 / 소리 재생 중지



Implementation of Manual Scene

- 급발진 대응의 3단계를 차에서 영상을 시청하듯이 교육
- 매뉴얼이 순서에 따라 나오고, 사용자가 각 단계에 브레이크 밟기, 기어 변속, 사이드 브레이크 잠그기를 직접 따라할 수 있음
- 차의 속도와 가속도를 낮추어 사용자가 교육 내용에 더욱 집중할 수 있도록 함

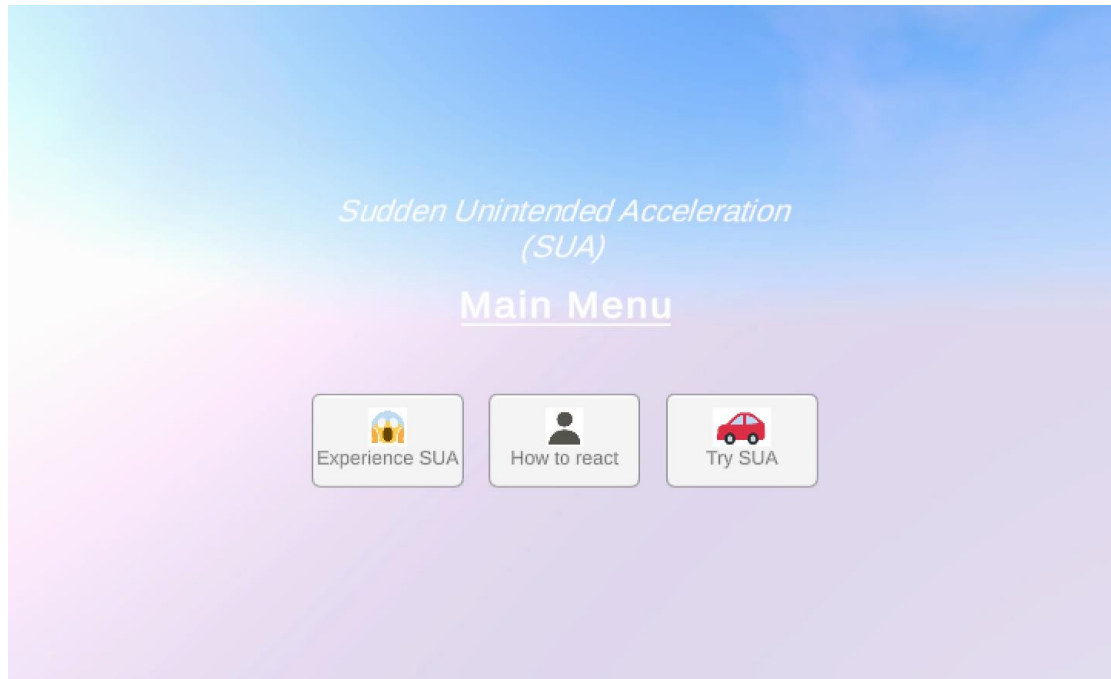


< 매뉴얼 scene >

Implementation of Main Menu Scene

- 메인 메뉴 화면

- Ray Interactor로 버튼 클릭
- 각 scene으로 이동 가능



Implementation of 'Back to Main Menu UI'

- 각 scene 내 메인 메뉴 이동 UI
 - 왼쪽 컨트롤러 Y버튼 클릭 시 안내 문구 나옴
 - 오른쪽 컨트롤러 B 버튼 클릭 시 메인 메뉴 scene으로 이동



< Back to Main >

Project Result

Content Overview

초기 화면

- 급발진 체험, 메뉴얼 숙지와 급발진 대처 scene 이동 선택

급발진 체험

- 급발진 상황 체험, 위험성과 대처방안 학습 중요성 확인

메뉴얼 숙지

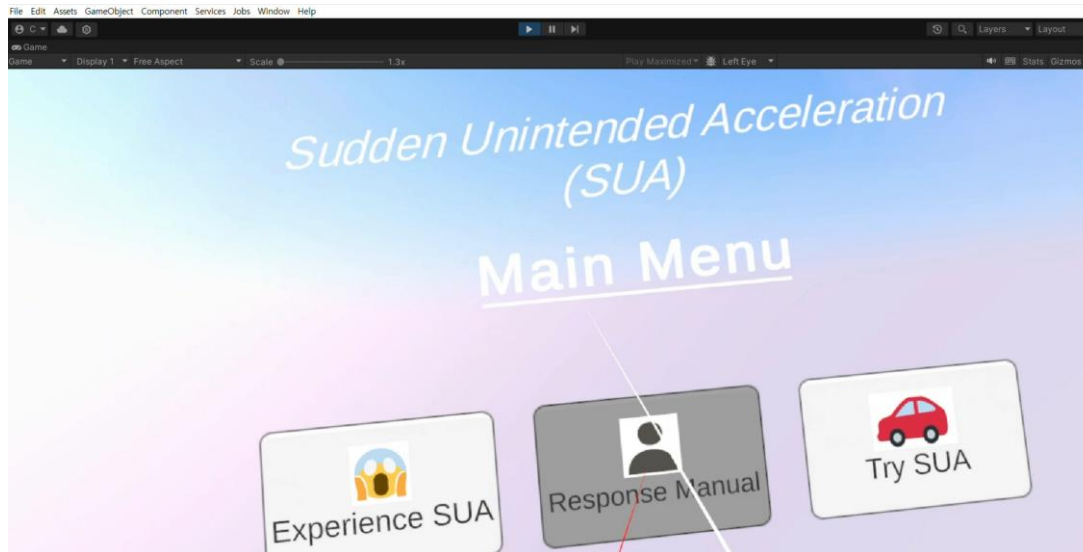
- 급발진 대응 메뉴얼 교육, 숙지 과정

급발진 대처

- 실제 급발진 상황에서 운전하는 차량을 멈추는 시뮬레이션 진행

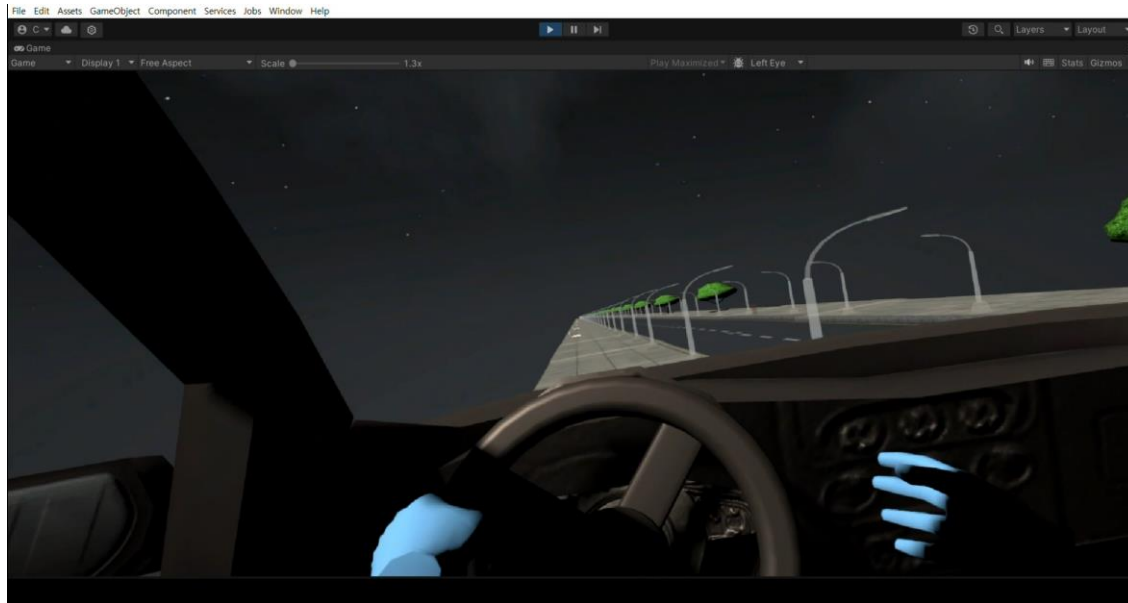
Starting Scene

- 메인 메뉴 UI가 존재하는 scene
- 1) 급발진 체험/2) 급발진 대처 매뉴얼/3) 급발진 대처 scene으로 이동할 수 있게 scene이 전환된다.
- 사용자는 Ray Interactor를 활용하여 UI 내 버튼과 상호작용하고 클릭을 할 수 있다.



SUA(Sudden Unintended Acceleration) Experience Scene

- 사용자가 급발진 상황을 체험
- 급발진 대처에 대한 어려움을 느끼게 함



< SUA Experience Scene >

Manual Scene

- 사용자가 급발진 상황에 대한 매뉴얼을 교육 받는다
- 직접 각 단계를 따라하며 직접 숙지할 수 있도록 함
- 차량이 천천히 구동하여 교육에 더욱 집중할 수 있도록 함



< Manual Scene >

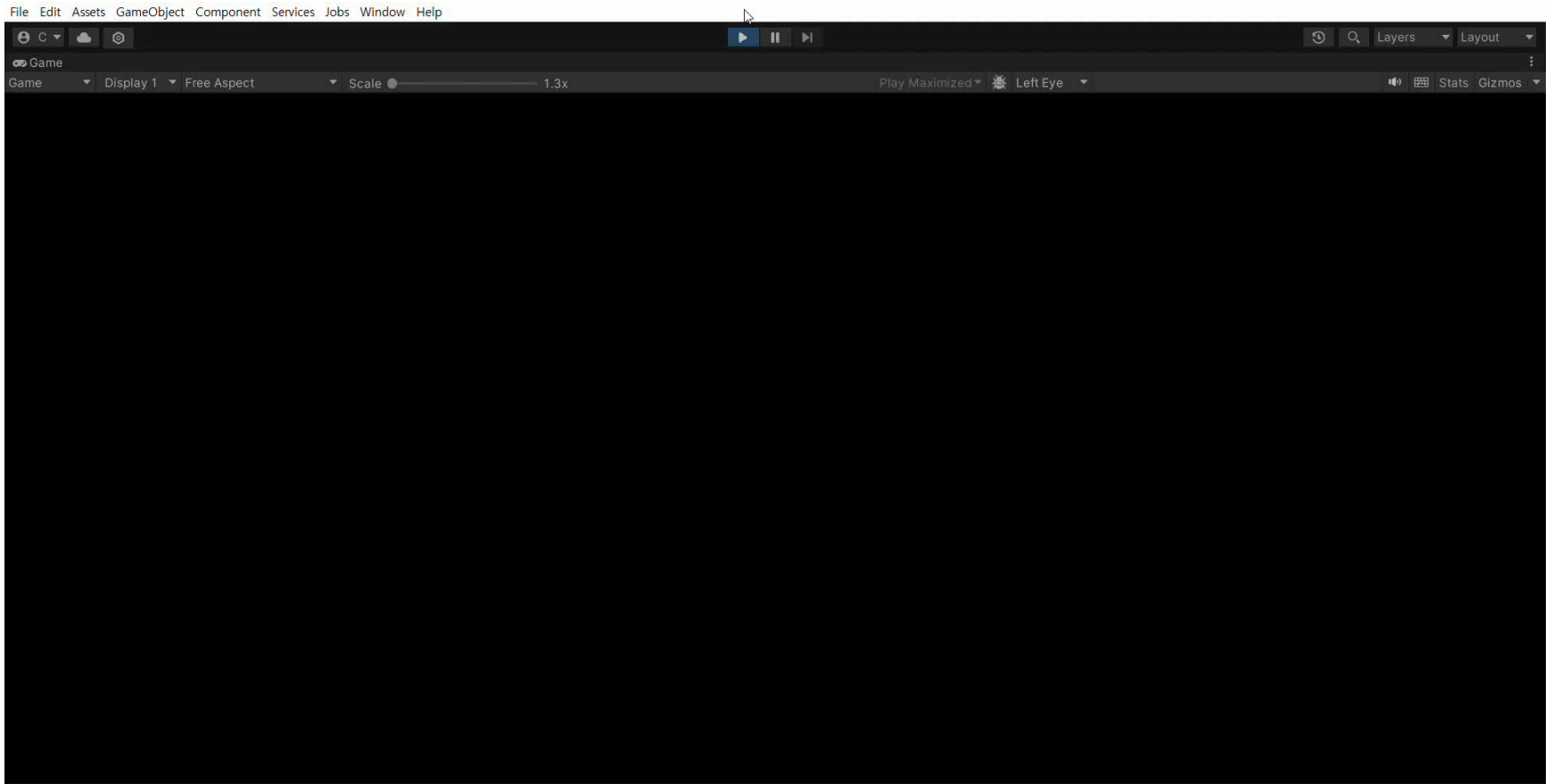
SUA(Sudden Unintended Acceleration) Coping Practice Scene

- 초기에 낮은 속도로 운행 시작
- 엔진 속도가 커지고, 가속도가 붙으면서 차량이 빨라지게 된다
- 숙지된 메뉴얼대로 1) 브레이크 작동, 2) 중립 기어 변속, 3) 사이드 브레이크 작동 순으로 대응 하면, 차는 금방 멈춘다
- 주어진 메뉴얼대로 실행하지 않더라도 본 시뮬레이션은 계속됨
- 반복적 학습을 통해 급발진 상황 시 대응 능력이 향상될 것



< SUA Coping Practice Scene >

Project Goals



< 소리 효과, 자동차 컨트롤 구현이 완료된 시연 영상 >

Expected Effects & Project Review

Expected Effects

- 급발진 사고는 대응 요령만 정확히 파악하고 신속하게 대처할 경우, 피해를 크게 줄일 수 있다
- 본 VR 콘텐츠를 급발진 사고 대응 교육 자료로 적극적으로 활용한다면, 급발진 사고로 인한 피해를 예방할 수 있을 것으로 기대



Improvements

- 더 실제와 같은 자동차 구현
 - P로 기어를 변속 → 핸들이 잠기는 문제
 - 사이드 브레이크를 갑자기 당김 → 차가 회전하는 위험한 상황 발생
- 더 실제와 같은 도시 환경 구축
 - 보행자, 차량 등을 추가
 - 정체 시, 주행 시 등 다양한 상황에서의 급발진 매뉴얼을 숙지

Improvements

- 평가 방식 도입
 - 정차까지 걸린 시간, 최고 속도, 충돌 여부 등의 다양한 지표를 통해 사용자의 급발진 대처를 평가하여 피드백
 - 위기 대처 능력을 증빙하는 평가 자료로도 활용 가능



Project Reviews

- 자동차 구현의 어려움
- VR 시장의 성장을 위해서는...
 - 개발자에게 더 친숙한 개발 플랫폼
 - 다양한 오픈 소스 에셋의 보편화



감사합니다 ☺