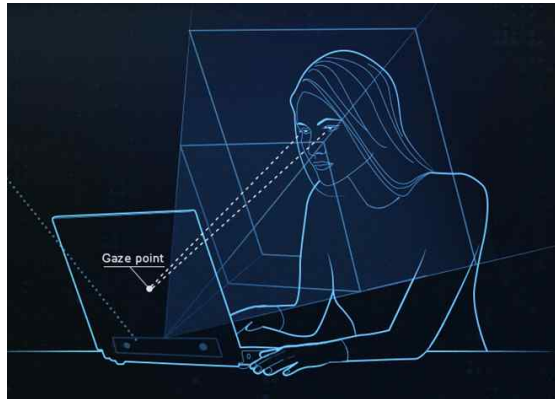


# CarDynavision\_TOBII

- 차량을 사용한 Dynavision 콘텐츠
- Unity3D로 구현
- 하드웨어

## 1. Tobii Eyetracker 4C



- 시선 추적 장치
- gaze & HeadPose를 가져올 수 있음
- 모니터에 장착

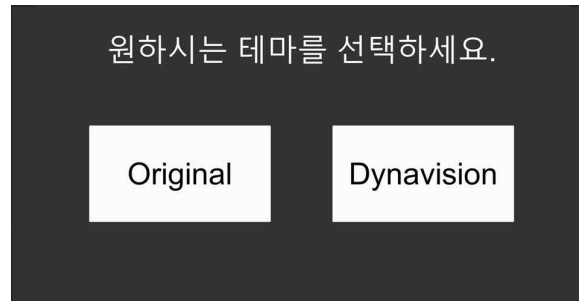
## 2. Logitech Steering Wheel G29



- Steering Wheel, 페달(엑셀,브레이크), 기어를 통해 자동차 운전 시뮬 구현
- 유니티와 연동하여 유니티 오브젝트와 장치 움직임을 동일시

- 화면구성

1. 메인화면



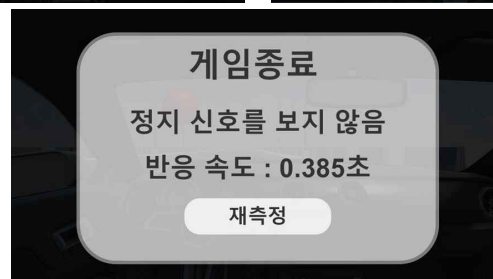
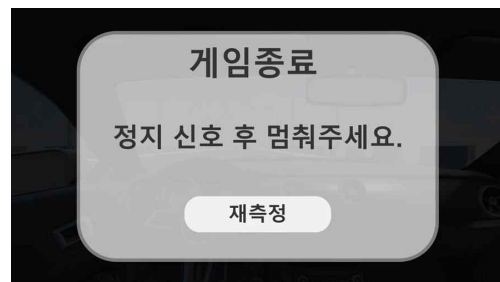
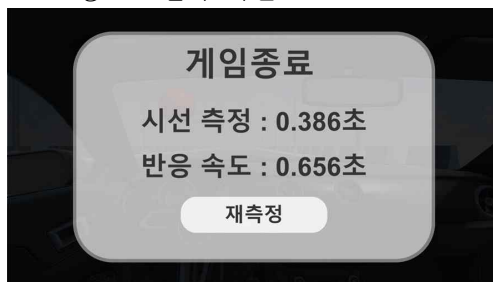
- 테마 선택 화면
- 총 2가지 버튼으로 구성 - Original / Dynavision

2. Original Scene



- 주행 중 랜덤 시간에 빨간 구 생성
- 생성 후부터 시선, 브레이크 반응 시간 측정

3. Original 결과 화면



- 시선, 브레이크 반응 시 결과 화면
- 시선 반응 속도, 브레이크 반응 속도 출력
- 1. 정상 반응, 2. 구 생성 전 정지, 3. 구를 보지 않고 정지

- 화면구성

#### 4. Dynavision Scene



- 차량 기준 속도 60km/h로 지정
- 60km/h를 넘으면 차량의 창문에서 위치 랜덤 점등 이벤트 발생
- 랜덤으로 켜지는 공의 총 개수 : 69개
- 응시할 시 / 5초 내에 응시하지 않았을 시 이벤트 종료
- 다시 60km/h를 넘으면 이벤트 발생
- 총 12번 반복 시 게임 종료
- 사이드 미러, 룸미러
  - Camera & RenderTexture 사용하여 구현, Script로 거울효과

#### 5. Dynavision 결과 화면



- 총 2가지로 구성 - 인지에 대한 시선 반응시간 / 인지 후 브레이크 반응시간
- 12회 각각의 반응 시간 , 총 평균 반응 시간 데이터를 보여줌
- 버튼을 통해 두 결과화면을 이동할 수 있게 구현

#### 6. Dynavision CSV 출력

|                  | A     | B             | C          | D   | E       |
|------------------|-------|---------------|------------|-----|---------|
| Car_Result.csv - | Stage | Response_Time | Brake_Time | O_X | BallNum |

- 각 단계, 시선 반응 시간, 브레이크 반응 시간, O\_X여부, 공의 번호

#### Saved\_Car\_Data.csv

| A    | B             | C             | D          | E          | F          | G          | H       |
|------|---------------|---------------|------------|------------|------------|------------|---------|
| Time | EyeLocation_x | EyeLocation_y | HeadPose_x | HeadPose_y | HeadPose_z | CheckPoint | LightOn |

- 게임 시간, 시선 x 좌표, 시선 y 좌표
- HeadPose x 값, HeadPose y 값, HeadPose z 값
- 공이 켜진 포인트에 공 번호, 공을 본 포인트에 공 번호

- Assets

1. Scene 구성

이름

Select.unity  
Original.unity  
Dynavision.unity

- 총 3개의 Scene으로 구성 - Select, Original, Dynavision

2. Scripts 구성 (사용하는 주요 스크립트)

이름

|                  |                          |
|------------------|--------------------------|
| Car              | CarUserController.cs     |
| SceneChange.cs   | CarController.cs         |
| GazeEvent.cs     | LogitechSteeringWheel.cs |
| GameManager.cs   |                          |
| DynaGazeEvent.cs |                          |
| CSVReader.cs     | HeadMovement.cs          |
| CarMirror.cs     | GazePlotter.cs           |
| Car_Data.cs      |                          |

- SceneChange.cs - 씬 관리
- GazeEvent.cs - Original 시선 이벤트 관리
- GameManager.cs - 전체 게임 관리
- DynaGazeEvent.cs - Dynavision 시선 이벤트 관리
- CSVReader.cs - CSV 입력 관리
- CarMirror.cs - 사이드 미러, 룸미러 좌우반전 효과 관리
- Car\_Data.cs - CSV 항목 관리
  
- CarUserController.cs - 하드웨어 변화에 맞춰서 차량 오브젝트 변화 관리
- CarController.cs - Standard Asset에서 제공, 차량 전체 관리 (속도,토크 등)
- LogitechSteeringWheel.cs - Logitech 장치 데이터 관리
  
- HeadMovement.cs - Tobii eyetracker 4C를 통한 HeadPose 값 출력 관리
- GazePlotter.cs - Tobii에서 제공하는 Unity Gaze 오브젝트 전체 설정 관리