

Document

Eye Tracking을 활용한 눈빛보내기 VR 재구성



과목명

가상및증강현실프로그래밍

담당교수

유용재

학생이름

정현욱, 곽도훈, 류시우

학과

인공지능

팀명

zeeeing

HANYANG UNIVERSITY

1. 프로젝트 개요와 소개

눈빛 보내기 VR (Gaze VR)

Eye Tracking VR Romance Simulation Game

Meta Quest Pro의 Eye Tracking 기술을 활용한 VR 시뮬레이션 게임입니다.

프로젝트 소개

고전 플래시 게임 '눈빛 보내기'를 VR 환경으로 현대적 재해석한 프로젝트입니다. Meta Quest Pro의 Eye Tracking API를 핵심 메커니즘으로 활용하여, 사용자의 실제 시선을 통한 감정적 상호작용과 게임플레이를 구현했습니다.

주요 특징

- Eye Tracking 기반 상호작용: 실제 시선으로 NPC와 감정 교류
- 감정-시선 매핑 시스템: 4가지 감정 상태와 시선 패턴 연동
- 미니게임: Color Gaze, Heart Gaze 등 시선 기반 게임
- 피버 모드: 특별한 시각 효과와 점수 부스트
- 몰입감 있는 학교 환경: 교실, 복도, 도서관 등 다양한 공간

2. 각자 역할에 대한 Ownership

이름	역할	담당 업무
정현 욱	Eye Tracking & Minigame	프로젝트 관리 및 문서화 <ul style="list-style-type: none">• 제안서 작성 및 발표• 중간 보고서 작성• 기술 문서 및 Readme 작성• 판넬 제작 디자인• 캐릭터 에셋 제작• 엔딩씬 제작
		게임 시스템 개발 및 구현 <ul style="list-style-type: none">• Eye Tracking Interaction 구현• Player Locomotion 구현• 게임 시스템 구현• 피버 모드 구현• 미니게임 구현 (ColorGaze, HeartGaze)• 게임-VFX 연동
		사운드 및 햅틱효과 <ul style="list-style-type: none">• Laser 제외 bgm3개, SoundEffect 2개 탐색 및 도입• Haptic Effect 도입
		UI/UX 및 최적화 <ul style="list-style-type: none">• UI 변수 정보 제공• 유저 피드백 및 게임 밸런싱 조절• 프레임 최적화
		버그 해결 및 기능 개선 <ul style="list-style-type: none">• 게임 각종 버그 해결 (NPC 응시, 점수 중복, 리스폰 위치 등)• 눈빛을 보낸 NPC가 플레이어를 바라보도록 하는 기능
		팀 리더 역할(조장) <ul style="list-style-type: none">• 팀원 버그 해결 지원• 방향성 설정• VFX 자료 제공

곽도 훈	UI/UX & System Integration	<p>프로젝트 관리 및 문서화</p> <ul style="list-style-type: none"> • 눈빛보내기 주제 제안 • 제안서 작성 • 중간 보고서 작성 • 판넬 제작 • 기술문서 및 readme 작성 <p>게임 UI/UX 제작</p> <ul style="list-style-type: none"> • 게임 UI/UX 제작 (점수판, 미니게임 화면, 엔딩 화면, LOVE 게이지, 남은 시간, 튜토리얼) • START 버튼을 눌러 게임 시작 기능 • NPC를 유혹하여 올라가는 점수 비율에 맞게 LOVE 게이지가 상승하여 피버모드가 되도록 하는 기능 • 미니게임 및 게임 시스템 UI 통합 • 엔딩 전환 제작 <p>게임 콘텐츠 및 시스템 개발</p> <ul style="list-style-type: none"> • 엔딩씬 제작 • NPC 배치 및 AI 기반 움직임 제작 • NPC가 유혹되면 플레이어를 따라오도록 만드는 기능 • NPC 캐릭터 제작 (11명) • 난이도 조절 <p>최적화 및 기타</p> <ul style="list-style-type: none"> • 프레임 최적화 진행 • 레이저 효과음 선정 <p>버그 해결 및 기능 개선</p> <ul style="list-style-type: none"> • 게임 각종 버그 해결 (NPC의 뒤통수를 바라봐야 꼬여지는 문제, 점수 중복 계산 문제, 플레이어 초기 리스폰 위치 오류 문제) • 눈빛을 보낸 NPC가 플레이어를 바라보도록 하는 기능 제작
---------	-------------------------------	--

류시 우	VFX & Animation	<p>프로젝트 관리 및 문서화</p> <ul style="list-style-type: none"> • 중간 발표 • 보고서 작성 • 유저 피드백 및 게임 밸런싱 조절 <p>LASER VFX 개발</p> <ul style="list-style-type: none"> • Visual Effect Graph 기반 레이저 빔 시스템 구현 • 레이저 빔과 스파크 파티클 시스템으로 구성된 복합 효과 • 충돌 감지 기반 동적 파티클 생성 및 위치 기반 방향 제어 • 깊이 버퍼 충돌과 스케일 조절을 통한 사실적인 레이저 표현 <p>Heart Particle 개발</p> <ul style="list-style-type: none"> • Legacy와 일반 버전으로 구성된 하트 파티클 효과 • 캐릭터 NPC 개발 <p>감정별 Animator Controller</p> <ul style="list-style-type: none"> • Happy/Sad/Angry/Neutral 전용 컨트롤러 • CharacterAura 연동으로 캐릭터 주변 비주얼 이펙트 구현(피비용) • Transform 기반 위치 추적과 파티클 생성률 제어 <p>통합 애니메이션 개발</p> <ul style="list-style-type: none"> • 통일된 매개변수 시스템 • 모든 컨트롤러에서 공통 Bool 매개변수 사용 (Happy/Sad/Angry/Neutral) • VFX와 애니메이션의 실시간 동기화로 몰입감 있는 캐릭터 표현 <p>NPC 시스템 통합</p> <ul style="list-style-type: none"> • 애니메이터 연결 • 감정 관련 시스템 및 미니게임 연결 • 캐릭터 별 애니메이션에 따른 콜라يدر 수정
---------	-----------------	--

3. 주요 기술 스택

기술 스택

개발 환경

- Unity 6000.0.43f1 - VR 게임 엔진
- Meta XR SDK - Quest Pro 하드웨어 최적화
- XR Interaction Toolkit - VR 상호작용 프레임워크

최적화 기술

- Occlusion Culling - 성능 최적화
- Tunneling Vignette - VR 멀미 방지

4. 사용 방법

설치 및 실행

시스템 요구사항

- VR 헤드셋: Meta Quest Pro (Eye Tracking 필수)
- 개발 환경: Windows 10/11, Unity 6000.0.43f1 이상

설치 방법

1. 저장소 클론

```
git clone https://github.com/Zeeeeeeing/zeeeeing.git  
cd zeeeeing
```

2. Unity에서 프로젝트 열기

- Unity Hub에서 "Add project from disk" 선택
- 클론한 폴더 선택하여 프로젝트 열기

3. Meta XR SDK 설정

- Window > Package Manager에서 Meta XR SDK 설치
- XR Management에서 Oculus 활성화

4. Eye Tracking 권한 설정

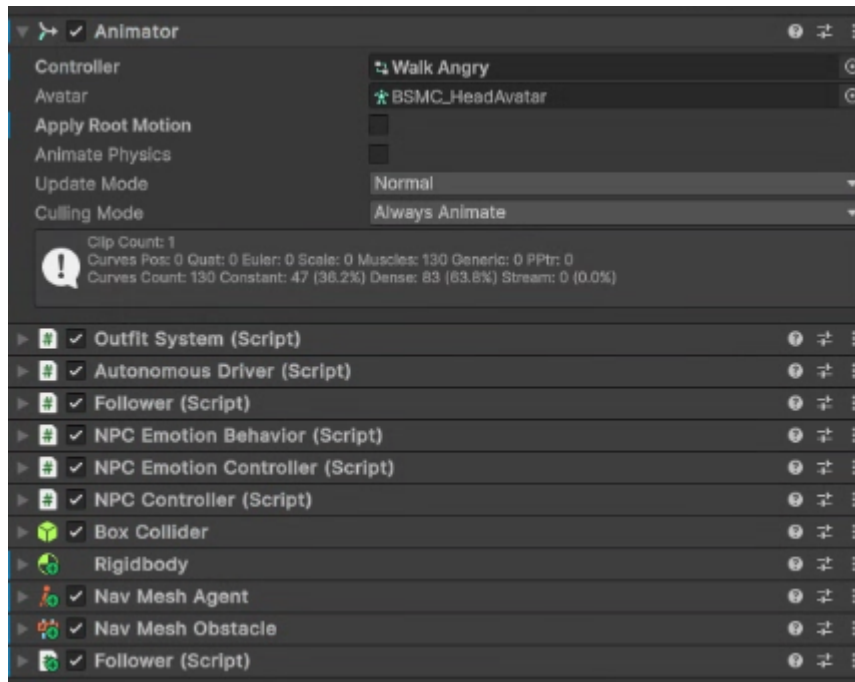
- Meta Developer Hub에서 개발자 모드 활성화
- Quest Pro에서 Eye Tracking 권한 허용

주의: 씬과 캐릭터 에셋의 경우 유료 에셋을 사용하였기에 이를 clone해와도 바로 사용하지는 못합니다.

씬: 사용하고자 하는 배경 환경의 경우 **Bake**를 통해 구워줍니다. 이를 통해 각 **NPC**들이 이동할 수 있는 경로를 탐색할 수 있게 됩니다. 최적화를 위해 해당 환경들을 클릭 후 아래와 같은 설정들을 변경해주세요.

1. Occlusion Culling
2. Cast Shadows 옵션을 Off
3. 이동하지 않는 모든 게임 오브젝트가 **Static**으로 설정

NPC:

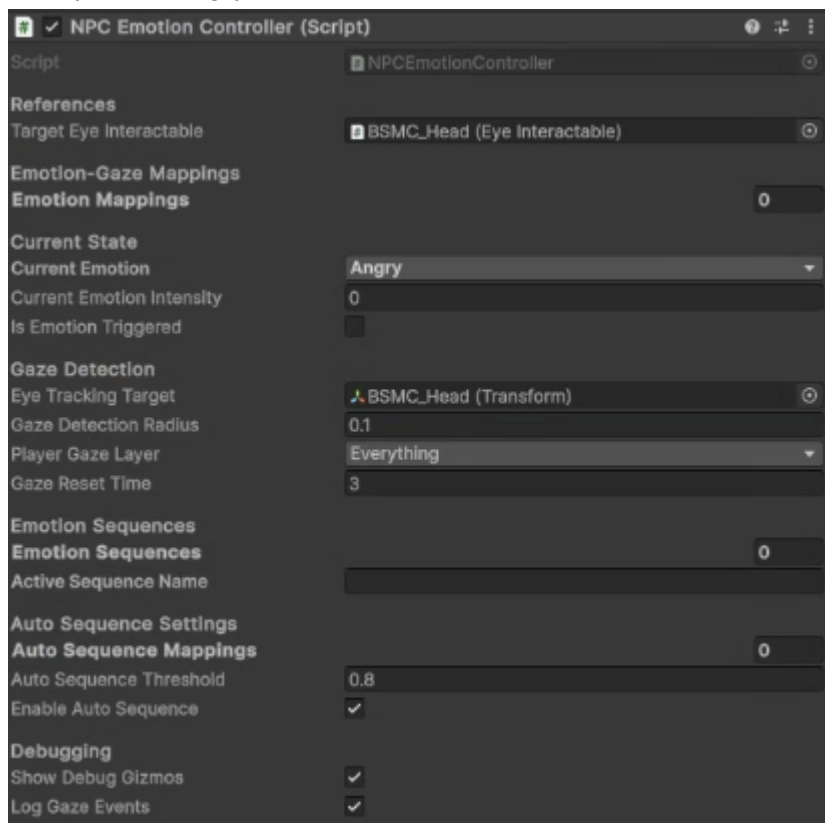


기본적인 NPC의 Inspector 구조입니다.

Outfit System의 경우 유료 에셋 제작을 통해 만들어진 스크립트이므로 무시하셔도 됩니다.

각 NPC들은 애니메이션과 Eye Interaction을 위한 얼굴 콜라이더 부분이 따로 존재해야 되기에 Rig가 존재해야 됩니다.

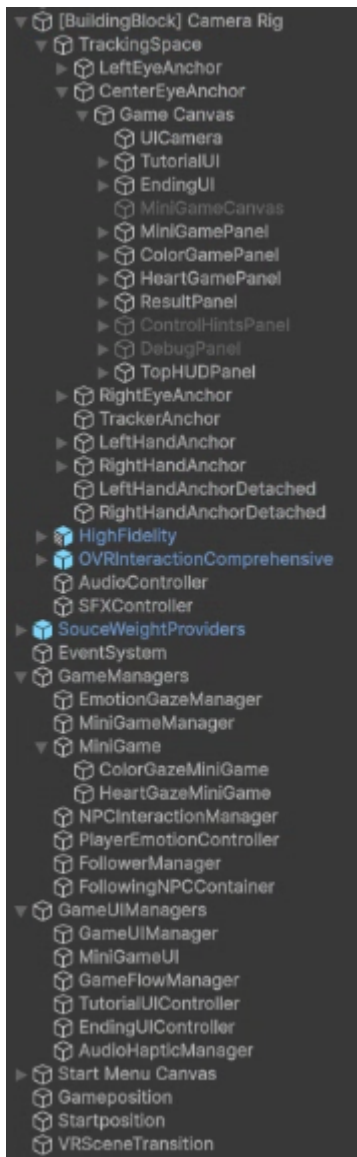
각 스크립트를 추가한 뒤 NPC Emotion Controller의 Current Emotion 부분을 Neutral, Happy, Sad, Angry 중 하나로 설정하시고 아래와 같이 설정해주세요.



Emotion Sequence의 경우 NPC별 Happy->Sad 와 같은 감정 변화를 원하신다면 사용하시면 됩니다. 데모에서는 빠른 템포를 위해 제거하였습니다.

NPCEmotionBehavior의 HeadTransform에도 NPC의 Head 부분에 해당하는 Object를 할당해주세요. 이외의 스크립트에서는 해당 NPC를 다시 할당하시면 됩니다.

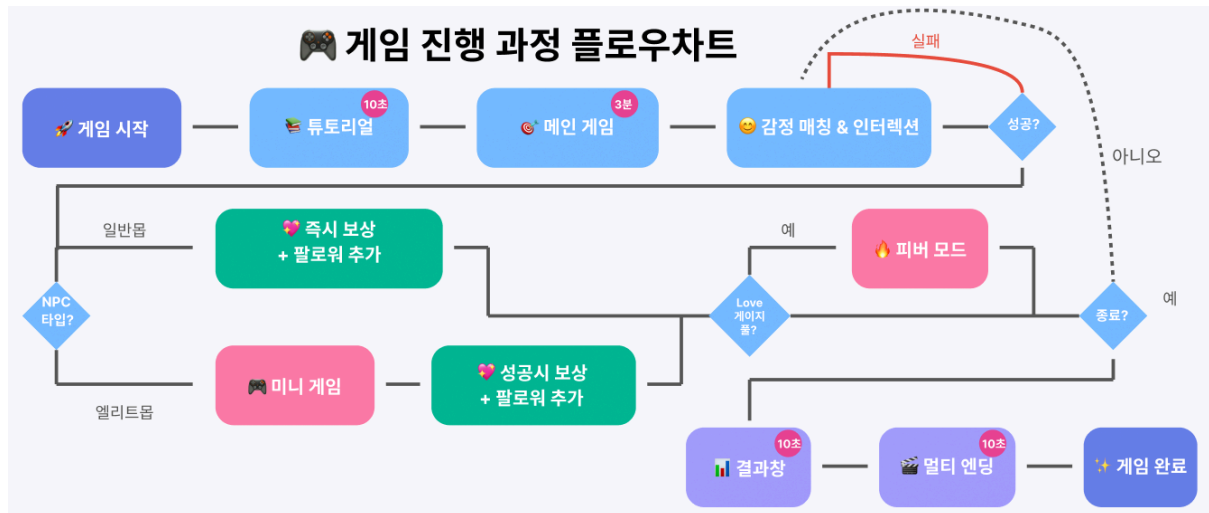
Hierarchy:



게임은 여러 **Manager** 코드들로 동작하지만 UI의 경우 썬을 드릴 수 없는 관계로 각 **UIManager**에 맞게 작성하셔야 합니다. 각각의 **Manager**객체들은 스크립트와 이름이 동일합니다. 이 중 유의할 점은 아래와 같이 2가지가 있습니다.

1. **NPCInteractionManager**의 **EliteNPCs**와 **GameFlowManager**의 **SceneNPCs**는 수동으로 NPC들을 참조해주셔야 합니다.
2. 메인 카메라 이외에 **Rendering Type**이 **Overlay**인 **UICamera**를 사용하여 각 UI가 벽 등에 가려지지 않고 플레이어의 시점과 관계없이 보일 수 있도록 설정해야 합니다.

플로우차트:



미니 게임의 경우 ColorGaze와 HeartGaze 2개로 나뉩니다.

Sound

BGM은 튜토리얼, 메인 게임, 피버 모드, 엔딩 총 4개

SoundEffect는 게임 시작, 레이저 총 2개가 존재합니다.

그 중 메인 게임과 피버 모드에서의 BGM은 MangMaru님의 유료 BGM을 사용하였습니다.

- 게임 시작시: <https://pixabay.com/ko/sound-effects/bubble-pop-4-323580/>
- 튜토리얼: <https://pixabay.com/ko/sound-effects/guitar-intro-ident-335301/>
- 게임 중: <https://www.sellbuymusic.com/musicDetail/50413>
- 피버 모드: <https://www.sellbuymusic.com/musicDetail/50921>
- 엔딩: Westminster-chimes
- 레이저 소리: <https://www.mewpot.com/sound-effects/2412>