

Final Report

Eye Tracking을 활용한 눈빛보내기 VR 재구성



과목명

가상및증강현실프로그래밍

담당교수

유용재

학생이름

정현욱, 곽도훈, 류시우

학과

인공지능

팀명

zeeing

HANYANG UNIVERSITY

1. 프로젝트 개요와 배경

1.1 프로젝트 정의와 목표

'눈빛 보내기 VR'은 2000년대 초반 플래시 게임 '눈빛 보내기'를 Meta Quest Pro의 Eye Tracking 기술을 활용해 완전히 새로운 VR 경험으로 재탄생시킨 프로젝트다. 단순한 포팅을 넘어, VR과 시선 추적 기술을 활용하여 플레이어가 직접 눈빛을 보내며 게임을 진행하게 만들었다. 원작 게임은 "지나다니는 남학생 얼굴에 커서를 갖다 대면 룩온이 되고, 그 상태에서 꾹 누르면 남학생을 향해 눈빛을 발사한다. 동시에 남학생 머리 위에 하트 게이지가 뜨는데, 계속 누르고 있으면 점점 채워진다. 끝까지 채우면 검게 불태워짐과 동시에 하트를 떨어뜨리며 하트 눈을 띄운 채로 주인공 뒤를 따라다닌다"는 구성이다. 본 프로젝트는 이러한 원작의 핵심 메커니즘을 VR 환경에서 재해석하되, 단순한 마우스 클릭을 넘어서 사용자의 실제 시선 추적을 게임의 핵심 입력으로 활용하는 시스템을 제작하였다.

2. 원작 분석과 VR 재해석 전략

2.1 원작 게임 메커니즘의 심층 분석

원작 '눈빛 보내기'의 핵심은 단순하면서도 중독성 있는 메커니즘이 있었다. 플레이어는 마우스로 남학생을 룩온한 후 클릭을 유지하여 하트 게이지를 채우는 단순한 작업을 반복했지만, 여기에 라이벌 시스템과 '뷰티 타임(피버 모드)'이라는 파워업 메커니즘이 더해져 단순하지만 전략적 움직임을 필요로 하는 게임이다. 특히 "만일 라이벌(다른 여학생, 엘리트 NPC)이 화면 안에 있는 채로 남학생을 유혹하거나, 유혹하는 도중에 화면 안으로 들어오면, 그 여학생이 느낌표를 띄우면서 주인공과 눈빛 경쟁을 하게 되는데, 이 때는 연타로 승부를 내야한다"는 경쟁 시스템은 단조로울 수 있는 게임 플레이에 긴장감과 변화를 제공했다. 또한 "원쪽 상단에 러브 게이지 9칸이 있는데, 전부 다 채우면 뷰티 타임이 발동된다. 반짝거리는 특수 효과가 생기고 눈빛이 깊어지며 남학생을 눈빛으로 유혹하는 속도가 비약적으로 상승한다"는 피버 모드 시스템은 플레이어에게 빠른 템포와 성취감을 제공하는 핵심 요소였다.

2.2 VR 환경으로의 전환

2D 마우스 클릭 기반 게임을 VR로 전환하는 과정에서 가장 큰 도전은 원작의 '시선'이라는 개념을 실제 물리적 시선으로 구현하는 것이었다. 원작에서는 마우스 커서가 시선을 대신했지만, VR에서는 사용자의 실제 시선 데이터를 Meta Quest Pro의 Eye Tracking API로 수집하여 게임 메커니즘으로 활용해야 했다.

2.3 원작 요소의 VR 적용과 확장

원작의 핵심 요소들을 VR 환경에 맞게 재해석하면서도 새로운 가능성을 탐구했다. 원작의 '눈빛 발사'는 Unity Visual Effect Graph를 활용한 시선 레이저 VFX로 구현했으며, 감정 상태에 따라 색상이 동적으로 변화하도록 설계했다. 하트 게이지 시스템은 3D 공간에서 실시간으로 NPC 머리 위에 표시되는 호감도 게이지로 발전시켰고, 원작의 '뷰티 타임'은 피버 모드 시스템으로 재해석하여 황금색 VFX가 플레이어 아래에서 생성되며 성공 시간이 동적으로 단축되는 메커니즘으로 구현했다. 특히 라이벌 시스템(2.1 밑줄)을 원작에서의 구성을 그대로 가져오게 된다면 VR에서 눈빛을 보낸다는 특성 이외에 다른 게임적 요소가 부족할 것 같다고 생각하였다. 또한 기존의 라이벌 시스템인 '연타'라는 요소는 컨트롤러에는 적합하지 않다 판단하였고, 추가적으로 미니게임을 제작하여 엘리트 NPC에게 적용하였다. 해당 미니게임에 대한 설명은 4.2에서 진행할 예정이다.

3. 감정-시선 매핑 시스템의 기술적 구현

3.1 시스템 아키텍처와 데이터 흐름

감정-시선 매핑 시스템의 핵심은 두 개의 독립적인 입력 스트림을 실시간으로 융합하는 것이다. 첫 번째는 Meta Quest Pro의 Eye Tracking API에서 수집되는 시선 데이터이고, 두 번째는 컨트롤러 버튼으로 입력되는 감정 상태 데이터다. 시선 데이터는 먼저 실시간 레이캐스팅을 거친다. 카메라에서 시선 방향으로 Ray를 발사하여 EyeInteractable한 객체를 감지한다. 그 다음 거리 기반 필터링을 통해 유익한 범위 내에 있는 NPC만을 '시선 대상'으로 선정한다. 동시에 컨트롤러 입력으로 받은 감정 상태는 EmotionGazeMapping 클래스를 통해 해당 감정에 특화된 파라미터 세트로 변환된다. 기존에는 각 감정 시퀀스별로 gazeStabilityDuration(시선 안정화 시간), gazeSensitivity(시선 감도), emotionBuildupRate(감정 상승률), emotionDecayRate(감정 감소율), gazeColor(시선 효과 색상) 등의 고유한 값을 통해 변화할 수 있게 설정하였다. 하지만 데모에서의 빠른 템포를 고려해 NPC별로 단일 감정을 갖도록 수정하였다. 플레이어는 컨트롤러의 trigger, grip의 총 4개 버튼을 이용해 neutral, happy, sad, angry로의 감정 변화를 할 수 있으며, NPC와 같은 감정을 통해 바라보며 공감을 하여 상대방을 유혹할 수 있다.

3.2 실시간 피드백 시스템

감정-시선 매핑의 결과는 다층적인 피드백 시스템을 통해 플레이어에게 전달된다. 시각적 피드백으로는 시선 레이저 색상과 NPC 주변의 파티클 효과가 있다. 올바른 감정으로 NPC를 바라본다면, 짙은 핑크 색상의 레이저가 나가며 주변에 하트 파티클이 생성되는 반면, 틀린 감정으로 NPC를 바라본다면, 얇은 빨간색 레이저가 나가게 개발하여 플레이어가 즉각적으로 피드백을 받을 수 있게 개발하였다. 즉각적 피드백으로는 플레이어를 성공적으로 유혹하는 순간에 컨트롤러 진동이 제공되어 성공과 실패를 명확하게 구분할 수 있게 한다.

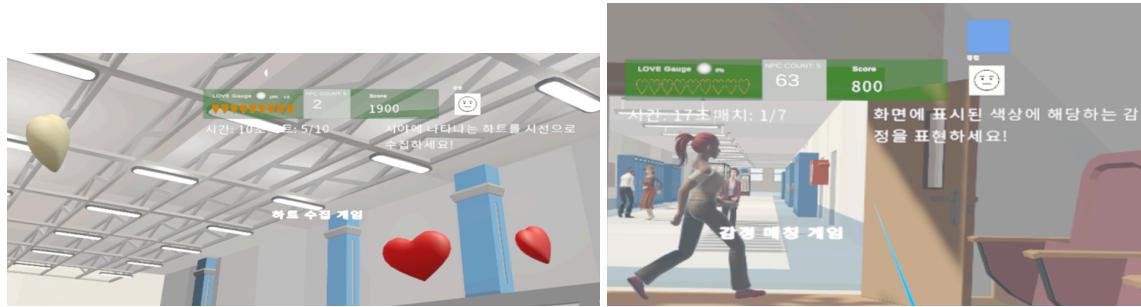
4. 게임 플레이 시스템

4.1 Demo 중심의 게임 플레이 구조

본 프로젝트는 데모 환경에서의 체험을 목표로 설계되어, 총 3분 30초(210초)의 완성된 경험을 제공한다. 10초 튜토리얼, 180초 메인 게임 플레이, 10초 결과화면, 10초 엔딩으로 구성하였다. 튜토리얼 단계에서는 감정-시선 매핑 시스템의 기본 조작법을 화면에 또는 instruction UI를 통해 학습시킨다. 메인 게임 플레이는 일반 NPC 유혹하기와 엘리트 NPC 미니게임으로 구성된다. 일반 NPC는 감정 매칭 후 Eye Tracking만으로 호감도를 쌓을 수 있지만, 엘리트 NPC는 앞선 과정 이후에도 Color Gaze(색깔 맞추기) 또는 Heart Gaze(하트 추적) 미니게임을 통과해야 성공할 수 있다.

4.2 미니게임 시스템 구현

Color Gaze 미니게임은 유혹하고자 하는 NPC의 순간순간 바뀌는 감정의 상태에 맞게 플레이어의 감정 상태를 동일하게 변경하며 일정 시간 이내에 일정 개수 이상을 맞춰야 NPC 유혹에 성공할 수 있게 만들었다. 난이도 또한 3개로 설정하여 게임에 재미요소를 더 추가하고자 하였다. Heart Gaze 미니게임은 "Eye tracking을 활용하여 화면에 나타나는 하트를 바라보며 하트를 수집"하는 3D 공간 게임으로, 하트들이 3D 공간에서 랜덤하게 생성되고 소멸하는 패턴을 추적해야 하므로, 플레이어의 공간 지각 능력과 빠른 시선 이동 능력이 동시에 요구된다. 해당 미니 게임 또한 난이도를 3가지로 설정하여 게임에 재미요소를 더 추가하고자 하였다.



4.3 피버 모드와 동적 점수 시스템

원작의 '뷰티 타임'을 VR에 맞게 재설계한 피버 모드는 NPC를 유혹할 때마다 화면의 원쪽 위 UI에 비어있는 하트가 점점 채워지게 된다 (LOVE 게이지). 해당 하트를 모두 채우게 되면 자동 활성화된다. 플레이어의 다리 밑에서 황금빛의 VFX가 발생하며 NPC 유혹 성공 시간이 0.1배로 단축된다. 이는 단순한 시간 단축이 아니라, 플레이어의 상태에 따라 게임의 템포 자체가 변화하게 만들어 더 긴박한 느낌을 줄 수 있게 제작하였다. 점수 시스템은 일반 NPC 성공 시 기본 100점, 엘리트 NPC 성공 시 추가 점수를 제공하며, 피버 모드에서는 모든 점수에 배율이 적용된다. 최종적으로 S/A/B/C/D 등급으로 성과를 평가하여 엔딩씬을 각각 다르게 제작하여 플레이어에게 명확한 성취도 피드백을 제공한다.

5. NPC AI와 감정 상태 머신

5.1 NPCEmotionController의 복합 상태 관리

각 NPC는 Happy, Angry, Sad, Neutral의 4가지 감정 상태를 가지며, 플레이어의 감정-시선 조합에 따라 동적으로 전환된다. NPCEmotionController는 Unity Animator와 연동하여 감정 변화 시 자연스러운 애니메이션 전환을 제공한다. 감정 상태 전환 시 DebugEmotionChange 메서드를 통해 모든 애니메이터 파라미터 상태를 로깅하여 디버깅을 지원한다. 또한 HasParameter 메서드로 애니메이터 파라미터 존재 여부를 사전 확인하여 런타임 오류를 방지한다. 특히 0.1초 지연 후 상태 확인하는 VerifyParameterChange 코루틴을 통해 애니메이터 전환의 안정성을 보장한다. 이는 Unity Animator의 상태 전환이 완료되는 시간을 고려한 안전장치로, 시각적 불일치를 방지한다.

5.2 NPCInteractionManager의 중앙 제어 시스템

NPCInteractionManager는 모든 플레이어-NPC 상호작용을 관리하는 중앙 컨트롤러로, 피버 모드 상태 감지, 감정 매칭 성공 처리, VFX 효과 동기화 등을 담당한다. ProcessSuccessfulEmotionMatching 메서드를 통해 성공적인 감정 매칭 시 성공 VFX를 표시한다. UpdateInteractionProgress 메서드는 피버 모드 상태를 실시간으로 반영하여 상호작용 진행도를 계산한다. 이때 GetCurrentRegularNPCTime()을 통해 동적 시간 조절을 구현하여, 피버 모드에서는 더 빠른 성공이 가능하도록 한다. FollowerManager를 통해 성공적으로 유혹된 NPC들이 플레이어를 따라다니는 시스템을 구현했다. 이때 Y축 위치 보정을 통해 NPC가 바닥 아래로 떨어지는 것을 방지한다.

6. VR 전용 UI 시스템과 HUD 구현

6.1 World Space Canvas와 VR 최적화 UI

본 프로젝트에서는 VR에 최적화된 UI 시스템을 구현했다. 가장 중요한 부분은 VR을 낸 상태에서 플레이어가 다른 오브젝트에 다가갔을 때, 플레이어의 앞에 위치한 UI가 다른 오브젝트에 가려지지 않게 만들어, 사용자의 불편함을 최소화하고자 하였다.

6.2 동적 UI 제어와 게임 상태 연동

본 프로젝트의 UI 시스템은 단순한 정보 표시를 넘어서 게임 상태와 완전히 연동되는 동적 시스템이다. 게임 진행 상황에 따라 UI 요소들이 선택적으로 활성화/비활성화된다. 튜토리얼 단계에서는 UI가 비활성화되어 플레이어가 기본 조작에 집중할 수 있게 하고, 메인 게임 단계에서만 점수, 타이머, LOVE 게이지 등이 표시된다. 특히 LOVE 게이지 시스템은 원작의 핵심 메커니즘을 VR에 맞게 재해석한 것으로, 9개의 하트 이미지를 통해 플레이어의 진행도를 시각적으로 표현한다. `scorePerHeart` 시스템을 통해 점수와 시각적 피드백이 직관적으로 연결되며, 피버 모드 활성화 시 `blockHeartIncreaseInFever` 옵션으로 피버모드 중일 때는 LOVE 게이지가 올라가지 않게 만들어 게임 밸런스를 조절한다.

6.3 미니게임 UI

본 프로젝트의 미니게임 UI 시스템은 VR 환경에서 직관적이고 몰입감 있는 상호작용을 제공하도록 설계되었다. **Color Gaze** 미니게임과 **Heart Gaze** 미니게임 UI는 현재 대상 색상, 남은 시간, 그리고 실시간 매칭 상태를 명확하게 표시한다. 동적 UI 제어 시스템은 게임 상태에 따라 적절한 UI 요소를 자동으로 활성화/비활성화한다. 미니게임 시작 시 해당 게임의 UI 패널만 표시되고, 결과 화면에서는 성공/실패 여부와 획득 점수가 명확하게 표시된다.

6.4 엔딩 UI

엔딩 UI 시스템은 플레이어의 게임 성과를 종합적으로 평가하고 시각적으로 결과 화면을 제공한다. 멀티 등급 시스템은 최종 점수를 기반으로 S/A/B/C/D 등급을 부여하며, 각 등급에 따라 고유한 색상(금색, 은색, 동색, 회색, 적갈색)으로 표시된다. 등급별로 차별화된 시각적 피드백을 제공하여 플레이어의 성취감을 극대화한다. 종합 통계 표시는 최종 점수, 성공한 NPC 수, 완료율, 가장 많이 사용한 감정 등의 상세한 게임 데이터를 제공한다.



7. NPC 경로 탐색과 AI 시스템

7.1 Unity NavMesh 기반 자율 이동 시스템

본 프로젝트의 NPC들은 AutonomousDriver 시스템을 통해 완전히 자율적으로 학교 환경을 탐색한다. Unity NavMesh를 기반으로 하되, 단순한 경로 추적을 넘어서 장애물 회피, 동적 목적지 선택, 상호작용 모드 전환 등의 기능을 추가로 구현했다. NavMeshTriangulation을 활용하여 게임 시작 시 전체 네비게이션 메쉬 정보를 캐싱하고, edgeMargin 시스템을 통해 NavMesh 경계로부터 일정 거리 이내의 안전한 지점만을 목적지로 선택한다. 이는 NPC가 벽이나 장애물에 너무 가까이 접근하는 것을 방지하여 자연스러운 움직임을 보장한다.



7.2 동적 상호작용 모드

AutonomousDriver에서 중요한 기능은 동적 행동 전환이다. 평소에는 자율적으로 이동하지만, 플레이어가 눈빛을 보내면 NPC가 플레이어를 즉시 정면으로 바라보며 정지할 수 있게 구현하였다. 이후 플레이어가 눈빛을 보내지 않아 상호작용이 종료되면 즉시 다시 길을 탐색하여 이동하게 된다.

7.3 장애물 회피와 경로 최적화

NavMeshAgent의 obstacleAvoidanceType을 HighQualityObstacleAvoidance로 설정하여 다른 NPC들과의 충돌을 자연스럽게 회피한다. 이는 여러 NPC가 동시에 움직이는 환경에서 현실적인 군중 시뮬레이션을 제공한다. 경로 계산은 agent.pathPending 상태를 확인하여 비동기적으로 처리되며, agent.remainingDistance와 arriveThreshold(0.5미터)를 비교하여 목적지 도착을 정확히 판정한다. PickNewDestination 메서드는 maxSampleAttempts(30회) 시도를 통해 최적의 새로운 목적지를 선택하며, 실패 시에도 안전한 대안을 제공한다.

8. VFX 시스템과 시각적 몰입도

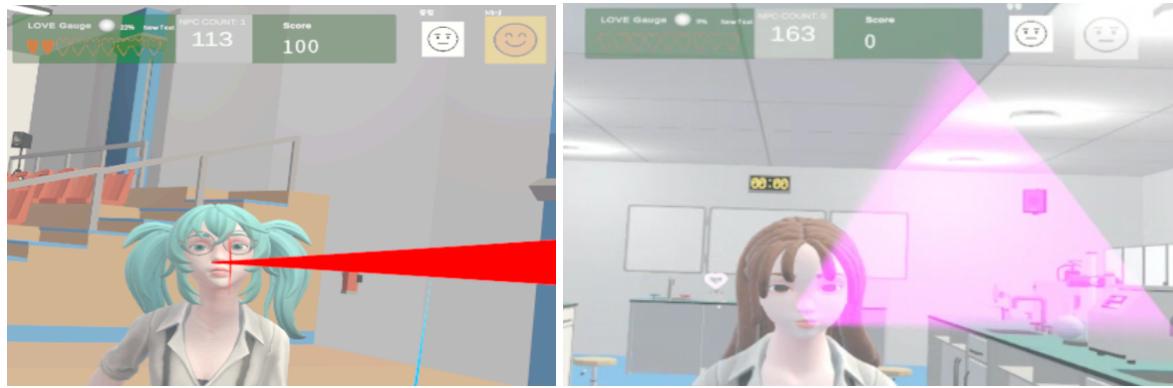
8.1 Unity Visual Effect Graph를 활용한 시선 레이저 시스템

8.1.1 시스템 개요

시선 레이저는 본 프로젝트의 가장 상징적인 시각적 요소로, Unity Visual Effect Graph를 활용하여 구현했다. VR 환경에서 플레이어의 시선 추적을 통해 직관적인 타겟팅 경험을 제공하며, 동시에 게임의 분위기를 강화하는 핵심 요소다.

8.1.2 레이저 발광 시스템

레이저의 발광 시스템은 "Laser가 자체적으로 빛을 나게 함"이라는 컨셉으로 구현했다. 단순히 시각적 효과에 그치지 않고 주변 환경을 실제로 밝히는 역할도 수행한다. 이를 통해 시선의 방향을 명확하게 표현하면서도 게임 세계의 몰입감을 높였다. 구현 과정에서 Strip 파티클 시스템을 선택한 이유는 레이저와 같은 연속적인 빔 효과를 표현하기에 가장 적합했기 때문이다. Unity 6에서 새롭게 추가된 Strip 관련 기능들을 활용해 더욱 효율적인 렌더링이 가능해졌다. 레이저의 생동감을 위해 미세한 크기 변화와 색상 펄스 효과를 추가했다. 이는 단조로운 직선이 아닌 살아있는 에너지 빔의 느낌을 연출하기 위함이었다.



8.1.3 총돌 이펙트 시스템

총돌 지점에서 발생하는 스파크 파티클은 조건부 스폰 시스템을 통해 구현했다. Hit 상태가 true일 때만 파티클이 생성되도록 하여 불필요한 연산을 줄였다. 스파크의 방향은 총돌 표면의 법선 벡터를 활용해 물리적으로 자연스럽게 표현했다. 특히 VR 환경에서는 깊이 버퍼와의 총돌 처리가 중요했다. 이를 통해 스파크가 벽면이나 객체와 자연스럽게 상호작용하도록 했다.

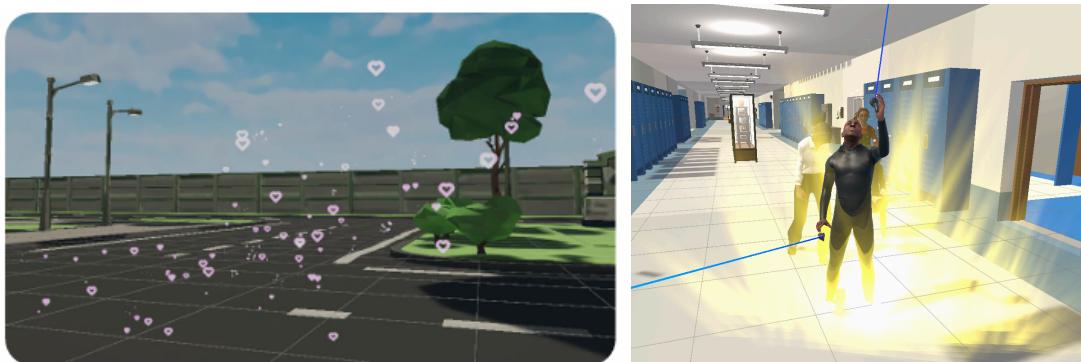
8.1.4 성능 최적화 고려사항

VR 환경에서는 안정적인 프레임레이트 유지가 필수적이기 때문에 여러 최적화 기법을 적용했다. **Strip**의 첫 번째 파티클만 실제로 렌더링하고, 총돌 상태에 따라 렌더링 방식을 동적으로 변경하는 등의 조건부 처리를 통해 성능을 확보했다.

8.2 감정별 파티클 시스템

8.2.1 NPC 감정 상태 시각화

"몹의 상태에 따른 particle 효과 추가 : 몹의 상태가 꼬서짐으로 변화함에 따라서 heart particle을 띠워 상태를 직관적으로 표현"하는 시스템은 플레이어가 NPC의 호감도를 한눈에 파악할 수 있도록 설계했다. 하트 파티클 시스템은 **Unity**의 기본 파티클 시스템을 활용해 구현했다. **Visual Effect Graph**보다는 전통적인 파티클 시스템이 이런 단순한 효과에는 더 적합하다고 판단했기 때문이다. 또한 런타임에서 파라미터 조정이 용이해 호감도 레벨에 따른 동적 변화를 구현하기 쉬웠다. 하트가 위로 떠오르면서 서서히 사라지는 애니메이션은 감정의 자연스러운 흐름을 표현하고자 했다. 중력 효과와 투명도 변화를 조합해 현실적인 연출을 만들어냈다.



8.2.2 피버 모드 전체화면 효과

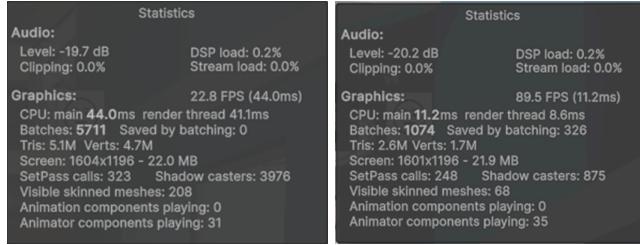
"Fever Time 각성 효과 추가 : 몸 아래에 황금빛의 Aura로 빛나는 시스템은 플레이어에게 강력한 성취감을 제공하는 것이 목표였다. 피버 모드 진입 시 단순한 시각적 변화를 넘어 게임 플레이의 특별함을 강조한다. 화면 전체를 덮는 황금빛 효과는 Post-Processing Volume을 활용해 구현했다. 이는 개별

파티클로는 구현하기 어려운 전역적 시각 효과를 가능하게 했다.

8.2.3 시스템 통합과 관리

모든 파티클 시스템은 통일된 관리 구조 하에서 작동하도록 설계했다. 이를 통해 게임 상황에 따른 일괄적인 품질 조정이나 성능 최적화가 가능하다. 특히 VR 환경에서는 파티클 효과가 과도할 경우 시각적 피로나 멀미를 유발할 수 있어, 플레이어 설정에 따른 강도 조절 기능을 포함했다. 하이브리드 접근법을 통해 각 효과의 특성에 맞는 최적의 파티클 시스템을 선택했다. 복잡한 효과는 **VFX Graph**를, 단순한 효과는 레거시 파티클 시스템을 활용하는 방식으로 개발 효율성과 성능을 모두 확보했다.

9. 성능 최적화와 사용자 경험



9.1 프레임 최적화

Unity의 내장 Occlusion Culling 시스템을 활용하여 "가려진 객체 렌더링 제외"를 구현함으로써 GPU 부하를 감소시켰다. 추가적으로 프로젝트에서는 **Static Batching** 최적화를 통해 드로우콜을 대폭 감소시켰다. 씬 내에서 움직이지 않는 모든 환경 오브젝트들에 대해 **Static** 플래그를 설정하여 Unity의 배치 시스템이 이들을 하나의 드로우콜로 통합하도록 하였다. 건물, 지형 요소, 장식용 오브젝트들을 포함하여 런타임 중에 이동하지 않는 모든 게임 오브젝트가 **Static**으로 설정되어 CPU 렌더링 부하가 현저히 감소하였다. 그림자 렌더링 최적화의 경우 과도한 그림자 생성으로 인한 GPU 부하를 줄이기 위해 선택적으로 그림자 렌더링을 비활성화하는 작업을 수행하였다. 시각적 중요도가 낮은 소형 장식 오브젝트들, 배경의 세부 요소들, 그리고 카메라로부터 상당한 거리에 위치한 오브젝트들의 **Renderer** 컴포넌트에서 **Cast Shadows** 옵션을 Off로 설정하였다. 이를 통해 핵심적인 게임 플레이 오브젝트들의 그림자는 유지하면서도 불필요한 그림자들을 제거하여 시각적 품질을 크게 해치지 않으면서도 성능을 개선할 수 있었다. 다음과 같은 최적화 방식으로 Meta Quest Pro의 최대 프레임인 90fps를 학교 환경의 복잡한 구조에서도 안정적으로 유지한다.

9.2 VR 멀미 방지

"Tunneling Vignette (멀미 방지) 시야 제한으로 멀미 감소" 시스템을 구현하여 빠른 시선 이동이나 화면 전환 시 시야 가장자리를 점진적으로 어둡게 처리한다. 이는 VR 멀미의 주요 원인인 시각적 충격을 완화하며, 개인 민감도에 따라 강도 조절이 가능하다.

9.3 제한시간과 난이도 조절 시스템

"제한시간 + 미니게임 (난이도, 템포 조절) 타이머 미니게임 난이도 시간 압박과 도전 과제로 플레이어 몰입도 조절" 시스템을 통해 게임 플레이를 구현했다. 특히 피버 모드에서는 시간 압박이 증가하지만 성공 시 더 큰 보상을 제공하여, 위험-보상 균형을 통한 전략적 선택을 유도한다. 이러한 시스템은 짧은

3-4분 데모 시간 내에서도 충분한 도전과 성취감을 제공한다.

Difficulty Settings - Optimized for Demo	
Easy Required Matches	3
Normal Required Matches	5
Hard Required Matches	7
Easy Time Limit	20
Normal Time Limit	20
Hard Time Limit	20

Difficulty Settings - Optimized for Demo	
Easy Target Hearts	5
Normal Target Hearts	8
Hard Target Hearts	10
Easy Game Time	20
Normal Game Time	20
Hard Game Time	20

9.4 유혹을 성공한 NPC 추적 시스템

성공적으로 유혹된 NPC들은 플레이어를 자동으로 따라다니는 시스템이 구현되어 있다. 이는 원작의 "하트 눈을 띄운 채로 주인공 뒤를 따라다닌다"는 메커니즘을 3D VR 환경에 맞게 발전시킨 것이다.



10. 프로젝트 한계와 개선 방향

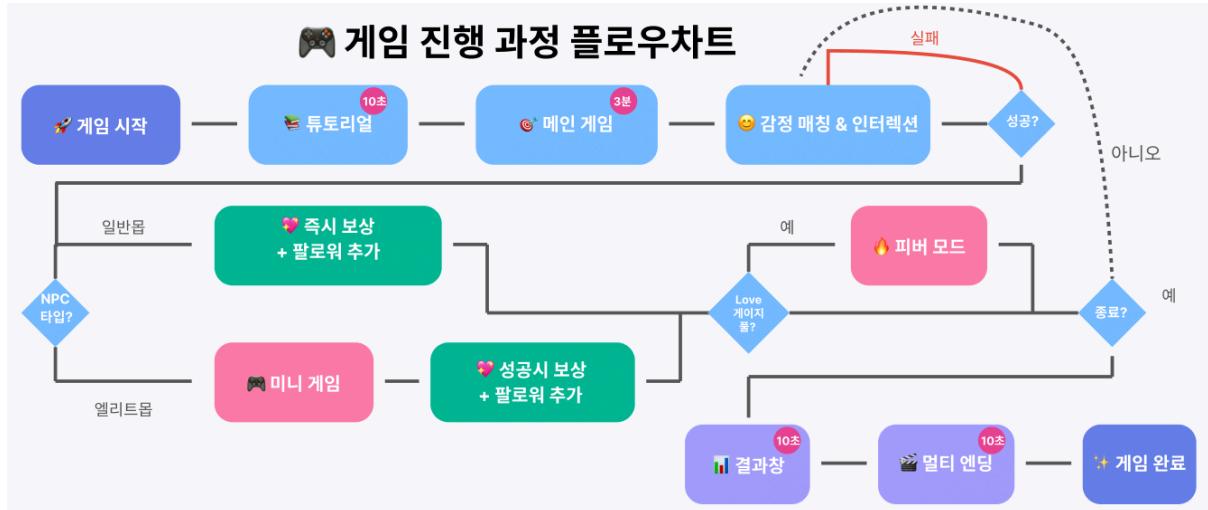
10.1 현재 시스템의 기술적 한계점

Meta Quest Pro의 Eye Tracking 기술 자체의 한계로 인해 안경 착용자나 특정 눈 형태를 가진 사용자에게서는 추적 정확도가 떨어지는 문제가 있다. 현재의 NPC AI는 미리 정의된 감정 전환 패턴을 따르므로, 충분히 플레이하면 예측 가능해진다는 한계가 있다. 진정한 의미의 '지능적' 반응을 위해서는 머신러닝 기반의 동적 학습 시스템이 필요하다.

10.2 콘텐츠 확장성과 재플레이 가치

현재 시스템은 학교 환경의 제한된 공간과 NPC로 구성되어 있어, 장기적 플레이를 위해서는 더 다양한 환경과 상황, 캐릭터가 필요하다. 도서관, 카페, 직장 등 다양한 사회적 상황에서의 감정-시선 상호작용을 추가하여 콘텐츠의 깊이를 확장해야 한다. 또한 현재의 점수 기반 평가 시스템을 넘어서 장기적 관계

발전, 스토리 진행, 캐릭터 성장 등의 RPG적 요소를 추가하여 재플레이 가치를 높여야 한다. 이를 통해 단순한 기술 데모를 넘어서 실제 게임으로서의 완성도를 높일 수 있을 것이다.



11. 최종 결론

본 프로젝트는 2000년대 초 플래시 게임 '눈빛 보내기'를 Meta Quest Pro의 시선 추적 기술을 활용하여 VR 환경으로 재구성한 기술 데모이다. 핵심적으로 플레이어의 시선 데이터를 게임 입력으로 활용하는 감정-시선 매핑 시스템을 구현하였으며, NPC AI, VR에 최적화된 UI 시스템, Unity Visual Effect Graph 기반의 시각 효과 등 다양한 기술 요소를 통합하여 몰입감 있는 경험을 제공한다.

현재 시스템은 Meta Quest Pro의 시선 추적 기술 자체의 한계, 예측 가능한 NPC 감정 패턴, 그리고 제한된 콘텐츠 확장성 등의 개선 여지를 가지고 있다. 향후 머신러닝 기반 AI, 다양한 환경 및 캐릭터 추가, RPG 요소 도입 등을 통해 완성도를 높이고 재플레이 가치를 향상시킬 수 있을 것으로 기대된다.

프로젝트 정보

- 개발팀: zeeeeing (정현욱, 곽도훈, 류시우)
- 소속: 한양대학교 인공지능학과
- 지도교수: 유용재 교수
- 개발기간: 2025.03 - 2025.06 (15주)
- 기술스택: Unity 6000.0.43f1, Meta XR SDK, Visual Effect Graph
- 플랫폼: Meta Quest Pro
- 리포지토리: <https://github.com/Zeeeeeing/zeeeeing>

Reference

MetaXRSDK: [Hello World | Meta Horizon OS Developers](#)

MovementSDK: [Movement SDK for Unity - Getting Started](#) (body, face, eye tracking)

EyeTracking: [Eye Tracking for Movement SDK for Unity](#),

[Add Eye Tracking Features With Oculus Integration For Unity](#)

Player Locomotion: [Create Locomotion Interactions](#)

Assets

- 학교 에셋: [LowPoly School](#)
- 캐릭터 제작 에셋: [BoZo: Stylized Modular Characters](#)

VFX

- 하트 파티클: <https://taimiso0319.booth.pm/items/3043804>
- 레이저: https://www.youtube.com/watch?v=_SaBXY-Ejgo
- 피버: <https://www.youtube.com/watch?v=h0rpKQEa7vQ>

SoundEffect

- 게임 시작 시: <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>