



Unity로 가상 및 혼합 현실 경험 제작

목차

들어가는 말	7
업계 용어.....	9
현실 유형 선택: VR, AR 또는 MR	11
모션 관련 고려 사항	13
XR 월드 구축하기	14
내러티브 개발	14
환경 스토리텔링	14
목적에 부합하는 디자인	14
시각적 단서와 청각적 단서	14
인터랙티브 요소	15
사용자 참여 전략.....	15
디테일이 주는 몰입감	15
상세한 탐색 유도.....	15
적응적 난이도	16
소셜 상호 작용	16
반복적 디자인	16
Unity 사용이 처음이라면	18
XR용 에셋 제작	19
컨셉 및 디자인	19
모델링	19
피벗 포인트	20
텍스처링	21
트림 시트.....	21
리깅 및 애니메이션	22
모델 최적화	22
익스포트	23

Unity의 XR: 주요 시스템 및 툴셋	25
렌더링 파이프라인	25
AR Foundation	27
XR Interaction Toolkit.....	28
XR 개발용 SDK.....	29
OpenXR	30
에디터 버전 및 모듈	30
VR 코어 샘플	31
버전 3.0으로 업데이트	32
튜토리얼 창	35
샘플 씬.....	35
샘플 씬 설정	37
프로젝트 템플릿	40
URP를 사용하여 새로운 VR 프로젝트 생성.....	41
3D URP 템플릿으로 시작	41
XR 플러그인 관리	42
상호 작용 프로파일 추가.....	44
XR Interaction Toolkit 구현.....	46
Starter Assets를 사용하여 XRI 살펴보기	47
데모 씬.....	47
Starter Assets 데모 씬에서 XR 상호 작용 설정	49
Input Action Manager.....	49
XR Interaction Manager	49
XR Origin	49
Character Controller Driver.....	50
XR Input Modality Manager.....	51

TrackedPoseDriver (Input System)	51
Controller Input Action Manager	52
XR Interaction Group 컴포넌트	52
XR Transform Stabilizer	52
상호 작용 설정	53
XR Direct Interactor	53
이동 시스템	53
방향 전환	54
이동	54
순간 이동	54
오르기	55
터널링 비네트	55
VR 월드와의 상호 작용	56
XRGrabInteractable	56
누르기 인터랙터블	56
XR Simple Interactor	57
XR Poke Filter	57
XR Poke Follow Affordance	57
응시 인터랙터블	58
XR Gaze Interactor	59
Gaze Input Manager	60
UI 요소와의 상호 작용	60
XR UI Input Module 컴포넌트	61
Tracked Device Graphic Raycaster	61
프로젝트 예시를 통해 알아보는 VR 개발	63
목표 설정	64
사전 계획	65

엔진 시작하기	66
Unity 버전 및 URP.....	66
빌드 설정.....	67
품질 설정.....	67
월드 구성.....	68
프로토타입 및 모델링	68
효율적인 텍스처링	69
조명 적용하기	70
조명 방법 선택하기.....	70
조명 최적화.....	71
가상 월드 탐색	72
순간 이동 유형	72
XR Interaction Toolkit을 사용하여 퍼즐 만들기.....	74
XR Grab Interactor.....	76
XR Socket Interactor.....	76
소켓 인터랙터 만들기	77
테스트 및 반복 작업.....	81
XR Device Simulator	82
XD Device Simulator 설치.....	83
Unity 프로파일러	84
빌드 및 배포	88
혼합 현실 및 공간 컴퓨팅 애플리케이션	89
사용자 상호 작용 및 인터페이스 디자인	89
공간 인식 및 물리	90
크로스 플랫폼 개발 전략.....	90
환경 디자인과 몰입.....	90

MR 템플릿.....	91
MR Interaction Setup 프리팹	92
AR Session.....	92
AR Plane Manager.....	93
패스스루	94
MR 템플릿 테스트.....	95
Apple Vision Pro를 사용한 공간 컴퓨팅.....	96
Unity에서 visionOS 시작하기.....	97
visionOS 및 Unity 소개.....	99
상호 작용.....	102
3D 터치 및 TouchSpace	103
골격 기반 손 트래킹	103
완전 몰입형 VR	104
MR 앱	107
Unity PolySpatial.....	109
그래픽스 및 시뮬레이션	109
볼륨 카메라.....	110
바운디드 카메라	111
언바운디드 카메라	112
Play To Device	112
윈도우형 앱.....	113
visionOS 프로젝트 템플릿.....	114
더 많은 리소스	116
프로페셔널 트레이닝 서비스	116
Unity에서 AR 구현하기	117
용어집.....	119

들어가는 말

몰입감 넘치는 가상의 세계를 여행하거나, 차원을 순간 이동하거나, 디지털 요소를 현실 세계와 결합하는 등, 크리에이터는 VR(가상 현실)과 MR(혼합 현실)을 통해 상상을 현실로 만들 수 있습니다.

유니티에서 제작한 이 종합 가이드를 활용하면, 초보 크리에이터와 숙련된 개발자 모두 Unity를 사용해 VR 경험과 MR 경험(또는 통칭 'XR')을 제작하는 복잡한 과정을 살펴보고 이해할 수 있습니다.

Unity는 XR 경험 제작 부문을 선도하는 플랫폼으로, 이는 다음과 같은 2024년 성과에서도 확인할 수 있습니다.

- ‘첫 주에 꼭 해 봐야 할’ Vision Pro용 Apple Arcade 게임 중 50% 이상이 Unity로 제작되었습니다.¹
- 2024년 2월 기준으로 가장 인기 있는 Quest 경험 중 3분의 2가 Unity로 제작되었습니다.²
- 2024년 2월 기준으로 가장 많이 판매된 Quest 게임 중 70% 이상이 Unity로 제작되었습니다.³
- 2023년 최고 매출을 기록한 Steam VR 경험의 60% 이상이 Unity로 제작되었습니다.⁴

1 출처: Apple Inc.

2 출처: SteamDB

3 출처: SteamDB

4 출처: SteamDB

여기서는 몰입도 높은 인터랙티브 현실을 제작하는 데 필수적인 툴, 방법론, 기법에 대해 알아봅니다. 환경 구성부터 직관적인 상호 작용의 구현까지, 생동감 있는 VR 및 MR 애플리케이션 개발에 필요한 팁과 가이드를 확인하실 수 있습니다.

현실과 상상이 만나고 실재와 허상의 경계가 허물어지는 세계로 들어가 보세요. Unity와 함께 매력적이면서도 새로운 아이디어를 선사하는 몰입형 현실 제작 여정을 시작하세요.

주요 저자

다니엘 스트링거는 풍부한 경험을 보유한 3D 아티스트이자 Unity 개발자 겸 콘텐츠 크리에이터로, 20년 동안 다양한 업계에서 전문성을 쌓았습니다. 다니엘은 20년 동안 게임, 교육 및 시뮬레이션, 건축 시각화, 영화, VFX 분야에 종사해 왔습니다. 다니엘은 수상 경력에 빛나는 VR 애플리케이션을 개발한 전적이 있으며, 영국의 의료 분야와 상업용 라디오 분야에 특히 기여하고 있습니다.

다니엘의 콘텐츠에 대한 내용은 주로 VR에 중점을 둔 Unity 관련 주제를 전문으로 다루는 [YouTube](#)와 [Udemy](#)에서 더 자세하게 확인할 수 있습니다.

도움을 주신 유니티 직원 분들

이 가이드는 XR 제품, 제품 마케팅, 콘텐츠 마케팅 팀에서 근무하는 많은 유니티 직원의 참여로 제작되었습니다.

아담 액슬러, 댄 마르티네즈, 에두아르도 오리즈, 아이작 시아, 케빈 샘플, 레아 마틴, 매트 푸아드, 미루나 두미트라스쿠, 샘 장, 산티 자카리아, 트리샤 베커 님께 감사의 인사를 전합니다.



왼쪽은 Population ONE(개발: BigBox VR), 오른쪽은 Job Simulator(개발: Owlchemy Labs)



업계 용어

Unity 기반 콘텐츠 개발에서 **XR**은 다음과 같은 애플리케이션 유형을 포함하는 포괄적인 용어입니다.

- **MR (혼합 현실):** MR이란 일반적으로 비디오 시스루 디스플레이를 사용하는 헤드셋 시스템을 말하며, 사용자는 카메라로 캡처된 현실 세계가 디지털 방식으로 가상 콘텐츠와 혼합된 모습을 보게 됩니다.
- **VR(가상 현실):** VR은 사용자를 전적으로 시뮬레이션된 환경에 빠져들게 하는 완전한 몰입형 디지털 경험을 제공하며, 헤드셋이나 특수 기기를 사용하는 경우가 많습니다.
- **AR (증강 현실):** AR은 투명 안경이나 바이저와 같은 옵티컬 시스루(optical see-through) 디스플레이를 통해 사용자에게 바로 보이는 현실 세계 위에 가상의 콘텐츠가 오버레이되는 시스템을 지칭합니다.
- **공간 컴퓨팅:** 디지털 정보가 물리적 월드에 원활하게 통합되어 가상 환경과 실제 환경이 혼합된 몰입형 경험을 창출하는 프로세스와 기술을 지칭합니다.

VR과 AR의 범위를 엄격하게 나누기는 어려우며, 특정 기기의 기능과 애플리케이션의 의도에 따라 MR은 두 가지 방식 사이의 어딘가에 속합니다.

이와 같은 용어는 몰입형 기술의 스펙트럼을 나타내며, 각 기술은 게임 개발과 그 밖의 분야에서 독특한 경험을 선사하고 창의적으로 탐구할 기회를 제공합니다.



위에서 아래 순: VR에서 실행 중인 Unity **URP 3D** 샘플, Meta Quest 3 헤드셋에서 플레이 중인 Demeo Battles(개발: Resolution Games), Unity의 visionOS 공간 지원에 대한 인터랙티브 샘플, 휴대폰의 카메라 피드에 요소를 추가하는 AR 애플리케이션

현실 유형 선택: VR, AR 또는 MR

이 섹션에서는 어떤 종류의 몰입형 게임을 제작할지 결정할 때 고려할 만한 요소를 살펴봅니다.



XR 게임 개발과 관련된 다양한 머리글자어와 선택 가능한 요소

몰입형 경험의 목표

구현하려는 몰입도에 따라 선택이 달라질 수 있습니다. VR은 완전한 몰입감을 제공하는 반면, AR과 MR은 현실의 맥락에 콘텐츠를 더하는 정도입니다.

여기서 내린 판단에 따라 VR 헤드셋, AR 호환 스마트폰 및 안경, MR 기기 등의 타겟 하드웨어도 결정됩니다.

MR 관련 참고 사항: MR을 지원하는 XR 헤드셋의 중요한 기능은 **패스스루**입니다. 패스스루는 일반적으로 전면에 장착된 카메라를 통해 사용자에게 현실 세계의 라이브 피드를 표시하는 헤드셋의 기능을 지칭하며, 이를 통해 사용자는 헤드셋을 착용한 상태에서도 실제 주변 환경을 볼 수 있습니다.

현실 세계에 디지털 요소가 오버레이되는 MR 앱을 사용하려면 패스스루가 필요합니다. 패스스루를 사용하면 헤드셋을 벗지 않고 주변 환경과 상호 작용할 수 있으므로 사용자의 안전과 편의성도 향상됩니다. 이 기능은 완전한 가상 몰입과 사용자가 있는 실제 물리적 공간 사이의 간극을 메워 줍니다.

사용자 상호 작용 및 참여

사용자가 애플리케이션에서 무엇과 상호 작용하기를 바라나요? 매우 중요한 질문입니다. AR과 MR이 물리적 환경과 가상 환경을 혼합한다면, VR은 사용자를 디지털 세계 내부에 격리합니다.

내러티브 및 게임플레이 요소

내러티브와 게임플레이 메카닉은 중요한 역할을 합니다. VR이 스토리 중심의 완전한 몰입형 경험에 적합한 반면, AR/MR은 위치 기반 게임플레이를 향상할 수 있습니다.

활용 목적

결정에 영향을 미치는 것은 실제 적용 방식입니다. AR/MR은 현실 세계와의 통합이 중요한 교육 시뮬레이션이나 교육 경험에 적합합니다.

개발 리소스 및 제약

개발자의 전문성, 리소스, 기술적 제약도 선택에 영향을 미칩니다. 현실 유형에 따라 다른 개발 접근 방식이 필요하기 때문입니다.



I Expect You To Die 2: The Spy and the Liar, 수상 경력에 빛나는 VR 게임의 속편(개발: Schell Games)



모션 관련 고려 사항

또 다른 요소로 모션 역시 고려해야 하며, 그 이유는 다음과 같습니다.

- **사용자 불편 방지:** 과도하거나 비현실적인 모션은 불편함이나 멀미를 유발할 수 있습니다. 앱은 항상 부드러운 움직임으로 작동하여 방향 감각 상실과 멀미를 방지할 수 있어야 합니다.
- **상호 작용 설계:** 사용자가 가상 공간 내에서 직관적으로 이동하고 상호 작용할 수 있어야 합니다. 경험 유형(예: 게임, 교육, 탐험)에 기반하여 이동 메카닉을 설계하세요.
- **성능 제약:** 빠르거나 복잡한 모션은 더욱 뛰어난 처리 성능을 필요로 하며, 이는 앱 성능에 영향을 미칠 수 있습니다. 앱이 타겟 기기에서 원활하게 실행되도록 개발 사이클 전반에 걸쳐 성능을 테스트하고 최적화하세요(이 가이드 후반부의 [테스트 및 반복 작업 섹션](#) 참조).
- **물리적 공간 제약:** 사용자의 물리적 공간을 고려하세요. 앱을 가정에서 사용하나요, 아니면 직장에서 교육용으로 사용하나요? 다양한 위치에서 안전하게 사용할 수 있도록 앱을 디자인하세요.
- **사용자 접근성:** 모든 사용자가 자유롭게 이동할 수 있는 것은 아닙니다. 거동이 불편한 사용자를 위한 대체 상호 작용 방법을 제공하여 더 많은 사용자가 앱을 이용할 수 있도록 지원하세요.

접근성이 뛰어나고 편안하며 흥미로운 AR/VR 경험을 개발할 수 있도록 이러한 측면을 파악하고 그에 맞게 대처하세요.



게임 또는 산업용 XR 애플리케이션에 따라 컨트롤, 모션, 물리적 공간 컨텍스트가 달라집니다.

XR 월드 구축하기

월드 구축은 시각적으로 뛰어난 환경을 구현하는 작업에 한정되지 않습니다. 스토리를 선보이고, 감정을 전달하며, 사용자의 마음을 사로잡는 몰입형 월드를 제작하는 것이 중요합니다. 이 섹션에서는 XR 월드를 구축하는 데 필요한 구성 요소를 살펴봅니다.

내러티브 개발

모든 가상 월드의 개발은 기반이 되는 강력한 컨셉으로부터 시작됩니다. 독특한 배경, 설득력 있는 스토리, 플레이어가 풀어나갈 수 있는 흥미로운 미스터리 등이 컨셉이 될 수 있습니다. 컨셉은 시각 디자인부터 인터랙티브 요소에 이르는 모든 요소의 기반이 되어야 합니다.

환경 스토리텔링

목적에 부합하는 디자인

가상 환경 속의 모든 요소는 스토리 진행, 분위기 고조, 단서 및 도전 과제 제공 등 분명한 목적을 가져야 합니다. 주의를 산만하게 하는 요소나 전반적인 경험에 도움이 되지 않는 요소는 사용하지 마세요.

시각적 단서와 청각적 단서

시각적 단서와 청각적 단서를 사용하여 스토리와 배경 설정을 사용자에게 안내하세요. 조명, 사운드스케이프 (soundscape), 건축 디자인으로 주의를 유도하고 분위기를 조성하며 기본 스토리에 대한 힌트를 제시할 수 있습니다.

인터랙티브 요소

월드와 스토리에 대해 더 많은 정보를 주는 인터랙티브 요소를 통합하세요. 역사의 일부를 알려주는 고대 유물, 1인칭 시점의 일기, 비밀 지역이나 설화를 잠금 해제하는 환경 퍼즐 등이 이러한 인터랙티브 요소에 해당합니다.



모든 주요 플랫폼을 지원하는 MR 게임인 Demeo(개발: Resolution Games)

사용자 참여 전략

디테일이 주는 몰입감

환경을 구성하는 작은 디테일에 주목하세요. 텍스처, 음향 효과, 인터랙티브 오브젝트가 아트 관련 의도에 부합해야 합니다. 전체적인 환경과 잘 어우러지고 일관된 경험을 구현하여 플레이어의 몰입도와 참여도를 높이세요.

상세한 탐색 유도

월드를 탐험하는 사용자에게 보상을 부여하세요. 숨겨진 지역, 비밀 설화, 이스터 에그가 있으면 사용자는 주요 경로를 넘어 다른 곳도 살펴보게 되며, 따라서 사용자의 경험이 더 풍성해지고 월드에 대한 참여도도 높아집니다.

적응적 난이도

사용자의 성과에 따라 도전 과제를 조정하는 적응적 난이도 메커니즘을 구현하세요. 그러면 기술 수준과 상관없이 모든 사용자가 까다롭기는 해도 포기할 정도는 아닌 흥미로운 경험을 즐기도록 할 수 있습니다.

소셜 상호 작용

가능한 경우라면 소셜 상호 작용을 월드에 통합하세요. 사용자들이 함께 월드를 탐색하고 퍼즐을 푸는 멀티플레이어 요소를 사용하면 참여도를 높이고 더 역동적인 경험을 구현할 수 있습니다.



Owlchemy Labs의 Job Simulator에서는 디테일이 풍부한 환경에서 점점 난이도가 높아지지만 재미있는 과제를 해결하게 됩니다.

반복적 디자인

피드백을 사용하여 월드를 반복적으로 개선하세요. 내러티브 플로를 조정하고 특정 영역을 다시 디자인하여 탐색 방식을 개선하거나 인터랙티브 요소를 더 추가하여 참여도를 높일 수 있습니다.

월드를 구축할 때 이러한 전략을 적용하면 VR 및 MR 경험의 품질과 효과를 향상할 수 있습니다. 내러티브 개발, 환경 스토리텔링, 사용자 참여에 집중하면 단순한 게임이나 애플리케이션이 아니라 몇 번이고 다시 사용자의 참여를 유도하는 매력적인 월드를 제작할 수 있습니다.

다음 전자책에서 Unity의 게임 디자인과 레벨 디자인에 대해 더 자세히 알아보세요.



[다운로드](#)



[다운로드](#)

Unity 사용이 처음이라면

Unity Learn은 에디터를 탐색하고 사용하는 방법을 알아보는 다양한 무료 리소스를 제공합니다. 신규 사용자라면 [세 가지 학습 길잡이 교육 과정](#)인 Unity 필수 과정, 주니어 프로그래머, 크리에이티브 코어부터 시작하는 것이 좋습니다. 이 기본적인 학습 길잡이 교육 과정을 완료한 후에는 [VR 개발 학습 길잡이](#)를 수강해 보세요.

유나이티드와 GDC의 심층 세션과 기조 연설을 시청하여 XR 분야와 유니티의 최신 개발 동향을 확인해 보는 것도 좋습니다. 다음 링크를 먼저 살펴보세요.

- [GDC 2024의 유니티](#)(영문)
- [유나이티드 2023](#)(영문)
- [XRI 세션 | XR 개발자의 밤 2024](#)(영문)
- [XR 로드맵 2024 | XR 개발자의 밤 2024](#)(영문)
- [손을 사용한 입력 및 OpenXR 개발 | XR 개발자의 밤 2024](#)(영문)
- [XR 합성 레이어 | XR 개발자의 밤 2024](#)(영문)
- [혼합 현실 저작 및 시뮬레이션 | XR 개발자의 밤 2024](#)(영문)
- [AR Foundation | XR 개발자의 밤 2024](#)(영문)

XR용 에셋 제작

XR 게임용 아트 에셋을 제작하거나 소싱할 때 중요하게 고려해야 할 사항이 있습니다. 작은 프랍(prop)부터 캐릭터까지, 이러한 팁을 따르면 개발 사이클 후반부에 드는 시간을 절약할 수 있습니다.

컨셉 및 디자인

우선 제작하려는 콘텐츠에 대한 명확한 컨셉을 정하세요. 아이디어를 스케치하고 3D 360도 공간에서 에셋이 어떻게 보일지 고려하세요. 이러한 컨셉은 환경의 형태부터 퍼즐이 작동하는 방식까지 아이디어를 구체화하는 데 도움이 됩니다. 스케치를 작성하면 다른 작업자에게 초기에 아이디어를 전달할 수 있으므로, 여러분과 공동 작업자가 더 풍부한 정보를 기반으로 의사 결정을 내리고 효과적으로 계획할 수 있습니다.

컨셉은 간단한 손그림으로 작성해도 되며, 특히 초기 단계에는 그런 방법을 통해 상호 작용이나 비주얼을 빠르게 계획할 수 있습니다. 대규모 스튜디오에는 아이디어를 실현하는 컨셉 아티스트가 있기도 하며, 개인이나 소규모 팀에서는 AI를 사용해 아이디어를 표현하기도 합니다.

모델링

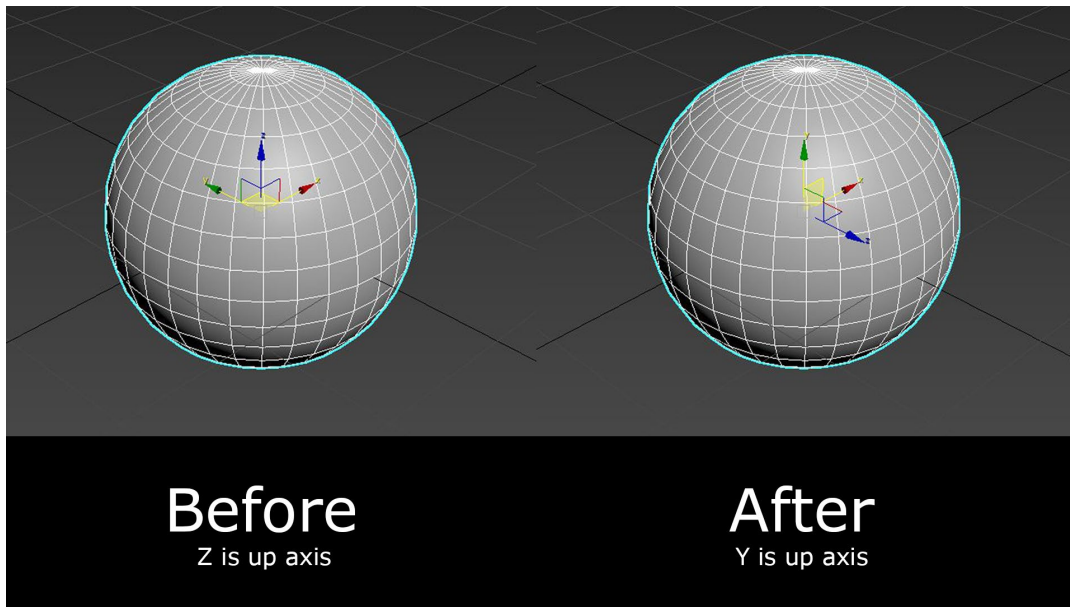
3D 모델링 소프트웨어(Blender, Autodesk Maya, 3ds Max 등)를 사용하여 모델을 제작할 수 있습니다. 성능을 고려해 폴리 카운트(폴리곤 수)를 염두에 두어야 하며, 모바일 AR의 경우에는 더욱 그렇습니다. 처음에는 하이 폴리곤 모델을 제작한 다음 나중에 최적화하면 됩니다.

피벗 포인트

피벗 포인트는 모델 회전 및 크기 조정의 중심 역할을 합니다. 피벗 포인트를 적절히 배치하면 애니메이션 프로세스를 간소화하여 움직임과 변형을 더욱 자연스럽게 직관적으로 구현할 수 있습니다. 예를 들어 피규어 지지대에 피벗 포인트를 사용하면 사람이 발을 디디는 것처럼 사실적이고 안정적인 회전을 구현할 수 있습니다. 피벗 포인트의 위치는 애니메이션과 게임 개발 모두에서 환경과의 정확한 상호 작용에 중요합니다.

Unity의 3D 씬에서 작업할 때 Y축은 수직 이동 또는 높이를 나타냅니다. 이 방향은 3D 그래픽스 환경 대부분에 적용되는 표준 규칙으로, X축은 수평 이동(좌우), Y축은 수직 이동(상하), Z축은 깊이(전후)를 나타냅니다. 이 축 시스템을 이해하면 Unity 내에서 오브젝트를 올바르게 배치하고, 회전하고, 크기를 조정할 수 있습니다.

3D 모델링 소프트웨어에서 작업할 때 모델의 피벗 포인트를 설정하여 방향을 Unity의 방향과 일치시킬 수 있습니다. 아래 예시에서는 3ds Max에서 Z축을 위쪽 방향으로 사용하는 피벗 포인트의 설정 방법을 소개합니다.



위쪽 방향 축 조정

Unity에서는 다음 두 단계를 통해 모델의 방향을 수정할 수도 있습니다.

1. 계층 구조(Hierarchy)에서 빈 게임 오브젝트를 만듭니다.
2. 이 게임 오브젝트 내에 모델을 중첩하고 모든 축에서의 위치를 0,0,0으로 설정하여 정중앙에 피벗 포인트를 둡니다.

이 방법을 사용하면 모델이 모든 축에 걸쳐 0에서 종립적으로 회전하는 부모 게임 오브젝트에 고정됩니다. 또한 Z축이 정면을 향하도록 모델의 방향이 올바르게 설정되며 모든 축의 스케일도 1로 균일하게 유지됩니다. 임포터에서 Bake Axis Conversion 옵션을 사용해 볼 수도 있습니다.

텍스처링

텍스처를 적용하면 모델에 사실적이거나 세련된 느낌을 줍니다. 디테일이 돋보이는 텍스처에는 UV 매핑을 사용하는 것이 좋습니다. 게임의 심미적인 측면에 따라 PBR 텍스처가 필요할 수도 있습니다. Unity의 기본 [URP\(유니버설 렌더 파이프라인\) 셰이더](#)는 물리 기반이므로 머티리얼의 물리적 특성을 사실적으로 시뮬레이션합니다. 이 셰이더는 텍스처, 반사도, 거칠기 등의 요소를 고려하여 광원이 표면과 상호 작용하는 방식을 정확하게 묘사하는 알고리즘을 사용합니다. Unity의 PBR(물리 기반) 셰이더에는 일반적으로 두 가지 주요 컴포넌트가 포함됩니다. 메탈릭-거칠기 워크플로와 스페큘러-광택도 워크플로에서는 각각 다른 측면의 머티리얼 프로퍼티를 처리합니다. 그렇게 하면 더욱 생생하고 역동적인 시각적 형상을 구현하여 Unity 기반의 게임이나 시뮬레이션에서 3D 씬과 오브젝트의 사실감을 향상할 수 있습니다.

트림 시트

트림 시트는 3D 모델링과 게임 개발에 사용되는 텍스처링 툴로, 몰딩, 가장자리, 테두리 등의 다양한 트림 패턴과 디테일이 포함된 단일 텍스처로 구성됩니다. 아티스트는 이러한 트림을 3D 모델의 다양한 부분에 적용하여 여러 가지 고유 텍스처를 사용하지 않고도 복잡한 디테일을 추가할 수 있습니다. 이 방법은 필요한 개별 텍스처의 수를 줄이면서도 시각적으로 복잡하고 다양한 디자인을 모델에 구현할 수 있으므로 효율적으로 게임 성능을 최적화할 수 있습니다.



환경의 다양한 메시에 사용되는 트림 시트



리깅 및 애니메이션

캐릭터와 같이 모델에 움직임이 필요하고 메시가 변형되어야 하는 경우, 모델을 골격 구조로 리깅하고 애니메이션화할 수 있습니다.

1인칭 VR 앱에서는 아바타의 손을 설정해야 합니다. [Animation Rigging 패키지](#)에 포함된 Runtime IK 기능을 사용하면 팔의 뼈대를 설정하고 TwoBoneIK 제약 조건을 적용하여 손을 트래킹 컨트롤러의 위치로 정렬할 수 있습니다.

팔꿈치는 보통 위치가 트래킹되지 않으므로 흥미로운 과제가 될 수 있습니다. 팔꿈치의 위치를 점차적으로 추론할 수 있다면 TwoBoneIK 제약 조건이 Hint 트랜스폼을 입력으로 사용하여 팔꿈치 뼈대를 정렬할 수 있습니다.



Arashi: Castles of Sin - Final Cut(개발: Endeavor One)에서는 VR 손과 팔에 TwoBoneIK 시스템을 사용합니다.

모델 최적화

Unity에서 모델의 성능을 고려해 폴리곤 개수와 파일 크기를 줄이고 텍스처의 크기를 효율적으로 조정하여 최적화해야 합니다. 모델의 메시 디테일은 노멀 맵을 만들어 캡처할 수 있습니다.

노멀 맵 3D는 표면의 픽셀 각각에 노멀(수직 벡터)의 방향을 인코딩하는 텍스처입니다. 이렇게 하면 모델의 폴리곤 개수를 늘리지 않고도 범프나 인덴테이션과 같이 디테일이 강조된 표면 텍스처를 구현할 수 있습니다. 이는 광원이 표면과 상호 작용하는 방식을 변경하여 평면에 깊이와 디테일이 있는 듯한 착시를 일으켜 사실감을 높입니다.



3D 모델링에 필요한 노멀 맵을 생성하는 최적의 방법은 다음과 같습니다.

- **하이 폴리곤 모델에서 추출:** 디테일이 강조된 하이 폴리곤 모델과 로우 폴리곤 버전을 만듭니다. 그런 다음 Blender 또는 Substance Painter와 같은 소프트웨어를 사용하여 하이 폴리곤 모델의 표면 디테일을 로우 폴리곤 모델에 노멀 맵으로 베이킹합니다.
- **2D 이미지 에디터 사용:** 플러그인이 설치된 Adobe Photoshop이나 GIMP와 같은 툴을 사용하여 그레이스케일 하이트 맵을 노멀 맵으로 전환할 수 있습니다. 이 방법은 직물이나 돌과 같은 표면 텍스처에 유용합니다.
- **전용 노멀 맵 제너레이터:** CrazyBump 또는 NormalMap Online과 같은 소프트웨어를 사용하면 2D 이미지에서 노멀 맵을 생성할 수 있습니다.
- **사진 측량:** 사진 측량 기법을 통해 실제 오브젝트를 디지털 모델로 변환하여 노멀 맵을 만들 수 있습니다.



SUPERHOT VR(개발: SUPERHOT Team)에는 테마에 어울리면서도 게임플레이에 필요한 성능을 제공하는 로우 폴리곤 스타일이 적용되어 있습니다.

익스포트

Unity와 호환되는 포맷으로 모델을 익스포트합니다(FBX 권장).

Unity에서 AR과 VR을 작업할 때는 3D 모델의 익스포트 스케일이 중요합니다. 스케일에 따라 모델이 가상 월드에서 작동하고 현실 세계의 차원과 상호 작용하는 방식이 결정되며, 특히 물리적 월드와의 상관관계가 중요한 AR에서는 더욱 그렇습니다. 스케일이 일치하지 않으면 비현실적이거나 무리한 상호 작용이 발생할 수

있습니다. 모델의 스케일이 Unity의 유닛과 일치하고, 원하는 실제 또는 가상 환경의 스케일과 일치하도록 해야 합니다.

Unity의 측정 스케일은 1유닛이 정확히 1미터에 해당하도록 직관적으로 설계되어 있습니다. 임포트한 에셋의 스케일을 측정하려면 씬에 큐브 게임 오브젝트를 추가합니다. 1m x 1m x 1m 크기의 이 큐브는 에셋의 스케일을 평가할 때 즉각적이고 정확한 기준이 됩니다.

환경과 프랍을 제작할 때 핵심은 미관과 성능이 균형을 이루도록 하고, 부드러우면서도 흥미로운 AR/VR 경험을 구현하는 것입니다.



ProBuilder로 레벨을 프로토타이핑한 후, FBX Exporter를 사용해 3D 모델링 소프트웨어로 익스포트하여 미세 조정할 수 있습니다.

Unity의 XR: 주요 시스템 및 툴셋

렌더링 파이프라인

Unity의 렌더링 파이프라인은 다양한 플랫폼 및 성능 요구 사항을 충족할 수 있도록 풍부한 렌더링 기능을 제공합니다.



왼쪽은 Unity URP 3D 샘플의 정원 환경, 오른쪽은 HDRP 볼류메트릭 환경

왼쪽은 Unity URP 3D 샘플의 정원 환경, 오른쪽은 HDRP 볼류메트릭 환경

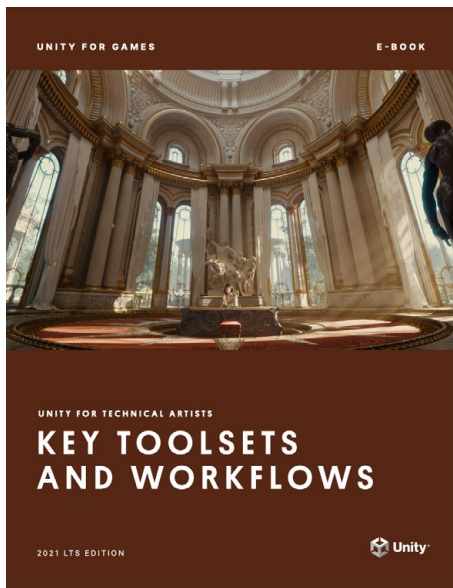
URP(유니버설 렌더 파이프라인)은 XR 애플리케이션 개발에 권장되는 파이프라인으로, 가벼우며 모바일부터 고사양까지 다양한 기기에 최적화되어 있습니다. URP는 최신 렌더링 기법, 최적화된 셰이더, 커스터마이징 가능한 포스트 프로세싱 효과를 제공합니다.

HDRP(고해상도 렌더 파이프라인)은 고사양 PC 및 콘솔 플랫폼을 위한 파이프라인으로, 물리 기반 렌더링, 고급 조명, 머티리얼, 복잡한 포스트 프로세싱을 적용해 최고급 비주얼 품질을 제공합니다. 그래픽 성능을 최대한으로 활용해도 상당히 높은 새로고침 속도를 낼 수 있는 타겟 하드웨어 경험에는 HDRP를 사용하세요.

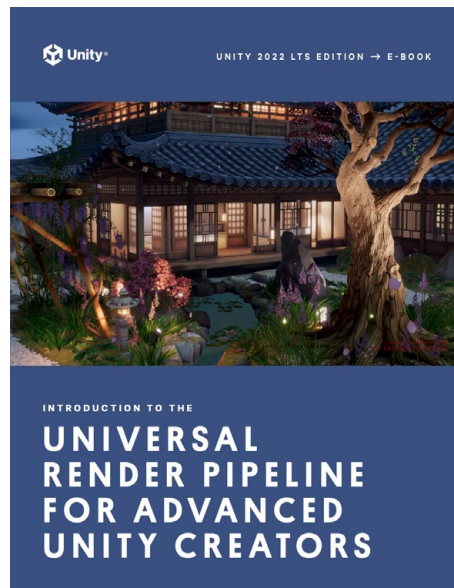
SRP(스크립터블 렌더 파이프라인)은 URP, HDRP, 그리고 특정 프로젝트 요구 사항 및 플랫폼 사양에 맞는 자체적인 커스텀 렌더 파이프라인의 기본 프레임워크입니다.

빌트인 렌더 파이프라인은 Unity의 레거시 파이프라인입니다. Unity 2022 LTS 이상에서는 일반적인 멀티플랫폼 게임 개발에 URP를 사용하는 것이 좋습니다.

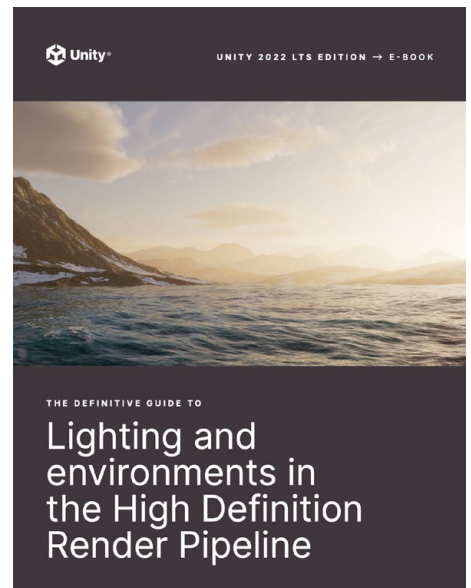
올바른 파이프라인의 선택은 프로젝트 요구 사항, 플랫폼 목표, 성능과 비주얼 품질 간의 균형에 따라 달라집니다. 각 파이프라인은 서로 다른 요구 사항을 충족할 수 있도록 다양한 수준의 성능 최적화 옵션과 시각적 충실도 옵션을 제공합니다.



[다운로드](#)



[다운로드](#)



[다운로드](#)

AR Foundation

AR Foundation은 몰입형 AR 경험 제작을 위한 강력한 통합 프레임워크를 제공하는 Unity 패키지입니다. 이 패키지를 사용하여 평면 검출, 월드 추적, 얼굴 감지, 환경 이해, 오클루전 등의 필수 AR 기능을 구현할 수 있습니다.

AR Foundation은 AR 기능을 위한 인터페이스를 포함할 뿐, 그 자체로 기능을 구현하지는 않습니다. 타겟 플랫폼에서 AR Foundation을 사용하려면 해당 플랫폼용으로 다른 제공업체에서 지원하는 플러그인 패키지도 있어야 합니다. Unity에서 공식적으로 지원하는 제공업체 플러그인은 다음과 같습니다.

- [Google ARCore XR 플러그인](#)(Android용)
- [Apple ARKit XR 플러그인](#)(iOS용)
- [Apple visionOS XR 플러그인](#)(visionOS용)
- [OpenXR 플러그인](#)(HoloLens 2용)
- [Unity OpenXR: Meta](#)(Meta Quest용)

AR Foundation에서 지원하는 기능과 플랫폼의 목록은 [이 페이지](#)에서 확인하세요.

AR Foundation은 여러 플랫폼에서 개발 프로세스를 간소화하는 API를 제공합니다. 플랫폼별 구현에 따른 복잡성을 추상화하므로, 사용자는 여러 기기에서 다양한 플레이어에게 도달하는 AR 경험을 제작할 때 창의적인 작업에 더 집중할 수 있습니다.



[AR 샘플](#)은 AR Foundation 기술 자료를 통해 확인할 수 있습니다.



XR Interaction Toolkit

XRI([XR Interaction Toolkit](#)) 패키지를 사용하면 잡기, 만지기, 이동, 오브젝트 조작과 같은 상호 작용에 대한 표준화된 접근 방식을 제공하여 XR 상호 작용을 효율적으로 개발할 수 있습니다.

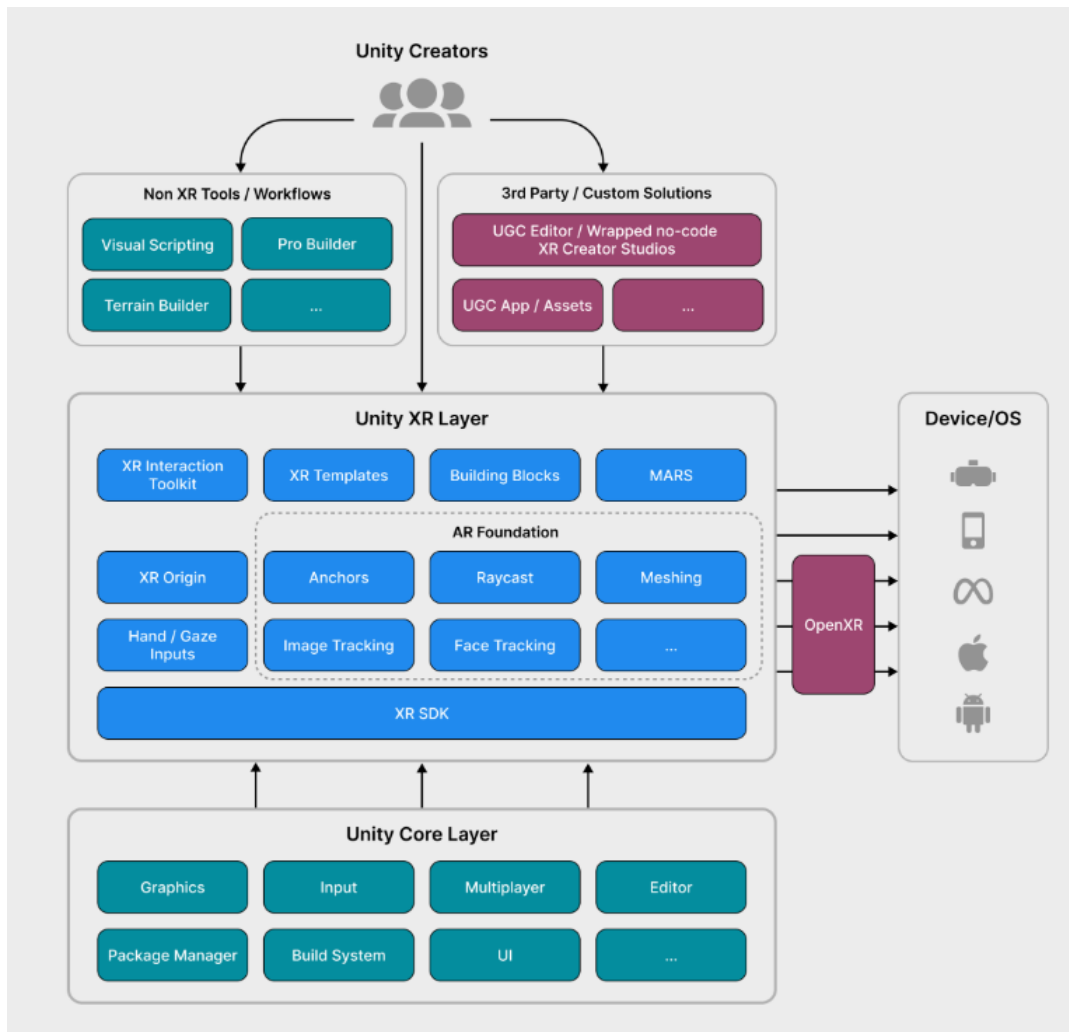
이 툴킷은 Unity 입력 이벤트에서 3D 및 UI 상호 작용을 사용할 수 있는 프레임워크를 제공합니다. 해당 시스템의 핵심은 기본 **Interactor** 및 **Interactable** 컴포넌트 세트와 이 두 가지 유형의 컴포넌트를 하나로 묶는 **Interaction Manager**입니다. Interaction Manager에는 이동과 비주얼 드로잉에 사용할 수 있는 컴포넌트도 포함됩니다.

Unity의 크로스 플랫폼 기능을 활용하는 XR Interaction Toolkit을 사용하면 개발자는 여러 플랫폼과 기기에서 서로 다른 상호 작용 모델을 복잡하게 관리할 필요 없이 XR 경험 디자인에 집중할 수 있습니다.

XR 개발용 SDK

Unity는 플러그인 프레임워크와 기능 툴 및 패키지 세트를 통해 XR 개발을 지원합니다. 개발자는 Unity Project Settings 창에서 [XR Plug-in Management](#) 패키지를 사용하여 타겟 XR 플랫폼에 사용 가능한 플러그인을 선택할 수 있습니다.

아래의 다이어그램은 현재 XR 플러그인 프레임워크가 어떻게 구성되고 플랫폼 제공업체 구현과 어떻게 통합되는지 보여 줍니다.



각 XR 플랫폼에는 일반적으로 해당 하드웨어의 기기 관련 기능을 모두 사용할 수 있는 자체 API가 있습니다. 예를 들면 Meta 기기에서는 [Meta XR All-in-One SDK를 활용합니다](#)(Unity 에셋 스토어에서 제공). 또한 OpenXR 호환 플랫폼에서는 비독점적인 기기(예: Meta 기기)의 기능도 지원하며, 이러한 기능은 OpenXR API를 통해 액세스할 수 있습니다.

하드웨어에 필요한 SDK에 액세스하는 방법은 기기의 개발자 가이드에서 확인하세요.

OpenXR

OpenXR은 Khronos가 개발한 로열티 프리 프레임워크로, 크로스 플랫폼 XR 경험을 제작할 수 있도록 설계되었습니다. Unity의 [OpenXR 플러그인](#)은 호환되는 OpenXR 런타임을 지원하는 모든 기기에서 작동합니다.

OpenXR은 XR 하드웨어를 위한 표준화된 인터페이스를 제공하므로 개발자는 기기별 코드를 작성할 필요 없이 다양한 XR 기기와 호환되는 애플리케이션을 제작할 수 있습니다. 따라서 다양한 플랫폼에서 XR 애플리케이션을 더 쉽게 개발 및 배포할 수 있어, 폭넓은 호환성을 보장하고 개발 시간과 노력을 줄일 수 있습니다.

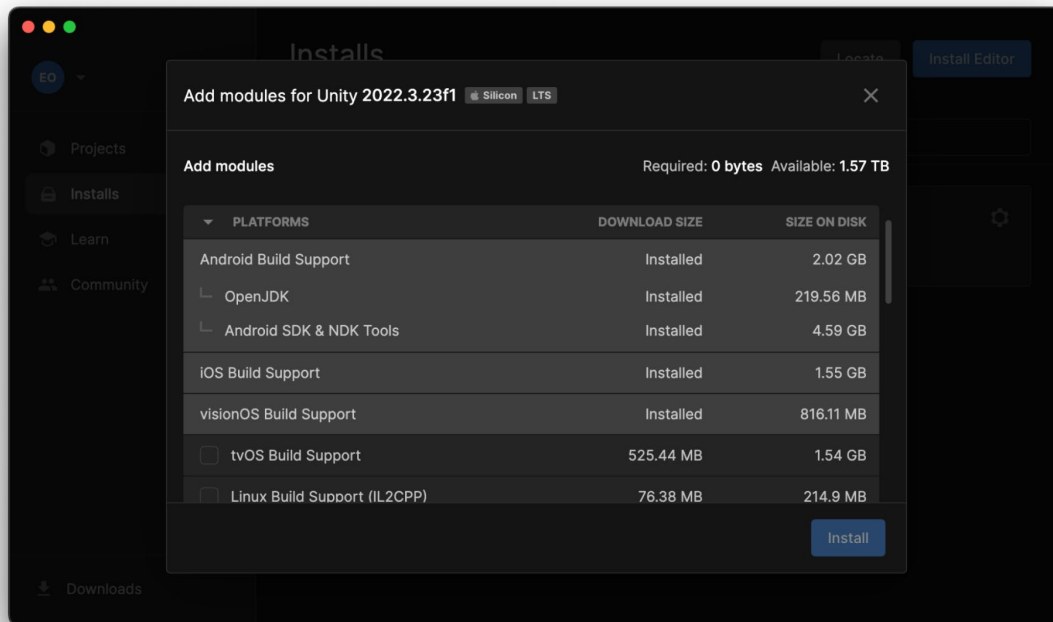
에디터 버전 및 모듈

이 가이드에서는 Unity 2022 LTS 에디터를 사용하므로 동일한 버전이나 최신 버전을 사용하는 것이 좋습니다.

기기	VR	MR/공간	AR	Unity 모듈	XR 플러그인 제공업체
Meta Quest 3	✓	✓		Android 빌드 지원	ARCore, Oculus, OpenXR
Meta Quest 2	✓	✓		Android 빌드 지원	ARCore, Oculus, OpenXR
Apple Vision Pro	✓			visionOS 지원, iOS 빌드 지원	ARKit
HTC Vive Focus 3	✓			Android 빌드 지원	Vive wave XR, OpenXR
HTC Vive Pro 2	✓			PC, Mac & Linux 스탠드얼론	Vive wave XR, OpenXR
PlayStation®VR2	✓			Sony 개발자 등록 필요	
Android 모바일 기기			✓	Android 빌드 지원	ARCore
Apple iPhone 및 iPad(A9 프로세서 이상)			✓	iOS 빌드 지원	ARKit
Microsoft HoloLens			✓	UWP(유니버설 Windows 플랫폼)	Windows Mixed Reality XR 플러그인
Magic Leap 2			✓	Android 빌드 지원	Magic Leap XR 플러그인
Meta Quest 3	✓	✓		Android 빌드 지원	ARCore, Oculus, OpenXR

Unity에서 지원하는 각 타겟 플랫폼에 대해 다운로드해야 하는 해당 플랫폼 빌드 모듈과 XR 플러그인은 다음과 같습니다.

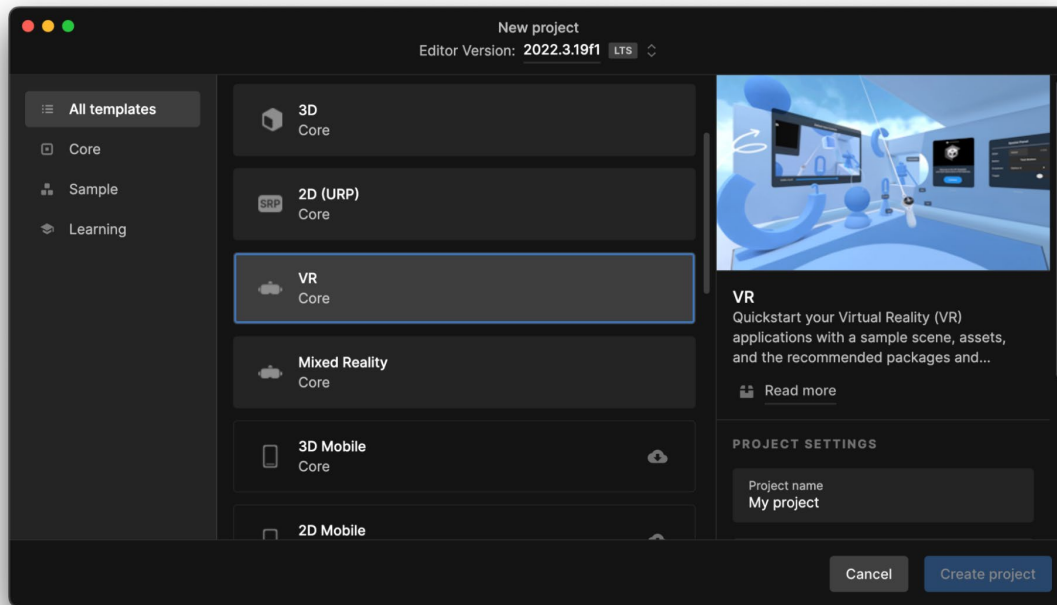
에디터를 설치하면 설치본에 모듈을 추가하라는 메시지가 표시됩니다. Meta Quest 2와 같은 모바일 VR 플랫폼을 지원하려면 타겟 모바일 기기에 따라 **Android Build Support**, **OpenJDK**, **Android SDK & NDK**, 또는 **iOS Build Support** 및 **visionOS Build Support** 모듈이 필요합니다.



VR 코어 샘플

Unity Hub에서 제공되는 VR 템플릿을 사용하여 새 VR 프로젝트를 시작하세요. 템플릿에는 샘플 씬, 에셋, 권장 패키지 및 설정이 포함됩니다. 템플릿을 받는 방법은 다음과 같습니다.

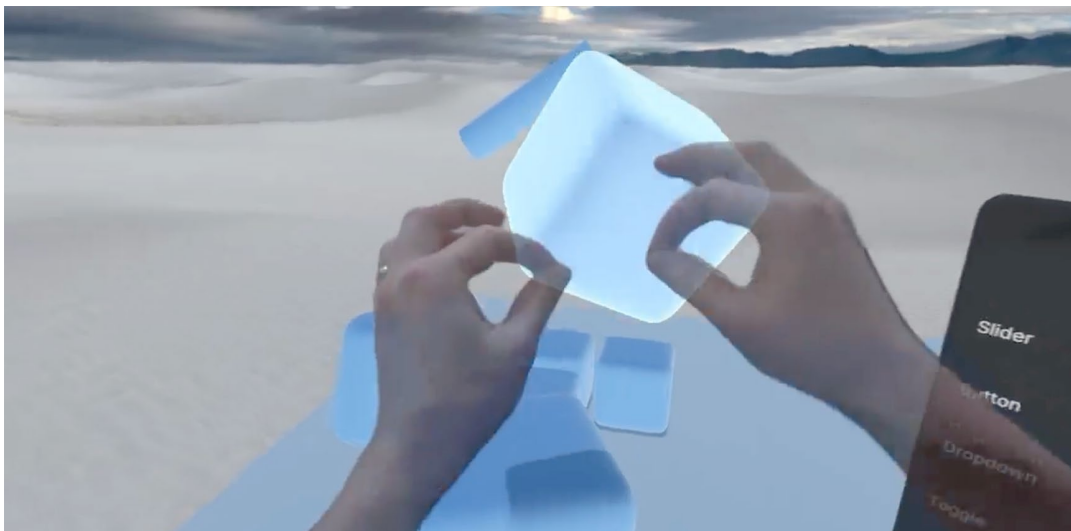
1. Unity Hub를 열고 파란색 **새 프로젝트** 버튼을 클릭합니다.
2. 에디터 버전을 사용 가능한 최신 **Unity 2022 LTS 버전으로 변경합니다.**
3. 템플릿 목록에서 **VR - Core** 템플릿을 찾습니다. **템플릿 다운로드** 버튼을 클릭합니다.
4. 프로젝트의 이름을 선택하고 프로젝트 설정에서 위치를 선택합니다. Unity가 열리고 시작 화면이 표시됩니다.



Unity Hub의 VR 코어 템플릿

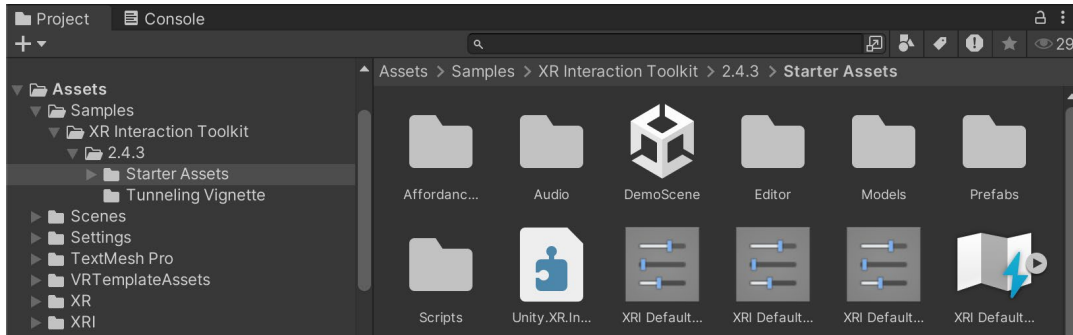
버전 3.0으로 업데이트

Unity Hub에서 이 템플릿을 기반으로 새 프로젝트를 시작하는 경우 기본적으로 XRI 2.5.x 버전을 사용 중일 수 있습니다(본 전자책의 공개 시점 기준). 이미 버전 3.0을 사용 중이라면 이 섹션을 건너뛰어도 됩니다. 가능하면 패키지 관리자에서 버전 3.0으로 업데이트하세요. 이 업데이트에는 visionOS와의 호환성 등의 여러 중요한 개선 사항과 기능이 포함되어 있습니다. visionOS의 Shared Space 입력을 처리하는 베스트 프랙티스를 확인 가능하며, 이 입력은 XRI의 새로운 Near-Far 및 Poke Interactor와 통합되어 있습니다. 관련 업데이트에 대한 자세한 내용은 기술 자료의 이 [페이지](#)에서 확인할 수 있습니다.



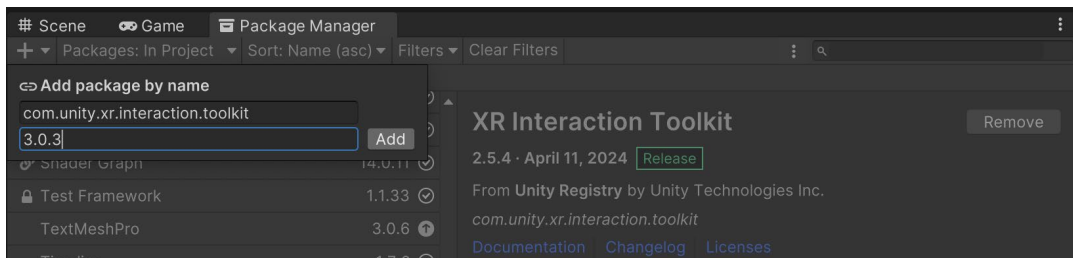
XRI 3.0 샘플에 포함된 visionOS 제스처 예시

버전 3.0을 설치하기 전에 이 [기술 자료](#) 페이지의 설명에 따라 버전 2.5.x에 속하는 Starter Asset 폴더를 제거하여 충돌을 방지하세요.



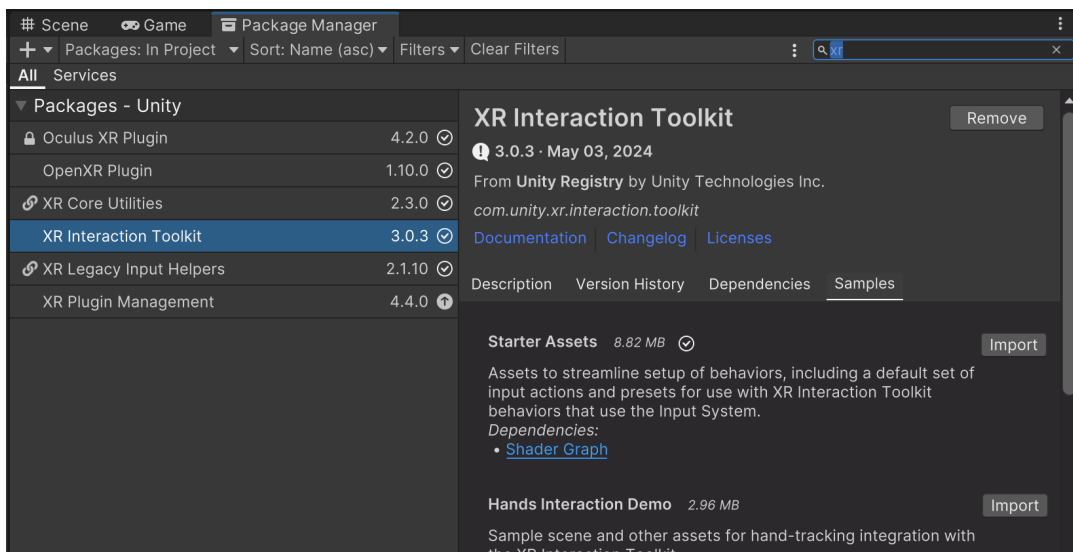
XRI 버전 3.0으로 업데이트하기 전에 Starter Assets 폴더 삭제

이제 업그레이드 옵션을 사용할 수 없는 경우, [기술 자료](#)의 설명에 따라 이름으로 패키지를 추가하고 해당 버전을 사용하겠다고 설정하여 XRI 3.0을 수동으로 설치할 수 있습니다.



XRI 패키지 및 샘플 수동 업데이트

XRI 3.0 버전이 설치되면 패키지 관리자를 통해 새로운 버전의 Starter Assets 샘플을 사용할 수 있습니다.

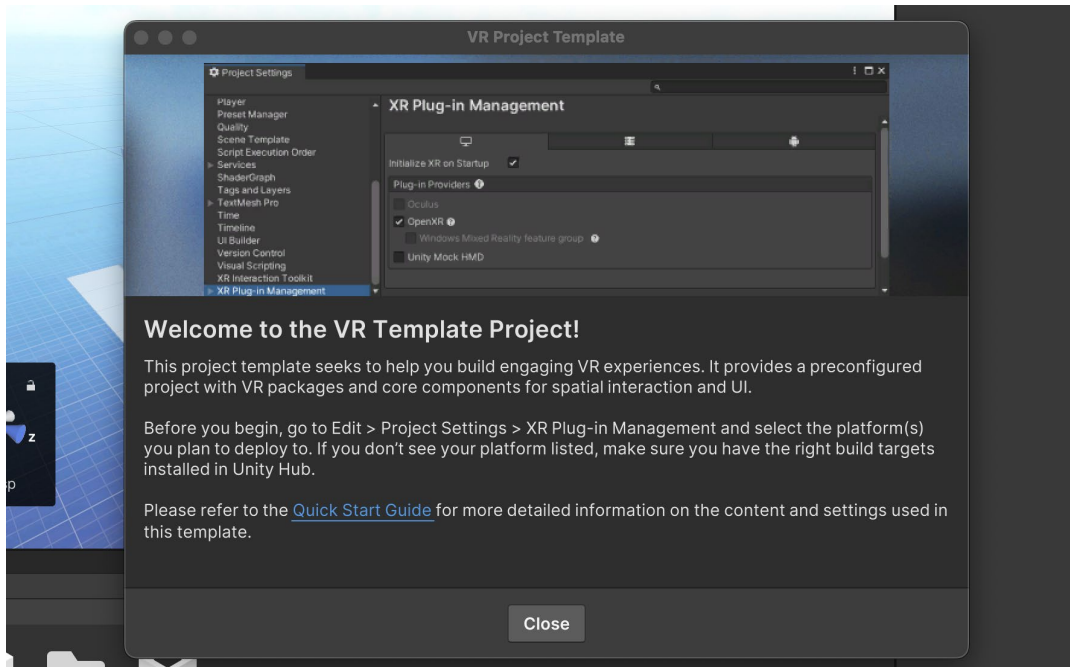


Starter Assets 버전 3.0 임포트



API를 사용하여 변경 사항이 있는 스크립트를 업데이트할지 묻는 메시지가 표시되면 Yes를 클릭합니다.

프로젝트를 탐색할 준비가 되면 **Welcome to the VR Template Project** 창이 표시되며, 이 창이 보이지 않는 경우 **Toolbar > Tutorials > Welcome** 다이얼로그에서 열면 됩니다.

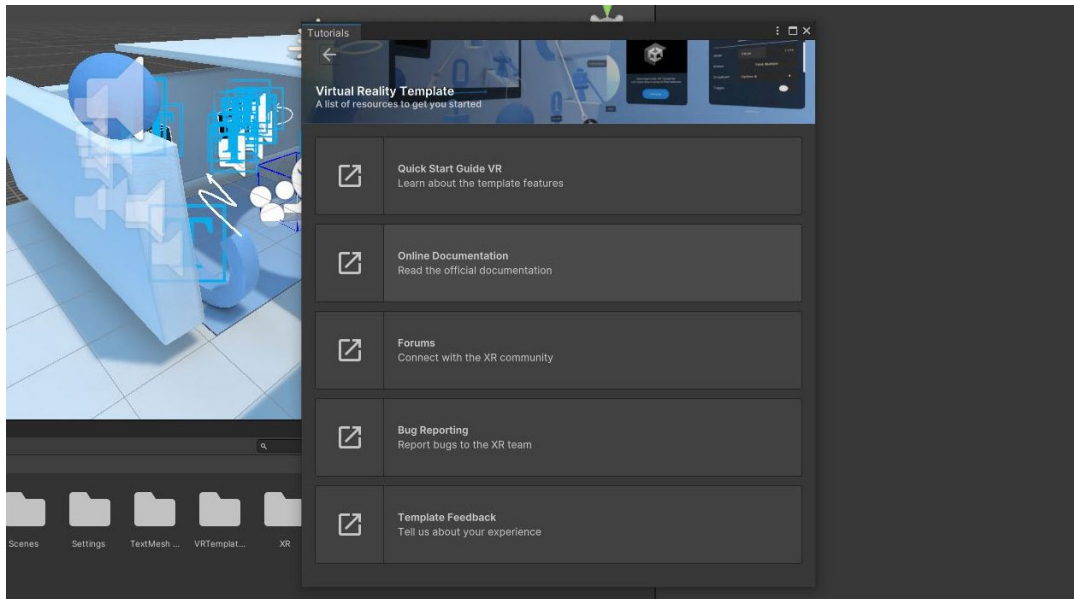


VR 템플릿 프로젝트의 시작 화면

시작 창이 열리고 **XR Plug-in Management** 설정으로 이동하여 배포할 플랫폼을 선택하라는 메시지가 표시됩니다. XR Plug-in Management 설정은 Project Settings 내에 있습니다. 플랫폼이 목록에 표시되지 않으면 Hub를 다시 방문하여 모듈을 추가로 설치하면 됩니다. XR Plug-in Management 설정에 대한 자세한 내용은 다음 섹션에서 확인할 수 있습니다.

튜토리얼 창

샘플 프로젝트에 대한 자세한 정보를 제공하는 튜토리얼 창도 표시됩니다.



튜토리얼 창

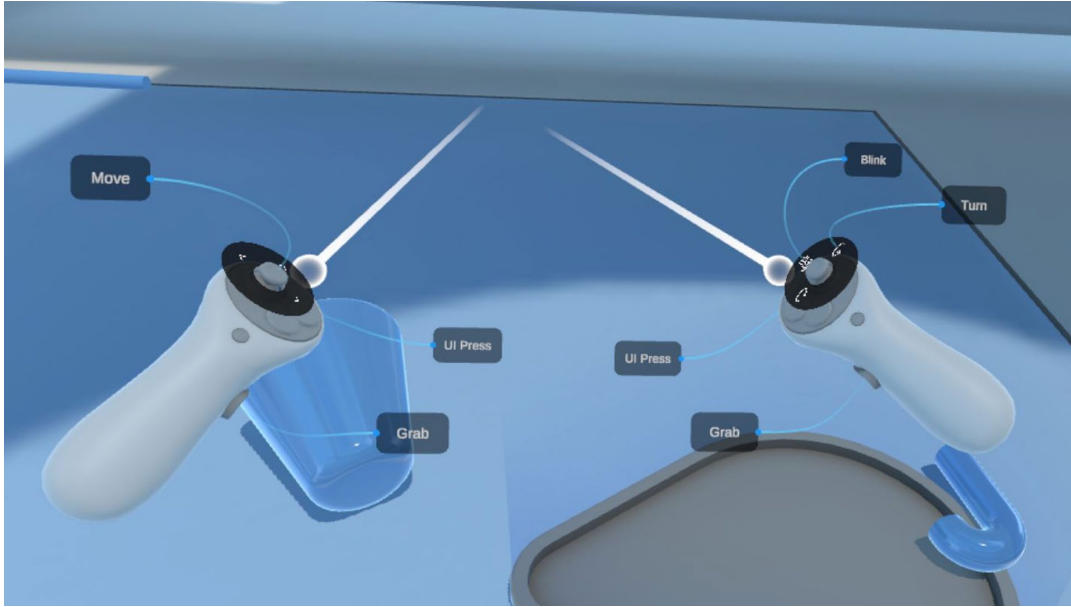
샘플 씬

VR 템플릿에 포함된 샘플 씬은 상호 작용과 탐색에 필요한 필수 요소로 구성되어 있습니다.



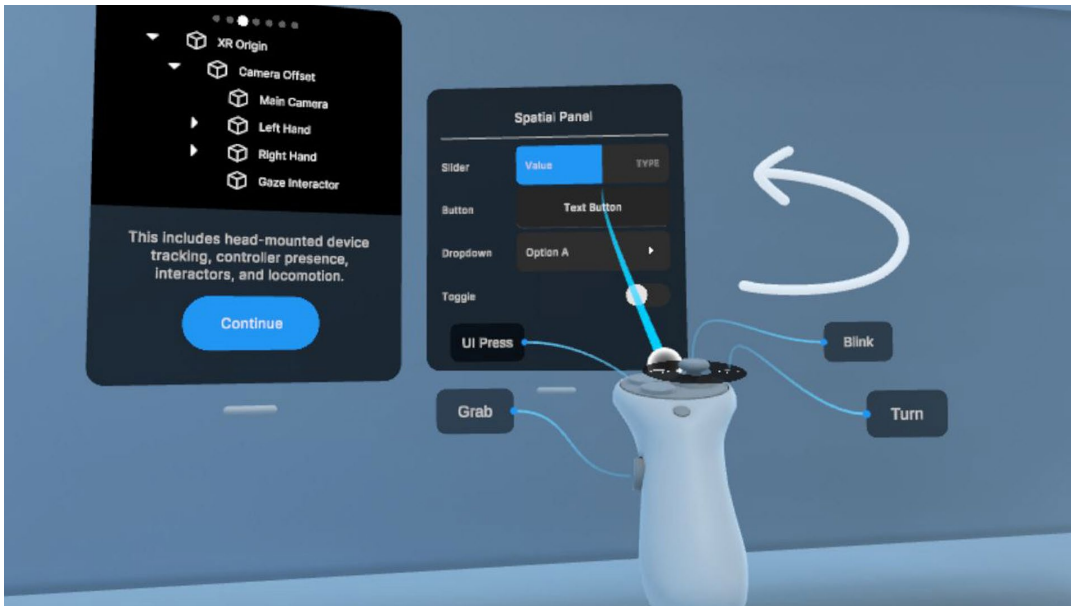
VR 샘플 씬 테스트

에디터에서 플레이 모드로 진입하여 씬의 기능을 테스트합니다. 컨트롤러에는 씬의 항목을 탐색하고 조작하는데 도움이 되는 UI 툴팁이 포함되어 있습니다. 헤드셋이 없는 경우, 이 전자책 후반부의 [XR Device Simulator](#) 섹션에서 헤드셋 없이 시뮬레이터를 사용해 테스트하는 방법을 참고하세요.



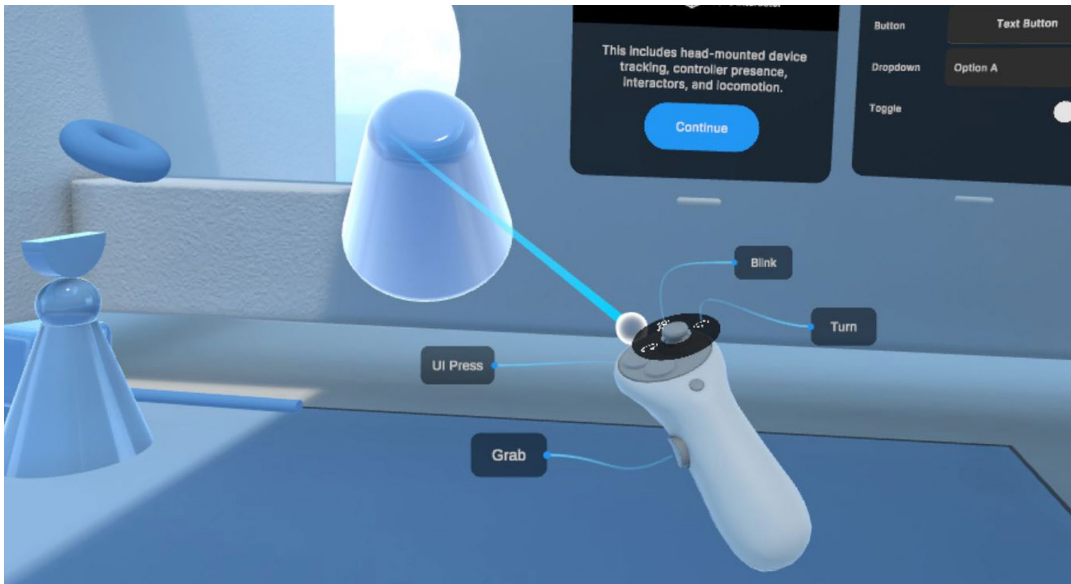
툴팁이 표시되어 있고 트래킹 중인 컨트롤러

VR 컨트롤러를 사용하여 다양한 UI 요소와 상호 작용합니다.



VR에서 UI와의 상호 작용

씬에 있는 다양한 오브젝트를 집어 들고 조작해 보세요.

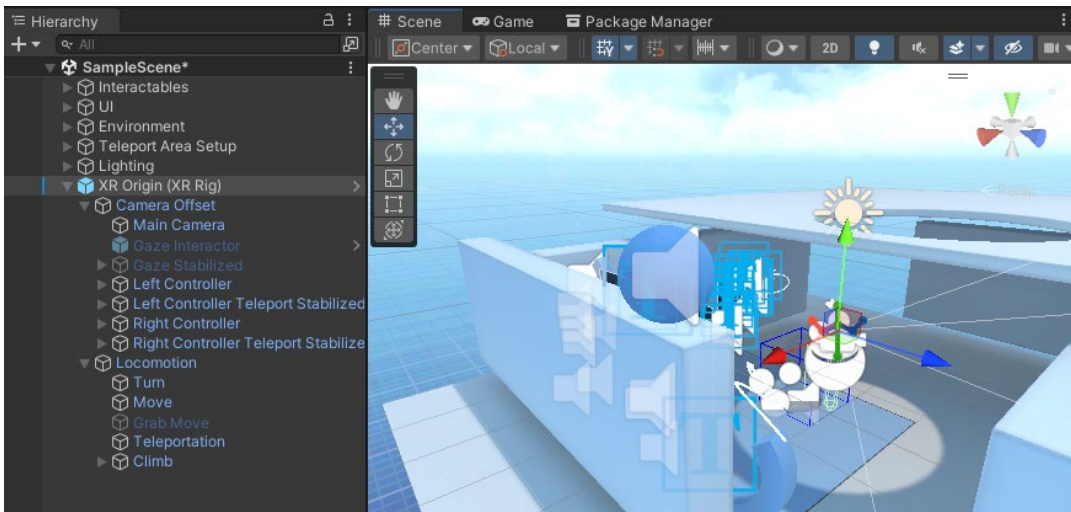


원거리 잡기를 사용하여 오브젝트 잡기

이 씬에는 게임 개발을 시작하는 데 도움이 되는 다양한 기능이 포함되어 있습니다.

샘플 씬 설정

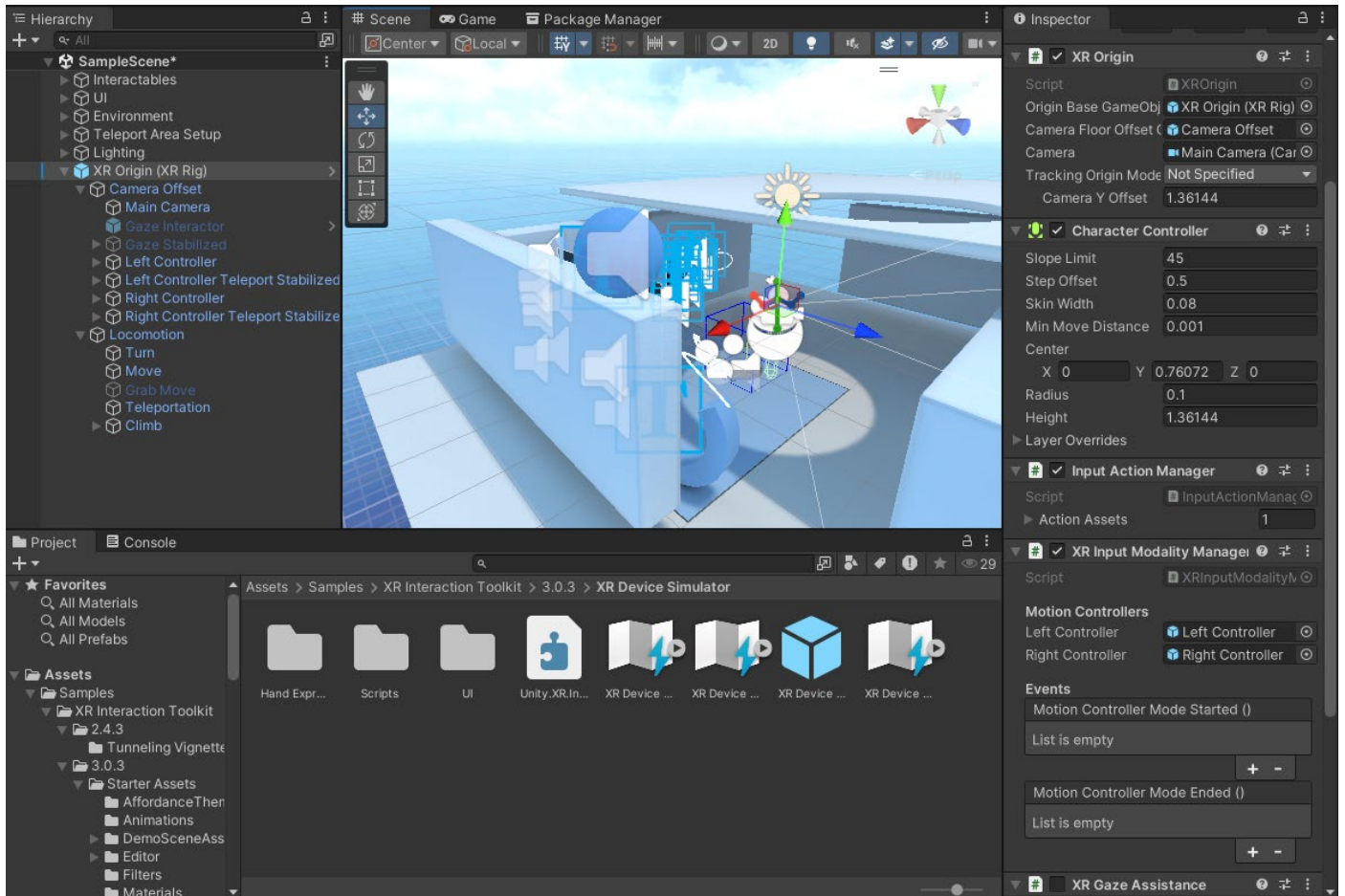
샘플 씬의 핵심 컴포넌트 몇 가지를 살펴보겠습니다.



XRI 2.5.x의 샘플 씬에 있는 계층 구조의 Complete XR Origin Set up Variant 프리팹 배리언트

계층 구조에는 2.5.x 버전의 **Complete XR Origin Set up Variant**라는 **프리팹 배리언트**가 있고, 여기에는 VR 월드를 탐색하고 상호 작용하는 데 필요한 모든 컴포넌트가 있습니다. XRI 3.0.3으로 업데이트한 후에는 해당 프리팹을 제거하고 3.0.3의 Starter Assets에서 **XR Origin (XR rig)** 프리팹을 추가해야 합니다.

여기에는 잡기, UI 컴포넌트와의 상호 작용, 이동 기법(순간 이동, 빠른 방향 전환, 컨트롤러의 썸스틱을 사용한 움직임 등)을 사용하는 내비게이션을 비롯하여 사용자 상호 작용의 원활한 작동에 필요한 모든 설정이 포함되어 있습니다.



XRI 3.0.3의 샘플 씬 계층 구조에 XR Origin (XR Rig) 프리팹 배리언트를 수동으로 추가해야 합니다.

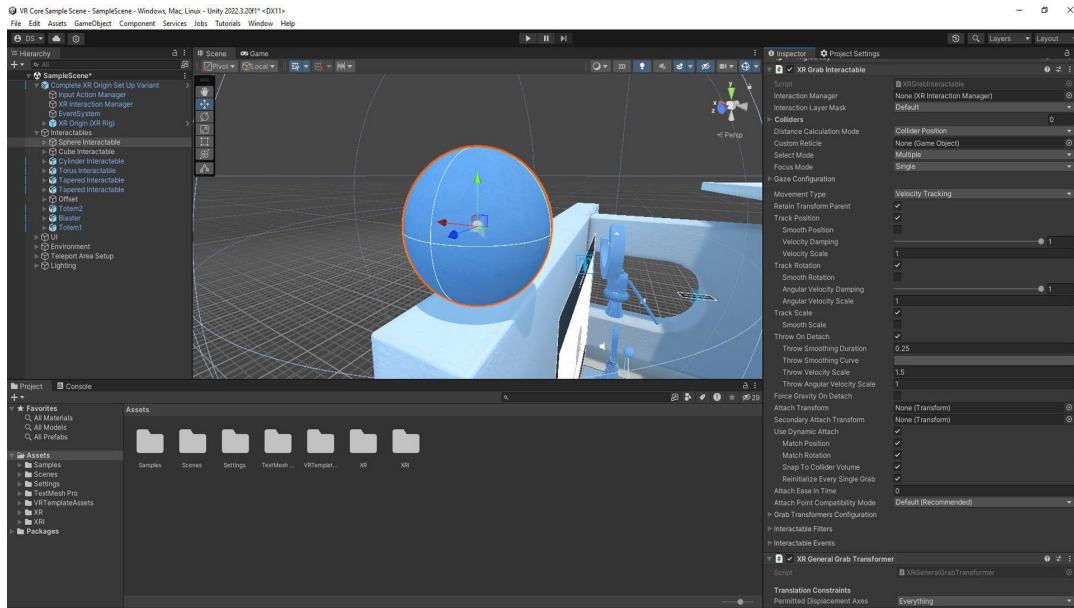
또한 프리팹에는 다음과 같은 모든 핵심 요소가 있습니다.

- **Input Action Manager:** 이 클래스는 XR 환경 내에서 발생하는 입력 동작의 활성화와 비활성화 관리를 담당합니다.
- **XR Interaction Manager:** XR 컨트롤러, 인터랙터블 오브젝트, 환경 간의 상호 작용 관리를 담당합니다.
- **XR Origin:** XR 세션의 공간 원점을 관리하며 물리적 움직임과 방향을 가상 월드로 변환하는 데 핵심적인 역할을 합니다. 카메라, 좌우 컨트롤러 트래킹, 이동 설정이 포함됩니다.

이러한 컴포넌트에 대한 자세한 내용은 다음 섹션에서 설명합니다.

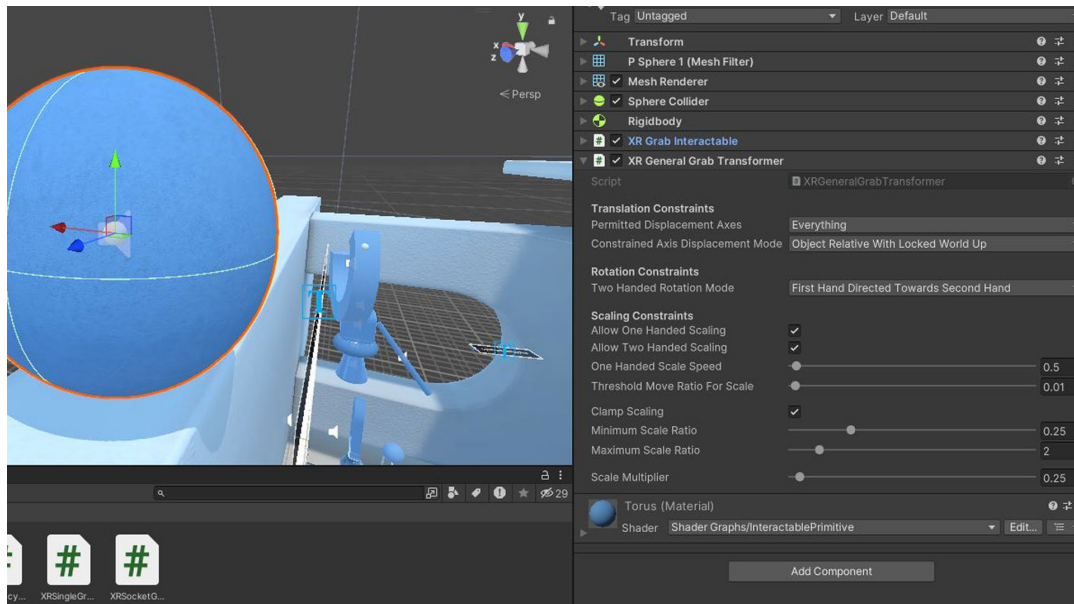
이 계층 구조에는 다양한 방식으로 상호 작용할 수 있는 모든 게임 오브젝트가 포함된 **Interactables**라는 부모 게임 오브젝트도 있습니다.

예를 들어 구체와 같이 잡을 수 있는 일반 오브젝트는 XR Interaction Toolkit의 컴포넌트인 XR Grab Interactable을 통해 구현할 수 있습니다.



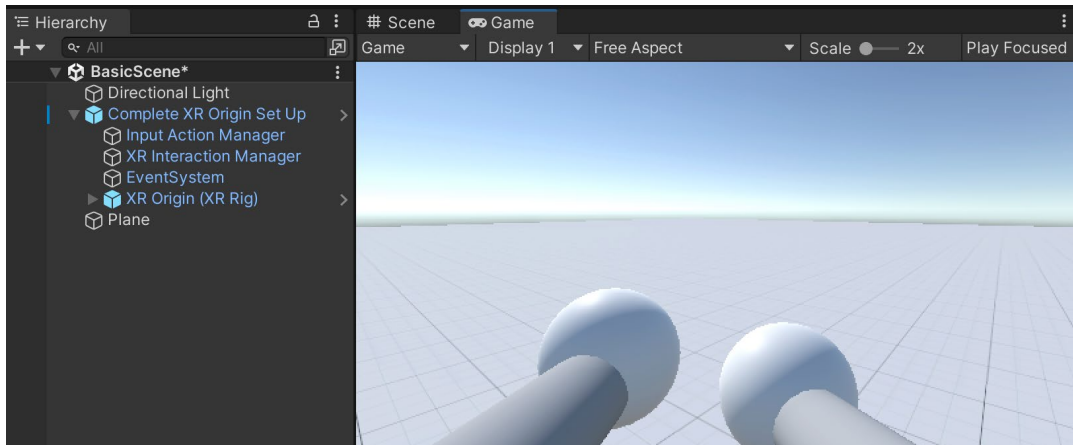
XR Grab Interactable 컴포넌트가 적용되어 잡을 수 있는 오브젝트

VR에서 인터랙터블 요소를 두 손으로 잡아 그 크기를 늘리거나 줄일 수도 있으며, 이 동작은 XR General Grab Transformer 스크립트를 통해 가능합니다.



XR General Grab Transformer를 사용하면 오브젝트를 조작할 수 있습니다.

프로젝트에는 BasicScene이라는 씬도 포함되어 있으며, 여기에는 환경과 상호 작용을 추가할 수 있는 설정 프리팹이 있습니다.



BasicScene에 이미 컨트롤이 설정되어 있어 환경을 추가할 준비가 완료된 상태입니다. XRI 3.0.3을 사용하는 경우 Complete XR Origin Set Up을 삭제하고 XR Origin (XR Rig)을 추가합니다.

다음 튜토리얼을 시청하며 샘플을 설정하는 과정을 따라해 보세요.

[튜토리얼 보기](#)

프로젝트 템플릿

Github에서 다양한 기능을 사용하는 방법에 대한 더 많은 예시를 다운로드하여 살펴볼 수 있습니다.

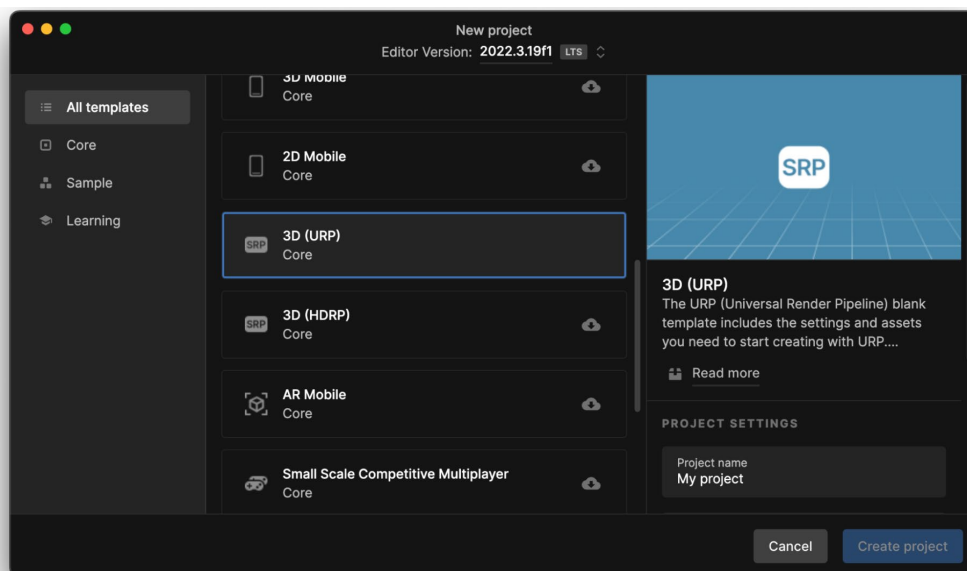
- [XR Interaction Toolkit 예시](#)
- [Meta-OpenXR용 MR\(혼합 현실\) 예시](#)
- [XR 시뮬레이션 환경](#)

URP를 사용하여 새로운 VR 프로젝트 생성

VR 코어 템플릿 설정 단계를 수행했다면, 이제 비어 있는 새 VR 프로젝트를 시작하고 주요 컴포넌트를 자세히 살펴보세요.

3D URP 템플릿으로 시작

새 프로젝트를 시작하려면 Unity Hub를 열고 **새 프로젝트**를 선택한 후 최신 버전의 **Unity 2022 LTS**를 선택합니다. 템플릿 목록에서 아래로 스크롤하여 **3D (URP)**를 다운로드합니다.



Unity Hub에서 3D URP 템플릿 선택

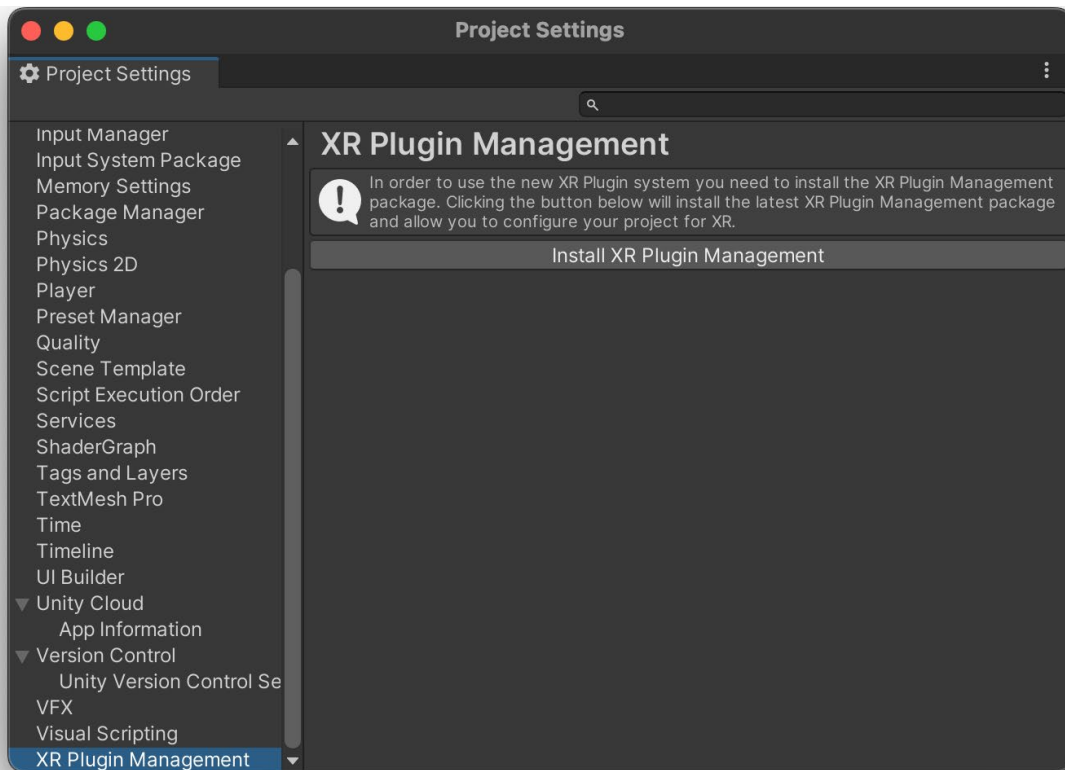
프로젝트 설정에 내용을 작성하고 프로젝트 생성을 클릭합니다.

에디터에서 URP 기술 자료, 포럼 및 버그 신고 링크와 함께 인스펙터(Inspector) 창이 열립니다.

XR Plug-in Management

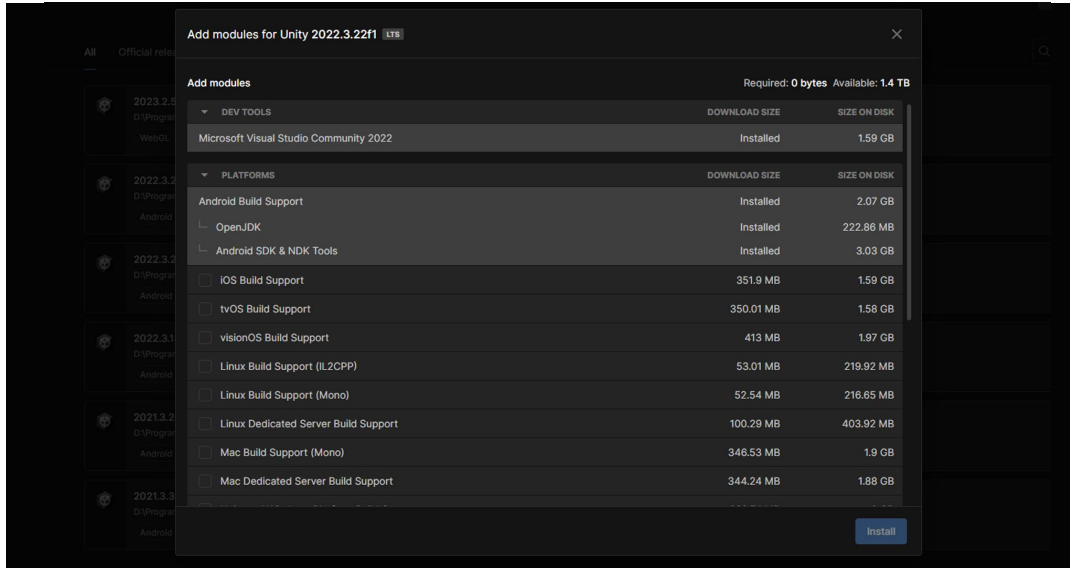
Project Settings 창에서 사용할 수 있는 XR Plug-in Management 패키지는 XR 플러그인의 통합과 설정을 관리합니다. 여기에는 VR, AR, MR 기기에 대한 설정이 포함됩니다. 프로젝트에서 다양한 하드웨어를 지원하는 데 사용할 적절한 XR SDK와 플러그인을 선택하고 설정할 수 있습니다. 이는 기본적으로 Unity 프로젝트가 다양한 XR 기술과 상호 작용하는 방식을 설정하고 커스터마이징할 수 있는 중앙 집중형 허브 역할을 수행합니다.

Project Settings 창으로 이동하여 목록에서 **XR Plug-in Management** 패키지를 선택하고 해당 바를 클릭하여 설치합니다.



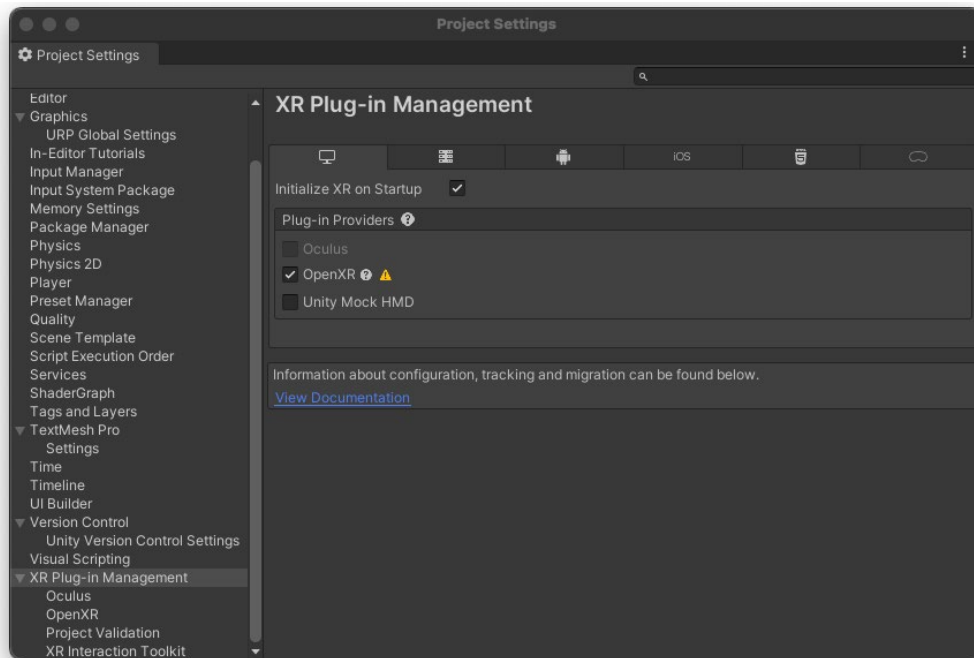
XR Plug-in Management 패키지 설치

이제 빌드 플랫폼과 플러그인 제공업체를 선택할 수 있습니다. 필요한 빌드 모듈이 설치되어 있는지 확인하세요. 해당 플랫폼이 목록에 표시되지 않는 경우 Unity Hub로 이동하여 설치 항목을 찾습니다. 여기에서 톱니바퀴 아이콘을 클릭한 후 **모듈 추가**를 클릭하여 모듈을 추가할 수 있습니다.



Hub에서 빌드 모듈 추가

에디터로 돌아가 보면 목록에서 플랫폼을 볼 수 있습니다. 각 플랫폼의 **플러그인 제공업체**를 선택합니다. 이 예시에서는 **OpenXR**을 선택합니다.



OpenXR을 빌드 타겟으로 선택

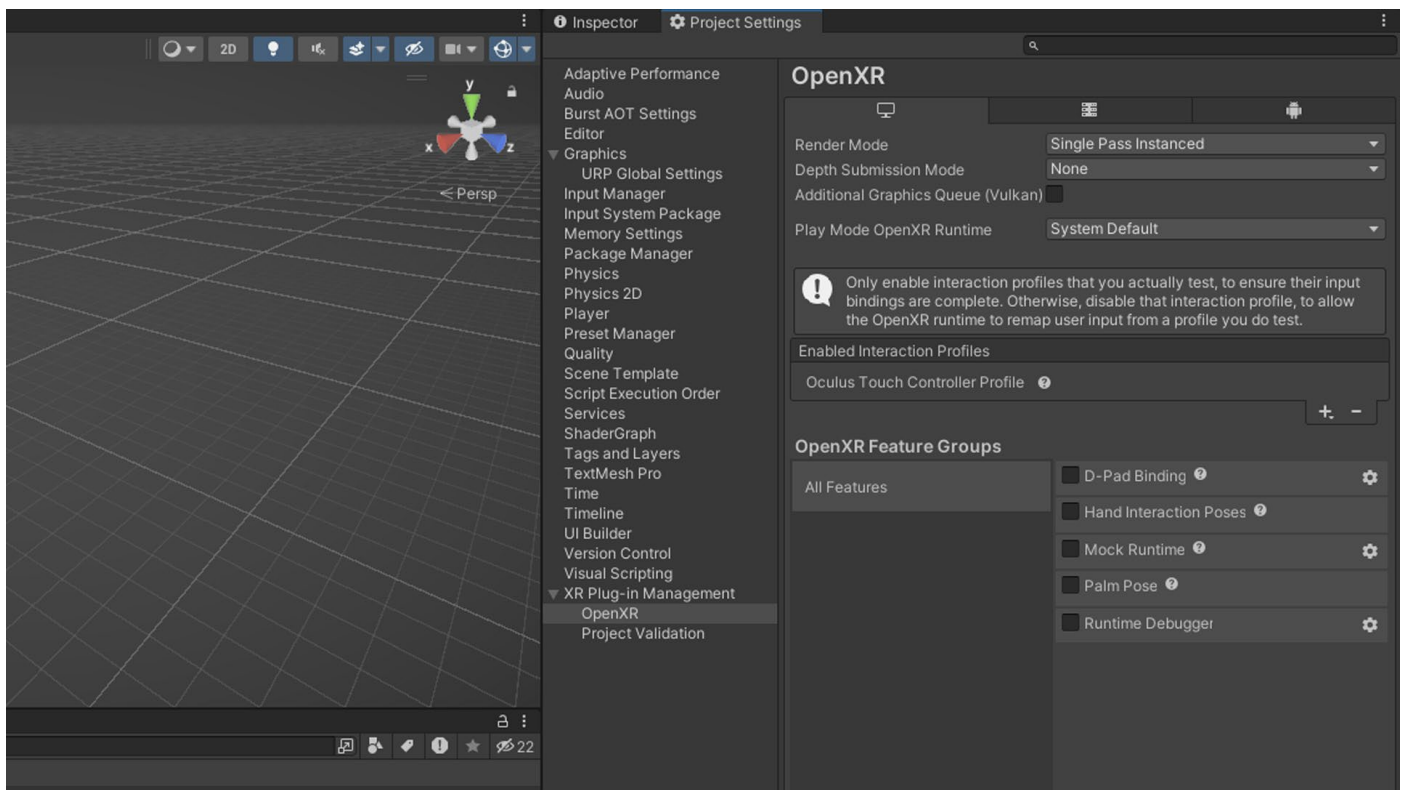
참고: 이 입력 시스템을 활성화하려는 경우 에디터를 다시 시작하라는 메시지가 표시될 수 있으며, 이때 에디터를 다시 시작한 후 OpenXR 옵션을 선택할 수 있습니다(사전에 선택되어 있지 않은 경우).

상호 작용 프로파일 추가

XR Plug-in Management 패키지의 상호 작용 프로파일을 사용하여 프리셋을 설정하면 타겟 기기와 상호 작용하는 방식을 게임에서 인식할 수 있습니다. 예를 들어 게임이 Meta Quest를 지원하는 경우 Oculus Touch 컨트롤러 프로파일을 선택하면 됩니다. 타게팅하는 기기가 여러 종류라면 다수의 프로파일을 할당할 수 있습니다.

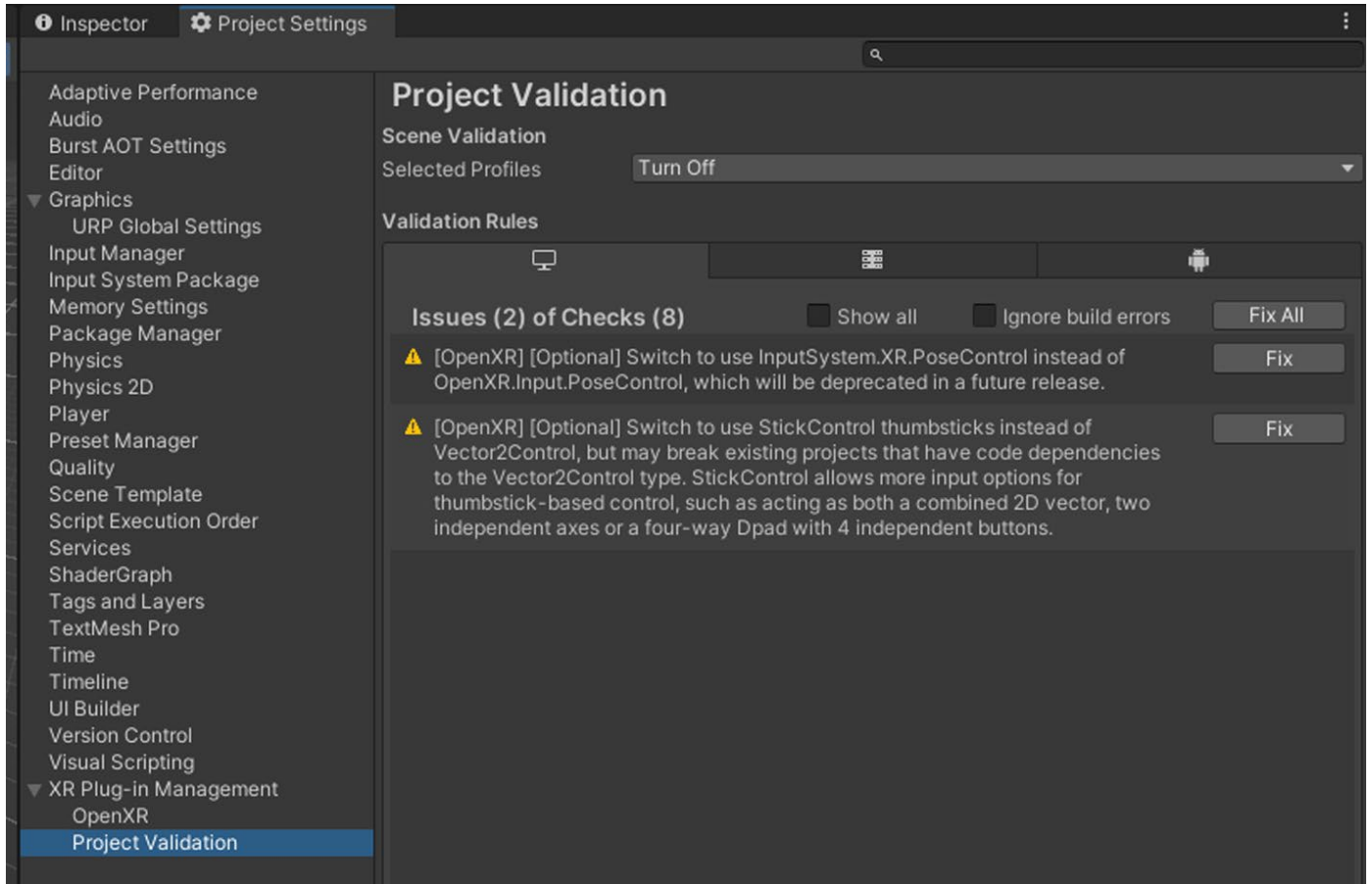
XR Plug-in Management 패키지가 설치되고 OpenXR로 설정되면 애플리케이션 실행 시 해당 컨트롤러를 트래킹할 수 있도록 상호 작용 프로파일을 OpenXR로 할당할 수 있습니다. 프로파일 설정은 타겟 하드웨어에 따라 다릅니다.

Enabled Interaction Profiles 섹션에서 + 아이콘을 클릭하면 목록에서 컨트롤러 유형을 선택할 수 있습니다.



상호 작용 프로파일 선택

Project Validation 탭에서 경고가 없는지 확인합니다. 왼쪽 열에서 **Project Validation**을 선택하여 이 작업을 빠르게 수행할 수 있습니다. 문제가 있다면 여기에 나열되며, **Fix All**을 클릭해 해결할 수 있습니다.



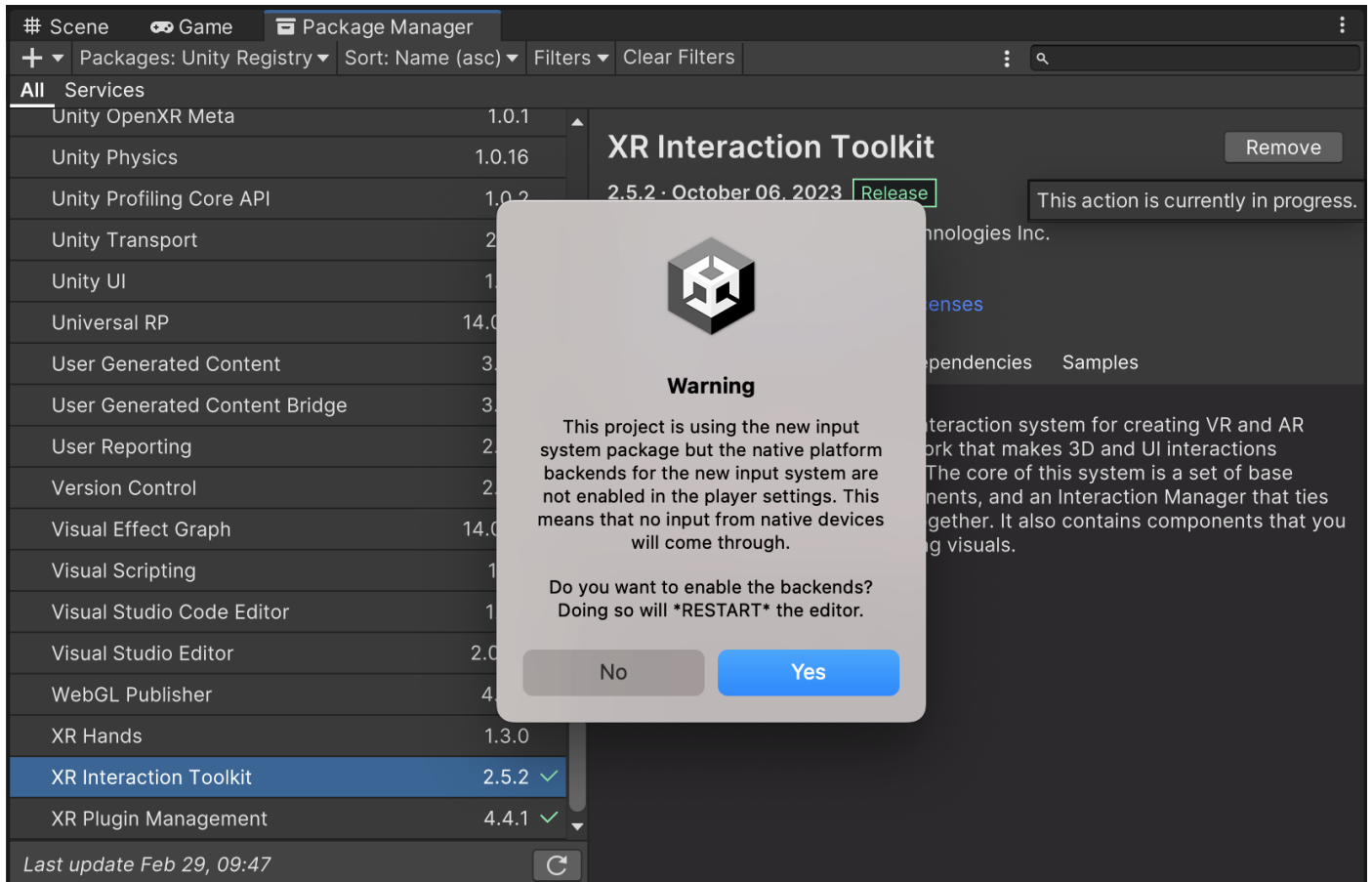
Project Validation

다음 단계는 XR Interaction Toolkit(XRI) 패키지를 설치하는 것입니다.

XRI Toolkit 구현

이 가이드의 전반부에서는 [XRI Toolkit](#)에 대한 소개 내용을 살펴보았고, 이제는 이를 구현하는 방법을 살펴보겠습니다.

Window > Package Manager를 통해 Unity Registry 드롭다운에서 XR Interaction Toolkit을 선택하고 Install을 클릭합니다. 입력 시스템 백엔드를 활성화할지 묻는 메시지가 표시되는 경우 **Yes**를 클릭하면 에디터가 다시 시작됩니다.

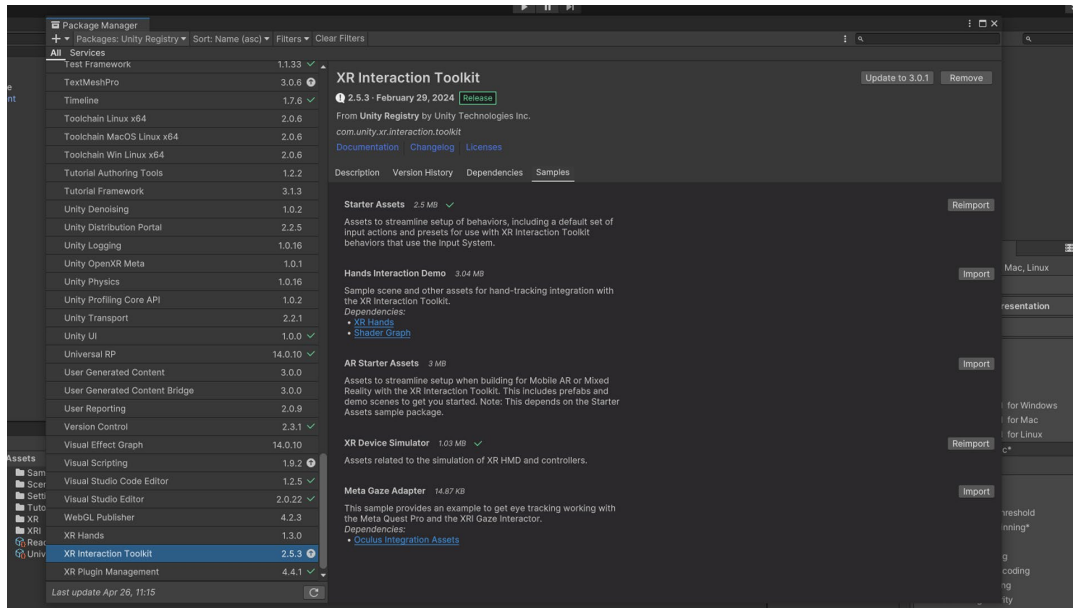


XRI Toolkit 패키지 설치

Starter Assets를 사용하여 XRI 살펴보기

다음 섹션에 나와 있는 단계는 패키지에 포함된 Starter Assets를 사용하는 경우를 기준으로 합니다. 이는 동작 설정, 입력 동작, XR Interaction Toolkit의 기능을 보여 주는 데모 씬을 간소화하는 효율적인 방법이 될 수 있습니다.

이 접근 방식에 관심이 있는 경우 **Samples**를 클릭하고 Starter Assets 샘플을 프로젝트로 임포트합니다.

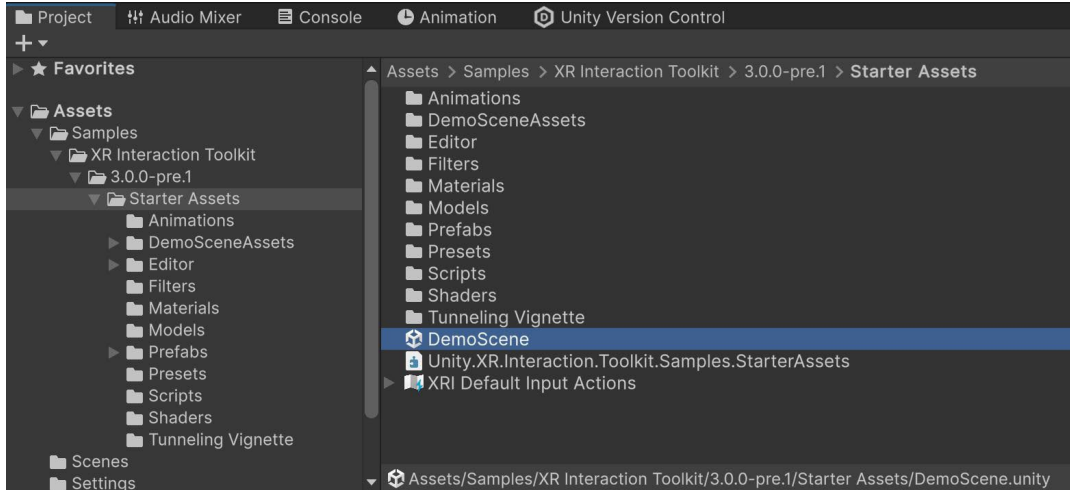


XR Interaction Toolkit Starter Assets

데모 씬

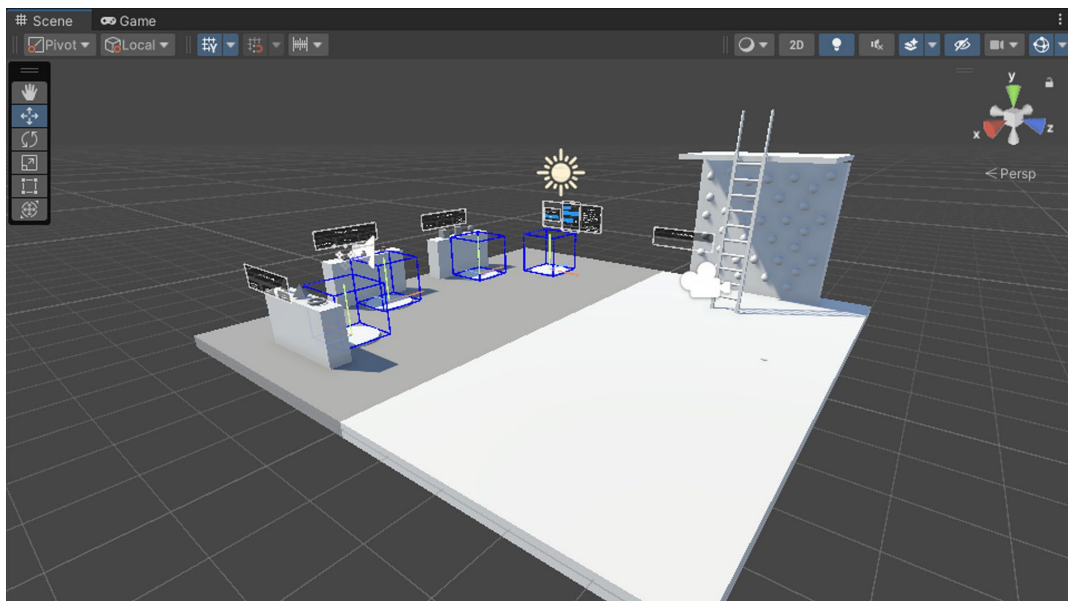
이 섹션에서는 XRI Toolkit 데모 씬에 포함된 각 프리팹과 컴포넌트의 역할을 살펴보겠습니다. 데모 씬을 찾을 수 없는 경우 이전 섹션의 설명대로 샘플을 설치했는지 다시 확인합니다.

Starter Assets 샘플에는 데모 씬이 포함되어 있습니다(경로: **Samples** 폴더 > **XR Interaction Toolkit** > (툴킷 버전) > **Starter Assets** > **DemoScene**).



이 데모 씬에는 원하는 VR 게임을 제작하는 데 필요한 모든 것이 포함되어 있습니다.

- 내비게이션
- 잡기 상호 작용과 다양한 움직임 유형
- 누르기 상호 작용
- 응시 상호 작용
- 오르기 상호 작용
- UI 상호 작용
- 오르기

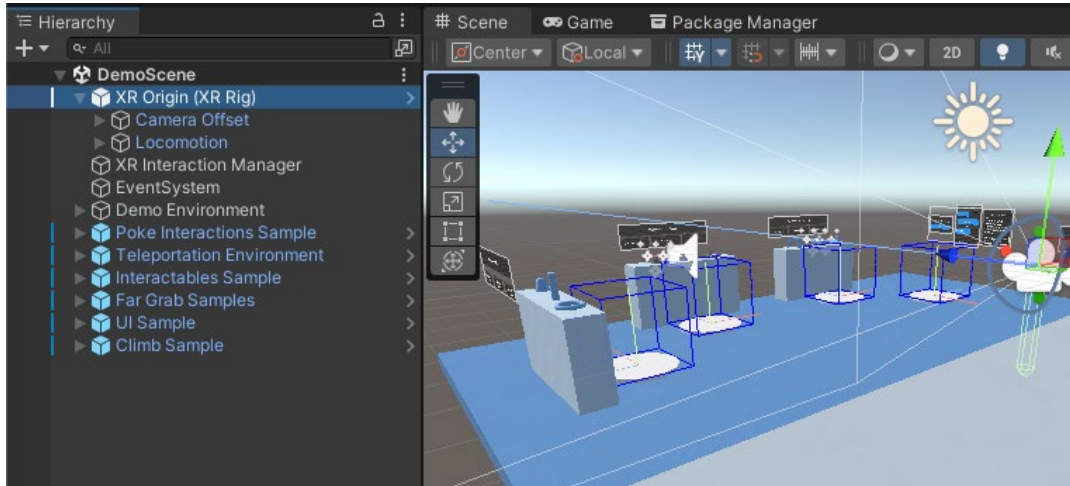


XR Interaction Toolkit 데모 씬



Starter Assets 데모 씬에서 XR 상호 작용 설정

XRI 3.0의 데모 씬에는 **Input Action Manager** 컴포넌트와 **XR Interaction Manager** 게임 오브젝트가 포함된 **XR Origin (XR Rig)** 프리팹이 있습니다.



XR Origin (XR Rig) 프리팹

Input Action Manager

Input Action Manager는 XR 환경 내에서 발생하는 입력 동작의 활성화와 비활성화를 관리합니다.

InputActionAsset 오브젝트 목록과 함께 작동하며, 여러 에셋 전반의 입력 동작을 중앙에서 제어할 수 있도록 합니다. 필요할 때는 동작에 반응하고 그렇지 않을 때는 휴면 상태가 유지되므로, 사용자 상호 작용과 경험에 동적 입력 처리가 중요한 VR 및 AR 시나리오에서 특히 유용합니다.

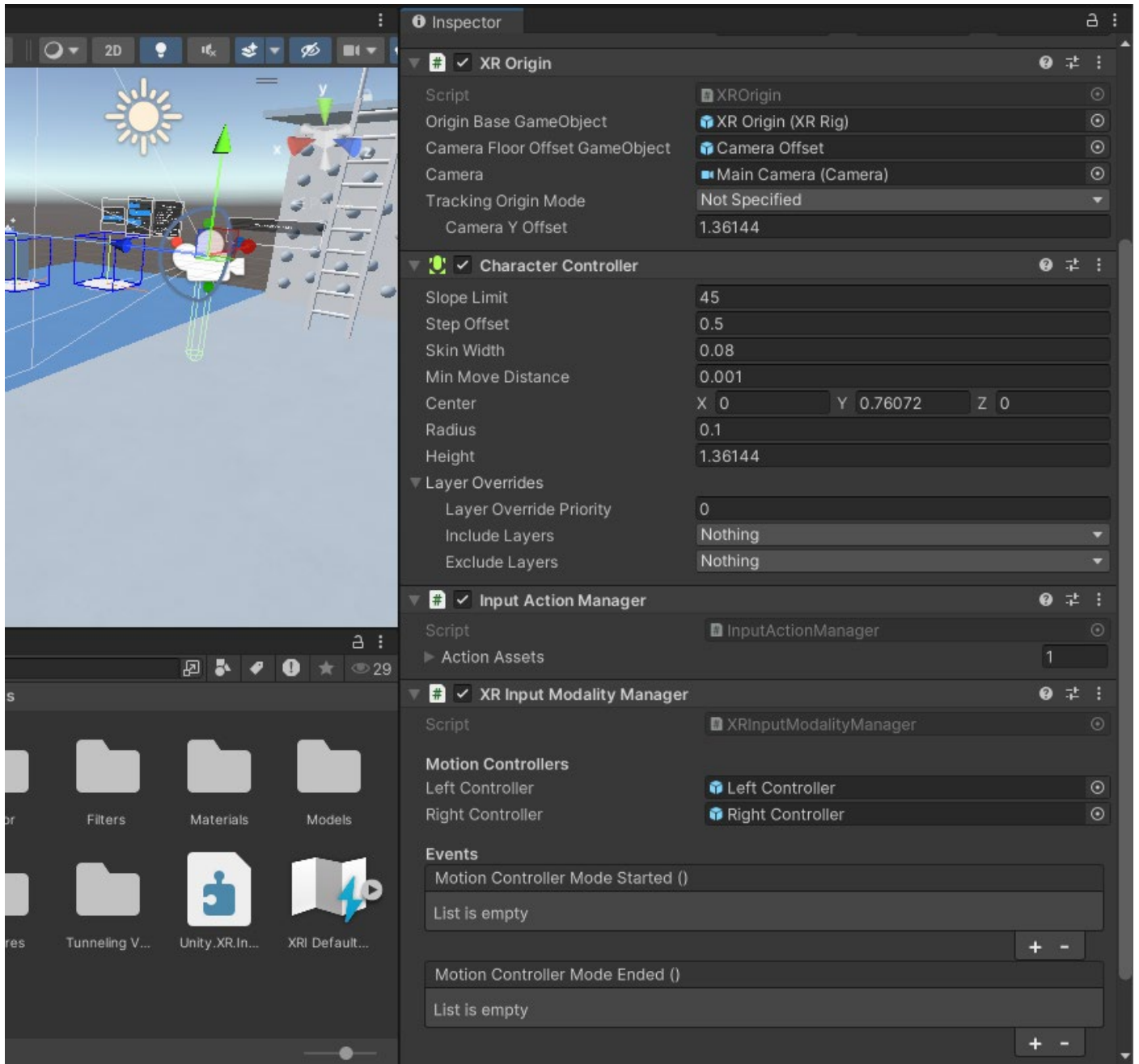
XR Interaction Manager

이 **관리자**는 인터랙터(예: VR 핸드 컨트롤러, 포인터)와 인터랙터블(VR/AR 공간에서 플레이어가 잡을 수 있는 버튼이나 프랍처럼 상호 작용 가능한 오브젝트) 간의 상호 작용을 관리하는 중앙 허브 역할을 하며, 인터랙터와 인터랙터블이 활성화되면 이들이 씬에서 제대로 커뮤니케이션 및 작동할 수 있도록 Interaction Manager를 통해 스스로를 등록하도록 합니다. 자세한 내용은 XR Interaction Toolkit [용어집](#)을 참조하세요.

XR Origin

XR Origin 프리팹은 XR 세션의 공간 원점을 관리합니다. ‘바닥’ 또는 ‘기기’ 높이와 같은 다양한 트래킹 모드를 지원하여 XR 기기와 연관된 카메라 위치를 처리합니다. 이 프리팹은 물리적 움직임과 방향을 가상 월드로 변환하는 데 중요한 역할을 합니다. 이는 가상 공간 내에서 정확하고 반응이 빠른 상호 작용을 보장하도록 Unity의 XR 환경을 설정하는 데 필수적입니다.

XR Origin 프리팹에는 **Character Controller**와 **XR Input Modality Manager**가 포함되어 있습니다.



XR Origin 컴포넌트

Character Controller Driver

Character Controller Driver는 가상 캐릭터의 키와 위치를 현실 세계에서 플레이어가 착용한 VR 헤드셋의 움직임에 따라 조정하도록 설계되었습니다. 플레이어가 VR 공간에서 움직이기 시작하고 멈추는 시점을 수신하며, 움직임과 충돌을 처리하는 [Character Controller 컴포넌트](#)를 업데이트합니다.

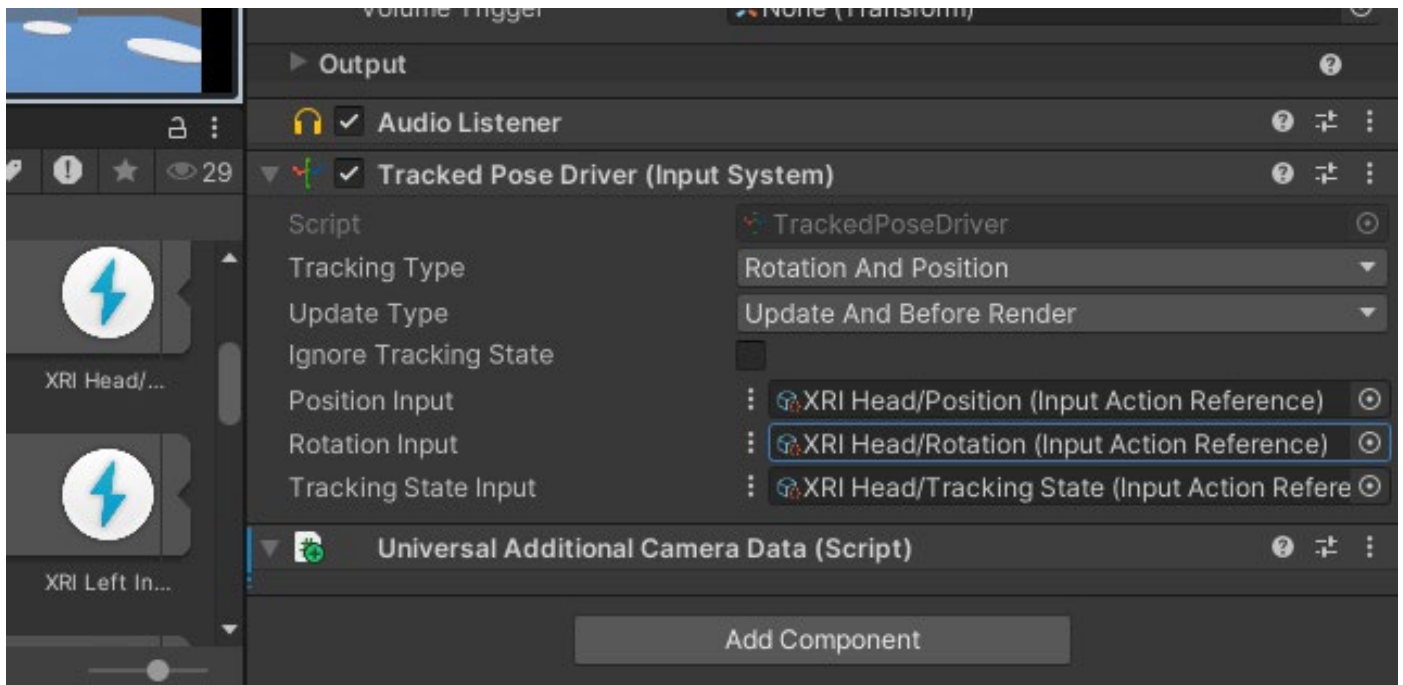
XR Input Modality Manager

XR Input Modality Manager는 런타임 중에 트래킹 상태에 따라 손 트래킹과 모션 컨트롤러 간의 동적 전환을 관리합니다. 손과 컨트롤러를 나타내는 다양한 게임 오브젝트 세트를 전환하여 적절한 입력 방법을 활성화합니다. 손 트래킹이 시작되면 인터랙터의 손 그룹으로 전환되고, 모션 컨트롤러가 사용 및 트래킹되면 모션 컨트롤러 그룹이 활성화됩니다. 이 컴포넌트는 컨트롤러가 트래킹되지 않을 때 기본 위치에 표시되는 것을 방지하므로 손 트래킹을 사용하지 않는 프로젝트에서도 유용합니다.

XR Origin 프리팹에는 2개의 자식 게임 오브젝트, 즉 메인 카메라와 컨트롤러 트래킹 기능이 있는 **Camera Offset** 및 **Locomotion System**이 있습니다.

TrackedPoseDriver (Input System)

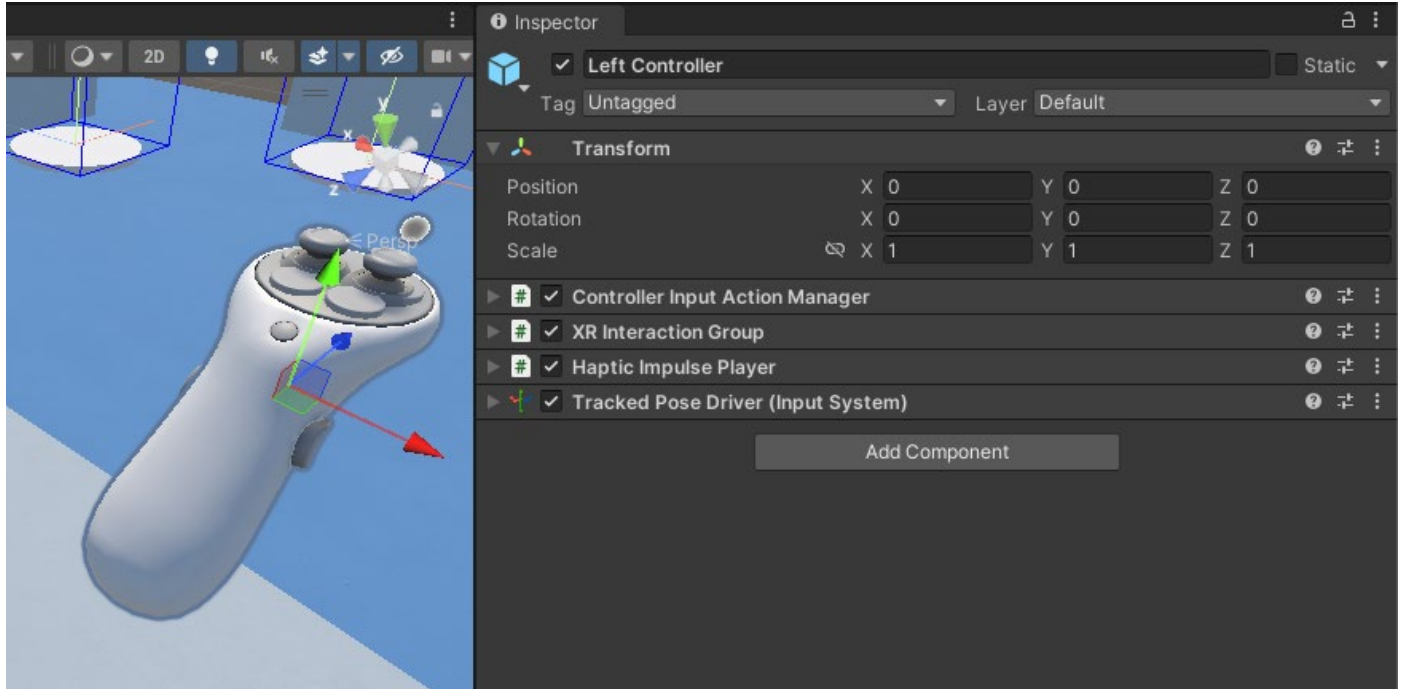
메인 카메라에는 **TrackedPoseDriver (Input System)** 컴포넌트가 있습니다. 사용자의 머리를 실시간으로 트래킹하는 것처럼 현실 세계의 물리적 움직임을 가상의 움직임에 매핑하여 해당 모션을 VR로 변환하는 데 중요한 역할을 합니다.



메인 카메라의 Tracked Pose Driver (Input System) 컴포넌트

입력은 XRI Toolkit에서 제공하는 입력 동작에 매핑되어 바로 사용할 수 있습니다.

좌우 컨트롤러에 대한 설정은 Camera Offset에도 포함되어 있습니다. 이 두 오브젝트에는 모두 **Controller Input Action Manager** 및 **XR Interaction Group** 컴포넌트가 동일하게 있습니다.



컨트롤러 관련 컴포넌트

Controller Input Action Manager

Controller Input Action Manager는 좌우 컨트롤러 모두에 적용되는 컴포넌트입니다. 이를 통해 VR 애플리케이션에서 VR 컨트롤러가 오브젝트와 상호 작용하고 사용자 입력에 반응하는 방식을 관리할 수 있습니다. 광선으로 멀리 있는 물체를 가리키거나 순간 이동하는 등의 다양한 이동 및 상호 작용 방식을 전환할 수 있습니다.

플레이어의 진행 상황에 따라 특정 시점에는 상황에 맞는 행동(예: 이동 또는 순간 이동)만 가능하도록 할 수 있습니다.

XR Interaction Group 컴포넌트

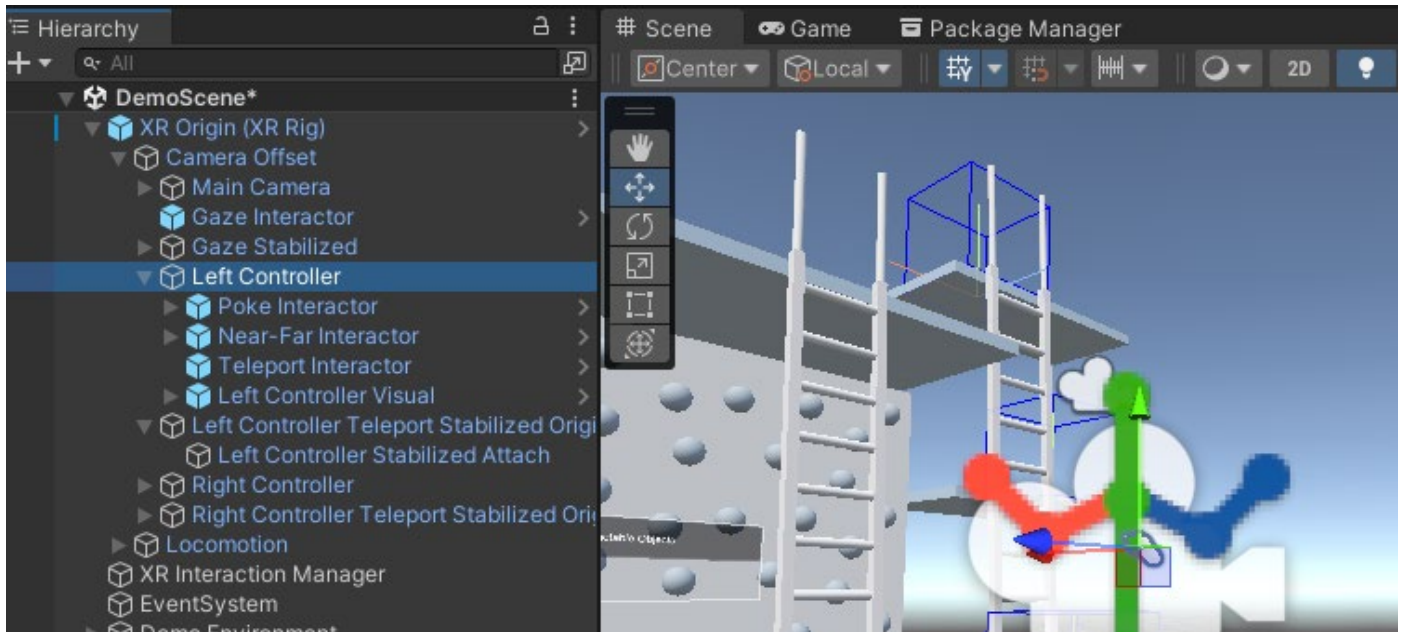
XR Interaction Group 컴포넌트는 인터랙티브 요소 그룹을 관리하는 시스템으로, 그룹 내에서 한 번에 하나의 인터랙티브 요소 또는 인터랙터만 활성화될 수 있게 합니다.

XR Transform Stabilizer

XR Origin 프리팹의 마지막 오브젝트 세트는 **Left/Right Controller Stabilized Origin** 게임 오브젝트입니다. 두 오브젝트에는 모두 **XRTransformStabilizer** 스크립트가 연결되며, 이 스크립트는 XR 환경이 더 정확하게 느껴지도록 하고 의도하지 않은 작은 움직임에 덜 영향을 받도록 합니다. Left Controller와 Right Controller의 트랜스폼을 타겟 오브젝트로 할당하면 움직임이 부드러워져 트래킹 데이터의 흔들리거나 불규칙한 움직임이 안정화됩니다.

상호 작용 설정

순간 이동, 잡기, 누르기, 상호 작용을 제어하는 시스템은 모두 Left Controller와 Right Controller에 포함되어 있습니다.



Left Controller와 Right Controller에 포함된 상호 작용 설정

이러한 인터랙터에는 각각 해당하는 컴포넌트가 연결됩니다. 예를 들어 플레이어가 오브젝트를 집어 들 수 있는 **Poke Interactor** 게임 오브젝트에는 **XR Poke Interactor**라는 컴포넌트가 있습니다.

XR Direct Interactor

XR Direct Interactor 컴포넌트를 사용하면 버튼, 레버, 그 외 다양한 오브젝트와 프랍 등, 플레이어가 만지거나 움직이거나 잡을 수 있는 인터랙터블 요소와 직접 상호 작용할 수 있습니다.

Poke, Ray 및 Teleport Interactor는 모두 VR 월드 내에서 특정 상호 작용에 사용할 수 있습니다. 이 전자책에서 다양한 인터랙터블 요소를 살펴보세요.

이동 시스템

가상 공간의 기준점(일반적으로 플레이어 또는 사용자가 위치한 곳)을 나타내는 XR Origin 프리팹에 대한 액세스 권한을 [관리 및 제어](#)합니다. 한 번에 하나의 이동 제공자(Locomotion Provider)만 XR Origin을 옮길 수 있도록 하여 충돌을 방지하고 XR 애플리케이션에서 매끄럽고 예측 가능한 이동 경험을 보장할 수 있습니다.

플레이어의 위치를 조작하는 [방법](#)은 회전, 이동, 순간 이동, 오르기입니다.



방향 전환

XRI Toolkit에서는 플레이어의 회전을 위한 두 가지 컴포넌트, 즉 **Snap Turn** 및 **Continuous Turn**을 제공합니다. Snap Turn을 사용하면 플레이어가 일정한 각도만큼 회전할 수 있으며, 이 각도는 **Snap Turn Provider** 컴포넌트를 통해 커스터마이징할 수 있습니다. Continuous Turn을 사용하면 **Continuous Turn Provider** 컴포넌트에서 설정한 가변 속도로 부드럽게 회전합니다. 두 컴포넌트 모두 Unity의 입력 시스템과 통합되어 간편하게 사용할 수 있습니다. 툴킷에는 사전 설정된 입력 동작이 포함되어 있으므로 개발을 간소화하고 이러한 방향 전환 컴포넌트를 최소한의 설정으로 바로 사용할 수 있습니다.

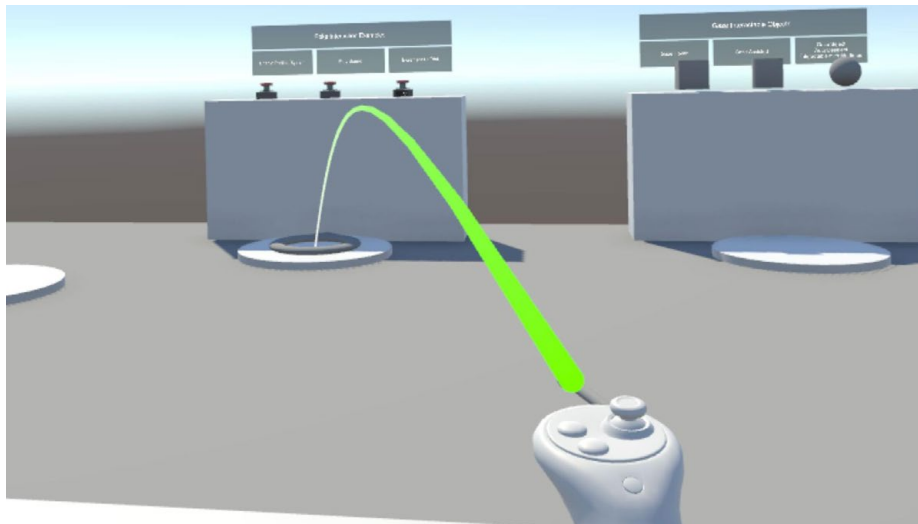
이동

DynamicMoveProvider는 연속적인 이동을 제어하는 배리언트로, 사용자 환경 설정에 따라 이동 방향의 기준 프레임을 조정하도록 설계되었습니다. 사용자의 양손 설정에 따라 머리나 손을 기준으로 전진 방향을 제어할 수 있습니다. 이를 통해 XR 환경에서 커스터마이징 가능하며 직관적인 이동 경험이 제공됩니다.

순간 이동

XRI Toolkit의 **Teleportation**을 사용하면 플레이어가 가상 환경 내 새로운 위치로 즉시 이동할 수 있습니다. **Teleportation Anchor**라고 하는 타겟 위치를 목표로 잡고 순간 이동 동작을 시작하면 됩니다. 순간 이동 시스템에는 순간 이동의 경로와 목표를 보여 주는 원호나 선과 같은 시각적 표시기가 있습니다. 넓은 가상 공간을 쾌적하게 탐색하는 데 필요한 핵심 기능으로, 특히 연속적으로 이동하기가 불편하거나 공간의 제약으로 불가능할 수 있는 시나리오에서 꼭 필요합니다.

Teleportation Provider 컴포넌트는 XR 환경에서 순간 이동을 처리합니다. 이는 사용자의 요청에 따라 XR Origin을 지정된 위치로 옮기는 역할을 합니다. 주요 기능으로는 순간 이동의 지연 시간 설정이 있으며, 터널링 비네트(vignette) 효과와 같은 멀미 방지용 설정에 사용할 수 있습니다. 또한 월드 공간 위쪽, 타겟 위쪽, 타겟 위쪽과 전방 등, 목표에 대한 다양한 유형의 방향 매칭을 지원합니다. 제공자는 순간 이동 요청을 대기열에 추가하고 방향 및 위치 변환을 비롯한 요청의 적용을 관리하여 원하는 순간 이동 효과를 얻을 수 있습니다.



앵커 포인트를 사용한 순간 이동



오르기

ClimbProvider 컴포넌트는 XR 환경의 오르기 메카닉을 지원합니다. 이 컴포넌트로 사용자가 **ClimbInteractable** 오브젝트와 상호 작용하도록 지원하며, 사용자의 동작에 반응하여 XR Origin을 움직이도록 할 수 있습니다. 오르기 모션은 사용자 인터랙터의 움직임에 따라 상쇄되며 인터랙터블 요소의 각 축에 따라 제한될 수 있습니다.

터널링 비네트

샘플 씬에는 Main Camera의 자식인 **TunnellingVignette** 프리팹도 있습니다. 씬을 탐색하며 움직이다 보면 검은색 테두리가 페이드인/아웃되는 것을 볼 수 있습니다. VR 비네팅 또는 동적 시야각(FOV) 감소라고 하는 이 효과로 멀미를 줄일 수 있습니다. 주변 시야 범위를 줄이는 식으로 작동하며 사용자의 주의를 화면 중앙으로 집중시킵니다. 주변의 모션을 줄이면 시야와 전정 입력 간 불일치가 줄어들어 멀미가 발생할 확률이 감소합니다.



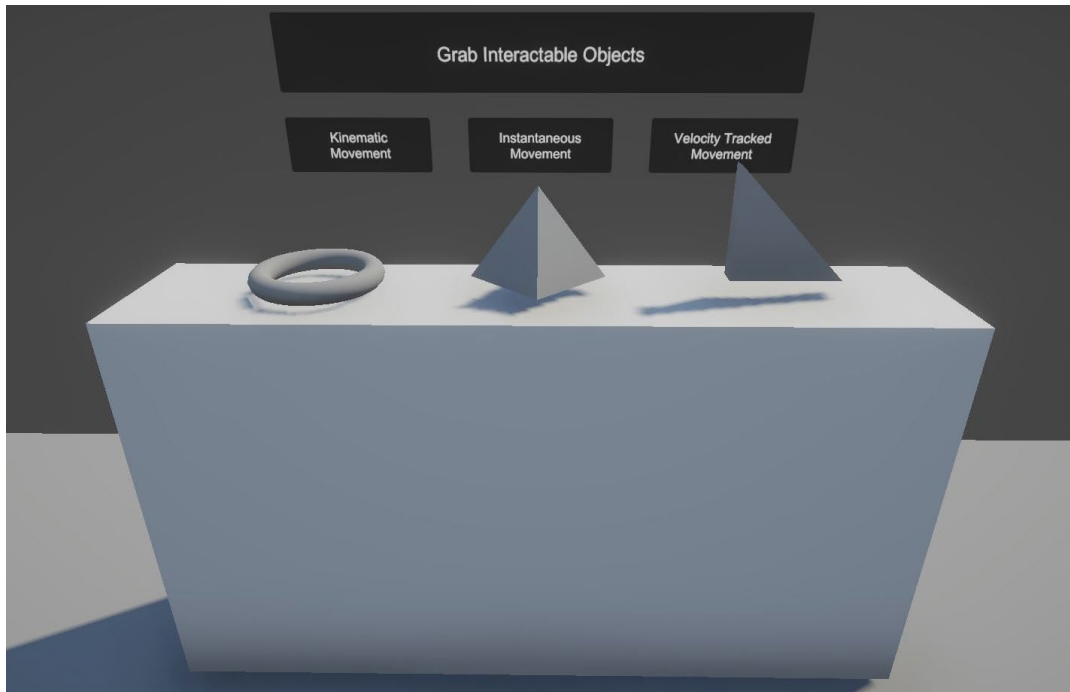
샘플 씬에 적용된 TunnellingVignette 효과



VR 월드와의 상호 작용

VR 게임을 제작할 때는 게임 내 아이템과 상호 작용할 수 있는 기능을 플레이어에게 제공할 수 있어야 합니다.

데모 씬에서는 세 가지 유형의 잡기 인터랙터블 요소를 선보이며 오브젝트를 집어 드는 다양한 방법을 제시합니다. 인터랙터블 오브젝트에는 **XRGrabInteractable** 컴포넌트가 필요합니다.



샘플 씬의 세 가지 잡기 인터랙터블 요소

XRGrabInteractable

XRGrabInteractable 컴포넌트를 사용하면 VR 또는 AR 환경에서 기본적인 잡기 기능을 활성화할 수 있습니다. 이 컴포넌트가 적용된 오브젝트는 인터랙터(VR 핸드 컨트롤러 등)를 사용해 잡거나 선택할 수 있고 그 움직임을 따라갈 수 있습니다.

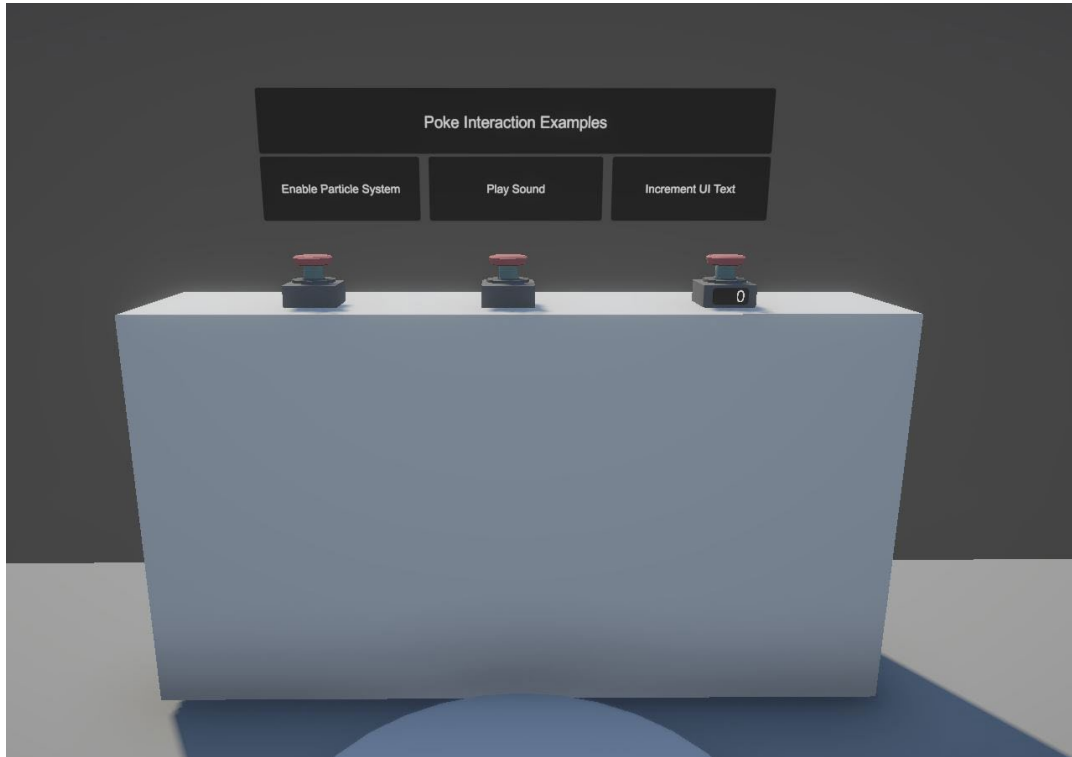
XRGrabInteractable 컴포넌트의 기능으로는 트래킹 또는 다음과 같은 **Movement Type**이 있습니다.

- **Kinematic:** 오브젝트가 물리적인 힘과 충돌을 무시하고 인터랙터와 함께 직접 움직이므로 위치와 회전을 정밀하게 제어할 수 있습니다. 예측 가능한 스크립트 기반의 상호 작용에 적합합니다.
- **Instantaneous:** 오브젝트의 위치와 회전이 물리 계산 없이 인터랙터에 맞춰 즉시 업데이트되므로 반응이 빠르고 직접적인 제어가 가능합니다.
- **Velocity Tracking:** 인터랙터의 모션에 따라 오브젝트의 속도를 동적으로 업데이트하는 방식으로 움직임이 발생하므로, 관성 및 운동량과 같은 사실적인 물리 상호 작용과 자연스러운 동작을 구현할 수 있습니다.



누르기 인터랙터블 요소

데모 씬에는 누르기 기능을 사용하여 세 가지 상호 작용, 즉 파티클 시스템 활성화, 사운드 재생, UI 텍스트 증가를 테스트할 수 있도록 지정된 영역도 포함되어 있습니다.



누르기 인터랙터블 유형이 있는 샘플 씬

XR Simple Interactor

어떤 작업이든 궁극적으로 버튼을 누르거나 가볍게 터치하는 행동을 해야만 수행할 수 있습니다. 누르기 인터랙터블 요소의 루트 게임 오브젝트에는 **XR Simple Interactable** 컴포넌트가 있습니다. 이는 **XRBaseInteractable** 클래스를 확장하여 해당 프로퍼티와 메서드를 상속합니다. **XRSimpleInteractor** 클래스는 다양한 상호 작용 상태와 관련된 이벤트를 제공하는 데 중점을 둡니다. 이러한 이벤트를 통해 개발자는 누르기가 감지될 때의 오브젝트 반응을 커스터마이징할 수 있습니다.

XR Poke Filter

XR Poke Filter는 VR 및 AR 환경에서 오브젝트에 누르기 상호 작용을 추가하는 컴포넌트입니다. 이 컴포넌트는 컴포넌트 자체와 **XRBaseInteractable**을 연결하는 식으로 작동하며, 데모 씬의 **XRSimpleInteractor**에서 살펴볼 수 있습니다. 또한 이러한 커스텀 설정에 따라 오브젝트가 누르기에 반응할지 여부와 그 방법을 결정하는 기능을 제공합니다. 이 클래스에는 초기화, 설정, 클린업을 위한 다양한 라이프사이클 메서드와 에디터 내에서 더 효과적으로 통합하고 디버깅하기 위한 함수가 있습니다. **XR Poke Filter**는 섬세한 누르기 상호 작용을 지원하여 오브젝트의 상호 작용성과 사실감을 향상하는 데 중요한 역할을 합니다.

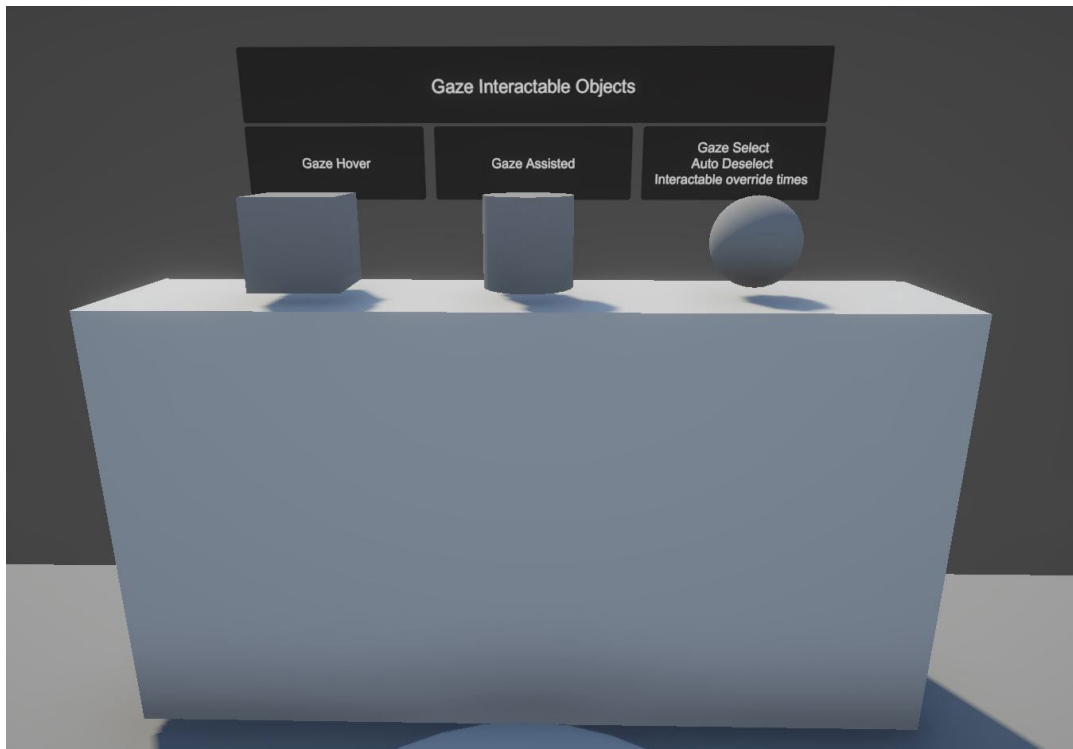
XR Poke Follow Affordance

XRI Toolkit의 `XR Poke Follow Affordance` 클래스는 VR 및 AR 설정에서 버튼과 같은 UI 요소가 눌리거나 상호 작용할 때의 애니메이션을 적용하는 데 사용됩니다. 주요 기능으로는 부드러운 움직임, 상호 작용 후 원래 위치로 돌아가는 능력, 사실적인 상호 작용 경험을 위한 이동 거리 제약 등이 있습니다.

응시 인터랙터블 요소

XRI Toolkit 컨텍스트에서 응시 인터랙터블이란 사용자가 가상 환경에서 단지 오브젝트를 응시하는 것만으로 상호 작용할 수 있는 상호 작용 방법을 말합니다. 특히 핸드 컨트롤러나 터치와 같은 기존 입력 방식이 효과적이지 않거나 사용이 불가능한 VR 및 AR 애플리케이션에 유용하며, 거동이 어려운 사용자도 VR/AR 콘텐츠를 이용할 수 있으므로 접근성 측면에서도 바람직합니다.

아래 이미지에 보이는 샘플 씬에서는 게임에 응시 기능을 사용하는 방법을 세 가지 예시로 선보입니다.



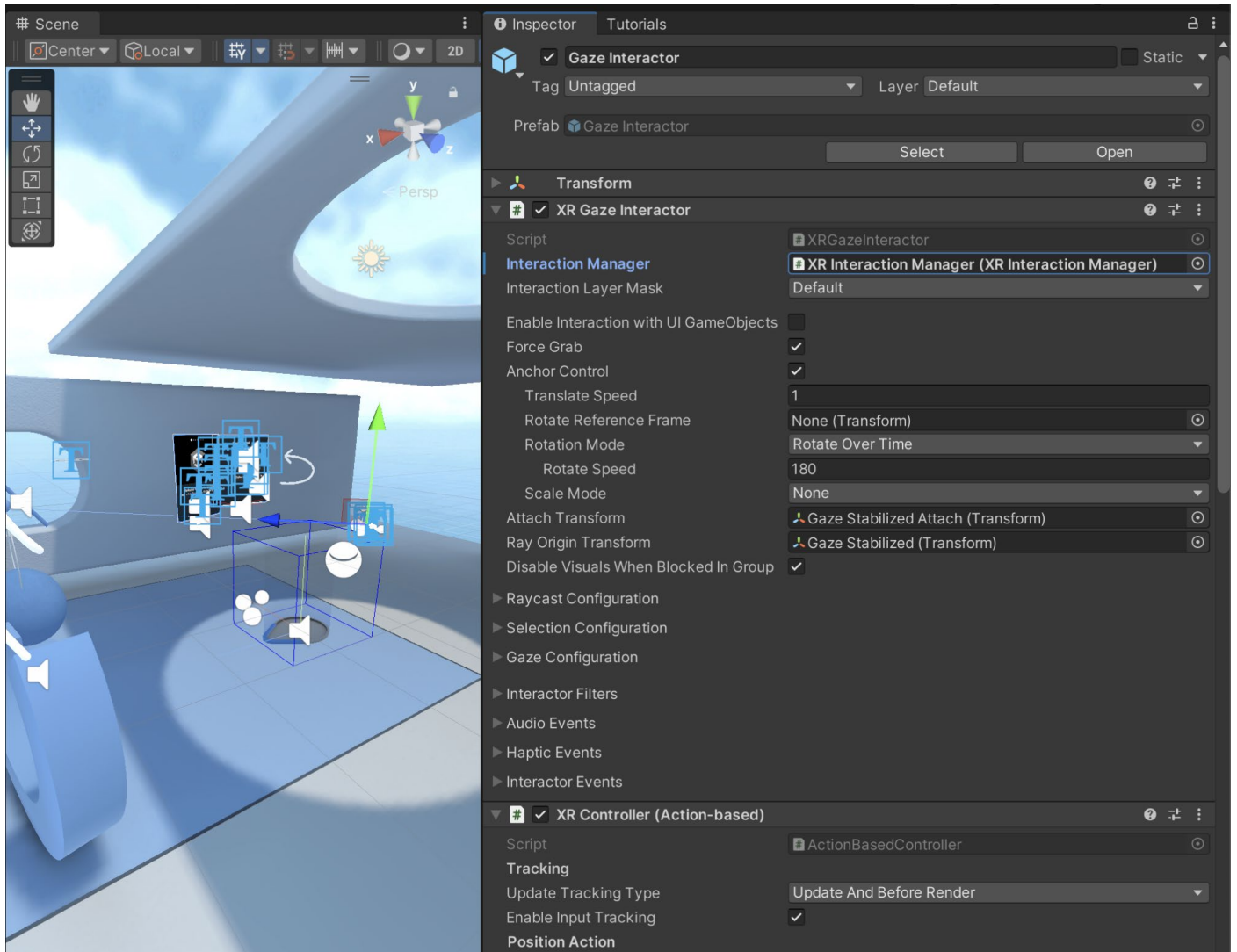
데모 씬의 응시 인터랙터블 설정

씬에서 이러한 인터랙터블 요소 중 하나를 클릭하면 해당 기능이 `XRSimpleInteractor` 컴포넌트에 의해 작동하는 것을 볼 수 있습니다. 이 컴포넌트의 이벤트는 사용자가 응시하면 작동하는 기능을 만드는 데 사용되었습니다.

응시 기능은 주로 XR Origin 내의 Gaze Interactor 오브젝트에 연결된 세 가지 컴포넌트, 즉 XR Gaze Interactor, Gaze Input Manager, Tracked Pose Driver (Input System)에 의해 작동합니다.

XR Gaze Interactor

XR Gaze Interactor는 VR에서 응시를 통해 오브젝트와 상호 작용할 수 있도록 **XRRayInteractor** 스크립트를 확장합니다. 이 스크립트에는 커스터마이징 가능한 응시 보조 기능이 있으므로 상호 작용 영역의 크기를 고정하거나 콜라이더에 기반하여 동적으로 크기를 조정할 수 있습니다. 응시 상호 작용의 감도와 정밀도를 조정하는 프로퍼티도 포함되어 있습니다. 응시를 보조하는 ‘스냅 볼륨’의 자동 생성 및 크기 조절이 지원되므로, 상호 작용의 정확성이 향상됩니다. 또한 이 스크립트는 인터랙터블 타겟의 처리를 관리하며, 다양한 오브젝트와의 상호 작용 기능을 확인할 수 있는 메서드를 포함합니다. **Hover to Select** 및 **Auto Deselect** 시간처럼 응시 상호 작용의 기본 동작을 오버라이드하는 기능도 지원하여, 응시 기반의 VR 상호 작용에 맞는 유연하고 포괄적인 솔루션을 제공합니다.



VR 코어 샘플의 Gaze Interactor 컴포넌트



Gaze Input Manager

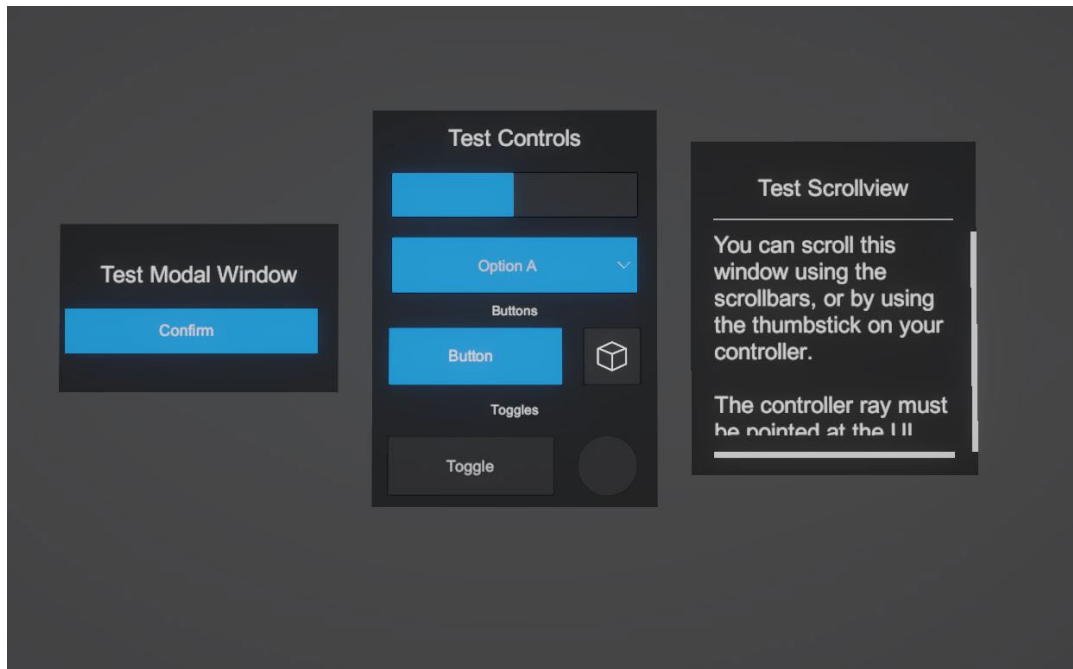
GazeInputModule 스크립트는 시선 트래킹을 사용할 수 없는 경우에 사용할 XRGazeInteractor의 입력 폴백(fallback)을 관리합니다. 시작할 때 시선 트래킹이 지원되는지 확인하고, 시선 트래킹 기기가 있는 경우 로그를 남깁니다. 그러한 기기가 감지되지 않는 경우 기기 연결 이벤트를 구독하여 나중에 기기가 연결되면 시선 트래킹을 활성화합니다. 이 스크립트에는 시선 트래킹이 수행되지 않는 경우 헤드 트래킹으로 폴백하는 토글도 포함되어 있습니다. 따라서 시선 트래킹을 사용할 수 없는 경우 헤드 트래킹으로 전환하여 인터랙터블 기능을 유지함으로써 해당 기능의 지속성을 보장할 수 있습니다. 종합하자면 GazeInputModule는 응시 상호 작용에 사용 가능한 입력 방법을 조정하여 사용자 경험을 향상합니다.

XRGazeInteractor와 XRSimpleInteractor는 함께 작동하며, 멀리서 응시하여 선택하는 방식부터 가까이서 직접 조작하는 방식까지 다양한 시나리오에 대응하는 유연하고 강력한 VR 상호 작용 시스템을 제공합니다.

UI 요소와의 상호 작용

XR 월드를 방문하는 플레이어는 일정 단계가 되면 사용자 인터페이스와 상호 작용을 하게 됩니다. XR Interaction Toolkit 씬 예시는 플레이어의 VR 컨트롤러 입력에 UI 컴포넌트가 어떻게 반응하는지를 보여 줍니다.

데모 씬에는 세 가지 UI 상호 작용 예시가 있습니다. UI 기능을 제작하기 위한 핵심 요소는 Event System 게임 오브젝트에 있는 XR UI Input Module 컴포넌트입니다.



샘플 씬의 UI 요소

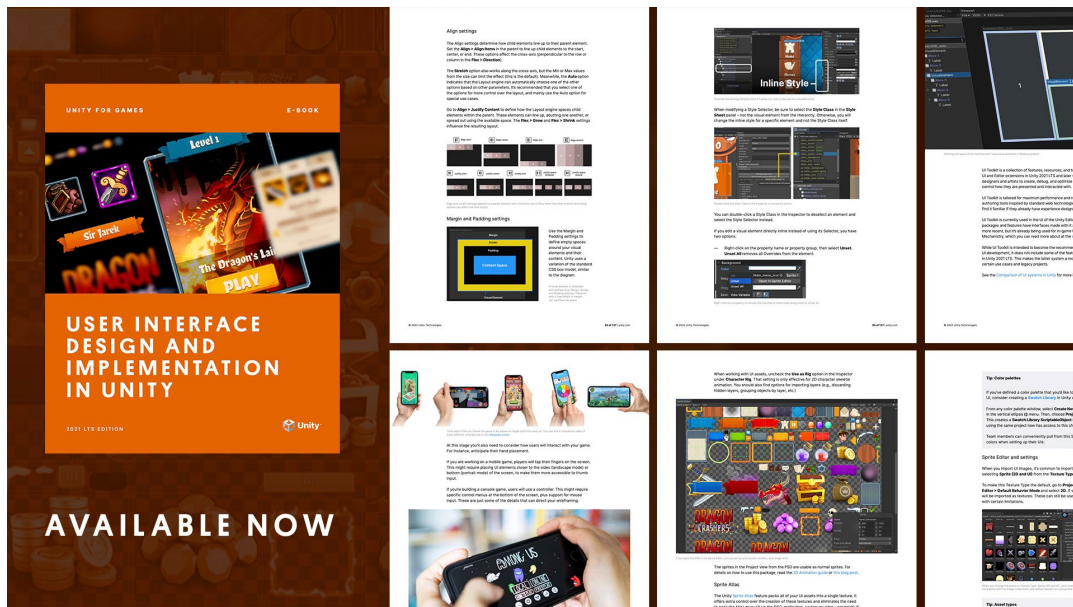


XR UI Input Module 컴포넌트

XR UI Input Module은 XR이 이벤트 시스템과 적절히 연동하는 데 필요한 컴포넌트입니다. XR 환경 사용자와 UI 요소 간의 상호 작용을 관리하며, XR Ray Interactor와 함께 작동하여 UI와 XR의 상호 작용이 제대로 처리되도록 합니다. 3D 트래킹 기기나 마우스, 터치, 게임패드 컨트롤과 같이 XR이 아닌 기타 소스를 비롯한 여러 입력 방법을 지원합니다(설정된 경우). 이 스크립트는 XR 인터랙터가 UI 요소와 상호 작용할 인터페이스를 정의하여 커서 올리기 및 클릭 등의 이벤트를 처리하며, Unity의 레거시 입력 관리자 및 최신 입력 시스템과 모두 호환됩니다. XR 환경에서 이루어지는 사용자 동작을 UI 상호 작용으로 변환하는 다리 역할을 수행합니다.

Tracked Device Graphic Raycaster

각 캔버스 오브젝트에는 **Tracked Device Graphic Raycaster** 컴포넌트가 있습니다. 이 컴포넌트는 XR 환경에서 레이캐스팅을 처리하도록 특별히 설계된 Unity XR Toolkit용 GraphicRaycaster를 커스터마이징하여 구현한 것입니다. XR에서 트래킹되는 기기와 함께 작동하도록 표준 GraphicRaycaster의 기능을 확장하여, 캔버스에 있는 UI 요소와의 상호 작용을 감지할 수 있도록 합니다.



이 무료 전자책과 함께 제공되는 샘플 프로젝트를 통해 Unity의 UI에 대해 자세히 알아보세요.

[다운로드](#)



XR 게임 개발에 도전하는 개발자라면 XRI Toolkit의 예시 씬을 참고하여 시작하는 것이 좋습니다. 이 데모는 기본적으로 여러 가능성을 확인해 볼 수 있는 곳으로, XR 경험의 기초가 되는 다양한 인터랙티브 요소, 컨트롤, 메카닉을 선보입니다.

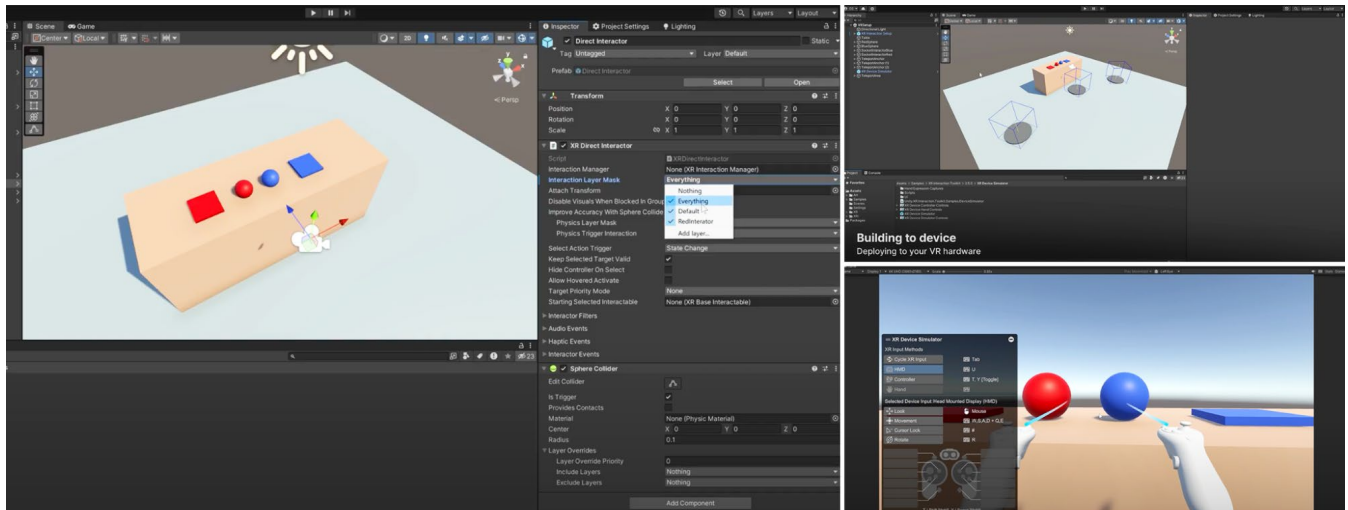
이 씬을 살펴보고 오브젝트 조작, 순간 이동, 3D 공간에서의 UI 상호 작용, 컨트롤러 입력 관리 등의 다양한 XR 기능을 구현하는 데 필요한 아이디어를 얻으세요.

이 씬의 장점은 모듈식으로 구성되어 있어 해당 요소의 수정과 결합, 확장이 간편하다는 것입니다. 몰입형 VR 게임, 교육용 AR 경험 및 그 외 다양한 작품을 제작하려는 경우 이 예시 씬이 비전을 실현하기 위한 발판이 될 수 있습니다.

프로젝트 예시를 통해 알아보는 VR 개발

이 섹션에서는 Unity에서 VR 프로젝트를 개발하기 위한 몇 가지 주요 단계를 소개합니다. 여기서는 각 개발 단계와 관련된 의사 결정은 물론 Unity를 기반으로 한 실무 단계와 워크플로를 다룹니다.

새 프로젝트를 시작하는 과정을 단계별로 따라하고 싶다면 이 동영상상을 시청하세요.



XR 전자책과 함께 제공되는 이 튜토리얼을 참조하세요. 해당 동영상에서는 Unity 2022 LTS용 XR Interaction Toolkit 2.5.x를 살펴봅니다.



튜토리얼 보기



아이디어와 컨셉 구상

게임 아이디어는 순식간에 떠오르기도 하고, 더 신중한 과정을 통해 오랜 시간에 걸쳐 떠오르기도 합니다. 어느 쪽이든 진정으로 열정을 쏟을 수 있는 장르나 테마로 컨셉을 고수하는 것이 좋습니다.



Unity에서 헤드셋을 이용한 초기 프로토타이핑

초기 아이디어 구상 단계에서는 폭넓고 개방적인 접근 방식을 유지하세요. 이 시점부터 세세한 부분까지 열중할 필요는 없으며, 순간적으로 떠오르는 아이디어에 살을 붙이며 자연스럽게 구체화해 보면 됩니다. 큰 그림 작품의 윤곽을 스케치한다고 생각하세요. 아직 세부 사항은 중요하지 않습니다. 그 대신 전반적인 주제, 불러일으키려는 감정, 플레이어에게 제공하고자 하는 핵심 경험에 집중하세요. 이 시점에는 현실적 제약 또는 기술적 제약에 구애받지 않고 상상력을 자유롭게 펼칠 수 있습니다. 세상을 놀라게 하는 모든 게임은 머릿속을 스치는 찰나의 아이디어로 시작되며, 그 아이디어를 꽃피우는 시기가 바로 이 단계입니다.

이 섹션에서 전개하려는 아이디어 예시는 플레이어가 단서를 해결해 탈출해야 하는 공포 방 탈출 게임입니다. 간략한 소개는 이 정도로 하고, 이제 구체적으로 알아보겠습니다.



사전 계획

초기 아이디어를 준비했다면 이제 이 넓은 범위의 컨셉을 체계적인 계획으로 전환해야 합니다. 이는 개발 프로세스의 토대를 마련하는 매우 중요한 단계입니다. 주요 단계는 다음과 같습니다.

- 1. 핵심 게임플레이 메카닉 정의:** 먼저 게임의 기본 메카닉을 자세히 기술합니다. 플레이어는 VR 환경과 어떻게 상호 작용하며, 플레이어가 주로 하게 될 행동은 무엇인가요? 우리의 예시 공포 게임에서는 으스스한 환경을 탐험하고, 퍼즐을 풀고, 위협을 피하거나 맞서 싸워야 할 수 있습니다. 이러한 메카닉을 Unity에서 어떻게 구현할지도 고민해야 합니다. 예를 들어 XRI Toolkit에서는 [XR Socket Interactor](#)처럼 퍼즐을 만드는 데 유용한 기능을 제공합니다.
- 2. 기본 스토리라인 또는 테마 만들기:** 내러티브 중심적인 게임이 아니더라도 기본적인 스토리라인이나 테마가 있어야 개발 방향을 설정할 수 있습니다. 공포 게임에서는 설정(귀신의 집, 버려진 정신병원, 으스스한 지하 묘지 등)과 배경 스토리, 주요 캐릭터 또는 적을 결정하는 작업을 이 단계에 해야 할 수 있습니다.
- 3. 레벨 디자인 및 환경 스케치:** 먼저 게임 환경을 시각화합니다. 이 단계에서는 레벨이나 주요 영역의 대략적인 레이아웃을 스케치합니다. 플레이어가 그러한 공간을 어떻게 탐색할지, 레이아웃이 게임 분위기에 어떻게 기여할 수 있는지 고려합니다.
- 4. USP(고유의 강점) 식별:** 동종 장르의 다른 게임과 어떻게 차별화할 수 있으며, 게임플레이 메카닉이 독특한가요? 스토리는 설득력 있으며, 혁신적인 VR 기술을 사용했나요? 특정 아트 스타일과 사운드 디자인이 적용됐나요? USP는 플레이어를 유치하고 유지하는 핵심 요소입니다.
- 5. 프로토타입 개발:** 먼저 간단한 프로토타입을 사용하여 핵심 게임플레이 메카닉을 테스트합니다. 게임플레이 방식을 파악하고 초기에 조정하는 것이 목적이므로 복잡하게 할 필요는 없습니다. VR에서 프로토타이핑하는 것은 일반 화면에서는 측정하기 어려운 스케일, 거리감, 상호 작용을 체험할 수 있으므로 매우 중요합니다.
- 6. 초반 피드백 수집:** 친구, 동료 개발자 또는 잠재적 플레이어에게 프로토타입을 선보입니다. 초기 피드백은 게임 컨셉과 메카닉의 강점과 약점을 파악하는 데 유용합니다.
- 7. 개발 프로세스 계획:** 앞서 언급된 단계를 마쳤다면 이제 프로젝트의 주요 마일스톤에 대한 윤곽을 잡을 시간입니다. 세부적인 게임 디자인, 아트 및 사운드 제작, 프로그래밍, 테스트 및 다듬기 작업이 포함될 수 있습니다. 1인 개발자나 소규모 팀이라면 현실적인 타임라인과 우선순위를 설정해 보세요.
- 8. 기술적 요구 사항 고려:** VR/AR 개발에서 고려해야 할 구체적인 기술적 사항으로는 멀미를 방지하기 위한 안정적인 프레임 속도 보장, 다양한 VR 하드웨어에 대한 최적화, 직관적인 VR 컨트롤 구현 등이 있습니다.
- 9. 디자인 문서화:** 상세한 GDD(게임 디자인 문서)를 작성합니다. 게임 메카닉, 스토리, 레벨 디자인, 아트 및 사운드 방침, 사용자 인터페이스 디자인, 기술적 요구 사항에 해당하는 모든 내용을 포함해야 합니다.
- 10. 예산 및 리소스 관련 계획 수립:** 소프트웨어, 하드웨어 및 기타 프리랜서 작업물을 비롯하여 현재 보유하고 있는 리소스와 필요한 리소스를 평가합니다.



이 단계는 초기 아이디어를 실행 가능한 최종 게임 개발 계획으로 전환하는 데 도움이 됩니다. 유니티에서 제공하는 이 두 권의 전자책을 통해 게임 기획 및 개발 방법에 대한 실용적인 팁과 아이디어를 추가로 얻을 수 있습니다.



[다운로드](#)



[다운로드](#)

다시 공포 게임 예시로 돌아가겠습니다. 위 단계를 진행한 우리는 이제 플레이어가 지하 묘지를 탐험하는 5~10분 분량의 VR 경험을 제작하는 것이 목표임을 알게 되었습니다. 플레이어는 방 중앙에 있는 관을 열 수 있는 열쇠를 얻기 위해 세 가지 퍼즐을 풀어야 합니다. 관을 통해 지하 묘지를 탈출할 수 있으나, 공포심과 압박감을 더하기 위해 못이 박힌 천장이 천천히 아래로 내려오기 때문에 플레이어는 재빠르게 행동해야 합니다.

엔진 시작하기

Unity 버전 및 URP

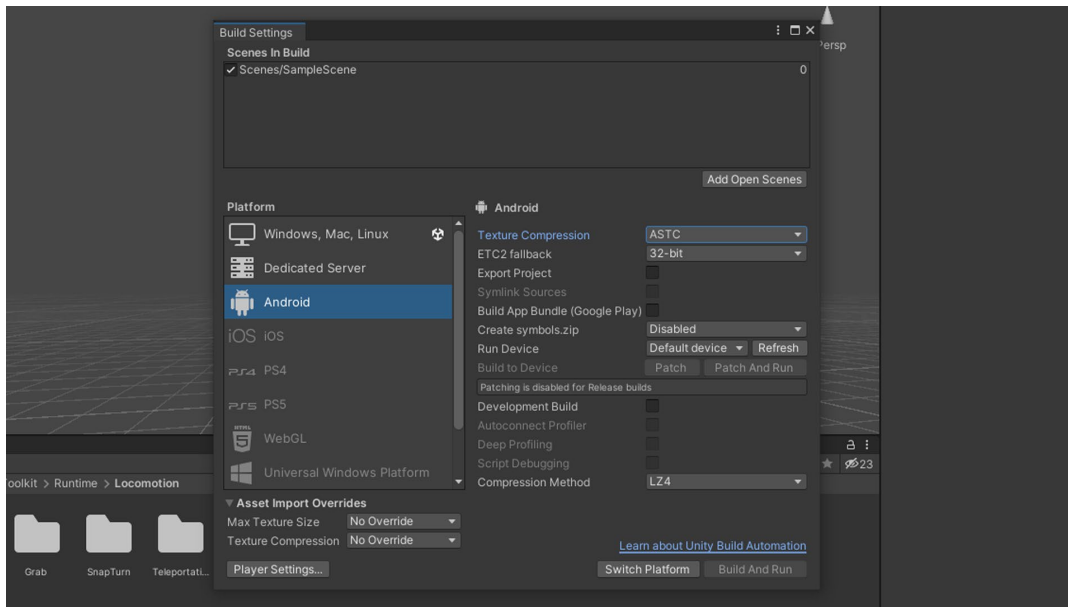
이제 Unity에서 게임을 제작해 볼 차례입니다. 이 예시에서는 XR 애플리케이션용으로 가장 성능이 뛰어난 렌더 파이프라인인 URP와 Unity 2022 LTS를 사용합니다. 타겟 하드웨어는 Meta Quest 2 헤드셋입니다.

XR Plug-in Management 패키지를 설치하고 모든 플랫폼에 대해 OpenXR을 선택합니다.

그런 다음 [인터랙션 프로파일](#)을 할당합니다. 이 예시에서는 Meta Quest 2 헤드셋을 사용하므로 **Meta Quest Touch Pro 컨트롤러 프로파일**을 추가합니다.

빌드 설정

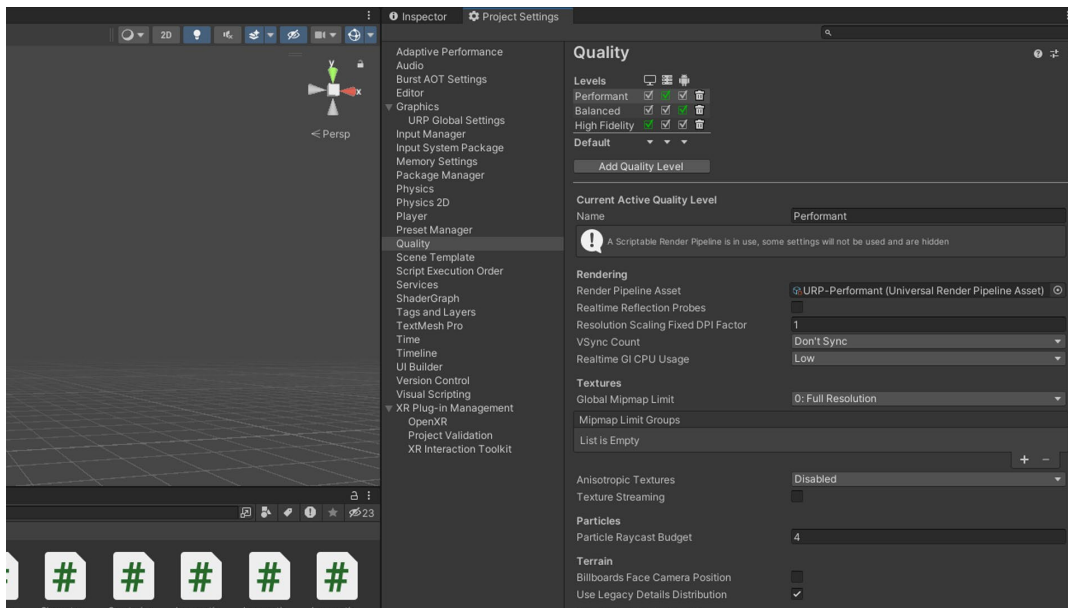
Build Settings에서 타겟을 Android로 설정하고 텍스처 압축을 VR 애플리케이션에 적합한 효율적 방법인 ASTC(Adaptive Scalable Texture Compression)로 변경합니다. 이렇게 하면 VR의 고품질 텍스처에 대한 수요와 기기의 성능 제약 사이에서 균형을 맞추는 데 도움이 됩니다.



공포 VR 프로젝트 예시의 Build Settings

품질 설정

마지막 설정 단계는 Project Settings 창을 통해 Quality 설정을 조정하는 것입니다. **Performant**를 기본값으로 선택합니다.



Quality 설정에서 Performant를 기본값으로 선택



고성능 URP 에셋은 렌더링 성능을 최적화하도록 설정됩니다. 최적화는 드로우 콜을 줄이고 셰이더 실행을 최적화하며 조명 및 그림자 렌더링을 더욱 효과적으로 제어하는 방식으로 이루어집니다. 타겟 플랫폼에서 고성능을 유지하면서도 시각적으로 풍부한 애플리케이션을 제작할 수 있으며, 이는 VR에서 몰입감을 유지하고 멀미를 방지하려면 반드시 필요한 요소입니다. URP는 [스크립터블 렌더 파이프라인](#)이므로 프로젝트의 특정 요구 사항에 맞게 렌더링 프로세스를 조정할 수도 있습니다.

전반적으로, 추구해야 할 중요한 목표는 최저 사양의 타겟 기기에서 원활하게 실행되는 멋진 게임을 만드는 것이며, 이는 후반부 빌드 작업의 강력한 기반이 됩니다.

월드 구성

프로토타입 및 모델링

이 단계에서는 3D 환경의 프로토타이핑을 포함한 아트 파이프라인이 시작됩니다. 이 공포 게임 예시는 지하 묘지라는 단일 환경에서 진행되므로 매우 간단합니다.

먼저 3D 블로킹을 진행해 공간의 규모를 확인하고 퍼즐을 배치할 위치를 알아봅니다. 영역은 단순한 셰이프로 형성할 수 있으므로, 이제 천장이 내려오는 메카닉에 대한 고민을 시작할 수 있습니다.



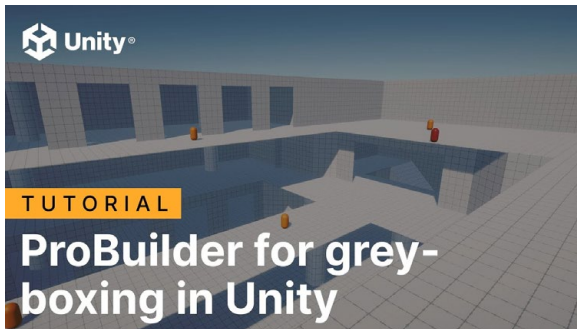
환경 블로킹

환경의 셰이프로는 오각형이 선택되었고, 플레이어가 지하 묘지로부터 탈출할 수 있는 경로가 될 작은 터널이 있습니다.

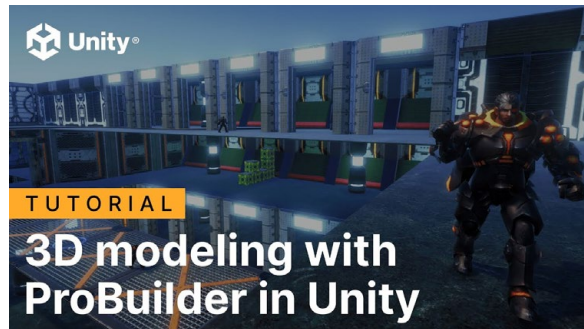


Unity에서는 **ProBuilder** 패키지로 효율적인 프로토타이핑이 가능합니다. 플레이할 환경 내에서 제작할 수 있으므로 유연성과 창의성을 극대화하고 반복 작업(iteration) 사이클을 효율화할 수 있습니다. 씬 내에서 스케일을 제대로 조정하는 것이 중요하므로 ProBuilder는 VR 애플리케이션에 유용합니다. ProBuilder를 사용해 아이디어를 모형으로 구현한 다음 헤드셋을 착용하여 테스트할 수 있으므로 개발 기간이 단축됩니다.

앞서 링크를 통해 알려드린 **레벨 디자인 전자책**의 한 섹션에서는 ProBuilder 툴 사용 방법에 대한 방대한 내용을 다루며, 다음 동영상 튜토리얼에서도 유용한 팁을 배울 수 있습니다.



[튜토리얼 보기](#)



[튜토리얼 보기](#)

효율적인 텍스처링

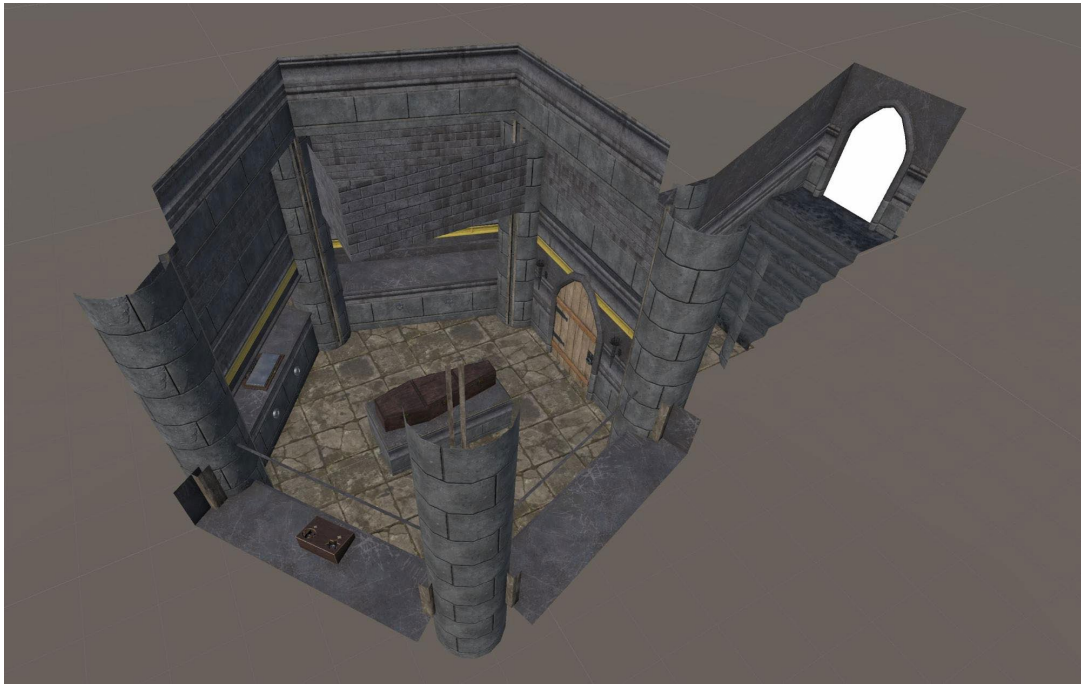
블로킹한 콘텐츠로부터 이제 모델링을 진행하고 텍스처를 추가할 수 있습니다. 공포 게임 예시에는 반복되는 요소가 많으므로 기둥, 벽, 출입구로 구성된 작은 모델 키트를 더 쉽게 만들 수 있습니다. 이러한 키트를 복제하여 환경을 구성할 수도 있고, 아래 이미지에서 볼 수 있듯이 트림 시트를 사용하여 전체 구조를 텍스처링할 수 있도록 여러 개를 연결해 일체화된 하나의 모델을 만들 수도 있습니다.



환경용 트림 시트



트림 시트 영역이 모델 영역에 적용되면 다음 이미지처럼 텍스처가 적용된 지하 묘지가 만들어집니다.



텍스처가 적용된 3D 환경

자세히 살펴보면 구조의 영역이 트림 시트에 어떻게 할당되는지 확인할 수 있습니다.

프로젝트 예시에서 계속해서 프랍, 퍼즐, 출구를 모두 만들고 Unity로 가져와 셰이더를 할당합니다(이 경우 기본값은 [URP 릿 셰이더](#)). 이제 이 환경에 조명을 사용하는 방법을 살펴보겠습니다.

조명 적용하기

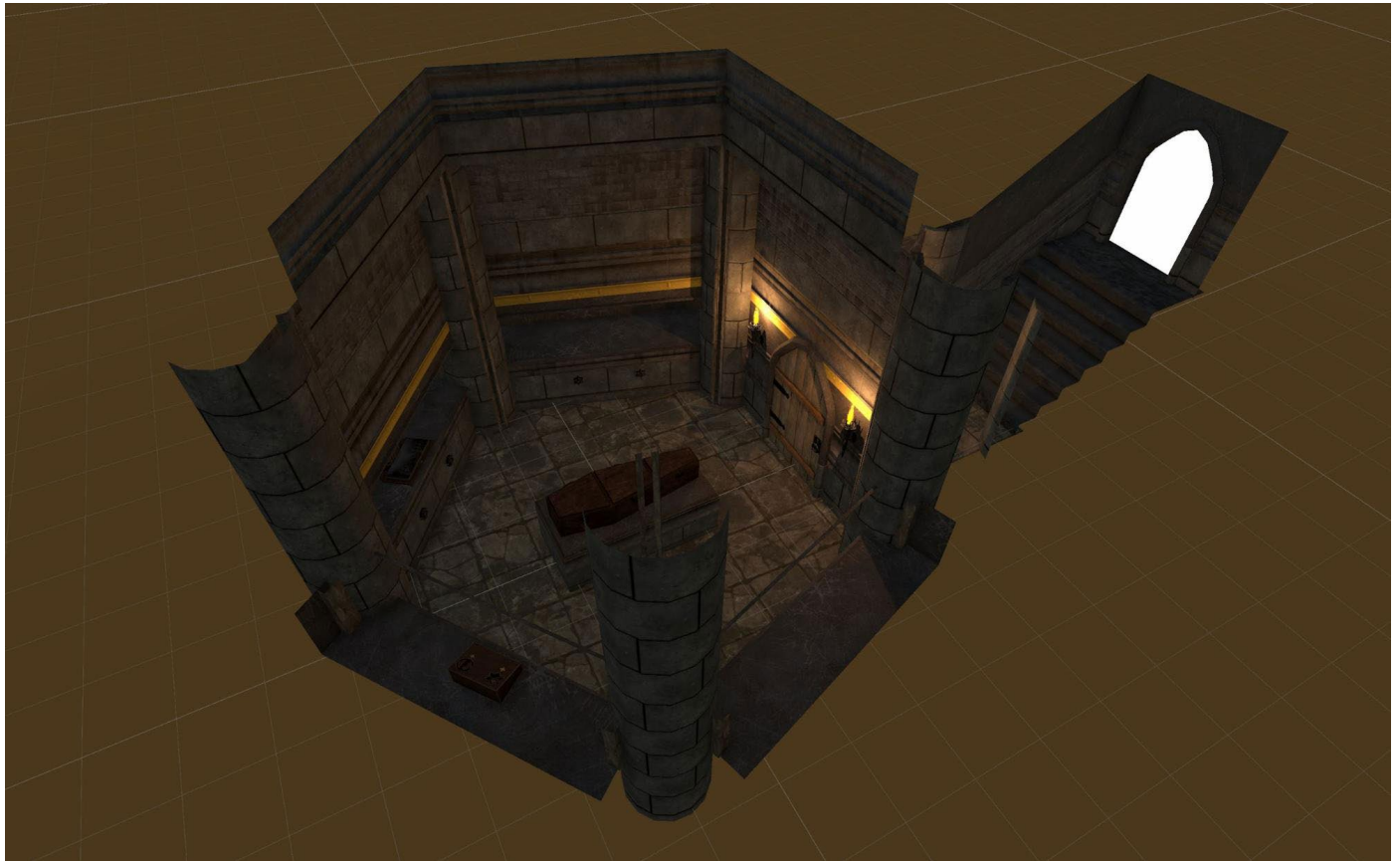
조명 방법 선택하기

지하 묘지의 광원은 채광창과 문 양쪽에 있는 두 개의 횃불에서 나옵니다. 횃불의 광원은 정적이므로 이 씬에는 베이킹된 조명(조명 정보를 미리 계산하여 텍스처 맵에 베이킹함)이 적합합니다.

하지만 문이나 관 뚜껑처럼 움직이는 오브젝트는 베이킹된 조명에서 그림자를 드리우지 않으므로, 정적 오브젝트에는 베이킹된 조명을 계산하고 동적 오브젝트는 실시간 조명을 받도록 혼합 조명을 사용해 볼 수 있습니다. 그렇다 하더라도 결국 플레이어가 관 뚜껑을 열 때 그림자가 없다는 사실을 눈치채지 못할 것이므로 베이킹된 조명을 선택하겠습니다. 숙련된 개발자라면 실제 프로젝트에서 이런 식의 타협이 이루어진다는 사실을 잘 알고 있을 것입니다.



아래는 베이킹된 조명이 적용된 씬의 이미지입니다.



조명을 베이킹한 결과물

이 씬은 하나의 512x512 라이트맵으로 조명을 비춥니다. 씬을 적절히 설정하려면 성능과 품질 사이의 균형을 맞춰야 합니다.

조명 최적화

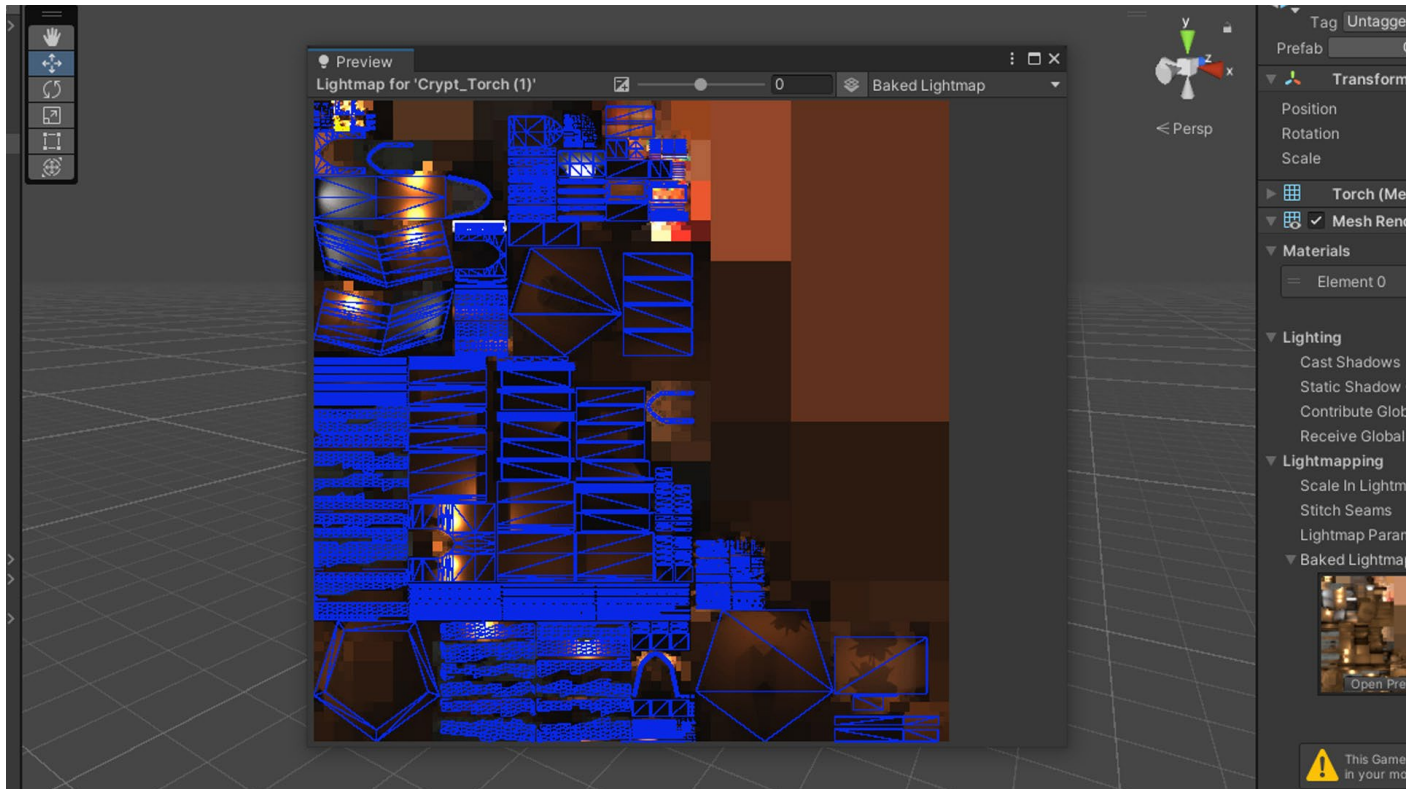
지하 묘지와 같은 소규모 씬에서는 하나의 라이트맵이 목표입니다. 라이트맵의 원래 크기는 1024로, 이 상태로는 멋지게 보였습니다. 테스트를 위해 이를 512로 줄여서 3개의 라이트맵을 생성하고 드로우 콜에 추가합니다.

[드로우 콜](#)이란 특정 설정(예: 머티리얼 및 텍스처)으로 지오메트리 배치를 렌더링하도록 GPU에 명령하는 것입니다. 드로우 콜이 발생할 때마다 일정량의 처리 오버헤드가 발생하므로 드로우 콜 수를 줄이면 특히 저사양 하드웨어에서 성능이 향상될 수 있습니다.

텍셀 밀도를 줄여 모든 것을 하나의 512 라이트맵에 담을 수 있습니다. 텍셀 밀도란 라이트맵의 픽셀과 3D 월드에서 이 픽셀이 나타내는 실제 물리적 공간의 비율을 말합니다. 이는 표면의 단위 면적을 나타내는 데 얼마나 많은 라이트맵 픽셀(텍셀) 수가 사용되는지 알 수 있는 척도입니다. 텍셀 밀도가 높을수록 단위 면적당 더 많은 텍셀이 사용되므로 해당 표면에 적용되는 조명 및 그림자 효과의 디테일과 해상도가 증가합니다. 반대로 텍셀 밀도가 낮으면 단위 면적당 더 적은 텍셀이 사용되므로 조명과 그림자의 디테일이 감소합니다.



텍셀 밀도 값을 40에서 20으로 변경하고 다시 베이킹해 보겠습니다. 이번에는 512x512 라이트맵 하나가 생성됩니다.



지하 묘지 환경의 베이킹된 라이트맵 데이터

라이트맵의 품질은 양호하며 헤드셋으로 테스트했을 때 전반적인 비주얼 품질에 미치는 영향은 미미합니다. 지금은 조명을 그대로 두었다가 나중에 게임이 성능 면에서 매우 뛰어난 경우 품질을 높이면 됩니다.

Unity 프로젝트에서 조명 문제를 해결하는 방법은 이 [블로그](#)를 읽어 보세요.

이제 플레이어가 이 공간을 어떻게 탐색하는지 살펴보겠습니다.

가상 월드 탐색

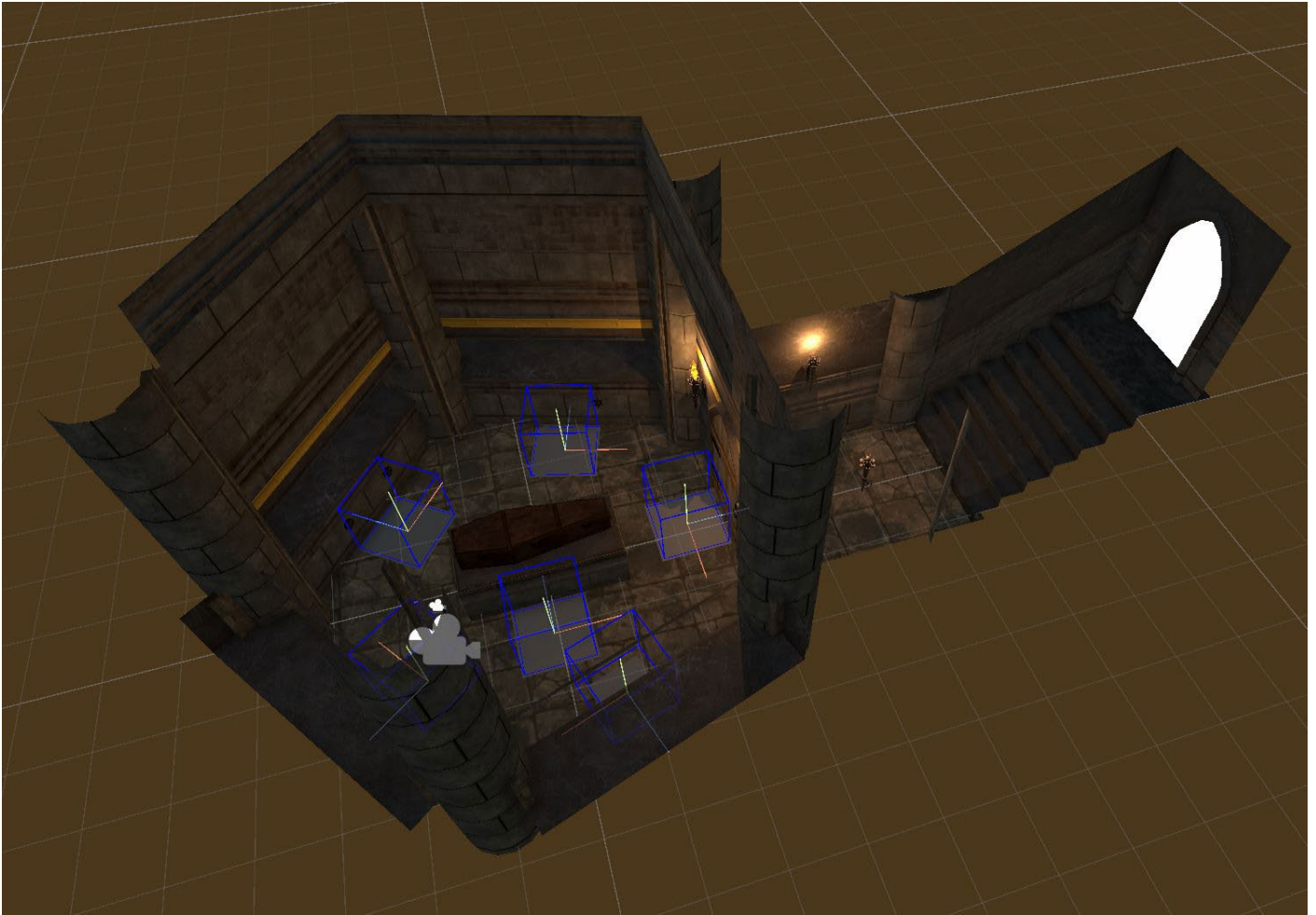
썬에 조명을 비추었으니 이제는 플레이어가 썬에서 움직일 수 있게 하는 기능을 추가하겠습니다. 데모 썬에 사용된 것과 동일한 프리팹을 XR 릭에 추가합니다. 그런 다음 빈 게임 오브젝트를 추가하고 이에 XR interaction Manager 컴포넌트를 추가합니다.

순간 이동 유형

플레이어가 어디서든 자유롭게 순간 이동하도록 할까요, 아니면 순간 이동 앵커를 사용할까요? 지하 묘지 환경에서 순간 이동 앵커를 사용하면 썬의 중요한 영역을 식별할 수 있으며, 이는 플레이어에게 일종의 힌트 역할을 합니다. 플레이어가 순간 이동할 수 있는 위치라면 틀림없이 중요한 위치일 것입니다. 시간 제한이 있는 게임의 특성상 이 방식은 바람직한 해결책입니다.



퍼즐 위치, 관, 문에 순간 이동 앵커를 배치해 보겠습니다.



순간 이동 앵커의 위치

순간 이동 앵커를 사용하면 순간 이동할 때 플레이어가 마주하게 될 방향을 커스터마이징할 수 있습니다. 예시 게임에서는 플레이어가 순간 이동했을 때 특정 위치에서 중요한 오브젝트(예: 퍼즐)를 마주해야 합니다. 그렇게 구현하려면 순간 이동 설정을 **Target Up And Forward**로 지정하여 플레이어를 순간 이동 앵커의 방향과 일치시켜야 합니다.

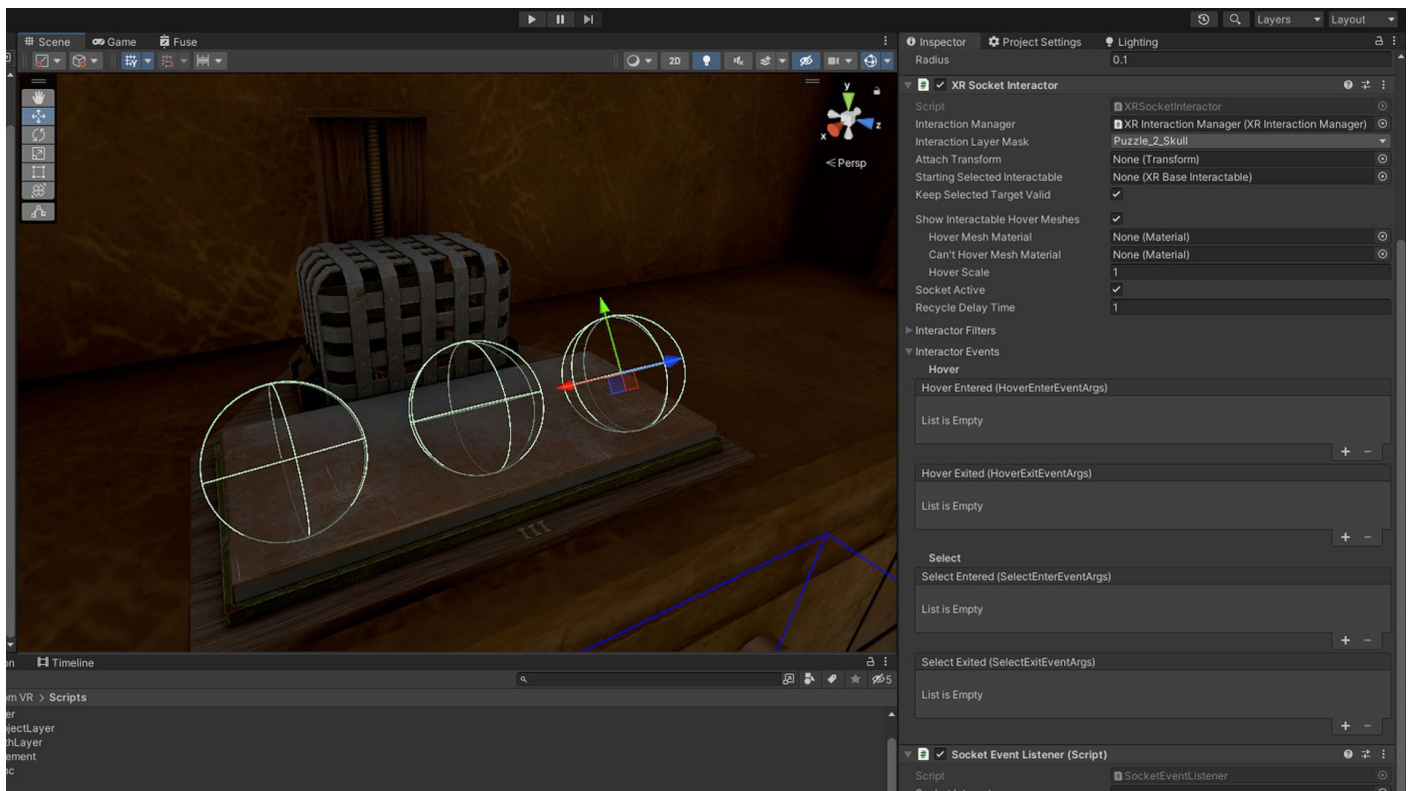
컴포넌트의 스타일은 쉽게 커스터마이징할 수 있습니다. 예를 들어 순간 이동기의 광선 스타일을 변경하려면 XR Origin의 Right Controller에서 **Teleport Interactor**를 선택하면 됩니다. XR Ray Interactor의 **Raycast Configuration**을 통해 줄바꿈 타입을 변경할 수 있으며, **XR Interactor Line Visual**을 통해 색상을 제어할 수 있습니다.



XR Interaction Toolkit을 사용하여 퍼즐 만들기

아래의 공포 게임 예시 씬에서 플레이어는 두 개의 흥미로운 퍼즐을 받고, 각 퍼즐을 풀 때마다 보상으로 열쇠를 획득합니다. 이 열쇠는 방 중앙에 있는 관의 잠금을 푸는 필수 아이템입니다. 두 퍼즐 모두 비슷한 인터랙티브 디자인을 공유하며, 방을 탐색하면서 퍼즐을 푸는 데 사용할 수 있는 특정 아이템을 발견하는 과제를 플레이어에게 부여합니다.

첫 번째 퍼즐은 방에 숨겨진 아이템을 찾는 것입니다. 이 아이템을 찾으면 플레이어는 서랍의 잠금을 해제하여 열쇠 중 하나를 발견할 수 있습니다. XR Interaction Toolkit을 통해 이 모든 인터랙티브 요소를 원활하게 통합하고 관리함으로써, 플레이어에게 매끄럽고 몰입도 높은 경험을 제공할 수 있습니다.



공포 게임 예시에서 소켓 인터랙터를 활용한 퍼즐

이 퍼즐에 사용할 수 있는 기능은 XR Grab Interactor와 XR Socket Interactor의 조합입니다. 예를 들어 이 퍼즐에는 내부에 오브젝트를 삽입해야 하는 소켓 인터랙터가 3개 있습니다.



아래에서 자세히 살펴볼 XR Socket Interactor에는 소켓 인터랙터가 상호 작용할 레이어를 지정할 수 있는 Interaction Layer Mask가 있습니다. 유효한 오브젝트가 소켓에 삽입되면 커스텀 스크립트에서는 XR Socket Interactor에서 들어오는 이벤트를 수신 대기합니다.

```
1 using UnityEngine;
2 using UnityEngine.XR.Interaction.Toolkit;
3
4 public class SocketEventListener : MonoBehaviour
5 {
6     [SerializeField] private XRSocketInteractor socketInteractor;
7     public Puzzle linkedPuzzle;
8
9     private void OnEnable()
10    {
11        socketInteractor.selectEntered.AddListener(SocketActivated);
12        socketInteractor.hoverEntered.AddListener(SocketHovered);
13        socketInteractor.hoverExited.AddListener(SocketHoverExit);
14    }
15
16    private void OnDisable()
17    {
18        socketInteractor.selectEntered.RemoveListener(SocketActivated);
19        socketInteractor.hoverEntered.RemoveListener(SocketHovered);
20        socketInteractor.hoverExited.RemoveListener(SocketHoverExit);
21    }
22
23    private void SocketActivated(SelectEnterEventArgs arg0)
24    {
25        linkedPuzzle.PuzzlePieceFound();
26        arg0.interactableObject.transform.GetComponent<Collider>().enabled = false;
27    }
28
29    private void SocketHovered(HoverEnterEventArgs arg0) => arg0.interactableObject.transform.GetComponent<ResetGrabbedItem>().enabled = false;
30
31    private void SocketHoverExit(HoverExitEventArgs arg0) => arg0.interactableObject.transform.GetComponent<ResetGrabbedItem>().enabled = true;
32 }
33
```

소켓 인터랙터를 수신 대기하는 커스텀 스크립트

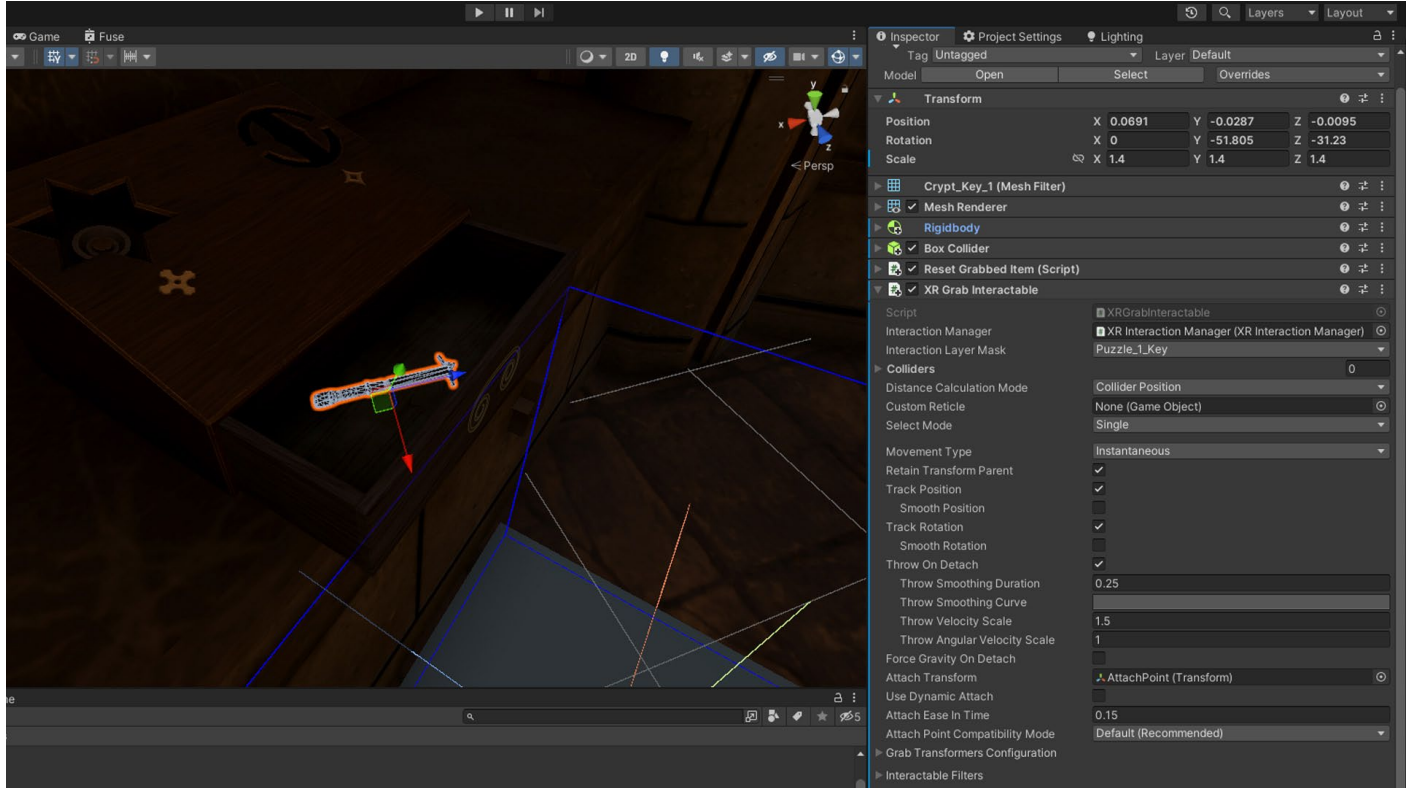
게임플레이를 제어하는 몇 가지 간단한 커스텀 클래스와 함께 XR Interaction Toolkit의 핵심 컴포넌트를 활용하면 복잡한 몰입형 상호 작용을 만들 수 있습니다.

공포 게임 예시에서 사용된 XR Interaction Toolkit 기능 몇 가지를 자세히 살펴보겠습니다.



XR Grab Interactor

이 잡기 인터랙터에 대한 내용은 이 전자책의 [VR 월드와의 상호 작용](#) 섹션에서 소개했습니다. 퍼즐에서 사용자가 집어 들 각 게임 오브젝트에는 XR Grab Interactor 컴포넌트가 할당됩니다.



XR Grab Interactable이 적용되어 잡을 수 있는 오브젝트

XR Socket Interactor

XRSocketInteractor 클래스는 XR 환경에서 소켓 기반 상호 작용을 처리하기 위해 설계된 정교한 툴입니다. 인터랙터를 오브젝트를 소켓에 연결할 때 No Scaling, Fixed Scaling, 또는 Dynamic Scaling과 같은 모드를 사용하여 인터랙터블 오브젝트의 크기 조정 방식을 정의하는 기능을 포함하여 다양한 기능을 제공합니다.

이 클래스는 가능한 상호 작용을 나타내는 호버 메시(hover mesh)를 표시하여 사용자 경험을 향상하고 소켓에 오브젝트를 스냅핑하는 옵션을 제공합니다. 사용 가능한 상호 작용과 사용 불가능한 상호 작용에 대한 Hover Mesh Material 설정이 있으며, 이를 통해 소켓 상호 작용의 활성화 및 비활성화가 가능합니다. 이 스크립트는 충돌과 타겟 처리를 관리하여 유효한 상호 작용 타겟과 충돌 이벤트를 적절히 처리합니다. 소켓 변환 및 인터랙터를 처리를 위한 고급 커스텀 옵션을 사용할 수 있습니다(소켓 스냅핑 반경 조정, 인터랙터를 변환 등). 또한 XRSocketInteractor는 다양한 Unity 라이프사이클 이벤트를 처리하여 XR 환경 내에 있는 인터랙터블 요소를 강력하고 효율적으로 관리하도록 합니다.

여기서는 잡은 오브젝트를 퍼즐에 배치하는 데 이 컴포넌트를 활용할 수 있습니다.



소켓 인터랙터 만들기

첫 번째 퍼즐에서는 두 개의 오브젝트를 특정 영역에 배치해야 합니다. 빈 게임 오브젝트를 생성하고 XR Socket Interactor 컴포넌트를 추가합니다. 스피어 콜라이더가 추가되고 트리거로 표시됩니다. 이 소켓 인터랙터는 특정 오브젝트 1개만 받아들여야 하며, 이 기능을 구현하려면 **Interaction Layer Mask** 드롭다운을 사용하여 레이어를 지정해야 합니다. 여기에 새 레이어를 생성하고 할당하면 소켓 인터랙터는 이 특정 레이어로만 제한됩니다. 이 드롭다운을 사용하여 **PuzzlePeice1**이라는 레이어를 생성하고 할당합니다.

적절한 오브젝트 하이라이트 메시지를 보여 주는 소켓 인터랙터



다음 단계는 이 소켓 인터랙터에 맞는 퍼즐 조각을 찾아 XR Grab Interactable의 Interaction Mask를 소켓 인터랙터와 일치하도록 변경하는 것입니다. 이제 잡을 수 있는 이 특정 오브젝트만 소켓 인터랙터에서 작동합니다.

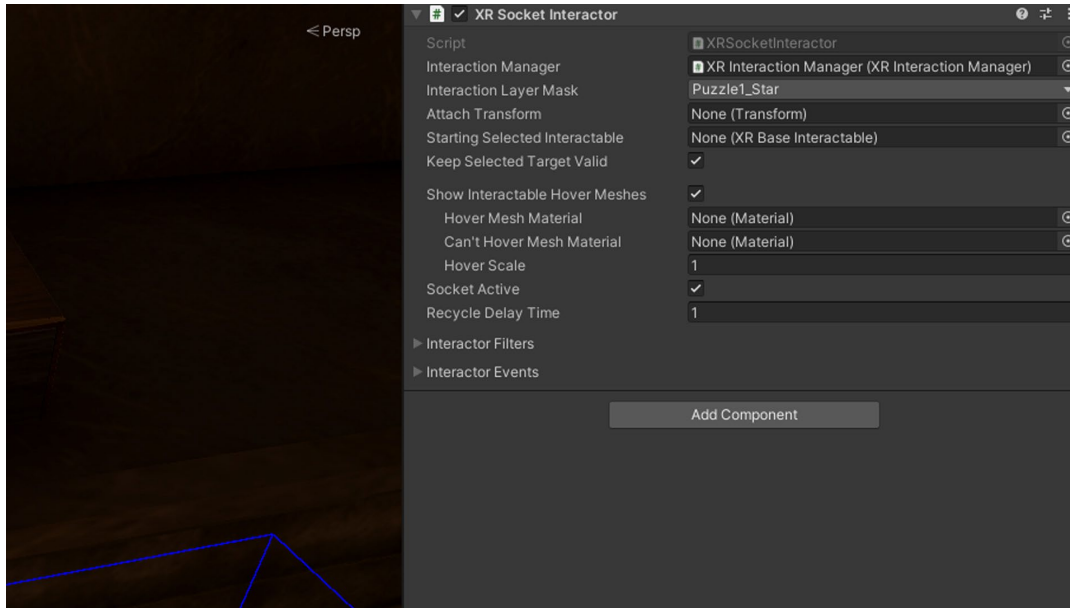
아래는 소켓 인터랙터가 적용된 퍼즐입니다.



적용된 소켓 인터랙터

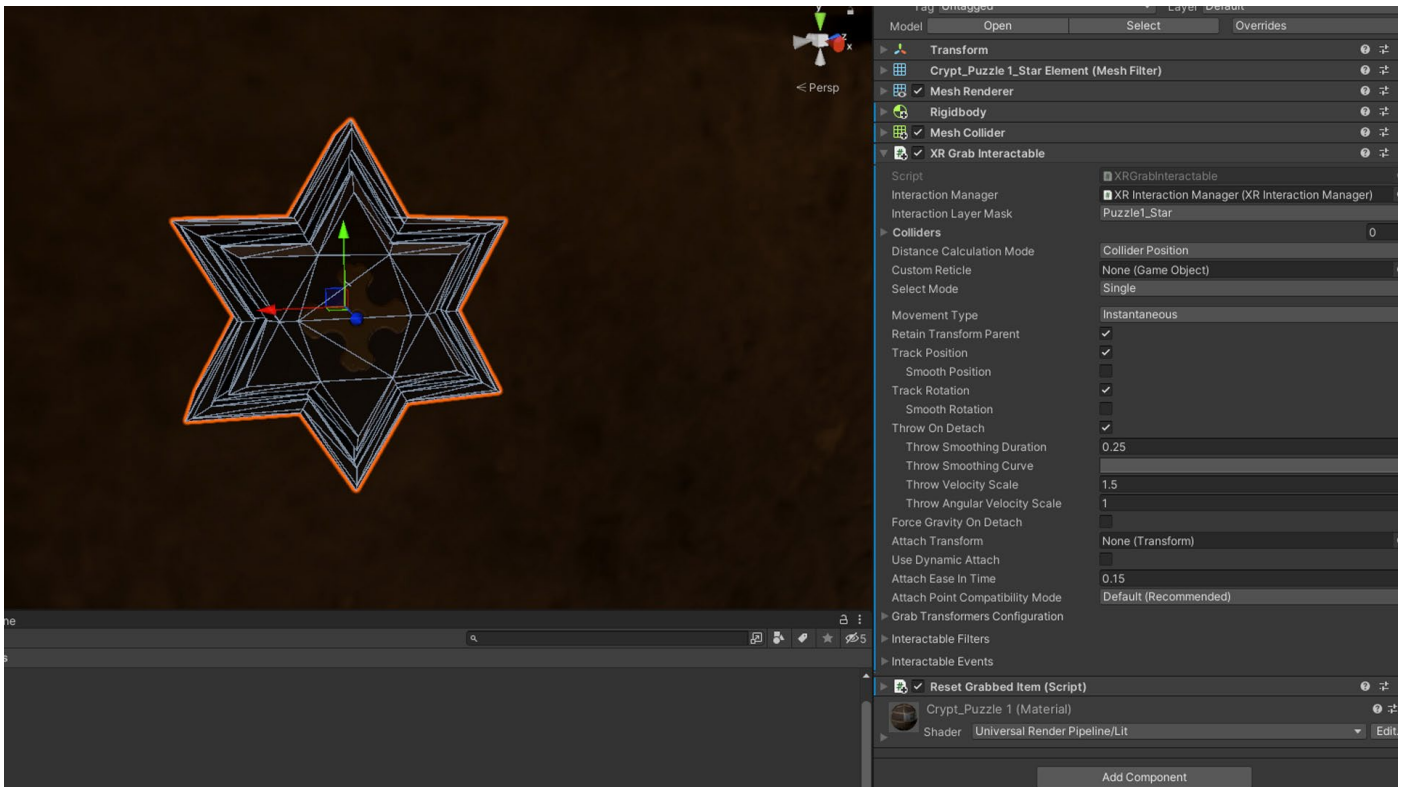


소켓 인터랙터에는 이를 받아들이는 레이어가 할당됩니다.



레이어 마스크 설정

퍼즐에 포함된 XR Grab Interactor 게임 오브젝트의 Interaction Mask도 소켓 인터랙터와 일치하도록 동일한 레이어로 설정되어 있습니다.



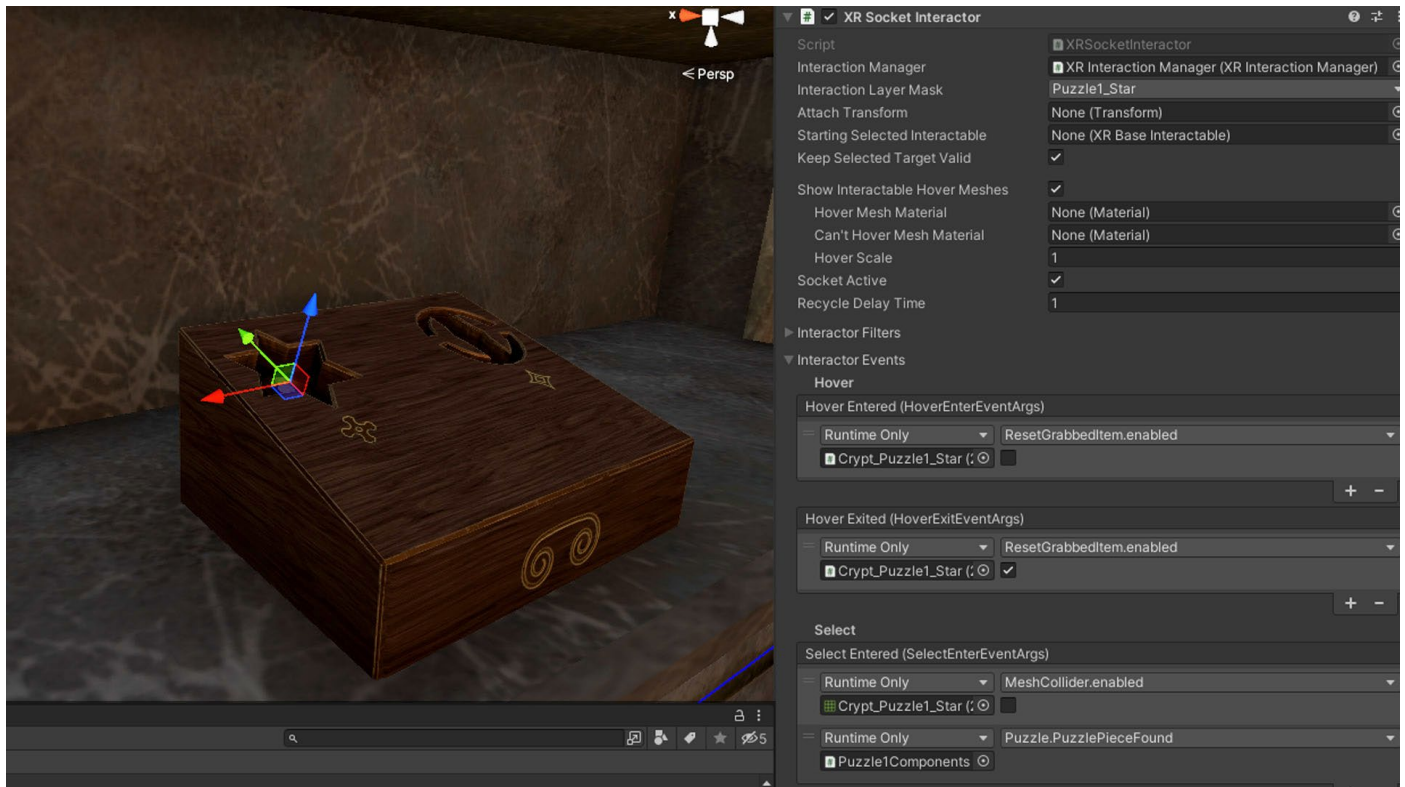
잡을 수 있는 이 오브젝트에는 소켓 인터랙터와 동일한 상호 작용 레이어가 설정되어 있습니다.



트리거에 가까이 가면 유효하거나 유효하지 않은 오브젝트를 나타내는 색상과 함께 메시지를 표시하도록 XR Socket Interactor를 추가로 커스터마이징할 수 있습니다.

이러한 방식으로 소켓 인터랙터를 사용하면 게임의 퍼즐 시스템을 빠르게 구축할 수 있고, 소켓 인터랙터 이벤트를 활용하여 기능을 커스터마이징할 수 있습니다.

공포 게임 예시에서는 퍼즐의 컴포넌트인 간단한 C# 스크립트를 작성할 수 있습니다. 이 스크립트는 퍼즐을 완성하는 데 필요한 단계 수를 int 단위로 인식합니다. 플레이어가 소켓에 올바른 오브젝트를 배치하면 소켓 인터랙터에 의해 **SelectEntered** 이벤트가 호출됩니다. 이 이벤트를 사용하면 조각을 발견했음을 퍼즐 스크립트에 알릴 수 있습니다.



상호 작용을 발생시키는 데 사용되는 소켓 인터랙터 관련 이벤트



puzzlePieceCounter가 증분된 다음 필요한 개수에 도달했는지 확인합니다. 필요한 조각이 발견되면 이벤트가 호출되고 퍼즐이 종료됩니다. 이 간단한 상호 작용을 제어하는 코드 스니펫은 다음과 같습니다.

```
1  using UnityEngine;
2  using UnityEngine.Events;
3
4  public class Puzzle : MonoBehaviour
5  {
6      //how many peices does this puzzle require
7      [SerializeField] private int requiredAmountOfPuzzlePieces = 0;
8
9      //keep track of found peices
10     private int puzzlePieceCounter = 0;
11
12     //Event for puzzle completion
13     public UnityEvent OnPuzzleCompleted;
14
15     //Track if the puzzle is completed
16     private bool isPuzzleCompleted = false;
17
18     //accessed by the socket interactors - 'SelectEntered' event
19
20     public void PuzzlePieceFound()
21     {
22         puzzlePieceCounter++;
23
24         CheckPuzzleCompletion();
25     }
26
27     //check to see if the user completed the puzzle!
28     private void CheckPuzzleCompletion()
29     {
30         if(isPuzzleCompleted)
31             return;
32
33         if(puzzlePieceCounter == requiredAmountOfPuzzlePieces)
34         {
35             OnPuzzleCompleted?.Invoke();
36             isPuzzleCompleted = true;
37             Debug.Log("Puzzle is completed");
38         }
39     }
40 }
```

퍼즐을 제어하는 코드

완료된 상호 작용을 지속적으로 확인하는 몇 가지 스크립트와 XRI Toolkit의 기능을 결합하여 게임의 퍼즐을 효율적으로 제작할 수 있습니다. XRI Toolkit, 잡기 및 소켓 인터랙터, 간단한 로직 레이어를 연동하여 플레이어에게 몰입감 있는 상호 작용을 구현할 수 있습니다. 플레이어는 못이 박힌 천장이 내려오는 동안 방을 탐색하여 퍼즐에 필요한 조각을 찾아 제자리에 배치한 후 그 보상인 열쇠를 손에 넣어야 합니다.

게임의 나머지 부분에는 XRI Toolkit에서 제공하는 컴포넌트를 사용하여 개발 속도를 높일 수 있습니다. 퍼즐을 다 만들었다면 이제 Unity 프로파일러를 통해 게임의 성능을 확인할 차례입니다.

테스트 및 반복 작업

테스트는 다음과 같은 여러 이유로 게임 개발 프로세스에서 매우 중요한 단계입니다.

- 버그 식별 및 수정
- 게임플레이 밸런스 조정
- 사용자 경험 및 사용성
- 성능 최적화
- 호환성 테스트
- 피드백
- 규정 준수
- 출시 준비

Unity에서는 Unity 프로파일러, Memory Profiler, Profiler Analyzer를 포함한 프로파일링 및 테스트 툴 제품군을 제공합니다. [이 Unity Discussions 게시물](#)에서는 [Unity 게임 프로파일링 완벽 가이드](#) 전자책을 비롯하여 프로파일링 툴과 관련된 고급 학습 리소스를 간략하게 설명합니다.



프로파일링 툴 외에도 자동화된 게임 테스트를 만들고, 관리하고, 실행할 수 있는 [Unity 테스트 프레임워크](#)도 있습니다. [Unity 베스트 프랙티스 허브](#)에서는 다음과 같은 문서를 비롯하여 고급 테스트, 디버깅, QA, 프로파일링과 관련된 다양한 정보를 확인할 수 있습니다.

- [프로파일링 및 디버깅용 툴](#)
- [Unity의 디버그 클래스로 QA 테스트 속도 향상 및 개선](#)
- [Roslyn 분석기로 게임 코드를 디버깅하는 방법](#)
- [Unity 프로젝트 테스트 및 품질 보증 관련 팁](#)

다음 동영상 튜토리얼에서 프로파일링 팁을 살펴보세요.



[시청하기](#)



[시청하기](#)



[시청하기](#)

Unity 에셋 스토어는 워크플로에 통합할 수 있는 타사 테스트 툴도 제공합니다. 이러한 방식을 도입하고 Unity 툴을 활용하면 게임의 품질과 성공 가능성을 크게 높일 수 있습니다.

XR Device Simulator

[XR Device Simulator](#)는 XR Interaction Toolkit 패키지의 샘플 애드온에 포함된 런타임 유틸리티입니다. 이 유틸리티를 사용하면 통상적으로 키보드와 마우스의 조합 또는 컨트롤러에서 키를 눌러서 이루어지는 사용자 입력을 시뮬레이션하여 씬에서 XR 헤드셋과 컨트롤러 기기를 구동할 수 있습니다.

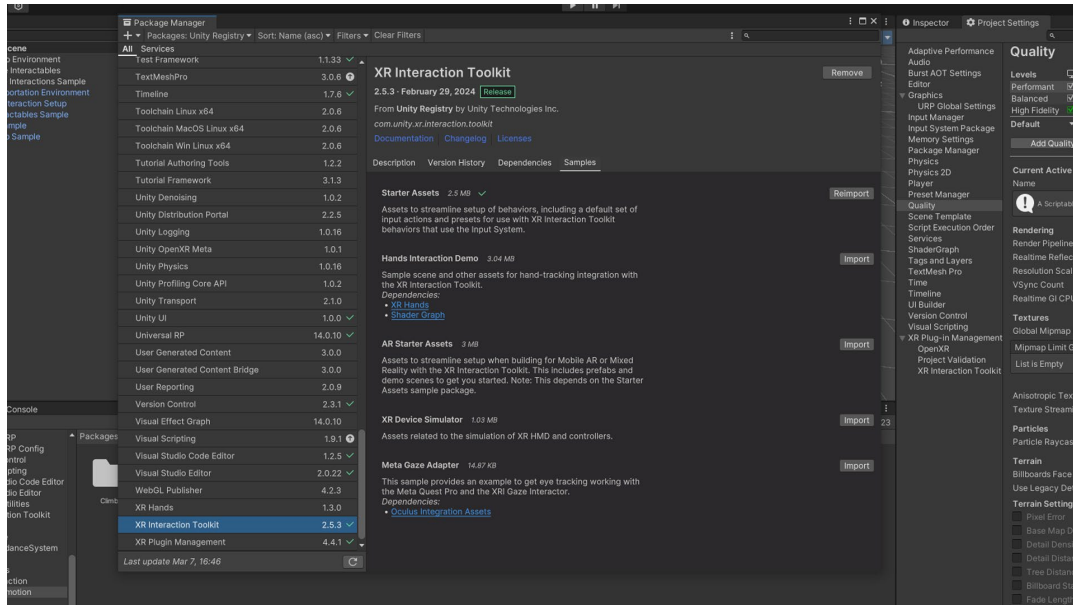
XR Device Simulator를 사용하면 실제 기기를 계속 착용할 필요 없이 Unity 에디터에서 바로 프로젝트를 효율적으로 테스트하고 반복 작업하여 시간을 단축할 수 있습니다. 다양한 환경 조건과 사용자 시나리오를 시뮬레이션하여 다양한 실제 조건과 사용자 입력을 처리하는 애플리케이션을 개발할 수 있습니다. 잠재적인 성능 문제에 대한 정보를 초기에 파악할 수 있으므로 개발 사이클 후반에 최적화와 하드웨어 테스트를 대규모로 수행할 필요성이 줄어듭니다.

앞서 사용한 샘플 씬에 시뮬레이터를 가져오는 방법을 살펴보겠습니다.



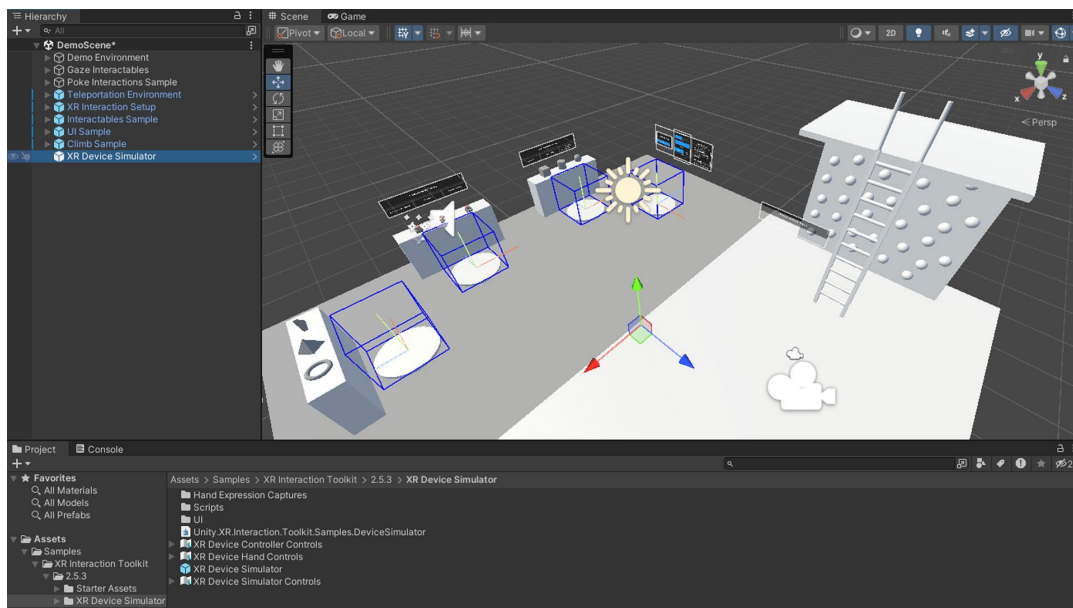
XD Device Simulator 설치

XD Device Simulator를 사용하려면 Starter Assets와 마찬가지로 임포트가 필요합니다. **Package Manager > XR Interaction Toolkit > Samples**로 이동합니다. 목록에서 해당 시뮬레이터를 선택하고 Import를 클릭합니다.



XR Device Simulator를 임포트할 준비가 완료된 Package Manager

XR Device Simulator로 씬을 손쉽게 테스트할 수 있습니다. 데모 씬에서 다음과 같이 프로젝트(Project) 창의 **XR Device Simulator** 프리팹을 찾아 씬에 놓으면 됩니다.



XR Device Simulator 프리팹을 찾아 데모 씬에 놓기



에디터에서 플레이 모드로 전환하면 시뮬레이터가 실행되며, 마우스와 키보드를 사용하여 이동할 수 있습니다. 무슨 기능이 어떤 키보드 키 또는 마우스 버튼과 연동되는지 안내하는 일련의 화면 지침이 표시됩니다.



플레이 모드에서 실행 중인 XR Device Simulator

이제 헤드셋을 착용한 것처럼 씬을 탐색하고 게임 내 요소와 상호 작용할 수 있습니다.

Unity 프로파일러

Unity 프로파일러는 CPU 사용, 메모리 할당, 렌더링, 네트워크 작동과 같은 게임 런타임 동작의 다양한 측면에 대한 자세한 정보를 제공합니다. 이 프로파일러는 읽기 쉬운 형식으로 실시간 데이터를 제공하므로, 개발자는 성능 병목 현상과 비효율성을 식별하고 분석할 수 있습니다.

이 프로파일러는 특히 복잡한 씬이나 저사양 하드웨어에서 원활한 게임플레이를 유지하는 데 리소스 관리가 중요한 개발 환경에서 유용합니다. 게임 성능 지표를 종합적으로 파악할 수 있으므로 최적화할 영역에 대해 정보에 입각한 결정을 내릴 수 있습니다.

프로파일러를 사용하여 XR 게임과 애플리케이션을 최적화하는 경우, 어떤 측면에 집중해야 하는지 알아야 합니다. 모니터링해야 할 주요 영역을 세분화하면 다음과 같습니다.

CPU 사용:

- **메인 스레드:** 메인 스레드에 과부하가 걸렸는지 확인합니다. 사용량이 많으면 프레임 속도가 떨어질 수 있습니다.
- **스크립트:** 스크립트 실행 시간을 살펴봅니다. 미흡하게 최적화된 스크립트는 많은 시간이 소요될 수 있습니다.
- **물리:** XR에서는 물리 계산이 많이 이루어질 수 있으므로, 그에 따라 CPU가 받는 영향을 모니터링합니다.



GPU 사용량:

- **렌더링:** XR을 사용하려면 높고 안정적인 프레임 속도가 필요합니다. GPU가 병목 현상을 일으키지 않도록 렌더링 시간을 모니터링하세요.
- **드로우 콜:** 드로우 콜이 많으면 GPU 속도가 느려질 수 있습니다. 더 나은 성능을 발휘할 수 있도록 드로우 콜 감소를 목표로 하세요.

메모리 사용량:

- **총 할당 메모리:** 애플리케이션의 메모리 사용량을 주시합니다.
- **가비지 컬렉션:** 가비지 컬렉션이 잦으면 성능 스파이크가 발생할 수 있으므로 할당량을 줄일 방안을 찾아봅니다.

프레임 속도:

- **안정성:** XR 애플리케이션은 일관된 프레임 속도를 유지해야 합니다. 감소 또는 비일관적인 부분이 있는지 찾아보세요.
- **타겟 프레임 속도:** 애플리케이션이 해당 XR 플랫폼의 타겟 프레임 속도를 충족하는지 확인합니다.

네트워크 성능(멀티플레이어 XR의 경우):

- **지연:** 지연이 길면 멀티플레이어 XR에서 사용자 경험이 저하될 수 있습니다.
- **데이터 전송:** 전송 중인 데이터의 양을 모니터링하고 최적화하여 지연을 줄입니다.

오디오 프로파일링:

- **오디오 소스 관리:** 오디오 소스를 여러 개 사용하면 부담이 될 수 있습니다. 효율적으로 사용하세요.
- **DSP 부하:** 특히 고급 오디오 효과를 사용하는 경우 디지털 신호 처리 부하를 확인합니다.

렌더링 파이프라인:

- **배치 및 SetPass 호출:** 배치와 SetPass 호출을 줄이도록 최적화하여 GPU 부하를 감소시킵니다.
- **VR 관련 지표:** 가능한 경우 지연 및 동공 간 거리 조정값과 같은 VR 관련 지표를 살펴봅니다.

UI 프로파일링:

- **캔버스 렌더링:** 복잡한 UI는 큰 부담을 줄 수 있습니다. 캔버스 요소와 해당 업데이트를 최적화하세요.



커스텀 프로파일링 마커:

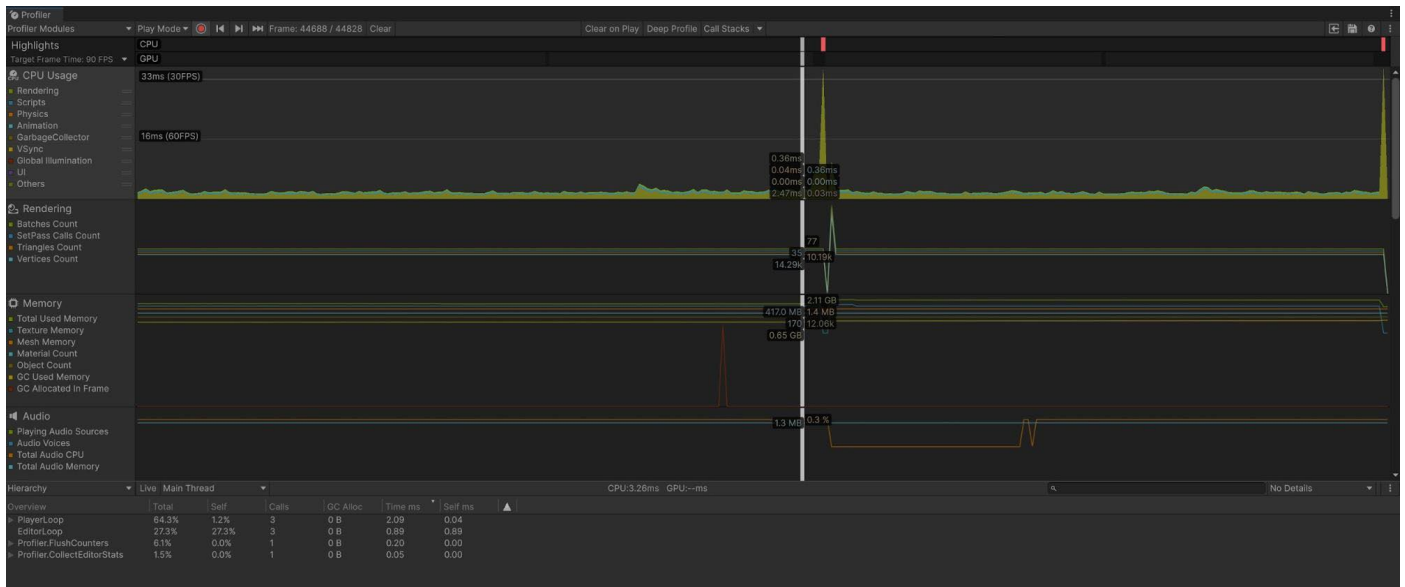
- **사용자 정의 마커:** 코드에 커스텀 마커를 사용하여 애플리케이션에 중요한 특정 메서드나 프로세스를 모니터링합니다.

서멀 스로틀링(특히 모바일 XR의 경우):

- **기기 온도:** 고성능이 요구되는 경우 모바일 기기가 과열되어 서멀 스로틀링(thermal throttling)이 발생할 수 있습니다.

이러한 영역을 면밀히 분석하여 성능 병목 현상을 파악하고 그에 따라 XR 게임 또는 애플리케이션을 최적화할 수 있습니다. 시각적 충실도, 반응성, 원활한 성능 간의 균형을 유지하는 것이 목표라는 점을 잊지 마세요.

Window > Analysis > Profiler를 통해 프로파일러 창을 열 수 있고, **Ctrl+7** 키를 눌러도 됩니다. 플레이를 누르면 프로파일러가 시작되고 프로젝트에서 캡처한 데이터로 뷰가 채워집니다.



프로파일러에서 프로젝트 프로파일링

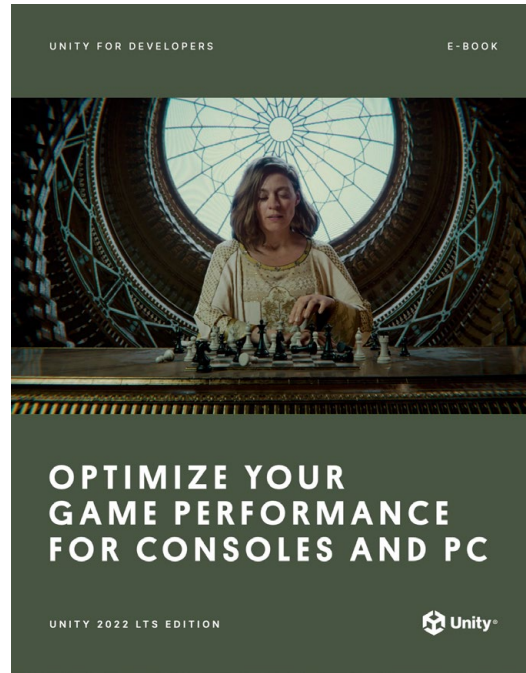
언제든지 일시 중지하여 게임 내에서 프레임 속도를 떨어뜨리는 영역을 확인할 수 있습니다. 왼쪽 메뉴를 사용해 **Rendering** 또는 **CPU Usage**와 같은 특정 레이어를 선택하여 자세한 정보를 확인할 수 있습니다. 각 항목은 색상으로 구분되어 게임 성능 문제가 발생하는 위치를 빠르게 파악할 수 있습니다.



앞서 언급한 [프로파일링 전자책](#) 외에도 다음과 같은 2가지의 고급 최적화 가이드에서 다양한 최적화 팁을 확인할 수 있습니다.



[다운로드](#)



[다운로드](#)

빌드 및 배포

월드를 구성하고 상호 작용을 빌드, 테스트 및 최적화하는 데 많은 시간을 들였습니다. 이제 게임을 세상에 선보일 시간이 되었습니다. 게임을 기기에 바로 빌드하거나 스토어에 제출할 수 있습니다. 스토어마다 앱에 대한 가이드라인과 요구 사항이 다르므로 앱 제출에 관한 기술 자료를 확인해야 합니다.



VR 앱 빌드

혼합 현실 및 공간 컴퓨팅 애플리케이션



MR(혼합 현실) 애플리케이션은 하이브리드 환경에서 현실 세계와 가상 월드를 연동하며, 이러한 요소가 적절히 혼합되어 사용자가 매끄럽게 느낄 수 있어야 합니다.

MR 개발에서는 다음과 같은 VR 및 AR 월드 구축 요소를 함께 사용합니다.

사용자 상호 작용 및 인터페이스 디자인

VR과 AR의 UX 및 UI 디자인에는 MR 앱에도 적용 가능한 일반 원칙이 있습니다. 다양한 기기에서 자연스럽게 느껴지는 직관적인 인터페이스와 상호 작용 모델을 디자인해야 합니다. 여기에는 사용자가 직관적으로 가상 월드와 상호 작용할 수 있도록 손 트래킹, 응시 컨트롤, 음성 커맨드를 활용하는 것이 포함됩니다.



공간 인식 및 물리

가상 월드의 물리를 이해하고 통합하는 것은 VR, AR, XR에서 매우 중요합니다. 여기에는 가상 오브젝트 간의 사실적인 상호 작용을 만들고, AR의 경우에는 이러한 상호 작용을 현실 세계와 혼합하는 작업이 포함됩니다. 개발자는 공간 인식을 고려하여 중력, 충돌 및 기타 물리적 프로퍼티에 대한 사용자의 기대와 일치하는 방식으로 가상 오브젝트가 동작하도록 해야 합니다.

크로스 플랫폼 개발 전략

스탠드얼론 헤드셋부터 모바일 기반 시스템까지 VR과 AR을 지원하는 기기는 다양하므로 크로스 플랫폼 호환성을 염두에 두고 개발해야 합니다. Unity는 개발자가 최소한의 조정만으로 여러 플랫폼에 배포 가능한 애플리케이션을 제작할 수 있는 툴과 워크플로를 제공합니다. 여기에는 다양한 기기의 기능과 한계에 맞게 사용자 인터페이스, 상호 작용 모델, 성능 최적화를 조정하는 작업이 포함됩니다.

환경 디자인과 몰입

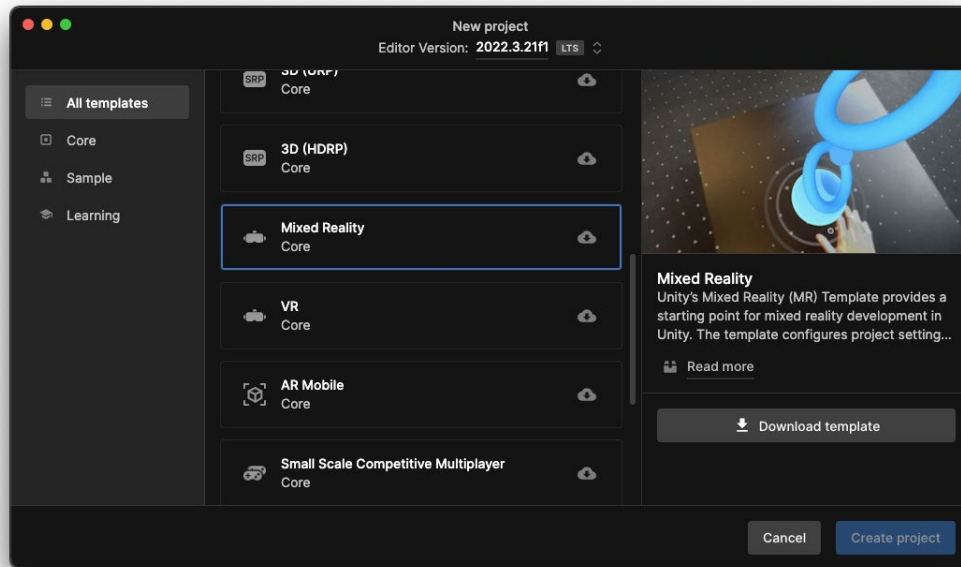
MR의 환경 디자인에서는 VR과 AR의 몰입 기법을 함께 사용합니다. 비주얼 측면뿐만 아니라 입체 음향, 햅틱 피드백 및 기타 감각적 입력을 통합하여 설득력 있고 흥미로운 월드를 만드는 작업도 여기에 포함됩니다. VR 내에 완전한 가상 월드를 구현하는 경우에도, AR의 물리적 세계에 디지털 오버레이를 적용하는 경우에도 사용자에게 본인이 가상 환경 속에 있다는 느낌을 주도록 해야 합니다.





MR 템플릿

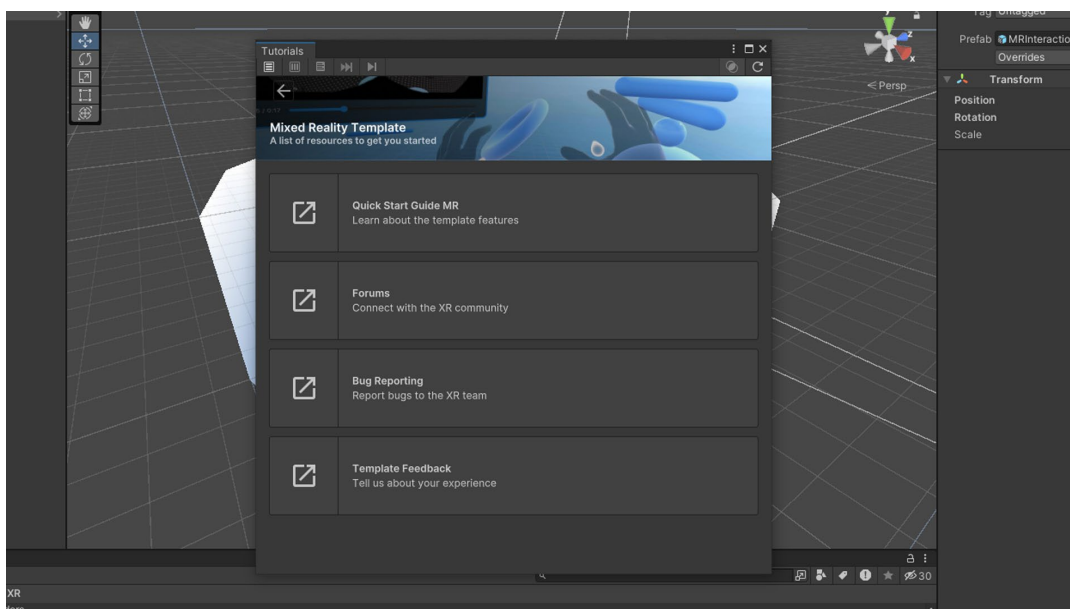
Unity의 MR 템플릿은 Unity를 통한 혼합 현실 개발의 시작점 역할을 할 수 있도록 설계되었습니다.



Unity Hub의 Mixed Reality 템플릿

템플릿을 다운로드하고 새 프로젝트를 만듭니다. Unity 2022 LTS 이상을 사용하는 것이 좋습니다.

프로젝트가 시작 화면과 함께 열립니다. 튜토리얼, 전용 포럼, 버그 신고 등에 대한 링크를 제공하는 창도 있습니다.



튜토리얼 콘텐츠 창

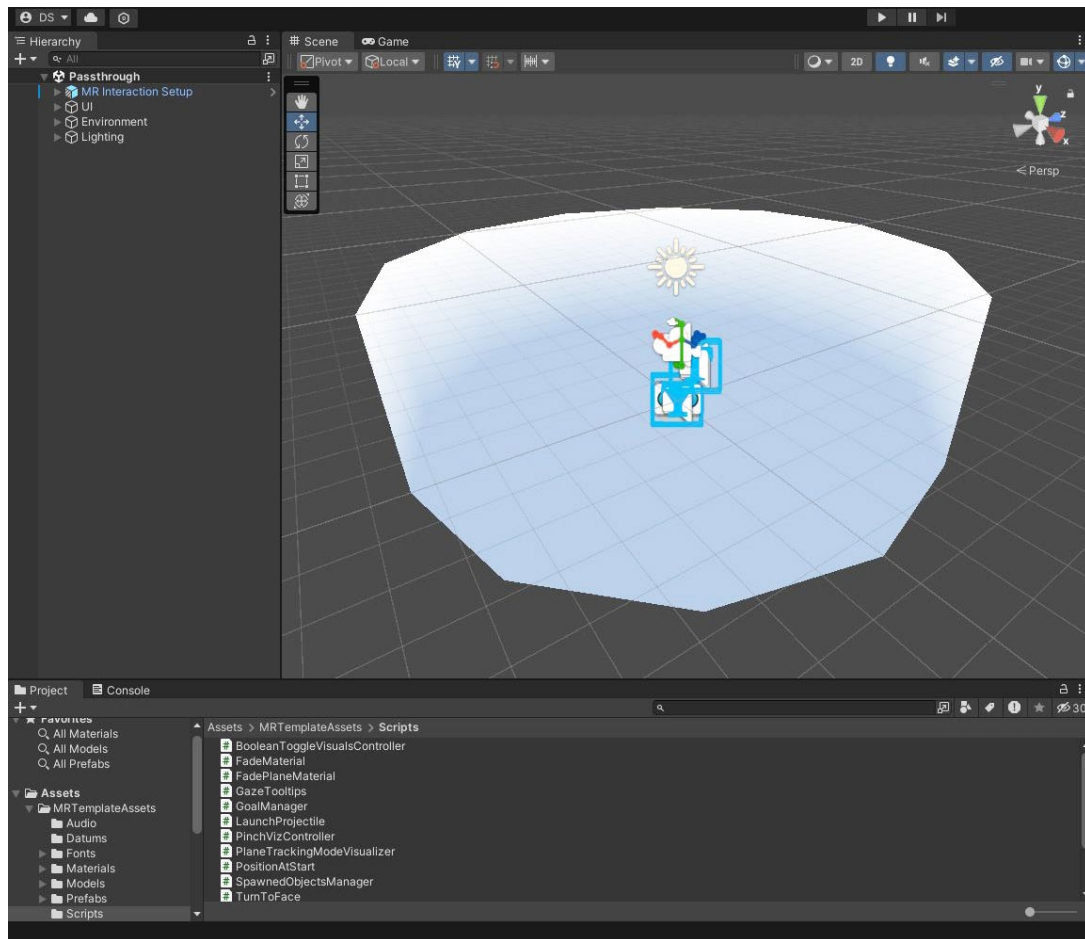


빠른 시작 목록에서는 MR 템플릿에 대한 [기술 자료](#) 링크도 찾을 수 있습니다.

MR 템플릿에서는 OpenXR, AR Foundation, XR Interaction Toolkit 툴셋을 활용합니다.

MR Interaction Setup 프리팹

MR Interaction Setup 프리팹은 XR 경험에서 XR 카메라와 원점을 설정하기 위한 기본 게임 오브젝트입니다. 컨트롤러 및 손과 같은 다양한 입력 시스템을 적용할 수 있고 누르기, 직접 접촉, 광선과 같은 다양한 인터랙터를 포함합니다.



MR 템플릿 계층 구조의 MR Interaction Setup 프리팹

MR Interaction Setup 프리팹에는 Input Action Manager, XR Interaction Manager 및 XR Origin (XR Rig) 등의 다양한 컴포넌트가 있습니다. 이 가이드의 이전 챕터를 읽었다면 해당 컴포넌트에 익숙할 것입니다. MR 템플릿 고유의 컴포넌트 몇 가지를 살펴보겠습니다.

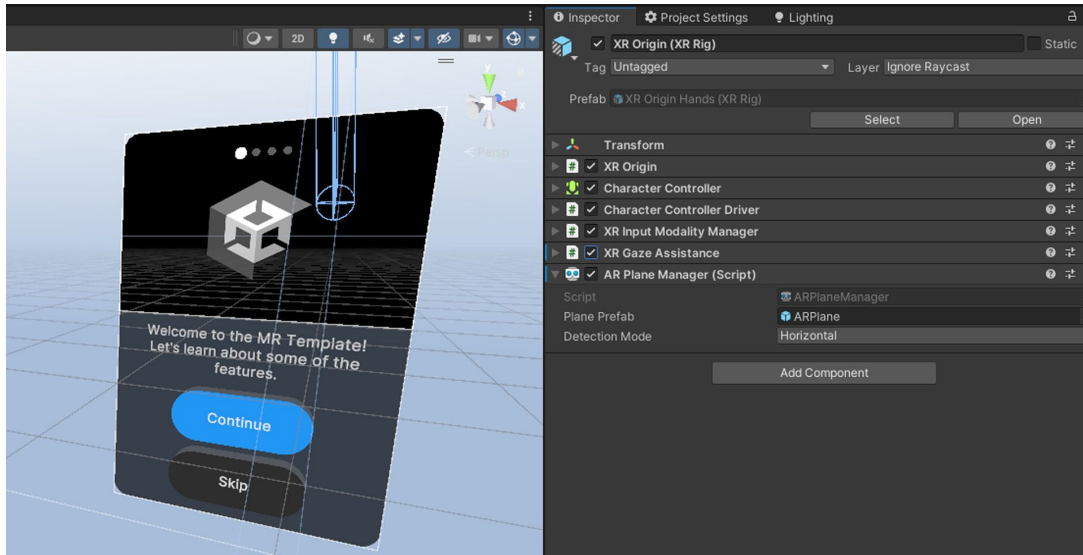
AR Session

AR Session 컴포넌트는 AR 세션의 라이프사이클과 설정을 관리합니다. 한 번에 하나의 세션만 활성화할 수 있습니다.



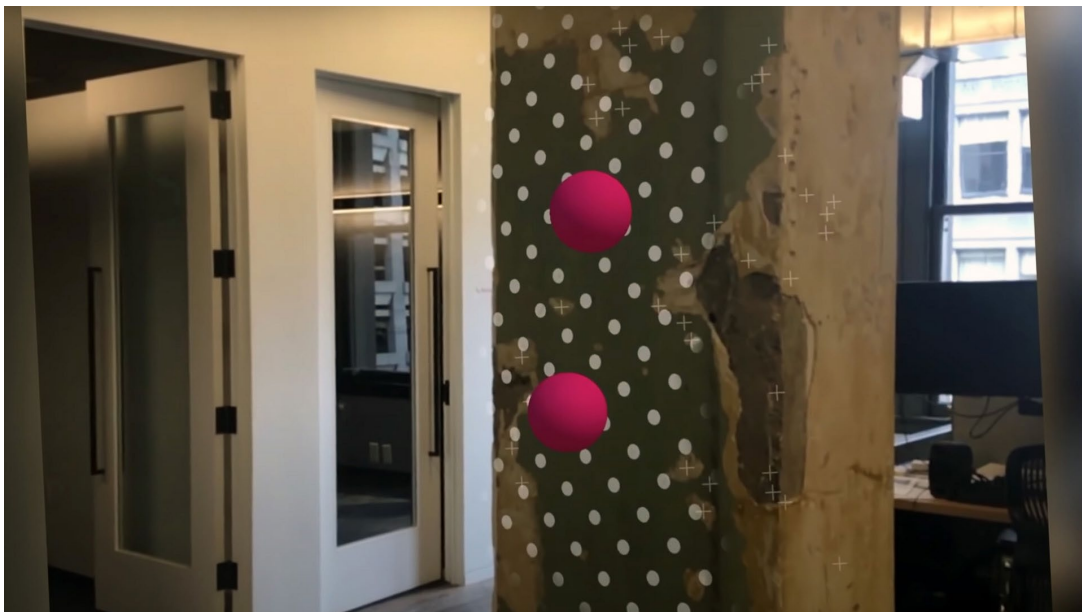
AR Plane Manager

AR Plane Manager는 플레이어 주변에서 검출된 각 평면에 대한 게임 오브젝트를 생성합니다. 기기의 카메라와 AR Foundation을 활용하여 검출된 평면을 관리합니다. 이 컴포넌트는 XR Origin 컴포넌트와 동일한 게임 오브젝트에 적용됩니다.

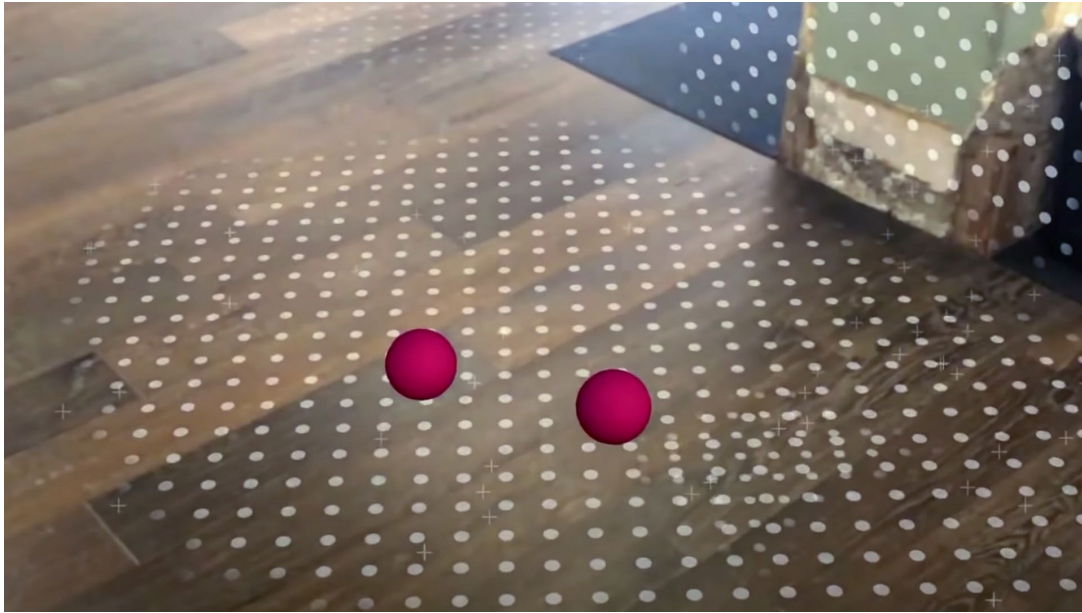


XR Origin에 적용된 Plane Manager 컴포넌트

평면은 포즈, 치수, 경계점으로 표현되는 평평한 표면입니다. 평면으로 검출할 수 있는 부분의 예로는 수평 테이블, 바닥, 벽, 조리대 등이 있습니다. 수평, 수직 또는 둘 다 설정 가능한 검출 모드를 사용할 수 있습니다. 이 관리자는 커스텀 프리팹을 사용하여 검출된 평면을 시각화하는 옵션도 제공합니다.



수직 표면의 AR 평면



바닥과 벽에서 검출된 평면

패스스루

패스스루 기술을 사용하면 VR 헤드셋에서 현실 세계와 가상 오브젝트를 동시에 볼 수 있으며, 기기의 카메라를 사용하여 주변의 사물을 캡처할 수 있습니다. MR 템플릿에서 패스스루는 소프트웨어와 하드웨어를 결합하여 디지털 콘텐츠를 현실 세계와 즉시 혼합하는 방식으로 이루어집니다. 이 기능은 주로 AR 경험에 사용되며, 이를 통해 사용자는 가상 씬에 있는 동안 실제 환경을 보면서 상호 작용할 수 있습니다. Unity에서 패스스루는 다음과 같은 몇 가지 요소를 필요로 합니다.

- **XR 플러그인 프레임워크:** 이 시스템은 카메라 피드 및 기타 하드웨어 리소스에 액세스하는 데 필요한 인터페이스와 컨트롤을 제공하므로 패스스루를 사용하는 데 필수적입니다.
- **하드웨어 지원:** 타겟 기기에는 사용자 환경에서 실시간으로 동영상을 캡처할 수 있는 카메라 등, 패스스루 지원에 필요한 하드웨어가 탑재되어 있어야 합니다. 이 하드웨어의 품질과 기능은 패스스루 경험의 효과와 현실감에 큰 영향을 미칠 수 있습니다.
- **카메라 피드 액세스:** 패스스루 기능의 핵심은 기기의 카메라에서 라이브 피드에 액세스하는 기능입니다. 이 피드는 사용자의 시야각(FOV) 내에서 실시간으로 렌더링되어 가상 환경 내에 현실 세계를 보여 주는 창을 구현합니다.
- **소프트웨어 컴포넌트:** ARKit 및 ARCore 기기용 AR Foundation 패키지 또는 Oculus 기기용 Oculus Integration 패키지 등, Unity 프로젝트의 특정 컴포넌트에서 패스스루 구현에 필요한 툴과 API를 제공합니다. 이들 컴포넌트는 카메라 피드 렌더링, 사용자 입력 처리, 디지털 콘텐츠를 물리적 환경과 연동하는 데 수반되는 복잡도를 관리합니다.



MR 템플릿 테스트

MR 템플릿 내의 프로젝트 설정은 사전 설정되어 있으며, XR Plug-in Management 패키지는 Meta Quest 기능 그룹과 함께 OpenXR을 사용하도록 설정되어 있습니다. 두 가지 상호 작용 프로파일, 즉 **Meta Quest Touch Pro 컨트롤러 프로파일**과 **Oculus Touch 컨트롤러 프로파일**이 추가되었습니다.

Unity는 Meta Quest에서 템플릿을 테스트할 예정입니다. Meta Quest는 스탠드얼론 VR/MR 앱용 Android 빌드 타겟과 Meta Quest Link를 통한 PCVR 앱용 Windows 빌드 타겟을 지원하는 Android 기반 플랫폼입니다. 이 템플릿의 경우 혼합 현실 기능은 현재 스탠드얼론/Android에서만 지원되므로 빌드 타겟이 Android로 설정되어 있습니다. 오류 메시지가 표시되면 프로젝트의 플랫폼이 Android로 선택되었는지 확인하세요. 기본 Windows, Mac, Linux 플랫폼에서는 오류가 발생할 수 있습니다.

평면 검출 기능을 사용하려면 사용자가 플레이하는 공간에 맞게 헤드셋을 설정해야 합니다. Meta Quest를 사용하는 경우 이 작업을 하려면 설정 안에 물리적 공간을 구성해야 합니다. 그러면 헤드셋은 벽, 바닥, 천장, 기타 아이템이 어디 있는지 알 수 있습니다.

프로젝트를 빌드하고 실행하면 샘플 프로젝트의 기능에 대한 간단한 안내를 제공하는 UI가 표시됩니다. 다른 기능과 함께, 수평면을 탭하여 3D 오브젝트를 배치하고, 이동하고, 손으로 활성화한 UI를 통해 오브젝트를 제거하는 등의 기능을 사용할 수 있습니다.

Unity의 혼합 현실 프로젝트 템플릿에는 패스스루 기술을 위한 까다로운 수많은 설정 작업이 간소화되어 있습니다. 따라서 개발자는 MR 프로젝트 제작용으로 사전에 조정된 설정, 스크립트, 리소스가 포함된 패키지를 사용할 수 있습니다. 디지털과 현실 세계를 혼합하는 애플리케이션을 더 쉽게 시작할 수 있도록 설계된 템플릿입니다. 그렇다 해도, 프로젝트에 필요한 사항과 타겟 기기에 따라 모든 것이 제대로 작동하려면 일부 설정을 미세 조정하거나, 요소를 더 추가하거나, 직접 코드를 작성해야 할 수 있습니다.

Apple Vision Pro를 사용한 공간 컴퓨팅



visionOS에서 윈도우형 앱으로 실행되는 Unity 샘플



Apple Vision Pro는 Apple 최초의 공간 컴퓨터입니다. 물리적 월드와 디지털 월드를 통합하여 사용자가 애플리케이션에서 두 월드와 동시에 상호 작용하도록 지원합니다. 이 기기는 고급 센서, 디스플레이 기술, 소프트웨어 알고리즘을 활용하여 매우 사실적인 VR 및 AR 경험을 제공합니다.

Apple Vision Pro의 공간 컴퓨팅은 사용자의 환경과 움직임을 인식하고 그에 반응함으로써 게임과 엔터테인먼트부터 교육과 전문 협업에 이르기까지 광범위한 애플리케이션을 지원합니다.



Apple Vision Pro에서 몰입형 경험으로 실행 중인 WHAT THE GOLF?(개발: Triband)

Unity에서 visionOS 시작하기

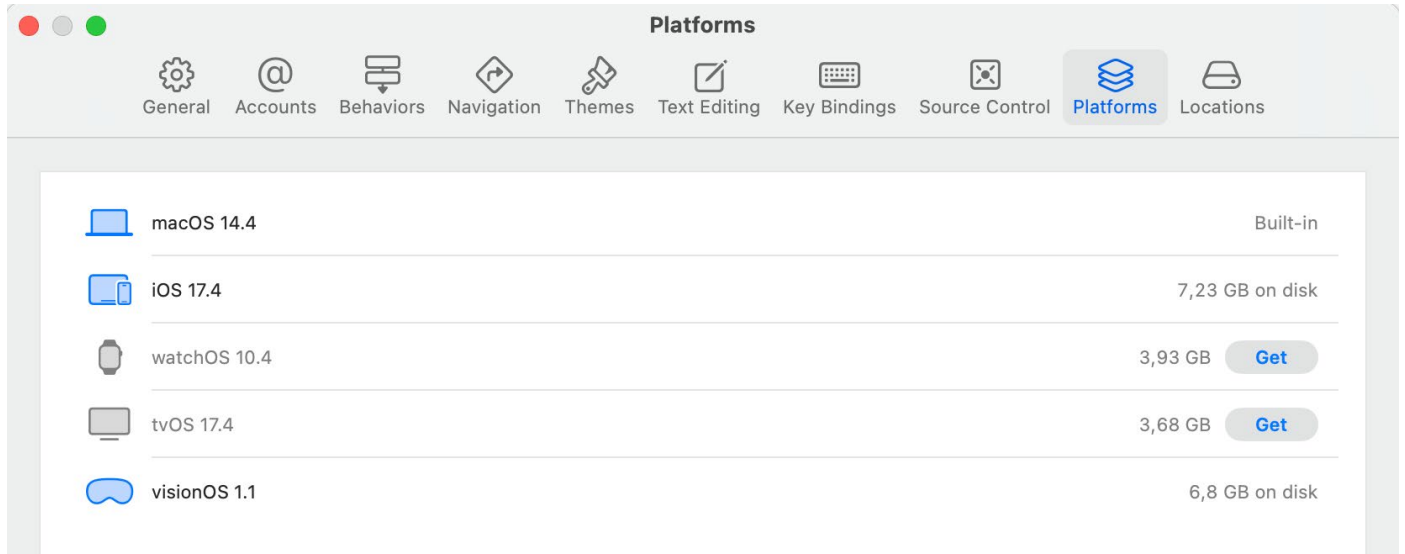
이 섹션에서는 Unity로 제작할 수 있는 다양한 유형의 Apple Vision Pro 경험을 살펴봅니다.

- **몰입형(혼합 현실):** 유니티에서 새롭게 개발한 PolySpatial 기술로 디지털 콘텐츠와 실제 세계를 혼합하고 Shared Space에서 다른 앱과 함께 실행할 수 있는 경험
- **완전 몰입형(가상 현실):** 기존 가상 현실 게임을 포팅하거나, 플레이어의 주변 환경을 다른 환경으로 대체하는 새로운 완전한 몰입형 경험
- **윈도우형:** 2D 창에서 실행하며, 사용자가 Shared Space에서 크기나 위치를 조절할 수 있는 콘텐츠



visionOS 앱을 제작하려면 다음과 같은 하드웨어가 필요합니다.

- visionOS 개발을 위해 Apple Silicon이 탑재된 Mac, Xcode 15.2 이상 버전 및 visionOS 시뮬레이터 필요
- visionOS 시뮬레이터 사용을 위해 Apple Silicon(M1 이상) Mac 필요
- Xcode 설정/플랫폼에 visionOS 시뮬레이터 설치 필요
- visionOS 하드웨어가 없는 경우 XCode의 빌트인 시뮬레이터 사용

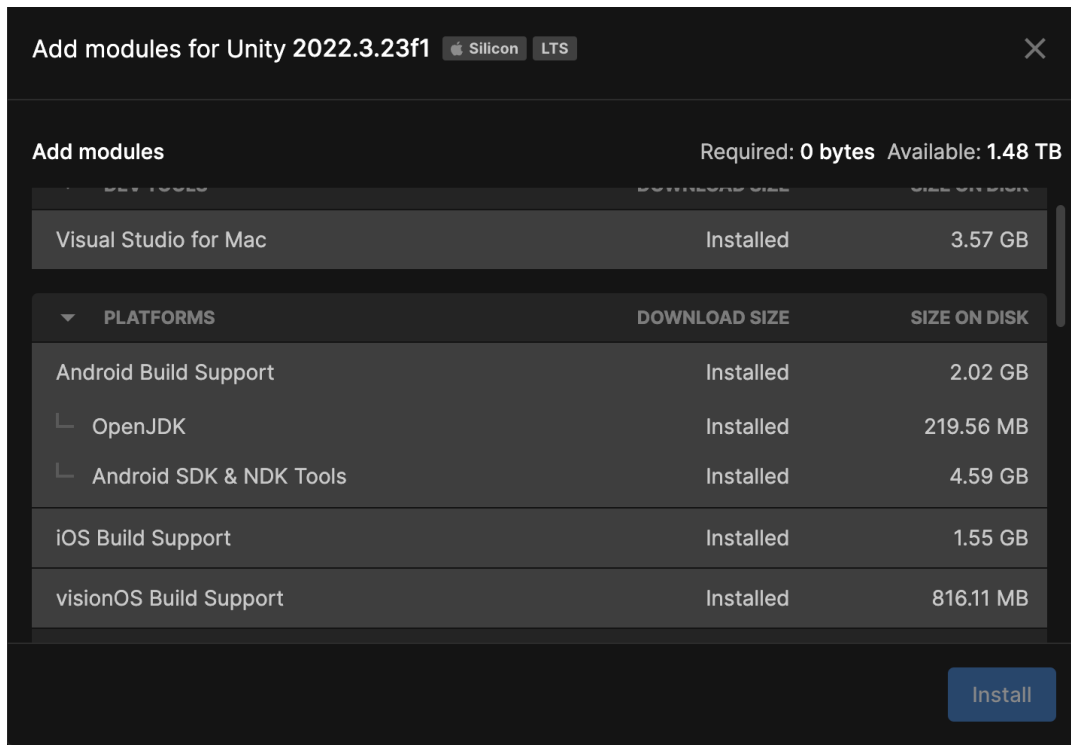


Xcode에서 지원하는 시뮬레이터

visionOS 앱 개발을 시작하려면 Unity Hub에서 최신 Unity 2022 LTS 버전을 다운로드하여 설치합니다.

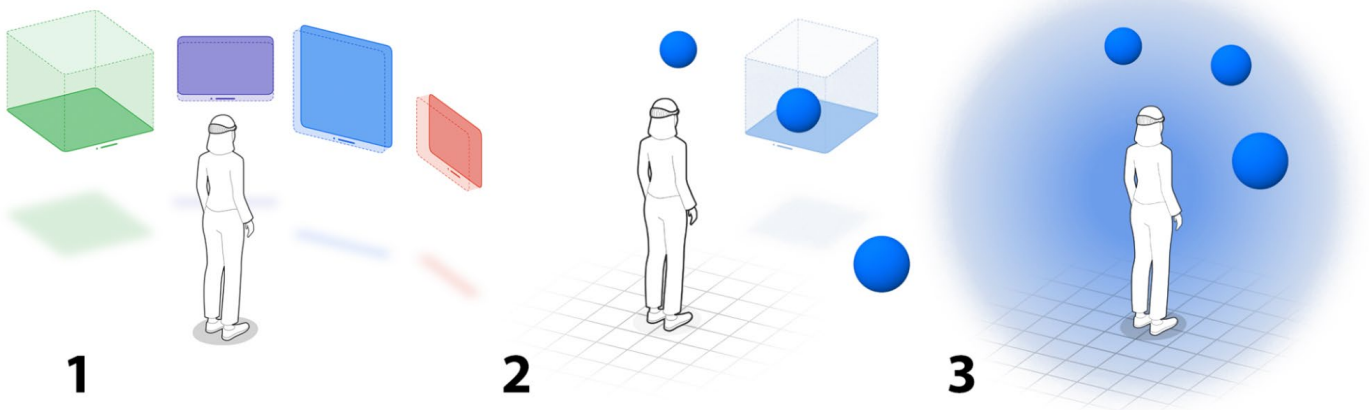


모듈 목록에서 **visionOS Build Support** 및 **iOS Build Support** 모듈을 선택하고 설치합니다.



Vision Pro에서 앱이 표시되고 액세스되는 방식을 보여 주는 그림(이미지 제공: Apple)visionOS 및 Unity 소개

visionOS 및 Unity 소개



Vision Pro에서 앱이 표시되고 액세스되는 방식을 보여 주는 그림(이미지 제공: Apple)



visionOS에서 앱은 iPhone 또는 iPad의 아이콘 그리드와 유사한 홈 보기의 아이콘에서 실행됩니다. 위 이미지에서 Vision Pro에서 앱이 표시되는 방식을 확인할 수 있습니다.

1. 창 또는 볼륨에서 콘텐츠를 실행하는 앱은 Shared Space(1)에서 실행되어 앱이 나란히 존재하게 되며, Mac 데스크톱의 다양한 앱과 마찬가지로 앱을 이동하고 재정렬할 수 있습니다.
2. 앱에서 볼륨(2)을 사용하는 경우 **Volume Camera** 내부에 있는 콘텐츠만 표시되며, 이때 카메라의 모드는 **Bounded**로 설정됨에 따라 Shared Space에 앱 콘텐츠가 표시됩니다.
3. 카메라의 모드를 **Unbounded**로 설정하면 Shared Space에 있는 앱은 숨겨지고 열려 있는 앱에만 포커스가 맞춰지므로 앱이 몰입형 공간(3)에서 제공되는 전체 공간을 차지할 수도 있습니다.

Bounded 모드와 Unbounded 모드는 모두 카메라 패스스루와 물리적 공간을 통해 사용자 앞의 현실 세계와 가상의 3D 오브젝트를 혼합합니다.



Bounded 모드가 적용된 Volume Camera로 앱이 실행되고 있는 Shared Space



카메라 피드와 물리적 공간이 애플리케이션 콘텐츠에 사용되지 않는 기타 두 가지 모드는 플레이어 또는 사용자를 완전한 가상 월드로 안내하는 **Fully Immersive VR**과 Shared Space에서 공존할 수 있는 2D 또는 3D 경험을 눈앞의 플로팅 창에 렌더링하는 **Windowed Apps**입니다. iPhone 및 iPad 앱은 visionOS에서 윈도우형 앱으로 실행됩니다.



기기에서 사용 가능한 모든 앱을 실행할 수 있는 visionOS 홈 보기

다음은 Apple Vision Pro에서 사용되는 다양한 XR 경험의 주요 기능과 요구 사항을 요약한 표입니다.

앱 유형	Shared Space	AR	렌더링 기술	Play To Device	Vision Pro 플러그인 필요
완전 몰입형 VR	지원 안 함	지원 안 함	Metal	지원 안 함	지원
바운디드 모드의 MR	지원	지원	RealityKit	지원	지원
언바운디드 모드의 MR	지원 안 함	지원	RealityKit	지원	지원
윈도우형 앱	지원	지원 안 함	Metal	지원 안 함	지원 안 함

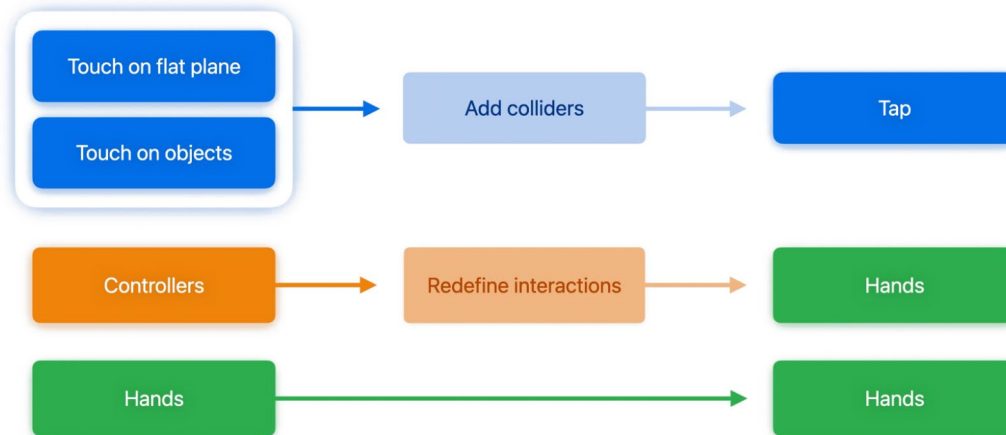
필요한 경우 Vision Pro 플러그인에서 visionOS 패키지를 자동으로 설치



Apple Vision Pro용 Unity 프로젝트에서 상호 작용이 어떻게 이루어지는지 살펴보겠습니다.

상호 작용

visionOS 앱에서 사용자는 눈과 손을 사용하여 콘텐츠를 보고 상호 작용하며, 손가락을 함께 탭하여 항목을 선택합니다. 사용자 권한에 따라 풀 핸드 트래킹 데이터는 물론 헤드 포즈 데이터도 앱에서 활용할 수 있습니다. 커스텀 제스처로 사실적인 느낌의 상호 작용을 만들 수 있으며, 일반적인 상호 작용 방식으로 탭 제스처가 사용됩니다.



기존 애플리케이션 상호 작용을 visionOS로 전환 출처: Apple Inc.

오브젝트에서 관련 이벤트를 수신할 수 있도록 오브젝트에 입력 콜라이더를 연결해야 합니다. 멀리서 보고 탭하여 오브젝트를 선택하거나, 손을 뻗어 직접 손가락으로 오브젝트를 터치할 수 있습니다.

Unity에서는 탭을 WorldTouch 이벤트로 사용할 수 있습니다. 이 이벤트는 2D 탭 이벤트와 유사하나 완전한 3D 위치를 갖습니다.

iPhone 등에서 이미 터치로 작업하고 있다면 적절한 입력 콜라이더를 추가하고 이어서 탭을 기본 입력 메커니즘으로 사용할 수 있습니다.

VR 컨트롤러를 사용하는 경우 상호 작용의 복잡도에 따라 탭 또는 손을 기반으로 한 입력의 관점에서 상호 작용을 재정의해야 합니다.

3D 터치 및 TouchSpace

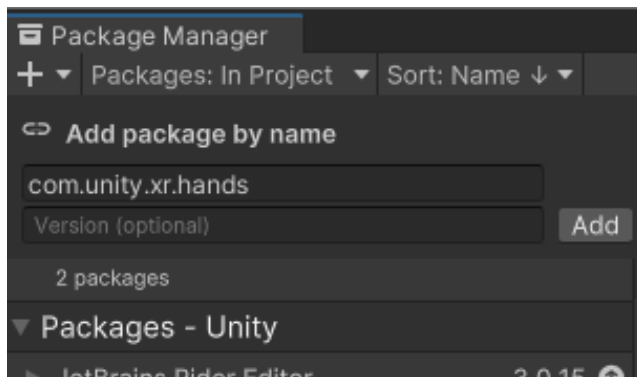
사용자가 입력 콜라이더가 적용된 오브젝트를 보고 ‘핀치’ 제스처를 수행하는 경우 바운디드 및 언바운디드 볼륨 모두에서 3D 터치 입력이 제공됩니다. SpatialPointerDevice 입력 기기에서 해당 정보를 개발자에게 제공합니다. 사용자가 핀치 제스처를 유지하고 있으면 드래그가 시작되고 애플리케이션에는 원래 시작점을 기준으로 한 ‘이동’ 업데이트가 제공됩니다.



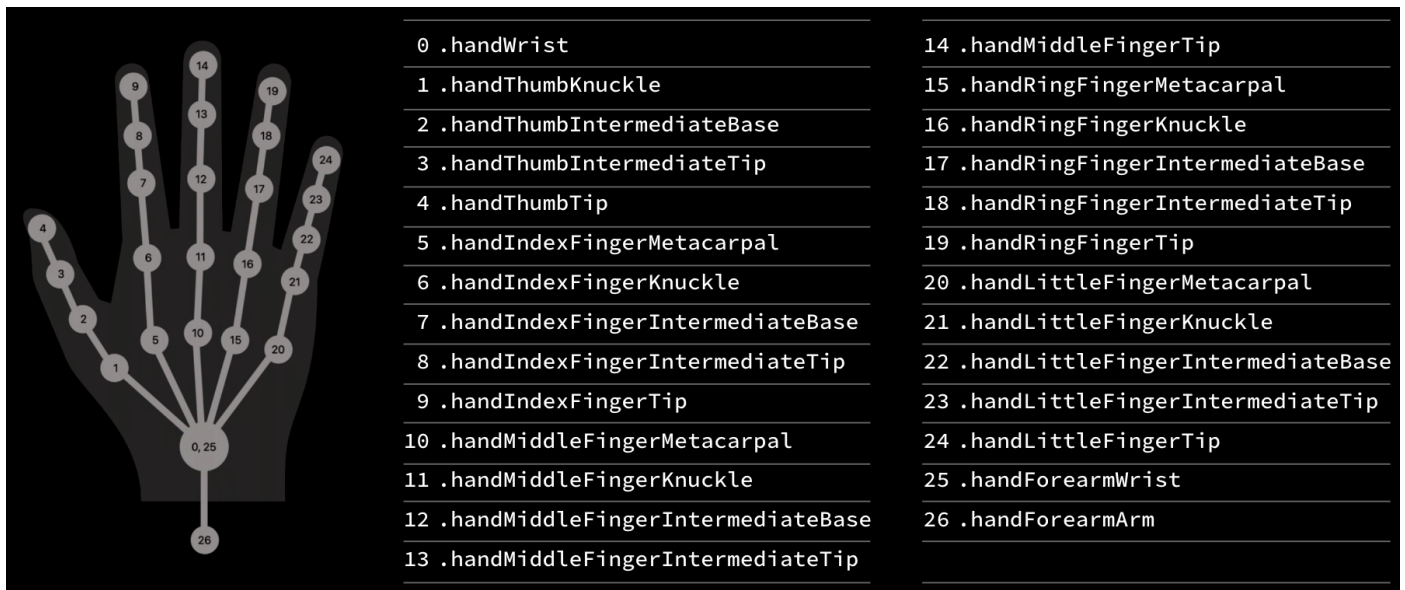
이 입력 기기에는 VisionOSSpatialPointerDevice라는 VR 대응 기기가 있습니다. 이 둘의 주요 차이점은 상호 작용에 콜라이더가 필요하지 않다는 점입니다. 따라서 VisionOSSpatialPointerDevice에는 상호 작용과 관련된 입력 컨트롤(targetId, interactionPosition 등)이 없습니다.

골격 기반 손 트래킹

골격 기반 손 트래킹은 XR Hands 패키지의 손 보조 시스템에서 제공됩니다. 씬에서 Hand Visualizer 컴포넌트를 사용하면 플레이어의 손에 스킨드 메시 또는 조인트별 지오메트리를 입힐 수 있을 뿐만 아니라 손 기반의 물리 상호 작용에 필요한 물리 오브젝트도 표시할 수 있습니다. 패키지 관리자에서 제공되는 **XR Hands** 패키지를 사용하여 원시 데이터에 액세스할 수도 있습니다.



XR Hands 패키지는 관절을 트래킹하며, 이를 통해 자신만의 제스처를 정의하거나 커스텀 메시로 손의 비주얼을 바꿀 수 있습니다.



손 데이터를 손 모습으로 매핑하기

다음 표에서 사용 가능한 입력 방법을 간략히 확인할 수 있습니다. 손 트래킹 데이터와 AR 데이터를 사용하려면 사용자의 허락이 필요합니다.

앱 유형	보고 탭하기 (입력 시스템 터치 공간 이벤트)	헤드 포즈 (입력 시스템 헤드 포즈 이벤트)	손 (XR Interaction Toolkit 또는 XR Hands 패키지)	AR 데이터 (AR Foundation 패키지)	Bluetooth 기기 (입력 시스템)
완전 몰입형 VR	지원	지원	지원	지원 안 함	지원
바운디드 모드의 MR	지원	지원 안 함	지원	지원 안 함	지원
언바운디드 모드의 MR	지원	지원	지원	지원	지원
원도우형 앱	지원	지원 안 함	지원 안 함	지원 안 함	지원

완전 몰입형 VR



Owlchemy Labs에서 개발한 Job Simulator는 visionOS를 지원하는 VR 앱입니다. [이 Unity 블로그 게시물](#)에서 게임에 대한 자세한 내용을 읽어 보세요.

이 전자책의 전반부에서 자세히 설명했듯이, VR 개발과 같이 익숙한 Unity 워크플로를 사용하여 VR 앱을 제작하거나 visionOS로 포팅할 수 있습니다.



visionOS의 빌드 프로세스는 iOS에서 배포하는 경우와 유사합니다. Unity에서 Xcode 프로젝트를 생성하면 시뮬레이터나 기기에서 프로젝트를 실행하고 퍼블리시할 수 있습니다.

Apple Vision Pro 개발의 맥락에서 보면, 완전 몰입형 VR 앱에서는 Unity의 그래픽스 기술과 [Metal 프레임워크](#)를 사용하여 프로젝트를 렌더링합니다.

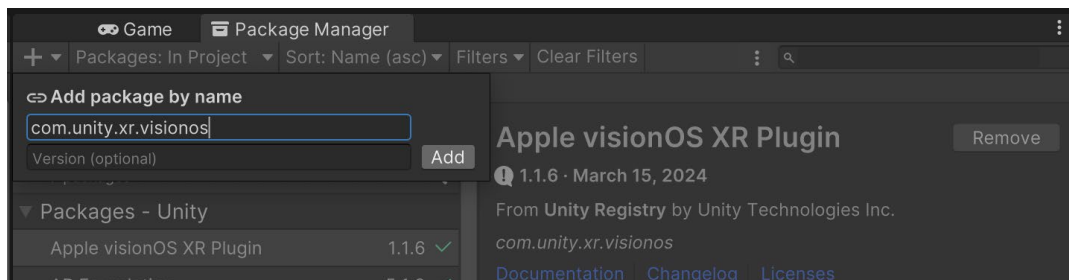
프로젝트가 이미 다른 VR 플랫폼에서 실행되고 있다면 visionOS에서도 실행될 가능성이 높습니다.

URP를 사용하는 것이 좋은데, URP에서 [포비티드 렌더링을 지원](#)하기 때문입니다. 이 기능은 시선이 집중될 확률이 가장 높은 시야 영역의 해상도에 더 많은 픽셀을 집중적으로 사용하고, 해당 영역 바깥쪽의 해상도는 낮춰 성능과 사용자 경험을 향상합니다. 포비티드 렌더링은 기본적으로 사용되며, 이를 사용하지 않으면 Apple Vision Pro에서 기기의 전체 네이티브 해상도로 렌더링할 수 없습니다. 기존 앱을 포팅하는 개발자는 문제가 생길 경우를 대비해 이 기능을 비활성화해야 할 수도 있지만 권장하지는 않습니다.

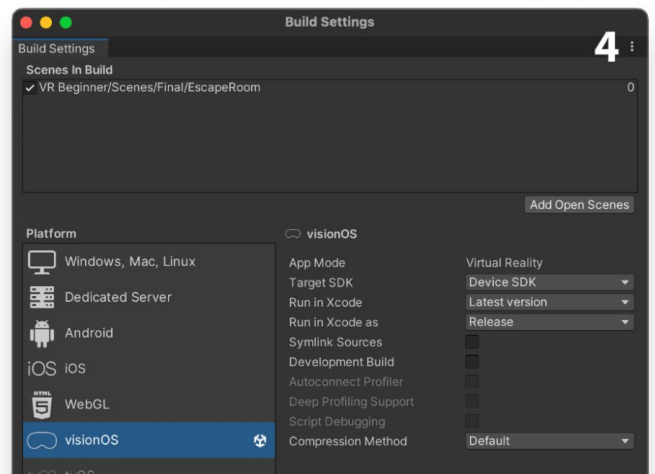
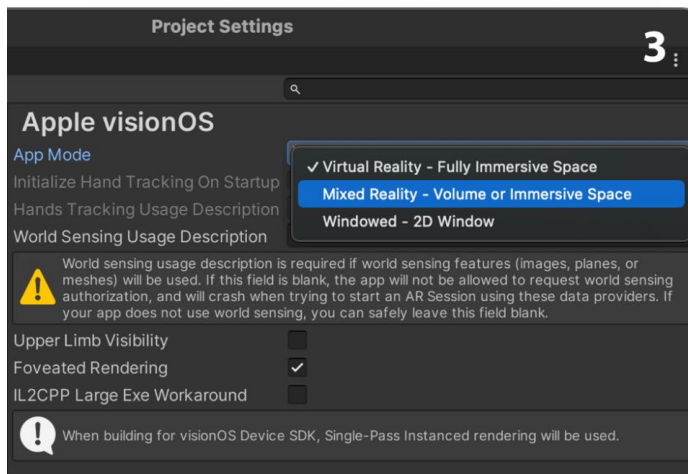
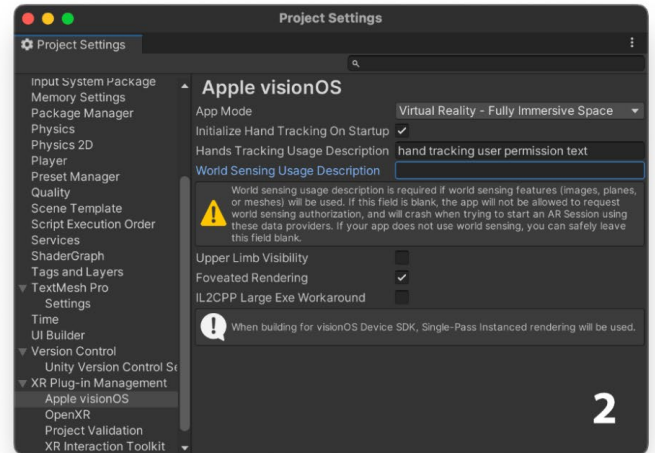
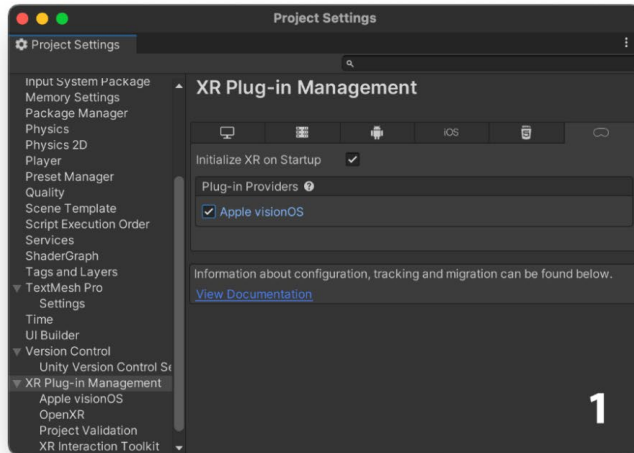
이제 Metal 그래픽스 API를 지원하는 [싱글 패스 인스턴스](#) 렌더링을 사용하여 프로젝트를 최적화할 수 있습니다. 싱글 패스 렌더링에서 Unity는 양쪽 눈에 대한 드로우 콜을 한 번 제출하므로 눈을 한 쪽씩 개별적으로 렌더링할 때 발생하는 CPU 오버헤드를 줄일 수 있습니다.

visionOS에서는 재투영에 데프 버퍼가 사용됩니다. 데프 정보가 누락된 경우 시스템에서 오류 색상을 렌더링하여 문제를 나타냅니다. 모든 Unity 셰이더는 바로 사용할 수 있도록 준비되어 있습니다.

앱을 완전 몰입형 모드로 실행하려면 Apple visionOS XR 플러그인을 설치해야 하며, 사용 중인 Unity 버전에서 이 플러그인이 표시되지 않는 경우에는 Package Manager 드롭다운에서 **com.unity.xr.visionos** 패키지를 선택하여 XR Plug-in Management 패키지에서 추가하면 됩니다.



또한 XR Plug-in Management 패키지의 Project Validation 섹션을 보고 visionOS에서 실행하기 위해 변경해야 하는 설정이 있는지 확인하는 것이 좋습니다.



visionOS 플러그인을 설치하여 어떤 유형(VR, MR 또는 Windowed)의 앱을 빌드할지 정의

XR Plug-in Management에서 visionOS 플러그인을 활성화한 후 왼쪽 메뉴에서 **Apple visionOS**를 클릭하고 앱 유형을 선택하면 Build Settings 창에 해당 유형이 반영됩니다.

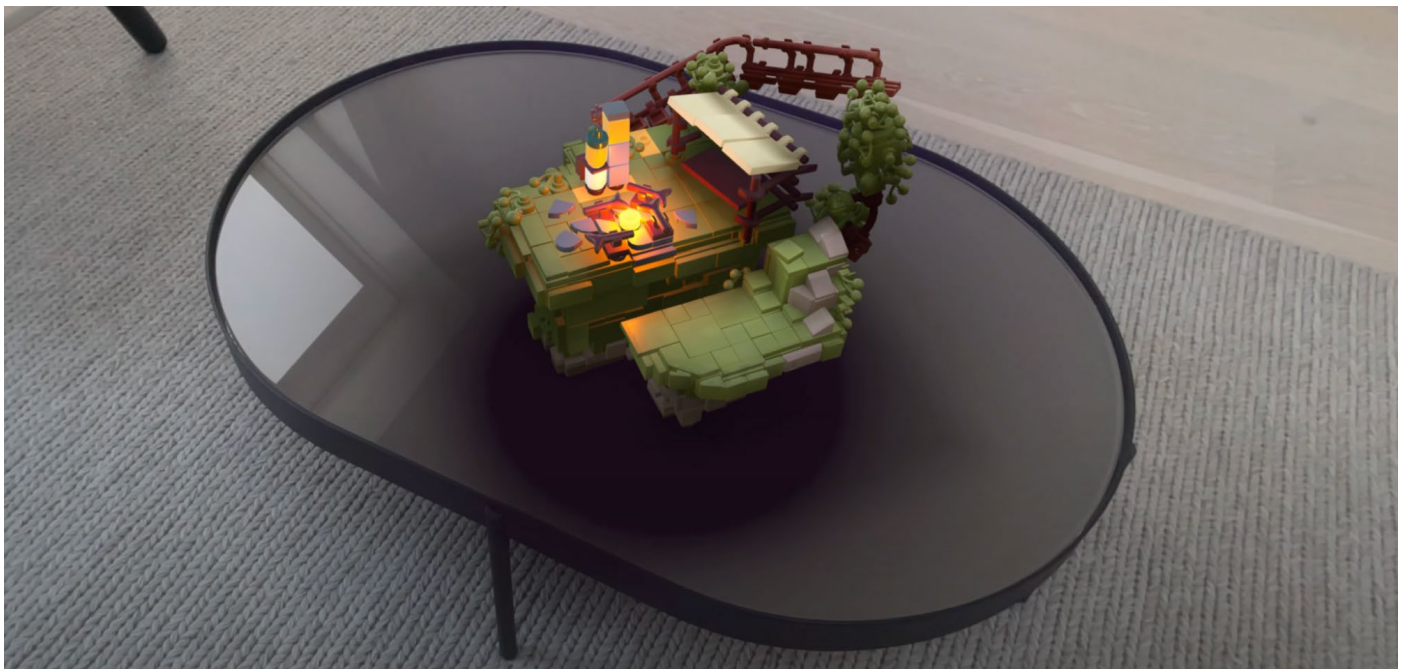
Xcode 시뮬레이터에서 프로젝트를 실행하려는 경우 Build Settings에서 **Target SDK>Simulator SDK**를 선택하고, Vision Pro 기기에서 프로젝트를 실행하려는 경우 **Device SDK**를 선택합니다.



visionOS 시뮬레이터에서 실행 중인 VR 앱

Vision Pro용 개발자 VR 앱에 대한 자세한 내용은 [기술 자료](#)를 읽어 보세요. 또한 유니티에서는 Unity 그래픽스 시스템과 툴을 사용하는 방법을 깊이 있게 알아볼 수 있는 [전자책](#)을 제공합니다.

MR 앱



6명으로 구성된 Light Brick Studio 팀에서 단 3개월 만에 LEGO® Builder's Journey를 Apple Vision Pro로 포팅한 [방법](#)을 알아보세요.



유니티의 PolySpatial 기술을 사용하여 Apple Vision Pro Shared Space에서 실행되는 MR 앱을 제작할 수 있습니다. 이러한 앱은 Unity로 시뮬레이션되지만, 렌더링은 visionOS의 시스템 렌더러인 RealityKit을 통해 이루어집니다. 대부분의 Unity VR 개발 패러다임이 여기에 적용되며, AR 컴포넌트가 추가됩니다.

MR 앱 개발을 시작하려면 Apple visionOS의 Project Settings 창으로 이동하여 앱 모드를 **Mixed Reality - Volume or Immersive Space**로 변경합니다. 그러면 PolySpatial을 설치할지 묻는 팝업 메시지가 표시됩니다. 또 다른 옵션은 com.unity.polyspatial.visionos라는 이름으로 패키지를 직접 추가하는 것입니다.

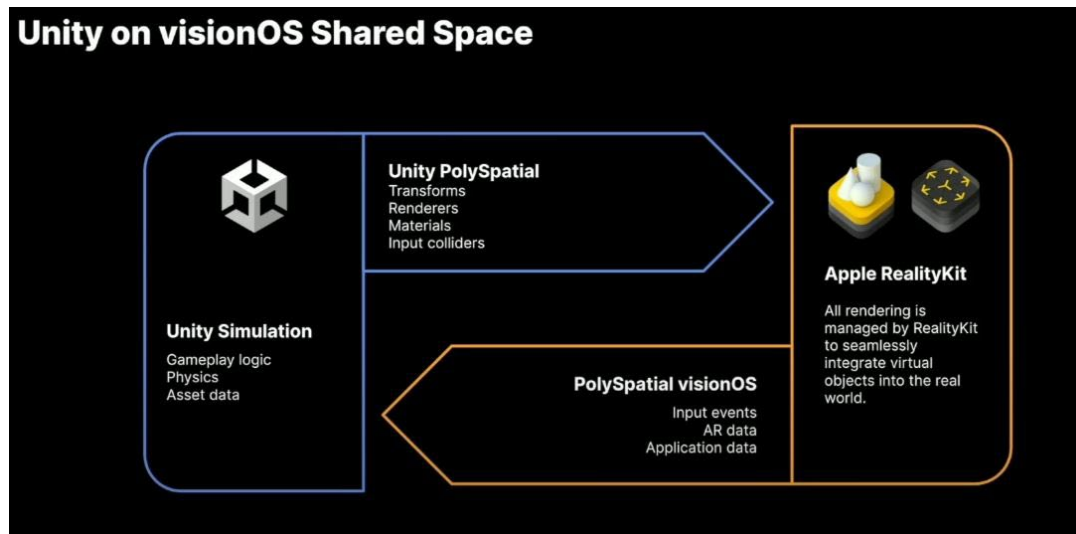


MR 앱을 제작하려면 visionOS 플러그인과 PolySpatial 패키지가 모두 필요



Unity PolySpatial

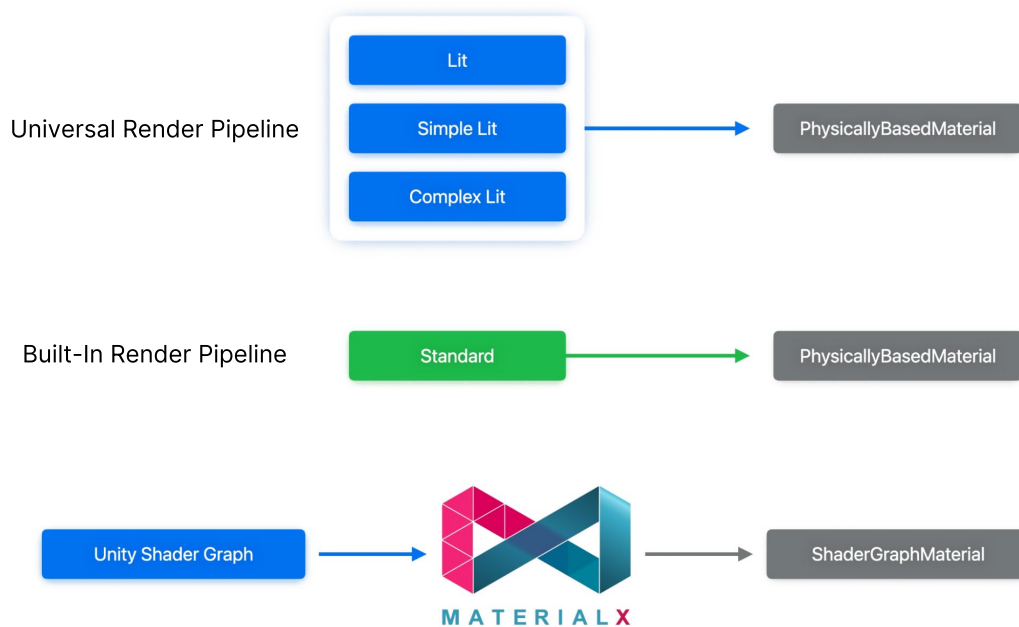
Unity의 visionOS 패키지에 포함된 Unity PolySpatial은 visionOS 플랫폼을 위한 공간 경험을 만들 수 있는 기반이 되는 기술로, 이를 통해 Shared Space 내에서 다른 앱과 공존 가능한 공간 경험을 제작할 수 있습니다.



Unity의 PolySpatial 기술을 통해 Apple의 RealityKit으로 전환되는 Unity의 시뮬레이션 및 그래픽스에 대한 개요

그래픽스 및 시뮬레이션

PolySpatial은 RealityKit을 사용해 그래픽스를 렌더링하는 데 필요한 변환 작업을 처리하고, Unity의 많은 기능과 워크플로가 MR에서 작동하도록 지원합니다.



Unity 머티리얼에서 RealityKit 머티리얼로 변환되는 내부 플로우

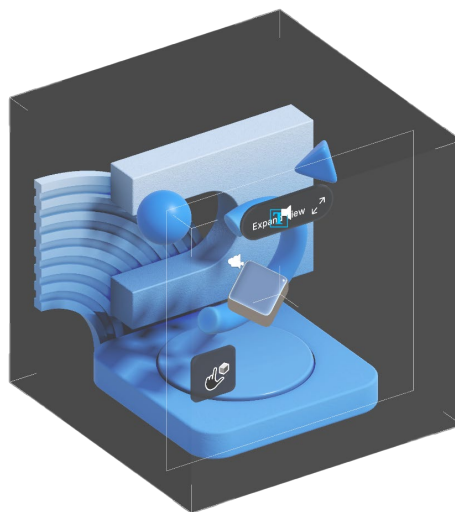


프로세스를 자세히 살펴보겠습니다.

- **머티리얼:** Shader Graph에서 만든 URP 릿, 심플 릿, 컴플렉스 릿 셰이더와 커스텀 머티리얼이 RealityKit에서 작동합니다. 복잡한 머티리얼과 커스텀 셰이더 전환을 위한 교환 형식으로 MaterialX가 사용됩니다. 대부분의 Shader Graph 노드가 지원되지만 [기술 자료](#)에서 지원에 대한 자세한 내용을 확인할 수 있습니다. 효과 머티리얼, 언릿 및 오클루전 머티리얼도 지원됩니다.
- **메시 렌더러:** 메시 렌더러와 스킨드 메시 렌더러가 URP 또는 빌트인 렌더 파이프라인에서 지원됩니다.
- **파티클 효과:** 빌트인 파티클 시스템으로 생성된 간단한 효과가 RealityKit으로 변환됩니다. 복잡한 효과는 메시로 베이킹되고 스프라이트 효과도 메시로 전환됩니다. VFX Graph는 MR에서 지원되지 않지만 효과를 렌더 텍스처로 렌더링하면 프로젝트에서 사용할 수 있습니다.
- **스프라이트:** 스프라이트가 지원됩니다.
- **시뮬레이션:** MonoBehaviour, 스크립터블 오브젝트 및 기타 표준 툴을 비롯한 다음과 같은 월드 시뮬레이션 기능은 계속 작동합니다.
 - 물리
 - 애니메이션 및 타임라인
 - 경로 탐색
 - MonoBehaviour
 - 렌더링에 해당하지 않는 기타 [기능](#)

볼륨 카메라

볼륨 카메라는 PolySpatial에서 새롭게 도입된 개념입니다. 앱은 언제든지 두 가지 유형 간에 전환할 수 있습니다. Volume Camera 컴포넌트는 레이어 마스크를 컬링하고 크기를 정의할 수 있으며(크기가 클수록 씬 뷰가 더 많이 표시됨), 앱 창의 변화에 따라 함수를 트리거하는 Unity 이벤트가 있습니다. 이 카메라 모드는 **Volume Camera Configuration** 에셋을 통해 설정됩니다.



Unity에서 바운딩드 카메라 설정하기



바운디드 카메라

이 모드에서는 Unity에서 실제 단위를 기반으로 기기에서 사용되는 공간의 면적을 정의하여 볼륨 카메라 내부에 콘텐츠를 표시할 수 있습니다.

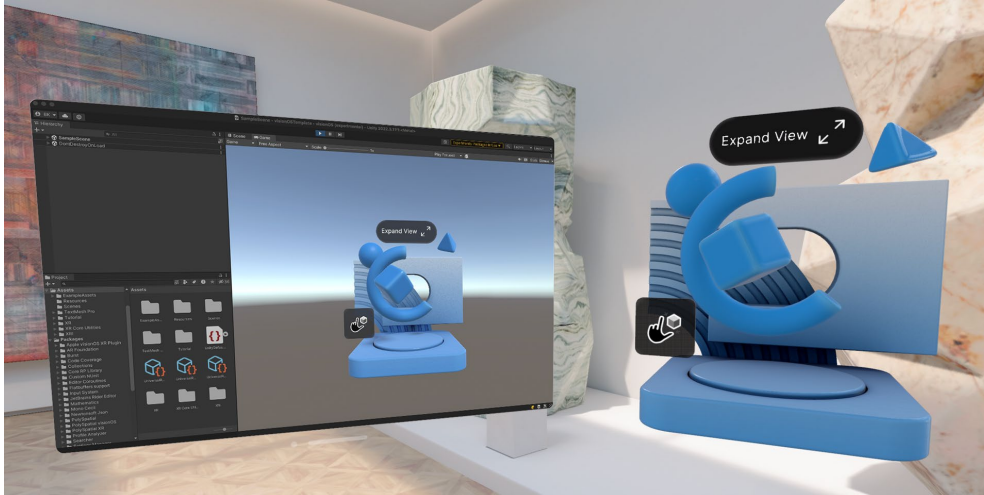


기술 자료에서 바운디드 카메라 샘플을 찾아 살펴보세요.



언바운디드 카메라

이 모드에서는 MR 경험을 전체 공간으로 전환할 수 있습니다. 여기서 앱은 사용자의 전체 시야를 자유롭게 활용할 수 있습니다. 이 공간에는 언바운디드 콘텐츠만 표시됩니다. 볼륨 카메라의 크기나 출력 크기는 설정할 수 없으며, Unity 단위는 실제 단위(1 Unity 단위 = 1미터)에 매핑됩니다. 언바운디드 카메라는 한 번에 하나만 있을 수 있습니다.

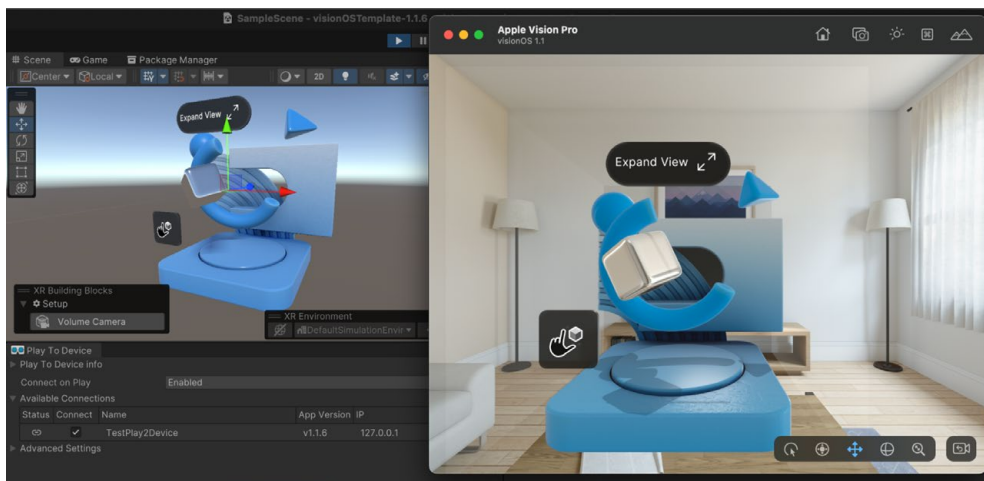


공간을 최대한 활용하는 언바운디드 MR 앱. [기술 자료](#)에서 테스트할 다른 샘플에 대해 읽어보세요.

Play To Device

이 기능을 사용하면 개발자는 Xcode 빌드를 제작할 필요 없이 플레이 모드에서 visionOS 기기와 시뮬레이터상의 콘텐츠를 미리 볼 수 있습니다. 이는 혼합 현실 앱인 RealityKit을 통해 렌더링된 앱에서 사용할 수 있습니다.

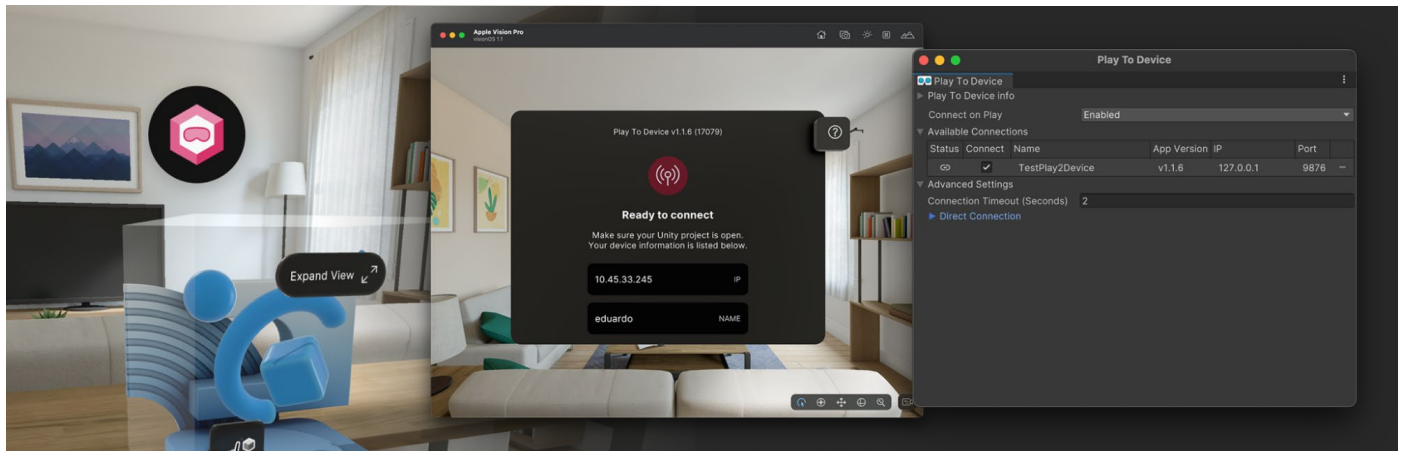
Play to Device를 사용하면 요소의 배치 상황을 효율적으로 확인하고, 머티리얼 텍스처와 Shader Graph를 변경하고, 상호 작용을 테스트하고, 값을 조정하는 등의 작업을 수행할 수 있습니다. 기기 또는 시뮬레이터에서 수행하는 모든 상호 작용은 에디터에 다시 동기화됩니다. 이 워크플로에 대한 자세한 내용은 [Discussions 게시물](#) 또는 [기술 자료](#)에서 확인할 수 있습니다.



Unity의 플레이 모드와 나란히 실행 중인 Play To Device



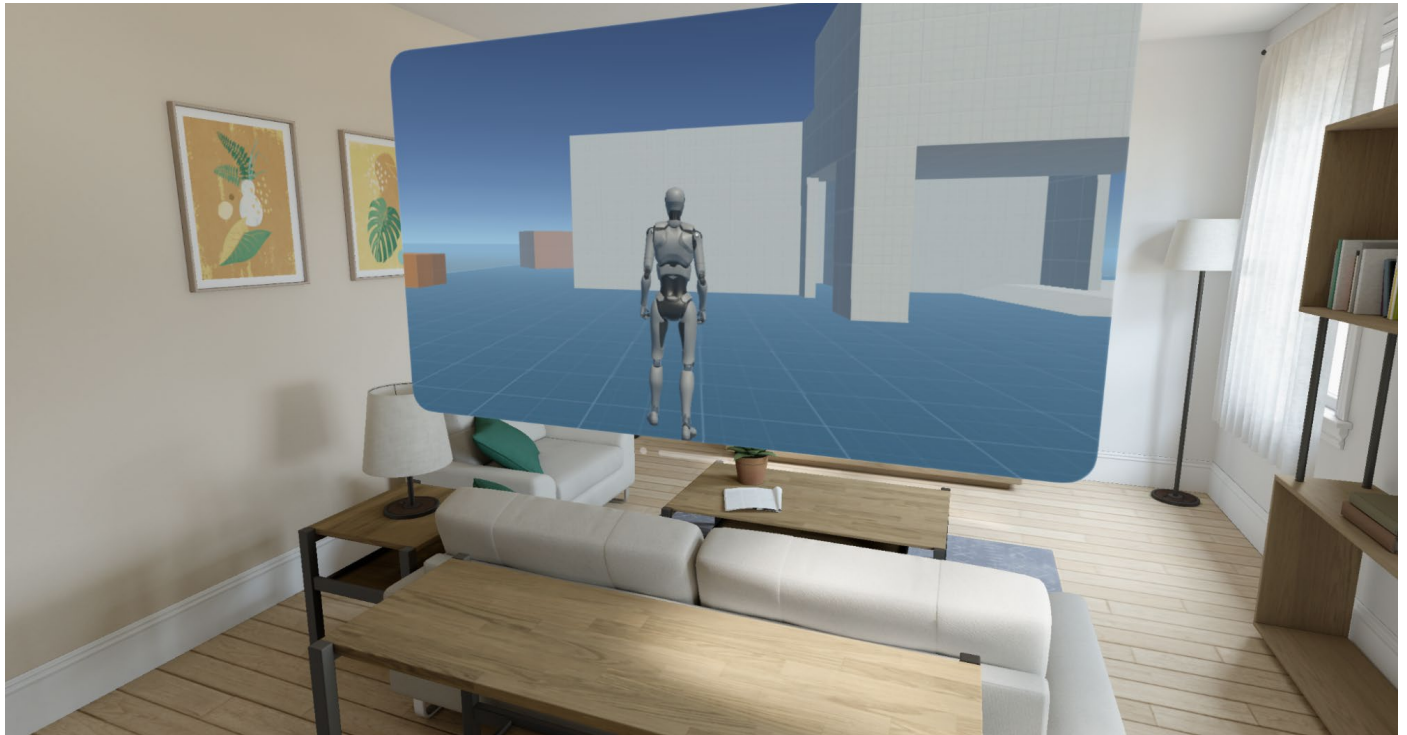
이 기능은 Play to Device Host 애플리케이션에서 제공되며, 이 애플리케이션은 visionOS 시뮬레이터에 설치하거나 Testflight를 통해 Apple Vision Pro에 바로 설치할 수도 있습니다.



시뮬레이터에서 드래그 앤 드롭으로 Play to Device를 설치하고 열면 Unity에서 해당 기능이 호스트 애플리케이션을 감지하며 플레이 모드에 진입 시 해당 애플리케이션에 연결됩니다(Connect on Play에서 비활성화할 수 있음).

윈도우형 앱

이 모드에서는 콘텐츠가 2D 창에 표시되므로 사용자는 Shared Space 내에 있는 콘텐츠의 크기와 위치를 조절할 수 있습니다. 완전 몰입형 VR 앱과 마찬가지로 윈도우형 앱은 Metal 기반의 Unity 렌더링을 활용하므로 기존에는 2D로 표시하도록 제작된 게임도 visionOS에서 윈도우형 모드로 실행할 수 있습니다.



visionOS에서 실행 중인 윈도우형 Unity 애플리케이션



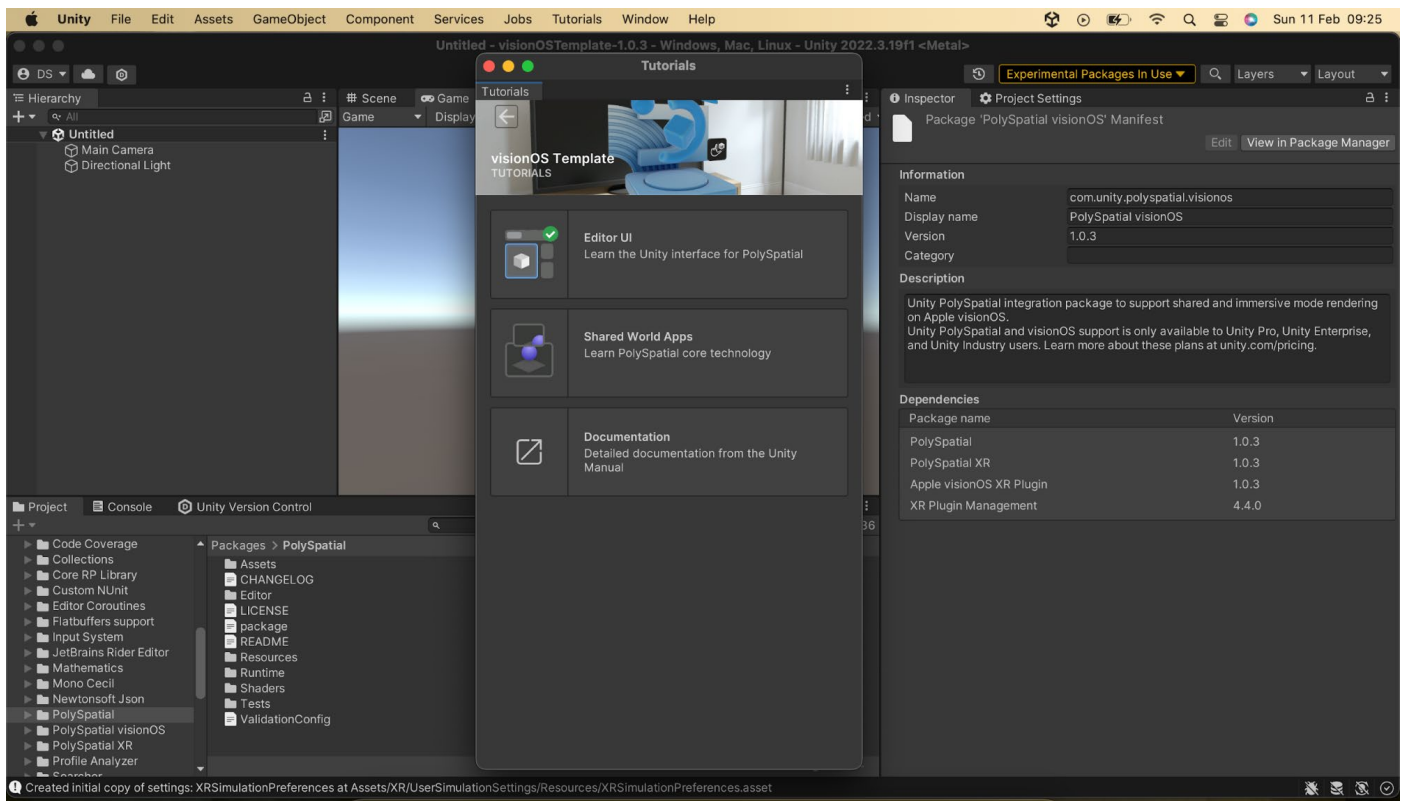
visionOS 프로젝트 템플릿

새 프로젝트를 시작하는 좋은 방법은 visionOS 프로젝트 템플릿을 다운로드하는 것입니다. [‘몰입형 앱 템플릿으로 새로운 visionOS 프로젝트 시작하기’](#)에서 제공하는 템플릿 링크를 통해 템플릿을 다운로드할 수 있습니다.

visionOS 템플릿에는 바운디드 및 언바운디드 MR 경험, 반응형 레이아웃, XR 입력 및 상호 작용의 예시가 포함되어 있습니다. 이 샘플의 언바운디드 볼륨 씬에서는 탭, 손 트래킹, 평면 검출을 사용합니다.

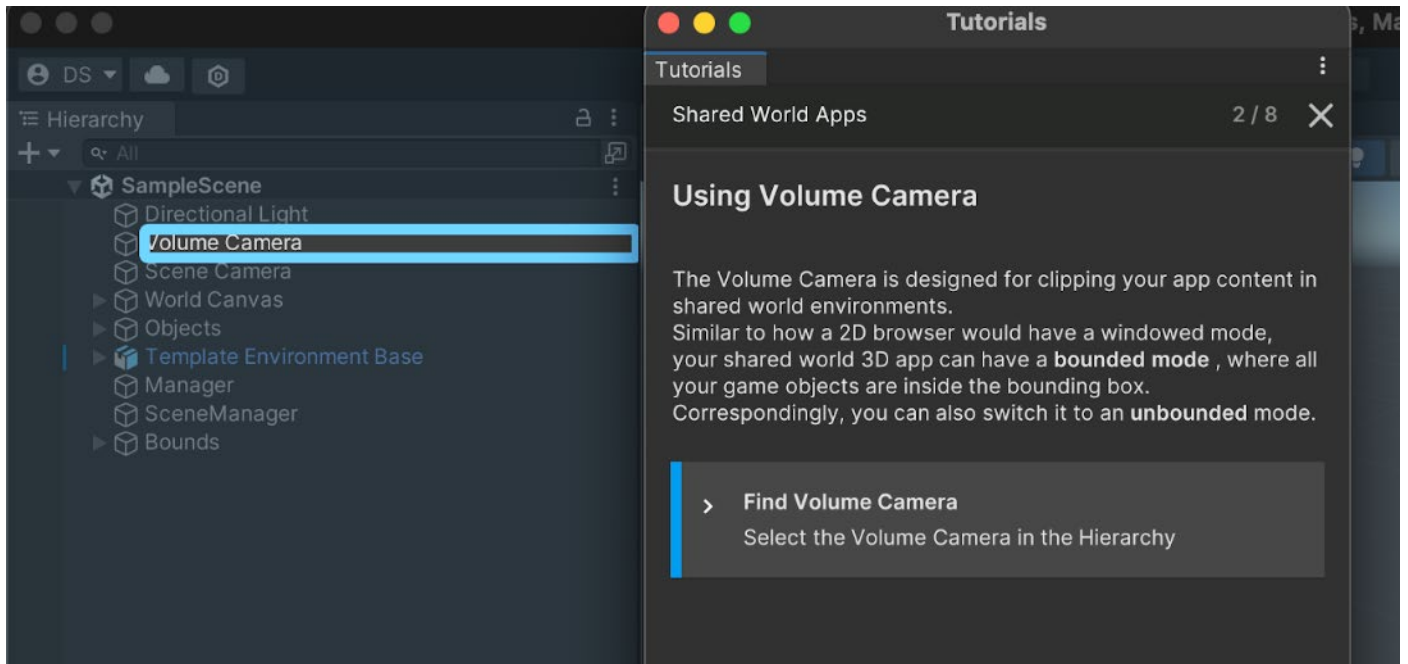
이 템플릿은 프로젝트 설정을 구성하고, 적절한 XR 패키지를 사전에 설치하고, 사전 설정된 다양한 에셋을 포함하여 visionOS에 배포할 준비를 마친 프로젝트를 설정하는 방법을 보여 줍니다.

Unity Hub에서 **디스크에서 프로젝트 추가** 옵션을 선택하여 템플릿을 열 수 있습니다. 템플릿이 열리면 에디터 내 튜토리얼을 사용할 수 있으며, 튜토리얼이 보이지 않으면 **Tutorials > Show Tutorials**로 이동합니다.



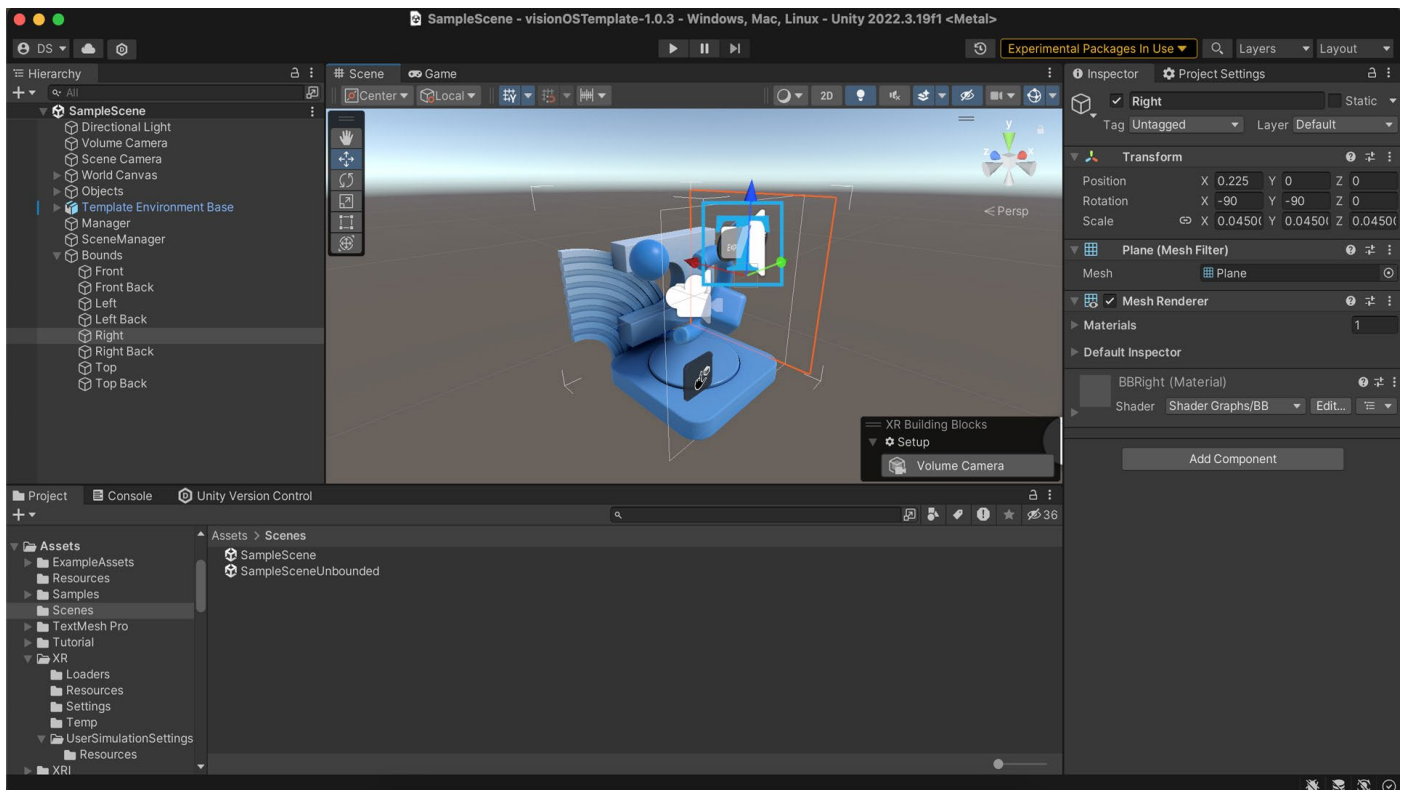
다양한 PolySpatial 리소스에 대한 액세스를 제공하는 Tutorials 창

튜토리얼에서는 템플릿의 다양한 기능을 안내합니다.



프로젝트와 컴포넌트에 대한 안내

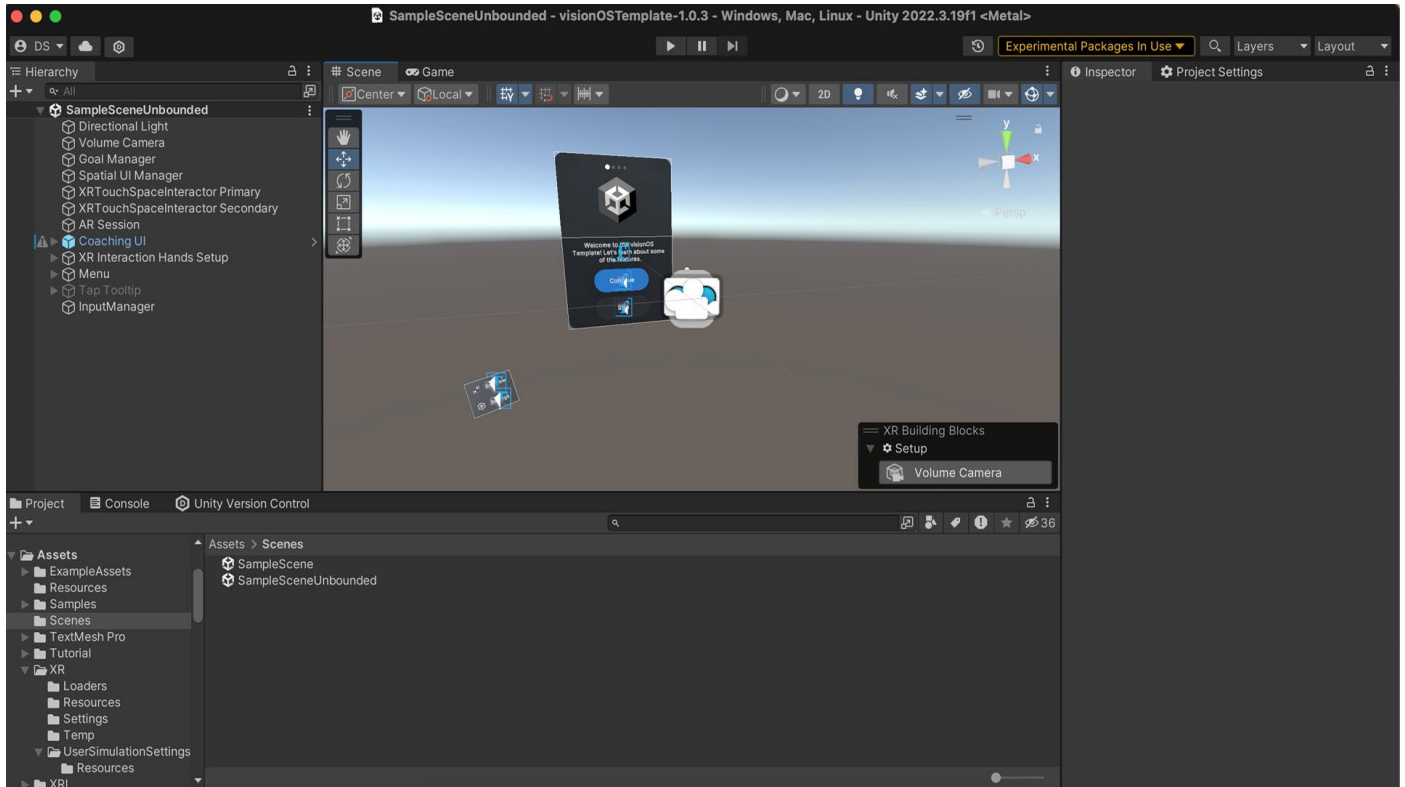
프로젝트에 포함된 두 개의 샘플 씬인 SampleScene과 SampleSceneUnbounded에서는 각각 바운디드 및 언바운디드 모드의 설정과 기능을 보여 줍니다.



바운디드 MR 앱의 접근 방식을 보여 주는 샘플 씬



앞에서 설명한 대로 visionOS 시뮬레이터와 Play To Device 애플리케이션을 사용하여 에디터 내에서 이러한 앱을 체험할 수도 있습니다.



언바운디드 MR 앱 설정을 보여 주는 SampleSceneUnbounded

더 많은 리소스

Unity 커뮤니티에 참여해 visionOS 개발과 관련된 Unity의 [Discussion 공간](#)에서 XR 개발에 대한 정보와 팁을 얻을 수 있습니다. 향후에 출시될 기능에 대한 자세한 내용은 [로드맵](#)을 참조하세요.

프로페셔널 트레이닝 서비스

visionOS에서 경험을 전문적으로 제작하고자 하는 사람들을 위해 유니티에서는 1:1 라이브 및 실습 교육을 제공합니다. [이 맞춤형 교육](#)에서는 Unity 전문가가 다음과 같은 내용을 소개합니다.

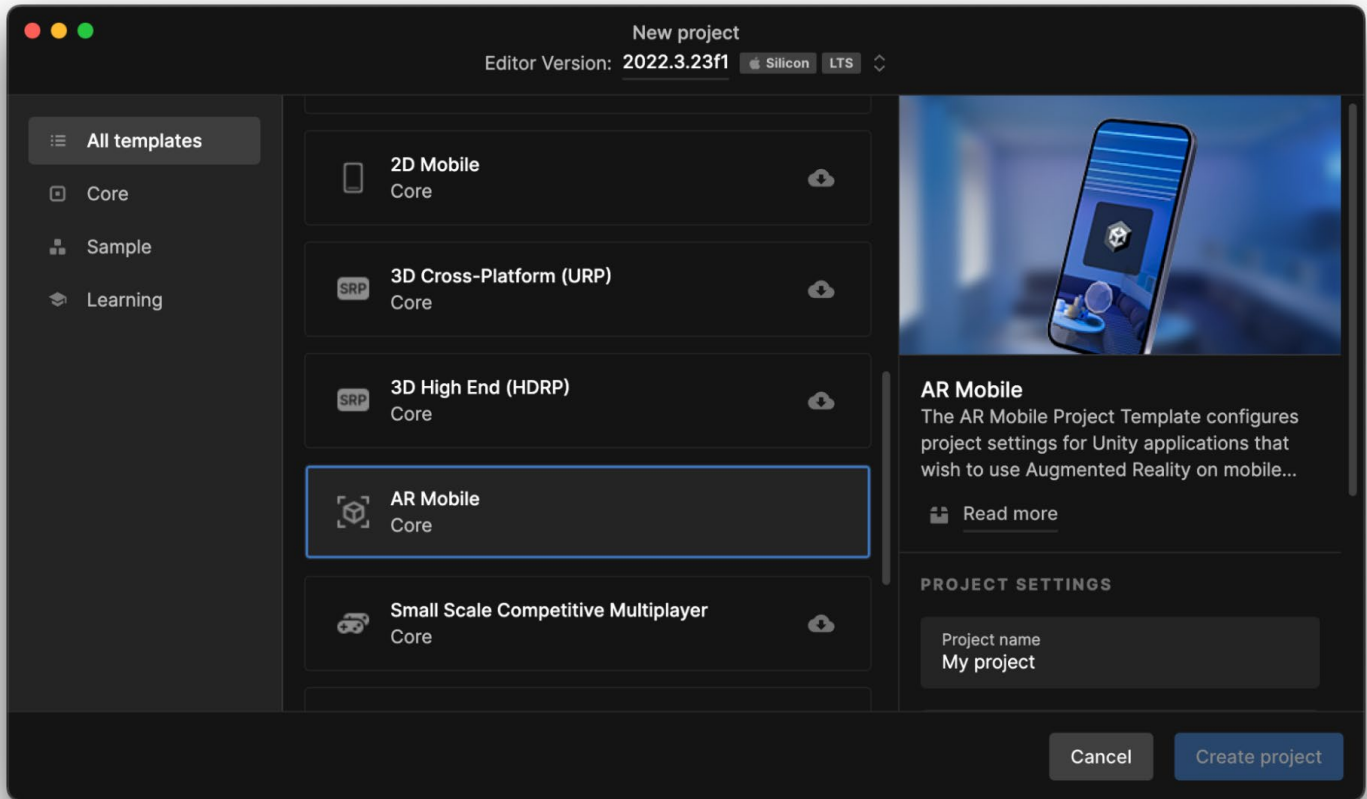
- visionOS와 관련된 목표와 고유한 프로젝트
- visionOS 개발의 핵심 개념, Apple Vision Pro와 프로젝트에 필요한 해당 기능에 대한 이해
- 개발 환경, 구성 설정, 첫 번째 애플리케이션 빌드

Unity에서 AR 구현하기



유명 AR 개발사 Niantic Labs의 최신 프로젝트인 Peridot 스크린샷

AR(증강 현실) 애플리케이션 개발은 ARKit(iOS용) 및 ARCore(Android용) 등의 AR 개발 플랫폼으로 구성된 [AR Foundation](#)과 같은 Unity의 강력한 프레임워크를 활용하는 흥미로운 과정입니다. 이 프레임워크에서는 다양한 플랫폼에 대한 폭넓은 지원과 광범위한 커뮤니티 리소스를 볼 수 있습니다.



처음 시작할 때 살펴보면 좋은 AR Mobile Core 템플릿. 자세한 내용은 [여기](#)에서 확인해 보세요.

용어집

A

AR Foundation: 개발자가 단일 API를 사용하여 여러 플랫폼에서 AR(증강 현실) 경험을 구축할 수 있는 Unity 프레임워크입니다.

ASTC(Adaptive Scalable Texture Compression): 품질과 성능 간의 균형을 관리하는 데 도움이 되는 텍스처 압축 기술로, 모바일 및 VR 애플리케이션에 특히 유용합니다.

B

베이컨된 조명: 사전 계산된 조명 정보가 텍스처 맵에 저장되어 실시간으로 조명을 계산할 필요가 없으므로 그래픽 애플리케이션의 성능을 향상하는 데 사용됩니다.

C

CPU 사용: 애플리케이션에서 사용 중인 프로세서 리소스의 양을 측정합니다. CPU 사용이 많으면 애플리케이션의 성능과 응답성에 영향을 미칠 수 있습니다.

크로스 플랫폼 개발: 거의 또는 전혀 수정하지 않고 여러 하드웨어 플랫폼에서 실행할 수 있는 애플리케이션을 개발하는 방식입니다.

D

드로우 콜: 그래픽스 렌더링을 위해 GPU로 전송되는 커맨드로, 드로우 콜 수를 줄이면 그래픽 성능을 크게 향상할 수 있습니다.

DSP 부하(디지털 신호 처리 부하): 오디오 신호를 처리하는 데 사용되는 처리 능력의 양입니다. DSP 부하가 많으면 성능에 영향을 미칠 수 있습니다.

F

프레임 속도: 초당 표시되는 프레임 수입니다. 원활한 게임플레이와 VR 경험을 보장하려면 프레임 속도가 안정적이면서 높아야 합니다.

G

가비지 컬렉션: 애플리케이션에 더 이상 필요하지 않은 오브젝트에 할당된 메모리를 회수하는 프로세스입니다. 잦은 가비지 컬렉션은 성능 스파이크를 발생시킬 수 있습니다.

GPU 사용량: 그래픽스 프로세서의 리소스 사용량을 측정합니다. 이는 렌더링 성능과 비주얼 품질을 결정하는 핵심 요소입니다.

H

HDRP(고해상도 렌더 파이프라인): 고사양 하드웨어에서 충실도 높은 그래픽스를 구현하도록 설계된 Unity 렌더링 파이프라인입니다.



I

입력 시스템: 키보드, 마우스, 게임 컨트롤러와 같이 다양한 기기의 입력을 통합된 방식으로 관리할 수 있는 Unity 시스템입니다.

Interaction Layer Mask: Unity의 XR Interaction Toolkit에서 레이어에 따라 인터랙터가 상호 작용할 수 있는 오브젝트를 지정하는 데 사용됩니다.

L

LOD(디테일 수준): 카메라와의 거리에 따라 3D 모델의 복잡도를 조정하여 성능을 향상하는 데 사용되는 기법입니다.

M

메모리 사용량: 애플리케이션에서 사용 중인 RAM의 양입니다. 메모리 사용량을 최적화하는 작업은 특히 VR과 같이 리소스가 제한된 환경의 성능에 중요합니다.

N

노멀 맵: 폴리곤 수를 늘리지 않고도 디테일이 뛰어난 표면을 시뮬레이션하여 표면에 깊이와 디테일을 더해주는 텍스처 유형입니다.

O

OpenXR: 다양한 플랫폼의 VR 및 AR 애플리케이션 개발을 표준화하는 것을 목표로 하는 로열티 프리 표준입니다.

최적화: 리소스 소비를 줄이고 성능을 개선하여 애플리케이션을 더 효율적으로 실행하는 프로세스입니다.

P

패스트루: 사용자가 기기의 카메라를 통해 주변 현실 세계를 볼 수 있는 VR 기기의 기능입니다.

성능 최적화: 애플리케이션의 속도와 효율성을 향상하는 데 사용되는 기술로, VR/AR에서 높은 프레임 속도와 원활한 작동을 유지하는 데 중요합니다.

PolySpatial: 공간 컴퓨팅과 관련된 기술 또는 개념으로, 물리적 공간 내에서 디지털 오브젝트와 환경이 원활하고 직관적인 방식으로 통합되고 상호 작용하는 데 중점을 둡니다. Unity에서 PolySpatial은 실시간 3D 콘텐츠 제작 및 상호 작용에 필요한 플랫폼의 기능을 활용하여 공간이 통합된 경험을 쉽게 제작할 수 있도록 설계된 톨, 프레임워크 또는 작업 방식을 의미합니다.

Q

품질 설정: 개발자가 애플리케이션의 그래픽 품질을 조정하여 성능이나 시각적 충실도를 최적화할 수 있는 Unity 설정입니다.

R

렌더링 파이프라인: 그래픽스가 처리되어 화면에 표시되는 프로세스입니다. Unity에서는 URP와 HDRP를 비롯한 여러 렌더링 파이프라인을 제공합니다.

리깅: 3D 모델이 사실적으로 움직일 수 있도록 뼈대의 구조를 만드는 프로세스입니다.

S

SDK(소프트웨어 개발 키트): 개발자가 특정 플랫폼 또는 기기에 맞게 애플리케이션을 만들 수 있도록 설계된 일련의 소프트웨어 톨과 라이브러리입니다.

SRP(스크립터블 렌더 파이프라인): 개발자가 프로젝트의 특정 요구 사항을 충족하도록 렌더링 프로세스를 커스터마이징할 수 있는 Unity 기능입니다.

T

텍셀 밀도: 모델에 적용되는 텍스처 픽셀(텍셀)의 밀도를 나타내며, 텍스처의 비주얼 품질과 성능에 영향을 미칩니다.

트림 시트: 다양한 디테일이 포함된 단일 텍스처를 사용해 3D 모델의 여러 부분에 적용함으로써 복수의 텍스처 없이도 복잡도를 더할 수 있는 텍스처링 기법입니다.

U

URP(유니버설 렌더 파이프라인): 모바일부터 고사양 PC에 이르기까지 다양한 기기의 성능에 최적화된 Unity 렌더링 파이프라인입니다.

사용자 상호 작용: 컨트롤러, 제스처, 보이스 커맨드 등을 비롯하여 사용자가 게임 또는 애플리케이션과 상호 작용하는 방식입니다.

V

VR(가상 현실): 컴퓨터로 생성된 환경과 사용자가 상호 작용할 수 있는 기술로, 주로 헤드셋을 사용한 몰입형 경험을 제공합니다.

Velocity Tracking Movement: 인터랙터의 속도에 따라 오브젝트의 움직임을 계산하여 사실적인 물리 상호 작용을 구현할 수 있는 Unity XR Interaction Toolkit의 기능입니다.

X

XR(확장 현실): AR(증강 현실), VR(가상 현실), MR(혼합 현실)을 비롯하여 컴퓨터를 통해 변형된 모든 형태의 현실을 포괄하는 용어입니다.

XR Interaction Toolkit: 인터랙티브 XR 애플리케이션을 빌드하는 데 필요한 일련의 툴과 컴포넌트를 제공하는 Unity 패키지입니다.



unity.com/kr