

# 유니티와 언리얼 엔진 4 에서의 가상현실 개발환경에 관한 비교연구

조윤식<sup>0,1</sup>

김진모<sup>1,2,\*</sup>

한성대학교 일반대학원 정보컴퓨터공학과<sup>1</sup>, 한성대학교 컴퓨터공학부<sup>2</sup>

{whdbstlrwkd, jinmo.kim}@hansung.ac.kr

## A Study on the Comparison of the Virtual Reality Development Environment in Unity and Unreal Engine 4

Yunsik Cho<sup>0,1</sup>

Jinmo Kim<sup>1,2,\*</sup>

Department of Information and Computer Engineering, Graduate School, Hansung University<sup>1</sup>,

Division of Computer Engineering, Hansung University<sup>2</sup>

### 요 약

게임엔진은 실감형 콘텐츠를 제작하는 데 있어서 최소한의 시각적 품질 보증과 함께 멀티 플랫폼 등의 지원으로 개발시간 단축 등 효율적인 콘텐츠제작이 가능하다는 이점이 있다. 최근 게임엔진은 가상현실 HMD(Head Mounted Display)를 활용하는 실감형 콘텐츠를 쉽고 빠르게 효과적으로 제작할 수 있는 다양한 기능들을 제공하고 있다. 따라서, 본 연구는 게임을 비롯하여 콘텐츠제작 산업 분야에서 많이 활용되는 유니티와 언리얼 게임엔진을 중심으로 오culus 퀘스트 2 HMD를 활용한 가상현실 콘텐츠제작에서의 개발환경에 관한 비교연구를 수행한다. 먼저 각 엔진이 제공하는 최소한의 기능과 설정이 포함된 가상현실 템플릿 프로젝트를 기반으로 오culus 퀘스트 2 HMD와 전용 컨트롤러를 활용하여 개발환경을 구축하는 기본 설정 과정을 비교한다. 다음으로 가상환경에서 상호작용할 수 있는 간단한 체험환경을 제시하고 전용 컨트롤러를 활용하기 위한 개발환경과 오culus 퀘스트 2에서 제공하는 손 추적기능을 통해 실제 손을 직접 활용하는 개발환경 구축과정을 엔진별로 비교한다. 이러한 과정을 통해 가상현실 개발환경을 구축하는 기본적인 처리 과정을 이해함과 동시에 엔진이 가지는 특징과 차이를 확인하여 다양한 실감형 콘텐츠제작에 응용할 수 있는 연구로 활용하고자 한다.

### Abstract

Game engines have the advantage of enabling efficient content production, such as reducing development time, with minimal visual quality guarantees and support for multi-platforms. Recently, game engines have provided various functions that can easily, quickly, and effectively produce immersive content using virtual reality (VR) HMD. Therefore, this study conducts a comparative study on the development environment in VR content production using Oculus Quest 2 HMD, focusing on Unity and Unreal game engines, which are widely used in the content production industry, including games. First, we compare the basic setup process of building a development environment using Oculus Quest 2 HMD and a dedicated controller based on a VR template project that includes the minimum functions and settings provided by each engine. Next, we present a simple experience environment that can interact in a virtual environment and compare the development environment to use a dedicated controller and the process of building a development environment that directly utilizes hands through the hand tracking function provided by Oculus Quest 2. Through this process, we will understand the basic process of building a VR development environment, and at the same time, we will check the characteristics and differences of the engine and use it as a research that can be applied to various immersive content production.

**키워드:** 게임엔진, 가상현실, 실감형 콘텐츠, 유니티 엔진, 언리얼 엔진

**Keywords:** Game engine, virtual reality, immersive content, Unity engine, Unreal engine

\*corresponding author: Jinmo Kim/ Hansung University (jinmo.kim@hansung.ac.kr)

## 1. 서론

정보통신 기술의 발전과 더불어 높은 성능의 하드웨어와 소프트웨어를 활용하여 게임을 비롯한 교육, 영화, 광고, 엔터테인먼트 등 다양한 산업 분야에서 디지털 콘텐츠가 제작되고 있다. 최근에는 사용자가 아바타를 통하여 실제 현실과 같은 가상공간에서 사회, 경제, 교육, 문화, 과학 기술 활동을 할 수 있는 메타버스 기술 및 플랫폼에 관한 연구들이 활발히 진행되고 있다. 네트워킹을 기반으로 다수의 사용자가 함께 참여하는 3차원 게임은 물론이고, 제페토(ZEPETO)[1]와 같이 가상의 공간에서 사용자를 대표하는 아바타를 통하여 다른 사용자와 대화를 하거나 상호작용을 수행하는 등 시간과 공간의 제약 없이 현실 세계에서 수행할 수 있는 모든 일을 가상의 공간에서도 누릴 수 있고 나아가 현실 세계에서는 경험할 수 없는 경험까지도 제공하여 사용자에게 높은 만족감을 제공하는 체험환경으로 발전되고 있다. 이와 관련하여 COVID-19로 인해 비대면 추세가 확산되는 사회적 배경과 더불어 대학교 캠퍼스를 가상환경으로 구성하여 입학식 등의 행사를 진행하기도 하였다[2].

다양한 3차원 기반의 가상환경을 효율적으로 제작하는 과정에서 유니티 등과 같은 게임엔진을 활용하는 사례가 증가하고 있다. 유니티 엔진이 제공하는 포스트 프로세싱기능을 활용하여 실감나는 가상현실 힐링 콘텐츠[3]나 GoogleVR SDK를 기반으로 유니티 엔진에서 모바일 플랫폼 가상현실에서의 자연스러운 이동 기능[4]에 대한 연구도 진행되었다. 이외에도 효과적인 운동 자세 트레이닝을 위한 시각적, 촉각적 피드백 설계[5], 데이터 기반의 인공지능이 정적 정서에 미치는 영향[6] 등 엔진을 활용하여 콘텐츠를 제작하는 것 외에도 다양한 방향으로 응용하는 연구들이 현재까지 이루어지고 있다.

게임을 포함한 대화형 구조의 콘텐츠 및 애플리케이션을 개발, 제작하기 위하여 자체적으로 게임엔진을 개발하기도 하지만 이는 큰 비용과 시간이 발생하는 문제점들이 존재한다. 따라서, 최근에는 게임엔진을 직접 개발하기보다는 라이선스 비용을 지불하여 사용하는 방식도 많이 채택되고 있다. 게임엔진 라이선스를 구매하여 콘텐츠를 개발하면 개발 시간을 단축할 수 있고 해당 엔진이 보증하는 최소한의 시각적인 품질까지도 구현할 수 있으며 다양한 플랫폼에서 실행할 수 있는 장점이 있다. 대표적인 상용 게임엔진으로 에픽게임즈의 언리얼 엔진, 유니티 테크놀로지의 유니티 3D 엔진, 크라이텍의 크라이엔진, 밸브 코퍼레이션의 소스엔진 등이 있다. 그리고 이러한 게임엔진들은 가상현실, 증강현실 기술을 포함하여 실감형 콘텐츠를 효과적으로 제작할 수 있는 플랫폼으로 발전되고 있다.

가상현실은 사용자에게 현실 세계와 유사한 경험은 물론이고 현실 세계에서는 경험할 수 없는 새로운 경험까지도 제공한다. 특히, HMD (Head Mounted Display)를 활용하여

시각적 몰입을 높이는 가상현실 시스템은 사용자에게 높은 만족감, 현존감을 제공한다. 그중 오쿨러스 퀘스트 2 (Oculus Quest 2)는 메타 (Meta)에서 제공하는 HMD로 향상된 올인원 폼팩터(form factor), 높은 수준의 해상도, 컨트롤러 및 손 추적기능 등을 탑재한 차세대 가상현실 콘텐츠 플레이에 최적화된 장비 중 하나이다. 현재 세계에서 가장 높은 가상현실 HMD 시장점유율을 차지하고 있는 것으로 분석되고 있다[7]. 따라서, 본 연구는 가상현실 HMD를 기반으로 하는 가상현실 개발환경 구축에 관한 비교연구를 위하여 오쿨러스 퀘스트 2를 활용하였다. 유니티 엔진을 비롯한 다양한 게임엔진들이 HMD를 활용한 가상현실 체험환경 제작을 위한 기능을 지원하여 게임은 물론 교육, 제조, 건설 등 다양한 산업 분야에 활용 가능한 가상현실 체험환경을 쉽고, 빠르게 효율적으로 제작할 수 있다.

본 연구에서는 대표적인 실시간 3차원 개발 플랫폼 게임 엔진인 유니티 3D와 언리얼 엔진 4를 활용한 가상현실 개발환경 구축에 관한 비교연구를 수행하고자 한다. 이러한 과정을 통해 가상현실 개발환경 구축에 있어 게임엔진이 가지는 특징과 개발 과정에서의 차이를 확인하고자 한다. 다음은 본 연구에서 핵심적으로 비교하고자 하는 내용을 정리한 것이다.

1. 유니티 엔진과 언리얼 엔진 4에서의 기본 가상환경 템플릿 (VR template)과 플러그인, 통합 개발환경 구축과정을 비교한다.
2. 전용 컨트롤러를 통한 사용자의 입력 및 상호작용 구조를 비교한다.
3. 오쿨러스 퀘스트 2의 손 추적기능을 통해 실제 손의 행동과 가상 객체와의 상호작용 과정을 비교 분석한다.

## 2. 관련연구

### 2.1 게임엔진 기술 및 응용

유니티 엔진[8]은 유니티 테크놀로지사에서 개발한 게임엔진으로 2차원 및 3차원 그래픽, 드래그 앤 드롭 기능 및 C# 개발 언어를 통한 스크립팅을 지원한다. 이를 활용하여 게임, 확장현실, 건축 설계, 영화, 애니메이션, 자동차, 운송, 제조 등 다양한 분야에 대한 솔루션을 제공하고 있다. 또한, 유니티 엔진은 오쿨러스, OpenXR 등 다양한 가상현실 플랫폼을 지원하여 가상현실 콘텐츠 개발을 하기에 적합한 환경을 제공한다. 이를 기반으로 유니티 엔진을 활용한 가상현실에서 사용자의 현존감 향상을 위한 HCI (Human Computer Interaction) 기술이나 새로운 체험환경을 제시하는 등 다양한 방법의 연구들이 진행되었다[9, 10, 11]. 언리얼 엔진 4[12]는 에픽게임즈에서 개발한 게임엔진으로 3차원 컴퓨터 그래픽, 드래그 앤 드롭 기능 및 C++ 개발 언어와 노드 기반 가능한 비주얼 스크립팅을 지원한다. 이를 활

용하여 게임, 영화, 애니메이션, 시뮬레이션 등 다양한 분야에 대한 솔루션을 제공하고 있다. 언리얼 엔진 4또한 오쿨러스, 마이크로소프트 홀로렌즈 (Microsoft HoloLens), HTC 바이브 (Vive) 등 다양한 확장현실 플랫폼을 지원하여 가상 현실 콘텐츠 개발에 적합한 환경을 제공한다. Ren et al. [13]은 언리얼 엔진 4기반의 몰입형 상호작용 시스템 설계에 관한 연구를 수행하였고, Liang et al. [14]은 언리얼 엔진 4를 활용한 가상현실 기반으로 교육형 시뮬레이션 시스템 설계에 관한 연구도 진행하였다. 이처럼 게임엔진은 게임 개발뿐만 아니라 건축, 인공지능, 제조, 공공 계획, 영화 및 TV 제작과 같은 게임 이외의 영역에서도 그 기능을 돋보이고 있어 학계와 대중의 관심을 요구하는 중요한 새로운 유형의 플랫폼으로 떠오르고 있다[15].

게임엔진을 활용하여 좋은 품질의 콘텐츠를 제작하기 위해서는 게임엔진에 관한 연구가 바탕이 되어야 한다. Lee et al. [16]은 유니티 엔진과 언리얼 엔진 4의 구조와 구현 방식을 FPS (first-person shooter) 게임 기반으로 비교하여 각 게임엔진 프로세스의 구성과 이벤트 처리가 어떤 차이점을 갖는지 비교하였고, Cho and Shin [17]은 효율적인 대학교육과정을 위해 유니티 엔진과 언리얼 엔진 4의 특성과 장단점을 비교 분석하기도 하였다.

## 2.2 가상현실

사용자가 가상환경에 대한 만족도를 높이기 위해선 가상환경에 몰입할 수 있어야 하고, 이를 위해 사용자가 현실 세계와 가장 유사한 감각을 받을 수 있는 환경이 마련되어야 한다. 현재 가상현실 체험환경은 가상현실 HMD를 기반으로 사용자의 시각적 몰입을 높이고 전용 컨트롤러 또는 손을 활용하여 가상 환경과 다양한 상호작용을 수행할 수 있다. 최근에는 수평면뿐만 아니라 사용자의 위아래 영역까지 고려한 현실과 더 유사한 서라운드 사운드 기술인 엠비소닉을 활용한 가상환경에 관한 연구[18], 가상환경에서 객체를 제어할 때 다양한 온도와 재질을 고려하여 사용자에게 열 피드백을 제공하는 장갑[19] 등 햅틱 시스템에 관한 연구를 통하여 사용자에게 더 큰 현존감 제공을 기대하고 있다. 이처럼 현실에서 느끼는 감각에 버금가는 사용자의 오감에 대한 가상현실 기술이 계속 연구되고 있다.

이러한 가상현실 연구, 기술 등을 토대로 직관적인 가상현실 기술뿐만 아니라 몰입형 가상현실을 활용한 메타버스 공학교육 콘텐츠를 제안하고 사용자 대상으로 긍정적인 답변을 얻기도 하였다[20]. 또한, 가상현실 사용자가 가상환경이나 또 다른 사용자와의 협업을 바탕으로 느끼는 현존감과 사회적 현존감 향상에 대한 연구도 진행되고 있다. Cho et al. [9]은 모바일 장비와 이미지 타겟을 활용한 증강현실 사용자와 HMD 기반의 몰입형 가상현실 사용자가 함께 협업하는 비대칭 가상환경에서 각 사용자의 플랫폼을 고려한 역

할을 부여하여 새로운 실감형 체험환경과 함께 현존감, 경험 등을 비교 분석하였고, Cho et al. [10]은 비대칭 가상환경에서 각 사용자의 상호작용 및 인터페이스 구조에 CNN (Convolutional Neural Networks) 기반 딥러닝을 적용하여 직관적인 상호작용이 가능한 가상환경을 제안하였다.

이처럼 게임엔진을 활용하여 게임 방식의 대화형 콘텐츠를 제작하는 과정을 비교하거나 게임엔진을 기반으로 사용자 중심의 몰입형 체험환경을 구축하는 연구들은 활발히 진행되는 반면 실감형 콘텐츠 제작에 많이 사용되는 유니티나 언리얼과 같은 대표적인 게임엔진의 개발환경 구축에 관한 내용을 비교 분석하는 연구들은 부족한 상황이다. 따라서 본 연구는 디지털 콘텐츠는 물론 가상현실, 증강현실 기술들을 활용한 실감형 콘텐츠 제작에도 많이 활용되고 있는 게임엔진인 유니티 엔진과 언리얼 엔진 4를 토대로 엔진별 가상현실 개발환경 구축과정과 핵심 기능 및 차이점 등을 비교 분석하고자 한다.

## 3. 가상현실 개발환경

### 3.1 개발환경 구성

유니티 엔진은 사용자가 가상현실 체험환경을 구축하기 위한 최소한의 기능이 포함된 가상현실 템플릿(Template)을 제공한다. 유니티 엔진의 가상현실 템플릿은 OpenXR과 오쿨러스 플랫폼을 지원하고 기본 장면(Scene)에 조명과 함께 가상현실 사용자 제어 객체(XRRig)가 기본으로 배치되어 있다. 가상현실 사용자 제어객체의 하위 객체로 가상현실 HMD와 연동하는 카메라(CameraOffset)와 전용 컨트롤러의 위치를 추적하는 객체(Left/Right Controller) 및 설정 객체(XRPlatformControllerSetup)가 포함되어 HMD를 기준으로 전용 컨트롤러의 위치 값과 회전 값을 계산하여 가상세계의 장면을 현실 세계에 정확하게 렌더링하고 가상환경과 상호작용할 수 있는 수단을 제공한다.

가상현실 템플릿 프로젝트는 가상현실 개발에 필요한 패키지 (헤드셋, 컨트롤러 추적 등)를 사전 설치하며, 오쿨러스 HMD (리프트, 퀘스트 등) 및 윈도우 혼합현실 헤드셋과 같은 장비를 지원한다. 오쿨러스 HMD를 활용한 독립된 개발환경 구축을 위해 오쿨러스 개발 지원 패키지(Oculus Integration)를 활용하면 효율적으로 오쿨러스 기반 가상현실 체험환경을 구축할 수 있다. 이는 유니티 엔진의 에셋스토어를 통해 쉽게 추가할 수 있고 전용 컨트롤러모델뿐만 아니라 가상현실 장치에 대한 고급 렌더링, 소셜, 플랫폼, 오디오 및 아바타 개발 지원에 대한 기능도 활용할 수 있다. 오쿨러스 개발 지원 패키지는 가상환경에서 HMD 기반의 사용자 시점과 전용 컨트롤러의 위치를 추적하는 가상현실 카메라 객체(OVRCameraRig)를 유니티 엔진의 프리팹(prefab) 기능을 활용하여 제공한다. 가상현실 카메라 객체



는 HMD 위치 정보를 추적하고 이를 사용자의 각 눈에 대응되는 카메라로 전달하는 카메라 객체 (Left/Center/RightEyeAnchor)와 전용 컨트롤러의 위치, 방향 정보를 추적하기 위한 앵커 (Left/RightHandAnchor)가 있다. 그리고 앵커의 하위 객체인 컨트롤러 앵커 (Left/RightControllerAnchor)에 컨트롤러 모델 또는 사용자가 정의한 객체를 하위 객체로 등록하면 추적된 컨트롤러의 위치, 방향에 대응되어 표현된다. 오쿨러스 퀘스트 2의 경우 손 추적기능이 탑재되어 컨트롤러가 아닌 손 추적으로 모드를 변경하게 되면 앵커에 추적된 손의 위치, 방향 정보가 대응되게 된다. 추가로, 가상현실 카메라 객체를 포함하면서 사용자가 플레이어 (또는 캐릭터) 역할을 수행하는 사용자 제어 객체(OVRPlayerController)도 프리팹으로 제공한다. 사용자 제어객체는 기본적으로 컨트롤러의 조이스틱 입력을 활용하여 사용자의 기준 축(ForwardDirection)을 기반으로 유니티 엔진의 캐릭터 제어 속성(Character Controller)를 활용하여 이동하는 기능이 포함되어 있다.

Figure 1은 유니티 엔진을 활용하여 가상현실 템플릿 프로젝트를 생성하고 오쿨러스 개발 지원 패키지를 추가하여 오쿨러스 기반 가상현실 개발환경 구축에 대한 과정을 보인다. 특히, Figure 1(c)에 보이듯이 유니티 엔진의 기본 가상현실 템플릿에서 제공하는 사용자 제어객체와 오쿨러스 개발 지원 패키지의 가상현실 카메라 객체의 구성 차이를 Table 1을 통해 확인할 수 있다. 각 구성 환경에서 가상현실 사용자를 위한 제어객체로 XRRig, OVRCameraRig (또는 OVRPlayerController)를 최상위 객체로 활용하고 있다. 가상현실 템플릿 사용자 객체의 카메라 역할은 Main Camera를 활용하고 오쿨러스 개발 지원 패키지의 사용자 객체는 CenterEyeAnchor 에 카메라 컴포넌트가 추가되어있다. 또한, 사용자의 키를 고려한 카메라의 초기 높이 설정을 위해서 기본 가상현실 템플릿에서는 Camera Offset 속성의 Y 값(Camera Y Offset)을 변경하고 오쿨러스 개발 지원 패키지에서는 OVRCameraRig의 OVRManager 속성의 추적 기준(Tracking Origin Type) 변경하여 설정한다. 마지막으로 컨트롤러에 대한 설정은 각 템플릿에서 제공하는 컨트롤러 객체(XRControllerRight/Left와 OVRControllerPrefab)를 활용한다.

	VR Template	Oculus Integration
<b>VR Player Object or Prefab</b>	XRRig	OVRCameraRig or OVRPlayerController
<b>Camera Setup for VR</b>	Main Camera	CenterEyeAnchor
<b>VR Camera Origin Setup</b>	Camera Offset	OVRManager
<b>Controller Object Registration</b>	XRControllerRight/Left	OVRControllerPrefab

Table 1. Comparison of environments of virtual reality (VR) templates and Oculus integration in Unity.

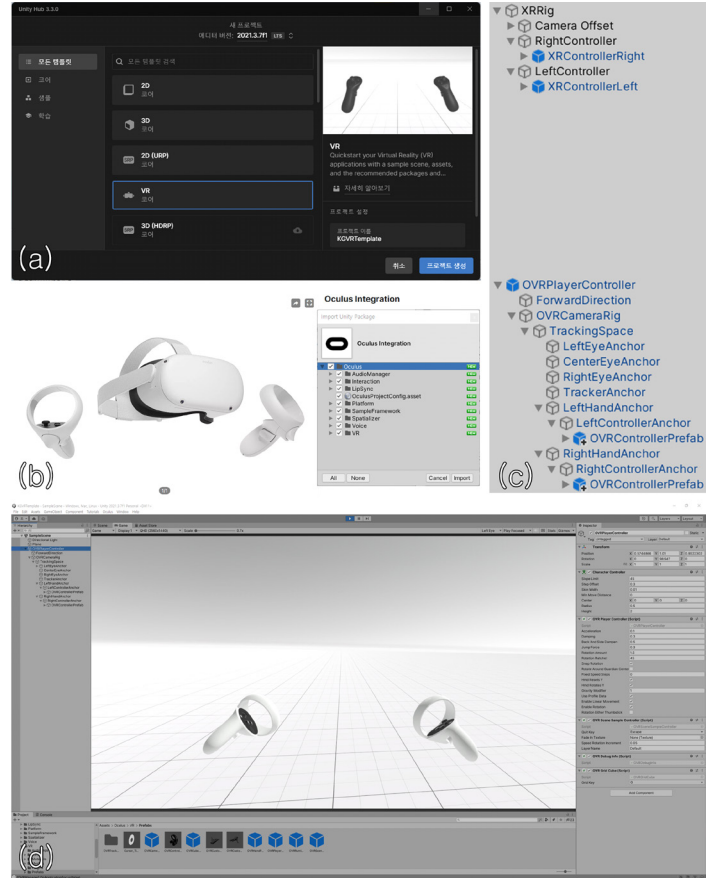


Figure 1. The process of establishing an Oculus-based virtual environment using Unity Engine: (a) Creating a VR template (b) Import Oculus Integration package (c) Configuring XRRig and OVRPlayerController objects (d) Base for VR development environment in Unity.

언리얼 엔진 4에서 제공하는 가상현실 템플릿을 활용하면 유니티 엔진과 마찬가지로 사용자가 가상현실 체험이 가능한 기능이 포함된 프로젝트를 생성할 수 있다. 기본적으로 OpenXR 과 오쿨러스 OpenXR 플러그인을 지원하여 추가 작업 없이 HMD와 전용 컨트롤러를 활용한 가상현실 환경을 체험할 수 있다. 오쿨러스 OpenXR 플러그인은 OpenXR 에서 오쿨러스 헤드셋과의 호환성을 지원하는 OpenXR 기반 플러그인이다. 만약 사용자가 전용 컨트롤러 대신 실체 손을 활용하여 가상 환경과 상호작용하는 체계를 구축한다면 오쿨러스에서 제공하는 손 모델을 사용하기 위해 오쿨러스 가상현실(OculusVR) 플러그인을 추가하면 된다. 이때 기존에 추가되어있던 오쿨러스 OpenXR 플러그인은 해제하여 두 플러그인 간의 충돌을 방지한다.

언리얼 엔진 4의 가상현실 템플릿의 기본 장면(VRTemplateMap)은 사용자 제어 객체(VRPawn)을 제공한다. 사용자 제어객체는 카메라와 좌우 컨트롤러 객체를 포함하고 있고 사용자의 카메라 초기 높낮이 위치를 설정하는 기능(SetTrackingOrigin)과 컨트롤러의 입력 버튼을 활용한 객

체 제어 기능이 있다. 또한, 가상현실 사용자가 가상환경 또는 객체 등과 상호작용 할 수 있는 블루프린트 객체(GrabComponent), 가상환경에서 3차원 UI(User Interface)를 활용할 수 있는 블루프린트 객체(WidgetMenu), 가상환경에서 사용자 제어객체를 위한 순간이동 기능을 제공하는 블루프린트 객체(VRTeleportVisualizer) 등 다양한 기능을 제공한다. 언리얼 엔진 4의 실행 구조는 가상현실 템플릿의 게임 모드(VRGameMode)에서 디폴트 폰 클래스가 사용자 제어객체로 설정되어 있어 게임이 실행되었을 때 최초로 사용자 제어객체를 생성한다. Figure 2는 언리얼 엔진 4를 활용하여 가상현실 템플릿 프로젝트를 생성하고 오쿨러스 가상현실 플러그인을 추가하여 오쿨러스 기반 가상현실 개발환경 구축에 대한 과정을 보인다.



Figure 2. The process of establishing an Oculus-based virtual environment using Unreal Engine 4: (a) Creating a VR Template (b) Add Oculus VR plug-in (c) Base for VR development environment in Unreal Engine 4.

### 3.2 상호작용: 전용 컨트롤러

가상현실 사용자가 가상 환경과 상호작용하는 기본적인 방법은 가상현실 HMD와 함께 제공되는 전용 컨트롤러를 활용하는 것이다. 전용 컨트롤러는 사용자의 각 손에 하나씩

(왼손, 오른손) 쌍으로 구성되어 있으며 이를 쥐어 사용자의 손을 대체하는 수단으로 활용할 수 있다. 가상환경에서 HMD는 전용 컨트롤러의 위치를 추적하고 이를 기반으로 사용자는 가상 환경과 상호작용이 가능하다. 또한, 컨트롤러의 버튼 입력과 조이스틱을 활용하여 가상환경에서의 다양한 상호작용을 수행할 수 있다. 본 연구는 가상현실 개발환경 구축에 있어 오쿨러스 퀘스트 2 장비를 활용함을 전제로 하고 있어 전용 컨트롤러로 터치 컨트롤러를 활용한다.

유니티 엔진의 경우, 오쿨러스 통합 개발환경 패키지의 전용 컨트롤러객체(OVRControllerPrefab)를 사용자 제어객체의 컨트롤러 앵커(Left/RightControllerAnchor)의 하위 객체로 등록하여 가상환경에서 상호작용이 가능하다. 전용 컨트롤러에 대한 입력과 버튼, 트리거 및 조이스틱 입력에 따른 상호작용 처리는 유니티 엔진의 C# 스크립트에서 오쿨러스 입력 클래스(OVRInput)를 활용하여 구현한다. 또한, 오쿨러스 통합 개발환경 패키지는 객체 잡기를 통한 상호작용기능(OVRGrabber, OVRGrabbable)을 제공한다. Figure 3은 유니티 엔진의 개발환경에서 컨트롤러 입력으로 가상환경에서 객체와 상호작용하는 과정과 처리 구조를 보인다.

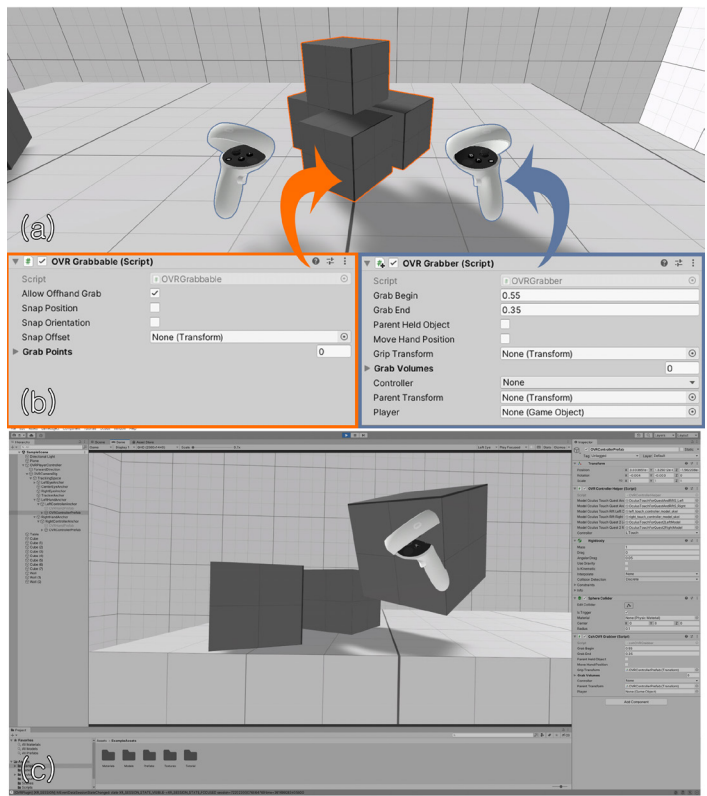


Figure 3. Process of interaction development environment using controller in the Unity Engine: (a) Grabber controller and grabbable Objects (b) Components for user interaction with objects (c) Interaction with objects using controller inputs in a virtual environment based on Unity Engine.

언리얼 엔진 4에서는 기본 사용자 제어객체에 오쿨러스 전용 컨트롤러객체가 포함되어 있다. 전용 컨트롤러의 버튼 입력에 따른 상호작용 처리는 프로젝트 설정의 입력 바인딩에서 정의하고, 블루프린트 클래스에서 입력 액션을 호출하여 구현한다. 사용자 제어객체의 기본 제공 블루프린트 함수로 사용자 제어객체의 이동과 방향 전환 기능으로 좌우컨트롤러의 각 조이패드를 활용하여 축 입력에 따른 카메라 방향 전환기능(Snap Turn, AddActorWorldRotation)과 순간 이동 기능(Teleport, SetActorLocation)을 제공하고 사용자 제어객체의 컨트롤러(손)를 활용하여 가상환경에서 객체를 잡고 놓는 블루프린트 기반 함수(TryGrab/Release)를 제공한다. Figure 4는 언리얼 엔진 4 개발환경에서 컨트롤러 버튼 입력으로 가상환경의 객체와 상호작용하는 과정을 보인다.



Figure 4. Process of interaction development environment using controller in the Unreal Engine 4: (a) Setting up a scene for the grab process (b) Blueprint of the process of grabbing or releasing objects (c) Interaction with objects using controller inputs with Objects in a virtual environment based on Unreal Engine 4.

### 3.3 상호작용: 손 추적기능 응용

실제 환경에서 손으로 직접 물건을 만지거나 도구를 잡아 행동을 수행하는 등 손은 상호작용에 있어 가장 대표적인 신체 부위이다. 따라서, 가상세계에서도 가상환경 또는 객체와의 상호작용에 손을 직접 활용하는 것이 가장 몰입감 높은 체험환경을 제시하는 방법이 된다. 오쿨러스 퀘스트 2 HMD는 사용자의 실제 손의 위치, 방향 정보를 추적하여 가상환경의 손으로 대응하여 렌더링할 수 있는 기능을 제공한다. 이를 활용하여 사용자는 전용 컨트롤러 없이 HMD만으로도 가상 환경과 직접적인 상호작용을 수행할 수 있는 기반이 마련된다.

유니티 엔진은 오쿨러스 개발 지원 패키지를 통해 사용자의 실제 손 위치, 방향 정보를 기반으로 가상환경에 대응시켜주는 가상 손 객체(OVRHandPrefab)를 프리팹으로 제공하고 있어 컨트롤러 대신 활용할 수 있다. 가상 손 객체에는 손의 관절정보가 포함되어 현재 손의 움직임(애니메이션 상태정보)을 알 수 있고 이를 기반으로 현실적인 상호작용을 수행할 수 있다.

언리얼 엔진 4에서 제공하는 오쿨러스 퀘스트 2의 손 추적기능을 기반으로 생성된 기본 가상의 손 모델은 에디터 상에서 접근 가능한 손 관절구조의 정보가 포함되어 있지 않다. 따라서 오쿨러스의 공식 사이트에서 제공하는 손 객체(Handsample)를 프로젝트에 포함시키고 가상 손의 골격메시(Skeletal Mesh)를 오쿨러스 핸드(OculusHand\_L/R)로 변경하여 관절정보가 포함된 가상 손을 활용한다. 이후 C++나 블루프린트 스크립팅을 통해 가상 손의 관절 위치 정보를 바탕으로 사용자가 가상환경에서 가상의 손을 활용한 상호작용을 수행할 수 있다. Algorithm 1은 사용자의 실제 손의 움직임에 대응하여 동작하는 가상 손의 관절정보에 따른 가상 객체 잡기 행동 구현 과정을 나타낸다. 또한, Figure 5는 유니티 엔진과 언리얼 엔진 4에서 오쿨러스 퀘스트 2의 손 추적기능을 기반으로 사용자가 실제 손을 활용하여 가상 환경에서 가상 손의 관절정보에 따른 가상환경 속 객체와 상호작용하는 과정을 보인다.

**Algorithm 1.** The process of implementing grab behavior based on virtual hand joint information

1. **procedure** Virtual Hand Grab based Real Hand
2.  $t[i]$ , for  $i = 0, \dots, 4 \leftarrow$  find fingertips by name.
3.  $d[i] \leftarrow$  distance between  $i$ -th fingertip and wrist.
4.  $avg_d \leftarrow$  the average of  $d[i]$  array.
5.  $t_g \leftarrow$  grab threshold value
6. **if** ( $avg_d < t_g$ ) **then**
7.     call Grab method
8. **else**
9.     call Release method
10. **end if**
11. **end procedure**



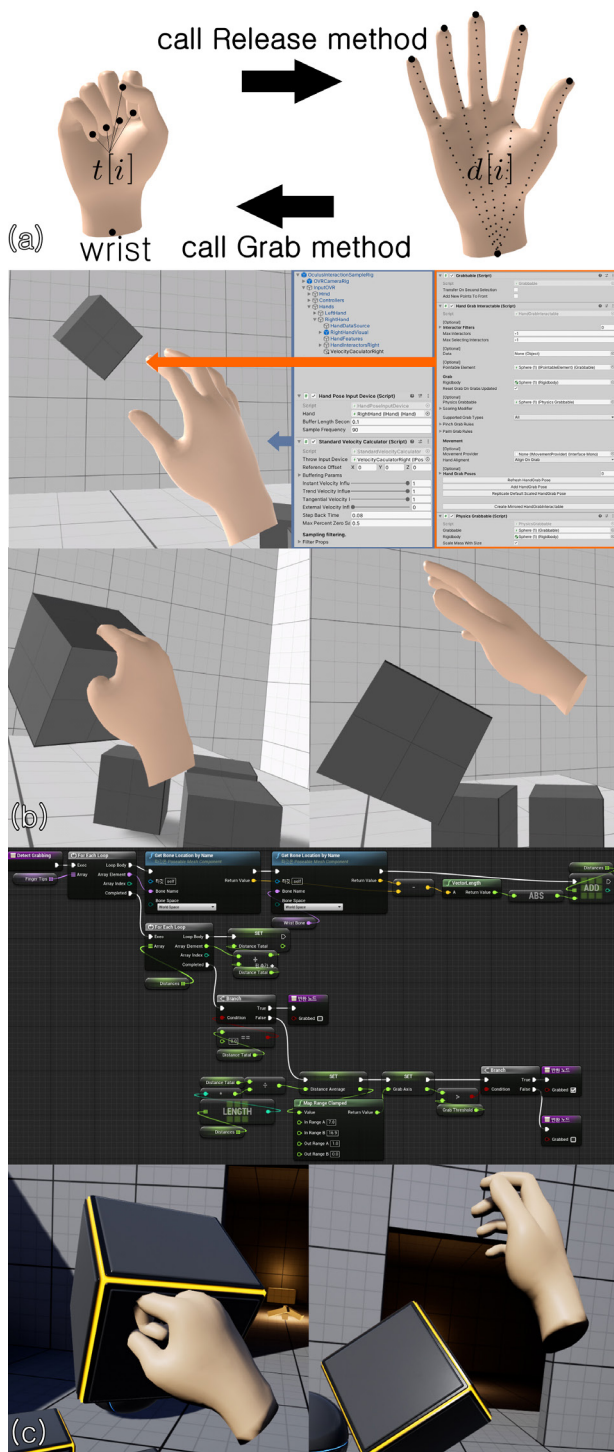


Figure 5. Examples of hand-based grab processing on Unity Engine and Unreal Engine 4: (a) Wrist and fingertip points in a virtual environment, the distance between each fingertip and wrist is shown on the right. (b) Interaction process with virtual environment according to joint position information of hand in Unity engine (Top: applying a physical inertial script using OculusInteractionSampleRig). (c) Interaction process with virtual environment according to joint position information of hand in Unreal engine 4. (Top: Blueprint Scripting Based on Algorithm 1)

유니티 엔진의 경우, 객체 잡기를 위한 상호작용기능 (OVRGrabber, OVRGrabbable)을 상속받아 가상 객체 잡기 행동을 구현하는 기본적인 방법 외에도 가상현실 카메라와 상호작용 기능이 포함된 객체(OculusInteractionSampleRig)를 추가로 제공한다. 이는 손과 컨트롤러를 통합 입력 구조를 보다 체계적으로 나눠 기능을 제공하고 있으며 특히, 손을 직접 활용하는 상호작용과 관련하여 상호작용을 위한 인터랙터(HandInteractor)와 함께 손을 활용한 행동 기능 (HandPoke(or Grab)Interactor 등), 물리 관성을 위한 스크립트 (StandardVelocityCalculator 등)를 세부적으로 제공하고 있다. 이외에도 상호작용 객체를 위한 속성과 기능 또한 다양하게 제공하여 객체를 놓을 때 물리 관성을 적용하거나 객체를 잡을 때 정교한 손 관절의 위치를 지정할 수 있어 훨씬 더 자연스러운 상호작용 제작이 가능하다.

#### 4. 실험환경 및 분석

유니티 엔진과 언리얼 엔진 4의 가상현실 체험환경 구축과정을 중심으로 비교하기 위해서 두 엔진 모두 오쿨러스 퀘스트 2 HMD와 전용 컨트롤러를 활용하여 동일한 장비를 활용하여 실감형 가상현실 체험환경을 구성하였다. 또한, 가상환경을 HMD와 전용 컨트롤러를 사용하는 경우와 HMD가 제공하는 사용자의 실제 손 추적기능을 활용하여 전용 컨트롤러와 사용자의 실제 손을 사용하는 경우 두 가지로 나누어 각각의 가상환경을 비교하였다.

유니티 엔진은 가장 최근에 정식 릴리즈로 출시된 2021.3.7.f1 버전을 활용하였고 언리얼 엔진 4 또한 가장 최신 정식 릴리즈 버전인 4.27.2를 활용하였다. 본 연구는 유니티 엔진에서는 C# 스크립트, 언리얼 엔진 4에서는 블루프린트 스크립팅을 활용하여 엔진별 가상환경에서 컨트롤러와 손을 활용한 상호작용 시스템을 구현하였다. 언리얼 엔진 4에서 C++ 스크립트를 활용하여 구현하는 방법도 있지만 본 연구는 비개발자도 쉽게 이해하고 활용할 수 있도록 비주얼 스크립팅 시스템인 블루프린트를 활용하여 구현하였다.

본 연구는 유니티 엔진과 언리얼 엔진 4 개발환경에서 가상현실 체험환경 구축을 위한 과정과 방법을 비교하기 위해 사용자가 전용 컨트롤러나 사용자의 손을 직접 활용하여 가상 객체와 상호작용하는 간단한 가상환경을 제작하였다. 특히, 실제 손을 활용하여 가상 객체와 상호작용하는 경우, 사용자 손의 관절 위치값을 고려하여 모든 손가락을 구부린 동작, 모든 손가락을 펴 동작으로 정의하고 객체를 잡고 놓는 행동을 구현하는 과정을 비교, 분석하였다.

Table 2는 유니티 엔진과 언리얼 엔진 4에서 오쿨러스 퀘스트 2 장비를 활용하여 가상현실 개발환경을 구축하는 과정에서의 기술적 내용을 비교하여 표로 정리한 것이다. 각 엔진에서 제공하는 가상현실 템플릿(VR template)을 기반으로 오쿨러스 퀘스트 2 HMD와 전용 컨트롤러를 활용하기

	Unity engine (2021.3)	Unreal engine 4
Establishment of development environment	Import Oculus Integration from asset store	Add Oculus VR plugin from project settings
Player controller	OVRPlayerController	VRPawn
Controller input	OVRInput or Input Manager	Input binding
Joint information of the virtual hand	Included	Handsample required
Physical inertia application method	Implement it yourself or use the provided script component	set in the virtual environment itself

Table 2. Comparison of the features provided by the Oculus Quest 2 based virtual environment of Unity engine and Unreal engine 4.

위한 개발환경 구축과 가상현실 사용자 객체, 전용 컨트롤러 입력을 통한 상호작용과정을 비교하였다. 또한, 오클러스 퀘스트 2가 제공하는 사용자 손 추적기능을 활용하여 가상 환경과 상호작용을 수행할 때 필요한 손 관절정보 포함 여부 등을 세부적으로 비교하였다. 우선, 개발 환경 구축에 있어 가장 큰 차이점은 유니티 엔진은 에셋스토어를 통해 오클러스 개발 지원 패키지를 외부에서 임포트시키는 방법이지만, 언리얼 엔진 4는 엔진 내부에서 설정을 통해 플러그인을 추가시키는 방법을 사용한다. 그리고, 가상현실 사용자를 하나의 플레이어(또는 캐릭터)로 하여 대화형 콘텐츠를 제작하는 과정은 유니티 엔진의 경우 캐릭터 제어 속성을 활용한 사용자 제어 객체(OVRPlayerController)를 언리얼 엔진 4는 월드 내 플레이어를 위한 제어 객체(Pawn)을 가상현실에 적합하게 수정한 방식(VRPawn)으로 구현한다. 즉, 플레이어 기능 구현을 위해 엔진에서 제공하는 속성, 객체를 가상현실 구조에 맞게 변형하여 사용하는 방식이다. 이외에도 컨트롤러에 대한 입력 방식은 유니티 엔진은 오클러스 입력 클래스를 사용하거나 엔진 자체 입력 관리자(Input Manager)를 사용하고 언리얼 엔진 4는 내부 입력 연결 구조를 활용하는 방식으로 구현한다. 이 역시도 플레이어와 같이 엔진의 자체 구조에 기반을 둔 방식으로 볼 수 있다.

손 추적기능을 통해 실제 손에 대응되는 가상 손의 움직임을 처리하는 과정에 있어서 유니티 엔진은 오클러스 통합 개발환경이 제공하는 손 객체(OVRHand)의 관절정보를 활용하여 기능을 구현하고, 언리얼 엔진 4는 기본적으로 제공하는 손 객체에는 관절정보가 포함되어 있지 않아 커스텀 골격 메시를 추가하여 활용한다. 그리고, 가상환경에서 객체를 잡았다가 놓을 때 사용자가 컨트롤러나 손을 움직이는 방향으로 자연스러운 관성 물리학 적용방식이 다른 점을 볼 수 있다. 유니티 엔진에서는 객체를 잡은 손의 움직임을 계산하고 객체를 놓는 순간의 힘을 객체에 전달하여 물리 관성을 표현하지만, 언리얼 엔진 4는 가상환경 자체에 항상 물리 관성이 적용되어 있는 구조이다. 유니티 엔진에서는 물리 관성 표현을 위한 힘을 직접 구현할 수도 있지만, 상호작용과 관련하여 제공되는 스크립트를 활용해서도 쉽게 표

현할 수 있다.

마지막으로, 유니티 엔진과 언리얼 엔진 4의 구조와 구현 방식을 비교한 기존 연구[16]와 본 연구를 비교 분석하도록 한다. Lee et al.의 비교 연구의 경우 FPS(First-Person Shooter) 게임의 구현을 기반으로 유니티 엔진과 언리얼 엔진 4에서의 사용자 객체 구성과 게임 로직 설계를 위한 상호작용 방식 및 기능을 비교하고 있다. 이는 게임을 비롯한 대화형 콘텐츠를 제작하는 과정에서 각 엔진의 기본적인 속성, 기능 및 구조를 비교하여 분석하고 있다. 본 연구는 기존 연구와 동일하게 각 엔진의 기본적인 기능과 구조를 토대로 실감형 콘텐츠 제작을 위한 가상현실 개발환경 구축과정 비교를 목표로 한다. 엔진별로 가상현실 체험환경 구성을 위한 초기 설정 과정에서부터 컨트롤러 및 손 추적기능을 활용한 객체와의 상호작용 설계 방식을 구체적으로 비교하였다.

## 5. 한계

본 연구는 유니티 엔진과 언리얼 엔진 4를 토대로 비교연구를 수행하였다. 다만, 유니티 엔진을 활용한 프로젝트 개발 경험이 언리얼 엔진 4를 활용한 프로젝트 개발 경험보다 많기 때문에 본 연구에서의 비교 분석은 유니티 엔진을 중심으로 언리얼 엔진 4를 비교하는 방식으로 연구가 수행된 점이 있다. 향후, 비교연구에는 언리얼 엔진을 더욱 깊이 있게 다뤄 각 엔진 측면에 관한 비교연구로 확장할 계획이며, 가상현실 장비 역시 오클러스 외에도 다양한 장비들을 포함한 개발환경 구축으로 발전시켜 나가고자 한다.

## 6. 결론 및 향후연구

본 연구에서는 유니티 엔진과 언리얼 엔진 4를 기반으로 오클러스 퀘스트 2 HMD와 전용 컨트롤러, 손 추적기능을 활용한 사용자의 실제 손을 활용하여 가상 환경과 상호작용이 가능한 가상체험환경을 제작하는 과정에서 각 엔진이 가상현실 구축을 위해 제공하는 기능과 개발환경에 대해 비교하였다. 먼저 유니티 엔진은 오클러스 퀘스트 2 HMD와 전용



컨트롤러를 활용하는 가상환경 개발 기반을 마련하기 위해 유니티 에셋스토어에서 제공하는 오쿨러스 통합 개발환경 (Oculus Integration) 패키지를 프로젝트에 포함하고 언리얼 엔진 4에서는 오쿨러스 가상현실(Oculus VR) 플러그인을 추가하여 오쿨러스 퀘스트 2기반 가상현실을 제작할 수 있는 환경을 마련하였다. 가상현실 체험환경은 전용 컨트롤러와 사용자의 실제 손을 활용하여 가상환경 속 객체를 잡고 놓는 방식의 간단한 체험환경을 제작하고 각 엔진에서 가상 환경과 상호작용할 수 있는 수단인 컨트롤러와 오쿨러스 퀘스트 2 HMD가 제공하는 손 추적기능을 활용하여 사용자가 실제 손을 기반으로 가상 환경과 상호작용을 수행할 수 있는 과정을 정리하였다. 이러한 내용을 토대로 본 연구는 유니티 엔진과 언리얼 엔진 4를 활용한 다양한 가상현실 콘텐츠 제작에 방향을 제시할 수 있을 것으로 기대된다.

몰입형 장비를 활용한 가상현실 체험환경은 교육, 체험, 게임 등 다양한 분야의 콘텐츠 제작이 가능하다. 또한, 최근 다수의 사용자가 함께 참여하여 다양한 경험, 활동 등을 공유하는 메타버스가 떠오르면서 메타버스 콘텐츠로서의 확장도 고려할 수 있다. 메타버스 콘텐츠를 제작하기 위해서 다른 사용자들과의 동기화를 통한 멀티플레이는 필수적인 요소이다. 유니티 엔진에서 기본적으로 제공하는 네트워킹 매니저인 UNet(Unity Networking)이나 포톤 엔진(Photon Engine)에서 제공하는 PUN(photon unity networking)과 같은 외부 네트워킹 매니저를 활용하면 다수의 사용자가 참여하는 멀티플레이 동기화 환경을 구성할 수 있다. 하지만 최근 유니티 엔진은 UNet에 대한 지원을 중단하고 새로운 네트워킹 시스템에 대해 개발하고 있어 UNet을 활용한 멀티플레이 네트워킹은 계속된 보장을 받을 수 없다. 언리얼 엔진 4는 클라이언트와 서버 간의 데이터 및 프로시저 호출을 동기화하는 네트워킹 프로세스인 복제(Replication)와 포톤이나 스템 네트워크 등 외부 플러그인을 활용한 네트워킹을 지원한다. 포톤 엔진은 유니티 엔진과 언리얼 등 다양한 플랫폼에서 포톤 클라우드 기반 실시간(realtime) 멀티플레이를 지원하는 멀티플레이어 엔진이다. 포톤 엔진의 PUN은 유니티 엔진에서 기본적인 네트워크 동기화를 무료로 지원하고 쉽게 적용하고 사용할 수 있어 많은 개발자가 유니티 엔진 환경에서 멀티플레이를 위한 네트워킹 동기화 수단으로 사용하고 있다. 언리얼 엔진 4에서는 포톤 리얼타임(Photon Realtime)을 지원하고 있어 이를 활용한 멀티플레이 메타버스 콘텐츠도 제작이 가능하다. 따라서, 향후 본 연구에서 수행한 엔진별 몰입형 가상현실 콘텐츠 제작과정에 대한 비교뿐만 아니라 증강현실이나 확장현실에 대한 제작 과정을 비교하고 메타버스가 주목받음에 따라 엔진별 지원 가능한 다양한 네트워킹 동기화 프로그램 비교를 수행하도록 한다. 또한, 유니티, 언리얼 엔진 4뿐만 아니라 크라이엔진, 소스 엔진 등 다양한 엔진을 기반으로 메타버스 기반 멀티플랫폼 가상환경 구축과정, 엔진별 성능과 함께 동일한

콘텐츠를 제작하는 과정에서 엔진별 개발환경의 차이가 작업에 미치는 영향에 관해서도 비교연구를 수행하고자 한다.

## 감사의 글

이 논문은 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. NRF-2020R1F1A1063442, 조윤식, Yunsik Cho). 또한, 본 연구는 한성대학교 학술연구비 지원과제임 (김진모, Jinmo Kim).

## References

- [1] ZEPETO app [Internet], Available: <https://play.google.com/store/apps/details?id=me.zepeto.main&hl=ko&gl=US>
- [2] 이동현, “순천향대, ‘2022 메타버스 입학식’ 개최,” *중앙일보*, [Internet] Available: <https://www.joongang.co.kr/article/25052623>, 2022.
- [3] Seyoung Choi, Sujin Kim, Nayoung Lee, Kihan Lee, Hyeyoung Ko. “Development of VR Healing Content ‘NORNIR’ Using Color Therapy,” *Journal of the Korea Computer Graphics Society*, vol. 26, no. 3, pp. 143-153, 2020.
- [4] Seunghyun Hong, Giri Na, Yunsik Cho, Jinmo Kim. “A Study on Movement Interface in Mobile Virtual Reality,” *Journal of the Korea Computer Graphics Society*, vol. 27, no. 3, pp. 55-63, 2022.
- [5] Yesol Cho, Yewon Lee, Dojeon Lim, Taedong Ryoo, Jonas John Claud, Daeyoung Na, Daseong Han. “The Effect of Data-Guided Artificial Wind in a Yacht VR Experience on Positive Affect,” *Journal of the Korea Computer Graphics Society*, vol. 28, no. 3, pp. 67-77, 2022.
- [6] Woohee Park, Jieun Kim, Jieun Lee. “A Study on the Design and Effect of Feedback for Virtual Reality Exercise Posture Training,” *Journal of the Korea Computer Graphics Society*, vol. 26, no. 3, pp. 79-86, 2020.
- [7] 윤상은, “작년 AR·VR 헤드셋 출하량 전년比 두 배 증가,” *ZDNET Korea*, [Internet] Available: <https://zdnet.co.kr/view/?no=20220323142242>, 2022.
- [8] Unity engine, UnityTechnologies, [Internet] Available: <https://unity.com/>, 2021.
- [9] Yunsik Cho, Jiewon Kang, Jaekyung Jeon, Jongchan Park, Mingyu Kim, and Jinmo Kim, “X-person asymmetric interaction in virtual and augmented realities,” *Computer Animation & Virtual Worlds*, vol. 32, no. 5, e1985, 2021x.
- [10] Yunsik Cho, Seunghyun Hong, Mingyu Kim, and Jinmo Kim. “DAVE: Deep Learning-Based Asymmetric Virtual

- Environment for Immersive Experiential Metaverse Content,” *Electronics*, vol. 11, no. 16, 2604, 2022.
- [11] Jinmo Kim, “VIVR: Presence of Immersive Interaction for Visual Impairment Virtual Reality,” *IEEE Access*, vol. 8, pp. 196151-196159, 2020.
- [12] Unreal Engine 4, Epicgames, [Internet] Available: <https://www.unrealengine.com/ko> 2021.
- [13] Ke Ren, Huaqun Liu, Mingyu Zhang, Xiaofeng Qiu, Huimin Yan, and Jinbo Zhang, “Design and Implementation of Immersive and Interactive System Based on Unreal Engine,” In *2021 5th International Conference on Artificial Intelligence and Virtual Reality (AIVR) (AIVR 2021)*, Association for Computing Machinery, New York, NY, USA, pp. 77–82, 2021.
- [14] Zhanti Liang, Yongkang Xing, Kexin Guan, Zheng Da, Jianwen Fan, and Gan Wu, “Design Virtual Reality Simulation System for Epidemic (Covid-19) Education to Public,” In *2021 4th International Conference on Control and Computer Vision (ICCCV'21)*, Association for Computing Machinery, New York, NY, USA, pp. 147–152, 2021.
- [15] Andreas Jungherr, and Damien Schlarb, “The extended reach of game engine companies: How companies like Epic Games and Unity Technologies provide platforms for extended reality applications and the metaverse,” *Social Media + Society*, vol. 8, no. 2, pp. 1-12, 2022.
- [16] HanSeong Lee, SeungTaek Ryoo, and SangHyun Seo. “A Comparative Study on the Structure and Implementation of Unity and Unreal Engine 4.” *Journal of the Korea Computer Graphics Society*, vol. 25, no. 4, pp. 17-24, 2019.
- [17] Hyung-Ik Cho, and Seung Jung Shin, “A Study on the Characteristics of Game Engines for Next-Generation Multimedia Production Education (Focusing on Unity and Unreal Game Engine),” *Journal of The Korean Society for Computer Game*, vol. 34, no. 3, pp. 29-38, 2021.
- [18] Aakanksha Rana, Cagri Ozcinar, and Aljosa Smolic. “Towards Generating Ambisonics Using Audio-visual Cue for Virtual Reality,” *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2019)*, pp. 2012-2016, 2019.
- [19] Shaoyu Cai, Pingchuan Ke, Takuji Narumi, and Kening Zhu, “ThermAirGlove: A Pneumatic Glove for Thermal Perception and Material Identification in Virtual Reality,” *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 248-257, 2020.
- [20] Yongsun Lee, and Taek-hee Lee, “Applying the Metaverse Platform and Contents in Practical Engineering Education,” *Journal of the Korea Computer Graphics Society*, vol. 28, no. 3, pp. 31-43, 2022.

## 〈 저 자 소 개 〉



### 조 윤 식

- 2021년 한성대학교 컴퓨터공학부 학사
- 2022년 한성대학교 일반대학원 컴퓨터공학과 석사
- 2022년~현재 한성대학교 일반대학원 정보컴퓨터공학과 박사과정
- 관심분야: 가상현실, 증강현실, HCI 등
- <https://orcid.org/0000-0003-2118-0904>



### 김 진 모

- 2006년 동국대학교 멀티미디어학과 학사
- 2008년 동국대학교 영상대학원 멀티미디어학과 석사
- 2012년 동국대학교 영상대학원 멀티미디어학과 박사
- 2012년~2014년 동국대학교 영상문화콘텐츠연구원 전임연구원
- 2014년~2019년 부산가톨릭대학교 소프트웨어학과 조교수
- 2019년~현재 한성대학교 컴퓨터공학부 조교수
- 관심분야: 컴퓨터그래픽스, 가상현실, 증강현실, 메타버스, 게임 공학 등
- <https://orcid.org/0000-0002-1663-9306>