

제페토 월드를 활용한 메타버스 콘텐츠 제작 공정에 관한 연구

박명석^{0,1} 조윤식¹ 조다솜¹ 나기리² 이자민² 조세홍² 김진모^{1,2,*}

한성대학교 일반대학원 컴퓨터공학과¹, 한성대학교 컴퓨터공학부²

ilsuntwo@naver.com, whdbstlrwkd@hansung.ac.kr, joda0604@gmail.com, {nagiri0209, jamin_00, chosh, jinmo.kim}@hansung.ac.kr

A Study on The Metaverse Content Production Pipeline using ZEPETO World

MyeongSeok Park^{0,1} Yunsik Cho¹ Dasom Cho¹ Giri Na² Jamin Lee²

Sae-Hong Cho² Jinmo Kim^{1,2,*}

Department of Computer Engineering, Graduate School, Hansung University¹,

Division of Computer Engineering, Hansung University²

요 약

본 연구는 대표적인 국내 메타버스 플랫폼의 대표 중 하나인 제페토 월드를 활용한 메타버스 콘텐츠 제작 공정을 제안한다. 유니티 3D 엔진을 기반으로 제페토 템플릿을 활용하여 제페토 월드를 구성하고, 제페토 스크립트를 통해 로직, 상호작용, 속성 제어 등 다중 사용자 참여가 가능한 메타버스 콘텐츠의 핵심 기능들을 구현한다. 본 연구는 제페토 스크립트의 속성, 이벤트, 컴포넌트 등의 기본 기능과 아바타 로딩, 캐릭터 이동, 카메라 제어 기능을 포함하는 제페토 플레이어를 활용한다. 그리고 제페토의 월드 멀티플레이(WorldMultiplay), 클라이언트 스타터(ClientStarter) 등의 속성을 토대로 객체 변환, 동적 객체 생성 그리고 속성 추가 및 실시간 속성 제어 등 멀티플레이 메타버스 콘텐츠 제작에서 필요한 핵심적인 동기화 처리 과정을 정리한다. 이를 기반으로 제페토 월드를 활용한 멀티플레이 메타버스 콘텐츠를 직접 제작함으로써 제안하는 제작 공정을 확인한다.

Abstract

This study proposes the metaverse content production pipeline using ZEPETO World, one of the representative metaverse platforms in Korea. Based on the Unity 3D engine, the ZEPETO world is configured using the ZEPETO template, and the core functions of the metaverse content that enable multi-user participation such as logic, interaction, and property control are implemented through the ZEPETO script. This study utilizes the basic functions such as properties, events, and components of the ZEPETO script as well as the ZEPETO player which includes avatar loading, character movement, and camera control functions. In addition, based on ZEPETO's properties such as World Multiplayer and Client Starter, it summarizes the core synchronization process required for multiplayer metaverse content production, such as object transformation, dynamic object creation, property addition, and real-time property control. Based on this, we check the proposed production pipeline by directly producing multiplayer metaverse content using ZEPETO World.

키워드: 메타버스 콘텐츠, 제페토, 유니티 3D 엔진, 제작 공정, 타입스크립트

Keywords: metaverse contents, ZEPETO, Unity 3D engine, production pipeline, TypeScript

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

1. 서론

메타버스(metaverse)는 닐 스티븐슨의 SF 소설 스노우 크래시(Snow Crash)[1]에서 유래한 것으로 초월을 의미하는 메타(meta)와 세계, 우주를 의미하는 유니버스(universe) 단어를 조합한 물리적 세계와 연결된 가상의 확장된 가상 세계를 나타낸다. 메타버스 개념의 처음 등장 이후로 컴퓨터로 생성된 우주로서 라이프로그, 미러월드(mirror world), 다중우주(omniverse) 등 매우 다양한 개념을 통해 정의되어왔다[2]. 그리고 정치, 경제, 사회, 문화 전반적으로 현실과 가상이 공존하는 가상 세계의 의미로 폭넓게 활용된다.

메타버스 콘텐츠 제작 및 응용과 관련된 연구들로 사회적 선(social good)을 위한 메타버스로 사회기반 시설(infrastructure), 상호작용, 생태계를 포함하는 거시적 관점에서의 메타버스 구조를 제안하고 대학 캠퍼스 프로토타입을 통해 고찰하는 연구를 수행하였다[3]. 또한, 사용자가 직접 또는 간접적으로 문화 콘텐츠를 경험할 수 있는 메타버스 문화 콘텐츠 매칭 플랫폼을 제안[4]하거나 메타버스 세계에서 대학생들이 효율적으로 친구를 사귄 수 있는 비대면 친구 사귀기 시스템을 디자인 및 구현하였다[5]. 이외에도 공간감, 상호작용, 아바타 그리고 경제 시스템의 4가지 요소를 적용한 국립민속박물관의 메타버스 콘텐츠 개발 연구를 진행하기도 하였다[6]. 최근에는 메타버스에 존재하는 사용자가 실제 공간에 있는 듯한 몰입감, 현존감을 제공할 수 있는 체험 환경을 구축하기 위하여 가상현실, 증강현실 등을 아우르는 확장현실에 대한 기술이 중요하게 인식되어 핵심 기술 중 하나로 활용되고 있다. 메타버스에서 가상현실 서비스를 위한 인센티브 메커니즘 프레임워크를 제안하였다[7]. 또한, 실감형 기술과 더불어 건축 교육에서 메타버스 가상환경을 사용하여 건축 전공 학생과 교수 및 강사 사이의 상호작용 관계를 제안[8]하거나 메타버스 환경에서 쇼핑과 같은 일상적인 작업 수행에 있어 가상현실과 증강현실 등의 응용 기술이 업무 수행에 어려움을 증가 또는 감소 시키는지에 대한 연구를 수행하였다[9]. 이밖에도 디지털 트윈(digital twins)과 가상 인간(virtual human)의 개발에서 언어와 의미 기술을 지원하는 유연한 메타버스 AI 기술 프레임워크인 metaAID를 제안하기도 하였다[10]. 이처럼 메타버스 콘텐츠 응용 연구를 비롯하여 가상환경에서 사용자 중심의 인터페이스 및 상호작용에 관한 연구들이 교육, 패션 등 다양한 분야에서 진행되고 있다[11, 12, 13, 14]. 그뿐만 아니라, 교육공간으로서 메타버스의 활용 가능성 또는 한계를 고찰[15]하거나 메타버스 구성원리[16], 메타버스 제페토 플랫폼을 이용하는 10대, 20대 사용자의 경험[17] 등을 연구하는 물론 다양한 사례 연구[18] 또한 진행되고 있다.

2006년에 출시되어 사용자가 게임을 직접 프로그래밍하고 이를 다른 사용자와 함께 즐길 수 있도록 구현된 로블

록스(Roblox)는 해외의 대표적인 메타버스 플랫폼 중 하나이다. 이는 1억 6,000만 명 이상의 사용자를 보유하고 있으며 현재까지도 많은 인기를 얻고 있다. 국내의 경우는 네이버Z의 제페토와 SK 텔레콤의 이프랜드(ifland)가 있다. 특히, 제페토의 경우 자신만의 3차원 아바타를 생성하고 다양한 가상 활동을 할 수 있다(Figure 1). 그뿐만 아니라, 아바타 의상부터 3차원 월드까지 직접 만들고 다른 사용자와 공유할 수 있는 것은 물론 제페토 템플릿을 통해 직접 메타버스 콘텐츠를 개발하고 서비스할 수 있는 환경을 제공하고 있다[19].

본 연구는 유니티 3D 엔진을 기반으로 제페토 템플릿과 제페토 스크립트를 통해 멀티플레이 메타버스 콘텐츠 제작에 필요한 핵심적인 기술들을 정리하고 이를 기반으로 제작 공정을 제안한다. 그리고 이를 바탕으로 제페토 월드를 활용한 멀티플레이 메타버스 콘텐츠를 직접 제작해 봄으로써 제안하는 제작 공정을 확인하고자 한다.

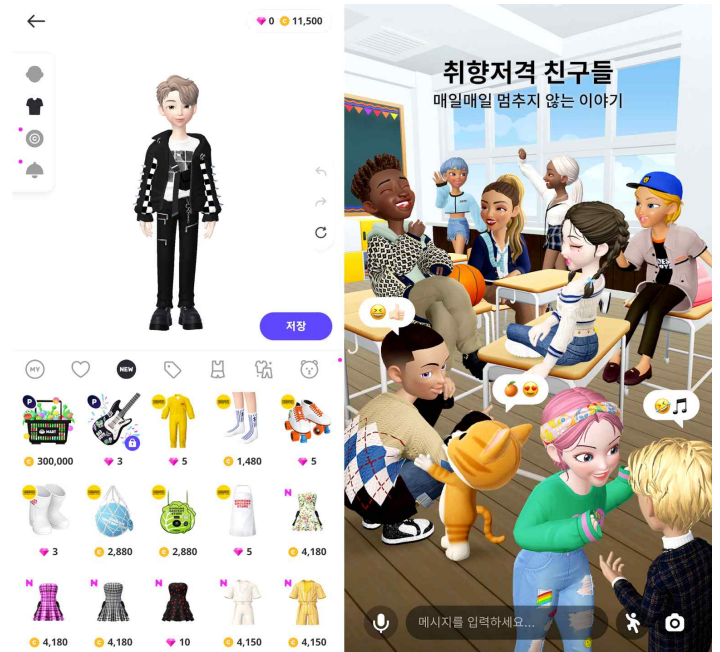


Figure 1. Example of Metaverse Platform ZEPETO Application[20].

2. 제페토 월드를 활용한 제작 공정

제페토 월드를 활용한 메타버스 콘텐츠는 유니티 3D 엔진을 기반으로 제작한다. 제페토 월드 템플릿 패키지에서 제공하는 기능, 속성과 제페토 스크립트를 통해 콘텐츠의 전반적인 로직, 상호작용 등의 콘텐츠 요소를 구현한다. 본 연구는 제페토 월드를 활용한 메타버스 콘텐츠 제작 공정을 제안하기 위하여 유니티 3D 엔진을 기반으로 제페토 월드 템플릿에서 제공하는 기능들과 함께 멀티플레이 메타버스 콘텐츠 제작을 위해 필요한 핵심적인 기능들을 정리한다.

2.1 제페토 월드기반 개발환경 설정

제페토 월드는 유니티 3D 엔진에 기반을 두어 개발할 수 있다. 유니티 3D 엔진에 제페토 월드 템플릿 패키지를 설치하여 콘텐츠를 제작하는데 이때, 제페토 월드는 Unity 2020.3.9f1 버전에서만 개발이 가능하다[21]. Figure 2는 제페토 월드 템플릿 패키지를 설치하여 유니티 3D 엔진에서 제페토 월드 프로젝트를 생성하는 과정을 나타낸다. 제페토 월드와 유니티 3D 엔진 프로젝트를 연결하기 위하여 제페토 스튜디오에서 월드를 생성하고, 설정한 월드 ID를 통일시키는 작업이 선행되어야 한다.

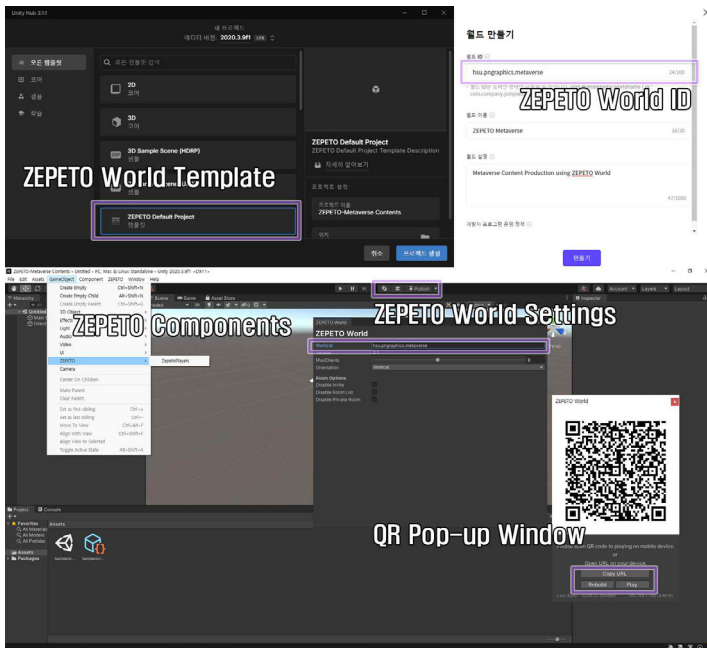


Figure 2. Process of project setting using ZEPETO World based on Unity 3D engine.

구현된 프로젝트는 유니티 3D 엔진의 에디터에서 실행 버튼을 통해서 확인할 수 있고, 모바일에 설치된 제페토 앱을 통해서도 실행할 수 있다. 상단 도구바에 생성된 제페토 버튼을 클릭하면 QR 팝업창이 표시되고, 제페토 앱을 통해 QR 코드를 스캔하면 앱에서도 실행할 수 있다.

2.2 제페토 스크립트

유니티 3D 엔진에서 C# 스크립트를 활용하여 콘텐츠의 대화형 구조, 게임성 등을 구현하는 것과 같이 제페토 월드의 로직, 상호작용 구현은 제페토 스크립트를 통해 구현한다. 제페토 스크립트는 타입스크립트(TypeScript) 프로그래밍 언어를 지원하고 있고 유니티 3D 엔진 C# 스크립트의 모노비헤이비어(MonoBehaviour) 인터페이스와 호환되도록 설계되어 있다. 다음은 제페토 스크립트 생성 시 작성되는

기본 코드이다.

```
import { ZepetoScriptBehaviour } from 'ZEPETO.Script'

export default class Sample extends ZepetoScriptBehaviour {
  Start() {
  }
}
```

스크립트가 동작하는 기본 구조는 유니티 3D 엔진의 C# 스크립트의 로직과 유사하며, 유니티 3D 엔진에서 제공하는 GameObject, Transform, Animator, Vector3과 같은 클래스, 속성 등을 동일하게 사용할 수 있는 것은 물론 코루틴(Coroutine), 입력(Input), 물리(Physics), 태그와 레이어, 충돌(OnCollisionEnter, OnTriggerEnter) 등의 기능들을 모두 활용할 수 있다. 단, 유니티 3D 엔진에서 제공하는 기능들을 제페토 스크립트에서 활용하기 위해서는 제페토 스크립트 문법에 맞춰 수정하는 작업이 필요하다.

2.3 제페토 플레이어

제페토 월드 템플릿은 제페토 앱을 사용하는 사용자가 설정한 아바타를 로딩하고 가상 공간 안에서 캐릭터를 이동, 제어하는 기능을 제공한다. 제페토 월드 내에서 카메라 제어, 캐릭터 이동, UI 제어 및 연동 등 캐릭터를 관리하는 객체로 제페토 플레이어(ZepetoPlayers)를 제공한다. 개발자는 유니티 3D 엔진의 속성 창을 통해 값을 수정하거나 UI, 애니메이션 등을 변경하는 작업이 가능하다. Figure 3은 제페토 플레이어 속성과 함께 제페토 앱을 통해 사용자의 아바타로 제페토 월드에 참여하여 동작하는 과정을 나타낸 것이다. 멀티플레이 콘텐츠 제작을 위해서는 월드에 참여하는 여러 사용자 각각의 캐릭터를 생성, 삭제 및 관리할 수 있는 기능이 필요하다. ZepetoPlayer는 제페토 캐릭터 인스턴스 단위로 이를 담당하며, 플레이어마다 고유의 아이디 값(UserId)을 통해 캐릭터 인스턴스를 장면에 불러온다.

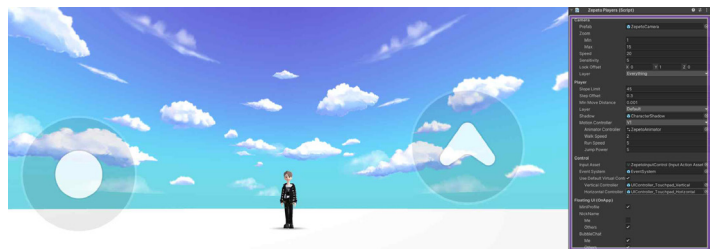


Figure 3. ZepetoPlayers component and user's avatar execution process.

2.4 멀티플레이 체험환경 구성

본 연구는 제페토 월드를 활용하여 멀티플레이 메타버스 콘텐츠 제작을 목적으로 한다. 따라서 다중 사용자 참여가 가능한 체험 환경과 이를 위해 필요한 핵심적인 기능들을 정리한다. 제페토 월드에는 다수의 사용자가 접속하여 동시에 월드 플레이를 수행할 수 있는 모듈과 함께 서버, 클라이언트 개발 및 로컬 테스트 환경을 제공하고 있다. 본 연구에서는 제페토 월드를 기반으로 멀티플레이 데모 프로젝트 [22]를 활용하여 멀티플레이 메타버스 콘텐츠를 제작하고자 한다. 멀티플레이 개발환경에는 서버 로직을 담당하는 WorldMultiplay와 클라이언트 기능을 수행하는 ClientStarter로 크게 구분할 수 있다. WorldMultiplay는 서버 메인 로직 코드를 담당하는 index.ts와 서버와 클라이언트 간 통신을 위한 데이터 구조를 정의하는 schemas로 구성되어 있다. index.ts에서는 서버에 플레이어(player)가 접속하면 방(room) 객체를 생성하고 방에 접속한 플레이어의 값들을 설정하게 된다. ClientStarter는 플레이어가 방에 입장하면 현재 방에 대한 정보와 서버 상태 변경을 수신하기 위한 이벤트를 등록한다. 멀티플레이 월드를 구성하는 데 있어 중요한 점은 다수의 사용자가 접속하여 행동을 수행하는 과정에서 각각의 플레이어들의 위치와 상태를 정확하게 동기화하는 것이 필요하다는 것이다. 따라서, ClientStarter는 내 캐릭터의 위치 정보를 전달, 다른 캐릭터 위치 수신 등의 기본 변환 정보를 제공하고 있다. 하지만, 다양한 기획 의도, 목적에 맞는 메타버스 콘텐츠를 제작하기 위해서는 콘텐츠 내에서 발생하는 다양한 상황들을 정확하게 동기화하는 과정이 필요하다. 본 연구는 이를 위하여 객체 변환, 동적 객체 생성, 그리고 제페토 캐릭터의 속성 추가 및 실시간 속성 제어 3가지 기능을 중심으로 정리한다.

Figure 4는 제페토 멀티플레이에서 제공하는 클라이언트, 서버(방, room) 사이의 메시지 송/수신 인터페이스를 나타낸 것으로 클라이언트에서 동기화가 필요한 이벤트가 발생하면 입장한 방에 이벤트의 형식(type)과 메시지(message)를 전달하고, 서버는 다시 현재 방에 입장한 전체 사용자에게 이벤트가 발생했음으로 알리게 된다(broadcast). 각각의 클라이언트들은 이벤트 메시지를 처리하기 위한 기능(AddMessageHandler)을 정의하고, 전달된 이벤트의 형식과 메시지에 맞게 구현된 기능을 동작하게 된다. 이때, 특정 클라이언트에게만 메시지를 전송하기 위해서는 client.send를, 방에 접속 중인 모든 클라이언트에게 메시지를 보내기 위해서는 broadcast를 사용하게 구현한다. 동기화를 위한 클라이언트 함수(room.Send, room.AddMessageHandler)는 제페토 스크립트를 새롭게 생성하여 구현하고, 서버 기능(onMessage, client.send/broadcast)은 index.ts 내에서 구현한다.

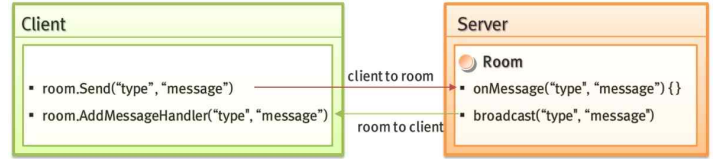


Figure 4. Interface of message sending/receiving between client and server(Room) of ZEPETO Multiplayer[21].

우선, 객체 변환 동기화는 현재 플레이어 클라이언트가 자신의 캐릭터가 아닌 제3의 객체의 변환 정보(위치, 방향, 크기)를 변경할 경우 객체 변환을 수행한 클라이언트는 방에 접속한 다른 클라이언트들에게 변환된 정보를 전달하여 동기화를 수행해야 한다. 제페토 멀티플레이의 경우 플레이어의 상태, 아이디, 해쉬코드 등의 기존 정보와 함께 유니티 3D 엔진의 변환 정보를 스키마로 미리 정의하고 있다. 따라서, 객체 변환 정보를 다룰 때는 스키마에 정의된 형식을 활용하면 효과적으로 구현할 수 있다. Figure 5는 제페토 멀티플레이에서 정의한 스키마를 나타낸 것으로, 개발자가 필요하면 스키마를 추가하여 활용하거나 기존의 스키마를 변경하여 사용할 수 있다.



Figure 5. Example of schema type settings for ZEPETO Multiplay

아래와 같이 스키마에 정의된 변환(Transform)의 위치(position)와 방향(rotation) 속성에 변경된 값을 저장하여 서버(방)에 메시지를 전달한다.

1. 클라이언트

1-1. 서버에 값 전달

```
const data = new RoomData();
const pos = new RoomData();
pos.Add("x", transform.localPosition.x);
pos.Add("y", transform.localPosition.y);
pos.Add("z", transform.localPosition.z);
data.Add("position", pos.GetObject());

const rot = new RoomData();
rot.Add("x", transform.localEulerAngles.x);
rot.Add("y", transform.localEulerAngles.y);
rot.Add("z", transform.localEulerAngles.z);
data.Add("rotation", rot.GetObject());
this.room.Send("onChangeObject", data.GetObject());
```

1-2. 서버로부터 값 전달

```
this.room.AddMessageHandler("onChangedObject", (message: Transform) => {
    //message.position, message.rotation 값을 통한 설정
});
```

2. 서버

```
this.onMessage("onChangedObject", (client, message) => {
    this.broadcast("onChangedObject", message);
});
```

현재 이벤트 형식은 onChangeObject로 설정하였고, 서버에서 메시지 함수(onMessage("onChangeObject", (client, message) => {}))를 통해 메시지를 받아 다른 클라이언트들에게 메시지를 전달(broadcast("onChangeObject", message))하며, 모든 클라이언트는 메시지 처리 함수(room.AddMessageHandler("onChangeObject", (message: Transform) => {}))를 구현하여 서버로부터 전달된 변환 정보(message: Transform)를 토대로 객체를 동기화한다.

다음은 동적 객체 생성으로 유니티 3D 엔진은 프리팹(prefab)으로 등록된 객체를 인스턴스화(Instantiate) 함수를 통해 복제한다. 다수 사용자가 참여하는 제페토 월드 역시 같은 방식으로 객체를 복제하게 되는데 복제된 객체가 모든 클라이언트에게 동일하게 표현되기 위해서는 객체를 생성하는 시점을 정확하게 정의해야 한다. 클라이언트에서 객체를 복제할 시점에 서버에게 객체 복제에 대한 메시지를 보내고(room.Send), 서버는 다시 모든 클라이언트에게 복제 메시지를 보낸다(broadcast). 마지막으로 클라이언트는 메시지 처리 함수(room.AddMessageHandler) 내에서 객체를 복제(Object.Instantiate)하면 모든 클라이언트에게 똑같이 동적으로 객체가 생성된다. Figure 6은 객체 변환과 동

적 객체 생성을 구현한 결과를 나타낸 것이다. 하나의 플레이어 클라이언트가 변환 또는 생성한 객체가 다른 클라이언트에게도 동일하게 표현되는 것을 확인할 수 있다.



Figure 6. Result of object transformation and dynamic object creation synchronization process.

마지막은 제페토 캐릭터의 속성 추가 및 실시간 속성 제어과정이다. 제페토 플레이어는 제페토 월드 내에서 현재 클라이언트의 카메라 제어, 캐릭터 이동 등의 기능을 담고 있다. 이때, ClientStarter에서는 현재 접속한 사용자의 아바타를 ZepetoPlayers 인스턴스를 통해 제페토 캐릭터로 월드 내에 생성하게 된다. 생성되는 캐릭터(아바타)는 제페토 앱에서 사용자가 커스터마이징한 캐릭터로 클라이언트마다 고유의 아이디 값(UserId)과 해시코드(hashcode)가 부여된다. 중요한 점은 제페토 캐릭터는 동적으로 생성되는 객체로 개발자가 사전에 새로운 속성을 변경, 추가, 삭제하는 등의 작업을 수행하는데 제한이 많다는 것이다. 따라서, 클라이언트마다 개별적으로 제어하는 제페토 캐릭터에 콘텐츠 기능 구현을 위해 속성을 추가하기 위해서는 스크립트를 통한 접근이 유일한 방법이다. 아래는 ClientStarter에서 클라이언트가 방에 접속한 다음 제페토 캐릭터가 생성된 후 추가로 속성들을 부여하는 과정을 보여주는 예이다. 본 연구에서는 유니티 3D 엔진에서 게임적 요소를 구현할 때 자주 사용되는 속성으로 태그, 충돌체(collider), 강체(rigidbody)와 구현된 제페토 스크립트를 추가하는 과정을 나타낸다. 속성 추가는 클라이언트가 방에 최초 접속 시 자동으로 호출되며, 이후 서버에서 상태가 변경될 경우 호출되는 함수

(room.OnStateChange()) 내에서 구현한다.

1. 현재 클라이언트와 제페토 플레이어 정보 호출

```
const player:Player = this.currentPlayers.get(sessionId);  
const zepetoPlayer = ZepetoPlayers.instance.GetPlayerWithU  
serId(player.zepetoUserId);
```

2. 태그(Tag) 속성 변경

```
zepetoPlayer.character.gameObject.tag = "Player";
```

3. 충돌체(Collider) 속성 추가 및 값 설정

```
var c = zepetoPlayer.character.gameObject.AddComponent  
<UnityEngine.BoxCollider>();  
c.isTrigger = true;  
c.center = new UnityEngine.Vector3(0, 0.6, 0);  
c.size = new UnityEngine.Vector3(4, 1.2, 4);
```

4. 강체(Rigidbody) 속성 추가 및 값 설정

```
var rd = zepetoPlayer.character.gameObject.AddComponent  
<UnityEngine.Rigidbody>();  
rd.isKinematic = true;
```

5. 제페토 스크립트 추가

```
zepetoPlayer.character.gameObject.AddComponent<"ScriptN  
ame">();
```

Figure 7은 제페토 월드 실행 후 사용자 아바타에 속성을 직접 추가한 결과를 보여주고 있다. 이외에도 메타버스 콘텐츠 제작을 위해 필요한 속성들을 자유롭게 추가하여 이벤트, 상호작용 등을 구현할 수 있다.

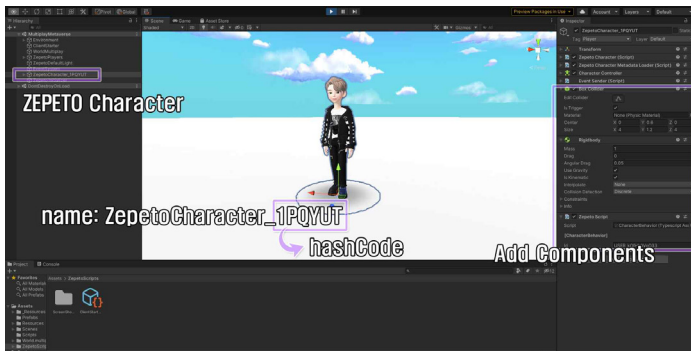


Figure 7. Result of adding components to ZEPETO character

제페토 캐릭터에 스크립트를 통해 속성을 추가할 경우 현재는 콘텐츠 내에서 클라이언트들이 동기화되는 과정에서 새롭게 추가된 속성에 다른 클라이언트들이 실시간으로 접근하는 것이 불가능하다는 문제가 있다. 본 연구는 이를 해결하기 위하여 제페토 캐릭터가 가지는 고유의 해시코드와 아이디를 통해 다른 클라이언트의 제페토 캐릭터가 접근하는 구조를 설계한다. 다음은 이를 정리한 것으로 생성되는

제페토 캐릭터의 이름 뒤 6자리가 해시코드로 부여되는 특징을 통해 해시코드로부터 아이디를 찾아 명령을 수행하는 구조로 정의한다(Figure 6, 7).

1. 클라이언트가 서버(방)에 접속 시 아이디와 해시코드를 배열에 매핑하여 저장
2. 현재 클라이언트가 다른 클라이언트의 속성에 접근해야 될 경우 제페토 캐릭터 이름 뒤 6자리(해시코드)를 분할하여 서버로 전달
3. 서버는 전달받은 해시코드로부터 클라이언트의 아이디를 찾고 검색된 아이디의 클라이언트에게 메시지를 전달
4. 호출된 클라이언트는 전달받은 메시지를 수행



Figure 8. Examples of backgrounds, scenes, UI and composition of produced metaverse content.

3. 메타버스 콘텐츠 제작



Figure 9. Results of created multiplay metaverse content using ZEPETO world.

제페토 월드를 활용하여 메타버스 콘텐츠 제작에 필요한 개발 환경 및 제작 공정에 대한 핵심 기능 정리를 통해 다수의 사용자가 함께 참여하여 체험하는 메타버스 콘텐츠를 직접 제작한다. 본 연구는 10대 사용자가 전체 사용자의 80% 가량 차지하는 제페토 앱의 특징을 고려하여 동화 콘셉트의 간단한 게임 방식의 콘텐츠로 제작하였다. 게임은 늑대와 양 동화를 배경으로 콘텐츠에 참여하는 사용자 중 한 명은 늑대, 나머지는 양이 되고 늑대가 양을 터치하면 양과 늑대가 바뀌게 된다. 그리고 주어진 게임 시간이 끝난 후 마지막에 늑대인 사용자가 패배하는 간단한 방식의 게임이다. Figure 8은 본 연구를 통해 제작된 멀티플레이 메타버스 콘텐츠의 배경, 장면, 구성 등을 나타낸 것이다[23]. 동화 콘셉트의 배경에 게임 진행에 필요한 UI(늑대 또는 양 상태 텍스트, 게임 준비 버튼 등)를 설정하였다.

4. 실험 및 분석

제안하는 메타버스 콘텐츠 제작 공정에 대한 실험과 콘텐츠 제작은 Unity 2020.3.9.f1(64bit)를 기반으로 제페토 월드 템플릿 패키지(ZEPETO.World.unitypackage 1.4.1)를 활용하여 구현하였다. 제작 공정을 위한 개발 환경과 메타버스 콘텐츠 제작을 위한 PC는 Intel Core i7-11800H, 16GB RAM, Geforce RTX 3050 Ti Laptop GPU를 탑재하고 있다. 제작된 메타버스 콘텐츠는 제페토 앱을 통해 사용자들이 접속하고 원하는 지점에서 준비(ready) 버튼을 모든 사용자가 다 클릭하게 되면 서버에서 접속된 제페토 플레이어의 아이디를 저장한 다음 임의로 하나의 아이디를 선택하여 늑대를 배정한 다음 콘텐츠가 시작된다.

Figure 9는 제페토 월드를 활용하여 제안한 제작 공정을 통해 제작된 멀티플레이 메타버스 콘텐츠 결과를 나타낸 것으로 다수의 사용자가 각자의 모바일을 통해 참여하여 콘텐

츠를 실행하는 과정을 보여주고 있다. 또한, 제작된 콘텐츠를 제페토 앱에 퍼블리싱하여 재페토 앱을 사용하는 불특정 다수의 다양한 사용자들이 직접 체험하고 의견을 피드백할 수 있도록 하였다. Figure 10은 제페토 앱을 통해 퍼블리싱된 멀티플레이 메타버스 콘텐츠 제작 결과를 나타낸 것이다. 메타버스 콘텐츠 제작이 완료되면 유니티 3D 엔진에 추가된 상단 도구바에 퍼블리시(publish) 버튼을 클릭한다(Figure 2). 그러면 서버로 직접 제작된 패키지 파일을 업로드하거나 빌드된 패키지 파일을 생성할 수 있다. 빌드된 패키지 파일을 직접 등록하는 방법은 제페토 스튜디오에 생성한 월드에 모든 정보(설명, 카테고리, 태그 등)를 작성하고 빌드된 패키지 파일을 업로드하는 것이다. 마지막으로 심사 제출 버튼을 클릭하면 심사 후 Figure 10과 같이 제페토 앱으로 출시된다.

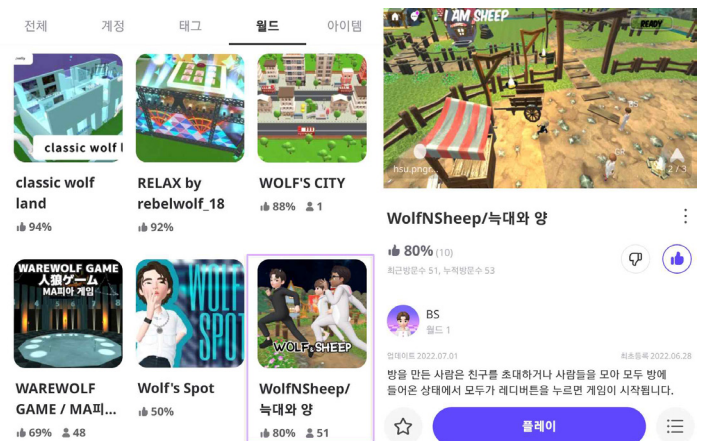


Figure 10. Publishing result of the produced metaverse content to the ZEPETO application.

5. 결론

본 연구는 제페토 월드를 활용하여 다수 사용자 참여가 가

능한 메타버스 콘텐츠 제작에 필요한 개발환경과 핵심 기능에 대한 제작 공정을 제안하였다. 유니티 3D 엔진에 제페토 월드 템플릿을 설치하여 제페토 월드 기반의 개발 환경을 구축하고 제페토 스크립트, 제페토 플레이어 등 제페토 월드에서 제공하는 기능들을 토대로 이벤트, 상호작용 등을 구현하는 과정을 정리하였다. 특히, 다수 사용자 참여가 가능한 멀티플레이 메타버스 콘텐츠 제작을 위하여 서버, 클라이언트 구조의 멀티플레이어 인터페이스와 함께 사용자들 사이의 동기화 처리를 위한 핵심적인 기능들(객체 변환, 동적 객체 생성 및 속성 추가와 실시간 속성 제어)을 정리하였다. 이러한 과정을 토대로 멀티플레이 메타버스 콘텐츠 제작하고 제페토 앱을 통해 사용자들이 참여 및 체험하는 과정을 직접 확인해봄으로써 제페토 월드를 활용한 메타버스 콘텐츠 제작 과정을 직접 확인할 수 있었다.

제페토 월드를 활용하여 메타버스 콘텐츠를 제작하는 과정은 유니티 3D 엔진을 기반으로 모든 개발이 진행되기 때문에 유니티 3D 엔진의 그래픽, 물리 등의 기능을 활용하고 있다. 그러나, 네트워크 등 일부 기능들을 구현하기 위해서 외부 엔진, 플러그인 등을 활용하기도 한다. 따라서 향후 멀티플레이 메타버스 콘텐츠를 제작하는 과정에서 PUN(Photon Unity Networking)과 같은 기존의 외부 응용 기술과 제페토 월드를 활용하는 방법과의 비교 분석을 수행함으로써 제안하는 제작 공정의 효율성을 확인하고자 한다. 또한, 제안하는 제작 공정을 통해 제작된 멀티플레이 메타버스 콘텐츠를 제페토 앱으로 퍼블리싱한 결과를 토대로 일반 사용자들의 의견을 피드백하여 분석하고, 10대를 비롯한 여러 사용자층에 대한 경험, 만족도 등을 분석할 계획이다.

감사의 글

이 논문은 2021년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임(연구개발 과제번호: 2021-0-00884, 비대면 협업용 솔루션 및 블록체인 기반 디지털 워크 통합 플랫폼 개발, 기여율 30%, 조세홍). 그리고 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (No. 2020R1F1A1063442, 기여율 40%, 박명석, 조윤식). 또한, 본 연구는 한성대학교 학술연구비 지원과제임 (기여율 30%, 김진모). 본 연구를 통해 제작된 멀티플레이 메타버스 콘텐츠는 제페토 월드 공모전에 출품될 계획임.

References

[1] J. Joshua. "Information Bodies: Computational Anxiety in Neal Stephenson's Snow Crash," *Interdisciplinary Literary Studies*, vol. 19, no. 1, pp. 17-47, 2017.

[2] L. H. Lee, T. Braud, P. Zhou, L. Wang, D. Xu, Z. Lin,

A. Kumar, C. Bermejo, and P. Hui, "All one needs to know about metaverse: A complete survey on technological singularity, virtual ecosystem, and research agenda," *arXiv preprint arXiv:2110.05352*, 2021.

[3] H. Duan, J. Li, S. Fan, Z. Lin, X. Wu, and W. Cai, "Metaverse for Social Good: A University Campus Prototype," *In 29th ACM International Conference on Multimedia (ACM MM 2021)*, Chengdu, China, October pp. 20-24, 2021.

[4] J.-G. Kim, "A Study on Metaverse Culture Contents Matching Platform," *International Journal of Advanced Culture Technology*, vol. 9, no. 3, pp. 232-237, 2021.

[5] H.-K. Chung, and J.-H. Ko, "Metaverse friend making system design and implement," *Journal of the Semiconductor & Display Technology*, vol. 20, no. 3 pp. 97-102, 2021.

[6] D.-A. Kong, "Development of the Museum Exhibition Contents using the Metaverse : Focused on the National Folk Museum of Korea," *Masters dissertation*, Graduate School of Business Administration, Kyung Hee University, 2022.

[7] M. Xu, D. Niyato, J. Kang, Z. Xiong, C. Miao, and D. I. Kim, "Wireless Edge-Empowered Metaverse: A Learning-Based Incentive Mechanism for Virtual Reality," accepted for publication, *IEEE ICC '22*, Seoul, Korea, May 2022.

[8] A. A. Gaafar, "Metaverse In Architectural Heritage Documentation & Education," *Advances in Ecological and Environmental Research*, vol. 6, issue 10, pp. 66-86, 2021.

[9] N. Xi, J. Chen, F. Gama, M. Riar, and J. Hamari, "The challenges of entering the metaverse: An experiment on the effect of extended reality on workload," *Information Systems Frontier*, Published Online, 2022.

[10] H. Zhu, "MetaAID: A Flexible Framework for Developing Metaverse Applications via AI Technology and Human Editing," *arXiv 2022*, *arXiv:2204.01614*, 2022.

[11] T. Kang, D. Lee, and J. Kim, "Production of Digital Fashion Contents based on Augmented Reality Using CLO 3D and Vuforia," *Journal of the Korea Computer Graphics Society*, vol. 26, no. 3, pp. 21-29, 2020.

[12] K. Jeong, S. Han, D. Lee, and J. Kim, "A Study on Virtual Reality Techniques for Immersive Traditional Fairy Tale Contents," *Journal of the Korea Computer Graphics Society*, vol. 22, no. 3, pp. 43-52, 2016.

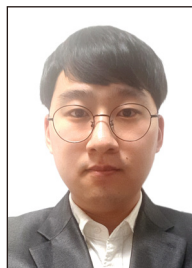
[13] Y. Cho, S. Cho, and J. Kim, "Design of Handwriting-based Text Interface for Support of Mobile Platform Education Contents," *Journal of the Korea Computer Graphics Society*, vol. 27, no. 5, pp. 81-89, 2021.

- [14] S. Hong, G. Na, Y. Cho, and J. Kim, "A Study on Movement Interface in Mobile Virtual Reality," *Journal of the Korea Computer Graphics Society*, vol. 27, no. 3, pp. 55-63, 2021.
- [15] Y. Kim, J. Heo, S. Choe, and S. Paik, "An Study of Metaverse as an Educational Space," *Convergence Education Review*, vol. 8, no. 1, pp. 1-14, 2022.
- [16] J.-H. Jeon, "A study on the Principle of Metaverse Composition with a focus on Roblox," *korean association for visual culture*, no. 38, pp. 257-279, 2021.
- [17] J.-M. Ko, and J.-E. Park, "A Study on the Experience of Generation Z using ZEPETO, a Metabus Platform," *The Korean Society of Science & Art*, vol. 40, no. 2, pp. 19-32, 2022.
- [18] H.-J. Kim, "A Study on the Metaverse as an Arts Educational Medium - Focusing on ifland of the SKT Metaverse Platform," *The journal of Convergence on Culture Technology (JCCT)*, vol. 7, no. 4 pp. 391-396, 2021.
- [19] J. Kim, "국내외 메타버스 플랫폼과 콘텐츠 비즈니스 동향," *Media Issue & Trend*, vol. 45, pp. 32-42, 2021.
- [20] ZEPETO app [Internet], Available: <https://play.google.com/store/apps/details?id=me.zepeto.main&hl=ko&gl=US>
- [21] ZEPETO studio [Internet], Available: <https://studio.zepeto.me/guides/creating-your-own-world>.
- [22] ZEPETO multiplayer demo project [Internet], Available: <https://github.com/naverz/zepeto-multiplay-example>
- [23] Greenwood Fantasy Village [Internet], Available: <https://assetstore.unity.com/packages/3d/environments/fantasy/greenwood-fantasy-village-modular-89052>

〈 저 자 소 개 〉

박 명 석

- 2022년 한성대학교 컴퓨터공학부 학사
- 2022년~현재 한성대학교 일반대학원 컴퓨터공학과 석사과정
- 관심분야: 가상현실, 증강현실, 실감형 콘텐츠 등
- <https://orcid.org/0000-0003-0341-5300>



조 윤 식

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



조 다 숨

- 2021년 한성대학교 컴퓨터공학부 학사
- 2021년~현재 한성대학교 일반대학원 컴퓨터공학과 석사과정
- 관심분야: 의료영상처리, 웹기반 가시화, 딥러닝, 인공지능
- <https://orcid.org/0000-0002-3437-4495>



나 기 리

- 2019년~현재 한성대학교 컴퓨터공학부 학사과정
- 관심분야: 가상현실, 증강현실, 실감형 콘텐츠 등
- <https://orcid.org/0000-0002-9812-2297>



이 자 민

- 2020년~현재 한성대학교 컴퓨터공학부 학사과정
- 관심분야: 가상현실, 증강현실, 실감형 콘텐츠 등
- <https://orcid.org/0000-0002-1414-9311>





조 세 홍

- 1983년 연세대학교 3년 수료
- 1991년 California State Univ. CS 학사
- 1996년 Arizona State Univ. CSE MCS
- 1999년 Arizona State Univ. CSE Ph.D.
- 1999년~2002년 대구대학교 정보통신공학부 조교수
- 2002년~현재 한성대학교 컴퓨터공학부 교수
- 관심분야: 메타버스, 가상현실, 증강현실, 디지털콘텐츠, 멀티미디어, 원격교육, 인공지능, 빅데이터 등
- <https://orcid.org/0000-0003-3974-8086>



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>