

언리얼엔진에서의 가상세계 창작을 위한 효율적 모듈화 연구

오민준*

송실대학교 글로벌미디어학부 학생

A Study on the Efficient Modularization of Virtual World Creation in Unreal Engine

Min-Jun Oh*

Student, Dept. of Global Media, Soongsil University

요약 기존 게임 개발에서 가상세계 제작은 기준 없이 게임 요소들을 하나씩 배치하는 방식으로 이루어졌다고 판단된다. 여기에서 주목할 것은 투자 대비하여 과연 양질의 가상세계를 효율적으로 제작 되었는지에 대한 궁금증이다. 본 연구에서는 언리얼 엔진에서 모듈화라는 개념을 바탕으로 효율적 가상세계를 구축할 수 있는 방법론을 제안하고자 한다. 먼저 선행사례들을 분석하고 모듈화를 위한 기준요소를 5가지로 추출하였다. 또한 인스턴스 제작 파이프라인의 개념을 4단계로 구분하여 제안 하였으며, 도심형 가상세계 제작을 위한 최소단위의 인스턴스 모듈을 4가지로 압축하였다. 최종적으로 최소단위의 모듈과 기준요소를 바탕으로 구성된 도심형 가상세계를 구현하여 제시하였다. 결론적으로 이러한 효율성을 중심으로 하는 제작방식에 대한 연구는 디자이너나 아티스트가 제작에 소비해야 했던 시간을 아이디어와 창의력에만 집중할 수 있다고 사료된다. 연구 한계점으로는 기본이 되는 최소 모듈을 도시로 한정된 점과 언리얼 엔진으로 구현함에 있어 도출한 기준요소와 제작 파이프라인들에 대한 검증이 이루어지지 않았다는 점을 들 수 있다. 이에 보다 발전적인 모듈화 연구를 통하여 다양한 가상세계 생성 방안이 도출되기를 기대해본다.

키워드 : 가상세계, 언리얼엔진, 인스턴스, 모듈화, 파이프라인

Abstract In the development of existing games, it is judged that virtual world production was done by arranging game elements one by one. What is noteworthy here is the question of whether quality virtual worlds were efficiently produced in preparation for investment. In this study, we propose a methodology that can build an efficient virtual world based on the concept of modularization in an unreal engine. First, precedents were analyzed and five reference elements for modularization were extracted. In addition, the concept of an instance production pipeline was proposed by dividing it into four stages, and the minimum-unit instance modules for urban virtual world production were compressed into four. Finally, an urban virtual world constructed based on the minimum unit module and reference elements was implemented and presented. In conclusion, research on the production method centered on this efficiency is thought to be able to focus the time that designers or artists had to spend on production only on ideas and creativity. The limitations of the research are that the basic minimum module is limited to the city, and the derived reference elements and production pipelines have not been verified when implementing them with an unreal engine. Therefore, it is expected that various virtual world creation plans will be derived through more advanced modular research.

Key Words : Virtual world, Unreal engine, Instance, Modularization, Pipeline

1. 서론

1.1 연구배경

기존 게임 개발에서 가상세계는 어떠한 기준도 없이 가상공간에 들어가는 게임 요소들을 하나씩 배치하는 방

식으로 이루어졌다고 판단된다. 이러한 개발 방식은 많은 시간과 예산이 투입되어야 제작이 가능한 방식이다[1,2]. 예를 들면, 최근 유행하는 메타버스에서 가상 월드를 구축하기 위하여 수많은 업체들이 현재 서비스를 구축하려고 제공하거나 개발 중에 있다[3]. 여기에서 주목할 것은

*Corresponding Author : Min-Jun Oh(jun.pentad@gmail.com)

Received July 19, 2022

Accepted November 20, 2022

Revised August 23, 2022

Published November 28, 2022

투자 대비하여 과연 양질의 가상세계가 효율적으로 제작되었는지에 대한 궁금증이다. Fig. 1과 같이 ‘엘더스크롤 5 스카이림(2011)’은 제작기간이 무려 5년에 걸쳐 제작되었으며, 개발 비용은 8천 5백만 달러로 상당한 시간과 개발비용이 투자된 사례이다.



Fig. 1. Elder scroll skyrim (2011)

반면에 현재의 가상세계 제작은 대규모 투자로 제작되는 방식에서 개인 창작자나 소규모 스튜디오에서 제작이 가능할 정도로 개발 툴이 강력해지고 있다[4]. 또한 효율적인 가상세계 및 에셋[1]들의 제작 파이프라인이 다양하게 시도되고 있다[5]. 따라서 본 연구에서는 언리얼 엔진에서 모듈화라는 개념을 바탕으로 효율적 가상세계를 구축할 수 있는 방법론을 제안하고자 한다.

1.2 연구방법 및 범위

먼저 기존의 가상세계를 효율적으로 제작하기 위하여 선행사례들을 분석하고 기준요소들을 추출하며 도출된 요소로서 모듈화 사이즈에 대한 정립을 최소 단위부터 최대 범위를 4단계로 구분하였다. 4단계의 모듈화 블록들을 조합하여 도심형 인스턴스 모듈을 바탕으로 가상세계를 제작하고자 한다. 연구에 사용된 툴은 언리얼엔진을 사용하기로 하였으며 그 이유는 나나이트와 루멘 등의 현실적인 실시간 렌더링이 가능하다는 점과 블루프린트의 컨스트러션 스크립트(Construction Script)²⁾를 통해 각각의 모듈화 건물 생성부터 머트리얼 및 다양한 프랍(prop)까지 디테일한 부분까지 섬세한 연출이 가능하다고 판단했기 때문이다. 인스턴스 모듈의 제작 범위는 도

1) 에셋(Asset) : 언리얼 엔진 프로젝트의 콘텐츠 조각으로, 파일에 시리얼라이즈하여 저장하는 UObject를 의미함.
 2) 컨스트러션 스크립트 : 블루프린트의 인스턴스 생성시 초기화 작업을 수행하기 위해 실행되는 노트 그래프를 의미함.

심 지형으로 한정하였으며 이는 모듈화 요소의 추출과 적용에 있어 자연 지형보다 유리하다고 판단하였기 때문이다. 종합해보면 아래 그림과 같이 연구 흐름도를 요약할 수 있다.

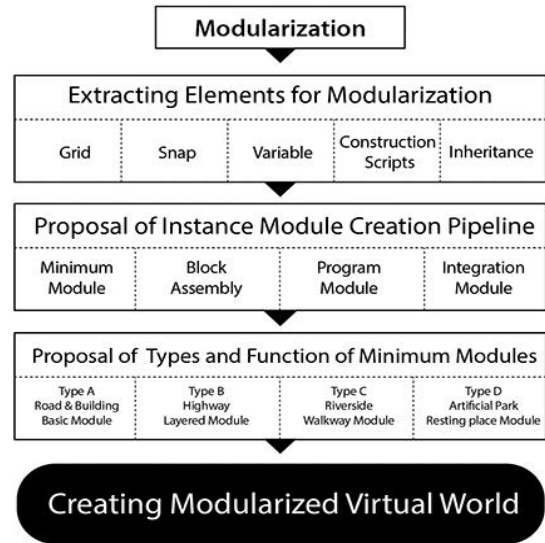


Fig. 2. Instance modularization mlow

2. 이론적 배경

2.1 가상세계 종류

가상세계는 Table 1과 같이 3가지로 구분할 수 있다.

Table 1. Types of virtual world





Type	Case
Digital Twin World	 Microsoft Flight Simulator
Virtual Reality World	 The Witcher 3 : Wild Hunt
Fusioned Reality World	 inFamous SECOND SON

디지털 트윈 월드 사례로 2020년 아소보 스튜디오에서 개발한 ‘마이크로소프트 플라이트 시뮬레이터 (Microsoft Flight Simulator 2021)’는 위성사진을 이용하여 현실에 존재하는 건물을 3D로 변환하여 가상 세계에 구현하는 방식을 사용하였다[6].

2005년 CD PROJEKT RED에서 개발한 ‘더 위쳐3, 와일드 헌트(The Witcher3, Wild hunt)’에서의 가상세계는 가상 현실 공간으로 현실에 존재하지 않는 세계관을 창작하고, 건물과 환경요소를 기초부터 제작하는 방식을 채용하였다. 각 지역에 어울리는 건물들을 새롭게 모델링하고 내부를 모듈화하여 배치하는 것으로 자연스럽고 커다란 가상세계를 제작해냈다[7].

2014년에 서커펀치 스튜디오에서 개발한 ‘인퍼머스 세컨드 선 (inFamous SECOND SON)’으로서 현실의 디지털 트윈과 새로운 설정 및 외형을 덧붙여 가상 속의 현실을 제작하는 융합형 가상세계를 들 수 있다.

Table 2. Method of virtual world creation

Type	Case
Unreal Engine Blueprint	 Project RYU (2022)
Houdini City Generator	 Matrix Awakens (2022)
Original Building Generator Tool	 Assassin's Creed Unity (2014)
Digital Twin	 Assassin's Creed Unity (2014)

언리얼엔진의 블루프린트 기능을 이용한 노드 형식의 프로그램을 통해 게임 엔진 상에서 모듈화가 이루어진다.

먼저 러셀(RYU Russell)³⁾의 언리얼 엔진 게임 개발 프로젝트에 등장하는 한국형 아파트는 블루프린트의 컨스트럭션 스크립트를 사용하여 인스턴스화된 하나의 건물 메뉴판을 기준으로 아파트들을 절차적 생성(Procedural Generation)으로 제작하였다.

다음은 후디니(Houdini)에서 에셋 제네레이터를 개발하고 그것을 게임 엔진에서 적용하는 방식이다. 그 사례로 언리얼 엔진 5와 함께 등장한 테크니컬 데모인 매트릭스 어웨이크스(Matrix Awakens)⁴⁾에서 선보인 샘플 시티(Sample City)가 있다. 개발진은 후디니의 노드 프로그램으로 제작한 플러그인을 사용하여 도시의 레이아웃을 만들고 도로를 구성했으며 거대한 가상세계를 제작하였다[9].

어쌔신크리드 유니티(Assassins Creed Unity 2014)⁵⁾에서 유비소프트(Ubisoft)가 개발한 거리 조성 프로그램은 독자적인 맵 에디터를 사용하는 모듈화 방식이다. ‘그레이박스(Grey Box)로 건물들의 한 묶음을 모듈로 만들어 거리를 임시 조성한 다음 실제 건물을 에셋으로 대치하는 방식으로 오픈 월드를 제작하였다. 마지막으로 현실에 존재하는 건축물을 처음부터 모델링하여 가상세계에 적용하는 디지털 트윈 방법이 있다. 높은 퀄리티의 현실과 동일한 건물을 제작하기 위해, 유비소프트 몬트리올에서는 파리의 노트르담 대성당을 2년에 걸쳐 내부까지 새롭게 모델링하여 배치하였다[10].

이처럼 다양한 방법으로 모듈화를 이용해 가상세계가 제작되는 사례를 살펴보았다. 요약하면 장시간의 노력과 대규모의 자본이 투자된 사례 보다는 러셀의 한국형 아파트를 효율적으로 구현하는 방식을 들 수 있으며 그 이유는 단시간에 소규모 인원으로도 방대한 건물로 이루어진 도심을 구현하는데 적합하다고 판단된다.

3. 가상세계의 효율적 모듈화 제안

3.1 모듈화를 위한 기준 요소

모듈화를 위한 기준으로 Table 3과 같이 요소를 추출하였다.

첫 번째로 바둑판 형태의 그리드를 들 수 있다. 왜냐하면 자연환경처럼 무작위적 생성이 아닌 4각형의 모듈을 사용해야 효율적으로 도심형 인스턴스 모듈을 제작할 수 있다고 판단했기 때문이다. 두 번째로는 당연할 수 있지만 모든 모듈을 조합할 때 스냅의 유무에 따라 제작과정

3) Project RYU, <https://www.youtube.com/c/RYURussell>

4) <https://www.unrealengine.com/ko/wakeup>

의 효율성에 영향을 주기 때문에 중요한 요소이다[11]. 세 번째는 변수로서 무한대에 가까운 다양한 생성을 위해 건물의 크기, 각도, 조명, 회전 등을 고려하면서 모듈화가 이루어져야 한다는 것이고, 랜덤하게 생성된 최소 모듈이 지나치게 동일하면 작위적으로 보일 수 있기 때문에 자동 생성 과정에 자연스러운 변화가 필요하다[12]. 네 번째로 컨스트러션 스크립트를 통한 질차적 생성의 스크립트의 구현이 블루프린트로 가능하다. 다섯 번째로 모듈화한 디테일 액터들은 인스턴스 과정을 거쳐 부모 액터에 상속됨으로서 차일드 액터의 수정만으로도 부모 액터까지 일괄적으로 변경이 이루어져야 한다.

Table 3. Elements for modularization

Type	Case
Grid	Square modules are used to efficiently build urban instance modules
Snap	The presence or absence of a snap when combining modules affects the efficiency of the fabrication process
Variable	Modularization should take into account various variables such as the size, angle, and lighting of the building
Programming	Scripting procedural generation with construction scripts
Inheritance	Modification of the child actor inherited from the parent actor allows you to change and manage the parent actor collectively

3.2 인스턴스 모듈 개요

인스턴스 모듈이란 앞서 언급한 모듈화의 기준요소를 대입하여 최소 단위의 모듈을 아이덴티티화 시킨 것으로서 한번 모듈화가 되면 다양한 변수에 의해 복제할 수 있다는 개념이다. 따라서 우선적으로 최소 단위가 되는 모듈을 제작하고, 이것을 Fig. 2와 같이 그리드 위에 마치 도장을 찍듯이 간편하게 배치할 수 있다.

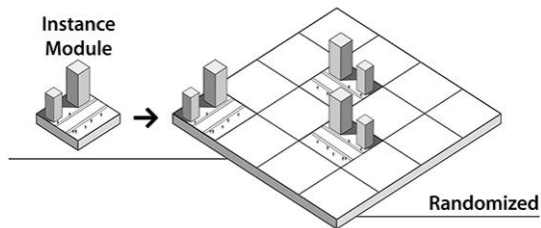


Fig. 3. Randomness of instance module

다음 단계로 인스턴스 제작 파이프라인의 개념은 Fig. 3과 같이 4단계로 제안한다. 그 이유는 많은 단계보다는 최소단계가 지금의 개념을 정립하는데 도움이 된다고 판단되기 때문이다. 해당 파이프라인에서는 작은 블록 단위의 모듈이 합쳐지며 커다란 가상세계를 구성하는 구조를

가지고 있다.

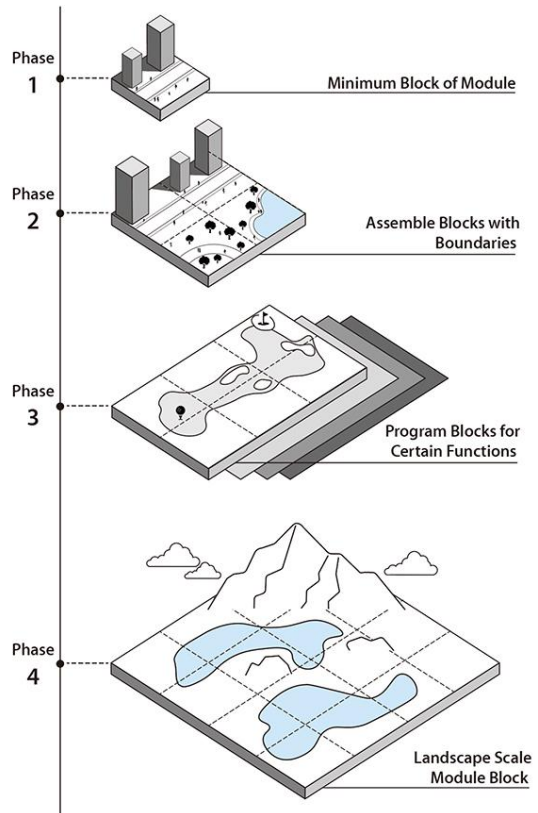


Fig. 4. Instance module creation pipeline

1단계는 그리드의 최소 단위가 되는 모듈로서 기본 플랫폼 위에 건물이나 다양한 프랩을 배치하는 것으로 구성할 수 있다. 도심과 환경요소 및 도로와 건물들의 결합으로 다양한 모듈들을 제작 가능하다. 2단계는 최소 단위의 블록들의 경계면을 고려한 합성 개념으로 모델이 가지고 있는 특성에 따라 다양하게 변경될 수 있다.

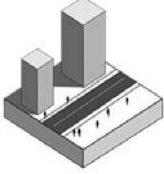
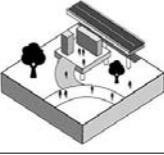
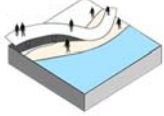

3단계는 특수한 목적을 지닌 건축물이나 지형 모듈인 프로그램을 적용하는 단계라고 할 수 있다. 프로그램 모듈은 골프장, 광장, 테마파크, 워터파크, 리조트, 대형 공원, 체육관, 스케이트장, 축구장, 야구장, 유원지 등 특정한 목적을 가지고 있지만 그 외형이 다른 모듈과 어울리지 않는 경우로서 별도로 분리하여 제작해야 한다.

마지막 단계인 4단계는 1단계부터 3단계의 모듈이 모두 결합된 가장 큰 사이즈의 집합으로서 전체적인 환경을 모듈화로 제작하고 지형과 건물의 밀집도나 위치를 고려하여 제작하는 단계이다.

4. 도심형 인스턴스 모듈 제작

도심형 가상세계 제작을 위한 인스턴스 모듈은 4가지 기본 모듈을 제작하였고 모듈화를 위한 기준요소들이 적용된 모듈을 제안하고자 한다.

Table 4. Example types of instance module

Type	Function
	Type A Road & Building Basic Module
	Type B Highway Layered Module
	Type C Riverside Walkway Module
	Type D Artificial Park Resting place Module

A 타입은 도시 모듈에서 가장 기본이 되는 모듈로서 도로와 건물이 랜덤 배치되는 모듈이다. 건물은 주거지형이나 상업형 등 다양하게 랜덤화되어 배치가 가능하다. B 타입은 고가도로나 순환로 등 포함된 A 타입에 레이어가 추가되는 모듈이다. C 타입은 도심의 하천 및 강변 산책로 등으로 구성된 모듈이다. 마지막으로 D 타입은 휴식공간으로서 도심 속에 존재하는 공원을 조성하기 위한 모듈이다.

이러한 4개의 타입은 하나의 메뉴판과 같이 사용되기 위해 제안하는 것으로서 가상세계의 특성이나 환경에 따라 유기적으로 변동할 수 있으며, 이러한 기본 모듈에 대한 고심이 깊어질수록 효율성은 극대화 될 수 있다. 왜냐하면 메뉴판에서 선택할 수 있는 변수가 결정되기 때문이다. 블루프린트에서 요소를 한 가지만 바꿔도 가상세계 전체에 영향을 주기 때문에[13] 부모 액터와 프랍들로 이루어진 자식 액터들은 매우 섬세하게 생성되도록 고려되

어야 한다. 다음 Fig. 4은 최소 단위의 모듈과 기준요소를 바탕으로 구성된 도심형 가상세계를 언리얼에서 제작한 프로토타입이다.

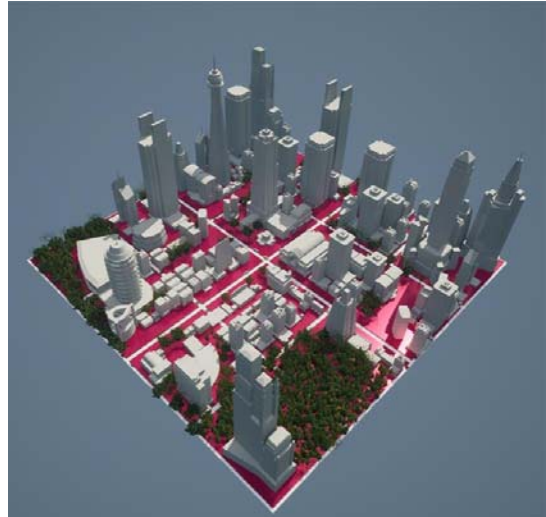


Fig. 5. Part of modularized virtual world



Fig. 6. City module cutscene



Fig. 7. City module intersection cutscene

최소 모듈은 인스턴스화를 거쳐 자동생성한 다음 건물의 배치를 임의로 조정하였으며 최종디자인은 최소 단위

의 모듈을 스네핑으로 그리드 위에서 손쉽게 배치하고 언리얼 엔진의 폴리시 툴을 이용해 자연식생과 도심의 프롭들을 생성하여 완성하였다. 본 연구에서 주지하고 있는 것처럼 최소 단위의 모듈과 다양한 변수에 대한 고심 및 인스턴스 메뉴판 등 3가지는 대단위 가상세계를 구성하는 중요한 기준이 된다고 할 것이다.



Fig. 8. City module field of view cutscene

5. 결론

본 연구는 앞으로 다가올 메타버스 시대의 가상세계를 구현하는데 필요한 저작 도구로서 실시간 게임엔진을 활용한 효율적인 모듈화를 통해 가상세계를 제작하였다. 가상세계 구축에서 효율성이라는 부분은 인스턴스화에 대한 부분이 가장 중요한 개념이며 이는 언리얼 에디터에서 제작 시 컨스트럭션 스크립트를 사용해 블루프린트를 제작해야 플레이어튼을 누르지 않더라도 실시간으로 즉각적인 반응을 살펴봄과 제작 할 수 있다. 또한 인스턴스를 시키지 않은 에셋을 대규모 랜드스케이프에 배치하는 것은 시간적이나 비용적 측면에서도 매우 비효율적이기 때문에 인스턴스화는 모듈화에 꼭 필요한 요소이다. 따라서 가상세계를 제작하는데 인스턴스화에 최적화된 에셋을 컨스트럭션 스크립트 블루프린트에서 실시간 반응에 의한 절차적 생성 방법이 매우 효율적인 제작 파이프라인이라고 강조할 수 있을 것이다.

최종적으로 가상세계를 제작한 의의는 다음과 같다.

첫 번째는 모듈화를 위한 기준으로 그리드, 스넵, 변수, 컨스트럭션 스크립트, 상속 등 5가지 요소를 추출하였다. 두 번째는 인스턴스 제작 파이프라인의 개념을 4단계로 구분하여 제안하였으며, 세 번째는 도심형 가상세계 제작을 위한 인스턴스 모듈을 4가지로 압축하였다.

결론적으로 이러한 효율성을 중심으로 하는 제작방식에 대한 연구는 디자이너나 아티스트가 제작에 소비해야 했던 시간을 아이디어와 창의력에만 집중할 수 있도록 하며[14], 새로운 가상공간을 창출함에 있어서 사용자에게 새로운 경험을 체험하게 할 수 있다고 사료된다[15].

연구 한계점으로는 기본이 되는 최소 모듈을 도시로 한정된 점과 언리얼 엔진으로 구현함에 있어 도출한 기준 요소와 제작 파이프라인 등에 대한 검증이 이루어지지 않았다는 점을 들 수 있다. 향후 연구는 사용자 경험 설문이나 전문가 인터뷰를 통하여 보다 신뢰할 수 있는 기준이 마련되어야 할 것이다. 이에 보다 발전적인 모듈화 연구를 통하여 다양한 가상세계 생성 방안이 도출되기를 기대해본다.

REFERENCES

- [1] Bleszinski, C., & Epic Games. (2000). The art and science of level design. *In Game Developers Conference.*, 4(3), 1-12.
- [2] H. J. Lee. (2003). *(A) study on the history of the game*. Master's thesis. Kookmin University, Seoul
- [3] J. H. Yoon & G. E. Kim. (2021). *Evolution prospects and innovation strategies of metaverse virtual world ecosystem*. Sejong : Science & Technology Policy Institute
- [4] M. J. Lee. (2013). A Study on Game Production Education through Recent Trend Analysis of 3D Game Engine. *Journal of the Korea Convergence Society*, 4(1), 15-20. DOI : 10.15207/JKCS.2013.4.1.015
- [5] S. G. Lee & S. M. Lee. (2019). *Changes in Content Industry Production and Distribution Structure and Challenges : Focus on gaming and new media content*. Seoul : Korea Culture & Tourism Institute
- [6] Y. Y. Kim. (2021). Digital Twin Concepts and Use Cases by Technology and Industry, *Intelligence Information Technology Trend*, 1(1), 31-47.
- [7] Noclip - Video Game Documentaries. (2017). *Designing The World of The Witcher 3*. Youtube (Online). <https://www.youtube.com/watch?v=oSS5T4od-GQ&t=173s>
- [8] J. W. Ko, B. P. Kyung, S. H. Ryu, D. L. Lee, W. B. Lee & D. Y. Lee. (2018). Create 3-Dimension Game World used Procedural Generation Algorithm. *Journal of Industrial Convergence*, 16(1), 35-40. DOI : 10.22678/JIC.2018.16.1.035
- [9] H. Rattey & Prashant. (2019). *Game design and procedural asset creation in unreal and houdini*.

5) 비용적 : CPU나 GPU의 사용도가 높아질수록 시스템에 부하가 늘어나기 때문에 이를 줄이기 위한 최적화의 필요성을 상대적으로 표현한 것임.

New Delhi : Indraprastha Institute of Information Technology.

- [10] A. Webster. (2019). *Building a better Paris in Assassin's Creed Unity*. TheVerge(Online). <https://www.theverge.com/2014/10/31/7132587/assasins-creed-unity-paris>
- [11] K. Huang, C. Wang, R. Liu & G.Chen. (2022). A Fast and Accurate Spatial Target Snapping Method for 3D Scene Modeling and Mapping in Mobile Augmented Reality. *ISPRS International Journal of Geo-Information*, 11(1), 69.
DOI : 10.3390/IJGI.11010069
- [12] Epic Games. (2004). Randomly Generated Foliage. Unreal Engine 4.26 Documentation(Online). <https://docs.unrealengine.com/4.26/ko/Resources/Showcases/BlueprintExamples/Foliage/>
- [13] Epic Games. (2004). *Blueprint class*. Unreal Engine 4.27 Documentation(Online). <https://docs.unrealengine.com/4.27/ko/ProgrammingAndScripting/Blueprints/UserGuide/Types/ClassBlueprint/>
- [14] Y. Zhong, T. S. Yun & B. C. Lee. (2021). The workflow of making realistic 3D model by combining photogrammetry and Nanite. *The Korea Contents Association ICCS symposium. Section H-1*. 241-242.
- [15] M. H. Kim. (2022). *Real-time Digital Human Era (Avatar) Industry Trends and Technology Utilization*. *KICS(Information & Technology Magazine)*, 39(5), 74-80.

오민준(Oh, Min Jun)

[학생회원]



• 2016년 2월~현재 : 숭실대학교 글로벌미디어학부 재학

- 관심분야 : 게임 기획, 개발
- E-Mail : jun.pentad@gmail.com