

3D 시뮬레이션 콘텐츠 설계 및 개발

Design and Development of 3D Simulation Contents

배재환^{1*}, 정영호²

Jae-Hwan Bac^{1*}, Young-Ho Jung²

요약

시뮬레이션(Simulation)은 모의실험이라는 뜻이다. 더욱 정확하게는 복잡한 문제를 해석하기 위하여 모델에 의한 실험, 또는 사회현상 등을 해결하는 데서 실제와 비슷한 상태를 수식 등으로 만들어 모의적으로 연산을 되풀이하여 그 특성을 파악하는 일. 시뮬레이션은 학문적인 분야와 일반인들 사이에서 사용되는 의미에서 다소 차이가 있다. 일반인들에게는 시뮬레이션 게임과 운전 시뮬레이션 장치와 철도 시뮬레이션, 항공기 조종사들이 조종을 연습하기 위해 사용하는 비행 시뮬레이션이 유명하다. 하지만 시뮬레이션의 원래 의미는 이와 차이가 있다. 시뮬레이션은 연구목적에 따라 필요한 부분을 현실에 맞게 구현한 후 이것을 무한에 가까운 정도로 수없이 반복하여 통계적으로 최적 해를 찾아내는 것이다. 이번 논문에서는 3D 시뮬레이션 콘텐츠를 공개 SW인 Unity3D API를 C# 및 html5 언어를 이용해서, 서버기반으로 설계 및 개발 하였다. 또한 향후 H/W 시뮬레이터 연동을 위한 라즈베리 파이를 이용한 시뮬레이터 시스템을 개발 하였다. 이를 통해서 국내 게임 및 3D 시뮬레이션 콘텐츠 산업 발전에 도움이 되고자 한다.

핵심어 : 오픈 SW, 3D 콘텐츠, 3D 시뮬레이션, Unity3D

Abstract

Simulation is a simulation. More precisely, in order to solve complex problems, it is necessary to solve the experiment by the model or the social phenomenon, and to simulate the characteristics by simulating the actual state similar to the actual state. The simulation is somewhat different from the academic field and the meanings used by the general public. The public is famous for simulation games, driving simulators, railway simulations, and flight simulations that aircraft pilots use to practice steering. However, the original meaning of the simulation is different. The simulation is to find the optimal solution by statistically repeating it by infinitely many times after realizing the necessary parts according to the research purpose. In this paper, we designed and developed the Unity3D API, which is a public SW software for 3D simulation contents, on a server basis using C # and html5 languages. We also developed a simulator system using Raspberry pie for interworking H / W simulator in the future. Through this, we intend to help develop domestic game and 3D simulation contents industry.

Keyword : Open SW, 3D content, 3D simulation, Unity3D

1 Department of Game Engineering, TongMyong University, Busan, Korea, [Professor]

e-mail : bjhmail@tu.ac.kr (Corresponding Author)

2 Department of Game Engineering, TongMyong University, Busan, Korea, [Graduate Student]

e-mail : elu6150@naver.com

* "본 연구는 과학기술정보통신부 및 정보통신기획평가원의 SW중심대학지원사업의 연구결과로 수행되었음 (No-1711081052)."

Received(March 28, 2019), Review Result(1st: April 12, 2019, 2nd: April 30, 2019), Accepted(June 03, 2019), Published(June 30, 2019)

1. 서론

2000년 6월 마이크로소프트가 닷넷(.NET) 플랫폼을 위해 개발하였다. C++(시플러스플러스)에 기반을 둔 언어로, 비주얼베이직이나 자바(Java)와도 비슷하다. 따라서 비주얼베이직과 자바·C++ 등의 장점을 지닌다. 곧 비주얼 언어가 가진 사용자 친화성, C++의 객체지향성, 자바의 분산환경처리에 적합한 다중성 등을 모두 지니는 컴포넌트 기반의 소프트웨어 개발 패러다임을 반영한다. 웹을 통해 정보와 서비스를 교환하고, 개발자들이 이식성(portability) 높은 응용프로그램들을 만들어 낼 수 있게 고안되었다. 즉, 이 프로그래밍언어를 사용하면 대대적인 개정 없이도 하나 이상의 OS(운영 체제)에서 사용될 수 있는 응용프로그램들을 만들어낼 수가 있다. 따라서 프로그래머가 별도의 코드를 만들지 않고서도 새로운 제품이나 서비스를 빠르고 값싸게 시장에 내놓을 수 있게 된다.

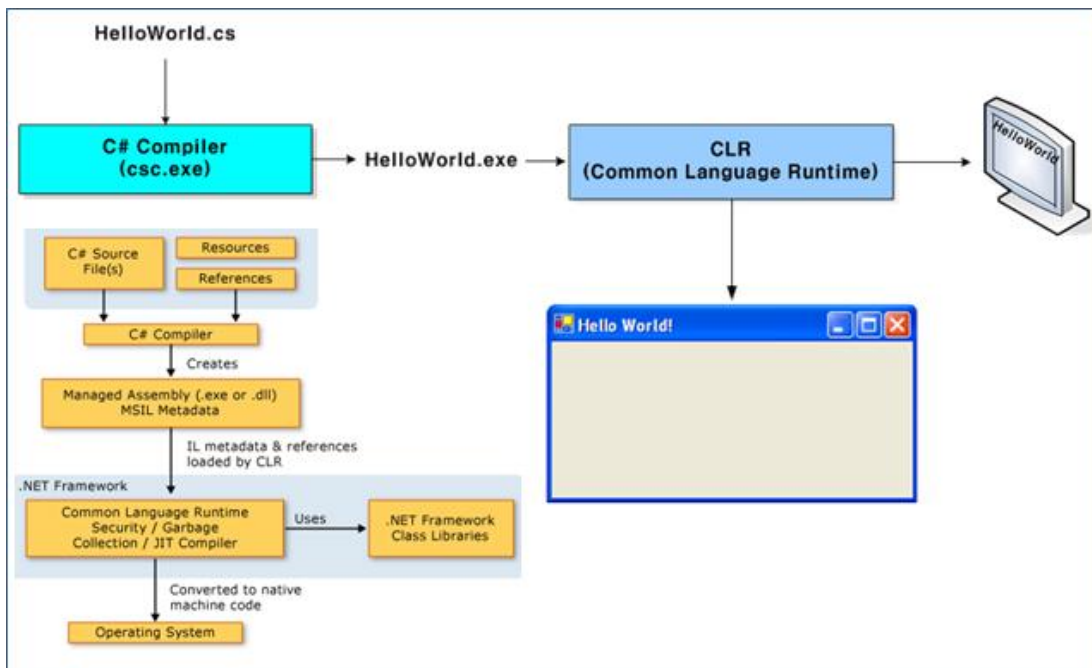
Simulation은 모의실험이라는 뜻이다. 더욱 정확하게는 복잡한 문제를 해석하기 위하여 모델에 의한 실험, 또는 사회현상 등을 해결하는 데서 실제와 비슷한 상태를 수식 등으로 만들어 모의적으로 연산을 되풀이하여 그 특성을 파악하는 일. 시뮬레이션은 학문적인 분야와 일반인들 사이에서 사용되는 의미에서 다소 차이가 있다. 일반인들에게는 시뮬레이션 게임과 운전 시뮬레이션 장치와 철도 시뮬레이션, 항공기 조종사들이 조종을 연습하기 위해 사용하는 비행 시뮬레이션이 유명하다. 하지만 시뮬레이션의 원래 의미는 이와 차이가 있다. 시뮬레이션은 연구목적에 따라 필요한 부분을 현실에 맞게 구현한 후 이것을 무한에 가까울 정도로 수없이 반복하여 통계적으로 최적해를 찾아내는 것이다[1][2]. 이번 논문에서는 3D 시뮬레이션 콘텐츠를 오픈 SW인 Unity3D API를 C# 언어를 이용해서, 설계 및 개발 하고자 한다. 이를 통해서 국내 게임 및 3D 시뮬레이션 콘텐츠 산업 발전에 도움을 주고자 한다. 논문의 주요 구성은 1장 서론, 2장 C# 및 Unity 3D 개요, 3장 3D 시뮬레이션 콘텐츠 설계 및 개발, 4장 결론으로 구성 된다.

2. C# 및 Unity 3D 개요

2.1 C# 언어

Java가 나오자 MS에서도 썬 마이크로시스템즈와 라이선스를 맺은 뒤 독자적인 Java 확장 언어 비주얼 J++를 만들었다. 그런데 썬에서 개발한 JVM에 MS가 임의로 부가 기능을 추가하면서 썬과 특허권 소송이 걸리는 바람에, MS의 비주얼 스튜디오에서 비주얼 J 시리즈와 MS VM(MS 버추얼

머신)을 사용할 수 없게 되었다. 하지만 Java라는 언어 자체가 매력적인 언어이고 MS 역시 이를 버릴 수 없다고 생각한 것인지, 다음 시리즈인 .NET Framework에서 C#이라는 이름으로 새로운 언어를 발표한다. Java가 자바 가상머신이 필요하듯이 C#은 닷넷 프레임워크가 필요하다. 닷넷 프레임워크 대신 비교적 작은 사이즈의 가상머신에 올려서 실행시키는 방법도 있다. 성능 관련해서는, 사실 Java가 처음 나왔을 때 성능 때문에 많은 포화를 맞았고 C#이 상대적으로 큰 논란 없이 입성하기는 했다. 일단은 Java보다 차세대 언어이므로 Java보다 성능적으로 우위인 부분이 많긴 하지만 (비슷한 코드에서 평균적으로 10~20% 정도 더 빠르다.) 여전히 C(++) 보다는 느린 부분이 있는 것도 사실이다. 언어적으로는 Java와 유사하지만, MS라는 특정 플랫폼에 강하게 묶여있다는 특성으로 애플의 Objective-C와 자주 비교가 된다[3][4]. 아래 [그림 1]에 C# 구조를 나타 내었다.



[그림 1] C# 구조

[Fig. 1] C# Architecture

2.2 Unity3D 엔진

오픈 SW인 Unity3D의 슬로건이 ‘게임 개발의 민주화(democratize game development)’다. 누구나 쉽게 게임을 만들 수 있게 도와 게임 업계를 성장시키겠다는 포부인 셈이다. 실제로 유니티는 모바일 게임 개발자에게는 없어서는 안 될 존재로 성장했다. Unity3D는 게임 엔진 기술이자 통합개발환경(Integrated Development Environment, IDE)이다. 이 기술을 처음 만든 이는 데이비드 헬가손

(David Helgason), 니콜라스 프렌시스(Nicholas Francis), 요하킴 안테(Joachim Ante)이다. 2002년 덴마크 개발자였던 니콜라스 프렌시스는 ‘오픈GL’과 관련된 게시판에 3D 객체를 보다 잘 표현할 수 있는 방법에 대한 문의를 올린다. 독일에 거주하고 있던 요하킴 안테는 니콜라스 프렌시스의 질문에 답을 올렸고, 이 과정에서 서로 소통하면서 하나의 게임엔진 안에서 협업할 수 있는 개발 환경이 필요하다는 것을 깨닫는다. 두 사람은 바로 게임 개발도구를 만들기로 결정했다. 데이비드 헬가손도 여기에 합류했다. 세 사람이 또 하나 신경썼던 부분은 라이선스였다. 당시 많은 게임 개발자들은 게임 콘텐츠를 고민하기도 전에 게임 개발도구의 라이선스나 로열티 같은 비용을 먼저 생각해야 했다. 유니티 설립자들은 “게임은 기술을 발전시키기 위한 필수 요소”라고 생각해 “게임을 먼저 생각하고, 라이선스는 나중에 생각할 수 있는 구조를 만들자”라는 데 의견을 모았다. 유니티의 장점은 크게 다음과 같이 살펴볼 수 있다[5][6].

첫째. 툴의 GUI가 아주 직관적이다. WYSIWYG 방식의 툴은 사용자가 직관적으로 내부 에셋들의 위치를 바꾸거나 적용하거나, 에셋의 임포트 등을 매우 손쉽게 가능하며 유니티가 소규모/인디 게임에 폭발적인 인기를 얻게 된 가장 큰 비결이었다. 게임 엔진에서 유저 친화적 UI가 얼마나 중요한지 상징적으로 보여줬으며, 향후의 게임 엔진들에 매우 많은 영향을 주었다.

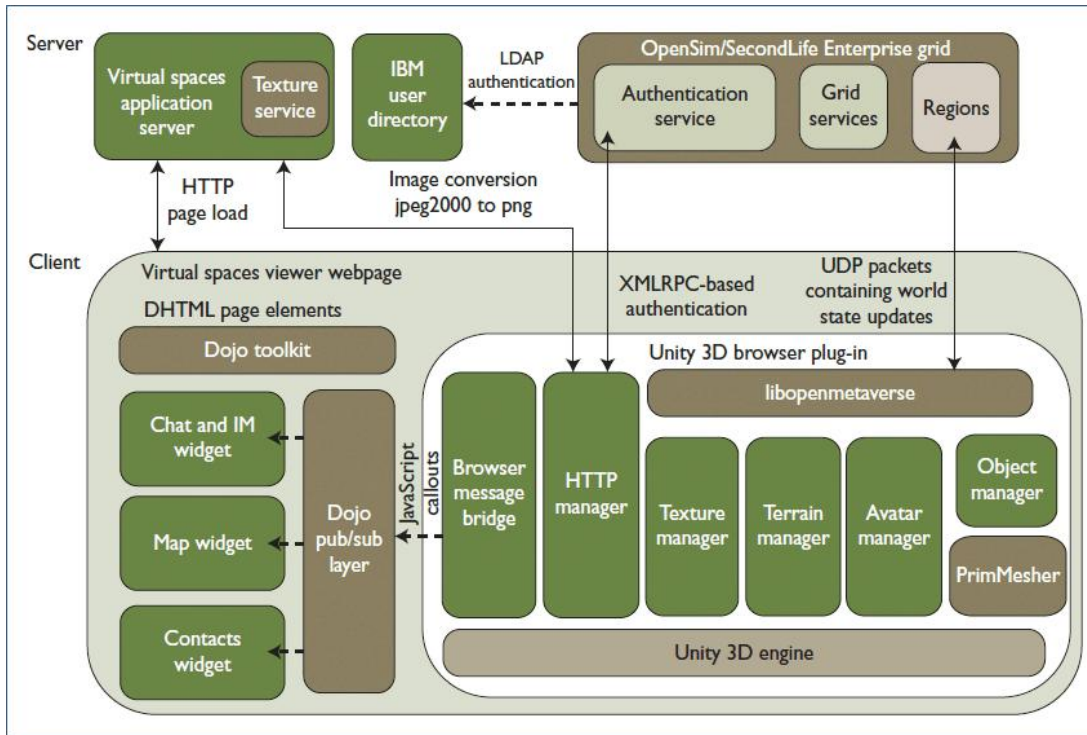
둘째. 버튼 몇 번만 누르면 다양한 플랫폼으로 빌드가 가능하다. 그러나 플랫폼별 특성에 맞게 어느 정도 조정작업 및 별도의 최적화도 필요하다. 플랫폼별로 특징이 다르므로 당연한 일이며 유니티를 쓴다고 해서 그런 조정작업이나 최적화가 자동으로 되지는 않는다. 게임 하나 만들어 놓고 별다른 노력도 없이 다양한 플랫폼으로 출시할 수 있는 건 아니니 반드시 유의해야 한다.

셋째. 인디 게임개발자나 초보 개발자들에게 유리한 이점을 제공한다. 바로 애셋 스토어의 존재다. 각종 리소스부터 스크립트, 플러그인으로 사용할 수 있는 다양한 기능, 미리 만들어 놓은 AI 등 수많은 기능들을 이용자들이 올려서 서로 사고 팔 수 있는 장터를 제공한다. 이 정책은 크게 성공하여 유니티 붐을 일으키는 데 일조했다. 사용자가 많을수록 애셋 스토어에 더 많은 상품들이 올라오고, 그것을 본 사람들은 유니티 사용에 긍정적으로 접근하게 되고, 신규 유입된 사람들이 다시 스토어에 유니티 기능을 만들어 올리는 등 선순환 과정이 형성되어 있다. 애플의 앱스토어가 아이폰 자체의 기능을 넘어서 사용자들에게 매력요소가 된 것과 비슷한 원리다. 그리고 유니티에서 운영하는 인디, 초보자들을 지원하는 프로그램 등 관련된 다양한 행사가 진행된다는 점도 강점이다. 엔진 자체의 확장성과 맞물려 게임 이외에 방송, 애니메이션, 툴 등 3D 게임엔진으로 할 수 있는 각종 에셋들이 존재하며, 찾아보면 이런 것도 있나 싶을 정도로 방대한 생태계를 자랑한다.

넷째. 저렴한 라이선스 비용. 2017 버전 이후 영구 라이선스는 없어졌으나 개인 라이선스는 여전히 무료로 유지하고 있으며, 연매출 20만불 이하일 경우 월 40,000원대, 그 이상은 140,000원의 라이선스 비용을 받는다. 매출이 10만불 이하일 경우는 무료 라이선스를 사용할 수 있다. 이는 각종 엔진들이 무료화하는 데 큰 영향을 끼쳤으며, 게임이 출시되더라도 별도의 라이선스 비용을 청구하지 않는다. 이는 개발의 민주화라는 모토와 맞물려 게임개발의 대중화 및 인디/소규모 개발팀

이 많이 생겨나는 계기를 만들어 냈다.

이러한 장점들 덕분에 벤처 기업이나 소규모 스타트업 창업을 하는 사람들 입장에서는 굉장히 유용하게 사용할 수 있으며, 2D 기능 역시 최신 기능들이 계속 추가되고 있기에 2D, 3D 모두 개발이 가능하다. 특히 소규모/인디 개발에만 한정되던 초기와는 다르게 MMORPG까지 다양하게 확장하고 있는 추세이다. 국내 안드로이드 매출 50위 안의 30~50% 정도가 유니티로 개발되고 있으며, 장르 역시 MMORPG에서 전략/퍼즐/액션까지 다양하게 개발되고 있다[7][8]. 아래 [그림 2]에 Unity3D 구조를 나타 내었다.



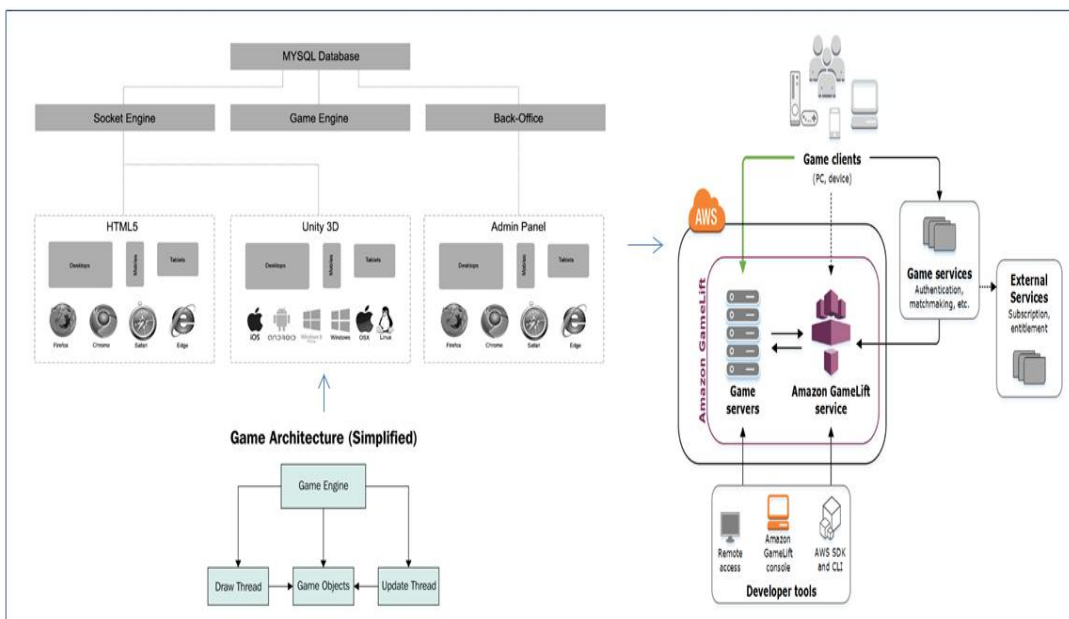
[그림 2] Unity 3D 구조

[Fig. 2] Unity3D Architecture

3. 3D 시뮬레이션 콘텐츠 설계 및 개발

시뮬레이션 게임(Simulation Game)은, 모의 시험 게임이라는 뜻이다. 보다 정확하게는 복잡한 문제를 해석하기 위하여 모델에 의한 실험, 또는 사회현상 등을 해결하는 데서 실제와 비슷한 상태를 수식 등으로 만들어 모의적으로 연산을 되풀이하여 그 특성을 파악하는 일. 즉 하나의 모델을 이용하여 실제와 비슷한 일을 재현하여 시험, 또는 체험한다는 뜻이다. 전반적으로는 현실의 무언가

를 비슷하게 따라하고 재현하는데 중심을 맞춘 게임이라고 볼 수 있다. 광의적으로는 모든 비디오 게임이 시뮬레이션이라 꺼맞출 수는 있으나, 현실적인 요소를 얼마나 충실히 반영 하느냐에 따라 시뮬레이션이나 보다 게임 적이나가 갈린다. 예를 들어 똑같이 자동차를 모는 게임이라고 해도 카트라이더를 시뮬레이션이라고 간주할 사람은 없을 것이다. 그란 투리스모 시리즈의 경우 시뮬레이션이네 아케이드네 하는 논쟁이 주기적으로 활활 타곤 하는데, 대충 팬덤에서는 그란투리스모를 기준으로 보다 현실적이고 복잡하면 심-레이싱으로, 보다 단순하고 화끈하면 아케이드-레이싱으로 구분하곤 한다. 또한 스타크래프트 등의 전략 시뮬레이션이라 불리는 장르가 과연 시뮬레이션이라 할 수 있느냐는 꽤 전통적인 떡밥 중의 하나인데, 그 기원이 군사용 워게임, 즉 전쟁 시뮬레이터였으니 딱히 틀린 말은 아니다. 하지만 원래의 사전적 정의인 모델에 대한 모의 실험이라는 측면에서는 인공지능이 아닌 명령한 대로 무조건 적으로 움직이고 게임상 가상의 유닛들이 많이 등장하는 대부분의 RTS는 시뮬레이션적 요소는 별로 없다는 반론도 일리가 있다. 반면에 클로즈 컴뱃 같이 군사의 사기나 세부적인 고증을 철저히 살린 게임의 경우 훨씬 더 시뮬레이션성이 강하다. 이번 논문에서는 3D 시뮬레이션 콘텐츠를 오픈 SW인 Unity3D API를 C# 및 html5 언어를 이용해서, 설계 및 개발 하고자 한다. 아래 [그림 3]에 3D 시뮬레이션 서비스 구조를 나타 내었다[9][10].



[그림 3] 3D 시뮬레이션 서비스 구조

[Fig. 3] 3D Simulation Service Structure

이번 논문에서는 3D 시뮬레이션 콘텐츠를 공개 SW인 Unity3D API를 C# 및 html5 언어를 이용해서, 설계 및 개발 하고자 한다. 또한 향후 H/W 시뮬레이터 연동을 위한 라즈베리 파이를 이용한

시뮬레이터를 개발 하고자 한다. 이를 통해서 국내 게임 및 3D 시뮬레이션 콘텐츠 산업 발전에 도움이 되고자 한다. 아래 [그림 4]에 서버 접속을 통한 3D 시뮬레이션 콘텐츠 실행 화면을 나타내었다[11][12].



[그림 4] 3D 시뮬레이션 콘텐츠 실행 화면
[Fig. 4] 3D Simulation Content Execution Screen

4. 결론

시뮬레이션(Simulation)은 모의실험이라는 뜻이다. 더욱 정확하게는 복잡한 문제를 해석하기 위하여 모델에 의한 실험, 또는 사회현상 등을 해결하는 데서 실제와 비슷한 상태를 수식 등으로 만들어 모의적으로 연산을 되풀이하여 그 특성을 파악하는 일. 시뮬레이션은 학문적인 분야와 일반인들 사이에서 사용되는 의미에서 다소 차이가 있다. 일반인들에게는 시뮬레이션 게임과 운전 시뮬레이션 장치와 철도 시뮬레이션, 항공기 조종사들이 조종을 연습하기 위해 사용하는 비행 시뮬레이션이 유명하다. 하지만 시뮬레이션의 원래 의미는 이와 차이가 있다. 시뮬레이션은 연구목적에

따라 필요한 부분을 현실에 맞게 구현한 후 이것을 무한에 가까울 정도로 수없이 반복하여 통계적으로 최적 해를 찾아내는 것이다. 이번 논문에서는 3D 시뮬레이션 콘텐츠를 공개 SW인 Unity3D API를 C# 및 html5 언어를 이용해서, 서버기반으로 설계 및 개발 하였다. 또한 향후 H/W 시뮬레이터 연동을 위한 라즈베리 파이를 이용한 시뮬레이터 시스템을 개발 하였다. 이를 통해서 국내 게임 및 3D 시뮬레이션 콘텐츠 산업 발전에 도움이 되고자 한다.

References

- [1] Gurevitch, Leon, The Cinemas of Transactions: The exchangeable currency of GC attractions across audiovisual e Popular Narrative Media, Liverpool University Press, Liverpool, (2010), Vol.2 No.2, pp.143-158.
- [2] Seung-Hoon Lee, A Study on the Status and Activation of Serious Game, Korea Creative Content Agency Research Report 13-11, (2013).
- [3] Eui-Jun Jeong, Hye-Rim Lee, A Study on the Concepts and Categorization of Serious Games Based on Strategic Purposes and Applied Theories, Journal of the Korean Society for Computer Game, (2013), Vol.26 No.3, pp.62.
- [4] Sung-Ho Cho, Jae-Bum Jung, Moon-Gee Choi, A Survey of effectiveness of games for serious games, Journal of the Korean Society for Computer Game, (2009), No.16, pp.20-22.
- [5] Tae Bok Yoon, A Study on the Performance Analysis and Activation of Serious Game, Korea Creative Content Agency Research Report 17-53, (2018).
- [6] KOCCA, "The Definition and Classification of Serious Games." Korea Creative Content Agency Online Teaching Material 008-3, (2011).
- [7] Ye-Sun Bae, Woo-Chun Jun, A Study on Current Status and Improvement Plans of Educational Serious Game for Elementary School Students, AJMAHS (2016), Vol.6, No.3, pp.266-268.
- [8] Hye-Won Han, A Study on Conceptual Definition and Types of Serious Games, Korea Humanities Content Society, (2010) No.19, pp.220.
- [9] Bon-Hong Gu, Study on the Application of ROK Military Virtual Combat Training Simulation using Serious Game, The Korean Society of Mechanical Engineers, spring-fall conference, (2013).
- [10] Lomax, Alyce, Advertising, Disrupted, The Motley Fool, (2010).
- [11] Owen, Rob, TV Q&A: ABC News,'Storage Wars and The Big Bang Theory, Pittsburgh Post-Gazette, (2012).
- [12] Rose, Steve. As seen on TV: why product placement is bigger than ever, The Guardian 24, (2014).