

일반논문 (Regular Paper)

방송공학회논문지 제22권 제1호, 2017년 1월 (JBE Vol. 22, No. 1, January 2017)

<https://doi.org/10.5909/JBE.2017.22.1.87>

ISSN 2287-9137 (Online) ISSN 1226-7953 (Print)

가상환경(VR) 운영체제 프로토타입 연구

김 은 솔^{a)}, 김 지 연^{a)}, 유 은 진^{a)}, 박 태 정^{a)†}

Study on Virtual Reality (VR) Operating System Prototype

Eunsol Kim^{a)}, Jiyeon Kim^{a)}, Eunjin Yoo^{a)}, and Taejung Park^{a)†}

요 약

본 논문에서는 게임 엔진(Unity3D) 상에서 헤드 마운트 디스플레이(HMD)와 손 동작 인식 기술을 이용한 가상 환경 운영 체제(VR OS) 프로토타입을 제안한다. 제안하는 OS는 Unity3D에서 제공하는 실시간 환경 상에 간단한 멀티태스킹 스레드 메커니즘을 구축한 형태로 제작되었다. 제안하는 가상 환경 운영 체제는 손 동작 인식 장치(Leap Motion)로부터 입력을 받아서 키보드와 마우스 역할을 수행하며 헤드 마운트 디스플레이(Oculus Rift DK2)를 통해 출력을 제공한다. 테스트 결과, 제안하는 시스템은 360도 가상 공간 내에서 보다 넓고 몰입도가 높은 작업 환경을 제공하였다.

Abstract

This paper presents a prototype for virtual reality operating system (VR OS) concept with head mount display (HMD) and hand gesture recognition technology based on game engine (Unity3D). We have designed and implemented simple multitasking thread mechanism constructed on the realtime environment provided by Unity3D game engine. Our virtual reality operating system receives user input from the hand gesture recognition device (Leap Motion) to simulate mouse and keyboard and provides output via head mount display (Oculus Rift DK2). As a result, our system provides users with more broad and immersive work environment by implementing 360 degree work space.

Keyword : Virtual Reality, Unity3D, Leap Motion, Oculus, Operating System

a) 덕성여자대학교 디지털미디어학과 Visual Media 연구실(Visual Media Lab., Department of Digital Media, Duksung Women's University)

† Corresponding Author : 박태정(Taejung Park)

E-mail: tjpark@duksung.ac.kr

Tel: +82-2-901-8339

ORCID: <http://orcid.org/0000-0001-5118-3271>

※ 본 연구는 덕성여자대학교 2015년도 교내연구비 지원에 의해 수행되었음.

· Manuscript received October 17, 2016; Revised November 24, 2016;

Accepted January 5, 2017.

1. 서 론

최근 가상 환경(VR) 기술의 발전에 따른 보급과 확산으로 이 분야에 대한 잠재력에 대해 많은 기대와 관심이 발생하고 있다. 가상 환경 기술은 여러 유형이 연구되어 왔으나 가장 상업적으로 높은 가능성을 보이고 있는 형태는 머리

의 움직임을 추적하고 360도 전방향으로 가상 공간을 양쪽 눈으로 재현할 수 있는 HMD (Head Mounted Display) 장치와 손의 움직임을 추적할 수 있는 모션 인식 장치나 컨트롤러 등으로 구성된다.

이러한 가상 환경 기술의 적용 분야는 게임, 가상 공연 실황 중계, 유명 여행지에 대한 가상적인 체험 등 주로 엔터테인먼트 분야에 집중되고 있다.

본 연구에서는 가상 환경 기술에 대한 좀더 실용적인 접근으로 HMD(Oculus Rift DK2^[2])와 모션 인식 장치(Leap Motion^[3])를 컴퓨터의 기본적인 입출력 장치로 활용하는 가상 환경 기반 운영 체제(OS)에 대한 프로토타입을 제작하고 그 활용성을 논의한다. 제안하는 운영 체제 프로토타입은 가상 환경 기술 개발에 용이한 게임 제작 엔진인 Unity3D^[4]의 기본적인 메커니즘을 활용하면서 이 위에서 멀티태스킹 운영체제 구현을 위해 간단한 멀티스레드 커널^[5]을 제작했다. 이렇게 제작한 운영 체제 프로토타입은 일반적으로 GUI 환경에서 공통적으로 제공되는 메모장, 그림장, 미디어 플레이어, 계산기 등 몇몇 응용 프로그램을 멀티태스킹 방식으로 실행할 수 있다.

구현 결과, 주로 2차원 모니터를 출력 장치로 활용하고 키보드와 마우스 등을 입력 장치로 활용하는 일반적인 운영 체제 환경에 비해서 작업의 몰입감은 물론, 360도 공간모두를 활용할 수 있는 장점을 확인할 수 있었다. 예를 들어

현재 사용하거나 빈번하게 사용하는 아이콘(응용프로그램, 파일, 데이터)은 전방에 배치하고 사용 빈도가 높기는 하지만 짧은 시간 동안 잠시 사용해야 하는 아이콘(예를 들어, 달력, 시계, 현재 재생 중인 미디어 플레이어)은 즉시 시야에 들어오지는 않지만 고개를 돌려 바로 확인할 수 있는 가상 공간의 상단(천장 부분)이나 좌우에 작업의 특징에 따라 배치할 수 있고 사용 빈도가 낮거나 당장 사용할 필요가 없는 아이콘은 후방에 배치함으로써 작업 효율을 높일 수 있었다.

유사한 연구로, 2016년 5월에 Google에서 Daydream이라는 VR OS 개념^[6]을 제시한 바 있으나 본 연구는 이미 2015년에 구현이 완료되었고 2016년 1월에 국내 학회^[7]에 결과 중심으로 확장 초록(extended abstract) 형식으로 이미 발표된 바 있다. 그러나 학회 발표에서는 내부 구조나 구현 세부에 대해서는 전혀 논의되지 않았으며 본 논문을 통해 그 자세한 내용을 소개, 논의하고자 한다. 또한 그림 1에서 본 시스템의 실제 시연 상황을 제시했으며 실제 시연 동영상은 온라인에서 확인할 수 있다^[1]. 그림 1에서는 HMD (Oculus Rift)의 라이브 스테레오 비전을 통해서 본 제안하는 시스템의 실제 이용 상황을 표시하고 있다. 이 그림에서 제시한 것처럼 사용자가 HMD와 제스처 인식 장치(Leap Motion)를 착용하고 (오른쪽 아래) 음악 재생을 위해서 가상 공간 내에서 아이콘을 선택

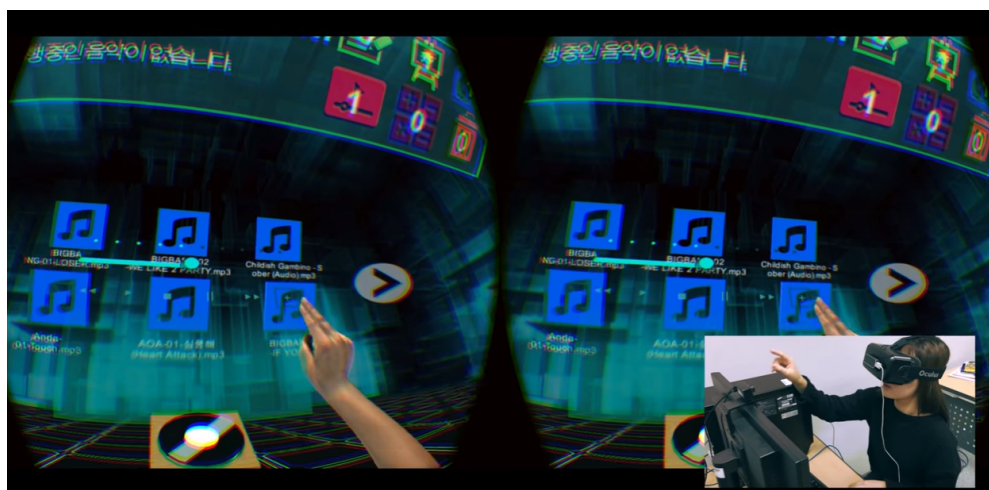


그림 1. HMD(Oculus Rift)의 라이브 스테레오 비전을 통해서 본 제안하는 시스템의 실제 이용 상황
Fig. 1. Live stereo vision via HMD (Oculus Rift) using the proposed system

하는 모습(실제 작동 동영상은 온라인 상^[1]에서 확인 가능)을 볼 수 있다.

II. 시스템 구조

그림 2에서는 전체 시스템 구조를 제시한다. 제안하는 시스템은 C#으로 구현되었으며 광선 추적(raycasting)와 같은 기본 제공 메커니즘과 일반적으로 PC 시스템 등에서 대략 초당 60회 호출이 되는 Update() callback 함수^[8] 등과 같은 Unity3D의 기본적인 메커니즘을 활용할 수 있도록 구현하였다. 이러한 메커니즘 위에 간단한 선점형 스케줄링(Pre-emptive Scheduling) 기반 스레드 관리자와 스케줄러가 구현되었으며 시스템 내부에서 파일 형태로 데이터를 저장, 로드할 수 있는 파일 관리자도 구현되었다. 구현한 요소 대부분은 Unity3D의 기본 객체인 GameObject의 상속을 받음으로써 Unity3D가 제공하는 기본적인 기능들과 메커니즘을 활용할 수 있도록 하였다.

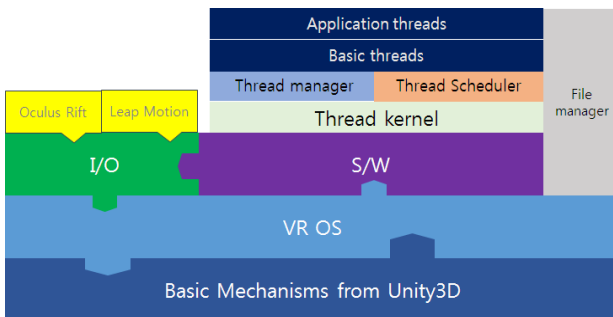


그림 2. 시스템 구조
Fig. 2. System architecture

1. I/O 하드웨어

그림 3은 제안하는 시스템의 주요 입출력 장치인 HMD와 제스처 인식 장치를 제시한다. HMD는 Oculus Rift DK2를 사용했으며 사용자의 손 및 손가락 인식을 위해 전방에 Leap Motion을 부착했다. Leap Motion에서 인식된 사용자의 양손은 그림 1, 그림 5, 6, 7에서 제시된 것처럼 가상 현실에서 3차원 모델로 표현된다.

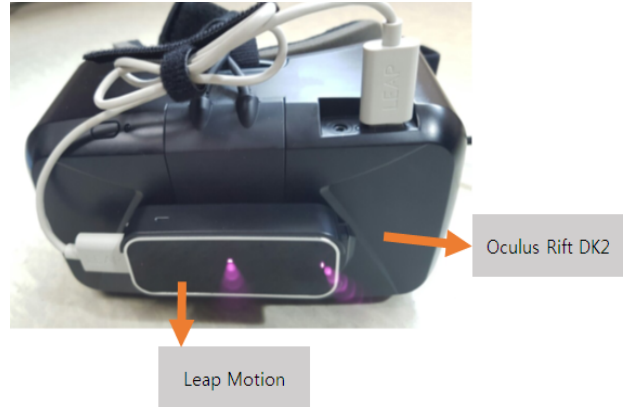


그림 3. 입출력 하드웨어
Fig. 3. I/O Hardware

2. 스레드 운영 및 관리

제안하는 시스템에서 모든 스레드는 Unity3D 환경에서 Instantiate 명령을 통해 ThreadBase 클래스의 상속을 받은 스레드 프리팹(prefab)의 인스턴스로 생성된다. 프리팹이란 Unity3D에서 일종의 실시간 클래스 또는 템플릿 역할을 하는 객체로 코드를 통해 Instantiate 명령으로 실시간으로 인스턴스를 생성할 수 있다. 일반적인 스레드 스케줄링 방식대로 Thread Manager는 Thread들의 생성과 삭제를 담당하며 각 스레드의 종류에 따라 상이한 우선순위를 지정할 수 있다. 제안하는 시스템에서 모든 스레드는 Unity3D 환경에서 Instantiate 명령을 통해 ThreadBase 클래스의 상속을 받은 스레드 프리팹(prefab)의 인스턴스로 생성된다. 프리팹이란 Unity3D에서 일종의 실시간 클래스 또는 템플릿 역할을 하는 객체로 코드를 통해 Instantiate 명령으로 실시간 생성된다. 그림 4에서 제시한 것처럼 Thread Manager는 Thread들의 생성과 삭제를 담당한다. 각 Thread는 종류에 따라 상이한 우선순위를 가지며 이 우선 순위에 따라 실행 빈도가 결정된다. 이렇게 각각의 우선순위가 할당된 후에 Thread Scheduler가 C#으로 구현된 Thread의 실제 내용을 읽어 실행을 담당한다. 생성된 Thread들의 정보 공유를 위해 Thread Manager와 Thread Scheduler는 생성된 Thread들의 List를 지속적으로 공유하며 쌍방향적인 구조를 가지고 있다.

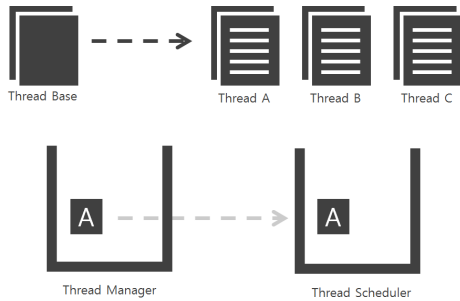


그림 4. Thread Base(상단) 및 Thread Manager와 Thread Scheduler의 관계(하단)

Fig. 4. Thread Base (upper) and the relationship between the Thread Manager and the Thread Scheduler (lower)

3. 제스처 구조

360도 가상 공간 내에서 제스처를 이용해서 효율적인 작업을 수행하기 위해서 Leap Motion으로 인식되는 제스처를 설계하였고 직관적으로 작업을 하면서 복잡성을 줄이기 위해서 3가지(app 이동, 제거, layer 전환) 제스처로 정리하였다. 그림 4, 5, 6은 이러한 세 가지 제스처를 제시한다. App 이동 시(그림 5)에는 1) 양손이 모두 Scene상에 존재하고, 2) Right Hand의 Palm에서 나오는 Ray가 인식 영역에 닿아야 한다는 조건을 적용하고 이 두 조건을 모두 만족할 때 Circle Gesture를 수행 하게 되면 이동이 가능한 상태가 되도록 하였다. App 삭제 시(그림 6)에는 양손이 모두 Scene상에 존재하고, Right Hand의 Palm Ray가 삭제하고자 하는 App Object에 일정 시간 이상 닿을 때, Right Hand에서 Swipe Gesture를 수행하게 되면 app을 휴지통(Trash Box)로 보내도록 하였다. 또한, 양손을 모두 Scene에서 인식하게 하고, 박수를 치면 Layer가 전환되도록 하였다(그림 7).

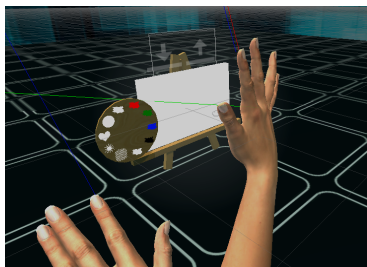


그림 5. 앱 위치 이동을 위한 제스처
Fig. 5. Gesture to move the location of app

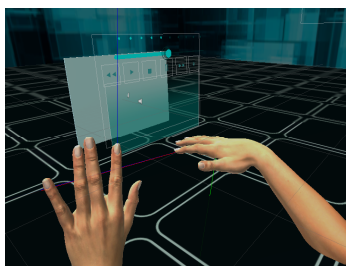


그림 6. 앱 제거를 위한 제스처
Fig. 6. Gesture to remove app



그림 7. 레이어 변경을 위한 제스처
Fig. 7. Gesture to switch layers

4. Layer 설계

앞서 언급한 제스처로 마우스 또는 키보드 입력을 대체하기 위해서 Leap Motion 장치로 인식한 손의 위치에서 ray를 투사하고 이 ray가 닿는 layer에 따라 여러 상호 작용을 수행할 수 있도록 설계하였다. 이 시스템에서는 Main screen layer, Thread layer, Folder layer, File open layer, File add layer 등 총 5개의 layer를 적용한다.

Main Screen Layer(그림 8)는 각 app과 검색 기능의 아이콘을 표시하는 layer로 Leap Motion으로 인식된 검지 손가락(pinch)에서 ray를 투사해서 아이콘을 선택한다. 이 상태가 일정 시간이 지속되면 선택으로 판정되고 app을 생성할 수 있는 Thread layer 형태로 배경이 변하게 된다. 이렇게 변화된 Thread layer(그림 9)에서는 배경 화면의 아이콘들이 일시적으로 사라지고 app이 실행되는 화면으로 구성된다. 이 layer에서는 검지 손가락으로 선택한 app(thread)를 손으로 쥐어 다음(grab) 고개를 돌려 원하는 위치를 결정한다. 그 후 손을 펴면 해당 위치에 app이 이동한다. 여러 app을 실행하는 과정일 때에는 사용하던 app(즉, focus가 적용된 app)이 아닌 다른 app을 바라보면 focus가 옮겨지면서 현재 사용 중인 app으로 강조된다. Main screen layer에서 단순화된 파일 관리자로 구현된 Searcher(그림 13)를 선택하면 Folder Layer(그림 10)로 이동한다. 이 때 Main Screen Layer에서 볼 수 있던 아이콘들이 사라지고 Searcher에서 선택이 가능한 확장자 아이콘이 나타나게 된다. 확장자는 현재 제작한 app에 맞춰 txt, png, mp3, avi로 구성된다.

Folder Layer 환경에서 원하는 확장자를 선택하면 File Open Layer(그림 11)로 전환된다. Folder Layer에 있는 Searcher 아이콘은 사라지고 선택한 확장자에 대한 File

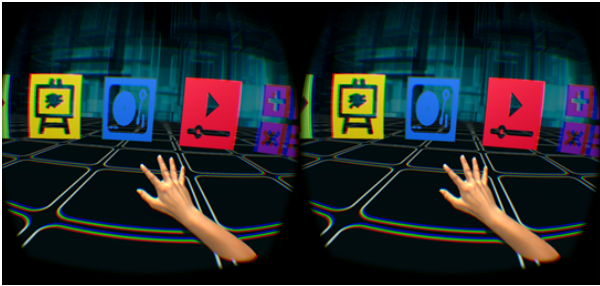


그림 8 . 메인 화면 레이어
Fig. 8. Main screen layer

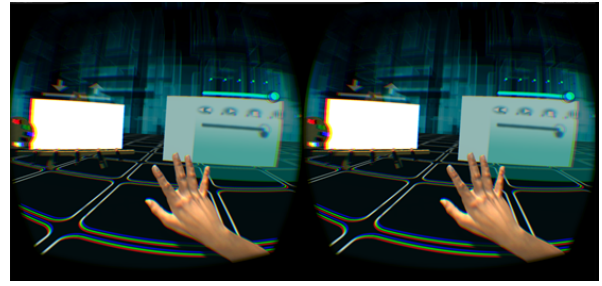


그림 9 . 스레드 레이어
Fig. 9. Thread layer



그림 10. 폴더 레이어
Fig. 10. Folder layer

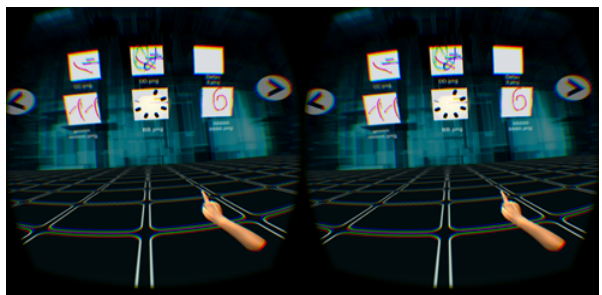


그림 11. 파일 열기 레이어
Fig. 11. File open layer

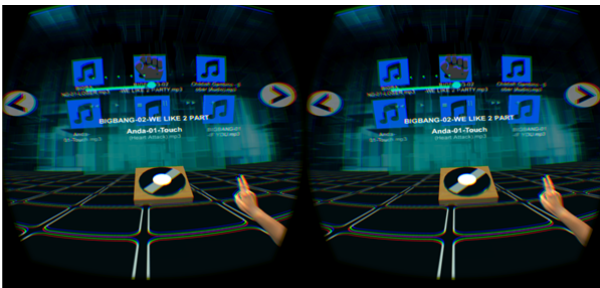


그림 12. 파일 추가 레이어
Fig. 12. File add layer

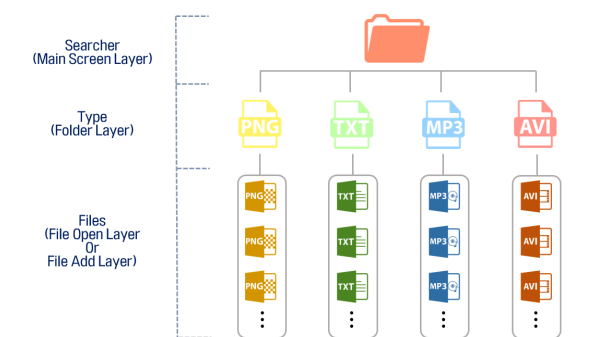


그림 13. Searcher 구조
Fig. 13. Searcher Architecture

List가 상단에 나타난다. 확장자에 따라 분류된 파일이 정렬되어 나오면 페이지를 넘기며 원하는 파일을 선택할 수 있다. 파일 선택 역시 검지 손가락(Index)으로 표시하면 효과와 함께 로딩 과정이 표시되고 로딩이 완료되면 확장자와 관련된 app이 생성된다.

마지막으로 File Add Layer(그림 12)는 Main Screen Layer의 생성된 app에서 File을 추가하는 Layer다. File Add Layer에서는 app이 선택(highlight)된 상태에서 상단에 고를

수 있는 File List가 표시된다. 각 app에 따라 ‘+’ 버튼 또는 Load 버튼을 누르게 되면 해당하는 확장자 File의 List가 표시된다. List의 한 페이지에는 파일이 총 6개씩 나열이 되고, 양 옆에 있는 화살표를 통해 다음 페이지의 File을 넘길 수 있다. 파일 선택은 검지 손가락(Index)으로 선택하면 강조 표시되고 로드 중임을 표시하는 애니메이션이 재생된다. 선택이 완료되면 주먹(Grab)을 쥐었다 피는 것으로 선택한 File을 Load 또는 Add 작업이 적용된 app이 생성된다.



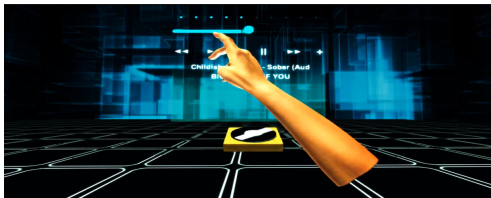
(a) Memo Application



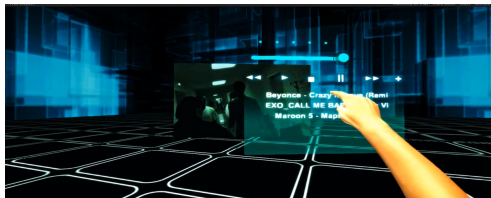
(b) Paint Application



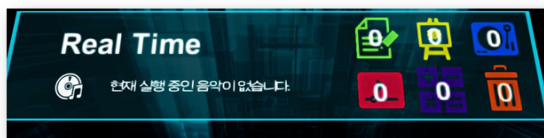
(c) Music Player Application



(d) Media Player Application



(e) Calculator Application



(f) Left: Widget Panel. Right: Trash can

그림 14. (a)-(e): 구현된 기본 앱 프로토타입 (f): 기타 기능

Fig. 14. (a)-(e): Basic application prototypes implemented (f): Miscellaneous features

5. 응용 프로그램

제안한 시스템의 테스트를 위해서 2.4에서 논의한 thread 운영 구조 위에 기본적인 5가지 app(메모장, 페인트, 음악 플레이어, 미디어 플레이어, 계산기)은 물론 widget 패널과 휴지통을 구현하였다(그림 14). 이러한 app들은 앞서 설명한 360도 작업 환경 내에서 작업 편의를 위한 최적의 위치에 배치할 수 있고 제스처를 통한 조작을 통해 멀티태스킹 방식으로 실행되며 기본적인 작업을 수행할 수 있다.

III. 결 론

본 논문에서는 게임 엔진 위에 기본적인 멀티태스킹 환경을 구축하고 대표적인 몇 가지 애플리케이션을 구현한 후 HMD와 제스처 인식 장치를 이용해서 360도 3차원 가상 공간 속에서의 OS 환경을 구현한 프로토타입의 구현과 결과를 소개한다. 테스트 결과 현재 HMD 기술이 가지고 있는 제약(해상도, 어지러움 등)에도 불구하고 상당히 작업 효율이 높은 컴퓨팅 환경을 구현할 수 있었다. 현재 HMD에서 일반적으로 발생하는 어지러움이나 해상도와 관련된 문제는 향후 HMD 관련 하드웨어 기술이 발전하면서 자연스럽게 해결될 것으로 볼 때, 제안하는 프로토타입에서 확인한 사용성 측면의 장점이 향후 크게 부각될 것이라고 판단한다. 효율성 관점에서, 한정된 2차원 공간을 제공하는

기존 모니터를 사용할 때보다 작업 공간의 면적이 가상 공간에서 넓어진다는 측면에서의 공간 활용도 향상은 물론, 실제 물리적 작업 공간에서 집중력을 방해하는 요소로 작용될 수 있는 2차원 모니터 이외의 공간을 통제할 수 있다는 측면에서 보다 깊은 몰입도를 얻을 수 있었다. 향후 이러한 장점으로 인해 향후 관련 VR 기술의 발전과 함께 본 연구에서 제안하는 방식의 가상 OS가 널리 사용될 것으로 전망한다.

참 고 문 헌 (References)

- [1] System demo video on YouTube (2016). <https://www.youtube.com/watch?v=xHFb1gXwUVQ> (accessed Oct., 13, 2016)
- [2] Oculus Rift DK2 official site (2016). <https://www3.oculus.com/en-us/dk2/> (accessed Oct., 13, 2016)
- [3] Leap Motion official site (2016). <https://www.leapmotion.com/> (accessed Oct., 13, 2016)
- [4] Unity3D official site (2016). <https://unity3d.com> (accessed Oct., 13, 2016)
- [5] P. B. Galvin, G. Gagne, and A. Silberschatz, "Operating System Concepts". John Wiley & Sons Inc, 2013.
- [6] Google Daydream official site (2016). <https://vr.google.com/daydream/> (accessed Oct., 13, 2016)
- [7] Eunjin Yoo, Eunsol Kim, Jiyeon Kim, and Taejung Park, "Prototype of Virtual Reality Operating System and Its Task Environment Using Oculus Rift and Leap Motion", Extended abstracts of HCI Korea 2016, pp. 221-222, 2016.
- [8] Unity3D documentation (2016). <https://docs.unity3d.com/ScriptReference/MonoBehaviour.Update.html> (accessed Oct., 13, 2016)

저 자 소 개



김 은 솔

- 2016년 2월 : 덕성여자대학교 디지털미디어학과 (공학사)
- ORCID : <http://orcid.org/0000-0002-1111-6418>
- 주관심분야 : 컴퓨터그래픽스, VR, 영상 처리

저 자 소 개



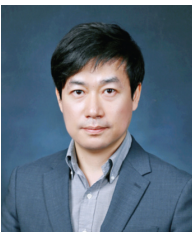
김 지 연

- 2016년 2월 : 덕성여자대학교 디지털미디어학과 (공학사)
- ORCID : <http://orcid.org/0000-0001-7708-6883>
- 주관심분야 : 컴퓨터그래픽스, VR, 소프트웨어공학



유 은 진

- 2016년 2월 : 덕성여자대학교 디지털미디어학과 (공학사)
- ORCID : <http://orcid.org/0000-0002-2387-1277>
- 주관심분야 : 컴퓨터그래픽스, VR, 모션 캡처



박 태 정

- 1997년 2월 : 서울대학교 전기공학부 (공학사)
- 1999년 2월 : 서울대학교 전기공학부 대학원 (공학 석사, 반도체 전공)
- 2006년 8월 : 서울대학교 전기컴퓨터공학부 대학원 (공학박사, 컴퓨터 그래픽스 전공)
- 2006년 9월 ~ 2013년 2월 : 고려대학교 연구교수
- 2013년 3월 ~ 현재 : 덕성여자대학교 디지털미디어학과 조교수
- ORCID : <http://orcid.org/0000-0001-5118-3271>
- 주관심분야 : 컴퓨터그래픽스, 병렬처리, 게임 물리, 수치해석, 3차원 모델링