

Unity 3D게임 및 화면분할 API를 이용한 안드로이드 Controller에 관한 연구

장은겸*, 서형준**, 이창현⁰, 최용락**

*장안대학교 인터넷정보통신과

**대전대학교 컴퓨터공학과

e-mail: jangeg@jangan.ac.kr*, floralcustom@gmail.com**, lch47@naver.com⁰, yrchoi@dju.ac.kr**

A Study on the Android Controller based on Screen Split API and Unity 3D Game

Eun-Gyeom Jang*, Hyung-Jun Seo**, Chang-Heon Lee⁰, Yong-Rak Choi**

*Dept. of Internet Communication, Jangan University

**Dept. of Computer Engineering, Deajeon University

● 요약 ●

모바일게임이 발전하면서 여러 디스플레이 제품과 유동적 연결이 가능하게 되었고 이를 통한 쾌적한 게임 환경이 가능하게 되었으나, 여전히 게임의 컨트롤러가 게임 진행 화면을 가려 몰입감이 떨어진다. 그렇기 때문에 Unity 3D 엔진을 이용한 First-person Shooter 게임 제작과 게임 컨트롤을 위한 Android Device Controller를 이용하여 외부 디스플레이 연결 시 컨트롤러와 게임화면이 분할되는 시스템을 개발하였다. 사용자가 모바일 디바이스를 게임 컨트롤러로 사용하여 콘솔게임 환경과 같은 편리한 조작이 가능하게 하는 효과가 있다.

키워드: 안드로이드(android), 컨트롤러(controller), 화면분할(split screen), 게임(game)

I. Introduction

스마트폰의 대중화에 의한 모바일 기기 및 환경이 발전하면서 모바일 게임도 PC게임 또는 콘솔게임과 같은 성능을 보이고 있다[1]. 최근 모바일 디바이스는 초기 모델에서 구현이 불가능했던 3D엔진 및 고성능 그래픽 처리가 가능하다.

모바일 디바이스의 게임 플레이 컨트롤러는 작은 공간에서 일정한 위치에 디스플레이되며 게임을 컨트롤할 수 있도록 한다. 일반적인 컨트롤러는 과거부터 현재 까지 여러 변화를 거치며 사용자가 게임을 진행하는데 편리함을 제시하였다.

작은 화면을 제공하는 모바일 디바이스에서 게임을 제어하는 컨트롤러를 보다 효율적이고 효과적인 컨트롤을 제공할 수 있도록 본 논문에서는 D-pad(십자키)뿐만 아니라 조이스틱형식의 아날로그 스틱과 동작인식 컨트롤러, 터치스크린을 통한 컨트롤러와 같은 다양

한 컨트롤러의 장점을 통합한 안드로이드 디바이스 게임 컨트롤러를 제안한다. 모바일 게임을 진행하면서 작은 화면과 불편한 조작감을 해소하는 어플리케이션을 제작하기 위해 게임화면과 컨트롤러를 분리하여 게임화면은 모니터 또는 TV에 송출하고 게임을 위한 컨트롤러는 단말을 통해 표시한다. 즉, 모바일 디바이스 자체가 다양한 구성이 가능한 게임 컨트롤러의 역할을 수행하도록 하는 방법이다.

안드로이드 디바이스 컨트롤러는 모바일 게임을 즐기는 사용자에게 안드로이드 디바이스를 이용한 최적의 게임 컨트롤러를 제공하며 게임에 편리성을 줄 수 있다. 또한 모바일의 작은 화면에서 벗어나 눈의 피로를 줄일 수 있고 게임 이외의 어플리케이션에 유용한 추가 디스플레이를 제공하기 위해 본 연구를 수행하였다.

II. Preliminaries

1. Related works

1.1 Unity 3D 게임 엔진

게임 엔진 시스템은 1980년대에도 존재하였다. 게임 엔진은 그래픽 작업을 위한 ‘렌더러’에서 시작 되어 현재에는 하나의 게임만을 위한 게임 엔진들도 개발되었다. 1990년 중반의 이드(id) 소프트웨어의 First person shooter 게임인 둠(DOOM)과 퀘이크가 인기를 끌며 개발자인 ‘존 카맥’은 게임 윌펜슈타인3D, 둠, 퀘이크의 소스를 일종의 셰어웨어(shareware) 형태로 배포하며 게임 회사들에게 로열티를 받았다. 이를 시작으로 게임 엔진의 상용화가 시작되었고 퀘이크엔진, 언리얼엔진, 크라이엔진 등과 같은 여러 게임 엔진이 개발되었다. 현재는 게임 개발 뿐 아니라 애니메이션, 가상현실, 3D모델링 등의 분야에서도 활용되고 있다. 위와 같은 여러 분야에서 활용 가능한 Unity 게임 엔진은 타 엔진들에 비해 빠른 프로그램 제작과 간편성, 멀티 플랫폼의 지원으로 모바일 게임 개발에 편리성을 제공한다.

1.2 Miracast

Wi-Fi를 활용한 무선으로 디스플레이를 연결하여 스마트폰, 노트북 등에 사용되는 기술이다. Intel사의 WiDi 기술이 Miracast로 이전되어 사용되고 WiDi와 Miracast는 상호 호환이 가능하다. 영상 뿐만 아니라 음성데이터도 함께 전송이 가능하지만 무선통신 기술의 특성상 연결된 기기의 영상이나 소리가 끊기는 현상이 자주 발생한다. 현재 인텔 WiDi 응용 프로그램과 관련 리시버 인증 프로그램의 마케팅 및 개발은 중단된 상태이며 Miracast를 통한 화면 복제 방식의 다중 디스플레이가 구현되어있다.

III. The Proposed Scheme

1. System Configuration

시스템 구성은 Presentation API의 기본적인 구성과 개발이 완료된 전체 어플리케이션의 기본 구조 및 게임과 컨트롤러 구조를 서술하였다. 추가적으로 가상키보드 어플리케이션을 개발하여 시스템 구성과 동작원리를 설명하였다.

1.1 Presentation API 기본 구성

Presentation API는 외부 디스플레이를 사용하기 위한 API이며 Presentation Class는 Dialog를 상속하여 구현되어 있다. 일반적인 Activity의 set Content로 설정되는 layout은 단말의 기본 LCD로 출력되지만 Dialog는 Display 객체를 활용하여 자신이 송출될 디스플레이를 설정 할 수 있다. 다만, 몇 가지 제약사항이 있다. 첫째로 Dialog를 상속함에 따라, Activity의 Life Cycle을 따라가지 못한다. 그로인해 신규 개발은 크게 문제가 없으나, 포팅하는 경우에는 Activity Life Cycle에 알맞게 설정되어 있는 로직을 변경해야한다. 둘째로 Presentation 내부에 설정되는 디스플레이를 중도 변경할 수 없다. 디스플레이가 변경되면 새로운 Presentation 객체를 생성해

야 하는데, 새로운 객체를 생성함으로써 인하여 기존에 사용 중이던 데이터에 대한 Sync를 맞추는 작업이 추가적으로 필요하다. 추가적인 작업이 없을시 무조건 어플리케이션을 재시작 해야 한다. 셋째로 해당 API가 Jelly Bean(API 17)이상에서만 동작한다. Jelly Bean(API 17)이하의 단말에선 구현 자체가 불가능하다.

1.2 Application 기본 구조

그림 1 과 같이 Application이 실행되면 최초 Gam ePadView를 포함한 MainActivity가 실행되고 이후 MainActivity는 Present를 실행하여 PresentGame을 통해 UnityActivity를 불러온다. 실제 게임을 실행하기 위해 PresentGame을 실행하는데, 이때부터 Unity를 통해 게임이 실행된다.

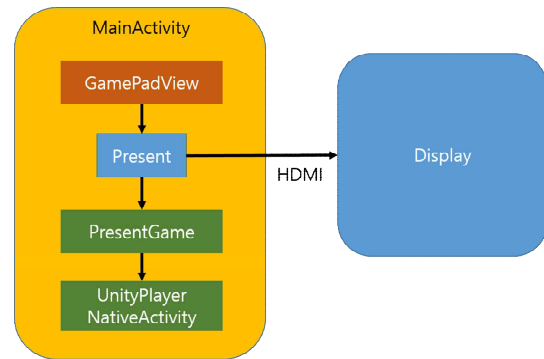


Fig. 1. Application Configuration

Present는 Android 단말과 디스플레이를 구성하는데, Dialog형태를 가지고 있기 때문에 게임의 실행에 따른 절치는 MainActivity에서 해주어야 하고, 각 인자 데이터를 전달해줘야 한다. 최초 시작 시 연결된 디스플레이를 확인하고 Present를 실행하여 GamePadView를 연결한다. Unity로 전달을 위해 Intent를 활용하여 전달한다[2]. MainActivity는 데이터 전달과 Lifecycle 관리, 게임패드 Interface 연동 등의 작업을 하는데, 문제가 되는 부분은 기존 Activity의 Lifecycle을 따르게 만들어진 로직을 Dialog로 만들어 일부 Lifecycle은 생략 및 통합하여 구현되었다.

1.3 First-person shooter Game과 Controller 구조

그림 2는 전체적인 UML 구조 다이어그램이다. 외부 디스플레이를 통해 구현된 View를 Presentation을 통해 출력한다. 단말에 연결된 외부 화면은 Presentation의 Display 객체를 통해 인식할 수 있다. 단 기본 단말 LCD 역시 Display를 생성할 수 있기 때문에 Presentation은 기본 단말에서도 볼 수 있다. 또한 Present내에서 외부 Display가 연결되거나 제거 변경 Event를 Listener를 통해 받을 수 있고, 각 Display 객체를 가져올 수 있다.

Unity 3D 엔진을 사용하여 만든 FPS게임은 Unity의 통합 개발 환경 덕분에 스테이지 디자인, 스크립트 코딩, 디버깅, 빌드 및 테스트 등의 일련의 개발 프로세스가 모두 Unity에서 이뤄진다. 캐릭터 모델과 스테이지 모델, 오브젝트 등의 리소스는 Unity에서 제공하는 Asset store에서 공개되어있는 무료 Asset을 사용한다[3]. C#을 기반

으로 MonoDevelop 개발 환경에서 개발을 한다. 화면 분할과 컨트롤러의 구동을 확인하기 위한 목적의 게임으로써 플레이어는 스테이지에서 무기를 획득하고 적과 대항하여 교전을 펼친다[4]. 게임 플레이어의 임무 완료 조건은 존재하지 않고 체력 값 또는 갈증 값이 0이 되면 플레이어의 캐릭터는 죽고 게임은 재시작 된다.

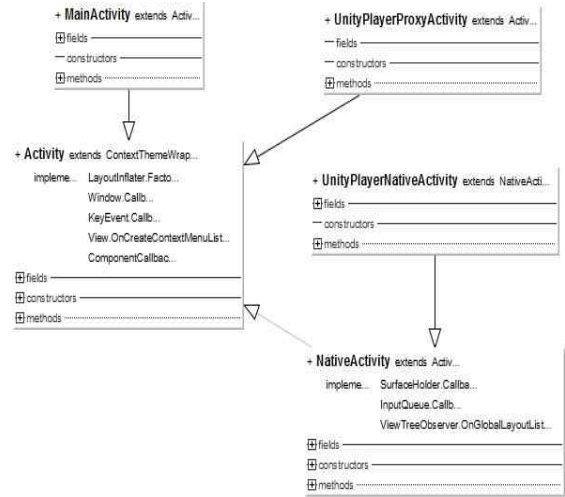
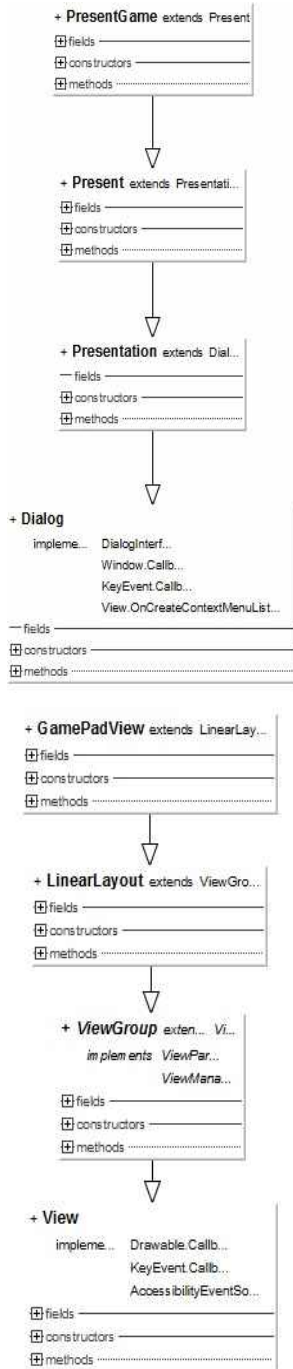


Fig. 2. Application Full Configuration

안드로이드 단말에서 구동될 게임 컨트롤러는 시점을 조종하는 십자키 버튼과 캐릭터의 움직임을 조종할 십자키 버튼을 중심으로 뛰기, 달리기, 무기교체, 무기발사 등의 기능 버튼들로 구성되어있다. 안드로이드 디바이스를 가로로 들었을 때 양손의 엄지손가락만으로 조작이 가능하도록 버튼의 위치를 배치하였다. 버튼의 위치는 자유롭게 조정 가능하다.

2. Implementation

안드로이드 디바이스가 HDMI를 통해 디스플레이와 연결이 되었을 때 그림 3과 같이 안드로이드 디바이스의 LCD 화면이 변경된다[5]. 각 버튼은 터치 입력이 생겼을 때 진동을 일으키게 하여 입력 반응을 확인 할 수 있는 효과가 있다.



Fig. 3. Android Controller

안드로이드 애플레이터 환경에서 구동했을 경우 게임 조작 컨트롤러는 게임 화면과 함께 그림 4와 같이 디바이스에 출력되고 각 버튼의 기능은 화면 분할 상태의 버튼과 일치한다[6]. 버튼 위치의 차이는 조정 가능하며 단독 실행 시 게임 플레이 환경에 최적화 하였다. FPS 게임을 화면 분할 상태에서 송출한 디스플레이 화면은 PC에서 구현한 것과 같이 그림 5의 화면 출력을 보인다. 외부 디스플레이와 연결 시 단말에 나타나는 컨트롤러와 단말 디스플레이로 게임을

실행 시 오버레이 되어 나타나는 컨트롤러는 UI 및 컨트롤 방식을 다르지만 단일 Interface로 사용하기 위해 게임패드 인터페이스를 implements하여 사용하도록 하였다. 안드로이드 디바이스의 성능에 따라 프레임 차이가 발생하는데, 테스트 모델인 LG전자 Optimus G Pro의 경우 30-40 FPS(Frame per Second)사이의 값이 나온다.



Fig. 4. Android Stand-alone



Fig. 5. Split Screen Output Display

IV. Conclusions

본 논문은 모바일 게임의 작은 화면을 벗어나서 게임에 대한 몰입을 높이기 위해 Presentation API를 사용한 화면 분할 기법을 제안하였다. 기존의 모바일 게임들은 게임 컨트롤러가 화면의 일부분을 가려서 몰입하는데 있어서 제약이 되는 문제가 있다.

화면 분할 기법은 모바일 게임뿐만 아니라, 스마트폰으로 문서 또는 메일 작성 시 키보드로 인해 가려지는 화면을 외부 디스플레이로 대체하고 키보드는 스마트폰에서 사용되도록 활용할 수 있으며, 이로 인해 스마트폰이라는 작은 화면의 제한을 극복할 수 있다. 스마트폰 게임의 전망을 보면 그래픽은 시장 경쟁력을 가지기 위한 중요한 요소이기에, 화면분할 기법으로 그래픽의 퀄리티를 강조하는 부분에 활용할 수 있다.

향후 연구로는, 케이블을 이용하여 외부 디스플레이와 연결되는 것이 아닌 60GHz 대역 초고속 무선전송기술[7]을 이용하여 통신거리 및 데이터 전송률을 높여 빠른 반응속도의 게임 컨트롤러 및 화면 분할을 이용한 어플리케이션 개발 및 개선하는 것이다.

References

- [1] MobileIndex, <http://www.mobileindex.com/market/mindex.asp>
- [2] Neil Smyth, "Android Studio Development Essentials," jpub, pp.368-369, 2015.
- [3] JaeHyun Lee, "Absolute course! An effective game creation technique that informs unity developers," Wiki books, pp.112-309, 2015.
- [4] Hui-Xuewu, and Yang-Kwon Jeong, "A study on special effects implementation algorithm based on Unity 3D," Journal of Korea Institute of Electronic Communication Science, Vol. 8, NO. 1, pp. 375-380, 2014.
- [5] Geonhyeok Go, and Mihyang Bang, and Jaewoo Seo, and Sunyoung Cho, "A Gesture-based Control Interface Design for Handheld Game Consoles Using Accelerometer," Proceedings of the 2007 HCI Conference, No. 2, pp. 381-386, Feb. 2007.
- [6] Inho Shin, and Donghun Cheon, and Hanhoon Park, "Implementing Leap-Motion-Based Interface for Enhancing the Realism of Shooter Games," Journal of the HCI Society of Korea, Vol. 11, No. 1, pp. 5- 10, Jan. 2016.
- [7] Junyeong Bok, and Heung-Gyoon Ryu, "Design and Performance Analysis of 60GHz Wireless Communication System for Low Power Consumption and High Link Quality," The Journal of Korean Institute of Communication and Information Sciences, Vol. 38A, No. 2, pp. 209-216, Feb. 2013.