

Development of 3D Defense Space Game using Oculus

Won-Gyu Iim*, Byeong Cheol Lee*, Soo Kyun Kim*, Syoungog An*

Abstract

Oculus Rift is the most universal VR (virtual Reality) headset for gamers and FPS (First Person Shooting) is the most suitable game genre to play with VR. Using VR can increase the player's sense of reality and make them feel as though they are in direct contact with the enemy while battling. The suggested VR game is a first person game where the player must defend a specific target against the surging enemy all within the time limit. Many objects will need to be used in this method. Object pooling will be used in order to manage all the numerous objects. When an object is repeatedly created and deleted it typically overwhelms the memory. To resolve this issue the game initially summons the object at the beginning of the scene and afterwards only uses the object when needed, lessening the burden on the memory. A ranking system is implemented to keep the game records in order to stimulate a competitive spirit between the players, and the game has received positive response during test play among college students in their 20s.

▶ Keyword: NGUI, Unity3D, Haptic, Oculus Rift, Virtual Reality

1. Introduction

현재 많은 분야에서 VR 장비를 이용하여 게임을 개발하고 있으며, 특히 오컬러스 리프트[1]를 활용하여 개발한 VR FPS 게임으로는 애리조나 선샤인[2], 둠 VR[3] 등이 인기를 얻고 있다. 애리조나 선샤인은 좀비를 소재로 한 게임으로 VR 컨트롤러를 최대한 활용해 무기 핸들링이 실감난다는 장점이 있으나, 타격감은 단점으로 뽑힌다. 둠 VFR은 둠 시리즈의 VR 작품으로 이 게임 역시 VR 게임의 장점인 현실성을 최고의 장점으로 꼽을 수 있지만, VR 게임의 고질적인 문제인 3D 멀미의 유발이 심한 것이 제일 큰 단점으로 손꼽힌다. 이러한 예시를 통해 VR 게임의 장점을 극대화 시키고 단점을 최소화 시키는 것이 중요한 VR 게임 개발 방법이라고 볼 수 있다.

본 제안 게임은 1인칭 디펜스 게임에 VR을 접목하여 플레이어가 직접 경험 하는 것처럼 실감나게 몰려오는 적군의 공세를 막아내는 게임의 규칙을 가진다. 본 게임의 규칙으로는 제한 시간 동안 정해진 탄약과 체력을 가지고 플레이어 간 랭킹 갱신

을 위해 더 많은 적군을 물리치며, 정해진 목표물을 방어하는 것이다. 한정된 자원으로 자원을 효과적으로 활용해 몰려오는 적을 물리쳐야하기 때문에 플레이어에게 좀 더 전략적이고 난이도 있는 플레이를 요구할 수 있게 된다.

플랫폼을 VR로 계획하게 된 이유는 그림1과 같이 PC게임의 대다수가 1인칭 FPS 장르 게임이 차지하고 있으며, 플레이어가 1인칭 시점에서 플레이 하는 VR 콘텐츠 특성과 매우 적합하여 유니티 엔진[4]과 오컬러스 리프트를 사용하여 개발하게 되었고 HMD (Head-mounted display)가 실존하는 느낌이 되도록 VR 기반으로 개발한다. 또한 오컬러스 리프트가 보편적이며 상용화된 게이머용 VR이기 때문에 제안방법에서는 오컬러스를 이용하여 개발한다.

-
- First Author: Won-Gyu Iim, Corresponding Author: Syoungog An
 - *Won-Gyu Iim (k990219@naver.com), Dept. of Game Engineering, Pai Chai University
 - *Byeong Cheol Lee (vbder123@naver.com), Dept. of Game Engineering, Pai Chai University
 - *Soo Kyun Kim (kimsk@pcu.ac.kr), Dept. of Game Engineering, Pai Chai University
 - *Syoungog An (sungohk@pcu.ac.kr), Dept. of Game Engineering, Pai Chai University
 - Received: 2019. 07. 30, Revised: 2019. 08. 26, Accepted: 2019. 08. 26.
 - This work was supported by the research grant of Pai Chai University in 2019.

Table 1. Differences between VR games

Name	Merits	Demerits
ArizonaSunshine[2]	Realistic use of weapons	a sense of hitting
DoomVR[3]	Reality	3D motion sickness
Proposed Game	Reality and Strike Sensibility	3D motion sickness

표1은 본 연구와 연관 있는 VR 게임에 대해 장단점을 분석하여 제시한다. 이러한 VR 게임에서 중요한 점은 HMD를 통해 몰입감[5, 6]을 높여줄 수 있는 고양된 즐거움, 시간적 해리 및 호기심을 자극하는 경험이며, 이를 통해 능동적 활동을 통해 몰입감을 높일 수 있다.



Fig. 1. Concurrent Steam users[7]

II. GAME DESIGN

1. Game Design

제안 방법은 그림2의 메인화면(그림2a)과 게임화면(그림2b)로 나눌 수 있으며 메인화면에서는 우주선이 우주공간을 날아다니는 연출을 주었으며 VR 환경을 통해 주변을 둘러 볼 수 있다.



(a) Main Screen (b) Play Screen

Fig. 2. Main and Play Scene

게임시작 버튼을 누를시 경고음과 적색경고를 통해 우주선이 위험함을 알리고 잠시 뒤 게임이 시작된다. 게임이 시작할 때 로딩화면을 통해 게임 씬을 불러오도록 하였으며 우주선이 운석에 충돌하는 연출을 주도록 하였다. 로딩이 완료 되면 플레이어의 입력을 받아 게임을 시작 하게 되고 각 위치에서 적군이 목표를 향해 몰려오며 플레이어는 체력이 0이 되지 않도록

제한 시간동안 적군으로부터 우주선을 지켜야 한다.

만약 제한 시간 이전에 플레이어의 체력이 0이 되거나 우주선의 체력이 0이 되면 게임에서 패배하게 되고 메인메뉴로 돌아가며 제한시간동안 목표 달성 시 승리하게 되고 점수를 랭킹에 기록한 후 메인메뉴로 돌아가도록 한다. 랭킹은 JSON를 사용해 파일로 저장하였고 게임 종료 후 다시 실행하여도 기록을 유지 할 수 있도록 하였다.

UI(User Interface)는 각각 왼손과 오른손에 배치하여 플레이에 지장이 없게 하였으며 플레이어가 직접 손을 들고 확인할 수 있게 했다. 그림2(a)의 화면에서 이를 확인할 수 있다. 그림 2(b)의 메인 화면은 게임을 시작 했을 때의 화면이다.

2. Diagram

본 절은 게임 흐름도를 보여준다. 그림 3과 같이 게임 시작시에 메인 화면에서 시작하여 우주공간에 떠다니는 우주선을 볼 수 있으며 주변을 둘러볼 수 있다. 메인 화면에서는 게임시작 버튼을 통해 게임 시작이 가능하며 게임 시작을 하면 경고음과 적색경고화면을 연출하여 우주선에 운석이 충돌하는 연출을 시작으로 게임 로딩 후 플레이어의 준비에 따라 게임이 시작된다. 이후 게임 패배시 바로 메인화면으로 넘어가며 게임 승리 시 랭킹 화면에서 현재 랭킹을 보여준 뒤 메인 화면으로 넘어간다.

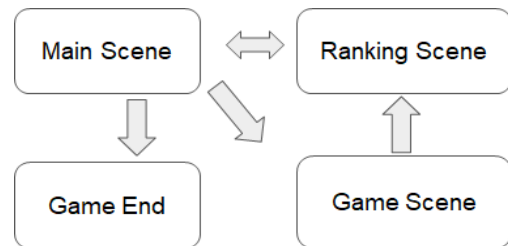


Fig. 3. Flowchart

3. Ranking System

게임을 단순히 게임을 플레이 후 목표가 없다면 유저들이 금방 지루함을 느낄 수 있기 때문에 플레이어들의 경쟁심과 성취감을 유발 할 수 있도록 랭킹 시스템을 넣어서 플레이어들의 게임 클리어 이후 점수 기록을 통해 순위를 매겨서 게임의 플레이타임을 좀 더 늘릴 수 있도록 하였다.



Fig. 4. Ranking System

기록된 랭킹은 프로그램이 종료 되더라도 랭킹 기록이 지워지지 않고 다시 불러 올 수 있도록 JSON (JavaScriptObjectNation)을 사용해 플레이어의 기록을 이름과 점수 JSON 데이터의 파일에 저장하고 필요시에 랭킹을 불러오도록 하였다. JSON은 경량의 DATA-교환 형식으로 사람이 읽고 쓰기에 용이하며, 기계가 분석하고 생성함에도 용이하다. 유니티에서 제공해주는 PlayerPrefs는 저장데이터 모양과 용량 제한이 있으므로 저장 가독성이 좋은 JSON을 사용하게 되었으며 유니티엔진[8, 9, 10, 11, 12]의 JsonUtility 클래스를 사용하여 오브젝트를 JSON 형식의 문자열로 변환하고 가져 올 수 있도록 하였다.


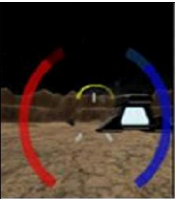

4. Design of VR Environment

제한방법은 오클러스 리프트를 사용하여 개발하였고 오클러스는 게이머 전문의 VR로 다양한 개발 지원을 하고 있다. VR을 사용함으로써 플레이어는 좀 더 현실적이고 입체적으로 게임 하는 느낌을 받을 수 있다. 가상환경에 본인이 직접 놓인 상황을 연출 하였고 표2와 같이 HMD를 통해 플레이어 본인이 직접 실감하는 느낌을 받을 수 있고 우측 컨트롤러에 총을 들고 있어 플레이어가 직접 무기를 사용하는 느낌을 받을 수 있도록 하였다. 하지만 VR 사용의 가장 큰 문제인 눈의 빠른 피로와 어지러움으로 인해 장시간 사용에 무리하고 오히려 반감이 되는 현상이 있기 때문에 어지러움[13] 관련하여 좀 더 개선이 되어야 한다.

5 User Interface

UI는 로컬 좌표가 아닌 월드 좌표를 사용하여 구현했다. 로컬 좌표는 특정 오브젝트를 중심으로 좌표가 계산되지만 월드 좌표는 원점(0,0,0)을 기준으로 좌표를 설정한다. VR의 UI의 구성에 있어서 로컬 좌표로 UI를 표현 할 수가 없으므로 월드 좌표로 UI를 지정하여 각 위치에 맞게 배치하였다. 플레이어의 상태를 나타내는 HMD는 플레이어의 시선에 따라 표1과 같이 중앙에 배치하였으며 게임의 진행상황을 알 수 있는 UI는 표1(b)처럼 왼손, 무기의 잔탄과 플레이어의 상태 수치를 오른손에 월드 좌표계는 표1(c)에 배치하였다.

Table 2. Game UI

게임 UI		
(a)	(b)	(c)
		
spacecraft strength Remaining time Score	Player shield Player strength Player Energy powder and shot	Player shield Player strength powder and shot

기존 UI의 입력은 단순히 마우스 클릭 및 터치로 기능을 수행 할 수 있었다. 하지만 VR상에서 UI의 이벤트를 사용하기 위해서는 기존의 UI의 입력을 받는 EventSystem컴포넌트에 OVRInputModule 스크립트를 더해 입력 기능을 수행하고 이벤트를 받을 캔버스에 VR 컨트롤러의 이동을 입력 받아 OVRRaycaster 스크립트로 입력받은 이벤트를 받도록 하여 버튼, 슬라이더, 토글 등 이벤트 기능을 수행 할 수 있도록 한다.

6 Controller Design

VR 헤드셋의 헤드트래킹을 지원하여 회전을 추적하여 플레이어의 시선에 따라 게임 화면이 따라가게 하였다. 그림5(a)의 좌측 컨트롤러를 통해 이동과 가속을 제어하고, 그림5(b)의 게임 진행 상황을 확인 할 수 있다. 우측 컨트롤러는 무기변경, 무기발사, 재장전 등 무기 관련 조작과 점프, 더블점프 기능을 제어하며 오른쪽 손에는 총을 부착하였고 손의 움직임에 따라 총을 움직일 수 있고 조준점은 총구로부터 발사 되도록 하였다. 플레이어의 이동은 양손 버튼에 전부 부착하여 플레이 편의성을 높였다.

7 Object Pooling

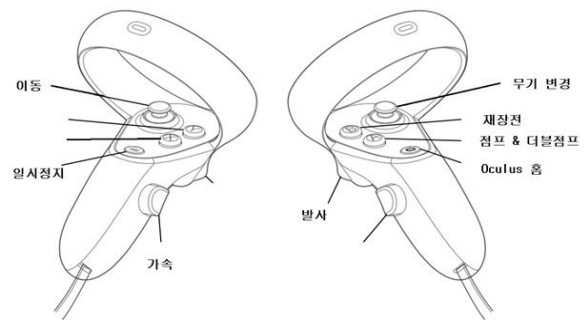


Fig. 5. Manual of Oculus Touch[1]

게임 오브젝트를 짧은 시간에 생성과 제거를 주기적으로 하는 오브젝트 수가 많을 경우 객체의 동적생성과 제거가 무수히 실행되어 메모리에 큰 부담을 주게 된다. 그렇기 때문에 그림6처럼 오브젝트 풀링(ObjectPooling)을 사용하여 게임 씬을 시작 할 때 가용할 오브젝트를 미리 생성하고 필요할 때 마다 오브젝트를 꺼내서 사용한다. 만약 더 필요시엔 오브젝트를 추가로 생성하고 비사용 시에는 비활성 한 뒤 오브젝트 풀에 넣어 오브젝트 관리에 좀 더 효율적으로 사용하였다. 이로써 메모리 단편화 및 가비지 컬렉터에 쌓이는 메모리를 방지할 수 있다. 또한 가비지 컬렉터에 메모리가 쌓이는 것을 방지함으로 프레임 드랍도 예방할 수 있다. 메모리의 생성 해체에 걸리는 시간도 줄어 처리 시간도 빨라 진다. 하지만 쓰지 않는 오브젝트를 미리 만들어두는 형식이기 때문에 메모리가 쌓일 수 있다는 단점이 있다.

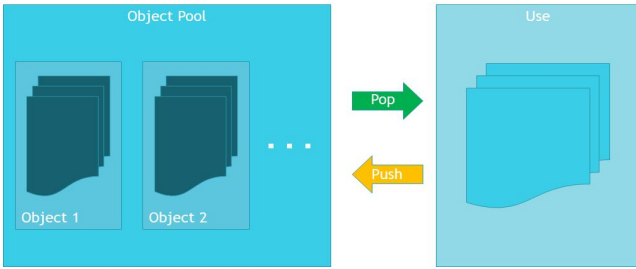


Fig. 6. Object Pooling

8. Singleton Pattern

그림 7에서와 같이, 싱글턴 패턴은 게임 씬에 한 개만 가질 수 있는 스크립트를 최초 한번만 인스턴스를 할당하고 재 사용 시 이 인스턴스를 사용하는 디자인 패턴이다. 이 패턴을 게임에도 적용시켜 게임씬에 1개만 존재하는 오브젝트에 적용시켰으며 최초에 생성한 인스턴스를 재생성 하지 않고 재사용함으로써 메모리의 낭비를 방지할 수 있다. 또한 static 변수로 전역에서 사용하기 때문에 다른 오브젝트에서도 데이터를 쉽게 참조 할 수 있다.

```
private static Object instance;
public static Object objInstance{
    get
    {
        if(instance==null)
            instance = this;
    }
    return instance;
}
```

Fig. 7. Singleton Pattern

9 Loading Scene

게임 씬을 불러올 때 한 씬에 많은 양의 데이터가 있다면 씬을 불러오는 동안 사용자는 씬이 멈춰있는 화면으로 씬 전환 간 어색한 연출이 보인다. 이를 보완하기 위해 그림8과 같이 LoadAsync를 쓰레드로 사용하여 씬을 불러온 후에 씬 로딩이 완료되면 완료 UI로 변경 하여 준 뒤 사용자의 입력을 받을 시에 다음 씬으로 교체한다. 이를 통해 사용자가 로딩이 끝난다면 게임이 시작된다는 것을 직관적으로 알 수 있고 로딩 진행 현황을 표시해 로딩이 얼마나 진행 되었는지 알 수 있도록 하였다.

```
IEnumerator LoadScene{
    yield return null;
    AsyncOperation oper = LoadSceneAsync();
    oper.allowSceneActivation = false;
    while(!oper.isDone){
        if(oper.progress==1.0f)
            oper.allowSceneActivation = true;
    }
}
```

Fig. 8. LoadAsync

10 Object Movement

오브젝트를 이동하는 방법에는 다양한 방법이 있다. 그 중 몬스터가 목표물에 이동하는 방법 중 NaveMesh 컴포넌트를 사용하여 A*알고리즘[14]으로 목표물에 최단 거리[15]로 이동하는 방법을 사용한 컴포넌트로 오브젝트와 충돌 까지 되어 매우 적합하였지만 목표물을 둘러싸 공격해야 하는 상황에서 매우 어색한 연출이 나와 MoveTowards 메소드를 사용하였더니 후자의 문제는 해결 되었으나 오브젝트와의 충돌을 검사하지 않아 벽과 오브젝트를 통과 하여 이 또한 매우 어색하였으며 각 몬스터에 플레이어와 동일한 캐릭터컨트롤러 컴포넌트를 추가하여 이동 명령을 수행 하였더니 두 가지 조건을 충족시켜 캐릭터 컨트롤러 컴포넌트를 사용하여 몬스터 이동을 구현한다.

III. Experimental Environment

실험 환경은 VR 게임 개발을 위해 고사양 PC를 채택하여 CPU는 Ryzen 2700X, RAM 16GB, VGA GIGABYTE AUROS 1080 11GB로 개발하였으며 사용한 툴은 UnityEngine 3D 2018.3.14에 VR 개발 툴인 Oculus Integration 에셋을 적용하였으며 Visual Studio 2017에 C#언어를 사용하여 개발하였다.

게임 실행 시에는 그림 9(a)와 같이 메인화면에서 컨트롤러로 플레이 버튼 실행 시 게임이 시작되며 그림 9(b) 및 (c)와 같이 게임 플레이 할 수 있다.



(a) Main Scene



(b) Game Play



(c) Game Play

Fig. 9. Game Play

IV. Conclusions

디펜스 게임을 1인칭 VR로 플레이함으로써 적군을 좀 더 실감나게 상대하며 플레이 하는 효과를 얻을 수 있고, 다양한 몬스터를 상대로 다양한 무기를 사용하여 전투를 함으로서 플레이어가 실제 같은 느낌으로 상황을 겪으며 플레이 할 수 있다. 그렇기 때문에 1인칭 장르 일수록 VR을 플레이 플레이어가 좀 더 실감 나고 현실 같은 간접 체험을 할 수 있으며 교육뿐만 아니라 다양한 게임으로도 접목 할 수 있다.

또한 게임 플레이를 완수함으로써 랭킹을 통해 사용자간 경쟁심을 유발하여 유저 대 유저 간 전투로 방향성을 노릴 수도 있다. VR 게임의 경우 2개의 디스플레이를 출력해야 하므로 고 사양의 장비를 요구 하는데 이를 좀 더 완화하기 위해서 최적화된 게임을 플레이하기 위해 오브젝트 풀링, 싱글턴 패턴, 로딩화면을 사용하여 가용되는 컴퓨터 자원을 효율적으로 사용하였다.

본 연구는 게임 제작과정에서의 중요한 부분을 논문으로 반영하여 보여주고 있으며, 향후에는 개발된 게임의 최적화에 대해 연구할 필요가 있다.

REFERENCES

- [1] Oculus Rift, <https://www.oculus.com/>
- [2] Arizona Sunshine, <https://store.steampowered.com/agecheck/app/342180/>
- [3] Doom VFR, <https://store.steampowered.com/agecheck/app/650000/>
- [4] Unity, <https://unity.com/kr>
- [5] M. Csikszentmihalyi, Play and intrinsic rewards, Journal of Humanistic Psychology Vol.15, pp.41-63, 1975.
- [6] Moonseok Kim, Youngho Seo, "A Study on the

- Relationships among Flow of Holograms 3D Contents and Learning Satisfaction for e-Learning", Journal of Korea Design Knowledge 28, pp. 13-22, December 2013.
- [7] Steam Stats, <https://store.steampowered.com/stats>
- [8] JaehYeon Lee, "Absolute Class! Unity5", Wiki Books 2015.
- [9] Patrick Felicia, "Unity 5 From Zero to Proficiency (Foundations): A step-by-step guide to creating your first game with Unity", Amazon Kindle, October 2015.
- [10] Matt Smith, "Unity 5.x Cookbook", Packt Publishing; 1 edition ,October 2015.
- [11] Alan Thorn, "Unity Animation Essentials", Packt Publishing, June 2015
- [12] Lucas Faustino, "Coroutines: Unity Focused Learning", Amazon Kindle, December 2015
- [13] Arttu Tiiri, "Effect of Visual Realism on Cybersickness in Virtual Reality", Master's Thesis, University of Oulu, 2018
- [14] A* Algorithm, https://ko.wikipedia.org/wiki/A*_%EC%95%8C%EA%B3%A0%EB%A6%AC%EC%A6%98
- [15] Dijkstra's algorithm, https://en.wikipedia.org/wiki/Dijkstra%27s_algorithm

Authors



Won-Gyu Im is attending in Game Engineering in Paichai University, Korea



Byeong Cheol Lee is attending in Game Engineering in Paichai University, Korea.



Soo Kyun Kim received Ph.D. in Computer Science & Engineering Department of Korea University, Seoul, Korea, in 2006. He joined Telecommunication R&D center at Samsung Electronics Co., Ltd., from 2006 and 2008. He is now a professor at

Department of Game Engineering at Paichai University, Korea. His research interests include multimedia, pattern recognition, image processing, mobile graphics, geometric modeling, and interactive computer graphics. He is a member of ACM, IEEE, IEEE CS, KACE, KMMS, KKITS and KIIT.



Syungog An received the Ph.D. degrees in Computer Science and Engineering from Korea University, Korea, in 1989. She is currently a Professor in the Department of Game Engineering, Paichai University.