

3D Content Design & Implementation of VR Horseback Riding Game

HyungSoo Park*, HoonKi Kim*, SiO Seo**

Abstract

Various 3D contents are being developed using Unity 3D game engine. In this paper, the 3D content of horseback riding game, the first VR game in the country, is designed and developed. The existing riding simulator is investigated and compared to the VR riding game developed. We consider various games developed using Unity 3D game engine and serve previously developed tangible games. It is expected that development of VR riding games will prepare a new chapter in VR experience-type games. We propose the content development environment and scenario of VR riding game and present the main algorithms and main modules for real-time synchronization. The developed riding game contents are deployed to the riding system and are operated for commercial use in conjunction with the riding device. Through monitoring VR riding system, problems are derived and improvement measures are proposed. We offer a variety of additional development options to make the game more realistic in the future.

▶ Keyword: Virtual Reality, Unity 3D Game Engine, Sensible Game, Horseback Riding Game

I. Introduction

현재 국내외에서 다양한 형태의 VR 게임이 개발되고 있는 실정이다[1,2]. VR 게임 종류로는 FPS(First Player Shooting) 게임, RPG(Role Playing Game), 체감형 게임 등이 있다[3,4,5]. 그 외에도 실생활에 가상현실을 접목시켜 의료용, 아동 교육용, 컴퓨터 운영체제, 운전 등 다양한 분야에 활용하고 있다[6,7,8,9,10,11]. 대기업의 경우에는 HMD를 자체 상표 이름을 걸고 개발해서 다양한 VR 콘텐츠와 상품으로 제공하고 있다. 중소기업에서는 다양한 HMD를 활용하여 3D 콘텐츠를 상용 개발하고 있다.

VR 게임 산업 분야는 블루오션 시장으로, 대기업에서는 선행 연구를 진행하면서 관망하고 있으며 중소기업에서는 Unity3D 게임 엔진을 활용하여 상용 콘텐츠를 개발하여 시장을 조금씩 넓혀 나가고 있는 추세이다[1,2]. VR은 4차 산업혁명의 핵심 분야들 중 하나로 전세계의 글로벌 기업에서 지속적인 관심을 보이면서 전시회나 박람회를 통해 새로운 기술들을 선보이고 있다[1,2].

승마 게임의 목적은 실제 승마를 탈 때의 균형 능력 운동 효과와 재미 요소를 충족시키기 위함에 있고, 이로 인해 승마 게임은 일반인

들의 균형 능력을 향상시키는 운동기기로 사용될 뿐 아니라 환자들의 재활 운동용 의료기기로도 사용되고 있다[12,13,14,15].

VR 기술을 적용하지 않은 승마 게임은 승마 시뮬레이터를 타고 스크린을 보면서 플레이하는 타입과 승마 시뮬레이터 없이 마우스 또는 조이스틱과 키보드를 이용하는 PC 타입이 있다. 이러한 기존 승마 게임은 단순히 화면으로 보면서 플레이를 하게 됨으로 플레이어의 몰입도가 현저하게 떨어지고 게임 화면과 승마기기간의 동기화가 낮아 게임 화면과 승마 기기가 종종 무관하게 동작하고 있음을 느끼게 된다. 또한, 30cm~1미터 이상 떨어져 있는 화면으로 인해 현실감과 게임의 흥미를 잃게 되어 단순히 운동 및 의료 기기로 사용되어져 왔다.

본 논문에서는 이러한 단점들을 극복하기 위하여 승마 게임에 가상현실 기기를 사용함으로써 몰입감을 극대화시켜서 재미로 운동에 대한 부담감을 제거했다. VR 기기의 HMD(Head-Mounted Display)는 플레이어의 눈으로부터 5cm 이내 있어 기존의 승마 게임들보다 훨씬 현실적인 체감을 할 수 있다. 또한, 실제 말의

• First Author: HyungSoo Park, Corresponding Author: HoonKi Kim

*HyungSoo Park (hspark@dongyang.ac.kr), Dept. of Computer Software Engineering, Dongyang Mirae University

*HoonKi Kim (kimhk@dongyang.ac.kr), Dept. of Computer Software Engineering, Dongyang Mirae University

**SiO Seo (sioseo@yesh.co.kr), YeshCompany Ltd.

• Received: 2019. 02. 20, Revised: 2019. 03. 19, Accepted: 2019. 04. 08.

• This work was supported by Dongyang Mirae Univ. Research Grant.

동작을 기기의 축의 변화와 진동 그리고 스피드까지 모두 게임과 연동한다. 게임과 승마기기 간 통신은 상호 간의 프로토콜 규약으로 처리되어 일체감을 준다. 같은 공간에서 친구 혹은 가족들 간에 경쟁레이스로 기획되어 승부욕을 자극시킨다. 게임에서 골과 랭킹 시스템을 적용함으로써 사용자들은 목적성을 가지게 된다. 현실에서 접하지 못하는 배경요소들과 환경요소들로 새로운 공간경험을 제공한다. 멀티네트워크 시스템으로 원거리 혹은 가상의 유저들과도 경쟁이 가능하다. 관전시스템을 채용함으로써 현재 경험하지 않는 사용자들은 레이스를 관람하도록 한다.



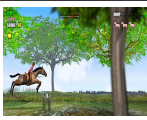
본 논문에서는 VR 승마 게임을 위한 콘텐츠를 개발함으로써 VR 체험형 게임의 새로운 장을 마련하고자 한다. 본론에서는 기존 하는 승마 시뮬레이터와 본 논문에서 개발된 VR 승마 게임을 비교하고 분석 평가한다. VR 승마 게임의 시스템 구성 요소에 대해 설명하고 VR 승마 게임의 콘텐츠를 실행 시 사용되는 DB에 대해 소개한다. 또한, 본 논문에서 개발된 VR 승마 게임의 주요 알고리즘과 실시간 동기화를 위한 주요 모듈들을 소개하고 개발 환경과 시나리오를 설명한다. 개발된 승마 게임 콘텐츠를 승마 시스템에 적용하고 상용화하여 문제점들(어지러움증, 시각적 불쾌감, 시각적 피로, 등)을 도출하고 이에 대한 개선 방안을 제안한다[16]. 결론에서는 향후 VR 승마 게임을 보다 실제적으로 체험할 수 있도록 다양한 추가 개발 방안을 제시한다.

II. Preliminaries

1. Comparison of the exist horse riding simulator

본 절에서는 본 논문에서 개발한 VR 승마 게임 콘텐츠와 기존 승마 시뮬레이터를 서로 비교 분석해 보고 각 제품의 사양과 기능을 설명한다. 표1은 개발한 VR 승마 게임 콘텐츠와 기존 승마 시뮬레이터를 비교한 것이다.

Table 1. Comparison of Horseback Riding Simulators

Item	VR Horse Racing (target)	Horse Riding Game (Screen)	Horse Riding Game (Pc Type)
Device Example			
Device Size (WDH)	450 x 867 x 820 mm	650 x 2600 x 1600 mm	-
Motor Spec.	190W, 60Hz	1.5kw 60Hz	-
Power	200W	1.5Kw	-
Weight	49kg	250kg	-
display	VR HMD	Beam Project, Large-Scale LED Monitor	LED Monitor
Kinds of Contents	Multi Racing	Multi Racing, Simulation	Character Raising, Arcade

승마기기형 시뮬레이터는 가정용 운동기구와 승마 시뮬레이션용 장비(스크린 승마용)로 크게 2가지로 구분한다. 가정용 승마기기는 2개의 서브모터를 활용하여 8자 운동(비틀기, 상/하/전/후, 슬라이드, 롤링)을 기본으로 동작하고 승마 시뮬레이션용 장비는 2개의 서브모터 및 실린더를 활용하여 보다 정교한 운동성을 가질 수 있도록 구성되어져 있다. VR 콘텐츠가 나오기 전에는 일반 디스플레이 및 3면 스크린(프로젝터)을 활용한 체험형 승마 시뮬레이션 시스템이 존재했다. 이후, Non-display Type에서 LED 모니터로, 3면 모니터 또는 3면 프로젝터로, VR HMD로 개발되었다. VR 기기의 호환성의 경우에는 장비측면이 아닌 콘텐츠 측면으로 호환 가능하고, 기존 모니터 및 프로젝터형 콘텐츠는 VR HMD에 적용하기 위해 새로 개발되어야 한다. 즉, 기존 승마 게임 시뮬레이터 콘텐츠는 VR 승마 게임과 호환될 수 없다.

승마기기의 경우, 간단한 IO보드 개발을 통해 VR 콘텐츠 및 시스템에 연동 가능하다. 단, VR 콘텐츠의 경우 시각적인 피로도가 높은 편이어서 오랜 시간 체험하기에는 스크린 형태에 비해 부적합할 수 있다. 이런 이유로 VR 콘텐츠의 경우에는 플레이시간이 짧은 레이싱 위주이고, 스크린 승마의 경우에는 레이싱뿐 만 아니라 승마 시뮬레이션 등 긴 시간 플레이가 가능한 콘텐츠를 적용할 수 있다.

VR 기반 승마 콘텐츠의 경우, 스크린 승마에 비해 몰입감이 높고, 핸드컨트롤러를 활용한 다양한 인터랙션이 가능하다. 예를 들어, 속도를 높이기 위해 채찍질을 할 수 있고, 재미를 위해 표적을 맞추는 등 다양한 요소를 추가하기 용이하다. 스크린 승마의 경우에도 이러한 인터랙션이 가능하지만, 부가적인 시스템(예: 립션, kinect 등)이 필요하다.

상대적 가격 경쟁력 측면에서 비교해 볼 때, VR 승마와 스크린 승마 모두 개인용을 구매하여 사용하기에는 매우 비싸고, 설치 및 관리가 번거로울 수 있다. 디스플레이 시스템의 차이 및 콘텐츠의 차이로 정확한 가격 비교는 어렵다. 콘텐츠 및 플랫폼을 제외하고 1기를 기준으로 시스템 비용을 고려하면 스크린 승마기기가 약 4배 이상 비싸다. VR 승마기기는 약 500만원 정도이고, 스크린 승마기기는 2000만원 이상이다.

2. Introduction of 3D Contents for VR Horse Riding

VR 승마 콘텐츠 시스템 구조는 통합 관리 서버, 운용 PC, VR기기, 클라이언트 PC, 승마 기기, IO보드로 구성된다. 통합 관리 서버의 관리자/점주 WEB이나 관리자/점주 스마트 폰의 인터랙티브 WEB을 통해 점포 관리, 콘텐츠 관리/통계, 광고영상 관리 기능을 처리한다(Fig.1.-①). 운용 PC는 Router와 인터넷을 통해 클라이언트 PC와 통신하여 클라이언트 PC를 관리하고 통합 콘텐츠를 실행하며 콘텐츠 실행 횟수(play count)를 통합 관리 서버에 전송하여 통합 관리 서버가 콘텐츠 관리/통계 기능을 수행하도록 한다(Fig.1.-②). 클라이언트 PC는 콘텐츠 실행 여부를 결정하고 운동성 제어 패킷을 승마기기에 전송한다(Fig.1.-③). 승마기기는 운동성 제어 패킷을 수신받아 해당 콘텐츠의 운동성 제어를 실행하게 된다(Fig.1.-④).

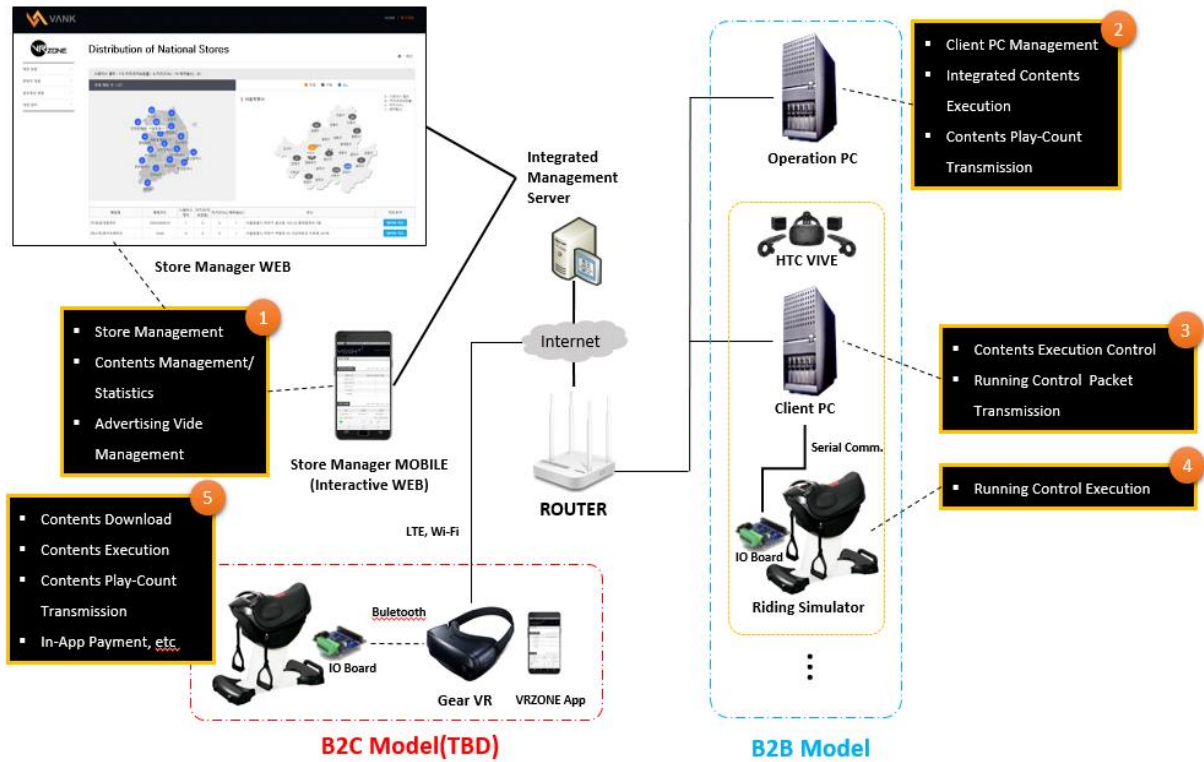


Fig. 1. System Architecture of VR Contents for Horseback Riding Game

승마 기기는 B2B 모델과 B2C 모델 두 종류가 있다. 본 논문에서는 B2B 모델을 설계하고 구현하였다. B2C 모델은 추후 연구 과제로 개발 예정이다. B2B 모델은 운용 PC, 클라이언트 PC, 승마기기 (Simulator)로 구성되어진다. 클라이언트 PC와 승마기기는 자체 제작된 IO보드로 연결되어 직렬(Serial) 통신을 하게 된다. 이 직렬 통신 패스를 통해 운동성 제어 패킷을 전송하여 승마기기의 운동성을 제어하도록 하였다. 본 논문에서는 승마기기와 IO보드 간 운동성 제어에 대해서는 논외로 하고 3D 콘텐츠 위주로 설계 및 개발을 제한한다. B2B모델의 VR기기는 HTC VIVE를 사용하도록 구현되었고 클라이언트 PC와 VR기기는 USB와 HDMI 케이블로 연결되어 통신하도록 구현되었다. B2C 모델은 VR 기기인 기어 VR로 구성되어지며, 클라이언트 PC 역할은 스마트폰이 담당하며, 승마기기의 IO보드와 VR기기 간 통신은 블루투스 통신을 사용한다. 승마게임 이용자는 자신의 스마트폰의 VRZONE App을 이용하여 통합관리 서버로부터 승마 3D 콘텐츠를 다운로드하고 실행하며 콘텐츠 실행 횟수를 통합관리 서버에 전송하고 인앱 결제 등을 처리한다.(Fig.1-⑤) 그림 1은 승마 VR 콘텐츠 시스템 구조도를 나타낸 것이다.

3. Data for VR Horseback Riding Game

VR 승마 시스템은 두 개의 데이터 서버가 존재한다. 하나는 게임 서버 동기화를 담당하는 포톤 서버이고 다른 하나는 기기 인증 및 로그를 관리하는 통합 관리 서버 내의 AWS 서버이다. 포톤 서버는 클라이언트 PC에서 승마 콘텐츠를 실행할 때 필요한 객체 정보들(room number, 승마기기 ID)을 인터넷을 통해 실시간으로 동기화한다. AWS 서버에서 승마 기기 인증에 사용

되는 데이터는 다음과 같다.

- 매장코드 : 매장 식별자 (예, VRZONE0028)
- 기기코드 : 승마 기기 식별자(예, VRZONE0028C01)
- 기기번호 : 승마 기기 번호(예, 1)
- 등록일 : 승마 기기 등록일(예, 1530444475)
- 해시키 : Operation PC 인증키
(예, d3f75abc18022bf9bc9d17d1497dca)

운용 PC는 승마 게임이 종료된 후 실시간으로 승마 게임 실행 로그를 AWS 서버로 전송하여 각 매장의 실행한 횟수를 통계로 관리한다. 시스템 구조 내에서 데이터 서버는 그림 2와 같은 게임 실행 로그는 일련번호, 매장코드, 기기코드, 콘텐츠코드, 시작일, 시작시간, 종료일, 종료시간, 플레이시간, 사용인원으로 구성된다. 실행 로그 예를 들면, 다음과 같다.

15223467,VRZONE0015,VRZONE0015C02,T2000,20190217,95343,20190217,95556,133,4

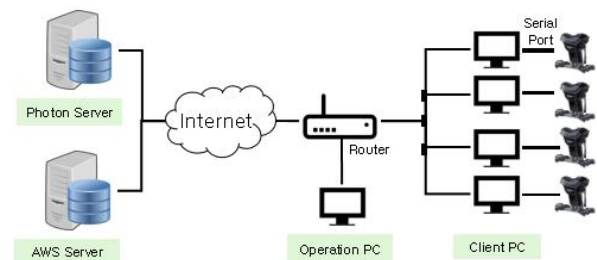


Fig. 2. Data Server in the System Architecture

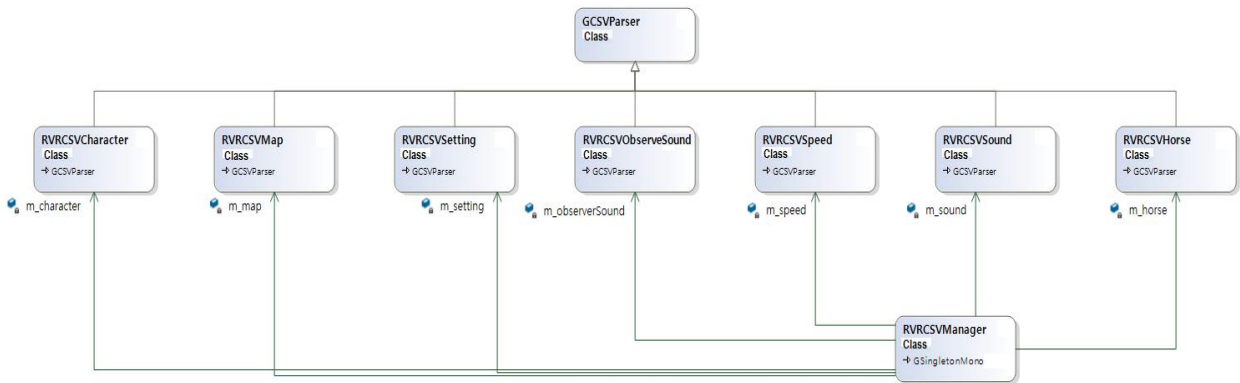


Fig. 3. Data Handling Class Diagram

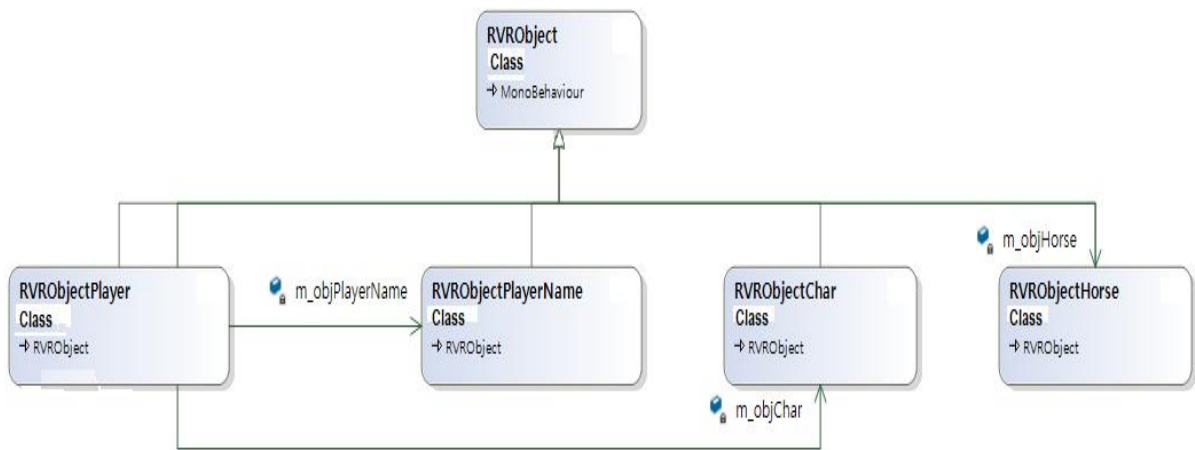


Fig. 4. Object Handling Class Diagram

III. The Proposed VR Contents System

II장에서 제시된 시스템의 구조에 따라 III장에서는 승마 게임 3D 콘텐츠를 설계하고 개발하며, 주요 모듈들과 알고리즘을 설명한다. 본 논문에서 제안된 VR 콘텐츠는 Unity 3D 게임 개발 엔진으로 개발되었다.

1. Design & Implementation

승마 게임 3D 콘텐츠의 환경 및 시나리오를 디자인하고 시나리오에 따라 승마 코스 콘텐츠를 개발하였다.

1.1 Class Diagram of Proposed Riding Game

승마 게임은 3D Unity 게임엔진으로 개발되어 사용된 언어는 C#으로 구현되었으며 에디터는 Visual Studio를 사용하였다. 승마 게임의 모듈은 클래스로 구성되었으며 크게 데이터를 처리하는 Data Handling Class 그룹과 Object Handling Class 그룹으로 분류될 수 있다. 그림3은 승마게임에서 사용되는 데이터를 다루는 클래스들의 상관 관계를 나타내는 클래스 다이어그램이고 그림 4는 오브젝트들의 처리 로직을 다루는 클래스들의 상관 관계를 나타내는 클래스 다이어그램이다.

RVRECSVManager 클래스는 CSV (Comma-Separated Values)파일 형태의 데이터들(character, Map, Sound, Speed, Horse, 등)을 읽어 이 데이터를 처리하는 각 클래스들에게 데이터를 전달해 준다. GCSVParse 클래스는 GCSVManager에서 읽어들이는 데이터를 불러오는 역할을 담당한다. RVRCsv Character 클래스는 PlayerID, 프리뷰, 카메라 위치, 출발지점에서의 애니메이션, 1등일 때의 애니메이션 등의 데이터를, RVRCsvMap 클래스는 맵의 모든 데이터를, RVRCsvSetting 클래스는 시간과 관련된 모든 데이터(초기 대기 시간, 출발 준비 시간, 카운트다운 시간, 결과창 UI보여 주는 시간 등)를, RVRCsvSound와 RVRCsvObserveSound 클래스는 게임의 사운드와 음성과 관련된 데이터를, RVRCsvSpeed 클래스는 PlayerID와 최대 속도 데이터를, RVRCsvHorse 클래스는 말과 관련된 데이터를 읽고 저장한다. 말과 관련된 데이터는 이동시 이펙트, 프리뷰, 채찍했을 때의 속도 값, 장애물 성공 및 실패 시 속도 증감량 등이다.

RVObject 클래스는 Unity에서 제공하는 MonoBehaviour 클래스로부터 상속을 받아 각각의 오브젝트 클래스들을 제어 관리한다. 각각의 오브젝트 클래스들 (RVObjectPlayer, PlayerName, Char, Horse, 등)은 RVObject로부터 상속을 받는다. RVObjectPlayer 클래스는 RVObjectPlayerName클래스에게

PlayerName을, RVObjectChar 클래스에게 Character를, RVObjectHorse 클래스에게 Horse를 전달해 준다. RVObjectPlayer 클래스는 플레이어의 속도변화 및 사운드, 골인 지점에서의 행동을 관리하고 캐릭터와 말, 관전카메라의 데이터를 불러온다. RVObjectPlayerName클래스는 플레이어 이름의 위치와 회전값을 조정하고 RVObjectChar 클래스는 플레이어일 경우와 관람자일 경우에 따라 각각의 애니메이션 처리 및 시선처리를 담당한다. RVObjectHorse 클래스는 말의 애니메이션 속도변화와 속도변화에 따른 애니메이션 처리를 담당한다.

1.2 Speed and Moving Control between Horseback Riding Simulator and Local PC

승마 기기의 움직임을 로컬 PC에서 제공된 속도 값으로 제어하는 모듈이다. 그림 5는 승마기기와 로컬 PC간의 스피드 및 무빙제어를 위한 API 소스코드이다.

```
void OnThreadMoter()
{
    float fSpeed = 0f; // the speed of riding simulator
    float fTime = 0f; // the time of motor manager

    if( motorManager is null )
        return;

    while( Thread is not exist )
    {
        try
        { // When the speed of riding simulator and
          // the speed of horse in the game is different,

          if(the speed of riding simulator !=
            the speed of horse in the game)
          {
              if( motorManager is not null )
              {
                  if(the speed of horse in the game is 0)

                      Stop the motor of riding simulator;
                  else
                      Set the speed of riding simulator

              }

              set the speed of riding simulator to
              the speed of horse in the game;
              fTime initialization;
          }
          // When the speed of riding simulator and
          // the speed of horse in the game is the same,
          else
          {
              increase time by 0.05;
              if(DEFAULT_TIME/2f <= fTime)
              {
                  if( motorManager is not null )
                      operate the motor in the speed;
                  fTime initialization;
              }
          }
        }
        catch(System.Exception e) // Execption Handling
        {
            Debug.Log ("yeshRidingMotorManager error
            message output");
        }
    }
}
```

Fig. 5. API Source Code for Speed and Moving Control

현재 승마 기기의 속도가 VR 승마 게임 말의 속도와 다를 때 승마 기기는 게임 말의 속도에 따라 동작하게 된다. 즉, 속도가 0이면 승마기기를 정지시키고 0이 아닐 경우에는 말의 속도와 같은 비율로 동작하게 한다. 승마 기기의 속도가 게임 말의 속도와 같을 때 승마기기의 모터가 구동되는데 모터 구동 시, DEFAULT_TIME만큼 지난 후 모터를 움직이면 기계가 멈췄다 움직이게 되므로 DEFAULT_TIME/2 만큼의 시간이 지났을 때 동작하도록 한다. 이처럼 구동하게 한 이유는 승마기기에서 DEFAULT_TIME 내에 속도 정보를 받지 못하면 승마기기가 정지 되도록 제작되었기 때문에 안정적으로 승마기기를 동작하기 위해 DEFAULT_TIME/2 내에 속도 정보를 전송하도록 구현하였다.

1.3 Speed Data Synchronization

승마 게임 이용자가 VR기기를 통해 말의 속도를 증가시키려고 할 때 VR 기기로부터 추가된 속도 데이터를 승마기기의 IO 보드에 전송하는 모듈이다. 종착지에 도착했을 때 관전모드일 경우 플레이어들의 기록을 1위부터 순서대로 보여주게 되며, 관전모드가 아닐 경우에는 내 순위와 기록을 제일 위에, 그리고 나머지 플레이어들은 순위대로 보여 준다. 그림 6은 VR기기와 승마기기간의 추가 속도데이터를 전송하는 소스코드이다.

```
#region RPC
public void SendAddSpeed(int gear)
{
    // speed data transmission.
    if( Player is null )
        return;
    if( !(Request Add Speed) )
        return;

    m_IsReqAddSpeed = true;
    Send AddSpeed to gear;
}
void OnRPCAddSpeed(int gear)
{
    // increase speed data.
    if( Player is null )
        return;
    Add Speed to gear;
    m_IsReqAddSpeed = false;
}

public void SendArrive(int endTimestamp, int
posIndex)
{
    // arrival transmission.
    if( Player is null )
        return;

    Send arrival message to PhotonTargets;
}

public void OnRPCArrive(int endTimestamp, int
posIndex)
{
    // arrival.
    if( Player is null )
        return;

    if(Observer mode is true)
        display the record of players by rank;
    else
        first of all, display my record, and then
        display the record of other players by rank;
}
}
```

Fig. 6. Source Code for Speed Change

1.4 Algorithm of VR Contents

사용자가 VR기기를 통해 운영 툴을 기동하면 로컬 PC IP 주소를 확인하고 정상이면 게임에 접속하여 승마기기 움직임을 제어할 수 있도록 준비한다. 게임이 시작되면 HMD 화면을 통해 디스플레이 된 경로에 따라 목표점까지 말을 경주하고 순위 정보를 DB에 저장하고 승마 게임을 종료한다. 승마 게임 3D 콘텐츠의 시작부터 종료 시까지 알고리즘을 그림 7에 나타낸 것이다.

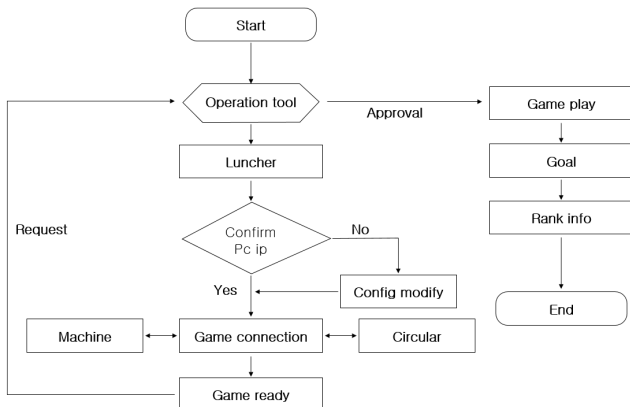


Fig. 7. Flow Chart for VR Contents

1.5 Environment and Scenario

기본적으로 승마경기의 기본요소인 트랙을 구성하고, 트랙은 오르막, 내리막, 코너링, 장애물 등의 구성으로 어트랙션을 최대한 느끼도록 구성하였다. 다양한 환경변화로 계절의 특성 및 환경적으로 최적의 효과를 볼 수 있는 폭포, 산맥 등을 구성하였다. 사용자에게 몰입감을 주기 위한 자연환경 소리 및 말 음성 및 효과음을 3d사운드로 구성하였다. 해설자 목소리를 적용해서 현장감을 살려 주고 레이스에 집중할 수 있도록 유도하였다. 눈, 비, 바람, 물튀김, 먼지 등의 이펙트 요소로 게임의 몰입도를 한층 극대화하였다. 서클레이터와 같은 하드웨어를 추가적으로 구성하여 달리는 속도감을 최대한 느낄 수 있도록 구성하였다. 시나리오는 다음과 같다.

“말을 주요 교통수단으로 쓰는 한 시골마을에 현직 승마선수들이 도전장을 내밀었다. 승마선수 생활을 하던 중 부상으로 고향으로 내려와 있던 주인공 케이는 이들의 도전에 맞서 우승을 거머쥘 수 있을까? 거대한 폭포와 산맥으로 이루어진 시골마을에서 이제 승마레이스가 펼쳐지려고 한다.”

2. Deployment and Result

본 논문에서 개발한 VR 승마 콘텐츠를 화성 동탄 일렉트로 VR 사이트에 적용하고 2018년 4월에 상용 운영하였다. 상용 시스템을 모니터링하면서 발생한 주요 문제점들을 제시하고 이에 대해 해결책을 제안하였다.

2.1 Title, Standby Screen

VR 콘텐츠를 승마기기 시뮬레이터에 적용하고 실행되는 화면을 캡처하였다. 그림 8은 타이틀 이미지 화면을 나타낸 것이고 그림 9는 대기 화면을 나타낸 것이다.



Fig. 8. Title Image Screen



Fig. 9. Standby Screen

2.2 Full Map, Detailed Map by Section

승마 게임은 4명이 동시에 멀티 레이싱이 가능하게 구성되어 있고 1,2번 플레이어는 Route A를 사용하고 3,4번 플레이어는 Route B를 사용한다. 그림 10은 레이싱 루트의 전체 맵을 표현한 것이다.

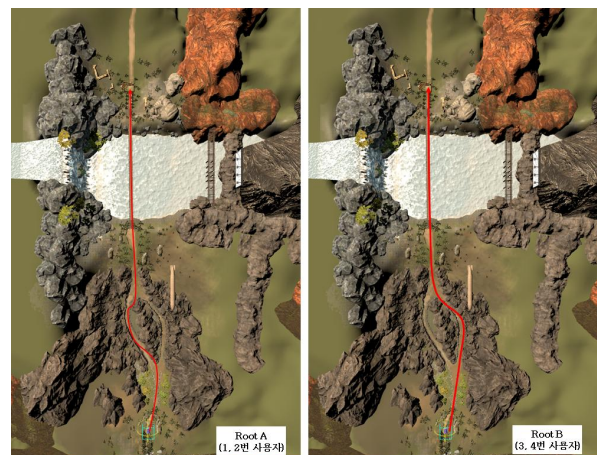


Fig. 10. Full Map Screen of Racing Route by player

레이싱 루트별 전체 맵은 구간별로 화면을 다르게 구성하여 게임 플레이어가 보다 실감있는 레이싱을 할 수 있도록 구현하였다. 시작점과 도착점 사이에 4개의 상세 구간을 구성하였고 주요 경쟁 구간과 도착점에서는 In-Game 광고판을 볼 수 있도록 하였다. 4개의 상세 구간은 시작점에서부터 내리막 지형지물 구간, 갈림길 구간, 오르막 지형지물 구간, 주요 경쟁 구간으로 구성된다. 그림 11은 각 구간별 상세 화면을 나타낸 것이다.

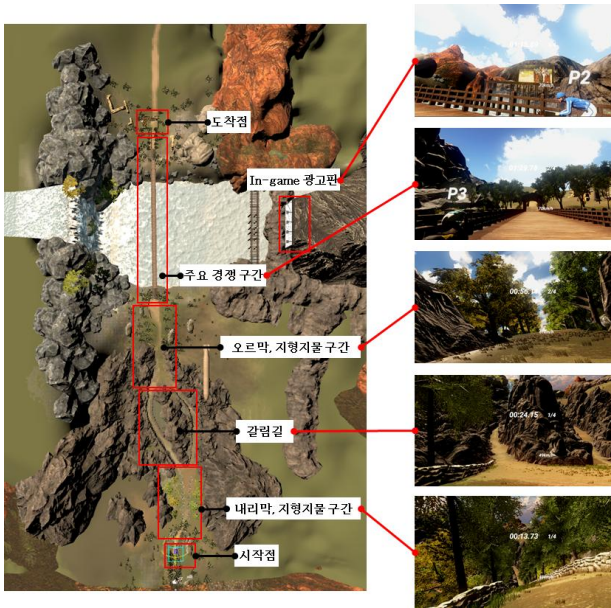


Fig. 11. Detailed Screen by Section

2.3 User Interaction and Play Screen

승마 게임 3D 콘텐츠는 두 가지의 UI를 제공할 수 있다. 레이싱 도중에 현재 자신의 기록 및 순위 정보와 속도 정보를 확인할 수 있고 게임 종료 후 플레이어별로 기록 및 등수를 확인할 수 있다.

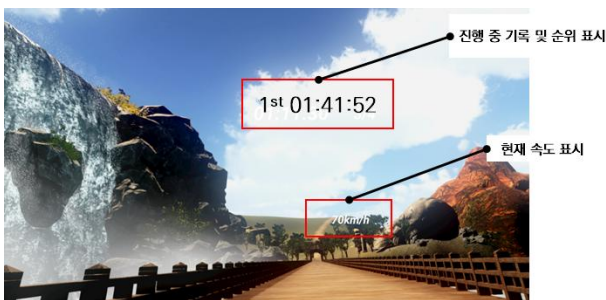


Fig. 12. UI Screen on Racing

그림 12는 경기 도중에 각 플레이어에게 HMD에 표시되는 UI 정보를 표시한 상세 화면이고 그림 13은 경기 종료 후 UI 정보를 표시한 상세 화면이다.

승마 게임 3D 콘텐츠는 4명이 동시에 레이싱을 할 수 있으며 관람자를 위하여 진행되는 경기 상황을 LED 모니터를 통해 확인할 수 있다. 그림 14는 국내 최초로 화성 동탄에 상용 설치

되어 운용되고 있는 VR 승마 구축 사례 화면이고 그림 15는 플레이어들이 시스템을 이용해 레이싱을 하고 있는 화면이다.



Fig. 13. Detailed Screen by Section



Fig. 14. Deployment Case



Fig. 15. Ride Racing Picture

2.4 Practical Contribution

기존 스크린형, PC게임형 승마와 비교하여 가장 큰 차이점은 VR HMD를 통해 콘텐츠를 보는 것이다. 외부의 요소가 99% 차단되어 시각적인 몰입감은 현존하는 디스플레이 장치 중 최고라고 할 수 있다. 실제 승마와 유사하게 채찍질을 하여 말의 속도를 높이는 개념으로, 사용자에게 실제와 유사한 경험을 제공한다. 승마 콘텐츠에서는 말의 속도에 비례하여 승마기기의 운동속도를 조절하고 있으며 별도의 진동 요소는 사용하지 않고 있다. 가상현실 레이싱형 콘텐츠에서 튼을 할 경우 어지러움증을 유발시킬 수 있으므로, 본 논문에서는 직선 위주로 맵을 구성하였고, 곡선 부분도 최대한 완만한 곡선으로 구성하였다. 속도에 따른 승마기기의 운동속도를 조절하여 사용자에게

게 몰입감을 제공하였다. 승마 게임을 국내 최초 3D VR 체감형 게임으로 상용화함으로써, 앞으로 이와 유사한 다양한 체감형 게임 개발에 새로운 장을 마련하였다.

2.5 Problems and Improvement

승마 게임을 실행할 때 두 가지 문제점이 발생하였다. 유저간 속도 편차가 심한 경우, 중도 포기하는 상황이 발생하였다. 이는 사용자의 인터랙션(컨트롤러 흔들기)을 통해 속도가 변하기 때문에 속도차이로 인해 빈번히 발생하는 문제로 기본 속도를 지정(인터랙션 없어도 레이스 진행 가능)하여, 목적지까지 상대를 볼 수 있도록 개선하였다.

콘텐츠 상 방향전환으로 인한 어지러움증이 유발되었다. 초기 맵 구성 시 좌/우측 40도 이상 방향전환이 있었는데, 승마기기의 운동성과 일치하지 않아 사용자로 하여금 어지러움증을 유발하였다. 대부분 직선 맵으로 변경하고 방향 전환 시 점진적으로 진행하도록 하여 문제를 해결하였다.

IV. Conclusions

4차 산업혁명을 선도하는 여러 가지 기술들 중 하나가 VR이다. 현재 VR 기술을 이용하여 FPS(First Player Shooting) 게임, RPG(Role Playing Game), 체감형 게임 등 다양한 게임을 개발하고 있는 추세이다. 본 논문에서는 체감형 게임의 일종인 승마게임을 위한 3D 콘텐츠를 B2B 모델로 개발하여 국내 최초로 상용화하였다. 개발한 VR 승마 게임 콘텐츠와 기존 승마 시뮬레이터를 서로 비교 분석해 보고 상용화 후 문제점들을 모니터링하고 문제점들을 개선하여 사용자들이 몰입하여 실감 있게 게임을 즐길 수 있도록 하였다.

향후, B2C 모델로 연구 개발을 추진할 예정이며 상용화된 3D 승마 게임 콘텐츠에도 몇 가지 추가 개선이 필요하다. 현재 로컬 네트워크로만 플레이 가능한 구조로 같은 공간(배장)의 사용자 간 경쟁만 가능하다. 이를 개선하여 인터넷기반 멀티플레이 기능의 추가가 필요할 것이다. 4D 이펙트로 바람이 있는데, ON/OFF 만 가능한 상태이다. 속도기반 바람세기 조절기능이 추가되어야 한다. 승마기기 운동성을 현재는 Pitch, Yaw 운동만 활용하고 있는데 점프 및 오르막/내리막 등을 느낄 수 있도록 heave 운동성의 추가가 필요하며, 이를 위해 승마 시뮬레이터에 2축 이상이 필요할 것이다. 다양한 맵 구성 및 난이도 구성을 통해 사용자로 하여금 선택하여 플레이 할 수 있게 변경이 필요하다. 현재는 탑승 위치에 따라 캐릭터 및 말이 정해지는데, 이 부분을 선택할 수 있게 하여 사용자의 자유도를 증대시키는 것이 요구되어 진다.

VR 승마 게임이 국내 최초로 개발되었기 때문에 위에서 제시했듯이 시스템의 많은 개선과 추가 개발 연구가 필요할 것으로 전망된다.

REFERENCES

- [1] Lee, Hyuk-Soo, "VR Industrial Ecology System and Korea VR Direction of Industrial Development," VR Summit, 2016
- [2] Young-Kee Cho and YoSep Song, "Virtual Reality(VR) Game Development Activation Strategy," Korea Creative Content Agency, 2016,
- [3] JaeHwan Bae, "Design and Development for Unity3D Game Engine using the Shooting Game," Korean Society For Computer Game, 29(1), 2016, pp. 93-100
- [4] Jong-Wan Kim, Joo-Sung Kim, and Tae-Eun Kim, "A study on FPS game using virtual reality," Proceedings of KIIT Conference, 2017, pp. 176-180.
- [5] Changmin Cha and Changhoon Park, "Implementation of Tangible Game Using Oculus Rift," The HCI Society of Korea, 2014.2, 583-586.
- [6] Ghee Young Noh, "A Structural Equation Modelling of the Relationship between User Experience, Self-efficacy and Game Performance in Healthcare Serious Game," Journal of Korea Game Society, 12(2), pp. 15-29, 4. 2012.
- [7] Young-A Kim and Young Jeon, "A Systematic Review of Domestic Research on Virtual Reality Programs using Wii Consoles for the Elderly," Journal of the Korean Society for the Elderly, 8(2), pp.39-46 November 2016.
- [8] Ta-Ko Huang, Chi-Hsun Yang, Yu-Hsin Hsieh, Jen-Chyan Wangf, and Chun-Cheng Hung, "Augmented reality (AR) and virtual reality (VR) applied in dentistry," Kaohsiung Journal of Medical Sciences (2018) 34, 243-248.
- [9] C. J. Lim, Won Dae Han, and Jeong Yun Guen, "Educational Game Making-Tool Development using Unity3D Engine Birth of Game," Journal of Korea Game Society 2014 Feb, 14(1), 29-38.
- [10] Eunsol Kima, Jiyeon Kima, Eunjin Yooa, and Taejung Park, "Study on Virtual Reality (VR) Operating System Prototype," Journal of broadcast engineering, Vol. 22, No. 1, January 2017.
- [11] Jae-Han Bae, Jae-Jin Kim, and Ghee-Young Noh, "An Experimental of the Effects of User Experience and Driving Attitude on Driving Simulation Game in Virtual Environment," Journal of Korea Game Society 2015 Jun; 15(3): 7-18.
- [12] Kwon Haeyeon and Kim Byeongjo, "The Effects of Horseback Riding Simulation Training on the Thickness of Abdominal Muscles and Functional Balance in Children with Down Syndrome," Journal of The Korean Society of Integrative Medicine, 2018, 6(4), 127~137.
- [13] Soon-Hyun Kwon, Ki-Un Kim, Jung-Bin Won, Dong-Kyun Park, and Dae-He Lee, "The effect of improve balance ability in Horseback riding simulator and virtual reality exercise for normal person," Korea

Academy Industrial Cooperation Society, 2014.05, 556-558.

- [14] Woon-Soo Cho and Sung-Hyoun Cho, "Effects of Mechanical Horseback Riding Exercise on Static Balance of Patient with Chronic Stroke," Korea Academy Industrial Cooperation Society, 16(3), 2015.03, 1981-1988.
- [15] Yeong-Hyeon Byeo, Myung-Won Lee, Jae-Neung Lee, and Keun-Chang Kwa, "A Trends Analysis of Domestic Research in Developing Horse Riding Simulator," Institute of Control, Robotics and Systems, 2013.5, 414-416.
- [16] Jungmin Han, Seon Hee Bae, and Hyeon-Jeong Suk, "Visual Discomfort and Visual Fatigue: Comparing Head-Mounted Display and Smartphones," J Ergon Soc Korea 2017, 36(4), 293-303.

Authors



HyungSoo Park received the B.S., M.S. and Ph.D. degrees in Computer Science and Engineering from Korea University, Korea, in 1992, 1995 and 2008, respectively. Dr. Park joined a researcher of R&D Center at LGE, Ltd. and ntelia, Ltd. Anyang, Korea, in

1995 and 2004. He is currently a Professor in the Department of Computer Software Engineering, Dongyang Mirae University. He is interested in mobile network, internet and mobile computing, and Information Security.



HoonKi Kim received the B.S., M.S. and Ph.D. degrees in Electronic Engineering from Hanyang University, Korea, in 1988, 1990 and 2002, respectively. After receiving his M.S. degree, he was with Core Network Research Labs., LG Electronics Inc., Korea,

from 1990 to 2001. He has been engaged in research of switching, CDMA cellular, PCS, IMT-2000, NGN system. Dr. Kim joined the faculty of the Department of Computer Science at Dongyang Mirae University, Seoul, Korea, in 2001. He is currently a Professor in the Department of Computer Science, Dongyang Mirae University. He is interested in embedded system, communication software and wireless communication networks.



SiO Seo received the B.S., M.S. and Ph. D. degrees in Computer and Information Science from Korea University in 2007, 2009 and 2015 respectively. Dr. Seo is currently a chief technical officer of VR/AR center at YESH company, Korea.

He is interested in virtual reality, argument reality, motion simulator system and digital contents.