



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

2017학년도

석사학위논문

가상현실 기반의 낚시 체감형  
시뮬레이션에 관한 연구

A Study on A Virtual Reality-based  
Fishing Motion-based Simulation

남서울대학교 대학원

가상증강현실학과 가상증강현실학전공

김 다 애

2017년 12월

가상현실 기반의 낚시 체감형  
시뮬레이션에 관한 연구

A Study on A Virtual Reality-based  
Fishing Motion-based Simulation

지도교수 강 민 식

이 논문을 석사학위논문으로 제출함

2017년 12월

남서울대학교 대학원

가상증강현실학과 가상증강현실학전공

김 다 애

# 김다애의 석사학위논문으로 인준함

심사위원장 송 은 지 ㉠

심 사 위 원 강 민 식 ㉠

심 사 위 원 이 석 희 ㉠

남서울대학교 일반대학원

2017년 12월

# 목 차

제 1 장. 서 론 .....	1
1.1 연구배경 및 필요성 .....	1
1.2 연구방법 및 구성 .....	2
제 2 장. 이론적 고찰 .....	4
2.1. 가상현실 .....	4
2.1.1 가상현실의 정의 .....	4
2.1.2 가상현실 기기 .....	4
(1) 오쿨러스 리프트 .....	5
(2) HTC 바이브 .....	6
(3) PS VR .....	7
2.2. 체감형 콘텐츠 .....	9
2.2.1 체감형 콘텐츠의 정의 .....	9
2.2.2 체감형 콘텐츠 사례 .....	10
2.2.3 체감형 디바이스 .....	14
2.2.4 기기 종류 .....	14
(1) Wii 리모컨 .....	14
(2) 키넥트 .....	15
(3) 플레이스테이션 무브 (Playstation Move) .....	16
2.2.5 기기 원리 .....	18
(1) 가속도계 .....	18
(2) 관성 센서 .....	19

제 3 장. 체감형 낚시 콘텐츠 기획 및 설계 .....	20
3.1. 설계도구 .....	20
3.2. 설계과정 .....	21
3.2.1 개발 설계 .....	21
3.2.2 구현 .....	24
3.2.3 구현 결과 .....	38
 제 4 장. 결론 및 제언 .....	 40
 참고문헌 .....	 42
국문초록 .....	43
ABSTRACT .....	45

## 표 목 차

[표 1] 플레이스테이션 무브 컨트롤러 비교 .....	17
[표 2] 던지는 시간과 상수 값에 따른 낚싯바늘의 최종 좌표 .....	27
[표 3] 던지는 거리와 상수 값에 따른 낚싯바늘의 최종 좌표 .....	28

## 그림 목 차

[그림 1] 오쿨러스 리프트 .....	5
[그림 2] 바이브 .....	6
[그림 3] PSVR(Playstation VR) .....	7
[그림 4] Wii Sport의 다양한 컨트롤러 .....	10
[그림 5] Wii Sport의 테니스와 컨트롤러 .....	11
[그림 6] Wii Sport의 골프와 컨트롤러 .....	11
[그림 7] Kinect Adventures .....	12
[그림 8] UFC Personal Trainer .....	13
[그림 9] Wii 리모컨 .....	14
[그림 10] 키넥트 .....	15
[그림 11] 플레이스테이션 무브 모션 컨트롤러 .....	17
[그림 12] 가속도계의 원리 .....	19
[그림 13] 낚싯바늘, 낚싯대 .....	21
[그림 14] 플로우 차트 1 .....	22
[그림 15] 플로우 차트 2 .....	23
[그림 16] 바이브 설정 .....	24
[그림 17] Configurable Joint 설정 .....	25
[그림 18] 던지는 방향 .....	29
[그림 19] 포물선 운동 .....	30
[그림 20] 낚싯대를 던졌을 때 찌가 날아가는 경로 .....	30
[그림 21] 릴 .....	32
[그림 22] Hinge Joint 설정 .....	33
[그림 23] 상수 값이 200일 때 .....	34
[그림 24] 상수 값이 400일 때 .....	35
[그림 25] 상수 값이 600일 때 .....	35

[그림 26] 상수 값이 800일 때 .....	36
[그림 27] 상수 값이 1000일 때 .....	36
[그림 28] 낚싯바늘이 매달려 있는 화면 .....	38
[그림 29] 낚싯바늘이 던져진 화면 .....	39
[그림 30] 물고기가 잡히고 릴을 돌리는 화면 .....	39

# 제 1 장. 서 론

## 1.1 연구배경 및 필요성

오쿨러스 리프트(Oculus Rift)를 시작으로 HTC사의 바이브(Vive), 소니의 PS(Playstaion) VR, 삼성의 기어 VR 등의 다양한 가상현실 기기가 등장함과 동시에 VR 콘텐츠가 주목받고 있다. 기존의 VR 콘텐츠는 키보드와 마우스를 이용하는 방식으로 사용자의 몰입감을 극대화시키는 데에 한계가 있었다. 이는 최초의 체감형 컨트롤러(Controller)로 인정받는 닌텐도(Nintendo)의 Wii 컨트롤러와 마이크로소프트(Microsoft)의 키넥트(Kinect)가 출시된 이후 달라졌다. HTC의 바이브와 Facebook의 오쿨러스 리프트 터치 등 상호작용을 위한 다양한 체감형 컨트롤러가 등장하기 시작했다.

체감형 컨트롤러를 이용한 콘텐츠는 일반적인 콘텐츠와는 다르게 직접적인 신체감각의 상호작용을 통해 사용자가 얻을 수 있는 만족감을 극대화 시켜준다. 체감형 콘텐츠로 ‘닌텐도 Wii 스포츠’ 나 ‘스윙골프 팡야’ 등의 스포츠 게임과 대표적인 레저 스포츠인 골프의 사례를 보면 가상현실 기술과 골프가 접목된 스크린골프장이 있다. 특히 스크린골프장은 저렴한 가격, 시간 및 공간에 제약 받지 않는 장점 등으로 인해 골퍼들에게 많은 인기를 얻고 있다.<sup>1)</sup> 레저 스포츠인 낚시도 최근 낚시 카페가 생길만큼 인기가 많지만 특정한 장소와 고가의 장비가 필요해서 접근하기가 어려운 특성을 가지고 있다. 본 논문에서는 이를 바탕으로 낚시 체감형 시뮬레이션을 제작했다.

---

1) 배일한. “가상현실 스포츠의 산업현황과 성장조건에 관한 연구 : 한국의 스크린골프 산업을 중심으로,” (석사학위, 고려대학교 언론대학원, 2010), 3.

본 연구에서는 HTC사 바이브의 가상현실 HMD(Head Mounted Display)와 모션 컨트롤러로 게임 엔진인 유니티(Unity)를 이용해서 낚시 시뮬레이션을 효과적으로 구현하는 방법에 대해 연구했다.

## 1.2 연구방법 및 구성

본 연구는 가상현실 HMD, 체감형 컨트롤러와 게임 엔진을 이용해서 낚시 시뮬레이션을 설계하고 구현하였다.

본 연구를 통해 체감형 컨트롤러를 사용한 효과적인 낚시 시뮬레이션 기법에 대해 제안하고자 한다.

자세한 연구 방법은 다음과 같다.

첫째, 논문과 문헌을 활용한 선행연구를 통해 가상현실과 체감형 컨트롤러에 대해 연구한다.

둘째, 낚시 시뮬레이션 설계를 위해 분석하고 플로우 차트를 작성한다.

셋째, 낚시 시뮬레이션 기법에 대해 연구하고 유니티 엔진을 사용해서 구현한다.

넷째, 효과적인 낚시 시뮬레이션 기법을 제안한다.

본 논문은 총 4장으로 다음과 같이 구성되었다.

제 2장에서는 이론적 고찰로 가상현실의 정의와 가상현실 기기에 대해 알아본다. 체감형 기기의 종류와 특징에 대해 알아본다.

제 3장에서는 유니티와 가상현실 기기인 바이브를 이용해서 낚시 시뮬레이션을 설계하고 구현하는 방법을 제시한다.

마지막 제 4장에서는 게임 엔진을 사용해서 체감형 컨트롤러를 이

용해 낚시 시뮬레이션 구현 결과를 확인하고, 이를 바탕으로 향후 연구 방향을 제안하였다.

## 제 2 장. 이론적 고찰

### 2.1. 가상현실

#### 2.1.1 가상현실의 정의

가상현실은 인공적인 기술로 만들어낸 현실이 아니지만 현실과 유사한 특정 환경이나 그 기술을 의미한다. 이렇게 만들어진 가상의 환경이나 상황 등은 사용자의 감각을 자극하여 현실과 유사한 경험을 하게함으로써 현실과 상상의 경계를 무너지게 한다. 또한, 사용자는 가상현실에 그냥 몰입할 뿐만 아니라 현실에 존재하는 디바이스를 이용해서 행동이나 조작을 하는 등 가상현실 안의 요소들과 상호작용이 가능하다.

가상현실의 중요한 특징 중에 몰입감과 상호작용이 있다. 사용자는 실시간으로 컴퓨터와 상호작용을 통해 대화하고, 이로 인해 컴퓨터가 가상현실 세계를 보여주어 사용자가 그 환경에 몰입하게 된다. 이러한 특징은 일반적인 시뮬레이션과는 구분된다.

#### 2.1.2 가상현실 기기

HMD(Head Mounted Display)는 헬멧에 장착하거나 머리에 착용하는 디스플레이 장치를 지칭하는 말이다.<sup>2)</sup> HMD는 여러 가지 종류가 있는데, 그 중 가상현실 헤드셋은 양 눈을 위한 분리된 디스플레이와 자이로스코프(Gyroscopes)와 가속도계(Accelerometers) 등을 포함한

---

2) Wikipedia, "Head-mounted display," Wikipedia, 2017, [https://en.wikipedia.org/wiki/Head-mounted\\_display](https://en.wikipedia.org/wiki/Head-mounted_display).

머리의 동작 추적 센서로 이루어져 있다. 몇몇 헤드셋은 눈 추적 센서와 컨트롤러도 포함한다.<sup>3)</sup>

### (1) 오쿨러스 리프트

오쿨러스 리프트는 페이스북(Facebook)에서 개발한 가상현실 헤드셋이다. 헤드셋이 머리의 움직임을 실시간으로 감지하여 머리가 어느 방향으로 움직이든지 그 방향으로의 시각을 제공한다. 또한 각각의 오른쪽, 왼쪽 렌즈는 오목하게 굽어진 파노라마 디스플레이 영상을 제공한다. 이는 넓은 시야각을 제공하여 눈동자를 움직여도 가상현실 디스플레이를 볼 수 있다.



[그림 1] 오쿨러스 리프트

헤드를 추적하는 기술과 양 쪽 눈에 제공되는 각각의 디스플레이는 마치 사용자가 가상현실에 들어와 있다는 착각을 하게 만드는 역할을 하게 된다. 하지만 시야각은 110°에 한정되어있고, 헤드 트래킹(Head Tracking)의 속도 문제, 시각과 청각 정보의 불일치 문제, 모션을 인

---

3) Wikipedia, "Virtual reality headset," Wikipedia, 2017, [https://en.wikipedia.org/wiki/Virtual\\_reality\\_headset](https://en.wikipedia.org/wiki/Virtual_reality_headset).

식 못하는 문제 등으로 완벽한 가상현실 구현에 한계가 있고, 사용 중 멀미 현상을 보이는 사용자가 많다는 단점이 있다. 이러한 단점은 기술의 발전으로 해결돼야 할 부분으로 지적된다.

## (2) HTC 바이브

바이브는 HTC사와 밸브(Valve)가 개발한 가상현실 헤드셋이다. 사용자의 동작을 인식하는 센서, 모션 컨트롤러 그리고 HMD로 이루어져 있다.



[그림 2] 바이브

전면에 달린 적외선 센서는 사용자가 서 있는 방을 탐지할 수 있다. 베이스 스테이션의 적외선 센서로 최대 4.5m X 4.5m의 공간 인식이 가능하다. 이러한 공간 감지센서 덕분에 사용하는 사용자의 머리 움직임 검출능력이 뛰어나다. 또한, 사용자가 가상공간에서 자유롭게 사용할 수 있는 가상공간을 제공해주는 역할을 하며, 이렇게 만들어진 가상공간 끝에는 파란색 격자무늬 벽이 생성되어 공간의 끝을 직

접 확인 할 수 있다. 또한, 스팀(Steam)과 함께 연동되어 VR 게임이  
나, 콘텐츠들을 좀 더 쉽게 이용할 수 있으며, 패키지에 들어있는 바  
이브 컨트롤러는 직접 손으로 쥐고 사용할 수 있으며 가상공간 속 물  
체와 상호작용할 수 있고, 장비 착용 후 컨트롤러가 있는 위치에 3D  
그래픽으로 컨트롤러를 표현해주어 사용자로 하여금 장비를 벗지 않  
아도 컨트롤러를 찾을 수 있다.<sup>4)</sup> 본 논문에서는 위의 이유로 바이브  
를 사용해 체험형 콘텐츠를 제작하였다. 또한 고글에는 카메라가 달  
려 있어 착용 후에도 전방을 볼 수 있고, 이 카메라를 통해 증강현실  
도 구현 가능하다. HTC는 모바일을 제작하던 회사인 만큼 PC뿐 아  
니라 전용 앱을 이용하여 안드로이드 및 애플 스마트폰과 블루투스를  
통해 연동이 가능하다.

### (3) PS VR

PS VR은 소니에서 발표한 VR 기기로, PS4에 연결하여 사용하는  
HMD이다. 시스템적인 측면에서는 무게와 해상도, 시야각은 타 VR기  
기보다 다소 떨어지나, RGB OLED 패널이 사용되어 타제품(펜타일  
사용)보다 SDE(스크린 도어 효과)가 개선되었다.



[그림 3] PSVR(Playstation VR)

4) 나무위키, "HTC Vive," 나무위키, 2017, <https://namu.wiki/w/HTC%20Vive>.

PS VR은 플레이스테이션4를 사용하며, 기존 오컬러스 리프트나 HTC 바이브와 같은 고사양의 PC를 용하지 않아도 플레이스테이션4만을 이용해 사용할 수 있다. 그렇기 때문에 기존 플레이스테이션4를 구입했던 소비자들이나 VR을 체험하고 싶은 소비자로서 상대적으로 가격부담이 덜하다는 장점이 있다. 플레이스테이션은 세계적으로 4,000만대 이상 판매 보급되어 있으며, PS VR 출시와 함께 게임 제작 회사들 또한 플레이스테이션4의 VR게임들을 내놓기 시작했고, 기존 게임들 또한 업그레이드하여 VR용으로 리메이크 하여 출시하기도 하였다. 또한 PS VR만을 구입하여 VR을 체험할 수 있기 때문에 손쉽게 가상현실을 즐길 수 있을 것이라는 평이 있지만, HMD의 고질적인 문제인 멀미를 해결하지 못했다는 의견이 제시되었다..<sup>5)</sup>

---

5) 김성욱, "플레이스테이션 VR의 가장 큰 걸림돌은 바로 이것?" 베타뉴스, 2017, <http://betanews.heraldcorp.com/article/646614>.

## 2.2. 체감형 콘텐츠

### 2.2.1 체감형 콘텐츠의 정의

체감형 콘텐츠란 신체의 다양한 움직임과 감각을 전용 컨트롤러를 이용해 실제 동작을 가상의 콘텐츠 내부에 반영하고 모방함으로써 사용자에게 현실감과 실재감을 상승시키고 새로운 경험을 제공하는 콘텐츠이다. 과거 시각과 청각에 의존하고 사용자의 움직임에 제한적인 콘텐츠 개발에 머물러 있었지만 컨트롤러가 발전하면서 물리적 감각의 상호작용과 신체의 다양한 감각을 이용한 다차원적인 정보를 이용한 콘텐츠가 개발되고 있다.

가상현실을 이용한 콘텐츠나 시뮬레이션은 사용자가 가상공간 내부에서 간접적인 체험을 경험하는 몰입감을 제공하지만 시각적 감각작용에만 의존하는 경향이 있다. 컨트롤러를 사용한 체감형 가상 콘텐츠는 사용자의 다양한 물리적 신체정보가 복합적으로 전달되면서 가상현실이 가지고 있는 몰입도와 사용자의 만족감이 상승한다. 기존의 컨트롤러는 다양한 버튼을 사용한 것이 대부분이지만 체감형 컨트롤러는 기존의 인터페이스의 틀을 과감히 버리고 사용자의 신체적 특징을 고려하여 인체공학적인 디자인을 사용하면서 다양한 동작과 실재와 같은 움직임을 효과적으로 유도한다. 컨트롤러를 통해 가상공간에서도 현실과 동일한 동작 구현이 가능하게 되고 사용자의 신체감각 상호작용으로 물리적, 공간적 제약을 감소시켜 사용자는 매개된 가상의 환경 속에서 더욱 실재감을 느끼는 주관적인 현장감이 상승한다. 이러한 체감형 컨트롤러의 현실감 있는 몰입감을 활용하여 다양한 시뮬레이션 콘텐츠가 개발되고 있다.

## 2.2.2 체감형 콘텐츠 사례

### (1) Wii Sports

대표적인 체감형 콘텐츠는 Wii 리모컨을 이용한 Wii Sports가 있다. Wii 리모컨의 동작인식 기능의 특징을 잘 살리면서 테니스, 볼링, 골프 등 실제 운동할 때와 비슷한 동작을 구현한 콘텐츠이다.



[그림 4] Wii Sport의 다양한 컨트롤러

Wii Sport는 다양한 스포츠 종목을 체감형 컨트롤러를 이용해 구현하고 각각 종목의 특징에 맞춰서 형태 및 기능을 변형한 컨트롤러를 제공하여 콘텐츠를 즐길 수 있다. 이러한 컨트롤러의 형태적 변화는 콘텐츠의 몰입도를 증가시키고 컨트롤러가 가지고 있는 물리적, 기능적 제한을 제거하고 움직임을 효과적으로 구현할 수 있게 도와 콘텐츠의 현장감을 상승시킨다.



[그림 5] Wii Sport의 테니스와 컨트롤러

Wii 테니스는 Wii 리모컨을 테니스의 라켓과 같이 사용하며 게임을 진행한다. 공을 치는 움직임과 동작을 Wii 리모컨과 센서를 이용해 감지하고 실제 테니스와 가까운 움직임을 구현해야 타격 및 게임진행이 가능하다



[그림 6] Wii Sport의 골프와 컨트롤러

골프는 넓은 필드(Field)의 특정한 장소와 장비의 필요로 누구나 쉽게 접할 수 없는 운동 중 하나이다. Wii 골프는 실제 골프를 치는 동작을 감지하여 언제, 어디서나 골프를 치는 간접적인 경험을 제공한다. 골프 전용 컨트롤러를 사용하면 좀 더 현실감 있는 게임이 가능하다.

## (2) 키넥트 어드벤처 (Kinect Adventures)

다양한 종류의 컨트롤러를 사용하여 많은 체감형 콘텐츠를 제공하는 Wii와 달리 키넥트는 컨트롤러를 사용하지 않고 사용자의 움직임을 직접적으로 인식하여 보다 자연스러운 상호작용을 구현한다.

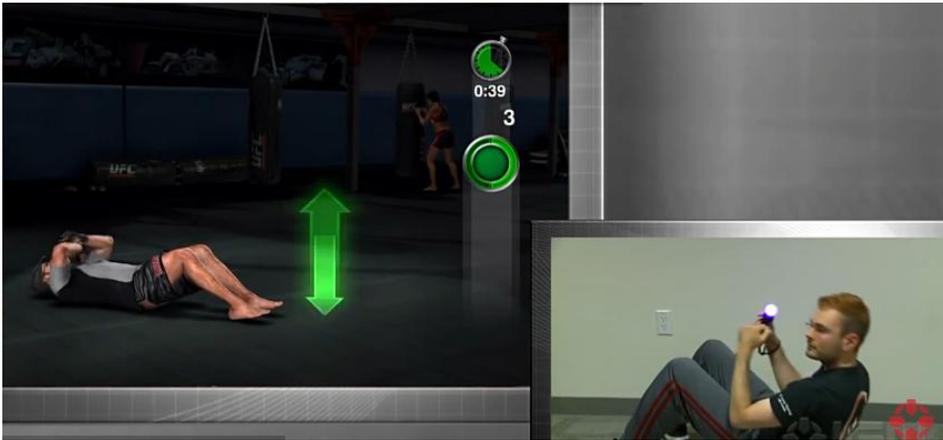


[그림 7] Kinect Adventures

키넥트 어드벤처는 사용자 자체가 컨트롤러가 되서 게임을 진행하는 컨셉의 체감형 콘텐츠이다. RGB카메라와 위치감지센서, 마이크 등을 이용해서 동작을 인식하고 외계 공간이나 위험한 급류, 고층에서 진행되는 장애물 코스 등 위험하고 현실에서 경험하기 장소에서 사용자의 움직임 제어를 통해 몰입감을 상승시키고 새로운 경험을 제공한다.

### (3) UFC 퍼스널 트레이너(UFC Personal Trainer)

플레이스테이션 카메라와 플레이스테이션 무브 컨트롤러를 이용하여 UFC 선수들에게 개인 교육을 받는 피트니스(Fitness) 콘텐츠이다. 워밍업부터 스트레칭, 선택한 운동을 하고 쿨다운(Cool Down) 스트레칭으로 끝난다.



[그림 8] UFC Personal Trainer

UFC 퍼스널 트레이너는 사용자가 자신의 체력에 맞는 운동을 하거나 원하는 운동을 하는 동안 콘텐츠 속 가상 트레이너는 조언이나 격려를 제공한다. 이는 실제로 개인 트레이닝을 받는 기분을 느끼게 한다. 콘텐츠를 진행하면서 보상을 획득할 수 있어 사용자에게 동기를 부여한다.

### 2.2.3 체감형 디바이스

닌텐도(Nintendo)가 2006년에 위(Wii)라는 콘솔 게임기를 출시하고 체감형 디바이스가 대중적으로 널리 알려졌다. 동작인식 기능을 가지고 있는 Wii 리모컨을 이용한 기존과는 다른 색다른 소재의 콘텐츠들이 만들어졌다.

이러한 체감형 디바이스가 주목받으면서 마이크로소프트(Microsoft)는 2010년에 키넥트(Kinect)를 선보였고, 소니컴퓨터엔터테인먼트(Sony Computer Entertainment)는 플레이스테이션 무브(Playstation Move)를 출시했다.

최근에는 가상현실 장비에 대한 수요가 높아지면서 체감형 컨트롤러가 개발되고 출시되고 있다.

### 2.2.4 기기 종류

#### (1) Wii 리모컨



[그림 9] Wii 리모컨

Wii 리모컨은 가속도와 기울기를 인식할 수 있는 가속도계 (Accelerometer)와 가리키는 방향을 감지할 수 있는 적외선 센서(IR Sensor)를 탑재하고 있다. 가속도계에는 작은 ‘진자’가 있고 이 진자는 고정된 두 전극의 사이를 움직인다. Wii 리모컨에 있는 가속도계는 전극과 고정된 두 전극과 진자에 달린 전극 사이의 거리 변화에 따라 발생하는 전기용량의 차이를 분석해 좌우, 앞뒤, 위아래 6 방향의 가속도의 방향과 크기를 측정한다.

Wii 리모컨은 Wii 콘솔의 센서 바(Sensor Bar)로부터 빛을 감지한다. 센서 바는 양 끝에 각각 5개의 적외선 LED가 붙어 있는 20cm 길이의 검정색 막대이다. 센서 바는 텔레비전의 위나 아래에, 화면의 맨 가운데에 수평이 되도록 설치한다. Wii 리모컨으로 화면을 가리킬 때 너무 옆쪽에서 가리키는 경우 시야각이 제한되어서 정확도가 떨어지게 된다. Wii 리모컨의 이미지 센서는 센서 바의 적외선을 찾아 위치를 계산하는 데 사용된다. 센서 바에서 나오는 2개의 밝은 빛은 이미지 센서에 의해 거리차 값과 빛을 방출하는 항상 같은 거리 값을 유지하는 두 개의 클러스터는 거리 값과 회전값은 Wii의 CPU 클러스터에서 삼각측량법에 의해 계산된다.

## (2) 키넥트



[그림 10] 키넥트

키넥트는 Microsoft의 Xbox 360, Xbox One, Microsoft Window PC에서 사용하는 주변기기다. 사용자가 컨트롤러를 사용하지 않고 동작과 음성을 인식해서 자연스러운 사용자 인터페이스로 콘솔(Console)이나 PC와 상호작용 할 수 있다.

키넥트는 중앙의 RGB 카메라와 양쪽 카메라 2개로 이루어져 있는데, 이를 통해 초당에 30번씩 인체 48개의 관절 움직임을 감지한다. 키넥트 센서는 실시간으로 깊이 정보와 RGB영상, 관절 추적 정보를 제공하는 저가의 깊이 카메라다. 키넥트의 깊이 정보 추출은 적외선 카메라가 객체를 3차원으로 표시한다. 키넥트의 3D 센서는 사용자의 움직임을 파악하고, RGB카메라는 사용자를 인식하고 플레이 중에 사진을 찍거나 비디오 촬영에 활용한다. 다중 마이크는 여러 개 설치되어 있는데, 음성 인식이나 채팅에 사용된다. 장착된 카메라 모듈을 이용해서 모션 캡처로 동작을 인식하고, 음성은 마이크 모듈로 인식한다.

키넥트의 RGB카메라 오른쪽에 있는 적외선 송출 프로젝터는 전면의 물체의 적외선을 픽셀단위로 송출한다. 이 송출된 점은 RGB카메라 왼쪽에 부착된 적외선 카메라가 반사되는 것으로 물체를 인식한다.

### (3) 플레이스테이션 무브 (Playstation Move)

플레이스테이션은 최대 4개의 컨트롤러를 사용할 수 있고, 플레이스테이션 무브 컨트롤러의 종류는 모션 컨트롤러와 내비게이션(Navigation) 컨트롤러 2가지가 있다. 두 컨트롤러의 차이점은 아래

표와 같다.

[표 1] 플레이스테이션 무브 컨트롤러 비교

	모션 컨트롤러	내비게이션 컨트롤러
주요 기능	동작 감지 위치 추적	아날로그 스틱
버튼 개수	8개	5개



[그림 11] 플레이스테이션 무브 모션 컨트롤러

플레이스테이션 무브 모션 컨트롤러는 Wii 리모컨과 비슷하게 사용자가 플레이스테이션 카메라 앞에서 동작 및 위치를 통해 콘솔과 상호 작용할 수 있게 하는 지팡이 컨트롤러다. 다양한 내부 센서가 있어서 방향을 측정한다.

플레이스테이션 무브 모션 컨트롤러의 머리에 달린 구는 RGB 발광 다이오드(LED)를 사용하여 빛을 낸다. 카메라에 포착된 사용자 환경의 색에 기초하여, 시스템은 사용자 환경과 구별되는 구의 색을 선택한다. 구의 색은 활성 마커 역할을 하며, 카메라의 위치에 따라 이미

지 평면을 따라 추적한다. 시스템은 구의 색과 크기를 통해 컨트롤러의 위치를 정밀하게 3차원으로 추적한다.

컨트롤러 내부의 한 쌍의 관성 센서, 3축 선형 가속도계 및 3축 각속도 센서가 전체 동작과 회전을 추적한다. 내부 자력계(Magnetometer)는 관성 센서에 의한 누적 오차에 대한 보정을 돕기 위해 지구 자기장에 대한 제어기의 방향을 교정한다. 또한, 내부의 온도 센서는 온도 효과에 대한 관성 센서 판독 값을 조정한다. 관성 센서는 카메라 추적이 불충분 하거나 컨트롤러가 플레이어 뒤에서 가려지는 경우에 현재의 위치를 추적한다.

플레이스테이션 무브의 이미지는 콘솔의 Cell 마이크로 프로세서(Microprocessor)에서 처리한다. 움직임 추적(Motion-tracking) 라이브러리를 사용하면 SPU(Synergistic Processing Unit)의 오버 헤드와 메모리에 충격이 가지만 그 영향은 적다.<sup>6)</sup>

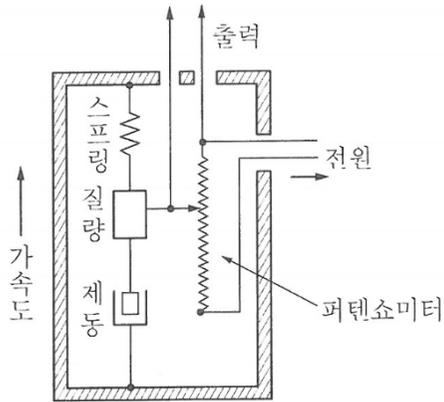
## 2.2.5 기기 원리

### (1) 가속도계

가속도계는 용수철에 매달린 추처럼 관성에 의한 반작용을 측정함으로써 직선 가속도 또는 각가속도를 측정하는 장치이다.

---

6) Wikipedia, "PlayStation Move," Wikipedia, 2017, [https://en.wikipedia.org/wiki/PlayStation\\_Move](https://en.wikipedia.org/wiki/PlayStation_Move).



[그림 12] 가속도계의 원리

가속도계의 원리는 위 그림과 같다. 케이스에 부착한 질량과 스프링의 조합으로 구성된 장치가 전체적으로 가속도를 받음으로써 케이스에 대한 질량의 상대 위치가 변화하는데, 이 변위를 재서 구한 힘의 크기를 추의 질량으로 나눈 값이다.<sup>7)</sup>

## (2) 관성 센서

운동의 관성력을 검출해서 측정 대상의 거리, 속도, 방향, 가속도 등 다양한 정보를 제공하는 장치이다.

7) 김동희, 전기용어사전 (n.p.: 일진사, 2010), 8.

## 제 3 장. 체감형 낚시 콘텐츠 기획 및 설계

본 장에서는 낚시 시뮬레이션 구현에 사용된 도구와 전체적인 플로우 차트에 대해 기술한다.

### 3.1. 설계도구

본 논문에서는 낚시 시뮬레이션 구현을 위한 개발 소프트웨어로는 유니티(Unity3D) 5.5.3 버전을 사용했고, 체감형 가상현실 기기로는 HMD와 체감형 컨트롤러를 포함하고 있는 HTC 바이브를 사용했다.

유니티(Unity3D)는 개발자가 멀티 플랫폼용 게임 개발을 쉽게 할 수 있게 해주는 혁신적인 3D 게임 엔진이지만 게임 이외의 다양한 분야에서 활용되는 범용 콘텐츠 개발 소프트웨어다. 유니티는 NVIDIA PhysX 라는 강력한 차세대 물리 엔진이 포함되어 있다. 개발자는 그 강력한 물리 엔진에 의해 복잡하고 어려운 스크립팅 없이 프로젝트에서 현실적인 여러 물리현상을 쉽게 구현할 수 있다. 또한, 유니티를 이용하여 개발된 프로젝트는 설정을 조금만 바꾸면 여러 가지 플랫폼으로 빌드가 가능하다. 이는 개발자가 여러 플랫폼을 지원하기 위해 따로 프로그램을 개발하지 않아도 된다.

## 3.2. 설계과정

### 3.2.1 개발 설계

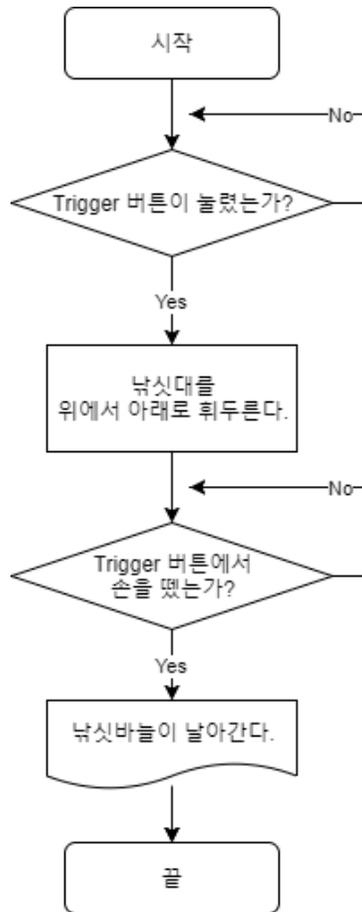


[그림 13] 낚시바늘. 낚시대 출처:google.co.kr

낚시하기 위한 장비로는 낚시대, 릴, 선, 후크나 미끼가 있다. 낚시는 크게 두 가지 동작이 필요하다. 낚시대를 던지는 동작과 물고기가 낚시바늘을 물면 물고기를 낚시 위해서 릴을 돌리는 동작이다.

#### (1) 플로우 차트 (Flow chart)

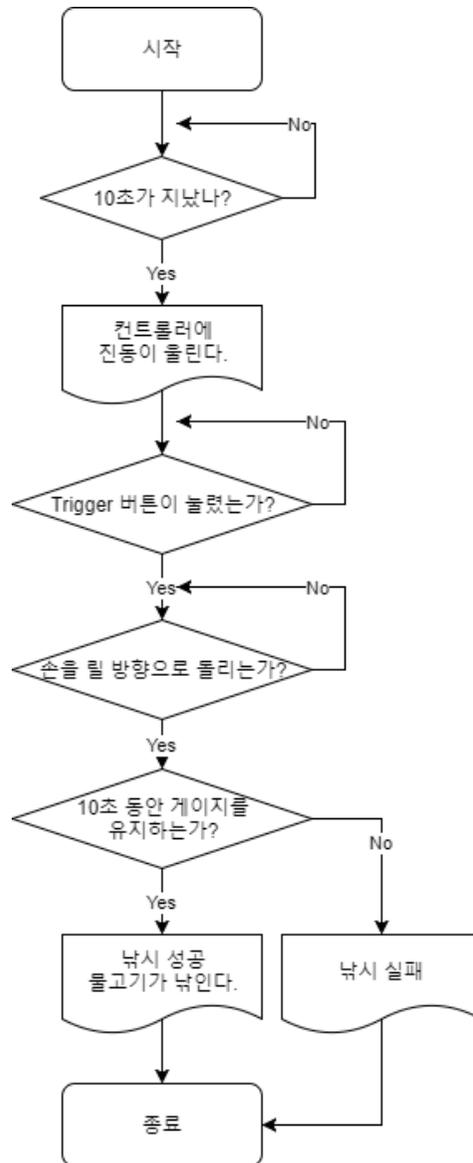
플로우 차트는 논리의 흐름을 특정한 도형이나 기호를 사용해서 도식적으로 나타낸 것이다. 본 연구에서는 낚시동작에 대해 크게 두 가지 블록으로 나누어서 플로우 차트로 나타냈다. [그림 14]는 낚시대를 휘둘러서 낚시바늘이 어떻게 날아가는지를 보여주는 플로우 차트다.



[그림 14] 플로우 차트 1

[그림 14]를 보면 프로그램을 실행시켜서 시작하면 사용자가 낙식대를 들고 있는 컨트롤러의 Trigger 버튼을 눌렀는지 확인한다. Trigger 버튼을 눌렀으면 낙식대를 위에서 아래로 휘두른다. Trigger 버튼을 누르지 않았으면 처음으로 돌아간다. 낙식대를 위에서 아래로 휘두르고 Trigger 버튼에서 손을 뗐는지 확인한다. Trigger 버튼에서 손을 뗐으면 낙식바늘이 날아간다. Trigger 버튼에서 손을 떼지 않았으면 낙식대를 휘둘렀을 때로 돌아간다. 낙식바늘이 날아가면 끝이 난다.

[그림 15]는 낚싯바늘이 바다에 떨어진 후, 물고기가 어떻게 낚이는지를 보여주는 플로우 차트다.



[그림 15] 플로우 차트 2

[그림 15]을 보면 낚싯대를 던진 후에, 10초가 지났는지 확인한다. 10초가 지났으면 물고기가 낚였다는 신호로 컨트롤러에 진동이 울린다. 10초가 지나지 않았으면 처음으로 돌아간다. 컨트롤러에 진동이 울리면 낚싯대를 들고 있지 않은 쪽의 컨트롤러의 Trigger 버튼이 눌렸는지 확인한다. Trigger 버튼이 눌렸으면 손을 반시계 방향으로 돌리는지 확인한다. Trigger 버튼이 눌리지 않았으면 전 단계로 돌아간다. 손을 반시계 방향으로 돌리면 10초 동안 게이지를 유지하는지 확인한다. 손을 반시계 방향으로 돌리지 않으면 Trigger 버튼이 눌리고 난 후로 돌아간다. 10초 동안 게이지를 유지하면 낚시를 성공해서 물고기가 낚이고 종료된다. 10초 동안 게이지를 유지하지 못하면 낚시를 실패하고 종료된다.

### 3.2.2 구현

유니티 에셋 스토어(Asset Store)에서 SteamVR을 다운받아서 Import한다.

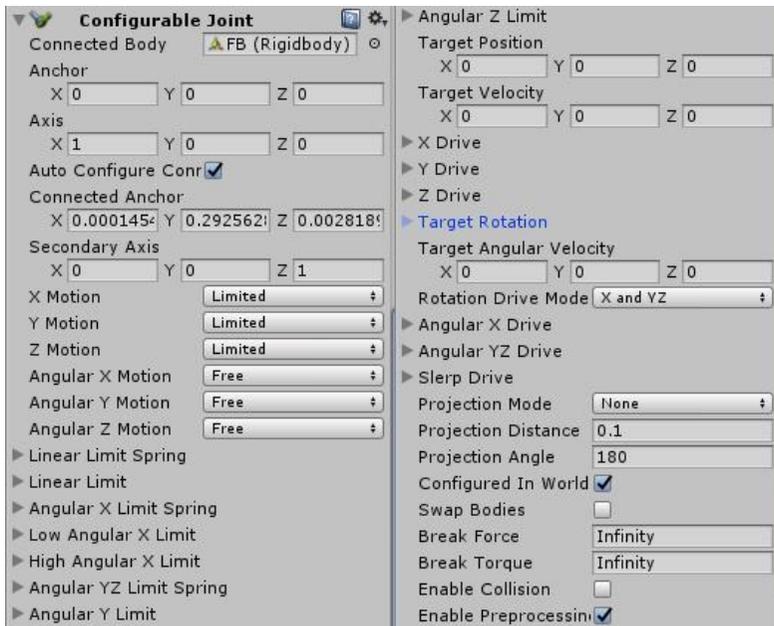


[그림 16] 바이브 설정

[그림 16]과 같이 Scene에 SteamVR의 Prefabs 폴더 안에 있는 [CameraRig]를 넣으면 바이브 개발을 위한 준비가 끝난다.

유니티에서 물리 엔진을 이용해서 객체의 물리 동작이 가능하려면 해당 객체에 Rigidbody 컴포넌트(Component)를 추가해야 한다. 낚시바늘과 Configurable Joint로 연결하는 낚시대 끝과 AddForce로 힘을 줘서 날아가야 하는 낚시바늘과 물고기가 잡혔을 때 돌릴 수 있게 Hinge Joint를 써야하는 릴은 Rigidbody 컴포넌트를 추가한다.

유니티에서 Joint는 물체와 물체 사이를 연결해준다. Joint를 이용해서 하나의 객체를 다른 객체나 고정된 점에 부착할 수 있다. 본 연구에서는 낚시대를 던지기 전, 낚시대 끝에 낚시바늘이 일정 거리에 매달려있을 수 있게 Configurable Joint를 이용했다. Configurable Joint는 모든 기능을 통합해서 가지고 있는 Joint로써 사용자가 원하는 대로 사용할 수 있다.<sup>8)</sup>



[그림 17] Configurable Joint 설정

8) 유니티 테크놀로지스, "유니티 - 매뉴얼: Configurable Joint," 유니티3D, 2017, <https://docs.unity3d.com/kr/current/Manual/class-ConfigurableJoint.html>.

낙싯대의 끝에 있는 객체에 Configurable Joint를 추가한다. [그림 17]처럼 Connected Body에 낙싯바늘 객체의 RigidBody를 연결한다. X, Y, Z Motion은 Limited로 설정하고 Configured in World Space 항목에 체크를 하면 낙싯대 끝에 낙싯바늘이 일정 거리를 유지하며 매달려 있게 된다. X, Y, Z Motion을 Limited 시키면 각 축으로의 이동을 제한한다. Configured in World Space 속성은 월드 좌표에서 물리계산을 할 지, 로컬 공간에서 물리 계산을 할지를 설정한다.

낙싯바늘을 던지기 시작할 때 모션 컨트롤러의 Trigger 버튼을 누르고, 던지는 동작이 끝나면 버튼을 떼는다. 이 때, 낙싯바늘이 날아가려면 낙싯대 끝 객체에 설정되어 있는 Configurable Joint의 Connected Body를 제거하고, 물체가 날아갈 방향으로 힘을 주어야 한다. 이를 구현하기 위해서 스크립트에서 컨트롤러의 Trigger 버튼이 눌리고 있을 때 Y 값이 가장 높은 좌표와 Trigger 버튼에서 손가락을 뺐을 때의 좌표를 이용해서 낙싯대를 던진 세기와 날아가는 방향을 정했다.

투사체가 날아가는 거리는 투사체를 던지는 속도에 비례한다.<sup>9)</sup> 본 연구에서는 팔을 휘두르는 속도를 이용해서 투사체가 날아가는 힘을 수치화하기 위해 실험을 진행했다. 컨트롤러의 Trigger 버튼이 눌리고 있을 때 Y 값이 가장 높은 좌표를 A라고 하고, Trigger 버튼에서 손가락을 뺐을 때의 좌표를 B라고 가정한다. 그리고 좌표 A일 때의 Time.time을 C라고 하고, 좌표 B일 때의 Time.time을 D라고 한다. 이 때 속도는 거리/시간이므로 던지는 속도(V)를 구하는 공식은 아래와 같다.

$$V = \frac{\|\vec{B} - \vec{A}\|}{D - C}$$

---

9) 김순윤, "창 던지기시 상지의 운동학적 분석에 관한 연구," (석사학위, 명지대학교 대학원, 1993), 5.

위의 공식으로 나온 속도 값을 AddForce의 인자로 넣게 되면 낚시바늘이 던지는 속도에 상관없이 항상 비슷하게 멀리 떨어지게 된다. 따라서 던지는 속도에 따라 낚시바늘이 날아가는 거리가 달라지도록 던지는 속도의 값을 보정할 임의의 상수 값이 필요하다. 이를 위해서 사용자가 낚시대를 휘두를 때 나오는 좌표 값과 시간을 이용해 2가지 실험을 진행했다. 첫 번째로 던지는 거리가 같고 던지는 시간이 다른 경우, 두 번째로는 던지는 거리가 다르고 던지는 시간이 같을 경우로 실험을 진행했다. 첫 번째 경우는 상수 값에 따라 좌표 A는 (0, 1, 0), 좌표 B는 (0, 1, 0.2)이고, 던지는 시간이 좌표 B일 때의 시간 D에서 좌표 A일 때의 시간 C를 뺀 0.2초, 0.4초, 0.6초로 실험을 진행했다. [표 2]는 첫 번째 실험 결과를 나타낸다.

[표 2] 던지는 시간과 상수 값에 따른 낚시바늘의 최종 좌표

	0.2 초	0.4 초	0.6 초
상수 1	(-4.7, -9.2, -361)	(13.4, -8.2, -122.2)	(-1.5, -8.2, -57.6)
상수 2	(0.4, -8.2, -104.5)	(0.5, -8.2, -35.5)	(-0.2, -8.2, -20.4)
상수 3	(-0.9, -8.2, -52.3)	(-1.1, -8.2, -18.8)	(-0.8, -8.2, -13.7)
상수 4	(0.1, -8.2, -39.7)	(0.2, -8.2, -14.8)	(-0.7, -8.2, -12.7)

[표 2]에서 상수 값이 1일 때, 던지는 시간이 0.2초라면 낚시바늘의 최종 좌표의 Z 값이 -361로 지나치게 멀기 때문에 던지는 속도를 AddForce 인자로 넣는 것은 부적합하다. 상수 값이 2일 때, 던지는 시간에 따라 적당하게 낚시바늘이 날아간다. 상수 값이 3일 때, 던지

는 시간이 0.4초 이상이면 낚싯바늘이 사용자 위치와 가까이 떨어지게 되므로 부적합하다. 상수 값이 4일 때, 던지는 시간에 상관없이 낚싯바늘이 사용자 위치와 가까이 떨어지게 되므로 부적합하다.

두 번째 실험은 상수 값에 따라 던지는 시간은 0.4초, 좌표 A는 (0, 1, 0)이고, 좌표 B는 (0, 1, 0.1), (0, 1, 0.2), (0, 1, 0.3)로 진행했다. [표 3]은 두 번째 실험 결과를 나타낸다.

[표 3] 던지는 거리와 상수 값에 따른 낚싯바늘의 최종 좌표

	좌표 (0, 1, 0.1)	좌표 (0, 1, 0.2)	좌표 (0, 1, 0.3)
상수 1	(0.6, -8.2, -36.3)	(-3.3, -8.2, -100)	(-4.3, -8.2, -253.9)
상수 2	(0.3, -0.2, -15.5)	(-0.2, -8.2, -35.2)	(-0.5, -8.2, -68.6)
상수 3	(-0.9, -8.2, -12.4)	(0.3, -8.2, -21.6)	(-0.7, -8.2, -42.3)
상수 4	(-0.6, -8.2, -12.5)	(0.1, -8.2, -16)	(0.1, -8.2, -26.9)

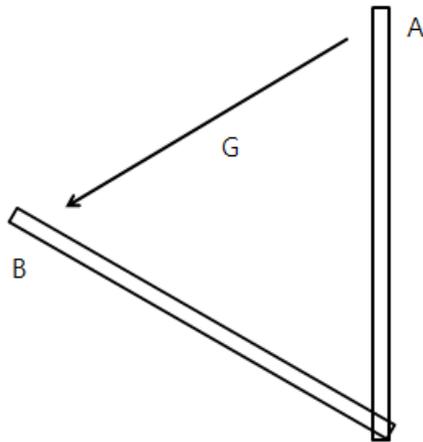
[표 3]에서 상수 값이 1일 때, 좌표 B가 (0, 1, 0.2) 라면 낚싯바늘의 최종 좌표의 Z 값이 -100으로 지나치게 멀기 때문에 던지는 속도를 AddForce 인자로 넣는 것은 부적합하다. 상수 값이 2일 때, 좌표 B에 따라 적당하게 낚싯바늘이 날아간다. 상수 값이 3일 때, 좌표 B가 (0, 1, 0.1), (0, 1, 0.2) 라면 낚싯바늘이 사용자 위치와 가까이 떨어지게 되므로 부적합하다. 상수 값이 4일 때, 좌표 B값에 무관하게 낚싯바늘이 사용자 위치와 가까이 떨어지게 되므로 부적합하다.

위 실험을 토대로 AddForce의 인자로 넣을 임의의 값 던지는 세기 (F)를 구하는 공식은 아래와 같다.

$$F = \frac{\|\vec{B} - \vec{A}\|}{2(D - C)}$$

던지는 세기를 구했으니 날아가는 방향을 구해야 한다. 낚싯바늘의 날아가는 방향(G)을 구하는 식은 아래와 같다.

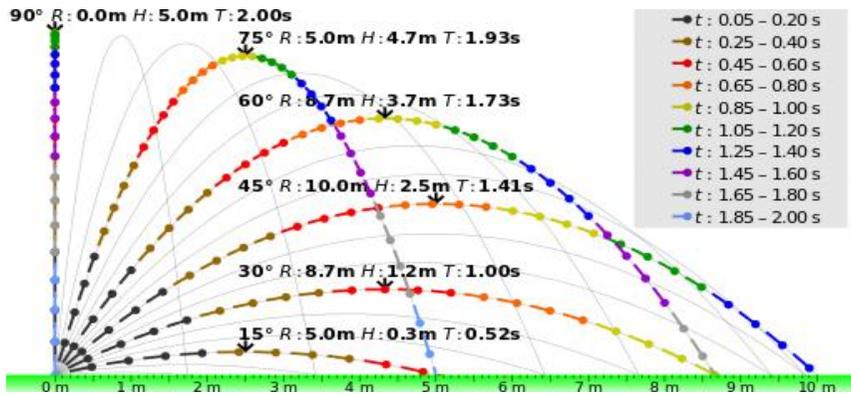
$$G = \vec{B} - \vec{A}$$



[그림 18] 던지는 방향

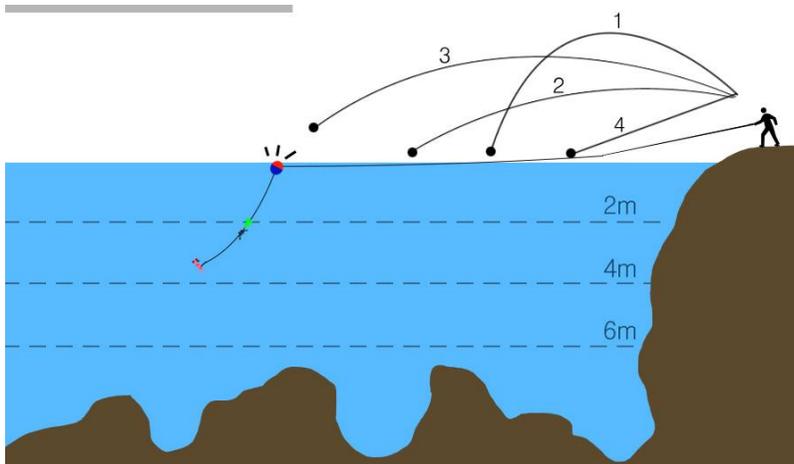
[그림 18]을 보면 벡터 G는 아래로 향한다. 낚싯바늘에 AddForce 함수를 이용해서 벡터 G 값을 넣으면 낚싯바늘은 포물선을 그리지 않고 아래로 떨어지게 된다. 이를 보완하기 위해서는 벡터 G의 Y축 각도가 위를 향해야 한다. 낚싯바늘이 날아가기 위해 적합한 각도를

찾기 위해 아래의 [그림 19] 각도에 따른 포물선 운동을 확인했다. 그러므로 벡터 G의 Y 값을 25로 바꾼다. 이렇게 나온 값 F와 G를 곱한 값을 AddForce라는 함수를 이용해서 낚싯바늘에 일시적인 힘을 가한다.



[그림 19] 포물선 운동

출처 : [https://en.wikipedia.org/wiki/Trajectory\\_of\\_a\\_projectile](https://en.wikipedia.org/wiki/Trajectory_of_a_projectile)



[그림 20] 낚싯대를 던졌을 때 찌가 날아가는 경로

출처 : <http://slds2.tistory.com/1315>

[그림 19]를 보면 다양한 포물선이 보인다. 낚싯대를 던졌을 때 이상적인 포물선 운동은 3번이다. 따라서 본 연구에서는 [그림 20]을 참고하여 y축의 각도를 30도로 설정했다.

위 내용을 바탕으로 코드를 함수화하면 아래와 같다.

```
void ThrowHangObject(Vector3 startPos, Vector3 endPos, float
                    startTime, float endTime)
{
    float throwPower = (endPos - startPos).magnitude /
        (endTime - startTime) / 7f;

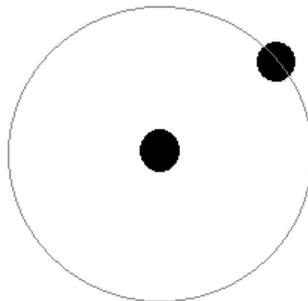
    endJoint.connectedBody = null;

    Vector3 throwAngle = (endPos - startPos) * 60f;
    throwAngle.y = 25f;

    var rigid = GetComponent<Rigidbody>();
    if (!ReferenceEquals(rigid, null))
    {
        rigid.AddForce(throwAngle * throwPower,
                      ForceMode.Impulse);
    }
}
```

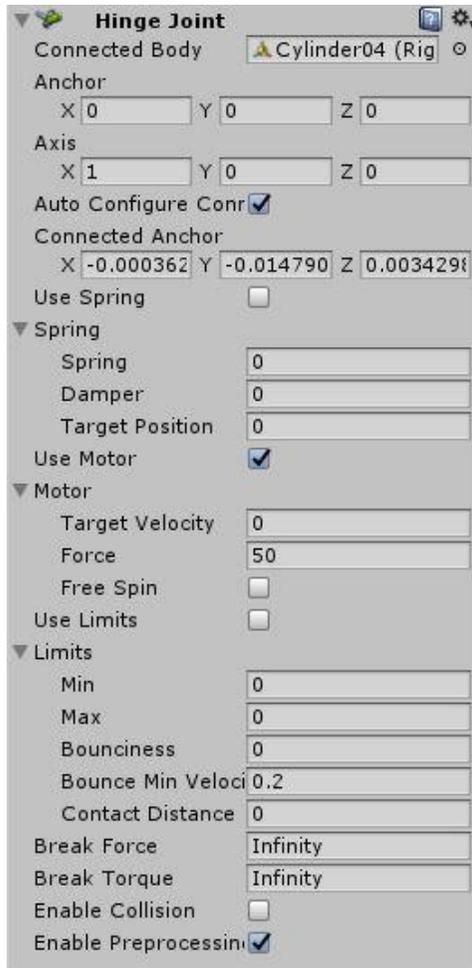
위 코드에서 endJoint는 낚싯대 끝, throwPower는 던지는 속도, startPos는 컨트롤러의 Trigger 버튼이 눌리고 있을 때 Y 값이 가장 높은 좌표, startTime은 그 때의 시간, endPos는 Trigger 버튼에서 손가락을 떼을 때의 좌표, endTime은 그 때의 시간, throwAngle은 던지는 방향, rigid는 이 스크립트가 들어있는 객체(낚싯바늘)의 Rigidbody이다.

물고기가 낚싯바늘을 물면 바이브의 모션 컨트롤러에 진동이 와야 한다. 모션 컨트롤러의 TriggerHapticPulse 함수를 이용하면 모션 컨트롤러에 약한 진동이 잠깐 온다. 본 연구에서는 긴 진동이 필요해서 코루틴(Coroutine) 기능을 이용해서 TriggerHapticPulse 함수를 여러 번 호출했다.



[그림 21] 휠

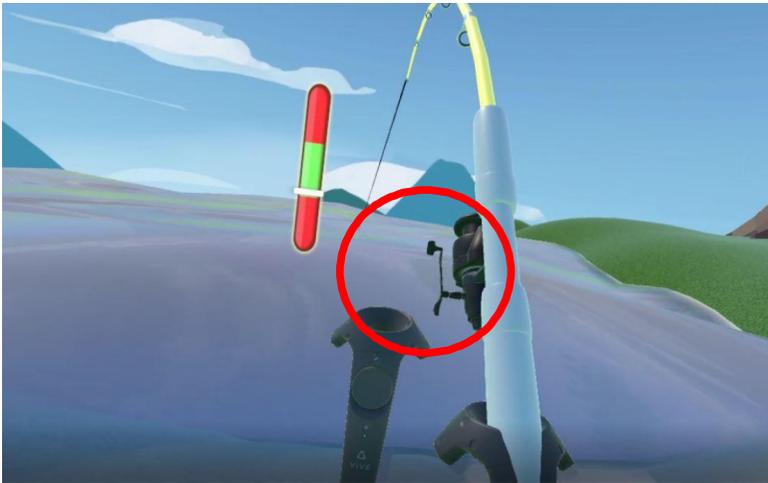
Hinge Joint는 연결된 물체를 그룹화해서 서로 연결되어 있는 것처럼 움직이도록 한다. [그림 21]처럼 휠의 손잡이 부분은 원을 그리면서 돌아가기 때문에 휠의 중앙에 객체 D를 추가하고 Hinge Joint를 이용했다.



[그림 22] Hinge Joint 설정

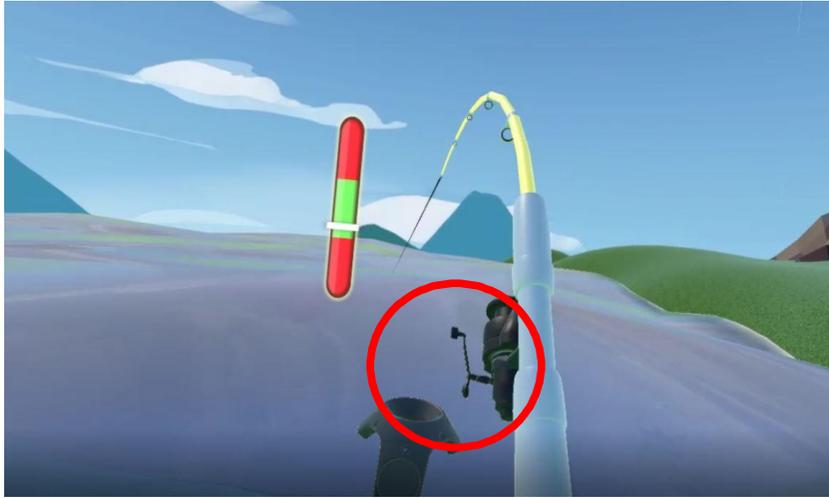
릴의 중앙에 있는 객체에 Hinge Joint를 추가한다. [그림 22]처럼 Connected Body에 릴 손잡이의 Rigidbody를 연결하고 Use Motor를 체크한다. Use Motor를 체크하면 Motor의 targetVelocity 값에 따라 물체가 회전한다. targetVelocity 값이 양수면 시계 방향으로 회전하고, 음수면 반시계 방향으로 회전한다. 모션 컨트롤러와 릴이 움직임이 같아야 하므로 본 연구에서는 targetVelocity의 값을 모션 컨트롤러의 속도(Velocity) 값의 크기를 이용했다.

모션 컨트롤러의 속도 값의 크기를 그대로 targetVelocity에 대입하면 모션 컨트롤러의 회전속도에 비해 릴의 속도가 현저하게 떨어진다. 이에 따라 본 연구에서는 모션 컨트롤러의 속도 값에 곱하는 상수 값을 200, 400, 600, 800, 1000으로 모션 컨트롤러를 반 바퀴 돌려서 모션 컨트롤러와 릴의 움직임이 똑같은지 실험했다.



[그림 23] 상수 값이 200일 때

[그림 23]을 보면 상수 값이 200일 때 모션 컨트롤러는 반 바퀴를 돌렸지만 릴은 처음 위치와 별반 차이가 없다.



[그림 24] 상수 값이 400일 때

[그림 24]를 보면 상수 값이 400일 때 모션 컨트롤러는 반 바퀴를 돌렸지만 릴은 상수 값이 200일 때보다는 더 돌아갔지만 릴의 위치는 처음 위치와 차이가 거의 없다.



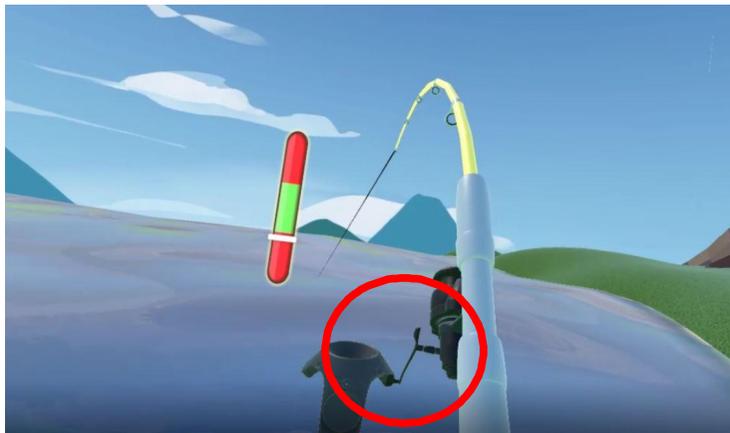
[그림 25] 상수 값이 600일 때

[그림 25]를 보면 상수 값이 600일 때 모션 컨트롤러는 반 바퀴를 돌렸지만 릴은 상수 값이 400일 때보다는 더 돌아갔지만 릴의 위치는 모션 컨트롤러의 위치보다 반 밖에 돌아가지 않았다.



[그림 26] 상수 값이 800일 때

[그림 26]을 보면 상수 값이 800일 때 모션 컨트롤러의 위치와 릴의 위치가 일치한다.



[그림 27] 상수 값이 1000일 때

[그림 27]을 보면 상수 값이 1000일 때 모션 컨트롤러보다 릴이 더 돌아가서 릴의 위치가 모션 컨트롤러의 위치보다 뒤에 있다.

위의 실험 결과를 통해 800일 때 모션 컨트롤러와 릴의 움직임이 비슷한 것을 확인했다. 이를 토대로 targetVelocity 값을 모션 컨트롤러의 속도 값의 크기에 릴을 감을 때 반시계 방향으로 돌리므로 -800을 곱한 값을 넣었다. 코드는 아래와 같다.

```
JointMotor                tempMotor                =  
jointPoint.GetComponent<HingeJoint>().motor;  
  
tempMotor.targetVelocity = -ControllerB.velocity.magnitude*800;  
jointPoint.GetComponent<HingeJoint>().motor = tempmotor;
```

위 코드에서 jointPoint는 릴의 중앙에 있는 객체고, ControllerB는 릴을 돌리는 손이 쥐고 있는 컨트롤러다.

### 3.2.3 구현 결과



[그림 28] 낚싯바늘이 매달려 있는 화면

[그림 28]은 낚싯대를 던지기 전에 낚싯바늘이 Configurable Joint를 이용해서 낚싯대에 매달려있는 화면이다.



[그림 29] 낚싯바늘이 던져진 화면

[그림 29]는 컨트롤러를 휘둘러서 낚싯바늘을 던지고 난 후의 화면이다.



[그림 30] 물고기가 잡히고 릴을 돌리는 화면

[그림 30]은 물고기가 잡히고 컨트롤러에 진동이 울려서 반대 손의 컨트롤러를 조작해서 릴을 돌리는 화면이다.

## 제 4 장. 결론 및 제언

가상현실은 사용자가 가상의 세계에 직접 종속되는 공간성을 가지고 있다. 이 공간성으로 인해 가상현실 기술은 모니터 디스플레이 시스템에 비해 확연히 높은 몰입감을 가지고 있다. 가상공간에서 유지되는 현장감은 사용자 본인이 제 3자가 아닌 가상공간 내의 주체가 됨으로써 새로운 시각적, 공간적 경험을 제공한다. 20세기 정보화시대를 거치며 발전한 컴퓨터 그래픽과 CG합성기술의 영향으로 가상현실 또한 현실과 같은 시각적 표현을 이루어내었다. 현재 가상현실 기술은 사실적 표현에 몰입감이 증대된 4차 산업혁명의 새로운 콘텐츠 분야로 인정받고 있다.

컨트롤러를 사용한 체감형 가상현실 콘텐츠는 사용자의 다양한 물리적 신체정보가 복합적으로 전달되면서 가상현실이 가지고 있는 몰입도와 사용자의 만족감이 상승한다. 컨트롤러를 통해 가상공간에서도 현실과 동일한 동작 구현이 가능하게 되고 사용자의 신체감각 상호작용으로 물리적, 공간적 제약을 감소시켜 사용자는 매개된 가상의 환경 속에서 더욱 실재감을 느끼는 주관적인 현장감이 상승한다. 이러한 체감형 컨트롤러의 현실감 있는 몰입감을 활용하여 다양한 콘텐츠가 개발되고 있다. 컨트롤러가 가상현실에 주는 몰입도와 현장감 상승을 고려하여 본 논문에서는 레저스포츠인 낚시로 체감형 시뮬레이션 제작하였다.

실감나는 체감형 콘텐츠는 현실에서 사용자의 모션 컨트롤러 움직임에 의해 현실과 비슷하게 동작해야 한다. 본 논문에서는 사용자가 가상공간에서 낚시할 때 필요한 물리적 요소를 효율적으로 구현하기 위한 연구를 진행했다. 체감형 컨트롤러를 실시간으로 추적하지 않고,

컨트롤러의 버튼을 이용해서 사용자가 낚싯대를 휘두르는 순간을 추적하여 낚싯바늘이 날아가게 구현했다. 낚싯바늘이 사용자가 낚싯대를 휘두르는 속도에 따라 낚싯바늘이 현실과 유사하게 날아가기 위해 낚싯바늘이 날아가는 거리를 실험을 통해 공식화하였다. 이는 낚시뿐만 아니라 유사한 다른 콘텐츠에서도 적용할 수 있다. 또한, 릴을 돌리는 동작은 컨트롤러의 속도 값을 사용해서 엔진의 물리기능을 이용하여 사용자가 릴을 돌리는 것처럼 손을 움직이면 릴이 돌아가도록 했다. 실험을 통해 컨트롤러의 움직임과 릴의 위치가 비슷하도록 컨트롤러의 속도 값을 보정했다. 이를 통해 컨트롤러를 사용하여 사용자가 중심점으로부터 일정한 거리를 유지하면서 돌아가는 물체를 구현할 수 있다. 그러나 낚싯대가 사용자의 움직임에 따라 자연스럽게 휘어져야 하는 것과 낚싯대의 움직임에 따른 낚싯줄의 움직임이 현실처럼 자연스럽지 않는 문제가 있다. 향후 이러한 문제를 해결하여 현존감을 높이는 연구가 필요하다.

## 참 고 문 헌

- 김순윤. "창 던지기시 상지의 운동학적 분석에 관한 연구." 석사학위, 명지대학교 대학원, 1993.
- 배일한. "가상현실 스포츠의 산업현황과 성장조건에 관한 연구 : 한국의 스크린골프 산업을 중심으로." 석사학위, 고려대학교 언론대학원, 2010.
- 김동희. 전기용어사전. n.p: 일진사, 2010, 8.
- 김성욱, "플레이스테이션 VR의 가장 큰 걸림돌은 바로 이것?" 베타뉴스, 2017, <http://betanews.heraldcorp.com/article/646614>.
- 나무위키, "HTC Vive," 나무위키, 2017, <https://namu.wiki/w/HTC%20Vive>.
- 유니티 테크놀로지스, "유니티 - 매뉴얼: Configurable Joint," 유니티 3D, 2017, <https://docs.unity3d.com/kr/current/Manual/class-ConfigurableJoint.html>.
- WiiBrew, "WiiBrew," WiiBrew, 2017, <http://www.wiibrew.org>.
- Wikipedia, "Head-mounted display," Wikipedia, 2017, [https://en.wikipedia.org/wiki/Head-mounted\\_display](https://en.wikipedia.org/wiki/Head-mounted_display).
- Wikipedia, "Virtual reality headset," Wikipedia, 2017, [https://en.wikipedia.org/wiki/Virtual\\_reality\\_headset](https://en.wikipedia.org/wiki/Virtual_reality_headset).
- Wikipedia, "PlayStation Move," Wikipedia, 2017, [https://en.wikipedia.org/wiki/PlayStation\\_Move](https://en.wikipedia.org/wiki/PlayStation_Move).

# 국 문 초 록

## 가상현실 기반의 낚시 체감형 시뮬레이션에 관한 연구

김다애

가상증강현실학과 가상증강현실학전공  
남서울대학교 대학원

다양한 가상현실 기기가 등장하면서 VR 콘텐츠가 주목받고 있다. 기존의 VR 콘텐츠는 키보드와 마우스를 이용하는 방식으로 사용자의 몰입감을 극대화시키는 데에 한계가 있었다. 이는 HTC사의 바이브와 Facebook의 오쿨러스 리프트 터치(Oculus Rift Touch) 등 상호작용을 위한 다양한 체감형 컨트롤러가 등장하면서 달라지기 시작했다. 체감형 컨트롤러를 이용한 콘텐츠는 일반적인 콘텐츠와는 다르게 직접적인 신체감각의 상호작용을 통해 사용자가 얻을 수 있는 만족감을 극대화 시켜준다. 이러한 체감형 콘텐츠로 인기가 있는 콘텐츠는 ‘닌텐도 Wii 스포츠’ 나 ‘스윙 골프 팡야’ 등이 있다. 위 콘텐츠들의 일반적인 특징은 일상생활에서 쉽게 접하기 힘들고 장비가 필요한 스포츠들이다. 낚시도 최근 낚시 카페가 생길만큼 인기가 많은 레저스포츠지만 특정한 장소와 고가의 장비가 필요해서 접근하기가 어렵기 때문에 체감형 낚시 시뮬레이션을 제작했다.

실감나는 체감형 콘텐츠는 현실에서 사용자의 모션 컨트롤러 움직임에 의해 현실과 비슷하게 동작해야 한다. 본 논문에서는 사용자가 가상공간에서 낚시할 때 필요한 물리적 요소를 효율적으로 구현하기

위한 연구를 진행했다. 체감형 컨트롤러를 실시간으로 추적하지 않고, 컨트롤러의 버튼을 이용해서 사용자가 낚싯대를 휘두르는 순간을 추적하여 낚싯바늘이 날아가게 구현했다. 낚싯바늘이 사용자가 낚싯대를 휘두르는 속도에 따라 낚싯바늘이 현실과 유사하게 날아가기 위해 낚싯바늘이 날아가는 거리를 실험을 통해 공식화하였다. 이는 낚시뿐만 아니라 유사한 다른 콘텐츠에서도 적용할 수 있다. 또한, 릴을 돌리는 동작은 컨트롤러의 속도 값을 사용해서 엔진의 물리기능을 이용하여 사용자가 릴을 돌리는 것처럼 손을 움직이면 릴이 돌아가도록 했다. 실험을 통해 컨트롤러의 움직임과 릴의 위치가 비슷하도록 컨트롤러의 속도 값을 보정했다. 이를 통해 컨트롤러를 사용하여 사용자가 중심점으로부터 일정한 거리를 유지하면서 돌아가는 물체를 구현할 수 있다. 그러나 낚싯대가 사용자의 움직임에 따라 자연스럽게 휘어져야 하는 것과 낚싯대의 움직임에 따른 낚싯줄의 움직임이 현실처럼 자연스럽지 않는 문제가 있다. 향후 이러한 문제를 해결하여 현존감을 높이는 연구가 필요하다.

**핵심어 : 가상현실, 콘텐츠, 컨트롤러, 체감형, 낚시, 시뮬레이션**

# ABSTRACT

## A Study on A Virtual Reality-based Fishing Motion-based Simulation

Kim Da-ae

Dept. Virtual-Augmented Reality

The Graduate School of Namseoul University

With the emergence of diverse virtual reality equipments, VR contents are coming into the spotlight. The existing VR contents were designed to use a keyboard and a mouse and thus had limitation in maximizing user immersion. This trend began to change with the advent of various motion controllers for interaction, such as Vive of HTC and Oculus Rift Touch of Facebook. Contents using motion controllers maximize user satisfaction from direct interaction of physical senses, unlike general contents. ‘Nintendo Wii Sports’ and ‘Super Swing Golf’ is popular as motion-based contents. Generally, these sports contents are hard to enjoy in everyday life and require subsidiary apparatus. Especially, fishing is a popular leisure sport and there are many fishing cafes recently. But it is hard to approach, since for fishing, particular place and high-priced equipment are needed. Therefore, this study aimed to produce a fishing motion-based simulation.

Realistic motion-based contents need to move in a similar way to reality using user's motion controller. This study focused on developing a method of implementing physical elements for fishing in a virtual space efficiently. Using the controller's button, the system implemented tracks the moment when an user swings the fishing rod to make the hook fly, not tracking the motion controller in real time. To make the hook fly in a similar way to reality according to the speed when an user swings the fishing rod, the flying distance of the hook was formulated, based on an experiment. This can be applied to other similar contents. Also, using the velocity value of the controller and the physical function of the engine, the reel was designed to rotate when an user moves his or her hands to turn it. Based on an experiment, the velocity value of the controller was corrected to make the motion of the controller similar to and the position of the reel. This helps an user operate a turning object at a certain distance from the central point using the controller. But the fishing rod should be bent naturally according to an user's motions and the motion of the fishing line should be natural along with the motion of the fishing rod. Further research needs to improve presence by solving these problems.

**Key word : VR(Virtual Reality), Contents, Controller, Motion-based, Fishing, Simulation**