

2017 년 졸업작품 기획서

팀 NAME

자각 마녀

조작 & Oculus touch

작성자: 전현우, 김민정

010 - 9822 - 8028, 010-7645-4351

wooloves@naver.com

수정내역

일자	내용	작업자
17.11.26	문서 초안 작성	전현우
17.11.29	플랫폼 효과 및 선정 사유 기술	전현우
17.11.30	모션 컨트롤 세부 분석 및 사용 방식 기술	전현우
17.12.02	마법진 설정 변경	김민정
17.12.06	문서 검토	전현우
18.08.09	썸 전개도 수정, 마법진 로직 오류 수정, 플레이 설정 내용 추가	김민정
18.08.10	플레이 설정 타입 플레이&UI 문서로 이동	김민정
18.08.11	검수 및 표현 교정 및 불필요한 내용 삭제	전현우

1.	개요	4
A.	조작 플랫폼	4
B.	조작 방식	4
C.	플레이 세부 기획	4
2.	조작 플랫폼	5
A.	플랫폼	5
B.	효과	6
C.	개발 범위	7
3.	조작 방식	8
A.	전체 키 설정	8
B.	인플레이	8
4.	플레이 세부 기획	9
A.	전체 씬 전개도	9
B.	기본 조작 타입	10
C.	마법진	12
D.	모션 컨트롤	15
E.	플레이 설정 타임 (구)	18

1. 개요

A. 조작 플랫폼

- i. 조작 플랫폼에 대해 설명한다.
- ii. 선택 사유와 그에 따른 효과를 설명한다.

B. 조작 방식

- i. 키 조작 역할을 설명한다.
- ii. 인플레이에서 동작하는 형태를 나타낸다.

C. 플레이 세부 기획

- i. 인플레이 조작을 세부적으로 설명한다.
- ii. 방식을 씬에 따라 설명한다.
- iii. 각각의 방식이 어떤 차이가 있는지 구별한다.

2. 조작 플랫폼

A. 플랫폼

i. 외관



ii. 소개

1. 오쿨러스사에서 직접 제작한 리프트 호환 디바이스
2. 사용자의 조작감과 그립감을 최적화 시켰다.
3. 기본 제약 조건으로 2 개의 센서를 통해 룸스케일을 구현하여, 스케일 내에서 오쿨러스 터치를 추적한다.

iii. 선택 사유

1. 오쿨러스 터치만의 그립감, 왼손과 오른손의 명확한 구분으로 컨트롤러 사용의 편안함을 더한다.
2. 모션 인식, 마법진 그리기 등 양손을 사용하거나, 한 손을 자유롭게 사용을 요구하는 기획에 맞춰 컨트롤러를 사용한다.
3. 양손의 진동을 각기 제어가 가능하여, 플레이어의 몰입도를 시각만이 아닌 감각적으로도 느끼도록 한다.
4. 룸스케일에서 플레이어의 위치를 트래킹하기 위해 반드시 컨트롤러의 추적을 필요로 한다.

B. 효과**i. 룸스케일****1. 소개**

2. 공간 내에서 완벽한 VR 환경을 조성한다. 공간 내 이동을 컨트롤러와 VR 이 추적하며, 실제 플레이어의 착용 여부도 판단한다.
3. 공간 내의 이점으로 인지 부조화가 일정 부분 감소된다.

ii. 모션 컨트롤

1. 2 개의 오큘러스 센서를 통해 공간안에 오큘러스 터치 위치, 회전 상태를 인지한다.
2. 트래킹을 활용하여, 모션의 가속도, 동작을 파악할 수 있다.

iii. 진동 센서

1. 컨트롤러의 진동의 패턴을 바꿀 수는 없다.
2. 진동의 세기는 양쪽 컨트롤러를 따로 제어한다.

3. 예시



진동의 변화를 통해 보다 몰입도 있는 플레이를 즐길 수 있다.

활을 예로 들 때, 실제로는 양손의 밸런스가 가장 중요하지만, 인간의 감각은 활시위를 당기는 손에 더 신경을 쓰게 된다. 양손의 진동을 달리함으로써 유저는 진동이 더 강한 컨트롤러에 신중을 기하며, 이는 플레이 몰입도에 지대한 영향을 끼친다.

- iv. 리프트와의 호환- 가장 문제가 되는 부분
 - 1. 룸스케일 내에서 컨트롤러의 위치를 추적하는 것은 정확하다.
 - 2. SDK 에 문제가 있다. 카메라 방향에 따른 이동을 구현 할 때, 컨트롤러와 실제 VR 의 위치가 프로그램에서 맞지 않아 모션 컨트롤 구현에 어려움을 가진다. 자체적인 개발을 필요로 한다.

C. 개발 범위

i. 개발 필요 요소

- 리프트와 컨트롤러 위치를 맞추기 필수 조건 요소

ii. ~~유저 편의성 제공~~

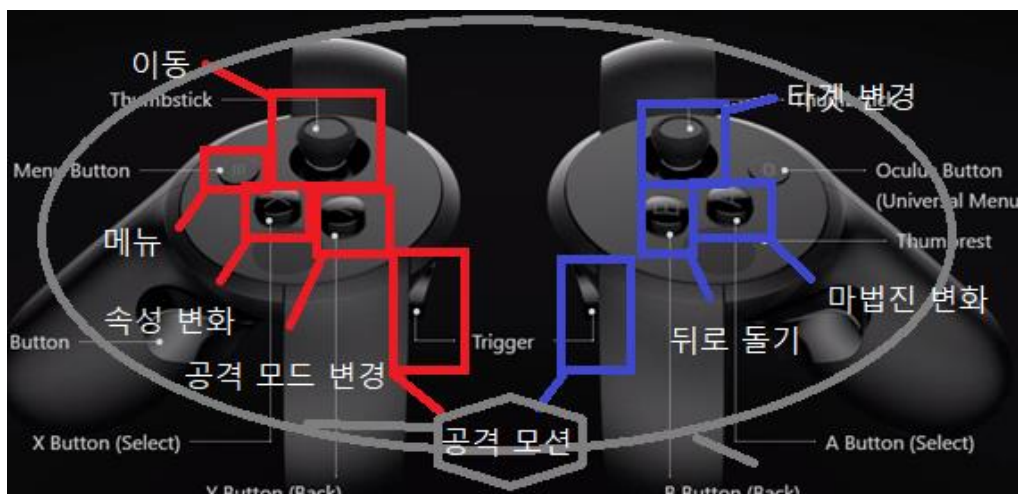
~~기본적으로 오른손잡이의 편의를 제공하나, 옵션으로 양손에 조작을 바꿀 수 있도록 구성한다.~~

3. 조작 방식

A. 전체 키 설정

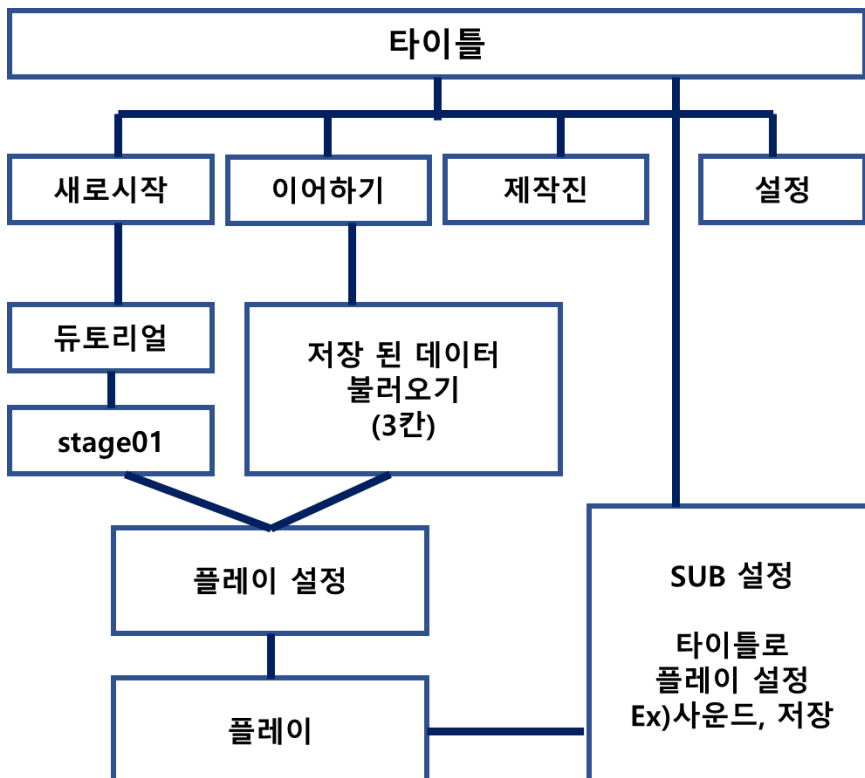
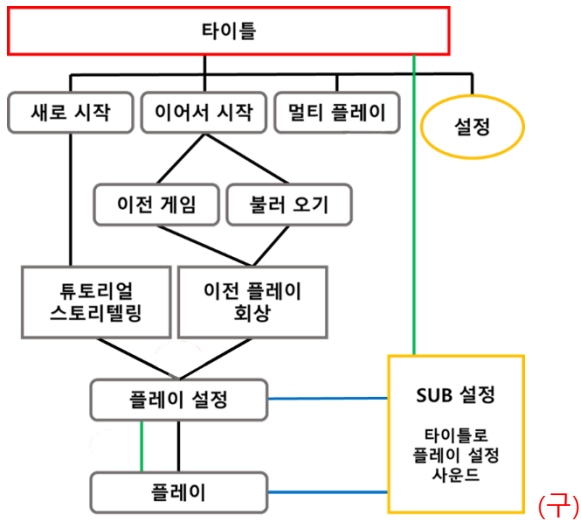
1. 양쪽 핸들 Thumbstick
 - A. 게임 내 옵션의 변경과, 인플레이에서의 이동 역할을 수행한다.
2. Trigger
 - A. 양쪽 핸들이 입력 값을 받을지 안 받을지 결정한다.
 - B. 트리거의 조작을 bool 형태의 값으로 추출 게임상의 유용한 움직임과 불필요한 움직임을 판단한다.
3. 기본 버튼 A, X
 - A. 각 옵션의 true/확인을 담당한다.
 - B. ~~인플레이에서 사용자의 주 컨트롤러 선택에 따라 부 컨트롤러의 버튼(오른손 잡이 기준 X 버튼)이 공격 대응 모드로 전환시킨다.~~
4. 기본 버튼 B, Y
 - A. 각 옵션의 false/취소를 담당한다.
 - B. ~~인플레이에서 사용자의 주 컨트롤러 선택에 따라 부 컨트롤러의 버튼(오른손 잡이 기준 Y 버튼)이 타겟 추적 모드로 전환 시킨다.~~
 - C. 각 state 의 이전 단계로 돌아가는 알람 문구 호출한다.
5. Menu Button
 - A. 플레이 설정과, 플레이 모드, 설정 state 를 제외하면 게임의 종료를 묻는 알람이 등장한다.
 - B. 플레이 설정은 저장 알람 문구와 함께 타이틀로 돌아가도록 한다.
 - C. 플레이 모드에선 Sub 설정 창을 형성, 게임이 일시 정지되도록 한다.
 - D. 설정 state에선 동작하지 않는다.

B. 인플레이

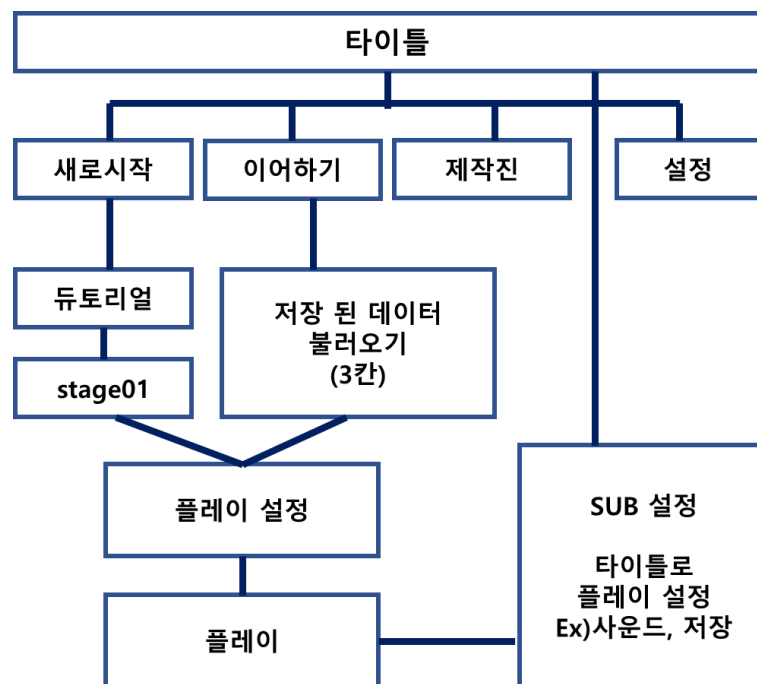


4. 플레이 세부 기획

A. 전체 씬 전개도



B. 기본 조작 타입



대부분의 영역에서의 조작

기존 VR 게임의 경우 헤드 트래킹 or 컨트롤러 위치에 의한 레이저 포인터 사용

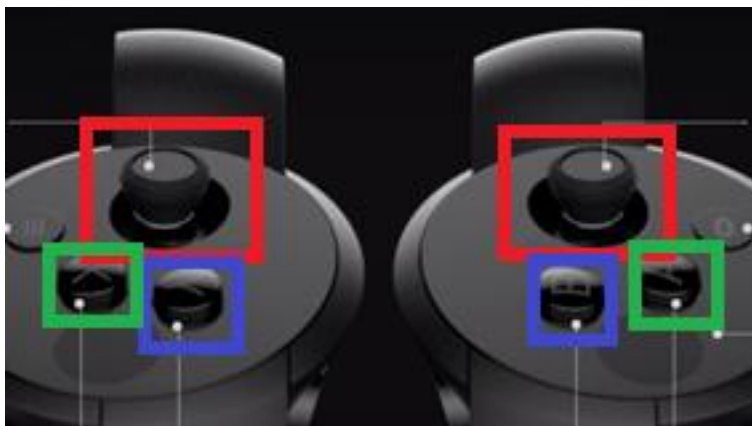
우리 작품에 경우, 이 부분은 기존 게임에서 택하는 키패드 방식을 채택한다. UI 위치에 구속 받지 않는 점을 제외하면 레이저 포인터의 이점을 크게 느끼기 어렵다.

인플레이에서 UI를 최소화하는 자각 마녀에 경우 레이저 포인터보다 키 패드 방식이 효율적이라 판단



Thumbstick으로 UI창을 활성화 시키며, A or X 버튼으로 채택, B, Y 버튼으로 이전 화면으로 복귀한다.

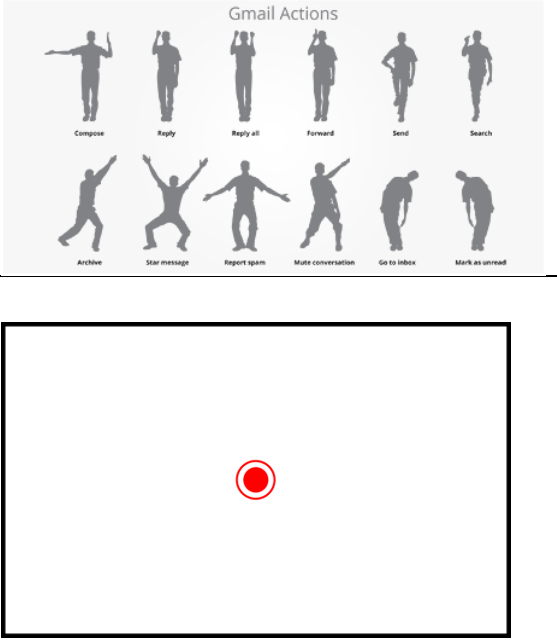
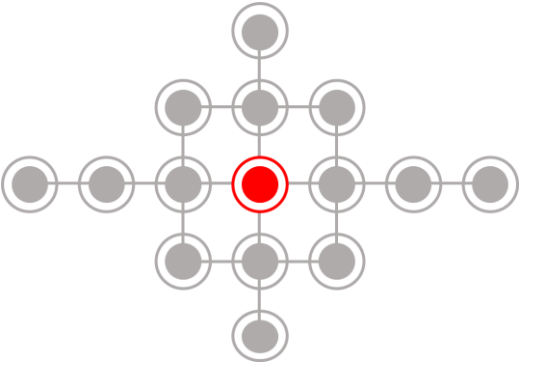
1 회성 판정의 경우 버튼 ON 에서 true, OFF 일 때 다시 false 처리를 하여 반복을 피한다.

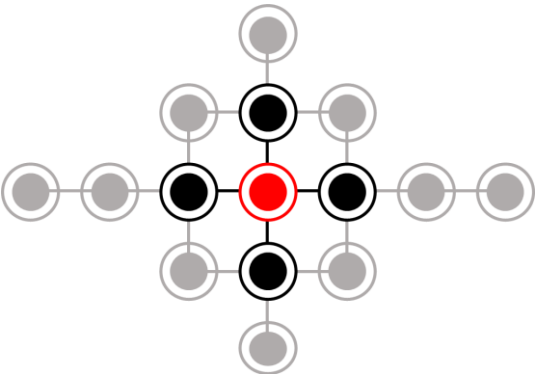
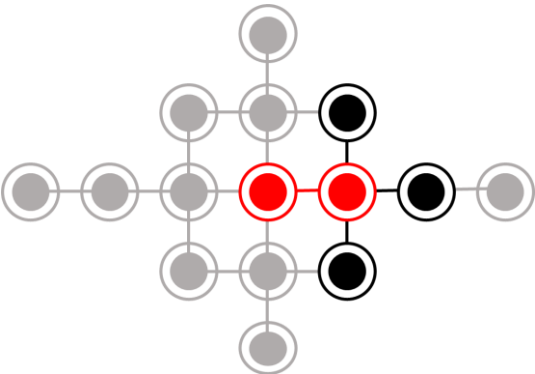


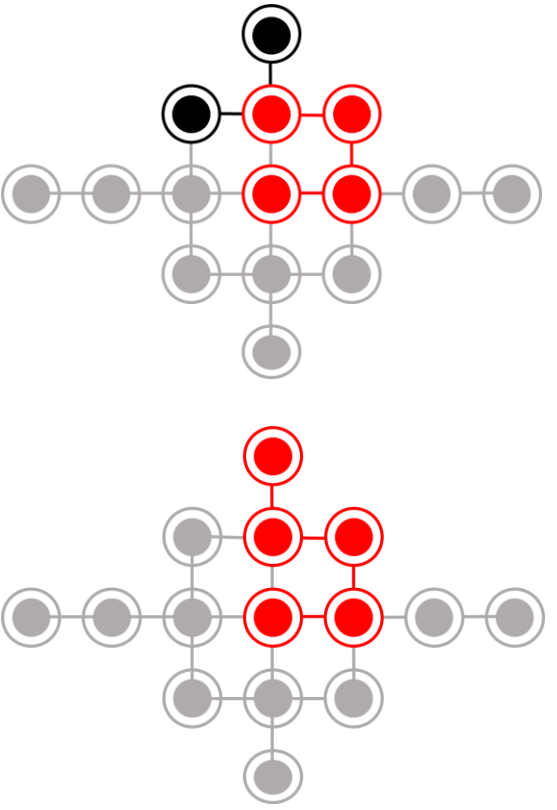
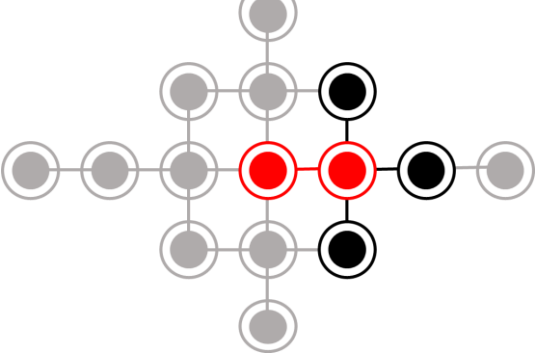
- UI 포커스 이동
- 취소, 이전 화면
- 선택 다음 UI 등장

C. 마법진

인플레이에서 주 컨트롤러의 A(X) 버튼을 누르고 있을 시, 활성화 되는 조작 상태 버튼을 누른 시점에서부터 주 컨트롤러의 위치를 트래킹하여 마법진을 입력 받는다.

시작점		<ul style="list-style-type: none"> - 컨트롤러의 버튼을 눌렀다 떼었을 순간동안 그려진 로직을 기준으로 스킬 선택을 판단한다. - 플레이어는 마법진을 그리거나, 속성을 택할 시, 각 속성에 정해진 형태를 가진 컨트롤러 모형을 가진다. - 플레이어의 위치에 상관없이 주 컨트롤러의 화면 상의 중앙에 첫 정점을 가진다.
로직 구현		<ul style="list-style-type: none"> - 장착하고 있는 스킬의 속성에 따라 정해진 로직이 다르다. - 정해진 로직 내에서라면 자유롭게 운용이 가능하다.

<p>판정</p>		<p>RED : 현재 활성화 된 마법진 BLACK : 갈 수 있는 점 (UI 표시) GRAY : 보이지 않는 점</p> <ul style="list-style-type: none"> - 버튼을 누르는 순간 바로 다음 턴으로 갈 수 있는 점이(검은점) 활성화 된다.
	 <p>(다음은 위의 활성화 된 점에서 오른쪽으로 컨트롤러의 위치를 옮겼을 때의 예)</p>	<ul style="list-style-type: none"> - 검은 점들의 위치 중 하나로 컨트롤러의 위치가 옮겨지면 옮겨진 위치의 검은점이 빨간점으로 바뀐다. - 기존의 검은점들은 모두 비활성화 되며 다시 현재 컨트롤러의 위치에서 이동할 수 있는 경로의 점들이 검은점으로 활성화 된다.

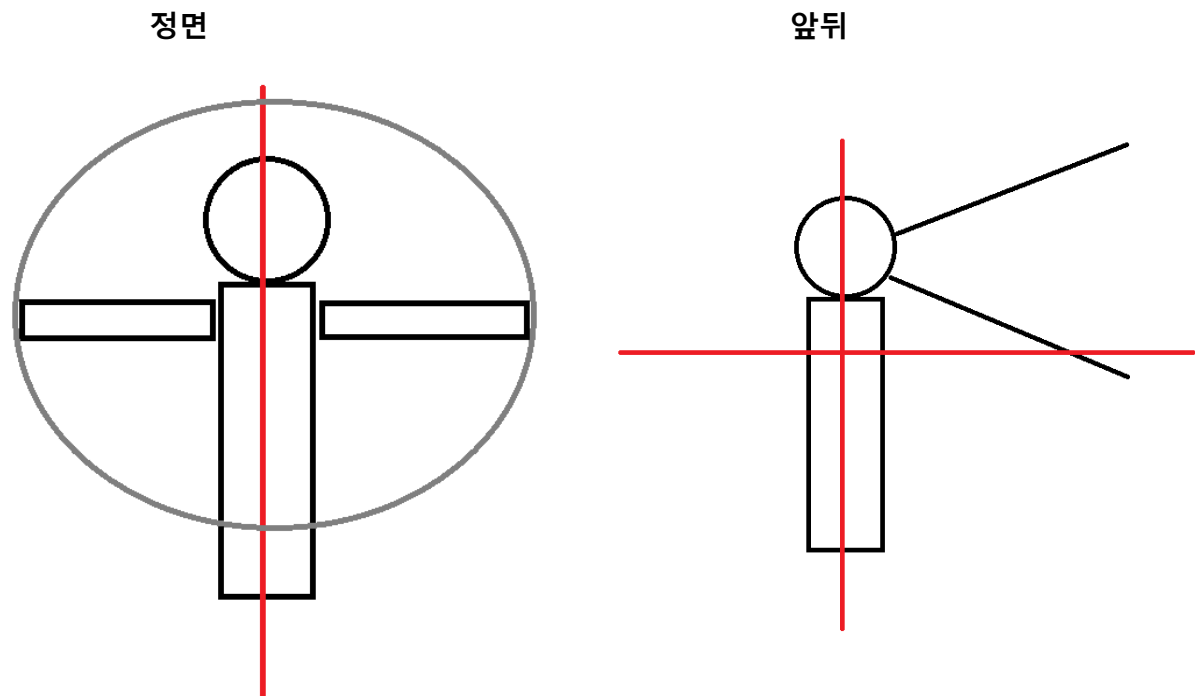
판정/성공		<ul style="list-style-type: none"> - 원하는 로직이 완성되면 컨트롤러의 버튼을 손에서 뗀다. - 손에서 떨어진 순간을 기준으로 그려진 로직을 판단하기 때문에 그 순간 검은점들이 활성화되어 있어도 완성된 로직으로 판단이 가능하다. - 왼쪽의 두 그림 모두 완성 로직 판단 가능. - 단, 기획 쪽에서 이에 대한 스킬이 등록되어 있을 경우이다.
판정/실패		<ul style="list-style-type: none"> - 현재 위치의 빨간점에서 이미 활성화 된 위치의 빨간점으로(이전 인덱스) 되돌아갈 경우 스킬이 취소된다. - 컨트롤러의 버튼을 놓았을 때 등록된 스킬 중 해당 로직과 일치하는 스킬이 없을 경우 기존 스킬이 남는다.
발동	<ul style="list-style-type: none"> - 판정 결과에 따라 해당 스킬이 저장되고, 스킬은 모션을 통해 발동, 플레이어의 모드에 따라 해당 타겟으로 향한다 공격 대응 모드 시, 플레이어를 향한 공격 중 가장 근접해 있는 공격을 타겟팅한다. 스킬 상쇄는 티어에 따라 다르며, 스킬의 티어는 추후 문서에서 작성 예정. >공격 대응 모드 제거 - 타겟 추적 모드 시, 현재 플레이어에게 타겟으로 선정된 오브젝트를 타겟팅하여 공격한다. 	

D. 모션 컨트롤

모션 컨트롤에서 컨트롤 방식을 크게 3 종류로 나눠 설명한다.

1, 영역에 따른 모션 컨트롤

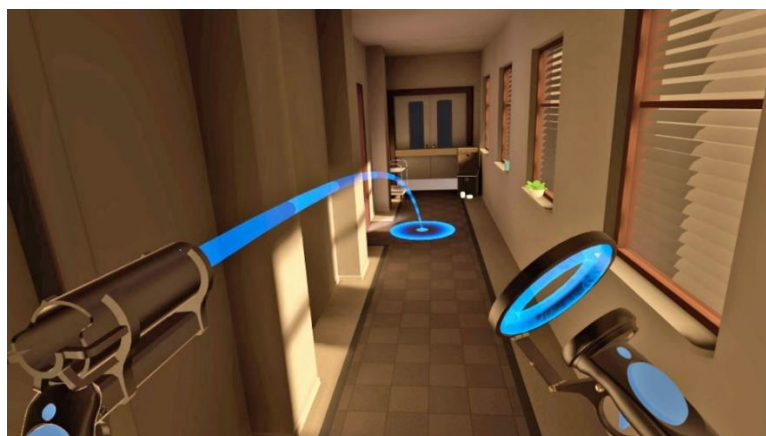
오쿨러스 터치의 활동 반경



터치의 활동 반경을 나누면 앞뒤, 상하, 좌우 총 8 개의 영역으로 나눌 수 있다. 이때, 실제 인체가 닿기 편한 반경을 감안한다면, 양 컨트롤러가 6 개의 영역에서 활동 할 수 있다.

이를 통해 영역에 따른 조작 방식을 가질 수 있다.

Ex) 텔레포트 포물선



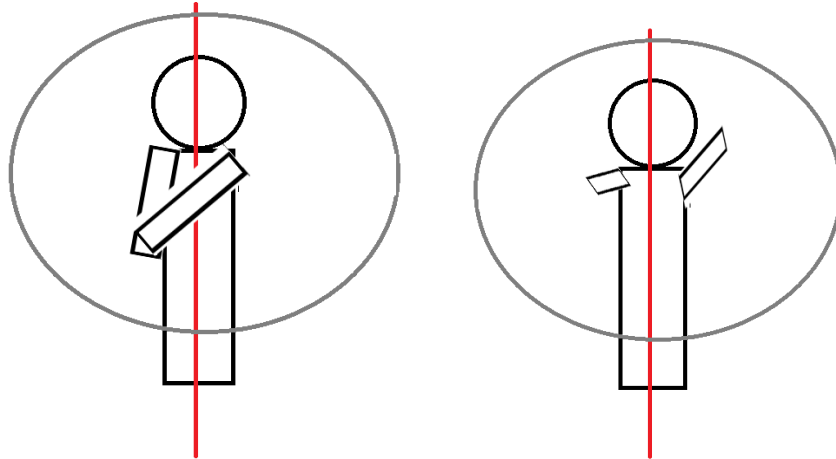
대부분에 VR 게임에서 이동 방식으로 채택하는 방식으로 이동 뿐 아니라, 소환물 제어로 사용 할 수 있다.

2, 양 컨트롤러 간의 거리 차이

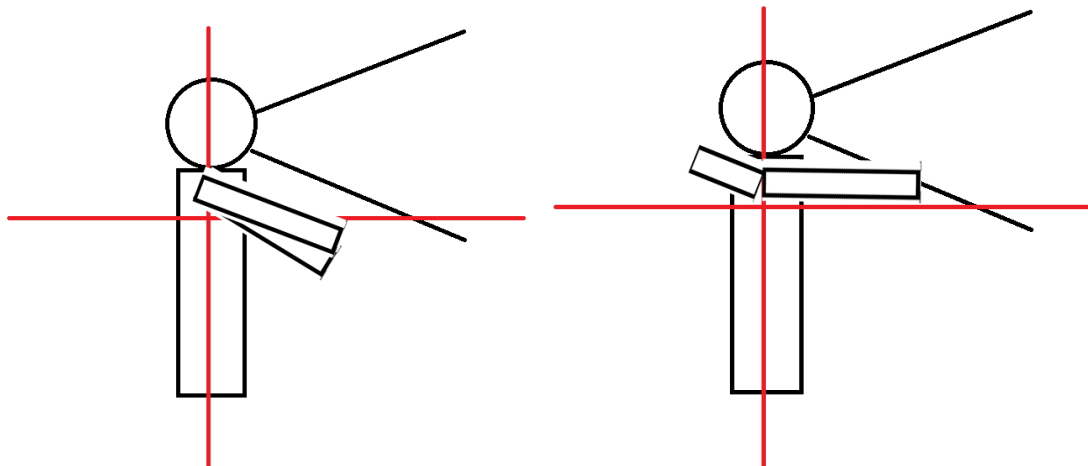
영역과 양 컨트롤러의 거리차를 이용하여 또 다른 조작 방식을 가질 수 있다.

Ex)활

정면



측면

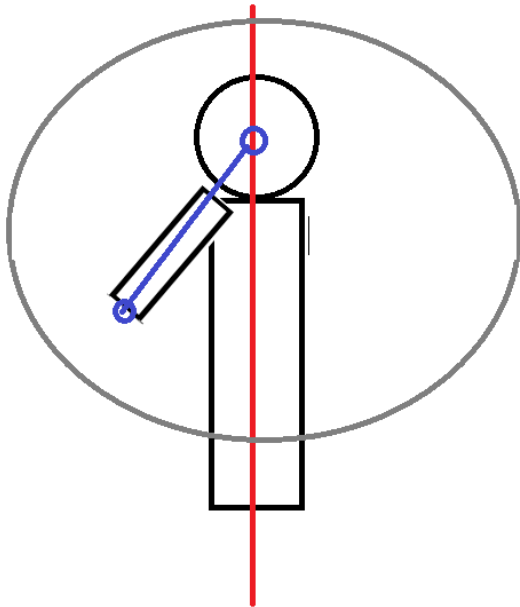


- 이를 통해 활, 창, 총 등의 모션을 취할 수 있으며, 공격 판정을 내릴 수 있다.

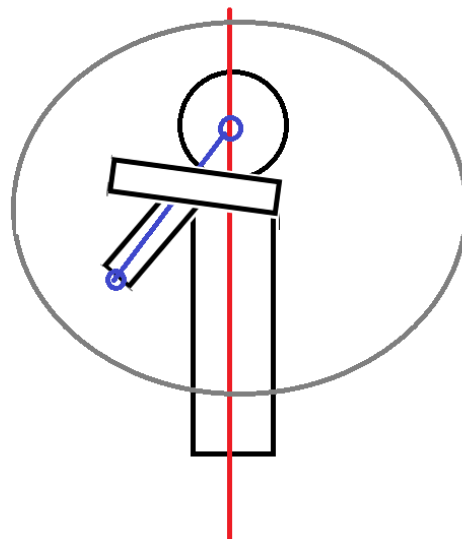
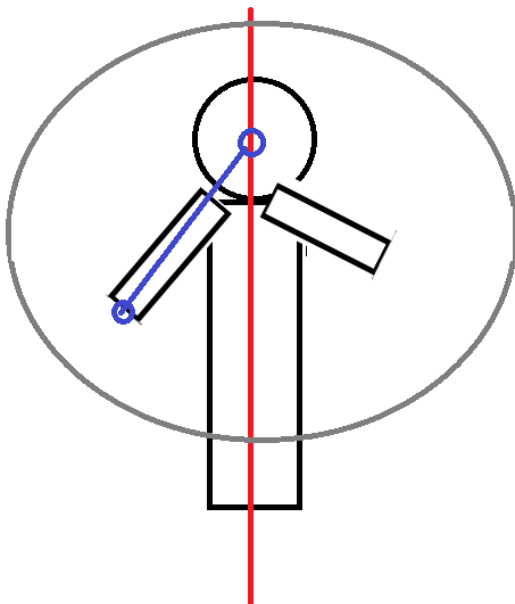
3, 컨트롤러 영역 분할

마지막으로, 2 개의 컨트롤러 중 하나를 추가적으로 영역 분할에 사용 할 수 있다.

기준점



영역 판별



이를 통해, 바이올린, 상대를 잡아서 참수 시키는 등의 패턴을 가질 수 있다.

E. ~~플레이 설정 타입 (구)~~

- 리뉴얼은 게임 플레이&UI 문서에 기재