

목차

1. 요구 조건 및 제약사항 분석에 대한 수정사항	2
1.1 요구조건.....	2
1.2 제약사항 분석에 대한 수정사항	2
2. 설계 상세화 및 변경 내역.....	3
2.1 사용자 행동(반응) 로깅	3
2.2 VR 환경 데이터 로깅 및 리플레이.....	3
2.3 VR 테스트 환경 구성	3
3. 갱신된 과제 추진 계획	6
4. 구성원별 진척도	6
5. 보고 시점까지의 과제 수행 내용과 중간 결과	7
5.1 로깅 및 리플레이 툴 개발	7
5.2 데이터 처리과정	8
5.3 중간 결론 및 개선점	9

1. 요구 조건 및 제약사항 분석에 대한 수정사항

1.1 요구조건

VR 환경 상에서 모든 상황(변화, 사용자 반응 등)을 로깅하고 리플레이한다.

- 가상 공간에서 발생하는 모든 것(사용자 행동 포함)을 로깅해서 재생
- 해당 로깅 시간과 싱크가 맞도록 사용자 데이터 로깅

1.2 제약사항 분석에 대한 수정사항

1.2.1 Unity 환경과 RealSense 측정 데이터 사이의 시간 동기화 문제

VR 환경상에서 사용자의 반응을 기록하고 3D 카메라로 VR 환경 외에서의 사용자 행동 데이터를 측정하는 과정에서 두 측정 기구의 환경이 다르기 때문에 시간적인 싱크 차이가 발생한다. 그로 인한 싱크 차이는 각각의 시간을 저장할 때 TimeStamp 를 찍고 각 환경에서의 시작 시간과 종료 시간을 기록한다. 현재 프로젝트에서는 Unity 환경상에서 신호를 주면 RealSense Camera 가 실행되는 방향으로 설정을 했기 때문에 Unity 환경이 실행되고, 얼마나 시간이 지난 후, RealSense Camera 가 실행되는지 시간 차이를 기록해 둔 후, 리플레이 시 시간 차이만큼을 더해준 상태에서 리얼센스 데이터들을 리플레이 할 계획이다.

1.2.2 Unity 로깅시 랜덤 데이터 로깅 문제

Unity 환경 상에서 리플레이시 Action Base 로 사용자의 Action 을 로깅한 후 그 Action 을 정해진 시간에 다시 재생함으로써 리플레이하는 방식을 사용하는데, Random Data 가 있을 시 Action 을 똑 같은 시간대에 다시 재생하더라도, 해당 Object 의 위치가 다르면 같은 상호작용이 일어나지 않는다. 해당 문제는 RandomDataManager 와 Randomable Script 를 만들어 사용자들이 Random Object 에 Script 를 붙여놓게 하고 그걸 파싱해서 관리할 생각이다.

1.2.3 RealSense 의 낮은 최대 fps 문제

RealSense 의 최대 fps 가 30fps 로 낮은편에 속해 사용자가 플레이한 unity 의 fps 와 차이가 많이 나, 리플레이를 할 시 끊겨 보이는 현상이 있을 수 있다. 이는 두 프레임의 스켈레톤 데이터를 선형보간으로 새로운 frame 을 생성하거나 IK 를 통해 프레임을 생성해내는 등 해결법을 찾아갈 것이다.

2. 설계 상세화 및 변경 내역

2.1 사용자 행동(반응) 로깅

원래는 NuiTrack이라는 툴을 이용하여, RealSense로부터 Skeleton Data를 추출하고 활용할 예정이었으나, 금전적인 부분이나 자료 등의 이유로 인해 Google의 MediaPipe를 사용하는 것으로 변경하였다. RealSense로 부터 RGB 데이터와 Depth 데이터를 이용하여 실제와 유사한 Skeleton을 재현할 수 있도록 하기 위해, MediaPipe의 x,y와 Depth를 이용한 z 값을 결합하여 Unity 상에서 재현하도록 할 것이다. 또한, RealSense의 RGB는 최대 30fps밖에 지원하지 않으므로, 최소 60fps로 만들기 위해, IK를 이용하여 fps를 올려줄 예정이다. 만약, 값이 갑작스럽게 튀는 현상이 발생하게 되면 칼만필터와 같은 노이즈 제거 필터를 활용하여 최대한 억제할 예정이다.

또한, 게임이 시작하면 바로 녹화를 시작할 수 있도록, 그리고 게임이 종료되는 즉시 녹화를 종료한 뒤에 Skeleton data를 추출할 수 있도록 하는 RealSenseManager 파일을 만들고자 한다.

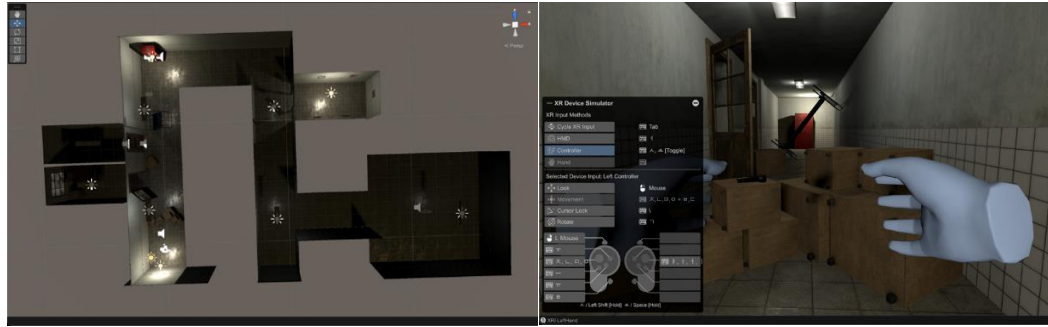
2.2 VR 환경 데이터 로깅 및 리플레이

XR Interaction Toolkit이 XR(VR+AR) 개발 환경을 점유해 나가고 있고, 차후 개발자들도 기존의 Tool보다는 XR Interaction Toolkit을 기반한 개발을 더 많이 할 것으로 생각 되어, XR Interaction Toolkit 기반으로 로깅 및 리플레이 툴 구현 방향을 변경하였다. 리플레이를 위해 로깅해야할 데이터를 크게 XR Controller Data, XR Origin Data, Random Object Data, XR Interactable Data로 크게 4종류로 구분하였다. 그 중 사용자가 직접적으로 하는 행동 데이터들은 XR Origin과 XR Controller인데 Replay Manager를 두어 Replay 모드 혹은 Recording모드에서 한번에 동기화 될 수 있게 설계 하였고, 나머지 Object 들은 Object Manager를 두어 로깅할 것이다.

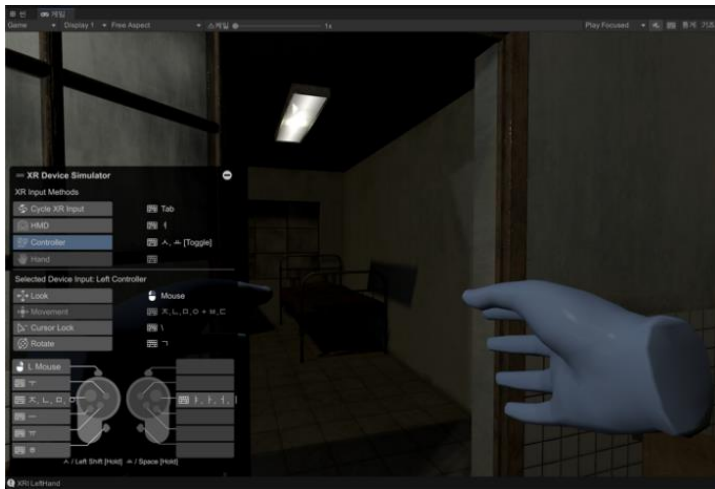
2.3 VR 테스트 환경 구성

리플레이 툴은 VR 환경 사용자의 반응에 따라 환경 또는 환경 내 오브젝트가 영향을 받았을 때, 모든 데이터를 기록하고 다시 재생한다는 점에서 효용성이 있다. 따라서 사용자의 반응(팔다리 움직임, 움짤거림, 놀람 등)을 이끌어낼 수 있는 환경을 제작하고자 하였고, XR Interaction Toolkit을 기반으로 한 간단한 좀비 게임을 설계하였다.

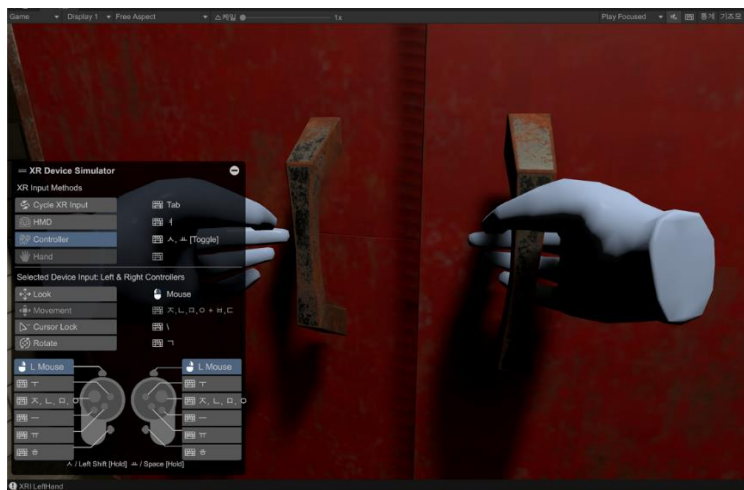
테스트 환경은 기본적으로 사용자의 공포 심리 유발을 통한 반응 도출을 목표로 한다. 이를 위해 폐병원을 배경으로 설정하였고, 조명, BGM, 사운드 이펙트, 이미지 파일 등을 사용하여 공포감을 느낄 만한 분위기를 조성하였다.



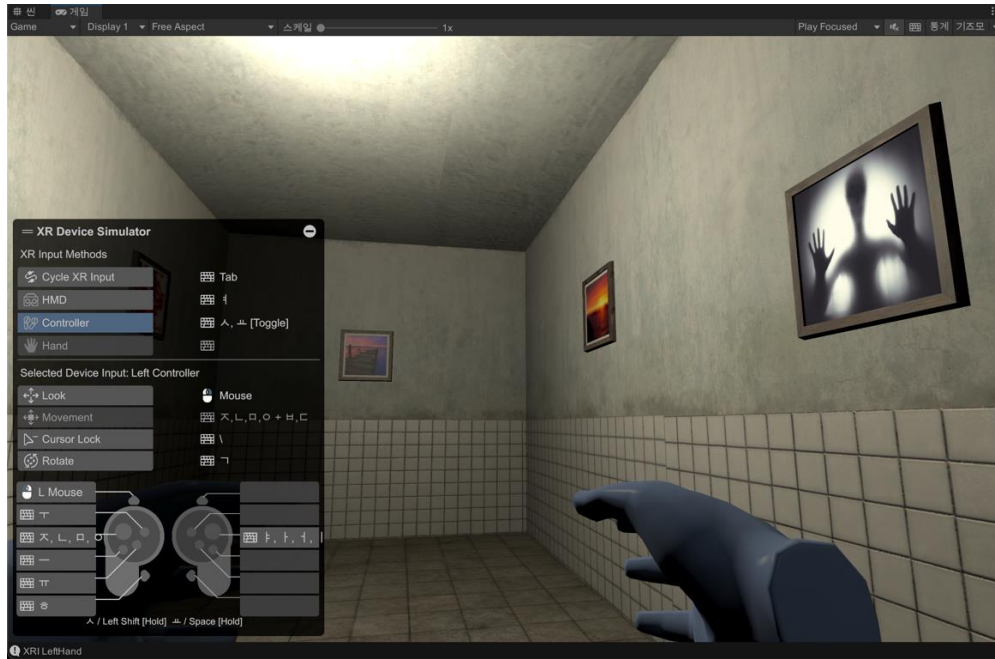
VR 플레이 시, 사용자는 맵 내의 오브젝트들과 다양하게 상호작용한다. 플레이 시 사용자는 HMD와 컨트롤러로 회전, 이동, 텔레포트 등을 사용할 수 있도록 구현하였다.



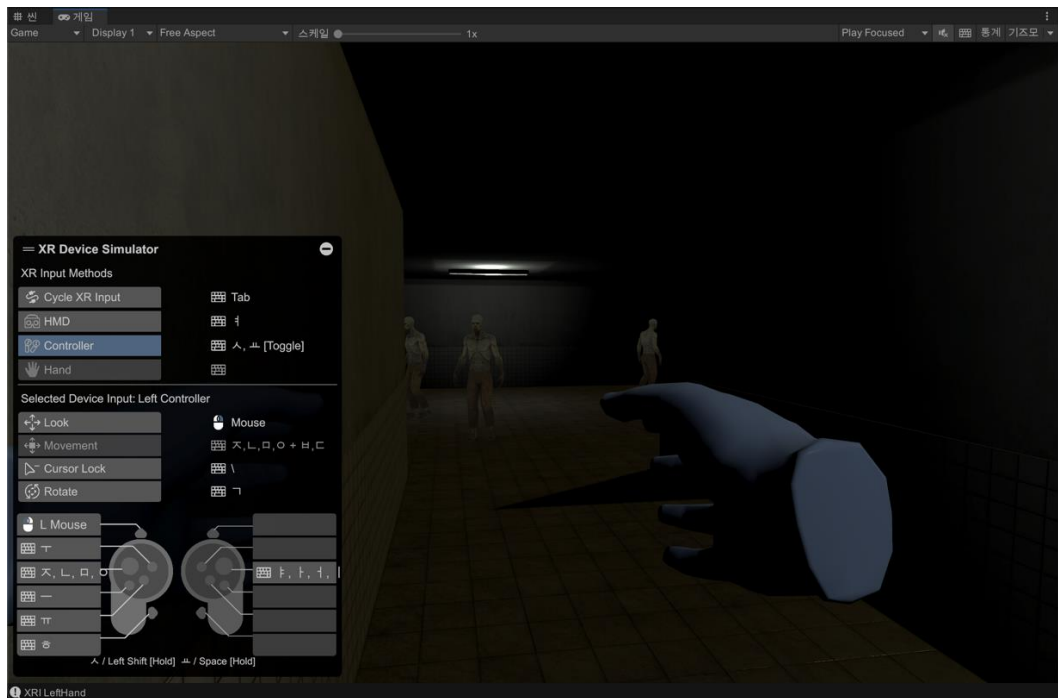
사용자와 문 사이의 거리를 프레임별로 측정하여 5m 이하가 되었을 때 창문이 깨지고 방 안에 줌비가 울음소리와 함께 등장하는 상호작용을 구현하였다.



복도 끝의 캐비닛 오브젝트를 열고, 이벤트 스크립트를 처리하여 캐비닛 안에서 박쥐 오브젝트가 튀어나오도록 구현하였다.



사용자가 액자 오브젝트를 바라보면 그림이 변경되는 상호작용을 구현하였다.



좀비 오브젝트를 배치하였다. 사용자가 오브젝트와 5m 이하로 가까워지면 좀비 오브젝트는 사용자를 타겟으로 하여 움직인다. 좀비 오브젝트에 닿으면 좀비의 공격을 받는다.

맵 내의 모든 오브젝트는 사용자와의 상호작용에 따라 State가 변경되며, 상호작용은 리플레

이 툴을 사용할 수 있도록 Action-based로 구성되었다.

랜덤 데이터를 이용해 구현할 수 있는 오브젝트들은 데이터 고정 버전과 랜덤 버전의 스크립트를 따로 작성하여 추가하였고, 차후 진행상황에 따라 활용할 예정이다.

3. 갱신된 과제 추진 계획

Task Name	5월 4주	5월 5주	6월 1주	6월 2주	6월 3주	6월 4주	7월 1주	7월 2주	7월 3주	7월 4주	8월 1주	8월 2주	8월 3주	8월 4주	8월 5주	9월 1주	9월 2주	9월 3주	9월 4주
Unity 및 VR 개발 공부	■	■																	
VR 환경 및 로깅 서버 구축			■	■	■	■													
데이터 로깅 환경 구축					■	■	■	■	■	■									
중간보고서 및 중간평가표 제출											■	■	■	■	■				
데이터 로깅 환경 및 피실험자 모집											■	■	■	■	■	■	■	■	■
사용자 평가 및 피드백																			
최종보고서 작성																			
최종보고서 및 최종평가표 제출																			

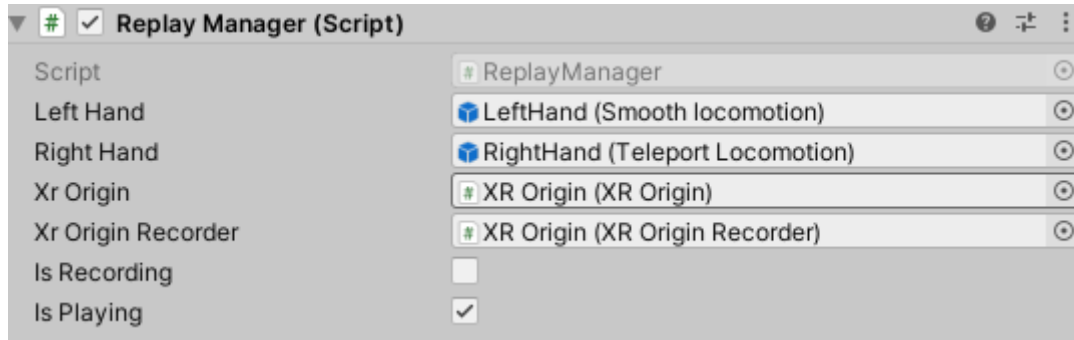
4. 구성원별 진척도

이름	진척도
최현호	리플레이 툴 개발
이희근	사용자 행동 데이터 처리
김지원	테스트 환경 구성

5. 보고 시점까지의 과제 수행 내용과 중간 결과

5.1 로깅 및 리플레이 툴 개발

Replay Manager를 통해 XR Controller Recorder와 XR Origin Recorder를 한번에 관리하도록 설계하였다.



XR Interaction Toolkit에서 제공하는 XR Controller Recorder를 이용하여 XR Controller의 각 프레임 별 Time, Position, XR Interaction State를 로깅하였다.

8.412242	(-0.17, 1.23, 0.31)	(0.03341, 0.18887, -0.3723)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0	0	0
8.466704	(-0.16, 1.23, 0.31)	(0.02503, 0.16443, -0.3691)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	0.7	0.8	0.8
8.524654	(-0.16, 1.23, 0.31)	(0.02214, 0.15977, -0.3698)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	0.9	0.9	0.9
8.577983	(-0.16, 1.23, 0.31)	(0.02156, 0.15677, -0.3682)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	0.9	1	1
8.642828	(-0.16, 1.23, 0.31)	(0.02131, 0.14830, -0.3683)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	1	1	1
8.691408	(-0.16, 1.23, 0.31)	(0.01885, 0.14727, -0.3699)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	1	1	1
8.758728	(-0.16, 1.23, 0.31)	(0.01787, 0.14852, -0.3709)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	1	1	1
8.803043	(-0.16, 1.23, 0.30)	(0.01715, 0.14947, -0.3717)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	1	1	1

XR Controller의 Roation 값은 Controller Recorder에서 로깅되지 않아, XR Origin Recorder를 따로 만들고 그 안에 XR Origin의 position 및 rotation Data, Main Camera의 상태, Controller의 Rotation 상태 등을 저장해 두었다.

▼ Element 81

Time	2.01690053939819					
Position	X	1.28733	Y	1.492	Z	-2.00903
Rotation	X	0	Y	90.96327	Z	0
Camera Position	X	1.311937	Y	3.10962	Z	-2.035412
Camera Rotation	X	15.05686	Y	115.6502	Z	359.9781
Left Transform Rotation	X	345.7411	Y	117.4708	Z	31.29398
Right Transform Rotation	X	10.55774	Y	107.768	Z	9.195623

▼ Element 82

Time	2.05439019203186					
Position	X	1.273846	Y	1.492	Z	-1.974049
Rotation	X	0	Y	90.96327	Z	0
Camera Position	X	1.297995	Y	3.10941	Z	-1.99975
Camera Rotation	X	15.11005	Y	115.7577	Z	0.1309182
Left Transform Rotation	X	346.2125	Y	117.3063	Z	31.35481
Right Transform Rotation	X	11.63319	Y	107.474	Z	9.308953

▼ Element 83

Time	2.0859522819519					
Position	X	1.262615	Y	1.492	Z	-1.944553
Rotation	X	0	Y	90.96327	Z	0
Camera Position	X	1.285792	Y	3.108565	Z	-1.968584
Camera Rotation	X	15.2963	Y	116.0298	Z	0.4545456
Left Transform Rotation	X	347.3126	Y	116.3923	Z	31.46185
Right Transform Rotation	X	13.40592	Y	106.9162	Z	10.03319

5.2 데이터 처리과정

RealSense의 최대 fps를 활용할 수 있도록 하기 위해, RGB Image와 Depth Image를 bag파일로 저장할 수 있도록 하였다. 그 뒤, Google의 MediaPipe에 RGB Image를 넣음으로써 Skeleton Data를 추출 하여, 정규화된 x,y값을 이용해 실제 RGB Image에서의 x, y값을 추정하도록 한다. 하지만 MediaPipe의 z 값은 실제 depth값이 아닌 추정값이므로 이를 RealSense Data의 Depth data로 변경해주도록 하였다. 위의 과정에서 얻은 RGB Image에서의 x, y값으로 부터, Depth Image에서의 z 값을 가져와 {Timestamp, Joint Num., x, y, z} 형태로 txt 파일로 저장하도록 하였다.


```

skeleton_data - Windows 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
1690919374600.5984, 0, 0.4560841917991638, 0.35473334789276123, 4310
1690919374600.5984, 1, 0.46071895956993103, 0.3454342484474182, 4255
1690919374600.5984, 2, 0.4639877378940582, 0.34582215547561646, 4310
1690919374600.5984, 3, 0.46654924750328064, 0.346244752407074, 4339
1690919374600.5984, 4, 0.4515480399131775, 0.34548091888427734, 4237
1690919374600.5984, 5, 0.4488369822502136, 0.34602686762809753, 4210
1690919374600.5984, 6, 0.44604548811912537, 0.34669259190559387, 4192
1690919374600.5984, 7, 0.4708287715911865, 0.35523802042007446, 4386
1690919374600.5984, 8, 0.4436652660369873, 0.35649147629737854, 4210
1690919374600.5984, 9, 0.4625716209411621, 0.3704507648944855, 4282
1690919374600.5984, 10, 0.451115220785141, 0.37066471576690674, 4237
1690919374600.5984, 11, 0.49467015266418457, 0.43701088428497314, 2263
1690919374600.5984, 12, 0.4266604483127594, 0.43838921189308167, 2284
1690919374600.5984, 13, 0.5089044570922852, 0.52731119211196899, 2313
1690919374600.5984, 14, 0.4108176827430725, 0.5287066698074341, 7442
1690919374600.5984, 15, 0.5253332853317261, 0.594721257686615, 4568
1690919374600.5984, 16, 0.39451146125793457, 0.5966523289680481, 4621
1690919374600.5984, 17, 0.5309135913848877, 0.6221402883529663, 3889
1690919374600.5984, 18, 0.38864845037460327, 0.6187639236450195, 4006
1690919374600.5984, 19, 0.5277819633483887, 0.62297123670578, 3829
1690919374600.5984, 20, 0.39122259616851807, 0.6168335676193237, 4022
1690919374600.5984, 21, 0.5246657133102417, 0.6161404252052307, 3799
1690919374600.5984, 22, 0.3947507441043854, 0.6092941761016846, 4210
1690919374600.5984, 23, 0.4780293107032776, 0.6231516599655151, 3792
1690919374600.5984, 24, 0.43612295389175415, 0.6211328506469727, 3912
1690919374600.5984, 25, 0.4830091595649719, 0.7809181213378906, 2966
1690919374600.5984, 26, 0.4290925860404968, 0.7832822799682617, 2914
1690919374600.5984, 27, 0.4940332770347595, 0.9228003025054932, 2205
1690919374600.5984, 28, 0.4185296893119812, 0.921965479850769, 2203
1690919374600.5984, 29, 0.48944395780563354, 0.9404228925704956, 2100
1690919374600.5984, 30, 0.4174986779689789, 0.9389181137084961, 2094
1690919374600.5984, 31, 0.5116727352142334, 0.9617682695388794, 1995
1690919374600.5984, 32, 0.40878811478614807, 0.9682214260101318, 2017
1690919374703.681, 0, 0.45608243346214294, 0.35457178950309753, 4192
1690919374703.681, 1, 0.460700660943985, 0.34499648213386536, 4264
1690919374703.681, 2, 0.46392232179641724, 0.345459520816803, 4282
1690919374703.681, 3, 0.46654486656188965, 0.34604412317276, 4292
1690919374703.681, 4, 0.4515942931175232, 0.34483397006988525, 4219
1690919374703.681, 5, 0.44885969161987305, 0.3452872633934021, 4210
1690919374703.681, 6, 0.4460813105106354, 0.345970094203949, 4201
1690919374703.681, 7, 0.47082725167274475, 0.35524144768714905, 4320
1690919374703.681, 8, 0.4436771869659424, 0.35621288418769836, 4157
1690919374703.681, 9, 0.4625658392906189, 0.3699345588684082, 4310
1690919374703.681, 10, 0.451066792011261, 0.3696064054965973, 4273
1690919374703.681, 11, 0.4946777820587158, 0.436229407787323, 2253
1690919374703.681, 12, 0.4370212550390073, 0.43040346406555176, 2284
Ln 18, Col 69 100% Windows (CRLF) UTF-8

```

5.3 중간 결론 및 개선점

- Random Data들을 위한 Object Manager가 아직 미흡하고, Recording File이 Unity Editor 가 꺼지면 같이 날라가는 현상이 있어 개선이 필요하다.
- 사용자를 쫓아오는 오브젝트로부터 도망칠 때 이동 시 VR 멀미가 발생하므로, 사용자의 VR 멀미를 완화할 수 있도록 개선할 예정이다. 그리고 사용자 행동에 따른 오브젝트의 반응이 최대한 딜레이가 없도록, 원활한 반응을 위해 코드 실행 사이 대기시간 조정, 음향과 조명 설정 변경 등으로 스크립트를 보완할 예정이다. 또한 이후 피실험자가 테스트 환경 시작과 종료 및 전체 플레이 상황을 더욱 쉽게 이해하도록 사용법 UI를 추가할 예정이다.
- RealSense로 부터 얻은 Skeleton Data를 Unity에 재현해본 결과, 약간씩 값이 튀는 현상이 발견되었고 이를 최대한 줄이기 위해 칼만필터를 적용할 예정이고, fps 저하문제를 해결하기 위해 IK를 적용할 예정이다.

- 사용자 생체 신호를 스마트 워치를 사용해 실시간으로 로깅할 수 있는지 알아보고, 만약 가능하다면 사용자의 심박수를 로깅하여 VR 환경 플레이 화면에 심박수가 디스플레이될 수 있도록 개발할 예정이다.