

종합설계 프로젝트 수행 보고서

| | |
|-------|--|
| 프로젝트명 | VR을 이용한 자세교정 시스템 |
| 팀번호 | S5-8 |
| 문서제목 | 수행계획서() 2차발표 중간보고서() 3차발표 중간보고서() 최종결과보고서(O) |

2020.11.23

팀원 : 류호성
양인모
이승룡

지도교수 : 이보경 (인)

문서 수정 내역

| 작성일 | 대표작성자 | 버전(Revision) | 수정내용 | |
|------------|-------|--------------|-----------|------|
| 2020.01.11 | 이승룡 | 1.0 | 수행계획서 | 최초작성 |
| 2020.01.18 | 류호성 | 1.1 | 수행계획서 | 내용추가 |
| 2020.02.29 | 양인모 | 2.0 | 중간발표1 보고서 | 내용추가 |
| 2020.05.02 | 이승룡 | 3.0 | 중간발표2 보고서 | 내용추가 |
| 2020.06.25 | 양인모 | 4.0 | 중간발표3 보고서 | 내용추가 |
| 2020.11.23 | 류호성 | 5.0 | 최종결과보고서 | 내용추가 |

문서 구성

| 진행단계 | 프로젝트 계획서 발표 | 중간발표1 (2월) | 중간발표2 (4월) | 학기말발표 (6월) | 최종발표 (10월) |
|------------|-----------------|-----------------|-----------------|-----------------|----------------|
| 기본양식 | 계획서 양식 | 계획서 양식 | 계획서 양식 | 계획서 양식 | 계획서 양식 |
| 포함되는 내용 | I. 서론 (1~6) | I. 서론 (1~6) | I. 서론 (1~6) | I. 서론 (1~6) | I II III |
| | II. 본론 (1~3) | II. 본론 (1~4) | II. 본론 (1~5) | II. 본론 (1~7) | |
| | 참고자료 | 참고자료 | 참고자료 | 참고자료 | |
| | | | | | |

이 문서는 한국산업기술대학교 컴퓨터공학부의 “종합설계” 교과목에서
 프로젝트 “VR을 이용한 자세교정 시스템”을 수행하는
 S5-8:류호성, 양인모, 이승룡이 작성한 것으로 사용하기 위해서는 팀원들의
 허락이 필요합니다.

목 차

I. 서론

| | |
|------------------------|----|
| 1. 작품선정 배경 및 필요성 | 4p |
| 2. 기존 연구/기술동향 분석 | 5p |
| 3. 개발 목표 | 6p |
| 4. 팀 역할 분담 | 7p |
| 5. 개발 일정 | 7p |
| 6. 개발 환경 | 8p |

II. 본론

| | |
|------------------------|-----|
| 1. 개발 내용 | 9p |
| 2. 문제 및 해결방안 | 10p |
| 3. 시험시나리오 | 11p |
| 4. 상세설계 | 12p |
| 5. Prototype 구현 | 17p |
| 6. 시험 / 테스트 | 19p |
| 7. Coding & DEMO | 22p |

III. 결론

| | |
|-----------------------|-----|
| 1. 연구 결 | 25p |
| 2. 작품제작 소요재료 목록 | 26p |

| | |
|------------|-----|
| 참고자료 | 27p |
|------------|-----|

I. 서론

1. 작품선정 배경 및 필요성

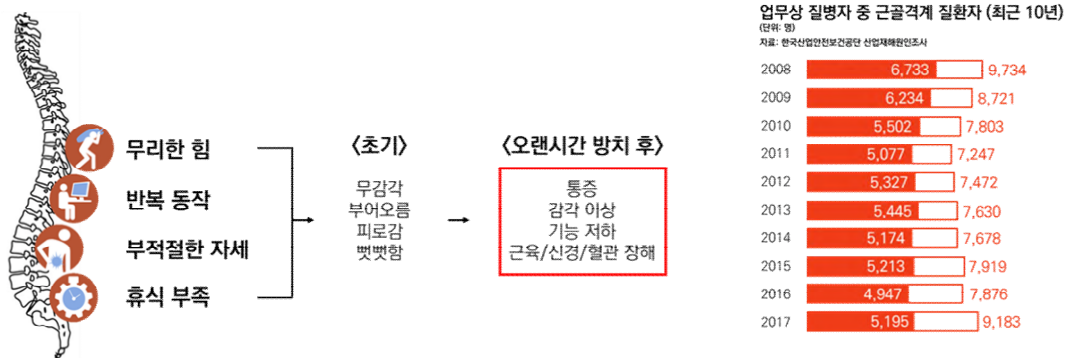
이전의 산업혁명으로 인하여 당시의 사회인들은 공장 또는 농장에서 장시간 똑같은 위치와 자세로의 노동을 강요받았다. 당시부터 급격하게 늘어난 퇴행성 관절 질환과 근골격계 질환들은 당시의 사람들을 괴롭혔다. 그렇다고 현대의 사회인들도 상황이 크게 다르지는 않다. 현대의 사회인들은 기존에 있던 시스템을 효율적이고 효과적으로 사용하기 위해서 시스템을 개선시키기 위한 노력을 지속적으로 하였으며, 그로인하여 정보혁명이 발생하였다. 정보혁명을 통해 수많은 사람들은 노동의 형태가 몸을 움직이는 것이 아닌 제자리에 앉아서 컴퓨터를 통해 사무를 처리하는 환경으로 변하게 되었다. 그리고 취미생활 역시 스마트폰을 통해 이루어지는 경우가 빈번해졌다.

이와 같은 사람들의 작업 매개체의 변화는 사람들에게 자세의 오랜 지속을 강요하였다. 이에 사람들은 훨씬 더 편하게 자세를 지속하고자 올바르게 앉은 자세를 취하게 되었고 그로인해 부적절한 자세를 오랜 시간 유지하는 습관이 많은 직업군의 사람들에게 자리 잡게 되었다. 오랜 시간 동안 부적절한 자세를 취한 사람들은 허리디스크, 목디스크 등과 같은 퇴행성 관절 질환 및 근골격계 질환이 발생하여 병원을 찾는 일이 증가하였다.

하지만 이러한 퇴행성 관절 질환 및 근골격계 질환은 평소의 간단한 스트레칭이나 자세교정만으로 충분히 예방이 가능하다고 한다.

그렇기에 현대 사회인들에게는 스트레칭이나 자세교정에 대하여 병원보다 간단하게 접근할 수 있는 방법이 필요하다.

이에 일반 가정집에서도 효과적으로 위와 같은 질환을 예방 및 치료하기 위한 방법이 필요해졌다.

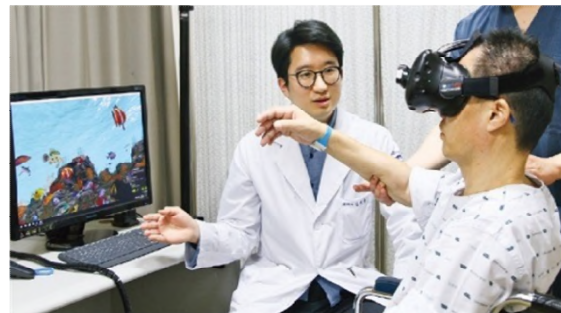
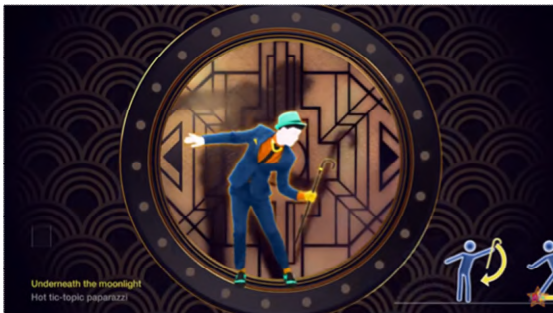


2. 기존 연구/기술동향 분석

사람의 신체를 분석하는 방법에는 크게 키넥트와 가상현실, 이 두 가지 방법이 있다. 키넥트는 카메라를 통하여 촬영된 이미지 중 영상처리 기술을 이용하여 신체의 외곽선과 관절의 추정위치를 토대로 분석하여 사용자의 모습을 유추할 수 있다. 가상현실은 컨트롤러와 트래커 등을 통하여 사용자는 디지털로 만들어진 세계 속의 화면을 보면서 컨트롤러와 트래커를 통해 얻은 좌표 값과 기울기 값을 통해 몸의 형태를 분석할 수 있다.

키넥트를 이용해 만들어진 시스템은 대표적으로 저스트 댄스가 있다. 저스트 댄스는 화면에 제시된 동작을 사용자가 따라하게 만들고, 사용자의 자세를 키넥트를 통해 인식하여 자세를 잘 따라했는지 확인시켜주며 자연스럽게 운동을 하는 효과를 가지고 있는 게임이다.

가상현실은 의료기관에서도 시범을 보인 사례가 있다. 분당차병원은 2018년부터 환자들의 재활치료에 가상현실을 도입하였다. 환자는 화면에 보이는 물고기를 잡는 행동을 반복하며 재활치료를 수행하며, 재활치료가 꾸준히 진행됨에 따라 손의 운동기능이 향상되어 일반인과 같은 움직임을 취할 수 있게 되었다.



3. 개발 목표

가상현실 기기를 통해 실제로 눈앞에 존재하는 것처럼 만들어 생동감을 높이고 사진과 동영상과 같이 뒷모습, 옆모습처럼 표면적으로는 확인할 수 없었던 스트레칭 자세의 세세한 부분도 사용자가 원하는 방향과 각도에서 직접 확인할 수 있도록 하여 자세의 이해도를 증진시킨다.

또한 가상현실의 이점인 흥미를 이끌어 기존의 심심하고 흥미가 없어 실제로 아프기 전까지는 필요성을 느끼지 못하는 자세교정이나 스트레칭을 더욱 사용자가 쉽게 접근할 수 있게 한다.

그리고 일반적인 사람들이 전문가의 특별한 도움 없이 가택 내에서 누구나 쉽게 스트레칭을 수행하며, 자신이 현재 가지고 있는 근골격계 질환을 파악할 수 있도록 한다. 또한 발병한 질환을 토대로 스트레칭을 수행할 자세를 추천하여 스트레칭 효과가 극대화 할 수 있도록 한다.



4. 팀 역할 분담

| vr 자세교정 프로그램 | | | |
|--------------|----------------|-----------|--------------|
| 작업 | 가이드 및 애니메이션 제작 | 유니티 UI 제작 | 유니티 로직 프로그래밍 |
| 담당자 | 양인모 | 류호성 | 이승룡 |

| 서버 | | | |
|-----|-----------|-----------|--------------|
| 작업 | 데이터베이스 설계 | AWS 서버 구축 | 데이터 분석 프로그래밍 |
| 담당자 | 류호성 | 양인모 | 이승룡 |

5. 개발 일정

| 개발일정 | |
|-----------------|---------------------------|
| 1월 05일 ~ 1월 10일 | VR 센서 데이터 확인 |
| 1월 11일 ~ 1월 14일 | AWS 환경 테스트 |
| 2월 15일 ~ 2월 22일 | 가이드 NPC 리깅 및 애니메이션 적용 테스트 |
| 2월 23일 ~ 2월 29일 | 데이터베이스 스키마 구성 |
| 3월 1일 ~ 3월 9일 | 사용자 - VR 아바타 연동 |
| 3월 10일 ~ 3월 14일 | 가이드 NPC - 사용자의 모션 오차 분석 |
| 3월 15일 ~ 3월 22일 | 프로토타입 애니메이션 제작 |
| 4월 02일 ~ 4월 10일 | 프로토타입 진척도 분석 |
| 4월 18일 ~ 4월 27일 | AWS 서버 구축 |
| 5월 11일 ~ 5월 23일 | 유니티 - AWS 동기화 |
| 6월 01일 ~ 6월 12일 | AWS - 데이터베이스 동기화 |
| 6월 25일 ~ 7월 27일 | 모든 스트레칭 애니메이션 제작 |
| 7월 04일 ~ 8월 03일 | 모든 스트레칭 누적 데이터 분석 |
| 7월 15일 ~ 8월 17일 | AWS 적재 및 실행 |

6. 개발 환경

| VR 제품(HTC VIVE) 사양 | |
|--------------------|---|
| 스크린 | Dual amoled 3.6 diagonal |
| 해상도 | 1080 * 1200 per eye |
| 시야각 | 110 degrees |
| 센서 | Steam VR tracking, G-Sensor, Gyroscope, 근접 센서 |

| 컴퓨터 최소 사양 | |
|-----------|-----------------------------|
| GPU | NVIDIA GeForce GTX970 |
| CPU | Intel Core i5-4590 |
| Memory | 4GB |
| 비디오 출력 | HDMI 1.4 또는 DisplayPort 1.2 |
| 운영체제 | Windows 10 |

| 제작 툴 | |
|------------|-----------|
| 프레임워크 | 유니티 |
| 프레임워크 언어 | C# |
| 캐릭터 생성 | makeHuman |
| 애니메이션 | 블렌더 |
| 데이터베이스 | mySQL |
| 서버 | AWS |
| 서버 프로그램 언어 | JAVA |

II. 본론

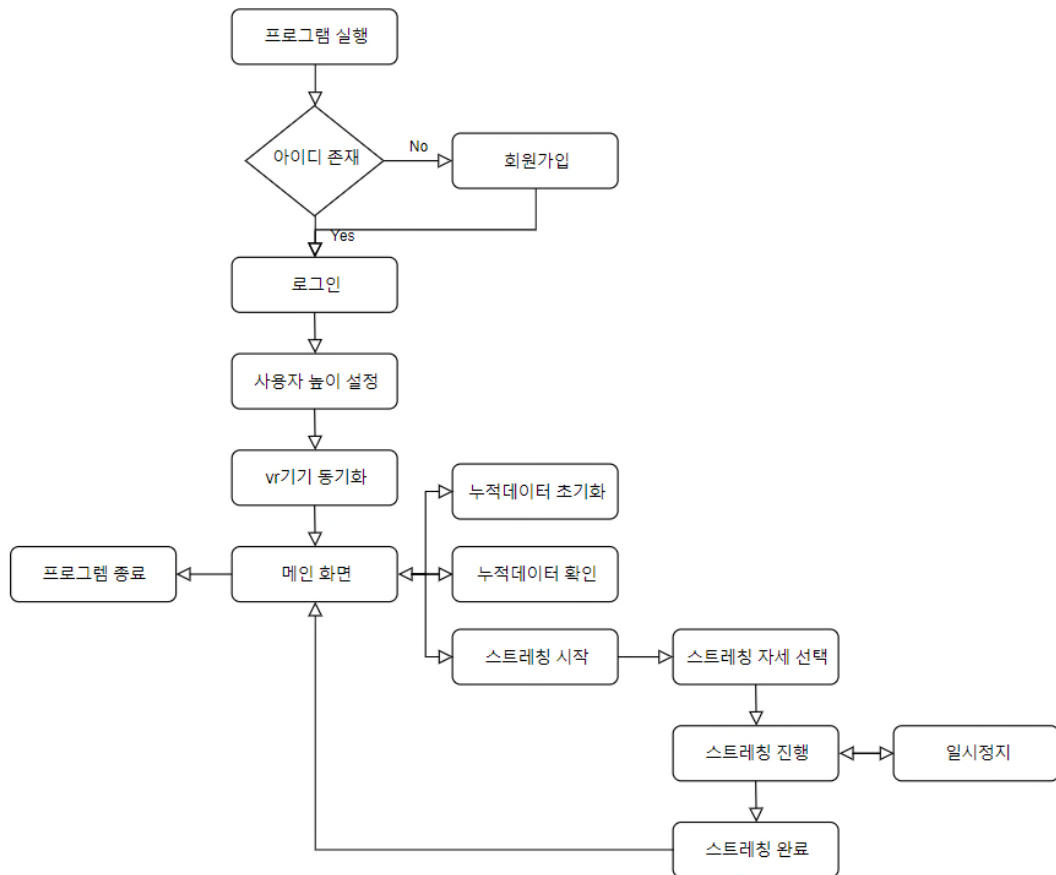
1. 개발 내용

- 사용자의 트래커와 애플리케이션 트래커의 동기화 개발
 - 애플리케이션의 의도대로 부위 인식을 수행한다.
- vr기기 화면에 표시할 가이드 제작
 - 스트레칭 자세의 움직임을 표현할 3D 모델을 틀을 통해 제작한다.
 - 3D 모델의 머리, 팔, 허리, 다리의 좌표 값을 움직일 때 마다 가져올 수 있어야한다.
- 사용자의 움직임을 vr기기로 인식해 좌표 값으로 저장할 수 있는 코드 개발
 - 사용자의 움직임을 머리, 팔, 허리, 다리에 부착된 센서의 좌표 값으로 가져온다.
 - 시간별 좌표 값을 저장하고 사용자의 움직임을 인식한다.
- 사용자의 움직임과 가이드의 움직임을 비교하고 화면에 표시하는 코드 개발
 - 가이드의 좌표 값과 사용자의 좌표 값을 비교해 각 부위별 오차를 인식한다.
 - 오차가 적을 경우와 오차가 클 경우를 인식하고 화면에 표시한다.
- 근골격계 질환 분석 개발
 - 정상적인 사람과 비교하였을 때 각 센서의 데이터 값을 토대로 사용자의 근골격계 질환을 파악할 수 있어야한다.
- 누적 데이터 분석 개발
 - 사용자가 수행한 스트레칭의 횟수를 알고 각 회차별 오차율을 표시한다.
 - 사용자가 가지고 있는 근골격계 질환에 대한 교정 진척도를 표시한다.

2. 문제 및 해결방안

- VR 기기의 인식부위가 적어 사용자가 취하는 특정자세의 인식이 잘 되지 않는다.
 - ☞ 같은 스트레칭 효과가 있는 인식이 원활한 자세로 수정한다.
- 센서의 오차가 발생하기 때문에 사용자가 움직이지 않더라도 센서 값에 변화가 생긴다.
 - ☞ 데이터를 일정 구간으로 나누고 평균값을 취해 오차의 영향을 최소화 한다.
- 트래커의 부착 순서에 따라 트래커의 부위 인식이 다르게 된다.
 - ☞ 헤드기어를 기준으로 z축(높이)값과 x축(좌우)값을 비교하여 트래커 정보를 조정한다.
- 스트레칭이 제대로 되지 않거나 VR기기를 착용한 채 자세를 취하기가 힘들다.
 - ☞ 스트레칭을 직접 수행해보고 자세를 수정한다.

3. 시험시나리오



- 사용자 높이 설정 : 프로그램 내부에서 사용자의 신장에 맞게 화면을 출력하도록 높이 설정.
- vr기기 동기화 : 각각의 vr기기가 신체 어느 부위에 부착되어있는지 동기화.
- 누적데이터 초기화 : 사용자의 데이터를 삭제 후 초기화.
- 누적데이터 확인 : 사용자의 지난 스트레칭 기록 정보를 확인.
- 스트레칭 진행 :
 1. 사용자가 수행해야할 자세를 vr화면의 가이드NPC를 통해 화면에 출력.
 2. 사용자는 vr기기를 착용하고 화면에 동작을 취하는 NPC의 동작을 따라함.
 3. vr기기의 좌표 값을 통해 사용자의 자세를 인식.
 4. 사용자와 NPC의 좌표 값을 비교해 오차를 측정.
 5. 자세가 일치할시 초록색, 차이가 심할 경우 빨간색으로 표시.
 6. 스트레칭 수행정보를 누적데이터로 저장.

4. 상세 설계

A. 테이블 설계

- 기준 데이터 테이블 : 각 애니메이션 클립별 VR기기의 기준 위치 좌표와 회전 좌표를 등록한다.

| Standard | | | | | | | |
|-------------|-------------|--------|--------|--------|--------|--------|--------|
| ani_name | parts | posX | posY | posZ | rotX | rotY | rotZ |
| varchar(40) | varchar(10) | double | double | double | double | double | double |

- 오차 시간 테이블 : 사용자가 자세 교정을 수행할 때 발생한 부위별 누적 오차시간을 저장한다.

| ErrorTime | | | | | | | |
|-----------|-------------|--------|--------|--------|--------|--------|--------|
| time | ani_name | posX | posY | posZ | rotX | rotY | rotZ |
| datetime | varchar(40) | double | double | double | double | double | double |

- 클립 테이블 : 자세 교정 항목에 존재하는 애니메이션 리스트와 각 애니메이션 별 임계값을 저장한다.

| ClipInfo | | |
|-------------|------------|-----------|
| ani_name | category | threshold |
| varchar(40) | varchar(8) | double |

B. 프로시저 설계

| 이름 | 대상 | 입력 | 출력 | 설명 |
|---------------------------|----------------------|-------------------------------|------------------------|--------------------|
| usp_add_Member | TB_Member | Id, Password | 0(실패) 또는 1(성공) | 회원가입을 위한 프로시저 |
| usp_login_Member | TB_Member | Id, Password | 0(실패) 또는 MemSeq(성공) | 로그인을 위한 프로시저 |
| usp_get_ PosCorrection | TB_ PosCorrection | 없음 | 모든pos/rot 필드 | 자세교정을 위한 좌표를 로드 |
| usp_get_Record | TB_Record | MemSeq | Data (교정기록 XML) | 회원의 교정기록을 출력 |
| usp_update_Record | TB_Record | MemSeq, Data (교정기록 XML) | 0(실패) 또는 1(성공) | 회원의 교정기록을 업데이트 |

C. 자세교정 설계

● 비교 데이터 취득

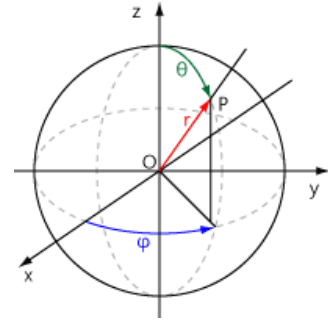
- ※ 스테이지가 시작되면 서버에 저장되어있는 자세교정의 기준값을 수신.
- ※ 사용자는 가이드 애니메이션을 따라하며 자세교정을 수행.
- ※ 사용자의 수행 데이터는 0.05초마다 1회 측정하며 그 값을 리스트에 저장
- ※ 수행 데이터의 에러를 최소화하기 위하여 최신 10개의 수행 값의 평균을 구하여 기준값과 비교할 샘플 값을 생성.
- ※ 샘플 값은 x, y, z로 구성된 좌표데이터와 사원수로 구성된 기울기 데이터를 얻지만 기울기 데이터는 x, y, z 성분 3개를 이용하여 계산.

● 좌표데이터 분석

- ※ 좌표데이터는 두 점 사이의 거리를 구하는 피타고라스의 정리를 이용하여 오차 거리를 계산.
- ※ $P_d = \sqrt{(P_{s_x} - P_{c_x})^2 + (P_{s_y} - P_{c_y})^2 + (P_{s_z} - P_{c_z})^2}$
- ※ P_s : 기준좌표값, P_c : 수행좌표값, P_d : 좌표오차값

● 기울기 데이터 분석

- ※ 각각의 기울기 데이터를 기울기 벡터 차원으로 변형.
- ※ 두 기울기 벡터의 내각의 차이를 벡터의 내적을 통해 계산.



- ※ $R_s R_{c_x} + R_{s_y} R_{c_y} + R_{s_z} R_{c_z} = \sqrt{R_{s_x}^2 + R_{s_y}^2 + R_{s_z}^2} \sqrt{R_{c_x}^2 + R_{c_y}^2 + R_{c_z}^2} \cos R_d$
- ※ $R_d = \cos^{-1}((R_s R_{c_x} + R_{s_y} R_{c_y} + R_{s_z} R_{c_z}) / (\sqrt{R_{s_x}^2 + R_{s_y}^2 + R_{s_z}^2} \sqrt{R_{c_x}^2 + R_{c_y}^2 + R_{c_z}^2}))$
- ※ R_s : 기준기울기값, R_c : 수행 기울기값, R_d : 기울기 오차값

● 오차율 분석

- ※ 분석한 좌표데이터와 기울기 데이터를 토대로 사용자의 수행 오차율을 분석.
- ※ 사용자가 완벽하게 데이터를 따라할 수 없는 점을 고려하여 가우시안 분포의 형태로 오차율을 계산하고 또한, 좌표데이터가 기울기 데이터보다 우선적이기 때문에 가중치를 더 많이 적용.

$$※ E = \frac{4P_d + R_d}{5}$$

D. 아바타 설계

● 사용자 아바타

- ※ MakeHuman 도구를 이용하여 사용자 아바타의 텍스처 및 리깅을 위한 뼈대를 생성.
- ※ Unity에서 아바타에 Animation Rigging, Rig Builder, Bone Renderer 스크립트를 작성하여 IK 형식으로 아바타의 모션을 제작.
- ※ VR 컨트롤러 오브젝트에 VR Rig 스크립트를 생성후 Rig Builder에 연결하여 VR 컨트롤러와 아바타의 움직임을 동기화.

● 가이드 아바타

- ※ MakeHuman 도구를 이용하여 가이드 아바타의 텍스처 및 리깅을 위한 뼈대를 생성.
- ※ Blender를 통해 가이드 아바타의 애니메이션 fbx 파일 생성.
- ※ Unity 화면에 가이드 아바타 애니메이션 출력.

E. 유저 인터페이스 설계

● 애플리케이션 실행 시

- ※ 사용자는 화면 전방에 있는 거울을 통해 자신의 움직임과 아바타의 연동 정도를 확인할 수 있다.
- ※ 사용자는 왼쪽 컨트롤러의 패드를 눌러 메뉴를 호출할 수 있으며, 메뉴가 호출된 경우 오른쪽 컨트롤러의 트리거를 통해 항목을 선택할 수 있다.

● 애플리케이션 메뉴 호출

- ※ 사용자는 옵션 메뉴, 자세 교정 메뉴, 기록 메뉴를 선택할 수 있다.
- ※ 옵션 메뉴는 사용자 키에 맞게 아바타의 크기를 조절하는 부분, 베이스 스테이션 위치에 따른 가상공간 위치 조절 부분이 있다.
- ※ 자세 교정 메뉴는 수행할 난이도와 부위를 선택하여 자세 교정을 수행할 내용을 선택할 수 있다.
- ※ 기록 메뉴는 현재까지 수행한 자세 교정을 종합적으로, 항목별로, 날짜별로 분류하여 결과를 확인할 수 있다.

● 애플리케이션 자세 수행

- ※ 전방에 있는 가이드 아바타를 통해 수행할 동작을 확인할 수 있다.
- ※ 자세 수행이 시작되면 사용자 컨트롤러에 대응되는 교정 구체가 생성된다.
- ※ 교정 구체는 오차 데이터가 임계값을 넘으면 초록색이되고, 넘지 못하면 주황색, 빨간색이 되어 에러의 정도를 눈으로 확인할 수 있다.

F. 서버 설계

● 클라이언트 - 서버

- ※ 각 자세교정 스테이지 시작 시 클라이언트로 기준 좌표 값, 기준 기울기 값 전달.
- ※ 각 자세교정 스테이지를 진행할 때마다 클라이언트로부터 사용자의 수행 데이터값 중 좌표 값, 기울기 값을 전달받음.
- ※ 좌표 값, 기울기 값을 토대로 사용자의 근골격계 질환을 분석.
- ※ 예시) 거북목 증후군 - 서 있는 자세에서의 자세교정을 수행하면서 취득한 헤드기어의 좌표가 허리에 부착된 트래커의 좌표와 신발에 부착된 트래커의 좌표 사이에서 신발에 부착된 트래커와의 가까운 정도에 따라 거북목 증후군을 분석.
- ※ 자세교정이 모두 진행된 후 클라이언트에게 분석한 근골격계 질환의 정도를 송신.

● 서버 - 데이터베이스

- ※ 각 자세교정 스테이지 시작 시 데이터베이스로부터 기준 좌표 값, 기준 기울기 값 수신.
- ※ 자세교정 스테이지 진행 시마다 좌표 오차 값, 기울기 오차 값, 오차율을 데이터베이스로 송신
- ※ 근골격계 질환 분석 후 분석 정보를 xml 형식으로 데이터베이스로 송신

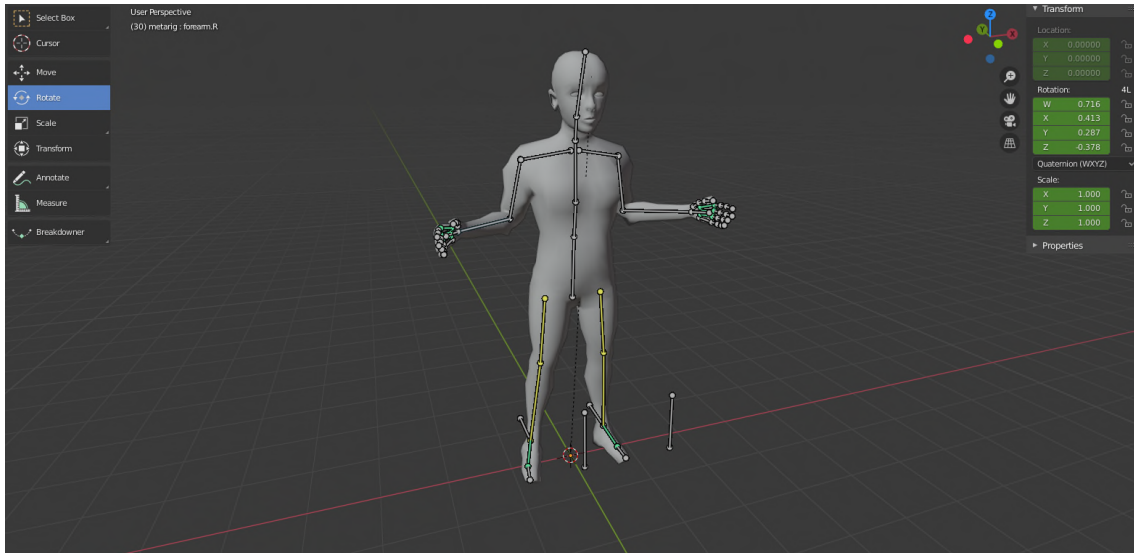
● AWS

- ※ AWS내에서 EC2를 리눅스로 설정하고 생성.
- ※ Putty로 리눅스 서버에 접속해 mysql을 설치.
- ※ 클라이언트에서 xml 형식으로 데이터를 받아 누적데이터 생성.
- ※ 누적데이터를 클라이언트가 요청하면 송신.

5. Prototype 구현

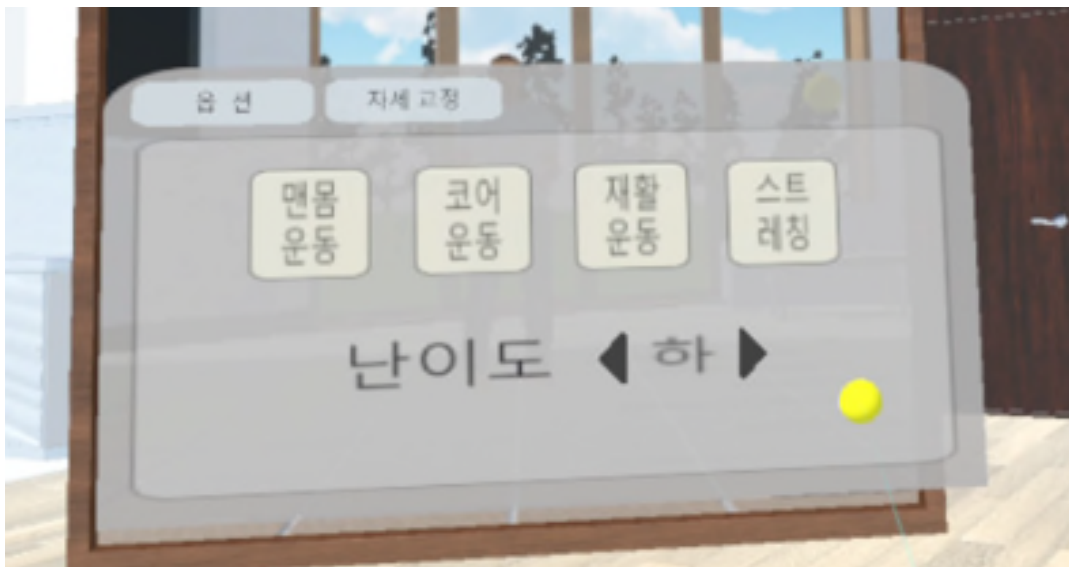
● 가이드 애니메이션 구현

※ 사용자가 따라 할 자세의 가이드 애니메이션 구현



● UI 설계 및 구현

※ 사용자가 원하는 자세를 수행할 수 있는 UI 구성



◎ 사용자 VR 기기와 가이드 애니메이션의 좌푯값 비교

※ 사용자 VR 기기와 가이드 애니메이션 좌표를 기록한 파일을 비교해 잘못된 부분에 색으로 표시

※ 좌푯값은 어느 시간에 어느 위치에 있어야 하는지 기록한 파일을 통해 비교



| ===== | | | | |
|-------------------------|-------------|-----------|------------|------------|
| Head Mounted Display(s) | | | | |
| ===== | | | | |
| deviceID | timestamp | posX | posY | posZ |
| 0 | 0.000000000 | 44.554699 | 165.519257 | -28.368557 |
| 0 | 0.016666667 | 44.554699 | 165.519257 | -28.368557 |
| 0 | 0.033333333 | 44.542263 | 165.519257 | -28.368120 |
| 0 | 0.050000000 | 44.532211 | 165.519257 | -28.367342 |
| 0 | 0.066666667 | 44.520748 | 165.519257 | -28.366489 |
| 0 | 0.083333333 | 44.509800 | 165.519257 | -28.365643 |
| 0 | 0.100000000 | 44.501934 | 165.516357 | -28.364887 |
| 0 | 0.116666667 | 44.500092 | 165.513138 | -28.364305 |
| 0 | 0.133333333 | 44.498745 | 165.509674 | -28.363979 |
| 0 | 0.150000000 | 44.497658 | 165.505539 | -28.363991 |
| 0 | 0.166666667 | 44.496716 | 165.503296 | -28.365965 |
| 0 | 0.183333333 | 44.495800 | 165.502518 | -28.370064 |
| 0 | 0.200000000 | 44.494793 | 165.502182 | -28.374722 |
| 0 | 0.216666667 | 44.500267 | 165.502106 | -28.380264 |
| 0 | 0.233333333 | 44.507130 | 165.502045 | -28.386303 |
| 0 | 0.250000000 | 44.514240 | 165.502640 | -28.396009 |
| 0 | 0.266666667 | 44.516182 | 165.514374 | -28.405008 |
| 0 | 0.283333333 | 44.517231 | 165.521103 | -28.413616 |
| 0 | 0.300000000 | 44.519829 | 165.528046 | -28.422813 |
| 0 | 0.316666667 | 44.528816 | 165.528900 | -28.438292 |
| 0 | 0.333333333 | 44.534187 | 165.529495 | -28.455259 |
| 0 | 0.350000000 | 44.542343 | 165.529892 | -28.475729 |

◎ Prototype 시나리오

1. 사용자는 VR 프로그램을 실행시킨다.
2. 사용자가 교정을 원하는 자세를 선택한다.
3. 가이드 애니메이션이 교정 자세를 보여준다.
4. 사용자가 가이드 애니메이션을 따라 하면 사용자의 VR 기기와 자세의 좌푯값을 비교해 많이 다른 부분을 붉은색으로 표시해준다.
5. 사용자는 잘못된 자세를 인지하고 자세를 바로 하여 올바르게 자세교정을 하게 된다.

6. 시험 / 테스트 결과

프로젝트가 잘 동작하는지 확인하기 위해 먼저 현실에서 실제로 사용자가 움직이는 것과 VR 프로그램 내부에서 아바타가 움직이는 화면을 비교하였다.

실제 수행과 프로젝트 내부 비교



실제 사람이 수행하는 화면



VR에서 사용자가 움직이는 모습

실제 수행과 프로젝트 내부 비교



실제 사람이 수행하는 화면



VR에서 사용자가 움직이는 모습

비교 화면을 통해 알 수 있듯이 사용자에게 움직임과 연동되어 아바타의 움직임이 잘 이루어지지만 한계점을 가지고 있었다. 트래커와 컨트롤러로 위치를 추적할 수 없는 부분에 대해서 아바타는 완전하게 연동하지 못하였고, 손을 비롯한 나머지 부분은 고정된 채 팔꿈치를 움직였을 때 VR속 아바타는 이를 제대로 구현하지 못하였다.

다음은 사용자가 가이드를 통해 동작을 따라하였을 때 정확도는 어느정도이며 오차를 분석할 수 있는지 테스트하였다.

동작 수행과 정확도 비교



정확히 동작을 따라한 화면



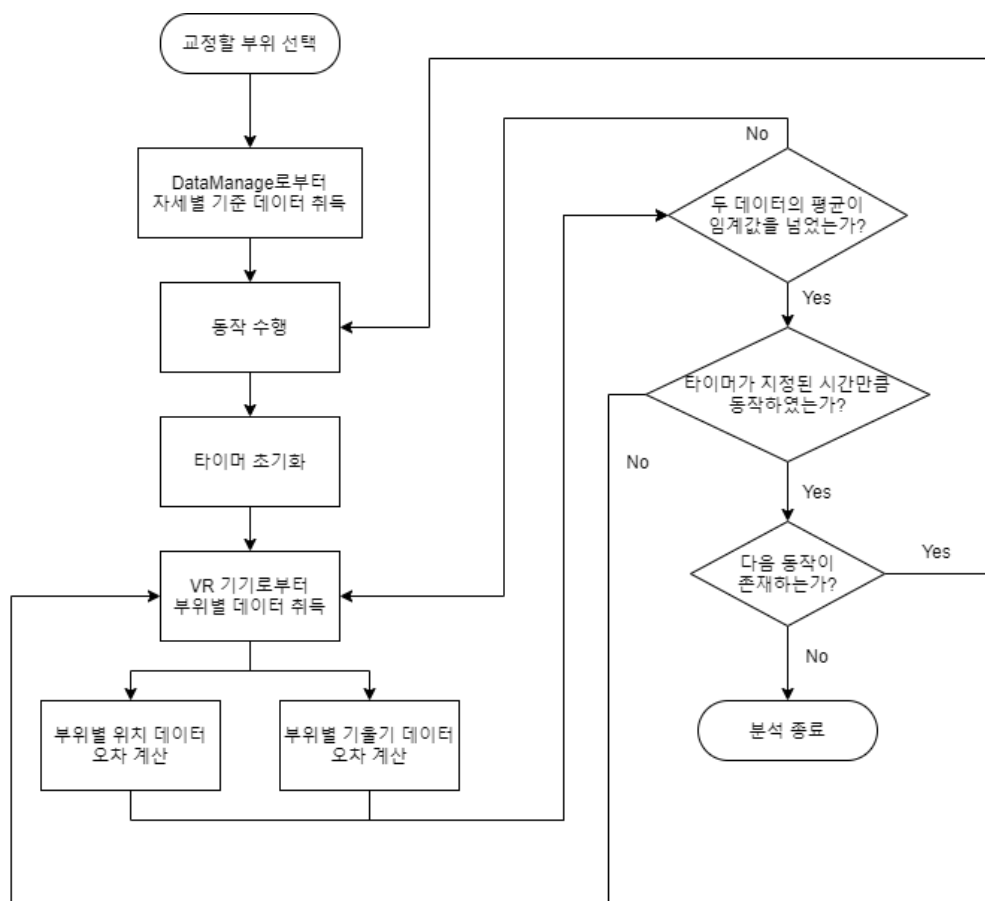
동작을 따라했을 때 정확도

사용자는 자세를 따라하면서 정확도를 알려주는 구체를 통해 실시간으로 어느 부위가 틀렸는지 알 수 있으며 이를 수정할 수 있었다. 또한 정확히 따라했을 경우와 제대로 따라하지 못했을 경우의 분석도 정상적으로 수행되고 있었다. 하지만 개선할점으로는 사용자는 자세교정을 수행하면서 틀린 부위는 알 수 있지만 어느 방향으로 어느 정도 틀렸는지 명확히 알기 힘들기에 방향과 강도를 알 수 있는 보완된 시스템이 필요하다.

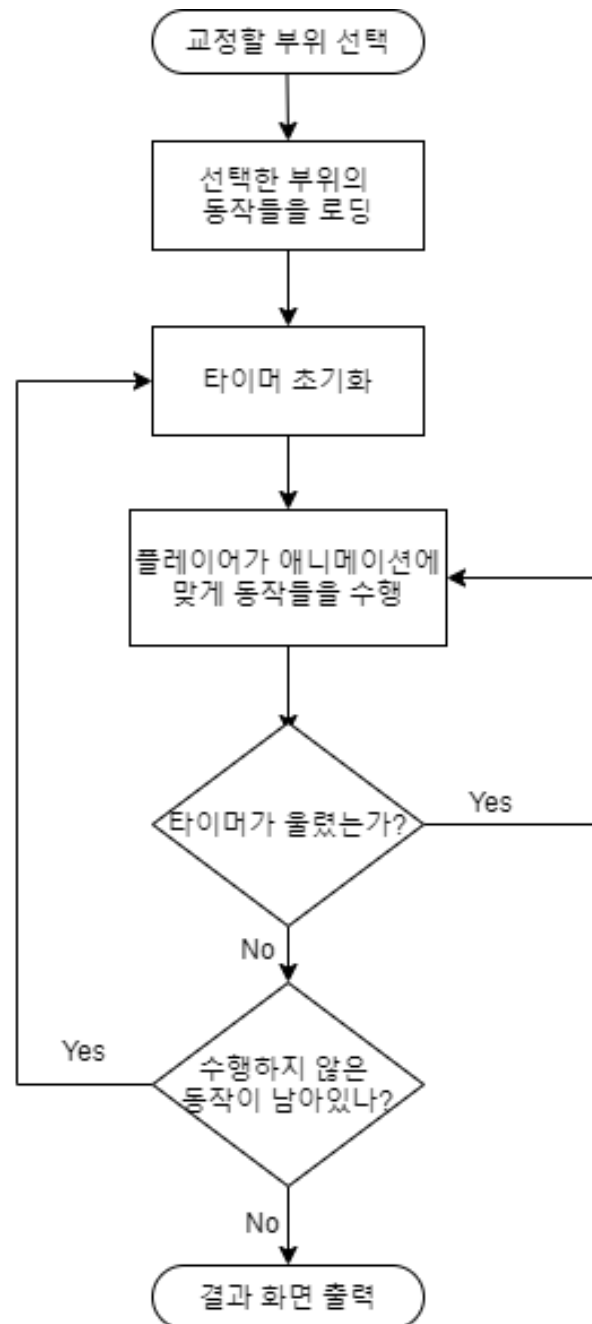
7. Coding & DEMO

프로젝트 내부에 사용된 코드로는 크게 사용자의 움직임을 분석하는 부분, 가이드 애니메이션을 동작시키는 부분, 서버와 데이터가 연동된 부분으로 구성되어있다.

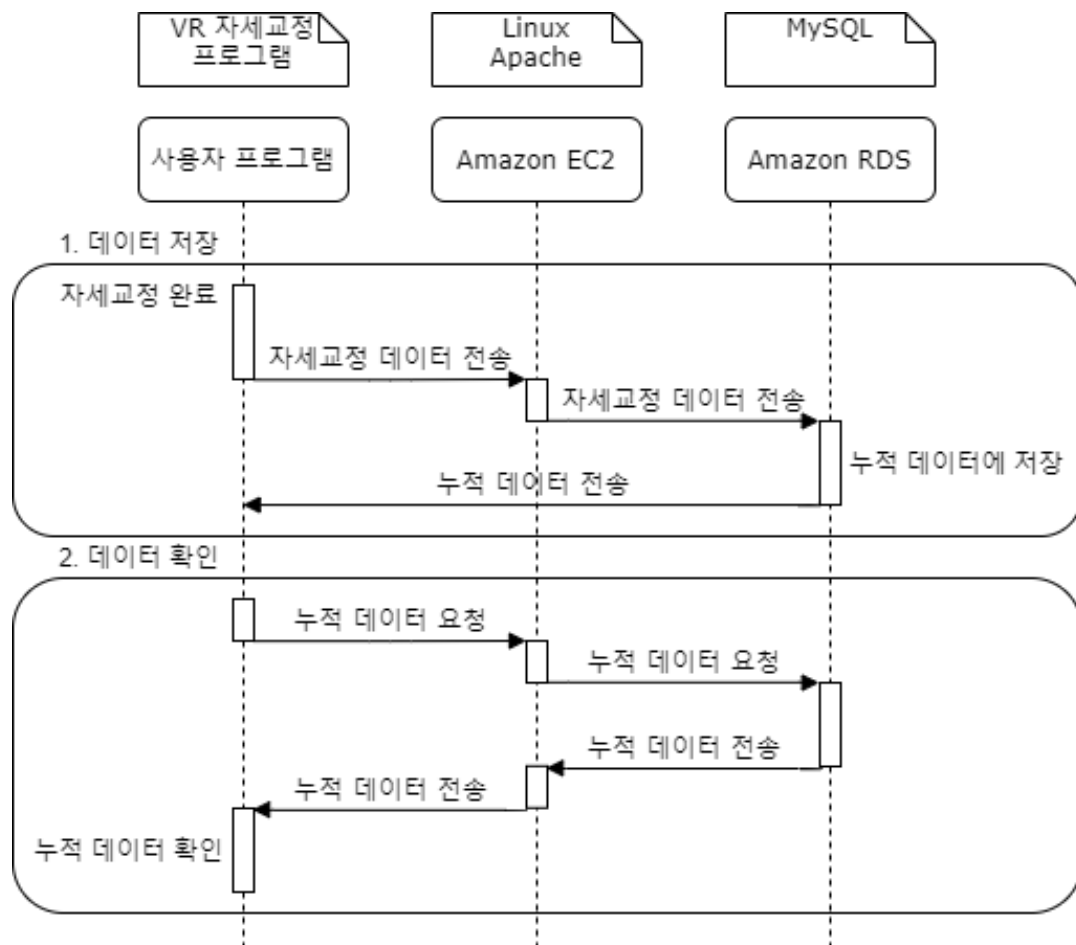
사용자의 움직임 분석은 현재 수행 중인 동작의 기준 데이터와 사용자가 수행된 데이터의 차이점을 분석하고 있다. 사용자의 데이터는 허리 트래커를 기준으로 계산되어 사용자의 절대 위치가 달라져도 움직임을 정상적으로 계산할 수 있다. 위치 데이터와 기울기 데이터는 4 : 1의 비율로 평균을 내어 임계값 확인에 사용중이다.



가이드 애니메이션은 수행해야할 자세가 많이 존재하고 애플리케이션 규모가 커졌을 때도 쉽게 확장할 수 있도록 프레임워크에서 애니메이션 오브젝트만 등록하면 확장될 수 있도록 구성하였다. 또한 사용자 분석을 통해 얻은 정확도를 기준으로 다음 동작으로 전환하는 제어구문을 가지고 있다.



자세 교정을 수행하는 동안 발생한 사용자 좌표와 기준 좌표의 오차는 Apache 서버로 전송되며 서버에 존재하는 데이터베이스를 통해 이를 관리한다. 또한 각 자세 수행이 끝나거나 메뉴에서 이전 기록을 확인할 때 서버에 존재하는 데이터를 불러와 사용자가 그동안 수행했던 자세수행 데이터를 확인할 수 있도록한다.



Ⅲ. 결론

1. 연구 결과

본 VR을 이용한 자세교정 시스템은 장시간으로 잘못된 자세를 오래 유지하여 근골격계 질환이 발생한 현대인들에 대해 병원에 방문하지 않고 일반 가정집에서 근골격계 질환을 치료할 수 있도록 개발되었다. 자세교정 수행을 통해 근골격계 질환을 예방 및 치료하는 부분은 효과가 있었으나, VR 장비 착용으로 인해 수행하기 힘든 동작이 확인되었다.

사람들이 흔히 하는 대표적인 운동으로 스쿼트, 런지, 플랭크 등이 있고 본 시스템에서 수행하는 운동으로 올인원, 쉐골 압수 등이 있다. 스쿼트, 런지, 올인원과 같은 정면을 보면서 수행하는 자세교정의 예방 및 치료 효과는 기대할 수 있는 수준으로 판측됐다. 움직임이 활발하고 균형유지를 동반해야 하는 동작의 정확도는 처음 수행했을 경우 낮은 수준에 머물렀지만, 자세교정을 반복적으로 수행하면서 높은 정확도로 수행할 수 있게 되었다. 하지만 플랭크, 쉐골 압수 등 정면만을 바라보지 않고 수행하는 자세교정에 대해서는 의미 있는 정확도의 향상이 이루어지지 않았다. 정면을 보지 않았을 경우 컨트롤러의 진동을 통해 다음 동작의 진행을 알 수 있었지만, 거울을 통해 자세교정의 정확도를 파악하는 본 시스템의 특성상 거울을 볼 수 없는 경우 자세교정이 잘 못 되어도 사용자 스스로 올바른 방법으로 되돌아오지 못했다.

사용자가 정면을 바라보지 못해 잘못된 자세교정이 수행될 경우를 고려하여 시스템은 개선되어야 할 부분이 있다. 가장 효과적이고 이해가 쉬운 방법으로 사용자가 보는 화면에 시점 고정형 UI를 통해 시점이 변하더라도 정확도를 확인할 수 있는 화면을 제공하는 것이다. 고정형 UI를 제공할 경우에도 주의할 점이 있다. 고정형 UI는 전역 화면을 통해 카메라의 위치가 변해도 동일한 화면을 제공하는 장점을 가지고 있다. 하지만 VR 시스템의 경우 카메라 2대를 이용하여 각각의 화면이 제공하는 오프셋을 조금 달리하여 입체적으로 보이는 특징이 있다. 따라서 전역 화면을 통해 화면이 일그러지는 현상이 나타날 수 있다. 이러한 부분을 예방하기 위해 시스템 내 오브젝트로 생성하여 사용자의 시선을 따라가도록 만들면 자세교정 수행 정확도를 향상 시킬 수 있을 것으로 예상된다.

2. 작품제작 소요재료 목록

HW



VIVE 헤드셋
사용자 화면 제공



VIVE 컨트롤러
사용자 상호작용



VIVE 트래커
사용자 위치 추적



베이스 스테이션
사용자 위치 추적



링크박스
하드웨어 인터페이스



USB 포트
USB 증설



허리 복대
트래커 부착(허리)



슬리퍼
트래커 부착(발)

SW



MakeHuman
아바타 제작



Blender
아바타
애니메이션 제작



Final IK
VR-사용자 동기화



Living Room
Interior Pack
시스템
인테리어 구성



Youtube DL
Video Player
시스템 내부
Youtube 실행



Adobe
PremierePro
동영상 편집



곰믹스 프로
동영상 편집

참고자료

분당 차병원 물리치료

http://health.chosun.com/site/data/html_dir/2018/01/24/2018012403237.html

AWS - <https://aws.amazon.com/ko/>

절대강좌! 유니티 VR/AR - 이재현/위키북스

유니티 애니메이션 에센셜 - 앨런 쏜/에이콘

유니티VR 프로그래밍 - 광고경기문화창조허브

캐릭터 디자인 - <http://www.makehumancommunity.org/>

SteamVR 포럼 - <https://steamcommunity.com/app/250820/discussions>