

키넥트를 이용한 홈 트레이닝 🏋️

졸업작품 제안서

2016152025 양수연 | 2015172039 정유하 | 2012181003 김두환



목차 |

GRADUATION
PORTFOLIO

0. 지적 사항

1. 종합 설계 개요

2. 관련 연구 및 사례

3. 시스템 수행 시나리오

4. 시스템 구성도

5. 시스템 모듈 상세 설계

6. 개발 내용 및 개발 환경

7. 업무 분담

8. 종합 설계 수행 일정

9. 필요 기술 및 참고문헌



종합 설계 개요:

지적 사항

	지난 발표에서의 지적사항	지적 사항에 대한 답변	관련 페이지
1	일치도에 대한 연구가 더 필요함	기존의 x, y좌표로 골격을 인식하는 것이 아니라 z좌표까지 추가하여 정확도를 높임	p.26
2	운동 시 지연이 일어났을 경우 넘어갈 수 있는 수준의 일치도 경계 값 추출	레퍼런스 영상을 바로 따라할 수 있게 실시간 영상 레이어 위에 반투명 레퍼런스 영상을 재생 함으로써 동작 지연을 방지함.	P.35
3	난이도 상중하에 따른 일치도 추출	일치율 오차범위에 차이를 둠	P.36
4	의미 있는 일치율을 뽑아야 함. (데이터 정제)	총 10개의 골격 사이의 각도 값을 구하고 데이터를 정제하여 이해할 수 있는 일치율을 보여줌	p.37
5	다양한 운동 동작 구현	신체 골격 각도를 제대로 활용 할 수 있는 운동을 선별함	

종합 설계 개요:

개발 배경

성인남녀 57%, 집에서 운동하는 '홈트족'

* 성인남녀 831명 대상 설문조사 결과, 자료제공 : 잡코리아X알바몬



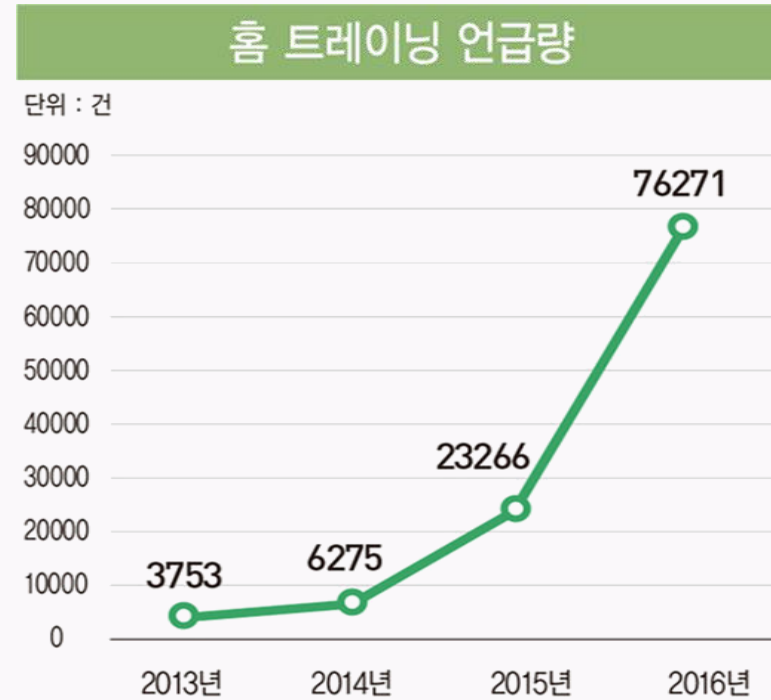
Q. 홈트족 생활을 하는 이유는 무엇인가요?



*홈트족 생활을 한다고 답한 응답자 대상, 복수응답

종합 설계 개요:

개발 배경



자료 : 다음소프트

✓ 홈트레이닝 장점

- 기존 PT에 비해 금전적 절약
- 시간과 공간의 제약에 자유로움
- 자신의 체력을 고려하여 계획한
- 프로그램 일정 대로 운동 가능

종합 설계 개요: 개발 배경

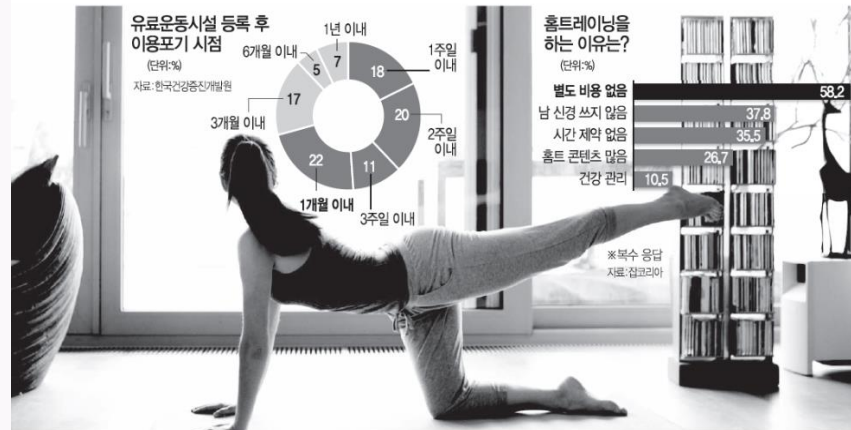
- 올바른 자세를 코칭 해줄 전문가 부재
- 잘못된 자세로 운동하여 부상의 위험 증가

→ 자세분석 기술의 필요성

유튜브 보며 '푸시업' 하다 빠곳... 홈트, 쉬운게 아니더라

2018-09-05 (수) 김광수 기자

불특정 다수 대상으로 한 홈트 영상, 체형·근력 등 신체조건 고려 않고
바른자세 잡기 어려워 부상위험 커, 회전근개파열 환자 6년새 89% 급증



잘못된 자세에서 실시하는 운동은 약이 아니라 독이다

이슬 기자 dew0514@hankooki.com

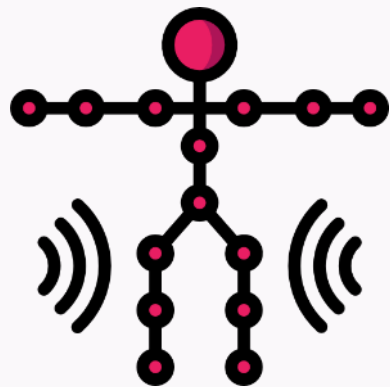
잘못된 홈트레이닝, 신체불균형과 같은 문제 유발 가능성 있어

오현지 기자 2018-08-08 09:21



종합 설계 개요:

개발 목표



- 키넥트의 **모션 인식 기능**을 활용하여
- 운동 자세 분석 및 교정을 위한
- **홈 트레이닝** 프로그램 개발

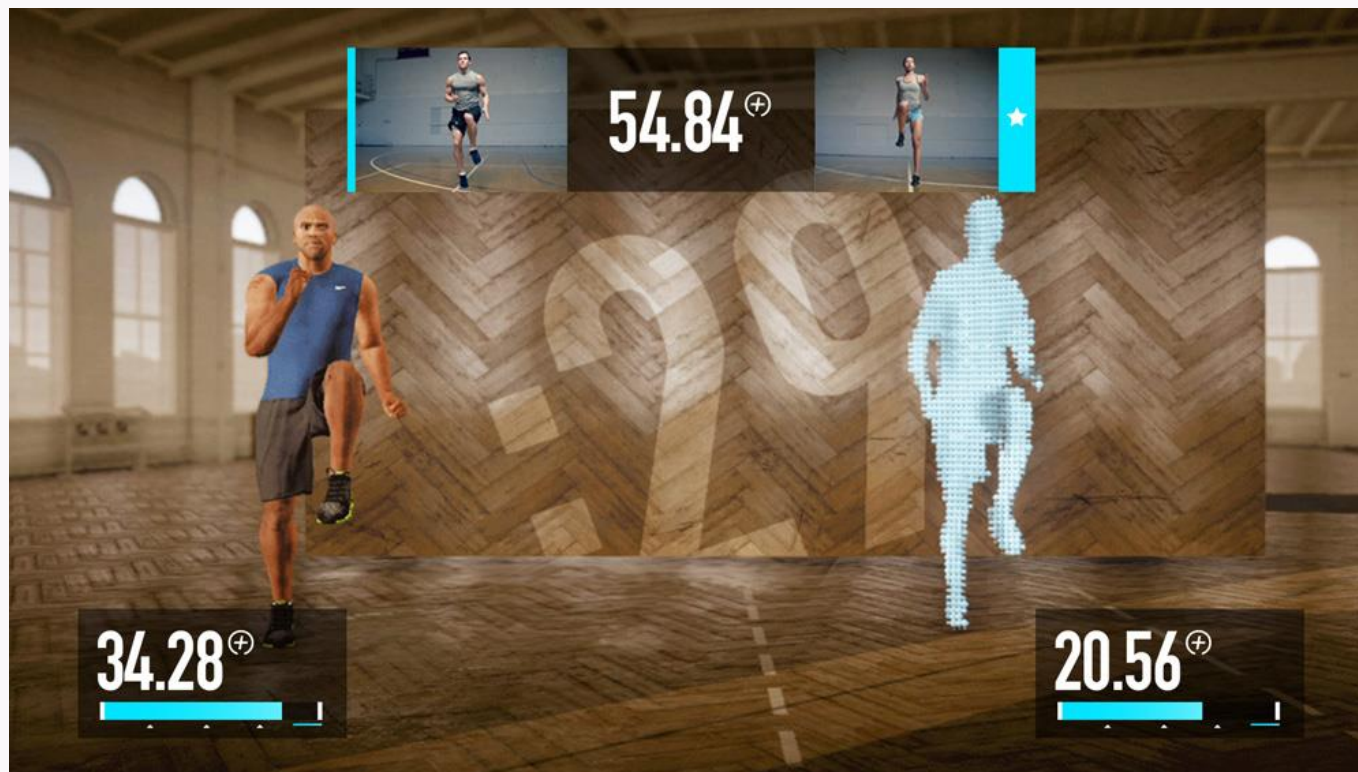
관련 연구

및 사례:

나이키 키넥트

✓ 나이키 키넥트

- 단순히 영상을 따라하는 정도의 기능
- 자세 개선 정도는 분석 불가능



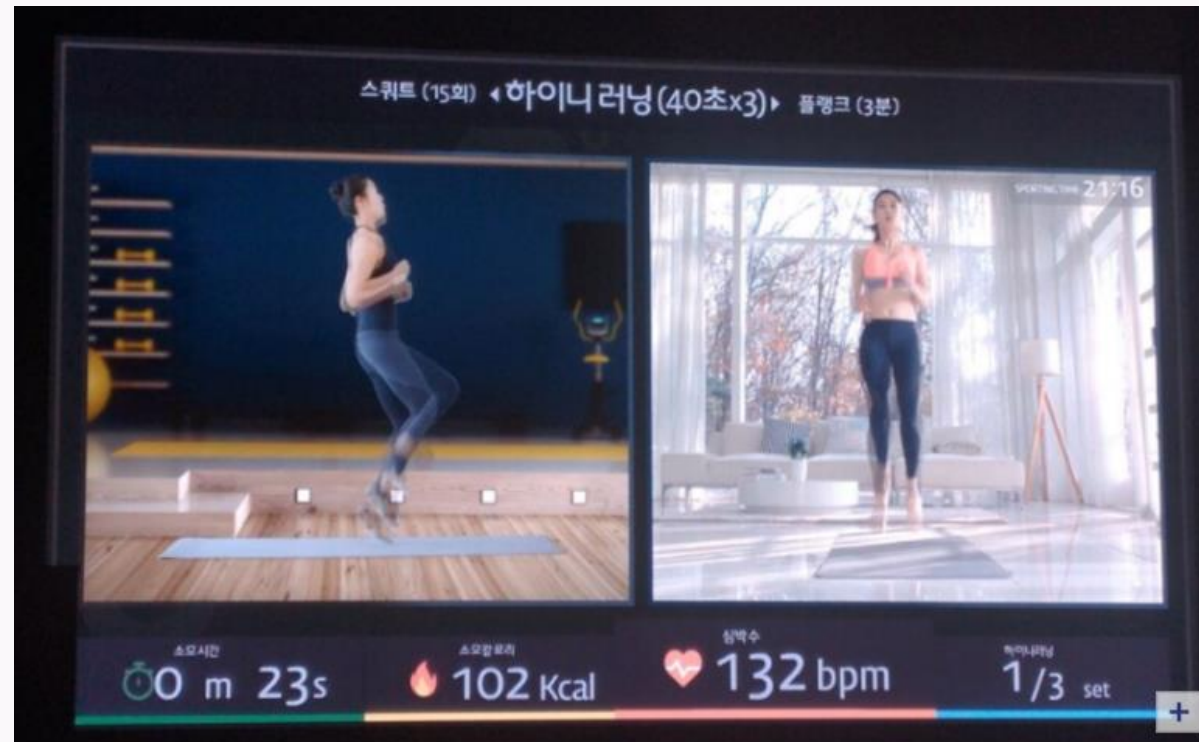
관련 연구

및 사례:

카카오 홈트

✓ 카카오 홈트

- 나이키 키넥트와 비슷한 정도의 기능 (자세 개선 정도 분석 불가능)
- 원격 트레이닝 기능 부재



관련 연구 및 사례: 엔핏

✓ 엔핏

- 자세 분석 기능 부재 → 부상 위험으로 이어짐
- 육체적인 서비스가 불가능



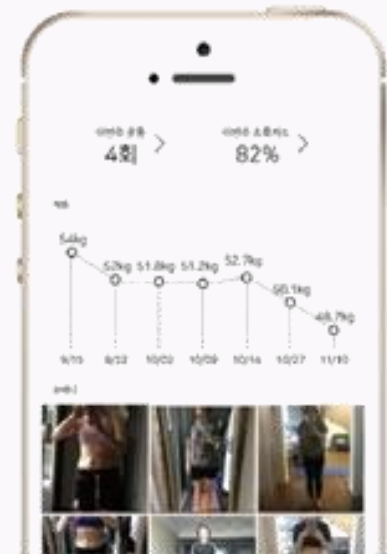
운동관리



식단관리



일상관리



성과

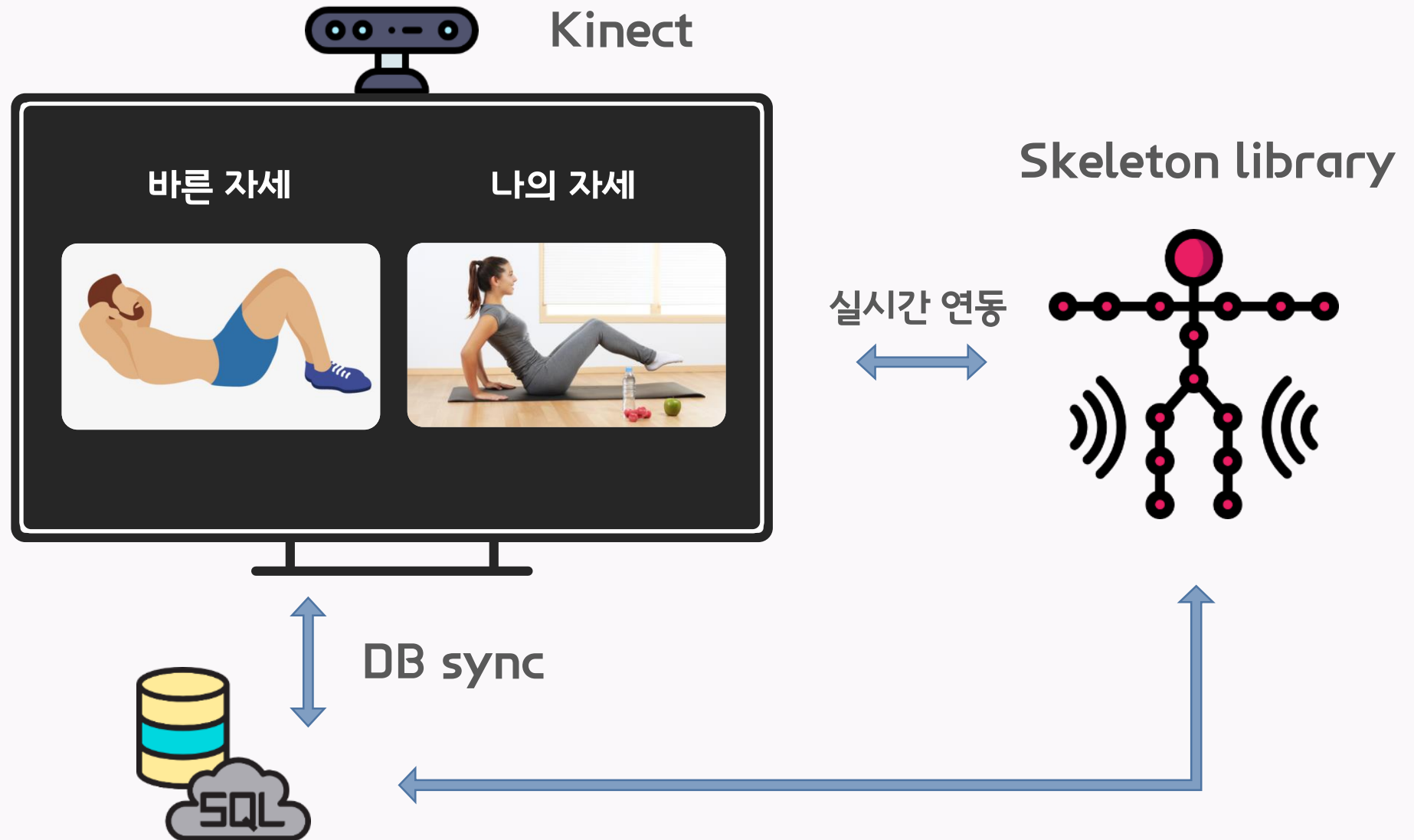
시스템 수행
시나리오:
메인



시스템 수행

시나리오:

트레이닝



시스템 수행 시나리오: 운동 일기

- 한달 동안의 운동 동작 별 **정확도 & 운동 시간** 시각적으로 확인 가능

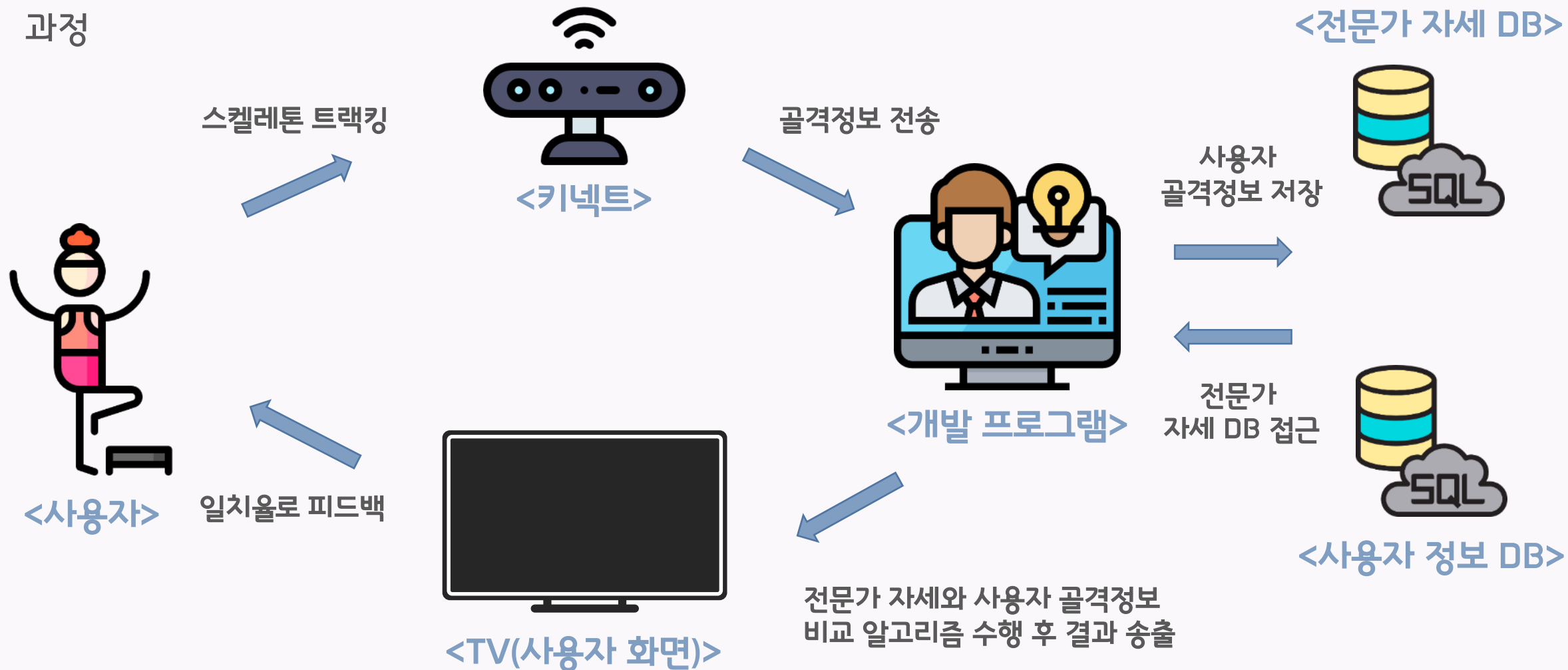


사용자 동작 데이터
DB 저장



시스템 구성도:

과정



개발 내용 및

개발 환경:

개발 환경



<Kinect for Windows v2>

권장 사양

OS	Windows 8 이상
컴파일러	Visual Studio 2012 이상
연결 단자	USB 3.0
CPU	Intel Core i7 3.1GHz
GPU	DirectX 11.0
RAM	4.0GB



시스템 모듈 상세 설계: 데이터 베이스 설계

ref_fit_info	
• fit_code	운동 코드
• fit_core	운동 핵심 부위
• fit_video	영상 리소스
• fit_index1_x	비교 부위1.x
• fit_index1_y	비교 부위1.y
• fit_index2_x	비교 부위2.x
• fit_index2_y	비교 부위2.y
• fit_index3_x	비교 부위3.x
• fit_index3_y	비교 부위3.y

정확도 알고리즘에 사용

fit_info	
• fit_code	운동 코드
• fit_name	운동 이름
• fit_core	운동 핵심 부위
• fit_video	영상 리소스
• fit_plan	운동 설명

user_info	
• user_id	아이디
• user_passwd	비밀번호
• user_name	이름
• user_gender	성별
• user_age	나이
• user_height	키
• user_weight	몸무게
• user_phone	번호

fit_record	
• user_id	사용자 아이디
• fit_code	운동 코드
• fit_time	운동 시간
• fit_accuracy	정확도
• fit_date	운동 날짜

user_info :
회원 개인 정보와 운동기록 관리
fit_info :
운동 관련 정보와 영상 리소스 경로 저장
fit_record :
회원 별 운동 기록 저장
ref_fit_info :
레퍼런스 운동 관련 정보 저장

시스템 모듈 상세 설계 : 키넥트 어플리케이션

회원가입 모듈 - WPF어플리케이션

~ 클래스

기능

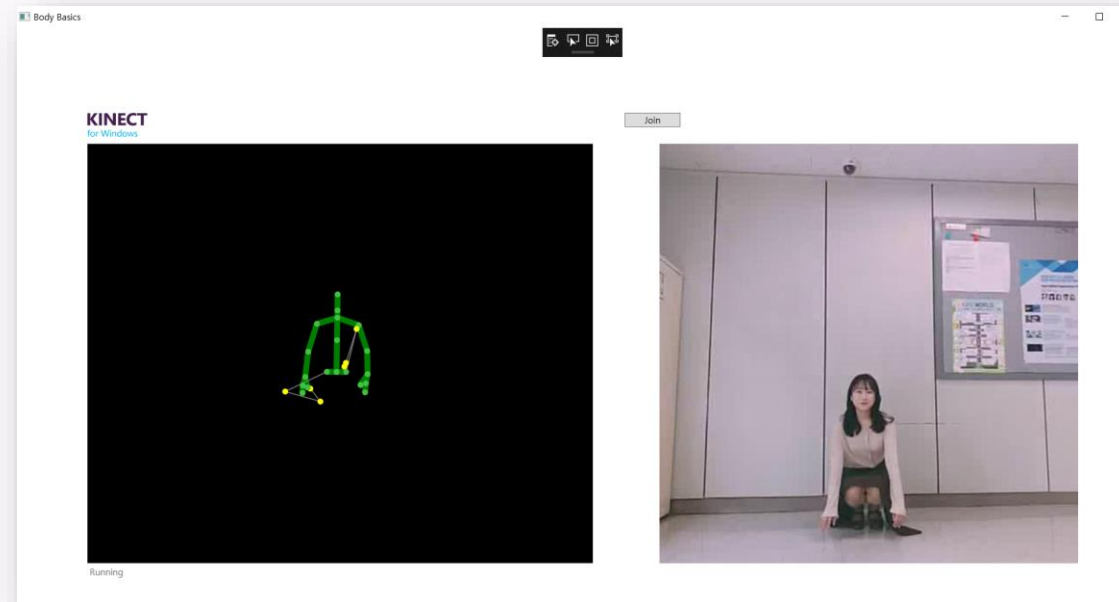
1. 회원가입에 필요한 정보 관리
2. 개인 별 운동 정보를 넣음
3. user_id, user_passwd, user_age, user_height, user_gender, user_weight, user_phone

다루는 정보

- client.BaseAddress - 서버와 연결
- client.PostAsJsonAsync() - 객체를 JSON타입으로 웹 서버에 보낸다.

회원가입을 DB에 반영하는 시점

사용자가 가입에 필요한 정보를 모두 올바르게 기입 하고 회원가입 버튼을 눌렀을 때.



아이디 :	<input type="text"/>
비밀번호 :	<input type="password"/>
이름 :	<input type="text"/>
성별 :	<input type="text"/>
나이 :	<input type="text"/>

키 :	<input type="text"/>
몸무게 :	<input type="text"/>
전화번호 :	<input type="text"/>
<input type="button" value="회원가입"/>	

시스템 모듈 상세 설계:

키넥트 어플리케이션

btn_goto_login_Click()	
형식	void btn_goto_login_Click(object sender, RoutedEventArgs e)
리턴값	Void (실행)
설명	<p>페이지 간의 전환</p> <ul style="list-style-type: none">• var login = new LoginPage(); //전환하고자 하는 페이지• This.Content = login;
예시	btn_goto_login_Click(object sender, RoutedEventArgs e)

시스템 모듈 상세 설계:

회원가입 페이지 - 1

btn_user_post_Click()

형식	private async void btn_user_post_Click(object sender, RoutedEventArgs e)
리턴값	Void (실행)
설명	<ul style="list-style-type: none">• 사용자 정보를 관리하기 위한 회원가입 함수• 회원가입 페이지(JoinPage.xml) 폼에서 입력 값 매핑 된 Uri로 POST방식으로 웹서버 통해 DB user_info 테이블에 저장
예시	btn_goto_login_Click(object sender, RoutedEventArgs e)

시스템 모듈 상세 설계:

회원가입 페이지 - 2

GetUsers();

형식	private async void GetUsers(object sender, RoutedEventArgs e)
리턴값	Void (실행)
설명	데이터베이스에 있는 값을 얻어온다. 회원 정보와 회원 전용 운동 정보를 얻어 올 수 있다.
예시	private async void GetUsers(object sender, RoutedEventArgs e)

시스템 모듈 상세 설계:

서버 - 컨트롤러 패키지

UserRestController: 클라이언트 요청을 웹서버 Url 매핑하여 해당 서비스 분기

String joinUserPost()	
형식	String joinUserPost() (UserVO uservo)
리턴값	String
설명	클라이언트 회원가입 폼을 통해 POST 요청 시 입력 받은 정보를 uservo 도메인 객체에 매핑하여 회원 추가 서비스 호출

ResponseEntity<UserVO> readUserGet()	
형식	ResponseEntity<UserVO> readUserGet(String userid)
리턴값	ResponseEntity<UserVO> return 객체는 HTTP 응답 메시지에 해당 회원 정보 도메인, 헤더 정보, 상태 코드 바인딩하여 리턴
설명	클라이언트에서 회원 별 정보 확인을 위해 GET 요청 시 id 값을 userid 인자에 매핑하여 해당 회원 정보 읽기 서비스 호출

시스템 모듈 상세 설계:

서버 - 도메인 패키지

UserVO: 회원정보라는 현실의 추상적 개념 -> 논리적 속성으로 표현한 클래스,
DB user_info 테이블 컬럼명과 변수명 동일하게 맵핑

UserVO

설명

```
- private String user_id;  
    - private String user_passwd; // 사용자 비밀번호  
    - private String user_name;   // 사용자 이름  
    - private String user_gender; // 사용자 성  
    - private int user_age;        // 사용자 나이  
    - private int user_height;     // 사용자 키  
    - private int user_weight;     // 사용자 몸무게  
    - private String user_phone;  // 사용자 번호
```

시스템 모듈 상세 설계:

서버 - 익셉션 패키지

Exception	
형식	DataNotFoundException: Exception UserControllerAdvice: Exception
리턴	오류 예방
설명	<ul style="list-style-type: none">• NotFoundException: Exception 클래스를 상속 예외 발생 클래스• UserControllerAdvice: Exception 발생 시 예외 관련 정보 노출을 막기 위해 별도 에러 페이지 리턴<ul style="list-style-type: none">- String handleException(Exception e)- String handleException(DataNotFoundException e)

시스템 모듈 상세 설계:

서버 - 퍼시스턴스

UserDAO	
형식	<ul style="list-style-type: none">• UserDAO: DB의 user_info 테이블 CRUD를 위한 인터페이스• UserDAOImpl: UserDAO 구현 객체 DB user_info SQL통한 액세스
리턴값	
설명	<ul style="list-style-type: none">- UserVO read(String user_id): user_id와 일치하는 회원 정보 리턴- List<UserVO> readList(): user_info 테이블에 모든 회원 정보 리턴- void add(UserVO uservo): 회원 추가- void update(UserVO uservo): 회원 정보 수정- void delete(String user_id): 회원 삭제

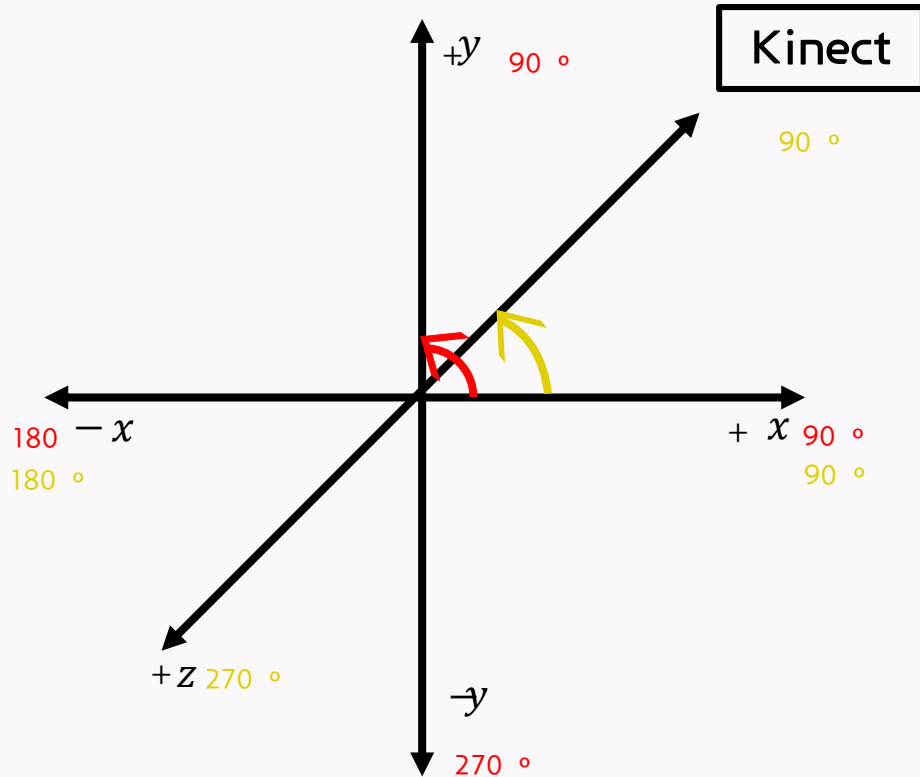
시스템 모듈 상세 설계:

서버 - 서비스 패키지

UserService	
형식	<ul style="list-style-type: none">UserService: DB의 user_info 테이블 CRUD 및 비즈니스 로직 포함한 서비스 인터페이스UserServiceImpl: UserService 구현 객체, DAO 메소드 호출을 통한 비즈니스 로직 처리
리턴값	
설명	<ul style="list-style-type: none">- UserVO readMember(String user_id)- List<UserVO> readMemberList()- void addMember(UserVO uservo)- void updateMember(UserVO uservo)- void deleteMember(String user_id)



시스템 모듈 상세 설계 : 자세 비교 알고리즘



z 좌표를 추가하여 정확도를 높임

Math.Atan2 :

매개변수 : y포인터 , x포인터

리턴 값 : $-\pi \sim +\pi$

반환 받은 Radian값을 일반 각도 값으로 변환
음수 값이 반환 되는 경우

리턴 값+360 처리하여

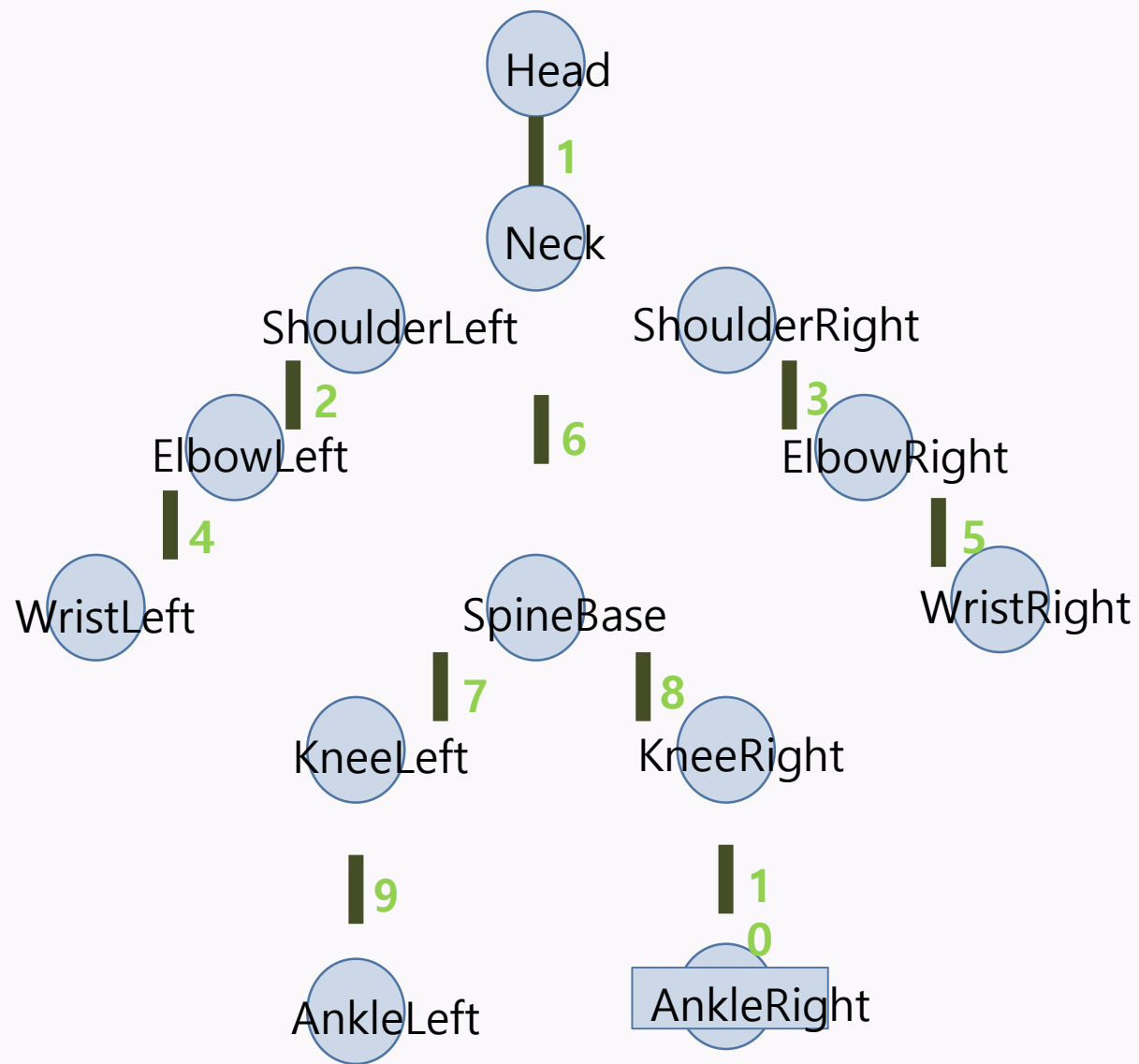
0~360 도 사이의 각도 값 반환

예시 :

```
if (Head_Neck_xy < 0)
{
    Head_Neck_xy += 360;
}
if (Head_Neck_xz < 0)
{
    Head_Neck_xz += 360;
}
```



시스템 모듈 상세 설계 : 자세 비교 알고리즘

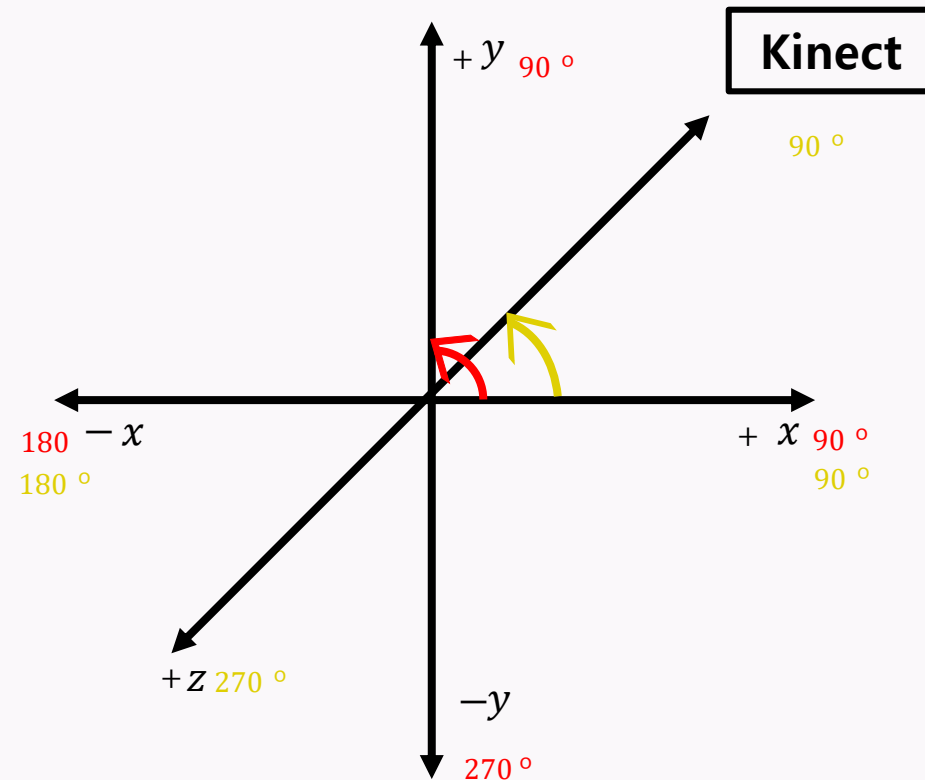
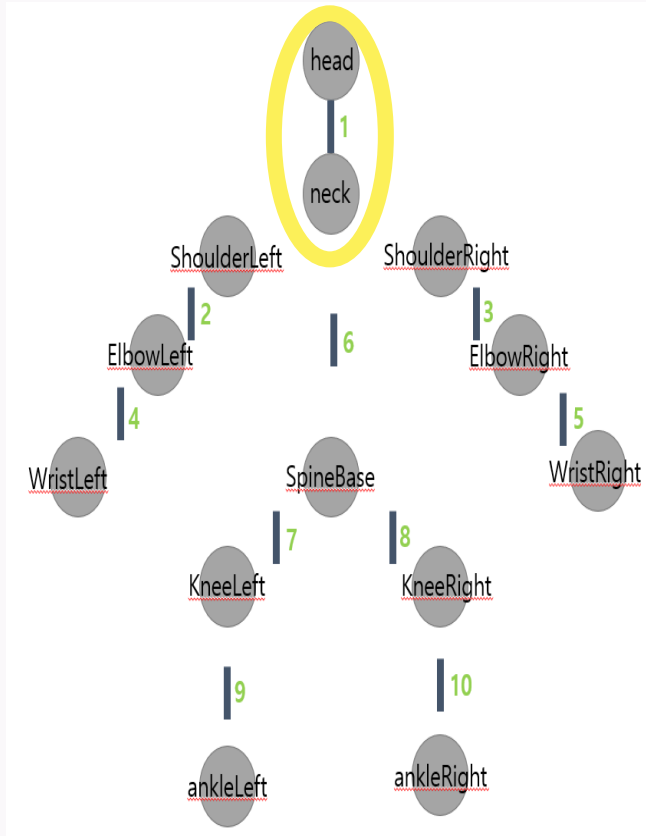




시스템 모듈 상세 설계 : 자세 비교 알고리즘

(1) Head-Neck

```
double Head_Neck_xy=Math.Atan2(Head_y - Neck_y, Head_x - Neck_x) * (180.0 / Math.PI);  
double Head_Neck_xz=-Math.Atan2(Head_z - Neck_z, Head_x - Neck_x) * (180.0 / Math.PI);
```

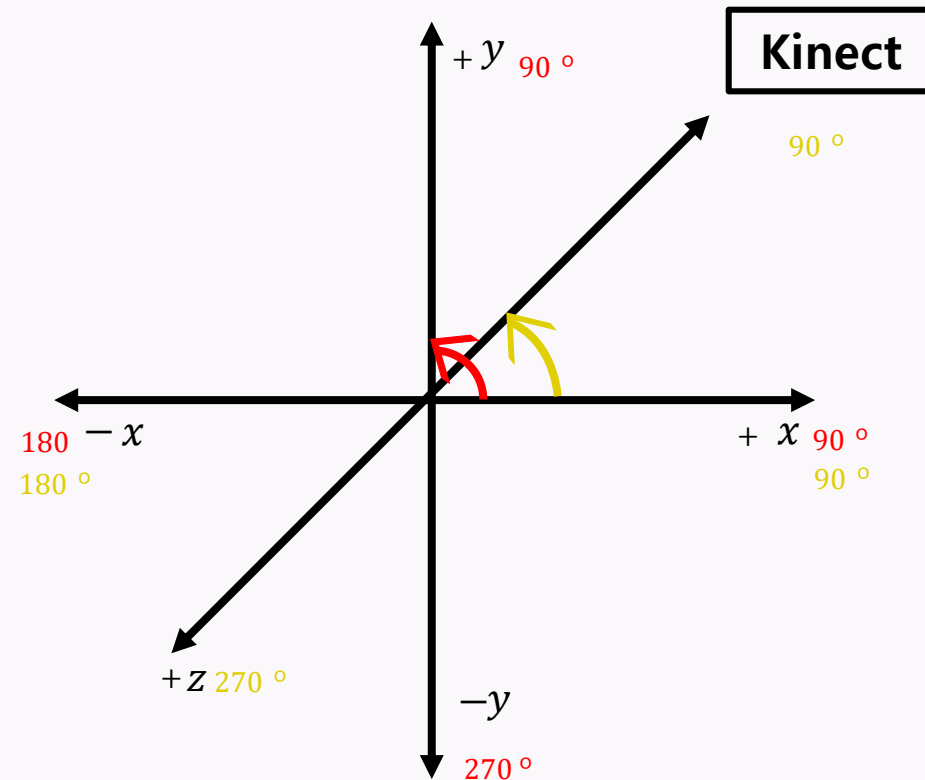
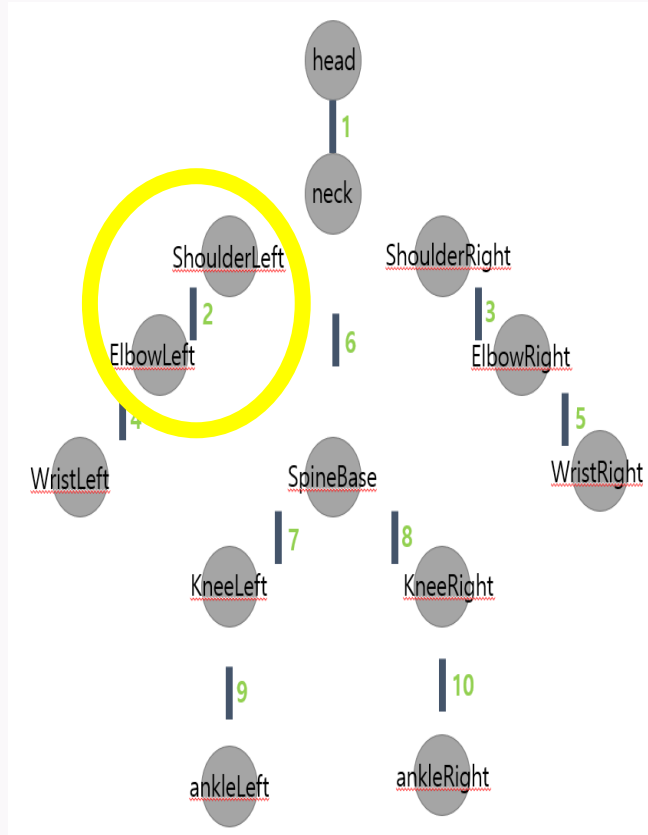




시스템 모듈 상세 설계 : 자세 비교 알고리즘

(2) ShoulderLeft-ElbowLeft

```
double ElbowLeft_ShoulderLeft_xy = Math.Atan2(ElbowLeft_y-ShoulderLeft_y,ElbowLeft_x-ShoulderLeft_x) * (180.0 / Math.PI);  
double ElbowLeft_ShoulderLeft_xz = -Math.Atan2(ElbowLeft_z - ShoulderLeft_z, ElbowLeft_x - ShoulderLeft_x) * (180.0 / Math.PI);
```

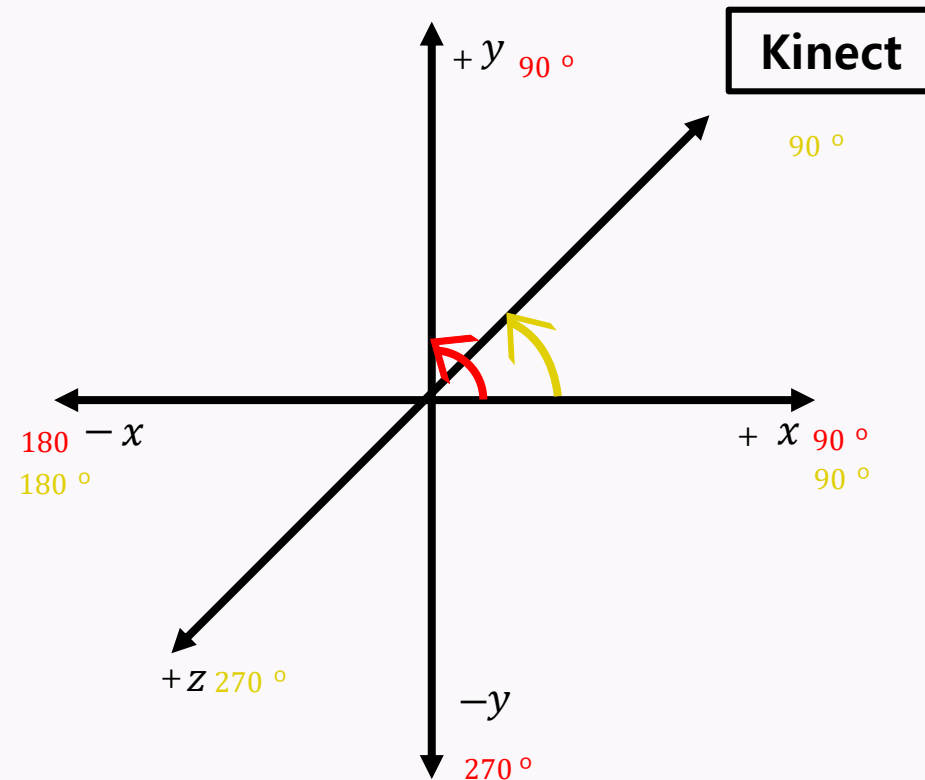
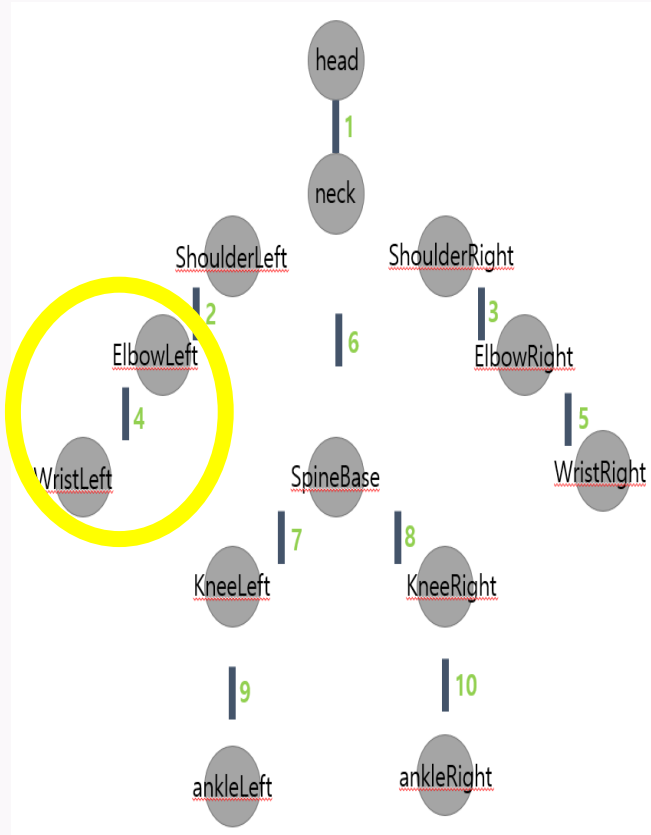




시스템 모듈 상세 설계 : 자세 비교 알고리즘

(3) ElbowLeft-WristLeft

```
double ElbowLeft_WristLeft_xy = Math.Atan2(WristLeft_y - ElbowLeft_y, WristLeft_x - ElbowLeft_x) * (180.0 / Math.PI); double  
ElbowLeft_WristLeft_xz = -Math.Atan2(WristLeft_z - ElbowLeft_z, WristLeft_x - ElbowLeft_x) * (180.0 / Math.PI);
```

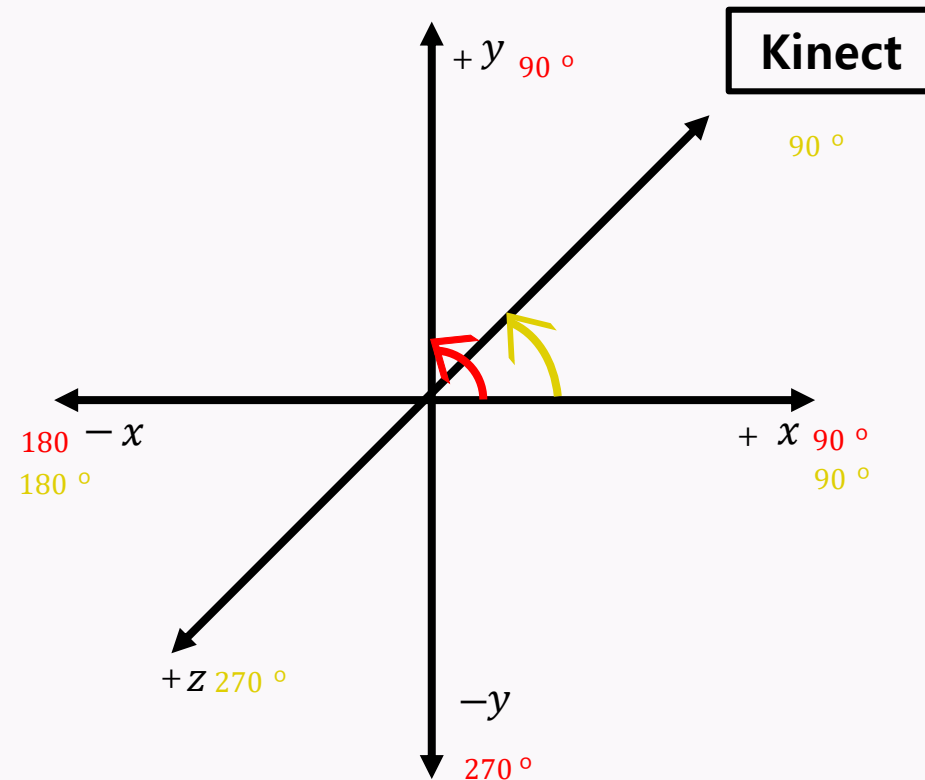
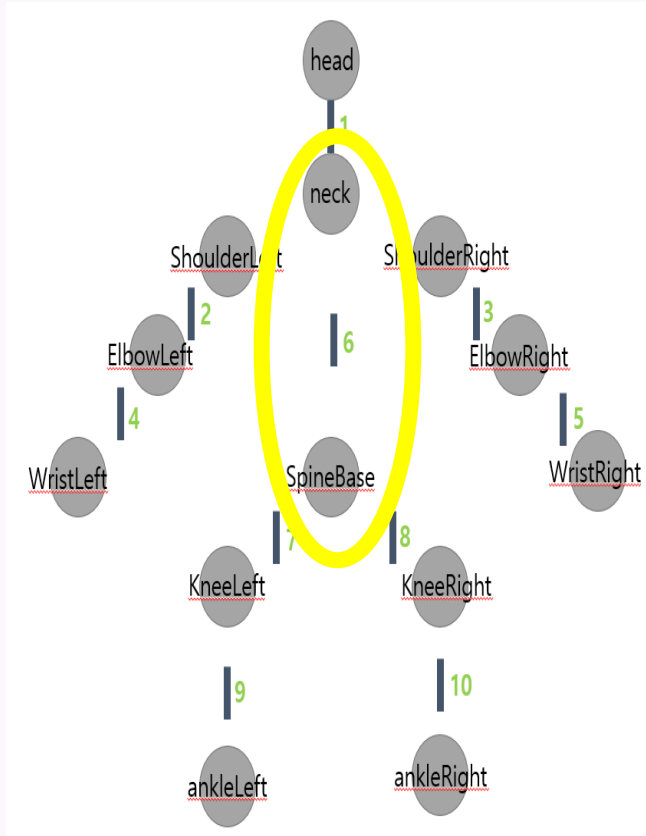




시스템 모듈 상세 설계 : 자세 비교 알고리즘

(4) Neck - SpineBase

```
double Neck_SpineBase_xy = Math.Atan2(Neck_y - SpineBase_y, Neck_x - SpineBase_x) * (180.0 / Math.PI); double  
Neck_SpineBase_xz = -Math.Atan2(Neck_z - SpineBase_z, Neck_x - SpineBase_x) * (180.0 / Math.PI);
```





시스템 모듈 상세 설계 : 자세 데이터 수집 방법



...	10(프레임)	...	20(프레임)	...	30(프레임)
...		



10프레임당 한 개의 자세 기울기 각도 데이터를 뽑아서 이차원 배열에 저장한다.


1	2	3	4	5	6	7	8	9	10	골격 번호
										10프레임
										20프레임
										...



시스템 모듈 상세 설계 : 자세 데이터 베이스 테이블


< 운동 1개당 2개의 테이블이 존재 >

골격	1	2	3	4	5	6	7	8	9	10
10 (프레임)										
20 (프레임)										
30 (프레임)										
40 (프레임)										
...										



ex) 30프레임에서 수집된 3번 뼈 x-y 기울기 각도 값

골격	1	2	3	4	5	6	7	8	9	10
10 (프레임)										
20 (프레임)										
30 (프레임)										
40 (프레임)										
...										



ex) 30프레임에서 수집된 3번 뼈 x-z 기울기 각도 값



시스템 모듈 상세 설계 : 자세 데이터 비교

[mySQL]

골격	1	2	3	4	5	6	7	8	9	10
10 (프레임)										
20 (프레임)										
30 (프레임)										
40 (프레임)										
...										



[Visual Studio]

골격	1	2	3	4	5	6	7	8	9	10
10 (프레임)										
20 (프레임)										
30 (프레임)										
40 (프레임)										
...										

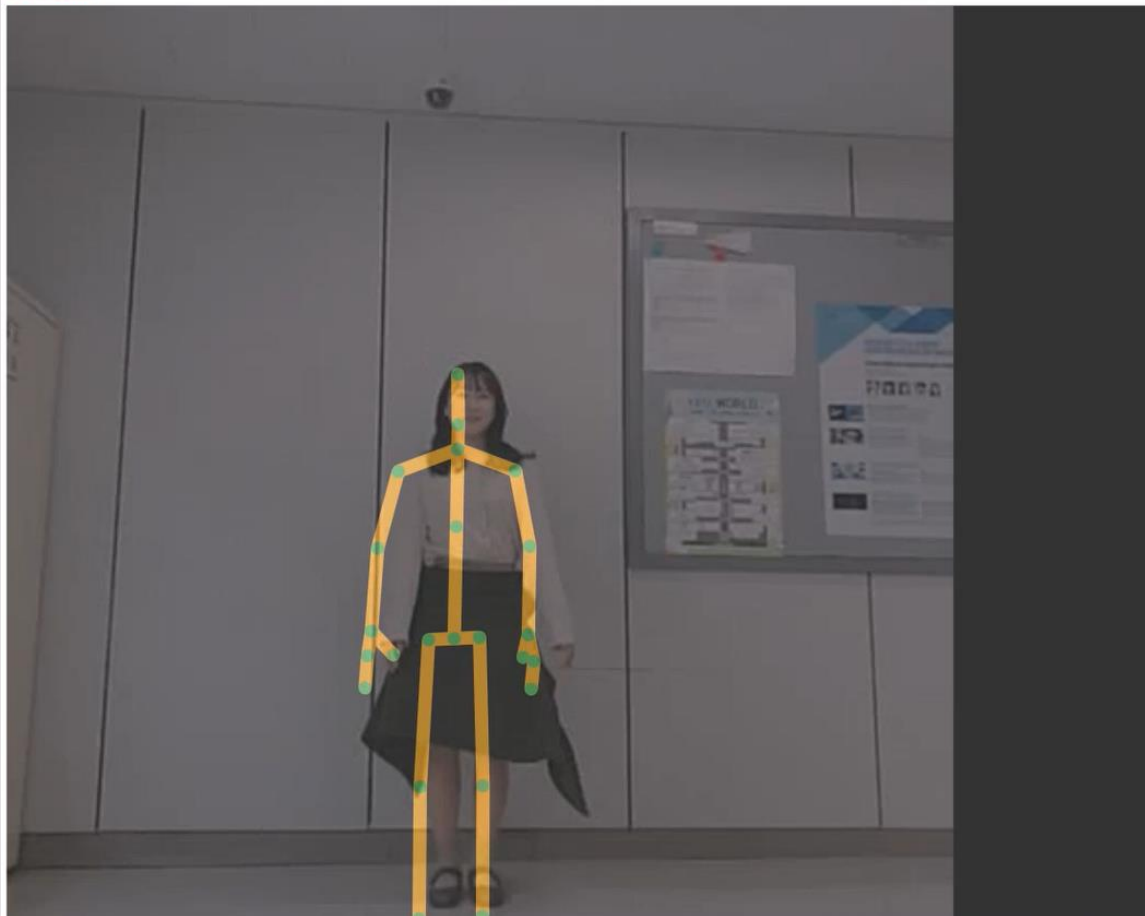
1. 선택한 운동의 데이터 테이블을 DB에서 가져온다.
2. 사용자가 운동을 시작하면 사용자의 자세를 10프레임 간격으로 가져온다.
3. DB에서 가져온 정보와 실시간으로 들어오는 자세를 비교한다.



시스템 모듈 상세 설계 : 동작 지연 방지

KINECT
for Windows

여기는 운동 2



Body Basics



레퍼런스 영상을 바로 따라할 수 있게 실시간 영상 레이어 위에
반투명 레퍼런스 영상을 재생 함으로써 동작 지연을 방지함.



시스템 모듈 상세 설계 : 유연성 상/중/하

필요성	사람들마다 유연성과 지구력이 다름			
해결 방안	유연성마다 난이도를 조절함			
구체적 방법	일치율 오차범위에 차이를 둠			
	<div>평가 \ 난이도</div>	<div> 상</div>	<div> 중</div>	<div> 하</div>
	Perfect	95 이상	90 이상	80 이상
	Excellent	90 ~ 95	80 ~ 90	70 ~ 80
	Good	85 ~ 90	70 ~ 80	60 ~ 70
	Bad	80 ~ 85	60 ~ 70	50 ~ 60
	Miss	75 ~ 80	50 ~ 60	40 ~ 50

26



시스템 모듈 상세 설계 : 일치율 도출 식

$$\left(1 - \frac{\text{레퍼런스 기울기} - \text{실시간 기울기}}{360} \times |Ref.lean - Real.lean| \right) \times 100$$

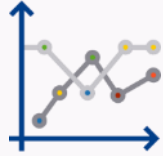
의미 있는 일치율 값을 출력하기 위하여 자세의 정확도를 백분율로 표현함



시스템 모듈 상세 설계 : 운동 일기

필요성	<ul style="list-style-type: none">단순히 운동을 하는 기능에서 끝나면 지속적인 운동을 하기 어려움												
해결방안	<ul style="list-style-type: none">운동일기를 통해 운동시간, 운동을 기록함데이터베이스 정보를 가져옴												
구체적 방법	<ul style="list-style-type: none">데이터 베이스 fit_record 테이블에서 사용자의 운동 정보와 기록을 C#과 연동C# Xaml과 cs파일에서 사용자 정의 코드를 사용함 <div data-bbox="1411 682 2262 1225"><table border="1"><thead><tr><th>날짜</th><th>값</th></tr></thead><tbody><tr><td>2019-06-15</td><td>300</td></tr><tr><td>2019-06-16</td><td>180</td></tr><tr><td>2019-06-17</td><td>120</td></tr><tr><td>2019-06-18</td><td>201</td></tr><tr><td>2019-06-19</td><td>60</td></tr></tbody></table></div>	날짜	값	2019-06-15	300	2019-06-16	180	2019-06-17	120	2019-06-18	201	2019-06-19	60
날짜	값												
2019-06-15	300												
2019-06-16	180												
2019-06-17	120												
2019-06-18	201												
2019-06-19	60												

개발 내용 및
개발 환경:
개발 내용



- 실시간으로 skeleton 값 비교

데이터베이스에 저장 되어있는

“레퍼런스 skeleton값”과 “사용자의 skeleton값”을 비교 분석

➔ 분석 알고리즘 구현



- 사용자의 운동 시간과 운동 자세 분석

사용자의 운동 자세와 운동 시간을

DB에 저장하여 그래프로 제시

개발 내용 및

개발 환경:

개발 환경



<Kinect for Windows v2>

속성	수치
해상도 / FPS	1920X1080 30fps
인물 영역	6명
관절	25관절
검출 범위	0.5 ~ 4.5m
각도(수평 / 수직)	70도 60도
손의 개폐 상태 인식	SDK에서 가능

개발 내용 및

개발 환경:

개발 환경



✓ MySQL

- 표준 데이터베이스 질의 언어인 SQL을 사용하는
공개 소스의 관계형 데이터베이스 관리 시스템
- 매우 빠름, 유연함, 사용하기 쉬움
- 다중사용자, 다중 스레드 지원
- C, C++, 자바, 파이썬 스크립트 등을 위한
응용 프로그램 인터페이스(API)제공

개발 내용 및

개발 환경:

개발 환경

KINECTTM
for Windows[®]

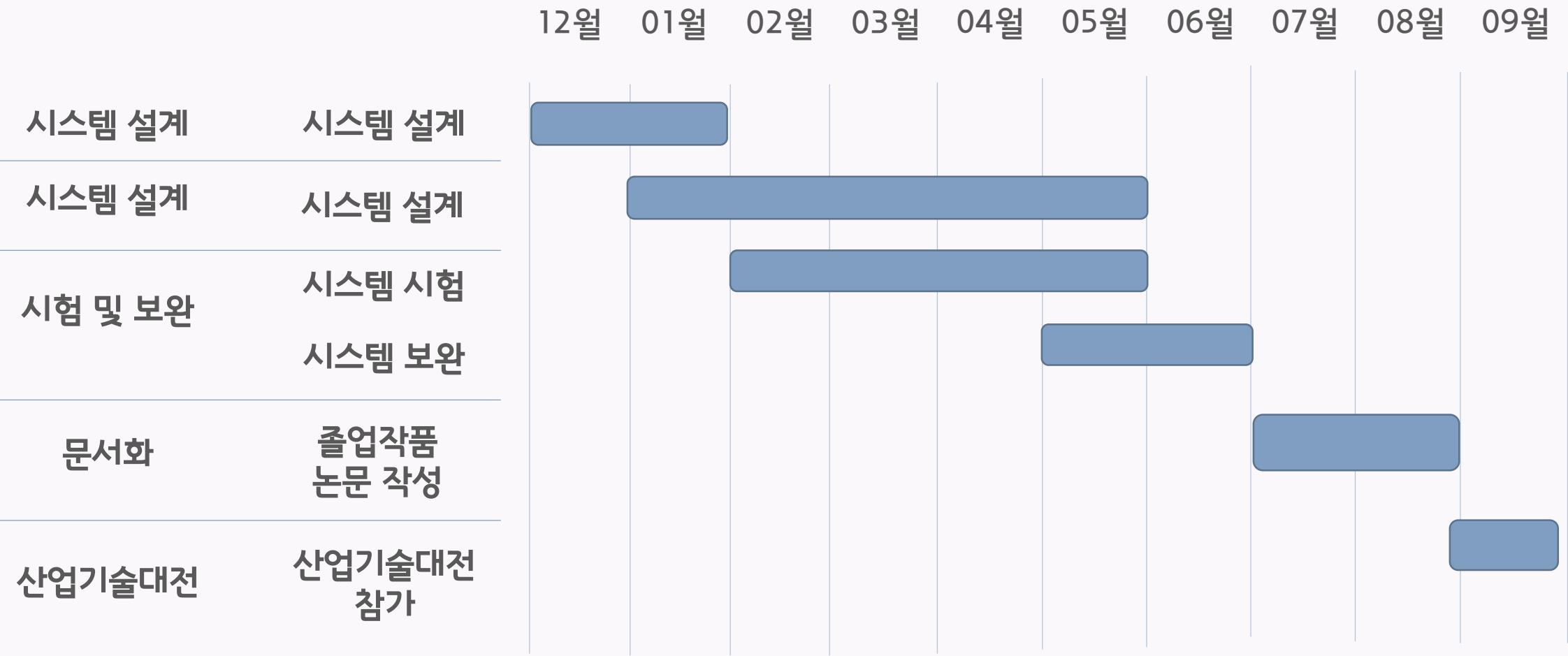
✓ Kinect SDK

- Microsoft Research에서 발표
- 32비트 및 64비트 지원
- 장치드라이버 , API 프레임워크 제공
- 예제와 문서 제공

업무 분담:

	<u>양수연</u>	<u>김두환</u>	<u>정유하</u>
자료수집	Kinect 사용법 , 클라이언트 개발	데이터베이스 / 서버 구현	비교 알고리즘 구현
설계	자료 설계	구조 설계	구조 설계
구현	WPF 구현	Skeleton 정보 DB 연동 모션 인식	Kinect skeleton 비교 알고리즘
테스트	Skeleton 테스트 DB 연동 테스트 모션인식 테스트		

종합 설계 수행 일정 : 진행과정



	양수연	김두환	정유하
Github ID	vcindyvdandi	carinodudu	yoocha9621
Github Address	https://github.com/vcindyv/dandi		

필요 기술 및 참고문헌:

- KITness : 키넥트 센서를 활용한 인터랙티브 운동 자세 분석 시스템
- <https://developer.microsoft.com/ko-kr/windows/Kinect>: 키넥트 스튜디오

Kinect for Windows SDK 2.0

Windows 10용 데스크톱 응용 프로그램을 빌드하거나 Microsoft Store에서 Kinect v2 UWP 앱을 판매하여 고유한 Windows 런타임 기능을 활용하고 배포를 확장합니다.

[KINECT FOR WINDOWS SDK 다운로드 >](#)



감사합니다!