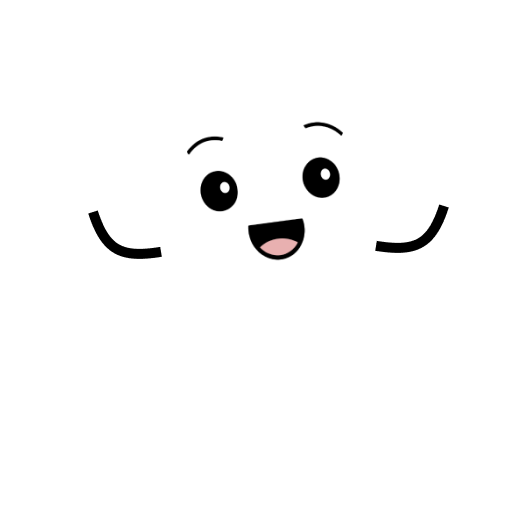


캡스톤디자인(1)

프로젝트 보고서



|  |  |
| --- | --- |
| 과목 | 캡스톤디자인(1) |
| 교수 | 박상오 교수님 |
| 팀 이름 | 아치 |
| 팀 구성원 | 20160342 김수진 |
| 20163228 남유선 |
| 20163704 박주현 |

**목차**

[1. 프로젝트 소개 3](#_Toc532252055)

[1.1 프로젝트 이름 3](#_Toc532252056)

[1.2 개발 배경과 목표 3](#_Toc532252057)

[1.3 기대 효과 3](#_Toc532252058)

[1.4 업무분담 및 프로젝트 일정 3](#_Toc532252059)

[1.5 기능 소개 4](#_Toc532252060)

[1.6 개발환경 5](#_Toc532252061)

[1.7 프로젝트 레퍼런스 6](#_Toc532252062)

[2. 개발 내용 7](#_Toc532252063)

[2.1 프로젝트 개요 7](#_Toc532252064)

[2.2 개발 진행 과정 12](#_Toc532252067)

[3. 프로젝트 결과 16](#_Toc532252073)

[3.1 프로젝트 완성도 16](#_Toc532252074)

[3.2 한계 및 개선 방안 16](#_Toc532252075)

[3.3 추후 개발 계획 17](#_Toc532252078)

[3.4 후기 18](#_Toc532252079)

# 프로젝트 소개

## 프로젝트 이름

아치(아름다운 치아)

## 개발 배경과 목표

양치질을 할 때 빠진 곳 없이 구석구석 '제대로' 양치질을 할 수 있도록 도와주는 모바일 어플리케이션 및 칫솔을 개발하고자 한다. 사람들은 누구나 양치질의 중요성을 알고 있으며 누구나 하루에 세 번씩 양치질을 한다. 그럼에도 불구하고 충치로 치과에 방문하는 사람들을 상당히 많으며, 그 중에서 "양치를 열심히 했는데 충치가 생겼다. 이해가 되지 않는다." 라고 말하는 사람도 쉽게 찾아볼 수 있다. 하지만 대부분은 양치질은 여러 번 했을지라도, '제대로' 양치질을 하지 않은 경우일 가능성이 크다. 이와 같은 문제를 해소하기 위해 양치질을 할 때 본인이 어느 곳을 양치했는가를 실시간으로 어플리케이션으로 한눈으로 확인하며 빠뜨리는 곳 없이 양치가 가능한 칫솔을 개발함으로써 사용자로 하여금 좀 더 확실한 양치질과 구강건강의 증진을 목적으로 한다.

## 기대 효과

우리의 칫솔을 통해 사람들은 양치질을 할 때 모바일 애플리케이션의 치아모델을 통해 양치질이 덜 된 곳을 시각적으로 확인 수 있다. 따라서, 사용자가 올바른 양치 습관을 형성 할 수 있도록 도와줌으로써, 치아를 건강하게 한다. 또한 아이들에게 양치에 대한 흥미를 유발하여 유아기 양치 교육용으로도 사용할 수 있을 것이다.

## 업무분담 및 프로젝트 일정

**김수진**

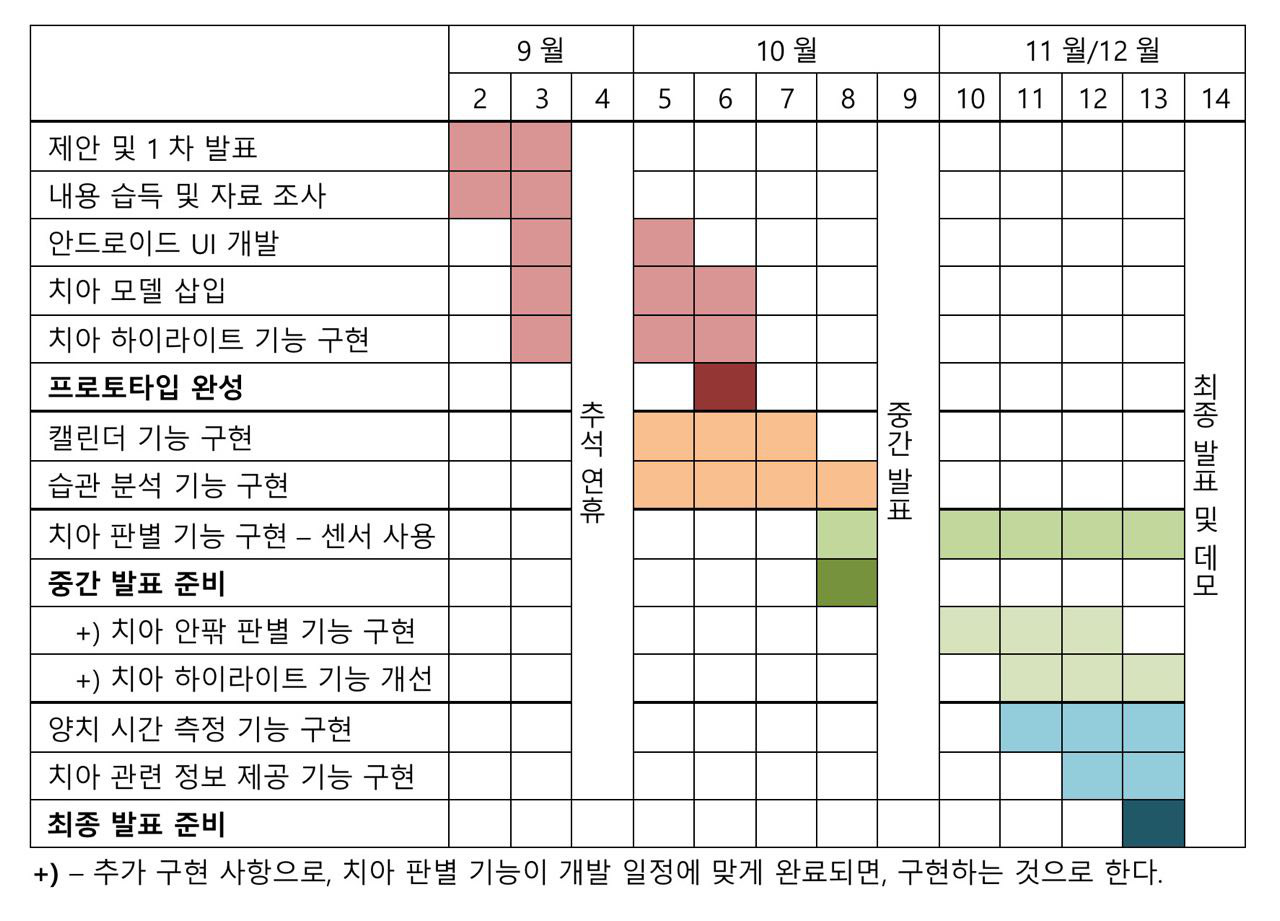
* + 치아 모델 삽입 및 하이라이트 기능 구현
  + 치아 판별 기능 구현
  + 치아 관련 정보 제공 기능 구현

**남유선**

* + 치아 모델 삽입 및 하이라이트 기능 개선
  + 양치 시간 측정기능 구현
  + 양치 습관 분석 기능 구현

**박주현**

* + 치아 안팎 판별 기능 구현
  + 안드로이드 UI 개발

****캘린더 기능 구현

## 기능 소개

**- 양치 모니터링**

칫솔과 모바일 애플리케이션을 연동하여 양치를 하는 동안 양치를 하고 있는 부분과, 양치를 완료한 부분을 애플리케이션을 통해 보여준다. 가속도 자이로 센서를 사용해서 칫솔의 위치 및 각도를 입력 받아 칫솔 솔이 어디에 위치해 있는 지 파악한다. 이 위치를 시각화 하여 애플리케이션을 통해 실시간으로 보여준다. 한 학기 프로젝트임을 감안하여, 사용자는 치아 중 한 면 만을 양치하는 것으로 가정한다.

**- 치아 관련 정보 제공**

건강한 치아를 위해서는 올바른 양치질만큼 생활 습관도 중요하다. 따라서 양치하는 동안 치아 건강 관련 정보 한 줄을 알려준다.

**- 양치 시간 측정**

양치를 너무 짧게 하면 이를 제대로 닦지 않은 경우가 많고, 또 너무 오래 하면 오히려 치아가 망가질 수 있기 때문에 적당한 시간동안 양치 하는 것이 올바른 양치질의 방법이다. 따라서 양치 시간을 보여주고 및 적정 종료 시간을 알람을 통해 알려준다.

**- 양치 습관 분석**

양치는 습관을 관리하는 것이 중요하다. 사용자의 양치 습관을 분석하여 나쁜 습관을 고칠 수 있도록 도와준다. (사용자의 일주일 양치 점수 및 지난주와 비교하여 이번주 양치에 대해 코멘트를 보여준다. 그리고 양치 한번에 대해 양치 시간과 점수, 양치를 분석하여 보여준다.) (양치에 대해 점수를 매기고 시간을 보여주고 그에 대한 분석을 하여 보여준다. ) 예를 들면, 한 곳을 너무 오랫동안 칫솔질 한다 거나, 자주 놓치는 부분이 있는 경우 이를 사용자에게 알려준다.

**- 캘린더**

캘린더에 하루 양치 횟수를 동그라미 개수를 통해 나타내고, 사용자의 한 달 양치 횟수를 분석하여 건강한 양치질과 더불어 하루 3번 양치 하도록 장려한다.

## 개발환경

* + JAVA SDK 10.0.2
  + Kotlin 1.2.71
  + Android SDK 26.0
  + Android Studio 3.1.41
  + Android Platform 8.0
  + Arduino 1.8.6
  + OpenGL 4.6

## 프로젝트 레퍼런스

|  |  |
| --- | --- |
| GitHub Organization Name : Team-Achi | |
| 아치 애플리케이션 | <https://github.com/Team-Achi/Android> |
| 전동칫솔 소프트웨어 | <https://github.com/Team_Achi/Electric-Toothbrush> |
| 문서 | <https://github.com/Team-Achi/Docs> |
| 3D 모델 뷰어 | <https://github.com/Team-Achi/android-3D-model-viewer> |

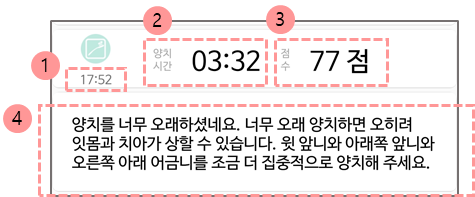
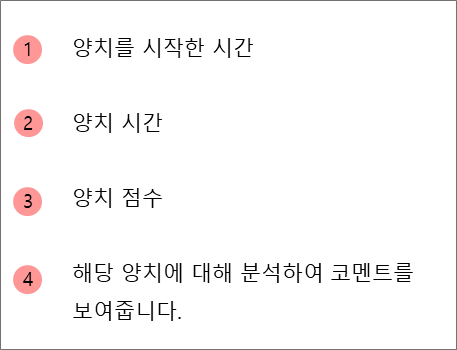
# 개발 내용

## 프로젝트 개요

### 애플리케이션

#### 인트로 페이지

#### 습관분석 페이지

****

#### 모니터링 페이지

#### 캘린더 페이지

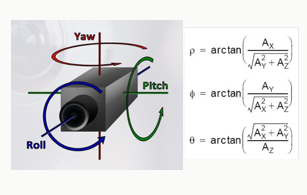
### 칫솔

#### 하드웨어 조립

#### 아두이노 프로그래밍

|  |  |  |  |
| --- | --- | --- | --- |
| **모듈** | **모델명** | **Module Pin #** | **Arduino Pin #** |
| 아두이노 보드 | 아두이노 우노 R3 SMD |  |  |
| 자이로 가속도 6축 센서 | MPU-6050 | VCC | 5V |
| GND | GND |
| SCL | A5 |
| SDA | A4 |
| INT | D2 |
| 블루투스 모듈 | 블루투스 HC-06 | VCC | 5V |
| GND | GND |
| IXD | D2 |
| RXD | D3 |
| 압력센서 | FSR | GND | GND, A0 |
| VCC | 5V |
| 진동모터 | DVM1234 | GND | D6 |
| VCC | 3.3V |
| 스위치 | Tact Switch | GND | GND |
| Other | D9 |

#### 3축 각도 분석 알고리즘을 통한 치아의 판별

* Roll: X축 주위로 회전한 각도
* Pitch: Y축 주위로 회전한 각도
* Yaw: Z축 주위로 회전한 각도
* 해당 각도들을 인식하여 해당 각도에 매핑되는 치아번호를 블루투스 모듈을 통하여 전송해 안드로이드 기기가 전달받을 수 있도록 합니다.

## 개발 진행 과정

### 안드로이드 UI 개발

팀원들과 전체적인 기능 및 세부사항 조율을 위해 안드로이드 애플리케이션 UI 개발을 먼저 시작하였다. UI개발을 하면서 팀원 간 프로젝트에 대해서 다르게 이해한 부분을 정리할 수 있었고, 안드로이드 UI 개발을 프로젝트 초기에 완료함으로써 남은 기간 동안 기능 구현에 집중할 수 있었다.

### 블루투스

블루투스 3.0 HC-06을 사용하여 아두이노와 애플리케이션을 통신하였다. 치아 모델을 눌러 블루투스 통신 시작 및 종료를 하였다. 만약 블루투스가 켜져 있지 않으면, 블루투스를 키고 페어링 하도록 사용자에게 메시지를 보여준다. 블루투스 통신을 할 때 “치아 번호/치아 번호의 Checksum/압력의 정도”의 형식으로 정보를 100ms 간격으로 받는다. 가끔 아두이노가 보내는 값과 애플리케이션에서 받는 값이 달라 이를 확인하기 위해 치아번호의 Checksum를 추가하였다. Checksum이 다른 값들은 양치 모니터링하는 데 사용하지 않으며, 양치 시간 측정에는 사용된다.

### 애플리케이션 치아 모델

**오픈소스 프로젝트 - 3D모델 뷰어**

본 프로젝트에서 ‘양치 모니터링’시 실시간으로 현재 양치중인 치아를 보여주는 기능을 제안했다. 해당 기능은 OpenGL 라이브러리를 사용하는 GitHub의 오픈소스 프로젝트 android-3D-model-viewer (<https://github.com/andresoviedo/android-3D-model-viewer>)를 수정하여 구현하였다. 해당 프로젝트의 전체 코드를 이해한 후 수정한 내용은 다음과 같다:

* + 1. 전체 소스코드를 이해하고 필요한 부분을 수정하였다. OpenGL 공간에 치아모델을 초기 소스 그대로 불러오면 각각의 치아가 OpenGL 공간의 정 중앙에 겹쳐서 나타났다. 따라서 각각의 치아 개체를 정 중앙에 위치하는 대신 절대위치를 보존하도록 소스를 수정하였다. 다음은 치아모델이 너무 작아 Surface View에 빈 공간이 많았다. 따라서 Object 파일을 불러올 때 치아 모델을 약 1.7배가량 확대하여 불러올 수 있도록 소스를 수정하였다. 외에도 치아모델 파일 자체에 오류가 있어 자체적으로 수정하였다.
    2. 치아번호를 입력 받아 해당 치아를 채색하는 메서드를 구현하였다. 초반에 생성한 각각의 치아 모델을 배열에 저장해두고, 블루투스 통신으로 치아번호 값을 입력 받으면, 배열의 인덱스로 변환하여 해당 치아를 채색하는 방식으로 구현하였다.
    3. 양치 중인 치아가 보이도록 모델을 회전시키는 기능을 구현하였다. 초기에는 양치 중인 치아를 채색하여 나타내는 방식을 사용하였지만, 치아모델이 정면을 향하고 있어 뒤쪽 어금니 같은 경우에는 채색만으로 양치 중인 치아를 알아보기 어려웠다. 따라서 치아 번호를 입력 받아 해당 각도로 카메라를 이동시키는 메서드를 구현하였고, 후에 부드럽게 회전할 수 있도록 구현을 완료했다.

### 습관 분석

먼저, 모니터링을 하는 동안, 블루투스를 통해 값을 받을 때마다 양치 시간을 100ms만큼 더했고, 전달 받은 양치 번호에 대한 횟수도 추가한다. 압력에 대한 정보도 판단한다. 치아를 위/아래, 왼쪽 어금니/앞니/오른쪽 어금니 6구간으로 나누었다.

1. 점수

100점에서 시작하여 감점하는 방식으로 채점한다.

* 총 양치 시간이 2분 30초에서 3분30초 사이면 감점이 없고, 각 시간에서 10초씩 넘어갈 때마다 5점씩 감점한다.
* 압력이 작거나 큰 횟수 \* 3점만큼 감점한다.
* 한 구간의 양치 시간이 총 양치 시간을 6으로 나눈 값의 1.5배 이상/이하이면 5점씩 감점한다양치 시간: 양치 모니터링을 하는 동안의 값을 더한다.

1. 코멘트

* 총 양치 시간이 2분 30초에서 3분30초 사이면 적당한 시간동안 양치했다는 코멘트를, 2분 30초 이하면 양치 시간이 짧다는 코멘트를, 3분 30초 이상이면 양치를 오래 했다는 코멘트를 보여준다.
* 압력이 작거나 큰 횟수에 대해서도 양치를 살살 하세요, 골고루 양치하세요 등의 코멘트를 보여준다.
* 한 구간의 양치 시간이 총 양치 시간을 6으로 나눈 값의 1.5배 이상이면 그 부분은 많이 양치했다고, 이하면 적게 양치했다고 코멘트를 보여준다.

1. 주간 코멘트

지난주 양치 점수와 횟수를 비교하여 이번주 양치에 대해 분석한다.

1. 캘린더와 월간 코멘트

하루 양치 횟수를 동그라미의 개수로 표현했고, 한달 동안의 총 양치 횟수와, 하루 2회/3회 한 양치 일수를 보여준다.

### 아두이노를 이용한 치아 판별

칫솔 기능을 구현하는데 사용된 아두이노 보드의 종류는 일반적으로 많이 쓰이는 기본 보드인 우노 보드이며, 같이 사용된 주변 기기로는 자이로 가속도센서, 전동모터, 압력감지센서, 블루투스 모듈, 스위치, 저항, 브레드 보드, 시리얼포트 연결 USB이다.

우선 위의 하드웨어 조립과 같이 아두이노에서 해당 기능들을 수행할 수 있도록 회로를 구성해 두었으며, 코드상에서 어떤 포트를 이용하고 잇는지 알 수 있도록 해당 핀 번호들을 코드 상단에 정의 해 두었다.

사용된 헤더파일들은 math.h, Wire.h, SoftwareSerial.h 인데, math 헤더파일은 칼만필터를 구하는 과정에서 삼각함수의 측량을 위해 사용되었으며, Wire.h 는 센서들을 보드와 핀번호를 통해 연결해 줄 수 있도록 돕기 위해, 그리고 SoftwareSerial.h는 블루투스만의 시리얼을 연결 해 주기 위해 선언하였다.

아두이노 코드의 셋업 부분에서는 기본적으로 시리얼과, 블루투스 시리얼의 통신속도를 9600으로 맞춰주는 작업과 자이로 가속도 센서가 회로를 구성한데로 잘 꽂혀 있는지 확인하며, 확인이 끝난 후에는 다른 센서들을 pinMode 함수를 통해서 값을 입력해준다.

또한 LUT를 선언하여 0부터 27까지의 배열에 각각의 치아번호를 입혀주었다.

루프 부분에서는 자이로 가속도 센서를 통해 칼만필터를 적용하는 작업과 작업된 값을 계속 전달해주는 역할을 계속 수행하며, -16383~16383 사이의 값으로 매핑 되는 값을 -200~200 사이의 값으로 변환하여 작고 미세한 값들을 버리도록 구현하였다.

또한 압력센서를 통해 일정 값이상의 센서 값이 들어오지 않으면 양치를 하고 있지 않는 상태로 판단하여 그 값을 버리는 값으로 판단한다.

스위치를 누르면 LUT애 0부터 입력할 수 있으며, 27까지 총 28개의 치아의 값을 입력 받을 수 있다. 마찬가지로 해당 모든 치아들의 정보가 입력되지 않는다면, 그 전까지는 블루투스로 치아번호를 전송하지 않으며, 모든 정보가 입력되었을 떄 부터 해당 치아번호와 Checksum 값을 블루투스로 보내게 된다.

치아 판별의 중요한 요점이 되는 부분은, Yaw값과, accelY 값으로 해당 값으로 각각 좌우, 위아래 판별에 도움을 준다.

값비교를 통해 센서 값의 오차가 1이하인 경우에만 해당치아로 판단하며, 엉뚱한 값이 들어가는 경우를 제외하기 위한 예외처리로 판단하면 된다.

루프부분 아래쪽에는 자이로 가속도 센서의 값을 받아와서 수행하는 연산인 예측과 업데이트에 대한 칼만필터 이론의 연산들이 추가되어 있다.

# 프로젝트 결과

## 프로젝트 완성도

|  |  |  |  |
| --- | --- | --- | --- |
| **기능** | **제안** | **구현** | **결과** |
| 주간 습관분석 | 해당 주차 양치 점수 분석  해당 주의 경향성을 고려한 분석 멘트 출력  일별 양치 기록과 코멘트 출력 | 해당 주차 양치 점수 분석  해당 주의 경향성을 고려한 분석 멘트 출력  일별 양치 기록과 코멘트 출력 | 완료 |
| 캘린더 기능 | 한 달의 치아 기록을 캘린더에 표시  하루 양치 기록 개수만큼 아치 아이콘 삽입  사용자의 평균 양치 횟수 출력 | 한 달의 치아 기록을 캘린더에 표시  하루 양치 기록 개수만큼 동그라미 아이콘 삽입  사용자의 평균 양치 횟수 출력 | 완료  미미한 변경사항 |
| 양치 모니터링 | 치아 모델 터치 시 모니터링 시작 및 종료  치아 모니터링  양치 시간 측정  간단한 치아상식 섹션  모니터링 종료 시 양치 분석 후 기록 저장 | 치아 모델 터치 시 모니터링 시작 및 종료  치아 모니터링  양치 시간 측정  양치 시간 알림  모니터링 종료 시 양치 분석 후 기록 저장 | 완료  기능 추가 |
| 치아모델 삽입 | 치아 모델 삽입  양치 중인 치아 실시간으로 하이라이트 | 치아 모델 삽입  양치 중인 치아 실시간으로 하이라이트  양치 중인 치아가 보이도록 치아모델 회전 | 완료  기능 추가 |
| 전동칫솔 제작 | 양치 중인 치아 판별  HM-10 블루투스 모듈을 이용한 블루투스 통신  전동칫솔의 진동에 대한 오차 제어  너무 강하거나 약한 칫솔 압력 체크 | 양치 중인 치아 판별  HC-06 블루투스 모듈을 이용한 블루투스 통신  진동에 대한 오차 제어 실패  너무 강하거나 약한 칫솔 압력 체크 | 90% 완료  미미한 변경 사항 |

## 한계 및 개선 방안

### 한계

정해진 범주 내에서는 정확하게 치아를 판별하지만 저가형 자이로 가속도 센서 하나만을 사용하여 인식할 수 있는 각도의 범위가 제한적이다. 따라서, 미리 설정한 각도와 거의 정확하게 일치할 경우에만 해당 치아를 판별 할 수 있어 예외 상황에 약하다. 그리고 전동 칫솔이므로 칫솔 자체에 진동이 존재한다. 약한 진동의 경우, 진동센서와 자이로 가속도 센서를 띄어 놓고, 칼만필터를 사용하는 등의 방법으로 값의 오차를 줄일 수는 있다. 하지만, 강한 진동의 경우 인식하는 값이 가만히 있는 경우에도 크게 달라져 제어가 불가능하다.

### 개선방안

고성능 센서를 사용함으로써 값을 더 정밀하게 측정할 수 있고, 오차를 감소할 수 있다. 그 외 위치를 판별할 수 있는 센서를 추가하여 정확도를 개선할 수 있다. 진동의 경우 실제 전동칫솔이 진동하는 방식과 다르다. 진동으로 변하는 값의 범위를 분석하여 진동을 제어하는 방법을 고안하고 실제 전동칫솔이 진동하는 방식으로 테스트를 해야 한다.

## 추후 개발 계획

양치 중에 카메라를 이용해 얼굴을 인식하는 방식을 추가하여 추후 개발할 수 있다. 사용자의 얼굴을 인식하여 양치중인 구획을 판단하고 센서 값 정보와 함께 실시간 모니터링에 사용할 계획이다. 얼굴 인식을 추가하면, 자이로 가속도 센서 하나만을 이용할 때보다 좀 더 세밀하게 치아를 판별하고 노이즈를 제거할 수 있을 것이다. 양치 중에 휴대폰을 거치하는 것이 번거로울 수 있으므로, 사용자에게 칫솔만을 사용할 것인지, 카메라도 함께 사용하여 모니터링을 진행할 것인지 선택권을 부여하는 방식으로 진행할 계획이다. 얼굴인식에 선택권을 부여하여 기능을 추가한다면, 좀 더 사용성이 높아지고 정확도가 높아질 것이다.

현재 치아 하나당 한 면 만을 양치한다는 가정에 치아를 판별한다. 따라서 하나의 치아를 양치할 때 닦을 수 있는 면의 경우의 수를 추가할 수 있도록 하거나 구하는 방법을 고안하여 치아 안팎을 판별하는 기능을 추가적으로 구현할 계획이다. 그리고 그에 따라서 치아 모델에도 현재 양치중인 치아의 부분도 구별하여 하이라이트를 하는 등 치아 하이라이트 기능도 개선할 수 있다. 다음과 같은 기능을 추가적으로 개발한다면 보다 정밀하게 모니터링을 할 수 있고, 사용자에게 좀 더 자세하게 양치에 대한 정보를 제공함으로써 좋은 양치 습관을 형성하는 데 도움을 줄 것이다.

## 후기

**김수진**

주로 치아 모니터링에 사용되는 치아모델과 관련한 부분을 구현하였다. OpenGL을 다뤄본 경험이 없어서 처음부터 공부를 해야 했다. OpenGL Document를 읽으며 예제를 만들어보면서 단순한 3D 개체를 OpenGL Surface에 나타내는 데 필요한 기본적인 행렬 연산에 대해서 공부하였고 3D 모델링 툴을 다뤄보면서 용어를 익혔다. 치아모델을 애플리케이션에 삽입하는데 큰 도움이 되었다. 또한 오픈소스 프로젝트를 사용하면서 버그를 발견하여 GitHub 오픈소스 프로젝트에 Issue를 등록하는 경험도 하게 되었다.

**남유선**

애플리케이션 UI와 기능, 오류 수정과 관련된 부분을 주로 구현했다. 안드로이드 앱을 개발한 경험이 별로 없을 뿐만 아니라 Kotlin이라는 새로운 언어로 개발하는 것이 어려웠다. 특히, 습관분석 탭에서 날짜별로 습관분석을 하는 부분이 Expandable ListView 안에 Expandable ListView를 넣는 방식으로 했는데 자료가 별로 없을 뿐만 아니라 모두 Java로 된 자료여서 Kotlin에 맞게 바꾸는 데 오래 걸렸다. 그 외 블루투스와 아두이노를 사용하는 방법 등 많은 것을 배울 수 있었다. 호흡이 긴 프로젝트를 처음으로 해 봤는데 계획 수립하는 것이 중요하다는 것을 느꼈고, 한 학기동안 힘들었지만 새로운 것을 경험할 수 있었다.

**박주현**

초반에 어플리케이션 캘린더 파트를 진행한 후 대부분의 시간을 칫솔을 구현하는데 대부분의 시간을 할애했다. 언제나 있는 기능을 가져다 쓰기만 했었는데 처음으로 캘린더뷰를 커스텀해서 사용했다는 점에서 발전함을 느낄 수 있었고, 기존에 아두이노로 수행했던 작업들은 언제나 단순한 작업이나, 민감하지 않은 작업들이었는데 처음으로 민감한 작업들을 수행하면서 어려움을 많이 겪었다. 역시 하드웨어는 생각했던 것 보다 제어가 쉽지 않다는 것을 다시금 느낄 수 있었고, 그만큼 더 많은 테스트와 연습이 필요하다는 것을 느낄 수 있었다. 또한 데모 때 갑작스러운 오작동으로 실행이 안된 것을 교훈으로 삼아서 다음 캡스톤 설계 수업에서는 좀 더 신중하고 정확하게 준비할 수 있도록 마음 먹은 계기가 되었다.