5) 어플리케이션

5-1) 개발환경

Swift5를 활용한 SwiftUI + UIKit 기반의 iOS 앱으로 제작했습니다. target version은 13.0으로 설정하였으며, IDE는 Xcode 11.5를 사용했습니다. 테스팅은 iPhone 7을 통해 진행하였습니다.

5-2) 구성

Model은 Subject, View는 Home, Group, Calendar, Camera 총 4개로 구성되었으며, Controller는 Camera, Calendar 총 2개로 구성하였습니다. 상태 기반의 SwiftUI를 바탕으로 이벤트반의 UIKit을 활용한 오픈소스를 활용함에 따라 혼합된 형태로 구성하게 되었습니다. 2019년 출시된 SwiftUI는 모든 Apple 플랫폼에서 UI를 구축할 수 있는 방법으로 소개되었지만, 아직까지 제대로 자리를 잡지 못하고 일부 기능이 지원되지 않아 Controller를 받아 사용하는 형태가 주로 사용되고 있습니다.

5-3) 기능

기능은 크게 기록과 공유, 두가지로 나누어 구현하였습니다. 먼저 기록의 기능으로, 공부화면 인식을 통한 실질적인 공부 시간을 기록하고 사용자가 과목별로 이를 관리할 수 있도록 구현하였습니다. 또한 공유 기능으로는 스터디 그룹을 생성하고 관리하도록 하였습니다. 이를 통해 사용자가 언택트 환경에서도 자기 통제력을 강화하도록 하였습니다.

5-4) 핵심 기능 구현

Server-Client 구조로 OpenPose를 활용한 공부자세 판별하는 모델과 휴대폰이 통신할 수 있도록 구현하였습니다. HLS(HTTP Live Streaming)을 활용하여 스트림파일을 전송하였으며, 그의 결과 값으로 Socket 통신을 통한 Json 형식의 데이터를 받아 파싱하여 분류에 활용하였습니다. 이 값을 바탕으로 공부시간을 측정하였고, 사용자가 이를 인지하도록 화면에 표현하였습니다. 또한 임계치를 위하여 초반 15초간의 촬영은 학습시간에 포함되지 않고 스트림을 통해 전송하였습니다.

5-5) 의견

AI 프로젝트를 온라인으로 진행하여 의사소통에 한계가 많았고, 원격접속으로 인한 자원의 제약사항이 많았습니다. 또한 기간이 2주나 줄어듦에 따라 앱의 완성도가 높지 못한 점이 아쉬웠습니다. 또한 iOS 앱 프로그래밍과 Swift를 처음 활용해보는 프로젝트라 생각치 못한 부분에서 어려움을 겪었습니다.  
 또한 충분한 테스트를 통한 수치화나 로컬환경이 아닌 실 환경에서 진행해보지 못해 아쉬웠습니다.

5-6) 참고

5-6-1) Haishinkit 1.0.9 (<https://github.com/shogo4405/HaishinKit.swift>)  
 Camera Stream을 위한 오픈소스 기반 라이브러리로 예제 코드를 기반으로 필요한 부분만을 활용하였습니다. 최근까지 업데이트가 이루어져있어 활용하였습니다.

5-6-2) FSCalendar  
 Calendar를 위한 오픈소스 기반 라이브러리로 다양한 형태의 달력과 모션을 기본으로 제공하여 커뮤니티에서도 활발하게 활용되는 라이브러리입니다.

