



# Weekly Updates

## Slow-Pitch Softball

SDMAY25-11





# Weekly Contributions



**01**

**Ethan**

Testing Videos, CMake installation methods

**02**

**Sully**

Flutter navigation, Flutter C++ integration

**03**

**Andrew**

C++ mobile development

**04**

**Cameron**

Research

**05**

**Casey**

Camera page and testing

**06**

**Josh**

Research





# Ethan

- Gathered research data for testing detection protocols using gridded videos filmed with a GoPro and tripod for validation.
- Began researching multithreaded approaches for layered detection using OpenCV, KCF, and YOLO to detect softballs.
- Working on creating a CMake toolchain for automatic installation of dependencies for Android and iOS.





# Sully

- Researching/Attempting C++/iOS integration





# Andrew

- Successfully implemented object detection and tracking code in C++ with improved reliability and no FPS impact.
- Began testing the code on mobile, but faced issues with OpenCV framework compatibility for iOS and Android.
- Discovered that the required OpenCV framework versions for mobile lack tracking algorithms; building the framework from source failed due to a bug in the latest OpenCV release.





# Casey

- Implemented basic screen designs for the final Flutter app, including a camera screen for viewing softball pitches and configuring pitch detection options, as well as settings and past pitch screens.
- Tested the app functionality on an iPhone to simulate the user experience.
- Reviewed previous mockups and began integrating those designs, with a focus on smoothly integrating the C++ solution into the app.

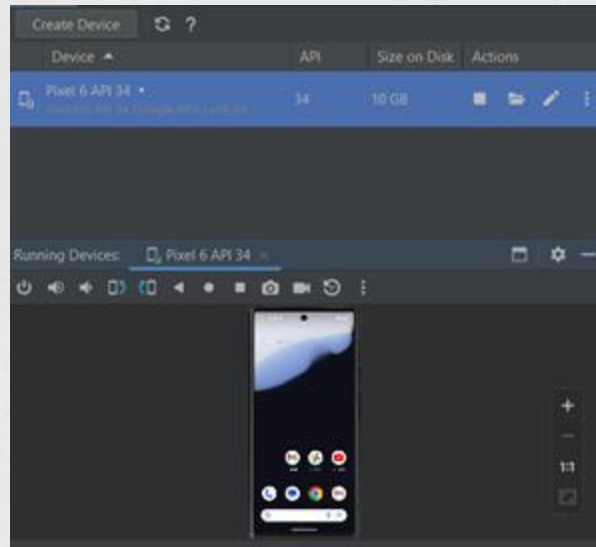
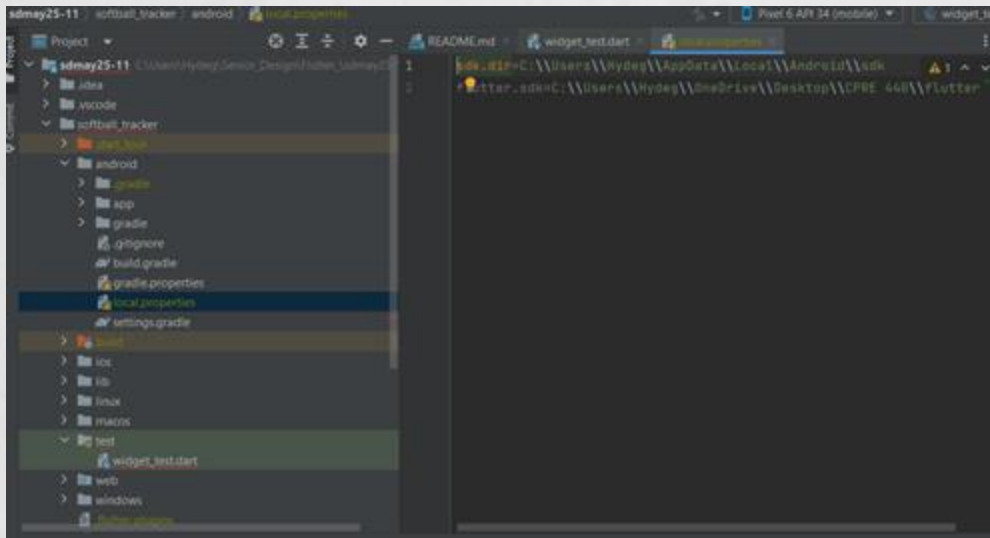




# Josh

- Researched and tested potential solutions for integrating the Android side of the app with the existing Flutter app functionality from iOS.
- Investigated previous height detection methods used in similar projects to verify the completeness of the current height detection models.









# Cameron

- Continued researching how to run C++ code on iOS and found that recent updates to Swift make it easier to integrate with C++.
- Discovered that including header files in Swift allows direct calling of C++ functions.
- Plans to test this approach on their iPhone next week and expects minimal difficulty in integrating the C++ code if the findings are accurate.





# Next Week...

- CMake toolchain and project setup
- C++ translation/development
- Continuing Flutter screen development

