Advanced Android Motion Detection App

Number of teams you can accommodate: max 6
Number of students per team: 3-4

Industry Challenge Background

The rapid rise of Android and iOS devices has made mobile technology a major part of everyday life, and as the power and pervasiveness of smartphones has continued to grow, many in the defense industry have started viewing Android-based devices as viable low-cost solutions for meeting high-priority military needs. One such area of growth in the DoD is Imaging, Surveillance, and Reconnaissance (“ISR”)—an area in which BAE Spectral Solutions in Honolulu specializes. This Industry Challenge will give students the opportunity to learn how to develop a real Android app, while also serving as a primer for working with some of the real-world issues that Spectral Solutions engineers encounter as they develop real-time ISR software products to help America’s warfighters.

Materials Needed

- Any Android device with a camera that is running Android 4.0 or later
  - A late-model device with a dual-core processor or better is strongly recommended (2013 models or later)
- A PC that is capable of installing and running the Android SDK

Overview & Guidelines

The team is challenged with building an Android application that will be able to perform motion detection on the video imagery acquired from the built-in camera of the device.

BAE Systems will provide a barebones software code skeleton that provides the foundation upon which the team will build the application. This will be very similar to the Android SDK that is published by Google.

Operational Concept

Our goal is to provide a soldier on the ground with a “second set of eyes” via an Android device. The app the team will be designing and building is the critical component of this tool, as it will essentially function as the brain of the system.

There are many scenarios (“use cases”) in which this device is envisioned to be useful in the field, such as:

- **Example Use Case 1:** A soldier is guarding an entryway, and uses the device to monitor a blind corner behind him. Since he cannot see anything coming from that direction, he needs the motion detector to alert him of any possible threats from this direction.
• **Example Use Case 2:** A soldier is viewing an area of dense vegetation where it is extremely difficult to isolate objects from their surroundings. She needs the motion detector to allow her to more easily view even the slightest movements in the area as an aide for finding her targets.

• **Example Use Case 3:** A soldier is tasked with monitoring the security of a civilian road from a station that gives him a birds-eye view of the road. He needs the motion detector to alert him of any vehicular traffic over a certain size, as he needs to then confirm that the vehicle is not a threat.

**Tasks & Features to Implement**

Based on the example use cases defined above, along with some added performance requirements, we have derived the following requirements/features for the app that is to be built:

• **Core Features:**
  1. Upon launch, the main screen of the app should show a live feed of the camera, with icon and informational overlays on that image. This layout should feel very similar to the usual Camera/Video Camera apps in both Android and iOS
  2. As one of the overlays on the image, the app should draw a box around the largest moving object it detects in the picture. If no motion is detected, a box is not displayed.

• **Bonus Features:**
  1. [Difficulty: Low] Display the number of frames per second being processed
  2. [Difficulty: Low] Allow the user to filter out moving objects that are smaller than X pixels wide or tall, where X is a configurable value
  3. [Difficulty: Med] Provide a toggle switch that allows the user to flip between the “regular” camera view, and a “motion only” view that shows only the parts of the image in which it detects motion. The toggling of this feature should perform similarly to toggling on/off the “map” vs. “satellite” layer view in Google Maps. Note that the “motion only” view can be just a black and white pixel map—black representing pixels in which no motion is detected, and white representing pixels in which motion is detected.
  4. [Difficulty: High] [Prerequisite: displaying the number of FPS] In most cases, the FPS number will drop as more motion is detected in the frame. Optimize your motion processing algorithm so that it maintains 30 FPS or higher (“real time”) regardless of the amount of motion that is detected.
  5. [Difficulty: High] Automatically detect whether the objects in the image are actually moving, or if the motion detection is a result of the camera itself being moved.

Only the two “Core Features” defined above are required to be implemented for the app to be considered complete.

After completing the “Core Features,” the team is free to develop any/all of the “Bonus Features” and/or any additional features they can conceive. The additional features can be based on the example use cases we have defined above, or any other brainstormed use cases. Creativity and innovation is encouraged.
Rules & Guidelines:

- Teams are free to use any resources available, including code, libraries, and development packages that are available online.
- The app needs to only work on one Android device that is chosen by the team. This can be a phone or a tablet; there is not a requirement for the app to run on multiple devices.

Evaluation Criteria

- Teams should be prepared to present their app in a sales pitch to judges that is similar in format to the presentations on the Shark Tank TV show.
- The goal of the sales pitch is to demonstrate the app the team has created, and more importantly, for the team to convince the judges that the app they have developed is the one that they should choose to entrust with the lives of our soldiers in the field. Show and tell the judges what makes your app outstanding, and essentially convince them of why your app is the one they should “buy.”
- The app with the best combination of performance, usability, and features, will be selected as the winner. Creativity and showmanship during the sales pitch will undoubtedly help as well.

Suggestions

- Before developing a feature, consider also how easily the functionality can be demonstrated in a live “Shark Tank” style demo, and the logistics of how you will actually demonstrate the functionality during your sales pitch.
- When using online resources, especially open source code, be very careful in inspecting and analyzing the packages. Make a conscious trade-off between the effort required to understand and integrate unknown code from the internet vs. developing it yourself.
- Don’t stress about the “Shark Tank” demo. It is intended to be fun, and it will be a lighthearted environment that is absent of the pressure that is intentionally manufactured on the TV show.