PSU CapStone Project Proposal

Tuesday, October 7, 2014 8:23 AM

Status

- CapStone Project entered into PSU system
    - Prototype Genymotion style emulator with sensor faking extensions to the HAL
    - Students get to choose, so we'll find out if we are chosen
    - A few meetings, some minor equipment (less than $500) and we have it all on hand now

Android Emulator Extension Project

Problem Statement:

In today's Android App development community, the typical app developer must choose between:

- Purchasing and maintaining a growing collection of physical mobile devices
- Outsourcing testing to a testing lab with a large collection of physical mobile devices
- Utilizing a web/cloud testing service who have & maintain a large collection of physical mobile devices

All 3 of these alternatives can quickly become expensive, especially for the smaller app development companies (12 people or less). This project is an exploration into the possibilities of utilizing the SW Android emulators to perform more work than they do today but in an automated fashion. If this is possible, the smaller app development companies may be able to reduce their costs associated with physical devices.

Goals:

The project goals are to create 3 prototypes (1 each for Windows, Mac, and Ubuntu):

- Research and select a virtual machine platform for each host OS
- Create a reliably functioning Android VM on each hose using either a public x86 Android distribution (e.g. www.android-x86.org/ or an Intel Android distribution)
- Add SW to the VM (e.g. services, apps, drivers, etc.) as needed to emulate sensor chips and the data that they return (at least GPA and accelerometer). In this case, emulation means to insert SW between the HW sensor and the OS. This SW should intercept the HW data from the sensor and replace it with data from another source (e.g. SD card).
- Add SW to the system to allow programmatic control of the fake sensor data (e.g. on/off)

The successful prototype would be able to fool an app that uses GPS data into thinking it's in Cairo, then Tokyo, then Paris without leaving town. With programmatic controls, a test developer could create an automated test that uses SD data to test how an app responds to sensor data.

Other sensors can be prototyped as well. This includes Hall, accelerometers, heat, fingerprint, camera, etc. The ability to fool multiple sensors at one time is highly desired but not required.