# Lab 7

## Mobile Sensors

In this lab you will practice handling data from mobile device sensors in JavaScript.

### Step 1: Explore and inspect device sensor capturing

In this lab you will create simple applications derived from the following Codepens that exemplify capturing data from different mobile devices:

- Geolocation
- Multi-touch
- Orientation
- Motion
- Camera

You will need to test these applications on a mobile device (smartphone or tablet) while still developing on your laptop. First fork (this is at the bottom right of the screen in Codepen) all of the Codepens linked above into your Codepen account. Then open a Chrome browser on your mobile device and log into your Codepen account. One by one, open the applications and test them to see what they do. The regular Codepen rendering will likely not work; instead choose "Actions->Debug View" to actually access device sensors. You might need to re-orient or move the device to see an effect. You may have different results depending on your device - if possible explore with tablet, phone, laptop, etc.

In the rest of this lab you will implement functionality building on these demos.

First you will use the camera to implement two simple functionalities:

- Determine light level in the environment
- Detect when the camera is blocked (i.e. image is almost completely black)

Implement these functionalities as a Codepen pen, possibly reusing parts of the camera demo linked above. To demonstrate that the functionality is working, include a light level indicator (e.g. "Current light is N%) and a separate 'black image' indicator on your interface (e.g. "Camera blocked? YES"). Make sure your interface clearly instructs the user about how to test it.

## Step 3: Multi-touch functionality

Next you will use the multi-touch sensing capability to implement **one** new interaction functionality of your choice. Here are some example functionalities you can choose from:

- Detect a set of single or multi finger touch *gestures*
- Draw a picture, color a black-and-white picture, or annotate an image using a single finger
- Draw certain shapes with multiple fingers (e.g. triangle with three fingers, axis-aligned rectangle with two fingers)
- Select an item from a set by drawing a circle around it
- Jointly select multiple items by touching all of them

Implement your functionality as a Codepen pen, possibly reusing parts of the demo linked above. To illustrate your functionality attach a response (e.g. text on the screen, change of background color) to the events you are detecting. Make sure your interface clearly instructs the user about what the functionality is and how to test it.

Next you will use the motion and/or orientation sensing capability to implement **one** new interaction functionality of your choice. Here are some examples:

- Detect when a device is dropped
- Determine how fast the device was moved within a certain time window (e.g. punchmeter / app)
- Detect when the device is lying still on a surface for a certain duration and whether is it face down or up
- Detect when a person might be viewing the phone
- Detect gestures drawn in the air while holding the phone (e.g. shake side to side)
- Determine current pose of the phone given a known starting pose
- Count number of steps, stair climbing steps, jumps, or other countable movement activities

Implement your functionality as a Codepen pen, possibly reusing parts of the demos linked above. To illustrate your functionality attach a response (e.g. text on the screen, change of background color) to the events you are detecting. Make sure your interface clearly instructs the user about what the functionality is and how to test it.

## Optional

For any of these steps, you may setup a python server to accomplish any of the backend work to achieve the desired, returned result to the User. For example, you may send accelerometer data to a python server that then implements your step counting algorithm from lab 2, and sends these results back to your front end application to be displayed.

## Step 5: Submit your code on Canvas

Complete this lab by submitting a public link to your Codepen pens or projects on Canvas, by Nov 24 Tuesday, 11:59pm. We will test the camera functionality on a laptop and the other two functionalities on two mobile devices (Android tablet and iOS phone) in debug mode of Codepen, interacting with the rendered pages to test the implemented functionality, making sure:

- The camera application correctly indicates changes in brightness and detects 'black image'
- Multi-touch functionality is clearly described and works as intended
- Motion/orientation functionality is clearly described and works as intended

We will inspect code as needed. See Canvas for a grading rubric.