Multi-touch app development with modern web tools

David Reagan, Advanced Visualization Lab
Advanced Visualization Lab

- A unit of the Research Technologies division of UITS
- Research Technologies is a PTI Cyberinfrastructure and Service Center
- "...to promote the innovative application of visual technologies to advance Indiana University’s missions in research, education, creative activity, and community outreach"
Touch hardware

- IQ-Table
- IQ-Tilt
- Touch monitors
- Tablets
IQ-Table v2

- 55” monitor in table orientation
- Quad-HD resolution
  - 3840 pixels by 2160 pixels
- Capable of 32-point multi-touch
- Built-in Windows PC
- Custom frame, legs, skirt
Early touch projects

- Brownsburg Challenger Center
- Places & Spaces: Illuminated Diagrams
- Conference kiosks
- Stereoscopic video kiosk
- Digital signage
- Digital Golden Book
- Digital *On the Road* manuscript
Collection Viewer

• Developed for the 2012 Smithsonian Folklife Festival, reused many times since

• Simple, multi-user interface to digital media collections

• Flash application using the Open Exhibits framework

• Expanded to serve as interface to 2nd display
So why are we talking about web technology?
Why not Open Exhibits?

- Limited lifespan for Flash, especially on the web

- Philosophical choice: configuration over code
  - Build app once, let community customize through configs
  - Complex configs $\rightarrow$ AVL still configured apps
  - Feature creep as clients request apps that were similar to, but not exactly, the existing software
  - Ultimately, we didn't save much time
Why web tech for touch applications?

• Recent maturation of HTML/CSS/JavaScript
  • Including tools for entire workflow (more on this later)

• Accessible to most developers

• Deployable to most platforms

• Design and content freedom
HAMMER.JS
You can touch this
Hammer.js  https://hammerjs.github.io/

“open-source library that can recognize gestures made by touch, mouse and pointerEvents”

• Provides recognizers for Tap, Press, Swipe, Pan, Pinch, Rotate
• Options to create custom recognizers like 2-finger pan or quadruple-tap
• Also available as jQuery plugin or Angular directive
• Emulator for non-touch devices
AVL development workflow
Development tools: Git + GitHub

“Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.”

GitHub: “Powerful collaboration, code review, and code management”

3 Git(Hub) talks at SWIT 2015
Development tools: Node.js

“a JavaScript runtime built on Chrome's V8 JavaScript engine”

• JavaScript as a server-side language
  • Developer-side for us

• npm: “largest ecosystem of open source libraries in the world”
  • Package manager for our development tools
Development tools: gulp

Automate development tasks, similar to *make*

- **gulp serve**: launch Browsersync server on source files
- **gulp test**: run unit tests with Karma
- **gulp protractor**: run e2e tests with Protractor
- **gulp build**: build optimized application for deployment
Front-end tools: Bower

“A package manager for the web”

• Specify frameworks, libraries, etc in config file
• `bower install`: download local copies
• `gulp` task to automatically inject includes into index.html
Front-end tools: AngularJS

- Framework for creating dynamic single-page web applications
- Maintained by Google and others
- Model-View-Controller architecture
- Declarative programming for UI, imperative for logic
Development tools: Yeoman

“The web's scaffolding tool for modern webapps”

• Generators to help you assemble new projects
• Automatically configure your build system, package manager, and more
• We use generator-gulp-angular
Example step 1: AngularJS ng-repeat

```html
<div>
  <md-card ng-repeat="awesomeThing in main.awesomeThings">
    <img ng-src="assets/images/{{ awesomeThing.logo }}" alt="{{ awesomeThing.title }}"/>
  </md-card>
</div>
```
Example step 2: Angular Hammer

Hammer gesture recognizers as Angular directives

```html
<md-card
  hm-panmove="card.touchThis"
  hm-pinchmove="card.touchThis"
  hm-rotatemove="card.touchThis"
  hm-panend="card.cantTouchThis"
  hm-pinchen="card.cantTouchThis"
  hm-rotateend="card.cantTouchThis"
  ondragstart="return false" />

<img ng-src="assets/images/{{ card.thing.logo }}"
  alt="{{ card.thing.title }}"
  ondragstart="return false" />

```
Example step 3: Angular Material

- Implementation of Google's Material Design in Angular
- We primarily use it for the touch ripple, but also useful for layout, UI components, etc

```html
<div>
  <avl-card ng-repeat="awesomeThing in main.awesomeThings | filter:main.query"
    thing="awesomeThing"></avl-card>
</div>

<div style="padding: 20px">
  <h1 class="md-title">Filter</h1>
  <div layout="row">
    <md-input-container>
      <label>Query</label>
      <input ng-model="main.query">
    </md-input-container>
    <md-button ng-click="main.query = ''">Clear</md-button>
  </div>
</div>
```
Example step 4: Angulartics

• Use familiar web analytics for user tracking
• Plug-ins available for >20 analytics vendors
• Automatic page view tracking
• Declarative event tracking within a page

```
<md-button ng-click="main.query = '\'"
  analytics-on="click"
  analytics-category="Filter"
  analytics-label="Clear">Clear</md-button>
```
Right now 2 active users on site.

Event Category | Event Action | Active Users | Active Users with Events
--- | --- | --- | ---
FILTER | CLEAR | 2 | 2 (100% of total)
Deployment
Deploy to desktop: browser

Best results with Chrome

• Set `chrome://flags` options
  • `Overscroll history navigation` = Disabled
  • `Enable viewport meta tag` = Enable
• Kiosk extension
• Lockdown application
• Disable right-click, charms, etc in Windows
• Run simple webserver for local file access
Deploy to desktop: standalone executable

• NW.js or Electron

• Bundle your webapp with node.js and Chromium

• Yeoman generators available for both
  • Swap sample app with your own
Deploy to mobile

• Deploy to mobile browsers over the web

• Bundle your webapp into a mobile app for Android, iOS, etc with Apache Cordova
Example app GitHub repo: http://go.iu.edu/LUA

dmareagan@iu.edu
vishelp@iu.edu