Gaia

Technical Walkthrough

Introduction

FirefoxOS enables the Open Web as a platform for mobile devices.

A mobile operating system that is:

- Free from proprietary systems
- Opportunistic for web developers
- Flexible for consumers

Technical Overview

What we'll be going over.

- Gaia environment setup and debugging
- Applications and source tree
- Building an application
- System APIs
- Workflow and testing

Environments

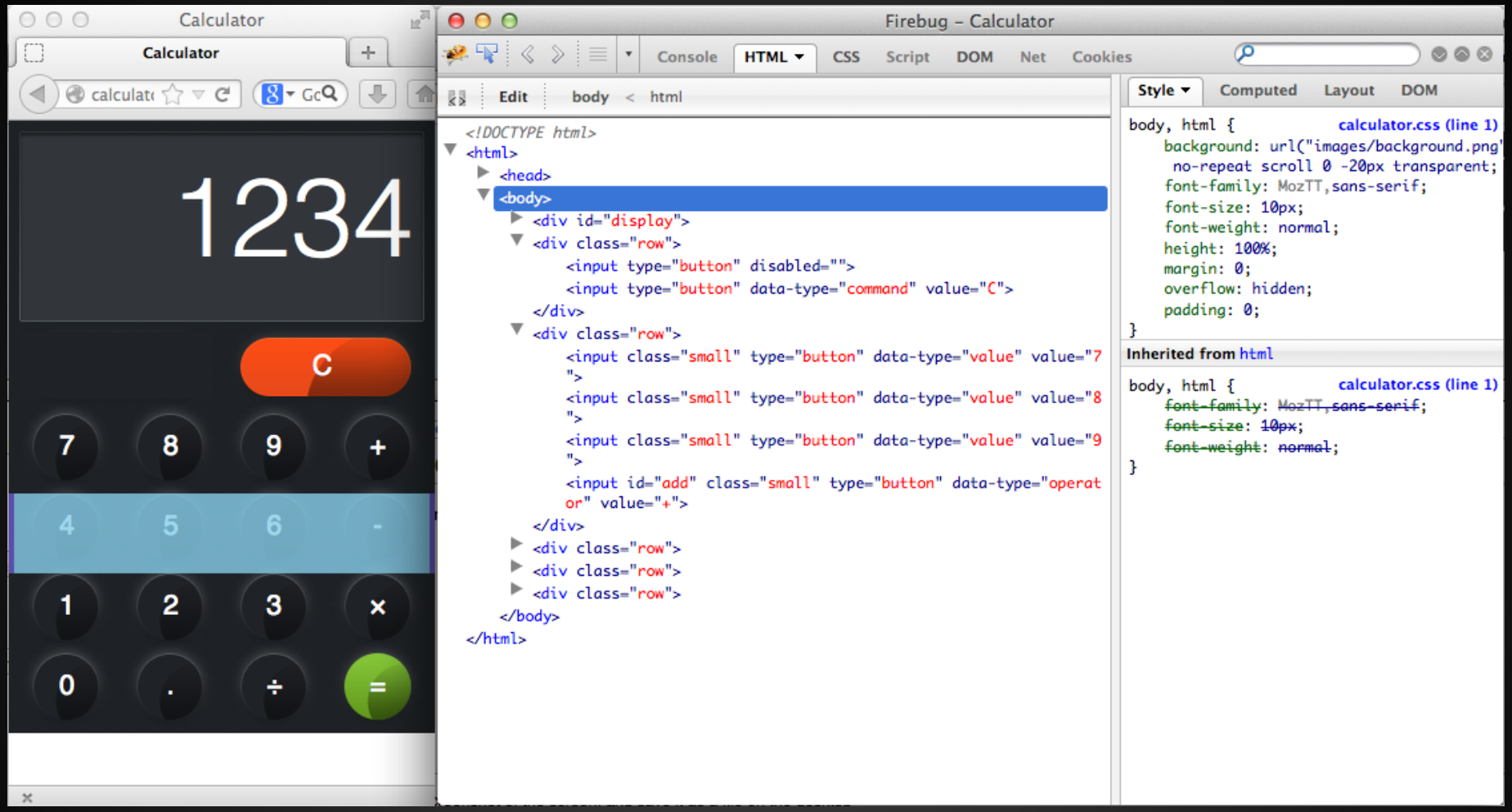
Firefox Nightly

Run apps directly in the browser. Fastest to develop, least like production.

1. Clone and make gaia
2. Run: <path-to-nightly>/firefox -profile profile/

Development Tools:

- Firefox debugger
- Firebug



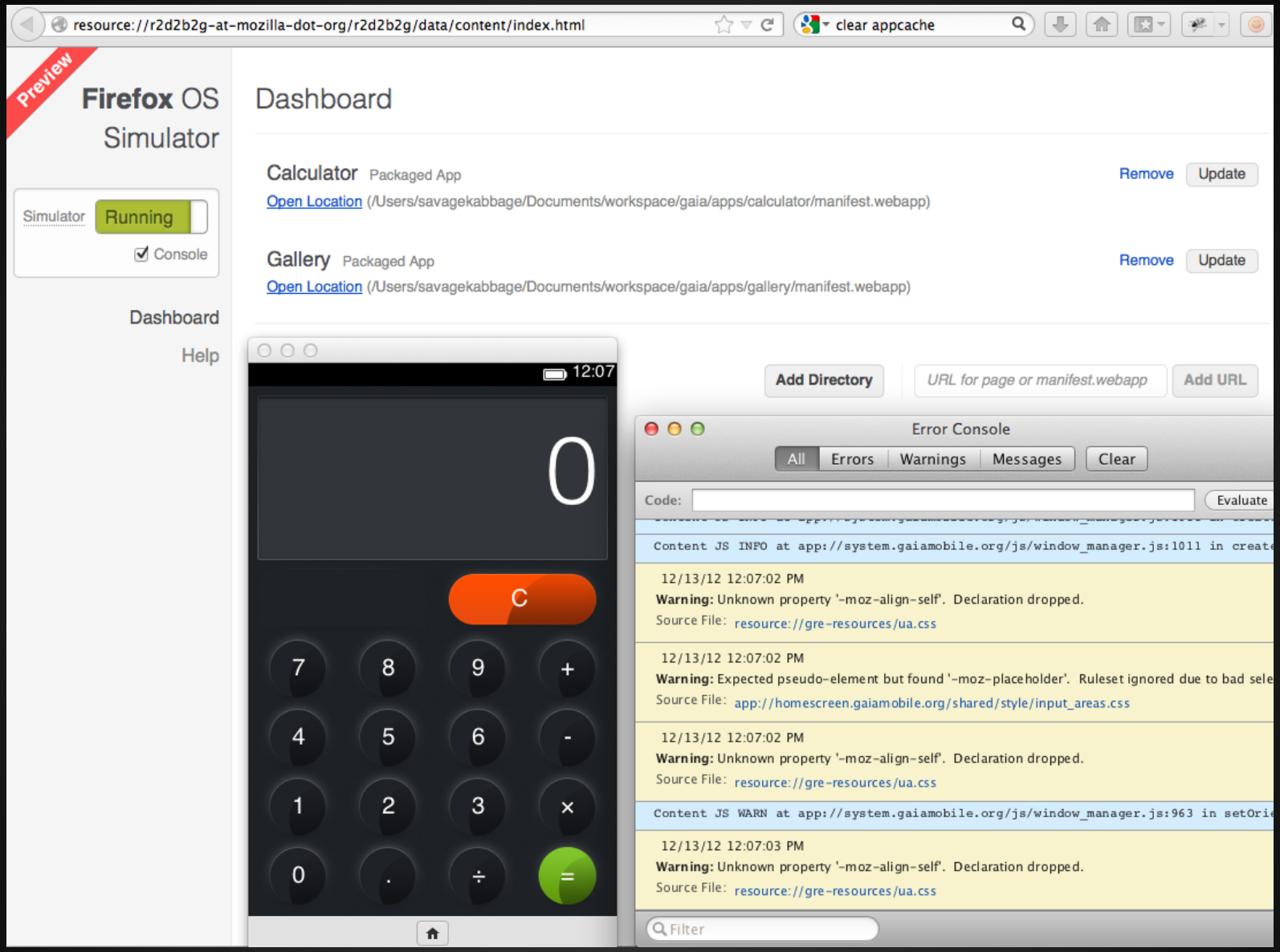
App running inside of Firefox Nightly.

FirefoxOS Simulator (r2d2b2g)

Runs as a plugin within Firefox.

Install the toolbar, point to a manifest.webapp.

https://addons.mozilla.org/en-US/firefox/addon/firefox-os-simulator/



FirefoxOS Simulator

B2G Desktop

An emulator which mimics FirefoxOS on your desktop. Mock interfaces such as mozKeyboard, mozSettings.

Nightly, or build your own (mozilla-central mercurial repository).

Using B2G Desktop

Install & Run

1. Download/Build B2G Desktop
2. Clone Gaia from Github
3. Make
4. <path-to-b2g-bin>/b2g -profile profile/

Development Tools:

- -jsconsole - Opens a standard javascript console along with the emulator.

FirefoxOS Device

A device flashed with FirefoxOS software. It is possible the flash FFOS onto some devices running Android 4. (Nexus S, Galaxy S2).

Development Tools:

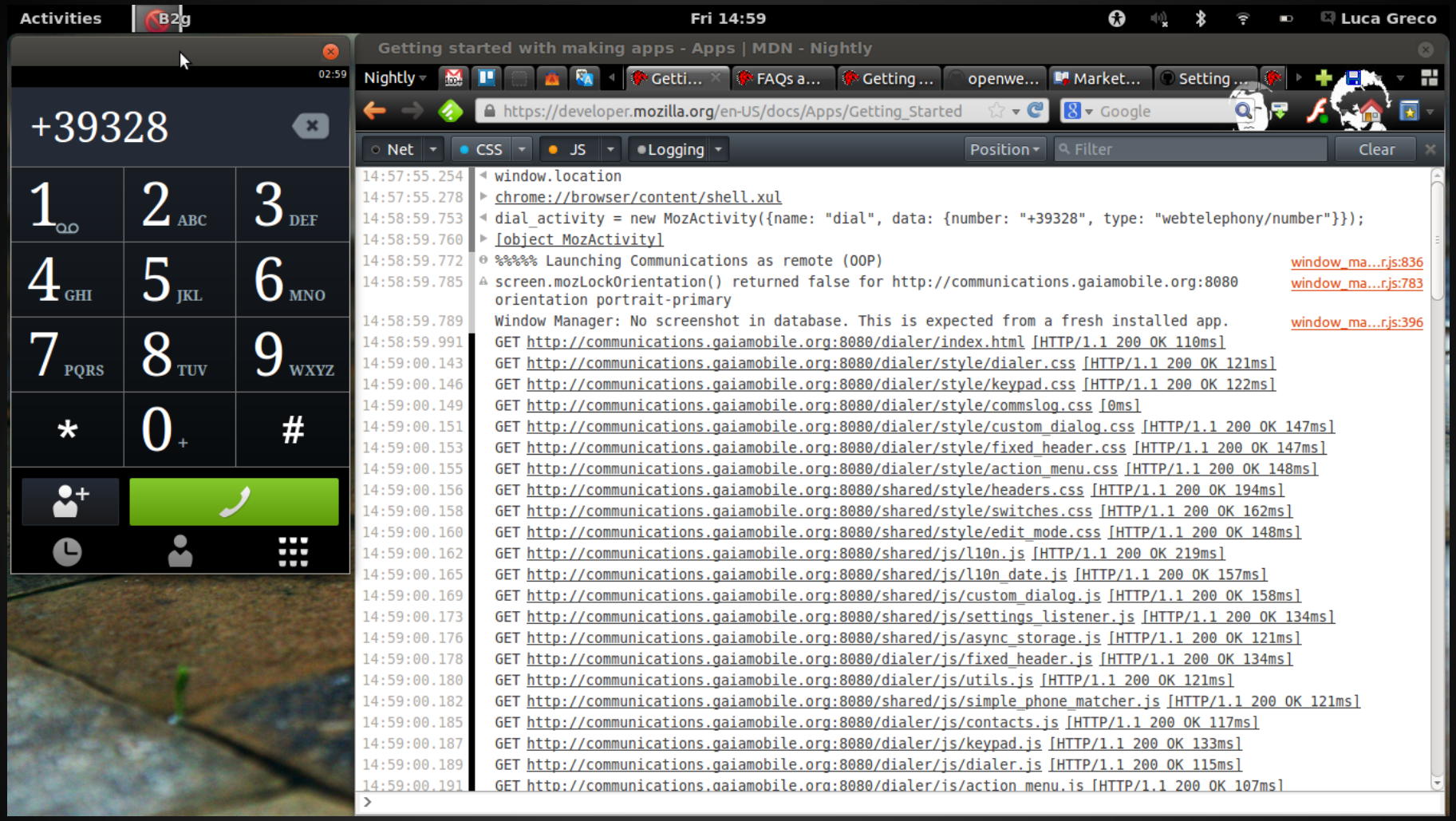
- adb, adb logcat
- Remote debugger

Remote Debugger

Debug Nightly, B2G Desktop, or a device.

http://blog.astithas.com/2012/10/debugging-firefox-os.html

- Need to disable OOP
  - Settings -> Device Info -> More Info -> Developer -> Disable OOP (check it)



Remote Debugger (Firefox Nightly)

Using FirefoxOS

Single-button Device

Physical buttons include:

- Home Button (bottom/center)
- Volume Rocker (left)
- Power Button (top/right)

Quitting an app: Hold home button (card-switcher) view, swipe up.

Simulate Hardware Buttons

in B2G Desktop

Linux

Home: Home Key

Power: End Key

Volume: Page Up/Down

Mac OSX

Home: fn + left arrow

Power: fn + right arrow

Volume: fn + up/down arrows

Apps

System App

The system app as a wrapping app, which managers, starts and stops webapps.

The system app controls several apps, each within an iframe. For example:

- Keyboard
- Homescreen
- Browser App
  - Browser Tab 1
  - Browser Tab 2

Launching an App

1. Home screen app gets app reference from mozApps API, and calls app.launch()
2. Gecko receives the request, fires mozChromeEvent to the System app with details of the app
3. System app handles the event by inserting the app iframe into it’s DOM
4. App loads within the iframe

Hello World App

FIle Structure

- apps/myapp/
  - index.html
  - manifest.webapp
  - style/
    - myimage.png
  - js/
    - app.js

manifest.webapp

```
{
"name": “MyApp",
"description": “Hello World!",
"launch_path": "/index.html",
"developer": {
        "name": "The Gaia Team",
        "url": "https://github.com/mozilla-b2g/gaia"
},
"permissions": [ "contacts", "settings" ],
"locales": {
        "en-US": { "name": “MyApp", "description": "Hello World!" }
},
"default_locale": "en-US",
"icons": {  "128": "/style/MyApp.png" }
}
```

Permissions

Need to be certified to use permissions.
(In apps folder).

Marketplace is the only place that can sign/certify apps besides the gaia make itself.

Where do I put it?

Packaged vs Hosted Apps

Packaged

- Permissions

Create a new directory in apps/.

Hosted

- AppCache

Look in the external-apps directory.

Install from the web

Use the Open Web Apps manifest : https://developer.mozilla.org/Apps

1. Launch the gaia browser, and navigate to your website.
2. Call navigator.mozApps.install(manifestURL) from your website within the browser.
    a. Alternatively: navigator.mozApps.installPackage()

Hosted App Basics

- webapp.manifest
- metadata.json
  - Points to an origin:
  - { "origin": "http://localhost:3000/"}
- Installation page
  - navigator.mozApps.install('http://localhost:3000/manifest.webapp')
- App is reachable at the origin.

Inside of the codebase

You can leverage the apps directory.
See test_apps/template/

- Create your app
- Build the profile
- Start B2G desktop
- App will be on homescreen

Installing the manifest

navigator.mozApp.install(manifestURL)

- Today: Download manifest only, register on your device
- Uses “appcache” to catch at first run

<html manifest="manifest.appcache">

- https://developer.mozilla.org/en-US/docs/HTML/Using_the_application_cache

Gaia Folder Structure

apps	an hour ago	Merge pull request #6954 from crdlc/bug-819
build	5 hours ago	Bug 817805 - Remove UA override for wikipe
dictionaries	4 months ago	Remove en dictionary. We already have en_g
external-apps	19 hours ago	Merge pull request #6925 from gregorwagner
locales	3 months ago	Merged with master [lodr]
media-samples	6 months ago	move sample photos from /sdcard/Pictures ba
shared	10 hours ago	Merge pull request #6926 from vingtetun/initlo
showcase_apps	22 days ago	Merge pull request #6444 from cpeterso/cpete
test_apps	an hour ago	Bug 820485 - Fix mozPay UI tests; r=albertop
test_media	5 months ago	Add a directory to allow for Test Media to be c
tests	8 days ago	Bug 817039 - Merge latest gaia-ui-tests to ga
tools	13 days ago	Bug 811328 - Remove all references of 'Firefo
.gitignore	a month ago	Bug 807529 - Gaia homescreen app: do not v
Android.mk	5 days ago	Bug 817040 - Add a build target to package g
LICENCE	8 months ago	Switch to Apache License. [andreasgal]
Makefile	10 hours ago	Bug 799714 - Define startup graphics and/or a
README.md	a month ago	Update test docs in README r=lightsofapollo
index.html	2 months ago	Bug 796149 - Test commit hooks from github.

https://github.com/mozilla-b2g/gaia

Gaia Folders

apps - Open web apps, calculator, dialer, system
build - Build scripts triggered by the makefile.
dictionaries - Predictive text completion (keyboard).
external-apps - "Hosted" apps. Each directory has a webapp.manifest and origin URL file.
media-samples - Sample media, make install-media-samples to deploy.
shared - Shared components (building blocks, localization)
showcase-apps - Sample games/apps (not being released).
test_apps - Test apps to test functionality.
test_media - Test video/music files, make install-test-media.
tests - Unit/Integration test support.
tools - XULRunner SDK Folder. (Command line javascript)

Building Blocks

What are they?

HTML and CSS for visual consistency across apps.

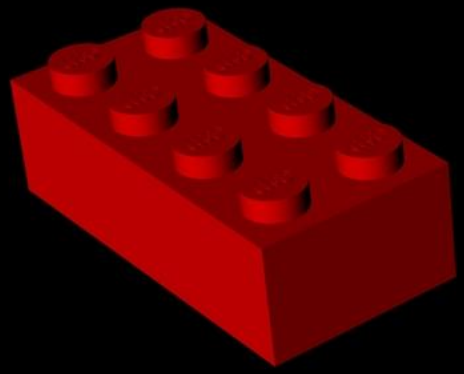
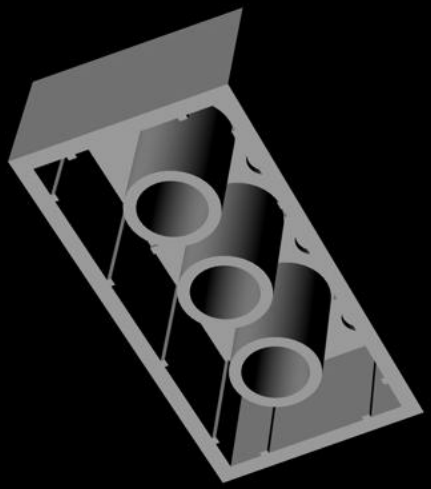
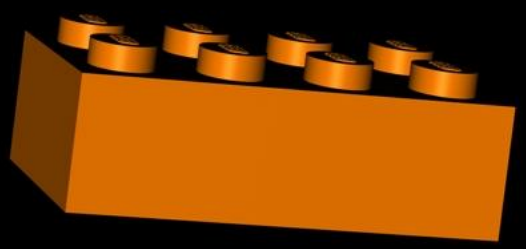
They are accessible, using aria roles.

Should generally speed up development.

Building Blocks Demo

http://buildingfirefoxos.com/



APIs and Communication

App Communication

Individual apps are run synchronously in a single thread - the same way a browser works.

Apps are run "OOP" (Out-of-process), and have no standard communication with each other.

There are a few ways to talk to apps indirectly through gecko.

MozActivity

Use Case: *AppA* delegates an activity to *AppB.*

Can only be started by a user interaction. (e.g., onclick event call stack)

- The 'host' app defines activities and filters in manifest.webapp.
- Consumers will fire off an event and delegate the event to the 'host' app.

https://wiki.mozilla.org/WebAPI/WebActivities

MozActivity Example

AppA manifest.webapp

```
"activities": {
  “pick": {
    "filters": {
      type: ["image/png", "image/gif"],
    },
    "disposition": "window",
    "returnValue": true
  }
}
```

AppA.js

```
navigator.mozSetMessageHandler('activity',
  function(req) {
    var image = getImg(req.source.data);
    req.postResult({
       type: image.type,
       url: image.url
    });
});
```

AppB.js

```
var req = new MozActivity({ name: "pick",
  data: {
    type: "image/png",
    multiple: false
  }
});


req.onsuccess = function() {
  var image = req.result;
  doSomething(image);
};


req.onerror = function() {
  alert("Fail to pick an image!");
};
```

Settings

- ril.data.enabled
  - ril - radio interface layer
  - rild - ril daemon (background process)
- wifi.enabled
- bluetooth.enabled
- geolocation.enabled

Settings get/set

Write: App -> mozSettings -> DB
Read: API -> read DB -> initial status
Represents the "intended" state - not actual.

Get a setting:

```
var req = settings.createLock().get('wifi.enabled');
req.onsuccess = function() {
  var enabled = req.result['wifi.enabled'];
};
req.onerror = function() {...};
```

Set a setting:

```
settings.createLock().set({'wifi.enabled': true});
```

Settings.addObserver

Listen for changed settings.

```
settings.addObserver('wifi.enabled', callback);

function callback(e) {
    var enabled = e.settingValue;
    doStuff(enabled);
}
```

Localization

Uses .ini property files.

Include in head:
<script src="shared/js/l10n.js">

- Provides: navigator.mozL10n

<script src="shared/js/l10n_date.js">

- Provides: navigator.mozL10n.DateTimeFormat

Localization

In HTML File:

```
<h1 data-l10n-id="settings">Settings</h1>
```

In Javascript:

Use the onLocalized event.

```
function onLocalized(callback) {
  if (navigator.mozL10n.readyState == 'complete') {
        callback();
  } else {
        window.addEventListener('localized', callback);
  }
}
```

asyncStorage

gaia/shared/js/asyncStorage
Async version of localStorage.
Convenient storage API - doesn't block.

asyncStorage.setItem('key', 'value');
asyncStorage.getItem('key', 'value', fn);

API: removeItem(), clear(), length(), and key()

Additional APIs

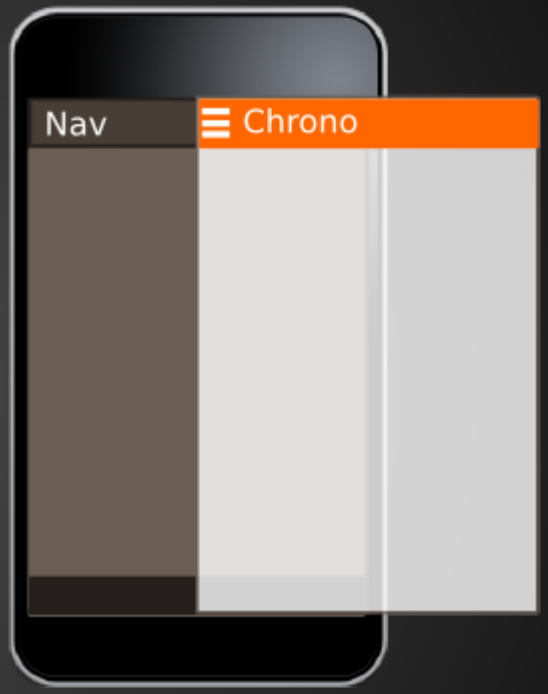
WebSMS/Telephony
Vibration
Device Storage
Orientation
Contacts
Light/Proximity Support

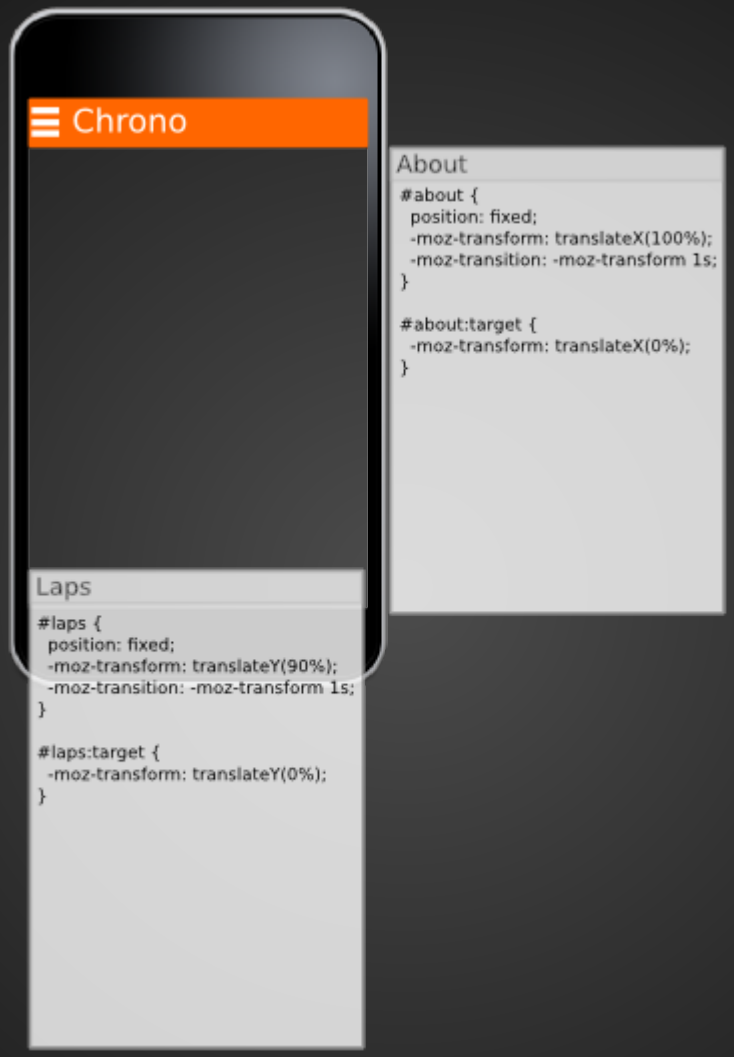
https://wiki.mozilla.org/WebAPI

Coding Strategies

Panels

Panels are simple DOM elements (e.g. a section or a div tag) initially positioned offscreen and moved on screen using a CSS transition:





Panel Example

Single Browser Support

Access to all of the HTML5 goodies.

- document.querySelector/querySelectorAll
- element.dataset.<>
- Canvas, Audio APIs

Recommendation: Avoid external libraries which implement extra cruft for older browsers.

Performance

Phone Developer Tools

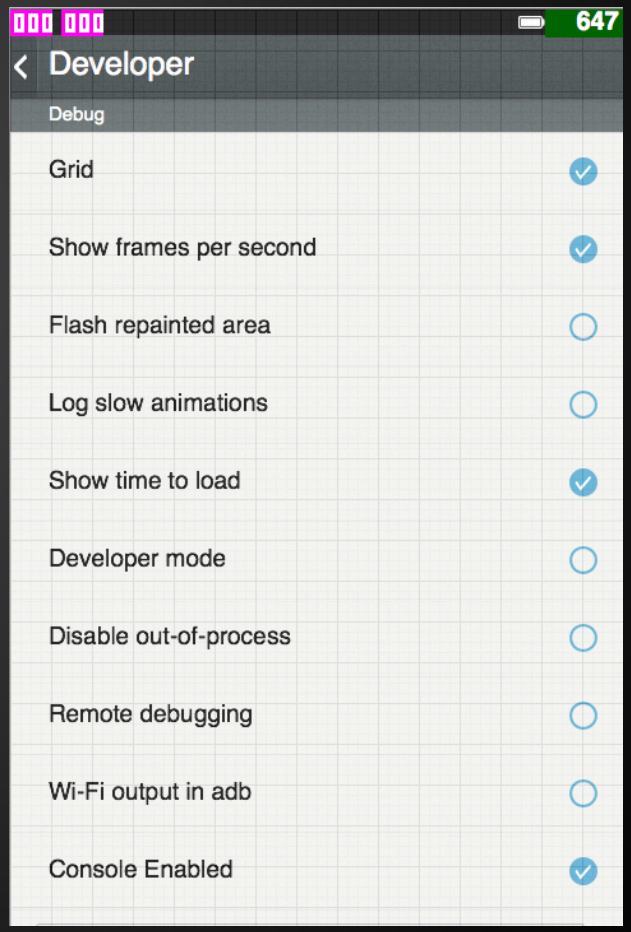
Settings -> Device Information -> More Information -> Developer

Time to load - Green number, top right.

Frames per second - Pink number, left

Flash repainted area - Highlights repaints with a random color.

Log slow animations - shows hints about animations in adb logcat.



Graphics Performance

Goal: 60 frames/second

CPU paints content once, GPU moves the content in memory.

Tips: Try to avoid animating CSS attributes, use transform where possible.

http://vimeo.com/48466888

Profiling

B2G Repo -> ./profile.sh

./profile start -> Starts the device with profiler

./profile ps -> Get a list of processes

./profile.sh capture [name] -> Caputres the profile.

Uses cleopatra UI: https://github.com/bgirard/cleopatra

Profile Memory

watch -n 1 'adb shell procrank | egrep "(RAM|b2g)"'

about:memory

B2G repo -> ./tools/get_about_memory.py

Tests

Unit Testing

Runs off of B2G Desktop.

- Run: make test-agent-server &
- Open the Test Agent app.
- Run: make test-agent-test
  - or: make test-agent-test APP=calendar

Tests will be run automatically as you save files.

Unit Test File Locations

apps/<APP>/test/unit

There is generally a 1-to-1 ratio between JS files and unit tests. E.g.,

- apps/calendar/js/app.js
- apps/calendar/test/unit/app_test.js

Integration Testing

Runs off of B2G Desktop or a phone.

1. Run B2G Desktop
2. Forward port 2828 from the emulator/device
    a. adb forward tcp:2828 tcp:2828
3. Run: make test-integration
    a. or: make test-integration APP=calendar

FirefoxOS devices must be flashed with marionette enabled.

Workflow

Linting

Check for syntax errors and style.

gjslint --nojsdoc my_file.js
or: make lint (lints all JS files)

What Device?
Fastest to develop -> Least Device Like
Most Device Like -> Slowest to Develop
FF nightly -> B2G Desktop -> Device

Misc

XULRunner

Firefox -> browser.xul (Menu toolbars, etc)
B2g -> shell.xul (empty - just an iframe)

Shell.xul loads the system app in an iframe.

System app acts on mozChromeEvents and notifies gecko through mozContentEvents.

Ask for help!

irc.mozilla.org

- #gaia
- #b2g

Mail list: https://lists.mozilla.org/listinfo/dev-gaia

Me:

- kgrandon@mozilla.com
- kevingrandon@yahoo.com

Resources

https://github.com/mozilla-b2g/gaia
http://mozilla.org/b2g
https://developer.mozilla.org/Apps
https://wiki.mozilla.org/Gaia/Hacking
https://wiki.mozilla.org/Gaia/Design/BuildingBlocks
http://mozilla-b2g.github.com/Gaia-UI-Building-Blocks/index.html
https://github.com/mozilla-b2g/Gaia-UI-Building-Blocks

Thanks!

Kevin Grandon, Engineer
kgrandon@mozilla.com