Mobile Programming for Educators

Ruben R. Puentedura, Ph.D.

The Lively Sketchbook

- Ubiquity
- Intimacy
- Embeddedness

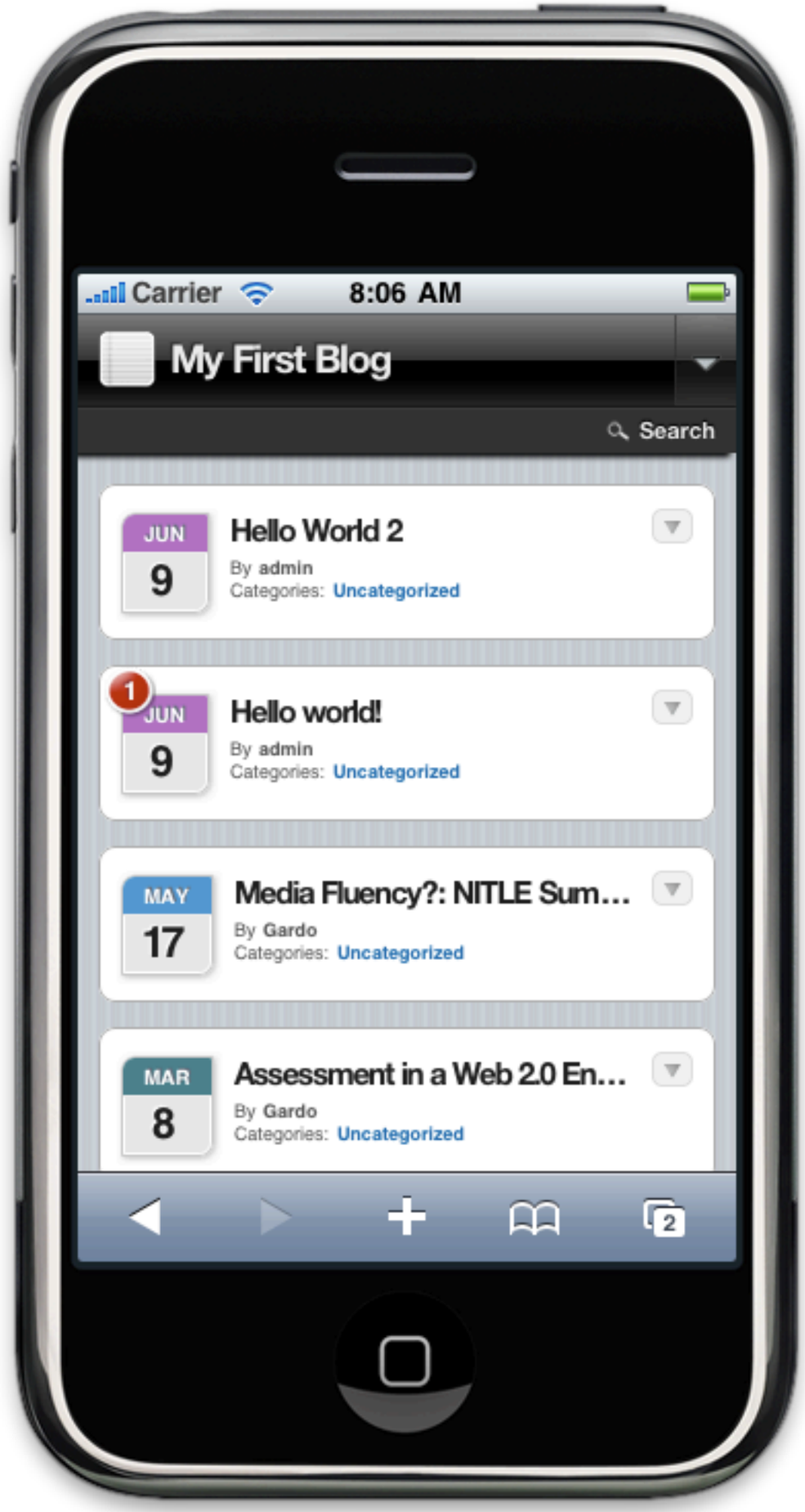

Writing Apps
Evernote
aNote
Quickoffice
MyWiki
WordPress
Analysis Apps
Instaviz
ProStats
SpaceTime
Image Apps
AutoStitch
PS Mobile
Flickr
SketchBook
SketchPad
PhotoForge
Strip Design


Audio Apps
Recorder
NLogFree
MultiTrack
MusicStudio
Video Apps
Storyrobe
Slideshow
Storyboard
ReelDirector
Broadcaster
Viewer
Utility Apps
Pastebot
Dictation
Air Sharing
Box.net
Jaadu VNC

The Feed App – On The Server and the Phone

Installing Wordpress

- Install MAMP – http://www.mamp.info/en/index.html
- Install WordPress – http://wordpress.org/
  - Unzip into MAMP's htdocs folder;
  - Create a database in MAMP;
  - Go to http://localhost:8888/wordpress/ to create your WordPress site
    - Use the database you've created and "root" as your user
- Login and go to the Plugins folder, then add:
  - WPtouch iPhone Theme
  - FeedWordPress


Carrier
8:06 AM
My First Blog
Search
JUN
9
Hello World 2
By admin
Categories: Uncategorized
1
JUN
9
Hello world!
By admin
Categories: Uncategorized
MAY
17
Media Fluency?: NITLE Sum…
By Gardo
Categories: Uncategorized
MAR
8
Assessment in a Web 2.0 En…
By Gardo
Categories: Uncategorized
2

The Feed App On The Phone

- Launch Dashcode
- Choose the RSS template for Mobile Safari
- Add a feed URL
- Add a title and offline viewing capability
- Add a web clip icon via drag-and-dropping an image
  - At least 57 pixels square, no shine or gloss
  - Note: a bug in Dashcode means that you may need to copy this icon into the Images folder by hand
- Save the project to disk, and test it.


Carrier
8:03 AM
My RSS Feed
WEDNESDAY, JUNE 9, 2010
Media Fluency?: NITLE Sum...
In late March of this year I was privileged to speak...
Assessment in a Web 2.0 Envi...
I agree in principle that we who work in education ...
Gratitude and clarifications
First, my thanks to everyone who responded to my...
Changing lives
I keep running into the same wall from different di...
Another bootstrapping experi...
Today's the second day we'll meet. Already the acti...
A happy birthday
Today the father of interactive computing, the thin...
"In Our Time" podcast series o...
There's a new set of four episodes from BBC Radio...

A Bit Of User Interface Design

The Wisdom of Apple

- Expect to spend about 60% of your total time designing the app
- Read the Human Interface Guidelines
  - http://developer.apple.com/iphone/library/navigation/index.html
- Three key steps:
  - Define your focus - i.e., define the solution you're creating
  - Refine the feature set
  - Determine your users' mental model

Some Key Elements of the HIG

- Minimum target: 22 pixels vertically, 44 pixels square
- Navigation follows a tree model, emphasized by visual animation - no home button or breadcrumbs
- Upper left corner navigates up the tree, upper right corner modifies/adds items
- Lists:
  - Ungrouped: textual – all items at a given level
  - Grouped: text + icons – classify and group items
- Pickers:
  - 12 items optimal
- Toolbars:
  - Icons only – correspond to tools
- Tab bars:
  - Text and icons – switch between different modes, groupings
- Buttons:
  - Have rounded corners to encourage tapping
- Animation:
  - Use to provide feedback
- Aesthetics:
  - Keep the look and feel clean and uncluttered
  - Emphasize tactile design
  - Prioritize

The Handheld Database

Creating the Database App

- Select the Browser template, and complete the basic fields
- Test the app
- Edit your datasource:
  - Add an image field (*imagelocation*) to *sampleData.js*
  - Add the images (e.g., in a folder labeled *Photos*)
  - Refresh the datasource
- Add an image part (*parkImage*) to the *detailLevel*
- In the "list" Data Source view, bind the *imagelocation* to the src of *parkImage*
- Add a Map Button (parkMap) to the *detailLevel*, and bind:
  - *location* to the *state* of *parkMap*
  - *name* to the *address* of *parkMap*


Carrier
10:19 AM
Browser
Acadia
Acadia
Maine, USA
The scenery in Acadia is amazing this time of year!

Getting Input

Building a Calculator

- Select the Custom Project template
- Add two Text Field parts (*numberOne*, *numberTwo*) to the *content* area
- Add a Text part (*resultSum*) below them
- Add a Lozenge Button (*calcSum*) below them
- Add an onclick event to calcSum
- Add the following code to the myClickHandler:

```
var firstnum = parseFloat(document.getElementById("numberOne").value);
var secondnum = parseFloat(document.getElementById("numberTwo").value);
var thesum = firstnum + secondnum;
document.getElementById("resultSum").value = thesum.toFixed(4);
```

Adding Memory

- Add to the load() function:

```
var oldfirstnum = localStorage.getItem("first_num");
var oldsecondnum = localStorage.getItem("second_num");
document.getElementById("numberOne").value = oldfirstnum;
document.getElementById("numberTwo").value = oldsecondnum;
```

- Add to myClickHandler:

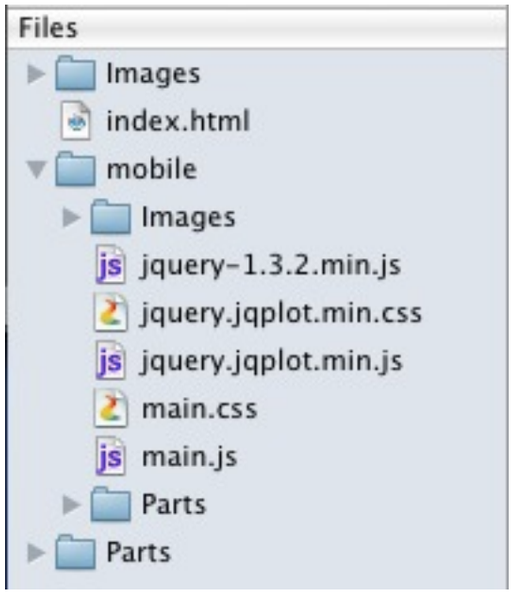
```
localStorage.setItem("first_num", firstnum);
localStorage.setItem("second_num", secondnum);
```


Carrier
10:58 AM
1.3
4.2
5.5000
Sum

Visualizing Data and Getting Images

Ingredients for Visualization

- Download the jqPlot library at http://www.jqplot.com/
- Drag the three key components into the Files window:
- On a Custom template, add:
  - A Lozenge Button (*visualizeIt*)
  - A Text part (*tableSetup*)
  - An empty div (`chartdiv`):
    `<div id="chartdiv" style="height:400px;width:300px; "></div>`
    - This last component will need to be typed in by hand
- Make the changes to *main.js* and *main.css* shown in the next page:

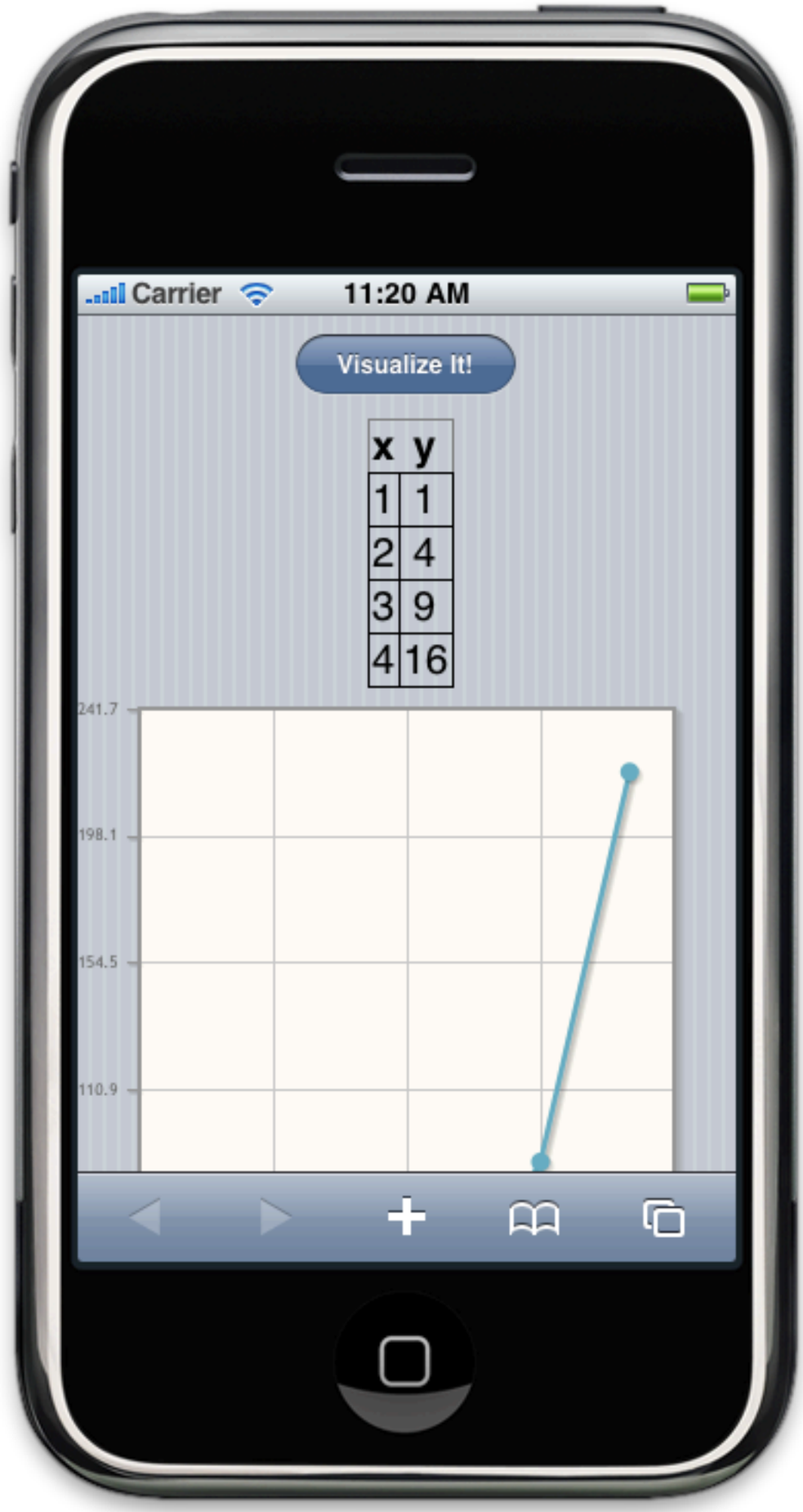


main.js <No selected symbol>

```js
/*
 This file was generated by Dashcode.
 You may edit this file to customize your widget or web page
 according to the license.txt file included in the project.
 */

//
// Function: load()
// Called by HTML body element's onload event when the web application is ready to start
//
function load()
{
    dashcode.setupParts();
}

function myClickHandler(event)
{
    var tablearray = [[1,1],[2,4],[3,9],[4,16]];
    var tablebegin = "<table id='myTable'><tr><th scope='col'>x</th><th scope='col'>y</th></tr>";
    var tableend = "</table>";
    var tablecontent = "";

    for (var i = 0; i < tablearray.length; i++)
    {
        xelement = "<td>" + tablearray[i][0] + "</td>";
        yelement = "<td>" + tablearray[i][1] + "</td>";
        tablerow = "<tr>" + xelement + yelement + "</tr>";
        tablecontent = tablecontent + tablerow;
    }

    document.getElementById("tableSetup").innerHTML = tablebegin + tablecontent + tableend;

    $.jqplot('chartdiv',  [[[1, 2],[3,5.12],[5,13.1],[7,33.6],[9,85.9],[11,219.9]]]);
}
```

main.css <No selected symbol>

```css

table {
    border-collapse: collapse;
    border-style: solid;
    border-width: 1px;
}

td {
    border-style: solid;
    border-width: 1px;
}

```

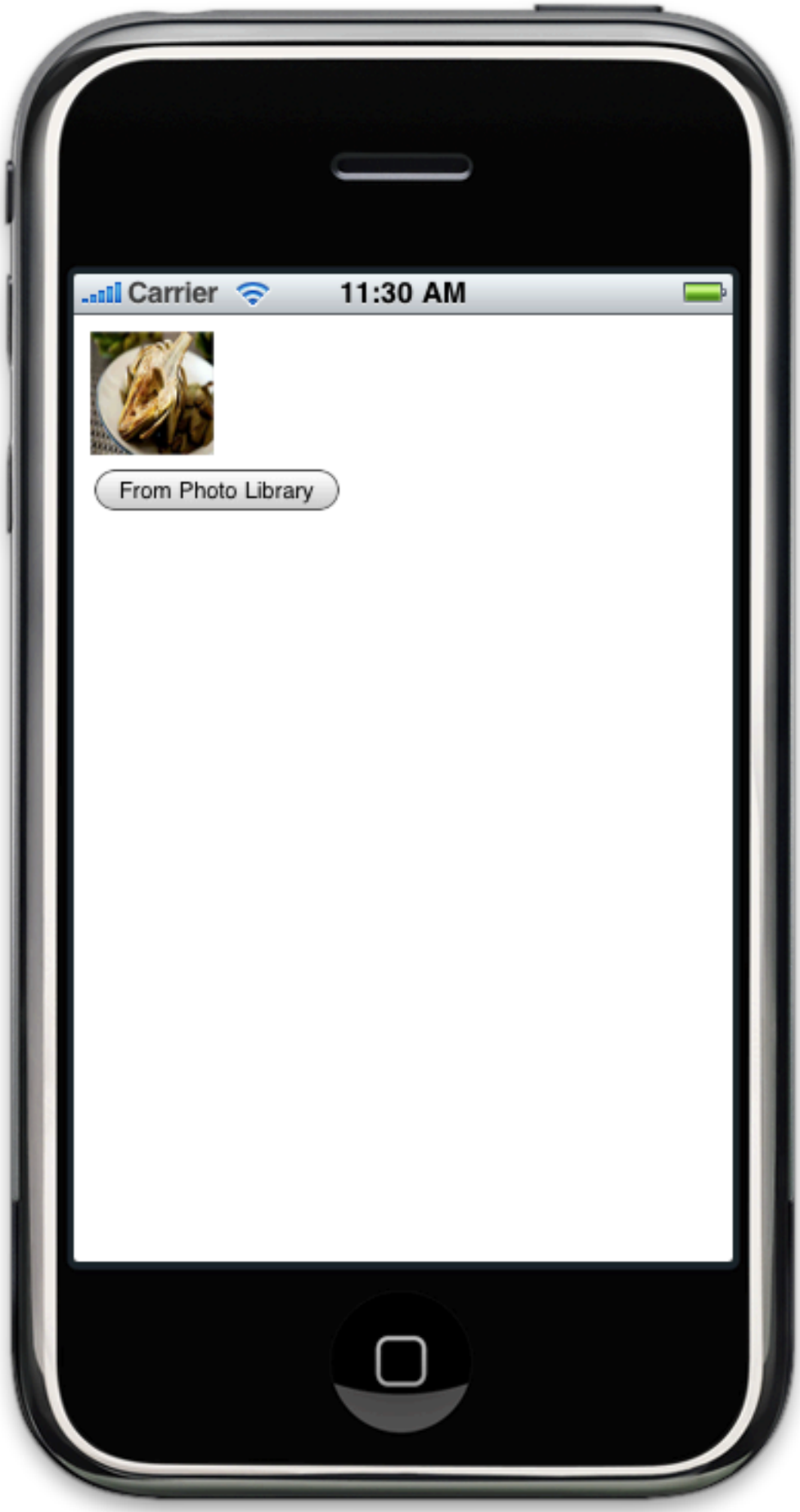

Carrier
11:20 AM
Visualize It!
x y
1 1
2 4
3 9
4 16
241.7
198.1
154.5
110.9

Accessing Native Functions: the Photo Library

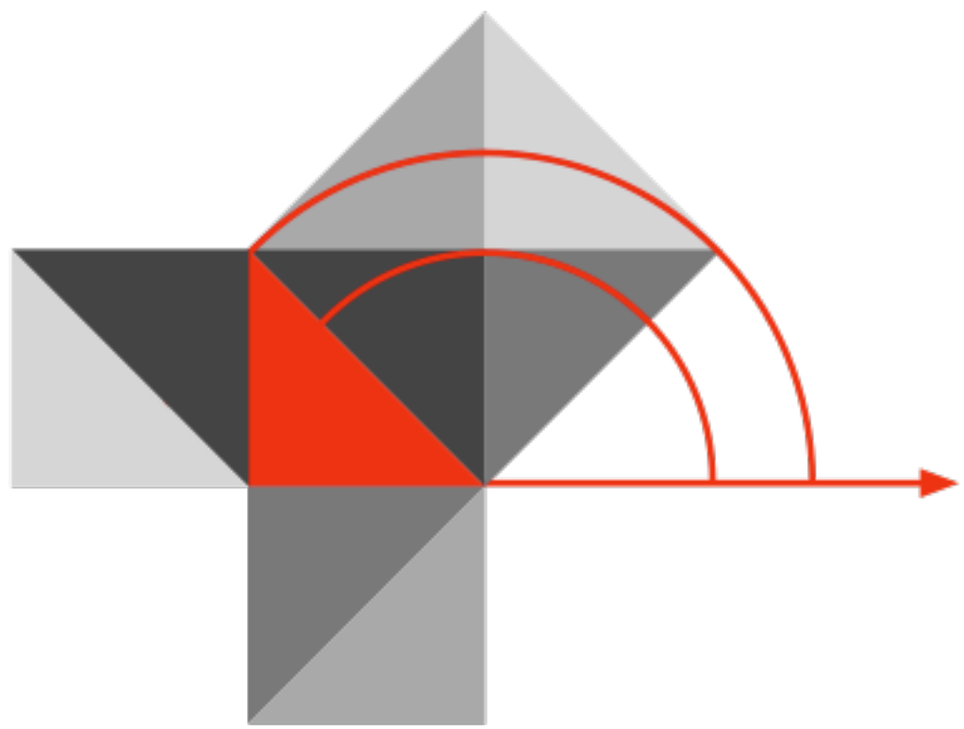
- Decompress PhoneGap (*phonegap-iphone.zip*)
- Install it by double-clicking on *PhoneGapLibInstaller.pkg*
- Should this not work, full instructions for an alternative install are available at: http://phonegap.pbworks.com/Getting-Started-with-PhoneGap-(iPhone)
  - This alternative install will require that you install Git first: http://code.google.com/p/git-osx-installer/downloads/list
- Launch XCode, and create a PhoneGap project
- Modify the index.html file (inside the www folder) as shown in the next page
- When running the finished code under iPhone Simulator 3.1.3, make sure to choose an image that is not one of the photos included with the simulator – due to a bug in the simulator, these photos will not work

index2.html:34 <body id="stage" class="theme">

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
  "http://www.w3.org/TR/html4/strict.dtd">
<html>
  <head>
    <meta name="viewport" content="width=320; user-scalable=no" />
    <meta http-equiv="Content-type" content="text/html; charset=utf-8">
    <title>PhoneGap Photo Library</title>
    <link rel="stylesheet" href="master.css" type="text/css" media="screen" title="no title" charset="utf-8">
    <script type="text/javascript" charset="utf-8" src="phonegap.js"></script>
    <script type="text/javascript" charset="utf-8">

    function PictureSourceType() {};
    PictureSourceType.PHOTO_LIBRARY = 0;
    PictureSourceType.CAMERA = 1;

    function getPicture(sourceType)
    {
        var options = { quality: 10 };
        if (sourceType != undefined) {
            options["sourceType"] = sourceType;
        }
        navigator.camera.getPicture(getPicture_Success, null, options);
    };

    function getPicture_Success(imageData)
    {
            document.getElementById("test_img").src = "" + imageData;
    }
    </script>
  </head>
  <body id="stage" class="theme">
    <img style="width:60px;height:60px" id="test_img" src="" /><br/>
    <button onclick="getPicture(PictureSourceType.PHOTO_LIBRARY)">From Photo Library</button><br />
  </body>
</html>
```


Carrier
11:30 AM
From Photo Library

Hippasus



http://hippasus.com/rrpweblog/
rubenrp@hippasus.com