Scientific m-Learning

4 – 7 June 2012

Mobile Application Development using App Inventor for Android Devices

TRIVEDI Kirankumar Rajnikant
Shantilal Shah Engineering College New Sidsar Campu,
PO Vartej Bhavnagar 364001 Gujarat
INDIA
Mobile Application Development using App Inventor for Android Devices

Kiran Trivedi, India

“Scientific m-learning Workshop”
Why App Inventor?

• No syntax
  – The blocks language eliminates the need to remember and type code

• Everything is right in front of you
  – The components and functions are organized into drawers. Just find, drag, and drop.

• Events at top level
  – "When this happens, the app does this" is the correct conceptual model. Down with Listeners!

• High-level components
  – The app inventor team has built a great library with simplicity the main goal.

• Only some blocks plug-in
  – You can't do things that don't make sense.

• Concreteness
  – You program components, not abstractions
What is the App Inventor?

• App Inventor lets you develop applications for Android phones using a web browser and either a connected phone or emulator.

• The App Inventor servers store your work and help you keep track of your projects.
What you need?

- The *App Inventor Designer*, where you select the components for your app.
- The *App Inventor Blocks Editor*, where you assemble program blocks that specify how the components should behave.
- You assemble programs visually, fitting pieces together like pieces of a puzzle.
Procedure

• Your app appears on the phone step-by-step as you add pieces to it, so you can test your work as you build.

• When you're done, you can package your app and produce a stand-alone application to install.

• If you don't have an Android phone, you can build your apps using the Android emulator, software that runs on your computer and behaves just like the phone.
On the website

• The App Inventor development environment is supported for Mac OS X, GNU/Linux, and Windows operating systems, and several popular Android phone models. Applications created with App Inventor can be installed on any Android phone. (See system requirements.)

• Before you can use App Inventor, you need to set up your computer and install the App Inventor Setup package on your computer.
System requirements

To use App Inventor, your computer must meet the following system requirements:

- Computer and operating system
- Macintosh (with Intel processor): Mac OS X 10.5, 10.6
- Windows: Windows XP, Windows Vista, Windows 7
- GNU/Linux: Ubuntu 8+, Debian 5+

- Browser
- Mozilla Firefox 3.6 or higher
- Apple Safari 5.0 or higher
- Google Chrome 4.0 or higher
- Microsoft Internet Explorer 7 or higher
Phone

- Applications created with App Inventor can run on any Android Phone. The development environment and Setup software itself directly supports the following phones:
- Your phone must have an SD card installed, or else it won't work with App Inventor.
- App Inventor also works with many other Android phones, including models from HTC, Samsung, and Dell, but in many cases you will need to download and install additional software from the manufacturer if needed.
Set up your Android phone

• To get your phone ready to work with App Inventor, follow these steps:
• Tap the Home button to go to your phone's Home screen.
• Tap the Menu button, then Settings, then Applications.
• If your phone has an Unknown sources setting, make sure it is checked.
Android architecture
App Inventor Architecture

- App
  - Interface
    - Component #1
    - Component #2
  - Behavior
    - Event Handler #1
    - Event Handler #2
USER INTERFACE

• Consists of Components
  – Components are same as Tools in windows forms or Web forms.

• Two Types of Components
  – Visible Components
  – Non-Visible Components
Visible component

- Widely used components
- These are common components
  - Button
  - TextBox
  - Label
  - CheckBox
  - Etc.
Non visible Components

• We won’t be dealing much with these controls today.
• These are controls such as timer and DataSource.
• These are not Visible on the screen but have their own functionality.
• Demo
Properties of the Components

• Select any Visible or non Visible Component.
• In the right corner we can see the properties of the item.
Behaviour

• Behavior is same as the code part in windows forms.
• Behavior is defined using Block Editor.
Define a variable

• Defining:
  – Go to Built in Tab Drag Drop the Define Variable
  – Set its name by clicking on ‘variable’.
  – Set its Datatype by clicking the ‘Down arrow’ next to ‘?’.
  – Assign it the value by clicking the newly appeared value and then typing in the new value.
  – Demo.
# Possible Events

<table>
<thead>
<tr>
<th>Event Types</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>User-initiated Events</td>
<td>Button click</td>
</tr>
<tr>
<td>Initialization Events</td>
<td>At App launch</td>
</tr>
<tr>
<td>Timer Events</td>
<td>After 1 sec do something</td>
</tr>
<tr>
<td>External Events</td>
<td>Receive a text</td>
</tr>
</tbody>
</table>

![Diagram](image.png)

Scientific m-learning Workshop
Define a function

• Set of activities grouped together.
• Can send in Multiple Arguments
• Two Types of functions
  – With return values.
  – With out return values.
Test and Debug

• **Deactivate**
  – Right Click on a block and choose Deactivate.
  – Choosing *Deactivate* from the block menu will keep the block from becoming part of the app when you package it.
  – Selecting *Activate* restores the deactivated block.

• **Collapsing blocks**
  – If your app has many blocks, they won't all fit on the screen at once.
  – Use *block collapsing*.
  – Click on the minus sign at the lower left of any block.
  – Only the title will be visible.
  – Click on the plus sign to restore the block to full visibility.
Emulator

• **Starting the Emulator**
  – You don't need to download any additional software to use the emulator.
  – It was included with the software you already downloaded as part of the App Inventor Extras Package.
  – Navigate to the directory where the App Inventor Extras software was installed, locate the folder called commands-for-appinventor
  – Run the command ‘run-emulator’.
Project upload/download

• **Download Source**
  – Go to the *My Projects* page,
  – Select a project,
  – Then choose *More Actions / Download Source*.
  – This will create a zip file which you can share with others.

• **Upload Source**
  – To upload a project, go to *My Projects*,
  – Choose *More Actions / Upload Source*,
  – Choose the zip file previously downloaded from App Inventor.
First Time Login
Scientific m-learning Workshop
Pallatte

Scientific m-learning Workshop
Viewer / Workspace
Component Properties

Welcome to the App Inventor beta preview release. Be sure to check the list of known issues.

Components

- Screen1

Properties

- Background Color
  - White
- Background Image
  - None
- Icon
  - None
- Screen Orientation
  - Unspecified
- Scrollable
  - On
- Title
  - Screen1
First Example!
My Blocks
Scientific m-learning Workshop
Step by Step Block Creation
Setting up Blocks for Hello_Lion
Running an Emulator

Please wait. It can take 2 or 3 minutes for the emulator to become ready — even longer the first time.

The emulator will be ready when the phone screen has a picture background as shown in the right image here. If it has a lock screen, as shown in the left, slide the bar with the green lock icon to the right to unlock it.

Even after the picture appears, continue to wait for the emulator to finish preparing the SD card: watch the notification area at the top of the emulator screen.

When the emulator is ready, choose it from the Connect to Device menu to connect.

The Troubleshooting Guide at http://beta.appinventor.mit.edu/learn/troubleshooting.html may provide further help.
Connect to Emulator

Scientific m-learning Workshop
Example: SMS Auto Responder
UI

Scientific m-learning Workshop
Basic Auto Responder
Define the Variable

def responseText as text

Hi, I'm Busy.
Add Click Event to Button for changing the response text
Add Message Received Event
Add If Else from built in control block

• This app will response to a specific number
Set the SMS text and phone number for sending the SMS

Scientific m-learning Workshop
Complete Application

when Button1.Click do
  set global responseText to TextBox1.Text

when Texting1.MessageReceived
  number
  name
  number
  messageText
  name
  messageText
  do
    if test
      value number = number 9998339966
      then-do
        set Texting1.PhoneNumber to value number
        set Texting1.Message to global responseText
        call Texting1.SendMessage
    end-if
  end-do

def responseText as text
  Hi, I'm Busy.
Now Let us make a Workshop Twitter
The UI for the Twitter App
Login Button Definition
Logout Button Definition
Ingredients
LabelStatus Definition
Twitt Button Definition

Scientific m-learning Workshop
Twitt Text Definition
Twitter1 Definition

- My Definitions
- ButtonLogin
- ButtonLogout
- HorizontalArrangement1
- Label1username
- LabelStatus
- Screen1
- Tweet_Button
- Tweet_Text
- Twitter1
- VerticalArrangement1

Scientific m-learning Workshop
Add Twitter to your Application

Extend your reach.
Multiply your audience.

Add Twitter to your website
ICTP m-learning Workshop

Application Details

Name: * ICTP m-learning Workshop
Your application name. This is used to attribute the source of a tweet and in user-facing authorization screens. 32 characters max.

Description: * ICTP Workshop
Your application description, which will be shown in user-facing authorization screens. Between 10 and 200 characters max.

WebSite: * http://mbsapps.org
Your application’s publicly accessible home page, where users can go to download, make use of, or find out more information about your application. This fully-qualified URL is used in the source attribution for tweets created by your application and will be shown in user-facing authorization screens.

If you don’t have a URL yet, just put a placeholder here but remember to change it later.

Application Icon

Change icon: Choose File No file chosen
Maximum size of 700k. JPG, GIF, PNG.
Scientific m-learning Workshop
## OAuth settings

Your application's OAuth settings. Keep the "Consumer secret" a secret. This key should never be human-readable in your application.

<table>
<thead>
<tr>
<th>Access level</th>
<th>Read-only</th>
</tr>
</thead>
<tbody>
<tr>
<td>Consumer key</td>
<td>fyG0Dn3hR1xRk3mT2X91q</td>
</tr>
<tr>
<td>Consumer secret</td>
<td>haYrTKo3j0w7zh2EtZAg4J0dJFXw29QItwXj9x9Pl9c</td>
</tr>
<tr>
<td>Request token URL</td>
<td><a href="https://api.twitter.com/oauth/request_token">https://api.twitter.com/oauth/request_token</a></td>
</tr>
<tr>
<td>Authorize URL</td>
<td><a href="https://api.twitter.com/oauth/authorize">https://api.twitter.com/oauth/authorize</a></td>
</tr>
<tr>
<td>Access token URL</td>
<td><a href="https://api.twitter.com/oauth/access_token">https://api.twitter.com/oauth/access_token</a></td>
</tr>
<tr>
<td>Callback URL</td>
<td><a href="http://twitter.com">http://twitter.com</a></td>
</tr>
</tbody>
</table>

**Your access token**

It looks like you haven't authorized this application for your own Twitter account yet. For your convenience, we give you the opportunity to create your OAuth access token here, so you can start signing your requests right away. The access token generated will reflect your application's current permission level.

Create my access token
### OAuth settings

Your application's OAuth settings. Keep the "Consumer secret" a secret. This key should never be human-readable in your application.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access level</td>
<td>Read, write, and direct messages</td>
</tr>
<tr>
<td></td>
<td><a href="https://dev.twitter.com/docs/apis/overview">About the application permission model</a></td>
</tr>
<tr>
<td>Consumer key</td>
<td>fygOCN98kNN1xRkk3mT2X91Q</td>
</tr>
<tr>
<td>Consumer secret</td>
<td>n2YGTkIo3jUw7ch2EeZEA4j02FeW29QIHwKjMxSPF90</td>
</tr>
<tr>
<td>Request token URL</td>
<td><a href="https://api.twitter.com/oauth/request_token">https://api.twitter.com/oauth/request_token</a></td>
</tr>
<tr>
<td>Authorize URL</td>
<td><a href="https://api.twitter.com/oauth/authorize">https://api.twitter.com/oauth/authorize</a></td>
</tr>
<tr>
<td>Access token URL</td>
<td><a href="https://api.twitter.com/oauth/access_token">https://api.twitter.com/oauth/access_token</a></td>
</tr>
<tr>
<td>Callback URL</td>
<td><a href="http://twitter.com">http://twitter.com</a></td>
</tr>
</tbody>
</table>

### Your access token

Use the access token string as your "oauth_token" and the access token secret as your "oauth_token_secret" to sign requests with your own Twitter account. Do not share your oauth_token_secret with anyone.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access token</td>
<td>26739783-oilG2yl5qhKyWQdKkTlPYShZg2MDpG6vZINvMf6X</td>
</tr>
<tr>
<td>Access token secret</td>
<td>EZN0FhblLb3rJua2a8ZheV4aZ9OkxQg9exxHEunp0</td>
</tr>
<tr>
<td>Access level</td>
<td>Read, write, and direct messages</td>
</tr>
</tbody>
</table>
Google Static Map App
Show Location (Manual Entry)
Show My Location (GPS)
Orientation Sensor Example

Scientific m-learning Workshop
Drawing on Canvas
Setting Colors

- **Red Button Click**
  - set `DrawingCanvas.PaintColor` to `Red`

- **Blue Button Click**
  - set `DrawingCanvas.PaintColor` to `Blue`

- **Green Button Click**
  - set `DrawingCanvas.PaintColor` to `Green`
Canvas Touched Event
Canvas Dragged Event
Setting Dot Size
Using Camera
Trieste Tour
Blocks for Trieste Tour App
Xylophone
Call / Text App
Blocks for the App

[Diagram of App Inventor interface showing blocks for actions like clicking buttons to make phone calls or send messages.]