AI2 Static Maps 101 ...the

Google Static Map Tutorial for AI2

Static Maps do not need to be be static. Need an active map, map pins, follow your location in real time, bla bla bla ? Did ya' know, Google Static Maps can be ACTIVE? The *AI2 Static Maps 101* tutorial demonstrates how to use Static Maps. A subsequent tutorial will show how to use them along with the LocationSensor. A network or WIFI connection is required to use Google's Static Maps.

The Google *Static Maps API V2 Developer Guide* is your best friend. The *Static Maps Developer Guide* (https://developers.google.com/maps/documentation/staticmaps/) shows an example of how to use a static map in a browser. Static Maps are accessed by posting location information to an url. Developers can build the url in an app and place map pins and display the generated map using the WebViewer component. The Google example is reproduced as an AI2 coding exercise and ways you can get fancy are described.

The Google "Quick Example" looks like this (see the link above):

"A Quick Example

The following example contains the URL of a static map image of downtown New York City, which is displayed below:

https://maps.googleapis.com/maps/api/staticmap?center=Brooklyn+Bridge,New+York,NY&zoom=13&size=600x300&maptype=roadmap
&markers=color:blue%7Clabel:S%7C40.702147,-74.015794&markers=color:green%7Clabel:G%7C40.711614,-74.012318
&markers=color:red%7Clabel:C%7C40.718217,-73.998284
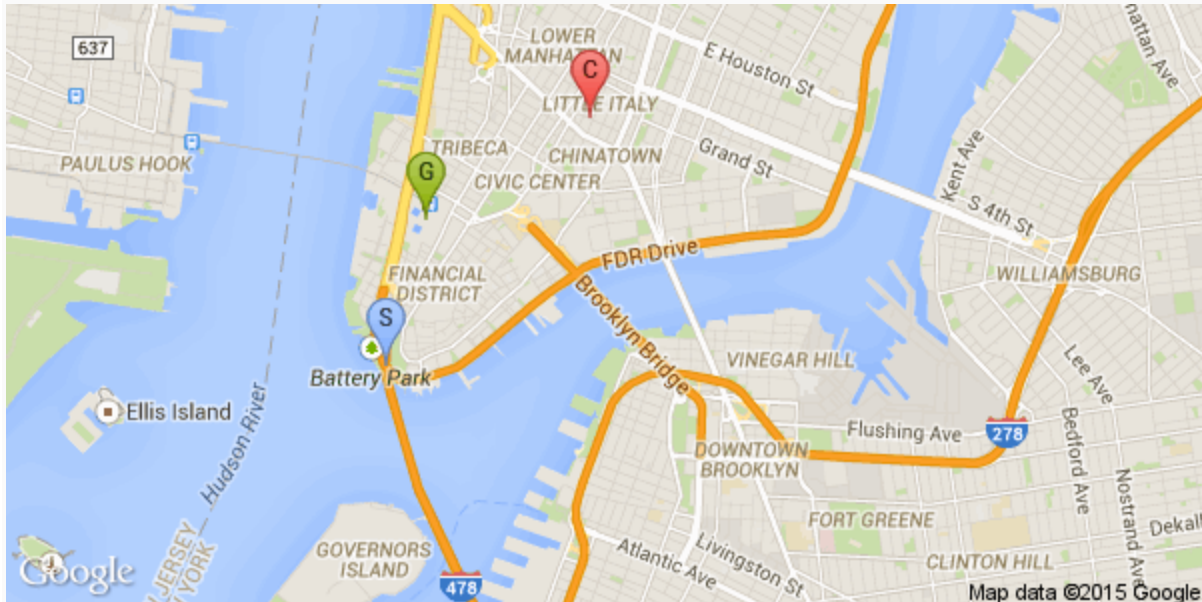
Notice that you don't need to do anything "special" to get this image to show up on the page. No JavaScript is required. All we needed to do was create a URL, and place it within an `<img>` tag. You can place a Google static map anywhere on your webpage where you can place an image."

AI2 Static Maps

AI2 developers can duplicate the Static Map example with *App Inventor* using very little code.
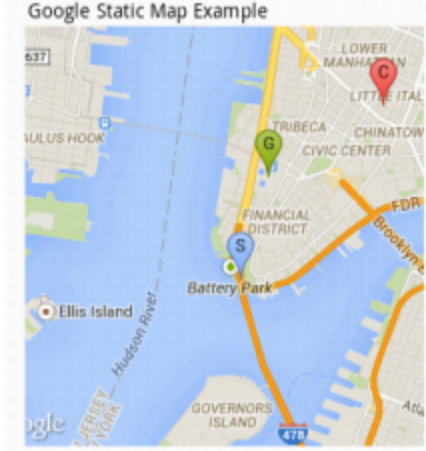

The text in the collapsed url puzzle slot shown is:

https://maps.googleapis.com/maps/api/staticmap?center=Brooklyn+Bridge,New+York,NY&zoom=13&size=600x300&maptype=roadmap
&markers=color:blue%7Clabel:S%7C40.702147,-74.015794&markers=color:green%7Clabel:G%7C40.711614,-74.012318
&markers=color:red%7Clabel:C%7C40.718217,-73.998284   .

Map pins can be fancy (with a simple ID tag, different colors and marker size etc.) or simple pins without annotation (the default red pin).

| "Fancy" Pins | "Simple" Pins |

The above two Static Maps are made with hard coded pin locations. The code instructions necessary to generate the Static Map do not need to be hard coded; a csv file provides all the information when included as a resource or loaded using the File control.

Simple pins from a csv file   *Simple pins from a csv file on the example map*
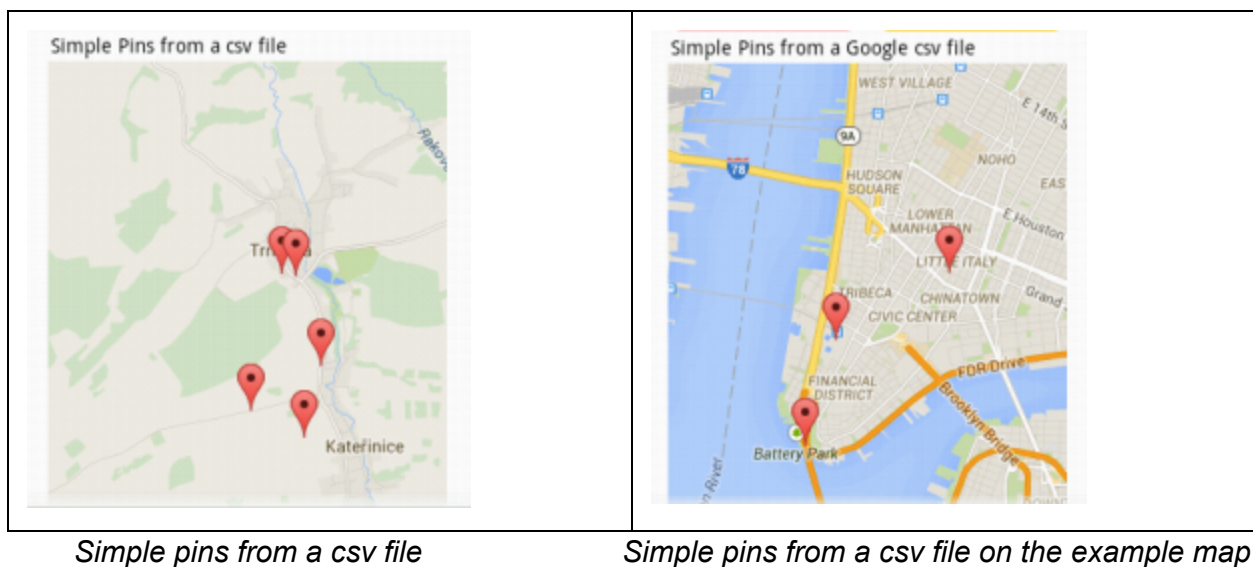
The url that creates the static map at Google must be less than 2048 characters in length. The Google limitation means 36 fancy pins or 63 simple pins can be used with the AI2 code described below and remain under the url character limit. The more bells and whistles included when generating the pins, the fewer pins it is possible to display on your device.

Static maps are created at different resolution levels (`&zoom=13`) -the larger the value the greater the zoom; 1 is the entire World). The example maps are at a zoom of 14 and 13 mostly. Specify the map size to display in pixels (`&size=600x300`), pin color and ID on a pin (`color:blue%7Clabel:S%7C`) -single characters only,this example ID's pin "S"; and the map type (`&maptype=roadmap`). Read the Google document for more information about the required flags. It is possible to many other surprising enhancements; some of which will be the topic of future tutorials.

The device's present location can be included in the Static Map url. The second part of this tutorial shows how to use the GPS location with a Static Map and how to update the map. How to do that is not part of this tutorial.

The Designer
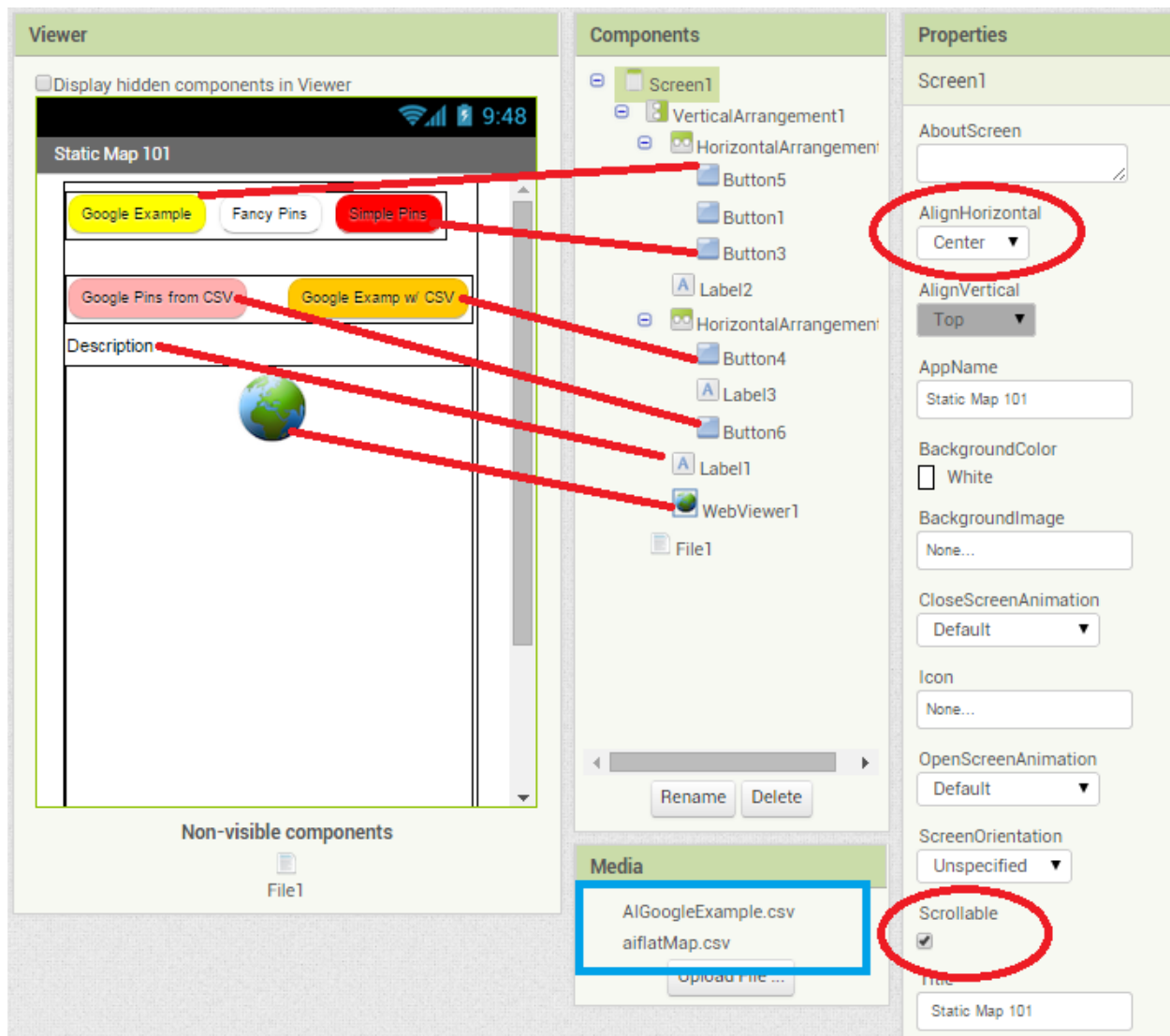
There is no special arrangement of Controls. This Designer image is all you need to relate the Blocks described below to the buttons in the app. There is no "Quit" button, no special error control … what you build is not a real app; just a demonstration of a possibly useful technique.

Make sure *Scrollable* is checked as circled above and note other settings. The *WebViewer* Height is set to 420 pixels.

The Blocks

The Blocks required are:

The BUTTONS
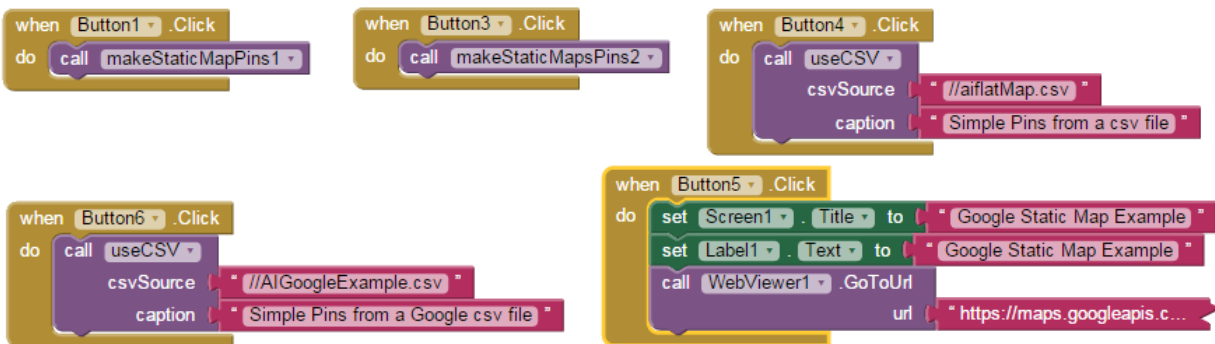
Very little goes into the Button event handlers. The work takes place in three procedures discussed below that are called in the event handlers.

The FILE Control

The VARIABLES

initialize global mapURL to " "   initialize global aiMapCSV to " "

initialize global pinList to create empty list   initialize global locationTxt to " "

initialize global staticMapURL to " "

The SCREEN

when Screen1 .Initialize
do   call File1 .ReadFrom
                fileName " //aiflatMap.csv "
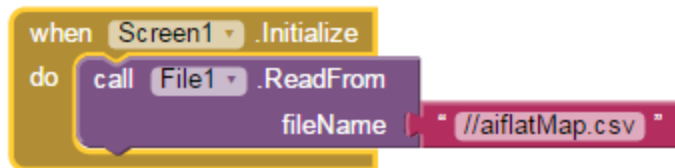
Three PROCEDURES

The *makeStaticMapPins1* Procedure produces a Static Map with annotated, colorful pins.

```
to makeStaticMapPins1
do  set global staticMapURL to  join  " http://maps.googleapis.com/maps/api/staticmap?center= "
                                       " 49.676000,18.184500 "
                                       " &zoom=14&size=300x400&maptype=roadmap "
                                       " &markers=color:blue%7Clabel:S%7C "
                                       " 49.676000,18.184500 "
                                       " &markers=color:green%7Clabel:S%7C "
                                       " 49.682600,18.178000 "
                                       " &markers=color:orange%7C "
                                       " 49.672600,18.177300 "
                                       " &markers=color:green%7Clabel:G%7C "
                                       " 49.682300,18.179200 "
                                       " &markers=color:green%7Clabel:G%7C "
                                       " 49.672000,18.179200 "
                                       " &markers=color:red%7Clabel:C%7C "
                                       " 49.676000,18.178000 "
    call WebViewer1 .GoToUrl
                    url  get global staticMapURL
    set Screen1 . Title to  " AI2 Static Map with Fancy Pins "
    set Label1 . Text to  " AI2 Static Map with Fancy Pins "
```
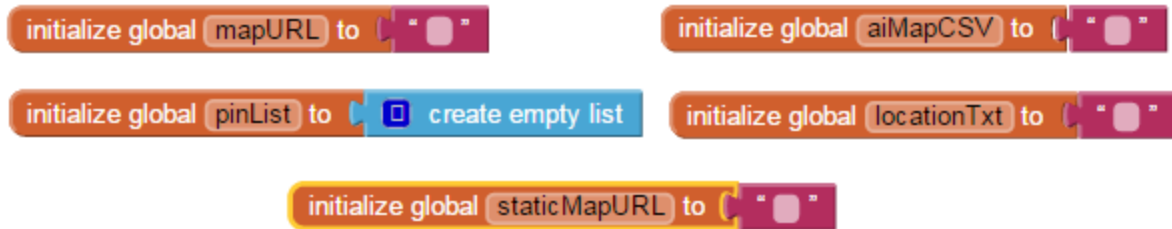
A bit of explanation:

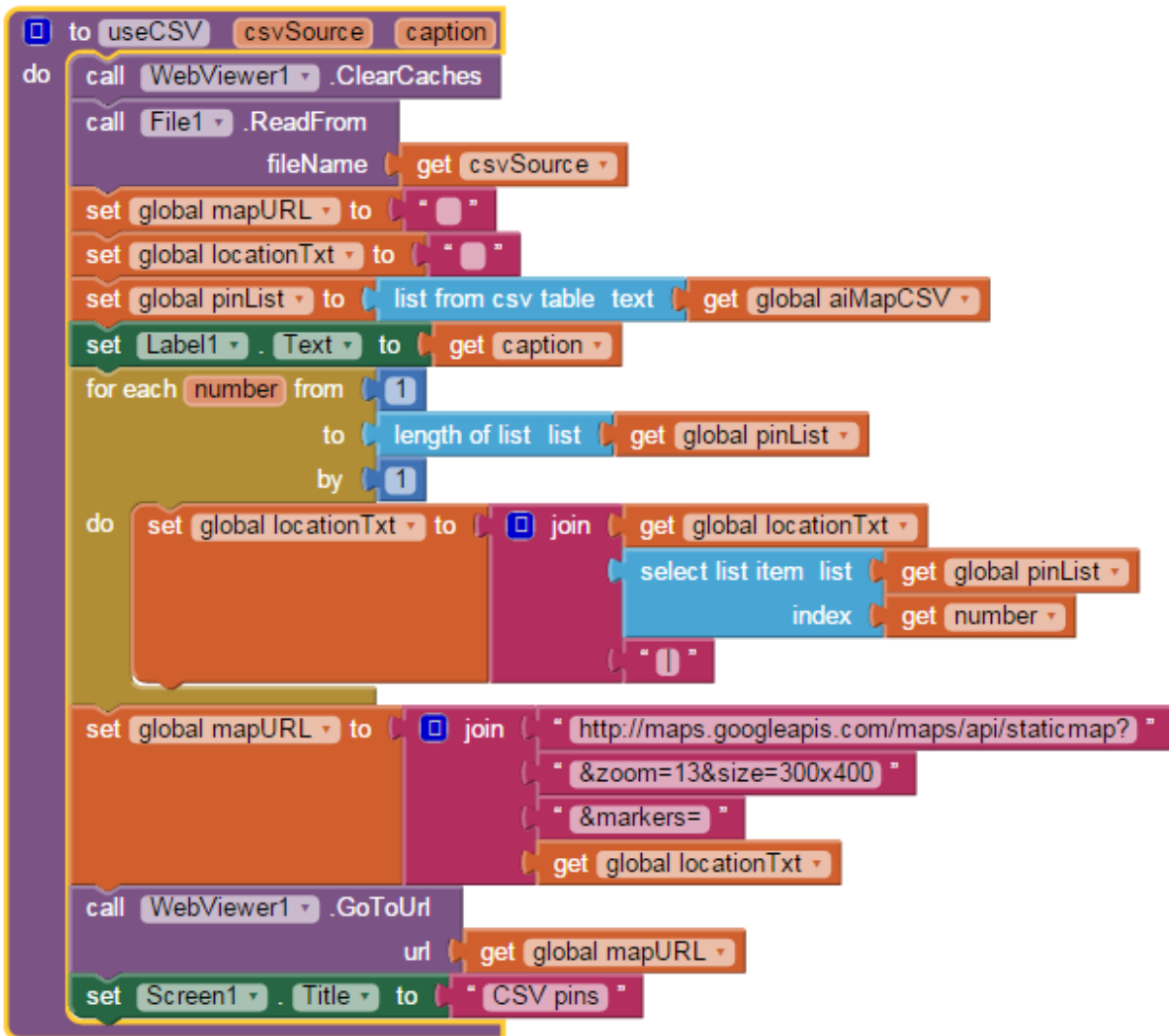| Block	Explanation
`http://maps.googleapis.com/maps/api/staticmap?center=`	
`49.676000,18.184500`	coordinates of the map's center
`&zoom=14&size=300x400&maptype=roadmap`	set zoom to 14, set map to display at 300x400 pixels and the maptype to roadtype
`&markers=color:blue%7Clabel:S%7C`	Describes the markers. Color is blue, followed by the code to add a label (%7Clabel:S%7C where the label is "C"
`49.676000,18.184500`	
`&markers=color:green%7Clabel:S%7C`	a green label
`49.682600,18.178000`	
`&markers=color:orange%7C`	an orange label
`49.672600,18.177300`	
`&markers=color:green%7Clabel:G%7C`	Green and annotated as pin "G"
`49.682300,18.179200`	
`&markers=color:green%7Clabel:G%7C`	another "G"
`49.672000,18.179200`	
`&markers=color:red%7Clabel:C%7C`	another "C" pin

The *makeStaticMapPins2* Procedure produces a Static Map with the default Google red pins. Use a pipe (|), the key usually on the shifted backslash key, to separate coordinates as shown. This procedure is nearly identical to the procedure that generates the fancy pins.

```
to makeStaticMapsPins2
do  set global staticMapURL to  join  "http://maps.googleapis.com/maps/api/staticmap?"
                                       "&zoom=14&size=300x400"
                                       "&markers="
                                       "49.676000,18.184500"
                                       "|"
                                       "49.682600,18.178000"
                                       "|"
                                       "49.672600,18.177300"
                                       "|"
                                       "49.682300,18.179200"
                                       "|"
                                       "49.672000,18.179200"
                                       "|"
                                       "49.676000,18.178000"

    call WebViewer1.GoToUrl
      url  get global staticMapURL
    set Label1.Text to  "AI2 Static Map with Simple Pins"
    set Screen1.Title to  "AI2 Static Map with Simple Pins"
```

The *useCSV* Procedure shows how to generate the url required to make a Static Map from a csv table location file.

```
to useCSV  csvSource  caption
do   call WebViewer1 .ClearCaches
     call File1 .ReadFrom
                    fileName  get csvSource
     set global mapURL to  " O "
     set global locationTxt to  " O "
     set global pinList to  list from csv table text  get global aiMapCSV
     set Label1 . Text to  get caption
     for each number from  1
                       to  length of list list  get global pinList
                       by  1
     do   set global locationTxt to  join  get global locationTxt
                                           select list item list  get global pinList
                                                          index  get number
                                           " | "
          set global mapURL to  join  " http://maps.googleapis.com/maps/api/staticmap? "
                                      " &zoom=13&size=300x400 "
                                      " &markers= "
                                      get global locationTxt
     call WebViewer1 .GoToUrl
                     url  get global mapURL
     set Screen1 . Title to  " CSV pins "
```

The above code constructs the required url from csv data. The blocks gather (*for each number* …) the pin locations from the *pinList*, placing the location data in a text file and separating the individual locations with a "pipe." This file of locations is joined with the required header information (zoom,size etc) to create the required *mapURL*. The *pinList* is made from one of the two csv files stored in Media. The calling routine (see Button1) requires the name of the csv (csvSource) and a caption for the map (caption).

The CSV File

The example app uses a List to temporarily store the pin coordinates information. There are several ways to create a suitable csv file to populate the pin coordinates List. This tutorial stores the locations in a table csv.

A *Notepad* created table csv "looks" like this:

Two correctly formatted csv files required in the app are included in the AI2 Media of the example aia file.

Is an aia File included?

Sure, look for AI2StaticMaps_Template.aia on the main blog page.   This aia is a coding head start. No finished project aia is included. Try some blocks.   Many developers may only need the two csv files from the aia; others will use the template to accelerate coding.

Maps AI2 - 102 Preview

How do you integrate the real time GPS location of the device on a Static Map?   Gosh, how
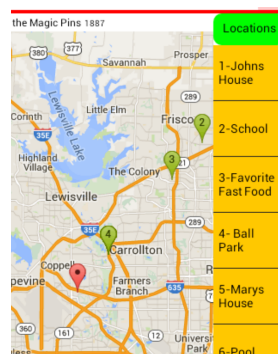
do you get this map   where the red pin is the current location? The *Magic Map Pins* tutorial will show one way to make the static map active--coming soon.