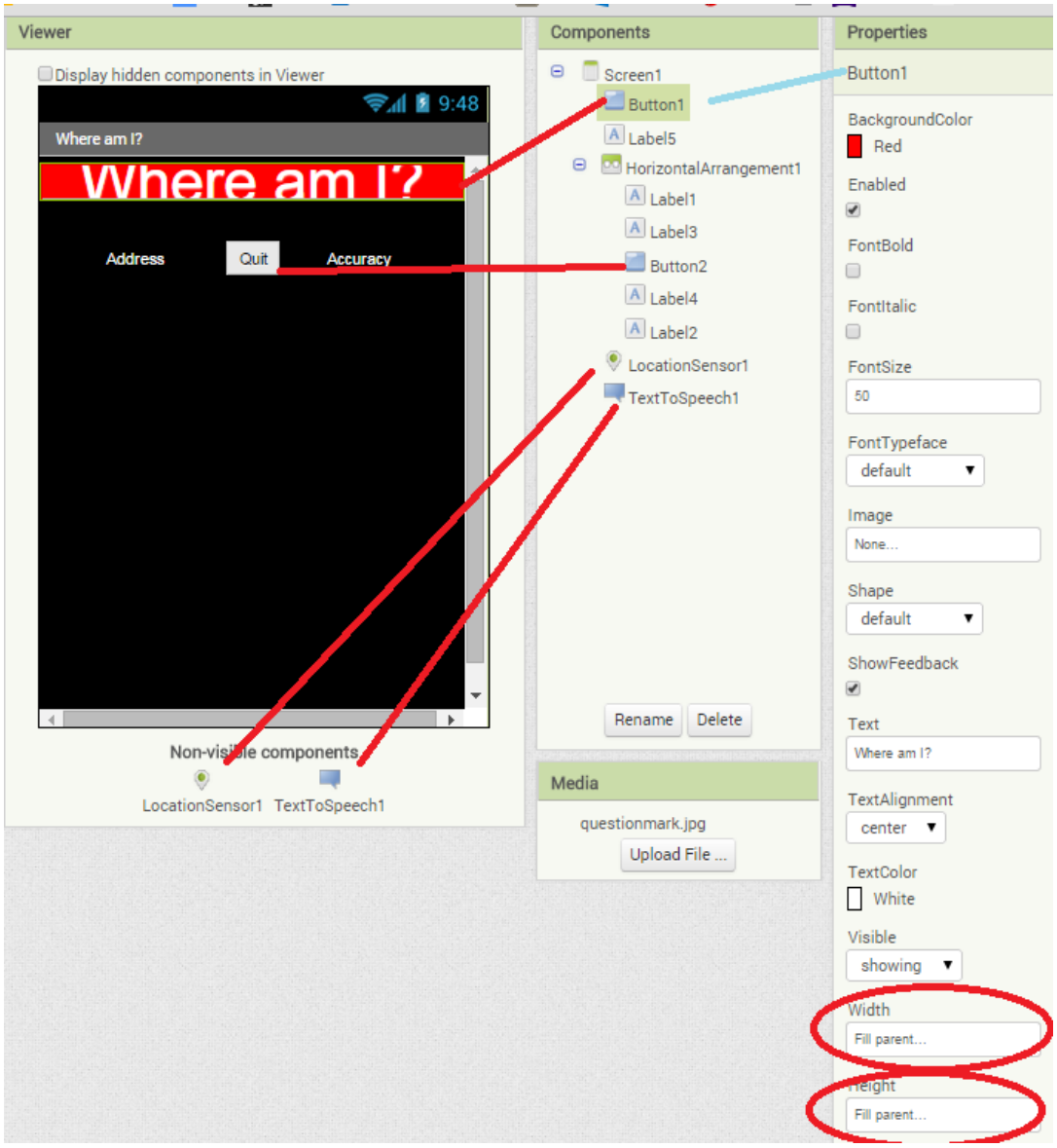
Where am I?

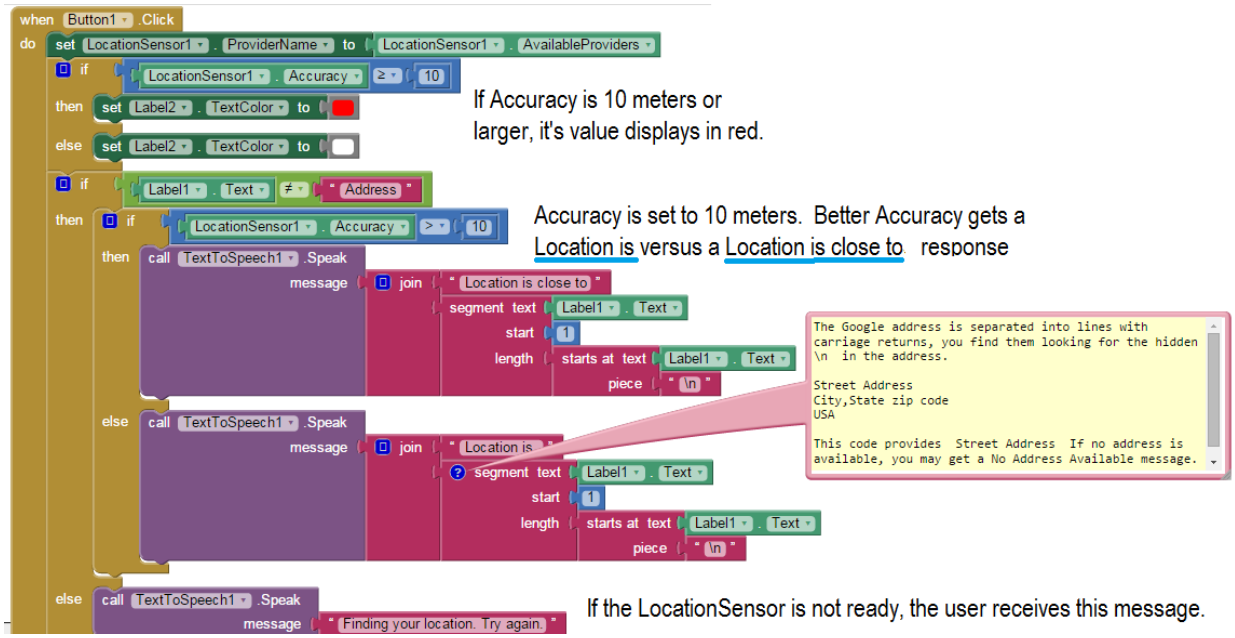
Where am I? is a simple *App Inventor* app that tells you where you are. Well, sometimes because Google does not have a list of all street addresses; sometimes all Google knows is **No Address Available.** Yes, this app talks to you if your *Android* has Text To Speech capability.

Here is the *Designer* screen. There is nothing special. Make the screen look like you want it to display. The following shows how to get a display like the above.

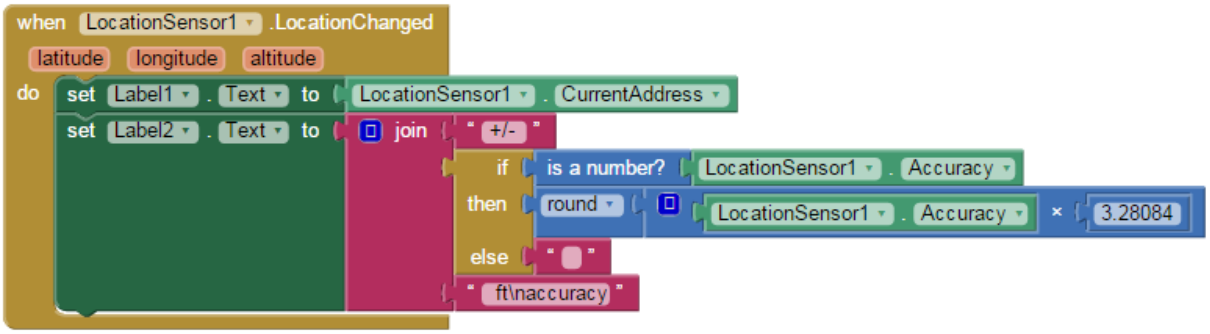


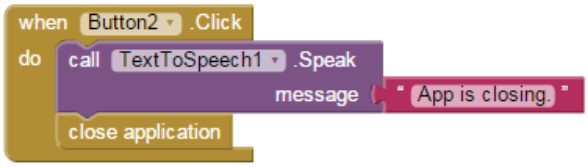
Labels3,4 and 5 are used as 'spacers.' Their text color is set to None. The large red button Width and Height are set to *Fill parent…* The image in the Media is used to set the app's icon; you do not need it...use your own 48x48 pixel jpg or png image. Note that several objects are within a Horizontal Arrangement.

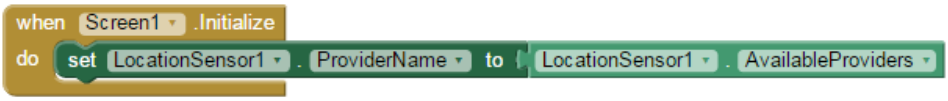
Here are the code blocks.



The following are the other blocks needed:







If you prefer to report Accuracy in meters, eliminate the Math block and change the text to refer to m instead of ft.

A routine is provided to convert meters to feet (I live where feet is understood). AI2's LocationSensor has a property called *Accuracy* that reports the reliability of the positional information the device's GPS or network provides; *Accuracy* is in +/- meters (one meter equals 3.2808 feet).

Without setting the *LocationSensor.TimeInterval* with blocks, the app will update its location about once a minute using the GPS receiver (if your device has one). The GPS receiver may take from 2 to 20 seconds or more to update, depending on many factors. It makes little sense to update more frequently than every 30 seconds. The default timing for updating the sensor, appears to be 60 seconds (60000 ms) for most if not all devices. If you need to change this interval or experiment, you probably should place the appropriate block in the *Screen1.Initialize* event handler.

Set **ProviderName** to **LocationSensor1.AvailableProviders** and the app will use what 'Provider' is available with probably this priority 1) gps, 2) network and 3) others. If the device can not get a gps fix (or the gps is turned off in the app, the sensor will look at network etc. to provide a location). All things being equal, gps will almost always provide more accurate location information. So optionally, set the Provider to gps. Be aware, if the device is in a building, under trees or if it is raining, a gps fix might not be better than a network location because a gps fix becomes difficult in those situations and might not be possible at all. If only three satellites usable satellites are received and used as a fix, the accuracy of the fix is diminished significantly below the capabilities of your device. The sensor's *Accuracy* property lets you know when the location is outside of parameters you can set. Here, 10 meters is used. Experiment with this (never going lower than 2 meters because then you are approaching the resolution of the GPS receiver in your device.

> *AI2 has no control to tell you how many satellites are being used in a fix or how strong the received signals are at the GPS. Instead, AI2 uses other information from the satellite data stream to establish a reliability circle, a distance circle with a radius of the indicated accuracy. It only reports the radius of that circle as described below. Your device may be in the center of the circle, you may be on the edge of the reliability circle or in the center. Accuracy demonstrates the quality of the satellite fix by reporting plus/minus x meters accuracy at any moment.* ***Lower values are better****.*
>
> *The Android Developer Reference defines Accuracy as "the radius of 68% confidence." If you draw a circle centered at this location's latitude and longitude, and with a radius equal to the accuracy, then there is a 68% probability that the true location is inside the circle. In statistical terms, it is assumed that location errors are random with a normal distribution, so the 68% confidence circle represents one standard deviation. In practice, location errors do not always follow such a simple distribution. This accuracy estimation is only for horizontal accuracy, and does not indicate the accuracy of altitude if those are reported. If the location does not have an accuracy, then 0.0 is returned.*

If you are not following this discussion, perhaps you should read the LocationSensor tutorial here http://appinventor.mit.edu/explore/ai2/tutorials3818.html?&page=2 .

There is no aia accompanying this tutorial; everything you need is in the article.

The app has some error control; depending on how you intend to use this app, you may have to include additional logic blocks. I am not sure how it will respond if Google returns a CurrentAddress of **No Address Available** (Sorry, I am unable to elicit that response). Users in rural areas, in some metropolitan areas with limited Google mapping etc. may get the no address response. No Address is a Google issue; not an AI issue that occurs when mapping information is unavailable or when the LocationSensor can not get a fix or a usable address. Users might require a logic block like (in pseudo code) CurrentAddress = No Address Available then speak No Address Available as one of the options if Google does not provide a carriage return following the word Available.

Isn’t it great to know where you are?

This app may be easily adapted to send a text message relating your whereabouts to others using the Texting component and the text provided by the Location sensor placed in the Texting component.