MagicTrick: Your First App Inventor App

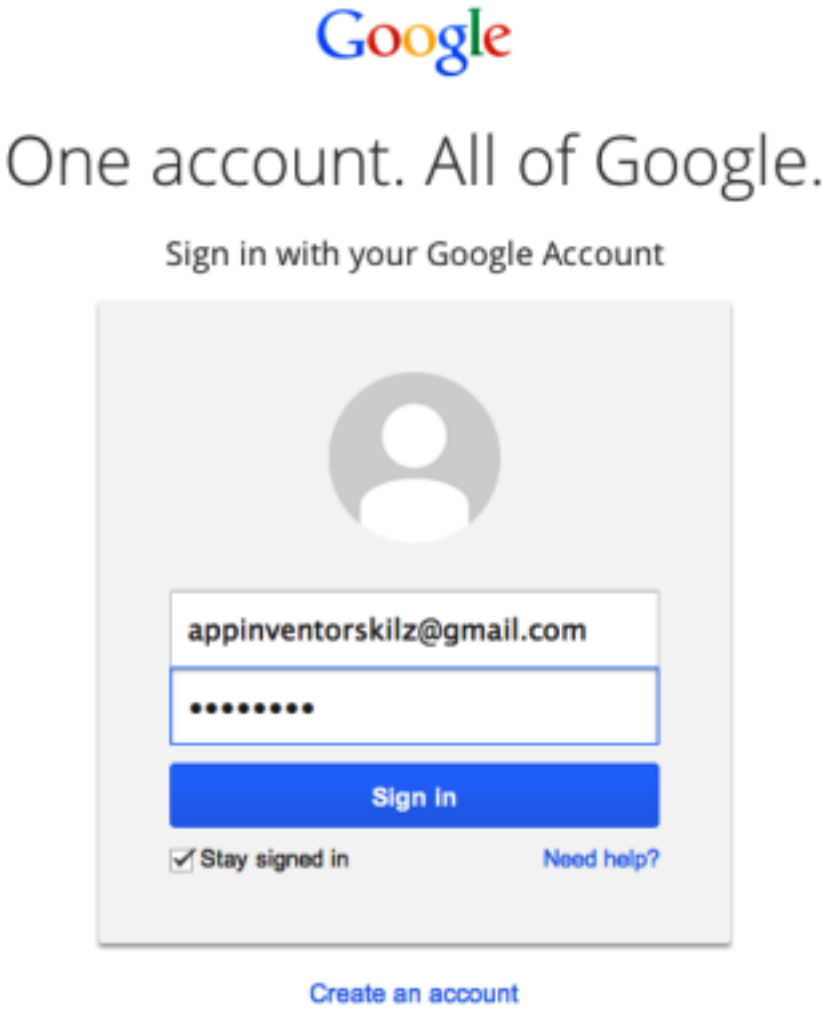
This step-by-step picture tutorial will guide you through making a magic trick app to make a rabbit appear in a top hat.

To get started, go to App Inventor on the web.

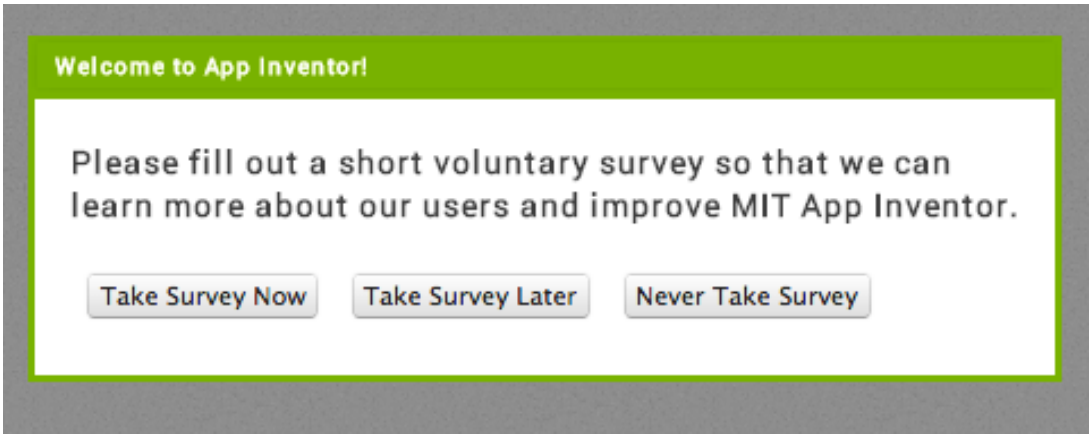
Go directly to **ai2.appinventor.mit.edu,** or click the orange "Create" button from the App Inventor website.

Log in to App Inventor with a Gmail (or Google) user name and password.

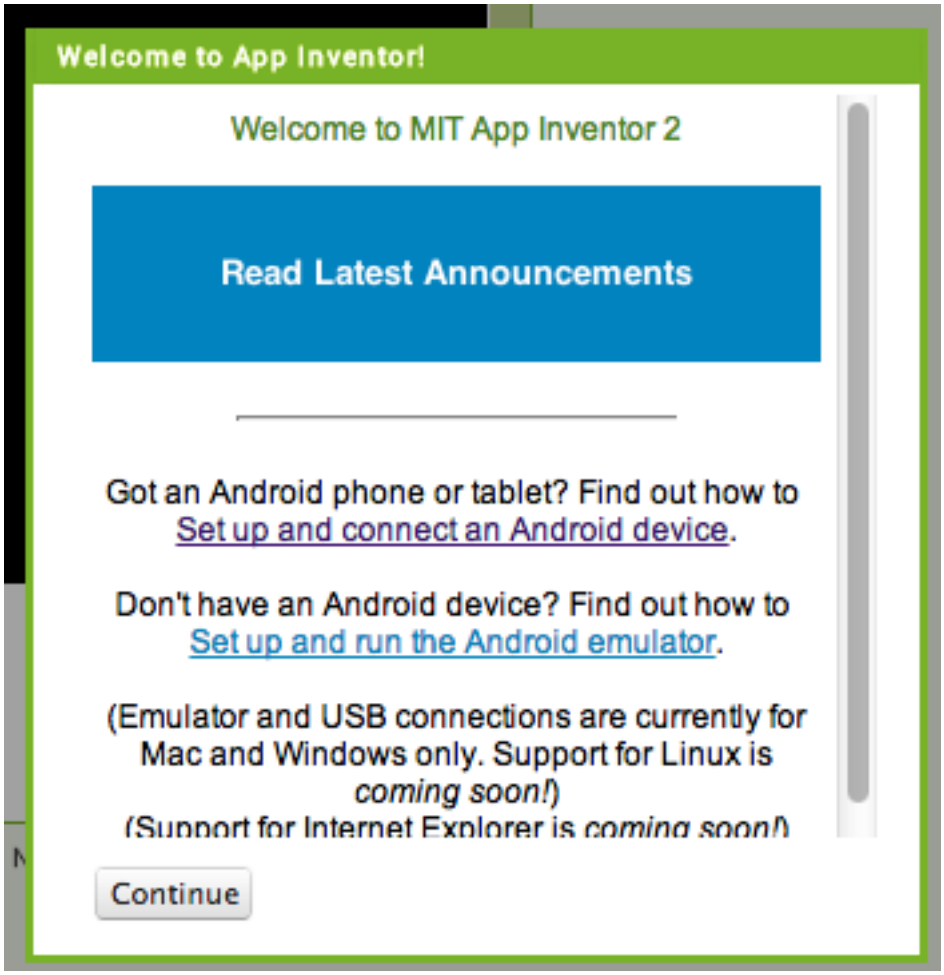
Use an existing gmail account or school-based google account to log in to ai2.appinventor.mit.edu
To set up a brand new gmail account, go to accounts.google.com/SignUp



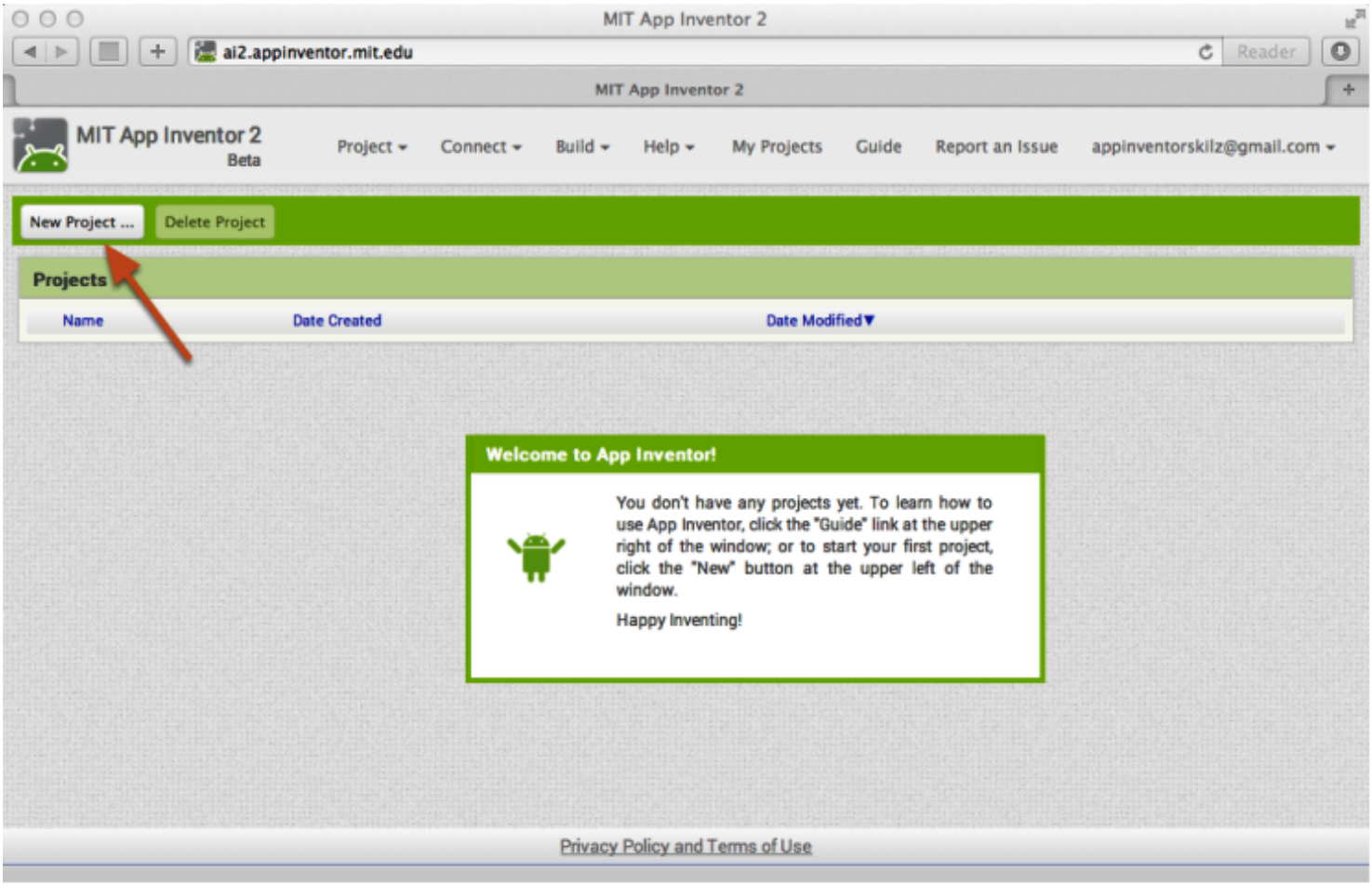
Feel free to take the survey, but click Take Survey Later if you want to start making an app now.



Click "Continue" to dismiss the splash screen.

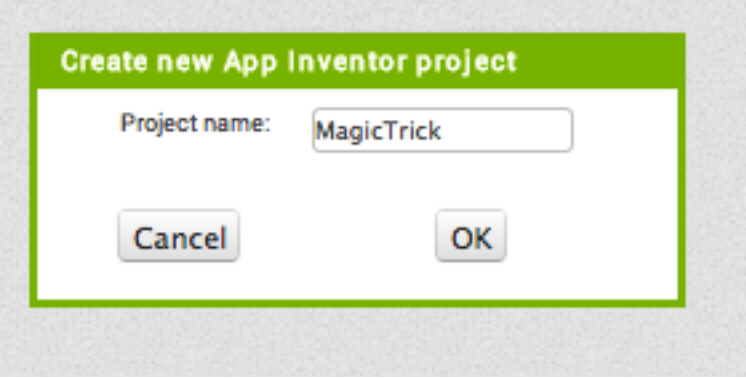


Start a new project.



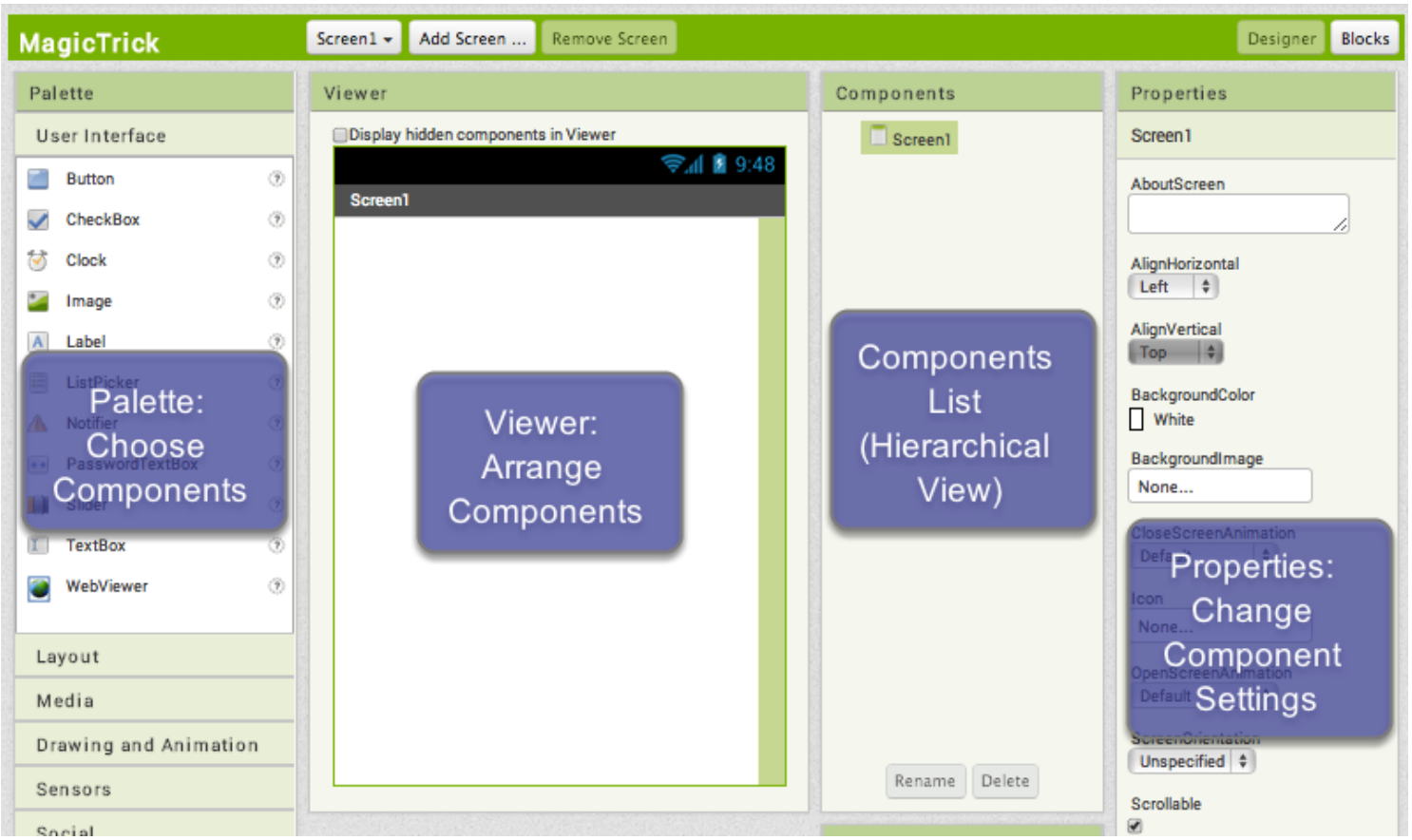
Name the project "MagicTrick" (no spaces!)

Type in the project name (underscores are allowed, spaces are not) and click OK.



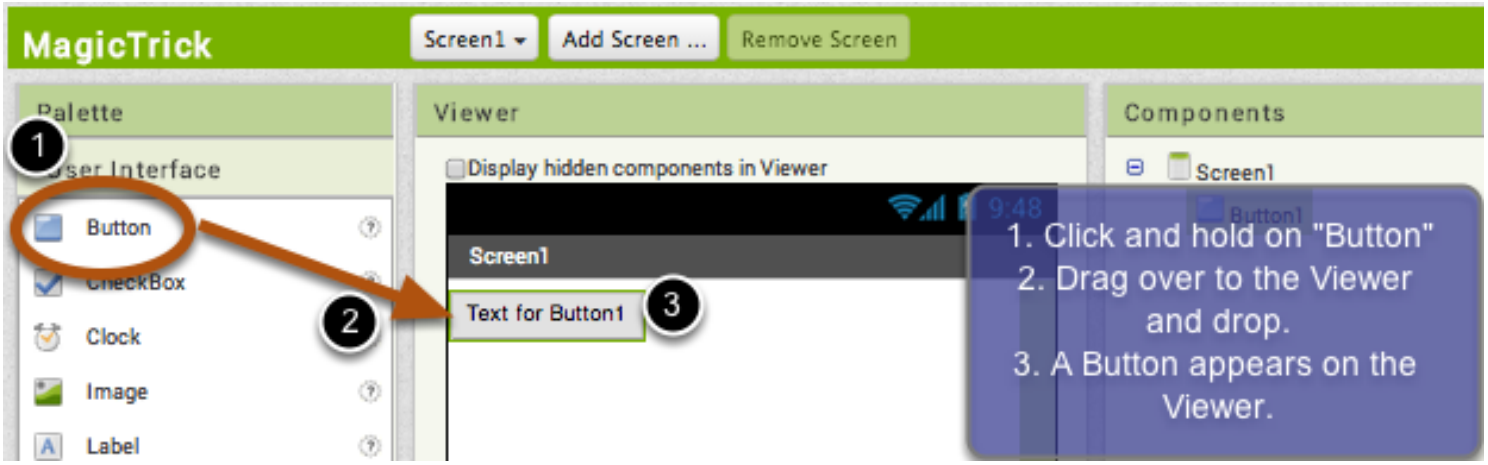
You are now in the Designer, where you lay out the "user interface" of your app.

The Design Window, or simply "Designer" is where you lay out the look and feel of your app, and specify what functionalities it should have. You choose things for the user interface things like Buttons, Images, and Text boxes, and functionalities like Text-to-Speech, Sensors, and GPS.



Add a Button.

Our project needs a button. ***Click and hold*** on the word "Button" in the palette. ***Drag*** your mouse over to the Viewer. ***Drop*** the button and a new button will appear on the Viewer.

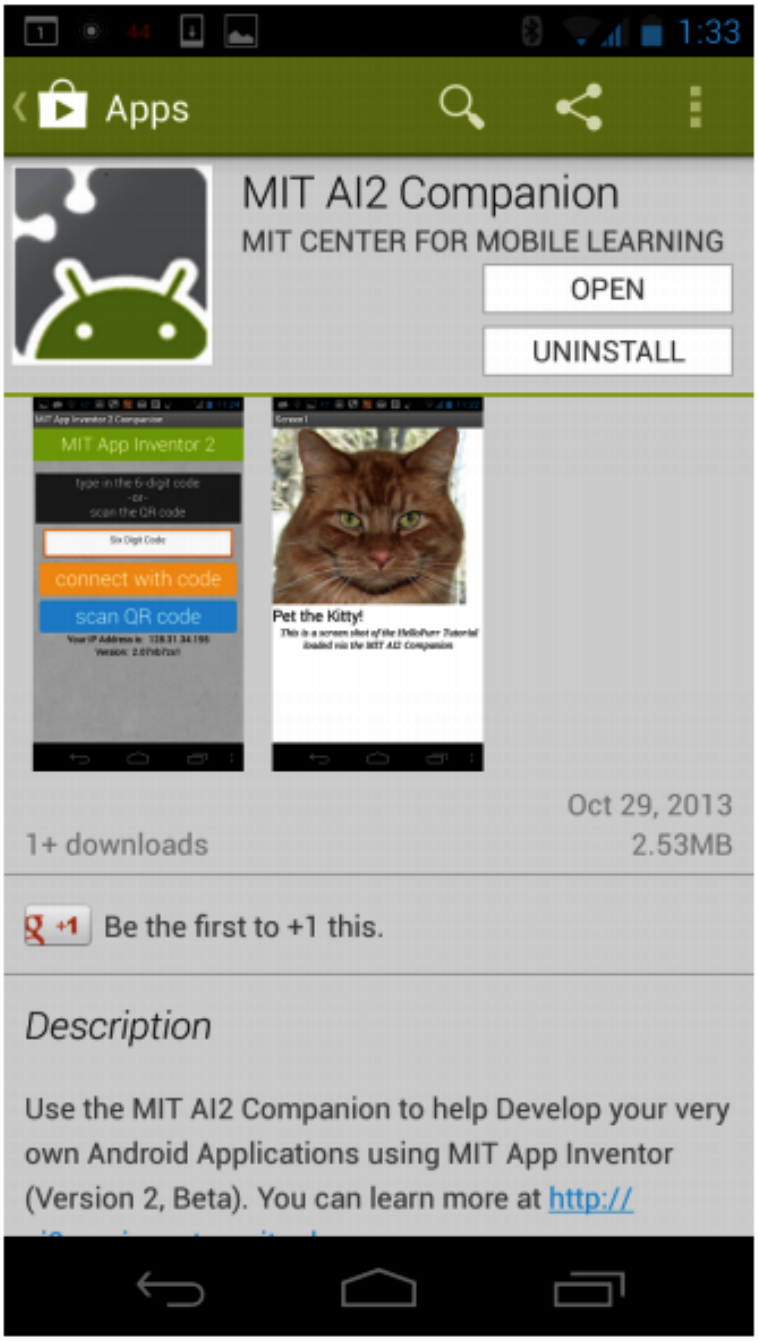


Connect App Inventor to your phone for live testing.

One of the neatest things about App Inventor is that you can see and test your app while you're building it, on a connected device. If you have an **Android phone or tablet, follow the steps below**. *If you do not have a device*, then follow the instructions for setting up the on-screen emulator (opens a new page) and then come back to this tutorial once you've gotten the emulator connected to App Inventor.

Get the MIT AI2 Companion from the Play Store and install it on your phone or tablet.

The preferred method for getting the AI2 Companion App is to **download the app from the Play Store by searching for "MIT AI2 Companion"**.



To download the AI2 Companion App to your device directly (SKIP THIS STEP IF YOU already got the app from Play Store)

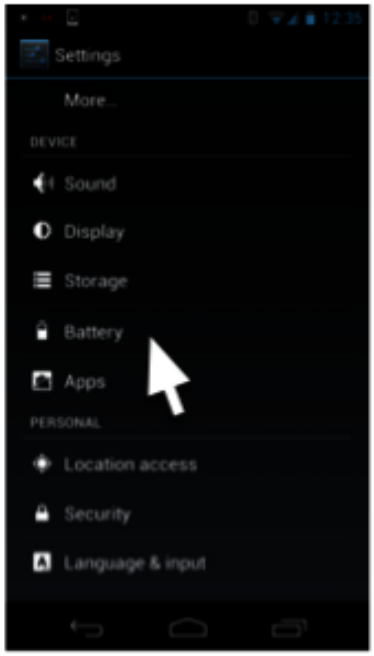
If for some reason you can not connect to the Google Play store, you can download the AI2 Companion as described here.

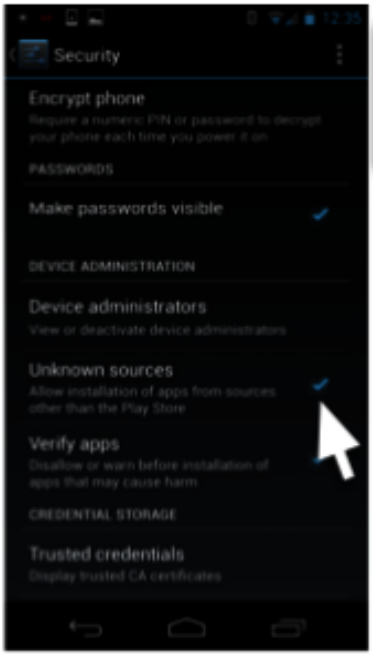
First, you will need to go into your phone's settings (#1), choose "Security", then scroll down to allow "Unknown Sources", which allows apps that are not from the Play Store to be installed on the phone.

Second, do one of the following:
A) **Scan the QR code above (#2)**

or

B) Click the "Need help finding..." link and you'll be taken to the download page. From there you can download the MITAI2Companion.apk file to your computer and then move it over to your device to install it.





SKIP THIS STEP if you already got the AI2 Companion from the Play Store

1

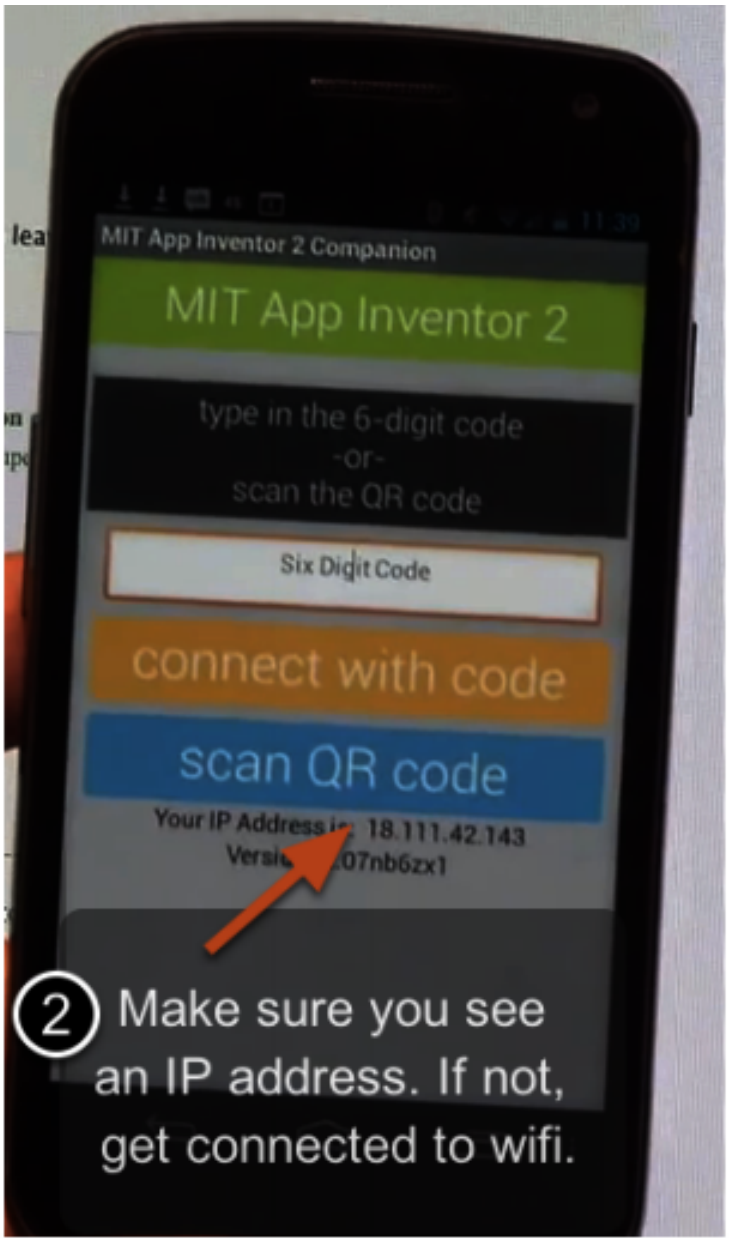
1. Open your phone's settings and click "Security".
2. CHECK the box for "Unknown sources"

2

Scan to download MIT AI2 Companion directly to phone

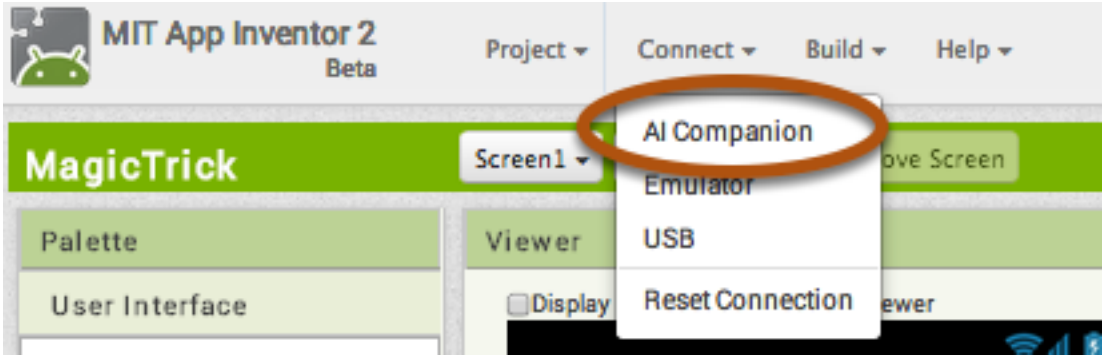
Start the AICompanion on your device

On your phone or tablet, click the icon for the MIT AI Companion to start the app. NOTE: Your phone and computer must both be on the same wireless network. Make sure your phone's wifi is on and that you are connected to the local wireless network. If you can not connect over wifi, go to the Setup Instructions on the App Inventor Website to find out how to connect with a USB cable.



Let's Connect!

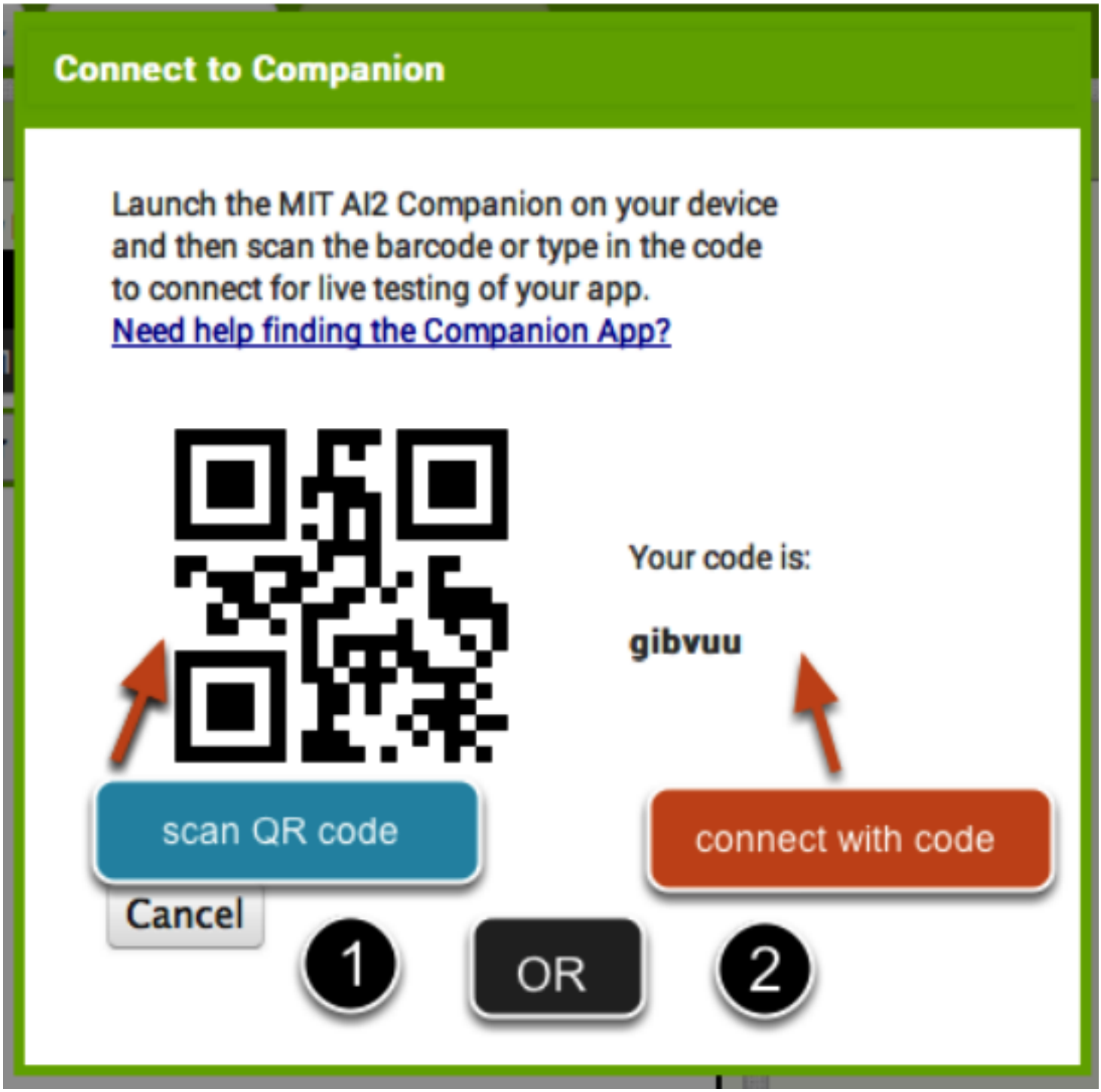
On the Connect menu, choose "AI Companion"



Get the Connection Code from App Inventor and scan or type it into your Companion app

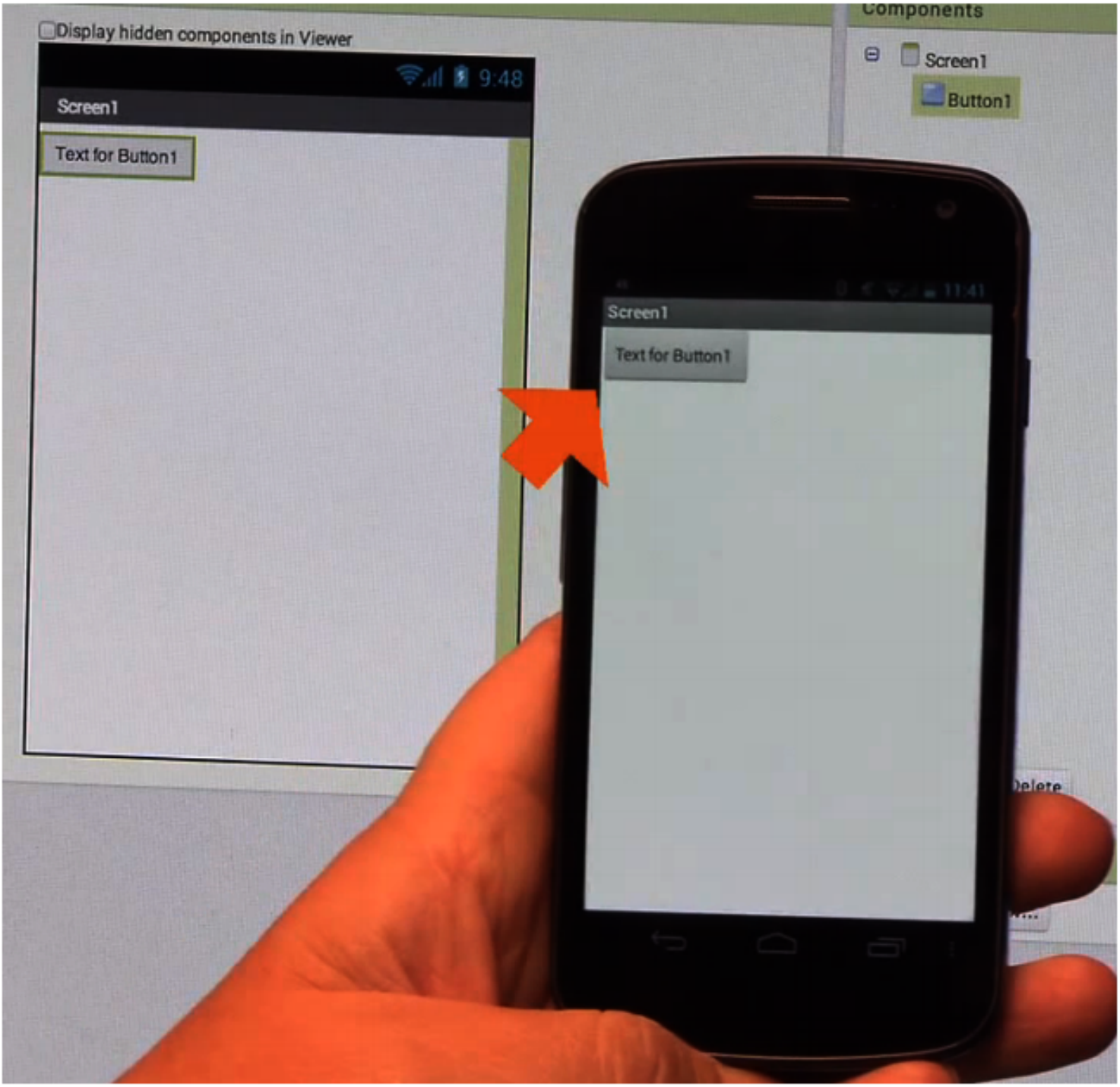
You can connect by:

1 - Scanning the QR code by clicking "Scan QR code" (#1).
or
2 - Typing the code into the text window and click "Connect with code" (#2).



See your app on the connected device

You will know that your connection is successful when you see your app on the connected device. So far our app only has a button, so that is what you will see. As you add more to the project, you will see your app change on your phone.



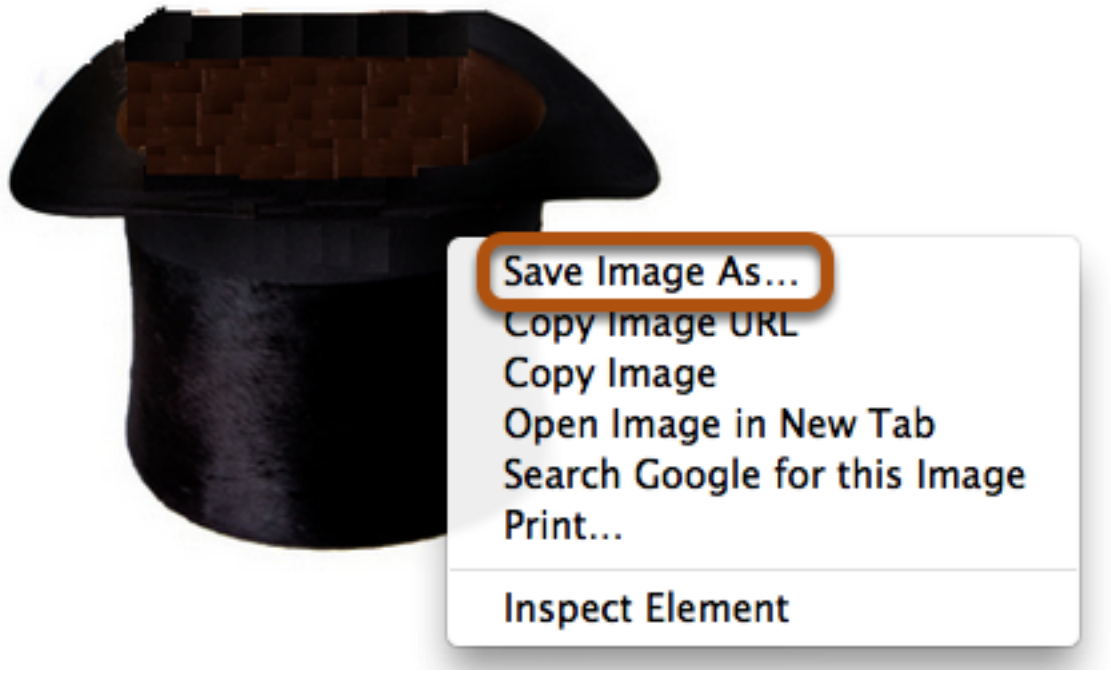
Download Images.

For this MagicTrick, you'll be clicking the button and changing a hat image to be an image of rabbit in a hat. Follow both of these links to open up the images in a new window.

1. Hat Image
2. Rabbit Image

In the new window, right click on the image. Depending on which browser (Chrome, Firefox, Safari, Internet Explorer) you are using, it will display a different set of options. Select the one most similar to "Save/Download Image as...". This will create a pop-up window for you to save the image with a new name. Keep the name as hat.jpg and rabbit.jpg

---

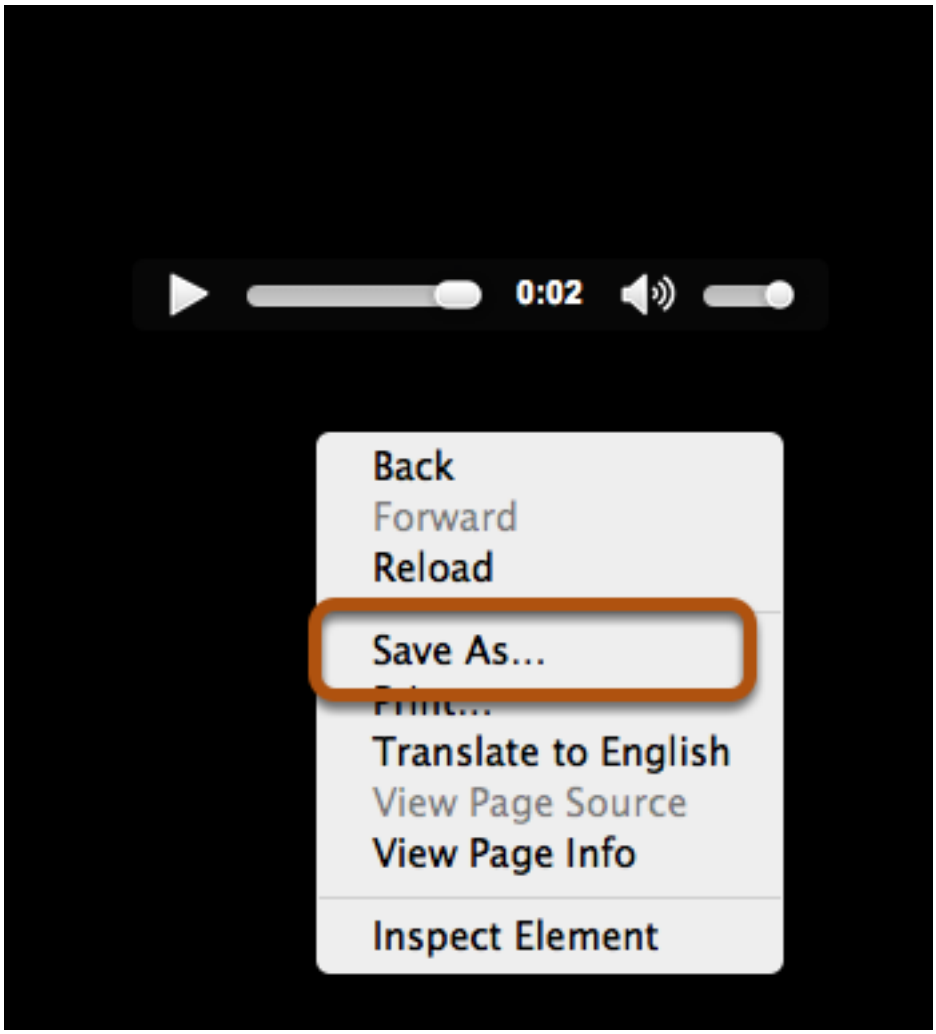


Download Sound.

This MagicTrick app will also involve a sound. Click the following link to download the "Ta-Da" sound for your app.

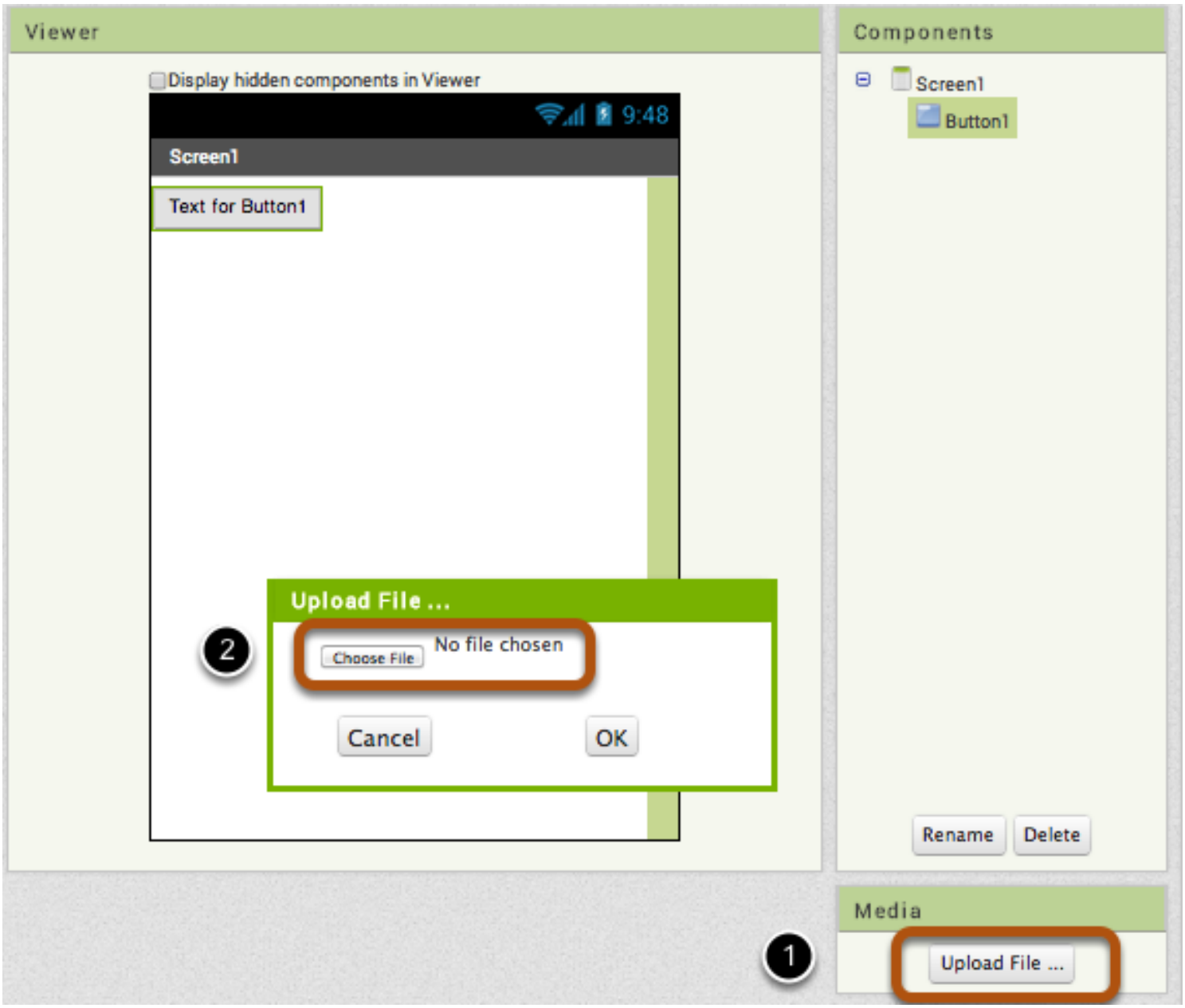
1. Ta Da Sound

In the new window, right click on the image. Depending on which browser (Chrome, Firefox, Safari, Internet Explorer) you are using, it will display a different set of options. Select the one most similar to "Save/Download as...". This will create a pop-up window for you to save the sound with a new name. Keep the name as TaDasound.mp3.



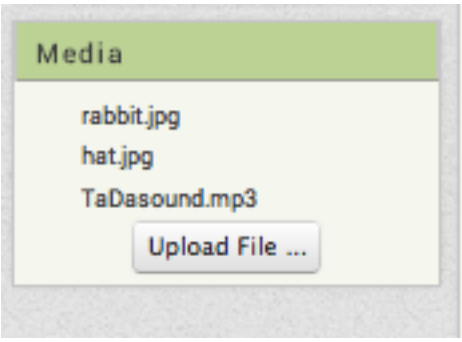
Uploading Media To The Project

Now we have to put all of this media into our app. On the Designer screen, find the Media tab and click Upload File. Choose the file you want to upload. Do this three times (for the two images and the one sound).



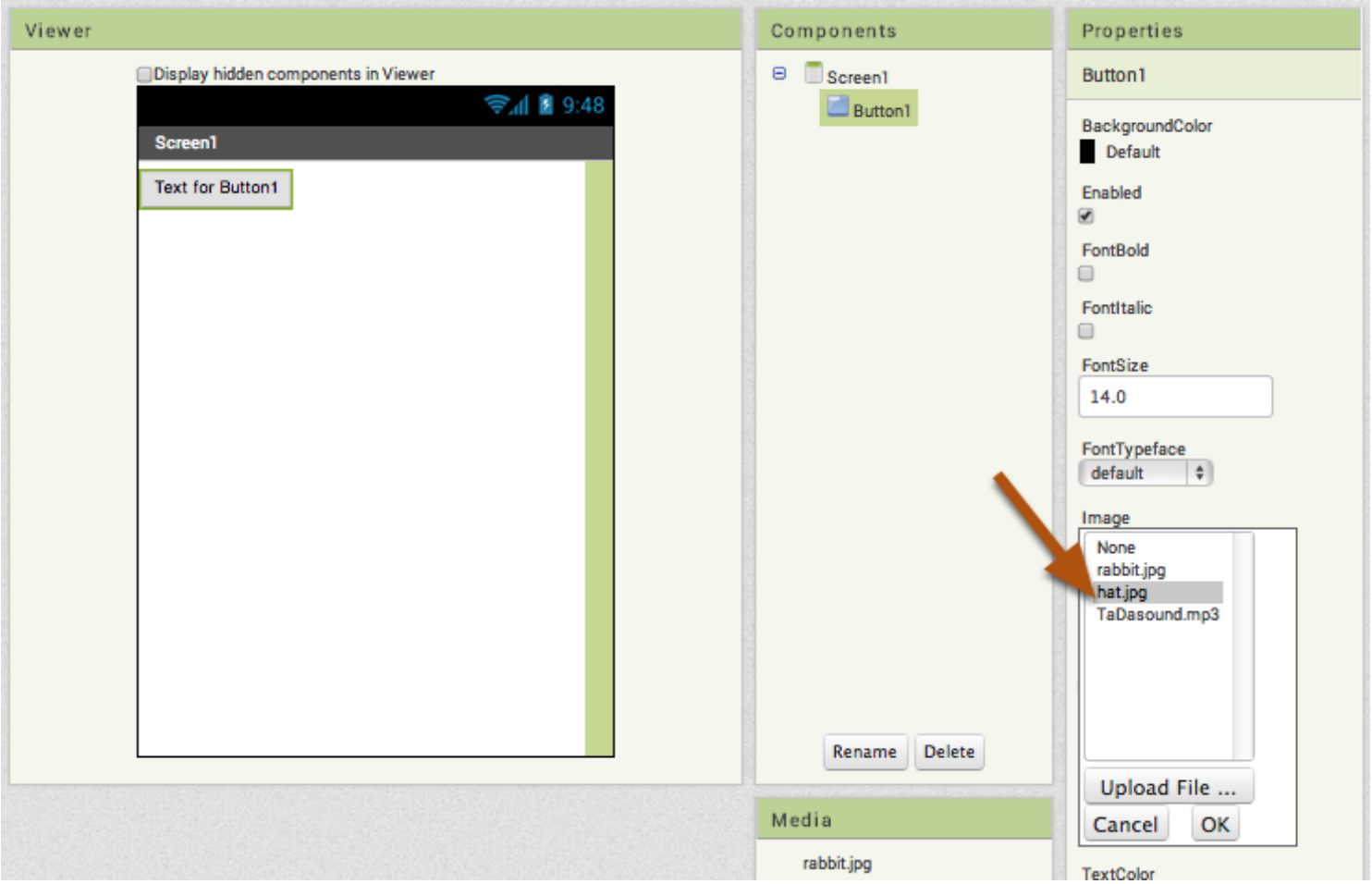
Uploaded Media: What It Should Look Like

Double check that your Media tab now contains three things: rabbit.jpg, hat.jpg, and TaDasound.mp3



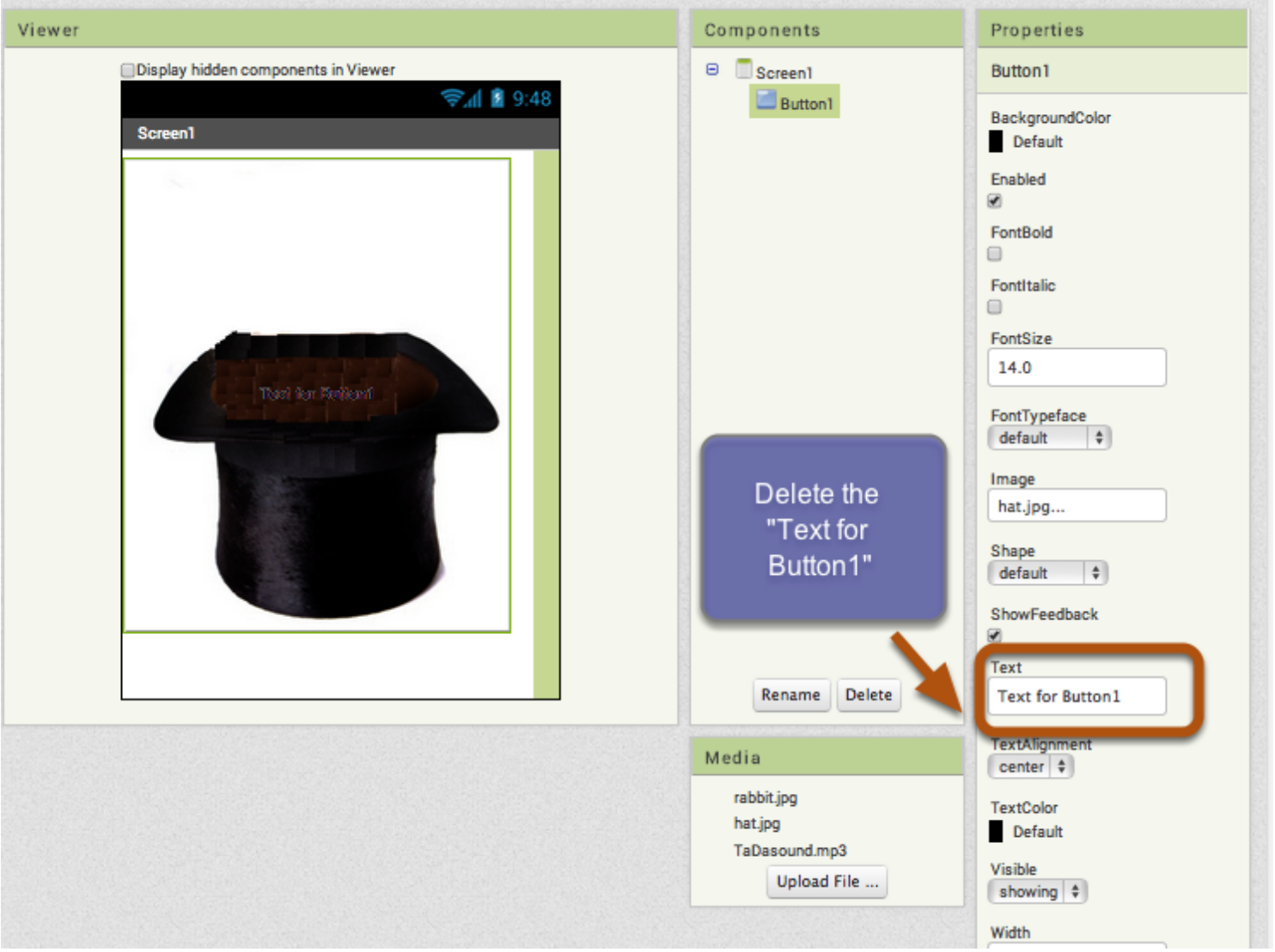
Change the Picture on the Button

Under the “Properties” pane for Button1, we see “None…” under the Image tab. Click on it and choose “hat.jpg”. Click OK. Notice that the image on your app's button changes right away.



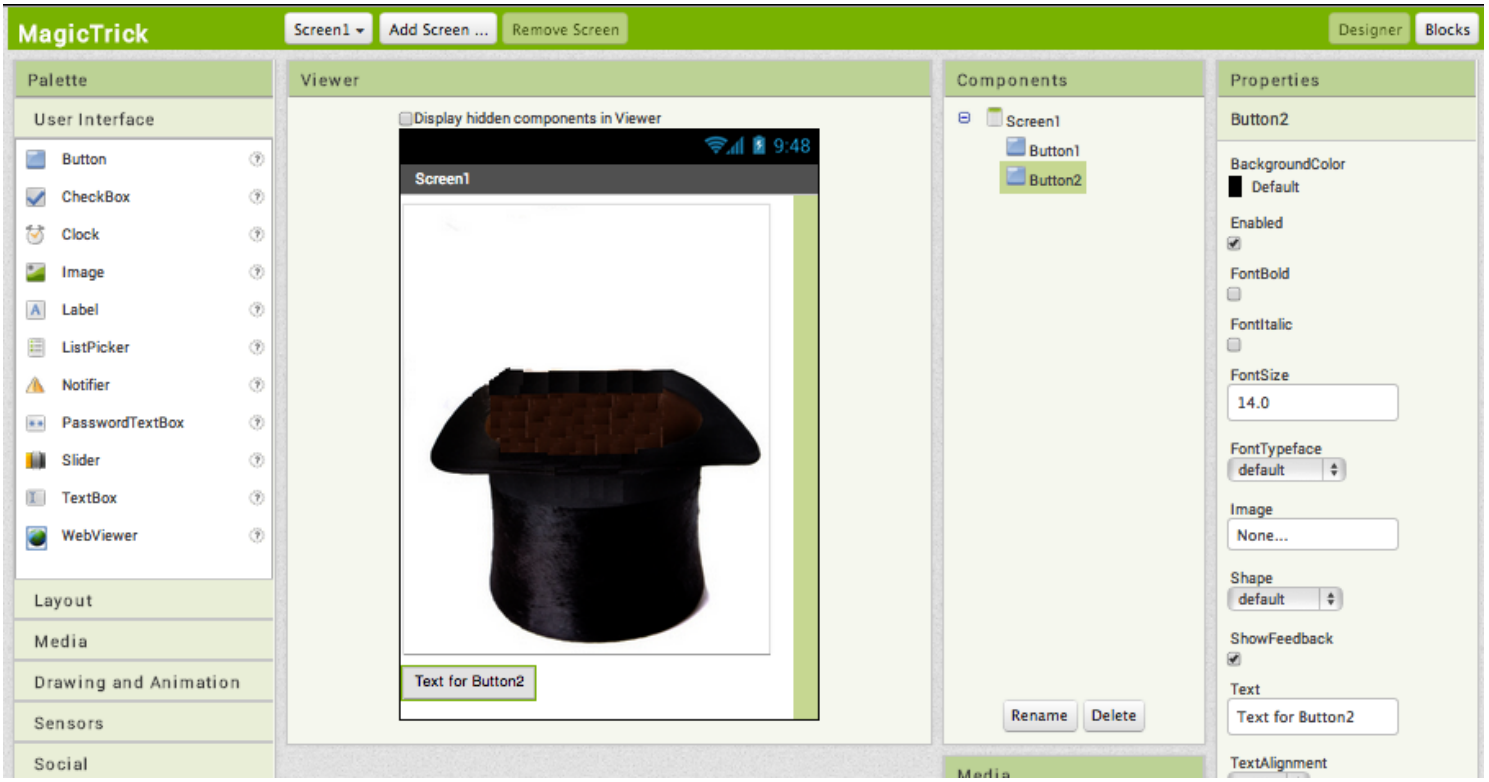
Changing the Text on the Button

Your button should now should a black top hat. It will also still contain the original text that said "Text for Button1". Under the "Properties" pane for Button1, go to Text and delete the "Text for Button1". Hit enter after changing the text.



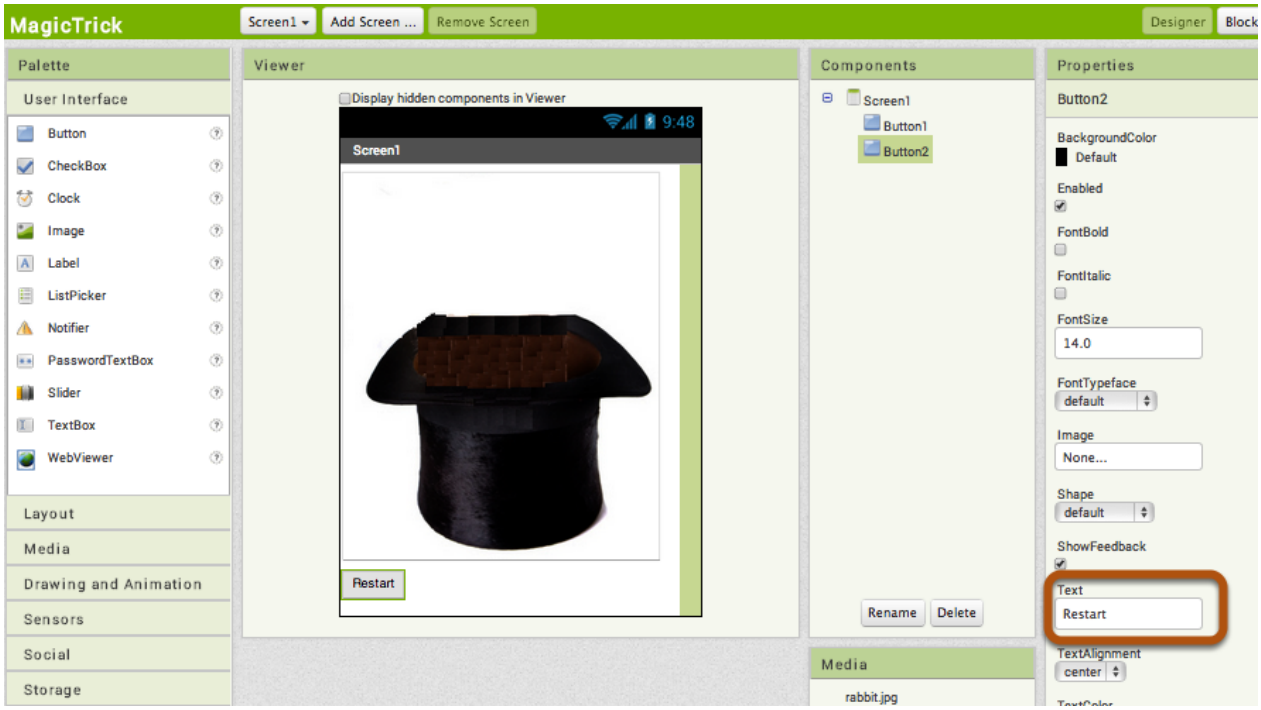
Make Another Button

Now let’s make another button. Remember to click and drag the button onto the Viewer.



Setting the Properties of the Button

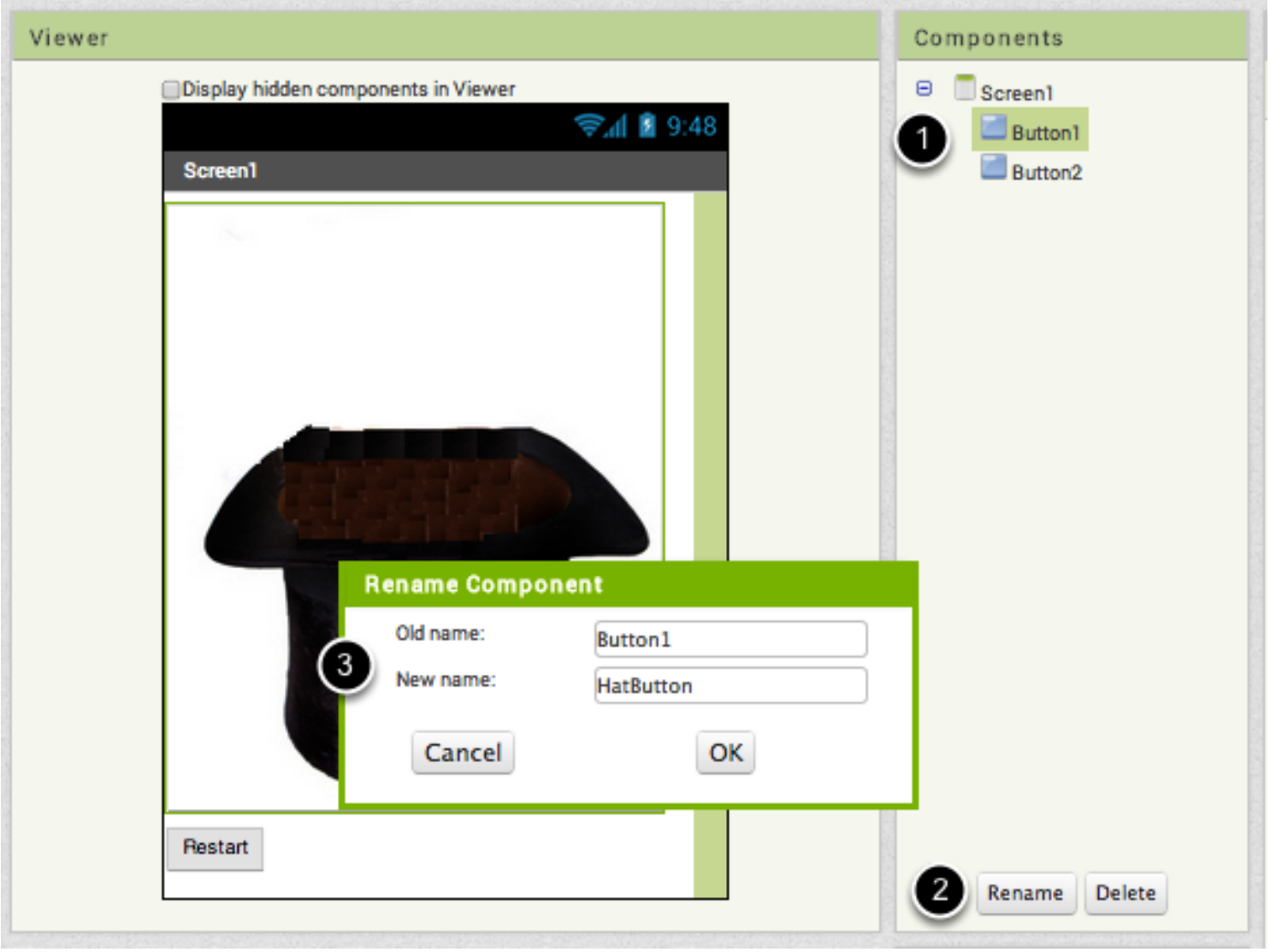
Unlike last time, we are not going to fill this button with an image. Instead, we will go to the "Properties" pane of Button2, and change its text to “Restart”. Hit enter after changing the text. Notice how the text on your app changes right away.



Changing the Names

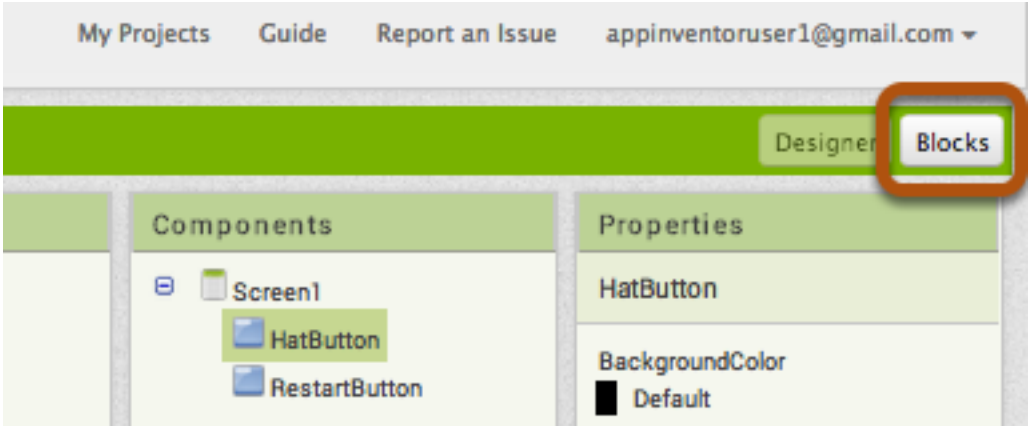
It can often get confusing when you have more than one type of the same component. To make it easier for ourselves, let's rename our Buttons to HatButton from Button1 and RestartButton from Button2. To rename components, follow these steps:

1. Click the component. [*Make sure it is outlined in green in the list of components*]
2. Click the rename button.
3. Type in the new name and click OK.



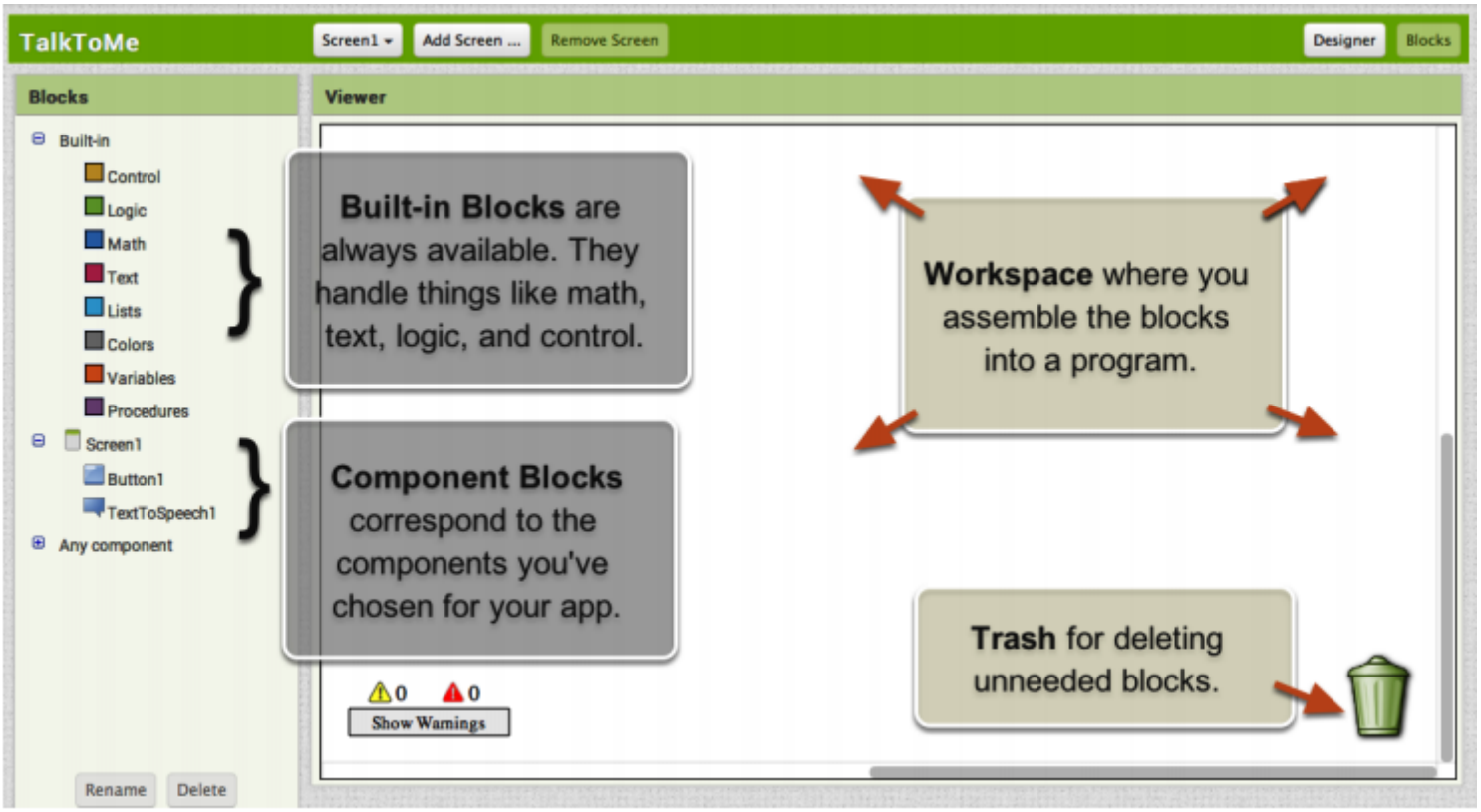
Switch over to the Blocks Editor

It's time to tell your app what to do! Click "Blocks" to move over to the Blocks Editor. Think of the Designer and Blocks buttons like tabs -- you use them to move back and forth between the two areas of App Inventor.



The Blocks Editor

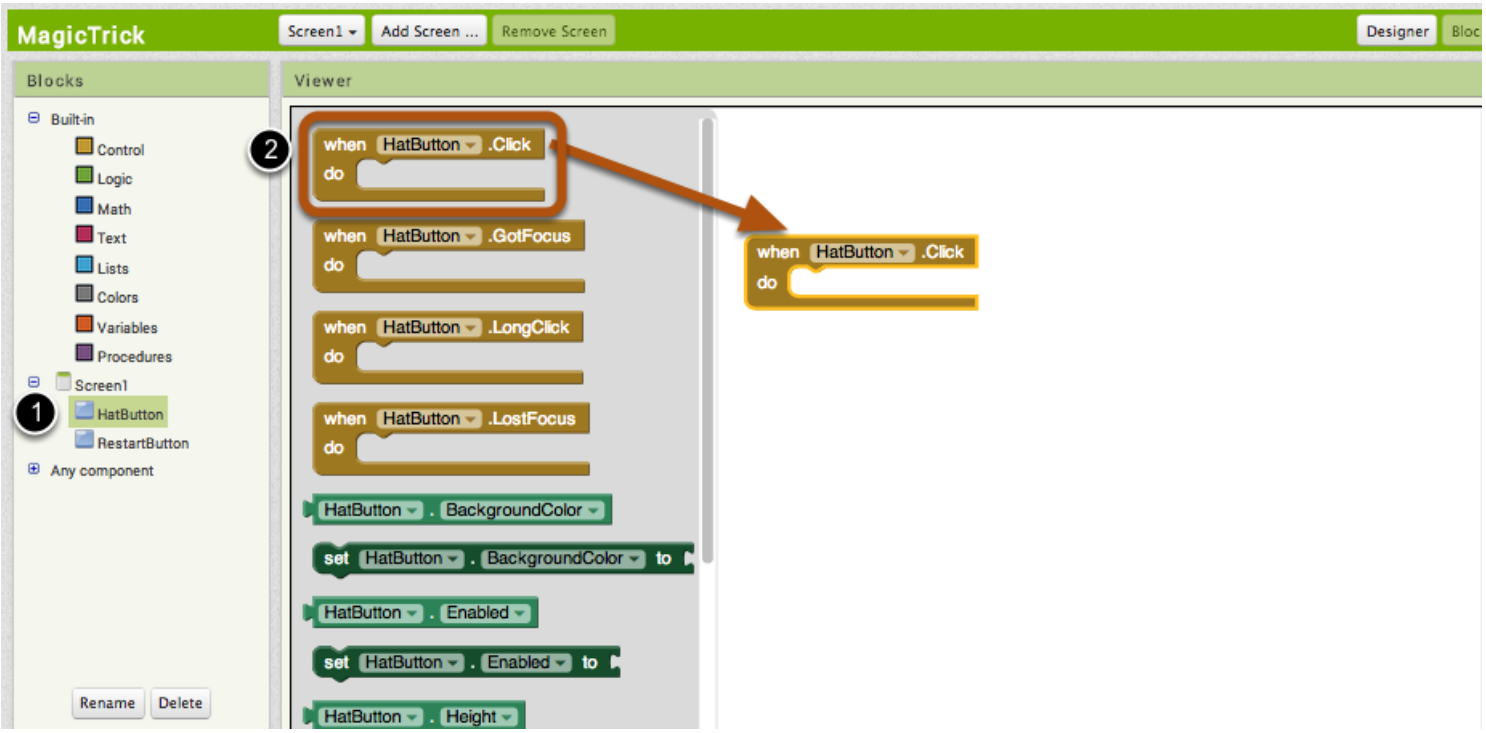
The Blocks Editor is where you program the behavior of your app. There are Built-in blocks that handle things like math, logic, and text. Below that are the blocks that go with each of the components in your app. *In order to get the blocks for a certain component to show up in the Blocks Editor, you first have to add that component to your app through the Designer.*



Make a button click event

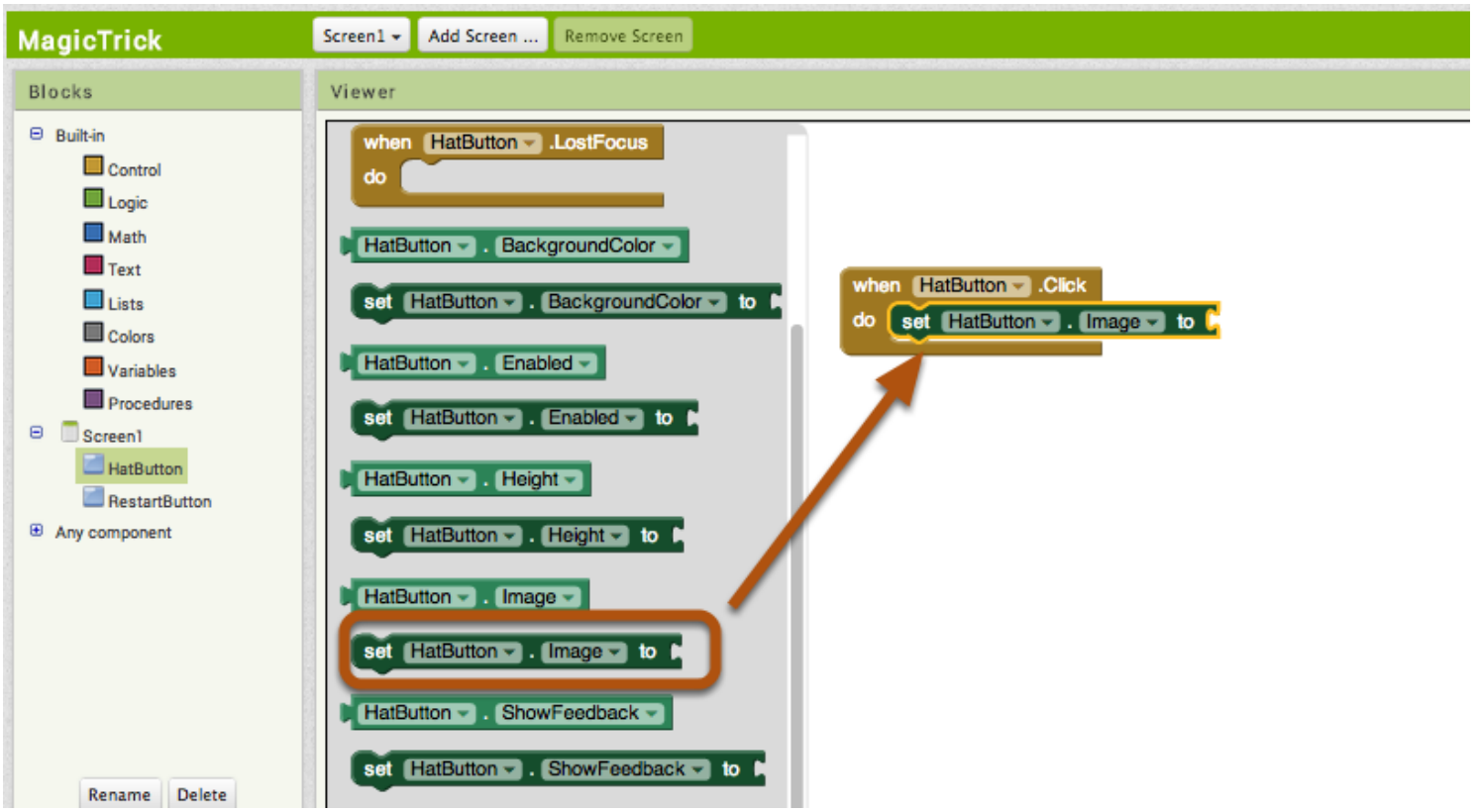
To make the image of the rabbit appear when we click on the hat, go to “Screen1” and choose “HatButton”. These are called Event Handler blocks. They are a light brown color and start with “When”, and in using these, we can decide what happens when a certain event takes place. The event we want is HatButton.Click. Here are the steps:

1. Click on the HatButton drawer.
2. Click and hold the when HatButton.Click do block. Drag it over to the workspace and drop it there. This is the block that will handle what happens when the button on your app is clicked.



Program the MagicTrick

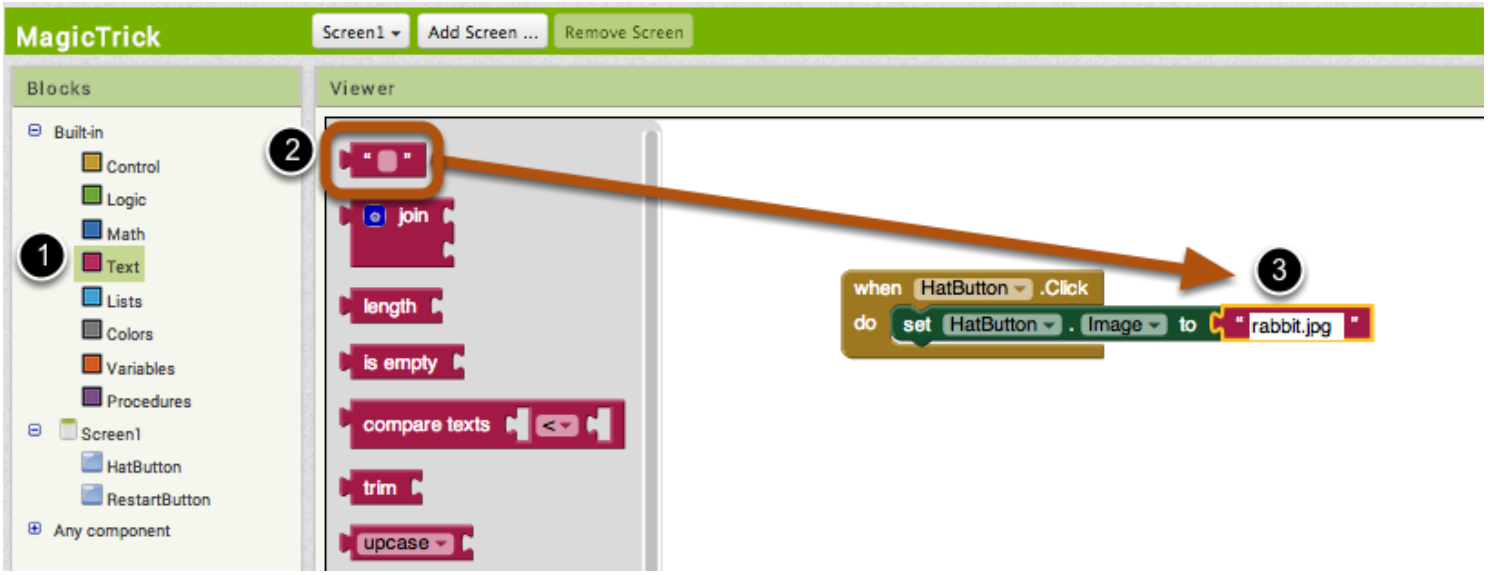
When we click the hat, we want the image to change from "hat.jpg" to "rabbit.jpg". So we want to change the Image property of HatButton. To do this, go back to “HatButton”, scroll down, and drag in “Set HatButton.image to”. Make sure it snaps inside the Event Handler block.



Now let's change the picture!

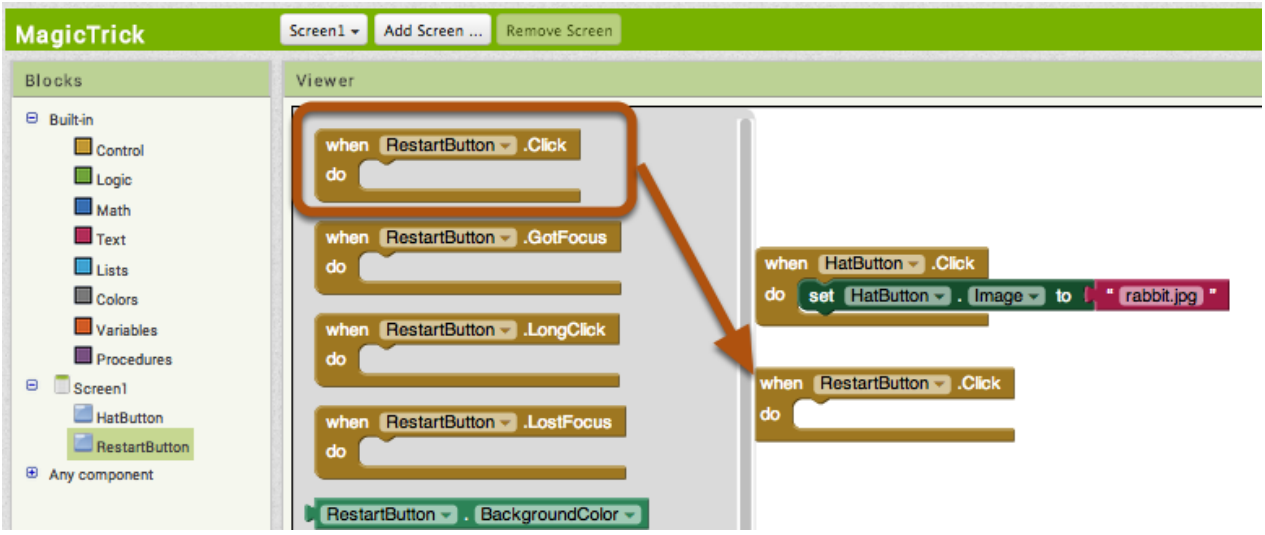
1. Go to "Text" under Built-In blocks.
2. Click on the small pink block, the top one shown and drag it to connect to "set HatButton.Image to". This block holds a string of text.
3. Once placed, click on the block and type in “rabbit.jpg”.

Whatever blocks you decide to place inside the event handler will happen when “HatButton” is clicked. So now when we click it, the original image will change to “rabbit.jpg”. Try it for yourself!



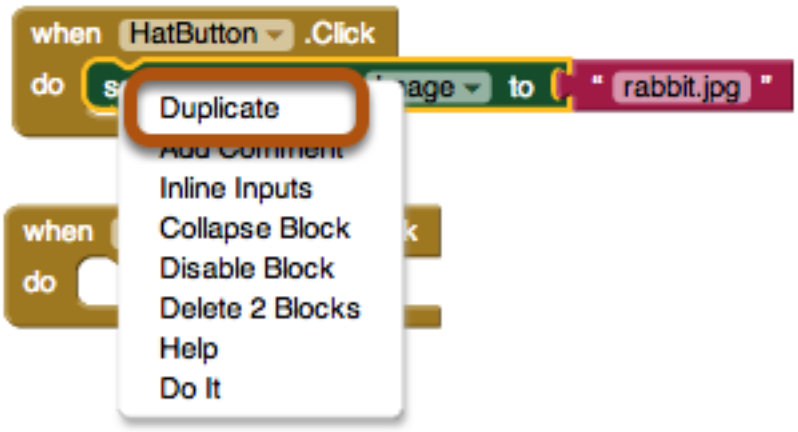
Program the Restart Button

You may have noticed that when you click the hat, it turns into the rabbit and there's no way to get the empty hat back. That's where the restart button comes in. We're going to program the restart button the same way that we programmed that HatButton *except* now when the Restart button is clicked, the image changes to "hat.jpg". Click on RestartButton and drag out "when RestartButton.Click" event handler.



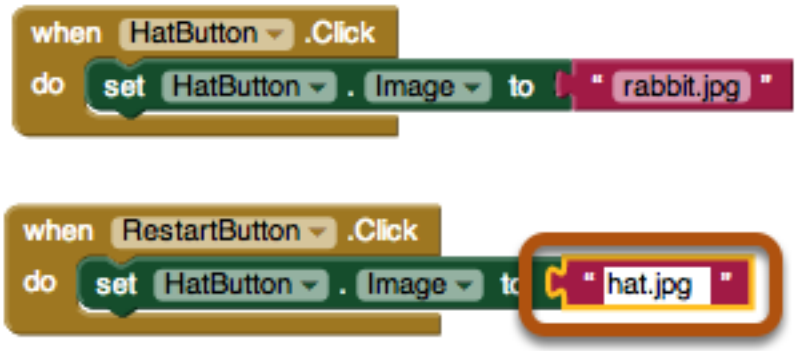
Program the RestartButton

Because we'll be using the same blocks as HatButton, we can "duplicate" them. Right click on the "set HatButton.Image" block and select "duplicate". Move this block into the RestartButton event handler.



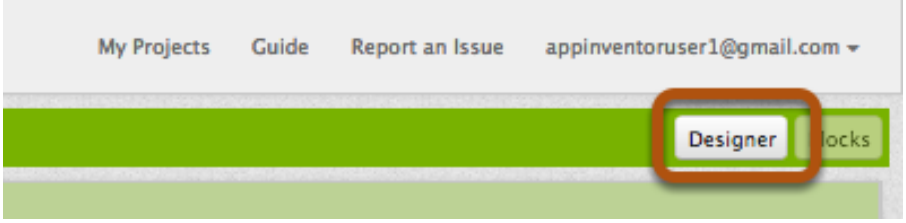
Changing the Image Back to Hat

Next click on the text in the pink block to change it to "hat.jpg". Test it out! Click on the hat. Do you see a rabbit appear? Then click on Restart. Does that hat become empty again?



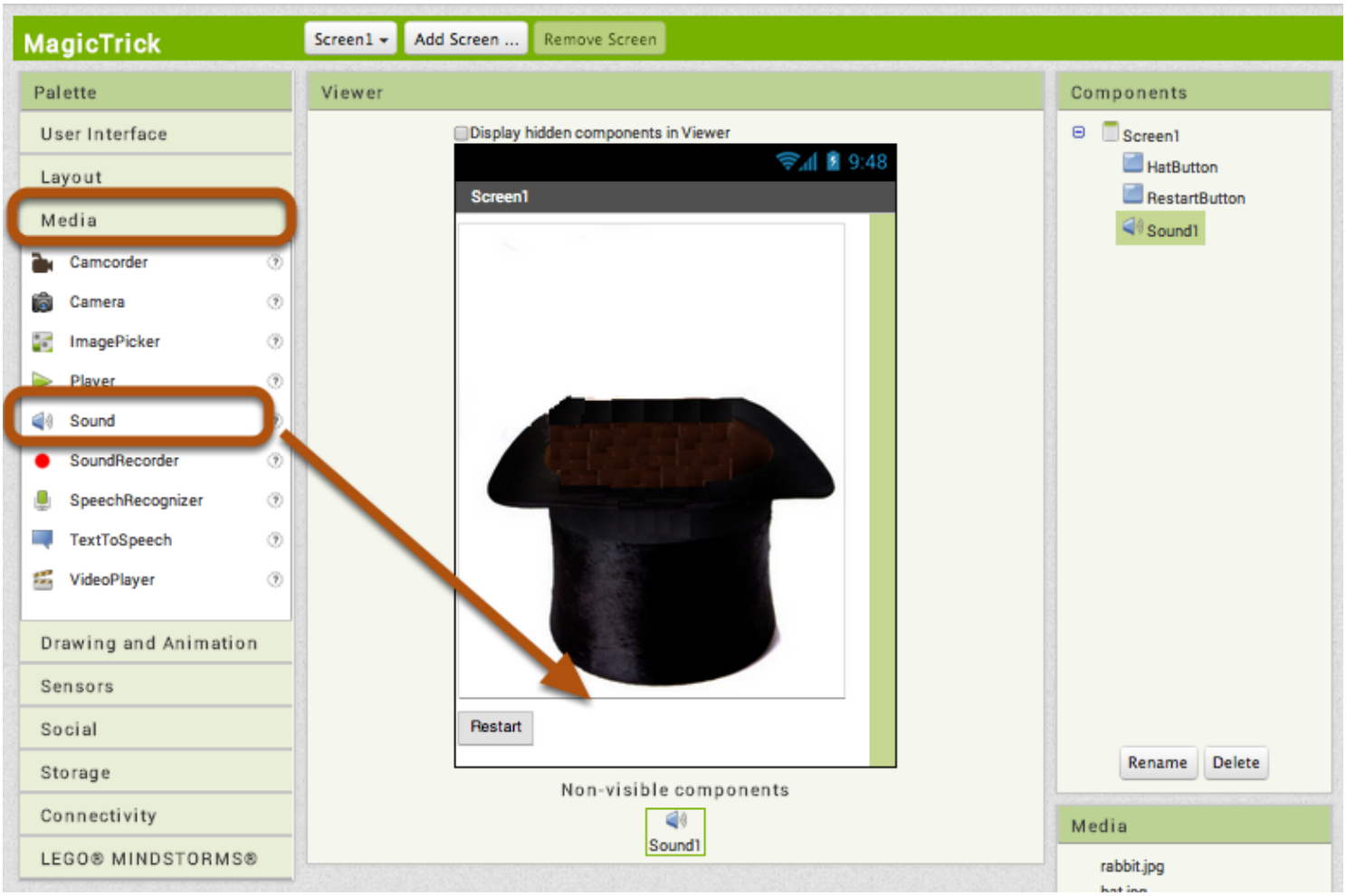
Adding Sound

Now, we're going to add sound to our app. To do this, we need to go back to the Designer window to add a Sound component. Click the "Designer" button to go back to the Designer.



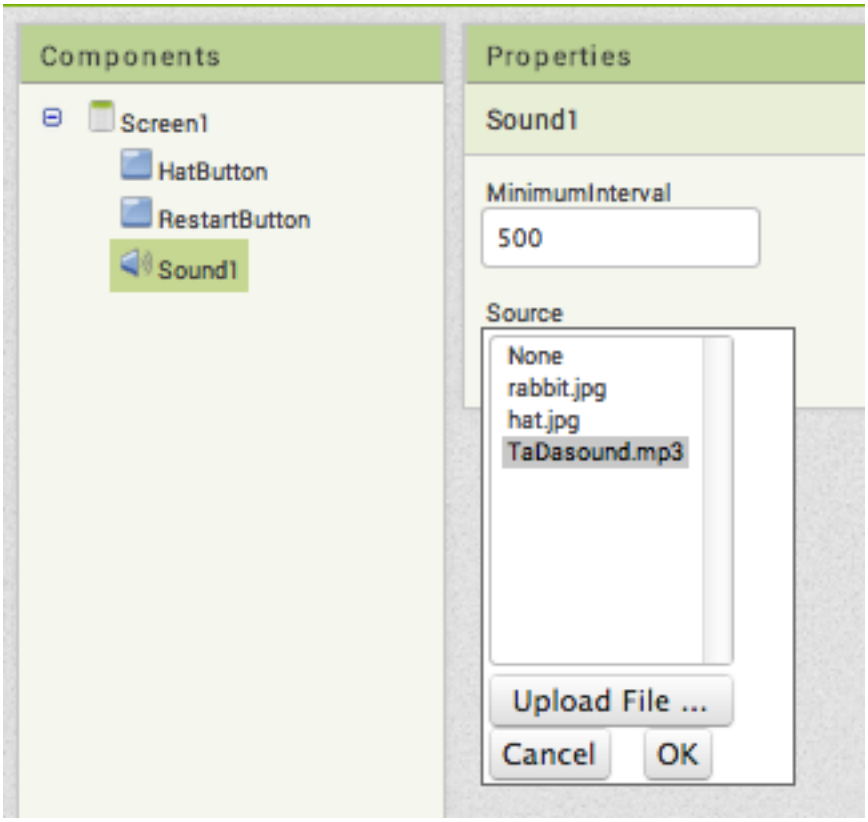
Add a Sound Component to Your App

Go to the Media drawer and drag out a Sound component. Drop it onto the Viewer. Notice that it drops down under "Non-visible components" because it is not something that will show up on the app's user interface. It's more like a tool that is available to the app.



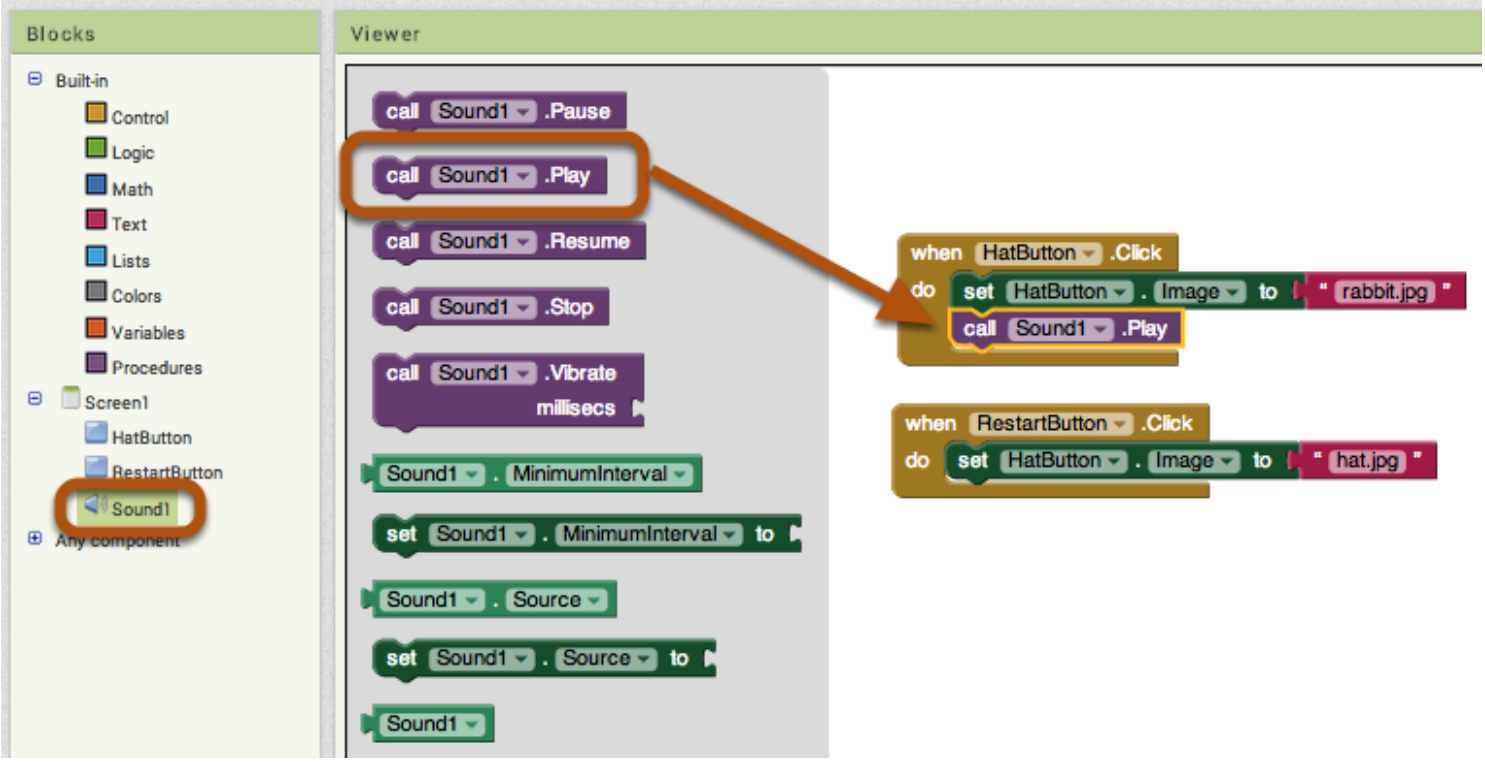
Change the Sound Properties

Click on Sound1 in the list of Components. Go to the "Properties" pane. The "Source" property is currently "None", click this and select "TaDasound.mp3". Click OK.



Programming the Sound

Go back to the Blocks Editor.
We want our app to make sound when we perform the trick. So what we want is a Sound to play when "HatButton.Click" is called.
Click on Sound1 under Screen1. Click and drag the block "call Sound1.Play" and place it into the "HatButton.Click" block.



Now test it out!

Go to your connected device and click the button. Make sure your volume is up! You should hear the phone speak the phrase out loud. (This works even with the emulator.)

Great job!

You've successfully built your very first app and become a magician! Ta-Da! If you'd like to keep building on this app, check out MagicTrick Part 2!