I Have a Dream

Most people have never even dreamed of building an app; software is this mysterious world of 0s, 1s and computer nerds. The goal of this site is to show people that they can create in this medium, that they can program software and contribute in today's highly digitized society.

This tutorial will get you building an app in minutes! You don't need to install anything on your computer-- App Inventor runs in the cloud! You will build the app on your computer and test it on your Android device. If you don't have a device, you can even test on an emulator

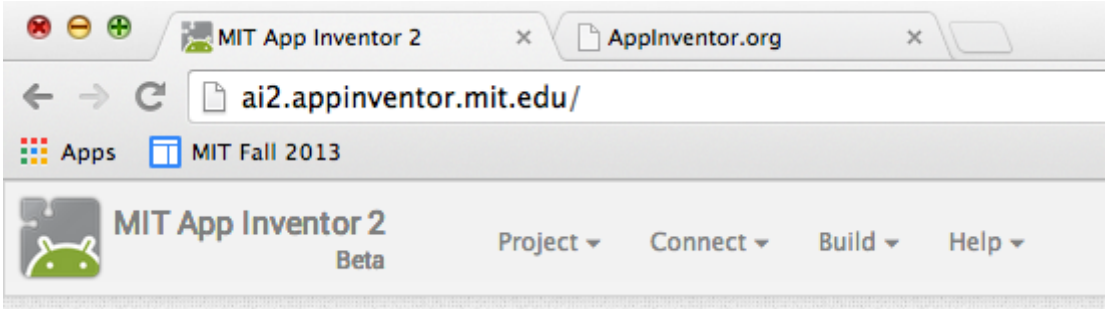
For your first app, you will build the "I Have a Dream" app. It has a picture of Martin Luther King that when touched plays the famous speech given to 250,000 on the steps of the Lincoln Memorial in 1963.After completing the first version, you will add an image of Malcolm X and one of his speeches, turning the app into a soundboard with these great leader's perspectives on the civil rights movement.

The tutorial should take you less than an hour. When you complete the two parts, you'll be ready to build soundboard apps on any topic. And you'll be a programmer!

Check out the app

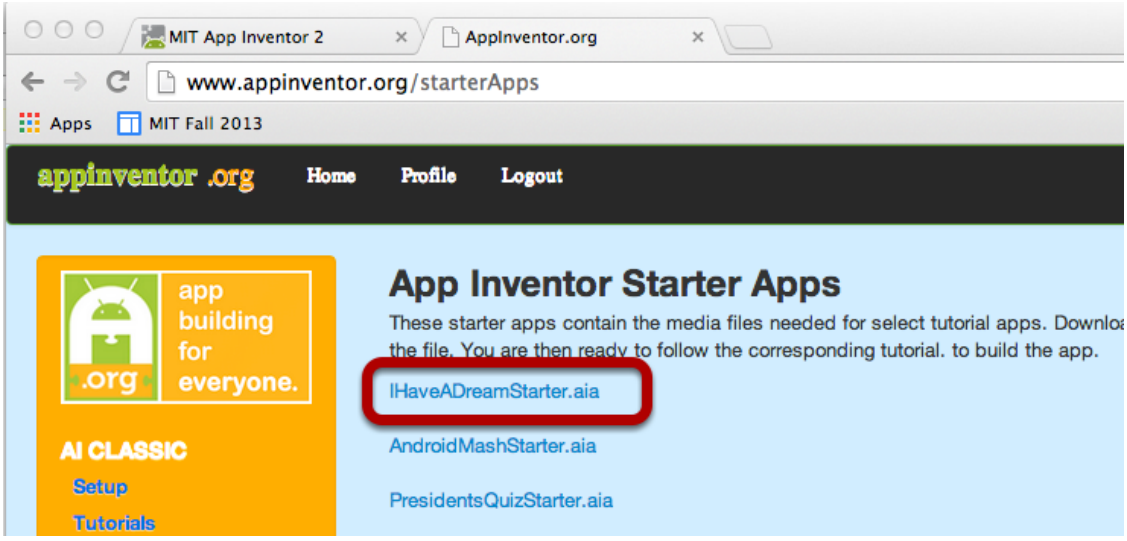
This is the app you'll buid.

Start at ai2.appinventor.org



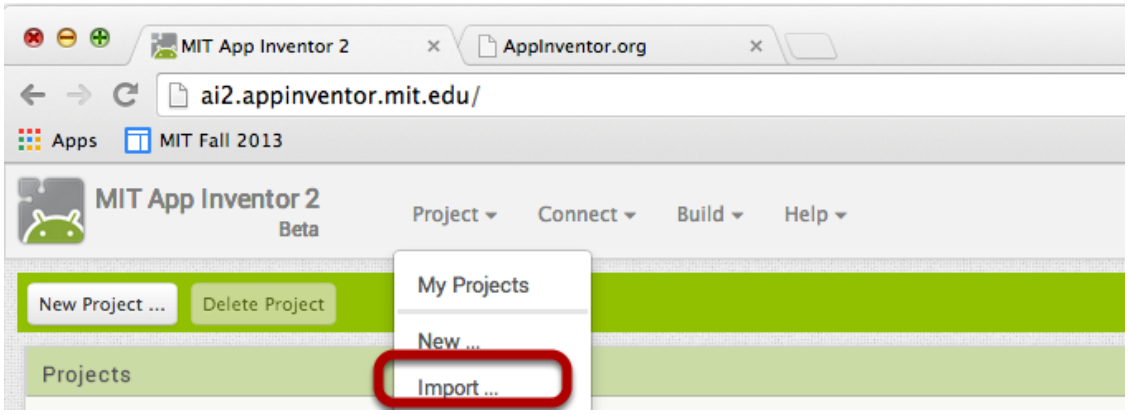
With the new version of App Inventor (AI2), you don't need to download anything to your computer. Just open a browser (Chrome, Firefox, or Safari) and go to ai2.appinventor.mit.edu and start creating apps.

Import the starter app

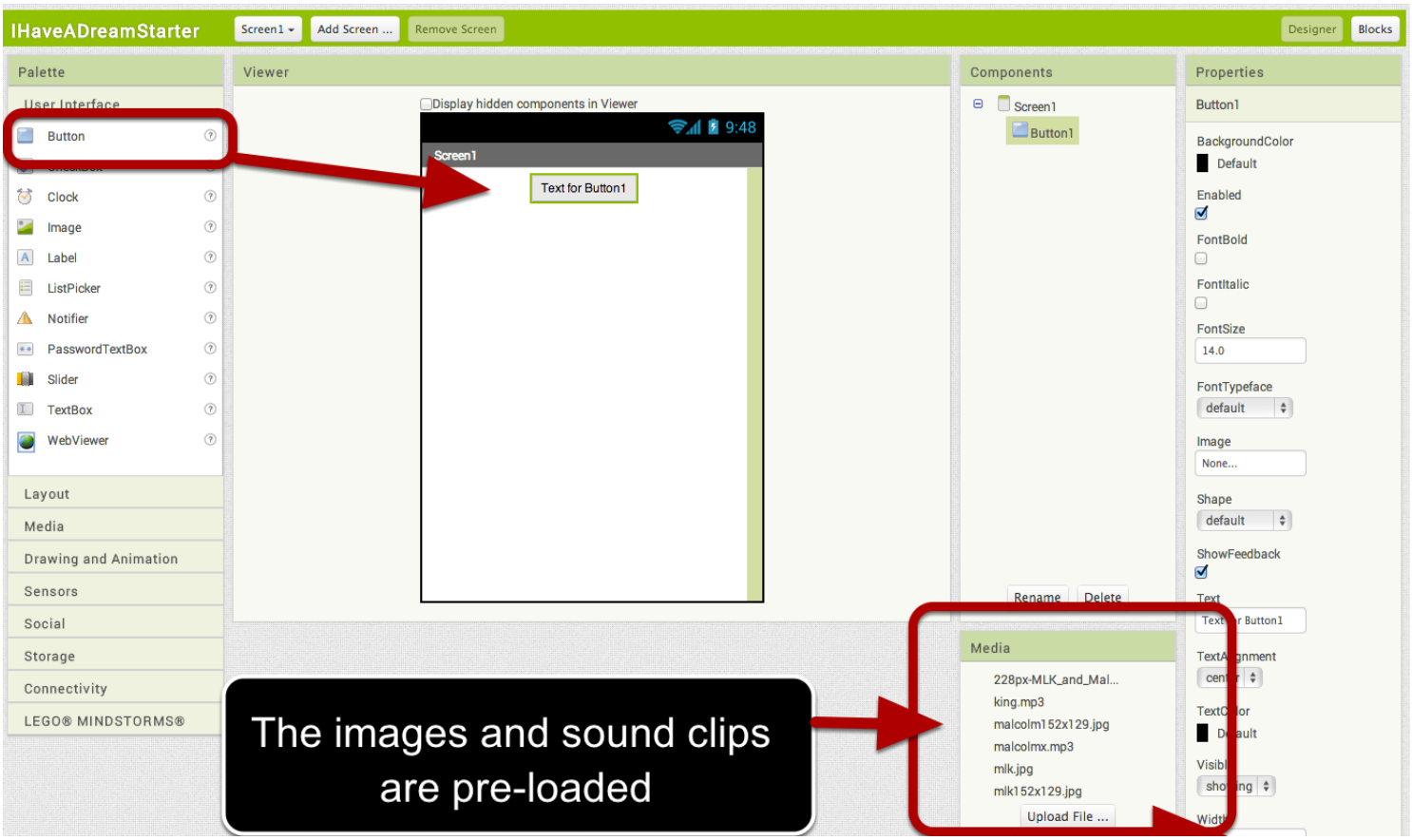


Open another window or tab and go to appinventor.org/starterApps. Download the IHaveADreamStarter.aia starter app to your computer. This app only has media files, no code. The media files are the pictures and speeches (.mp3 files) for this app. Note that you can upload any media you want into an App Inventor app, the starter app is here only for convenience.

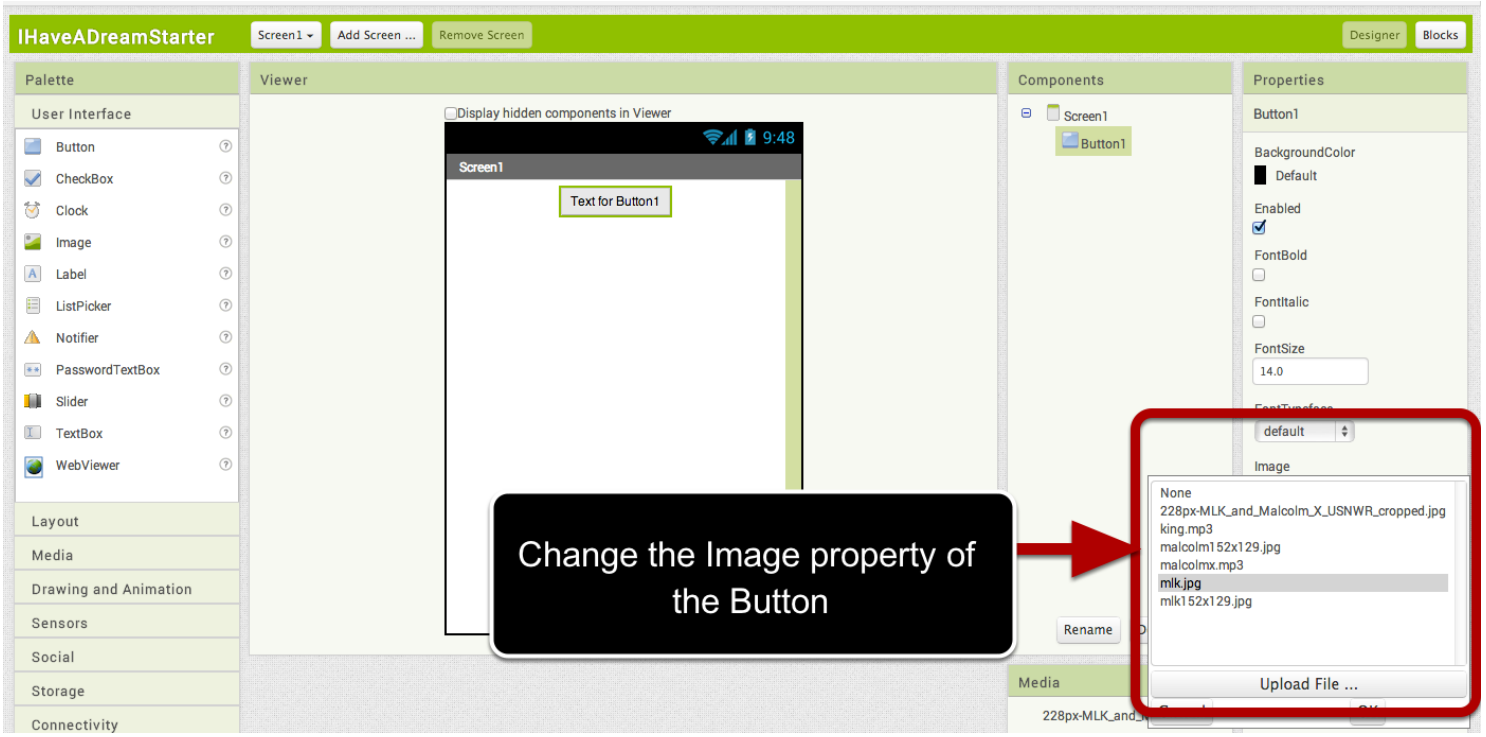
Import the starter app into App Inventor



In App Inventor, click on Project | Import and choose the file you just downloaded.

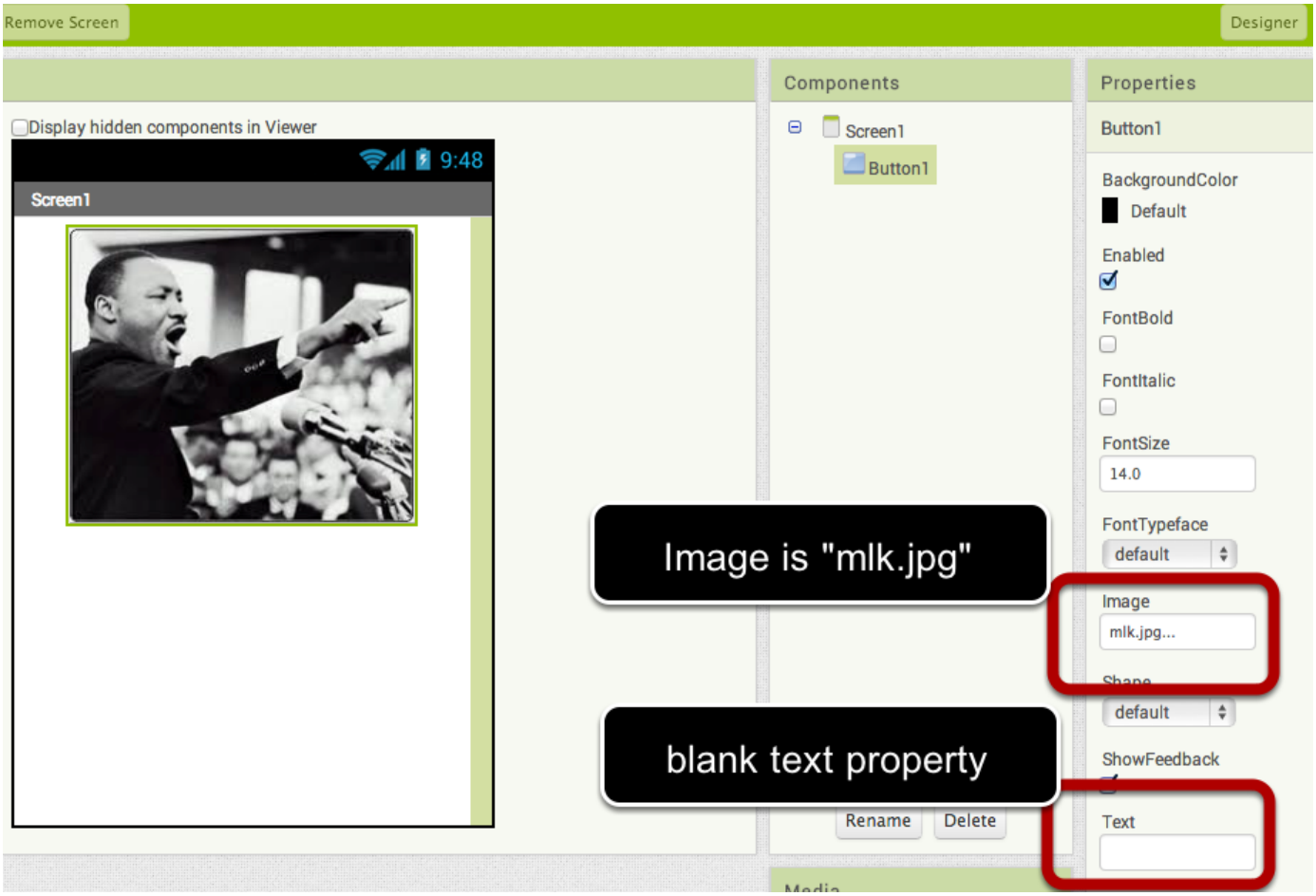


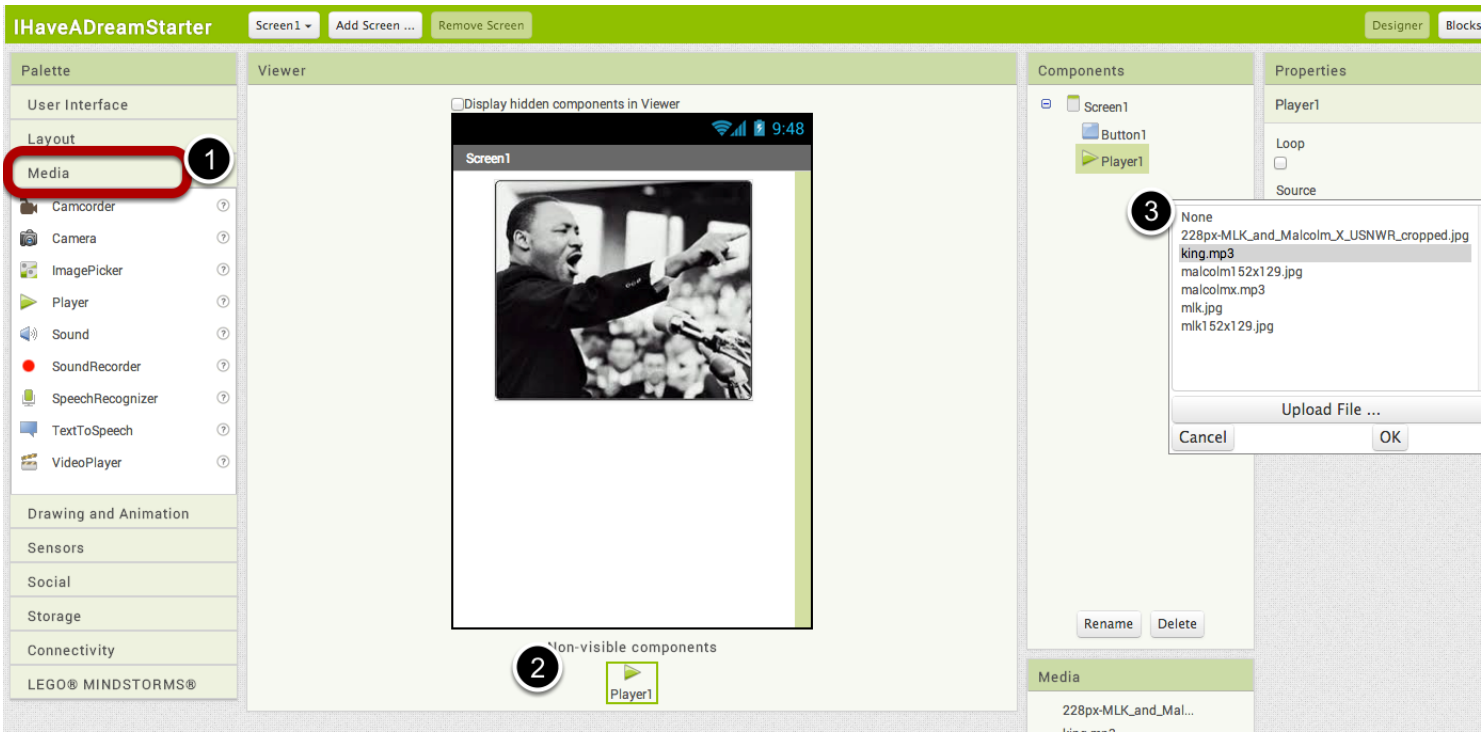
When you import the starter app (or create a new project)), the App Inventor designer appears. This is where you design the user interface. The starter app had the images for the app pre-loaded into the app, but they are not yet part of the user interface. So drag out a button component, then in the next step you'll place a picture on top of it.



Click on the Image property of button then choose the file mlk.jpg (the big picture of MLK). Also, blank out the Text property of the button so the text doesn't appear.

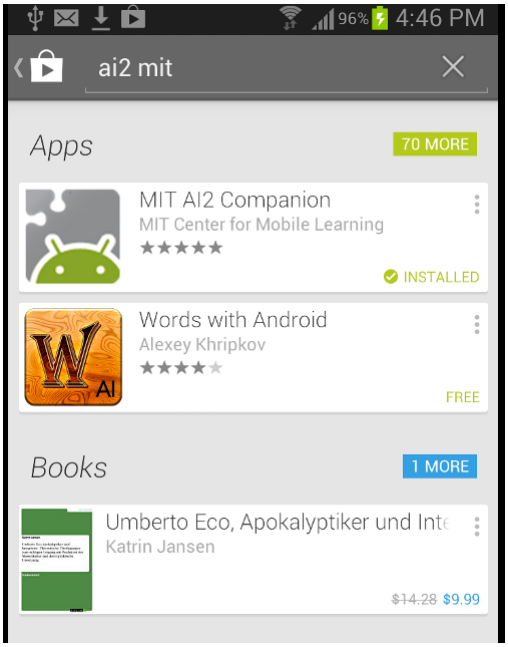
Check that the UI looks like the following:





You still need to set things up so you can play the sound clip of MLK's speech. Open the Media drawer (1) and drag out a Player component (2). Then set the "Source" property of the Player to king.mp3 (a short clip from "I Have a Dream!". The user interface is ready to go!

Get ready to test: download the AI2 Companion from the Play Store

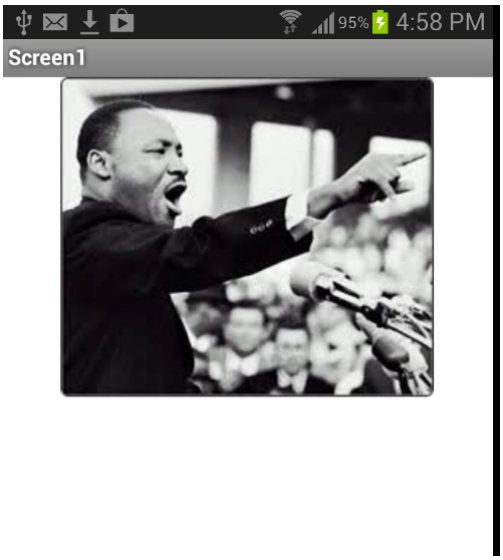


On your Android phone or tablet, go to the Play Store and download the MIT AI2 Companion. This app will allow you to test your App Inventor apps as you build them. Search for "ai2 mit companion" and you should find it. The download and install (on your device, not computer!).

Run the companion and connect your app to it

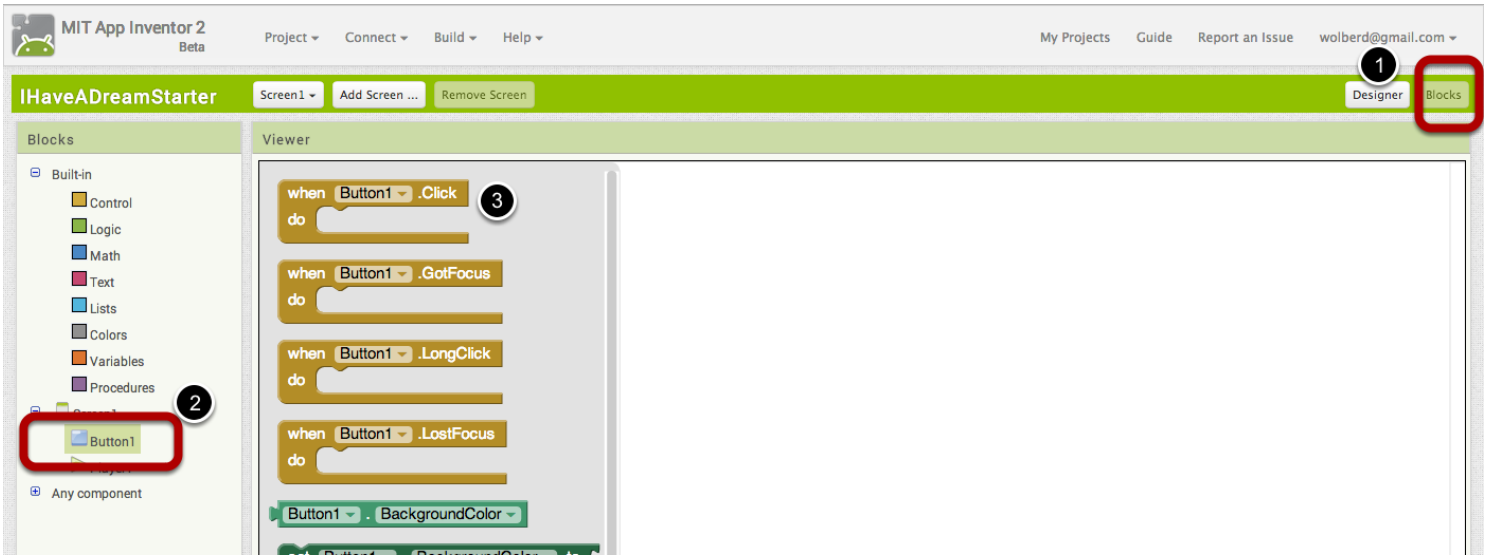
When you start the Companion on your phone, it will look like (1). Back in App Inventor, choose "Connect" then "AI Companion. This will cause a QR code to appear (3) You can then scan the QR code (4) with your phone to see your app live. NOTE: for live testing to work, **both your computer and phone/tablet must be connected to the same WiFi station.** Using WiFi is the easiest way to connect, but if you're at school/work you may have firewall issues. If so, you can connect using a USB cable. If you don't have an Android device, you can connect your app to an emulator that runs on your computer. For any setup issues, please see
http://appinventor.mit.edu/explore/ai2/setup.html

Check that your app appears on your device



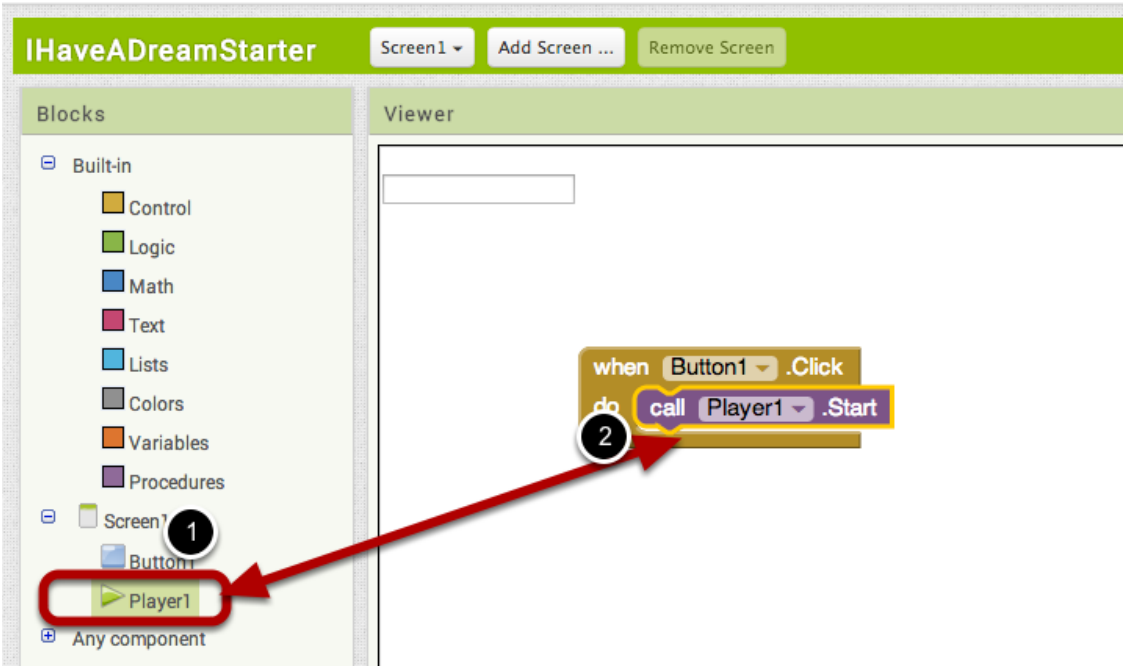
When you scan the QR code, your app should appear on your phone/tablet. Note that even if you click MLK, the speech won't play. Your app has the sound clip, but it doesn't know that clicking the button should play it. It is time to code the behavior.

Open the Blocks editor to code the behavior.



Click on "Blocks" in the right top to move from the UI Designer to the blocks editor (1). Then click on the Button drawer (2) and drag out the "Button1.Click" event (3).

Drag out a Player.Start block

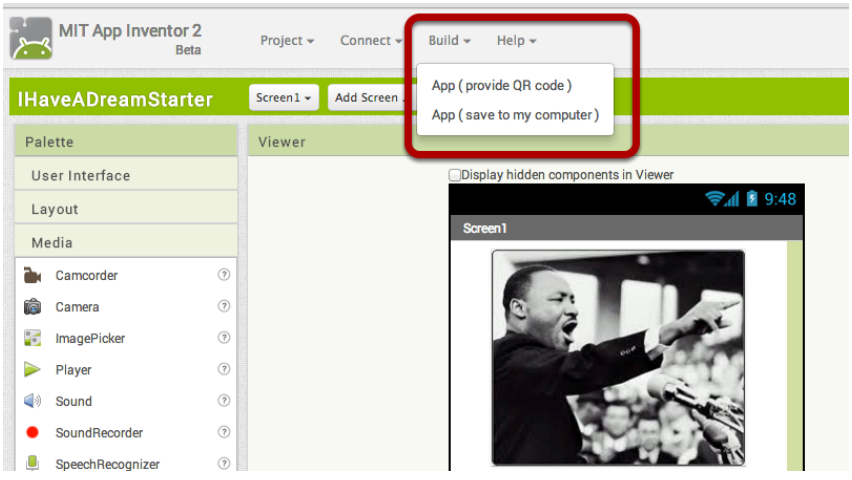


Open the Player1 drawer (1) and drag out a "Player1.Start" block. Place it within the Button1.Click (2). You have just created an "event handler". You have specified that on the "Button1.Click" event, the app should start the speech.

Test your app!

Is your phone still connected? If so, tap on the picture of MLK. Does the speech play?

Download the app to your computer

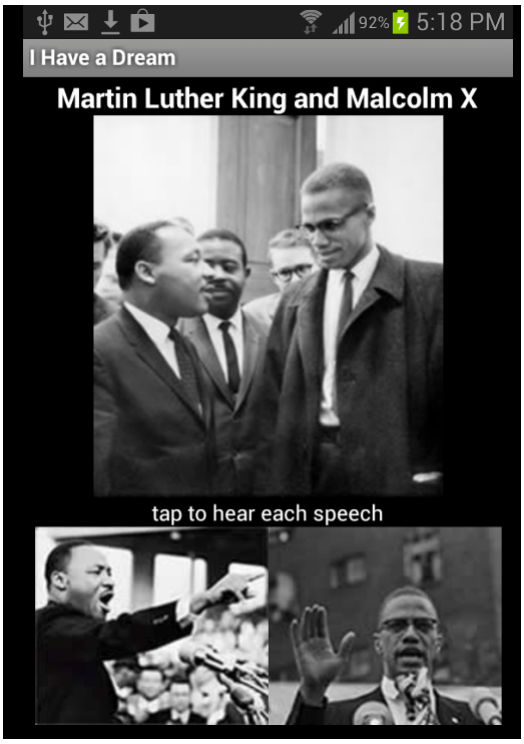


Your app is only working within the testing app, the Companion. If you disconnected from WiFi, you wouldn't have the app, and so far you can't send it to a friend. So go back to the Designer and choose "Build | App (save to my computer)" This will download a ".apk" file, an executable Android

app. Once you download it you can email it to yourself or a friend. Then you (or your friends) can open the email on their Android phone and install the attachment. Note: on some Android phones, you need to change a security setting to allow "Unknown Sources". This allows the device to install apps from places other than the Play Store.

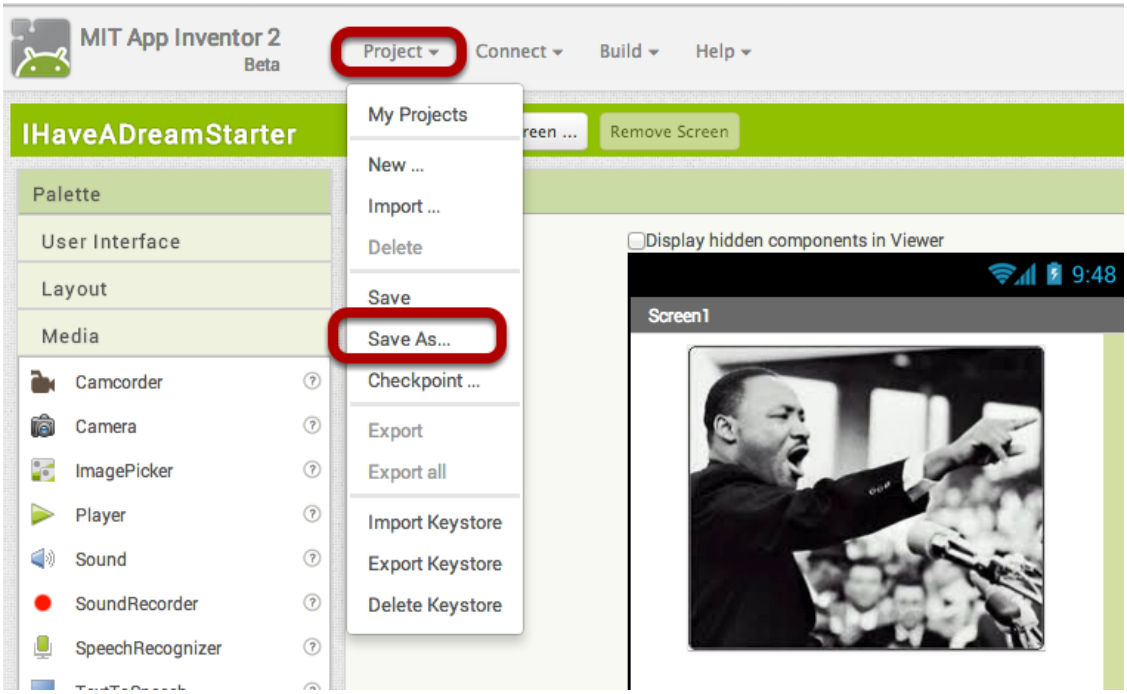
Congratulations! You've built your first app!

PART 2: Add Malcolm X and his speech



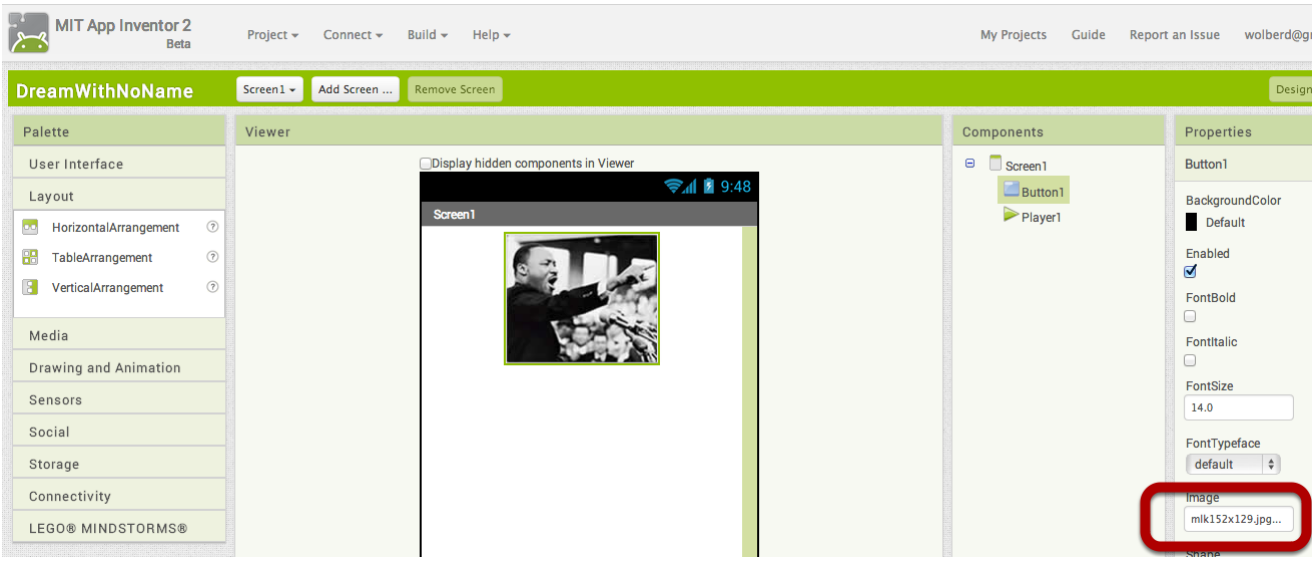
Now that you have your feet wet, let's make things more complicated. MLK and Malcolm X were Civil Rights leaders of great contrast. Let's build an app that shows this contrast. You'll need to modify the UI, then the behavior. And your code will be more complex as will make sure the speeches don't overlap and allow you to pause the app.

SaveAs your app to a different name



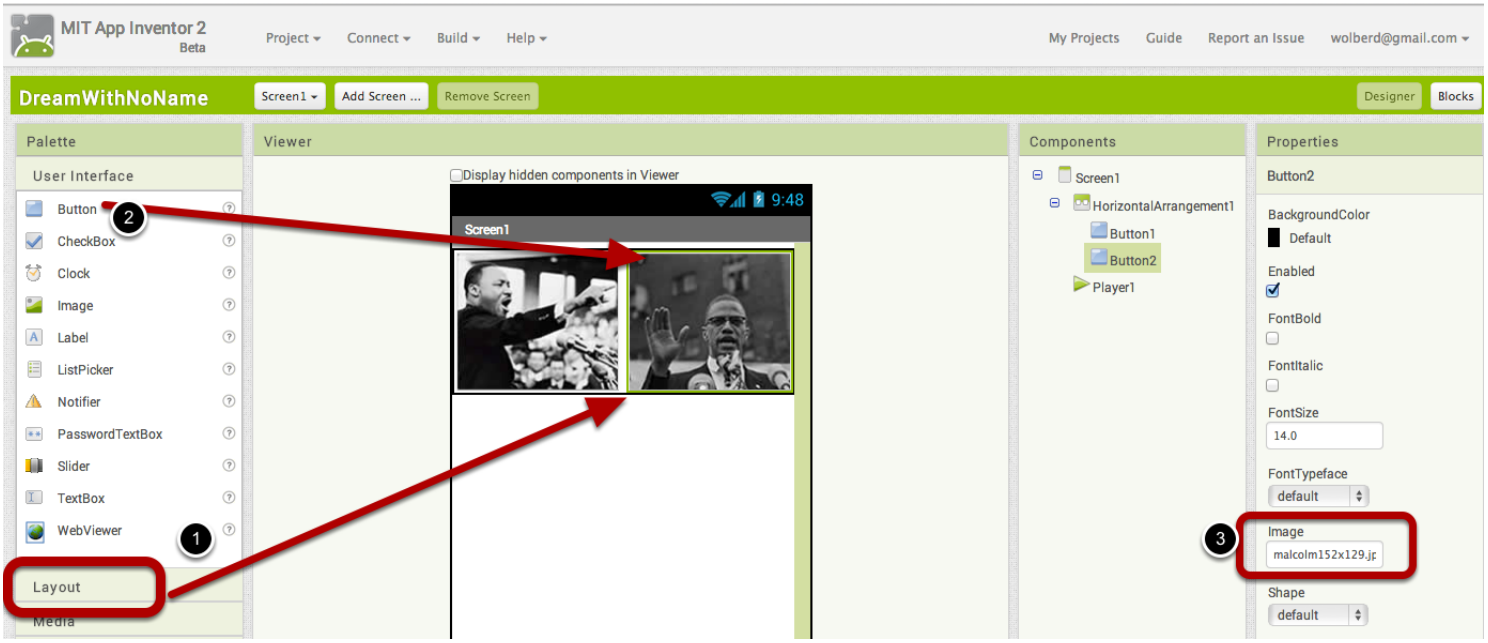
choose Project | SaveAs and name your new app something else (e.g., DreamWithNoName).

Choose a smaller image for MLK



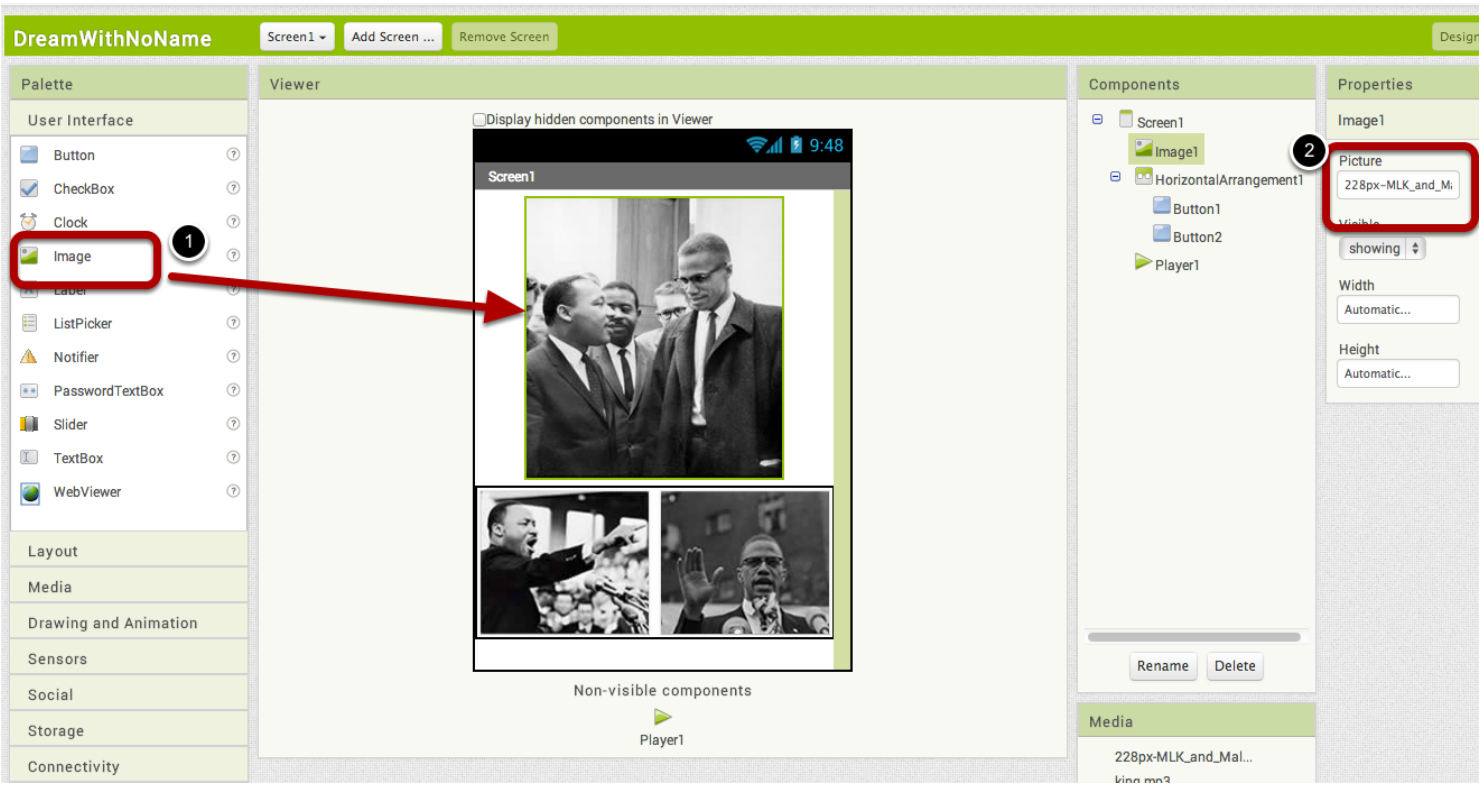
Click on the Button1 component and change its Image property to "MLK152x129.jpg". This will make room for Malcolm X.

Add the picture of Malcolm X



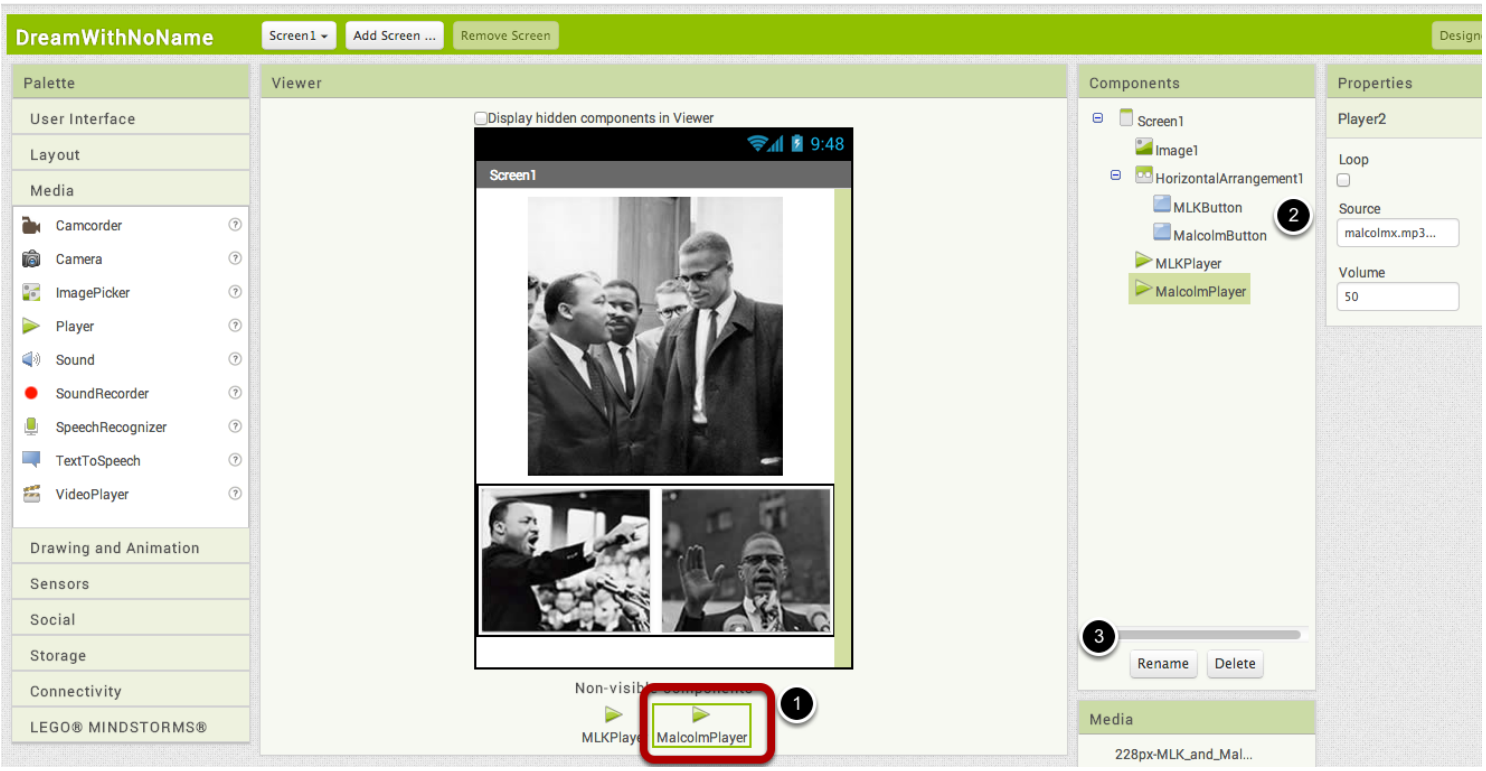
Drag in a HorizontalArrangement from the Layout drawer, place the MLK button in it, then add a new Button in it. Change the new Button image property to the picture of Malcolm X.

Add an Image of the two leaders



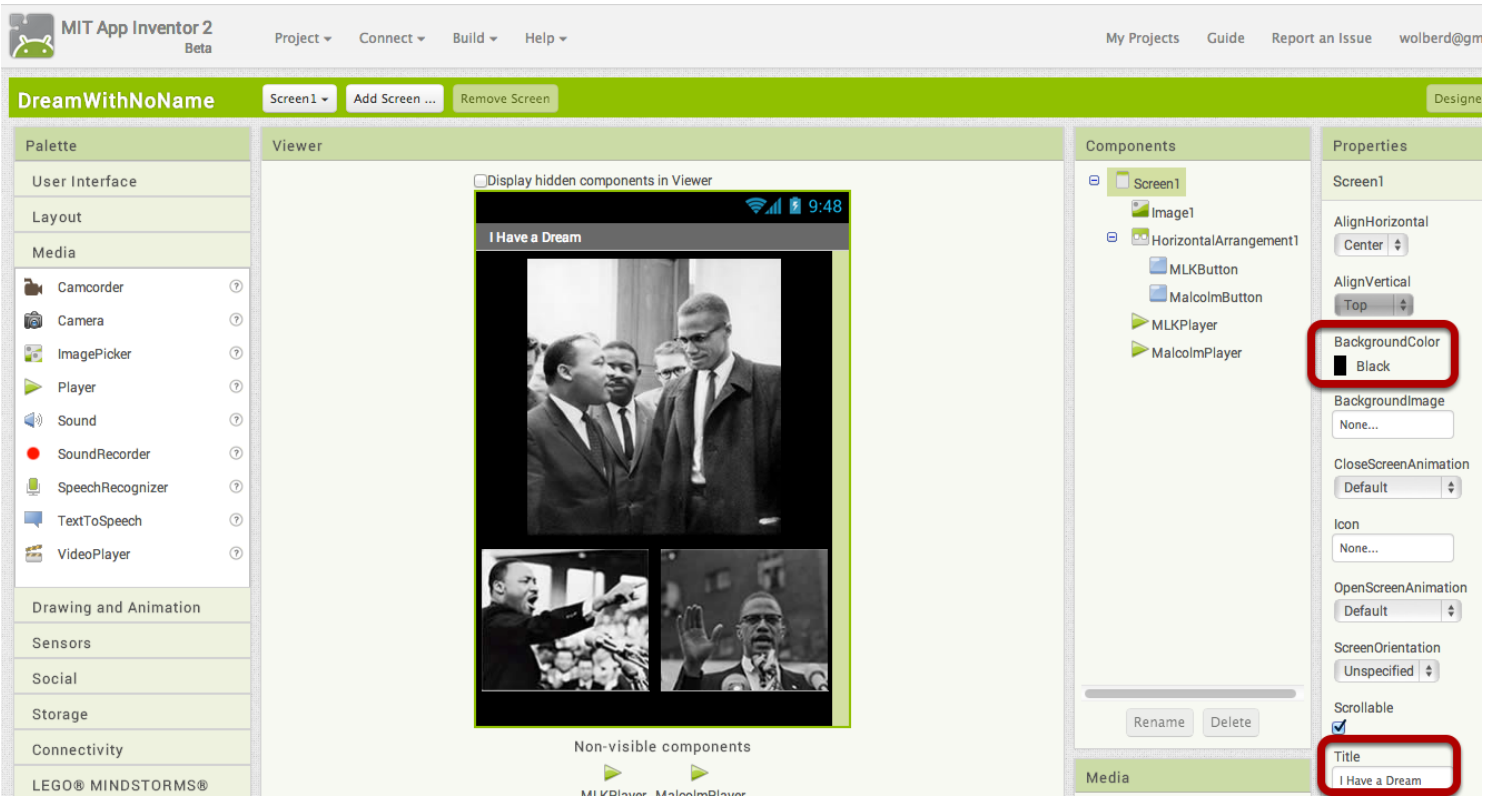
Add an Image component (1) then set its Picture property to the picture of both leaders.

Add a second Player component then rename the components.



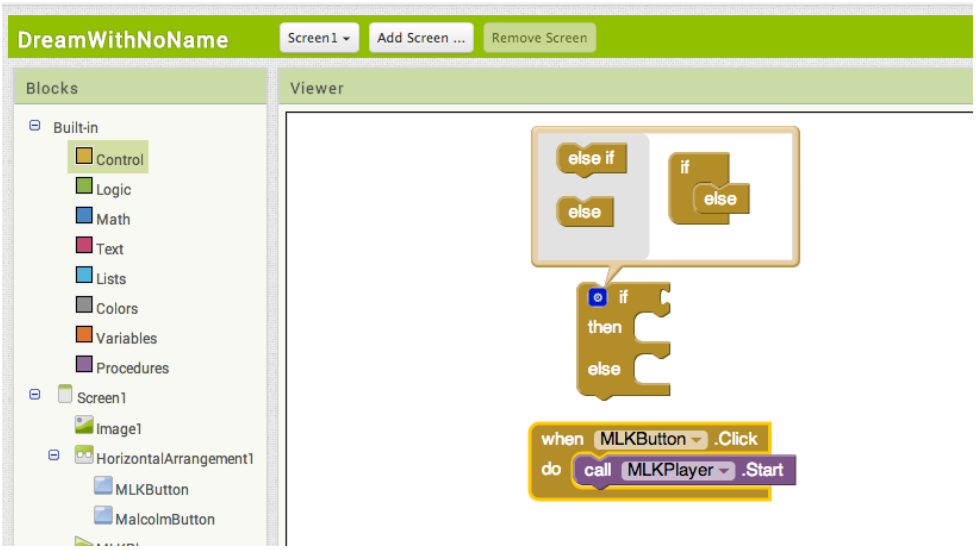
Drag out another Player component (1) and set its Source property to Malcolm's spech (2). Then rename the components (3) so we can distiguish them easily in the blocks editor. A rule-of-thumb is to give a descriptive name with a suffix which is the component type, e.g., MalcolmPlayer.

Set the Screen title and background



To complete the UI, set the BackgroundColor and Title properties. Now you're ready to program the behavior.

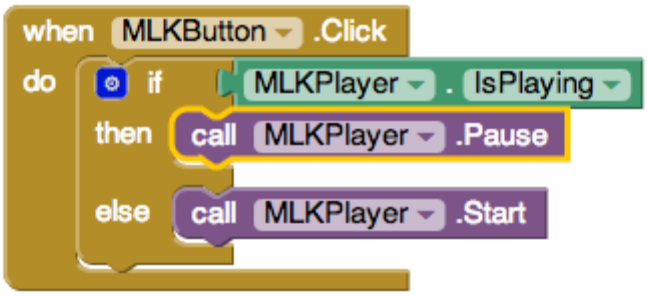
Make it so you can pause/play MLK's speech: Conditional If-else block



Open the Blocks editor. Note that the Button1.Click event and other blocks have been renamed automatically (based on your renames in the designer).

The behavior we want is for the first click to start the speech, next one pause it, next one start it, and so on. So you don't always want to do the same thing when the button is clicked. To program this, you use an if-else block. If-else allows the app to ask questions, such as, "is the speech already playing?". To code this, drag out an if-block from the Control drawer, then click on the blue "modifier" icon. The modifier allows you to add branches-- in this case, we just need an if and an else.

Code it so the app pauses/plays based on the if-else

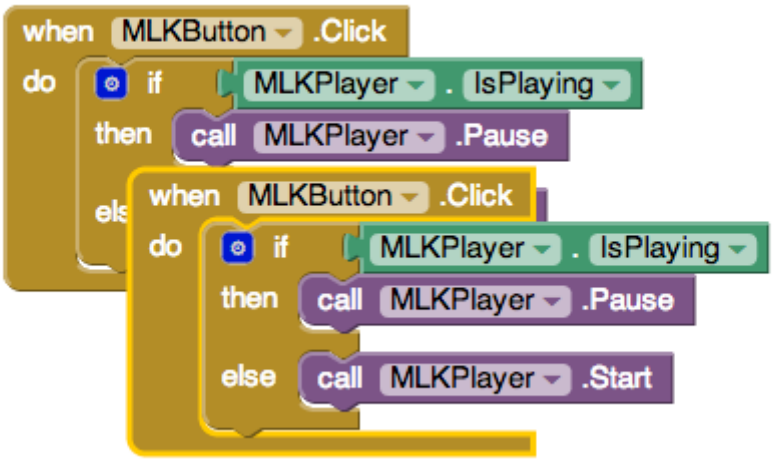


Place the if-else block in the event handler. Then drag out an MLKPlayer.IsPlaying property block from the MLKPlayer drawer. This block is true if the speech is playing, false if not. If it is playing, you want to pause the speech, so drag out this block from MLKPlayer. If it is not playing, the "else" branch will be taken, so MLKPLayer.Start is called.

Test the behavior

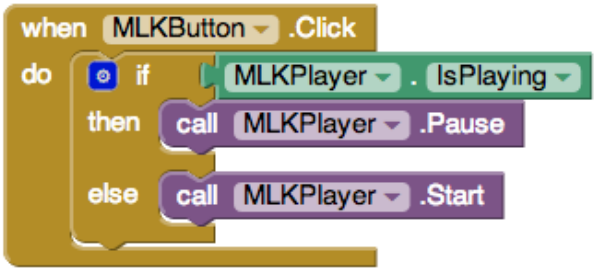
Is your phone/tablet still connected? Test this updated behavior to see if it works.

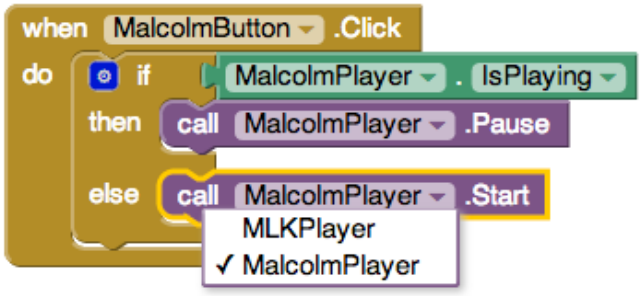
Copy and paste the blocks for playing the MLK speech



You want the Malcolm button to behave similarly to the MLK button, so copy-paste. Open the Blocks editor, then select the MLKButton.Click event handler Copy and paste, using command-c and command-v on Mac (ctrl-c and v on Windows).

Change the copy to work for Malcolm X.



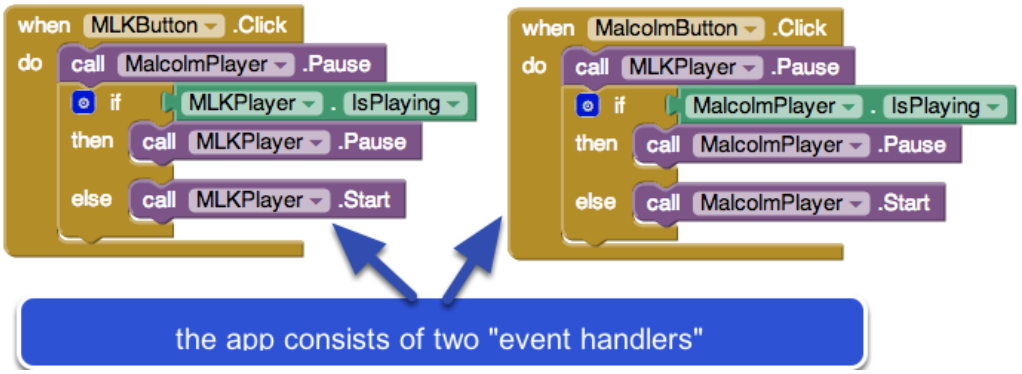


Using the upside down triangle widget, switch the copied blocks to refer to MalcolmPlayer instead of MLK.

Test the behavior

Test this behavior. You should be able to start and pause the speeches independently. But what happens if you click MLK then Malcolm? Do the speeches overlap?

Add blocks to pause each speech when the other is played



You just need to pause Malcolm when MLK speaks, and vice-a-versa. The final blocks for the app are shown above.

Note that the app, or at least its behavior, consists of two event handlers. Each event handler consists of an event (a click in this case) and a response which is a sequence of blocks (they're executed in order). Some blocks in the response are only conditionally executed.

This app is simple, but indicative of the architecture of most apps. You've not only created an app, you've learned some programming lingo. So next time your hanging out with your friends, smack talk a bit about "event handlers" and "if-else conditionals".