|  |  |
| --- | --- |
| This is the "I Have a Dream" app, the first app you’ll build for the Mobile CSP course. You'll learn that app building is a creative process. You will write code that plays Martin Luther King’s “I Have a Dream” speech when the user touches the phone’s screen.  **Objectives:**  In this lesson you will learn to :   * follow an instructor-led walkthrough to create the I Have a Dream app on a mobile device; * navigate the App Inventor programming platform; * develop your understanding of what an App Inventor program is; * develop your understanding of event handlers; * develop your understanding of how an app makes decisions with if-else control blocks. | ***[Click to watch Preview Video](https://www.youtube.com/watch?v=0vLz8TIwido)*** |

# 

# Getting Ready

1. You’ll start by downloading a *starter app that has the image and sound files you need but is otherwise a blank app.* Download this app at <http://www.appinventor.org/starterApps/IHaveADreamStarter.aia>
2. Open [App Inventor 2](http://ai2.appinventor.mit.edu).
3. Go to *Project*
4. Select *Import Project (.aia) from my computer.* Then choose the file you just downloaded to your computer.
5. The starter app will open in App Inventor.

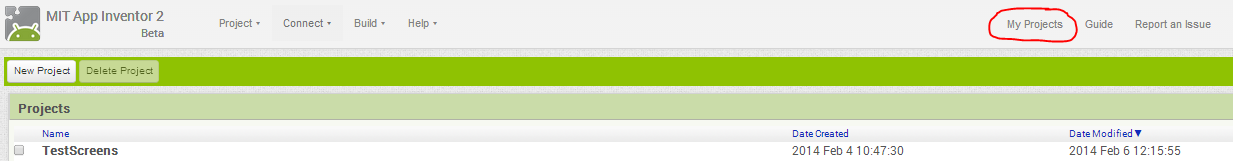
# I Have a Dream Tutorial

## Navigating Among the Three Main Views

There are three main views that you will use in building apps.

### My Projects View

The *My Projects View* provides you with a list of your projects. To get to this view, click on the *My Projects button.*  Use this view to create, upload, save, delete and otherwise manage your projects. To open the *IHaveADreamStarter* project, just click on its name.

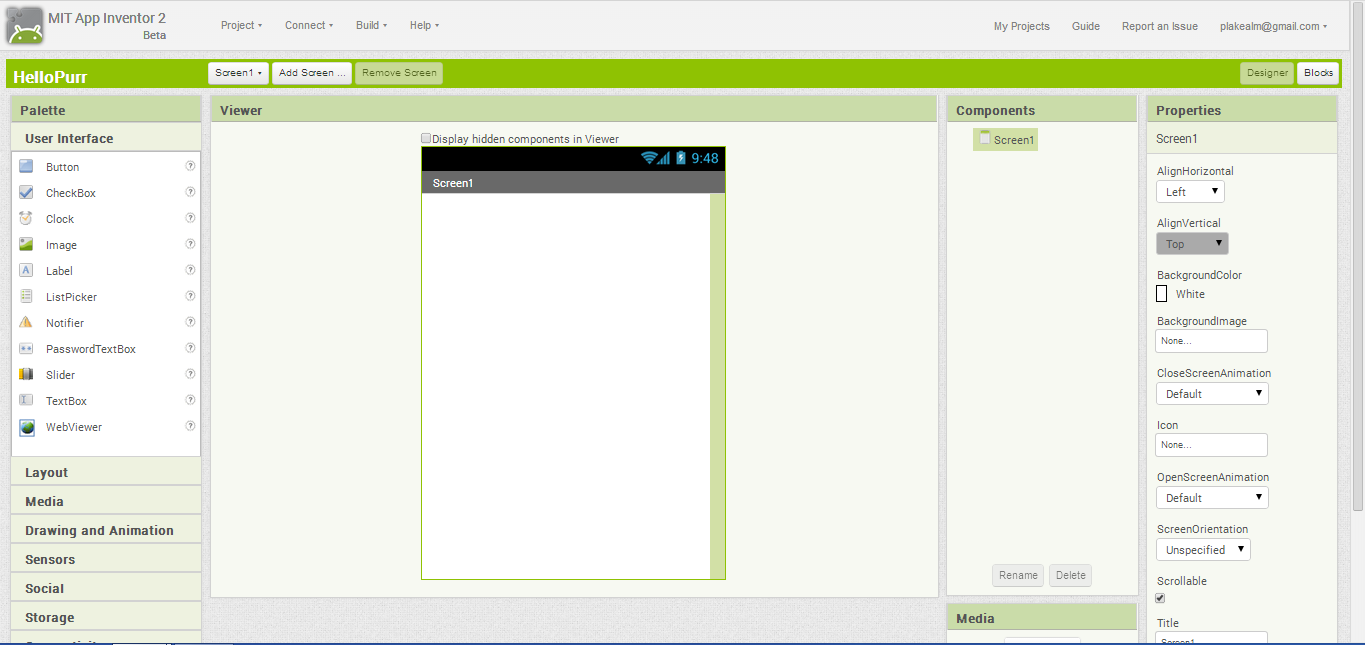


### 

### Designer View

Your project will open in the *Designer View,* which is used for designing the app’s *User Interface (UI).*

**Note:** The Designer button is greyed out, because you are already in the Designer View.



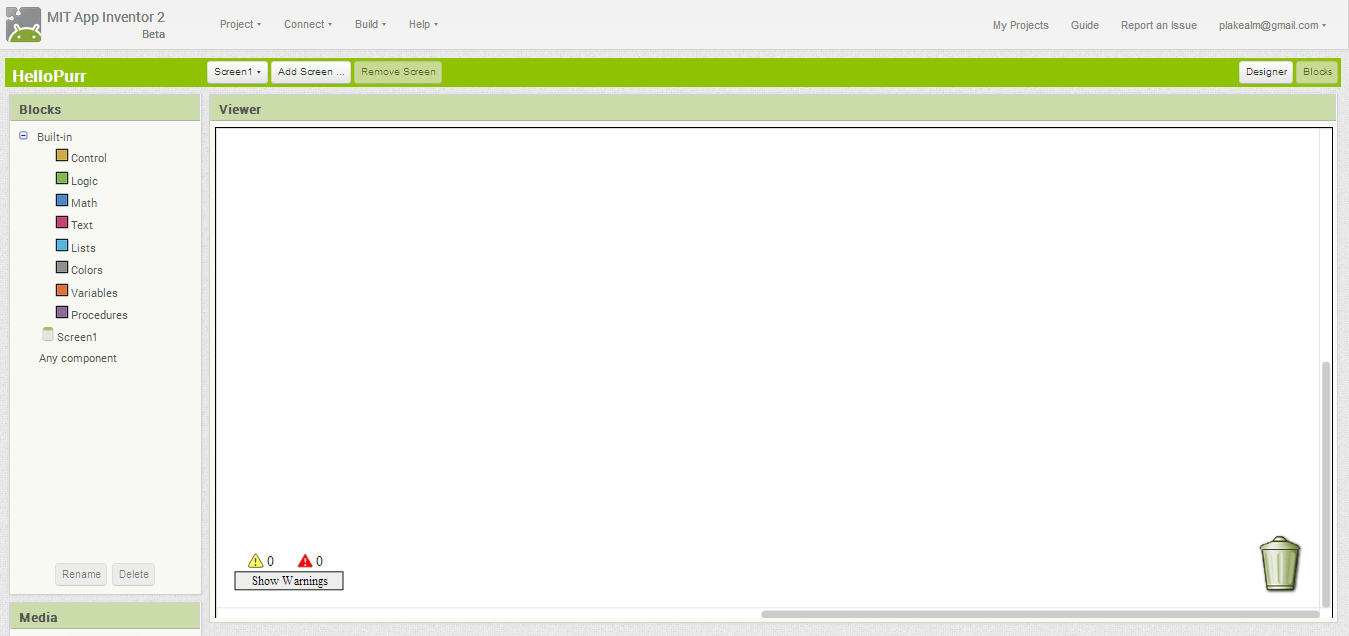
As you can see, the Designer View contains five panels: the *Palette,* which contains a list of categories of various elements; the *Components* panel, which will include the list of elements contained in your app; the *Media* panel, which will include the list of project resources that have been uploaded; the *Viewer,* which contains a mock-up of your app’s user interface; and the *Properties* panel, which contains a list of the properties for the selected component. Make sure you can identify each of these panels.

For the app shown here, the *Screen1* component is selected in the Components panel and its properties are shown in the Properties panel. As the app designer, you can change the default values of the screen’s title and other properties.

1. Change the screen’s Title to “Speech Player”
2. Change the screen’s ScreenOrientation to “Portrait.”
3. Change the screen’s BackgroundColor to a black

### Blocks Editor View

The *Blocks Editor View* is used to create the code blocks for your apps.To navigate to the Blocks Editor View, click on the Blocks button.



The Blocks View consist of two main panels, the *Blocks* panel on the left, which is also known as the *Toolbox,* and the *Viewer* panel on the right, which is known as the *Workspace*. The Blocks panel contains a list of *Built-in* blocks, which include blocks for doing Math, setting Colors, and other tasks. For example, if you click on “Math” you will see a drawer full with all of the various math operations that you can do in App Inventor. The Toolbox also contains a list of *Component* blocks. So far there is only the *Screen1* component. If you click it you will see the various operations you can perform on that component.

## The I Have a Dream User Interface

**User Interface**

## 

What is the app’s *user interface* (UI)? It is that part of the app that interacts with the user. It includes any elements that the user can see, hear or feel. Designing an attractive and easy-to-use UI is an important part of building good mobile apps.

The user interface (UI) for our I Have a Dream app will consist of four *Components,* a Button, a Sound, and two Labels. Remember that this is done in the Designer View.

Your Viewer panel should now look like this when your app is complete.

### 

### 

### 

### 

### 

### Adding a Label Component

1. *Drag and drop* a Label component from the Palette’s User Interface category to the Viewer. It will be named *Label1.*
2. Select *Label1* by clicking on it in the Viewer or the Components panel.
3. Click on its *Text* property and change it to “Martin Luther King”.
4. Add a second Label component and set its Text property to “Tap to hear the speech”

### 

### Adding a Button Component

1. *Drag and drop* a Button component from the Palette’s User Interface category to the Viewer. It will be named *Button1.*
2. Select *Button1* by clicking on it in the Viewer or the Components panel.
3. Click on its *Image* property and select the “mlk.jpg” image from the drop-down list.
4. Click on its *Text* property and change it to an empty string by deleting “Text for Button1”.

### Adding a Sound Component

1. Click on the Palette’s *Media* category to open it.
2. Drag and drop a *Player* component from the Media category to the Viewer. It will be named *Player1* and it will appear at the bottom of the Viewer under Screen1 as a *non-visible component.*
3. Select *Player1* by clicking on it in the Viewer or the Components panel.
4. Click on its *Source* property and select the “king.mp3” sound file from the drop-down list.

## Coding the App’s Behavior

It’s time to get the app to play the speech when we touch its screen. For this we will be using the Blocks editor. So switch now to the *Blocks Editor View.*

### Event Driven Programming

Mobile apps use a style of programming known as *event driven programming.* An app’s behavior depends on how the user programs the app to respond to various events. An example of an *event* would be when the user clicks on a button or when the phone’s location changes or when a text message is received. We’ll write apps that respond to all of these events.

For the I Have a Dream app, there is only *one* event that we care about, the *Button click event.*

1. Click on the *Button1* component in the Toolbox.
2. Drag and drop the *When Button1 Click* block into the Workspace.

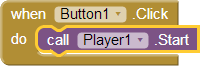


This is an example of a *when event block*, which is also called an *event handler block.*

Notice that it has a empty *do slot.*  When *Button1* is clicked, the app will *do* whatever code we put into this slot.

For the I Have a Dream app, we want it to play Martin Luther King’s speech when the button is clicked.

1. Click on the *Player1* component in the Toolbox. This will open the *Player1* drawer.
2. Drag and drop the *call Player1.play* block into the do slot. If your computer’s sound is not muted, you should hear an audible ‘click’ as the blocks snaps into place.



To summarize, whenever the user clicks on *Button1*, the app will respond by playing *Player1’s* media file. This completes the coding of the I Have a Dream app.

# Running the App on the Phone (or Tablet)

Did you remember to set up your device? If yes, continue. If not, follow these [setup instructions](http://appinventor.mit.edu/explore/ai2/setup-device-wifi.html) and come back to continue.

1. Start the the *MIT AI2 Companion* app on your phone or tablet.
2. In App Inventor, click Connect and then select AI Companion. This will display a 6-letter code both as a barcode and as plain text.
3. On the companion app, either type in the 6-letter code and click the “Connect to App Inventor” button or scan the barcode.

If everything is configured properly, you should see the I Have a Dream app on the phone and when you click the button, you should hear the speech.

***Nice work! Complete the Self-Check Exercises and Portfolio Reflection Questions as directed by your instructor.***

# 