

Hello Purr

Building your first app: HelloPurr

App Inventor Classic • App Inventor Classic • FOR APP INVENTOR 2 [CLICK HERE](#) • App Inventor Classic • App Inventor Classic

This information pertains to App Inventor 1 (Classic). For tutorials about App Inventor 2, go to the [App Inventor 2 Tutorials](#).

Now that you've set up your computer and device, and you've learned how the Designer and the Blocks Editor work, you are ready to build the HelloPurr app. At this point, you should have the Designer open in your browser, the Blocks Editor open in another window (which will show up as a java coffee cup icon in your task bar or dock), and either an Android device or an Android emulator connected to the Blocks Editor. (See [Setup instructions](#) if you do not have these things running.)

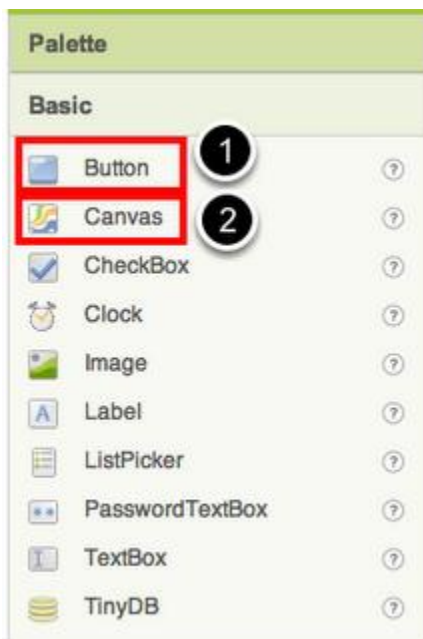
HelloPurr: tap the kitty, hear him meow

HelloPurr is a simple app that you can build in a very short time. You create a button that has a picture of a cat on it, and then program the button so that when it is clicked a "meow" sound plays.

To build HelloPurr, you'll need a image file of a cat and an audio file with a "meow" sound. Download these files to your computer by clicking the following links. To download: after clicking a link, right click on the image or sound bar and select "Save As." Save both files onto your desktop or downloads folder, or anywhere that you can easily find later.

- Kitty picture: [kitty.png](#) (Right-click and Save)
- Meow sound: [meow.mp3](#) (Right-click and Save)

Select components to design your app

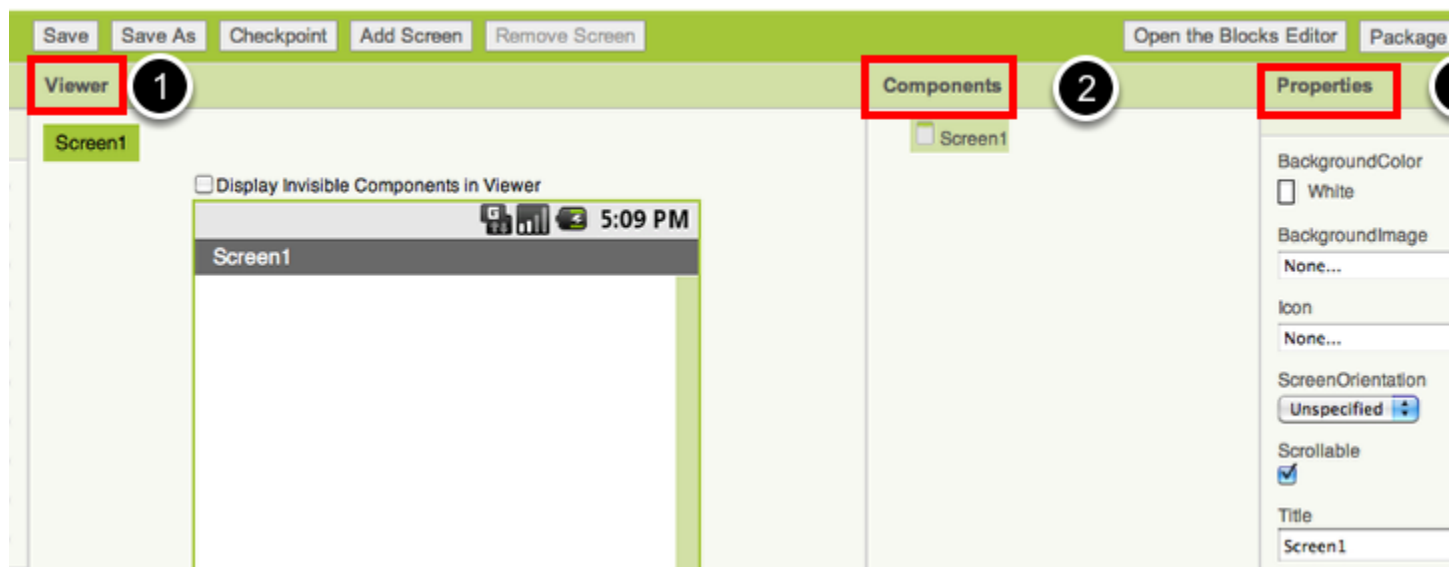


The App Inventor **Components** are located on the left hand side of the *Designer Window* under the title **Palette**. Components are the basic elements you use to make apps on the Android phone. They're like the ingredients in a recipe. Some components are very simple, like a **Label** component, which just shows text on the screen, or a **Button** component (#1 left) that you tap to initiate an action.

Other components are more elaborate: a drawing **Canvas** (#2 left) that can hold still images or animations, an **Accelerometer** sensor that works like a Wii controller and detects when you move or shake the phone, components that send text messages, components that play music and video, components that get information from Web sites, and so on.

To use a component in your app, you need to click and drag it onto the viewer in the middle of the **Designer**. When you add a component to the **Viewer** (#1 below), it will also appear in the components list on the right hand side of the Viewer.

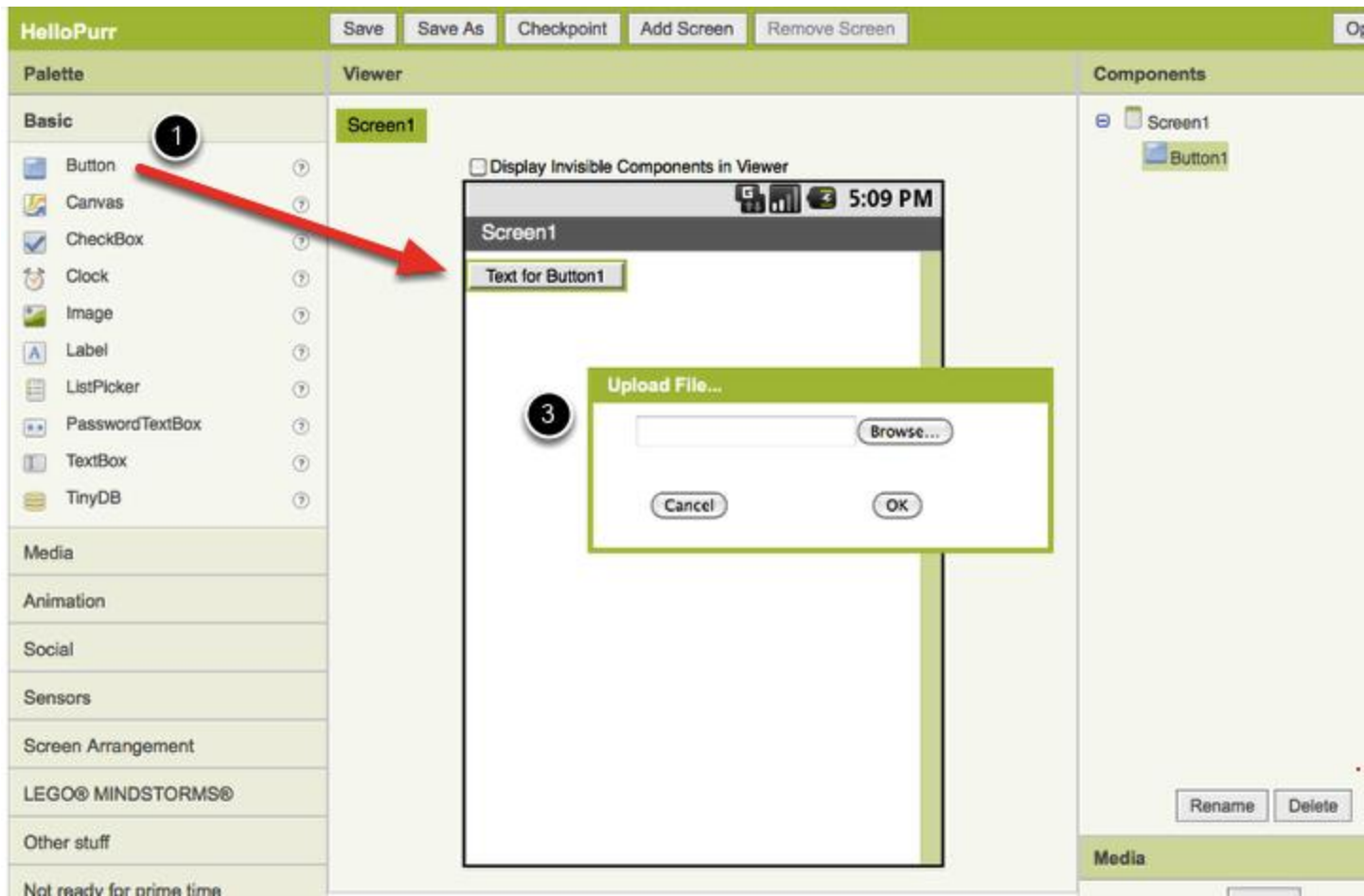
Components (#2 below) have properties that can be adjusted to change the way the component appears or behaves within the app. To view and change the properties of a component (#3 below), you must first select the desired component in your list of components.



Steps for selecting components and setting properties

HelloPurr will have a **Button** component that displays the image of the kitty you downloaded earlier. To accomplish this:

Step 1. From the **Basic** palette, drag and drop the **Button** component to Screen1 (#1). To make the button have an image of a cat, in the **Properties** pane, under Image, click on the text "None..." and click "Upload New..." (#2). A window will pop up to let you choose the image file. Click "Browse" and then navigate to the location of the *kitty.png* file you downloaded earlier (#3). Click the *kitty.png* file, click "Open", and then click "OK".



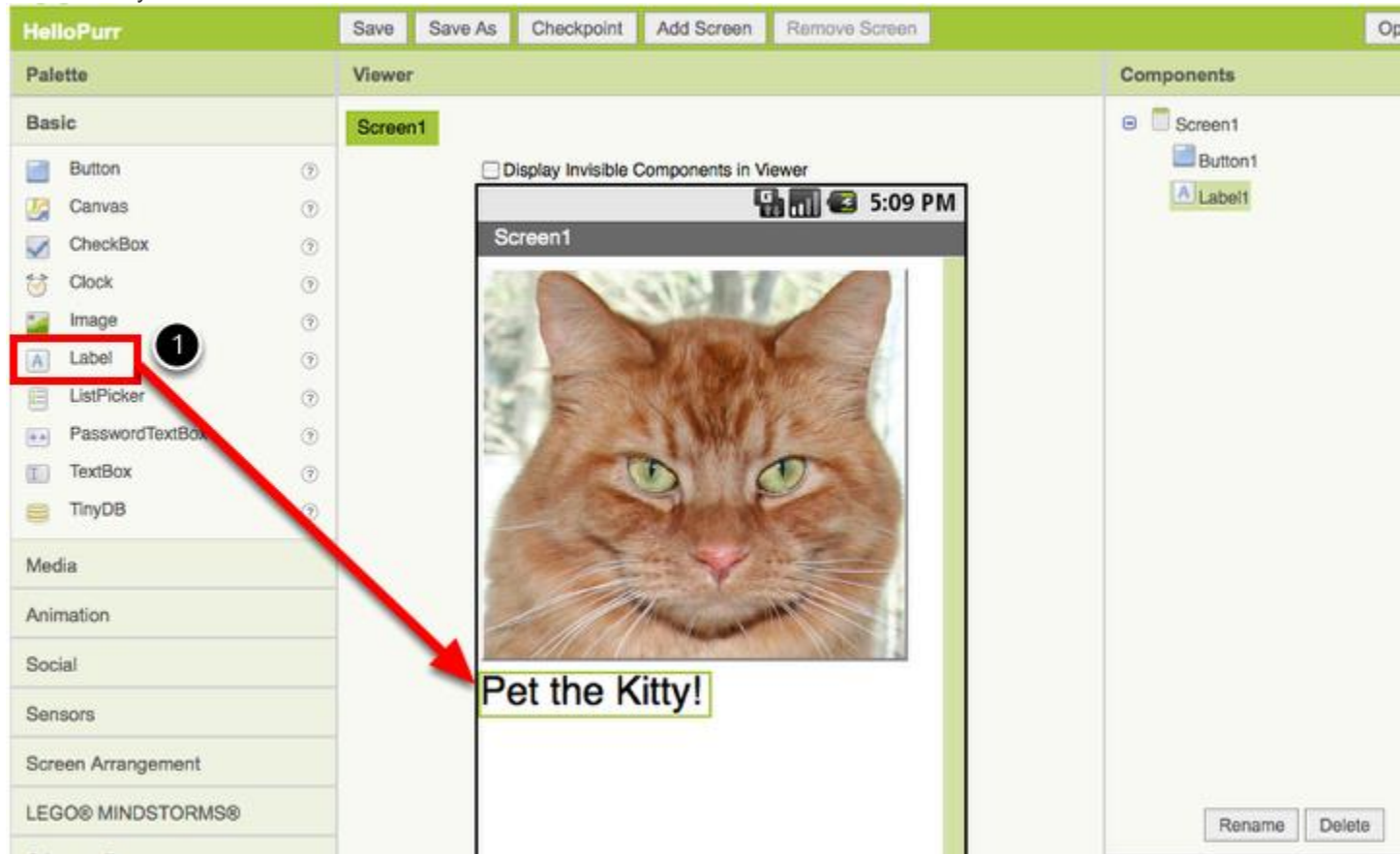
Step 2. Change the Button's **Text** property: Delete "Text for Button1", leaving the Button's text property blank so that there is no writing over the kitty's face. Your Designer should look like this:
If the entire kitty picture is not showing up, you can fix this by setting the Height and Width properties of the button to "Fill Parent". To do this, click on the Button component, go to the right-hand Properties pane, scroll down to the very bottom to where it says Width and click on the word "Automatic..." to activate the drop down list. Choose "Fill Parent". Do the same for the Height property.

HelloPurr Save Save As Checkpoint

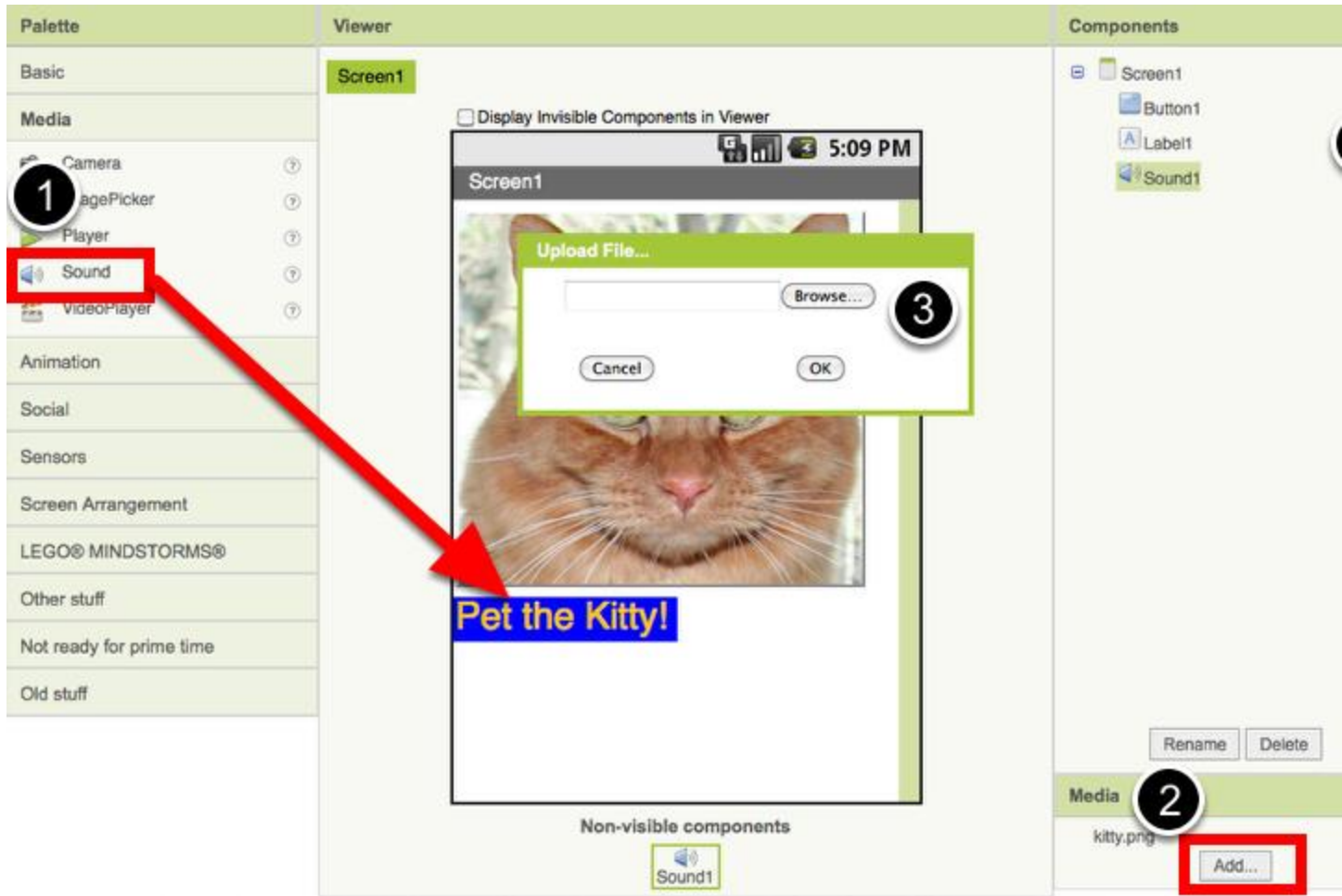
Palette	Viewer	Component
Basic <ul style="list-style-type: none"> Button ? Canvas ? CheckBox ? Clock ? Image ? Label ? ListPicker ? PasswordTextBox ? TextBox ? TinyDB ? 		<ul style="list-style-type: none"> Screen Button <p>Rename...</p> <p>Media</p> <ul style="list-style-type: none"> kitty.png <p>Add</p>
Media		
Animation		
Social		
Sensors		
Screen Arrangement		
LEGO® MINDSTORMS®		
Other stuff		
Not ready for prime time		
Old stuff		
For internal use only		

Step 3. From the **Basic** palette, drag and drop the **Label** component to the Viewer (#1), placing it below the picture of the kitty. It will appear under your list of components as **Label1**. Under the **Properties** pane, change the **Text** property of Label1 to read "Pet the Kitty" (#2). You'll see the text change in the Designer and on your device. Change the **FontSize** of Label1 to 30 (#3). Change the **BackgroundColor** of Label1 by clicking on the box (#4): you can change it to any color you like. Change

the TextColor of Label1 (#5) to any color you like. Here, the background color is set to blue and the text color is set yellow.



Step 4. Under Palette, click on the **Media** drawer and drag out a **Sound** component and place it in the Viewer (#1). Wherever you drop it, it will appear in the area at the bottom of the Viewer marked **Non-visible components**. Under the **Media** pane, Click *Add...* (#2) Browse to the location of the **meow.mp3** file that you downloaded earlier and upload it to this project (#3). Under the Properties pane, see that the **Source** property currently says *None...* Click the word *None...* to change the Sound1 component's **Source** to *meow.mp3* (#4).



Programming with the Blocks Editor

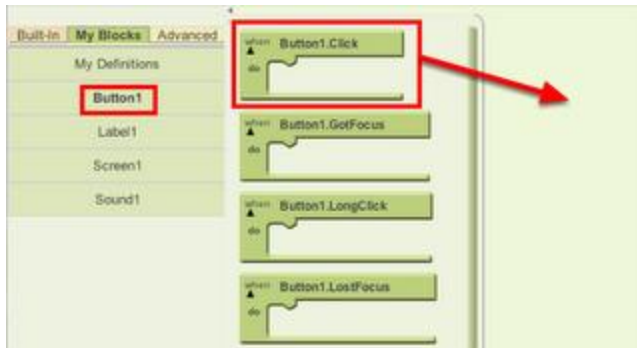
So far you have been arranging your app's screen and components in the *Designer*, which is in a web browser window. To start programming the behavior of the app, you need to go to the *Blocks Editor*. If you do not have a Blocks Editor running, click the button **Open the Blocks Editor** in the upper right of the Designer window. If you need help getting the Blocks Editor started, go back to the [setup instructions](#) for help.

Note: An easy way to switch between the Blocks Editor and Designer is to use the task bar that shows the applications running on your computer. The Blocks Editor is running locally as a java program and is represented by a Coffee Cup icon. The Designer is running in your web browser so you can get to it by clicking on your browser icon.

Once you have the Blocks Editor in front of you, continue to the next step to start programming your app with blocks.

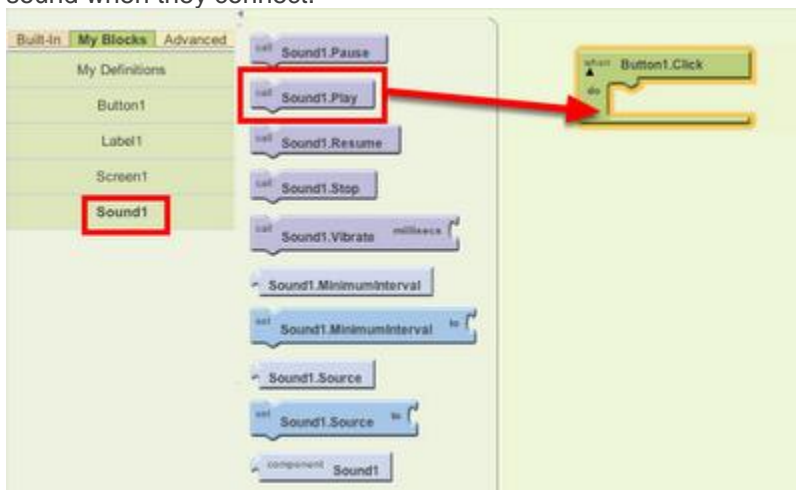
Making the sound play

Step 1. Under the My Blocks palette on the left side of the Blocks Editor, click the **Button1** drawer to open it. Drag and drop the Button1.Click block in the work area (the open area on the right).



Those green blocks are called **event handler** blocks. The event handler blocks specify how the phone should respond to certain events: a button has been pressed, the phone is being shaken, the user is dragging her finger over a canvas, etc. The event handler blocks are green in color and use the word *when*. For example, when Button1.Click is an event handler

Step 2. Click the **Sound1** drawer and drag the Sound1.Play block and connect it to the "do" section of the when Button1.Click block. The blocks connect together like puzzle pieces and you can hear a clicking sound when they connect.



The purple and blue blocks are called **command** blocks, which are placed in the body of event handlers. When an event handler is executed, it runs the sequence of commands in its body. A command is a block that specifies an action to be performed (e.g., playing sound) when the event (e.g., pressing Button1) is triggered.

Your blocks should look like this at this point:



Now you can see that the **command block** is in the **event handler**. This set of blocks means; "when Button1 is clicked, Sound1 will play." The event handler is like a category of action (e.g., a button is being clicked), and the command specifies the type of action and the details of the action (e.g., playing sound a specified sound).

You can read more about how blocks work here: [Understanding Blocks](#).

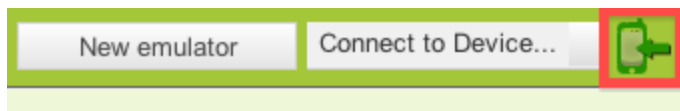
Try It! When you click the button you should hear the kitty meow. Congratulations, your first app is running!

Note: there is a known issue with the Sound component on some devices. If you see an "OS Error" and the sound does not play - or is very delayed in playing, go back into the Designer and try using a Player component (found under Media) instead of the Sound component.

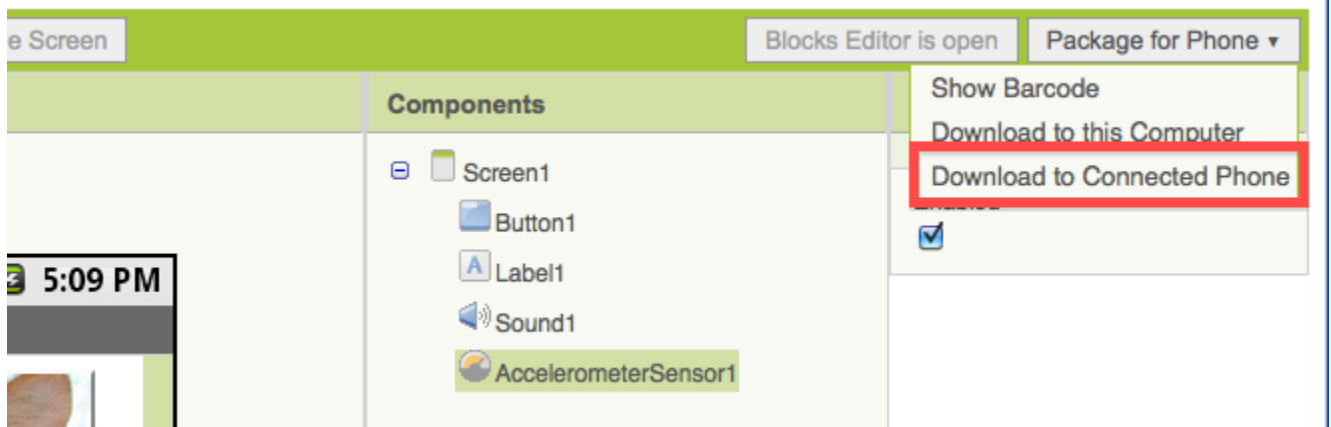
Packaging your app

While your device (emulator or phone/tablet) has been connected to App Inventor, your app has been running in real time on your device. If you disconnect the emulator/phone/tablet from the Blocks Editor, the app will vanish. You can always make it return by reconnecting the device. To have an app running without being connected to App Inventor, you must "**package**" the app to produce an application package (apk file).

To "package" the app to your phone, connect your phone to the Blocks Editor and make sure that the phone icon's color in the upper right of the Blocks Editor is green.



Then go back to the Designer and choose "Package for Phone" at the upper right of the designer page. App Inventor will present three options for packaging:



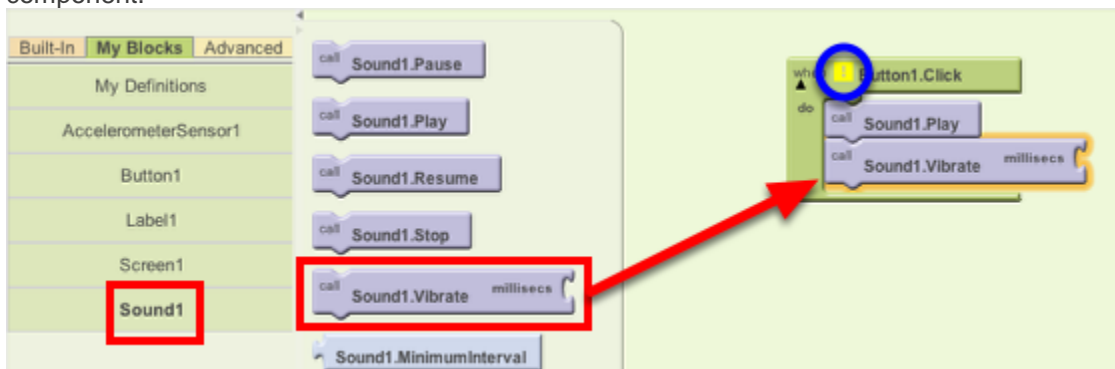
1. **Show Barcode:** You can generate a Barcode (a QR Code), which you can use to install the app on a phone or tablet that has a camera, with the aid of a barcode scanner, like the ZXing barcode scanner (freely available in Google Play).

Note: this barcode works only for your own device because it is associated with your google account. If you want to share your app with others via barcode, you'll need to download the .apk file to your computer and use a third-party software to convert the file into a barcode. More information can be found [here](#).

2. **Download to this Computer:** You can download the app to your computer as an apk file, which you can distribute and share as you like by manually installing it on other devices. (sometimes called "[side loading](#)").

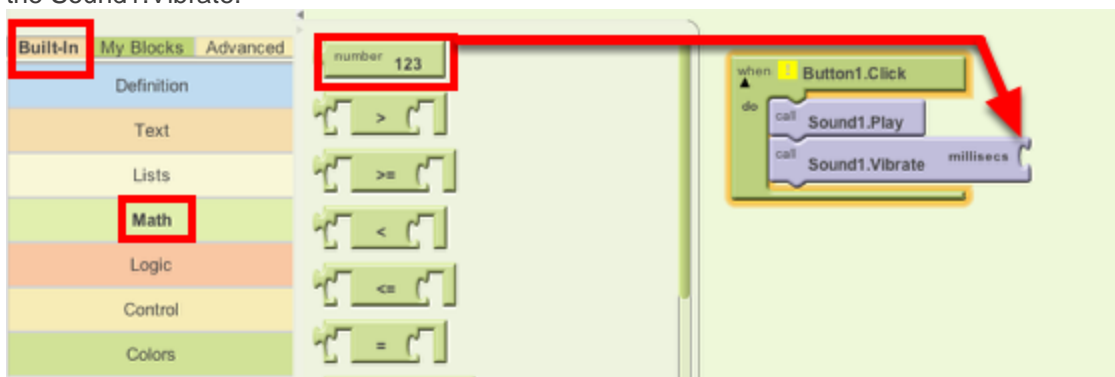
3. **Download to Connected Phone** You can download the apk file directly to the device that is connected to the Blocks Editor. Note that this works even if you are using the emulator as your device!
Challenge! Make the cat purr

The challenge is to have the cat purr when the phone is shaken. Go to the Blocks Editor and open the **Sound1** drawer and drag the Sound1.Vibrate block and place it under the Sound1.Play block. You will see a yellow warning icon at the left corner of the block, which means the block has a missing component.

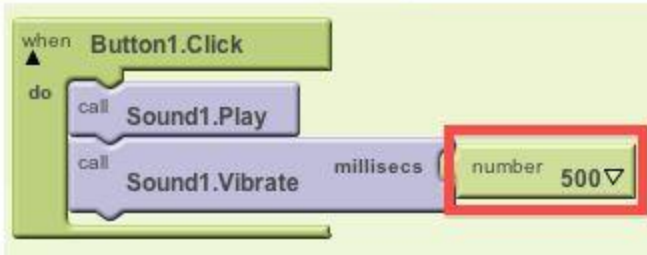
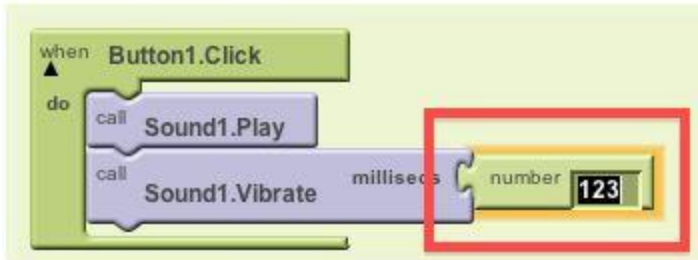


The Sound1.Vibrate block has an open slot, which means you need to plug something into it to specify more about how the behavior should work. Here, we want to specify the length of the vibration. Numbers are calculated in thousandths of a second (milliseconds): to make the phone vibrate for half a second, we need to plug in a value of **500 milliseconds**.

Go to the Built-In palette, go to the **Math** drawer, drag the number block and place it at the socket of the Sound1.Vibrate.



After you place the number block, click the number "123". It highlights the number in black: type "500" with your keyboard.



Done! Notice the yellow warning icon is gone: the block has no longer missing component.

