Setting up the environment

In this activity we're going to prepare our enviroment to start programming apps. We'll need:

- A computer connected to our home WIFI network. This is the computer we're going to design and program the app.

- An smartphone (or even better a tablet) running an Android operating system connected to the same home WIFI network as the computer. It will be used to test and finally install the app.

Steps:

1. Go to GooglePlay and install the MIT AI2 App on your smartphone or tablet.

Ilustración 1: Image from Google Play

2. Open a browser (Firefox, Chromium, Chrome etc) and go to MIT appinvento's site: link

3. Click on the "Create Apps" button and register using your Google account or any other email account you want (If you use a Gooogle account you won't need to sign up)

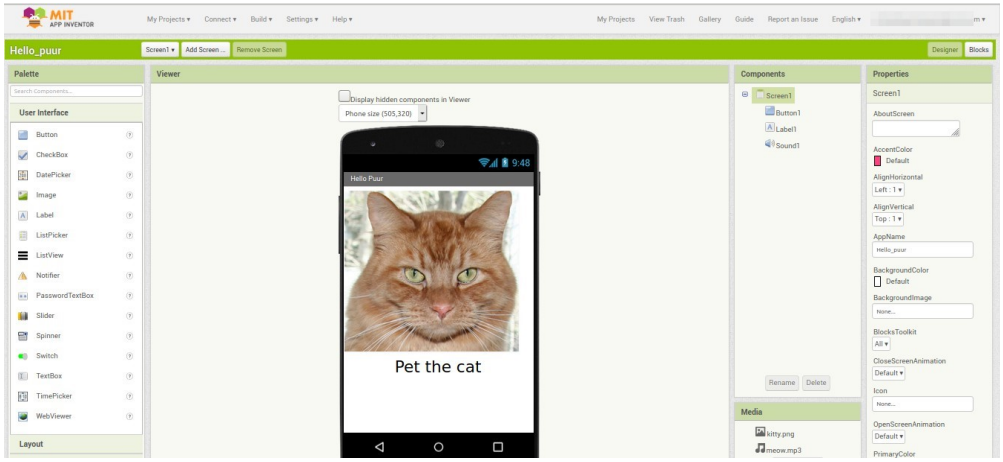


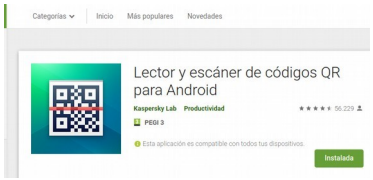
Ilustración 2: AppInventor welcome page Screenshot

4. On your tablet/smartphone launch the MIT AI2 Companion APP

Ilustración 3: MIT AI2 Companion screen

5. Congratulations!!!, now your're ready to start programming your first app.

6. When the app is finished, to be able to install it on your terminal you'll need a QR code reader, if your terminal doesn't have one install it from GooglePlay.



My first app hello purr

This simple app is usually the starting point for most appinventor programmers. It illustrates the basic techniques, components and workflow to create any app.

We we'll describe the app making process in great detail.

This app consists of just a main screen diplaying a button with the picture of a cat and a label with the text "Pet the cat". When the button is touched the cat meows and the phone vibrates for 1 second.

In addition of the smartphone/tablet and the computer properly configured common to all the applications in this document, in this case you'll ned: to download the following resources:

Ilustración 4: Final result

- A picture with the image of a cat: link

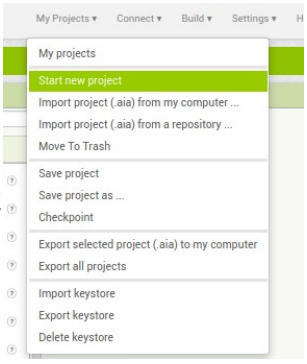
- An mp3 audio file with the meoow sound: link

Steps:

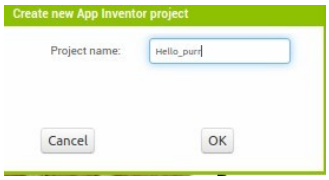
1. Download the picture and the meoow sound of the cat on your downloads foder.

2. Enter the appinventor programming environment and lauch the MIT AI2 companion app in your smartphone/tablet.

3. In the appinventor programming environment main menu click on *My projects Start new project*

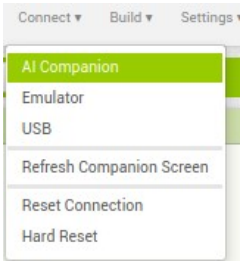


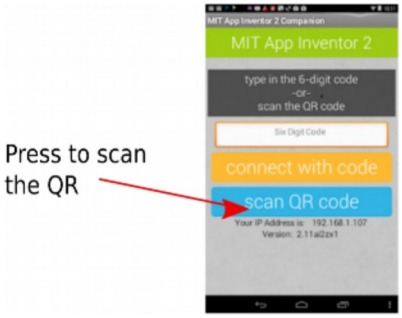
4. Write **Hello_purr** as the project name and click ok.



5. In the appinventor main menu click on *Connect|AI Companion*

When the app QR code pops up scan it with your smartphone/tablet clicking on the button **Scan QR Code**

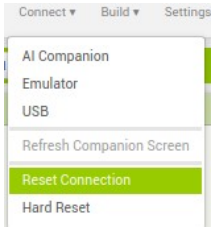




After a while you'll see that the screen of your smartphone/tablet displays a functional version of the app you're programming, every change you make in the programming environment propagates inmmediately to the smartphone/tablet.

Synchronizing the computer and the smartphone/tablet allows you to test the app as yoy make it, before finishing it completely, and without needing to install it on your terminal.

It might happen that the appinventor programming environment and the smartphone/tablet desynchonize. In this case do this:

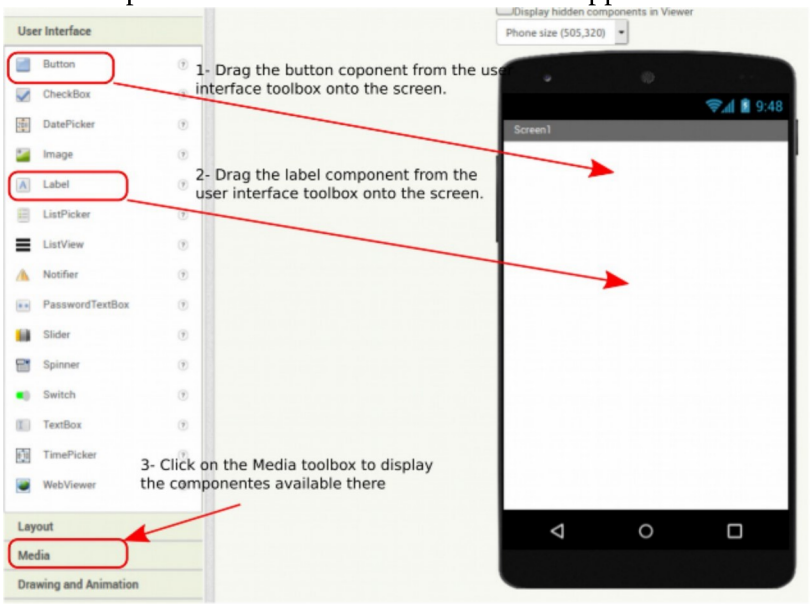


- In the appinventor menu click on *Connect | Reset connection*

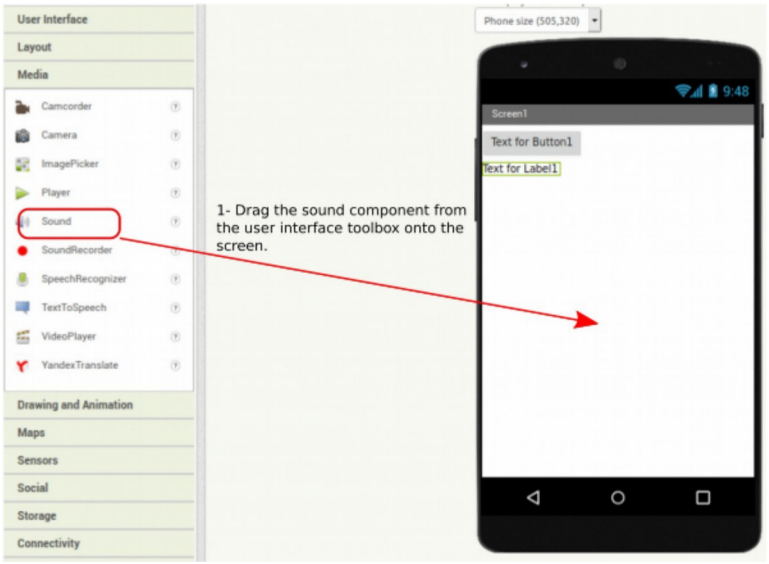
- Restart the MIT AI2 Companion app on your smartphone/tablet

- In the appinventor menu click on *Connect | AI Companion* and when the QR code appears scan it to reconnect the computer and the *smartphone/tablet* again.

6. By now we only have an application with an empty screen. We'll drag all the visual and non vusual components on the screen to create the app user's interface.



7. In the appInvetor programming environment select the Media toolbox on the left panel, drag the Sound component onto the bottom part of the empty screen.



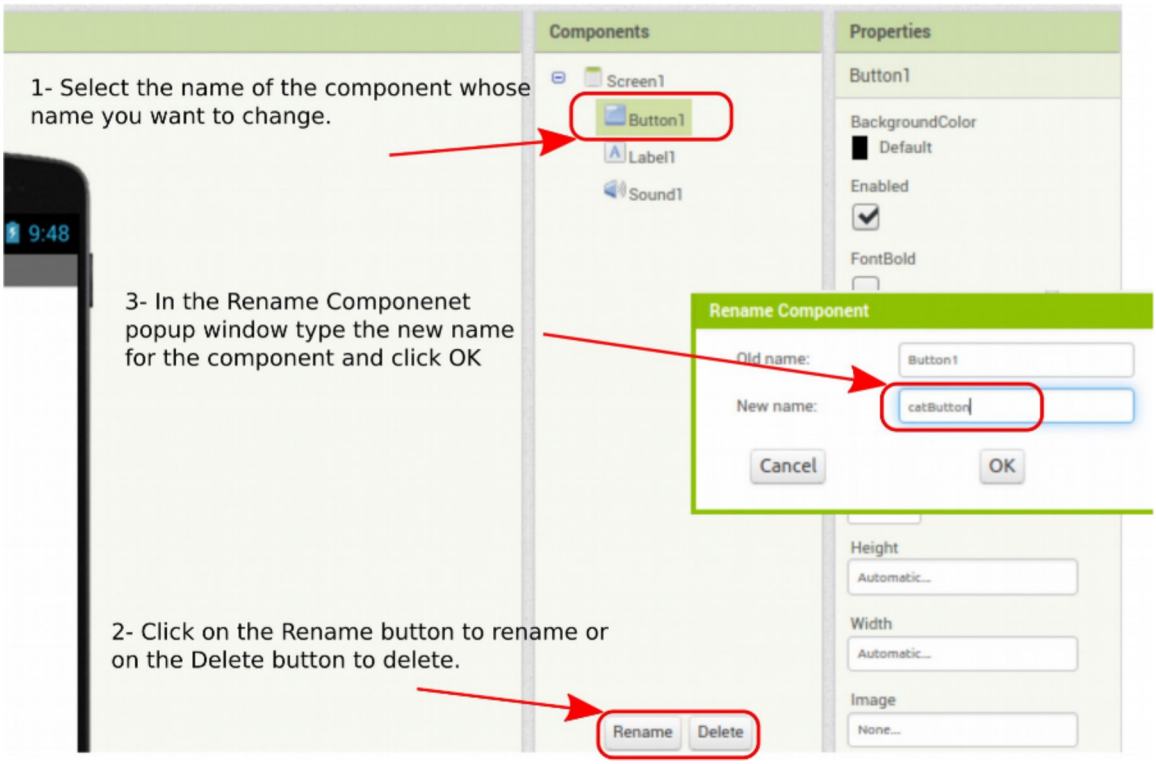
8. Now we're going to change the defualt name of the components we've just added by a more meaningful one. Using meaningful names for components and variables is always a good programming practice.

The new names will be:

for the button: catButton

for the label: instructionsLabel

for the sound: meoowSound.

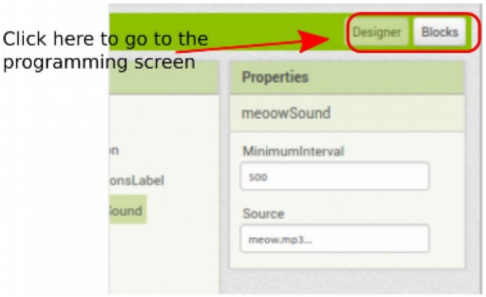


9. Next thing we're going to do is to change several properties of the components that make up our app. Each component has a number of properties that can be customized, the properties for the component selected on the **Components** panel are shown on the **Properties** panel, on the right end of the screen.

In the table below we'll summarize the properties to be changed. When you had set them all the app scrren has to look exactly the same as the image "Final result" at the beginning of the activity.

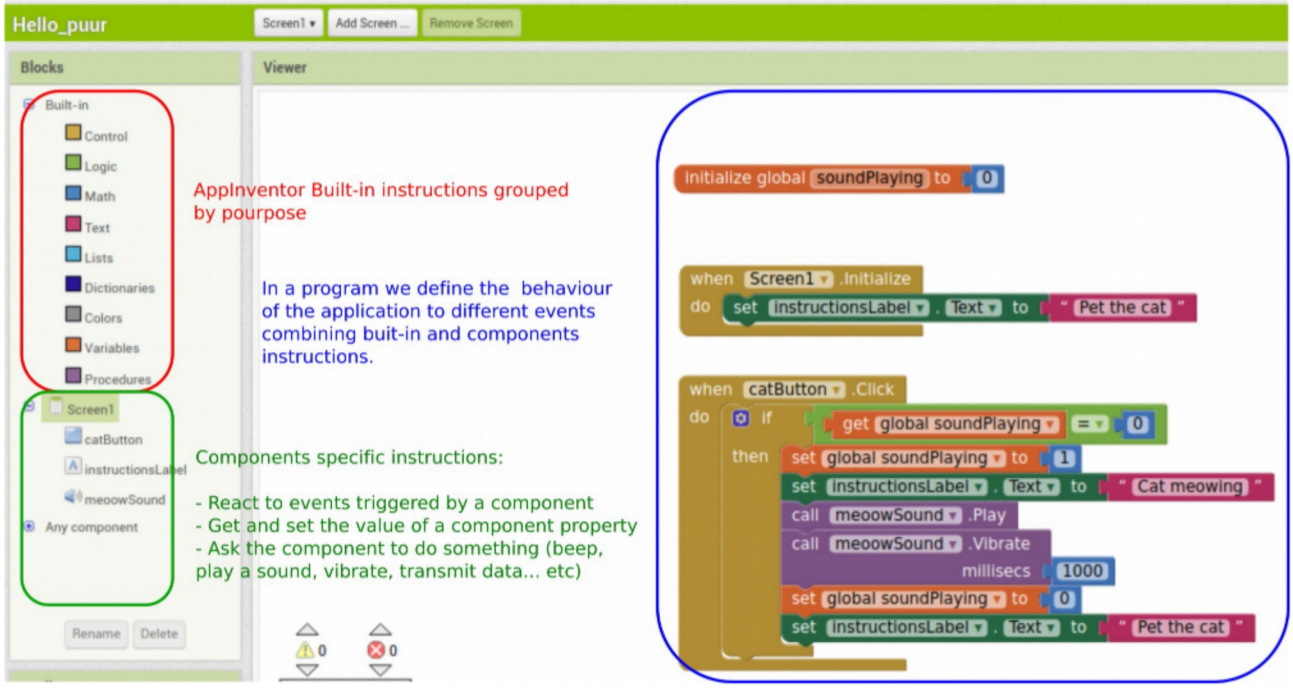
Component	Property Name	Property Value	Remarks
Screen1	Title	Hello Purr	
CatButton	Width	Fill Parent	The button will be as wide as its parent component, in this case the Screen.
	Image	Kitty.png	Select *Upload File* and upload the cat image downloaded in step 1 from the Downloads folder.
	Text		Leave it blank
instructionsLabel	Text	Pet the cat	
	Width	Fill Parent	
	Font size	30	Make the text 30 points in size
	Text alignement	Center	Center the text on the screen
MeowSound	Source	meow.mp3	Select *Upload File* and upload the meow mp3 file downloaded in step 1 from the Downloads folder.

10. The app user interface is finished now. We'll continue programming its behavoir: When the user touches the cat button the meow sound will be played and the terminal will vibrate for 1 second (1.000 milliseconds).



To go to the app programming screen we have to click on the B**locks** button on the upper right corner of the window.

11. Now we're going to learn how to program our app. In our screen the programming area **(viewer)** will be blank, below you can see the way it looks when there's a program on it.



On the left side of the programming screen we can find two sets of programming instructions:

Built-in instructions: Are the programming instructions common to other programming languages, they're grouped by pourpose, each group has a color, the most commonly used groups are:

- Control (brown-yellow) : Instructions that allow the programmer control the flow of the program, opening and closing screens, terminating the app, stopping the execution for a period of time etc…

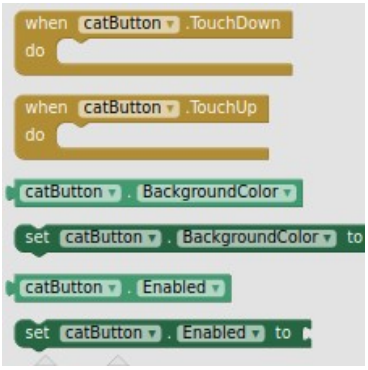
- Logic (green): Instructions to verify logic conditios

- Math: (blue) instructions to do mathematical calculations, generate random numbers, manipulate and compare numbers.

- Text (purple): Instructions to manipulate textual information.

- Variables: (orange): Instructions to create and manipulate variables

- Procedures: (violet): Instructions to create new user defined instructions



Components specific instructions: Are the programming instructions that allow the programmer interact with its different components. They can be classified as:

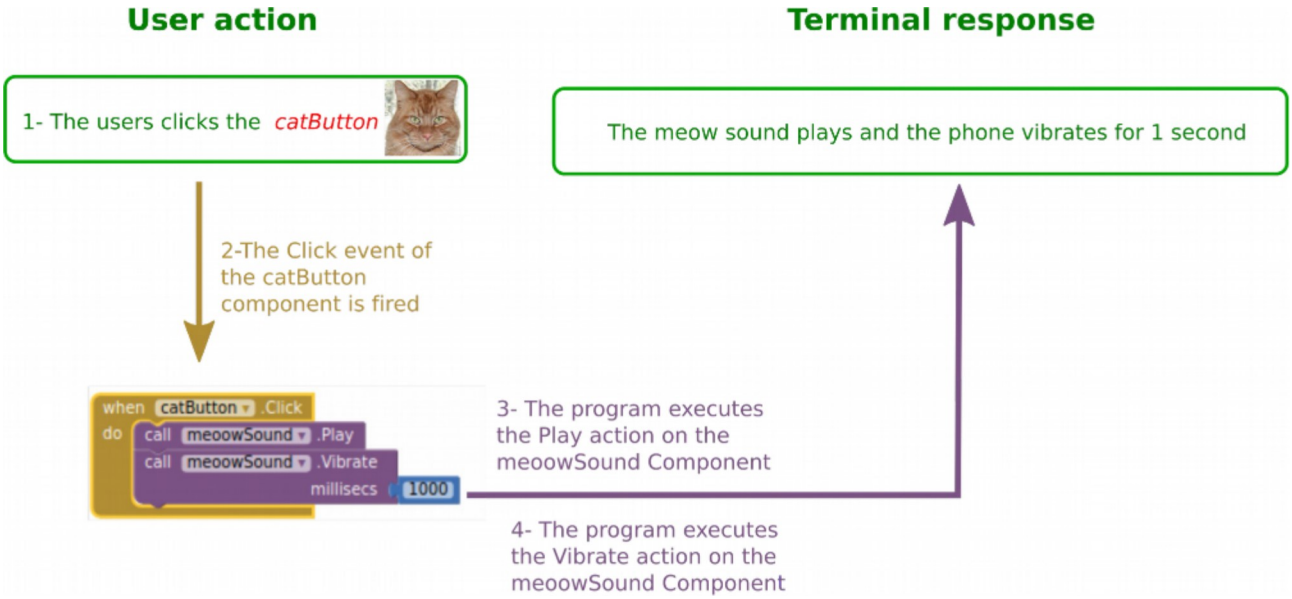
- Event handlers (brown-yellow) : The instructions inside the block will be executed when a given event happens (i.e a button is pressed or released, the used touches the screen, a timer goes off, data are received via bluetooth...etc.

- Property setters (dark green): Allow the programmer to set a componet property to a given value when the app is running (i.e disable a button, hide a text, change the text displayed in a label component etc).

- Property getters (light green): Contain the current value of a compoenent propertt. They allow the programmer to get the value a componet property when the app is running (I.e: read the text entered by the user in a form, check whether a box has been checked etc).

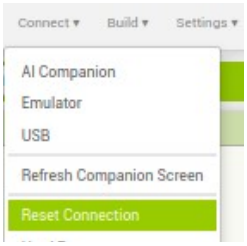
- Action doers (violet): Make the component do something (I.e: Play a sound, transmit data via bluetooth, vibrate the terminal etc...)

12. Now we're going to choose and combine the right event handlers and instructions to create our program.



Drag onto the programming area **(Viewer)** the blocks shown in the image above to program the app.

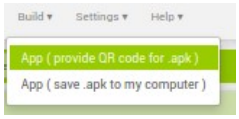
When ythe program is finished, go back to the *Designer* screen clicking on the **Design** button on the upper right corner of the window.



13. Test the app in the terminal synchronized with the computer, it should work!!.

14. When you had finished testing the app disconnect your terminal from the computer clicking *Connect/Reset Connection* and close the MIT AI2 Companion app on your smartphone/tablet

15. Now we're going to install the app on your terminal in a permanent way. On the main appInventor menu click on B*uild/Generate QR Code* to generate the apk file that contains the app installer. When the QR Code appears on the screen take a picture of it .



The QR Code reader on your smartphone will open a browser to download the **hello_purr.apk** containing the app installer..

When you download the app, your smartphone/table will ask you for permission to install an insecure app, grant the premission only for this time and go on.

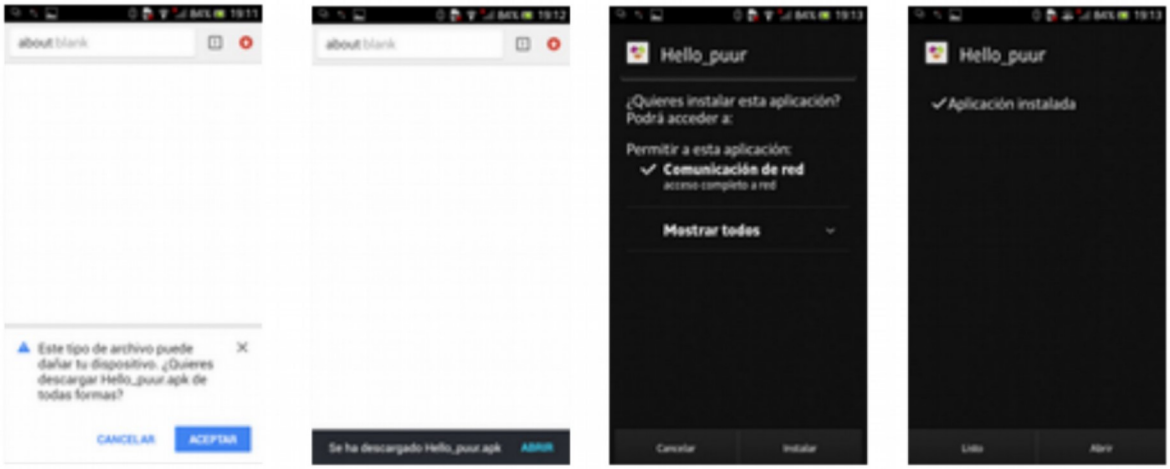


Ilustración 5: Sequence of screens to install the app on a smartphone

16. When the app is downloaded and installed click on **Abrir** and enjoy it!!.

Congratulations, you're one of the very few people in the whole world that know how to create, deploy and install a Google app.