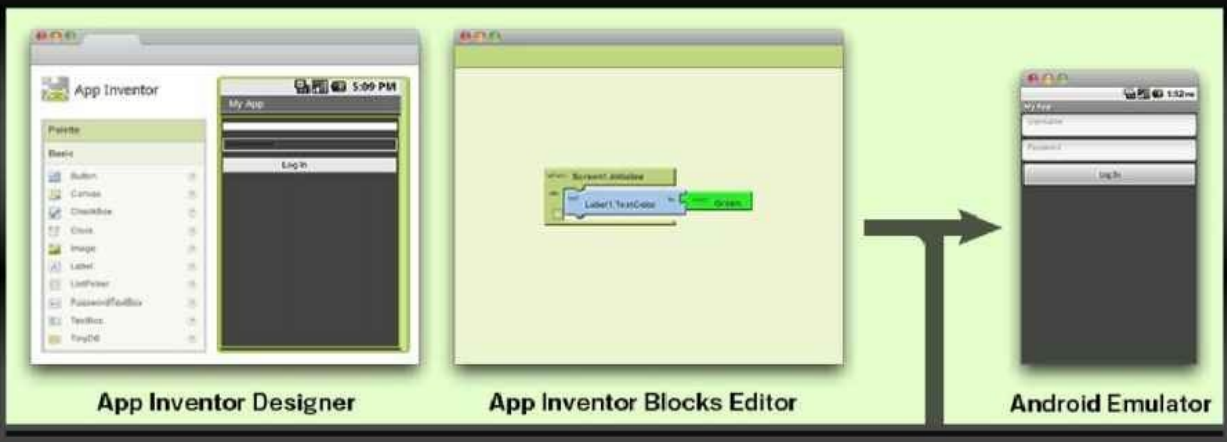
CREATE YOUR OWN APP(BASIC)

WITH APP INVENTOR 2



Features:-

- 25 Basic Android Apps Tutorials.
- Easy-to-Follow with screenshot given for each step.
- All colored Pages
- App Based Learning and Interactive Design
- No coding or Programming Experience Required to use this book.

RUPESH TIWARI

Contents

Chapter 1: Introduction to APP INVENTOR 2

What is App Inventor 2?

MIT App Inventor is an innovative beginner's introduction to programming and app creation that transforms the complex language of text-based coding into visual, drag-and-drop building blocks. The simple graphical interface grants even an inexperienced novice the ability to create a basic, fully functional app within an hour or less. You just need to open below link in your browser.

http://ai2.appinventor.mit.edu/

Overview of App Inventor

Every Android app consists of two Parts:-

1. **Frontend Design** (Look and feel of Android app) i.e, App Screen .Frontend Design is created using Designer in App inventor 2.
2. **Backend Logic** (This contains the logic which will decides how our app will work).Backend Logic is created in the Blocks Editor.

App Inventor consists of the Designer and the Blocks Editor. These are described in detail below.

Designer:-

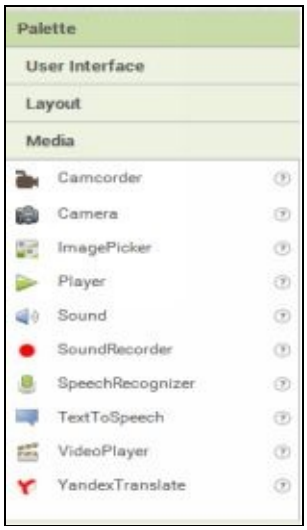
Palette:-

It has different types of Visible components (e.g, Textbox, image which will be displayed on app screen) and Non-Visible components (e.g, Camera,Player which will not be displayed on the app screen but will be used when we will create Backend Logic) which we can use to design our app screen:-

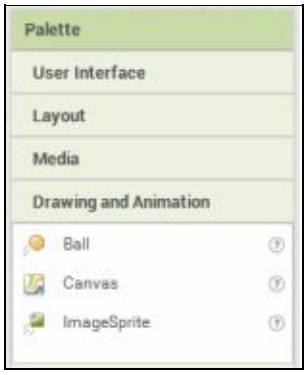
1. **User Interface:**

2. Layout:

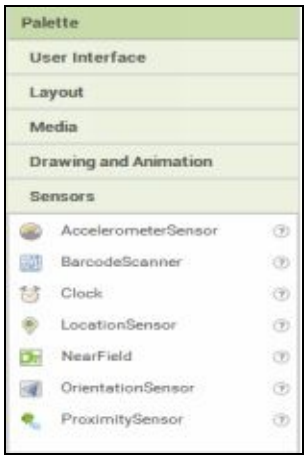
3. Media:



4. Drawing and Animation:

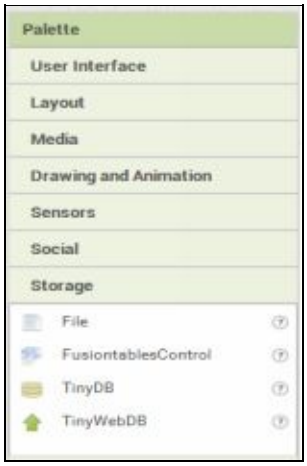


5. Sensors:

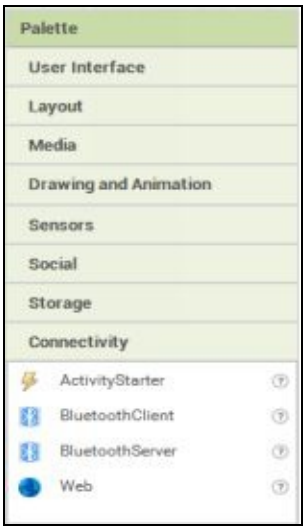


6. Social:

7. **Storage:**



8. **Connectivity:**

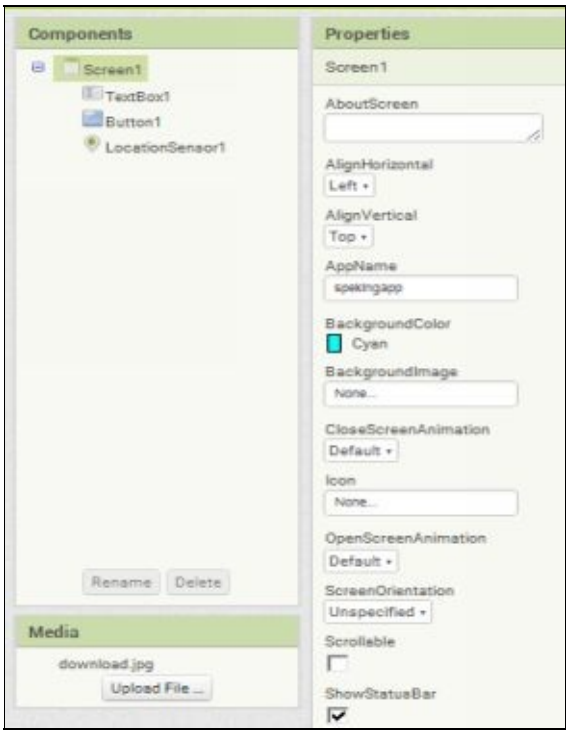


Viewer:-

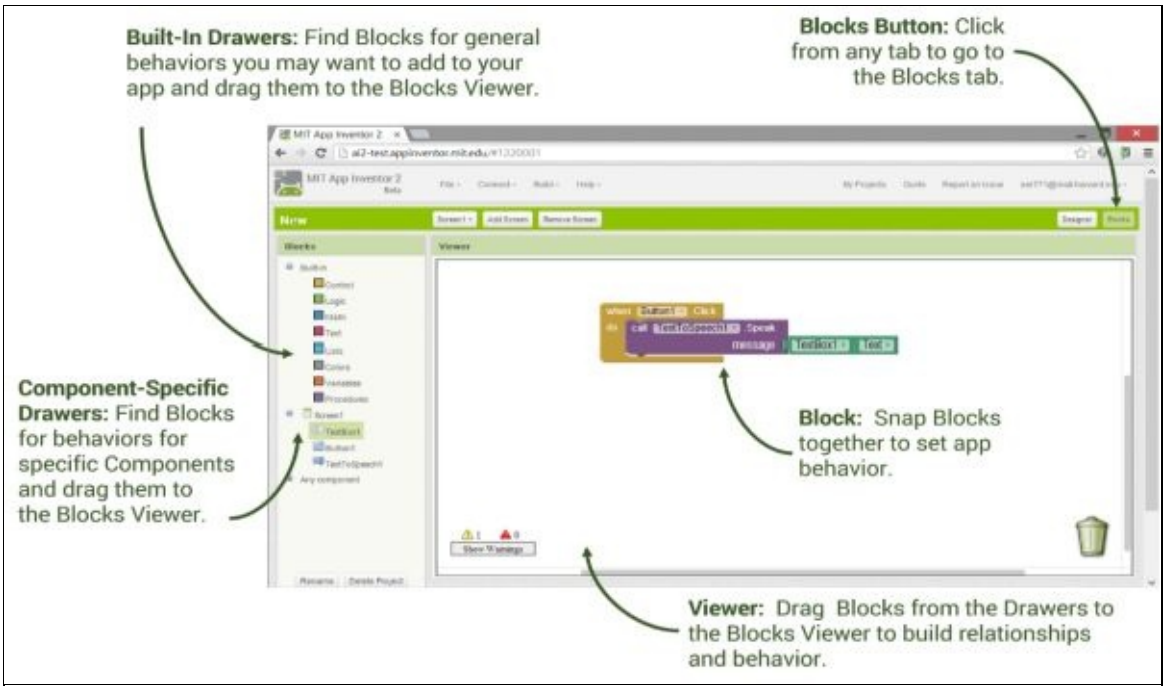
In viewer you will see how your app screen will look like and what components will be shown on your app screen.

Components and Properties:-:

Under components you will be able to see all the objects/Components added to your app and Under Properties Tab you will see Component/Object related properties which you can modify.

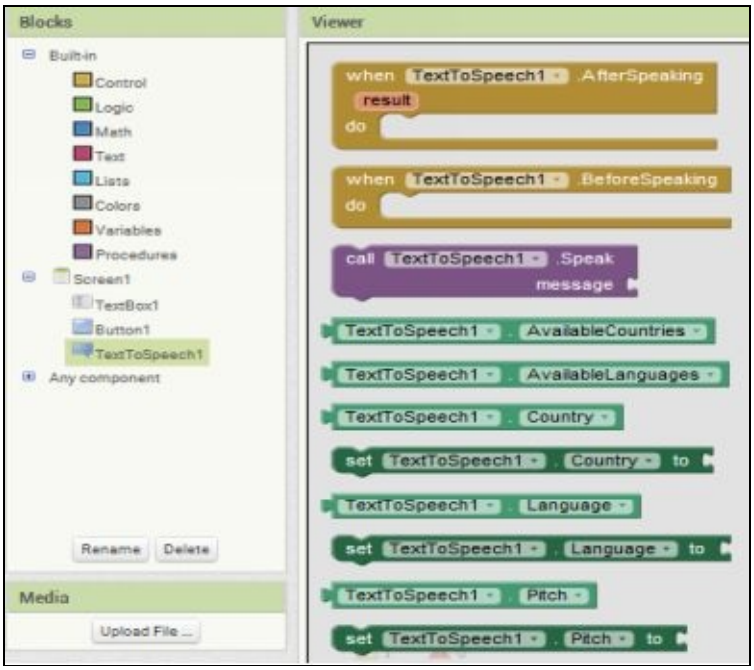


Blocks Editor:-

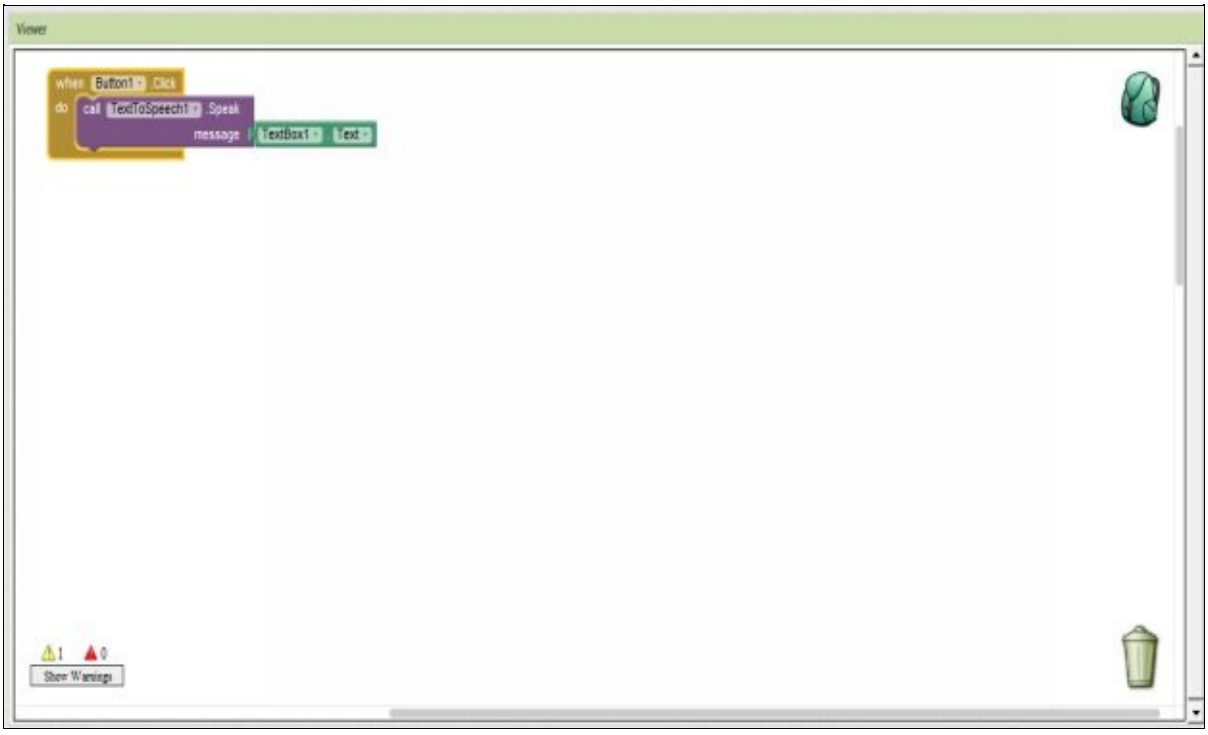


Blocks:-

Under Blocks all the visible and non-visible components will be shown and when you click on a component you will see all the associated logic blocks which you can use to create your app logic.



Viewer



Setting up App Inventor and Testing APP:-

You can use App Inventor without downloading anything to your computer! You'll develop apps on website: ai2.appinventor.mit.edu. To do live testing you can use one of the below options:-

1) **Live testing on your Android Device (Phone/Tablet/etc…)**

Step 1: Download and install the MIT AI2 Companion App on your phone.

After downloading, step though the instructions to install the Companion app on your device. You need to install the MIT AI2 Companion only once, and then leave it on your phone or tablet for whenever you use App Inventor.

Note 1: If you are unable to use the QR code, you can still install MIT AI2 Companion on your phone or tablet. Use the Web browser on your device to go to the Google Play Store; look for MIT AI2 Companion in the store. Once you find Companion, click the INSTALL button for the Companion app.

Note 2: If you choose not to go through the Play store and instead load the app directly (aka "side load), you will need to enable an option in your device's settings to allow installation of apps from "unknown sources". To find this setting on versions of Android prior to 4.0, go to "Settings > Applications" and then check the box next to "Unknown Sources". For devices running Android 4.0 or above, go to "Settings > Security" or "Settings > Security & Screen Lock" and then check the box next to "Unknown Sources" and confirm your choice.

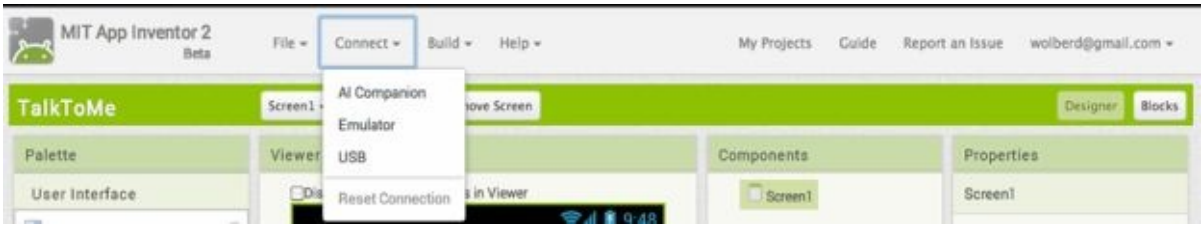
Step 2: **Connect both your computer and your device to the SAME WiFi Network**

App Inventor will automatically show you the app you are building, but only if your computer (running App Inventor) and your Android device (running the Companion) are connected to the same WiFi Network.

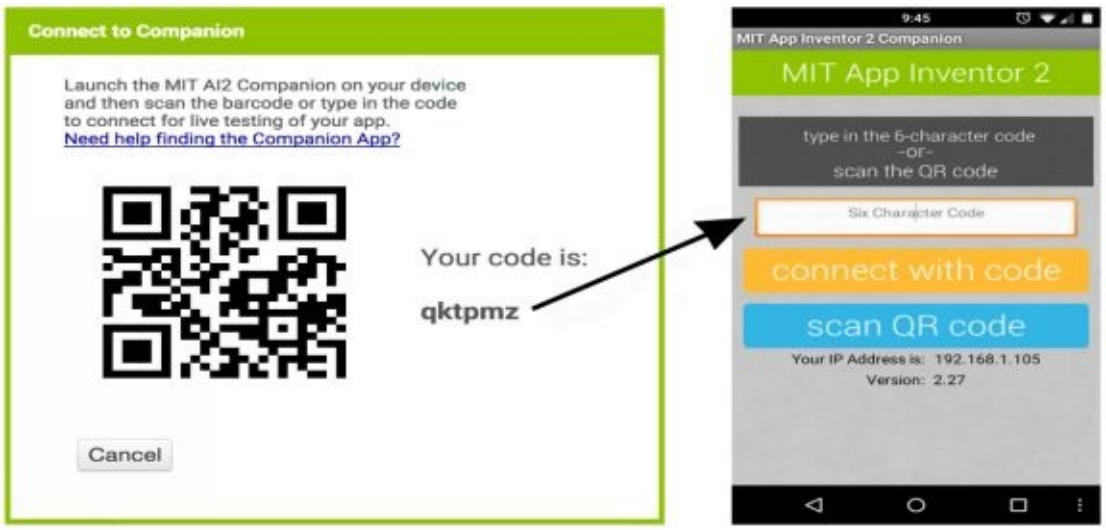
Step 3: **Open an App Inventor project and connect it to your device**

Go to App Inventor and open a project (or create a new one — use *Project* > *Start New Project* and give your project a name).

Then Choose "Connect" and "AI Companion" from the top menu in the AI2 browser:



A dialog with a QR code will appear on your PC screen. On your device, launch the MIT App Companion app just as you would do any app. Then click the "Scan QR code" button on the Companion, and scan the code in the App Inventor window:



Within a few seconds, you should see the app you are building on your device. It will update as you make changes to your design and blocks, a feature called “live testing”.

If you have trouble scanning the QR code or your device does not have a scanner, type the code shown on the computer into the Companion’s text area on your Android device exactly as shown. The code is directly below where the screen on your PC shows “Your code is” and consists of six characters. Type the six characters and choose the orange “Connect with code”. Do not type an Enter or carriage return: type just the six characters followed by pressing the orange button.

2. Installing and Running the Emulator in AI2

If you do not have an Android phone or tablet, you can still build apps with App Inventor. App Inventor provides an Android emulator, which works just like an Android but appears on your computer screen.

Step1:-Installing App Inventor 2 Setup on Windows

Installing the Windows software for App Inventor Setup has two parts:

1. Installing the App Inventor Setup software package. This step is the same for all Android devices, and the same for Windows XP, Vista, and 7.
2. If you choose to use the USB cable to connect to a device, then you'll need to install Windows drivers for your Android phone.

NOTE: App Inventor 2 does not work with Internet Explorer. For Windows users, we recommend using either Chrome or Firefox as your browser for use with App Inventor.

Installing the App Inventor Setup software package

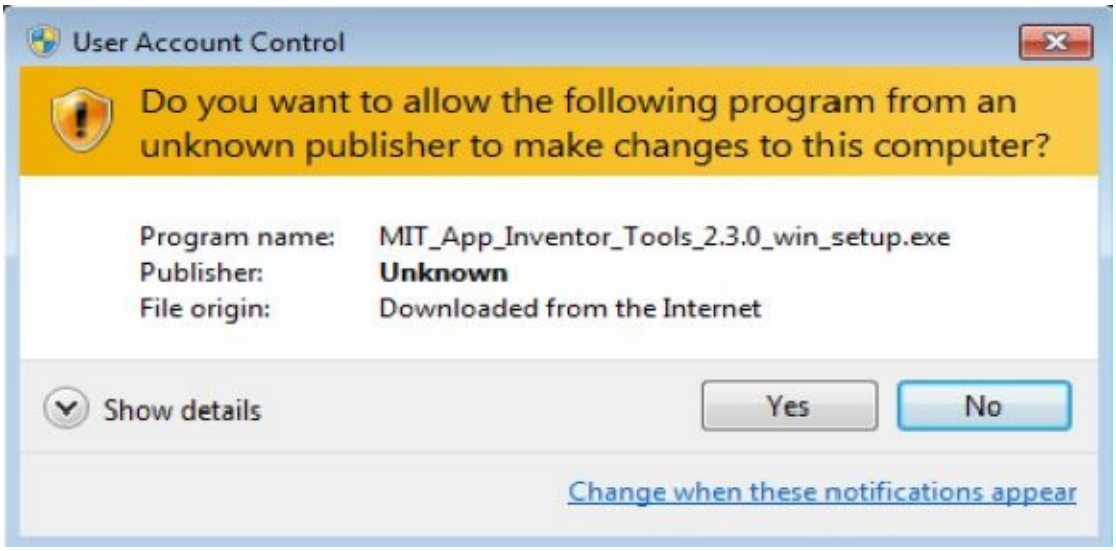
You must perform the installation from an account that has administrator privileges. Installing via a non-administrator account is currently not supported.

If you have installed a previous version of the App Inventor 2 setup tools, you will need to uninstall them before installing the latest version.
Follow the instructions given below:-

Installing App Inventor 2 Setup on Windows

Download the installer from http://appinv.us/aisetup_windows

1. Locate the file MIT_Appinventor_Tools_2.3.0 (~80 MB) in your Downloads file or your Desktop. The location of the download on your computer depends on how your browser is configured.
2. Open the file.
3. Click through the steps of the installer. Do not change the installation location but record the installation directory, because you might need it to check drivers later. The directory will differ depending on your version of Windows and whether or not you are logged in as an administrator.
4. You may be asked if you want to allow a program from an unknown publisher to make changes to this computer. Click yes.



Locating the Setup software

In most cases, App Inventor should be able to locate the Setup software on its own. But if it asks for the location of the software, the path to enter is C:\Program Files\Appinventor\commands-for-Appinventor. If you are using a 64-bit machine you should type Program Files (x86) rather than Program Files. Also, if you did not install the software as an administrator, it was installed in your local directory rather than in C:\Program Files. You'll need to search for it to find the correct pathname.

Installing App Inventor 2 Setup on Mac OS X

To get the Android emulator for your Mac, download and install the Setup Package. Click the blue link below to begin the download.

Download the installer from http://appinv.us/aisetup_mac

1. Double-click the downloaded file to start the installer. (You may need to look in your browser's downloads folder. The file is named AppInventor_Setup_v_X.X.dmg (where the X.X is the version number)
2. Click continue.

3. Read and accept the software license agreement.
4. On the Standard Install screen, click Install. Don't change the install location

5. If asked, enter your password to confirm that you really want to install software. Click OK.
6. The installer confirms that the App Inventor Setup package was installed.

7. If you are updating a previous version of the setup software, log out and log back in before continuing to use App Inventor

Installing App Inventor 2 Setup on GNU/Linux

You'll need sudo privileges to do the installation.

Note: The setup programs are 32-bit software. If you have a 64-bit system you may need to install libraries the let your machine run 32-bit software. One way to do this is to run the
command sudo apt-get install lib32z1, but this might not work on all GNU/Linux distributions, and you may need to do some investigation for your particular system.

If you have previously installed the App Inventor setup software, you should remove those files before installing the new software:

sudo rm -rf /usr/google/appinventor

sudo rm -rf ~/.appinventor

Instructions for systems that can install Debian packages

Use these instructions for systems that can install Debian packages
(e.g. Debian or Ubuntu):

Note: If you previously installed the setup package for App Inventor Classic, you should remove it, since it can interfere with the new installation. Remove the package with sudo apt-get remove appinventor-setup.

Download the http://appinv.us/aisetup_linux_deb . This is a file named appinventor2-setup_2.3_all.deb. It is a Debian package installer file. The place it will end up on your computer depends on how your browser is configured. Typically, it will go into your Downloads folder.

1. If your system can install packages simply by clicking on the package file, then do that.
2. If your system doesn't support clickable package installers, then navigate to the directory where the file is located and run the command

   sudo dpkg —install appinventor2-setup_2.3_all.deb

   With either method, you might need to ensure that the deb file as well as the directory it's in are world readable and world executable. On some systems, sudo does not have the default privileges to read and execute all files.
3. The software will be installed under /usr/google/appinventor.

You might also need to configure your system to detect your device. See the Android developer instructions at http://developer.android.com/tools/device.html#setting-up . Follow the instructions under the step "set up your system to detect your device" in the bullet under

“If you’re developing on Ubuntu Linux”.

Instructions for other GNU/Linux systems

Download the http://appinv.us/aisetup_linux_other . This is a file named appinventor2-setup_2.3.tar.gz. It is a Gzip compressed tar file.

1. Install the files using a method appropriate to your operating system. You’ll need to check that the commands-for-Appinventor directory ends up under/usr/google/appinventor.

Starting aiStarter

The aiStarter program manages communication between the Web browser and the Android device. It must be running whenever people use the emulator or the USB cable; it does not need to be running when people are using the wireless companion. Whenever someone logs in to use App Inventor with the emulator or USB, they will need to start aiStarter. This can be done with the command

/usr/google/appinventor/commands-for-Appinventor/aiStarter &

For convenience, you might want to arrange for this command to be run automatically whenever someone logs in, or when the system starts. The precise way to do this depends on which GNU/Linux distribution you are using. Consult the documentation for your distribution.

Locating the Setup directory

In most cases, App Inventor should be able to locate the installed Setup software on its own. If it does ask you where the software is located, the directory path you should enter is

/usr/google/appinventor/commands-for-Appinventor

Step 2. Launch aiStarter

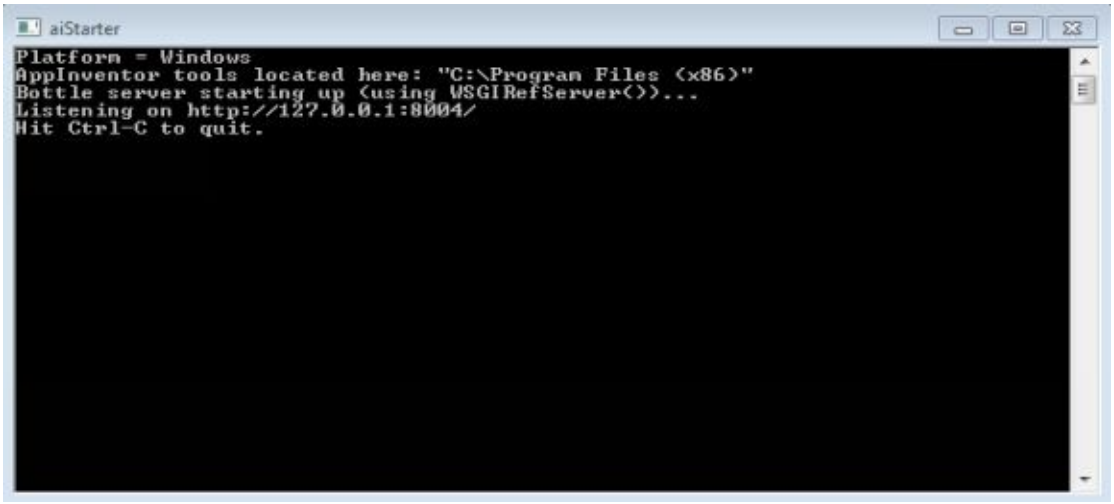
Using the emulator or the USB cable requires the use of a program named *aiStarter*. This program is the helper that permits the browser to communicate with the emulator or USB cable. The aiStarter program was installed when you installed the App Inventor Setup package. You do not need aiStarter if you are using only the wireless companion.

- On a Mac, aiStarter will start automatically when you log in to your account and it will run invisibly in the background.

- On Windows, there will be shortcuts to aiStarter from your Desktop, from the Start menu, from All Programs and from Startup Folder. If you want to use the emulator with App Inventor, you will need to manually launch aiStarter on your computer when you log in. You can start aiStarter this by clicking the icon on your desktop or using the entry in your start menu.

The aiStarter Icon on Windows

To launch aiStarter on Windows, double click on the icon (shown above). You'll know that you've successfully launched aiStarter when you see a window like the following:

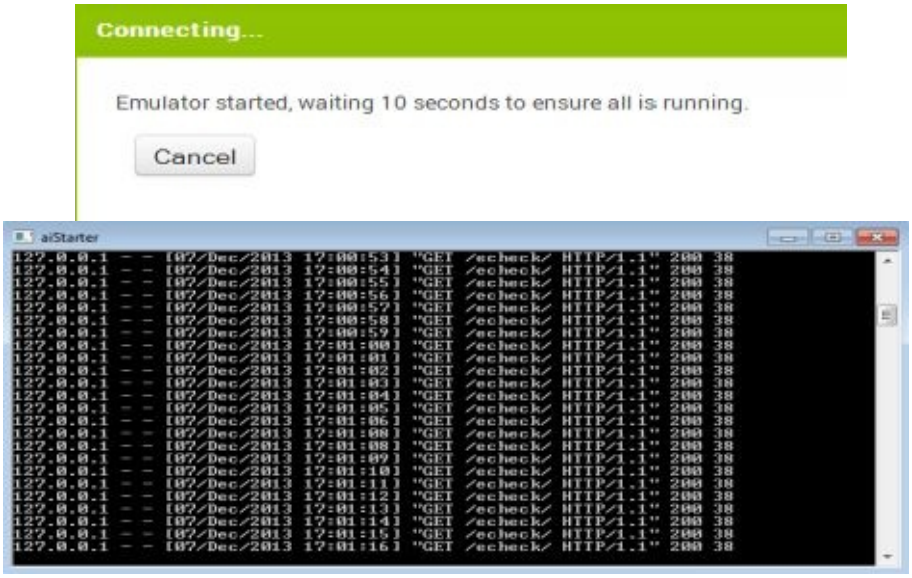


Step 3. Open an App Inventor project and connect it to the emulator

First, go to App Inventor and open a project (or create a new one — use *Project* > *Start New Project* and give your project a name).Then, from App Inventor's menu (on the App Inventor cloud-based software at ai2.appinventor.mit.edu), go to the Connect Menu and click the Emulator option.

You'll get a notice saying that the emulator is connecting. Starting the emulator can take a

couple of minutes. You may see update screens like the following as the emulator starts up:



The emulator will initially appear with an empty black screen (#1). Wait until the emulator is ready, with a colored screen background (#2). Even after the background appears, you should wait until the emulated phone has finished preparing its SD card: there will be a notice at the top of the phone screen while the card is being prepared. When connected, the emulator will launch and show the app you have open in App Inventor.

> If this is the first time you are using the emulator after installing the App Inventor Setup software, you will see a message asking you to update the emulator. Follow the directions on the screen to perform the update and reconnect the emulator.
> You will need to do this kind of update whenever there is a new version of the App Inventor software.

Reference

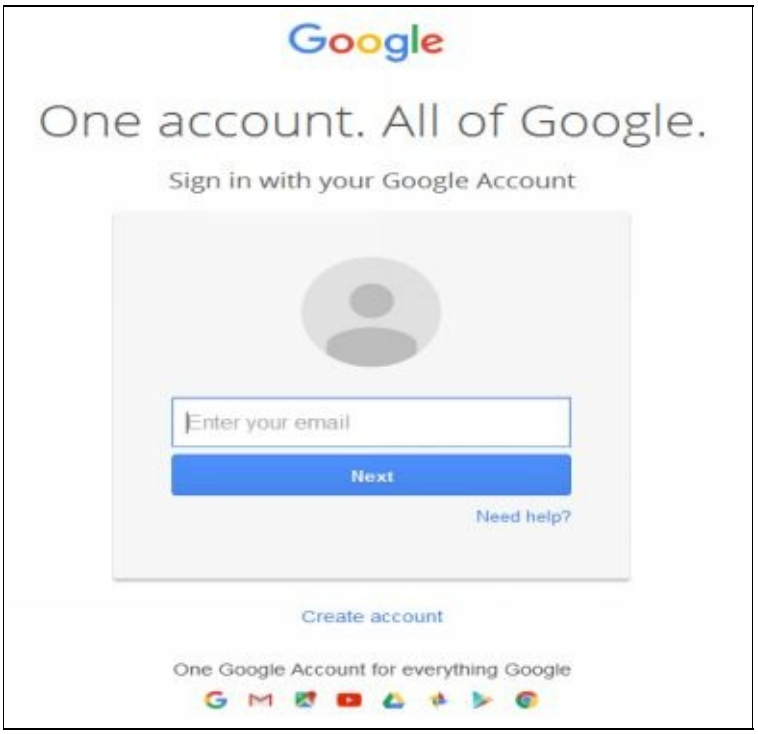
- http://appinventor.mit.edu

Chapter 2: Text-to-Speech App

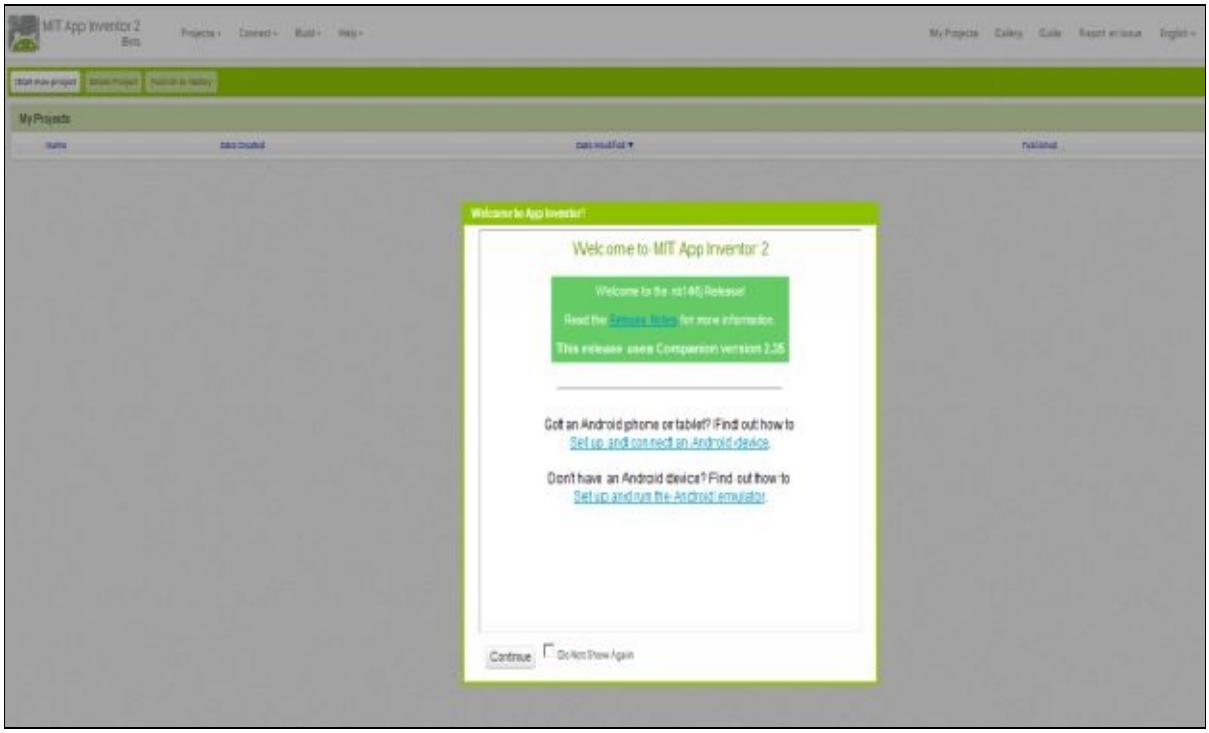
Resources:-All the media files/other objects used in this book can be downloaded to your local computer by just opening this link in your browser :-https://docs.google.com/uc?id=0B2muK1BFOy2haGlRdE1VdElZVUU&export=download

1. Connect to http://ai2.appinventor.mit.edu/ and Login to your Gmail account. If you

don’t have one then create a gmail account first.



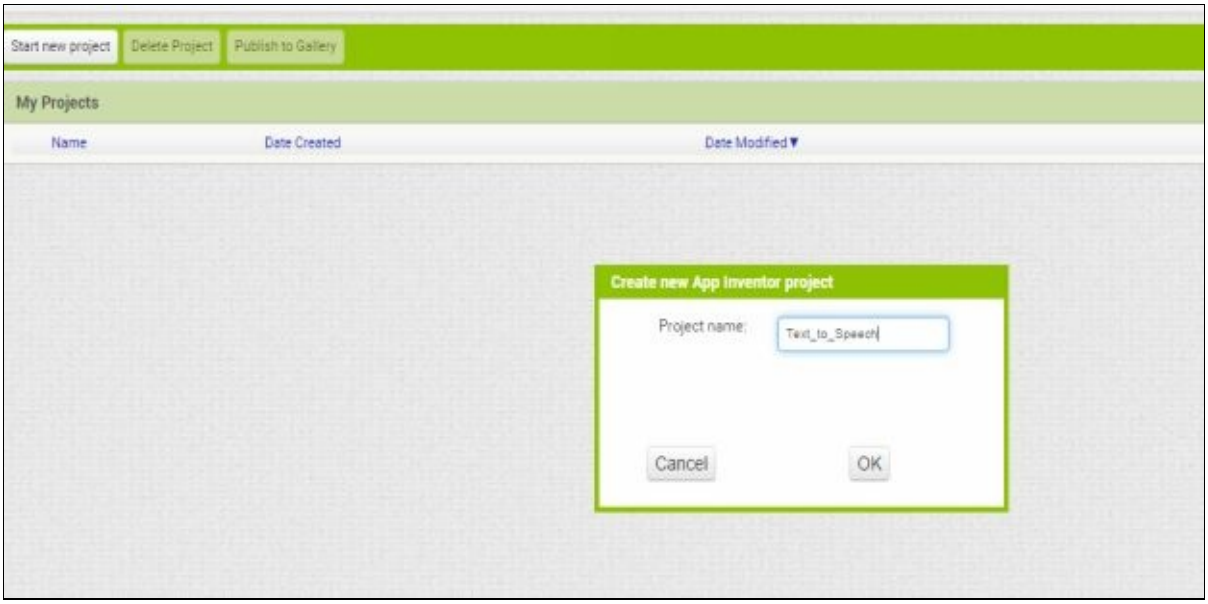
2. When you login to App Inventor you will see the below screen. Click Continue.



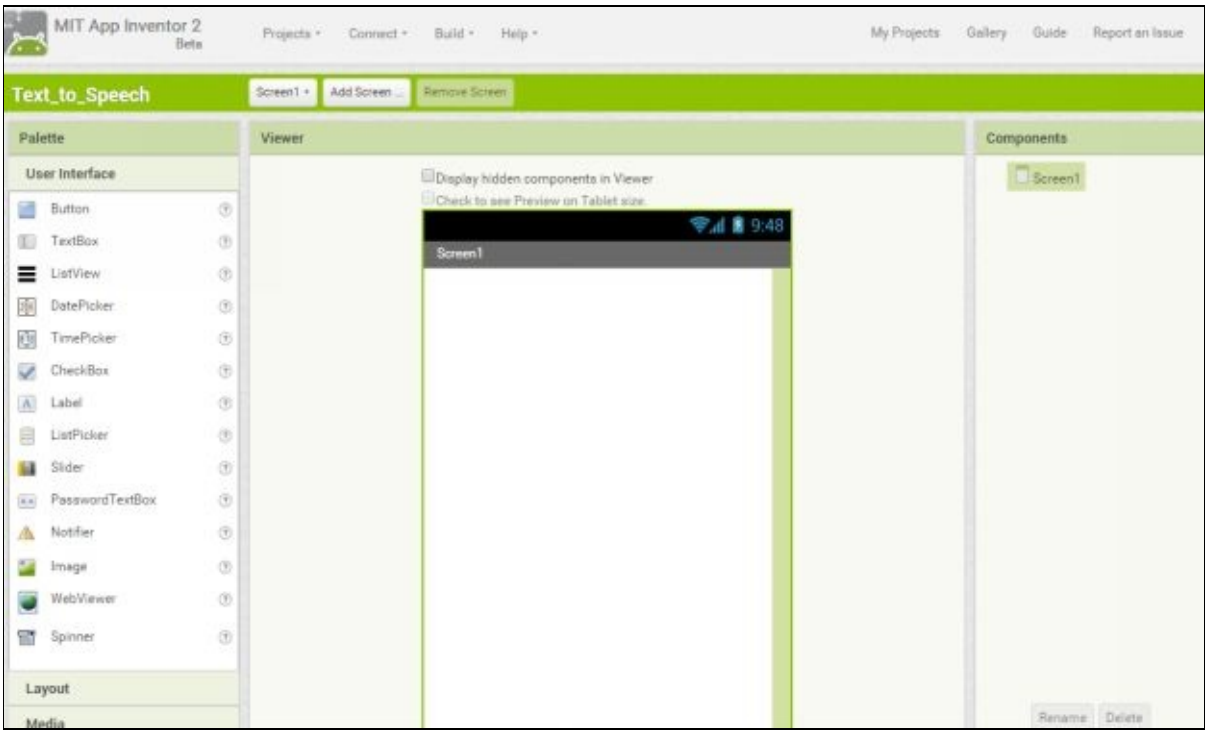
3. When you login for the first time you will not be able to see any Projects. So it will

show Blank Screen.

4. Click on Start new project and give it name Text_to_Speech. And click Ok.

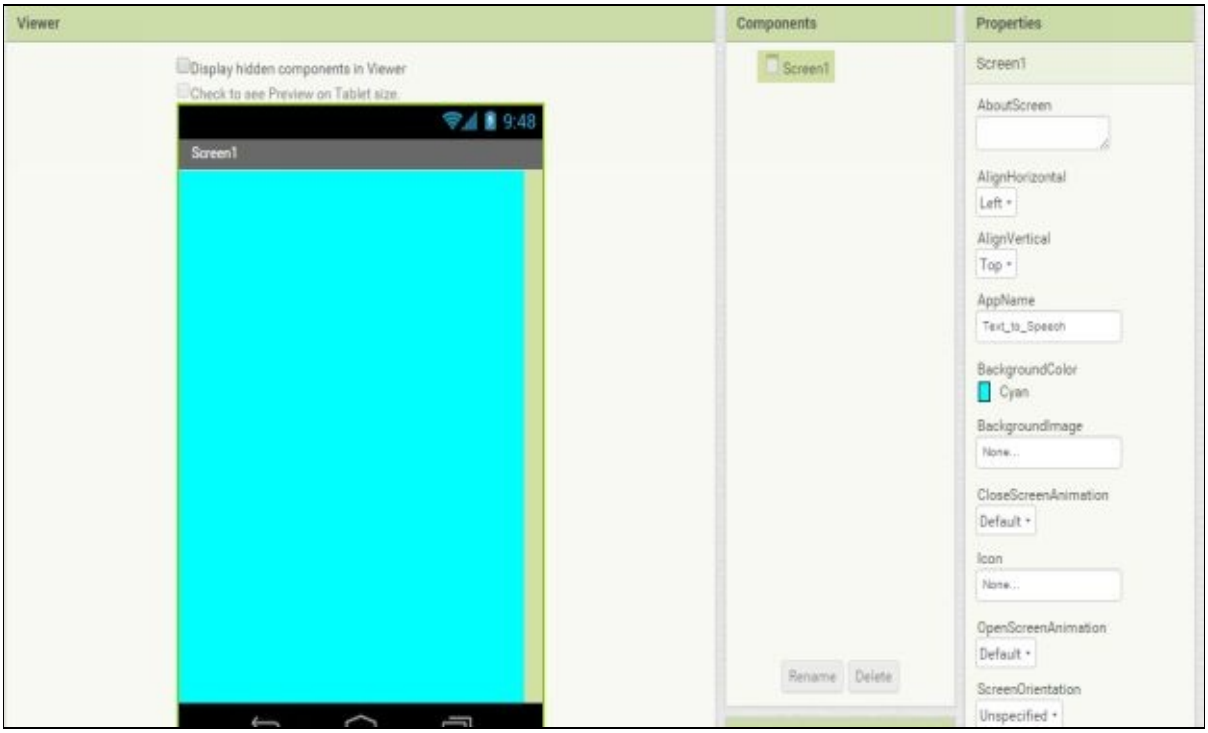


5. When you click OK you will see below screen.

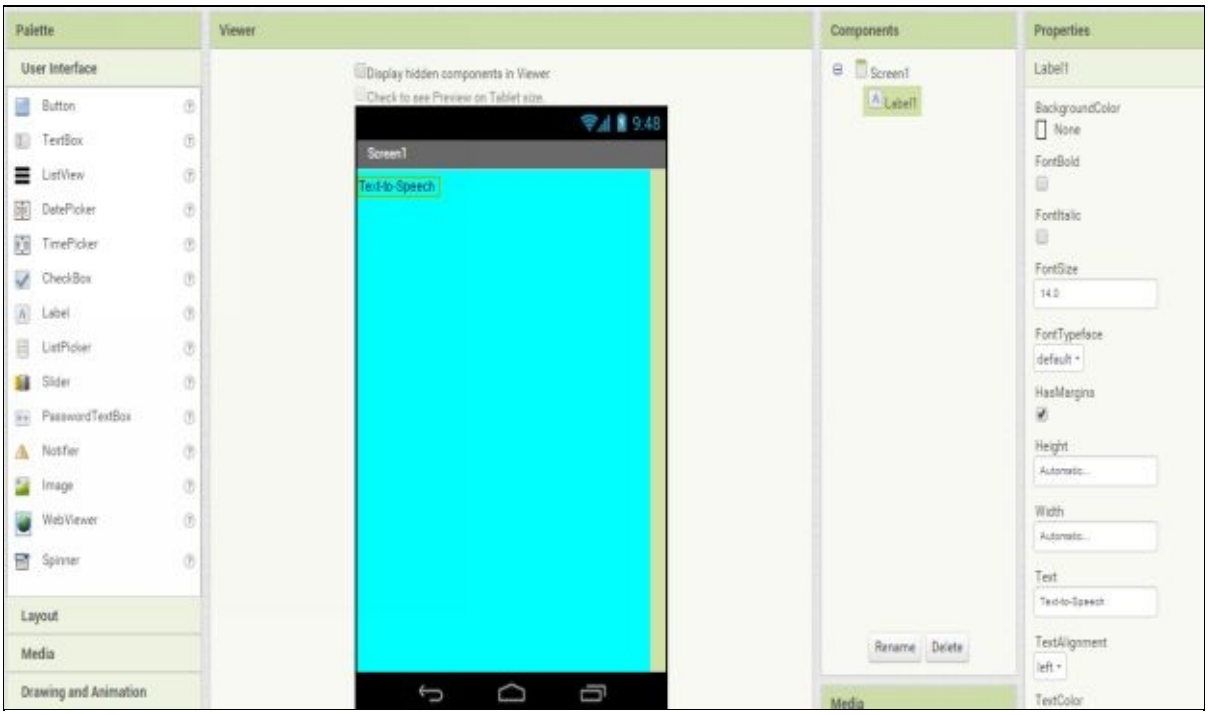


6. Now perform below actions:

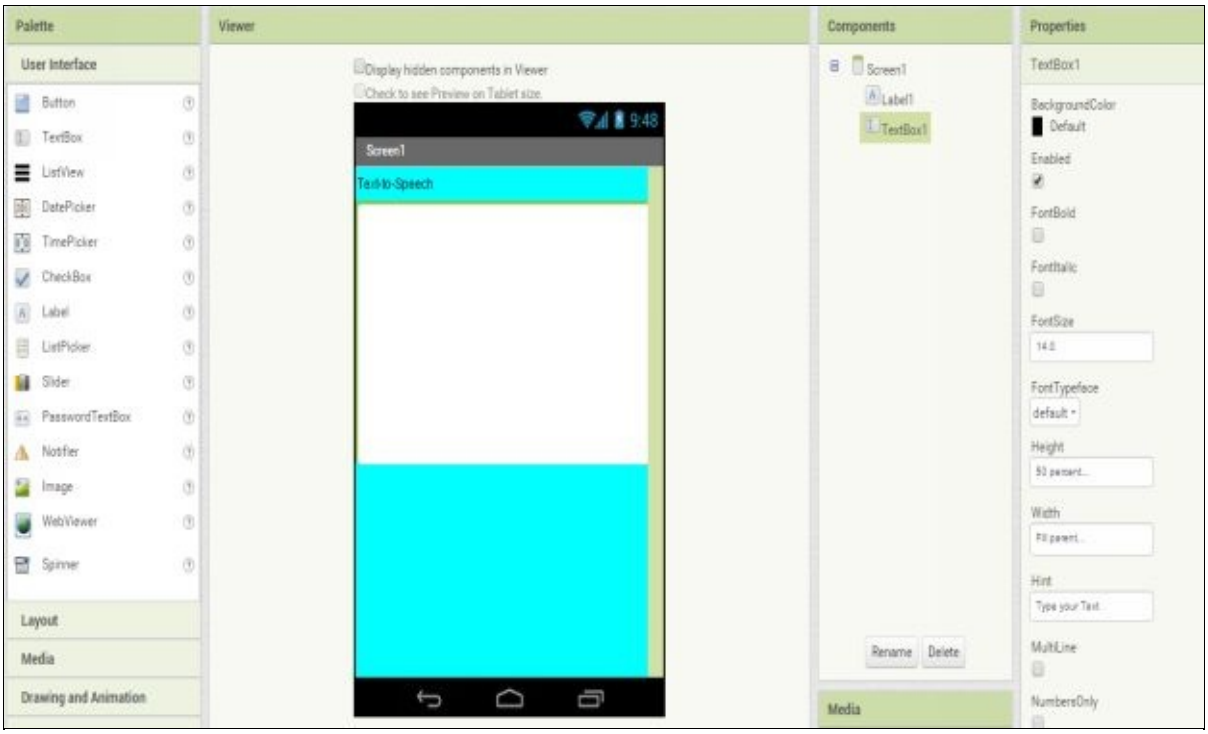
1. Select Screen1 in Components and change its BackgroundColor.



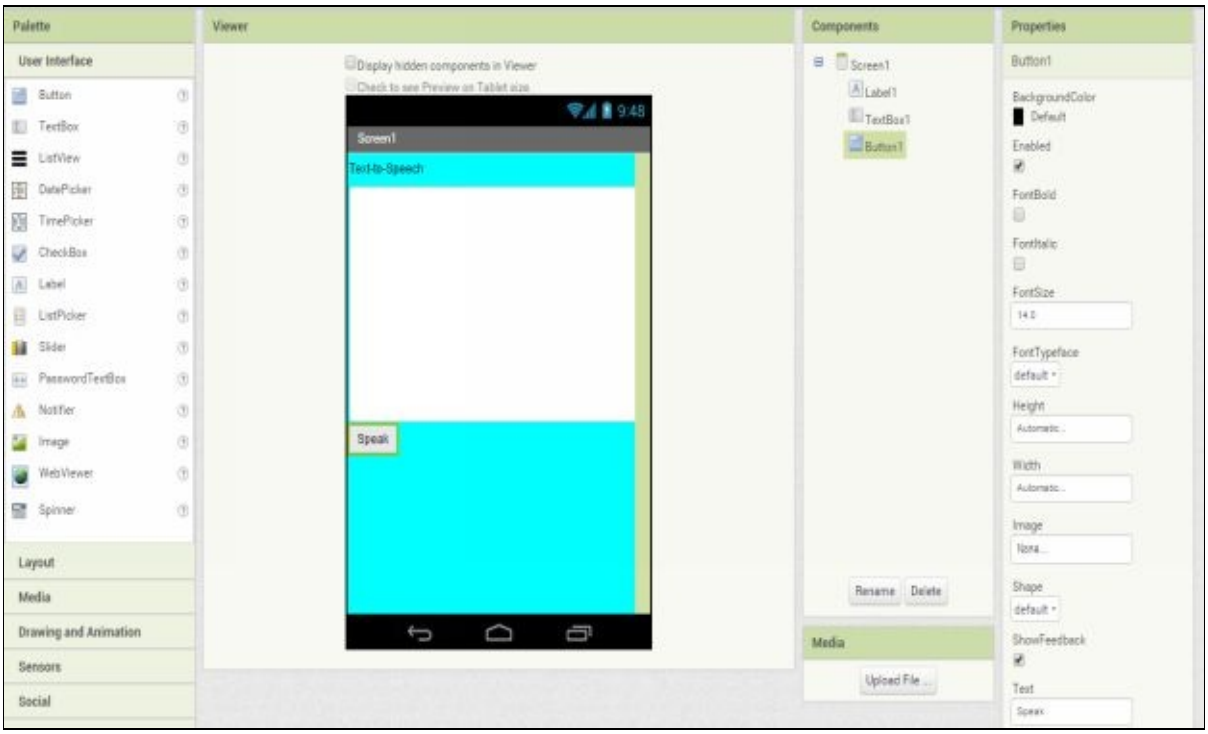
2. Drag a Label from Palette to Screen1 and change its Text to Text-to-Speech.



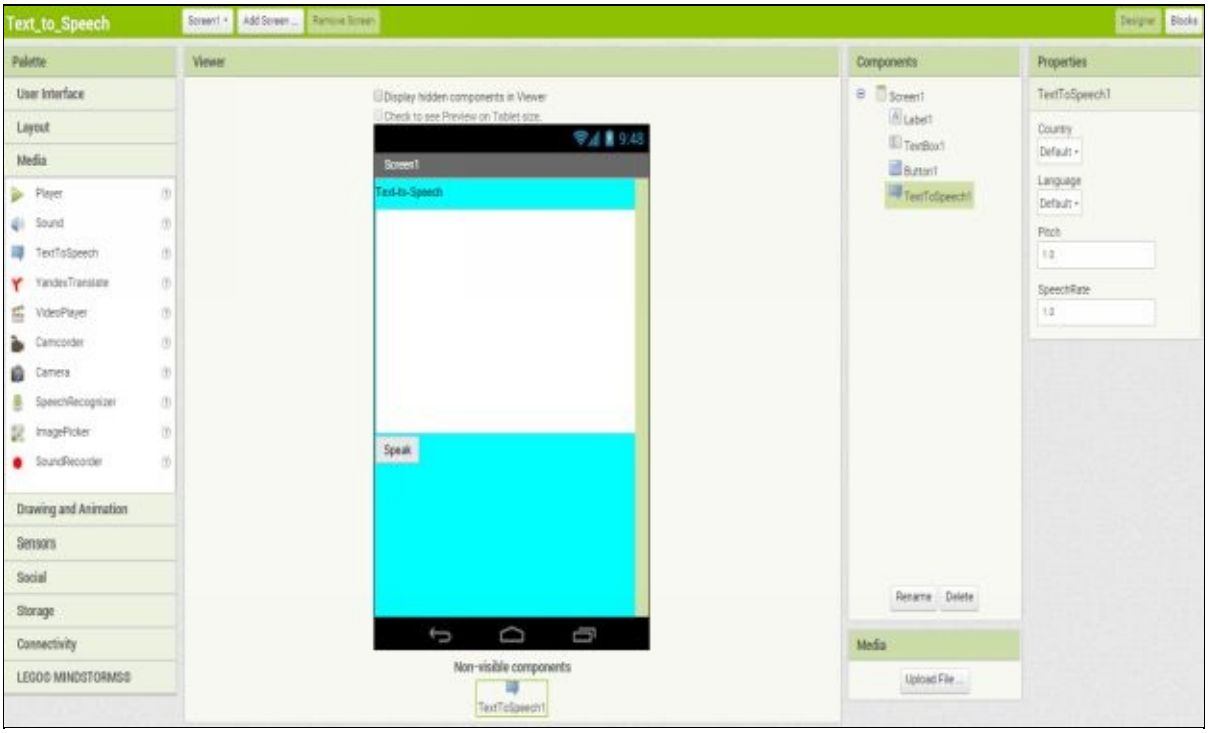
3. Drag a TextBox from Palette to Screen1 and set its Height to 50 percent and width to Fill Parent.



4. Drag a button to Screen1 and change its Text to Speak.

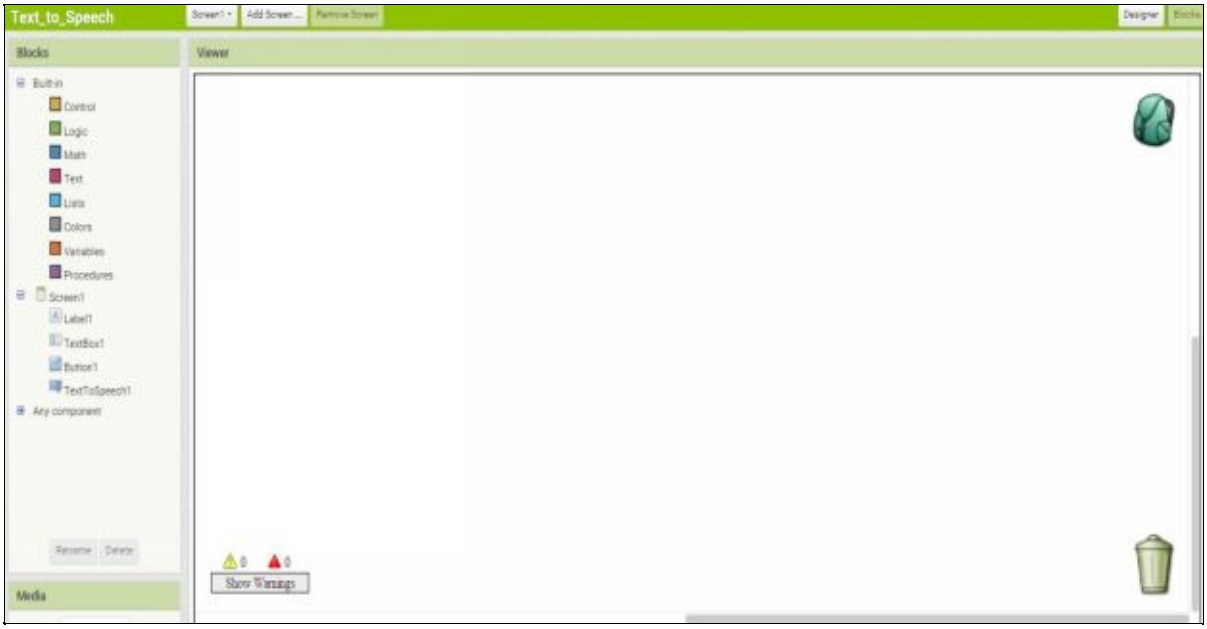


5. Now go to Media under Palette and Drag TextToSpeech component to Screen1.It is a Non-visible component so it will be shown below screen as Non-visible component.

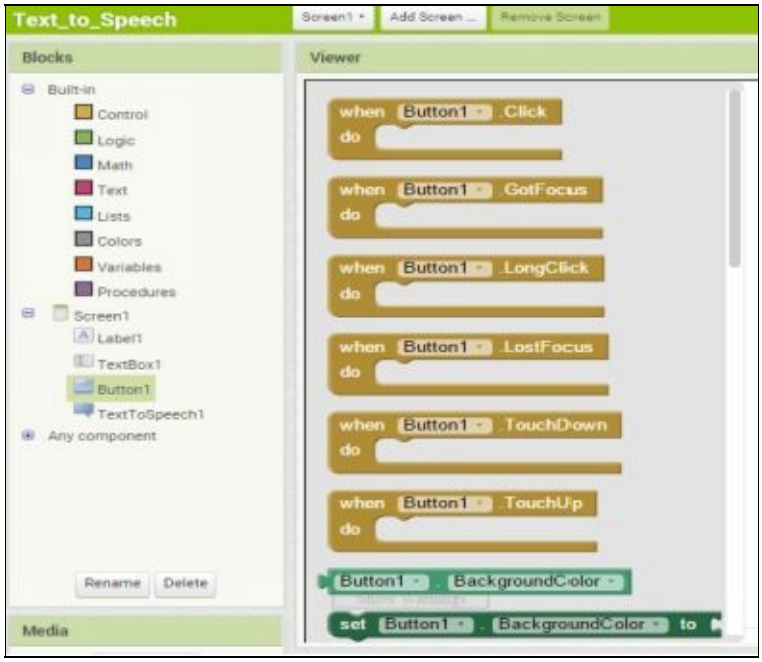


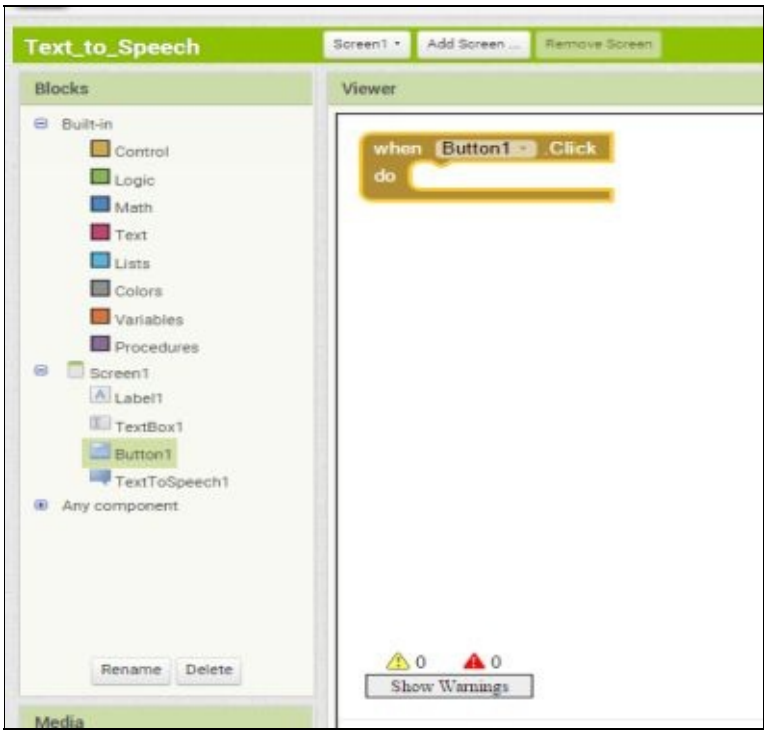
6. Now click on Blocks button on top right side to create logic for your app.

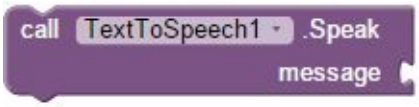
7. You will be landed to below screen. This is Block editor.

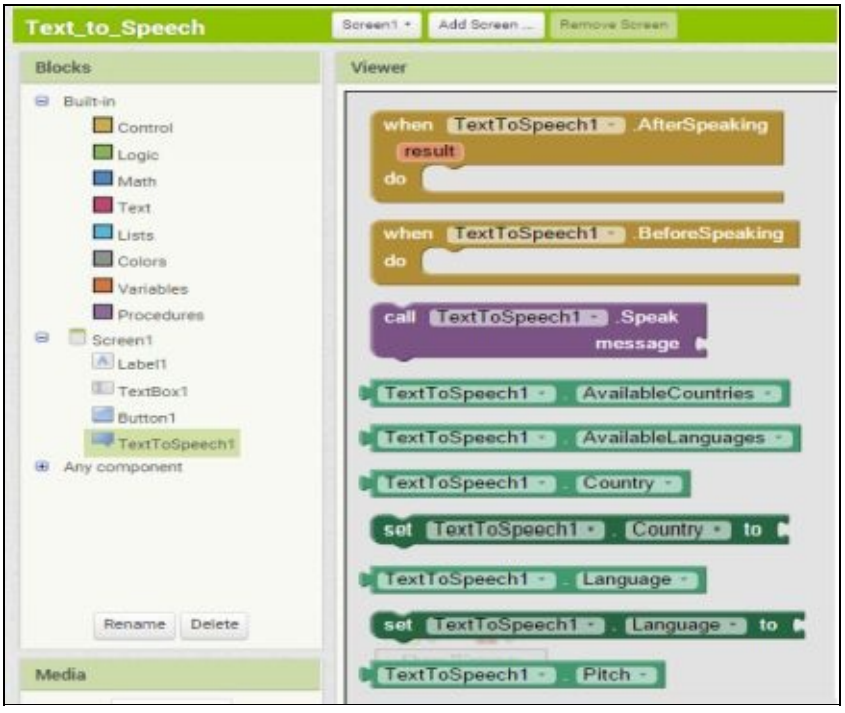


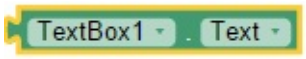
8. Click on Button1 on left side you will see a number of associated blocks with this component. Click on  block. This is an event block which will decide what your app will do when this button will be clicked. Here click is an event for Button1 component.



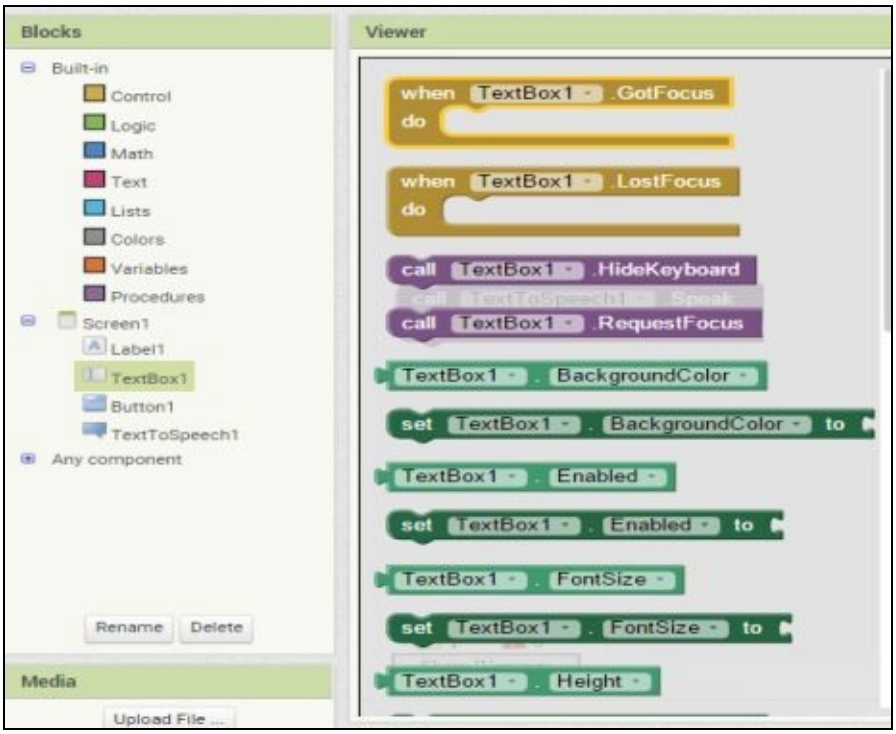


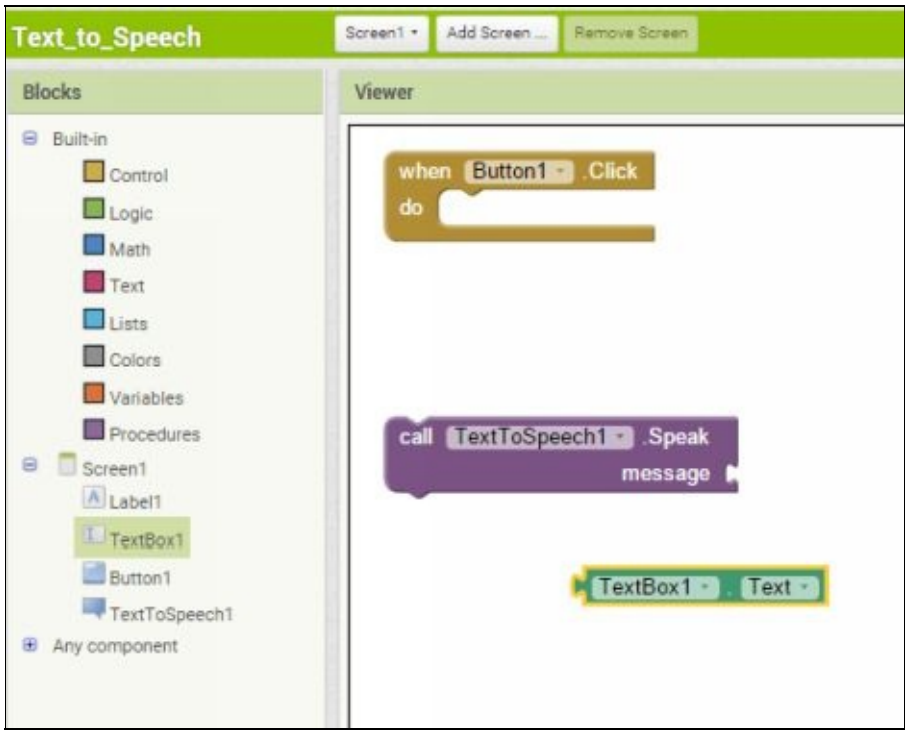
10. Click on TextToSpeech1 component on left hand side. You will see a number of associated blocks with this component. Click on  block. This block will speak some message which will be attached to this block late.

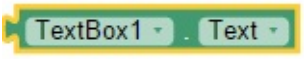


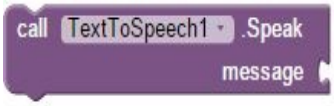
11. Now click on TextBox1 on left hand side. You will see all the associated blocks with this component. Scroll down and find 

block. This block contains the text (Message) which is entered in TextBox1.

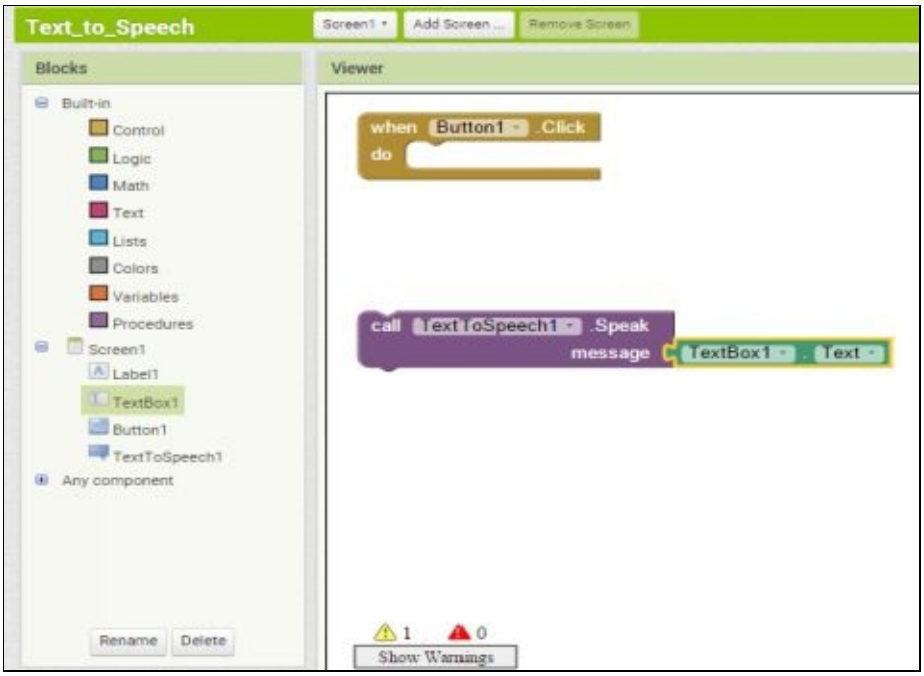




12. Now click on 

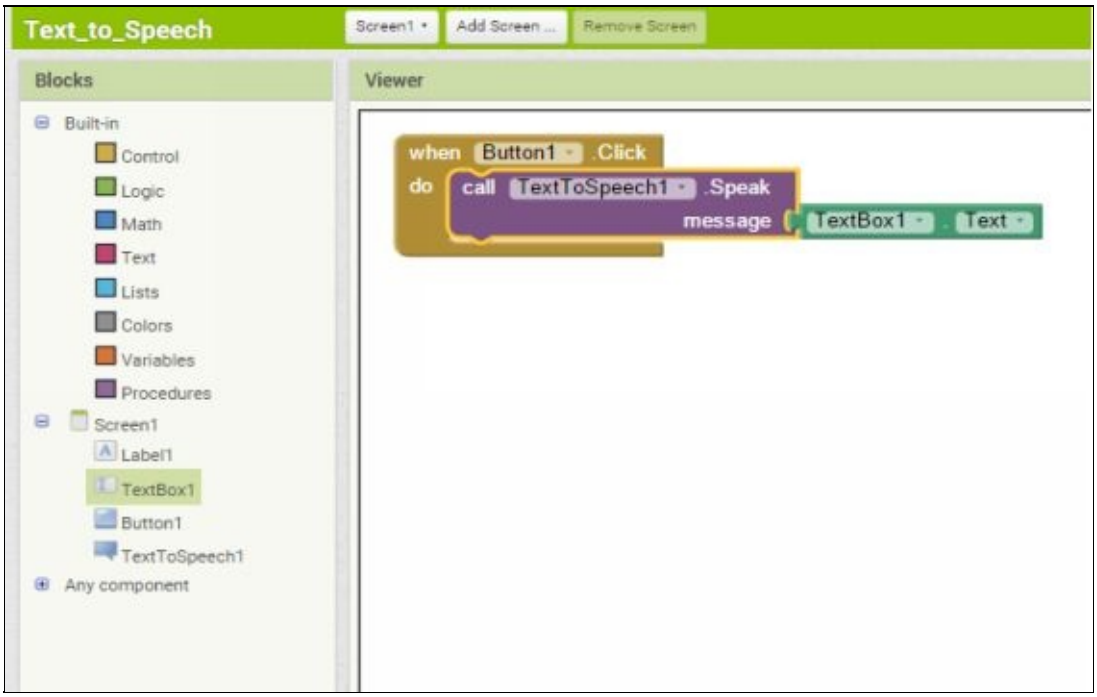
block and drag it to connect to 

block as shown below. This will tell our app to speak the text which is there in TextBox1.

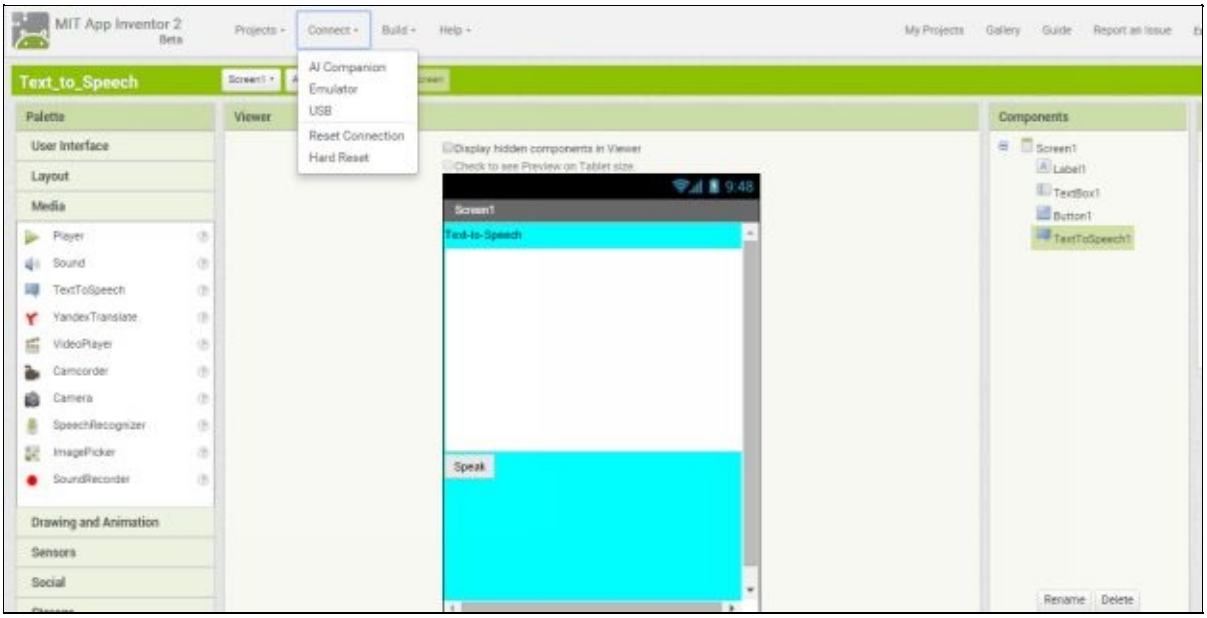


Now connect both the blocks to 

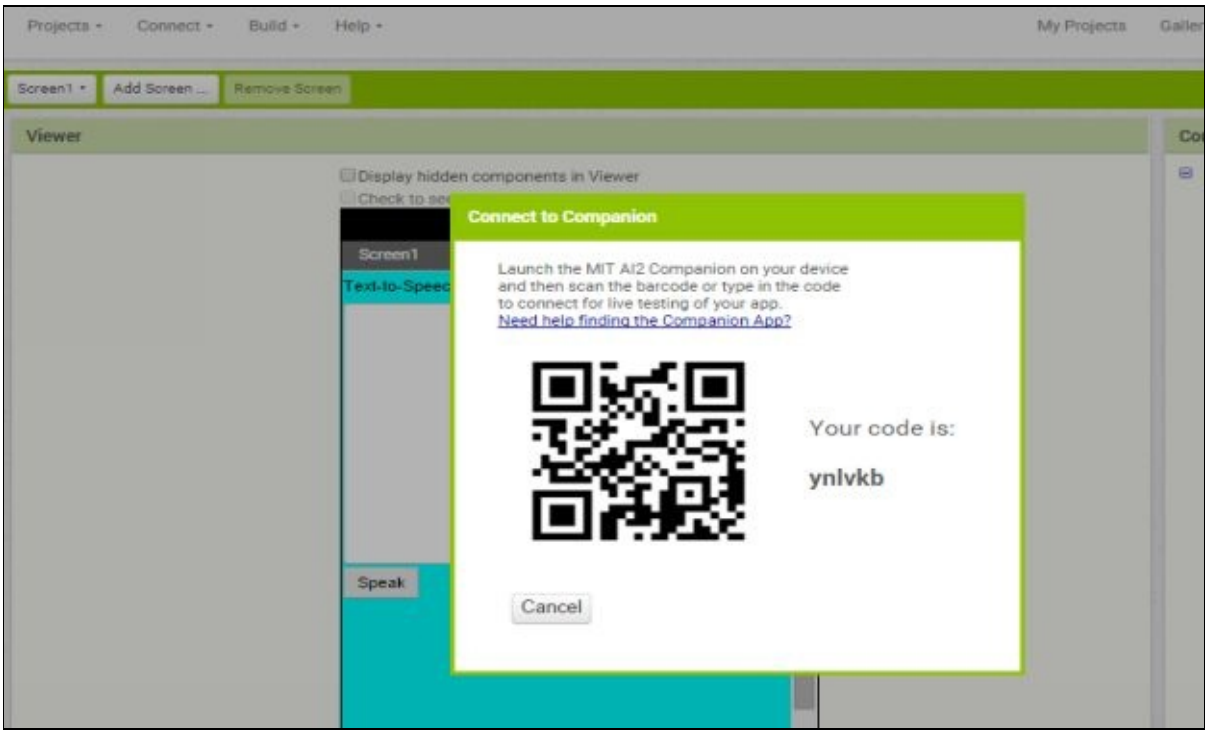
block. So that when Button1 is clicked your app will speak the text/message which is entered in TextBox1.



13. Now go to Designer and Change Screen1 Title to Text-to-Speech and click on Connect and select AI Companion.



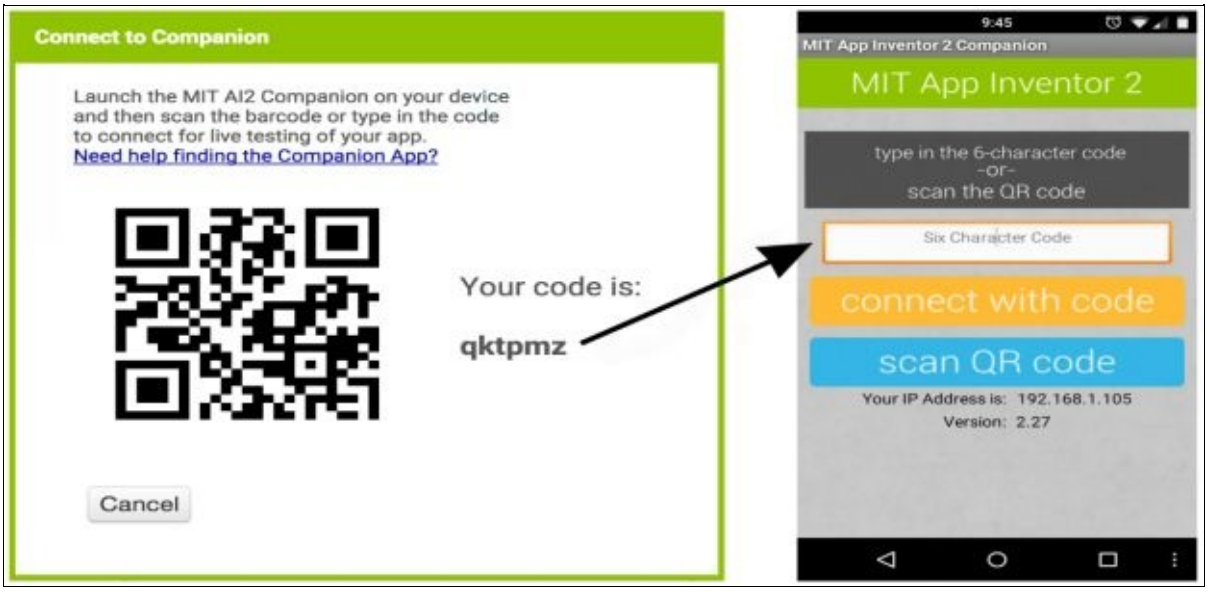
14. You will see something like this.



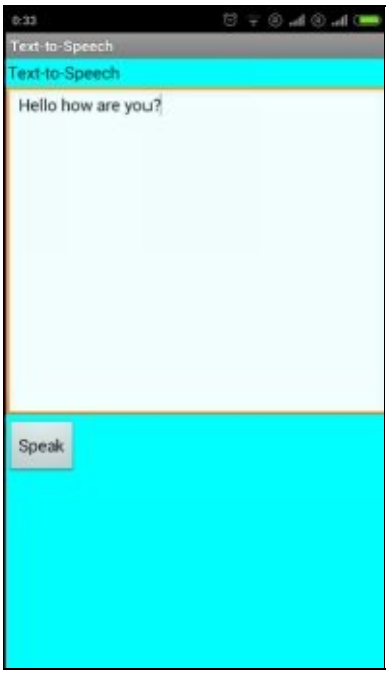
15. Now open MIT App Inventor 2 Companion in your mobile/Tablet.

And enter the code or scan the QR code from your mobile and click on connect with code.

Note:-Your Mobile/Tablet should be connected to same wi-fi /network to which your computer/Laptop is connected.



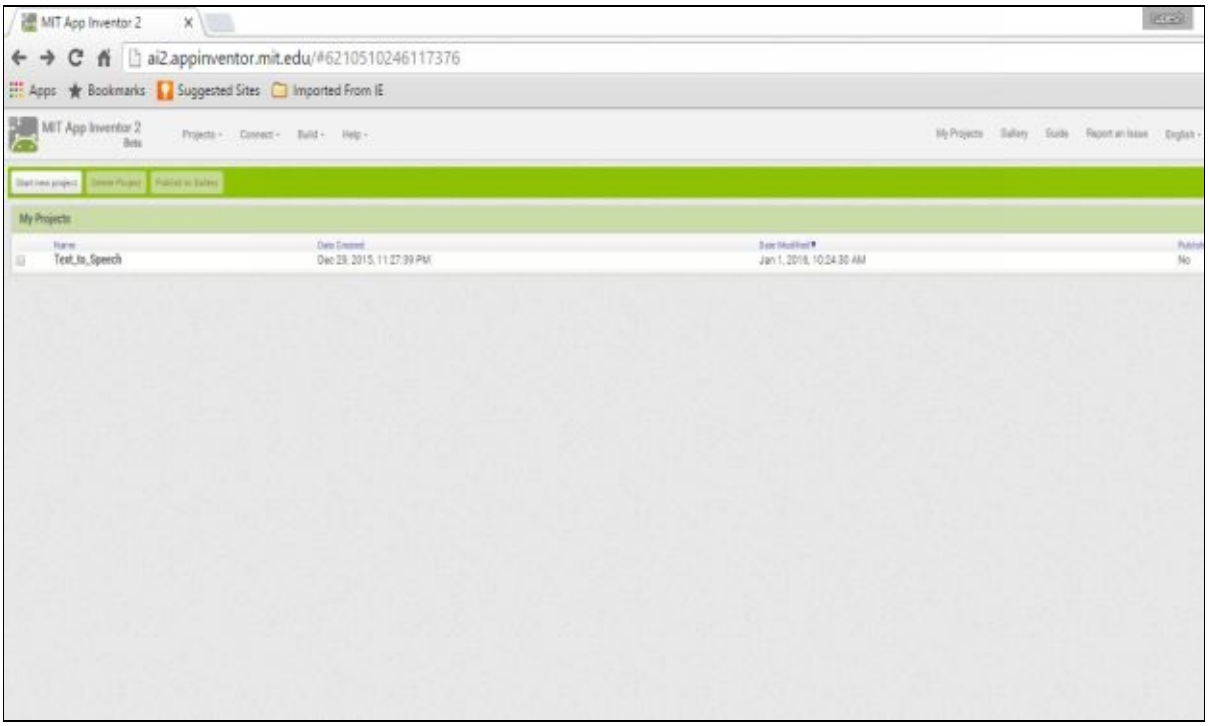
16. Now you should see your app in your phone for live testing. Type Any Text in the TextBox and Press Speak Button. Your Mobile should speak the text that you have typed in the TextBox.



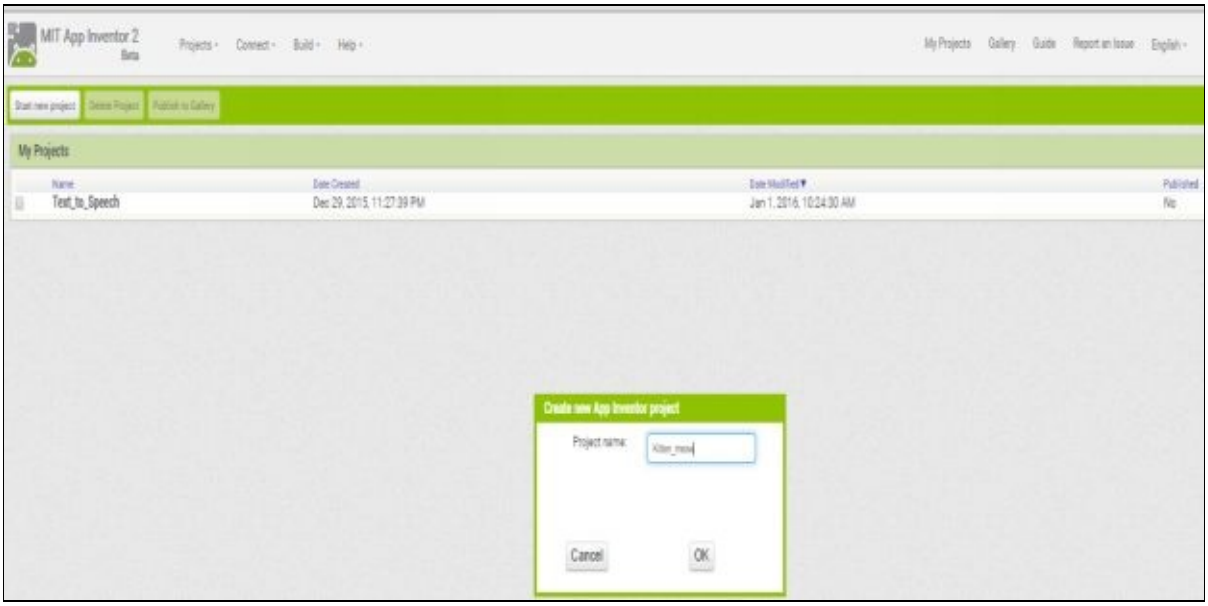
Congratulations you have made your first app.

Chapter 3: Kitten Meow App

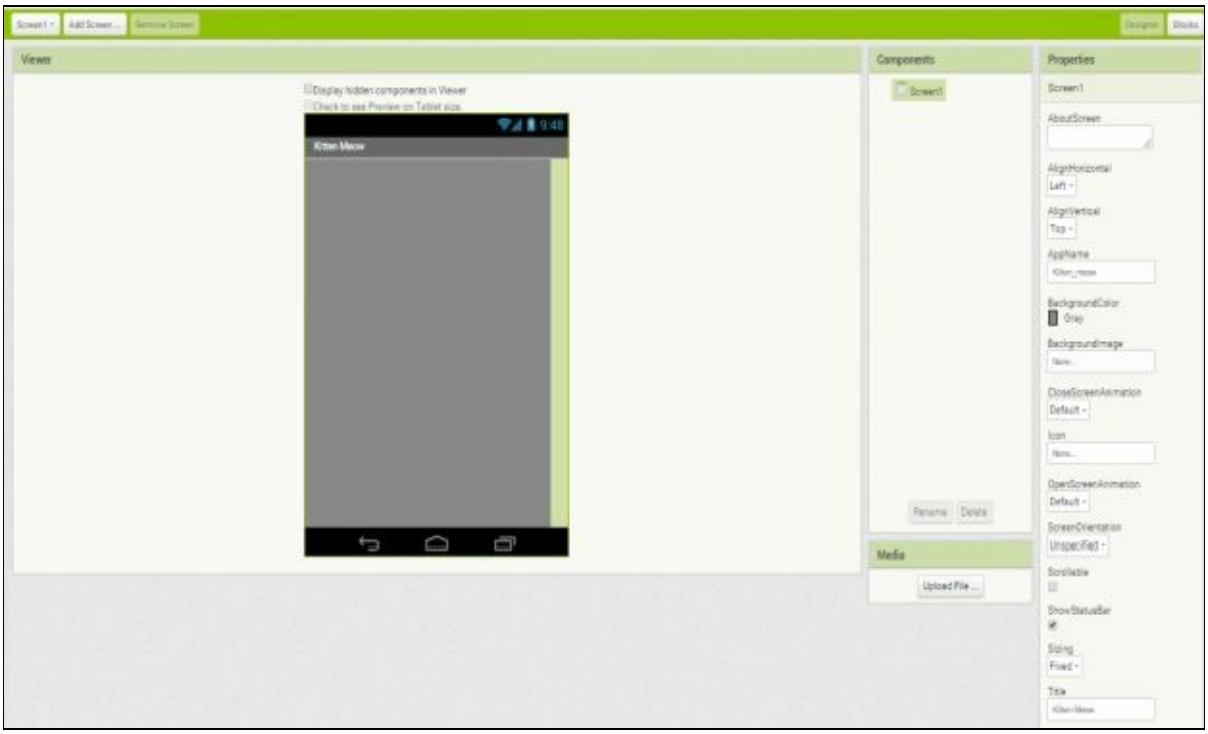
1. Login to App Inventor http://ai2.appinventor.mit.edu/ .



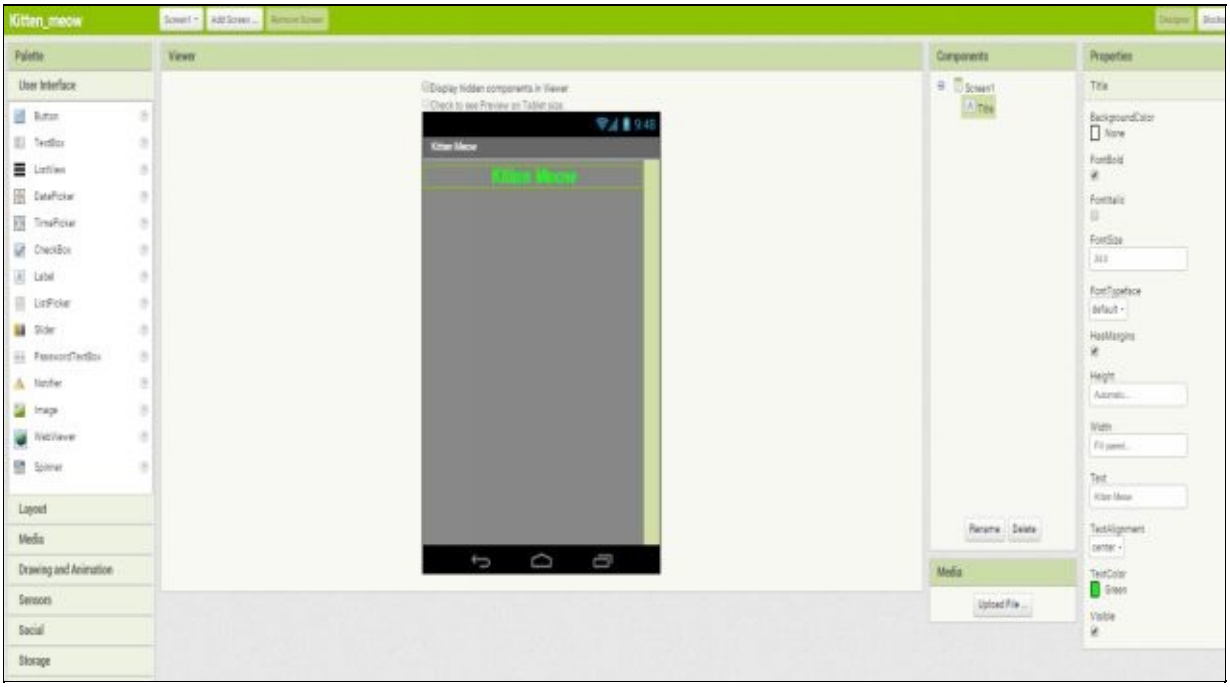
2. Click on Start new project and give it name Kitten_meow and click OK.



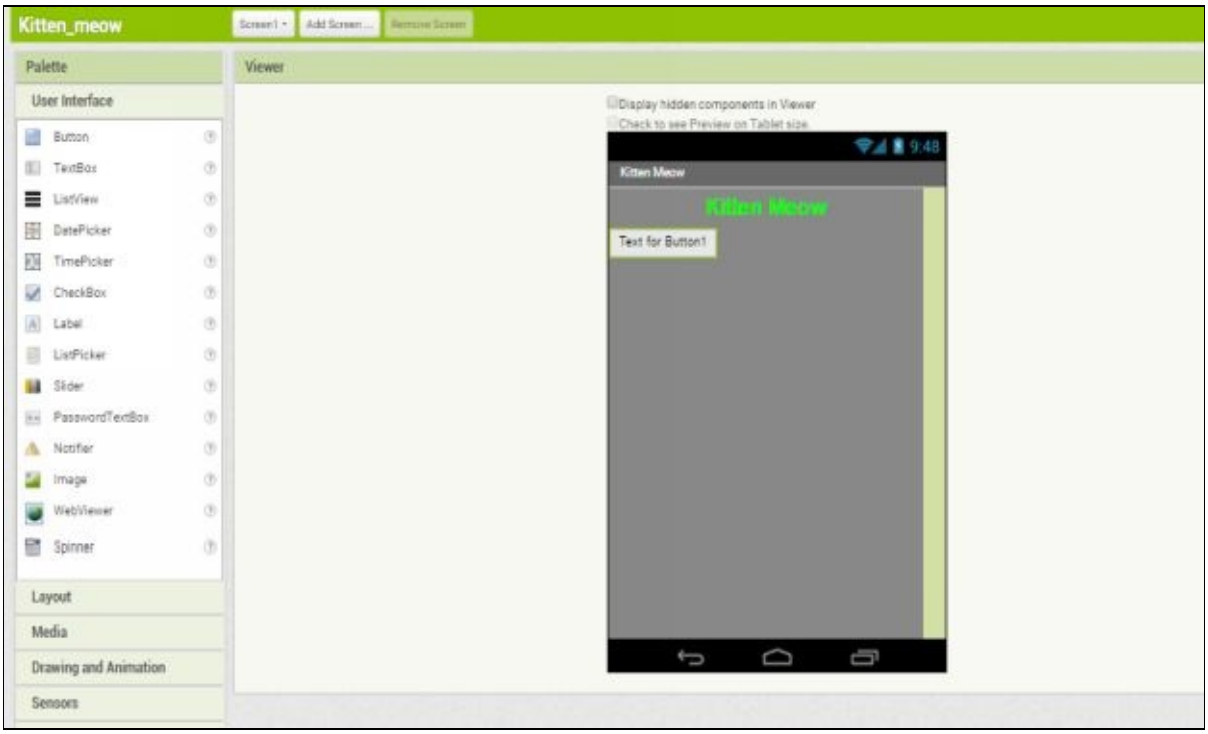
3. Change Screen1 Title to Kitten Meow and BackgroundColor to Grey color.



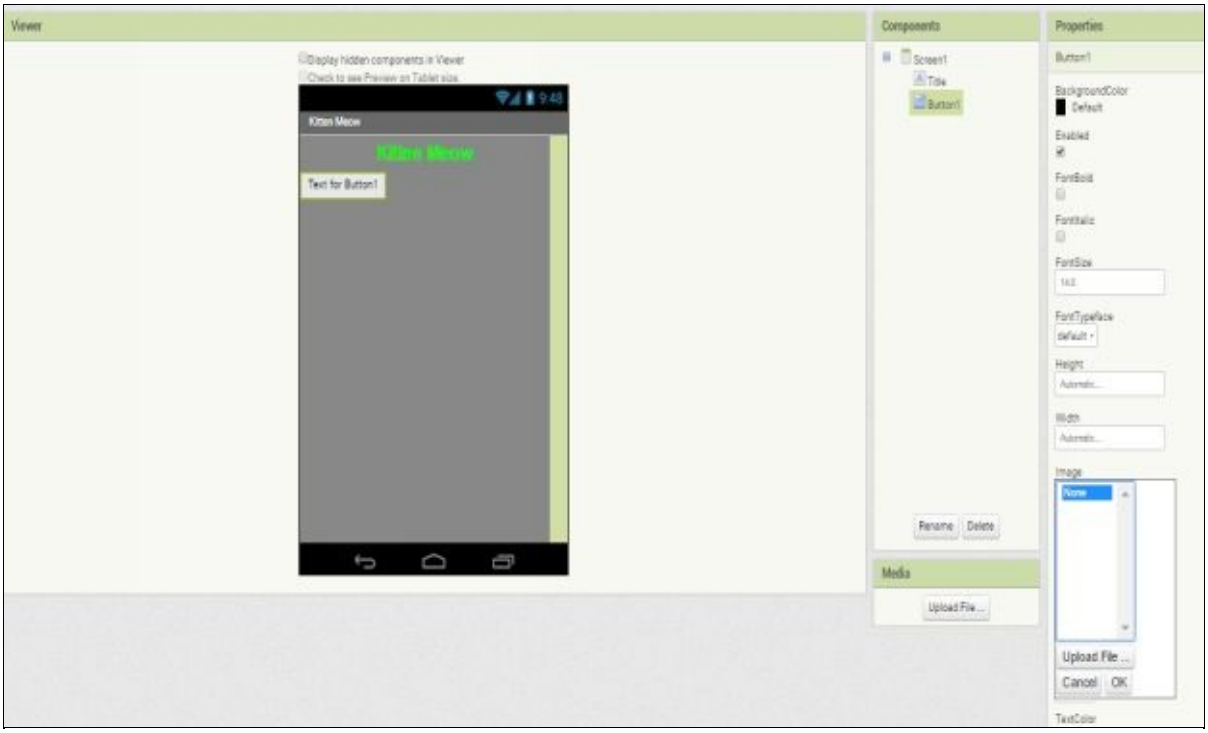
4. Drag a Label from Palette to Viewer screen and change its Text and other properties as shown below



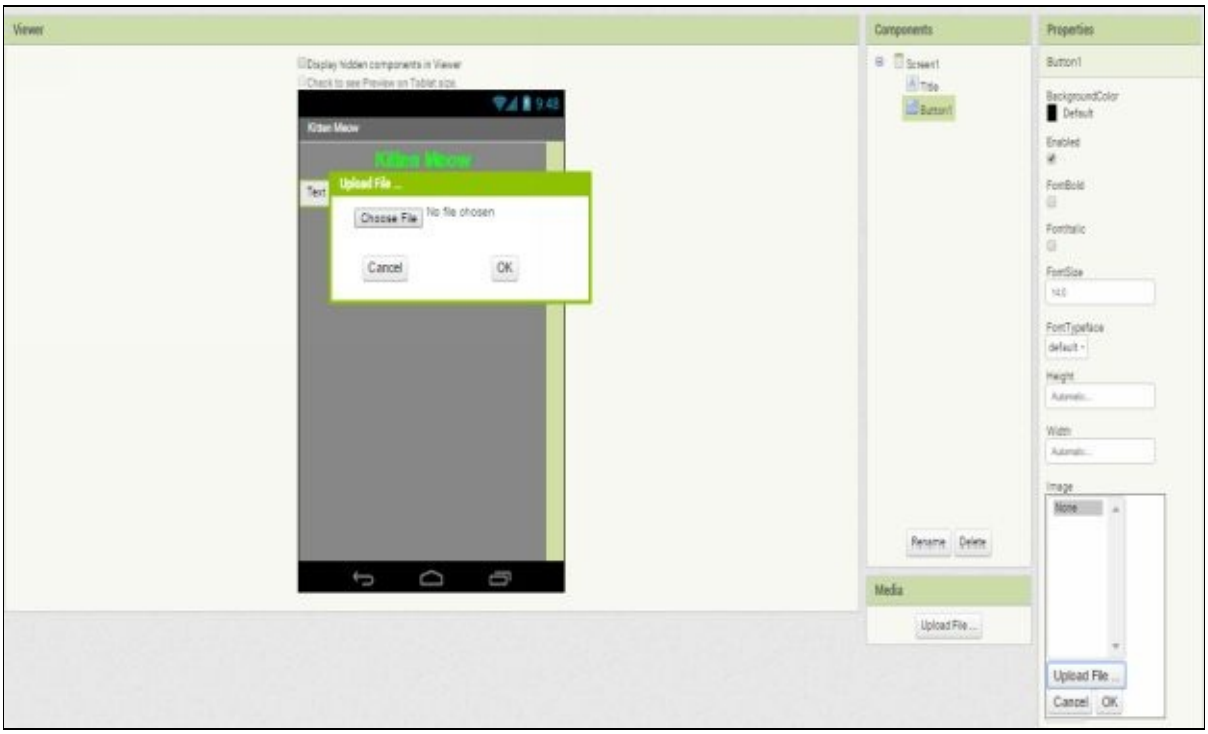
5. Now drag a Button from Palette to Viewer screen.



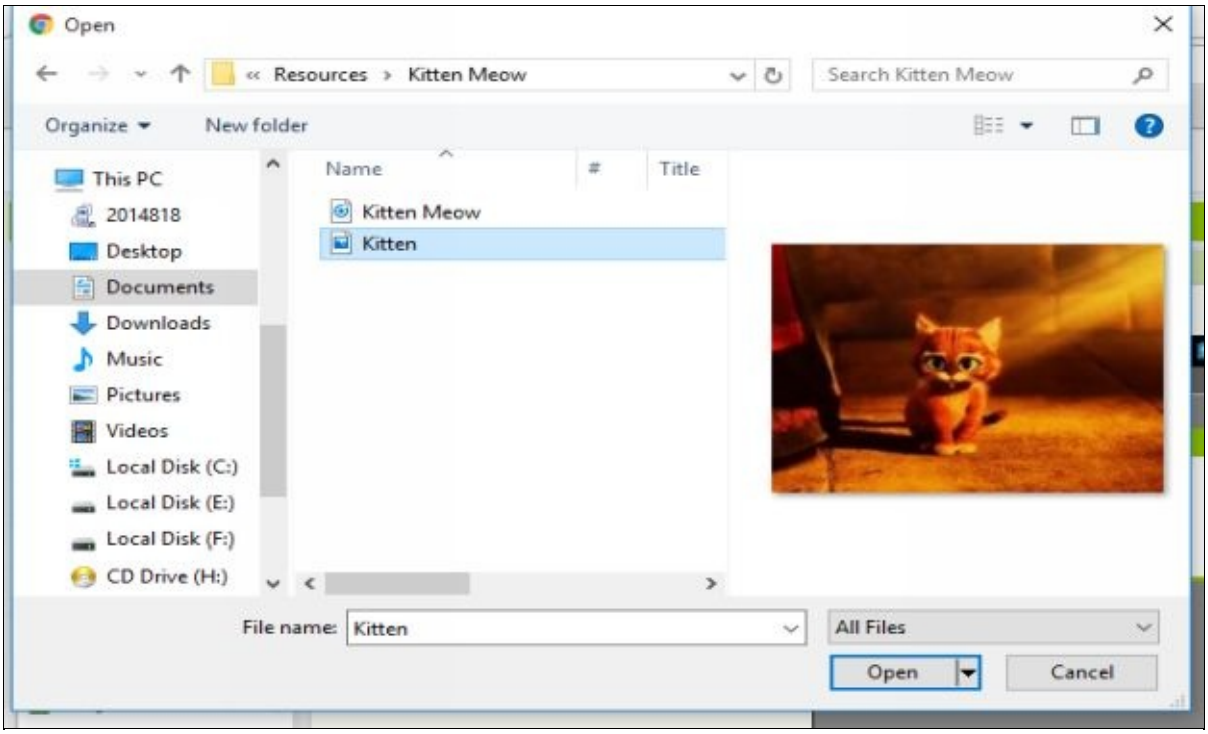
6. Now change Button Image to Kitten.jpg under Properties.



Click on Upload file.

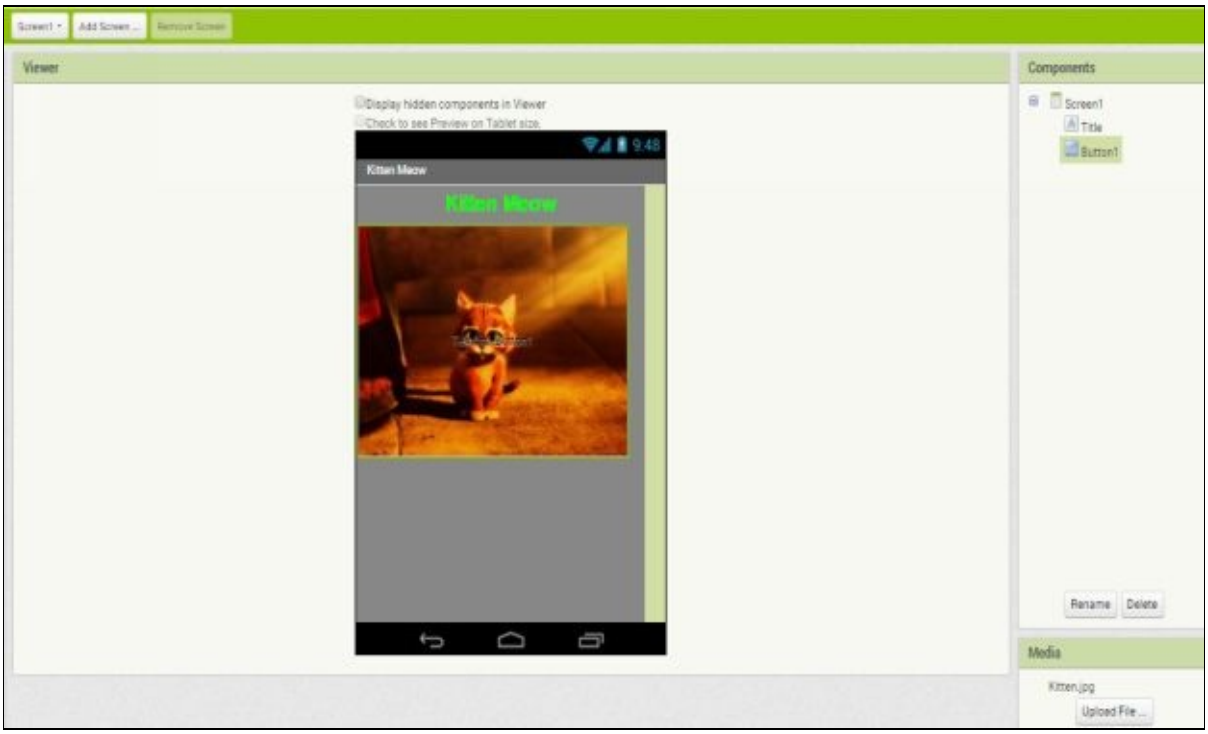


Click on Choose File and select Kitten.jpg image and click Open.

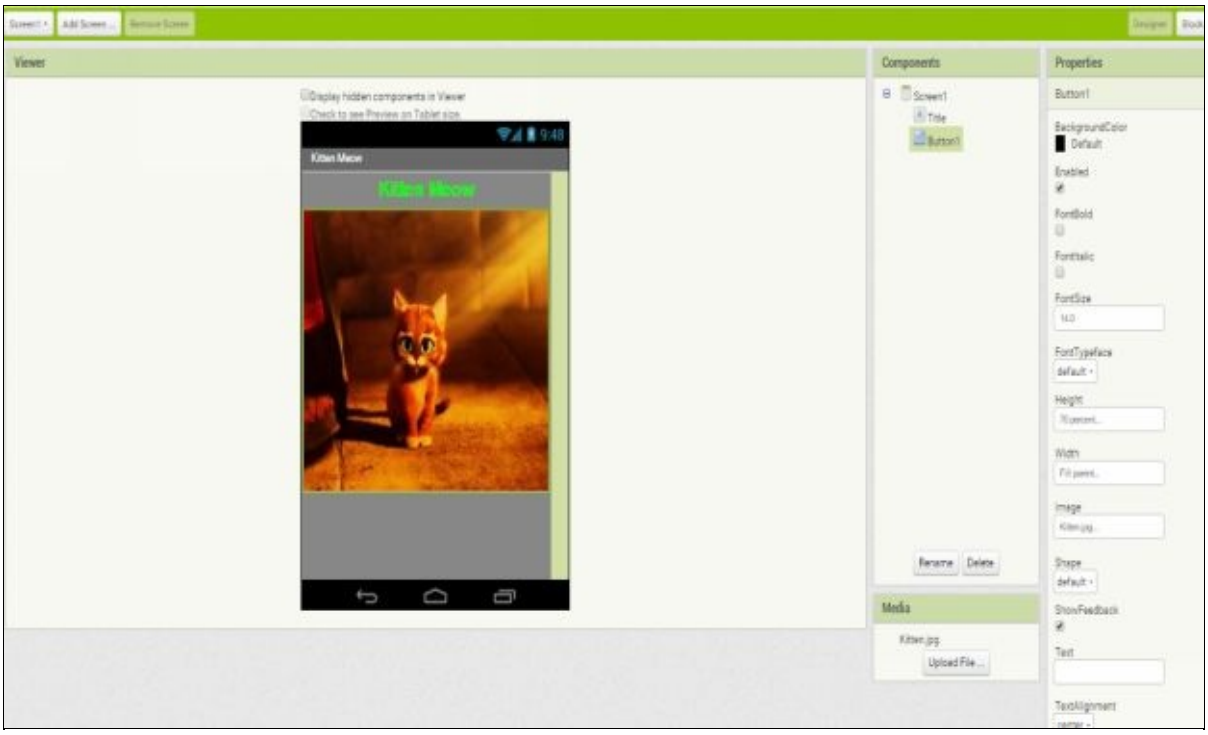


Click OK.

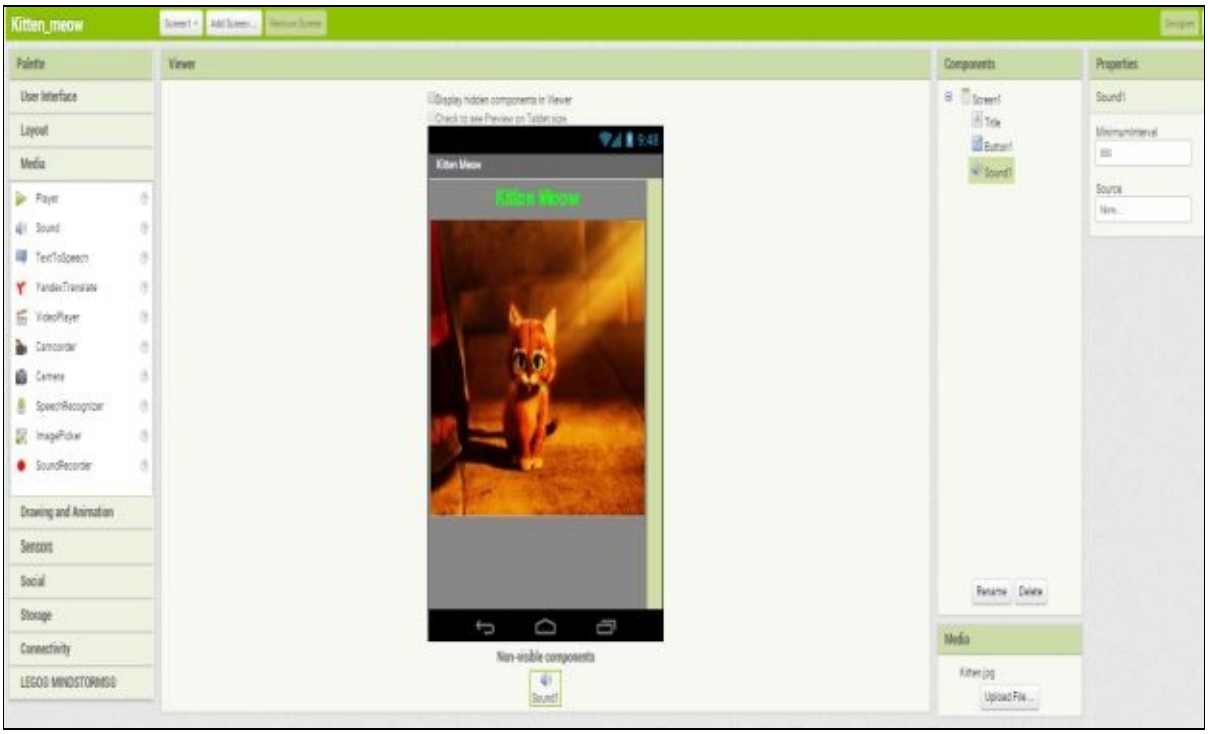
7. Now Image will be shown in your viewer screen.



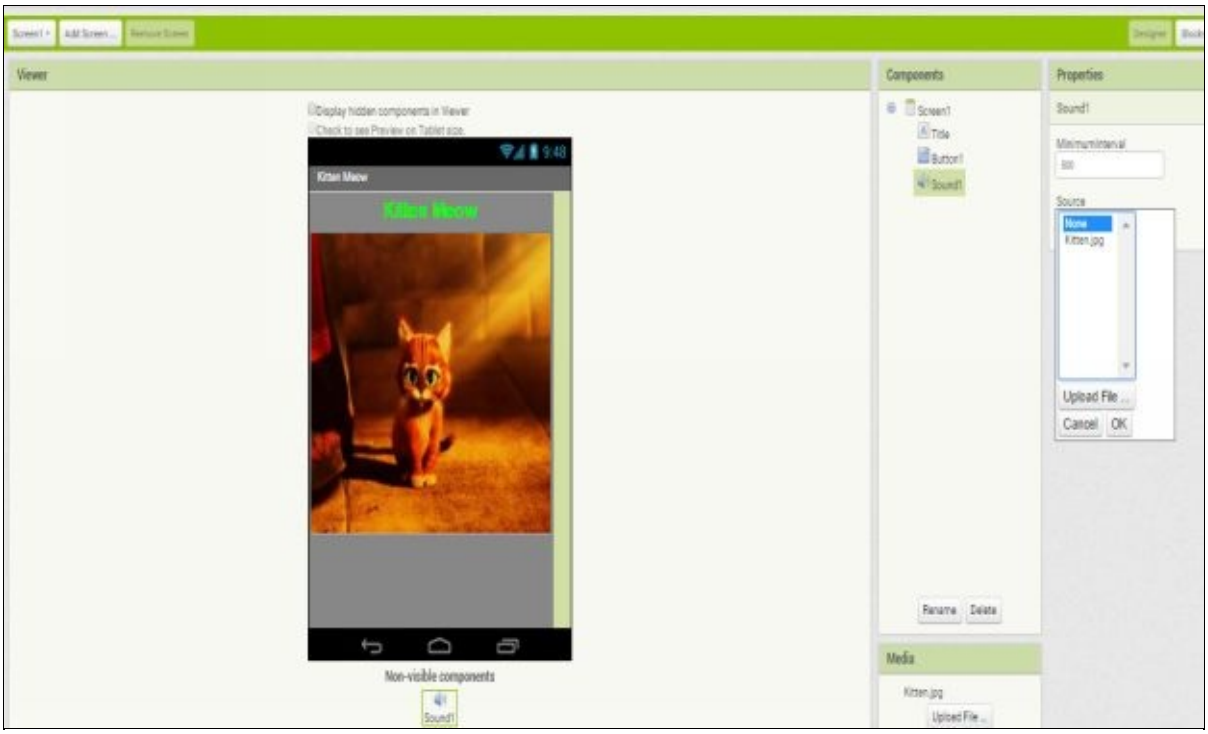
8. Delete Button1 Text under Button1 Properties and set Button1 width to Fill parent and Button1 Height to 70 percent.



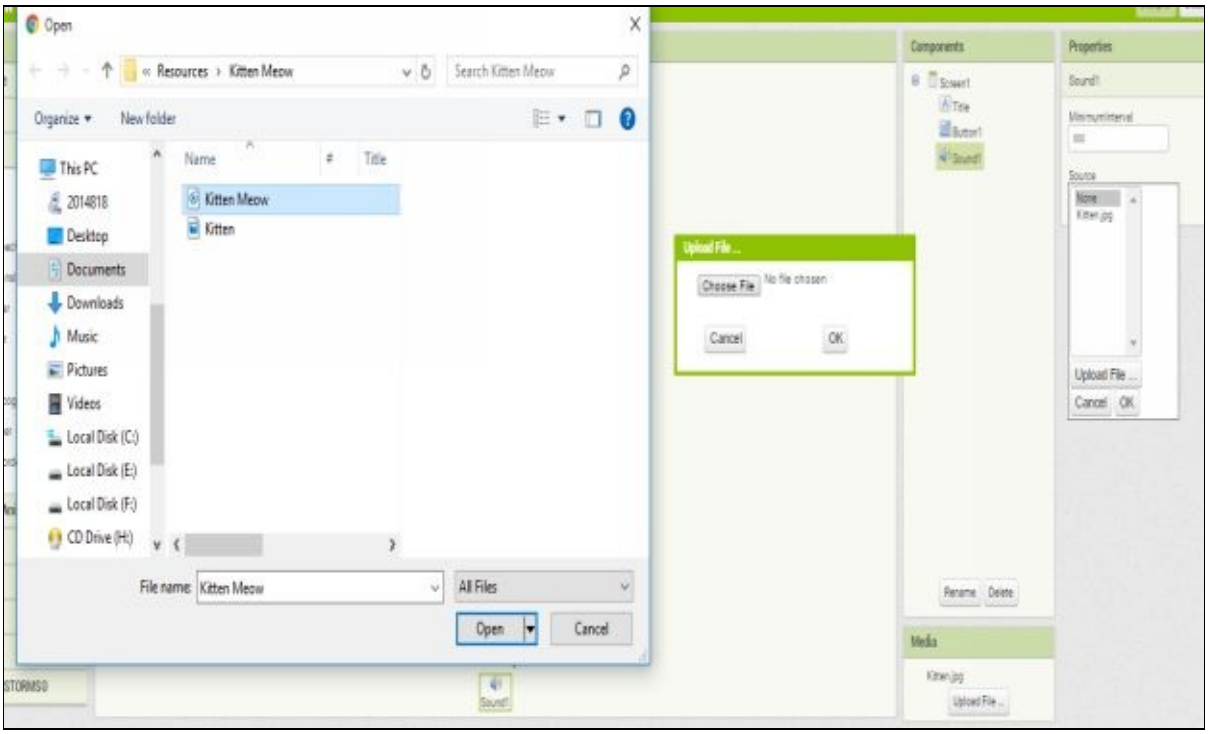
9. Now drag a Sound component from Palette to Viewer.



10. Click on Source property of Sound1 component under Properties.

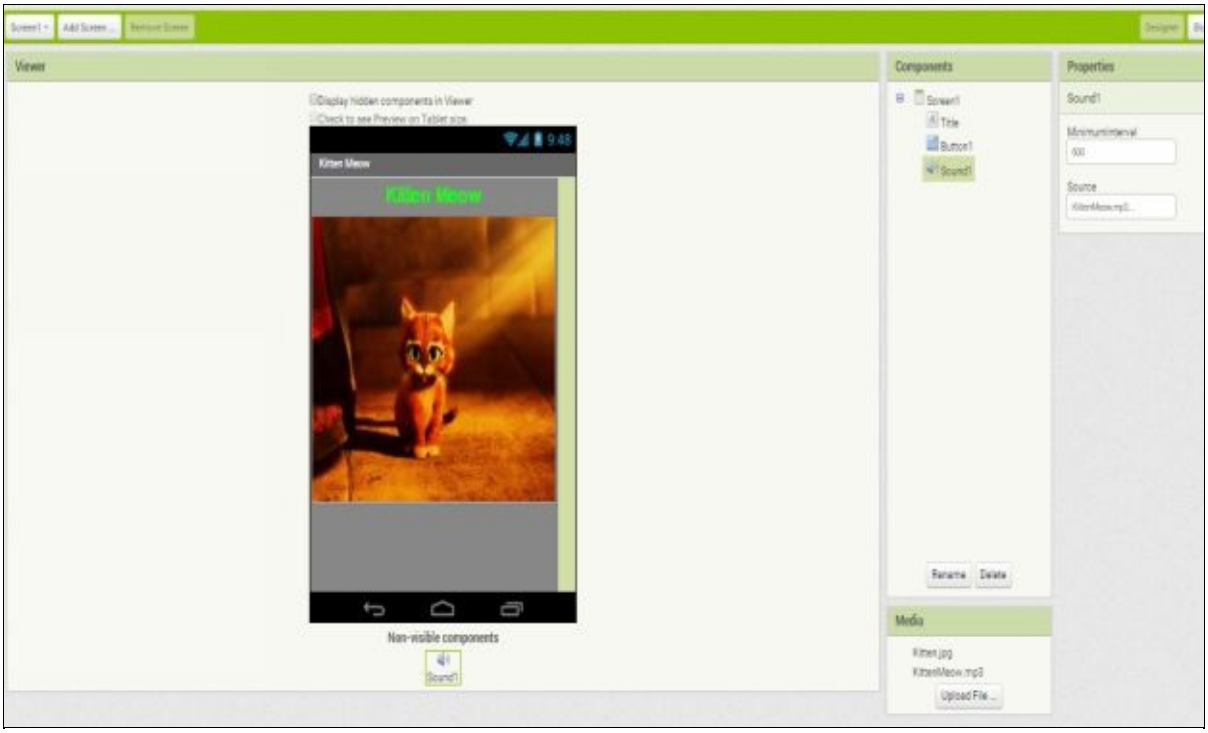


Click on Upload File and then click on Choose File.



Select KittenMeow.mp3 File and click Open.

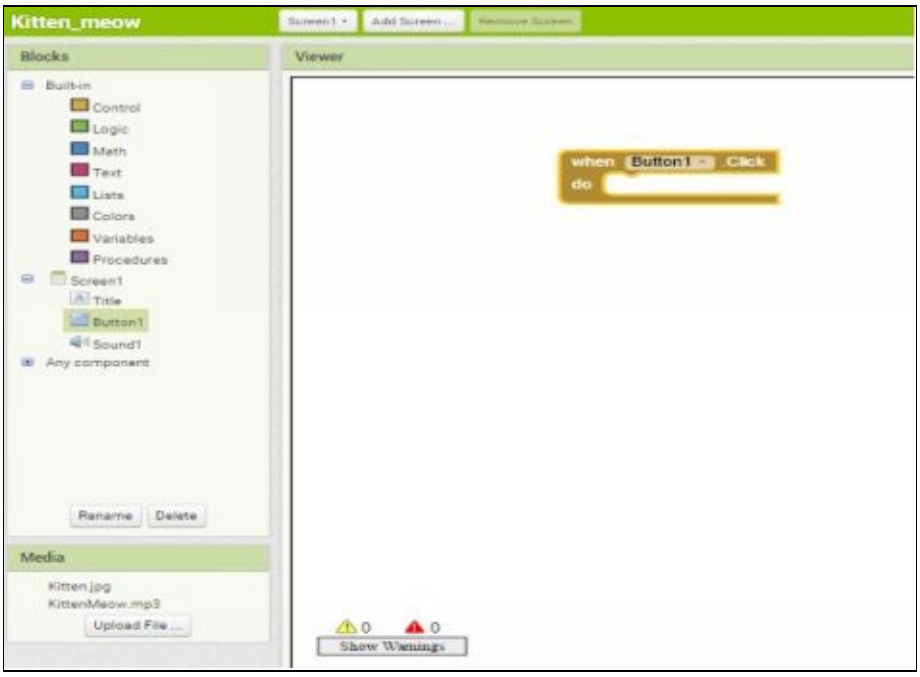
Click OK.

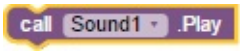


11. Now click on Blocks.

12. Now click on Button1 component on left hand side and choose 

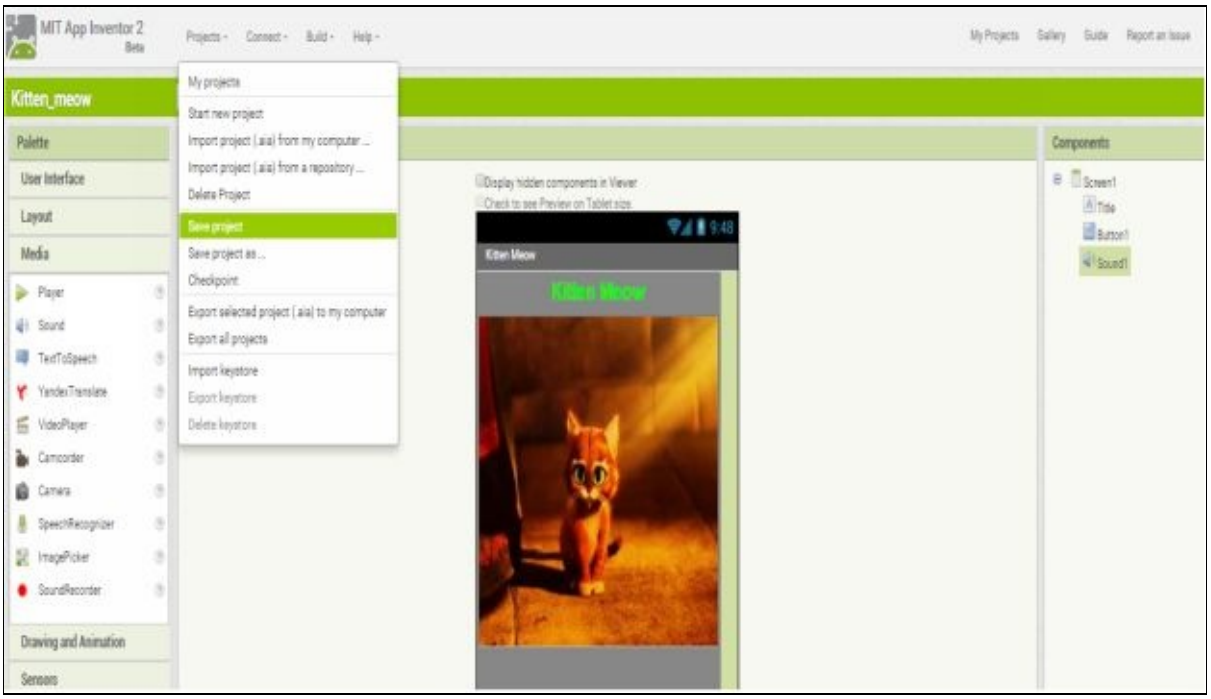
block. This is an event block which will decide what your app will do when this button will be clicked. Here click is an event for Button1 component.



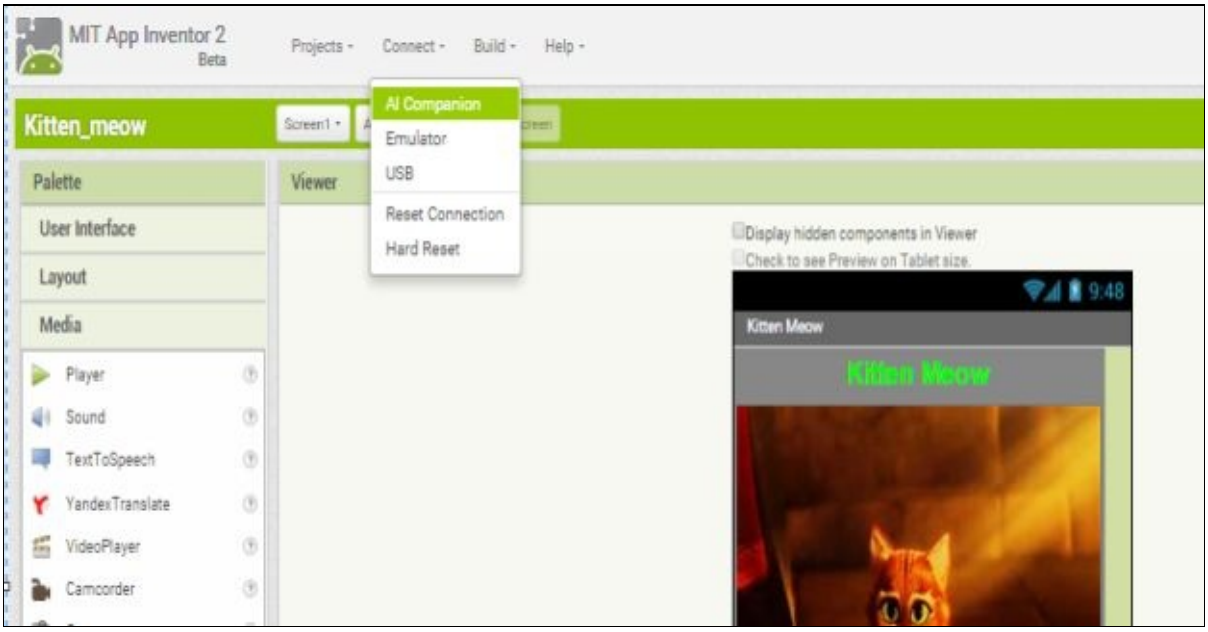
13. Now click on Sound1 Component and select  block. This block will play some Sound1's source file(KittenMeow.mp3).

14. Now connect these blocks as shown below. This block will tell our app to play KittenMeow.mp3 when Button1 is clicked.

15. Now go to Designer and save the Project.



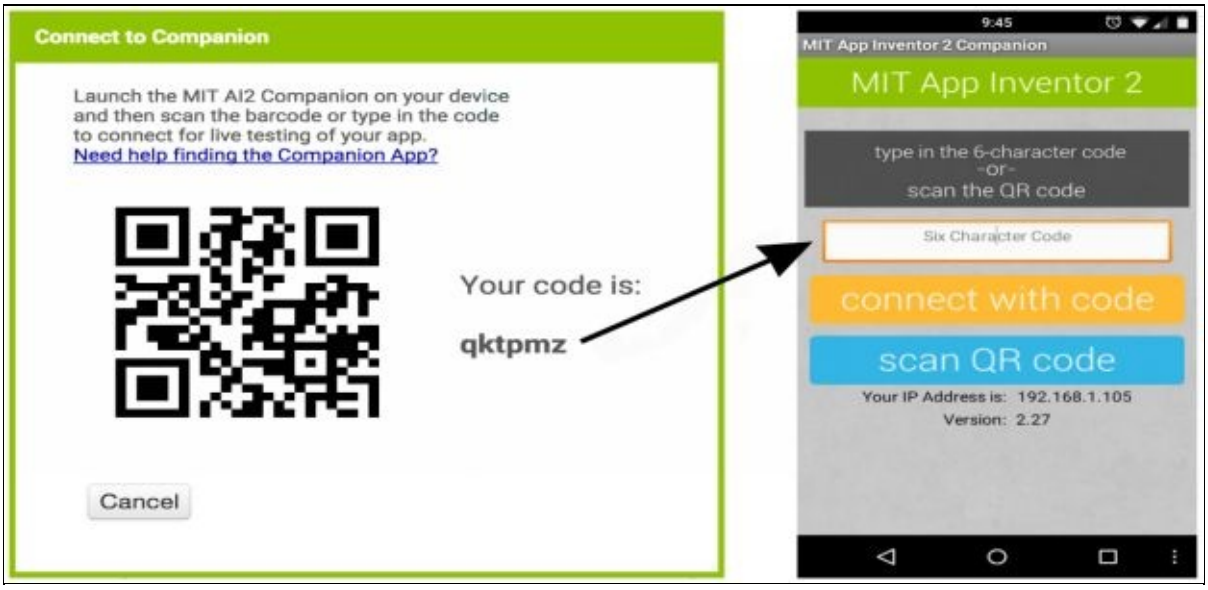
16. Now click on Connect and select AI Companion.



17. Now open MIT App Inventor 2 Companion in your mobile/Tablet.

And enter the code or scan the QR code from your mobile and click on Connect with code.

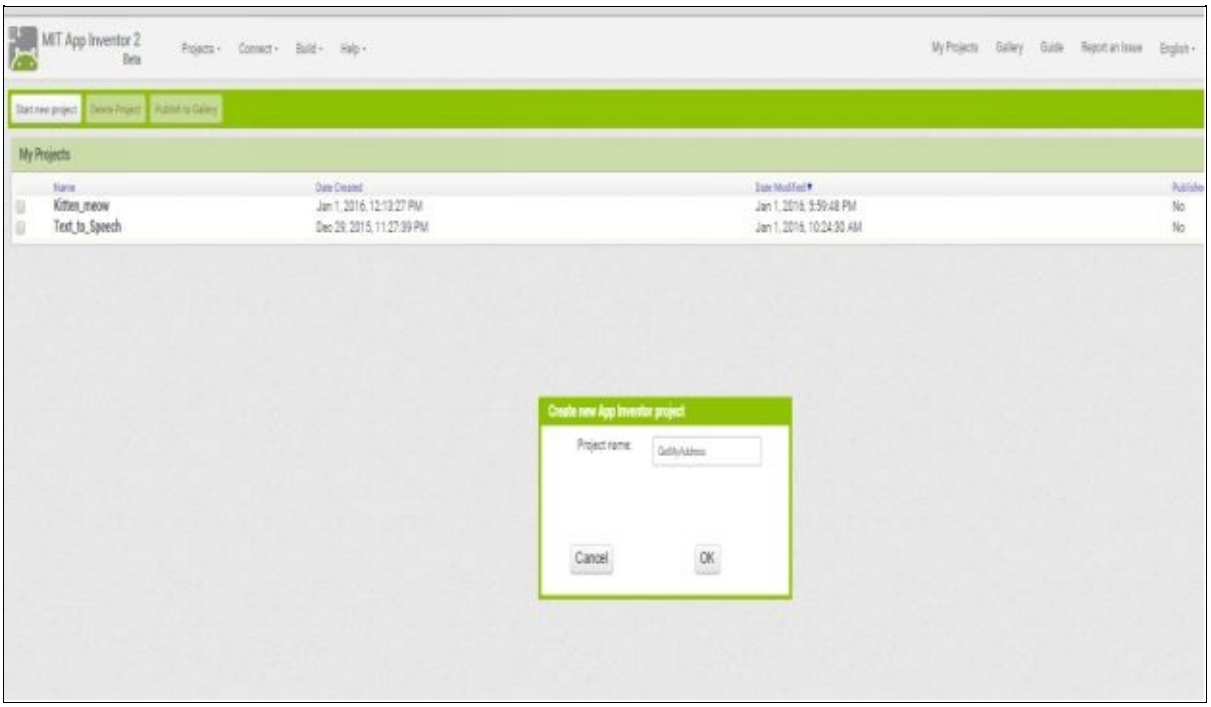
Note:-Your Mobile/Tablet should be connected to same wi-fi/network to which your computer/Laptop is connected.



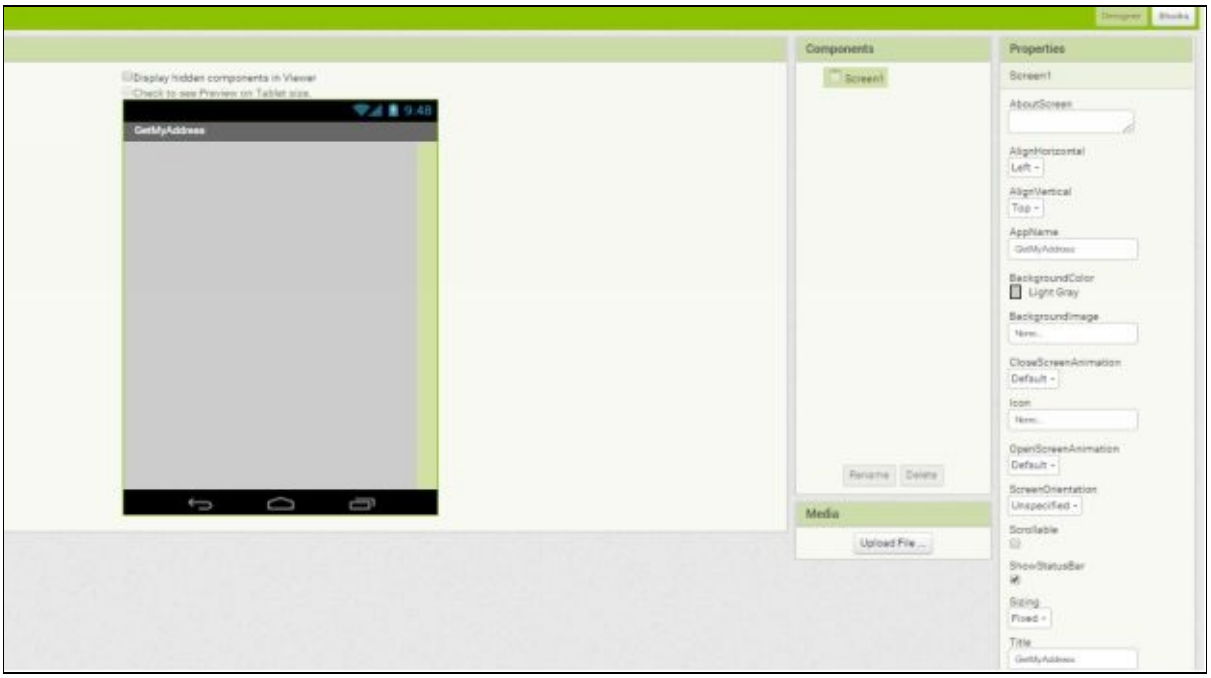
18. Now you should see your app in your phone for live testing. Touch the Kitten and it should play Meow sound.

Chapter 4: GetMyAddress (GPS) App

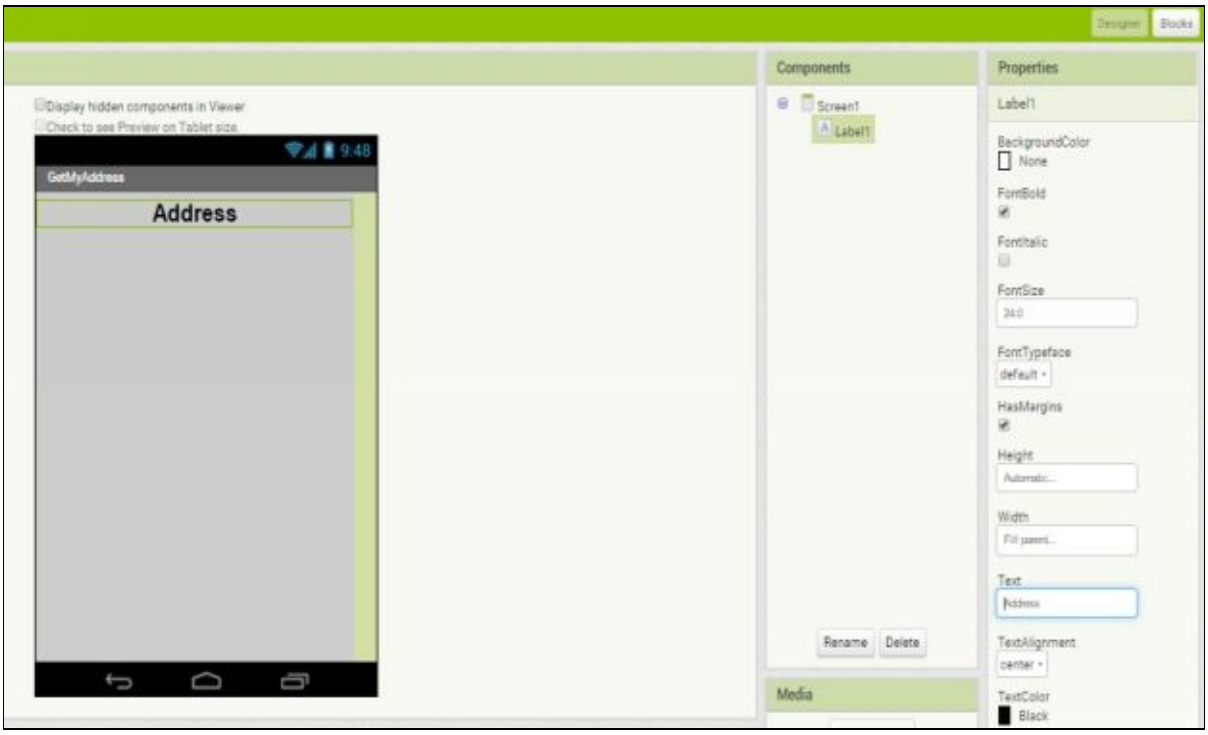
1. Click on Start new project and give it name GetMyAddress and click OK.



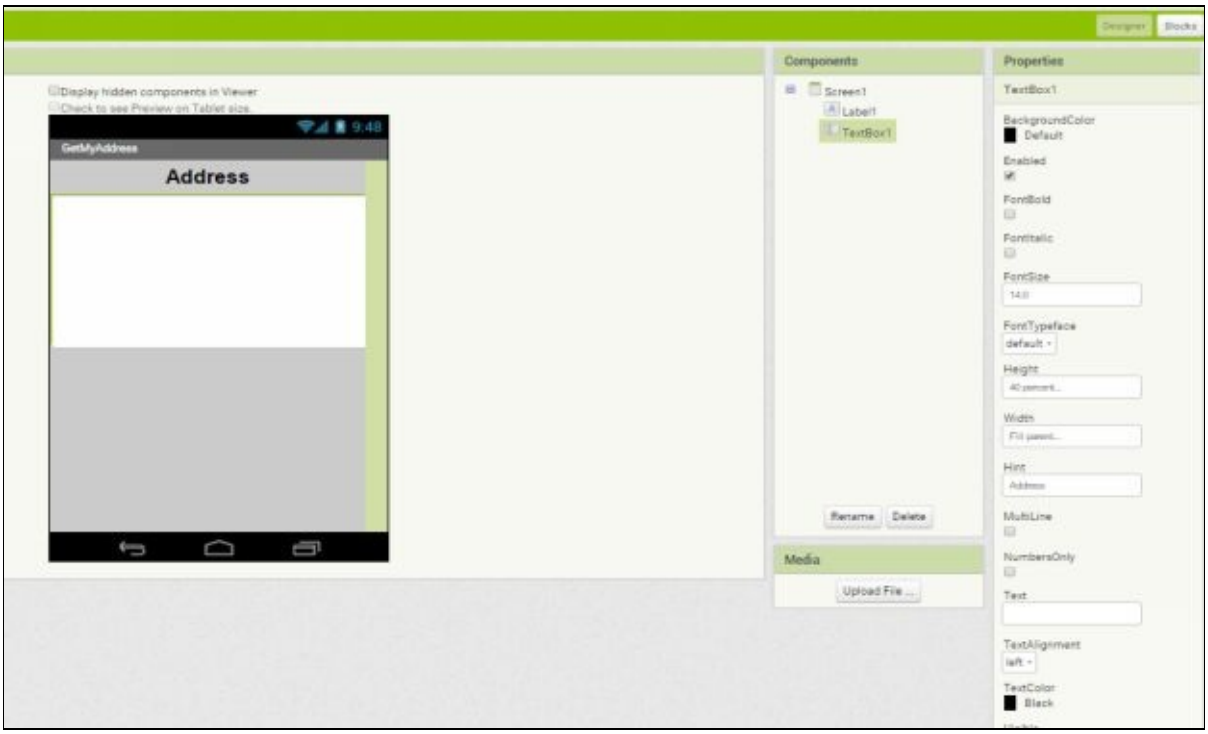
2. Change Screen1 Text to GetMyAddress and BackgroundColor to LightGray.



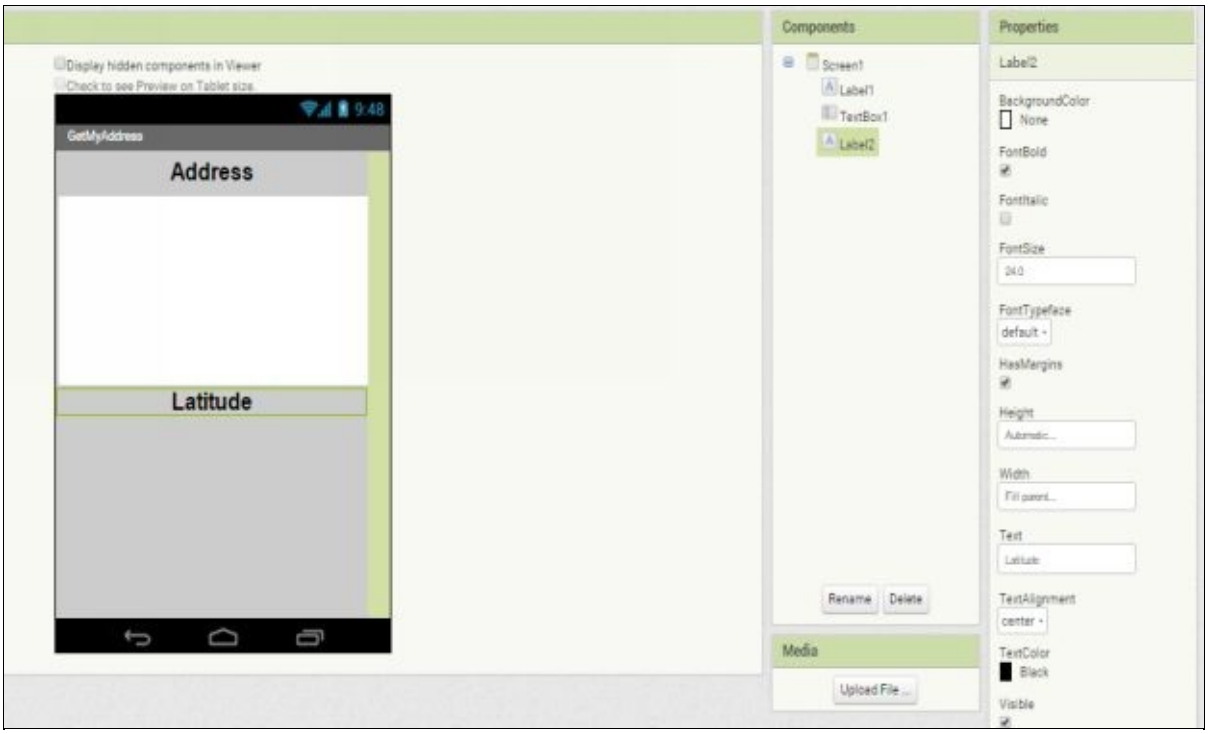
3. Drag a Label from Palette to Viewer screen and set its Properties as shown below.



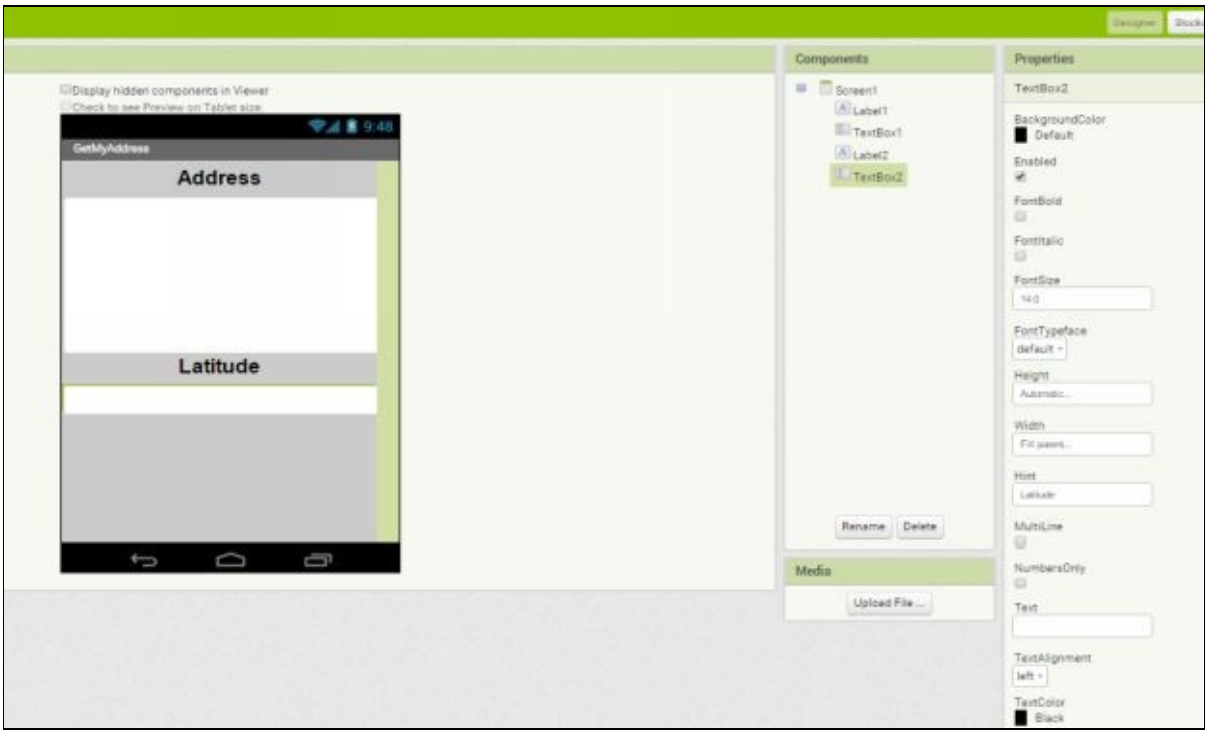
4. Drag a TextBox from Palette to Viewer screen and set its Properties as shown below.



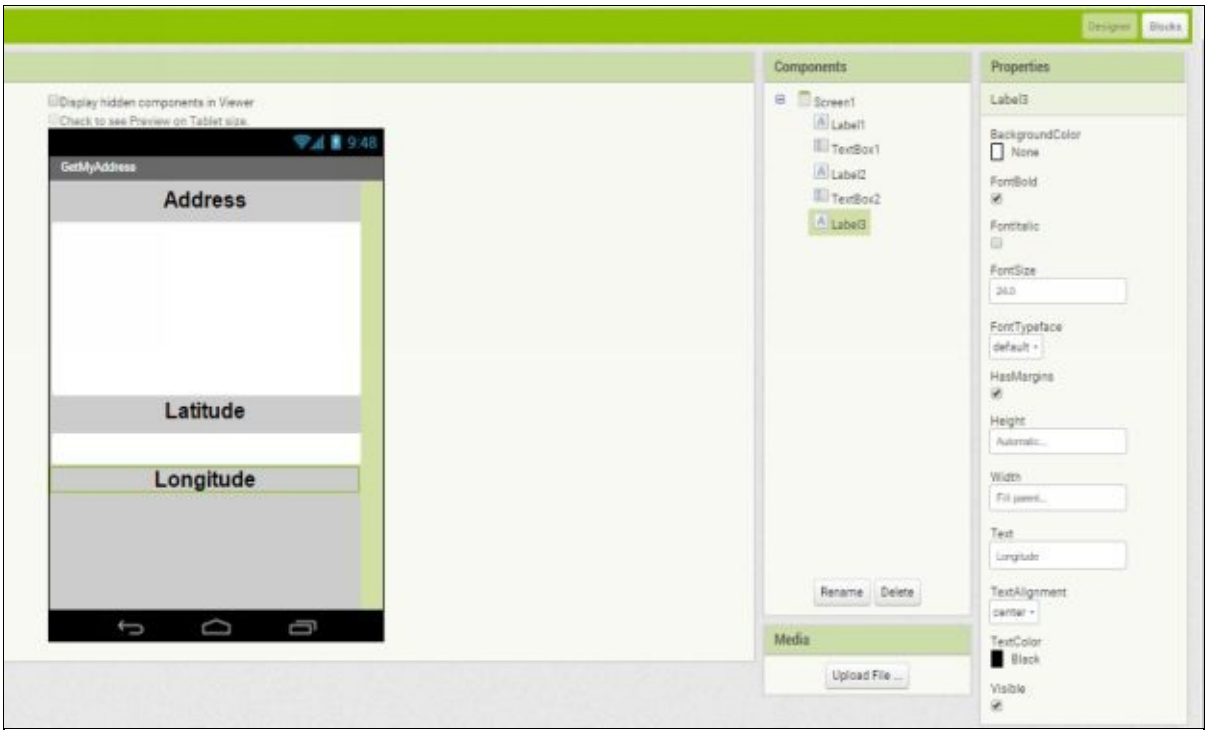
5. Drag another Label from Palette to Viewer screen and set its Properties as shown below.



6. Drag another TextBox from Palette to Viewer screen and set its Properties as shown below.

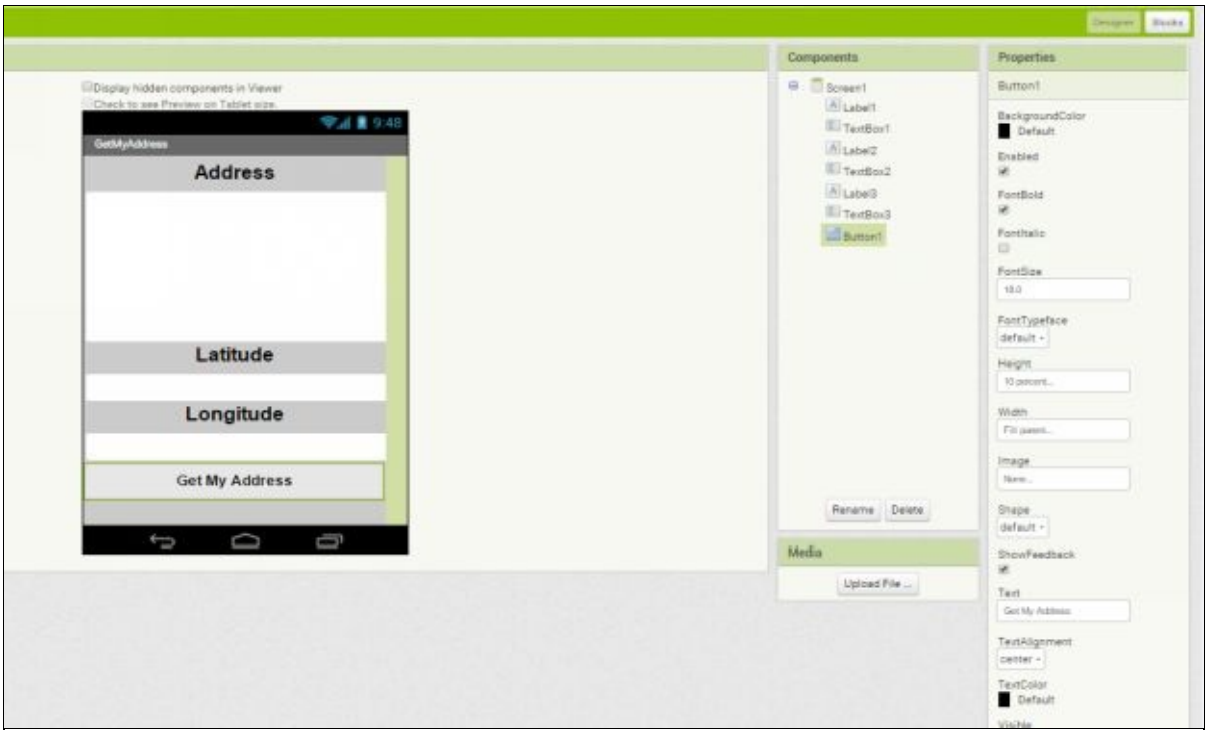


7. Drag another Label from Palette to Viewer screen and set its Properties as shown below.



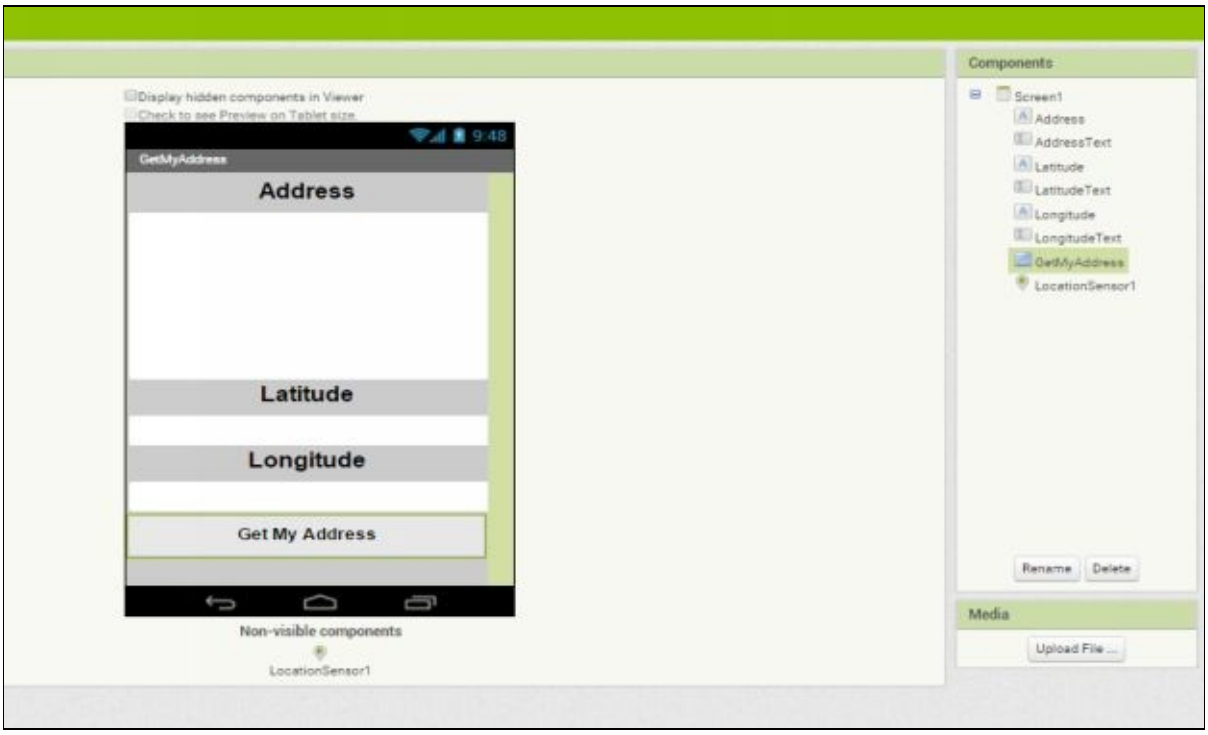
8. Drag another TextBox from Palette to Viewer screen and set its Properties as shown below.

9. Drag a Button from Palette to Viewer screen and set its Properties as shown below.



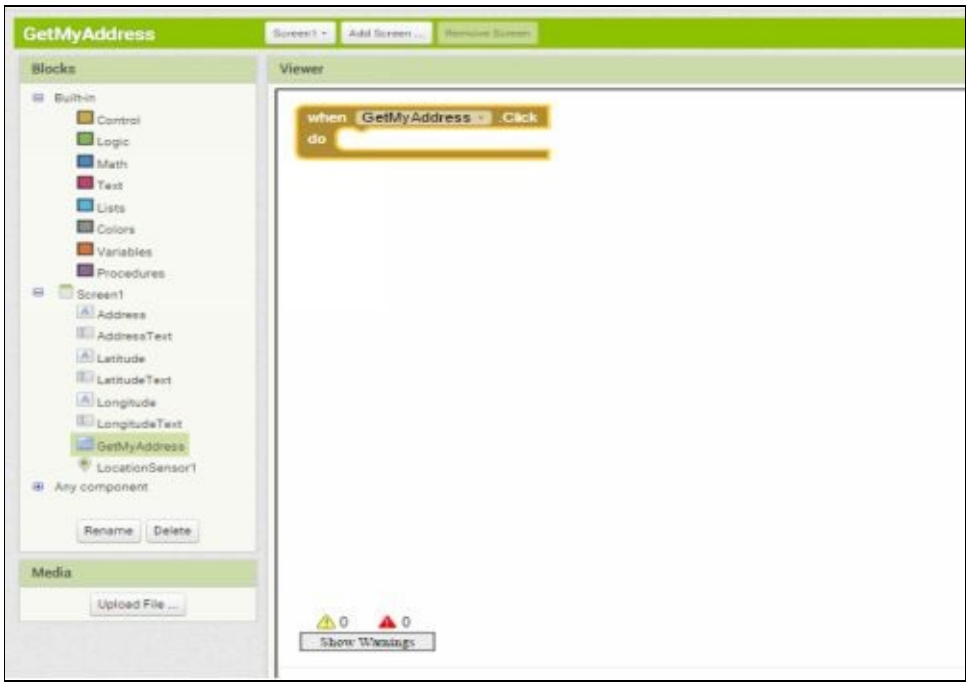
10. Drag LocationSensor component from Palette to Viewer screen.

11. Rename the components as shown below.



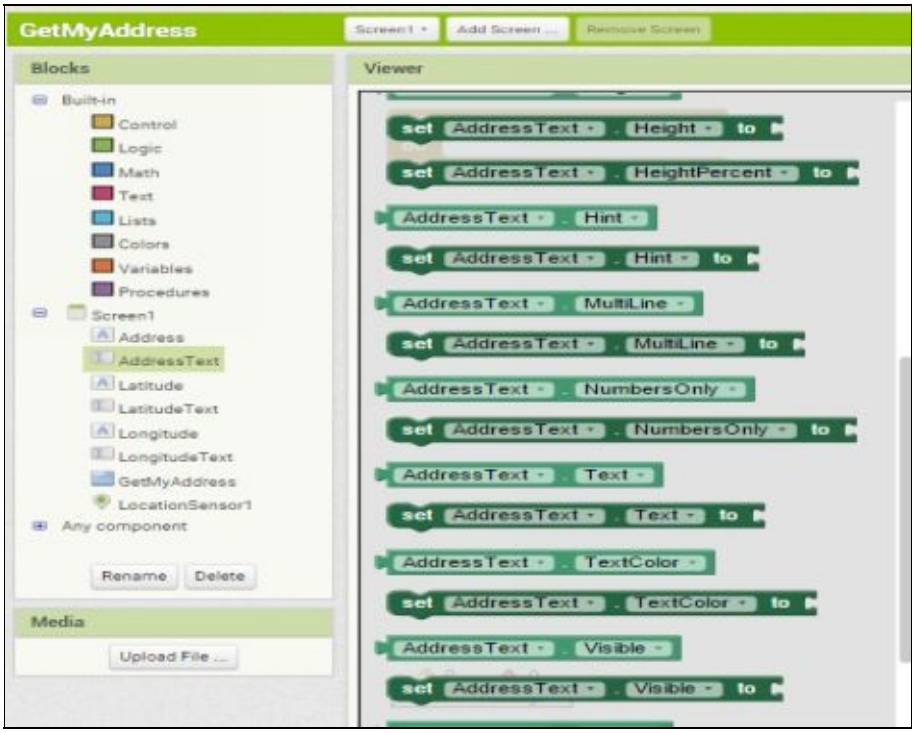
12. Now go to Blocks.

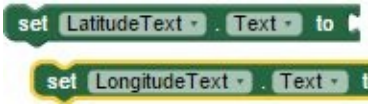
13. Click on GetMyAddress button component in left side and click on block. This is an event block which will decide what your app will do when ‘Get My Address’ button will be clicked. Here click is an event for Button1 component.



14. Now click on AddressText Text component and scroll down to select 

block.

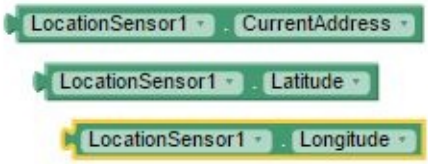


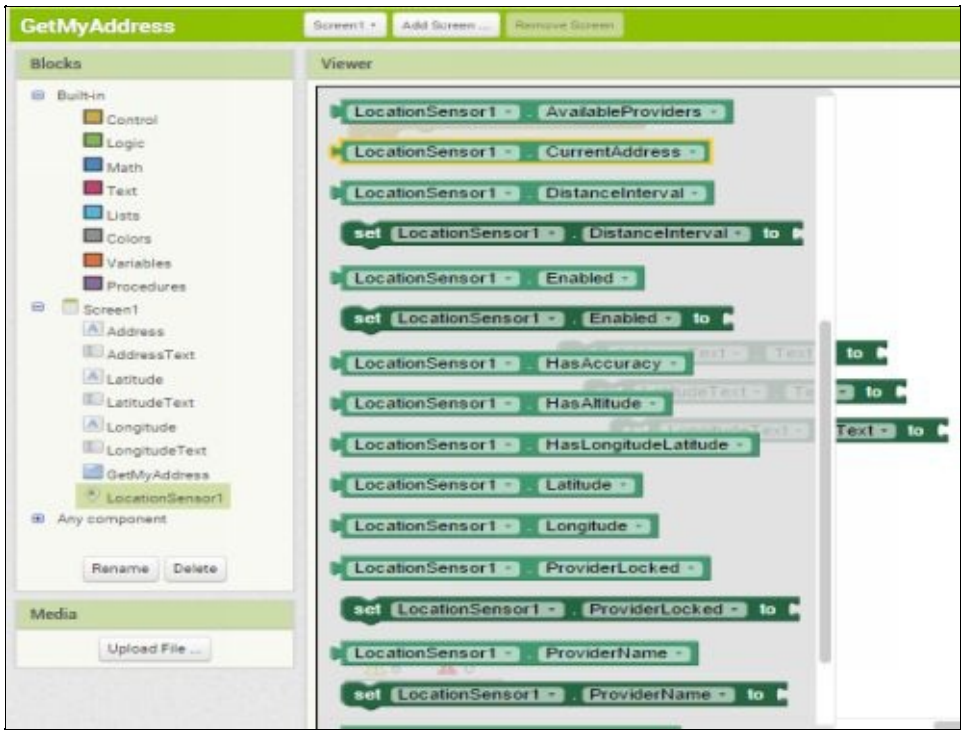
15. Now similarly select 

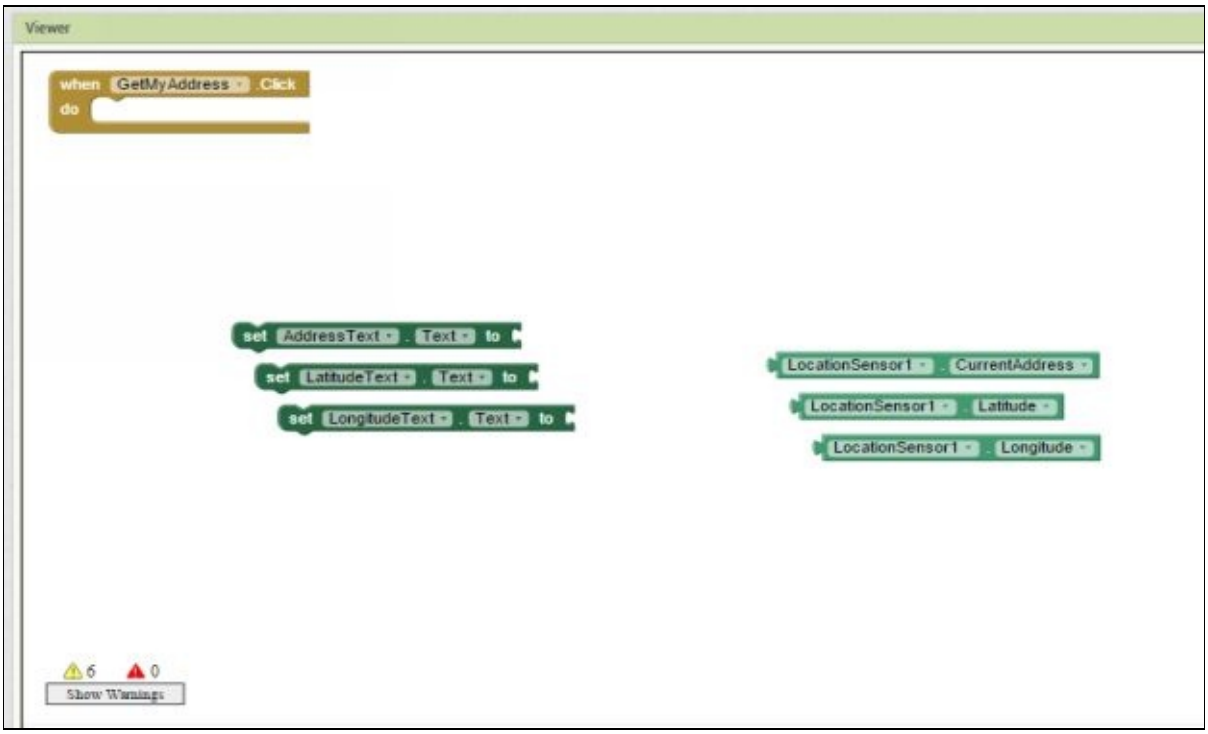
blocks From LatitudeText and LongitudeText components.

16. Now click on LocationSensor1 Component and select these 3 blocks. These will use

your device’s GPS or location sensor and will get your CurrentAddress, Latitude and Longitude.



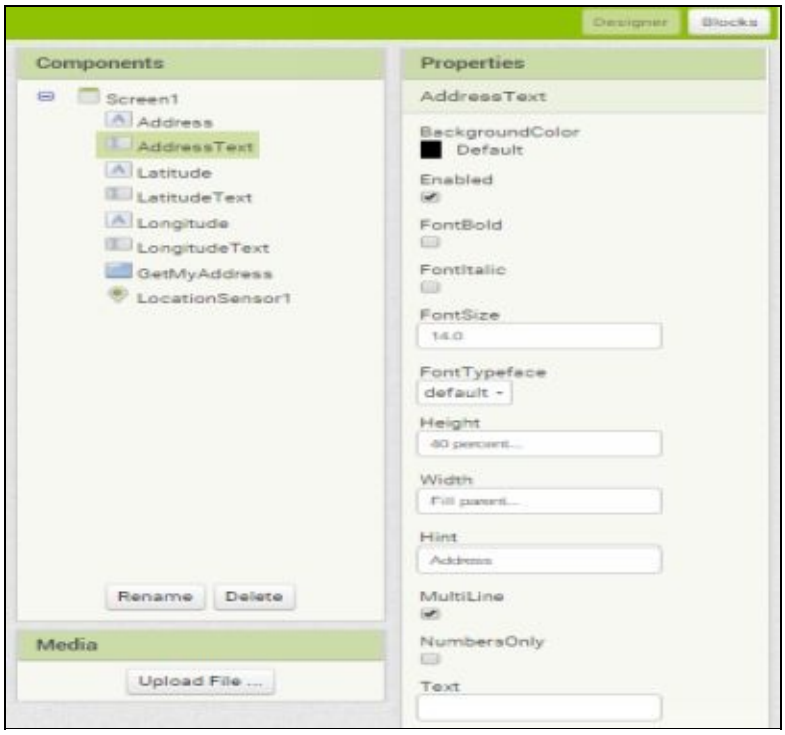




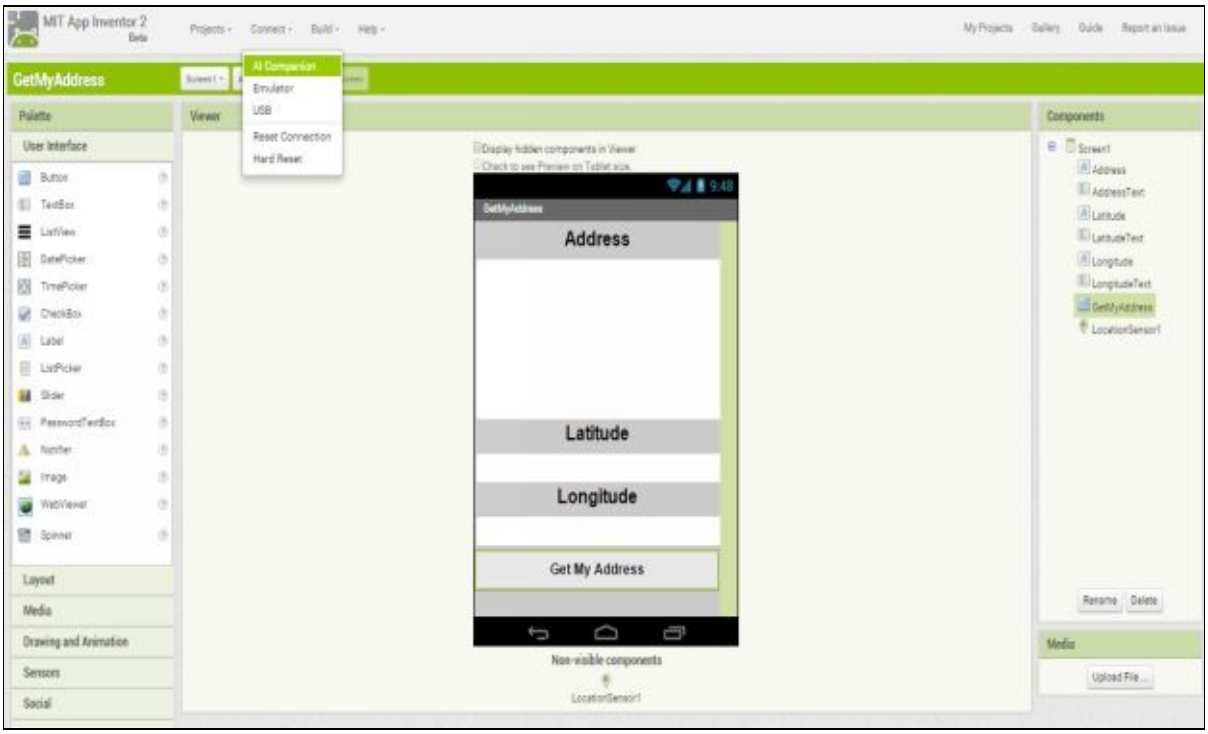
17. Now attach these blocks as shown below. So that when ‘Get My Address’ button is clicked your app will show your CurrentAddress,Latitude and Longitude in respective

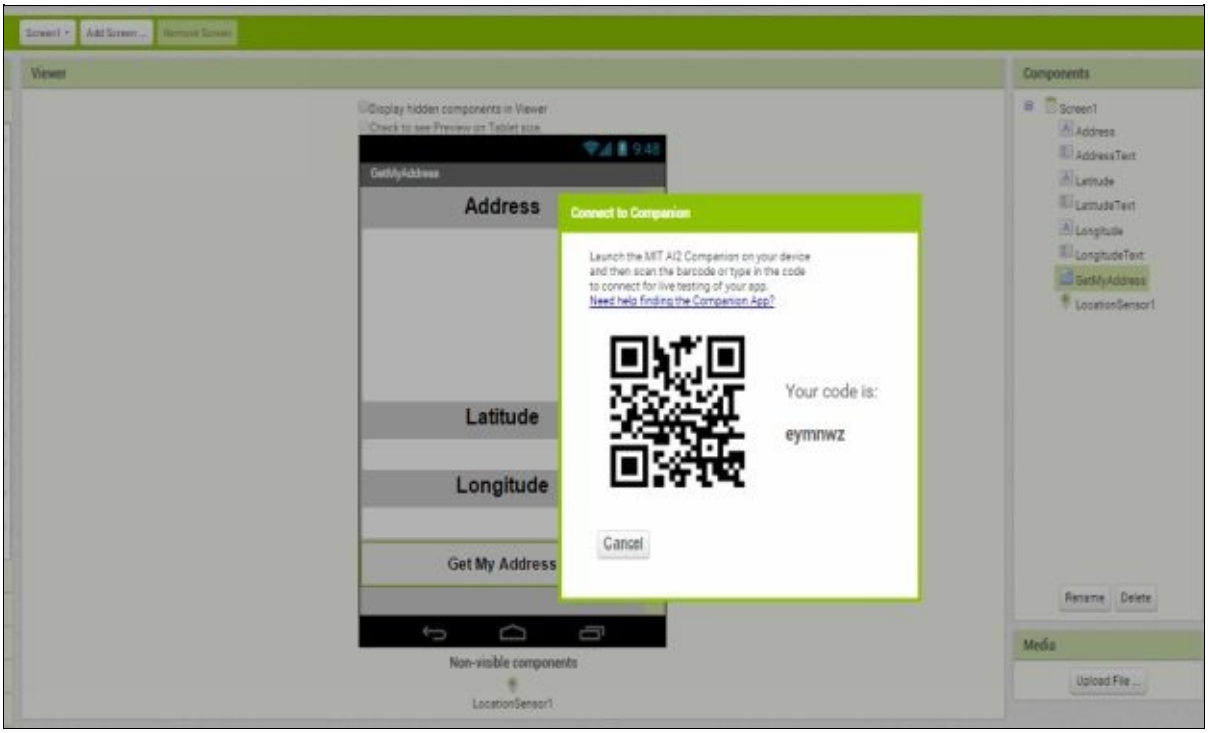
Textboxes.

18. Now go to Designer and enable Multiline for AddressText.

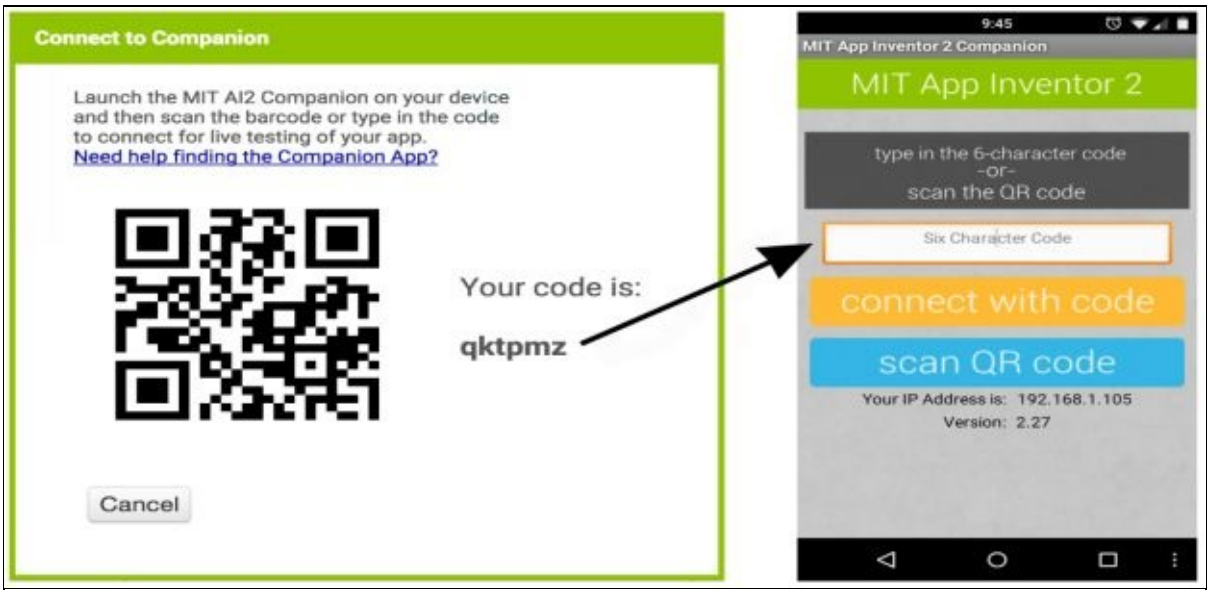


19. Click on Connect and select AI Companion.





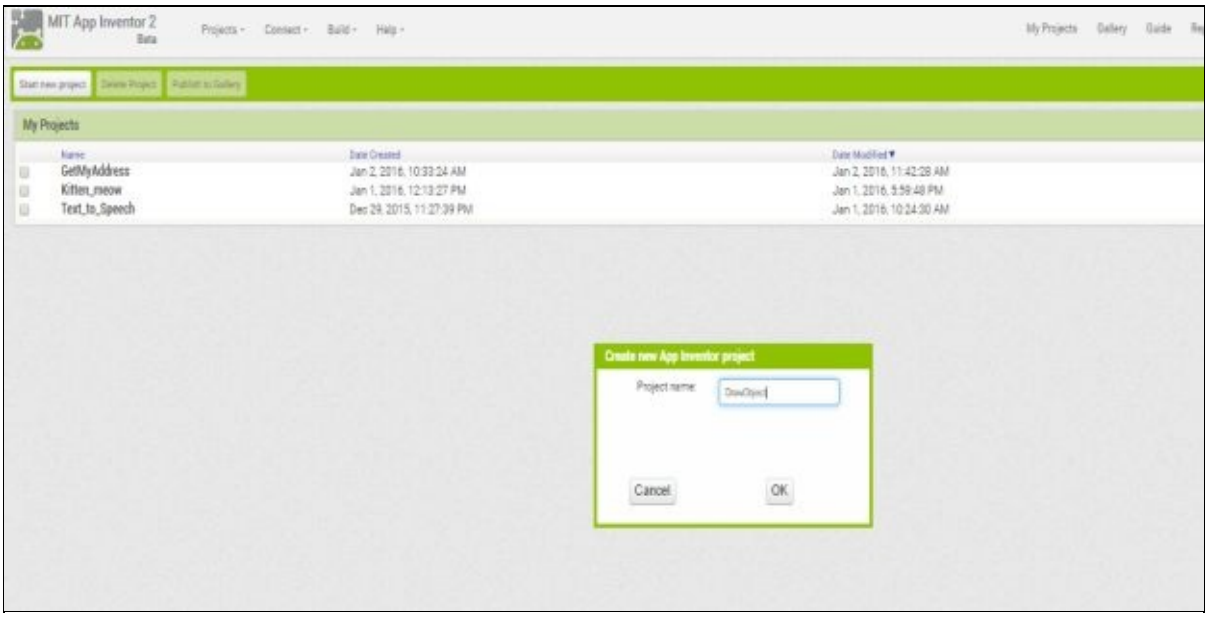
20. Now open MIT App Inventor 2 Companion in your mobile/Tablet.And enter the code or scan the QR code from your mobile and click on connect with code.Note:-Your Mobile/Tablet should be connected to same wi-fi/network to which your computer/Laptop is connected.



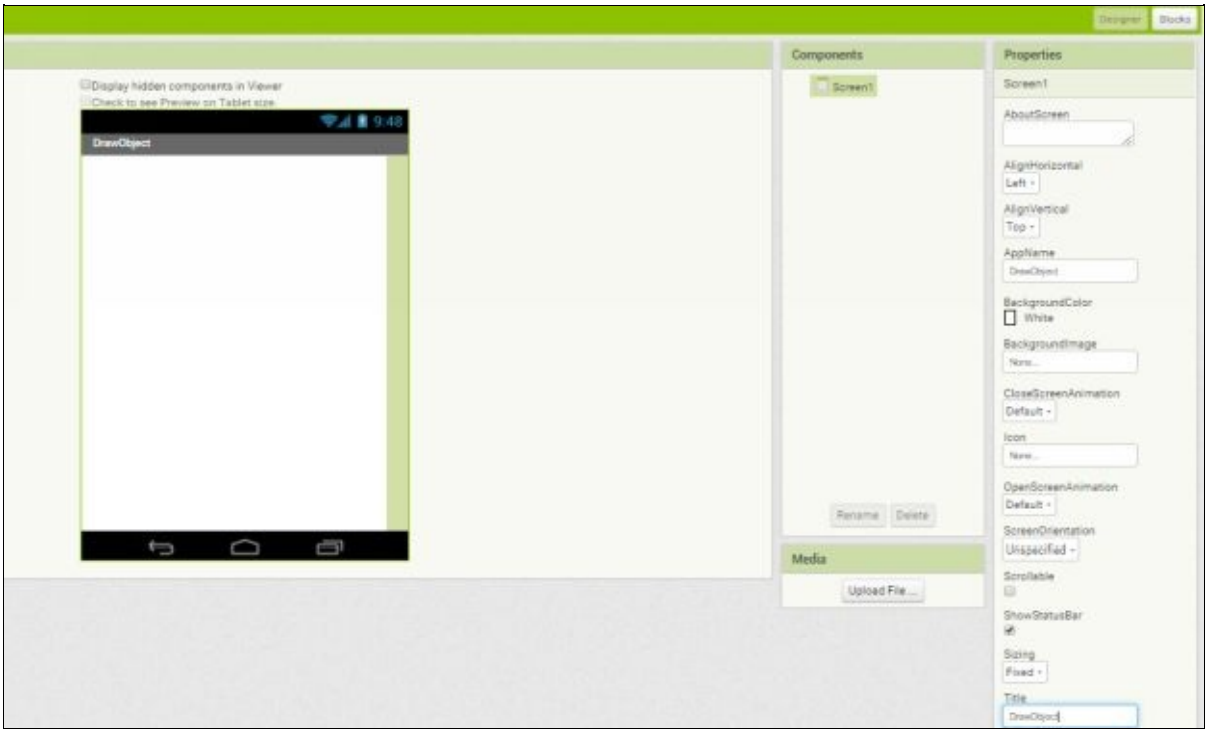
21. Now you should see your app in your phone for live testing. Enter Click on Get My Address Button to get your current Address,Latitude and Longitude.Note:-To test this app switch On GPS sensor in your phone/Tablet.

Chapter 5: DrawObject App

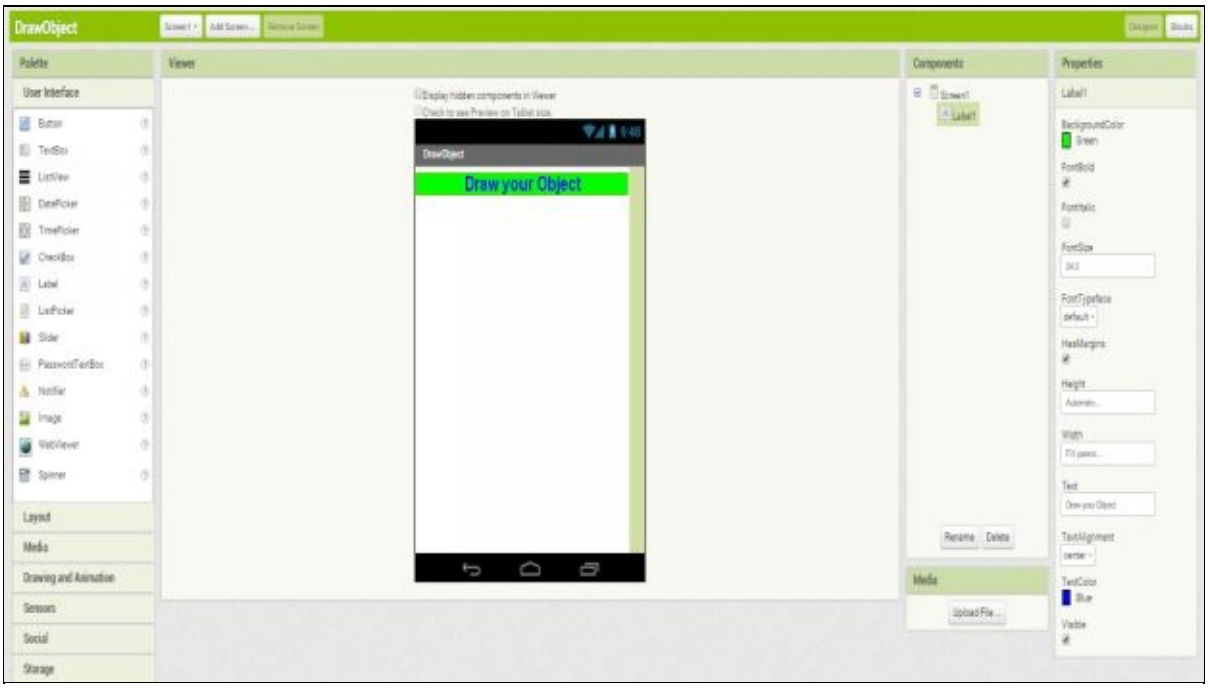
1. Click on Start new project and give it name DrawObject and click OK.



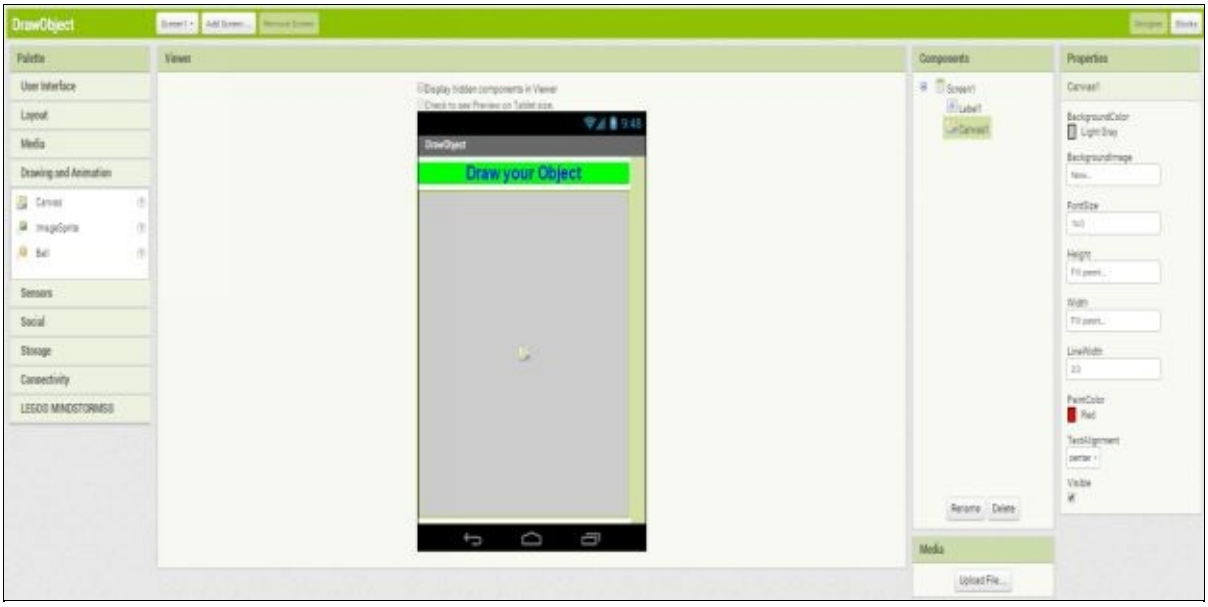
2. Change Screen1 Title to DrawObject.



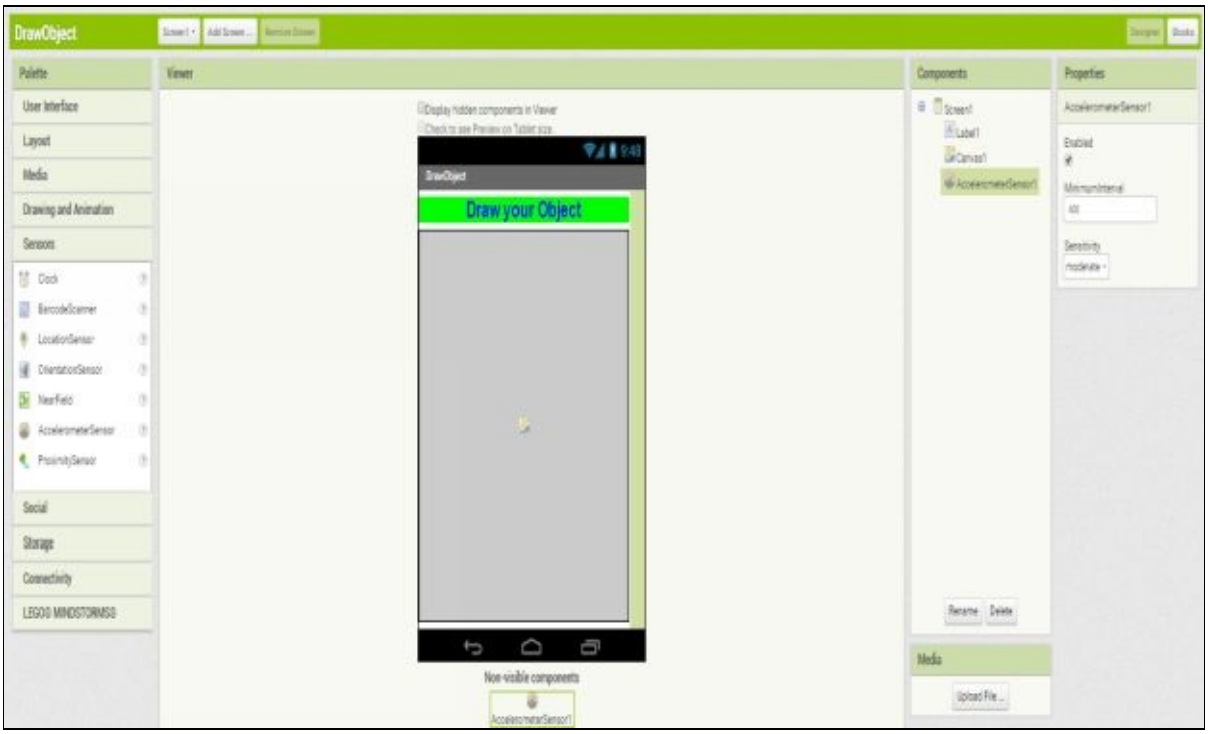
3. Drag a Label from Palette to Viewer Screen and set its Properties as shown below.



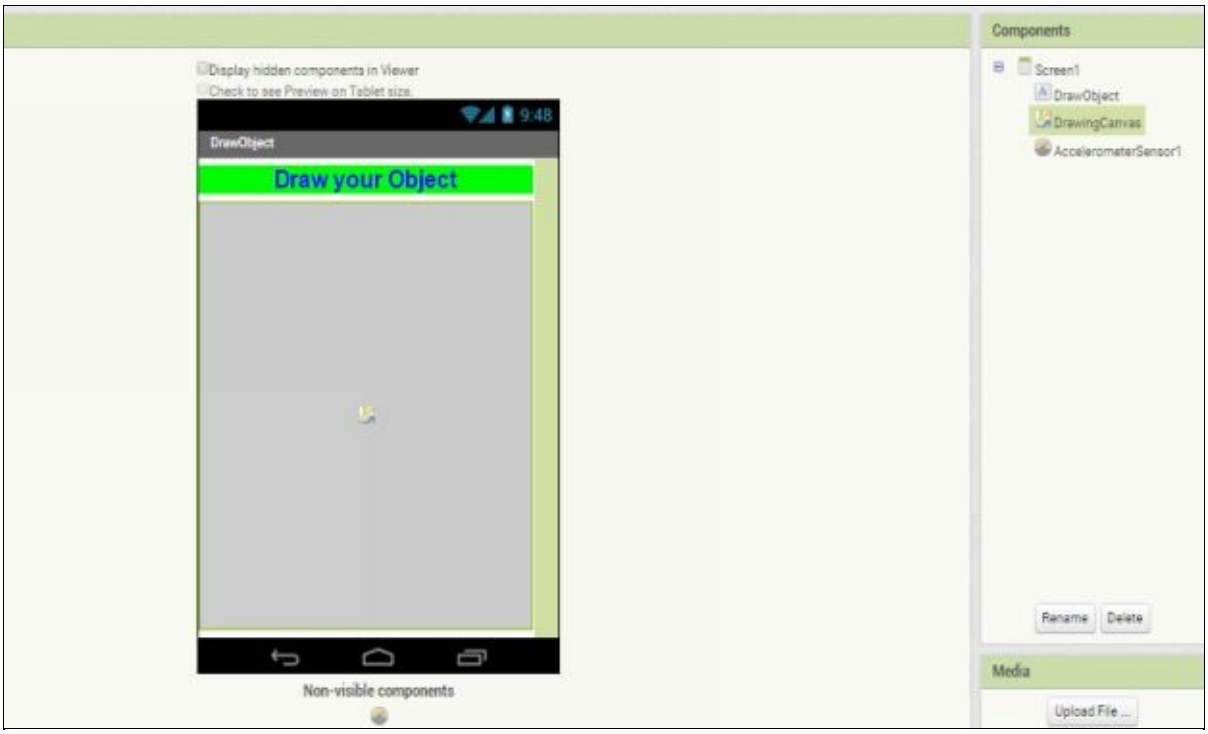
4. Drag a Canvas from palette to Viewer screen and set its Properties as shown below.



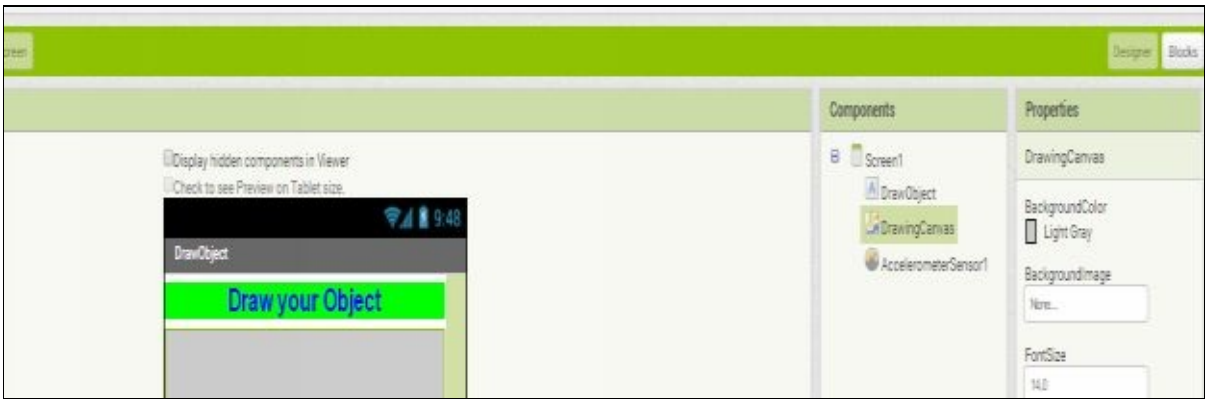
5. Drag an AcclerometerSensor from Palette to Viewer screen.

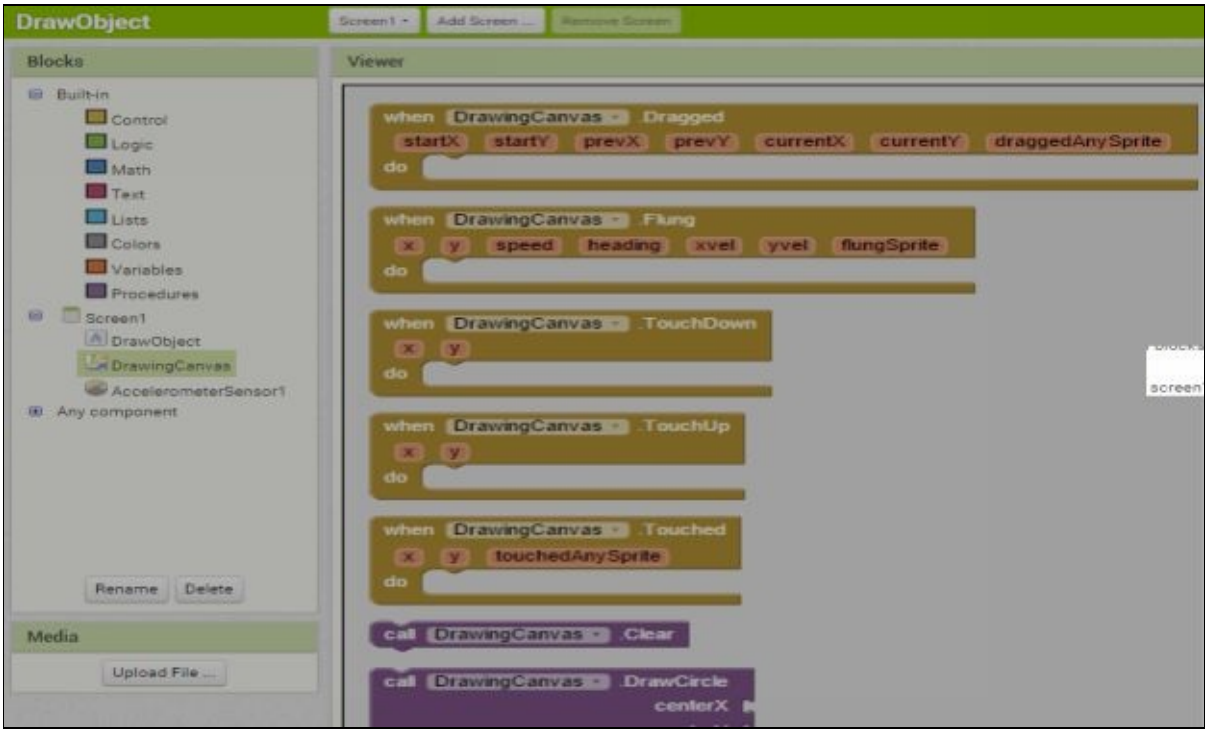


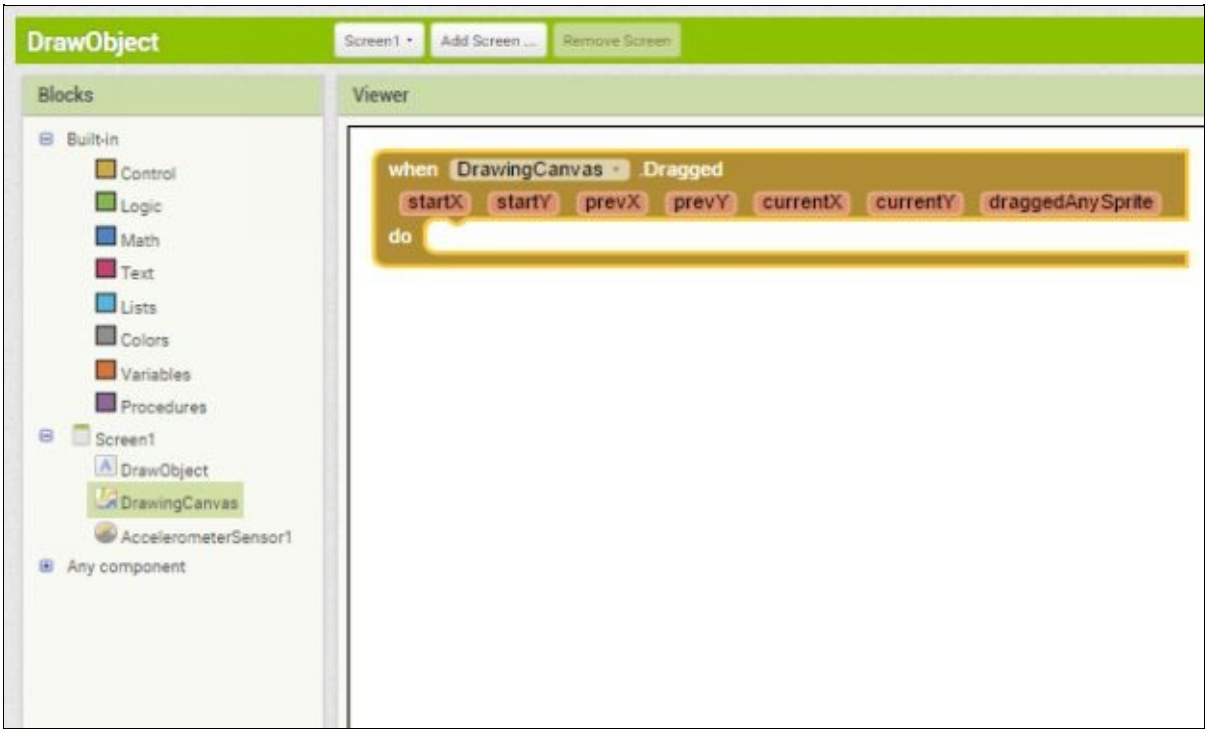
6. Change component names as shown below.

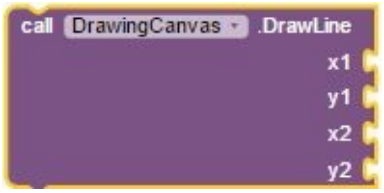


7. Now go to Blocks and click on DrawingCanvas component on left side and select block. This block will identify the event when you touch your app screen/canvas and drag it.

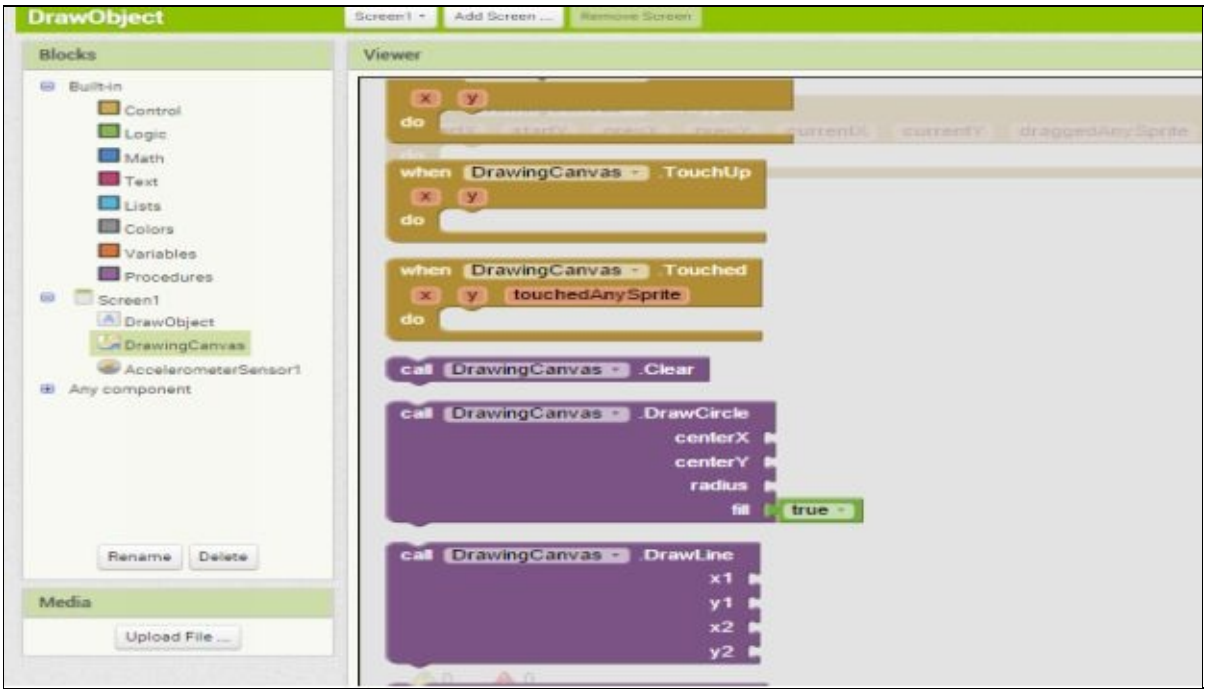

Designer
Blocks
Components
Properties
Display hidden components in Viewer
Check to see Preview on Tablet size.
9:48
DrawObject
Draw your Object
Screen1
DrawObject
DrawingCanvas
AccelerometerSensor1
DrawingCanvas
BackgroundColor
Light Gray
BackgroundImage
None...
FontSize
14.0

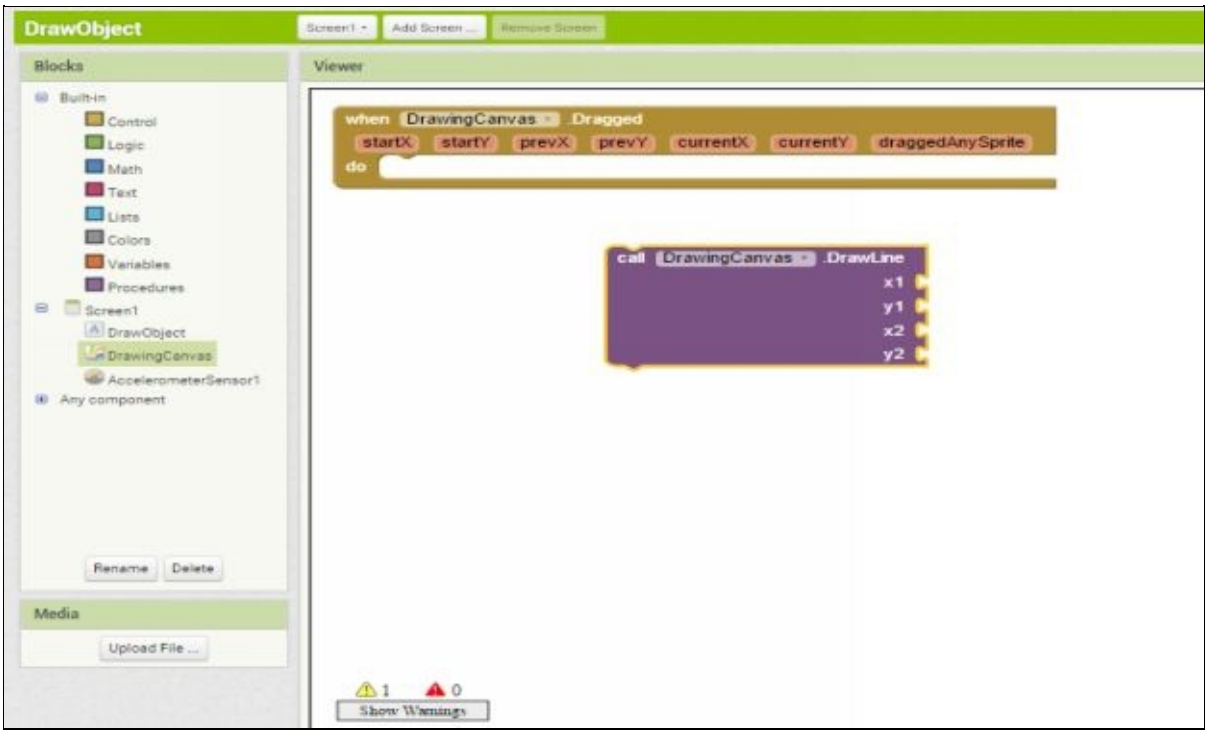

DrawObject
Screen1
Add Screen ...
Remove Screen
Blocks
Viewer
Built-in
Control
Logic
Math
Text
Lists
Colors
Variables
Procedures
Screen1
DrawObject
DrawingCanvas
AccelerometerSensor1
Any component
Rename
Delete
Media
Upload File ...
when DrawingCanvas .Dragged
startX startY prevX prevY currentX currentY draggedAnySprite
do
when DrawingCanvas .Flung
x y speed heading xvel yvel flungSprite
do
when DrawingCanvas .TouchDown
x y
do
when DrawingCanvas .TouchUp
x y
do
when DrawingCanvas .Touched
x y touchedAnySprite
do
call DrawingCanvas .Clear
call DrawingCanvas .DrawCircle
centerX
screen


DrawObject
Screen1
Add Screen ...
Remove Screen
Blocks
Viewer
Built-in
Control
Logic
Math
Text
Lists
Colors
Variables
Procedures
Screen1
DrawObject
DrawingCanvas
AccelerometerSensor1
Any component
when DrawingCanvas .Dragged
startX startY prevX prevY currentX currentY draggedAnySprite
do

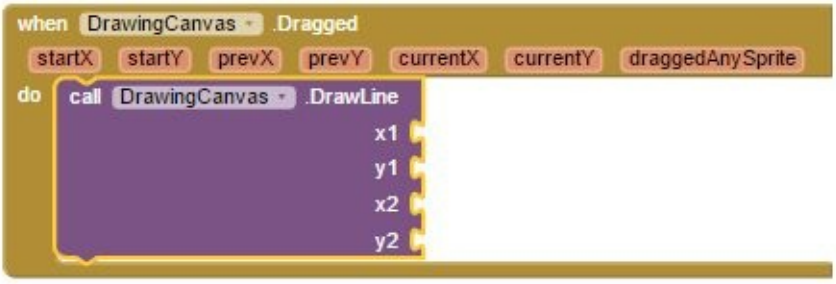
8. Now select 

block from DrawingCanvas. This block will draw a line on your canvas. A line has starting point(x1,y1) and ending point(x2,y2).



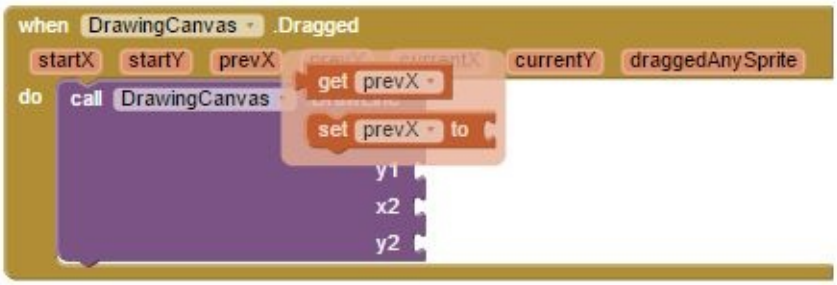


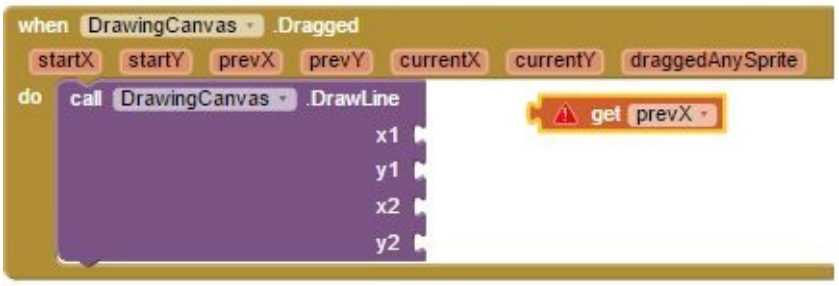
9. Now attach these blocks as shown below. So that when you touch the screen and drag it , a line will be drawn on your canvas. Here prevX and prevY will be the starting point of line and currentX and currentY will be the ending point of line.



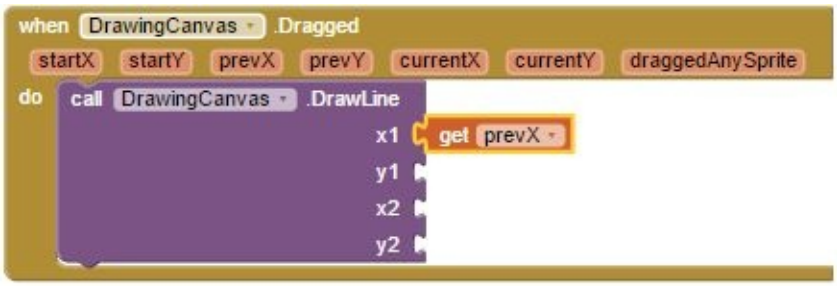
Now Mouseover on prevX and select 

block.

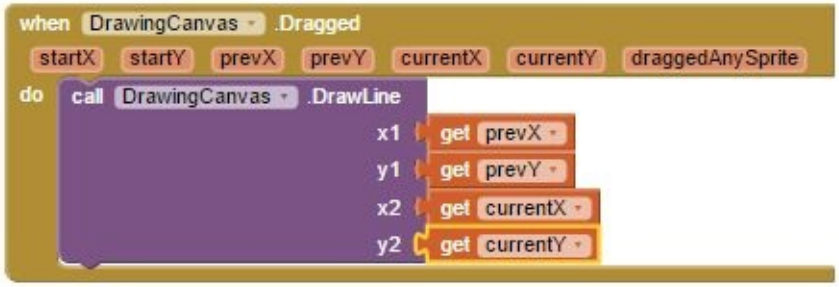




Now attach this as show below.



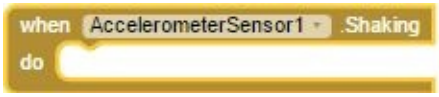
Similarly do it for y1,x2,y2 and attach as shown below.



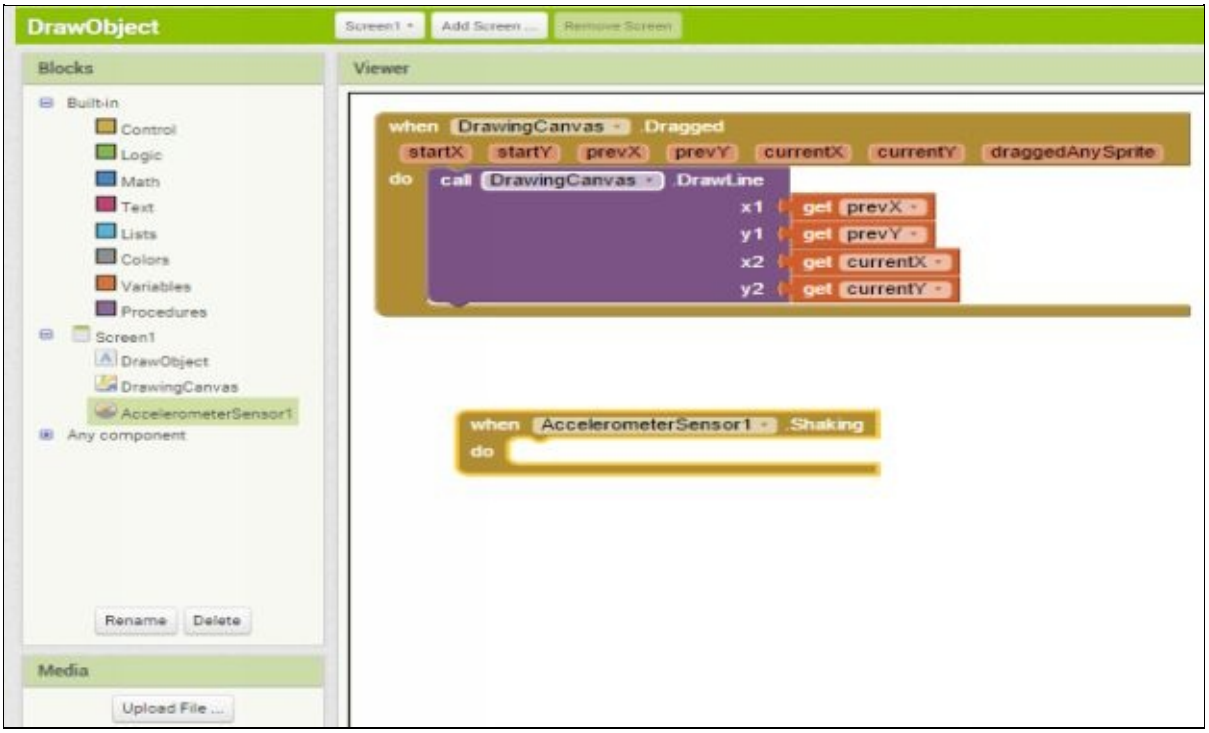
This logic will draw a line on canvas where prevX and prevY are the co-ordinates of starting point of line and currentX and currentY are the co-ordinates of end point of line.

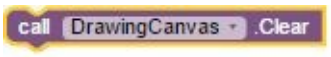
10. Now we will add logic to clear canvas on shaking the phone.

For this click on AccelerometerSensor1 component on left side and select



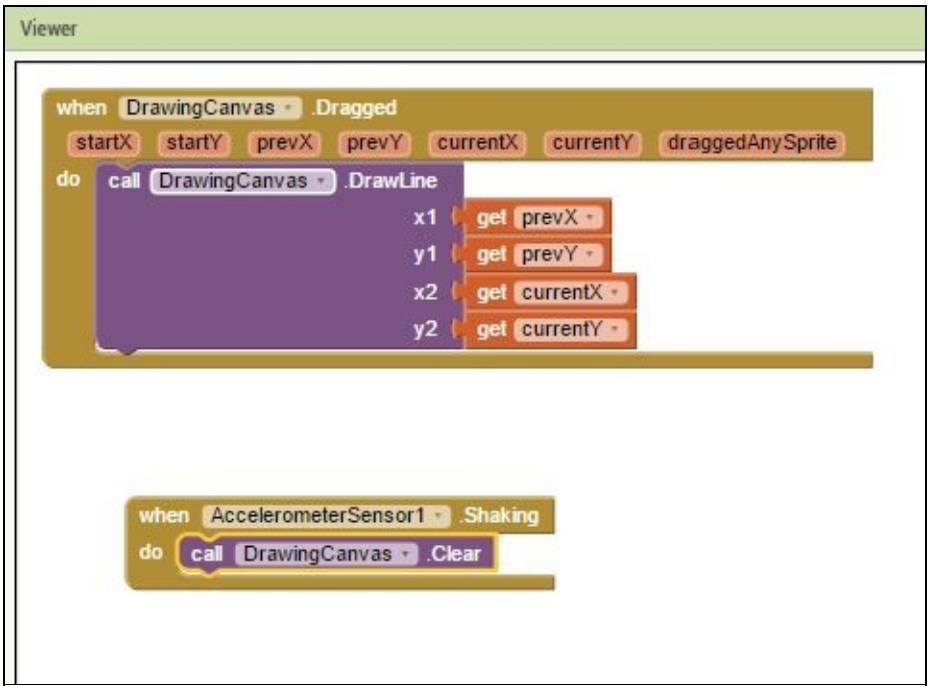
block. This block will identify shaking activity of your phone.



11. Now click on DrawingCanvas and select 

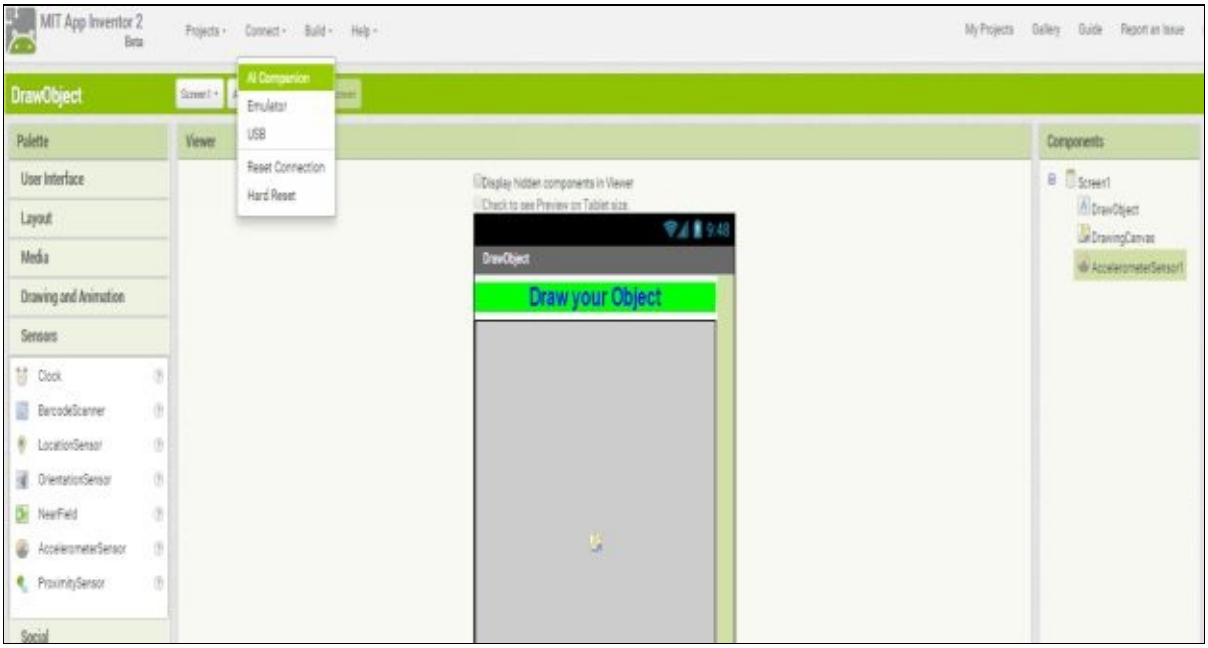
block.

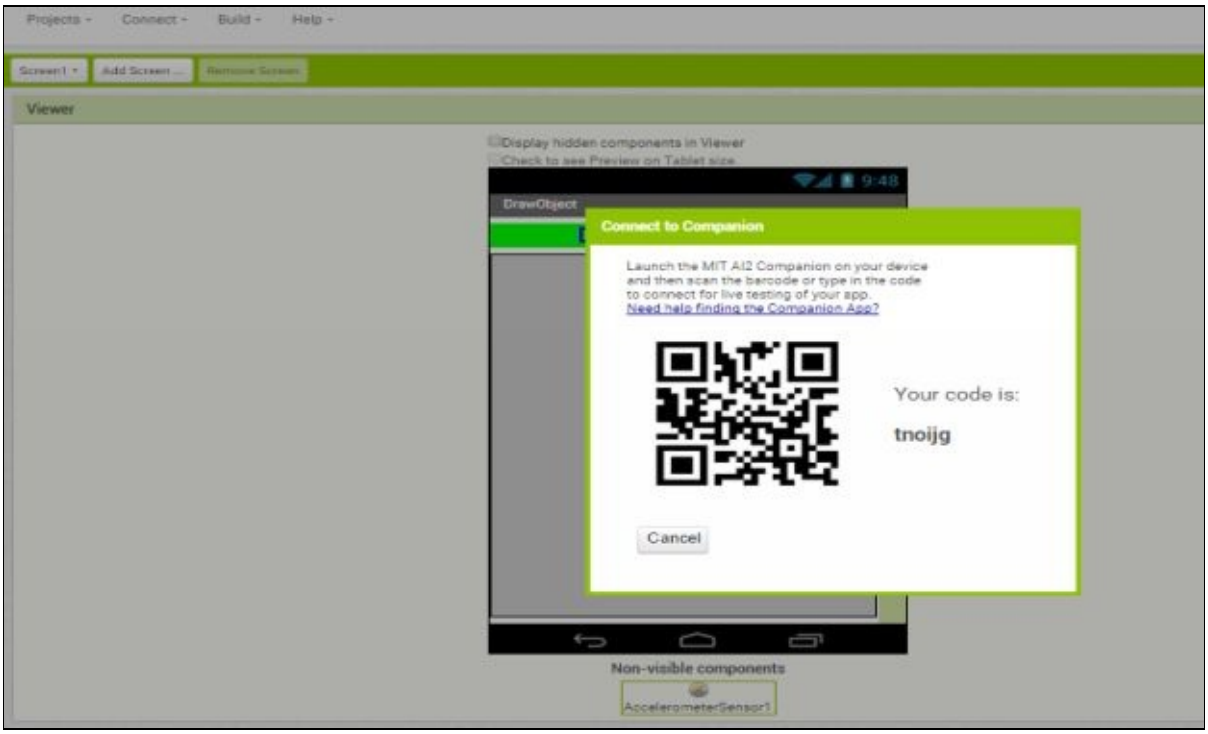
12. Now attach blocks as shown below. So when you shake your phone Canvas should be cleared.



This block will clear our canvas when we shake our phone.

13. Now go to Designer and click on Connect and select AI Companion.

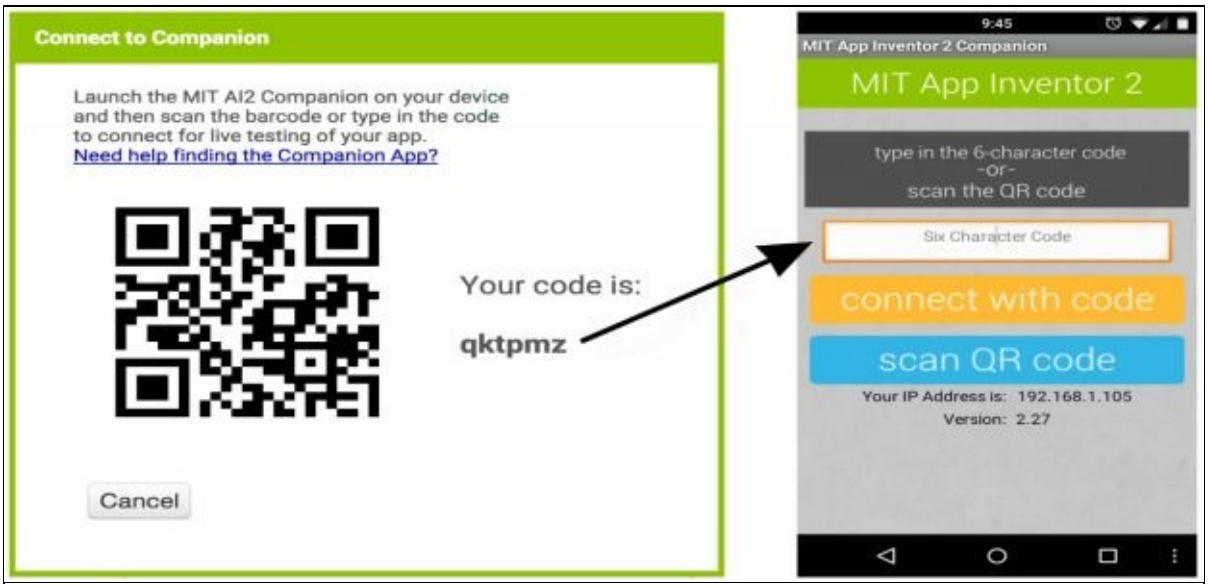




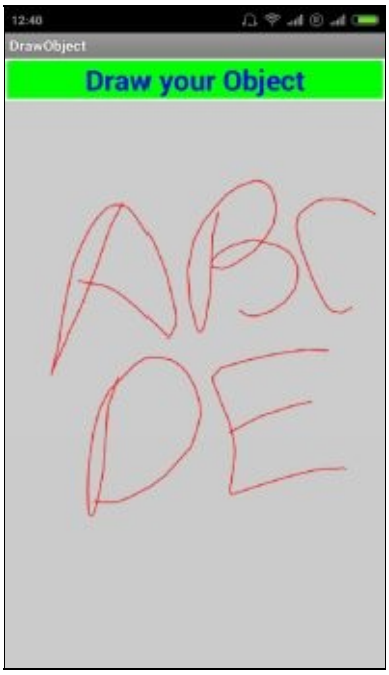
14. Now open MIT App Inventor 2 Companion in your mobile/Tablet.

And enter the code or scan the QR code from your mobile and click on connect with code.

Note:-Your Mobile/Tablet should be connected to same wi-fi/network to which your computer/Laptop is connected.



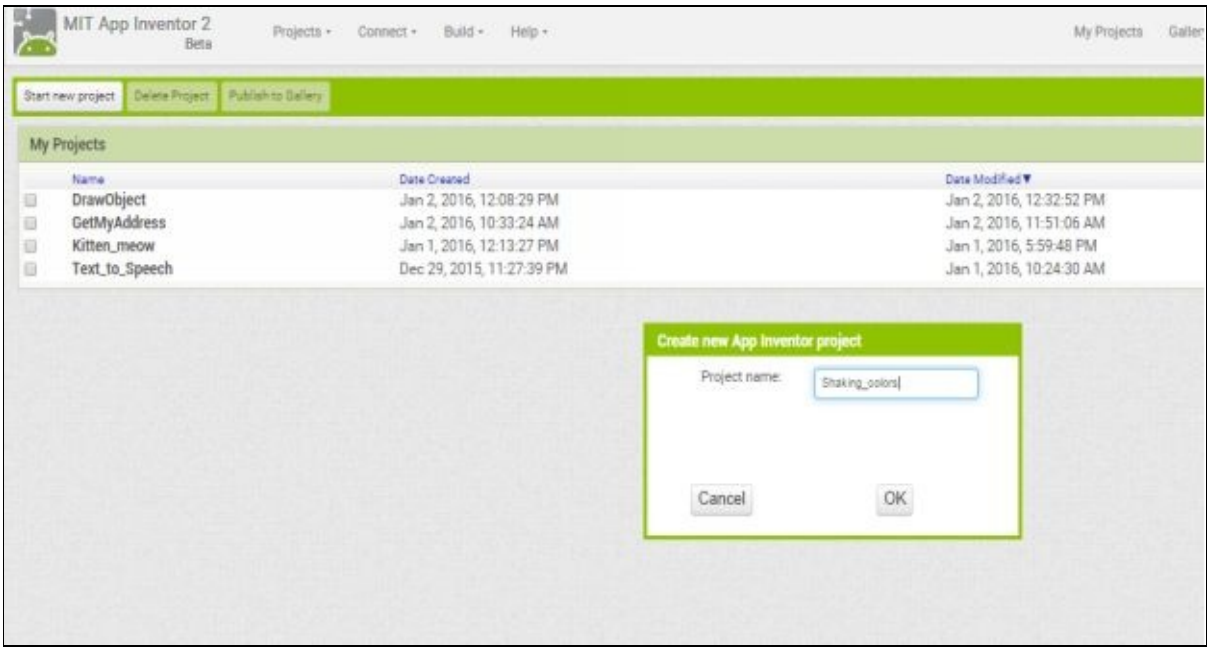
15. Now you should see your app in your phone for live testing. Draw anything on canvas by touching screen and dragging it to draw your object. And shake your phone to clear your canvas.



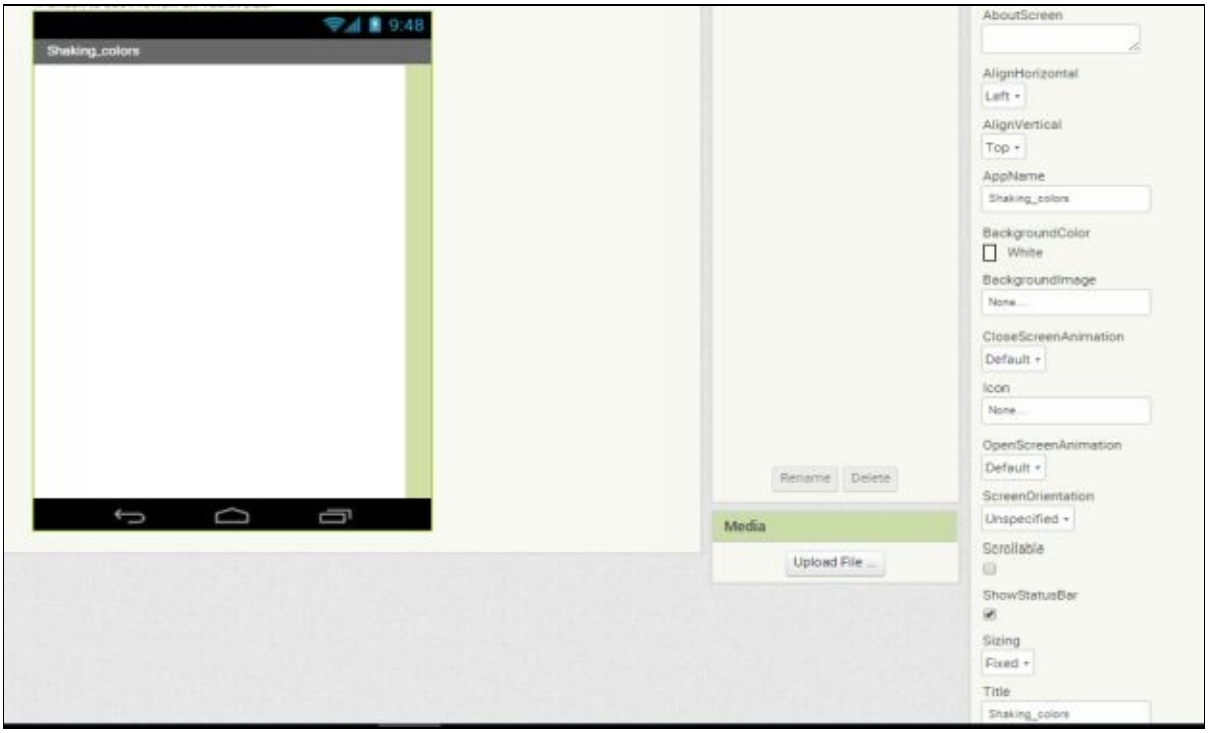
Shake your phone to clear your canvas !!!

Chapter 6: Shaking Colors App

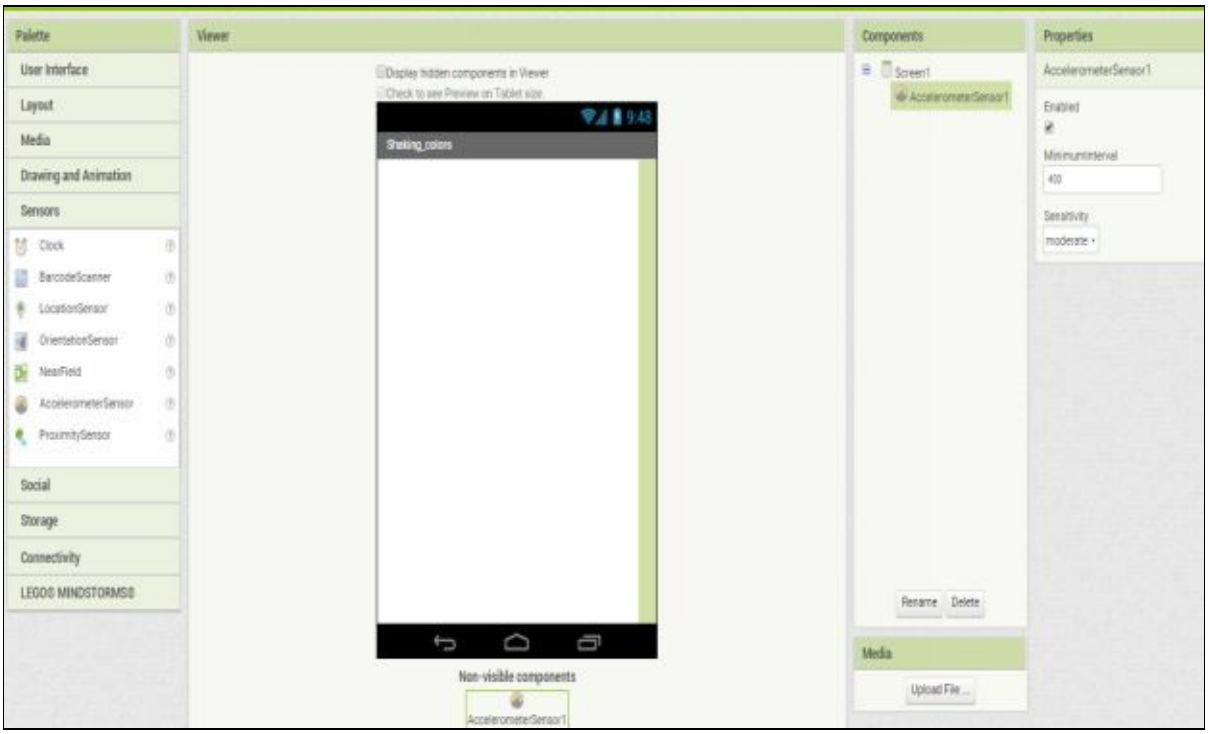
1. Click on Start new project and give it name Shaking_colors and click OK.



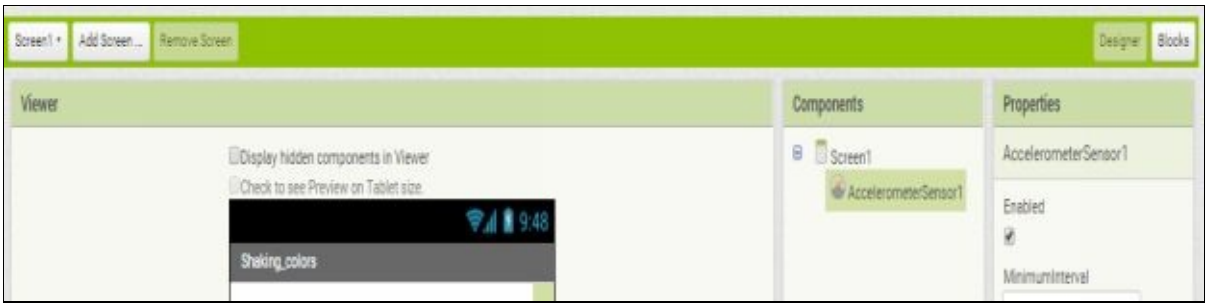
2. Change Screen1 Title to Shaking_colors.

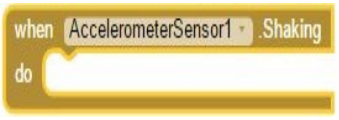


3. Drag an AcclerometerSensor component from Palette to Viewer screen.

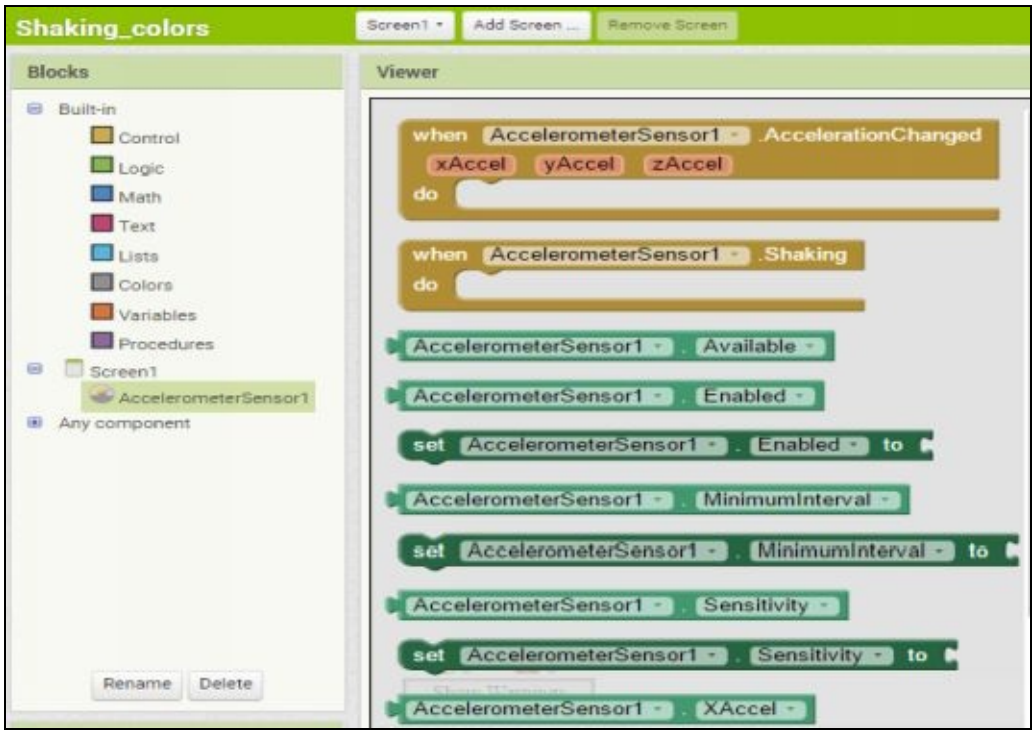


4. Now go to Blocks.



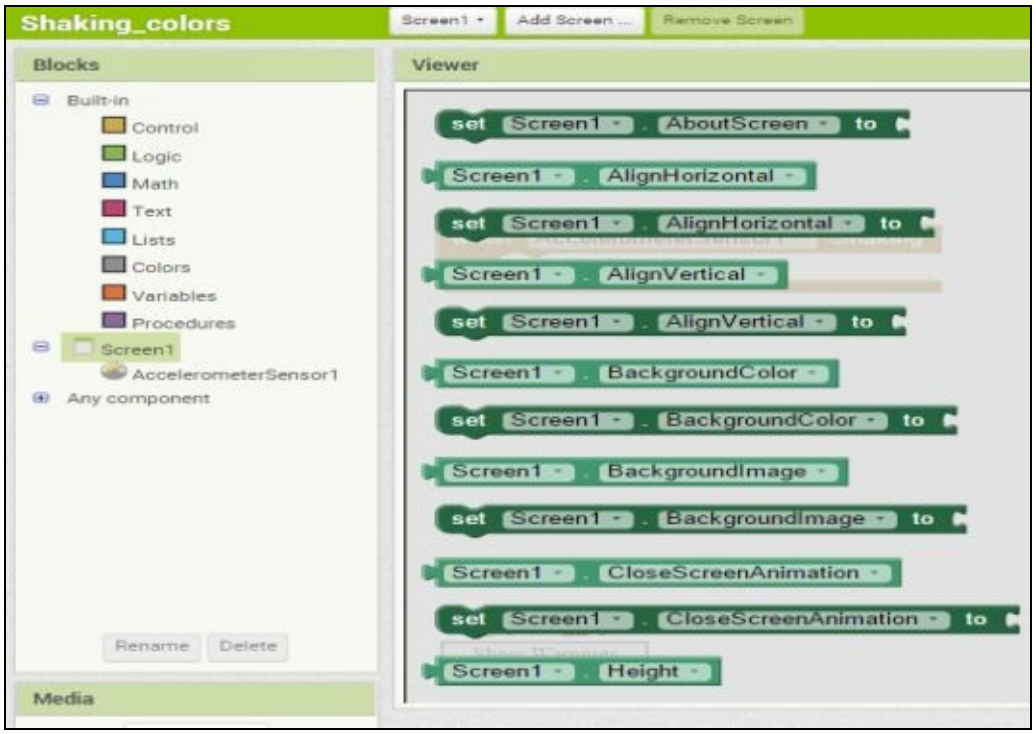
5. Click on AccelerometerSensor1 component on left side and select 

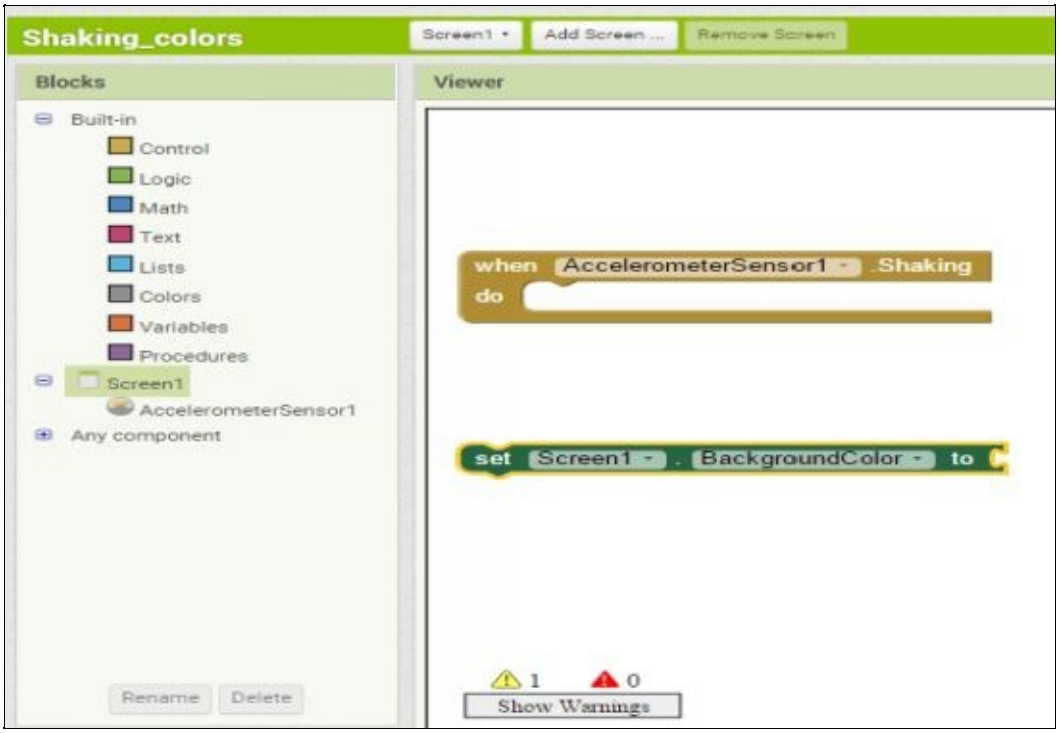
block. This block will identify shaking activity of your phone.



6. Now click on Screen1 and scroll down to select 

block.This block will set the background color of your screen.



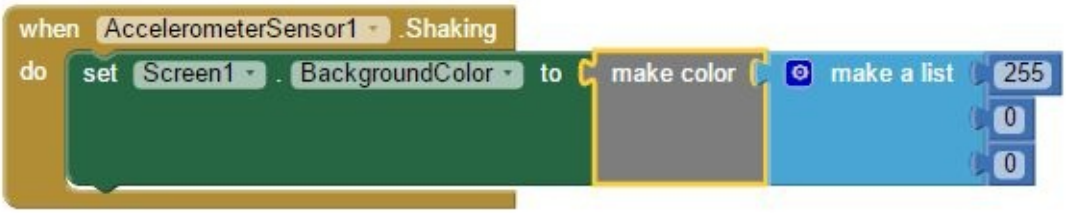


7. Now connect the blocks as shown below. So that when you shake your phone app screen color will change.

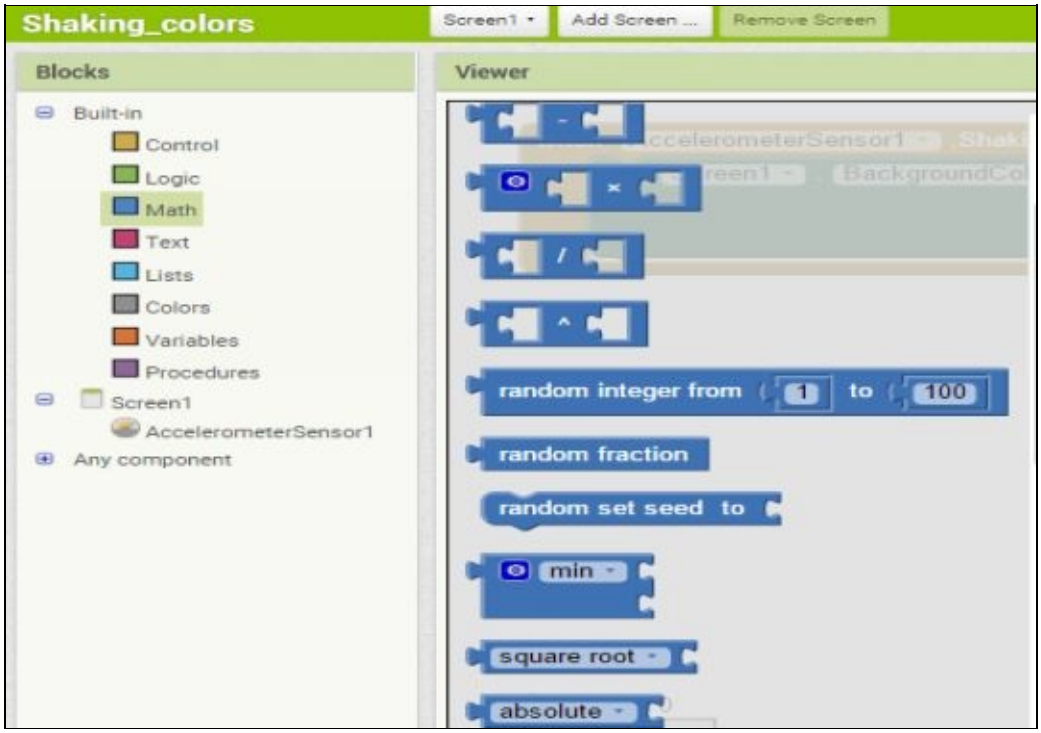
8. Now click on Colors under Built-in and scroll down to select

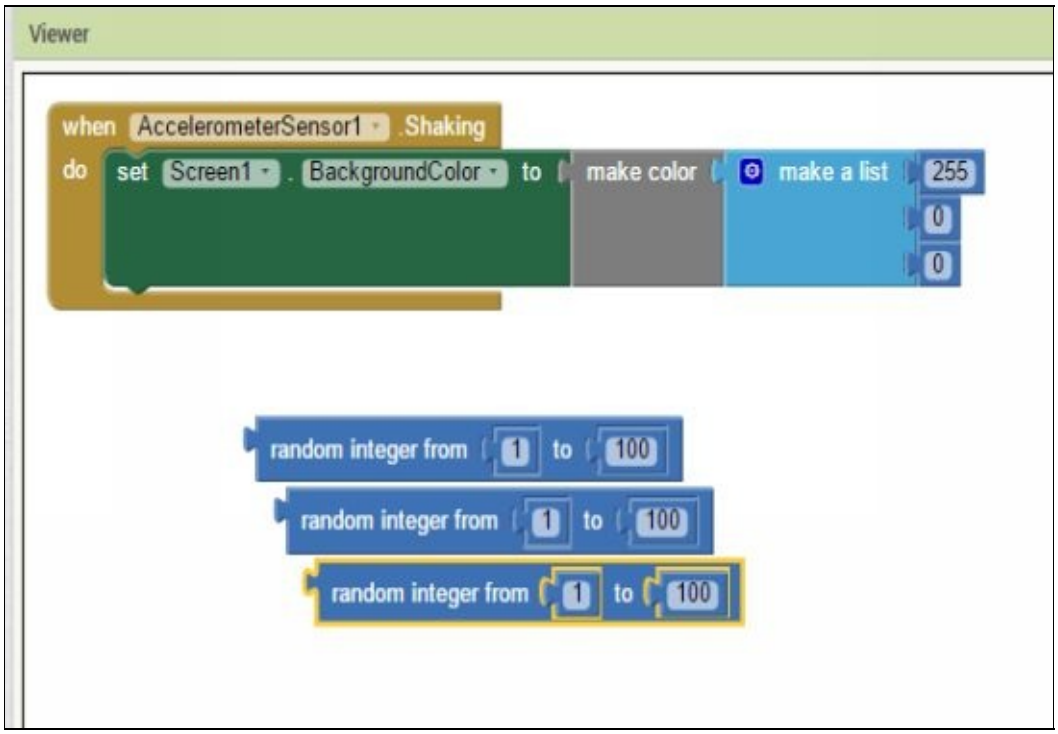
block. This block will define the color of the app screen to be changed. These 3 values(255,0,0) in this block will make up a color.

9. Now attach the blocks as shown below.So that when you shake your phone your app screen color will be changed.



10. Now click on Math under Built-in and select  block 3 times or you can copy paste the same block 3 times. This block will return an integer value between 1 and 100.

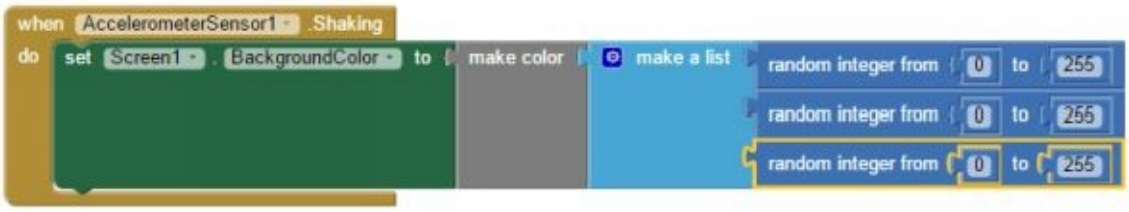




11. Now change the values for  blocks to 0 to 255. Because a color is made up of thse 3 values which range between 0 and 255.

12. Now attach blocks as shown below. So when you shake your phone your screen color will be changed randomly which will be defined by ‘make color’ block.

Note:-To delete a block, you can select that block and press delete button or drag that block to bin.

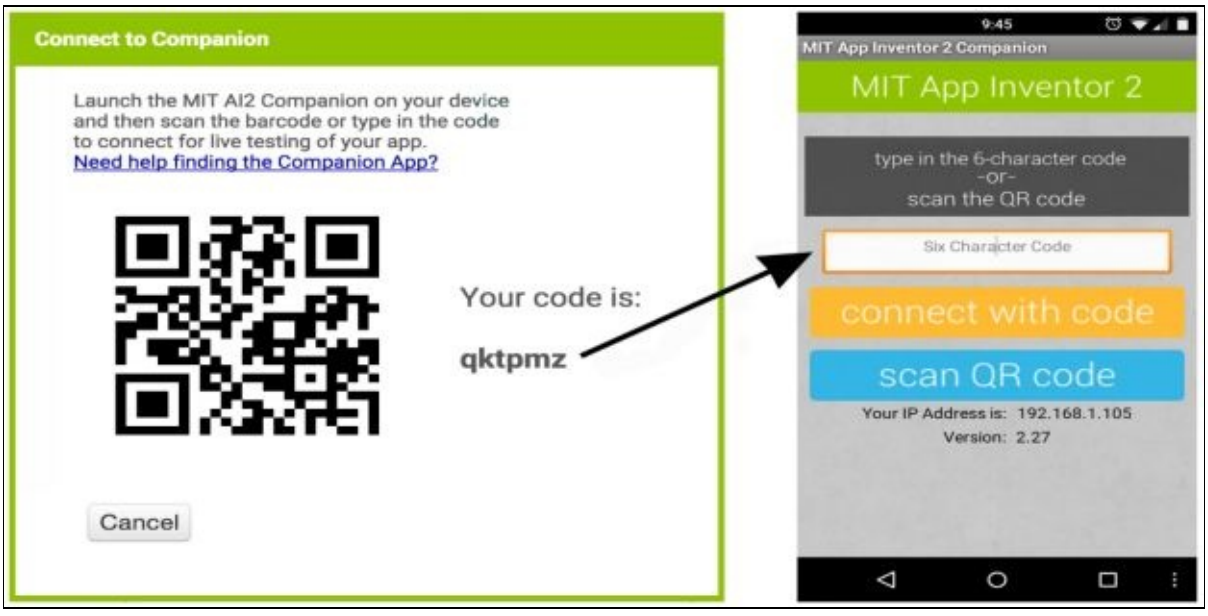


13. Now go to Designer and click on Connect and select AI Companion.

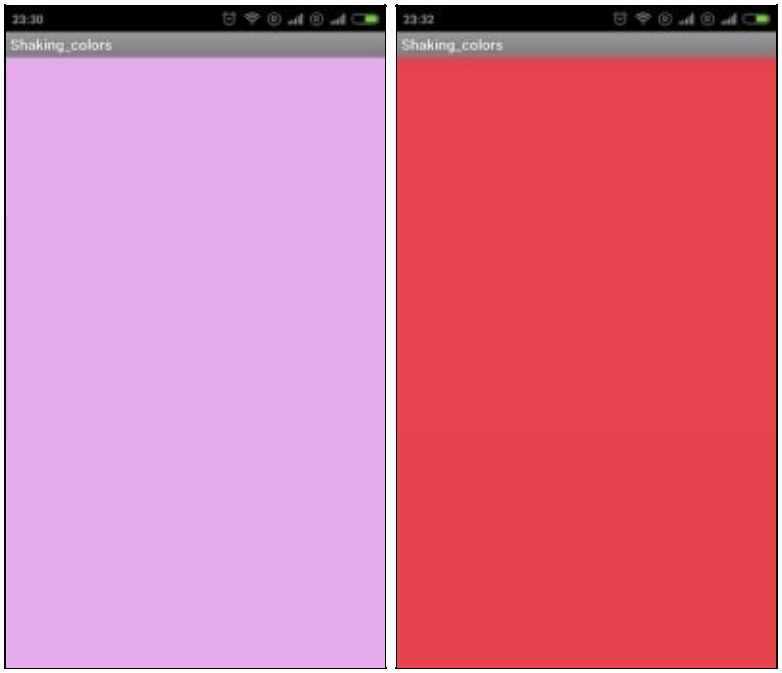
14 Now open MIT App Inventor 2 Companion in your mobile/Tablet.

And enter the code or scan the QR code from your mobile and click on connect with code.

Note:-Your Mobile/Tablet should be connected to same wi-fi/network to which your computer/Laptop is connected.

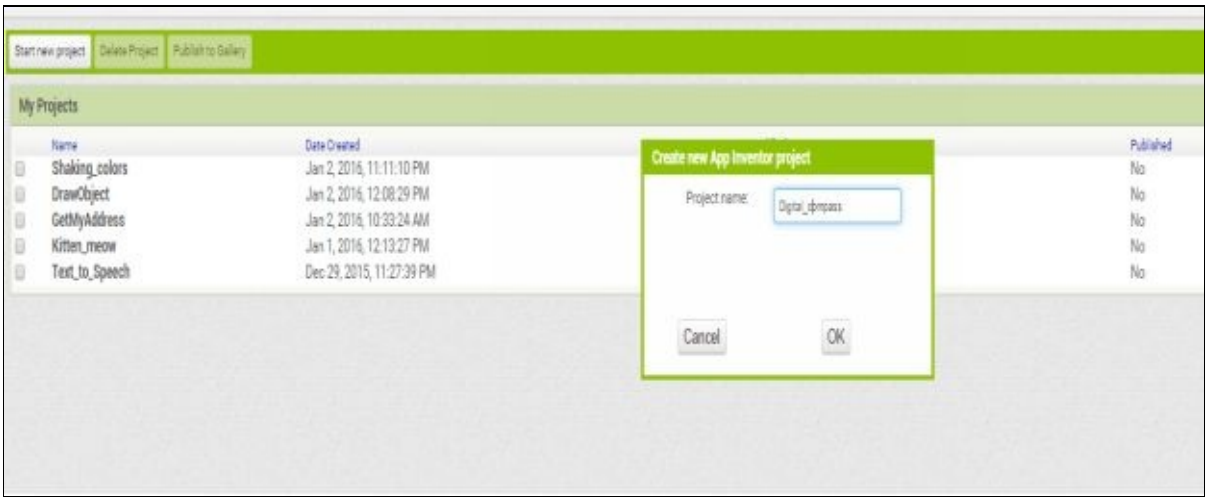


15. Now you should see your app in your phone for live testing. Shake your phone to change screen colors.

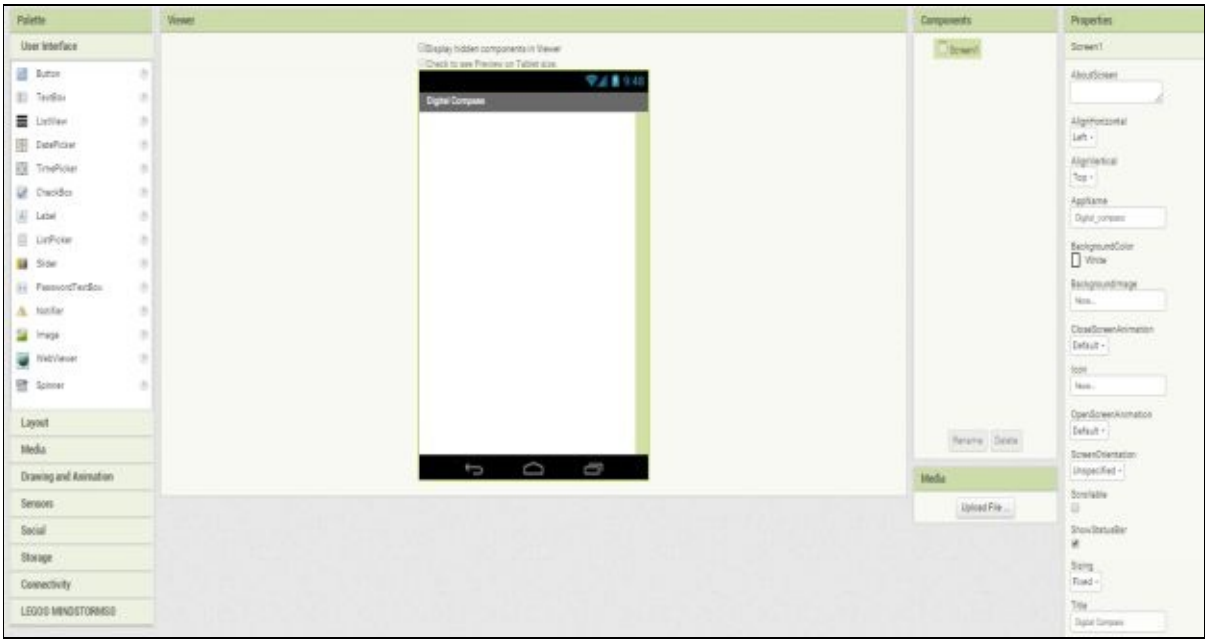


Chapter 7: Digital Compass App

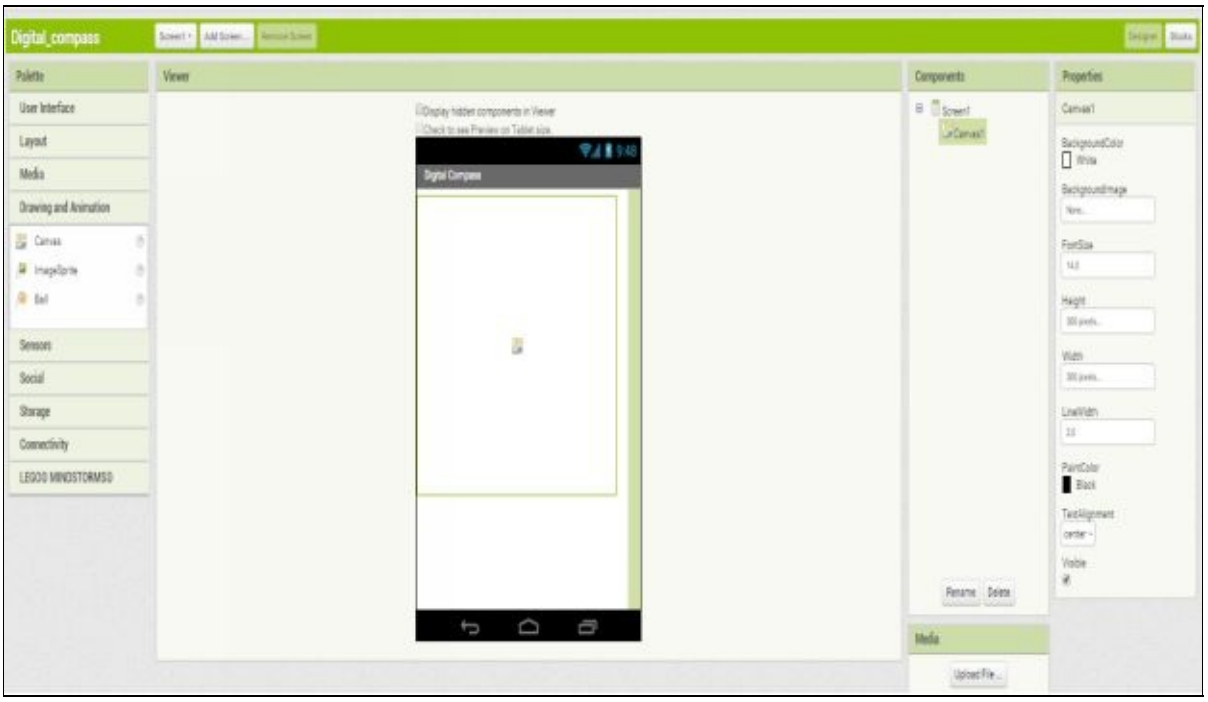
1. Click on Start new project and give it name Digital_compass and click OK.



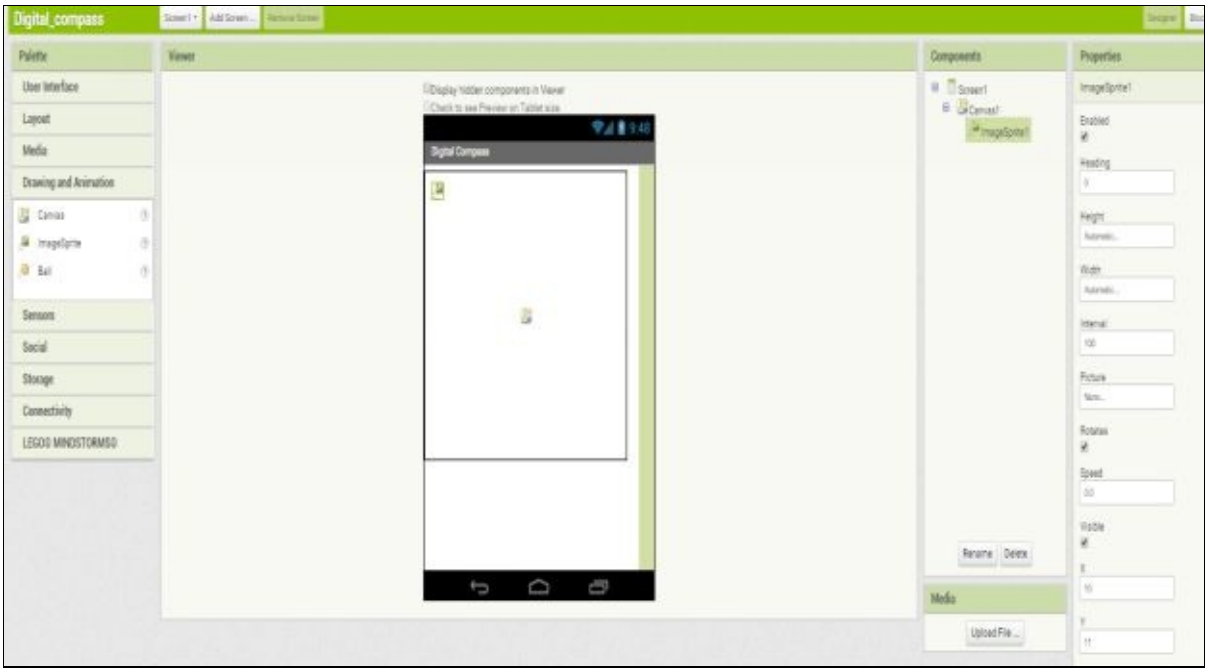
2. Change the Screen1 Title to Digital Compass.

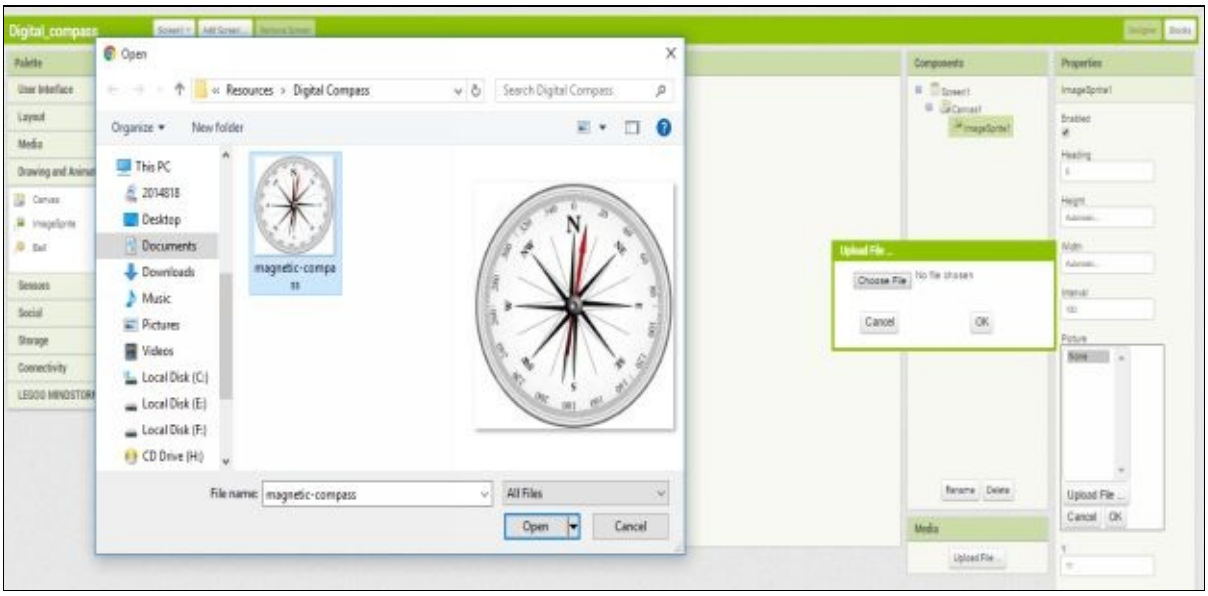


3. Now Drag a Canvas to the Viewer screen and set its Height and Width to 300 pixels for both.

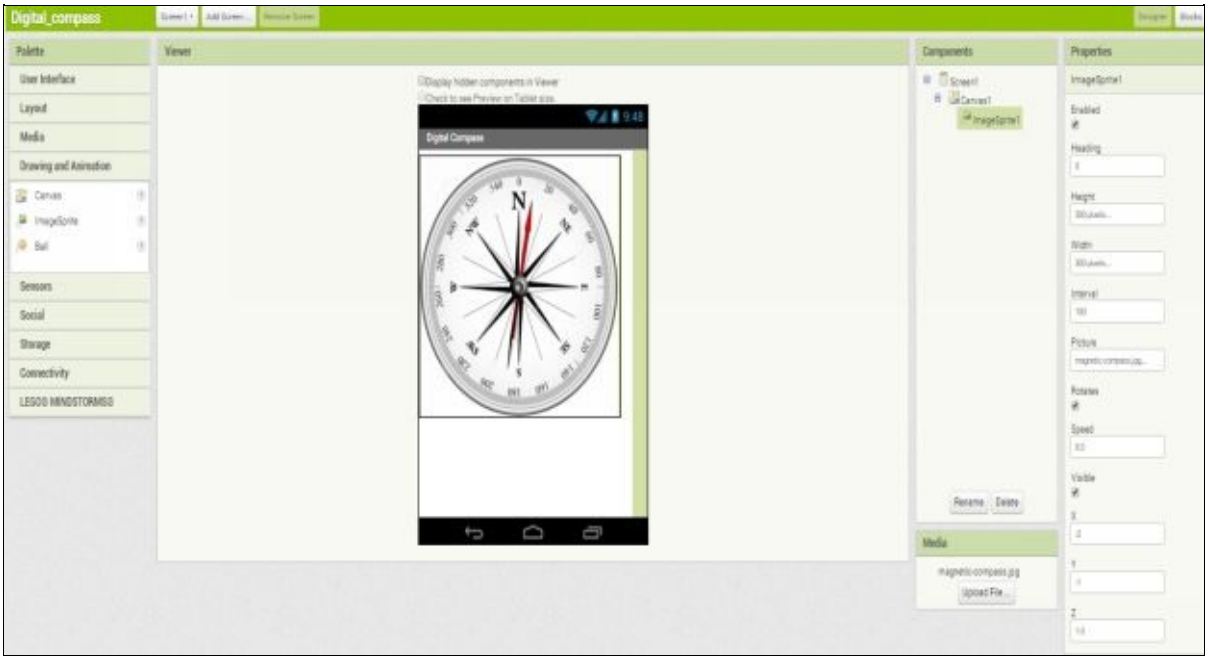


4. Now Drag an ImageSprite inside the canvas on Viewer screen and set Image to be magnetic-compass image. This image represents directions.

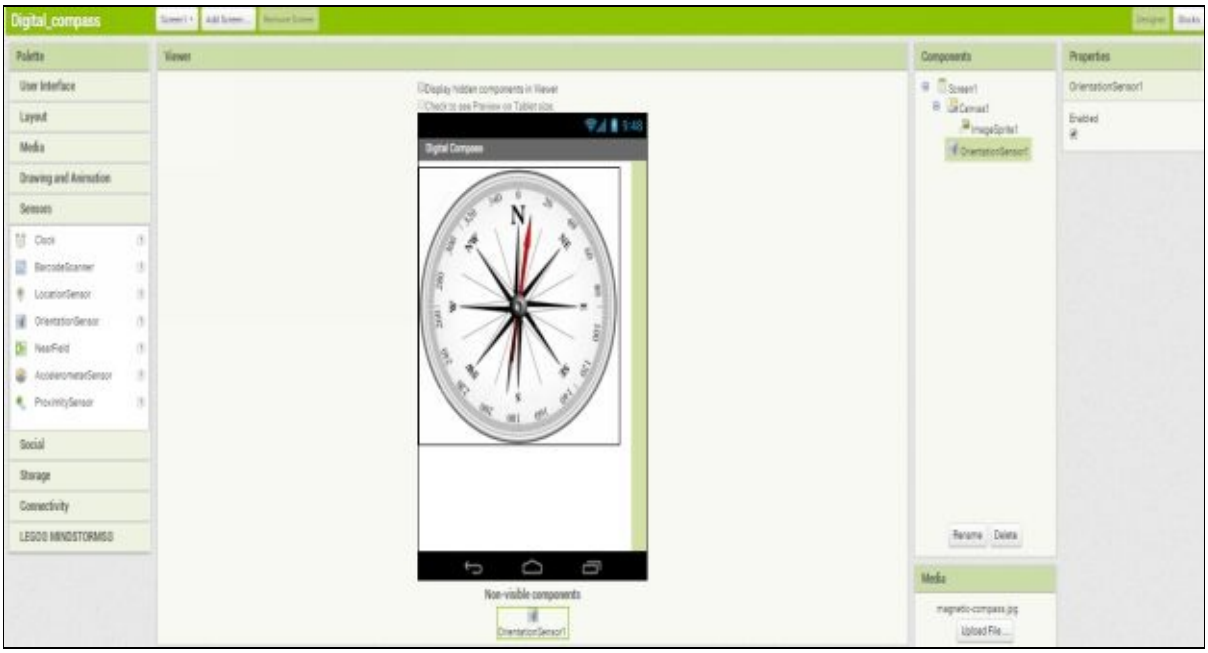




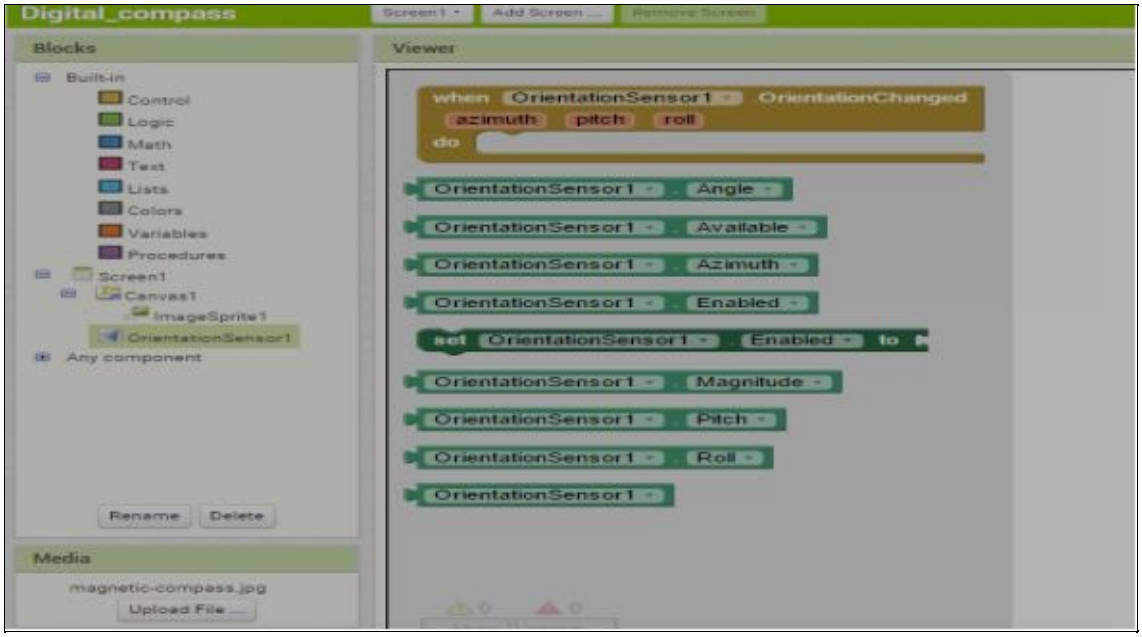
5. Now set ImageSprite1 Image Height to 300 pixels and Width to 300 pixels and adjust the Image on Canvas to fit it.

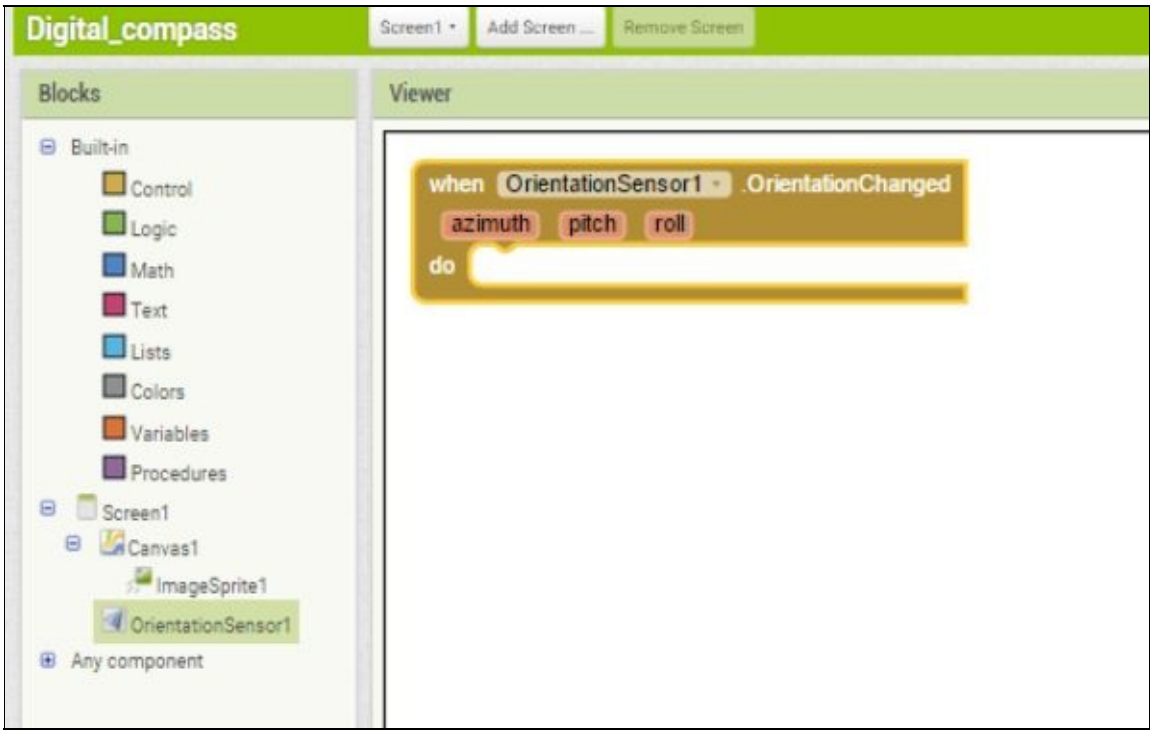


6. Now drag an OrientationSensor from Palette to Viewer screen. This is a non-visible component.



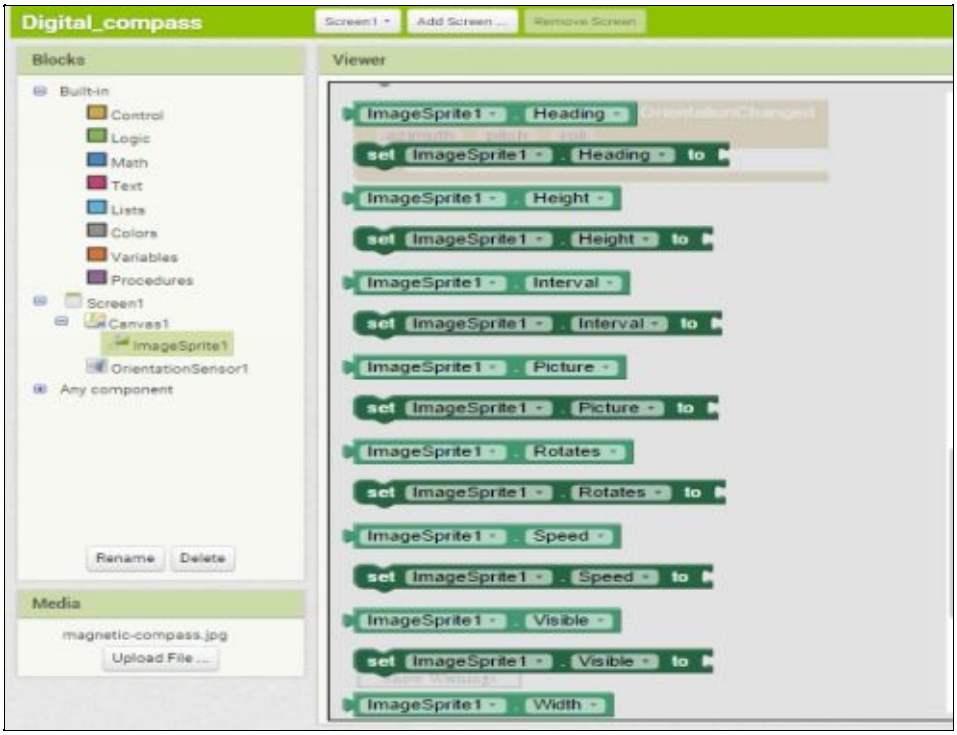
7. Now go to Blocks and click on OrientationSensor1 on left side and select block. This block will decide what your app should do when your phone's orientation is changed.



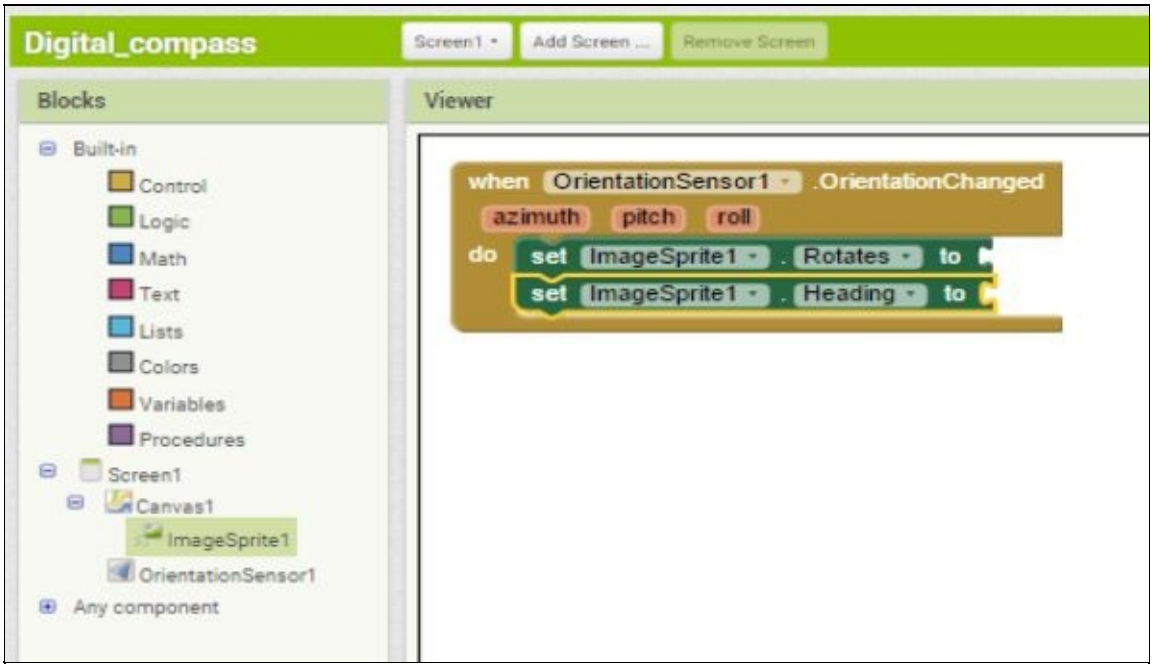


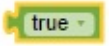
8. Now click on ImageSprite1 and scroll down to select 

blocks. These blocks will help image to rotate and pint to a particular direction.

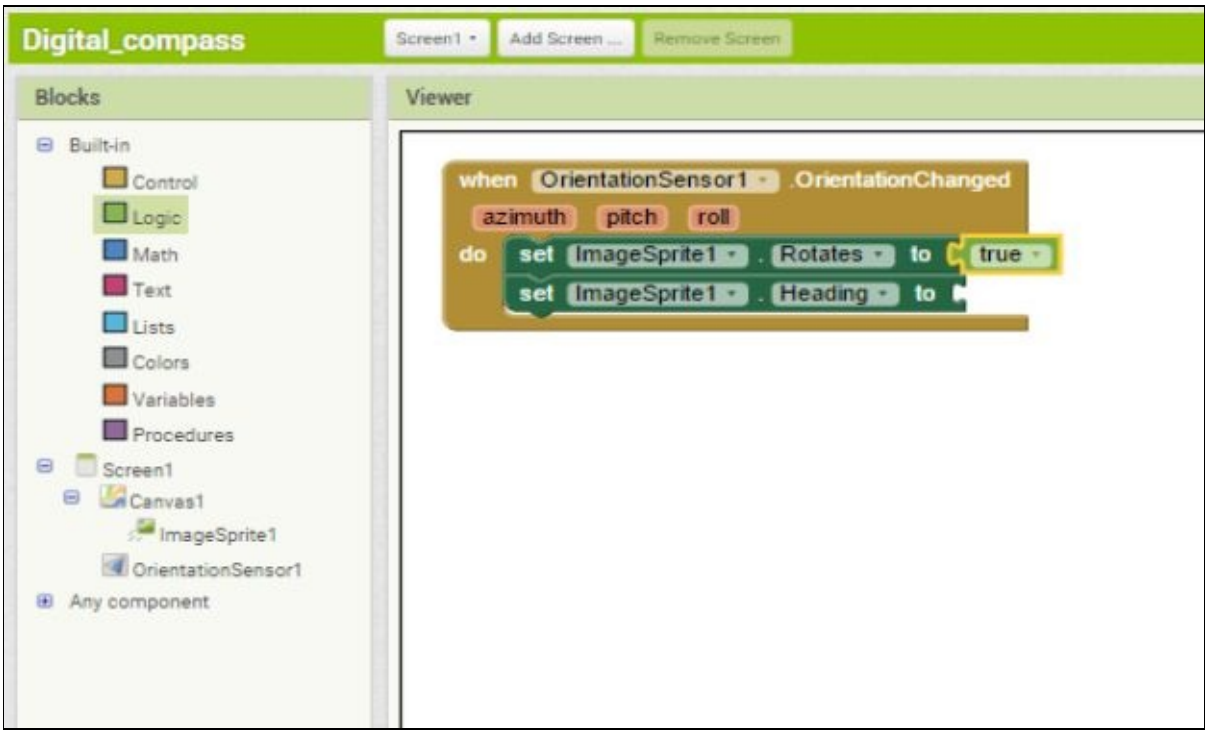


9. Now connect the blocks as shown below. So when You change your Phone's orientation your compass will rotate to a direction.



10. Now click on Logic under Built-in and select 

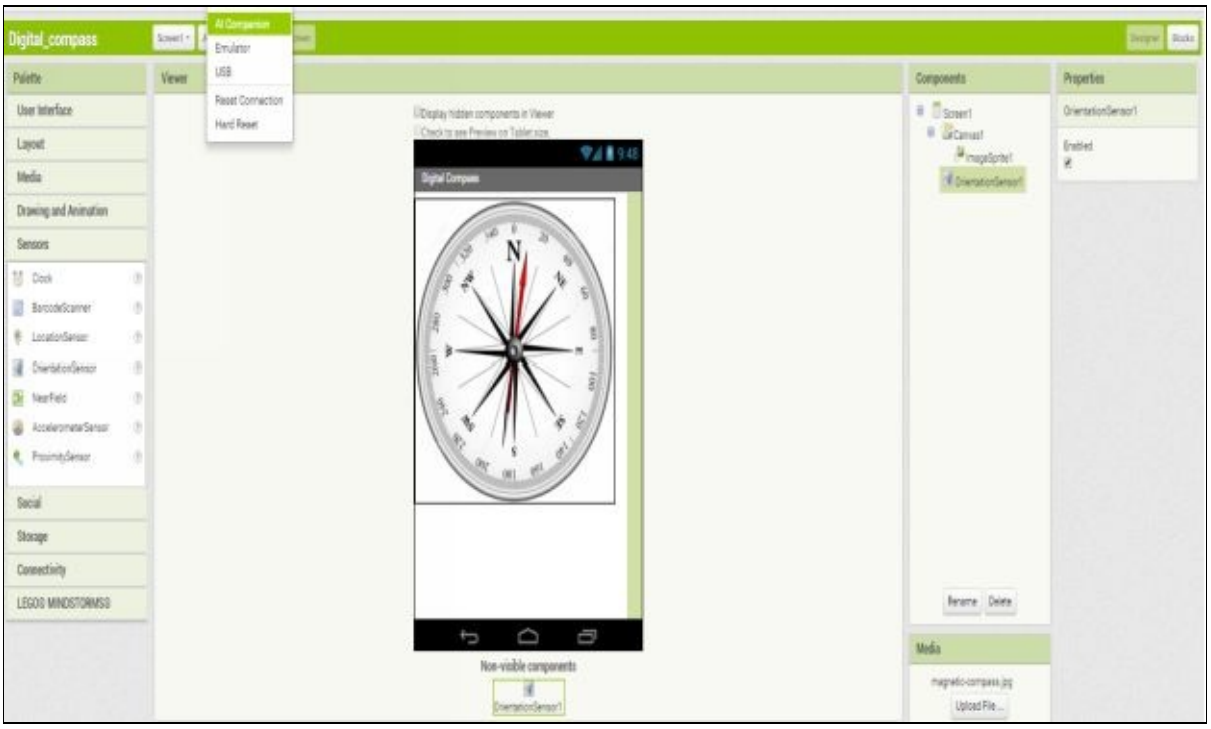
block and attach it like shown below. This will enable compass to rotate.



11. Now mouse over on azimuth and click on  block.

Attach it to existing block as shown below. This block will get the direction to which compass has to rotate.

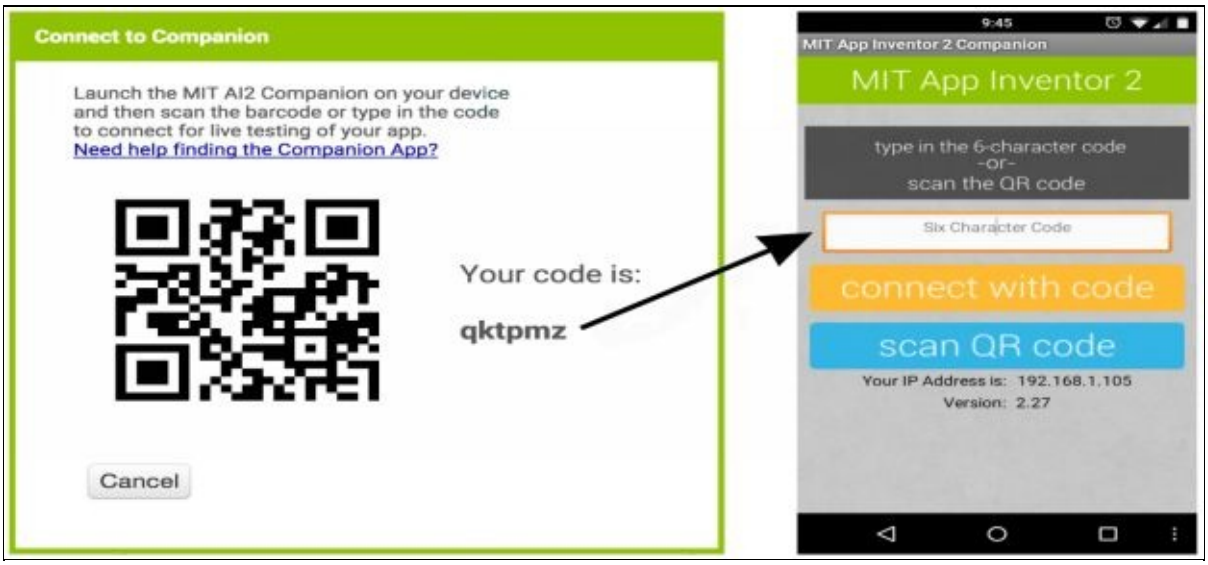
12. Now Go to Designer and click on Connect and select AI Companion.



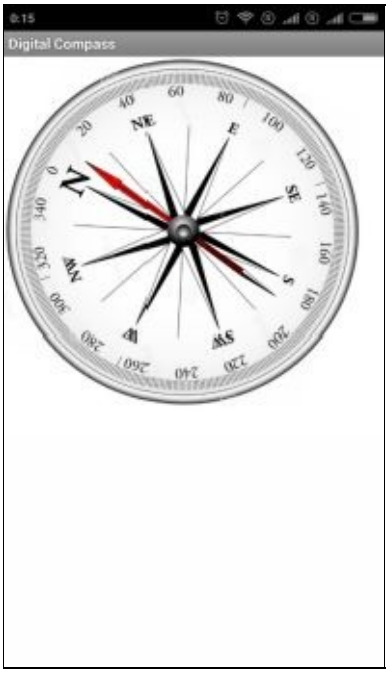
13. Now open MIT App Inventor 2 Companion in your mobile/Tablet.

And enter the code or scan the QR code from your mobile and click on connect with code.

Note:-Your Mobile/Tablet should be connected to same wi-fi/network to which your computer/Laptop is connected.

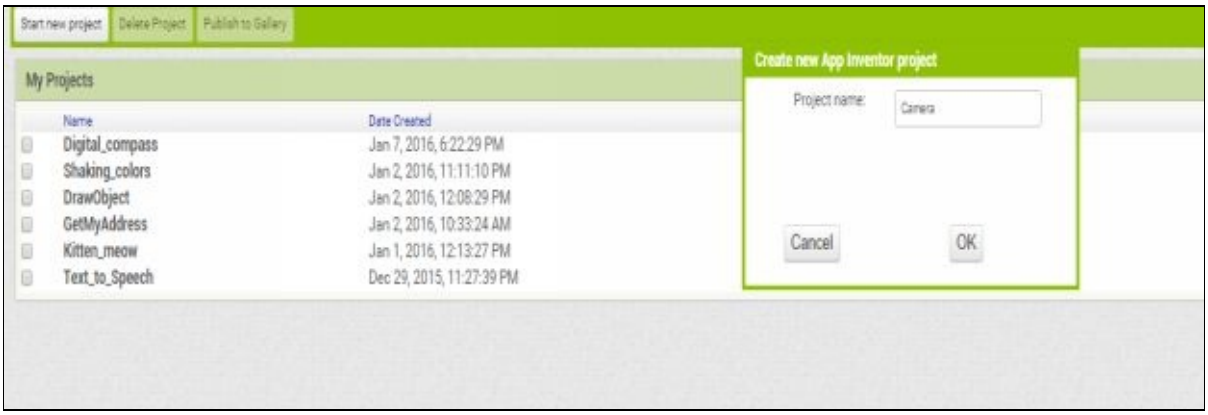


14. Now you should see your app in your phone for live testing. Now move your phone to get the directions.

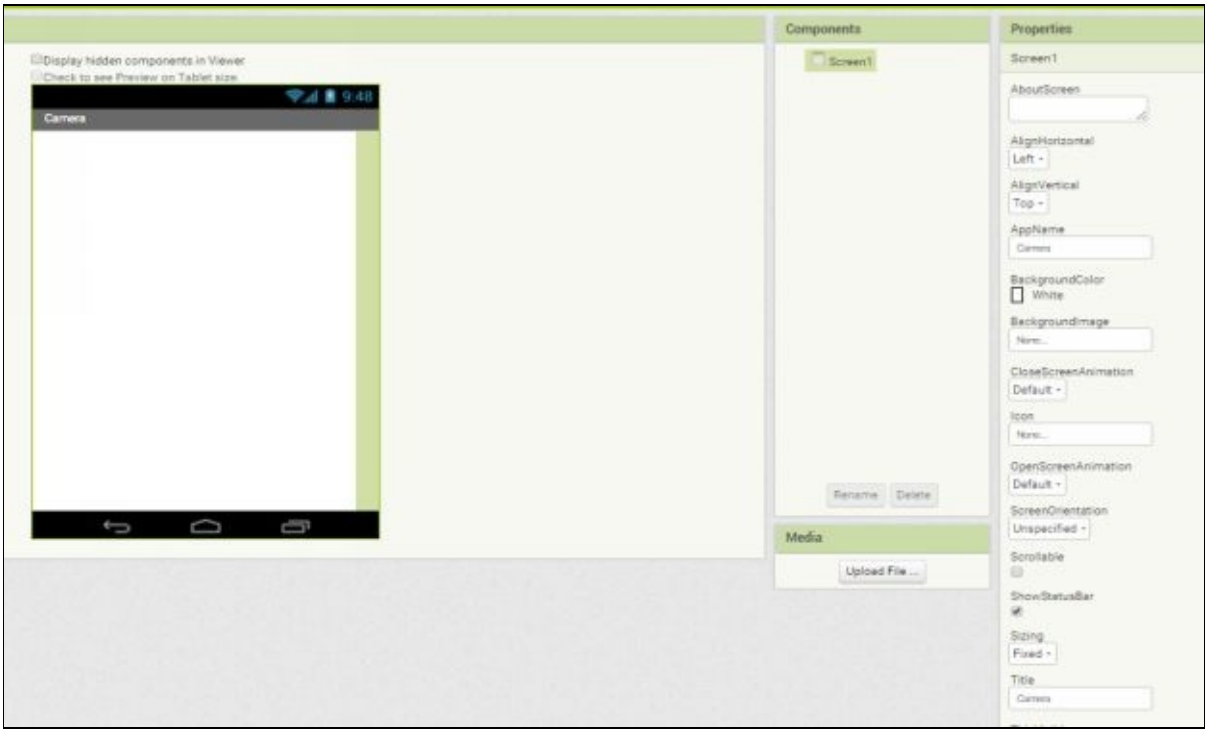


Chapter 8: Camera App

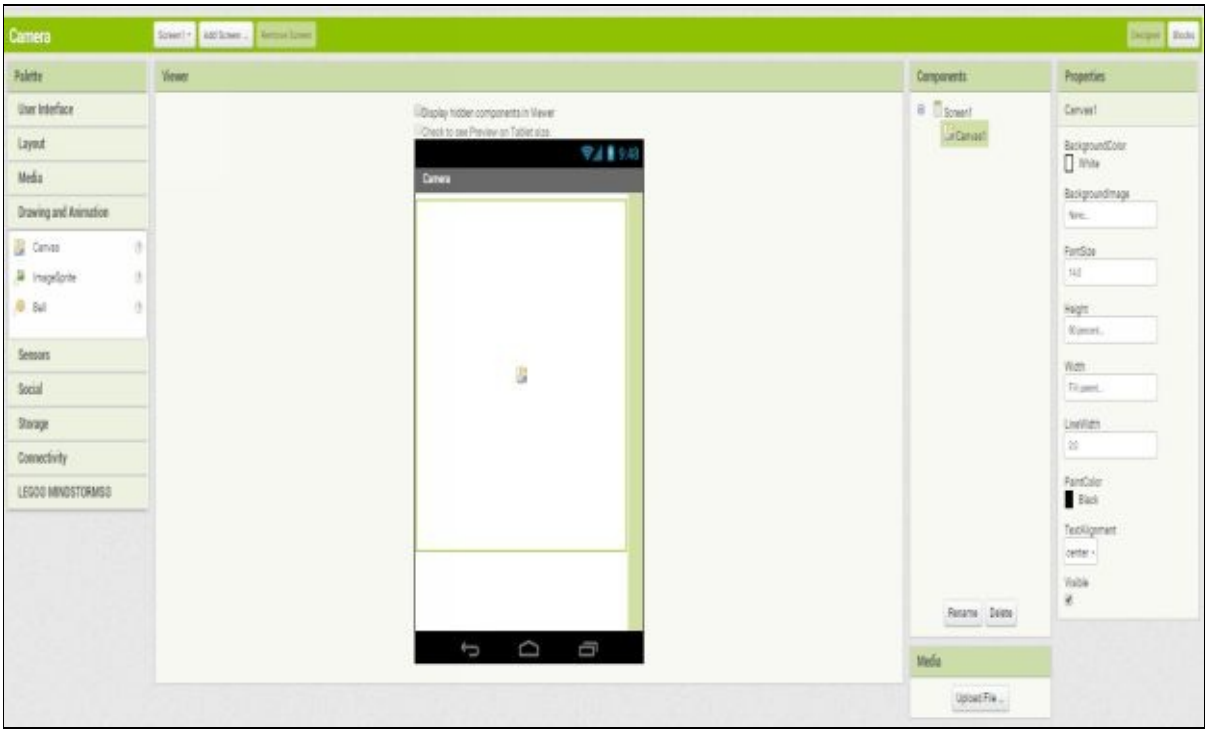
1. Click on Start new project and give it name Camera and click OK.



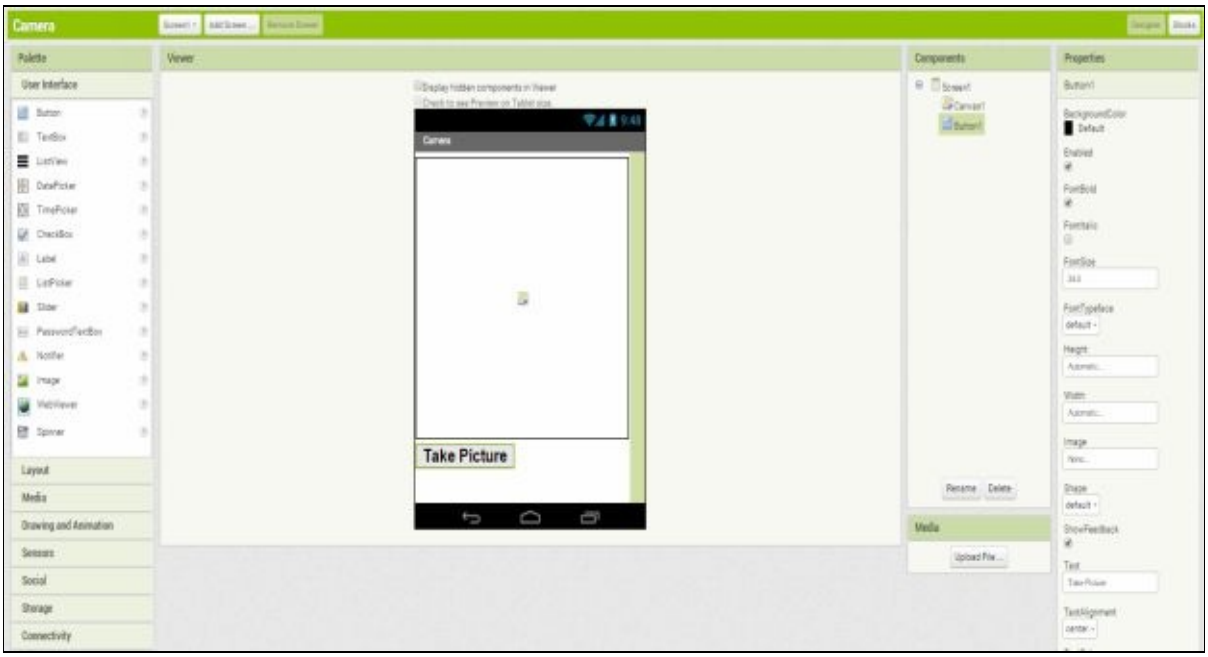
2. Change Screen1 Title to Camera.



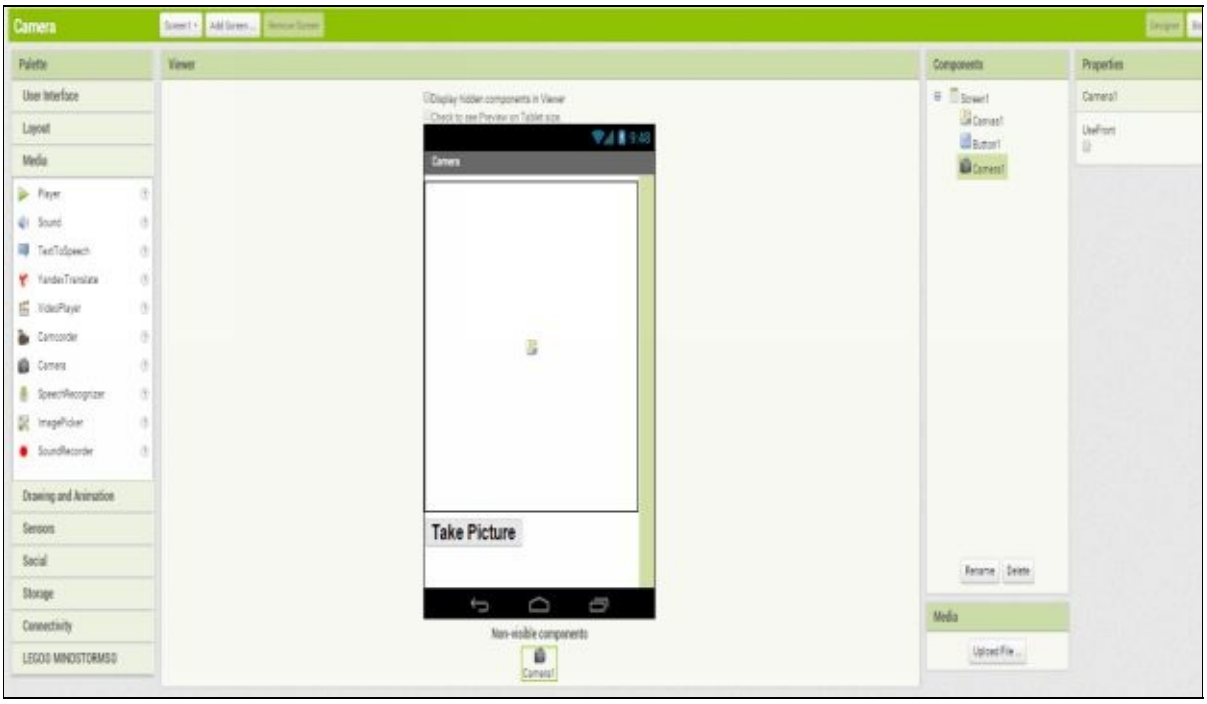
3. Now drag a Canvas from Palette to Viewer screen and set its Height to 80 percent and Width to Fill parent.



4. Now drag a Button from Palette to Viewer screen and set its Properties as shown below.



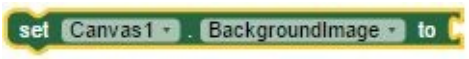
5. Now drag a Camera component from Palette to Viewer screen.



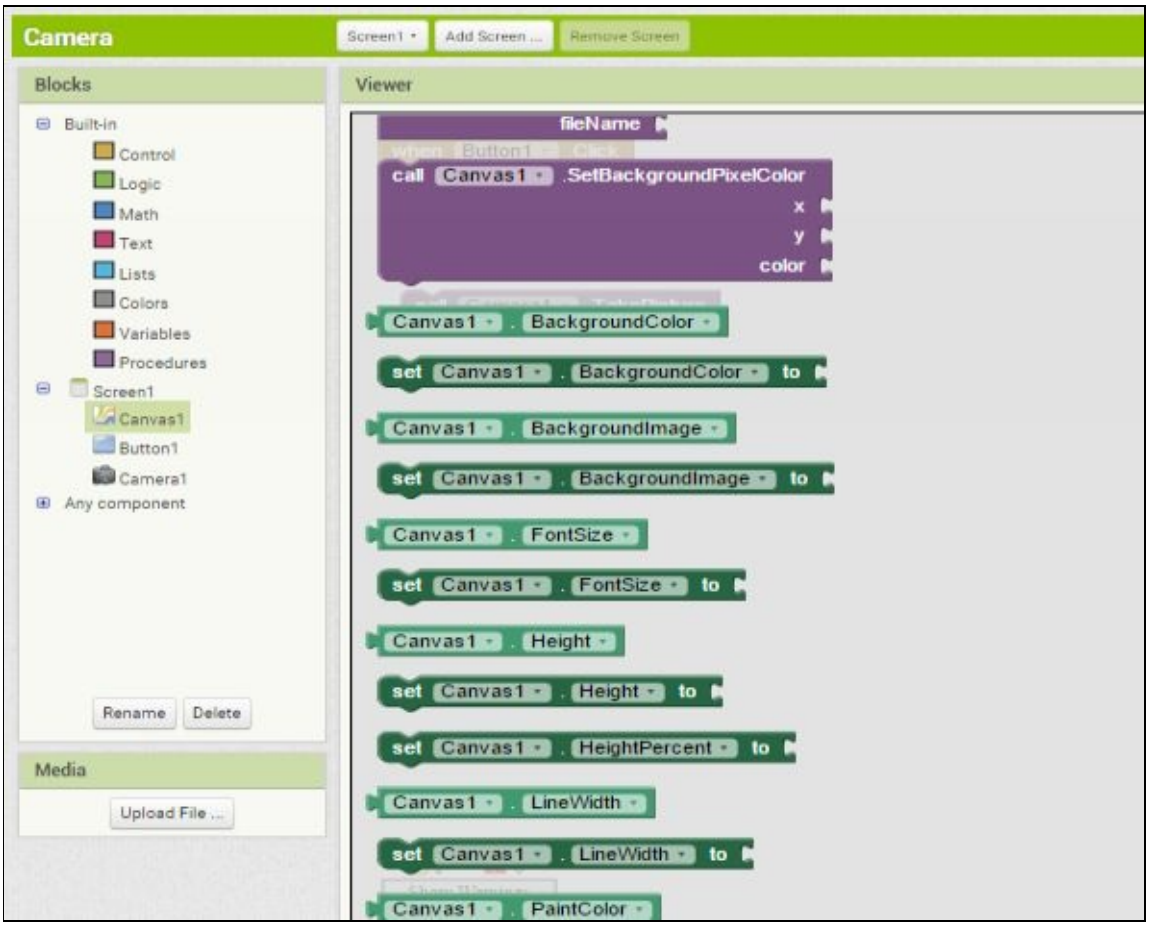
6. Now go to Blocks and click on Button1 on left side and select  block. This is an event block which will decide what your app will do when this button will be clicked. Here click is an event for Button1 component.

7. Now click on Camera1 on left side and select 

block.This block will open your device's camera and take picture.

8. Now click on Canvas1 and select 

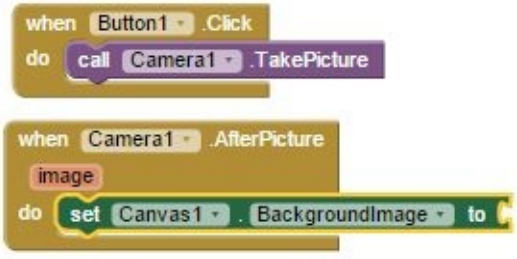
block. This block will set your canvas's background image.



9. Now click on Camera1 and select 

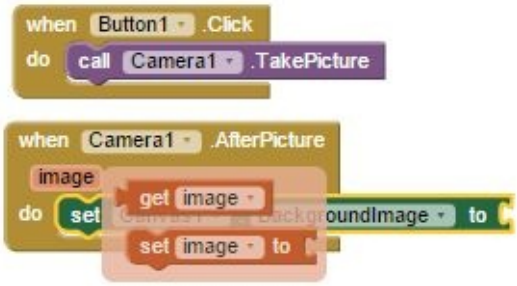
block. This is an event block which will decide what to do after taking the picture.

10. Now attach the blocks as show below. So that when you click Button1 , your device's camera will open and take picture.

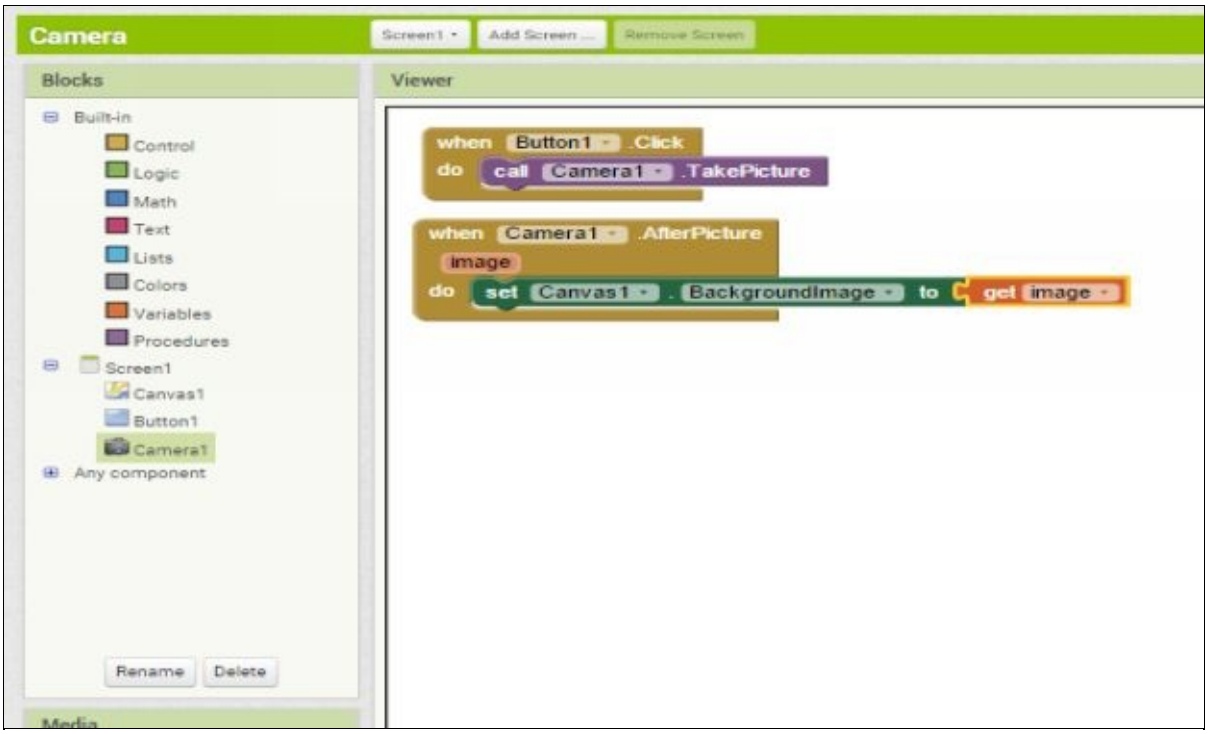


11. Now mouse over on image and click on  block. This block will represent

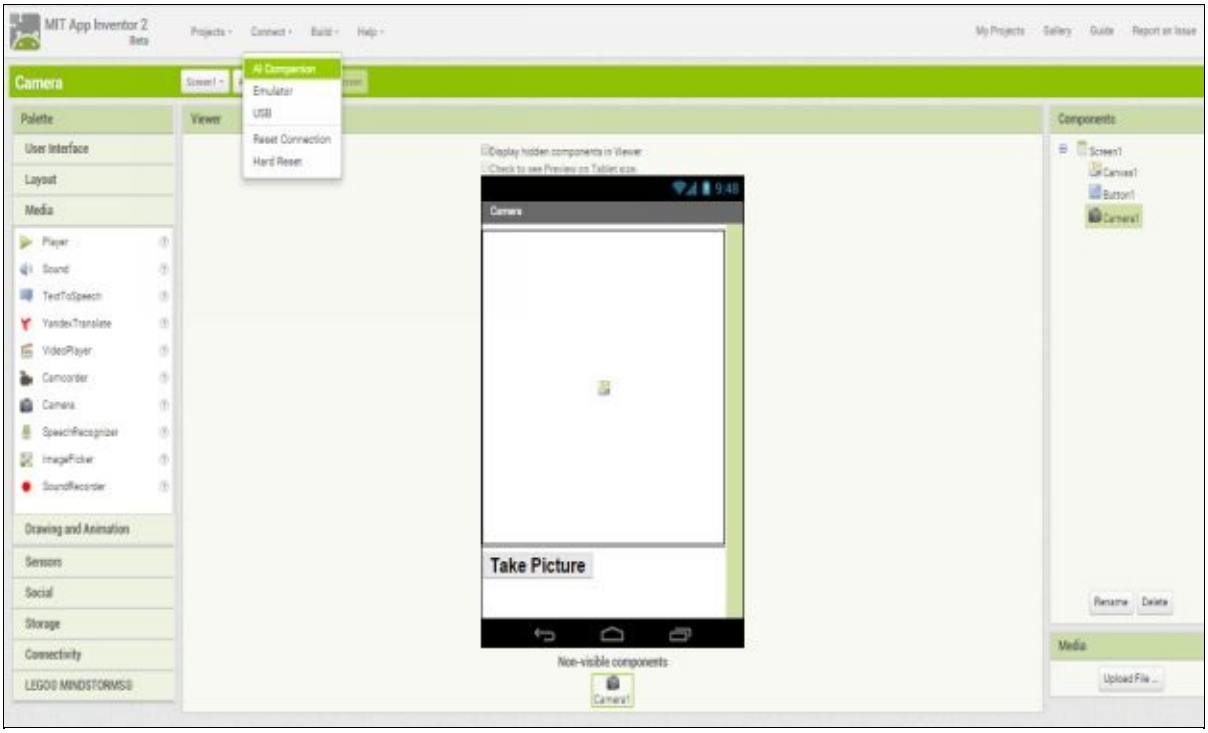
the image taken by your app camera.



12. Now attach the blocks as shown below. So After taking picture your app will set the image as canvas;s background image.



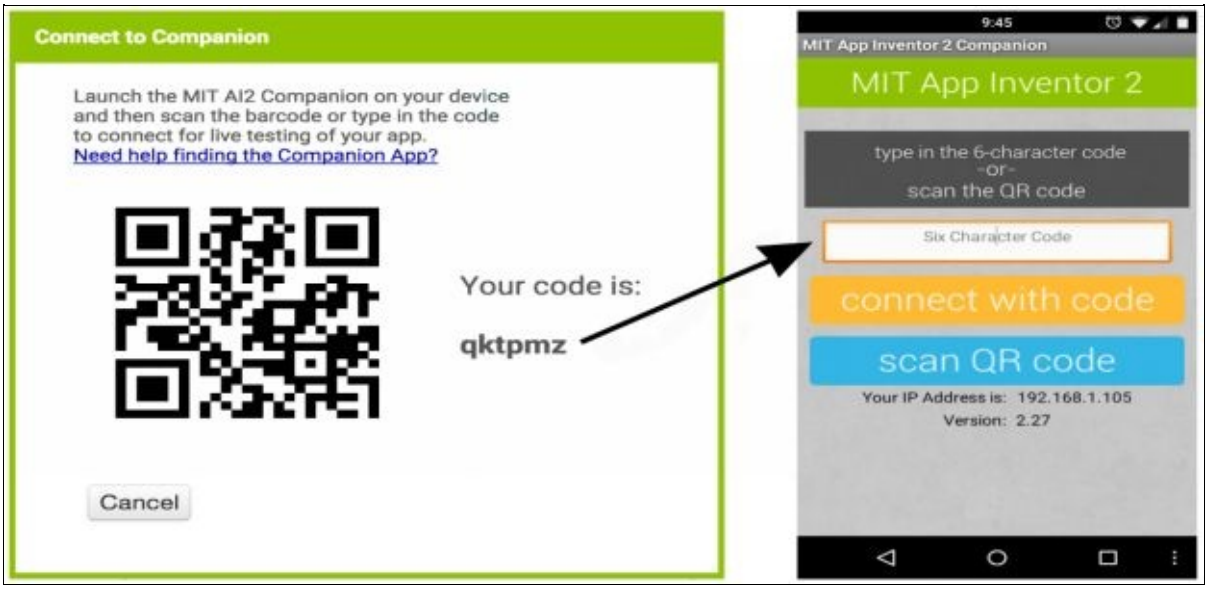
13. Now go to Designer and click on Connect and then AI Companion.



14. Now open MIT App Inventor 2 Companion in your mobile/Tablet.

And enter the code or scan the QR code from your mobile and click on connect with code.

Note:-Your Mobile/Tablet should be connected to same wi-fi/network to which your computer/Laptop is connected.

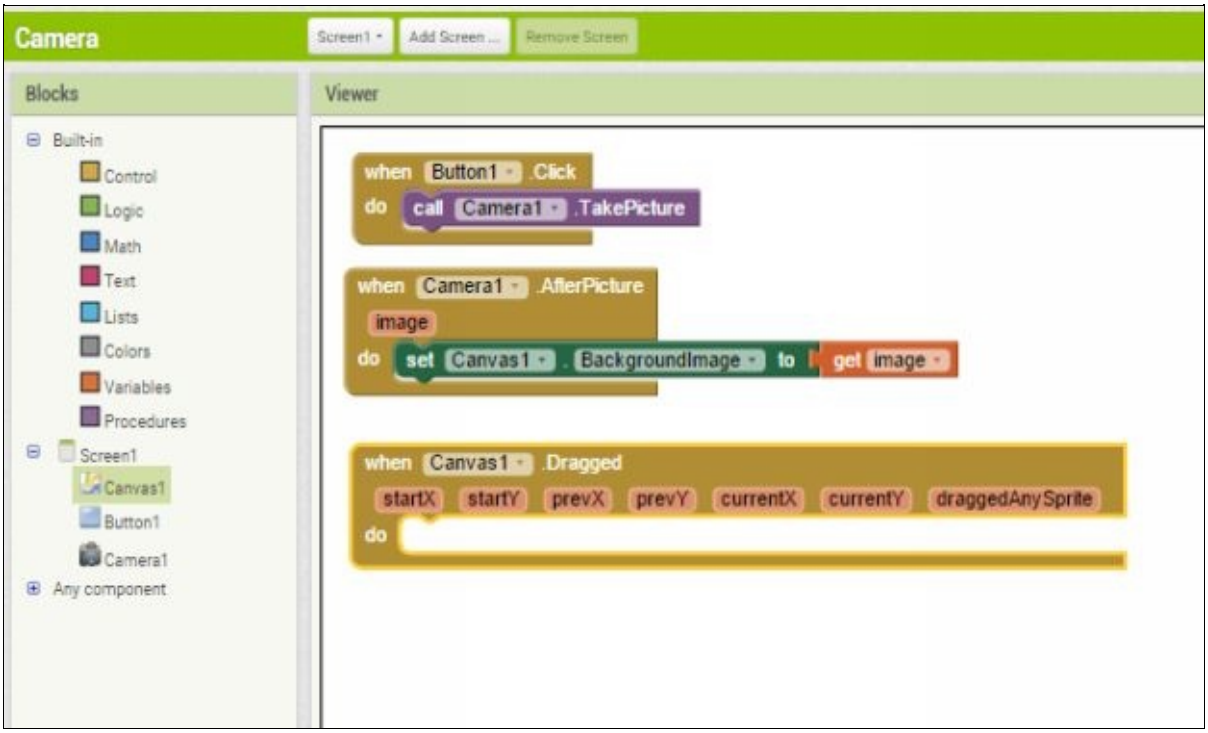


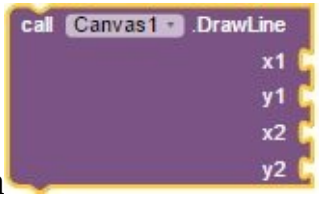
15. Now you should see your app in your phone for live testing. Now click on Take Picture Button and then your device Camera will open and take picture .Picture will be shown in the canvas that you have taken.

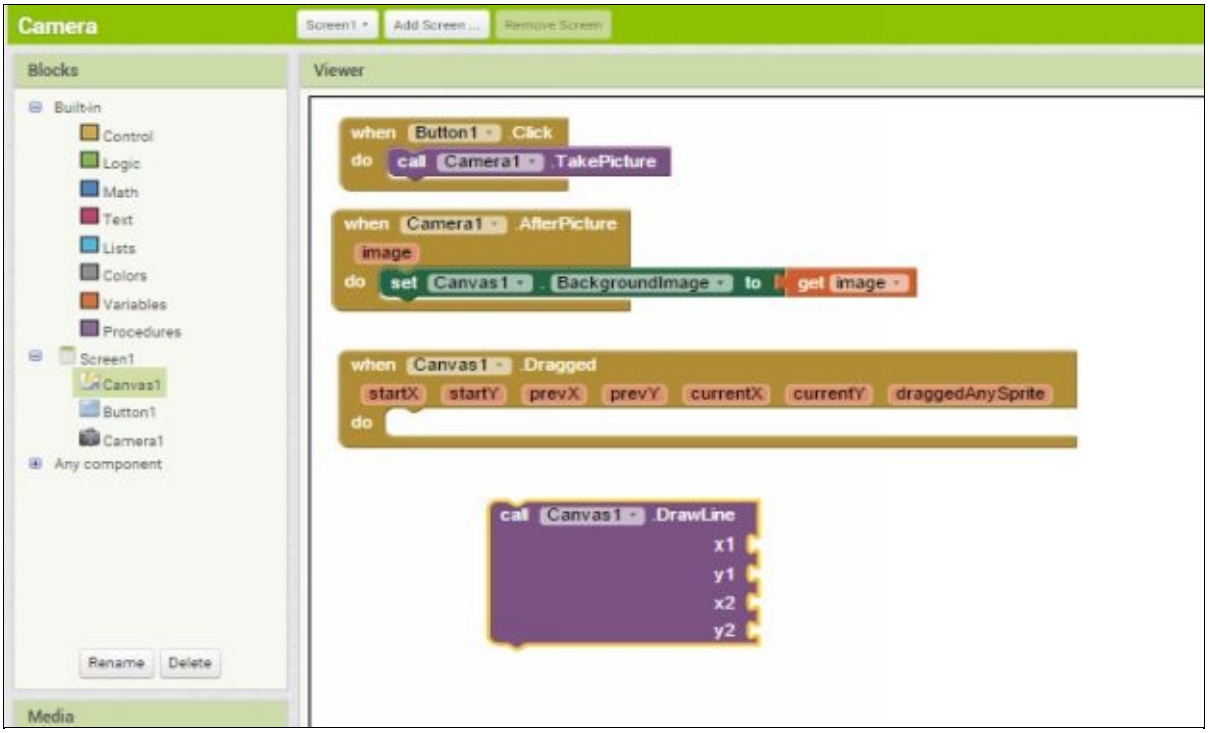
Chapter 9: Digital Doodle App (Extending the Camera app)

16. Now open the Camera app and go to blocks and click on Canvas1 on left side and click on  block. This block will

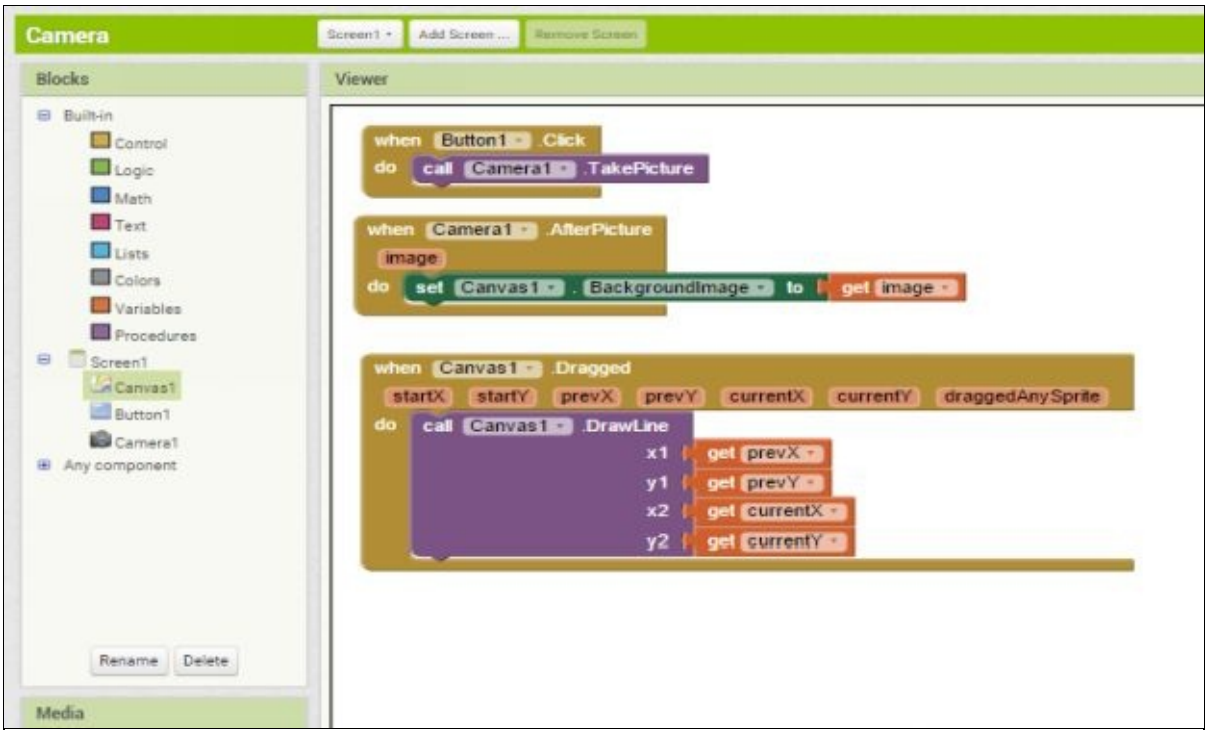
identify dragging activity on your canvas and decide what to do when dragged.



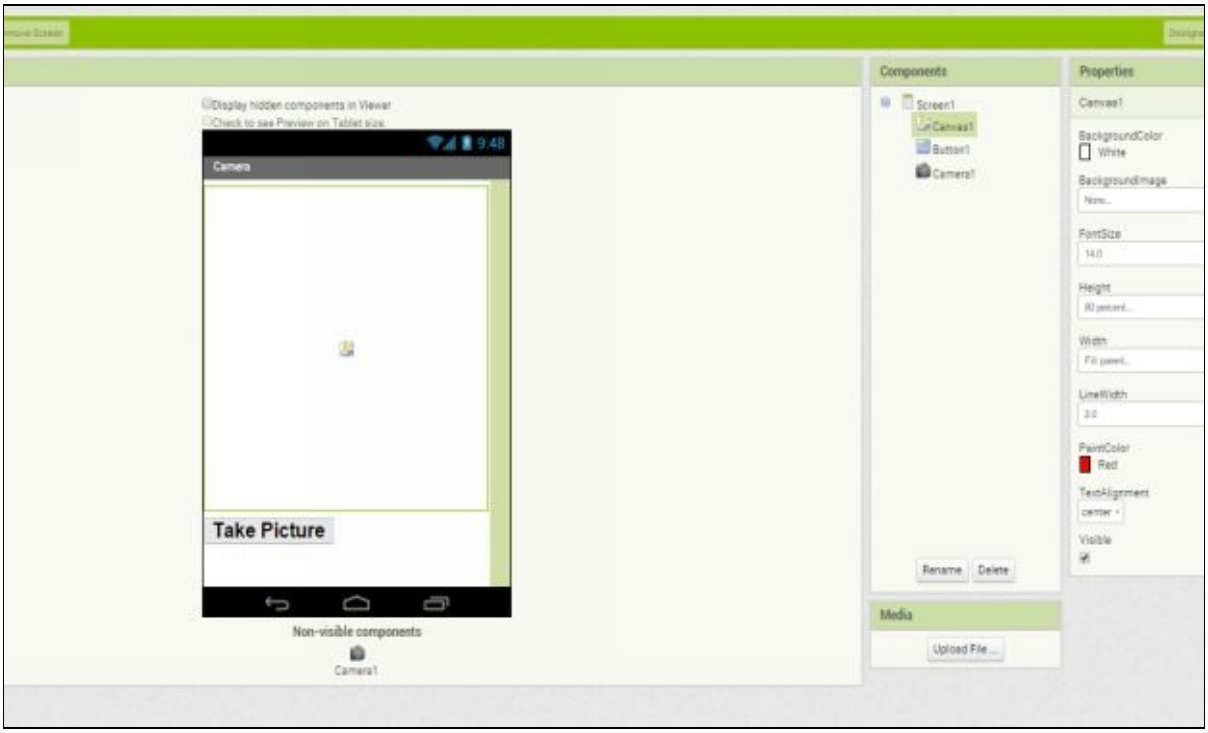
17. Now click on Canvas1 on left side and click on  block. This block will draw a line on your canvas. A line has starting point(x1,y1) and ending point(x2,y2).



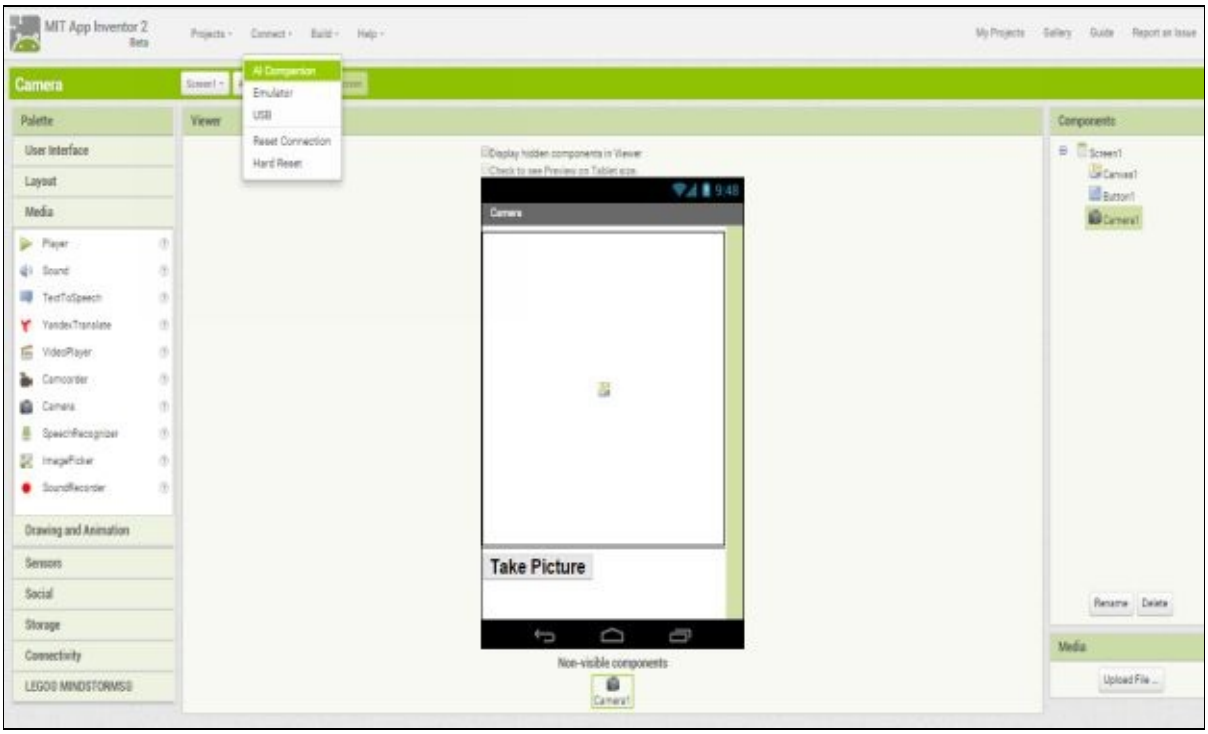
18. Now connect the blocks as shown below. This logic will draw a line on canvas where prevX and prevY are the co-ordinates of starting point of line and currentX and currentY are the co-ordinates of end point of line.

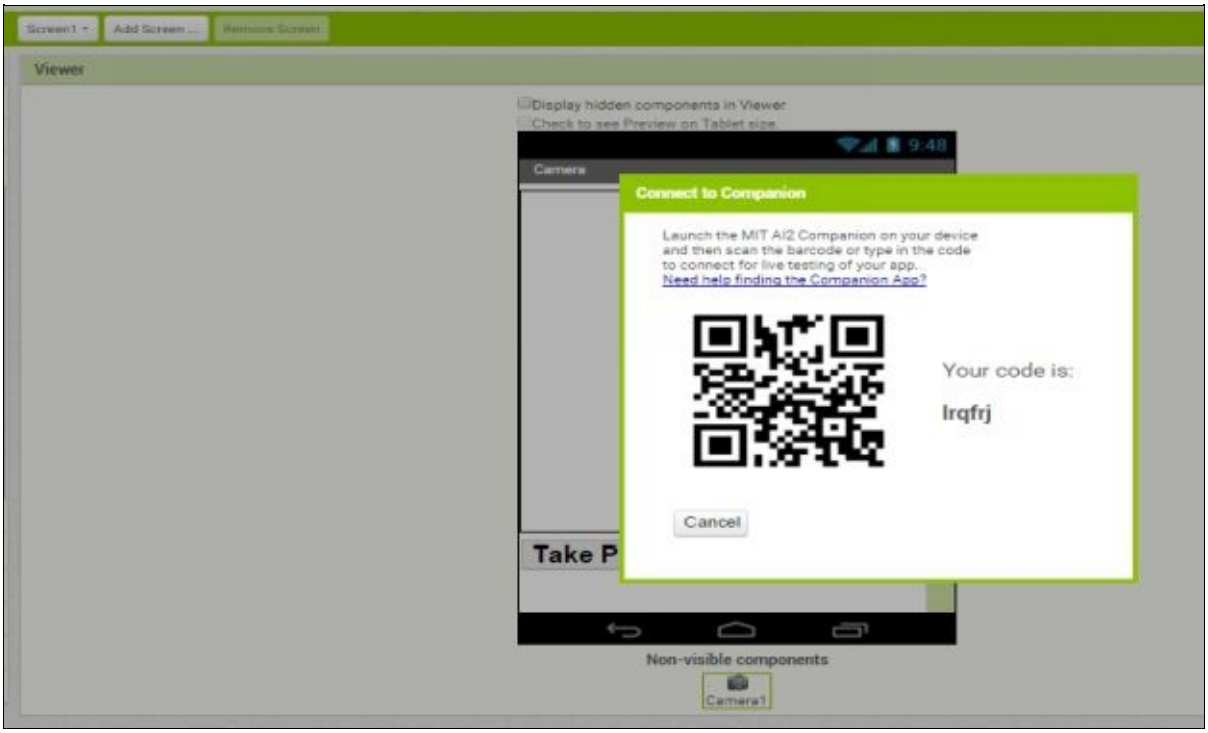


19. Now go to Designer and change PaintColor for Canvas1 Component in Properties.



20. Click on Connect and then AI Companion.

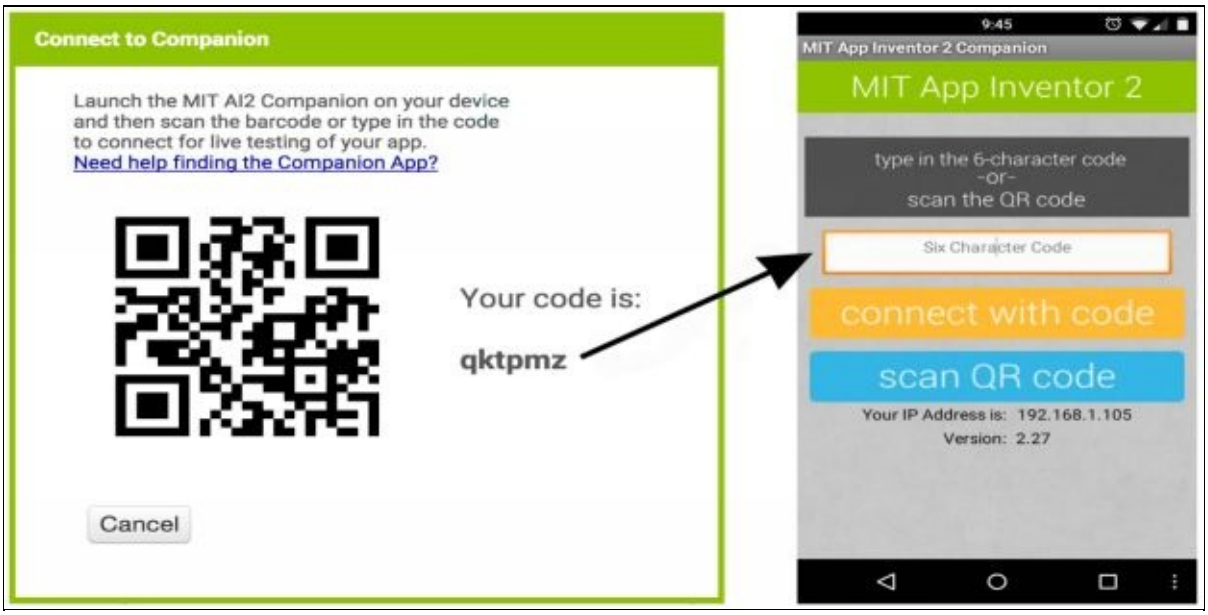




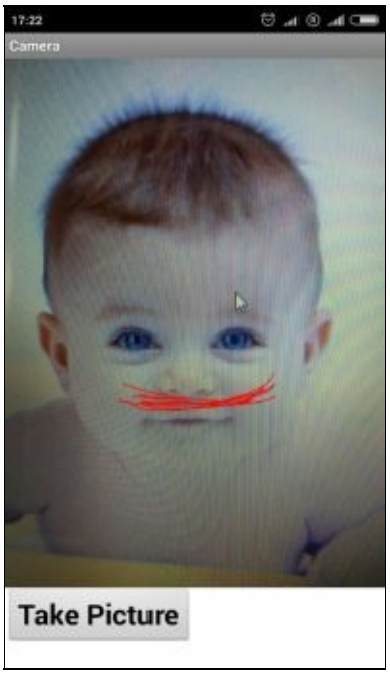
21. Now open MIT App Inventor 2 Companion in your mobile/Tablet.

And enter the code or scan the QR code from your mobile and click on connect with code.

Note:-Your Mobile/Tablet should be connected to same wi-fi/network to which your computer/Laptop is connected.

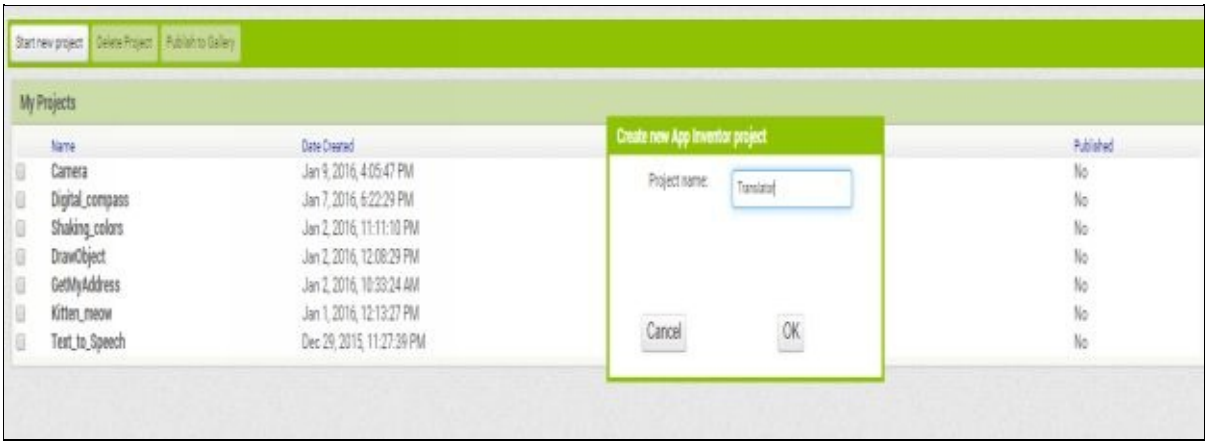


22. Now you should see your app in your phone for live testing. Now click on Take Picture Button and then Camera will open and take picture .Picture will be shown in the canvas that you have taken. Now here you can draw on your taken picture.

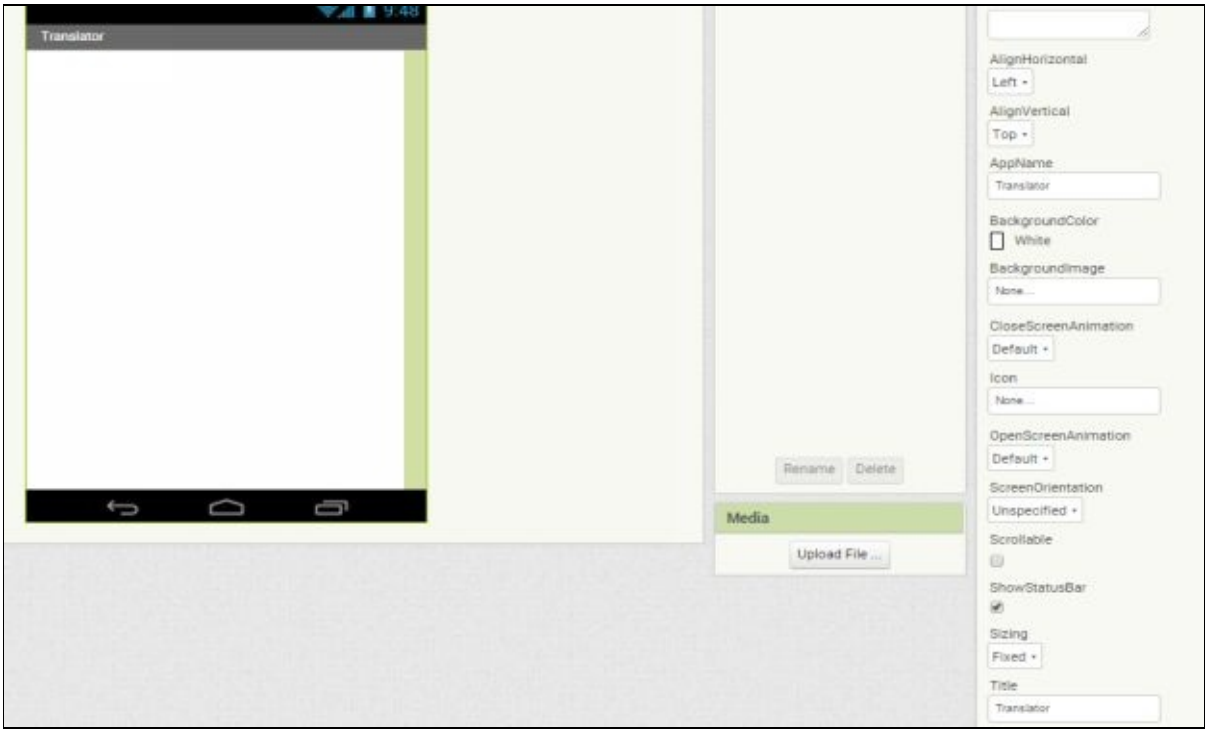


Chapter 10: Translator App

1. Click on Start new project and give it name Translator and click OK.

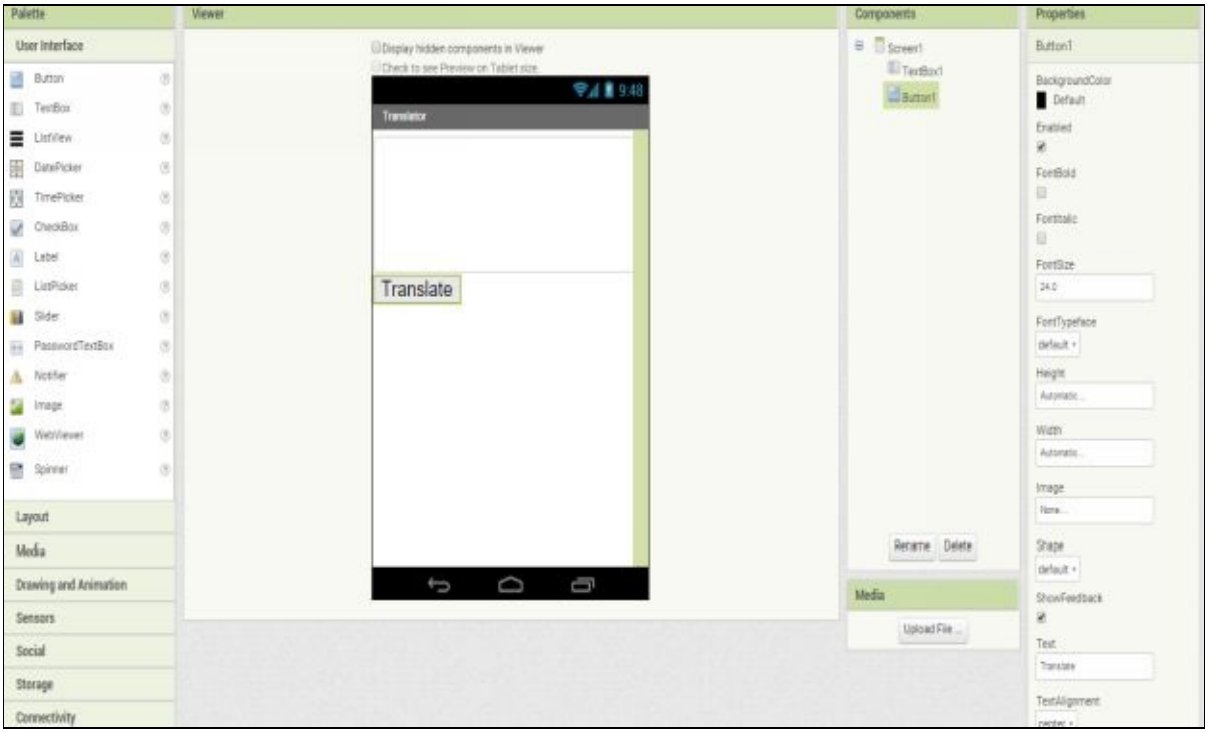


2. Change Screen1 Title to Translator.

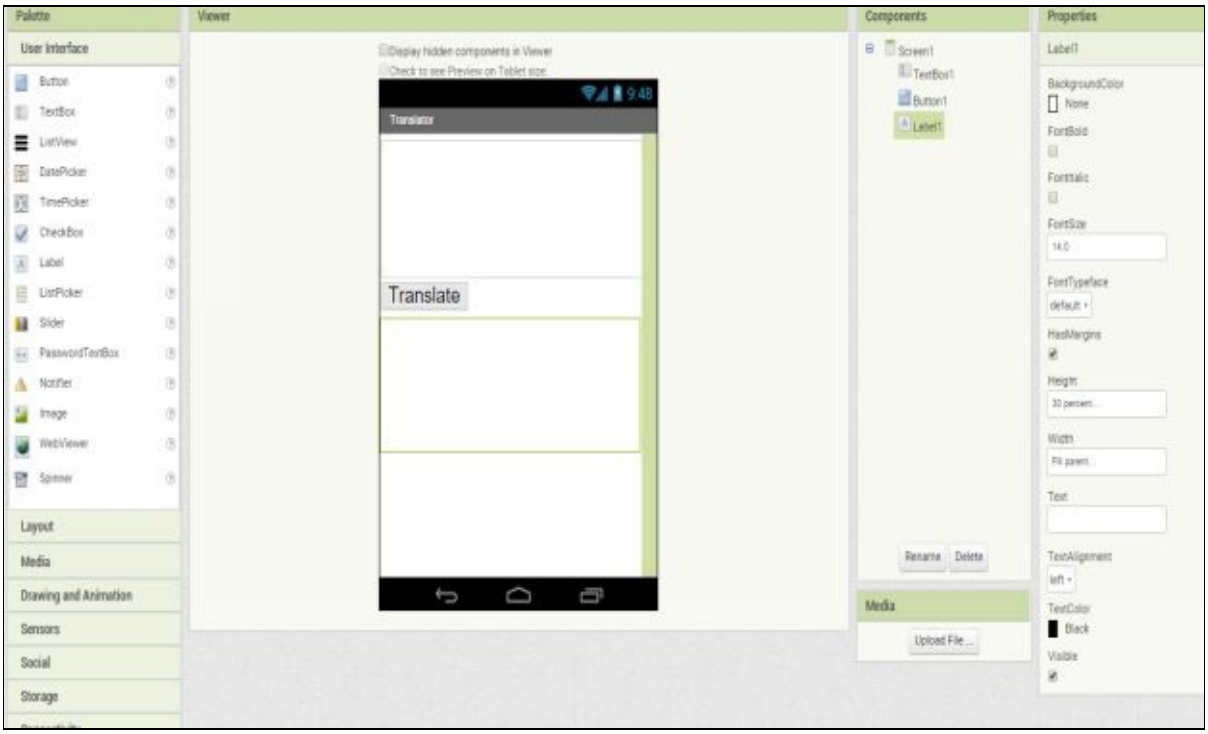


3. Drag a TextBox from Palette to Viewer screen and set its Properties as show below.

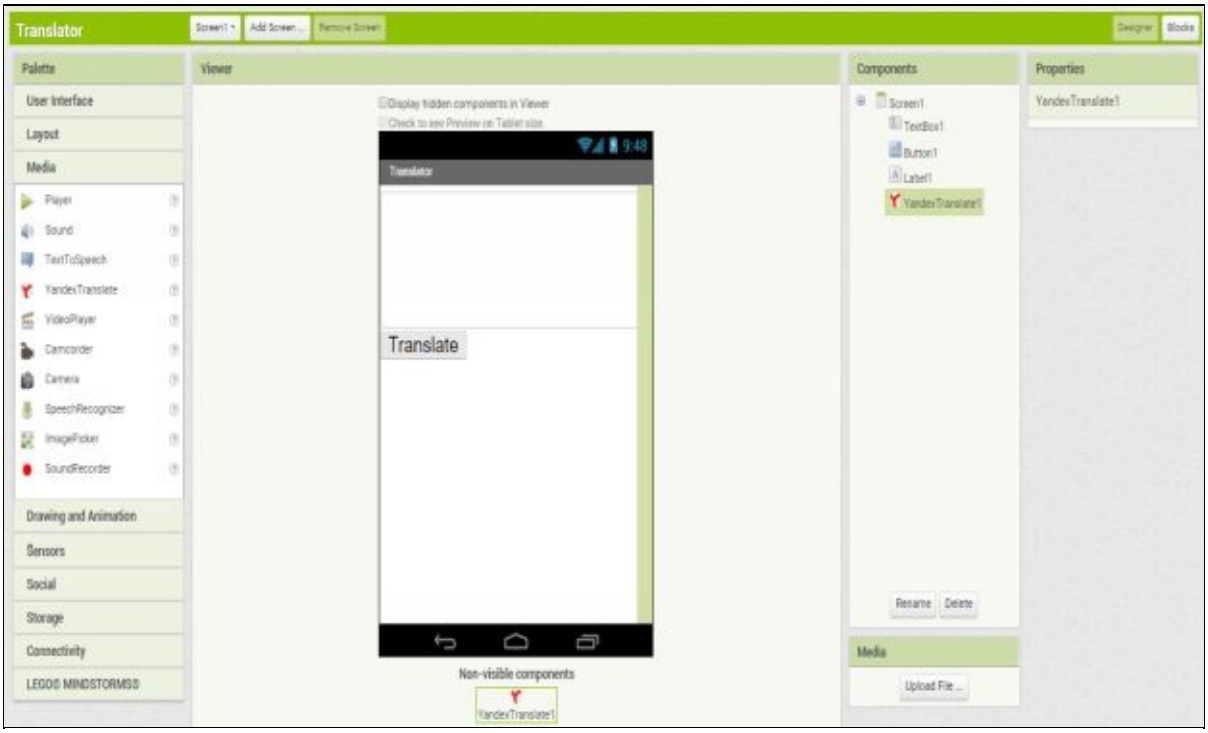
4. Now drag a Button component from Palette to Viewer screen and change its Text to Translate and FontSize to 24.0.



5. Now drag a Label from Palette to Viewer screen and set its Properties as shown below.



6. Now Drag a YandexTranslate component from Palette to Viewer screen. It's a non-visible component.

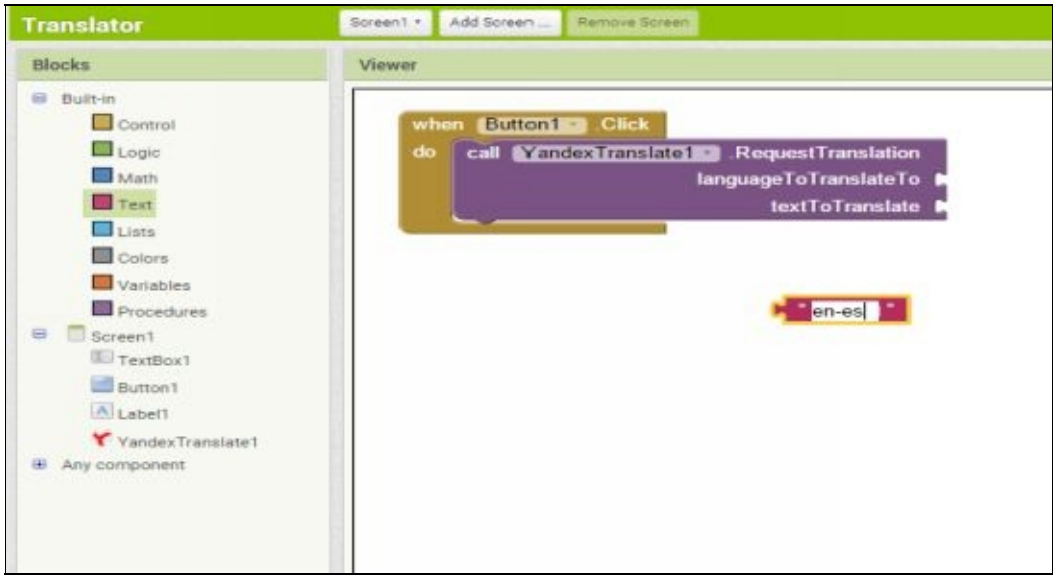


7. Now go to blocks and click on Button1 component on left side and click on block. This is an event block which will decide what your app will do when this button will be clicked. Here click is an event for Button1 component.

8. Now click on YandexTranslate1 component and click on block. This block will translate the Text.

9. Now attach these blocks as show below. So that when you click Button1 your app will translate some text.

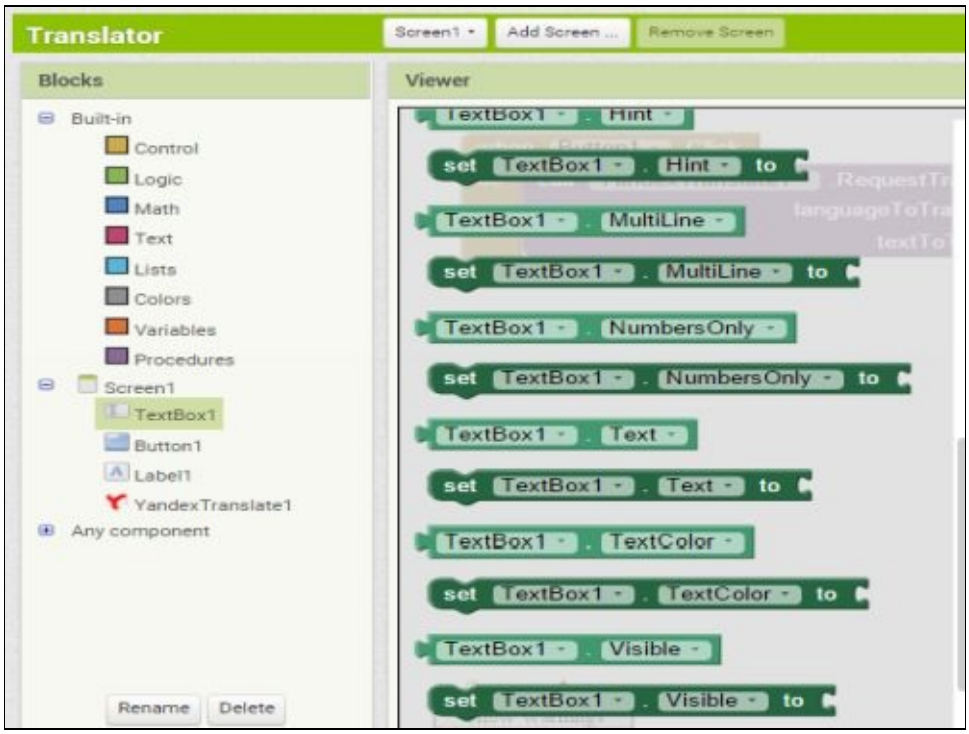
10. Click on Text under Built-in and click on  block and type en-es in the block that will tell the translate engine to take the text in English and translate to Spanish.



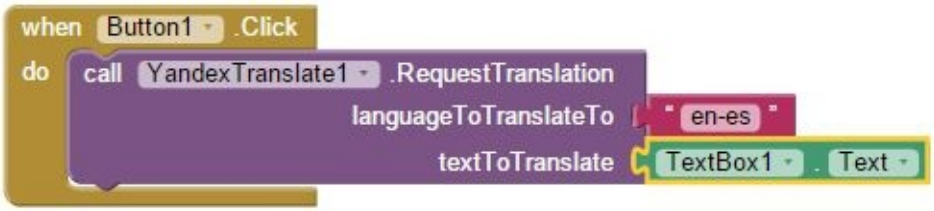
11. Now attach the blocks as show below. This will tell your app to translate from English language to Spanish when Button1 is clicked.

12. Click on TextBox1 Component and scroll down to select 

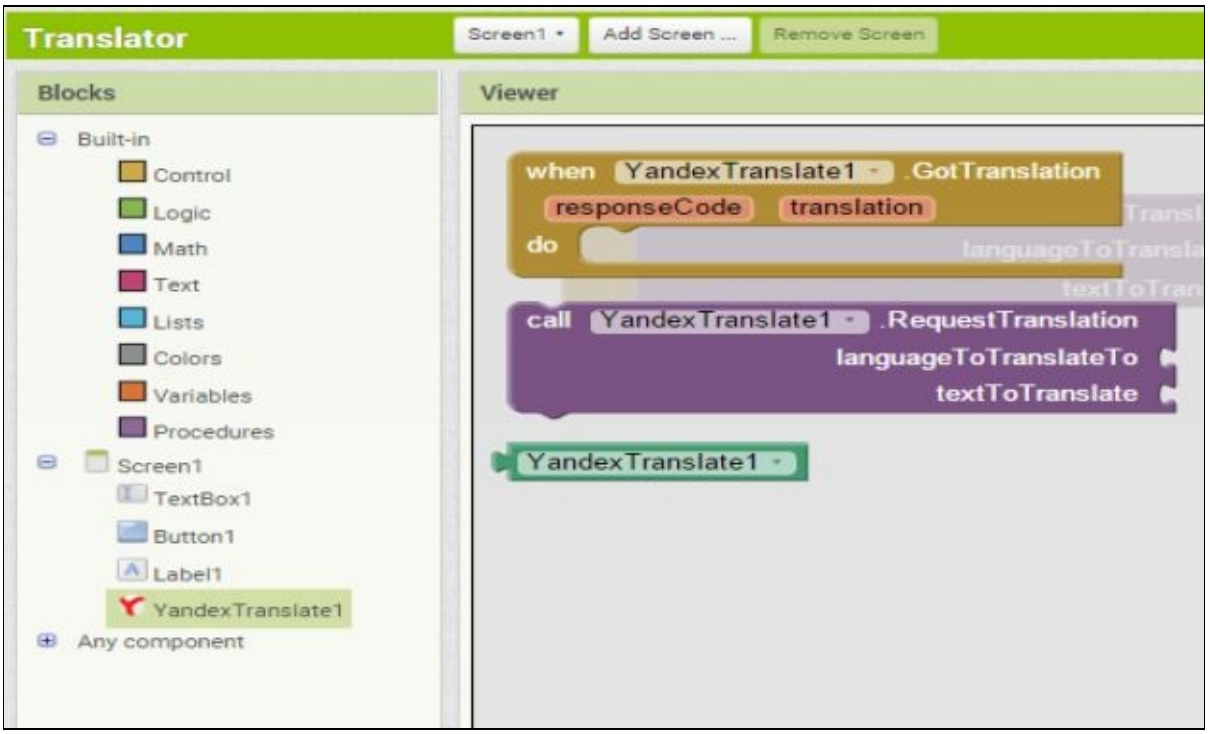
block.



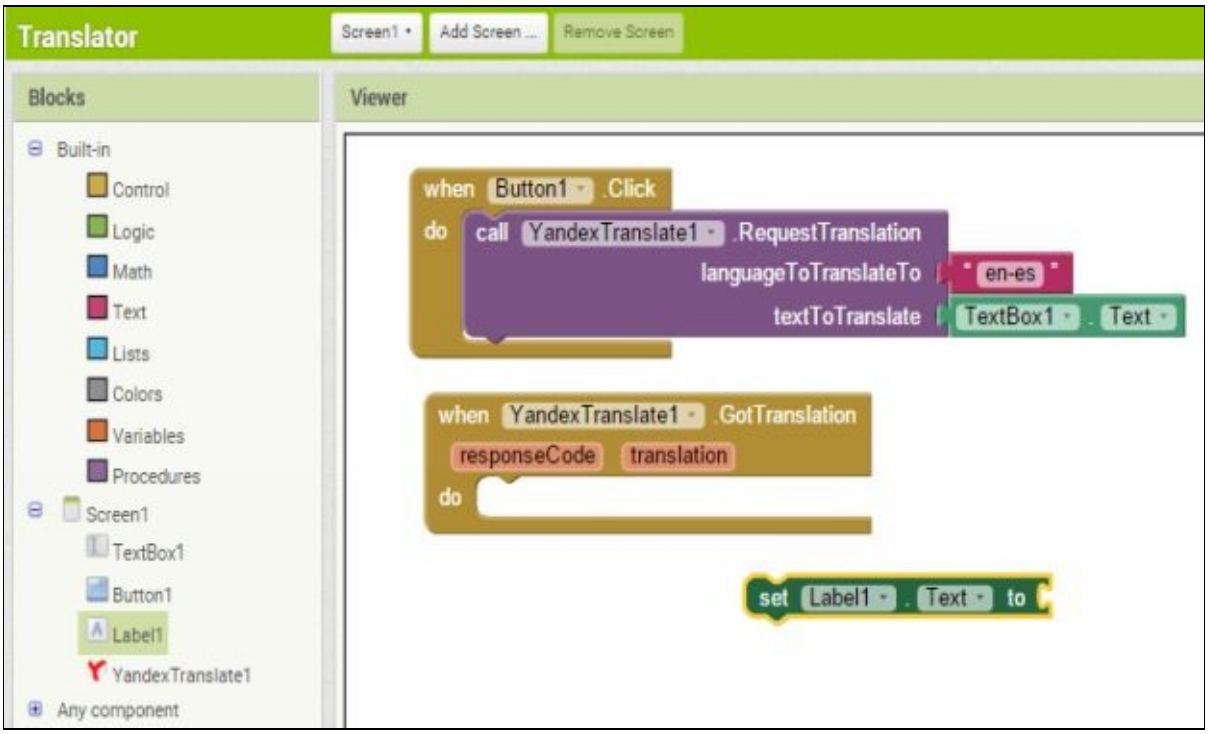
13. Attach the blocks as show below. So when Button1 is clicked your app will translate the text which is there in TextBox1.



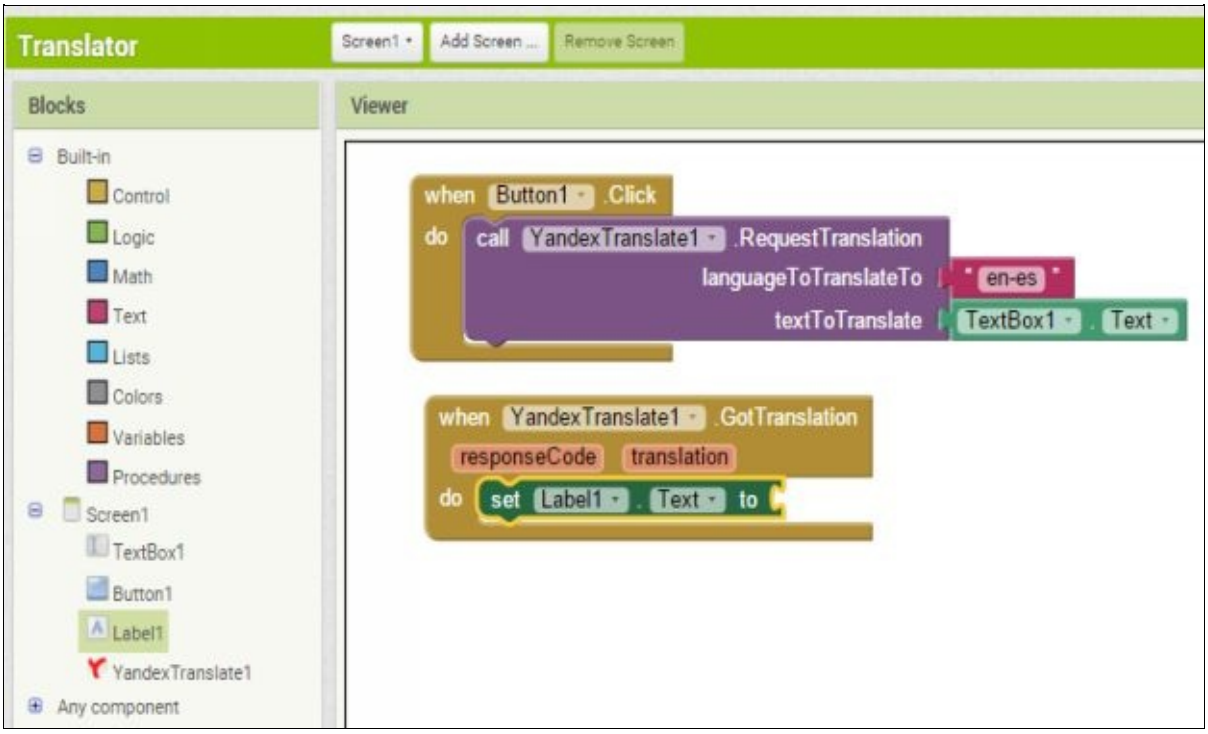
14. Now click on YandexTranslate1 and click on block. This will decide what to do when Translation is completed.



15. Now click on Label1 component and select  block.

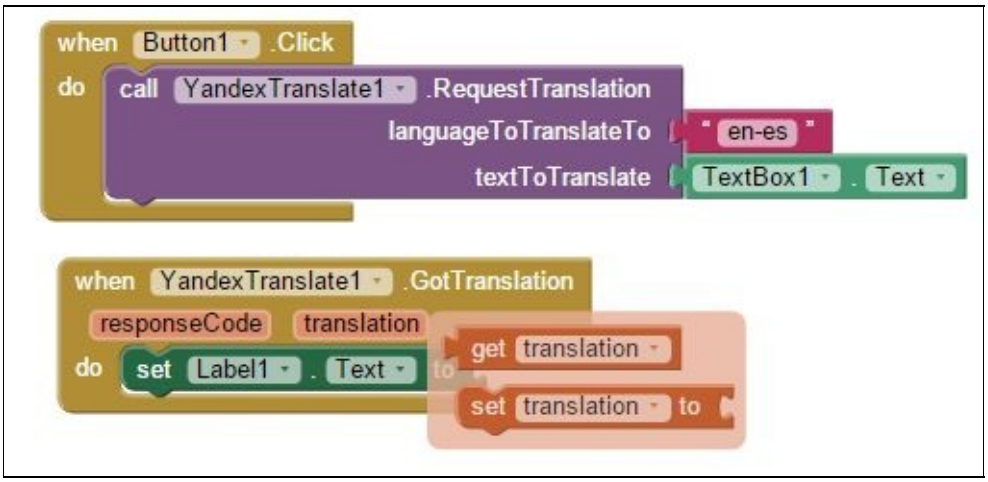


16. Attach the blocks as show below.

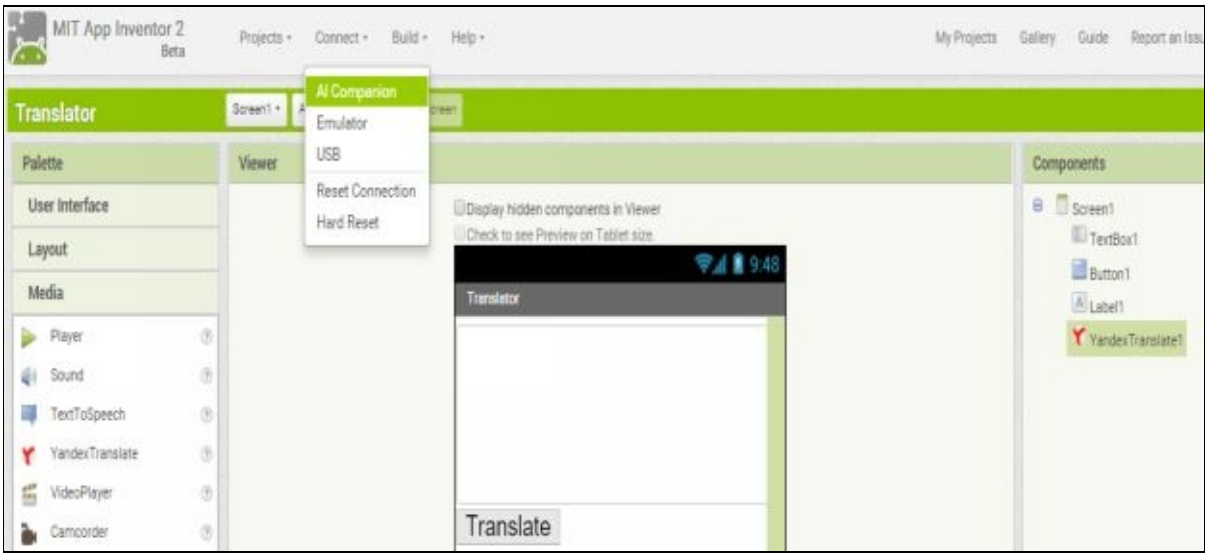


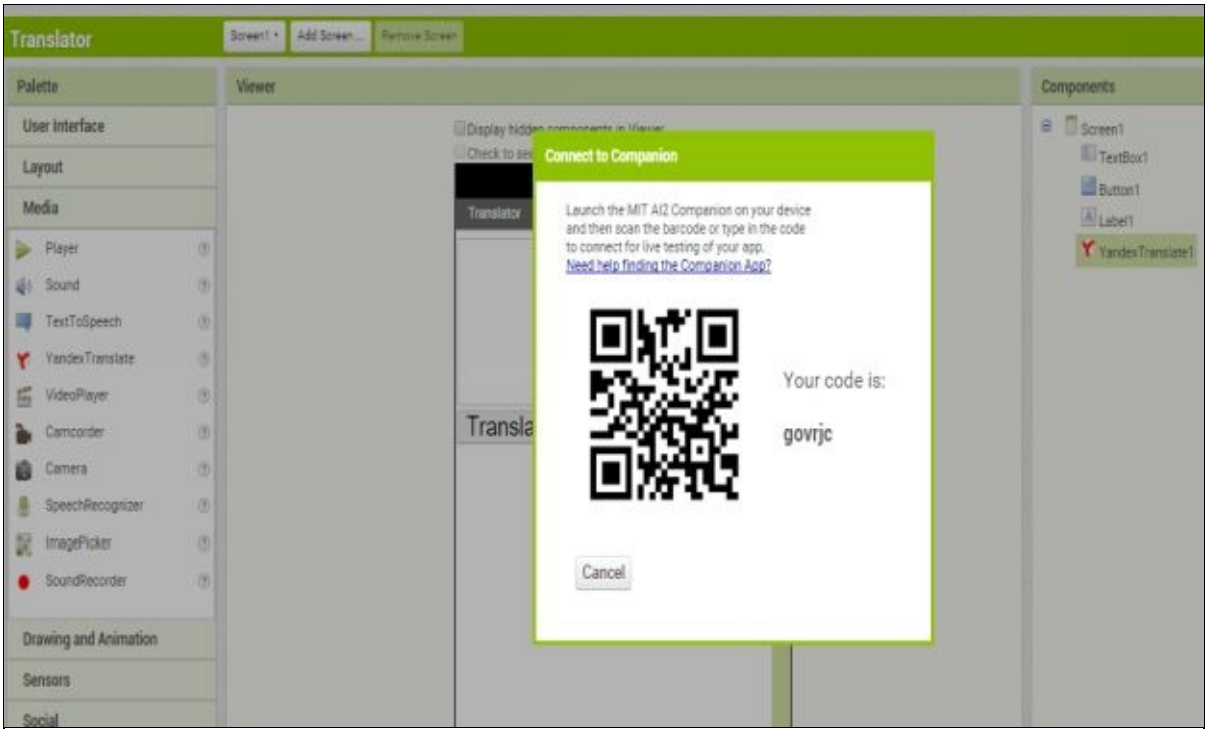
17. Now mouse over on translation and select 

block and attach the blocks as show below. So when translation is complete your app will set Label's Text as Translated text.



18. Now go to Designer and click on Connect-> AI Companion.

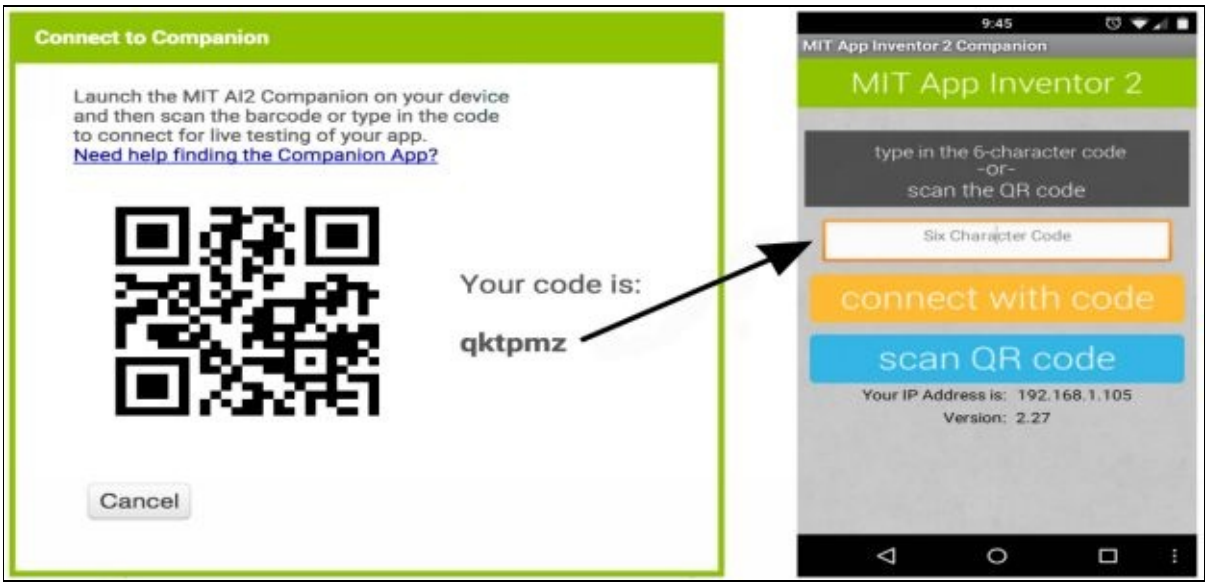




19. Now open MIT App Inventor 2 Companion in your mobile/Tablet.

And enter the code or scan the QR code from your mobile and click on connect with code.

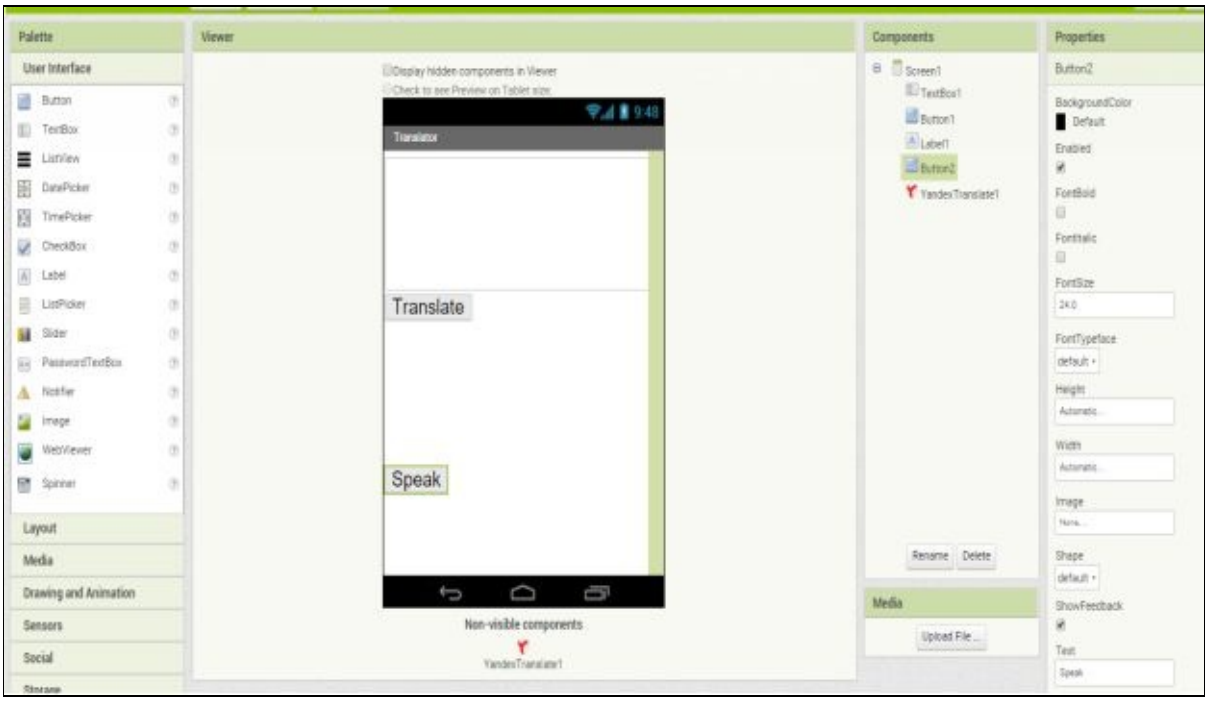
Note:-Your Mobile/Tablet should be connected to same wi-fi/network to which your computer/Laptop is connected.



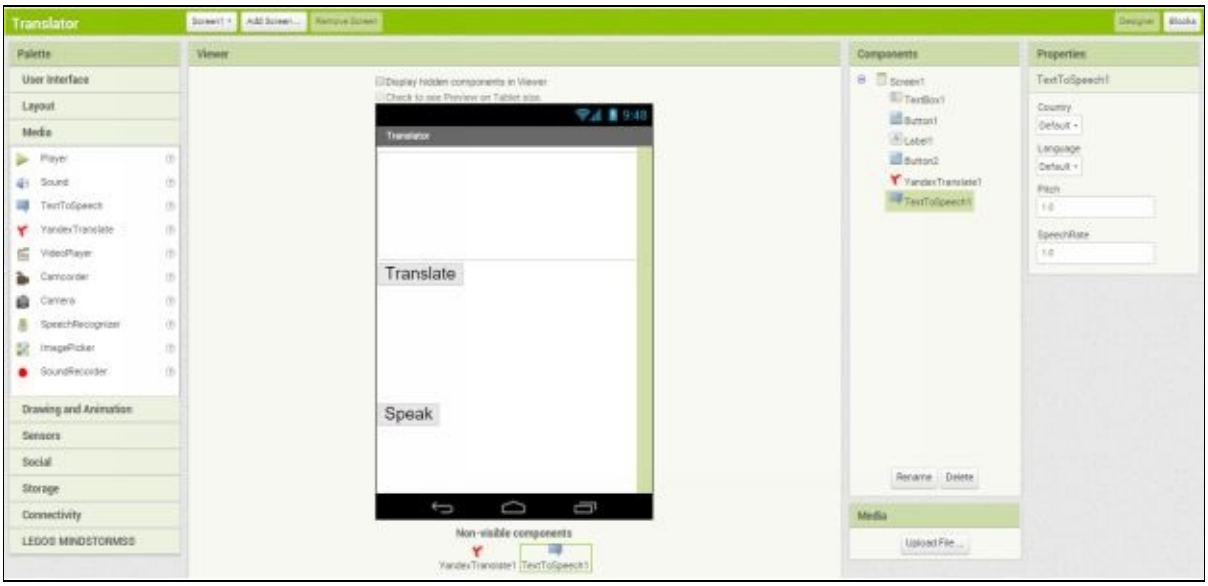
20. Now you should see your app in your phone for live testing. Type anything in the textbox and the click on Translate button and the Spanish text will be shown below Translate Button.

Chapter 11: Translator App (Extended)

21. Open your Translator app and Drag a Button from Palette to Viewer screen and set its Properties as shown below.

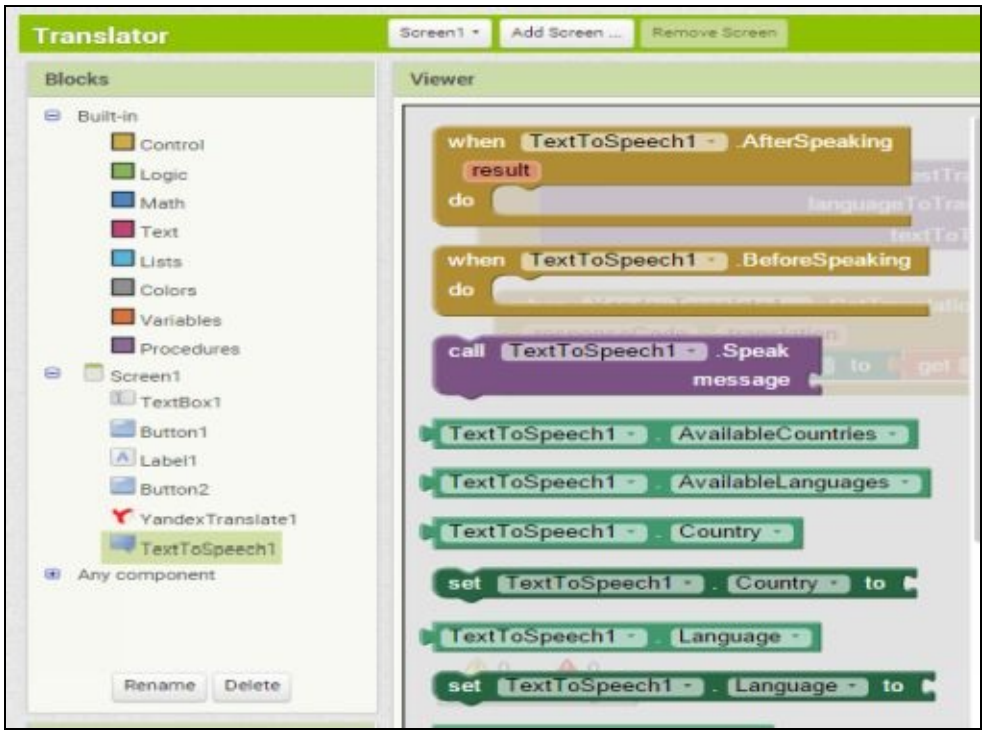


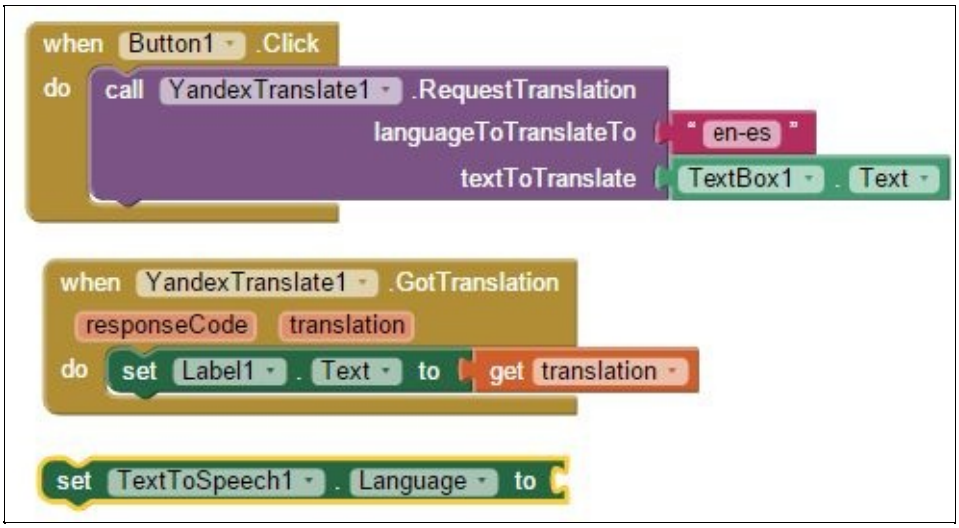
22. Now drag a TextToSpeech component from Palette to Viewer screen. It's a non-visible component.



23. Now go to blocks and click on TextToSpeech and click on 

block. This block will make your app speak the text in a particular language.



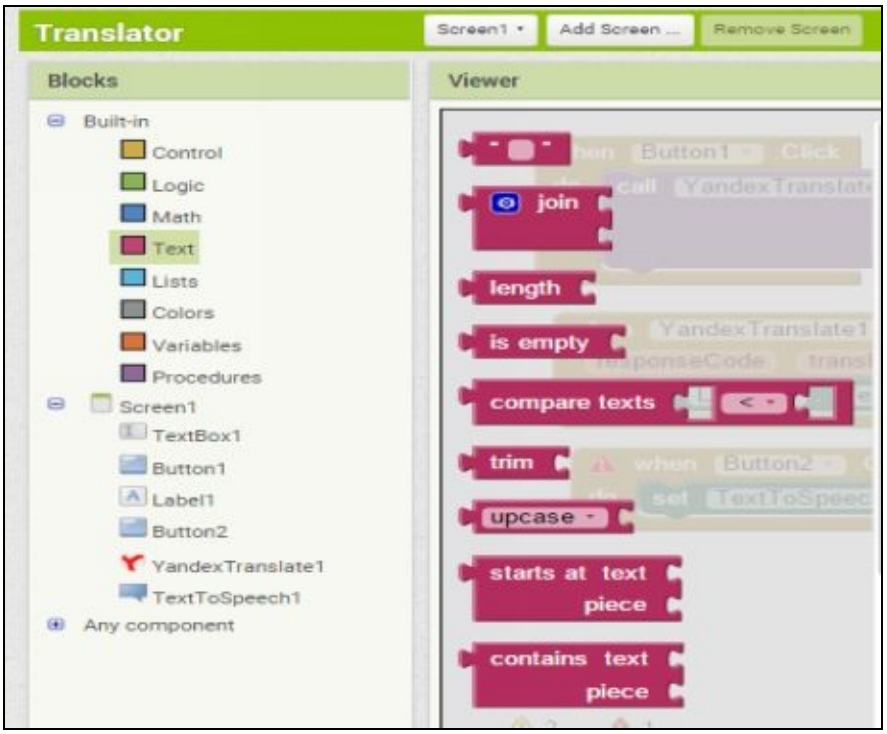


24. Now click on Button2 and click on 

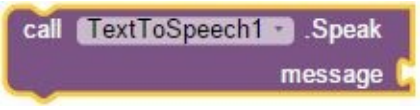
block. This block will decide what to do when Button2 is clicked.

25. Now attach the blocks as shown below.

26. Now click on Text under Built-in and click on  block.

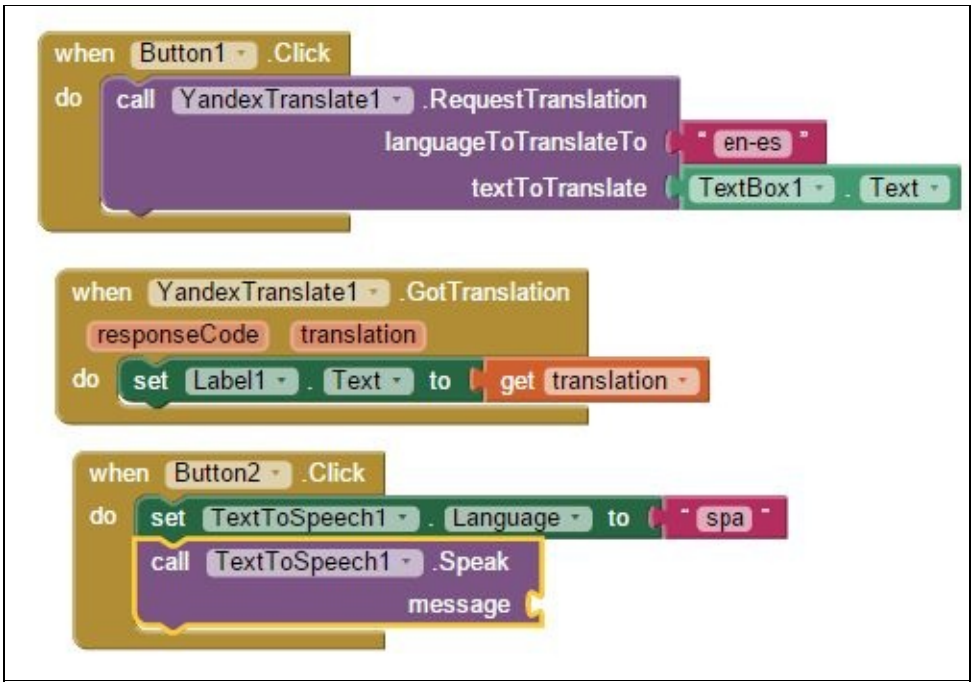


27. Type ‘spa’ as text and attach the blocks as shown below and attach the blocks as shown below. This will tell your app to speak in ‘Spanish’ language.

28. Now click on TextToSpeech1 and click on 

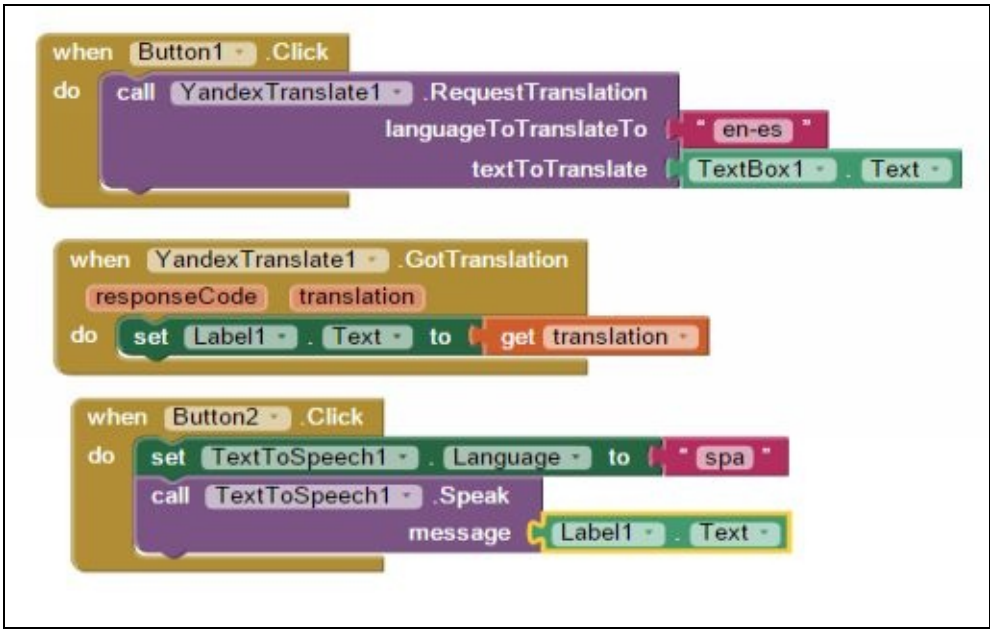
block. This block will speak some message given to it as input.

29. Attach the blocks as shown below.

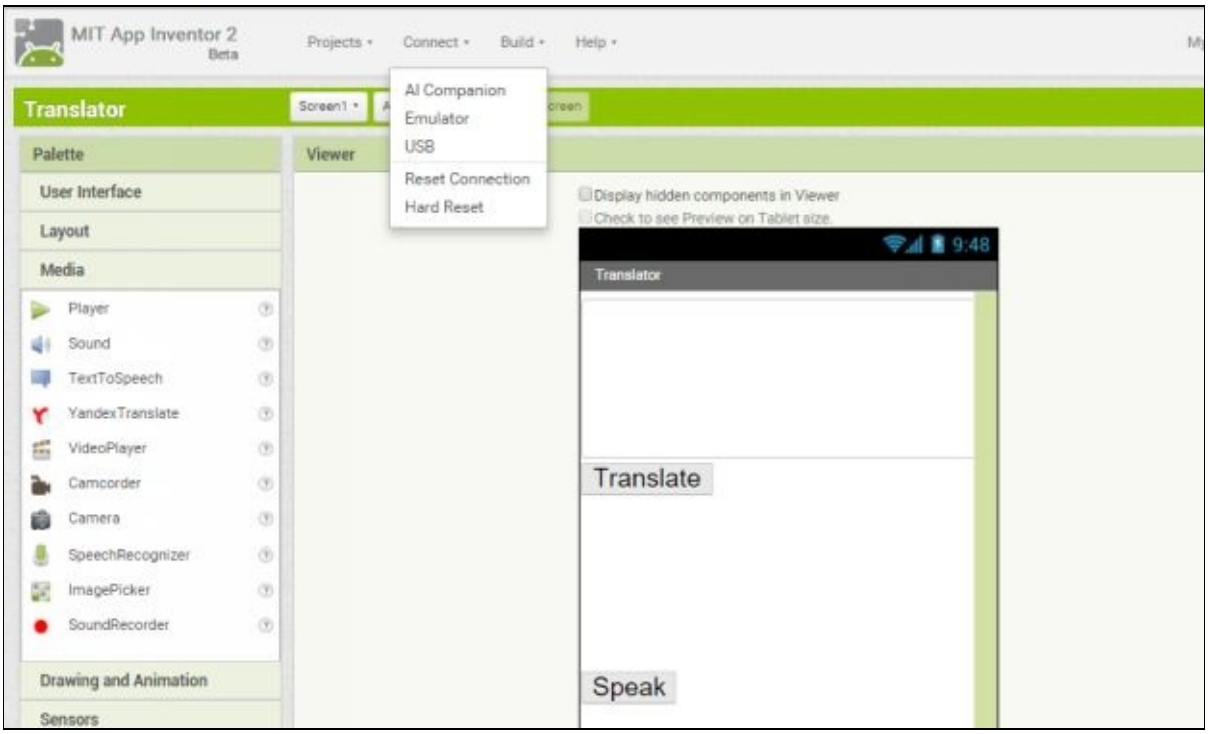


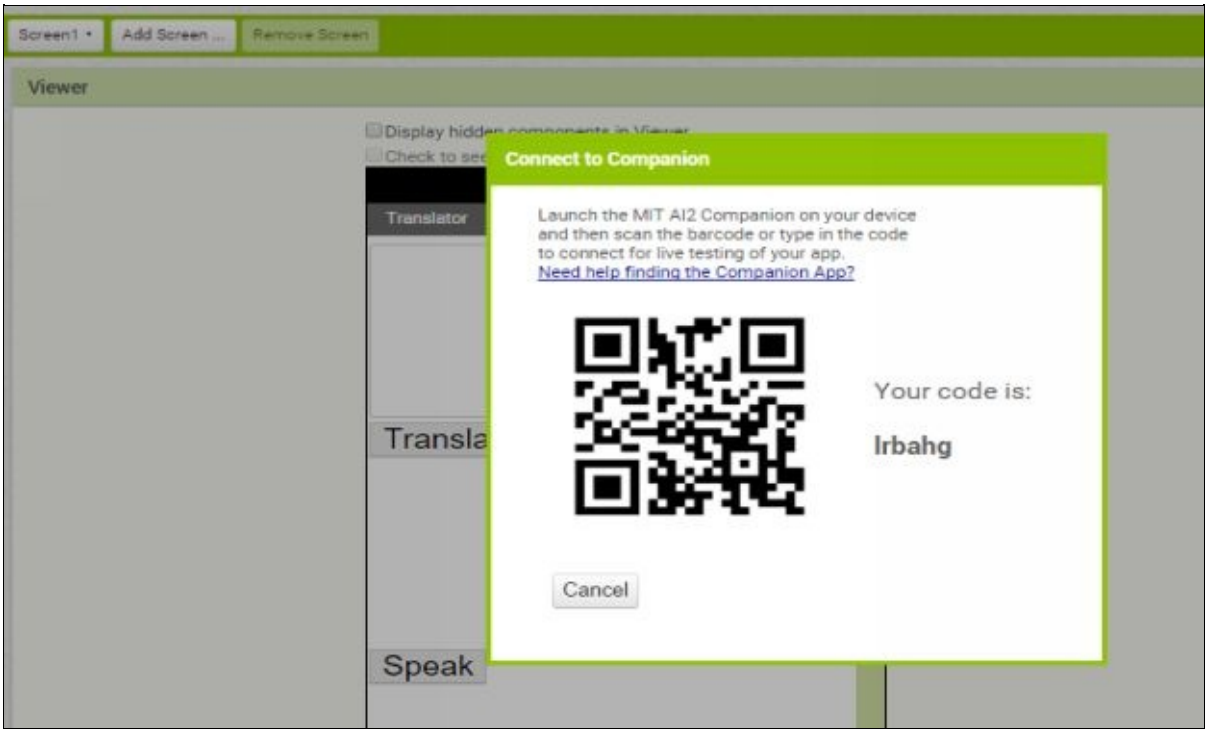
30. Click on Label1 and select  block.

31. Attach the blocks as shown below. So when you click Button2 your app will speak Label1's Text .



32.Now go to Designer and click on Connect and the AI Companion.

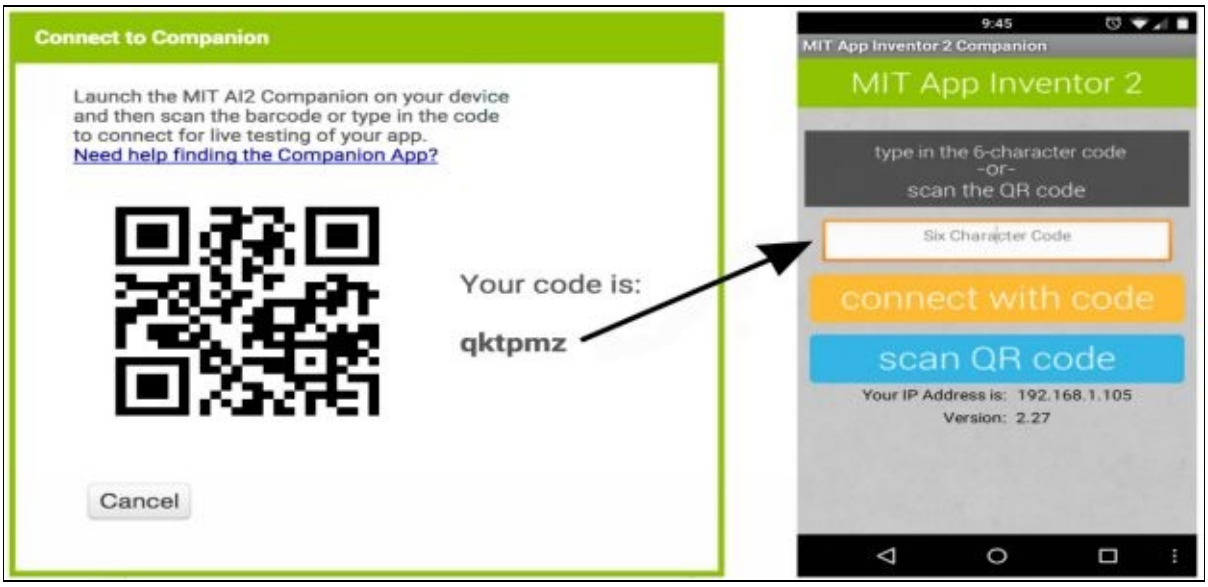




33. Now open MIT App Inventor 2 Companion in your mobile/Tablet.

And enter the code or scan the QR code from your mobile and click on connect with code.

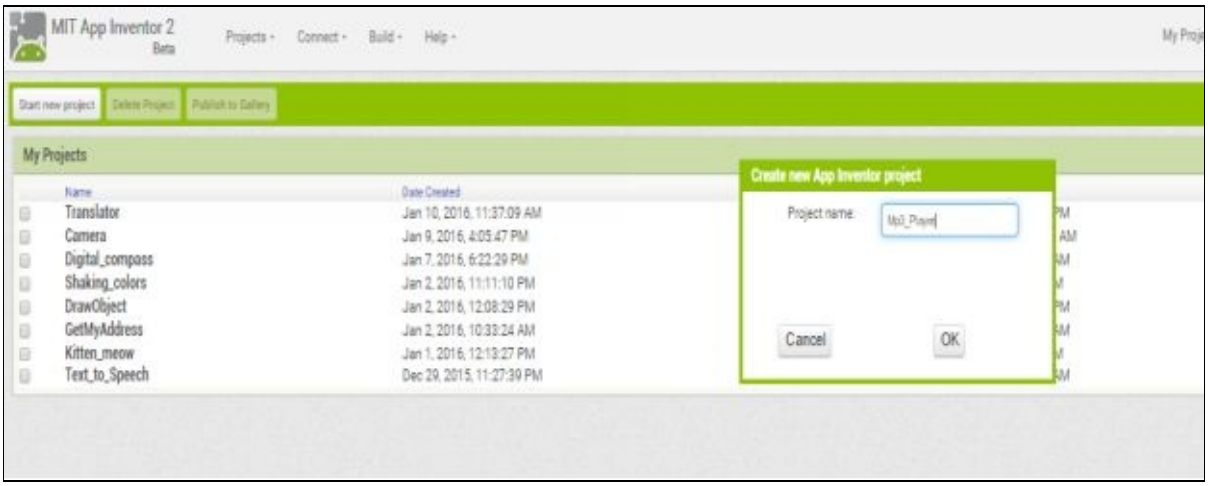
Note:-Your Mobile/Tablet should be connected to same wi-fi/network to which your computer/Laptop is connected.



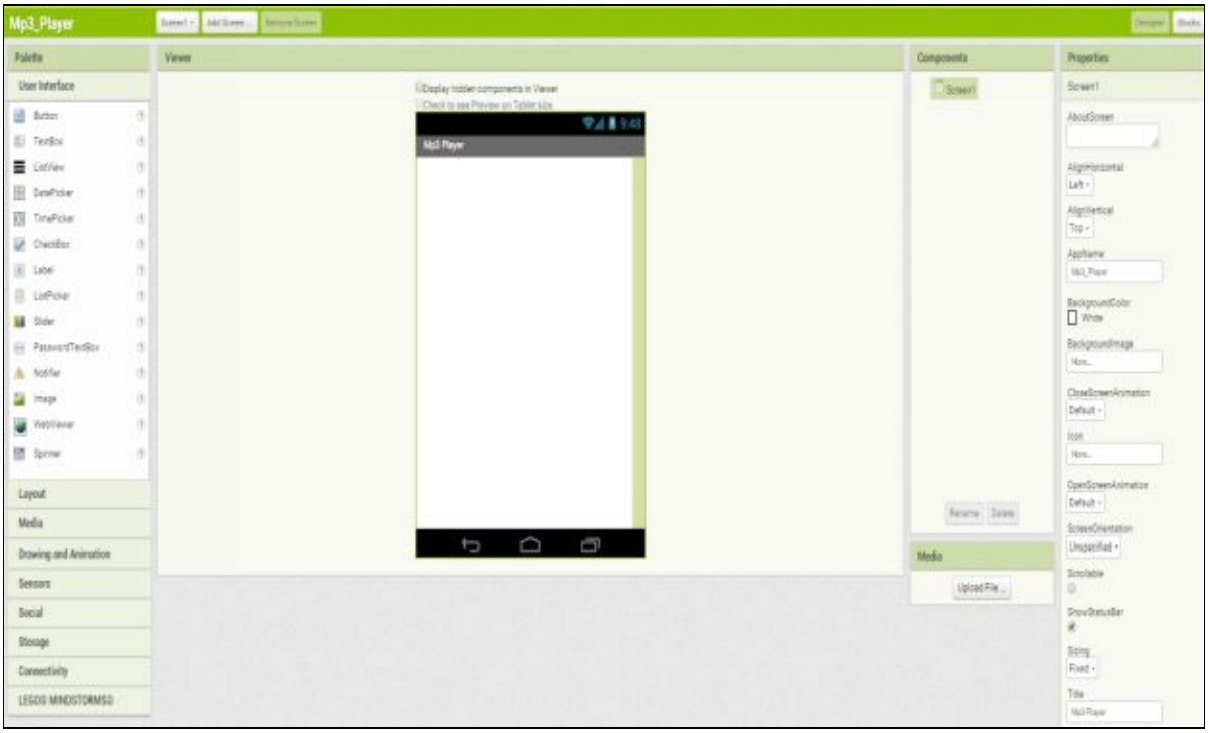
34. Now you should see your app in your phone for live testing. Type anything in the textbox and the click on Translate button and the Spanish text will be shown below Translate Button.Now click on Speak Button, your phone will speak the text in Spanish.

Chapter 12: Mp3 Player App

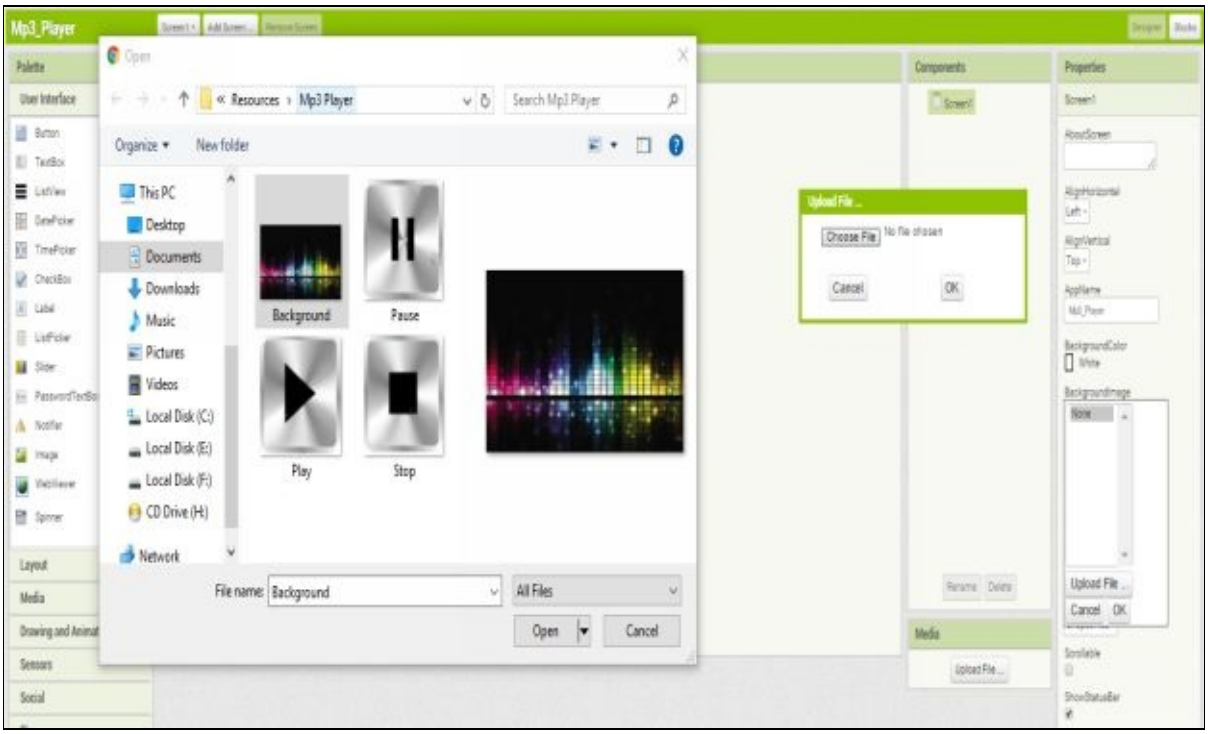
1. Click on Start new project and give it name Mp3_Player and click OK.

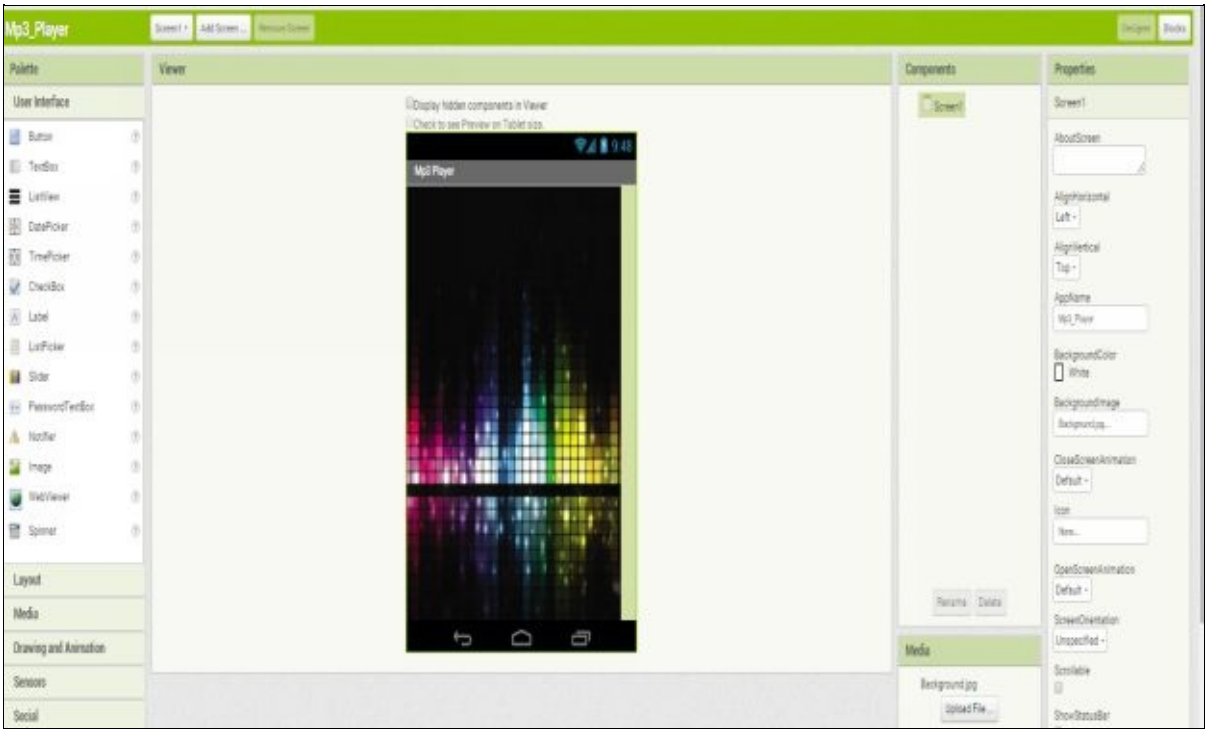


2. Change Screen1 Title to Mp3 Player.

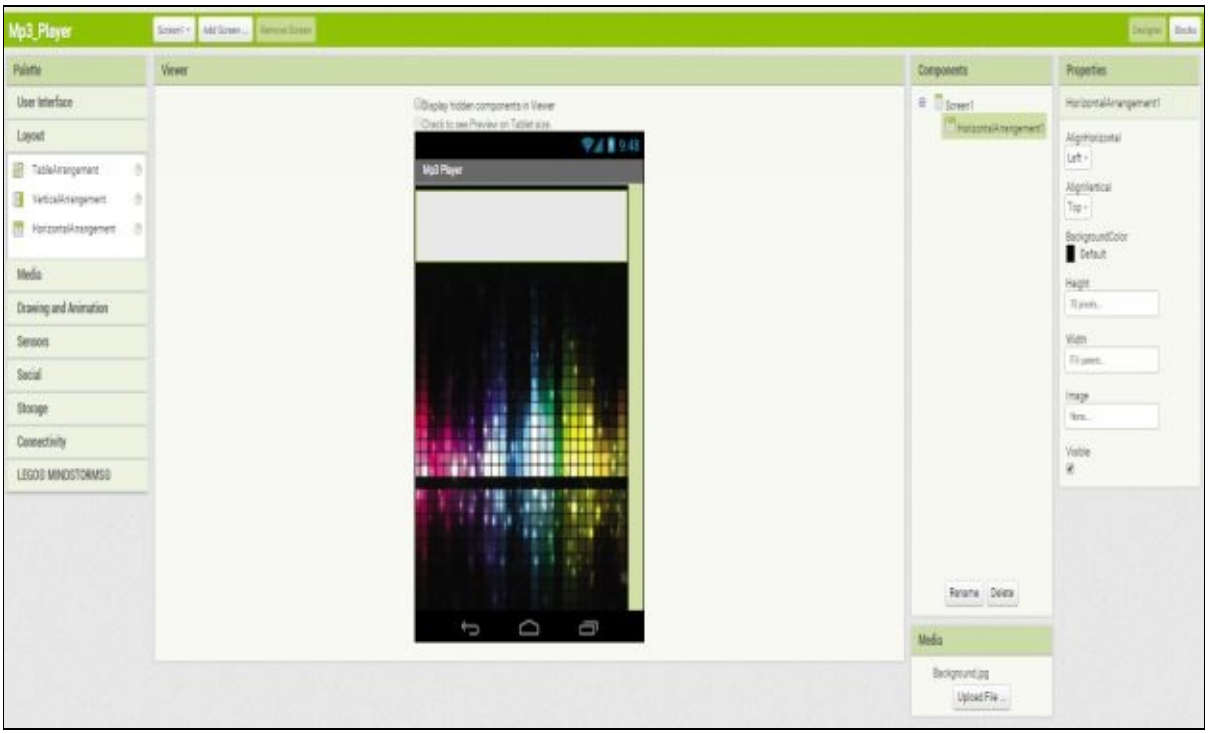


3. Change screen Background image to Background.jpg and click OK.

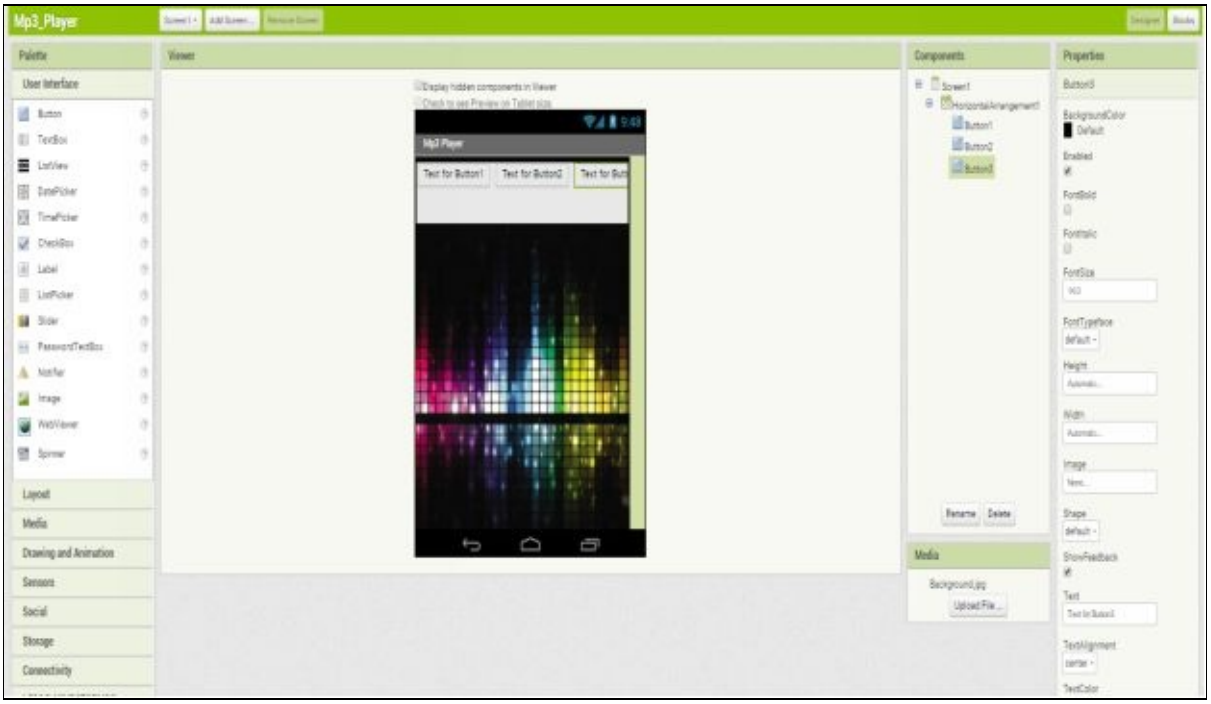




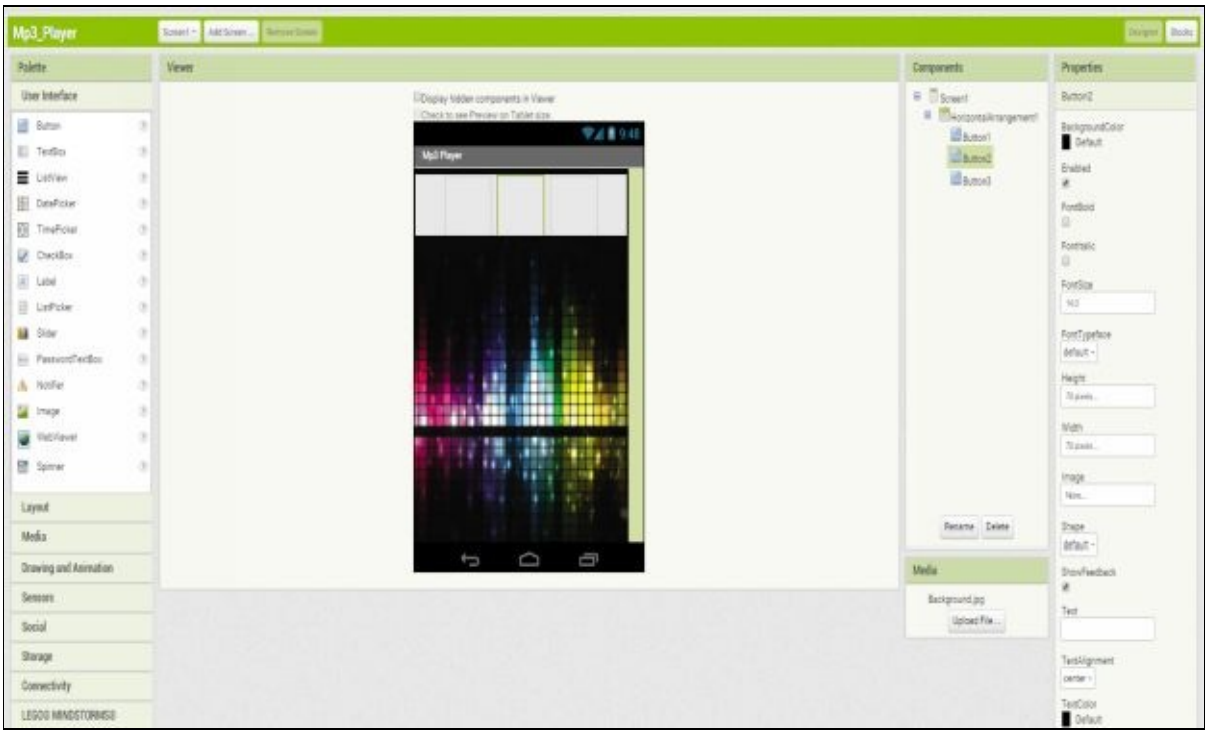
4. Now drag a HorizontalArrangement component from Palette to Viewer screen and set its Height to 70 pixels and Width to Fill Parent.



5. Now drag 3 Buttons from Palette to Viewer screen inside HorizontalArrangement1 component.



6. Set All Button's text to blank and set Height and Width both to 70 pixels for all the 3 Buttons.



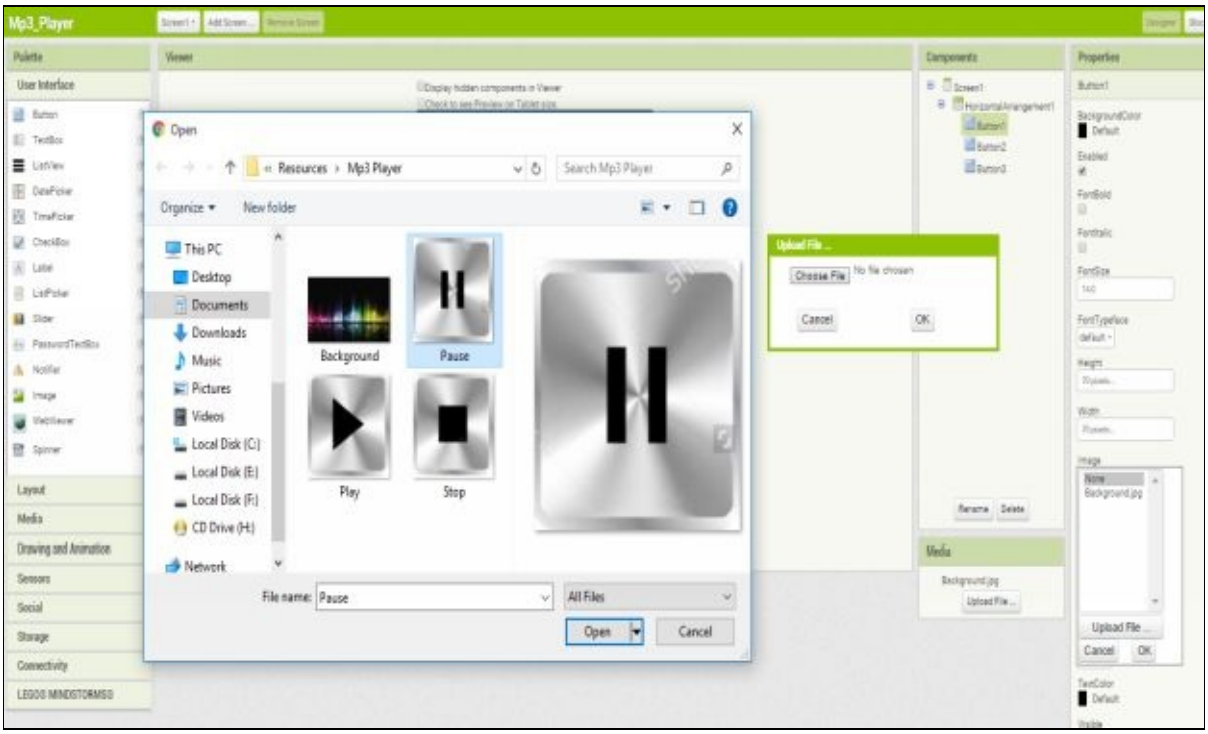
7. Now Upload the Button's images as given below:-

Button1-Pause button.

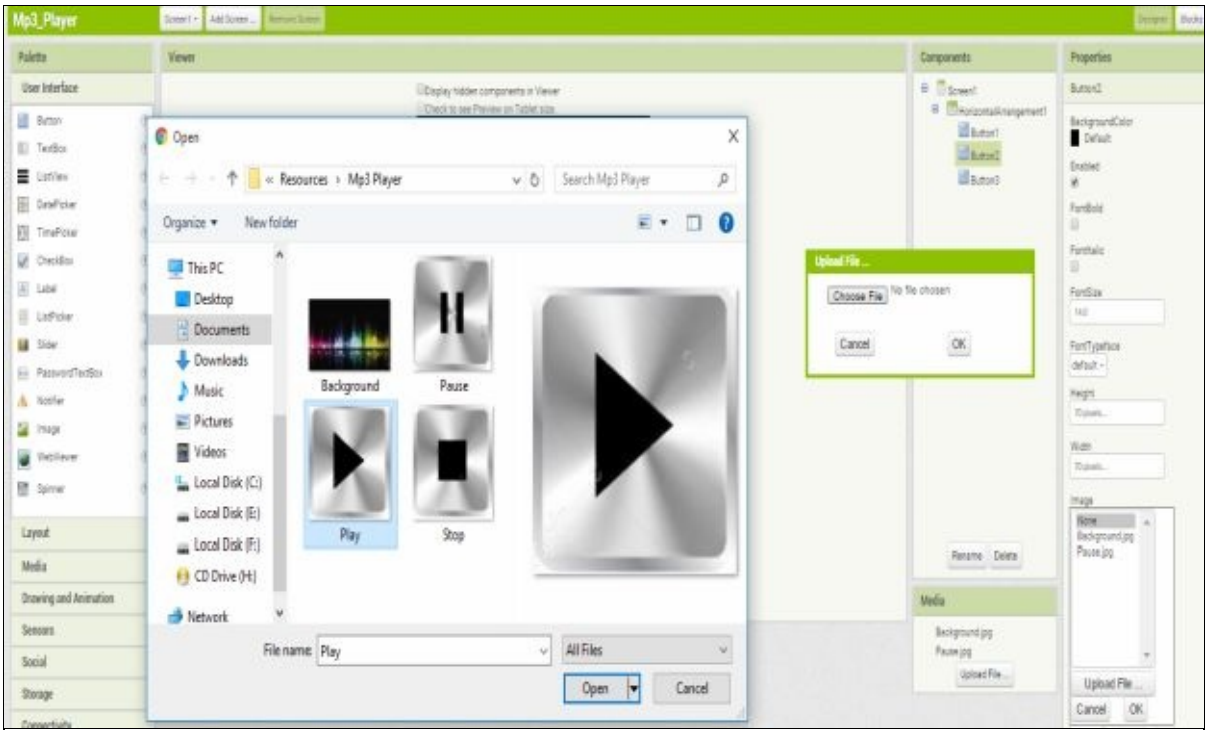
Button2-Play button

Button3-Stop button

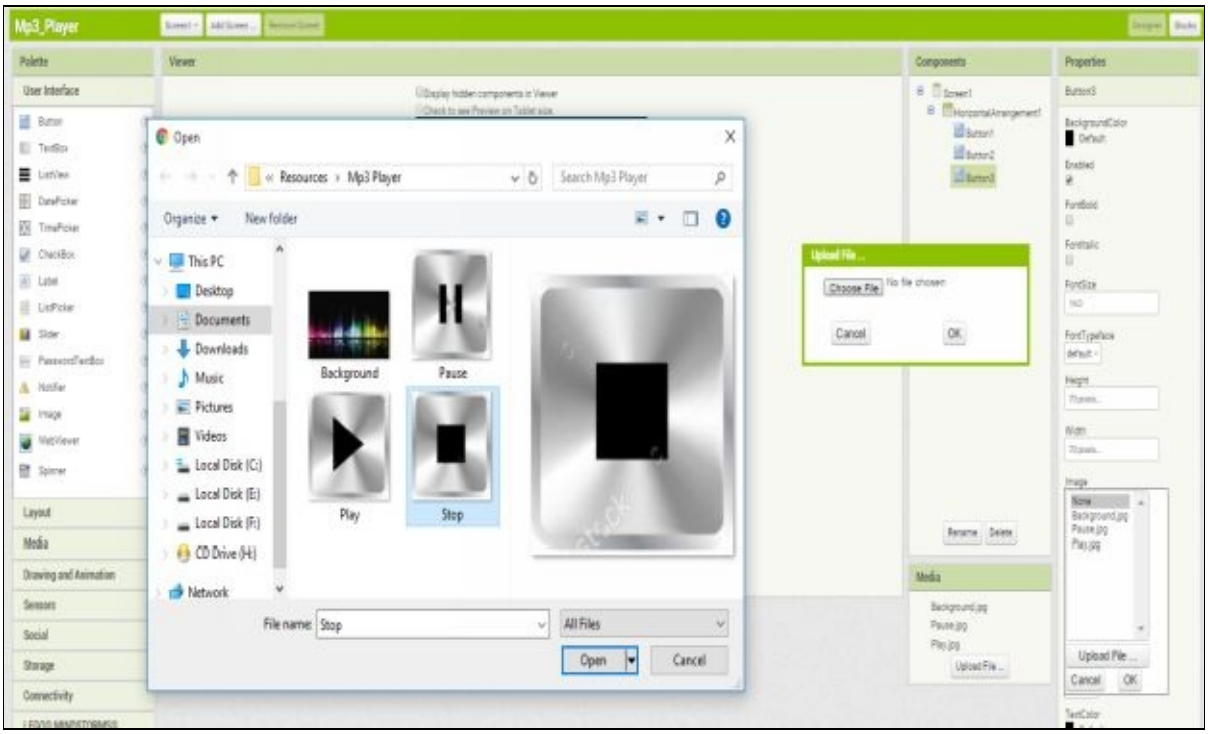
Button1:



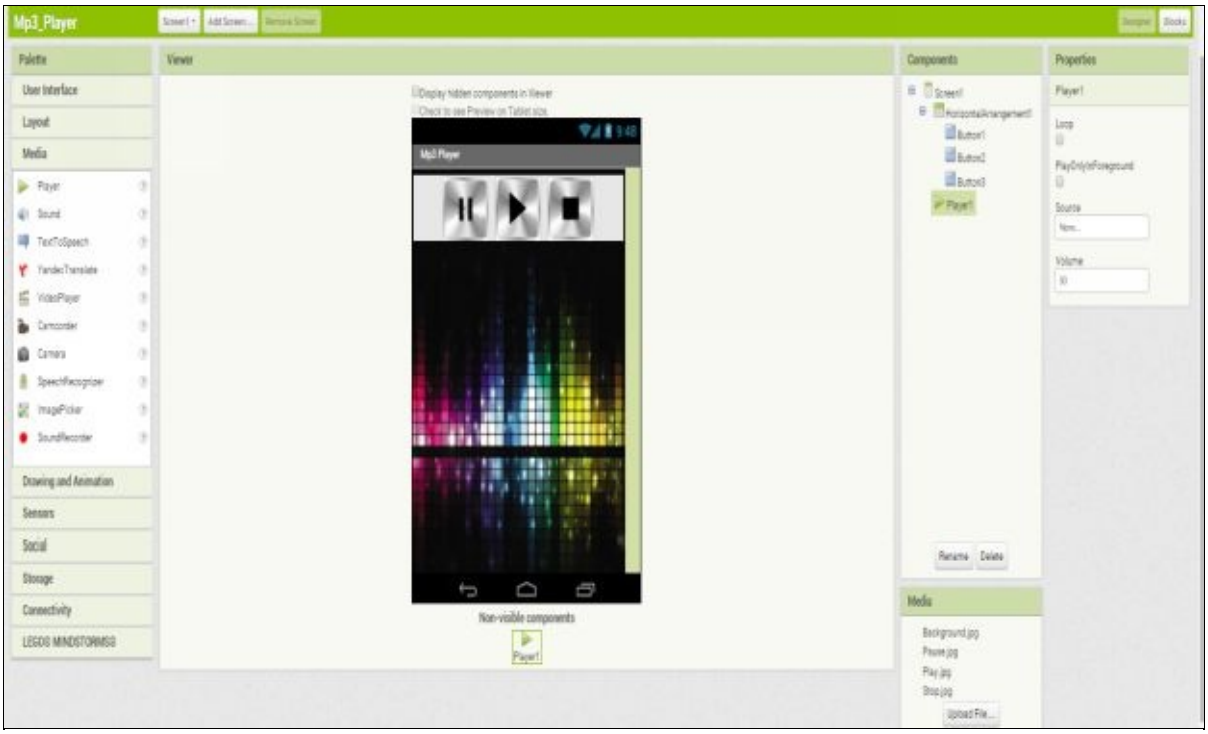
Button2:



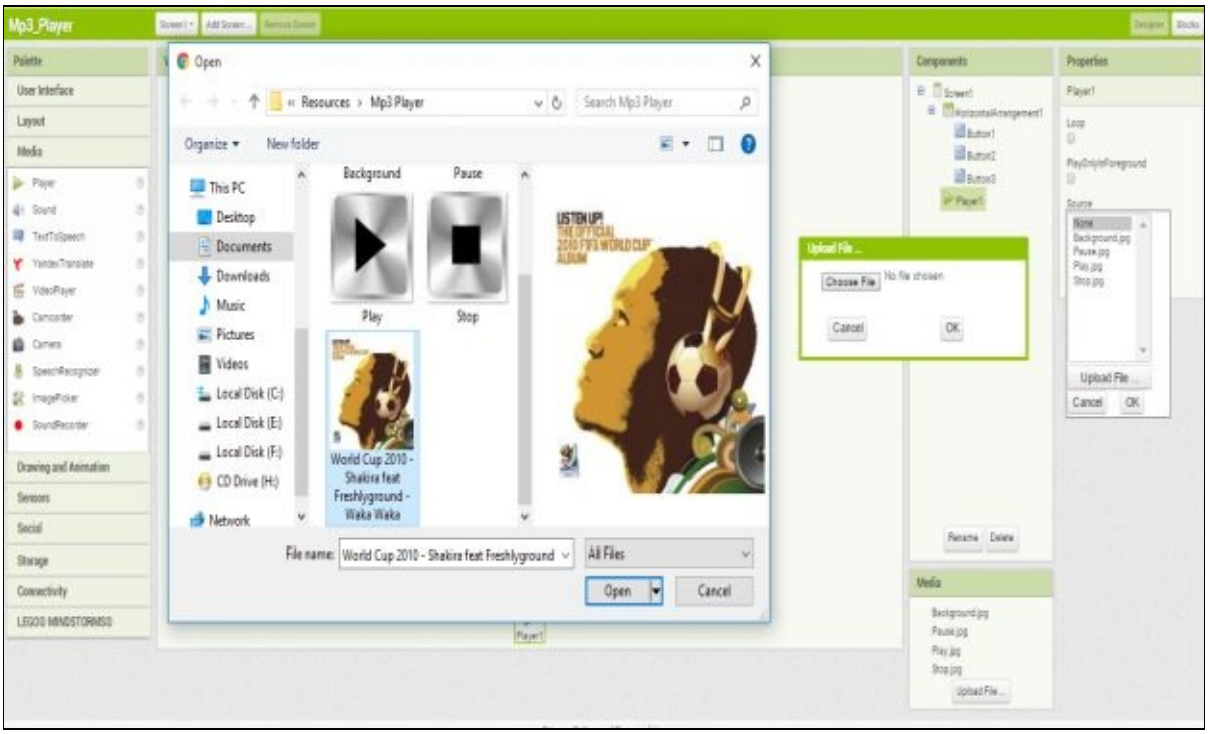
Button3:

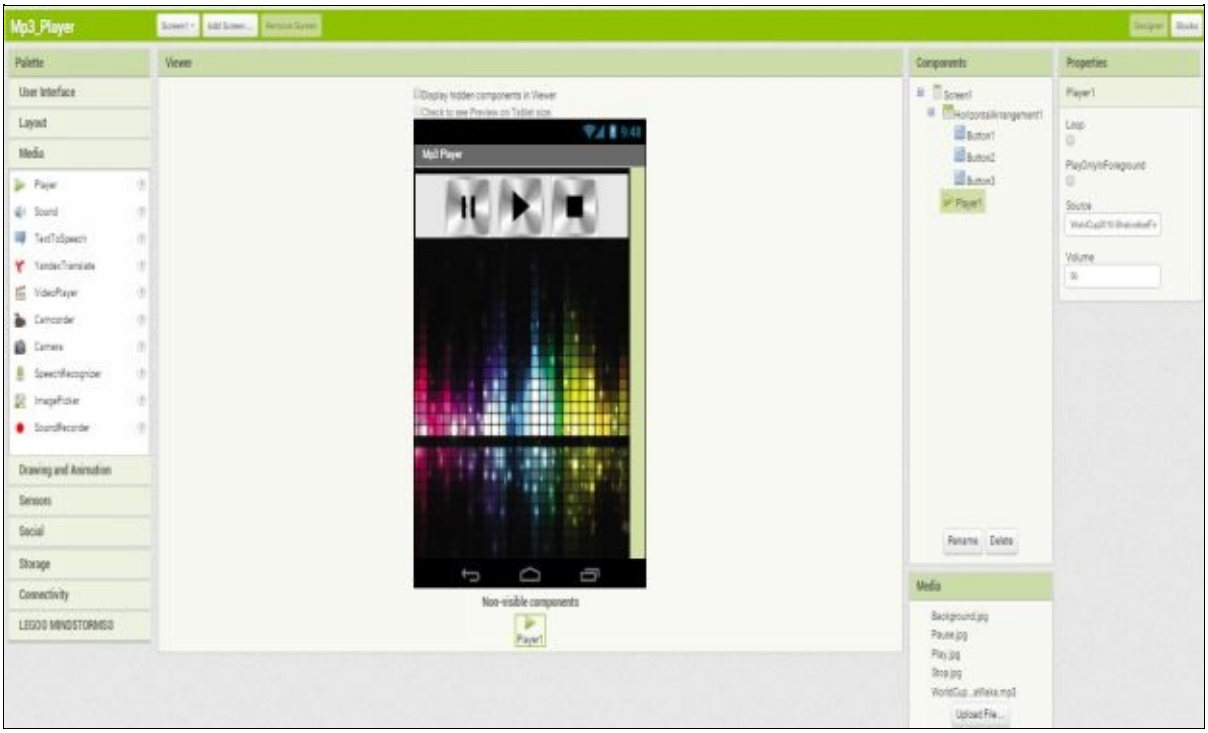


8. Now Drag a Player (non-visible) component from Palette to Viewer.

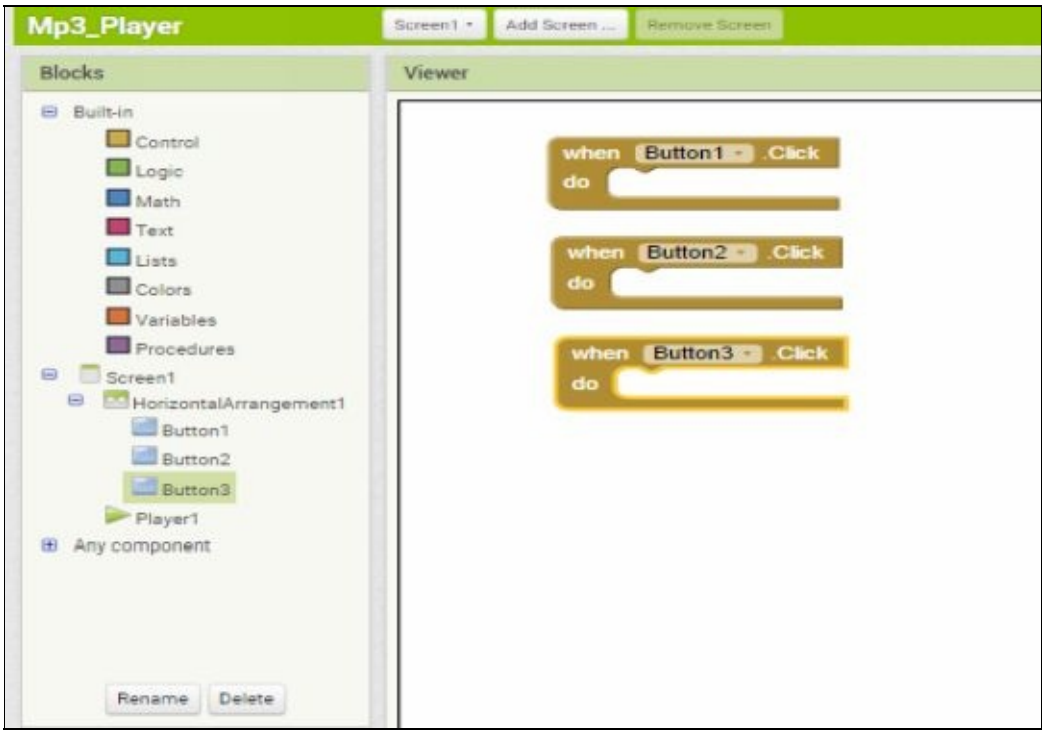


9. Change Player1 source to Mp3 file.

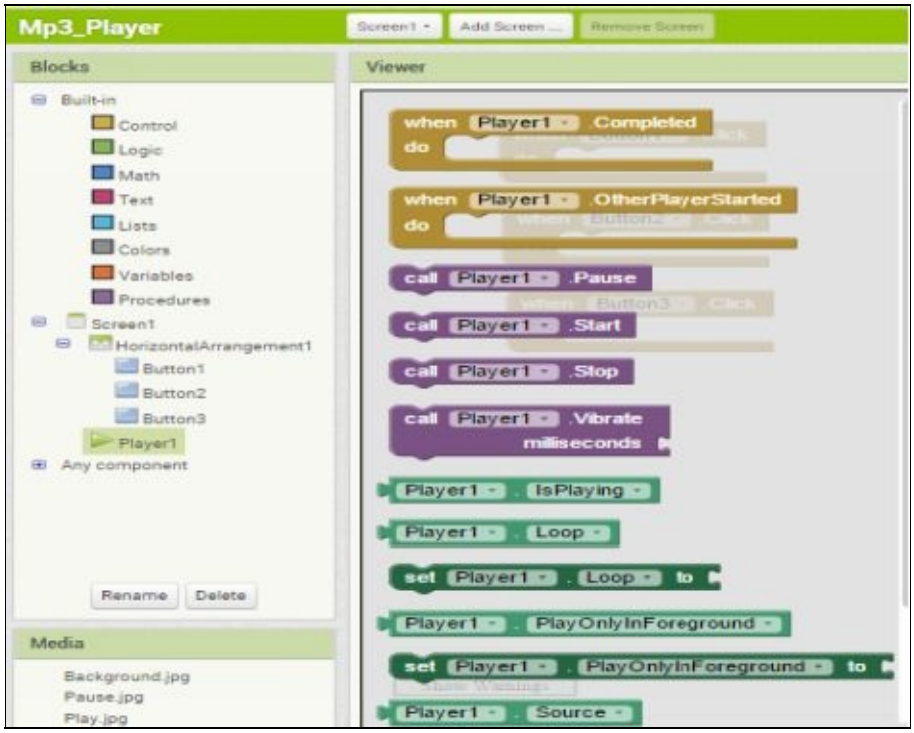


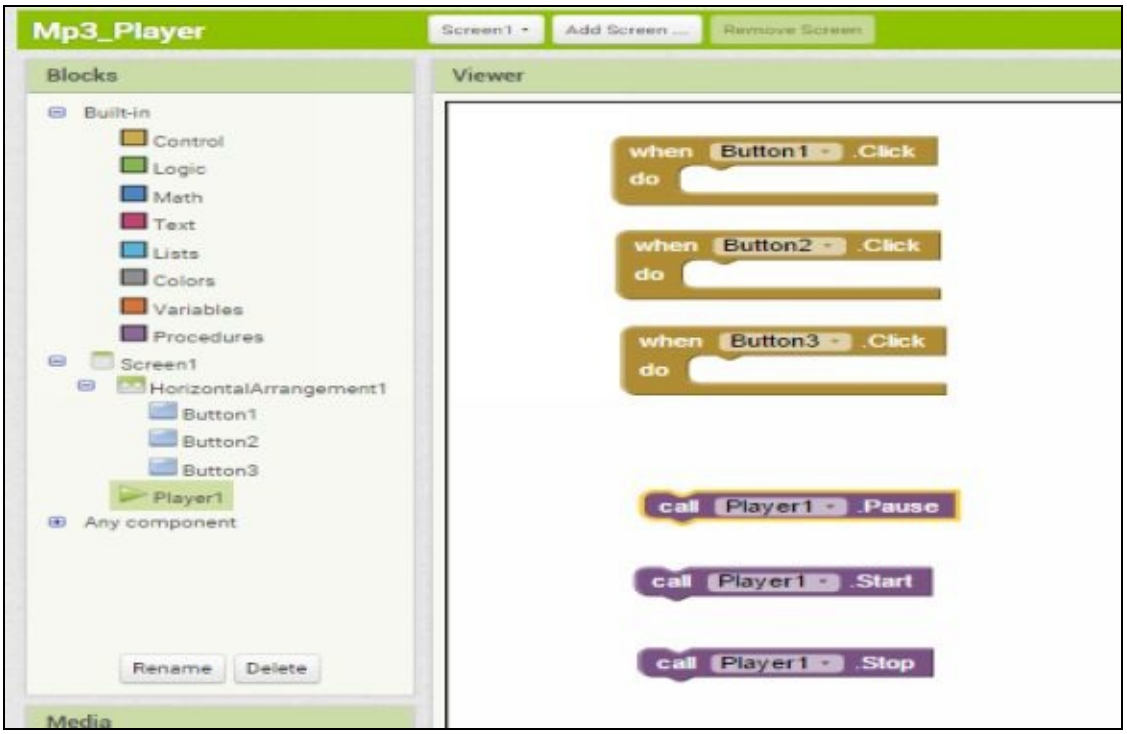


10. Now go to Blocks and select these 3 blocks shown below.

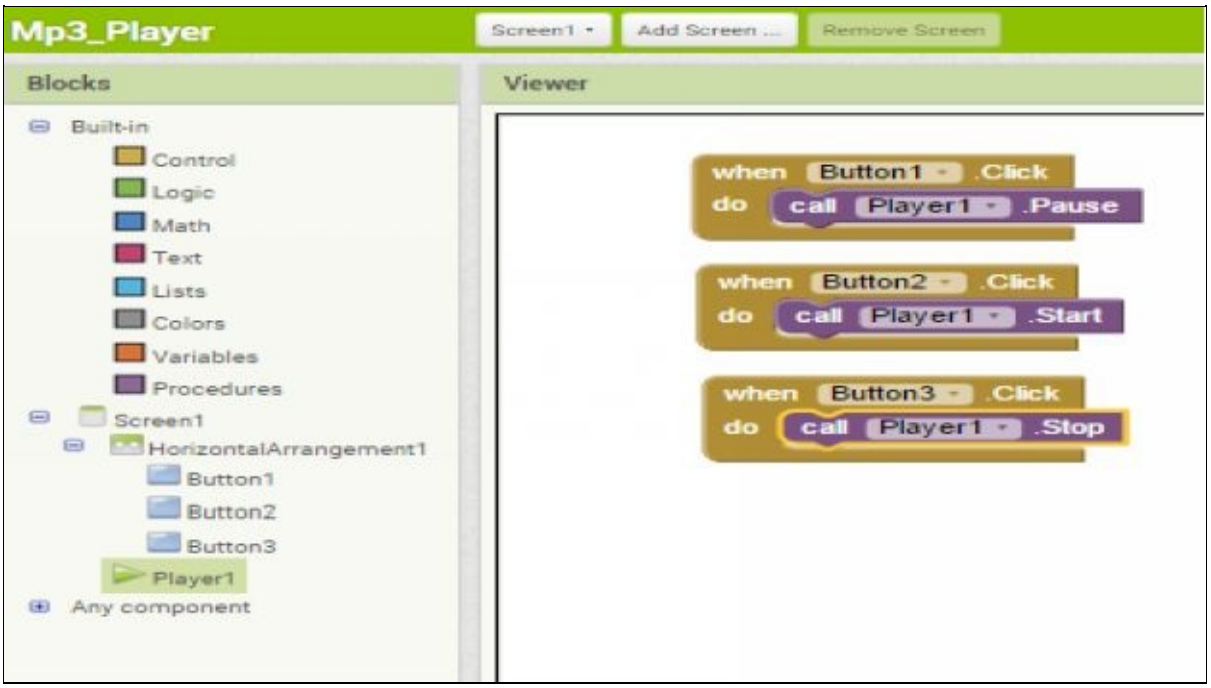


11. Click on Player1 and select call Player1 .Stop blocks.

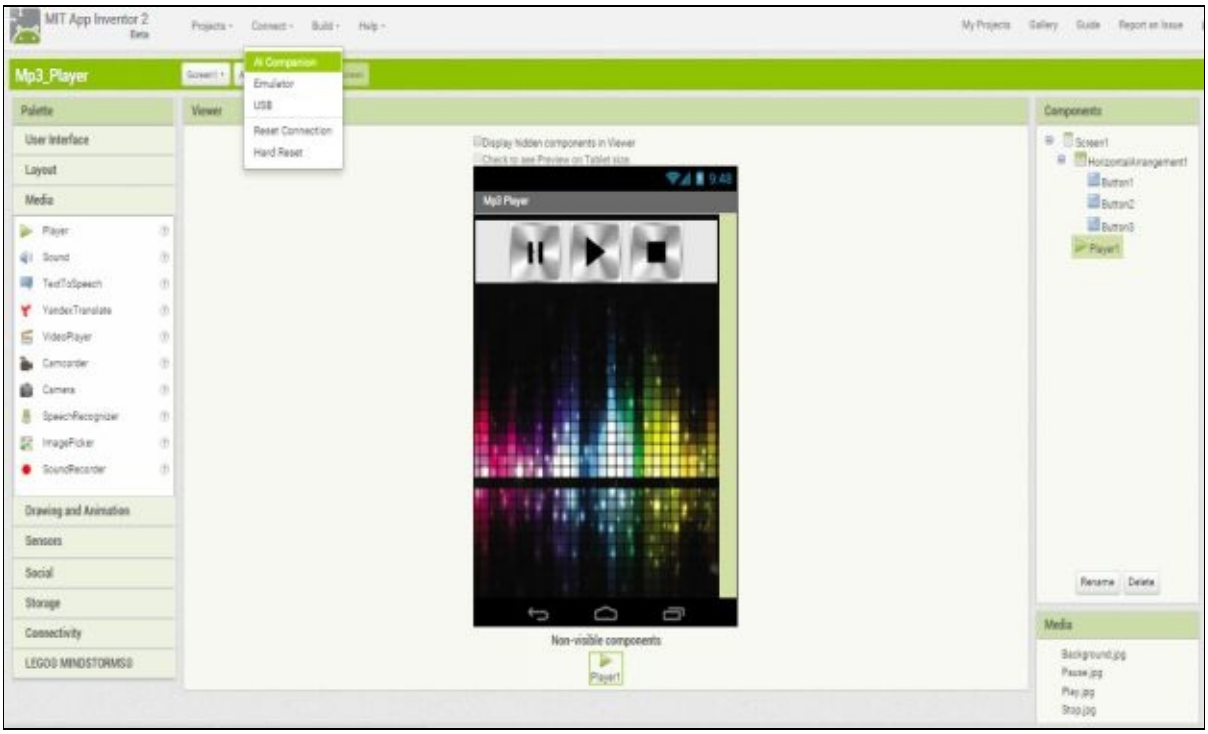




12. Attach the blocks as shown below. So when you click Button1 your Player will pause and when you click Button2 your Player will start and when you click Button1 your Player will stop.



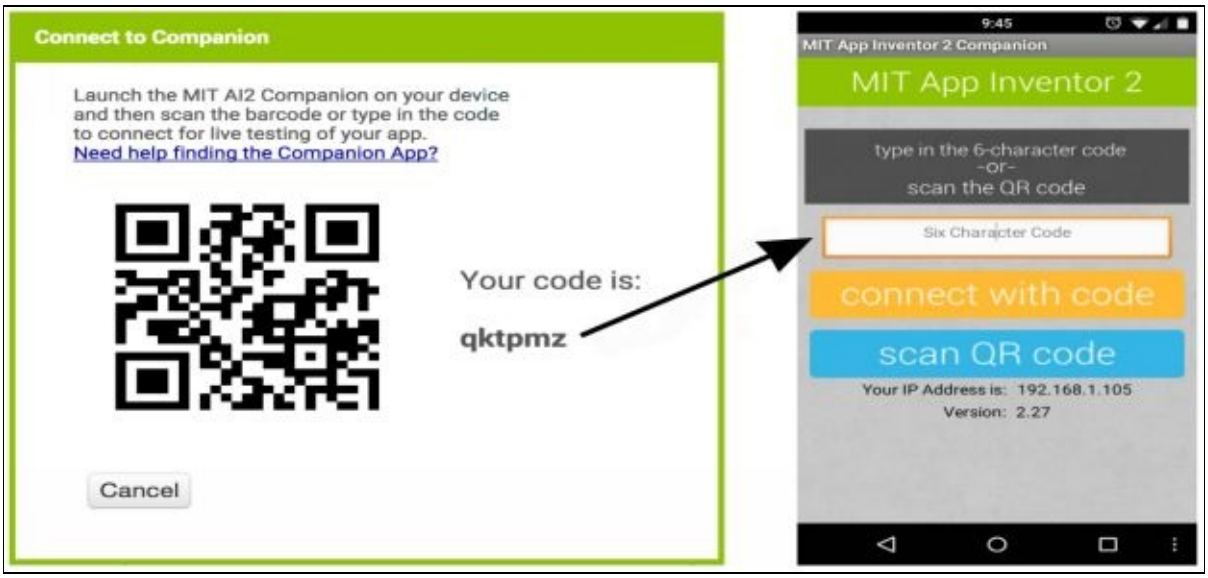
13. Now go to Designer and click on Connect and then click on AI Companion.



14. Now open MIT App Inventor 2 Companion in your mobile/Tablet.

And enter the code or scan the QR code from your mobile and click on connect with code.

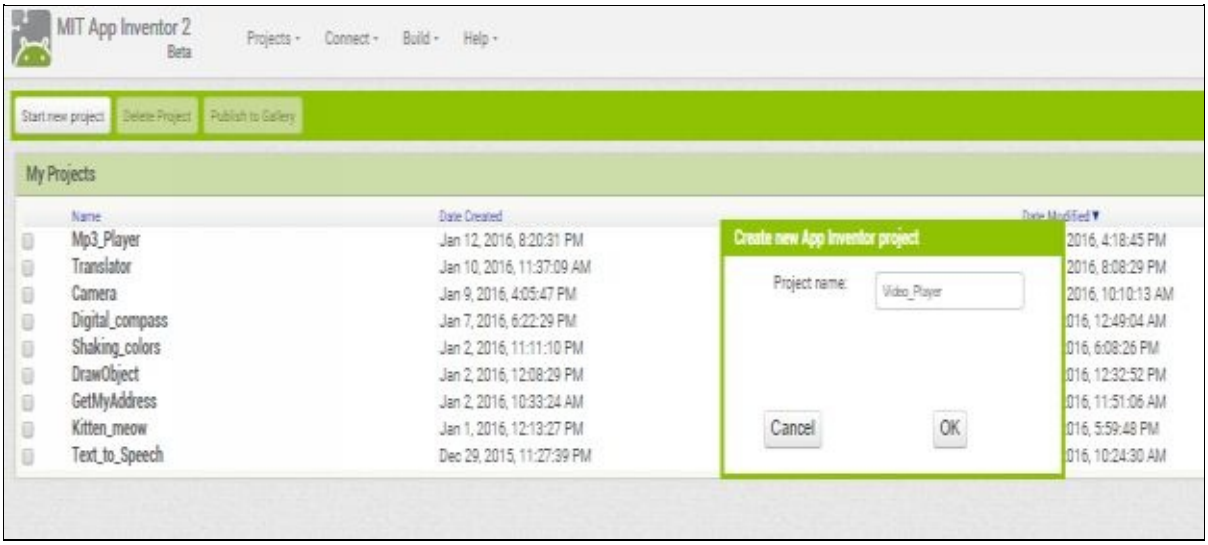
Note:-Your Mobile/Tablet should be connected to same wi-fi/network to which your computer/Laptop is connected.



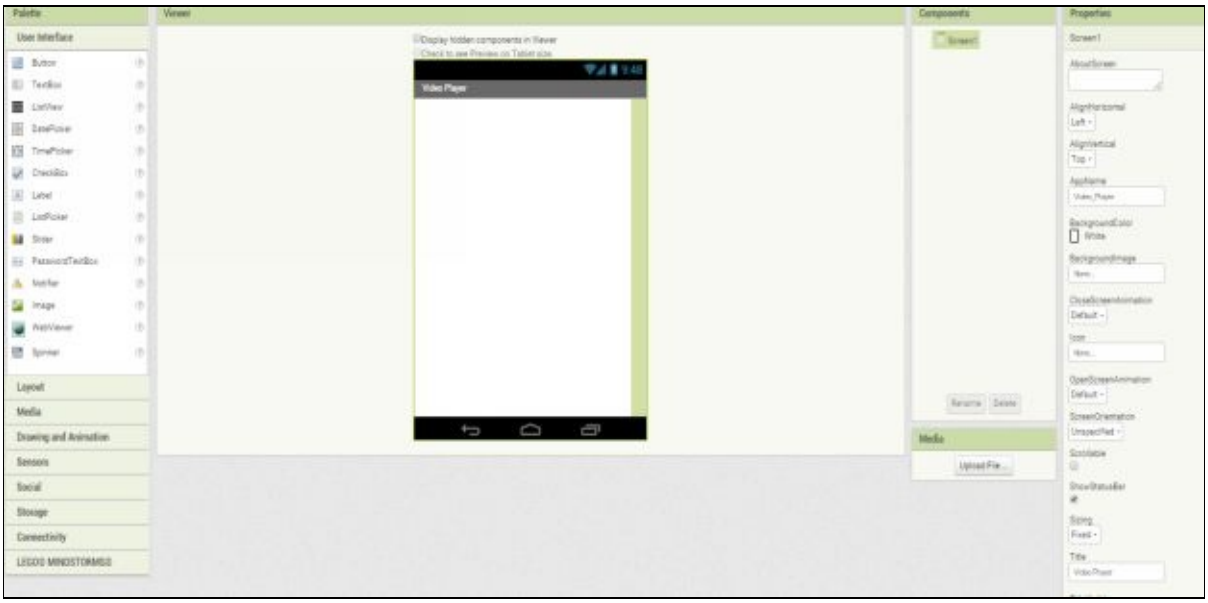
15. Now you should see your app in your phone for live testing. Click on play Button to test your Mp3_Player App.

Chapter 13: Video Player App

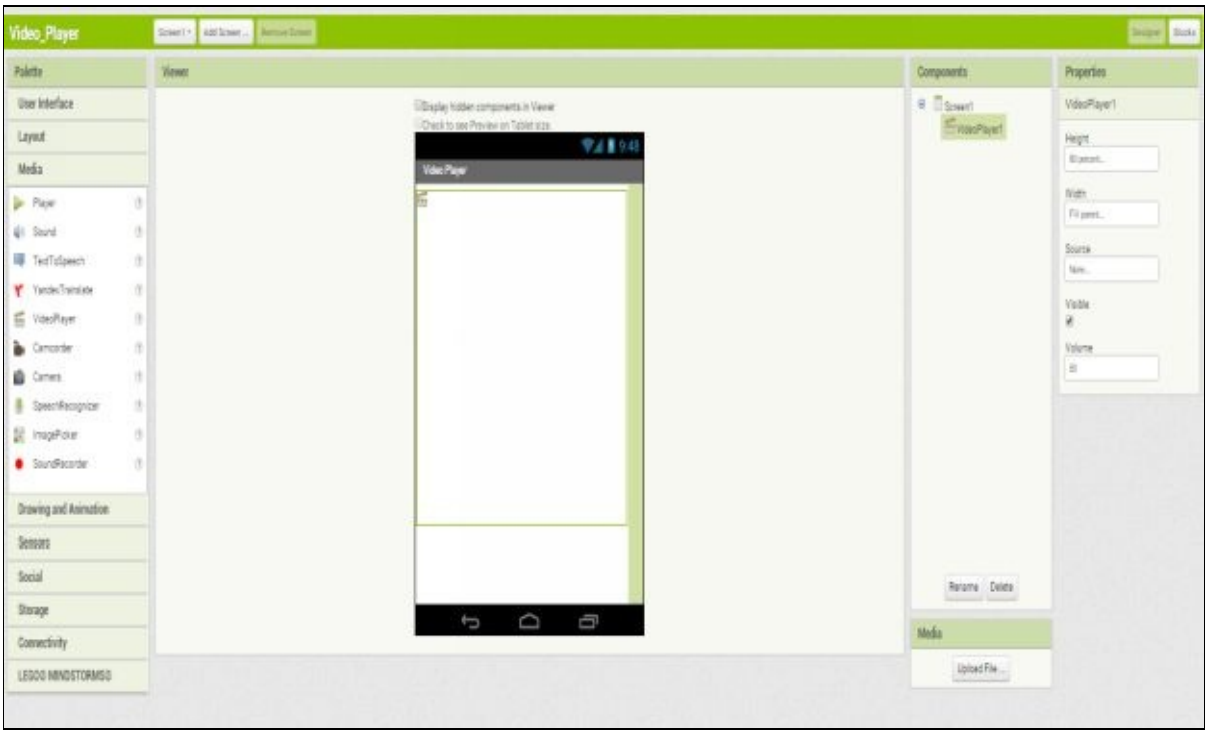
1. Click on Start new project and give it name Video_Player and click OK.



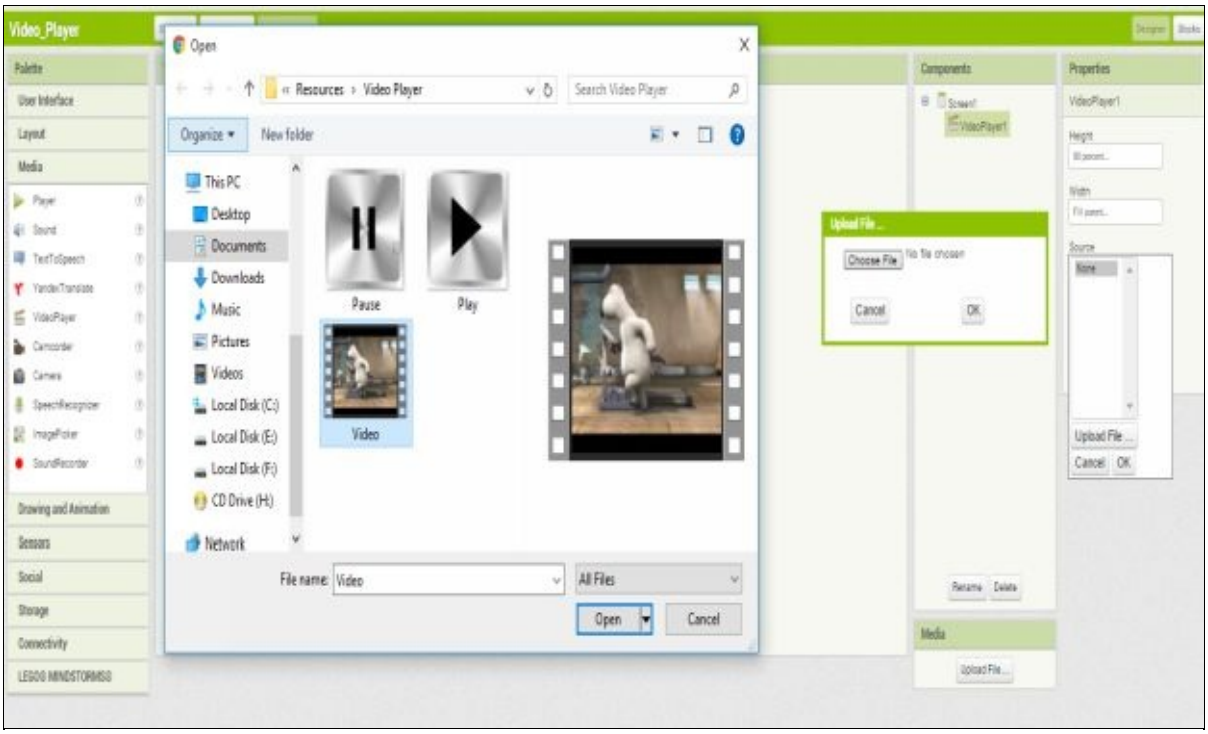
2. Now change Screen1 Title to Video Player.

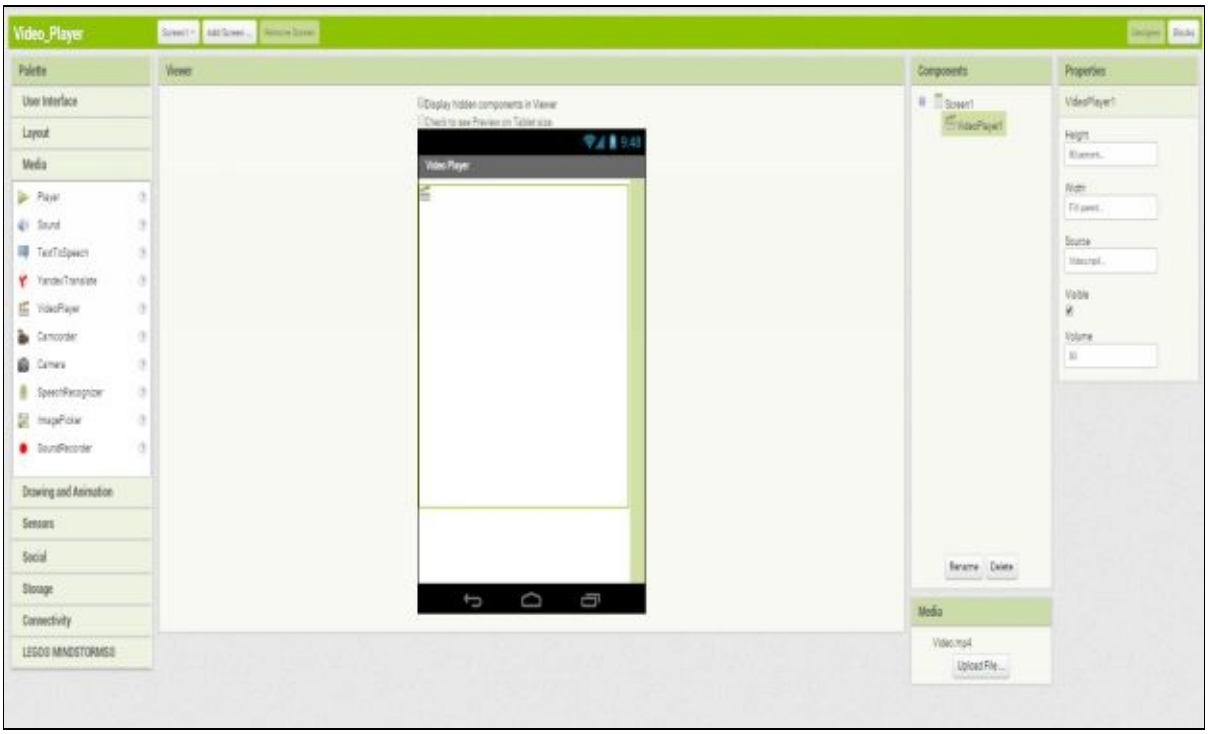


3. Drag VideoPlayer component from Palette to Viewer and set its Height to 80 percent and Width to Fill Parent.

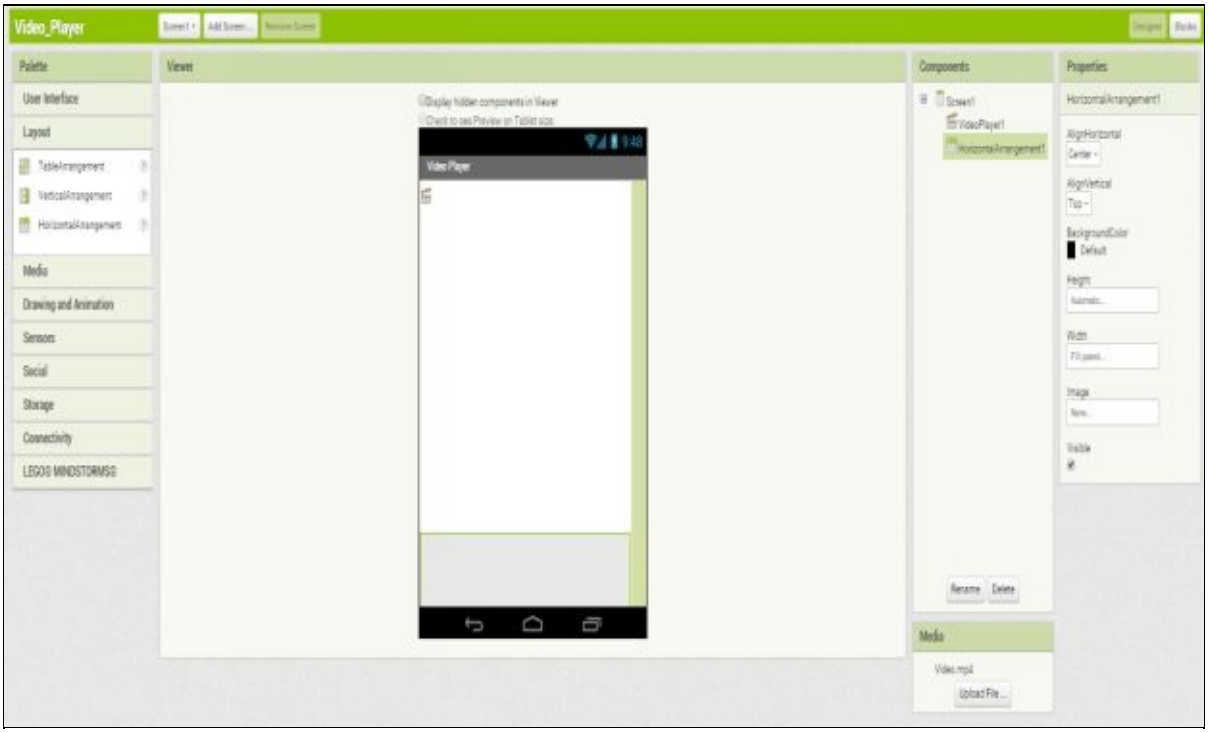


4. Change VideoPlayer1’s Source to Video.mp4.

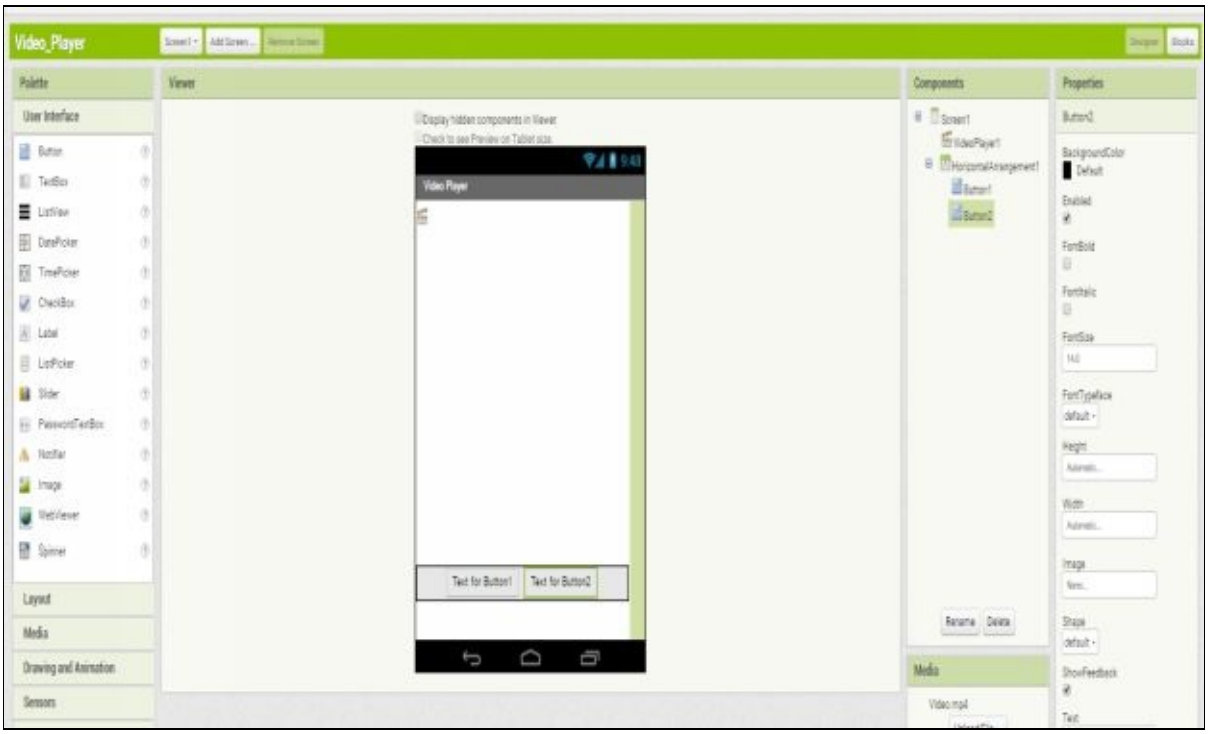




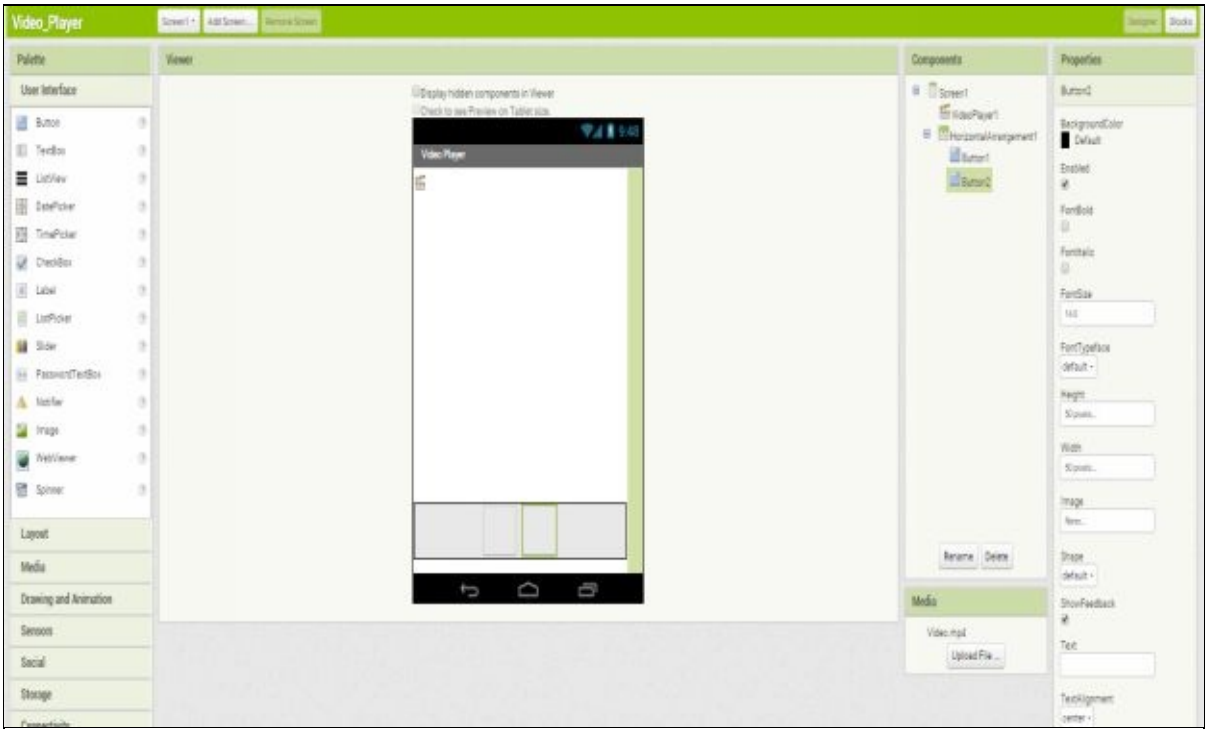
5. Drag a HorizontalArrangement component from Palette to Viewer and set its Width to Fill Parent and AlignHorizontal Property to Center.



6. Now drag two Buttons from Palette to Viewer inside HorizontalArrangement1.

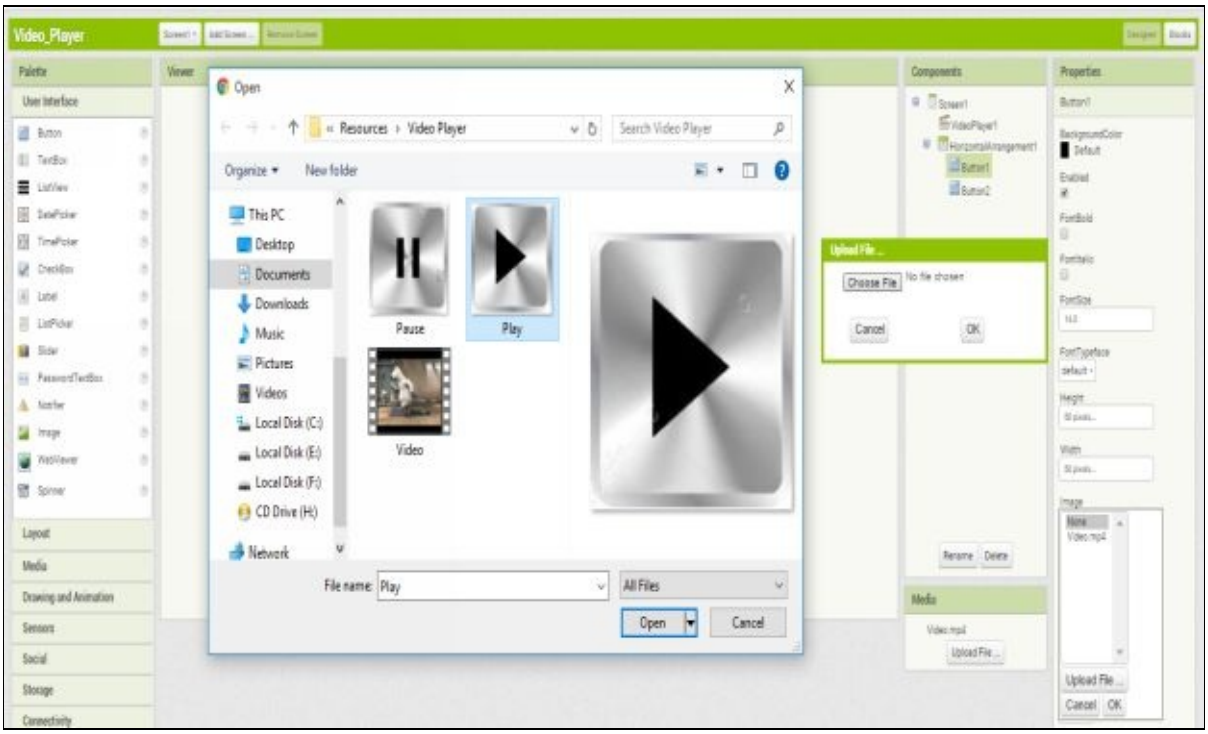


7. Change both Button's Width and Height to 50 pixels and Delete the Texts for both the buttons.

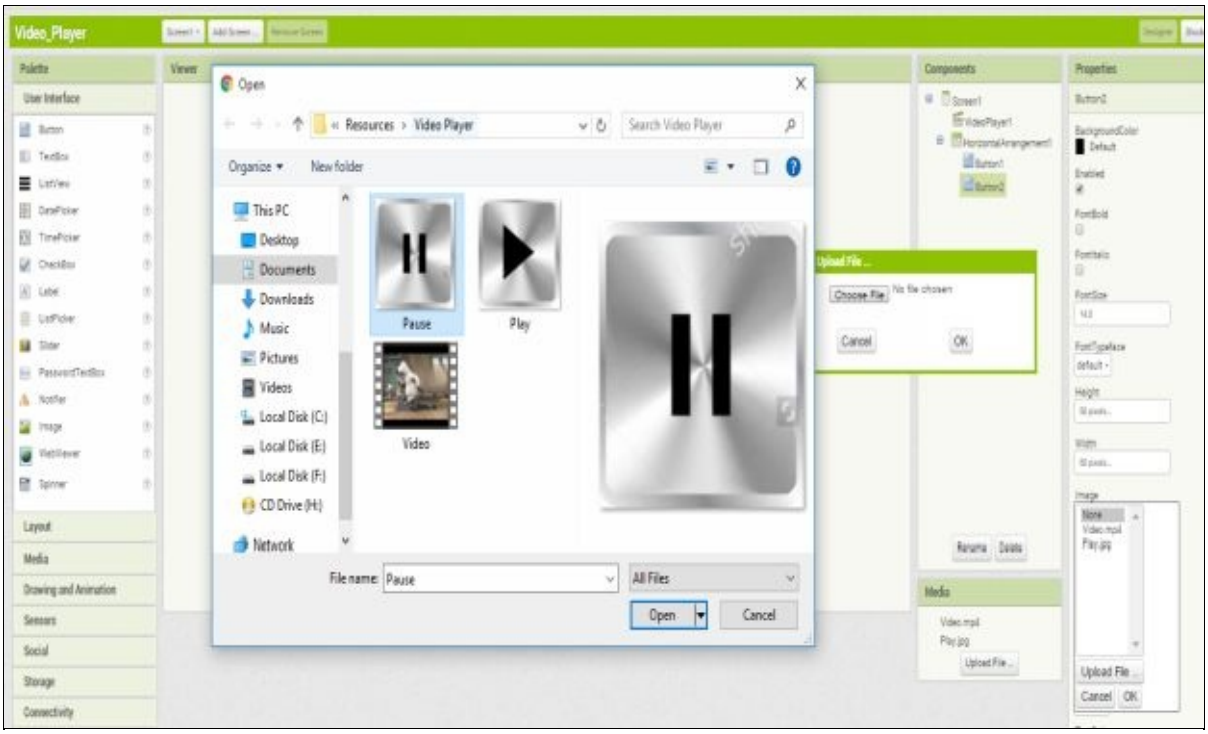


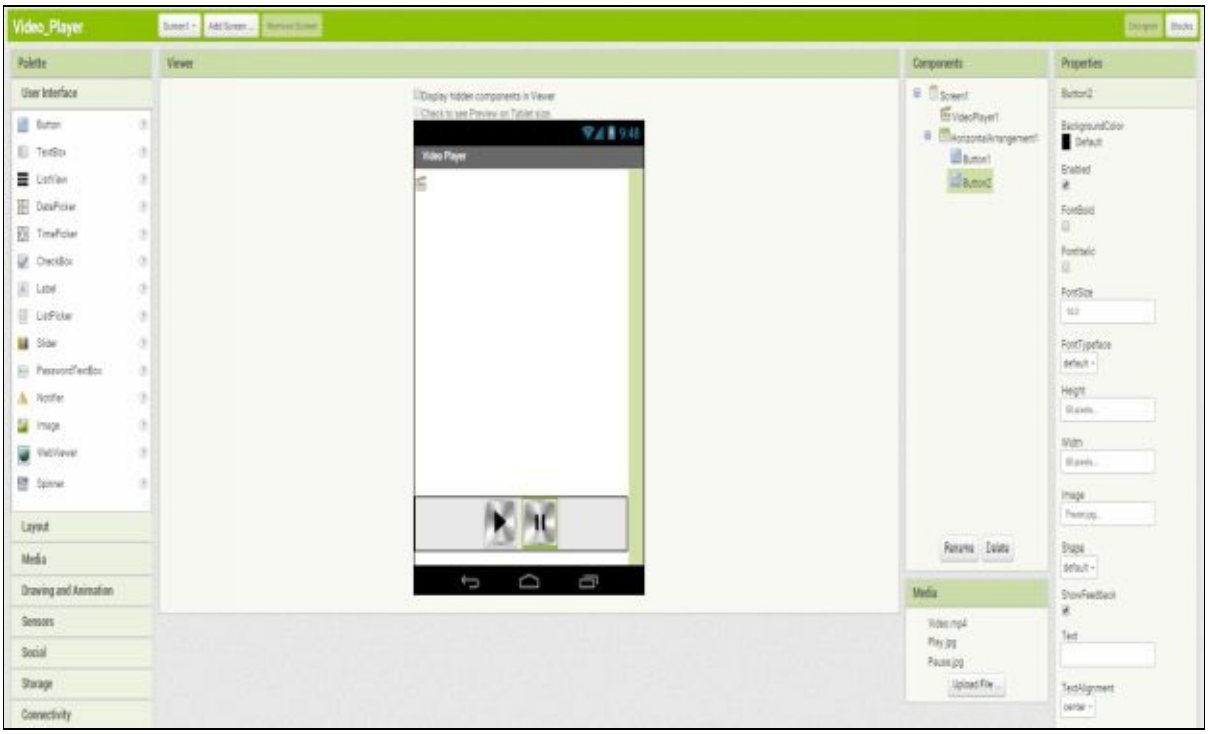
8. Change Button's Image as shown below.

Button1:


Video_Player
Open
Resources › Video Player
Search Video Player
Organize
New folder
This PC
Desktop
Documents
Downloads
Music
Pictures
Videos
Local Disk (C:)
Local Disk (E:)
Local Disk (F:)
CD Drive (H:)
Network
Pause
Play
Video
File name: Play
All Files
Open
Cancel
Upload File ...
Choose File
No file chosen
Cancel
OK
Components
Properties
Media
Rename
Delete
Upload File ...
Cancel
OK

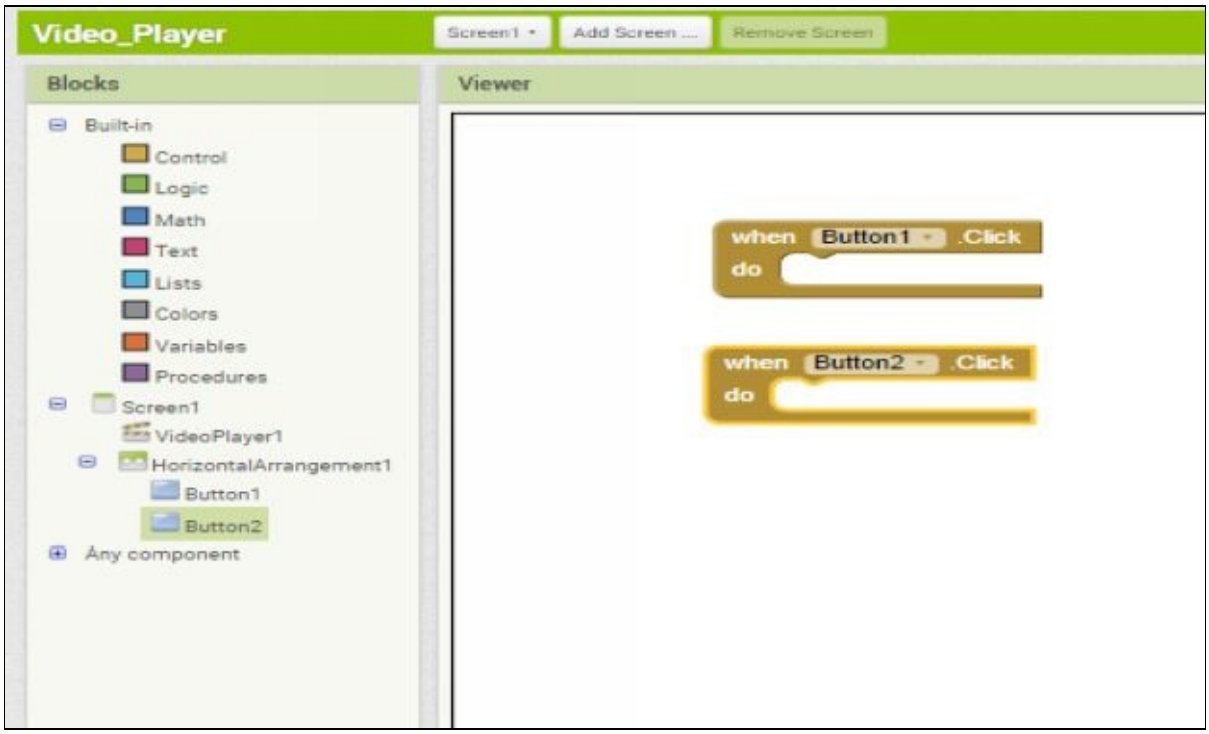
Button2:

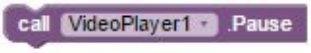

Video_Player
Open
Resources › Video Player
Search Video Player
Organize
New folder
This PC
Desktop
Documents
Downloads
Music
Pictures
Videos
Local Disk (C:)
Local Disk (E:)
Local Disk (F:)
CD Drive (H:)
Network
Pause
Play
Video
File name: Pause
All Files
Open
Cancel
Upload File ...
Choose File
No file chosen
Cancel
OK
Components
Properties
Media
Rename
Delete
Upload File ...
Cancel
OK



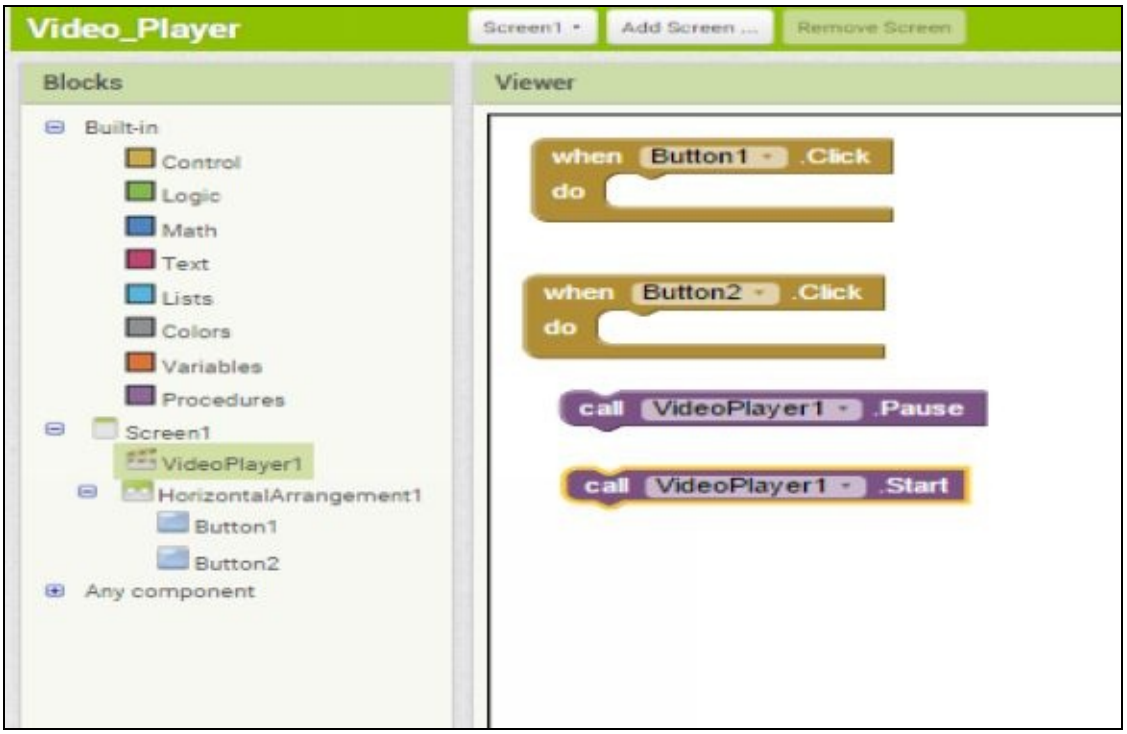
9. Go to Blocks and click on Button1 component and select  block

10. Click on Button2 component and select  block.

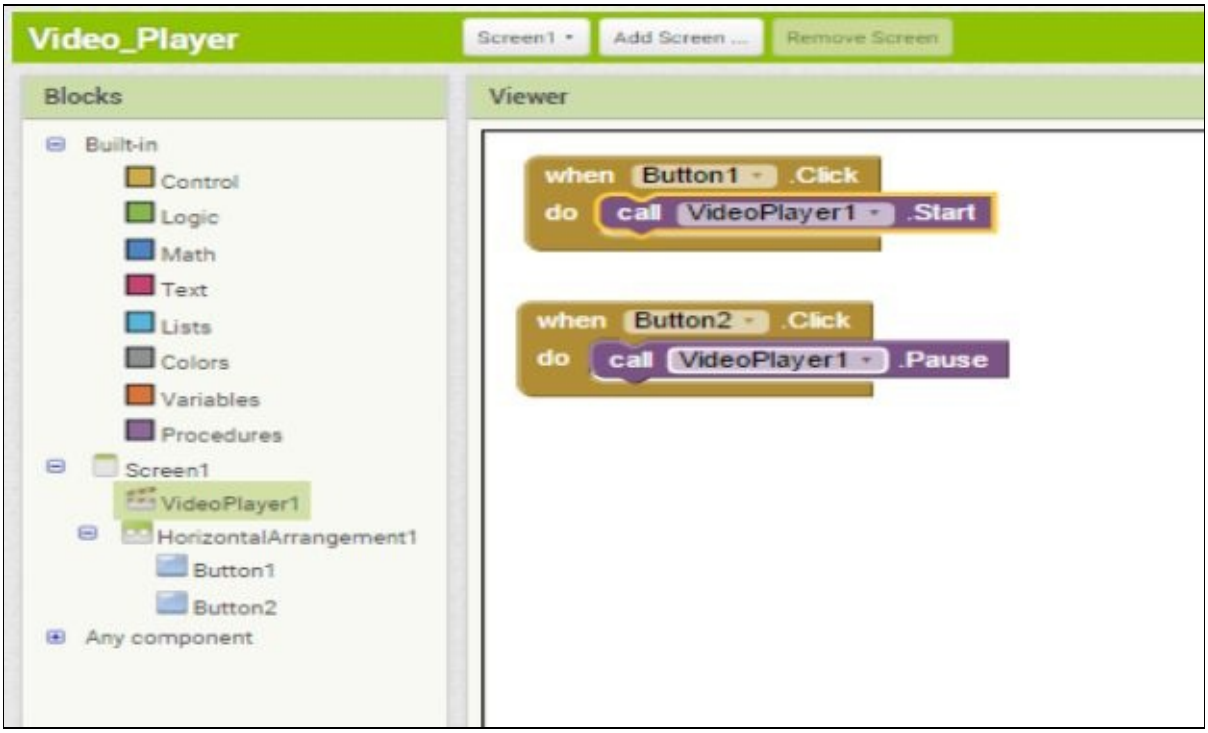


11. Now click on VideoPlayer1 component and select 

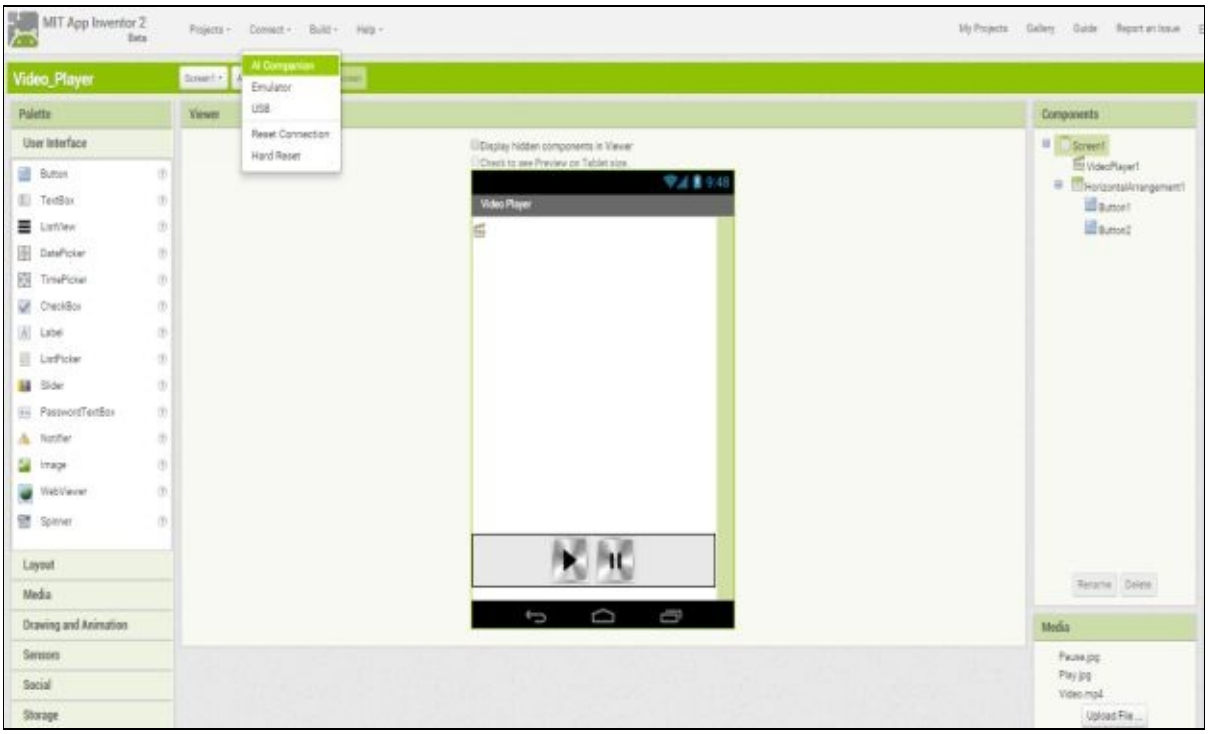
blocks.

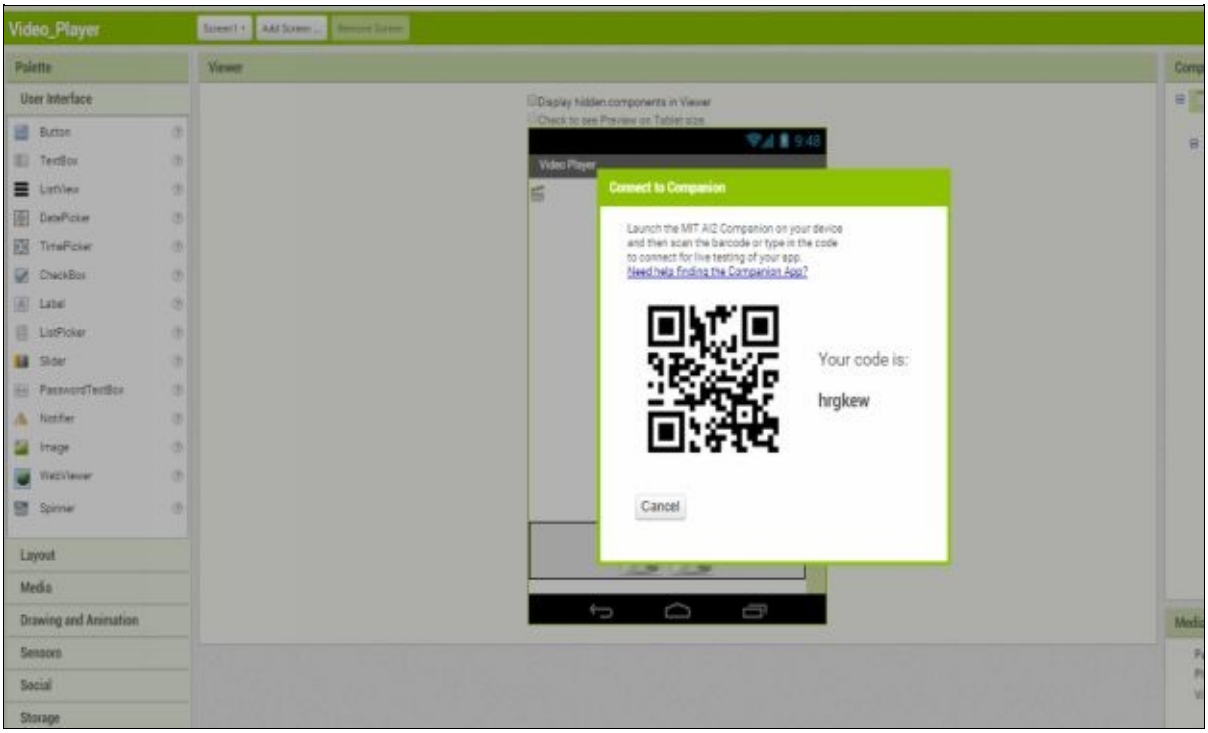


12. Now attach the blocks as shown below. So when you click Button1 your VideoPlayer will Start and when you click Button2 your VideoPlayer will Pause.



13. Now go to Designer and click on Connect then click on AI Companion.

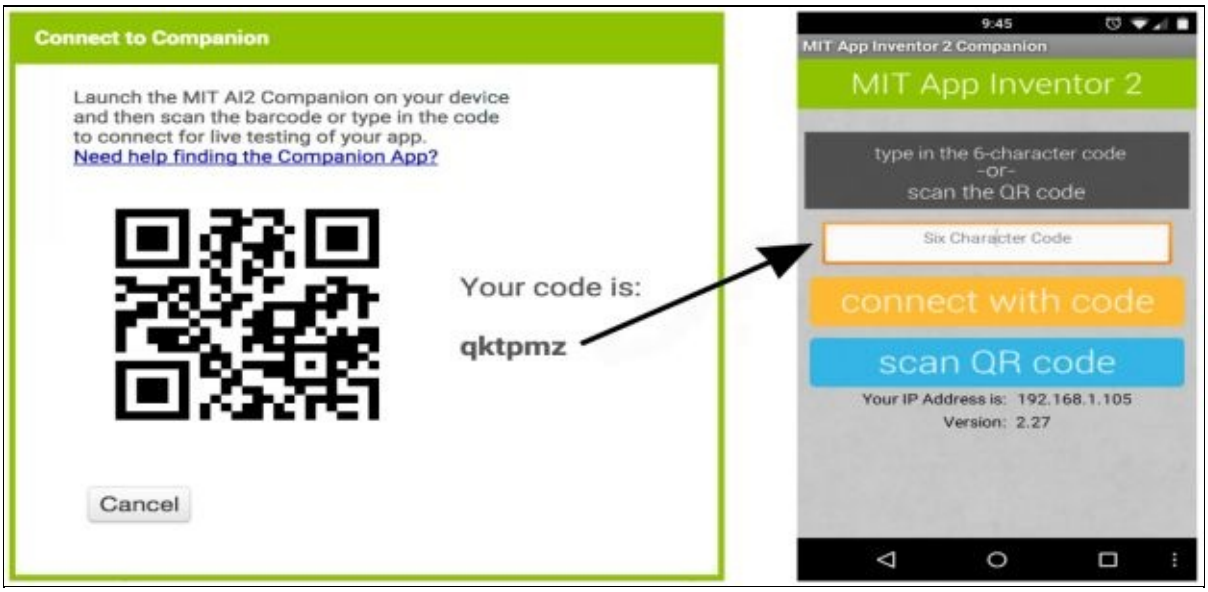




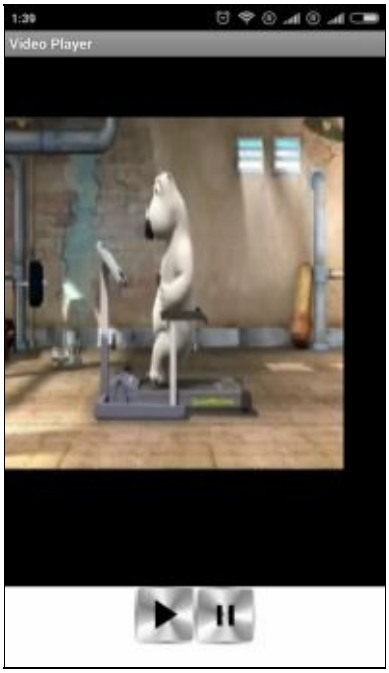
14. Now open MIT App Inventor 2 Companion in your mobile/Tablet.

And enter the code or scan the QR code from your mobile and click on connect with code.

Note:-Your Mobile/Tablet should be connected to same wi-fi/network to which your computer/Laptop is connected.

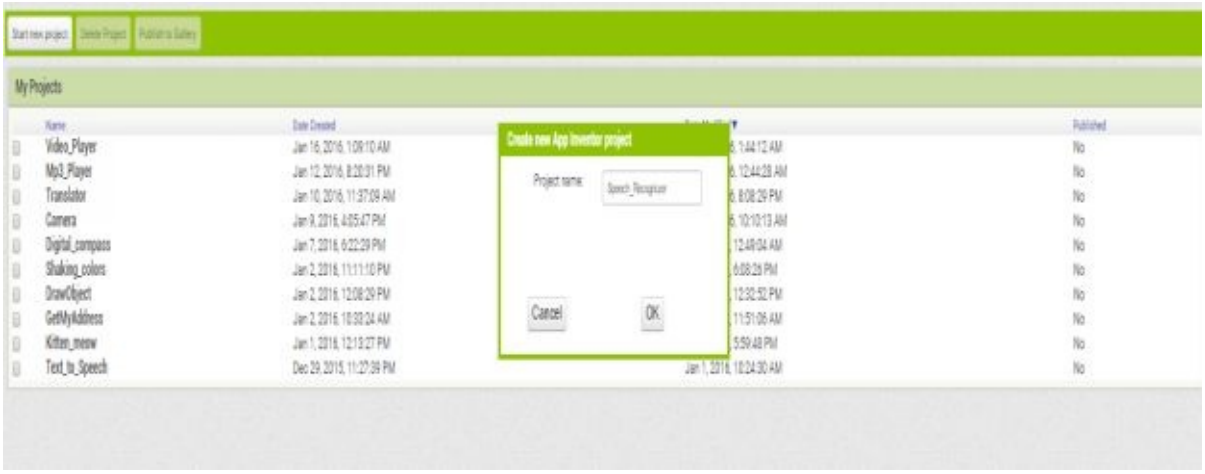


15. Now you should see your app in your phone for live testing. Click on Play Button to test your Video_Player App.

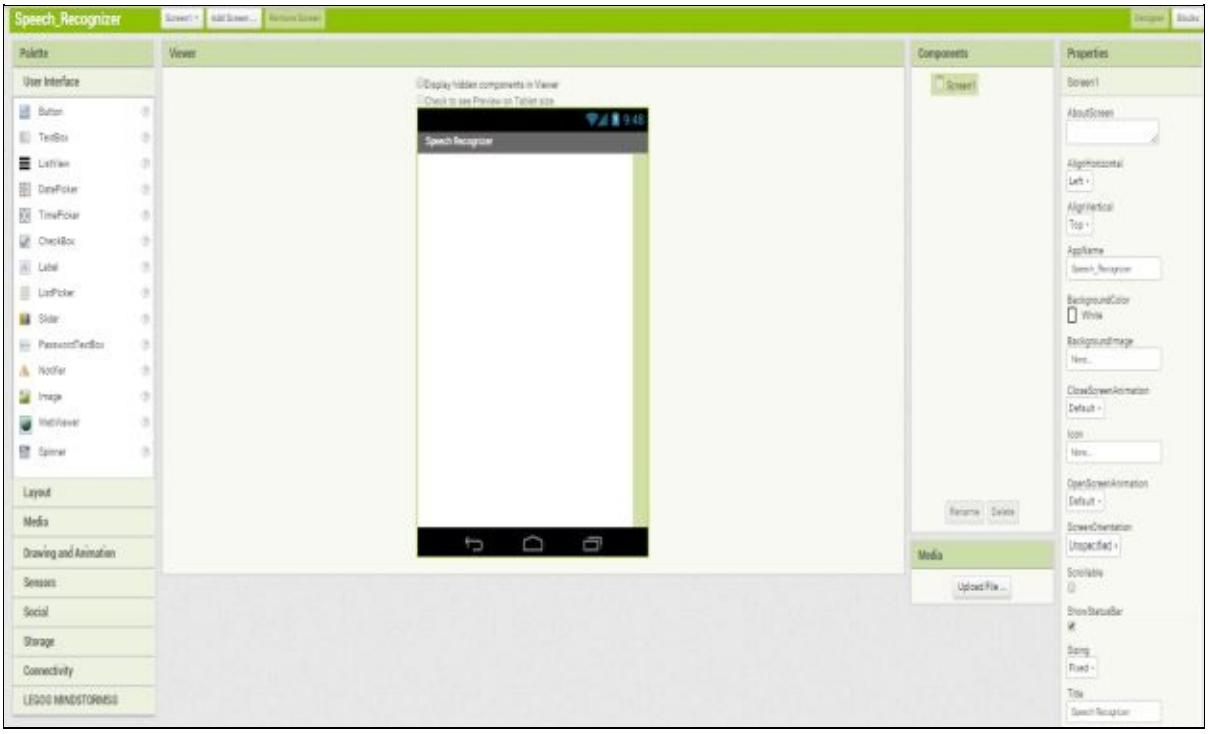


Chapter 14: Speech Recognizer App

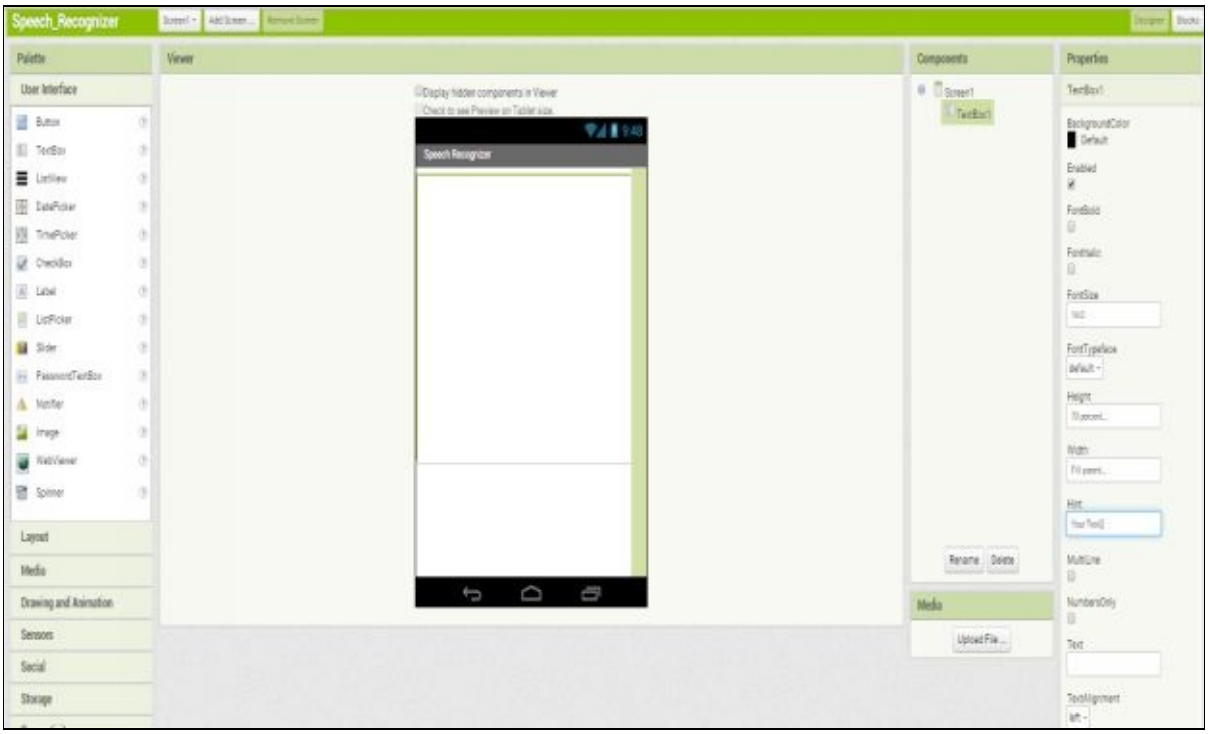
1. Click on Start new project and give it name Speech_Recognizer and Click OK.



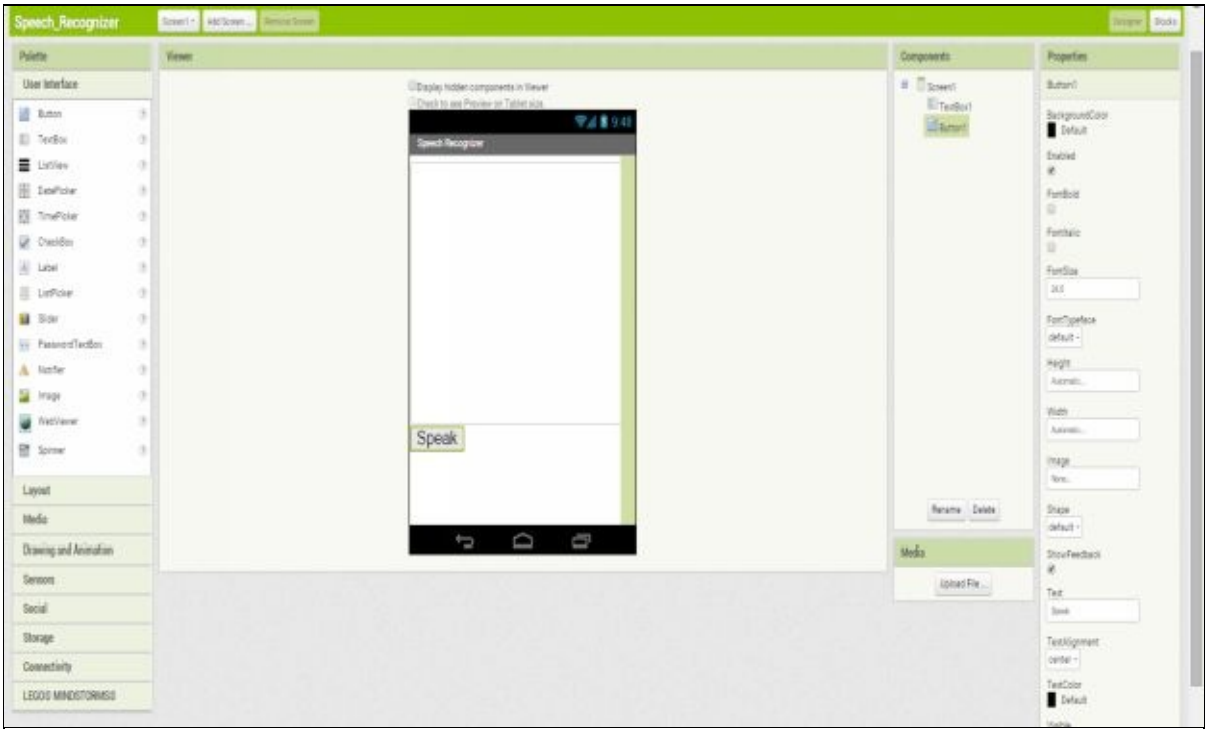
2. Now change Screen1 Title to Speech Recognizer.



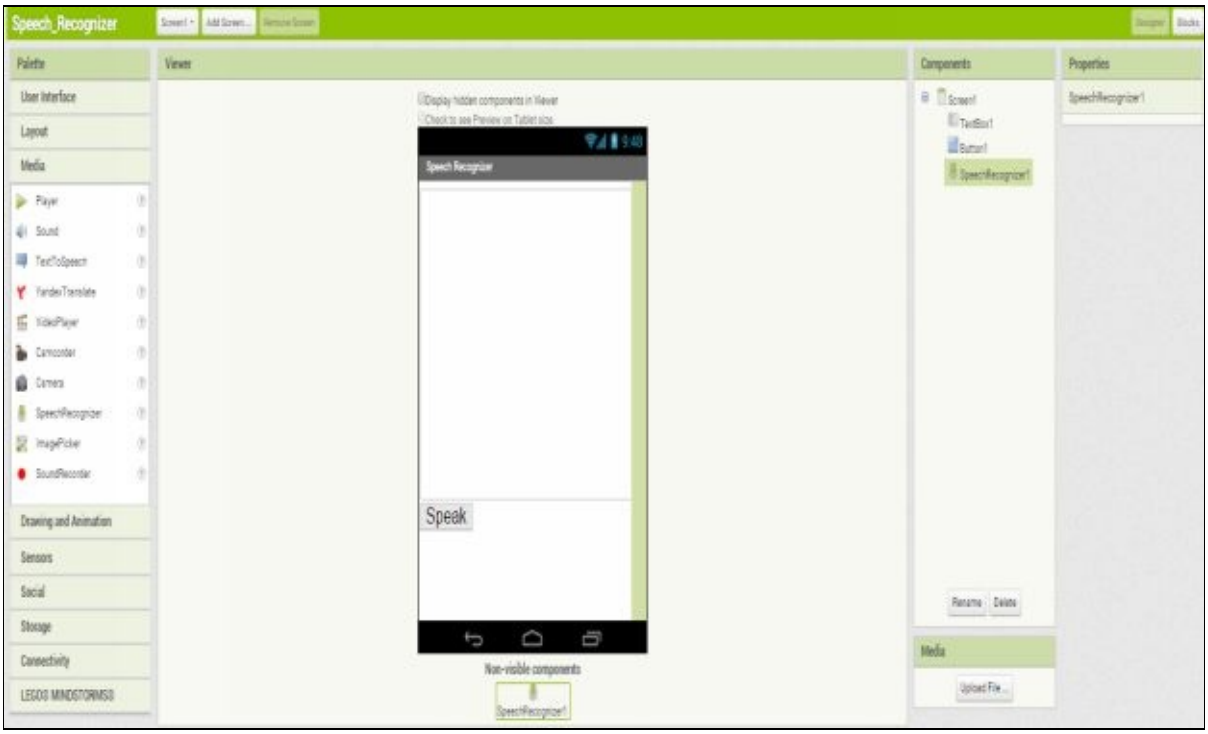
3. Now Drag a TextBox from Palette to Viewer screen and set its Width to Fill Parent and Height to 70 percent and change Hint to 'Your Text'.



4. Now drag a Button from Palette to Viewer screen and change it’s Text to Speak and FontSize to 24.0.

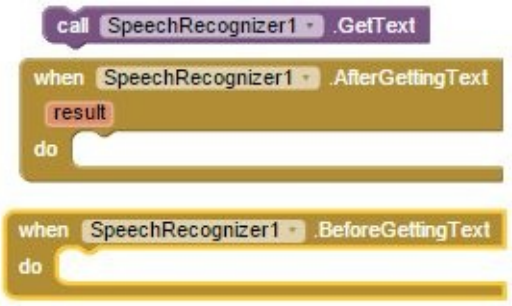


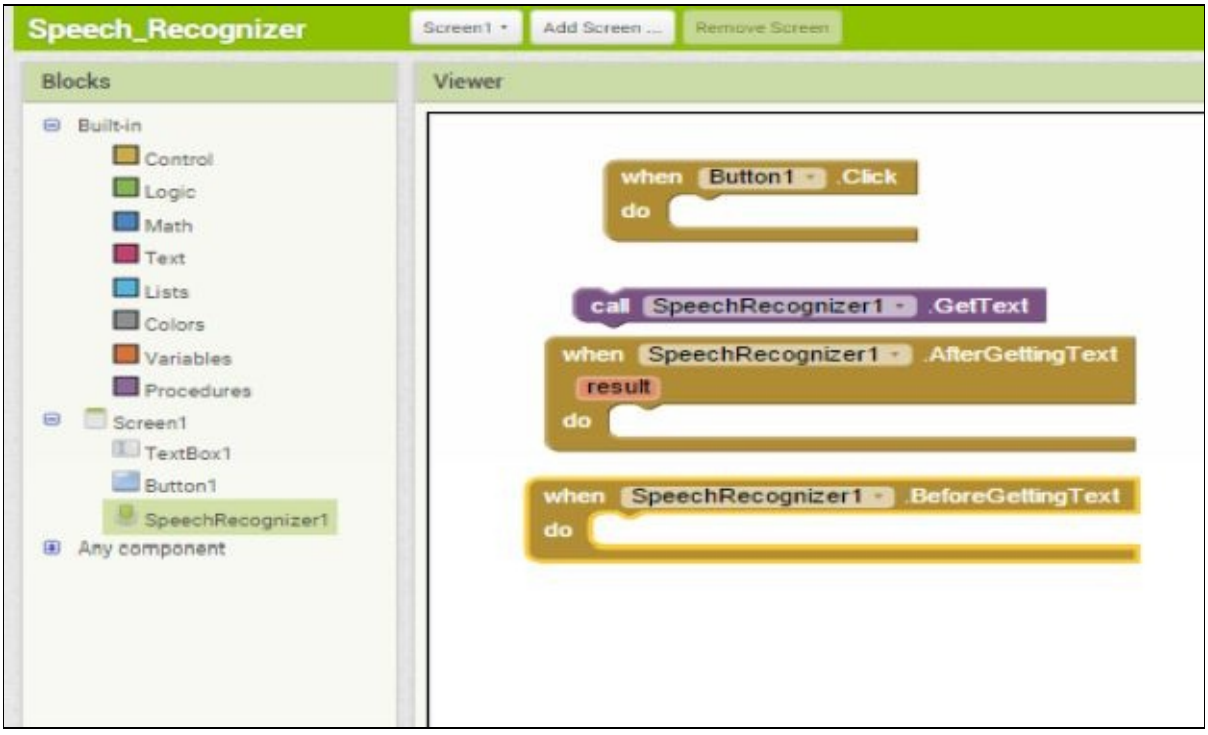
5. Drag a SpeechRecognizer (Non-Visible) component from Palette to Viewer screen.



6. Go to Blocks and Click on Button1 and select  block. This is an event block which will decide what your app will do when this button will be clicked. Here click is an event for Button1 component.

7. Click on SpeechRecognizer1 and select these blocks. These blocks will open your phone's microphone and convert your speech to Text.

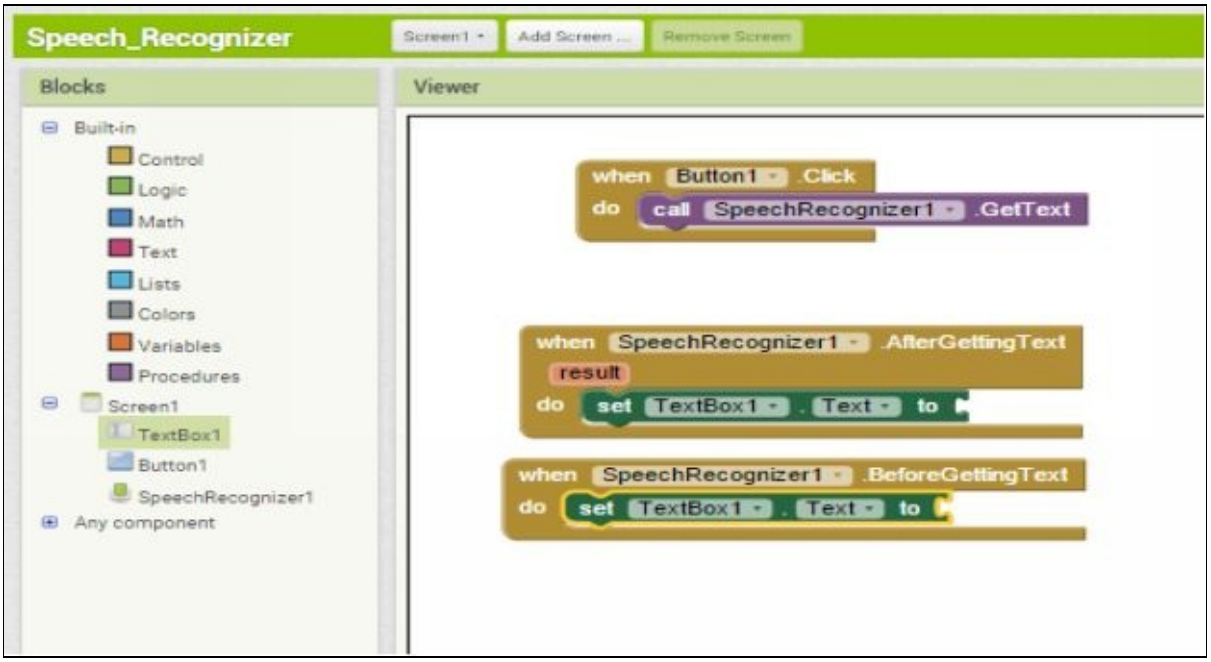




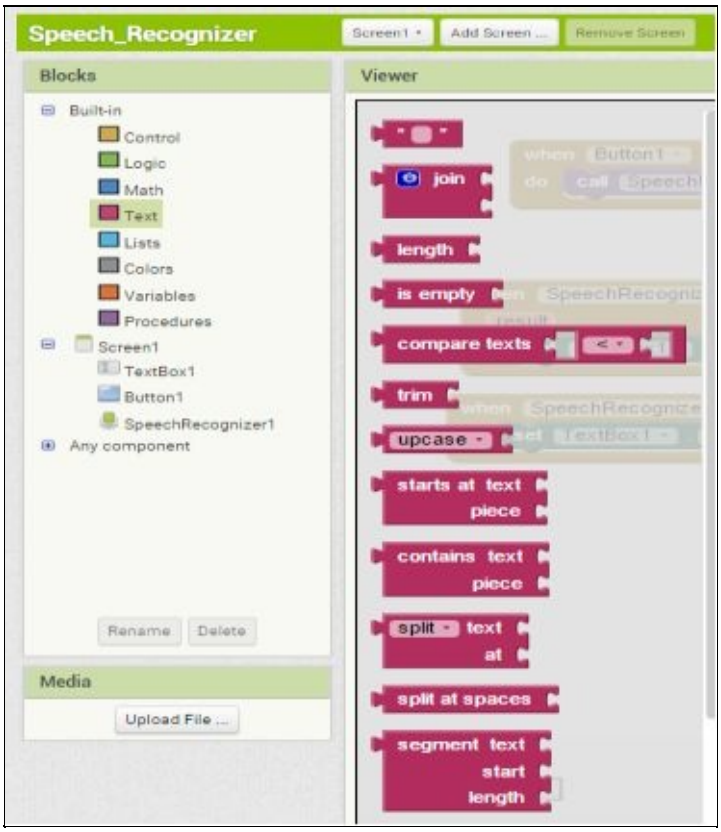
8. Now click on TextBox1 and select 

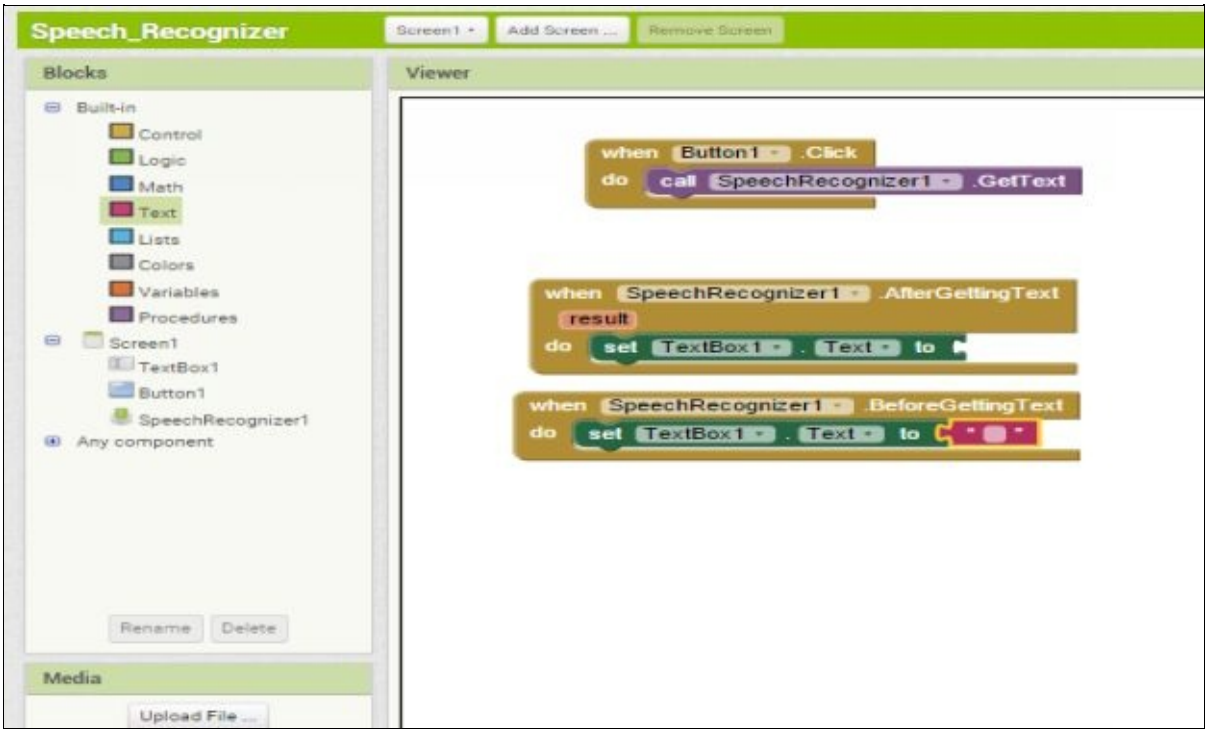
block twice.

9. Attach the blocks as shown below. So when you click on Button1 ,app will open your device's microphone and then you can speak something.



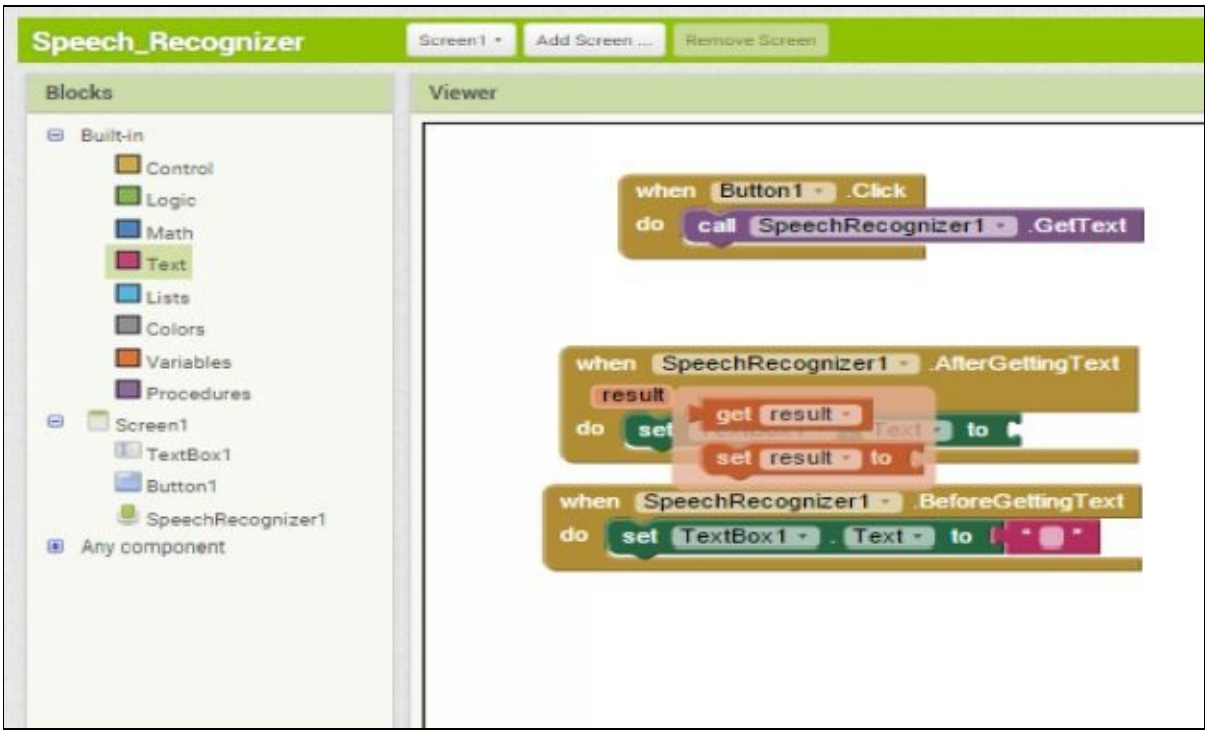
10. Now click on Text under Built-in and select  block.

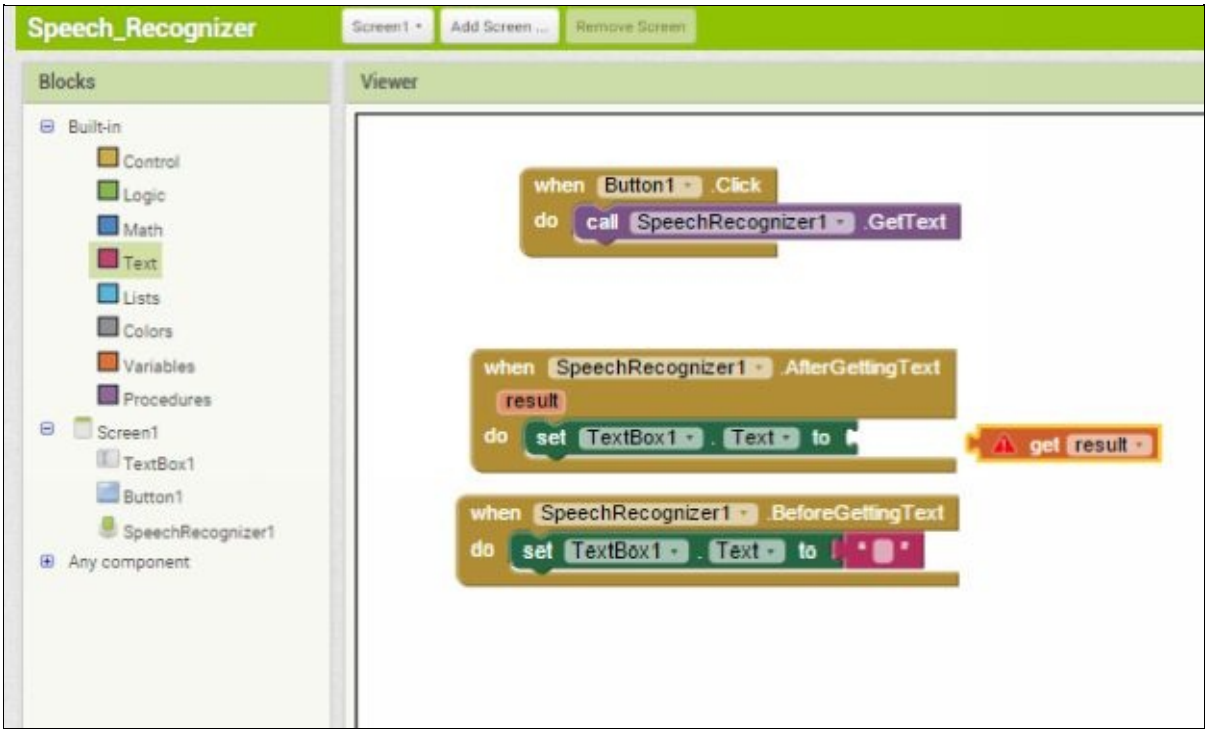




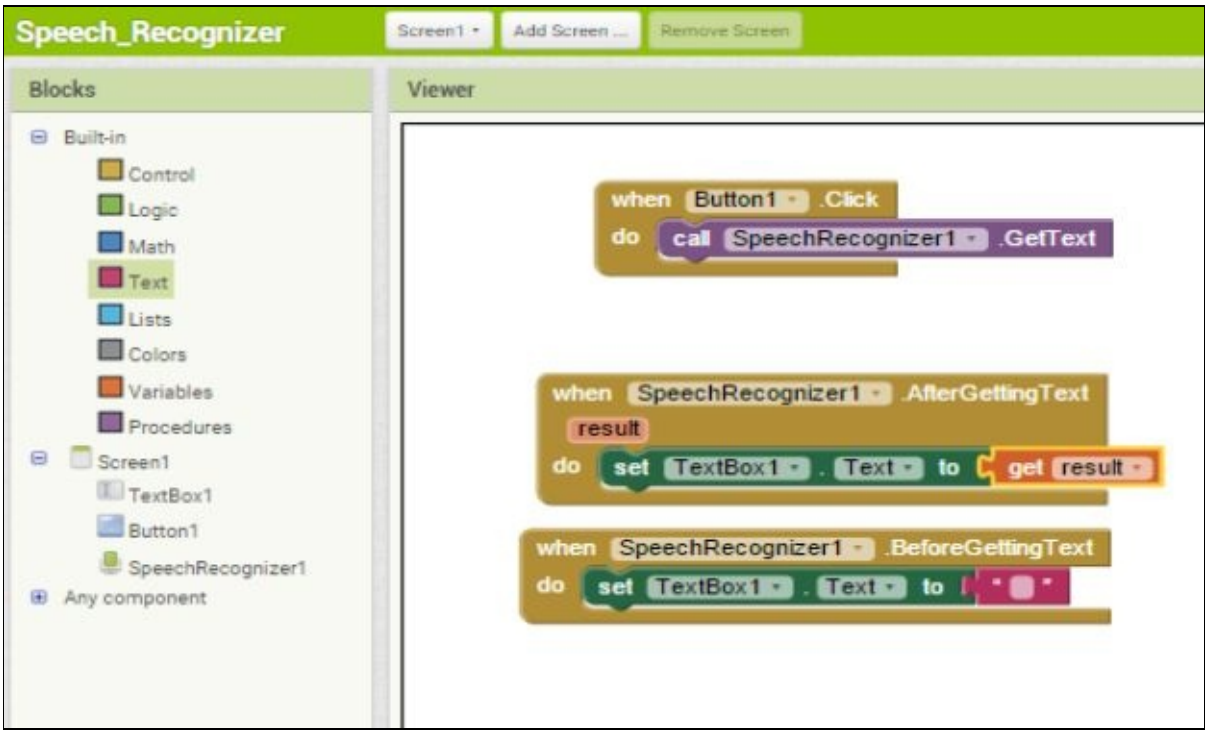
11. Now mouse over on result and click on 

block. This block contains the Text which you spoke.

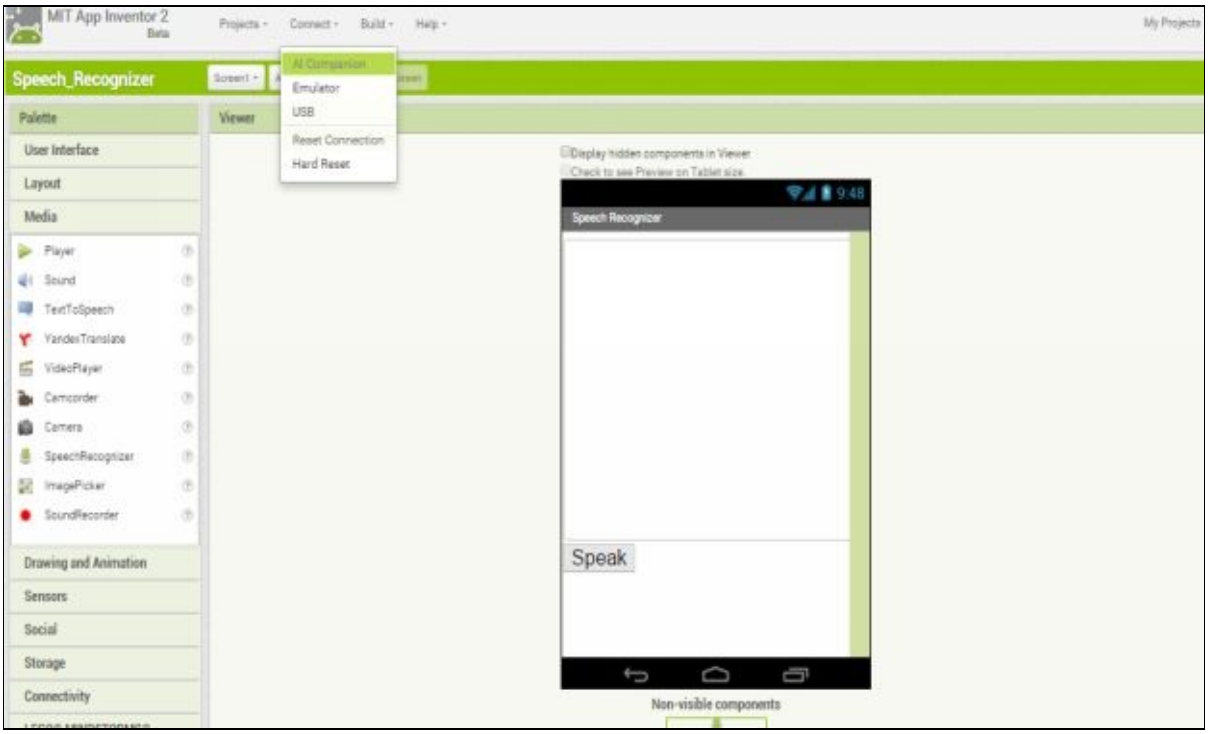




12. Attach the blocks as shown below. When you click Speak Button your app will open microphone and once you have completed speech, app will show the text in TextBox1 that you spoke.



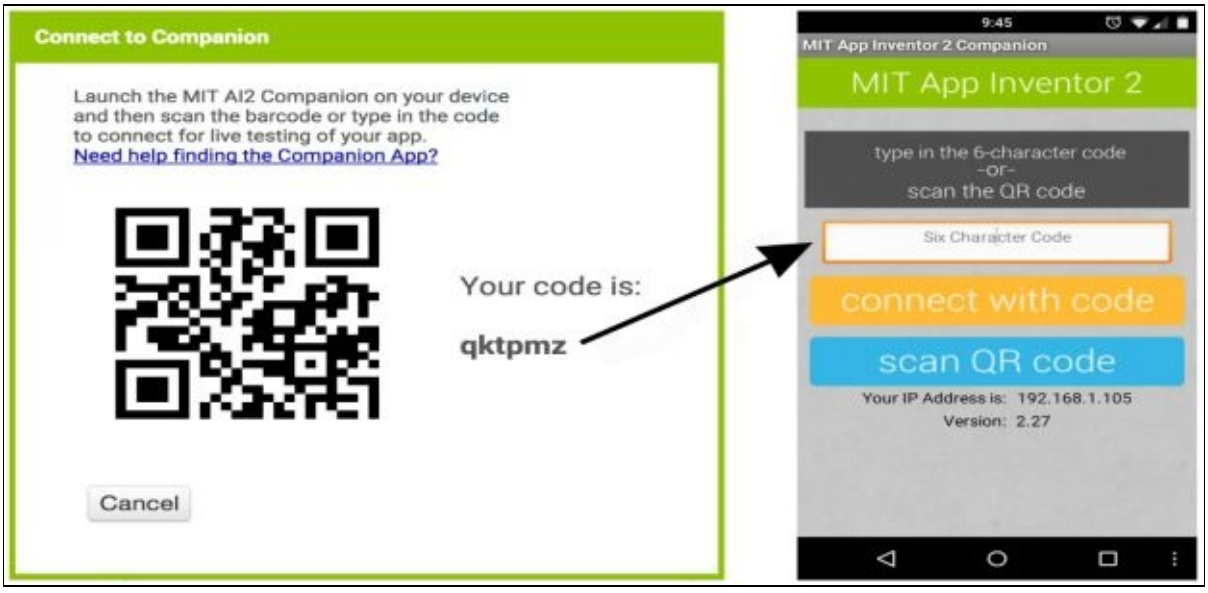
13. Now go to Designer and click on Connect and select AI Companion.



14. Now open MIT App Inventor 2 Companion in your mobile/Tablet.

And enter the code or scan the QR code from your mobile and click on connect with code.

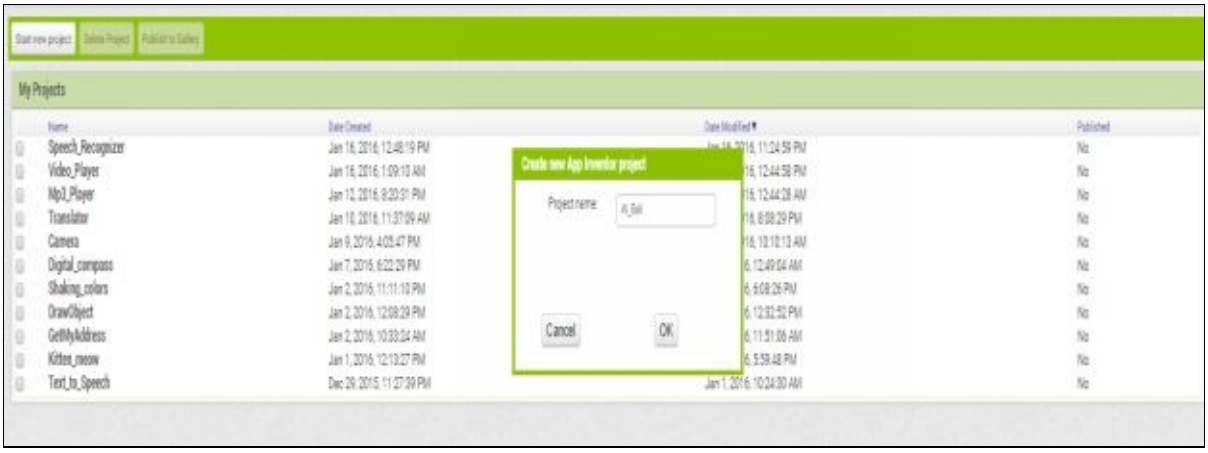
Note:-Your Mobile/Tablet should be connected to same wi-fi/network to which your computer/Laptop is connected.



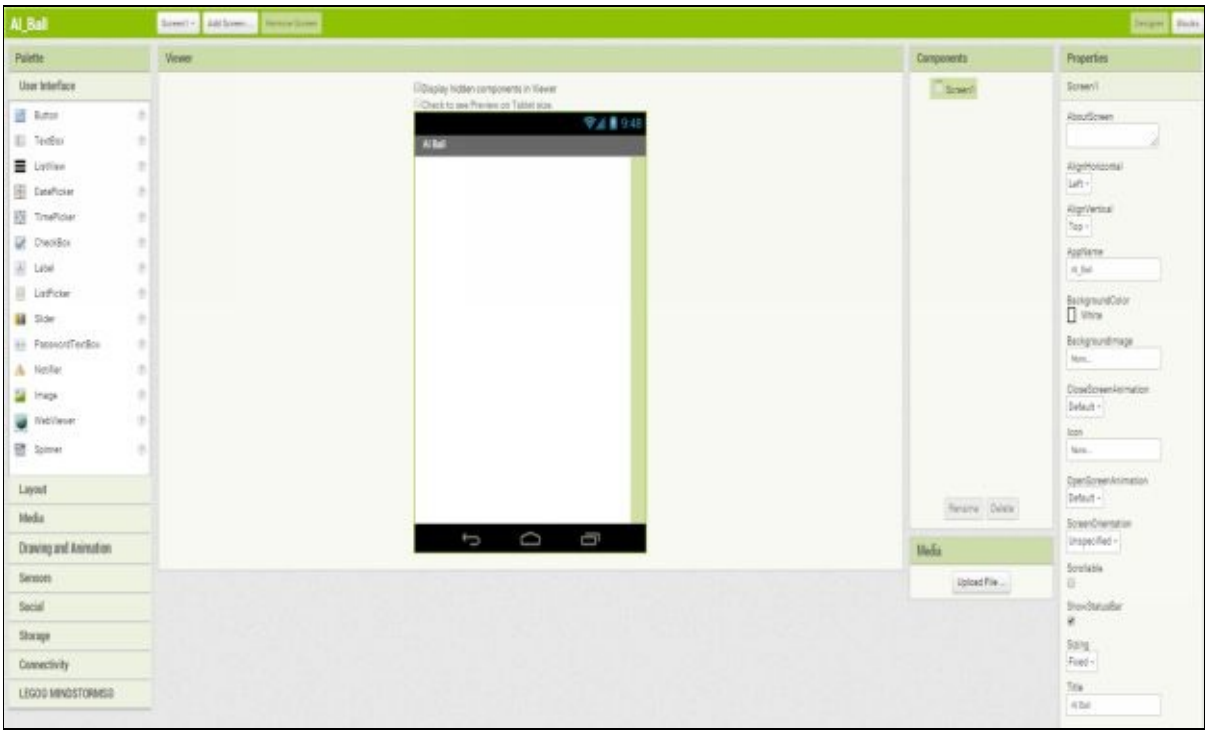
15. Now you should see your app in your phone for live testing. Click on Speak Button and Speak You should see the text in TextBox that you spoke.

Chapter 15: AI Ball App

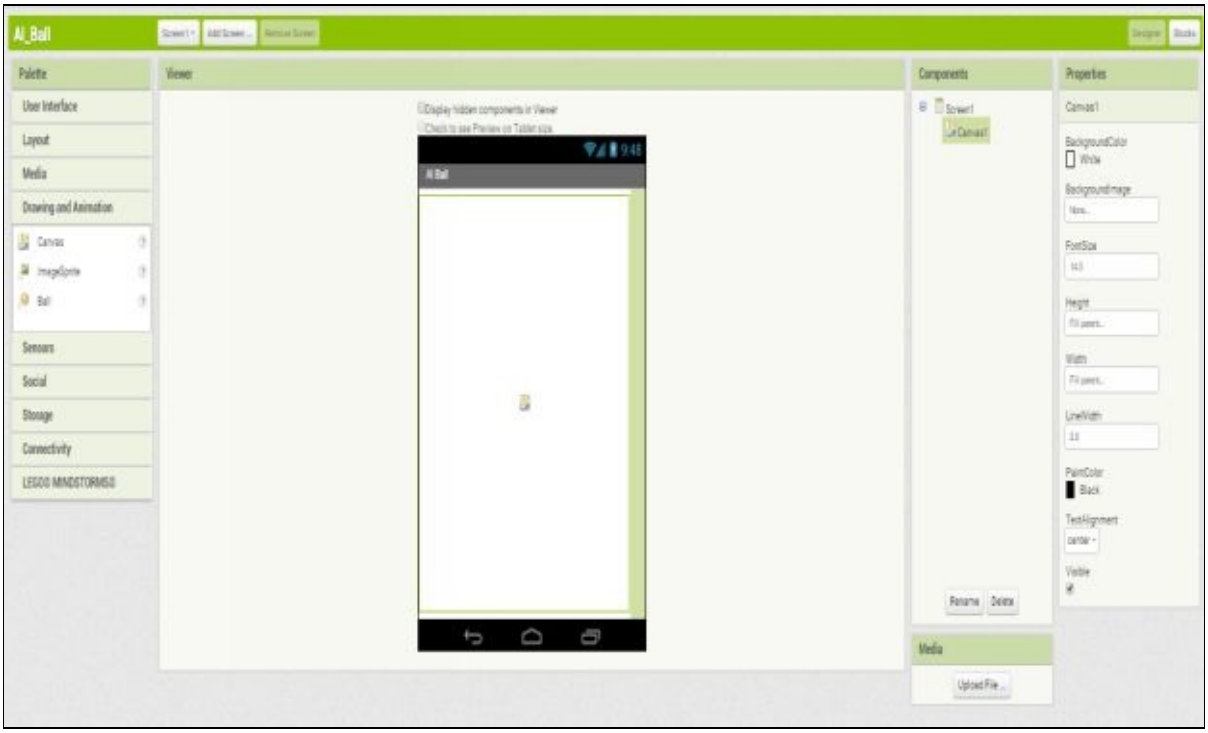
1. Click on Start new project and give it name AI_Ball and Click OK.



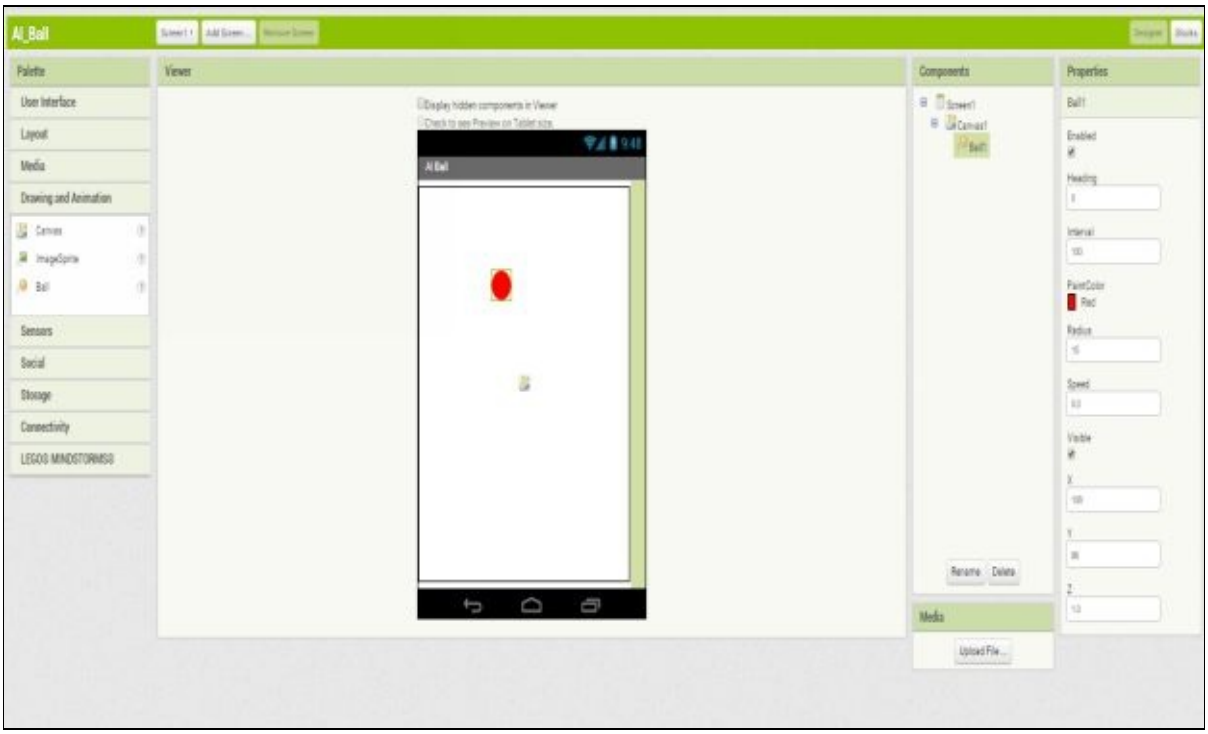
2. Change Screen1 Title to AI Ball.



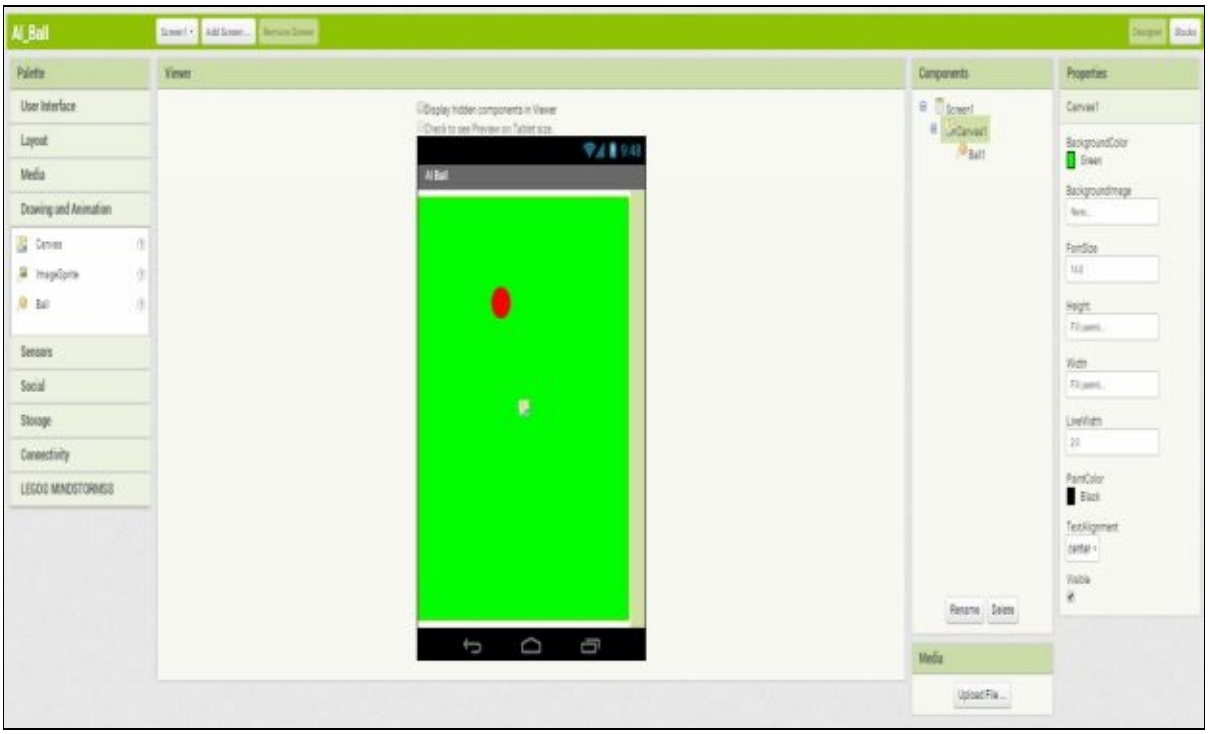
3. Drag a Canvas from Palette to Viewer screen and set its Height and Width to Fill Parent.

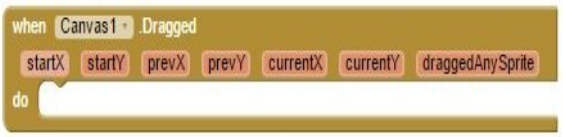


4. Drag a Ball inside the Canvas1 and set it's radius to 15. And set its PaintColor to Red.

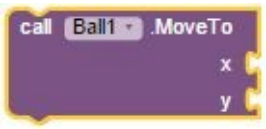


5. Now click on Canvas1 and change it’s BackgroundColor to Green.

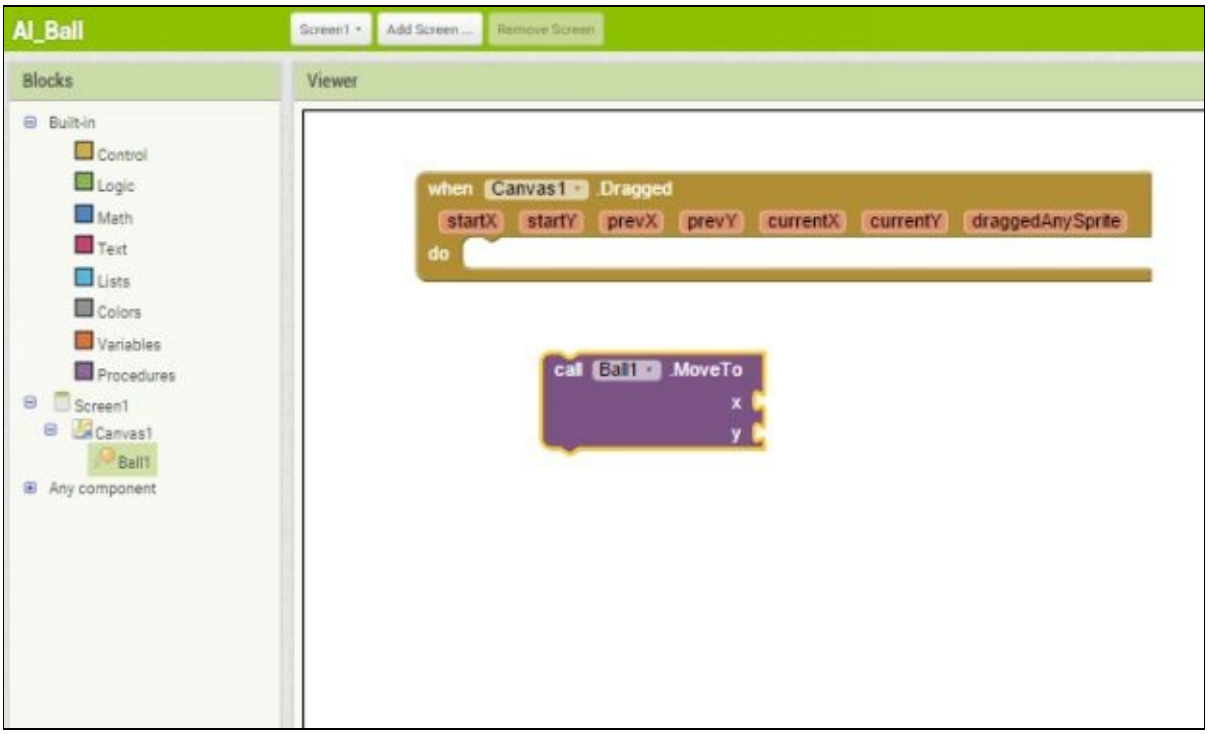




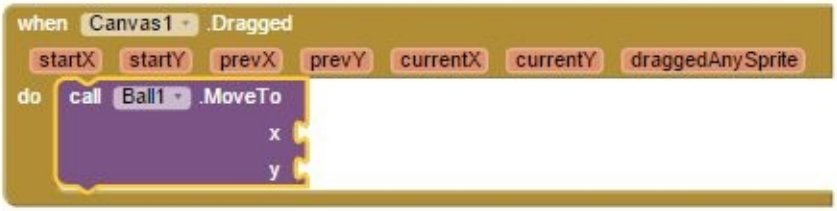
6. Now go to Blocks and click on Canvas1 and select block. This block will identify dragging activity on your canvas and decide what to do when dragged.

7. Now click on Ball1 and scroll down to select 

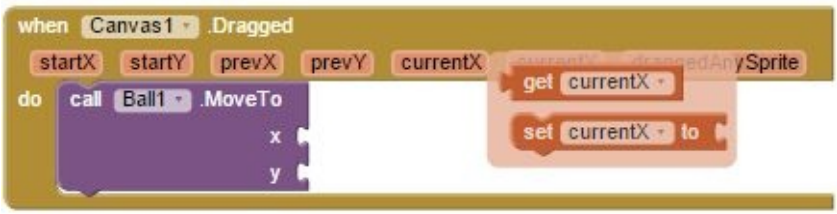
block. Every point on Screen is represented by X and Y co-ordinates. This block will move the ball to some point on screen represented by X and Y co-ordinates.

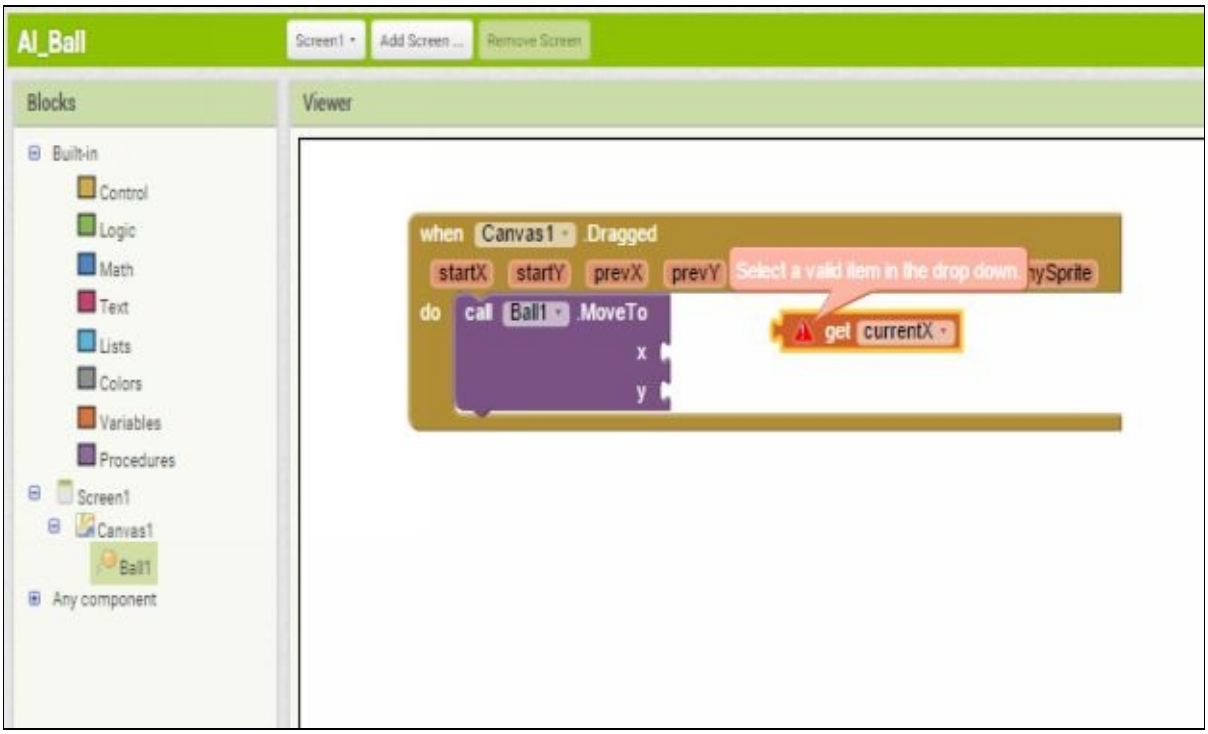


8. Attach the blocks as shown below. So that when you drag on the canvas your app will move the ball with your touch impressions on the canvas.

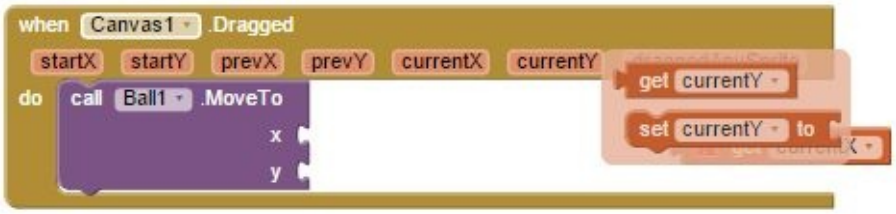


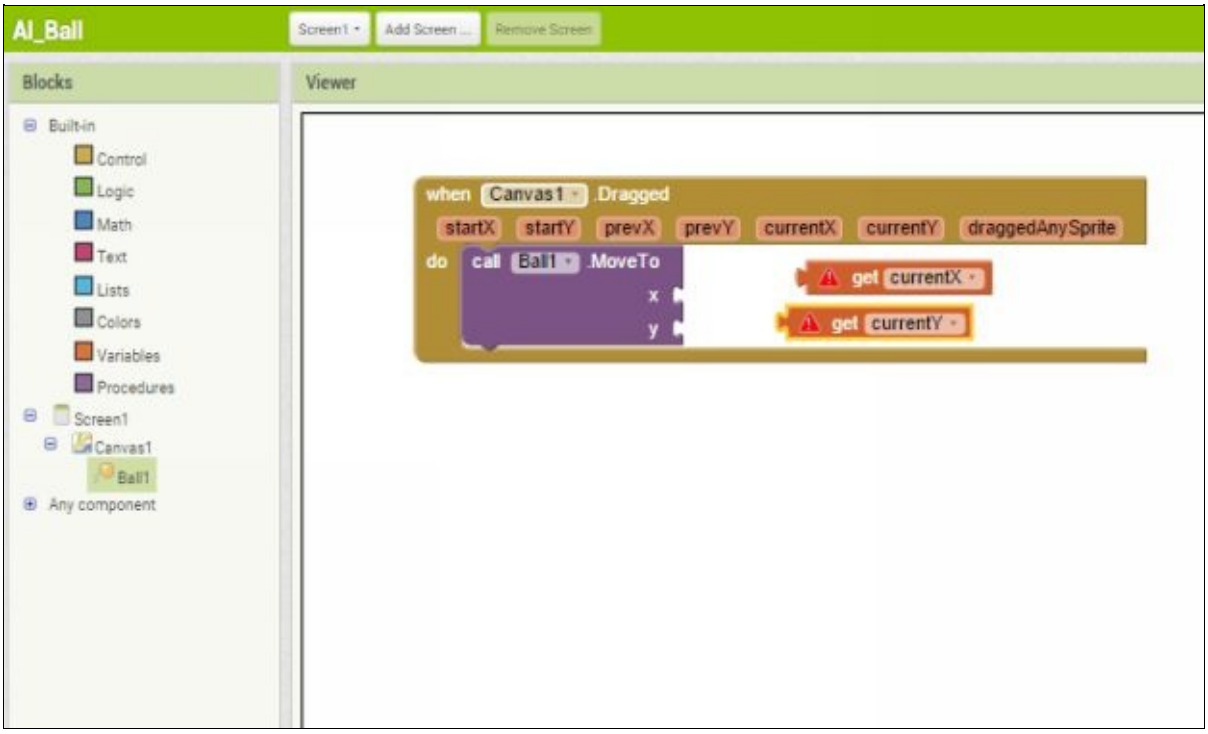
9. Now Mouse over on currentX to get  block. Here currentX and currentY represents X and Y coordinates of the Touchpoint on canvas.



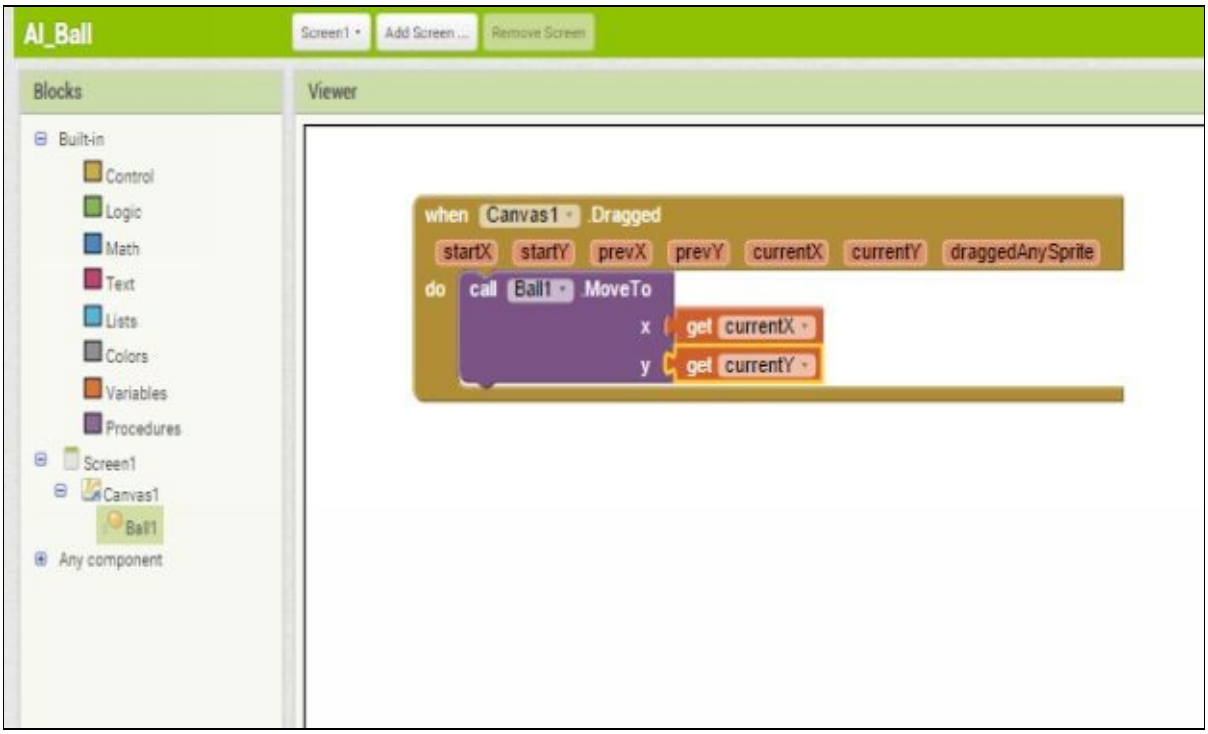


10. Similarly Mouse over on currentY and get  block.

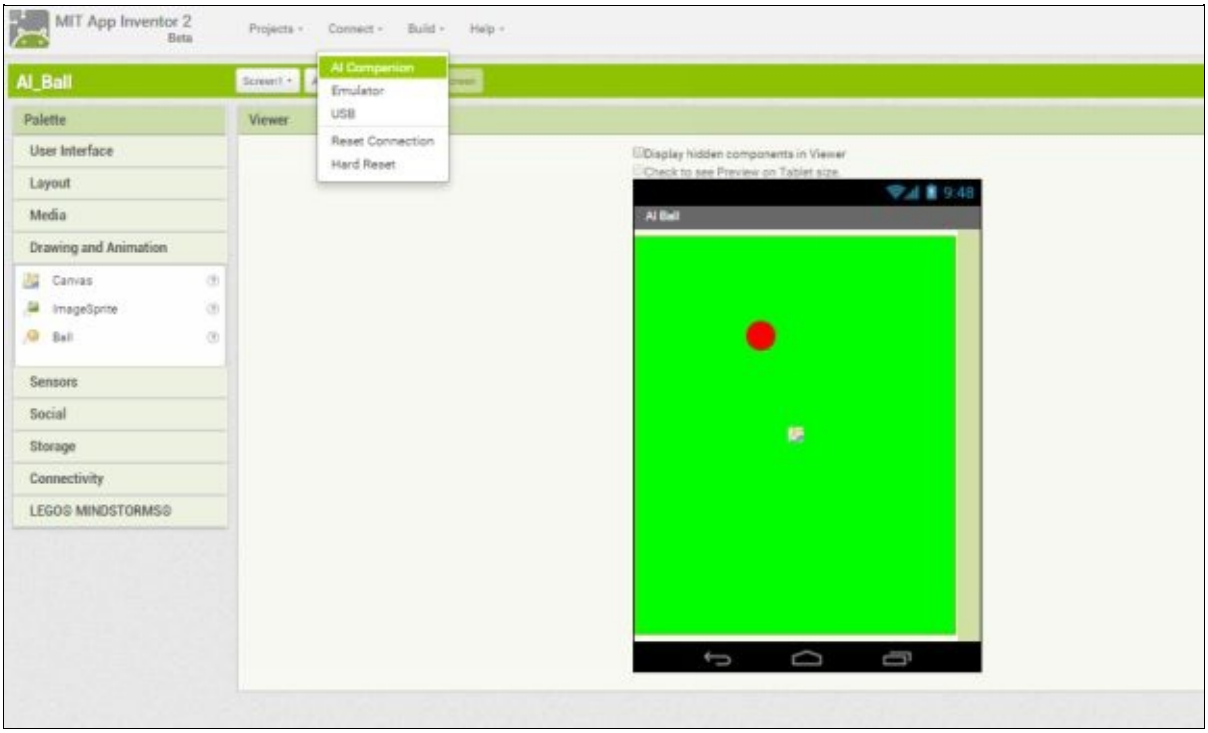


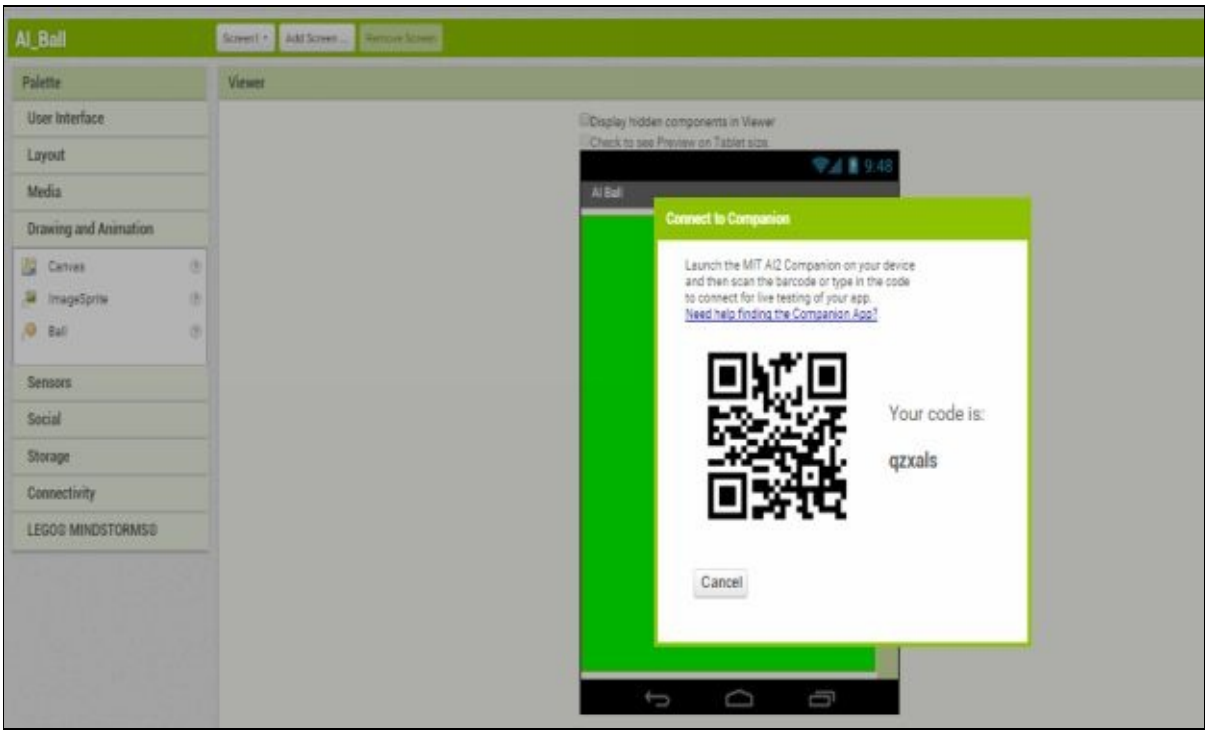


11. Now attach the blocks as shown below. So that your app will move the ball to the point you touch on the screen and move it with your finger on app screen.



12. Now go to Designer and click on Connect then click on AI Companion.

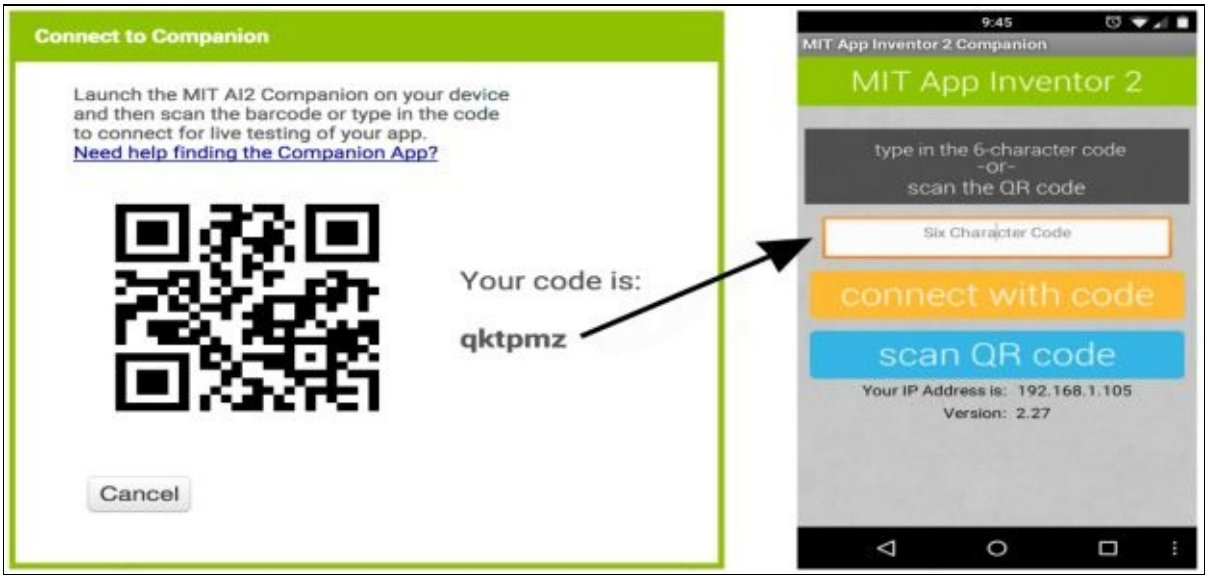




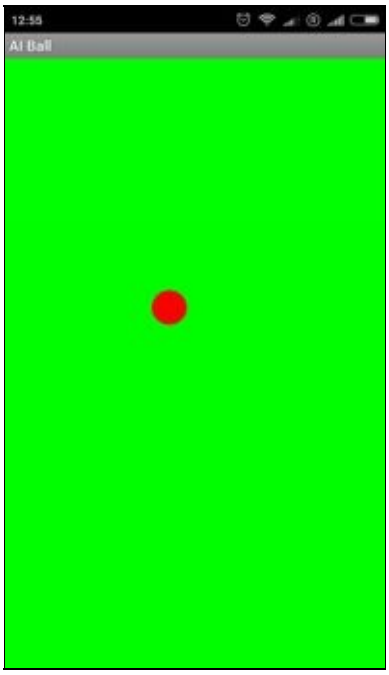
13. Now open MIT App Inventor 2 Companion in your mobile/Tablet.

And enter the code or scan the QR code from your mobile and click on connect with code.

Note:-Your Mobile/Tablet should be connected to same wi-fi/network to which your computer/Laptop is connected.

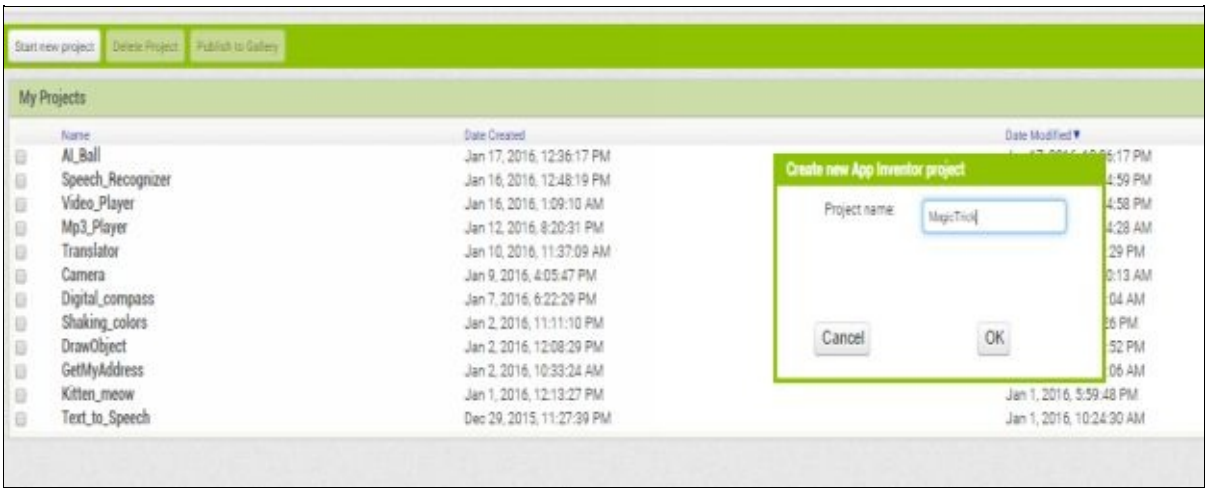


14. Now you should see your app in your phone for live testing. Touch anywhere on the screen and drag it to move the Ball.

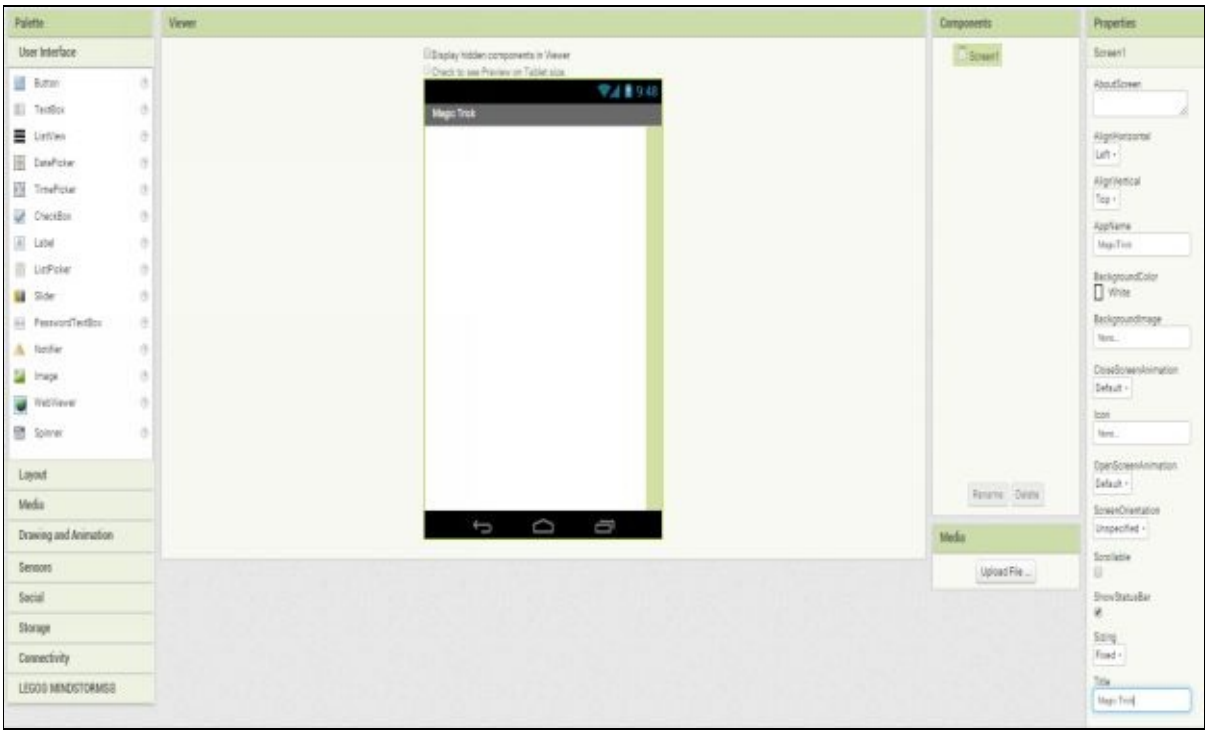


Chapter 16: Magic Trick App

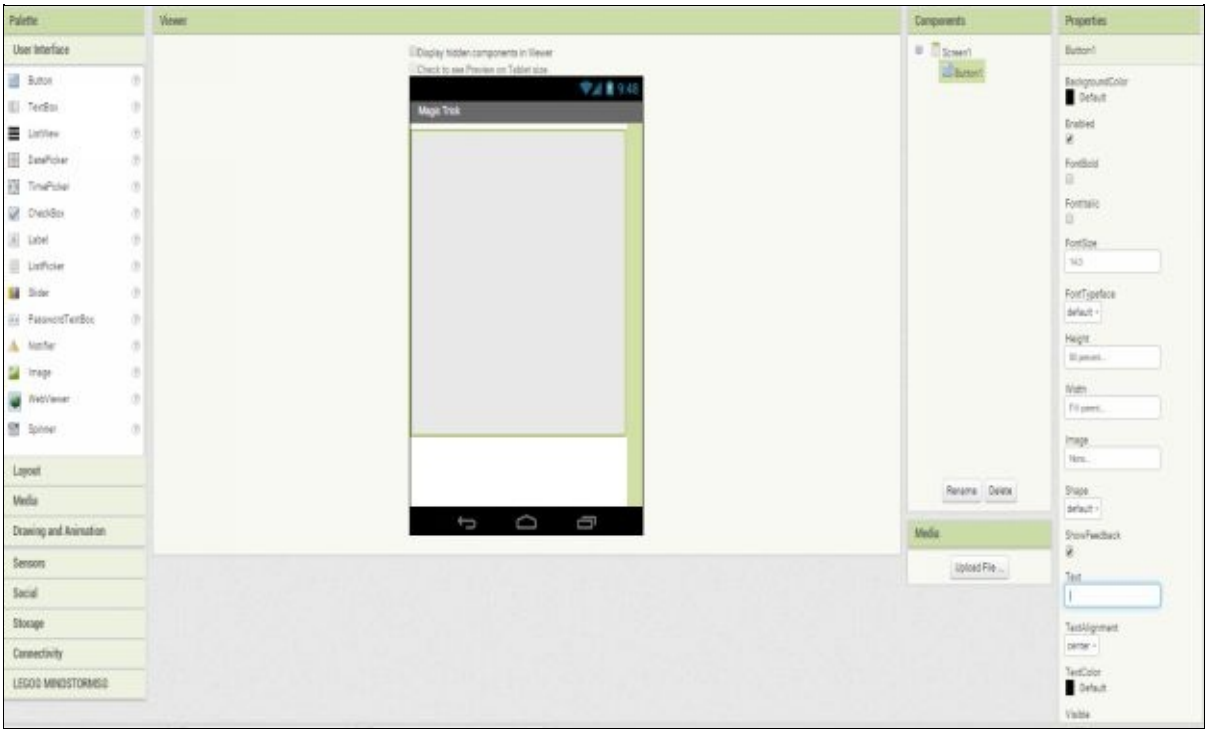
1. Click on Start new project and give it name MagicTrick and Click OK.



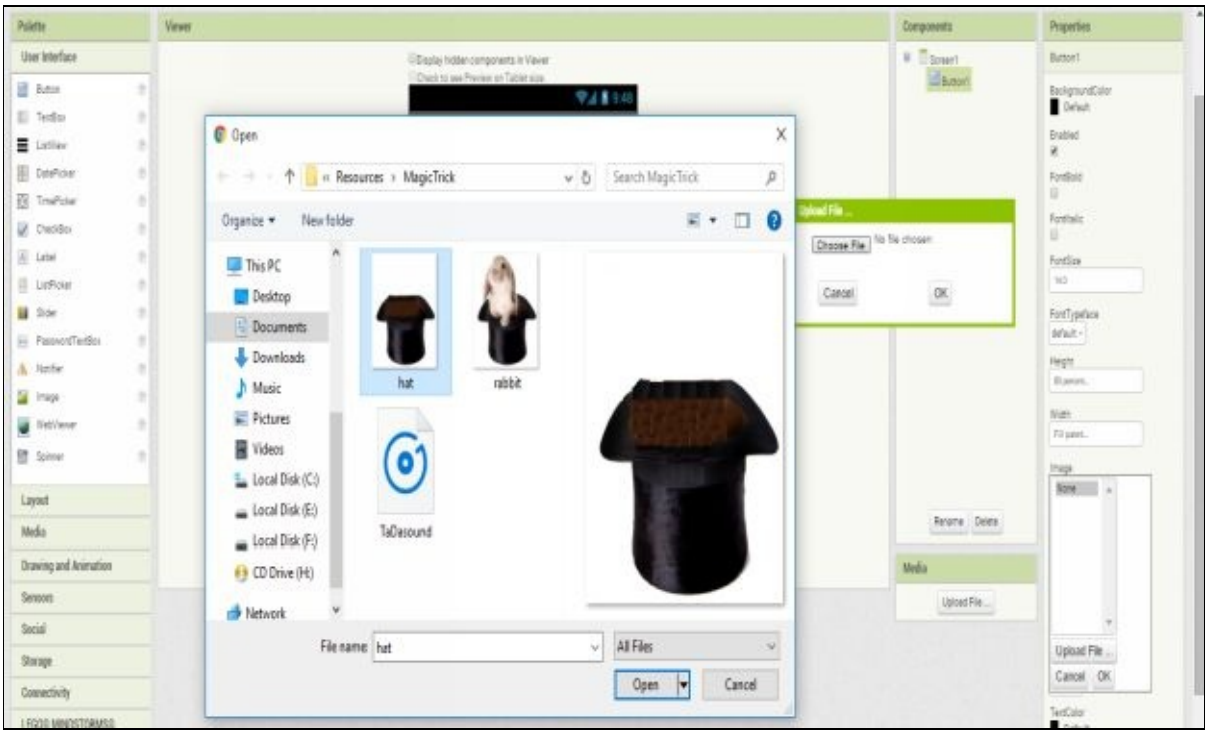
2. Change Screen1 Title to Magic Trick.

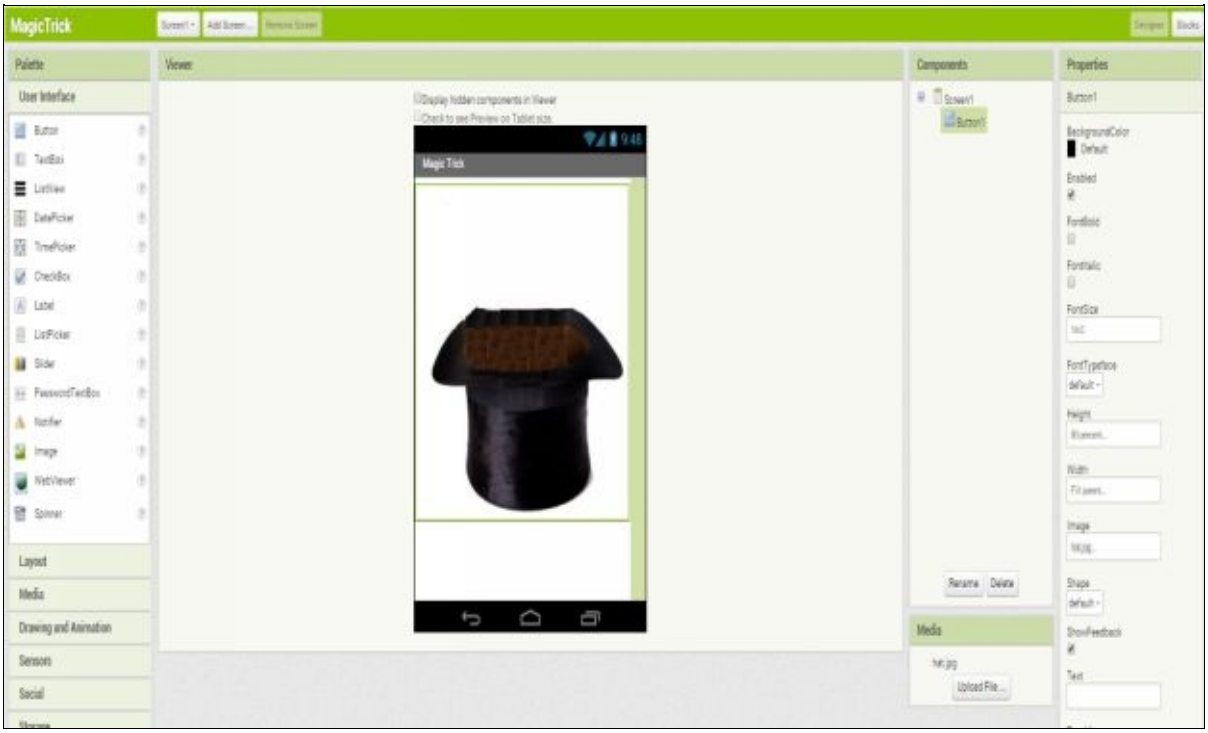


3. Now Drag a Button from Palette to Viewer screen and set its Height to 80 percent and Width to fill parent and Delete it's Text.

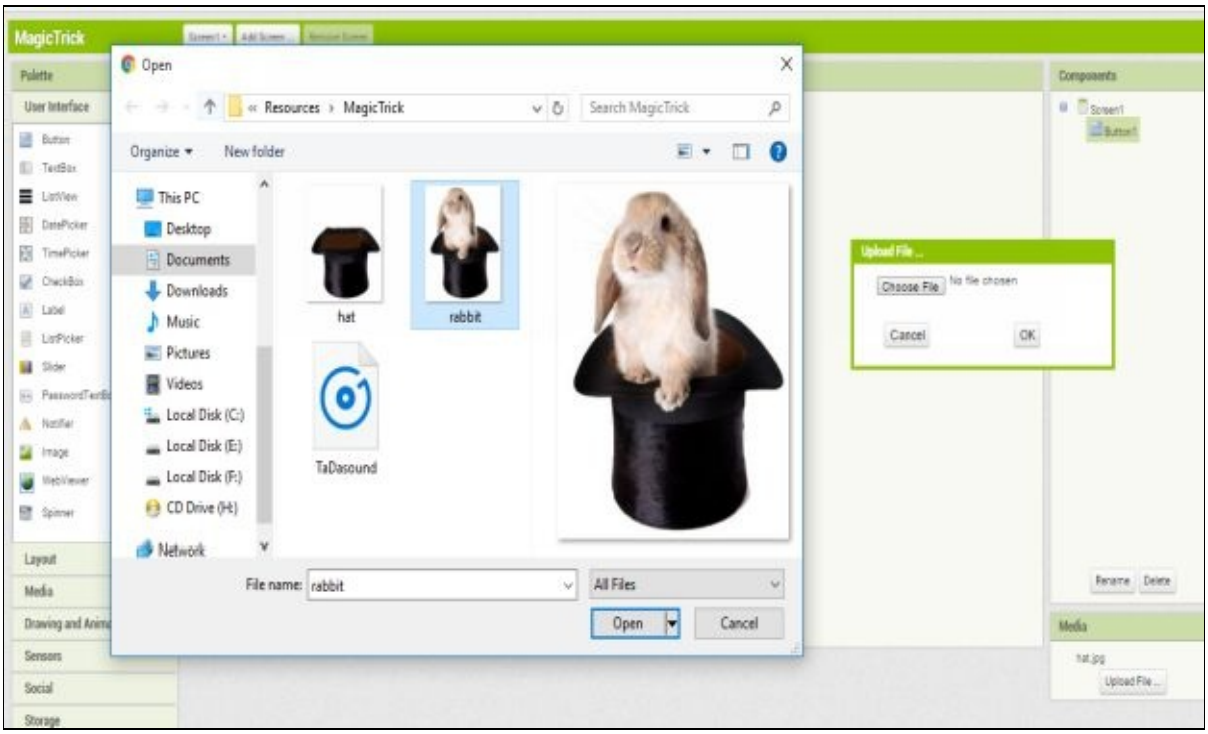


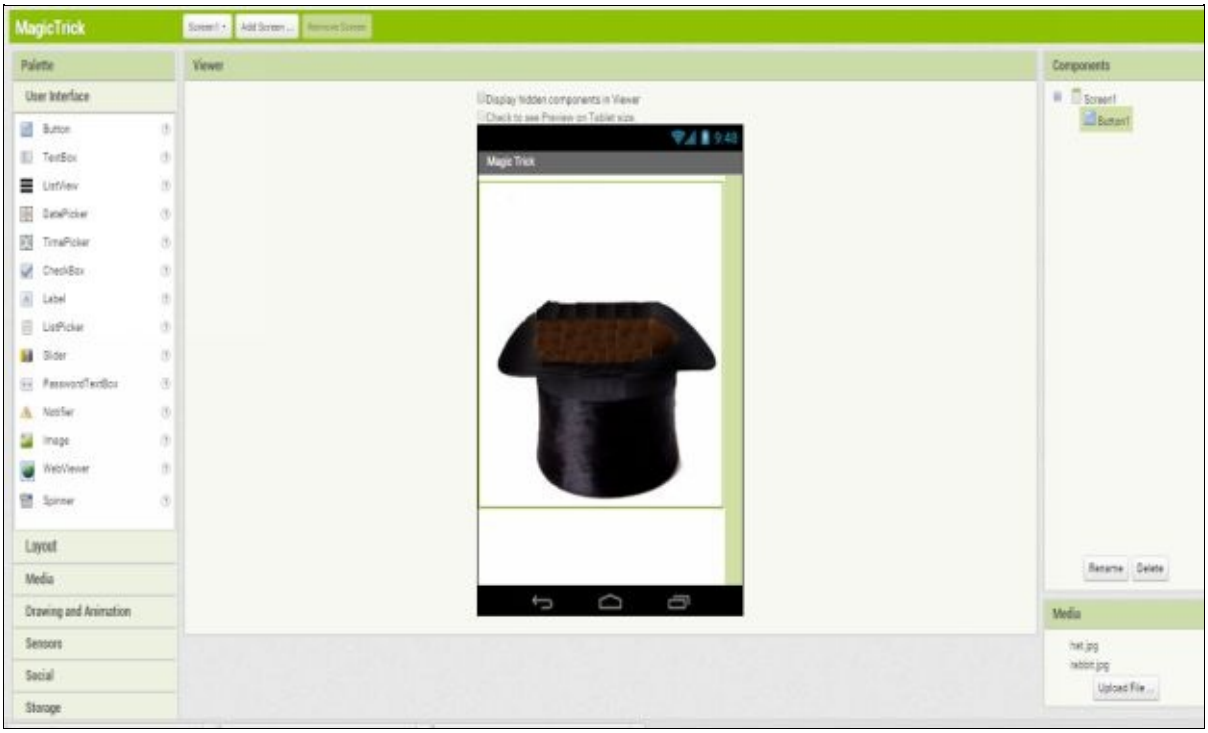
4. Now Change Button1 Image to hat.jpg and click OK.



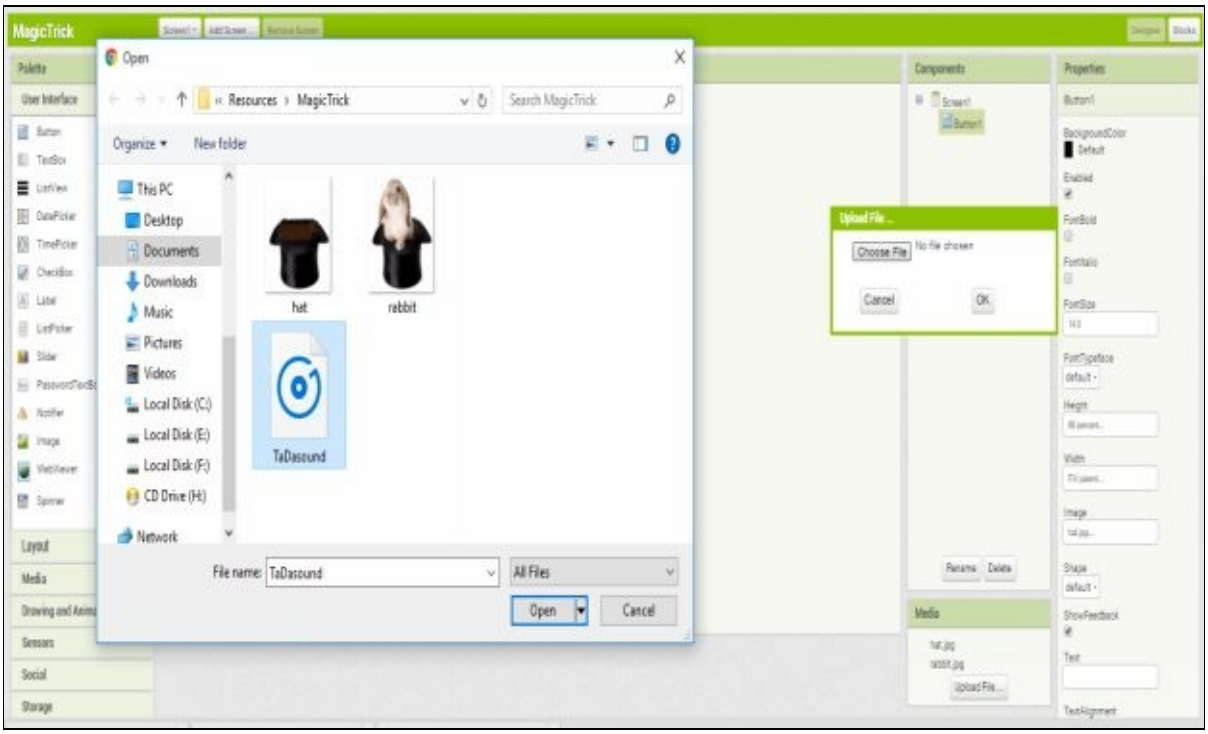


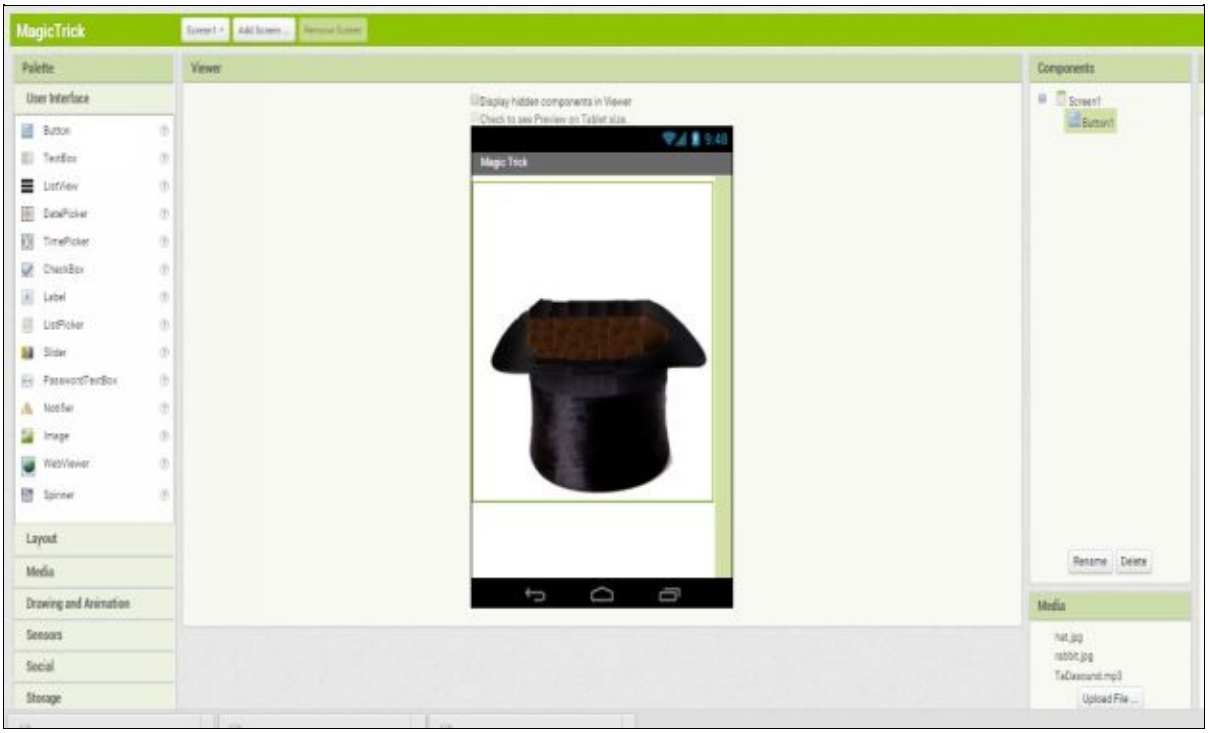
5. Click on Upload File button and upload rabbit.jpg image.



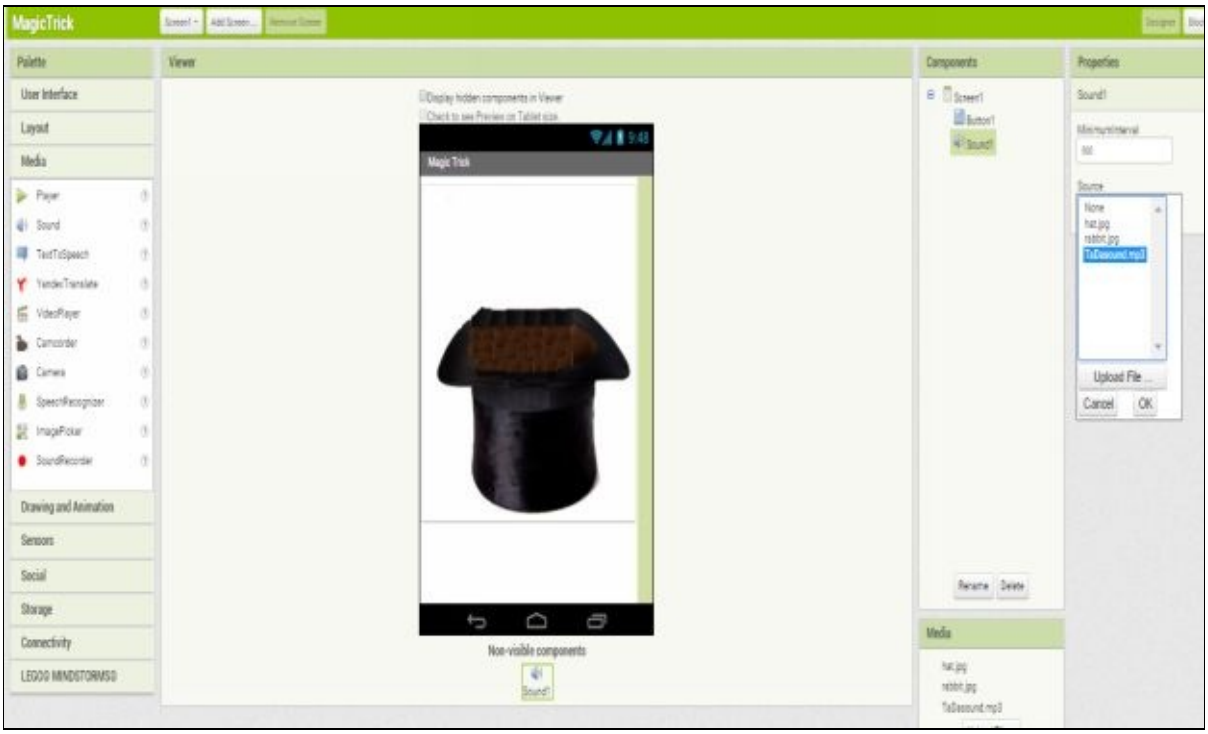


6. Click on Upload File button and upload TaDasound.mp3.

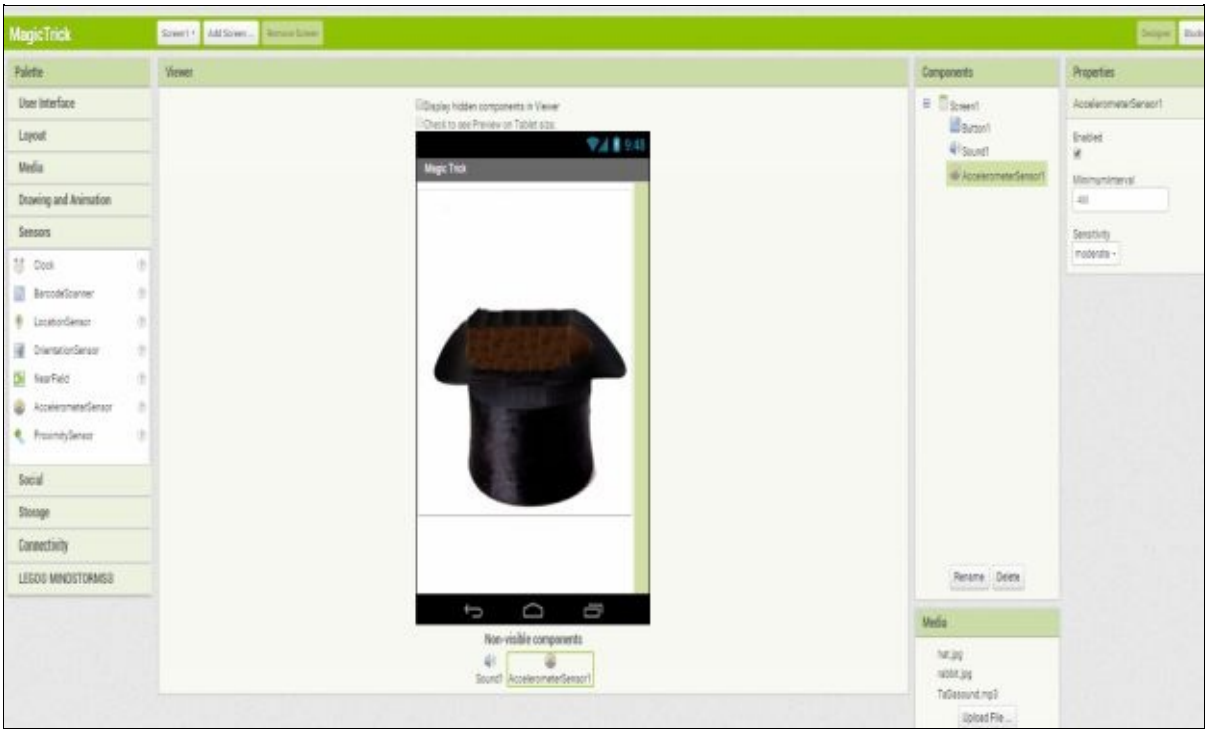




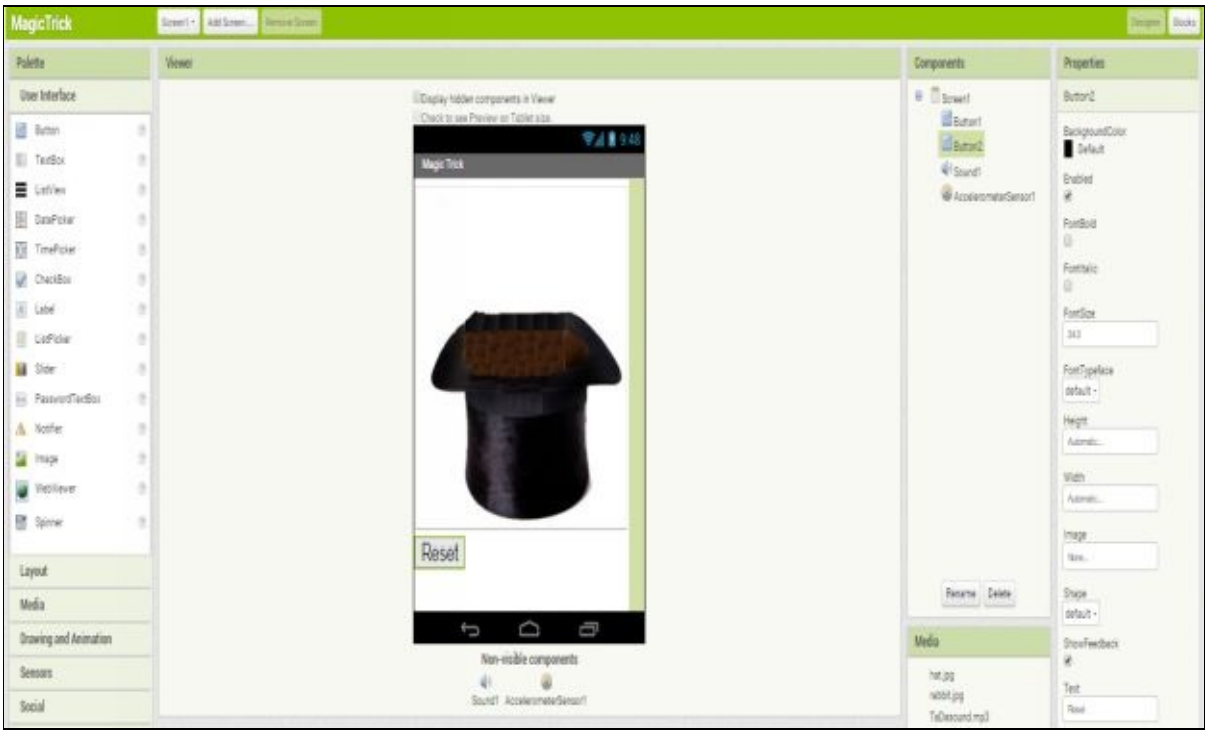
7. Now drag a Sound component from Palette to Viewer screen and select it's Source to TaDasound.mp3 and click OK.

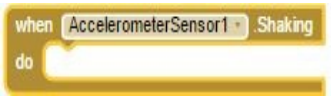


8. Now drag an AccelerometerSensor from Palette to Viewer screen.



9. Drag another Button from Palette to Viewer and change it's Text to Reset and FontSize to 24.0.



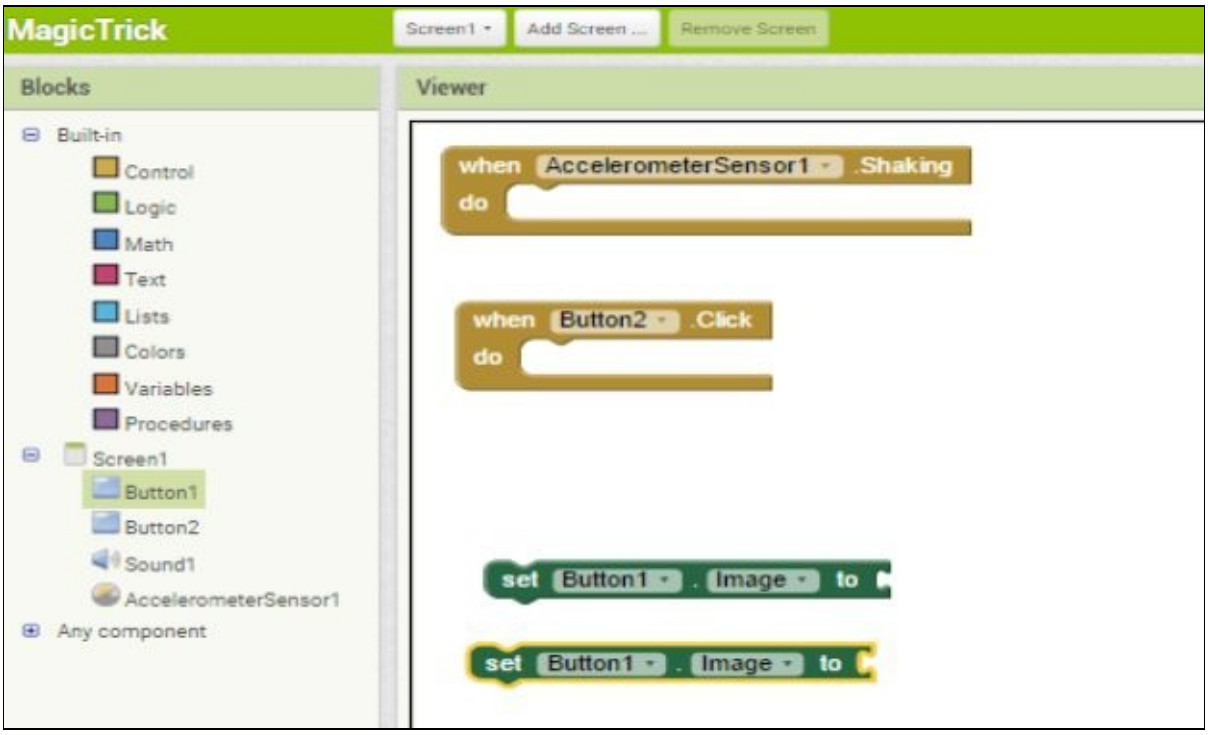
10. Now go to Blocks and click on AccelerometerSensor1 and select 

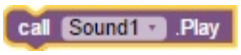
block. This block will identify shaking activity of your phone and decide what to do when your phone is shaking.

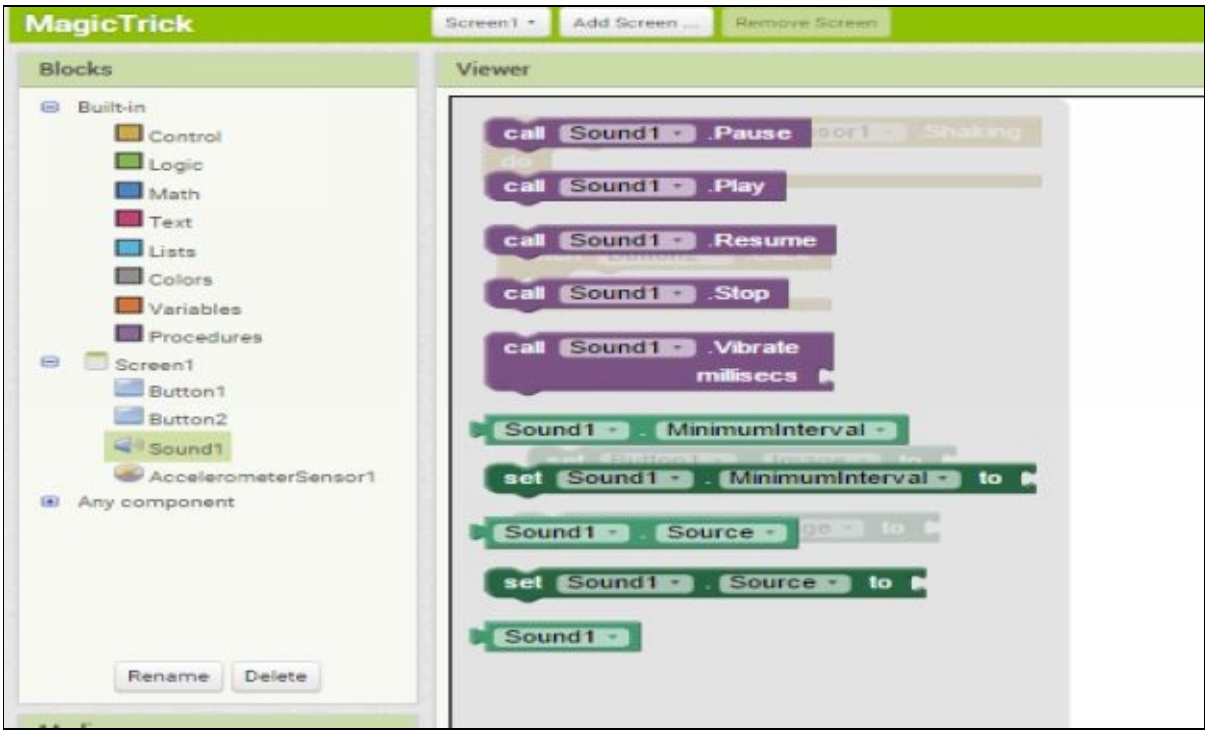
11. Now click on Button2 and select 

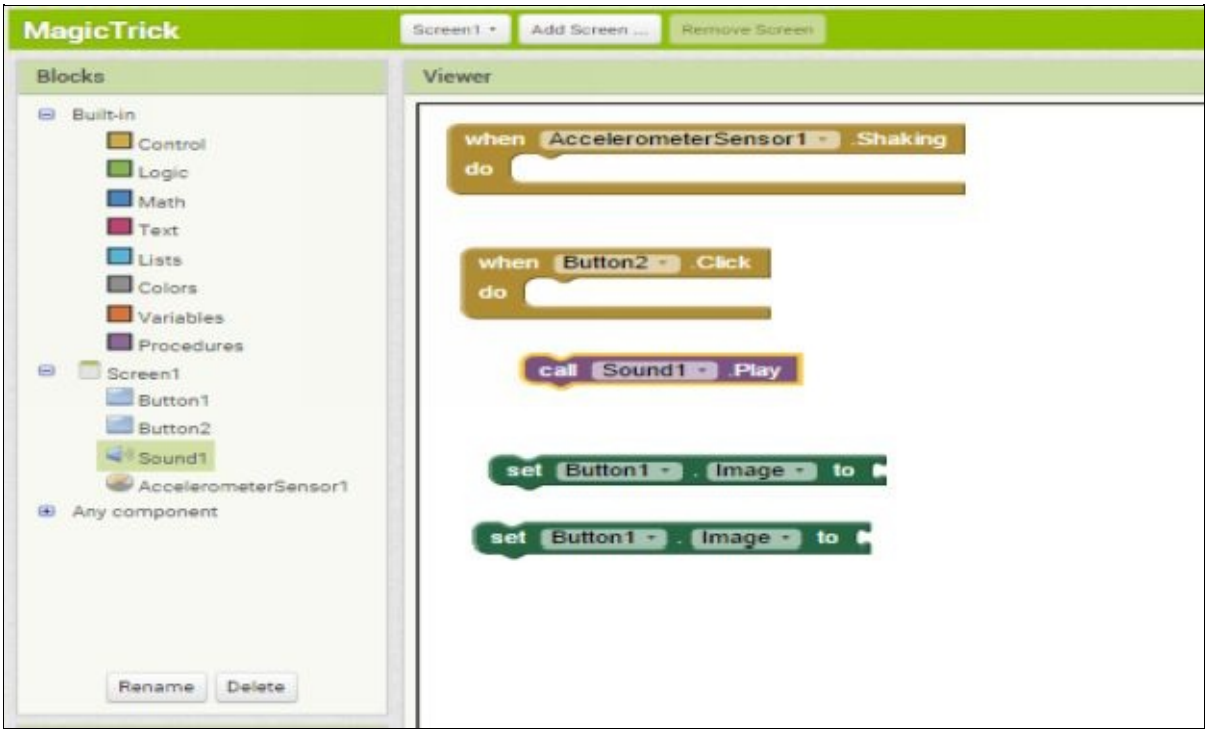
block. Here Button2 is Reset button of your app.

12. Now click on Button1 and scroll down to select  block twice.



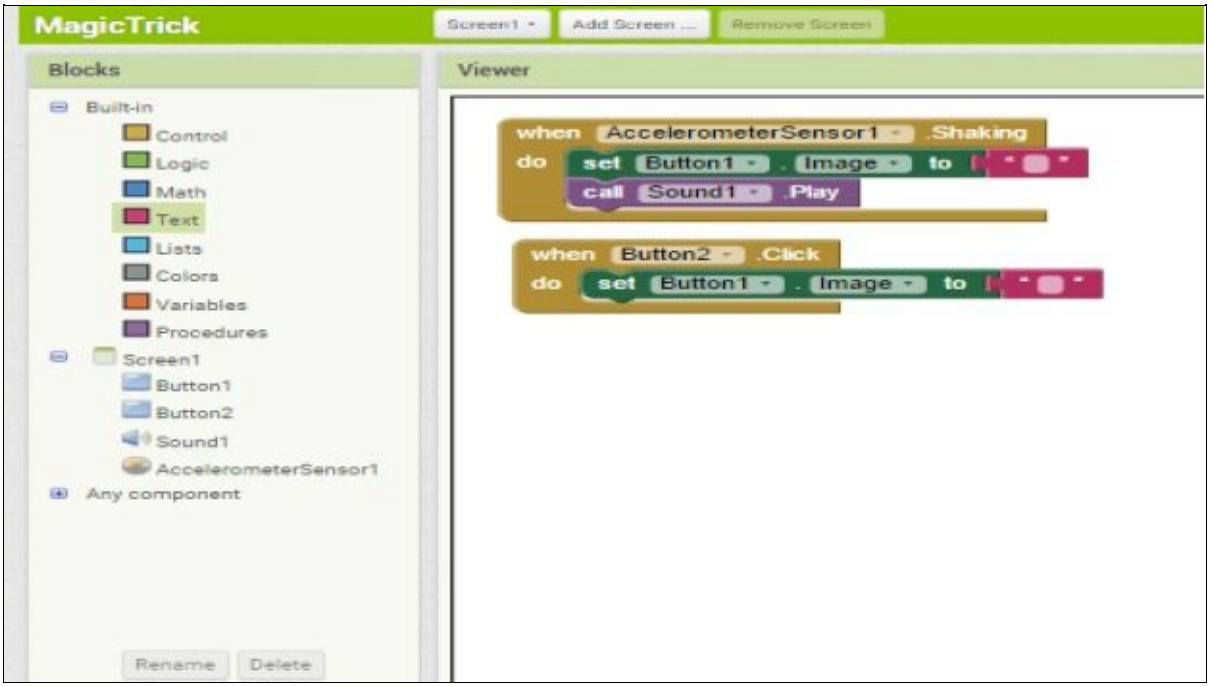
13. Now click on Sound1 and select  block. This block will play sound TaDasound.mp3 .





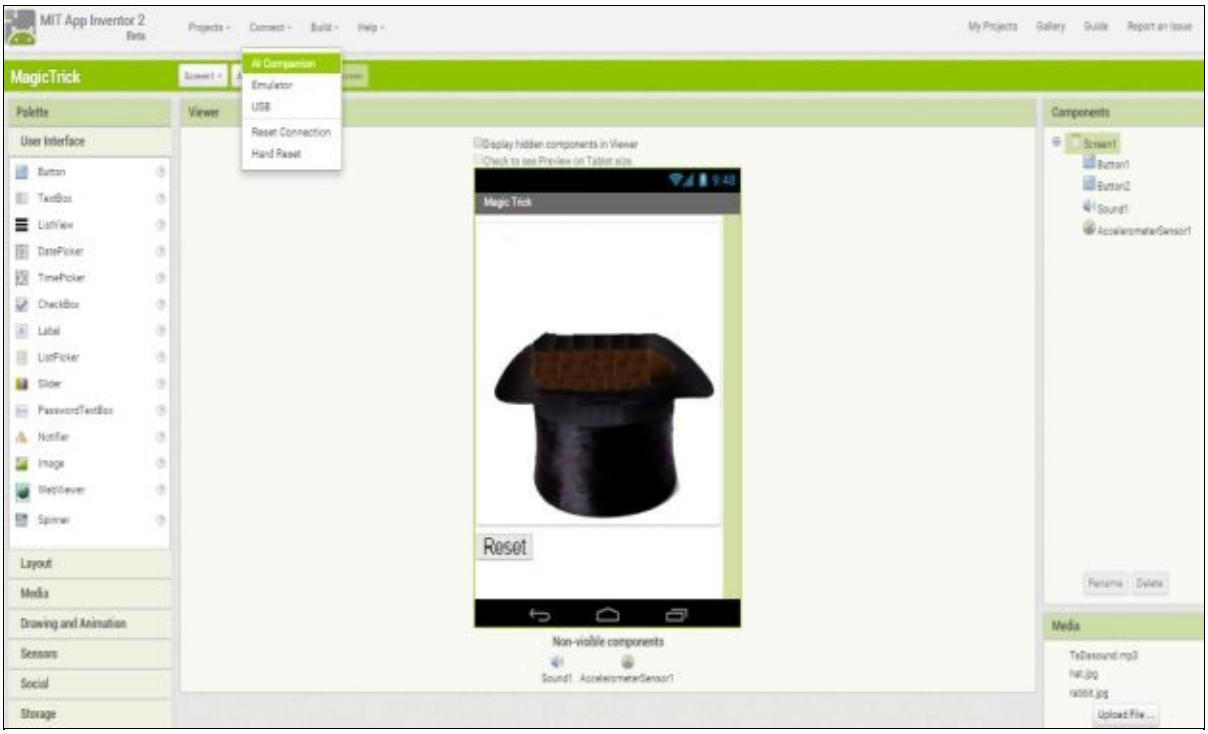
14. Attach the blocks as shown below.

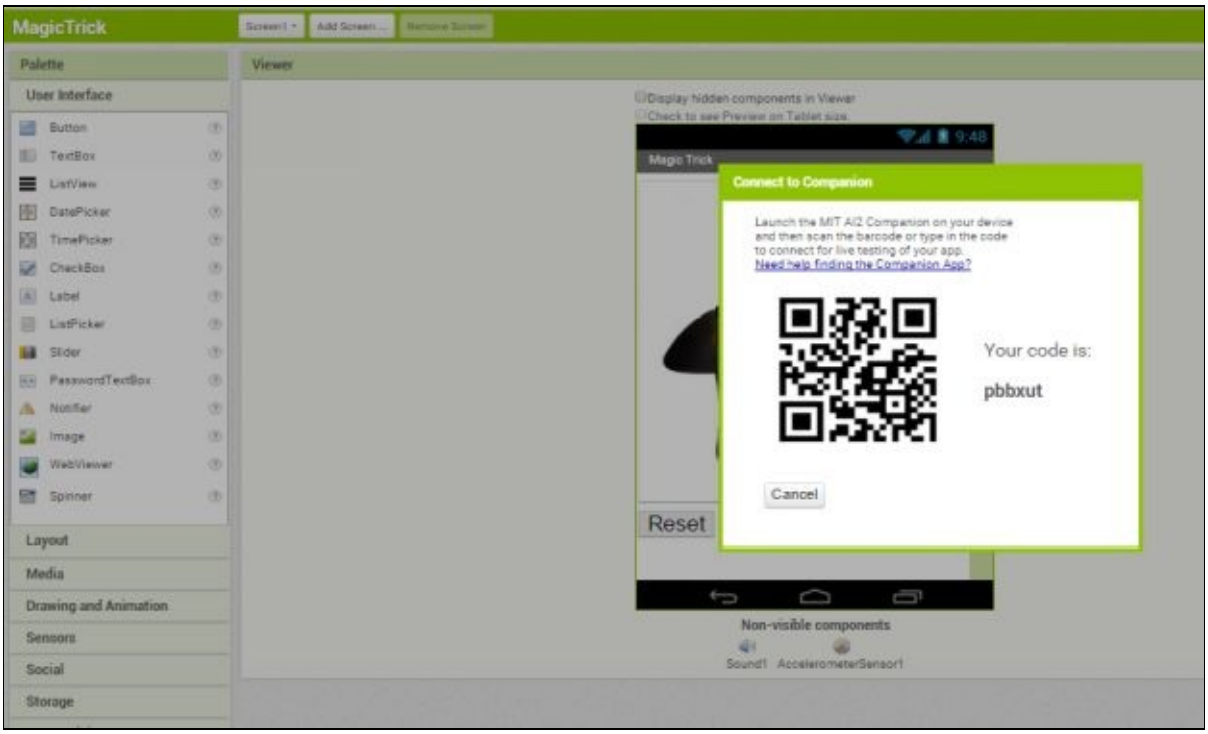
15. Click on Text and select  block twice and attach the blocks as shown below.



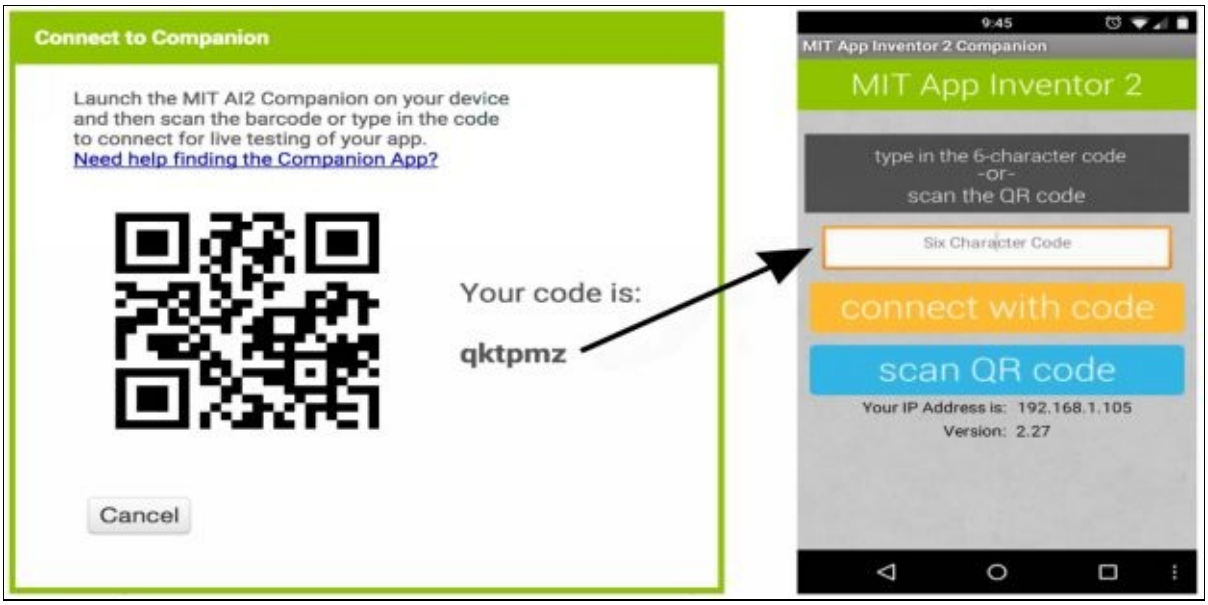
16. Now type rabbit.jpg and hat.jpg into  blocks as shown below. So when you shake your phone Button1's image will be changed to rabbit.jpg and when you click Button2 app will again change Button1's image to hat.jpg.

17. Now go to Designer and click on Connect and then select AI Companion.

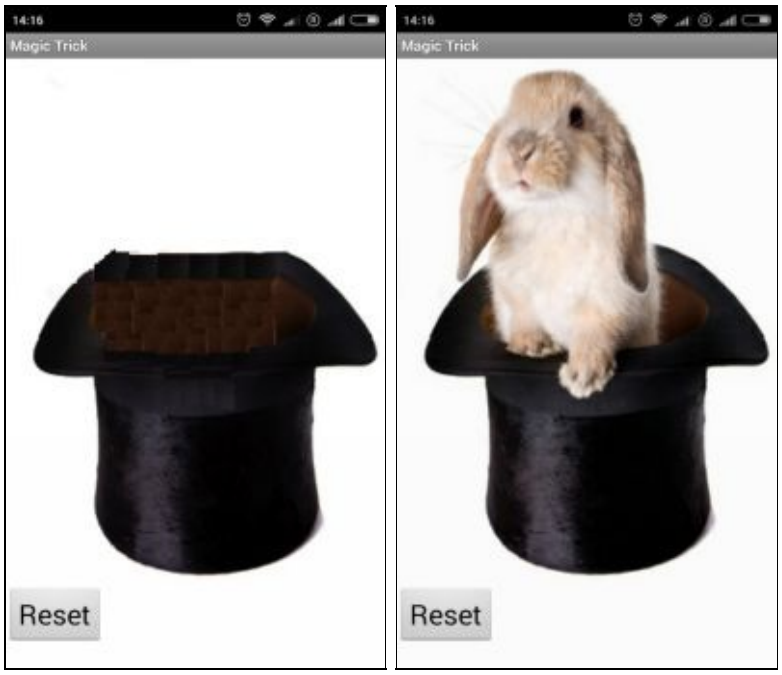




18. Now open MIT App Inventor 2 Companion in your mobile/Tablet.And enter the code or scan the QR code from your mobile and click on connect with code.Note:-Your Mobile/Tablet should be connected to same wi-fi/network to which your computer/Laptop is connected.



19. Now you should see your app in your phone for live testing. Shake your phone and see the magic and click on reset button to reset the magic hat.

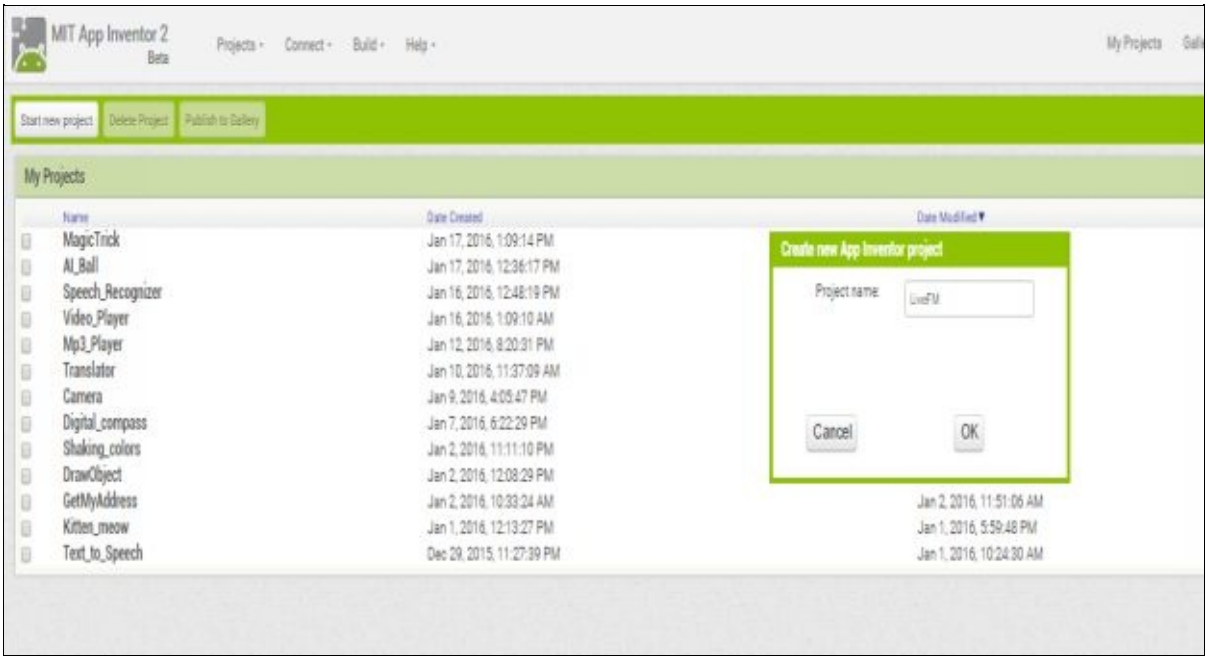


Reference

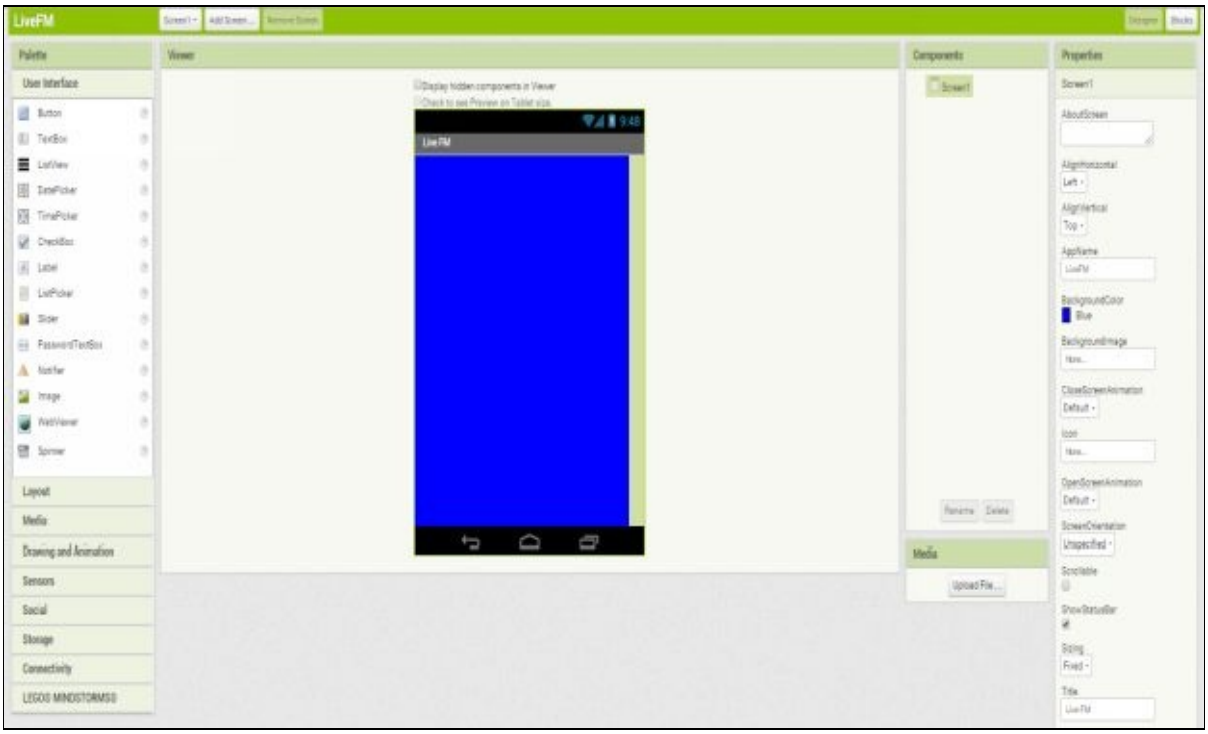
- http://appinventor.mit.edu

Chapter 17 Live FM App

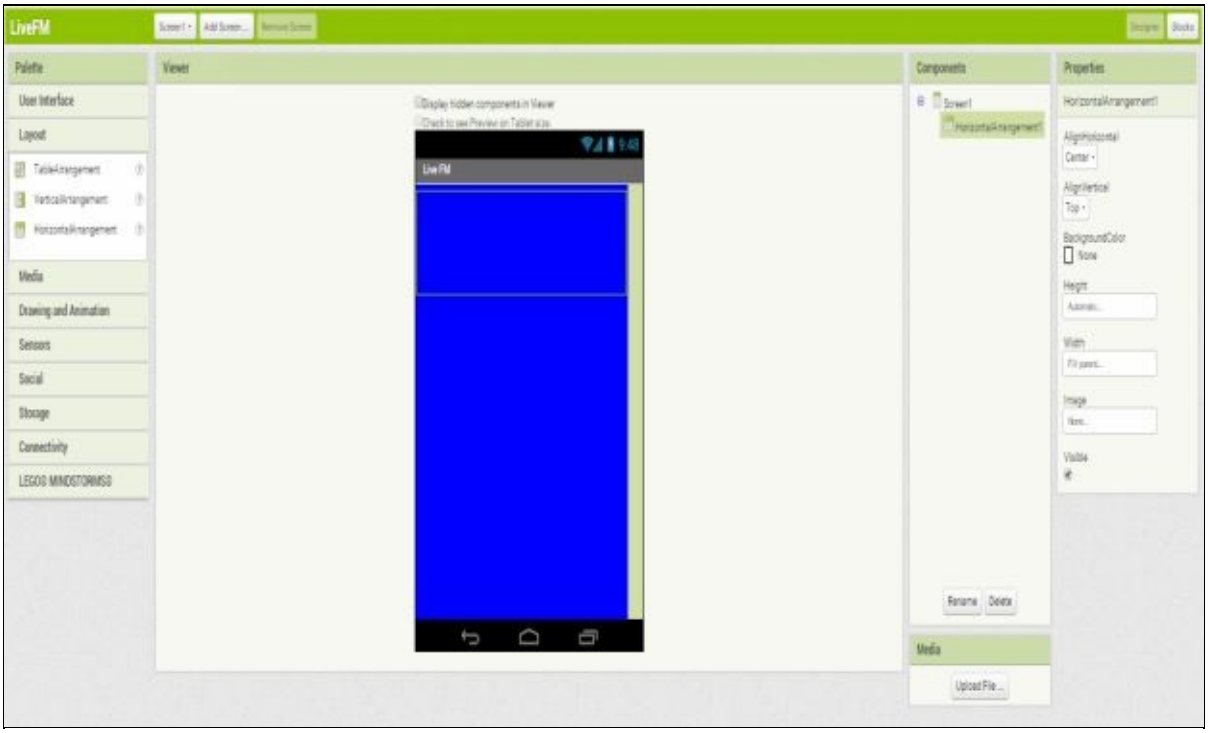
1. Click on Start new project and give it name LiveFM and Click OK.



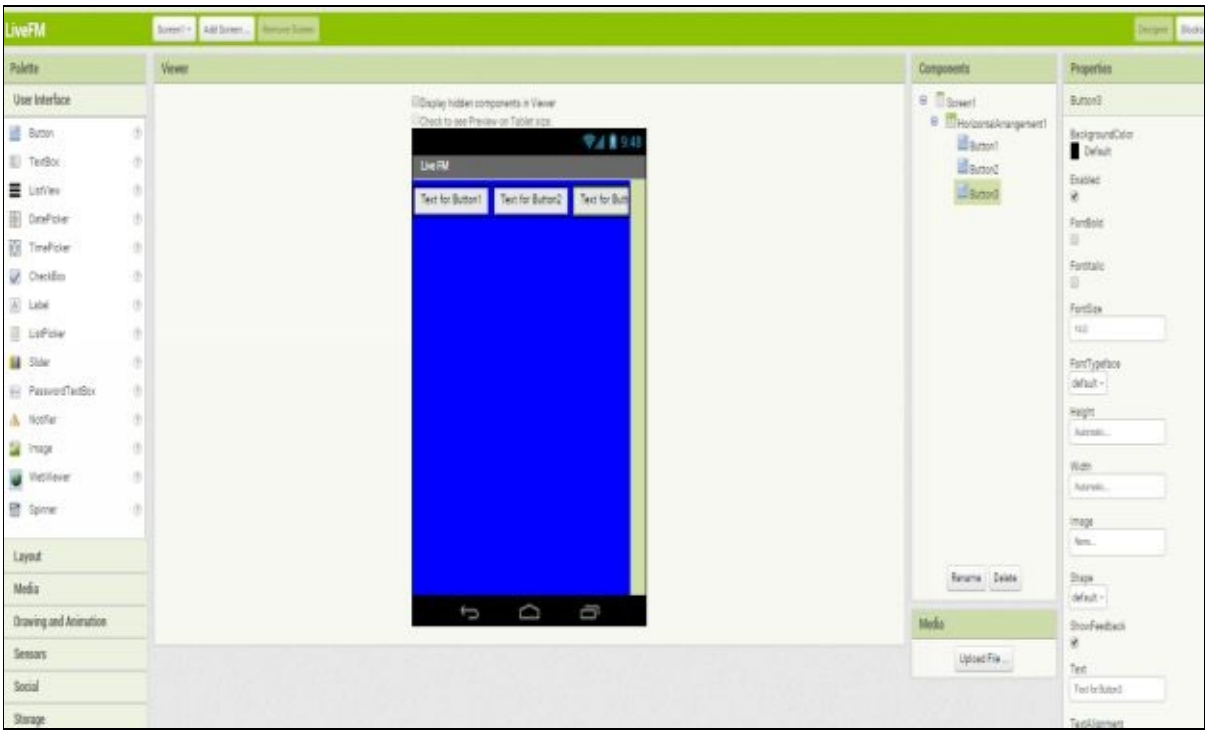
2. Change Screen1 Title to Live FM and BackgroundColor to Blue.



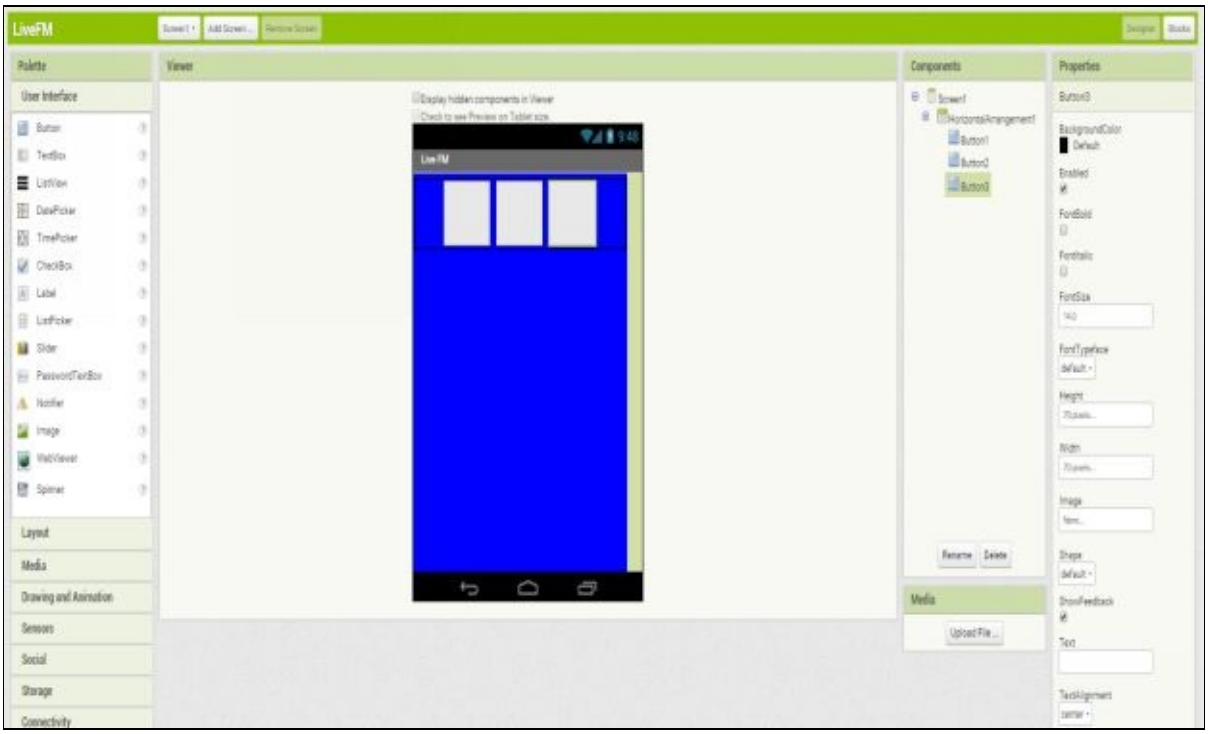
3. Drag a HorizontalArrangement component from Palette to Viewer screen and set its Width to Fill parent and BackgroundColor to None and Set AlignHorizontal to Center.



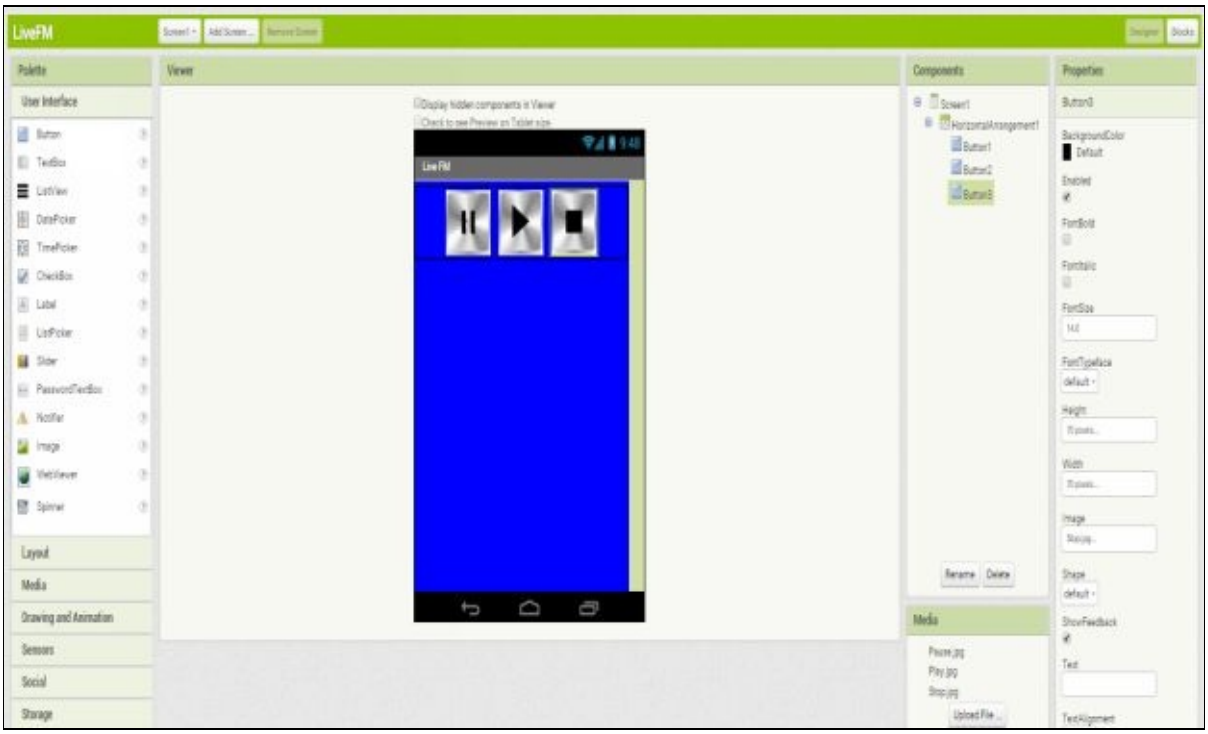
4. Drag 3 Buttons from Palette to Viewer screen, inside HorizontalArrangement.



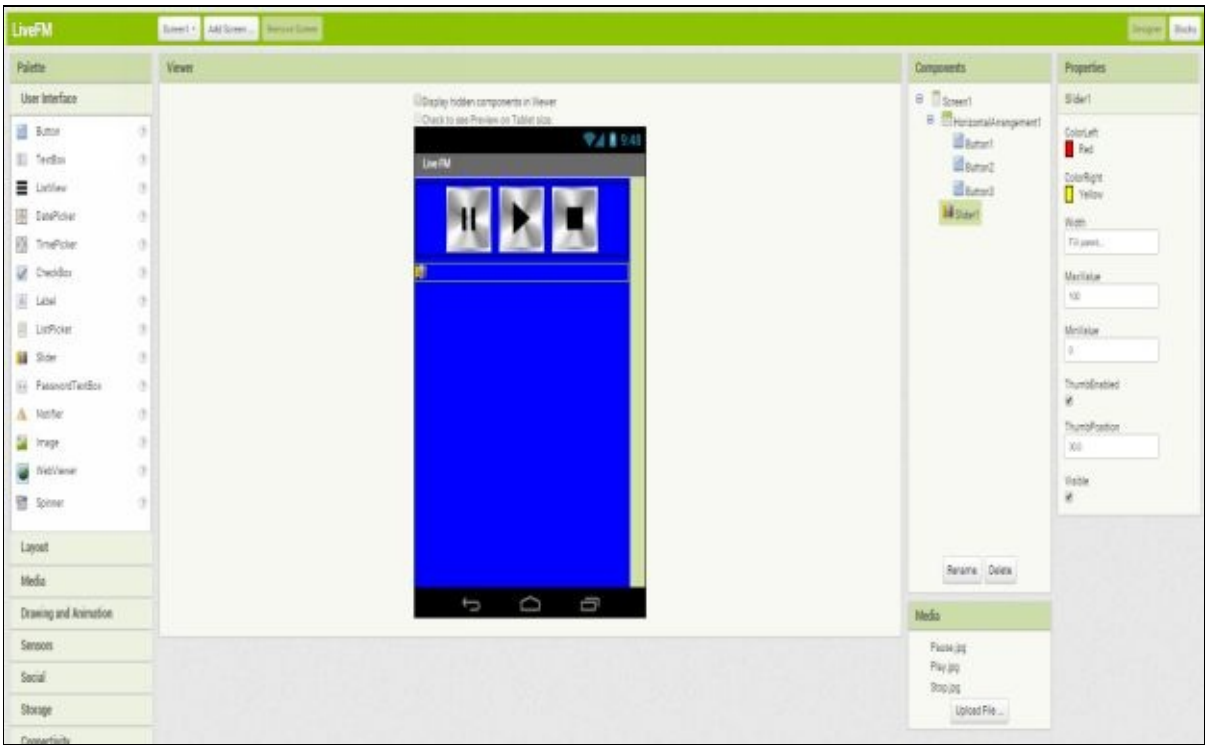
5. Delete the Button's Text and set their Height and Width both to 70 pixels each.



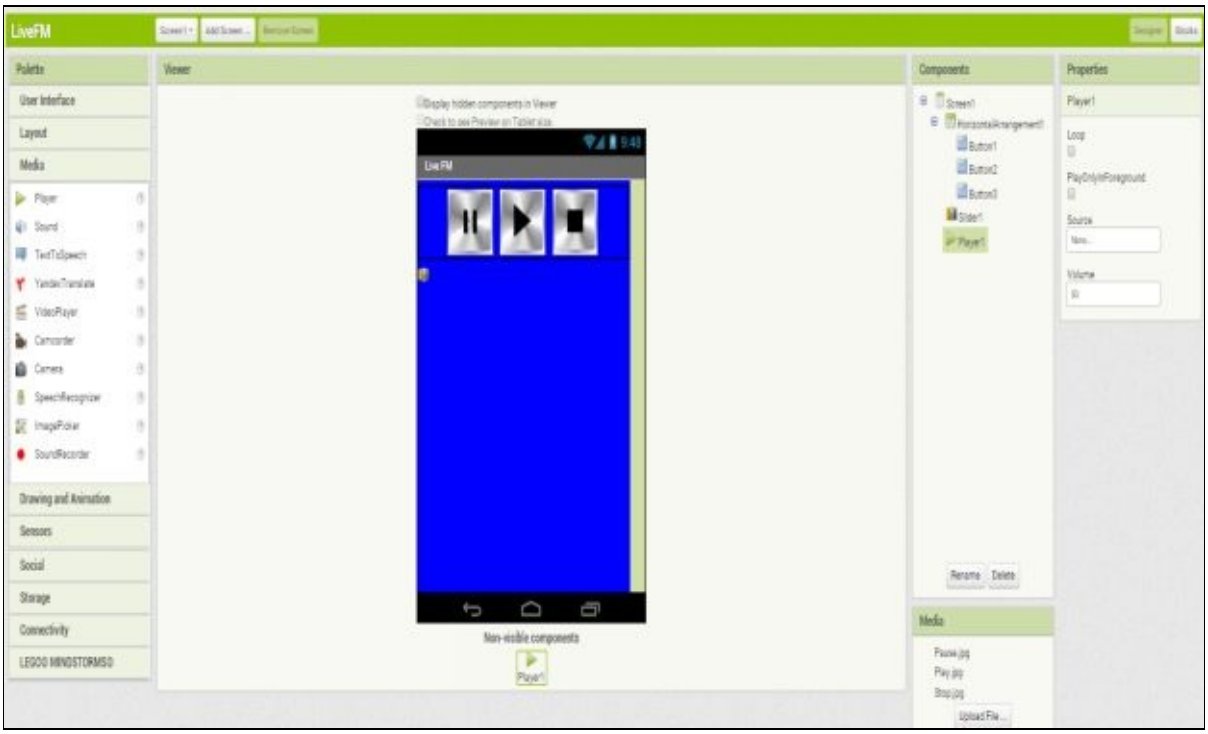
6. Set Button's Images as shown below.



7. Drag a Slider from Palette to Viewer screen and set it’s Properties as shown below.



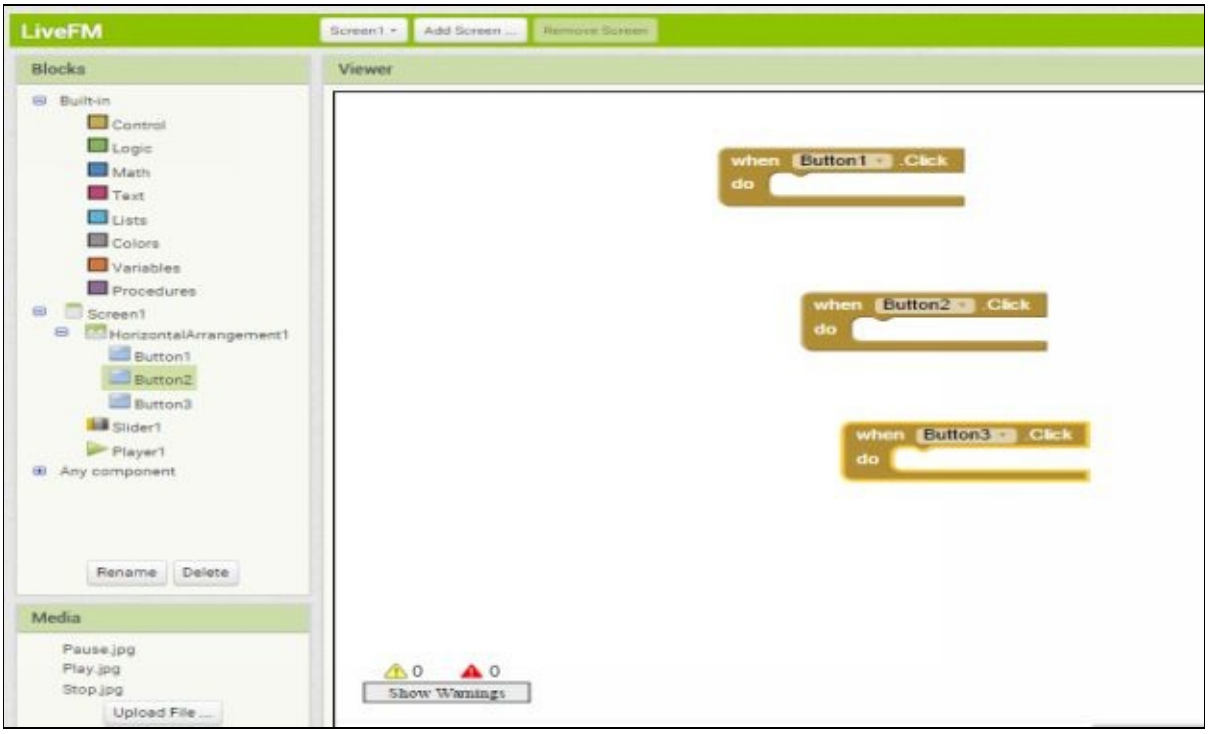
8. Drag a Player component from Palette to Viewer.



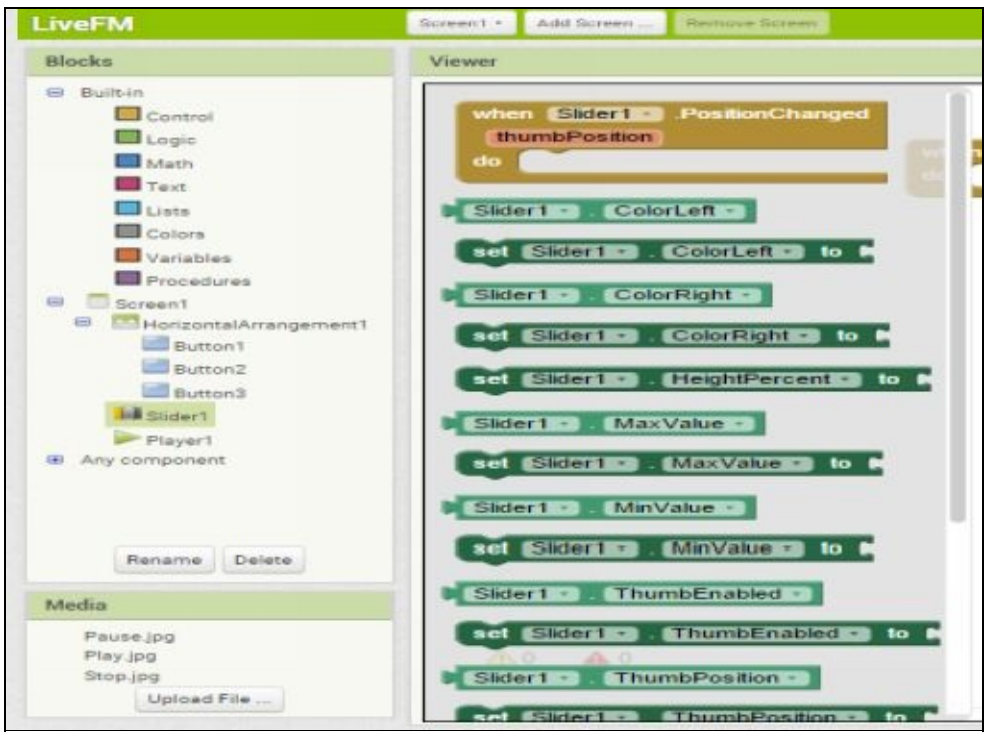
9. Now go to Blocks and Click on Button1 component and select  block.

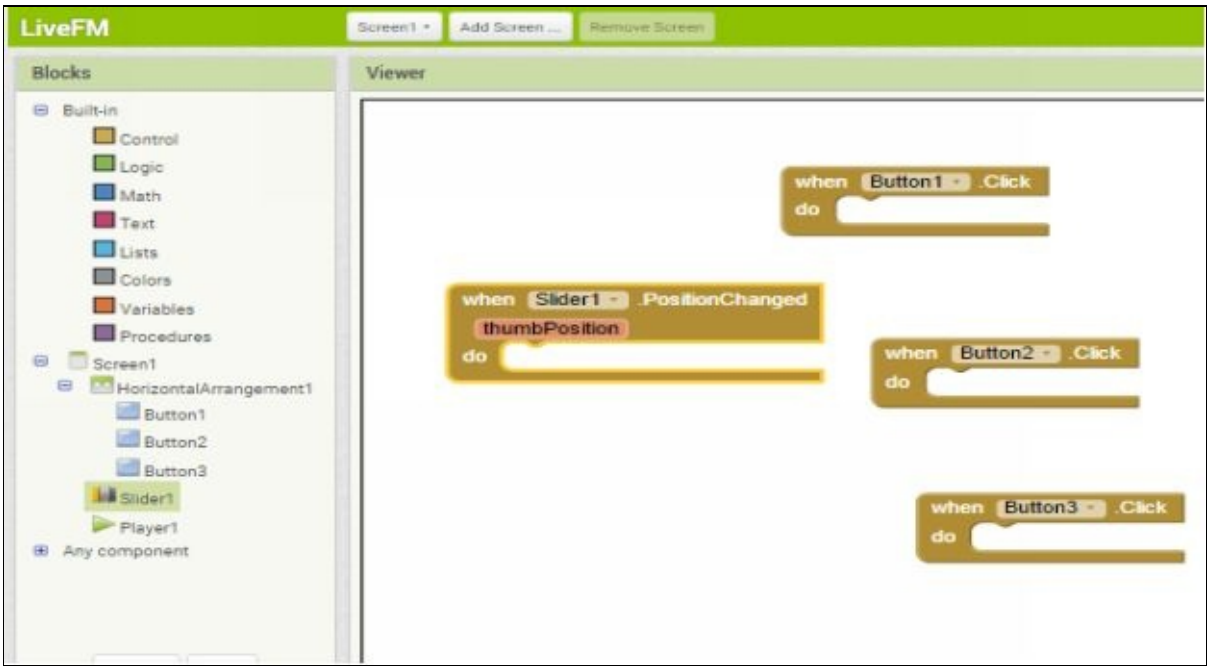
This is an event block which will decide what your app will do when this button will be clicked. Here click is an event for Button1 component.

10. As shown in step 10 select blocks by clicking on Button2 and Button3.

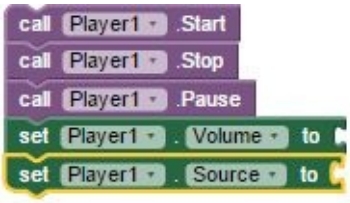


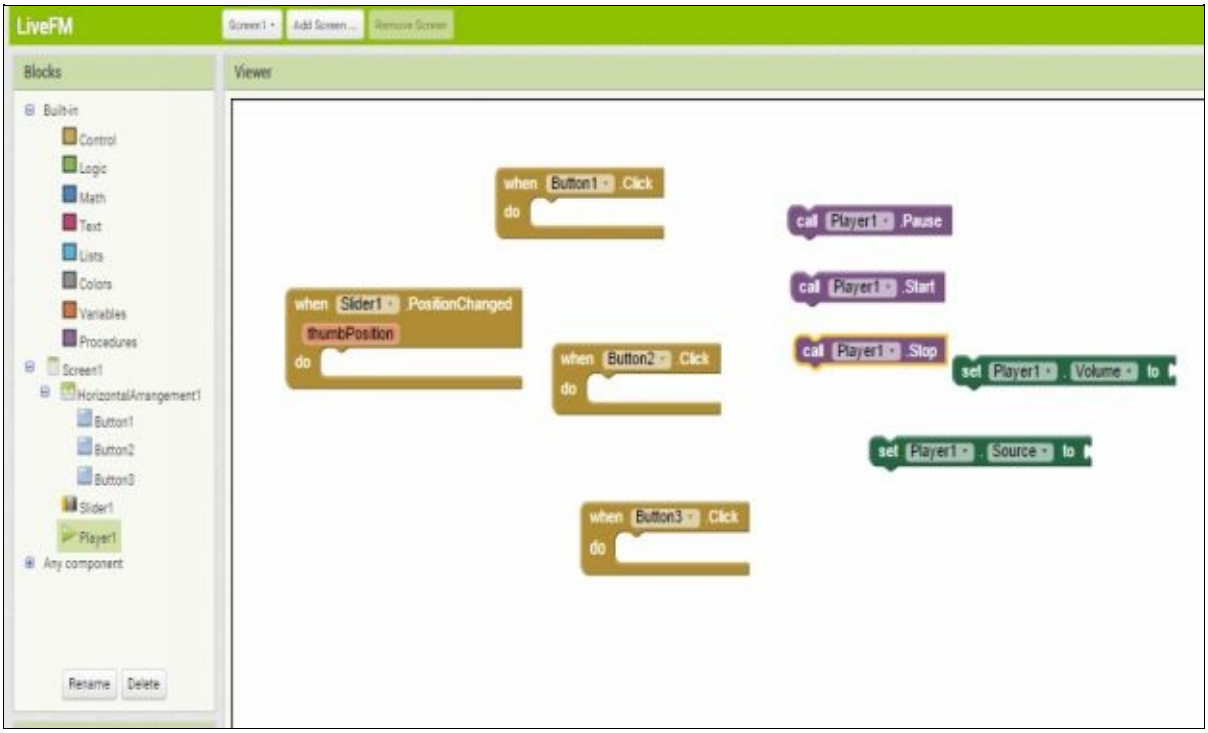
11. Now click on Slider1 and select  block.



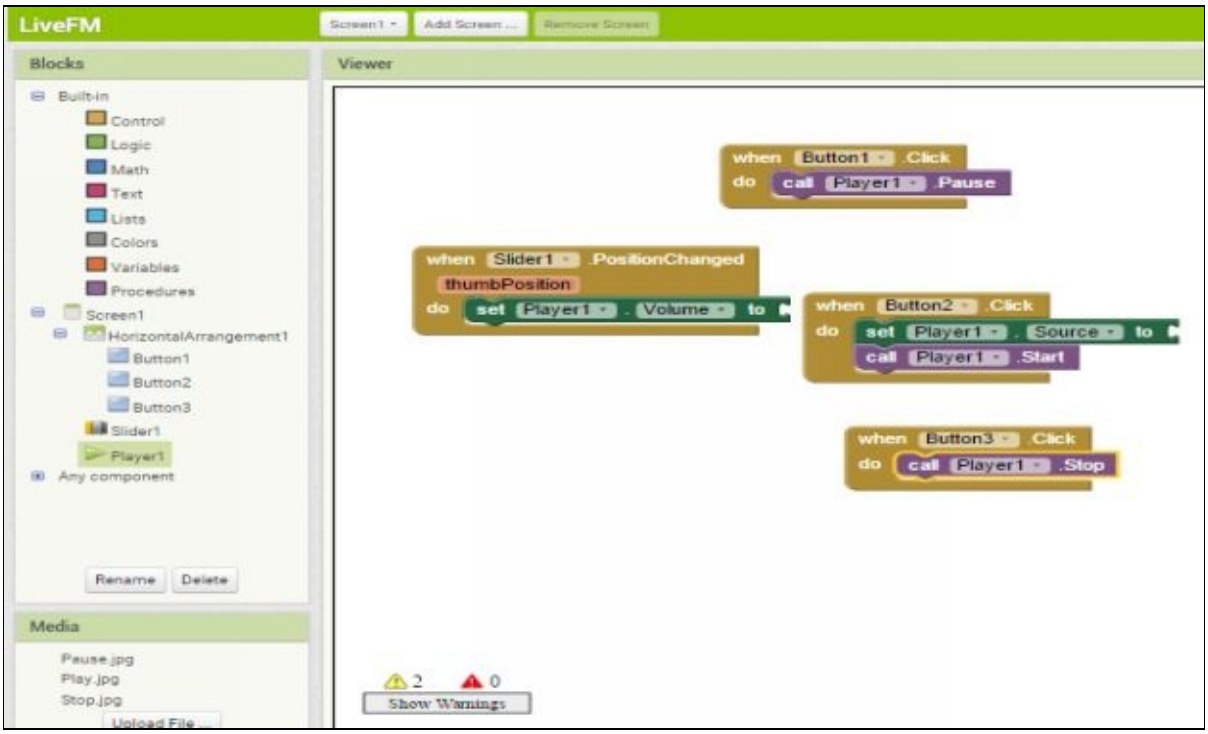


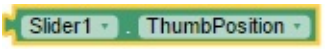
12. Click on Player1 and select below shown blocks.



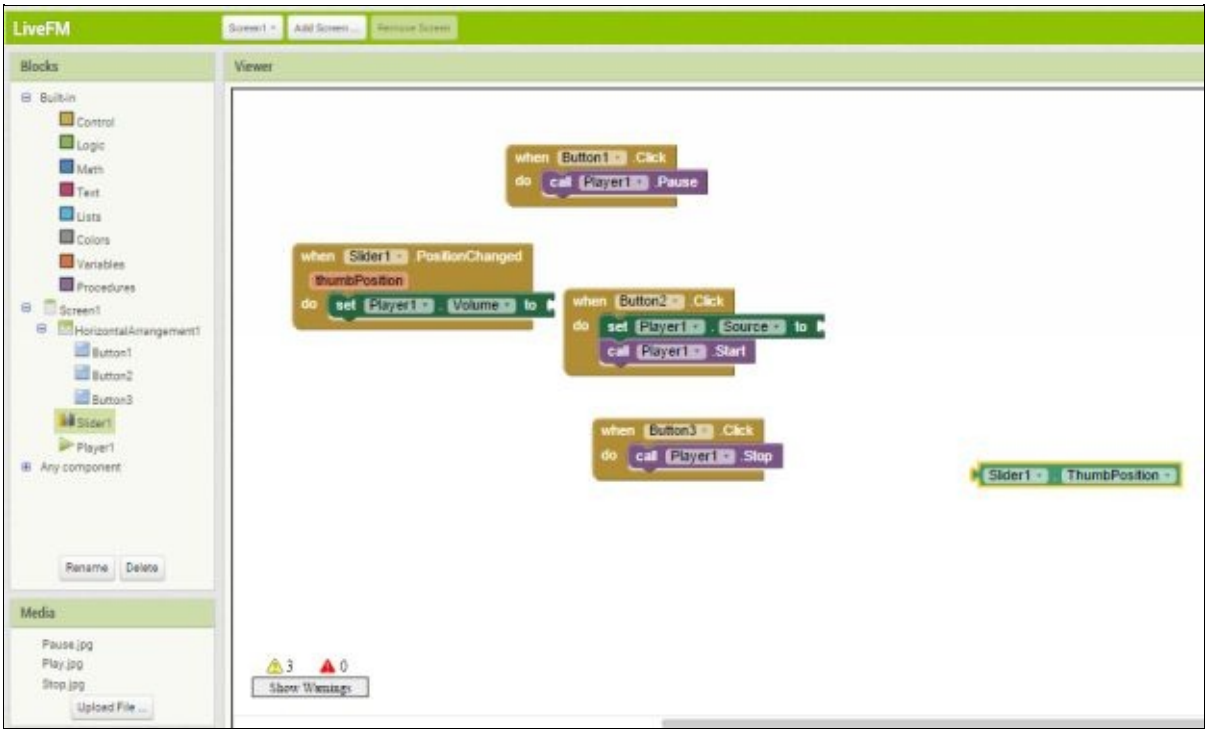


13. Attach the blocks as shown below.

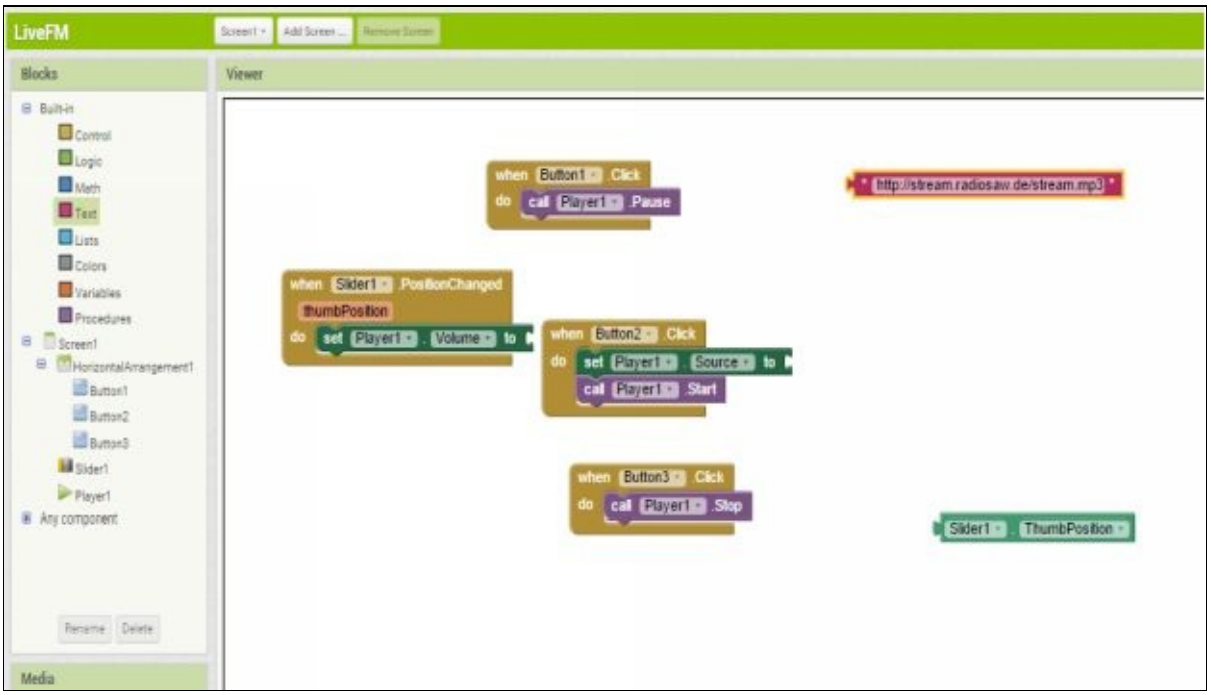


14. Click on Slider1 and select 

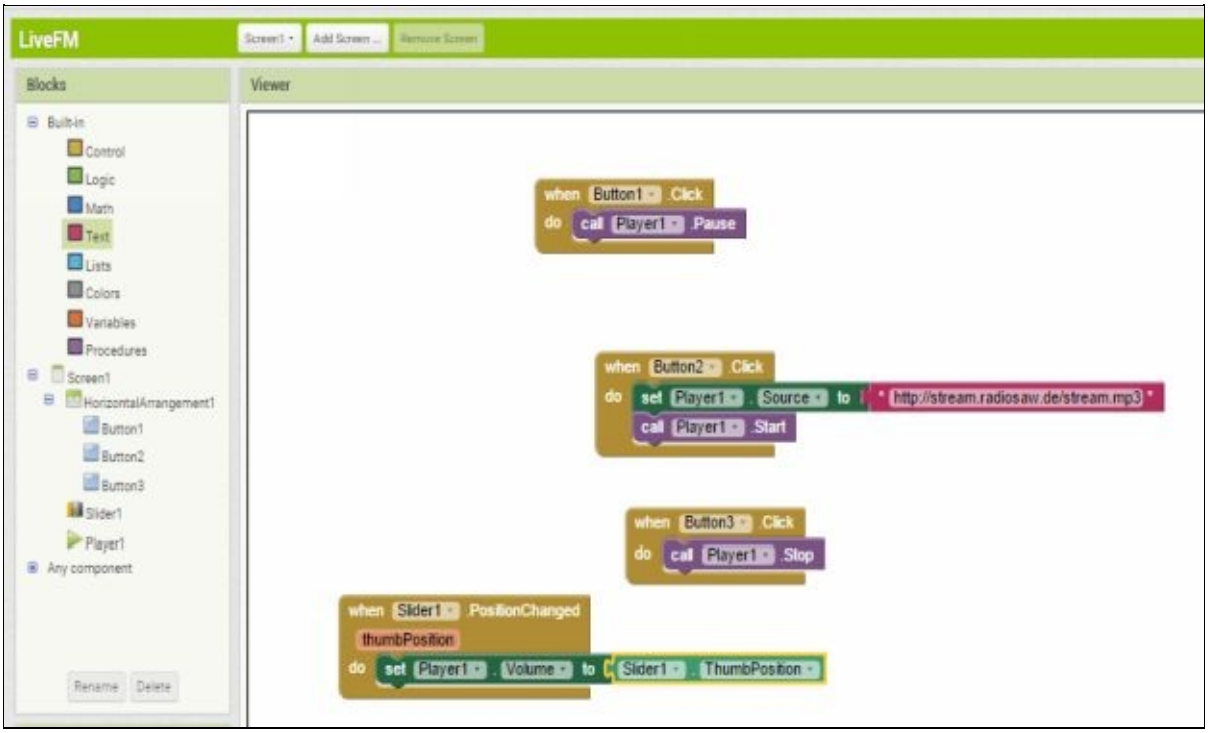
block. This block will represent a value depending on your ThumbPosition on slider.



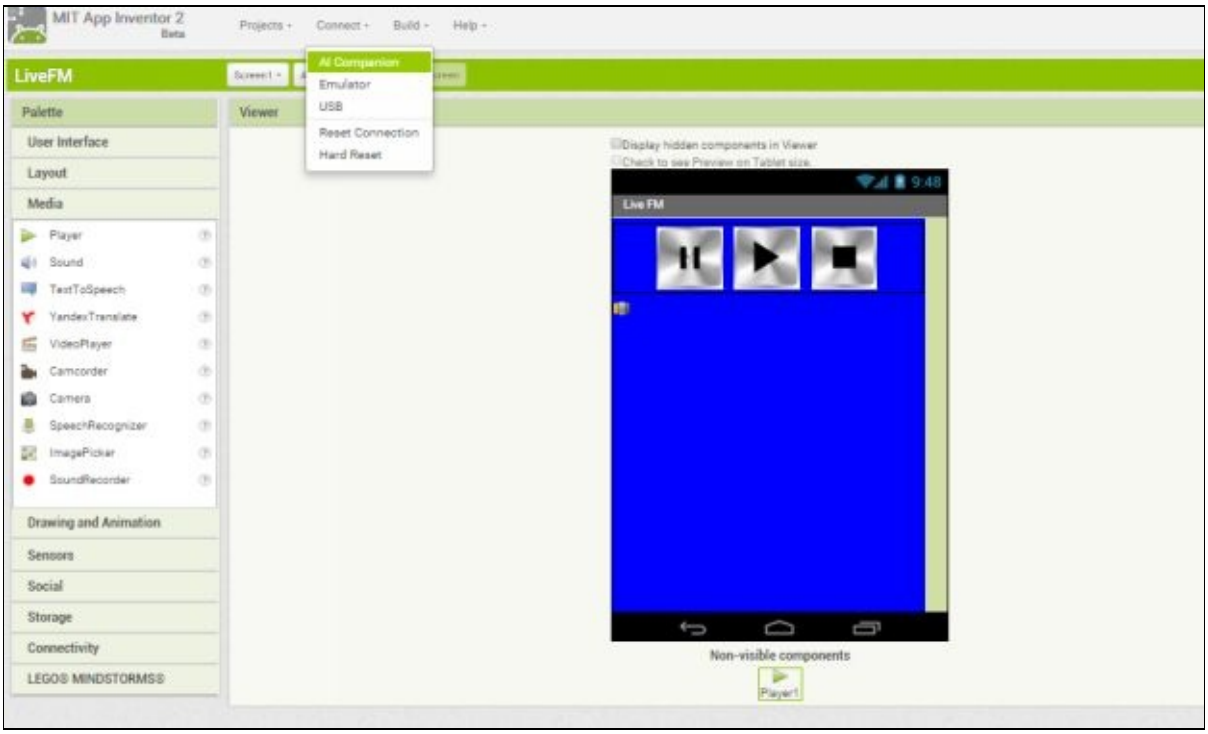
15. Click on Text under Built-in and select  block and type http://stream.radiosaw.de/stream.mp3 inside the block.

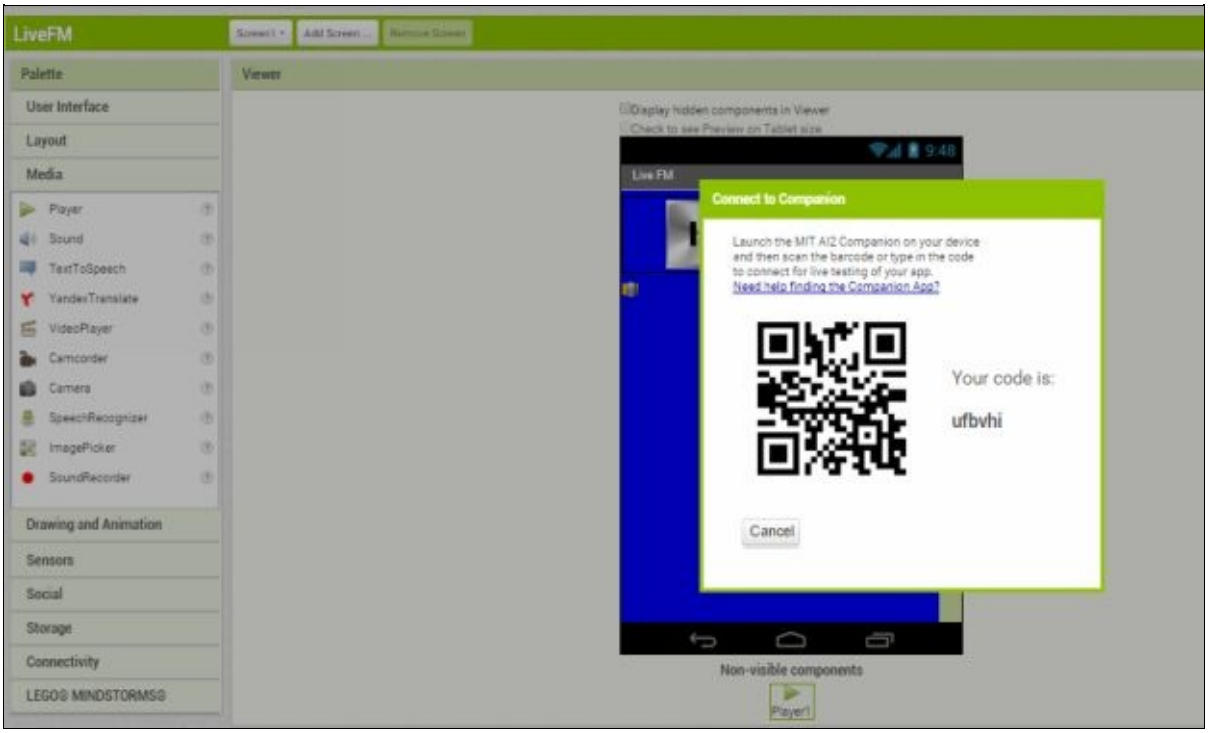


16. Attach the blocks as shown below.



17. Go to Designer and click on Connect then select AI Companion.

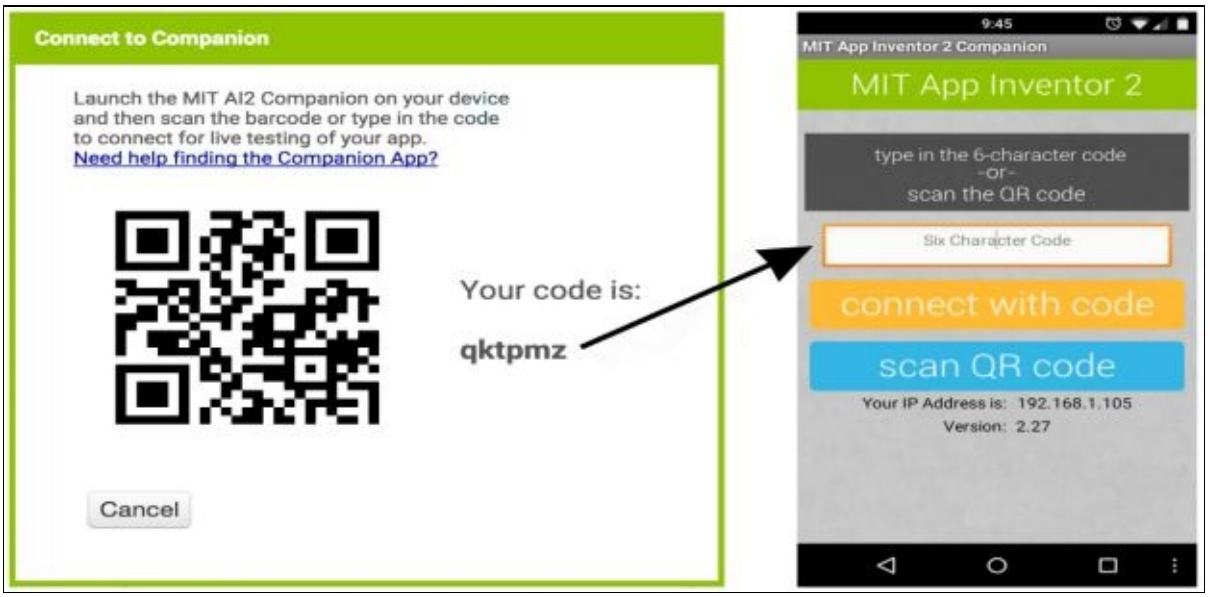




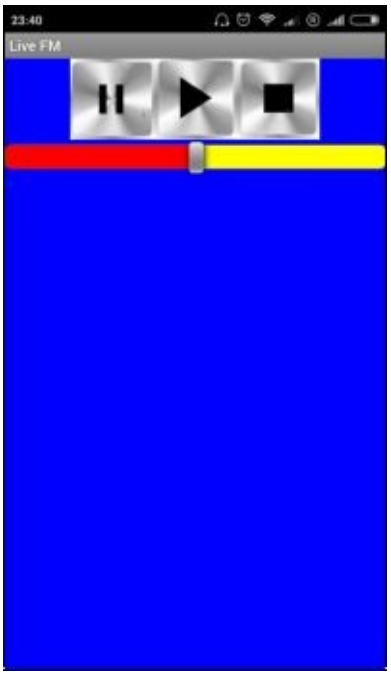
18. Now open MIT App Inventor 2 Companion in your mobile/Tablet.

And enter the code or scan the QR code from your mobile and click on connect with code.

Note:-Your Mobile/Tablet should be connected to same wi-fi/network to which your computer/Laptop is connected.

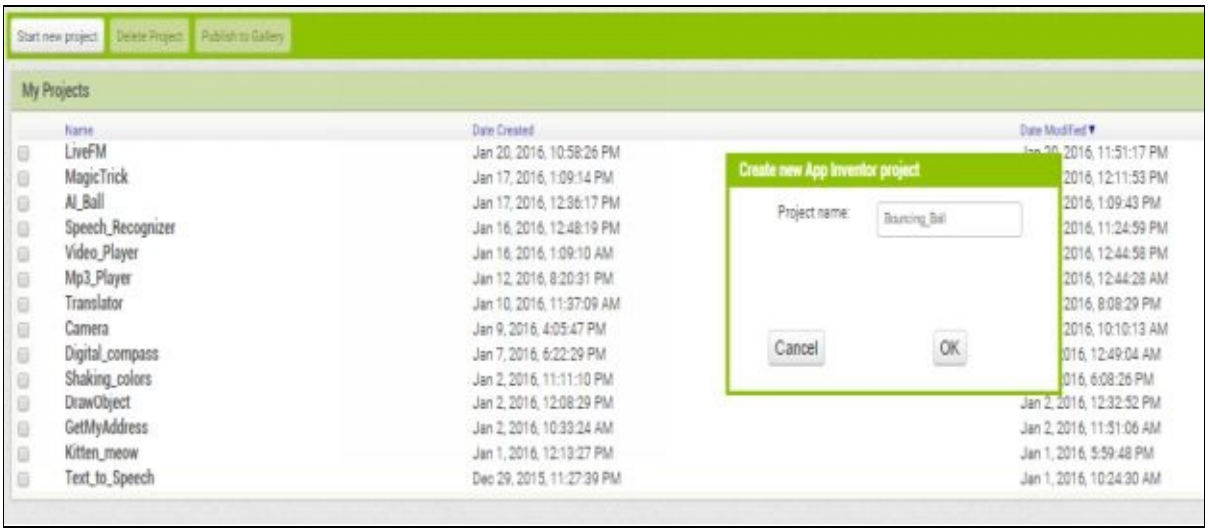


19. Now you should see your app in your phone for live testing. Click on Play Button to listen to FM. Try other buttons also. You can adjust Volume by adjusting the slider.

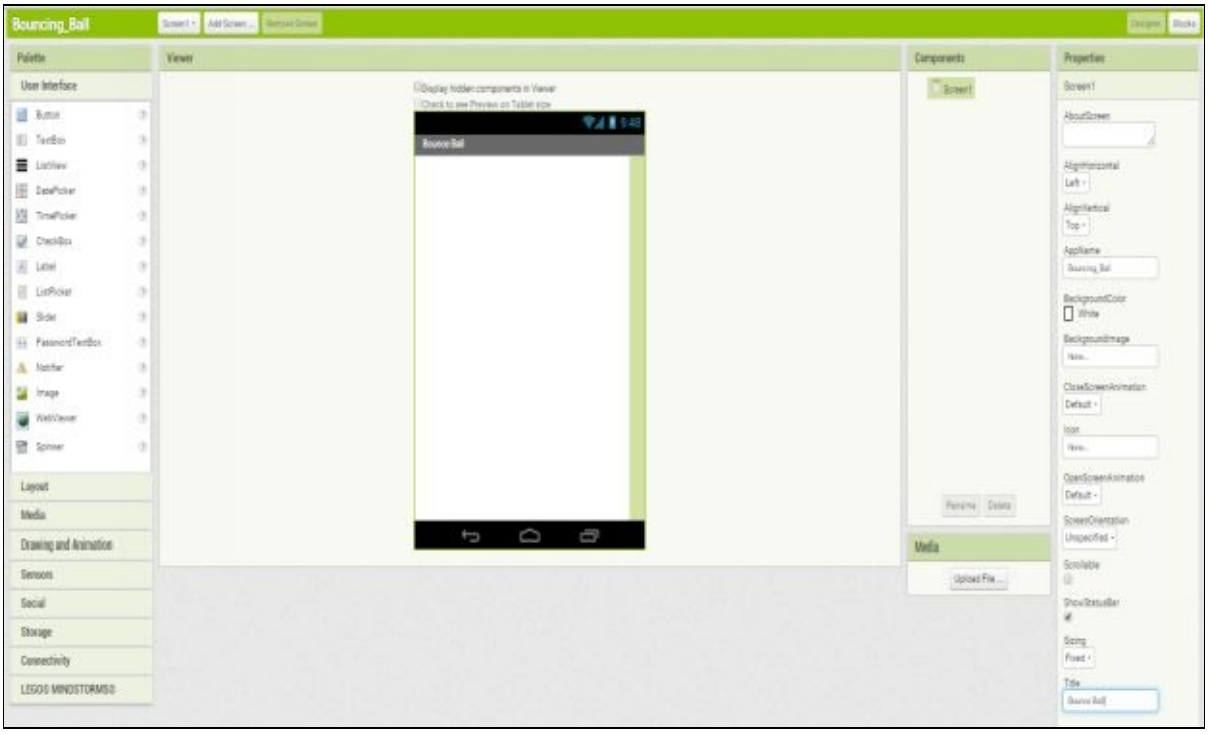


Chapter 18 Bounce Ball App

1. Click on Start new project and give it name Bouncing_Ball and Click OK.

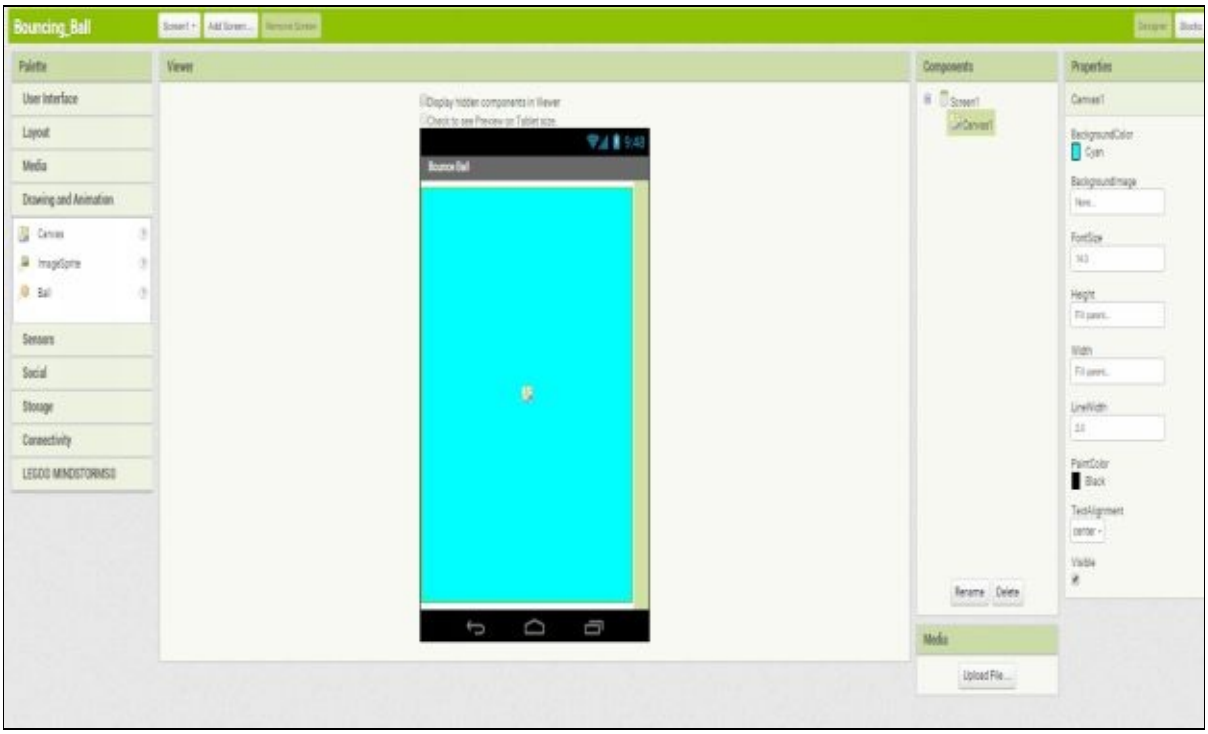


2. Change Screen1 Title to Bounce Ball.

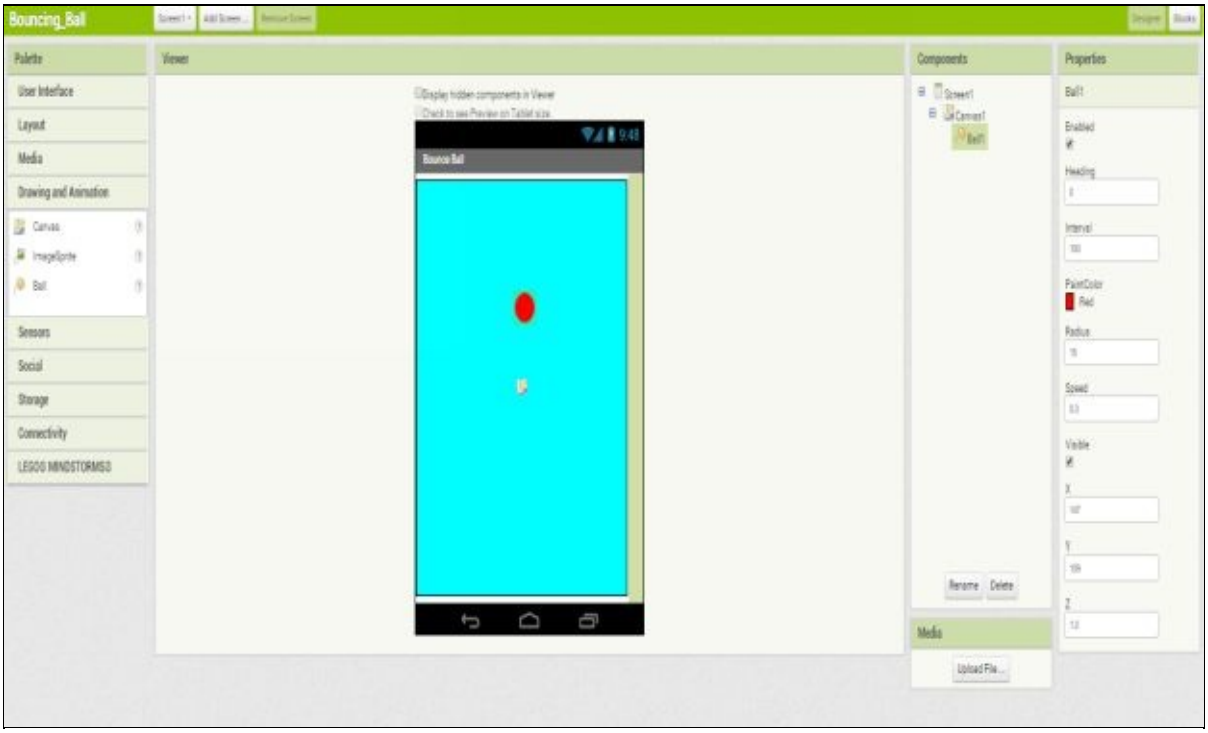


3. Drag a Canvas from Palette to Viewer screen and set it's Height and Width to Fill parent.

Change Canvas1's BackgroundColor to Cyan.

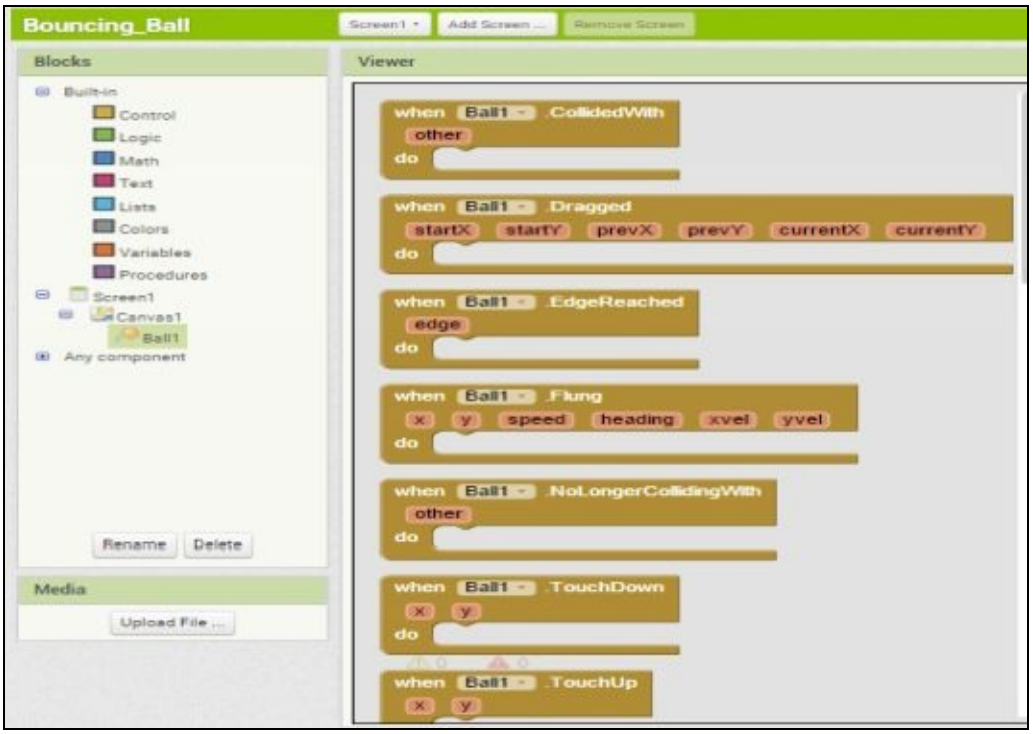


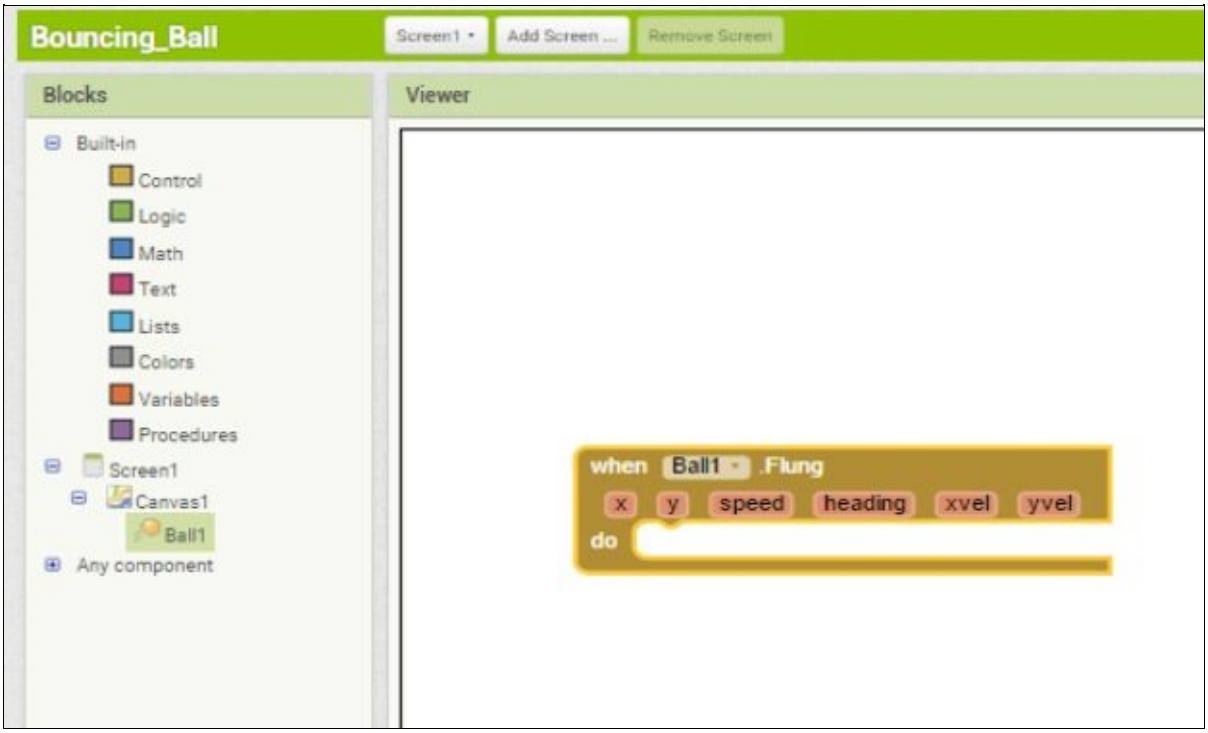
4. Drag a Ball from Palette to Viewer Screen and set it's Radius to 15 and PaintColor to Red.

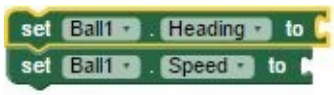


5. Go to Blocks and click on Ball1 and select 

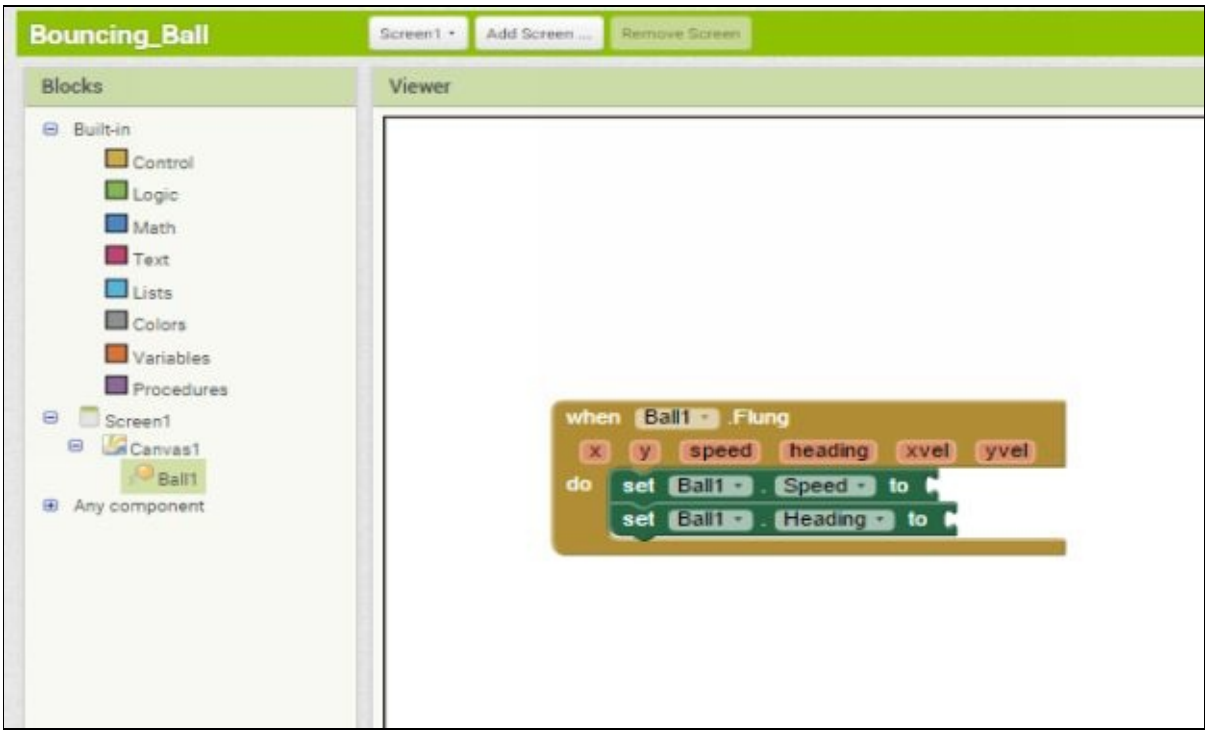
block. This block will identify flung activity on your screen.



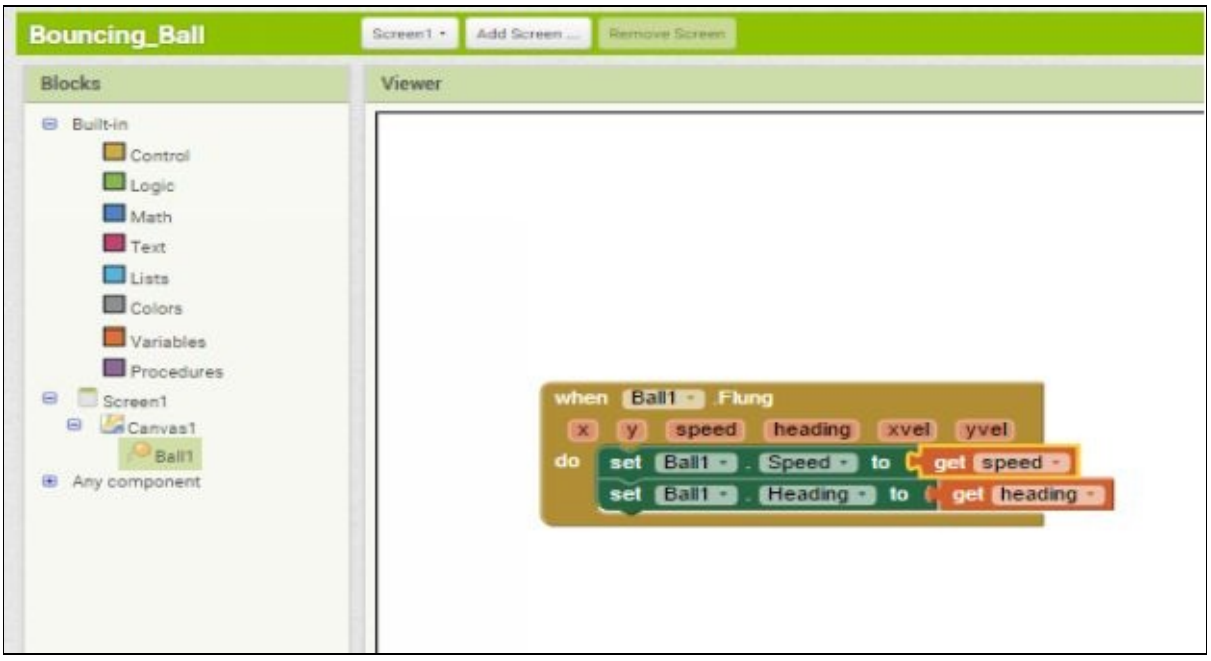


6. Click on Ball1 and select 

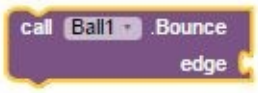
blocks. These block will set ball's moving speed and direction.

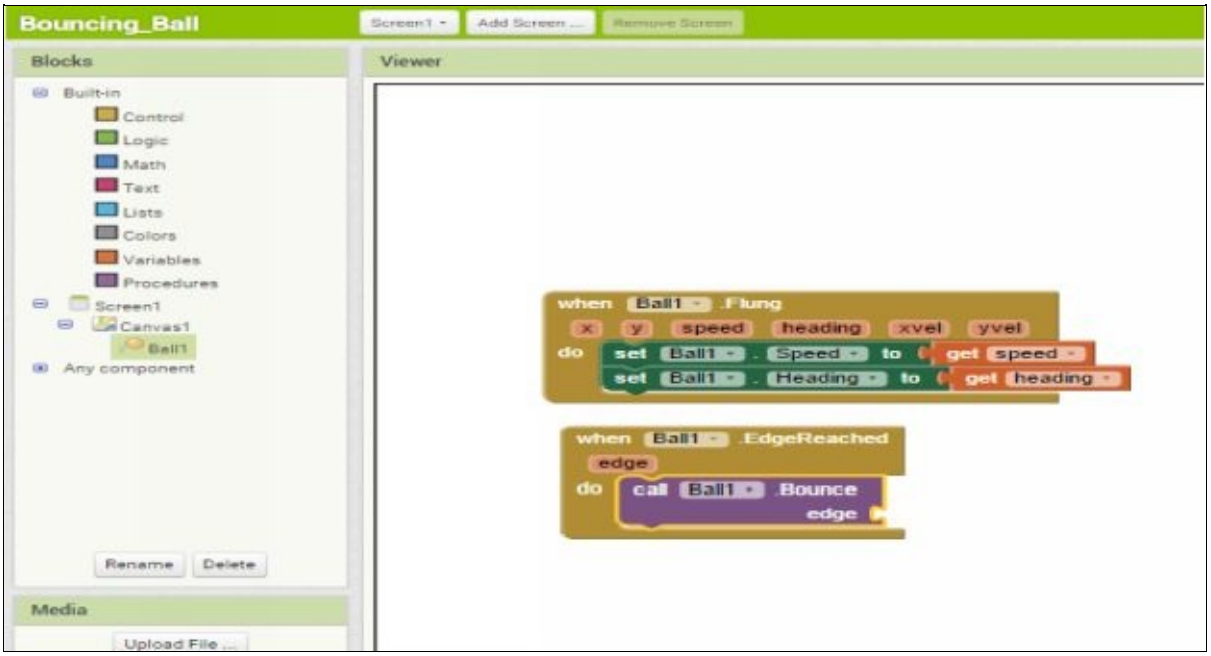


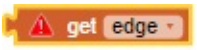
7. Mouse over on speed and heading to get  blocks. These blocks will return the flung direction and speed.



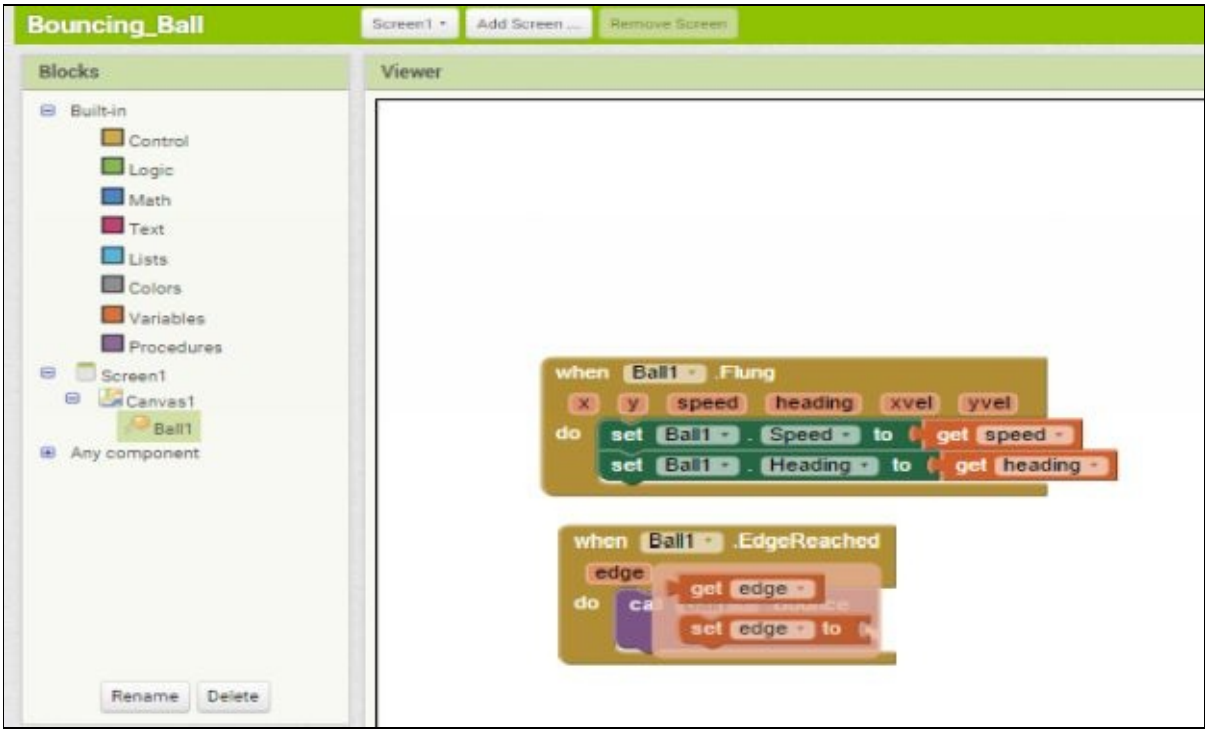
8. Click on Ball1 and scroll down to select below blocks. This block will tell your app what to do when ball has reached an edge on the screen.

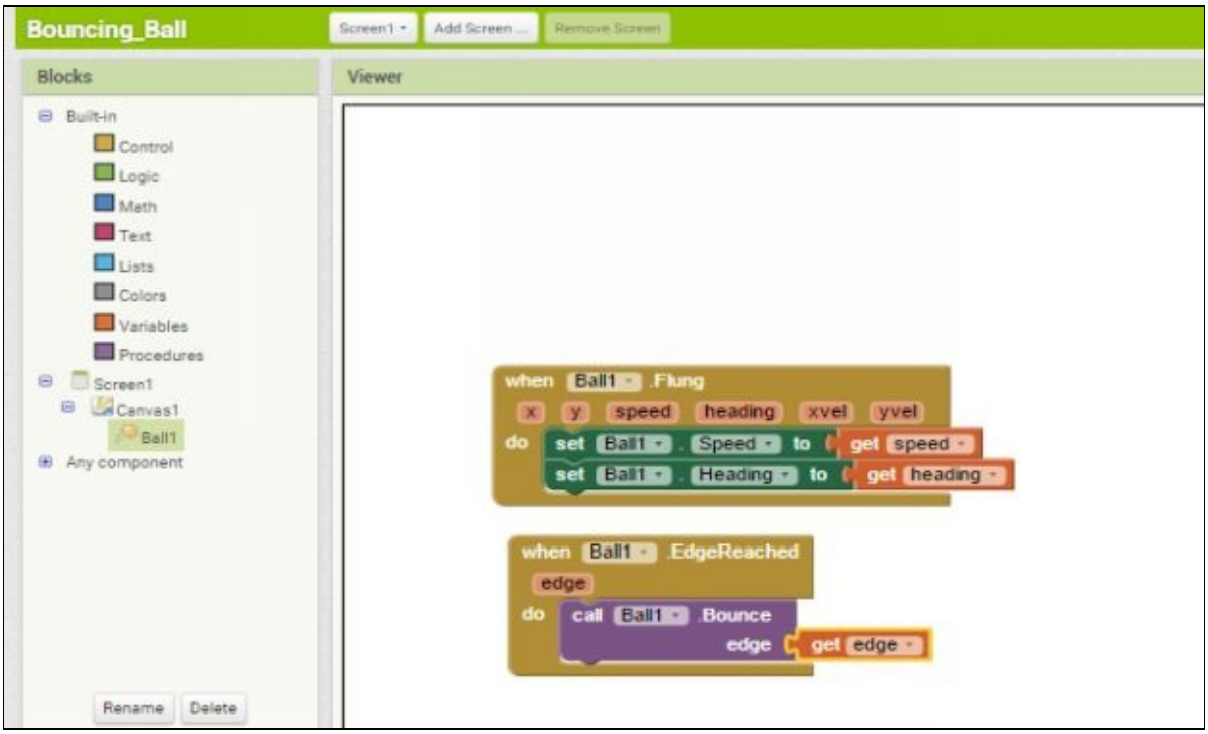




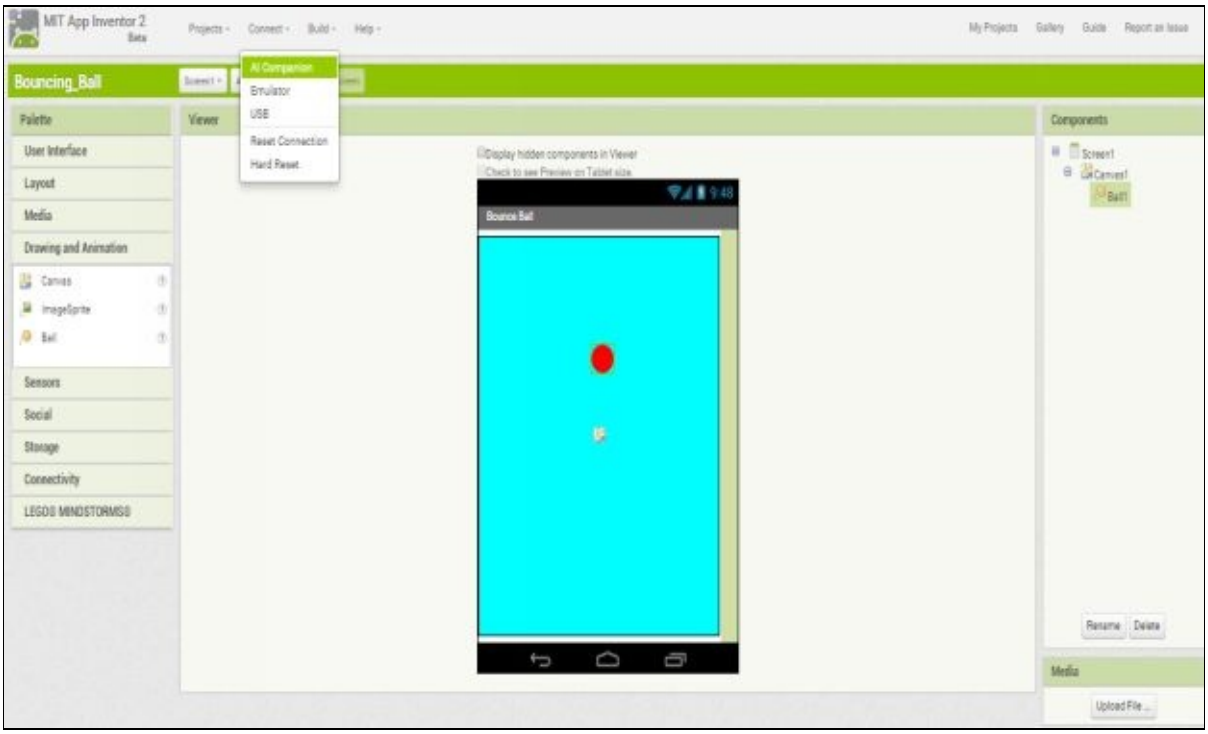
9. Mouse over on edge to get 

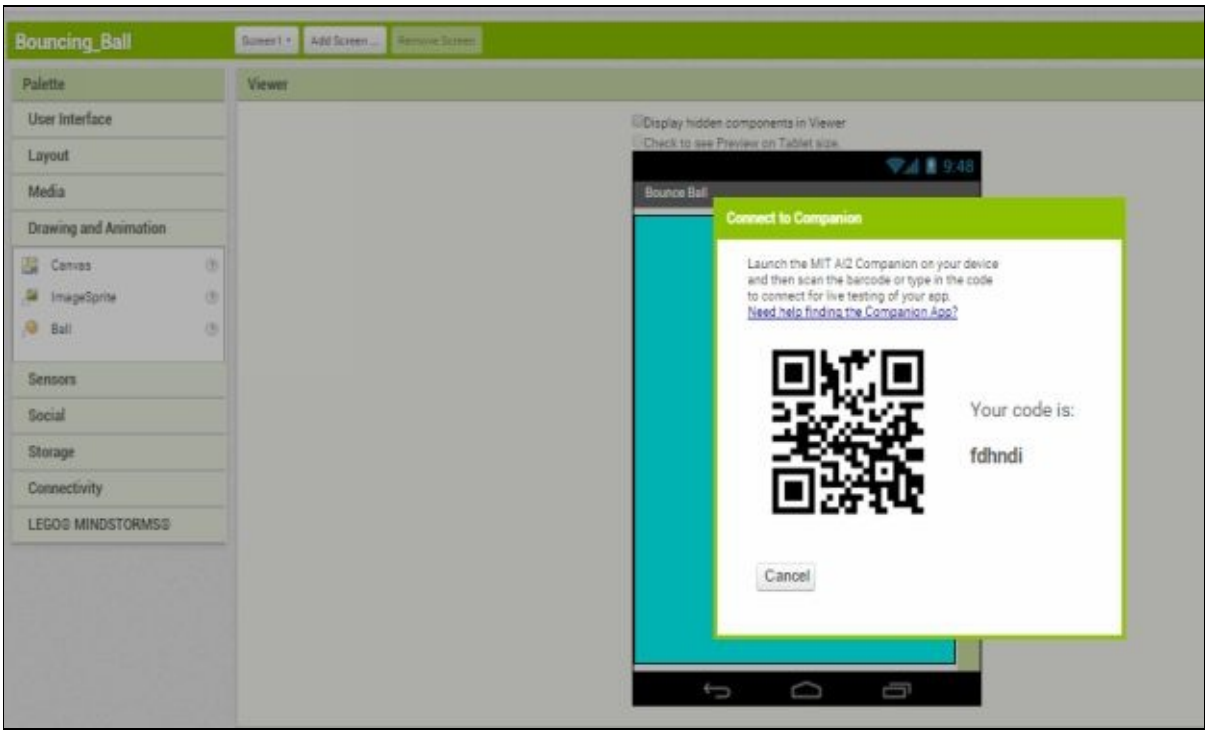
block. This block will return the edge to which ball has reached on the screen. So when ball will reach an edge on the screen , it will bounch on that screen.





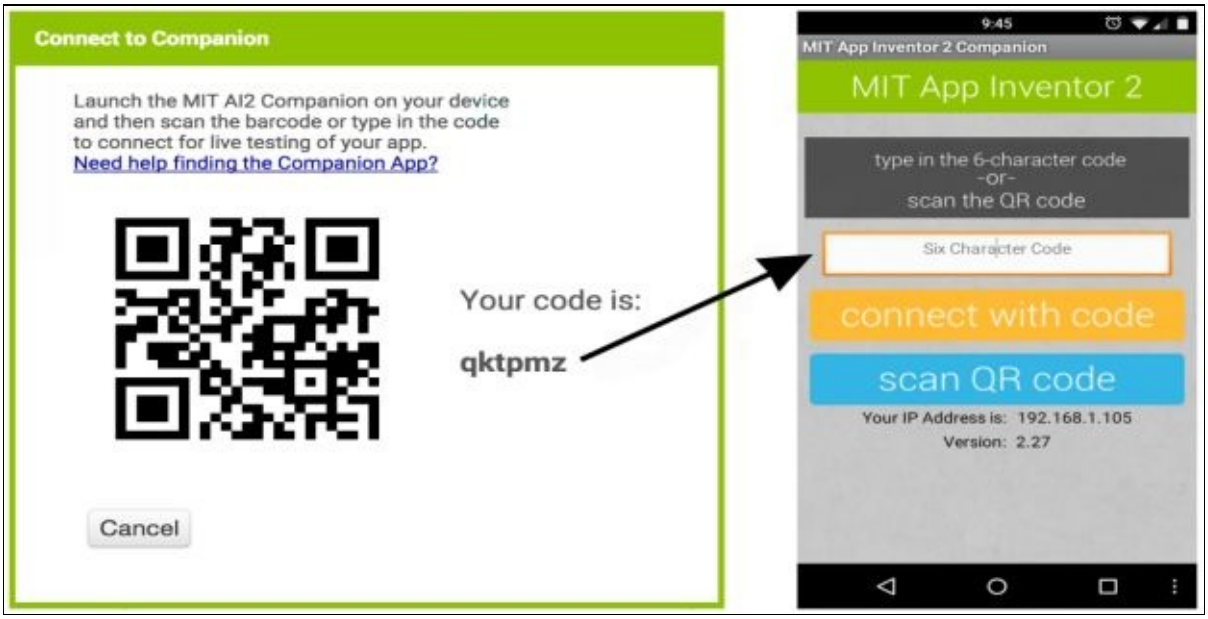
10. Go to Designer and click on Connect then select AI Companion.



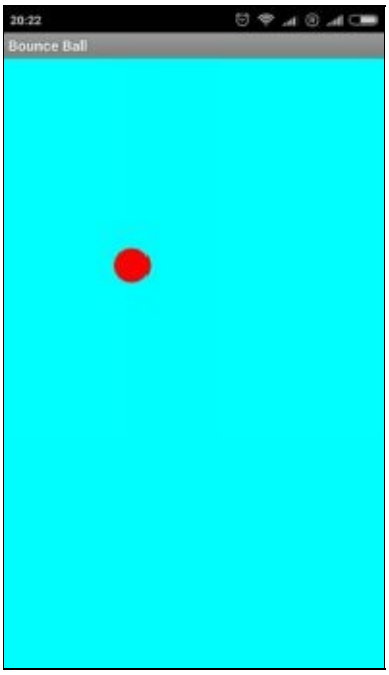


11. Now open MIT App Inventor 2 Companion in your mobile/Tablet.And enter the code or scan the QR code from your mobile and click on connect with code.

Note:-Your Mobile/Tablet should be connected to same wi-fi/network to which your computer/Laptop is connected.

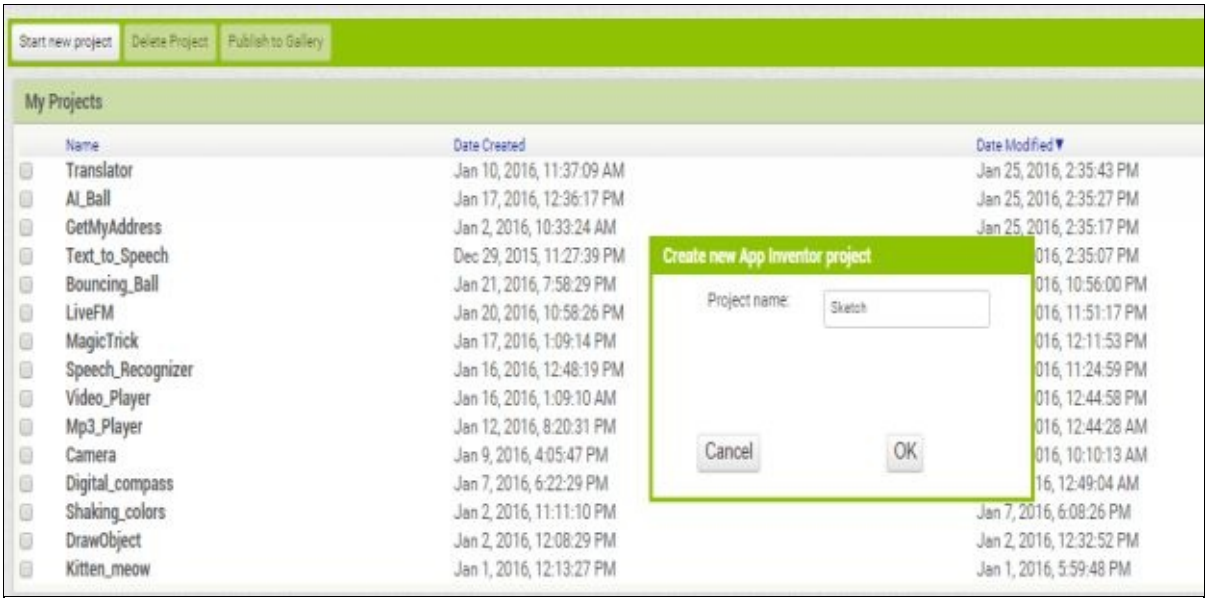


12. Now you should see your app in your phone for live testing. Touch the ball and drag it to some direction .Ball will start moving to that direction and will bounce when reached an edge.

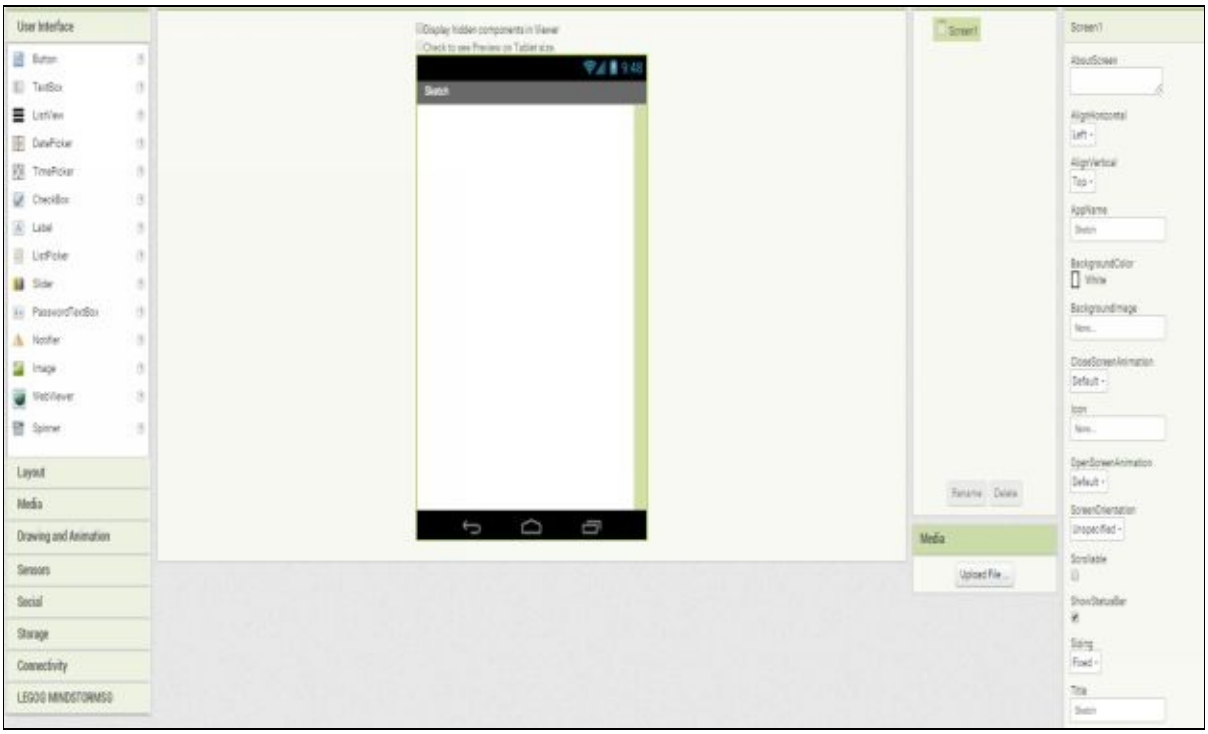


Chapter 19 Sketch-A drawing app

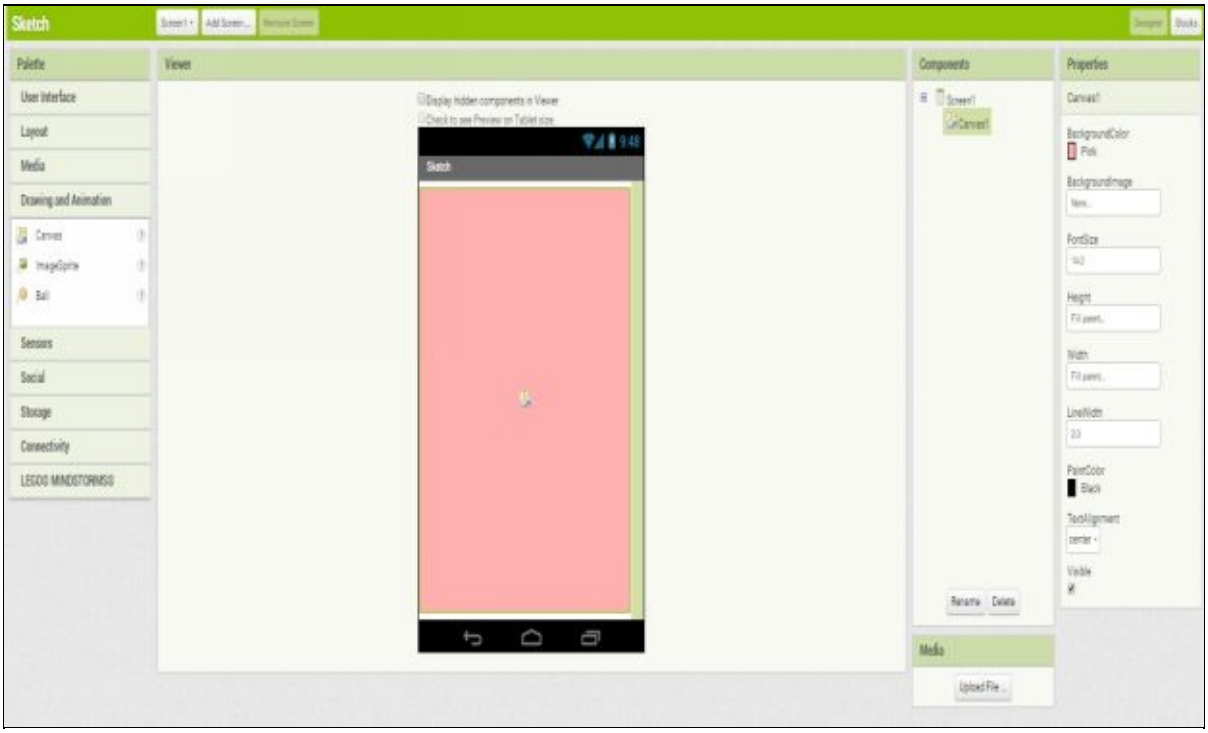
1. Click on Start new project and give it name Sketch and Click OK.



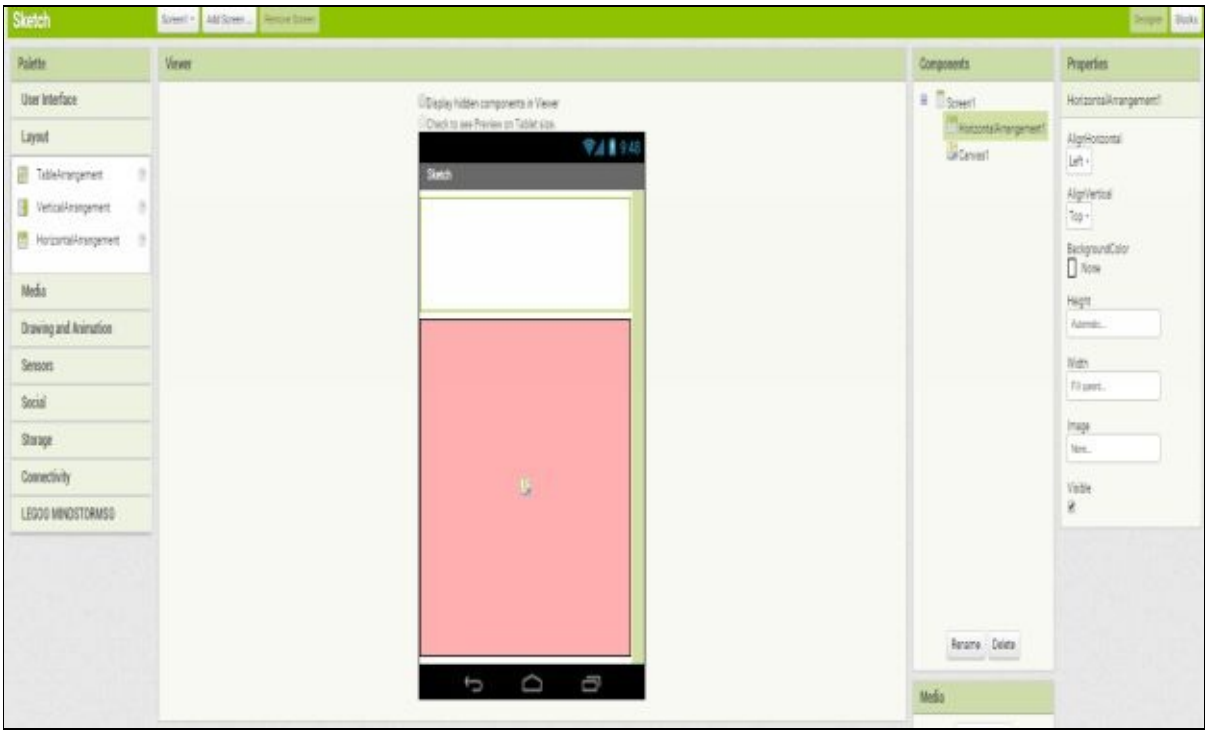
2. Change Screen1 Title to Sketch.



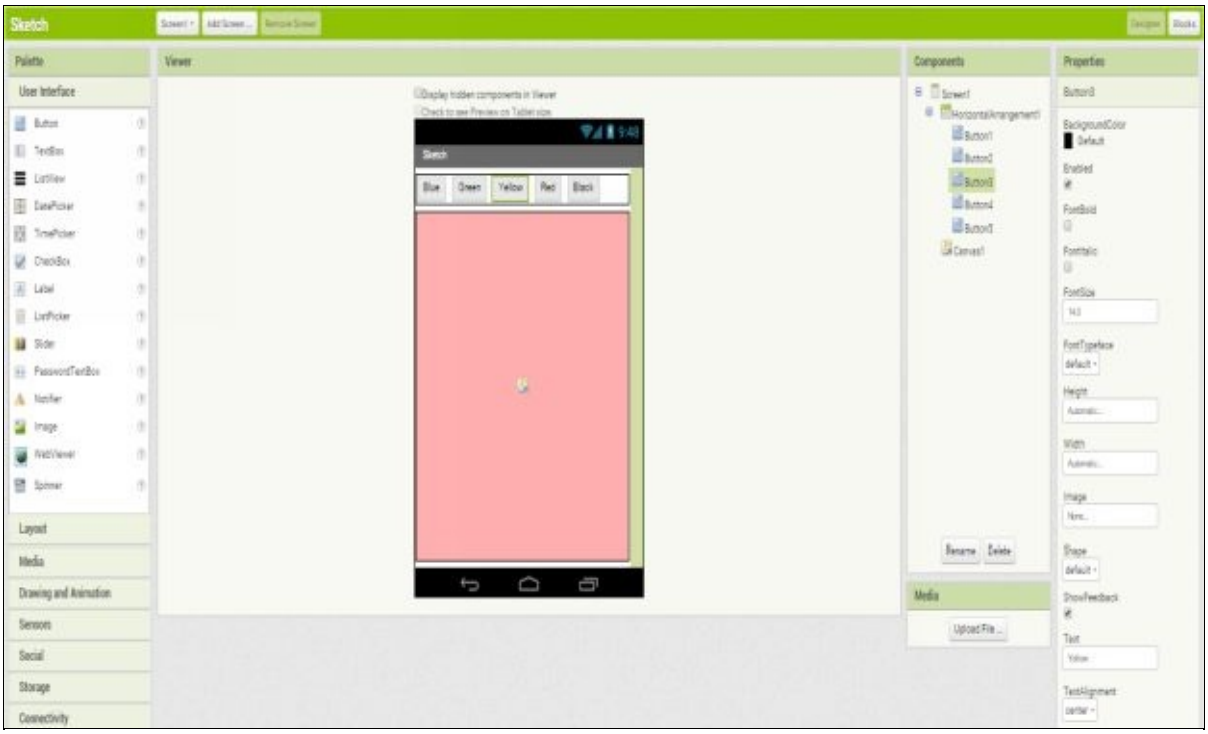
3. Drag a Canvas from Palette to Viewer screen and set it’s Height and Width to Fill parent and change its BackgroundColor to Pink.



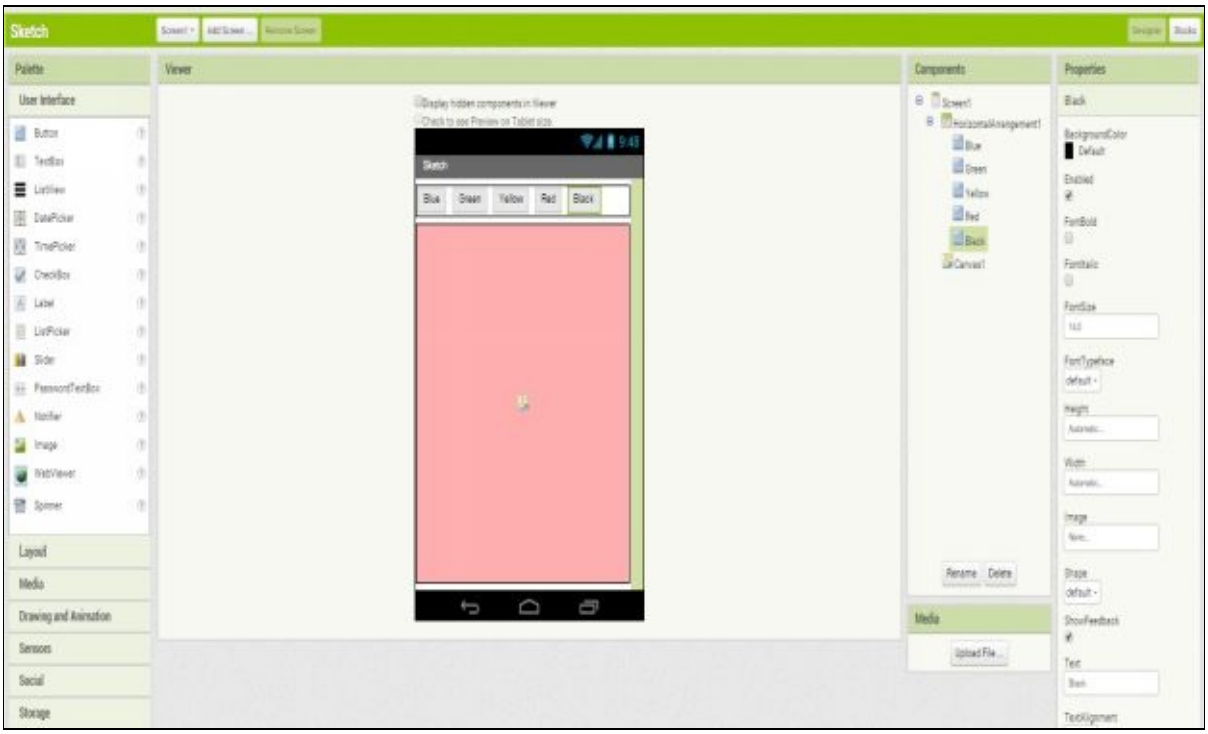
4. Drag a HorizontalArrangement component from Palette to Viewer and set it’s Width to Fill parent and change BackgroundColor to None.



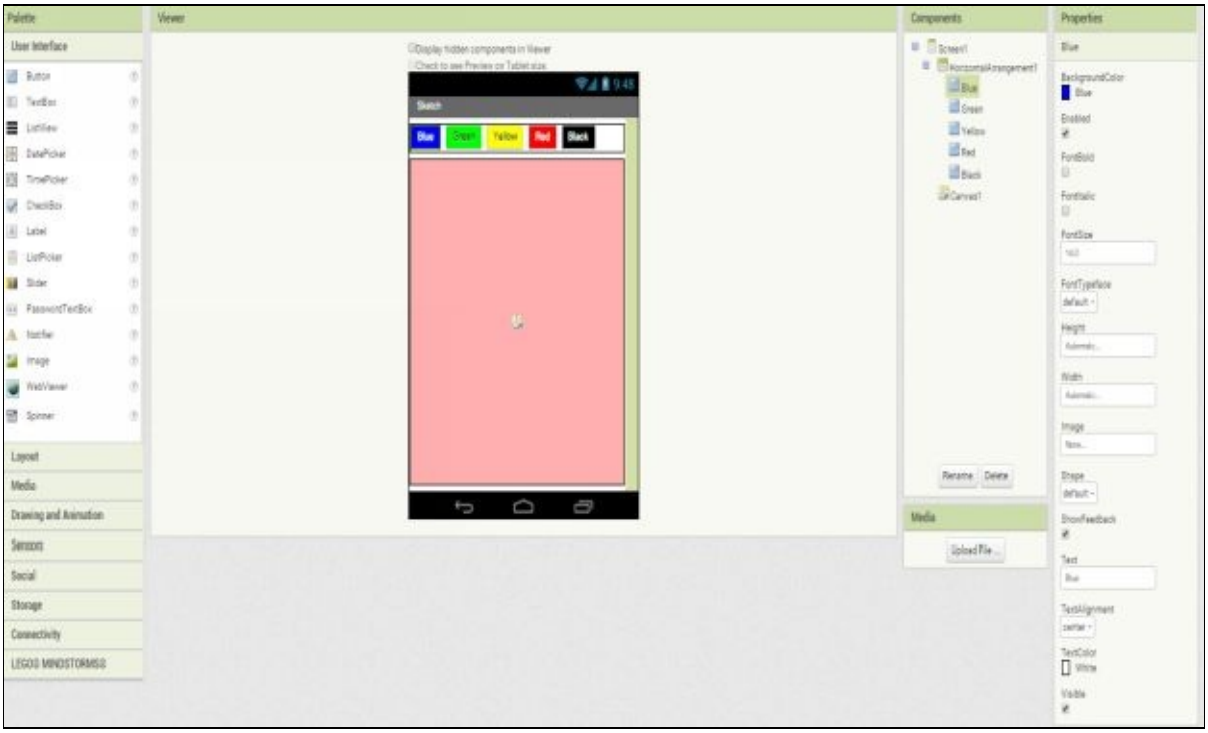
5. Now drag 5 Buttons from Palette to Viewer and change it’s Text as shown below.



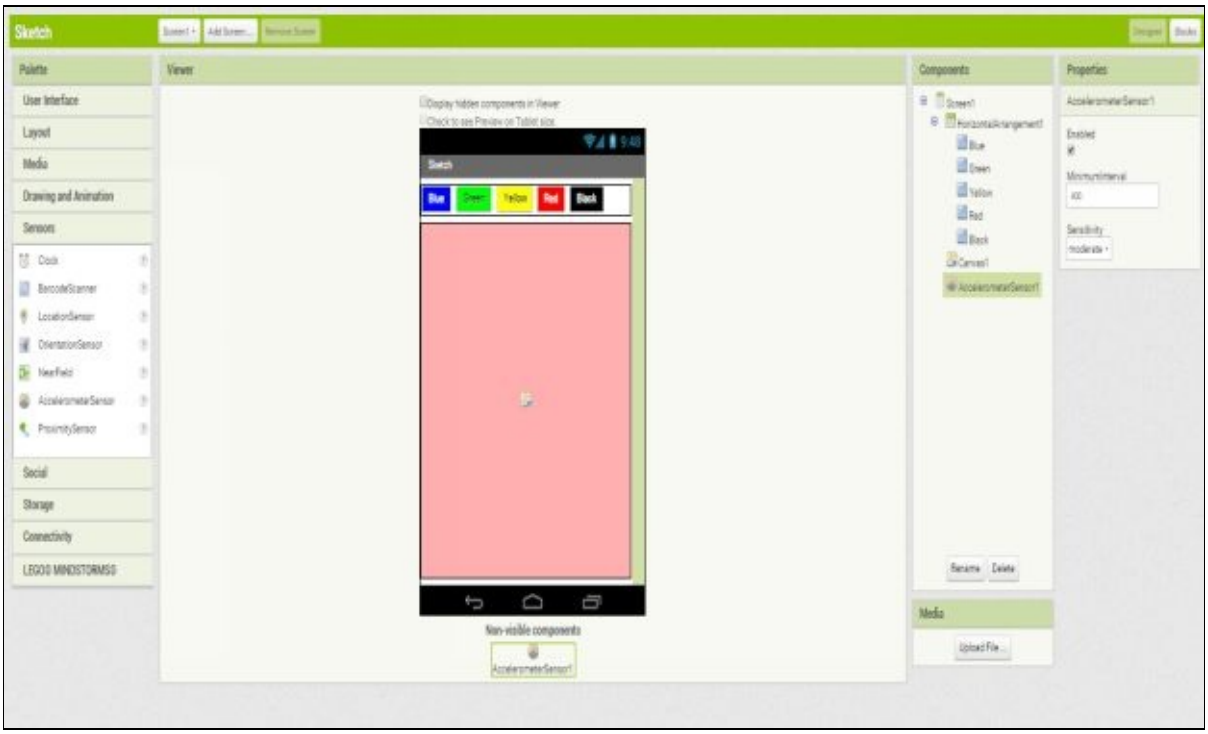
6. Change Button Names as well.



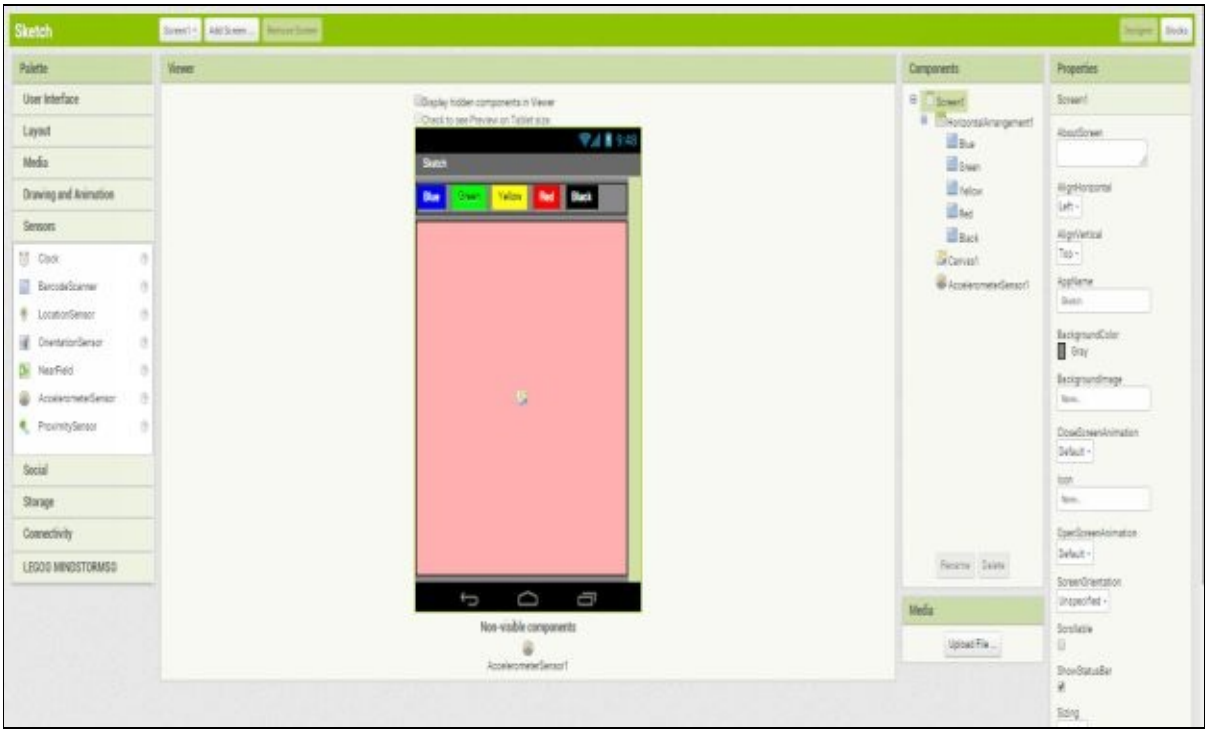
7. Change Button BackgroundColors and TextColors.



8. Drag an Accelerometer (Non-visible) component from Palette to Viewer.

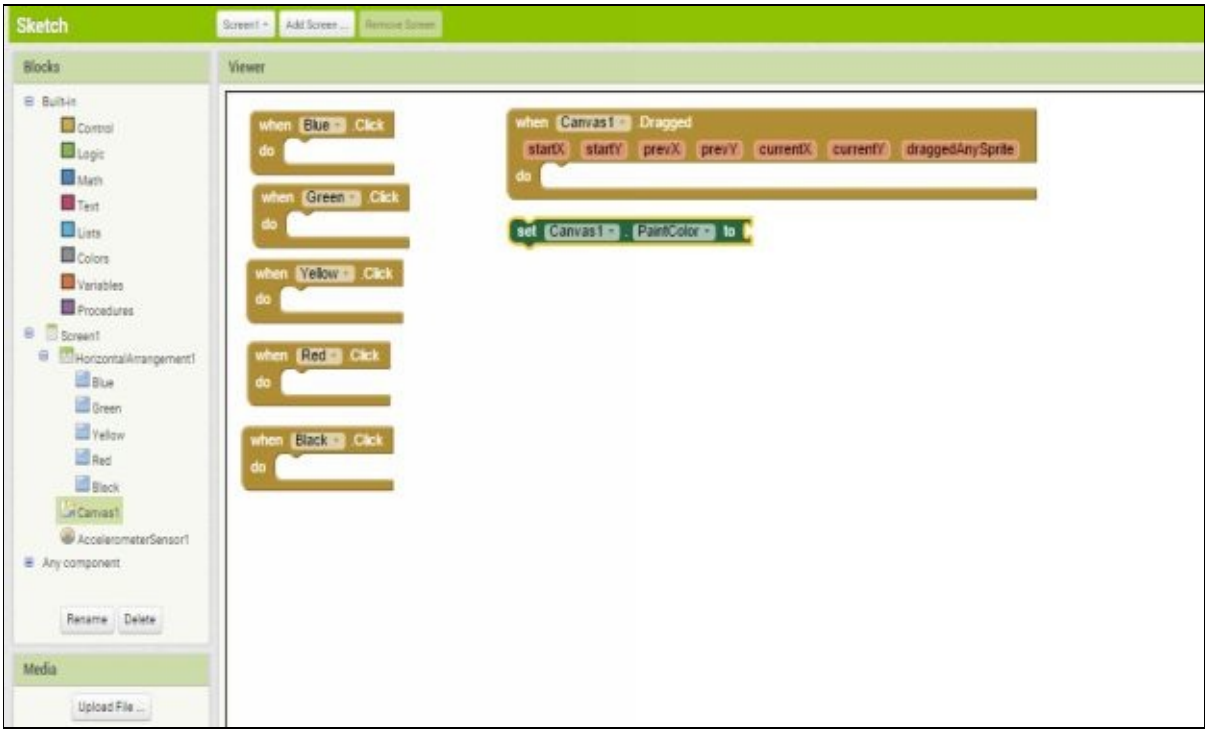


9. Change Screen1 BackgroundColor to Gray.



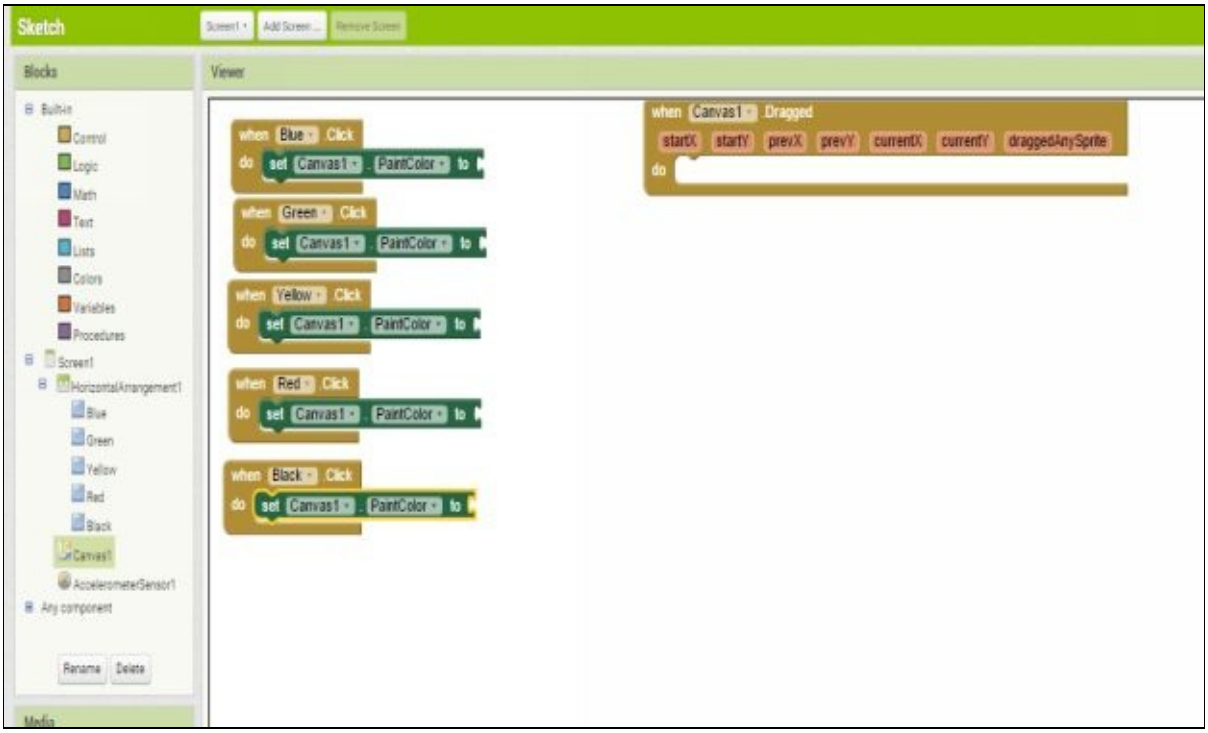
10. Go to Blocks and select blocks as shown below. These blocks will decide what to do when clicked on each of the button.

11. Now click on Canvas1 and select these two blocks. This block will identify dragging activity on your canvas and decide what to do when dragged.

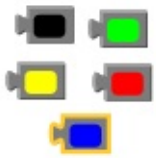


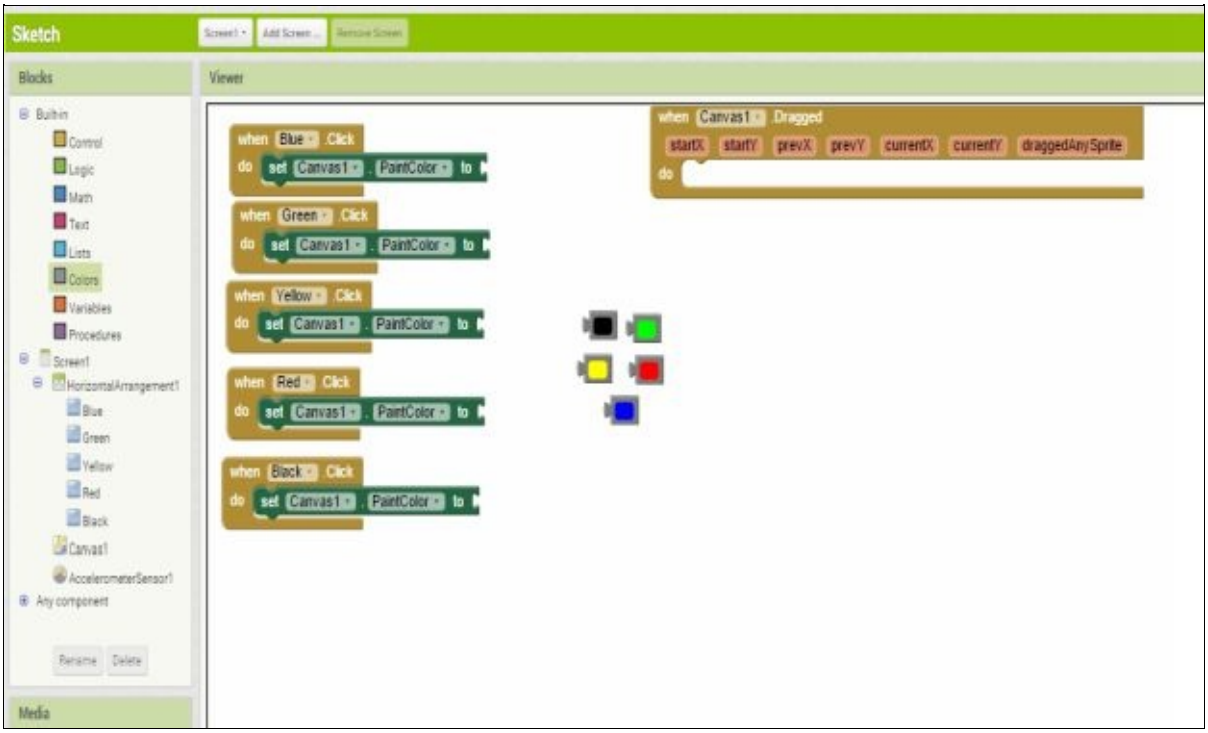
12. Now copy and paste 

block 5 times and connect the blocks as shown below. This block will change the PaintColor of the Canvas when clicked on each of the button.

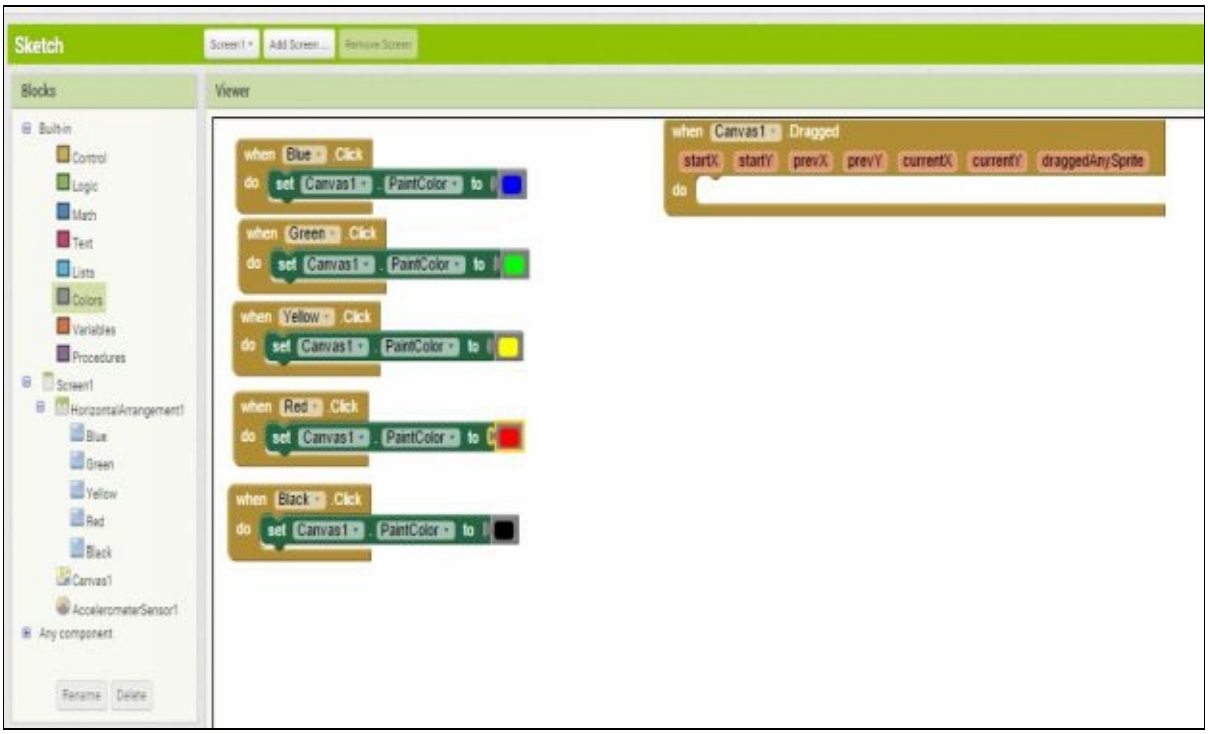


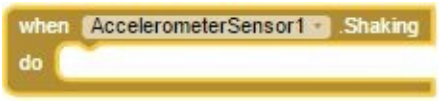
13. Now click on Colors and select these color blocks.

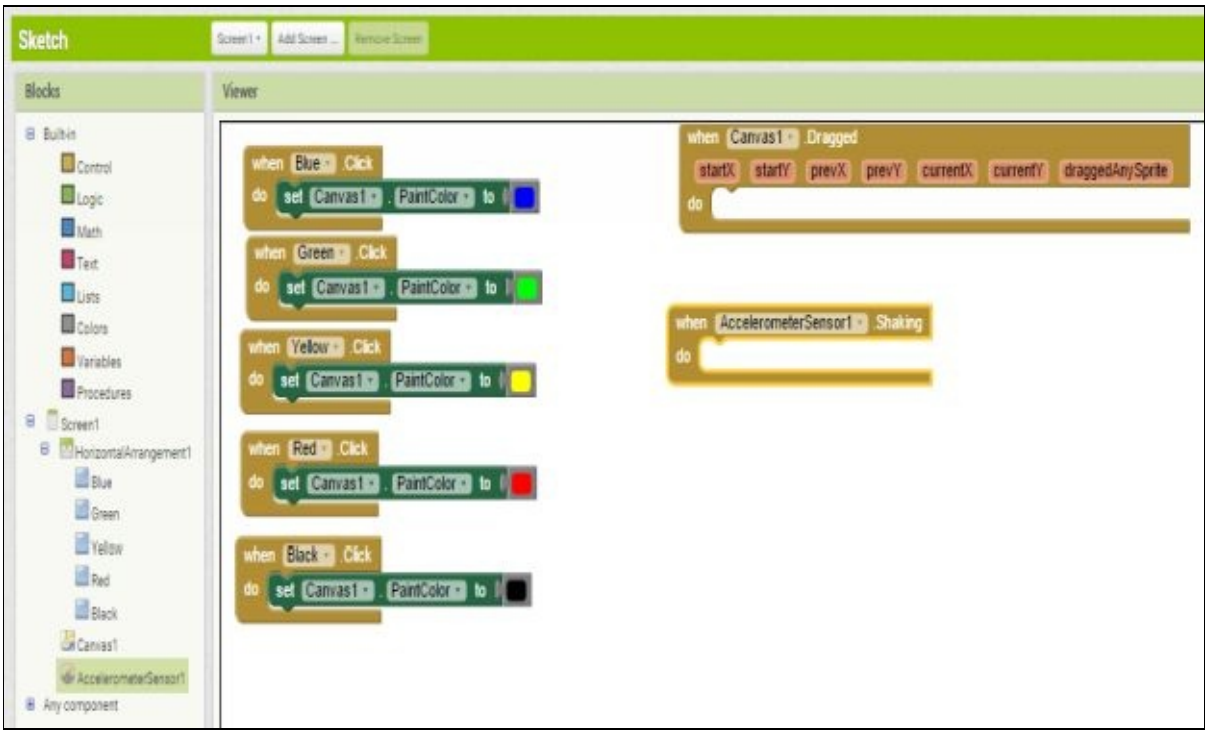




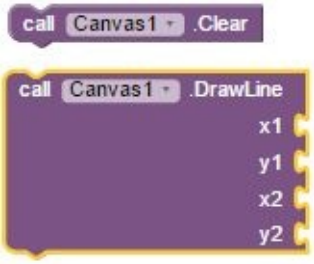
14. Attach the blocks as shown below. So when you click on Blue Button,PaintColor will be changed to Blue and similarly PaintColor will be change when you click on the other buttons.

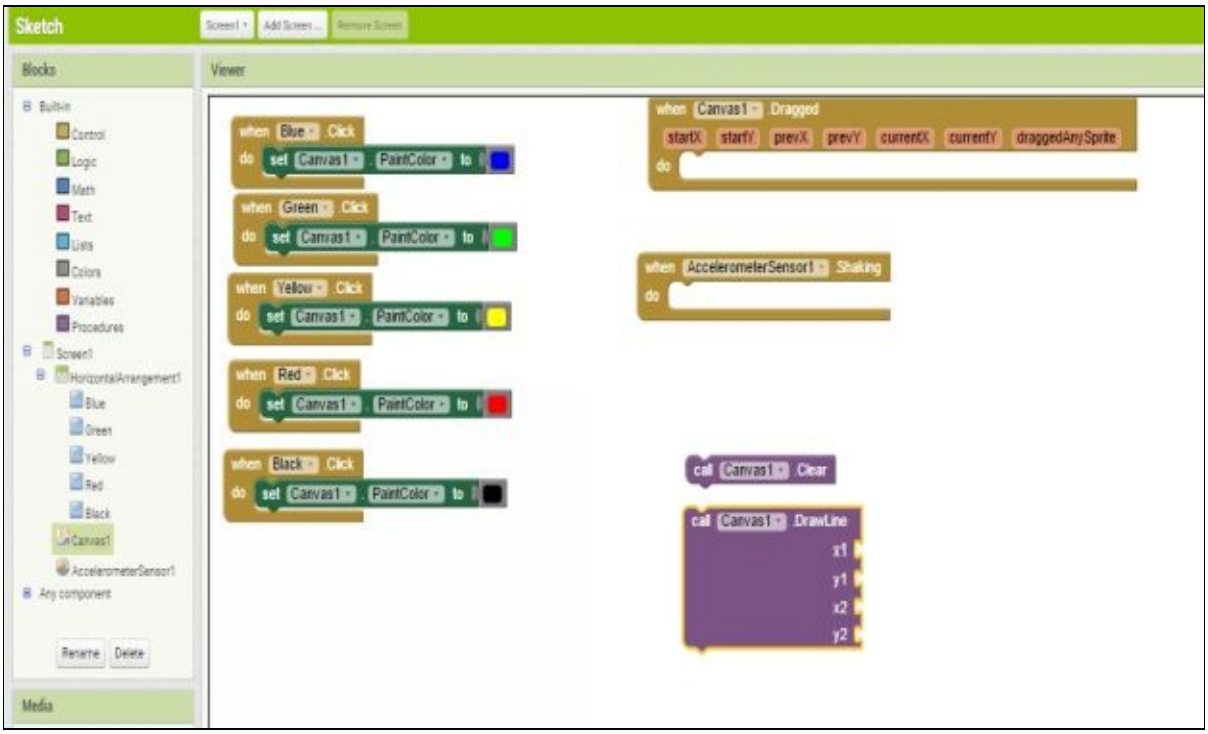


15. Click on AccelerometerSensor1 and select  block.

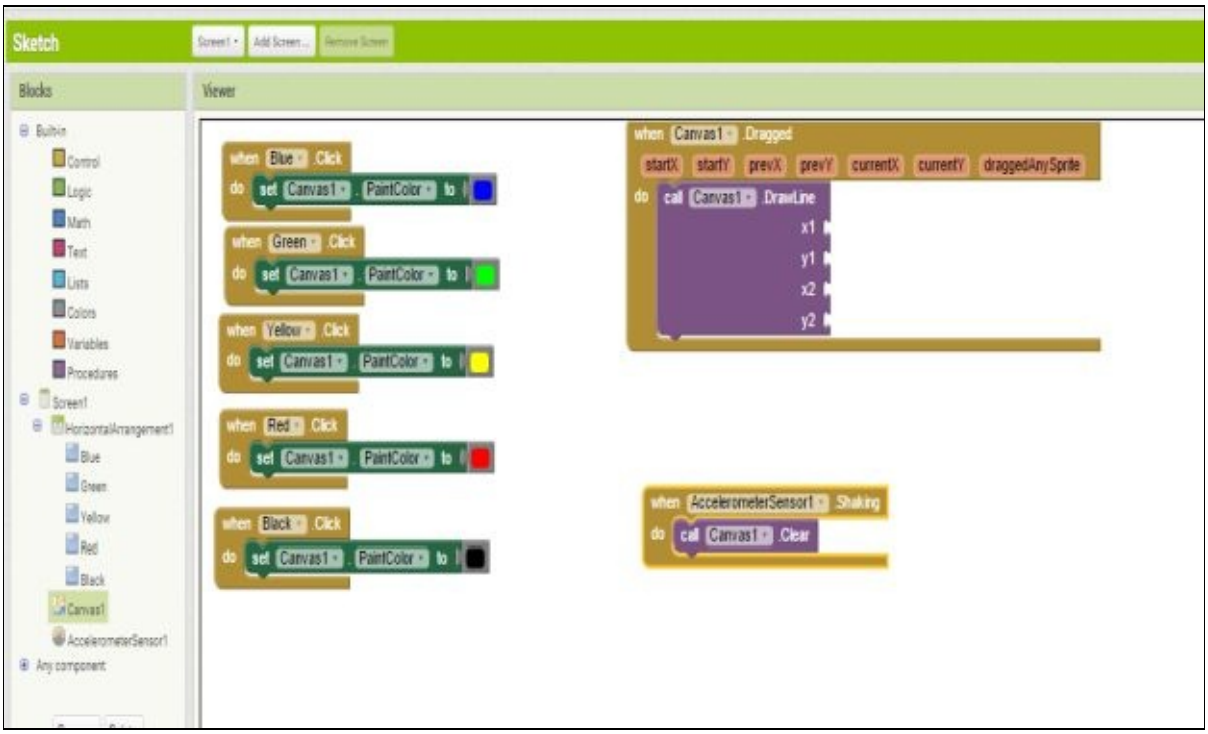


16. Now click on Canvas1 and select these two blocks.

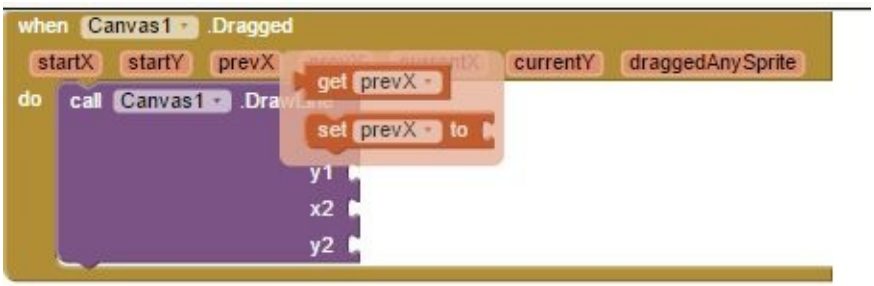


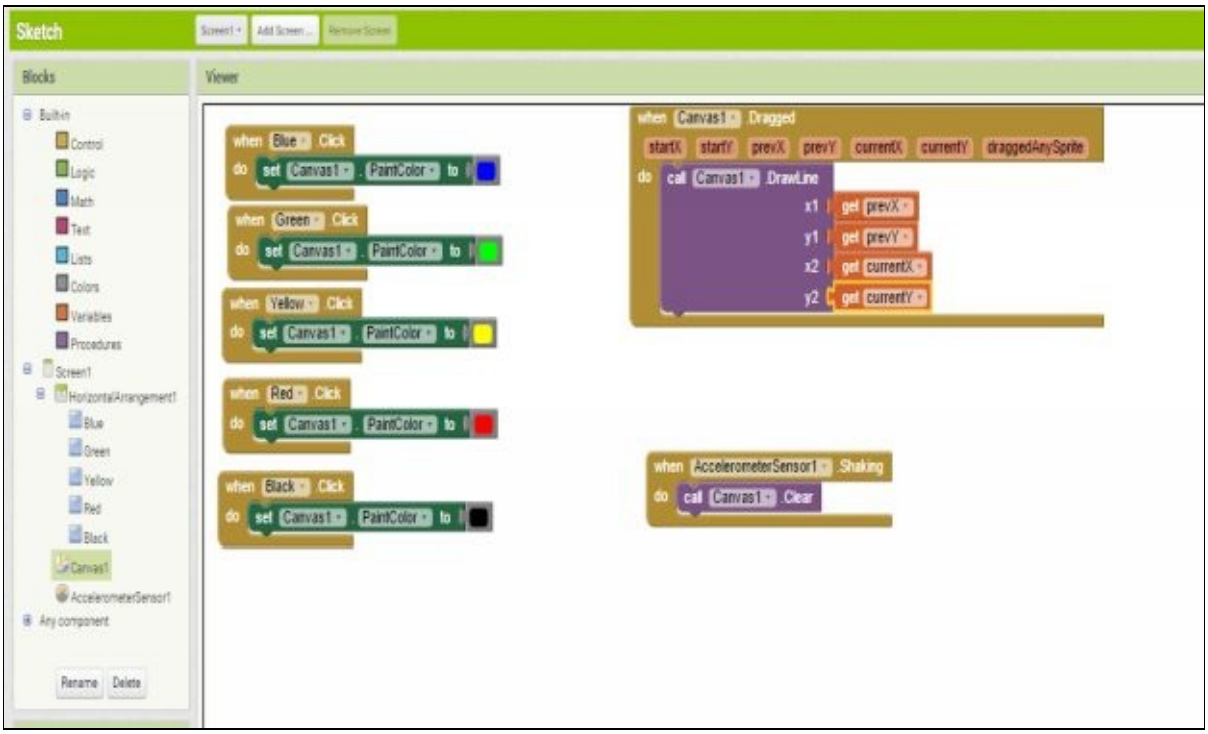


17. Attach the blocks as shown below. So that when you shake your phone app will clear your canvas.

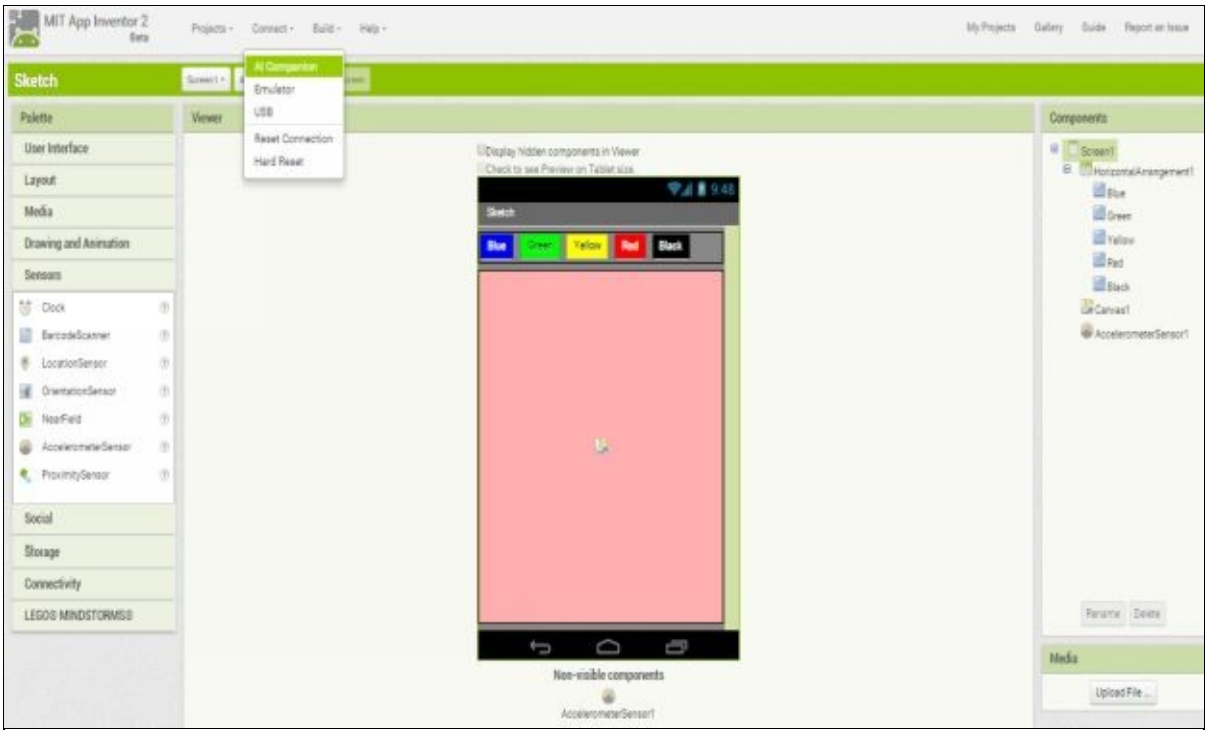


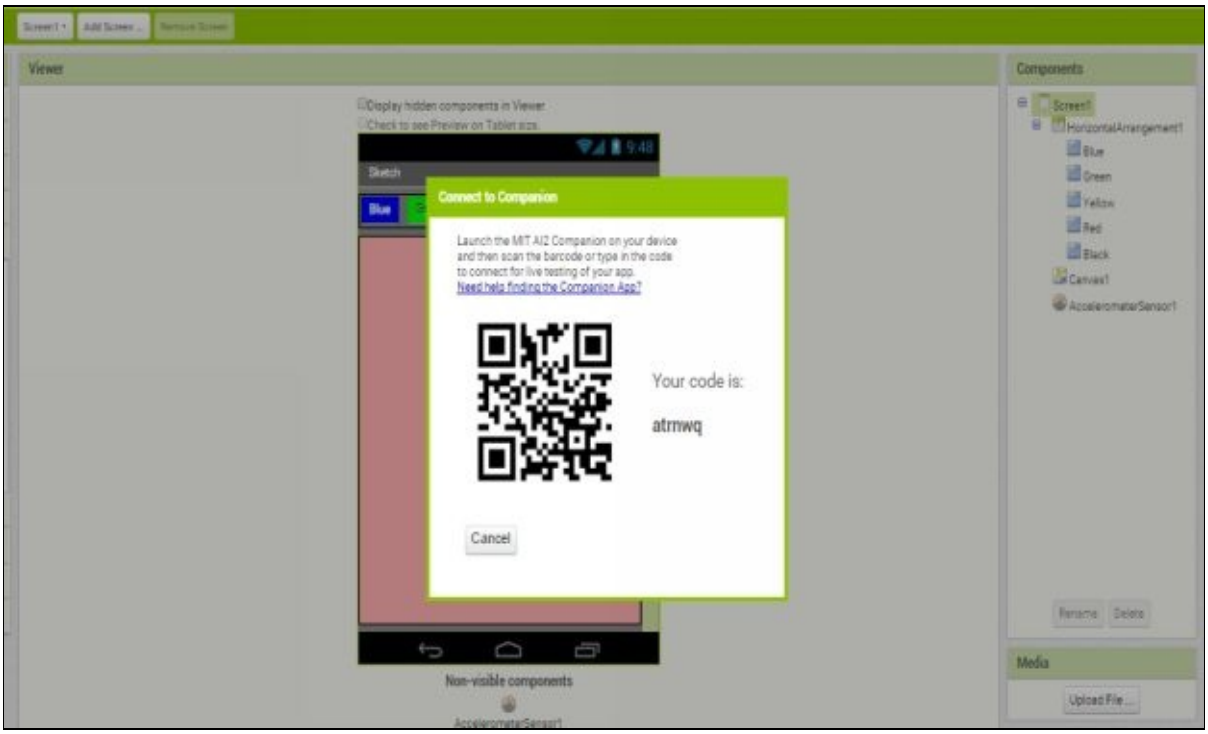
18. Now Mouse over on prevX, prevY, currentX and currentY to get these blocks  respectively and connect the blocks as shown below. These will represent start and end point of anything you draw on canvas.





19. Now go to Designer and click on Connect and then select AI Companion.

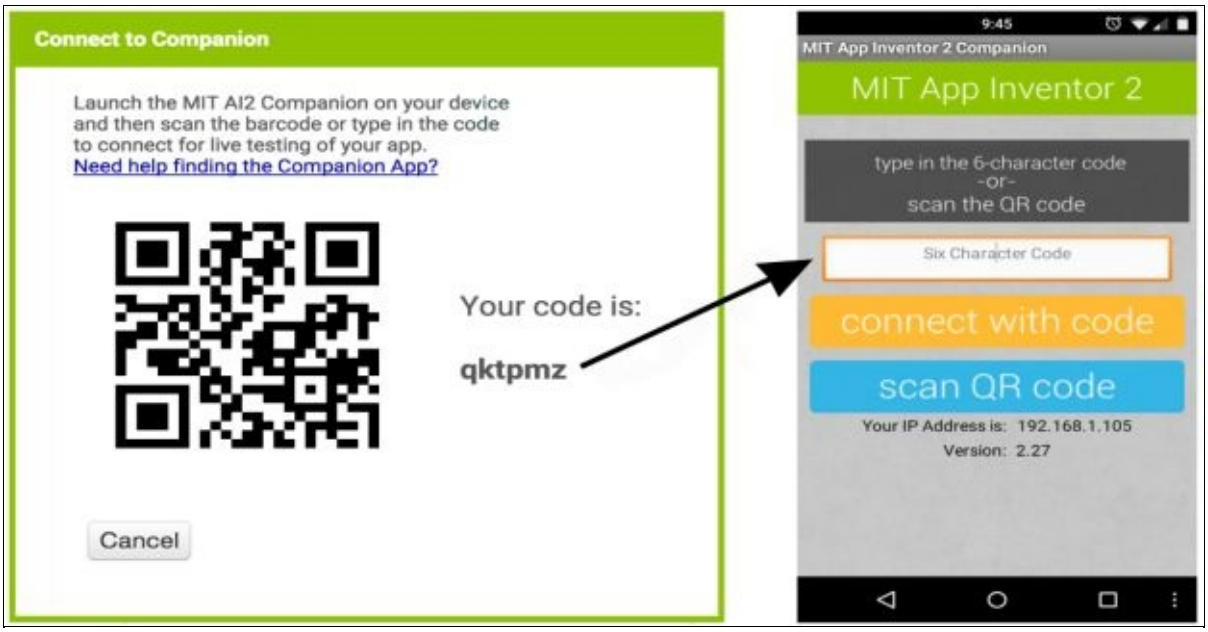




20. Now open MIT App Inventor 2 Companion in your mobile/Tablet.

And enter the code or scan the QR code from your mobile and click on connect with code.

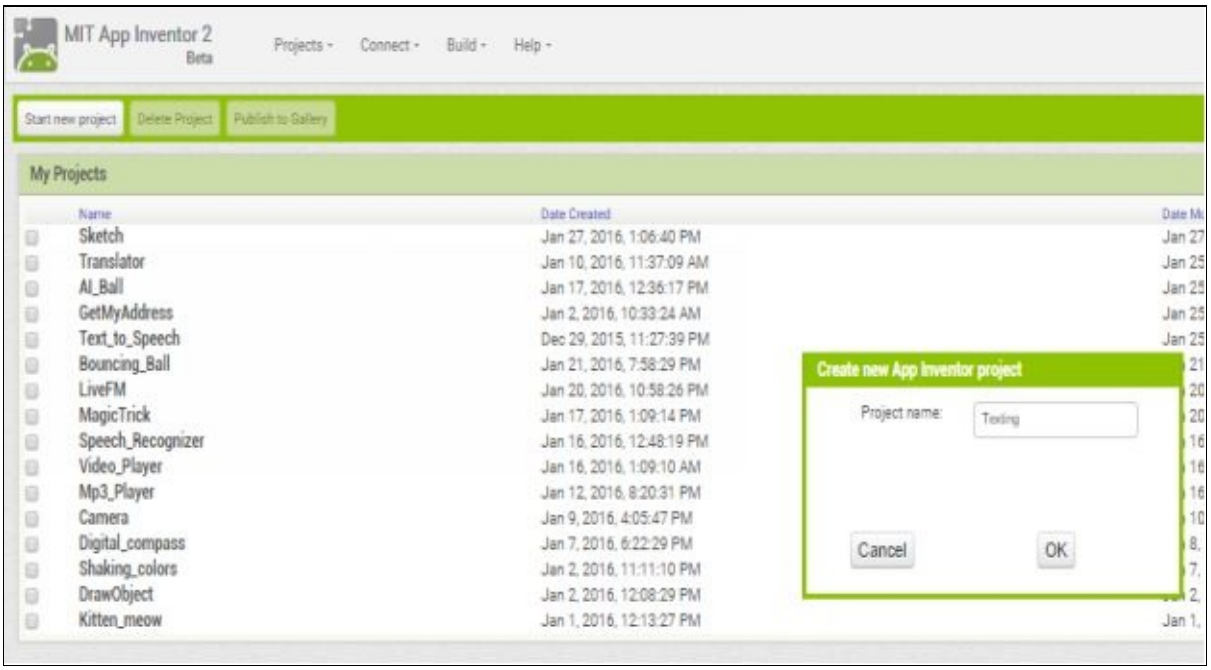
Note:-Your Mobile/Tablet should be connected to same wi-fi/network to which your computer/Laptop is connected.



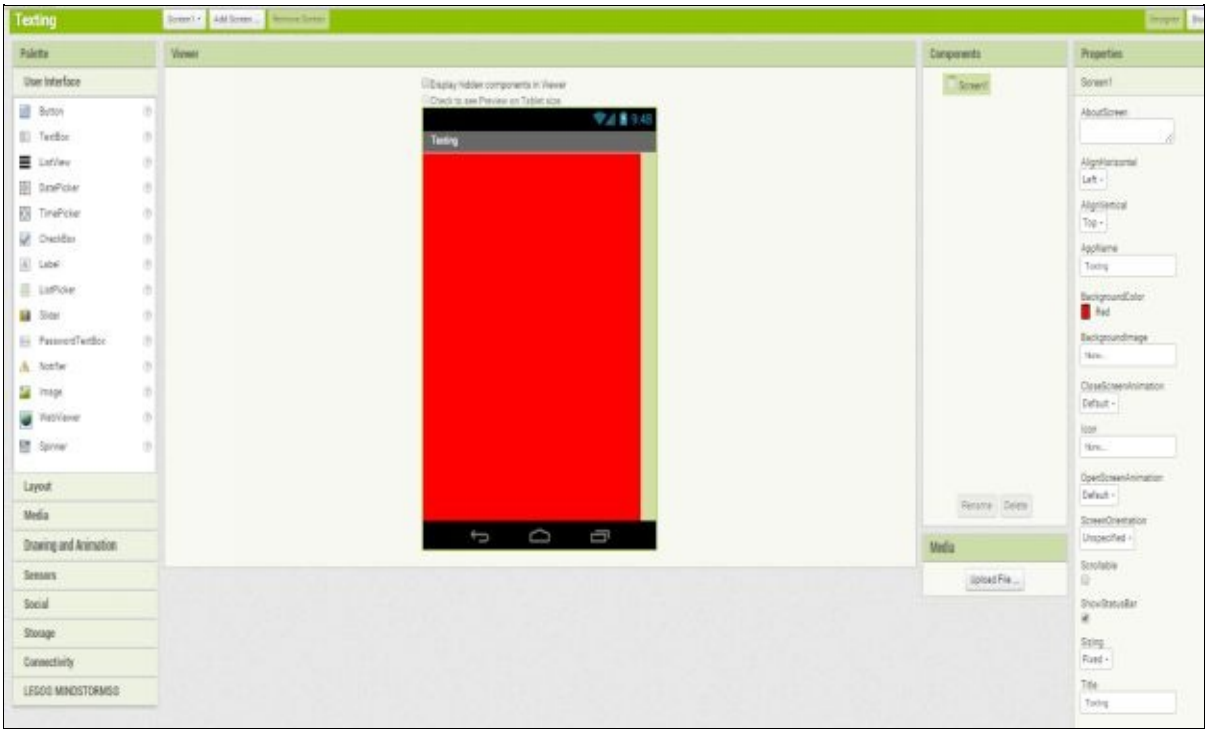
21. Now you should see your app in your phone for live testing. Click on any Color button to draw using that color and draw anything by dragging your finger on screen. To clear your sketch, just shake your phone.

Chapter 20 Texting-A messaging app

1. Click on Start new project and give it name Texting then click OK.

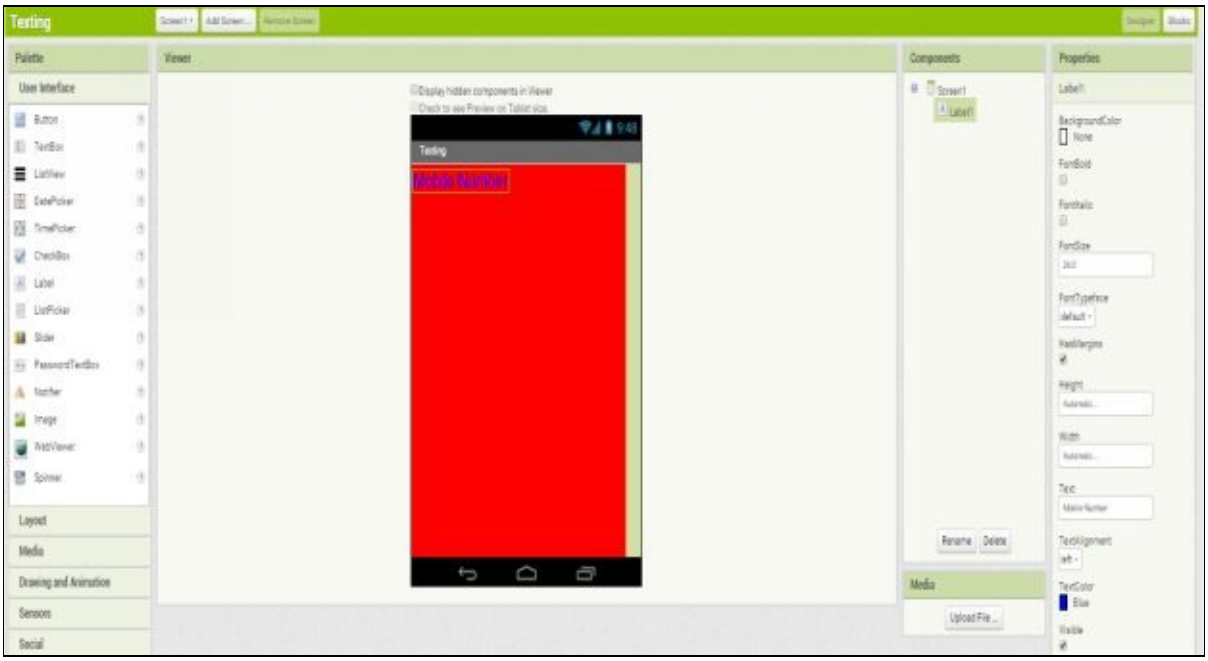


2. Change Screen1 Title to Texting and change BackgoundColor to Red.

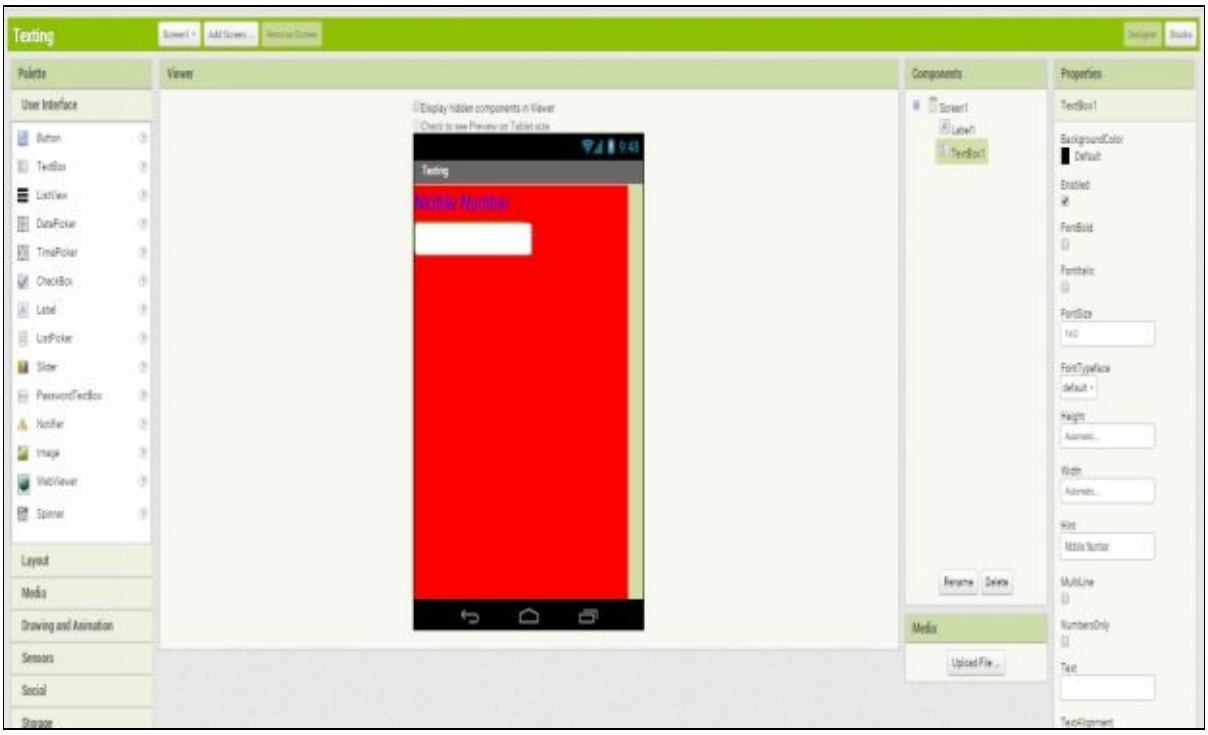


3. Drag a Label from Palette to Viewer screen and change its Text to Mobile Number.

Set its Properties as shown below.

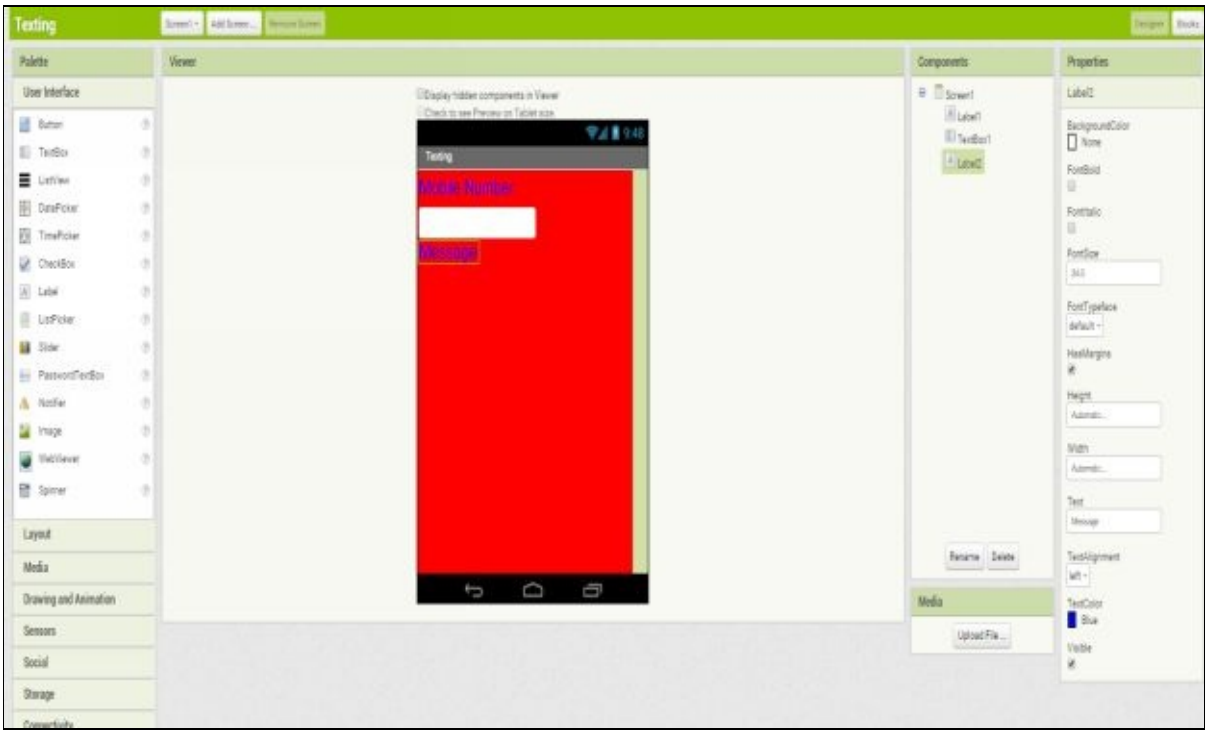


4. Drag a TextBox from Palette to Viewer screen and change its Hint Text to ‘Mobile Number’.

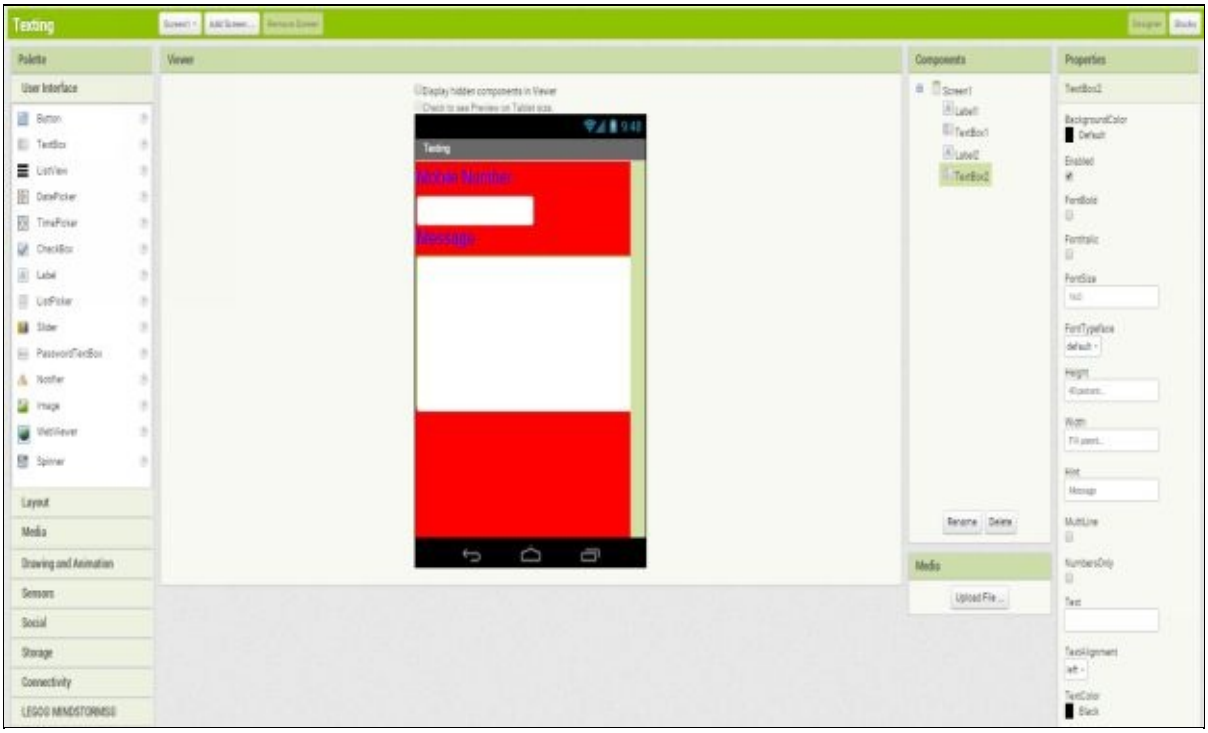


5. Drag a Label from Palette to Viewer screen and change its Text to ‘Message’.

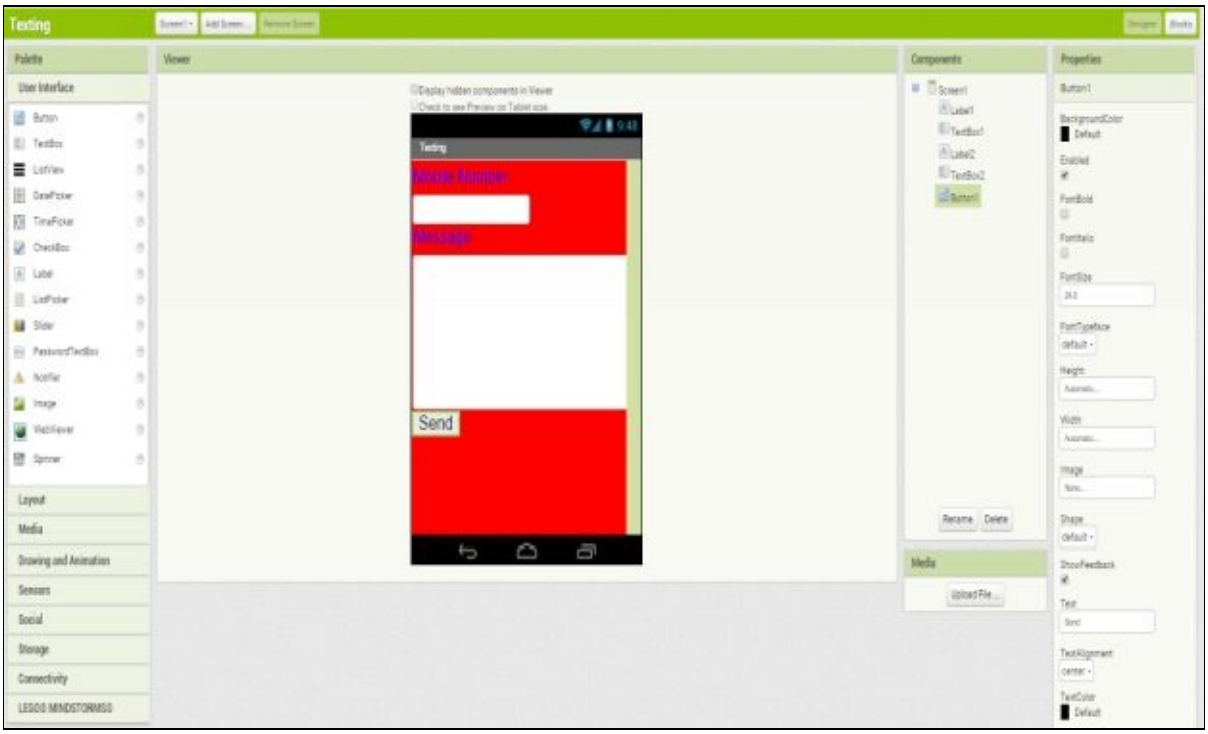
Set its Properties as shown below.



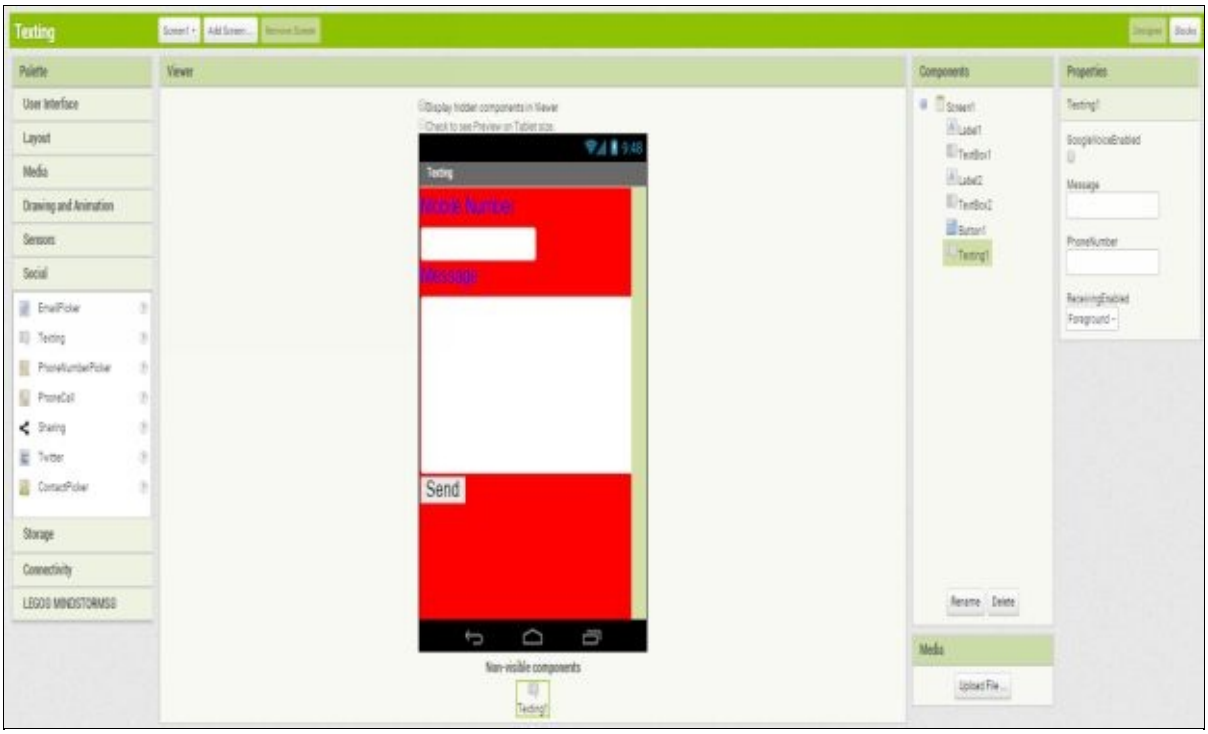
6. Drag a TextBox from Palette to Viewer screen and set its Height to 40 percent and Width to Fill parent. Change it's Hint Text to 'Message'.



7. Drag a Button from Palette to Viewer screen and change it's Text to Send and FontSize to 24.0.

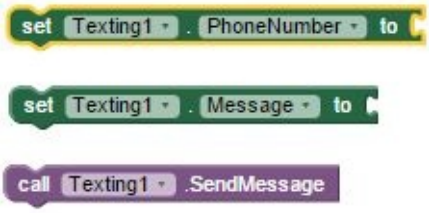


8. Drag a Texting (Non-visible) component from Palette to Viewer.

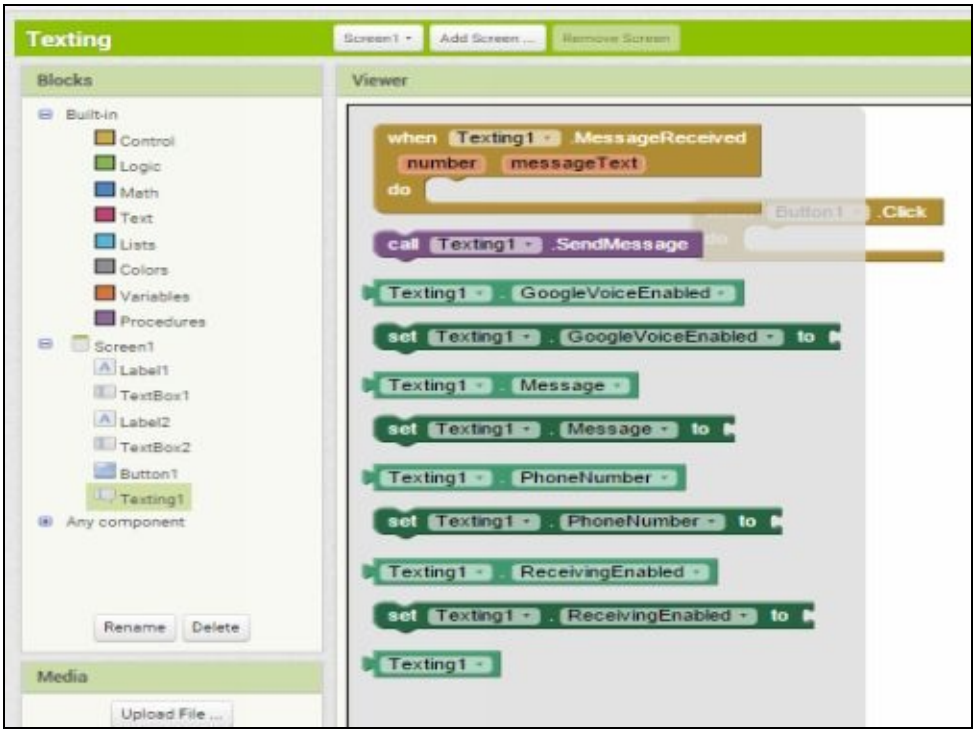


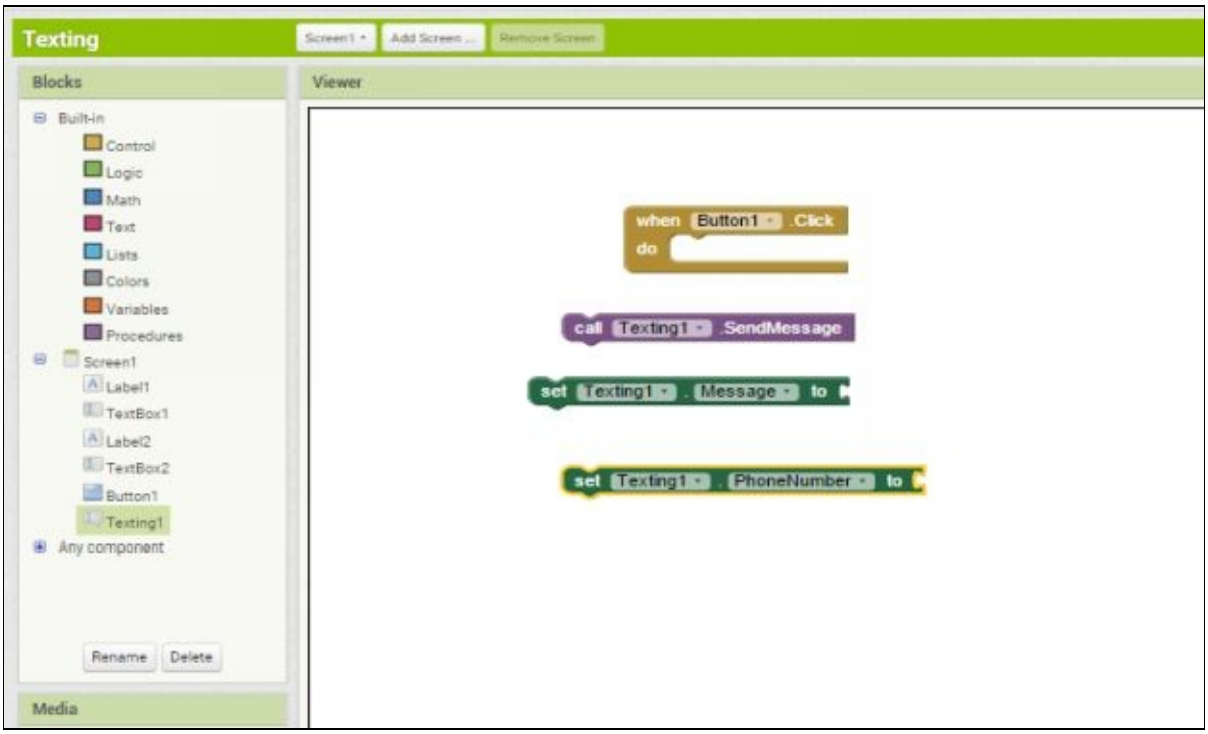
9. Go to Blocks and click on Button1 and select 

block. This is an event block which will decide what your app will do when this button will be clicked. Here click is an event for Button1 component.

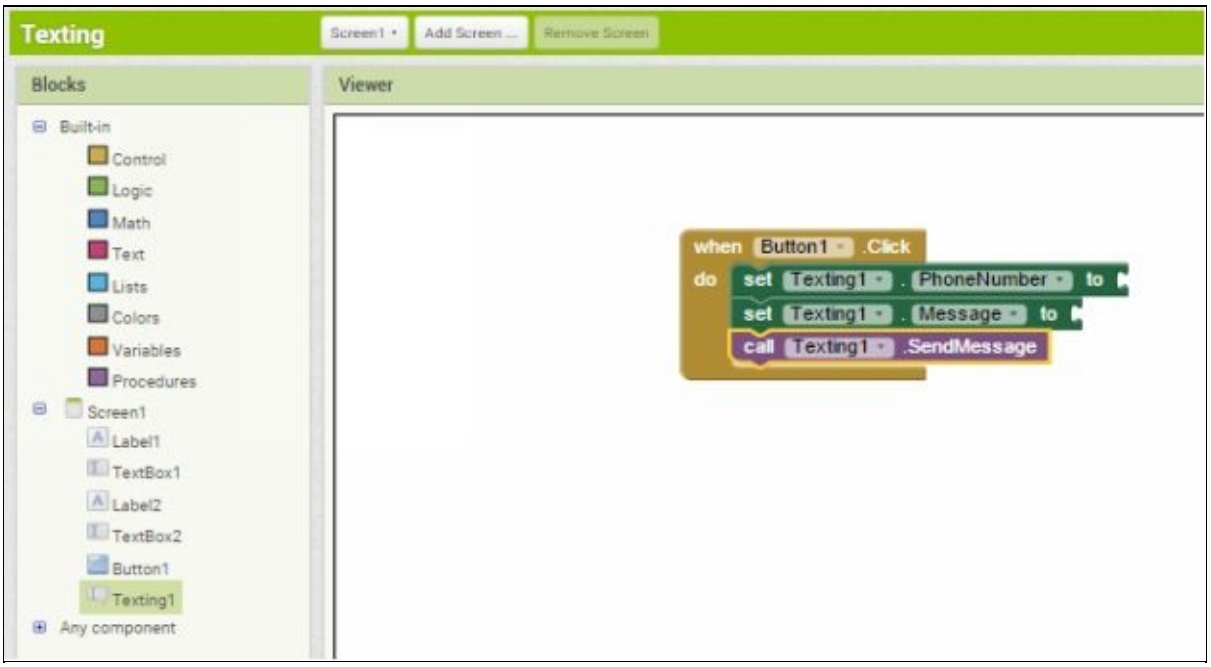


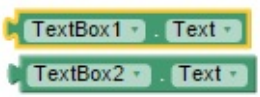
10. Click on Texting1 and select blocks. Texting component will use your Phone's messaging and send a message to a PhoneNumber. These blocks will set PhoneNumber and Message to be sent.



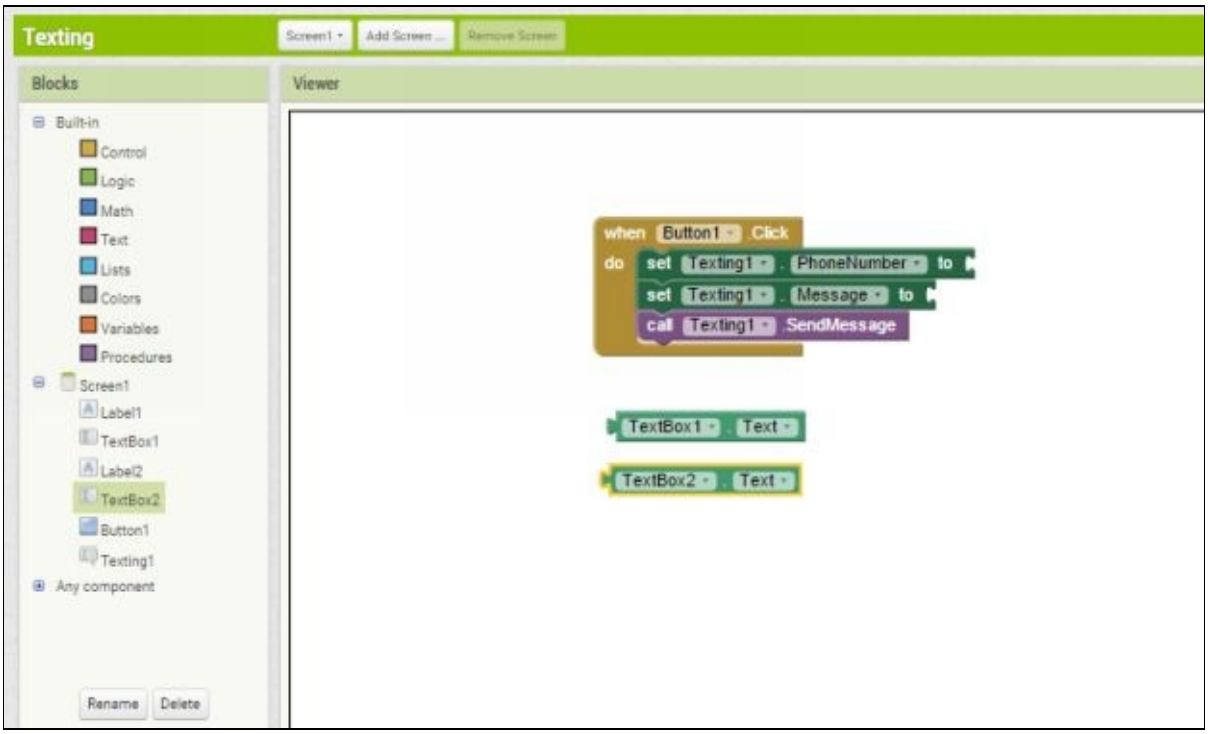


11. Attach the blocks as shown below. So when you click Button1 it will set Phone number and message and then call Texting1.SendMessage to send Message.

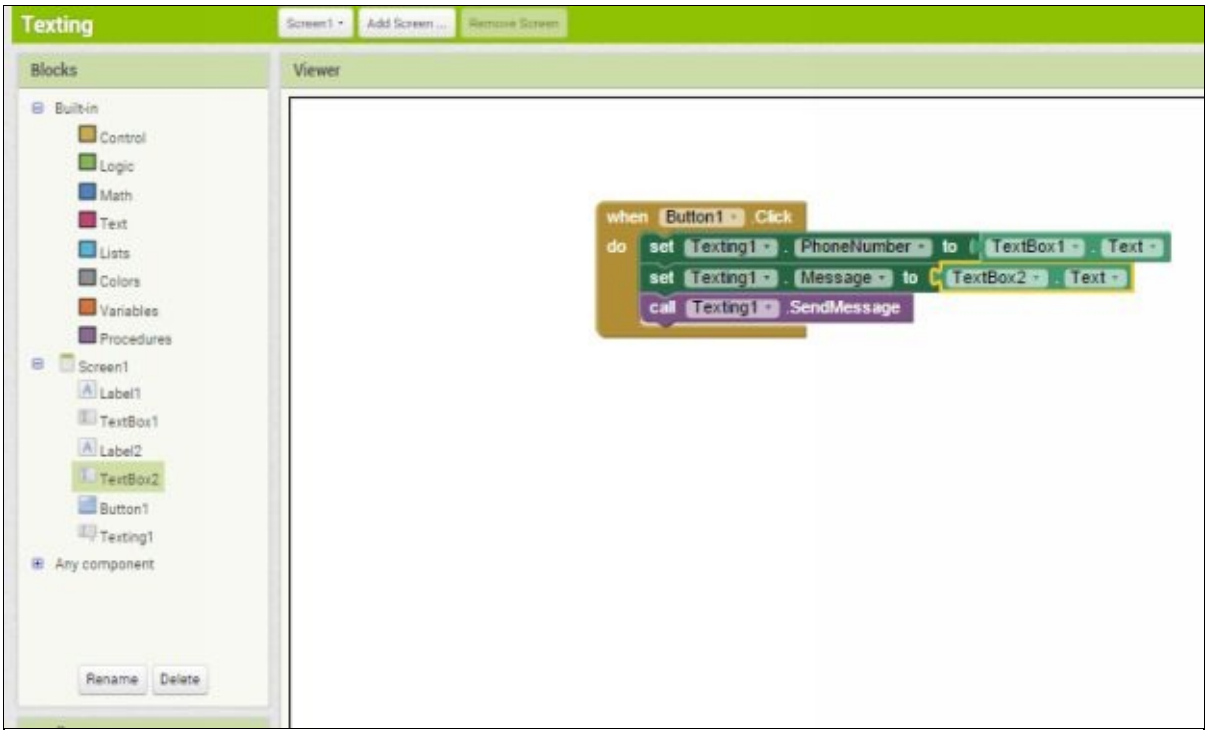


12. Click on TextBox1 and TextBox2 and scroll down to select 

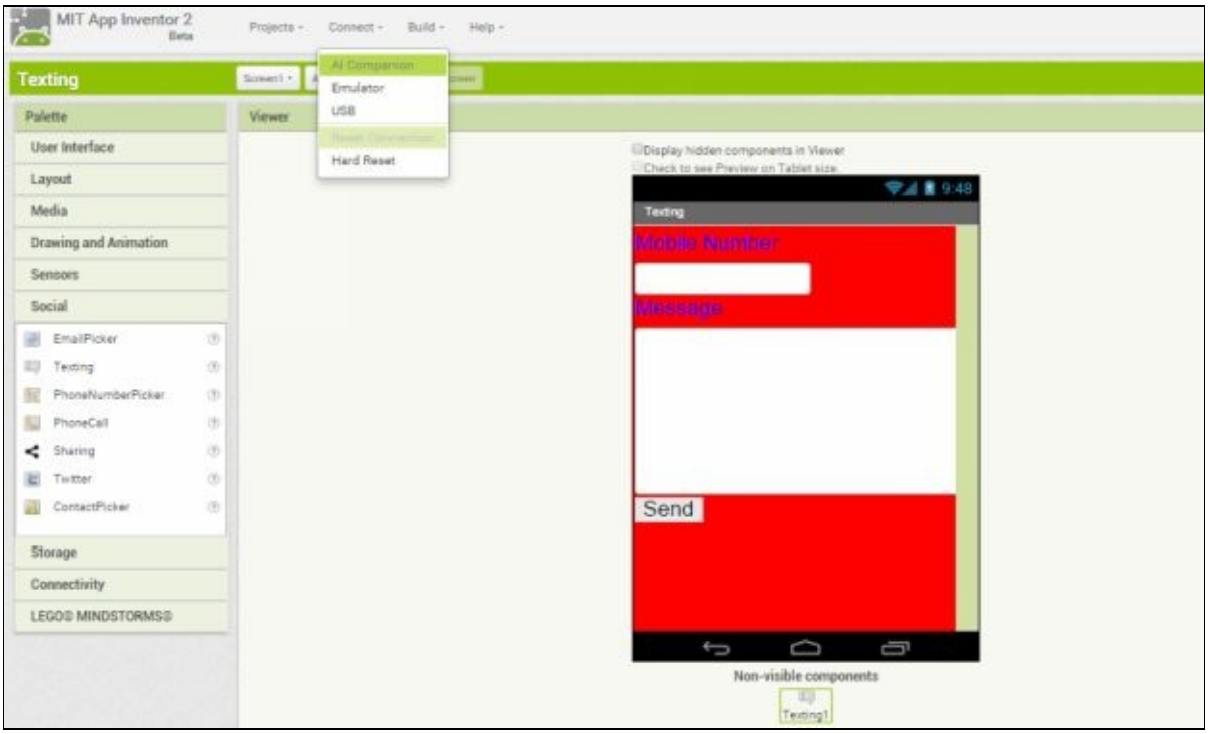
blocks. Here TextBox1.Text represents Phone number and TextBox2.Text represents Message.

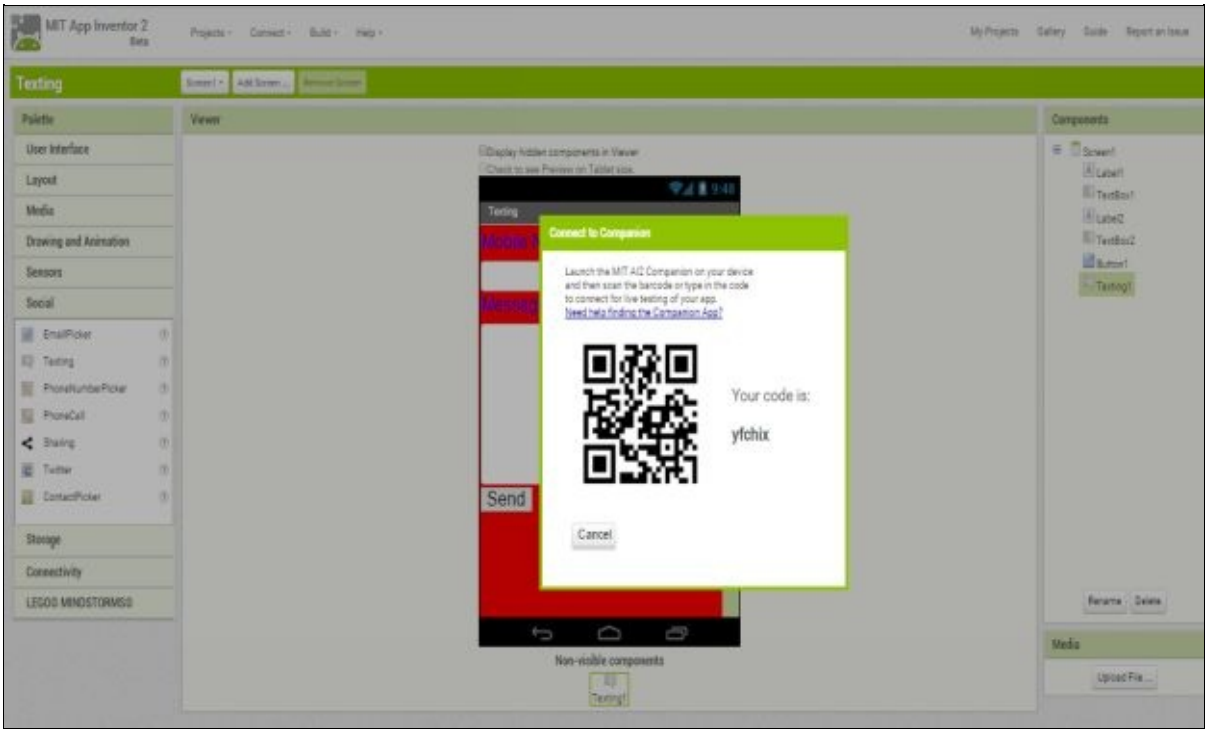


13. Attach the blocks as shown below.



14. Go to Designer and click on Connect and then select AI Companion.

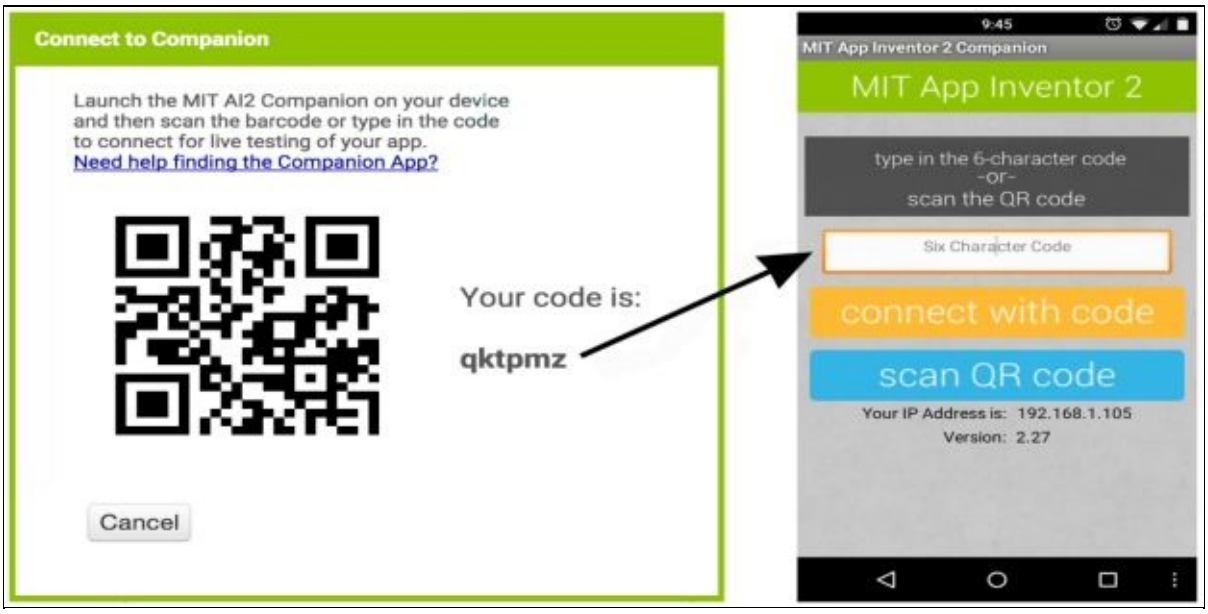




15. Now open MIT App Inventor 2 Companion in your mobile/Tablet.

And enter the code or scan the QR code from your mobile and click on connect with code.

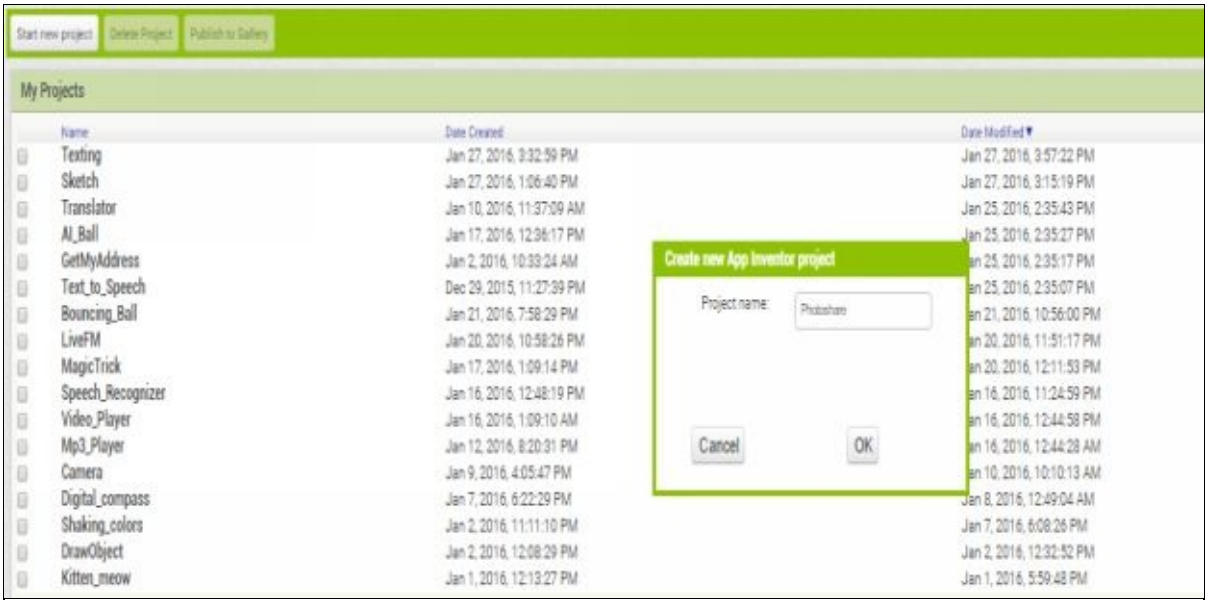
Note:-Your Mobile/Tablet should be connected to same wi-fi/network to which your computer/Laptop is connected.



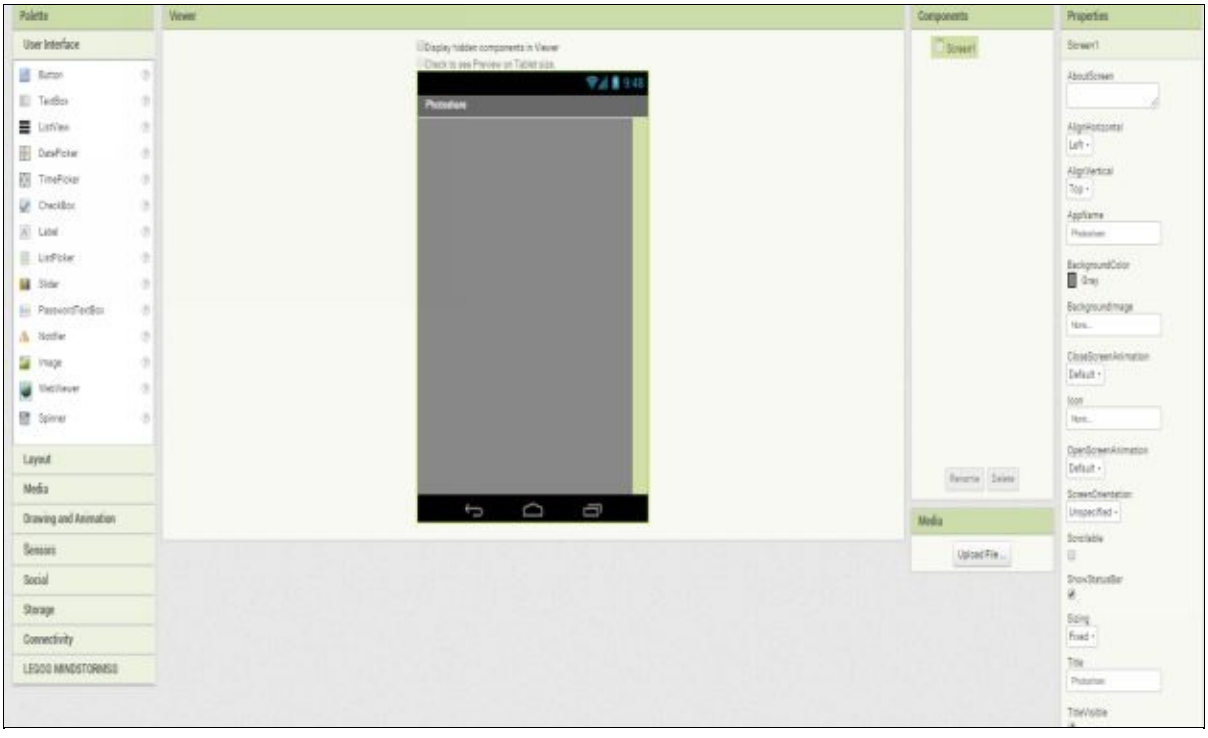
16. Now you should see your app in your phone for live testing. Enter Mobile number and Message and click Send button to send message.

Chapter 21 Photo share app

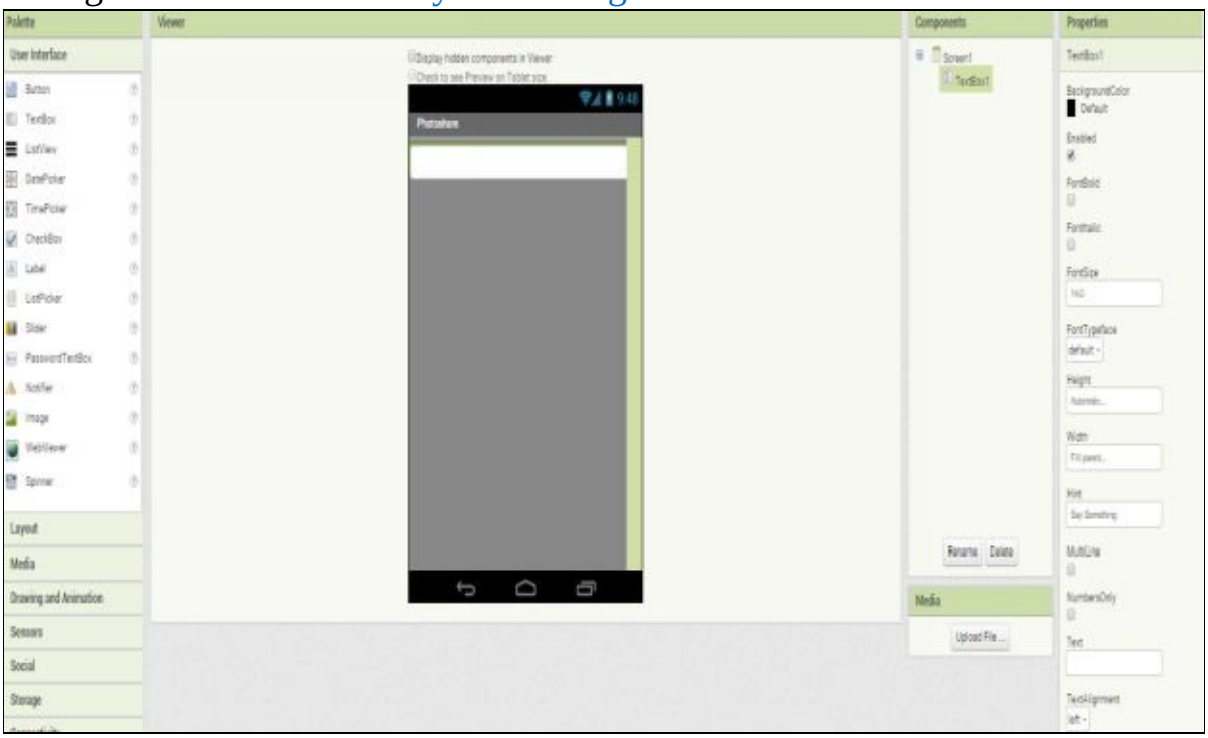
1. Click on Start new project and give it name Photoshare then click OK.



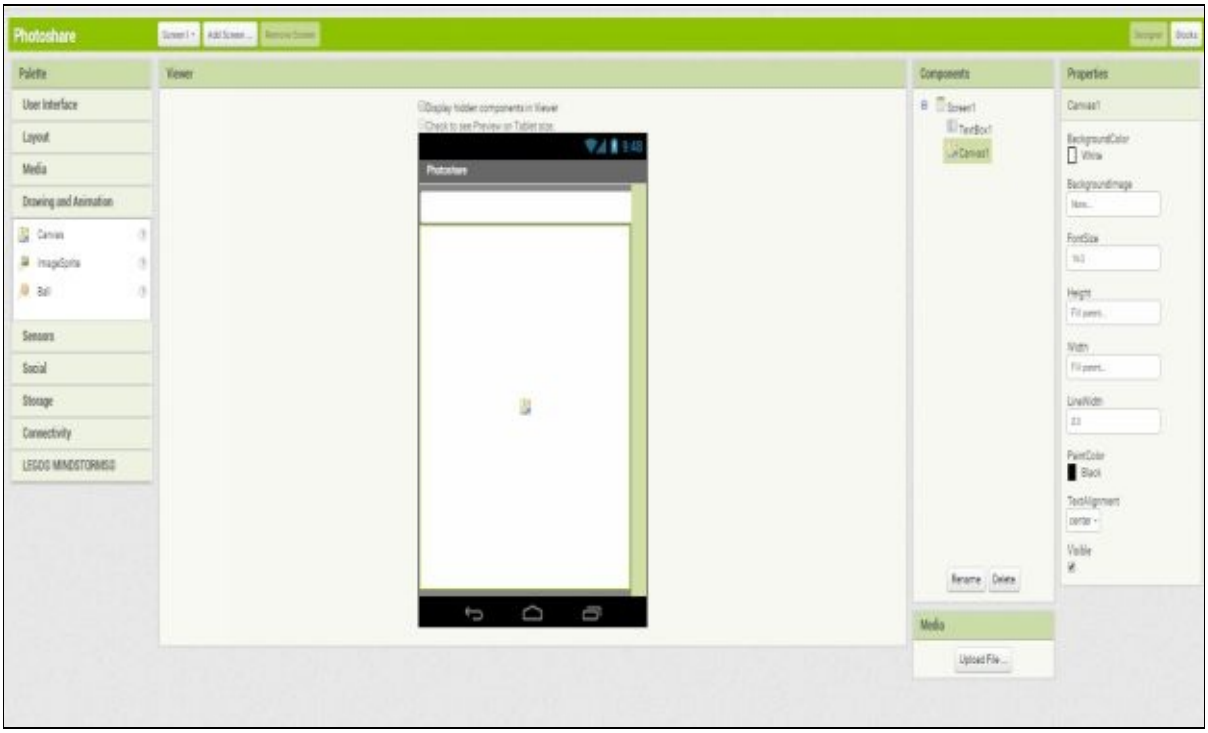
2. Change Screen1 Title to Photoshare and change BackgroundColor to Gray.



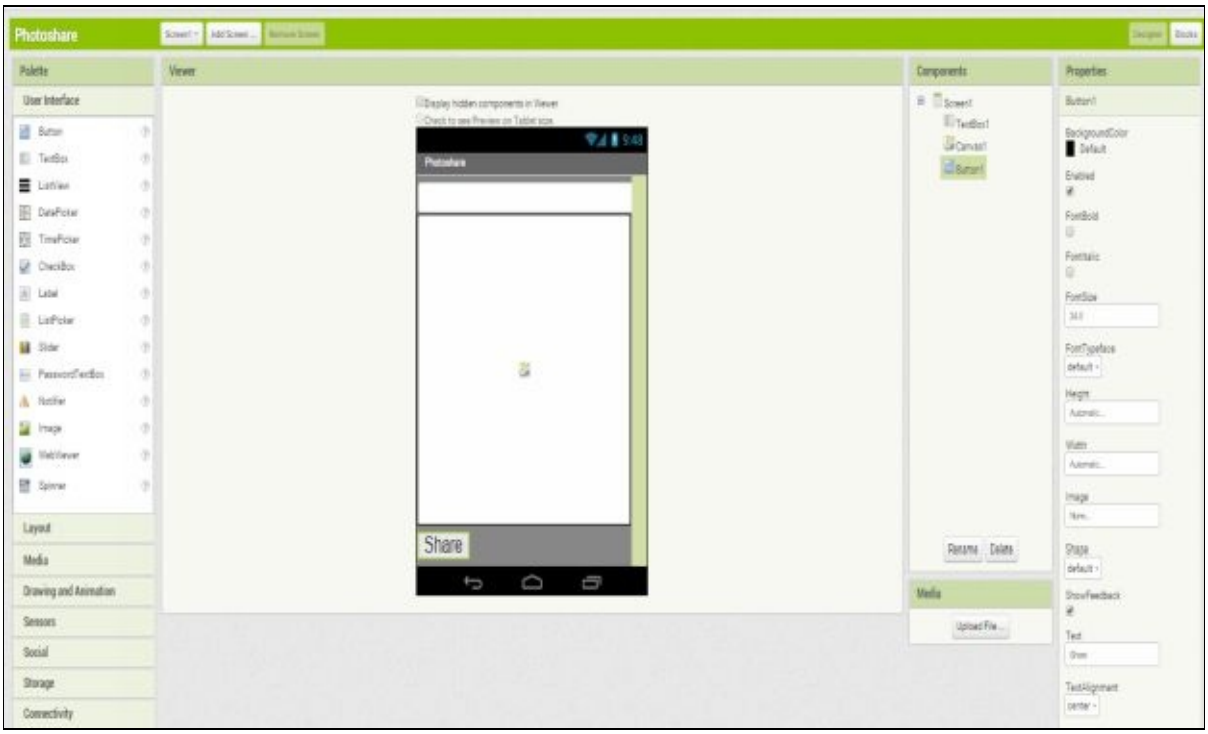
3. Drag a TextBox from Palette to Viewer screen and set it's Width to Fill parent and Change it's Hint Text to 'Say Something'.



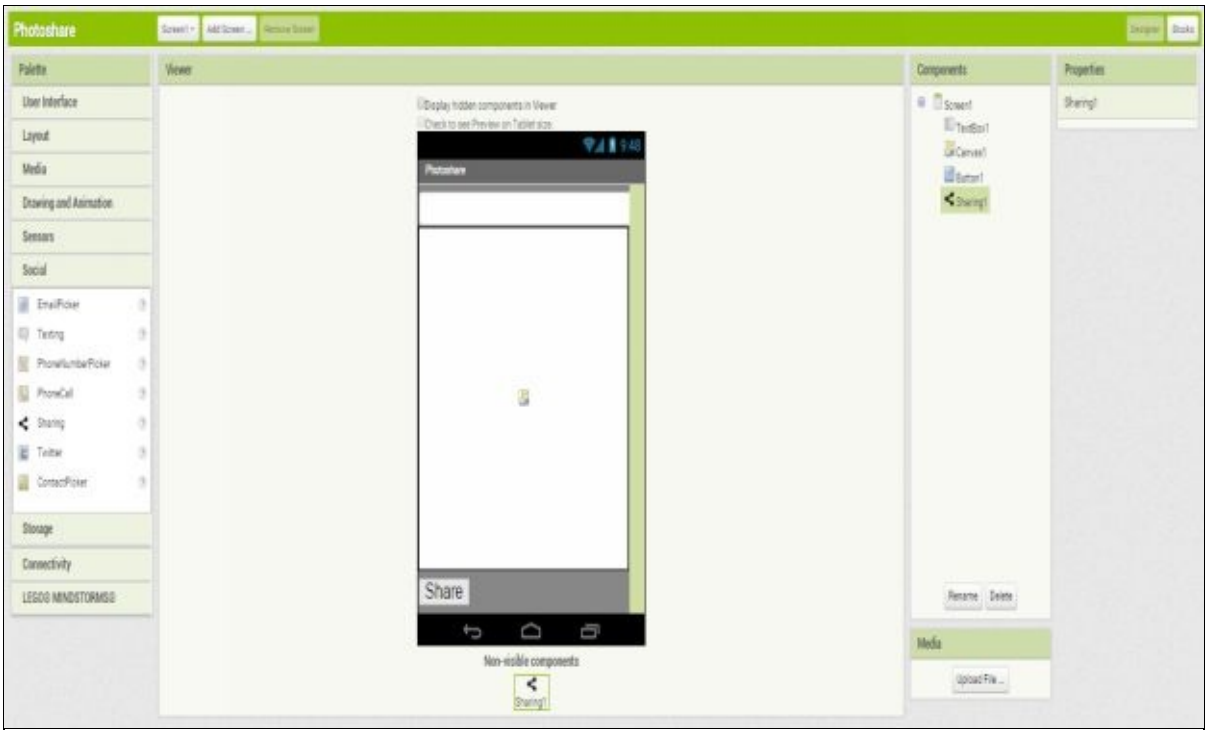
4. Now Drag a Canvas from Palette to Viewer screen and set it's Height and Width to Fill parent.



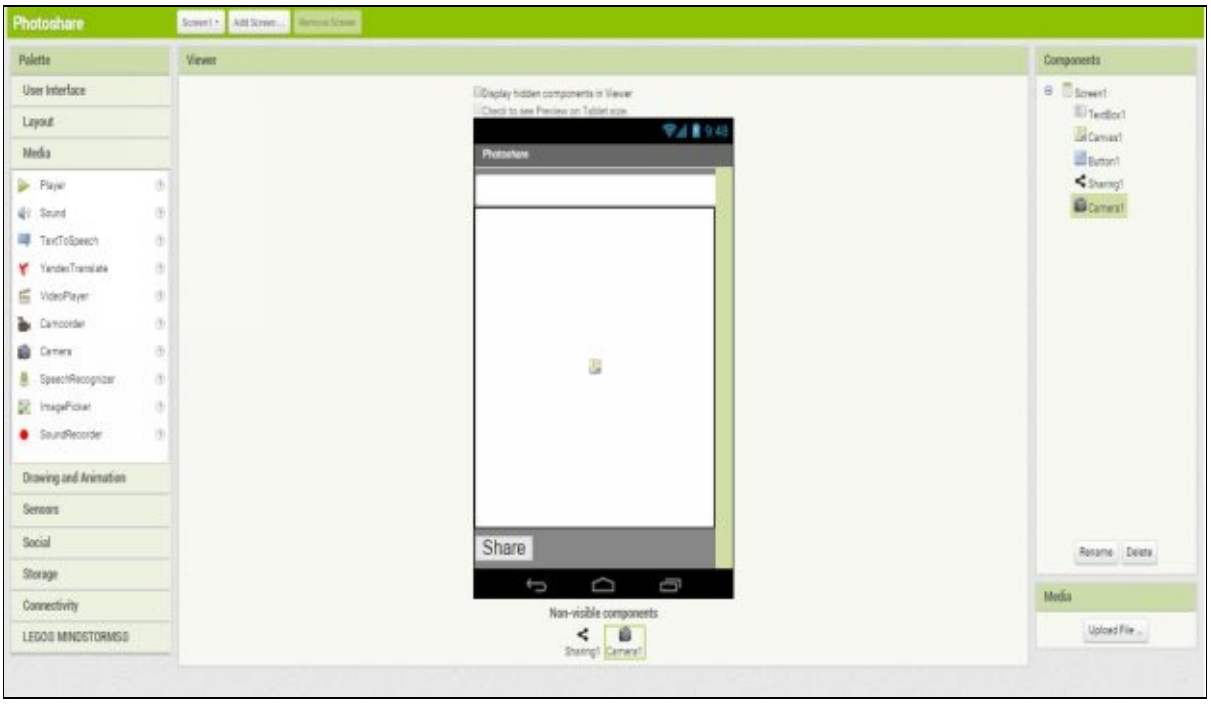
5. Now Drag a Button from Palette to Viewer and change it's Text to 'Share'. Change it's FontSize to 24.0.



6. Drag a Sharing (Non-visible) component from Palette to Viewer.

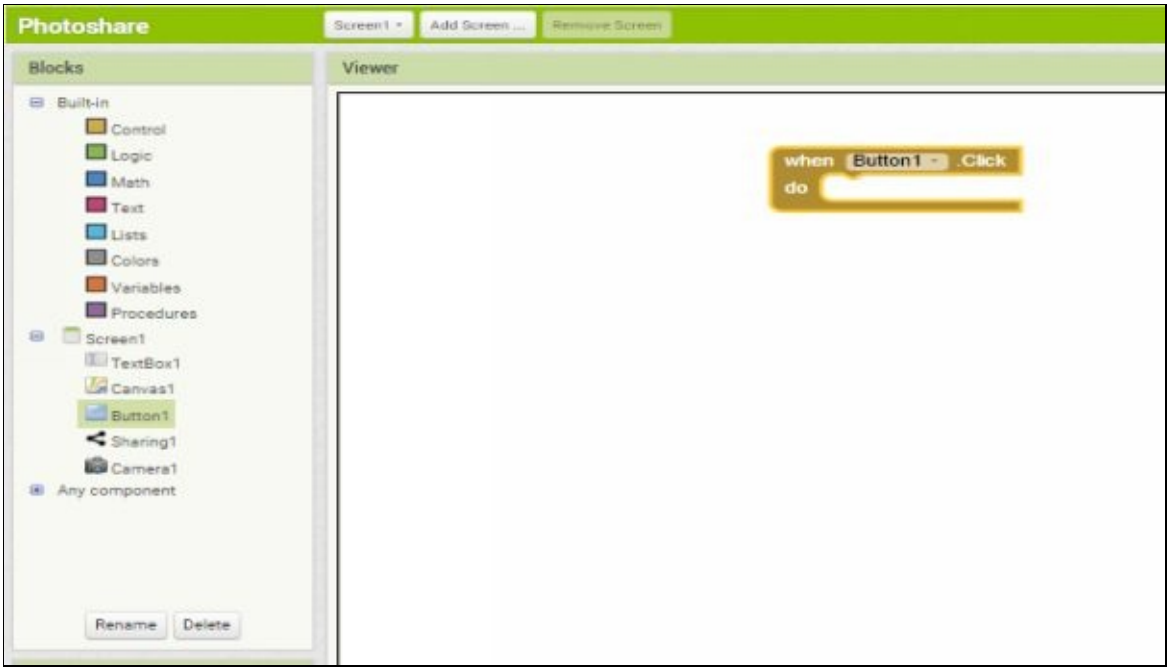


7. Drag a Camera (Non-visible) component from Palette to Viewer.



8. Go to Blocks and click on Button1 and select 

block. This is an event block which will decide what your app will do when this button will be clicked. Here click is an event for Button1 component.

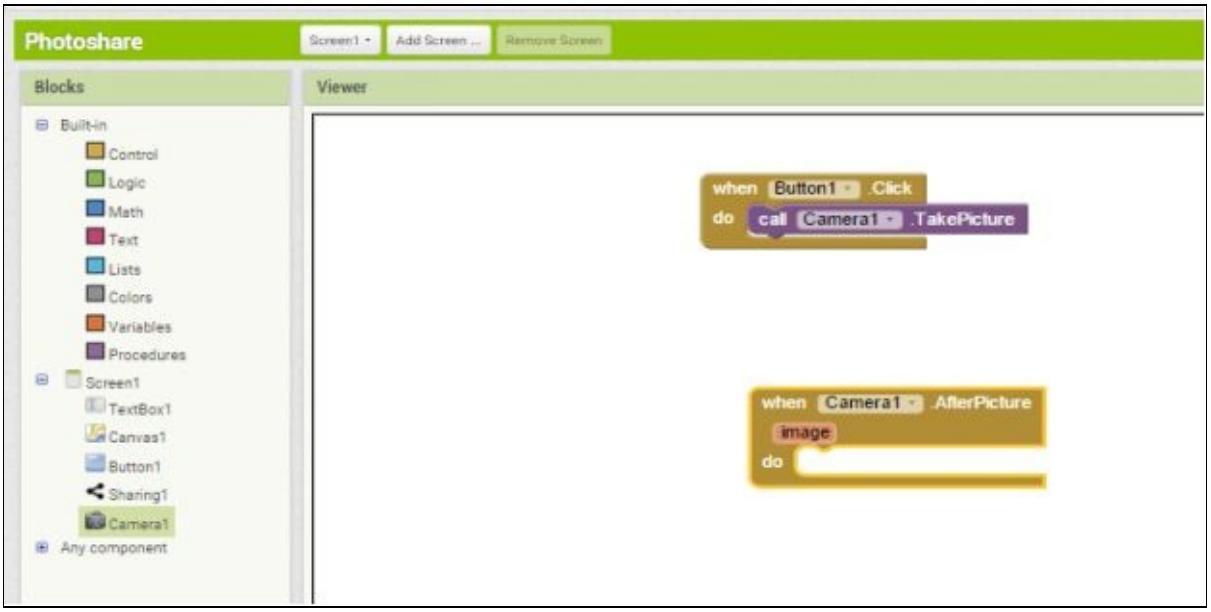


9. Click on Camera1 and select 

block. When you click Button1 ,app will open your device's camera to take picture.

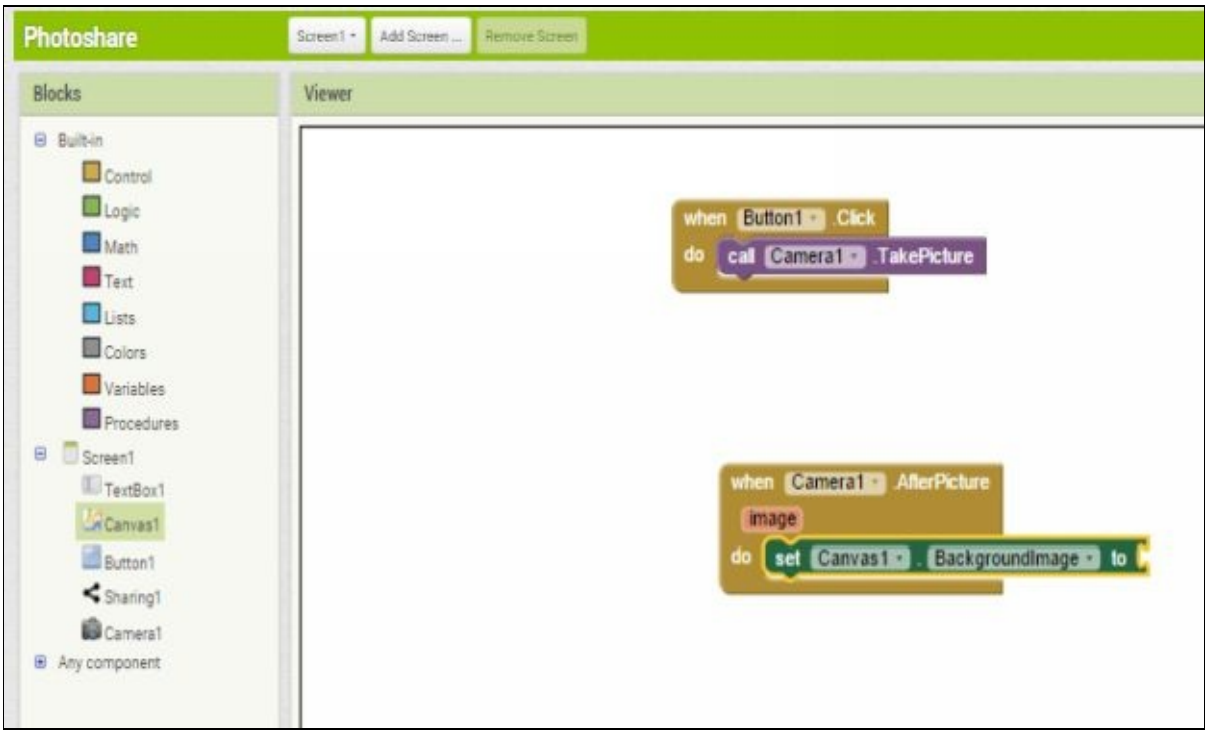
10. Click on Camera1 and select 

block. This block will decide what to do after taking picture.

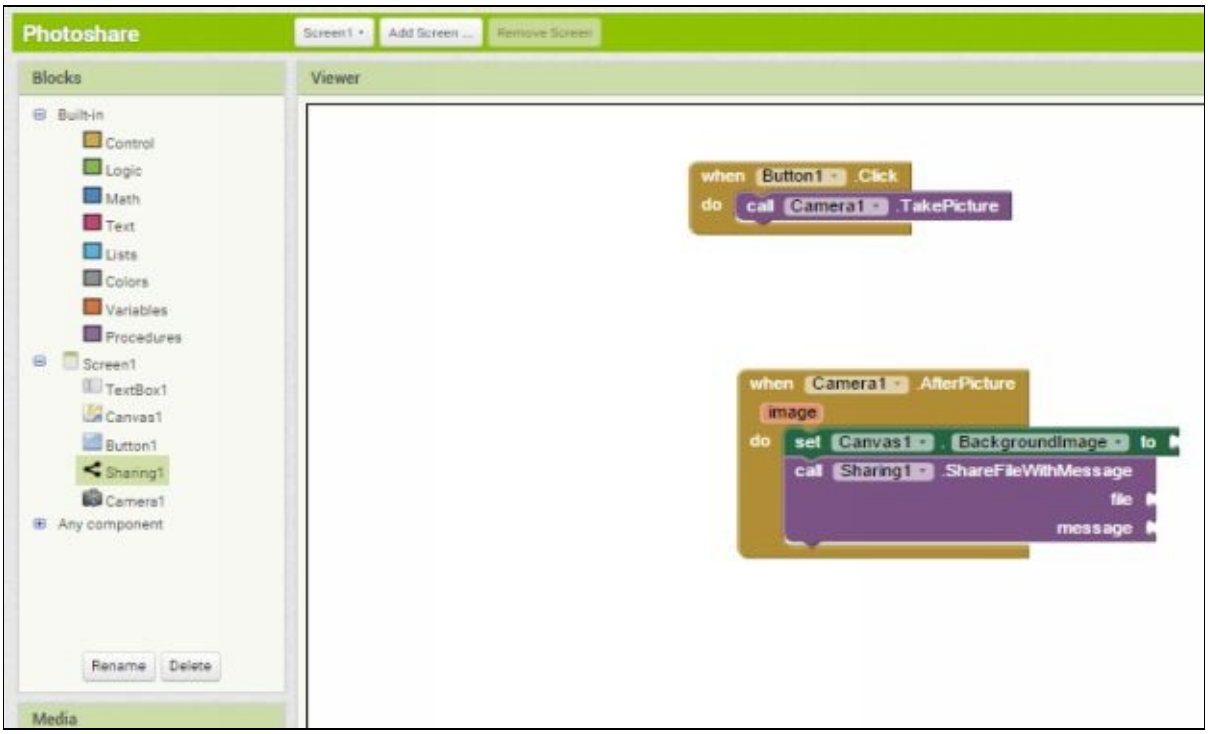


11. Click on Canvas1 and select 

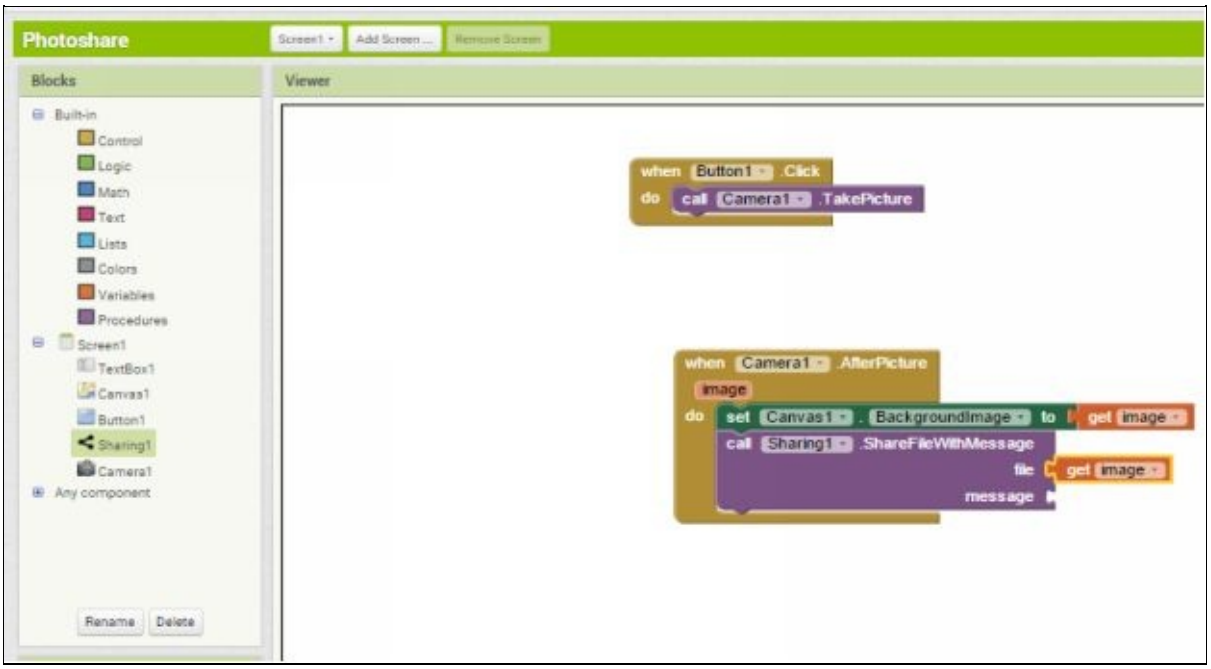
block. This block will set the Canvas BackgroundImage.

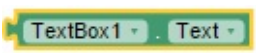


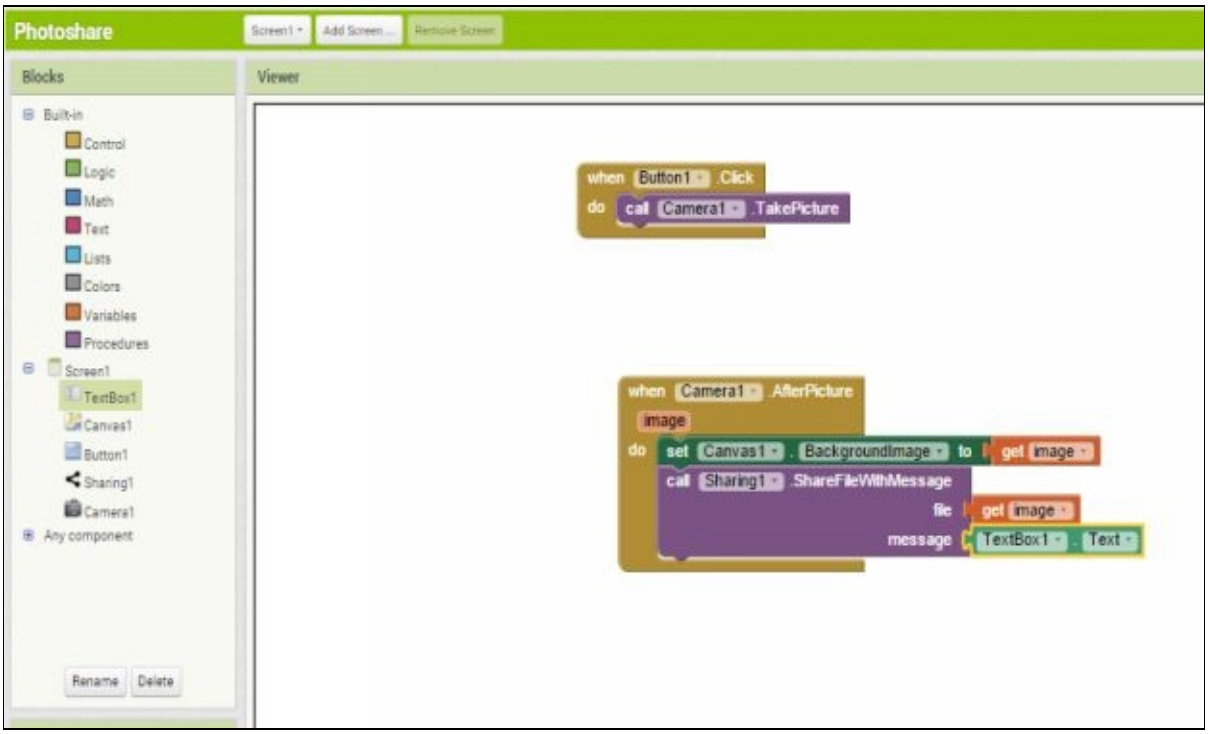
12. Click on Sharing1 and select  block. Sharing component will open all the sharing apps from your phone through which you can share a message or file or Both.



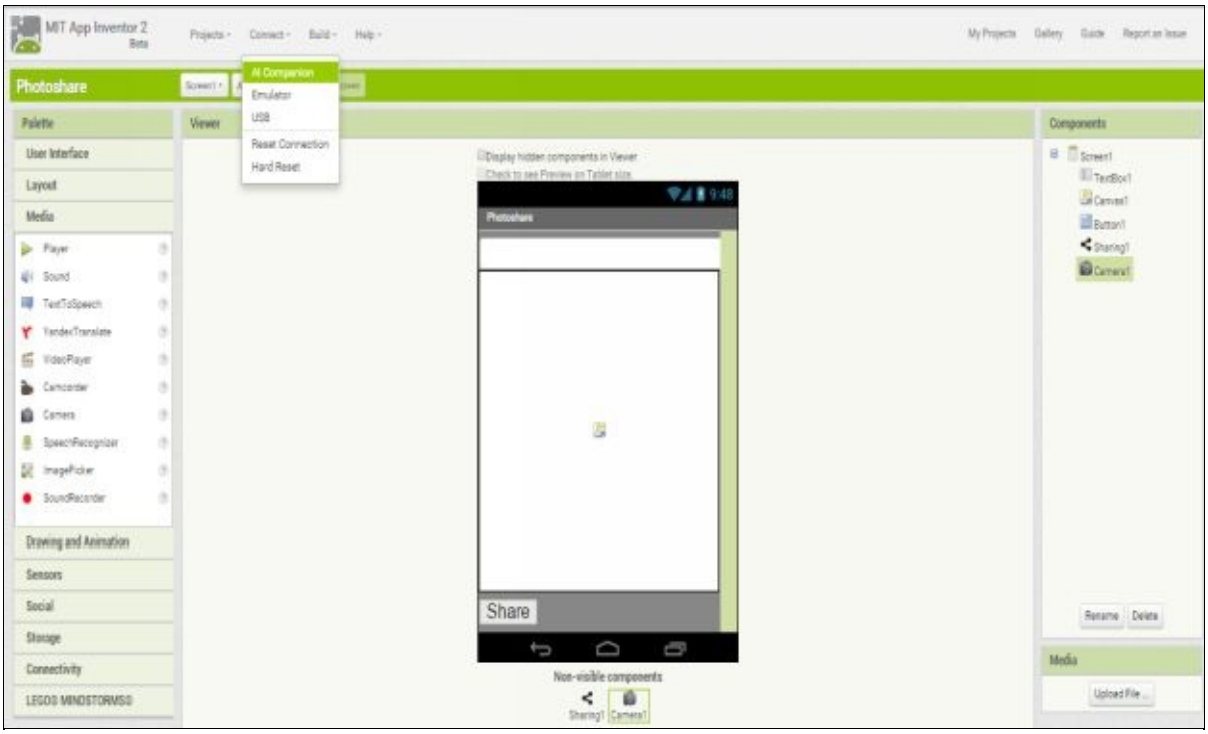
13. Mouse over on image and select  block. This block will return the image taken by Camera.



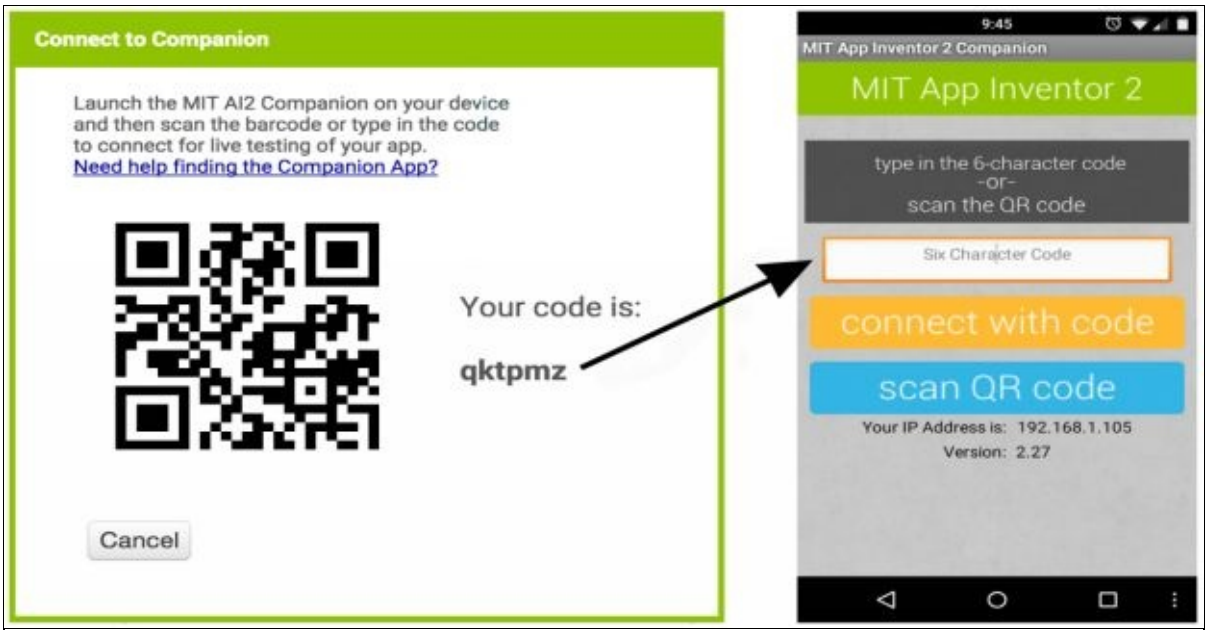
14. Click on TextBox1 and scroll down to select  block. So after taking picture Canvas BackgroundImage will be changed to the picture taken by camera and app will show you a window to choose app to share your app and message.



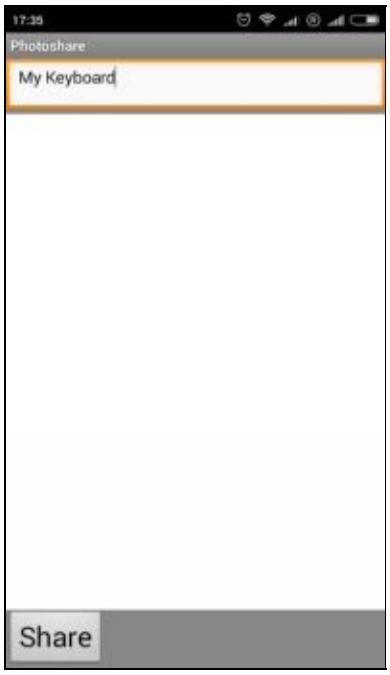
15. Go to Designer and click on Connect and select AI Companion.

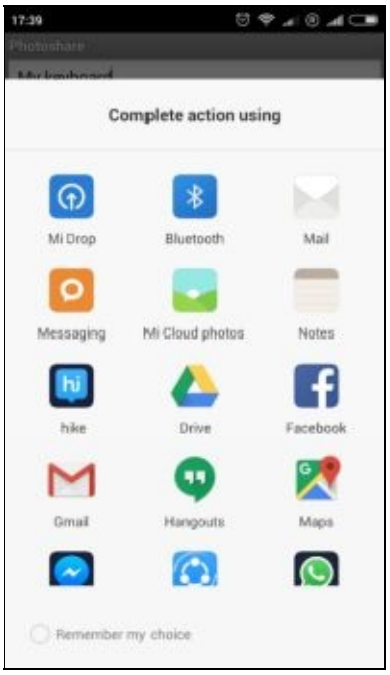


16. Now open MIT App Inventor 2 Companion in your mobile/Tablet.And enter the code or scan the QR code from your mobile and click on connect with code.Note:-Your Mobile/Tablet should be connected to same wi-fi/network to which your computer/Laptop is connected.



17. Now you should see your app in your phone for live testing. Enter some text in TextBox and click on Share button to take picture and share. When you click on Share button your device camera will open to take picture and when you take picture it will show all the apps using which you can share your Message with your picture taken.


17:35
Photoshare
My Keyboard
Share

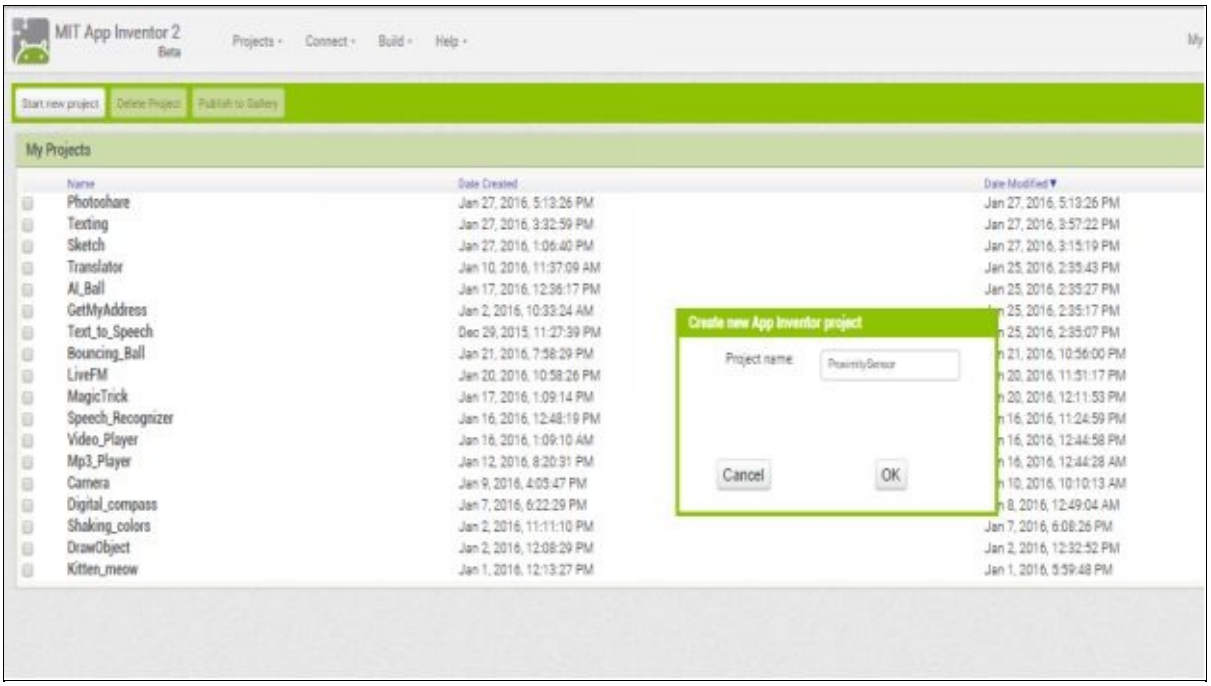

17:39
Photoshare
Complete action using
Mi Drop
Bluetooth
Mail
Messaging
Mi Cloud photos
Notes
hike
Drive
Facebook
Gmail
Hangouts
Maps
Remember my choice


17:45
Compose
To
Subject
My Keyboard
q w e r t y u i o p
a s d f g h j k l
z x c v b n m
?123
@
Return

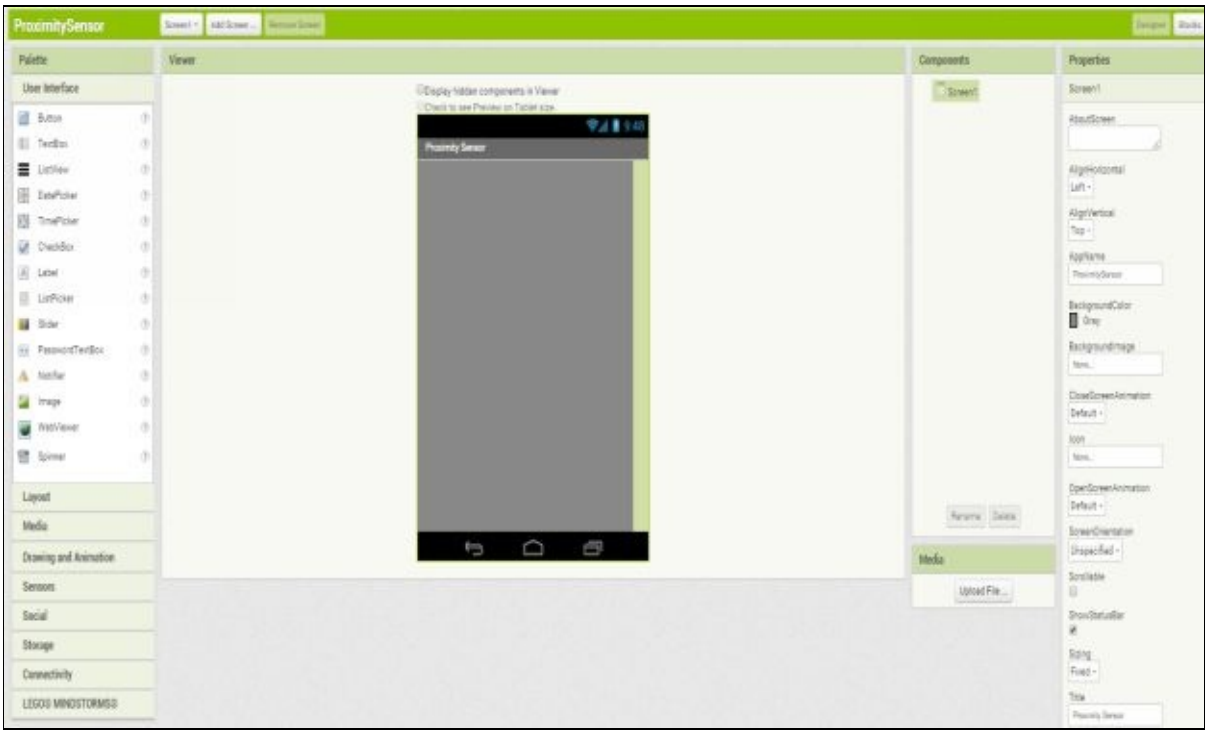

17:46
Photoshare
My Keyboard
Share

Chapter 22 Proximity Sensor app

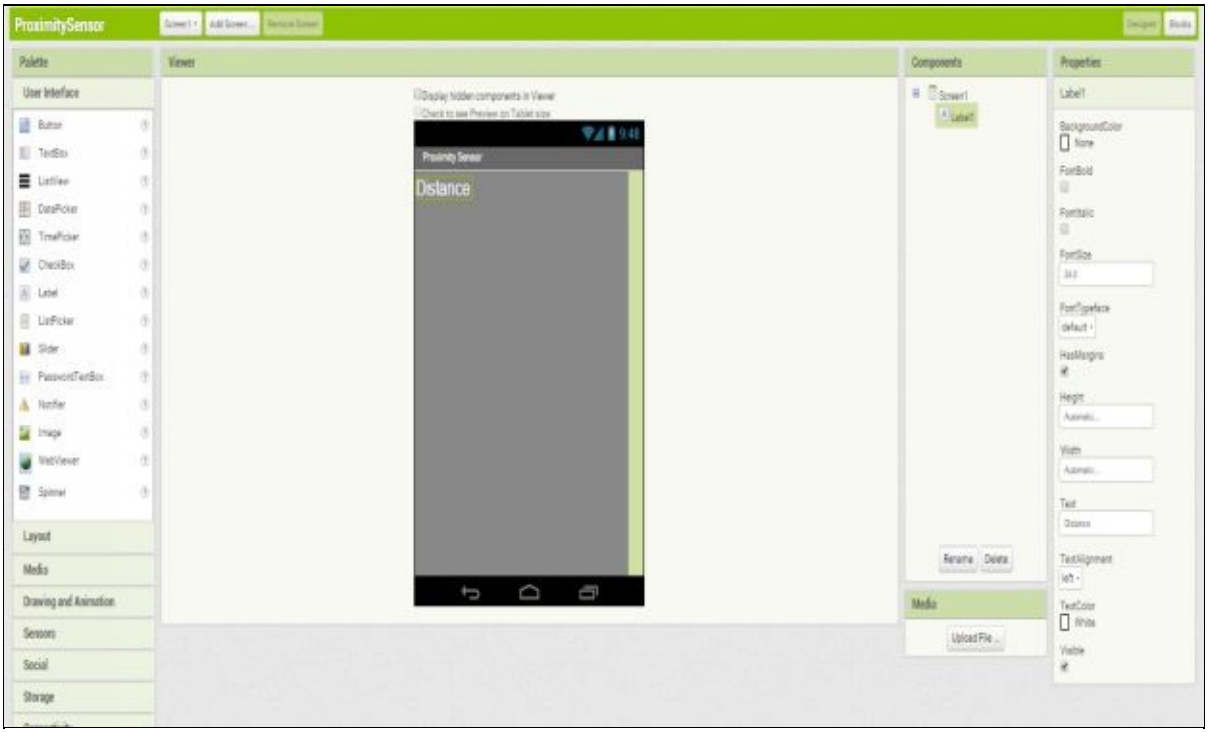
1. Click on Start new project and give it name ProximitySensor and click OK.



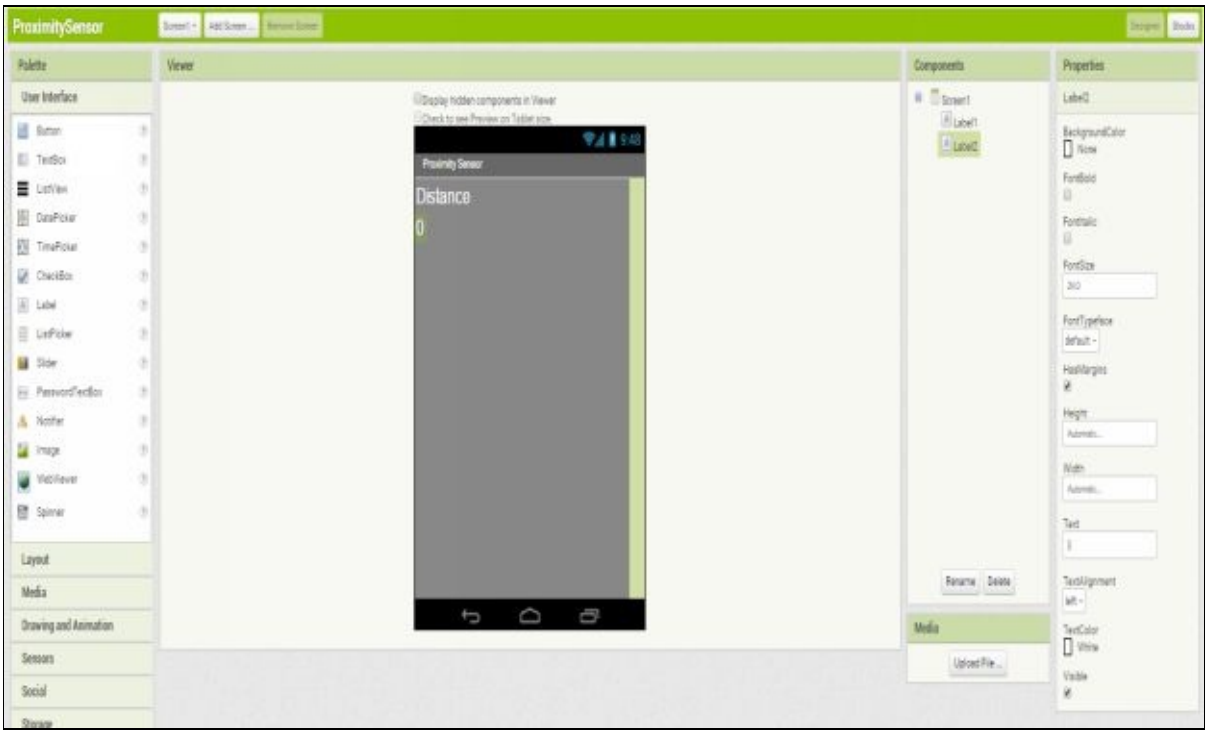
2. Change Screen1 Title to ‘Proximity Sensor’ and BackgroundColor to Gray.



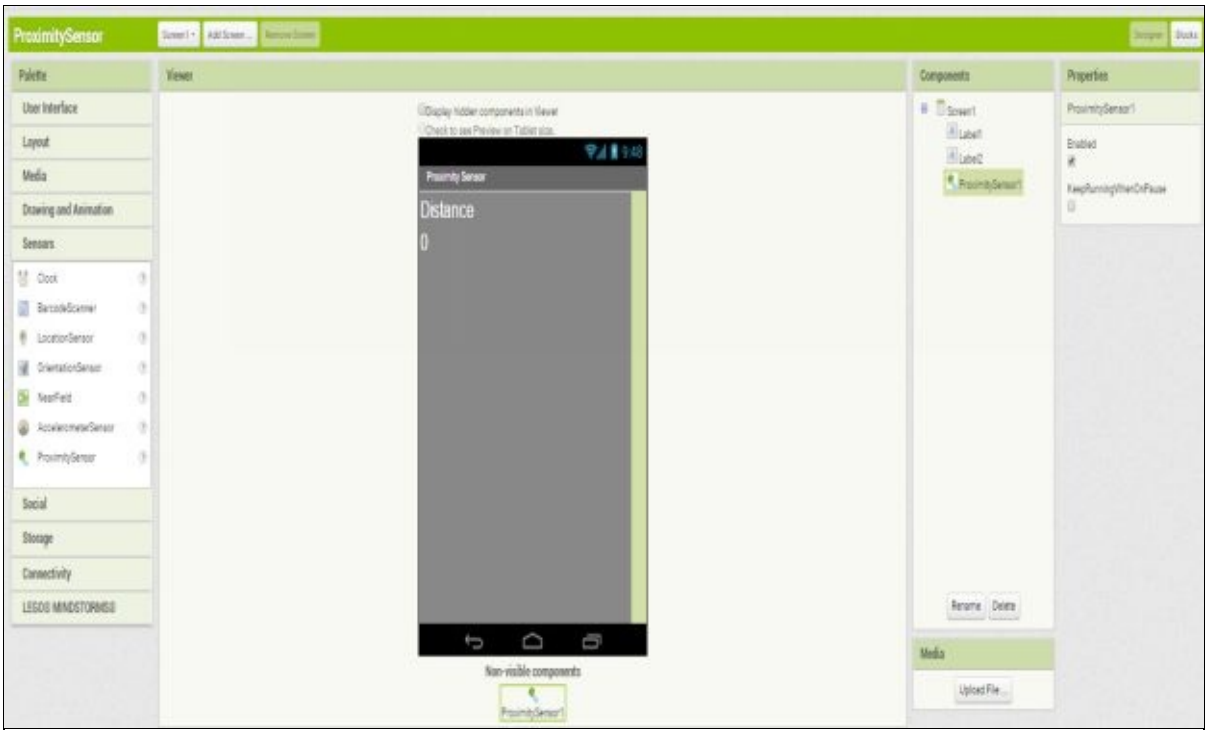
3. Drag a Label from Palette to Viewer screen and change its text to Distance and FontSize to 24 and TextColor to White.



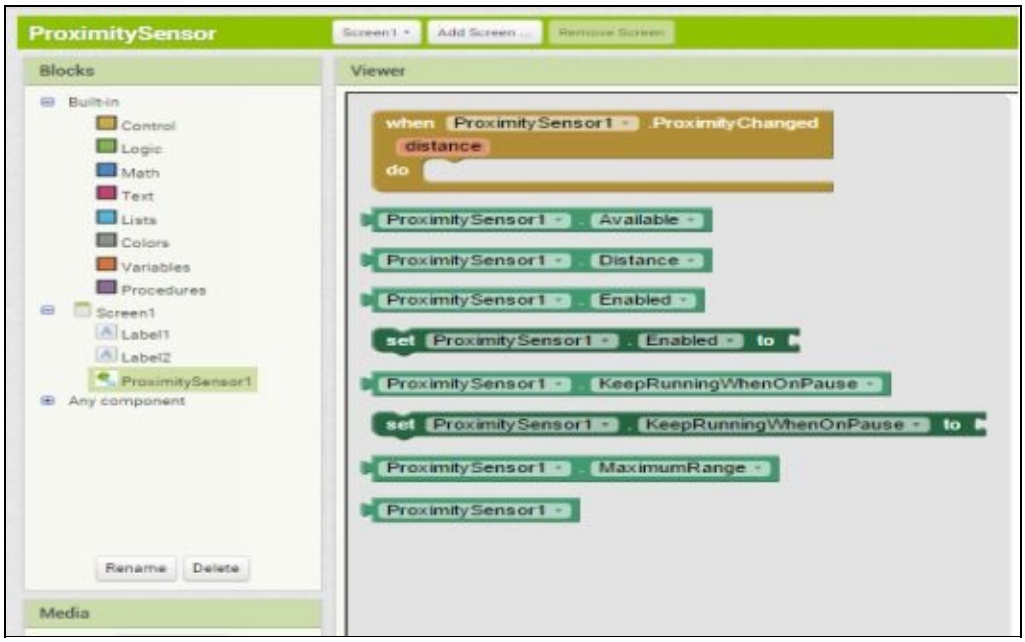
4. Drag another Label from Palette to Viewer screen and change its text to 0 and FontSize to 24 and TextColor to White.

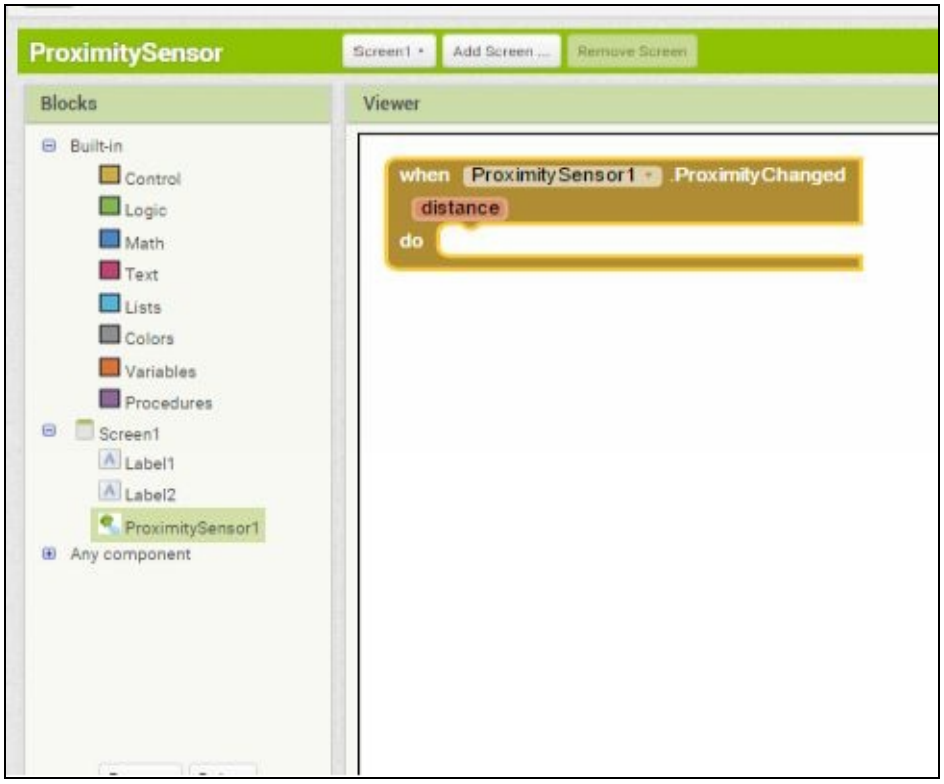


5. Drag a ProximitySensor (Non-visible) component from Palette to Viewer.



6. Go to Blocks and click on ProximitySensor1 and select block. This block will decide what to do when Proximity is changed.





7. Click on Label2 and select  block.

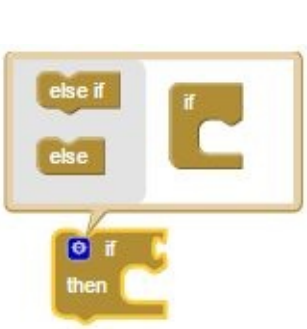
8. Click on ProximitySensor1 and select 

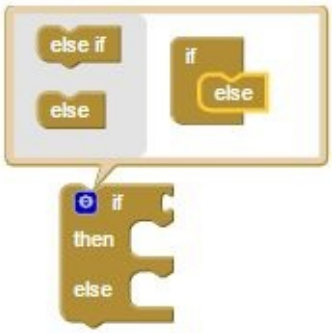
. block. This block will represent Proximity distance. Label2 will show the distance on your app screen.

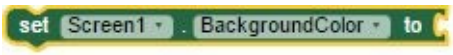
9. Click on Control under Built-in and select 

block. This is a conditional block. If given condition is true the it will choose some action to be performed and if condition is false/not true then it will execute else part.

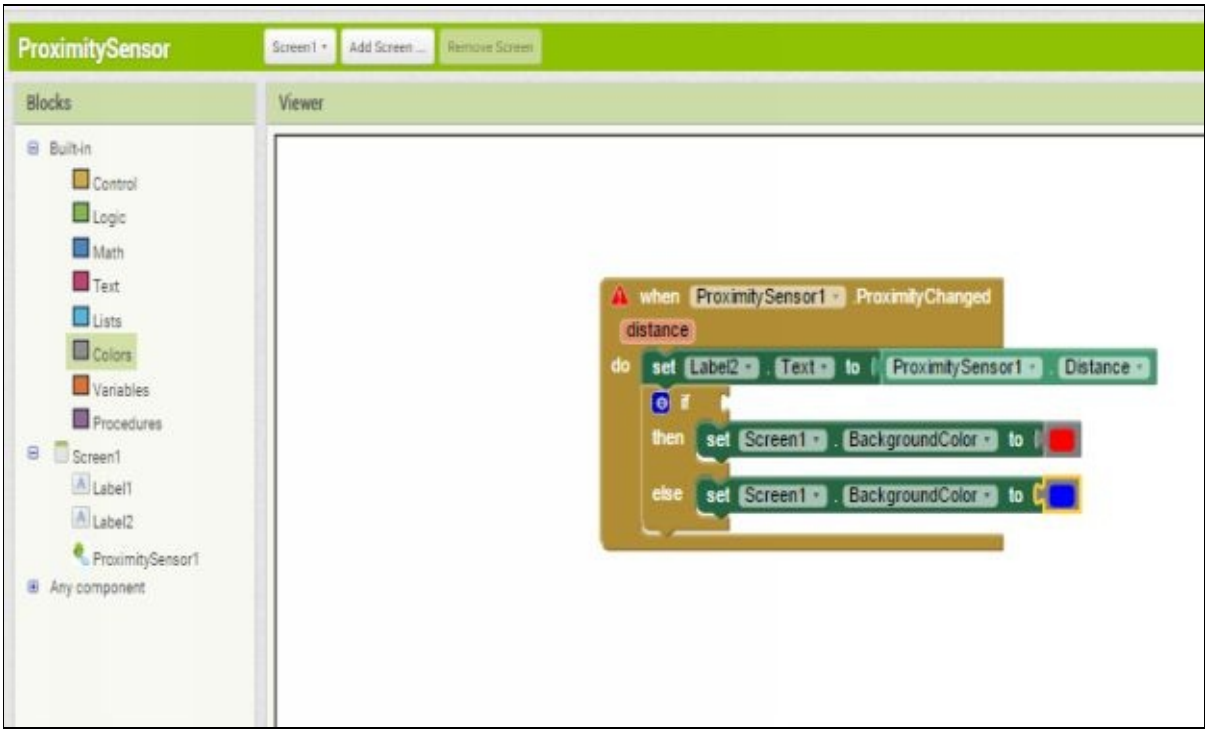
10. Click on  and add one else block to  block.



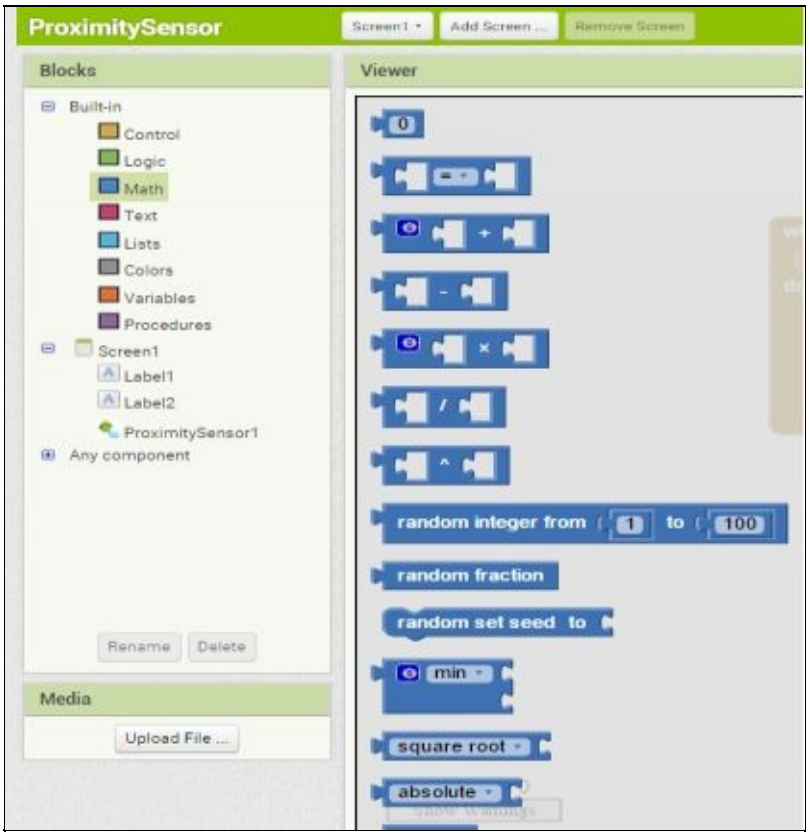


11. Click on Screen1 and scroll down to select  block twice.

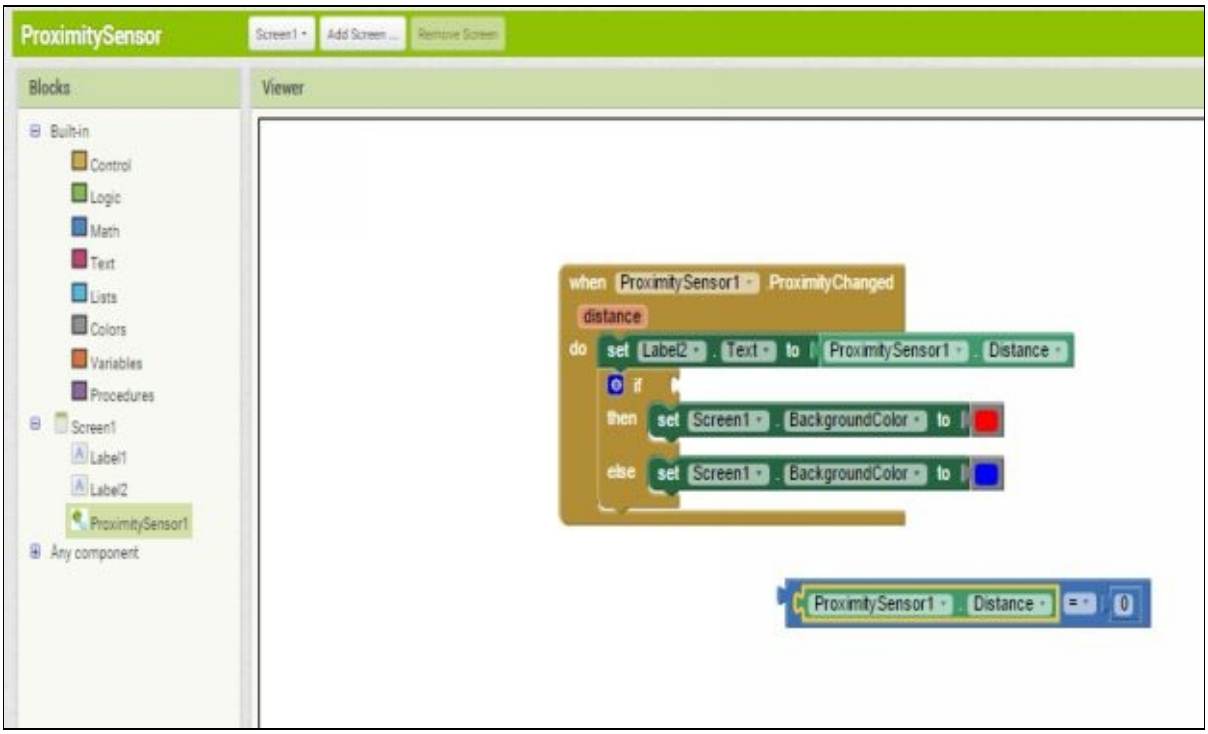
12. Click on Colors and select  blocks.



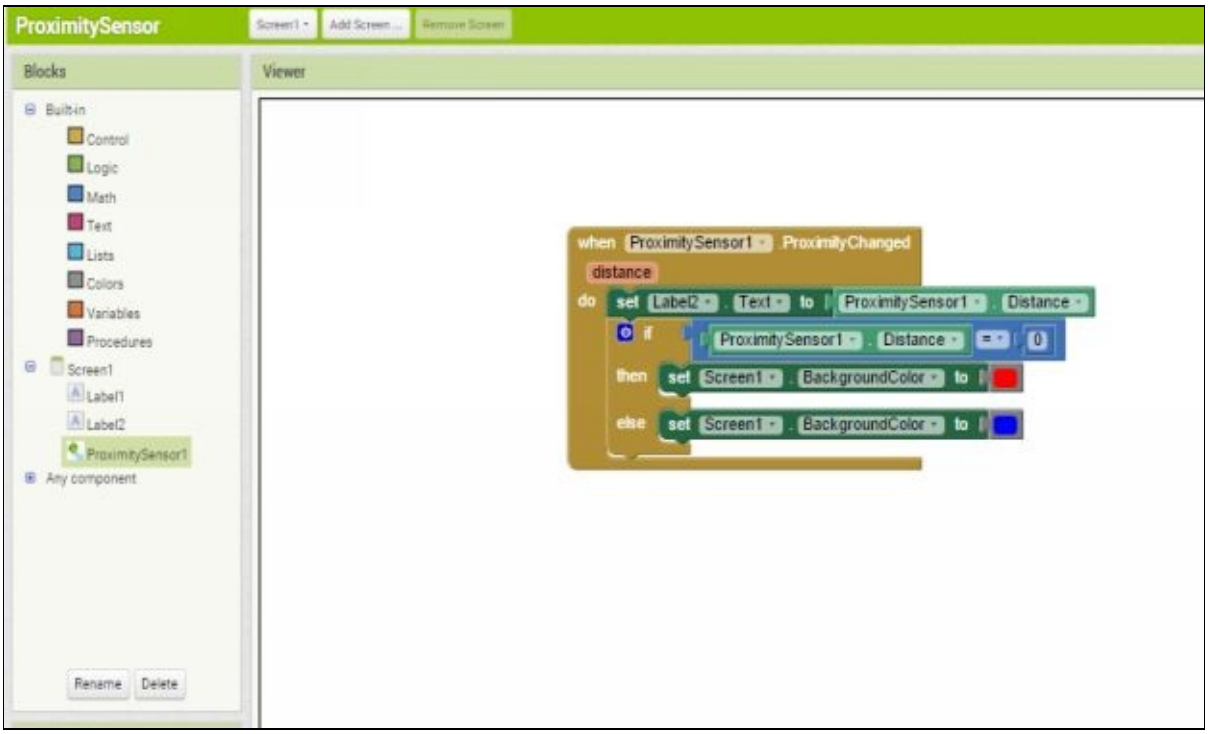
13. Click on Math under Built-in and select  blocks.



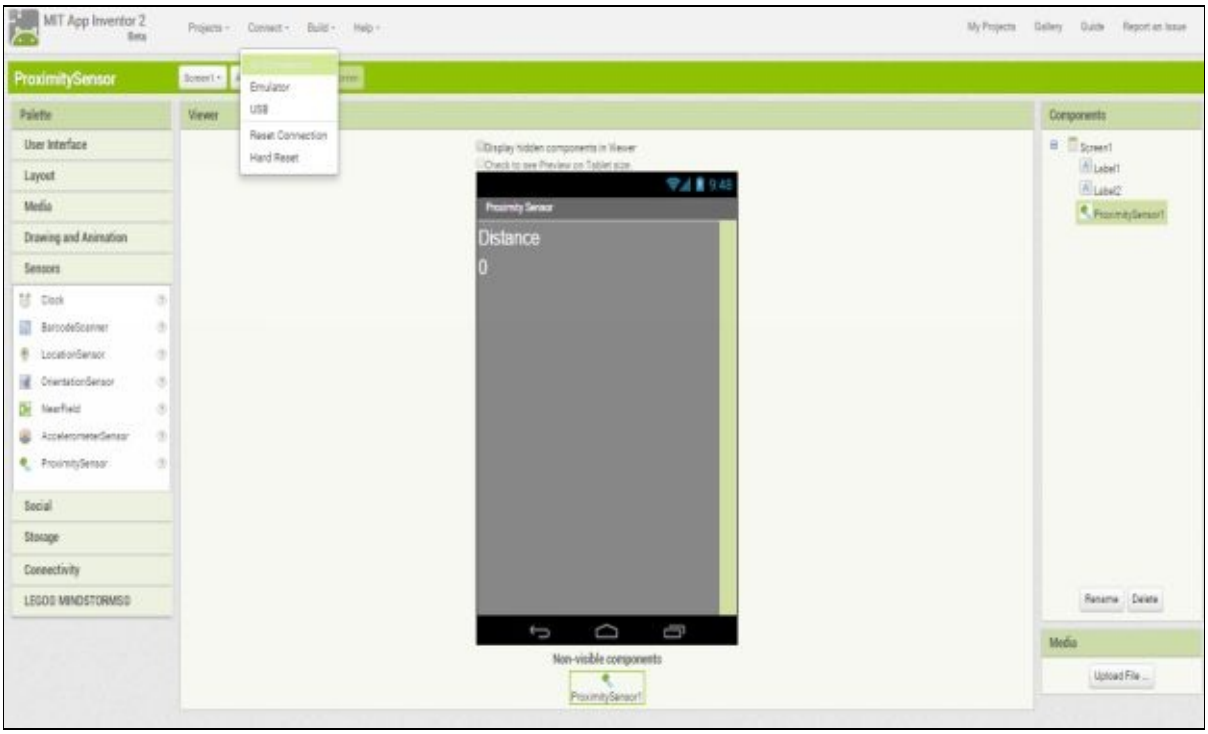
14. Click on ProximitySensor1 and select  block.



15. Attach the blocks as shown below. So if Proximity distance is equal to 0 the app will change screen color to Red and if distance is not equal to 0 then else part will be executed i.e, screen color will be changed to Blue.

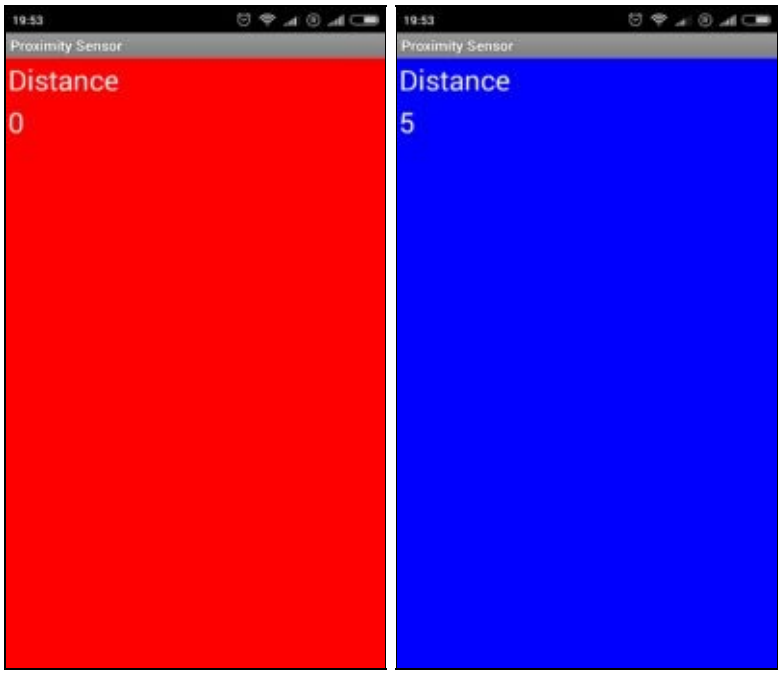


16. Go to Designer and click on Connect and then select AI Companion.



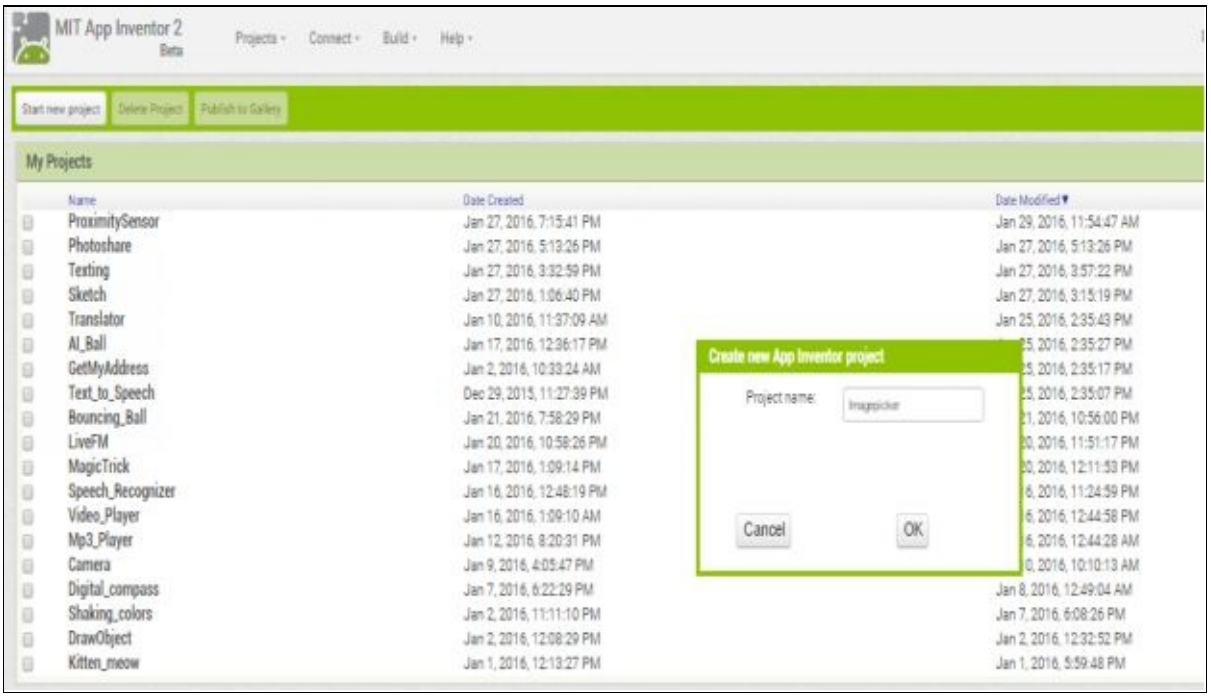
17. Now open MIT App Inventor 2 Companion in your mobile/Tablet.And enter the code or scan the QR code from your mobile and click on connect with code.Note:-Your Mobile/Tablet should be connected to same wi-fi/network to which your computer/Laptop is connected.

18. Now you should see your app in your phone for live testing. Proximity sensor is generally available on top of your phone. Bring your hand near sensor and see the distance and color changing with changing Proximity.

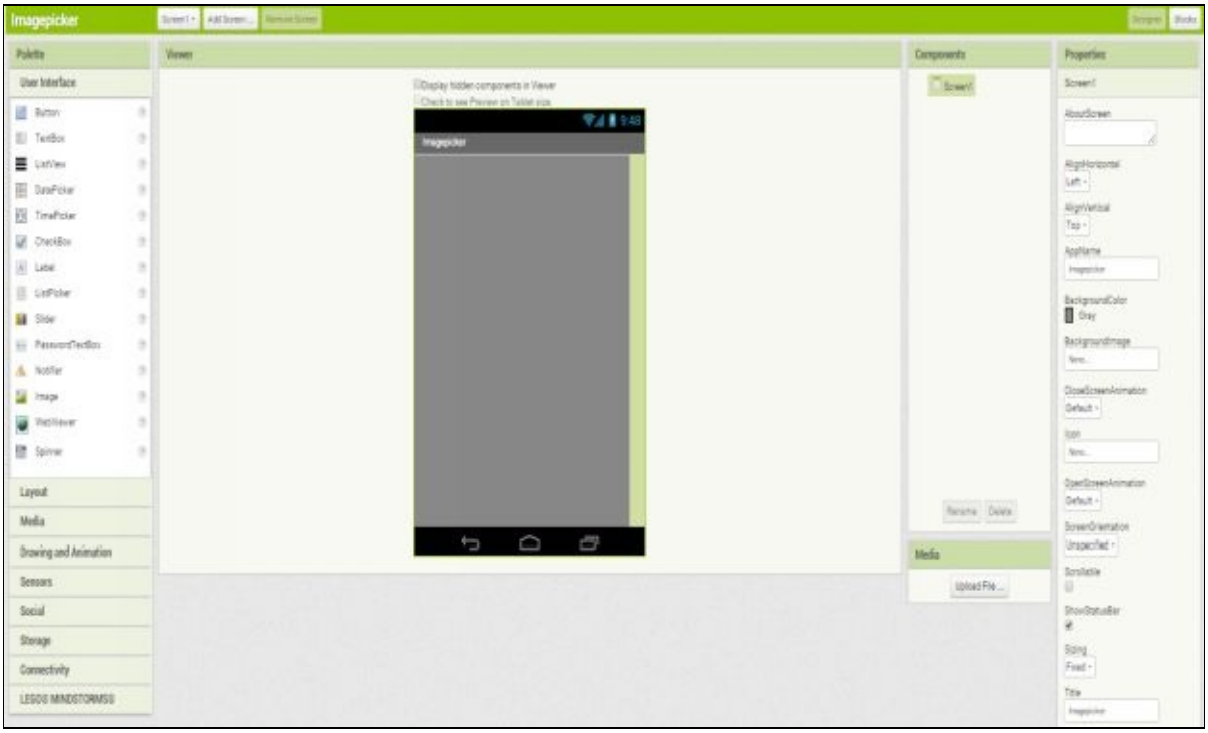


Chapter 23 Image Picker app

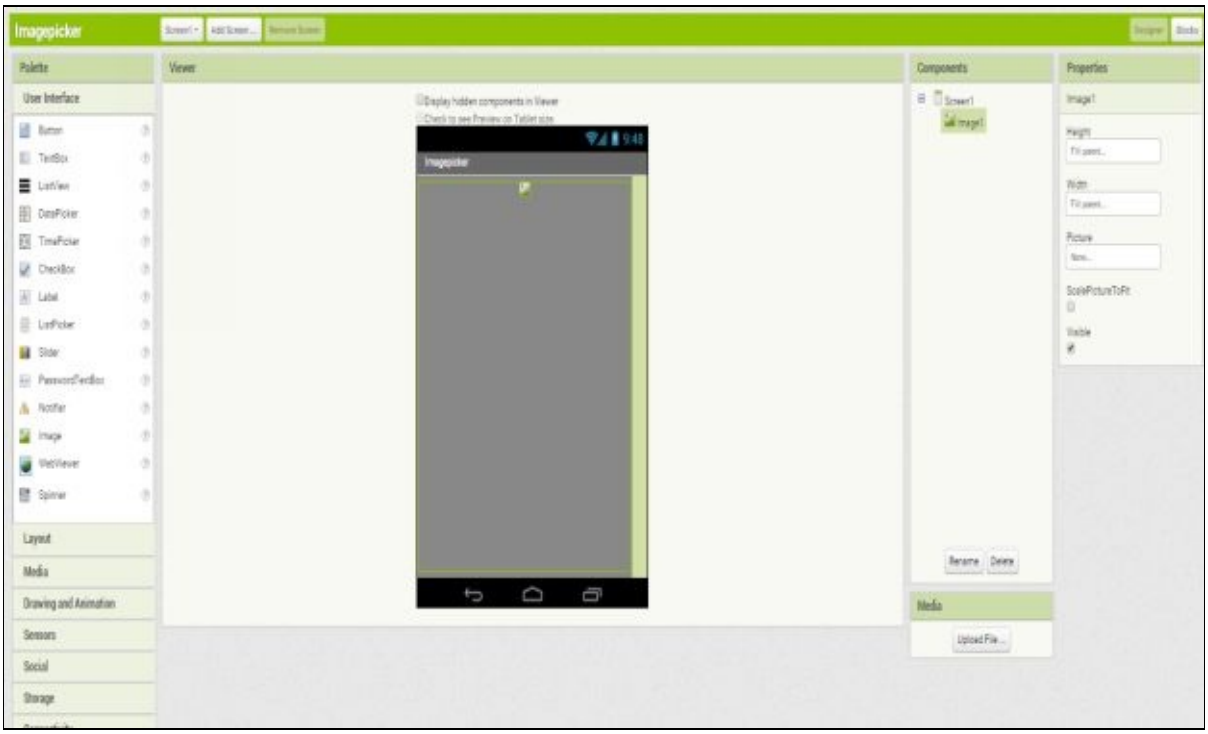
1. Click on Start new project and give it name Imagepicker and click OK.



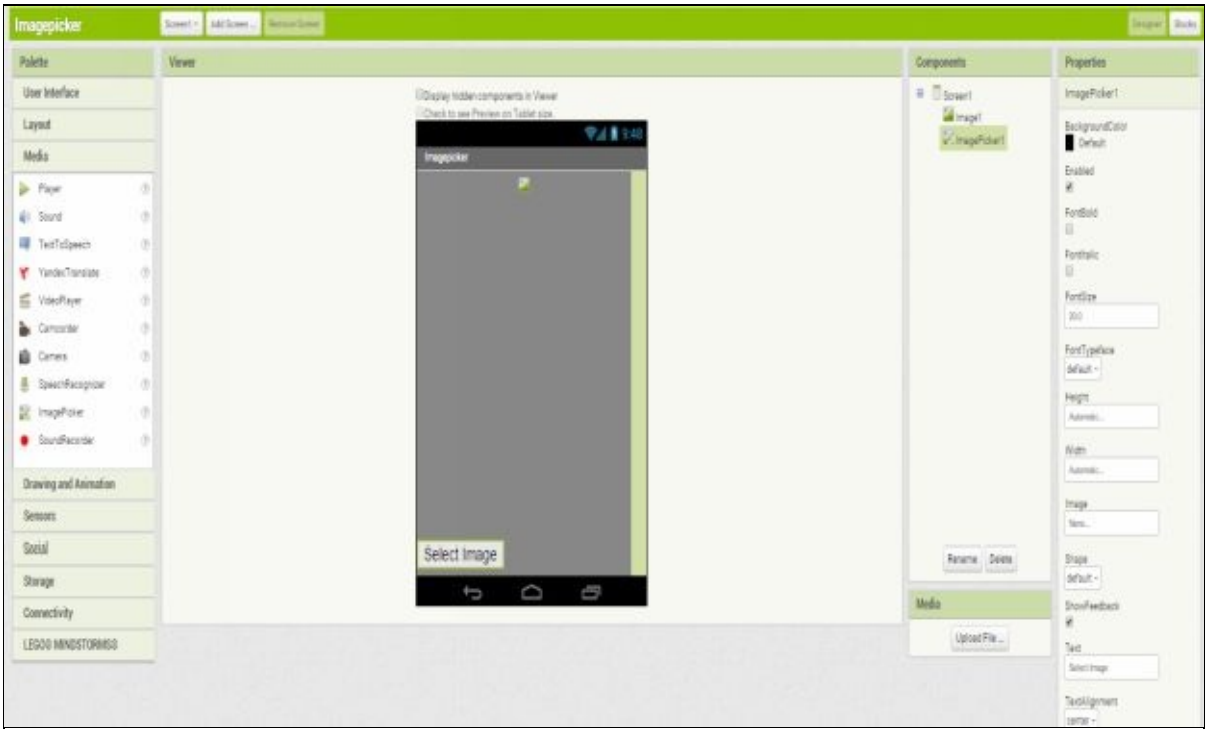
2. Change Screen1 Title to Imagepicker and BackgroundColor to Gray.



3. Drag an Image from Palette to Viewer screen and set its Height and Width to fill parent.

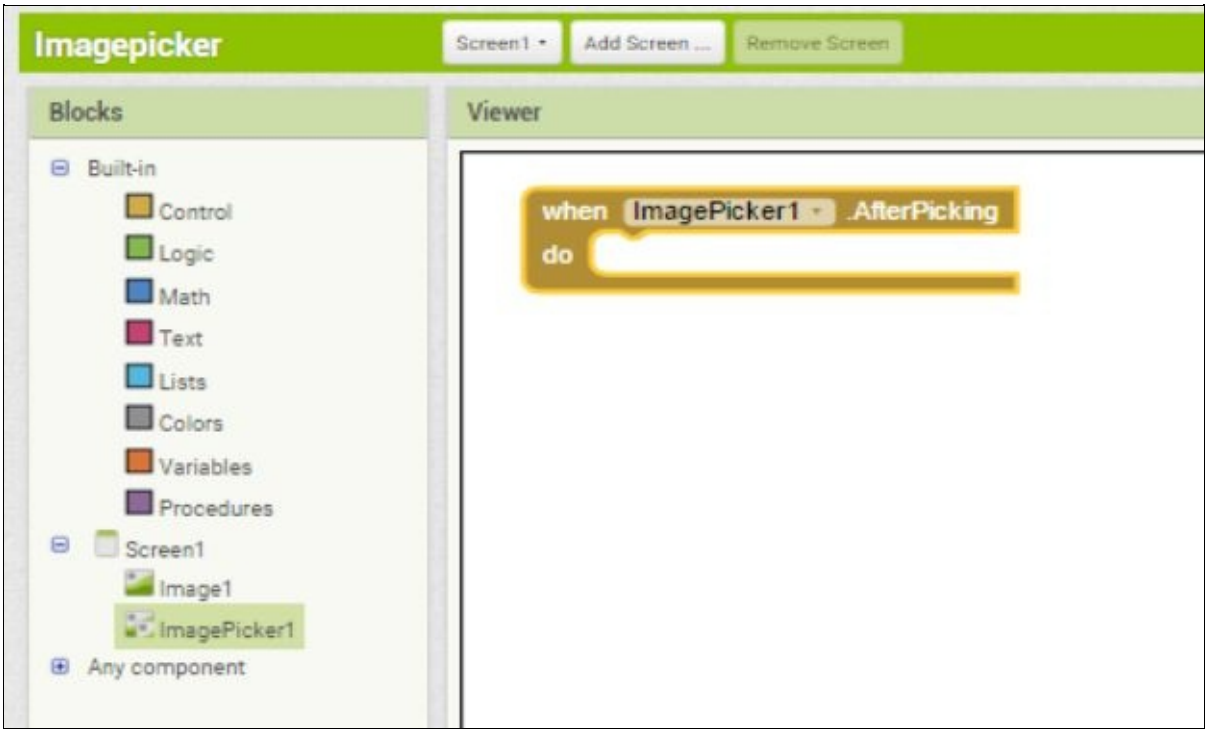


4. Drag an ImagePicker from Palette to Viewer screen and change its Text to Select Image and FontSize to 20.0.



5. Go to Blocks and click on ImagePicker1 and select 

block. This block will open your phone's gallery and decide what to do with the selected image.

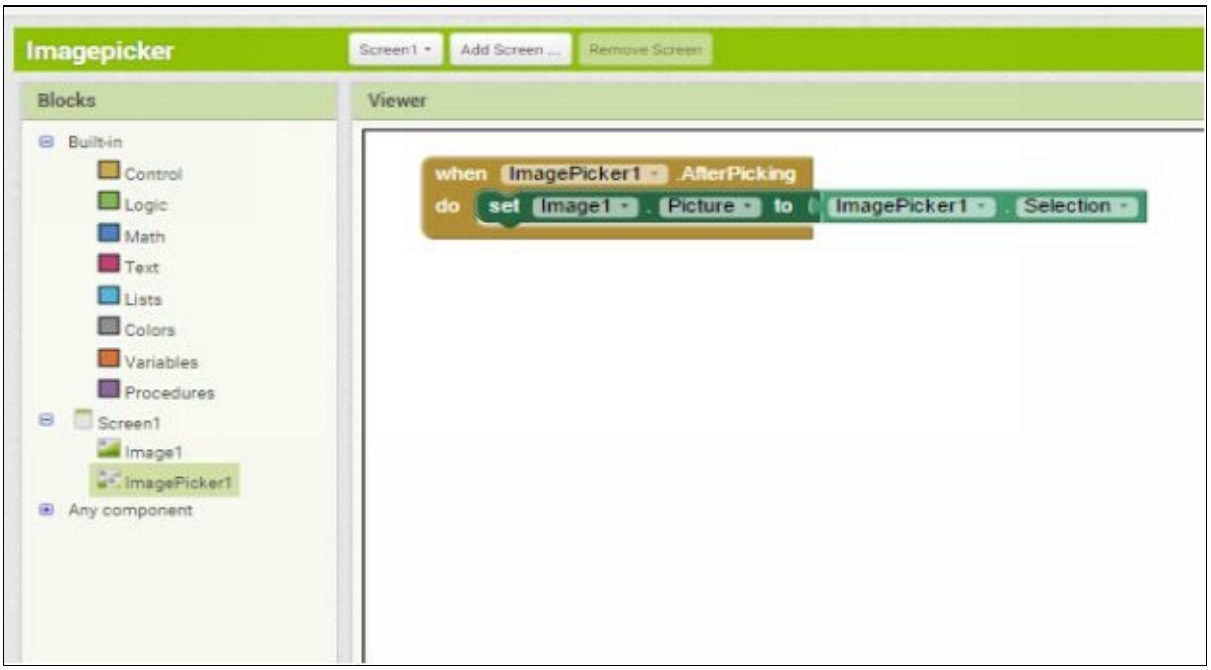


6. Click on Image1 and select 

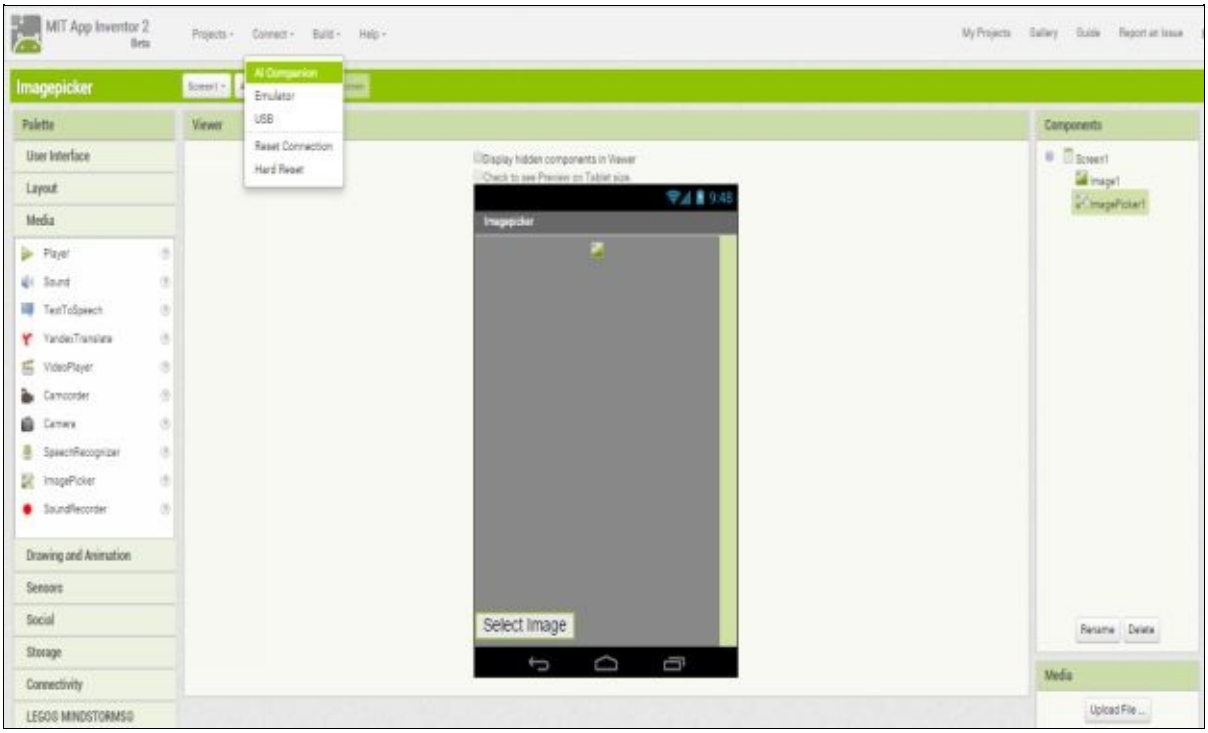
block. This block will set Image1's picture.

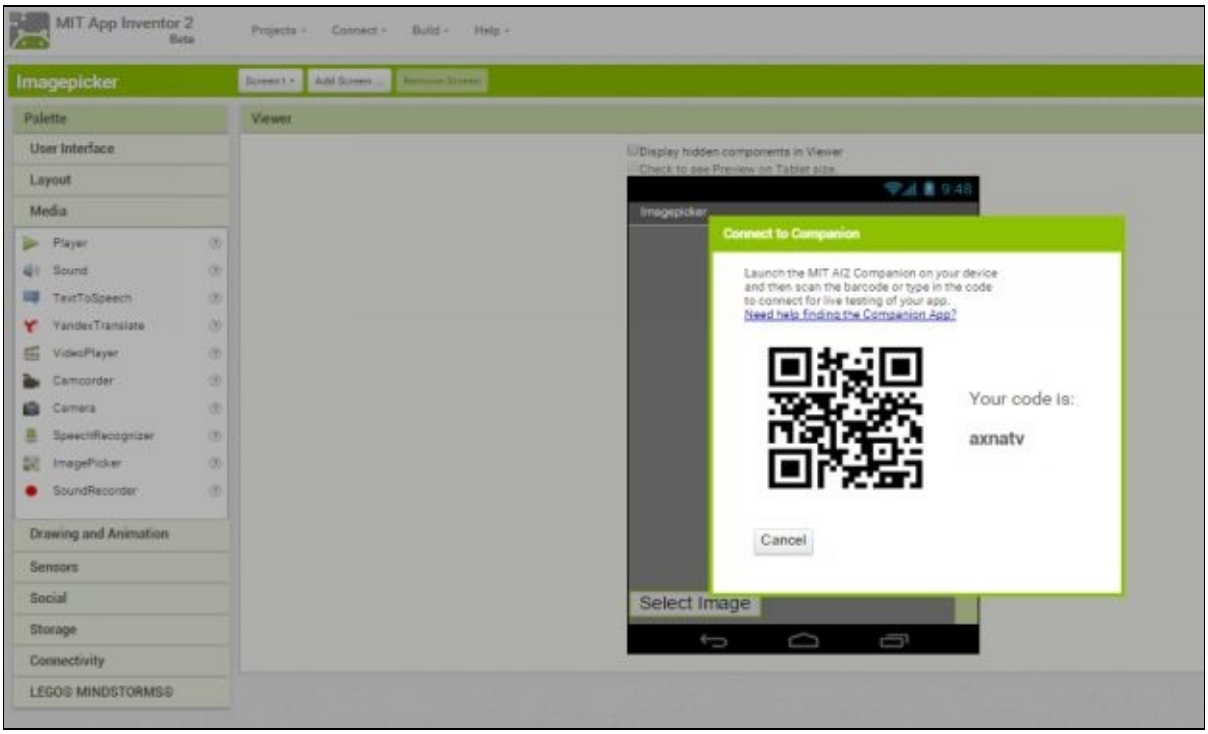
7. Click on ImagePicker1 and scroll down to select 

block. This block will represent the selected image from gallery. So when you click ImagePicker1 , app will open Phone's gallery and you can choose an image and after picking the image it will be shown as Image1's image.



8. Go to Designer and click on Connect and select AI Companion.

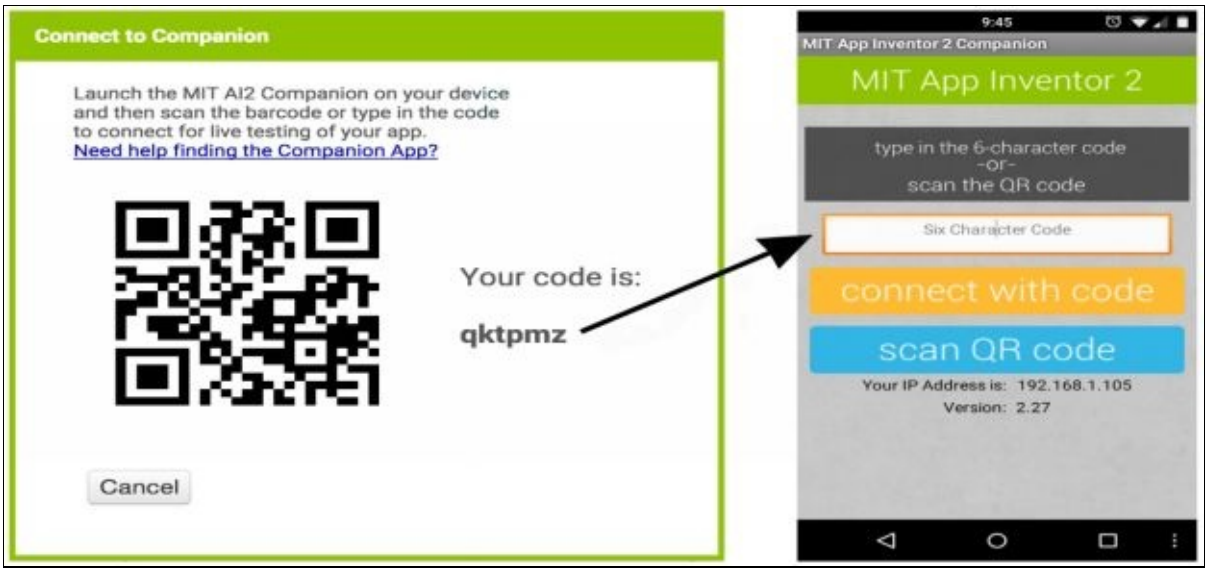




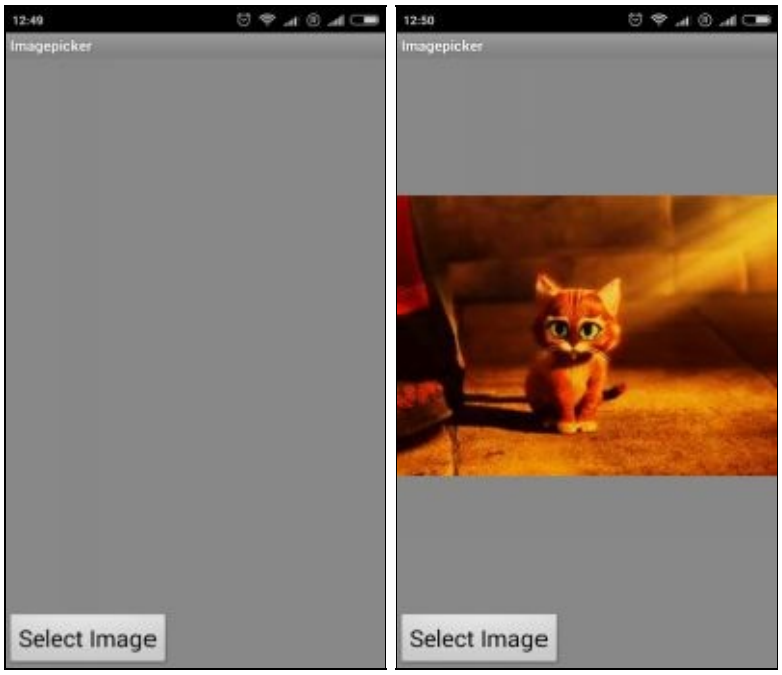
9. Now open MIT App Inventor 2 Companion in your mobile/Tablet.

And enter the code or scan the QR code from your mobile and click on connect with code.

Note:-Your Mobile/Tablet should be connected to same wi-fi/network to which your computer/Laptop is connected.

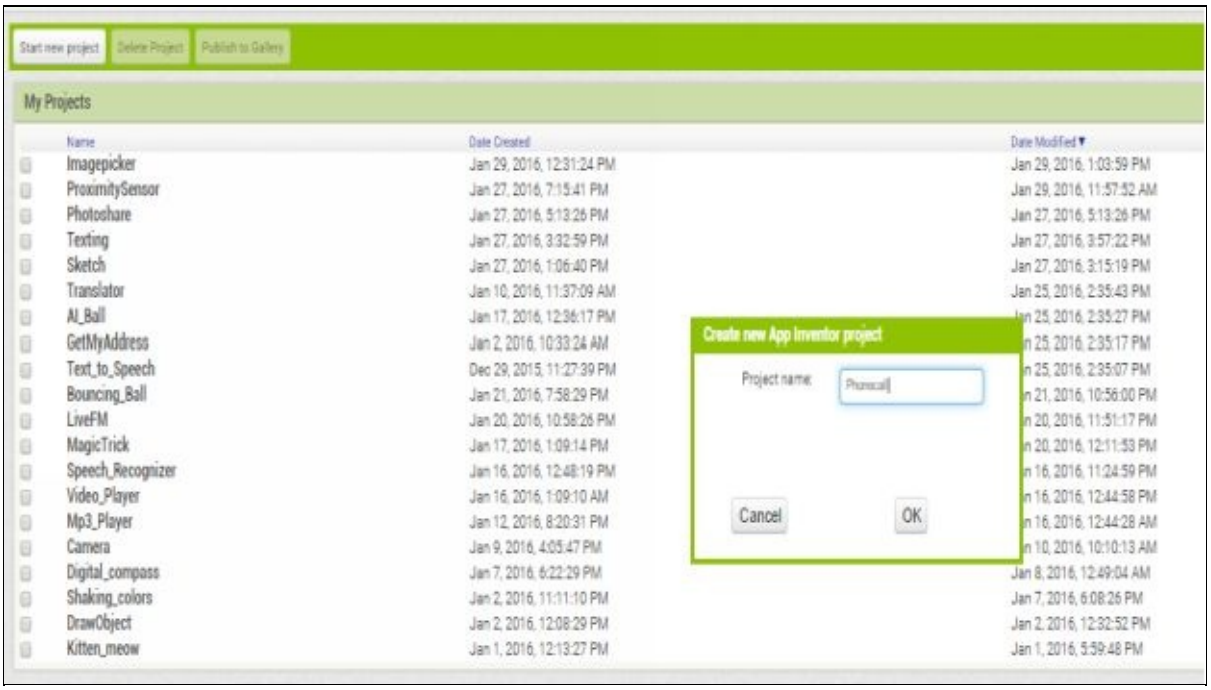


10. Now you should see your app in your phone for live testing. Click on Select Image Button to open Gallery and select Image from your phone.

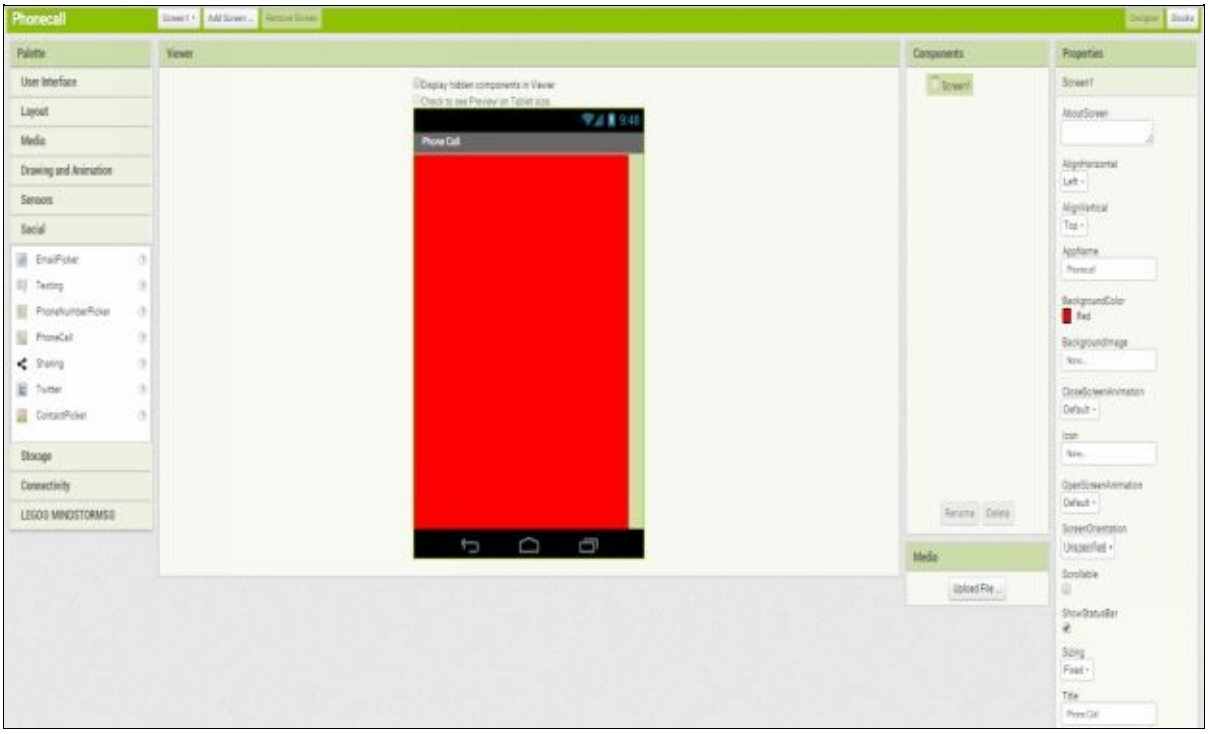


Chapter 24 Phone call app

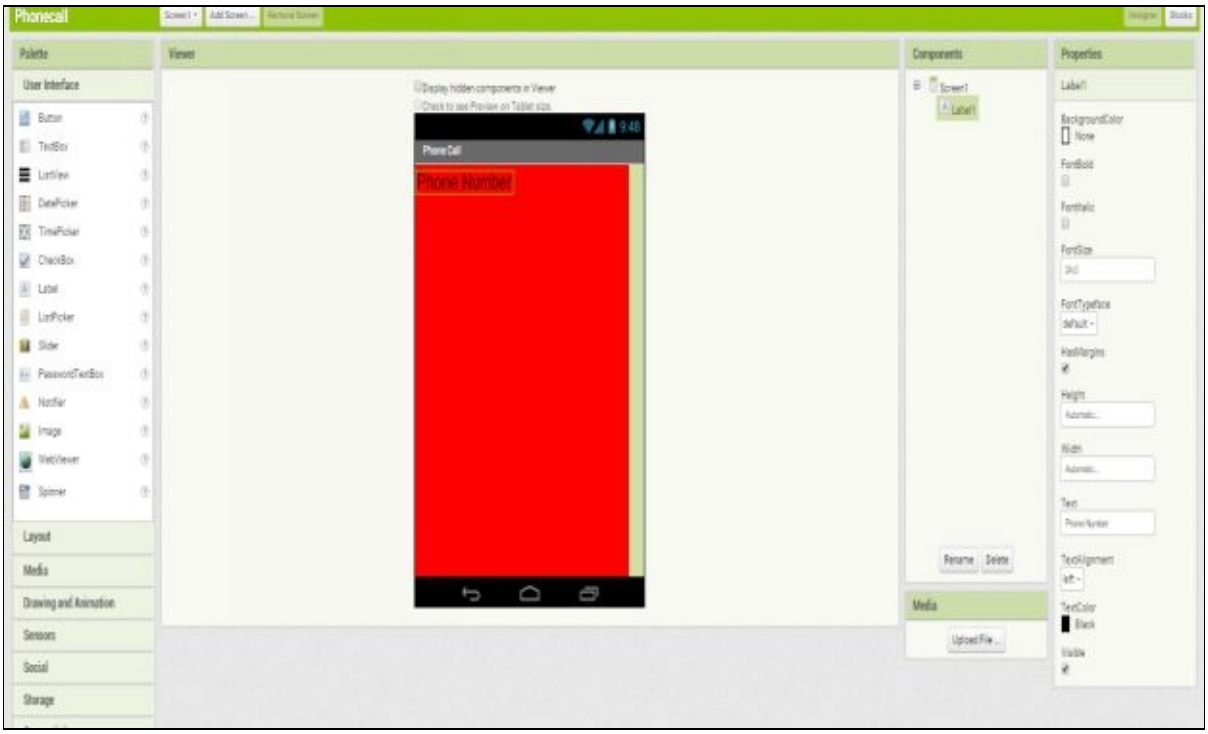
1. Click on Start new project and give it name Phonecall and click OK.



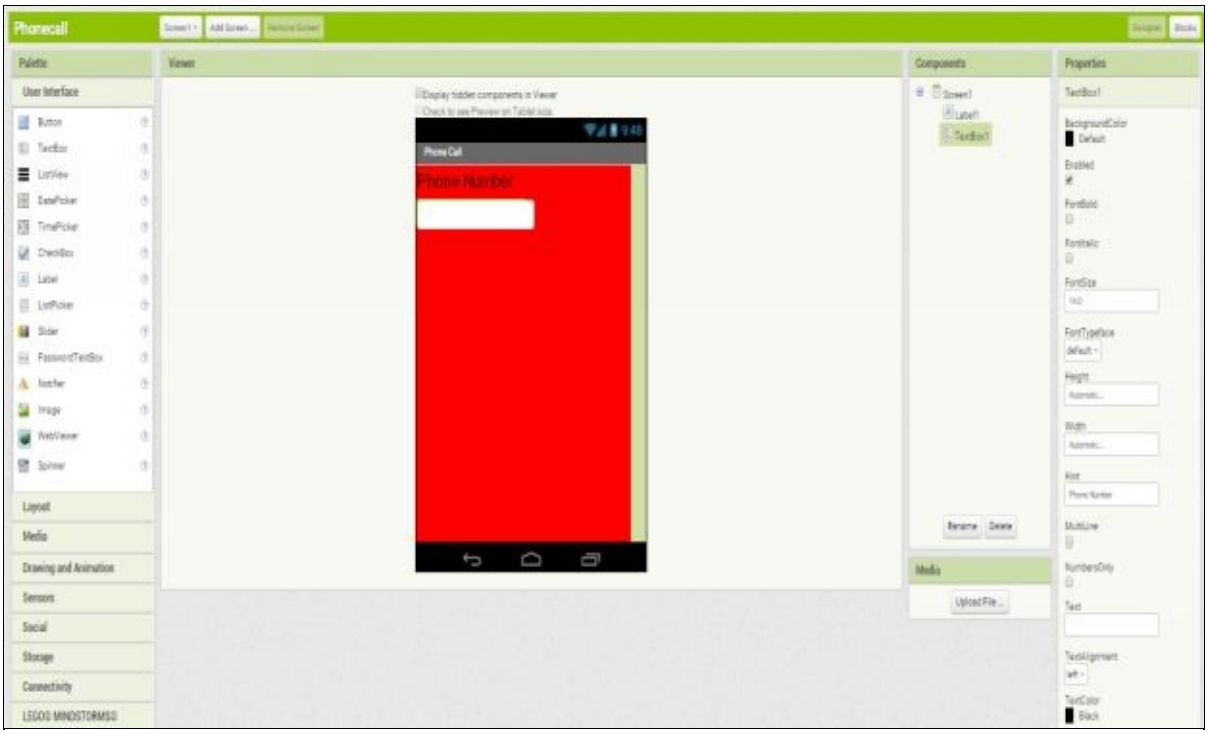
2. Change Screen1 Title to Phone Call and BackgroundColor to Red.



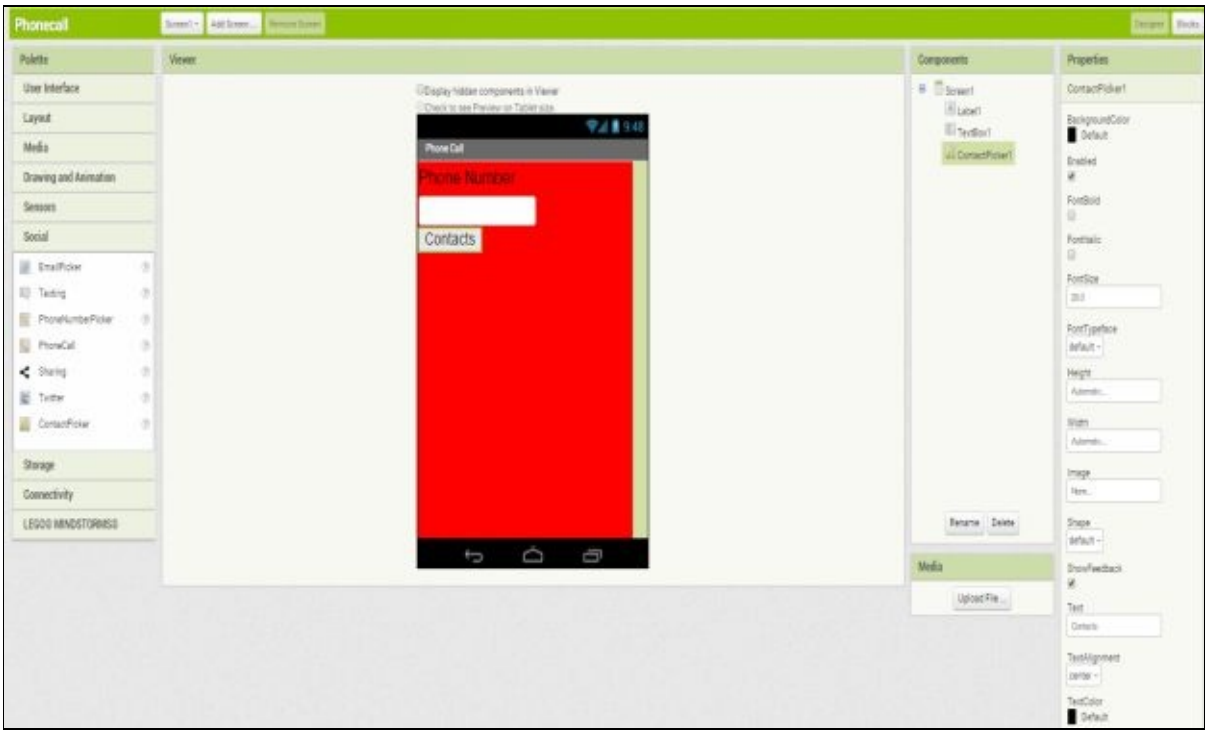
3. Drag a Label from Palette to Viewer screen and change its Text to Phone Number and FontSize to 20.0.



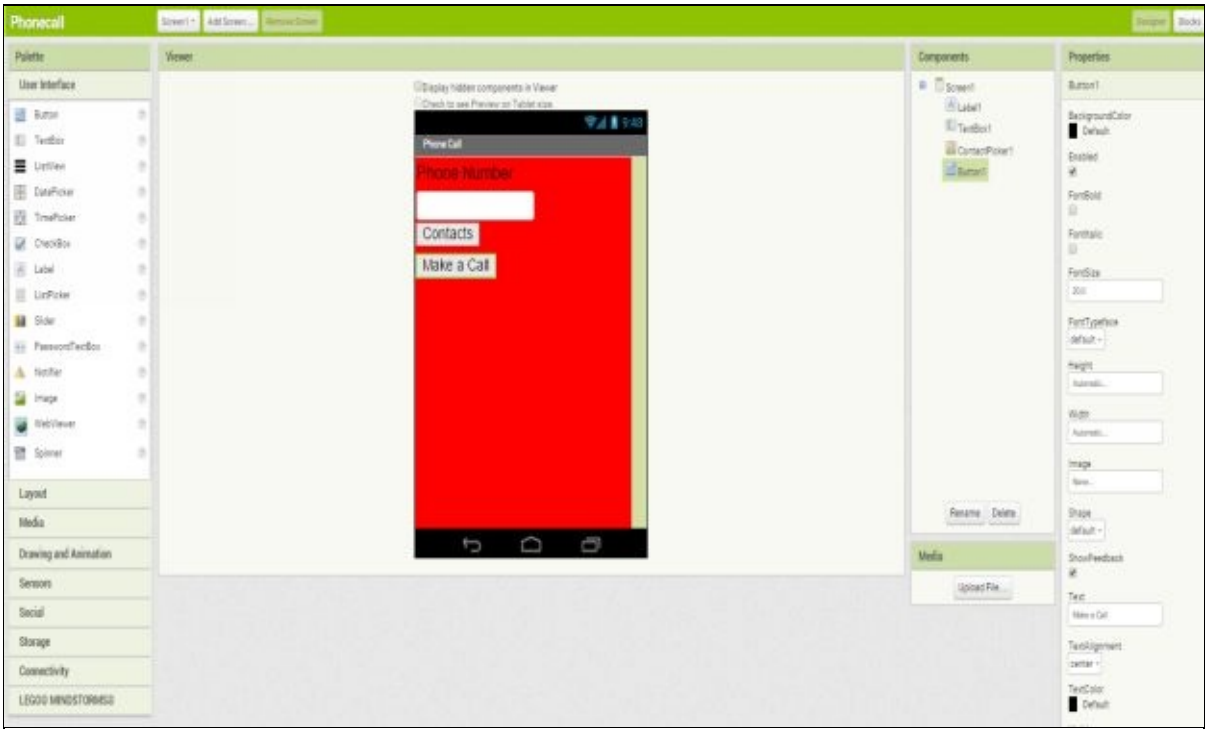
4. Drag a TextBox from Palette to Viewer screen and Change its Hint to Phone Number.



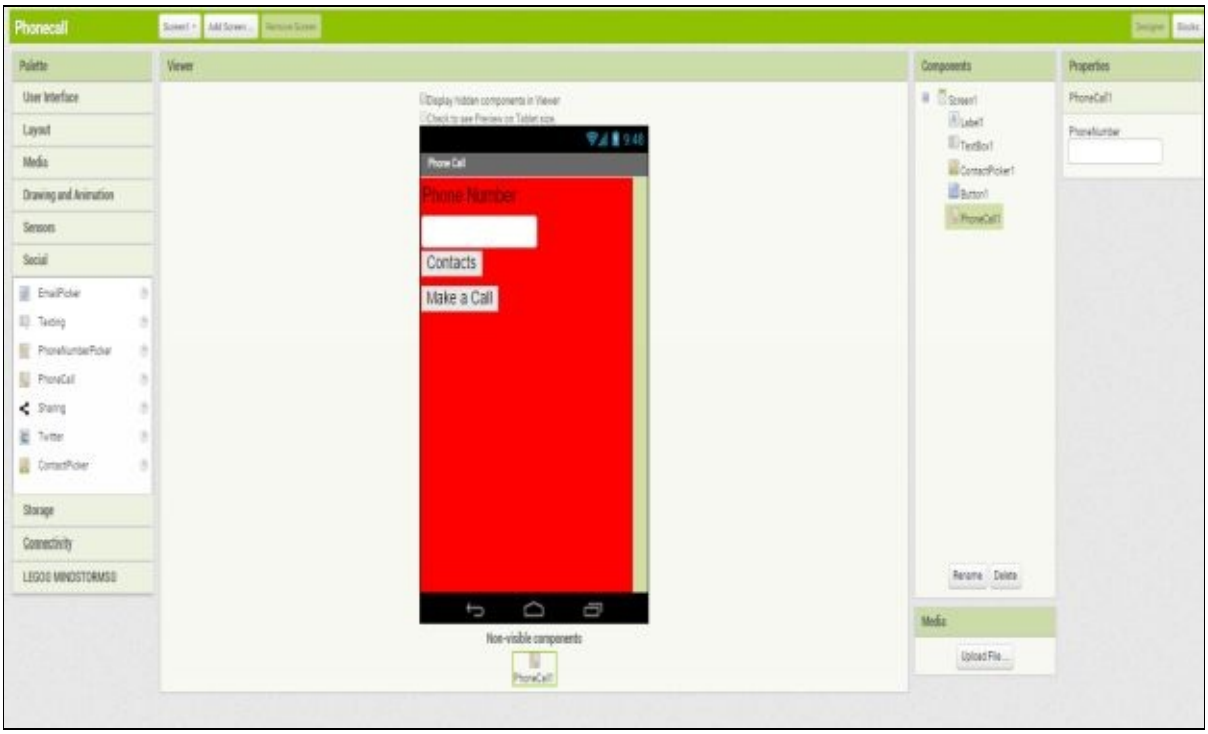
5. Drag a ContactPicker from Palette to Viewer screen and Change its Text to Contacts and FontSize to 20.0.



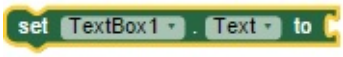
6. Drag a Button from Palette to Viewer screen and Change it's Text to Make a Call and FontSize to 20.0.



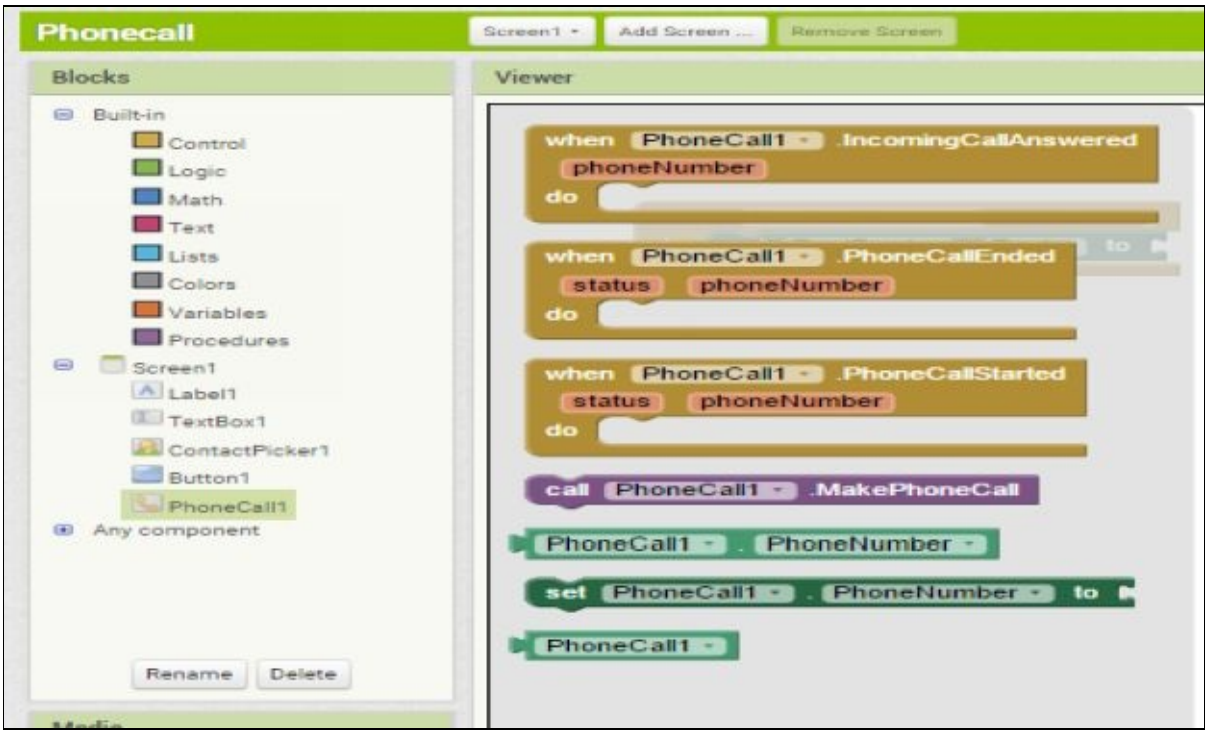
7. Drag a PhoneCall (Non-visible) component from Palette to Viewer.

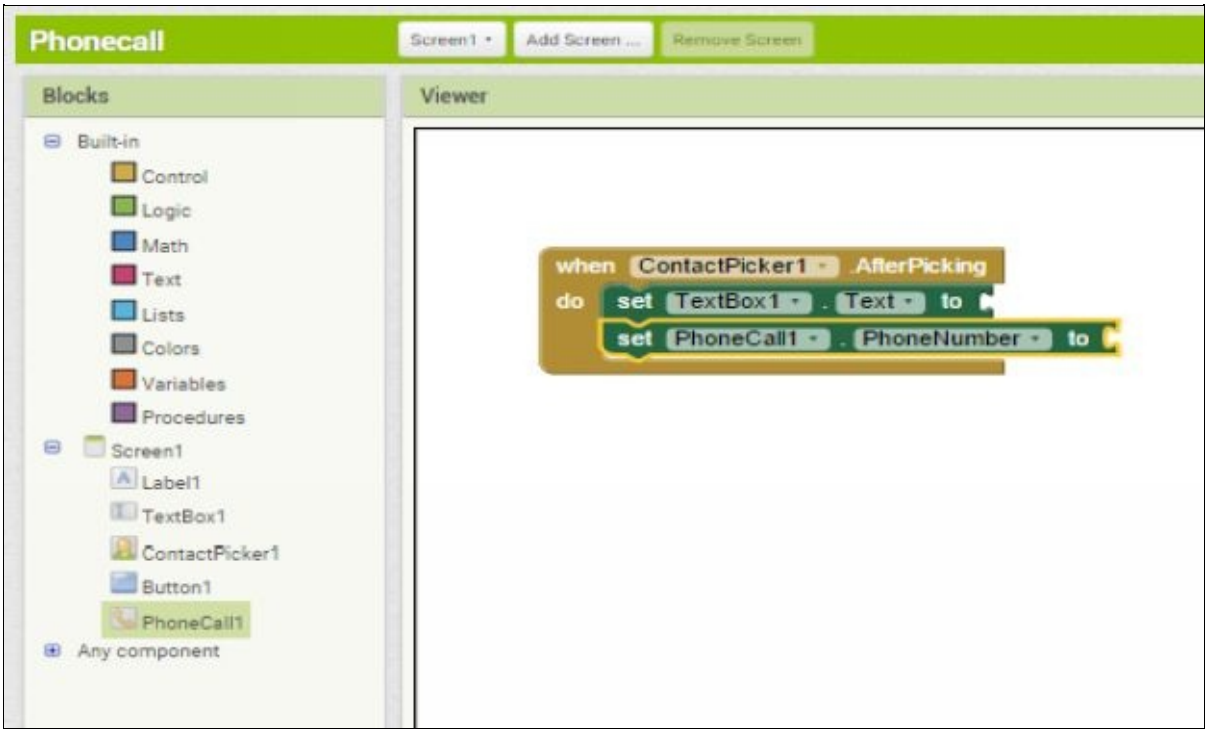


8. Go to Blocks and click on ContactPicker1 and select  block. This block will Open your phone's contact list and decide what to do after you select a contact.

9. Click on TextBox1 and scroll down to select  block.

10. Click on PhoneCall1 and select  block. This block will set the Phone Number for PhoneCall.

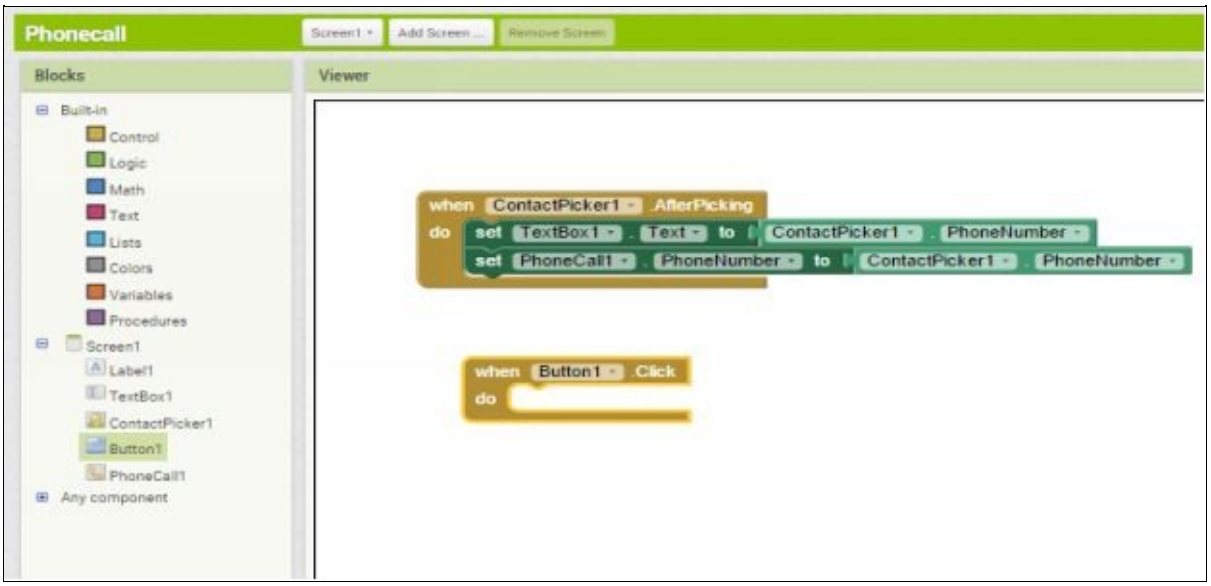




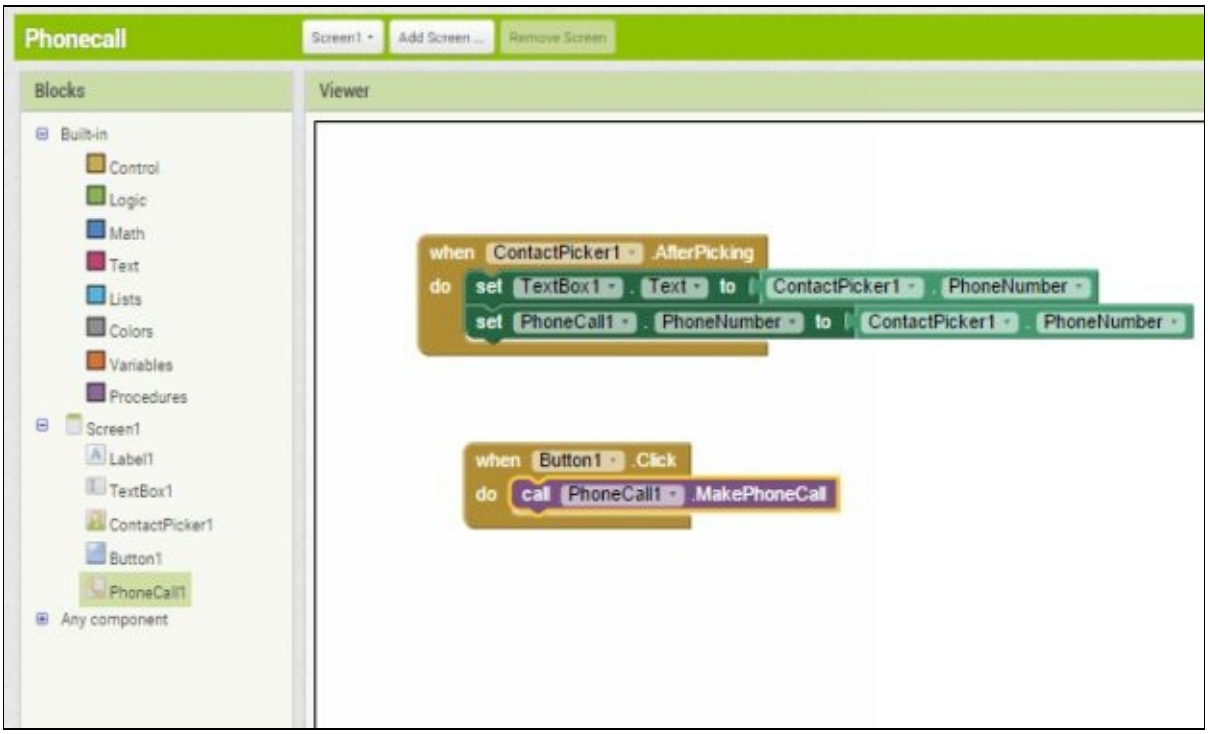
11. Click on ContactPicker1 and scroll down to select 

block twice. This block will represent Phone Number of selected contact from Phone's Contact list.So after picking the contact, app will how the Phone number in TextBox1 and also set the Phone Number for PhoneCall.

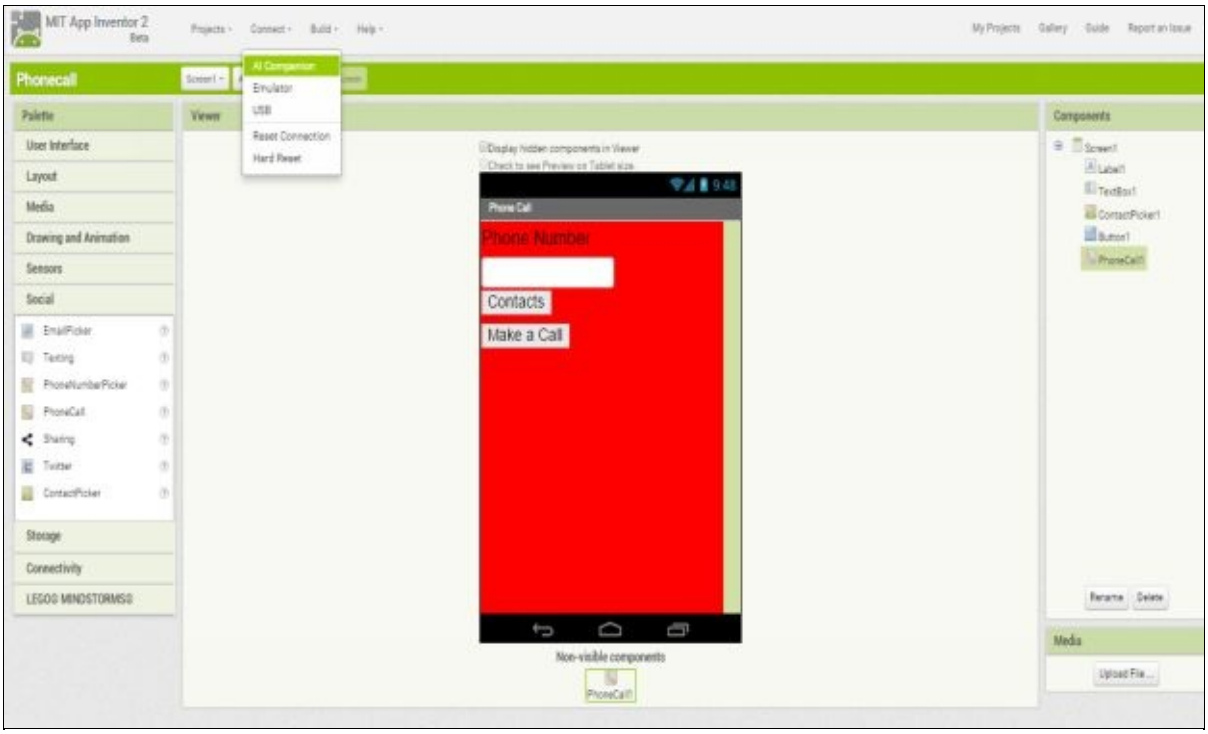
12. Click on Button1 and select  block. This is an event block which will decide what your app will do when this button will be clicked. Here click is an event for Button1 component.

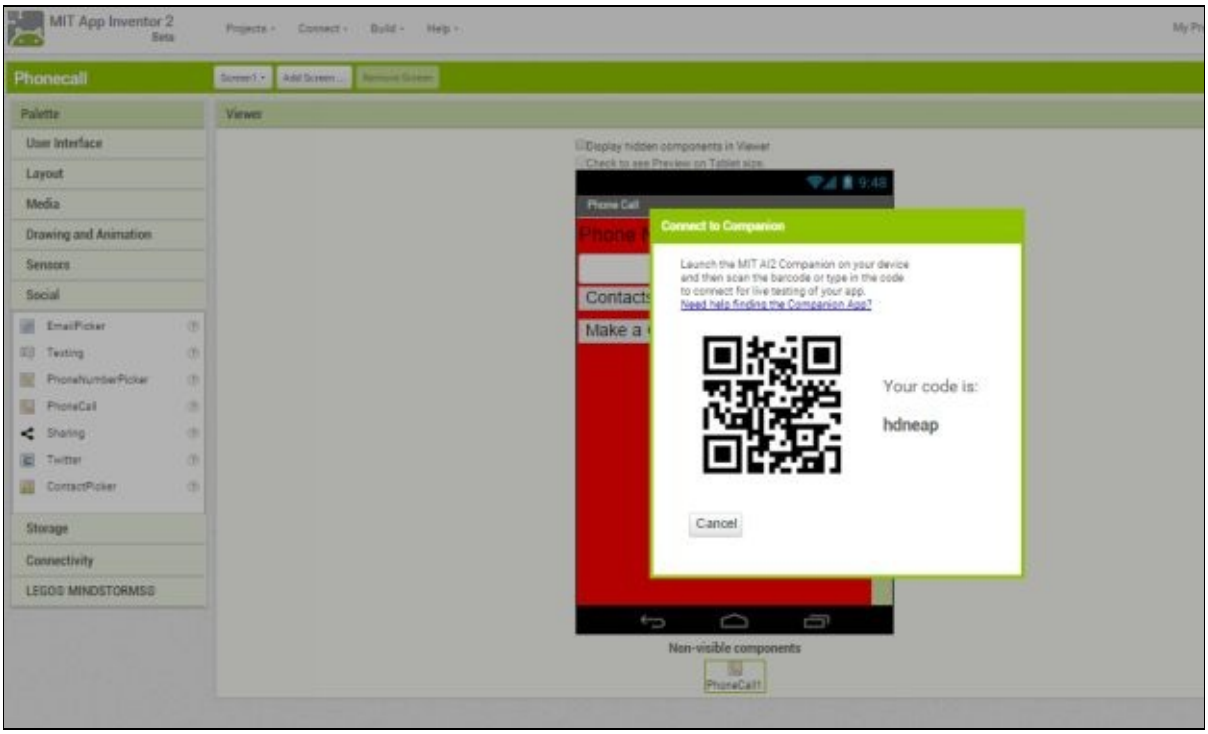


13. Click on PhoneCall1 and select  block. This block will make a call to selected Phone Number.



14. Go to Designer and Click on Connect then select AI Companion.

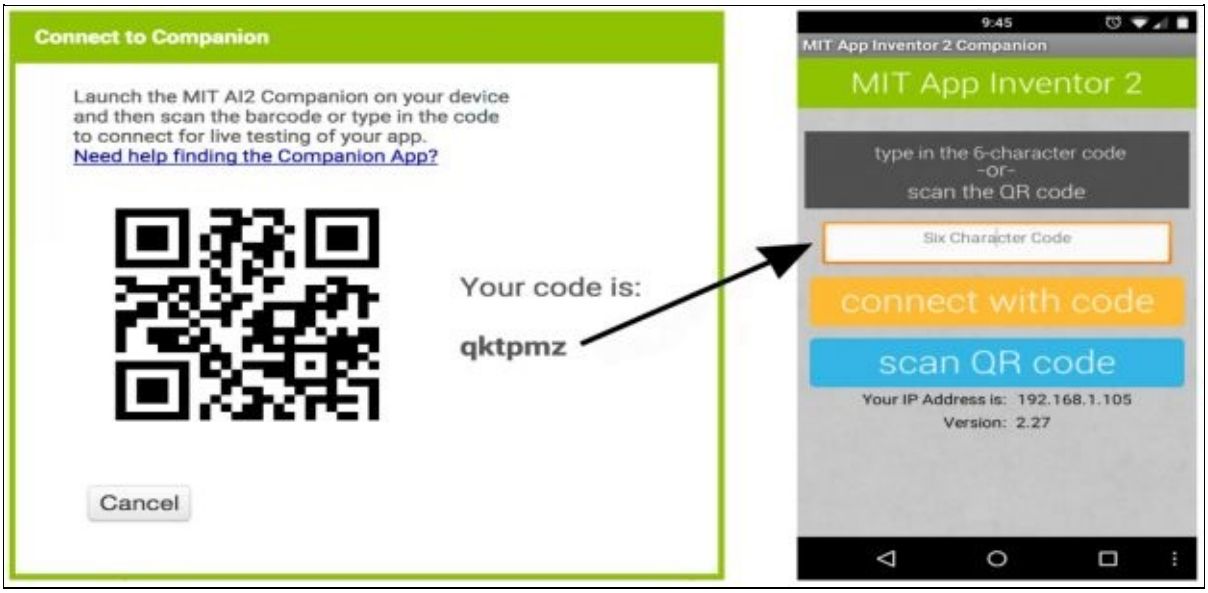




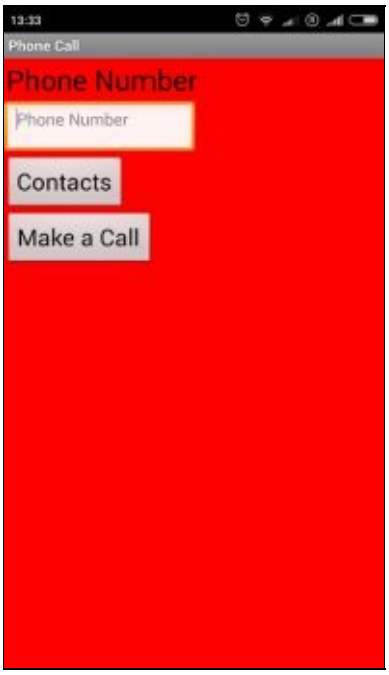
15. Now open MIT App Inventor 2 Companion in your mobile/Tablet.

And enter the code or scan the QR code from your mobile and click on connect with code.

Note:-Your Mobile/Tablet should be connected to same wifi/network to which your computer/Laptop is connected.

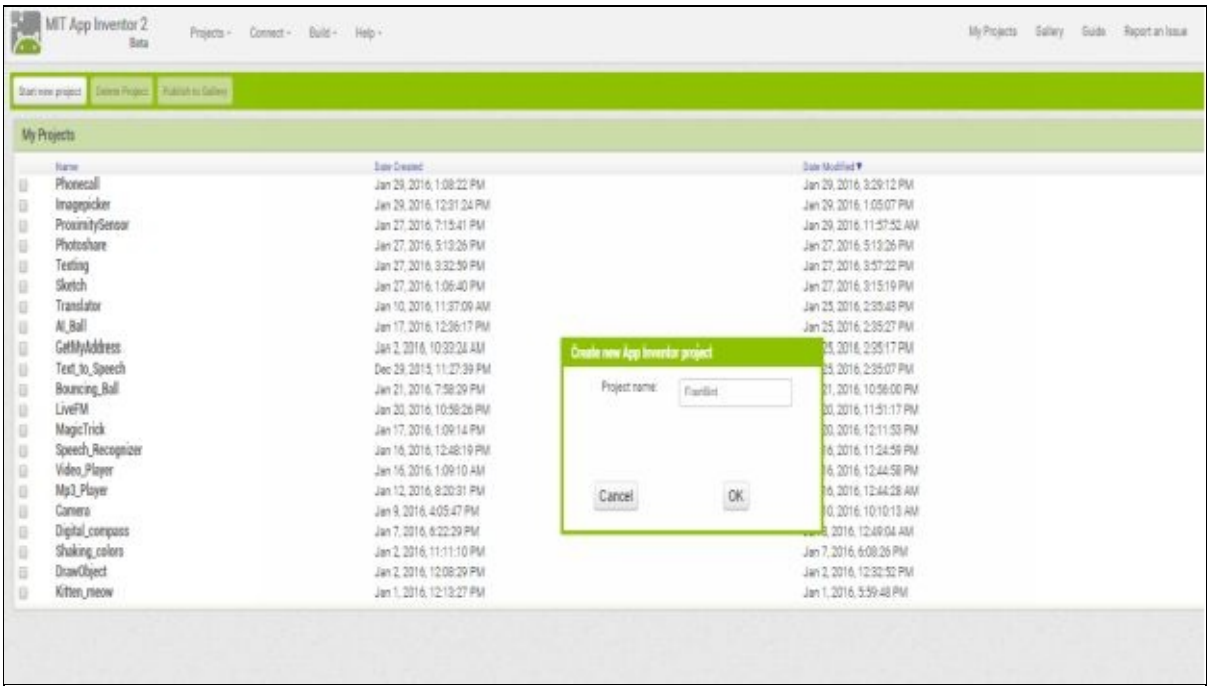


16. Now you should see your app in your phone for live testing. Click on Contacts Button to select a Phone number from your phone contacts and then click on Make a Call Button to call that phone number.

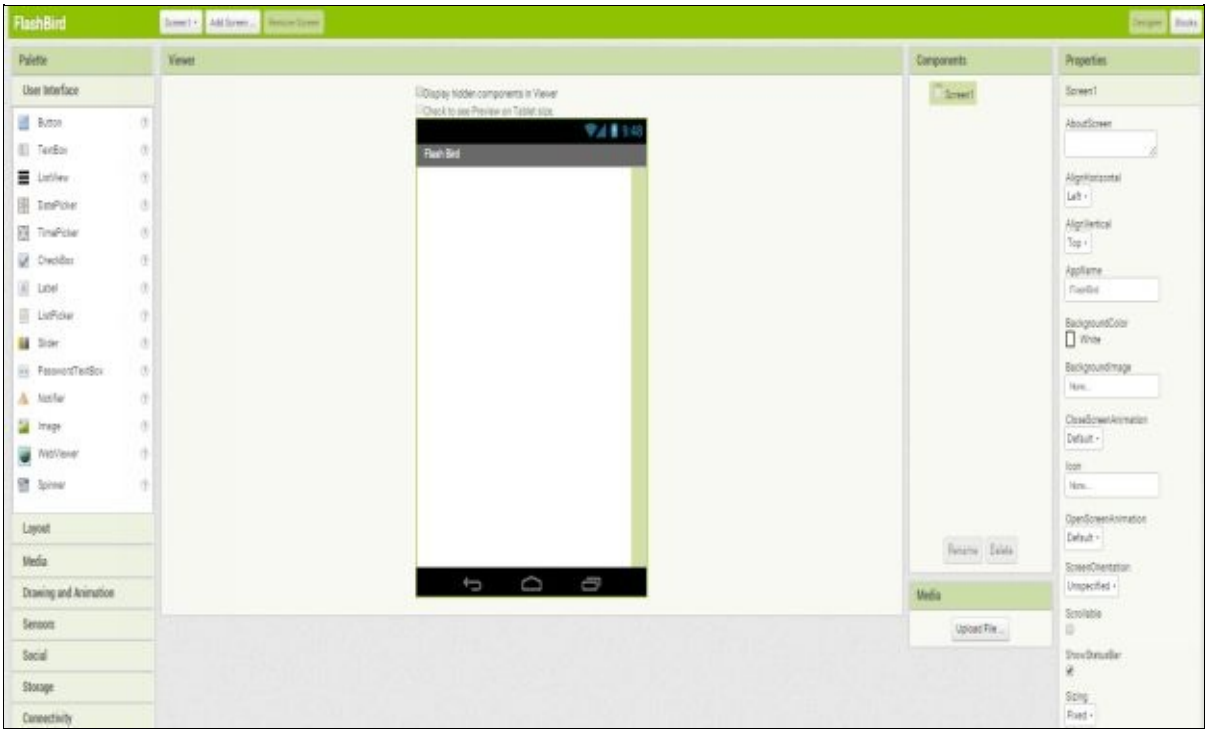


Chapter 25 Flash Bird app

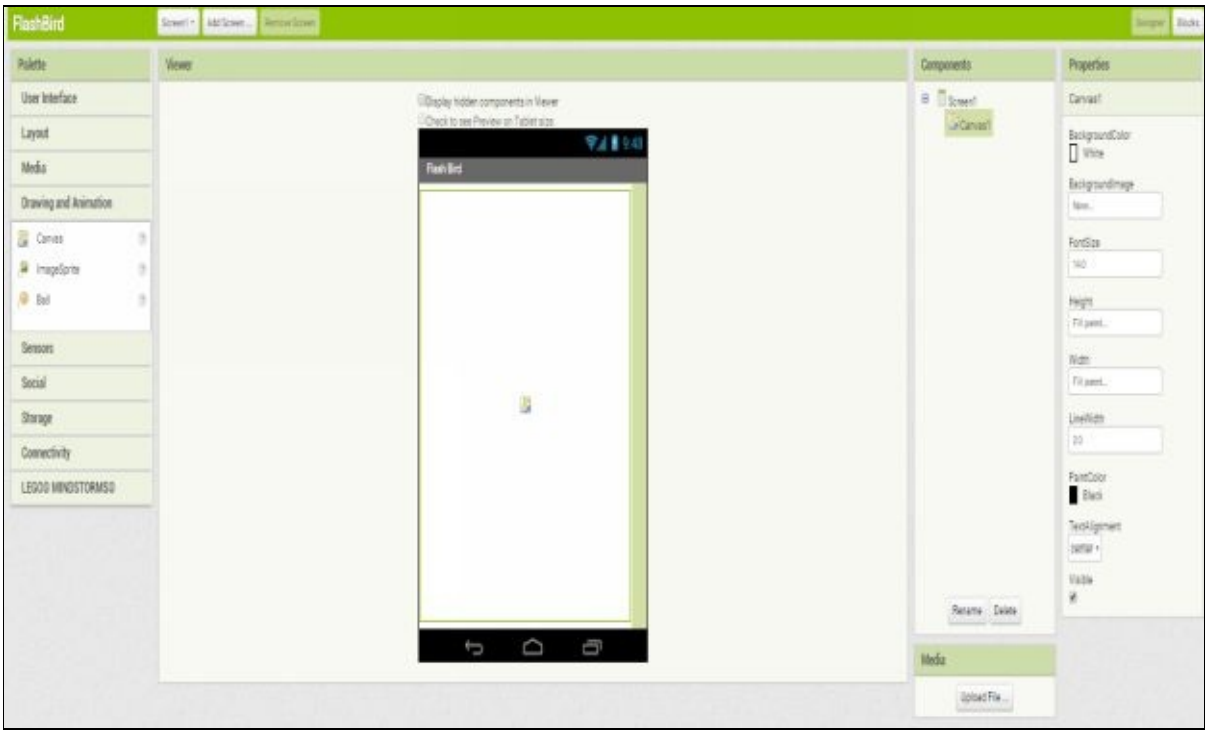
1. Click on Start new project and give it name FlashBird and click OK.



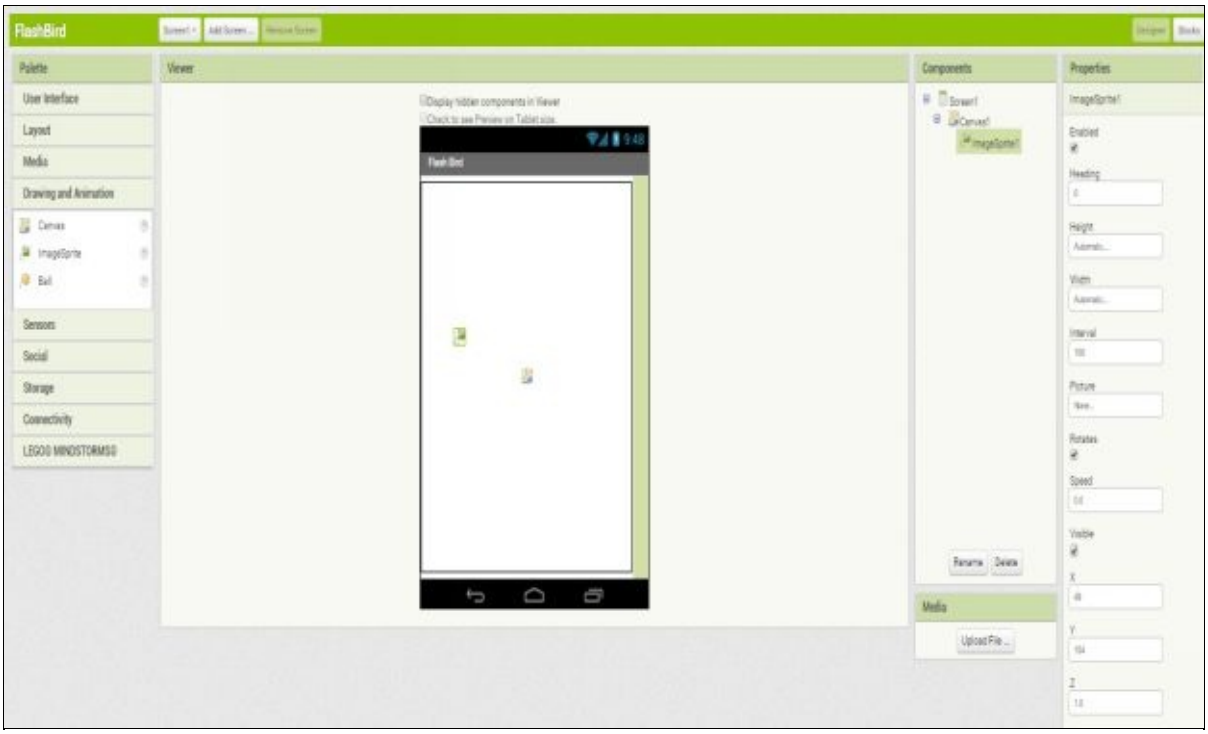
2. Change Screen1 Title to Flash Bird.



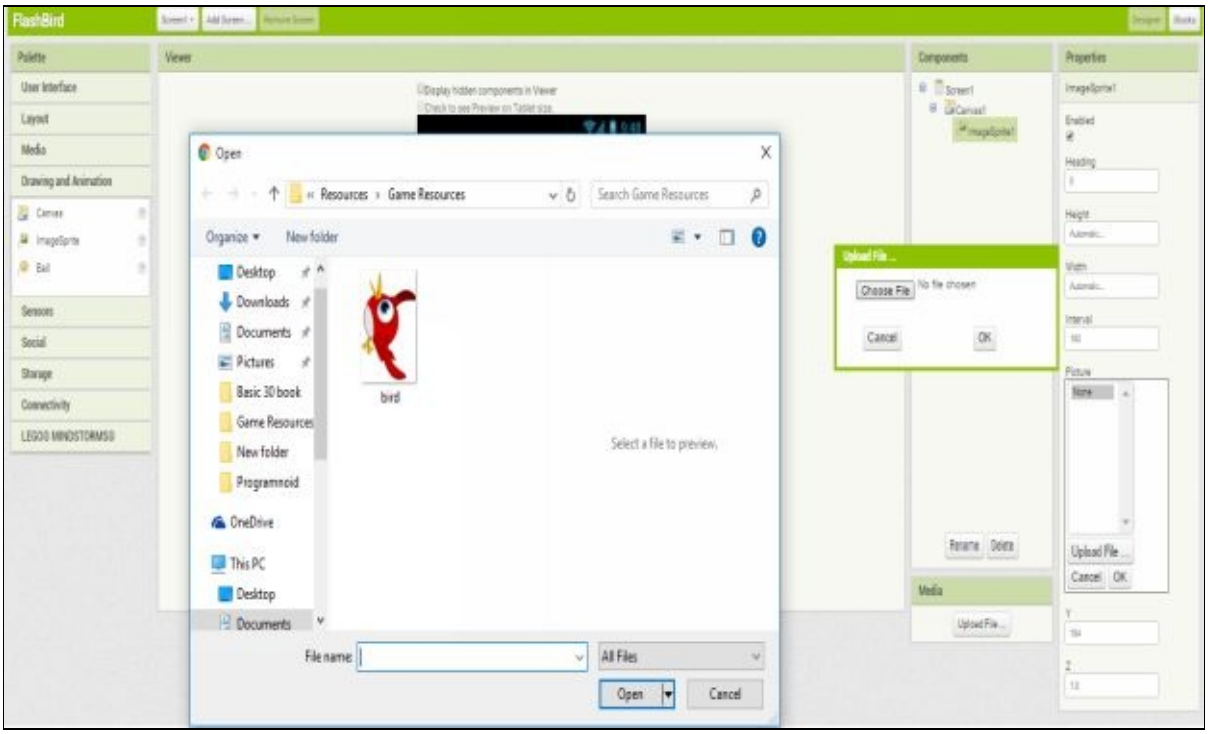
3. Drag a Canvas from Palette to Viewer screen and Change its Height and Width both to Fill parent.

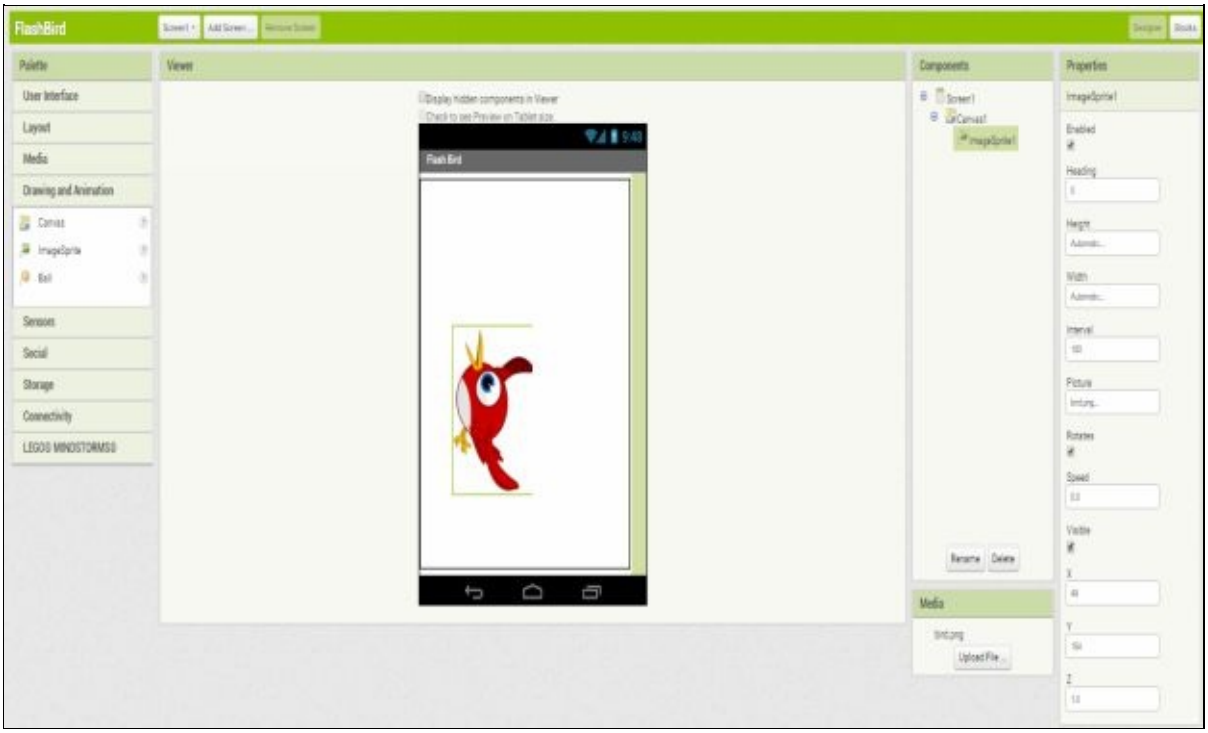


4. Drag an ImageSprite from Palette to Viewer.

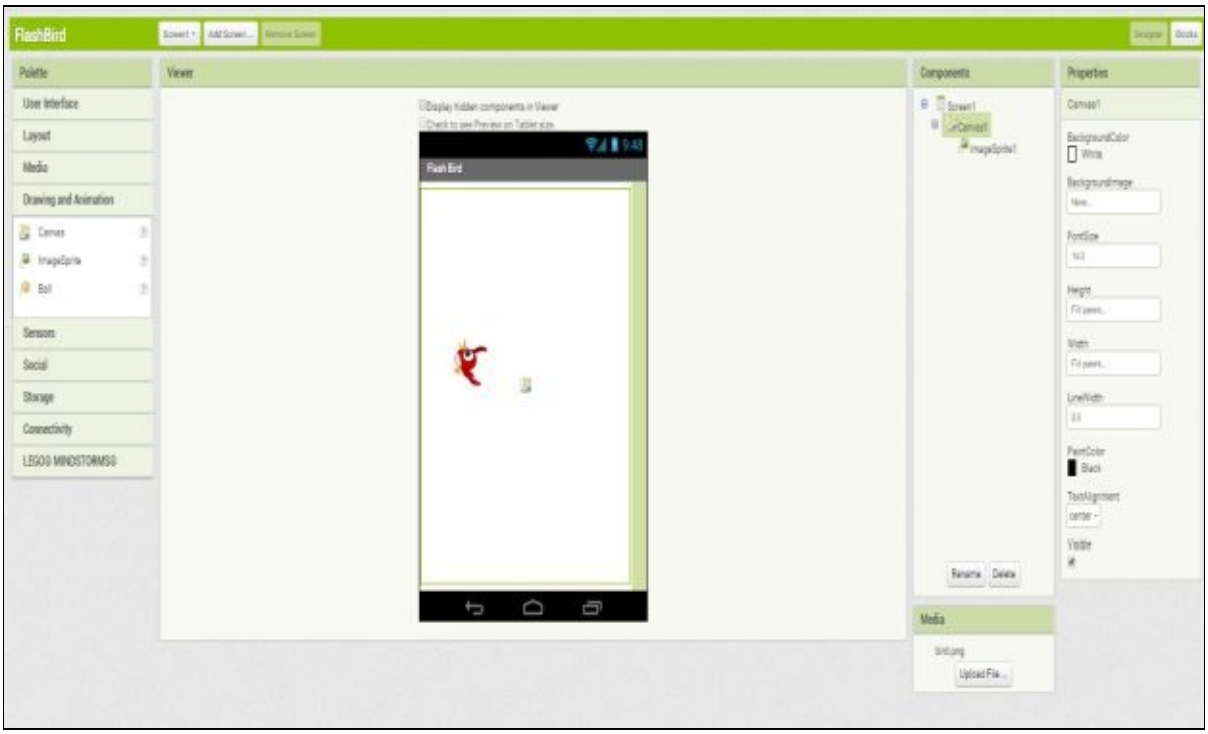


5. Change ImageSprite1's Picture to bird.png.

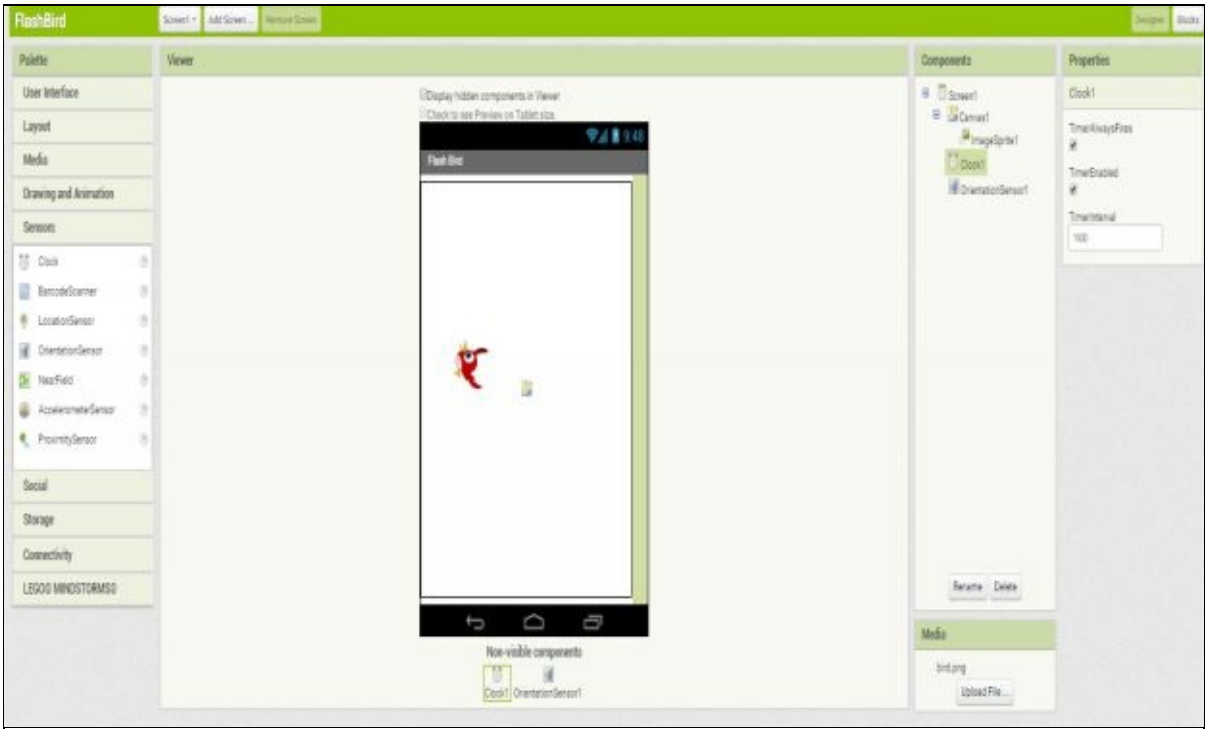




6. Change ImageSprite1's Height and Width both to 50 pixels.

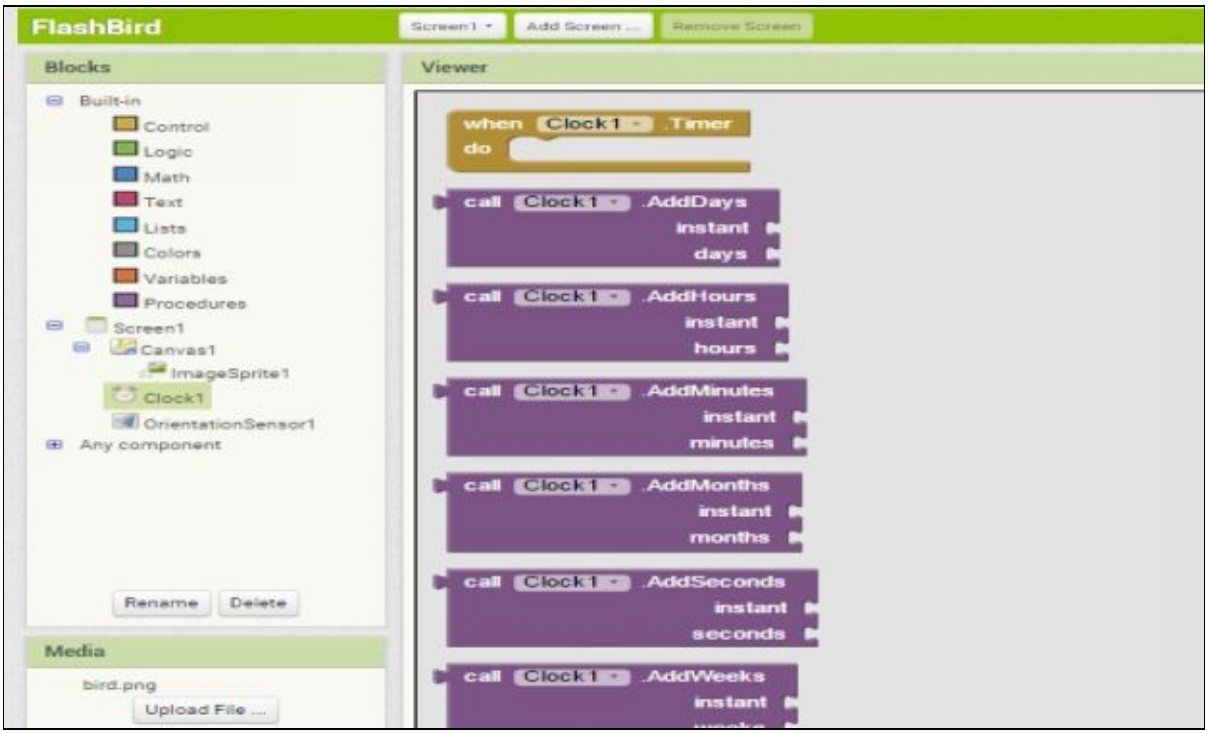


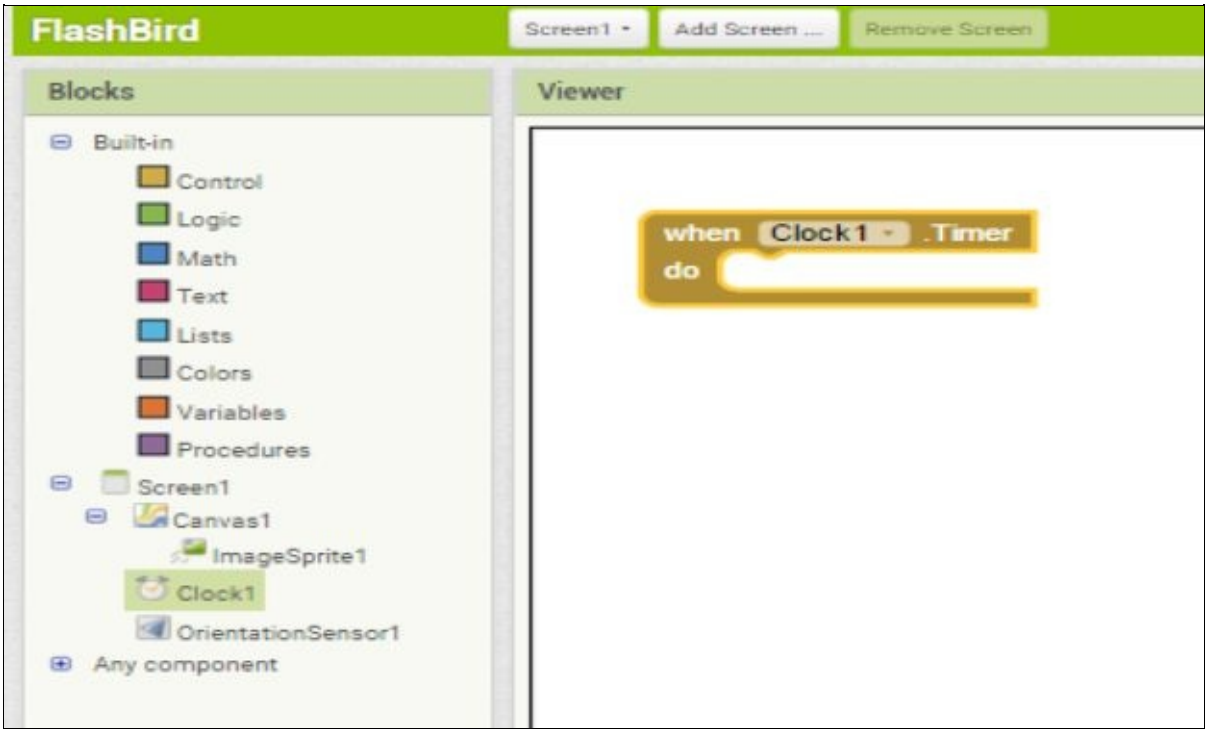
7. Drag an OrientationSensor and Clock from Palette to Viewer and set Clock1's TimerInterval to 10.

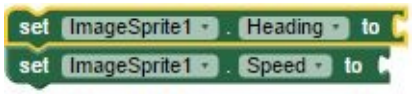


8. Go to Blocks and click on Clock1 and select 

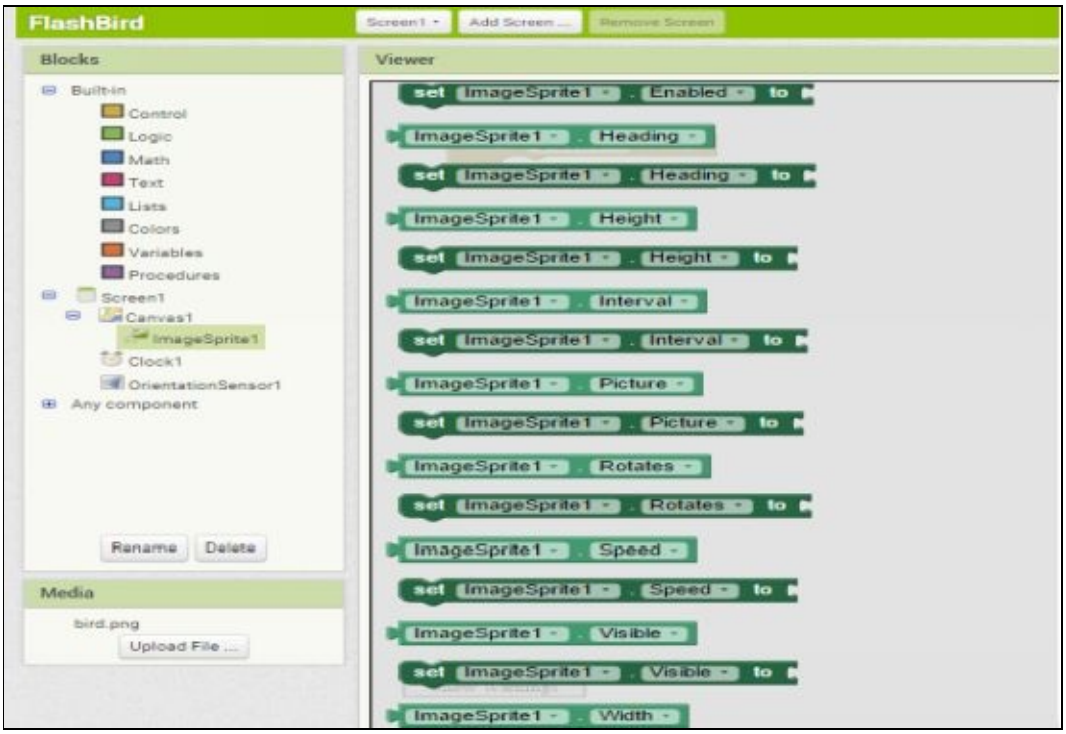
block. This block will decide what to do when Clock Timer completes it's given duration everytime.

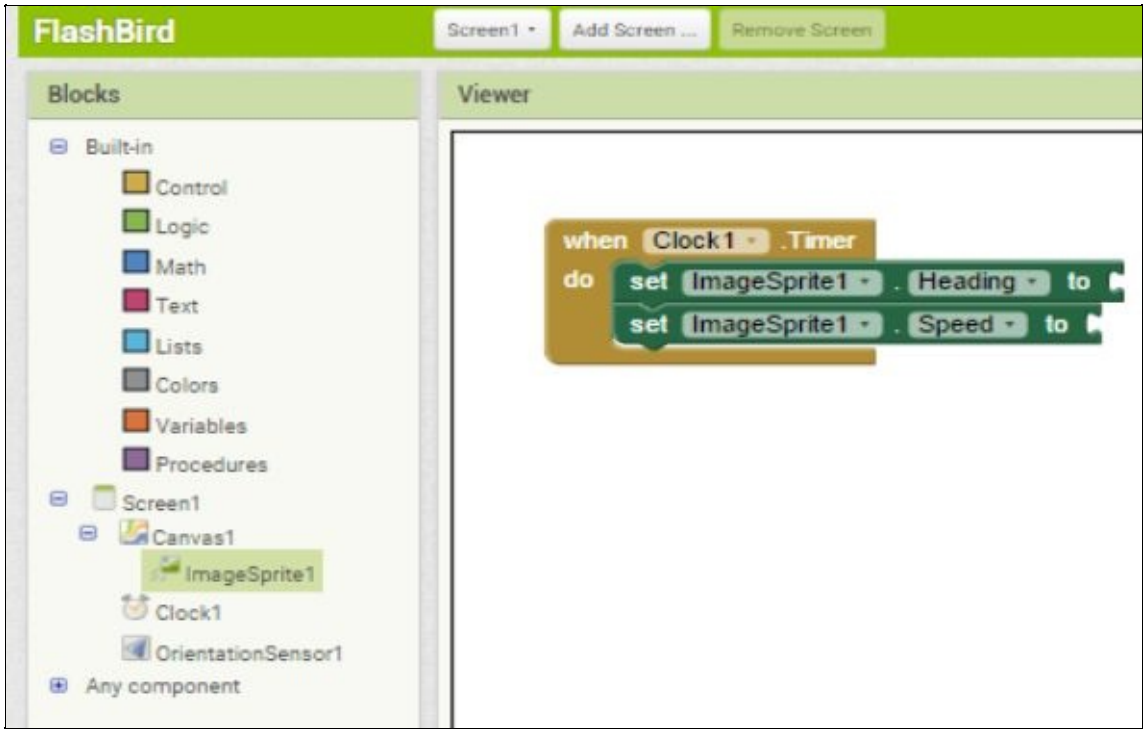


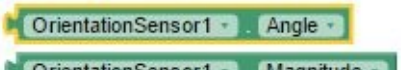


9. Click on ImageSprite1 and scroll down to select 

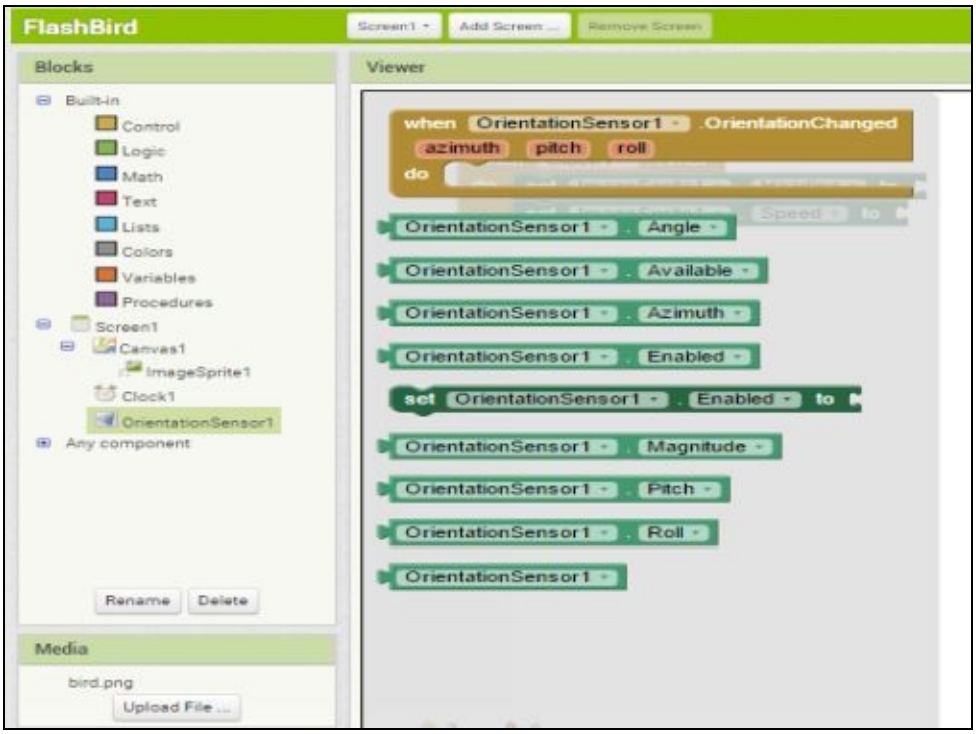
blocks. These blocks will set ImageSprite's speed and direction.

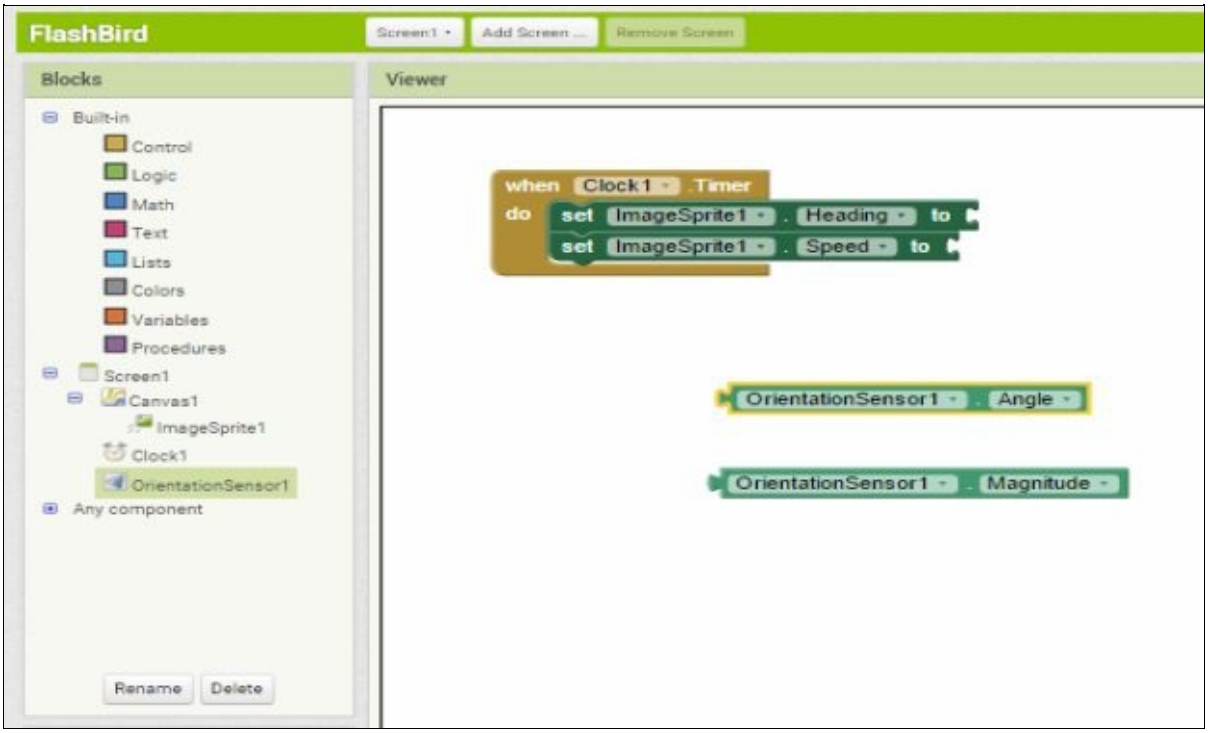




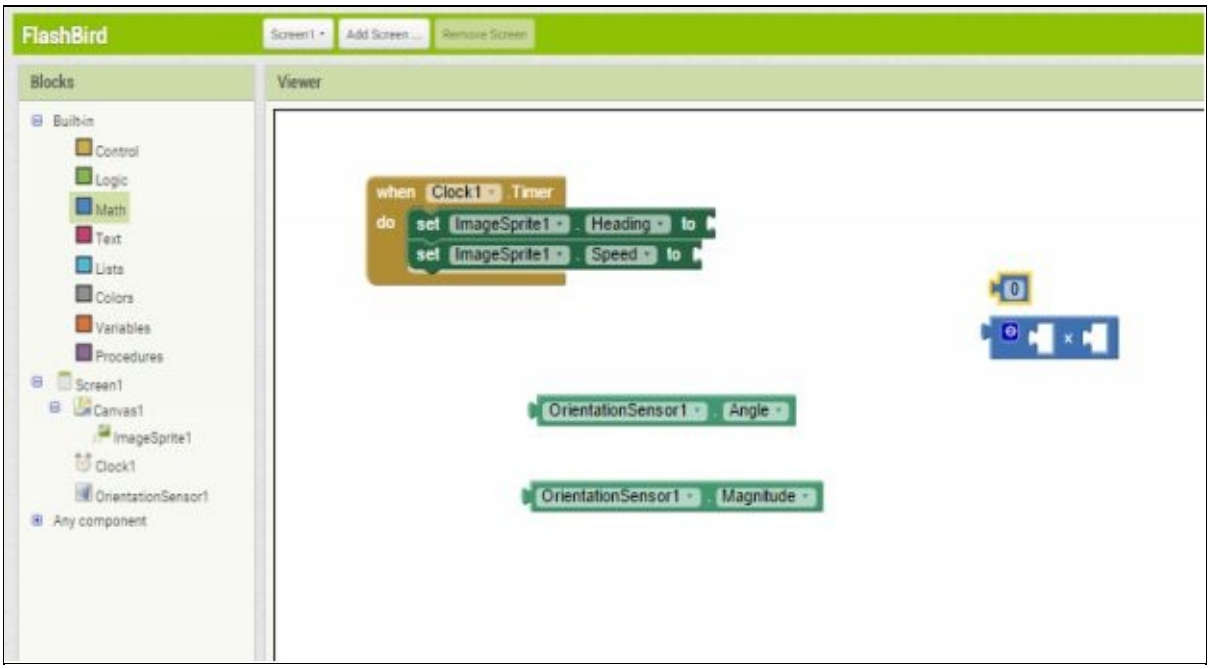
10. Click on OrientationSensor1 and select 

blocks. These blocks will represent angle and magnitude of your Phone's tilt.

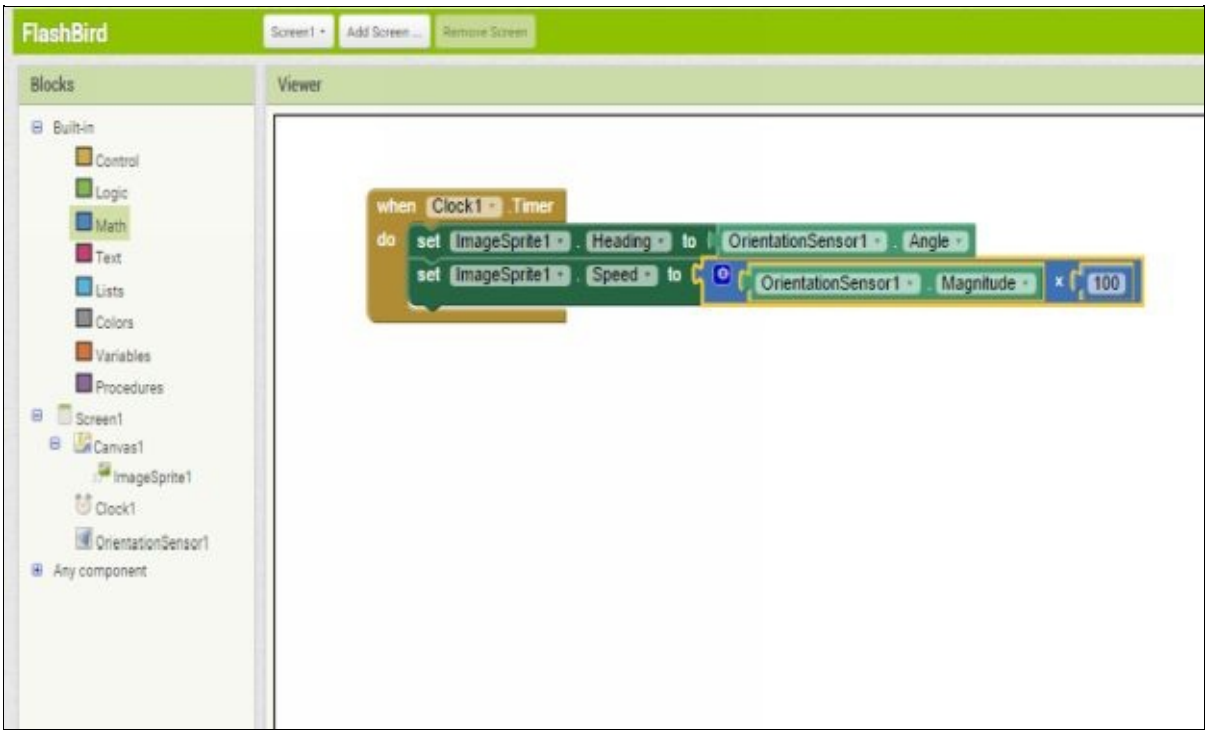




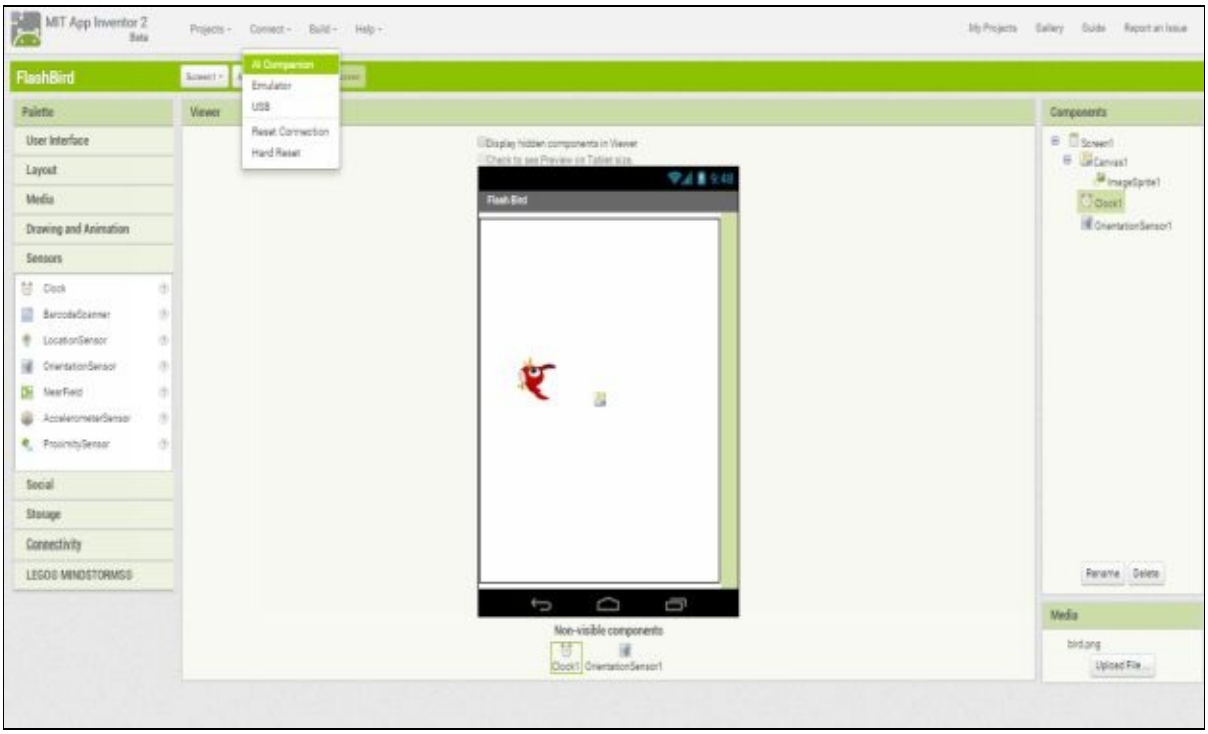
11. Click on Math under Built-in and select  blocks.



12. Attach the blocks as shown. So Everytime Timer duration gets completed app will move ImageSprite to your tilt direction angle and with the speed of tilt's magnitude * 100.



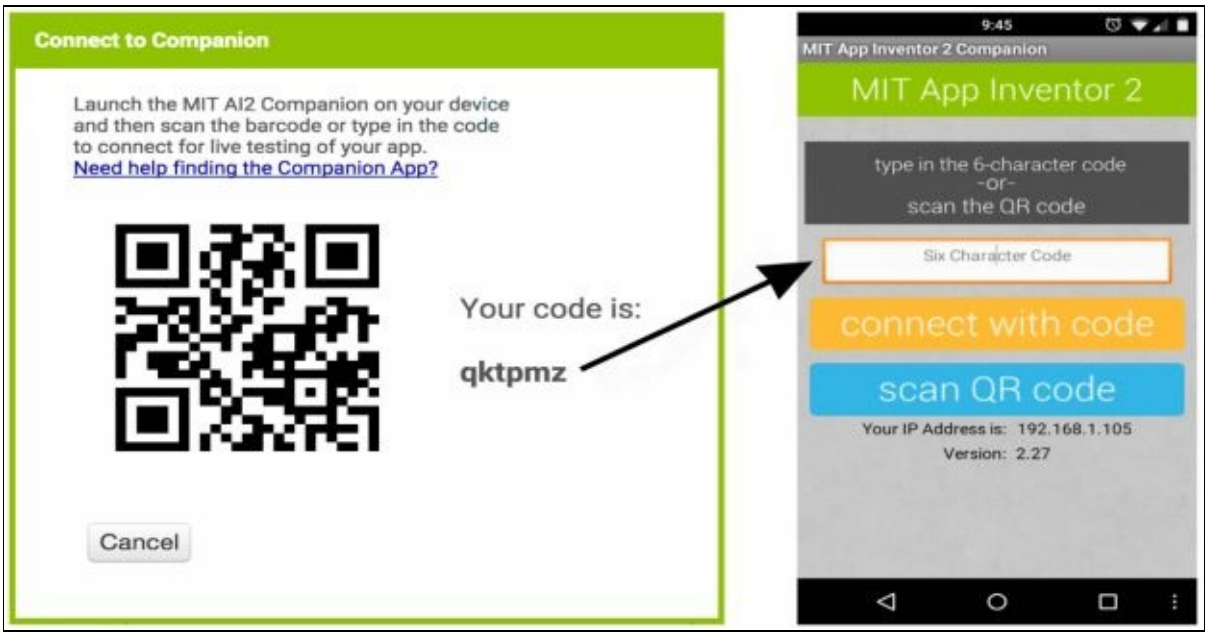
13. Go to Designer and click on Connect then Select AI Companion.



14. Now open MIT App Inventor 2 Companion in your mobile/Tablet.

And enter the code or scan the QR code from your mobile and click on connect with code.

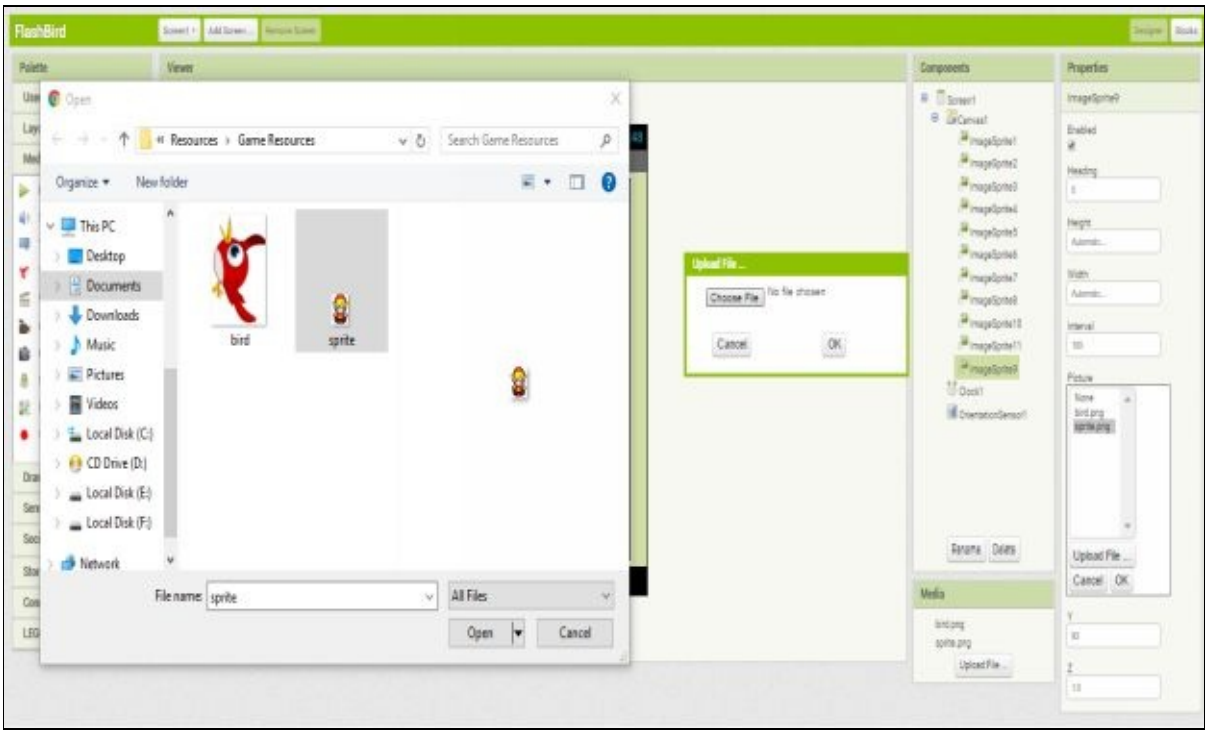
Note:-Your Mobile/Tablet should be connected to same wi-fi/network to which your computer/Laptop is connected.

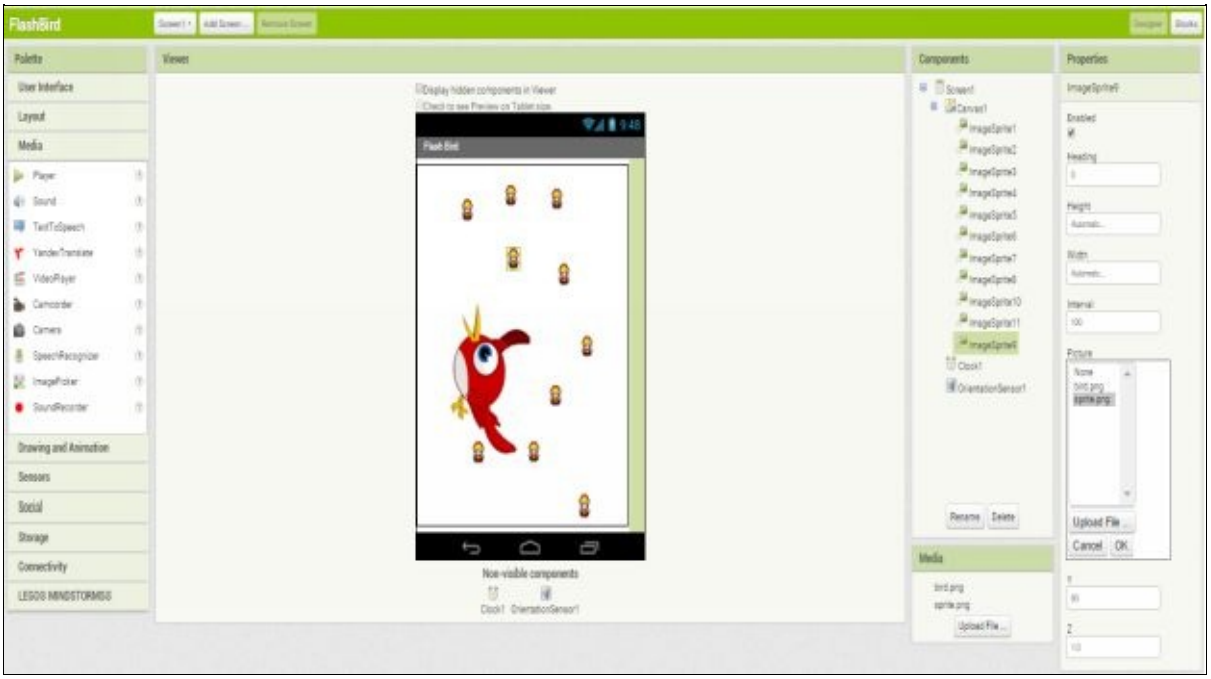


15. Now you should see your app in your phone for live testing. Tilt your phone to move the bird.

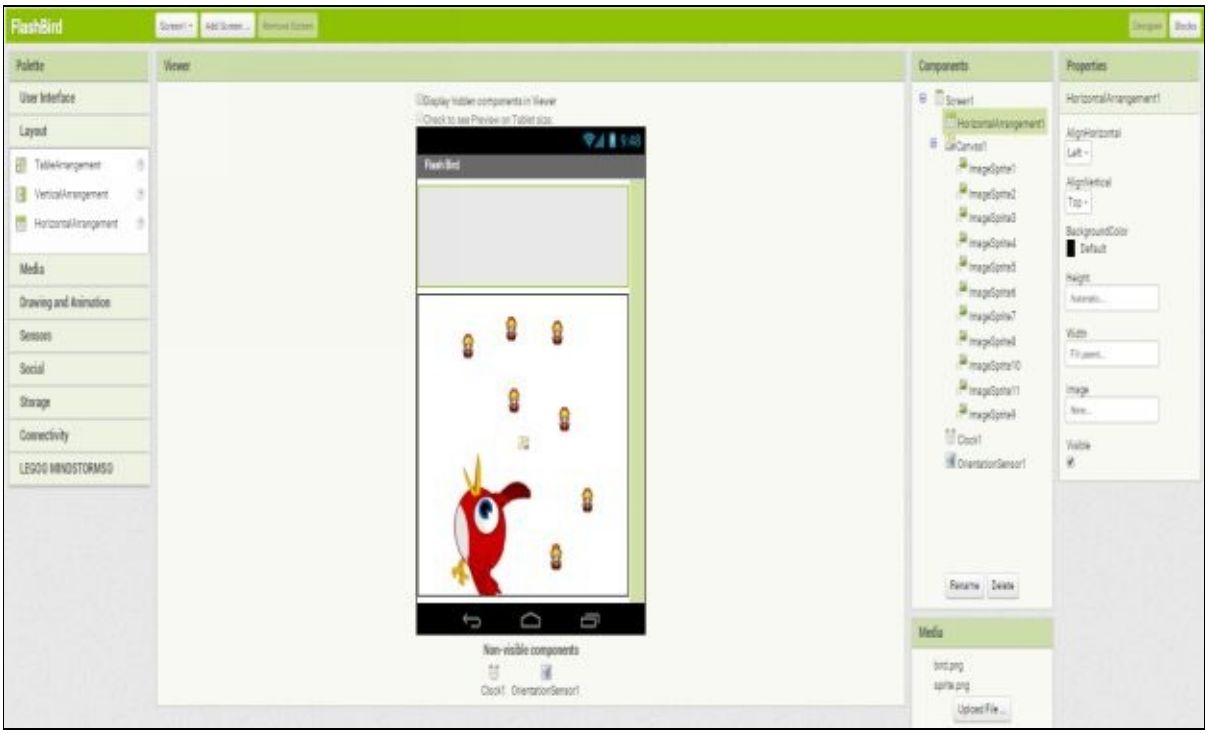
Chapter 26 Flash Bird Game (Extending Previous app)

16. Open your FlashBird app and Drag 10 more ImageSprites from Palette to Viewer screen and set their picture as sprite.png.

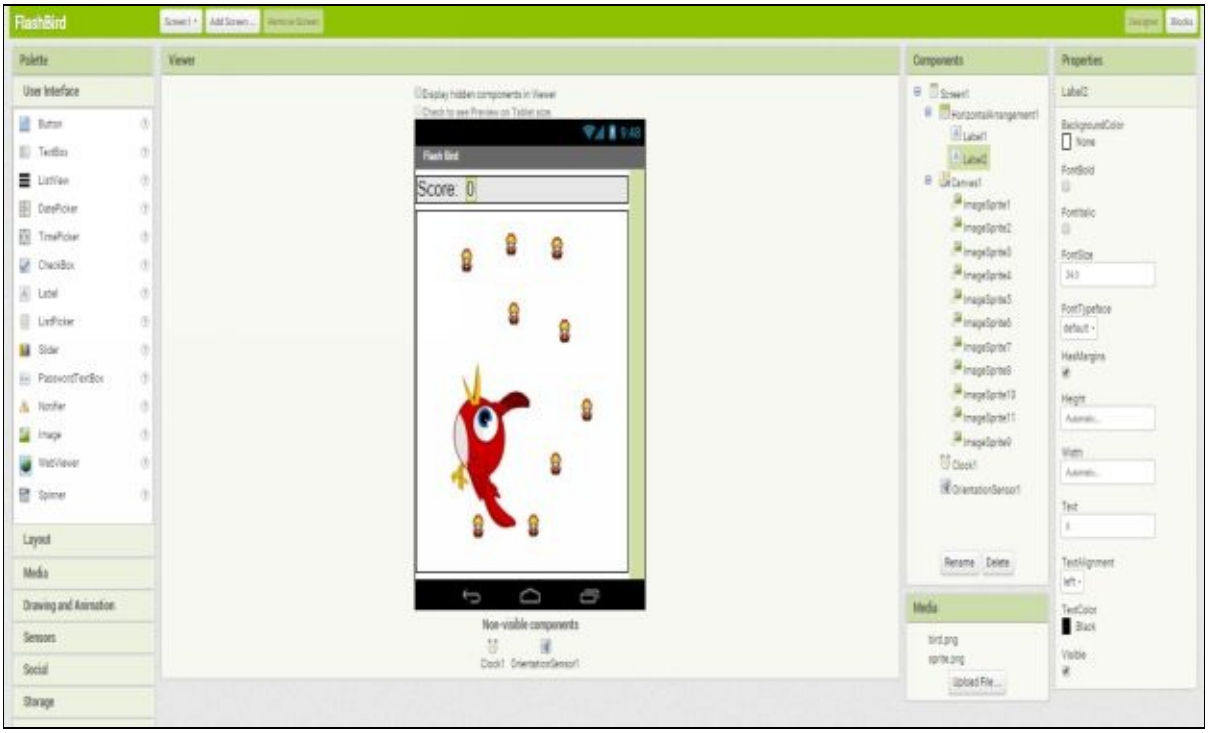




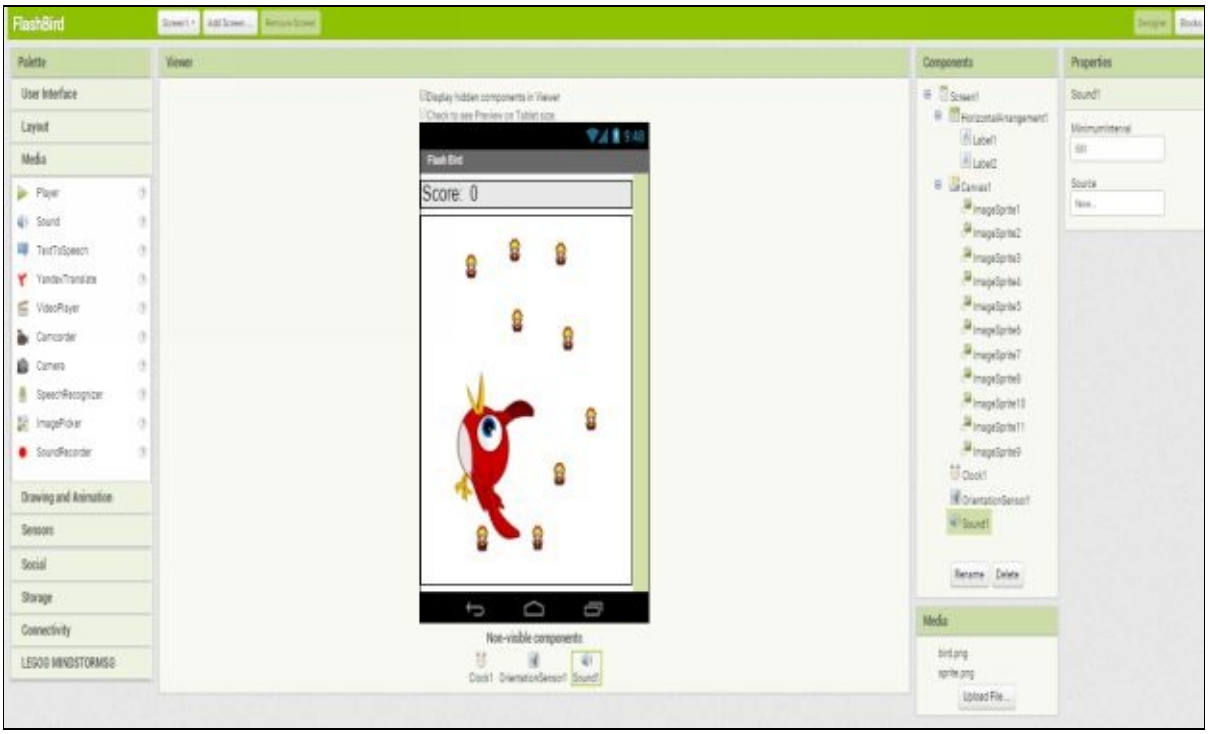
17. Drag a HorizontalArrangement from Palette to Viewer and set its Width to Fill parent.



18. Drag two Labels from Palette to Viewer and set first Label's Text as Score: and Second Label's Text as 0.Also set FontSize to 24.0 for both Labels.

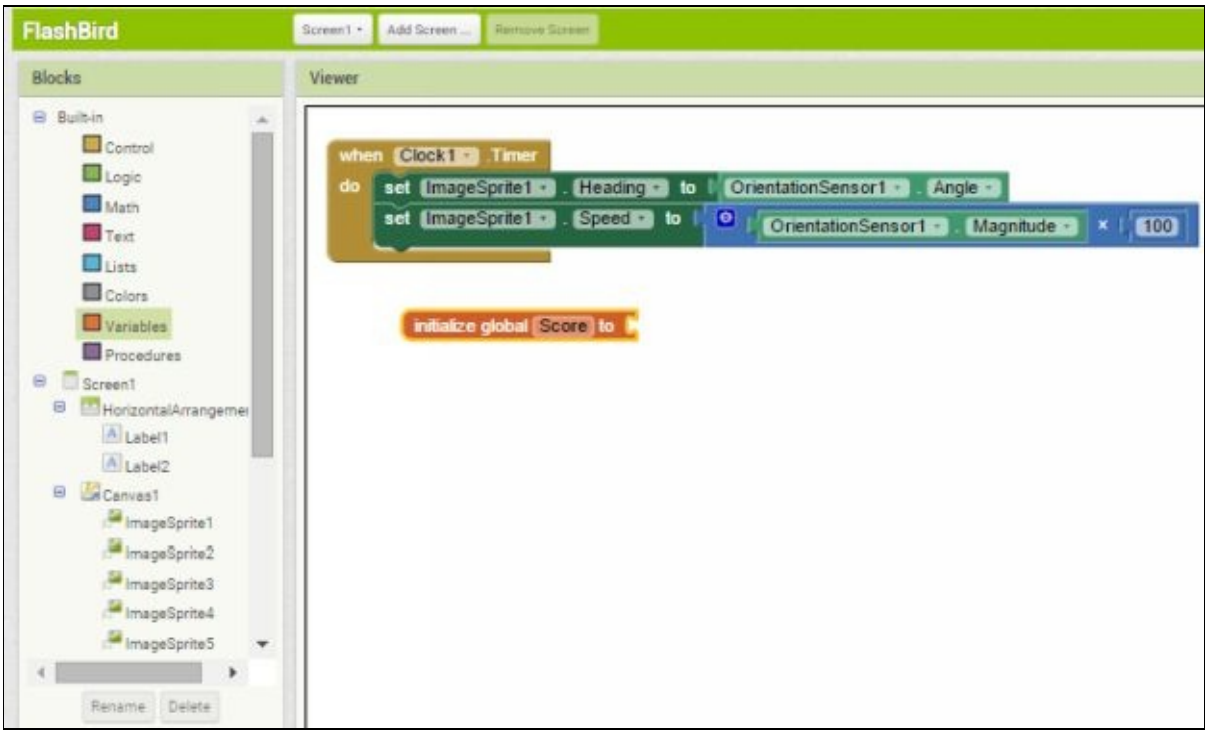


19. Drag a Sound Component from palette to viewer.



20. Go to Blocks and click on Variables under Built-in and select  block.

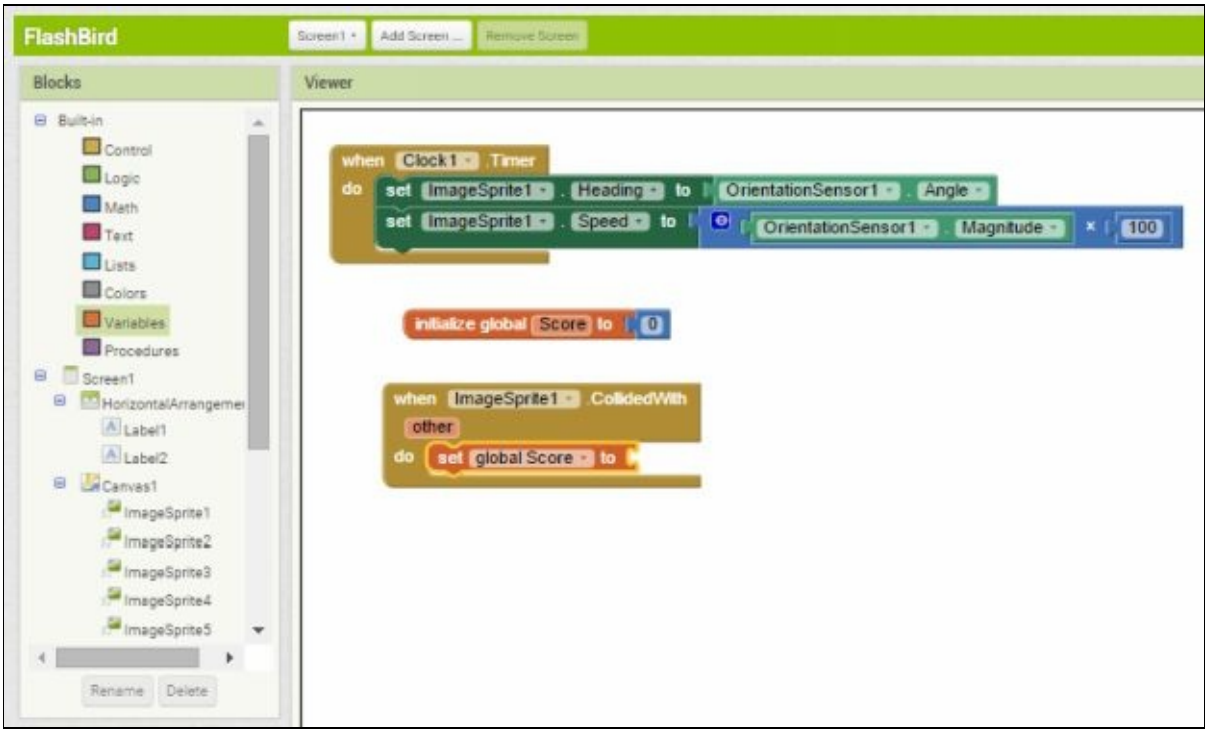
And change name to Score. This is called defining a variable. A variable is a placeholder of a value which can be updated.



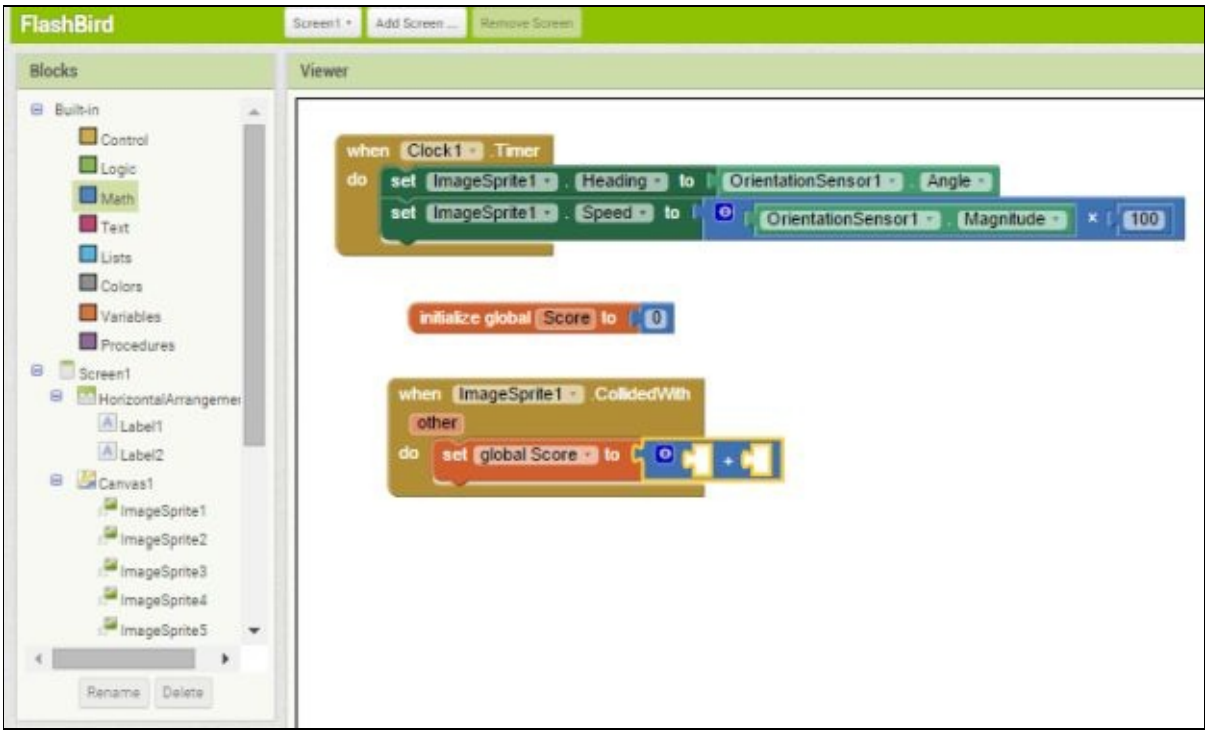
21. Click on Math under Built-in and select  block. Here we are initializing the Score variable to 0.

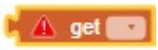
22. Click on ImageSprite1 and select  block. This will identify the Collide event for ImageSprite1 and will decide what to do when ImageSprite1 collides with other ImageSprites/objects in the game.

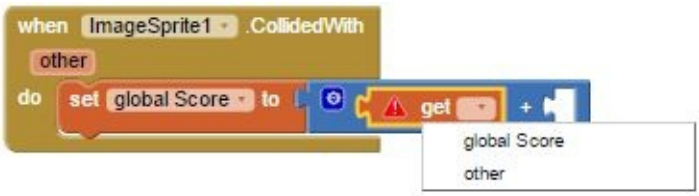
23. Click on Variables and select  block and select global Score as Variable name. This block will set the value for Variable.



24. Click on Math and select  block.

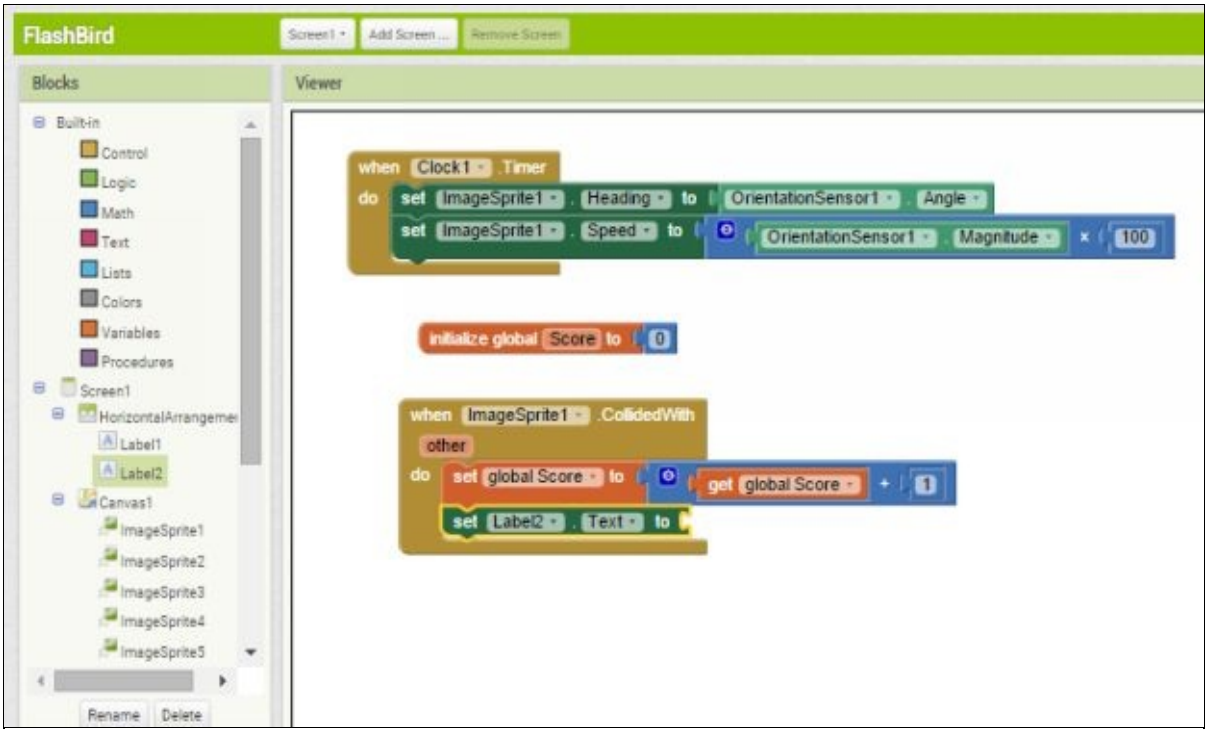


25. Click on Variables and select  block and select global Score as variable. This block will return the current value of Score variable.

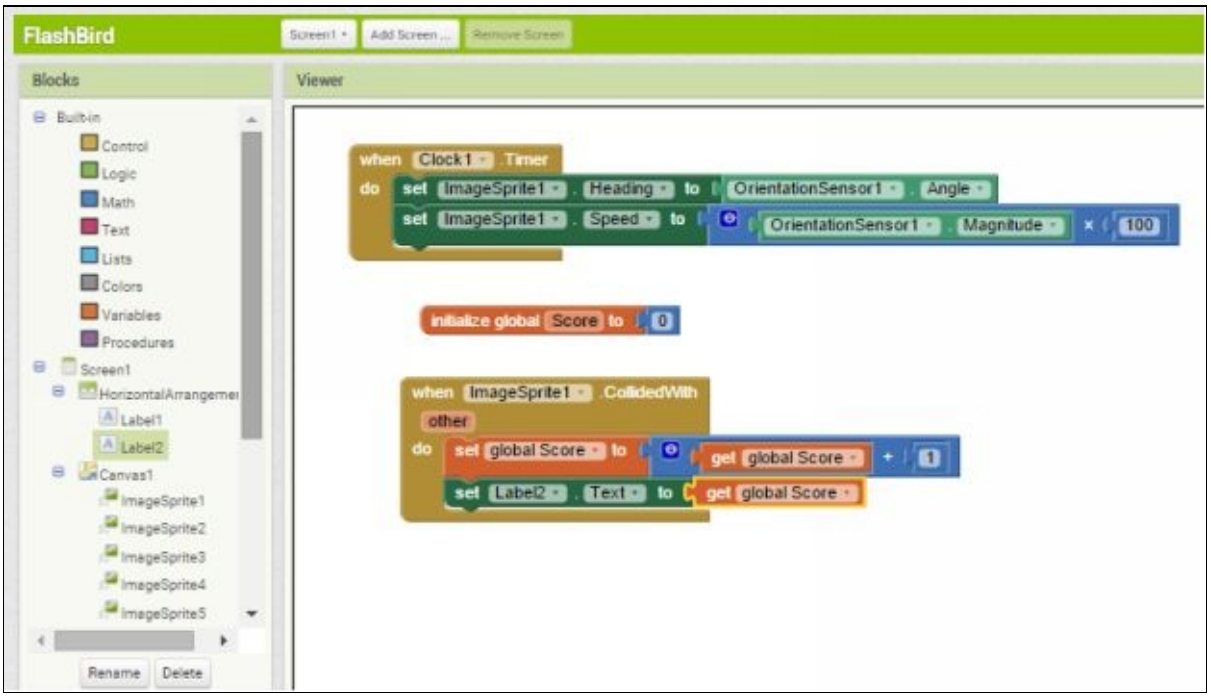


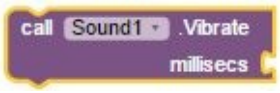
26. Click on Math and select  block and change its value to 1. So that when ImageSprite1 collides with other ImageSprites, your Score variable's value will be increased by 1 more that current value.

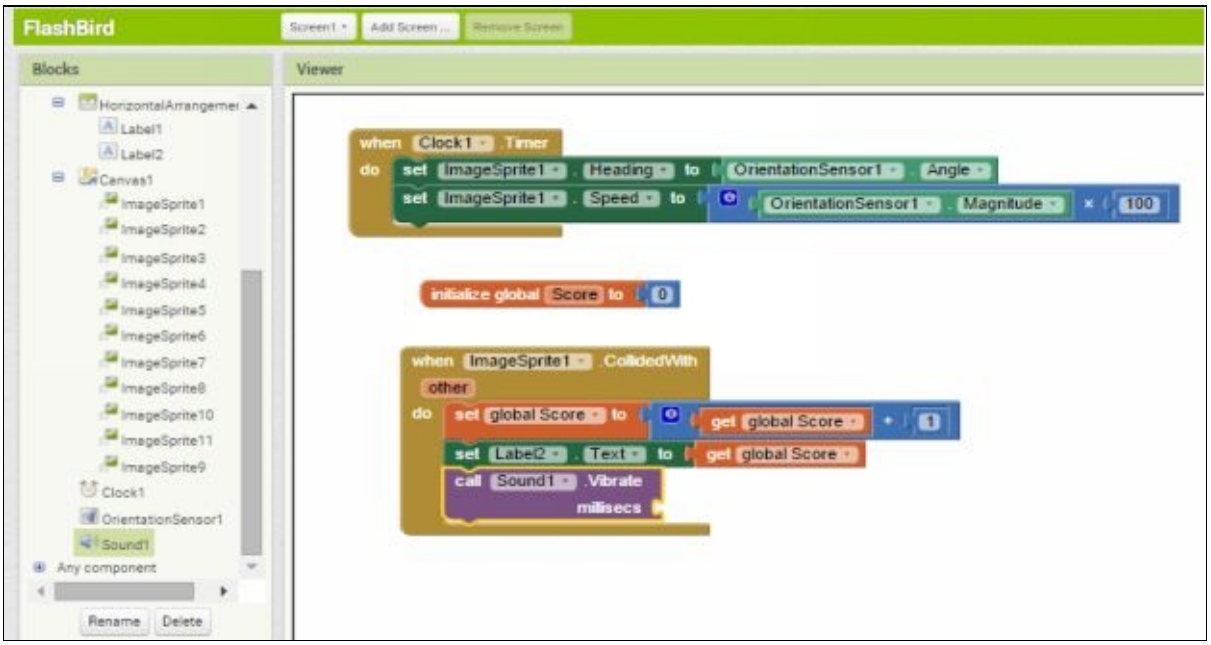
27. Click on Label2 and select  block.



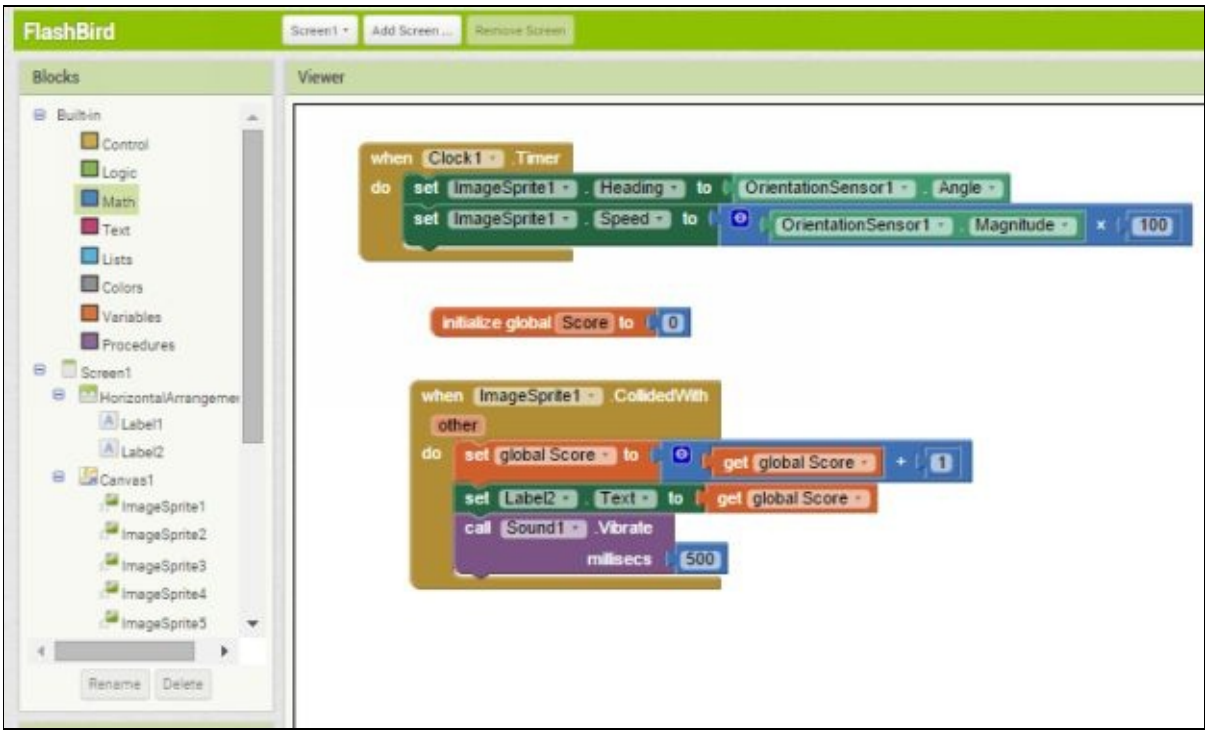
28. Click on  and copy it. This block will return the current value of Score variable. So everytime ImageSprite1 collides with others your Score will be increased by 1 and set it as Label2's Text.



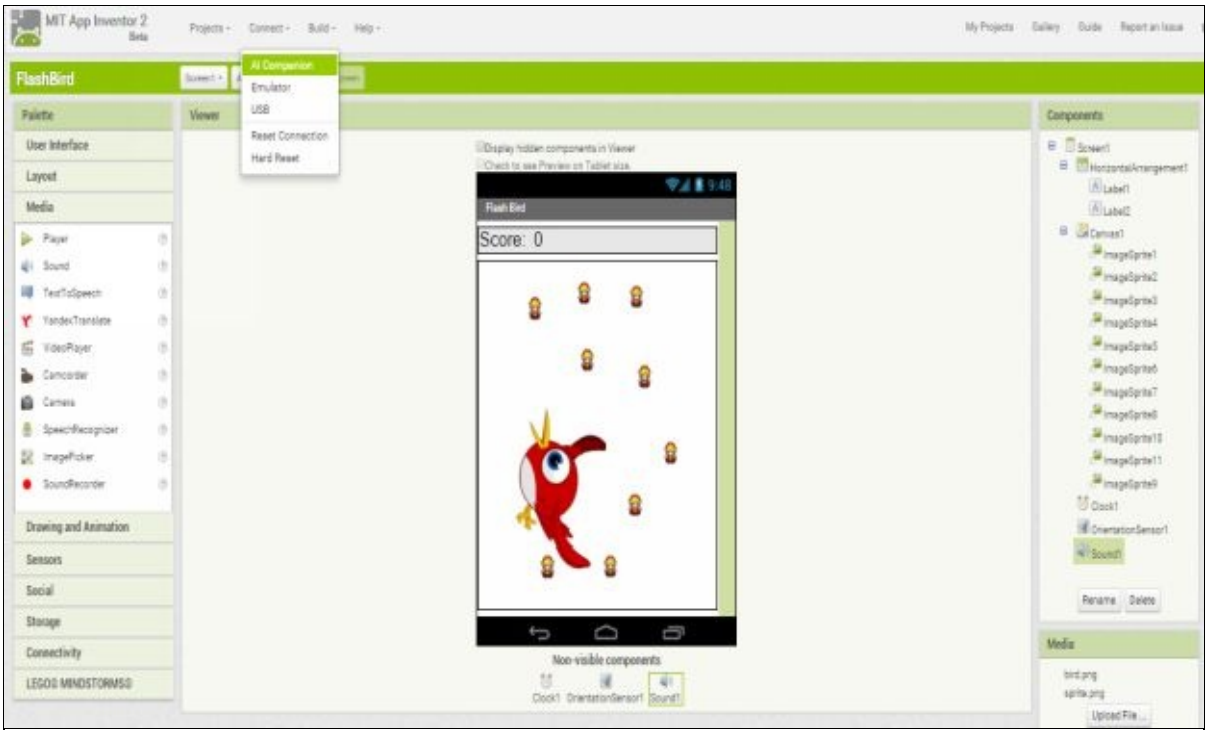
29. Click on Sound1 and select  block. This block will make your phone to vibrate.

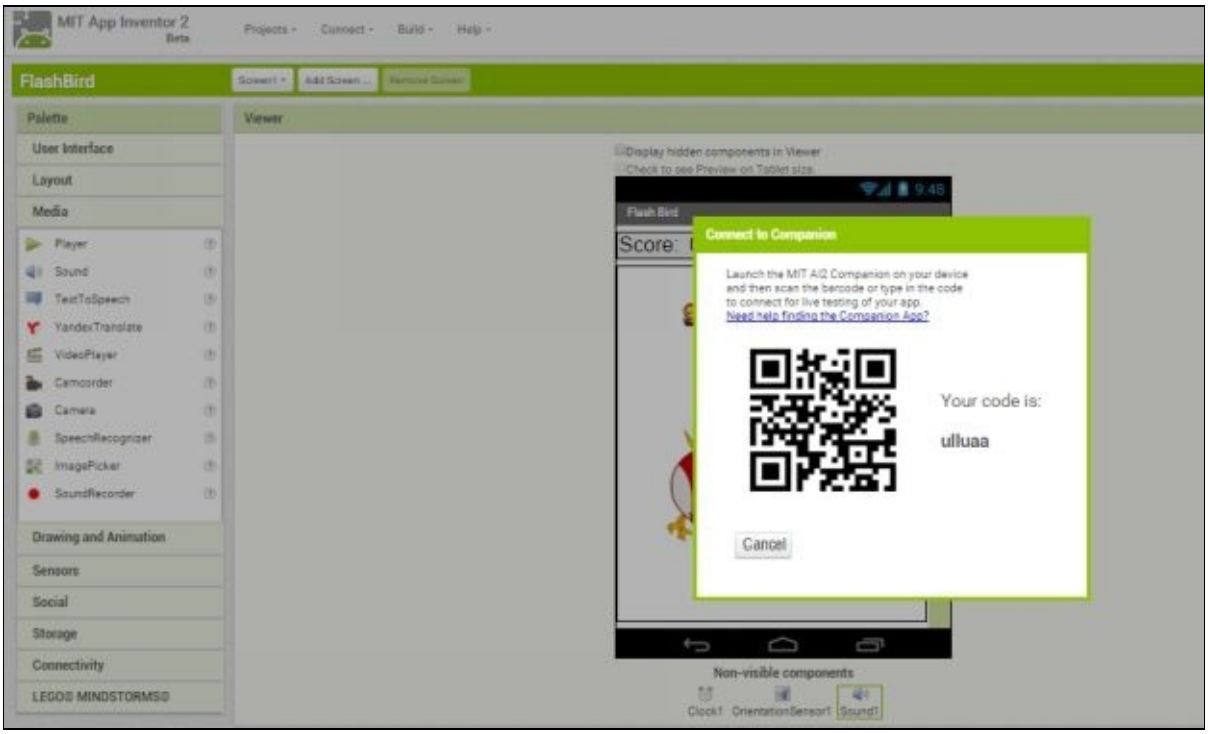


30. Click on Math and select  block and change its value to 500.



31. Go to Designer and click on Connect and then select AI Companion.

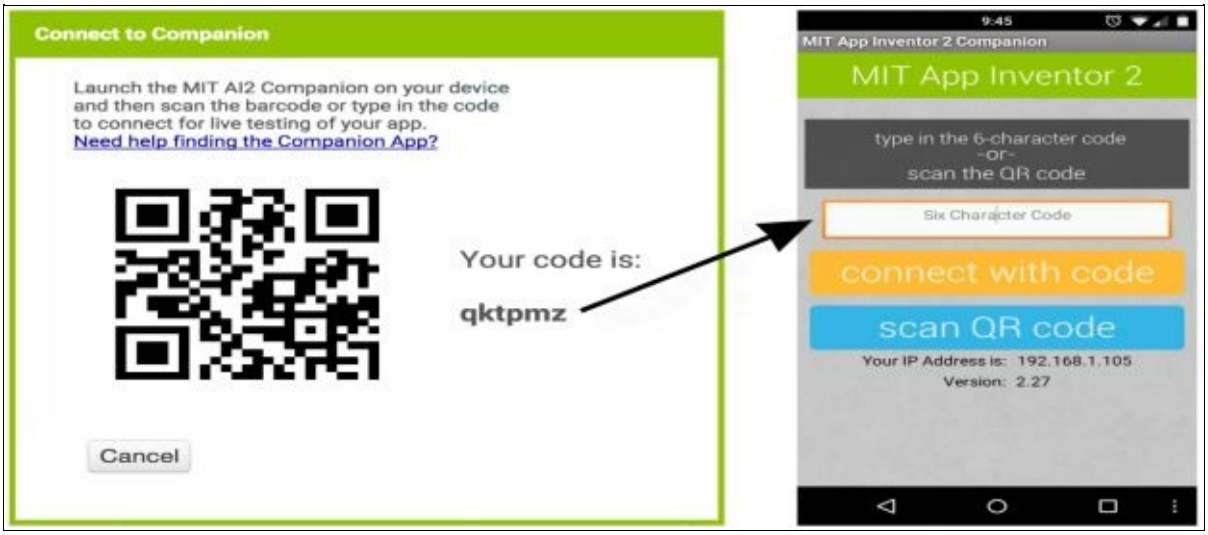




32. Now open MIT App Inventor 2 Companion in your mobile/Tablet.

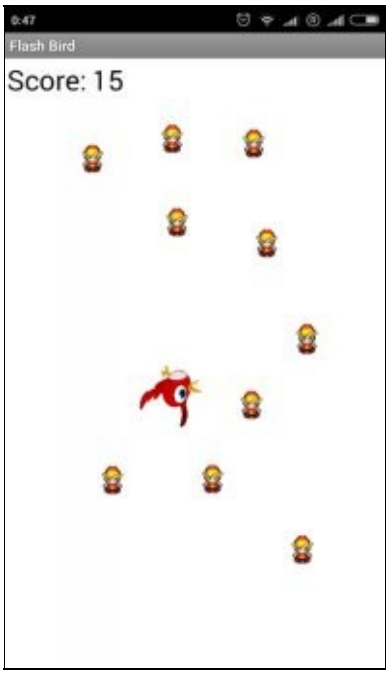
And enter the code or scan the QR code from your mobile and click on connect with code.

Note:-Your Mobile/Tablet should be connected to same wi-fi/network to which your computer/Laptop is connected.



33. Now you should see your app in your phone for live testing. Tilt your phone to move the bird and when bird touches any of the sprite your phone will vibrate and score will increase. This is just a simple way to demonstrate how you can build logic to create games using app inventor.

Try to create some more complex games.



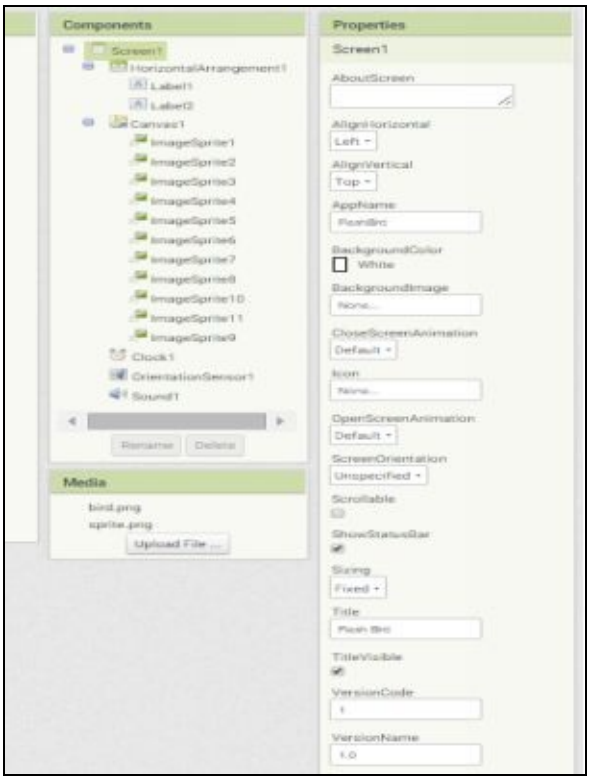
Chapter 27 Packaging & Publishing to Google Play Store

Packaging:-

Applications built with App Inventor can be uploaded to Google Play!

1. Open your app and click on Screen1 and Set following Properties.

- AppName
- VersionCode
- VersionName



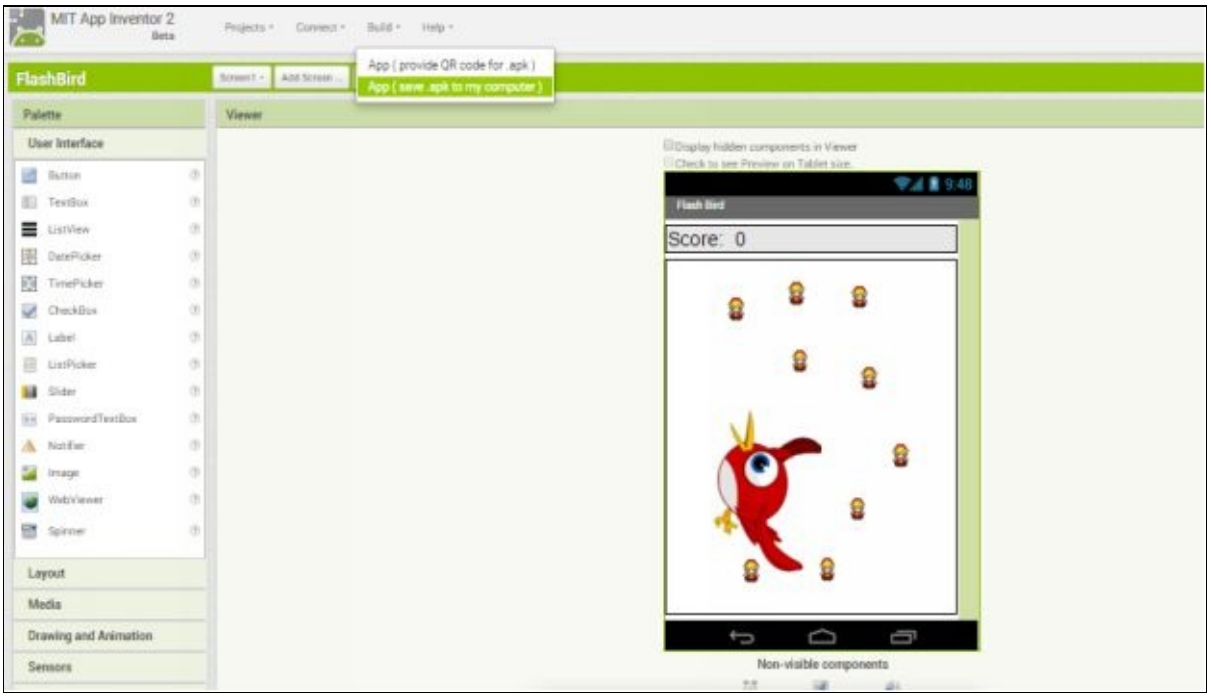
Every app you publish must have a VersionCode and a VersionName. You can set these in the designer under the properties panel for the Screen1 component.

VersionCode is an integer value that will not be visible to Google Play Store users. It is used by other apps to check if your app has been upgraded or downgraded. It defaults to 1 and should be increased by one with every successive change whether it is a major change or a minor change.

VersionName is a String which can be anything you would like. It is defaulted to 1.0. A common structure is a decimal number which is increased by 1 for every major change and 0.1 for every minor change. For example, an initial VersionName could be 1.0 which can be updated to 1.1 after a small change and 2.0 after a larger change.

You will need to increase the VersionCode and change the VersionName of your application when you upload a new version to the Play Store.

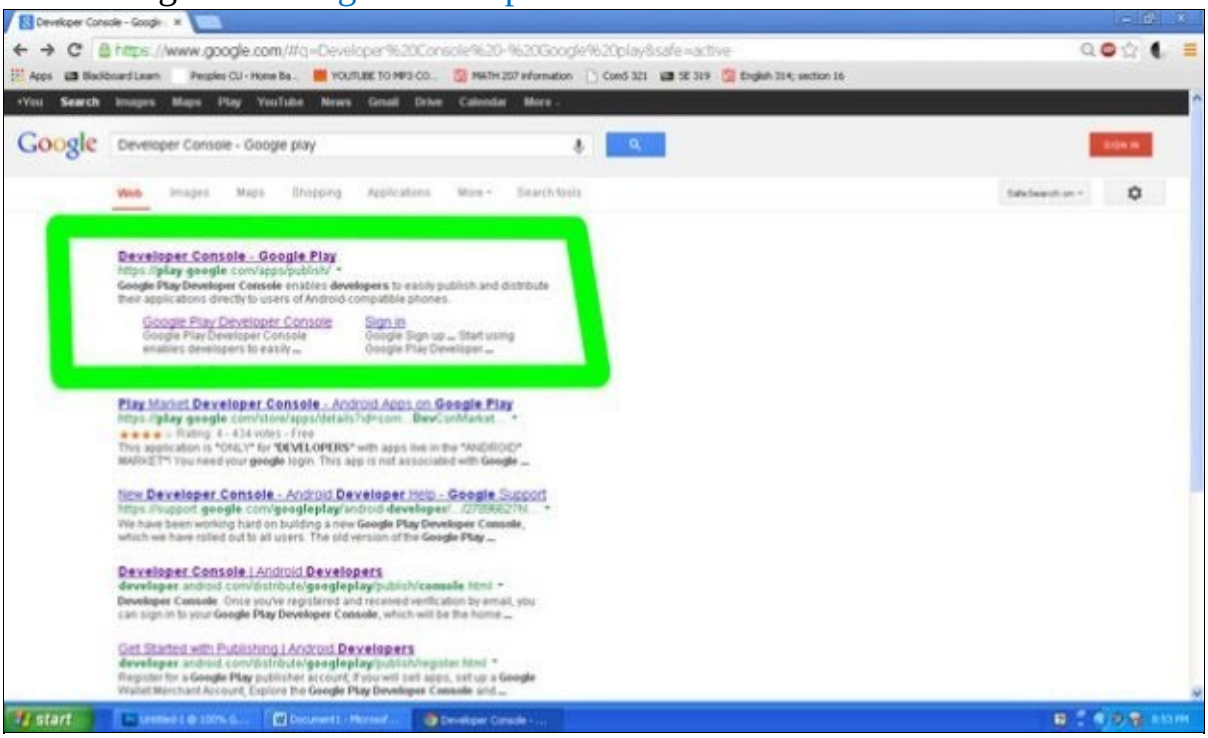
2. Click on Build and then click on App (save .apk to my computer).



This will prompt you to save the application somewhere. Once you have the .apk downloaded, you are ready to begin the publishing process.

Publishing Apps to Google Play:-

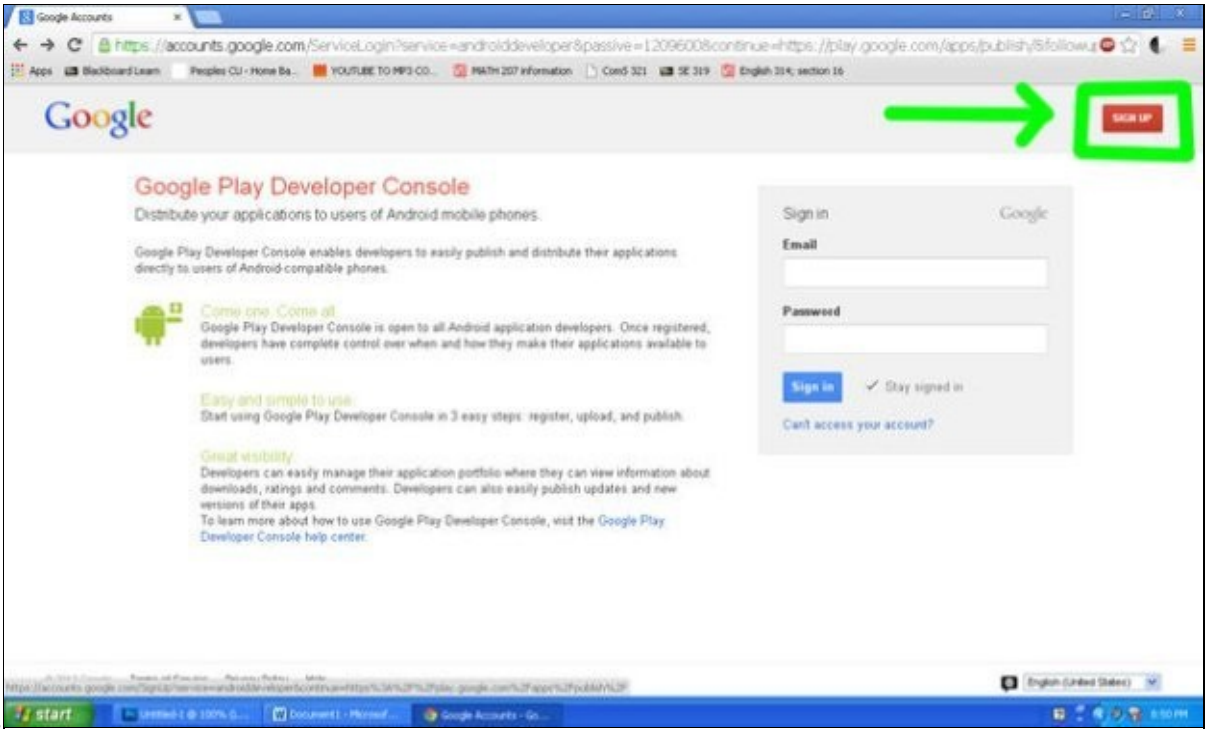
1. Navigate to Google Developer Site.



Open up a web browser, search for "Developer Console - Google Play," and navigate to the site with that as its name.

The proper site is shown in the green box in the figure attached to this step.

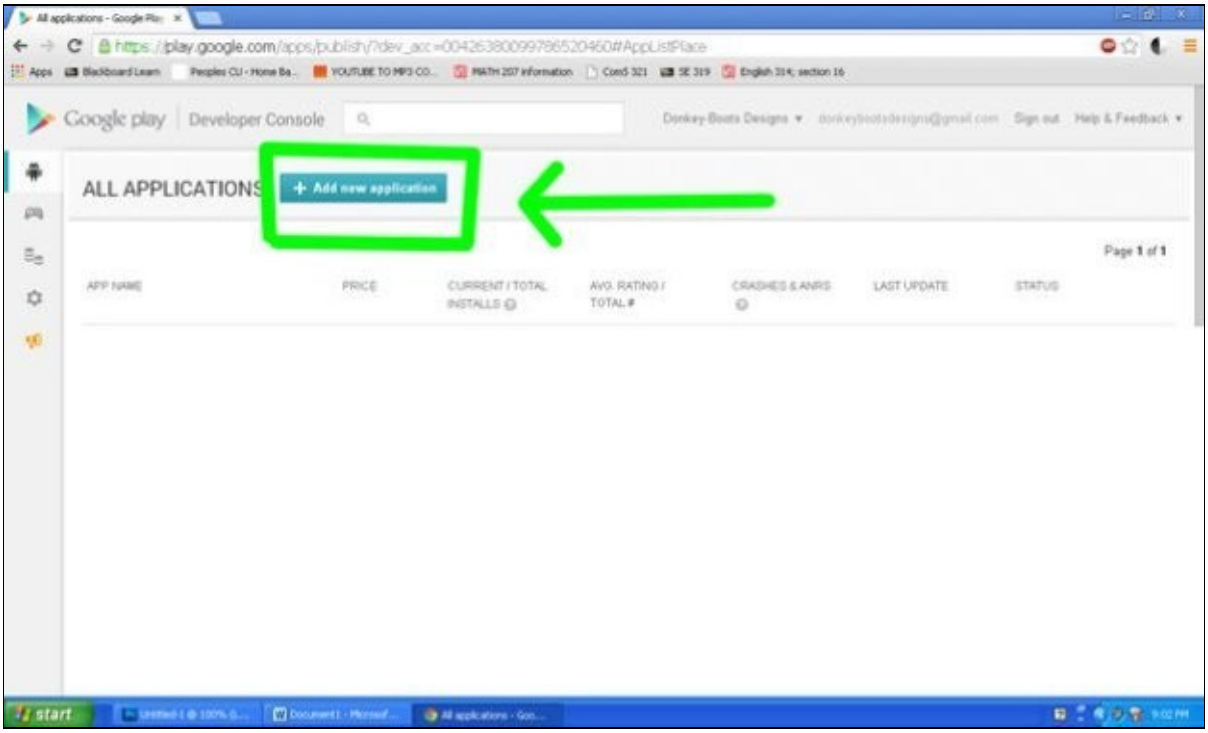
2. Login to Google Developer Console



Get logged in to google's developer console.

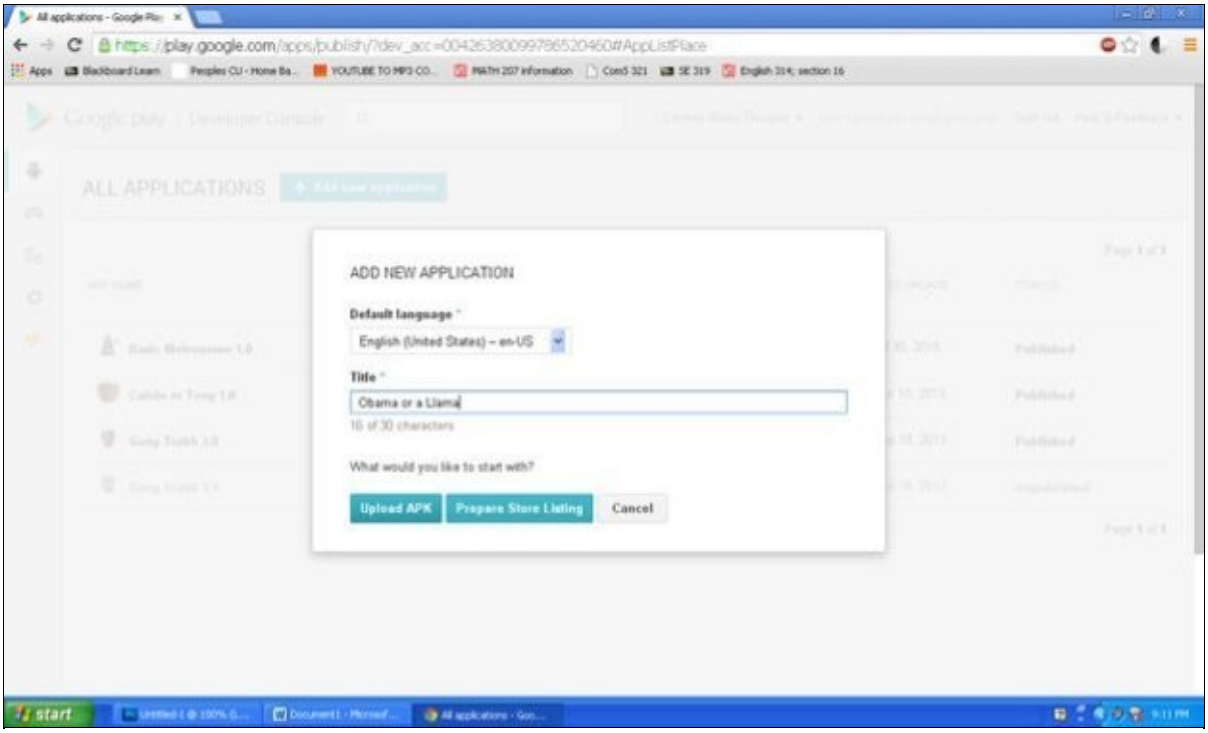
If you do not already have a google developer account, follow Google's steps to create one, by clicking "Sign Up," in the top-right of your screen. The last step in the process of creating a developer account is paying a one-time $25 developer fee using a credit or debit card.

3. Adding a New Application.



Once inside the console, click the blue button at the top labeled, “+ Add New Application.”

4. Initiating APK Upload.



In the "Add New Application" dialog, ensure that the correct language is selected from the drop-down menu and then type the name of the app as you wish for it to appear in the Google Play store.

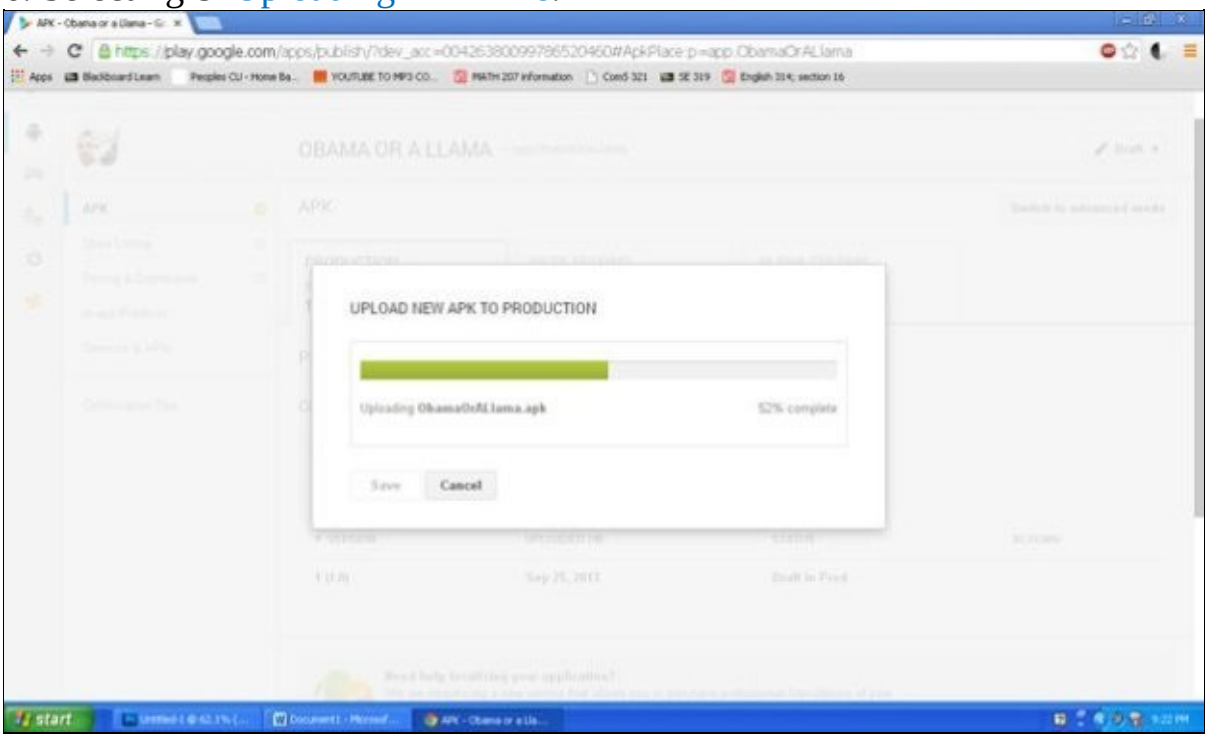
Then, select the "Upload APK" button at the bottom.

The console will then take you to the new homepage for your app.

5. Initiating APK Upload (part 2).

At the new homepage for your app, select the blue button labeled “Upload your first APK to Production,” centered on your screen.

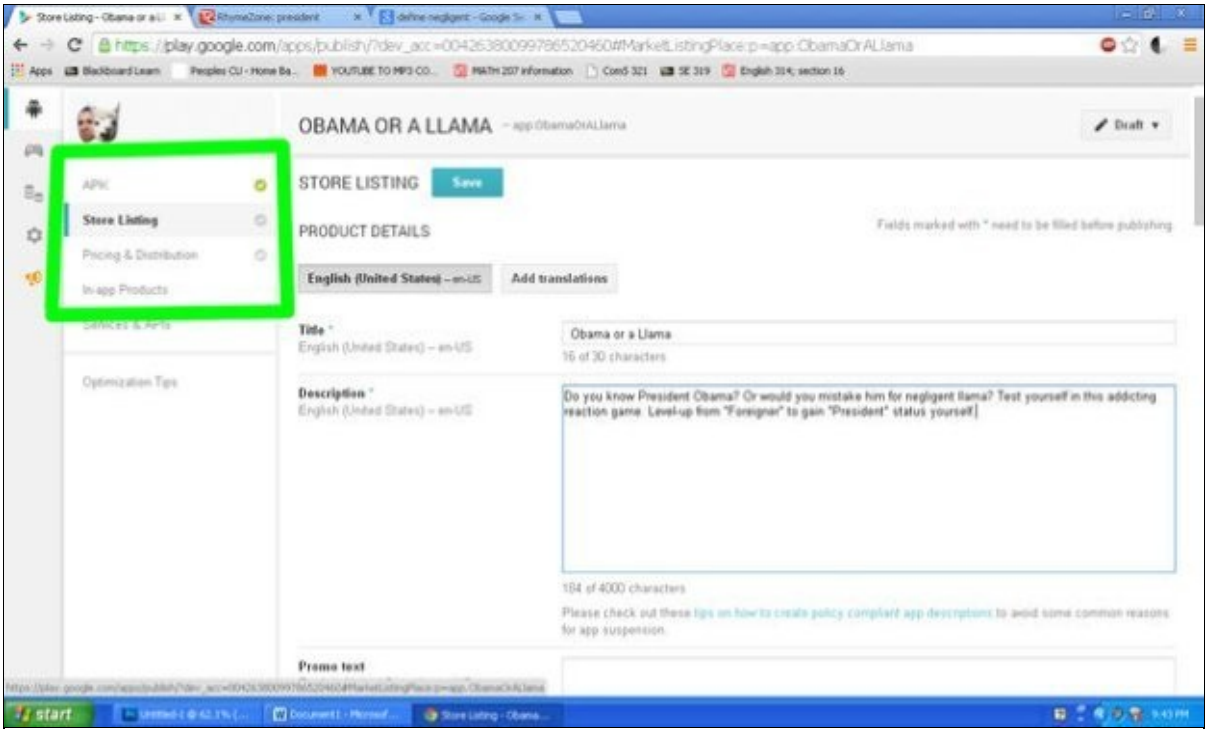
6. Selecting & Uploading APK File.



A dialog will appear, where you can “browse” and “open” your “.apk” file you downloaded.

Upload progress is shown, and if the upload is successful, you will be taken back to the console.

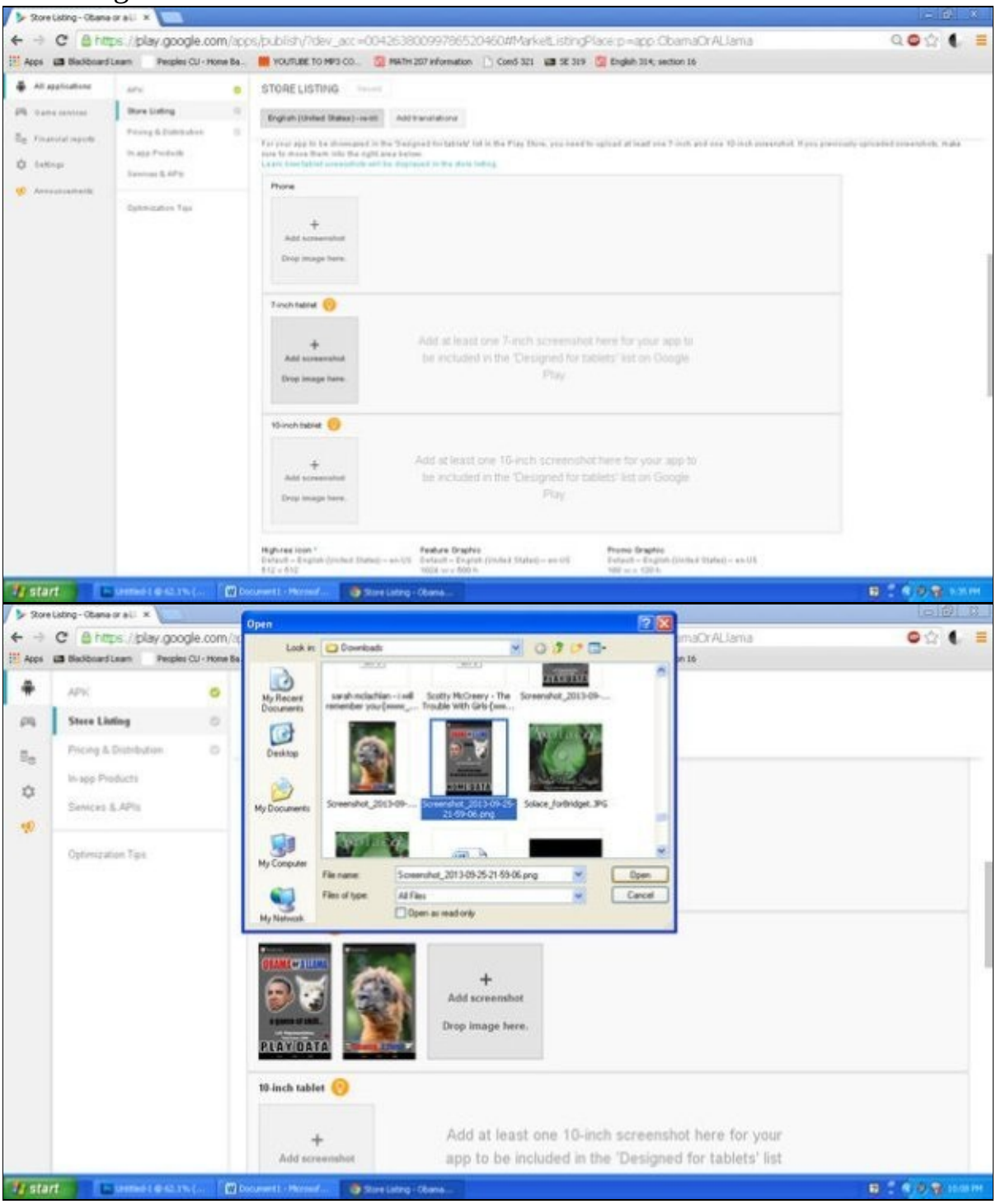
7. Adding a Description.



After successful uploading of the APK, a description needs to be added to the “store listing” page found by navigating the tabs at the left (tabs are shown in green box in attached figure).

Type an application description in the “description” text box. The description will appear in the Google Play store on the page for your app.

8. Adding Screenshots.

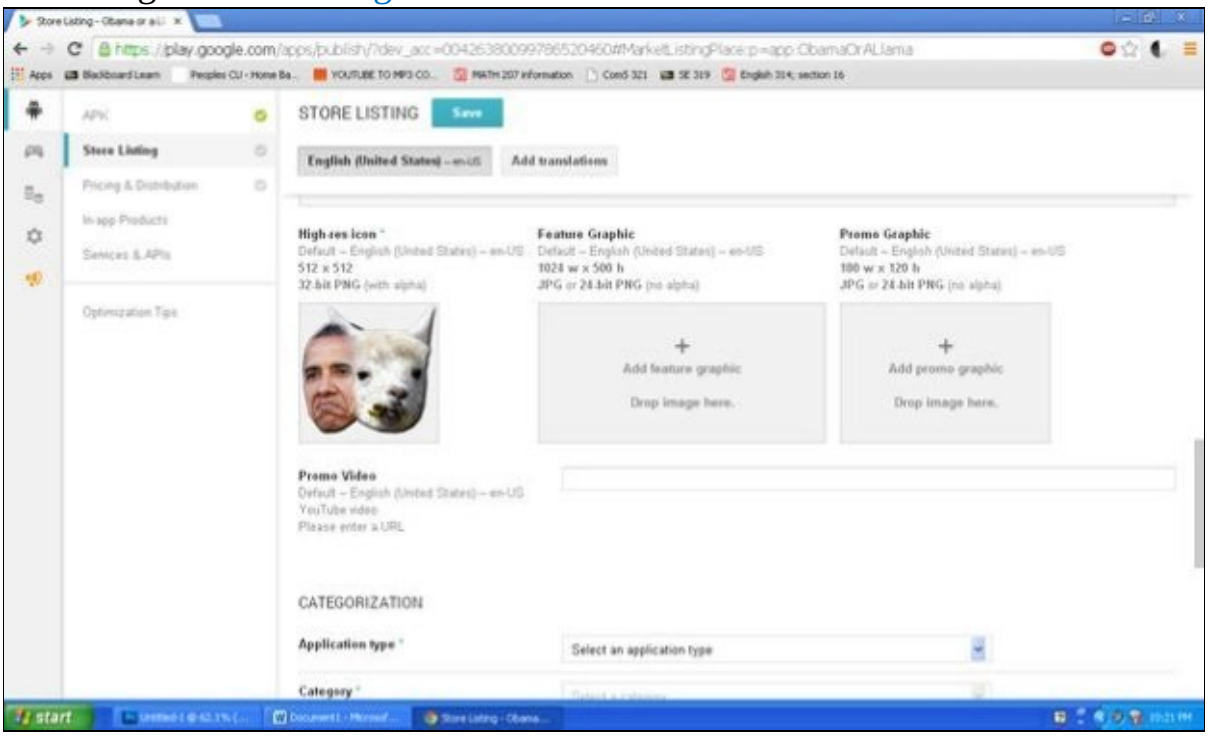


The next step in the "Store Listing" tab is adding sample screenshots.

To upload screen-shots, scroll down and click one of the three "+ Add Screenshot" buttons corresponding to the type of device they were taken on.

A browse dialog box will open allowing you to select the screenshots from a directory on your computer.

9. Adding a Store Listing Icon.



Next, add a store listing icon.

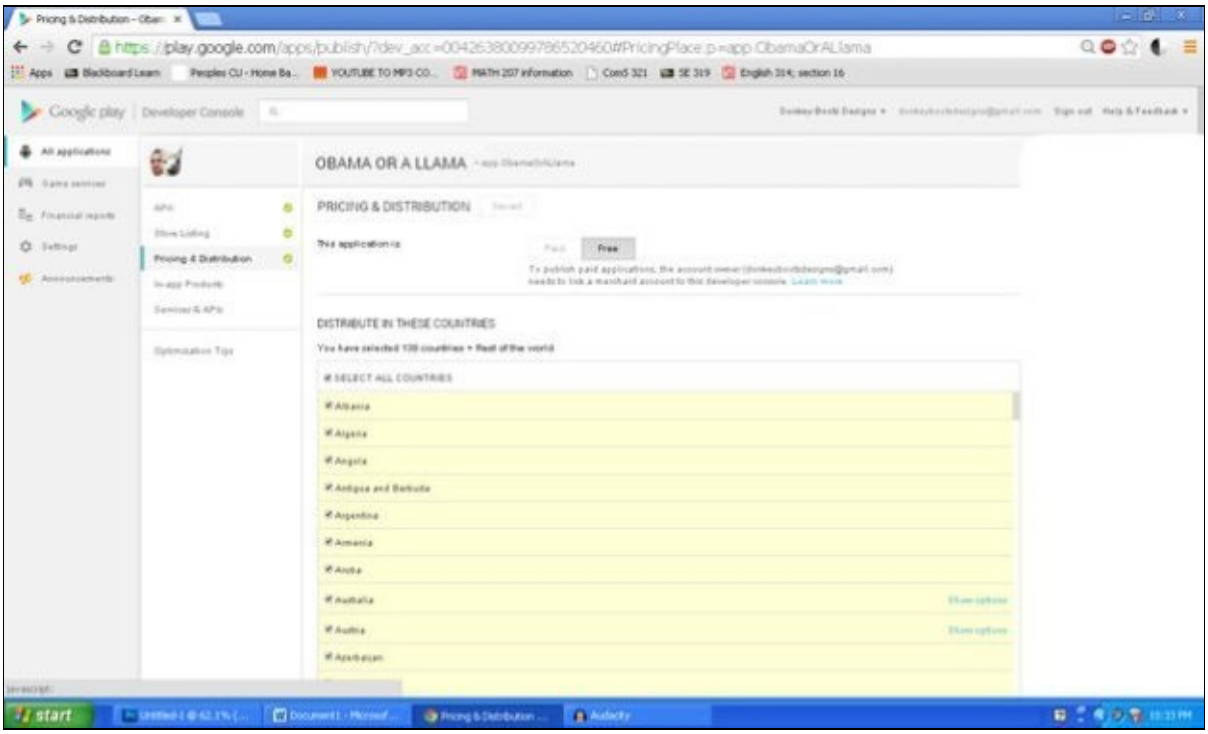
The listing icon is the main picture that pops up at the top of the screen in the Google Play Store.

A browse dialog box will open allowing you to select the icon the same way Step 17 allowed you to select a screenshot.

10. Filling in Details

The final step on the “Store Listing” page is filling out the categorization, contact details, and privacy policy, located below the icon selection area.

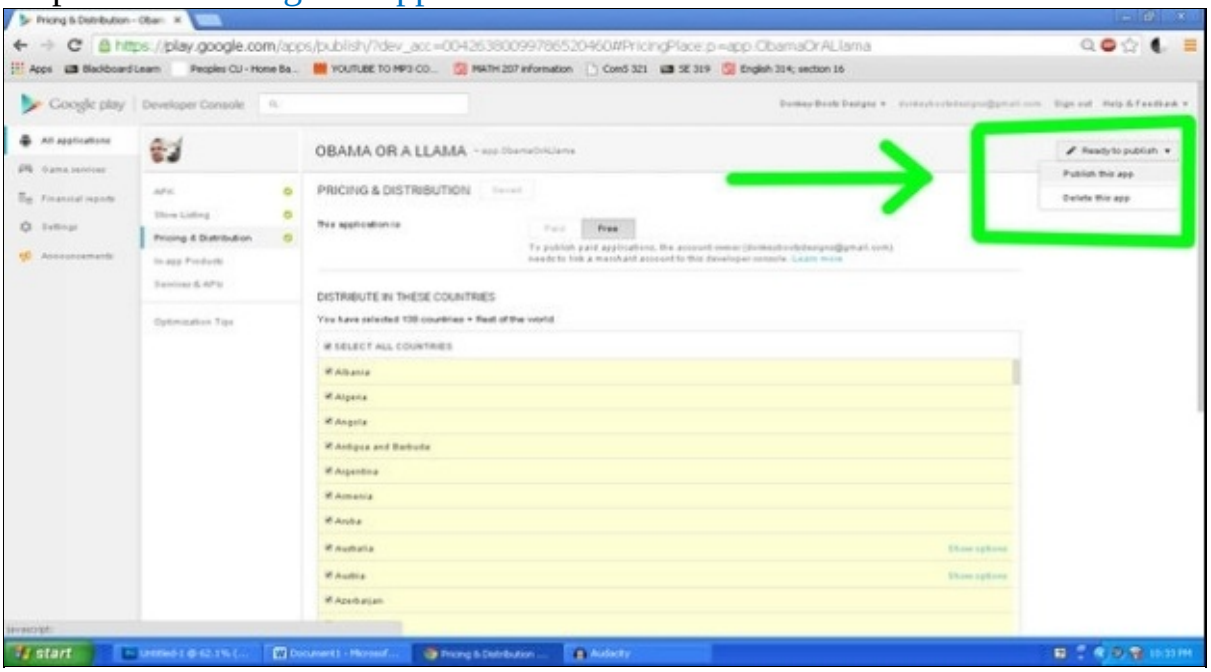
Fill out the required fields as you wish and hit save. The “store listing” tab should now have a green check-mark next to it.



Finally, click the “Pricing & Distribution” tab, where you will select paid or free, distribution countries, and check the boxes saying that your app complies with content guidelines and US export laws.

Select “Save,” at the bottom.

Step 11: Publishing the Application.



Once all three tabs at the left have a green check-mark, you are now able to select “Publish this app” from the “Ready to Publish” drop-down menu in the top right corner of the developer console.

A confirmation bar should appear at the top, stating that your app will appear in the Google Play store in just a few hours.

Step 12: Conclusion

This completes publishing an Android app to the Google Play Store.