MODULE TEACHING & LEARNING

MOBILE APPS DEVELOPMENT

MR SARAWANAN A/L LETCHUMANAN

CREATIVE DESIGN CENTRE

MODUL REKABENTUK KREATIF
POLITEKNIK IBRAHIM SULTAN

Published in 2020 by Creative Design Centre
Politeknik Ibrahim Sultan
KM 10, Jalan Kong Kong,
81700 Pasir Gudang, Johor.
Tel : 07 261 2488
Fasks : 07 261 2402
cdecpis.wixsite.com/mycdec

ISBN :

Manufactured in Malaysia

Module Mobile Apps Development

Chief Editor: Ts. Dr. Maria Binti Mohammad
Editor: Nadirah Binti Abdul Aziz
Author: Sarawanan Letchumanan
Proof-reader: Siti Faezah Binti Ahmad Sazali
Designer : Afham Ilmam Bin Mohamad Dimyati

CONTENTS PAGE

INTRODUCTION

We all know that mobile is the future and with that comes mobile apps, but have you ever wondered how your favourite mobile applications are developed? A mobile application, most commonly referred to as an app, is a type of application software designed to run on a mobile device, such as a smartphone. Mobile applications frequently serve to provide users with similar services to those accessed on PCs. Apps are generally small, individual software units with limited function. Mobile app development requires a person to understand two distinct but related modes: development, wherein an app is made functional; and design, through which an app is made usable. Mobile app development course makes it easy and quick for participant to build clickable and ready-to-deploy mobile apps for android phones.

AIMS

This course will equip you with the basic knowledge and skills required to create your very own app for android phones. Whether you are doing this course for personal or professional reasons, you will learn to design, code and develop basic applications using a simple web-based platform.

LEARNING OUTCOMES

At the end of this module, participants will be able to:

1. Describe the components and structure of MIT App Inventor
2. Describe block programming
3. Apply mobile app designing and coding to develop an application.
4. Demonstrate mobile app designing and coding to develop an application.

SYNOPSIS

This module contains 3 topics:

Topic 1 INTRODUCTION TO MIT APP INVENTOR introduces the background of MIT App Inventor, characteristics and application. A detailed explanation on setting up and getting started to work with MIT App Inventor is noted.

Topic 2 INTRODUCTION TO CODING defines coding and the purpose of coding. Explanation and block-based coding and how it works is noted.

Topic 3 BUILDING APPS introduces the steps involved in building an app. Designing, coding and testing the app is explained step by step for five different types of project.

ASSESSMENT

To enhance participants knowledge and experience in the course, selected assessment is included.

REFERENCES

List of references is at the end of the book.

TOPIC 1: INTRODUCTION TO MIT APP INVENTOR

Topic 1

INTRODUCTION TO MIT APP INVENTOR

Learning outcomes

At the of this topic, participants will be able to:
1. Explain MIT App Inventor.
2. Use MIT App Inventor.
3. Understand three methods to test app

Input

1.1 What is MIT App Inventor

MIT App Inventor is a free, cloud-based service that allows you to make your own mobile apps using a blocks-based programming language. It allows you to build apps right in your web browser. It is a simple and easy to understand technique which can be easily learned by anyone to build a fully functional apps for smartphones.

1.2 Getting Started

Using MIT App inventor to build app does not require you to download or install any application into your computer. What you need is a computer to work on your project, valid Gmail account to login into app inventor platform and an android device to test your app.

01	**Internet and WIFI** To access the platform and built app. It can be accessed from anywhere
02	**Email Account** Login to MIT account, preferably an active Gmail account. All your project will be saved under this account.
03	**Android Phone** To test your app functions and deploy in it your phone
04	**MIT AI2 Companion** Interface to communicate with MIT App Inventor. Download from Google Play

Upon building your apps, at a point it needs to be tested to ensure its functional. There are three option for setting up to testing your apps.

Option 1 – Recommended
If you have a computer, an Android device, and a WIFI connection, this is the easiest way to test your apps. You will develop apps on ai2.appinventor.mit.edu website and do live testing on your Android device. However, it requires the user to install the MIT App Inventor Companion app on your Android phone or tablet. Once the Companion is installed, you can open projects in App Inventor on the web, open the companion on your device, and you can test your apps as you build them:

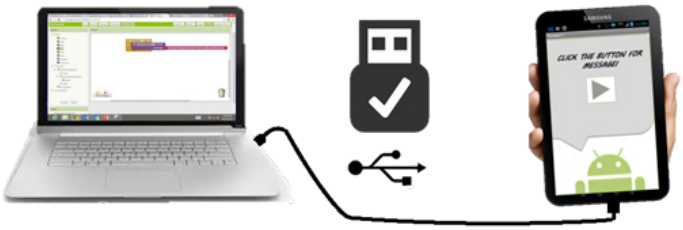
Build your project on your computer Test in real – time on your device

Option 2 – Don't have android device
If you don't have an Android phone or tablet, you can still use App Inventor. You'll develop apps on ai2.appinventor.mit.edu website and do testing on emulators which works just like an Android but appears on your computer screen. An emulator is a hardware device or software program that enables one computer system (also known as a host) to imitate the functions of another computer system (known as the guest). It enables the host system to run software, tools, peripheral devices and other components which are designed for the guest system. Emulators can be of different types, replicating things such as hardware, software, OS or CPU. However, in most cases hardware architecture is emulated to provide an environment similar to a guest system

Build your project on your computer Test in real – time on your computer with the on-screen emulator

To use the emulator, you will need to install some software on your computer. Follow the instruction according to your operating system on this link to install and setup. http://appinventor.mit.edu/explore/ai2/setup-emulator.html

Option 3 – Don't have WIFI
If you don't have WIFI connection but would like to do live testing on you android device rather than using emulator you can connect your phone or tablet with a USB cable.



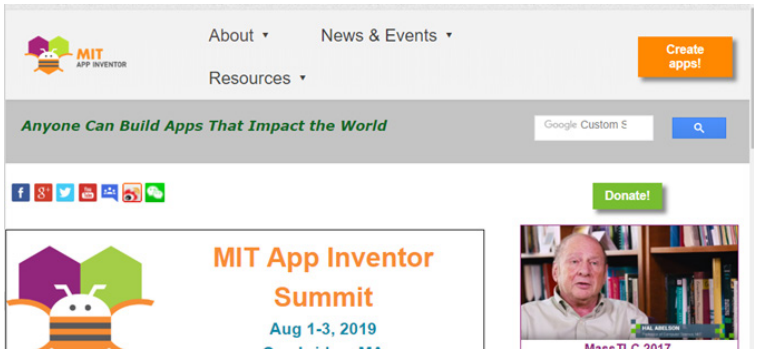
Build your project on your computer Test in real – time on your device

Setting up a USB connection can be awkward. You will need to install some software on your computer. Follow the instruction according to your operating system on this link to install and setup. Unfortunately, different devices may require different approach to enable live testing on the android phone which can be looked up on the android phone user manual. Follow the steps for beginning to use App Inventor with the USB cable in this link.

http://appinventor.mit.edu/explore/ai2/setup-device-usb.html

1.3 App Inventor Environment

To login simply type in http://appinventor.mit.edu/explore/ and click the create app button on the top right end side.



Upon clicking, the app inventor platform will require login using a valid Gmail account. Type in your selected Gmail account and enter. All your project and work will be saved under the chosen account. It will not appear if you use another account to login.

Designer and Blocks Editor

App Inventor consists of the Designer and the Blocks Editor. The Designer environment in figure 1.1 allows the user to design the app's user interface by arranging both on- and off-screen components. In another word it's how your app will look like.

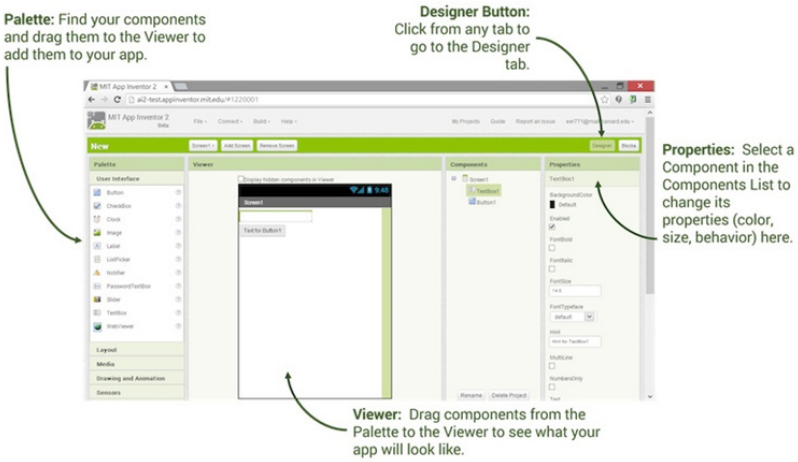


Figure 1.1 Designer Environment

The Block Editor environment in figure 1.2 will Program the app's behaviour by putting blocks together. It allows you to code your app to give it functionality--that is, make it do stuff!

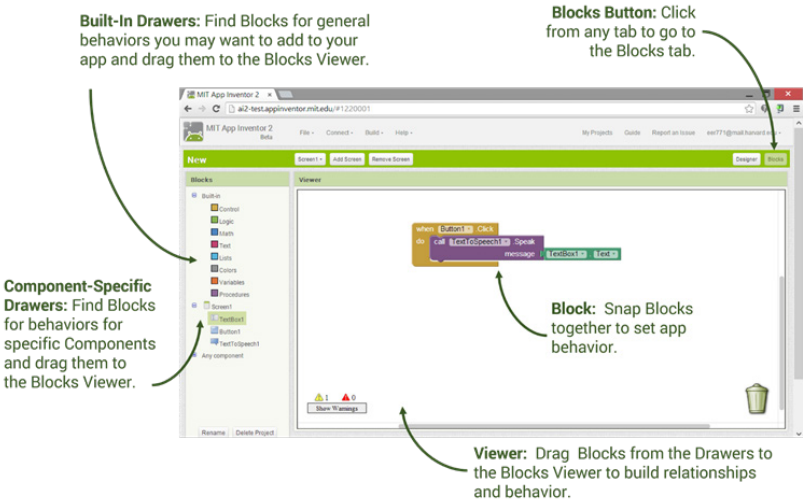


Figure 1.2 Block Editor Environment

TOPIC 2: INTRODUCTION TO CODING

Topic 2

INTRODUCTION TO CODING

Learning outcomes

At the end of this topic, participants will be able to:
1. Define coding.
2. Understand block-based coding.

Input

2.1 What Is Coding

Coding (or programming) is the construction of software. Coding involves writing 'steps', which in computing it is called an algorithm. It is a language that a computer can understand. When the computer runs the code we wrote, it follows the 'steps', step by step.

2.2 Block Based Coding

MIT App Inventor uses block-based coding environment, a programming language that permits the creation of blocks, including blocks nested within other blocks. Block Based Coding is the de facto way to teach new comers introductory programming instead of traditional, text-based programming which can be difficult in some way as you need to type the commands. Block based coding involves dragging "blocks" of instructions. It is simple and easy to understand.

2.3 App Inventor Built-In Blocks

Built-in blocks are available regardless of which components are in your project. In addition to these language blocks, each component in your project has its own set of blocks specific to its own events, methods, and properties. By clicking on this built in Blocks you will be able to see its commands and expressions.

When an event handler fires, it executes a sequence of commands in its body. A command is a block that specifies an action to be performed on the phone (e.g., playing sounds)

Figure 2.1 Playing Sound

The command in figure 2.1 tells that when button1 is clicked, it will execute sound player 1 and plays the sound.
Some commands require one or more input values also known as parameters or arguments to completely specify their action. For example, refer to figure 2.2.

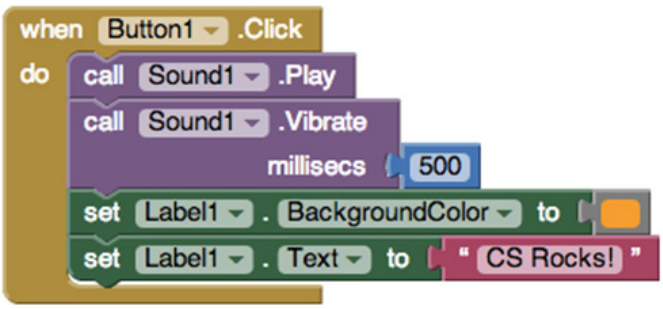


Figure 2.2 Nested Command

This nested command has multiple steps to be executed. When Button1 is clicked:

- Call Sound1 and play.
- Call sound 1 and vibrate for 500ms; needs to know how long to vibrate
- Set Label1.BackgroundColor to orange; needs to know the new background colour of the Label 1
- Set Label1.text to CS Rocks; needs to know the new text string for the label.

The need for input values is shown by sockets on the right edge of the command. These sockets can be filled with expressions, blocks that denote a value

TOPIC 3: BUILDING APPS

Topic 3

BUILDING APPS

Learning outcomes

At the end of this topic, participants will be able to:
1. Design the Apps.
2. Code the Apps.
3. Test the Apps

Input

3.1 Setting Up New Project

To start with a new project, make sure you are in Designer Environment. This environment enables you to design your app by putting in components. Click on Projects in the tool bar and choose Start New Project as in figure 3.1. When prompt for project name type in your project name.

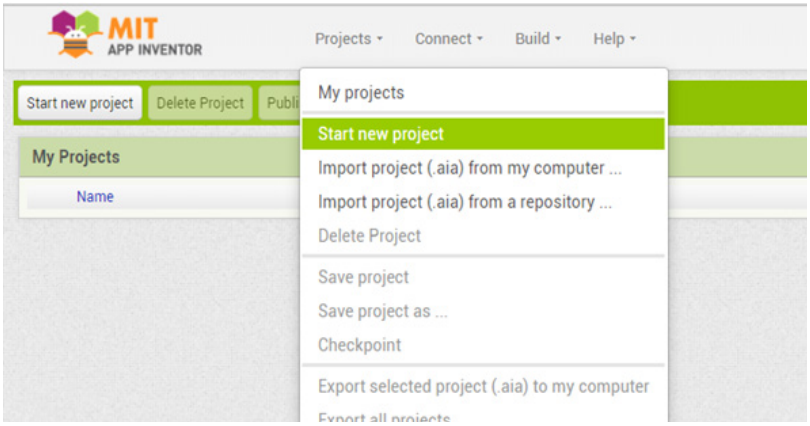
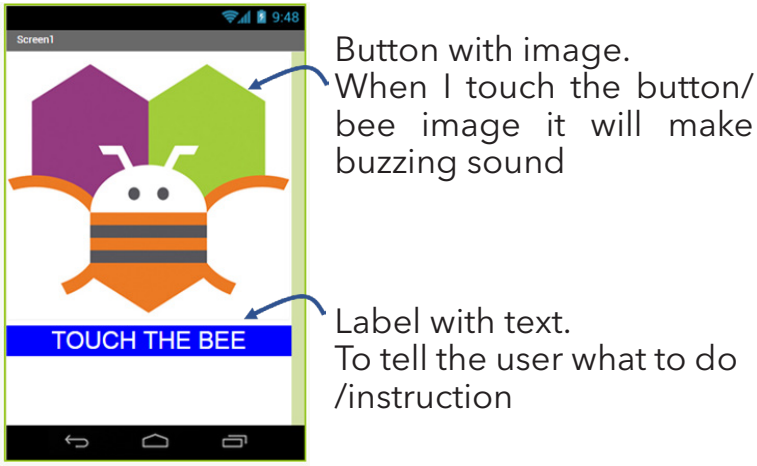


Figure 3.1 Starting New Project

3.2 Touch The Bee

This is a simple app that when you touch the bee picture, it will make a buzzing sound. We will need a Button and Label from the interface.



Step 1 Add Button

From the User interface palette, drag in a Button and drop it onto the viewer.

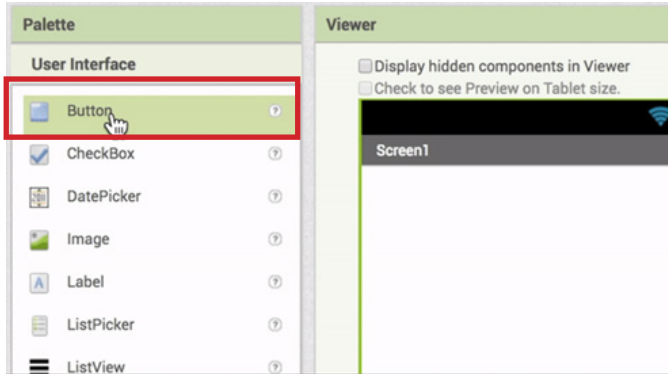
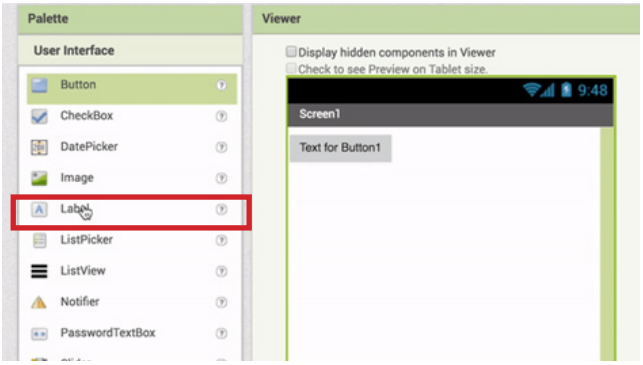


Figure 3.2 Add Button

Step 2 Add Label

From the User interface palette, drag in a Label and drop it onto the viewer. Below the Button



Step 3 Add Sound

From the Media palette, drag in a Sound component and drop it onto the Viewer. Because this is a non-visible component, the sound will drop to the bottom, rather than appearing on the Viewer screen itself.

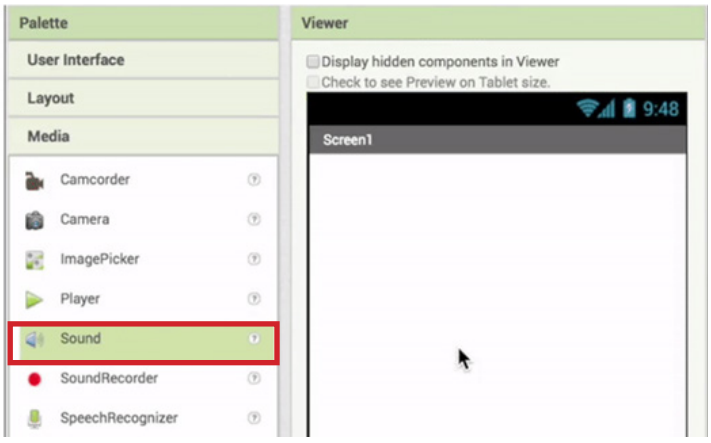


Figure 3.3 Add Sound

Step 4 Setting The Button Properties

In the Components widow, click on Button1. The Properties window for Button1 will appear. Figure 3.4

- Set the Height property to Automatic
- Set the Width property to Fill Parent. This will allow the Button width to be large and same as your device screen.
- Set the image property to codi.jpg (Or any bee image available)
- Set the Text property to be empty. Just erase/delete the text in it.

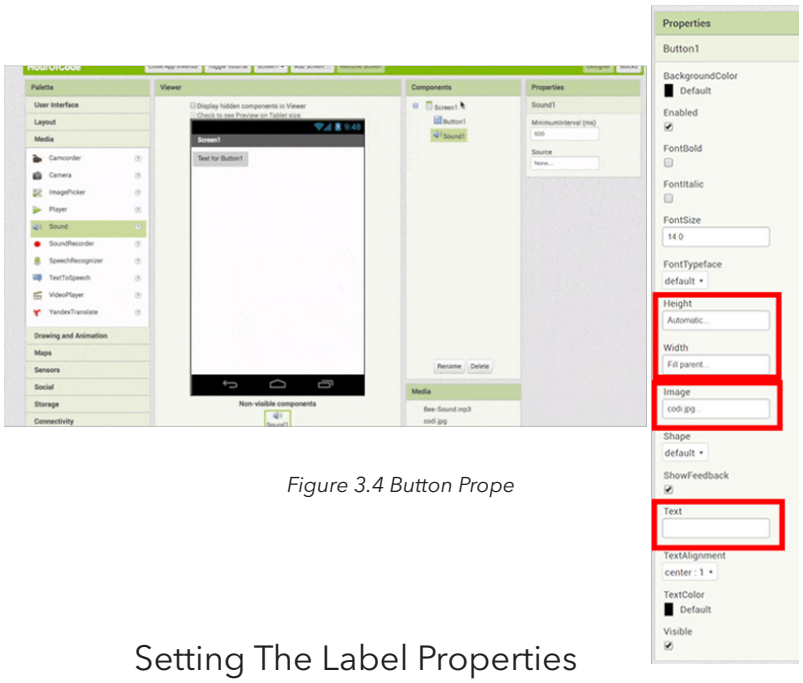


Figure 3.4 Button Prope

Step 5 Setting The Label Properties

In the Components widow, click on Label 1. The Properties window for Label 1 will appear.

- Set the BackgroundColour property to any colour you like
- Set the FontSize property to 30.
- Set the Text property to "TOUCH THE BEE".
- Set the TextColor property to be any color you like.

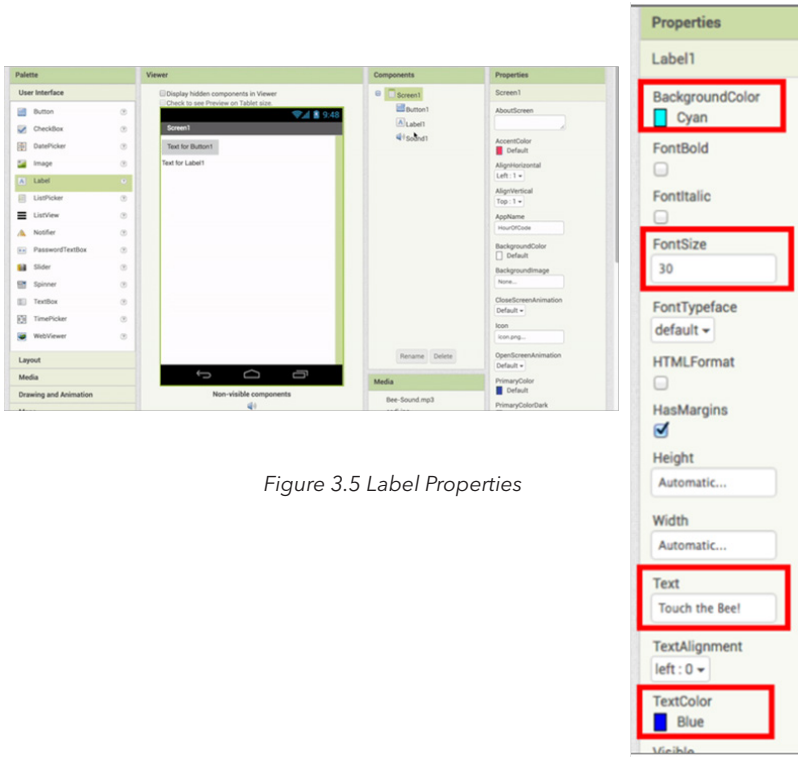


Figure 3.5 Label Properties

Step 6 Setting The Sound Properties

In the Components widow, click on Sound1. The Properties window for Sound1 will appear.

- Set the Source to Bee-Sound.mp3. Or any mp3 bee sound you have downloaded

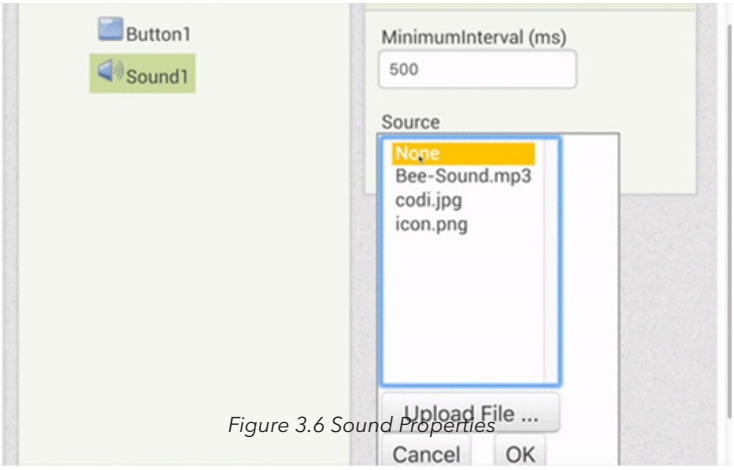


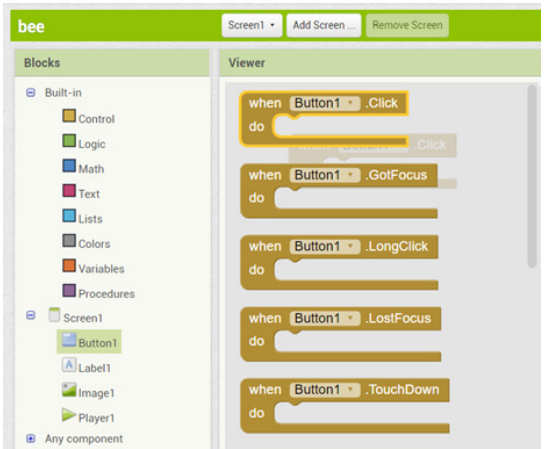
Figure 3.6 Sound Properties

Step 7 Building The Code

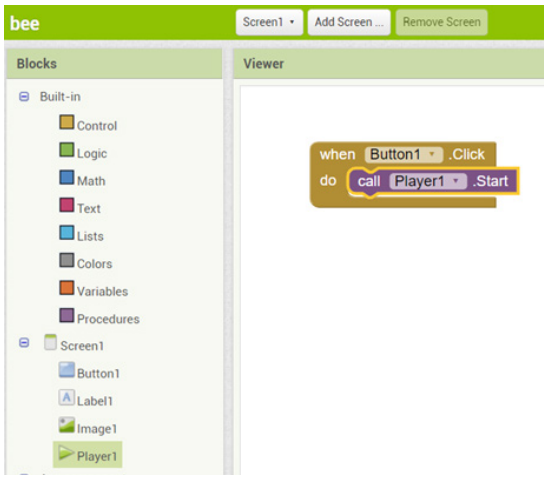
Select the Blocks Editor Button at the top right hand side. It will open the Block editor enivorment which is to enable you to code.

Step 8 Setting The Button Function

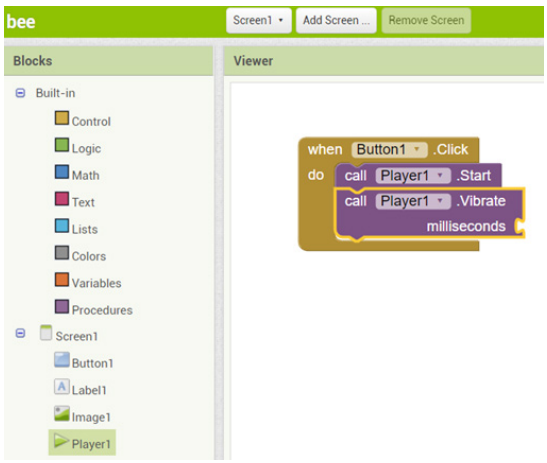
To enable the button to do desired functions. It needs to be code accordingly. From the Button1 drawer drag in "when button1.Click" event block and drop it onto the Viewer. This code will tell what to do when Button1 is clicked.



From the Sound1 drawer, drag in “call Sound1.Play” block and drop it into the Button1.Click block. This code will call the Player1 and play the bee sound which has been set earlier.



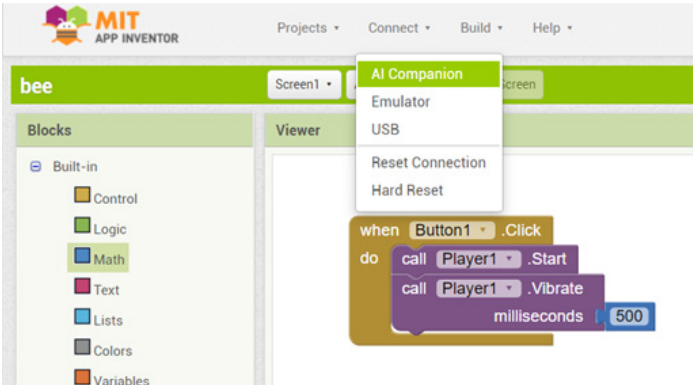
From the Sound1 drawer, drag in “call Sound1.Vibrate” block and drop it into the Button1.Click block under “call Sound1.Play”. This code will call the Player1 and vibrate the phone. T



Notice that the vibrate block millisecond socket is empty. This socket is an extension. From the math drawer drag in block and snap it to the socket. In the block change 0 to 500. This parameter tells how long the phone will vibrate.

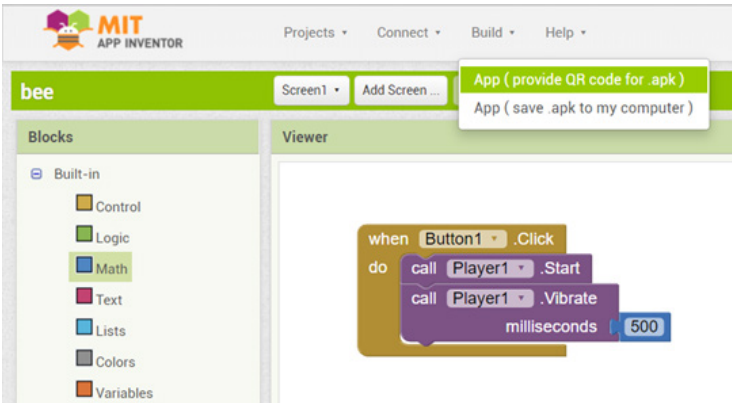
Step 9 Testing The App

Click on Connect in the menu bar and choose AI Companion. This will allow the testing to been done direct on your android device. Make sure your android device is connected to Wi-Fi and MIT AI2 Companion App is deployed. Once the QR code is flashed scan the QR code by selecting scan QR code in your android device. Now you can test your app in the android device



Step 10 Installing The App in your Android Device

Once your project is working good and ready, it is ready to be transferred to your phone. Click on Build in the menu bar and choose Scan QR. Using the MIT AI2 Companion App scan the QR code and the project will be installed in your android device



3.3 Adding Two Numbers

This is a simple app that can add two numbers. When the ANSWER Button is pressed it will add numbers in TextBox1 and TextBox2. Click on Projects and choose Start New Project. Name your project ADD or any suitable name of your choice. Add the components as in figure 3.7 and design you app.

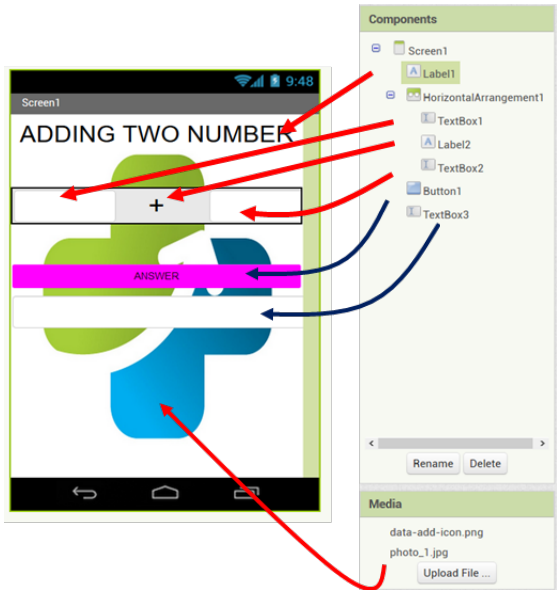


Figure 3.7 Adding Two Numbers

Step 1 Adding Components and Setting properties

Label1 – Information/Text for user

- Text size is 25 (or preferred size)
- Height is Automatic
- Width is Fill Parent
- Text is Adding Two Numbers

HorizontalArangement1 – It is used to arrange components horizontally. TextBox1, Label2 and TextBox2 is placed inside HorizontalArangement1.

- Height is Automatic
- Width is Fill Parent

TextBox1 & 2 – It is used to enable user to key in data/number

- Text size is 18 (or preferred size)
- Height is Automatic
- Width is Fill Parent

Label2 – Information/Text for user

- Text size is 18 (or preferred size)
- Height is Automatic
- Width is Fill Parent
- Text is +

Button1 – User will press Button1 to get the answer

- Text size is 25 (or preferred size)
- Height is Automatic
- Width is Fill Parent
- Text is ANSWER
- Background colour is purple (or preferred colour)

TextBox3 – The sum of numbers will be shown in here.

- Text size is 18 (or preferred size)
- Height is Automatic
- Width is Fill Parent

Screen1 Background Image – image as background (you can set any image you like or can set the Background Colour to any colour you like.

Step2 Building The Code

When Button1 is clicked TextBox3 will Display the sum of TextBox1 and TextBox2. The proper code for this task is as in Fig 3.8

Figure 3.8 Code For Adding Two Numbers

3.4 Multiple Screens

This is an app that has multiple screens or functions in it. We will design so that it has 4 different screens or function to do. Click on Projects and choose Start New Project. Name your project Multiple Screen or any suitable name of your choice. Add the components as in figure 3.9 and design your app main page.

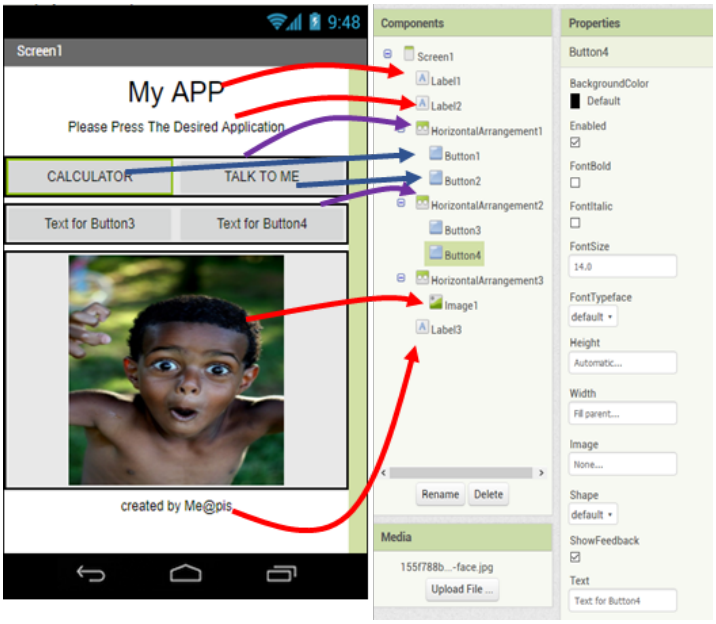


Figure 3.9 Main Page For Multiple Screens

Click on Add Screen. Type new screen name calculator (or any given name) as in figure 3.10. A new Screen will be created with the name calculator.

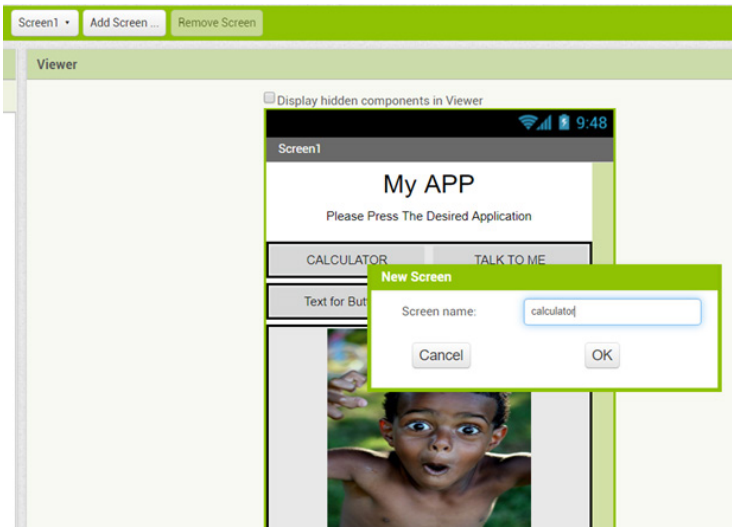


Figure 3.10 Adding New Screen

You can choose the screen created to work on by clicking on the selected screen as in figure 3.11. Click on Screen1 and code the following. When Button1(Calculator) is clicked it will open screen calculator. Make sure the words are exactly as the screen name.

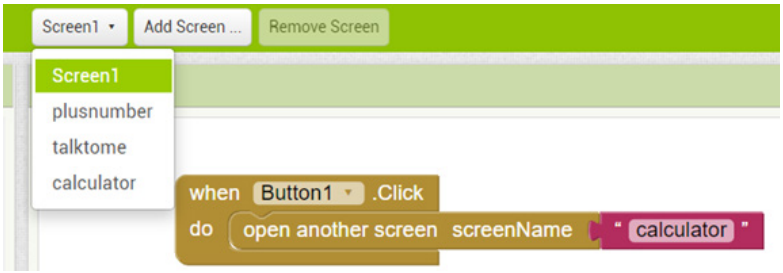


Figure 3.11 Code For Open Another Screen

3.5 Calculator

Choose screen calculator. Add the following components to create your main page. This calculator be able to plus, minus, times and divide two numbers. When the plus button is clicked it will plus Number1 and Number2. The answer is shown in TextBox3. When the minus button is clicked it will minus Number1 and Number2. The answer is shown in TextBox3, and the same goes to times and divide button. Components needed:

Label1 and TextBox1 in HorizontalArangement1

Label2 and TextBox2 in HorizontalArangement2

4 Buttons in HorizontalArangement3. Each buttons with background image as in figure 3.12.

Label3 with Text ANSWER

TextBox3 to display the answer

A Reset Button in HorizontalArangement4 which when clicked will reset all values in TextBox1,2 and 3.

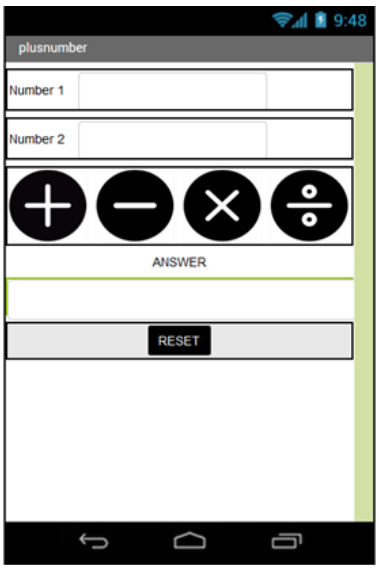


Figure 3.12 Main Page For Calculator

For the code, whenever button plus, minus, times or divide is clicked it will do the intended operation and display the answer in TextBox3. The Reset Button will be used to clear all entries in TextBox. The code is as in fig 3.13.

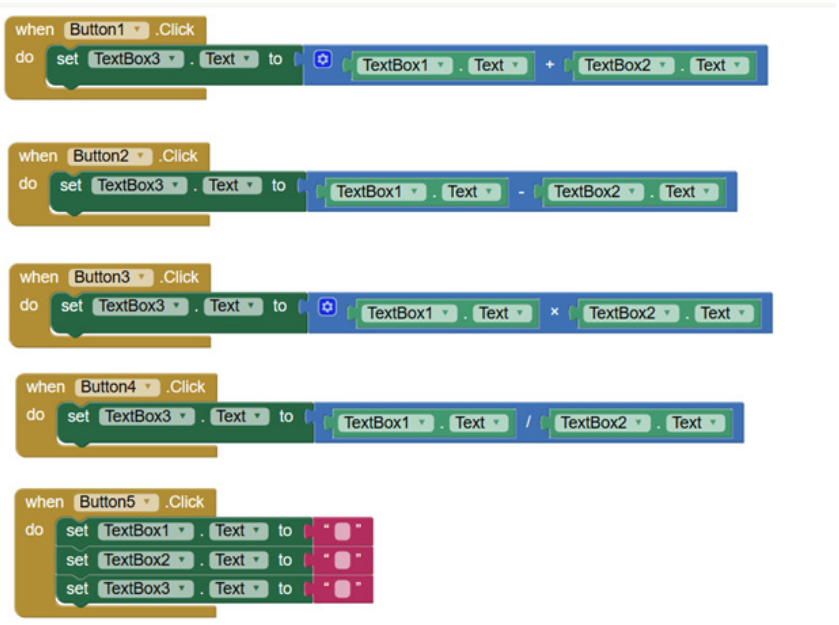


Figure 3.13 Code For Calculator

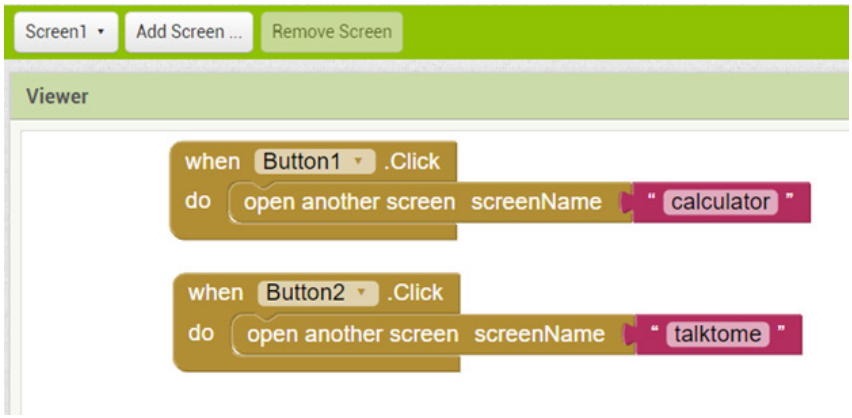
SELF ASSESSMENT

Activity 1

1. How to go back to Screen1/Main page? Try to think what need to be added and coded.

3.6 Talk To Me

Click on Add Screen. Type new screen name talktome (or any given name). A new Screen will be created with the name talktome. Click on Screen1 and code the following, when Button2, Talk To Me is clicked it will open screen talktome.



Choose screen talktome . Add the following components to create your main page as in figure 3.14. This app will speak the words typed in TextBox1.

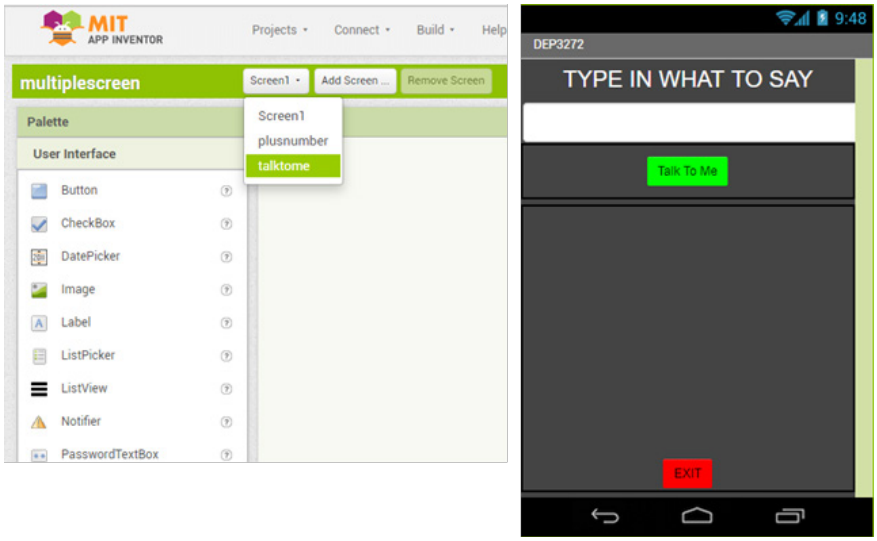


Figure 3.15 Main Page For Talktome

Components needed:

- Label1 (TYPE IN WHAT TO SAY)

- TextBox1 (Word to be converted to speech)

- Button1 (To click after words are typed in TextBox1)

- Button2 (To go back to Screen1)

- TextToSpeech (A Function to convert text to speech, it is found under Media as in figure 3.16

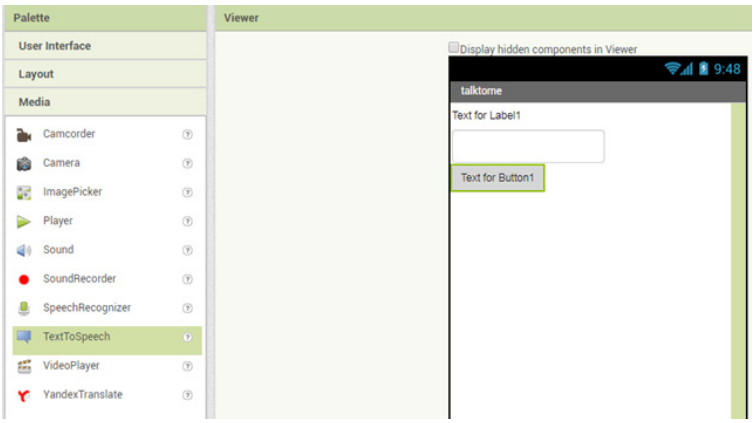


Figure 3.16 TextToSpeech

For the code, When Button1 is clicked it will call the TextToSpeech component and speak whatever message is written in Text Box1. To exit or back to Screen1, Button2 is clicked. The code is as in figure 3.17

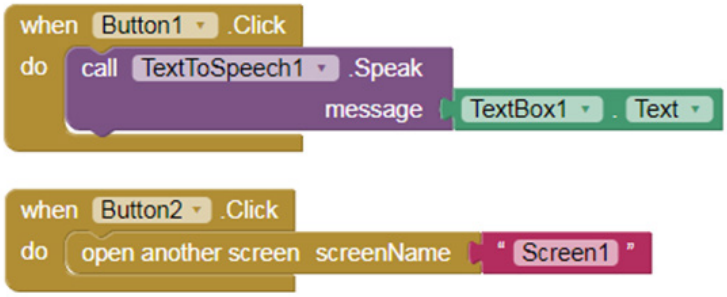


Figure 3.17 Code For TextToSpeech

SELF ASSESSMENT

Activity 2

1. If the text can be converted to speech using the TextToSpeech component, can our calculator speak the answer? Try to think what need to be added and coded in screen calculator.

3.7 Selecting Images

Let's create an app that will show images of accidents. When a type of accident is clicked, the image of the chosen accident will appear. Add a new screen and build the main page of the screen as in figure 3.18.

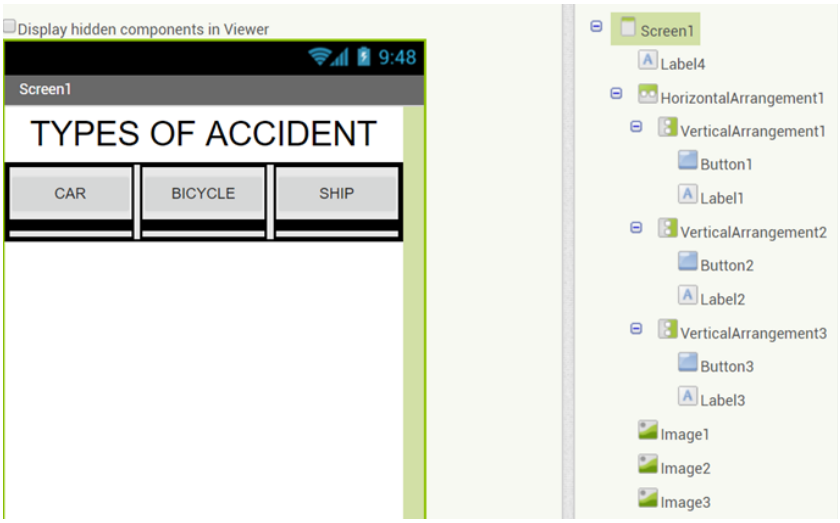
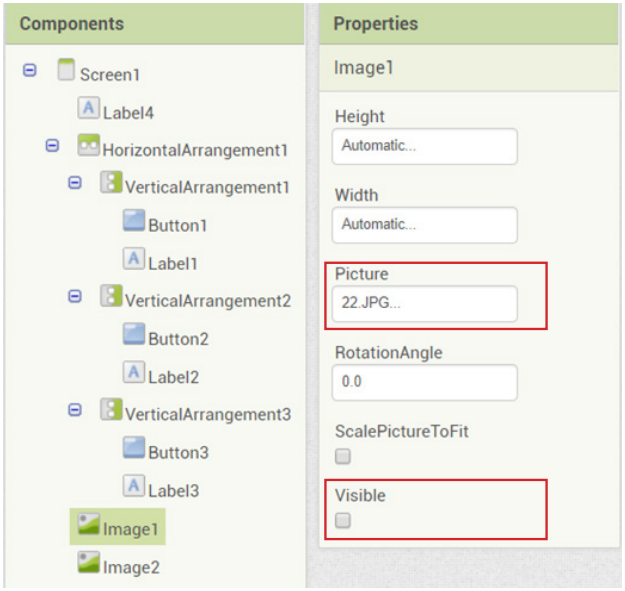


Figure 3.18 Images Selection

Three vertical arrangement is placed in the horizontal arrangement box. Each vertical arrangement box has button and label. The height of the labels in vertical box is 7 pixel and the height are fill parent. The background colour for label is black.

Drag and place three image under the horizontal arrangement box. For each image set the Picture to the selected image which is downloaded. All the image is set invisible by unchecking the Visible box under the image properties.



For the code, When Button1, 2 or 3 is clicked it will show the image accordingly. The label will change colour to green when the selected button is clicked. The code is in figure 3.19

Activity 2

1. If the text can be converted to speech using the TextToSpeech component, can our calculator speak the answer? Try to think what need to be added and coded in screen calculator.

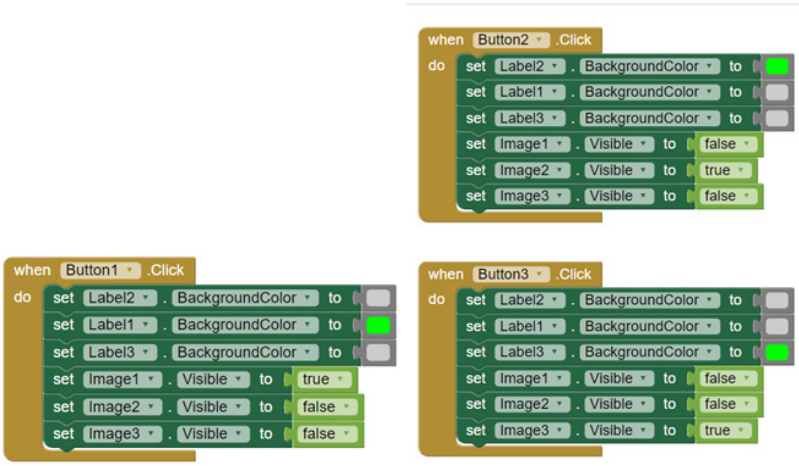


Figure 3.19 Code For Selecting Image

SELF ASSESSMENT

Activity 3

1. It is boring to view picture without a music being played at background. What can be added to play a music and how is the code?

Activity 4

1. For the last screen we will build an app that can show either you passed or failed in your test by typing your marks. The passing mark is 50. If any number above 100 is typed, a notification message will appear.
2. Try showing the following grade:

0 – 49 F	FAIL	
50 – 59	D	PASS
60 – 69	C	PASS
70 – 79	B	PASS
80 – 100 A	PASS	

Reference

1. http://appinventor.mit.edu/explore/
2. https://www.techopedia.com/definition/4788/emulator

About The Author

Name: Sarawanan Letchumanan

Biodata:
Sarawanan Letchumanan is a senior lecturer in Polytechnic Ibrahim Sultan and been working there since 1999. He is a lecturer in Electrical Engineering Department and specializes in Communication Engineering Field. He is a graduate from Malaysia Technology University with Bachelor's Degree in Electrical Engineering and Masters of Education. Currently he is also serving as the Head for E learning committee in Polytechnic Ibrahim Sultan.

Direka Oleh :

CREATIVE DESIGN CENTRE

Creative Design Centre,
Politeknik Ibrahim Sultan
KM10, Jalan Kong Kong,
81700 Pasir Gudang,
Johor Darul Ta'zim.

No Telefon : +607-261 2488/2813
Faks : +607-261 2402
Laman Web : www.pis.edu.my