



CCE60220

Perangkat Bergerak (TKOM)



Fakultas Ilmu Komputer Universitas Brawijaya

Agenda Perkuliahan

1. ~~Intro dan overview perkuliahan~~
2. ~~Sejarah dan perkembangan teknologi perangkat bergerak~~
3. ~~Komponen perangkat keras dan perangkat lunak~~
4. ~~Pengenalan dan instalasi android studio serta aplikasi sederhana~~
5. ~~Intent dan passing data pada Android Studio~~
6. ~~Android Studio: Sensor reading~~
7. ~~Android Studio: Storage & shared preference~~
8. =====**UTS**
9. Pengenalan dan aplikasi sederhana dengan MIT AppInventor
10. Appinventor: variable, looping, conditional, tinyDB, file
11. appInventor: sensor reading & **persiapan project**
12. Appinventor: Akuisisi gambar dan suara
13. Appinventor: komunikasi bluetooth
14. Appinventor: basic animation
15. **Presentasi kelompok**
16. =====**UAS**



FILKOM | UB



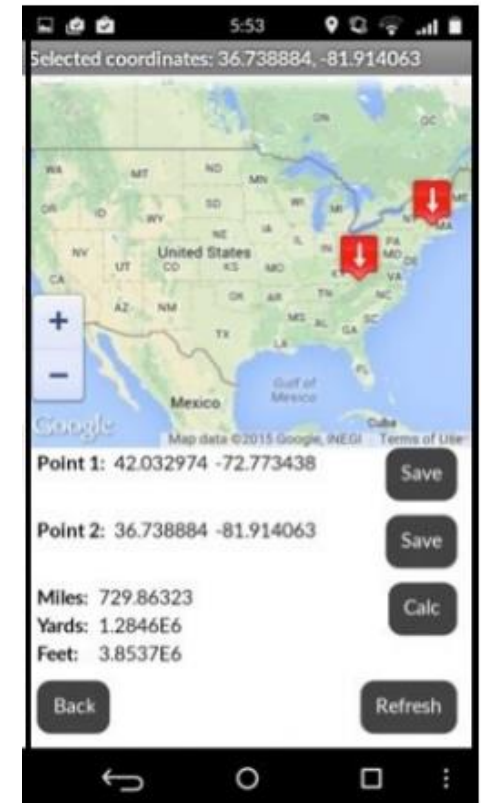
AppInventor

intro

MIT App Inventor 2

- MIT App Inventor 2 is a free, **drag-and-drop**, blocks-based visual programming language that enables people, regardless of their coding experience, to create mobile apps for **Android devices**.
- MIT professor Hal Abelson had the idea to create an easy-to-use programming language to make mobile apps that would harness the power of the emerging smartphone technology.
- In December 2013, Abelson and his team of developers launched MIT App Inventor 2 (from here on referred to as MIT App Inventor), an even easier to use web-based application version featuring an Integrated Development Environment (IDE)

Example of app using AppInventor

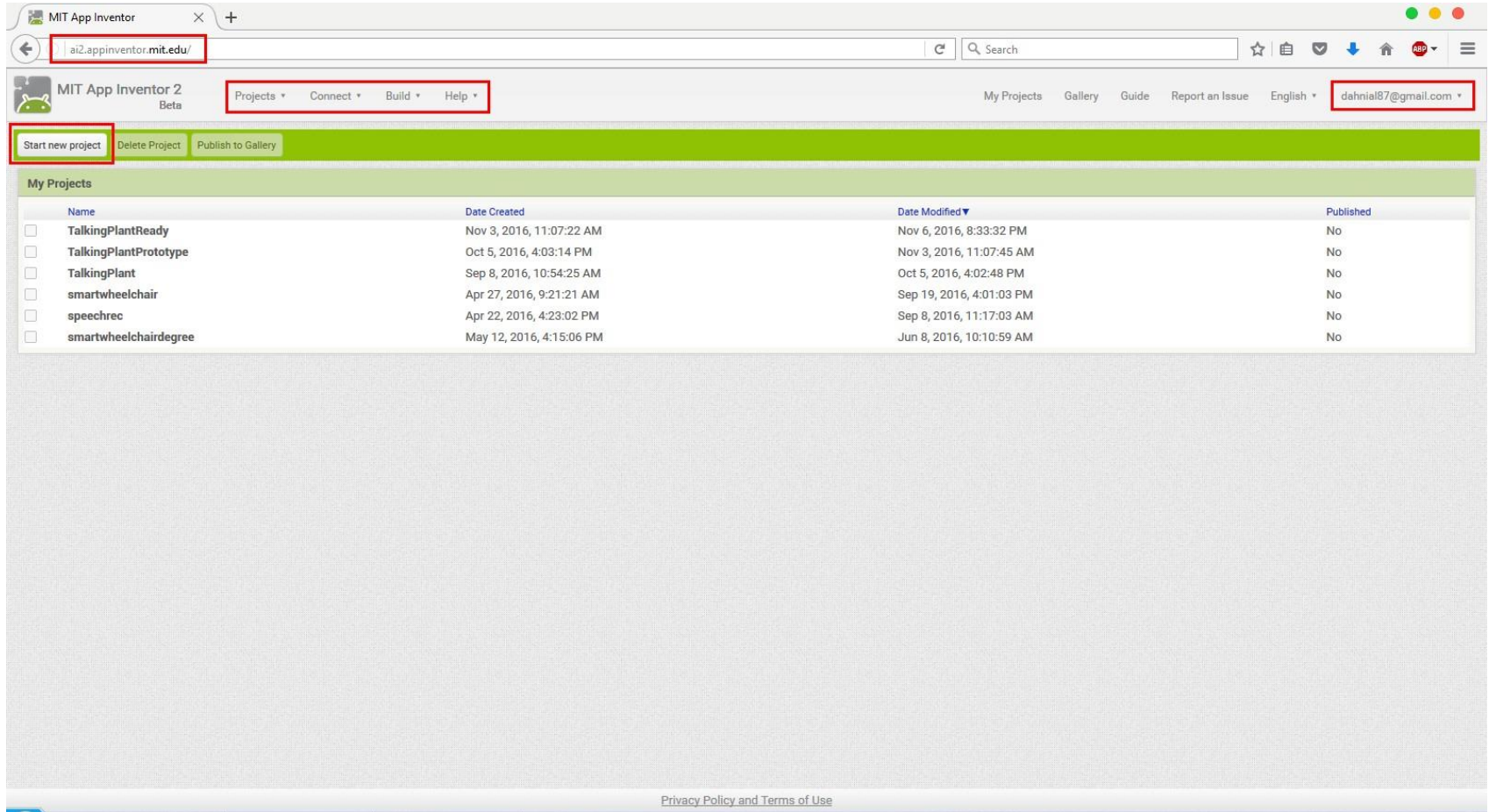


Example of app using AppInventor



- **The MIT App Inventor (<http://appinventor.mit.edu/>) interface includes two main screens:**
- a **Designer Screen**, which is a graphical user interface (GUI) where you can create the look and feel of the app (choosing the components you want the app to include),
- and the **Blocks Editor**, where you can add behavior to the app by coding it with colorful blocks.
- **Users build apps by dragging components and blocks from the menu bars onto a workspaces (called Viewers) and a connected Android device**

User Interface

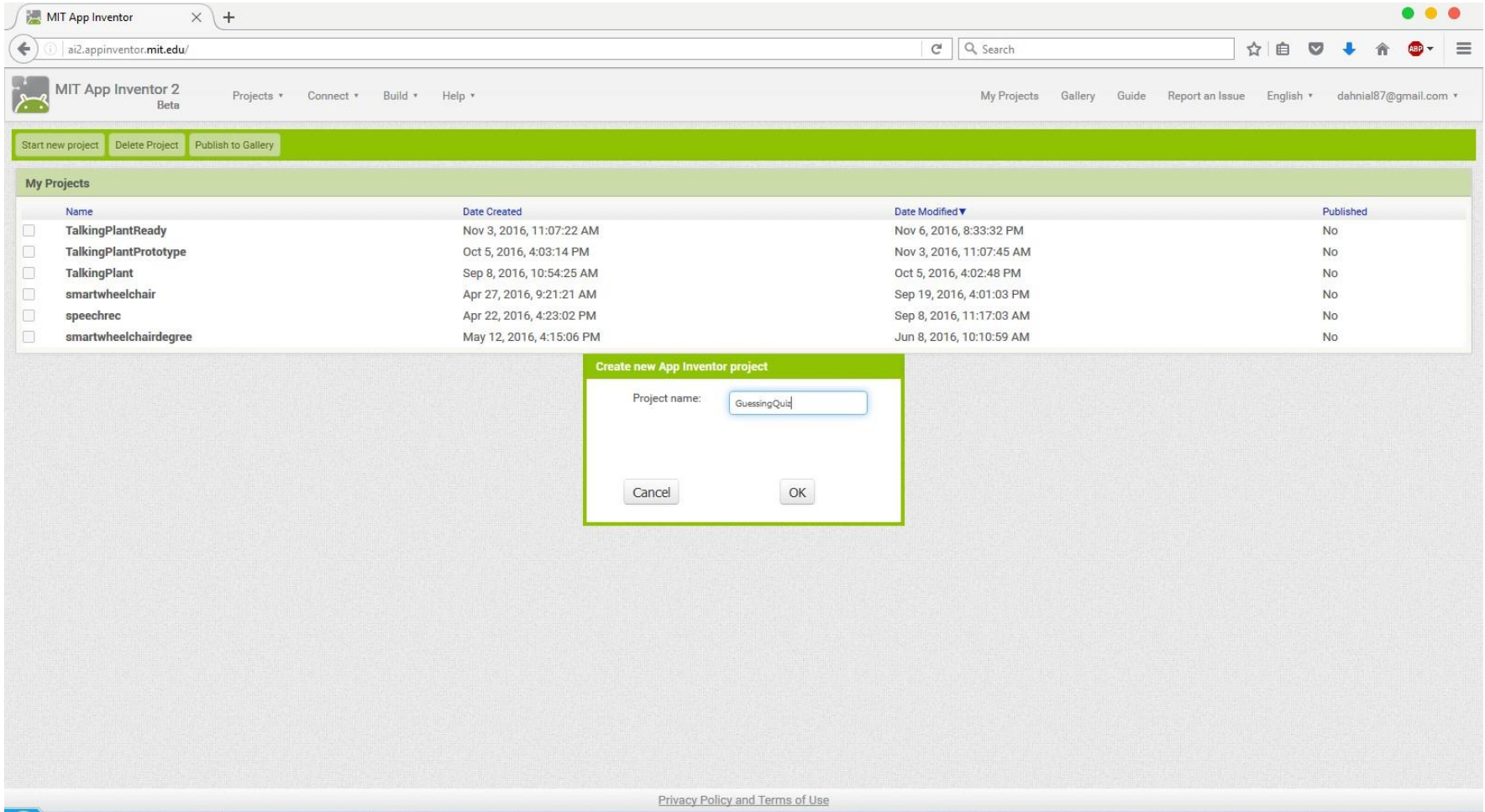


The screenshot shows the MIT App Inventor 2 Beta web interface. The browser address bar shows `ai2.appinventor.mit.edu/`. The page header includes the MIT App Inventor 2 Beta logo, navigation menus for `Projects`, `Connect`, `Build`, and `Help`, and a user profile dropdown for `dahnial87@gmail.com`. A green action bar contains `Start new project`, `Delete Project`, and `Publish to Gallery` buttons. The main content area is titled `My Projects` and displays a table of project entries.

	Name	Date Created	Date Modified	Published
<input type="checkbox"/>	TalkingPlantReady	Nov 3, 2016, 11:07:22 AM	Nov 6, 2016, 8:33:32 PM	No
<input type="checkbox"/>	TalkingPlantPrototype	Oct 5, 2016, 4:03:14 PM	Nov 3, 2016, 11:07:45 AM	No
<input type="checkbox"/>	TalkingPlant	Sep 8, 2016, 10:54:25 AM	Oct 5, 2016, 4:02:48 PM	No
<input type="checkbox"/>	smartwheelchair	Apr 27, 2016, 9:21:21 AM	Sep 19, 2016, 4:01:03 PM	No
<input type="checkbox"/>	speechrec	Apr 22, 2016, 4:23:02 PM	Sep 8, 2016, 11:17:03 AM	No
<input type="checkbox"/>	smartwheelchairdegree	May 12, 2016, 4:15:06 PM	Jun 8, 2016, 10:10:59 AM	No

[Privacy Policy and Terms of Use](#)

Create New Project



The screenshot shows the MIT App Inventor 2 Beta web interface. The browser address bar shows 'ai2.appinventor.mit.edu/'. The main navigation bar includes 'Projects', 'Connect', 'Build', and 'Help'. A secondary navigation bar contains 'Start new project', 'Delete Project', and 'Publish to Gallery'. Below this is a 'My Projects' section with a table of existing projects. A modal dialog box titled 'Create new App Inventor project' is open, with a text input field containing 'GuessingQuiz' and 'OK' and 'Cancel' buttons.

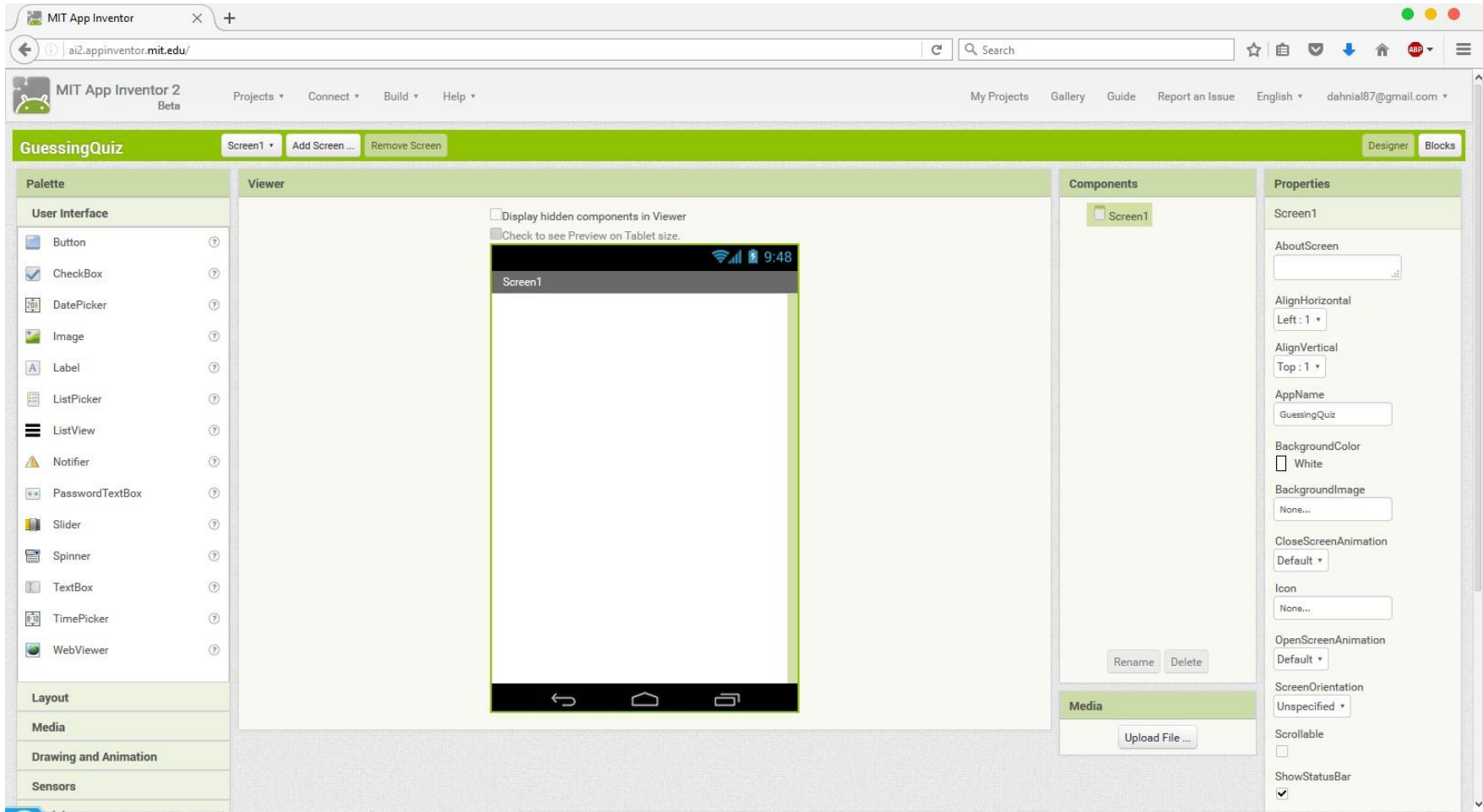
Name	Date Created	Date Modified	Published
<input type="checkbox"/> TalkingPlantReady	Nov 3, 2016, 11:07:22 AM	Nov 6, 2016, 8:33:32 PM	No
<input type="checkbox"/> TalkingPlantPrototype	Oct 5, 2016, 4:03:14 PM	Nov 3, 2016, 11:07:45 AM	No
<input type="checkbox"/> TalkingPlant	Sep 8, 2016, 10:54:25 AM	Oct 5, 2016, 4:02:48 PM	No
<input type="checkbox"/> smartwheelchair	Apr 27, 2016, 9:21:21 AM	Sep 19, 2016, 4:01:03 PM	No
<input type="checkbox"/> speechrec	Apr 22, 2016, 4:23:02 PM	Sep 8, 2016, 11:17:03 AM	No
<input type="checkbox"/> smartwheelchairdegree	May 12, 2016, 4:15:06 PM	Jun 8, 2016, 10:10:59 AM	No

Create new App Inventor project

Project name:

[Privacy Policy and Terms of Use](#)

UI: Designer



The screenshot displays the MIT App Inventor 2 Beta web interface in the Designer mode. The browser address bar shows the URL `ai2.appinventor.mit.edu/`. The application title is "MIT App Inventor 2 Beta". The main workspace is titled "GuessingQuiz" and contains a single screen named "Screen1".

The interface is divided into several panels:

- Palette:** A list of UI components categorized into "User Interface", "Layout", "Media", "Drawing and Animation", and "Sensors". The "User Interface" category is expanded, showing components like Button, CheckBox, DatePicker, Image, Label, ListPicker, ListView, Notifier, PasswordTextBox, Slider, Spinner, TextBox, TimePicker, and WebViewer.
- Viewer:** A central area showing a preview of the mobile application. It includes a status bar at the top with the time "9:48" and a navigation bar at the bottom. A checkbox "Display hidden components in Viewer" is checked, and another checkbox "Check to see Preview on Tablet size." is present.
- Components:** A panel on the right showing the "Screen1" component with "Rename" and "Delete" buttons.
- Properties:** A panel on the right showing the properties for "Screen1". The "AboutScreen" property is set to a text input field. Other properties include "AlignHorizontal" (Left: 1), "AlignVertical" (Top: 1), "AppName" (GuessingQuiz), "BackgroundColor" (White), "BackgroundImage" (None...), "CloseScreenAnimation" (Default), "Icon" (None...), "OpenScreenAnimation" (Default), "ScreenOrientation" (Unspecified), "Scrollable" (unchecked), and "ShowStatusBar" (checked).

UI: designer palette

Palette

User Interface

- Button
- CheckBox
- DatePicker
- Image
- Label
- ListPicker
- ListView
- Notifier
- PasswordTextBox
- Slider
- Spinner
- TextBox
- TimePicker
- WebView

Palette

User Interface

Layout

- HorizontalArrangement
- HorizontalScrollArrangement
- TableArrangement
- VerticalArrangement
- VerticalScrollArrangement

Media

Drawing and Animation

Sensors

Social

Storage

Connectivity

LEGO® MINDSTORMS®

Experimental

Extension

Palette

User Interface

Layout

Media

- Camcorder
- Camera
- ImagePicker
- Player
- Sound
- SoundRecorder
- SpeechRecognizer
- TextToSpeech
- VideoPlayer
- YandexTranslate

Drawing and Animation

Sensors

Social

Storage

Connectivity

LEGO® MINDSTORMS®

Experimental

Palette

User Interface

Layout

Media

Drawing and Animation

- Ball
- Canvas
- ImageSprite

Sensors

Social

Storage

Connectivity

LEGO® MINDSTORMS®

Experimental

Extension

Palette

User Interface

Layout

Media

Drawing and Animation

Sensors

- AccelerometerSensor
- BarcodeScanner
- Clock
- GyroscopeSensor
- LocationSensor
- NearField
- OrientationSensor
- Pedometer
- ProximitySensor

Social

Storage








Connectivity





LEGO® MINDSTORMS®





Experimental















Extension

UI: designer palette

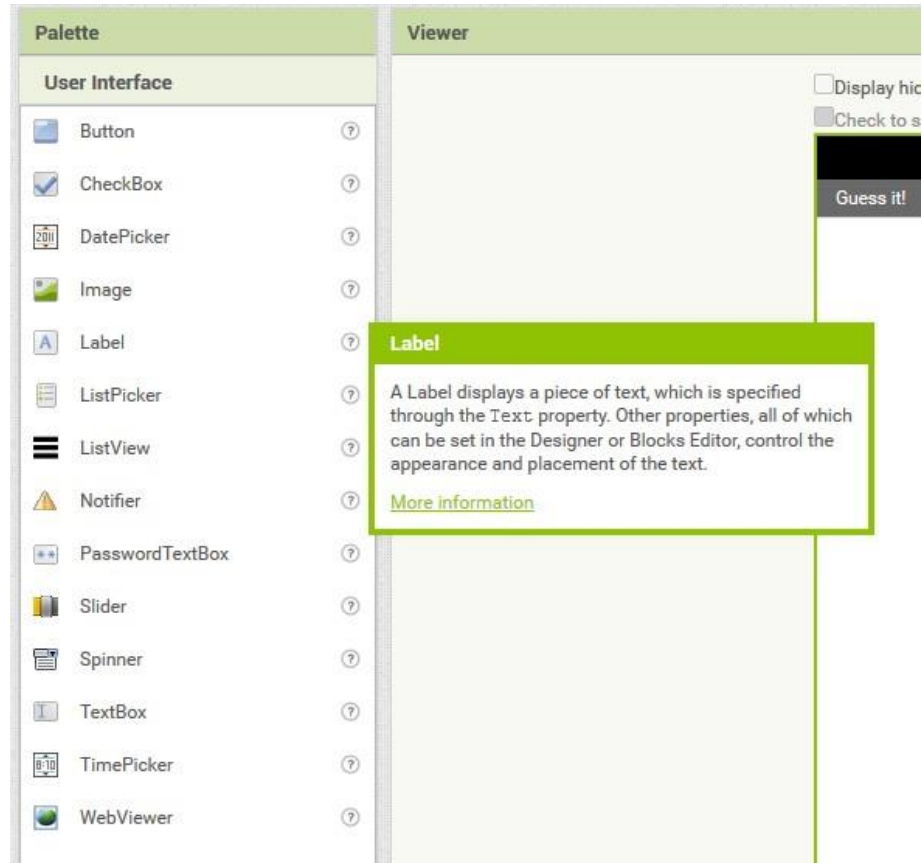
Palette	
User Interface	
Layout	
Media	
Drawing and Animation	
Sensors	
Social	
	ContactPicker ?
	EmailPicker ?
	PhoneCall ?
	PhoneNumberPicker ?
	Sharing ?
	Texting ?
	Twitter ?
Storage	
Connectivity	
LEGO® MINDSTORMS®	
Experimental	
Extension	

Palette	
User Interface	
Layout	
Media	
Drawing and Animation	
Sensors	
Social	
Storage	
	File ?
	FusionablesControl ?
	TinyDB ?
	TinyWebDB ?
Connectivity	
LEGO® MINDSTORMS®	
Experimental	
Extension	

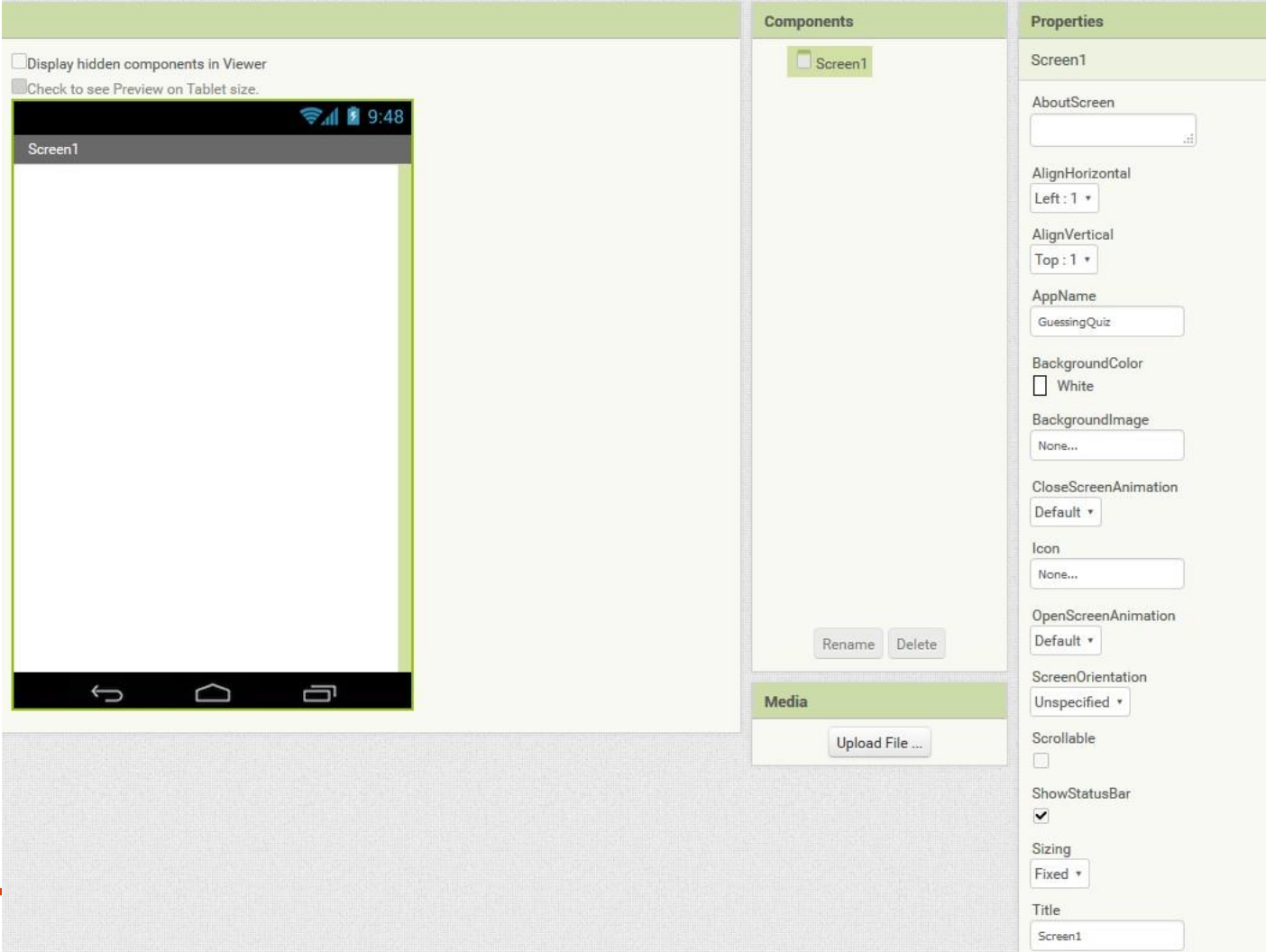
Palette	
User Interface	
Layout	
Media	
Drawing and Animation	
Sensors	
Social	
Storage	
Connectivity	
	ActivityStarter ?
	BluetoothClient ?
	BluetoothServer ?
	Web ?
LEGO® MINDSTORMS®	
Experimental	
Extension	

Palette	
User Interface	
Layout	
Media	
Drawing and Animation	
Sensors	
Social	
Storage	
Connectivity	
LEGO® MINDSTORMS®	
	NxtDrive ?
	NxtColorSensor ?
	NxtLightSensor ?
	NxtSoundSensor ?
	NxtTouchSensor ?
	NxtUltrasonicSensor ?
	NxtDirectCommands ?
	Ev3Motors ?
	Ev3ColorSensor ?
	Ev3GyroSensor ?
	Ev3TouchSensor ?
	Ev3UltrasonicSensor ?
	Ev3Sound ?
	Ev3UI ?

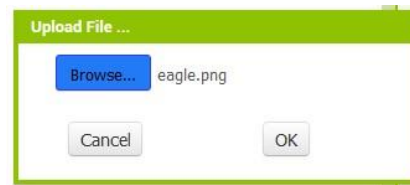
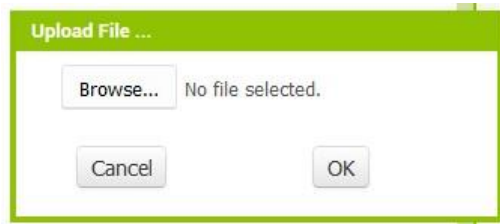
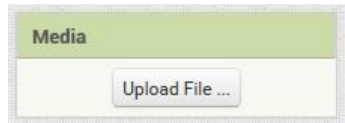
You can get “help” from each item in the palette menu by clicking “?” button



Adding resource, for ex. Icon, sound, image



The screenshot displays the Android Studio interface. On the left, a preview window shows a mobile screen with a status bar at the top displaying signal strength, Wi-Fi, battery, and the time 9:48. The screen content is labeled "Screen1" and is currently blank. Below the preview are navigation icons for back, home, and recent apps. On the right, the "Properties" panel is open for "Screen1". It lists various attributes: "AboutScreen" (text input), "AlignHorizontal" (Left), "AlignVertical" (Top), "AppName" (GuessingQuiz), "BackgroundColor" (White), "BackgroundImage" (None...), "CloseScreenAnimation" (Default), "Icon" (None...), "OpenScreenAnimation" (Default), "ScreenOrientation" (Unspecified), "Scrollable" (unchecked), "ShowStatusBar" (checked), "Sizing" (Fixed), and "Title" (Screen1). Below the Properties panel is a "Media" section with an "Upload File ..." button. The "Components" panel on the right shows "Screen1" as the selected component, with "Rename" and "Delete" buttons below it.



Properties

Screen1

AboutScreen

AlignHorizontal

Left : 1 ▾

AlignVertical

Top : 1 ▾

AppName

GuessingQuiz

BackgroundColor

White

BackgroundImage

None...

CloseScreenAnimation

Default ▾

Icon

None
eagle.png

Upload File ...

Cancel OK

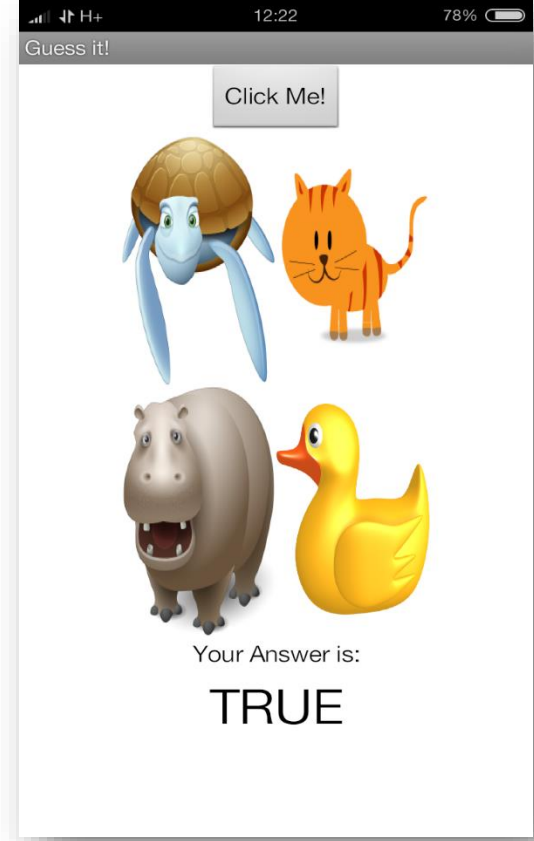
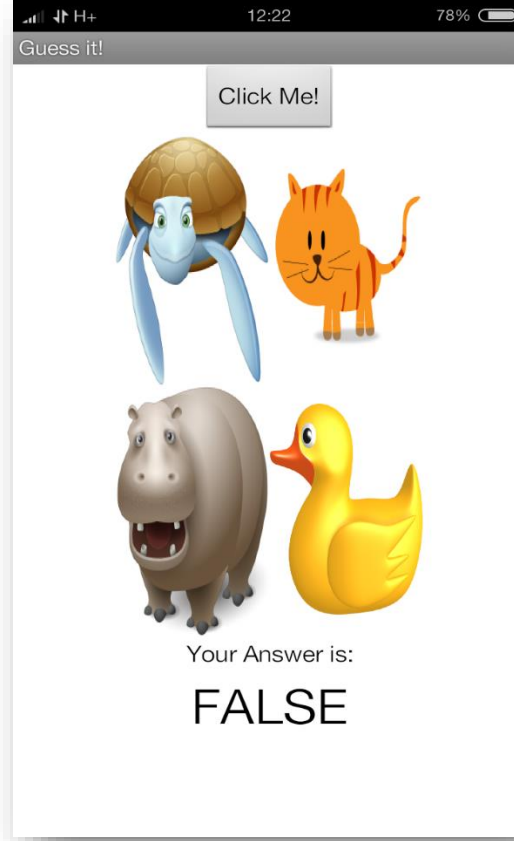
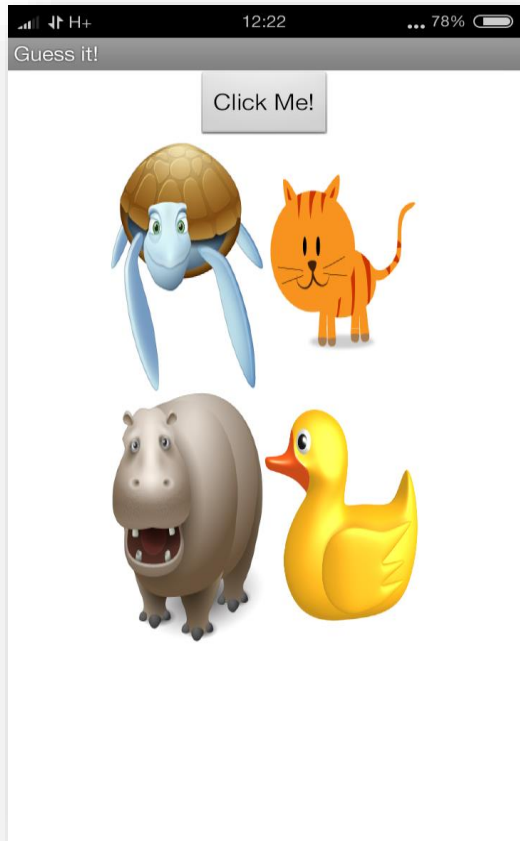
Sizing

Fixed ▾

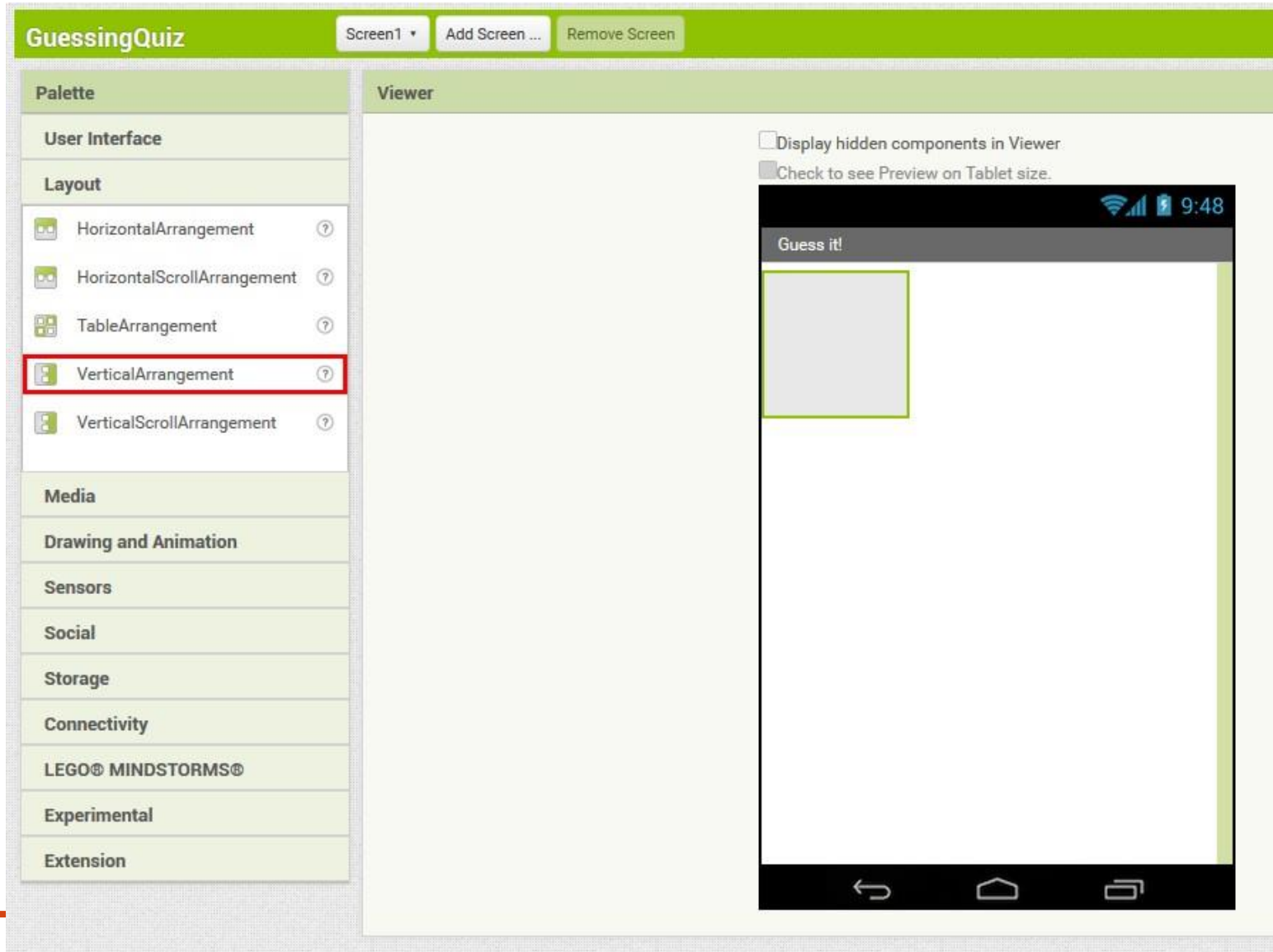
Title

Guess it!

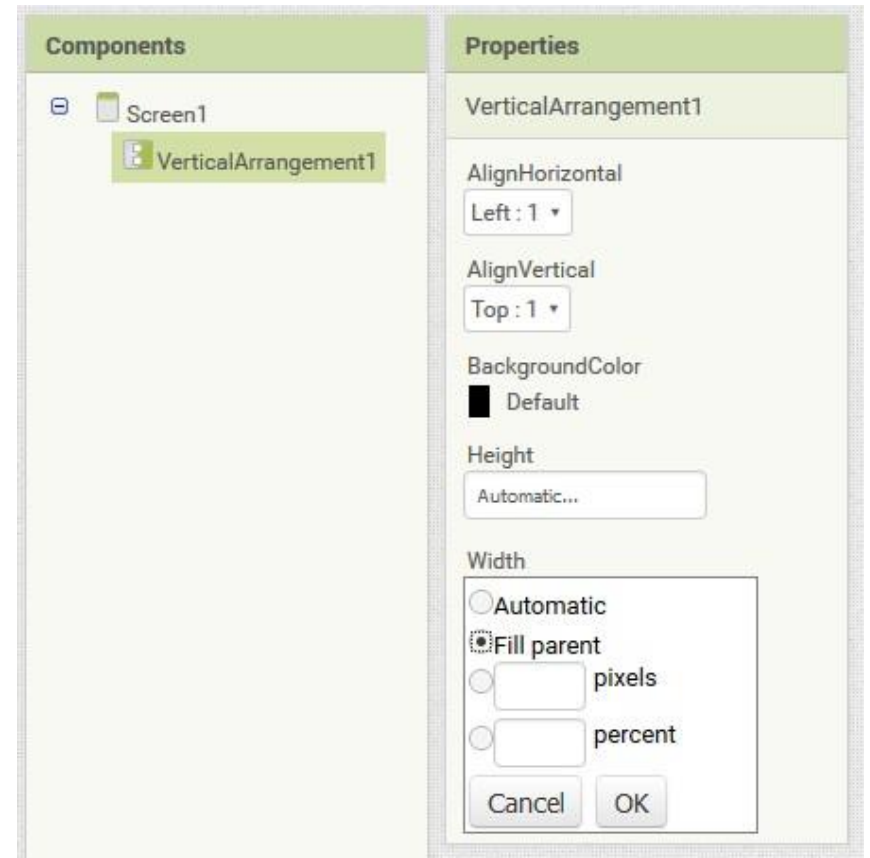
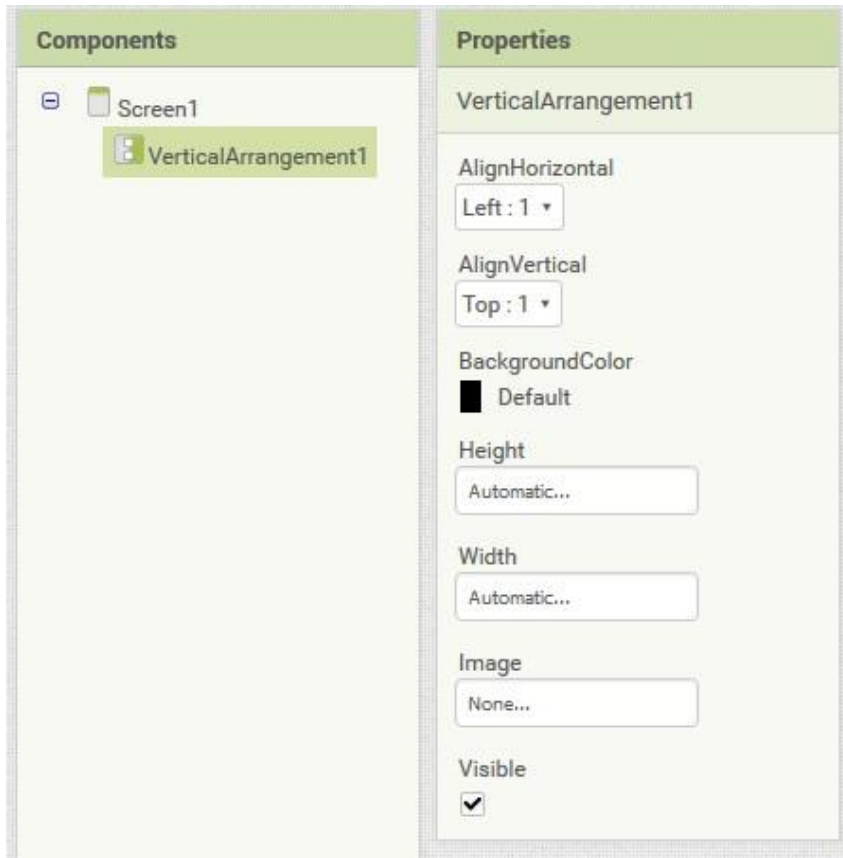
Example: Simple Quiz App



Using layout: vertical arrangement



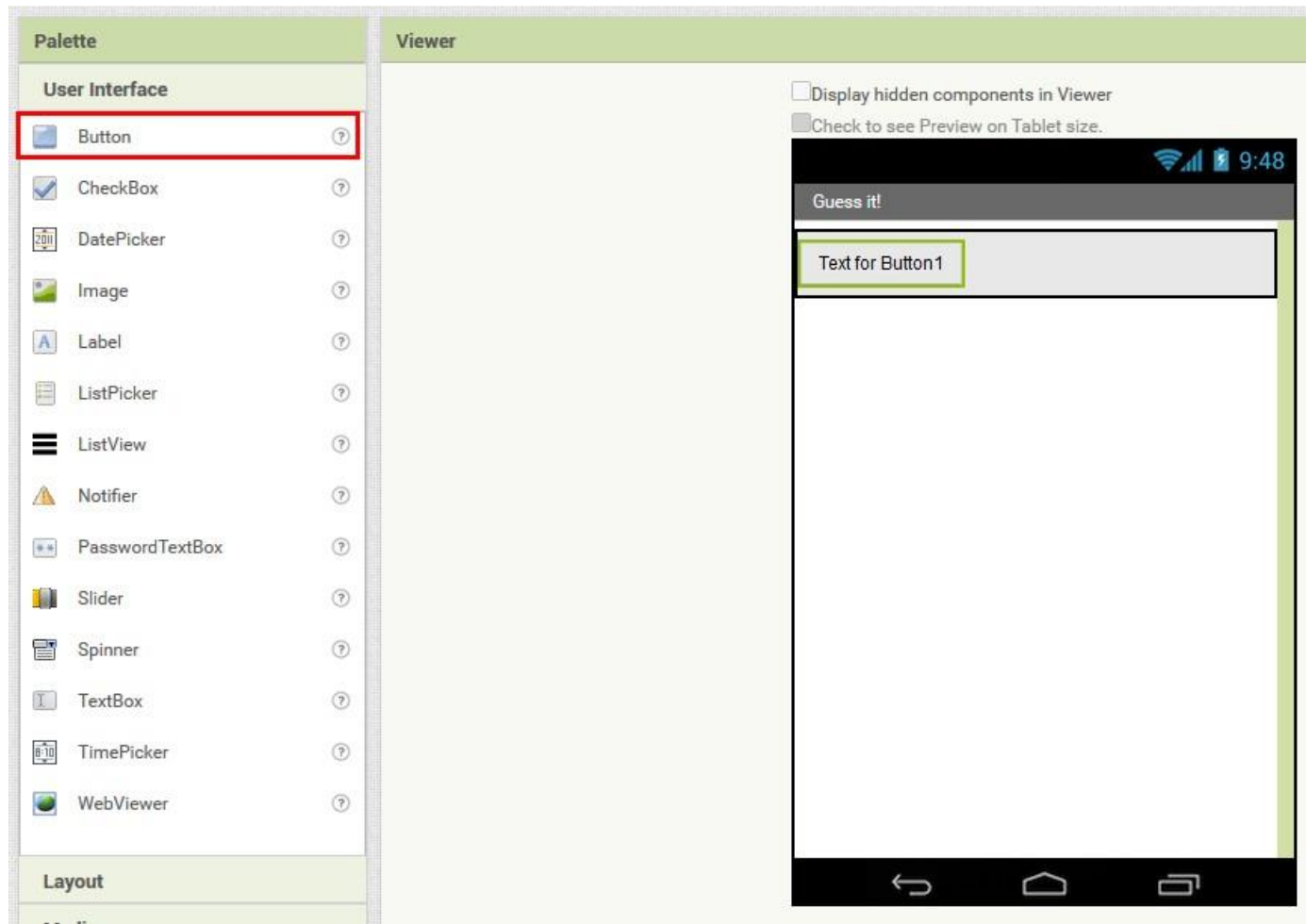
The screenshot displays the Android Studio interface for a project named "GuessingQuiz". The top bar shows "Screen1" and buttons for "Add Screen ..." and "Remove Screen". The left sidebar, labeled "Palette", is organized into categories: "User Interface", "Layout", "Media", "Drawing and Animation", "Sensors", "Social", "Storage", "Connectivity", "LEGO® MINDSTORMS®", "Experimental", and "Extension". Under the "Layout" category, several options are listed: "HorizontalArrangement", "HorizontalScrollArrangement", "TableArrangement", "VerticalArrangement" (highlighted with a red border), and "VerticalScrollArrangement". The right sidebar, labeled "Viewer", contains two checkboxes: "Display hidden components in Viewer" (unchecked) and "Check to see Preview on Tablet size." (checked). Below these is a preview of a mobile device screen. The screen has a black status bar at the top with icons for Wi-Fi, signal strength, and battery, and the time "9:48". The main content area of the screen has a grey header with the text "Guess it!". Below the header is a large, empty white rectangular area, which is the visual representation of the selected "VerticalArrangement" layout. At the bottom of the screen is a black navigation bar with three white icons: a back arrow, a home circle, and a recent apps square.



Set the layout so that the **width will fill the parent**

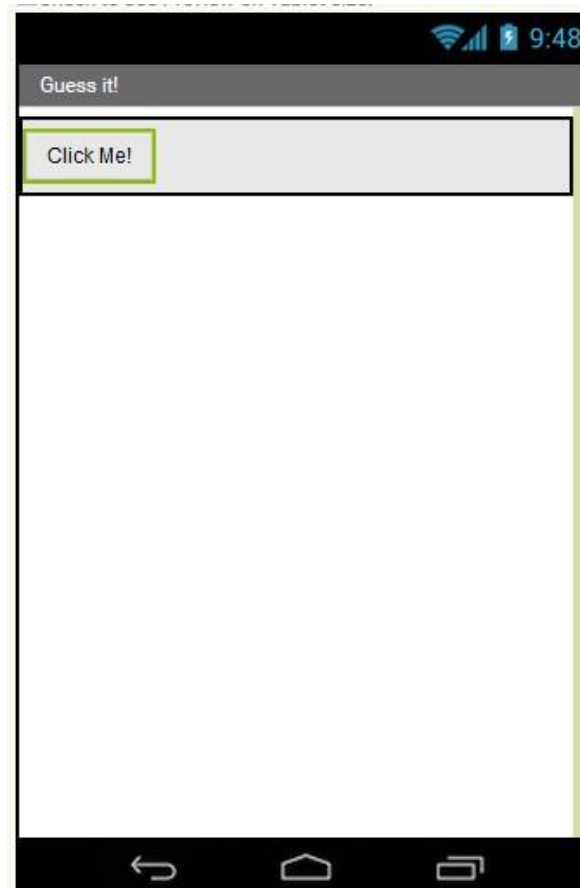


Add button from palette



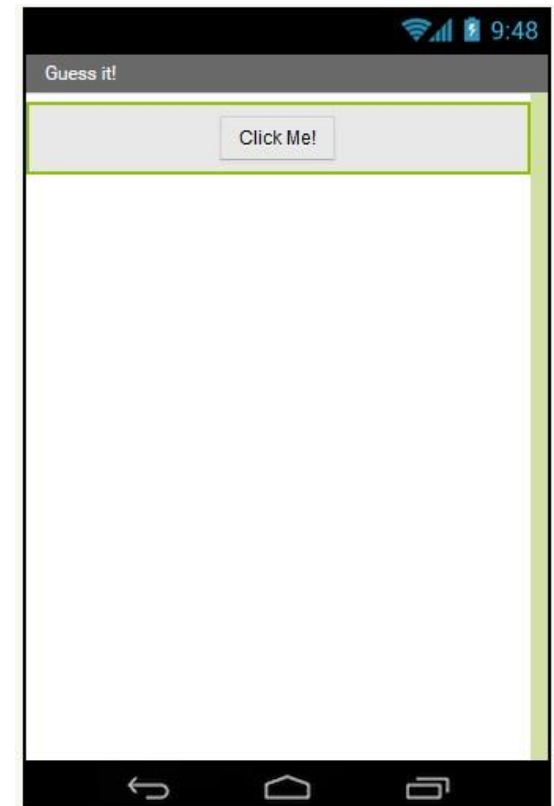
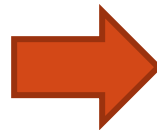
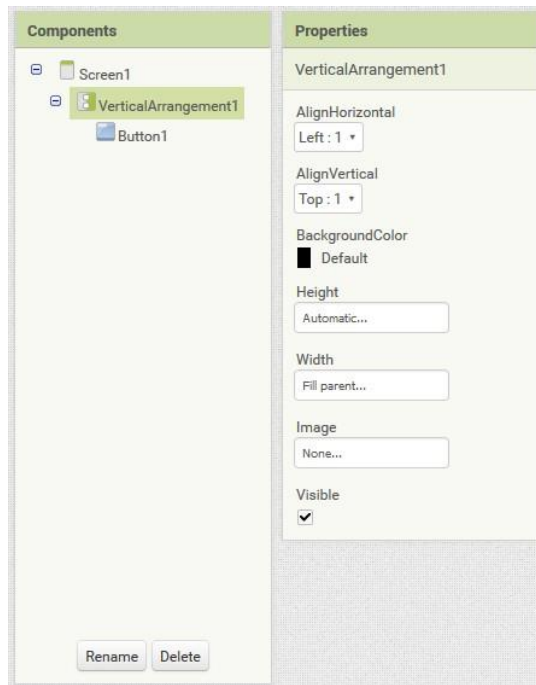


Then change the text caption of the button

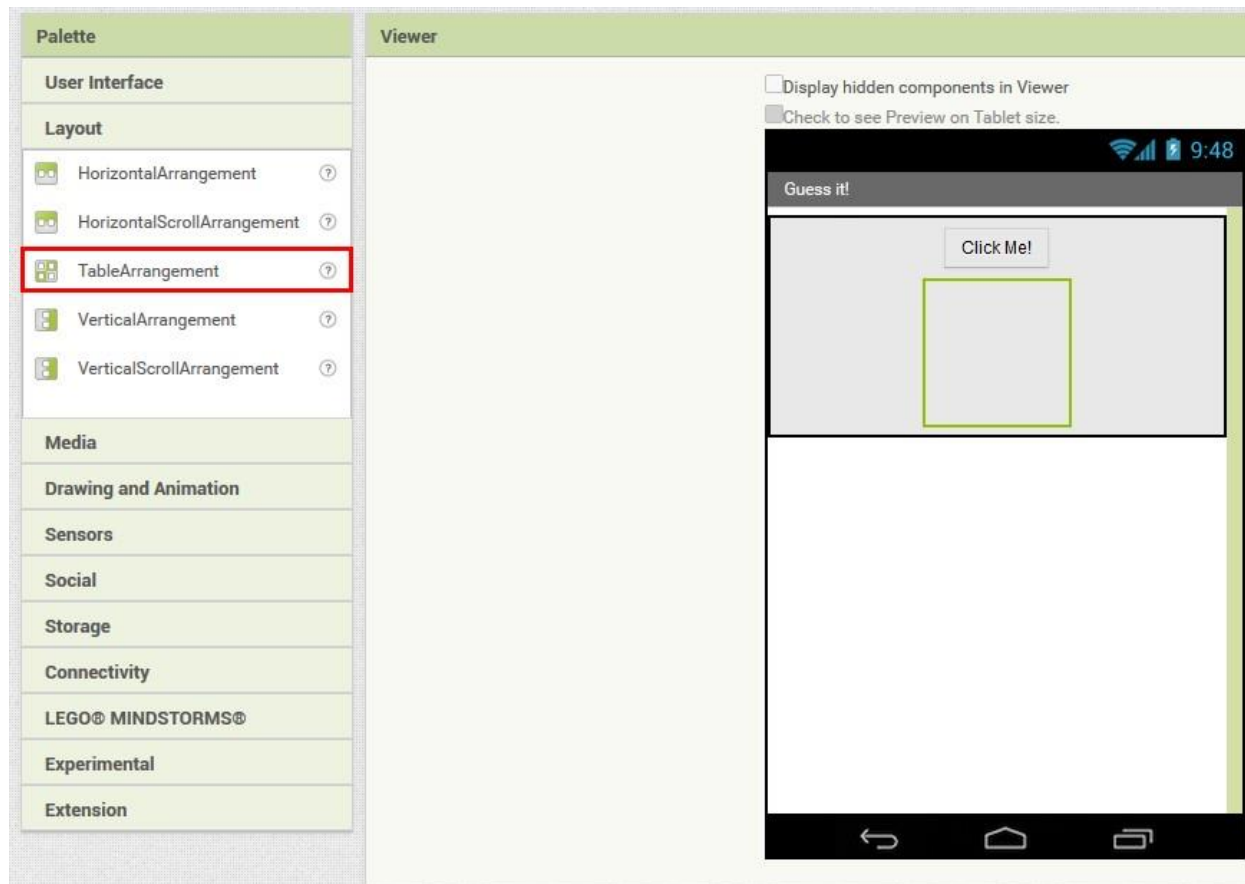


Now let's make the button **centered** in the layout. Go to previously added vertical layout, then **click align horizontal**

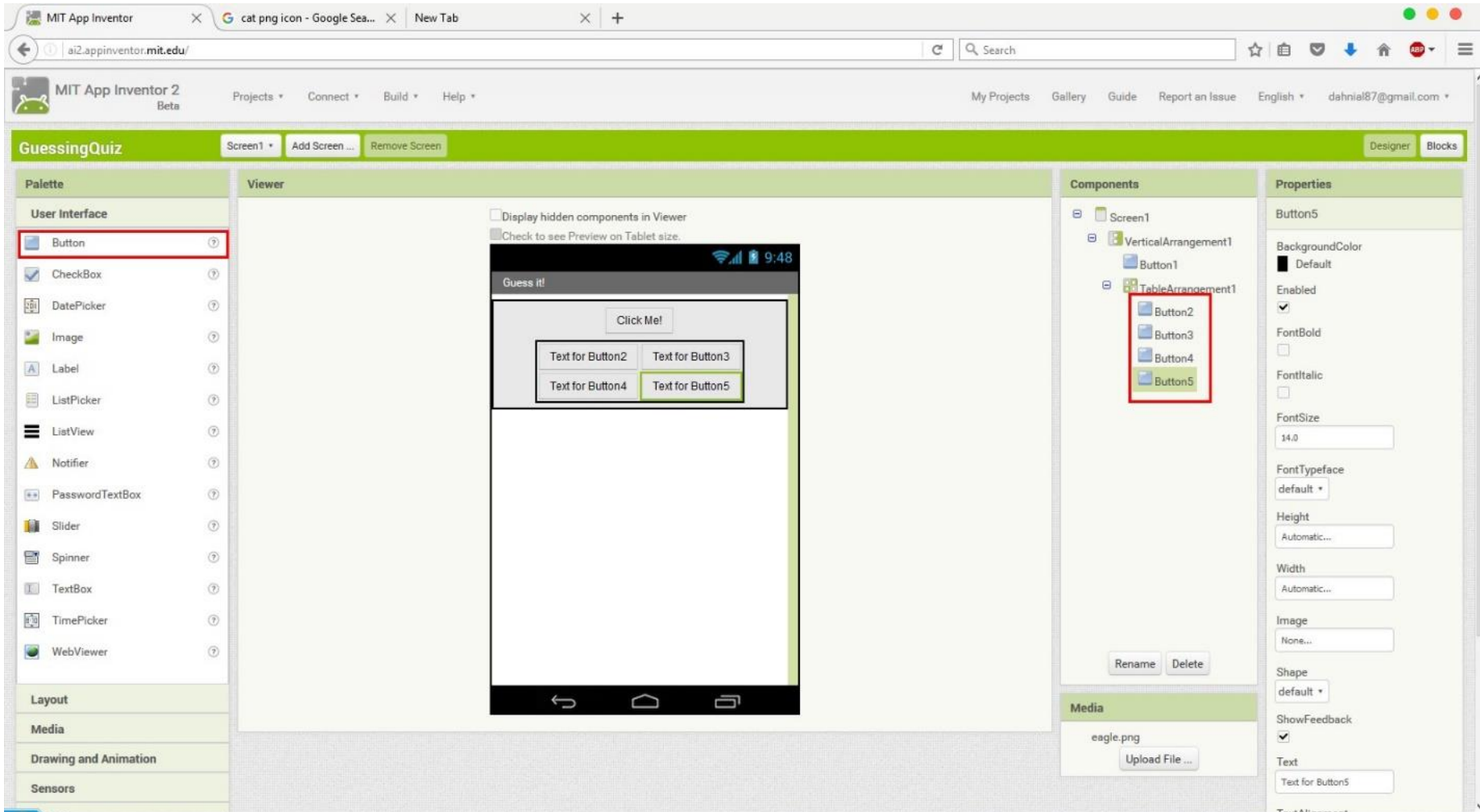
RESULT:



We will add 4 button to be used as option. Add “table arrangement” layout first.



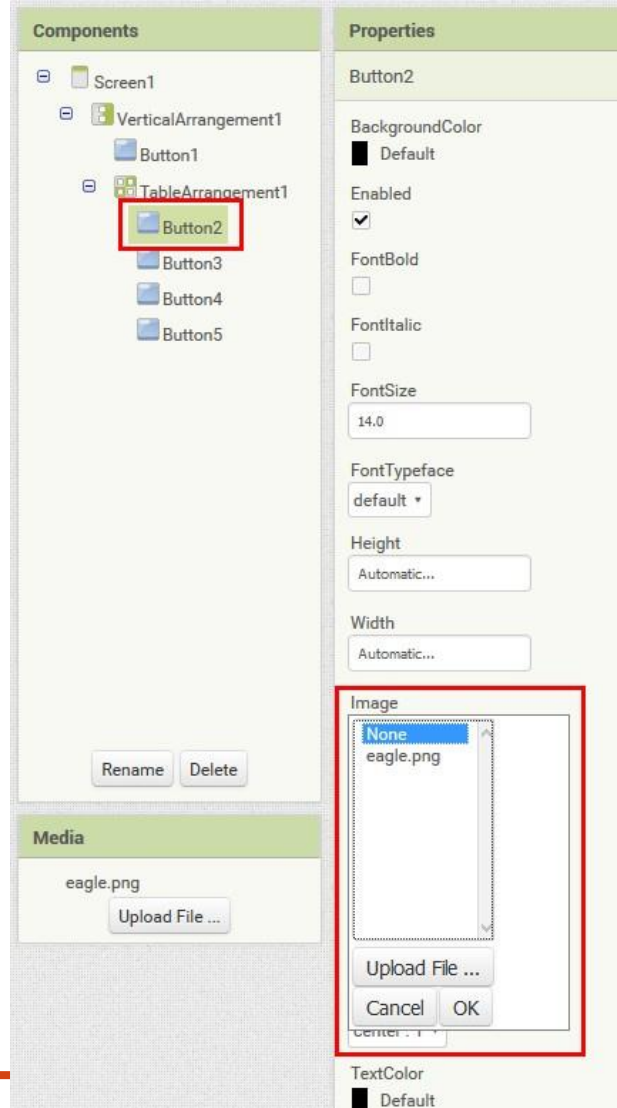
Then add the button for each



The screenshot displays the MIT App Inventor 2 web interface. The browser address bar shows `ai2.appinventor.mit.edu/`. The application title is "GuessingQuiz". The interface is divided into several panels:

- Palette:** Lists various UI components. The "Button" component is highlighted with a red box.
- Viewer:** Shows a mobile device preview of the app. The app has a title "Guess it!" and a "Click Me!" button. Below it are five text boxes labeled "Text for Button2" through "Text for Button5".
- Components:** A tree view showing the app's structure. A "TableArrangement1" component is highlighted with a red box, containing five buttons: "Button2", "Button3", "Button4", and "Button5".
- Properties:** Shows the properties for the selected "Button5" component, including "BackgroundColor", "Enabled", "FontSize", "FontTypeface", "Height", "Width", "Image", "Shape", "ShowFeedback", and "Text".

We will add custom **image** for each **button**.



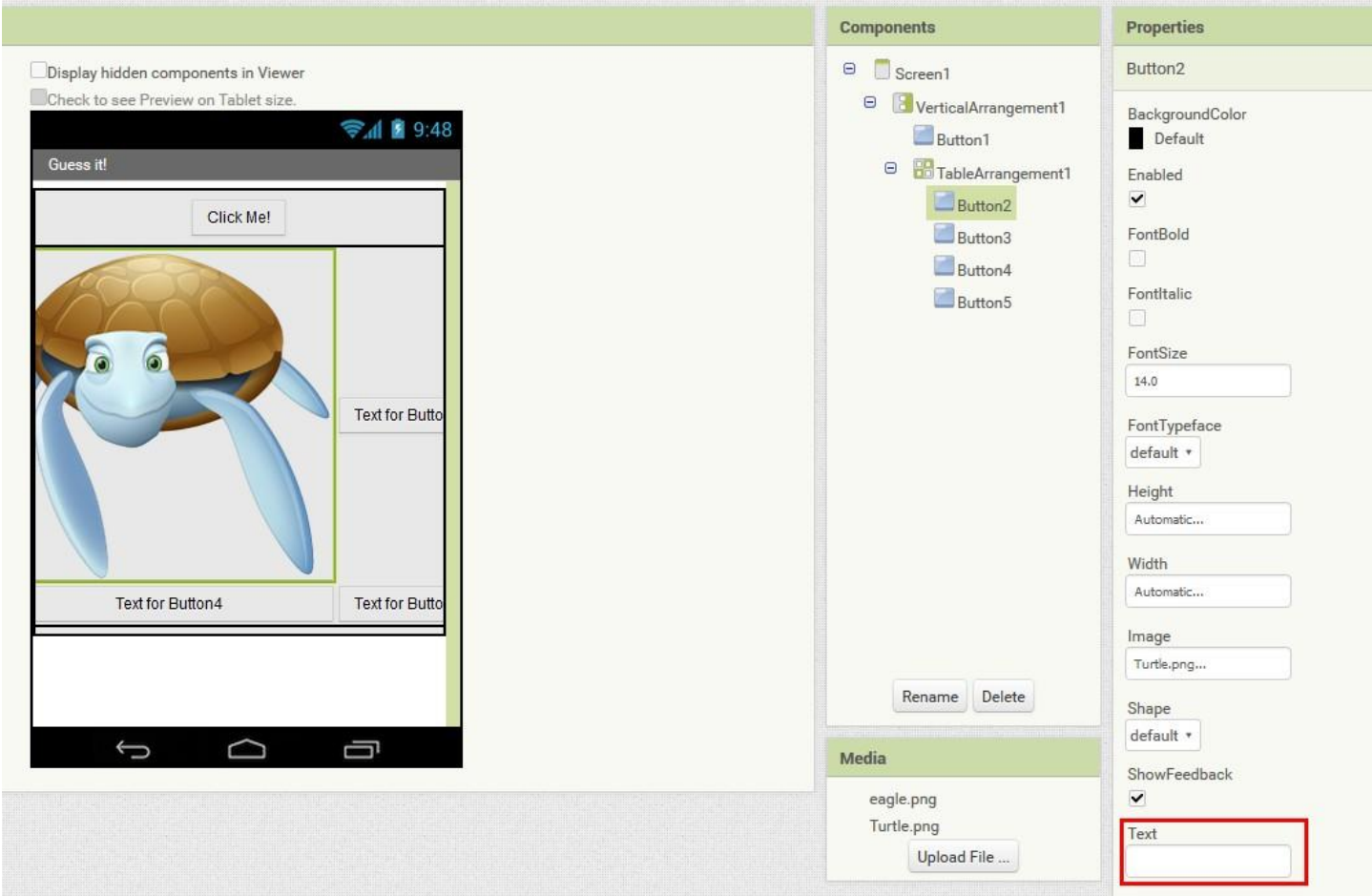
The screenshot displays a software development interface with two main panels: Components and Properties.

Components Panel: Shows a hierarchical tree structure. Under "Screen1", there is a "VerticalArrangement1" containing "Button1". Below that is a "TableArrangement1" containing "Button2", "Button3", "Button4", and "Button5". The "Button2" component is highlighted with a red box.

Properties Panel: Shows the properties for the selected "Button2" component. The "Image" property is highlighted with a red box and is currently set to "None". Below the list, there is an "Upload File ..." button, "Cancel", and "OK" buttons.

Media Panel: Located at the bottom, it shows a list of media files. "eagle.png" is listed, and there is an "Upload File ..." button next to it.

And **remove the text caption** in the button



The screenshot displays the Android Studio interface. On the left, a mobile app preview shows a screen with the title "Guess it!". Below the title is a button labeled "Click Me!". In the center is a large image of a cartoon turtle. To the right of the turtle are two text labels: "Text for Butto" and "Text for Button4". At the bottom of the preview are three Android navigation icons.

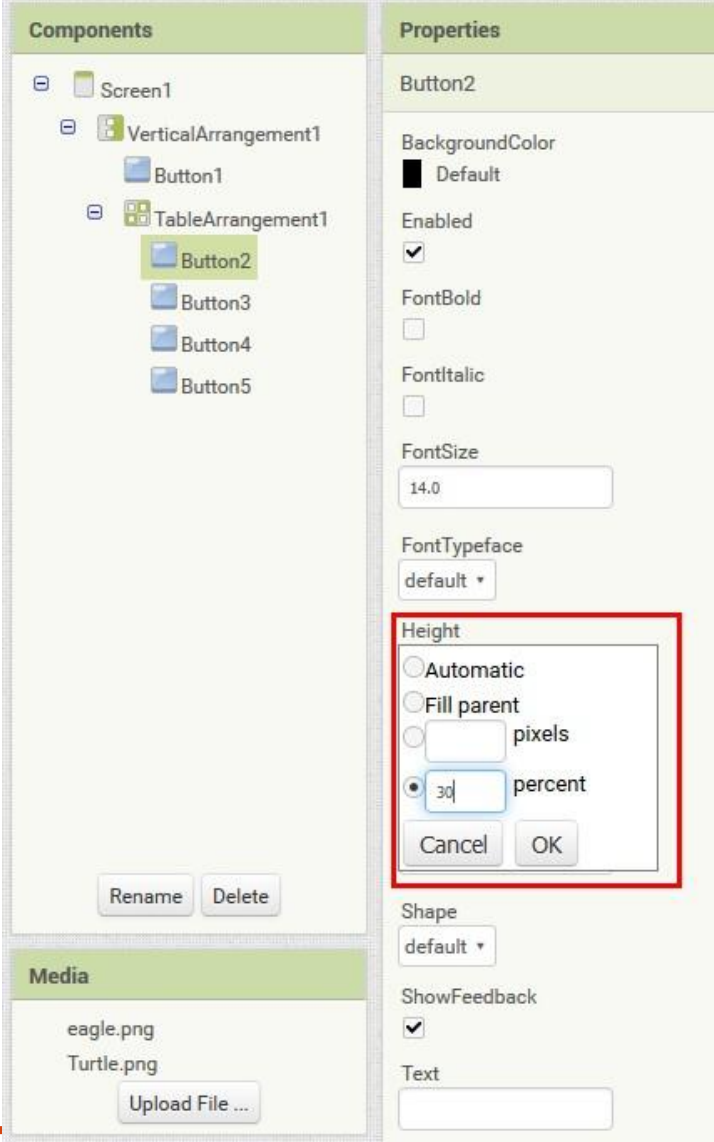
On the right side, the "Properties" panel is open for "Button2". The "Text" property is highlighted with a red box and is currently empty. Other visible properties include:

- Background Color: Default
- Enabled:
- Font Bold:
- Font Italic:
- Font Size: 14.0
- Font Typeface: default
- Height: Automatic
- Width: Automatic
- Image: Turtle.png
- Shape: default
- Show Feedback:

The "Components" panel shows a tree view with "Screen1" containing "VerticalArrangement1", which contains "Button1" and "TableArrangement1". "TableArrangement1" contains "Button2", "Button3", "Button4", and "Button5".

The "Media" panel at the bottom shows "eagle.png" and "Turtle.png" with an "Upload File ..." button.

Resize the button a little bit ...



The screenshot displays a software development interface with two main panels: Components and Properties.

Components Panel: Shows a hierarchical tree structure. Under 'Screen1', there is a 'VerticalArrangement1' containing 'Button1'. Below that is a 'TableArrangement1' containing 'Button2', 'Button3', 'Button4', and 'Button5'. 'Button2' is highlighted with a green background.

Properties Panel: Shows the configuration for 'Button2'. The 'Height' property is highlighted with a red box. It is set to '30 percent'.

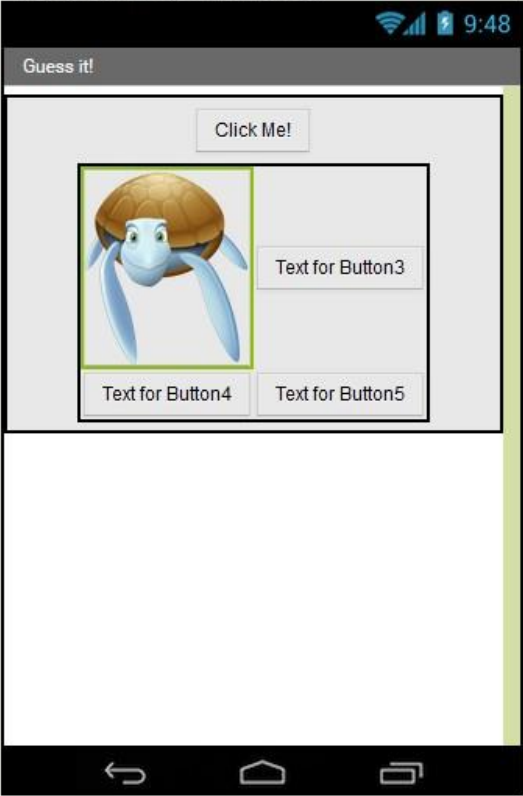
Media Panel: Located at the bottom, it contains 'eagle.png' and 'Turtle.png' with an 'Upload File ...' button.

Properties List:

- Button2
- BackgroundColor: Default
- Enabled:
- FontBold:
- FontItalic:
- FontSize: 14.0
- FontTypeface: default
- Height: Automatic, Fill parent, [] pixels, 30 percent
- Shape: default
- ShowFeedback:
- Text: []


Resize for the **height** and **width**

Display hidden components in Viewer
 Check to see Preview on Tablet size.



Guess it!

Click Me!



Text for Button3

Text for Button4

Text for Button5

Components

- Screen1
 - VerticalArrangement1
 - Button1
 - TableArrangement1
 - Button2
 - Button3
 - Button4
 - Button5

Properties

Button2

BackgroundColor
Default

Enabled

FontBold

FontItalic

FontSize
14.0

FontTypeface
default

Height
30 percent...

Width
30 percent...

Image
Turtle.png...

Shape
default

ShowFeedback

Text

Media

- eagle.png
- Turtle.png

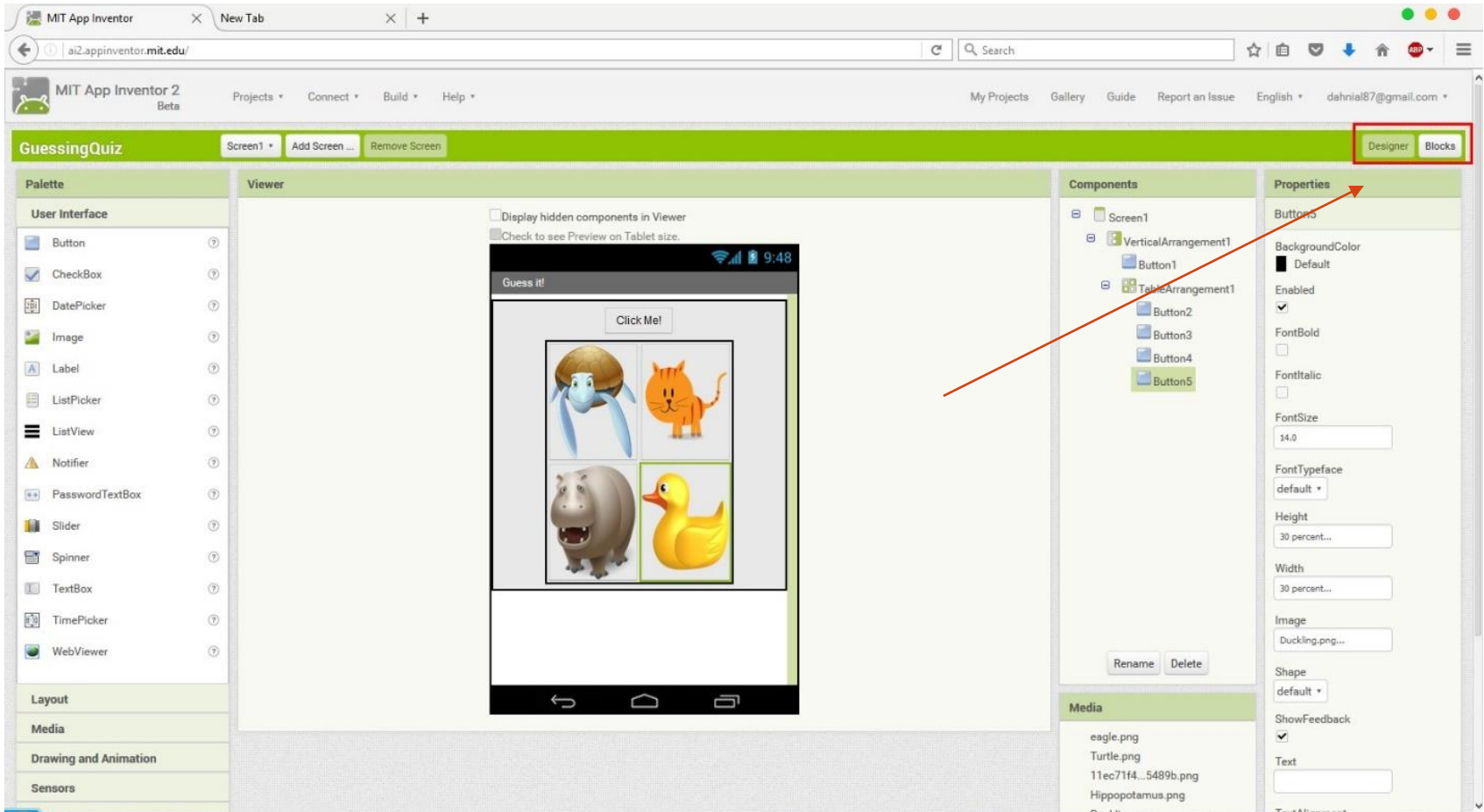
Upload File ...

Rename Delete

UI Result:

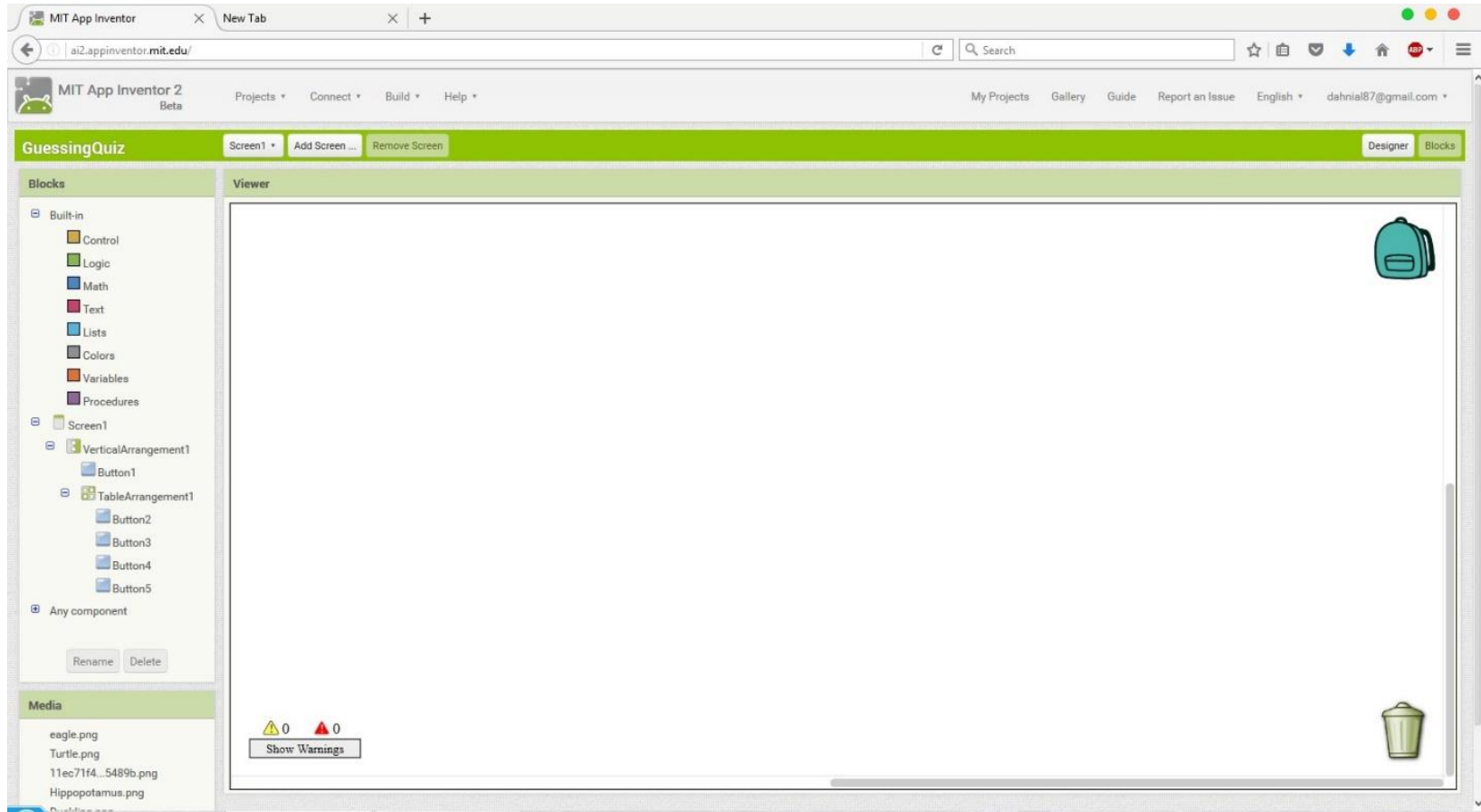


Now we will Program the function. Click the “blocks” button



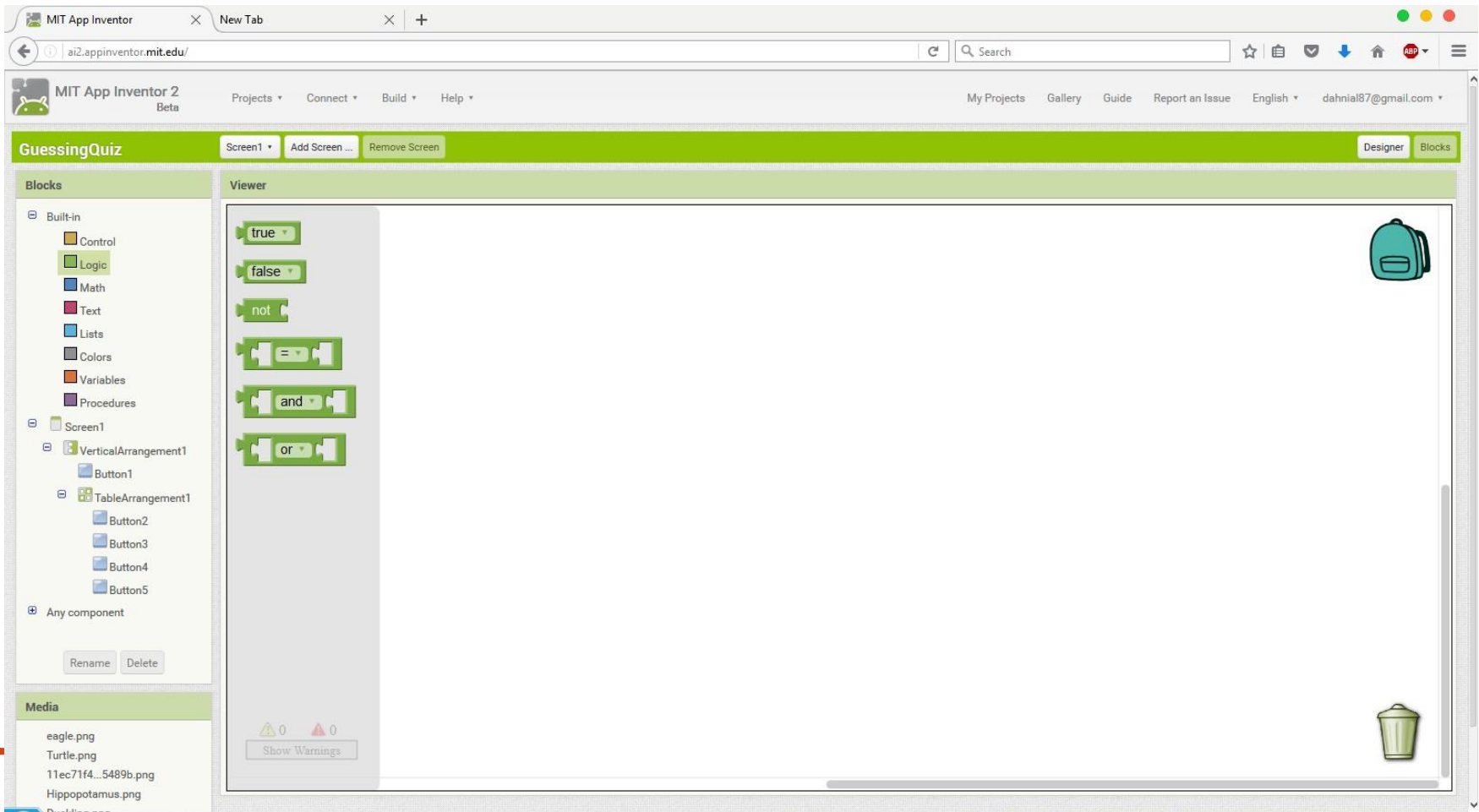
The screenshot displays the MIT App Inventor 2 web interface. The browser address bar shows 'ai2.appinventor.mit.edu/'. The page title is 'MIT App Inventor 2 Beta'. The main interface is titled 'GuessingQuiz' and includes a 'Designer' tab and a 'Blocks' tab. The 'Blocks' tab is highlighted with a red box, and a red arrow points to it from the right. The interface is divided into several panels: 'Palette' (User Interface, Layout, Media, Drawing and Animation, Sensors), 'Viewer' (displaying a mobile app preview with a 'Guess it!' screen and a 'Click Me!' button), 'Components' (listing various UI elements like Button1, Button2, etc.), and 'Properties' (showing settings for the selected component, such as BackgroundColor, Enabled, FontSize, etc.).

User Interface: Blocks



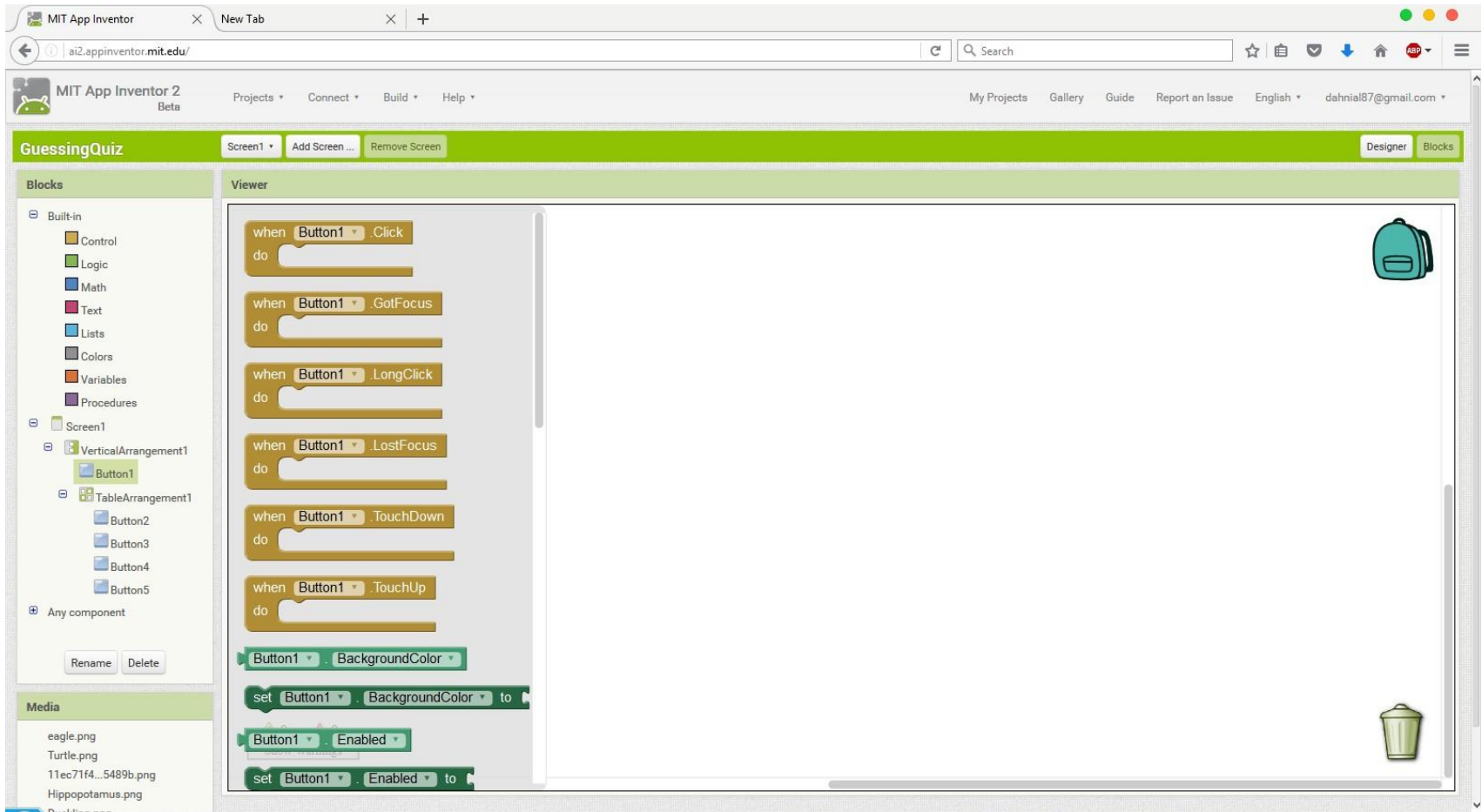
Appinventor uses **block programming** instead of scripting. Let's take a look:

Logic Blocks



The screenshot displays the MIT App Inventor 2 Beta web interface. The browser address bar shows `ai2.appinventor.mit.edu/`. The application title is "MIT App Inventor 2 Beta". The main workspace is titled "GuessingQuiz" and shows a "Designer" view. The "Blocks" palette on the left is expanded to the "Logic" category, showing various logic blocks like "true", "false", "not", "and", and "or". The "Viewer" on the right shows a preview of the application with a blue backpack icon. The "Media" palette at the bottom left lists image files like "eagle.png", "Turtle.png", "11ec71f4...5489b.png", and "Hippopotamus.png".

Block from a Button in UI screen



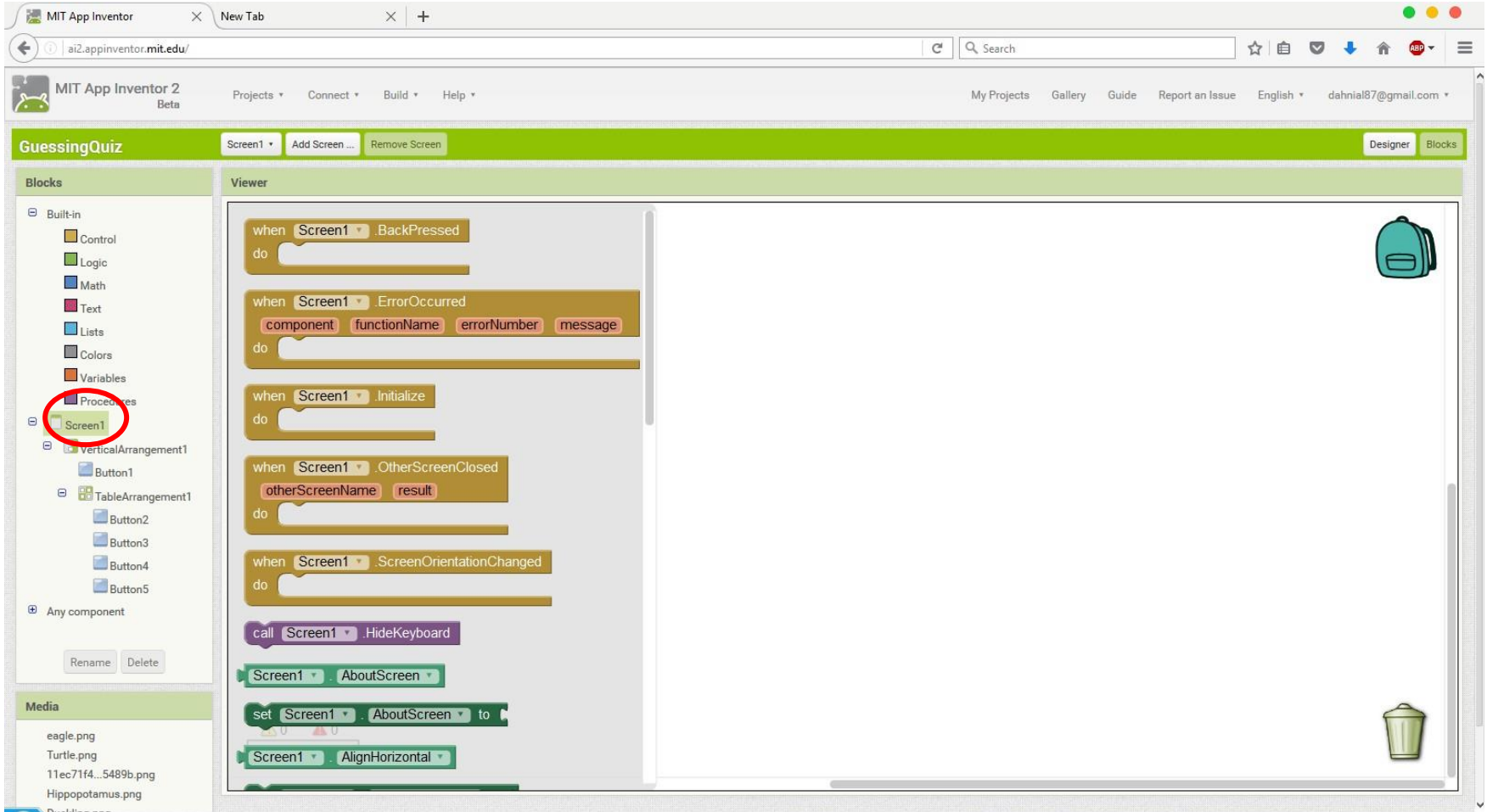
The screenshot displays the MIT App Inventor 2 web interface. The browser address bar shows `ai2.appinventor.mit.edu/`. The page title is "GuessingQuiz" and the current screen is "Screen1". The interface is divided into three main sections:

- Blocks:** A sidebar on the left containing categories like "Built-in" (Control, Logic, Math, Text, Lists, Colors, Variables, Procedures), "Screen1" (VerticalArrangement1, Button1, TableArrangement1, Button2, Button3, Button4, Button5), and "Any component".
- Viewer:** A central workspace showing a list of event blocks for "Button1":
 - when Button1 . Click
 - when Button1 . GotFocus
 - when Button1 . LongClick
 - when Button1 . LostFocus
 - when Button1 . TouchDown
 - when Button1 . TouchUp
- Designer:** A large empty area on the right where the visual representation of the app screen is built. It contains a blue backpack icon and a trash can icon.

At the bottom of the blocks list, there are additional blocks for "Button1":

- Button1 . BackgroundColor
- set Button1 . BackgroundColor to
- Button1 . Enabled
- set Button1 . Enabled to

Block from a screen



The screenshot shows the MIT App Inventor 2 web interface. The browser address bar displays `ai2.appinventor.mit.edu/`. The page title is "MIT App Inventor 2 Beta". The main workspace is titled "GuessingQuiz" and shows a "Screen1" selected in the "Blocks" panel. The "Screen1" block is circled in red. The "Viewer" panel displays the code for "Screen1", including the following blocks:

```
when Screen1 . BackPressed
do

when Screen1 . ErrorOccurred
component functionName errorNumber message
do

when Screen1 . Initialize
do

when Screen1 . OtherScreenClosed
otherScreenName result
do

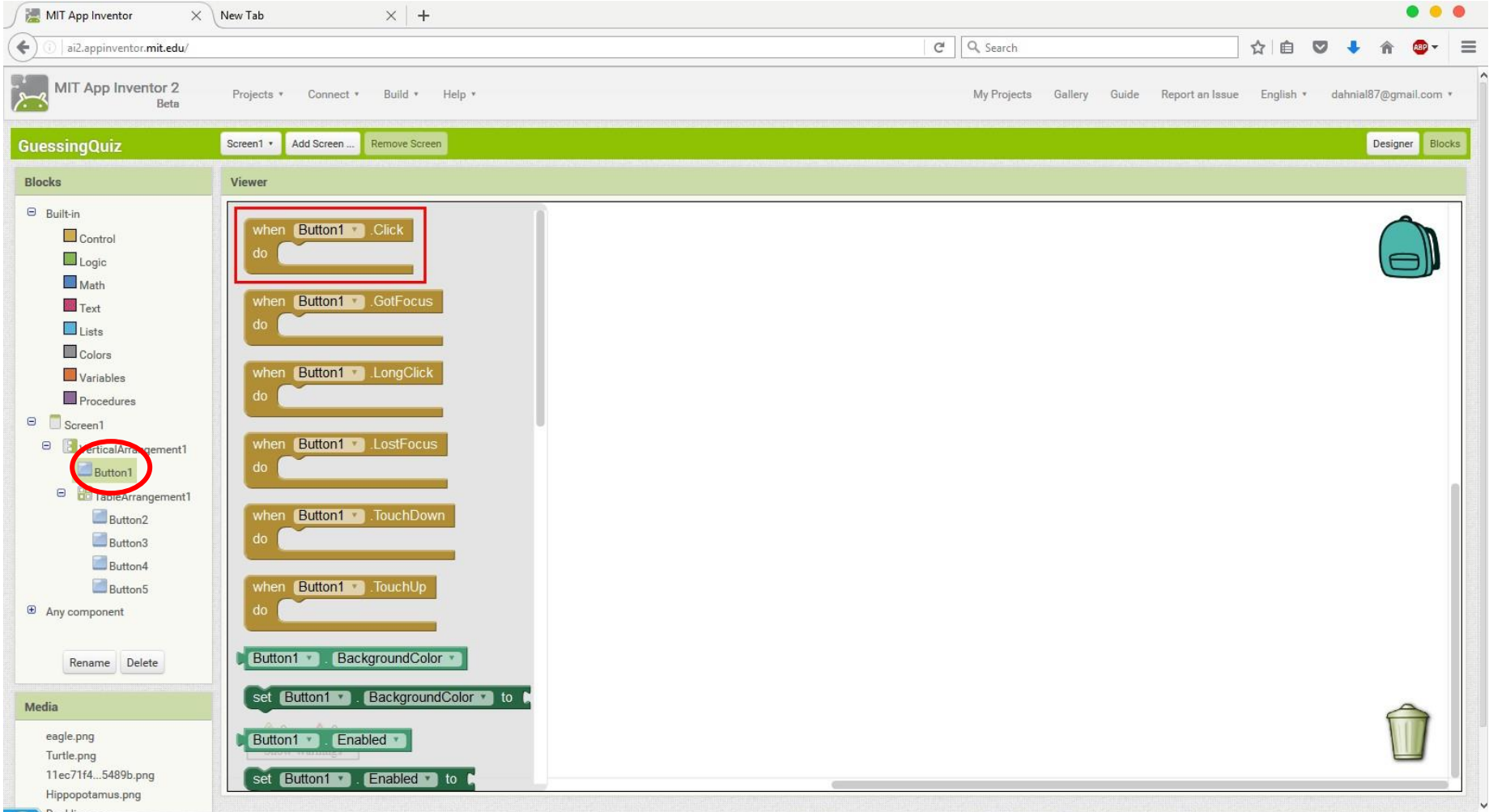
when Screen1 . ScreenOrientationChanged
do

call Screen1 . HideKeyboard

Screen1 . AboutScreen
set Screen1 . AboutScreen to
Screen1 . AlignHorizontal
```

The "Media" panel at the bottom left lists several image files: `eagle.png`, `Turtle.png`, `11ec71f4...5489b.png`, and `Hippopotamus.png`.

For example, a function **when button1 is clicked**:

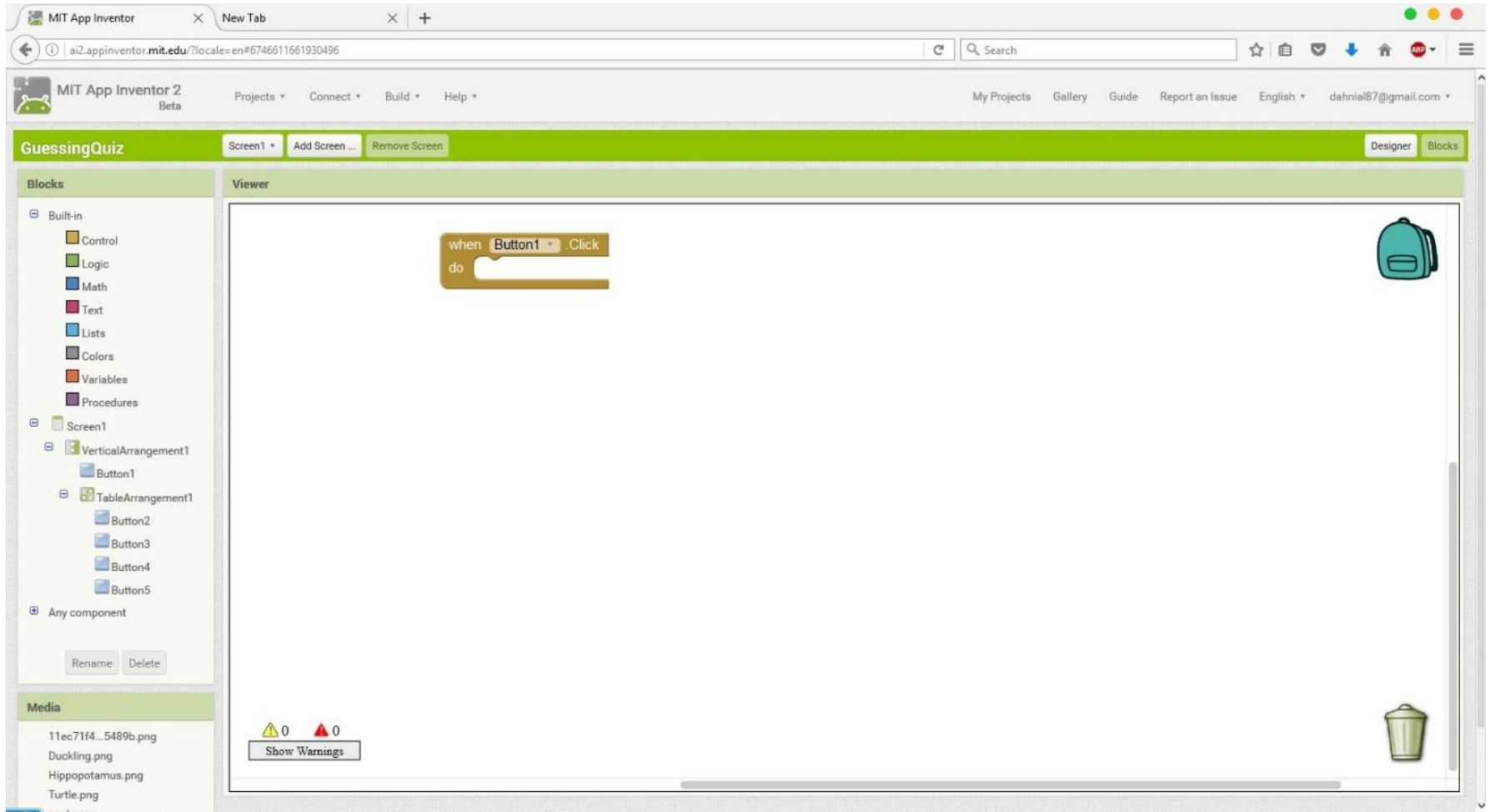


The screenshot displays the MIT App Inventor 2 web interface. The browser address bar shows `ai2.appinventor.mit.edu/`. The page title is "MIT App Inventor 2 Beta". The project name is "GuessingQuiz". The interface is divided into three main sections: "Blocks", "Viewer", and "Designer".

In the "Blocks" section, the "Built-in" category is expanded, and the "Logic" sub-category is selected. The "when Button1 .Click" block is highlighted with a red rectangle. The "Screen1" section is also expanded, and the "Button1" component is circled in red.

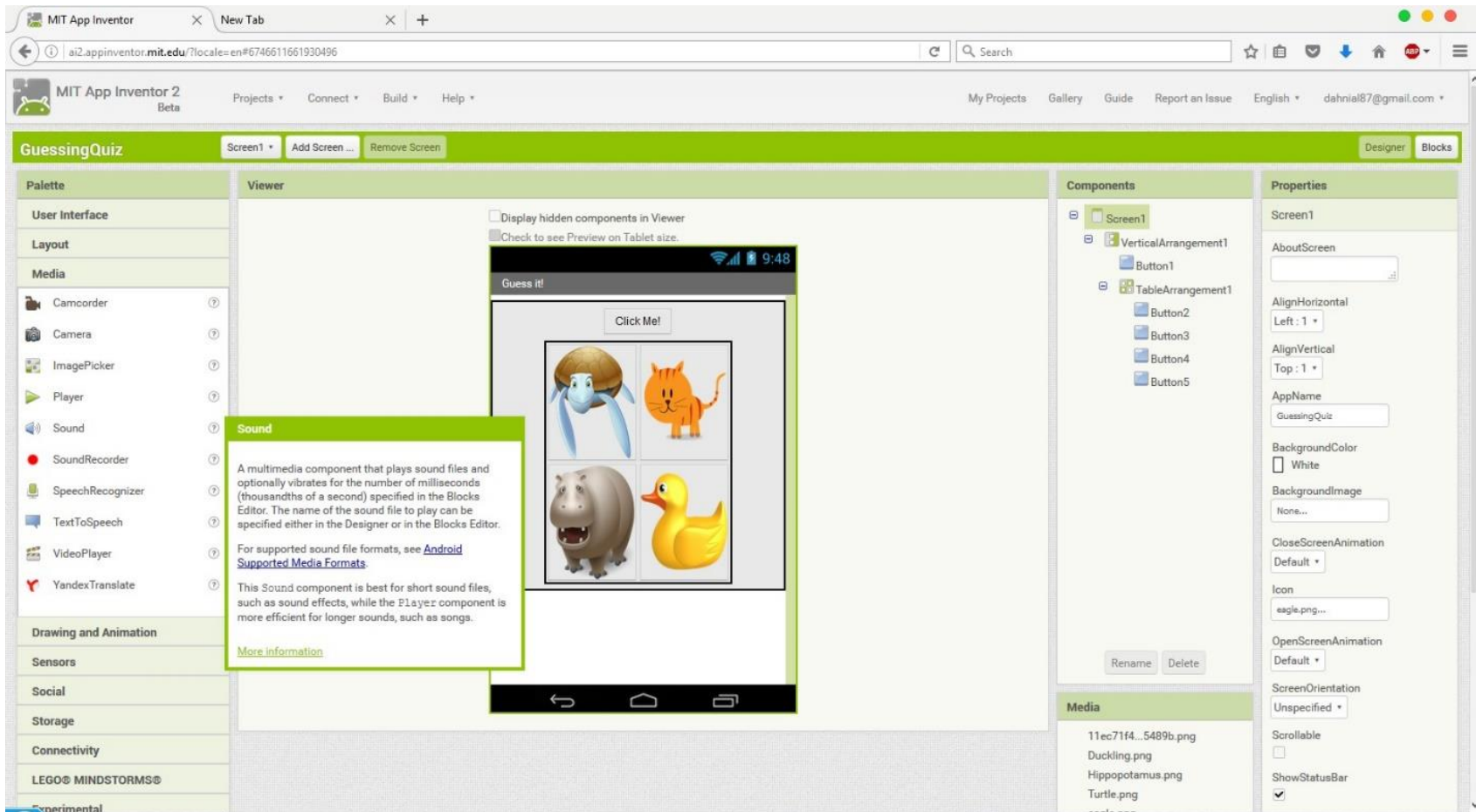
In the "Viewer" section, the "when Button1 .Click" block is highlighted with a red rectangle. Below it, several other event blocks are visible, including "when Button1 .GotFocus", "when Button1 .LongClick", "when Button1 .LostFocus", "when Button1 .TouchDown", and "when Button1 .TouchUp". Below these, there are several "set Button1 .BackgroundColor" and "set Button1 .Enabled" blocks.

Drag and drop to the screen



The screenshot shows the MIT App Inventor 2 web interface. The browser address bar displays `ai2.appinventor.mit.edu/?locale=en#6746611661930496`. The application title is "MIT App Inventor 2 Beta". The main workspace is titled "GuessingQuiz" and shows a "Screen1" tab. The "Blocks" panel on the left lists categories: Built-in (Control, Logic, Math, Text, Lists, Colors, Variables, Procedures), Screen1 (VerticalArrangement1, TableArrangement1), and Any component. The "Viewer" panel shows a single block: "when Button1 Click" with a "do" block below it. The "Media" panel at the bottom left lists image files: "11ec71f4...5489b.png", "Duckling.png", "Hippopotamus.png", and "Turtle.png". A "Show Warnings" button is visible at the bottom of the viewer area.

App: when “Click me” button is clicked we want it to **play a wav sound**.
We need to **add sound component** from “media” palette



The screenshot shows the MIT App Inventor 2 interface. The main workspace displays a mobile app preview titled "GuessingQuiz" with a "Click Me!" button and a 2x2 grid of animal images (a turtle, a cat, a hippo, and a duck). The left sidebar contains the "Media" palette, where the "Sound" component is highlighted. A tooltip for the "Sound" component is visible, providing details about its functionality. The right sidebar shows the "Properties" panel for the selected "Sound" component, with fields for "SoundFile" (set to "11ec71f4...5489b.png") and "Volume" (set to "100").

Sound

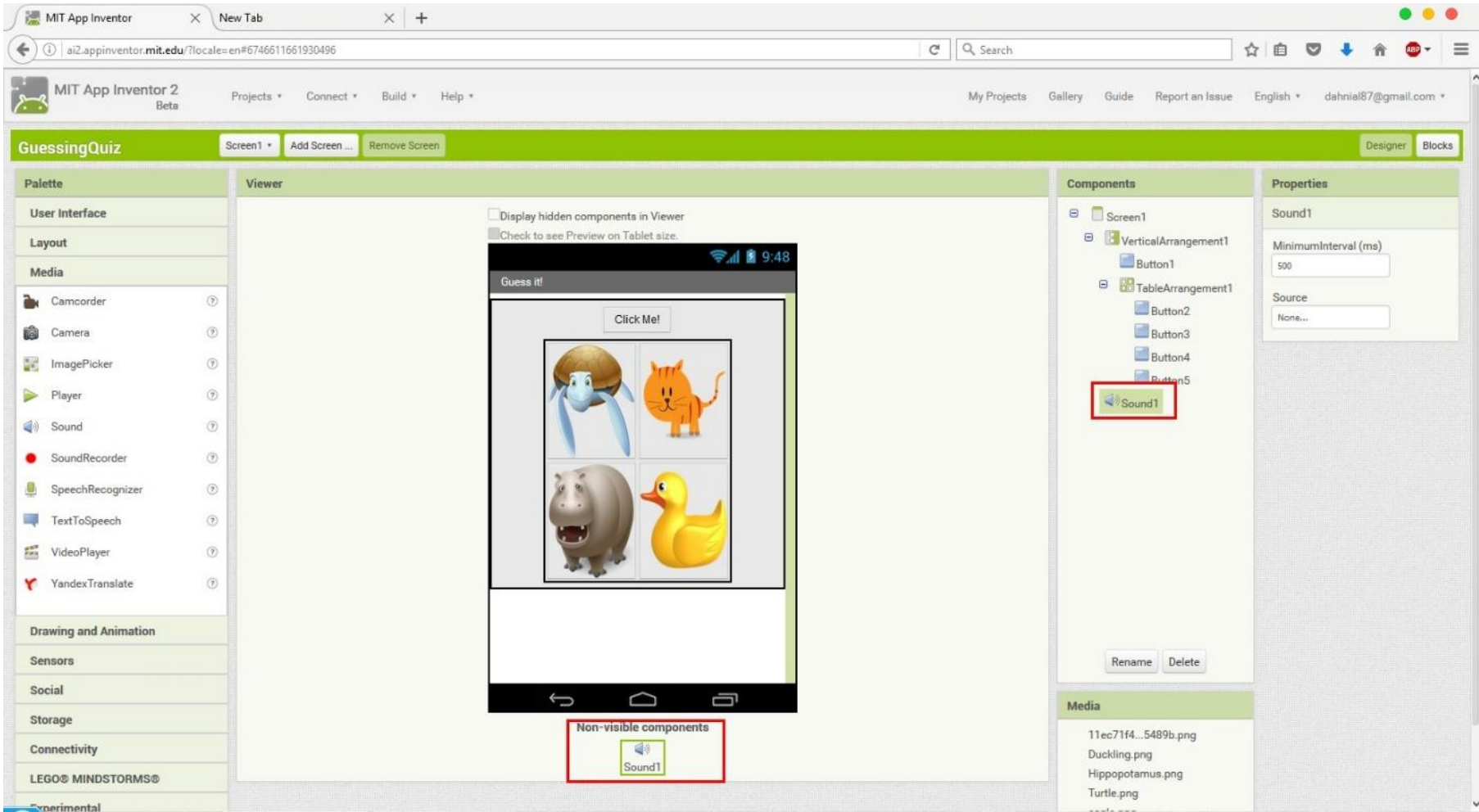
A multimedia component that plays sound files and optionally vibrates for the number of milliseconds (thousandths of a second) specified in the Blocks Editor. The name of the sound file to play can be specified either in the Designer or in the Blocks Editor.

For supported sound file formats, see [Android Supported Media Formats](#).

This Sound component is best for short sound files, such as sound effects, while the **Player** component is more efficient for longer sounds, such as songs.

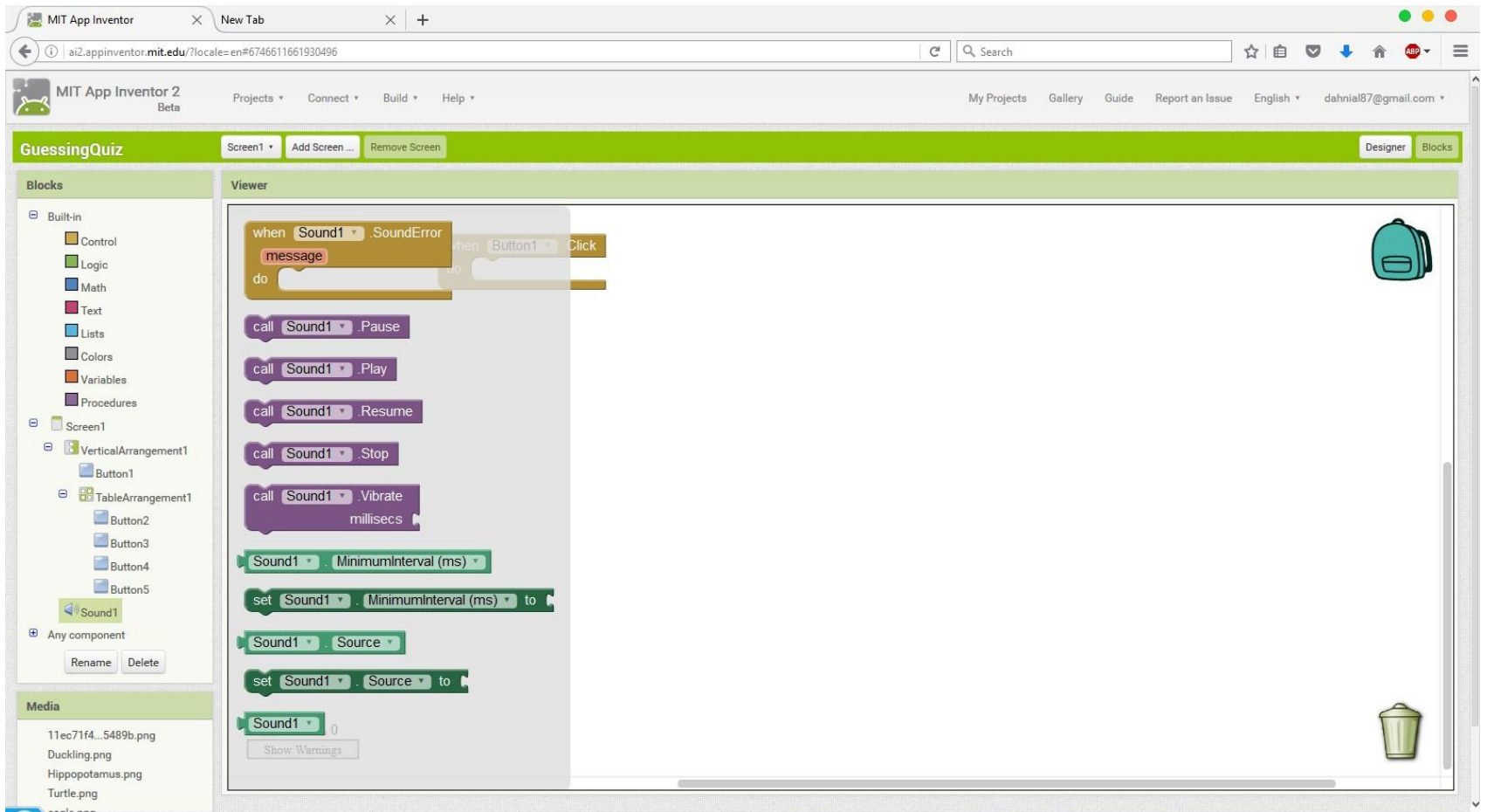
[More information](#)

Drag and drop to the screen



The screenshot displays the MIT App Inventor 2 web interface. The browser address bar shows the URL `ai2.appinventor.mit.edu/?locale=en#6746611661930496`. The application is titled "GuessingQuiz" and is in "Screen1" mode. The interface is divided into several panels:

- Palette:** Lists various components for the app, including User Interface, Layout, Media, Drawing and Animation, Sensors, Social, Storage, Connectivity, and LEGO MINDSTORMS.
- Viewer:** Shows a preview of the app on a mobile device. The app screen displays a "Guess it!" title, a "Click Me!" button, and a 2x2 grid of images: a blue crab, an orange cat, a brown hippopotamus, and a yellow duck. Below the viewer, a "Non-visible components" box contains a "Sound1" component.
- Components:** A tree view showing the hierarchy of components on the screen: Screen1, VerticalArrangement1, Button1, TableArrangement1, Button2, Button3, Button4, and Button5. A "Sound1" component is highlighted with a red box.
- Properties:** Shows the properties for the selected "Sound1" component, including "MinimumInterval (ms)" set to 500 and "Source" set to "None...".
- Media:** Lists media assets used in the app, including "11ec71f4...5489b.png", "Duckling.png", "Hippopotamus.png", and "Turtle.png".

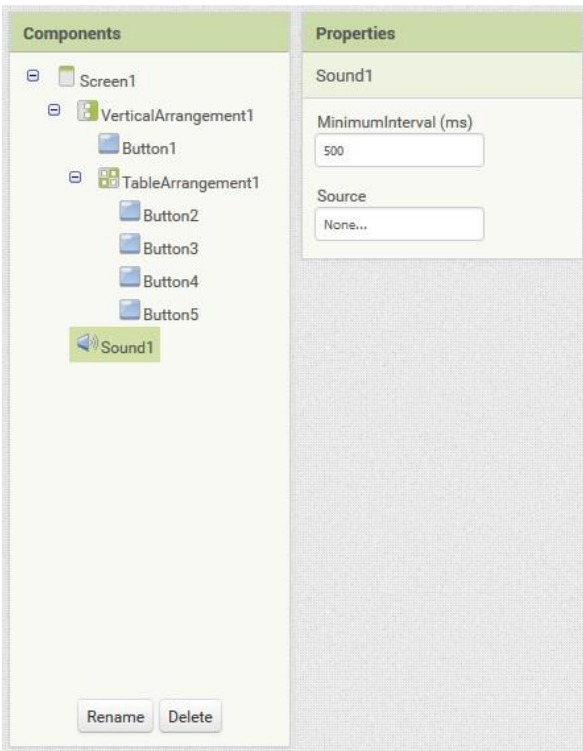


The screenshot shows the MIT App Inventor 2 Beta web interface. The browser address bar shows the URL: `ai2.appinventor.mit.edu/?locale=en#6746611661930496`. The page title is "MIT App Inventor 2 Beta". The main header includes "MIT App Inventor 2 Beta", navigation menus for "Projects", "Connect", "Build", and "Help", and user information for "dahnial87@gmail.com".

The interface is divided into several sections:

- Project Name:** "GuessingQuiz".
- Screen Management:** "Screen1", "Add Screen...", "Remove Screen".
- Designer:** A visual workspace for designing the app's look, currently showing a blank white area with a blue backpack icon and a trash can icon.
- Blocks:** A sidebar containing various code blocks categorized into:
 - Built-in: Control, Logic, Math, Text, Lists, Colors, Variables, Procedures.
 - Screen1: VerticalArrangement1, TableArrangement1 (containing Button1 through Button5).
 - Any component: Sound1.
- Viewer:** A code editor showing the following block structure:
 - when Sound1 . SoundError
 - do
 - message
 - do
 - when Button1 . Click
 - do
 - call Sound1 .Pause
 - call Sound1 .Play
 - call Sound1 .Resume
 - call Sound1 .Stop
 - call Sound1 .Vibrate milliseconds
 - Sound1 . MinimumInterval (ms)
 - set Sound1 . MinimumInterval (ms) to
 - Sound1 . Source
 - set Sound1 . Source to
 - Sound1 0

The sound component need to be linked with source sound. We will upload a wav file.



Components

- Screen1
 - VerticalArrangement1
 - Button1
 - TableArrangement1
 - Button2
 - Button3
 - Button4
 - Button5
 - Sound1

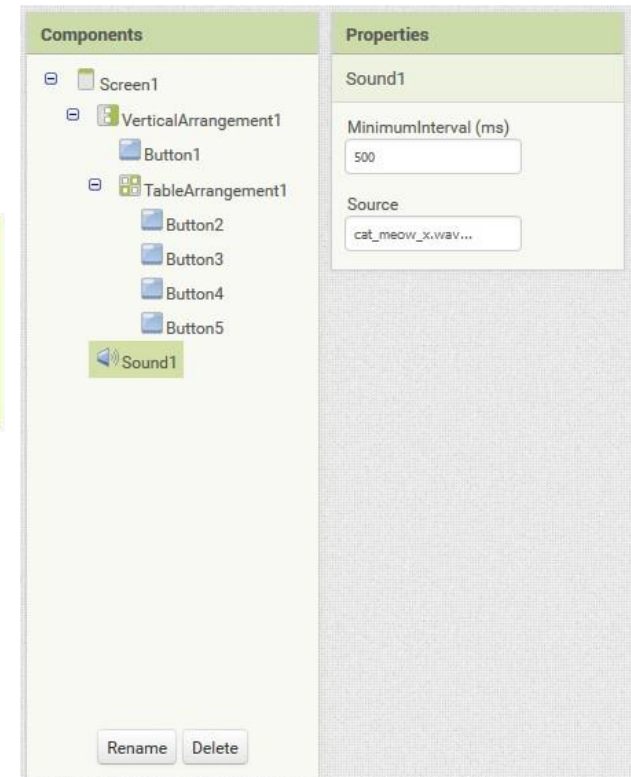
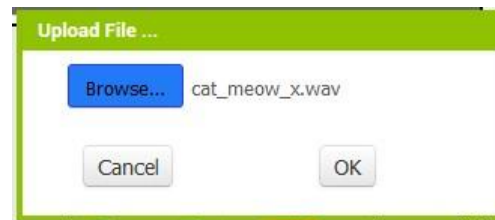
Properties

Sound1

MinimumInterval (ms)
500

Source
None...

Rename Delete



Components

- Screen1
 - VerticalArrangement1
 - Button1
 - TableArrangement1
 - Button2
 - Button3
 - Button4
 - Button5
 - Sound1

Properties

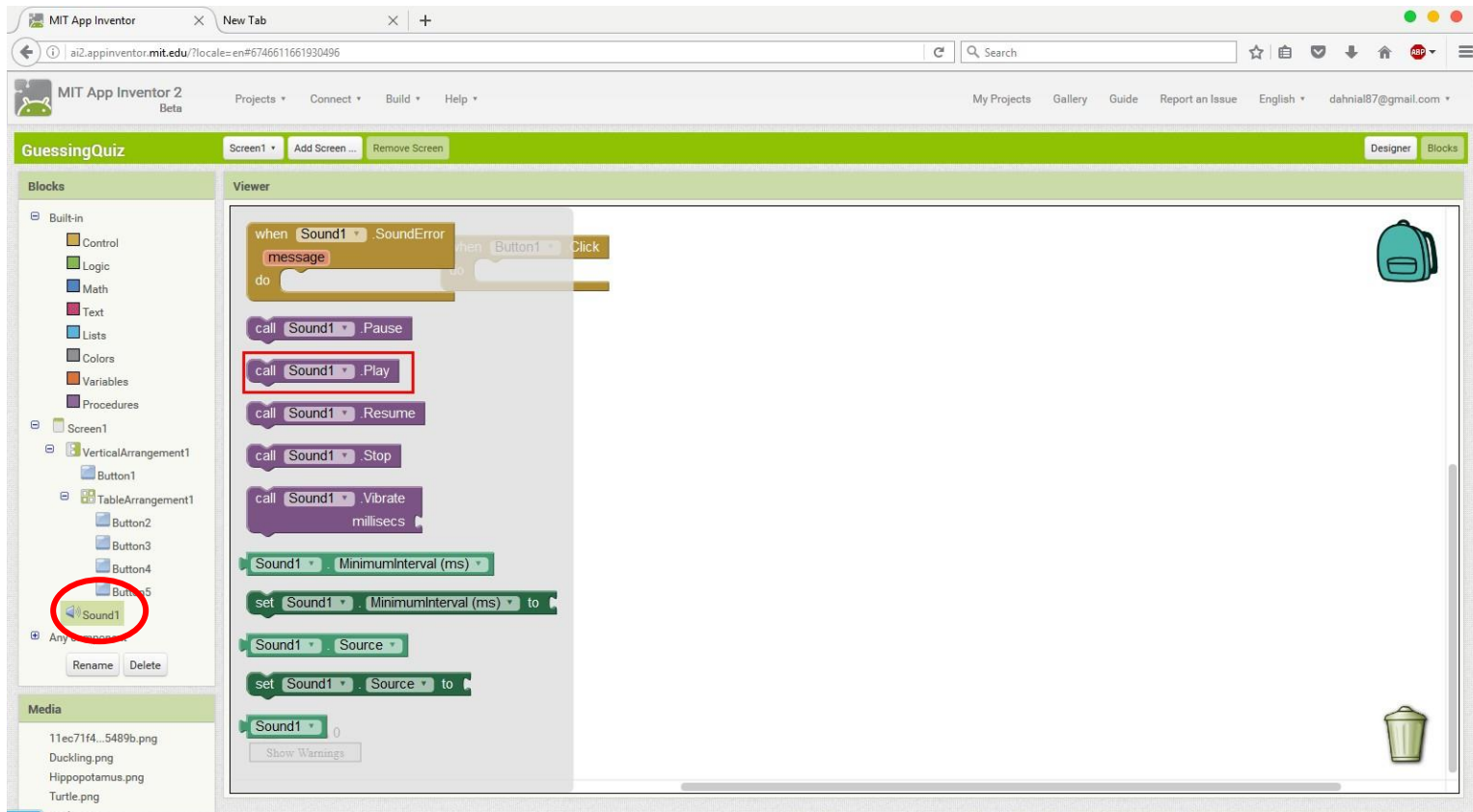
Sound1

MinimumInterval (ms)
500

Source
cat_meow_x.wav...

Rename Delete

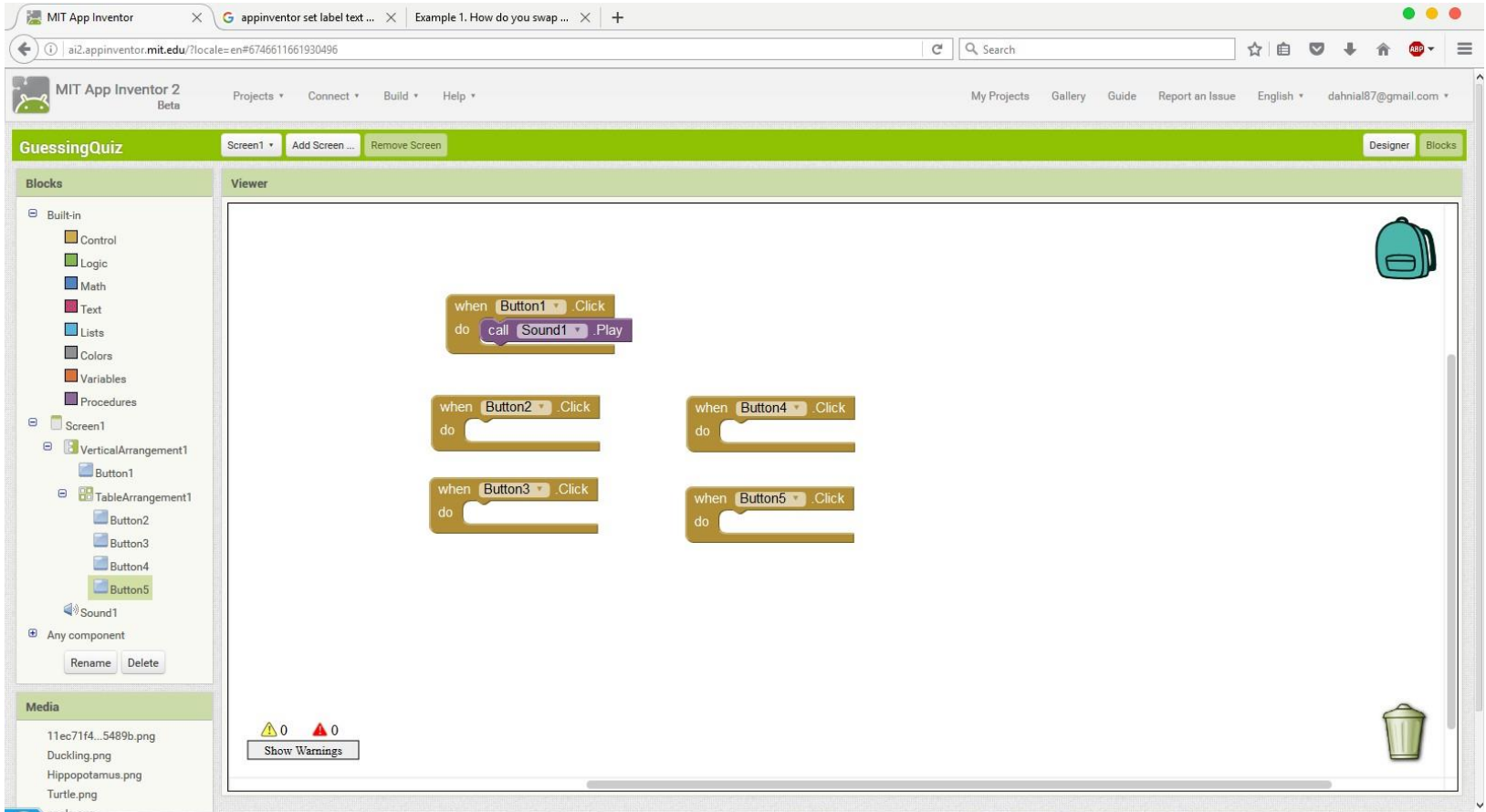
We give the function: **when button1 is clicked, play sound1** component



Result:

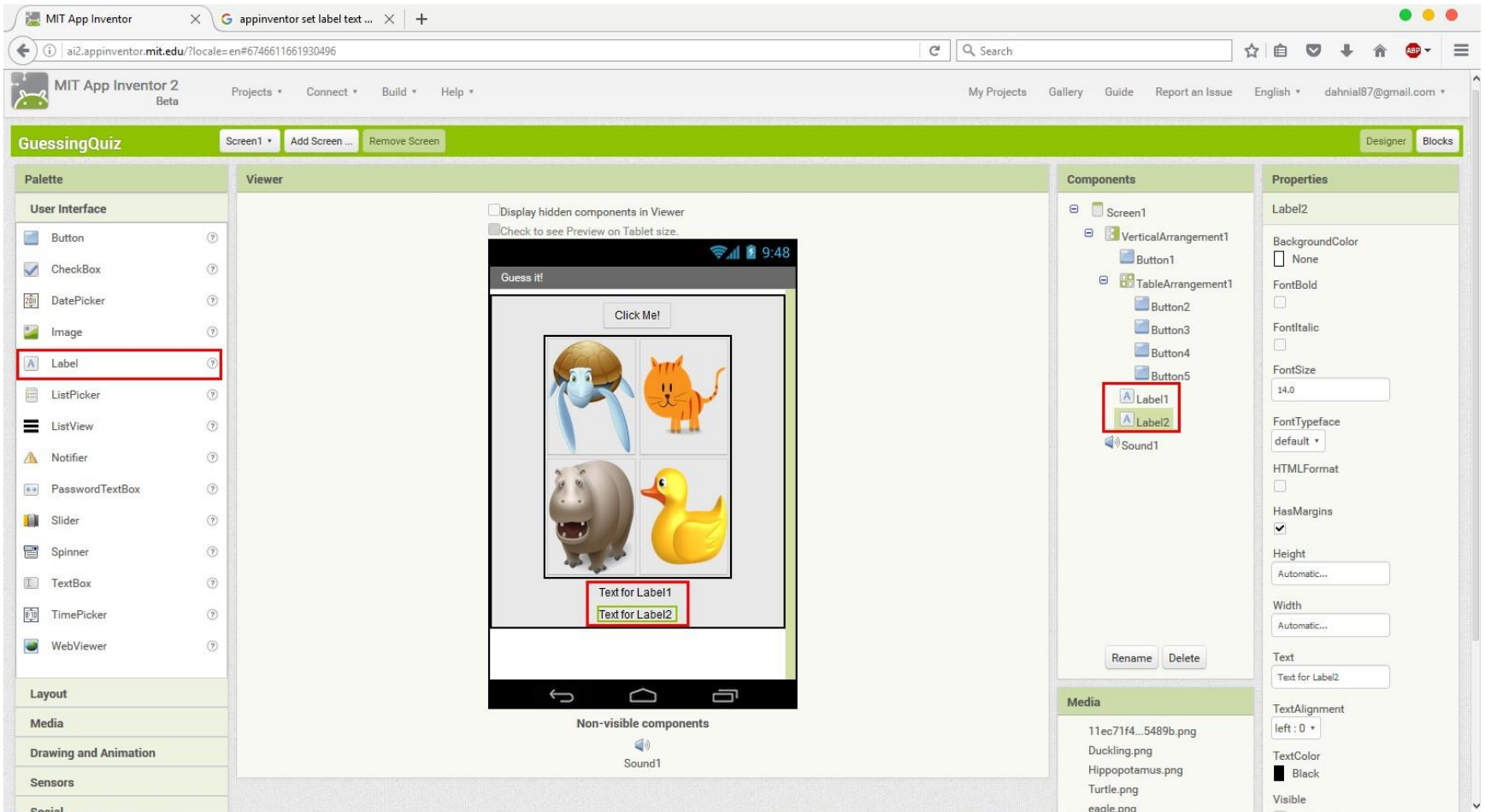


Add more “when button is clicked” blocks for each button



The screenshot shows the MIT App Inventor 2 web interface. The browser address bar displays `ai2.appinventor.mit.edu/?locale=en#6746611661930496`. The page title is "MIT App Inventor 2 Beta". The main workspace is titled "GuessingQuiz" and is in "Designer" mode. On the left, the "Blocks" palette is visible, showing categories like "Built-in", "Screen1", and "Media". The "Screen1" category is expanded, showing a "TableArrangement1" containing five buttons: "Button1", "Button2", "Button3", "Button4", and "Button5". The "Viewer" area on the right shows the visual representation of the app. It features a blue backpack icon in the top right corner and a green trash can icon in the bottom right corner. In the center, there are five "when button is clicked" blocks, each with a "do" block containing a "call Sound1 .Play" block. At the bottom left of the viewer, there is a "Show Warnings" button with a yellow warning icon and a red error icon, both with a count of 0.

Go to designer ui, add text label for the score



The screenshot shows the MIT App Inventor 2 Designer interface for a project named "GuessingQuiz". The interface is divided into several panels:

- Palette:** Lists various UI components. The "Label" component is highlighted with a red box.
- Viewer:** Displays a mobile app preview. It features a "Guess it!" screen with a "Click Me!" button and a 2x2 grid of animal images (a blue crab, an orange cat, a grey hippopotamus, and a yellow duck). Below the grid, two text labels are visible: "Text for Label1" and "Text for Label2", both highlighted with red boxes.
- Components:** Shows a tree view of the app's components. "Label1" and "Label2" are highlighted with red boxes.
- Properties:** Shows the properties for the selected "Label2" component. The "Text" property is set to "Text for Label2".

The browser address bar shows the URL: `ai2.appinventor.mit.edu/?locale=en#6746611661930496`. The MIT App Inventor 2 Beta logo and navigation menus are visible at the top of the application window.

MIT App Inventor 2 Beta

Projects Connect Build Help My Projects Gallery Guide Report an Issue English dahnia187@gmail.com

GuessingQuiz

Screen1 Add Screen ... Remove Screen Designer Blocks

Palette

User Interface

- Button
- CheckBox
- DatePicker
- Image
- Label
- ListPicker
- ListView
- Notifier
- PasswordTextBox
- Slider
- Spinner
- TextBox
- TimePicker
- WebView

Layout

Media


Drawing and Animation

Sensors

Social

Viewer

Display hidden components in Viewer
 Check to see Preview on Tablet size.



Non-visible components

- Sound1

Components

- Screen1
 - VerticalArrangement1
 - Button1
 - TableArrangement1
 - Button2
 - Button3
 - Button4
 - Button5
 - Label1
 - Label2
 - Sound1

Rename Delete

Media

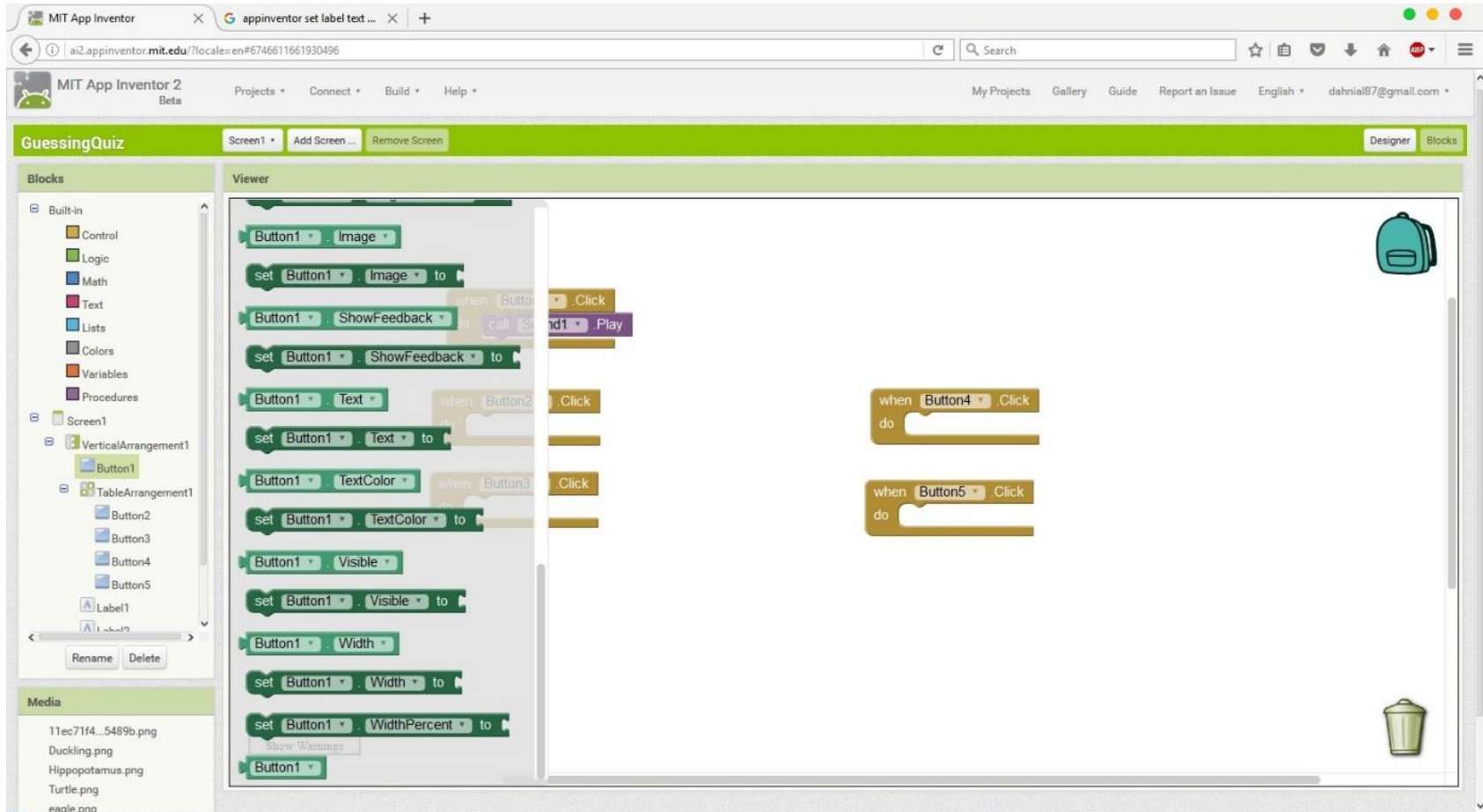
- 11ec71f4...5489b.png
- Duckling.png
- Hippopotamus.png
- Turtle.png
- eagle.png

Properties

Label2

- BackgroundColor: None
- FontBold:
- FontItalic:
- FontSize: 30
- FontTypeface: default
- HTMLFormat:
- HasMargins:
- Height: Automatic...
- Width: Automatic...
- Text: Text for Label2
- TextAlignment: left: 0
- TextColor: Black
- Visible:

We want the **player can only play sound once**, so we will hide the button

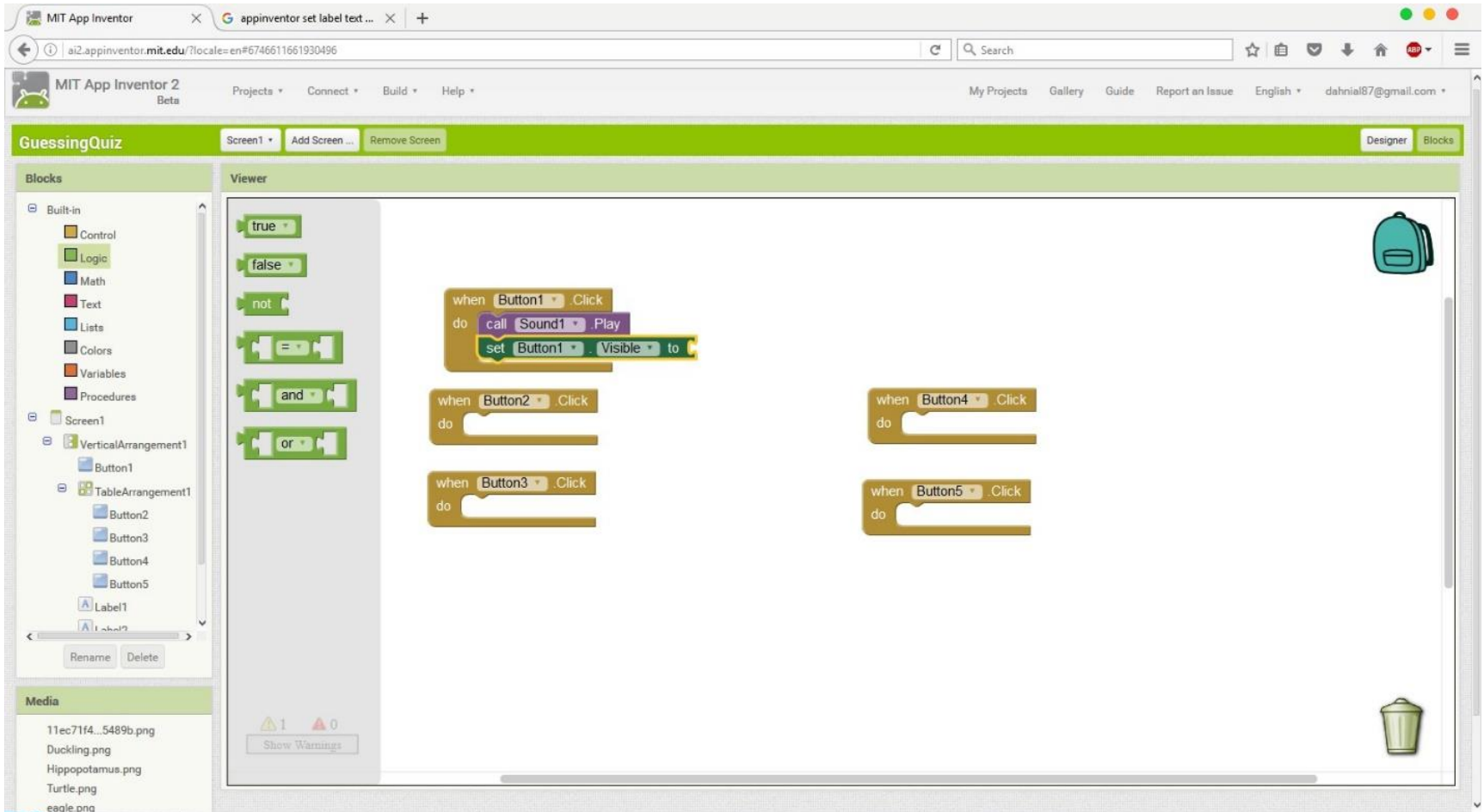


The screenshot shows the MIT App Inventor 2 interface for a project named "GuessingQuiz". The "Blocks" palette on the left is expanded to show the "Text" category. The "Viewer" area displays a script for Button1 with the following blocks:

- set Button1 Image to
- when Button1 Click to call Sound1 Play
- set Button1 ShowFeedback to
- when Button2 Click to set Button1 Text to
- when Button3 Click to set Button1 TextColor to
- Button1 Visible
- set Button1 Visible to
- Button1 Width
- set Button1 Width to
- set Button1 WidthPercent to
- Button1

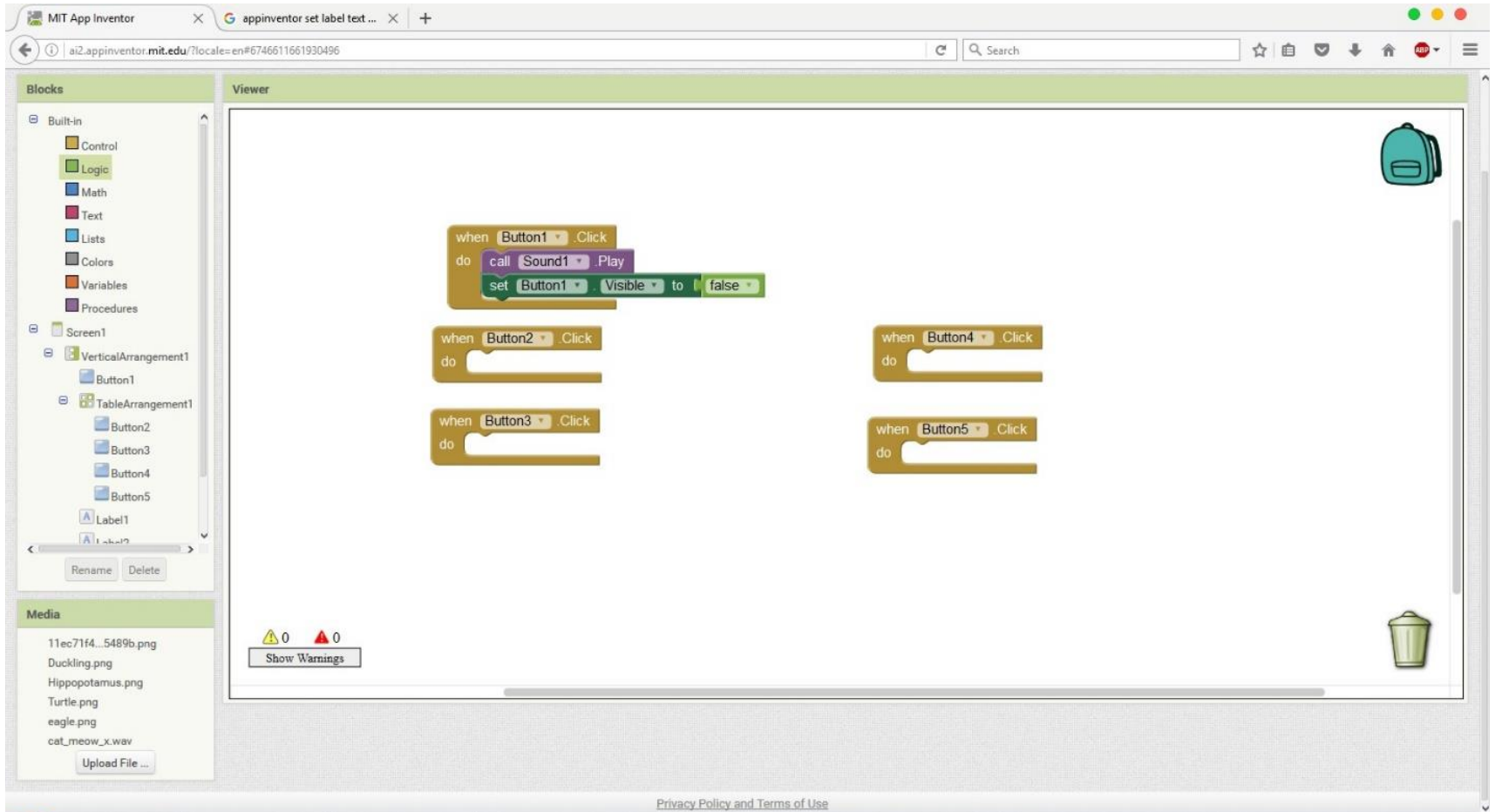
The "Media" section at the bottom left lists several image files: 11ec71f4...5489b.png, Duckling.png, Hippopotamus.png, Turtle.png, and eagle.png.

Result:



The screenshot displays the MIT App Inventor 2 Beta web interface. The browser address bar shows the URL `ai2.appinventor.mit.edu/?locale=en#6746611661930496`. The application title is "MIT App Inventor 2 Beta". The project name is "GuessingQuiz". The interface is divided into three main sections:

- Blocks:** A sidebar on the left containing categories like Control, Logic, Math, Text, Lists, Colors, Variables, and Procedures. Under "Screen1", there is a "VerticalArrangement1" containing "Button1" and a "TableArrangement1" containing "Button2", "Button3", "Button4", and "Button5". A "Label1" is also present.
- Viewer:** The central workspace showing the visual layout of the app. It features a blue backpack icon in the top right and a trash can icon in the bottom right. The logic blocks are as follows:
 - Button1 Click:** A "when Button1 Click" block with a "do" block containing "call Sound1 Play" and "set Button1 Visible to" (with a dropdown menu).
 - Button2 Click:** A "when Button2 Click" block with an empty "do" block.
 - Button3 Click:** A "when Button3 Click" block with an empty "do" block.
 - Button4 Click:** A "when Button4 Click" block with an empty "do" block.
 - Button5 Click:** A "when Button5 Click" block with an empty "do" block.
- Media:** A sidebar at the bottom left listing media files: "11ec71f4...5489b.png", "Duckling.png", "Hippopotamus.png", "Turtle.png", and "eagle.png".

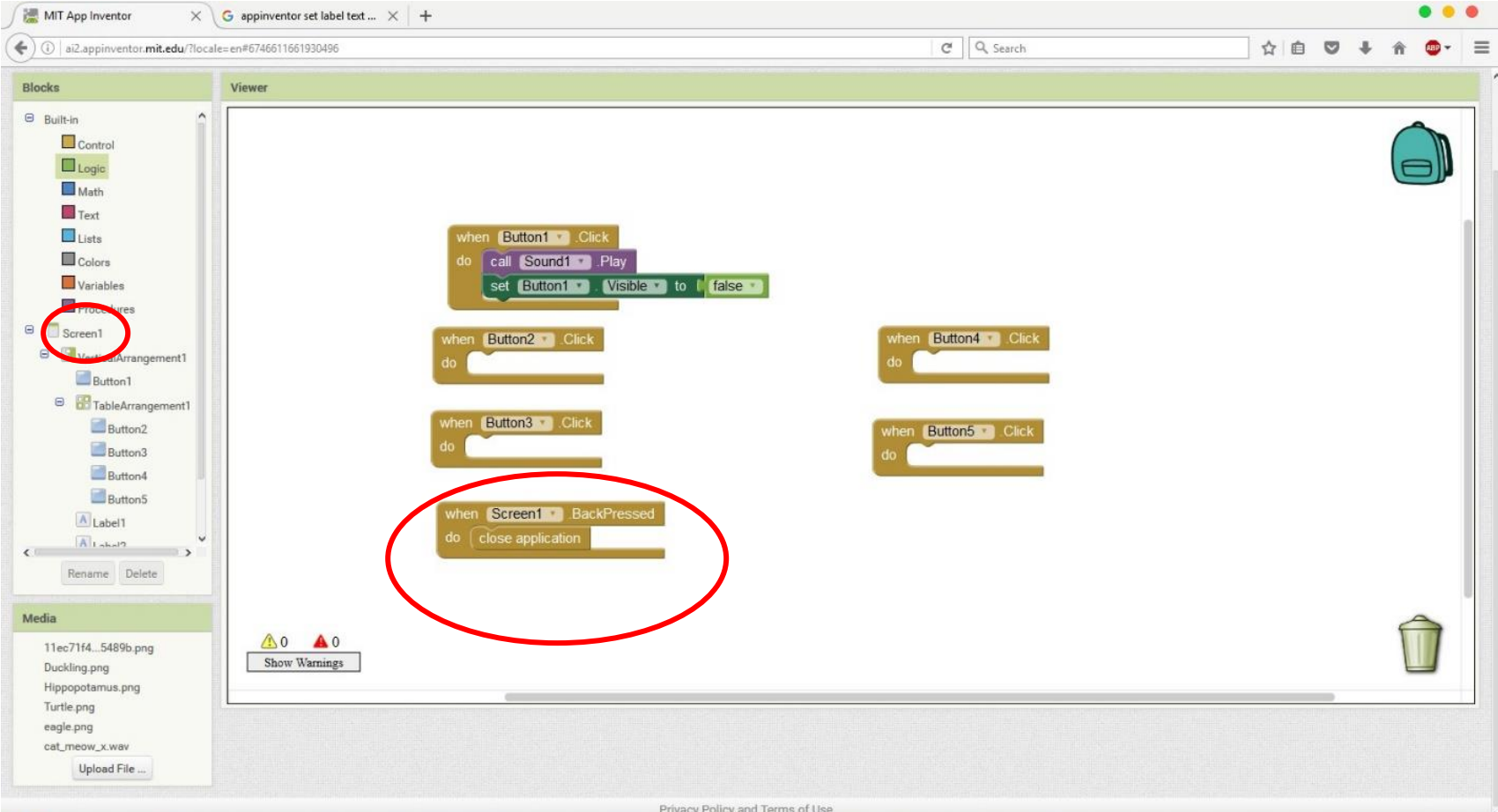


The screenshot shows the MIT App Inventor web interface. The browser address bar displays `ai2.appinventor.mit.edu/?locale=en#6746611661930496`. The interface is divided into three main sections: Blocks, Viewer, and Media.

- Blocks:** A sidebar on the left lists various block categories: Built-in (Control, Logic, Math, Text, Lists, Colors, Variables, Procedures), Screen1, VerticalArrangement1 (containing Button1), and TableArrangement1 (containing Button2, Button3, Button4, Button5, Label1, and Label2). Buttons for 'Rename' and 'Delete' are visible at the bottom of the list.
- Viewer:** The central workspace shows a visual programming canvas with several code blocks:
 - A 'when Button1 Click' block containing a 'call Sound1 Play' block and a 'set Button1 Visible to false' block.
 - Three empty 'when Button2 Click', 'when Button3 Click', and 'when Button4 Click' blocks.
 - One empty 'when Button5 Click' block.
- Media:** A sidebar at the bottom left lists media files: `11ec71f4...5489b.png`, `Duckling.png`, `Hippopotamus.png`, `Turtle.png`, `eagle.png`, and `cat_meow_x.wav`. An 'Upload File ...' button is located at the bottom.

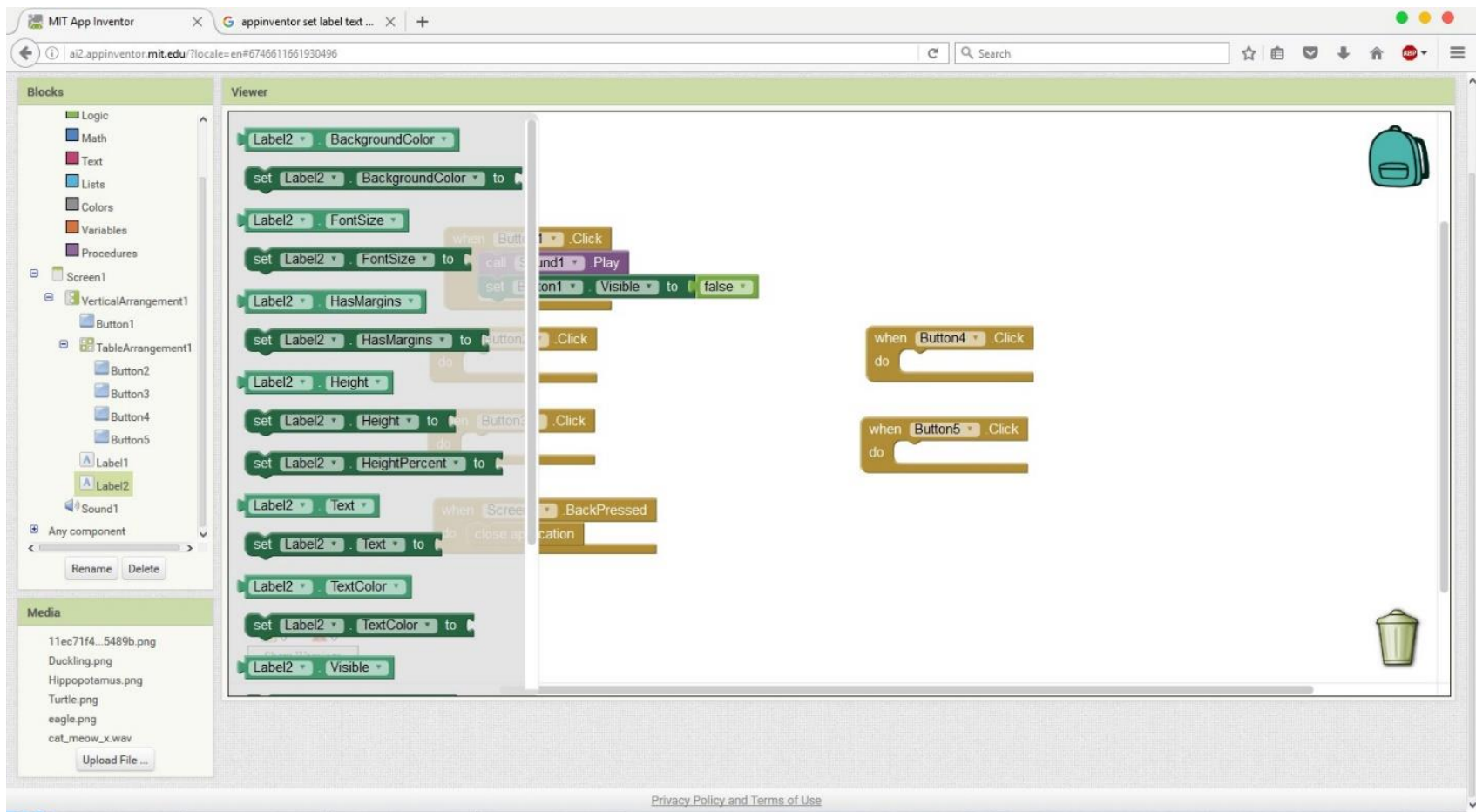
At the bottom of the Viewer, there is a 'Show Warnings' button with a warning icon and a '0' next to it. A trash can icon is located in the bottom right corner of the Viewer area. The footer of the page contains the text 'Privacy Policy and Terms of Use'.

add block so when the user presses back press in screen1, it will close the app



The screenshot shows the MIT App Inventor web interface. On the left, the 'Blocks' panel is visible, with 'Screen1' selected and circled in red. The 'Viewer' panel on the right displays a visual programming workspace with several blocks. A block for 'when Screen1 BackPressed' is circled in red, with a 'do' block containing 'close application'. Other blocks include 'when Button1 Click' (do: call Sound1 Play, set Button1 Visible to false), 'when Button2 Click', 'when Button3 Click', 'when Button4 Click', and 'when Button5 Click'. The interface also shows a 'Media' panel at the bottom left and a 'Show Warnings' button at the bottom center.

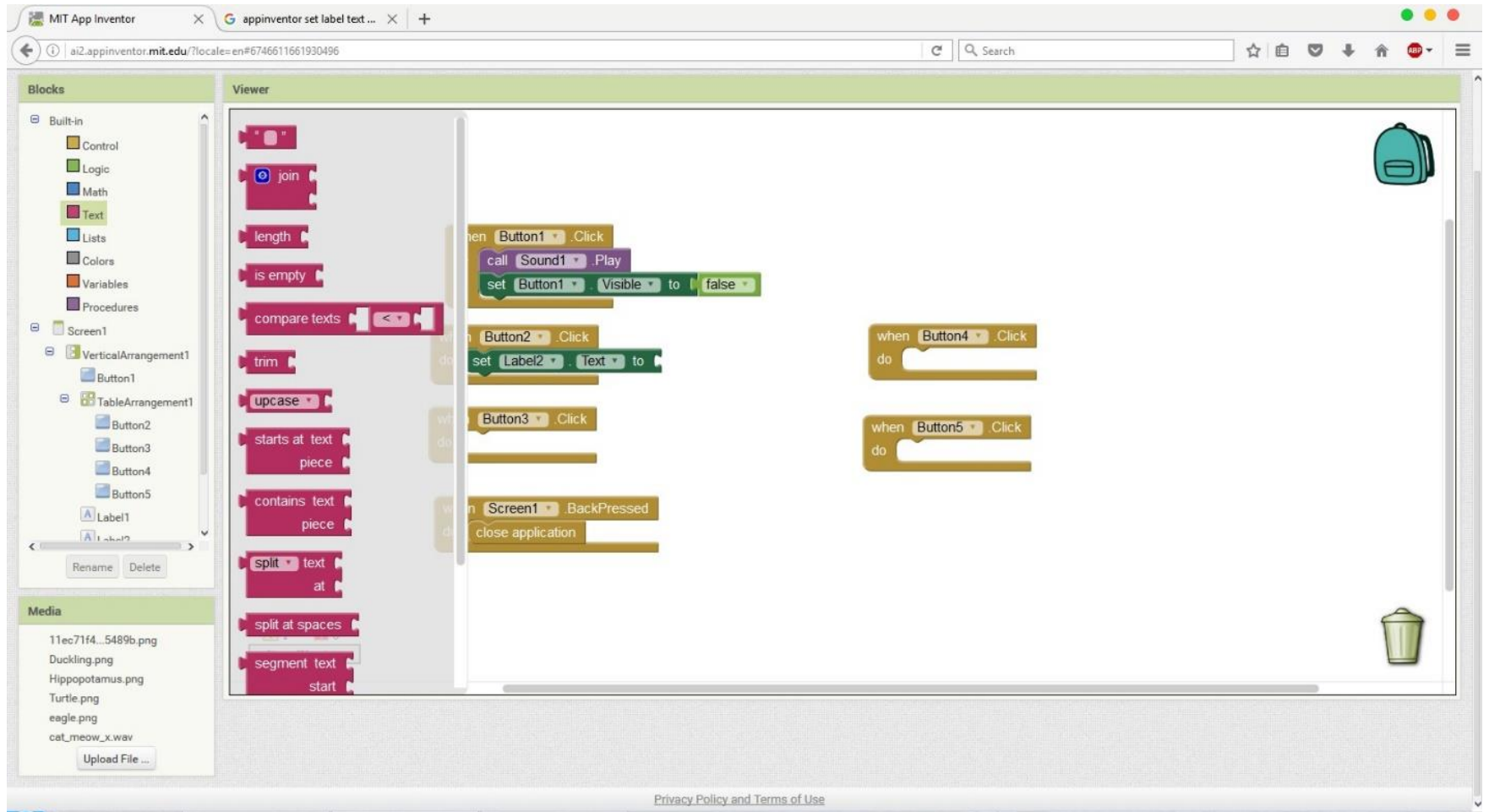
Add label for the score:



The screenshot shows the MIT App Inventor web interface. The left sidebar contains a 'Blocks' palette with categories like Logic, Math, Text, Lists, Colors, Variables, and Procedures. Under 'Text', 'Label2' is selected. The 'Viewer' pane shows a list of code blocks for 'Label2':

- Label2 BackgroundColor
- set Label2 BackgroundColor to
- Label2 FontSize
- set Label2 FontSize to
- Label2 HasMargins
- set Label2 HasMargins to
- Label2 Height
- set Label2 Height to
- set Label2 HeightPercent to
- Label2 Text
- set Label2 Text to
- Label2 TextColor
- set Label2 TextColor to
- Label2 Visible

The right pane shows a visual representation of the app with several buttons and a label. The bottom of the interface includes a 'Privacy Policy and Terms of Use' link.

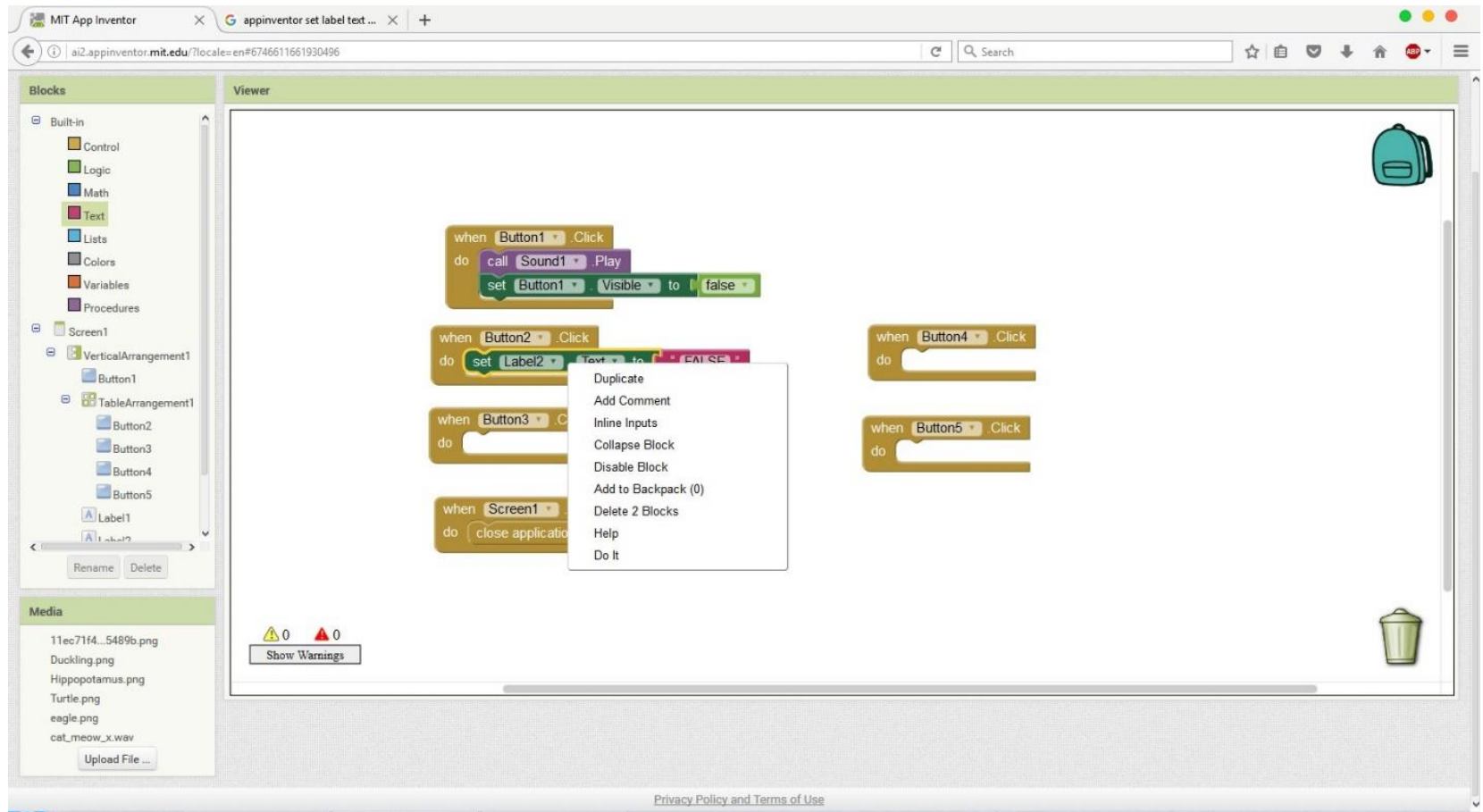


The image shows a screenshot of the MIT App Inventor web interface. The browser address bar shows the URL `ai2.appinventor.mit.edu/?locale=en#6746611661930496`. The interface is divided into several sections:

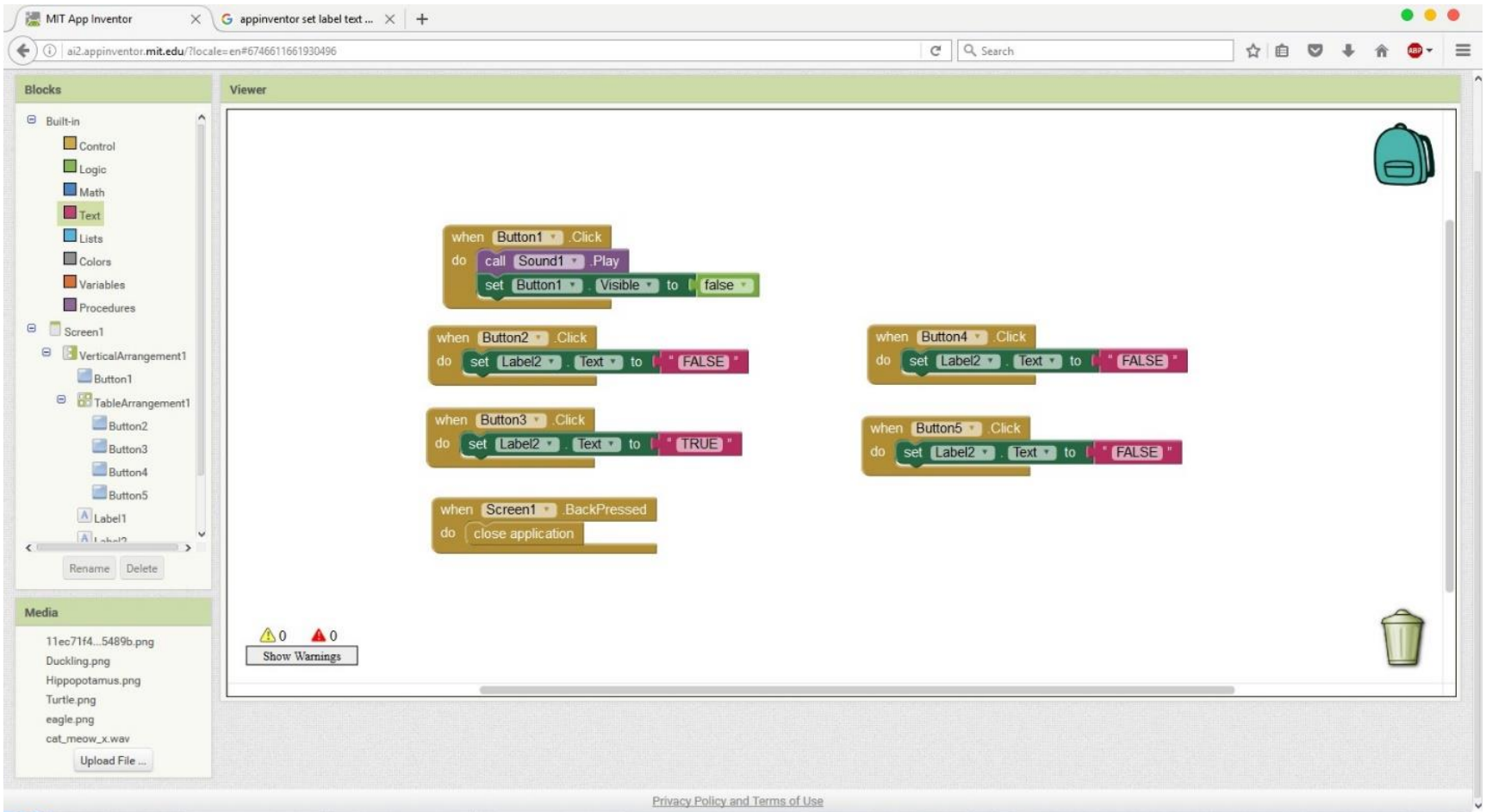
- Blocks:** A sidebar on the left containing categories like Built-in (Control, Logic, Math, Text, Lists, Colors, Variables, Procedures), Screen1, VerticalArrangement1, and TableArrangement1. The 'Text' category is currently selected.
- Media:** A section below the blocks sidebar listing various image and audio files, such as `11ec71f4...5489b.png`, `Duckling.png`, `Hippopotamus.png`, `Turtle.png`, `eagle.png`, and `cat_meow_x.wav`. There is an `Upload File ...` button.
- Viewer:** The main workspace on the right where visual programming blocks are assembled. It shows several event-driven blocks:
 - `when Button1 .Click` triggers `call Sound1 .Play` and `set Button1 .Visible to false`.
 - `when Button2 .Click` triggers `set Label2 .Text to`.
 - `when Button3 .Click` (empty).
 - `when Screen1 .BackPressed` triggers `close application`.
 - `when Button4 .Click` (empty).
 - `when Button5 .Click` (empty).

At the bottom of the interface, there is a `Privacy Policy and Terms of Use` link.

Duplicate, For faster process



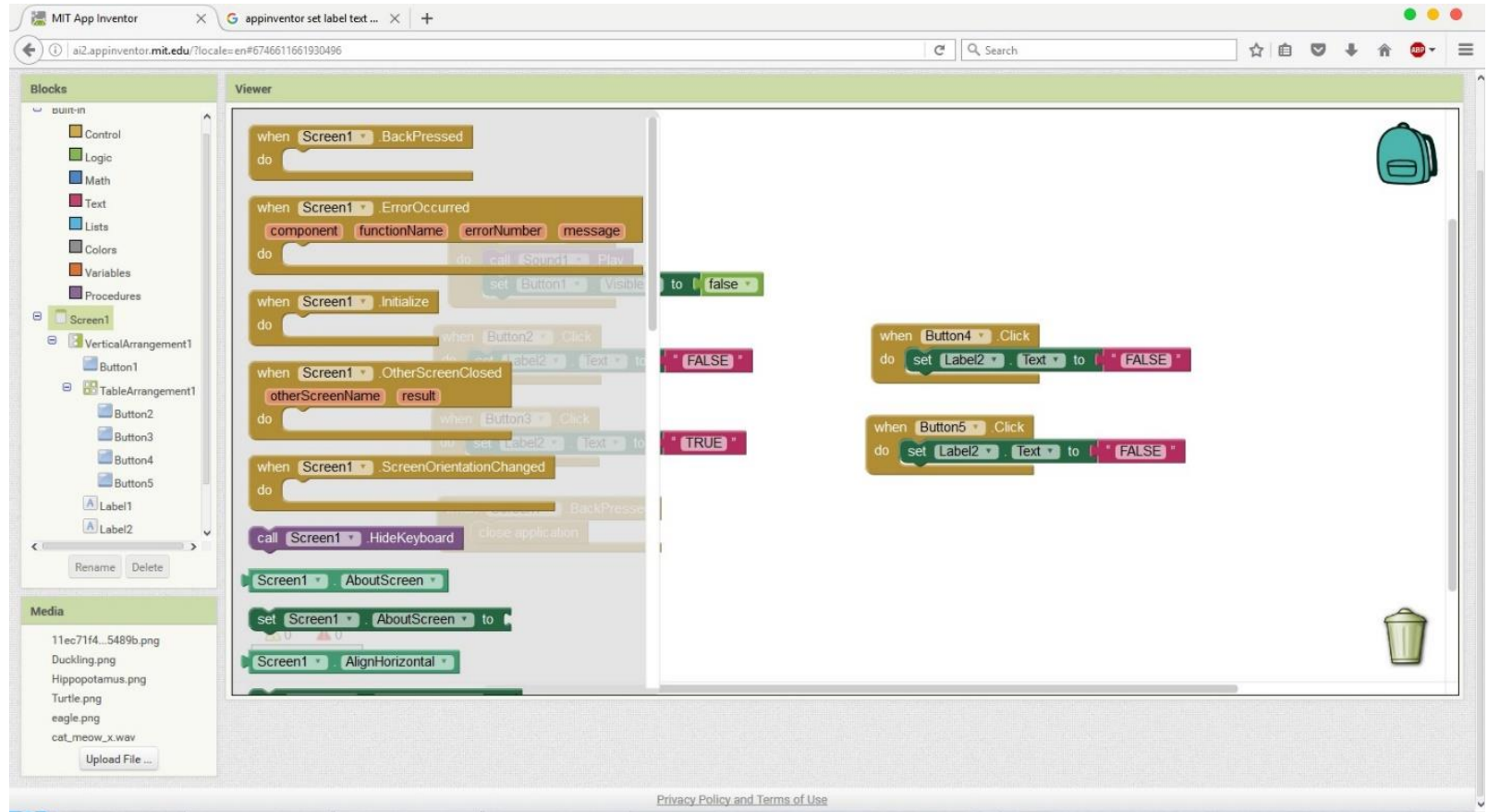
Result: TRUE only when button3 is clicked)

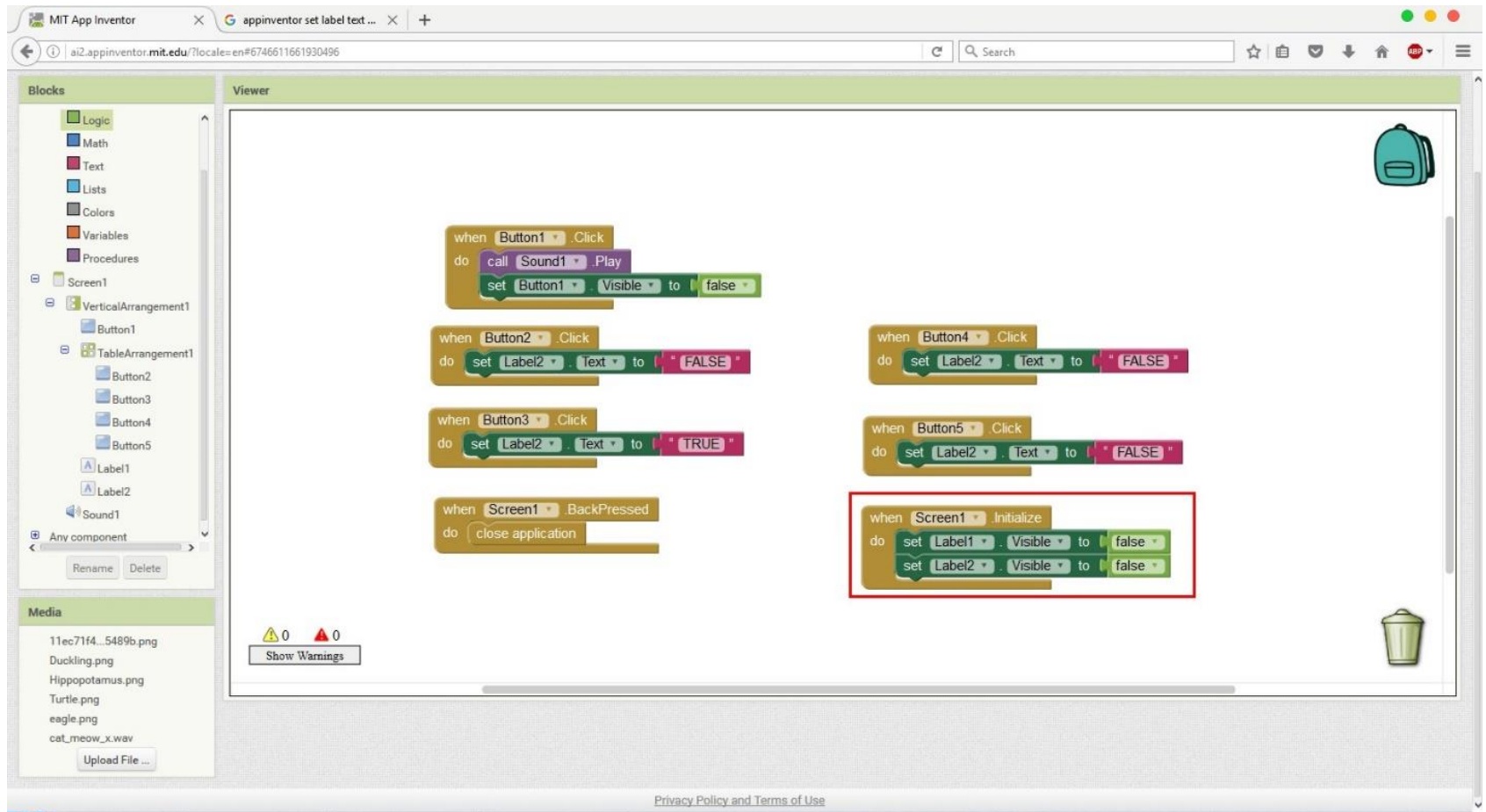


The screenshot shows the MIT App Inventor web interface. The browser address bar displays `ai2.appinventor.mit.edu/?locale=en#6746611661930496`. The interface is divided into three main sections: Blocks, Viewer, and Media.

- Blocks:** A sidebar on the left contains categories: Built-in (Control, Logic, Math, Text, Lists, Colors, Variables, Procedures), Screen1, VerticalArrangement1 (containing Button1), TableArrangement1 (containing Button2, Button3, Button4, Button5), and Label1. A 'Media' section at the bottom lists image and audio files with an 'Upload File...' button.
- Viewer:** The central workspace displays several code blocks:
 - when Button1 .Click**: do call Sound1 .Play, set Button1 .Visible to false.
 - when Button2 .Click**: do set Label2 .Text to "FALSE".
 - when Button3 .Click**: do set Label2 .Text to "TRUE".
 - when Button4 .Click**: do set Label2 .Text to "FALSE".
 - when Button5 .Click**: do set Label2 .Text to "FALSE".
 - when Screen1 .BackPressed**: do close application.
- Media:** A 'Show Warnings' button is visible at the bottom left of the viewer area, showing 0 warnings.

We need to **hide the score** when the **app initializes**





The screenshot shows the MIT App Inventor web interface. The browser address bar displays `ai2.appinventor.mit.edu/?locale=en#6746611661930496`. The interface is divided into three main sections: a left sidebar, a central workspace, and a bottom status bar.

Left Sidebar:

- Blocks:** A category menu with sub-categories: Logic, Math, Text, Lists, Colors, Variables, and Procedures. Under 'Screen1', there is a 'VerticalArrangement1' containing 'Button1' and a 'TableArrangement1' containing 'Button2', 'Button3', 'Button4', and 'Button5'. Below these are 'Label1', 'Label2', and 'Sound1'. At the bottom, there is an 'Any component' search bar with 'Rename' and 'Delete' buttons.
- Media:** A list of media files: '11ec71f4...5489b.png', 'Duckling.png', 'Hippopotamus.png', 'Turtle.png', 'eagle.png', and 'cat_meow_x.wav'. An 'Upload File ...' button is at the bottom.

Central Workspace (Viewer):

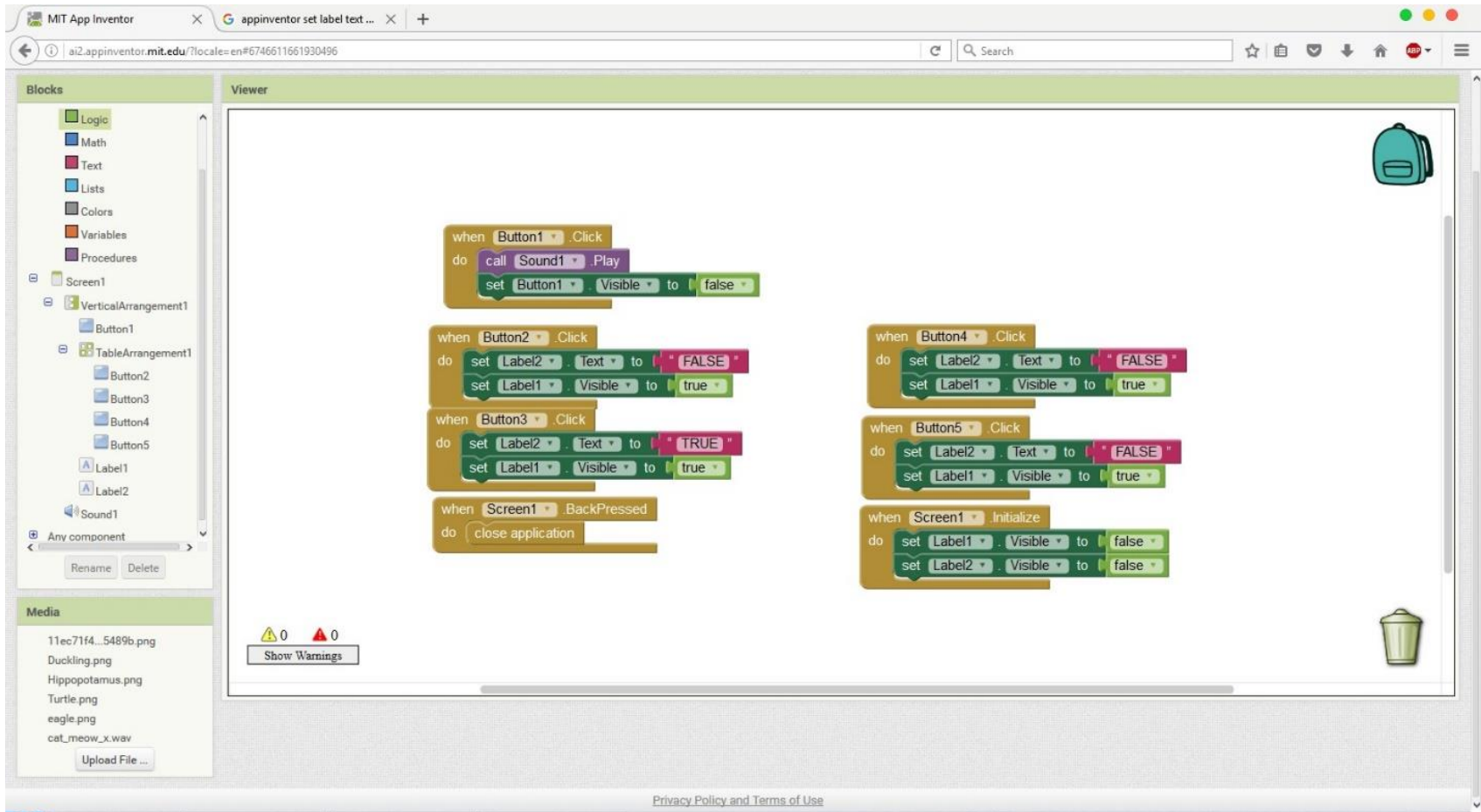
The workspace contains several event-driven code blocks:

- when Button1 . Click:** do call Sound1 . Play, set Button1 . Visible to false.
- when Button2 . Click:** do set Label2 . Text to "FALSE".
- when Button3 . Click:** do set Label2 . Text to "TRUE".
- when Button4 . Click:** do set Label2 . Text to "FALSE".
- when Button5 . Click:** do set Label2 . Text to "FALSE".
- when Screen1 . BackPressed:** do close application.
- when Screen1 . Initialize (highlighted with a red box):** do set Label1 . Visible to false, set Label2 . Visible to false.

Right Side: A blue backpack icon is in the top right, and a green trash can icon is in the bottom right.

Bottom Status Bar: Shows '0' warnings and '0' errors, with a 'Show Warnings' button. At the very bottom, there is a link for 'Privacy Policy and Terms of Use'.

But **we will show the result** when the button is pressed

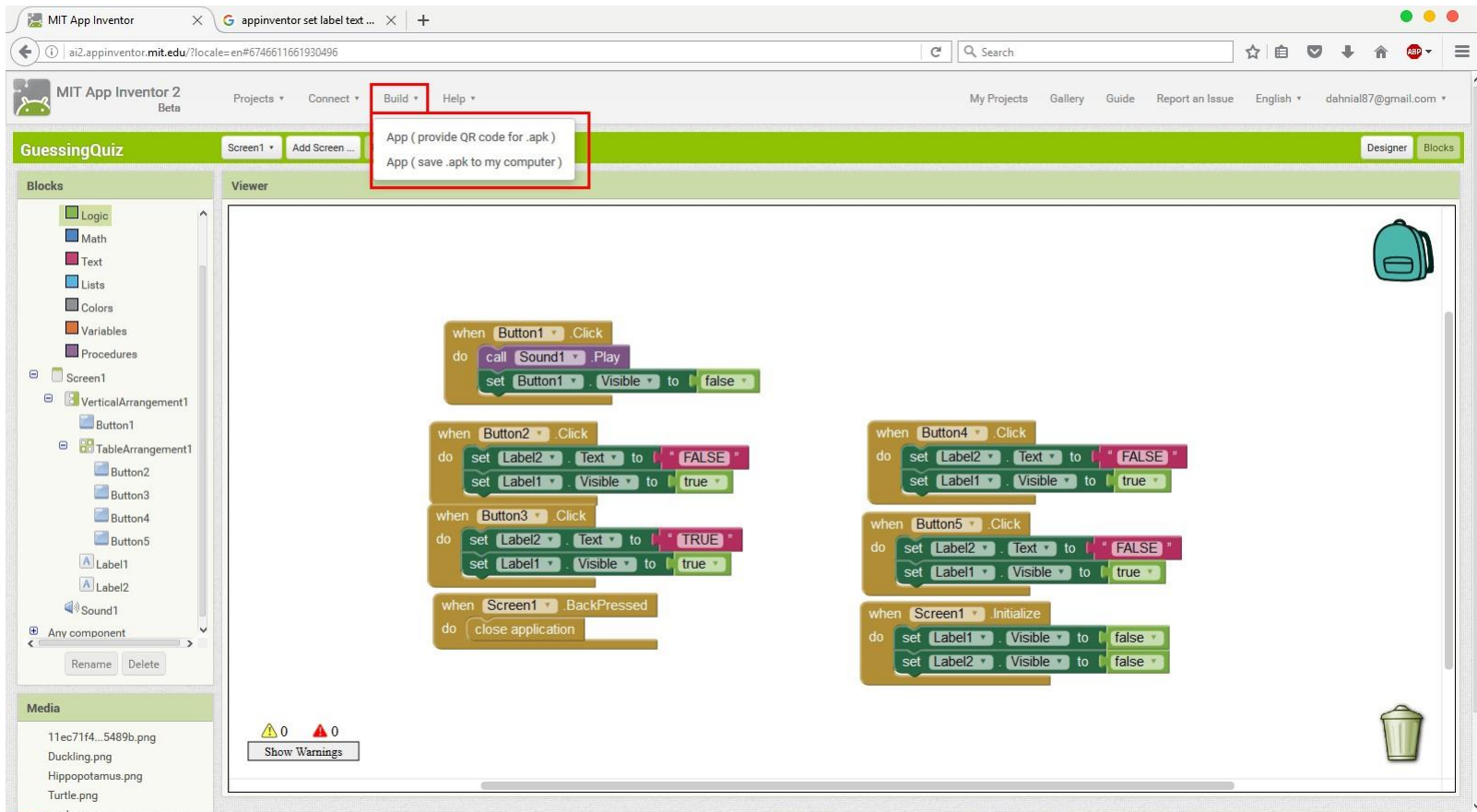


The screenshot shows the MIT App Inventor web interface. The left sidebar contains a 'Blocks' palette with categories like Logic, Math, Text, Lists, Colors, Variables, and Procedures. Under 'Screen1', there are 'VerticalArrangement1' and 'TableArrangement1' containers, each containing five buttons (Button1 to Button5). There are also 'Label1' and 'Label2' components. A 'Media' section at the bottom left lists image and audio files with an 'Upload File...' button. The main 'Viewer' area displays the following code blocks:

- when Button1 . Click**
 - do call Sound1 . Play
 - do set Button1 . Visible to false
- when Button2 . Click**
 - do set Label2 . Text to "FALSE"
 - do set Label1 . Visible to true
- when Button3 . Click**
 - do set Label2 . Text to "TRUE"
 - do set Label1 . Visible to true
- when Button4 . Click**
 - do set Label2 . Text to "FALSE"
 - do set Label1 . Visible to true
- when Button5 . Click**
 - do set Label2 . Text to "FALSE"
 - do set Label1 . Visible to true
- when Screen1 . BackPressed**
 - do close application
- when Screen1 . Initialize**
 - do set Label1 . Visible to false
 - do set Label2 . Visible to false

At the bottom of the viewer, there are warning icons (0) and a 'Show Warnings' button. A 'Privacy Policy and Terms of Use' link is visible at the very bottom.

Compiling the apps



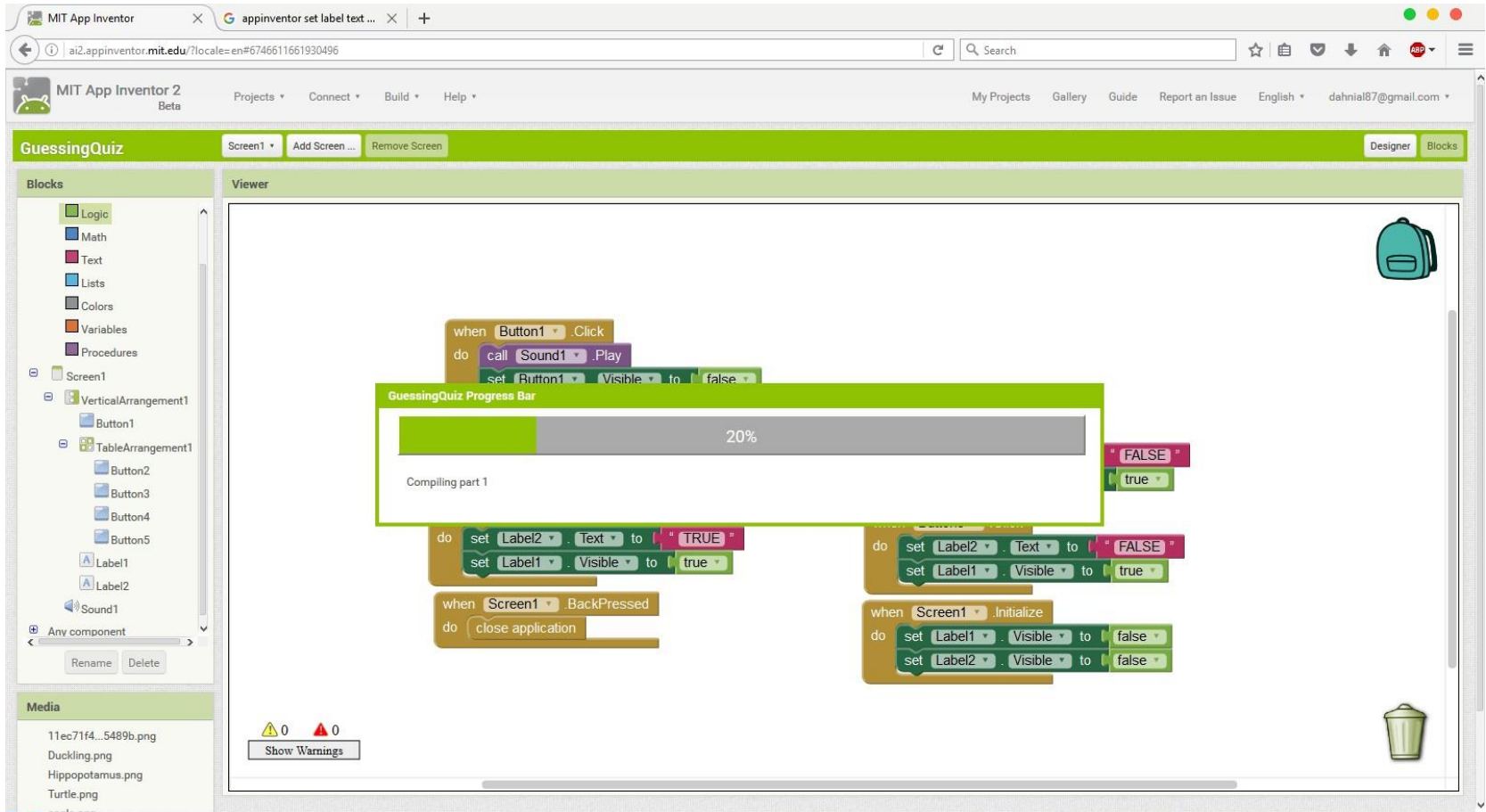
The screenshot shows the MIT App Inventor 2 web interface. The browser address bar displays `ai2.appinventor.mit.edu/?locale=en#6746611661930496`. The application title is "GuessingQuiz". The "Build" menu is open, showing two options: "App (provide QR code for .apk)" and "App (save .apk to my computer)".

The "Blocks" palette on the left lists categories: Logic, Math, Text, Lists, Colors, Variables, and Procedures. Under "Screen1", there are components: VerticalArrangement1, Button1, TableArrangement1, Button2, Button3, Button4, Button5, Label1, Label2, and Sound1. The "Media" section lists images: 11ec71f4..._5489b.png, Duckling.png, Hippopotamus.png, and Turtle.png.

The "Viewer" area displays the following code blocks:

- when Button1 Click: do call Sound1 .Play, set Button1 .Visible to false
- when Button2 Click: do set Label2 .Text to "FALSE", set Label1 .Visible to true
- when Button3 Click: do set Label2 .Text to "TRUE", set Label1 .Visible to true
- when Screen1 BackPressed: do close application
- when Button4 Click: do set Label2 .Text to "FALSE", set Label1 .Visible to true
- when Button5 Click: do set Label2 .Text to "FALSE", set Label1 .Visible to true
- when Screen1 Initialize: do set Label1 .Visible to false, set Label2 .Visible to false

At the bottom left, there are warning icons (0) and a "Show Warnings" button. A trash icon is visible in the bottom right corner of the viewer area.



The screenshot shows the MIT App Inventor 2 Beta web interface. The browser address bar displays `ai2.appinventor.mit.edu/?locale=en#6746611661930496`. The application title is "GuessingQuiz". The interface is divided into three main sections: "Blocks", "Viewer", and "Media".

Blocks Panel: Lists various code blocks categorized by Logic, Math, Text, Lists, Colors, Variables, and Procedures. A "Screen1" folder contains a "VerticalArrangement1" with "Button1", and a "TableArrangement1" with "Button2", "Button3", "Button4", and "Button5". There are also "Label1", "Label2", and "Sound1" components.

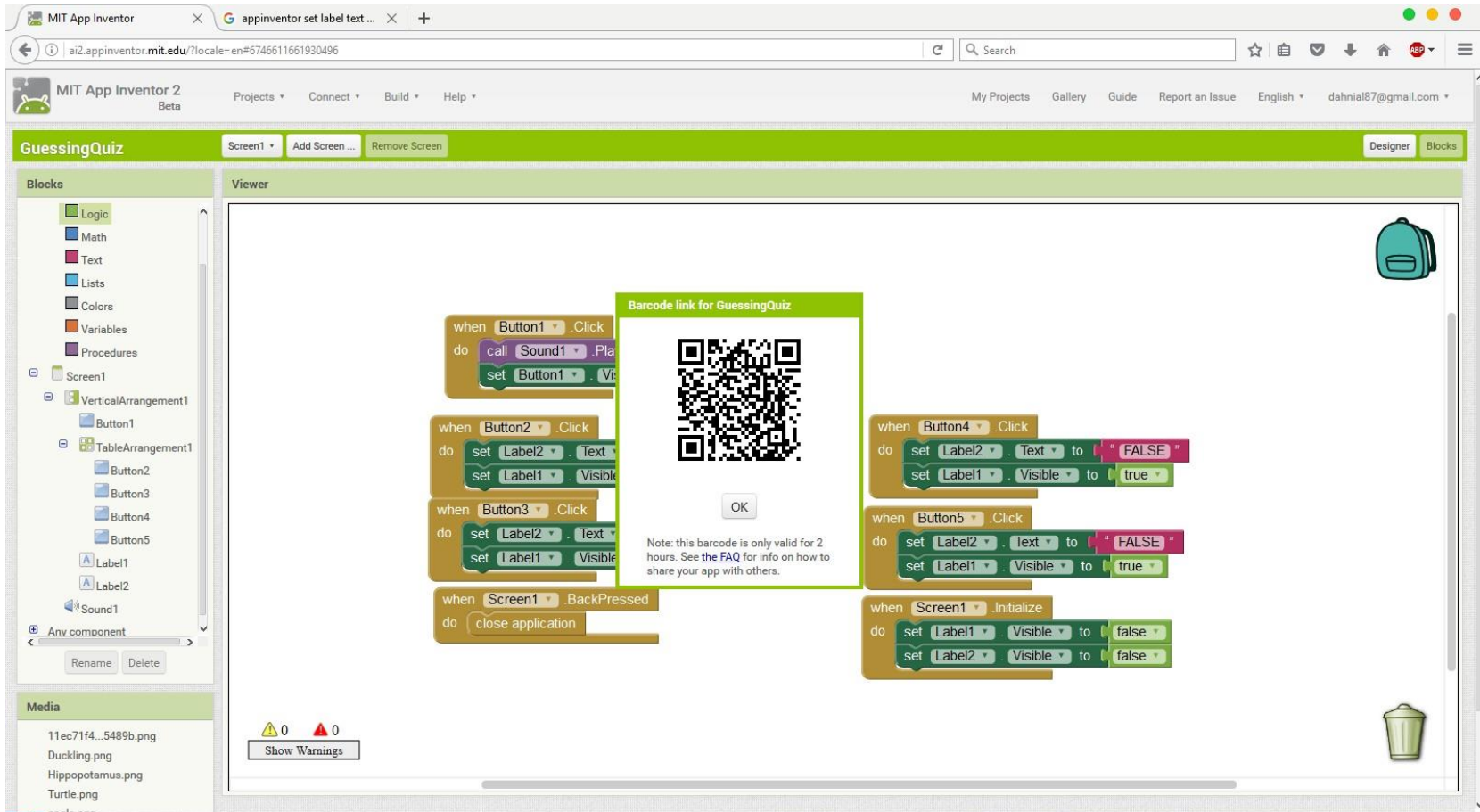
Viewer Panel: Displays the visual representation of the app. It features a "GuessingQuiz Progress Bar" with a green segment and the text "20%". Below the bar, it says "Compiling part 1". A "FALSE" label is visible on the right. The background includes a blue backpack icon in the top right and a trash can icon in the bottom right.

Code Blocks:

- when Button1 .Click**
 - do call Sound1 .Play
 - set Button1 .Visible to false
- when Screen1 .BackPressed**
 - do close application
- when Screen1 .Initialize**
 - do set Label1 .Visible to false
 - do set Label2 .Visible to false
- do set Label2 .Text to "TRUE"**
- do set Label1 .Visible to true**
- do set Label2 .Text to "FALSE"**
- do set Label1 .Visible to true**

Media Panel: Lists image assets: "11ec71f4...5489b.png", "Duckling.png", "Hippopotamus.png", and "Turtle.png".

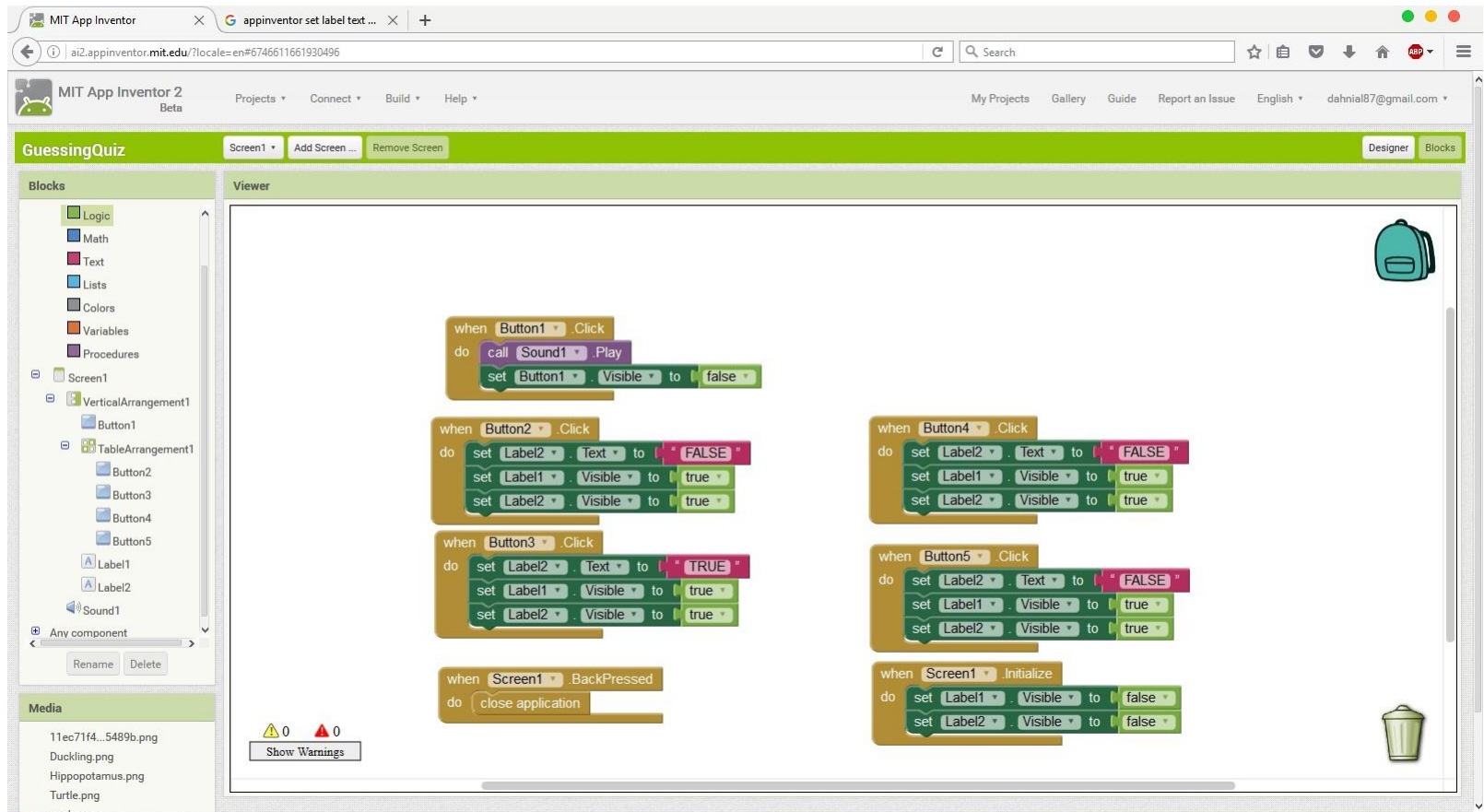
Using qr code (download to device using browser and manual installation)



The screenshot displays the MIT App Inventor 2 web interface. The browser address bar shows the URL: `ai2.appinventor.mit.edu/?locale=en#6746611661930496`. The application title is "GuessingQuiz". The interface is divided into several sections:

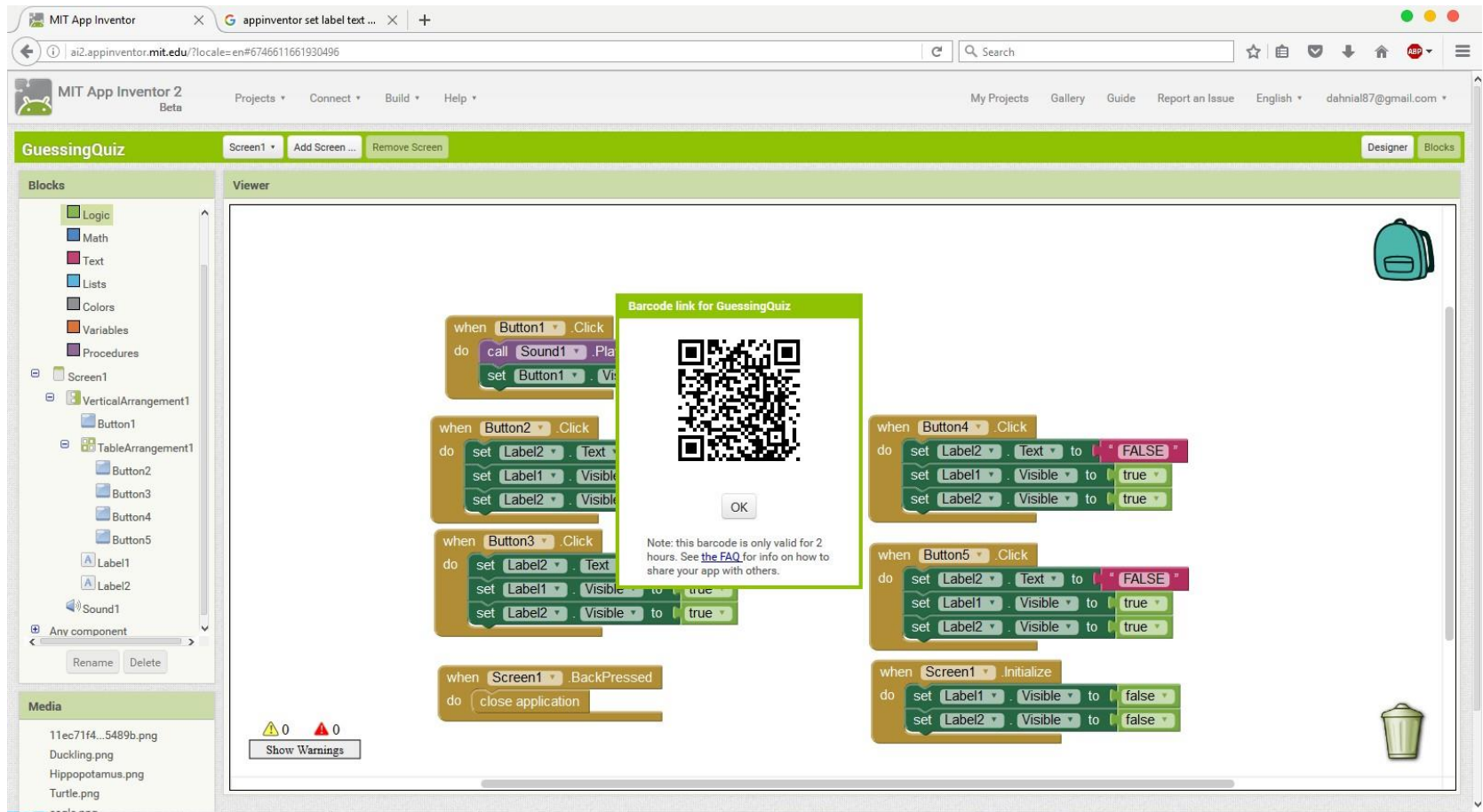
- Blocks:** A sidebar on the left lists various code blocks categorized by Logic, Math, Text, Lists, Colors, Variables, and Procedures. It also shows a tree view of the app's components, including Screen1, VerticalArrangement1, and TableArrangement1.
- Viewer:** The central workspace shows a visual representation of the app screen. A central dialog box titled "Barcode link for GuessingQuiz" is displayed, containing a QR code and an "OK" button. Below the QR code, a note states: "Note: this barcode is only valid for 2 hours. See the FAQ for info on how to share your app with others." The background of the viewer shows a grid with several buttons (Button1 to Button5) and labels (Label1, Label2) with associated code blocks.
- Code Blocks:** The blocks are organized as follows:
 - when Button1 Click:** call Sound1.Play, set Button1.Visible to true.
 - when Button2 Click:** set Label2.Text to "FALSE", set Label1.Visible to true.
 - when Button3 Click:** set Label2.Text to "FALSE", set Label1.Visible to true.
 - when Button4 Click:** set Label2.Text to "FALSE", set Label1.Visible to true.
 - when Button5 Click:** set Label2.Text to "FALSE", set Label1.Visible to true.
 - when Screen1 BackPressed:** close application.
 - when Screen1 Initialize:** set Label1.Visible to false, set Label2.Visible to false.
- Media:** A sidebar at the bottom left lists media assets: 11ec71f4...5489b.png, Duckling.png, Hippopotamus.png, and Turtle.png.
- Warnings:** A "Show Warnings" button is located at the bottom left of the viewer area, showing 0 warnings.

Oops, forget to show label2 when the button is clicked



The screenshot shows the MIT App Inventor 2 web interface for a project named "GuessingQuiz". The interface is divided into several sections:

- Blocks:** A sidebar on the left containing categories like Logic, Math, Text, Lists, Colors, Variables, and Procedures. Under "Screen1", there are components for VerticalArrangement1, TableArrangement1 (containing Button1-5), Label1, Label2, and Sound1.
- Viewer:** The main workspace showing a visual representation of the app screen with a blue backpack icon in the top right and a trash can icon in the bottom right.
- Code Blocks:** Several event-driven code blocks are visible:
 - when Button1 . Click:** do call Sound1 . Play, set Button1 . Visible to false.
 - when Button2 . Click:** do set Label2 . Text to "FALSE", set Label1 . Visible to true, set Label2 . Visible to true.
 - when Button3 . Click:** do set Label2 . Text to "TRUE", set Label1 . Visible to true, set Label2 . Visible to true.
 - when Button4 . Click:** do set Label2 . Text to "FALSE", set Label1 . Visible to true, set Label2 . Visible to true.
 - when Button5 . Click:** do set Label2 . Text to "FALSE", set Label1 . Visible to true, set Label2 . Visible to true.
 - when Screen1 . BackPressed:** do close application.
 - when Screen1 . Initialize:** do set Label1 . Visible to false, set Label2 . Visible to false.
- Media:** A section at the bottom left showing a list of image files: 11ec71f4...5489b.png, Duckling.png, Hippopotamus.png, and Turtle.png.
- Warnings:** A "Show Warnings" button is located at the bottom left of the viewer area, with a warning icon and a red triangle icon next to it.



The screenshot displays the MIT App Inventor 2 Beta web interface. The browser address bar shows the URL `ai2.appinventor.mit.edu/?locale=en#6746611661930496`. The application title is "GuessingQuiz".

Left Panel (Blocks):

- Logic: Math, Text, Lists, Colors, Variables, Procedures
- Screen1
 - VerticalArrangement1
 - Button1
 - TableArrangement1
 - Button2
 - Button3
 - Button4
 - Button5
 - Label1
 - Label2
 - Sound1
- Any component

Right Panel (Code Blocks):

- when Button1 Click**
 - do call Sound1 . Play
 - set Button1 . Visible to true
- when Button2 Click**
 - do set Label2 . Text to []
 - set Label1 . Visible to true
 - set Label2 . Visible to true
- when Button3 Click**
 - do set Label2 . Text to []
 - set Label1 . Visible to true
 - set Label2 . Visible to true
- when Button4 Click**
 - do set Label2 . Text to FALSE
 - set Label1 . Visible to true
 - set Label2 . Visible to true
- when Button5 Click**
 - do set Label2 . Text to FALSE
 - set Label1 . Visible to true
 - set Label2 . Visible to true
- when Screen1 BackPressed**
 - do close application
- when Screen1 Initialize**
 - do set Label1 . Visible to false
 - set Label2 . Visible to false

Center View:

A dialog box titled "Barcode link for GuessingQuiz" is displayed, containing a QR code and an "OK" button. Below the QR code, a note reads: "Note: this barcode is only valid for 2 hours. See the [FAQ](#) for info on how to share your app with others."

Bottom Panel:

- Media: 11ec71f4...5489b.png, Duckling.png, Hippopotamus.png, Turtle.png
- Warnings: 0 (yellow triangle), 0 (red triangle), Show Warnings button

RESULT



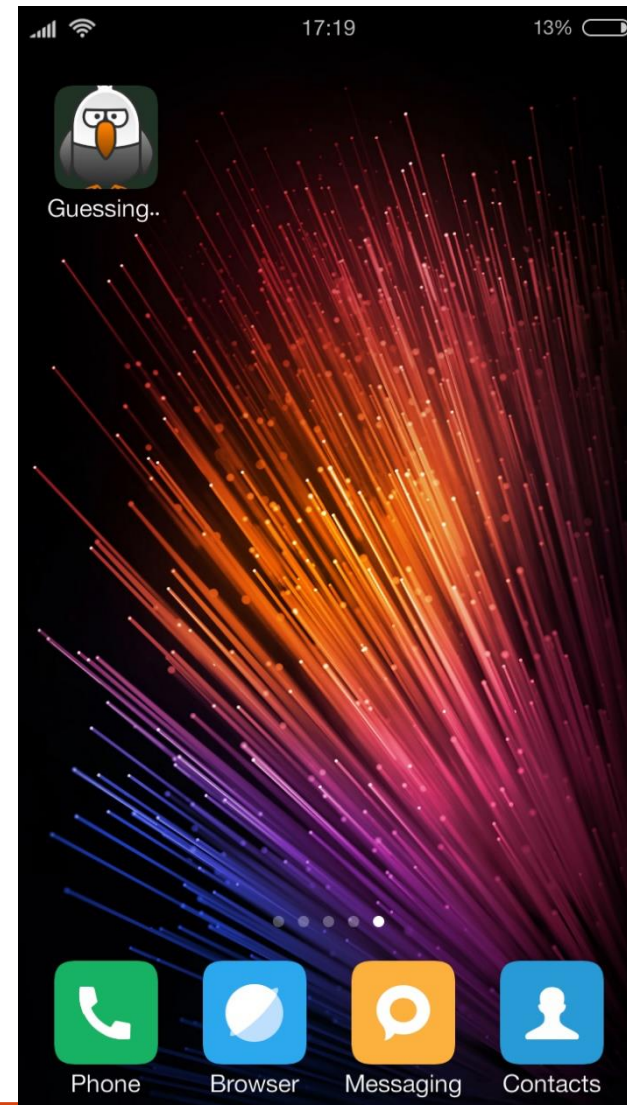
Permission request from
com.android.keyguard

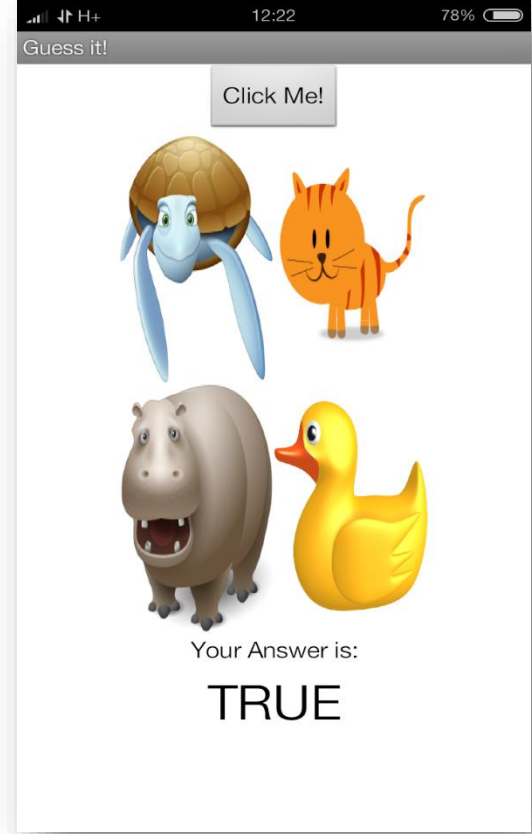
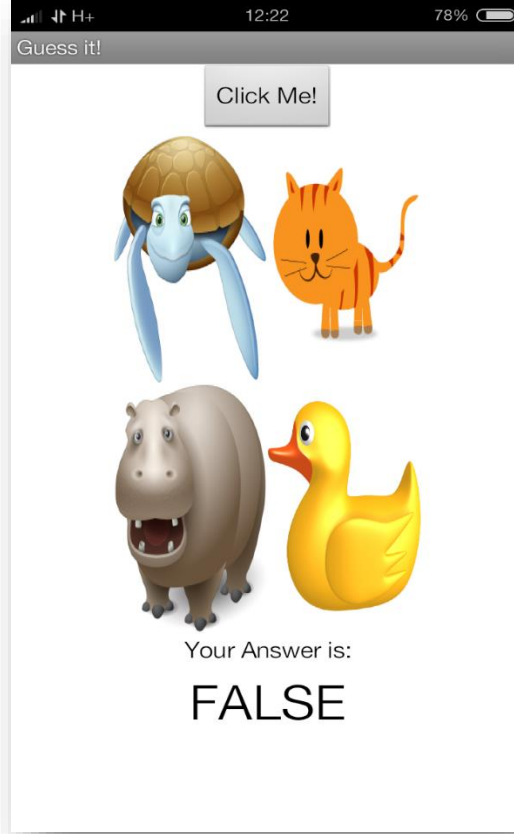
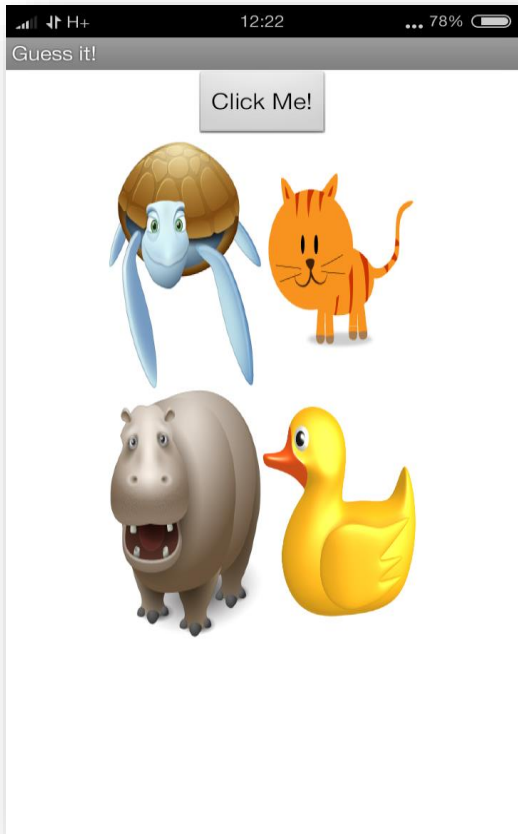
Security | 1

Other | 3

Cancel

Install





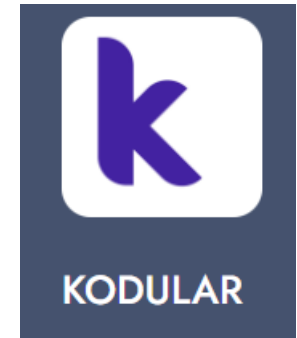
Alternatives: Thunkable & Kodular



MIT
APP INVENTOR



VS



VS



Thunkable

Updated material design for Android 4+
Google Maps component
Ability to import custom .ttf fonts
Extra camera component methods
Extra WebViewer properties

TUGAS INDIVIDU

Buat Aplikasi sederhana dengan salah satu
AppInventor/Thunkable/Kodular

(1 screen saja)

Terapkan minimal:

1 layout DAN 4 komponen BERBEDA (bebas) dengan fungsional
bebas. **Aplikasi HARUS** memiliki icon unik di launcher!

Yg dikumpulkan dokumen berisi

1. nama dan NIM
2. Layout DAN Komponen yg diterapkan
3. Code Blocks
4. Beberapa screenshot saat aplikasi running

Dikonversi jadi file PDF dan dikumpulkan via eling

TERIMA KASIH