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**

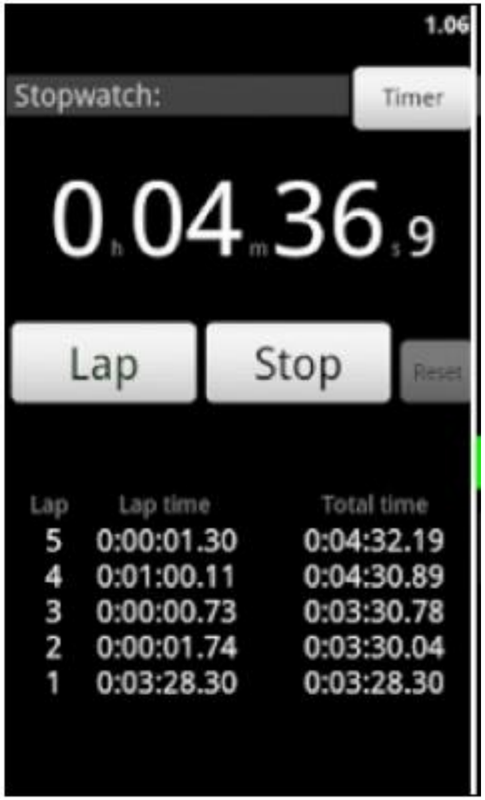
AppInventor

intro

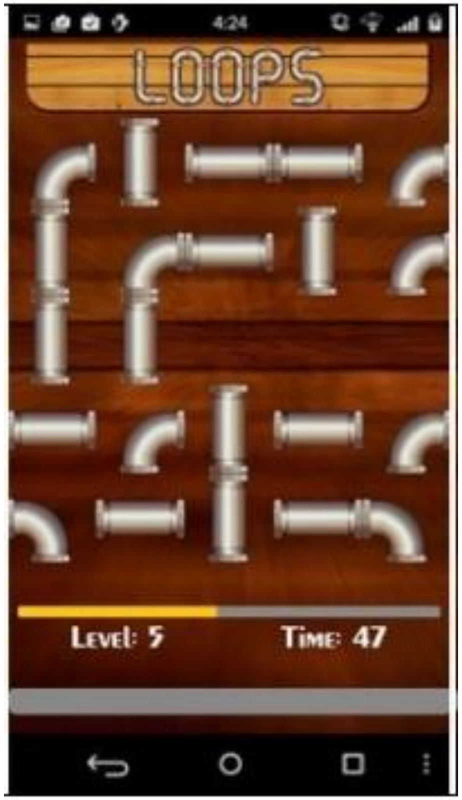
MIT App Inventor 2

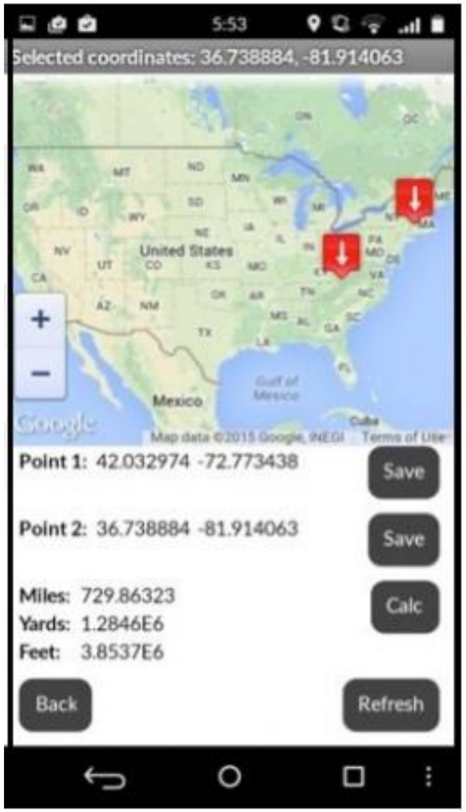
- MIT App Inventor 2 is a free, drag-and-drop, blocks-based visual programming language that enables people, regardless of their coding experience, to create mobile apps for Android devices.
- MIT professor Hal Abelson had the idea to create an easy-to-use programming language to make mobile apps that 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



Lap	Lap time	Total time
5	0:00:01.30	0:04:32.19
4	0:01:00.11	0:04:30.89
3	0:00:00.73	0:03:30.78
2	0:00:01.74	0:03:30.04
1	0:03:28.30	0:03:28.30





Selected coordinates: 36.738884, -81.914063

Point 1: 42.032974 -72.773438

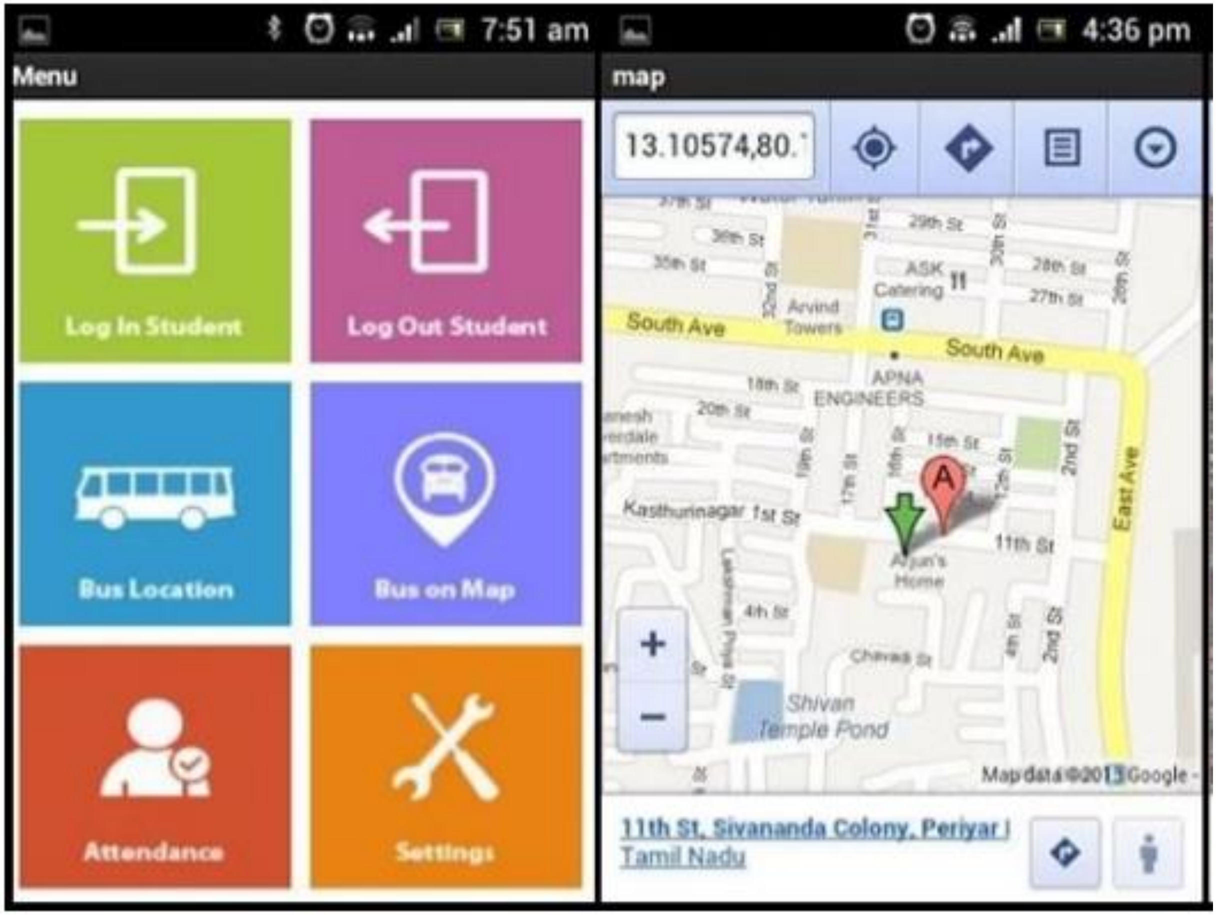
Point 2: 36.738884 -81.914063

Miles: 729.86323

Yards: 1.2846E6

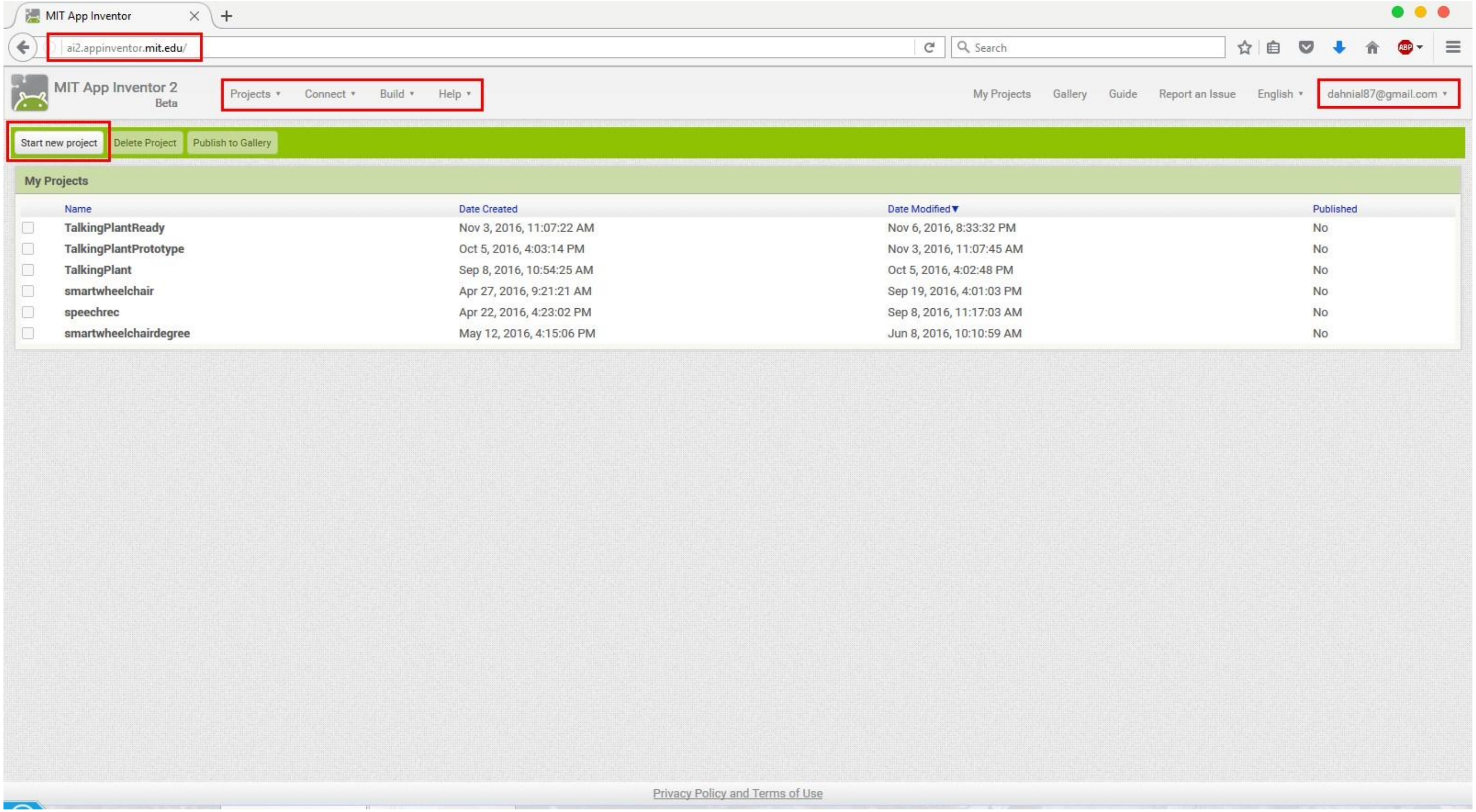
Feet: 3.8537E6

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**

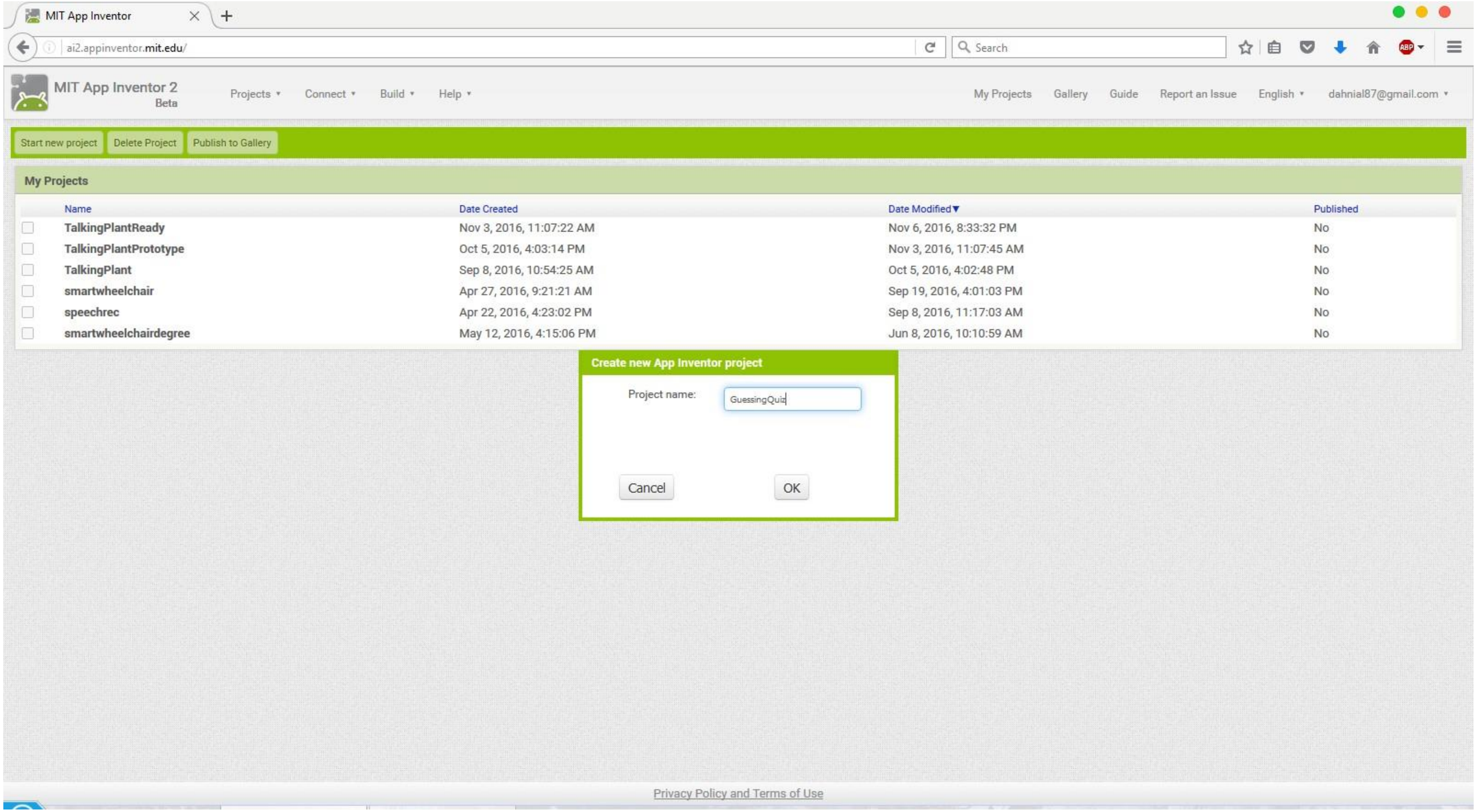
User Interface



My Projects

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

Create New Project



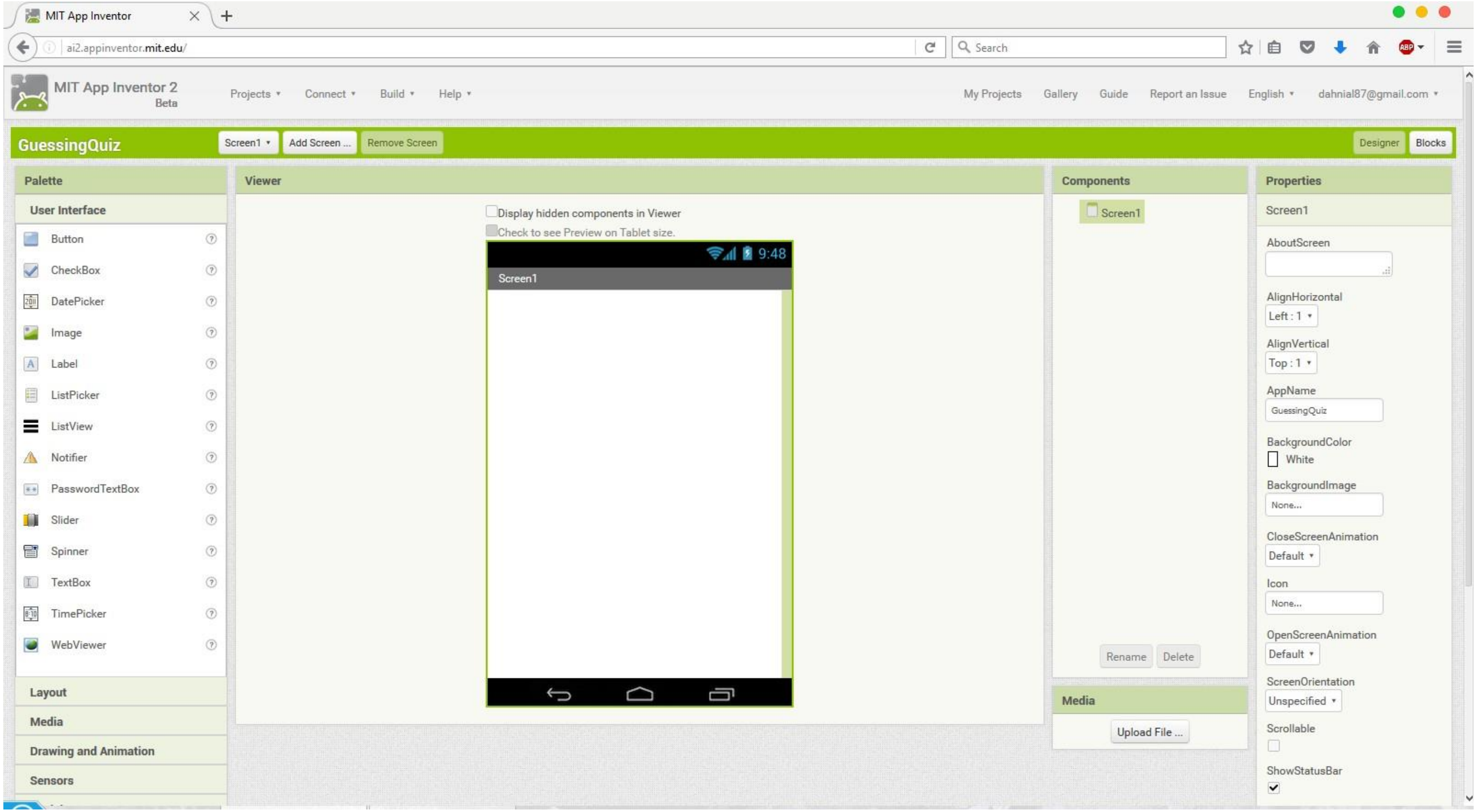
My Projects

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

Create new App Inventor project

Project name: GuessingQuiz

UI: Designer



Screen1

AboutScreen

AlignHorizontal: Left : 1 ▾

AlignVertical: Top : 1 ▾

AppName: GuessingQuiz

BackgroundColor: □ White

BackgroundImage: None...

CloseScreenAnimation: Default ▾

Icon: None...

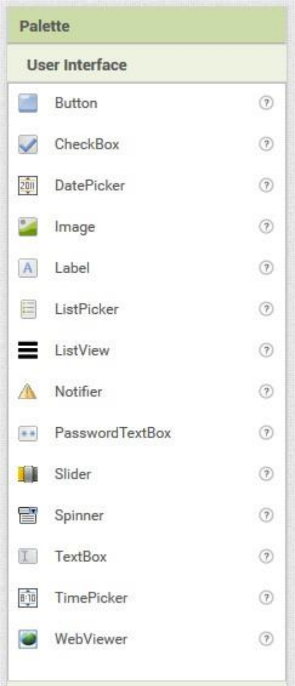
OpenScreenAnimation: Default ▾

ScreenOrientation: Unspecified ▾

Scrollable: □

ShowStatusBar: ☑

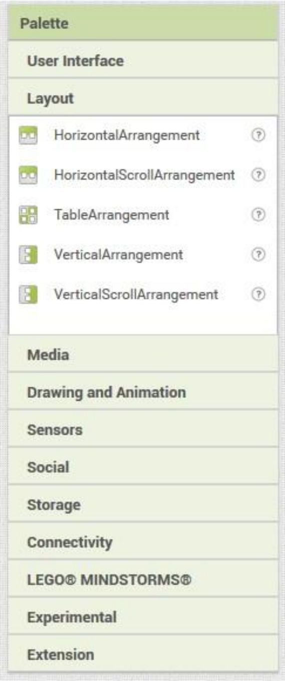
UI: designer palette



Palette

User Interface

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



Palette

User Interface

Layout

- HorizontalArrangement
- HorizontalScrollArrangement
- TableArrangement
- VerticalArrangement
- VerticalScrollArrangement

Media

Drawing and Animation

Sensors

Social

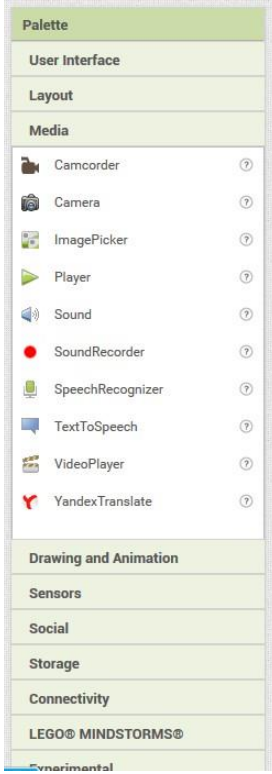
Storage

Connectivity

LEGO® MINDSTORMS®

Experimental

Extension



Palette

User Interface

Layout

Media

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

Drawing and Animation

Sensors

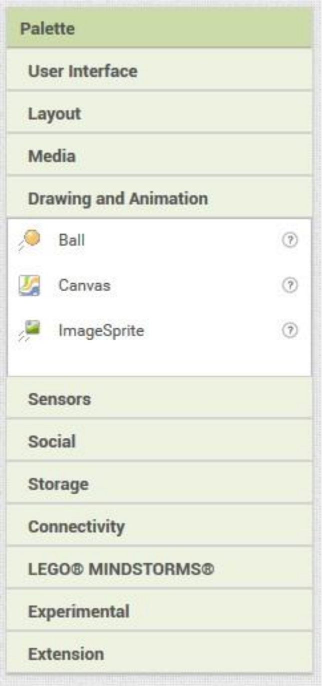
Social

Storage

Connectivity

LEGO® MINDSTORMS®

Experimental



Palette

User Interface

Layout

Media

Drawing and Animation

- Ball
- Canvas
- ImageSprite

Sensors

Social

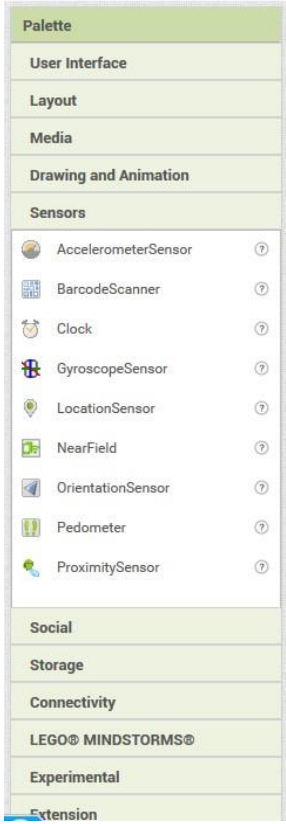
Storage

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

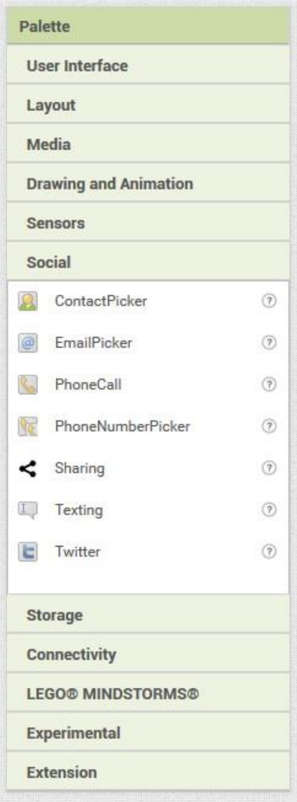
Connectivity

LEGO® MINDSTORMS®

Experimental

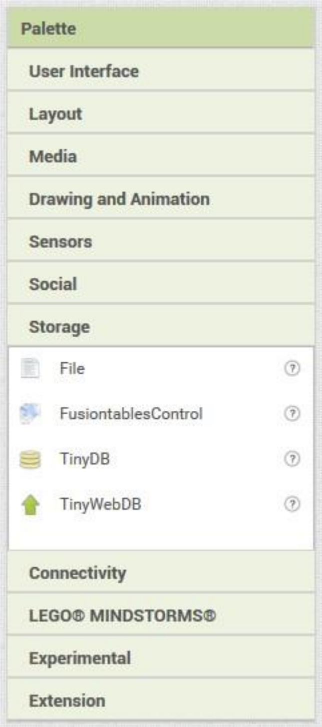
Extension

UI: designer palette



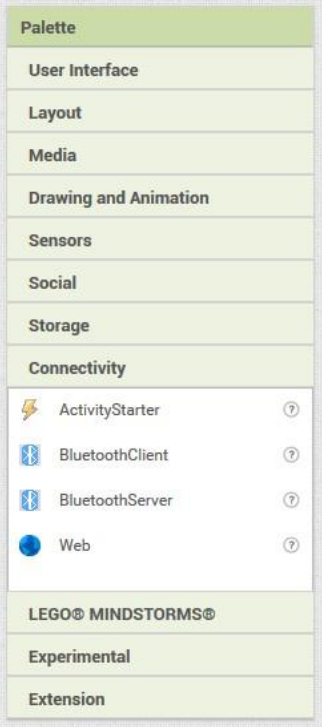
Palette

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



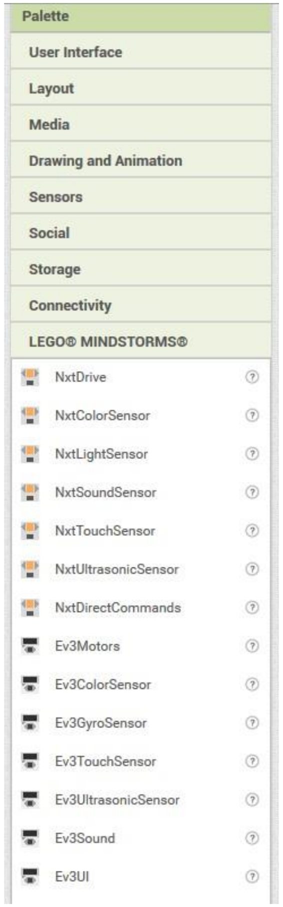
Palette

- User Interface
- Layout
- Media
- Drawing and Animation
- Sensors
- Social
- Storage
  - File
  - FusiontablesControl
  - TinyDB
  - TinyWebDB
- Connectivity
- LEGO® MINDSTORMS®
- Experimental
- Extension



Palette

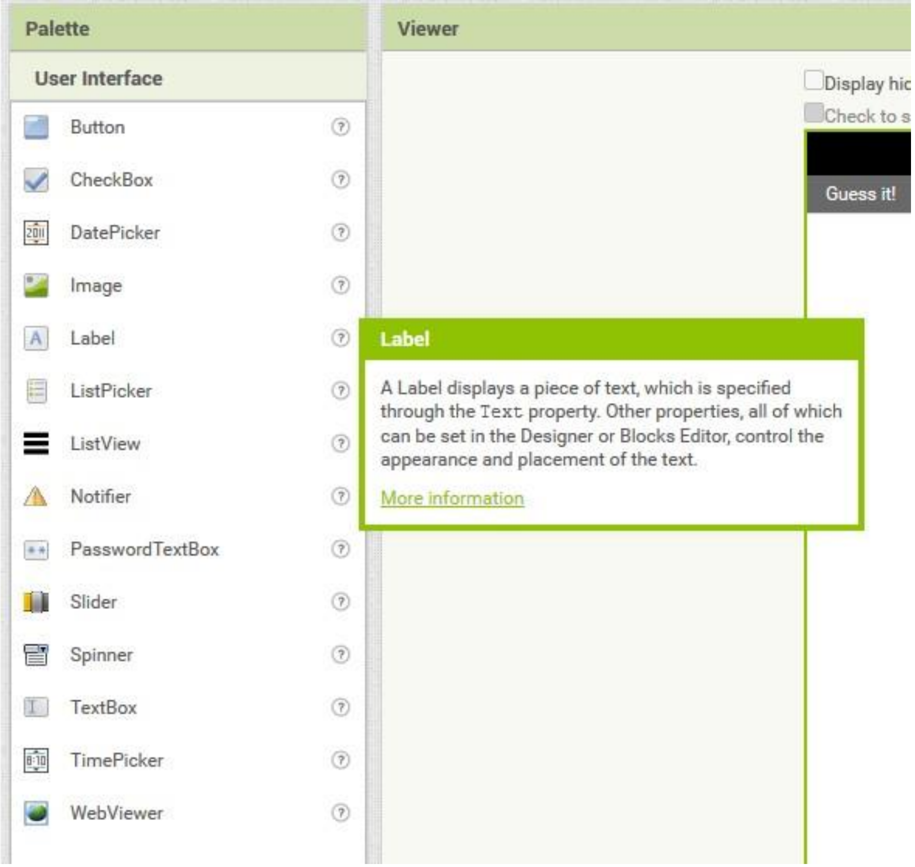
- User Interface
- Layout
- Media
- Drawing and Animation
- Sensors
- Social
- Storage
- Connectivity
  - ActivityStarter
  - BluetoothClient
  - BluetoothServer
  - Web
- LEGO® MINDSTORMS®
- Experimental
- Extension



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

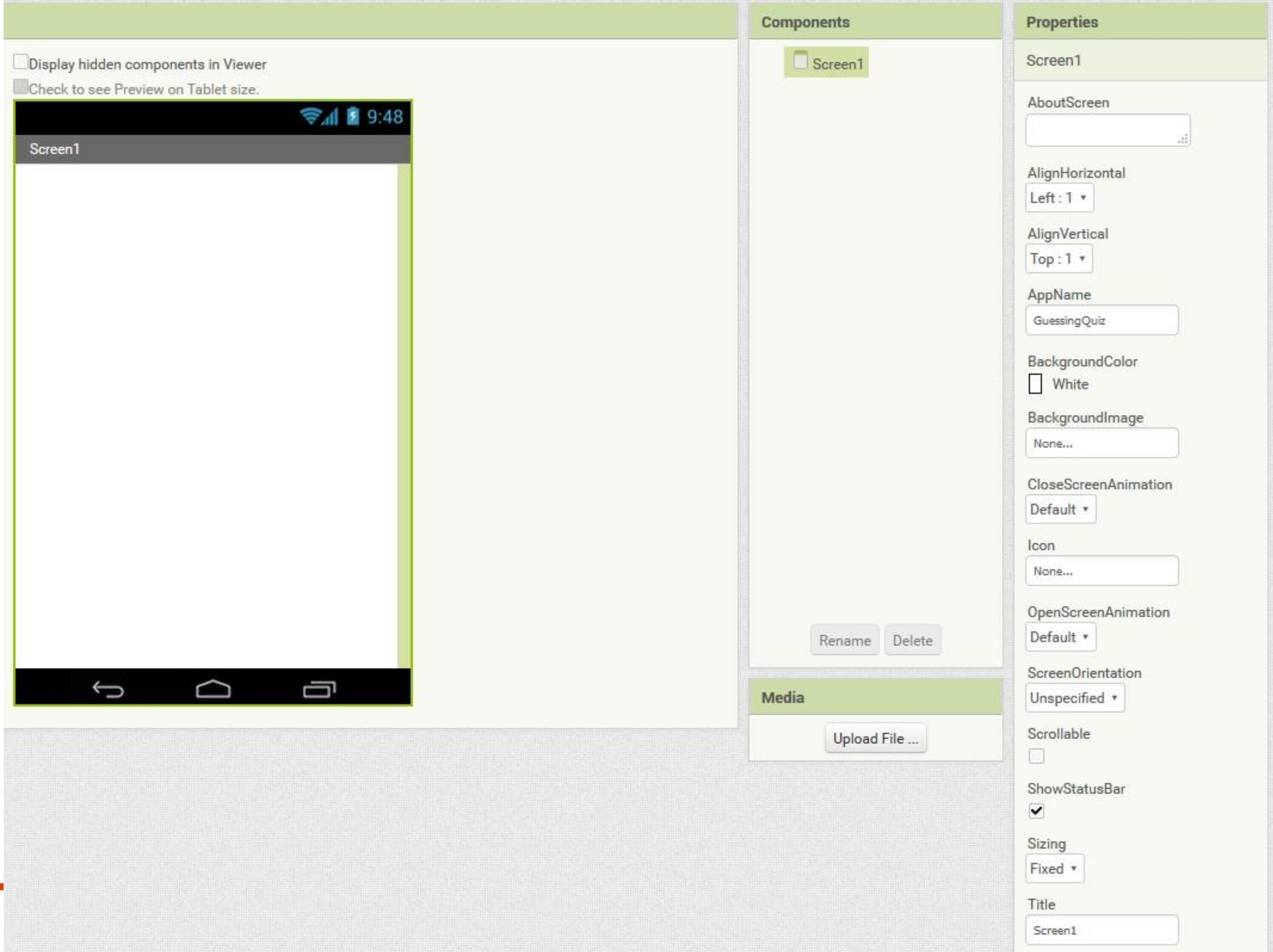


Label

A Label displays a piece of text, which is specified through the `Text` property. Other properties, all of which can be set in the Designer or Blocks Editor, control the appearance and placement of the text.

More information

Adding resource, for ex. Icon, sound, image



☐ Display hidden components in Viewer

☐ Check to see Preview on Tablet size.

Screen1

Screen1

Screen1

AboutScreen

AlignHorizontal
Left : 1

AlignVertical
Top : 1

AppName
GuessingQuiz

BackgroundColor
□ White

BackgroundImage
None...

CloseScreenAnimation
Default

Icon
None...

OpenScreenAnimation
Default

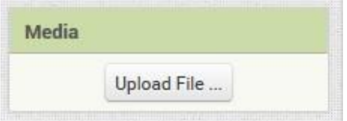
ScreenOrientation
Unspecified

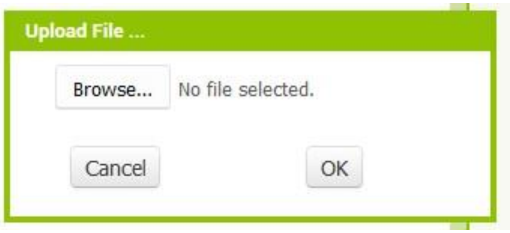
Scrollable
☐

ShowStatusBar
☑

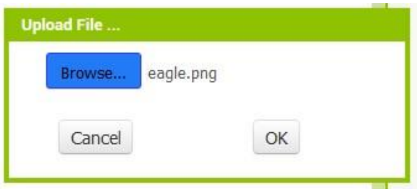
Sizing
Fixed

Title
Screen1

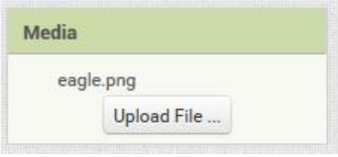




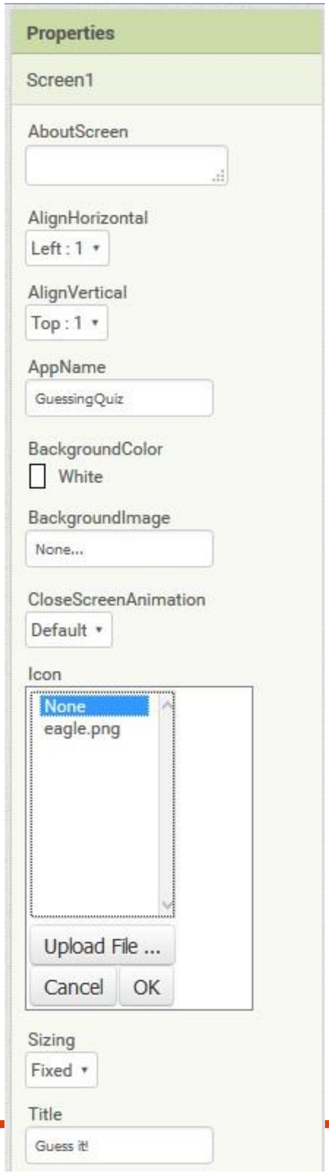
No file selected.



eagle.png



eagle.png



Screen1

AboutScreen

AlignHorizontal: Left : 1

AlignVertical: Top : 1

AppName: GuessingQuiz

BackgroundColor: □ White

BackgroundImage: None...

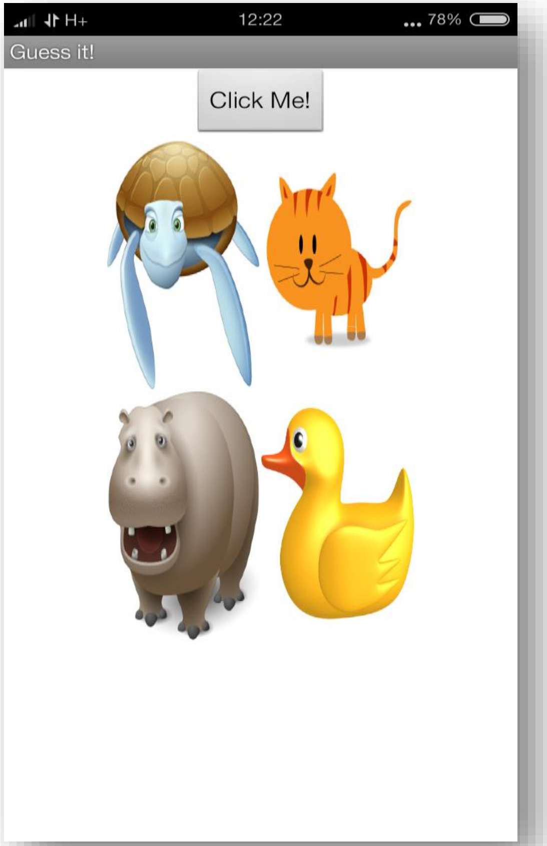
CloseScreenAnimation: Default

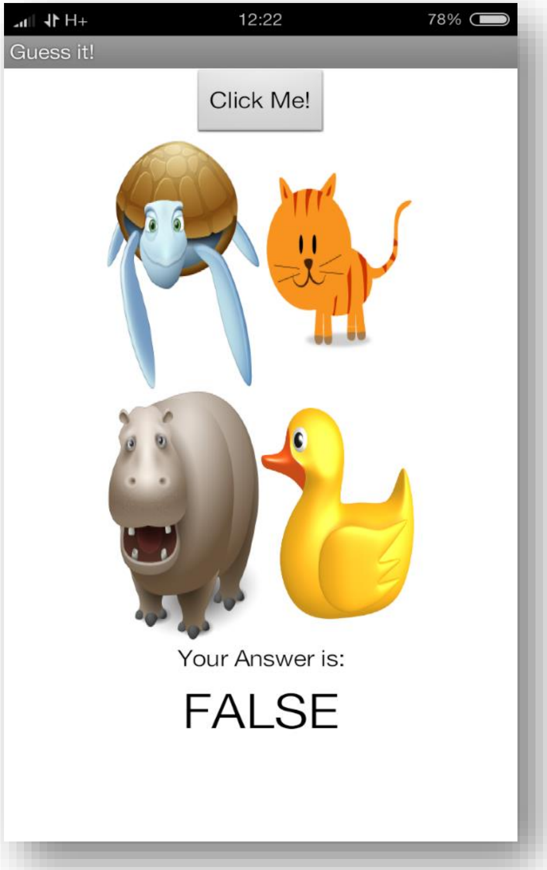
Icon: None
eagle.png

Sizing: Fixed

Title: Guess it!

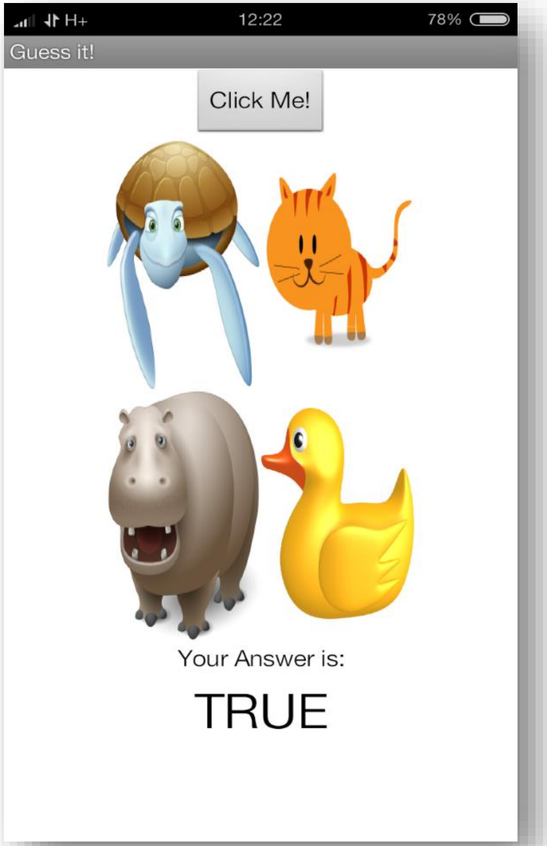
Example: Simple Quiz App





Your Answer is:

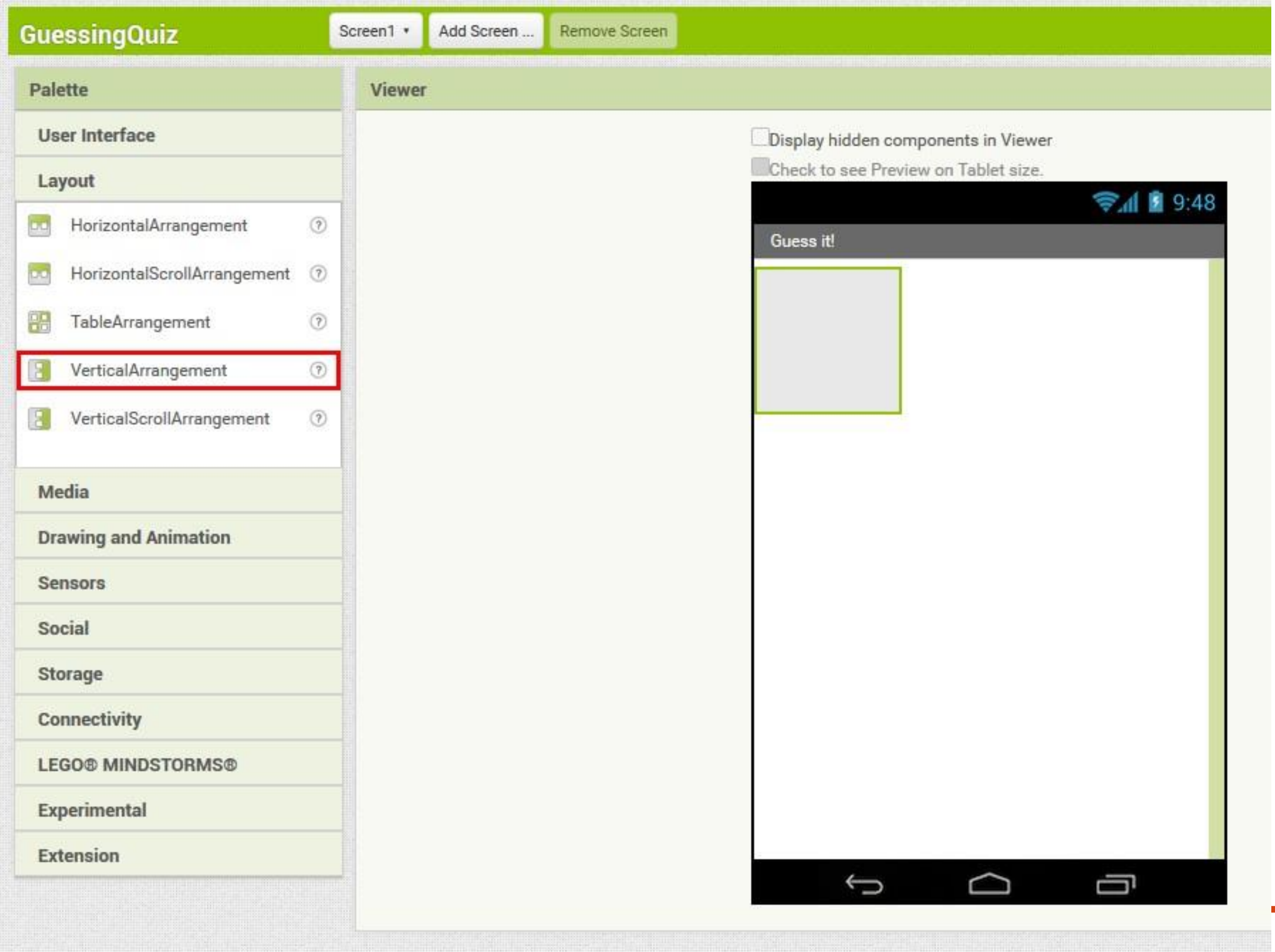
FALSE

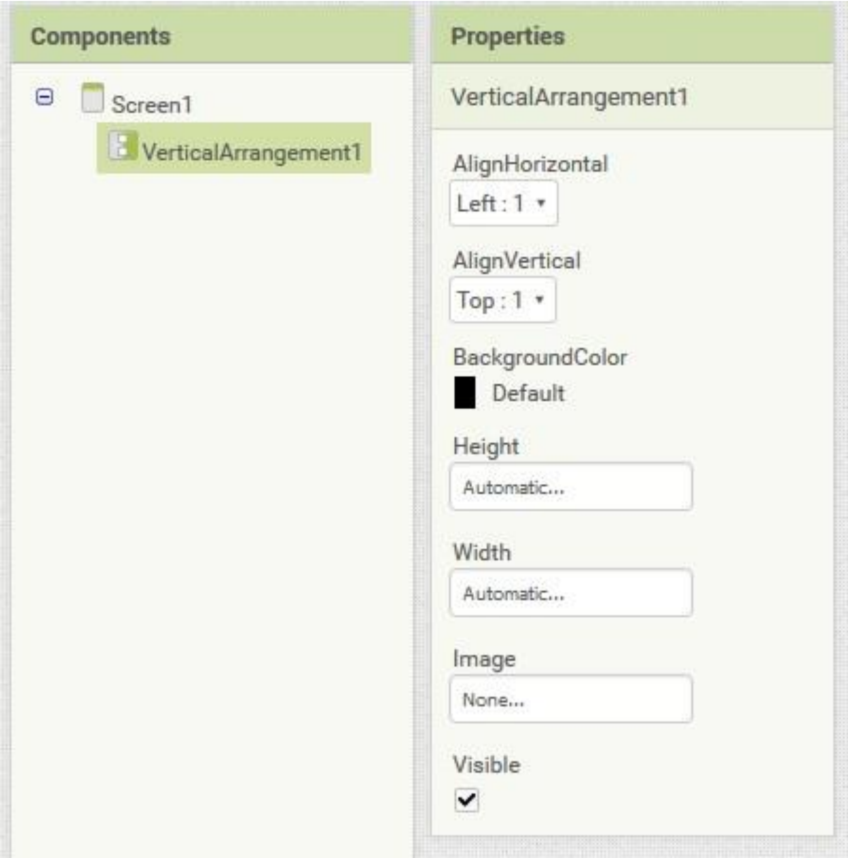


Your Answer is:

TRUE

Using layout: vertical arrangement





Components

Screen1

VerticalArrangement1

Properties

VerticalArrangement1

AlignHorizontal: Left : 1

AlignVertical: Top : 1

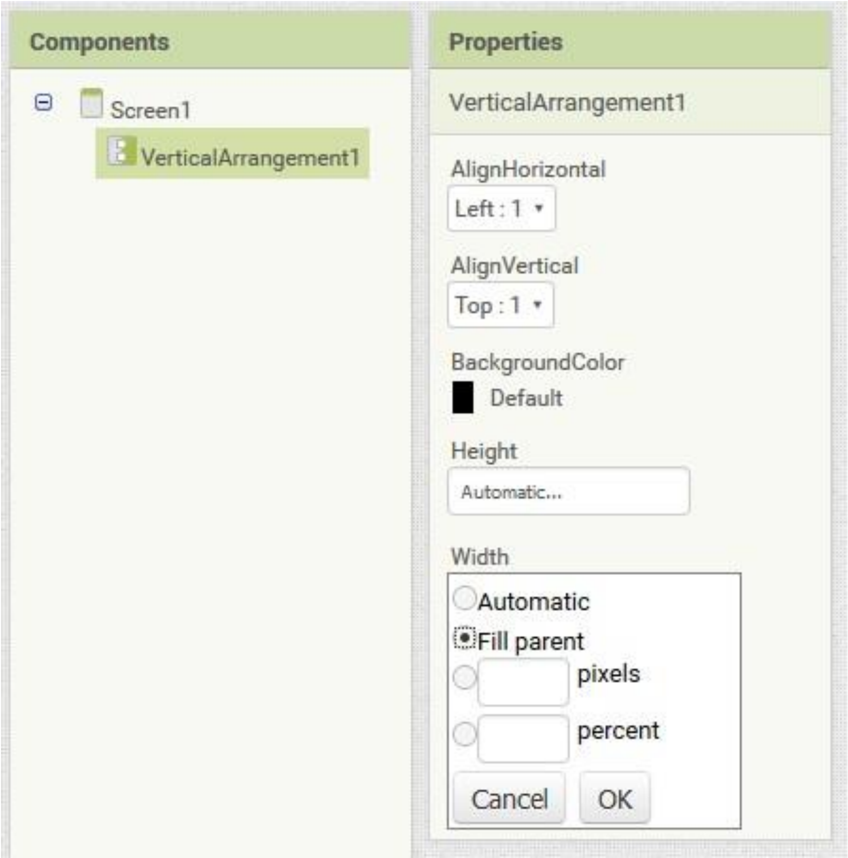
BackgroundColor: ■ Default

Height: Automatic...

Width: Automatic...

Image: None...

Visible: ☑



Components

Screen1

VerticalArrangement1

Properties

VerticalArrangement1

AlignHorizontal: Left : 1

AlignVertical: Top : 1

BackgroundColor: ■ Default

Height: Automatic...

Width

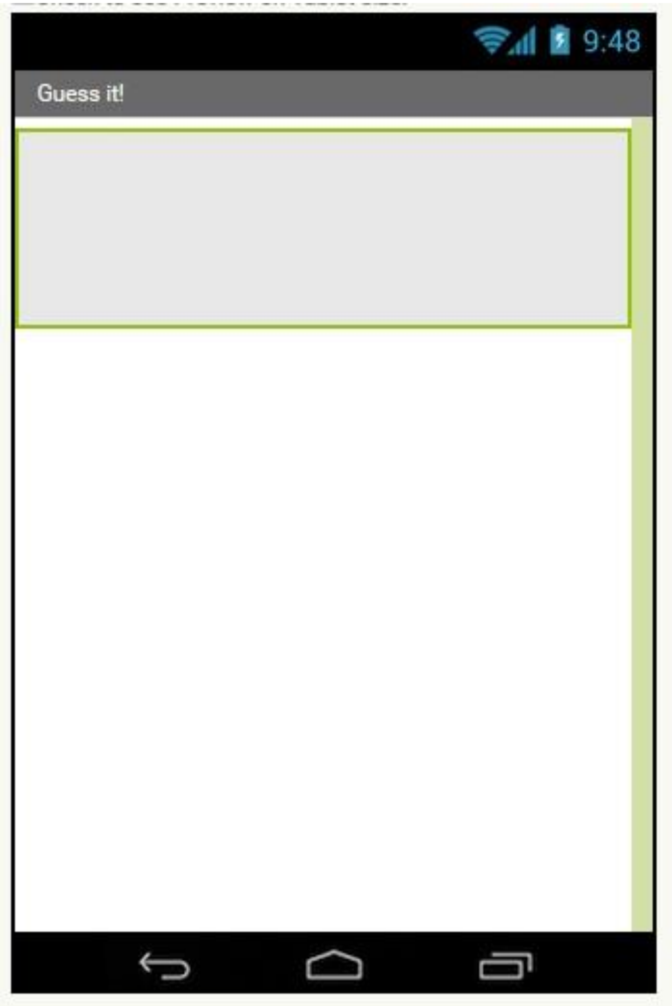
○ Automatic

● Fill parent

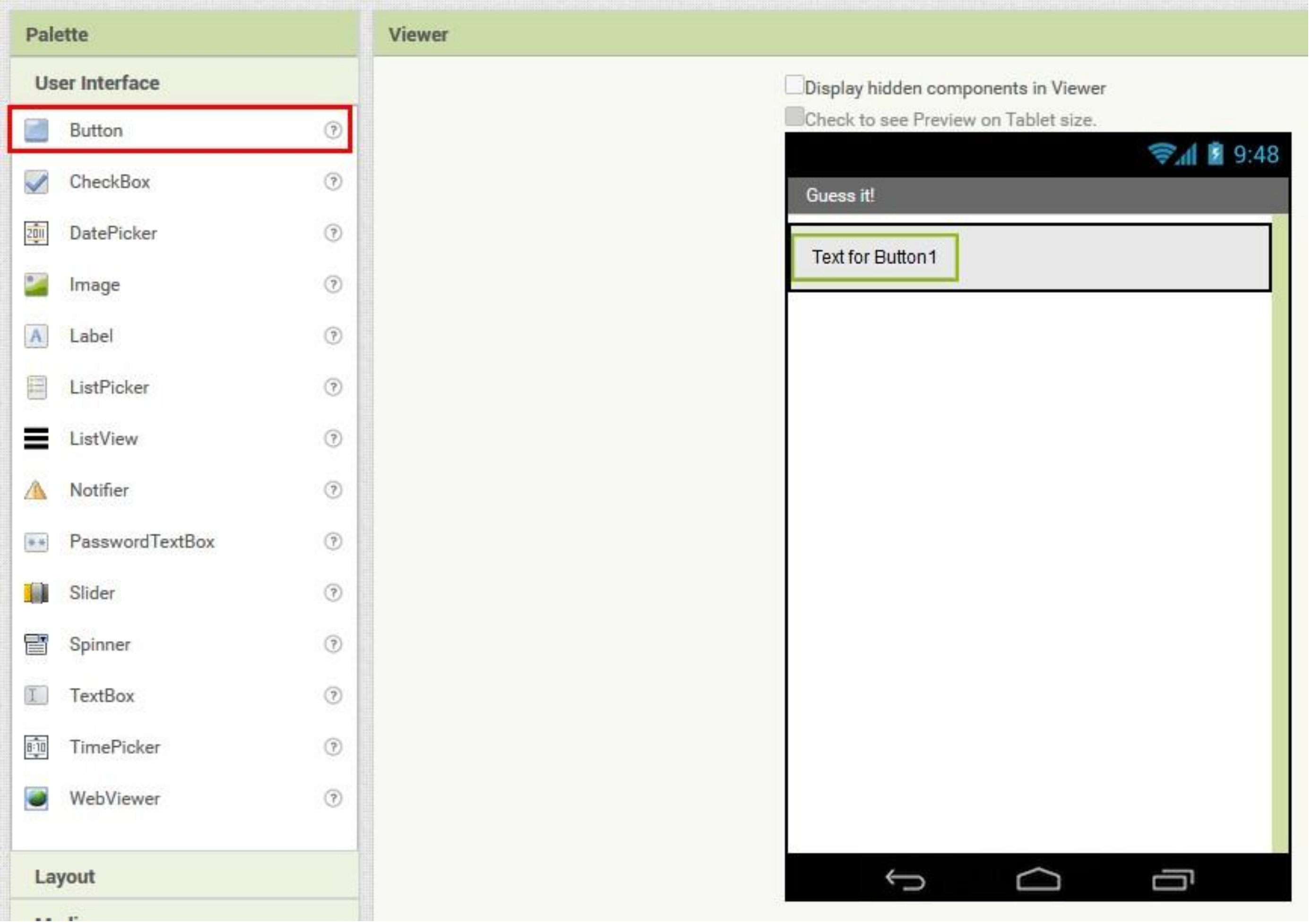
○ pixels

○ percent

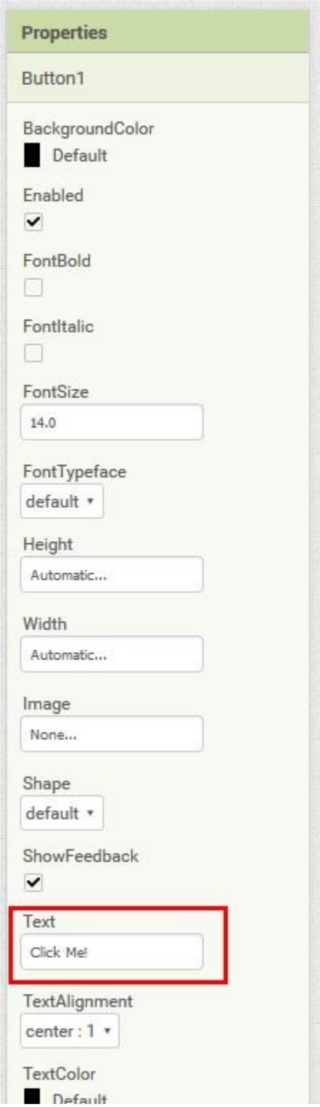
Set the layout so that the width will fill the parent


9:48
Guess it!

Add button from palette



Then change the text caption of the button



Properties

Button1

BackgroundColor
■ Default

Enabled
☑

FontBold
☐

FontItalic
☐

FontSize
14.0

FontTypeface
default ▾

Height
Automatic...

Width
Automatic...

Image
None...

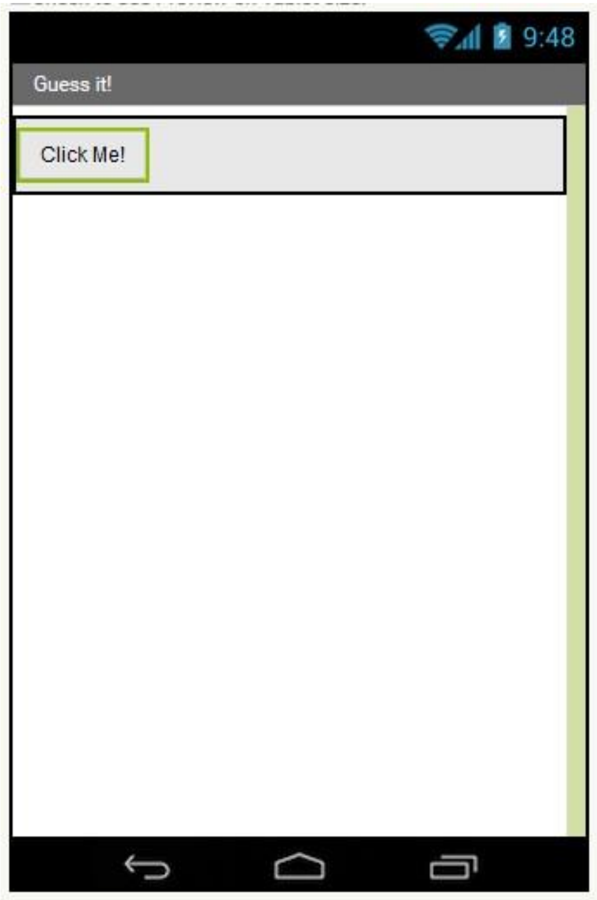
Shape
default ▾

ShowFeedback
☑

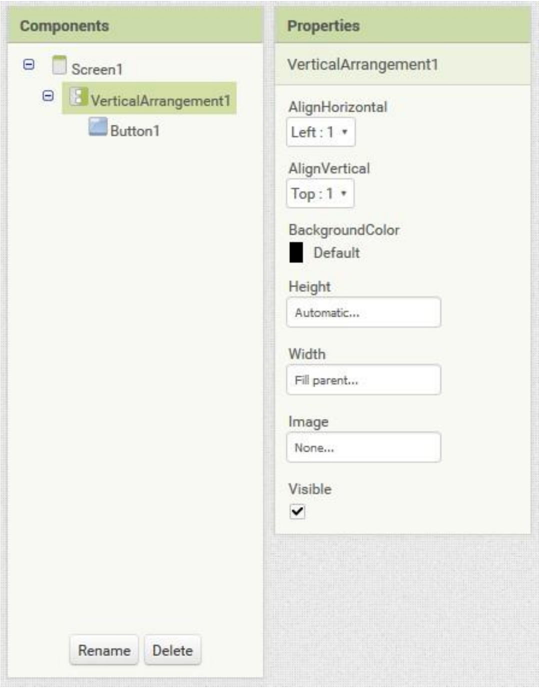
Text
Click Me!

TextAlignment
center : 1 ▾

TextColor
■ Default



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



Components

- Screen1
  - VerticalArrangement1
    - Button1

Rename Delete

Properties

VerticalArrangement1

AlignHorizontal: Left : 1

AlignVertical: Top : 1

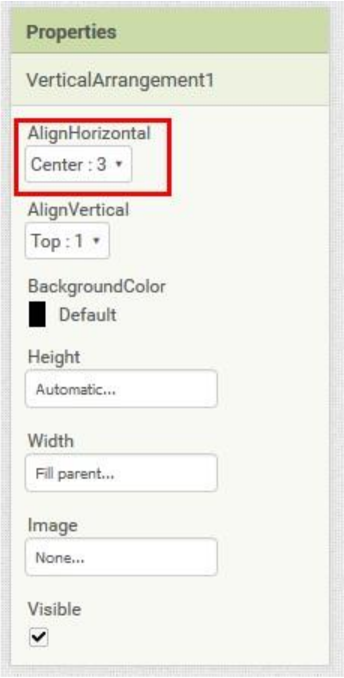
BackgroundColor: Default

Height: Automatic...

Width: Fill parent...

Image: None...

Visible: ☑



Properties

VerticalArrangement1

AlignHorizontal: Center : 3

AlignVertical: Top : 1

BackgroundColor: Default

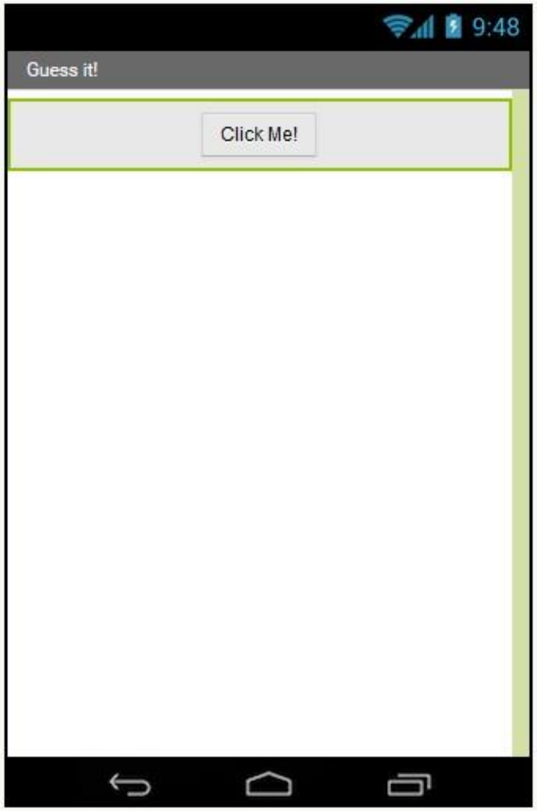
Height: Automatic...

Width: Fill parent...

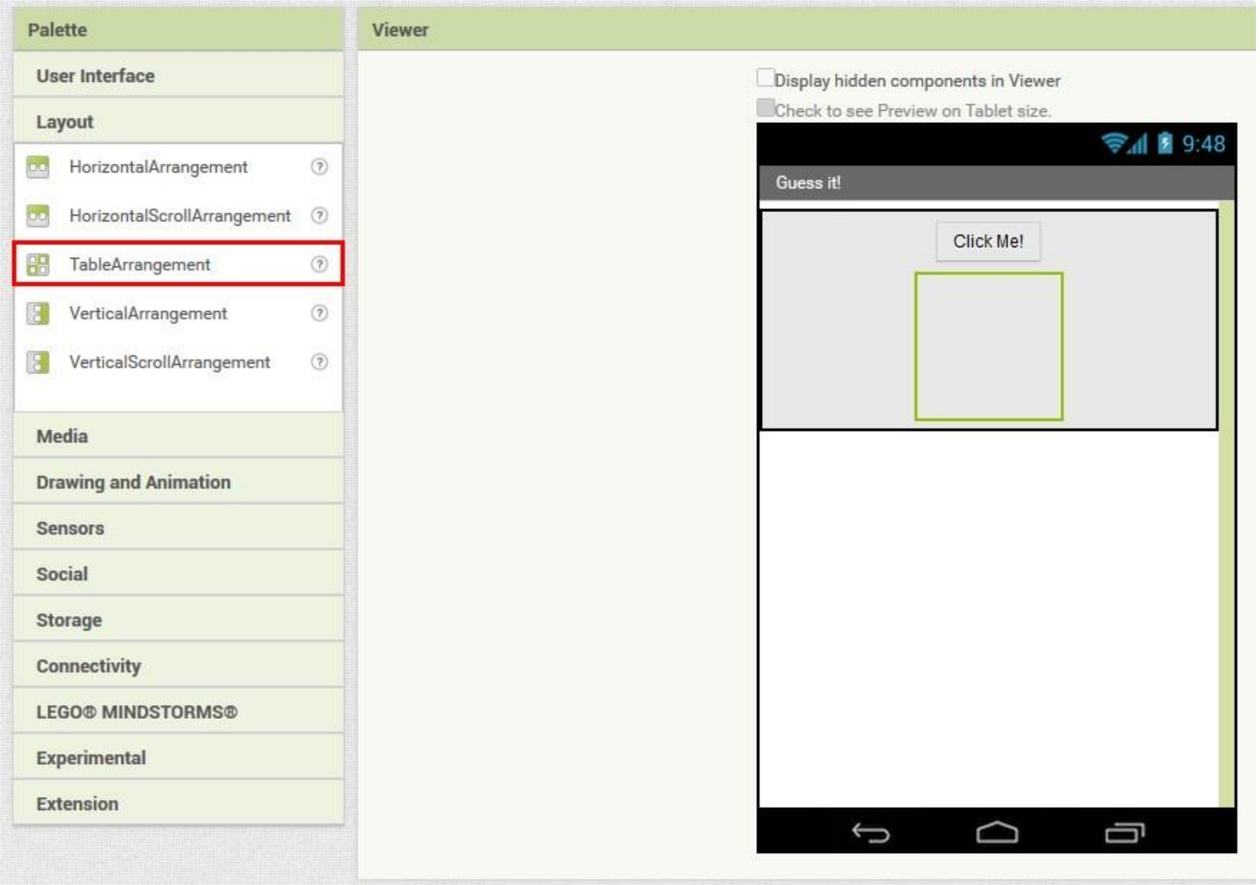
Image: None...

Visible: ☑

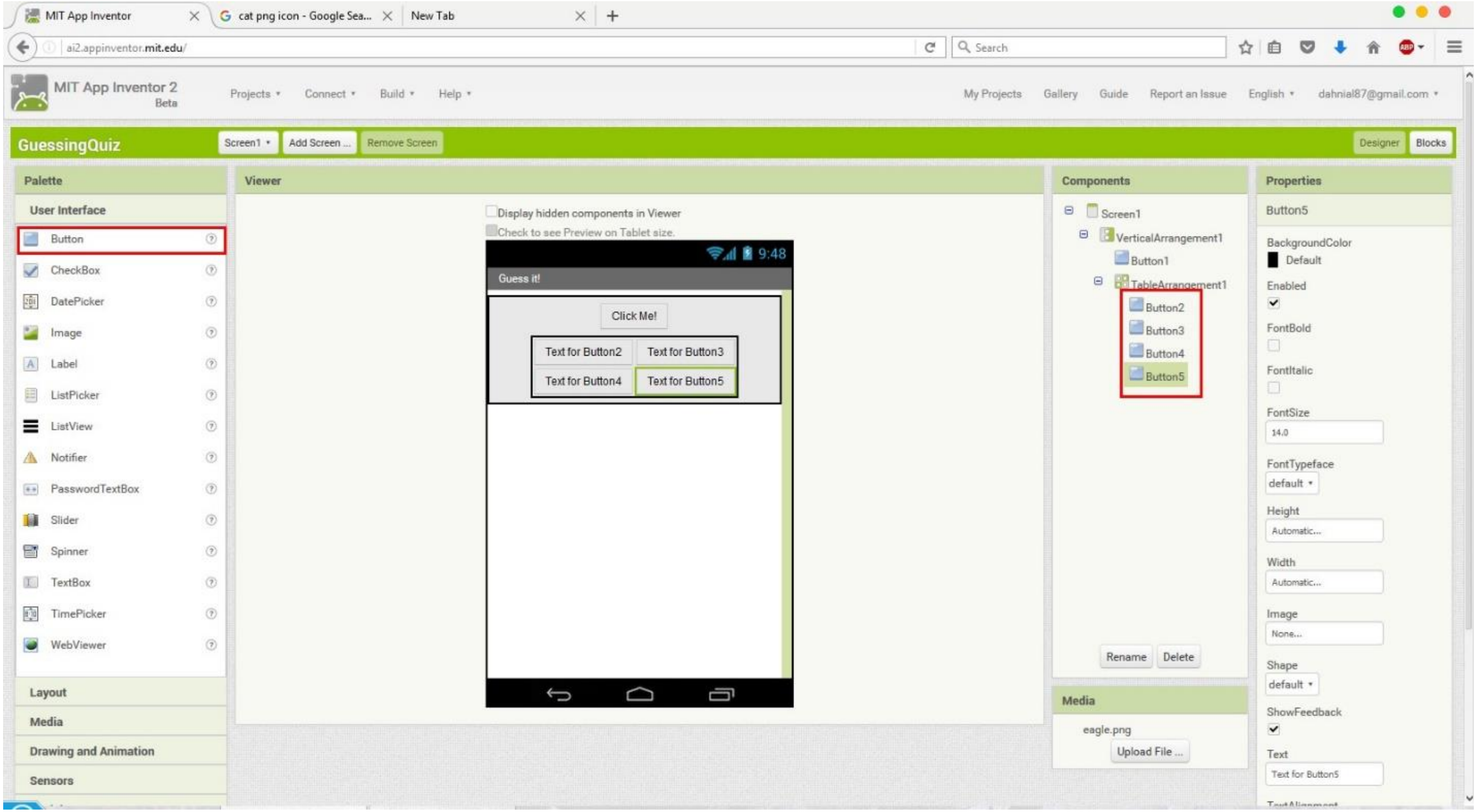
RESULT:



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



Then add the button for each



Button5

BackgroundColor: Default

Enabled: ☑

FontBold: □

FontItalic: □

FontSize: 14.0

FontTypeface: default

Height: Automatic...

Width: Automatic...

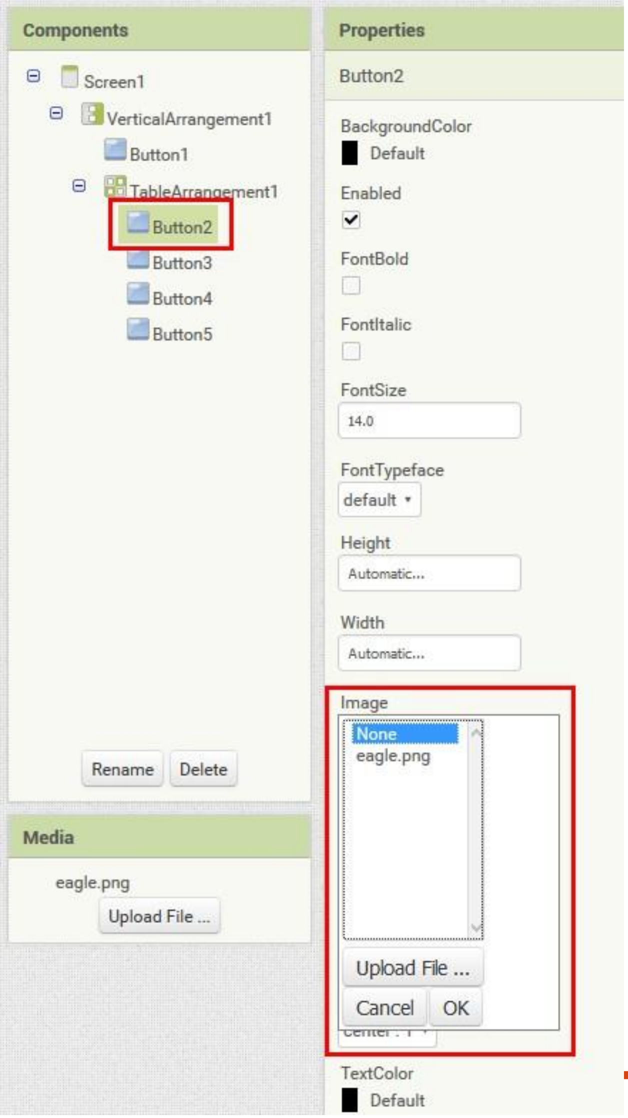
Image: None...

Shape: default

ShowFeedback: ☑

Text: Text for Button5

We will add custom image for each button.



BackgroundColor
Default

Enabled
☑

FontBold
□

FontItalic
□

FontSize
14.0

FontTypeface
default

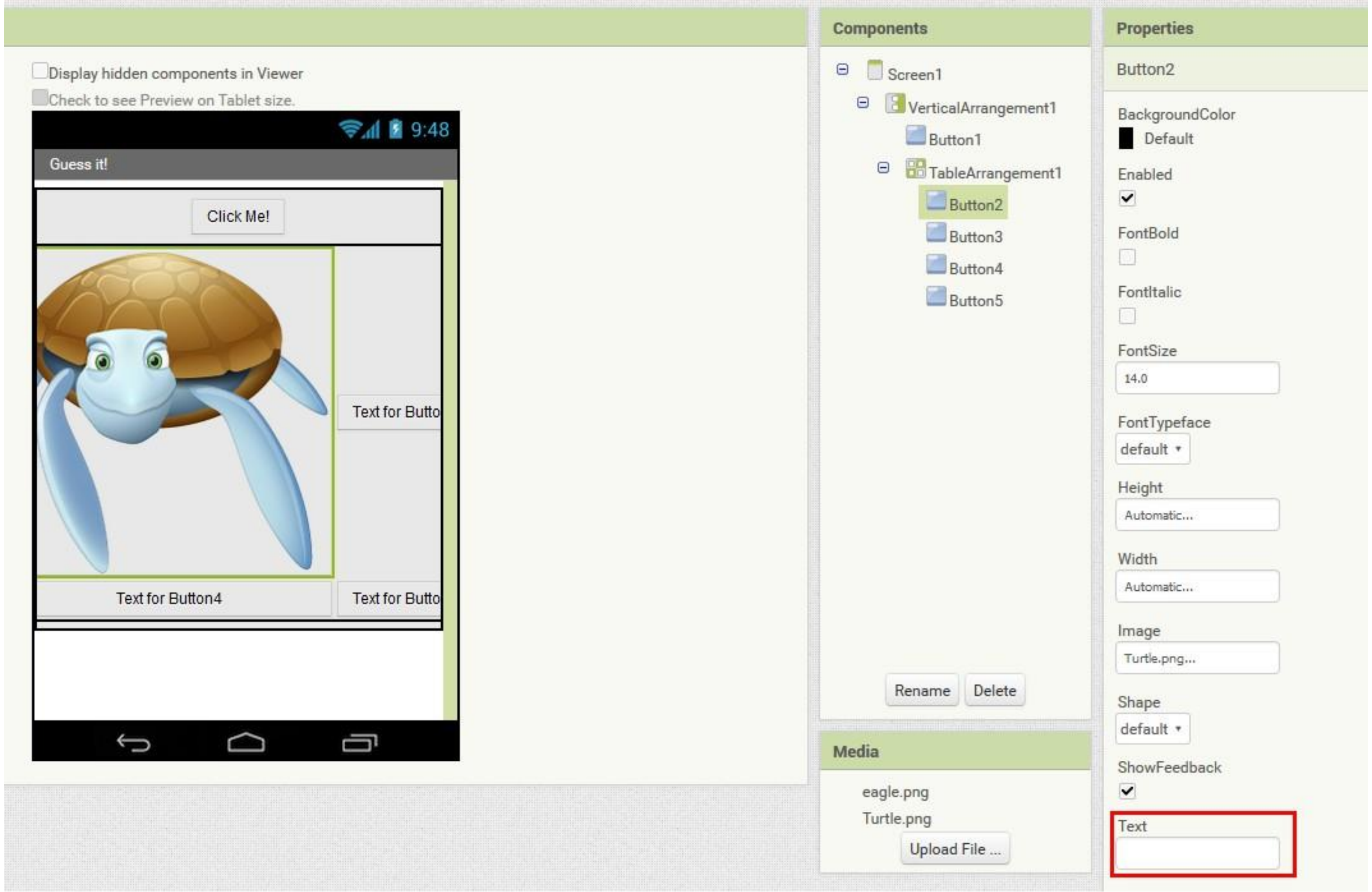
Height
Automatic...

Width
Automatic...

Image
None
eagle.png

TextColor
Default

And remove the text caption in the button



☐ Display hidden components in Viewer

☐ Check to see Preview on Tablet size.

Components

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

Media

- eagle.png
- Turtle.png

Properties

Button2

BackgroundColor: Default

Enabled: ☑

FontBold: ☐

FontItalic: ☐

FontSize: 14.0

FontTypeface: default

Height: Automatic...

Width: Automatic...

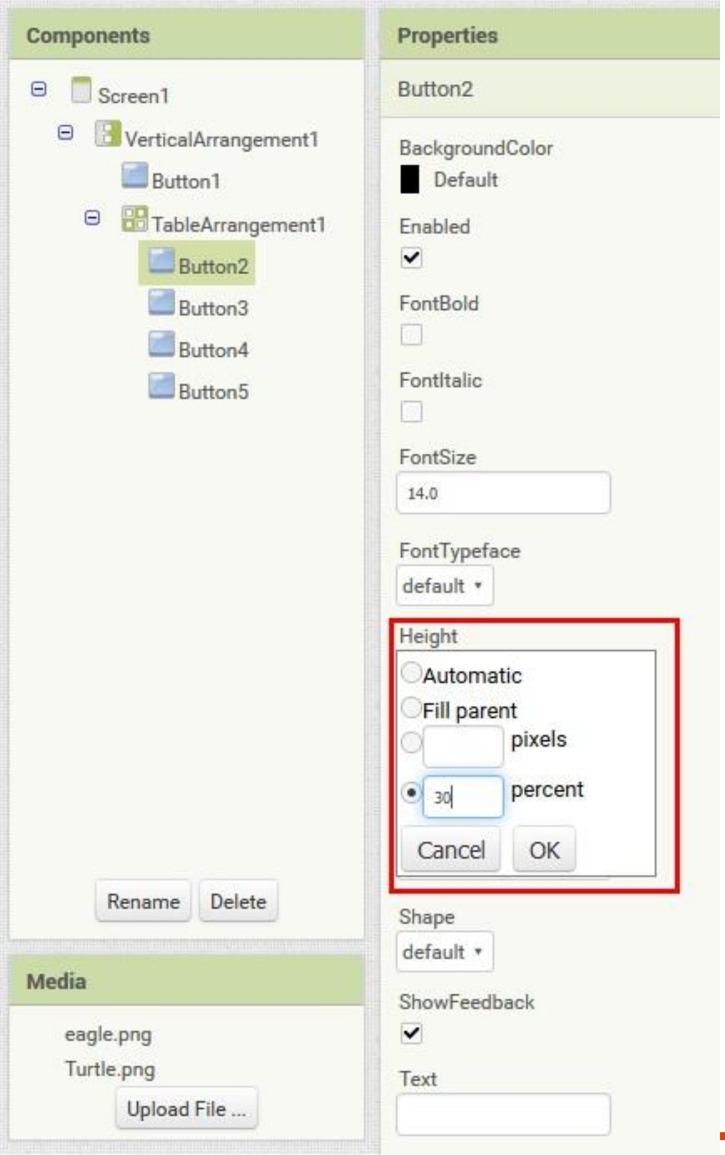
Image: Turtle.png...

Shape: default

ShowFeedback: ☑

Text:

Resize the button a little bit …



Components

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

Rename Delete

Media

eagle.png
Turtle.png
Upload File ...

Properties

Button2

BackgroundColor
■ Default

Enabled
☑

FontBold
☐

FontItalic
☐

FontSize
14.0

FontTypeface
default ▾

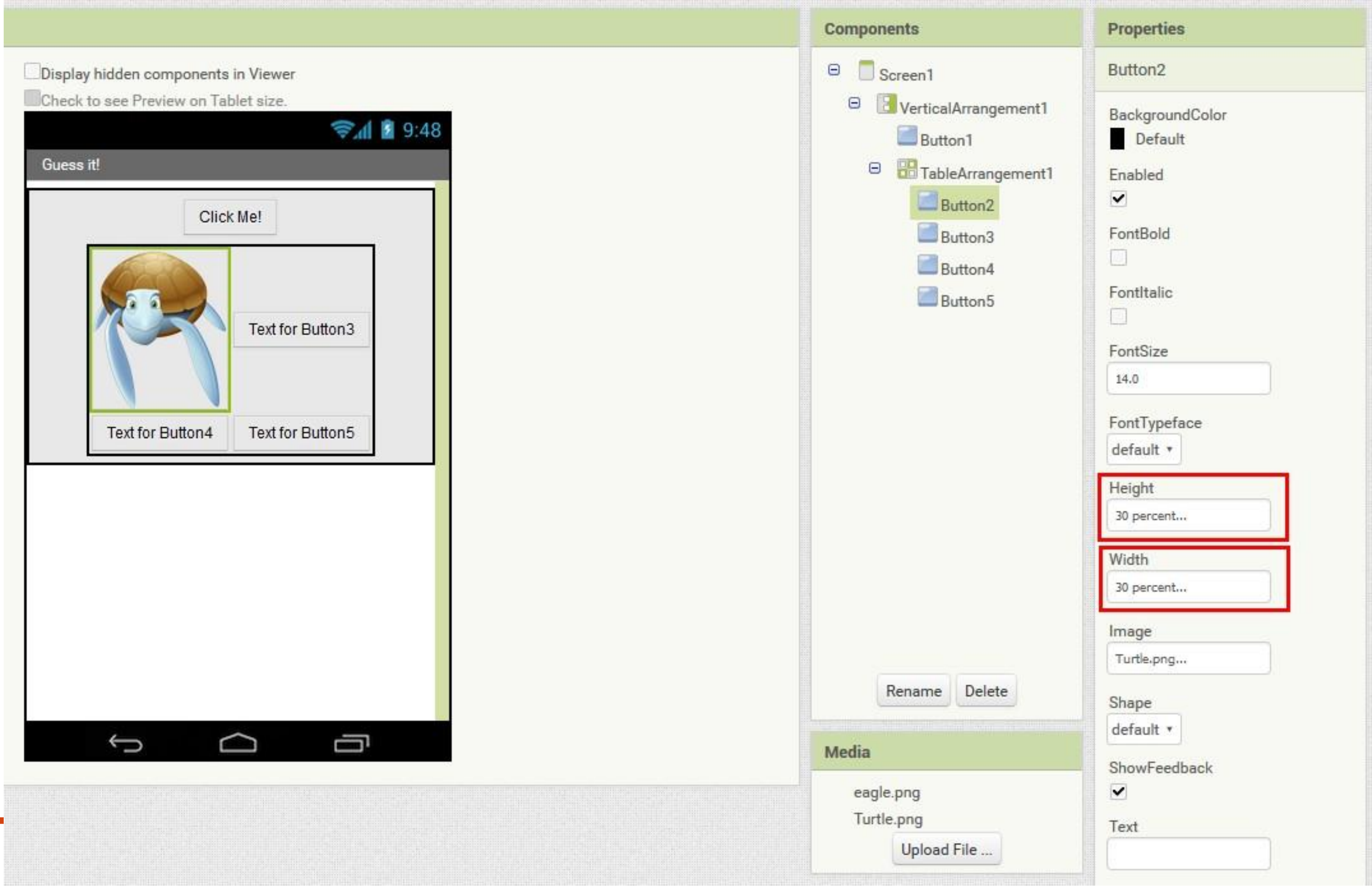
Height
○ Automatic
○ Fill parent
○ pixels
● 30 percent
Cancel OK

Shape
default ▾

ShowFeedback
☑

Text

Resize for the height and width



Button2

BackgroundColor: Default

Enabled: ☑

FontBold: □

FontItalic: □

FontSize: 14.0

FontTypeface: default

Height: 30 percent...

Width: 30 percent...

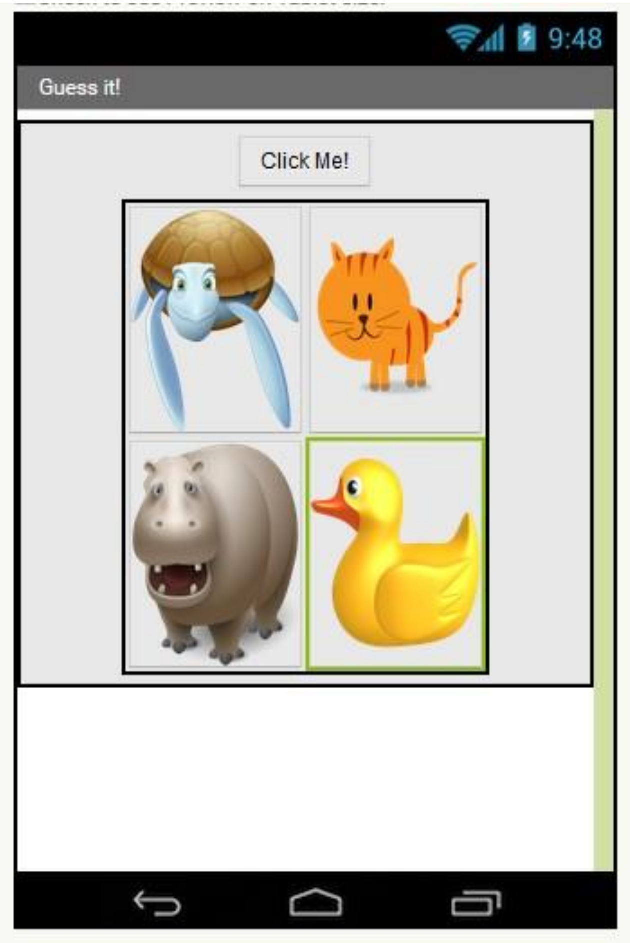
Image: Turtle.png...

Shape: default

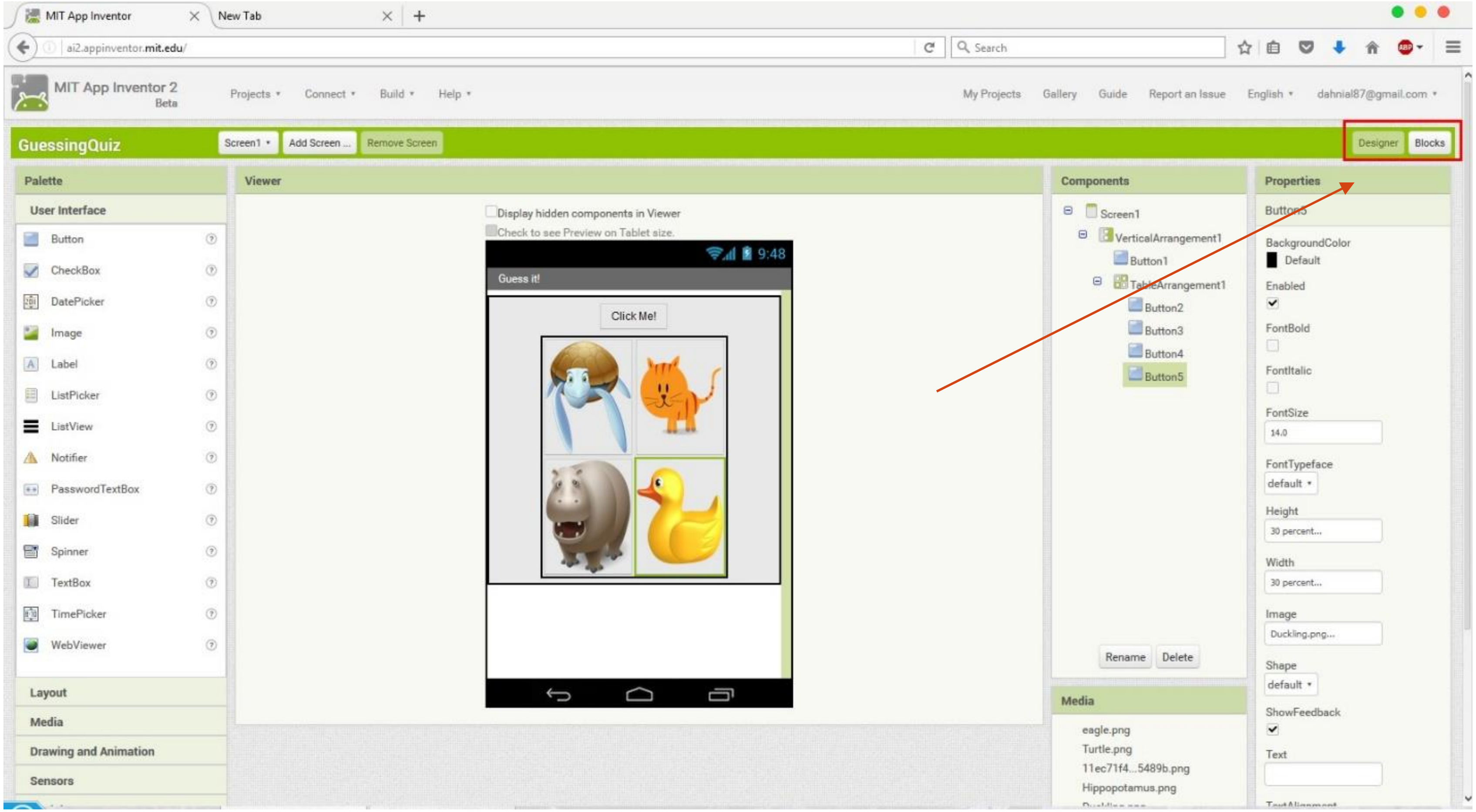
ShowFeedback: ☑

Text:

UI Result:



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



Button5

BackgroundColor
■ Default

Enabled
☑

FontBold
□

FontItalic
□

FontSize
14.0

FontTypeface
default ▾

Height
30 percent...

Width
30 percent...

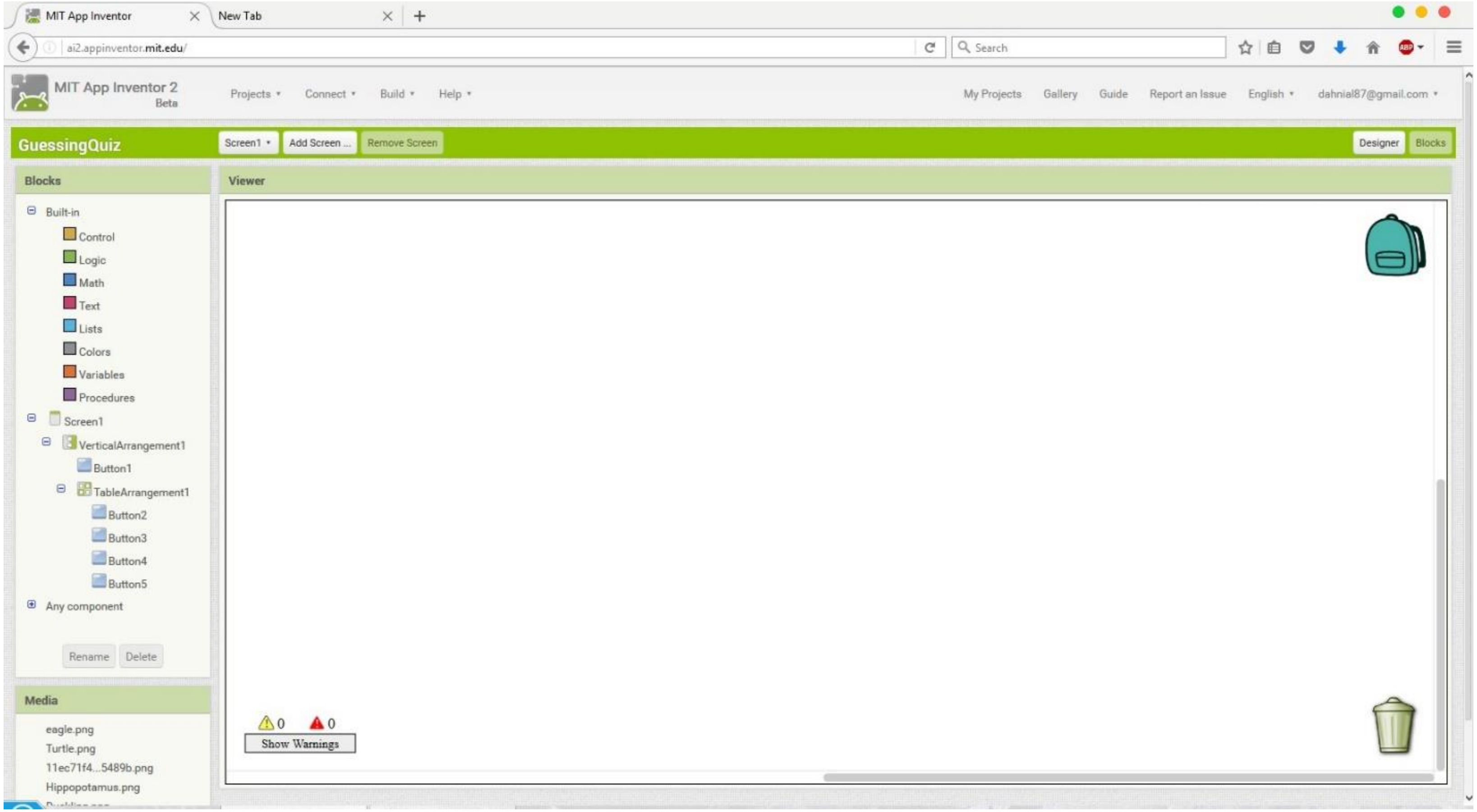
Image
Duckling.png...

Shape
default ▾

ShowFeedback
☑

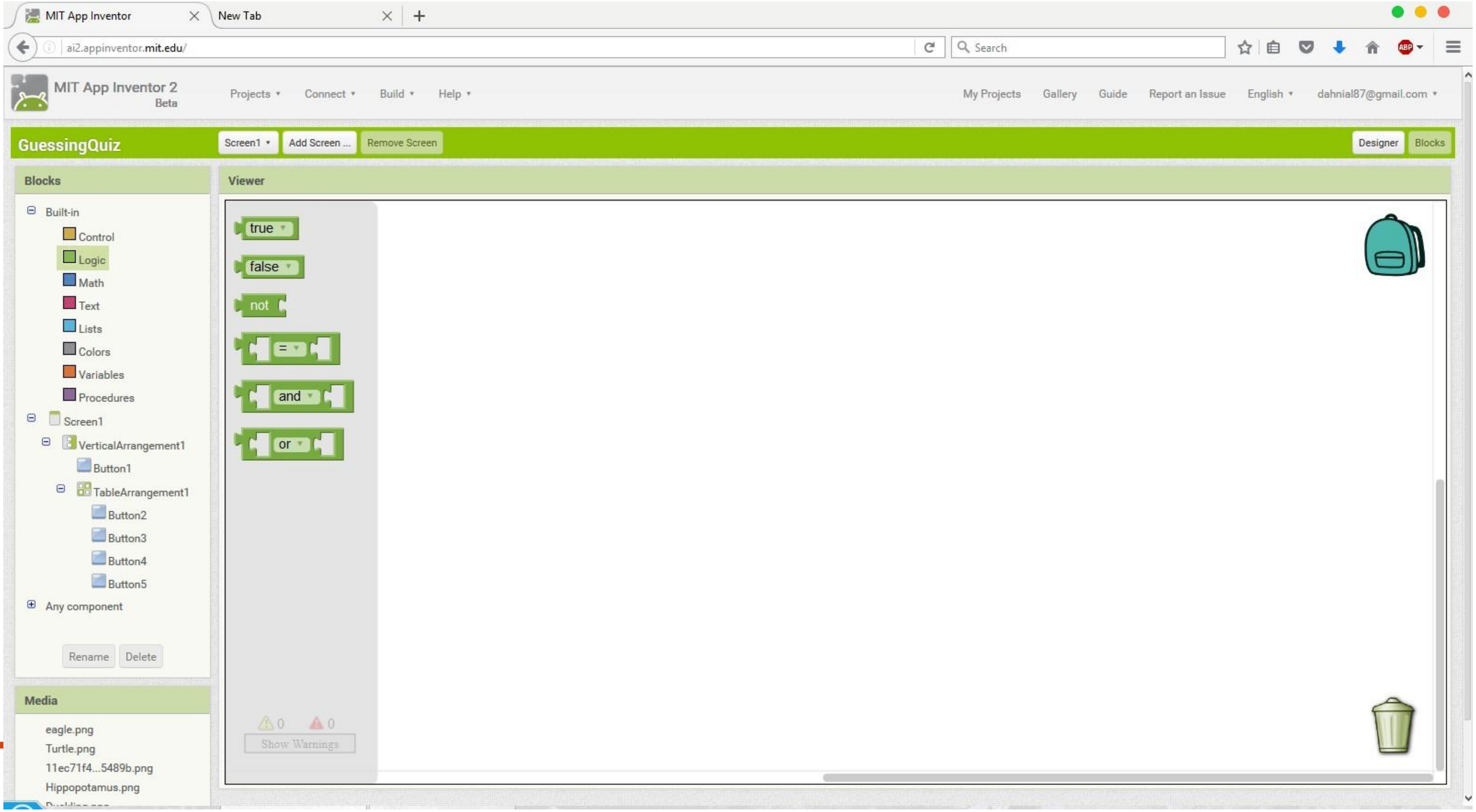
Text

User Interface: Blocks

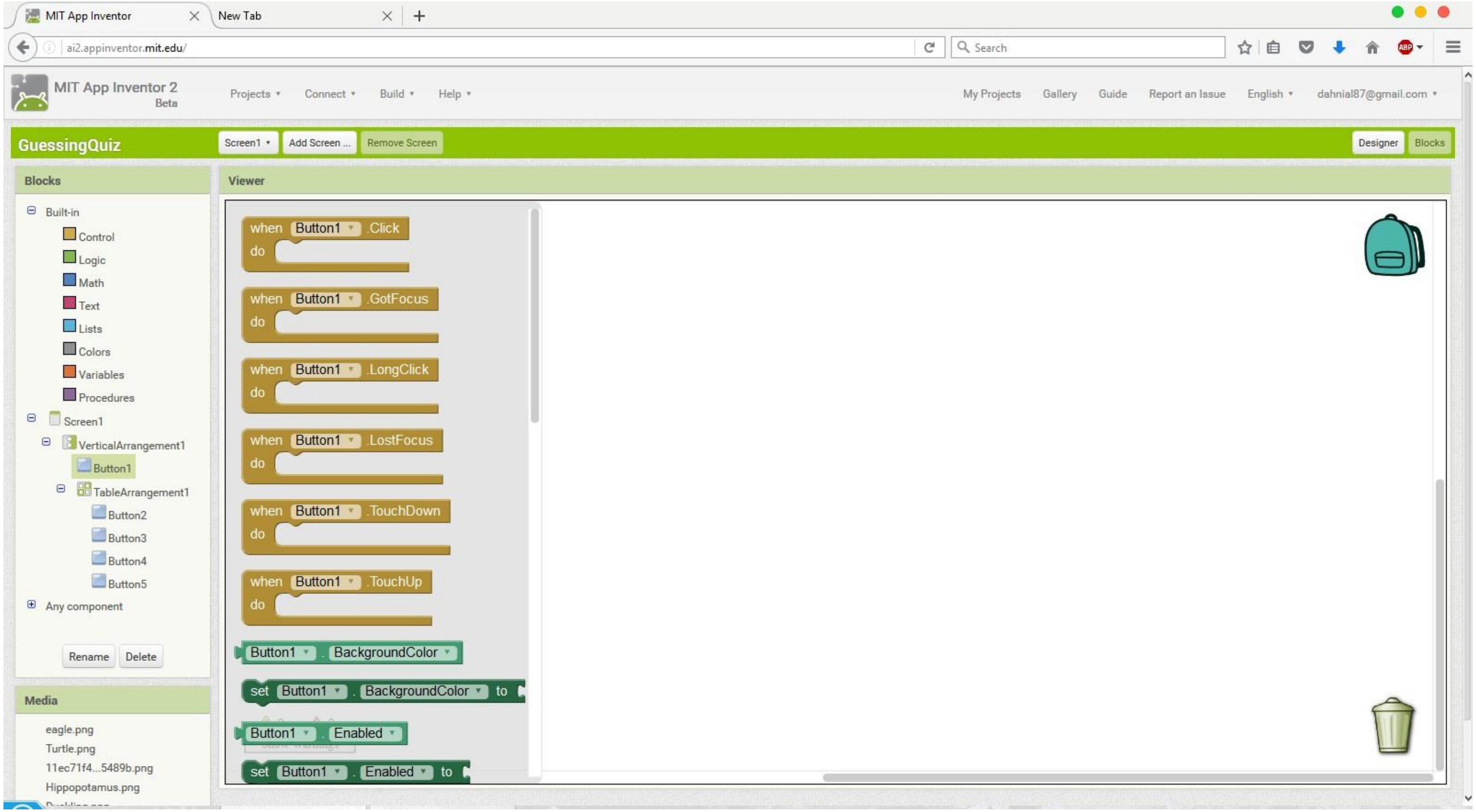


Appinventor uses block programming instead of scripting. Let's take a look:

Logic Blocks

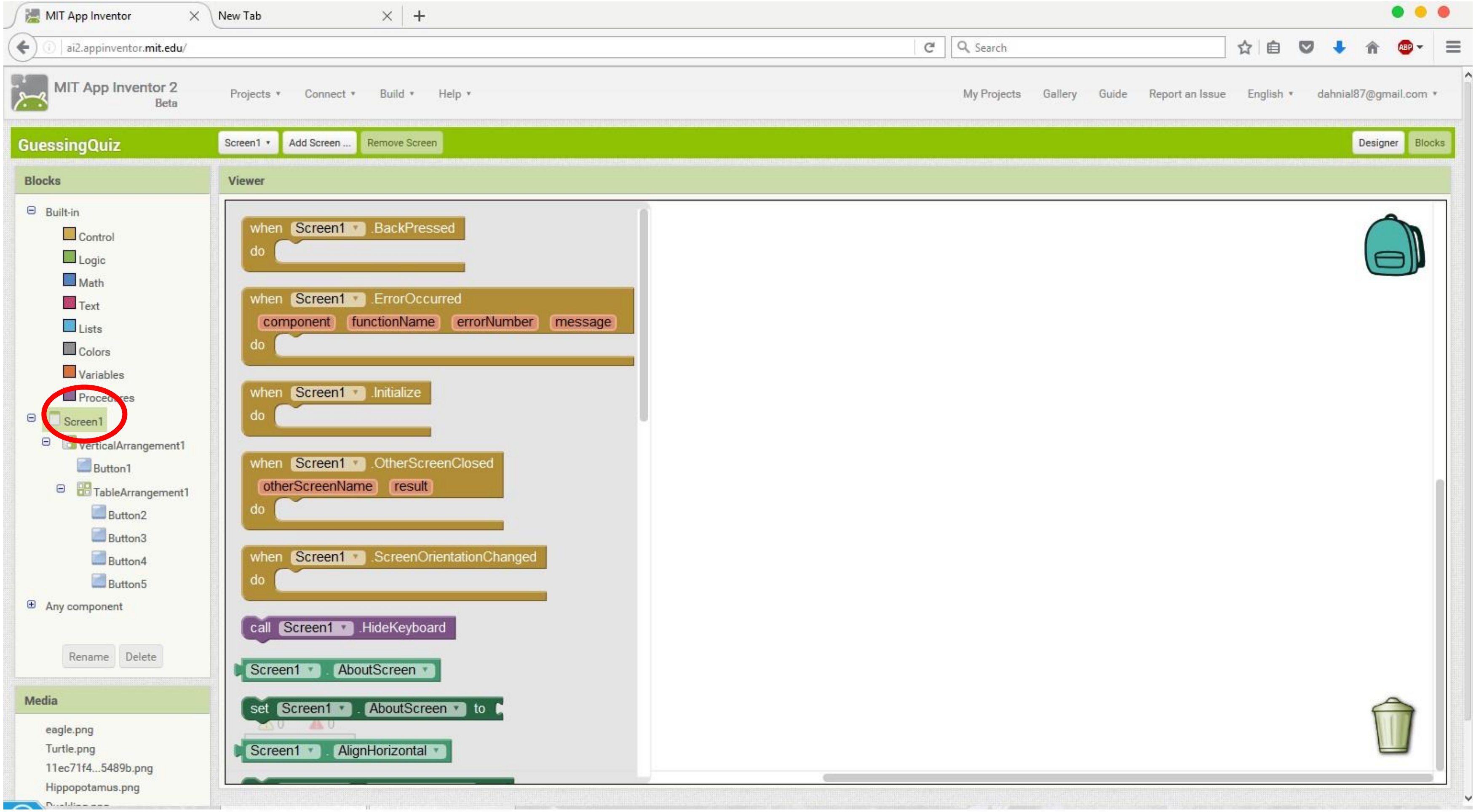


Block from a Button in UI screen

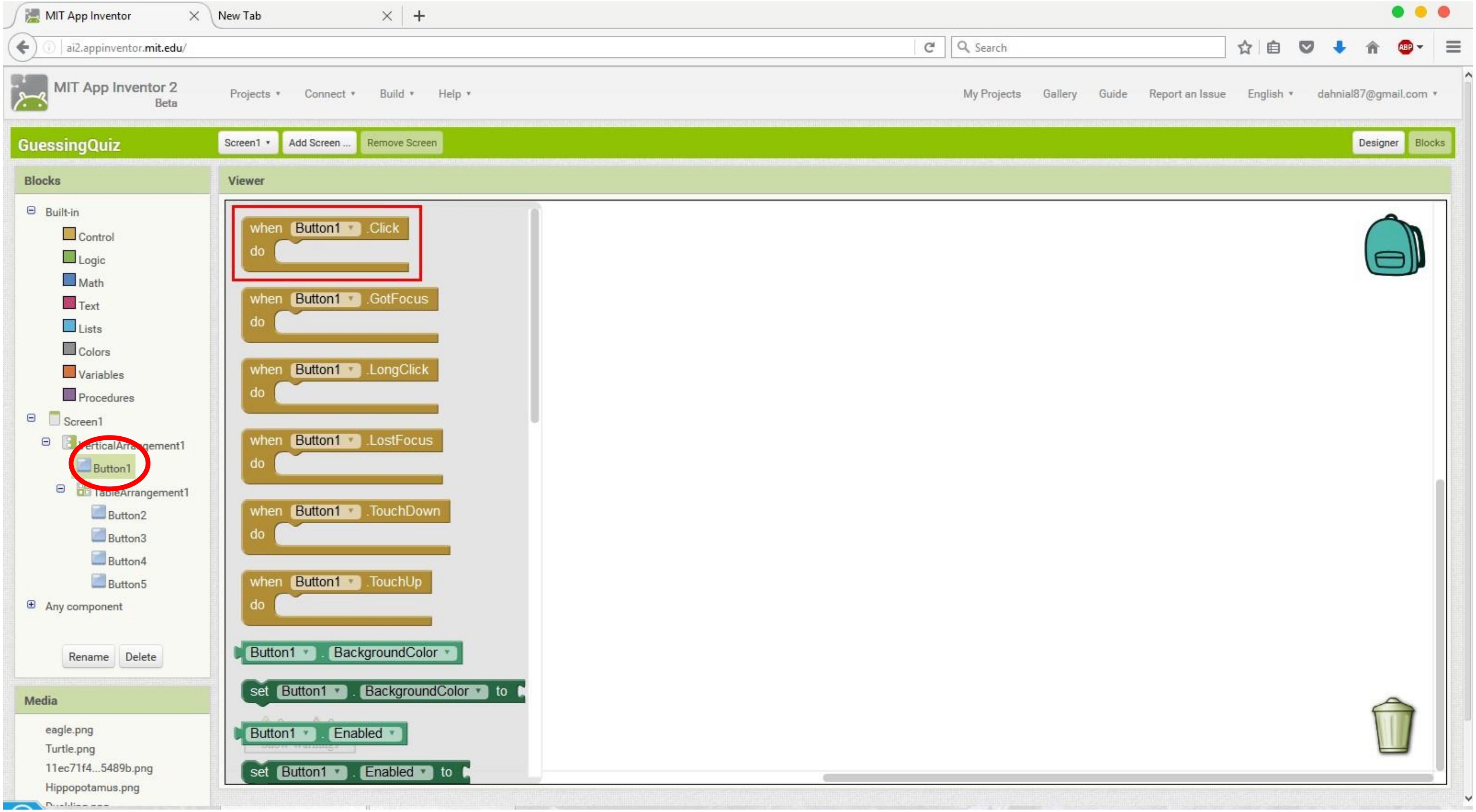


```
when Button1 ▾ .Click
do

when Button1 ▾ .GotFocus
do

when Button1 ▾ .LongClick
do

when Button1 ▾ .LostFocus
do

when Button1 ▾ .TouchDown
do

when Button1 ▾ .TouchUp
do

Button1 ▾ . BackgroundColor ▾

set Button1 ▾ . BackgroundColor ▾ to

Button1 ▾ . Enabled ▾

set Button1 ▾ . Enabled ▾ to
```

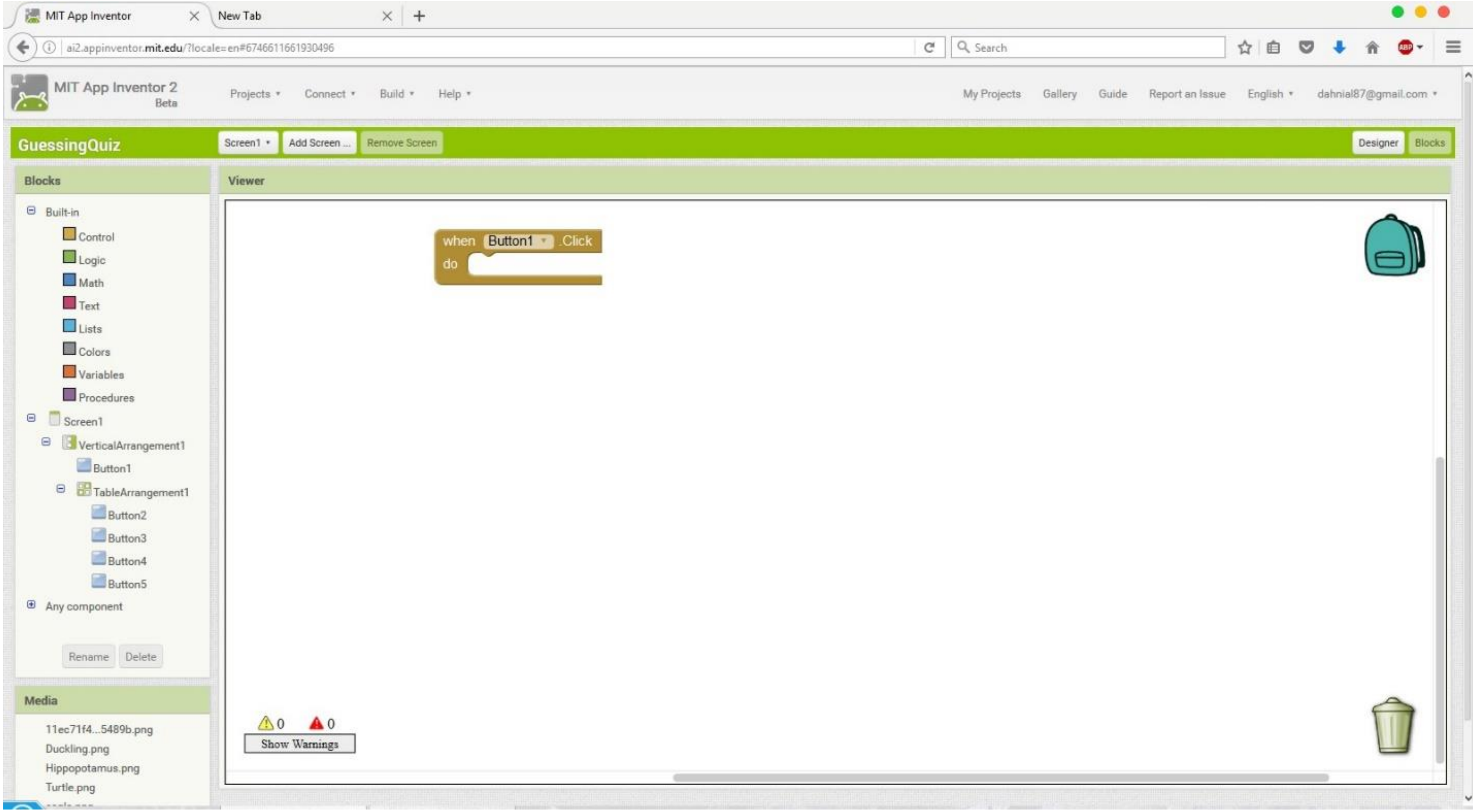
Block from a screen



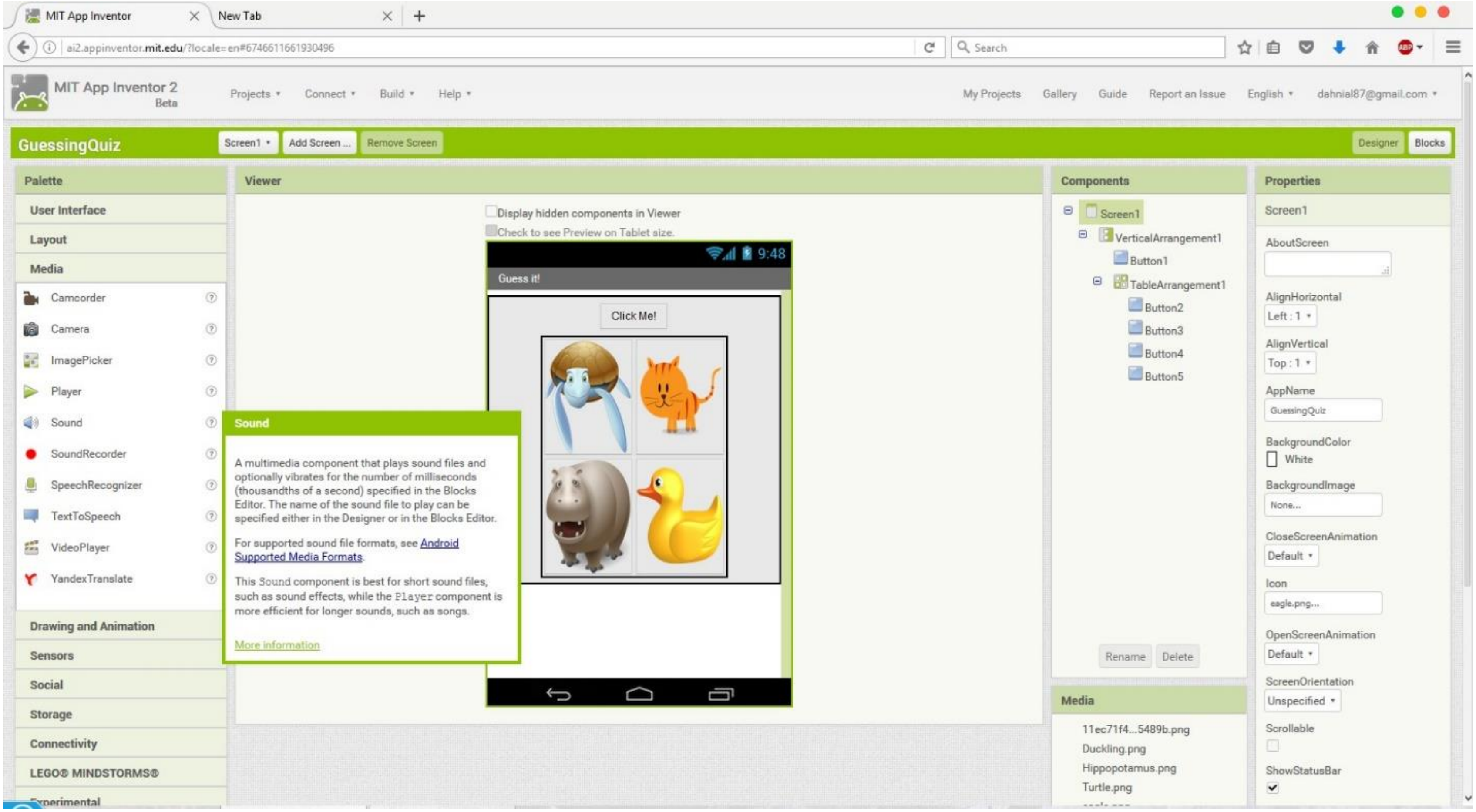
For example, a function when button1 is clicked:



Drag and drop to the screen



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



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

Components:

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

Media:

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

Properties:

Screen1

AboutScreen

AlignHorizontal: Left : 1

AlignVertical: Top : 1

AppName: GuessingQuiz

BackgroundColor: White

BackgroundImage: None...

CloseScreenAnimation: Default

Icon: eagle.png...

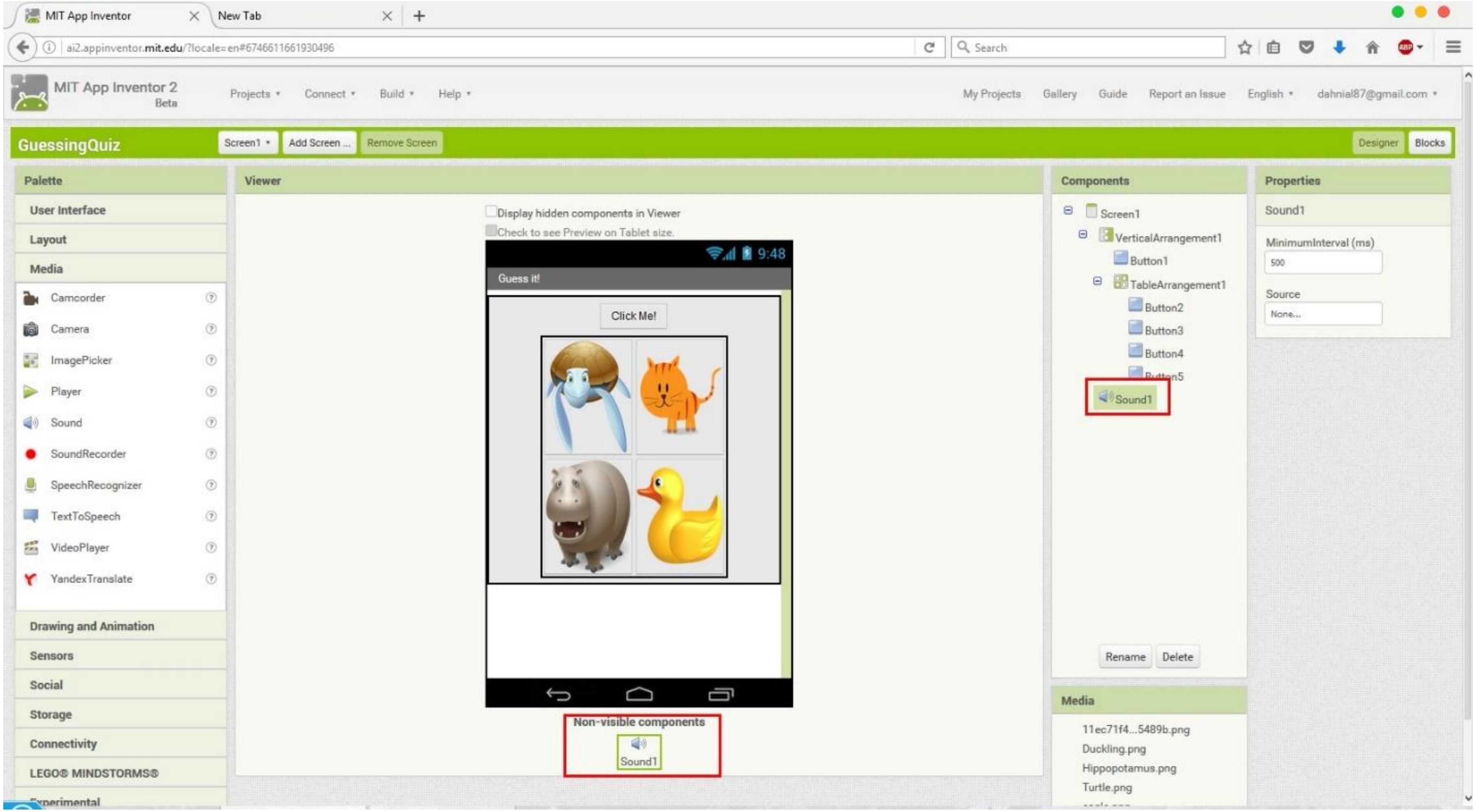
OpenScreenAnimation: Default

ScreenOrientation: Unspecified

Scrollable: □

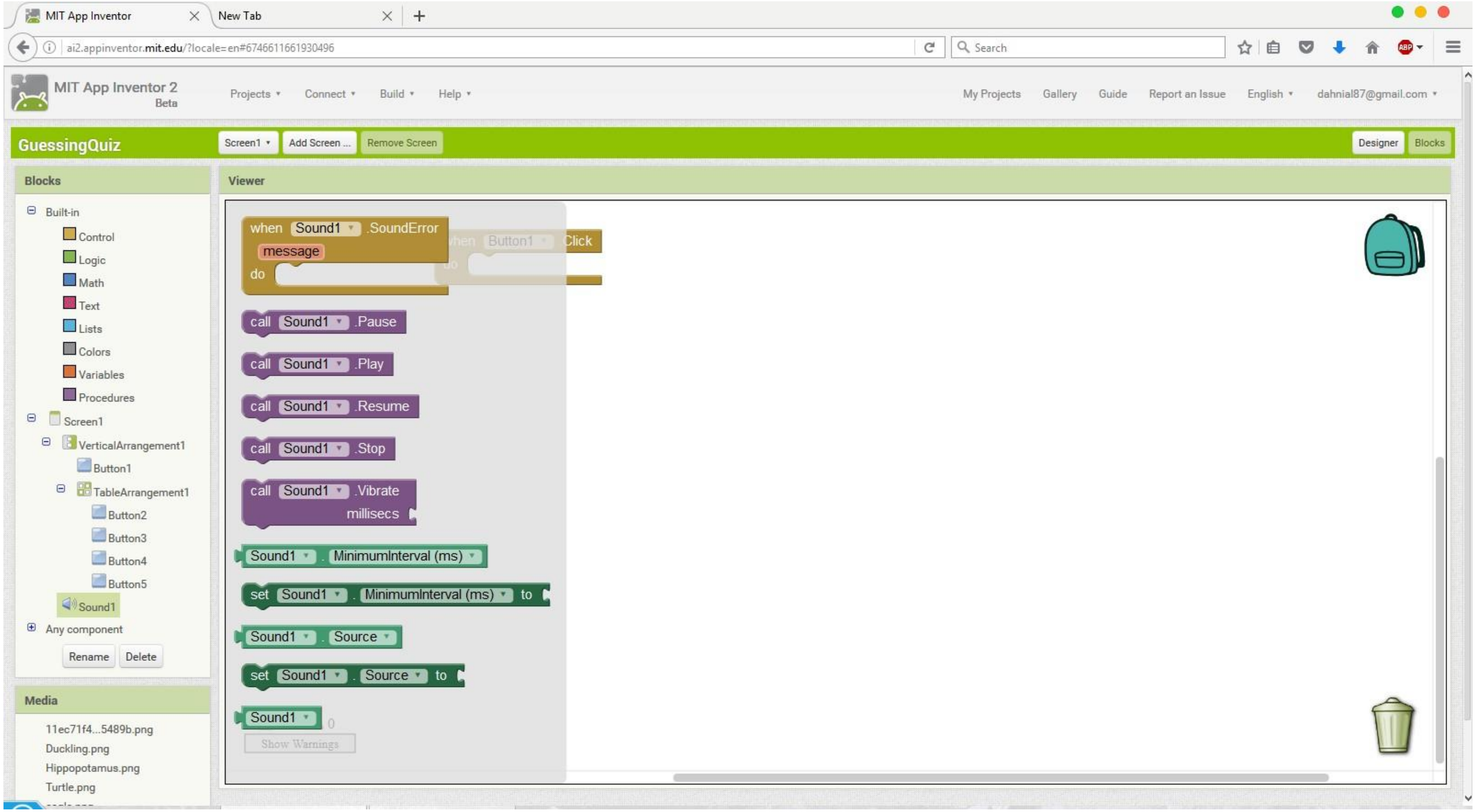
ShowStatusBar: ☑

Drag and drop to the screen



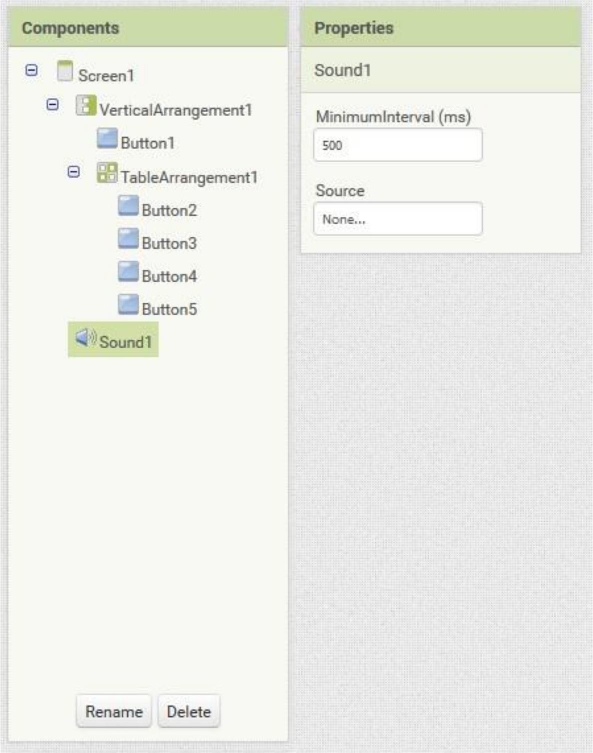
MinimumInterval (ms): 500

Source: None...



```
when Sound1 .SoundError
  message
do

call Sound1 .Pause
call Sound1 .Play
call Sound1 .Resume
call Sound1 .Stop
call Sound1 .Vibrate
                millisecs
Sound1 . MinimumInterval (ms)
set Sound1 . MinimumInterval (ms) to
Sound1 . Source
set Sound1 . Source to
Sound1
```

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



Components

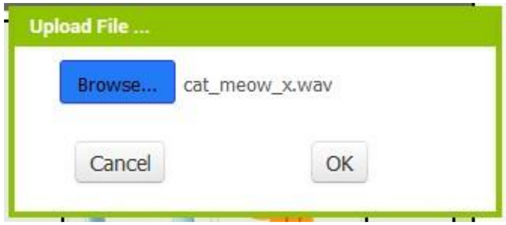
- Screen1
  - VerticalArrangement1
    - Button1
    - TableArrangement1
      - Button2
      - Button3
      - Button4
      - Button5
  - Sound1

Properties

Sound1

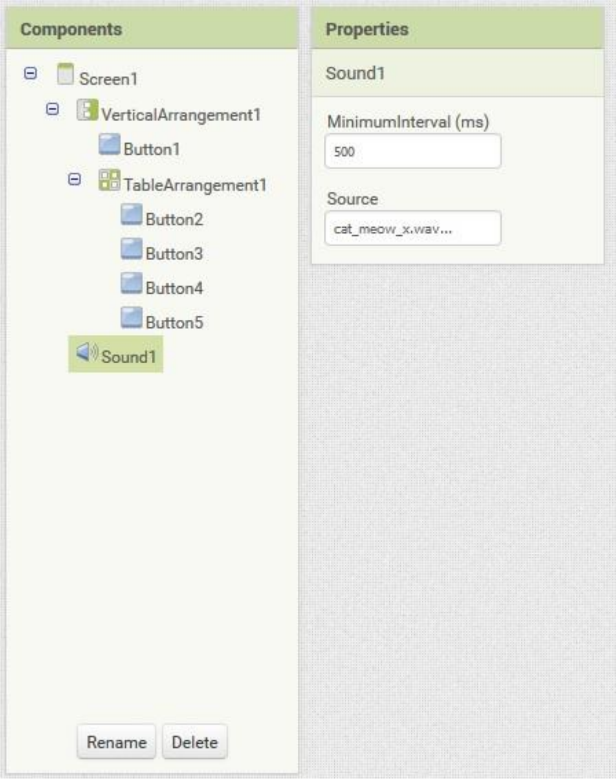
MinimumInterval (ms): 500

Source: None...



Upload File ...

cat_meow_x.wav



Components

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

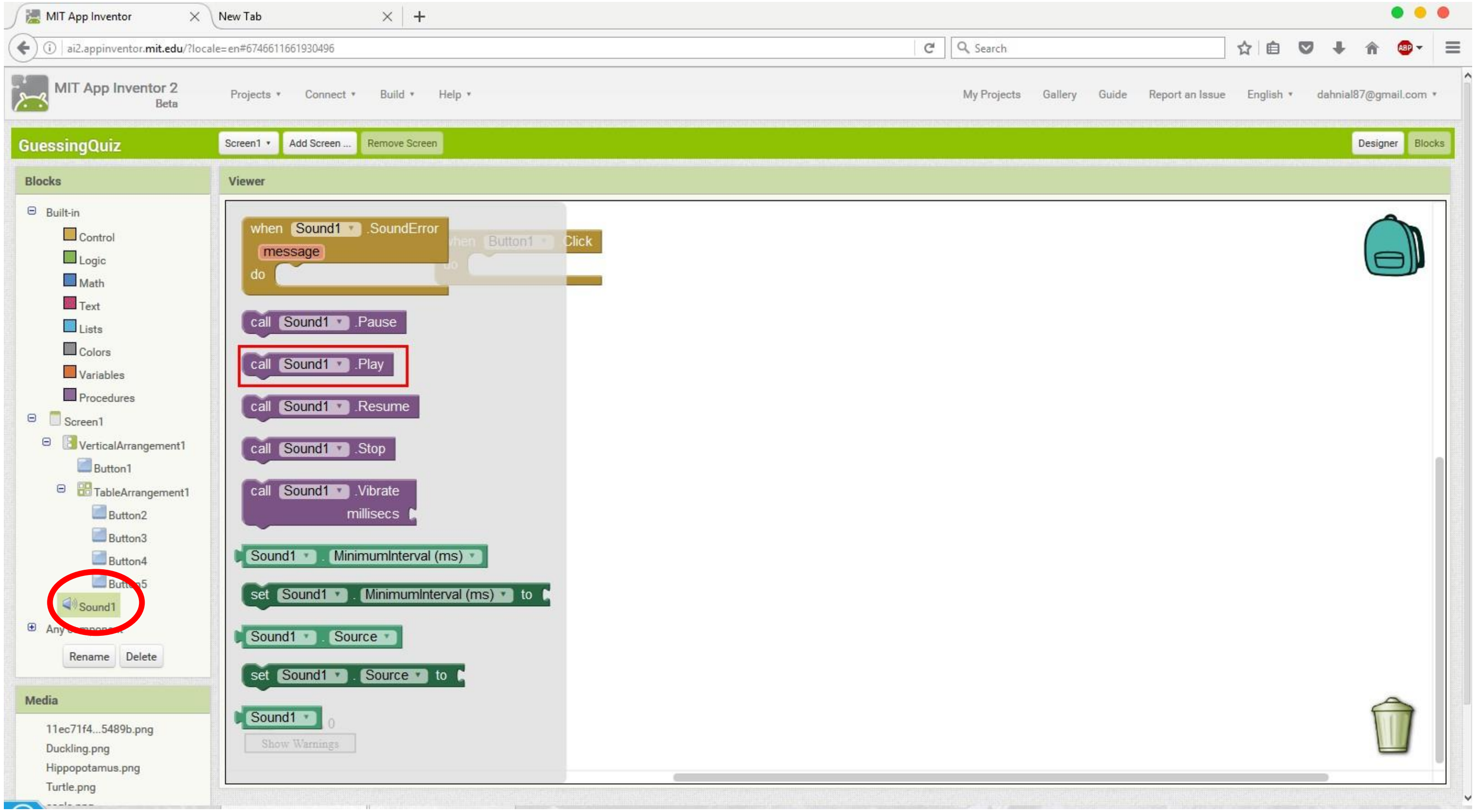
Properties

Sound1

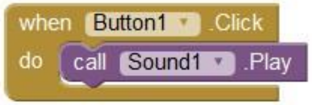
MinimumInterval (ms): 500

Source: cat_meow_x.wav...

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

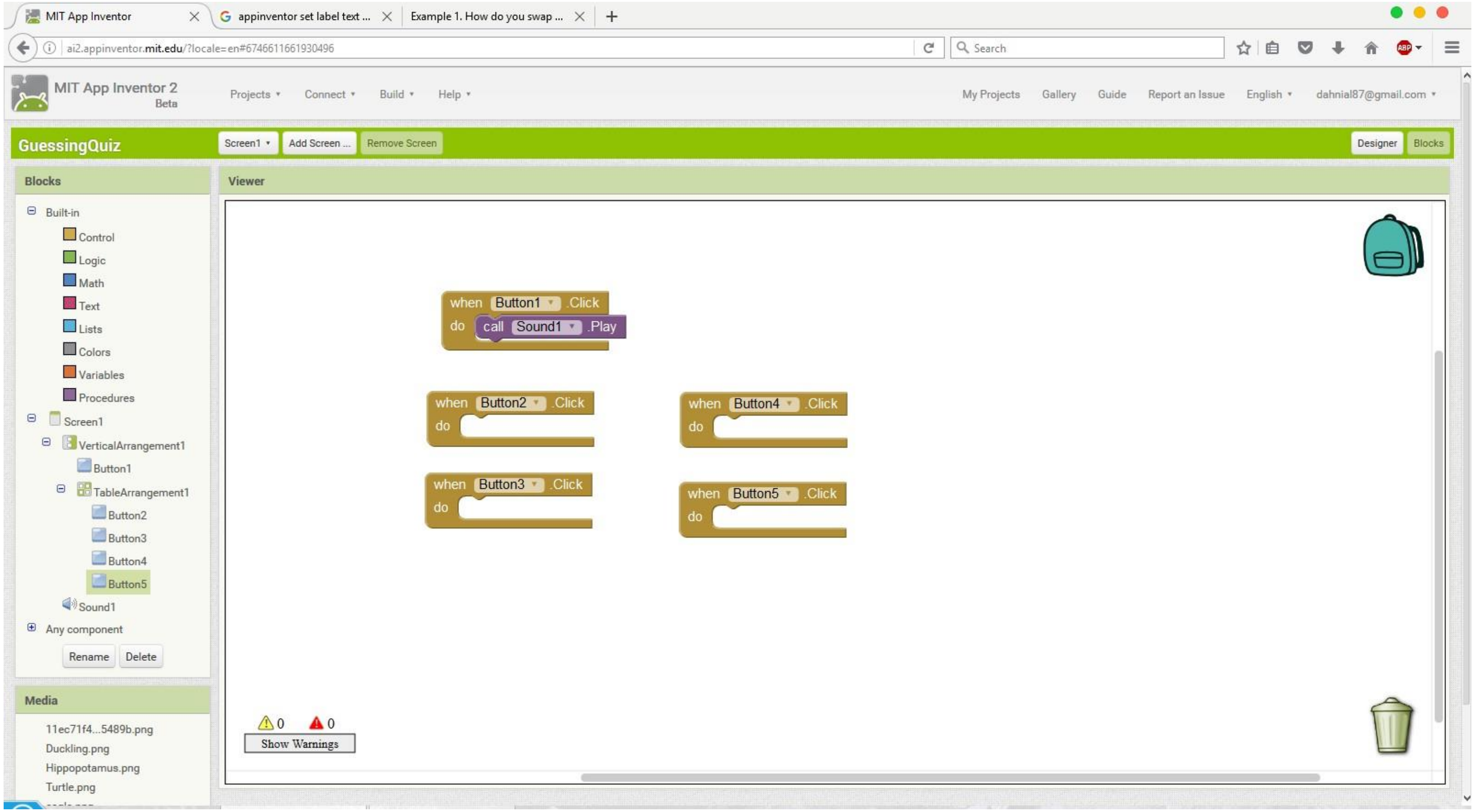


Result:



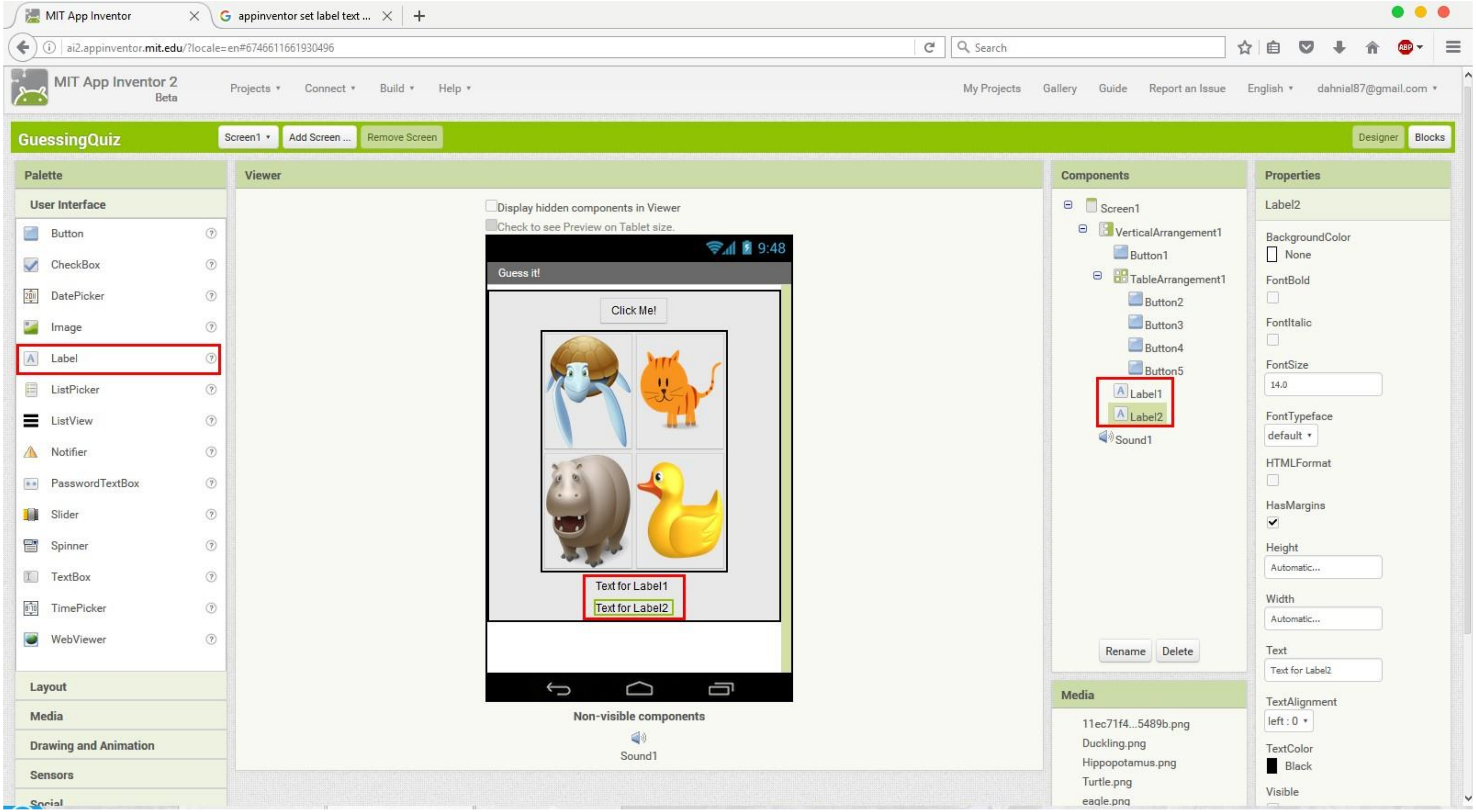
```
when Button1 .Click
do  call Sound1 .Play
```

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



```
when Button1 .Click
do  call Sound1 .Play

when Button2 .Click
do

when Button4 .Click
do

when Button3 .Click
do

when Button5 .Click
do
```

Go to designer ui, add text label for the score



Label2

BackgroundColor
□ None

FontBold
☐

FontItalic
☐

FontSize
14.0

FontTypeface
default

HTMLFormat
☐

HasMargins
☑

Height
Automatic...

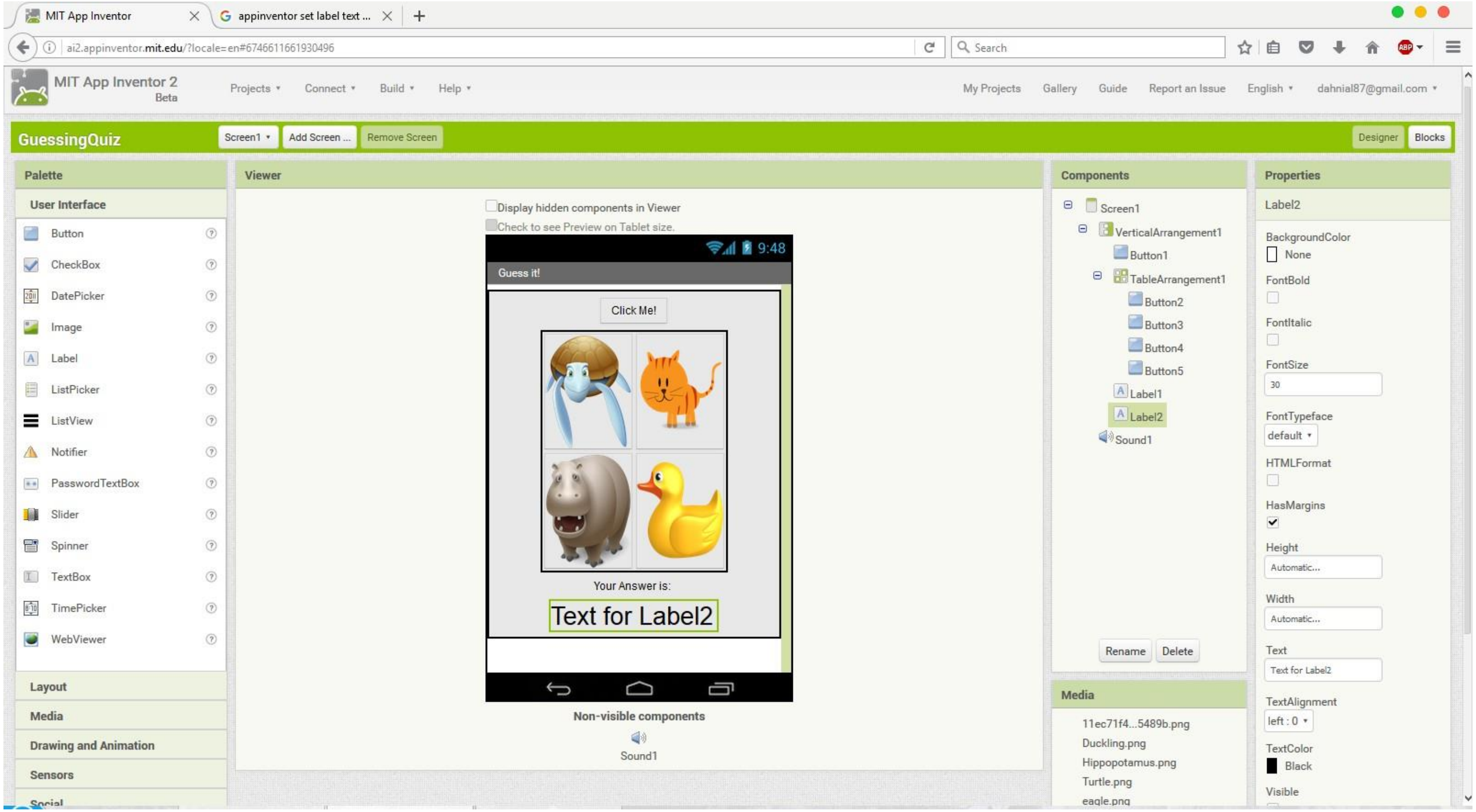
Width
Automatic...

Text
Text for Label2

TextAlignment
left : 0

TextColor
■ Black

Visible



□ Display hidden components in Viewer

□ Check to see Preview on Tablet size.

9:48

Guess it!

Click Me!

Your Answer is:

Text for Label2

Components:

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

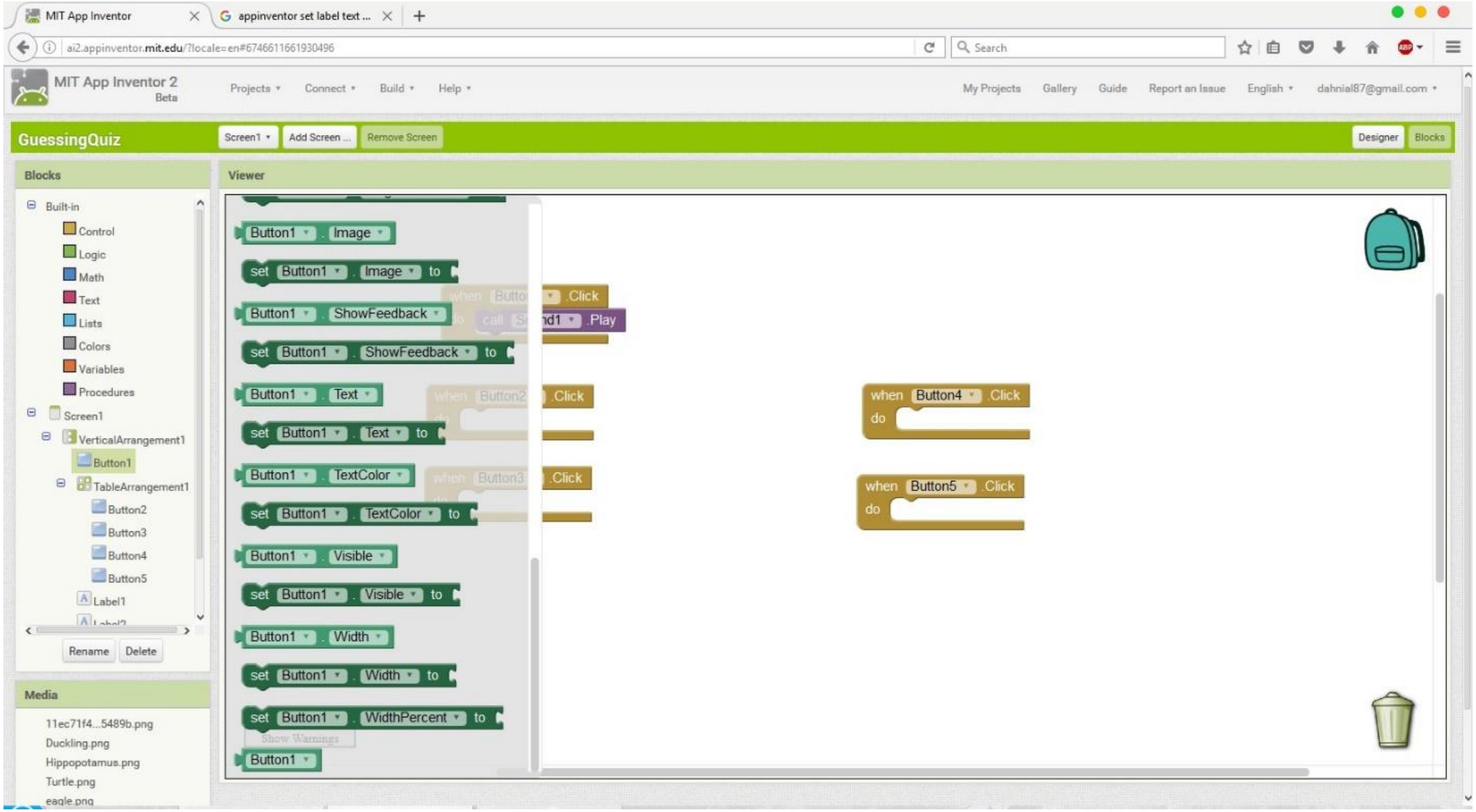
Media:

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

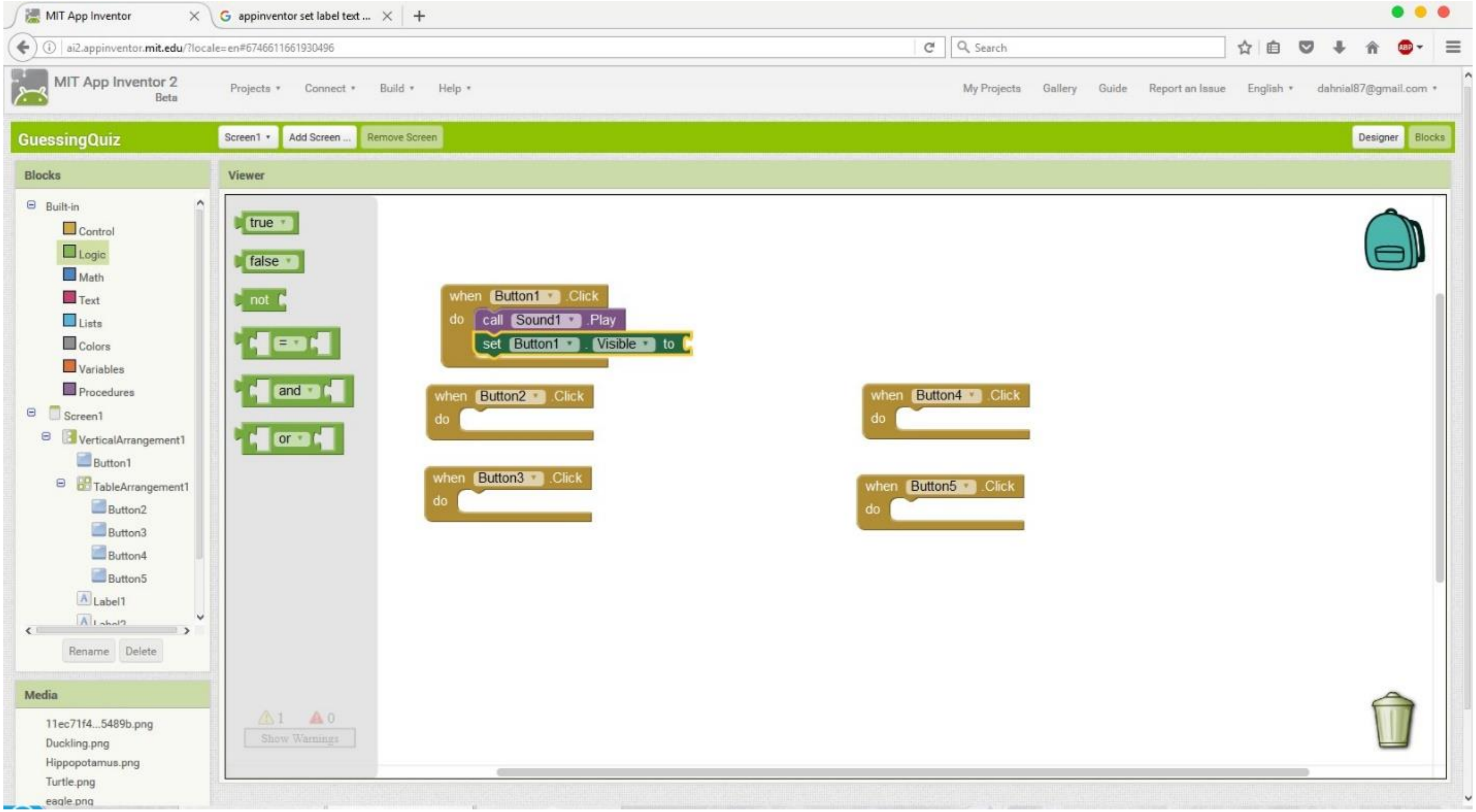
Properties: Label2

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

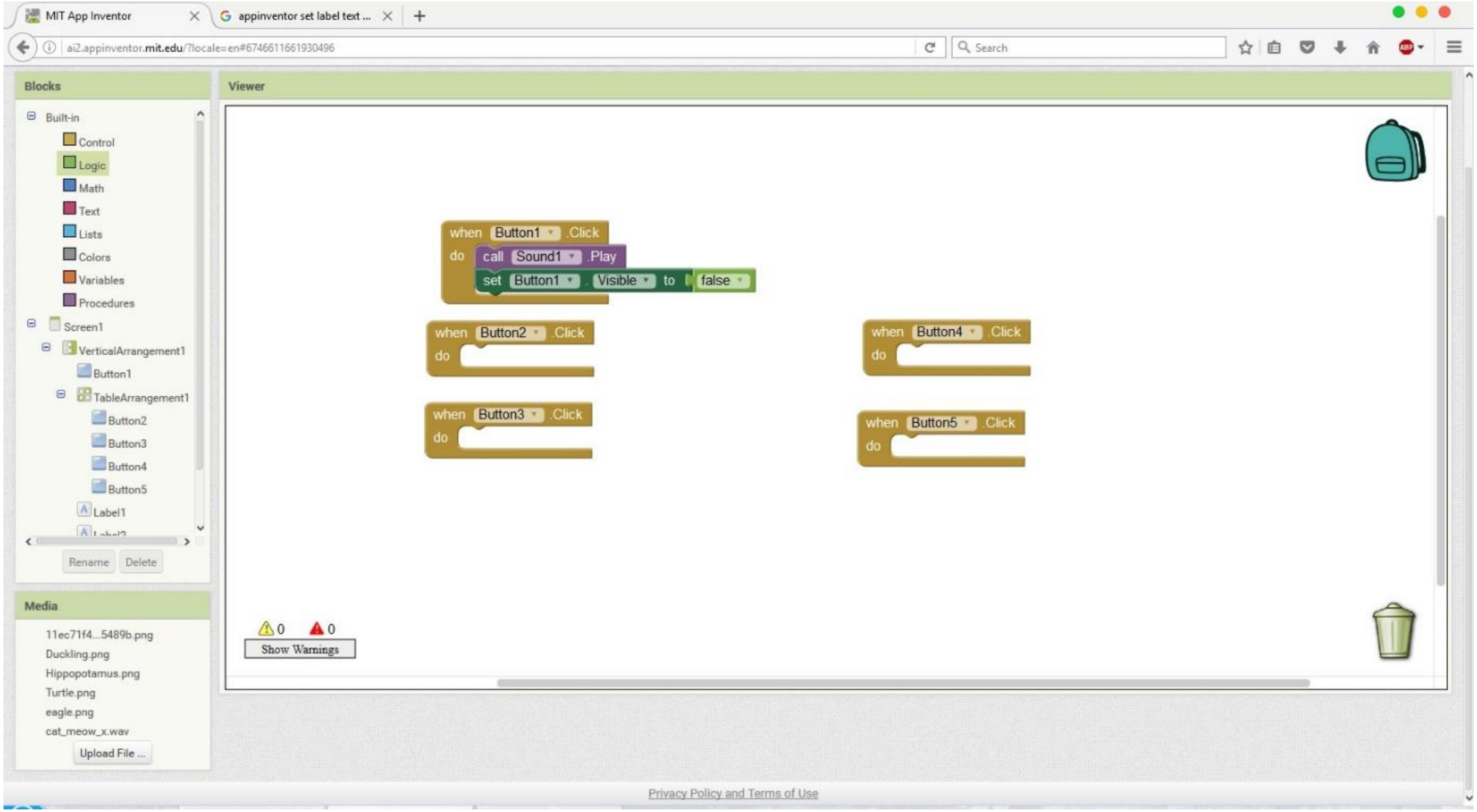
We want the player can only play sound once, so we will hide the button



Result:

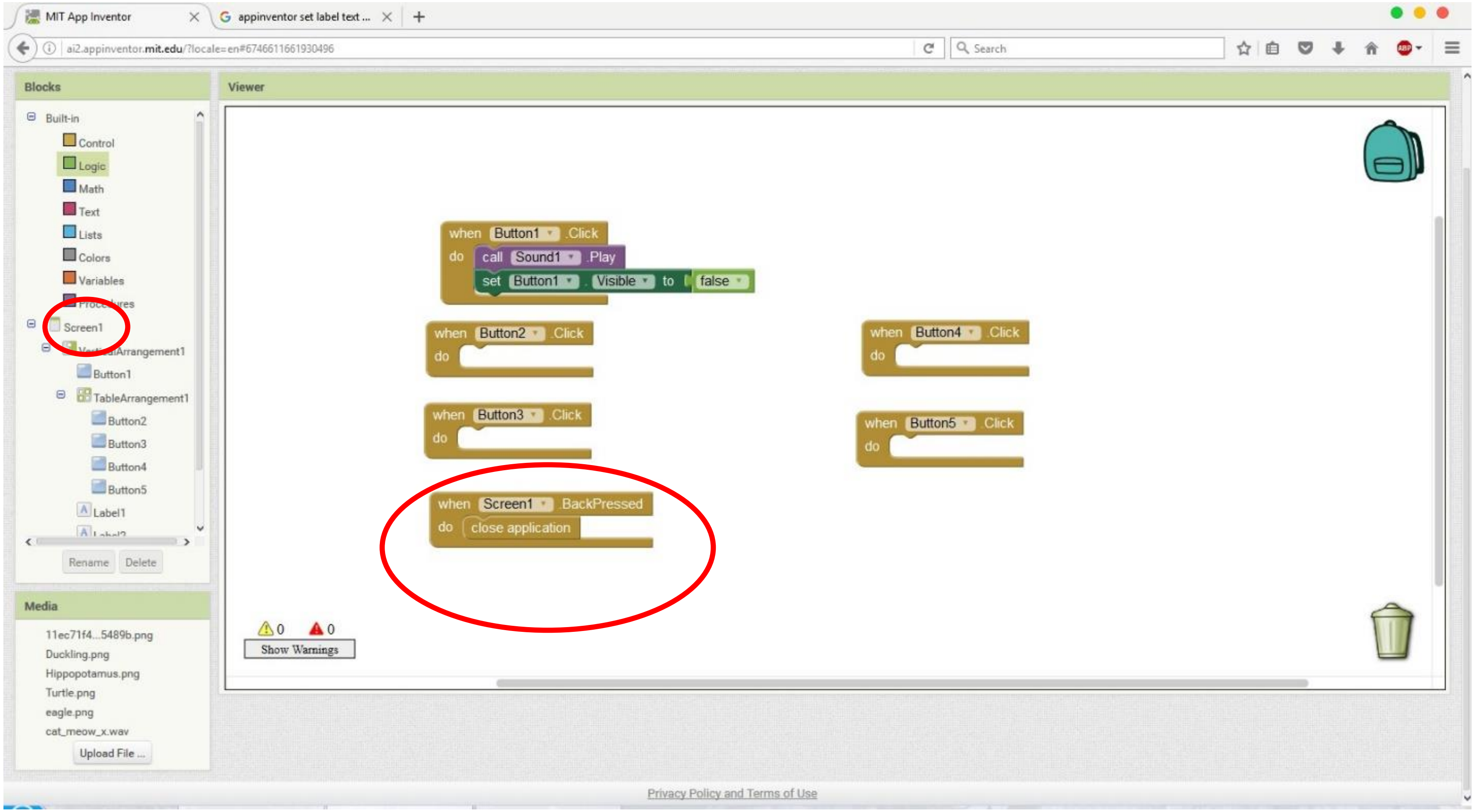


```
when Button1 .Click
do  call Sound1 .Play
    set Button1 . Visible to

when Button2 .Click
do

when Button3 .Click
do

when Button4 .Click
do

when Button5 .Click
do
```



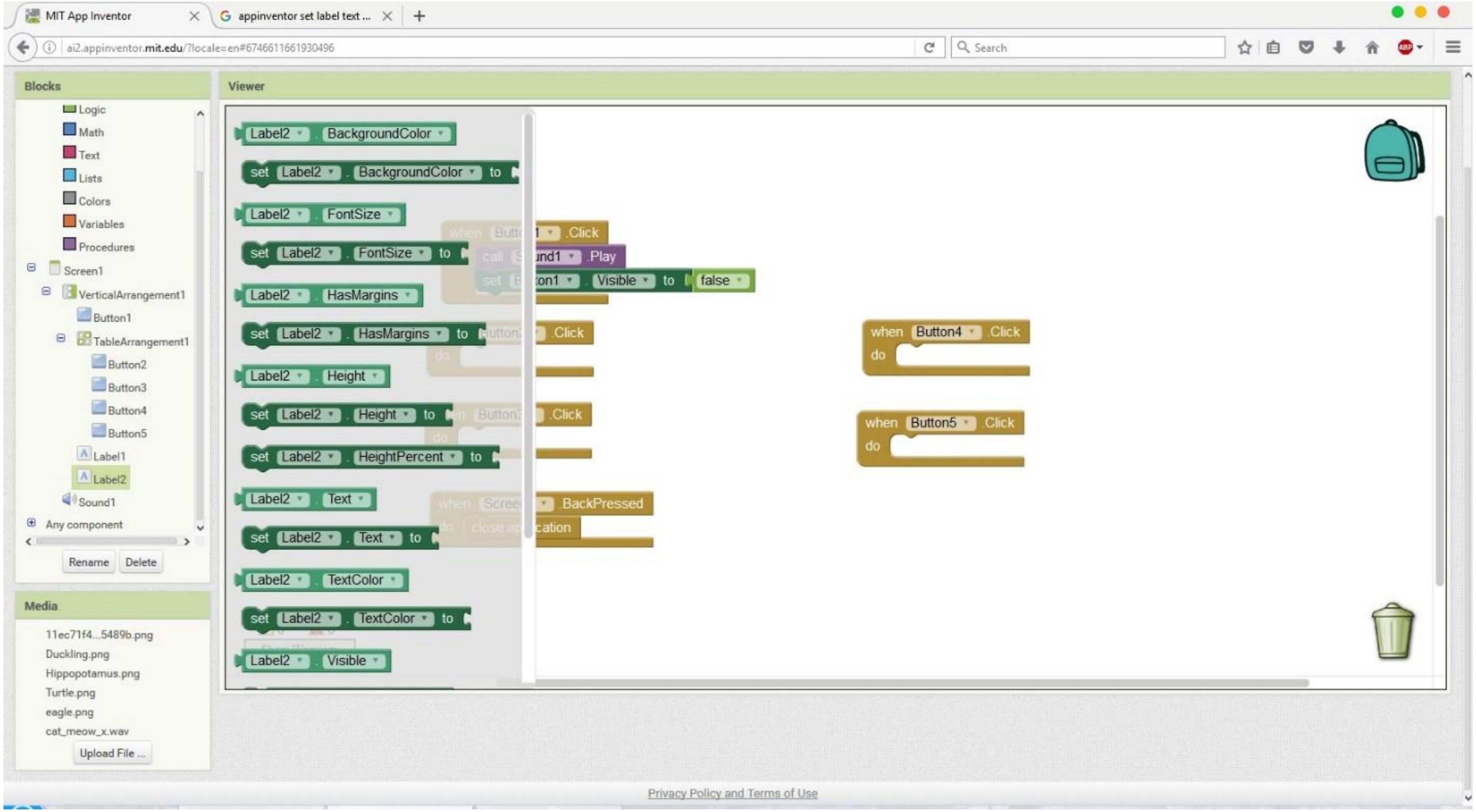
```
when Button1 .Click
do   call Sound1 .Play
     set Button1 . Visible to false

when Button2 .Click
do

when Button3 .Click
do

when Button4 .Click
do

when Button5 .Click
do
```

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



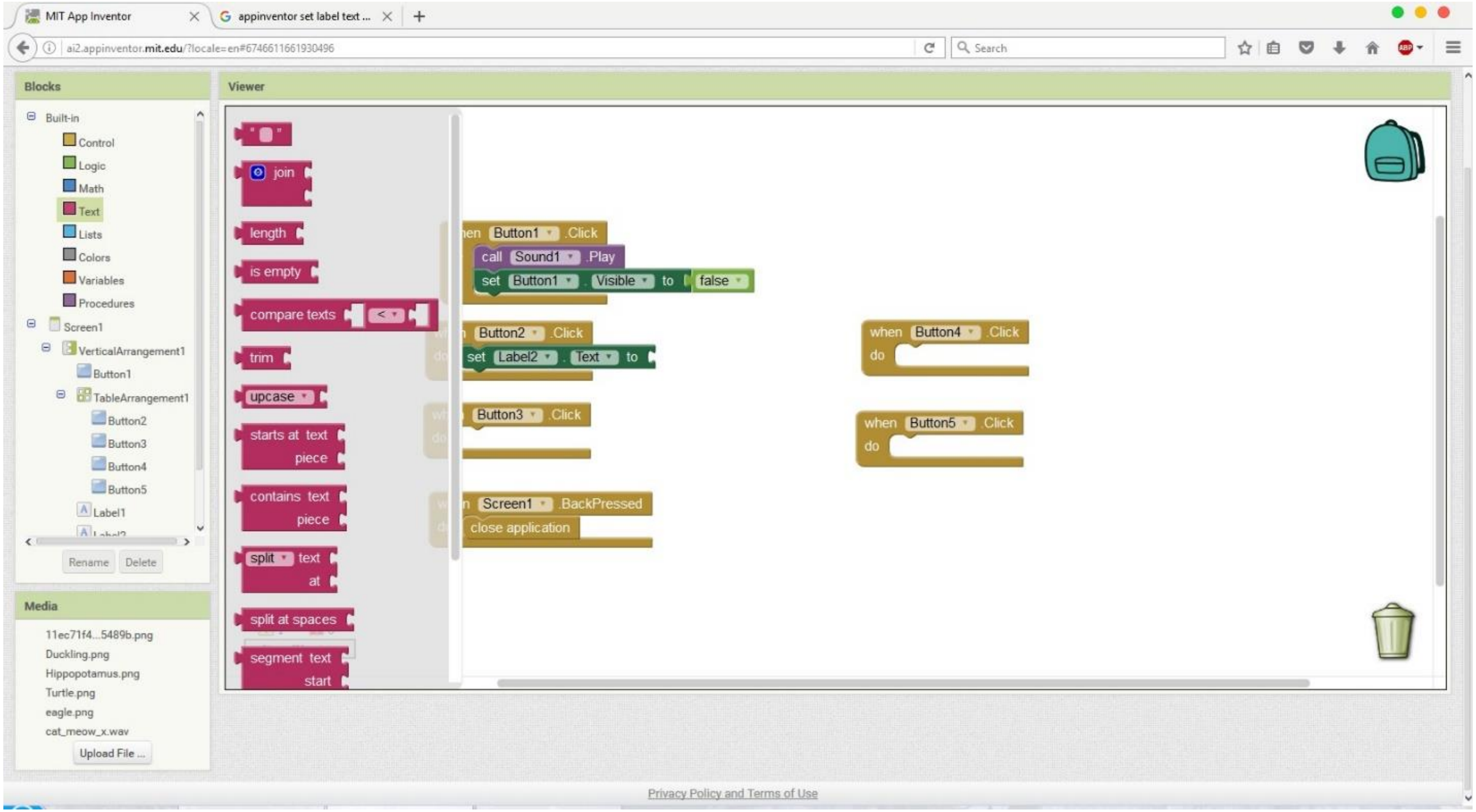
```
when Button1 .Click
do  call Sound1 .Play
    set Button1 . Visible to  false

when Button2 .Click
do

when Button4 .Click
do

when Button3 .Click
do

when Button5 .Click
do

when Screen1 .BackPressed
do  close application
```

Add label for the score:



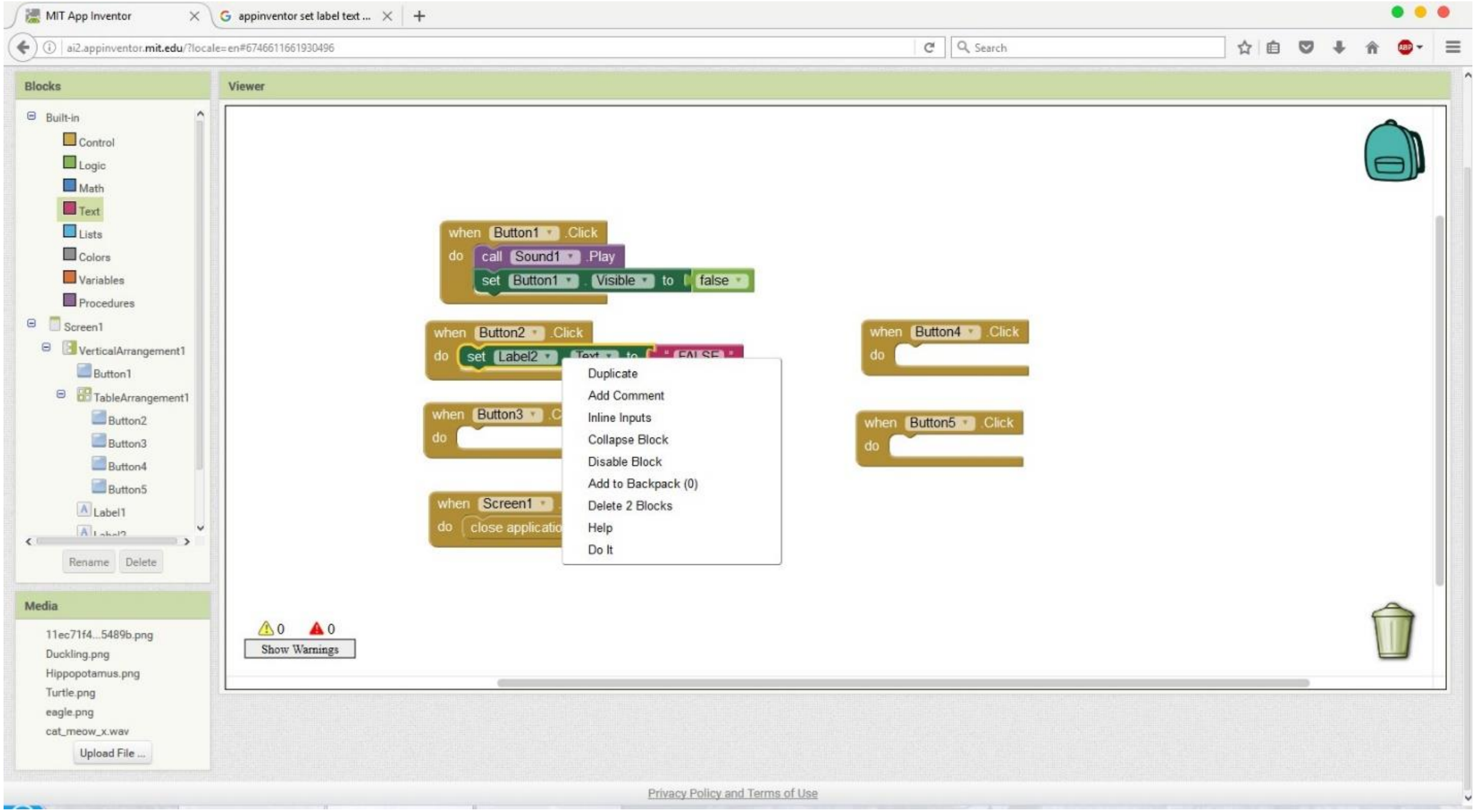
```
Label2 . BackgroundColor
set Label2 . BackgroundColor to
Label2 . FontSize
set Label2 . FontSize to
Label2 . HasMargins
set Label2 . HasMargins to
Label2 . Height
set Label2 . Height to
set Label2 . HeightPercent to
Label2 . Text
set Label2 . Text to
Label2 . TextColor
set Label2 . TextColor to
Label2 . Visible
```

```
when Butt 1 .Click
  call S und1 .Play
  set B ton1 . Visible to false

when Button4 .Click
do

when Button5 .Click
do

when Scree .BackPressed
do close ap cation
```

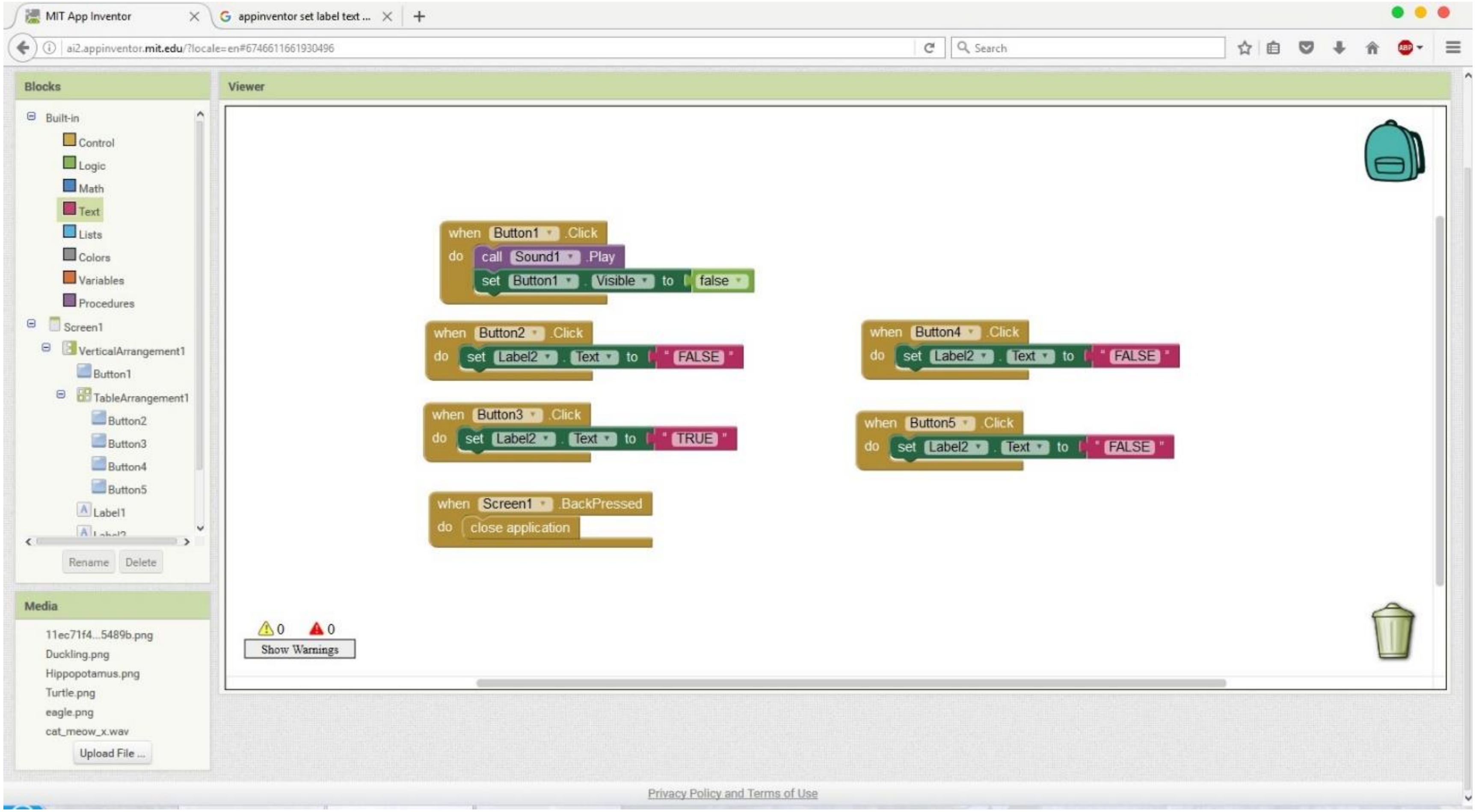


```
when Button1 .Click
do  call Sound1 .Play
    set Button1 . Visible to false

when Button2 .Click
do  set Label2 . Text to

when Button3 .Click
do

when Screen1 .BackPressed
do  close application

when Button4 .Click
do

when Button5 .Click
do
```

Duplicate, For faster process

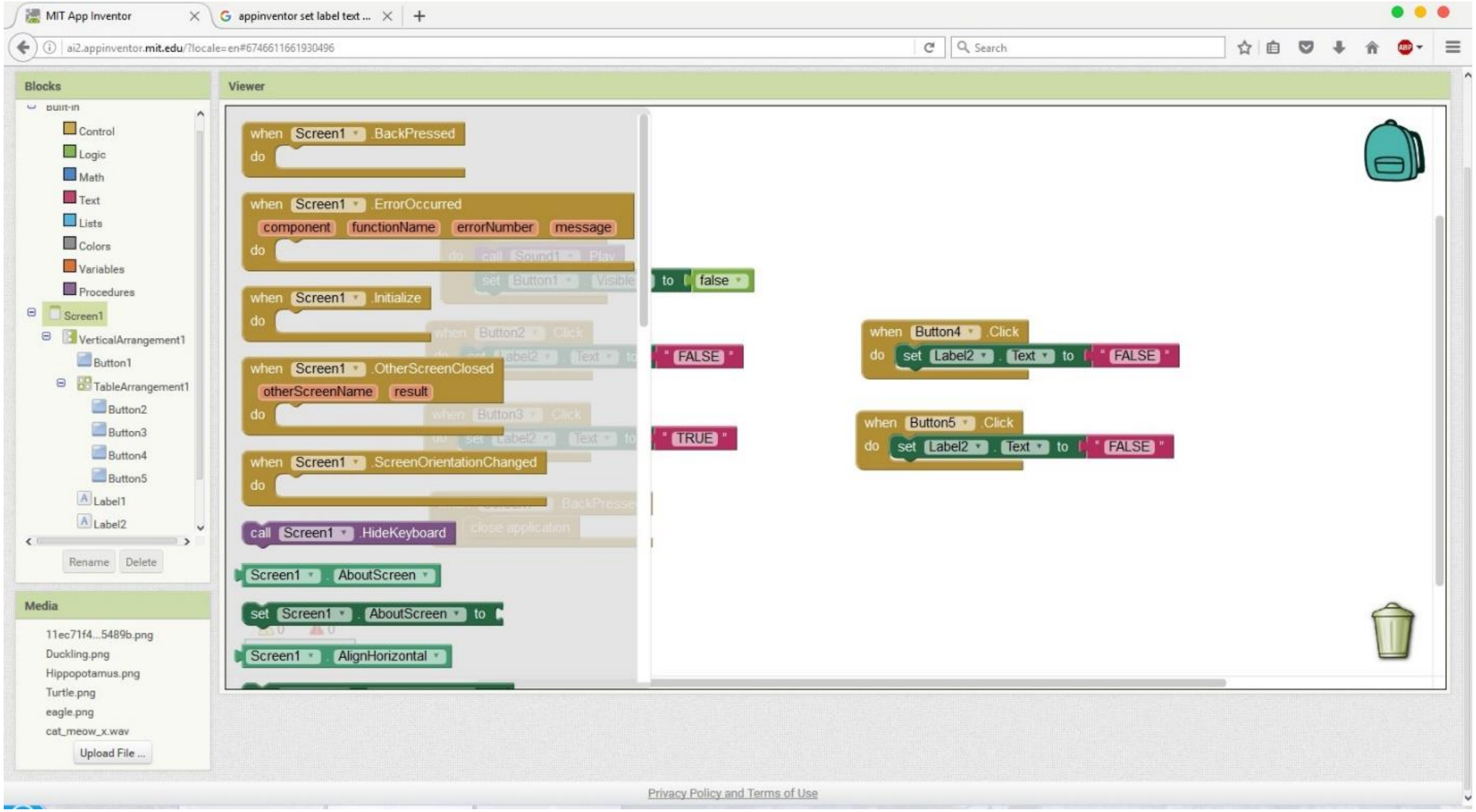


Result: TRUE only when button3 is clicked)

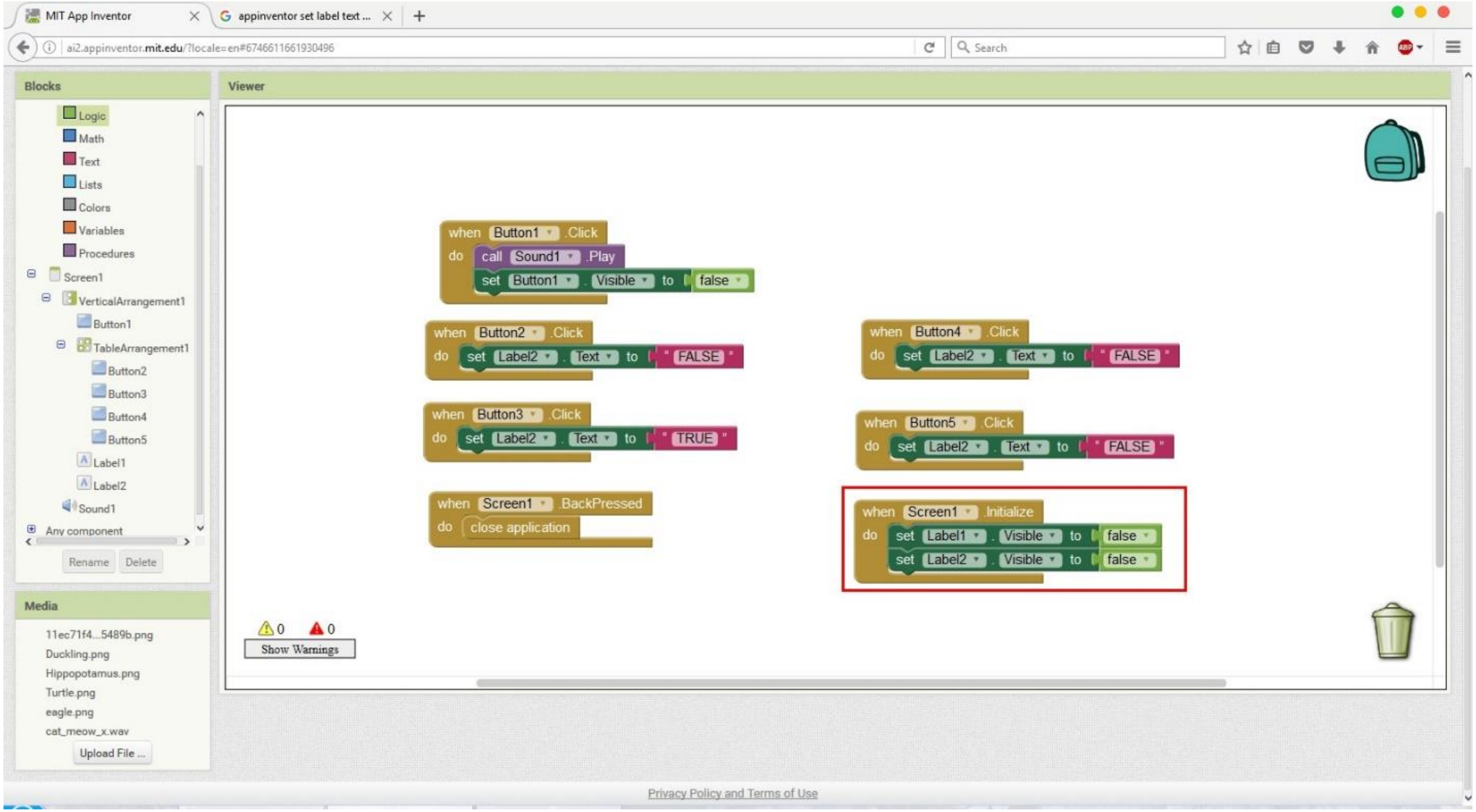


```
when Button1 .Click
do  call Sound1 .Play
    set Button1 . Visible to false

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
```

We need to hide the score when the app initializes

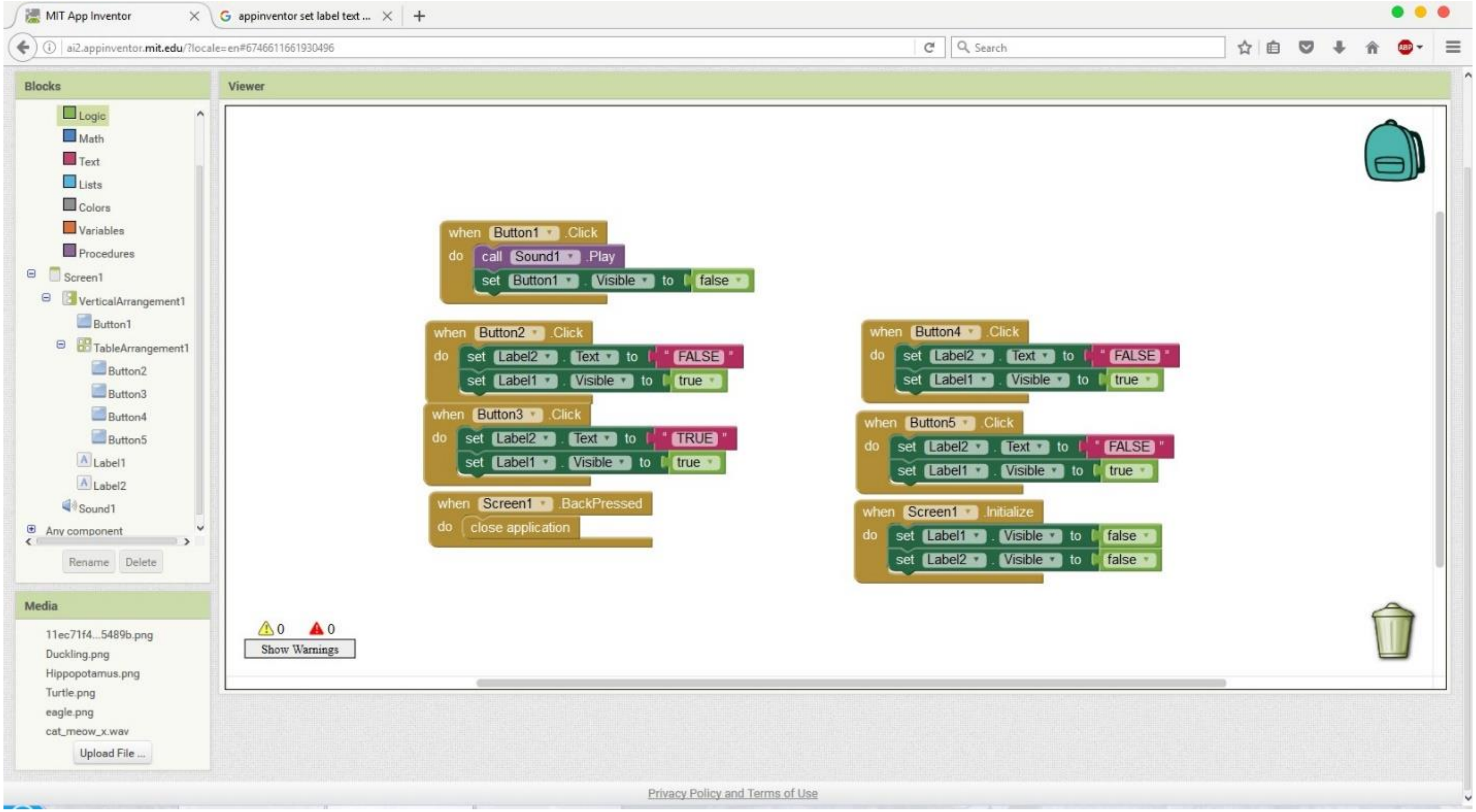


```
when Screen1 .BackPressed
do

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

to false

" FALSE "

" TRUE "

when Button4 .Click
do set Label2 . Text to " FALSE "

when Button5 .Click
do set Label2 . Text to " FALSE "
```



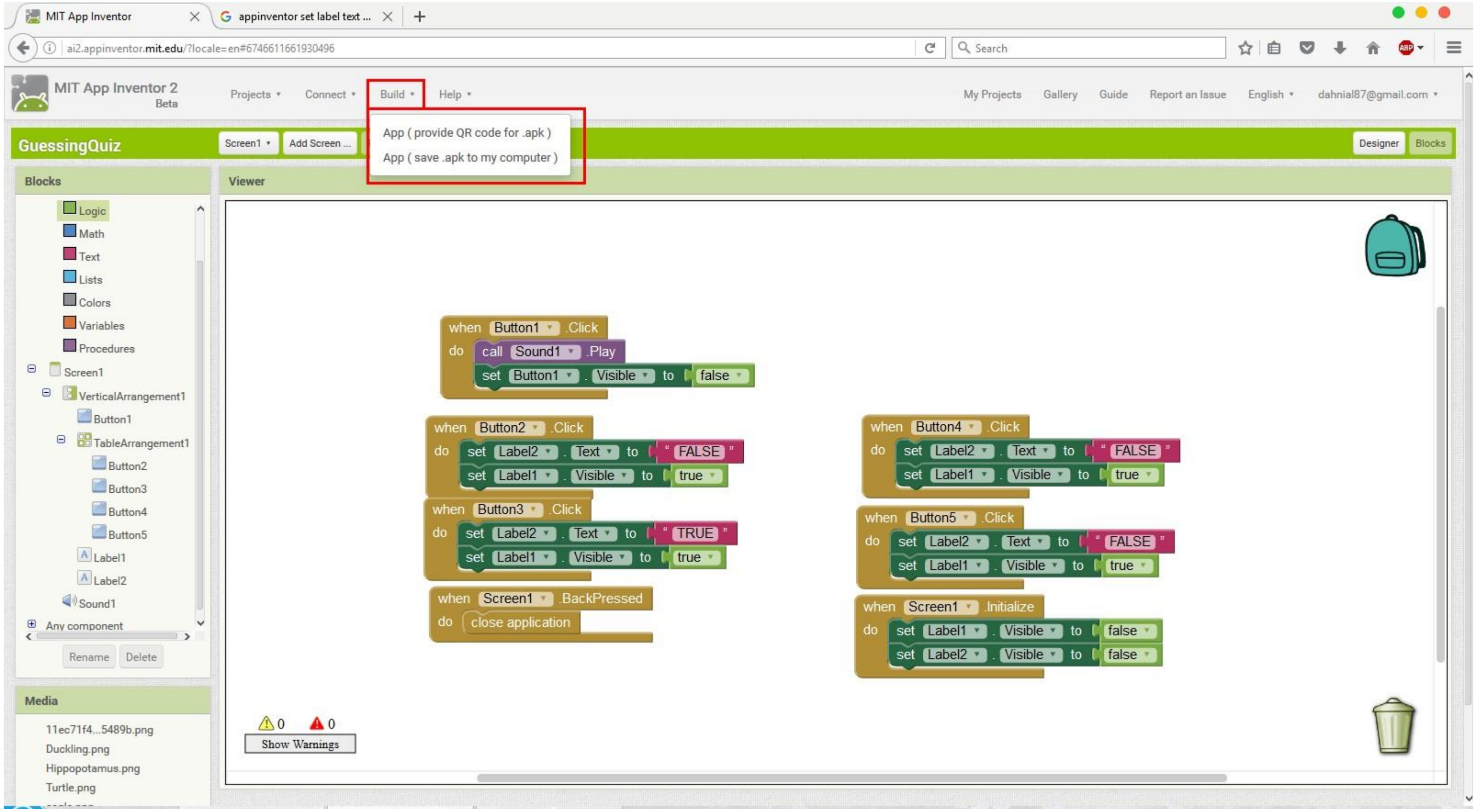
```
when Button1 .Click
do  call Sound1 .Play
    set Button1 . Visible to false

when Button2 .Click
do  set Label2 . Text to " FALSE "

when Button3 .Click
do  set Label2 . Text to " TRUE "

when Screen1 .BackPressed
do  close application

when Button4 .Click
do  set Label2 . Text to " FALSE "

when Button5 .Click
do  set Label2 . Text to " FALSE "

when Screen1 .Initialize
do  set Label1 . Visible to false
    set Label2 . Visible to false
```

But we will show the result when the button is pressed

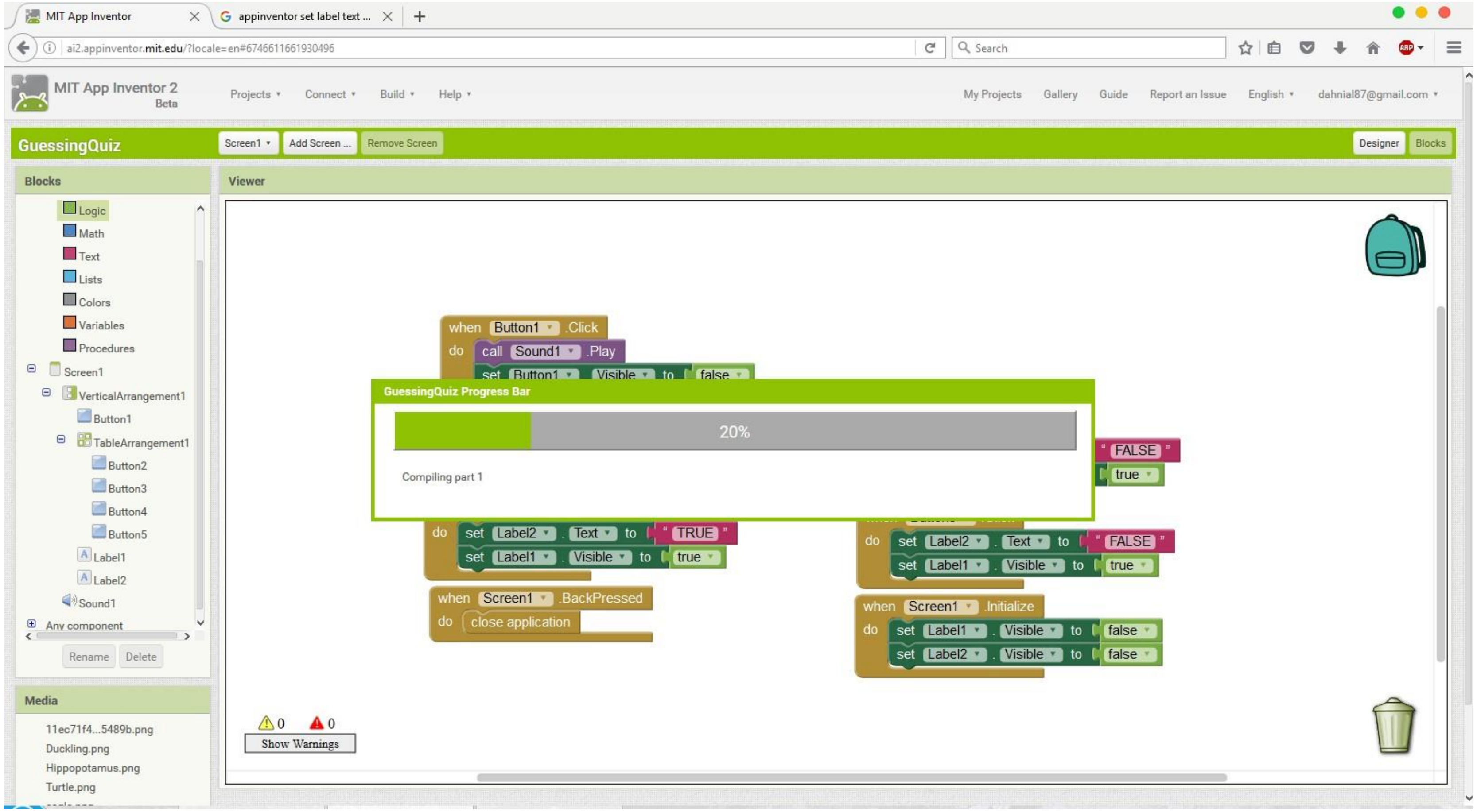


```
when Button1 .Click
do  call Sound1 .Play
    set Button1 . Visible to false

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
```

Compiling the apps



```
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
```



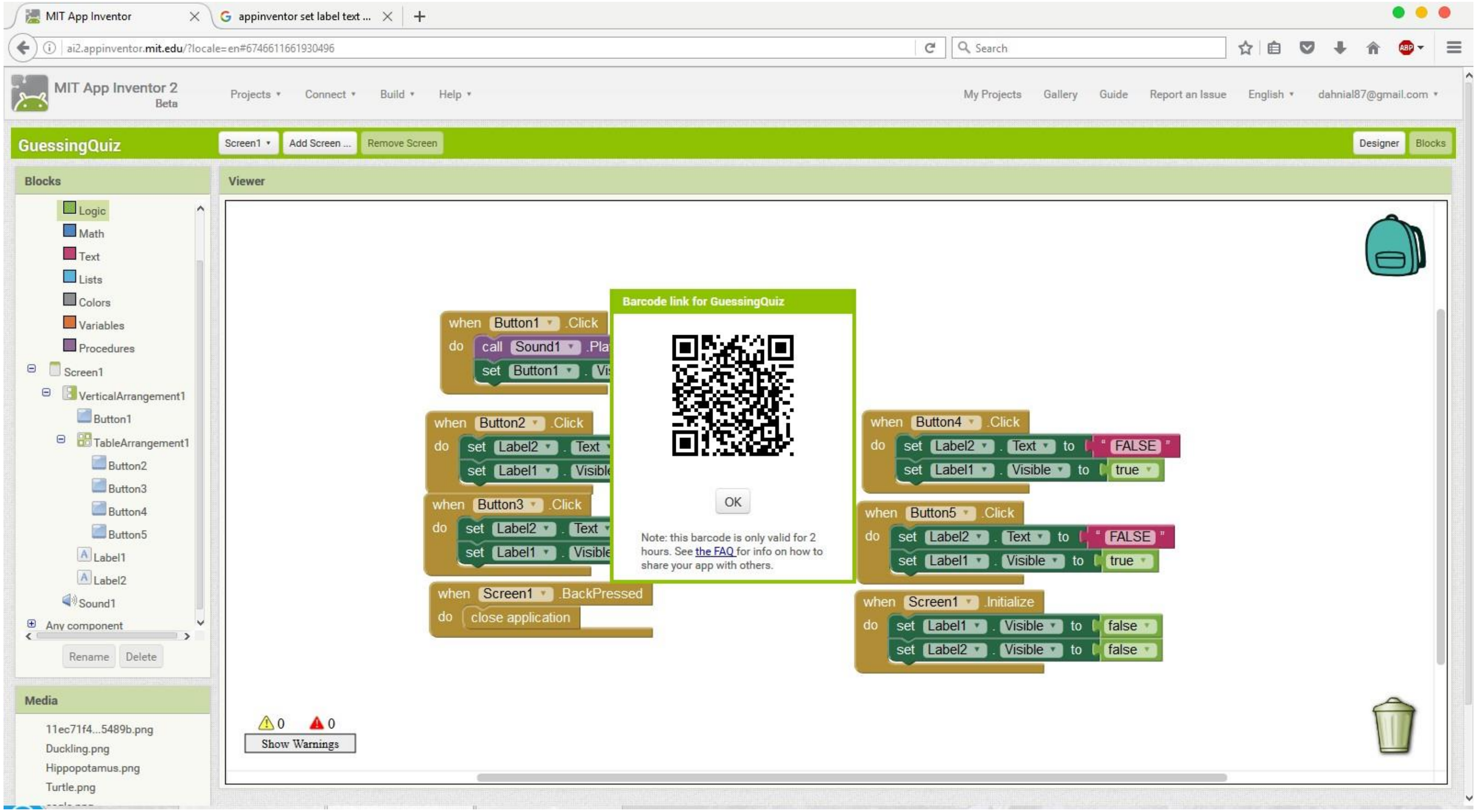
```
when Button1 .Click
do  call Sound1 .Play
    set Button1 . Visible to false

" FALSE "
true

do  set Label2 . Text to " TRUE "
    set Label1 . Visible to true

do  set Label2 . Text to " FALSE "
    set Label1 . Visible to true

when Screen1 .BackPressed
do  close application

when Screen1 .Initialize
do  set Label1 . Visible to false
    set Label2 . Visible to false
```

GuessingQuiz Progress Bar

20%

Compiling part 1

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

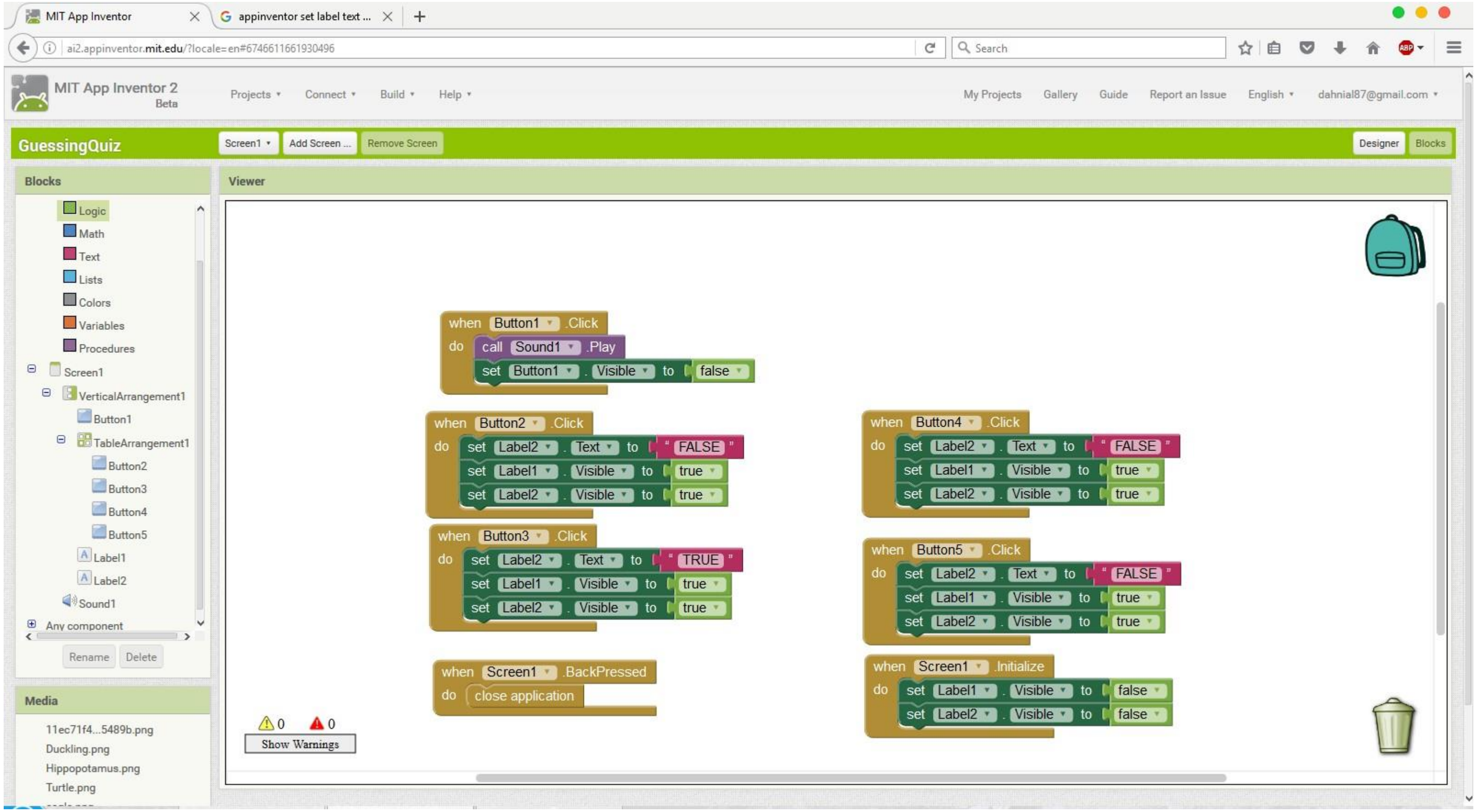


Barcode link for GuessingQuiz

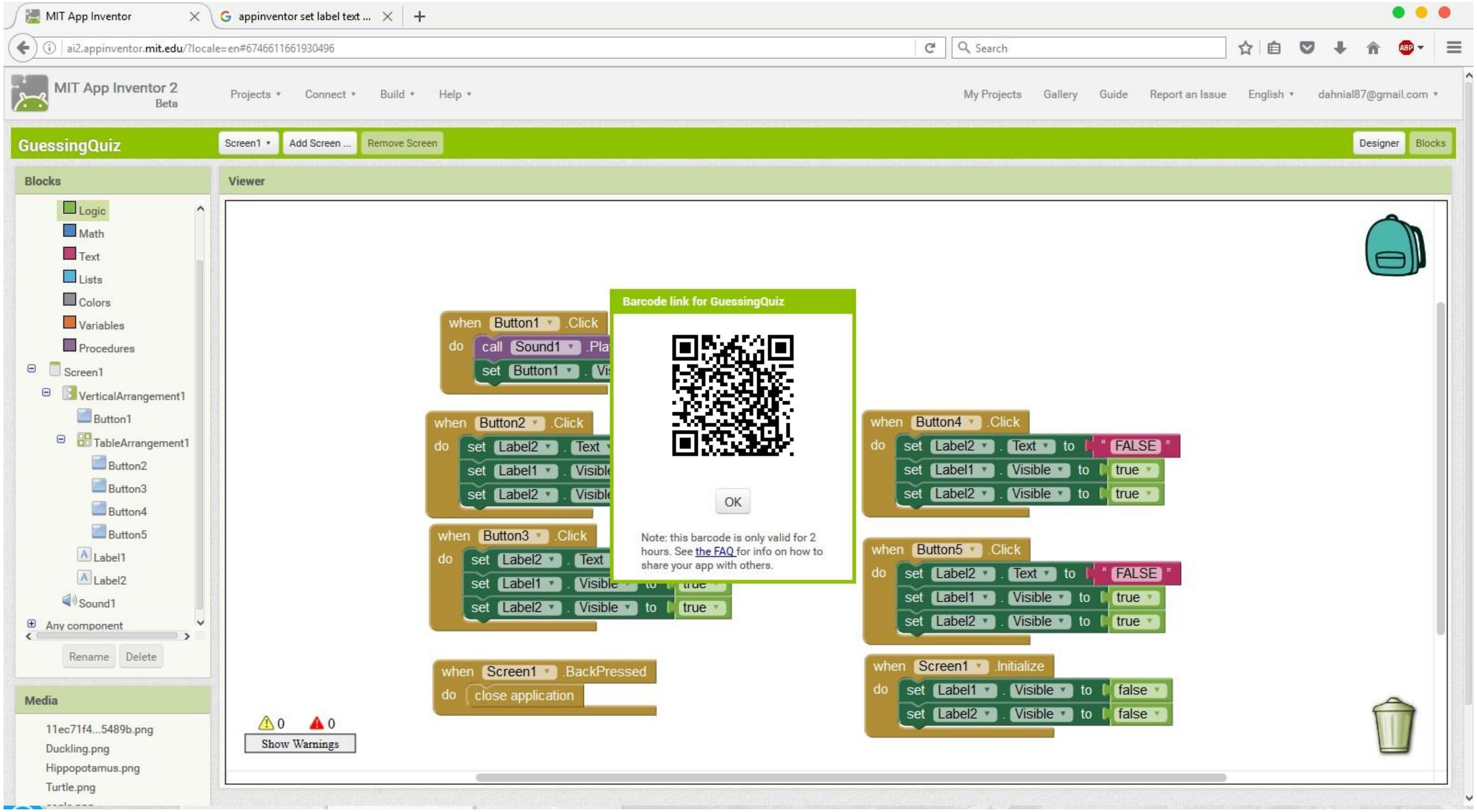
OK

Note: this barcode is only valid for 2 hours. See the FAQ for info on how to share your app with others.

Oops, forget to show label2 when the button is clicked



```
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 Screen1 .BackPressed
do  close application

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 .Initialize
do  set Label1 . Visible to false
    set Label2 . Visible to false
```



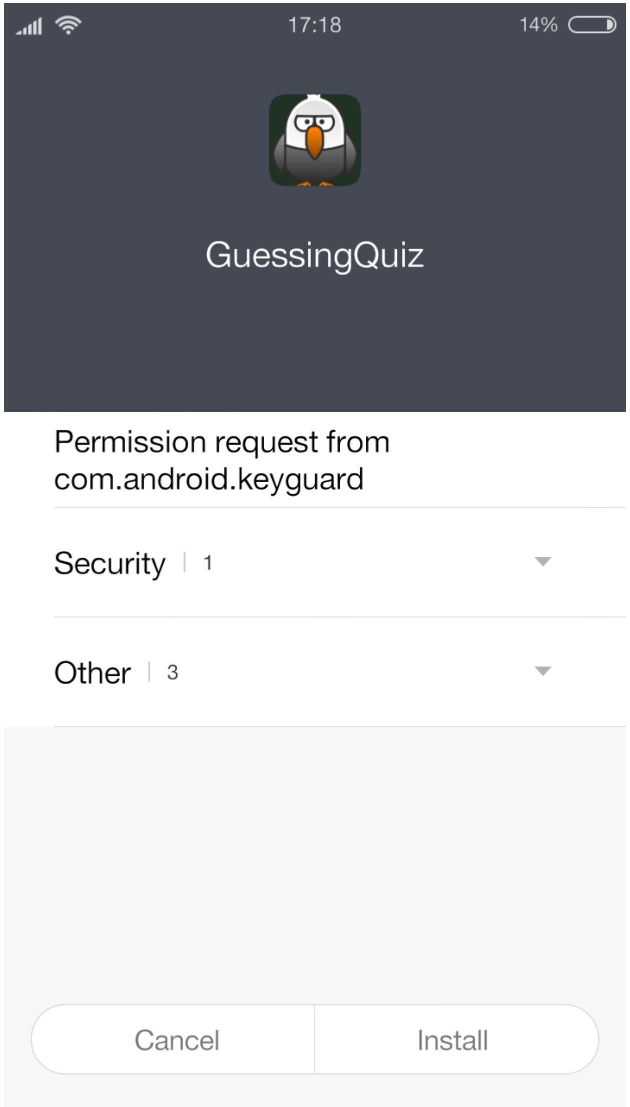
```
when Button1 .Click
do  call Sound1 .Pla
    set Button1 . Vi

when Button2 .Click
do  set Label2 . Text
    set Label1 . Visible
    set Label2 . Visible

when Button3 .Click
do  set Label2 . Text
    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
```

Barcode link for GuessingQuiz

OK

Note: this barcode is only valid for 2 hours. See the FAQ for info on how to share your app with others.

RESULT

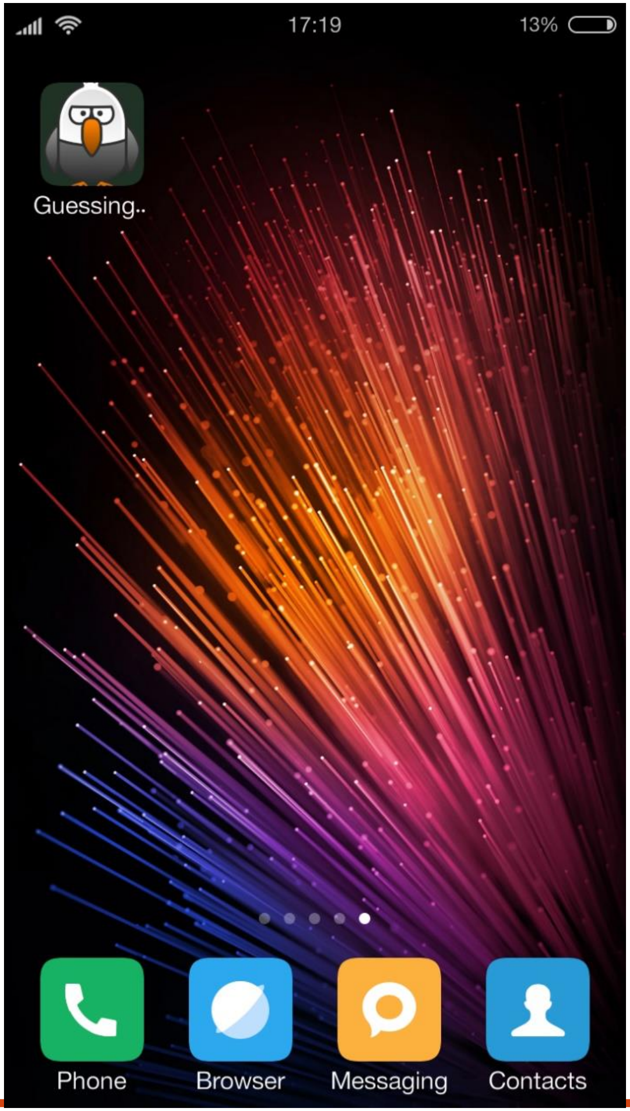

17:18 14%

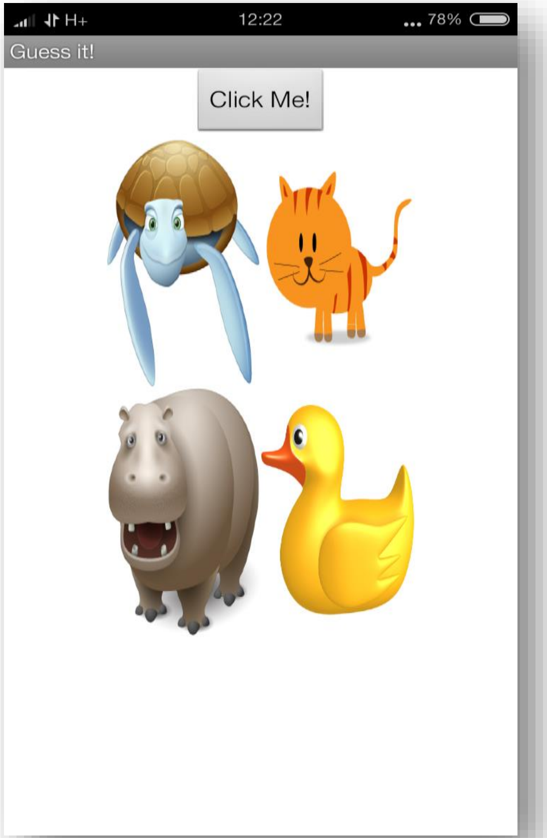
GuessingQuiz

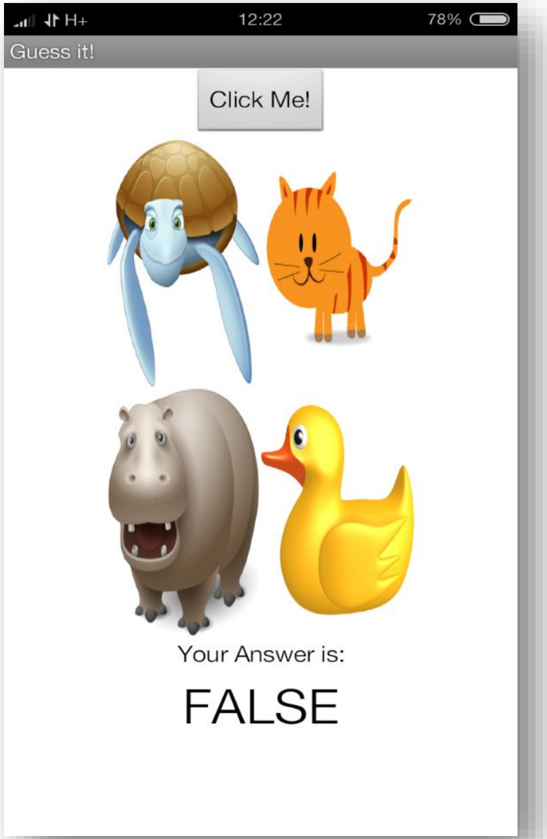
Permission request from com.android.keyguard

Security | 1

Other | 3

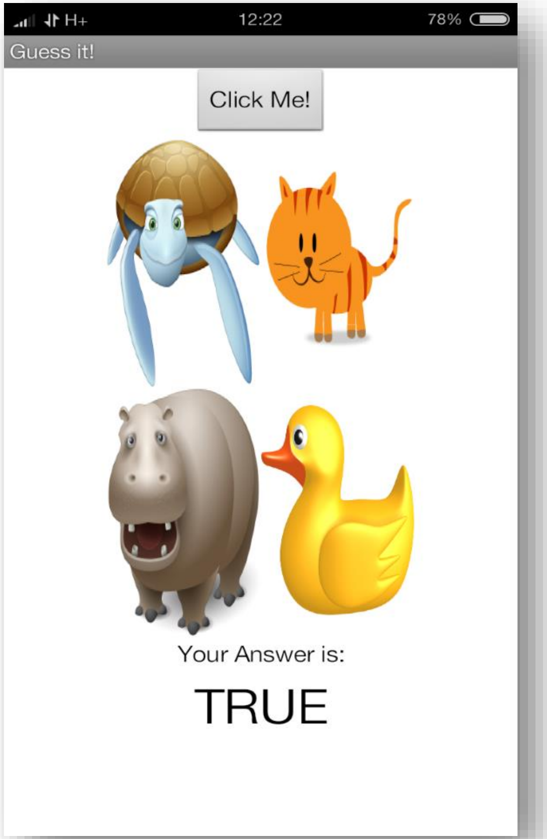






Your Answer is:

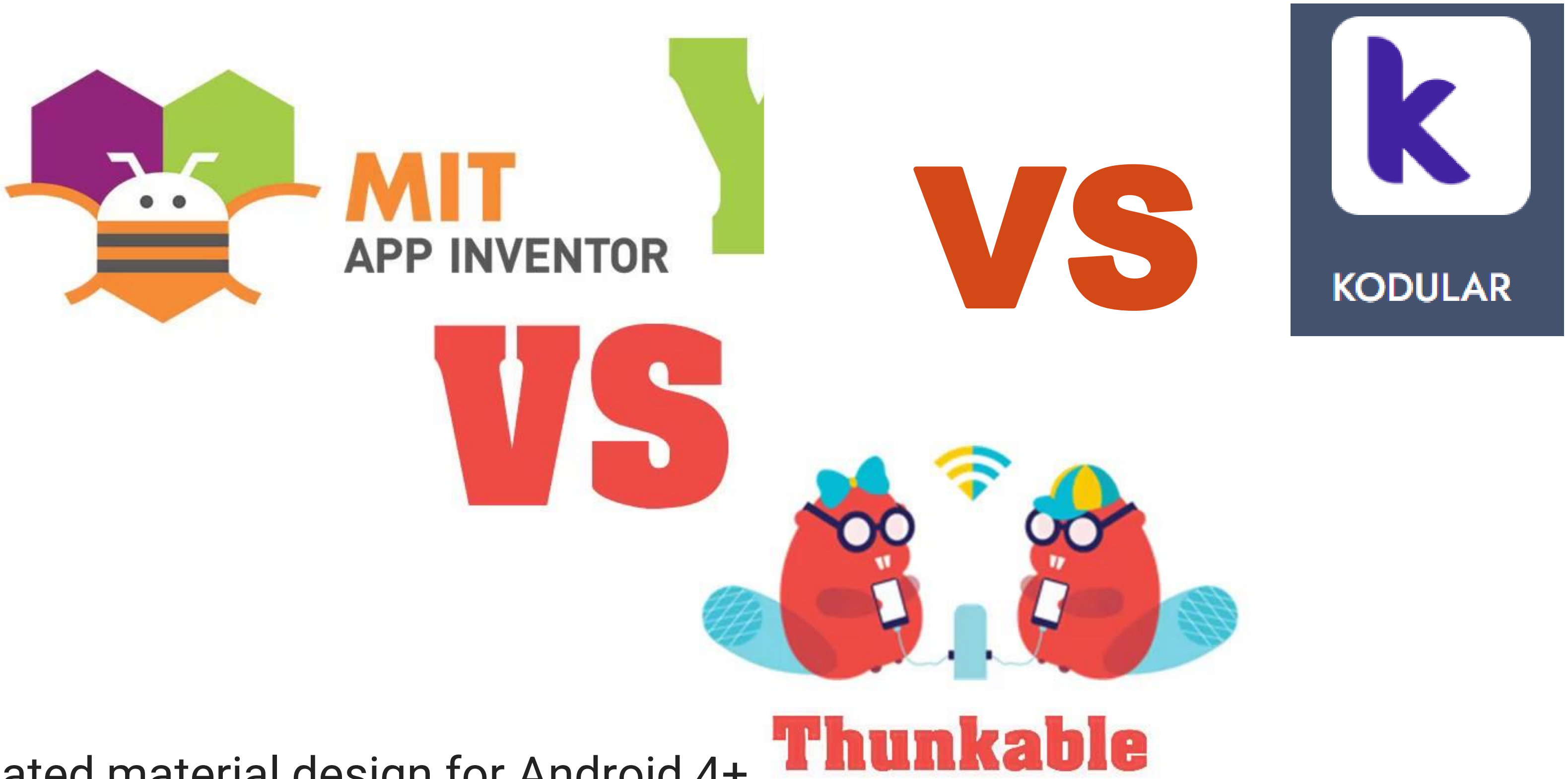
FALSE



Your Answer is:

TRUE

Alternatives: Thunkable & Kodular



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