

APP Design (Day 1)

- Introduction to App Inventor
- Training Plan
- Interface and Control
- Block programming
- Mini-Project



MIT
APP INVENTOR

Introduction to App Inventor

- Created by MIT Computer Science & Artificial Intelligence Laboratory
- Visual programming environment for Android phones and tablets
- Programming editor based on Blockly
- Other Blockly based editors:
 - Blockly Games
 - Code.org
 - MakeCode (used for micro:bit)

Introduction to App Inventor

- Primarily used for teaching coding, but many have created practical and useful apps using it
 - App to help blind navigate around the school in Texas
 - Crowd-sourcing app to locate safe drinking water in Moldova
 - App to help parents track students on buses

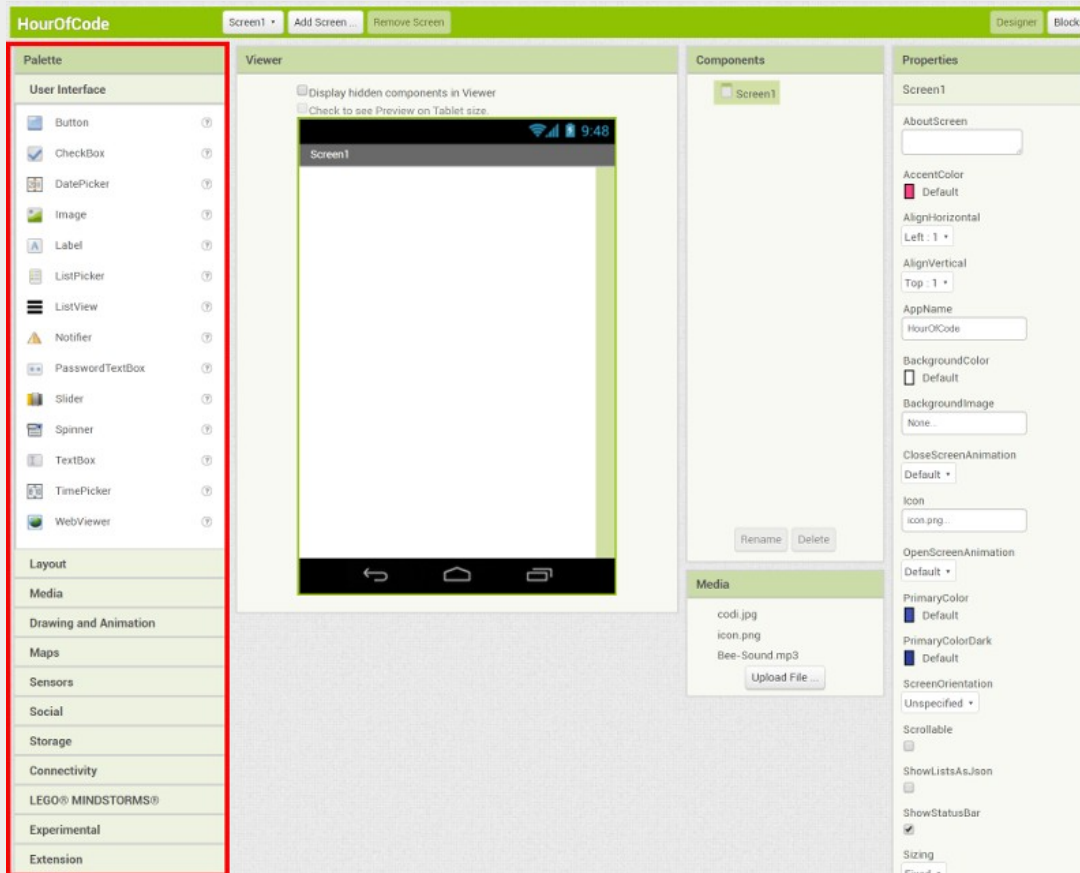
Training Plan

- Day 1: Intro and Basics
- Day 2: Sprites and Graphics
- Day 3: GPS, Map, inter-app communications
- Day 4: Camera, File, Physics
- Day 5: Project Work
- Day 6: Project Presentation

Setting Up (Part 1 of 2)

- Visit <http://appinventor.mit.edu> or do a search for “app inventor”
- Click “Create apps!”
- Login with your Google Account (create a new account if you don't have one)

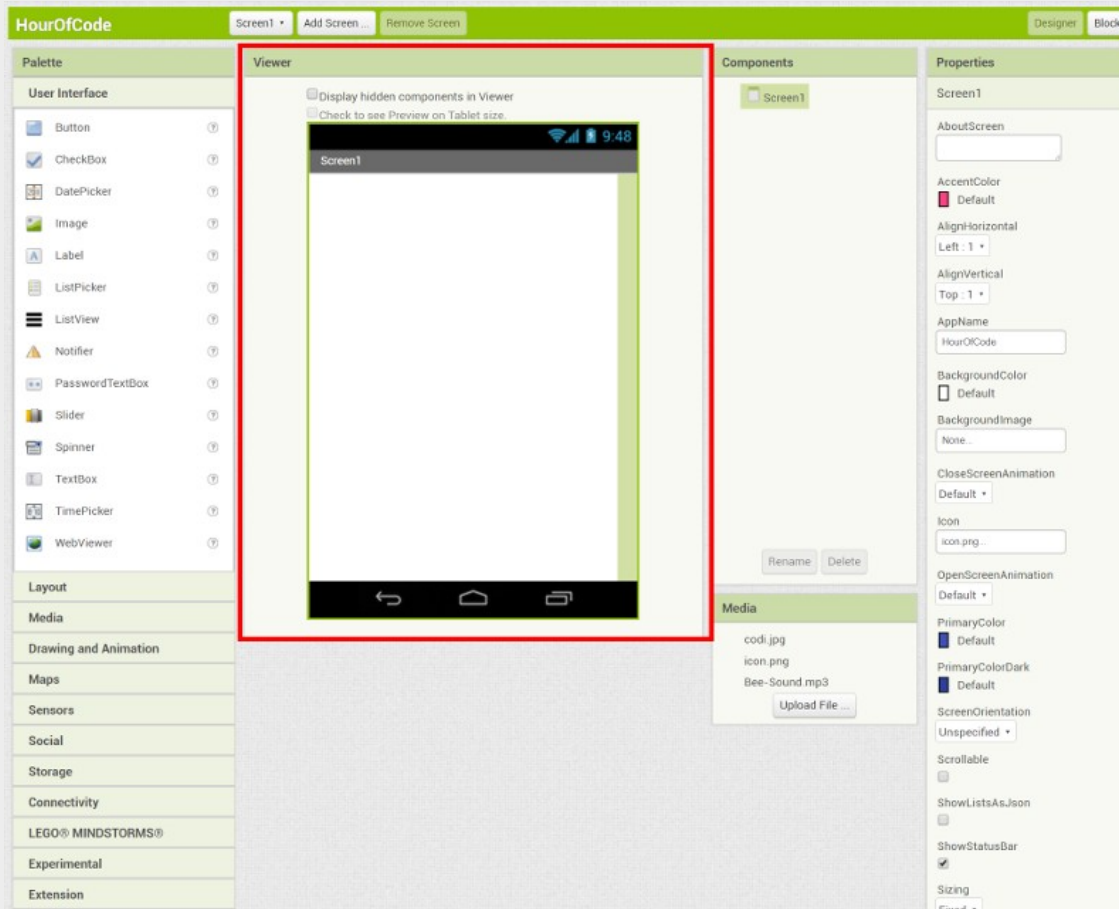
Interface and Control



Palette

Where you select and drag components into the viewer

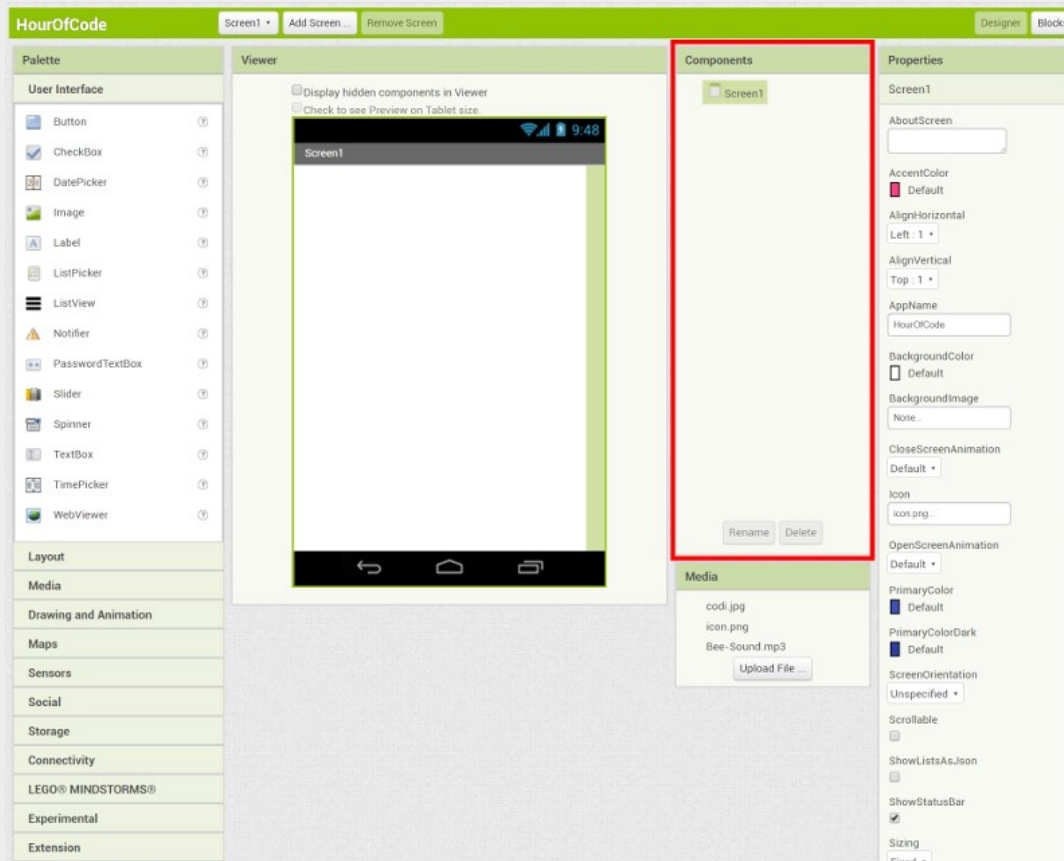
Interface and Control



Viewer

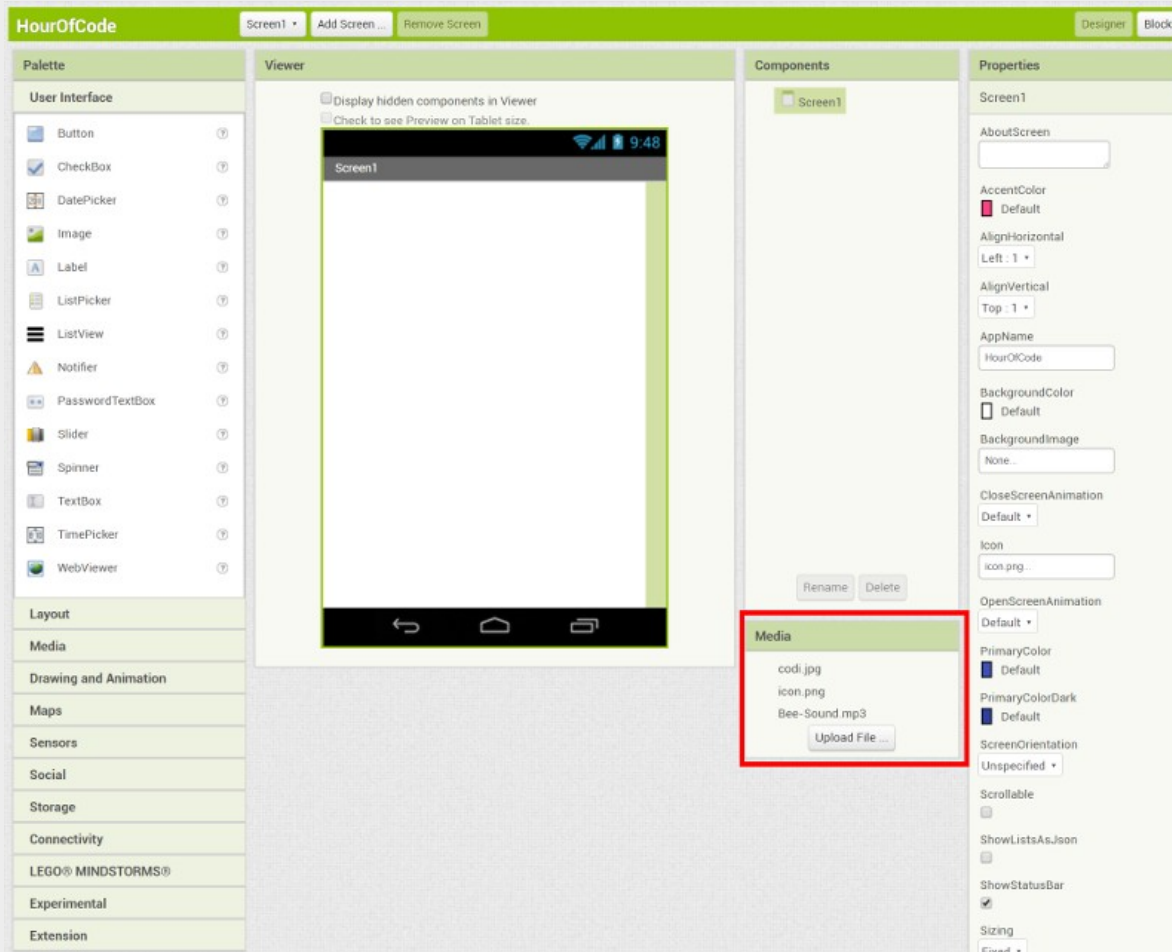
Represents your phone. You can drag components in here.

Interface and Control



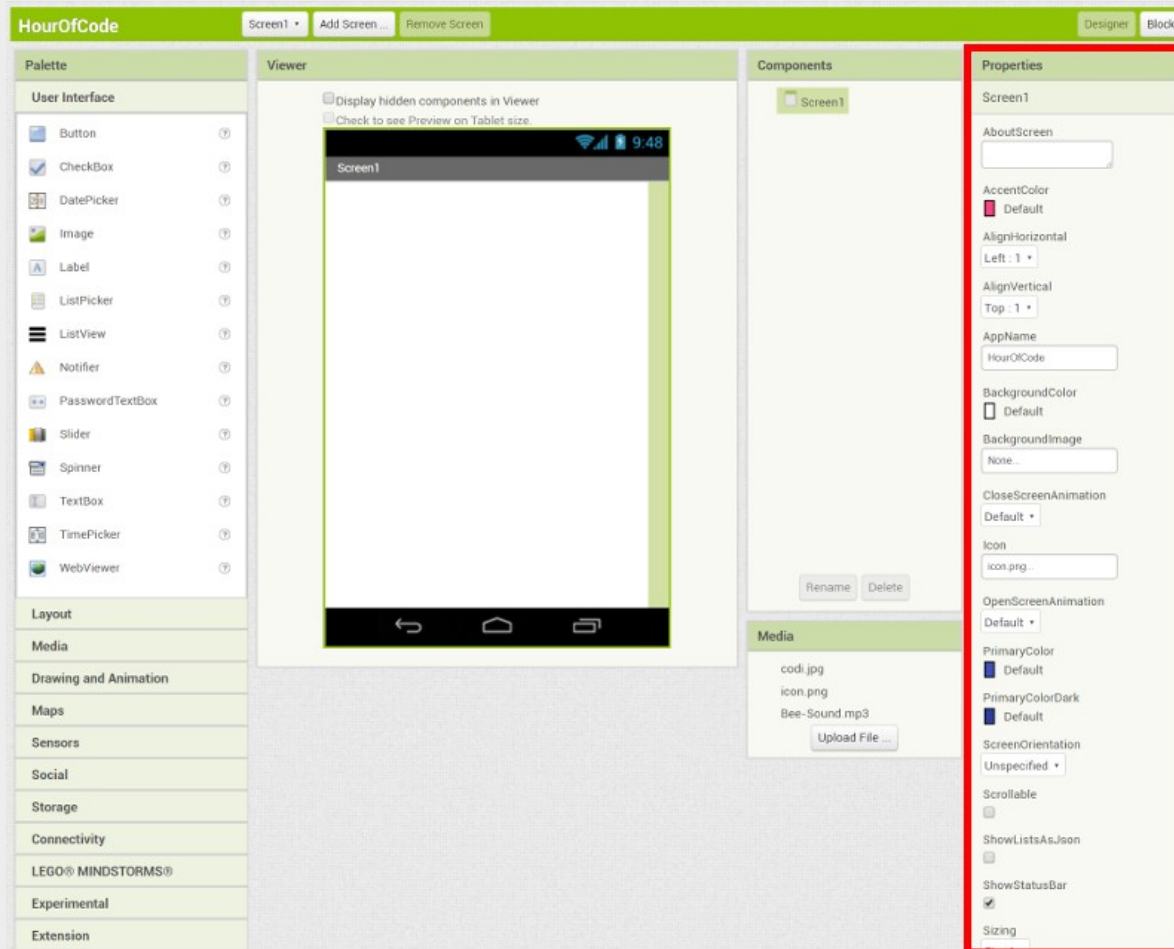
Components
List all added components. Screen1 is already added by default, and you can add multiple screens if needed.

Interface and Control



Media
List all uploaded media (eg. sound, images).

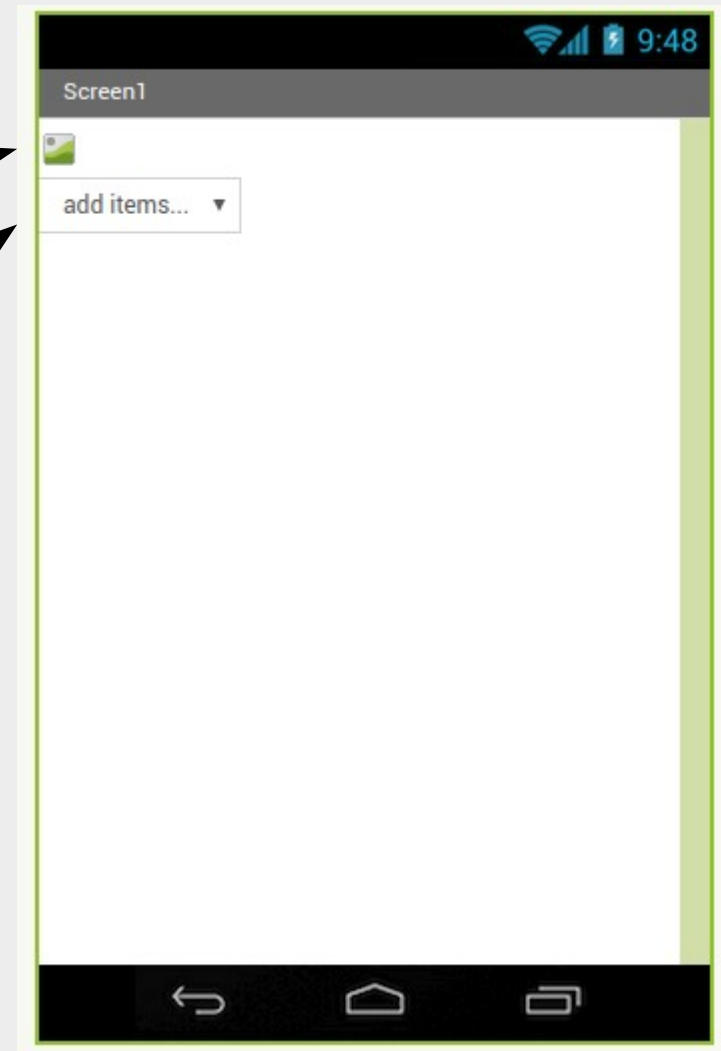
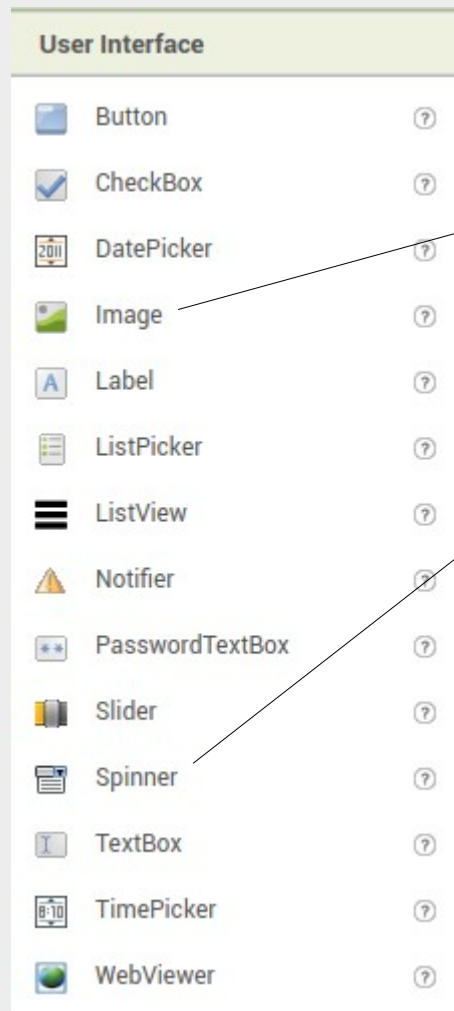
Interface and Control



Properties

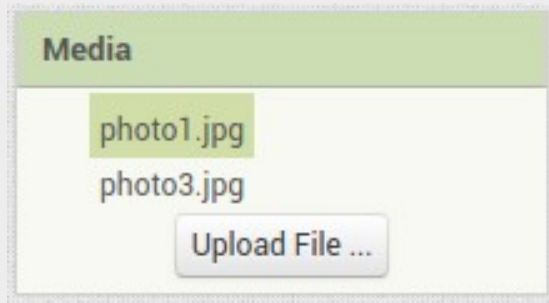
Allow you to change properties for any component (eg. change background color of a button)

Creating Our First Interface

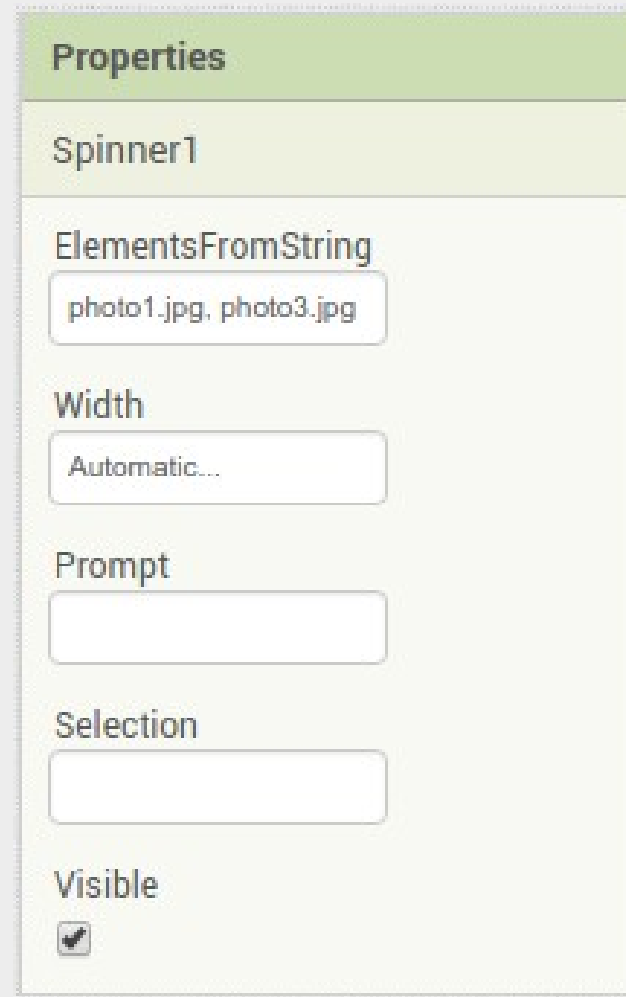


Insert an "Image" and a "Spinner" into the viewer

Creating Our First Interface



Upload 2 or more images through the media window



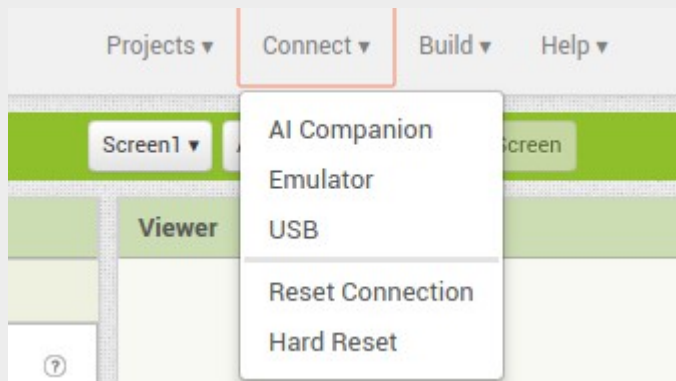
Select the “Spinner”.

Add the image filename into “ElementsFromString”, separating each element with a comma.

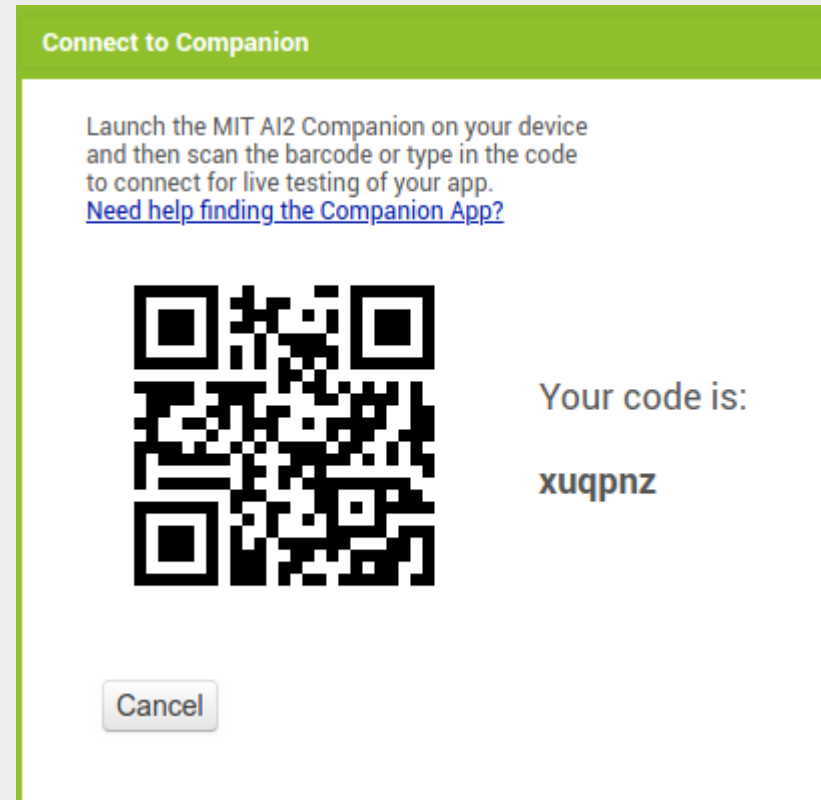
Setting Up (Part 2 of 2)

- On your Android phone, go to the Play Store and search for “mit ai2 companion”
- The app author should be “MIT Center for Mobile Learning”
- Install the app

Connecting App Inventor



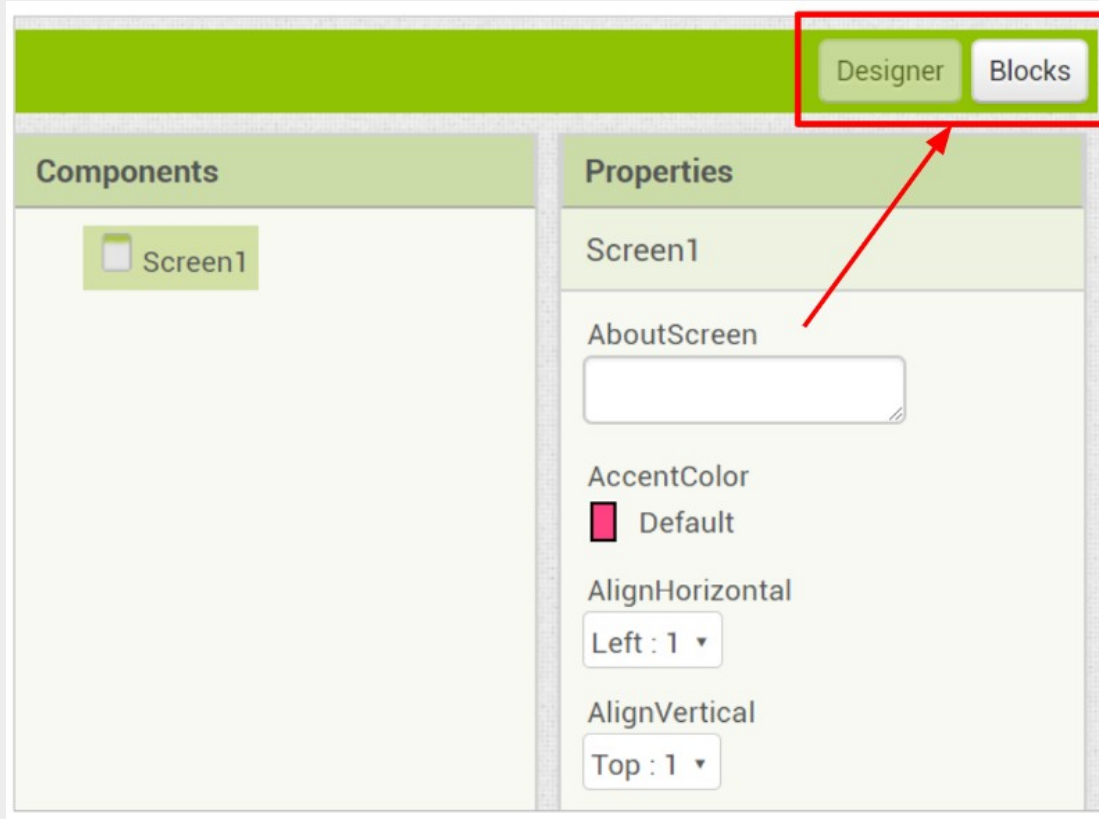
On your browser, click “Connect” followed by “AI Companion”



Scan the QR code with your phone or enter the 6 characters code.

Wait for the connection. The screen you have designed will appear upon connection.

Block Programming

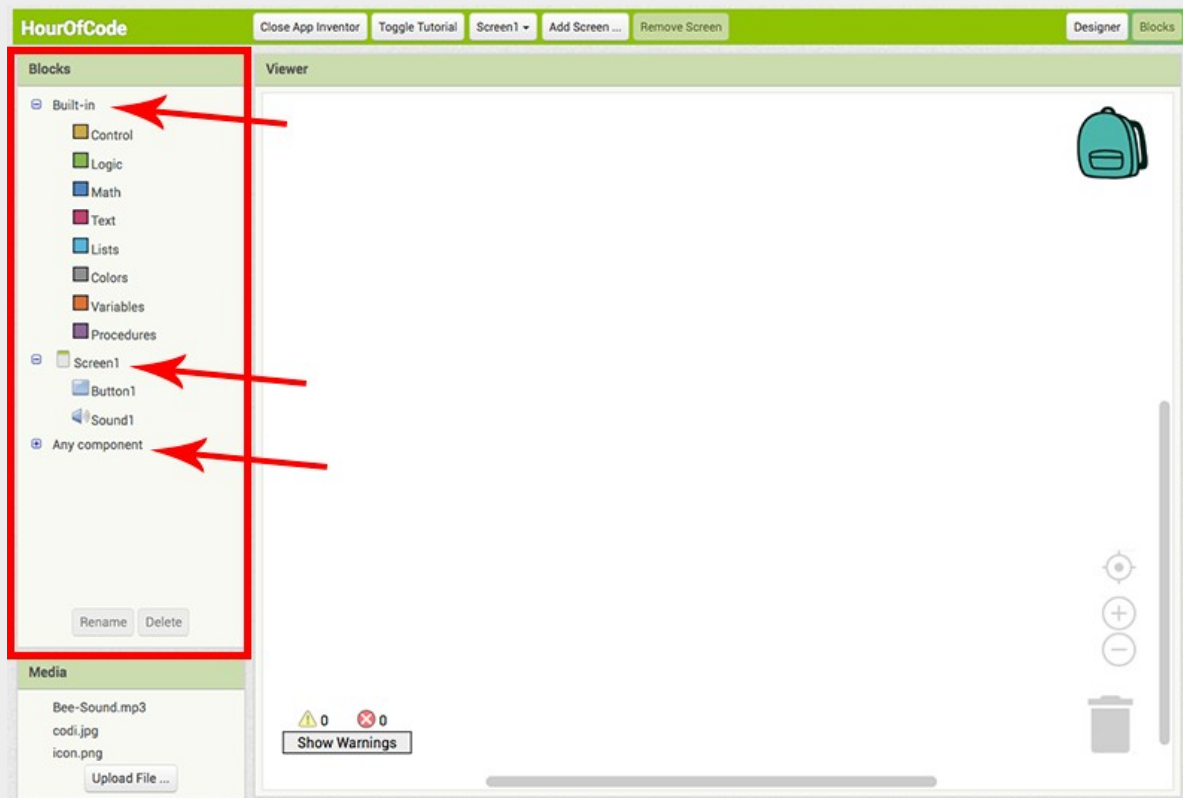


Switch to Blocks Editor

Designer: For designing the user interface

Blocks: For coding program behavior

Block Programming



Blocks window / palette

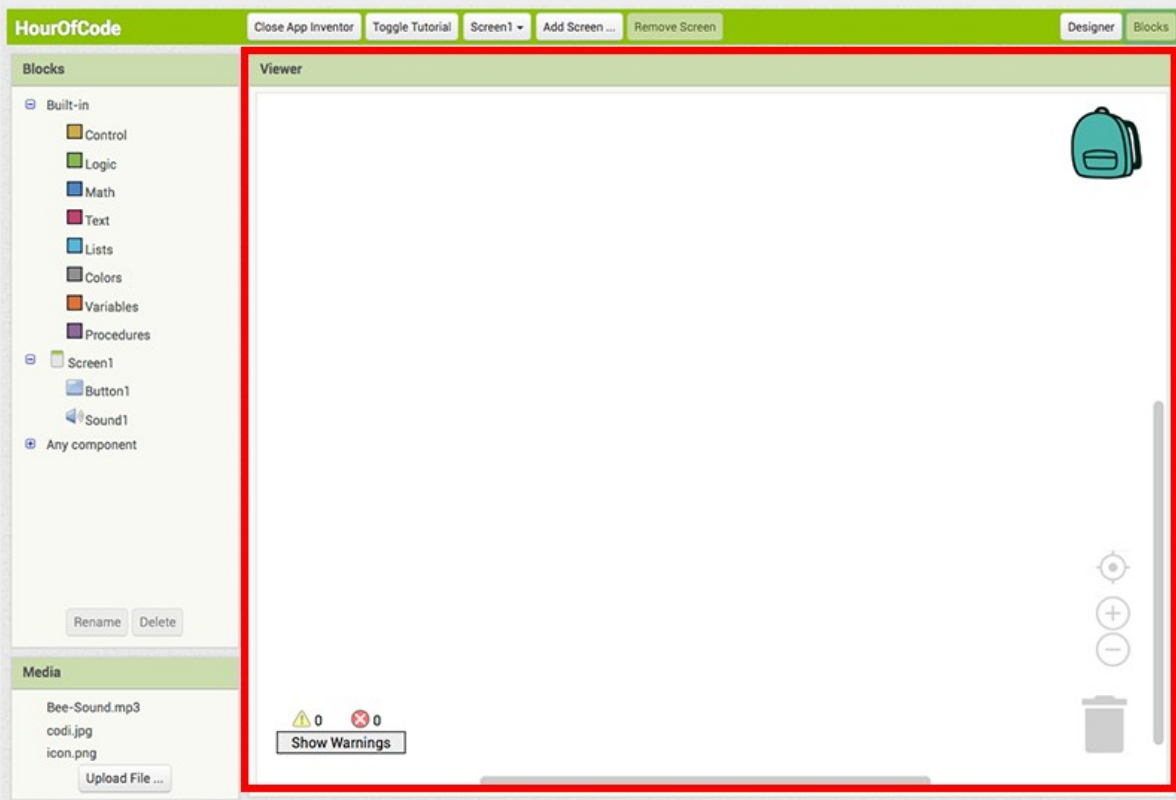
Click to open a list of available blocks

Built-in: Always available regardless of what components are in use.

Components: Blocks for a specific component (eg. Button1, Button2)

Any Components: Blocks for a component type (eg. Any Button)

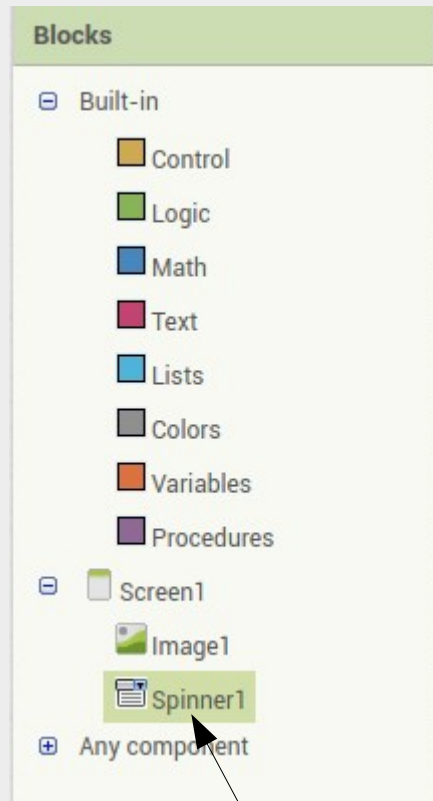
Block Programming



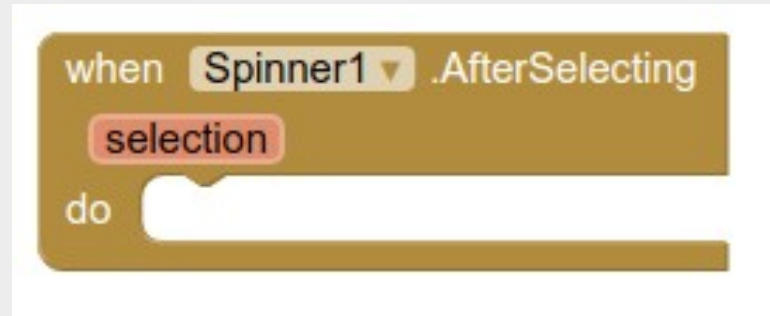
Viewer

Area where you drag your blocks to program your app

Programming the App



Select your spinner...



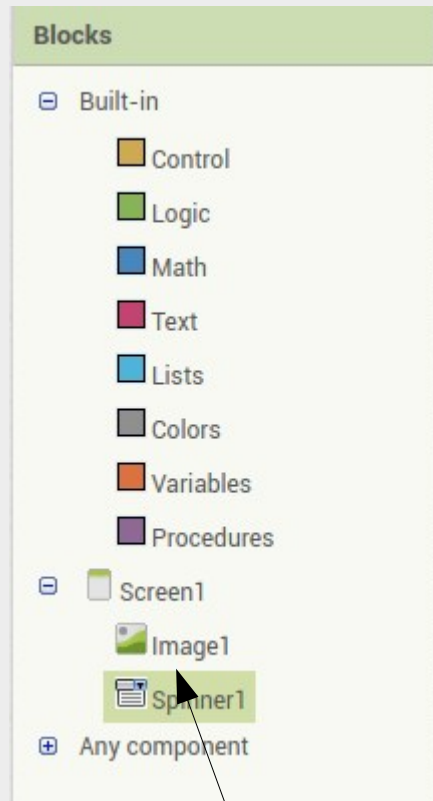
...and drag the above block into your viewer window

Event-driven programming

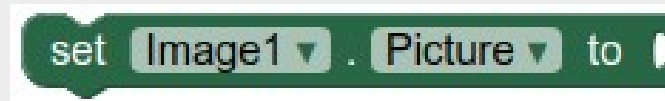
Programming for graphical user interface typically uses an event-driven approach. In this approach, the program flow occurs in response to an event. The “when” block above represents an event.

While any programming language can be used with an event-driven approach, most introductory courses in programming uses a procedure-driven approach.

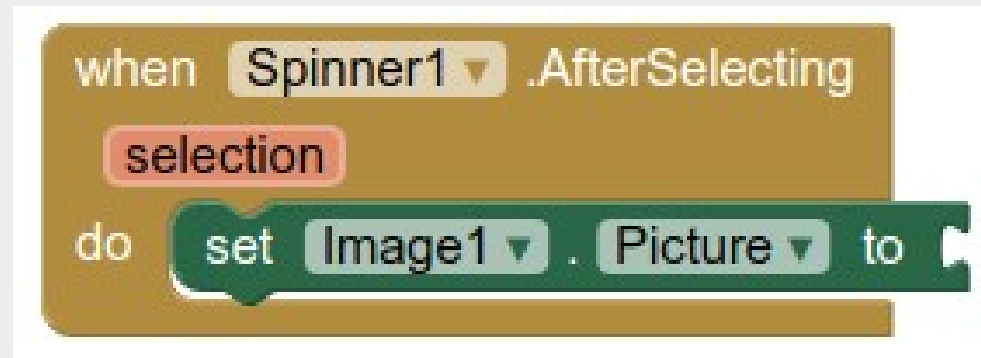
Programming the App



Next, select your image...



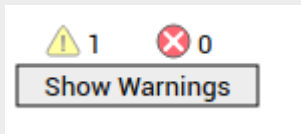
...and drag the above block into your viewer window. Put it inside the "when" block like this...



Programming the App

Warnings and Errors

After the last step, you should see this at the bottom left of your viewer.

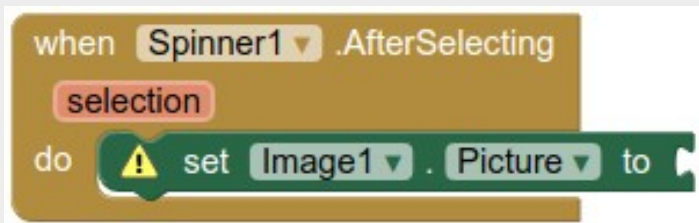


The symbol on the left represents a “Warning”, and there is currently one warning.

The symbol on the right represents an “Error”, and there is currently no errors.

Clicking on “Show Warnings” will cause the warning to be displayed on the blocks.

In this case, the warning occurs because we haven't provided the value for setting the picture (...empty socket on the right).



Programming the App

Warnings and Errors

Error

An unrecoverable problem. The program cannot continue.

Warning

The program encountered a problem, but can still continue. For example; in the current case, since we didn't provide a filename, the program will assume we are setting it to a blank and continue based on that.

Most programming languages have errors and warnings, and the meaning of these two terms are largely consistent across most programming languages.

Programming the App



Hover your mouse cursor over “selection” and the “get” and “set” blocks will appear.

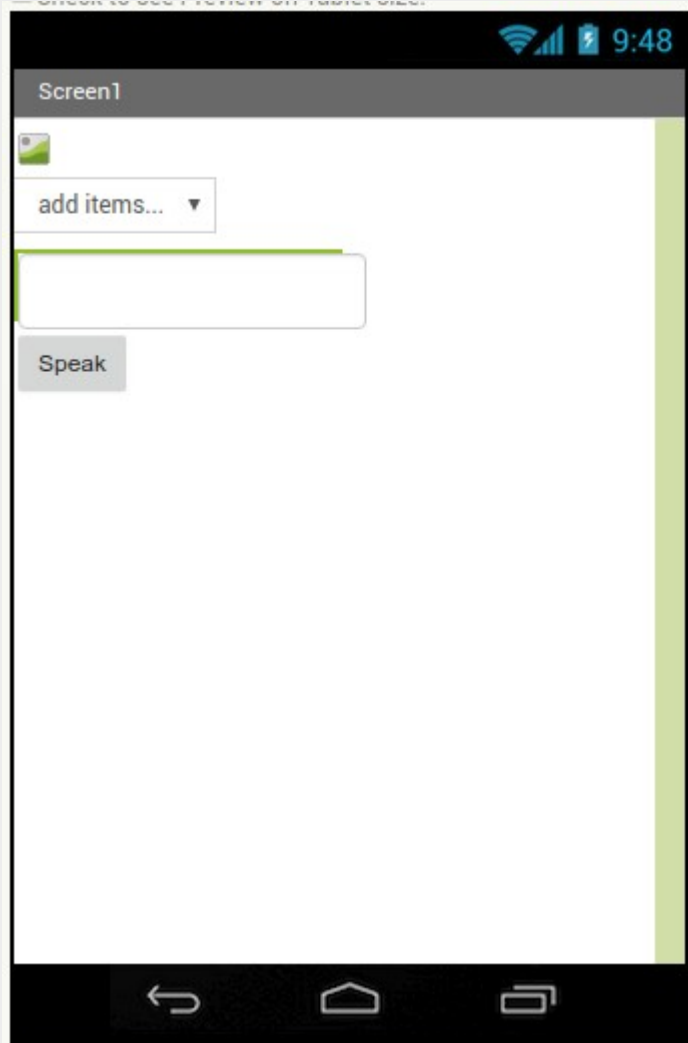
Pull out the “get selection” block and attach it to the end of the “set image” block.

“selection” is a variable name that represents which spinner option was selected. You’ll learn more about variables in a later lesson.

Programming the App

- Try out the app on your phone
- If it doesn't work, try checking...
 - Under “ElementsFromString” are the values the same as the filename in media?
 - App Inventor may change the filename upon upload. Double check that the names match!
 - Try resetting the connection between your browser and your phone

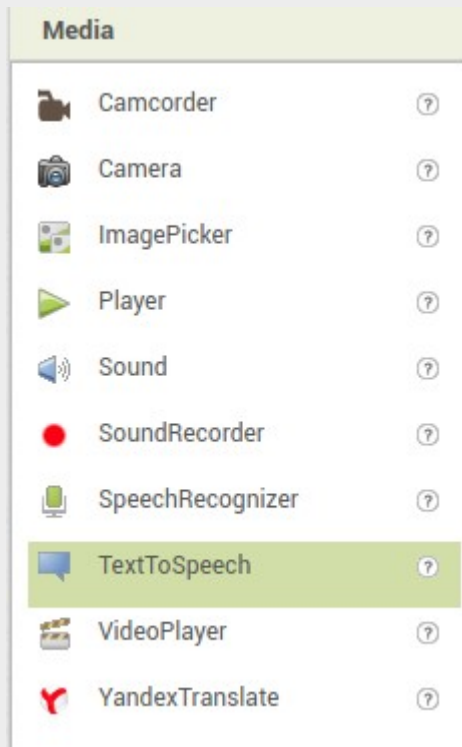
Adding Components



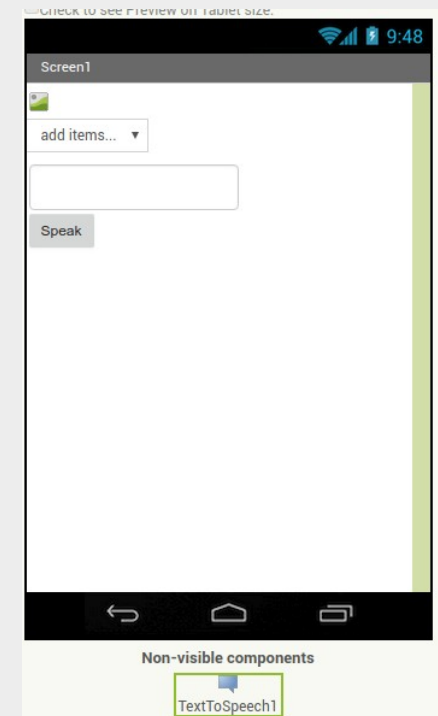
Add a “TextBox” and a “Button” into your viewer.

Select your button, then from the properties window, change the text to “Speak”

Adding Components



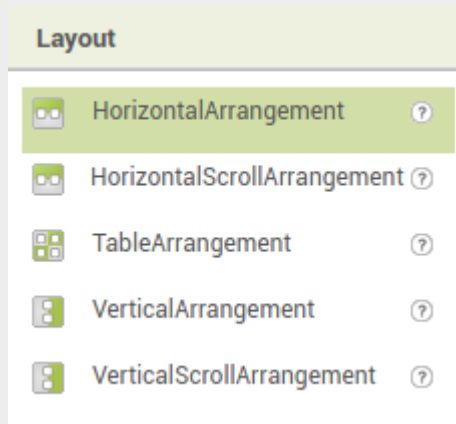
From the component palette, select “Media” then drag a “TextToSpeech” component into the viewer.



The “TextToSpeech” component will appear at the bottom, as it is not a visible component.

Adding Components

What if we want the button to be on the right of the text box? You can't just drag it into position!

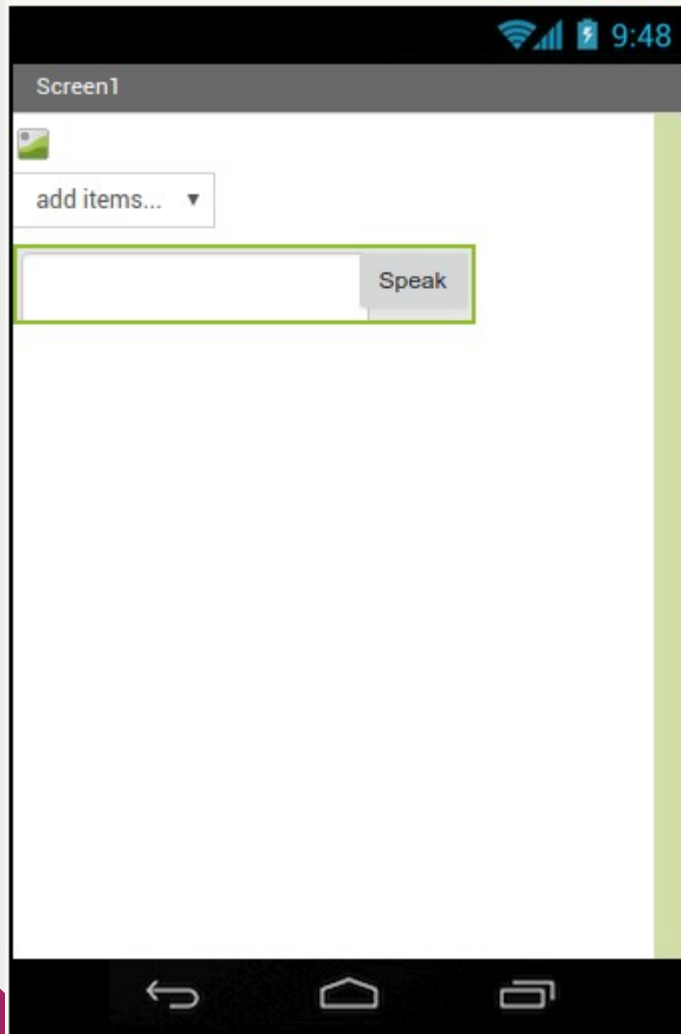


Drag a “HorizontalArrangement” component into your viewer.

The “HorizontalArrangement” component doesn't do anything by itself. It is used to contain other component, and any components inside it will be given a horizontal arrangement.

Many graphical interfaces restricts your ability to place a component at an exact position. While this may seem inconvenient, it is because your app is expected to run on many different devices with different screen size and shapes. Components sizes may also change when the app is running. Setting an exact location may look fine on one phone, but not on another.

Adding Components



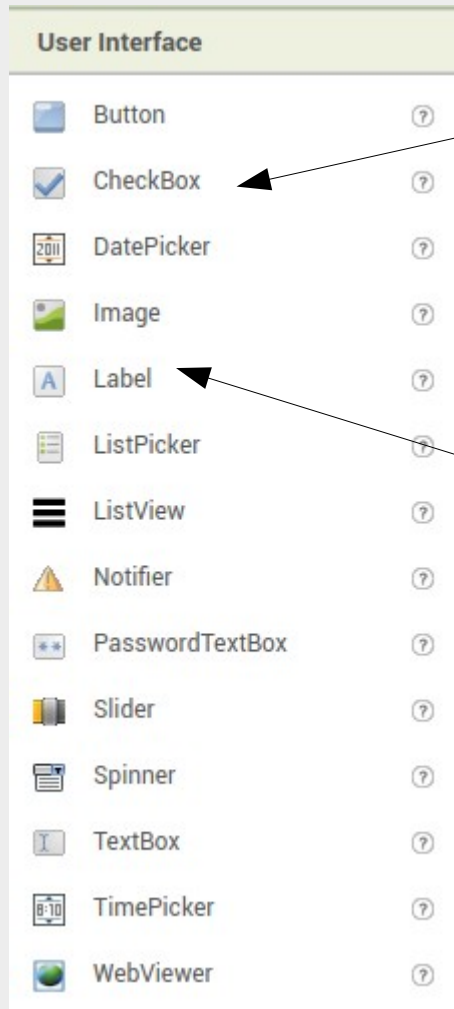
Drag the text box and the button into the horizontal arrangement box.

Optional: Set the width of the horizontal arrangement box to "Fill Parent".

Optional: Set the width of the text box to "Fill Parent".

What happens if you set "Fill Parent" on the text box, but not on the horizontal arrangement box?

Adding Components

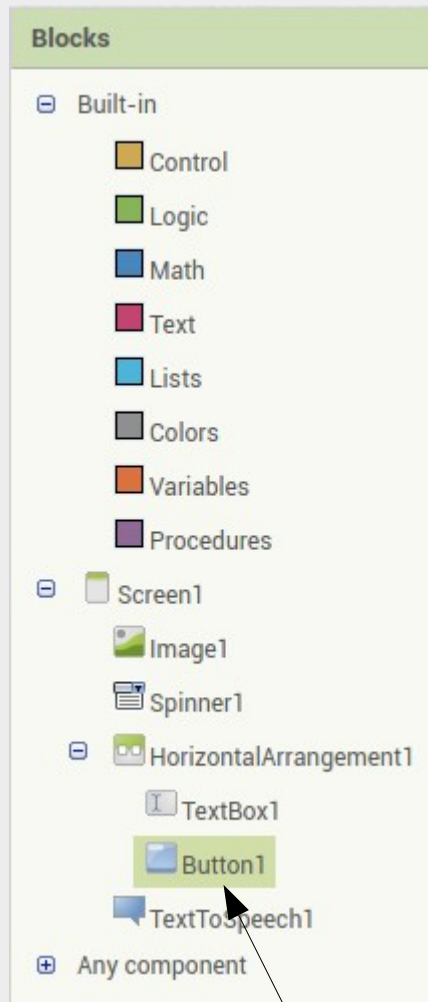


Add in a checkbox and set the text to “Talk fast”

Add a label and change the text to “Press speak to make the app talk”

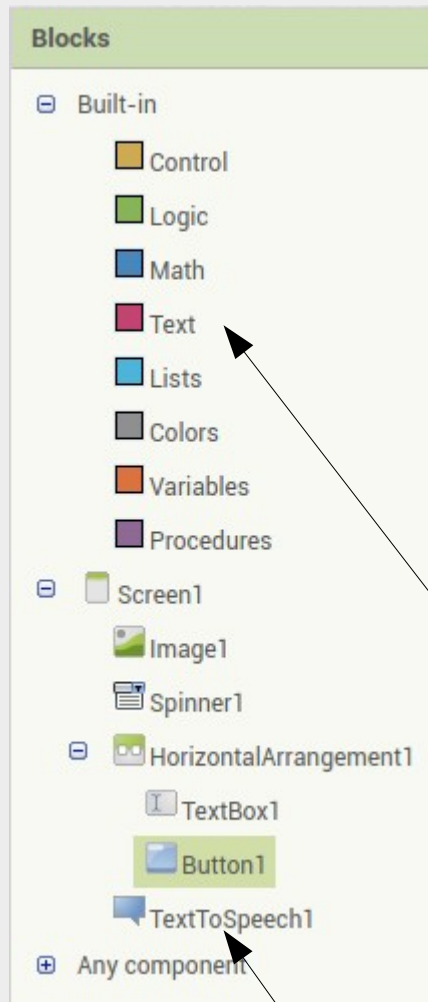
Labels are not interactive components. Through the block programmer, you can change their text, color, fontsize, etc, but it cannot detect clicks.

Programming the App



...and drag the above block into the viewer.

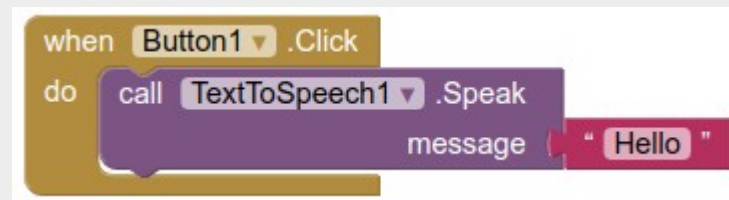
Programming the App



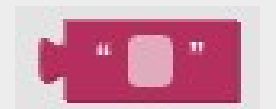
Next, select
TextToSpeech...



...drag out a "call
TextToSpeech1.Speak" block and add
it inside the button click event block.

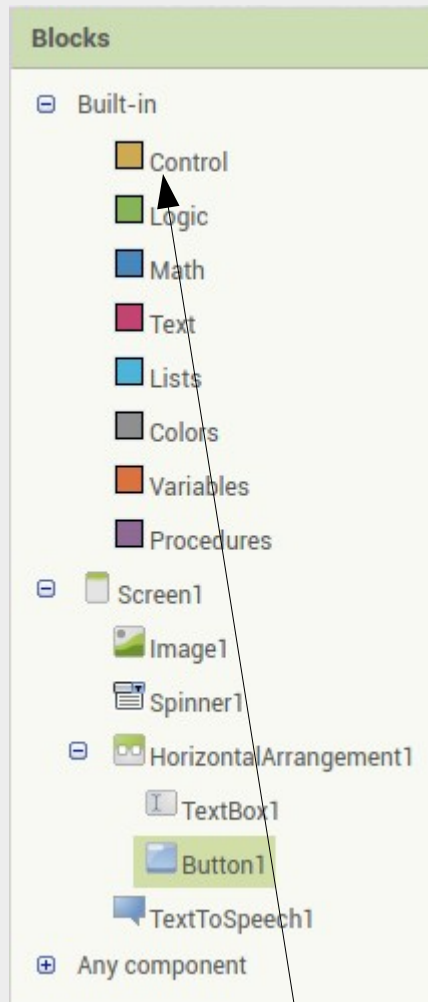


From "Text", drag a empty text block,
add it to the speak block and type in
the message "Hello"

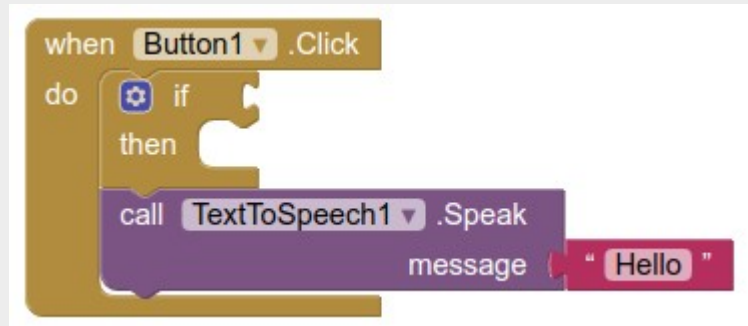


Empty Text

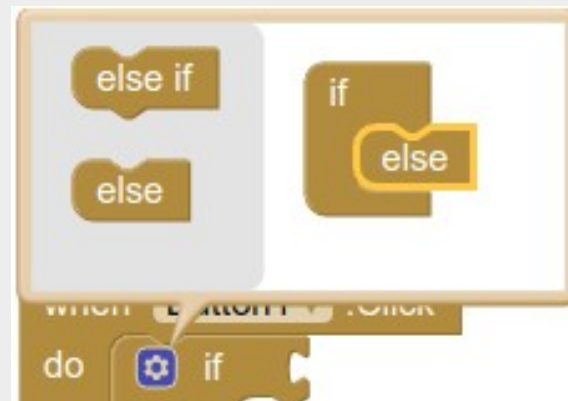
Programming the App



Next, select Control...

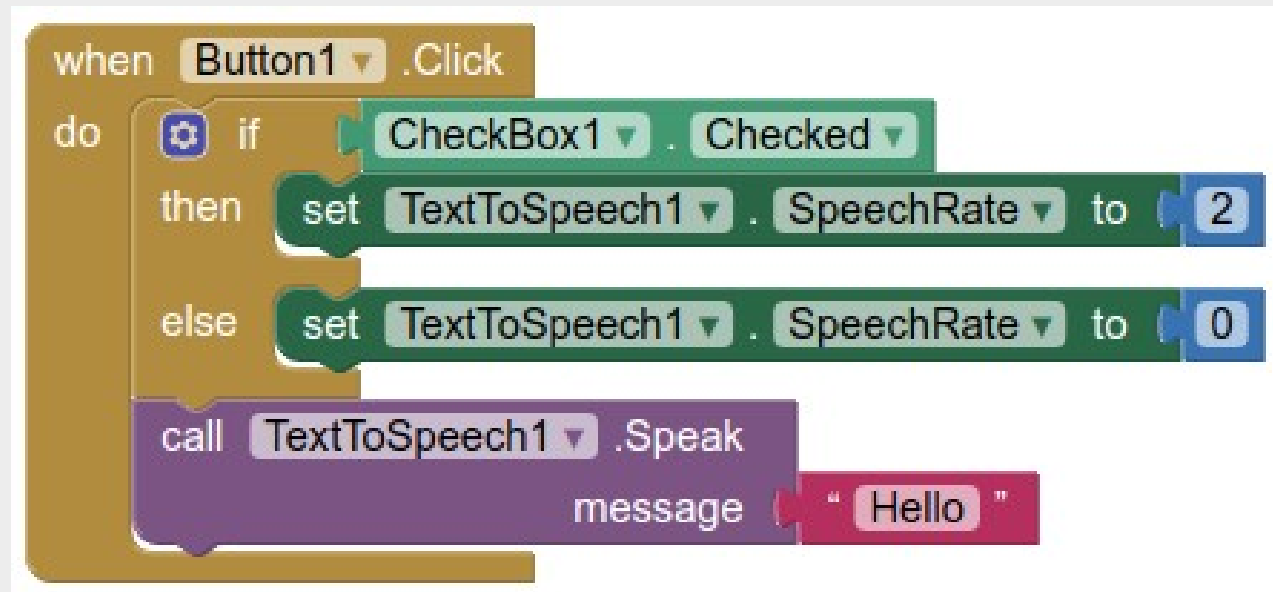


...drag out an "if" block and add it to the top of the button click event block.



Clicking on the gear will open up an option window. Add an "else" into the if block.

Programming the App



Add the following blocks into the “if” block.

“Checkbox Checked”: This sets the condition for the if.

“Set SpeechRate” to 2: The “2” can be found under Math. This sets the action to be done if the condition is true.

“Set SpeechRate” to 0: This sets the action to be done if the condition is not true.

Programming the App

- Try out the app on your phone
- If it doesn't work, try resetting the connection between your browser and your phone

Challenges

- We haven't used the textbox yet. Make the app speak the text in the textbox.
- Bonus Challenge: Explore the YandexTranslate component (found under media). Can you make the app translate the text to Spanish and speak it? (Tip: Language code for Spanish is "es")

Mini-Project



- Create a fortune telling app
- The machine should produce a random fortune each time you shake it
- Fortune can be visual, spoken, or both
- Bonus:
 - Add in sound effects
 - Add in a gender selection, and change your fortune text to reflect the user's gender
 - Add in graphical effects