CEE-445 Embedded System Design

Embedded System Design:

Last Update:
8/8/2017

FPGA 2 Part 2:
FPGA Bluetooth Android App Design Using MIT App Creator 2

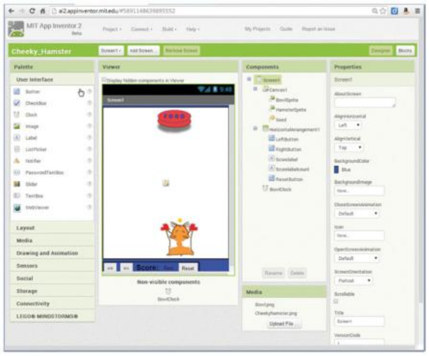
Lab Goal

The goal of this lab is to introduce you with the main parts of App Creator 2, and then build your first Android App.

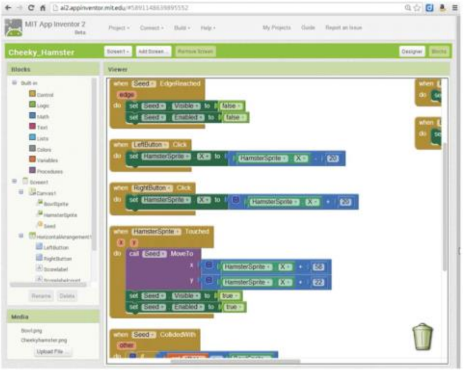
Introduction

MIT App Creator is an intuitive and visual programing environment to build fully functional Android apps for smartphones and tablets. Java programming experience is not required. In this lab, you will be familiar with three steps in the app design: **1)** Design the app screen by using the app *Designer*, **2)** Tell the app what to do by programming the *Blocks Editor*,**3)** Test the program using your *phone. An Android emulator can also be used*.

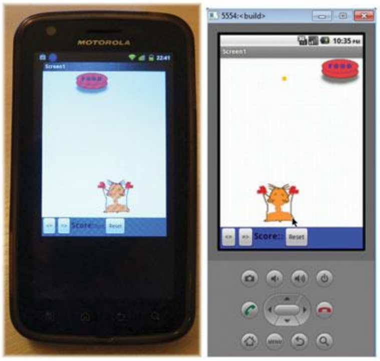
Screen# 1



Screen# 2



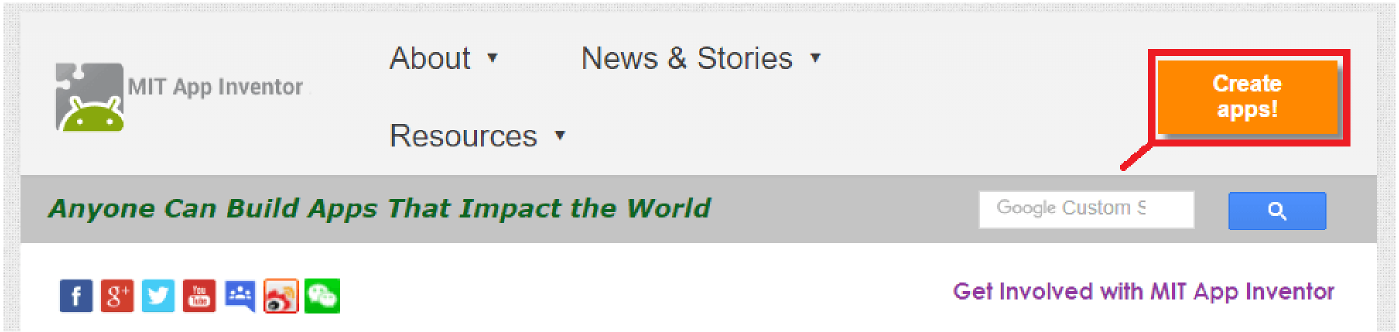
Screen# 3



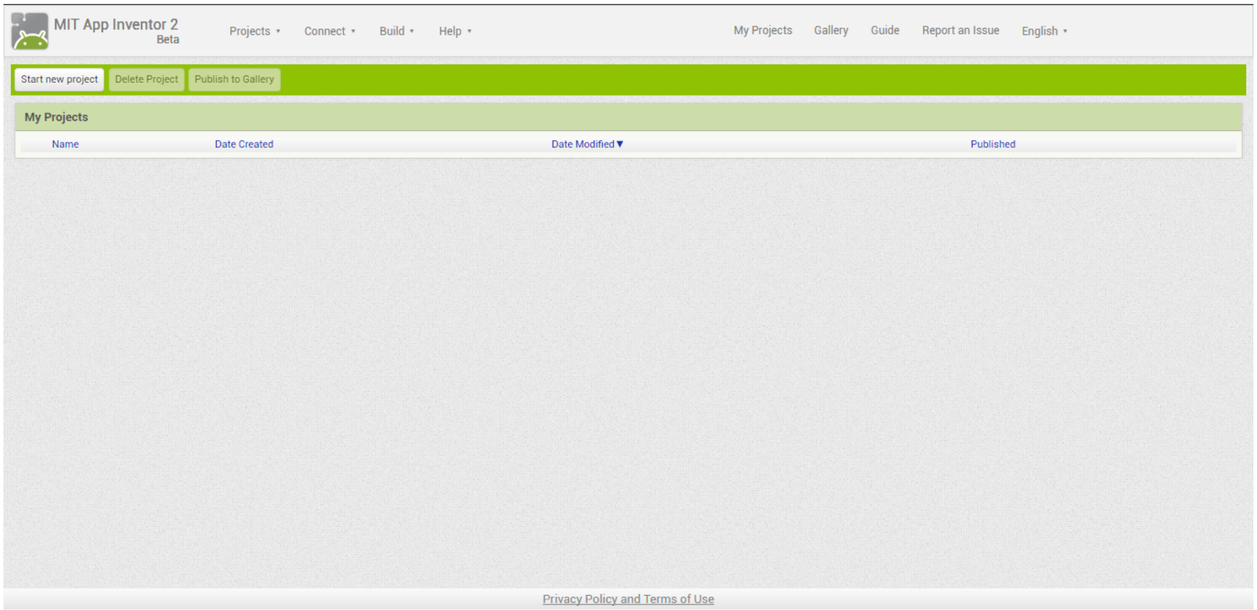
Procedure

This procedure includes an introduction to MIT App Creator. If you already know how to set up the App Creator, you may skip this section 1 starting below.

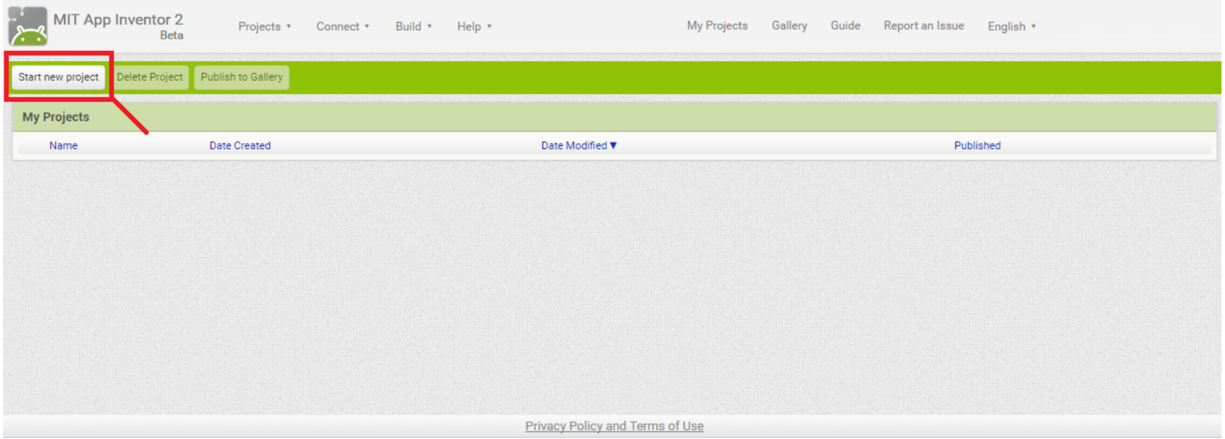
1. Introduction to MIT App Creator 2:
   Accessing MIT App Creator 2: go to http://appinventor.mit.edu/explore/# and press the orange Create Apps button.


MIT App Inventor
About ▾
News & Stories ▾
Resources ▾
Create apps!
Anyone Can Build Apps That Impact the World
Google Custom S
Get Involved with MIT App Inventor

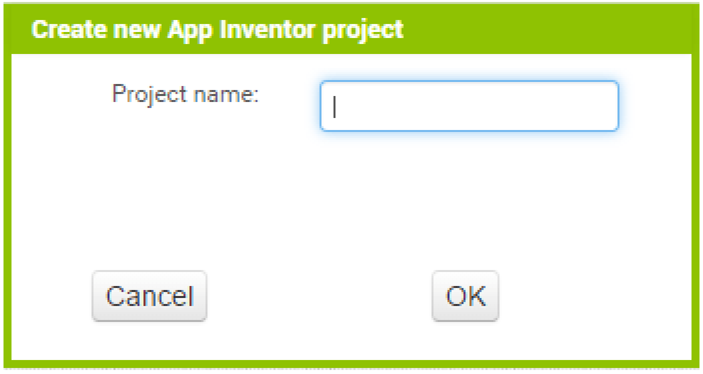
a. You will need to log in to your Google account. Create one if you don't have it. Follow the on-screen steps to log into MIT Creator 2. Once log in, you will see the build window as shown below.



b. Next, click on Start New Project as shown.



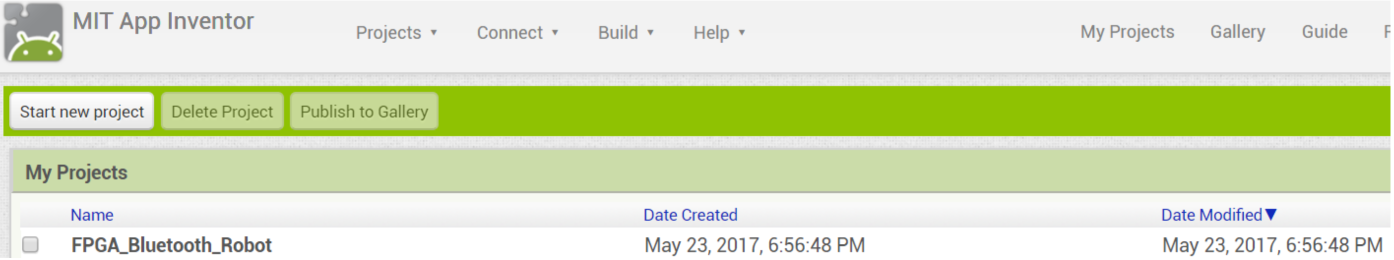
c. Next, name your project title (e.g. FPGA Bluetooth Robot)



Create new App Inventor project

Project name:

d. Click **OK.** Your project is automatically saved. If you go to **Projects > My Projects** you can see all your saved projects.

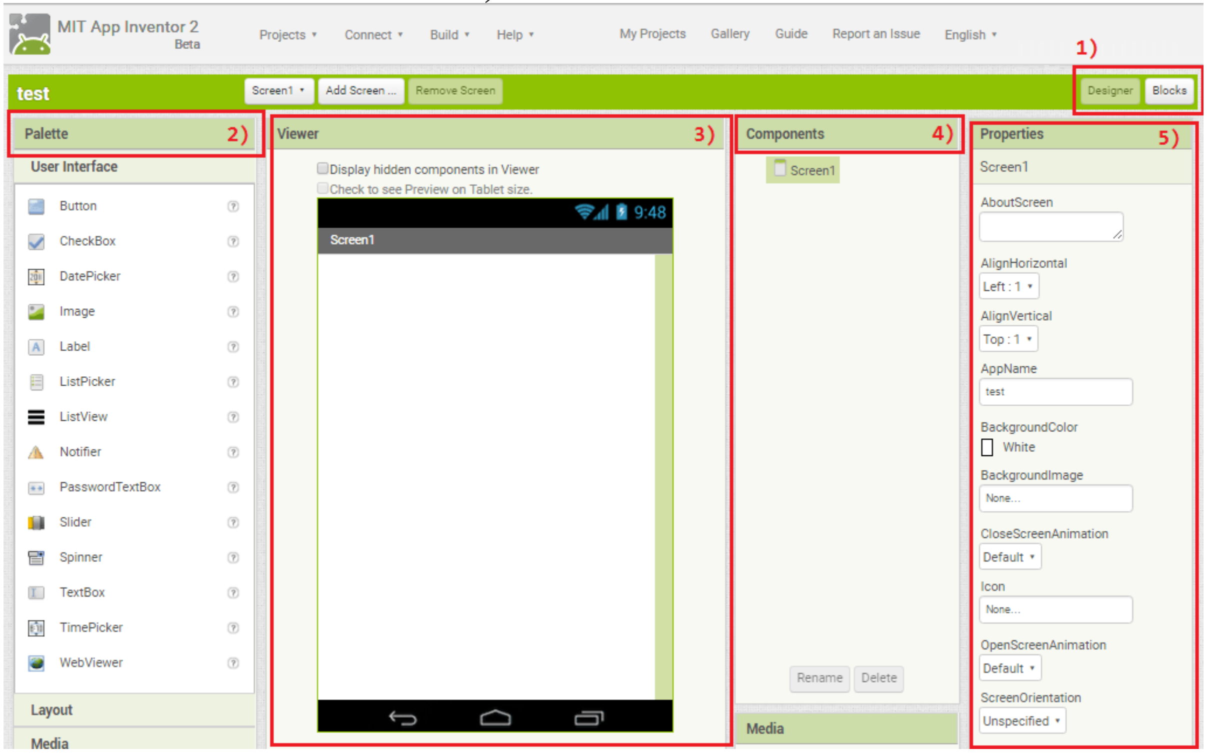


My Projects

	Name	Date Created	Date Modified▼
□	FPGA_Bluetooth_Robot	May 23, 2017, 6:56:48 PM	May 23, 2017, 6:56:48 PM

e. Click on the project name to open the app builder if it is not open.

f. This app is mainly involved with two programming windows in MIT Creator – Designer and Block as shown below marked with 1).



Screen1

AboutScreen

AlignHorizontal: Left : 1 ▾

AlignVertical: Top : 1 ▾

AppName: test

BackgroundColor: □ White

BackgroundImage: None...

CloseScreenAnimation: Default ▾

Icon: None...

OpenScreenAnimation: Default ▾

ScreenOrientation: Unspecified ▾

The Designer and Blocks Editors: The designer editor gives you the ability to add buttons, add text, add screens and edit the overall app look, and the **Blocks** editor allows to create custom functionality for your app.

g. The **Palette** contains the components to build the app design like buttons, sliders, images, labels.

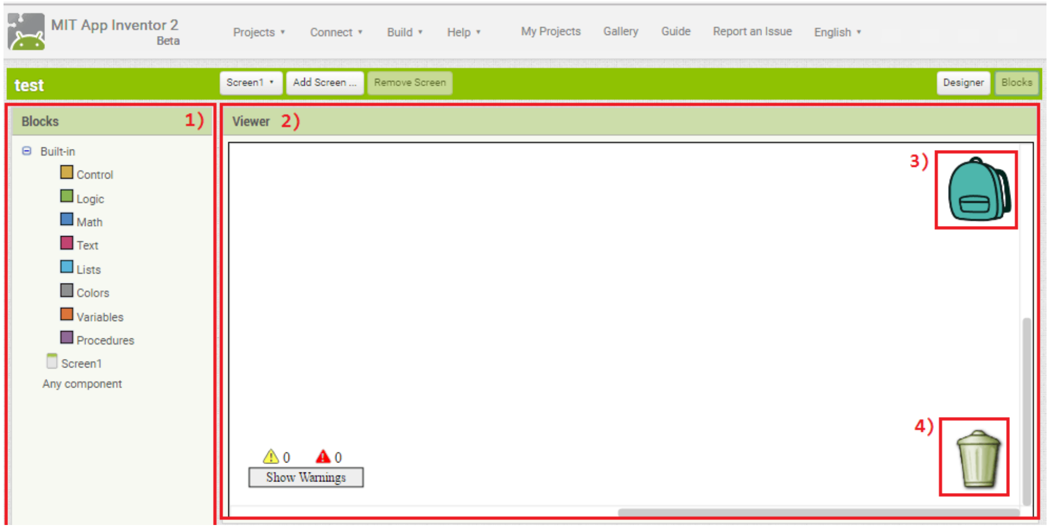
h. **The Viewer** allows to drag the components to build the app look in the viewer.

i. The Components allow to see all the components added to your app and how they are organized hierarchically.

j. **Properties** allows to select your components' properties like color, size and orientation.

Blocks Editor

Select the **Blocks** editor tab**.**

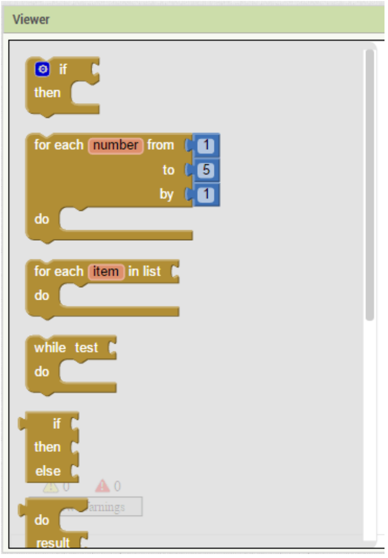


In the **Blocks** editor tab, you have several sections:

k. The Blocks contains the built-in blocks for creating the app's logic. This is what makes the app define the buttons functionalities, send commands to FPGA or microcontroller, or connect to the Bluetooth module. In addition, the blocks look like puzzle pieces that fit into each other. If you can't do something with certain blocks, they won't fit into each other just like the ways we play puzzle games.

l. We have several blocks grouped by categories:

- **Control**: *if/else* statements, *while loops*, and more
- **Logic**: *True*, *False*, *equal*, *not equal*, and more
- **Math**: math operators
- **Text**: blocks that deal with text
- **Lists**: blocks for handling lists
- **Colors**: blocks to handle colors, like choosing a color, make color and split colors
- **Variables:** initialize variables, setting variables values, get variables values, and more
- **Procedures:** procedures are like functions. A procedure is a sequence of code blocks with a given name.



m. In the **backpack icon** where code blocks can be saved to use later. The blocks in the **dustbin** are to be deleted.

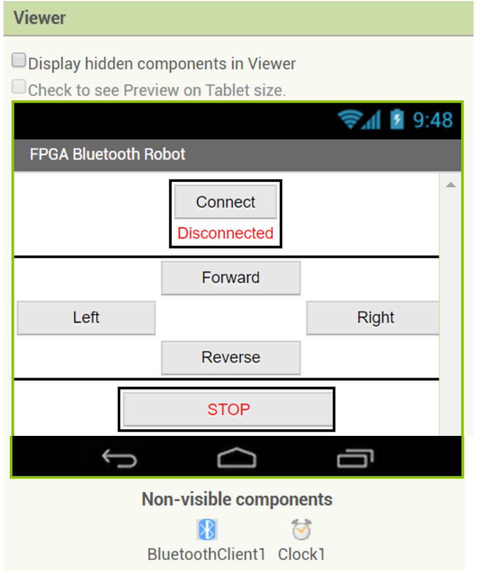
2. The first step in creating an app is to set up a new project and then design it. Designing means both laying out what the app looks like and also deciding which small components you need to make the app work—things like buttons and labels and sounds. We'll build on that idea and make apps that remember and use collections of things using a special type of variable: the *list*. Lists are all around you—especially in the technology you use. If you think about it, we bet you can think of at least five lists you've used today. Here are some examples:

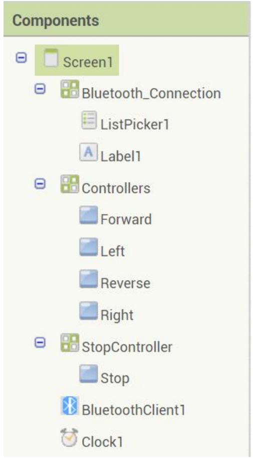
- Choosing a person to send a message to from your contacts list
- Choosing a ring tone from a list of sound files
- Reading a list of updates from your friends on social media
- Choosing a TV program to record from a TV planner list
- Selecting a way to pay from a list of options when internet shopping

Readymade lists

Your phone already contains lists of useful information, and App Inventor can access and use a few of them. You access these lists using *list pickers*.

We are going to create a FPGA Robot Navigation App as shown below. The app was created on the MIT App Creator software, and the viewer window displays the name of the app "FPGA Bluetooth Robot", the button to connect the app to the bluetooth if there is a BT available on the robot and is powered, and five navigation buttons (Forward, Background, Left, Right, and Stop).

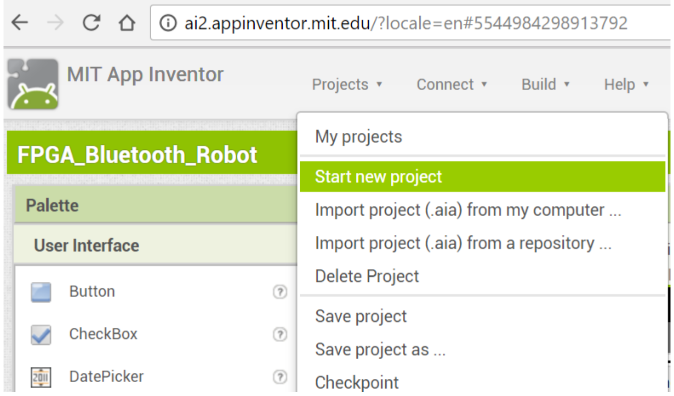




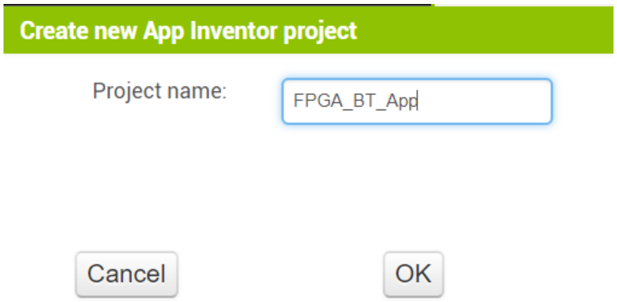
Components

- Screen1
  - Bluetooth_Connection
    - ListPicker1
    - Label1
  - Controllers
    - Forward
    - Left
    - Reverse
    - Right
  - StopController
    - Stop
  - BluetoothClient1
  - Clock1

a. Start a new project: select *Projects* → *Start New Project* → *Enter Project Name* → *OK*



ai2.appinventor.mit.edu/?locale=en#5544984298913792

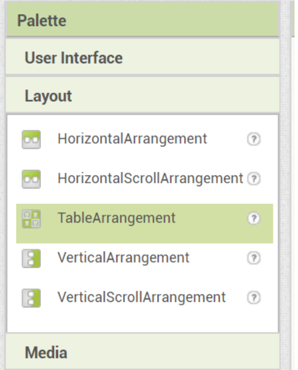


Create new App Inventor project

Project name: FPGA_BT_App

Designer:

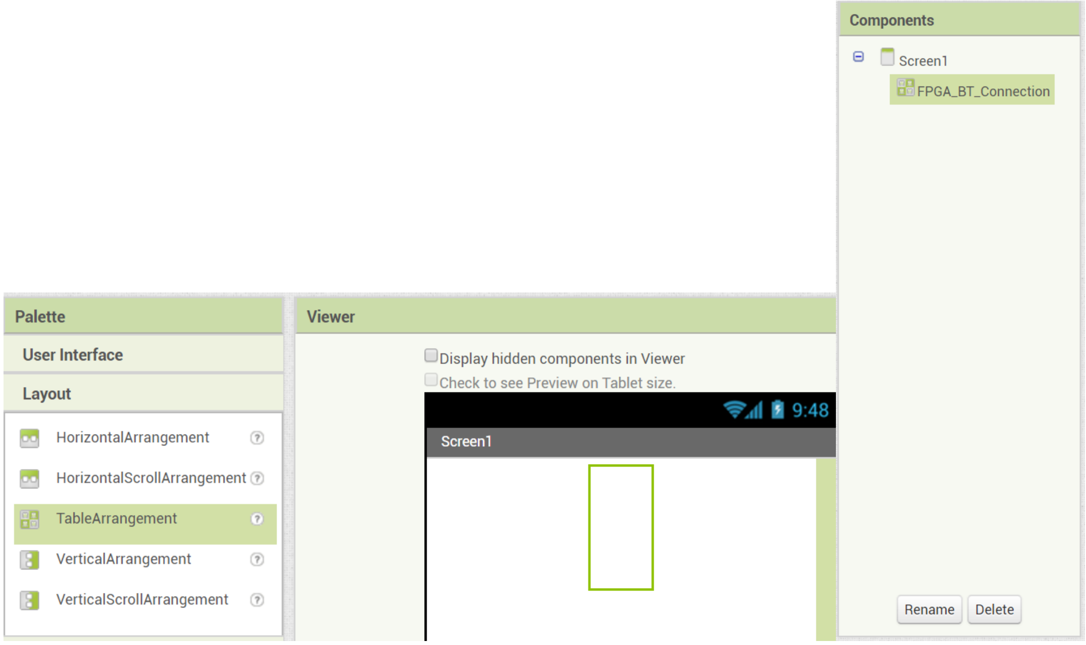
b. Select Designer tab → Layout → Select TableArrangement.



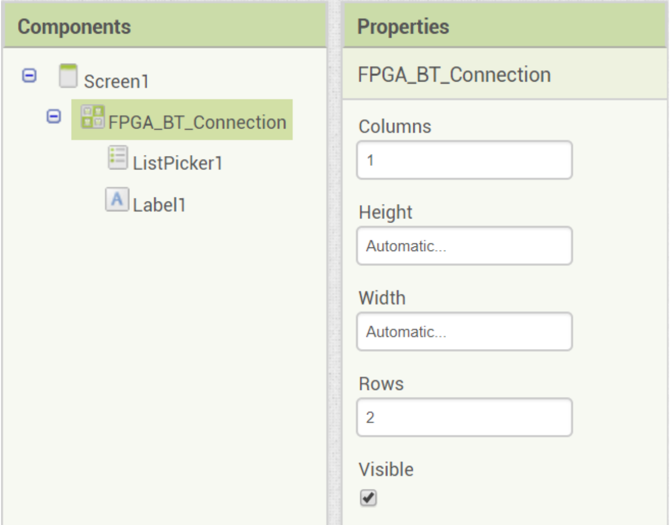
c. Drag and drop the icon ▦ to screen 1 as shown.

d. Go to Components window, Screen 1 and use the Rename tab for a name change to 'FPGA_BT_Connection' as shown below.

e. Go to Properties window, change column to 1 and row to 2.



f. Return to Palette → User interface, select and add ListPicker1 and label component as shown.



FPGA_BT_Connection

Columns: 1

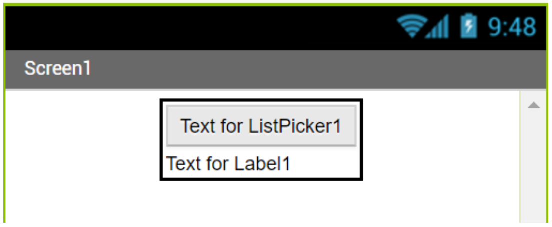
Height: Automatic...

Width: Automatic...

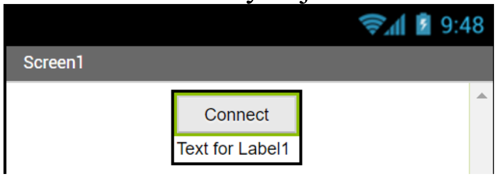
Rows: 2

Visible: ☑

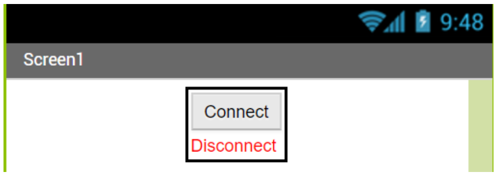
g. Screen 1 should look like the picture below.



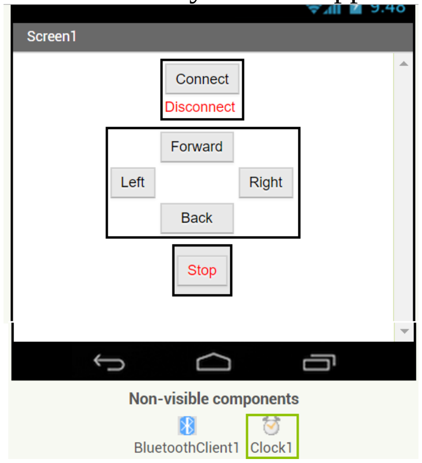
h. Select ListPicker1 you just added and go to its properties and change its text to 'Connect'.



i. Select label1 you just added and go to its properties and change its text to 'Disconnect'. You may change its text color to red. Ask the instructor if you have questions.

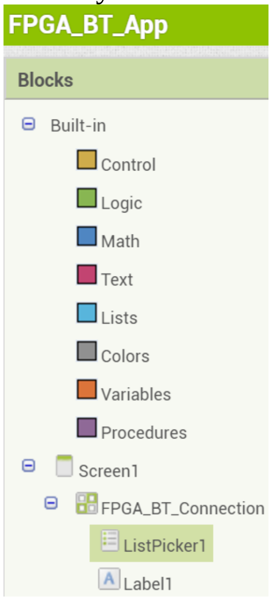


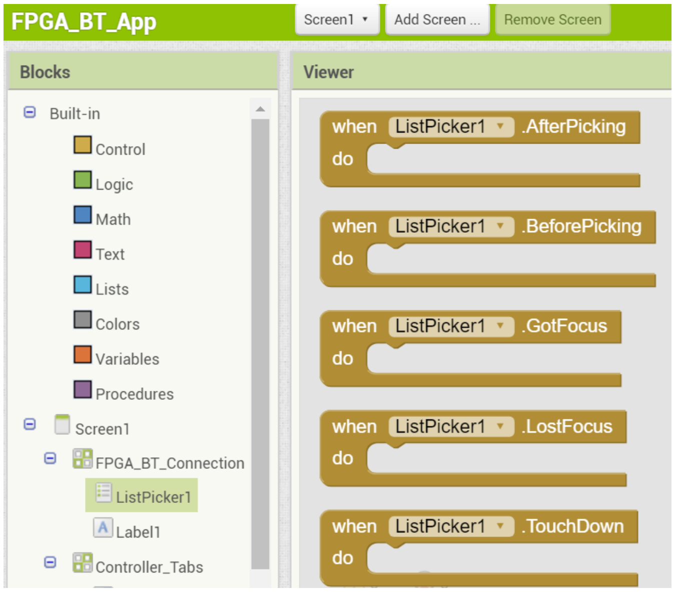
j. Use the same method to create controller buttons (left, right, forward, and back), a BluetoothClient and Clock component as shown. Note that last two components are non-visible component, which means that they will not appear in the Screen1 window.



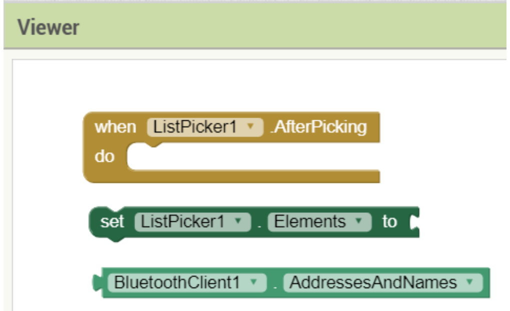
Blocks:

k. Select the Blocks tab. In the Blocks view window, select ListPicker as shown, then the Blocks menu allows you to add components.

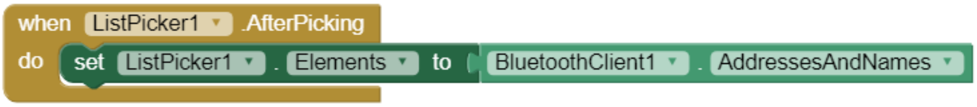




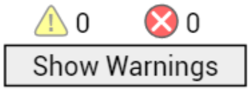
l. Select the following three components.


Viewer
when ListPicker1 .AfterPicking
do
set ListPicker1 . Elements to
BluetoothClient1 . AddressesAndNames

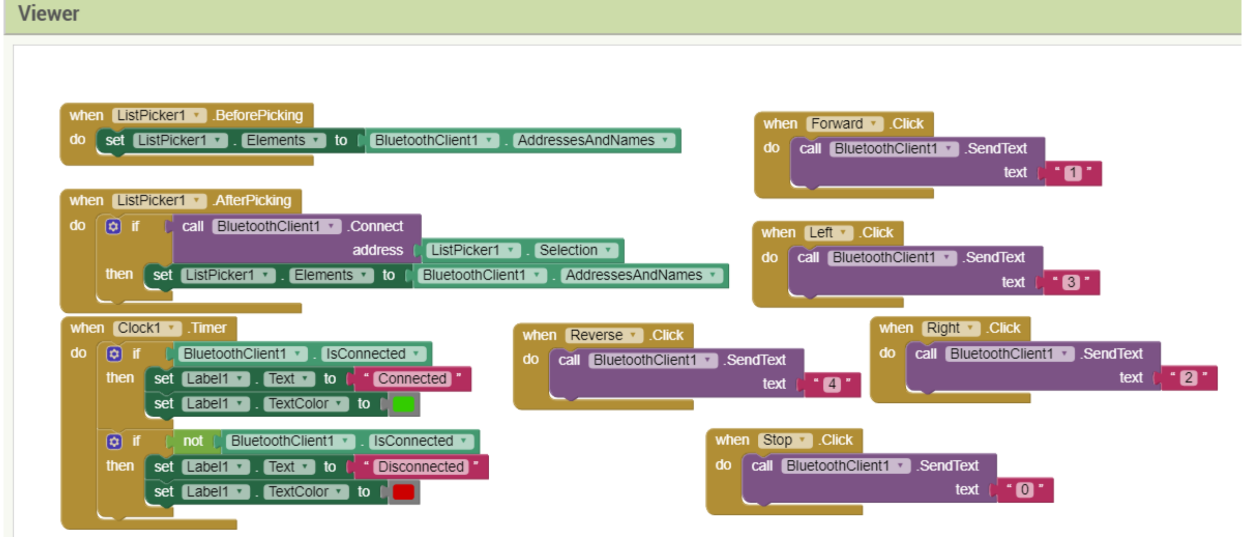
m. Connect the three components together as shown. Errors or warnings will not occur if the components are connected successfully.



```
when ListPicker1 .AfterPicking
do   set ListPicker1 . Elements to BluetoothClient1 . AddressesAndNames
```

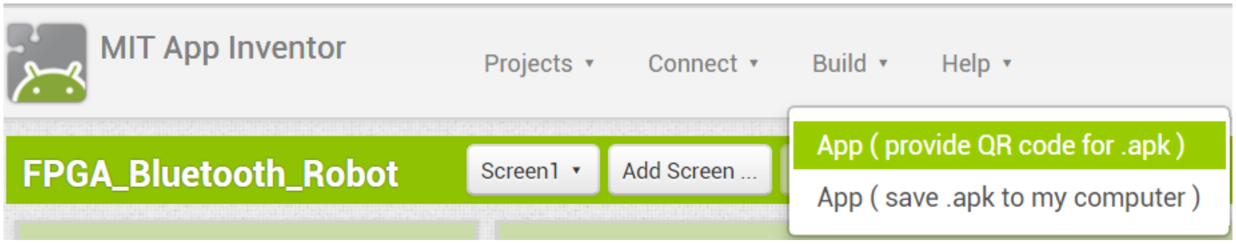


n. The final design for your Blocks windows should look like the picture below.



```
when ListPicker1 .BeforePicking
do   set ListPicker1 . Elements to BluetoothClient1 . AddressesAndNames

when ListPicker1 .AfterPicking
do   if   call BluetoothClient1 .Connect
               address ListPicker1 . Selection
     then set ListPicker1 . Elements to BluetoothClient1 . AddressesAndNames

when Clock1 .Timer
do   if   BluetoothClient1 . IsConnected
     then set Label1 . Text to " Connected "
          set Label1 . TextColor to 🟩
     if   not BluetoothClient1 . IsConnected
     then set Label1 . Text to " Disconnected "
          set Label1 . TextColor to 🟥

when Forward .Click
do   call BluetoothClient1 .SendText
          text " 1 "

when Left .Click
do   call BluetoothClient1 .SendText
          text " 3 "

when Reverse .Click
do   call BluetoothClient1 .SendText
          text " 4 "

when Right .Click
do   call BluetoothClient1 .SendText
          text " 2 "

when Stop .Click
do   call BluetoothClient1 .SendText
          text " 0 "
```

o. Build the project. Select Build tab and create your App to run on your Android device.



p. Follow the setup instructions (http://appinventor.mit.edu/explore/ai2/setup.html).

q. Download and install your app to your Android device.

r. Open your app can tab on the Connect button and find if the bluetooth module connected to the FPGA is detected by your Android device. If found, select the Bluetooth device, and you need to find out the MAC address of your BT to make sure you have connected to the right one.

s. Once your app is connected to the FPGA and the Bluetooth module, you can use the app to navigate the robot with the five buttons.