manting@mit.edu

What it does:

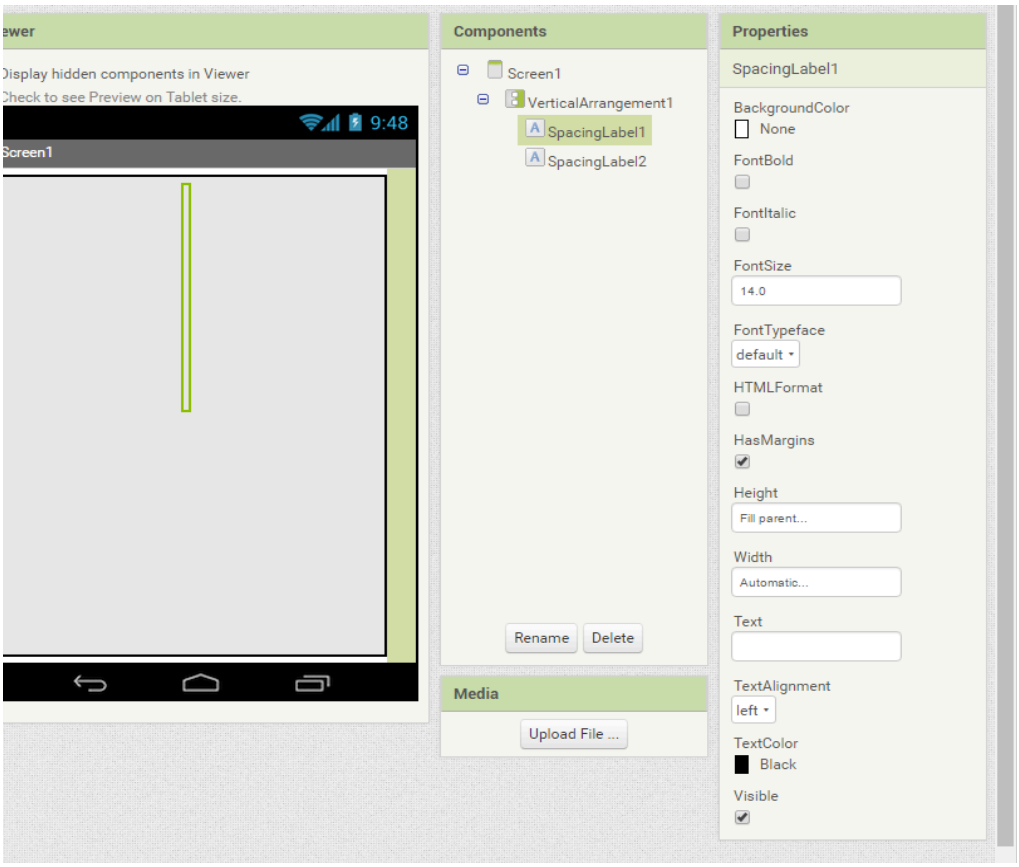
-
- Demonstration Video

Making the application:

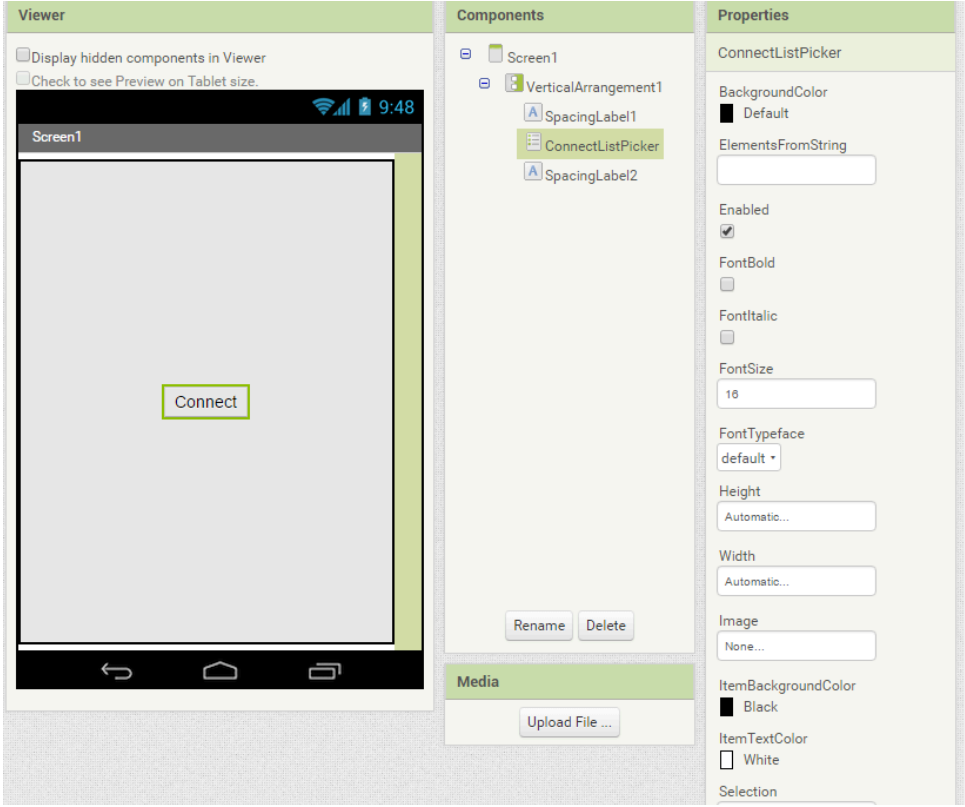
In the Designer:

1. Drag a **VerticalArrangement** from the Layout category onto the Designer. This will contain all of our components for the Color Searcher application.
    a. Be sure to change the **AlignHorizontal** and **AlignVertical** properties to “Center”.
    b. Change the **Height** and **Width** properties to “Fill Parent”.

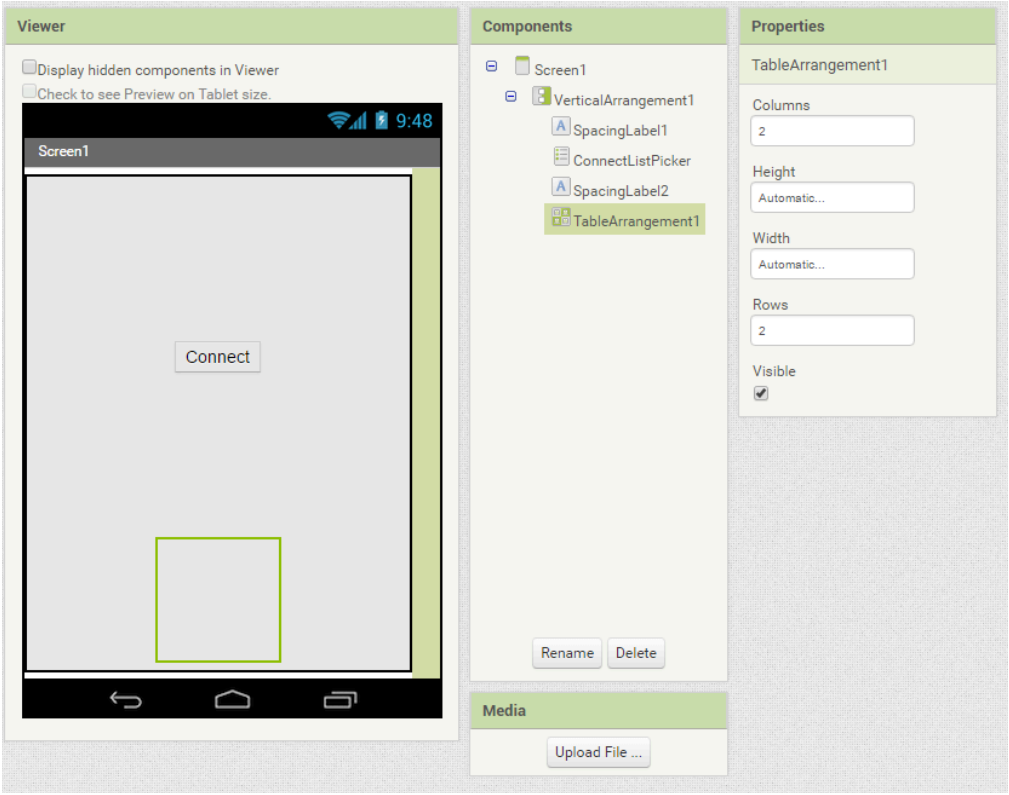
2. Next, drag two **Labels** onto the Designer space.
    a. The two labels will be used for spacing, so name accordingly, such as “SpacingLabel1,” which can be changed in the name property.
    b. Remove the text by deleting all the text under the **Text property** of each label.
    c. Change the **Height property** of both labels to “Fill Parent,” but do not change the **Width property**.



3. Next, drag a **ListPicker** onto the Designer. Place the ListPicker between the two Labels.
    a. Rename the ListPicker to “ConnectListPicker”.
    b. Change the text lto “Connect” in the ListPicker’s **Text property**.
    c. For this application, I changed the font to size 16 in the **Font property**.



4. Now drag a **TableArranagement** onto the Designer and placed it within the **Vertical Arrangement**, and below t properties are both “Automatic.”

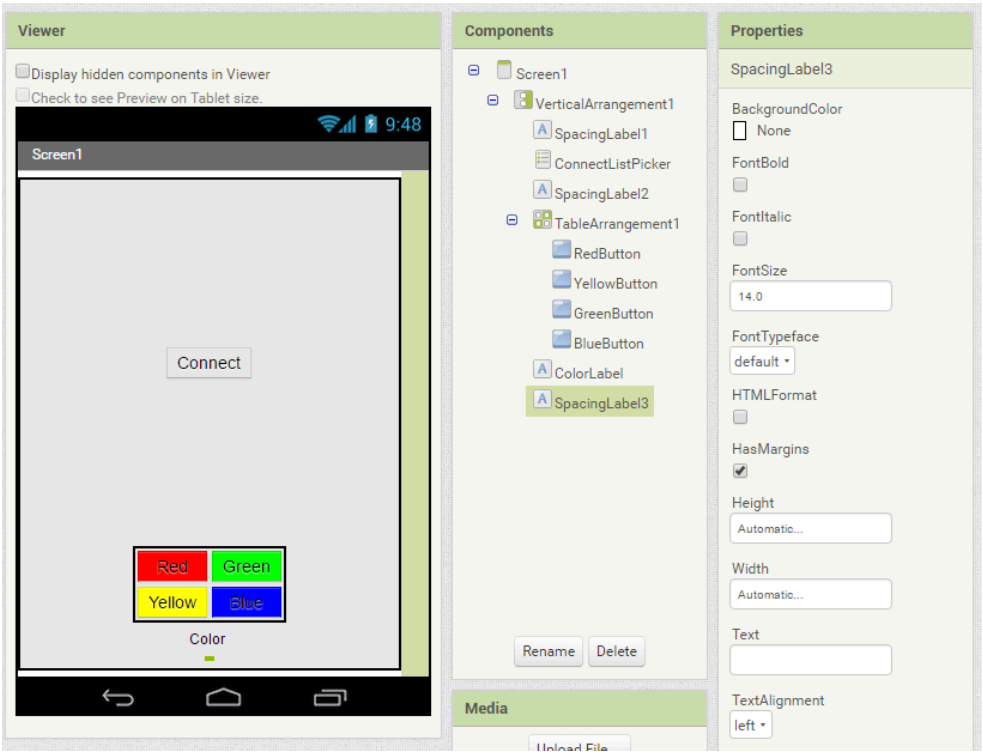


5. Drag four **Buttons** onto the Designer, within the **TableArranagement**. This will ensure that the four buttons are in a table arrangement for your application.
    a. These buttons will be labeled **RedButton, BlueButton, YellowButton, GreenButton**.

i. Change the text for these buttons to a specific color, and green.
ii. For this application, I changed the font size of all the buttons to 16.
iii. **Be sure to uncheck the Enabled checkbox for all of the color buttons.**

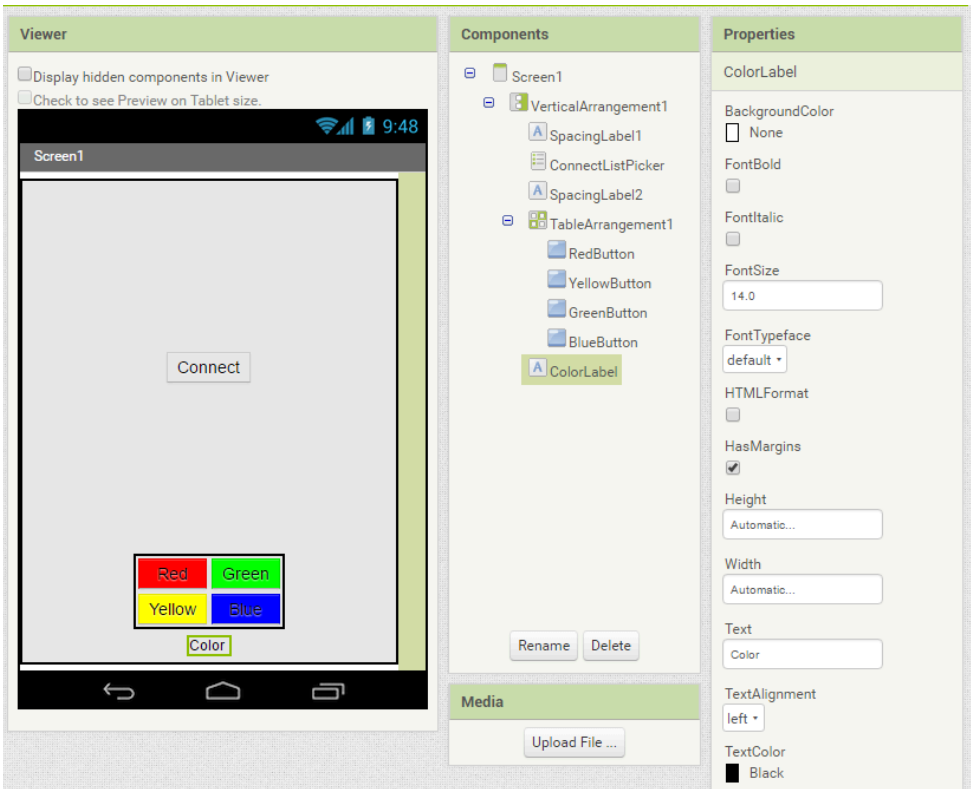
6. Drag a **Label** onto the Designer, within the vertical arrangement and below the table of four buttons that you have just created.
    a. Rename the label to “ColorLabel”.
    b. Change the text for the label to “Color”.

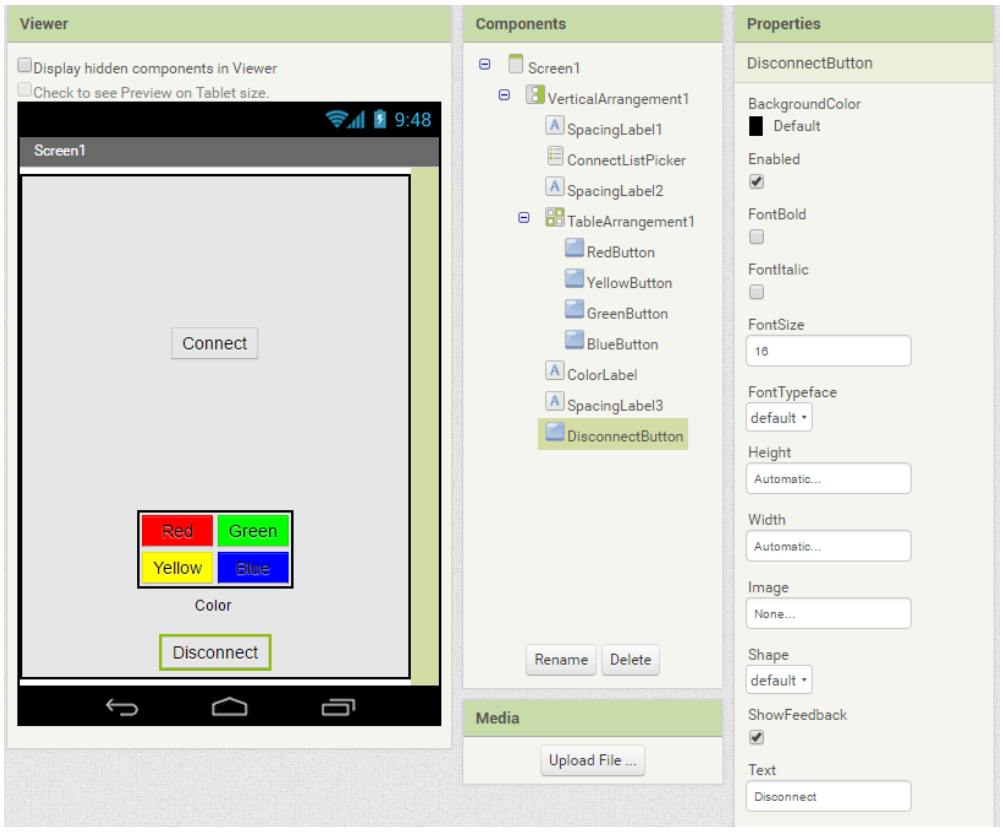
7. Drag another **Label** onto the Designer, now below the label created in Step 6.
    a. Delete all the text in the label, leaving it empty.
    b. Rename it as a spacing label, such as “SpacingLabel3”.



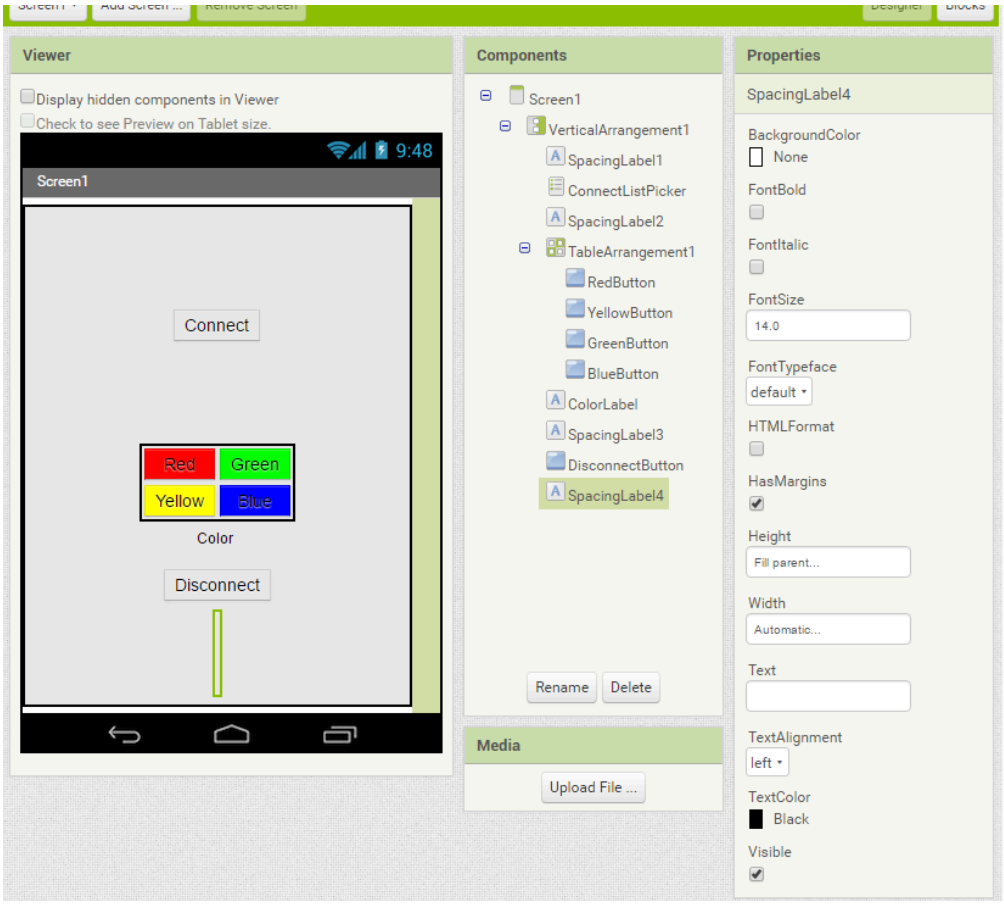
8. Drag a **Label** onto the Designer, within the vertical arrangement and below the table of four buttons that you have just created.
    a. Rename the label to “ColorLabel”.
    b. Change the text for the label to “Color”.

9. Drag another **Label** onto the Designer, now below the label created in Step 6.
    a. Delete all the text in the label, leaving it empty.
    b. Rename itc

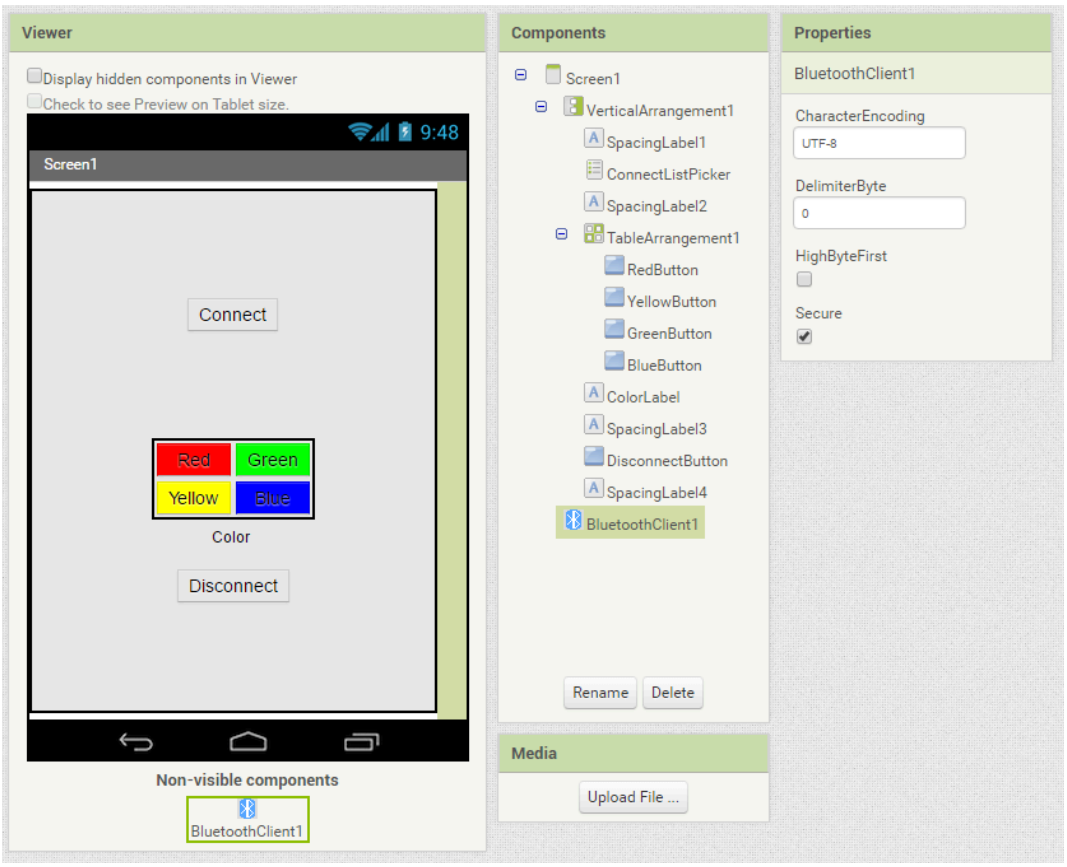




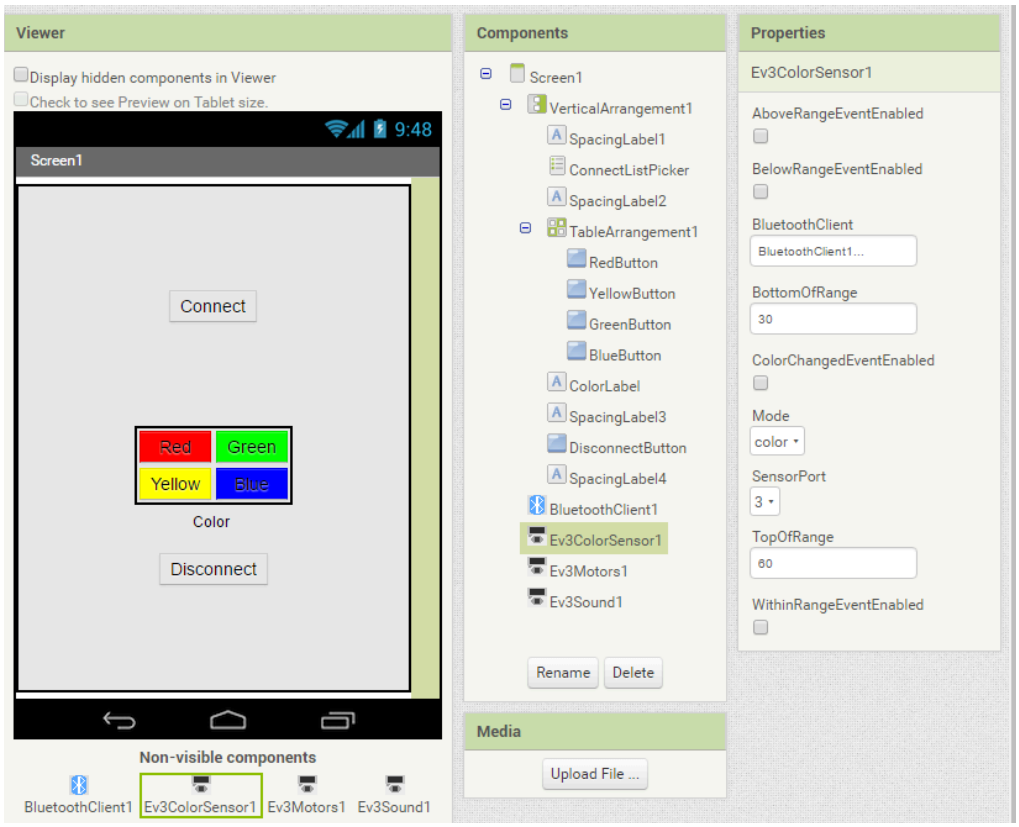
10. Drag another **Label** onto the Designer, below the button created in Step 8.
    a. Again, delete all the text in the label, leaving it empty.
    b. Rename it properly as a spacing label, such as “SpacingLabel4”.
    c. Change the **Height** property to “Fill Parent”, but leave the **Width** property as “Automatic”.



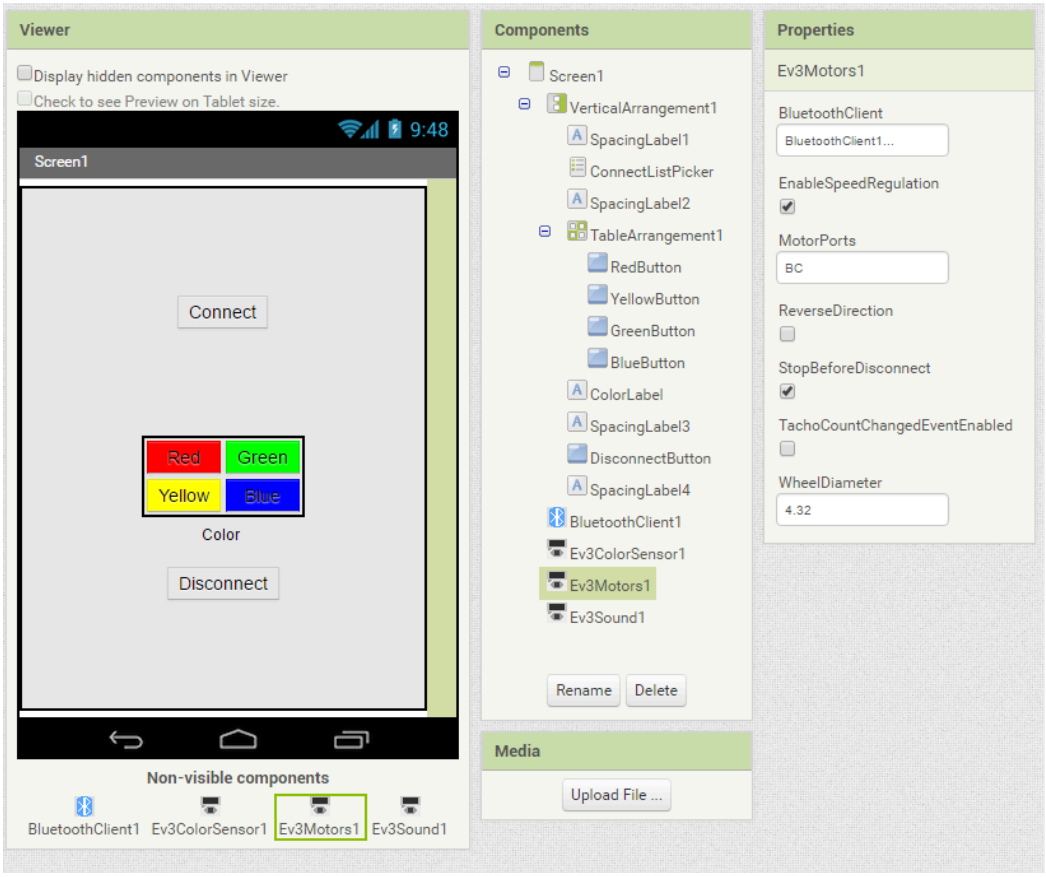
11. From the **Connectivity** category of the Palette, drag a **BluetoothClient** onto the Designer. It will be listed under the **Non-visible components**.



12. From the **LEGO MINDSTORMS** category of the Palette, drag a **Ev3ColorSensor**, **Ev3Motors**, and **Ev3Sound** onto the Designer. Again, it will be listed under the **Non-visible components**.
    a. Ev3ColorSensor
        i. Under the Ev3ColorSensor properties, change the BluetoothClient to the first option, **BluetoothClient1**.
        ii. Change the **Mode** property to “Color” and **SensorPort** property to 3.

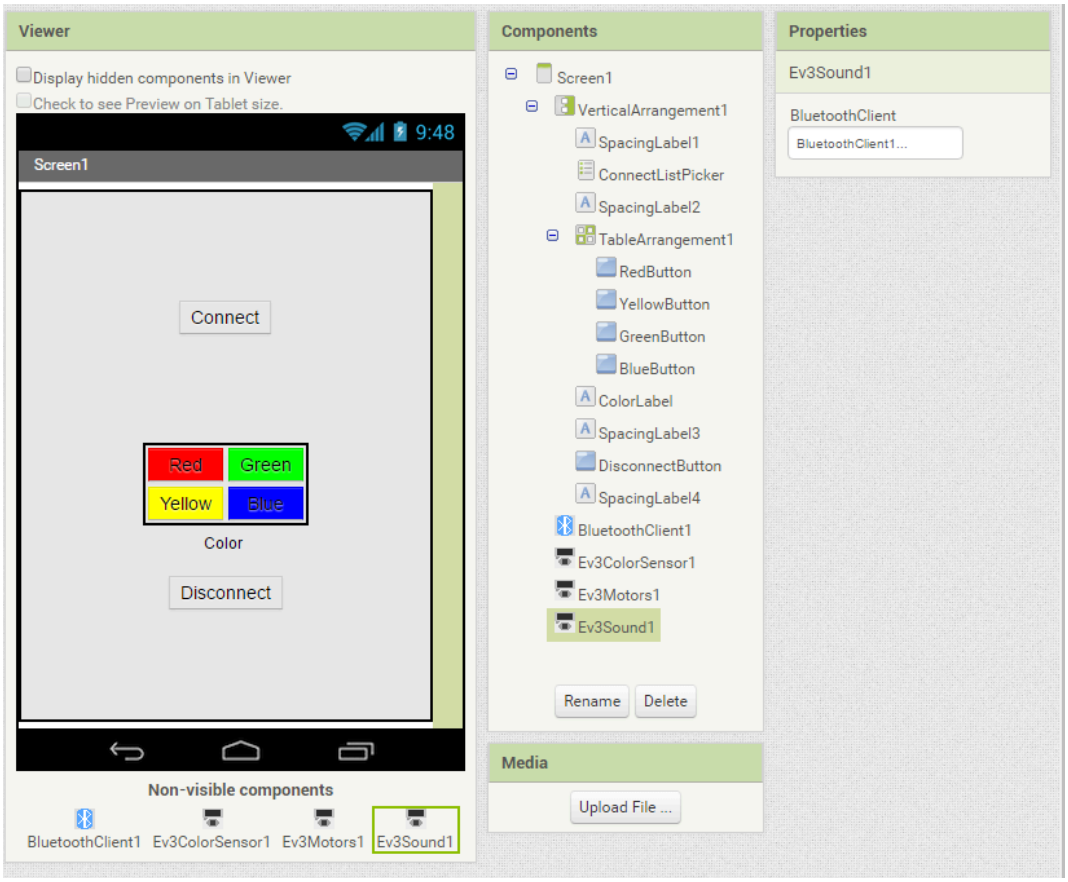


    b. Ev3Motors
        i. Under the Ev3Motors properties, change the BluetoothClient to the first option, **BluetoothClient1**.
        ii. Change the **MotorPorts** property to “BC”, or whichever ports your EV3 motors are connected to.



c. Ev3Sound

i. Under the Ev3Sound properties, change the BluetoothClient to the first option, **BluetoothClient1**.

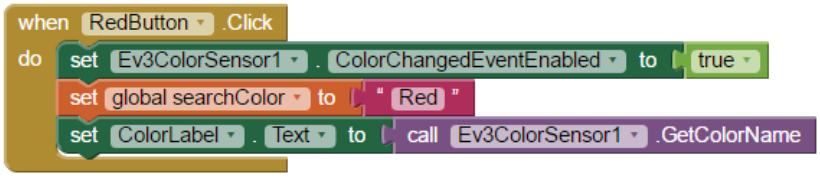


:

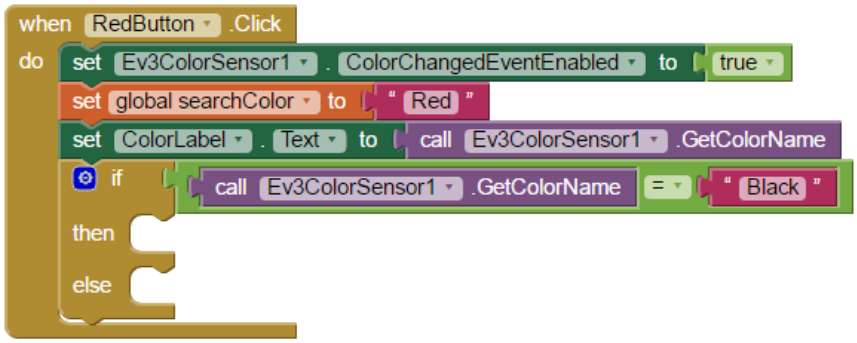
1. First, we will program the color buttons - Red, Blue, Yellow, Green buttons to instruct the robot to begin seeking for a specific color that the user chooses.
    a. Select a **Variable** block under Built-in Blocks. Drag out an "**initialize global name to**".
        i. Rename the variable by clicking on the "name" field and renaming it to "**searchColor**".
        ii. Set **searchColor** to " " by dragging out a text block and **not entering any text into the block**.

    b. Select the **RedButton** to access its drawer. Drag out the "**when RedButton.Click do**" block.
    c. Now select the **Ev3ColorSensor1** drawer. Drag out the "**set Ev3ColorSensor1.ColorChangedEventEnabled to**" block.
        i. Now drag out a **true** block from the **Logic** drawer underneath the Built-in blocks, and attach it to the block, making it set **Ev3ColorSensor1.ColorChangeEventEnabled to true**.
        ii. Place this block in the "**do**" section of the "**when RedButton.click do**" block.
    d. Drag out the "**set global searchColor to**" by hovering over the "**initialize global searchColor to block**" and set the block to "**Red**"

    e. Now select the **ColorLabel** drawer and drag out the "**set ColorLabel.Text to**" block.
        i. To complete the block, attach a "**call Ev3ColorSensor1.GetColorName**" block to the set **ColorLabel.Text** block. The "**call Ev3ColorSensor1.GetColorName**" block can be retrieved from the **Ev3ColorSensor1** Drawer.
        ii. Place this block in the "**do**" section of the "**when RedButton.click do**" block.

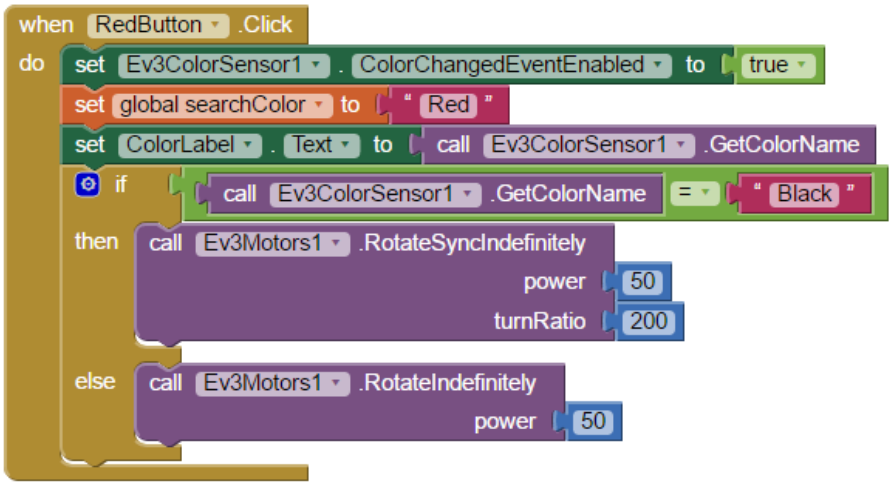


    f. Now drag out an "**if then**" block from the Control category of the Built-in blocks. Place it inside the "**when RedButton.Click do**" block.
        i. In the top left hand corner of the "**if then**" block**, click the blue area and drag an else block into the "if then" block on the right.** This will create an "**if then else**" block.
        ii. Drag out an "**=**" block from the **Logic** drawer under Built-in blocks.
            1. In the first slot of the "**=**" block, place a "**call Ev3ColorSensor1.GetColorName**" block, which can be found in the **Ev3ColorSensor1** drawer.
            2. In the second slot of the block, place a text block named "Black" by **dragging a text block and typing "Black" in the empty text field.**
            3. Place the completed block in the "**if**" socket of the "**if then else**" block.



iii. Drag out the "**call Ev3Motors1.RotateSyncIndefinitely**" from the **Ev3Motors1** drawer. Set **power to 50** and **turnRatio to 200** using the **Math** block under Built-in Blocks. Place this block in the "**then**" section of the "**if then else**" block.

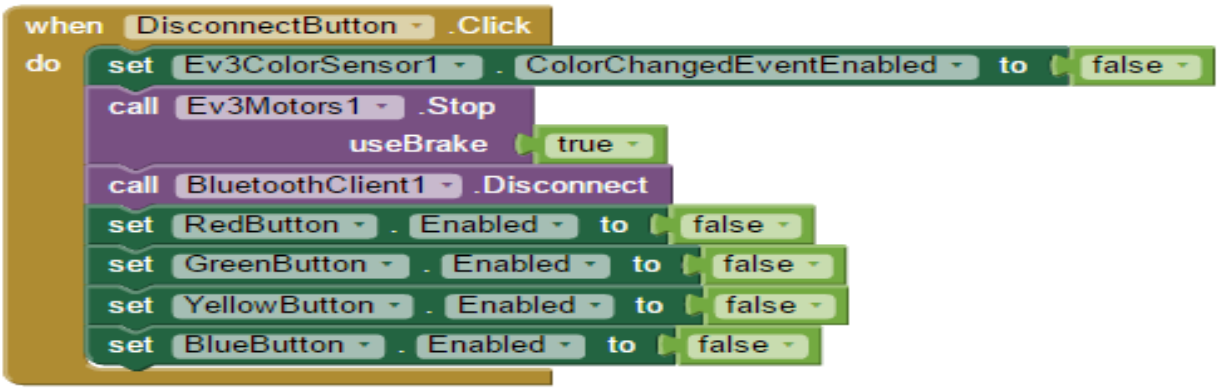
iv. Drag out the "**call Ev3Motors1.RotateIndefinitely**" Block from the **Ev3Motors1** drawer. Set **power to 50**. Place this block in the "**else**" section of the "**if then else**" block.



g. Repeat this process for the rest of the buttons (Yellow, Blue, and Green). The only difference between the code blocks for the color is that the "**when do**" block will refer to a **specific color** and the "**set global searchColor to**" block should be that **specific color as well.**

2. Now we will program the Connect ListPicker.
   a. Drag out a "**when Connect ListPicker.BeforePicking do**" block from the Connect ListPicker drawer.

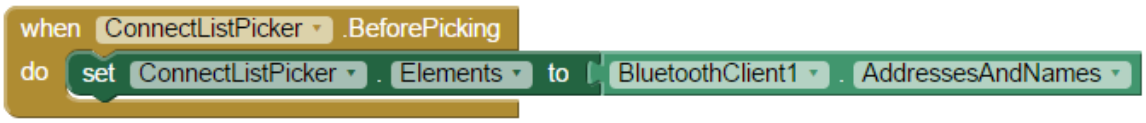
i. Drag out a "**set Connect ListPicker.Elements to**" block from the same drawer.



ii. Drag out a "**BluetoothClient 1.AddressesAndNames**" block from the BluetoothClient drawer.

iii. Connect the "**Bluetooth Client1.Addresses Names**" block to the "**set ConnectListPicker.Elements to**".

iv. Place this completed block under the "**do**" section of the "**when ConnectListPicker.BeforePicking do**" block.



b. Drag out a "**when ConnectListPicker.AfterPicking do**" block from the **ConnectListPicker** drawer.

i. Drag out an "**if then**" block from the Control drawer under Built-in blocks.

ii. Drag out a "**call BluetoothClient1.Connect**" block. Also, drag out a "**ConnectListPicker.Selection**" block from the **ConnectListPicker** drawer.

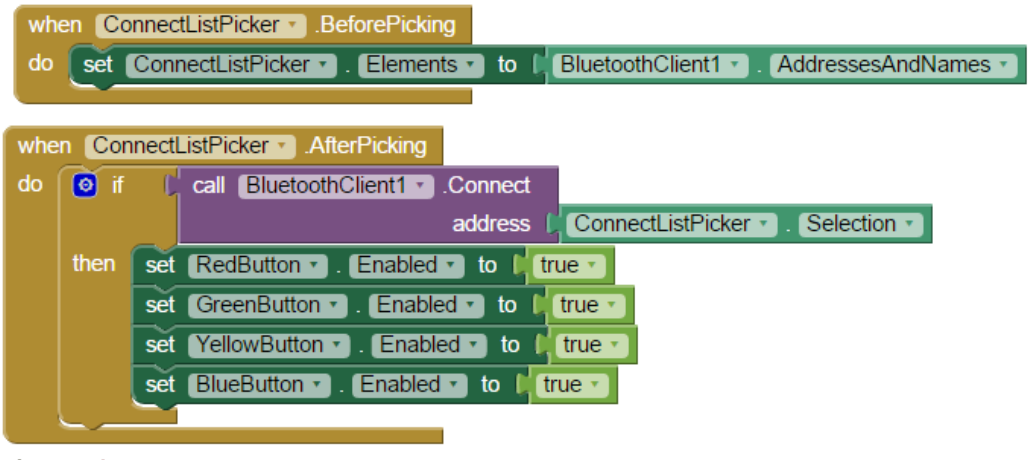
iii. Connect the "**call BluetoothClient1.ConnectAddress**" block with the "**ConnectListPicker.Selection**" block, and place the completed block in the "**if**" socket of the "**if then**" block.

iv. Drag out the "**set RedButton.Enabled to**" block. Complete the block by placing a "**true**" block from the Logic draw under Built-in blocks in the "**to**" socket.

1. Do this for GreenButton, BlueButton, and YellowButton by copying and pasting the RedButton blocks and selecting the different buttons on the dropdown menu.

v. Place all the blocks (RedButttton, GreenButton, YellowButton, BlueButton) in the "**then**" socket of the "**if then**" block.

vi. The final code block for this section should look like this.

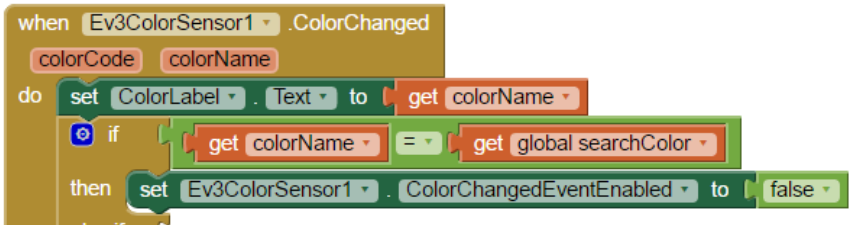


3. We now will have to program for the event when the robot senses a change in color.

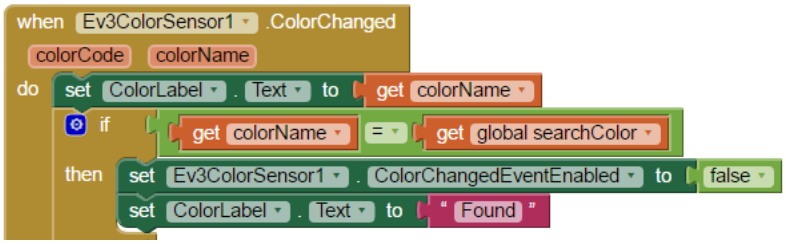
4. Drag out the when "**Ev3ColorSensor1.ColorChanged do**" block from the **Ev3ColorSensor1** drawer.
    a. Drag out the "**set ColorLabel.Text to**" block from the **ColorLabel** drawer. Hover over the **colorName** in the "**when Ev3ColorSensor1.ColorChanged do**" block to drag out the "**get colorName**" block. Connect these two blocks and place the completed block in the "**do**" section.

    b. Drag out the "**if then**" block from the **Control** drawer from the Built-in blocks, and click on the blue area on the top left corner of the block. First, drag the "**else if**" block under the "**if**" block to the right. After this is done, drag the "**else**" block underneath the "**else if**" block to create an "**if else if**" block.
        i. Drag out an "**=**" block from the **Logic** drawer under Built-in blocks. Place a "**get colorName**" block in the first slot of the "**=**" block and a "**get global searchColor**" block in the second slot. Place this completed block in the "**if**" socket.

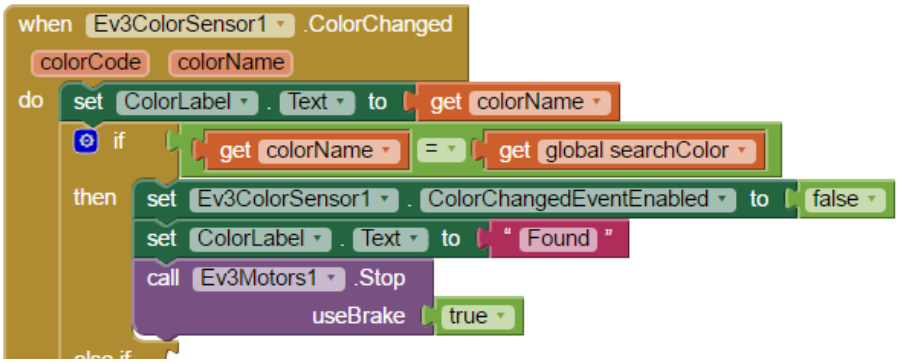
    c. Drag out the "**Ev3ColorSensor1.ColorChangedEventEnabled to**" block and place a "**false**" block from the **Logic** drawer to complete the block. Place the block in the "**then**" under the "**if**" section.



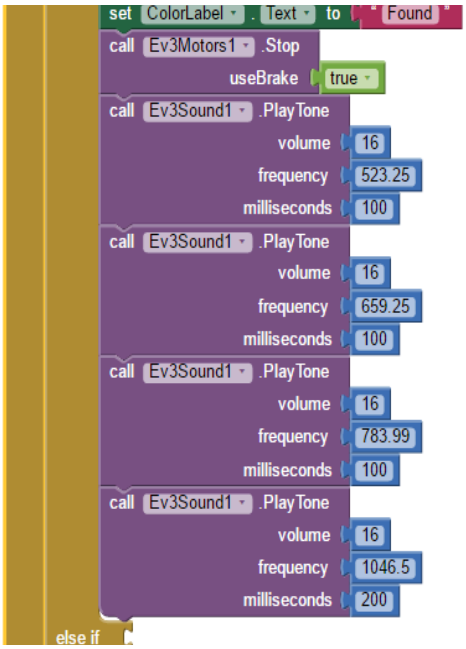
d. Drag out the "**set ColorLabel.Text to**" block from the **ColorLabel** drawer and place a Text block "Found" from the **Text** drawer in the "**to**" socket. Place the completed block under the previous ones.



e. Drag out the "**call Ev3Motors1.Stop**" block and place a "**true**" block from the Logic drawer in the "**useBrake**" socket. Place the completed block under the previous ones.

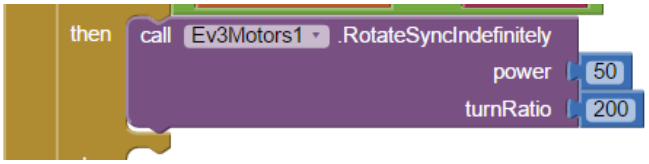


f. Drag out the "**call Ev3Sound1.PlayTone**" block from the **Ev3Sound1** drawer. **Copy and paste the block four times.** Fill the **volume** socket for all four blocks with **16** (obtained from Integer drawer). Fill the **milliseconds** socket for the first three blocks with **100** and for the final block, fill the **milliseconds** socket with **200**. Each of the four "**call Ev3Sound1.PlayTone"** blocks will play a different frequency. The frequency are as follows: **523.25, 659.25, 783.99, 1046.5**.

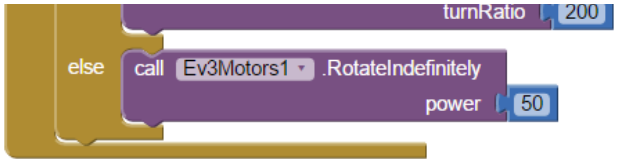


g. Drag out another "**=**" block from the Logic drawer. Place the "**get colorName**" block in the first slot and the **Text** block "Black" in the second slot. Put the completed block in the "**else if**" socket.

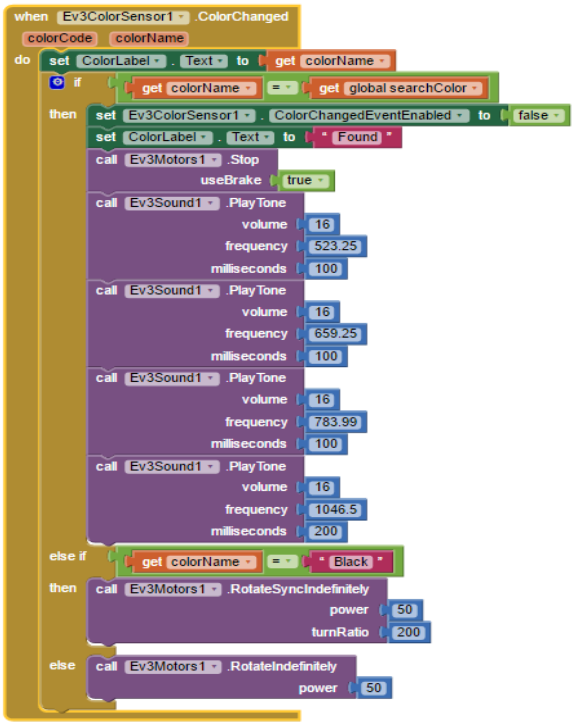
h. Drag out the "**call Ev3Motors1.RotateSyncIndefinitely**" block from the **Ev3Motors1** drawer, setting the **power to 50** and **turnRatio to 200**. Place the completed block in the "then" section after the "else if".



i. Drag out the "**call Ev3Motors1.RotateIndefinitely**" block from the **Ev3Motors1** drawer, setting the **power to 50**. Place the completed block in the "**else**" section under the "**else if**" section of the block.



j. The final block of code for this section should look like this.



5. Finally, we will program the Disco nnectButton for the application.
    a. Drag out the "**when DisconnectButton.Click do**" from the **DisconnectButton** drawer.
    b. **Ev3ColorSensor1** drawer and a "**false**" block from the **Logic** drawer. Complete the block by placing the false block in the "**to**" socket of the "**set** Drag out the "**set Ev3ColorSensor1.ColorChangedEventEnabled to**" block from the **Ev3ColorSensor1.ColorChangedEventEnabled to**" block. Place this completed block in "**do**" section of the "**when DisconnectButton.Click do**" block.

    c. Drag out the "**call Ev3Motors1.Stop useBrake**" block from the **Ev3Motors1** drawer and a "**true**" block from the **Logic** drawer. Complete the block by placing the "**true**" block in the "**useBrake**" socket. Place the complete block in the "**do**" section of the "**when DisconnectButton.Click do**" block.

S-----

    d. Drag out the "**call BluetoothClient1.Disconnect**" block from the **BluetoothClient1.Disconnect**. Place this block in the "**do**" section of the "**when DisconnectButton.Click do**" block.

Once pairing is complete, open the EV3_Demo app

    e. Drag out the "**RedButton.Enabled to**" block from the **RedButton** drawer and complete the block by placing a "**false**" block in the "**to**" socket.
        i. Do the same for the rest of the buttons (YellowButton, GreenButton, BlueButton). A quick way to do this is by copying and pasting the RedButton block that has been created and changing RedButton to YellowButton, GreenButton, and BlueButton using the dropdown menu.

f. The final block of code for this section should look like this.

6. The full entire code blocks should look like this.

AIA and APK files:

- **APK Files**
- **AIA Files**

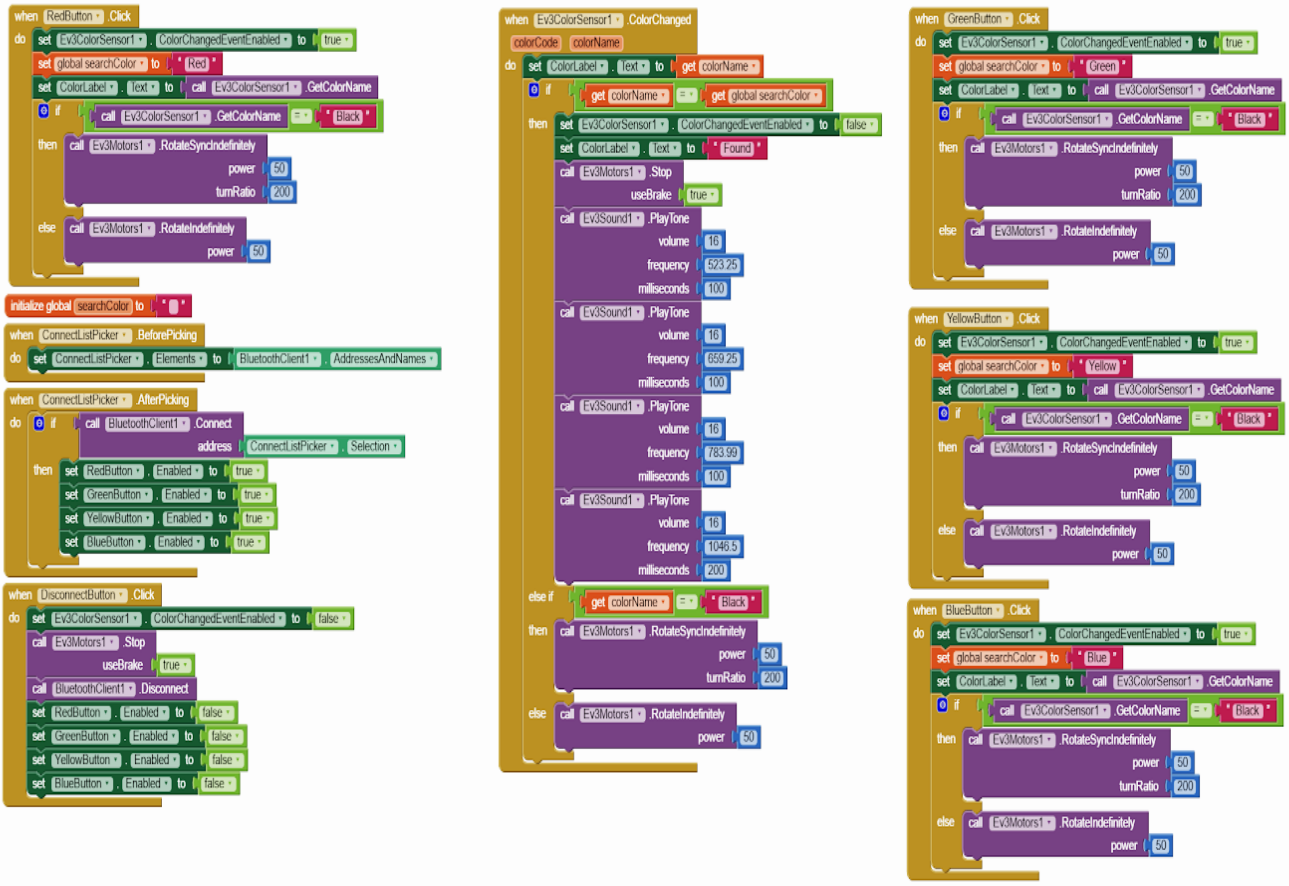
Instructions for running the application:

- Make sure the motors are plugged into ports B and C on the EV3 Brick.
    - If you want the motors in other ports, you can change the “MotorPorts” property in the Designer for the EV3Motors component.
- Turn on Bluetooth for your phone/tablet
- On the LEGO EV3 Brick, make sure Bluetooth is turned on AND iPhone/iPad/iPod is off for the EV3 ("Settings")
- Pair tablet with EV3 (passcode is probably 1234 but will show up on the EV3 Brick screen)
- Once pairing is complete, open the
- Click "Connect" and connect with the EV3
- Pick a color!

Notes:

- My EV3 sometimes incorrectly identified Green and Blue. Red worked with very high fidelity

- The searching/turning



at Black algorithm can be refined to perform more optimally as a challenge :)