Introduction to App Development - Lesson 3

Conditionals & Lists

Let's review some quick tips

- Open your HelloRyerson Project from last lecture.
- If you do not have the project, you can import it after downloading it from the course website:

- caltoc.scs.ryerson.ca

Testing Functions

- You can make your phone execute individual function calls without waiting for an event
- Try it, build or select the following block:

Then, right-click the purple block (not on the light part) and select "**Do It**". Your phone must be connected for this to work.

Testing functions:

- Keep in mind that testing some functions may change the state of your app:

- This might be beneficial, as you may want to reset some variables without having to reset your app
  - This was even more important for last iteration as they were using emulators.

Just do it!

- Drag the following block, right click it and select "Do It"
- What is the result?

*When debugging, don't let your dreams be dreams

Testing values

- You can also test values
  - You also need your phone connected for this to work
- When testing values, the result gets added to the comment for that block.
- You can safely delete the comment when you are done with it.

Use these tools to help you identify errors in your logic!

Moving on..

Are you ready to start?

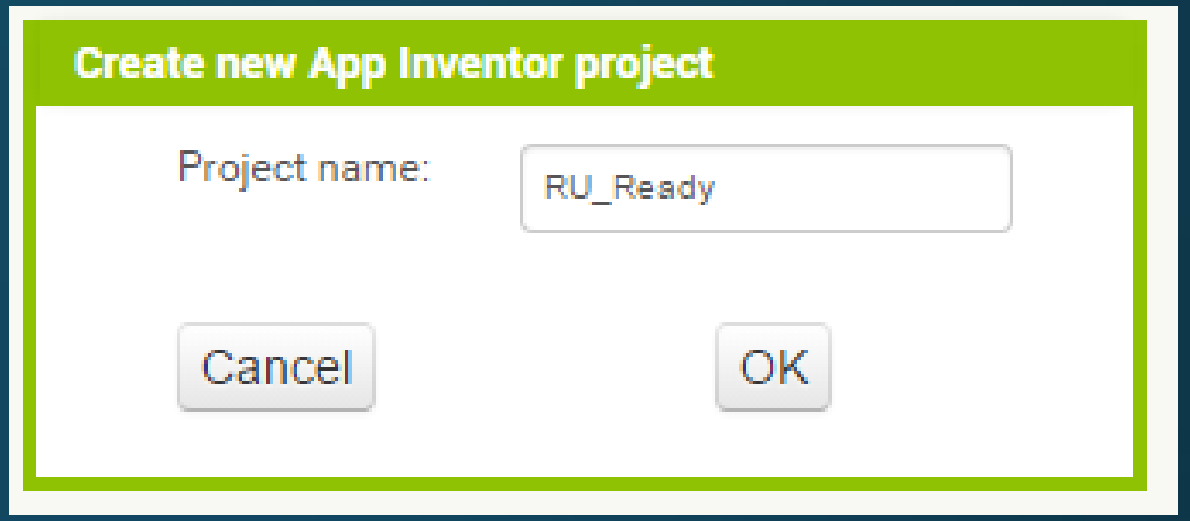
- If the class is ready to start today's lecture, go to the next slide.
- Otherwise, Stay on this slide and wait for the class to be ready.

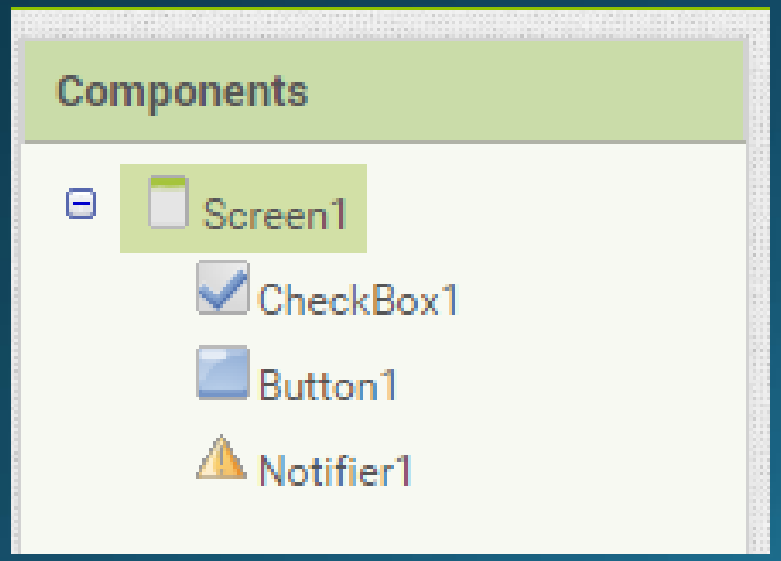
Wouldn't it be great to have an app for this?

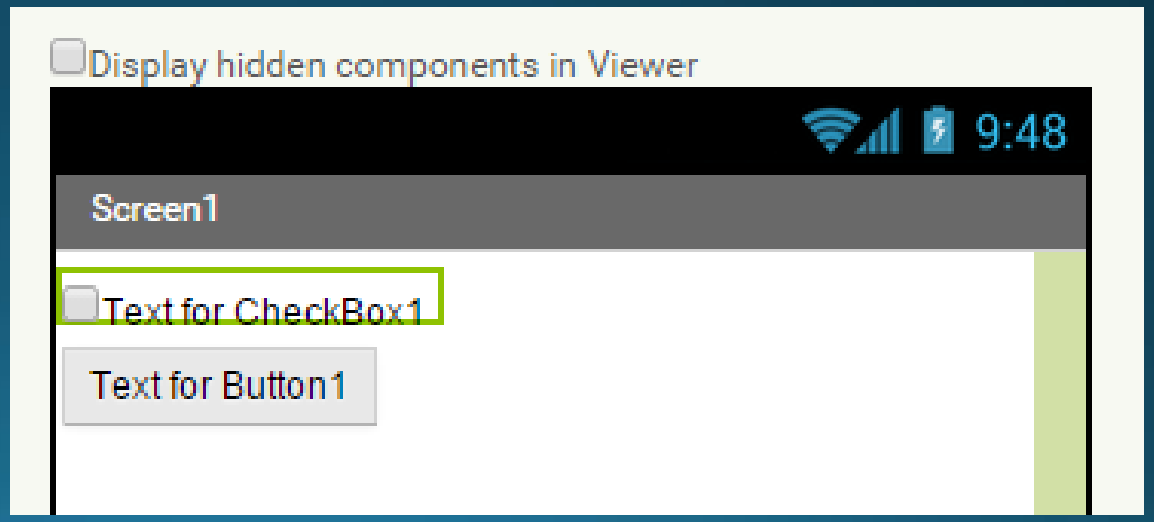
- We can use this app to warm-up for today's lesson!

Let's make it!

- **Start a new Project**
- **Drag in the following components:**



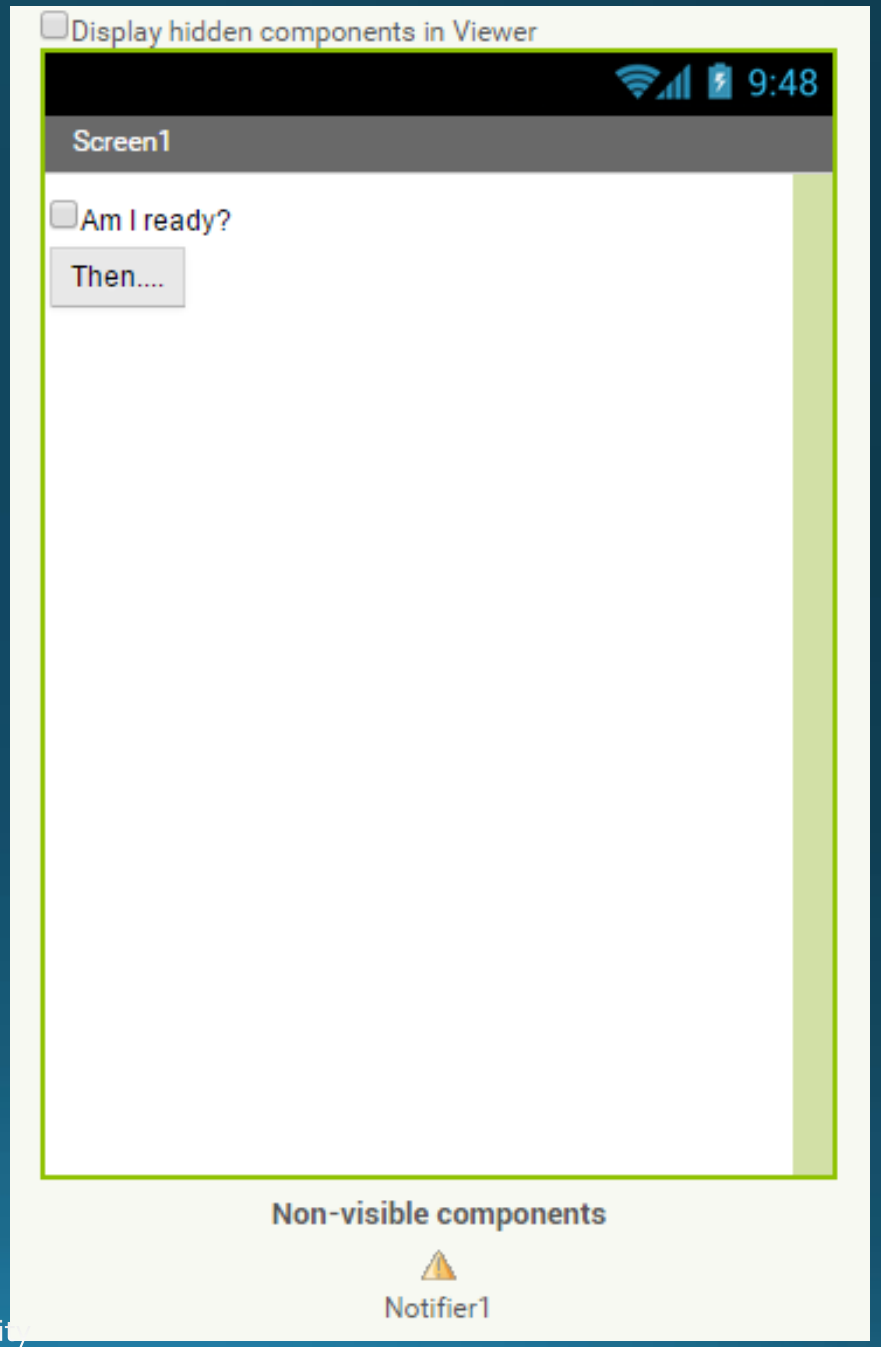




First: Layout

- Change the text on the **Checkbox** to read the question:
  - Am I ready?

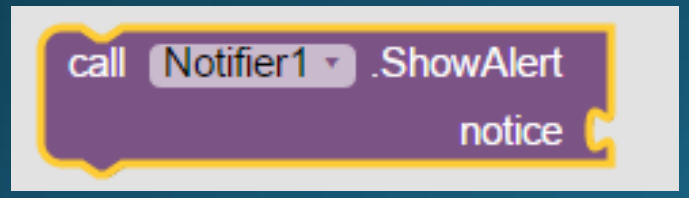
- Change the text on the **Button** to say:
  - Then....

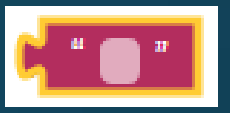


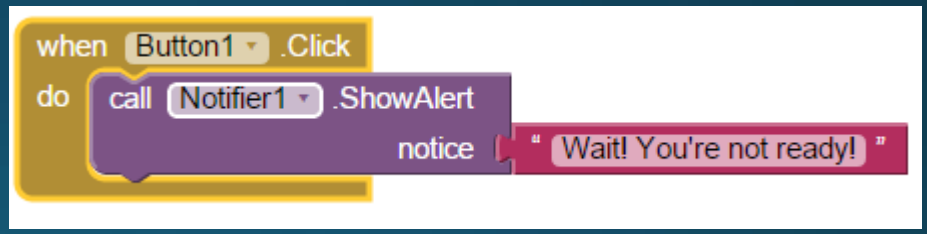
Blocks!

- Switch to block view.
- Drag in a

  from the button component
- Drag in a



- Drag in a  block from the Text menu.
  - Set the text to:
    - Wait! You're not ready yet!



- Test your app.

Why aren't we ready?

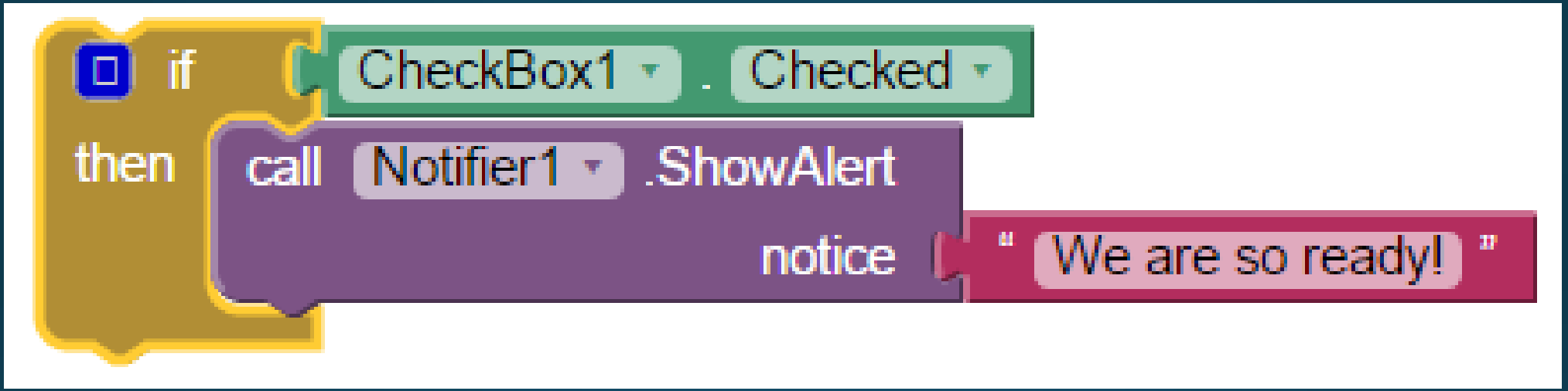
We need to learn conditionals!

if block

Concept #6

- Drag in an if block from the Control menu.

- Drag in the CheckBox1.Checked property block from the Checkbox1 component.

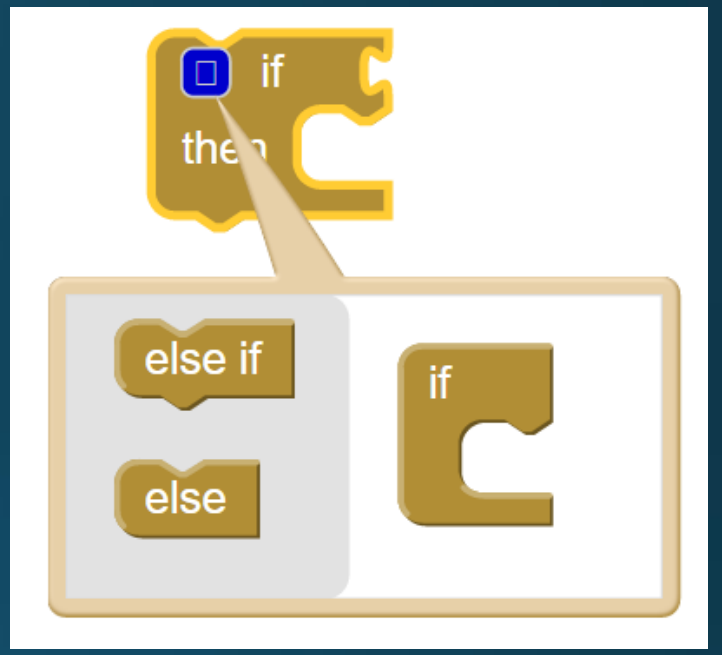


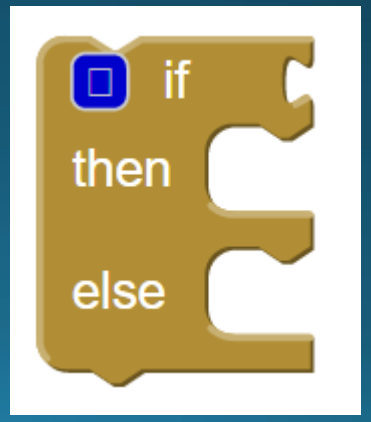
- What does this code do?
- What happens if the CheckBox is not checked?

if-else block

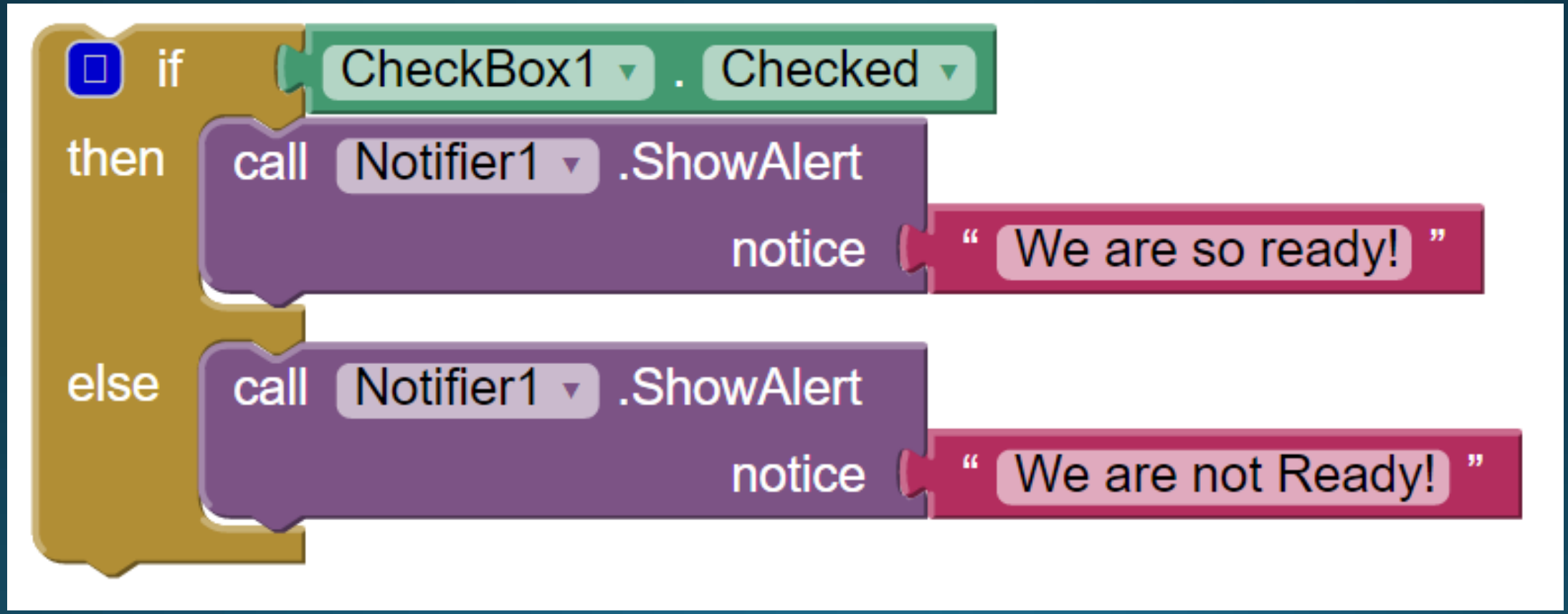
- On the if block, click the blue box.
- Inside the popup pane, drag an else block into the slot.

- You have just changed the structure of your if block!



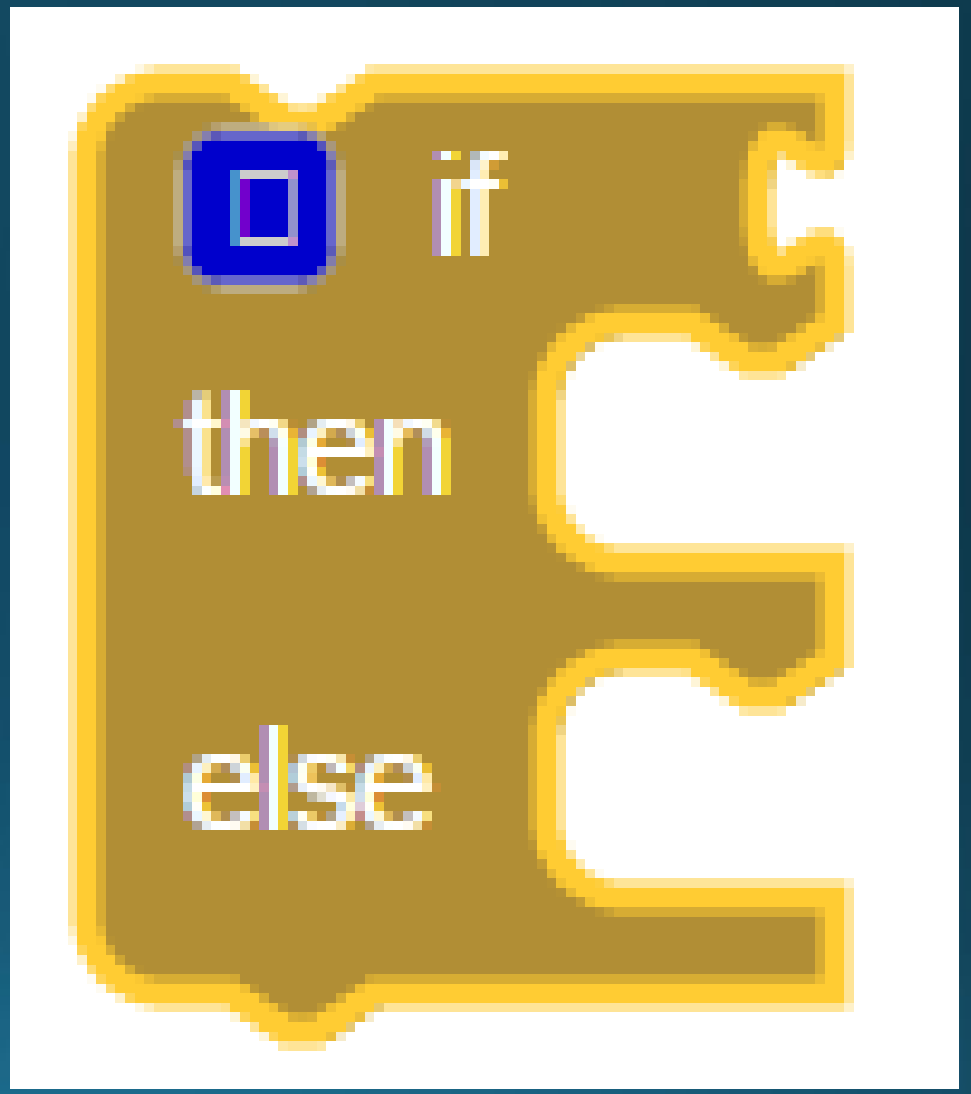


How did the app change?

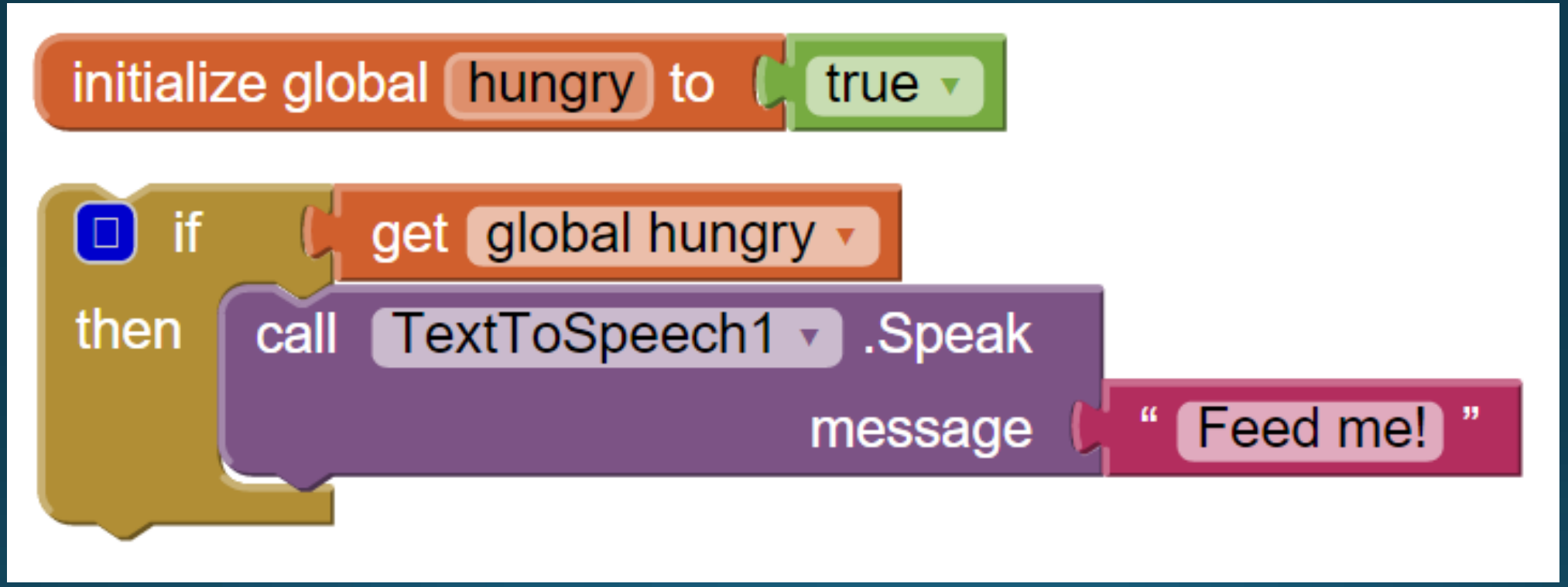


Conditionals!

- A conditional is the way that your app will make decisions
- Your app asks a question and chooses how to act based on the answer
  - If the answer is *true*, the code inside the then slot is run
  - Otherwise (when the answer is *false*), the code in the else block is run.

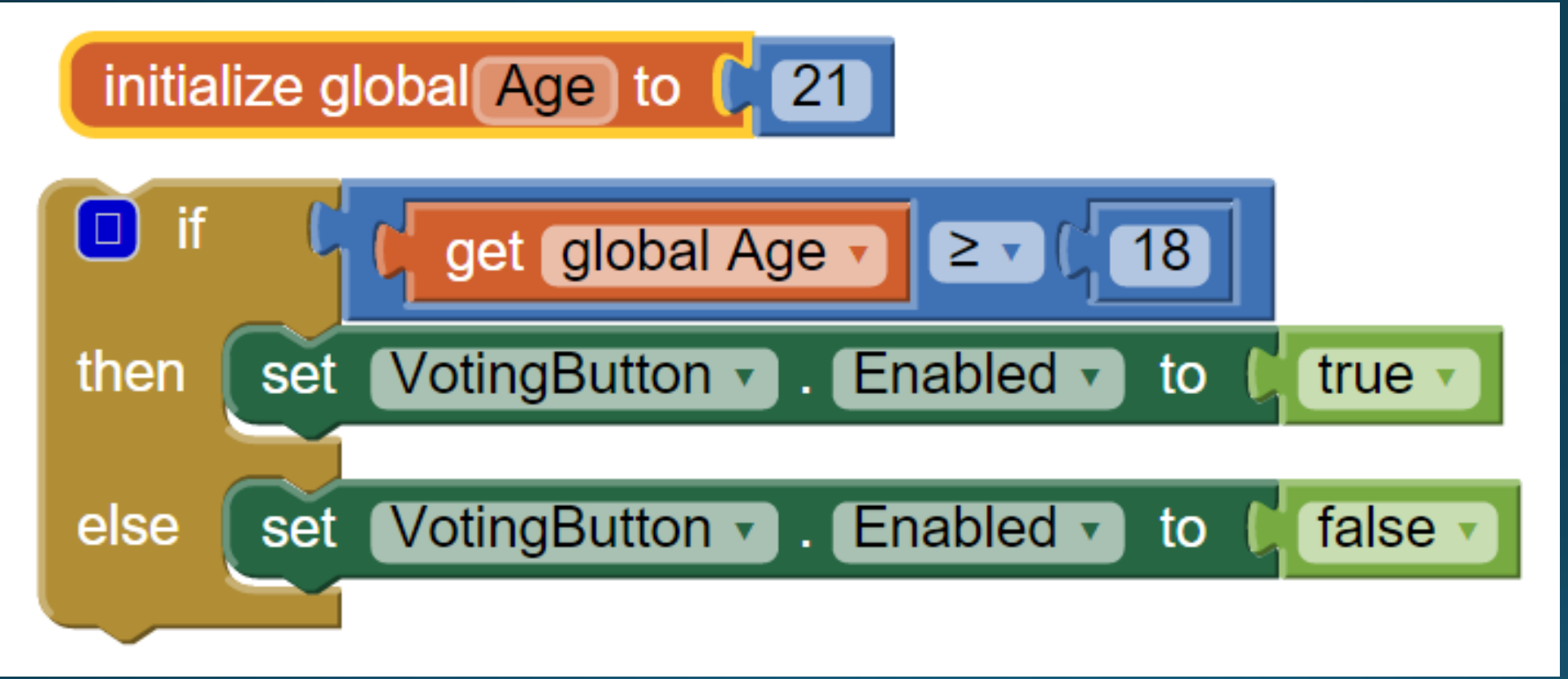


Examples



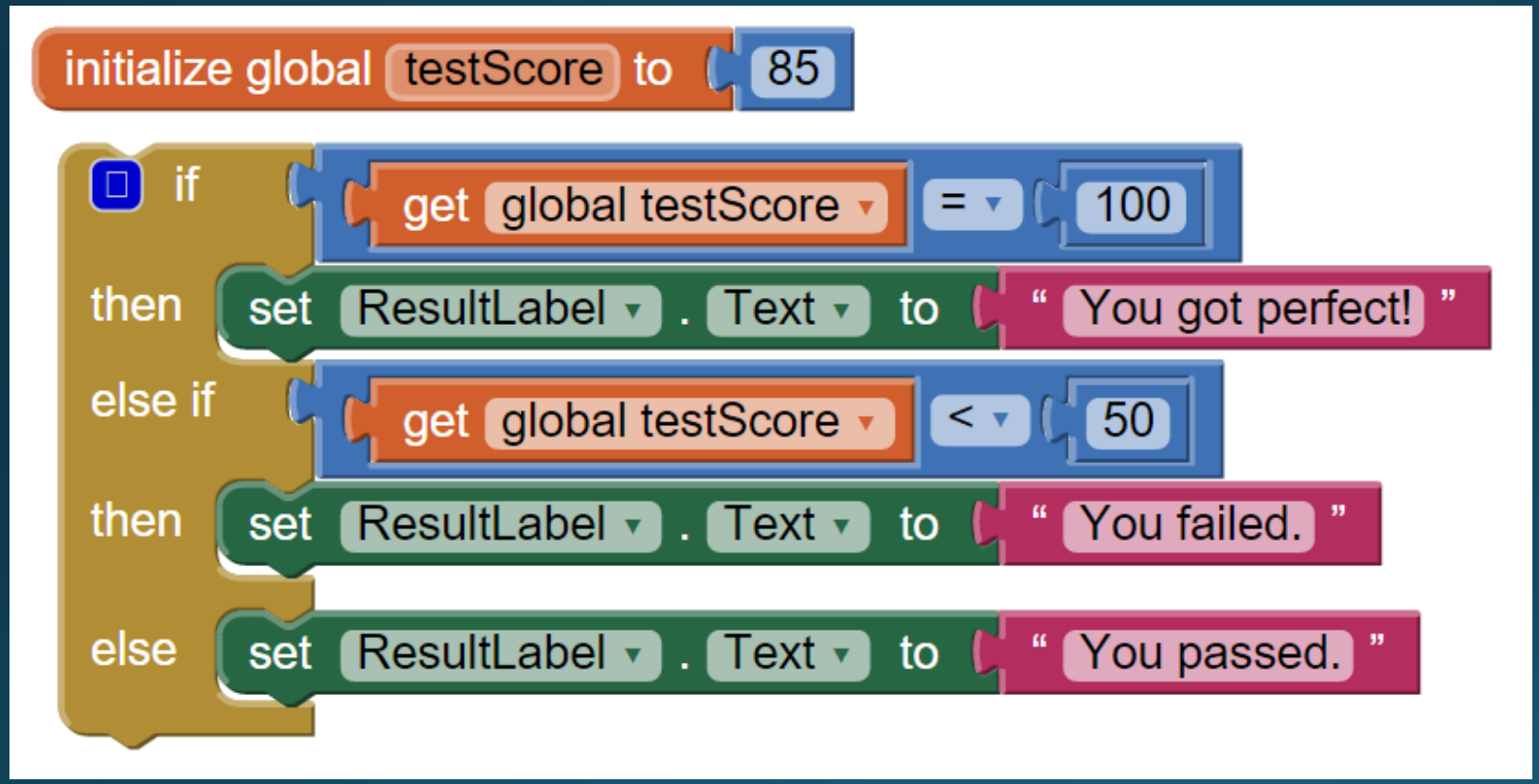
What will happen in this example?

Examples



What about this example?

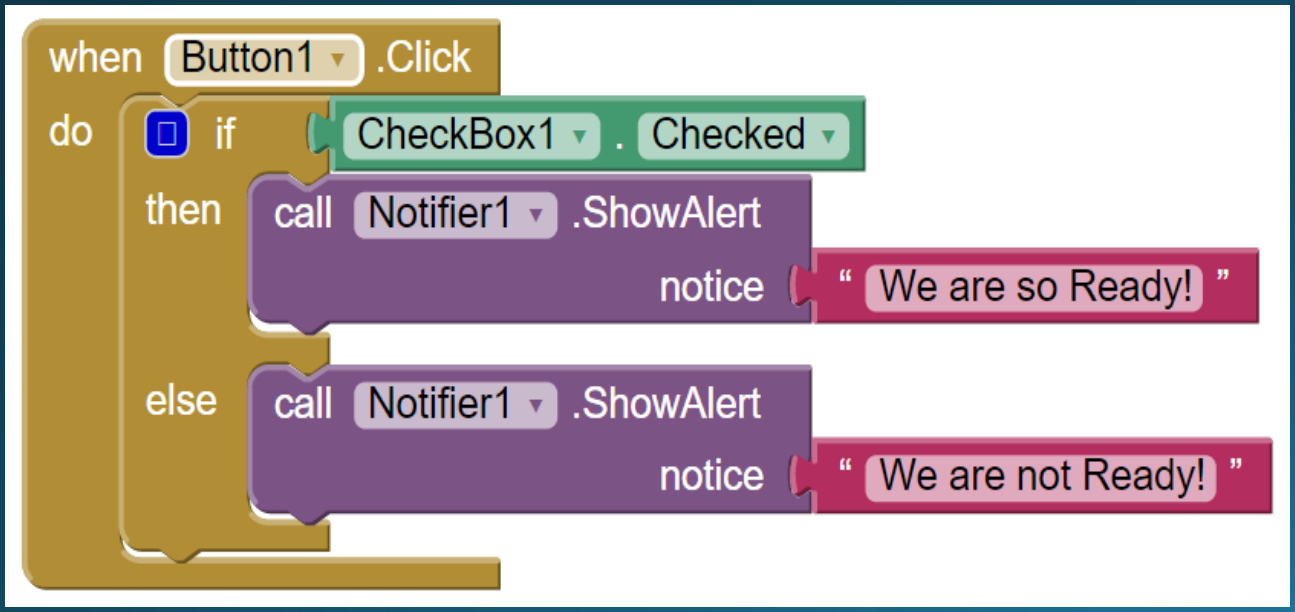
Examples

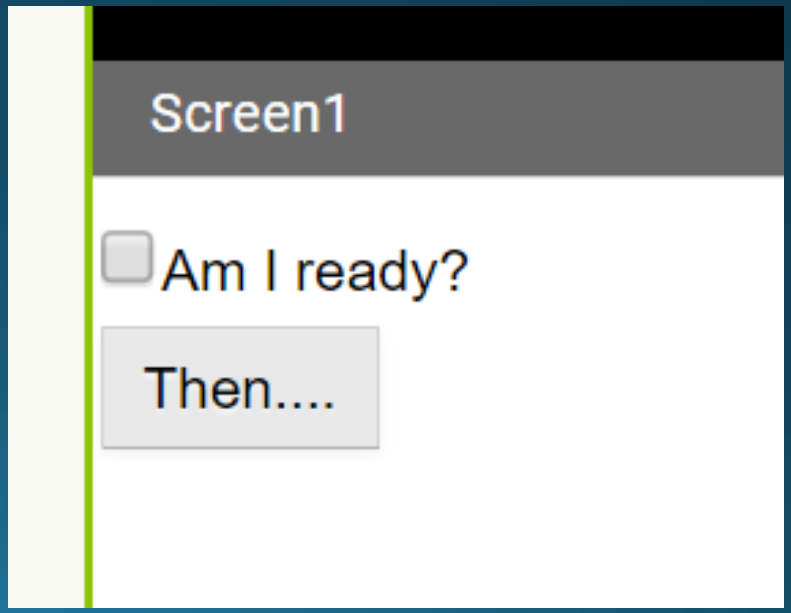


You can optionally use an 'else if' to chain many conditions together.

Back to our app…

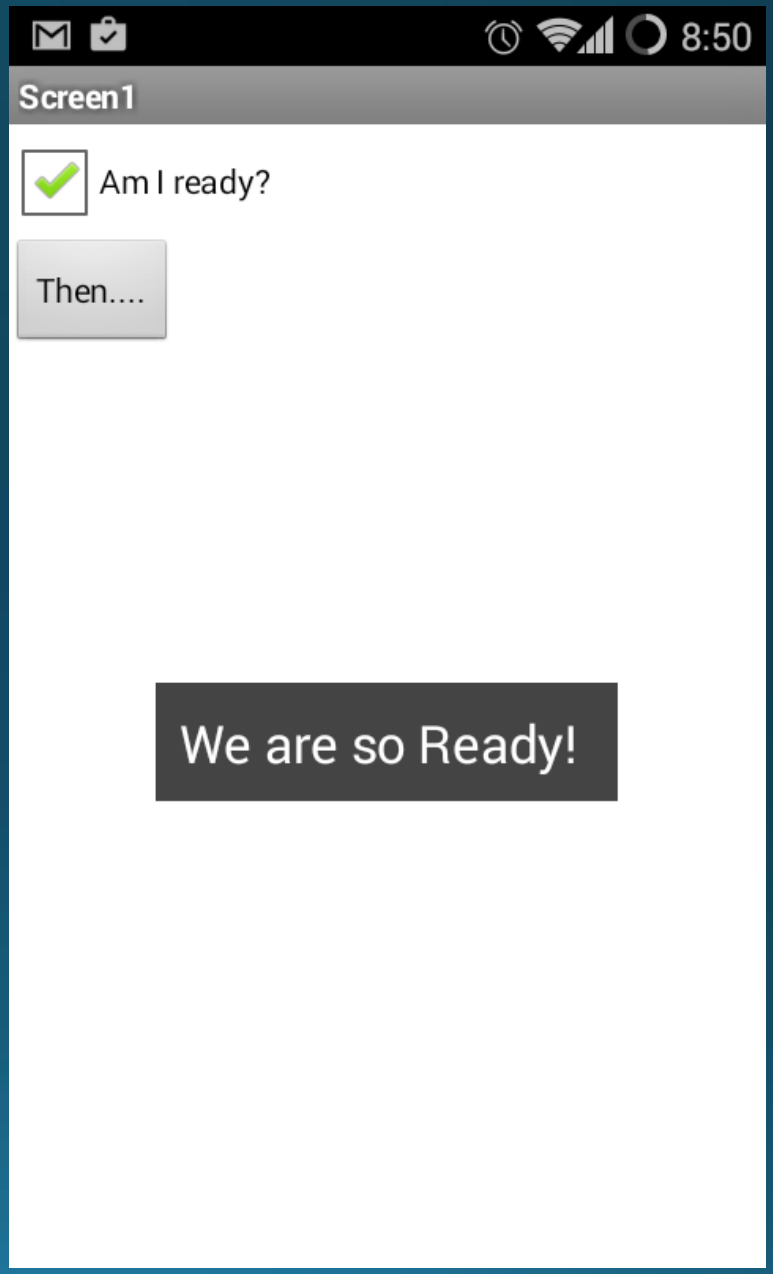
- Complete the if block.
- Place it inside the Button1 click event.





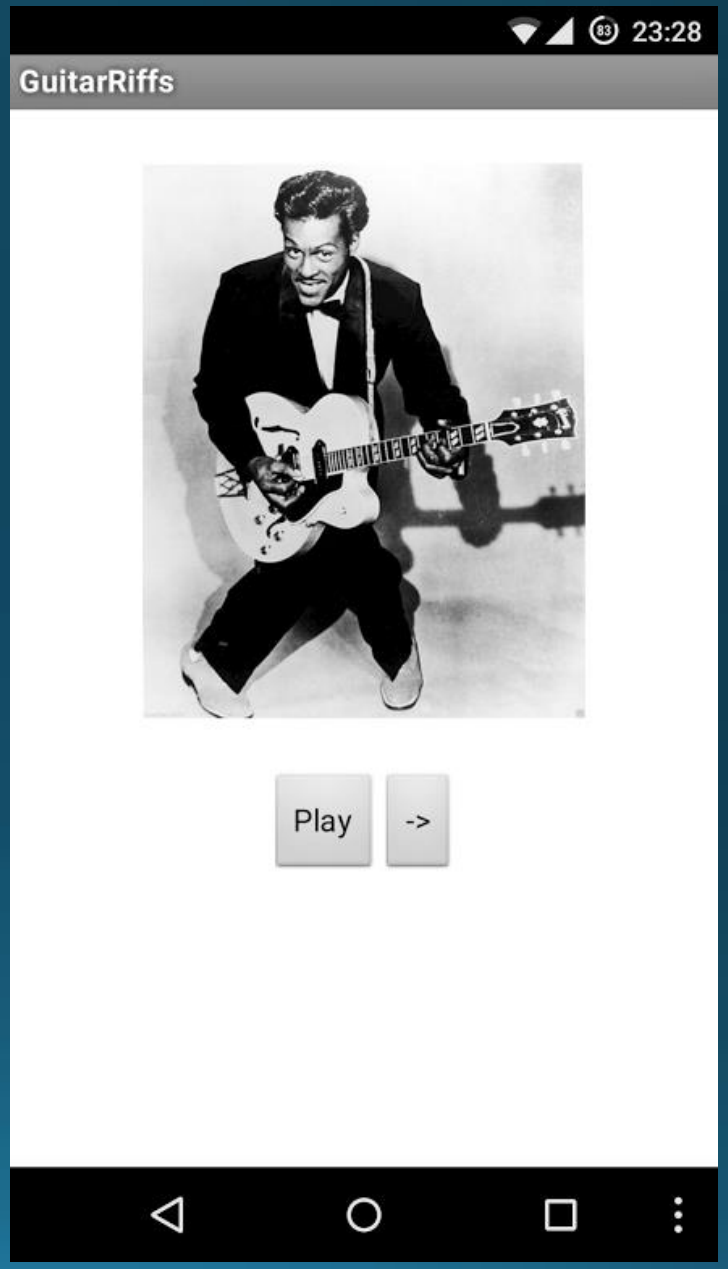
Test the app!

Are we ready to start today's app?



Today's App:

Guitar Riffs

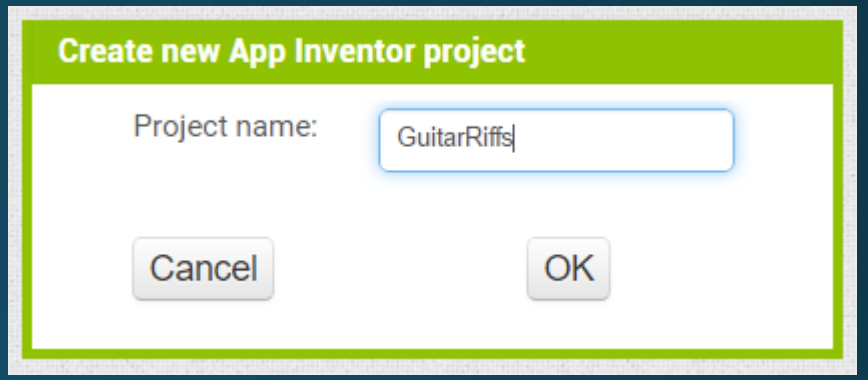


GuitarRiffs – Simple Version

- **Show a picture of the artist**
- **Click on a button**
- **Play a piece of music**

Make another new project

- Call the project **GuitarRiffs**

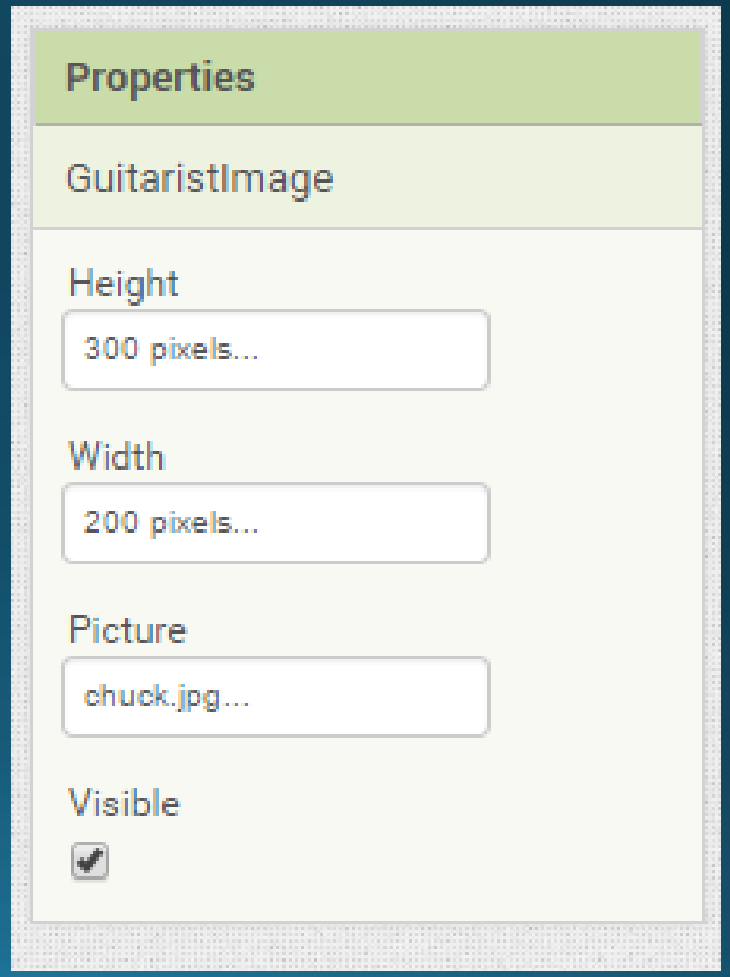


- Download and unzip the App files from the course website:

caltoc.scs.ryerson.ca

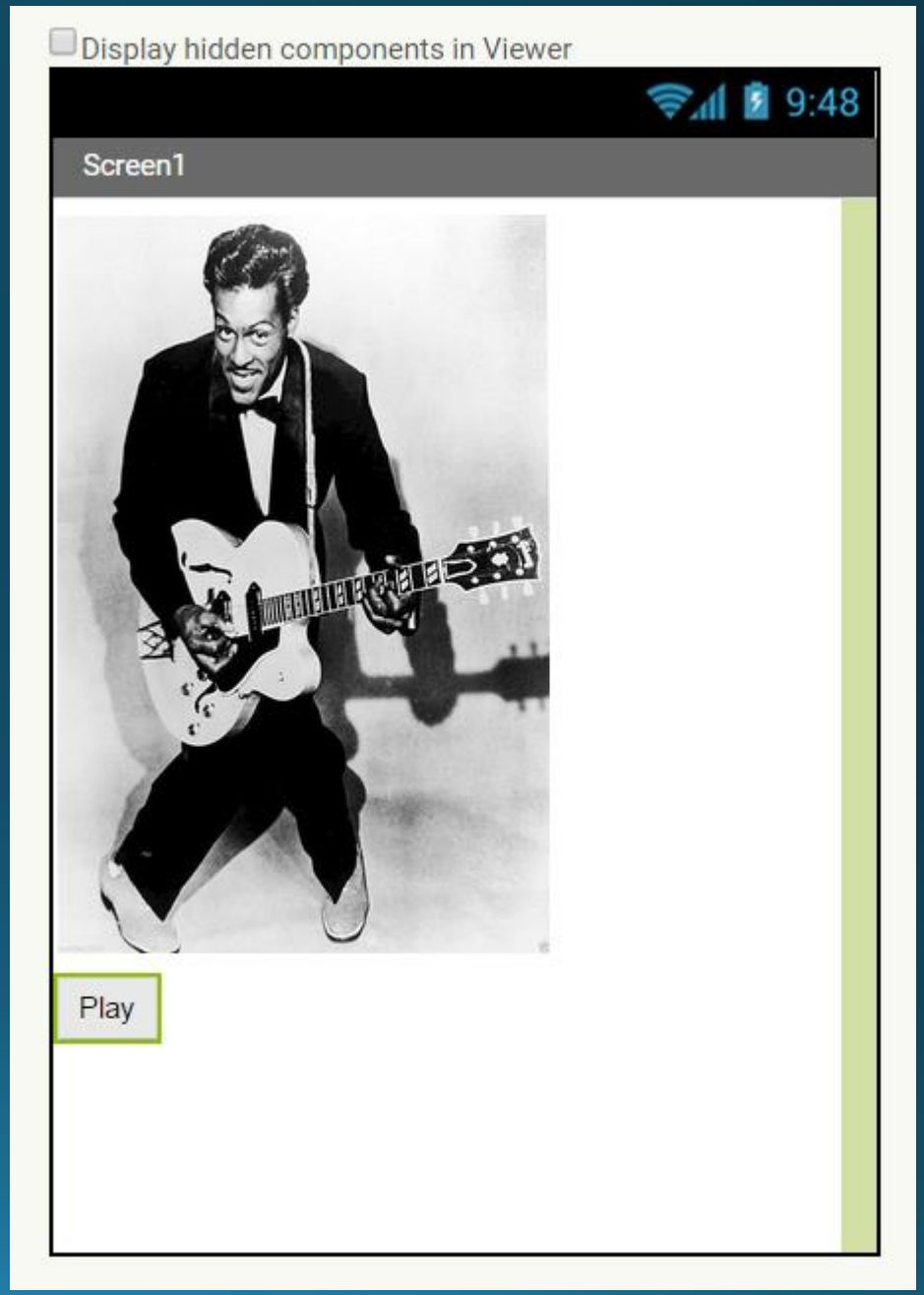
Building the User Interface:

- Drag in an Image component.
  - Rename the component from 'Image1' to 'GuitaristImage'
  - Set height property to 300 pixels
  - Set width property to 200 pixels
  - Click the picture property, and upload 'chuck.jpg' by clicking 'choose file' and finding it on your computer where you downloaded the files.



Add a button

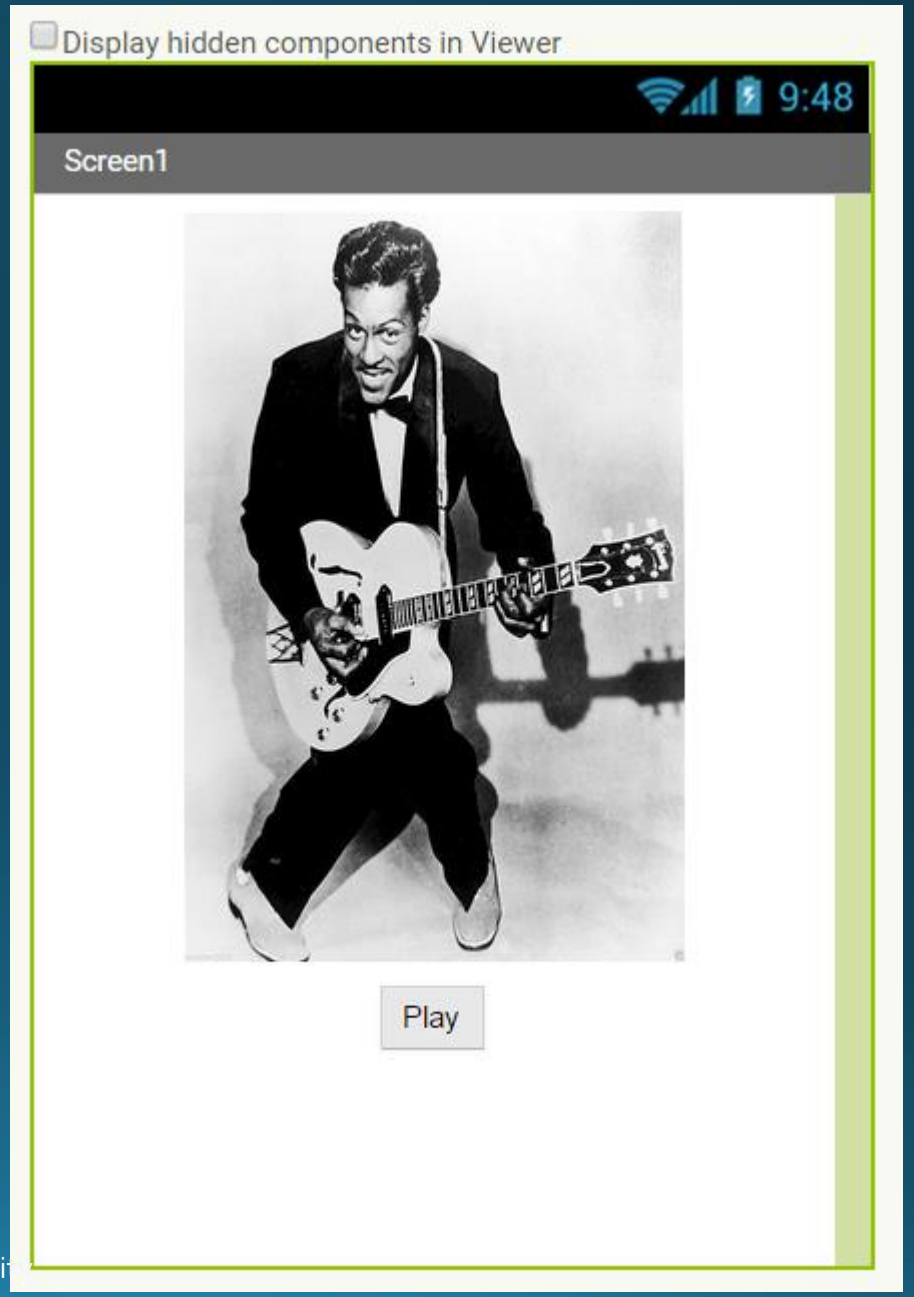
- Drag in a **Button** component underneath the **Image**
- Rename the button from **'Button1'** to **'PlayStopButton'**
- Set the text property to **'Play'**



Center for niceness

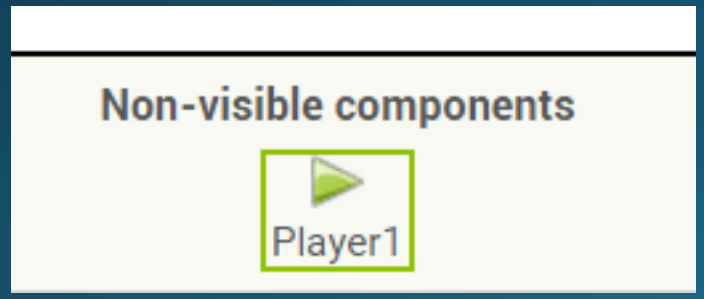
- On the Screen1 component, set the **AlignHorizontal** property to **'Center'**.

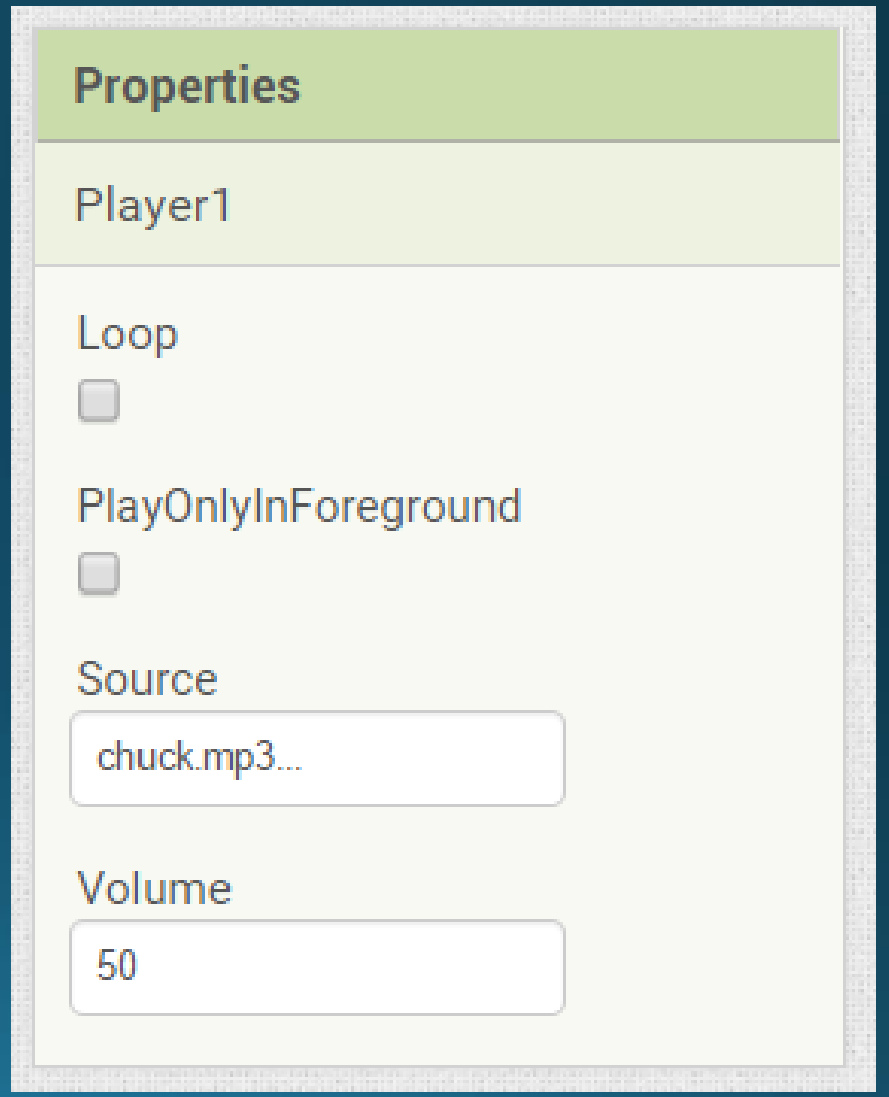
- Nice! Remember, the Screen1 component is actually a **VerticalArrangement** in disguise, and behaves the same way.



Music to your ears

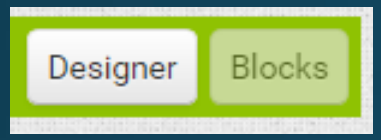
- Add a **Player** component, found under the **Media** tab in the palette.
- Set the **Source** property of Player1 to the chuck.mp3 file you already downloaded.



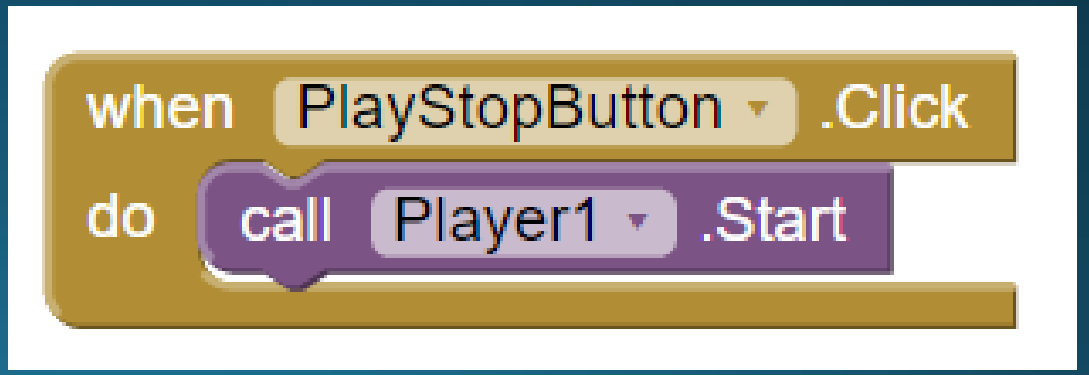


Making the Player work

- Switch to the Blocks view.



- We want to make the Player start the music when the button is clicked.



- So, use the button click event as well as the Play function call on the Player component.

Try to test the app.

Does the sound play?

For some android phones, You'll have to restart the app.

Making a decision

- So, great, now we have sound playing inside the app!
- But, what *if* we want to make the play button become a 'stop' button while the sound is playing
- When the user presses the button, we'll have to decide whether we:
  - start the music (*if* it is not playing)
  - or stop the music (*if* it is playing).

Condition Blocks!

- ***if*** **the music player is playing, stop the music.**

- ***else*** **(otherwise) if the music is not playing, we want to instead play the music**

- **Drag in the necessary blocks shown to complete the logic**

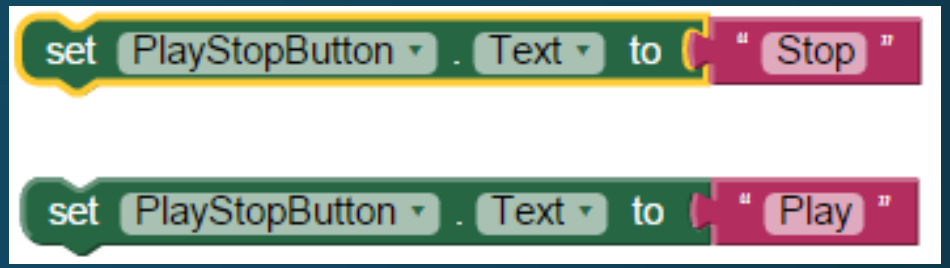
Discussion:

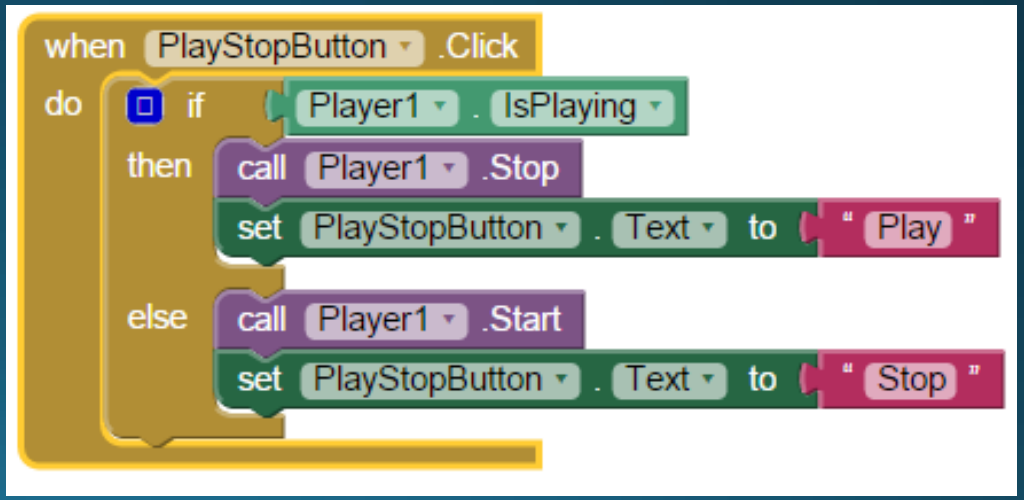
- **Is there anything that might confuse the user about the current functionality?**

- **If so, how can we fix this?**

Changing the Button's Text

- Drag out the setter block for the **PlayStopButton**'s **Text property**.
- Set the text to **'Play'** using a red text block.
- Make a copy of this entire block with **"Stop"** instead of **"Play"**
- Where do these blocks go?





Take a short break before we continue.

Good work!

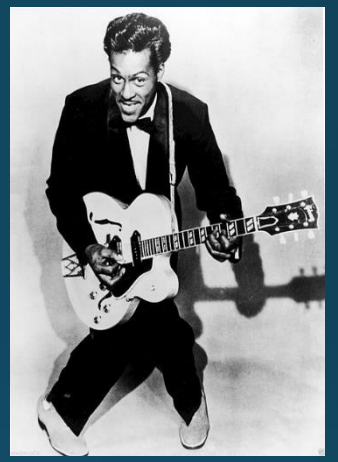
Cycling through Guitarists

- Next, we are going to add the ability to cycle between the different guitarists (Chuck Berry, Jimi Hendrix, Joe Perry)

- The order will be:

Chuck -> Jimi -> Joe -> Chuck -> ...

Chuck



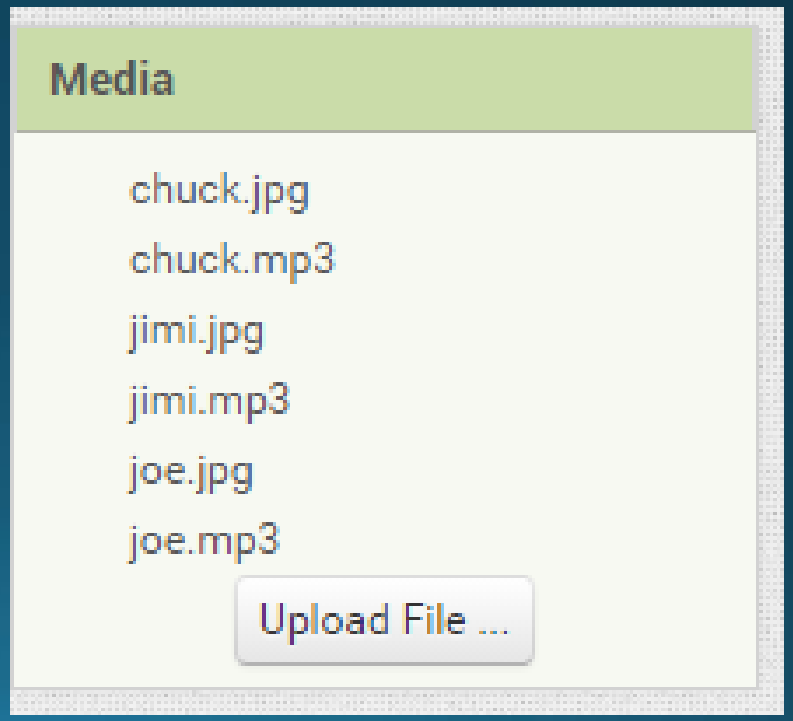
Jimi

Joe

Adding more Guitar Riffs!

- We want to cycle through the guitarists images, as well as changing the player to the corresponding mp3 file.

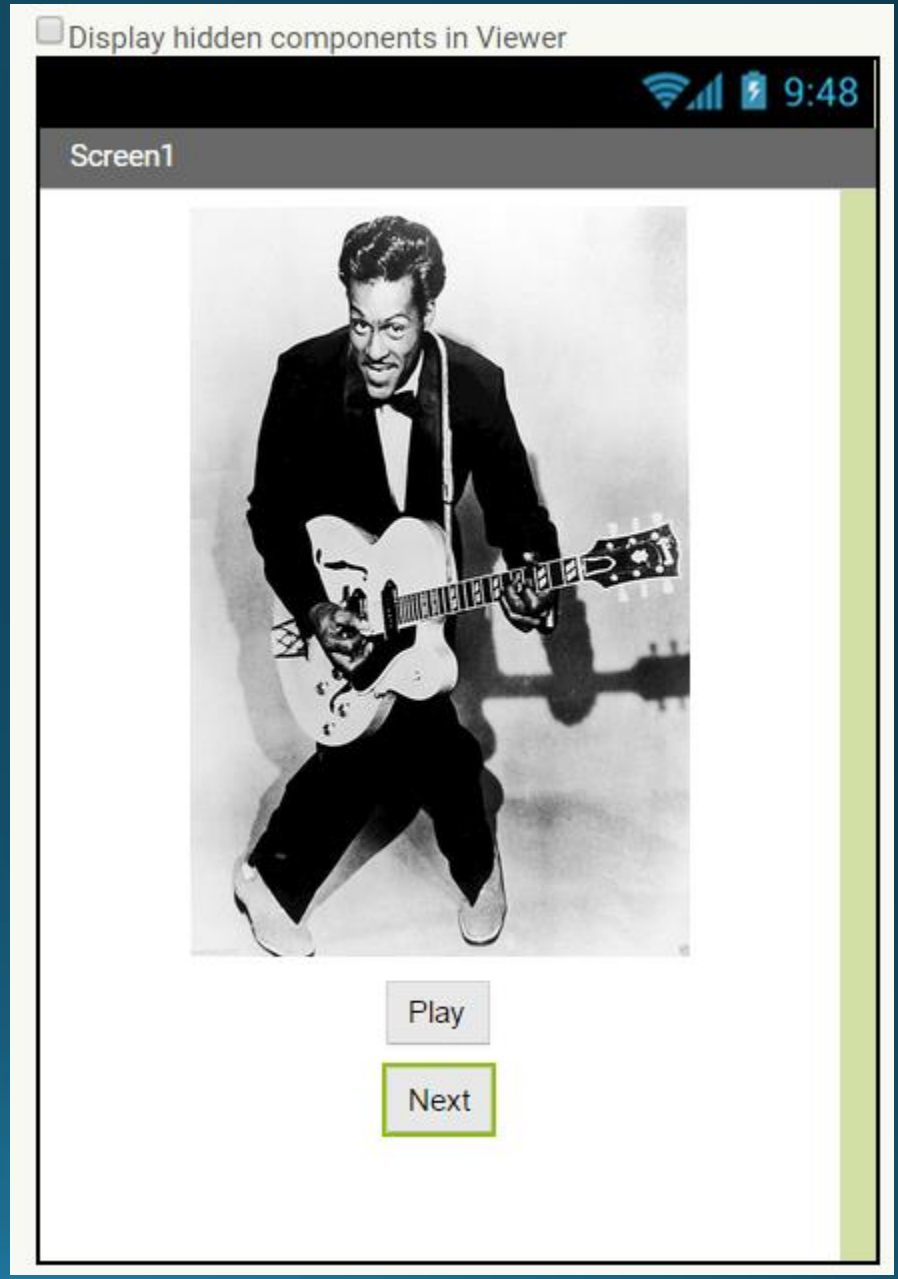
- Back in Designer view, under the **Media** panel, upload the rest of the files (jimi.jpg, jimi.mp3, joe.jpg, joe.mp3) by clicking **'Upload File'** and then **'Choose File'**.



Adding another Button

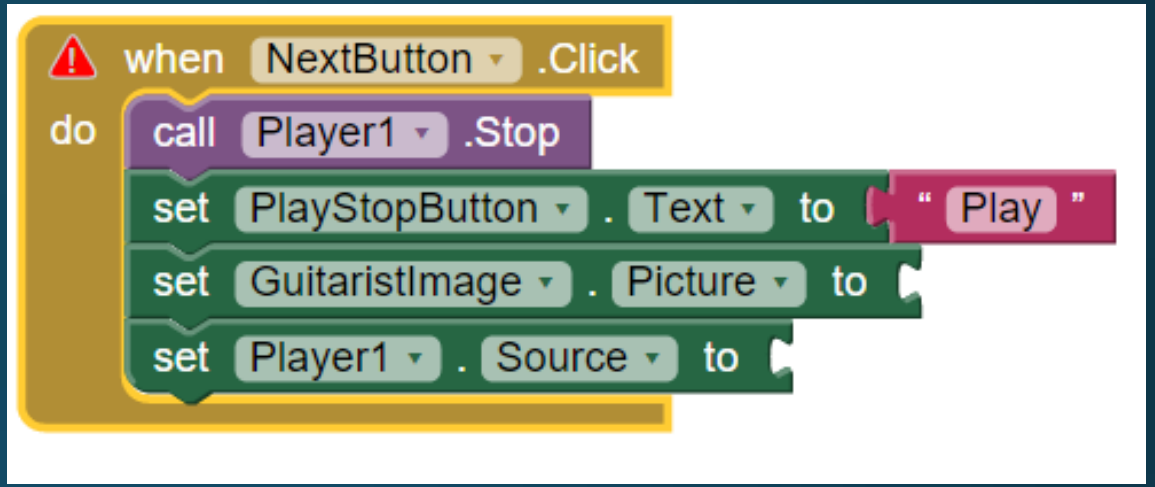
- Add a **new button** to the screen.
  - Call this new button **NextButton**
  - Set its text to **'Next'**

- Now, back in the blocks editor, we will add logic to cycle through the three guitarists as the user clicks the **NextButton**



Implementing the *NextButton* The Basics:

- For starters, when the **NextButton** is clicked, we'd like to:



1. *Stop* the Player
2. *Set* the PlayStopButton text to 'Play'
3. *Change* picture to the next guitarist
4. *Change* the Sound to the next riffs

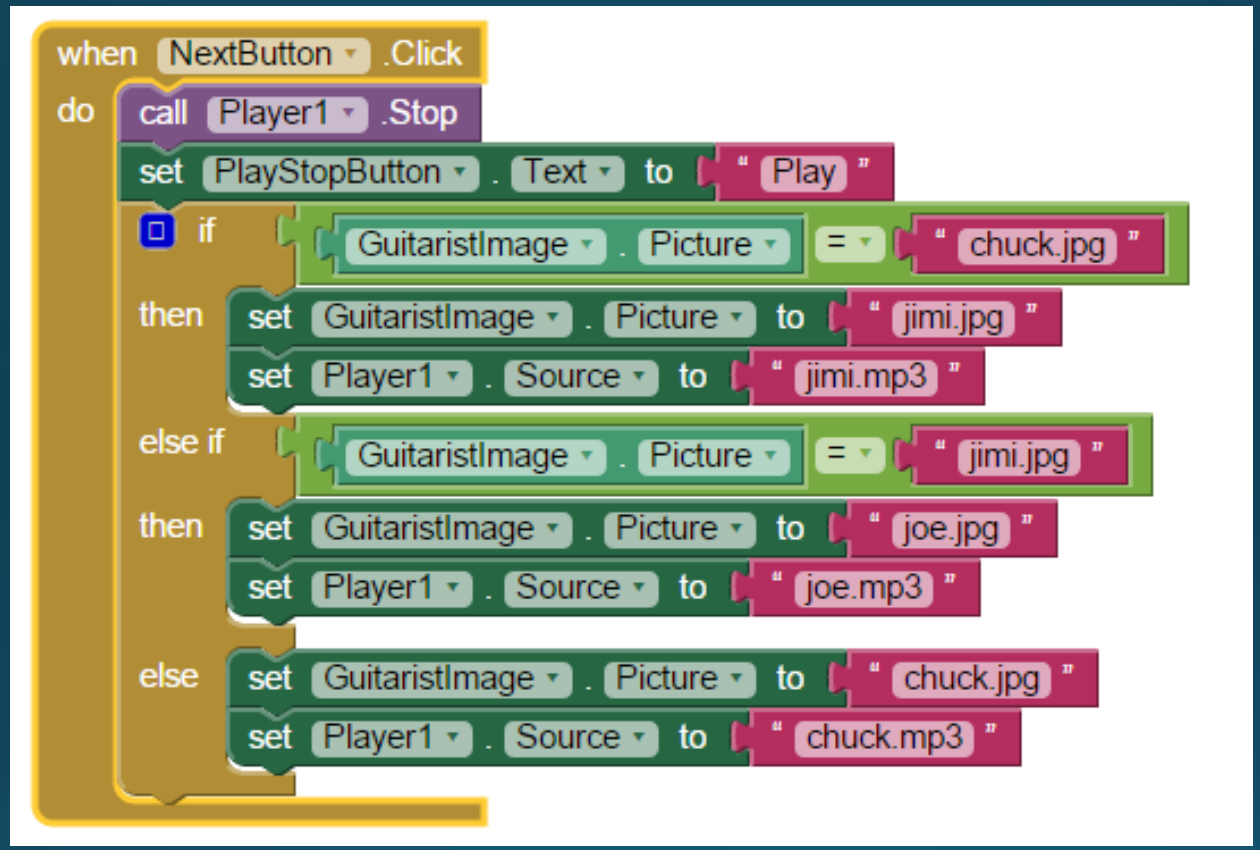
- Question: Who is the next guitarist?
  - That depends on who the current one is

In programming, there are many ways to solve a problem.

- **Let's look at a solution that might not be the best approach.**

The plan:

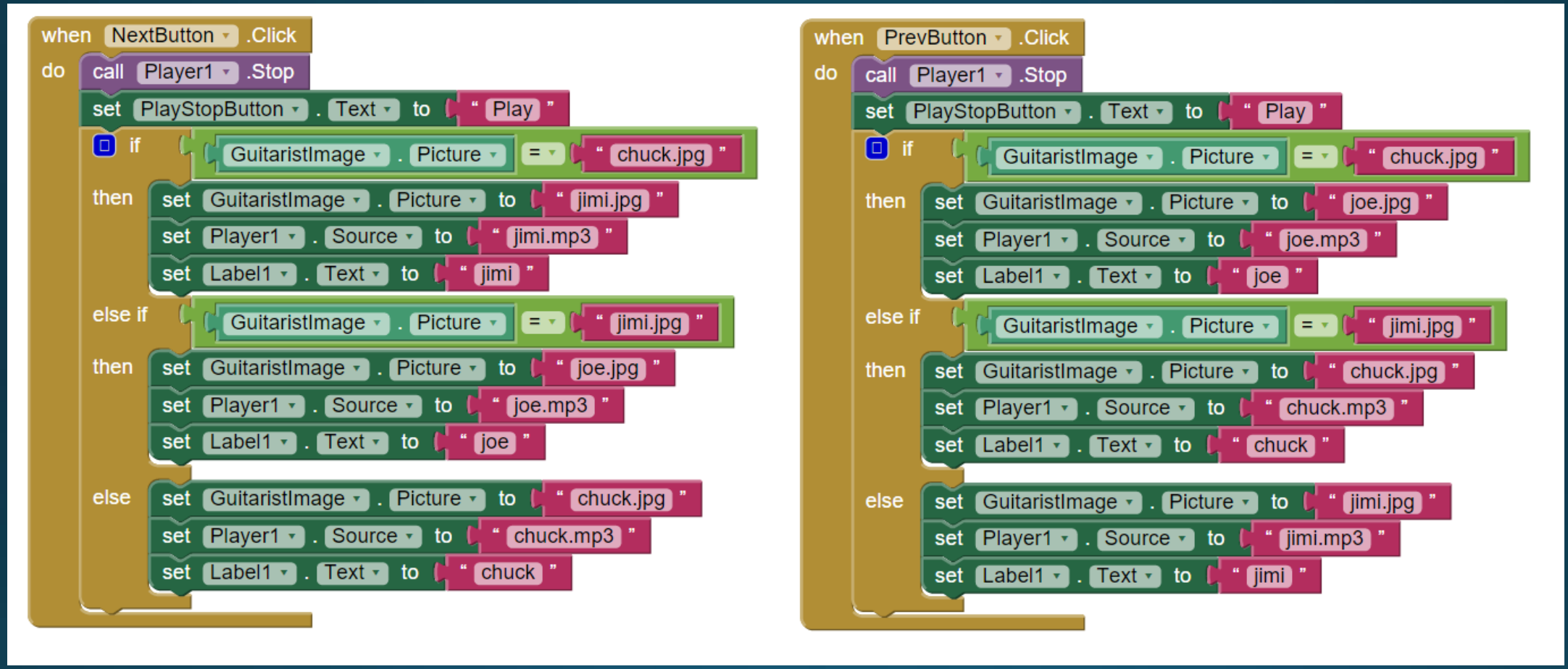
- If the Current guitarist is Chuck,
  - Set the image to Jimi
  - Set the sound to Jimi's Riff
- If the Current guitarist is Jimi,
  - Set the image to Joe
  - Set the sound to Joe's Riff
- If the Current guitarist is Joe
  - Let's go back to Chuck.
- In other words….



- Take a second to look at the code
- What happens when we add more guitarists?

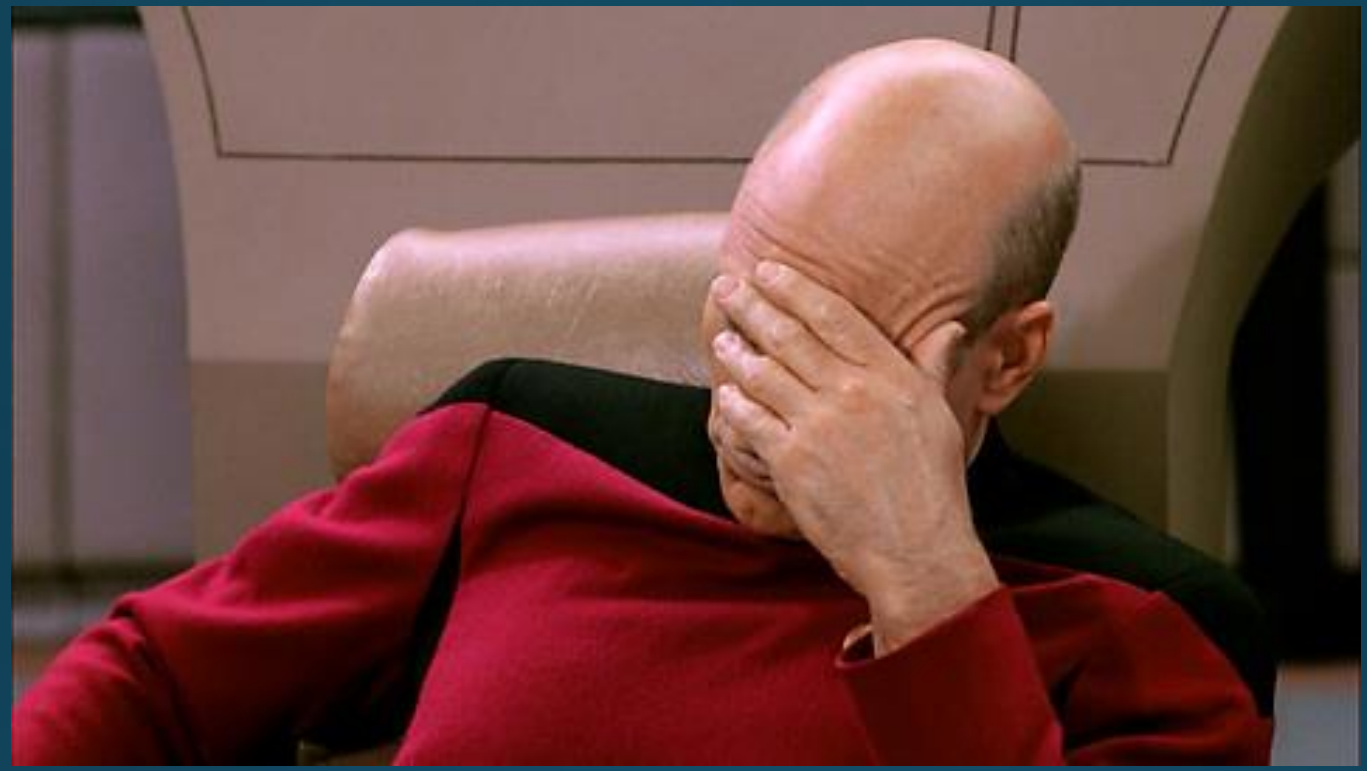
And the *really bad news?*

- **What would it take to implement a "Previous" button?**
  - **You don't need to follow along for this next slide.**



Again, this is for just 3 guitarists!

We had to duplicate all our code for a 'Previous' button.

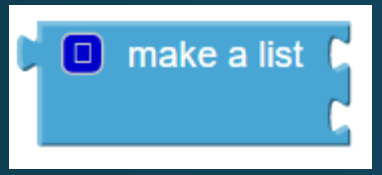


What is the alternative?

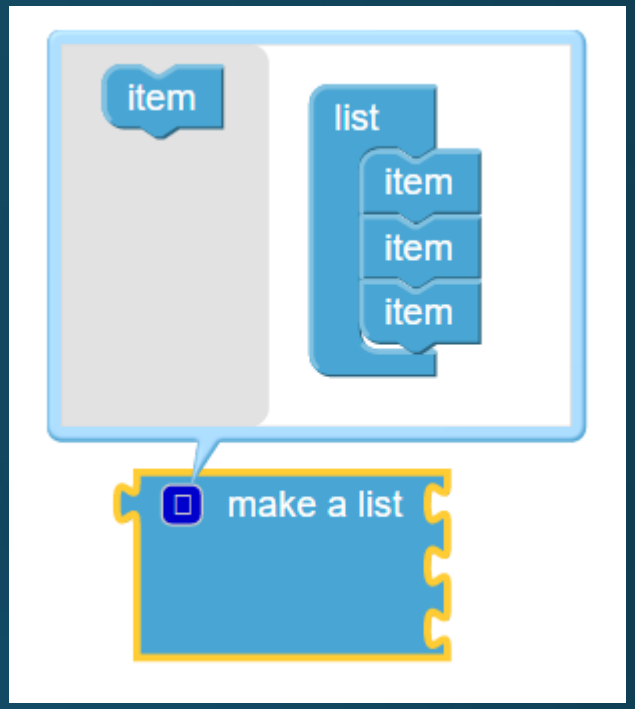
Lists!

Concept #7

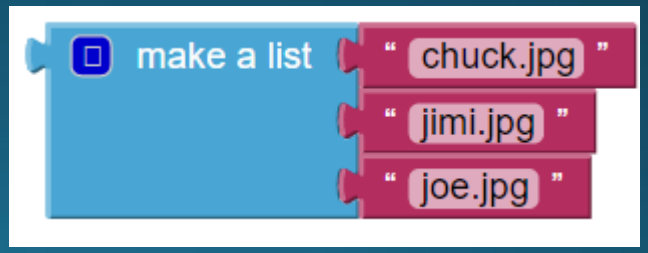
- In the blocks view, drag a



block from the Lists menu

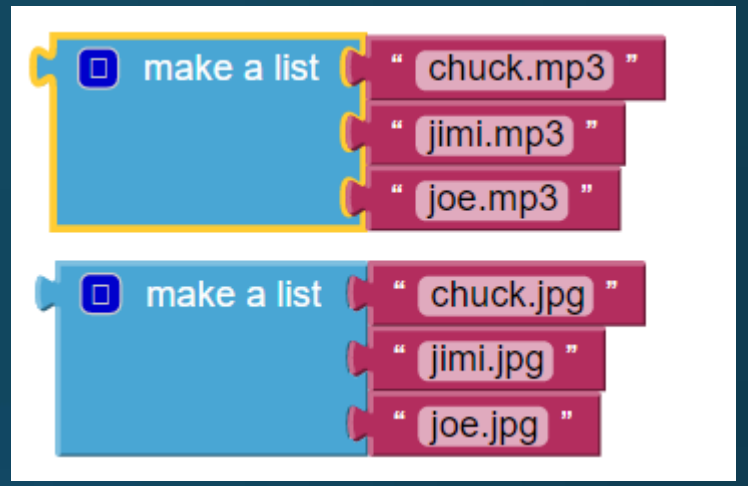


- Click the dark blue square to change the size of the list
- Drag the Image names into the list.



A second list

- Once you are done, Make another list for the Sounds.

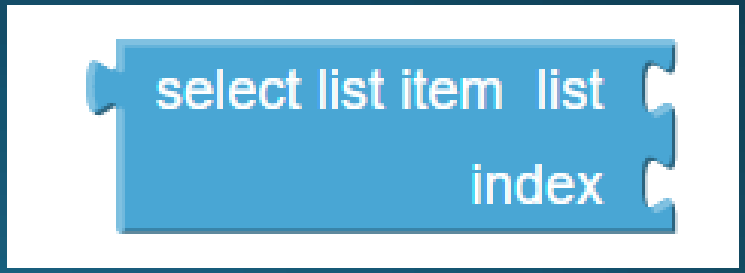


- In order to access the **lists** later, we'll have to store them in variables.
  - Make Two variables called
    - SoundList
    - ImageList

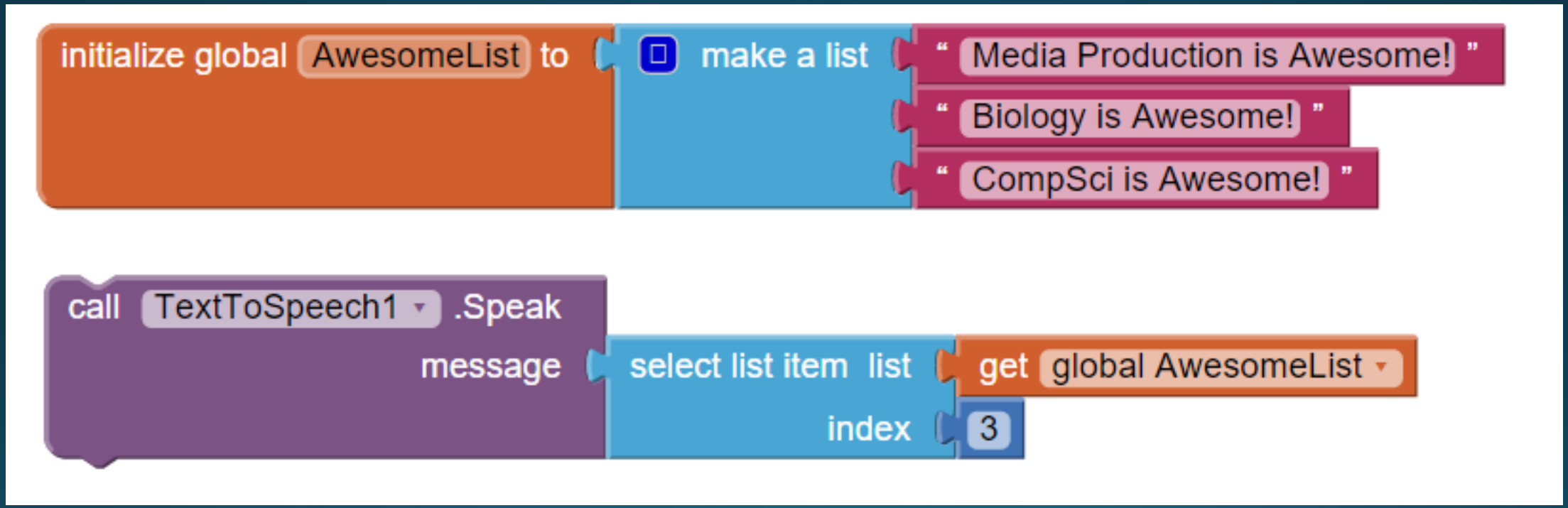
What are Lists?

- **Lists** are a way to group together a set of values and keep them in order.

- Each element in a list has a number that can be used to access it
  - This is called the index.

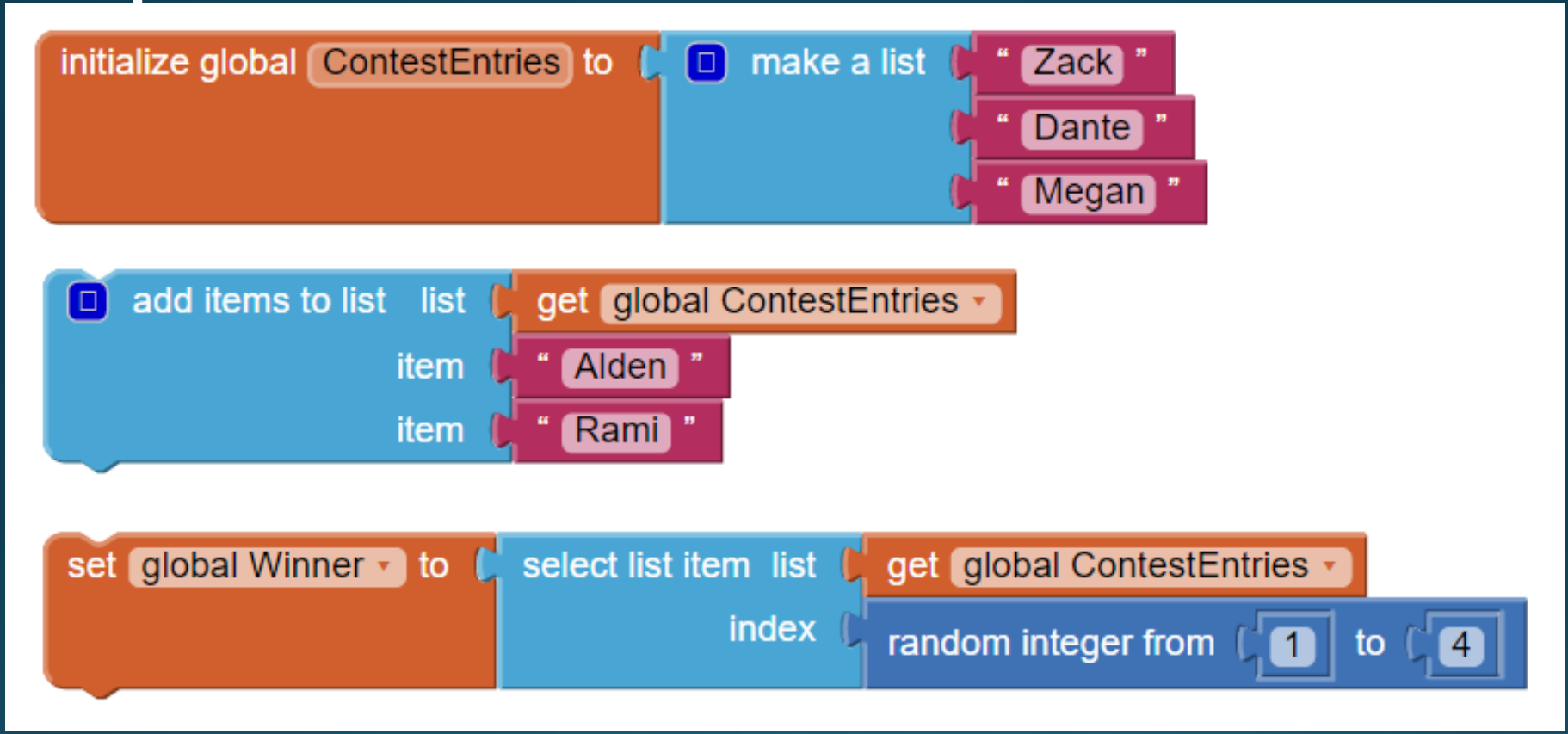


Example:



What does this code do?

Example:

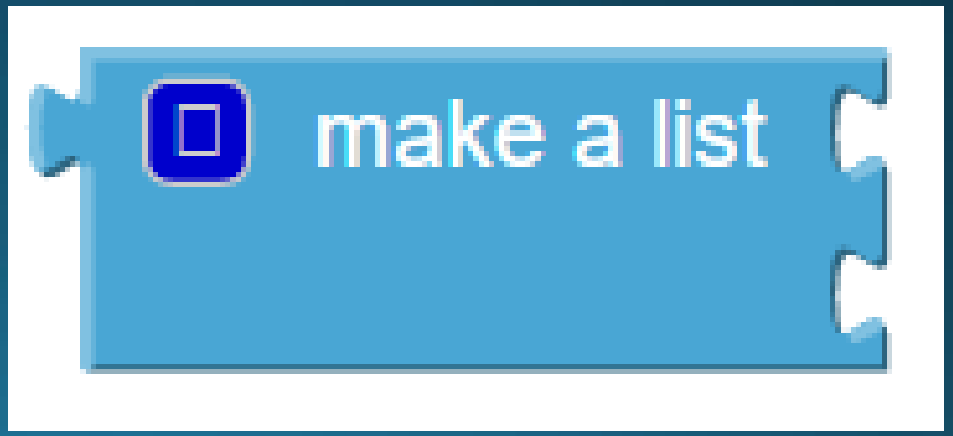


What does this code do?
Does everyone have a fair chance of winning?

Lists are Values

- Notice how the list block has a peg on the left side

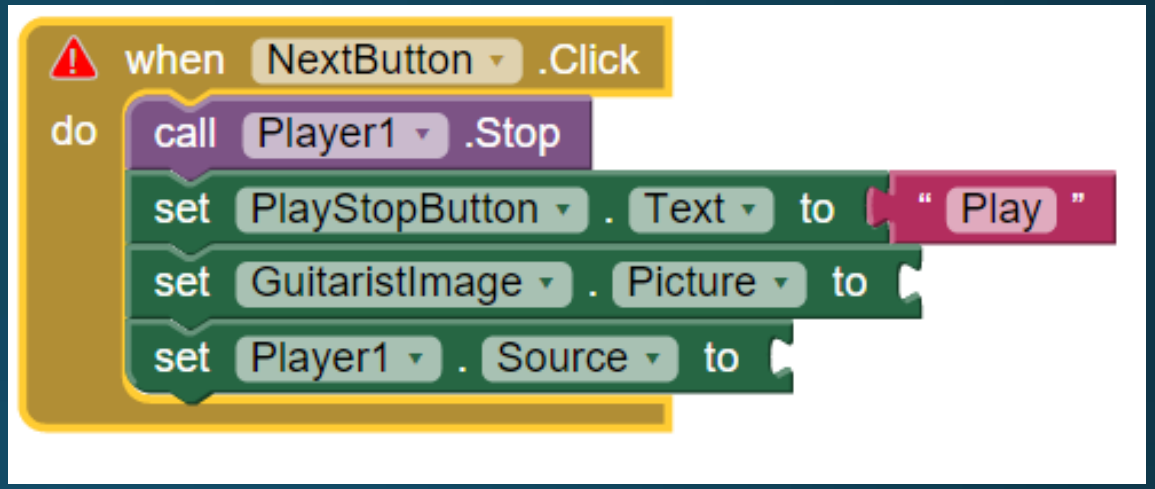
- This is similar to text blocks and number blocks in that they are *Values*
  - This is why we have to store lists in variables.
  - Does that mean we could make a list of lists…?



Back to our App…

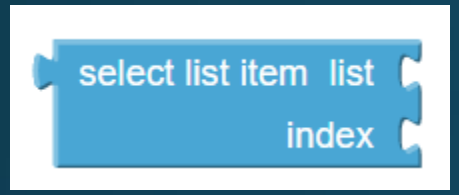
- Let's look at the Basic logic again →
- In order to set the picture and sound to the appropriate element of our lists, we'll have to keep track of an Index Variable!
- Make a Variable named *Index*

Rock on!



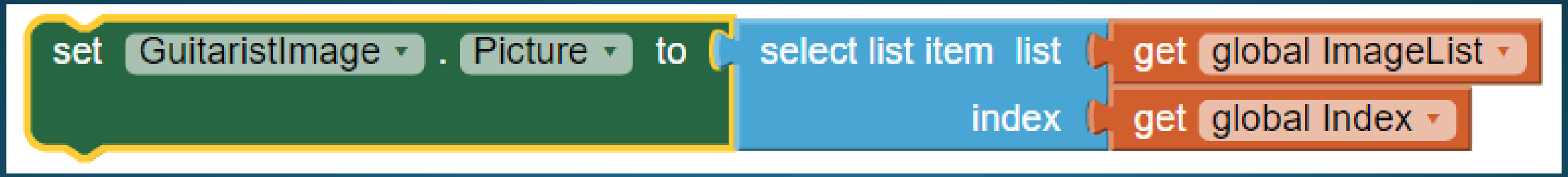
Getting the image filename from the ImageList

- We will use the



block to get the item using our index variable.

- Assemble the block below.
- Do the same for the SoundList and Player1.Source.

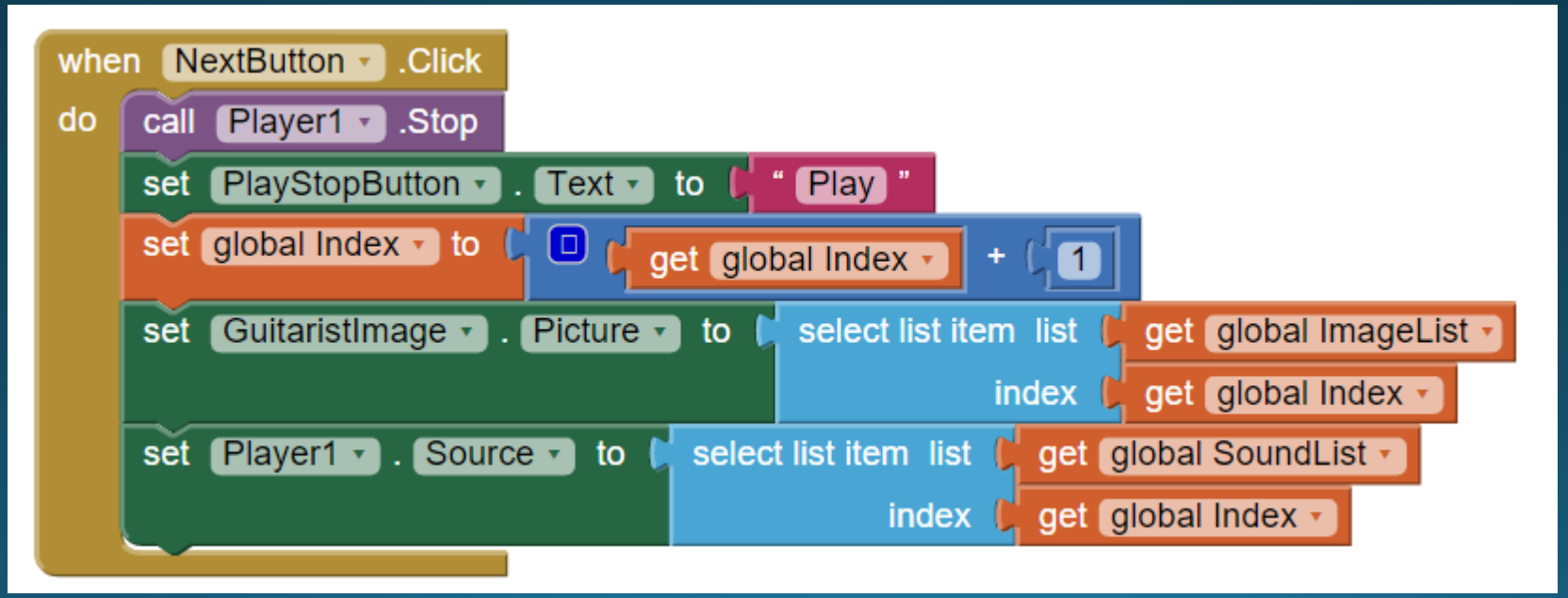


So far, our NextButton event looks like this:

- Try the app.
- Does it work?
- What do we need to add?

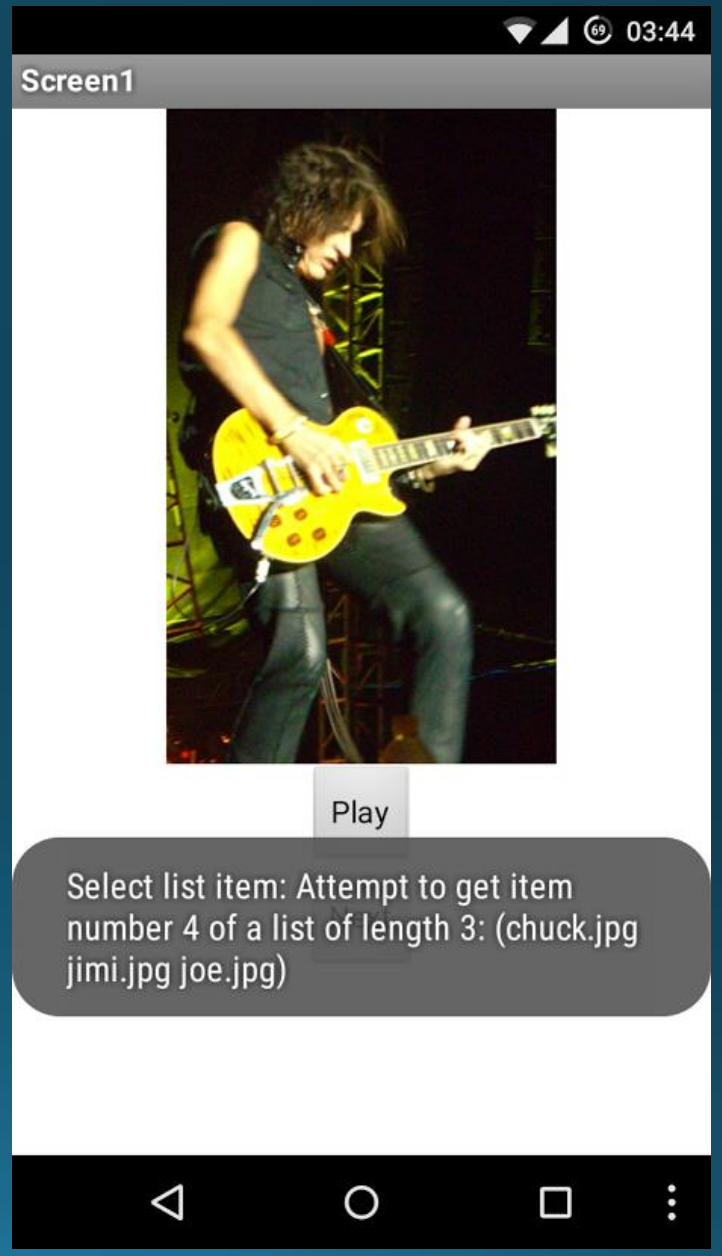
Moving to the next list item

- Every time the user clicks the NextButton, we'd like to add 1 to our *Index* variable (increment it).
- Add this logic to the event, in the proper place!



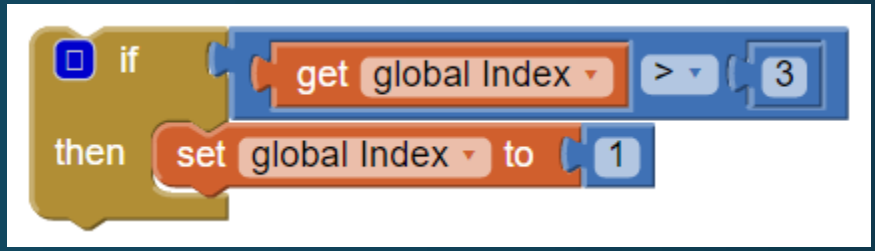
Now, test the app again.

- It seems that we can move to the next guitarist when pressing Next.
- However…
- Once we go past the last item in the list, we go out of bounds, and receive this error!

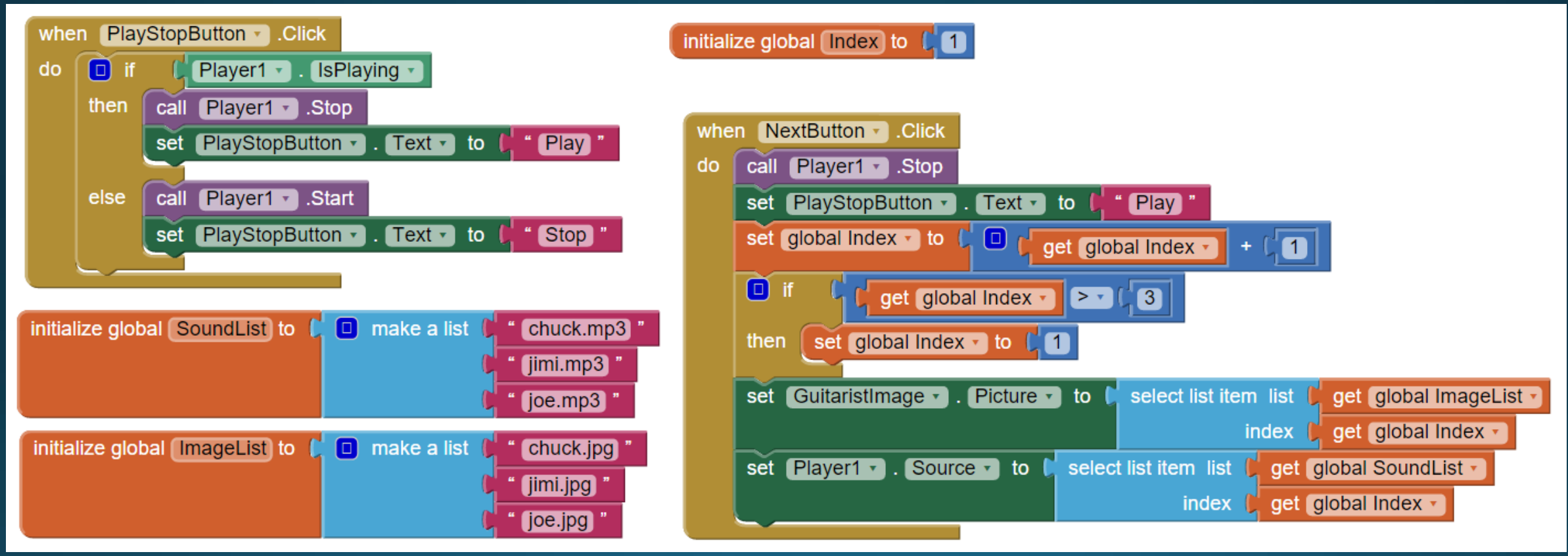


To finish the app

- We can fix this problem using a conditional.
- Every time we add 1 to the index, we want to check if the index is greater than the list size.
- If the index is out of bounds (greater than 3), we must set it back to 1.
- Find the right place for this block!



The finished blocks:



Enjoy the music!

Take a break. Challenges ahead.

End of GuitarRiffs app

End of lecture

- Don't forget to:
  - Return all phones to the front
  - Fill out the survey for today's lecture
  - If you have any questions regarding the course material, please email any of the TAs (or all three).
  - Lecture slides and materials are available online at the course website:

    caltoc.scs.ryerson.ca