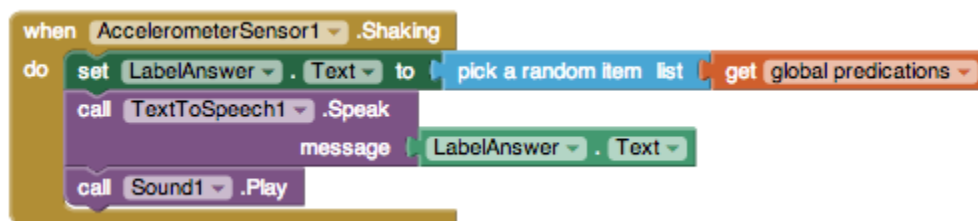


Magic 8-Ball Mini Projects

Here are some creative projects. Working in pairs, complete all of them. If you have questions, ask your instructor for help or, if working online, post a question in the forum.

1. Change the predictions that were used in the tutorial to your own predictions, perhaps one that are more humorous.
2. **If/Else Algorithm.** If you recall from the Tutorial video, one problem with the current version of the Shaking block is that it causes the Sound and the Voice to occur at the same time, making it hard to hear the prediction:



One way to fix this is to use an *if/else* algorithm to perform either the speaking action or the sound action but not both. Of course, the decision to do one or the other has to depend on some *boolean condition*. What could that be?

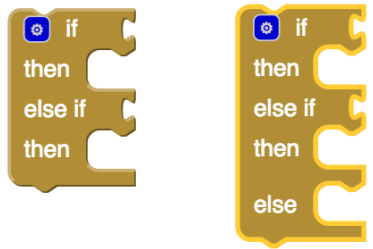
One idea is to add a feedback setting to your app, which controls the type of feedback the user receives -- either a sound or a spoken prediction (in addition, of course, to the written prediction in the Label). To allow the user to choose which option they prefer, you can add a settings menu to your app that allows the user to select between either Speaking the prediction or playing an alert sound.

A nice UI component for this type of menu is a [ListPicker](#). The ListPicker looks like a button but it has an *ElementsFromString* property that can be initialized in the Designer to a comma-separated list of choices. For example, if you initialize it to *Speak, Sound*, then when you click on the ListPicker, it lets you select one of those two choices. The ListPicker has an *AfterPicking* event handler:



What action should you take after the user has selected an option? You will need to remember what setting the user has selected. HINT: Use a **global variable** to remember the setting and use an **if/else** block to test the value of that variable in the *AccelerometerShaking* event handler.

3. **If/Elseif Algorithm.** Modify your solution to the previous exercise to allow a 3-part menu that includes the following options: *Speak*, *Sound*, *Silent*. In the Silent case, the app should silently display the prediction in the label. HINT: You'll need to *mutate* the if-block to include a second boolean condition. To do so click the blue mutator widget on the if-block. Your block should look like one of these. Notice that there are 2 when slots where you can plug an equals operator:



4. **Advanced:** Another shortcoming of the current app is that the Magic 8 Ball provides a prediction whenever it is shaken, even if the user hasn't asked a question. To fix this, require the user to "speak" their question to the Magic 8-Ball **before** Magic 8-Ball says its prediction. For this, you'll need a [SpeechRecognizer](#) component (Media drawer). Here's a [One Minute Lesson on the SpeechRecognizer](#). HINT: To have the app "wait" for the user to speak, you'll have to call the *SpeechRecognizer.GetText* procedure when the device is shaken. And you'll have to use *SpeechRecognizer.AfterGettingText* to provide the feedback to the user.
5. **Be Creative.** Come up with your own ideas to enhance your app or to create a variation of this app.

Come up with your own ideas to enhance your app or to create a variation of this app. Now reflect on what you learned in this lesson and then retry the interactive exercises with the goal to answer each question correctly.