|  |  |
| --- | --- |
| Magic 8-Ball is a fortune-telling app. In this app, a virtual magic 8-ball can be asked questions and provide predictions. A user can ask a question, shake the phone, and hear the magic 8-ball’s prediction to their question. This app uses an accelerometer to handle the shaking event, a list of possible predictions, and text-to-speech to have the phone speak the magic 8-ball’s prediction, which is randomly selected from the list of predictions. CSP Learning Objectives:  * *Creativity: 1.1.1 - The student can use computing tools and techniques to create an artifact.* * *Creativity: 1.3.1 - The student can use programming as a creative tool.* * *Programming: 5.3.1 - The student can develop a correct program.* * *Programming: 5.4.1 - Employ appropriate mathematical and logical concepts in programming. [P1]* |  |

## 

# Getting Ready

For this lesson you can begin with the app you created in the Magic 8 Ball Tutorial or you can open [App Inventor with the Magic 8 Ball Projects](http://ai2.appinventor.mit.edu/?repo=appinventor.cs.trincoll.edu/csp/gcb/Magic8BallProjects/Magic8BallProjects.asc) template.

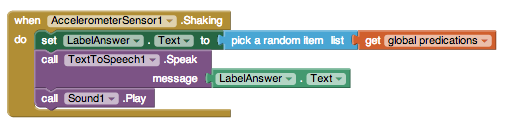
In either case, use the *Save As*  option to rename your project to *Magic8BallV2* (for version 2) or something like that.

# 

# 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. **Challenging.** 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 put a delay between the *Play* and the *Speak* actions. The [Clock](http://ai2.appinventor.mit.edu/reference/components/sensors.html#Clock) component (Sensor drawer) can be used for this purpose. When the Clock is enabled, it generates a ***Timer event*** every so often depending on the value of its ***TimerInterval*** property. The default interval is 1000 milliseconds. Timer events are handled by the *Clock.Timer* event handler:



Using the timer correctly can be tricky. Here are some suggestions:

* Initially the Timer should be *disabled*(Clock.enabled property is false).This will prevent it from “ticking” -- i.e., it will prevent its Clock.Timer block from firing.
* When you want to delay an action, say *Action A*, enable the Timer. This will start the Timer “ticking”. The Clock.Timer block will fire on every *TimerInterval --* i.e., by default, every second.
* When the *Clock.Timer* block fires, perform *Action A* and disable the Timer again.

You might also want to take a look at this [One Minute Lesson on the Clock Timer](http://www.youtube.com/watch?v=sdDcZfeCBXc). Try to use these ideas to put a delay between the playing of the Sound and speaking of the prediction.

1. **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*](http://ai2.appinventor.mit.edu/reference/components/media.html#SpeechRecognizer)component (Media drawer). Here’s a [One Minute Lesson on the SpeechRecognizer](http://www.youtube.com/watch?v=hSLNx6-mYKY).
2. **Be Creative.** Come up with your own ideas to enhance your app or to create a variation of this app.

## *Nice work! Complete the Self-Check Exercises and Portfolio Reflection Questions as directed by your instructor.*

## 