

# Project: Magic eight ball

Difficulty level: Easy



## Description

In this project we'll be creating a *Magic 8 ball* using the Sense HAT. If you've never seen one before, you might remember it from Toy Story: <https://www.youtube.com/watch?v=mFORacFCIBg> (<https://www.youtube.com/watch?v=mFORacFCIBg>).

It's a very simple toy: You ask it a question, you shake the ball (or Sense HAT), and it'll give you an answer.

You can try an example solution of the project here:

**trinket**

Run

To get the answer, press the middle joystick button (ENTER on your keyboard).

In the version you will make, you will shake the Sense HAT to get the answer. Unfortunately you can't shake the virtual Sense HAT on Trinket, so in the version above we used the joystick instead.

## Project Manual

This project guide will tell you step-by-step the main things you have to do in order to create a *magic eight ball*. For some of the steps, you'll have to use your own creativity to proceed, good luck!

---

## Introducing the project

The first thing you should do is open the *skeleton code* for the project. In programming, skeleton code means code that only has the basic elements of a program. It is up to you to fill in the rest!

You can find the skeleton code on Trinket, here:

<https://goo.gl/bzqdNi> (<https://goo.gl/bzqdNi>)

**If you can, you should also create an account and log in to Trinket. This will allow you to save the Trinket projects. Otherwise you have to copy the code on to your computer to save it.**

On Trinket, you'll be able to test your code on a *virtual* Sense HAT, before you try your code on the real thing.

As you might see, the skeleton code is split up into sections, divided by the headlines. For example:

```
#### 2. Code section
```

This guide will go through the various sections (not necessarily in order), and help you write your code. **Very important note:** *You should add the code in the specified section in your skeleton code as you follow this guide.*

The next part of this guide will explain the stuff that's already in the skeleton code when you first open it.

### Explanation of the skeleton code

Before we get on to the coding, it's worth looking over the *skeleton code* and make sure you are familiar with it.

The first few lines in the script are:

```
#### 1.1 Import libraries

import sys
sys.path.insert(1, '/home/pi/Go4Code/g4cSense/skeleton')

from sense_hat import SenseHat
from senselib import *
```

Without going into detail, these lines are called *import statements*. They are used to *import* code from other Python files into your own file. This is useful because you can use other people's code to simplify your own.

The next part of the code (Sec. 1.2) creates some important *Objects* (don't worry if you're not sure what that means) that we'll use in the later on.

Sec. 1.3 is where you will write all the messages that the eight-ball can give.

Sec. 2 is where you will write the code that checks if the Sense HAT is shaken.

---

## Writing the code

### (Sec. 1.3) Write answers

When you shake the Sense HAT, we want it to give an answer. In this section you will write the answers that the magic eight ball can give.

If you look at the section, you'll see the following code:

```
answers = []
```

You'll have to add the answers as *strings* in the list, for example:

```
answers = ["yes", "no"]
```

A string is a piece of text enclosed within either double " or single ' quotation marks. The strings in the list have to be separated by commas (,).

This is an empty list, that you should fill with possible answers. Here are some typical magic eight ball answers that you can use:

- Outlook good
- Yes
- No
- Ask again later
- Better not tell you now
- Concentrate and ask again
- Don't count on it

But instead of using these, you should try to come up with your own!

### (Sec. 2) Main program code

Before we start coding, we'll explain a bit what's going on in Sec. 2.

In this part of the code, we'll sense if the user has shaken the Sense HAT. If the user shakes it, the program should display one of the answers we made up in the previous section by random.

*Code explanation:* If we look at the skeleton code, we see that the section is within a *while*-loop.

Remember that code inside a while loop runs again and again, until we tell it to stop. The reason why we want our program to be in a loop is because we want to continually check if the user has shaken the device.

Because the code is in the *while*-loop, remember that you have to add a *Tab* at every line, to make it indented. Like this:

```
while True:
```

```
    # write your code like this,  
    # with a tab at the start of the line.
```

(Sec. 2.1) Check if the user shaken the Sense HAT)

**This part is a bit complicated, so pay attention! Don't be afraid to ask a supervisor if you don't understand something.**

To see if the user has shaken the Sense HAT, we'll have to use the *accelerometer* inside the device. The

accelerometer basically tells us how fast the device is accelerating.

If you don't know what the word *acceleration* means, the next bit will give you a short lesson in physics.

#### Acceleration:

If *speed* is how fast

something is moving, *acceleration* is how fast the speed is changing. For example,

let's say you drop a ball from a cliff. At first, when you first drop the ball, its speed is 0 (it's not moving). But due to gravity, it'll *accelerate* and gain speed. By the time the ball has reached the ground, it's gained a lot of speed.

To get the information from the accelerator, use the following code:

```
ac = sense.get_accelerometer_raw()  
x = ac["x"]  
y = ac["y"]  
z = ac["z"]
```

Now, in the *x*, *y* and *z* variables, we have information recorded by the accelerometer. This is all a bit complicated, but the basic gist of it is this, calculate this:

```
shake = x*x + y*y + z*z
```

We have stored *x times x plus y times y plus z times z* into the variable *shake*. Now, the larger the variable *shake* is, the more has the Sense HAT been shaken.

We're going to say that if *shake* is *larger* than 5, the Sense HAT has been shaken. To do this, you're going to have to use an *if*-statement.

If you have detected that the Sense HAT has been shaken (in other words, if *shake* is larger than 5) you want to display a one of the messages we wrote earlier by random. To do this, we use the *random.choice* function, like this:

```
random_message = random.choice(answers)
```

To then display the message stored in *random\_message*, we use the functions *sense.show\_message*. Check the *Function Reference* to see how to use it.

This might seem complicated, but it's fairly straightforward. If something confuses you, ask a supervisor for help.

To break it down, what you need to do is this:

- Get the accelerometer information.
- Calculate *shake*, as shown above.
- If *shake* is greater than 5, show a random message.

## **Finished!**

If it's all done, correctly, the game should now work! Don't worry if it doesn't, things often go wrong in programming. Errors in code are usually called *bugs*. If you have a bug in your code, you'll have to *debug* it!

If it works, congratulations! You can either move on to another project or try to come up with new things to add to the current project. Use your creativity! You can discuss any ideas you have with a supervisor.

---

**Author:** Lukas Kikuchi

**Date:** August 09, 2017

**Copyright (c)** 2017 Go4Code All Rights Reserved.