Sound Effects

In this lesson we are going to introduce sound. To do this lesson we need to first import a sound library for processing.

Tasks

- **Import the minim library.** Do this by going “Sketch → Import Library… → Add Library…”, search for “sound”, select **Minim** and install.

- To use this library we are going to write the code below to begin our program. Minim and AudioPlayer in the code below are what are called objects. Objects are like variables but you can call specific functions on them. You don’t need to worry about them too much except that if you want to add a second audio file you will need to create a new AudioPlayer object.

```
import ddf.minim.*;
Minim minim;
AudioPlayer myAudioPlayer;

void setup() {
  size(200, 200);
  //Set up the sound library variable.
  minim = new Minim(this);
  // use sound library to load the initial file into
  // the audio player. File needs to be in the project folder.
  myAudioPlayer = minim.loadFile("Ping.mp3");
}

void draw() {
  //tell the player to play the music
  myAudioPlayer.play();
}
```

- For this code to work you need to make sure you have an mp3 file called “Ping” saved in your sketch folder. You can find this file if you go to “www.theacademyofcode.com/resources”. If you have that done and your headphones are plugged in you should hear a “Ping” sound when you run the sketch. (It won’t repeat).

- We are going to use this sound as an effect when a ball hits the sides of the window. To do this we are first going to make a ball bounce around the screen. Look back at your code from the Bouncing Balls lesson that you did last term.

- If you still have that code you only need to copy it into your new sketch. If you don’t have that code here’s what you need to do.
  - Create variables for x and y position, x and y speed and gravity.
  - Create a display function that will show an ellipse on the screen at the variables position.
  - Create a move function that will use the x and y speed to change the x and y position of the ball.
  - Create a gravity function that will change the y speed of the ball.

  - Create a bounce function that will make the ball bounce off the edges of the screen. (Remember this means multiplying the speed by -1).

- When you have the ball successfully bouncing around the screen we are going to add in the sound effect when it hits an edge. Use the appropriate line from the code above to play the "ping" sound when the ball hits an edge.

- You might notice that the sound effect will only happen the first time the ball hits an edge. To fix this we are going to add a 0 in between the play brackets, like below. This will start the audio from the very start every time it is run.

```
myAudioPlayer.play(0);
```

- Run your code for a while to see what happens. We can hear that the sound effect happens a little bit after the ball hits the edge of the screen. To fix this we can start the sound file later by changing the 0 to another number, try changing it to 575.

- When you get that sound effect working like you want try getting another mp3 file from the resources folder and having it run when you hit a separate wall.

- To do this you will need to create a new AudioPlayer object with a new name and use the loadFile function to load in the new sound.