BEGINNING METAL

HANDS-ON CHALLENGES

Beginning Metal

Caroline Begbie

Notice of Rights

Notice of Liability

This challenge and all corresponding materials (such as source code) are provided on an "as is" basis, without warranty of any kind, express of implied, including but not limited to the warranties of merchantability, fitness for a particular purpose, and noninfringement. In no event shall the authors or copyright holders be liable for any claim, damages or other liability, whether in action of contract, tort or otherwise, arising from, out of or in connection with the software or the use of other dealing in the software.

Trademarks

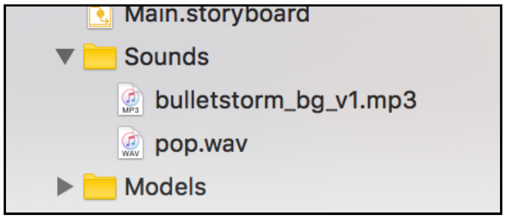
Challenge #13: Music and Sound

By Caroline Begbie

No game is complete without music and sound effects.

In this challenge, you're going to create a sound class and add music and sound effects to your app.

You'll find all the sound files in the Resources folder for this video. Drag them into your game from Finder to the Project Navigator.



Here's the background music `bulletstorm_bg_v1.mp3` and the bounce noise `pop.wav`.

Create a new file called **SoundController.swift**, and replace the contents with this:

```
import AVFoundation

class SoundController {
}
```

`SoundController` will be a singleton, so add a property to return the singleton instance:

```
static let shared = SoundController()
```

Create properties for the background music and property for the bounce sound effect:

```
var backgroundMusicPlayer: AVAudioPlayer?
var popEffect: AVAudioPlayer?
```

Create a method to preload a sound effect from a file:

```
func preloadSoundEffect(_ filename: String) -> AVAudioPlayer? {
  return nil
}
```

At the top of this method, get the url from the file name

```
if let url = Bundle.main.url(forResource: filename,
                             withExtension: nil) {
}
```

Loading the audio file must be done in a try catch:

```
do {
  let player = try AVAudioPlayer(contentsOf: url)
  player.prepareToPlay()
  return player
} catch {
  print("file \(filename) not found")
}
```

Create an initalizer, and pre-load the bounce sound effect:

```
private init() {
  popEffect = preloadSoundEffect("pop.wav")
}
```

Create a method to play the background music:

```
func playBackgroundMusic(_ filename: String) {
  backgroundMusicPlayer = preloadSoundEffect(filename)
  backgroundMusicPlayer?.numberOfLoops = -1
  backgroundMusicPlayer?.play()
}
```

Here you preload the file and set the number of loops to -1. This means that the music will loop over and over. (And over and over...) :].

And create the method to play the bounce sound effect:

```
func playPopEffect() {
  popEffect?.play()
}
```

That's all that's needed in your sound controller - you have a method to play background music and a method to go pop when the ball bounces.

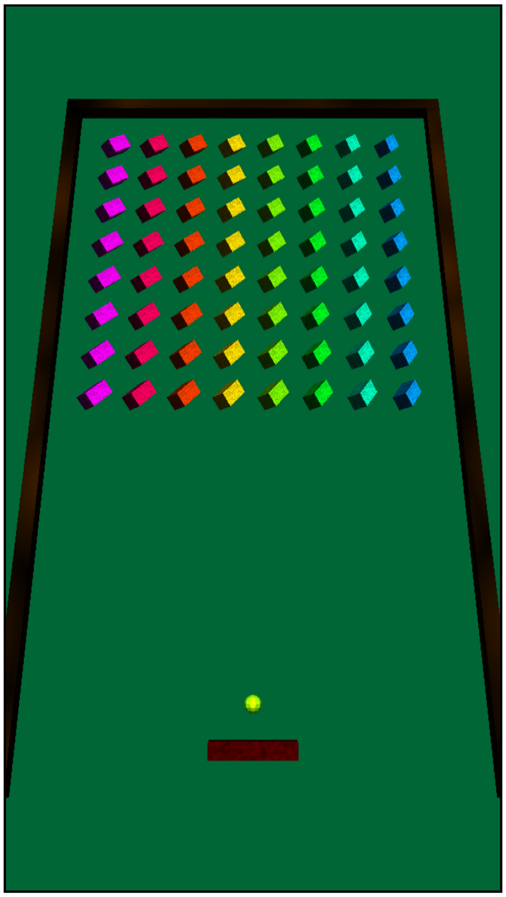
Now to add these to your game scene.

The background music should play all the time, no matter what scene you're in. So in **ViewController.swift**, add this to the end of `viewDidLoad()`:

```
SoundController.shared.playBackgroundMusic("bulletstorm_bg_v1.mp3")
```

Here you access the shared instance of `SoundController` and play the music. If you had several scenes with different music for each, then you would add the background music to the scene instead of to the view controller.

Build and run the game.



It should look exacty as it did before, but have a background sound track.

(You can comment out

```
SoundController.shared.playBackgroundMusic("bulletstorm_bg_v1.mp3")
```

in `ViewController` when you've had enough :])