

Pygame - Dev Notes - Music - Intro

- Dr Nick Hayward

A brief intro on adding music to a game window with Pygame.

Contents

- Intro
- Game music and sound effects
- Import sounds and effects
- Use sound effects
- Use music in a game
- References
- Demo

Intro

For a game's sound effects, there are many different options and sources for these sounds.

We may try open source examples, such as

- [Open Game Art](#)

or perhaps create our own custom sounds using a utility such as **SFXR**, or its derivative website option,

- [BFXR](#)

Game music and sound effects

For most of these sound effects, we'll be using a *WAV* format for the sound files for these effects. We may also use other file formats such as *OGG*.

We can then add these files for our sound effects to the game assets directory,

```
|-- shootemup
  |-- assets
    |-- images
      |__ ship.png
    |-- sounds
      |__ laser-beam-med.wav
      |__ explosion-med.wav
```

Import sounds and effects

The first thing we need to add is support for Pygame's `mixer`. For example, we may add the following call after we initialise Pygame itself,

```
# add sound mixer to game
pygame.mixer.init()
```

To use these sounds and effects in our game window, we need to add the directory location, as we did for images. For example,

```
# relative path to music and sound effects dir
```

```
snd_dir = os.path.join(assets_dir, "sounds")
```

We can then start to add our required music and sound effects, again following a similar pattern to loading and using images. So, we can add our sound files as follows,

```
# load music and sound effects for use in game window
# laser beam firing sound effect
laser_effect = pygame.mixer.Sound(os.path.join(snd_dir, 'laser-beam-med.wav'))
# explosion sound effect
explosion_effect = pygame.mixer.Sound(os.path.join(snd_dir, 'explosion-med.wav'))
```

We can add these lines of code right after we've loaded our images, just before we start the game loop itself.

Use sound effects

After importing and loading our sound effects, we may then choose where we need to play these sound effects.

For example, as a player fires the laser beam to destroy falling mob objects, we may also call the appropriate sound effect. e.g.

```
# fire projectile from top of player sprite object
def fire(self):
    ...
    # play laser beam sound effect
    laser_effect.play()
```

So, each time a player now fires at the projectiles an accompanying sound effect will be played.

We can also add sound effects for each mob object explosion. We'll add the following `play()` call as part of the check for collisions,

```
# play laser beam sound effect
laser_effect.play()
```

We now have a sound effect for firing a laser beam, for example, and an explosion as each mob object collides with a projectile.

Use music in a game

As we add sound effects, we may also load music to play in the game.

For example, we may want music playing in the background whilst a player enjoys the game. So, we can load our required background music as follows,

```
# load music for background playback in game window
pygame.mixer.music.load(os.path.join(snd_dir, 'space-music-bg.ogg'))
```

We can also set a relative volume for this background music, in effect creating ambience and not overwhelming the player. e.g.

```
# set music volume - half standard volume
pygame.mixer.music.set_volume(0.5)
```

References

- [BFXR](#)
- [Open Game Art](#)
- [pygame.mixer](#)

- [Python API](#) - os

Demo

- basicmusic1.py
- basicmusic2.py
- shooter0.9.py
 - add music and sound effects to the game window
 - add pygame mixer
 - load sounds directory in assets
 - load required sounds and sound effects
 - call `play()` for each required sound effect and game music...