



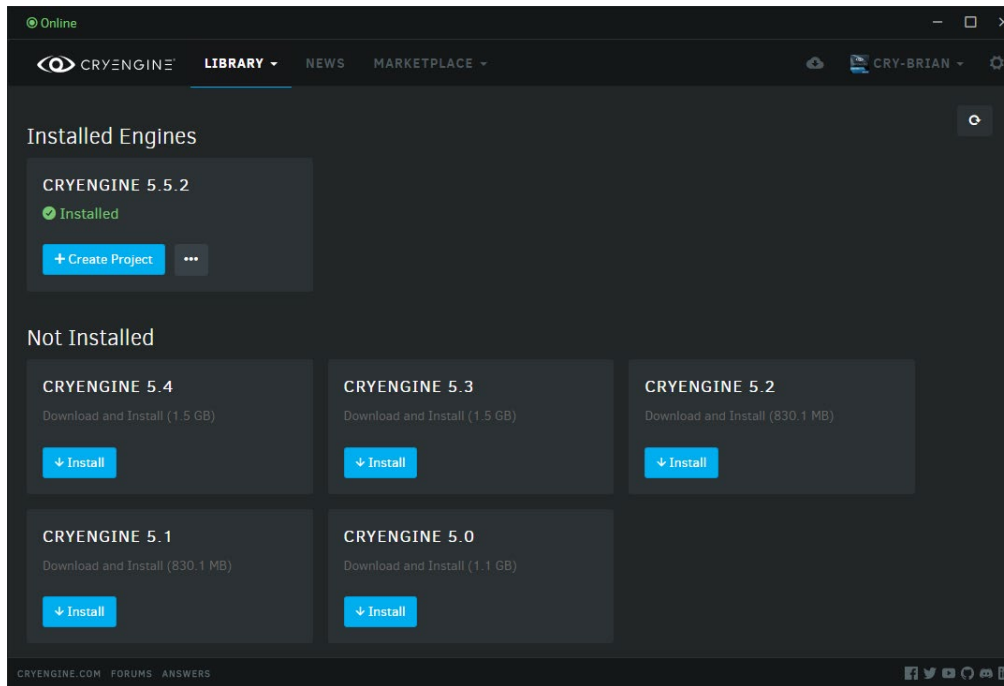
CRYENGINE®

Course Setup

Installing CRYENGINE

This course requires CRYENGINE **5.5.2** or higher. Here are the steps to install it:

1. Download and install the free CRYENGINE Launcher from [here](#) by clicking on the **Download** button. Please note that you will need to create a free CRYENGINE account if you don't already have one. If you need additional help creating your account, refer to our [Installation Quick Start guide](#).
1. Start the Launcher and **sign in** to your CRYENGINE account.
2. Install the latest version of the engine using the Library > My Engines tab in the Launcher.



Course Learning Objectives

Upon completion of this course, users will know how to:

- Navigate the CRYENGINE interface, Editor, and tool set
- Add ambient sounds
- Link ambient sounds to changing time of day
- Add audio triggers
- Link sounds to character actions (footsteps, shooting, etc.)
- Link sounds to particle effects

This tutorial covers the use of FMOD Studio. Please refer to the [online tutorial](#) for information on using SDL Mixer and FMOD.

Course Workbook Conventions

To help you understand the text in this workbook, we have used the following **formatting** conventions:

- Important concepts are highlighted in **bold**, or refer to something you can **do** or **click on**: a button, entity, menu item, etc.
- Nested items in **menus** or **tool panels** are **separated** with the **right caret character**, e.g.: **Tools > Level Editor > Create Object**.
- Concepts and goals are explained first, followed by the exact **step-by-step instructions** to achieve them. Tips about best practices look like this grey box.

Pro Tip:

This is a “pro tip” suggesting best practices or informing you about something of particular importance.

Prerequisites

This course assumes a working familiarity with CRYENGINE. If you're new to the engine, please download our ["Flappy Boid" beginner's course](#) from the CRYENGINE Marketplace and/or watch the [video tutorial series](#) on our YouTube channel.

You can also view the video tutorials for this course instead of or in conjunction with this workbook. We recommend that you watch the videos and consult the course book for additional clarification.

You also need to be comfortable navigating the Windows operating system, File Explorer, and using a mouse. Experience with 3D programs is useful but not required. Dual monitor setup will also be useful given the number of tools used, but is certainly not necessary.

Audio Tutorial #1: Setting Up Your Project

1. Open the CRYENGINE Launcher and create a new First Person Shooter C++ project.
2. Click on the **gear** icon next to your new project and choose **Reveal In Explorer**.
3. We need to download and unzip the assets for the audio showcase level in your new project. Open a web browser, navigate to the [documentation page](#) for the first chapter of the online tutorial, click on the [Showcase_Level.zip package](#), and download it to your project folder.
4. Once it's downloaded, **right-click** on the ZIP file and unzip it in the project root folder, **replacing** any existing file with the same name.

This file not only contains the showcase level, but it also includes some changes to the default setup, including deleting the reference to .animEvents from the player skeleton and setting **sound_obstruction** within the SurfaceTypes.xmls file to 0, as we won't be dealing with sound obstruction in the first parts of this tutorial.

Middleware Setup

This tutorial uses FMOD for example purposes; please refer to the [online tutorial in the documentation](#) if you need instructions using SDL Mixer or Wwise.

If you are already comfortable using your preferred middleware and just want to know how to integrate your finished audio assets into CRYENGINE, simply download the [Showcase Assets.zip](#) package into your project folder and **skip the sections labeled “ in FMOD.”** This ZIP file contains the **finished** middleware assets, with sounds imported, layered, and fully set up in the respective audio middleware applications.

Otherwise, to run through the entire tutorial on how to use your preferred middleware and how to set up your audio assets in CRYENGINE, download the [Showcase Audio Source.zip](#) file into your project folder and unzip it.

At the time this tutorial was created, CRYENGINE supports FMOD studio version **1.10.08**. At the time of writing, FMOD Studio's license agreement makes its commercial use free once annually for indie studios with game development budgets under \$500,000, but you should check their latest [licensing agreement](#) for details before you start a project for commercial release.

First, let's open your new project's **Assets\Audio** folder and **create** an “fmod” folder.

Inside the fmod folder, make sure you have two folders called **ace**, which is for our Audio Controls Editor, and **assets**, which is where our actual sounds will be stored.

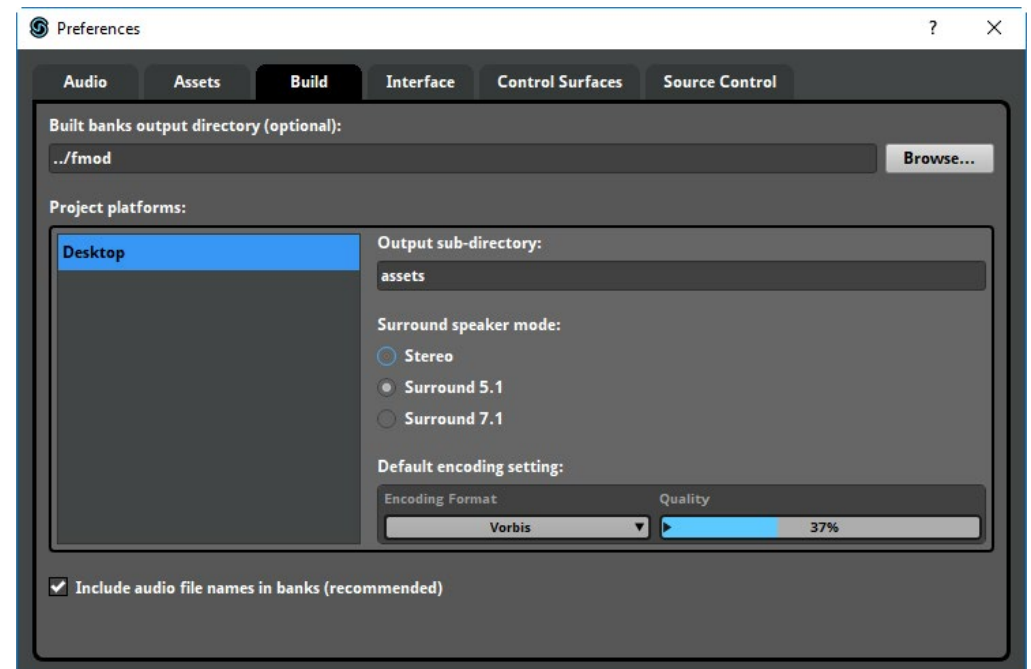
Pro Tip:

CRYENGINE's Audio Controls Editor (ACE) needs to know the path of the FMOD Studio project at all times in order to be able to read created Events and Parameters.

Launch FMOD Studio and create a **new project**. Name and save it in the **assets\audio** folder of your CRYENGINE project.

Let's set where FMOD saves Soundbanks so that they can be read by the engine:

1. From the FMOD main menu, select **Edit > Preferences**.
2. In the **Build** panel, enter **../fmod** in the Built banks output directory field. Or click the Browse button and find your Assets/Audio/fmod folder.
3. In the same panel under the **Project platforms** section, click the **Desktop** pa-



parameter and set the value of the **Output** subdirectory field to “assets.” Close the preferences panel.

4. Pressing the **F7** key should generate the Master Bank. **Note:** if you get an error, make sure you haven’t already downloaded the finished assets. For example, you will already see a Master Bank.bank file in the Assets\Audio\fmmod\assets folder, and FMOD Studio will not overwrite it.

Enabling Middleware in CRYENGINE

Now we’re ready to open our CRYENGINE project. Once it’s open, we need to tell the engine to switch from the current middleware (the default is SDL Mixer) to FMOD: open the **Console** tool. In **5.5 or earlier**, type the CVar `s_AudioImplName` followed by “CryAudioImpl” and the name of the middleware you want to activate. In this case, that’s **CryAudioImplFmod**; in your case, that could also be **CryAudioImplSDL-Mixer** or **CryAudioImplWwise**. In **5.6 and later**, the CVar is `s_ImplName`.

Ultimately, you’re going to need to make this happen outside of the editor when a player loads your game. The best place for that is in the `user.cfg` file, which you’ll find in the root folder of your project. Make sure the file isn’t read-only, then open it in a plain text editor and add the CVar and middleware you’re using, e.g.:

```
s_AudioImplName = CryAudioImplFmod
```

Note that this will also ensure that the correct middleware is loaded when you start the **editor**. If you are using CRYENGINE 5.6 or later, the cVar is slightly different:

```
s_ImplName = CryAudioImplFmod
```

There are actually **three different configuration files** where you can set console variables for your game, and you can also put them in your `.cryproject` file.

```
%USER%/game.cfg
```

```
root/system.cfg      Used by both Editor and Launcher
```

```
root/user.cfg
```

You can read [this thorough page](#) in the documentation to learn more about console variables and configuration files. And by the way, console variables are **not** case-sensitive, although we usually write them with initial caps just to make them easier to understand.

Opening the Level and Audio Library

Open the audio showcase level in your CRYENGINE project: File > Open > Levels > showcase_audio.

You’ll see that this level is a pretty simple low-poly garden with a few walkways, a campfire, a marketplace behind a closed gate, and a forest and mountains around the enclosure. A player is defined that can navigate and shoot.

Open the **Audio Controls Editor** from the **Tools** menu. Your **triggers** are on the left, which comes from the Assets\Audio\fmmod\ace folder, their **Properties** are in the middle, and the **raw data** is on the right, which is the contents of your Assets\Audio\fmmod\assets folder.

The basics of the Audio Controls Editor are covered in the [audio chapter](#) of the Flappy Boid course. Just keep in mind that your controls all have to be stored in a library. On a new project, you can add libraries by **right-clicking** in the **Triggers** panel and choosing **Add > Library**, or by clicking on the big **Add** button.

Creating an Event and Soundbank in FMOD

We’ve supplied a suitable raw sound for the shooting event: `_Audio_Assets/FMOD/chapter_01/w_gun_fire_01.wav`. Let’s create a new event and sound bank in FMOD:

Drag this .wav file from the Explorer window into the **left** side of the **Events** tab in your FMOD project. The **Create Event** window will open. **Select 2D Event** and click the **Create** button. You’ll note that all audio assets that you add to an FMOD project will be listed under the **Assets** tab.

Now that we’ve defined an event, we need to create a **soundbank**. Soundbanks store meta-data related to audio assets as well as containing compressed, encrypted audio.

Open the **Banks** tab on the left of your FMOD project. **Right-click** in the left panel under Master Bank and select **New Bank**. Let’s call the Soundbank “Weapon.”

Save the FMOD project, then switch back to the

Pro Tip: 2D vs. 3D Events

The reason behind gunfire as a 2D Event is that we’d like it to sound like it’s wherever the player is - i.e., we don’t want it to be positioned anywhere in 3D space. Creating a 2D Event allows us to place the Audio Trigger **anywhere** in the level, but it will always fill the headphones without coming from any particular direction. Note that this works only with a single player level; it isn’t practical for multiplayer settings.

Events tab of the FMOD project. An *#unassigned* tag next to the Event reminds you that it hasn't been associated with a Soundbank yet. Let's fix that:

Right-click on the `Play_w_gun_fire_01` event and select **Assign to Bank > Weapon**. Press **F7** to build the created Soundbank.

Creating an Audio Trigger in CRYENGINE

For our event to be triggered in the level, it needs to be connected to an audio Trigger in the Audio Controls Editor. In the ACE, you should see the folder structure of your FMOD project in the Middleware Data panel, with the `Play_w_gun_fire_01` event listed under the Events folder and the Weapon Soundbank under the Soundbanks folder. Let's create a trigger:

1. **Right-click** in the Controls panel and select **Add > Library**. Name the library "Weapon."
2. **Drag** the `Play_w_gun_fire_01` event from the Middleware Data panel and **drop** it onto the **Weapon** Library.

A green **Trigger icon** will appear next to your new Trigger. But for it to work, the Weapon Soundbank also has to be imported:

1. **Create** a new **Library** in the Audio System Control panel and call it "Preloads."
2. **Drag** the Weapon Soundbank from the Middleware Data panel and **drop** it into the Preloads library. This creates a Preload Request with the same name in the Audio System Control panel, and insures that the soundbank will be preloaded when the level is opened.
3. **Save** your audio libraries. Here's a tip: **you can only save audio libraries using this save button in the ACE**. Ctrl-S only saves the current *level*, **not** the audio libraries.

Save both the Weapon and Preloads Libraries by clicking the SAVE button at the upper left corner of the ACE. To preview the created Trigger in the ACE, first refresh the audio system by clicking the speaker button at the upper left corner of the ACE.

To test a trigger, the easiest way is simply to click it, then press the **spacebar** key. If you want to adjust the volume of a trigger, you have to do that back in FMOD:

1. Press **F5** to switch to the **Designer** layout.
2. From the **Audio** tab of the Project Explorer, **select** your `Play_w_gun_fire_01` event. Use the volume fader in the panel on the right to adjust the volume.

3. Once you're done, **save** your FMOD project and **rebuild** your SoundBanks with **F7**.
4. Don't forget to **reload** the audio engine in the CRYENGINE ACE.

To change the volume of the `Play_w_gun_fire_01` sound:

1. Open FMod Studio and **click** on the `Play_w_gun_fire_01` event in the **Events** tab.
2. Use the **volume dial** on the Event's Audio 1 track or the bigger dial at the bottom of the window to change the master volume of that event.
3. Changes can be previewed in FMOD by pressing the **spacebar** with the Event selected. Remember to **save** the project, **rebuild** Soundbanks and **reload** the audio engine on the ACE to ensure that all changes are carried over to CRYENGINE.

Adding Tree Movements in FMOD

Again, this tutorial just covers FMOD setup; refer to the documentation for step by step instructions on setting up SDL Mixer and Wwise.

In the `_Audio_Assets\FMOD\chapter_01` folder, you'll find four audio tracks named `l_cg_amb_one_shot_tree_movement_0#.wav`.

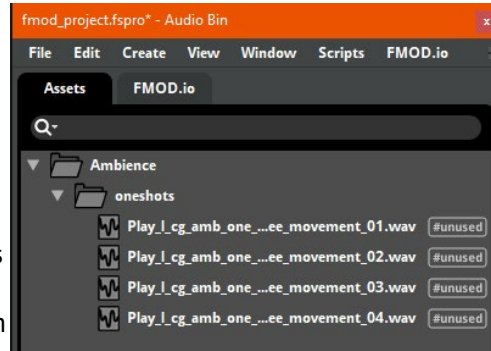
These sounds are four variations on the sound of leaves rustling in the breeze that we want to use as part of our background ambience. What we need to do is assign all four of them to the same 3D event in an FMOD SoundBank, and have the event randomly select between the four variations.

Creating an Event and SoundBank in FMOD

1. **Select Window > Audio Bin** in the FMOD main menu (or press Ctrl + 3). The Audio Bin is where an FMOD project's audio assets are stored.
2. To keep the Audio Bin organized, **create a folder** named "Ambience." Within it, **create another folder** called "oneshots." **Drag** the four audio asset files `Play_l_cg_amb_one_shot_tree_movement_#.wav` from an Explorer window into the **oneshots** folder.
3. Click on the **Events** tab in FMOD Studio. Just as we did in the Audio Bin, **right-click** and **create a folder** called "Ambience," and a "oneshots" folder within it.
4. Next, drag the four audio asset files `l_cg_amb_one_shot_tree_movement_0X.wav` **from the Audio Bin into the oneshot folder of the Event tab**. As before, this opens the **Create Event** window with the option to create a 2D Event or 3D

Event.

5. **Select 3D Event** in the Create Event window.
6. Under **Additional options**, check the **Create a new event with one multi-instrument**, then click **Create**. A multi-instrument event randomly plays one of its assigned audio files each time it's triggered, creating variation in the audio trigger.
7. **Double-click** and **rename** the Event to **Play_I_cg_amb_one_shot_tree_movement**.
8. Press the **spacebar** to **test** your sound event. You'll notice that the event **randomly chooses** from the four sounds (A, B, C, or D) each time it's triggered.



As with the Play_w_gun_fire_01 Event created in the previous step, a **Soundbank** is required to use and hear the Play_I_cg_amb_one_shot_tree Event in our CRYENGINE project:

1. **Create** a new bank called "Ambience" in the **Banks** tab of your FMOD project
2. **Drag** the Play_I_cg_amb_one_shot_tree Event from the **Events** tab into the **Ambience** Soundbank in the **Banks** tab. (Drag the event onto the Banks tab name and hold the cursor there to activate the Banks tab, and then continue dragging onto the "Ambience" soundbank.
3. **Save** changes, and then press **F7** to build all soundbanks.

Ambience Event and Soundbank

If you switch to the Audio Controls Editor in CRYENGINE, you'll now find the Play_I_cg_amb_one_shot_tree Event listed under Events > Ambience > oneshot in the Middleware Data panel. Let's connect this to an audio trigger:

1. **Create** a **Library** called "Ambience" in the Audio System and within it, a folder named "oneshot."
2. **Drag** the Play_I_cg_amb_one_shot_tree Event from the Middleware Data panel into the oneshot folder on the Audio System Controls panel.

The Event's Ambience Soundbank containing its audio playback information needs

to be preloaded into memory for it to function. Let's create a Preload Request:

1. In the ACE, **right-click** and **create a new Library** called "Preloads," then **drag** the "Ambience" Soundbank from the Banks folder in the Middleware Data Panel into it.
2. **Save** the audio Libraries and preview the Play_I_cg_amb_one_shot_tree Trigger in the ACE.

Audio Trigger Spot

Now we need to add an Audio Trigger Spot (ATS) entity to trigger our audio event at a specific position in the level:

1. In the **Level Explorer**, **double click** on the layer where you want your audio trigger spot to be saved. We suggest the **global_audio** layer.
2. From the **Create Object** tool panel, select **Audio > Audio Trigger Spot**. Click anywhere on the ground of the Audio Showcase level to place the new audio trigger spot.
3. Make sure your new audio trigger spot is selected, then in the **Properties** panel, click on the browser button next to the **PlayTrigger** field and **select** the **Play_I_cg_amb_one_shot_tree** Trigger in the ambience > oneshots folder.
4. The Play_I_cg_amb_one_shot_tree sounds should now be audible both in the Perspective Viewport and in Game Mode. If you can't hear it, reload the audio engine by clicking on the Refresh Audio System button on the toolbar of the ACE.
5. If you want the Trigger to execute more often, set the value of the **Behavior** property to **TriggerRate**. This causes the Trigger to execute every x seconds after the previous Trigger instance, where x is a random number of seconds between the **MinDelay** and **MaxDelay**.

For example, try setting the values to 1000 and 100,000 so that the tree ambience is re-triggered every 1-100 seconds.

NOTE: Although the **MinDelay** and **MaxDelay** values are defined in seconds, they're specified in milliseconds within this tutorial due to a bug in CRYENGINE version 5.5.2 that produces unexpected results.