Game Audio

Tutorial 4 – DIG6041

# Pre-requisites

* Basic understanding of Digital Audio (Games Technology students)
* Basic understanding of the role of FMOD in the Game Audio toolchain.

# Topics

* Getting to learn FMOD Studio Suite
* Understand the power of FMOD as audio middleware and avoid using it as a traditional DAW.
* Create a series of FMOD events to simulate developer requests.

# FMOD Basic Setup

Basic components of the FMOD GUI and a bit of clarification on the terminology.

* The importance of the Audio Bin and organising sounds.
* FMOD and the different kinds of events:
  + 2D VS 3D Events
  + Action VS Timeline Events
  + Are events different?
* FMOD Sheets:
  + Parameter Sheets
  + Action Sheets

# FMOD Event Module Types (Instruments)

A brief overview of all the types of modules that can be used inside an event, and all the different strengths of each module:

* Single Instrument – simple single-sample player
* Multi Instrument – multi sample player with randomisation and probability features
* Scatterer Instrument – more powerful multi instrument with spatial features
* Audio Weather – VST-like module that generates sound rather than playing samples
* Event Instrument – events can be nested in events and used as modules

## Exercise 1 - Delivery of a programmer-requested FMOD event

Create an event that plays a gunshot sound and a car siren caused by said gunshot.

## Exercise 1a:

Create the event as described, trying to use the previously taught modules when needed.

## Exercise 1b:

The programmer adds that the event will be played more than 15 times in the level they are designing. Can you try to find a way to add some level of variation to make the repetitions less obvious?

## Exercise 1c:

Can you think of a way of making the siren continue indefinitely until the event is stopped?

# Randomisation and Parameter Changes

We can randomise almost any parameter in the event by right clicking on the control of said parameter and setting a range for the parameter to randomise. These changes can be performed also with other modifiers (i.e. LFO) but for now we’re mostly interested in randomisation.

* Randomise the volume of the gun explosion from the previous exercise
* Randomise the pitch of the fun explosion from the previous exercise

# Automation and Parameter Sheets

Automation is typically done tying a specific parameter to time, in traditional DAWs. For example, if we were automating volume of a podcast to remove the noise typical of the silent bits of the track, we would be effectively tying volume to time.  
  
In FMOD, we are not bound by time. We can tie any parameter we want to custom variables (called parameter sheets) that can then be referenced from the game to make sure that the game is responsible for the dynamic change of our sound parameters.

We showcased this in class by adding a “Witness” variable to our siren part of the gunshot, making the siren play only when the witness was “present”.

* Parameter sheets (continuous, discrete, labelled)
* Automation curves and ramps
* Notes to communicate “remotely” with the developer and to comment on the event features

## Exercise 2 - Creatures

In the audio bin you should have some small creature and large creature sound files. Create a new event, separating the two “creature types” in two different tracks.

## Exercise 2a:

Can you think of. Away to play small creature sounds during the day, and large creature sounds during the night?

## Exercise 2b:

Instead of playing the sounds separately, can you think of a way of combining the sound during the night using automation? (Hint: all the automation can be done on the small creature sounds).

* Using parameters to play events from the beginning of a game scene and triggering different sounds according to a specific parameter value.
* Stacking events and parameters to have multiple variables affecting sound parameters and event playback

# Built-in parameters

Normal custom parameters need to be created in FMOD and implemented into Unity (see Kart engine example with RPM parameter).

Built-in parameters automatically send a defined set of values to FMOD without any additional programming.

* Direction
* Orientation
* Elevation