GameSoundCon 2010: Procedural Sound Design

Leonard J. Paul
Vancouver, Canada

http://VideoGameAudio.com

http://twitter/VideoGameAudio

http://school.VideoGameAudio.com
Overview

- **Creative ::** How to think about audio for games from an artistic standpoint

- **Technical ::** Make the sound design come to life within the limits of the hardware & software

- **Process ::** The steps used to combine the creative and the technical into the final sound
Creative

- Top-down or bottom-up approach
- **Top-down:** Figure out your “big idea” and worry later about how to actually do it
- **Bottom-up:** Discover what the technology is capable of & figure out how to get the most of it
- Often you'll be going back and forth between the two as technical issues and process limitations will force you to be more creative
Technical

• Find out the limits of the software and hardware and push them

• Expand the software capabilities and design better audio tools to suit the game

• Realize that often you will be pressed by the team for memory & CPU, so hold your ground

• Things can work one day and be broken (and useless!) the next
Process

• What do you need to do to get your audio into the game? Over-prepare for the worst

• What resources will support your creative design? Work well with your coder & producer

• A good process can be found by making all the right mistakes once and taking risks - exciting!

• Audio is always last, so be prepared for feature drop, no money, no time & no love :)
Example Demonstrations

• Let's see how audio might be done for a current game, such as Rockstar's western action game *Red Dead Redemption*
Simple Ambience

- Creative challenge is how do we make a realistic sounding ambience that can be heard for a long time?
- Technical challenges include how to fit it in memory, data compression type, making a seamless loop, mixing it at the right level and making it transition smoothly to the next ambience
Loop Editing

- Looping using Wavelab (now available on Mac):
  - Start/End
  - Match slope
  - Live audition
  - Auto x-fades
  - Crossfade loop
  - Volume flatten
  - Pitch flatten
  - Spectral edit

< www.Steinberg.net >
Layered Ambience

• A layered ambience is made up of several different loops
  – Country ambience could consist of frogs, wind, birds and flies
• Reactive ambience would have frogs and birds disappear for a while after a gunshot
• Flies would decrease with wind and vice-versa
Procedural Sound Design

• Generate audio in real-time based on rules and behaviours
• Has been used in games since the beginning
• Parametric, generative & real-time (ie. flexible)
• Can combine samples with synthesis
• Allow flexibility of layers and real-time effects instead of stereo mixdown
Time-of-day Ambience

- City ambience of horses, dogs, distant wolves and crows could react to time
- Crows and wolves would trigger randomly with more crows during the day and wolves at night
- Horses would be more present during the day and dogs during the morning and early evening hours
- This method would more likely be used in a mobile, handheld or online title
Gunshots

• Creative challenge is the difficulty of making a loud sound frequently heard sound satisfying

• Use layering of recordings and include a low-end sweetener with printed effects for maximum impact

• Can use real-time effects to add variation and cue location
Granular Concatenation

- Split the sound into grains on transients and concatenate them randomly in real-time:
Granular : Speech

- Slightly change the tempo contour of a line of speech while playing back to increase variation:
Simple Synthesis

• In Retro City Rampage almost all sounds are synthesized from pulse, triangle and noise waves

• Mod tracker allows mix of synthesis and samples but is also tricky to learn

• The Unity game engine now allows mod files which is good for online, handheld & mobile
Train Sounds

• Use loops for far, medium and close recordings of train sounds
• Crossfade in real-time between depending on distance
• Add slight playback rate modification to loops for additional modulation
• Add train track sound
• Add reverb from environment
But...

• Often there isn't enough time, so people will go with “what they know” (ie. samples) and advanced techniques are overlooked

• Long time required for development and tuning

• Leap of knowledge and time to learn for those invested in older techniques

• Requires more CPU for DSP and synthesis

• Balance creative & technical ? =>
Current Research

• Utilize a hybrid method of samples plus modal & granular synthesis to make audio for games (Cécile Picard et al.):

Demo Review

• Using Pure Data:
  – Procedural ambience
  – Time of day ambience
  – Granulation: Horse's hooves
  – Engine: Trains

• Now, let's examine *Half-Life 2* and OSC:
  – Granulation, layering and more
HL2-OSC Game Audio Pipeline

Game Code (C++)

OSC Layer

Game Audio Code (Pure Data)

OSC Layer

Sound Driver Code (PD)

Commercial Sound Tool

X,Y,Z

game events

sound events

.wav

// LEN: Send trace via OSC

if ( ep.m_pflSoundDuration )
{
  *ep.m_pflSoundDuration = engi;
}

Trace Emit Sound("Emit Sound: Raw:
ep.m_pSoundName, entindex");
Pure Data (PD)

- Relatively easy to learn open-source interactive audio implementation environment
- Can integrate directly into game code (as in EA's Spore), but not optimal as it is interpreted
- Create compiled code from (simple) patches as shown by pd2j2me
- www.puredata.info
- Get “PD Extended”

```
public void met0Bang() {
    double a;
    a = counter.bang();
    a = a * 3;
    a = a / 4;
}
```
Half-Life 2

- *Half-life 2* is a PC/Mac/360/PS3 game that won over 40 “Game of the Year” awards when released in 2004 and allows users to change the maps and other content of the game using the “Source engine” technology in C++

- *Left 4 Dead* and other games based on Source

- Make your own mods using the HL2 SDK
**Half-Life 2 Source Mod**

- A mod is a modification of the game that can be quite simple or be advanced enough to create an entirely new game (ie. total conversion)

- Change code using Microsoft's free compiler *Visual C++ 2010 Express Edition*

- Cost for replicating this system is just the cost the game - $9.99
HL2 Demonstration
Unity

• *Unity* is a game building platform on the Mac and PC for games on the Wii, PS3, 360, PC, Mac, web, Android and iPhone.

• The non-pro version became free in Oct '09

• Disadvantage vs. HL2 OSC is code must be added for each game

• Fmod used as sound driver
Realtime 3D Preview

- Create soundscapes in 3d editor and listen in realtime
- Makes it much easier to tune reverb areas and ambient sound sources
- Easily see sound spheres
- Tune rolloff curves using tables
Reverb Areas

- Areas in which different reverb types are applied
- Can view area of influence in real-time and tune rolloff curves
- Default settings to create different types of reverb
- Areas may overlap
- Only available in pro version of Unity
VST and PD

- PD can load different VST and VSTi plugins.
- Can make complicated chains of soft synths and real-time effects for prototyping.
- Control VSTs by various game parameters.
- Can “extend” power of PD by using other plugins via VST such as Reaktor.
Unity-OSC Audio Pipeline

Game Code (C#, Boo, Javascript)

OSC Layer

Game Audio Code (Pure Data)

Sound Tool

Sound Driver Code

3D Editor

OSC Layer

mp3/ogg/mod

sound events

if (oscHandler == null)
{
    throw new System.NullReferenceException();
}

oscM = oscHandler.StringToOscMessage();
oscHandler.Send(oscM);
Questions?

info { at } VideoGameAudio.com

web: VideoGameAudio.com

School.VideoGameAudio.com

twitter.com/VideoGameAudio