Noise is Good for You!
Introduction to Sound Effects for Video Games
Presented at the August Twin Cities Chapter Independent Game Developers Association Meeting by Damian Kastbauer (LostChocolateLab)

Overview
Hi and welcome to an introduction to sound effects for video games. This isn’t meant to be a comprehensive introduction to all aspects of sound as they relate to video games, but hopefully will give you a overview of some of the things you might give thought to during you game playing and creating.

Introduction
The seeds were planted early on, a penchant for experimentation, creative problem solving, a deep organizational psychosis, and unique ear for the world. Over the years it grew out of guitar based music steeped in effect pedal ethics, ambient experiments in feedback and noise, taken to the extreme with no-input pedal board noise feasts and finally ending up being implemented as sound effects in video games.

I look back over the development of sound in my life and it seems like such a natural progression.

At the crossroads of a career change I asked myself:
"Is there a place that will pay me for making weird noise?"

Like the Wizard of Oz, my world swiftly returned to the pixelated technicolor landscape of my youth where for days on end I was transported to alien lands, seeking untold fortune, submersed in a sonic landscape of squonking bleeps and sqeeking blops, fending off challenge after challenge in an endless stream of disproportionately oversized end-bosses on my way to the goal that lie always just out of reach.

It was here that I could find my place in life...the decision was clear.

Since then it's been a steady stream of experience and hard work.

Late nights spent sculpting sonic goodness out of raw materials, coffee addled, sleep deprived, phasers set to kill. Sounds sourced from nature, molded by man & machine, the power of the universe unlocked in a waveform. This is what I’m thinking as the cacophony surrounds me in a womb of subterranean safety.

Brief History
While sound for video games will never be as visceral as the sound coming from a Frogger cabinet a dive bar in the 7th year of my life, I like to think that I continue to hear glimpses of that moment in the spontaneous noise feasts brought on by modern gaming. It may have started with a single sinewave triggered by the contact of ball on paddle, continued in custom synthesizers built into game computers, escalated to samples of increasingly more realism…but it will always be the part about games that suspends my disbelief, if for a moment. The sounds of the world have shut off, and here is a new world, with sounds not like those I’m used to.

Questions/Caveats
I’m not an expert, I don’t know everything. I’ve been working with sound for video games a little less than a year. I’ve assisted on 15 projects, seen 6 through to competition, am currently working deeply on 3. I haven’t developed for any platforms, I’ve never had to worry too much about file size, and I don’t do music. Other than that, I am passionate about noise, sound effect implementation is the new experimental music for me, and I love working with creative people dedicated to their art.

Industry
Job Titles
Composer/Sound Designer: The eternal quest for a person capable of doing everything sound related on a given project.

This is a tough one out there today. In an industry that is still very much like the Wild West, it is hard to pin anyone down with the specifics of what they do. Assumptions are made, responsibilities flex and grow, roles are shared, budgets are cut, and all of a sudden everyone can make weird noises.

Some of the titles we’d like to see used would be:
Music Lead/Director/Manager: Scheduling, Budgeting, Overall Scope, Implementation
Composer: Someone who writes music.

Audio Lead/Director/Manager: Scheduling, Budgeting, Overall Scope, Implementation
Sound Designer/Sound Engineer: Creates with sound, Edits Sound

What I’d like to see more of is recognition of the different disciplines in game audio, and seeing the sound arts as equally valuable.
Work Flow

Pre-Production

Pre-production for audio, when does it start?
The easy answer is, when everything else does. Audio can be as integral to the game design as any other aspect of it’s planning.
Someone who follows the pre-development of a game has a chance to exact changes, or at least put up flags that may make for
smoother implementation down the road. The point here is that making a game is usually a collaborative process, and the audio people
(will) have as much invested as everyone else by the end of the project. Include them early and be rewarded.

Design Docs

Ground Zero

Brainstorming /Fuzzy Spec

This is the time when ideas get thrown around, a loose formulation of all aspects of game design are churning in an endless upheaval
while attempts are being made to nail down what will make the game.
Questions you or your audio team might be asking during this phase might be:
What is the game trying to get across with sound?
In what ways could sound enhance game play?
What has been done before, what hasn’t?

There’s nothing radical here except for the idea that sound & music could contribute to a fuller gaming experience.
A "'''Fuzzy Sound Doc'''" could look like this:

1. Ambient Water
   * Waves, splashes.
   * Waves cracking the rocky coast (when it is rock).
   * Running water, waterfalls.
2. Siege
   * Onager moving.
   * Onager being destroyed.
   * Onager being selected. Possibly mixed with voices of Siege Engineers.
   * Onager shooting (grinding and cranking sound of weapon being drawn back and prepared for attack).
   * Onager reloading (sounds of stringing ropes and similar).
   * Ballista shooting bolt.
   * Ballista being selected. Possibly shared with Catapult.
   * Ballista being destroyed. Possibly shared with Catapult.
   * Ballista moving - Don't know if it will even roll, possibly pack and unpack.
   * Ballista reloading. Possibly shared with Catapult.

Defining Assets

Kinds of Assets
Music, HUD/UI, Character, Ambient, Object, VO

Organization (Spreadsheet example)
Suggestions of what to track:

<table>
<thead>
<tr>
<th>Group Name</th>
<th>/Folder Location</th>
<th>Description</th>
<th>Biome</th>
<th>Season</th>
<th>Time of Day</th>
<th>Intensity</th>
<th>Stereo/Mono</th>
</tr>
</thead>
<tbody>
<tr>
<td>Duration</td>
<td>Loop</td>
<td>Random</td>
<td>Gain</td>
<td>Pitch</td>
<td>Pitch-Range</td>
<td>Entity</td>
<td>Action</td>
</tr>
<tr>
<td>Event Start</td>
<td>Event End</td>
<td>Priority</td>
<td>Assigned to</td>
<td>Assign Date</td>
<td>Status</td>
<td>Complete Date</td>
<td></td>
</tr>
</tbody>
</table>

Naming Standards (Relevance vs. size)
Ask for experience

Directory Structures

Storage/ Access
Logical Progression

Defining Tech

Audio Engines

Basics/ Examples/ Differences
An audio library is a software interface to audio hardware. The interface consists of a number of functions that allow a programmer to
specify the objects and operations in producing high-quality audio output, specifically multichannel output of 3D arrangements of
sound sources around a listener. Essentially a jump start on the low level functionality of the programming for an environment where
sound plays a role. Audio engines deal with the playback, conversion, repetition, and other characteristics of the audio assets, as well
as the way those assets react to the defined state of the listener.

Different Engines for different things;
OpenAL has a solid basis of limited, well defined parameters for open source platform independance.
FMOD includes similar standard definitions, but also includes more extensive DSP/ effects.
ActiveX/DirectX is Microsoft’s proprietary audio engine.
Xact the Xbox Audio Creation Tool
Scream the Sony SCRiptable Engine for Audio Manipulation
Parameters

Formats

Streaming vs. Buffers vs. Sound Banks

Implementation I

Theory

Implementation is the way that sound is transferred from the static sound captured in a sound file, to the living breathing alternate reality that is the game world. This is as easy as a cause and effect relationship between two objects, and as difficult as the audio modeling of an object and the sounds it would create in realtime. Cause and effect, easy to explain: fire gun – hear shot. Beyond that there is a whole world of intricacies when you are dealing with the realistic playback of sound in an interactive environment. I stood and watched a Buffalo idling in a field eliciting no less than 5 individual sounds, just standing there.
1. Panting in the hot sun.
2. Movement
3. Tail Wagging
4. Grass Chewing
5. Grunting

Then there was the flies buzzing, the bird sitting on his back pecking the bugs off, and the sound of the wind in the tree’s. To attain a truly realistic representation of that scene would require 8 types of sounds, with variation plus randomization and then there’s making sure they play at the right time. That is implementation; the dissection of a sound scene and the way you go about reproducing it interactively and believably.

Trigger/ Rollover/ Zone

How do we accomplish this? We’ve got our afore mentioned parameters, there’s what we can bake into the sound file itself, and then there are the action/ reactions in-game that can act as our cause and effect.

Trigger: A location or event based call for action. Can be a placed directly on a map or in a sequence of events, and will then play back the associated audio file.

Rollover/ Mouse over: Another kind of trigger. Like on a website where a cursor moves over the top of a graphic or location and a sound is triggered.

Zone: Zones allow for different area specific sounds tied to locations on a map. An example would be footsteps. If the player is walking on the beach in the zone associated with sand then play the “sand footstep” sound, if they walk out into the lapping waves of the water zone play “water footeps”. In this way you can define different sounds for the same action based on the map location or in this case material represented.

One Shot/ Looping

XML/ IXML

Extensible Markup Language. I am not the expert, but I see this as an easy way to deal with the process involved with the programming of events in relation to objects and entities that interact. XML provides an assignable common language that can be used for the programming of events.

Integrators

iMUSE, GameCODA, EAGLE, EAX
Noise is Good for You!
Introduction to Sound Effects for Video Games

Overview
  Introduction
  Brief History
  Questions/Caveats

Industry
  Job Titles
  Common Words/ Concepts

Work Flow
  Pre-Production
    Design Docs
      Ground Zero
      Fuzzy Spec
      Brainstorming
    Defining Assets
      Kinds
      Organization
      Naming Standards
    Directory Structures
      Storage/ Access

Defining Tech
  Audio Engines
    Basics/ Examples/ Differences
    Parameters
    Formats
      Wav/Aiff
      Mp3
      Ogg
      GMidi/ SPMidi
      XMF
    Streaming vs. Buffers
    Sound Banks

Implementation I
  Theory
  Trigger/ Zone/ Rollover
  One Shot/ Looping
  XML/ IXML
  Integrators
Production
  Asset Creation
  Techniques
    Foley
    Synthesis
    Editing
    Manipulation
  Implementation II
    In-Game Feedback
    Adjustments/ Parameters

Team Work
  Management
    Communication
      Techniques
      Tools
  People
    Play the Game
    Work Together
    Challenge Each Other

Examples
  Playful Minds
    Elemental Wars
      Basic Trigger/ Buffer Limitation
    Castle Cards
      Multiple Animation Trigger
    Valandil
      HL2 Engine/ SDK & Scheme
  0.A.D.
    Priority
    Intensity
    Tools
    XML
    Scripts

Closing

Questions
SFX Drama Reinactment

Hero
Steps grass the gravel
Add armor
Unsheath sword
Whip around
Dragon walk/ growl
Whip/slice flesh impact
Dragon Death yell
Triumph sound

Shooter
Steps
Unholster
Check clip
Fire
Ricochet
Impact/flesh
Death yells
Broody ambient

Bring:
Audio recording Equipment
Paper Handouts 11X17 (Color)
    Audio Engines
    Web Links/ Resources
    Formats

CD’s of Clint Bajakian Speech
Books

Vocal Assets
Game Videos
Quicktime