THE DESIGN AND IMPLEMENTATION OF SOUND, VOICE AND MUSIC IN VIDEO GAMES

A Non-print Media Project

Presented to the Faculty of California State University, Chico

In Partial Fulfillment of the Requirements for the Degree Master of Arts in Music

by

© Matthew David Copeland 2012

Fall 2012
THE DESIGN AND IMPLEMENTATION OF SOUND, VOICE AND MUSIC IN VIDEO GAMES

A Non-print Media Project

by

Matthew David Copeland

Fall 2012

APPROVED BY THE DEAN OF THE GRADUATE SCHOOL AND VICE PROVOST FOR RESEARCH:

Eun K. Park, Ph.D.

APPROVED BY THE GRADUATE ADVISORY COMMITTEE:

Lloyd S. Roby, D.M.A., Chair

David Dvorin, M.F.A.
PUBLICATION RIGHTS

No portion of this non-print media project may be reprinted or reproduced in any manner unacceptable to the usual copyright restrictions without the written permission of the author.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Publication Rights</td>
</tr>
<tr>
<td>Abstract</td>
</tr>
</tbody>
</table>

## CHAPTER

### I. Introduction

- Non-print Media Acknowledgement | 1
- Interest and Problem Statement | 1
- Scope and Audience | 2
- Purpose | 2
- Limitations | 3

### II. Video Tutorial Supplement

- Explanation | 4
- Video Game Industry - Perspective | 4
- Console Development Challenges | 5
- File Format and Fidelity | 7
- Asset Procurement | 7
- Editing Practices | 8
- Batch Processes | 10
- Sound Design and Implementation | 10
- Audio Middleware | 11
- Scripting | 12
- Voice | 13
- Music | 14
<table>
<thead>
<tr>
<th>CHAPTER</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>III. Treatment</td>
<td>15</td>
</tr>
<tr>
<td>Narrative Video Tutorial Assembly</td>
<td>15</td>
</tr>
<tr>
<td>Equipment List</td>
<td>16</td>
</tr>
<tr>
<td>IV. Summary, Conclusions and Recommendations</td>
<td>17</td>
</tr>
<tr>
<td>Summary</td>
<td>17</td>
</tr>
<tr>
<td>Conclusions</td>
<td>17</td>
</tr>
<tr>
<td>Recommendations</td>
<td>18</td>
</tr>
<tr>
<td>References</td>
<td>19</td>
</tr>
<tr>
<td>Appendices</td>
<td></td>
</tr>
<tr>
<td>A. Sheet Music</td>
<td>22</td>
</tr>
<tr>
<td>B. Copyright Citations</td>
<td>31</td>
</tr>
<tr>
<td>C. Statements of Legal Clearance</td>
<td>33</td>
</tr>
</tbody>
</table>
ABSTRACT

THE DESIGN AND IMPLEMENTATION OF SOUND, VOICE AND MUSIC IN VIDEO GAMES

by

© Matthew David Copeland 2012

Master of Arts in Music

California State University, Chico

Fall 2012

The non-print media project The Design and Implementation of Sound, Voice and Music in Video Games is designed to inform collegiate and trade school graduates in related fields of study on the common practices and processes employed by audio professionals in the video game industry, in addition to introducing a set of standard software tools used to design, edit and implement audio into video games. While audio-related curriculums provide a fundamental understanding of sound engineering and recording practices, there remains a dearth of information and general instruction on how these core competencies can be made applicable for use in the game industry. This project endeavors to fill that perceived gap, and provides a jumping off point to further individual study for the aspiring professional interested in a career in the field of game audio.
The project consists of eight narrative video tutorials that collectively span approximately one hour and forty minutes in length. The tutorials should be watched in numerical order, as concepts introduced in earlier segments will be revisited and built upon as the project progresses. All examples of music composition, voice work, level design, system design and sound design are original. The project shifts focus back and forth between the Unreal Development Kit (and its propriety sound engine) and FMOD Designer 2010 (an audio middleware sound engine), effectively comparing and contrasting two industry standard audio implementation environments. Finally, the project demonstrates the value of approaching audio implementation in games as an essential and creative form of design.
CHAPTER I

INTRODUCTION

Non-print Media Acknowledgement

The following text is best used in tandem with the non-print media project: *The Design and Implementation of Sound, Voice and Music in Video Games*. The body of this document serves to summarize and in some ways expand upon the topics introduced in the project (a set of video tutorials), but should by no means replace them. Along with incorporating insight from audio professionals currently working in the game industry, I will offer my perspective as a sound designer and fellow industry professional of more than four years (currently employed by Activision Blizzard).

Interest and Problem Statement

I became interested in a study of game audio design and implementation techniques as I left California State University, Chico (after completing a B.A. in Recording Arts) to find work in the video game industry as a sound professional. While unable to initially secure my desired occupation, I instead was employed as a quality assurance tester for several major industry developers. Even at an entry-level position, these work experiences exposed me to a variety of tools used and processes employed by game developers while designing games. During this process, I realized I lacked a significant amount of knowledge regarding the field in which I was trying to enter. In
order to improve my chances of finding career-level employment, I began fleshing out my understanding of the design and implementation techniques and processes used when incorporating sound, voice and music assets into games. This knowledge was further solidified with real world experience when I landed my first sound design internship with Insomniac Games, and again when a contract sound design position with Activision Blizzard turned permanent.

Scope and Audience

The project consists of eight video tutorials and focuses on the following topics in this order as they pertain to audio design and implementation for games: Planning, Processes and Design, Introduction to Audio Middleware, Introduction to a Proprietary Audio Engine, Spatiality, Sound Effects and System Limitations, Parameters, Event Effects and 3D Properties, Kismet Visual Scripting, Non-linear Music Structures and Voice, Localization and Naming Conventions. All examples of music composition, voice work, sound design, level design and audio system design are original. This project targets the aspiring audio professional with an interest in the video game industry as they complete their studies at a collegiate or trade school level. As such, the primary tools I have used in the project for demonstration purposes are freely available for non-commercial and/or educational use, but are also heavily used in a professional capacity.

Purpose

The purpose of the project is to inform and demystify the target audience in regard to the project’s subject matter and present a jumping-off point from which a member of the audience can continue their own personal study. The project’s video
tutorials uniquely present and compare approaches to implementing audio into a video
game using a middleware engine or a propriety sound engine. The content of the project
is both informational and instructive.

Limitations

This project does have limitations. Any three-dimensional level design work
should be considered rudimentary, as I am not a level designer (this is a specialized
position within the video game industry). The levels designed however, sufficiently
convey any visual cues necessary to support the audio related work and the topic at hand.
Additionally, any discussions on scripting or programming remain largely conceptual as
an in-depth analysis of programming is beyond the scope of this project. As discussed in
the video tutorials, sound library usage is commonplace in the video game industry.
While most of the project’s sound design elements have been originally recorded, some
stock sounds have also been acquired. That said, these elements have been incorporated
and redesigned for use in the project in a unique and original fashion.
CHAPTER II

VIDEO TUTORIAL SUPPLEMENT

Explanation

As discussed in the introduction, the following section will summarize and expand upon the ideas presented in the video tutorials, and enhance the project with references to other noted works in the field and interviews with game industry professionals.

Video Game Industry Perspective

Each new generation of home console hardware brings possibilities and challenges for an audio professional working in the video game industry. Upon its release in 1988, the Sega Genesis (or Mega Drive) was capable of six-channel FM stereo synthesis, with one 8-bit pulse code modulation sample channel (Collins 2008, 39-40). In 2006, almost twenty years later, the Sony PlayStation 3 could handle up to 512 audio channels and was capable of streaming in 7.1 surround sound (Collins 2008, 71). The Blu-ray media format would eventually boast 50 gigabytes of data storage capacity.

In-step with this audio evolution, the video game industry has grown to be larger and more profitable than the film industry, and is expected to jump from “$67 billion in 2012 to $82 billion in 2017” in total industry revenue (Gaudiosi 2012). With large game publishers being publically traded on the stock market, developers are beholden to release
quality games worldwide on a number of platforms to drive profitability. For publishers who develop for the current home console generation, this often means comparable releases on the Sony PlayStation 3, Microsoft Xbox 360 and Nintendo Wii. The challenge, from an audio development perspective, comes with the disparities between these devices. The Xbox 360 and Nintendo Wii support the DVD format, which provides a fraction of the data storage space available to the PlayStation 3’s Blu-ray format. The PlayStation 3 and the Nintendo Wii have half (or less) the available system RAM that the Xbox 360 does. The Nintendo Wii and its successor, the Wii U, are only capable of streaming audio at less than CD quality: 32Khz (Ferris 2012)\(^1\).

**Console Development Challenges**

Media or disc space is a particular concern when it comes to a game’s voice assets. Activision Blizzard’s October 2012 retail release “Skylanders: Giants” shipped with eleven localized languages, consisting of over 70,000 individual voice files. Often, several languages must fit on one region-based stock-keeping unit (SKU) while competing for space with the remainder of a project’s assets (sound effects, music, art, pre-rendered movie files and code). Releasing a dual-layered DVD or more than one SKU in a geographical region (Europe, North America, etc.) can increase manufacturing costs for a project, and therefore may not be an option. Finding an acceptable fidelity level for audio assets when storage space is in short supply is often a decision reached by consensus, with input from several development and production team members.

---

\(^1\) Lorien Ferris, Interview by author with Toys for Bob Audio Manager, Novato, California, November 1, 2012.
While some hardware configurations designate RAM for dedicated functionality (i.e. video), system RAM is usually the most common resource available in modern home consoles. Like disc space, however, system RAM is in constant demand, often meaning a smaller overall audio allotment. As such, short, time-sensitive sound effects are often assigned to RAM, while longer voice or music assets are candidates for streaming. While audio loaded into RAM is available for immediate playback, streaming assets may trigger with a slight delay in game as the correct content is searched for and read from a disc or hard drive. In addition, while some consoles have several streams (or voices) available for use, audio functionality may have to be paired down to work with the lowest common hardware denominator in a multiplatform release. Audio middleware solutions like FMOD Designer 2010 go a long way toward addressing the problem, allowing a designer to designate platform-specific project layers. However, there may not be enough time during a development cycle to tailor additional assets for each platform. Beyond all of this, an audio department must also account for when a game starves a console for resources, creating a system of fail behaviors if too many RAM or stream assets are triggering at the same time.

While a generational leap in console hardware often means forward progress for audio capability, this is not always the case. Nintendo is the first major hardware manufacturer to jump into the next generation of home console releases with the Wii U this November 2012 in North America (Benedetti 2012). Both the Nintendo Wii and the Wii U only support up to a 32Khz audio sample rate, meaning that the audio output capability on Nintendo consoles will remain below CD quality for two console cycles. According to the Nyquist sampling theorem, an audio file sampled at 32Khz can only
reproduce a frequency response of up to 16khz (Huber and Runstein 2005, 219). This is a serious limitation, and represents a level of quality that is “inadequate to deliver the full frequency range of human hearing without noticeable sonic artifacts” (Ferris 2012).

File Format and Fidelity

Even in the face of hardware limitations, it is prudent to select an audio file format and level of fidelity that can be capitalized on by the higher-end consoles in a multiplatform release. Audio assets sound better when down-sampled from a higher quality format (Brandon 185), as scaling up introduces noise into a signal through dithering. The pulse code modulation wave file format is a smart choice for use in almost any game development project. It is a standard file type supported by commercial and proprietary audio software worldwide. If not wave, another uncompressed format should be used when cataloging a game's source sound effect, voice and music assets. While these assets may undergo some form of compression or sample rate reduction tool-side, the source files will remain uncompromised. As the average human being can perceive around 20khz, a 44.1khz sample rate (CD quality audio) should be acceptable for most projects (or 48khz for the audiophiles). Higher bit depths, while harder to audibly perceive, may reduce the noise floor in a digitized signal and more closely approximate an analog waveform. 16 and 24-bit are common and acceptable depths.

Asset Procurement

Once a format is selected and a fidelity determined, field and Foley recording sessions represent two ways to secure original audio content for a game, however many effects that could be captured in the field or in the studio are likely to be well represented
in commercially available sound libraries. Every project I've worked on has used a sound library in some fashion. Particularly under tight game development deadlines, having tens of thousands of sounds at a designer's fingertips can be extremely helpful. It may make more financial sense for a project's bottom-line to utilize a sound library instead of scheduling what could be a costly recording session (i.e. renting a gun range to record firearms). A potential downside to library usage could be file format discrepancies as many older libraries conform to CD quality audio and may not match a project's specifications. Additionally, as these libraries are publicly available to other sound and media teams, the samples contained within may not be inherently unique. For instance, browsing an internal database of several licensed sound libraries while working on Skylanders: Giants, I came across a cave ambience that was used in Konami’s 1997 PlayStation release Castlevania: Symphony of the Night. Moreover, the ambience was also baked into Michiru Yamane’s track “Path of the Departed” on the game’s official CD soundtrack. As such, using a stock library sample is ill advised. It is essential that a sound designer identify ways to combine and/or processes to alter library elements to make an asset “new and fresh” (Aarons 2001, 227).

Editing Practices

While gathering and preparing audio content for a project, employing a few editing practices can greatly decrease overall file size and increase asset usability. Many stereo sound effect files can be summed to mono, along with a majority of a game's voice assets. Background sound elements can undergo a sample rate reduction and still properly convey sonic information to the player as needed. With sound effects, music, voice and
ambience all competing for a player’s attention, audio assets with excessive fades or lengthy reverb tails can be heavily cropped to reduce file size.

In order to be usable, field recordings may require noise reduction to negate unwanted ambient or background sound elements, though aggressive processing can also strip away content that should be preserved. Some experimentation may be required to find an acceptable level of noise reduction that does not compromise the integrity of an asset. While not ideal, sound files that do not match a project's format specifications should not be overlooked for inclusion. Introducing a small amount of noise to a waveform during the resampling process through dither may be acceptable and practically inaudible.

Many software audio editors are capable of sample accurate editing, allowing detailed waveform and phase analysis. While visually inspecting audio assets for anomalies is great way to identify potential issues, auditory perception should always be employed to confirm any suspicions. For instance, a break in a waveform, excessive compression, hard limiting or even clipping can be acceptable in the right circumstances. If no audible clicking or popping occurs on playback, a broken waveform may be satisfactory. An asset that contains excessive compression or limiting may work fine in a project, given that it is not competing with other audio elements containing similar frequency content in game. Likewise, an audio file that clips slightly may convey a desirable sense of intensity with acceptable amounts of distortion.
Batch Processes

Where possible, redundant editing processes should be identified and automated. On the technical side, channel conversion, bit depth conversion, resampling, trimming, normalization and equalization (or filtering) are all processes that can be handled in an audio editor and strung together into a sequential batch process. These tasks, in general, do not require a designer's touch and are therefore good candidates for automation. Once acceptable settings are determined for each step of the process (and save options defined), large quantities of files can be quickly processed and brought in line with a project's specifications. Batch processes can also be helpful when applying effects to voice assets. For instance, if we know a particular character's voice is robotic in natural, we can create a multi-step process to be used across all that character's voice files (in all localized languages). Batching becomes particularly useful as localized voice assets are delivered toward the end of a project’s development cycle.

Sound Design and Implementation

Fundamentally, almost all sound sources are generated from oscillation and resonance and sound design can certainly be approached with these properties in mind (Farnell 2010, 39). Programming languages like Pure Data and design environments like Native Instruments Reaktor series let knowledgeable users construct and process audio signals from what are essentially the buildings blocks of sound (Farnell 2010, 149). In reality, these are not often effective approaches to sound design for video games. Tight development cycles, often less than a year in length, prohibit such a detail-oriented and time-consuming approach to design. More frequently, sound designers take a layered
approach, combining elements from several audio sources or samples. This process should not however, be considered rudimentary. Designers must still have a solid understanding of the content they choose to incorporate while designing in this manner. Consideration should be given to how each sound element contributes to the overall design. When adding layers, designers must also be aware of frequency build up, particularly when creating loud sounds (i.e. explosions, impacts). These sounds are quick to clip, and knowing when additional elements are no longer enhancing, but taking away from an asset (through masking or frequency build up) is essential. Should a project lean more toward realism, there is still design work to be done by sweetening a sound (adding supplemental audio elements to augment a desirable set of frequencies).

From a development perspective, the idea of implementing audio into a game in a strategic and impactful fashion is of great importance. While solid sound design is the fundamental backbone of an enjoyable aural experience, the way in which sound, music and voice assets come together during a gameplay experience will likely be more noticeable or perceptible to an end user (or not, depending on the intention of the designer). Audio implementation is inherently design, an idea explored in significant depth in the non-print portion of this project.

Audio Middleware

Audio middleware programs, whether Creative Labs’ GameCODA, RAD’s Miles Sound System, Firelight Technologies FMOD Designer or Audiokinetic’s Wwise, serve one purpose: to allow audio professionals to link sounds to in-game objects, events and atmospheres without the need of advanced programming skills (Brandon 2007).
FMOD Designer, a popular middleware solution, offers a graphic user interface (or GUI) that will be familiar to many audio professionals that have experience with multi-track audio software suites. Additionally, middleware offers extensive behavioral control over audio assets in a game environment as well as virtual voice functionality. Beyond this (per Firelight Technologies product description), virtual voice support allows a game to play several thousand sounds at once on limited hardware platforms without the need for programmer-designed logic to toggle sounds on and off. When properly configured, this functionality can alleviate starving issues on low-end technology. Ultimately, audio middleware providers have support teams that are “dedicated to improve the software on a daily basis [to ensure] that the product stays on the cutting edge of game audio technologies and gives its end users the features they need in today’s market” (McMenamy 2012). This is a huge benefit, as having a reliable external tech team can significantly reduce an audio department’s demand on its internal team of programmers.

**Scripting**

Kismet Visual Scripting represents another way designers can alleviate an audio department’s reliance on a project’s programmers. Kismet can be defined as an object-based scripting environment in which gameplay events, actions, variables and conditions can be linked together to create or prototype gameplay and audio systems within the Unreal Development Kit (Unreal Developer Network 2012). Kismet also partially exposes the functionality of UnrealScript; Epic Games’ internally developed scripting language. In reality many of the actor classes, including audio emitters, present

---

2 Jamie McMenamy, E-mail interview by author Insomniac Games Senior Sound Designer, November 1, 2012.
in the Unreal Development Kit all have associated scripts. As such, working with Kismet has the added benefit of exposing a sound designer to traditional scripting concepts like macros. “A macro is essentially a shortcut to a much more complex function” (Laabs 2012)³. Kismet’s action objects are largely equivalent to scripting macros. Additionally, variables are constantly used in script and are also exposed for use in Kismet. Having even a fundamental understanding of scripting is a skill highly sought-after, and greatly increases an aspiring audio professional’s chances of finding work in the game industry.

Voice

Before a dialog script has been finalized for a game project, temporary voice assets are often required to prototype events or cutscenes relevant to the game's story, characters or progression. As it makes no sense to schedule sessions with paid voice actors until a script is at or near completion, scratch voice recordings are often used as placeholder and can be provided by any member of the development team that feels comfortable in front of a microphone. The downside of this process is the removal of a developer from their primary duties. When dealing with a particularly large amount of placeholder dialog lines, a text to speech software package (preferably one with command line access) can also work as a temporary solution. Command line access allows a programmer to write scripts that control a program’s functionality without using a comparatively sluggish GUI. For synthetic speech, this often means repeatedly pulling information from a raw text or comma-separated value document (i.e. filename, dialog

³ Dan Laabs, Interview by author with Toys for Bob Sound Scripter, Novato, California, October 12, 2012.
content) to feed the program and quickly generate assets that match a project’s specifications. Even early in development, this process can give an audio team a better estimate of overall voice asset file size. With a finalized script in hand, union affiliated voice actors can be brought in for traditional studio recording sessions. Development projects with large budgets may also record voice acting on location during motion-capture sessions in the hope of getting a more honest performance from a production's actors. However, depending on the style of a game and the nature of its characters, this may not be an option.

Music

In-house composers are rare in the video game industry. More and more frequently, studios turn to contracted talent to satisfy a project's music needs. Competition for composition contracts can be extremely stiff, with noted Hollywood composers like Hans Zimmer actively involved in soundtrack work on flagship game productions (Snider 2009). With an influx of traditional film composition talent, orchestral soundtracks on AAA game titles are not uncommon, though a studio’s internal audio team will rarely handle associated recording duties. No matter the composer or the budget, when working with an external composer it is important to convey project deadlines and expectations, as well as format requirements for music deliveries. As it is still the audio department’s responsibility to implement a game’s music assets, requests for loops and clean, automation free instrument stems (surround or otherwise) should be anticipated (Angell 2009, 221). These isolated assets are integral for crafting a dynamic musical atmosphere as well as creating an overall sense of immersion within a game.
CHAPTER III

TREATMENT

Narrative Video Tutorial Assembly

Video games are inherently multimedia, combining sound, visuals and text to create products whose sums are greater than their parts. Creating a narrative set of audio-visual tutorials to document the practices associate with sound design for video games was a logical step. In progressing to the completion of this project, several game engine and audio middleware solutions were evaluated prior to the selection of the Unreal Development Kit and FMOD Designer 2010. With the bulk of the project’s software chosen, several sub-topics within the scope of the project were identified, then honed down to help create a sense of focus. This process determined the content of each video tutorial. At this stage, sound design elements were gathered or recorded, voice work was performed, and music composed to demonstrate the concepts to be addressed in each tutorial. Game levels were also designed in the selected engine, and middleware projects created. As sound, voice and music assets were finalized; they were incorporated into the appropriate game or middleware environment. A narrative outline was drawn up for each tutorial, with narrative voice work captured separately from the tutorial footage. These elements were then combined in a video editor and rendered as standalone movie files. Each video tutorial was rendered in high definition and all audio assets designed at CD
quality. Additionally, sheet music was created for each of the original music compositions contained in the project.

Equipment List

Due to the nature of this project, there is no traditional production script containing information on set requirements, props, etc. However, making the project’s video tutorials required the following: a personal computer to run all project related software (Mac Mini with Boot Camp and Windows 7 installed), microphones for recording purposes (Rode NT2-A for voice work, Sennheiser ME66 and MKH 30 for mid-side field recording), TASCAM HD-P2 portable field recorder (to digitally capture recordings), Mackie MR5 reference monitors (to mix video tutorial audio levels), Epic Games’ Unreal Development Kit (to design levels and implement audio in a proprietary sound engine), Firelight Technologies’ FMOD Designer 2010 (to implement audio in a middleware environment), Sony Creative Media’s Sound Forge 10 (to demonstrate batch processing and waveform analysis), Audacity Team’s Audacity 2.0.1 (to illustrate noise reduction), and Techsmith’s Camtasia Studio 8 (to record, edit and render video footage).
CHAPTER IV

SUMMARY, CONCLUSIONS AND RECOMMENDATIONS

Summary

In summary, this project has been created to encompass core concepts and processes associated with the design and implementation of audio assets for modern video games. It has also put a spotlight on a number of freely accessible audio software packages and game development toolkits available to college or trade school students with an interest in audio development in the video game industry. This project has additionally provided its target audience with enough tangible demonstrations, overall product knowledge and design and implementation philosophies to either allow a continued study using the concepts and tools demonstrated within the project, or to pursue other avenues equally relevant to the field.

Conclusions

Upon reflection, the scope of this project turned out to be substantially larger than first surmised. The project was initially proposed to contain video tutorial footage totally approximately an hour. Upon completion, the project consists of over an hour and forty minutes of audio-visual content. Focusing on just one aspect of the proposed project, whether sound design, music or voice work, would have proved an effective
vehicle to deliver the project’s intentions to its target audience. That said it is not uncommon for audio professionals in the game industry to focus specifically on one of these areas of expertise. As an introduction (and more) to these areas, and as a statement to the importance of audio asset implementation as its own form of design, I believe the project succeeds.

Recommendations

The video game industry as a whole has matured and developed tremendously over the last twenty years, and will continue to do so off into the future. As of the completion of this project, Firelight Technologies has announced a successor to FMOD Designer 2010 currently in development. The Nintendo Wii U is taking the first into the next generation of home video game consoles, and Sony and Microsoft are sure to follow. It is recommended that anyone with an interest in the video game industry from an audio standpoint continue their own studies, and stay abreast of new game audio development technologies and evolving platform capabilities.
REFERENCES
REFERENCES


APPENDIX A
COPYRIGHT CITATIONS

The following citations have been included at the request of the software makers featured in this project. Additional legal clearances have been provided to the California State University, Chico Office of Graduate Studies at the time of the project’s submission.

“Audacity(R) software is copyright © 1999-2012 Audacity Team. [Web site: http://audacity.sourceforge.net/. It is free software distributed under the terms of the GNU General Public License.] The name Audacity(R) is a registered trademark of Dominic Mazzoni.”


Sound Forge Pro 10 © Sony Creative Software.

Unreal Development Kit © 2008-2012, Epic Games, Inc.
APPENDIX C
STATEMENTS OF LEGAL CLEARANCE

Statement of Legal Clearance:

Matthew David Copeland’s educational usage of FMOD Sound System video footage for The Design and Implementation of Sound, Voice and Music in Video Games, a non-print media project for the Master of Arts in Music at California State University, Chico, requires no additional licensing or permissions on behalf of Firelight Technologies. It is understood that the project, once approved by the Office of Graduate Studies, will be uploaded to California State University, Chico Digital Repository; an archive for scholarly work accessible via the World Wide Web.

Brett Paterson
Name (Print)

Name (Signature)

Title

Firelight Technologies
Company

11/11/12
Date

Matthew David Copeland
Name (Print)

Name (Signature)

Firelight Technologies
Company

11/13/12
Date (Draft)
Statement of Legal Clearance:

Matthew David Copeland’s usage of the Unreal Development Kit is covered by Epic Games’ UDK End-user license agreement, which states the following:

“The UDK is FREE for educational use. Feel free to use it in your school’s curriculum. Educational institutions can use the UDK for free even though they’re charging tuition. No additional license is required for faculty or students. Your students are encouraged to use UDK inside or outside of school to learn game development.

The UDK is also FREE for non-commercial use. Feel free to use the UDK to make any application (game or otherwise) for free distribution. No additional license (beyond the EULA) is required. Just go for it.”

[Signatures]

Name (Print)          Name (Print)

Name (Signature)      Name (Signature)

Title

Company

Date

Date (Draft)
Statement of Legal Clearance:

Matthew David Copeland's educational usage of Sound Forge Pro 10 video footage for *The Design and Implementation of Sound, Voice and Music in Video Games*, a non-print media project for the Master of Arts in Music at California State University, Chico, requires no approval or permission on behalf of Sony Creative Software. It is understood that the project, once approved by the Office of Graduate Studies, will be uploaded to California State University, Chico Digital Repository, an archive for scholarly work accessible via the World Wide Web.

STEVE FOLDVAR
Name (Print)

Steve J. T.
Name (Signature)

Promotions Manager
Title

SOEY CREATIVE SOFTWARE
Company

November 13, 2012
Date

Matthew David Copeland
Name (Print)

Matthew David Copeland
Name (Signature)

Date (Draft)
Statement of Legal Clearance:

Matthew David Copeland's educational usage of Audacity 2.0.1 video footage for The Design and Implementation of Sound, Voice and Music in Video Games, a non-print media project for the Master of Arts in Music at California State University, Chico, is acceptable to the Audacity Team. Specific instructions have been provided for credit/citation purposes and will be adhered to. It is understood that the project, once approved by the Office of Graduate Studies, will be uploaded to California State University, Chico Digital Repository; an archive for scholarly work accessible via the World Wide Web.

Gale Andrews
Name (Print)
Name (Signature)

Matthew David Copeland
Name (Print)
Name (Signature)

Quality Assurance
Title

Audacity Team
Company

13 November 2012
Date
11/18/12
Date (Draft)

Citation Requirements
> We would request if at all possible that Audacity receives the following citation respecting our copyright and trademark in the scrolling credits at the end of the video (or elsewhere in it):
> "Audacity(R) software is copyright (c) 1999-2012 Audacity Team. [Web site: http://audacity.sourceforge.net/. It is free software distributed under the terms of the GNU General Public License.] The name Audacity(R) is a registered trademark of Dominic Mazzoni."
Statement of Legal Clearance:

Matthew David Copeland (interviewer) has permission to use any statements gathered from Lorien Ferris (interviewee) during an interview conducted on November 1st, 2012 in The Design and Implementation of Sound, Voice and Music in Video Games, a non-print media project for the Master of Arts in Music at California State University, Chico. It is understood that the project, once approved by the Office of Graduate Studies, will be uploaded to California State University, Chico Digital Repository; an archive for scholarly work accessible via the World Wide Web.

Lorien Ferris
Interviewee Name (Print)

[Signature]
Interviewee Name (Signature)

Senior Sound Designer
Title

Toys For Bob
Company

11/13/2012
Date

Matthew David Copeland
Interviewer Name (Print)

[Signature]
Interviewer Name (Signature)

11/13/12
Date (Draft)
Statement of Legal Clearance:

Matthew David Copeland (interviewer) has permission to use any statements gathered from Dan Laabs (interviewee) during an interview conducted on October 10th, 2012 in *The Design and Implementation of Sound, Voice and Music in Video Games*, a non-print media project for the Master of Arts in Music at California State University, Chico. It is understood that the project, once approved by the Office of Graduate Studies, will be uploaded to California State University, Chico Digital Repository; an archive for scholarly work accessible via the World Wide Web.

Dan Laabs  
Interviewee Name (Print)

[Signature]  
Interviewee Name (Signature)

Sound Scripter  
Title

*Top for Bob*  
Company

11/13/2012  
Date

Mathew David Copeland  
Interviewer Name (Print)

[Signature]  
Interviewer Name (Signature)

11/13/12  
Date (Draft)
Statement of Legal Clearance.

Matthew David Copeland (interviewer) has permission to use any statements gathered from Jamie McMenamy (interviewee) during an interview conducted via e-mail on November 1st, 2012 in The Design and Implementation of Sound, Voice and Music in Video Games, a non-print media project for the Master of Arts in Music at California State University, Chico. It is understood that the project, once approved by the Office of Graduate Studies, will be uploaded to California State University, Chico Digital Repository, an archive for scholarly work accessible via the World Wide Web.

Jamie McMenamy
Interviewee Name (Print)

Interviewee Name (Signature)

Senior Sound Designer
Title

Insomniac Games
Company

11/13/12
Date

Matthew Copeland
Interviewer Name (Print)

Interviewer Name (Signature)

11/13/12
Date (Draft)