

高度な専門的知識を有する職業人育成のための サウンドプログラミング教育の実践

研究期間 平成 29 年度

研究代表者名 藤沢望

I. はじめに

プログラミングによって音の生成やサウンドエフェクトなどの様々な音響処理を行う「サウンドプログラミング」は、音響理論を理解し、OS やデバイス環境ごとに異なったサウンドアーキテクチャに合わせたプログラミング技法を必要とするため、非常に高度で専門性が高い。本研究の目的は、楽器・音響機器・音響関連ソフトウェアメーカー等の開発現場で活躍できる高度な専門的知識を有する職業人を育成するために、効果的なサウンドプログラミング教育を行うことである。その実践の場として、情報メディア学科学生への課外講座「サウンドプログラミング勉強会」を行い、音響理論やデジタル信号処理の知識修得と同時に実践的なサウンドプログラミング技法を指導する。

本年度は同課題の初めての取組みということで、①対象 OS およびプログラミング環境の検討、②課外講座の実践、③課題の発見、を行うことを目的とした。

II. 研究内容

II-1. 対象 OS およびプログラミング環境の検討

サウンドプログラミングを行うにあたってはまず、「どの OS で実行するのか」「どのようなプログラミング環境を使用するのか」ということを検討する必要がある。ここではまず、主要なデスクトップ OS である Windows および Mac のサウンドプログラミング環境について概説する。

(1). Windows のサウンドプログラミング環境

Windows はバージョン 3.0 の時代から、MME (Multi-Media Extension) という拡張機能を提供しており、WinMM というオーディオ API が利用できるようになっていた。その後、ゲーム・マルチメディア API の DirectX の一部として DirectSound、XACT、Xbox との互換性を持つ XAudio2、Windows Vista では新しいサウンドコンポーネントである Core Audio が導入され、現在では WASAPI[1]というオーディオ API が標準となっている。これらは Microsoft 社が提供している OS 標準のサウンド環境であるが、サードパーティ製のサウンド環境として Windows, Mac のクロスプラットフォームである Steinberg 社の ASIO 規格[2]も提供されている。

Windows のプログラミング環境としては、Microsoft 社提供の Visual Studio を用いることで、C・C++・C#・Visual Basic などの開発言語や、コマンドラインで動作する CUI アプリ、Windows 環境で動く GUI アプリ、Windows タブレット用アプリなど、様々なプ

プログラミング言語・実行環境の組み合わせを選択できる。

(2). Mac/iOS のサウンドプログラミング環境

Apple 社の販売する Mac では、旧来の Mac OS 9 までで使用されていた Sound Manager に代わり、Mac OS X から Core Audio[3]（Windows と同名だがこちらのほうが導入は早い）というオーディオアーキテクチャが導入されており、Audio Toolbox や Audio Unit というオーディオ API が利用できる。さらにクロスプラットフォームのオーディオ API である OpenAL も利用できる。また Apple 社のモバイルデバイス用 OS である iOS は、Mac OS X をモバイル用に改良したもので、Core Audio もいくつかの違いを除いて利用できる。

プログラミング講座の実践にあたって、近年の学生が特に興味を持っている iOS や Android デバイス用のモバイルアプリ開発も検討したが、プログラミング言語修得の困難さ（iOS の Objective-C や Swift）や、動作デバイスの準備、プログラミング実行やデバッグ環境を構築する手間などを考慮すると、サウンドプログラミング以外の部分での説明や指導に大きく時間を取られることになるため、今回は断念した。Mac も iOS と同様の問題があるために除外し、今回のプログラミング講座では、Windows および Visual Studio を用いて Windows アプリとして動作するサウンドプログラムを作成していくことにした。

II-2. 課外講座「サウンドプログラミング勉強会」の実践

(1). 参加者

今回の講座では、上位学年で音響学の知識を有する少数名を対象とすることとし、情報メディア学科 3 年生および 4 年生のゼミ学生から参加者を募り、4 名が参加した。

(2). 開催日程・内容

平成 29 年 10～12 月にわたって、全 9 回の講座を行った。各回の内容を表 1 に示す。

表 1 サウンドプログラミング講座の内容

第 1 回	開発環境のインストール、バイナリエディタの使い方、C++コンソールアプリの作成
第 2 回	バイナリファイルの解析、WAV ファイルの読み書き
第 3 回	サウンドエフェクト「ディレイ」の作成
第 4 回	ディレイ機能の拡張
第 5 回	エフェクト機能の関数化、ディストーション、リミッター、コンプレッサーの作成
第 6 回	Windows デスクトップアプリの作成、WAV ファイルのオープン
第 7 回	WAV ファイルの解析、WAV ファイルの書き出し
第 8 回	GUI の構築、エフェクト（ディストーション）機能の追加
第 9 回	サウンド再生機能の追加

講座の第 1 回～第 5 回までは、C++言語によるコンソールアプリとして、WAV ファイルの読み書きやサウンドエフェクト機能のプログラミングを行った。これらのプログラミングにあたっては、青木[4]の書籍およびサンプルプログラムを参考に、Visual Studio 環境で動作するように改良を加えている。サウンドプログラミングでは、音声の時系列データがどのように扱われるかを理解しておく必要があるため、バイナリエディタを導入し、WAV ファイルのバイナリデータ解析も行っている。

第 6 回～第 9 回までは、受講学生の要望により、Windows デスクトップで動作するアプリケーション開発を行った。C#言語および.NET Framework による、標準的な Windows フォームアプリケーションの作成方法を用いている。GUI 構築には Visual Studio による標準コントロールを用いている。

各回の講座実施にあたっては、PowerPoint による教材を作成し、参考テキストのコピー、ソースコードなども配布した。PowerPoint 教材の一例を図 1 に示す。

SoundEffectsAppへの再生機能の追加

- Form1.cs[デザイン]にコントロールを追加する

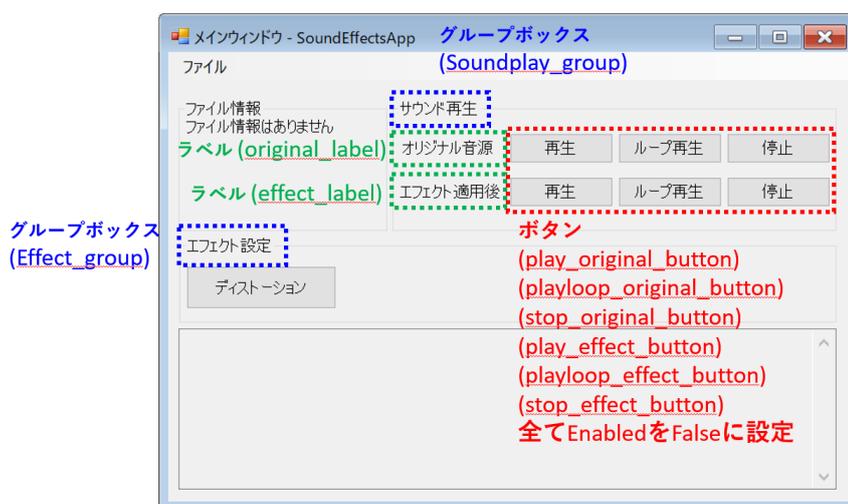


図 1 PowerPoint 教材の一例

II-3. 講座実施によって発見された課題

(1). 受講者のプログラミング技能の不足

今回の講座では、音響学の知識を有する上位学年の学生として、情報メディア学科 3 年生および 4 年生のゼミ学生に参加してもらったが、プログラミング技能がそれほど高いとは言えない学生が多かった。これまでに受講した授業で C 言語や JAVA 言語によるプログラミングの経験はあるものの、C#言語や Windows アプリケーションの作成は全員が初め

てということであり、そのうち 1 名はプログラミングの授業もあまり履修していないということであった。

そのため、ソースコード入力におけるタイプミス、プログラム構文の間違いや命令文の順序・挿入位置の間違いなど、初歩的な部分でのつまずきが多く見られ、思うように講座内容を進めることが難しかった。

(2). GUI アプリケーション作成の経験不足

今回の講座では、C#言語・.NET Framework による Windows アプリケーションの作成を行ったが、受講者全員が初めての経験であり、これも講座内容が進まない要因の一つとなった。しかし受講者からは、実際に動作するアプリケーションの作成が経験できて良かった、との感想もあった。

(3). アプリケーションへのサウンド機能導入の困難さ

今回は .NET Framework で利用できる最も簡単なサウンド再生機能として、SoundPlayer クラスを用いた。これは WAV ファイルを読み込み、再生・停止するだけのクラスであり、音量やパンニングのコントロールは出来ない。Windows アプリケーションで高度なサウンド機能を導入するには、前述の WinMM や DirectSound、WASAPI などのオーディオ API を利用する必要があるが、いずれもプログラミング初心者には難易度が高く、多くの予備知識やプログラムへのコード追加が必要となる。

(4). 今後の展望

今回の講座実施によって上記のような課題が明らかになったが、その根本的な原因は、受講者のプログラミング技能とサウンドプログラミングの困難さのミスマッチにあったのではないかと考えられる。プログラミングに不慣れな初級者に、高度なサウンド機能をプログラミングさせるのは困難なことであるが、これを解決するためには、講座以外の部分で受講者のプログラミング技能を向上させるような工夫が必要だと思われる。そのためには、正課カリキュラムとの連携なども必要となってくるだろう。

一方、C++コンソールアプリでのサウンドエフェクト作成では、一度、アプリケーションの枠組みを作ってしまうと、あとはエフェクト機能を個別に実装していく部分に集中できたので、受講者の理解も速く、プログラムの改変や機能追加にも取り組むことが出来た。このようなサウンド処理のコア部分のプログラミングは、アーキテクチャやプログラミング言語に依存しないものなので、サウンドプログラミング教育の内容として多いに有効であると言える。

III. 研究成果

今回の講座を実施して、サウンドエフェクト機能の作成など、アーキテクチャやプログラミング言語に依存せず、初級者でも比較的容易にプログラミングが行える部分があることが明らかになった。この知見は、情報システムで行っている高校生向けの公開講座や、情報システム学科の担当授業でも取り入れて実践していく予定である。

IV. おわりに

本研究では、高度な専門的知識を有する職業人を育成するために効果的なサウンドプログラミング教育を行うことを目的とし、課外講座「サウンドプログラミング勉強会」を実施した。講座を通じて明らかになった課題は、受講者のプログラミング技能とサウンドプログラミングの困難さのミスマッチであった。これを解決するには、正課カリキュラムとの連携などによって、受講者のプログラミング技能の向上を進めることが重要であろう。そのうえでサウンドプログラミングの教育内容を充実させ、正課カリキュラムへの導入を検討していきたい。

参考文献

- [1] Windows Dev Center, “About WASAPI”.
<https://msdn.microsoft.com/en-us/library/windows/desktop/dd371455.aspx>
- [2] Steinberg, Developers サイト.
<https://www.steinberg.net/en/company/developers.html>
- [3] Apple Developer, “Core Audio の概要”.
<https://developer.apple.com/jp/documentation/CoreAudioOverview.pdf>
- [4] 青木直史, “C 言語ではじめる音のプログラミング—サウンドエフェクトの信号処理,” オーム社 (2008).