

和歌山大学協働教育センター クリエプロジェクト  
＜2020年度ミッション成果報告書＞

プロジェクト名：Sound as a System

ミッション名：直感的な GUI のクロスプラットフォーム立体音響演出システムの開発

ミッションメンバー：システム工学部 3 年中野裕介，システム工学研究科 1 年近藤伊佐直，システム工学部 3 年大森伊月，システム工学部 3 年北林悠河，システム工学部 3 年山本創大

キーワード：立体音響，クロスプラットフォーム，GUI，JUICE Framework，VST プラグイン

## 1 背景と目的

本ミッションは、専用の音源や高価な設備を必要としない立体音響演出を実現する目的で、昨年度のミッション「位相制御を用いた立体音響演出システムの開発」を発展させて遂行したミッションである。

昨今の映画やゲームといったデジタルコンテンツにおいて、立体音響演出は広く活用されている。これらの立体音響演出は、映画館等の施設に設置してある多数のスピーカーから、あらかじめ緻密に計算された音源を再生することで実現している。そのため、音源の制作から再生機器に至るまで大掛かりな設備や特殊な機材を必要とすることが多く、低コストでの実現が困難であった。

そこで昨年度、より多くの方に様々な場面で立体音響演出を楽しんでもらうため、低コストで体験できる立体音響演出システムを開発した。具体的には、位相制御技術を用いることで使用機器を必要最小限にし、かつ一般的な音源形式に対応させることでコストの削減に成功した。

一方で、課題点も新たに見つかった。それは、正位相・逆位相を制御するユーザインタフェース (GUI) にスライダーを採用しており、専門知識を持たないユーザにとって思い通りの演出を行うことが困難であった点である。

そこで本ミッションでは、様々な場面で運用できるシステムに改良することと、立体音響演出を思い通りに行えるようにすることの 2 つを目的に掲げた。具体的には、昨年度のシステムを基に、クロスプラットフォームアプリケーションとして、専門知識を必要としない直感的な GUI を持つシステムを完成させることを目標として設定した。

## 2 活動内容

本ミッションでは、GUI を刷新したクロスプラットフォームアプリケーションを開発するにあたり、専門知識を必要としない直感的な GUI の実現を最も優先すべきと考えたため、まず GUI のデザインを行ったのち、それに合わせて開発環境を選定し、その後に開発と評価を行った。以下では、これらについて詳細に述べる。

### 2.1 GUI のデザイン

昨年度開発したシステムの GUI を図 2 に、本ミッションで開発した GUI を図 1 に示した。

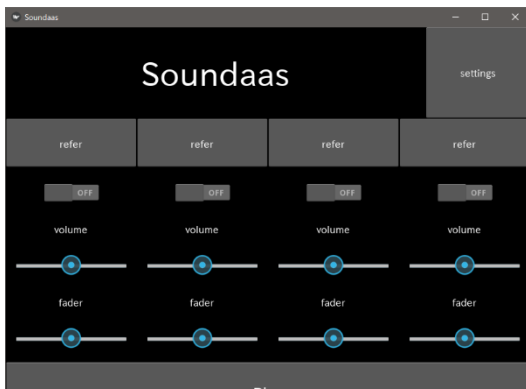


図 2：従来システムの GUI

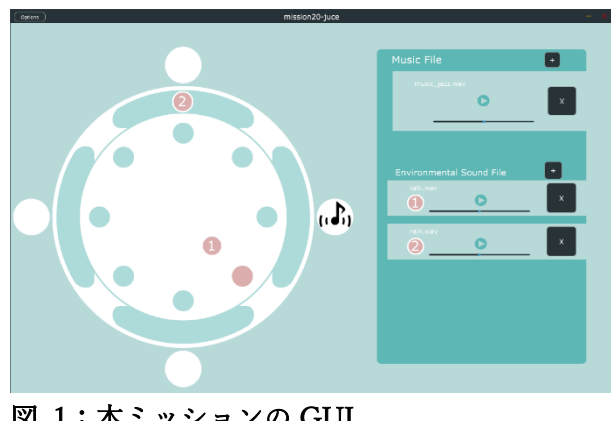


図 1：本ミッションの GUI

昨年度開発したシステムのユーザインタフェースは、四隅に配置したスピーカーからなる正方形の各辺をスピーカー対として扱い、各スピーカー対に対して正位相・逆位相を制御するスライダーを対応させて操作する方式を採用していた。これでは実際の空間と操作画面の対応付けができず、実際にどのように空間に音を配置しているかがイメージしにくい。そこで、実際の空間を模した仮想的な空間「Sound Space」を画面上に表示し、その空間内に音源オブジェクトを配置するインタフェースとしてデザインした。

## 2.2 開発環境の選定

本ミッションのシステムは、JUICE を用いて C++ で開発することとした。JUICE は、ネイティブアプリケーションや VST プラグインのビルドが可能なクロスプラットフォームアプリケーション開発フレームワークであり、音声信号処理や GUI 開発が行いやすいという特徴を持つ。また、音楽制作ソフトウェア上で動作するプラグインを開発する標準的なフレームワークとしても知られている。

昨年度は、位相制御によって立体音響演出を行うことは一般的な手法ではなく、初の試みであったことを考慮し、企画から実装までの試行錯誤を行いやすいよう、軽量言語である Python を用いて開発を遂行した。これに対し、今年度は昨年度の開発実績があるため、開発コストよりも動作速度や安定性、対応しているプラットフォームを重視し、開発言語・フレームワークを選定した。

## 2.3 開発と評価

開発は、前節で選定した JUICE を用いて行った。チーム開発の形態も昨年度からさらに効率化を図った。具体的には、処理を行うクラスやメソッドの定義を明確にし、メソッドのシグネチャを事前に共有したうえで、GitHub を用いて各担当者が並行して開発を行った。また、デザイナーとエンジニアとのコミュニケーションを円滑に行うため、GUI の設計には Adobe Xd を使用した。これにより素材の受け渡しや配置についての共有を正確かつ迅速に行うことができた。

なお、開発したシステムの評価は、プロジェクトメンバー複数人で聴取によって行った。聴取は、スピーカー4基とオーディオインタフェースをコンピュータに接続して行うもので、昨年度と同様の設備で行うことができた。結果については、3.2 節で詳細に述べる。

# 3 活動の成果や学んだこと

## 3.1 開発システム

開発したソフトウェアは、主に GUI 部と内部処理部に大別される。

GUI 部では右側のボタンを通じて BGM の選択や環境音の動的な追加・削除、スライダーを使った基準音量の設定ができる。左側の GUI (Sound Space) では、ドラッグアンドドロップによって BGM や環境音を 2 次元的に空間に配置することができる。

内部処理部では、GUI の入力を受け取り、再生・停止・位相・音量・出力位置のそれぞれを制御している。処理において工夫した点として、GUI によって入力されたパラメータから各種制御を行うパラメータへの変換アルゴリズムが挙げられる。Sound Space 上にある音源オブジェクトの座標 (中心点からの距離と偏角) を算出したのち、偏角に対応する方向に再生位置を設定し、距離によって音量を変化させている。なお、距離が一定以上離れた場合 (GUI の外周部分にオブジェクトが配置された場合) には、位相を反転させることで音像をぼかし、周囲から音が聞こえるようにする効果を演出している。また、GUI の変更はすべてリアルタイムに処理され、出力に反映される。

さらに、スタンドアロンのアプリケーションとしてだけでなく、VST3 プラグインとしてもビルドし、VST ホストアプリケーション上で動作することを確認できた (図 3)。これによってホストアプリケーションやほかのプラグインと連携し、さらに自由度の高い音響演出を行うことが可能になった。

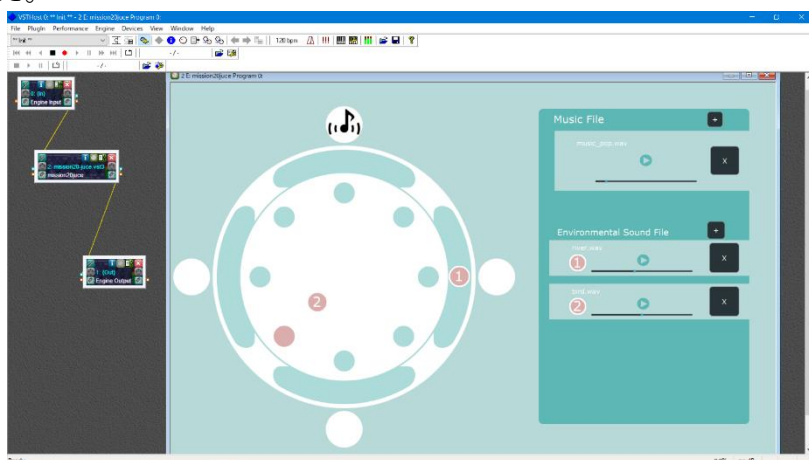


図 3：VST プラグインとしての動作

### 3.2 検証の結果と考察

本ソフトウェアを使用した立体音響演出の検証の様子を図 4 に示す。検証の結果、Sound Space の中心近くに配置した音源はより近くにあるように感じられ、中心から遠い位置に配置した音源はより遠くにあるように感じられた。いずれの場合も、音源のある方向を正確に感じ取ることができた。外周部分に配置した音源は、音の聞こえる方向がわずかにあいまいとなり、より”ユーザ周囲の環境音”としての特性を増したように感じられた。

環境音の再生中に Sound Space 上の音源オブジェクトを移動させることで、指定した方向に音源が実際に動いているように感じられた。画面上での操作と実際の空間での演出との連動が強化されたことによる GUI の直感性を確かめることができた。



図 4：立体音響演出システムの検証風景

### 3.3 学んだこと

開発したソフトウェアは、専門知識を必要とせずいかに直感的に操作できるかを追求して GUI のデザインを行った。エンジニアも含めたメンバー全員が GUI のデザインについて議論し、様々な視点からの意見を取り入れることで、ソフトウェアの設計思想を GUI に反映させることができた。これは、各々が考えるシステムのビジョンを言語化して共有することで、システムの要件定義・設計を行う経験だといえる。そのため、システム開発における上流工程に携わる

際に必要な知識や観点、取り組み方を身に着けることができたと考えられる。

昨年度の Python 開発とは異なり、JUCE を用いた開発では、フレームワークの提供しているテンプレートにならい、定義と実装を分けて行う開発を経験し、C++による効率的なチーム開発の方法を学ぶことができた。内部処理の実装では、オーディオデータの扱いや音声の入出力方式の違いを理解する必要があり、より一層サウンドプログラミングについての理解が深まった。また、JUCE コンポーネントを用いた GUI の実装では、オブジェクト指向に基づく GUI 開発の経験ができた。

## 4 今後の展開

本ミッションでは、目標に設定したシステムを完成させることができ、昨年度のシステム以上に演出が容易になっただけでなく、音声処理の安定性や遅延等の品質も格段に向上したため、ゲームやライブパフォーマンス等での実運用も行えるものであると考えられる。また、クロスプラットフォームアプリケーションとして開発したため、様々な環境でプラグインとして動作させることができ、昨年度のシステムと比較して活用の機会も格段に増加した。

そこで、Unity Native Audio Plugin としてビルドし、ゲームエンジンである Unity を用いたゲーム開発に組み込むことや、VST・AU・AAX プラグインとしてビルドし、作曲やライブパフォーマンスで広く活用されているデジタルオーディオワークステーション (DAW) 上で動作させることが今後の展開として考えられる。

Sound as a System プロジェクトで並行して遂行していたミッション「立体音響演出システムを用いたメディアアートの制作」では、立体音響演出を効果的に活用できるメディアアート作品について模索した。そこで探究した演出に、本ミッションで開発したシステムを効果的に応用することを、具体的な今後の展開として計画している。

## 5 まとめ

本ミッションでは、より多くの方に様々な場面で立体音響演出を楽しんでもらうことを目的に、昨年度開発した低コストで体験できる立体音響演出システムを改良した。昨年度開発したシステムでは、直感性を欠く GUI が採用されており、操作には一部専門知識が必要であった。また、Python アプリケーションとして動作していたため、プラットフォームの制限が大きかった。そこで、GUI の刷新とクロスプラットフォームアプリケーション化を目標に、C++を用いて開発を行った。

開発したシステムでは、専門知識を必要とせず直感的に操作できるよう GUI を刷新した。それに伴って、GUI で入力されたパラメータを用いて出力音声を制御できるよう、内部処理にも大幅な変更を加えた。その上で、クロスプラットフォームアプリケーションとしてビルドし、Windows ネイティブアプリケーションとして動作させるだけでなく、macOS 上での動作や VST プラグインとして作曲ソフト上での動作も行えるようにした。

これらの改良によって、システムを用いた演出はより直感的に行えるようになり、システムの活用機会も大幅に増加した。

本ミッションを通して、音声信号処理や GUI の実装についての理解が深まっただけでなく、オーディオプラグイン開発のデファクトスタンダードとなっているフレームワーク JUCE を用いた開発では、オーディオプラグイン開発の技術を体系的に習得できた。JUCE でのプラグイン開発の経験は、今後の応用の機会も多いと考えられる。

今後は、本ミッションで開発した立体音響演出システムと、Sound as a System プロジェクトで模索してきたメディアアートや空間演出手法とを融合させ、より多くの方に音による感動を提供していきたいと考えている。