

令和5年度 卒業研究

# バイオリン経験の浅い人をサポートするための アプリケーションケーションの研究開発

Research and development of applications to support people

with limited violin experience

函館工業高等専門学校  
生産システム工学科 情報コース  
杉本 颯天  
指導教員 東海林 智也

# 目次

1. 序論 .....	3
1.1 研究目的 .....	3
1.1.1 和文.....	3
1.1.2 英文.....	3
1.2 研究背景 .....	4
1.3 開発環境 .....	5
2. 関連技術 .....	6
2.1 pyaudio .....	6
2.2 MuseScore 4 .....	6
2.3 Music21 .....	7
3. アプリケーション概要 .....	8
4. 研究結果 .....	9
4.1 楽譜作成プログラム .....	9
4.2 録音プログラム .....	10
4.3 音声比較プログラム .....	11
5. 今後の展望 .....	12
謝辞 .....	13
参考文献 .....	14

# 1. 序論

## 1.1 研究目的

### 1.1.1 和文

本研究は、バイオリン演奏経験の浅い人に対して、開発したアプリケーションを通じて簡単な曲を最初から指導し、演奏できるところまで能力を引き上げることを目的としている。また、バイオリン演奏経験がある程度ある中級者以上に対して、本アプリケーションを通じて自分の弾きたい曲を選択して弾けるようにサポートすることも目的としている。

### 1.1.2 英文

The main target audience of this research is inexperienced violin players, and we aim to raise their ability to the point where they can play simple pieces by instructing them from the beginning through the application we have developed. The application is also intended to support intermediate and advanced violin players who have some experience in playing the violin by helping them select the pieces they wish to play.

## 1.2 研究背景

スマートフォンやPCのアプリケーションにはピアノ経験の浅い人をサポートするアプリケーションが数多くあるが、バイオリン経験の浅い人をサポートするアプリケーションはあまり存在しない。そこで本研究ではバイオリン経験の浅い人をサポートするためのアプリケーション開発を行う事にした。

## 1.3 開発環境

PC : DELL

OS : Windows11

プログラミング言語 : python

開発環境 : Anaconda, Jupyter Notebook

使用ライブラリ : pyaudio

MuseScore 4

Music21

## 2. 関連技術

### 2.1 pyaudio

pyaudio は録音や再生などオーディオ処理を行うライブラリであり、本研究では Anaconda を用いてインストールを行った。

### 2.2 MuseScore 4

本研究では楽譜を作成するために 2 つのライブラリを使っている。ひとつ目が MuseScore 4 であり、以下の手順で使用できる。

1. MuseScore をダウンロードしてインストールする
2. 初回起動時にウィザードに従ってテーマやアクセントカラーを選択する
3. 使用している DAW でメロディの MIDI ファイルを作成する
4. MuseScore 4 で MIDI ファイルを読み込み、歌詞を入力する
5. テンポ (BPM) を合わせる
6. Musicxml ファイルとして書き出す

## 2.3 Music21

本研究で楽譜作成に使用しているもう1つのライブラリが Music21 である。Music21 は主に楽譜や音楽を分析するために利用されている。Music21 を実行すると内部 MuseScore4 が呼び出されて楽譜が表示される(図 1)。さらに再生ボタンを表示して音声再生することも可能である(図 2)。



図 1 Music21 を実行した時の楽譜

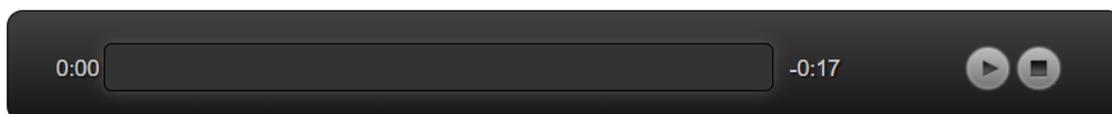


図 2 Music21 を実行した時表示される再生ボタン

### 3. アプリケーション概要

本研究で開発したアプリケーションは、「楽譜作成プログラム」、「録音プログラム」、楽譜と録音した音声を比較する「音声比較プログラム」の3つのプログラムから構成されている。演者はあらかじめ用意しておいた楽譜に従ってバイオリンを弾き、楽譜と録音した音とを比較することで間違えている音を見つけることが可能となる。

## 4. 研究結果

### 4.1 楽譜作成プログラム

music21 と MuseScore 4 を用いて楽譜作成プログラムを作成した[1]。図 3 はカエルの合唱の楽譜を表示するためのプログラムであり、C4、D4、E4、F4、G4、A4 の様にコードと音長を指定する事で楽譜表示を行う事ができる。図 3 を実行すると図 4 の楽譜が表示される。

```
4
5 # 新しい楽譜を作成
6 score = stream.Score()
7 #カエルの合唱
8 # パートを追加
9 part = stream.Part()
10 score.append(part)
11
12 # 拍子やキー、テンポの設定
13 part.append(metadata.Metadata())
14 part.metadata.title = "カエルの合唱"
15 part.append(instrument.Violin())
16 part.append(meter.TimeSignature("4/4"))
17 part.append(key.KeySignature(0)) # キー: Cメジャー
18 part.append(tempo.MetronomeMark(number=120)) # テンポ: 120 BPM
19 # メロディと休符の生成 (Cメジャースケール)
20 notes_data = [
21     ("C4", 1), #ド
22     ("D4", 1), #レ
23     ("E4", 1), #ミ
24     ("F4", 1), #ファ
25     ("E4", 1), #ミ
26     ("D4", 1), #レ
27     ("C4", 2), #ド
28     ("E4", 1), #ミ
29     ("F4", 1), #ファ
30     ("G4", 1), #ソ
31     ("A4", 1), #ラ
32     ("G4", 1), #ソ
33     ("F4", 1), #ファ
34     ("E4", 2), #ミ
35     ("C4", 2), #ド
36     #間を開けるか音を伸ばす
37     ("C4", 2), #ド
```

図 3 楽譜作成プログラムの一部



図 4 実行結果の楽譜

## 4.2 録音プログラム

図5は録音を行うプログラムである[2]。実行すると図6の様に指定した時間まで録音が行われる。

図5 録音プログラム

```
5 FORMAT = pyaudio.paInt16
6 CHANNELS = 1
7
8 #サンプリングレート
9 RATE = 44100
10
11 #録音時間を入力
12 RECORD_SECONDS = 20
13
14 p = pyaudio.PyAudio()
15
16 stream = p.open(
17     format = FORMAT,
18     channels = CHANNELS,
19     rate = RATE,
20     input = True,
21     frames_per_buffer = chunk
22 )
23
24 all = []
25 for i in range(0, int(RATE / chunk * RECORD_SECONDS)):
26     data = stream.read(chunk)
27     all.append(data)
28     print(i)
29 stream.close()
30 p.terminate()
31
32 data = b''.join(all)
33
34 #保存するファイル名、wは書き込みモード
35 out = wave.open('sample.wav', 'w')
36 out.setnchannels(1)
37 out.setsampwidth(2)
38 out.setframerate(RATE)
39 out.writeframes(data)
40 out.close()
```

図6 録音時間の表示

```
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
```

## 4.3 音声比較プログラム

音声比較プログラムについてはPCの容量不足等により実行することが出来なかった(図7)[3]。

```
1 import librosa
2 import librosa.display
3 import matplotlib.pyplot as plt
4
5 def load_audio(file_path):
6     audio, sr = librosa.load(file_path)
7     return audio, sr
8
9 def plot_waveform(audio, title="Waveform"):
10    plt.figure(figsize=(15, 4))
11    librosa.display.waveshow(audio, alpha=0.5)
12    plt.title(title)
13    plt.show()
14
15 # 音源ファイルのパスを指定
16 source_file_path = "path/to/your/カエルの合唱楽譜.ipynb"
17
18 # 録音した音声ファイルのパスを指定
19 recorded_file_path = "path/to/your/Rokuon.ipynb"
20
21 # 音源の読み込み
22 source_audio, source_sr = load_audio(source_file_path)
23 plot_waveform(source_audio, title="Source Audio")
24
25 # 録音した音声の読み込み
26 recorded_audio, recorded_sr = load_audio(recorded_file_path)
27 plot_waveform(recorded_audio, title="Recorded Audio")
```

図7 音声比較プログラム

## 5. 今後の展望

今回はPCの容量不足もあってアプリケーションの完成までに至らなかった。そこで  
今後は開発環境を再構築してアプリケーションの完成を目指す。そして完成後に実証  
実験を行う。

## 謝辞

本研究を進めるにあたって、ご指導くださいました東海林智也准教授と協力して頂いた研究室のメンバーに感謝いたします。

## 参考文献

[1] 楽譜作成プログラム

[python プログラミング で楽譜作成！ | music21 ～基本操作① | 魅せる IT 編集部 \(miseruit.com\)](#)

[2] 録音プログラム

[Python で音を録音して保存する方法 |  \$\beta\$  short Lab \(betashort-lab.com\)](#)

[3] 音声比較プログラム

[初心者でもできる！Python で音声認識プログラムを作る方法まとめ | AIZINE \(エイジン\)](#)