

令和2年度 卒業研究
簡単なブロック操作による
作曲アプリケーションの開発
Development of Musical Composition Application
by Simple Block Operations

函館工業高等専門学校

生産システム工学科 情報コース

長谷川 湧大

指導教員 東海林 智也

目次

第1章 序論

- 第1節 英文アブストラクト
- 第2節 研究背景
- 第3節 研究目的
- 第4節 環境開発

第2章 関連技術

- 第1節 wx.Python
- 第2節 python-vlc
- 第3節 BandLab

第3章 アプリケーションの開発

- 第1節 インターフェースの作成
- 第2節 音源の作成
- 第3節 実行結果

第4章 まとめ

参考文献

第 1 章 序論

第 1 節 英文アブストラクト

Pick up on the atmosphere of compose music is the most difficult for an inexperienced person. The atmosphere of the melody, the rhythm, and the flow of the composition. By getting the general atmosphere, you can gradually build up the song. It is difficult to maintain motivation because those atmospheres can only be cultivated through actual experience.

The purpose of this research is to develop an application that allows users to compose music intuitively, thereby lowering the bar for compose music.

Specifically, I am creating a GUI application using python that can create a rough song by simply arranging blocks with song information added.

Key words : compose music, GUI application, arranging block

第2節 研究背景

未経験者が作曲する際に一番難しいことは作曲の雰囲気をつかむことである。メロディの雰囲気、リズムの雰囲気、作曲の雰囲気などを掴むことで、徐々に曲ができていく。しかしこれらは実際に経験しないと培われないため、モチベーションの維持が難しい。

そこで、簡単な操作のみで作曲できるアプリケーションを作ることによって、作曲する際の雰囲気を未経験者に掴んでもらうことができると考えた。

第 3 節 研究目的

本研究の目的は、簡単な操作のみで直感的に作曲ができるアプリケーションを開発し、未経験者が作曲する上でのハードルを下げることである。具体的には、曲情報を付加したブロックを並べるだけで大まかな楽曲が作れる GUI アプリケーションを Python[1]により作成する。

第 4 節 開発環境

PC : OS Ubuntu 18.04.5LTS
CPU Intel Core i5-2300 CPU @2.80Ghz × 4
RAM 8.00GB

使用言語 : Python[1]

ライブラリ : wx.Python[2][3]
python-vlc[4]

開発ツール : Jupyter Notebook
BandLab[5]

第 2 章 関連技術

第 1 節 wx.Python

wx.Python[2][3]は Python を使って GUI アプリケーションを作るためのライブラリである。ウィジェットが豊富で外見や操作感をカスタマイズしやすい点や海外の参考文献が多い点を踏まえ、本研究ではこのライブラリを用いて開発を行うことにした。

第 2 節 python-vlc

python-vlc[4]は、Python を用いて音楽再生を制御するためのライブラリである。ほぼ全てのマルチメディアフォーマットに対応している「vlc media player」を制御できるため、本研究ではこのライブラリを用いて開発を行うことにした。

第 3 節 BandLab

BandLab[5]は、Web 上で利用できる無料の DTM アプリである。他の作曲アプリに比べて使用できる楽曲の種類が多く、Web 上で作曲できるため、本研究では BandLab を用いて音源(音声ファイル)作成を行うことにした。

第3章 アプリケーションの開発

第1節 インターフェースの作成

はじめに、プログラミング学習アプリケーションである Scratch の様に、ブロックをドラッグ&ドロップし並び変えることで直感的な操作により作曲できるインターフェースを持つアプリケーションを作成した。図1に作成したアプリケーションの実行画面を示す。画面左側にある色付きの各ブロックには別々の音源(音声ファイル)が割り当たっており、画面右側のメインパネル内にブロックを配置したり並び変えたりすることでブロックを再生するタイミングを変更できる。またメインパネルは4つのレーンに分かれており、上下に配置したブロックは同時に再生される。

作曲が終わったら左上の「play」ボタンをクリックすることでメインパネルの左側に配置されたブロックから順に再生される。なお図1ではブロックは3つしか表示されていないが実際には複数個ある。

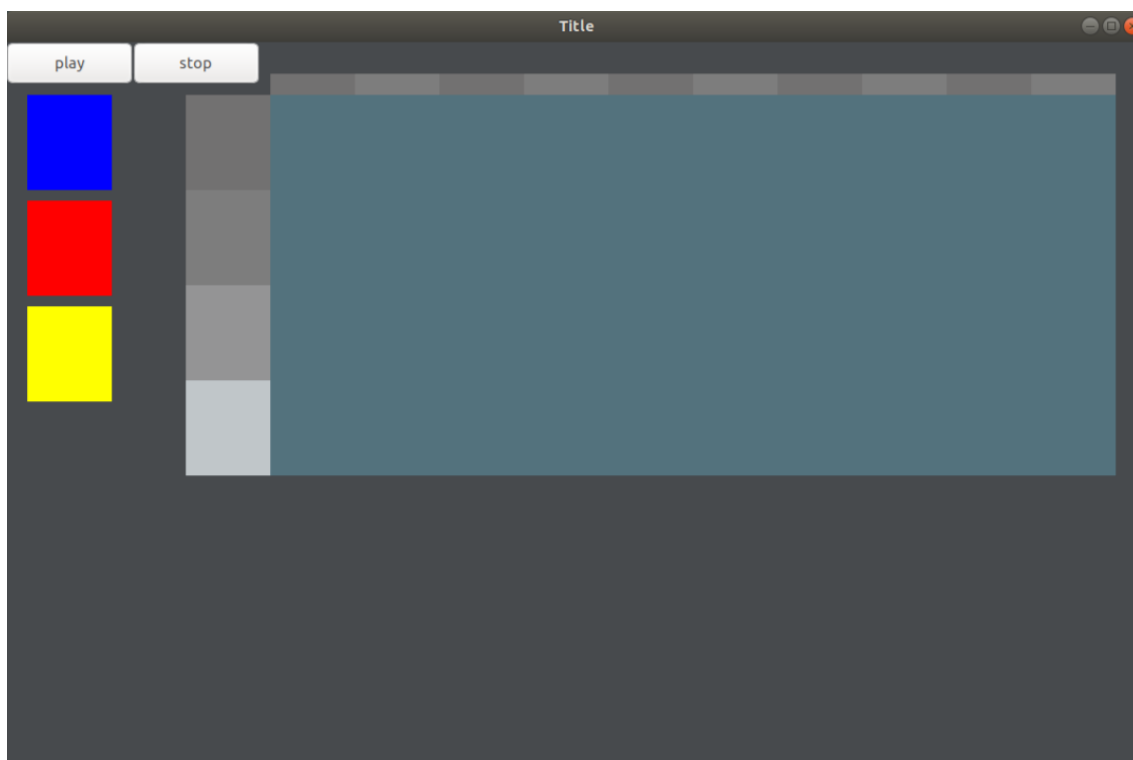


図1 実行画面

第2節 音源の作成

次に各ブロックに割り当てる音源(音声ファイル)を BandLab[5]を用いて作成した。

なおブロックは「メロディブロック」「コードブロック」「ドラムブロック」の3種類に分けられる。メロディブロックの音源は、基礎的なコードである「王道進行」「小室進行」「JTTU 進行」を組み替えた合計7つのコードから作成した[6]。ドラムブロックの音源は、基礎的なリズムである「8beat」「16beat」「4つ打ち」からループ音源として作成した。

第3節 実行結果

完成したアプリケーションを実行したところ、音楽は無事に再生されてディレイ等も無かった。また被験者に作成したアプリケーションの評価をしてもらったところ、操作を迷うこともなく高評価が得られた。よって本研究の「直感的に作曲ができる洗練されたデザインの GUI を持つアプリケーションを開発して作曲のハードルを下げる」という目的は達成できたと考えられる。

第4章 まとめ

簡単な操作のみで直感的に作曲ができるアプリケーションを開発して未経験者が作曲する上でのハードルを下げるといふ本研究の目的は達成できた。

ただし今の段階ではブロックの数が少なく、ブロックを適当につなぐと不自然な曲が再生されるときがある。またブロックを並べ変える以外の複雑な操作をすることが出来ない。

そこで今後はブロックの数を増やし、コード進行やキーデンス情報を表示する、曲の雰囲気に合わせておすすめブロックを表示するなどの機能を追加する。

また、作曲経験が少ないと曲がきれいに聞こえるまで少しずつ調整していくことが多いので、再生バーを追加して細かな調整ができるようにする。

また、見た目に関しても通常のテーマ以外にもダークテーマ等のテーマを複数用意する。様々なテーマを用意することで、ユーザーが長時間の作曲をする際に目を傷めない効果がある。またテーマを変えることで作曲の効率が上がって新しいアイデアにもつながるため、作曲に対しても効果的であると考えられる。

さらに、作曲未経験者のためのヒント機能を追加する。未経験者が作曲をする際は、作曲におけるヒントが多いと手軽に作曲できて長続きすると考えられる。

参考文献

- [1] Python: <https://www.python.org/>
- [2] wx.Python: <https://www.wxpython.org/>
- [3] wx.Python: <https://maku77.github.io/python/wxpython/drag-component.html>
- [4] python-vlc: https://jellyware.jp/kurage/raspi/python_vlc.html
- [5] BandLab: <https://www.bandlab.com/products/cakewalk>
- [6] music code : <https://sakky.tokyo/post-3586/>