

令和 5 年 6 月 13 日現在

機関番号：13201

研究種目：基盤研究(C) (一般)

研究期間：2020～2022

課題番号：20K03175

研究課題名(和文) 教員志望文系学生を対象としたプログラミング教育カリキュラムの開発

研究課題名(英文) Development of a programming education curriculum for liberal arts students aspiring to become teachers

研究代表者

黒田 卓 (Kuroda, Takashi)

富山大学・学術研究部教育学系・教授

研究者番号：80262468

交付決定額(研究期間全体)：(直接経費) 1,400,000円

研究成果の概要(和文)：本研究では、Sonic Piというプログラミング言語を用いて、小学校の教員を目指す学生を対象とした、音楽の楽典とプログラミングを同時に学べる教材を開発した。

音楽の楽譜は、コンピュータプログラムと同様に時系列的な指示が表現されたものである。この共通性を利用して、音楽の楽典とプログラミングを同時に学ぶことが可能となる。コンピュータの基本的な処理構造である「順次処理」、「反復」、「条件分岐」は、楽譜では、「小節」や記譜方法、「リピート(反復)記号」、「1番かっこや2番かっこ」で表記される。これらを実際に行いながら学習できる教材を作成した。

研究成果の学術的意義や社会的意義

本研究では、プログラミングを用いて音楽を演奏することを通して、文系の学生でも取り組みやすい、教員養成におけるプログラミング教育カリキュラムおよび、小・中・高等学校のプログラミングの授業でそのまま利用できる教材の開発を行った。音楽の学習はこれまで芸術的視点で行われていたが、特に楽譜は演奏の時系列的指示を表記したプログラムであると考え、プログラミング言語の理解との共通性も高い。それを利用し、プログラミングを苦手とする教員志望の文系学生でも、音楽の学習として取り組みながら、プログラミング的思考の育成や、身近なものの動き等をプログラミング的に考えることができるようにすることができる。

研究成果の概要(英文)：In this research, using the programming language called Sonic Pi, we developed a teaching material that allows students to learn musical notation and programming at the same time for students aiming to become elementary school teachers.

A musical score expresses chronological instructions, just like a computer program. By using this commonality, it becomes possible to learn musical notation and programming at the same time. "Sequential processing", "Repetition", and "Conditional branching", which are the basic processing structures of a computer, are represented by "measures", notation methods, "repeat (repetition) symbols", "first parentheses and second marked with brackets". We have created teaching materials that allow students to learn while actually doing these things.

研究分野：教育工学

キーワード：プログラミング教育 カリキュラム開発 教員志望文系学生 総合的な学び SonicPi

1. 研究開始当初の背景

Society5.0、IoT、AI、ビッグデータといった言葉が様々な政策文書において取り扱われ、新学習指導要領においても、小学校におけるプログラミング教育が導入されるなど、これからの時代に求められる能力の一つとして、プログラミング教育に注目が集まっている。社会生活の様々な場面において ICT があたりまえのように活用される時代になっているが、それらの仕組みについての理解はまだ不足しており、その結果、情報を使えるものとそうでないもの間の格差は、ますます広がっていくことが予想される。

コンピュータを始めとする情報機器は、理系・文系を問わず様々な分野ですでに利用されているが、高校で理系・文系に分けて指導が行われる日本においては特に文系学生からは縁のない遠い存在として見られるということが長く続いてきている。

平成 14 年の学習指導要領改訂において、高等学校に教科「情報」が選択必修科目として設置され、平成 23 年の改訂においてすでに普通教科情報科「情報の科学」においてプログラミングが取り入れられたが、実際にこれを履修した生徒は少なく、高等学校までにプログラミングについて学習した学生は少ない。平成 30 年の改訂では「情報 I」(必修)、「情報 II」(選択)に変わり、「情報 I」の内容としてプログラミングが明記されたことにより、殆どの生徒がプログラミングについて学ぶことになった。同時に、小学校の「総合的な学習の時間」、「算数」、「理科」の一部や、中学校「技術科」にプログラミングによる計測・制御といった内容が盛り込まれ、何らかの形でプログラミングを学んで、大学に進学してくるようになることが期待される。しかしながら、学校における学習環境の整備の遅れやプログラミングに関する指導能力を持たない教員が指導しなければならないことを考えると、それほど大きな変化は期待できないという見方もある。

このような状況を踏まえ、本研究では、プログラミング教育の目的を改めて見直し、その実施・展開の中心となるはずである教員養成系の文系学生に対し、敷居を低くし、取り組みやすい、しかしながら内容的には必要な事項を網羅したカリキュラムを開発し、これらの実践を通して、授業でそのまま利用できる教材の作成を行う必要性を感じるに至った。

2. 研究の目的

新学習指導要領の本格実施に伴い、プログラミング教育への注目が集まり、様々な実践が始まっている。教員を目指す学生にとっても、学生時代にプログラミングについて習得しておくことが求められているが、教員養成課程で教員を目指す学生、特に幼稚園、小学校教諭を目指す学生の多くは文系の学生であり、どちらかというプログラミングに関しては、理系のイメージが強く、やらなければならないという思いは強いが、とっつきにくさを感じている学生も多い。

そこで、本研究では、プログラミングを用いて音楽を演奏することを通して、文系の学生でも取り組みやすい、教員養成におけるプログラミング教育カリキュラムおよび、小・中・高等学校のプログラミングの授業でそのまま利用できる教材の開発を行うことを目的とする。

3. 研究の方法

本研究では、研究の進め方として以下の 2 段階を考え、取り組んだ。

本研究では、まず、学習指導要領、協力者会議の議論、国内外の 21stSkills に関する理論的および実践事例に関する文献等の調査を実施した。特に、教員養成、現職教育等指導者育成でのプログラミング教育の取り扱いに注目し、これらを整理し、すべての子どもたちが身につけるべきプログラミング教育に求められる内容と、それを指導する教員に求められる能力を明らかにした。

次に、これらの成果を元に、教員養成におけるプログラミング教育のカリキュラムモデルの開発を行う。また、カリキュラムへの取り組みやすさ、学習意欲の維持等の視点からアンケート等によりその効果について評価を、また、これらの実施を通して発生した問題点等についてその解決方法の検討を行う。プログラミング言語には、英ケンブリッジ大の Sam Aaron 氏が開発したサウンドプログラミング言語 Sonic Pi を用いる。Sonic Pi は、まつもとひろゆき氏が開発し世界的に利用されている Ruby 言語をベースとして、音響合成技術と合わせてプログラムで本格的に音楽を表現し、演奏することのできる言語である。

ここでは、Raspberry Pi を用い、各個人のスタンドアロンの学習環境においてこれらを実施し、コンピュータのハードウェア的な面の簡単な学習内容も合わせて学習できるようにする。さらに、カリキュラムの修正を行い、継続してその内容の評価に取り組むとともに、本カリキュラムを修了した学生とともに、各学校段階の学習において利用できる教材の開発を行った。

4. 研究成果

開発した教材

プログラムと楽譜の共通性に着目し、プログラミングと音楽を同時に学ぶ教材を開発した。

教材の内容

本教材は、次のような内容で構成されている。

- ① Sonic Pi とは？
- ② プログラミングで考える時の3つの基本
 - 1 音を鳴らしてみよう！「順次」
- ③ 1拍って何秒？
 - 2 同じ処理を繰り返してみよう！「反復」
 - 3 リストを使ってみよう！
 - 4 while 文を使ってみよう！
 - 5 loop do を使ってみよう！
 - 6 もしも～なら実行してみよう！「分岐」
 - 7 再生時のオプションを指定しよう！
- ④ リリース・タイムって何？
 - 8 attack と release を組み合わせた命令をしてみよう！
 - 9 chord や scale を使ってみよう！
- ⑤ minor(長調)と minor(短調)の違いって何？
 - 10 ランダムに音を再生してみよう！
- ⑥ sonic pi の音は実はランダムではない！？
 - 11 乱数を扱ってみよう！
 - 12 音色を変えて音を鳴らしてみよう！
 - 13 サンプル音源を使ってみよう！
 - 14 live_loop を使ってみよう！
- ⑦ まとめ

教材の概要

本教材は、プログラミングの基本となる「順次」「反復」「分岐」の3つを意識して構成したものとなっている。以下に作成した教材の一部を紹介する。

図1は、さくらさくらの楽譜である。楽譜は最初の小説から順に演奏を行う(順次処理)構造になっている。音符はそれぞれの音の高さと長さを表しており、それをプログラムで表すと図2のように表すことができる。

The image shows a musical score for the song 'Sakura Sakura' in G major, 4/4 time, with a tempo of 72. The score is written on a single treble clef staff. The lyrics are written below the notes, and chords are indicated above the staff. The lyrics are: さくら さくらの やまも さとーも み わ た す か ぎーり か す み か く もーか あ さ ひ に に おーう さくら さくらは な ざーか り

図1 さくらさくらの楽譜

```

1 use_bpm 72 14 play:a5 30 play:e5 46 play:a5 62 play:e5 78 play:a5 90 play:e5
2 play:a5 15 sleep 1 31 sleep 1 47 sleep 1 63 sleep 1 79 sleep 1 91 sleep 1
3 sleep 1 16 play:b5 32 play:c5 48 play:b5 64 play:c5 80 play:a5 92 play:f5
4 play:a5 17 sleep 1 33 sleep 1 49 sleep 1 65 sleep 1 81 sleep 1 93 sleep 1
5 sleep 1 18 play:c6 34 play:e5 50 play:c6 66 play:e5 82 play:b5 94 play:b5
6 play:b5 19 sleep 1 35 sleep 1 51 sleep 1 67 sleep 1 83 sleep 2 95 sleep 0.5
7 sleep 2 20 play:b5 36 play:f5 52 play:b5 68 play:f5 84 play:a5 96 play:a5
8 play:a5 21 sleep 1 37 sleep 1 53 sleep 1 69 sleep 1 85 sleep 1 97 sleep 0.5
9 sleep 1 22 play:a5 38 play:e5 54 play:a5 70 play:e5 86 play:a5 98 play:f5
10 play:a5 23 sleep 1 39 sleep 1 55 sleep 1 71 sleep 1 87 sleep 1 99 sleep 1
11 sleep 1 24 play:b5 40 play:e5 56 play:b5 72 play:e5 88 play:b5 100 play:e5
12 play:b5 25 sleep 0.5 41 sleep 0.5 57 sleep 0.5 73 sleep 0.5 89 sleep 2 101 sleep 2
13 sleep 2 26 play:a5 42 play:c5 58 play:a5 74 play:c5
27 sleep 0.5 43 sleep 0.5 59 sleep 0.5 75 sleep 0.5
28 play:f5 44 play:b4 60 play:f5 76 play:b4
29 sleep 2 45 sleep 2 61 sleep 2 77 sleep 2

```

① ② ③ ④ ⑤ ⑥ ⑦

図2 さくらさくらのプログラミングコード

図2の例では、単に音符で表されている1つひとつの音の高さと長さをコード化した命令の並びとして表現している。これでも演奏は可能であるが、プログラミングの効率化のため、音の高さと長さをデータとして扱えるように「反復処理」を利用して表現しなおしてみると、図3のように表現できる。

```

1 use_bpm 72
2 A=[:a5,:a5,:b5,:a5,:a5,:b5,:a5,:b5,:c6,:b5,:a5,:b5,:a5,:f5,:e5,:c5,:e5,:f5,:e5,:e5,:c5,:b4,:a5,:b5,
   :c6,:b5,:a5,:b5,:a5,:f5,:e5,:c5,:e5,:f5,:e5,:e5,:c5,:b4,:a5,:a5,:b5,:a5,:a5,:b5,:e5,:f5,:b5,:a5,:f5,:e5]
3 B=[1,1,2,1,1,2,1,1,1,1,1,1,0.5,0.5,2,1,1,1,1,1,0.5,0.5,2,1,1,1,1,1,0.5,0.5,2,1,1,1,1,1,0.5,0.5,2,1,
   1,2,1,1,2,1,1,0.5,0.5,1,2]
4 for i in 0..49
5   play(A[i])
6   sleep(B[i])
7 end

```

図3 反復処理を利用し、データを分けた例

このように表現することにより、データ部分のみを別の曲のデータを入れ替えるだけで、違う曲でも演奏できるようになることを理解させる。

同様に、whileを用いた「反復処理」のプログラムを図4に示す。

```

1 use_bpm 72
2 A=[:a5,:a5,:b5,:a5,:a5,:b5,:a5,:b5,:c6,:b5,:a5,:b5,:a5,:f5,:e5,:c5,:e5,:f5,:e5,:e5,:c5,:b4,:a5,:b5,
   :c6,:b5,:a5,:b5,:a5,:f5,:e5,:c5,:e5,:f5,:e5,:e5,:c5,:b4,:a5,:a5,:b5,:a5,:a5,:b5,:e5,:f5,:b5,:a5,:f5,:e5]
3 B=[1,1,2,1,1,2,1,1,1,1,1,1,0.5,0.5,2,1,1,1,1,1,0.5,0.5,2,1,1,1,1,1,0.5,0.5,2,1,1,1,1,1,0.5,0.5,2,1,
   1,2,1,1,2,1,1,0.5,0.5,1,2]
4 i=0
5 while i<50
6   play(A[i])
7   sleep(B[i])
8   i=i+1
9 end

```

図4 図3のプログラムをwhileで書き直したもの

図1、図2のプログラムでは、繰り返しの回数を明示的に示しておく必要があるが、さまざまな長さの曲データに、データ部分のみを入れ替えることで演奏を可能とするため、分岐処理を入れてデータの終了判定を行えるように改良したプログラムを図5に示す。

図5 開発した教材の例4

```
1 use_bpm 72
2 A=[:a5,:a5,:b5,:a5,:a5,:b5,:a5,:b5,:c6,:b5,:a5,:b5,:a5,:f5,:e5,:c5,:e5,:f5,:e5,:e5,:c5,:b4,:a5,:b5,
   :c6,:b5,:a5,:b5,:a5,:f5,:e5,:c5,:e5,:f5,:e5,:e5,:c5,:b4,:a5,:a5,:b5,:a5,:a5,:b5,:e5,:f5,:b5,:a5,:f5,:e5,999]
3 B=[1,1,2,1,1,2,1,1,1,1,1,1,0.5,0.5,2,1,1,1,1,1,0.5,0.5,2,1,1,1,1,1,0.5,0.5,2,1,1,1,1,1,0.5,0.5,2,1,
   1,2,1,1,2,1,1,0.5,0.5,1,2]
4 i=0
5 loop do
6   if A[i]==999
7     break
8   else
9     play(A[i])
10    sleep(B[i])
11    i=i+1
12  end
13 end
```

教材の評価

本教材の問題点や改善点を明らかにすることを目的に、現職教員（小・中・高・特別支援教員）に本教材を見てもらい、意見を頂いた。調査は質問紙法を用い、令和5年1月に実施した。9名の教職大学院学生より回答をいただいた。その一部を下記に示す。

Q1. 本教材でプログラミング言語の意味を理解できたか

- ・プログラミングの知識がない人には、少し難しい。
- ・非常にシンプルかつ分かりやすい言葉で説明されている。
- ・文字だけの説明が多いので分かりづらく感じた。
- ・コードの意味は理解できるが、1から自分でコードを組み立てるとなると難しい。

Q2. プログラミングに興味を持てたか

- ・プログラミングで音を鳴らすことができるという経験ができることは、それだけで興味深いから。
- ・説明と文字ばかりであまり興味が湧かなかった。
- ・音の高さや拍数を自由に変えることができるのでおもしろいと思ったから。
- ・演奏することに対するハードルの高さを感じ、あまり興味が湧かなかった。

Q3. 本教材の改善すべき点はどこか

- ・画像や動画を組み合わせることでより理解しやすい教材となる。
- ・動画などで生徒たちが見ながらやることができれば、教師の負担も少なくなり、生徒の興味もよりひくことができる。
- ・教材自体はおもしろいので、解説動画などでの説明があったらもっと理解しやすい教材となる。

本研究では、プログラミングを用いて音楽を演奏することを通して、文系の学生でも取り組みやすい、教員養成におけるプログラミング教育カリキュラムおよび、小・中・高等学校のプログラミングの授業でそのまま利用できる教材の開発を行った。プログラムが苦手な文系の学生だけでなく、楽器を演奏することが苦手な学生にとっても、音楽とプログラミングの両方に興味を持ってもらえる可能性があることが分かった。

今後、教材の表記の仕方や、内容の再構成を行い、より使いやすい教材に仕上げていく必要がある。また、解説動画なども導入することで、より理解しやすい教材ができると考える。

5. 主な発表論文等

〔雑誌論文〕 計0件

〔学会発表〕 計0件

〔図書〕 計0件

〔産業財産権〕

〔その他〕

-

6. 研究組織

	氏名 (ローマ字氏名) (研究者番号)	所属研究機関・部局・職 (機関番号)	備考
--	---------------------------	-----------------------	----

7. 科研費を使用して開催した国際研究集会

〔国際研究集会〕 計0件

8. 本研究に関連して実施した国際共同研究の実施状況

共同研究相手国	相手方研究機関
---------	---------