

ビジュアル型言語からテキスト型言語への シームレスな移行について

梅澤 克之[†] 中澤 真[‡] 平澤茂 一^{††}
 湘南工科大学[†] 会津大学短期大学部[‡] 早稲田大学^{††}

1. はじめに

近年、プログラミングの入門としてビジュアル型のプログラミング言語（以降、ビジュアル型言語と呼ぶ）が使われるようになってきている。その後はC言語やJava言語などのテキスト型プログラミング言語（以降、テキスト型言語と呼ぶ）に移行していくことになる。しかしシームレスな移行方法は確立されていない。本研究プロジェクトでは、ビジュアル型言語からテキスト型言語への移行の方法論を確立することを目的とする。本研究が確立されると、プログラミング言語の初学者がビジュアル型言語による学習から始めて、その後、シームレスかつ自動的にテキスト型言語の学習に遷移できるようになることが期待できる。

我々は、2021年前学期に高校生向けに音楽を作成するという課題を扱うプログラミング授業を行った。本報告では、我々が行った高校生向けのプログラミング授業が、ビジュアル型言語とテキスト型言語の中間に位置づけられる特徴を有することをアンケートにより評価する。

2. 従来研究

Mladenović et al. [1] は、プログラミングの基本概念的な1つであるループについて、小学生を対象に調査を実施し、構文の問題が解消されるためプログラミングの初学者向けにはビジュアル型言語を使用する正当性を主張している。ただしこの研究では両言語のギャップを埋める事には言及していない。

Daskalov et al. [2] は、初学者向けのテキスト型言語とビジュアル型言語のハイブリッド型言語を使用するための環境を提案している。ブロック型ではなくフロー型のビジュアル言語とテキスト型言語のハイブリッド環境であり、初学者のプログラミングトレーニングに適していると主張している。

Weintrop [3]は、テキスト型言語とビジュアル型言語とハイブリッド型言語の比較を行った。

Seamless transition from visual-type to text-type languages

[†] Katsuyuki Umezawa, Shonan Institute of Technology

[‡] Makoto Nakazawa, The University of Aizu

^{††} Shigeichi Hirasawa, Waseda University

結論として、ハイブリッド型言語が両方の祖先の特徴を示している一方で、特定の領域ではブロック型言語とテキスト型言語よりも優れていることを明らかにした。

3. 中間コンテンツの提案

3.1 概要

現状のビジュアル型言語は見た目が楽しい、すぐ実行して動作する、文法エラーが存在しないなどの特徴がある。これに対してテキスト型言語は、文字ばかりであり、1文字でも間違えれば動作せず、グラフィカルなことをやろうとするととても手間がかかる、というような特徴がある。提案する中間コンテンツは、シンプル（テキスト型言語特有の追加知識不要）、素早いフィードバック（実行結果がすぐわかる）、文法エラーが起こりにくく、論理エラーの箇所がわかり易い、というような特徴を持つ必要があると考えた。

3.2 中間コンテンツを用いた授業について

前述の特徴を持ったコンテンツとして、JavaScriptで音楽を作るという授業を行った。使用したシステムはJSFiddle [4]を活用した。JSFiddleは、Webブラウザのみを用いてJavaScriptのコーディングから実行結果の表示までを1つの画面で行えるWeb版の統合環境である。このJSFiddleに音を出すためのライブラリ（Beeplay）を追加することで簡単に音楽を作れる。例えば“ドレミ”と音を出すには図1のようにコーディングすれば良い。

```
function myPlay() {
  var song1 = beeplay();
  song1.play('C4', 1);
  song1.play('D4', 1);
  song1.play('E4', 1);
}
```

図1 ドレミの音を出すプログラム

上記、音を鳴らすプログラムは、playというメソッドのみで実現できるため、多くの単語（予約語）や文法を覚える必要がなく、実行したらすぐ音が鳴り、結果が素早く確認できる。

さらに、期待した音と違う音になってしまうような論理エラー（プログラムは間違えていないのでコンパイラはエラーを出力しない）の場所の音を聞いて容易に判断できる。また、音楽は繰り返し構造があり、また、2 回目の繰り返しの時には異なる音を鳴らす、というようなこともある。これらは繰り返し（for 文）や条件分岐（if 文）としてプログラムで表せる。さらに小節毎に繰り返すような構造もあり、小節を関数（function）として定義することもできる。このように音楽とプログラミングはとても相性が良いと思われる。また、なんとといっても自分の好きな音楽が徐々に出来上がっていくのはとても楽しい。

3.3 授業への参加者

湘南工科大学附属高等学校の3年生（21名）に対して、2021年度に7回の授業を実施した。

4. アンケートによる評価

4.1 アンケート項目

アンケートは下記の設問とした。回答は全て4段階とした。

設問1: JsFiddleでの音楽作成は楽しかったですか（1:楽しかった～4:楽しくなかった）

設問2: JsFiddleでの音を鳴らすプログラムは、覚えなければならない英単語(playなど)は多いと思いますか、少ないと思いますか（1:とても少ない～4:とても多い）

設問3: JsFiddleでの音を鳴らすプログラムは、結果がすぐわかりますか（実行後すぐ音を確認できるなど）（1:実行結果がすぐわかる～4:すぐわからない）

設問4: JsFiddleでの音を鳴らすプログラムは、間違えている箇所がわかりやすいですか（音がずれている箇所が耳で聞いてすぐわかるなど）（1:わかりやすい～4:わかりにくい）

4.2 アンケート結果

3.3節で述べた授業を受けた高校生のうち、研究倫理に同意してもらえた13名からアンケートを収集した。結果を表1に示す。

5. 考察

表1に示したように、今回実施したJavaScriptで音楽を作るプログラミング授業は、ビジュアル型言語とテキスト型言語の間に位置づけられる特徴を有することがアンケート結果より明らかになった。なお覚える単語が多い

表1 アンケート結果

設問	平均値	最頻値	解釈
1	1.77	1	少し楽しかった
2	2.46	3	覚えるべき英単語は「少ない」と「多い」の間あたり
3	1.54	1	実行結果が「すぐわかる」と「ややすぐわかる」の間
4	1.85	1,2	やや間違えている箇所がわかりやすい

か少ないかのアンケート結果に関しては、まだテキスト型言語を学習していない段階でのアンケートのため、本格的にテキスト型言語を学習する際に覚えなければならない英単語（予約語など）の個数を知らないためにこのようなアンケート結果になったものと思われる。この点に関しては、テキスト型言語の学習の経験のある大学生に対して同じようなアンケートを取り更なる分析を試みたい。

6. まとめと今後の課題

本研究では、高校生向けに音楽を作成するという課題を扱うプログラミング授業を行い、アンケートによる評価を行い、ビジュアル型言語とテキスト型言語の間に位置づけられる特徴を有すること示した。

謝辞

本研究はJSPS 科研費 JP21K18535, JP20K03082, JP19H01721 の助成を受けたものです。また、本研究の一部は、早稲田理工研特別勘定 1010000175806 NTT 包括協定共同研究、および、経営情報学会「ICTと教育」研究部会の助成を受けたものです。本研究成果の一部は早稲田大学理工総研プロジェクト研究「次世代 e-learning に関する研究」の一環として行われたものです。

参考文献

- [1] M. Mladenović, I. Boljat, Ž. Žanko, “Comparing loops misconceptions in block-based and text-based programming languages at the K-12 level,” *Education and Information Technologies* 23(4), pp. 1483-1500, 2018.
- [2] R. Daskalov, G. Pashev, S. Gaftandzhieva, “Hybrid Visual Programming Language Environment for Programming Training,” *TEM Journal*. Volume 10, Issue 2, pp. 981-986, May 2021.
- [3] D. Weintrop, U. Wilensky, “Between a Block and a Typeface: Designing and Evaluating Hybrid Programming Environments,” *IDC '17: Proceedings of the 2017 Conference on Interaction Design and Children*, pp. 183-192, June 2017.
- [4] JSFiddle: <https://jsfiddle.net/>