プログラミング導入教育における
ビジュアルプログラミング環境を用いた教材の検討

占部 弘治

Email: k.urabe@niihama-nct.ac.jp

***1: 新居浜工業高等専門学校電子制御工学科**

◎ **Key Words**　ビジュアルプログラミング環境　インタラクティブ教材　プログラミング導入教育

1　はじめに

2020 年より小学校、2021 年より中学校でプログラミング教育が必修化されたとはいえ、実践的なプログラム作成能力を育成する必要のある工業高等専門学校ではプログラミングを基礎から改めて教えることは重要と考えられる。そこで新居浜高専電子制御工学科では 1 年生に対して C++を学ぶ講義科目と講義で学んだ知識で実際にプログラムを作成する演習科目を実施してる [(1)]。しかしながら、講義科目において実際にプログラムの作成と実行の確認をする機会がなく、理解と定着を確認することが困難であった。

そこで容易にプログラムの作成･実行ができるビジュアルプログラミング環境に注目した。ビジュアルプログラミング環境とはグラフィカルなインターフェイスを活用し、ブロックやオブジェクトで構成された部品を配置したり、矢印で接続することでプログラミングを行うことのできる環境のことである。この環境についてはマサチューセッツ工科大学のメディアラボが開発した Scratch[(2)] や National Instruments Corporation 開発の LabVIEW[(3)] などがよく知られている。今回は講義に合わせた環境が必要であるため、カスタマイズ可能であり、Web ページ上で動作可能な Blockly[(4)] を用いて開発することにした。Blockly は Google が開発したブロック型の JavaScript で記述されたビジュアルプログラミング環境である。すでに Blockly を用いた C 言語のビジュアルプログラミング環境については佐野･香川によって報告 [(5)] がなされているが、今回は本校の講義に合わせて機能を変更できること、および C++ の記述方法と一致させることを理由に独自に教材の検討を行い、開発を行った。

本稿では、今回開発した教材の検討および試用について述べ、試用の際に行ったアンケートの結果と考察について述べる。

2　教材の検討

今回のビジュアルプログラミング環境の開発について行った検討を述べる。教材を用いるのはプログラミング初学者が受講する C++ の講義とし、C++の構文を理解の補助を目的にする。

教材を用いることにした講義では C++ の文法を元にプログラムの構成や作成方法の説明を行い、サンプルプログラムの解説を行っている。しかし、実際に C++ のプログラムを入力し、実行することはこの講義科目とは別の時限で実施されている演習科目で行っている。講義科目と演習科目を分離することでプログラミングに関わる時間が増加し、演習問題の数を増やすことができた。このとこは 2023 PC Conference で報告をしいてる [(1)]。この演習科目では C++ のプログラムの入力はエディタを用い、コンパイルは Microsoft Visual Studio のコマンド `cl` を用いて行っている。この講義科目および演習科目を受講する学生は高等専門学校に入学したばかりの 1 年生であるため、キーボードを用いた入力にも不慣れなものも多い。よってプログラムの作成およびコンパイル・実行にかかる時間も学生によって大きく異なっている。このことにより講義においてプログラムの動作を実感するためにプログラミングの演習問題を示しても、それを解くには十分な時間をとることが難しいと考えた。

以前、開発した 似たような教材として HTML のドロップダウンメニューを用いたインタラクティブ教材がある [(6)]。この教材の例を図 1 に示す。

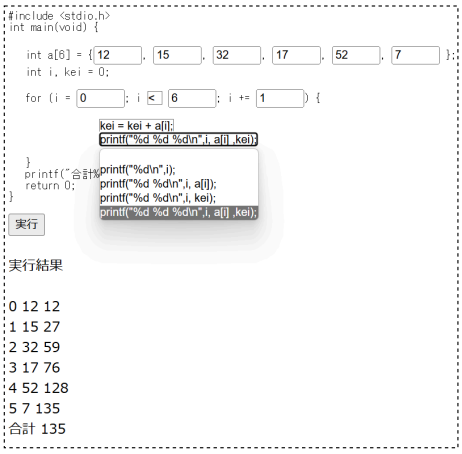

図 1: ドロップダウンメニューを用いた教材の例

この教材は JavaScript を用いて疑似的に実行結果を表示するものであった。しかしながら、教材の作成および実行の確認は手動で行う必要があったため、多くの教材が作成できず、動作は確認できるもののプログラムを作成した体験には繋がらないものであった。

以上より、講義の途中の短い時間で実行ができ、プログラミング体験が簡易にできる教材の必要性を感じた。そこでビジュアルプログラミング環境を用いることを考え、カスタマイズが自在である Blockly を使って独自の教材を開発することにした。

多くの Blockly を用いた環境が自然言語に近い形でプログラムの制御構文を表現しているが、今回開発する環境では C++ の構文に慣れること、プログラミングの体験を実感できることを理由に C++ の表現をそのままブロックの表示とした。

講義と連動して教材を使うため、クリック 1 回で説明に用いたサンプルプログラムを用意する機能を搭載することにする。また、この教材で作成したプログラムの再利用のために作成したブロックプログラムを XML 形式のファイルへ保存する機能、保存した XML ファイルを読み込んでブロックプログラムを再構成する機能を搭載するした。

作成したブロックプログラムを C++ へ変換する機能も搭載することにした。

3 開発教材

前節で述べた C++ のプログラミングをビジュアル環境で行うことのできる教材を作成した。この教材の全体図を図 2 に示す。

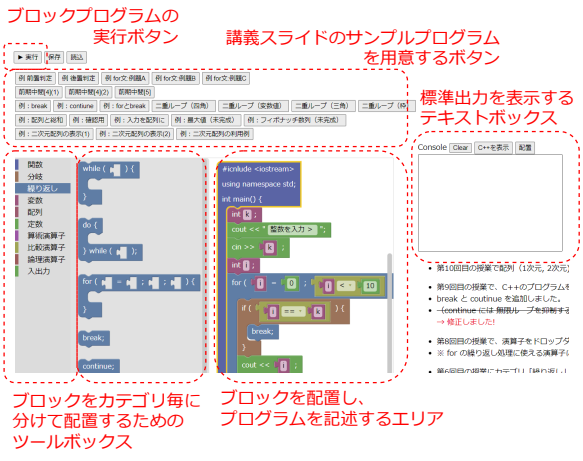

図 2: 開発したビジュアルプログラミング環境

左下のブロックを配置し、プログラムを記述するエリアは Blockly を用いて構成し、C++ の制御構文や変数の宣言、演算子などのブロックは Blockly Developer Tools を用いて独自に作成した。作成された独自ブロックはプログラム記述エリアの左側のツールボックスにカテゴリ別に配置され、自在に選ぶことが可能である。

演算子ブロックはドロップダウンメニューを用いて演算子を変更できるようにした。これによって演算子ブロックがブロックプログラムに組み込まれても容易に演算子を変更できるようになった。また、ブロックを選択するツールボックスでおいては全ての演算子についてドロップダウンメニューが選択された状態にした演算子ブロックを用意した。これは演算子をブロックに組み込む際に演算子を選択する手順を省くためである。

左上最上部には作成したブロックプログラムを実行する「実行」ボタン、作成したブロックプログラムを XML 形式のファイルへ保存する「保存」ボタン、保存した XML ファイルを読み込み、ブロックプログラムを再構成する「読込」ボタンを設置している。これらのボタンの下に、クリックすれば講義で用いたスライドのサンプルプログラムをブロックプログラムで用意するボタンを配置する。

プログラムの実行による標準出力はプログラミング環境の右側に用意したテキストボックスに表示することにした。このテキストボックスの上に「Clear」「C++を表示」のボタンを配置した。「Clear」ボタンをクリックすると、テキストボックスに表示された文字列が消去される。「C++を表示」ボタンをクリックすると、ブロックプログラムを C++ のプログラムに変換し、これがテキストボックスに表示される。

今回作成したビジュアルプログラミング環境教材は HTML と JavaScript のみで構成されているので、本校で採用している学習マネジメントシステムの Web-Class には資料として公開することが可能である。

4 講義における試用

4.1 試用状況

新居浜高専電子制御工学科 1 年のプログラミング導入の専門科目の講義「情報処理 1」においてこのビジュアルプログラミング環境の試用を行った。表 1 にこの「情報処理 1」の令和 6 年度の講義計画を示す。

第 4 回目の「分岐を行うための構文（1）」の講義から今回作成したビジュアルプログラミング環境を学生に公開した。さらにこの回以降は講義の進捗に合わせて扱うことのできるブロックを増やした。

第 6 回の「繰り返しを行うための構文（1）」の回からクリックすると講義の解説に用いたスライドに掲載されたサンプルプログラムをビジュアルプログラミング環境に用意するボタンを追加した。講義の回が進む

とサンプルプログラムの数が増えるので、ここのボタンも増えることなった。

表 1: 令和 6 年度の「情報処理 1」の講義計画

回	講義内容
1	コンピュータの基本構成とソフトウェア
2	コンソールの入出力と書式設定
3	データ型と算術演算子
4	分岐を行うための構文（1）
5	分岐を行うための構文（2）
6	繰り返しを行うための構文（1）
7	中間試験
8	中間試験返却・復習
9	繰り返しを行うための構文（2）
10	配列、配列と繰り返し
11	二次元配列
12	関数（1）
13	関数（2）
14	ローカル変数とグローバル変数
15	期末試験
16	期末試験返却・復習

このビジュアルプログラミング環境を学生が活用するとどのような効果が得られるかを調べるために、第 11 回の講義では演習とアンケートを実施した。

図 3: 配列の要素を全て表示するブロックプログラム

演習には不完全なブロックプログラムを与え、これにブロックを追加することでプログラムを完成させる問題を 4 問用意した。例えば、図 3 に示すような配列の全ての要素を表示するプログラムを与え、白の矢印の箇所にブロックを追加することで全ての要素の中で最も大きい整数が求めるプログラムを作成する問題である。

4.2 アンケート結果と考察

この演習を実施した 11 回目の講義において作成したビジュアルプログラミング環境の評価を調べるためアンケートを行った。アンケートの回答者はこの講義を受講している電子制御工学科の 1 年生のうち欠席者を除いた 41 名である。

このアンケート結果を図 4 〜図 7 に示す。

図 4 の結果から、あらかじめ公開していたビジュアルプログラミング環境をほとんどの学生が演習を行う講義より前に使っていることが分かった。

Q. この授業で紹介したビジュアルプログラミング環境が 4 回目の授業から紹介されていましたが、これを試してみましたか？

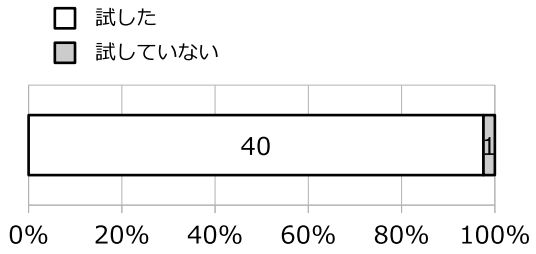

図 4: アンケート結果 (1)

Q. このビジュアルプログラミング環境に第 9 回の授業から C++ のプログラムが表示できるようになりました。この機能を使って C++ のプログラムをテキストエディタにコピーして保存し、実際に Visual C++ コンパイル・実行をしたことがありますか

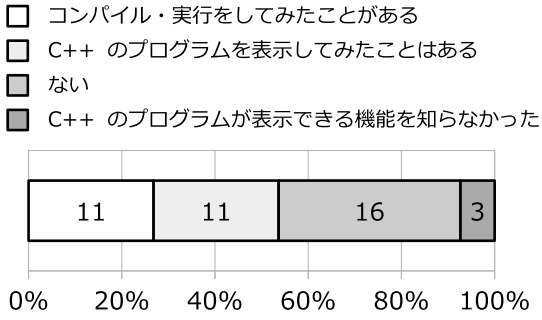

図 5: アンケート結果 (2)

また、図 5 の結果より、C++のプログラムを表示

できることは学生のほとんどが知っていたが、実際に表示した学生はほぼ半数で、これをエディタを使って保存し、実際にコンパイラを使って実行した学生はほぼ 1/4 であった。

これらのことより、環境の存在と利用については学生に周知することができたが、これを用いてコンパイラで実行できることを試してみようと思う学生が多くなかったと考えられる。

図 7 の結果から 2/3 の学生がこの環境を使いやすいと感じているものの、1/3 の学生はそう感じていないことがわかった。また、図 6 の結果はプログラムをエディタで作成し、コンパイルで実行する方法と比較した場合を示しているが、1/2 近くの学生がこの環境よりもコンパイルでの実行のほうが解きやすいと考えていることがわかった。この環境はプログラミングの理解を促すことが目的であるためエディタとコンパイルでプログラムで実行すること同等以上の使いやすさが必要であると考える。よって、この環境の利用を促すためには不便と感じる箇所を調査し、改良する必要があると考える。

5 まとめ

本稿ではプログラミング導入のための講義において学生が講義の合間や復習のために容易にプログラムの動作を確認できるビジュアルプログラミング環境を構築したことを報告した。また、この環境を実際の講義で学生に利用させたときの使用感に関するアンケートについての報告を行った。アンケートの結果から、多くの学生が使いやすと回答したが、そう思っていない学生も少なくないため今後の改良が必要である。

しかしながら、理解に対する効果の調査を行うことができていない。これは今後の課題と考える。

この学年では Visual Studio などの総合開発環境や Visual Studio Code のようなソースコード作成に特化したエディタの利用を行っていないため、これらの環境との比較が行うことができなかった。この調査も必要と考える。

参考文献

(1) 占部, 永井, 眞鍋: “プログラミング導入教育における演習拡充の実践と効果”, 2023 PC Conference 論文集, pp.199-pp.202 (2023)

(2) “Scratch - Imagine, Program, Share -”, https://scratch.mit.edu/

(3) “NI LabVIEW とはテストと測定のためのグラフィカルプログラミング”, https://www.ni.com/ja/shop/labview.html

(4) “Blockly | Google for Developers”, https://developers.google.com/blockly/

(5) Y.Sano, K.Kagawa: “Design of a Programming Environment for Non-Procedural Programming Languages using Blockly”, The International Journal of E-Learning and Educational Technologies in the Digital Media (IJEETDM) Volume 5 Issue 3, pp. 93-101 (2019)

(6) 占部: “プログラミング科目の理解を助けるインタラクティブ教材の検討”, 2016 PC Conference 論文集, pp.63-pp.64 (2016)

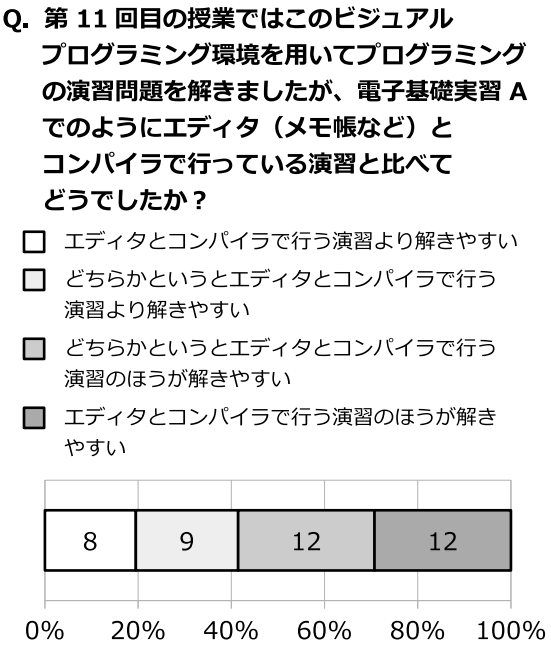

図 6: アンケート結果 (3)

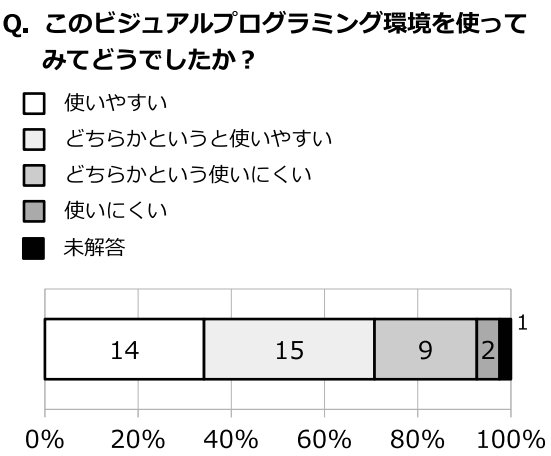

図 7: アンケート結果 (4)