Vol.2021-CE-162 No.20
Vol.2021-CLE-35 No.20
2021/12/5

アルゴリズム教育におけるブロックプログラミング言語の利用

内田保雄[1]　玉城龍洋[2]　大西淳[3]

概要：初学者向けのプログラミング教育コースにおいてブロックプログラミング言語が広く使われるようになってきている．しかしながら，その後の本格的なテキスト型プログラミング言語への学習接続は必ずしも円滑とはいえない．そこで，アルゴリズム教育においてブロックプログラミング言語 EduBlocks を併用することによりテキスト型ブログミングへの円滑な移行を意図した学習メソッドを提案する．本稿では，テキスト型プログラミング言語を学ぶ前段階でのアルゴリズム教育における，ブロックプログラミング言語を利用した理解促進の試みの評価と今後の活用法について議論する．

キーワード：アルゴリズム教育，ブロックプログラミング，EduBlocks

Utilization of Block-Based Programming in Algorithm Education

YASUO UCHIDA[†1]　TATSUHIRO TAMAKI[†2]
ATSUSHI ONISHI[†3]

1. はじめに

初学者向けのプログラミング教育コースにおいてブロックプログラミング言語が広く使われるようになってきている．たとえば「小学校を中心としたプログラミング教育ポータル」サイトに報告されたデータから集計されたプログラミング教育ツールの利用調査[1]によれば，2021 年 4 月 13 日の時点で,最もよく使われていたのが代表的なブロックプログラミング言語の Scratch[2]であり，37.7%を占めていた．また，高校でも Scratch が 27%の割合で教えられているという調査結果もある[3].

しかしながら，その後の本格的なテキスト型プログラミング言語への学習接続は必ずしも円滑とはいえないことが指摘されている[4].

そこで，アルゴリズム教育においてブロックプログラミング言語を併用することによりテキスト型ブログミングへの円滑な移行を意図した学習メソッドを提案する．

本研究では，アルゴリズムの入門教育として，問題の分析から始めてフローチャートの作成とトレースを中心とした学習にブロックプログラミング言語である EduBlocks[5]を併用することにより，概念の理解と処理の表現技術の効果的な習得を試みた．

2. 先行研究

プログラミング教育とりわけアルゴリズム教育における学習補助手段としては，フローチャートやトレースなどがよく使われている．それらを活用しながら，最終的には何らかのプログラミング言語を用いて記述・実装する訳であるが，初学者にとっては，その作成過程は必ずしも容易ではないとされている[6].

そこで先ず，ブロックプログラミング言語を含むビジュアル型言語とテキスト型言語との関係について概観する．

プログラミング初学者を対象としたプログラミング教育支援システム「BlockEditor」の提案が試みられた[7]．ビジュアル型プログラミング言語から C や Java などのテキスト記述型言語へのシームレスな移行が考慮されていないという問題意識から提案され，プログラミングの学習が進行するにつれて, BlockEditor から Java へ徐々に移行していくこと，およびそのタイミングには個人差があることが定量的に示された．また，プログラミングに苦手意識を持つ学生ほどビジュアル型言語の選択率が高く，言語の相互変換環境が言語の交ぜ書きを促進し，Java 言語習得の足場かけとなることが示されており，ブロック型言語の有効性の一端が明らかにされている．

ビジュアル型言語とテキスト型言語の間には大きな壁があるという視点から取り組んだ研究がある[4]．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方法を確立することは重要とした上で，ビジュアル型言語とテキスト型言語のそれぞれの学習過程の違いについて脳波を計測することによって明らかにするという独特の研究である．具体的には，ビジュアル型言語とテキスト型言語の学習について難易度の異なる問題を解く実験を行い，その際の脳波を計測し，その結果，ビジュアル型言語とテキスト型言語の学習過程には異なる思考が行われている可能性が示唆されたと結んでいる．ただし，その得失を明らかにするところまでは至っていない．

1 宮崎産業経営大学
Miyazaki Sangyo-keiei University
2 沖縄工業高等専門学校
National Institute of Technology, Okinawa College
3 津山工業高等専門学校
National Institute of Technology, Tsuyama College

Vol.2021-CE-162 No.20
Vol.2021-CLE-35 No.20
2021/12/5

ビジュアル型言語からテキスト型言語への橋渡し教材の開発を行った研究がある[8]．まず，Scratch や Google Blockly[9]のようなビジュアル型プログラミング言語は，初心者にとってプログラミング的思考を学修するために適切な言語である， と指摘している．そして，ブロック型言語の機能の制限や限界について言及している．最終的には，テキスト型のプログラミング言語として十進 BASIC を選択し，ビジュアル型のプログラミングの形態を模写する形で，ビジュアル型の言語からテキスト型の言語への橋渡しとなる教材の開発を試みることにより，テキスト型言語が，発展的な学習としての位置づけとなり得ることを示しており，言語学習の接続性について参考になるものである．

高校情報科レベルにおけるプログラミングの授業設計の指針を得ることを目的として，ビジュアル型とテキスト型のプログラミング言語の学習順序が与える教育的効果について検証を行った研究がある[10]．ビジュアル型とテキスト型の学習順序によって，テキスト型学習への情意面に有意な差は見られなかった，との結果を得ている．一方，事前・事後テストでは，ビジュアル型先行群が全ての設問において肯定的な回答が見られた．この結果から，テキスト型と同等な学習内容のビジュアル型を事前に行うことが効果的な点と知的技能を中心とした課題の重要性が示唆された，と報告している．中等教育段階でもビジュアル型のプログラミング言語の存在意義が示されたといえる．

3. 提案手法

(1) 本研究の背景

本研究の対象となる，筆者が担当する科目の概要について述べる．所属大学は社会科学系の大学であり，対象科目は経営学部の初年次生を対象とした選択科目である．科目名は「アルゴリズム I 」であり，前期 15 回の科目である．カリキュラムの上の位置づけとしては，2 年次に履修するプログラミング科目に先立ちアルゴリズム(とデータ構造)の入門を学ぶことである．そのため，本格的なプログラミングを用いないことが想定されており，主にフローチャートやトレースにより学習を進めるものである．本科目より前にコンピュータの基礎を学ぶ科目は配置されておらず，プログラミングを学ぶ前提知識を設定せずにスタートすることになる．したがって，学習を効果的に進める上では，フローチャート・シミュレータのようなツールが望まれるが一般的ではないため，主に手書きによる学習形態となっている．そのため，最大の問題点は作成したフローチャートのアルゴリズムをプログラミング言語として記述・実装したときに正しく動作するかどうか判定できず中途半端な理解で終わってしまう可能性があることである．先にも述べたように，本格的なプログラミング言語の利用は想定していないため，正確な問題解決への障壁となっている．

(2) EduBlocks

そこで，本研究では導入難易度が低く学習コストが少ないブロックプログラミング言語を併用することとした．代表的なブロックプログラミング言語には Scratch , MakeCode[11], Blockly などがあるが, 本研究では EduBlocks を用いることとした．その理由は，Python 言語[12]への変換機能を有するためテキスト型言語への接続性が高いことである．

EduBlocks はジョシュア・ロウが主宰する EduBlocks プロジェクトにより開発されているビジュアルブロックベースのプログラミングツールである．ブラウザ上でブロックを組み合わせて記述・実行でき，その名のとおり教育に適した設計となっている．EduBlocks は，他のビジュアル言語と同様に初心者にもわかりやすいツールでありながら内部で Python 言語が動作するため, 将来の発展性や拡張性にも期待できるという特徴がある．また，グラフ表示や Web スクレイピングなどにも対応できる機能も備えている．その実力は, 平成 30 年告示高等学校学習指導要領に対応した令和 7 年度大学入学共通テストからの出題教科・科目のサンプル問題『情報』の一部を容易にプログラムとして実装できることでも確認できる（図 1）.

図 1 サンプル問題『情報』の実装例

したがって，アルゴリズム学習の入門レベルに十分対応できかつテキスト型言語への親和性も高いといえる．

Vol.2021-CE-162 No.20
Vol.2021-CLE-35 No.20
2021/12/5

EduBlocks の Python3 モードについて概要を説明する．機能的には Python3 のコア部分が実装されており，さらに直接 Python コードを埋め込むブロックを用いることでより応用的な利用も可能となっている．ユーザーインタフェースは日本語対応ではなく英語表記であるが，テキスト型言語への接続性を考えると必ずしも障壁ではなく，むしろ親和性が高いと言える（図 2）．

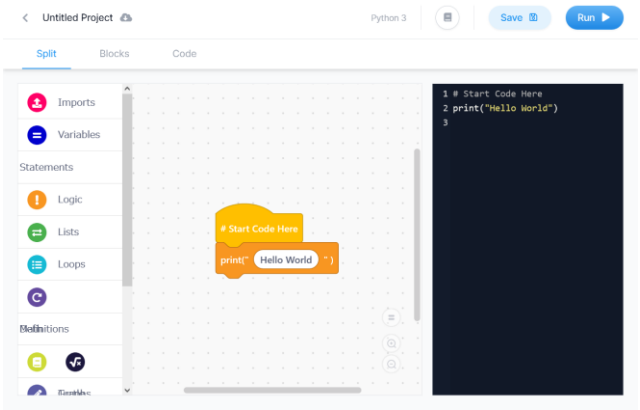

図 2　EduBlocks の Python3 モード

ブロックを組み合わせて作成したプログラムはそのまま Python 言語のコードに対応しており，ブロックを配置すると即座に Python のソースコードが生成される仕組みとなっている．また，ブロックと Python コードを並べて表示することもできる．ただし，以前のバージョンではコードからブロックへの変換対応機能もあったが，現バージョンではブロックからコードへの 1 方向の変換となっている．なお，EduBlocks プロジェクト・ファイルに加え Python コードのダウンロードも可能となっている．

EduBlocks は Scratch のようなキャラクタを利用したゲーム性のあるプログラムの作成用途には向いていないが，アルゴリズム教育のような知的技能の醸成を目指した用途には十分な効果が期待できるものである．

(3) 提案手法による学習の流れ

本授業科目における学習の流れは以下のとおりである．

① 例題や練習問題を分析し，問題解決手順を理解する．
② フローチャートを作成する．なお，本授業ではフローチャートの記述は手書きで行っている．
③ 必要に応じてトレース作業を行う．
④ フローチャートから EduBlocks へ対応付けてブロックを配置してプログラムを作成する．
⑤ EduBlocks のプログラムを実行して結果を確認してフローチャートの的確さ，さらにはアルゴリズムの正しさのチェックを行い，必要に応じてデバッグ作業に取り組む．

フローチャートを作成すると，図 3 のようにフローチャートの記号と EduBlocks のブロックとをほぼ一対一で対応させて作成できることがわかる．

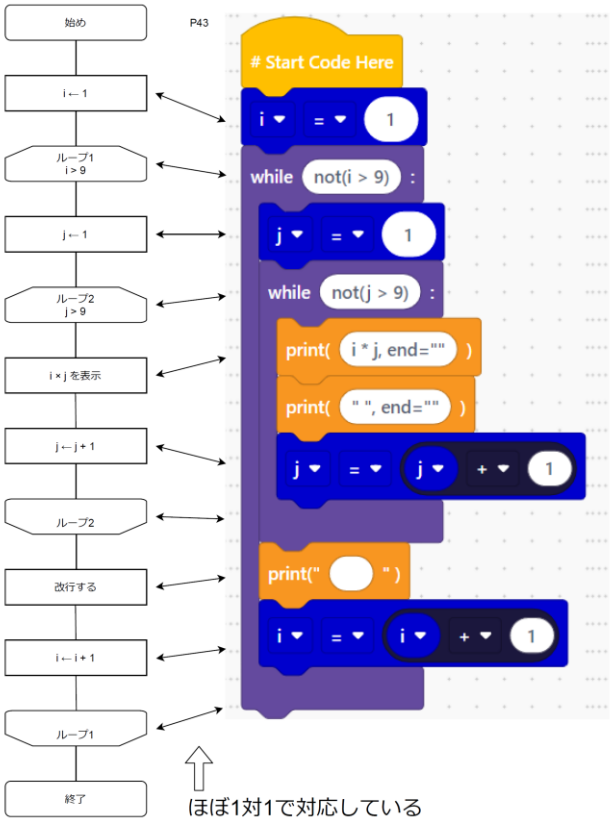

図 3　流れ図と EduBlocks の対応

また、トレースの例を図 4 に示す。

流れ図	実行される命令	命令実行後の内容 変数x	命令実行後の内容 変数i	終了条件 i > 10	繰返し条件 i <= 10
開始	なし	不定	不定		
0 → x	x = 0	0	不定		
加算 i=1,2,…,10	for i in range(1, 10 + 1):	0	1	No	Yes
x + i → x	x = x + i	1	1		
加算 i=1,2,…,10	for i in range(1, 10 + 1):				
x + i → x	x = x + i				
加算 i=1,2,…,10	for i in range(1, 10 + 1):				
x + i → x	x = x + i				

図 4　トレースの例

4. 実践授業

4.1 シラバス

本授業科目は従来から開講されており，内容は概ね従来のものを踏襲した．授業のシラバスは表 1 のようになっている．ただし，本研究対象となる授業自体は筆者が今回初めて担当して実施したものである．

表 1　授業のシラバス（修正版）

授業回	内容
1	ガイダンス
2	アルゴリズムとは，アルゴリズムとデータ構造

Vol.2021-CE-162 No.20
Vol.2021-CLE-35 No.20
2021/12/5

3	アルゴリズムと流れ図
4	プログラミング言語への実装
5	プログラム流れ図
6	プログラム流れ図の基礎
7	プログラム論理の構造化
8	変数と代入文
9	条件分岐
10	繰返し処理
11	2 重ループ
12	文字列処理
13	配列処理の基礎
14	多次元配列処理
15	まとめ

なお，教科書は『基本情報技術者 大滝みや子先生のかんたんアルゴリズム解法 ～流れ図と擬似言語～ 第 4 版』，大滝みや子，リックテレコム刊を使用している．

4.2 実践の概要

授業は，2021 年度の 4 月から 7 月に行われ，1 回は 90 分（1 コマ）で 15 回を行う 2 単位授業 15 週で実施した．受講者は，経営学部経営学科の 33 名であった．ティーチングアシスタントの配置はなく教員 1 名で担当している．今年度も COVID-19 の影響により，ハイフレックス型授業として実施し，対面授業での受講者が 3 割程度で残りは Zoom によるオンライン受講となった．

本科目の目的は，問題の定義，分析または解法の図的表現であるフローチャートの利用を通してプログラミングに必要な論理的思考力を身につけることであり，そのために手書きによるフローチャートの記述にこだわった．そこで，成績評価対象の一部である学習ポートフォリオ作成においては必ず手書きのフローチャートを添付するように義務付けた．手書きのフローチャート例を図 5 に示す．

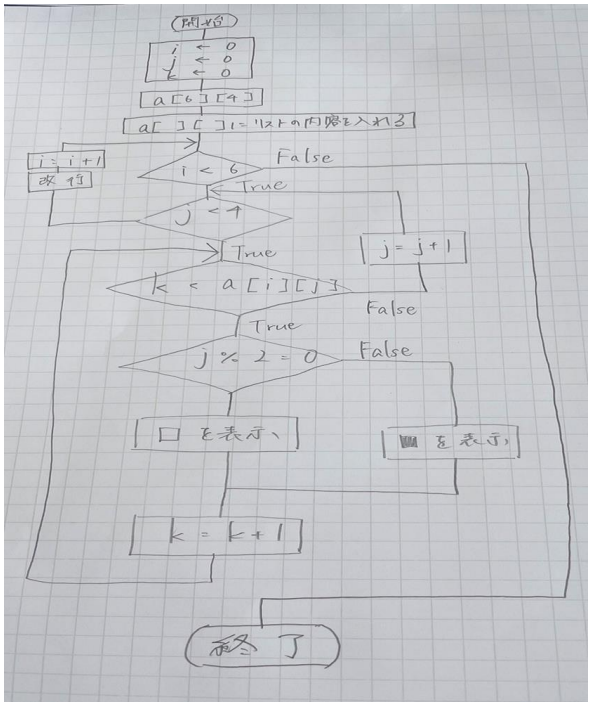

図 5 手書きのフローチャートの記述例

今回の実践では，授業回ごとに初めに教員がスライド等を用いて概要を説明した後，学生が各自でフローチャートの記述やトレース作業を行いその後 EduBlocks でプログラムを作成するという従来型の授業進行で実施した．ただし，オンライン受講者が多く，質問などが難しい状況を勘案して，ポイントをまとめた短いビデオ教材を用意して，復習や自学自習ができるように配慮した．

今回は，成績評価において COVID-19 の影響を考慮して，ペーパーテストではなくレポート試験を実施した．レポート試験の概要は以下のとおりである．指定課題と自由課題の 2 つの課題を設定した．

＜指定課題＞

課題名：色を数で表す （CS アンプラグド）

課題：CS Unplugged（コンピュータサイエンス・アンプラグド）の Activity 2（学習２）: 色を数で表す （画像表現）をシミュレート（模倣）するフローチャートおよびプログラム（EduBlocks 4.0 BETA）を作成する．

実行例（データと出力）：

[[1, 3, 1],	□■■■□
[4, 1],	□□□□■
[1, 4],	□■■■■
[0, 1, 3, 1],	■□□□■
[0, 1, 3, 1],	■□□□■
[1, 4]]	□■■■■

＜自由課題＞

課題名：(各自で設定)

課題: (今までに学習した内容を反映させた課題を各自で考案する)

技法：条件判断（分岐），繰り返し（１重ループ，多重ループ），リスト（1 次元配列，多次元配列）

5. 実践授業の評価

5.1 アンケート

アンケートは LMS のアンケート機能を用いて最終の授業回に実施し，16 件の有効回答を得た．

以下にアンケート項目およびアンケート結果を示す．% 表示があるものは百分率で示しており，数値のみのものは人数である．

(1) アルゴリズムの理解度について教えてください．

よく理解できた（0%） だいたい理解できた（25%）
普通（31%） あまり理解できなかった（25%）
ほとんど理解できなかった（19%）

(2) フローチャートの作成（学習）は，アルゴリズムの学習に役立ちましたか？

そう思う（38%） ややそう思う（44%）
どちらでもない（0%） あまり思わない（0%）

思わない（19%）
(3) EduBlocks によるプログラム化は，アルゴリズムの学習に役立ちましたか？
そう思う（31%） ややそう思う（44%）
どちらでもない（0%） あまり思わない（6%）
思わない（19%）
(4) アルゴリズムの学習で難しいと感じる点は何ですか？（複数回答可）
特にない（1） 問題の意図がわからないこと（6）
問題から処理手順を考えること（12）
フローチャートを書く（構成する）こと（10）
順次・分岐（比較）・繰り返し（反復），変数やリスト（配列）などの概念（7）
自分の書いたフローチャート（の処理）が正しいかどうかわからないこと（9）
その他（0）
(5) 今回の授業の中で理解に役立った教材（または学習方法）は何ですか？（複数回答可）
特にない（2） 教科書（6） 配布資料（プリント）（4）
Moodle 内のスライド（6） Moodle 内の説明ビデオ（5）
教員の口頭による解説（5）
自分で探したネット上の資料（7）
テーブルトレース（1） 作問学習（3） その他（2）
(6) 高校での学科を教えてください．
普通科（13%） 総合学科（0%） 商業系の学科（商業科，経営情報学科，くらしの科学科など）（81%）
工業系の学科（情報技術科，システム工学科，電子情報科など）（6%） その他（0%）
(7) 過去にフローチャートを学習したことがありますか？（複数回答可）
ない（5） 高校の授業で（8） 中学校の授業で（3）
高校のクラブ等で（1） 中学校のクラブ等で（0）
学習塾で（0） オープンキャンパス等のイベントで（0）
個人的に（0） その他（0）
(8) 過去にプログラムを作成したことがありますか？（複数回答可）
ない（6） 高校の授業で（7） 中学校の授業で（2）
高校のクラブ等で（0） 中学校のクラブ等で（0）
学習塾で（0） オープンキャンパス等のイベントで（1）
個人的に（1） その他（0）
(9) (8)で「ない」以外を選択したひとだけ回答してください．使用したことのあるプログラミング言語をすべて答えてください．（複数回答可）
Scratch（1） ビスケット（1） micro:bit（2）
HTML/CSS（3） JavaScript（2） Python（2）
Visual Basic（1） C 言語（2） Java（1） マクロ言語（0）
SQL（2） その他（2）
(10) この授業に対する意見・提案・感想があれば教えてください．

- レポートが自分にとっては難しかったです．授業はほとんど出席したのですが，難しかったです．
- 問題を自分で考えるといっても全くわからないことがあります．なので，先に解説したほうがいいと思います．
- あまり理解できていないままどんどん先に進んでしまったので，余計にわからなくなってしまうことが多かった．
- 少々授業の進行が速いように思った．また，オンライン受講の際に生じるトラブルが多かったように思うので，そこが改善されるとありがたく思う．
- 授業内容は比較的分かりやすかったのですが，時折内容が複雑になり分からない問題もありました．これまで培ってきた知識を今後も色々な場面で今まで習ったことを生かしていきたいと思います．
- 授業中の画面が映っていなかったりなどはすぐにわかってほしい機材トラブル等はなくすことや起こったとしてもすぐに対応できるようにしてほしい．時間配分をしっかりとして計画的に授業を進めてほしい．課題等はとても面白かったです．
- 何も高校で習ってない意見としては少し難しく感じた．しかし，学ぶのは楽しかったです．
- 高校の授業で商業科に所属していたが，プログラミングの授業がなかったので理解することが困難だった．
- 流れ図を作る際の説明（どのような内容の流れ図を作るのか）をもう少し詳しく説明して欲しいです．
- 私はプログラミングの検定を持っているため簡単な流れ図の作成などはできたけど，どんどん難しくなってわからないものが増えた．普通科から来た人や，商業科だったけどプログラミングを習わなくて，大学で初めて学んだ人にとってはすごく難しかったと思う．
- Edublocks があまりにも難しいので流れ図だけにしてほしい．教員の説明はわかりやすかった．
- フローチャートやプログラムを作成したり，問題を考えたりするのが難しかったです．

6. 議論

まず，アルゴリズムの理解度についての質問に対する回答では，「あまり理解できなかった」「ほとんど理解できなかった」を合わせると 44%となった．その理由についての記述式回答の主なものは，授業進度が途中から速くなったと感じたことやレポート試験を難しく感じたことなどであった．

フローチャートの作成（学習）はアルゴリズムの学習に役立つかという質問では，82%が肯定的な回答を示していた．ただし，記述式回答からはフローチャートの有用性を認めながらも自分で作成するとなると難しいとの印象がう

かがえた.

EduBlocks によるプログラム化はアルゴリズムの学習に役立ったかという質問では, 75%が肯定的な回答を示した. しかしながら，たまたま授業期間の中ごろに EduBlocks 自体の大幅なバージョンアップの変更があったため，戸惑った学生も見受けられた.

アルゴリズムの学習で難しいと感じる点は何か（複数回答可）という質問に対する回答（人数）で最も多かったのが, 問題から処理手順を考えること（12 人）であり, 以下, フローチャートを書く（構成する）こと（10 人），自分の書いたフローチャート（の処理）が正しいかどうかわからないこと（9 人）と続いている.「問題から処理手順を考えること」はアルゴリズム学習の本質につながるものであり, 例題や練習問題だけでは十分でないと判断し，作問学習の要素も取り入れたが学生の習熟度には開きが出たようである.「自分の書いたフローチャート（の処理）が正しいかどうかわからないこと」という点は，まさに本研究の主眼とするところであり，EduBlocks による学習支援の可能性も示唆されていると考えられる.

理解に役立った教材（または学習方法）は何か（複数回答可）という質問に対する回答（人数）で最も多かったのが,「自分で探したネット上の資料」(7 人）で，僅差で「教科書（6 人)」，Moodle 内のスライド（6 人）などが続いている．やはり今どきの学生らしくネットの情報を活用しているようであるが，一方で筆者がネット上で提供したビデオ教材はあまり視聴されていなかった. その原因としては, ネット回線の安定性や速度が十分でない学生が少なからずいること，教材提供のタイミングが少し遅かったことなどが考えられる.

高校での所属学科については，アンケートに回答した学生の実に 81%が商業系の学科と答えた．もともと本学の経営学科の入学生に占める高校での出身学科に占める商業系の割合は高いのであるが，100 人超の新入生の中でも商業系の出身の学生が多く本科目を選択していたようである.

フローチャートの学習経験に関する質問（複数回答可）では，高校の授業で（8 人），ない（5 人）などとなっているので前提知識に差があったことが少なからず習熟度に影響を与えていると考えられる.

プログラミングの作成経験に関する質問（複数回答可）でも，高校の授業で（7），ない（6）などとなっているのでフローチャートの場合と同様の結果となっている.

使用したプログラミング言語に関する質問では，突出したものはなく Scratch も一人のみであった. しかしながら, 今後はプログラミング経験者も徐々に増えてくることになる.

最後の質問，授業に対する意見・提案・感想に対する回答では，初期段階はおおむね理解ができていたが，条件分岐や繰り返し処理を組み合わせる段階になると進度が速くなって急に難しくなったと感じたという感想が多くあった.

7. まとめ

初学者向けのアルゴリズム教育科目において，フローチャート作成学習を補完する試みとしてブロックプログラミング言語を併用した授業を実践した．日本ではまだあまり利用されていない EduBlocks を使った初めての試みであり, 必ずしも我々の狙い通りの結果が得られたわけではない. 他の先行事例等でも示されているように今後はますますブロックプログラミング言語やビジュアル言語の活用が進展すると考えられる．そのときには，その後の本格的テキスト型プログミングへの円滑な移行を意図した学習メソッドに関する議論が活発化すると想定される.

今後は，フローチャートとブロックプログラミング言語との対応性を明確にした教材作成やテキスト型プログラミング言語との連携技法の構築などに取り組むことが課題となる.

謝辞 本研究は JSPS 科研費 19K03104 の助成を受けたものです.

参考文献

[1] “データ見るビスケット 2020 -ファクトシート公開 後編-”. https://devroom.viscuit.com/2021/06/07/post-2074/, (参照 2021-10-28).
[2] “Scratch”. https://scratch.mit.edu/, (参照 2021-10-28).
[3] “高等学校でのプログラミング教育に関するアンケート（中間報告）”. https://edu.monaca.io/archives/5151, (参照 2021-10-28).
[4] 梅澤克之，石田崇，中澤真，平澤茂一．ビジュアル型言語とテキスト型言語の学習状況の比較．信学技報, 2021, ET2020-72, p. 115-120.
[5] “EduBlocks”. https://edublocks.org/, (参照 2021-10-28).
[6] 胡石帆，野崎浩成．フローチャートと Scratch を利用したプログラミング授業の方法. 2019, 2019 年度 JSiSE 学生研究発表会・発表論文, p. 81-81.
[7] 松澤芳昭，酒井三四郎．ビジュアル型言語とテキスト記述型言語の併用によるプログラミング入門教育の試みと成果．研究報告コンピュータと教育（CE）, 2013, 2013-CE-119(2), p. 1-11.
[8] 中島敏．ビジュアル型言語からテキスト型言語への橋渡し教材の開発．群馬高専レビュー, 2019, (37), p. 65-74.
[9] “Blockly”. https://developers.google.com/blockly, (参照 2021-10-28).
[10] 岡本恭介，安藤明伸．ビジュアル型とテキスト型プログラミングにおける学習順序が教育的効果に与える影響．日本教育工学会論文誌, 2020, 44(Suppl.), p. 97-100.
[11] “Microsoft MakeCode”. https://makecode.microbit.org/, (参照 2021-10-28).
[12] “Welcome to Python.org”. https://www.python.org/, (参照 2021-10-28).