

## 相互評価機能を持ったブロック型プログラミング環境の設計と開発

Design and Development of Block Type Programming Environment  
Implemented Peer Assessment

福井 昌則<sup>\*1</sup>, 萩倉 丈<sup>\*2</sup>, 佐々木 雄司<sup>\*3</sup>, 黒田 昌克<sup>\*4</sup>, 森山 潤<sup>\*4</sup>, 平嶋 宗<sup>\*1</sup>  
Masanori FUKUI<sup>\*1</sup>, Jo HAGIKURA<sup>\*2</sup>, Yuji SASAKI<sup>\*3</sup>, Masakatsu KURODA<sup>\*4</sup>, Jun MORIYAMA<sup>\*4</sup>,  
Tsukasa HIRASHIMA<sup>\*1</sup>

<sup>\*1</sup> 広島大学

<sup>\*1</sup>Graduated School of Engineering, Hiroshima University

<sup>\*2</sup> 関西学院大学

<sup>\*2</sup>School of Science and Technology, Kwansei Gakuin University

<sup>\*3</sup> 慶応義塾大学

<sup>\*3</sup>Faculty of Environment and Information Studies, Keio University

<sup>\*4</sup> 兵庫教育大学

<sup>\*3</sup>Hyogo University of Teacher Education

Email: fukui@lel.hiroshima-u.ac.jp

あらまし：本研究の目的は、相互評価機能を持ったブロック型プログラミング環境の設計と開発を行うことである。プログラミング未経験者や入門者が抵抗なく取り組めるプログラミング環境として、Scratch などといったブロック型プログラミング環境がよく用いられている。このブロック型プログラミング環境を活用し相互評価を取り入れることで、多くの児童・生徒が相互評価を取り入れた協調学習を行うことが期待できる。そこで Google が提供しているライブラリである Blockly を用い、ユーザ同士が相互に評価を行えるブロック型プログラミング環境およびデータを介するサーバを構築した。このシステムを用いることで、プログラミング未経験者や入門者も相互評価を取り入れた協調学習が可能となる。

キーワード：プログラミング教育, Blockly, 相互評価, 協調学習, プログラミング的思考

## 1. はじめに

文部科学省「学びのイノベーション事業」では、ICT 活用や協働学習などを用い、従前行われていた教育をさらに深める実践が多数報告されており<sup>(1)</sup>、協調学習を取り入れ、従前の学習内容をより深く理解する活動を展開することは重要である。Laal, M. & Ghodsi, S. M.は、協調学習は、問題解決や課題解決を行うことに対し、教授方法としても学習方法としても有望なアプローチであると指摘している<sup>(2)</sup>。

Kollar, I. & Fischer, F.は、グループにおける相互評価は、協調学習を活性化するために最も効果的な方法の1つであるとしている<sup>(3)</sup>。つまり、相互評価を取り入れた協調学習を行うことにより、従前の学習内容をより深く理解させることが期待できる。

一方、2020年から義務教育課程においてプログラミング教育が必修化されることとなり、プログラミング教育においても、協調学習を取り入れることは重要な方略の一つであると考えられる。例えば、加藤らは、初学者が協調プログラミングを行える学習支援システム“Checo Pro”を開発し、Javaを用いた実践について報告している<sup>(4)</sup>。北らは、Peer Reviewを取り入れたプログラミングの授業支援ツールを開発し、そのツールを用いたC言語を用いた実践を通して、ツールの有用性について報告している<sup>(5)</sup>。しかし、これらを含めたプログラミングの学習で協調学習を行うことを扱った先行研究では、実際にプログラミング言語を用いて協調学習で行うことを目的と

して開発されているものがほとんどであり、義務教育段階におけるプログラミング教育で身につけることが期待されている「プログラミング的思考」を育成しうるものであるかどうかは定かではない。義務教育段階においては、Scratch<sup>(6)</sup>などのブロック型プログラミング環境がよく用いられており、プログラミング的思考の育成のためにブロックを組み上げていく、ブロック型プログラミング環境の活用を行っていくことで、いわゆるテキストベースのコーディングを必須としないプログラミング教育を推進することが可能となる。また、単にブロック型プログラミング環境を活用するだけでなく、相互評価を取り入れた協調学習を行うことで、従前のプログラミング教育より、深い学習を行えるようにすることが求められる。ブロック型プログラミング環境では、命令がブロック化され、一つの単位として可視化されている。そして、違ったブロック(リソース)を提供することで、同じ結果を得る場合であっても、違った組み立てをすることとなる。平嶋らは、異なるリソースに基づいて異なる指示を組み立てることが、単なる手段ではなく、プログラミング的思考が求められる理由であると述べている<sup>(7)</sup>。異なったリソースを提供し、与えられたリソースを用いてプログラミングを行うことを取り入れることで、従前のブロック型プログラミング環境を用いるだけでは得られない深い学習を行うことができ、テキストベースのプログラミング言語を用いたときには行えない、認

知構造の違いに着目した深い学習が期待できる。

本研究では、以上のことを実現するための第一歩として、相互評価機能を持ったブロック型プログラミング環境の設計と開発を行った。

## 2. 相互評価機能を取り入れたプログラミング環境の設計および開発

### 2.1 本システムの概要

本研究では、相互評価機能を取り入れたブロック型プログラミング環境を提案する。本システムは、Block Assembly System, Browsing System, Server の3つのプログラムから構成されており、前者の2つはGoogleが提供するライブラリであるBlocklyを用いた<sup>(8)</sup>。Serverの開発にはPythonを用いた。Block Assembly Systemを図1に示す。この図では、例としてJavaScriptを動かしている。Block Assembly Systemは、左側の領域にブロックを組み上げる部分、その下にコードを表示するDisplayボタン、プログラムを実行するRunボタンが配置されている。そして、右側の領域に、課題となる問題、他ユーザからのフィードバック、レーティングが表示されている。次に、Browsing Systemを図2に示す。Browsing Systemでは、Block Assembly Systemの組み上げたブロックを閲覧することができ、それに対してコメントやフィードバック、レーティングを行うことができる。

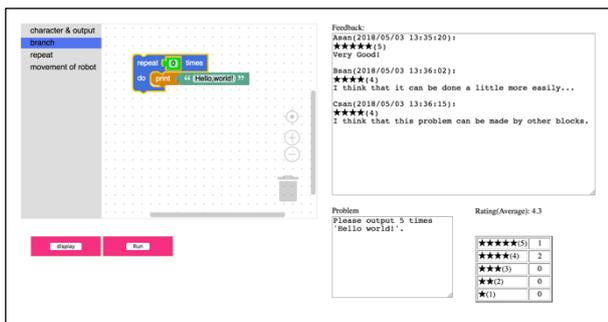


図1. Block Assembly System

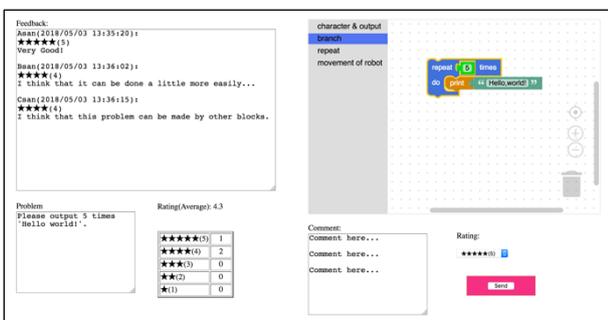


図2. Browsing System

それぞれのシステムの関係について図3に示す。つまり、クライアントAがBlock Assembly Systemで組み上げたブロックを、他のクライアントB,C,...はBrowsing Systemの画面で閲覧することができる。そして、クライアントB,C,...は、クライアントAが組み上げたブロックを見て、コメントや評価、レーテ

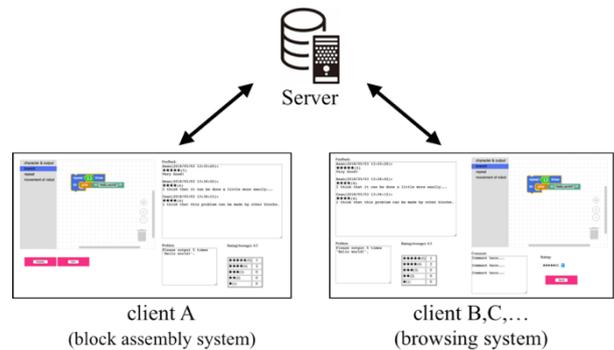


図3. 本システムの全体像

イングなどが行うことができる。このシステムを用いることで、従前のプログラミングの学習をさらに深めることが期待できる。

### 3. まとめと今後の展望

本稿では、相互評価機能を取り入れたブロック型プログラミング環境の設計を行い、その第一歩として、ブロック型プログラミング環境で相互評価を行うためのシステムの機能およびサーバを開発した。

今後、自身のコードと他者のコードの比較を行える機能、その相違点を指摘させる機能の実装を行う。また、学習段階に応じて、学習者にどちらのコードが正しいか、より適切かなどといったことを選ばせるような機能を実装し、従前のプログラミング教育では実現できなかった、深い理解を促すプログラミングの学習の方略を模索する必要がある。

#### 謝辞

本研究の一部は、科研費(18J20759)の支援を受けて実施されたものである。

#### 参考文献

- (1) 文部科学省, “学びのイノベーション事業”. (2014) [http://jouchouka.mext.go.jp/school/pdf/manabi\\_no\\_innovation\\_report.pdf](http://jouchouka.mext.go.jp/school/pdf/manabi_no_innovation_report.pdf)
- (2) Kollar, I. & Fischer, F.: “Peer assessment as collaborative learning: A cognitive perspective”, *Learning and Instruction*, **20**(4), pp.344-348, (2009)
- (3) Laal, M. & Ghodsi, S. M.: “Benefits of collaborative learning”, *Procedia - Social and Behavioral Sciences*, **31**, pp.486-490. (2012)
- (4) 加藤優哉, 松澤芳昭, 酒井三四郎: “初学者の協調プログラミングにおけるインタラクションの分析”, 情報教育シンポジウム 2015 論文集, pp.177-184. (2015)
- (5) 北栄輔, 山梨樹里: “Peer Review に基づいたプログラミング実習授業支援ツールの開発”, 名古屋高等教育研究, **7**, pp.341-353. (2007)
- (6) MIT: “Scratch- Image, Program, Share”, (n.d.) <https://scratch.mit.edu>
- (7) 平嶋宗, 福井昌則, 林雄介: “異種リソースから作られた指示の対比によるプログラミング的思考の振り返り”, 第43回教育システム情報学会全国大会講演論文集, (2018)
- (8) Google for Education: “Blockly”, (n.d.) <https://developers.google.com/blockly/>