

子ども向けプログラミング体験オンラインイベントの実施と学習効果の調査

松本 章代^{1,a)} 稲垣 忠¹ 菅原 研¹

概要：東北学院大学教養学部情報科学科では、2014年度から毎年、子ども向けプログラミング体験イベントを開催している。2020年度はコロナ禍の中、初めてオンラインでイベントを実施した。2020年度は計3回開催し、実数で22名（延べ数43名）の子どもたちが参加した。また、各回のイベントの中で「プログラミング的思考力を測るテスト」をプログラミング体験の前後に実施した。本稿では、プログラミングを教えるイベントをオンラインで開催するという試みで得られたノウハウと、テストの結果について報告する。

1. はじめに

近年、子ども向けのプログラミング教育に注目が集まっている。特に文部科学省が小学校でのプログラミング教育を2020年度から必修化する方針を打ち出して以来、世間的な関心も顕著に高まっている。文部科学省によると、小学校におけるプログラミング教育のねらいのひとつに「プログラミング的思考を育むこと」が挙げられる [1]。一方、「プログラミングを通じて学べることは何か？実際にプログラミングを学ばせたことで、論理的思考力や、創造性が高まったか？さらには、その他の教科（数学など）での成績が上がったか？」といった因果関係をダイレクトに実験的に調査し、結論を出しているものはない [2]。

そこで本研究では、プログラミングイベントを開催しその参加者を対象として、プログラミングを学ぶことでプログラミング的思考は養われるのかについて調査を実施する。なお、本研究ではプログラミング的思考に基づく思考力のことを「プログラミング的思考力」と呼ぶ。

2. 本研究の位置づけ

2.1 これまでの経緯

東北学院大学教養学部情報科学科では2014年度から毎年、子ども向けプログラミング体験イベントを公開講座として開催している。プログラミング体験をとおして「コンピュータに指示を出し思い通りに動かす楽しさ」を味わってもらい、ひいては子どもたちに「情報」という学問分野に興味を持ってもらうというねらいである。こちらの指示

に従って画一的なプログラムを作成するのではなく、子ども自身が発想したものを形にできる体験となるよう配慮している。2017年度までの取り組みにおいては「プログラミングに興味・意欲を持たせることに成功した」と言える結果が得られた [3]。

そこで2018年度および2019年度は、参加した子どもたちがイベント会場のみならず家庭内でも自発的にプログラミングを学んでくれることを目指した。参加者の子ども・保護者を対象としたアンケート調査の結果、2019年度は多くの子どもたちが家庭でもプログラミングに取り組んでいる状況を確認できた [4]。

2020年度、すなわち本研究では、ビデオ会議システムを用いてオンラインでイベントを開催し、新たにプログラミング的思考力に焦点を当て、子どもがプログラミングを学ぶことでプログラミング的思考が養われるのかを調査する。

2.2 子どもを対象としたプログラミング教育の効果

栗山らは小学4～6年生のプログラミング教育を受けた群と受けていない群に対してプログラミング的思考力テストを実施し、「実施群の方が未実施群よりも思考力テストが伸びた可能性が示唆される」と述べている [5]。具体的なプログラミングの教育内容や思考力テストの内容が公表されていないため本研究との比較は難しいが、被験者の規模が数千単位と多く興味深い。

2.3 子どもを対象としたオンライン教育の取り組み

他大学で開催されている子ども向けプログラミング教室においても、新型コロナウイルスの影響を受け、オンラインで実施したという報告がある [6]。この島崎らの取り組み

¹ 東北学院大学
Tohoku Gakuin University, Sendai, Miyagi 981-3193, Japan
^{a)} akiyo@mail.tohoku-gakuin.ac.jp

みでは、独自にオンライン学習システムを開発し利用している。我々はオンライン化にあたり Zoom を利用したが、より適したツールの利用も検討していく必要があると思われる。

3. イベントに使用した教材

3.1 教材概要

2018 年度と 2019 年度は micro:bit と MakeCode を組み合わせ、モーターやスピーカーを接続して制御する内容でおこなっていたが、オンラインでは難しいと判断した。そこで、2020 年度のイベントでは、micro:bit と Scratch3.0 を用いることにした。Scratch と組み合わせることによって出力先が PC のディスプレイモニターになるため、トラブルが起こりにくいことが期待できる。

micro:bit を利用せず Scratch のみでプログラムを作成すると、プログラミングから実行まで Scratch の中だけで完結してしまうが、Scratch と micro:bit を bluetooth で接続し連携することで、micro:bit を入力装置とした PC のゲームを作ることが可能となる。たとえば、micro:bit には加速度センサが付いているため、それを利用して micro:bit を傾けるとそれに応じて PC 画面上のキャラクタを動かしたりすることができる。

3.2 イベントで用いるプログラミングのテキスト

子どもたちが作成するプログラムの題材は、各イベントごとに独自に考案しテキストを作成する。

2020 年度のイベントでは、プログラムの作成手順を解説するテキストを PDF ファイルや動画ではなく紙媒体の冊子で用意した。紙媒体にした理由は、PC モニタにはテキストを表示せず Scratch の画面が占有した方が作業の効率が良いためである。

micro:bit と冊子テキストは、事前に参加者の自宅に郵送する。

3.3 プログラムを拡張するための教材

イベントで作成するプログラムをさらに拡張するための教材を用意する。この拡張機能の作り方を紹介した PDF ファイルおよび完成動画はウェブページで提供する。イベント中にテキストの内容を早く終えて時間が余った子どもには、その時点でこのウェブページに誘導し、さらにプログラムを発展させることができる。それ以外の子どもにもイベント終了時にこのウェブページをアナウンスし、イベント終了後にもプログラミングに取り組んでもらえるようにする効果が期待できる。

各イベントごとに拡張機能は複数用意しており、その中から各自作りたいものを選択できるようになっている。また、我々が考案した拡張機能を参考にして子ども自身が自由にプログラムを拡張することにもつながるはずである。

4. イベント概要

4.1 募集内容および方法

2020 年度のイベントは、4 週間おきに計 3 回、毎回異なる内容で実施することにした。時間は午前と午後の 1 日 2 回（各 120 分間）、同一内容で実施し参加者が都合の良い時間帯を選べるようにし、定員をそれぞれ 8 名と設定した。対象学年は原則として小学 4 年生～中学生とした。2020 年度はオンラインでの開催となるため、「ネットに接続しているカメラ・マイク・Bluetooth 付きの Windows10 のパソコンを用意できること」を参加条件とした。都合の良い 1 回のみ参加も、複数回の参加も、どちらも認めることにした。

イベントへの参加者の募集には、Peatix や大学ウェブサイトなどで告知をおこなった他、昨年度のイベント参加者へも声をかけた。Peatix とは、イベント参加者の募集やイベント管理ができるウェブサービス・モバイルアプリである。

4.2 実施方法

今回は、ビデオ会議システム Zoom を用いてオンラインでイベントを実施する。参加者にはあらかじめ、自宅の WindowsPC に Scratch3.0 と ScratchLink (Scratch と micro:bit を bluetooth で接続するためのソフト) をインストールするよう依頼しておく。その同じ PC で Zoom に接続してもらう。Zoom の画面共有機能を使用して、参加者のプログラミングの状況を講師が確認するためである。

プログラミングを教える際やプログラミング的思考力を測るテストを実施する際には、Zoom のブレイクアウトルーム機能を用いて講師が子どもとマンツーマンになる。講師は常時 4 名で子どもは最大（同時に）8 名とし、各講師は 1 人 2 台のノート PC を利用して各 PC に子ども 1 名のみが繋がった状態となる。

イベントの流れは次のとおりである。

- (1) まずイベントの冒頭で、プログラミング的思考力を測るテストを実施する。
- (2) 全体に対し、プログラミングとはどのようなものか、プログラムがどのように社会に関わっているのか、といった話をまずおこない、その後イベントの内容について説明する。
- (3) 子どもがテキストを見ながら各自のペースでプログラミングを体験する。この間、講師は質問に答えて参加者のプログラミングをサポートする。
- (4) プログラミング終了後、再びプログラミング的思考力を測るテストを実施する。テストの内容は、イベント開始直後のテストと類似した問題である。
- (5) 最後に、子どもと保護者それぞれにアンケートに回答してもらう。

なお、プログラミング的思考力を測るテストとアンケー



図 1 野球ゲーム

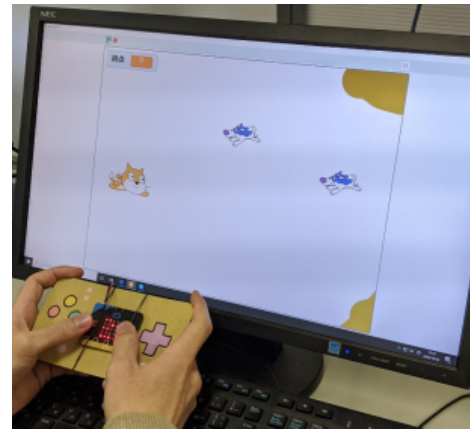


図 2 シューティングゲーム

トはオンラインで解答/回答できる。

4.3 参加者

第 1 回, 第 2 回, 第 3 回のイベントの参加人数と内訳を表 1 に示す。実数で 22 名 (延べ数 43 名) の子どもたちが参加した。そのほとんどが Peatix からの申し込みであり, 北は北海道から南は九州まで, 全国からの参加があった。

表 1 参加人数と内訳

	初参加	継続参加	合計
第 1 回	13	-	13
第 2 回	4	11	15
第 3 回	5	10	15
合計	22	21	43

4.4 イベント内容

4.4.1 第 1 回イベント

第 1 回のイベント内容は「野球ゲームの作成」である (図 1)。micro:bit をバットに見立て, 「micro:bit をタイミングよく振ったら, ボールが得点ゾーンに飛ぶ」プログラムを Scratch で作成する。プログラムは「変数」を扱う内容となっている。なお, 参加者にはイベント当日までに新聞紙等を使って, ゲームに使用するバットを作ってもらった。

4.4.2 第 2 回イベント

第 2 回のイベント内容は「シューティングゲームの作成」である (図 2)。PC の画面に表示されるキャラクターが micro:bit の A ボタンと B ボタンで上下移動し, micro:bit を傾けると敵キャラクターに攻撃するプログラムを Scratch で作成する。プログラムは第 1 回と同様に「変数」を扱う内容となっており, 難易度があがっている。なお, 参加者にはイベント当日までにダンボールを使って, ゲームに使用するコントローラーを作ってもらった。

4.4.3 第 3 回イベント

第 3 回のイベント内容は「ドライブゲームの作成」である (図 3)。micro:bit を左右に傾けると画面内の車も左右

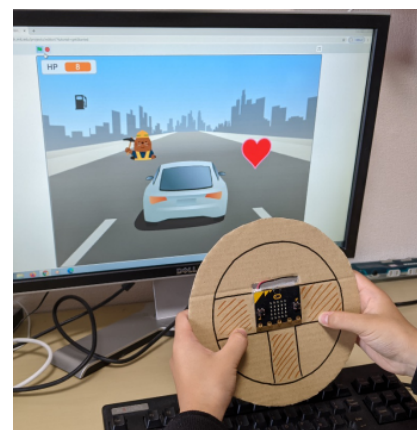


図 3 ドライブゲーム

に移動し, ゲーム背景の道路の白線を動かして前進を表現するプログラムを Scratch で作成する。プログラムは「変数」に加えて新たに「関数」を扱う内容となっており, さらに難易度があがっている。なお, 参加者にはイベント当日までにダンボールを使って, ゲームに使用するハンドルを作ってもらった。

4.5 参加者の作品集

第 1 回イベント終了時に参加者から「自分で作ったゲームを見せる会をしたい」という意見があったため, 第 2 回, 第 3 回のイベントにおいては参加者が作成したゲームをアップロードしてもらい, ウェブ上に作品集を作成し公開した。公開対象は Scratch のプログラムと実行画面のスクリーンショット, 作者のニックネームである。実際の作品集の一部を図 4 に示す。

5. プログラミング思考力を測るテスト

5.1 テストの目的

本研究では, プログラミングイベントを開催しその参加者を対象として, プログラミング思考力を測るテストを実施する。プログラミングを学ぶことでプログラミング的思考は養われるのかについて調査することを目的とする。プ

2020年12月12日プログラミング体験教室 作品集

画像上でクリック（タップ）するとダウンロードできます。



図 4 作品集

プログラミング体験前後や約 3 か月間における得点の変化に着目する。

5.2 テストの概要

我々が開発したプログラミング思考力を測るテストの形態は、ウェブアプリである。今回、ペーパーテストではなくウェブアプリにしたのは、以下の理由からである。

- スマートフォンのクイズアプリで遊ぶような感覚で気軽にテストを受けてもらいたかったため。
- オンラインでのイベント開催となったため、ウェブアプリの形態が解答の回収をもっとも効率的におこなえるため。

以下、プログラミング思考力を測るテストのウェブアプリをテストアプリと呼ぶ。

実施の際は、参加者に Zoom のチャットでテストアプリ 1 回分の URL を送る。URL にアクセスすると参加者の手元にある PC でテストを受けることができる。

問題の内容は、市販のドリル [7] を参考にして作成した。

問題は、テスト 1 回分が大問 6 問で構成されており、問題数は 10 問である。詳しい問題構成については表 2 に示す。各大問の先頭には「例題」の画面を設けて解き方の説明を入れている。

問題はイベント参加回数に応じて変える。つまり 1 回目の参加者用、2 回目の参加者用、3 回目の参加者用とそれぞれ問題を用意する。

プログラミング体験前後の点数の増減を確認するため、イベント開始直後とイベント終了直前のテスト内容は類似問題とする。例として、1 回目のテストアプリの第 1 問について、イベント開始直後とイベント終了直前の問題を図 5 と図 6 に示す。

参加者がテストアプリを利用し最後の問題の解答を終了すると、サーバに解答内容が保存される。

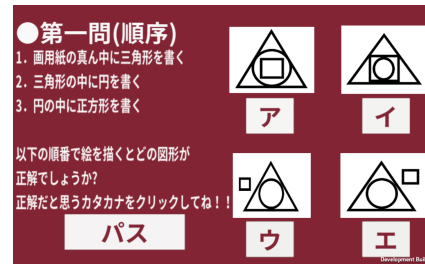


図 5 イベント開始直後

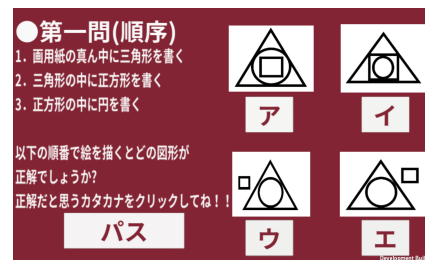


図 6 イベント終了直前

表 2 問題構成

内容	大問	問題数
順序	1 問	1 問
繰り返し	1 問	1 問
分岐	2 問	2 問
変数	2 問	6 問
計	6 問	10 問

5.3 テストの実施方法

テストアプリを用いて、イベントの開始直後と終了直前の 2 回、プログラミング的思考力を問う問題を解いてもらう。問題数は計 10 問、制限時間は特に設けない。分からない問題は直感で解答せずにパスするよう指示する。

5.4 テストの結果

イベント参加者に ID:1~ID:22 の番号を割り振る。

1 回目のテストに取り組んだのは、第 1 回イベント参加者で 13 人、第 2 回イベントで 4 人、第 3 回イベントで 5 人であり、計 22 名である。

2 回目のテストに取り組んだのは、第 2 回イベント参加者で 11 人、第 3 回イベントで 2 人であり、計 13 名である。

ここで、5.2 節で述べたように、1 回のテストの問題数は 10 問である。ただし、第 2 回イベントでの 2 回目のテストの問題の中に 1 問、正答のない問題を出題してしまったため、2 回目のテストのみ 9 点満点になっている。表 3 の中で、9 点満点の欄に*印が付いている。

3 回目のテストに取り組んだのは、第 3 回イベント参加者のうち 8 名である。

イベント開始直後のテストを前、イベント終了直前のテストを後とし、1 回目~3 回目までのテストの各参加者の得点をまとめたものを表 3 に示す。また、表 3 の中に参加

者の学年とプログラミング経験についての基本情報も含めている。プログラミング経験については次の選択肢から被験者の保護者に回答してもらっている。

- (a) プログラミング経験がない
- (b) プログラミング教室に通っている
- (c) プログラミングイベントに参加したことがある
- (d) 自宅で独学

表 3 テスト得点一覧

	学年	経験	1 回目		2 回目		3 回目	
			前	後	前	後	前	後
ID:1	小 4	(b)	3	3	*3	*2	1	5
ID:2	小 4	(d)	10	10	*8	*9	3	7
ID:3	小 4	(c)	6	9	*8	*6	3	8
ID:4	小 2	(c)	8	7	*9	*9	2	5
ID:5	小 5	(b)	8	10	*9	*9	6	8
ID:6	小 5	(c)	9	10	*9	*9	6	10
ID:7	小 4	(a)	2	3	*1	*1	—	—
ID:8	小 4	(c)	2	2	*4	*3	—	—
ID:9	小 4	(d)	6	3	*3	*5	3	8
ID:10	中 1	(c)	9	8	—	—	—	—
ID:11	小 5	(b)	3	7	—	—	—	—
ID:12	小 5	(c)	6	8	*8	*9	6	9
ID:13	小 4	(c)	5	4	*6	*9	—	—
ID:14	小 6	(c)	3	8	—	—	—	—
ID:15	小 4	(c)	4	2	—	—	—	—
ID:16	中 1	(c)	5	6	*9	*8	—	—
ID:17	小 4	(c)	5	5	*3	*2	—	—
ID:18	小 4	(c)	1	4	—	—	—	—
ID:19	中 2	(d)	5	8	—	—	—	—
ID:20	小 5	(b)	7	6	—	—	—	—
ID:21	小 4	(c)	5	7	—	—	—	—
ID:22	小 4	(c)	6	5	—	—	—	—
平均 (全体)			5.4	6.1	6.2	6.2	3.8	7.5
平均 (3 回目参加者のみ)			7.0	7.5	7.1	7.3	3.8	7.5

5.5 分析および考察

5.5.1 プログラミング体験前後における得点の増減

プログラミング体験の前後で、得点が増加した事例が 24 件 (24/43 = 55.8%)、増減なしが 8 件、減少が 12 件であった。ただし増減なしのうち 5 件 (5/43 = 11.6%) は満点である。

ID:9 の被験者は、1 回目のテストで唯一、プログラミング体験前後でテストの得点が -3 と大きく下降している。そこで、ID:9 のテストの解答結果をより細かく分析したところ、表 4 のようにイベント開始直後のテストとイベント終了直前のテストの解答が全く同じ、という問題が数問見受けられた。また ID:1 の被験者も同様の傾向が認められた。つまり、図 5 と図 6 に示したとおり、イベント開始直後のテストとイベント終了直前のテストで見た目そっくりな問題が出題されるため、同じ問題と勘違いした可能性

が高い。この仮説を検証するため、第 3 回のイベントにおいては、イベント終了直前のテストを実施する際にイベント開始直後の問題とは異なる問題であることを参加者に伝えた。その結果、第 3 回のテストでは 2 つの解答結果が同一であることによってイベント終了直前のテストを間違えている被験者は 1 人もいなくなり、3 回目のテストを受けた全員、さらに第 3 回に 1 回目のテストを受けた被験者 (ID:18~22) も全員、得点が増加した。

表 4 ID:9 の 1 回目のテストアプリの解答の一部

	第 1 問 第 2 問 第 3 問 第 4 問			
	第 1 問	第 2 問	第 3 問	第 4 問
イベント開始直後	1	4	1	2
イベント終了直前	1	4	1	2

5.5.2 約 3 か月間における得点の変化

約 3 か月間における得点の変化について考察する。

表 3 の平均 (全体) を確認すると、3 回目のイベント開始直後の平均点が他と比較して極端に低く、それを除けば概ね順調に得点が増加しているように見える。ただし 3 回目までテストを受けている被験者 8 名のみを対象として平均点を求めると、「順調な得点上昇」といった傾向は見受けられない。別の見方をすれば、1 回または 2 回しかイベントに参加していない被験者と、3 回イベントに参加している被験者とで、得点率に差があるということになる。3 回イベントに参加している子どもは、それだけプログラミングに対し意欲的で、それがテストの結果にも影響している可能性がある。

続いて、3 回目のイベント開始直後の平均点が他と比較して極端に低かった理由について述べる。10 問中 6 問が変数の問題であり、この部分の正答率が過去 2 回 (1 回目 38.6%、2 回目 61.5%) と比べて非常に低い (18.8%) ことがわかった。1 回目と 2 回目の変数の問題は、数値を変数に代入したら上書きされるという条件に基づいていた。しかし、3 回目の問題は、変数に代入したときに既に中身が入っていた場合は上書きせずにその時空いている別の変数に代入するという条件に変えていた。通常のプログラミングでは変数の代入は上書きなので、問題文をよく読まずに先入観で解くと間違えてしまう。例として 2 回目と 3 回目のテストアプリの変数の問題 (例題) をそれぞれ図 7 と図 8 に示す。意図したわけではないが結果としてひっかけ問題になってしまったことは失敗であった。

5.5.3 学年別の分析

「国語」や「算数」などの学校で学ぶ科目であれば、その理解度が学年によって大きく差があることは当然である。一方「プログラミング」はこれまで学校で扱ってきかず、学年によって「プログラミング的思考力」に明確な差はない可能性がある。

そこで、1 回目のテスト結果において学年別に平均点を求めたところ、小 4 と小 5 以上に分けたときに大きな差が

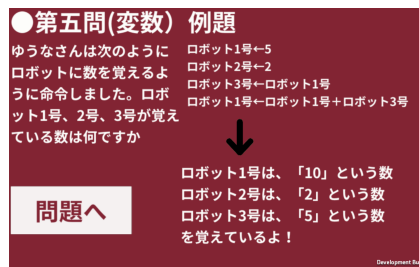


図 7 2 回目のテストアプリの変数の例題

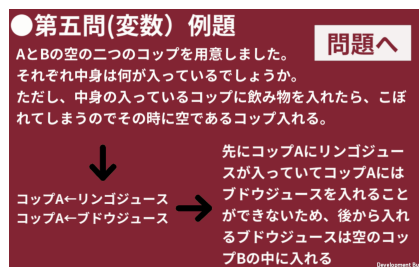


図 8 3 回目のテストアプリの変数の例題

あり、小5以上は学年による明確な差は認められなかった(表5)。

表 5 学年別の平均点 (1 回目)

	前	後
小4	4.6	4.8
小5	6.6	8.2
小5～中学生	6.1	7.9

5.5.4 参加者のプログラミング経験別の分析

普段から日常的にプログラミングに触れている子の方がプログラミング的思考力が高いのかを確認するため、参加者のプログラミング経験別に平均点を求めた。すると明確に (d) すなわち自宅で独学の平均点が高かったが、これは (d) に該当する被験者が 3 名のみであり、ID:2 の影響(前・後とも満点)が大きいためであることが分かった。

プログラミング経験で分析するのであれば、より詳細にプログラミングに取り組む頻度や期間を考慮する必要がある。また選択肢 (a)～(d) は複数回答可にすべきであった。

6. まとめ

本研究では、独自に子ども向けプログラミング教材の開発を行い、プログラミングイベントをオンラインで開催した。さらに、イベント参加者を対象としてプログラミング的思考力を測るテストを実施し、子どもたちがプログラミングを体験するとプログラミング的思考が養われるのかについて調査した。その結果、プログラミング体験前後で得点が増加したケースおよび前後とも満点のケースは合わせて 67.4%であった。3 か月間における得点の伸びは認められなかったが、3 回とも参加するようなプログラミングに意欲がある子どもは初回から得点率が高いことが確認さ

れた。

今後の課題としては、テストアプリの以下の問題点を改善すべきである。

- テストアプリの制限時間を設けなかった点
- テストアプリの問題を先に進めてしまうと前の問題に戻ってもう一度解きなおすことができない点
- テストアプリの難易度設定を正確にできなかった点

これらの課題を解決することができれば、より正確なデータを取得し分析することができると思われる。

謝辞 本研究は JSPS 科研費 JP19K03090 の助成を受けたものである。

参考文献

- [1] 文部科学省: 小学校プログラミング教育の手引 (第三版), https://www.mext.go.jp/content/20200218-mxt_jogai02-100003171_002.pdf
- [2] 細田千尋: 脳科学からプログラミング教育を考える, 情報処理学会誌, Vol.61, No.11, pp.1120-1125 (2020.11)
- [3] 松本章代: 手順解説動画を用いたプログラミング教育—大学生と小学生, それぞれを対象とした実践報告—, 日本教育メディア学会研究会論集, No.44, pp.91-96 (2018).
- [4] 松本章代, 稲垣忠, 菅原研: 家庭での継続につながるような子ども向けプログラミングイベントの開催, 教育システム情報学会 (JSiSE) 2019 年度 第 6 回研究会 (2020.3)
- [5] 栗山直子, 森秀樹, 齊藤貴浩, 前川真一, 西原明法: プログラミングの思考力テスト (小学生版 Ver.1) の開発と検証—Pepper プログラミング教育における効果検証—, 日本教育工学会全国大会講演論文集, No.34 (2018.9)
- [6] 島崎俊介, 朝尾直己, 宮澤修, 安部博文: 電気通信大学プログラミング教室におけるオンライン学習システムの開発と試行, 教育システム情報学会 (JSiSE) 2020 年度 第 1 回研究会 (2020.5)
- [7] 兼宗進: ドリルの王様 5,6 年の楽しいプログラミング, 新興出版社啓林館 (2019)