

プログラミング的思考を操作として展開・評価する学習環境の提案

小林 祐太[†] 長谷川 忍[‡]

[†] 北陸先端科学技術大学院大学 先端科学技術研究科 〒923-1292 石川県能美市旭台 1-1

[‡] 北陸先端科学技術大学院大学 情報社会基盤研究センター 〒923-1292 石川県能美市旭台 1-1

E-mail: [†] s1910095@jaist.ac.jp, [‡] hasegawa@jaist.ac.jp

あらまし 令和2年度から導入された初等教育におけるプログラミング学習では、学習の過程で獲得される「プログラミング的思考」の育成が推進されている。しかし、現段階では児童のプログラミング的思考の程度を捉えた定量的な評価方法は明確にはなっておらず、適切に育成していくために有効な学習環境が整っていない状況にある。そこで本研究では、プログラミング的思考を演習中の操作から展開・評価する学習環境を構築することを目的とする。先行研究で示されたプログラミング的思考の教育目標を展開した演習課題を設定するとともに、課題遂行時の操作を定量化することができる Web アプリケーションを開発することにより学習環境を実現する。

キーワード プログラミング的思考, 操作, 思考外化, プログラミング教育, 学習環境

A Proposal for Learning Environment to Evaluate Computational Thinking for Elementary School Students in Japan from Operation Point of View

Yuta KOBAYASHI[†] and Shinobu HASEGAWA[‡]

[†] Graduate School of Advanced Science and Technology, JAIST 1-1 Asahidai, Nomi-shi, Ishikawa, 923-1292 Japan

[‡] Research Center for Advanced Computing Infrastructure, JAIST 1-1 Asahidai, Nomi-shi, Ishikawa, 923-1292 Japan

E-mail: [†] s1910095@jaist.ac.jp, [‡] hasegawa@jaist.ac.jp

Abstract This research aims to propose a learning environment to assess the "computational thinking" of elementary school students from their operations during exercises. Programming learning in the Japanese elementary school introduced in 2020 has promoted the development of computational thinking acquired during the learning process. However, at the current stage, there is no precise quantitative evaluation method to grasp their computational thinking ability and no productive learning environment to develop the children appropriately. To realize the situation, we have designed exercises based on the educational goals for computational thinking presented in some previous studies, and we are developing a web application that quantifies children's operations during the exercises and connects them to assess their computational thinking abilities.

Keywords Computational Thinking for Elementary School in Japan, Operation, Externalization of Thinking, Programming Education at Elementary School, Learning environment

1. はじめに

平成29年に発表された新学習指導要領により、令和2年度から小学校でのプログラミング教育が必修化された[1]。この取り組みの背景には、プログラミングを学習する過程で獲得される「プログラミング的思考」の育成を政府が推進していることにある。プログラミング的思考とは「自分が意図する一連の活動を実現するために（中略）論理的に考えていく力」のことであり、今後の予測不能な社会における問題解決能力として期待されている[2]。しかし、必修化における問題の一つにプログラミング的思考の評価方法が明確でないことが挙げられる。これはプログラミング的思考が暗黙的であり、そのレベルや成長を外から観測すること

が難しいためである。そのため、文部科学省の提唱するプログラミング的思考と、その原点とされる「コンピュテーショナル・シンキング（計算論的思考）」のそれぞれの観点から、日本の初等教育で扱うプログラミング的思考の育成目標や評価規準について、議論が重ねられてきた。実際に小泉らは、プログラミング的思考の評価規準を策定するため、英国で実施されていたコンピュテーショナル・シンキングに基づき、文部科学省の提唱するプログラミング的思考を6つの構成要素で整理した[3]。さらに、Benesseは文部科学省が定めたプログラミング教育における「思考力・判断力・表現力等」において、この6要素を「プログラミング教育を通じて目指す育成すべき資質・能力」の目

標として位置付けた[4].

従来の評価手法においては、プログラミング的思考という目に見えない概念を捉えることから、前述したプログラミング的思考を根幹とした活動について、研究者が定めた達成度による評価が行われている。例えば西野らは、小泉らの評価規準を基にプログラミング学習中の思考を捉えるべく、3段階の客観的評価を行うルーブリックを作成し、運用した[5]。しかし、達成度の基準による位置づけにとどまり、より詳細な能力のレベルを定量的に評価するまでには至っていない。

一方、プログラミング学習を推進するための教材は、現在も研究、開発が進められており、文部科学省からは、プログラミング的思考の育成について触れた、プログラミング学習を導入するための研修教材が提供されている[6]。しかし、プログラミング的思考の構成要素を可視化し、要素に応じた育成を目的とした学習環境については未だ整っていない。

そこで本研究では、児童の学習中のプログラミング的思考の程度を捉えるため、実際のプログラミング学習を通じながら、思考による能力を形成的に評価する学習環境の構築を目的とする。具体的には、プログラミング的思考の構成要素に基づく学習の活動を定義し、演習中の操作から定量的な数値データ（操作回数や解答結果等）を抽出する Web アプリケーションを開発する。

2. 関連研究と本研究の位置づけ

2.1. プログラミング学習の流れ

文部科学省の「小学校プログラミング教育の手引き」によれば、問題を見出し、問題解決に至る間のステップとしてプログラミング的思考が位置付けられており、問題の解決プロセスが例として挙げられている[7]。ただし、プログラミング学習としての流れは明示されておらず、学校の方針に応じ、指導計画が行われている状態である。

一方、中等教育ではあるが、大村はソフトウェア設計の考え方を取り入れたプログラミング学習のプロセスを考案している[8]。大きな特徴としては、目的のプログラムを構築するにあたり、設計のプロセスも評価した学習体系にするという点である。プログラムへの具体化は、設計における過程（エラーや仕様等）も含めて反映されるということであり、設計段階から論理的な思考が働くことは容易に想像できる。

しかしながら、プログラミング学習支援システムの多くは開発と評価の部分にフォーカスが当てられており、設計部分を意識させることが少ないのが現状である。例えばロボットの動作を出力とするプログラミング教材では、メインの活動としてコーディングを扱う

ものの、直接ロボットの動きを確認しながらコーディングを行うことを主体としているため、これらの教材のみでの設計部分の意識づけを行うことは難しい。

2.2. プログラミング学習支援システム

近年では、教師が教具として用いることを想定したプログラミング学習支援システムが開発されている。

大日本印刷の「SWITCHED ON Computing 日本版」では、紙媒体ではあるが、指導書や指導案も含めたプログラミング学習全体を捉えた学習環境を提供している[9]。また、Matayoshi らは学習者のプログラミング中の制御構造の抽象的な理解度を把握、支援するため、自然言語で記述できるエディタ内のコメント機能をつり構造として再現し、操作可能にしたプログラミング学習支援システムを制作している[10]。

2.3. プログラミングの自動評価システム

プログラミングの操作の結果より、学習中の思考を捉えようとした例もいくつかある。

太田らは、プログラミング的思考の原点とされるコンピュータシミュレーション・シンキングの概念について、実際に学習者の構成したプログラムから発現された操作の自動評価システムを開発している[11]。また、國宗らはプログラムを作成する上での操作や制御構造を自動評価するシステムを作成しており、教員の作成したテストケースより、学習者が組み上げたコードの達成度を可視化させている[12]。

2.4. 本研究の位置づけと方針

本研究の位置づけとしては、プログラミングにおける制御構造の構築の際の思考を捉えるのではなく、プログラミング学習全体を通して、課題解決の度に発現するプログラミング的思考の要素に基づく活動を捉えようとしている。つまり、設計の段階においても具体的な活動を提示し、実行結果の振り返りまでの学習について扱う。これを実現するには、思考の要素に基づいた学習中の活動を定義し、各要素が最も働く場面での活動を抽出して評価する必要がある。このプログラミング的思考の要素に基づく活動の観測を、本研究では「プログラミング的思考の外化」とし、これを実現するシステムを開発するため、要素に基づく活動（思考活動）の定義、プログラミング学習の流れ（学習フェーズ）の設定、活動を捉える課題（思考課題）のデザインを行った。なお、思考課題との混同を避けるため、学習フェーズを通して行われる問題解決のための題材を「問い」と定義する。

最終的な学習支援システムとしては、教師がプログラミング学習を進める上での、一つのテンプレートと

して学習環境を展開し、中でもプログラミング的思考における思考能力の程度を把握するための支援を行う。先行研究にならない、設計と開発のそれぞれに根差したアプリケーションをデザインし、学習者に対するプログラミングの補助として、制御構造を視覚的に理解できるような自然言語を用いたエディタを提供する。

3. プログラミング的思考の外化プロセス

3.1. 思考活動の定義

関連研究を基に、プログラミング的思考を観測するための外化プロセスを考えていく。

まず、プログラミング学習中に発生する思考を詳細に捉えるため、表 1 の Benesse の定めたプログラミング的思考の教育目標より、6 つの構成要素に基づく学習中の活動を「思考活動」として定義し、表 2 にまとめた。思考活動は、後述のシステムを利用した学習を通じて発生する操作を想定しており、この活動における頻度を児童のプログラミング的思考に対する形成的評価の指標として捉える。

表 1 プログラミング的思考における評価規準 (Benesse[4])

構成要素	教育目標
論理的に考えを深める	コンピュータの動きを自らの問題解決で使うために論理的推論を行う
動きに分ける	大きな事象を解決可能な小さな事象に分割する
記号にする	分割した事象から適切な側面・性質を抜き出す
一連の活動にする	記号（動き）の類似部分を特定し、別の場面でも利用できる内容にする
組み合わせる	目的に合わせてよりよい手順を創る
振り返る	目的に対する評価の観点を考え、結果が意図した活動に近づいたか評価する

表 2 思考活動の定義

思考活動	Bnesse該当目標	定義
分割	動きに分ける	大きな事象を解決可能な事象に分割する。
抽象化	記号にする	事象・概念から必要な側面や性質、要点を抜き出す。
一般化	一連の活動にする	事象から類似パターンを見つけ規則化する。
順序立て	組み合わせる	動作や組み合わせの順番を考える。
制御	論理的に考えを深める	コードによる動作を理解し、論理的推論の下でコーディングする。
分析	振り返る	1. 目的に対する評価の観点を考える。 2. 結果が意図した活動に近づいたかどうか評価する。

表 2 における「分割」、「抽象化」、「一般化」、「順序立て」は、学習中の操作を想定しているため、多少の文脈の違いはあるが、文意は Benesse の教育目標と同じである。「制御」についても同様であるが、今回は実際のコーディングを通じて観測を試みることを想定して定義している。「分析」においては、Benesse の教育目標を読み取ると 2 つの意味合いの下で定義されている。分析における本研究上での解釈として、定義 1 は演習の結果に対してどのような操作を行えば目的の意図に近づくのか、解決（操作）の観点を見出そうとす

ることであり、定義 2 は演習の結果が意図する結果になっているかどうか、確認を行おうとすることである。

3.2. 学習フェーズの設定

次に「学習フェーズ」を設定し、最も観測されやすい思考活動を各フェーズ内に位置付けた。表 3 は学習フェーズの定義と思考活動の位置づけをまとめたものである。フェーズを設定することにより、初等教育での授業設計において、本研究で提案する学習環境を導入する際に適応しやすくなると考える。

表 3 学習フェーズの定義と思考活動の位置づけ

学習フェーズ	観測する思考活動	定義
設計	抽象化 順序立て	目的のプログラムを実現するための仕様を策定するフェーズ
開発	分割 制御	設計に基づいた仕様をプログラムとして具体化するフェーズ
評価	一般化 分析	設計、開発フェーズでの振り返りや、結果の概念整理を行うフェーズ

学習フェーズは設計、開発、評価の 3 つで構成される。設計は出題された問いを解決するための大まかな方法を抽象化し、手順を組み立てるといったプランニングを行うフェーズであり、抽象化、順序立てといった思考活動が主に観測される。開発は抽象化された方法を詳細なコードに変換してプログラムを構築するフェーズであり、分割、制御などの思考活動が主に対象となる。評価においては、設計や開発の結果に対する振り返りを実施し、取り組んだ内容の整理を行うフェーズであり、分析や一般化を主に行うことが想定される。

3.3. 思考課題のデザイン

学習フェーズに対応するように、思考活動を捉えるための 4 つのパターンの「思考課題」を設計した。思考課題はシステム上で取り組まれる課題であり、操作の時間や回数など、学習者の思考活動を操作として検出する。図 1 は思考課題における設計イメージである。推論課題は「抽象化」と「順序立て」を捉える設計フェーズの思考課題であり、与えられた問いを解決するために、いくつかの行動が書かれたオブジェクトの中から必要な行動を抽象化し、それらを並び替えることで目的の順序の組み立てを行う。分割課題は「分割」を捉える開発フェーズの思考課題であり、推論課題で抽出した各行動を単一の動作（コード）として分解する。制御課題は「制御」と「一般化」を捉える開発フェーズの思考課題であり、関数等を用いながら制御構造を捉えた開発を行う。分析課題は「分析」を捉える

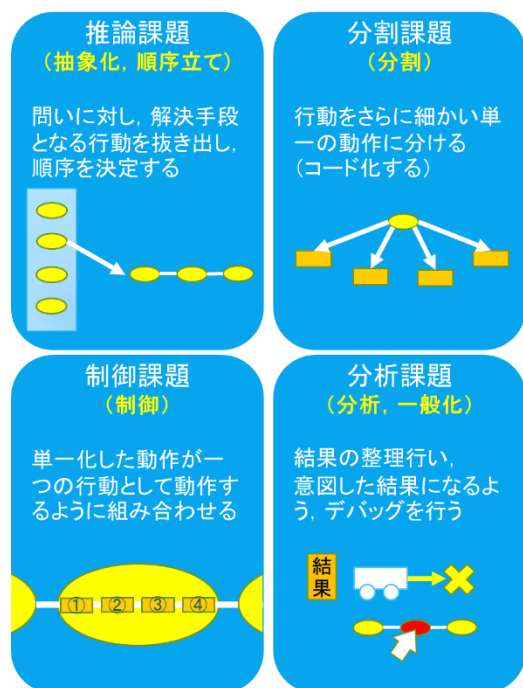


図 1 思考課題イメージ

思考課題であり，設計と開発の各フェーズにおいてデバッグを実施する．

これらの思考課題と学習フェーズ，思考活動の位置づけを図 2 に示す．

プログラミング学習			
フェーズ	設計	開発	評価
思考課題	推論課題	分割課題 制御課題	分析課題
思考活動	抽象化, 順序立て	分割 制御	分析, 一般化

図 2 プログラミング学習における各定義の位置づけ

4. Thinkron (シンクロン) の開発

4.1. Thinkron の設計指針

プログラミング的思考の外化を行うため，実際のプログラミング学習をサポートするシステムを提供し，システムにおける演習中の操作をプログラミング的思考の構成要素と対応付けることによって展開する．そこで，プログラミング的思考の外化プロセスを反映した，思考の育成支援及び学習評価を行うシステム「Thinkron (シンクロン)」を開発している．

まず，外化プロセスより実際のプログラミング学習中の操作を捉えることから，パソコンやタブレットにおける入力装置から思考活動の抽出を行う必要がある．デバイスについては各教育現場における環境の違いから複数種類に対応する必要がある，アプリケーション

の導入については，児童が操作することを考慮し，手動によるインストールの手間を省けることが好ましい．そこで，システム自体は Web アプリケーションとして開発し，学習フェーズに則ったプログラミング学習ができるものとした．

次に要件定義として，学習者の取り組む問いの設定を行う．形成的評価を行うことができる学習環境の構築にあたっては，教師が学習者のプログラミング的思考に対して，問いの内容をコントロールができることが望ましい．そこで教師用のフォームを作成し，問いの投稿や評価の閲覧ができるような機能を付与する．

4.2. 使用する JavaScript ライブラリ

今回は操作性の高い動的なアプリケーションを再現するため，「p5.js」という JavaScript のライブラリを用いる[13]．p5.js とはビジュアルデザイン用に設計された言語「Processing」のコーディング環境を JavaScript 上で再現したもので，Processing 特有のフレーム単位のアニメーション処理を JavaScript の DOM 操作と併用して実装することができる．これにより，html におけるタグ要素に対して高度な動作表現を実現することが可能になる．

学習中のエディタについては，ビジュアルプログラミングによるコーディング環境を提供する「Blockly」を用いる[14]．Blockly は Google が開発した JavaScript ライブラリであり，ブロックタイプのコードを組み合わせることでプログラムを表現することができる．また，作成したプログラムは JavaScript のコードとして変換することが可能であり，eval 関数 (JavaScript の標準ビルトインオブジェクト) と組み合わせることで，作成したコードをそのままブラウザ上で実行することができる．

つまり，JavaScript による DOM 操作と p5.js で各課題を構成し，コーディング部分においては上記に加えて Blockly を用いたものとなっている．

4.3. Thinkron の遷移モデル

Thinkron を具現化するために，遷移モデルについて整理した．図 3 に示すように，初めに教師側が問いを設定し，学習者は学習フェーズにしたがって思考課題に取り組む．設計と開発のそれぞれのフェーズ後に，目的的操作ができたかどうか，振り返りを実施する評価フェーズを設けており，学習者の判断で，次のフェーズ，もしくは学習を終了する．

全体像から見える Thinkron の強みとしては，これらの複数の思考課題を統合した UI で連続的なプログラミング学習が行えることにあり，この学習フェーズの流れから，プログラミング的思考を操作の一つ一つと

して観測・評価し、学習者の伸ばすべき思考要素を取り上げた学習内容を組み込んだプログラミング学習を再度展開することができる。

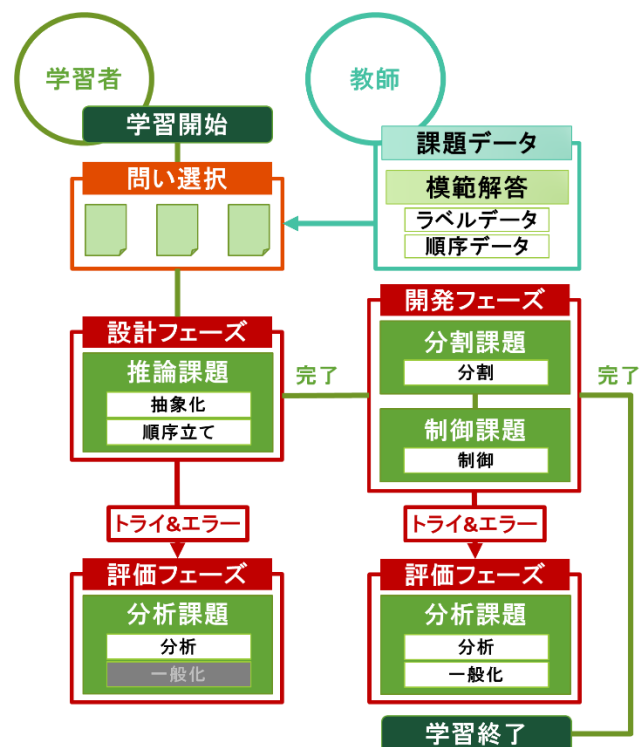


図3 Thinkron の遷移モデル

4.4. 思考課題の実装

4.4.1. 推論課題

図4に開発段階の推論課題を示す。出題された問いに対し、あらかじめ用意された行動ラベル群から解決に必要なラベルをキャンパス上のレシーバに選出していく。更に、選出された行動ラベルを適切な順序に並び替えることによって、問いを解決する手順を作り出す。抽象化においては、レシーバの増減による操作過程の他、行動ラベル群における正解ラベルの含む割合（再現率）や児童の選択の正解の割合（適合率）より評価する。順序立てにおいては、レシーバ上のラベルの入れ替え数と、ラベル順序における最適解との比較により評価を行う。

4.4.2. 分割課題

図5に開発段階の分割課題を示す。推論課題で選定した行動ラベルのそれぞれを構成するように、動作ラベル群の中から必要な動作ラベルをキャンパス上にある該当レシーバの分割リスト内に移動する。分割においては、分割リストにラベルを追加、または削除した操作過程の他、動作ラベル群の再現率と選出した動作ラベルの適合率から定量化を図る。

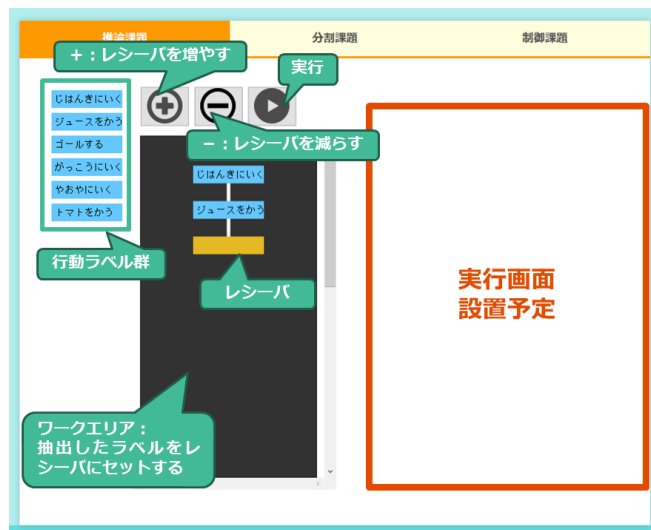


図4 推論課題



図5 分割課題



図6 制御課題

4.4.3. 制御課題

図6に開発段階の制御課題を示す。目的のプログラムを作成するために、ブロック状のコードを用いて自由に組み合わせていく。実行画面については、Blockly

Games における「迷路」を参考にした[15]. これは、「迷路」の出力結果において、組み上げたコードが直接的に探索の挙動として反映されることから、論理的推論が行いやすいためである.

一般化においては、関数ブロックおよび、ループブロックの内包回数や、一般化したブロックの正答の割合から評価する、制御においては、コードブロックの操作の他、組み上げたコードについてそれぞれ最適解との比較を行い、一致度の割合について評価する.

4.4.4. 分析課題

推論課題、制御課題における実行ボタンを押すことにより実行画面が作動し、それに対して修正を行う際（トライ&エラー）の操作を捉える. 定義 1 は、エラーの解決時の操作数を対象に、再び実行ボタンを押すまでのボタン、ブロック等の操作回数、2 はトライ&エラーを終えるまでの試行回数や解決数を抽出する.

5. おわりに

本稿ではプログラミング学習中に行われる思考「プログラミング的思考」について、観測できる操作として展開し、定量的な評価が行える学習環境「Thinkron」を提案した. Thinkron は学習者のプログラミング的思考の育成を行うため、個々の形成的な評価を基にプログラミング的思考の苦手部分を洗い出し、教師の提供する問いと Thinkron の思考課題によって、着実にプログラミング的思考を育むことを想定している. そのため、Thinkron はコーディング中の一部の思考を切り出すことはせず、図 3 の通りあくまでも学習フェーズという設計段階やトライ&エラーも含んだプログラミング全体の流れから、思考活動の抽出を行っている. 単純に学習者にプログラミングを行わせて思考を抽出する場合、必ずしも目的のプログラミング的思考の構成要素が育まれるとは限らず、取り上げる題材によっては各構成要素が反映されたものを作成していくのは難しい. またそれらの要素が含まれているのかどうかを確認するのも困難である. だからこそ、一つ一つの構成要素に焦点を当てた操作を行う課題を持ち、それらについて可視化を行う Thinkron は、プログラミング的思考を育むことに特化した学習環境であると言える.

本研究における学習環境は開発段階であるため、GUI や操作性に関しては議論を重ねた上で適宜改善していく必要がある. また、定量化においても意図した操作が取れるかどうか、検証を基に再度設計を行っていく. 今後の展開としては、Thinkron の学習者用フォーム（学習フェーズ）を完成させ、思考課題が目的の思考活動を捉えるかどうか検証を行う. その後、学習者に対して Thinkron を用いたプログラミング学習を

実施し、思考課題から得た定量化データより、思考活動における評価指標を作成する.

本研究が完遂することにより、プログラミング的思考の評価規準としての役割を担うと同時に、課題による思考の発現の程度や成長の推移の可視化が容易になることが期待される.

文 献

- [1] 文部科学省, “小学校 学習指導要領 (平成 29 年告示), https://www.mext.go.jp/content/1413522_001.pdf, 参照 Aug.3,2020.
- [2] 文部科学省, “小学校段階におけるプログラミング教育の在り方について (議論の取りまとめ), https://www.mext.go.jp/b_menu/shingi/chukyo/chukyo3/053/siryo/_icsFiles/afieldfile/2016/07/08/1373901_12.pdf, 参照 Aug.3,2020.
- [3] 小泉力一, 小田理代, 後藤義雄, 星千枝, 永田衣代, “小学校段階におけるプログラミングで育成する資質・能力の評価規準開発,”教育システム情報学会,42,pp.435-436,(2017).
- [4] Benesse, “プログラミングで育成する資質・能力の評価規準 (試行版), <https://benesse.jp/programming/beneprog/wp-content/uploads/2018/08/ver2.0.0.pdf>, 参照 Aug.3,2020.
- [5] 西野和典, 田中太志朗, 近藤秀樹, 山口真之介, 大西淑雅, “プログラミング的思考を評価するルーブリックの作成とその試用,”教育システム情報学会,43,pp.141-142,(2018).
- [6] 文部科学省, “小学校プログラミング教育に関する研修教材, https://www.mext.go.jp/a_menu/shotou/zyouhou/detail/1416408.htm, 参照 Aug.3,2020.
- [7] 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, 参照 Aug.3,2020.
- [8] 大村基将, 紅林秀治, “ソフトウェア設計に基づく初学者のためのプログラミング学習の過程に関する考察,”教科開発学論集 (4), pp.151-159, (2016).
- [9] 大日本印刷, “小学校で求められるプログラミング的思考を育む教材ソフト DNP プログラミング教材ソフト「SWITCHED ON Computing 日本版」, https://www.dnp.co.jp/biz/solution/products/detail/1192363_1567.html, 参照 Aug.3,2020.
- [10] Matayoshi Y., Nakamura S., “Abstract Thinking Description System for Programming Education Facilitation,”HCII 2020,vol.12206,pp.76-92,(2020).
- [11] 太田剛, 森本容介, 加藤浩, “プログラミング能力の発達段階と要因に関する定量的分析,”情報教育シンポジウム,pp.85-92,(2017).
- [12] 國宗永佳, 中林清, “プログラミング導入演習に対して学習支援システムと自己調整学習が与える影響の分析,”通信ソサエティマガジン,Vol.50,秋号,pp.100-109,(2019).
- [13] “p5.js, <https://p5js.org/>, 参照 Aug.3,2020.
- [14] “Blockly, <https://developers.google.com/blockly>, 参照 Aug.3,2020.
- [15] “Blockly Games, <https://blockly.games/?lang=ja>, 参照 Aug.3,2020.