

視覚的なクエリからのプログラム例示による創作支援

立命館大学情報理工学部協調メディア研究室

2020年9月

初等中等教育におけるプログラミングの授業やワークショップにおいては、学習者自身にとって意味があり、興味を持ったプロジェクトに取り組むことが重要である。本論文では、そのような学習活動を支援するための環境として、プログラムの動作を視覚的に表現することによってプログラム例を検索できるような機能を提案し、実際のワークショップに適用することによって得られた知見について報告する。本環境の利用によって、例示されたプログラムを模倣したり、多少の改変を加えたりして、新たなプロジェクトの創作を支援できている様子が伺えた。また、視覚的クエリを用いた検索に関しては、児童が興味を持って使用していたこと、プログラミングを教える側と児童のコミュニケーションを助けるような場面があることがわかった。その一方で、プログラム例を改変して組み合わせるような創作を支援できなかった場面や、プログラムの実行結果を示す視覚的クエリの表現が理解しづらい場面があるという課題が残った。

In programming classes and workshops in primary and secondary education, it is important for learners to work on projects of their own interest. As an environment to support such learning activities, we propose a function that enables children to search program examples by visually expressing the program execution scene, and report our findings obtained by applying it to actual workshops. By using this environment, children were able to be supported in creating new projects by copying the program examples and making some modifications to the program examples. As for the search function using the visual query, we found that not only the children used it with interests, but also it sometimes helped the children to communicate with the workshop supporters. On the other hand, we noticed that the environment could not support the creation with modification and combination of program examples. We also discovered that it was sometimes difficult for children to understand the expression of visual queries indicating the program execution.

1 はじめに

日本における初等・中等教育でのプログラミングの必修化や、世界におけるSTEM教育の広がりなどに伴って、児童を対象にしたプログラミング学習が注目を集めている。児童を対象にしたプログラミングの授業やワークショップ自体は従来から様々な取り組みが行われているが、その目的や手段が確立されているとは言い難い。現在、児童・生徒向けのプログラミング環境に対して広く利用されているものにScratch (Scratch 2020)があるが、Scratchを活用してどのような授業やワークショップを実施するのか、また、そのような授業やワークショップを支援するための環境としてどのようなものが必要なのかを検討していかなければならない。

児童向けの授業やワークショップでは、プログラミング能力自体そのものを育成するのではなく、プログラミングを通して仕事や日常生活で役立つ様々なスキル、すなわちコンピューショナルシンキング (Wing 2006) を身につけることを目的にしている場合が多い。Scratchでは、児童が興味があり意味があると考えられるプロジェクトに取り組むことにより、自ら問題を発見し、それを解決していく素養を習得していくことを重要視し

ている。すなわち、与えられた課題を達成するだけでなく、自ら取り組みたい課題を見つけ、それに組み込んでいくというプロセスを授業やワークショップでどこまで実現できるかが重要である。

このようなプロセスを実現するに当たっては、プログラミング能力自体の育成を目的としないものの、児童が取り組んでいる課題を達成するためのプログラムを記述できるような支援を行う必要がある。そこで、本研究では、プログラムの例示を行うことによって創作を支援する環境の構築を行い、実際のワークショップに適用することによって評価を行う。本環境の特徴は、Scratch等のプログラミング環境に見られる、プログラムの実行結果が画面上の物体の動きとして表現されるという特性を利用し、児童が参考にしたいプログラムの動きを、文字ではなく、視覚的なクエリによって表現できるという点である。児童は、例示されたプログラムを参考にして、それを再利用あるいは拡張することによって自らのプロジェクトを創作できるようになることを期待している。

本論文の構成は以下のとおりである。まず、2章ではプログラミング学習に関する取り組みについて概観し、プログラミングを支援する立場の関連研究を紹介する。

3章では、本研究で提案するプログラム例示による創作支援を行う環境について述べる。4章では、本環境をプログラミングワークショップに適用して得られた知見を示す。5章では、本論文のまとめと今後の課題について述べる。

2 関連研究

2.1 プログラミング学習と創作

初等教育や中等教育において、日本ではプログラミング教育の必修化を実施している（文部科学省 2016）。また、イギリスやオーストラリアでもプログラミング教育が必修化され、児童を対象にしたプログラミング学習が広がっている（e.g., Department for Education 2013 ; 太田剛ほか 2016）。さらに、アメリカでの Code.org (Code.org 2020) という団体の活動を例に児童がプログラミングを用いて学習するという動きがみられる。それぞれの学習では、プログラミングを通して思考力を育てることを重要としている。思考力としては先に述べた「コンピューショナルシンキング」をあげている場合が多い。コンピューショナルシンキングとは、コンピュータを使った問題解決だけでなく、身の回りの問題解決にも応用できる考え方であると示されている。一見難しそうな問題を既に解き方を知っている問題に変化させることがこの思考法の特徴とされている。

初等教育におけるプログラミングで広く用いられている Scratch は、ブロックをつなげてスプライトにプログラムを作成するプログラミング環境である。Scratch を開発している MIT の Resnick らは、児童に創造的な思考を持つように成長してほしいという考えのもとで、創造的な思考を育成する上で「コーディングするために学ぶ」のではなく「学ぶためにコーディングする」べきであると指摘している（Resnick and Robinson 2017）。また、創造的な思考の必要性は Papert の構築主義という考え方からも Resnick は説明しており、構築主義とは児童は自ら進んで大事なものを創っているときに最もよく学ぶとする理論である。

これらを踏まえると、学習者はプログラミングを用いて興味があり意味があると考えられるプロジェクトに取り組むことが重要である。児童を対象にした授業やワークショップで児童が効率よく学習するためには、児童に興味をもたせるような内容を実施し、指導者に強いられなくても児童自身がモチベーションをもって学習できるような環境を用意すべきである。

2.2 初学者を対象にしたプログラム例示

初学者のためのプログラミング学習支援を目的として、iSnap によるヒントアルゴリズム (Price et al 2017) や、Worked Example に関する研究 (Zh et al 2019)

がある。これらはプログラムを例示するための手法を提案しており、授業やワークショップで提示される課題に対して問題解決の支援を行っている。これらの研究で初学者が取り組む課題として挙げられている学習内容は、条件分岐や乱数といったプログラミングの概念である。これらの課題に取り組むことにより、初学者がプログラミングに必要な概念を学んでいるといえる。しかし、学習者自身の興味によるプロジェクトに取り組んでいるとは限らず、モチベーションを持って学習できているとは必ずしもいえない。

そこで、本研究では提示された課題を解決していく過程で学んだプログラムを利用し、学習者自身の興味のある新たなプロジェクトを創作すべきであると考えられる。新たなプロジェクトを創作する際には、学んだプログラムを利用するために学習者がプログラムを再度確認することが想定される。したがって、本研究では、学習者がプログラム例を確認することを支援する。すでに学んだプログラムを再度確認できるようになることで、プログラム例を用いた再利用や拡張を含む創作を児童が行いやすくなると思われる。

3 プログラムの例示による創作支援

本章では、視覚的なクエリによるプログラムの例示を行う手法について述べる。

3.1 ユースケース

創作活動における本環境を用いたユースケースについて述べる。ユーザは授業やプログラミングワークショップで与えられた課題に取り組んだ後、新たなプロジェクトの創作を始める。新たにプロジェクトを作成する中で、与えられた課題で示されたプロジェクトに含まれるプログラム例を調べて、プログラム例を自らのプロジェクトに利用するという場面を図1に示す。プログラム例の実行結果として現れるスプライトの絵や動きの情報を手掛かりに、ユーザは本手法によるクエリを入力し検索を行う。検索結果によりプログラム例が表示され、新たに作成しているプロジェクトにプログラム例を利用する。

3.2 視覚的なクエリ

ユーザが学習した課題から特定のプログラムの例を探し出すための視覚的なクエリについて説明する。図2に示すように、視覚的なクエリとは、課題で学習したプログラムに含まれているスプライトの画像であるコスチュームや動きを視覚的な情報で示す絵として扱い、プログラムの実行結果を表現するものである。ユーザが利用したいプログラム例を検索する場合、「猫がすすむ」「もし、猫がネズミにあたると、ゲームが止まる」という視覚的な情報から検索を行うことが想定される。



図1 ユースケース

動作内容	視覚的クエリ
スプライトの動作	
特定の動作をきっかけとするスプライトの動作	
スプライト同士がきっかけとなる動作	
特定のスプライトと動作をきっかけとする、スプライトの動作	

図2 クエリー一覧

まず、これらの視覚的な情報を静的な絵として表現するため、「猫がすすむ」という実行結果であれば、「猫」「すすむ」という視覚情報に分ける。また、「もし、猫がネズミにあたると、ゲームが止まる」であれば、「猫」「ネズミ」「あたる」「ゲームが止まる」に分ける。

次に、分けた視覚的情報はスプライトのコスチュームを示すものとスプライトの動きを示すものに分類する。もし、「猫がネズミにあたると、ゲームが止まる」を例にあげると、「猫」「ネズミ」がスプライトのコスチュームで、「あたる」「ゲームが止まる」が動きである。分類されたスプライトのコスチュームを示す絵は課題のプログラムに使用されるスプライトのコスチュームの画像を使用する。スプライトの動きの視覚的情報を示す絵はプログラミング環境で使用されているアイコンやユニバーサルデザインである絵文字、もしくは、それらを参考に作成した絵で表現する。プログラム例の実行結果を示すための視覚的クエリには、図2の例に示すように以下の4パターンを定義する。

- スプライトの動作
スプライトが方向転換を行うという内容を示して

いる。

- 特定の動作をきっかけとするスプライトの動作
スプライトがクリックされると色が変わるという内容を示している。
- スプライト同士がきっかけとなる動作
スプライトがもう一方のスプライトに触れると、ゲームが止まるというものである。
- 特定のスプライトと動作をきっかけとする、スプライトの動作
スプライトがクリックされると他のスプライトの色が変わるというものである。

3.3 クエリを用いた検索

3.3.1 検索ページ

検索ページを図3に示す。検索ページでは、ユーザが利用したいプログラムの実行結果を示したクエリを入力しプログラム例の検索を行う。まず、利用したいプログラム例が課題で学習したどのプロジェクトに含まれているのかを示すために、プロジェクトを選択する。検索ページには、プロジェクトが動作する様子を示す動画が添付されており、ユーザは利用するプログラム例を含むプロジェクトを確認できる。他のプロジェクトに含まれるプログラム例を利用したい場合は、プロジェクトを示す動画の下部にボタンが配置されている「プロジェクトをかえる」というボタンを押すことで他のプロジェクトを選択できる。

ユーザは「スプライト」や「うごき」と示されているタグの下部にある入力欄にクエリを記述するための絵を入力する。入力欄をクリックすると、選択したプロジェクトに含まれるプログラム例を示す絵の一覧が表示されるので、絵を選択し、入力欄に挿入する。

入力欄に絵が挿入され、記述されたクエリが本環境で用意しているプログラム例を示すクエリと一致した場合、画面下部に配置した検索ボタンが赤くなる。赤い状

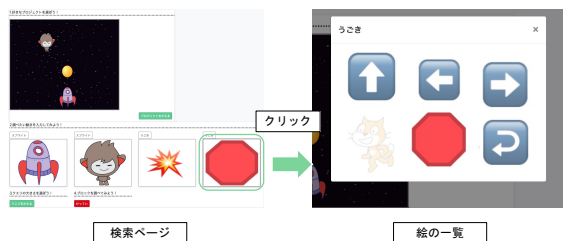


図3 検索ページ

態で検索ボタンを押すと、検索結果ページに遷移し、プログラム例が示される。また、画面下部の「クエリをかえる」というボタンを押し、クエリ数を選択することにより、クエリに用いる絵の数を変更できる。

3.3.2 検索結果ページ

検索結果ページを図4に示す。検索結果ページでは、入力されたクエリが示すプログラム例が動作している動画と、プログラム例が示される。ユーザは動画を閲覧することにより、利用したいプログラムの動作であるかを確認できる。示されるプログラム例は、単体で動くものも一部あるが、どのようにプロジェクトに利用するのかの組み合わせを考えなくてはならないものがほとんどである。これは単に模倣するのみではなく、プログラムの再利用・拡張による創作を通してプログラムがどのような意味をもつのかをユーザが学習する必要があると考えるためである。また、検索結果ページには戻るボタンがあり、ボタンを押すと検索ページが表示され、再びクエリを入力してプログラム例を検索することができる。

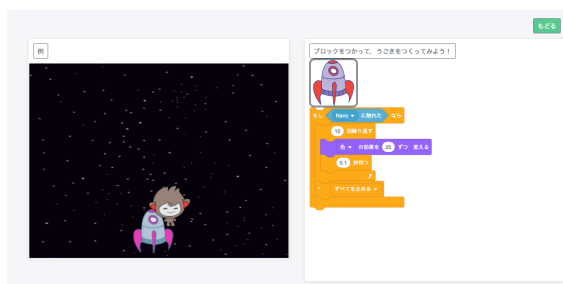


図4 検索結果ページ

4 実践による評価

視覚的クエリを用いたプログラム例示による創作支援の効果を実際のワークショップに適用し、検証した。本章では実践の結果と考察について述べる。

4.1 評価基準

本環境によるプログラム例示による創作支援の効果を以下の2つの基準から評価した。

- 視覚的クエリがプログラムの動作結果を示せていたか。
 - － 児童がスプライトとスプライトの動きを示す絵を入力し、プログラムを検索可能であったか。
- プログラム例示により児童が新たなプロジェクトを作成できていたか。
 - － 本環境が示すスプライトのプログラム例を写し、模倣のみを行なったか。
 - － 本環境が示すスプライトとは違うスプライトにプログラム例を再利用していたか。
 - － 児童が作成していたプログラムとプログラム例を組み合わせてプログラムを拡張できていたか。

本環境はプログラム例とともに、課題で示されているプロジェクト内でプログラム例が利用されていたスプライトのコスチュームも示している。そのため、プログラム例を模倣するのみであれば、プロジェクトが示すコスチュームをもつスプライトにプログラムを作成することになる。児童の興味のあるプロジェクトに改変する場合、プログラムを異なるスプライトのコスチュームに再利用する場面が想定される。さらに、課題で学習したプロジェクトとは違う新たなプロジェクトを作成するためにプログラム例を引用したい場合や、プログラム例を組み合わせて新たなプロジェクトを作成したい場合において、プログラムを拡張する改変が想定される

4.2 実践方法

本環境の効果を検証するために、NPO 法人スーパーサイエンスキッズ（スーパーサイエンスキッズ 2020）が主催するプログラミングワークショップで適用を行った。ワークショップは月に2度程度開催されており、小学生から中学生を対象にしている。ワークショップではScratchがプログラミング環境として採用されている。通常、ワークショップでは、児童は3時間のプログラミングを用いた制作を行う。3時間の制作では、まず、講師が課題として例示したプロジェクトの模倣を行い、プログラミング環境の操作方法やプログラムの作成方法を学んでいく。その後、自由制作として、児童が学習内容を活用し、新たなプロジェクトを作成する。

本実践では、自由制作の機会を利用し、本環境についての効果を検証した。ワークショップ開始から2時間はワークショップのテーマに沿って用意されたプロジェクトを講師の指導のもと児童は模倣した。その後、本環境の機能と操作について児童に15分で説明した。説明内

容として、本環境で検索を行いながらプロジェクトの拡張・再利用を行なって新しいプロジェクトを作成してほしいという趣旨を児童に伝えた。説明が終了後、児童は本環境を使用し、1時間の自由制作を行った。

実践の実施日は、2019年11月10日、12月22日の2日間である。それぞれのワークショップは、1日目はシューティングゲーム、2日目は宝探しゲームを制作するというテーマで開催された。そのため、1日目には「猫から逃げるゲーム」「シューティングゲーム」、2日目は「宝探しゲーム」といったプロジェクトに取り組んだ。1日目で制作した「猫から逃げるゲーム」は「シューティングゲーム」を制作するための予備知識を説明するためのプロジェクトである。本環境はそれぞれの日程で制作したプロジェクトのプログラムを例示できるように、プロジェクトに含まれるプログラム例とプロジェクトの実行結果を示す動画、プログラム例に対応するクエリ、および例示されるプログラム例を用意した。

2日間で行った実践では1日目に9人、2日目に6人の児童が参加した。1日目のワークショップでは、全員が本環境を利用した。一方、2日目のワークショップでは、参加者は12人であったが、提供するシステムの都合により、本環境を利用したのは6人であった。いずれの実施日にもScratchを用いたプログラミング経験がある児童とない児童が含まれていた。また、ワークショップには情報理工学を専攻する大学生がサポーターとして派遣されており、児童がわからないことがあれば講師またはサポーターが質問に対しすぐに対応できる環境であった。児童1人に対し、本環境を利用できるタブレット端末と、プロジェクトを作成するPCを用意した。評価のために、制作過程がわかるように10分おきにプロジェクトの保存を行い、本環境を利用した検索ログと児童が制作している様子の動画を取得した。

4.3 実践結果

ワークショップでの本環境の適用をふまえ、作成されたプロジェクトと検索ログ、児童の様子を示す動画から得られた知見について述べる。

4.3.1 児童の様子

動画からみられた児童の様子について述べる。本環境による検索を行い、プログラム例を閲覧している様子が見られた。児童はクエリの入力、プログラム例の閲覧を行うことに対して興味を示す場合が多かった。プログラム例を閲覧してプログラムを模倣している児童もみられた。児童がプログラム例を表示した本環境のページを閲覧しながら、プログラム例を自身のプロジェクトへ組み合わせられないと質問するといったサポーターとのコミュニケーションもみられた。創作を開始する前に行った本環境の操作説明以外の場面では本環境を利用しな

かった児童も見られたが、その児童はサポーターに質問していたり、制作を行っていたりするなどの積極的な行動をとっていた。

4.3.2 プログラム例の創作への活用

1人の児童（以降、児童Aとする）が本環境をどのように利用したのか、検索ログと作成されたプロジェクトの経過から状況を説明する。児童Aが使用していた端末の検索ログから、24回検索ボタンを押し、21回検索結果の表示を行なったことが確認された。

図5に示すプログラムを作成した児童Aの1時間の制作過程を時系列で説明する。まず児童Aは「敵がボールに当たると敵が消える」「ボールがまっすぐ動く」というプログラム例を示す検索結果ページを閲覧した。検索結果ページが示すどちらのプログラム例も自身のプロジェクト上に模倣できていた。しかしながら、模倣できたプログラムの一つである「敵がボールに当たると敵が消える」というプログラムは、変更が足りず、実行可能ではなかった。また、もう1つの「ボールがまっすぐ動く」というプログラムは、変更が加えられたスプライトのコスチュームに使用され、実行可能であったが、課題で学習したプロジェクトを再現するためにはボールを初期位置に戻すというような変更が必要であった。これらのプログラムの作成後、児童Aは作成したプログラムをすべて消去していた。次に「敵が横に動く」「敵が方向転換する」といったプログラムの検索を行い、表示されたプログラム例を組み合わせることにより、学習したプロジェクトで示した特定のスプライトの動きを再現できていた。最後に再び「敵がボールに当たると敵が消える」というプログラム例を閲覧したが、模倣のみで実行可能なプログラムまで拡張することはできなかった。

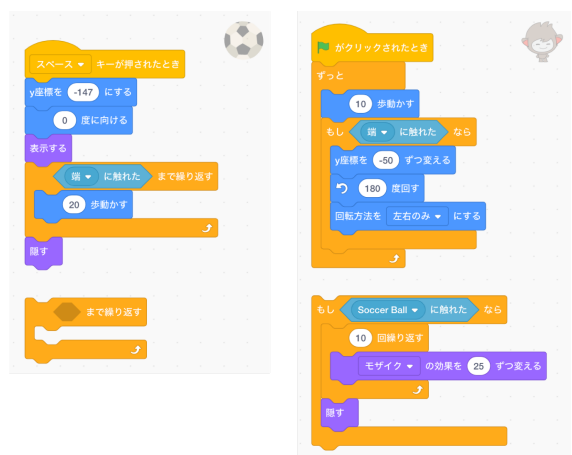


図5 児童Aが作成したプロジェクト

これらから、児童Aは例示されたものとは違うスプ

ライトを使用してプロジェクトに改変を加えようという意図が読み取れた。児童 A には、プログラム例の模倣、違うスプライトへの例示プログラムの再利用、および、プログラム例を組み合わせるような拡張がみられた。一方で、改変や組み合わせが必要なプログラム例を利用できなかった場面も確認できた。

他の児童にも同じく、プログラムを写したりコスチュームを改変したりした利用が多くみられ、拡張も少数ではあるがみられた。実践中には、プログラムに含まれるブロックの値の変更や、プログラムの組み合わせについてなど、プログラムの拡張に関する児童からの質問が多かった。

4.3.3 視覚的クエリによるプログラム例の検索

検索ログからみられた検索結果ページの表示回数を表 1 に、検索ボタンのクリック数を表 2 に示す。本環境の操作説明においてクエリによる検索を 2 回促したが、ほとんどの児童は自身でさらに検索を行っていたことがわかる。2 回目の実践においては、本環境の利用が減少するという結果となったが、視覚的クエリを構成するほとんどの絵が検索に利用されていた。

表 1 検索結果ページの表示回数

	平均	最大値	最小値
1 日目	8.8	21	3
2 日目	4.7	11	2

表 2 検索ボタンのクリック数

	平均	最大値	最小値
1 日目	13.3	25	4
2 日目	8.8	20	5

4.4 考察

4.4.1 創作支援

今回の実践では、児童がプログラム例を模倣する場面や、プログラム例が示すスプライトのコスチュームに改変を加えプログラム例を模倣していた場面、プログラム例を組み合わせる拡張するという場面がみられた。これにより、児童のプログラム例を利用した創作を本環境がある程度支援できたことがわかる。しかし、プログラム例を拡張するといった創作支援については、児童がプログラムを拡張できなかった場面もみられた。具体的な場面としては、ブロック内の数値を変えなければならない場面や、プロジェクトが示すスプライトの一連の動きを再現するためにプログラムを組み合わせなければならない場面である。このような場面では、プログラム例を模

倣しただけでは目的の動作をしないために、児童が自身のプロジェクトへのプログラムの利用方法を見出すことができなかった。本研究では、プログラムを模倣し組み合わせることが学習として必要であると考え。しかし、プログラム例の利用方法が難しすぎると、児童が興味をなくすことが考えられるため、プログラムの拡張が難しいと児童が判断し、自身のプロジェクトへの利用ができない場合、サポーターなどの人や端末がヒントを児童に与えるような補助が必要である。

4.4.2 視覚的クエリを用いた検索

視覚的クエリを用いた入力に関しては、ほとんどの児童が操作説明以外でもクエリを入力し、ワークショップ後半も継続して使っていたため、本環境の絵を選ぶという操作性に難しさはなく、興味を持たせるものといえる。また、児童が入力したクエリからプログラム例を表示させて利用していたことから、児童が利用したいと想定していた実行結果に近いものをクエリで表現できたと考えられる。

2 日目における検索回数が 1 日目に比べ少ない結果となったことについては、児童のプログラミング能力や、実践したプロジェクト、プロジェクトに用意したクエリなどによる条件の違いが要因として考えられる。児童のプログラミング能力に関しては、2 日目の方がプログラミングワークショップ参加経験者が多かったが、Scratch を操作することの経験が大きく影響しているとは考えられなかった。また、各実践日のグループ間で年齢に大きく差があったわけではなかった。実践したプロジェクトのテーマに関して、1 日目のテーマである「猫から逃げるゲーム」「シューティングゲーム」が 2 日目の実践で実施した「宝探しゲーム」より少ないスプライトに複雑なプログラムを作成するようなものになっていた。このことから、プロジェクトの複雑さから、児童に検索の興味を与えたことが考えられる。プロジェクトに用意したクエリに関しては、2 日目のテーマに比べて 1 日目のテーマではそれぞれの視覚的クエリが示す実行結果が他の視覚的クエリと類似しないものが多かった。そのため、1 日目の実践では児童が視覚的クエリを区別でき、検索回数の増加につながったことが考えられる。つまり、類似するようなプログラムの実行結果があった 2 日目の実践では、被験者にとって視覚的クエリが区別しにくかったと想定される。

4.4.3 学習に関連するコミュニケーション

本環境の検索結果ページのプログラム例を閲覧しながら児童がサポーターに質問する様子がみられたことから、児童が学習したプロジェクトが含むプログラムのどの点が理解できていないのかを本環境を用いて示せていることがわかる。検索ページでも検索したい内容の一部

を絵で示すことで、視覚的クエリを構成する絵がサポーターに質問する手掛かりになるような場面もみられた。

これらのことから、一度学習したプログラムのわからない点を質問するという再学習の効果や、言葉でうまくサポーターに伝えられないときにコミュニケーションの手掛かりになるという効果があったことがわかる。

5 おわりに

本論文では、プログラミング学習において、与えられた課題を解決するのみではなく、課題で示されたプログラムを利用した創作の必要性をあげ、視覚的なクエリからのプログラム例示による創作支援について述べた。また、本研究ではプログラミングワークショップにおいて効果の検証を行なった。その結果、課題に含まれるプログラム例を利用し、プログラム例を写したりスプライトのコスチュームに改変を加えたりするような新たなプロジェクトの創作をある程度支援できたことがわかった。一方、プログラムを組み合わせることによるプログラムの拡張は児童が創作したプロジェクトにみられたが、プログラムの組み合わせを行えない場面も確認できたため、プログラム例を提示する手法に改善が必要である。

今後、これらの効果の検証をすすめるために、さまざまなプログラム例やクエリを準備し、今後も継続して検索機能の利用を続ける。

参考文献

- 文部科学省 (2018) 小学校プログラミング教育の必修化に向けて. https://www.mext.go.jp/b_menu/shingi/chukyo/chukyo3/004/siryo/_icsFiles/afiefieldfile/2018/10/05/1409851_6.pdf (accessed 2020.01.18)
- Department for Education (2013) National curriculum in england: computing programmes of study. http://www.mext.go.jp/b_menu/shingi/chousa/shotou/122/attach/1372525.htm (accessed 2020.01.18)
- 太田剛, 森本容介, 加藤浩 (2016) 諸外国のプログラミング教育を含む情報教育カリキュラムに関する調査. 日本教育工学会論文誌, 40(3), 197-208.
- Code.org (2020) <https://code.org/> (accessed 2020.01.18)
- Wing, J. M (2006) Computational thinking. Communications of the ACM, 49(3), 33-35.
- Scratch (2020) <https://scratch.mit.edu/> (accessed 2020.01.18)
- M. Resnick and K. Robinson (2017) Lifelong Kindergarten: Cultivating Creativity Through Projects,

Passion, Peers, and Play. The MIT Press

Price, T. W., Dong, Y., and Lipovac, D. (2017) iSnap: towards intelligent tutoring in novice programming environments. In Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education, pp.483-488

Zhi, R., Price, T. W., Marwan, S., Milliken, A., Barnes, T., and Chi, M(2019) Exploring the impact of worked examples in a novice programming environment. In Proceedings of the 50th ACM Technical Symposium on Computer Science Education, pp.98-104

スーパーサイエンスキッズ (2020) <http://www.npo-ssk.com/> (accessed 2020.01.18)