

Scratchにおける再利用作品のプログラム差分分析

橋谷 直樹^{1,a)} 伊原 彰紀^{1,b)}

概要: ビジュアルプログラミング言語 Scratch は、リミックスと呼ばれる機能を用いることで作品を再利用でき、新たな作品を制作することができる。しかし、ユーザに対するリミックスの有用性について明示でなく、リミックスすることで学習につながる作品を特定することは容易でない。本論文では、変更内容と変更量をリミックスすることの学習効果の違いを再利用された作品に分析する。

1. はじめに

初等教育のプログラミング教育における学習環境として Scratch^{*1} の利用が進められている。Scratch は、「命令を出す」、「イベントを受け取る」などの命令処理をブロックによって視覚的に表現し、ユーザはそれぞれのブロックを組み合わせることで直感的にプログラムを作成して作品制作することができる。ユーザは、自身が制作した作品を Scratch の Web サービス上に公開することができ、また他のユーザが製作した作品のプログラム内容の確認、実行、評価することができる。

ユーザは作品制作を通して、プログラム作成に必要な思考であるコンピューテーショナル・シンキング (CT) スキル [1] を習熟する。表 1 に CT スキル 7 概念とそれぞれの評価基準を示す。Scratch を利用するユーザは、ゲーム、アニメーション、アートなどの多様なカテゴリの作品を自由に制作する過程で CT スキルを習熟し、学習する。しかし、初学者などのユーザは作品制作が困難であるため、多様な作品制作を手助けするために「リミックス」と呼ばれる機能がある。リミックスは、公開された作品を複製し、プログラムに対して処理の追加や削除を行うことで新たな作品として制作することができる。現在、サービス上に公開されている 30 % 弱の作品^{*2} はリミックスした作品であり、多くのユーザはリミックスを頻繁に利用し、作品制作に取り組んでいる。従来研究では、ユーザのリミックスをきっかけに多様なブロックを使うようになり、学習効果が

あることを明らかにしている [2] [3]。しかし別の従来研究では、プログラミング概念や基本パターンの使用に繋がらず CT スキルの獲得に至っていないことを定量的に示している [4]。従って、ユーザのリミックスを行うことの学習について明示でなく、リミックスがユーザの作品制作に与える学習効果を分析して、より CT スキルなどが習熟できるようにリミックスの価値ある作品を特定する必要があると考える。作品をリミックスしたユーザは、作品のプログラムを編集 (追加・削除・変更) することで作品制作に取り組むため、プログラム内容を理解する必要があると考える。それ故に、リミックスしたユーザの理解度 (学習度) によってプログラムの変更内容および変更量が異なると推測される。本研究では、リミックスすることによる学習効果についてプログラム変更量の観点からリミックス元とリミックス先の作品を分析する。具体的には、リミックス元となる作品とリミックスした作品のプログラムの差分がリミックス後に制作した作品の CT スキル獲得などに影響を与えるか否かを分析する。

2. Scratch における再利用作品のプログラム差分分析

本論文では、CT スキル概念を用い学習効果を測るため、Scratch3.0 をリリースした 2019 年 1 月 3 日以降に Scratch で公開した作品を収集し、その中からリミックスされた作品 11,080 件とその作品をリミックスした作品 54,989 件およびリミックスを行ったユーザが制作した作品を分析対象とする。目的を達成するために、3 つの RQ を設定した。

RQ1: リミックスされた作品のプログラムはどの程度変更されるのか

RQ2: ユーザがリミックスする前と後によってプログラム習熟度に違いはあるのか

RQ3: リミックスした作品のプログラム変更量によって

¹ 和歌山大学
Wakayama University, Wakayama 640-8510, Japan

^{†1} 現在、和歌山大学
Presently with Wakayama University

^{a)} s226329@wakayama-u.ac.jp

^{b)} ihara@wakayama-u.ac.jp

^{*1} Scratch: <https://scratch.mit.edu/>

^{*2} リミックスの割合: <https://scratch.mit.edu/statistics/>

表 1 CT スキル 7 概念の評価基準

概念	抽象化 (Abstraction)	並列 (Parallelism)	論理 (Logic)	同期 (Synchronization)	フロー制御 (Flow Control)	ユーザ対話性 (User Interactivity)	データ表現 (Data Representation)
評価基準	機能単位に分割	並列処理の使用	論理制御の使用	待機処理の使用	反復処理の使用	ユーザの入力動作の使用	変数の使用

習熟度に違いがあるのか

RQ1 では、リミックスした作品とリミックス元となる作品のプログラム差分を計測し、どの程度プログラムを変更しているのかを明らかにする。RQ2 では、ユーザがリミックスすることでリミックス後に作成した作品の CT スキル、使用ブロック種類数、使用ブロック数が増加するか否かで学習効果を明らかにする。RQ3 では、RQ1 で明らかにしたリミックス作品のプログラム差分と RQ2 で明らかにした学習効果は相関があるのかを調査し、プログラムの変更量を用いることでリミックスすることで学習価値のある作品を特定することが可能か否かを分析する。

2.1 RQ1：リミックスされた作品のプログラムはどの程度変更されるのか

Scratch における再利用作品のプログラム差分を取得するために、本節では Scratch における再利用であるリミックスを用いた作品とリミックス元となる作品を比較し分析する。リミックス元の作品（親）とリミックスした作品（子）のプログラム差分を取得するために、本論文では編集距離を用いる。編集距離は、とある文字列 A と文字列 B がどの程度異なっているかを表す距離の指標で、1 文字の挿入・削除・置換の操作を何回行うことで文字列 A が文字列 B と等しくなるかを示す。本論文では、Scratch プログラムを木構造に変換し、木構造の編集距離を用いて Scratch プログラムである 1 ブロックの挿入・削除・置換の操作を何回行うことでプログラムが等しくなるかによってプログラム変更量を取得可能である。編集距離の算出後、値が 1 以上のプログラムを対象に、どのようなプログラム変更が行われたか調査することでプログラム差分を把握する。

期待される結果：リミックス作品のプログラム差分を明らかにする

2.2 RQ2：ユーザがリミックスする前と後によってプログラム習熟度に違いはあるのか

ユーザが作品を再利用したことで Scratch におけるプログラムの変更量による学習効果を分析するために、本節ではリミックスしたユーザがリミックス前後に作成した作品を分析する。具体的には、作品をリミックスしたユーザがリミックス後に作成した作品（オリジナル作品）を対象に、CT スキル・使用ブロック種類数・使用ブロック数が向上したか否かをリミックス前に制作した作品と比べて比較する。CT スキルの評価には、作品評価ツール Dr.Scratch [5] を用いて、表 1 の 7 概念のスコアを 0 点から 3 点で評価

する。

期待される結果：リミックスすることでリミックスしたユーザに対して学習効果がある

2.3 RQ3：リミックスした作品のプログラム変更量によって習熟度に違いがあるのか

RQ1 で明らかにしたリミックス作品のプログラム差分に基づき、RQ2 で明らかにした学習効果との相関があるのか分析することで、ユーザのリミックスによって学習価値のある作品の特定を目指す

期待される結果：プログラム変更量とユーザの学習効果には相関があり、ユーザのリミックスの経験が CT スキルの向上に寄与するか否かを明らかにする。

3. おわりに

本論文では、Scratch において作品を再利用することによるユーザの学習効果についてプログラム変更量の観点から再利用された作品を分析する。今後は、本論文の分析結果を用いて、リミックスすることでコンピューテーションルシンキングスキル獲得などの学習につながる作品の特定手法を検討する。

参考文献

- [1] Wing, J. M.: Computational thinking and thinking about computing, *Journal of PHILOSOPHICAL TRANSACTIONS OF THE ROYAL SOCIETY*, Vol. 366, pp. 3717–3725 (2018).
- [2] Dasgupta, S., Hale, W., Monroy-Hernández, A. and Hill, B. M.: Remixing as a pathway to computational thinking, *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing (CSCW'16)*, pp. 1438–1449 (2016).
- [3] Yang, S., Domeniconi, C., Reville, M., Sweeney, M., Gelman, B. U., Beckley, C. and Johri, A.: Uncovering Trajectories of Informal Learning in Large Online Communities of Creators, *In Proceedings of the Second ACM Conference on Learning @ Scale (L@S'15)*, pp. 131–140 (2015).
- [4] Amanullah, K. and Bell, T.: Evaluating the Use of Remixing in Scratch Projects Based on Repertoire, Lines of Code (LOC), and Elementary Patterns, *2019 IEEE Frontiers in Education Conference (FIE)*, pp. 1–8 (online), DOI: 10.1109/FIE43999.2019.9028475 (2019).
- [5] Moreno-León, J., Robles, G. and Román-González, M.: Dr. Scratch: Automatic analysis of scratch projects to assess and foster computational thinking, *RED. Revista de Educación a Distancia*, No. 46, pp. 1–23 (2015).