

Scratch における Remix が行われていない 類似プログラムの特定にむけて

安東 亮汰^{a)} 伊原 彰紀^{b)}

概要：本論文では、Scratch プログラミングにおいて、再利用が可能な作品が存在するか否かを調査するために、コードクローン検出システムを使用して作品間の類似スクリプトを特定した結果、部分的な再利用が可能と考えられる作品が存在することを確認した。

Toward identifying similar program without using Remix in Scratch

1. はじめに

プログラミング言語学習環境の一つとして、MIT メディアラボが Web プログラミング学習サービス Scratch を開発している。Scratch はプログラミングの初学者が「イベントを受け取る」、「命令を出す」、「値を返す」などの命令を実行するブロックを組み合わせることで直感的に作品を制作することで学習するサービスである。ブロックは役割によって形が異なり、ブロックを組み合わせることでひとまとめたものをスクリプトと呼ぶ。また、制作した作品はサービスに公開し、他のユーザと作品を共有することが可能である。

利用者は Scratch において一から作品を制作することもできるが、他の利用者が公開した作品を複製し、機能や処理の追加、削除をすることで新たな作品を制作する Remix と呼ばれる機能をもつ。Scratch ではこれを Remix と呼ぶ。従来研究は、Remix を使用することで、異なる知識を持った他の利用者が制作した作品を理解することでプログラミング学習の効果を示している [1]。

Aivaloglou らが公開しているデータセット [2] から、Scratch プロジェクトにおいて Remix を使用している作品は約 5.2% (約 250,000 件中、約 13,000 件) のみであった。その要因として、利用者が Remix の機能を把握していない、Remix は作品の完全複製を行う機能であるが故に拡張するための作業量が多くなるなどが考えられる。

ソフトウェア開発では、開発効率の向上のために既存のプログラムからコード断片の部分的な複製が行われているが、従来研究では Scratch の作品制作におけるスクリプトの部分的な再利用について調査していない。

本論文では、Scratch2.0 を対象に作品間における類似スクリプトを特定し、部分的な再利用が可能な作品が存在するか否かを調査する。Scratch においてスクリプトを部分的に再利用する機能は存在しないことから、Scratch のサービスから 6,829 件のスクリプトを収集し、作品間の類似スクリプトの存在を確認する。

2. Scratch の類似スクリプトの特定

2.1 データセットの収集

Scratch2.0 は作品のページから Scratch 独自の拡張子 (.sb2) で作品のプロジェクトファイルをダウンロードすることができる。プロジェクトファイルは、作品内で使用されているスクリプト (命令処理が記述されたプログラム)、画像、音声データなどを含む。本論文は、類似するスクリプトを特定したのちに、再利用されている可能性のあるスクリプトか否かを目視で確認するため、Scratch2.0 において公開されているデータセットから無作為に抽出した 463 件 (信頼区間 95%) のプロジェクトファイルをダウンロードする。

2.2 類似スクリプトの特定

類似スクリプトの特定には、Cordy らが開発したコードクローン検出システム NiCad [3] を使用する。NiCad は LCS (Longest Common Sequence) アルゴリズムを使用して C, Java, Python などの言語を対象に類似コードを検

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

^{a)} s216009@center.wakayama-u.ac.jp

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



(a) Remix 関係にある作品間の類似スクリプト (b) 作品内の類似スクリプト (c) 関係性のない作品間の類似スクリプト
 図 1 3 種類の類似パターン

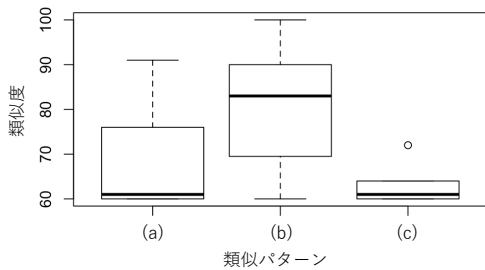


図 2 3 種類の類似度の分布図

出するシステムであり、タイプ 1, 2, 3 のコードクローンを検出可能である。本論文では、GitHub で公開されているツール Scratchfoot^{*1} を使用することでプロジェクトファイルに含まれるスクリプトを Java プログラム (function) に変換する。

変換された異なる 2 件の Java プログラムを対象に NiCad を用いて類似度を算出し、類似度の高いプログラム対の類似断片の特徴を目視で確認し、再利用された可能性のあるスクリプトであるか否かを分析する。

3. 結果と考察

本論文は、分析対象とした 463 プロジェクトが保有する 6,829 件のスクリプトを Java プログラムに変換した。6,829 件のプログラムから 2 件のプログラム対 (合計 23,314,206 件のプログラム対) を抽出し、NiCad を使用して類似度を算出した。類似度が低いプログラム対は再利用された可能性が低いため、本論文は類似度 60% から 100% を持つ 68 組の類似プログラムを特定した。

次に、類似プログラム対を 3 種類 (「Remix 関係にある作品間の類似スクリプト」、「作品内の類似スクリプト」、「関係性のない作品間の類似スクリプト」) に分類し、それぞれの種類の類似断片の特徴を目視で分析した。図 1 は各種類の類似スクリプトの実例を示す。図 2 は各種類に分類されたプログラム対の類似度の分布を箱ひげ図で示す。

(a) Remix 関係にある作品間の類似スクリプト: 約 7% (68 組の類似プログラム中、5 組の類似プログラム) が分類された。図 2 の (a) に示すように、スクリプト間は、一部のブロックの入れ替えや、引数の値のような 1, 2 行程度

の変更が多く、Remix を使うプロジェクトでは大規模な変更が加えられることはないと考えられる。

(b) 作品内の類似スクリプト: 約 81% (68 組の類似スクリプト中、55 組の類似スクリプト) が分類された。スクリプト間は、一部のブロックの入れ替え、引数の値を変更することもあるが、図 2 の (b) に示すように完全に複製されたスクリプトも存在した。Scratch ではスクリプトの一連の処理を 1 つのブロックにまとめる関数のような役割を果たす define と呼ばれるブロックが用意されている。しかしながら、define ブロックを使用可能なスクリプトであっても使われていないことが多く、同一プロジェクト内においても類似プログラムが多く存在していた。

(c) 関係性のない作品間の類似スクリプト: 約 12% (68 組の類似スクリプト中、8 組の類似スクリプト) が分類された。図 2 の (c) に示すように、作品間で部分的に再利用された可能性のある作品が存在する。今後は、Scratch における部分的なスクリプトの再利用が与える学習効果について調査する。

4. おわりに

本論文では、Scratch における作品間の類似スクリプトの有無を確認し、部分的な再利用が可能と考えられる作品を確認した。今後は、部分的な再利用にはどの程度の学習効果があるのかを調査し、部分的な再利用が可能な作品を Scratch の利用者に提供するシステムの開発に取り組む。

謝辞 本研究は JSPS 科研費 JP18KT0013 及びテレコム先端技術研究支援センター SCAT 研究の助成を受けたものです。

参考文献

- [1] 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).
- [2] Aivaloglou, E., Hermans, F., Moreno-León, J. and Robles, G.: A Dataset of Scratch Programs: Scraped, Shaped and Scored, *Proceedings of the 14th International Conference on Mining Software Repositories (MSR'17)*, pp. 511–514 (2017).
- [3] Cordy, J. R. and Roy, C. K.: The NiCad Clone Detector, *2011 IEEE 19th International Conference on Program Comprehension (ICPC'11)*, pp. 219–220 (2011).

^{*1} Scratchfoot: <https://github.com/VictorNorman/ScratchFoot>