



## 直観的な Scratch 作品のためのユーザ入力タイミング特定の試み

岡本 圭悟<sup>†</sup>      伊原 彰紀<sup>‡</sup>      三倉 舞子<sup>§</sup>      橋谷 直樹<sup>¶</sup>  
和歌山大学<sup>†</sup>      和歌山大学<sup>‡</sup>      和歌山大学<sup>§</sup>      和歌山大学<sup>¶</sup>

### 1 はじめに

ビジュアルプログラミング言語の学習環境の一つである Scratch<sup>\*1</sup>では、膨大な作品が公開されている。ユーザは多様な実装方法を参照するために、作品をキーワード検索によって絞り込むことができる。しかし、ユーザが実装する動作イメージを言語化することは容易でないため、そのイメージに類似する作品の検索は困難である。従来研究 [1] では、ユーザがイメージするオブジェクトの移動軌跡をマウス操作によって入力し、それをクエリとする作品検索を実現しているが、検索にはあらかじめ作品を自動実行してスナップショットを撮影する必要があるため、入力操作を持つ作品は対象にできない。

本研究の事前分析では Scratch の入力操作を持つ作品数、および実装に必要な計算論理的思考能力を Dr.Scratch [2] を用いて定量的に計測した。Scratch の入力操作を持つ作品は、入力操作を持たない作品の 10 倍以上存在する。また、図 1 は入力操作を持つ作品と入力操作を持たない作品の実装に必要な計算論理的思考を 0 点から 21 点までの値で定量化した分布を示す。入力操作を有する作品は、入力操作を持たない作品に比べて計算論理的思考の熟練が必要であり、マンホイットニーの U 検定を用いて統

図 1: 入力操作をもつ作品と持たない作品の CT スコアの分布

計的有意差 ( $p$  値  $< 0.01$ ) を確認した。従って、入力操作を有する作品を検索の対象に含むべきであるが、入力のタイミングをオブジェクトの移動軌跡のみから判断することは容易でない。

本研究は、入力をもつ作品検索の実現に向けて、作品中でユーザが入力操作するタイミングを特定し、自動実行するシステムを提案する。具体的には、Scratch プログラムから、入力前後の命令処理、入力内容、入力前後のオブジェクトの位置を収集し、ユーザが入力操作するタイミングを特定する。

### 2 提案手法

本研究で検索対象に新たに含む作品の条件を示し、作品中においてユーザが入力操作するタイミングを特定、および入力操作を自動化する手順を述べる。

- **TODO: [XX]** のため、ユーザの操作を有するブロックのうち、文字の入力でないキー入力のみを有するブロックを含む作品
- **TODO: [XX]** のため、ユーザの定義した変数や関数を含まない作品
- **TODO: [XX]** のため、座標移動を行うブロックを含む作品

Toward identifying user input timing for intuitive Scratch work search

<sup>†</sup> Keigo Okamoto, Wakayama University

<sup>‡</sup> Akinori Ihara, Wakayama University

<sup>§</sup> Maiko Mikura, Wakayama University

<sup>¶</sup> Naoki Hashitani, Wakayama University

<sup>\*1</sup> Scratch: <https://scratch.mit.edu/>

1. **ブロック情報の取得:** **TODO: [事前に Scratch API によって取得した作品のプログラム情報 (sbi ファイル?) を解析し, ]** 対象の Scratch 作品に含まれる座標ブロック、ユーザの操作を有するブロックの種類、およびブロッ

クの実行順序を取得する。

2. 座標の計算: **TODO:** [「入力操作が発生するタイミング、および入力操作によりオブジェクトの移動場所を特定するため?」], Scratch 作品のプログラム実行よりオブジェクトの移動元, 移動先の座標を計算する。

3. 入力タイミングの決定: ブラウザ操作を自動化する Selenium を用いて, 入力操作が発生するタイミングにキー入力を実行する。作品のスキプットの処理が終了するまで, または作品のスクリーンショットの撮影時間が終了するまで繰り返す。

「ループブロック」などの制御ブロック中に処理終了をスキプットの処理の終了とする。  
**TODO:** [←よくわからん] また, 並列にスキプットが存在する場合も同様の手順でスナップショットを撮影する。

### 3 ケーススタディ

Scratch において公開される作品の中から **TODO:** [入力操作を有する? 作品を] ランダムに 10 件抽出し, 本手法で撮影したスナップショットを確認する。その結果, 10 件中 9 件の作品に対して入力のタイミングを自動特定することができた。入力のタイミングを取得できなかった作品\*2のプログラムを図 2 に示す。この作品では, スペースキーの後に上矢印キーのように順に入力すべきである。しかし, 提案手法では座標移動の情報のみで入力のタイミングを特定しているため, 入力ブロック間に座標移動ブロックが存在しないようなプログラムでは入力のタイミングを特定できない。今後は, **TODO:** [解決方法があれば描きたい]

### 4 おわりに

本論文では, 入力を有する作品を含んだ学習者の動作イメージと類似する作品検索に向け



図 2: 入力タイミングを取得できなかったプログラム

て, 自動で入力操作を行うタイミングの特定を試みた。今後は, 本手法を用いて撮影したスナップショットを用いて検索候補を増加し, 検索結果の多様性の向上を目指す。

### 参考文献

- [1] 福地ユキ, 伊原彰紀, 山本豪志朗, 橋谷直樹, ビジュアルプログラミング作品検索のためのオブジェクト操作データの時系列解析, 情報処理学会関西支部, 2021
- [2] J. Moreno-Le'on, G. Robles, and M. Rom'an-Gonz'alez, "Dr. scratch: Automatic analysis of scratch projects to assess and foster computational thinking," RED.Revista de Educaci'on a Distancia, vol.15, no.46, pp.1-23, 2015.
- [3] E. Aivaloglou, F. Hermans, J. Moreno-Le'on, Gr. Robles, "A dataset of scratch programs: scraped, shaped and scored," Proceeding of the International Conference on Mining Software Repositories (MSR'17), pp.511-514, 2017.

\*2 <http://hogege>