

Scratch プログラム学習過程の可視化

吉田万莉^{†1} 吉澤実乃莉^{†1} 来住伸子^{†1} 吉田葵^{†2}

概要: 情報化が進む現代, 子供のプログラミング教育に注目が集まっており, 子供にプログラミングを通してものを創る楽しさなどを提供することを目的としたツールが多く作られている. 今後, 教員は, こうしたツールを使って子供が作った多数のプログラムや子供の学習過程を把握する必要がある. 本研究では, 学習過程の把握を支援するための, インタラクティブな可視化ツールを設計し実装した. 本ツールでは, Scratch サイトで公開されている Scratch プログラムの可視化を行い, オーバービュー・ズーム・フィルター・ディテールなどの基本操作を提供している. 実際のプログラムの作成日時順などの表示により, 子供の学習過程がより簡単に確認できるようになった.

キーワード: プログラミング教育, 可視化, Scratch

Visualization of Learning Process for Scratch Programs

MARI YOSHIDA^{†1} MINORI YOSHIZAWA^{†1}
KISHI NOBUKO^{†1} AOI YOSHIDA^{†2}

Abstract: Programming education for children is getting huge attention in this age of information technology. There are a variety of programming tools and environments for children today. In the near future, the teachers of programming will need to understand many children's processes of learning programming, which may require reading a large number of programs. In this research, we created an interactive visualization tool to support the understanding of children learning to program. This tool visualizes Scratch programs available at <http://scratch.mit.edu>. Its visualization model is based on "an overview, zoom and filter, and the details on the demand" mantra by Ben Shneiderman. By displaying various attributes of Scratch programs in the chronological of order of program creation or modification, it will help the teachers find children's characteristics or the stages of the learning process more easily.

Keywords: Programming education, Visualization, Scratch

1. 本研究の背景

情報化が進む現代, 私たちは普段の生活の様々なところでコンピュータやプログラミングの恩恵を受けている. コンピュータにプログラミングを通して, 人間の意図した処理を行わせることができるのだということを, 子供に理解させることの必要性が叫ばれるようになり, 日本でも 2020 年から小学校のプログラミング教育が必修化されることとなった[1]. それに伴い, 子供のプログラミング教育に注目が集まり, プログラミングを通してものを創る楽しさなどを提供することを目的としたツールが多く開発されている. 中でも, マサチューセッツ工科大学のメディアラボが開発した, 子供のためのブロックプログラミング環境 Scratch は, 日本でも注目されている. Scratch では, 命令の書かれたブロックを積み木のように組み合わせることによりプログラムを作成する. Scratch により作成した作品(プロジェクト)を公開, 共有する場として, Scratch サイト

(<https://scratch.mit.edu/>) があわせて提供されていることも特徴のひとつである. マウスによるドラッグ&ドロップによりプログラムを作成できることや, コンパイルエラーが発生しないことから, 小学校でプログラミング教育を取り入れる際にも多く利用されるのではないかと考える.

一方, プログラミング教育が必修化されるにあたっては, プログラミングを教えることに慣れていない教員への対応が課題となっている.

2. 本研究の目的

現在, Scratch を利用している教員をはじめとした教育関係者は, 子供がどういった作品を作っているのかを確認する際, ひとりひとりの作品を実行するのに加えて, プロジェクトの中身(プログラム)を確認する. そのため, 子供の人数が増えると, 確認にかかる時間も増えてしまう. また, 教員は今後, 少なからず子供のプログラムを評価し, 学習過程を把握する必要性が出てくると考える. そこで, 本研究では, Scratch を利用する教育関係者が子供の学習過程や習得した機能を簡単に確認できることを目的とした, Scratch サイトで公開されているプログラムを解析し, 可視化して提示するツールを設計・実装した.

^{†1} 津田塾大学
Tsuda University
^{†2} 青山学院大学
Aoyama Gakuin University

3. 可視化について

可視化とは、人間の目には見えない事物や現象を、映像やグラフ・表などにして分かりやすくすることである。可視化の研究は、科学的可視化と情報可視化に分けられる。科学的可視化は、流体力学などの主に物質的なデータを対象としている。一方、情報可視化では、金融や言語学などに関する抽象的なデータを扱う。本研究は、後者の情報可視化にあたる[2]。

1996年、メリーランド大学の Ben Shneiderman 氏は、情報可視化システムを設計するガイドラインとして、「最初にオーバービュー（概要）を示し、ズームとフィルターを用意し、最後にオンデマンドでディテール（詳細）を示す」という、「Visual Information-Seeking Mantra」（情報可視化のマントラ）を定義した。『情報を見える形にする技術』の著者、Riccardo Mazza 氏は、情報可視化のマントラについて、次のように述べている。

このマントラは、情報可視化システムが情報を検索するプロセスで、どのようにユーザーをサポートできるかを明確に示しています。ユーザーがデータセット全体の理解を得られるように、データのコレクション全体の概要を提供する必要があり、それからユーザーはデータをふるいにかけて（フィルタリング）、特別関心のある特定の部分に焦点を合わせるかもしれません。最後にユーザーが要求すれば、データの特定のインスタンスの全ての詳細を表示しなければなりません。（Mazza, 2011, 中本訳, p. 124）

このマントラに基づいたデザインパターンは、単にデータセットを一目見たい人から、しっかりと目的を持ってデータを探索したい人まで、様々な人々のそれぞれのニーズに応えることができる[3]。

私たちは、この概念に基づいて、JavaScript ライブラリの D3 を用いて可視化ツールの設計を行なった。

4. D3.js について

D3 は、データビジュアライゼーションを作成するための JavaScript ライブラリであり、正式名称を Data-Driven Document という。D3 や d3.js とも表記され、HTML や SVG、CSS を用いて、多様な形式でデータを表現することができる[4]。ソフトウェアデザイナーの Scott Murray 氏は、D3 について次のように述べている。

基本的に、D3 はデータを使ったウェブドキュメントの生成、操作を助けるエレガントなソフトウェアだ。D3 は、次のようにしてこの仕事を進める。

1. ブラウザのメモリにデータをロードする。
2. ドキュメント内の要素にデータをバインド（結び付け）し、必要に応じて新しい要素を作る。
3. 要素内にバインドされたデータを解釈し、それに従ってビジュアルプロパティを設定して、要素を変換する
4. ユーザーの入力に反応して要素の画像を遷移させる（アニメーションで画面を切り替える）。

D3 の使い方を学ぶということは、どのようにデータをロード、バインドしたいか、要素をどのようにに変換し、遷移させたいかを D3 に指示する構文を学習することだ。

図 1 は上記の仕事の流れを表したものである。

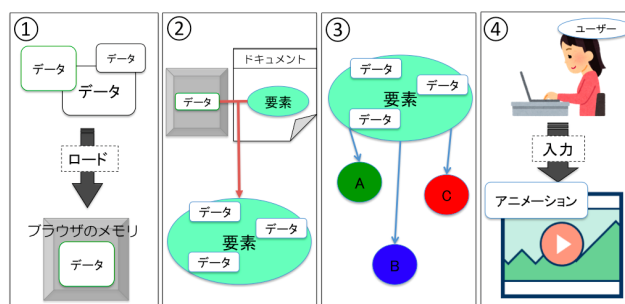


図 1 D3.js の仕事の流れ

Figure 1 Work Flow of D3.js

5. 設計方針

可視化ツールを作成するにあたって、「最初にオーバービューを示し、ズームとフィルターを用意し、最後にオンデマンドでディテールを示す」という情報可視化のマントラ[5]に基づいて、オーバービュー・フィルター・ズーム・ディテールの基本操作を提供する方針を立てた。

6. 可視化ツールについて

6.1 データの受け渡し

まず、Scratch サイトよりプロジェクトのデータを JSON 形式でエクスポートし、Python3 を利用してデータを抽出、解析を行い、JSON 形式のデータを出力する。そのデータを JavaScript ライブラリである D3.js を用いて可視化する。

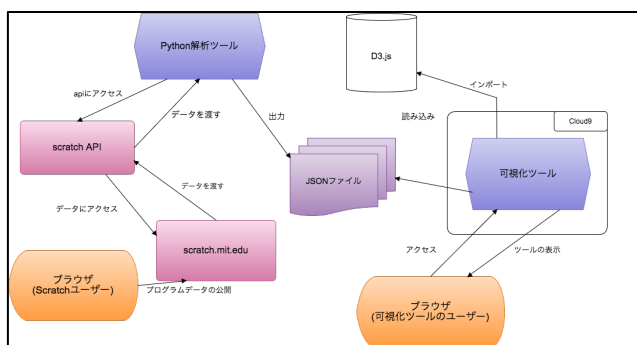


図 2 データの受け渡し図
Figure 2 Data Flow Diagram

本可視化ツールで用いる Scratch プロジェクトのデータに含まれる情報を以下に示す。

- 各カテゴリのブロック数

Scratch に用意されているブロックは、機能によって 10 個のカテゴリに分類されている。本データでは、カテゴリごとにブロック数を集計し利用している。

- スプライト数

スプライトとはオブジェクトを表す用語である。Scratch では各スプライトに対してプログラムを書くことができる。「スプライト数」は、プロジェクト内にあるスプライトの数を表す。

- 作成日時、共有日時、最終更新日時

作成日時は Scratch プロジェクトが作成され始めた日時であり、共有日時は Scratch プロジェクトが Scratch サイトで共有された（公開された）日時である。また、最終更新日時はそのプロジェクトが最後に更新された日時である。

上記の情報に加えて、データには、ユーザー名、プロジェクトタイトル、プロジェクト ID、プロジェクトサムネイルが含まれる。

6.2 可視化ツールの画面構成と実装機能

本可視化ツールの画面は、「折れ線グラフ」、「ヒートマップ」、「一覧表」の 3 つの部分から構成されている(図 3)。「折れ線グラフ」部分では、縦軸にブロック数、横軸に日時をとった折れ線グラフを表示する。折れ線グラフの上部には、フィルタリングのためのチェックボックスを配置している。「ヒートマップ」部分では、縦軸に各カテゴリ名、横軸にプロジェクト ID をとったヒートマップである。使用しているブロックの数が多いほどマス色が濃く表示されている。ヒートマップ上のラジオボタンによりヒートマップの表示を「ブロックの数の表示」、「各カテゴリの種

類数の表示」の 2 種類に変更することができる。また、マウスにマウスを合わせると、使用されているブロックの数がマス上に表示される。「一覧表」部分では、プロジェクト ID、プロジェクトのサムネイル、プロジェクトタイトル、ユーザー名を 1 列ずつ表示する一覧表である。すべての部分において、各カテゴリブロックを区別するために使用している色は、実際に Scratch で使用されている色である。

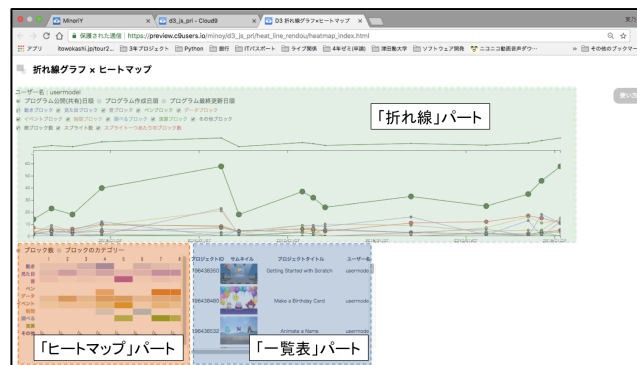


図 3 画面構成図

Figure3 Screen Configuration Diagram

本可視化ツールで実装した機能と情報可視化の「マントラ」との対応を表 1 に示す。対応の詳細については、次項で操作方法とともに述べる。

表 1 実装した機能とマントラの対応

情報可視化の「マントラ」	機能
オーバービュー	折れ線グラフの表示 ブロック全体のヒートマップ表示
フィルター	チェックボックス
ズーム	brush
ディテール	カテゴリごとのヒートマップ表示

6.3 可視化ツールの操作方法

ユーザーは初めに、折れ線グラフとヒートマップでデータの概要を見ることができる。折れ線グラフでは、総ブロック数、スプライト数、スプライト一つあたりのブロック数、カテゴリごとのブロック数の計 13 本の折れ線を確認する。ヒートマップでは、カテゴリごとのブロック数や、使用されているカテゴリの種類数を確認する(図 4)。

次に、折れ線グラフで知りたい情報を絞り込む場合は、折れ線グラフのチェックボックスによって、データにふるいをかけ、フィルタリングする(図 5)。また、折れ線グラフで気になるポイントがある場合には、brush 機能を使用し

て、グラフの表示範囲を設定し、そのポイントに焦点を合わせて、ズームしていく (図 6)。

最後に、ユーザーが各カテゴリでどのブロックが使われているのかを知りたい場合は、より詳細なヒートマップが表示されるカテゴリごとのヒートマップにアクセスして、ディテールを確認する (図 7)。

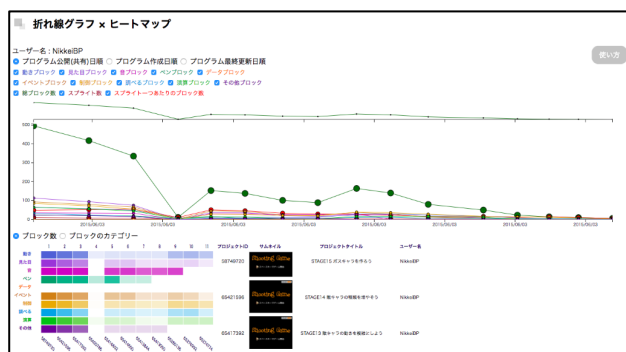


図 4 オーバービュー部分
(折れ線グラフとブロック全体のヒートマップ)

Figure 4 Overview
Line Graph and Heat Map for All Blocks

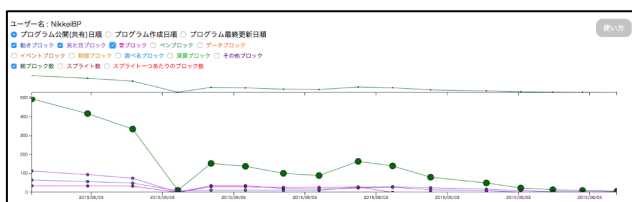


図 5 チェックボックスによるフィルタリング
Figure 5 Filtering by Checkbox Group

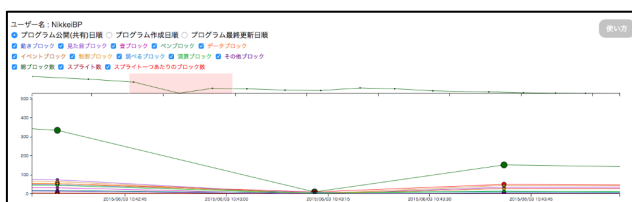


図 6 折れ線グラフのズーム
Figure 6 Zoom of Line Graph

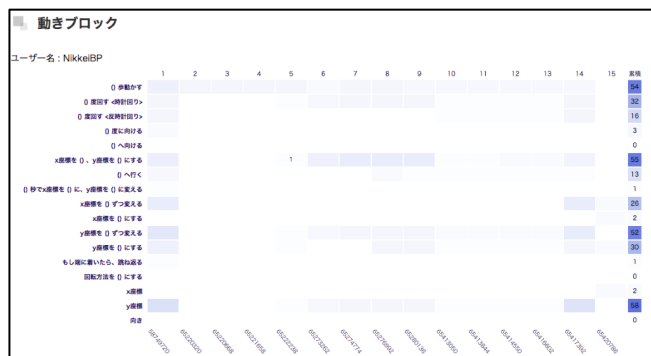


図 7 カテゴリごとの詳細ヒートマップ
Figure 7 Detailed Heat Map for Each Category

7. 仮説と調査例

7.1 仮説

可視化ツールの有用性を検討するにあたって、以下の 2 つの仮説を立てた。

仮説 1 ブロック数やスプライト数、使用するブロックのカテゴリ数は、学習が進むにつれ増加していく。

仮説 2 多くのプログラムを作成していくに伴い、その他音声、画像ファイルも増加していく。

ユーザーは、Scratch を始めて間もない頃、モデルプログラムを参考にプログラム作成を行うことが多いため、参考プログラム内で使われていないカテゴリのブロックはあまり使用しないと予測した。中でも、変数やリストに関する「データ」カテゴリや関数定義に関する「その他」カテゴリでは、自分でブロックを作成しなくてはならないため、初心者が使用するのは難しい。そこで、たくさんのプログラムを作成するにつれ、これらのブロックを使いだしていくと考えた。また、Scratch サイト全体で使われているブロックの利用頻度を表した図 8 (資料[5]) によると、「ペン」カテゴリは Scratch ユーザー全体で見ても、利用頻度が最も少なく、初心者はあまり使わないと推測される。

本稿では、「モデルユーザー」と「実在のユーザー」を対象に調査し、仮説を検証できるかどうかで本ツールの有用性を検討する。「モデルユーザー」とは、ある一連のプロジェクトを、Scratch のユーザーが作成したプロジェクトと見立てたものである。

「モデルユーザー」として、3 種類のデータを調査した。1 つ目は、「Why! プログラミング」のプログラム、2 つ目は、Scratch 公式サイトのチュートリアルに沿って作成したプログラム、そして、3 つ目は、日経 BP 社 (ユーザー名: NikkeiBP) が公開している、『Scratch ではじめよう! プログラミング入門』[6] のスタジオのプログラムである。これらのプログラムは、プログラミングの専門家が携わって制作されているため、仮説の検証に適しているのではないかと

考えた。

「実在のユーザー」のデータには、NHK のプログラミング教育番組「Why!?プログラミング」での「Why!?大喜利」という企画に作品を投稿し、2016 年 9 月から 2017 年 11 月に優秀な作品に贈られるワイワイ賞を受賞した Scratch ユーザー 14 人[7]のプログラムを利用した。

使用されている総ブロック数、スプライト数、カテゴリのブロック数、共有された日時などのデータを折れ線グラフ・ヒートマップの 2 種類で可視化を行なった。

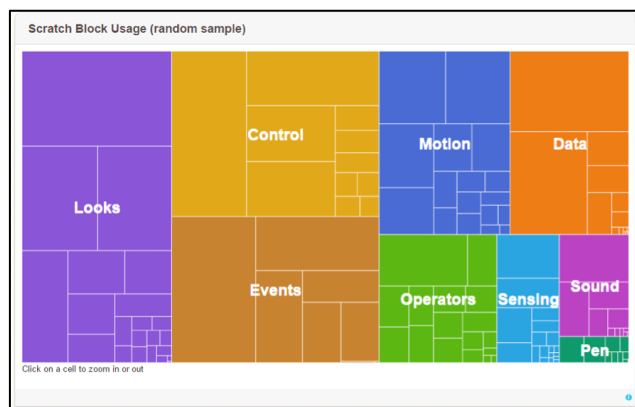


図 8 全ユーザーカテゴリ別利用頻度

(2016 年 10 月 12 日時点) [5]

Figure 8 9 Categories of Blocks Use Frequency

(October 12, 2016)

7.2 モデルユーザーの可視化の調査例

モデルユーザーのプログラムを本ツールを用い可視化した。その結果、Scratch 公式サイトのチュートリアルは、例外はあるものの、右上がり総ブロック数が増えていることを確認できた。また、単純なブロックを多く使用していることに加え、「ペン」カテゴリと「その他」カテゴリは使用していないことから、Scratch にユーザー登録したばかりの初心者向けの教材であると想定できる (図 9)。

一方、「Why!?プログラミング」のモデルプログラムについては、ブロックをおよそ 500 個使用したプログラムもあった。ヒートマップの機能を利用し、Scratch 公式サイトのチュートリアルに比べて使用している各カテゴリの種類数の多かったため、初心者向けというよりは、経験者向けのプログラムであると推測できた (図 10)。

さらに、『Scratch ではじめよう！プログラミング入門』[6]に掲載されているプログラムでは、共有日順 (図 11) と作成日順 (図 12) と最終更新日順 (図 13) の折れ線グラフに並べ替えたところ、共有日順では STAGE01 から STAGE15 まで順番に並んでいた。しかし、作成日順に並び替えると、まず、STAGE15 を作成した後に、STAGE01 から順に作成をしていた。その後、最終更新日順に並べ替えてみると、STAGE15, STAGE14, と共有日順とほぼ逆の順番で更新をしていた。これらのことから、このユーザーは

STAGE15 の大きなプログラムをまず作成をしてから、徐々にブロック数を減らし、簡単なプログラムへと STAGE15 のプログラムを変更していったと推測できた。

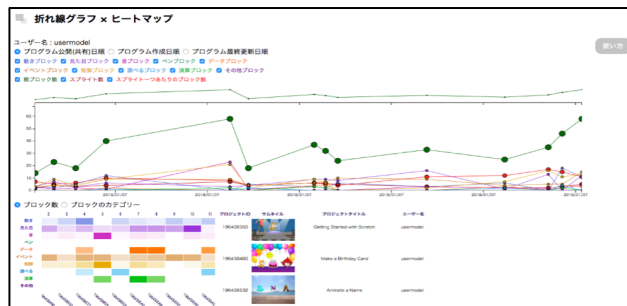


図 9 Scratch 公式サイトのチュートリアルの可視化

Figure 9 Visualization of Scratch Official Tutorial Programs

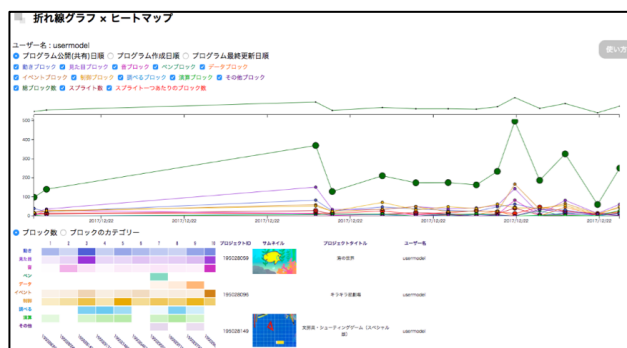


図 10 「Why!?プログラミング」の

モデルプログラムの可視化

Figure 10 Visualization of Model Programs of

“Why!? Programming”

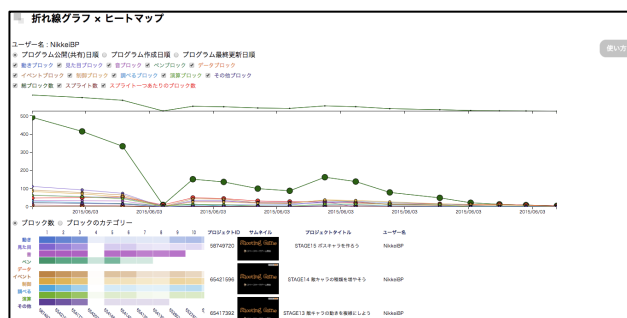


図 11 NikkeiBP のプログラム共有日順

Figure 11 Visualization of NikkeiBP's Programs

in Order of Shared Date

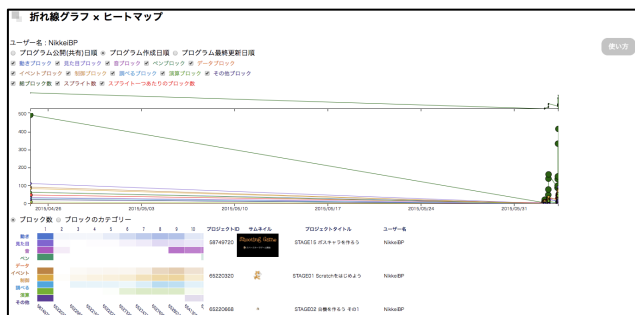


図 12 NikkeiBP のプログラム作成日順

Figure 12 Visualization of NikkeiBP's Programs in Order of Created Date

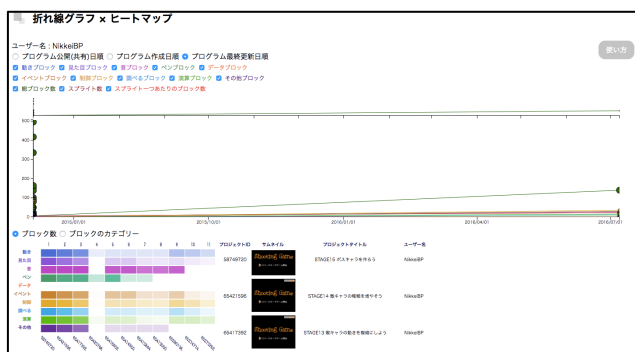


図 13 NikkeiBP のプログラム最終更新日順

Figure 13 Visualization of NikkeiBP's Programs in Order of Modified Date

7.3 実在のユーザーの可視化の調査例

ワイワイ賞受賞者 14 人の中から、特徴が見られた A さん、B さん、C さんの 3 人の実例を提示する。

まず、A さんは、共有日順 (図 14) と作成日順 (図 15) に並べたところ、最もブロック数の多かったプログラムが作成日から 1 年以上経ってから公開していることがわかった。このことから、時間をかけて少しずつプログラムを改善していったのではないかと考えられる。作成日順に可視化したときに「その他」カテゴリが初めて登場したのが 33 個目のプログラムだとヒートマップからわかる。このカテゴリのブロックは自身でブロックを定義するという難しいブロックであるため、A さんは Scratch プログラミングにおける学習が進み始めていると推測できる。

B さんのプロジェクトを可視化した結果を図 16 に示す。共有日順に並べて見たところ、y 軸の総ブロック数の最大値が 1000 個を超えていることがわかった。突出してブロック数が多いものや、初めて共有したプログラムでも、使用しているブロックの種類数が多い。また、先述した「その他」カテゴリを 1 つ目のプログラム作成時より使用している。詳細のヒートマップを見ても、網羅的にブロックを使用していることから、B さんはプログラミング経験者、あるいは Scratch 上級者の可能性があると推測した。

C さんのプロジェクトを可視化した結果を図 17 に示す。

例外はあるものの、右上がり総ブロック数が増えている様子が見られた。また、総ブロック数と「見た目」カテゴリのブロック数の折れ線がほぼ同じ形で推移しており、「見た目」カテゴリを多く使っていることが分かった。そのことから、物語をよく作っているのではないかと推測した。

Scratch で物語を作る際に、多く使うと考えられるブロックが複数ある。例えば、図 18 の「見た目」カテゴリの「表示する」、「隠す」を使って、場面の切り替え時にキャラクターを表示・非表示にする、などの事例が考えられる。詳細のヒートマップを見ると、C さんも、それらのブロックを多く使用する傾向があった。C さんの作品を Scratch サイトで確認したところ、本ツールから読み取れたとおり、物語の作品が多かった。

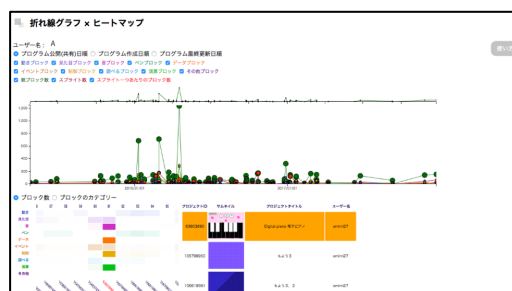


図 14 A さんのプログラム共有日順

Figure 14 Visualization of A's Programs in Order of Shared Date

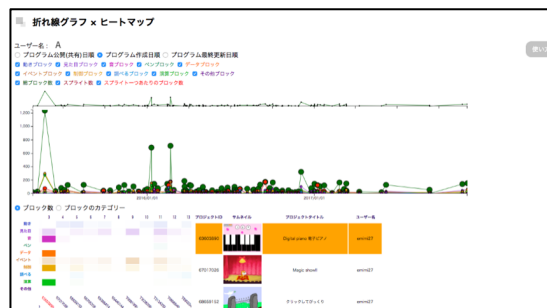


図 15 A さんのプログラム作成日順

Figure 15 Visualization of A's Programs in Order of Created Date

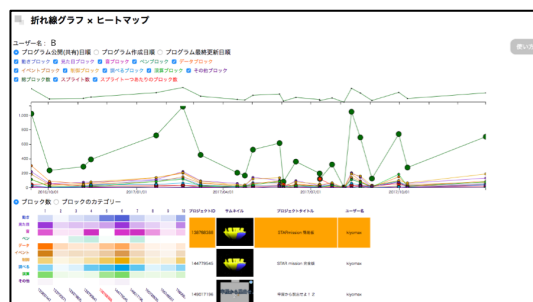


図 16 B さんのプログラム作成日順

Figure 16 Visualization of B's Programs in Order of Created Date

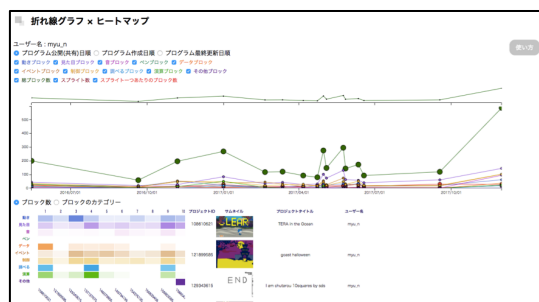


図 17 C さんのプログラム作成日順
Figure 17 Visualization of C's Programs
in Order of Created Date

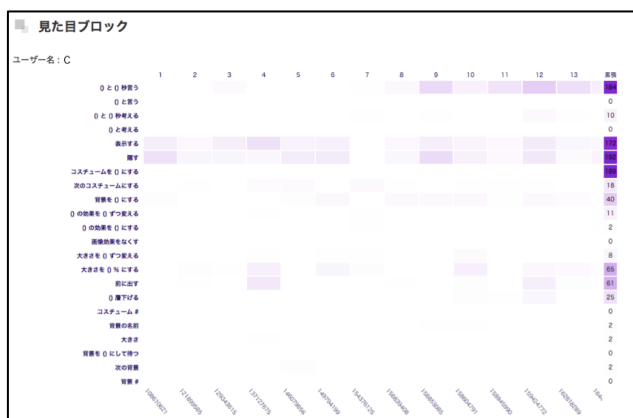


図 18 C さんの見た目ブロック詳細ヒートマップ
Figure 18 Detailed Heat Map for Looks Blocks
of C's Programs

8. 考察

モデルユーザーのプログラム例を対話的に可視化することにより、この可視化ツールは Scratch プログラム例を含む教材サイトの傾向を確認することにも役立つと考えられる。可視化によって、各教材サイトの特徴的なプログラムの発見があった。プログラム例から学習が進むにつれ、ブロック数や各カテゴリのブロックの種類の数が増える傾向が確認でき、仮説 1 が支持される可能性を示した。

さらに、実在のユーザーの可視化によって、Scratch プログラムの作品の特徴を可視化することができた。ヒートマップでは、ブロック数(ブロックの使用されている回数)の表示と、使用されているブロックのカテゴリ(使用されているブロックの種類数)の表示の 2 種類を提供することによって、ユーザーがよく使用するブロックを見つけやすくなることができた。

また、現段階では、音声ファイルや画像ファイルの有無がわかる可視化を実装していないため、仮説 2 の検証はできていない。

以上のことから、定性的な評価のみではあるが、本ツールの有用性を示した。

9. まとめと今後の課題

9.1 まとめ

本稿では、Scratch プログラムの可視化ツールを D3.js, Python3 を用いて作成を行なった。本ツールを用いた可視化により、ユーザーの学習過程を簡単に確認することができた。また、特徴的なプログラムの発見があり、ツールの有用性を確認することができた。そして、子供のプログラムの可視化だけでなく、Scratch プログラムの教材の難易度判定という可視化ツールの新たな有用性が期待できそうだ。

9.2 今後の課題

本ツールの評価においては、定量的な評価はまだ行っていない。今後は実際に Scratch 教育者にこのツールを使用してもらい、より多くのユーザーのデータを可視化することで定量的評価を行いたいと考えている。

本ツールの機能実装に関しては、グラフの表示範囲選択のみであったが、折れ線グラフで選択した範囲に合わせて、ヒートマップとプログラムの一覧表が選択範囲内のものだけになり、一目見ただけでよりわかりやすくなるよう改善を加えたいと考えている。また、複数のユーザーのデータを表示できるようにし、ユーザー間の学習過程を比較できるようにしたい。

そして、Python3 のプロジェクト解析プログラムと JavaScript の可視化プログラムを連携させ、ユーザー名やスタジオの ID を入力し、実行ボタンを押すと、指定したデータの折れ線グラフやヒートマップなどの可視化が表示されるという Web サービスとしての公開を目指す。

参考文献

- [1] 「小学校学習指導要領解説 総則編」文部科学省
http://www.mext.go.jp/component/a_menu/education/micro_detail/_icsFiles/afieldfile/2017/07/12/1387017_1_1.pdf (参照 2018-02-10) .
- [2] 高間康史(2017)『情報可視化-データ分析・活用のためのしくみと考えかた』森北出版株式会社
- [3] Mazza, R(2011)『情報を見える形にする技術 - 情報可視化論』(中本浩訳)ポーンデジタル
- [4] 「データ・ドリブン・ドキュメント」D3.js - 日本語ドキュメント
<http://ja.d3js.node.ws/> (参照 2018-01-03) .
- [5] 「File:ScratchBlock Usage.png」Scratch-Wiki
https://wiki.scratch.mit.edu/wiki/File:Scratch_Block_Usage.png (参照 2018-01-10) .
- [6] 杉浦学(2015)『Scratch ではじめよう!プログラミング入門』阿部和広監修, 日経 BP 社.
- [7] 「Why!?大喜利」ワイワイプログラミング
<http://www.nhk.or.jp/school/programming/oogiri/index.html> (参照 2018-01-09) .