



情報の授業をしよう!

本コーナー「情報の授業をしよう!」は、小学校や中学校で情報活用能力を育む内容を授業で教えている先生、高校で情報科を教えている先生や、大学初年次で情報科目を教えている先生が、「自分はこの内容はこういう風に教えている」というノウハウを紹介するものです。情報のさまざまな

内容について、他人にどうやって分かってもらうか、という工夫やアイディアは、読者の皆様にもきっと役立つことと思います。そして「自分も教え方の工夫を紹介したい」と思われた場合は、こちらにご連絡ください。

(E-mail : editj@ipsj.or.jp)

Scratch を活用した段階的なプログラミング指導例



谷川佳隆 | 千葉県立八千代東高等学校

Scratch とは

Scratch (スクラッチ) は、MIT (アメリカ・マサチューセッツ工科大学) のメディアラボが無償で公開しているプログラミング環境である。いろいろなブロックの組合せ方や数値等を設定することで、スプライトと呼ばれる画像のオブジェクト等をプログラミングすることができる。ブロックはカテゴリ別に色分けがされていて、コードはとても見やすく確認しやすい。Scratch にはスプライトと背景にたくさんの画像があらかじめ多数用意されている。また、スプライトと背景は外部から画像ファイルを読み込むことができる。ブロックなどの表示はいろいろな自然言語で表示でき、国籍年齢を問わず気軽に使用できる。また、Scratch のサイトで作品を公開することができ、交流が盛んに行われている。

■表-1 オンライン

	必要な環境	拡張子
Scratch 2.0	Adobe Flash Player ^{☆2}	sb2
Scratch 3.0	Google chrome ^{☆3} , Mozilla Firefox ^{☆4} , Microsoft Edge ^{☆5} などの新しいブラウザ	sb3

Scratch の種類

Scratch には 2018 年 10 月末現在以下の種類がある。オンライン (表-1) とオフライン (表-2) があり、種類により、使用環境や画面構成やブロックの種類、拡張子などが少しずつ異なる。

オンラインでは、プロジェクトファイルをオンライン上でもローカルにも保存することもできる。

Scratch 2.0^{☆1} は 2018 年までしか使えず、2019 年 1 月 2 日に Scratch 3.0 (図-1) に切り替わる。

■表-2 オフライン

	必要なソフト	拡張子
Scratch 1.4 ^{☆6}	×	sb
Scratch 2.0 オフラインエディター ^{☆7}	Adobe AIR ^{☆8}	sb2

☆1 <https://scratch.mit.edu/>

☆2 <https://get.adobe.com/jp/flashplayer/>

☆3 https://www.google.com/intl/ja_ALL/chrome/

☆4 <https://www.mozilla.org/ja/firefox/new/>

☆5 <https://www.microsoft.com/ja-jp/windows/microsoft-edge>

☆6 https://scratch.mit.edu/scratch_1.4/

☆7 <https://scratch.mit.edu/download>

☆8 <https://get.adobe.com/jp/air/>

Scratch を活用した指導と評価

ここでは、説明をしなからでも 30 分以内に作成できる 4 つの題材を紹介する。題材はまず簡略化したものを作成してから、そのプログラムを改良していくように指導している。

題材のコードは紙面上から確認しにくいので、Scratch のサイトに公開しておく。

私は授業で完成したらどう実行するのかを生徒に見せてイメージを持たせてから、一緒にプログラミングを始めるようにしている。授業中にわざと間違えてプログラミングしたものを実行して見せたり、生徒に質問しながら授業を進めたりするようにしている。どのカテゴリのどのブロックを使うのか、どのように数値を設定したらよいかなど生徒に答えてもらうようにしている。また、生徒同士教え合うことを推奨しているので、席を動いて教えている生徒がいたら、先に進んでいいか確認している。

題材 1：数当て^{☆9} (図-2)

指導のポイント

1 から 100 までの乱数 (当てるべき数) を発生させ、



■ 図-1 Scratch 3.0 の画面



■ 図-2 「数当て」実行画面例

☆9 <https://scratch.mit.edu/projects/247835566/>

数を予測し、その予測と当てるべき数を判断し、ねこにその結果 (当たり・大きい・小さい) をしゃべらせる。ここで学ぶ内容は、乱数の活用、変数の活用、条件判断、二分探索である。

まずは、当たりか外れの判断ができることを確認する。次に、予想より大きな数で外れたのか小さな数で外れたのかを判断できるように改良する (図-3)。

生徒に当たりの数を予想させながら、当たりの数を当てるには二分探索を活用するとよいことに気付かせる。

この題材は、スプライト数 1、スクリプト数 1、最小ブロック数 11、ブロックの種類 6、用意する画像は特にない (図-4)。

評価のポイント

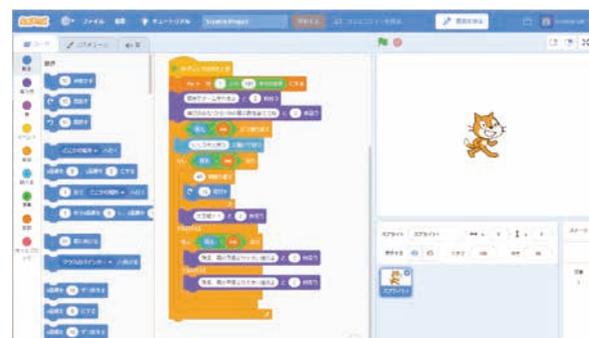
プログラミングを最低限 (A) と改良 (B) とさらなる改良 (C) の 3 つの段階で評価すると、ここでは、以下のように評価できる。

ねこが予想結果を正しくしゃべる (A)。当てるまでの回数を表示できる (B)。当たり外れ時の画面表示を工夫している (C)。

次の題材も 3 つの段階で評価する例を載せる。



■ 図-3 判定の改良例



■ 図-4 数当て作成の画面例

題材2：動画作成^{☆10} (図-5)

指導のポイント

連続する複数の画像を用意し、それを取り込み、一定間隔で入れ替えることで動画として見るができるようにする。画面を切り替える間隔は上下キーで変更できるようにする。ここで学ぶ内容は、画像を並べる工夫(ソートの体験)、動画の原理、キーによる変数の変更である。

まずは、数値を固定して画像を切り替えて、動画に見える値を探す。次に、変数を利用してキーにより間隔の数値を変更できるようにする (図-6)。

この題材は、スプライト数1、スクリプト数1、最小ブロック数11、ブロックの種類4、用意する画像は連続する16枚ぐらいになる (図-7)。

評価のポイント

画面を切り替える間隔は上下キーで操作できる (A)。間隔がマイナスにならない (B)。画像を逆再生できる (C)。

題材3：流れ星^{☆11} (図-8)

指導のポイント

1つの星の画像を使い、流れ星が右上から左方向

に不定期に20個流れてくるようにする。ここで学ぶ内容は、画像の表示・非表示、クローンの活用、乱数の活用である。

まず、星の位置を決めて、星を表示しその後動かそうとしても星が1つしか流れないことを確認する。次に、星を複製するのではなくクローンを作るブロックを活用する (図-9)。

この題材は、スプライト数2、スクリプト数3、最小ブロック数18、ブロックの種類5、用意する画像は特にならない (図-10)。

評価のポイント

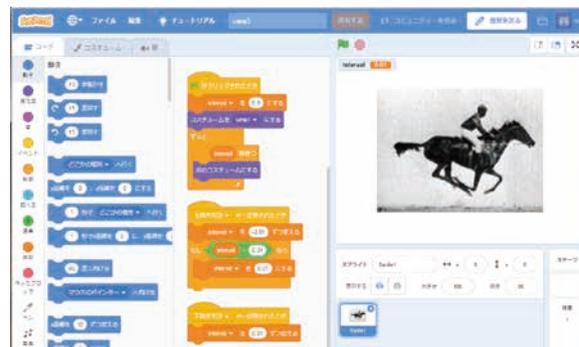
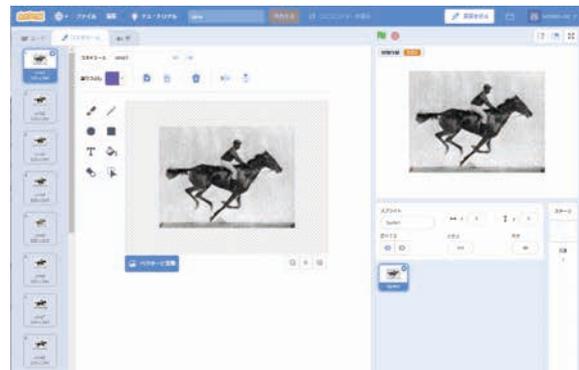
流れ星が右上から左方向に不定期に20個流れて



■ 図-5 「動画作成」実行画面例



■ 図-6 判定の改良例



■ 図-7 動画作成の画面例

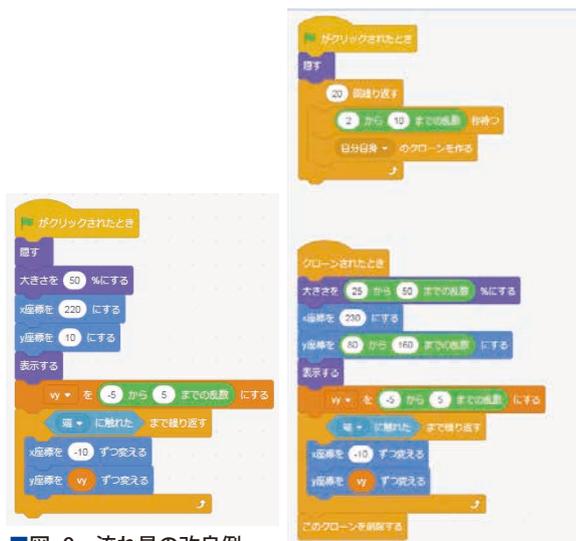


■ 図-8 「流れ星」実行画面例

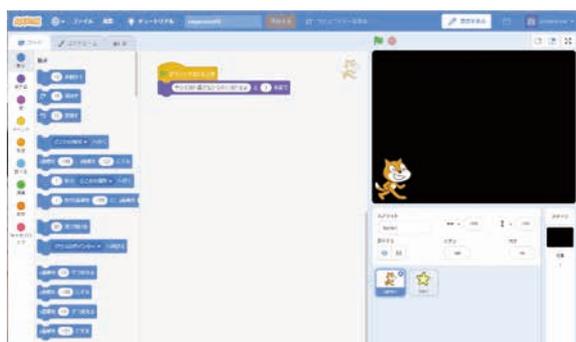
^{☆10} <https://scratch.mit.edu/projects/137990439/>

^{☆11} <https://scratch.mit.edu/projects/253610535/>

くる (A). 流れ星の出る高さや流れ星の大きさなどはランダムである (B). 最初に流れてくる流れ星の数を決められる (C).



■図-9 流れ星の改良例



■図-10 流れ星の作成画面例



題材4：さいころ^{☆12} (図-11)

指導のポイント

乱数によってさいころの画像を変えるシミュレーションをする。流れとしては、ねこがしゃべった後に、さいころを振り、振った後にねこが出た目の合計をしゃべる。ここで学ぶ内容は、乱数の活用、メッセージを使用したスクリプト間のやりとり、スプライトの複製利用、演算の利用である。

まずは、1つのさいころで出た目を言うようにする。次に、さいころを2つに増やしても目の和の合計を表示できるように考える (図-12)。

この題材は、スプライト数3、スクリプト数3、最小ブロック数21、ブロックの種類5、用意する画像はさいころの目の6枚になる (図-13)。

評価のポイント

さいころ2個の出た目の合計を表示できる (A). 改良は出た目の合計によって、表示を変える (B). 出た目の合計をカウントするようにする。さいころの個数を増やす (C).



■図-11 「さいころ」実行画面例



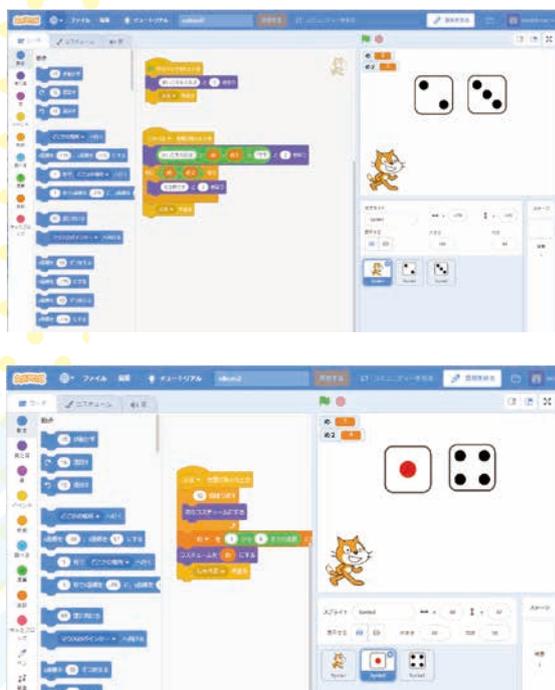
■図-12 さいころの数の改良例

^{☆12} <https://scratch.mit.edu/projects/258523548/>

Scratch でのプログラミング

2020年にはすべての小学校でプログラミング教育が始まり、小学校でScratchを経験する児童が増えると予想される。また、Scratch風のブロックもしくはタイルのようなものを組み立ててプログラミングできるものは数多く存在し、それを経験してきた児童・生徒が多くなると思われる。

したがって、Scratchを活用したプログラミングの指導する場合、小学校と同じような内容ではなく、高校生の発達段階にあったプログラミングの指導が必要になる。プログラミングするだけでなく、作品の説明書の作成や、作品の改良や応用方法などを考えさせてもよい。ただ、『情報』が2単位と授業時間が限られているため、プログラミング指導だけに時間をかけられず授業内に大きな作品を作ることや作品の説明書などに時間をかけることは難しいかもしれない。時間をかけたいのであれば、放課後の活用や長期休業の宿題などにして時間の確保が必要になる。



■ 図-13 動画作成の画面例

できれば授業内に時間を設けて、自由作品を作らせて、お互いに作品を見せ合い、相互評価するという活動を入れると、お互いの作品の良い点を学び合うことができる。

プログラミングの評価について

プログラミング指導においてはどのように評価するのかという観点をしっかり持っていないと評価が安定しない。また、生徒一人ひとりの作品をくまなく見ていては時間がいくらあっても足りない。そこで観点を絞り評価していく必要がある。そして、どのような評価をするかあらかじめ生徒に伝えておく方が望ましい。

生徒の作品1つについて3つ項目をチェックするとして、どのくらいの時間が必要なのかの見積もりができていないと評価することが難しくなる。すべての作品を教員が評価するのではなく、自己評価や相互評価などを取り入れることも考えられる。

生徒のプログラミング能力はばらばらであるが、生徒・教員の双方がどんなポイントで評価するのかを理解しておくことが大切かと考える。

指導に向けて

Scratchを活用したプログラミングは関連書籍やWebサイト、そしてテレビ番組まであり、参考になるものはたくさんある。その中でこの記事が参考になればうれしく思う。

まずは、指導者がScratchを楽しんで授業に活用していただければ幸いである。

(2018年10月22日受付)

谷川佳隆 (正会員) yotanikawa@gmail.com

千葉県立八千代東高等学校 (主幹教諭)。2003年より情報をメインで担当。