

# Scratch による初心者プログラミング教育の考察

## Consideration to educate programming to a beginner by Scratch

市川 弘幸<sup>†</sup>

Hiroyuki Ichikawa<sup>†</sup>

<sup>†</sup> 市川技術士事務所

<sup>†</sup> Ichikawa Professional Engineer Office

E-mail: <sup>†</sup> jtyyc953@ybb.ne.jp

### 1. はじめに

Scratch は、8～16 才の年齢層を対象に、米国の MIT のメディアラボにより開発された、プログラミング学習環境であり、誰でも対話的なストーリー、アニメーション、音楽、ゲームなどの作品を簡単に作成することができる。Scratch2.0 は WEB 上で動作し、無償で利用可能であると共に、作品を WEB サイト上の他のユーザと共有することができる。現在 WEB サイトに 1,170 万人以上のユーザが登録され、1,460 万以上の作品が共有されている。10～16 才の年齢層のユーザが最も多いのが特徴である。

子供がプログラミングを始める目的は、面白い作品を簡単に作り、その作品を自分で楽しんだり、他のユーザと共有することである。そのために、Scratch には、子供が興味を持ちそうな作品を簡単に作るための要素が組み込まれている。また、Scratch は、キー入力を極力減らし、マウス操作のみでプログラムを作成することができるようになっている。さらに、Scratch には、オブジェクト（スプライト）を簡単に作り、並行動作させる仕組みが組み込まれている。

Scratch は子供向けのプログラミング学習環境であり、大人や上級者は高度な作品を作れないと思われるかもしれない。後述するが、Scratch は他のアプリケーションとの間で大量のデータを交換できるので、高度な作品を作ることができる。

Scratch は、高度な作品を子供にも作成可能にした画期的なプログラミング学習環境であり、世界中に普及している。日本においては、Scratch のユーザ数は 9 万人を超え、小中学生を中心に作品が作られ、共有されている。

筆者の経験では、C 言語などのソースコード記述型言語を初心者に教育する場合、キーのタイプミス、文法エラー、実行時エラーが障害になり、本来の目的であるアルゴリズムの習得のための時間が少なくなるという問題が発生していた。

そこで、子供向けのプログラミング学習環境である Scratch を使用して、初心者プログラミング教育を行

う場合、初心者にとって、どのような利点があるかを考察する。また、いわき市で実施した小中学生プログラミング体験教室において、Scratch を使用した事例について報告する。

### 2. Scratch を初心者教育に使用する利点

Scratch を使用して、初心者プログラミング教育を行う場合、初心者にとって、どのような利点があるかを考察する。

#### 2.1 キーのタイプ

初心者共通する傾向は、キーのタイプに慣れていないことである。一方、マウス操作は初心者でもすぐに習得することができる。Scratch において、キーのタイプが必要なのは、次の内容を入力する場合に限られる。

- (1) 変数名、リスト名、メッセージ名、背景名、スプライト名、コスチューム名、音名、定義ブロック名、仮引数名の定義（それぞれ 1 回のみ）
- (2) ブロック（命令）内のパラメータ
- (3) スプライトが発する言葉、考えのセリフ
- (4) リスト内データ
- (5) ユーザに問かける質問内容
- (6) コメントの記述

上記の中で、(1)、(2)はプロジェクト作成上必須であるが、(3)～(6)は任意でよい。ブロック内のパラメータは、主に数字キーが使われ、英字キーが使われる頻度は少ない。そのため、テンキーが主に使用され、初心者は簡単にタイプすることができる。日本語は必要に応じて記述してもよいが、一般的には英語で記述することが多い。名称の誤りによる文法エラーは発生しない。なぜなら、名称を 1 回定義しておけば、名称の参照はプルダウンメニューから選択するか、変数レポーターをドラッグすればよいからである。名称が変更されると、その名称を参照しているブロックは、自動的に変更される。これは初心者にとって、大きな利点で

ある。

## 2.2 マウス操作

Scratch は、ほとんどマウス操作のみによりプログラミングを行う。必要とされるマウス操作は次のものであり、これは初心者にとって、習得し易い。

- (1) クリック
- (2) 右クリック
- (3) シフト+クリック
- (4) ドラッグ
- (5) ダブルクリック

## 2.3 スクリプトの作成

スクリプトは、ブロックパレット上の視覚的に色分けされたブロックを、マウス操作によりドラッグし、スクリプトエリアにドロップしてブロックを積んでいく操作により作成される。スクリプトの文法ミスは発生せず、スクリプトは自動的に構造化される。これは初心者にとって、大きな利点である。

## 2.4 スクリプトの起動

スクリプトを起動するには、スクリプトエリアに積まれたブロックのどれかをクリックすることで、その積まれたブロックの上から下までの全部を実行することができる。また、イベントを発生させ、同時に多数のスクリプトを起動することができる。起動用のスクリプトの作成は不要である。これは初心者にとって、大きな利点である。

## 2.5 スクリプトの実行

スクリプトを実行する際、スクリプトは Java アプレットに変換されるが、このとき文法エラーは、全く発生しない。また、実行時エラーは発生せず、スクリプトは最後まで実行される。エラーによる思考の中断がなくなる。これは初心者にとって、大きな利点である。

## 2.6 スクリプトのテスト

変数やリストや他のレポーターの表示用のチェックボックスをオンにすることにより、スクリプトを実行中に、その内容をステージ上に表示することができる。内容を表示するためのスクリプトは不要である。これは初心者にとって、大きな利点である。

また、スクリプトの実行を途中で中断し、変数やリストや他のレポーターの内容を確認することができる。このとき、中断したい箇所に、中断用のブロックを挿入する必要がある。

スクリプトの実行中に、スクリプトを参照しながら、ステージ上に表示されるスプライトの動きや作図状況

を確認することができる。また、スクリプトの実行中に、スクリプトを参照しながら、音、音声、音楽をスピーカから再生することができる。

## 2.7 Scratch の特長

Scratch はソースコード記述型言語と比較して次の特長がある。

- (1) スクリプトを簡単に作成できる
- (2) スクリプトを簡単に起動し、実行できる。起動用のスクリプトの作成は不要である。
- (3) スクリプトの実行中に、変数やリストや他のレポーターの内容をステージ上に表示することができる。内容を表示するためのスクリプトの作成は不要である
- (4) 文法エラーや実行時エラーは全く発生しない
- (5) スクリプトの実行を途中で中断して、変数やリストや他のレポーターの内容を確認することができる
- (6) スクリプトの実行中に、スクリプトを参照しながら、ステージ上でスプライトの動き、作図状況を確認することができる
- (7) スクリプトの実行中に、スクリプトを参照しながら、音、音声、音楽をスピーカから再生することができる

上記の特長は、プログラミング教室において、限られた時間内に多くの例題の演習を行う場合に有効である。スクリプトの作成、起動、実行、テスト、修正のサイクルを高速に廻して、作品を作ることができる。

## 3. Scratch による高度な作品作りの可能性

Scratch は、視覚的に色分けされたブロックをスクリプトエリアに積んでいくビジュアルな言語仕様であるため、高度なプログラミングはできないのではないかと、思われがちである。そこで、Scratch の言語仕様を、ソースコード記述型言語と比較しながら、高度な作品作りの可能性を検討する。

### 3.1 変数

変数はデータ型の宣言は不要である。データ型は自動的に判別され、整数、小数、文字、文字列のいずれかの型になる。変数の種類はグローバル変数と、ローカル変数がある。

### 3.2 リスト

1 次元の配列であり、データ型や配列サイズの宣言は不要である、データ型の自動判別は、変数の仕様と同様である。リストの種類は変数と同様に、グローバ

ルリストとローカルリストがある。リストはスタック形式の操作でも使用することができる。

### 3.3 演算ブロック

演算ブロックには、四則演算(+,-,\*,/)、一様乱数、比較演算(>,<), 論理演算(かつ、または、ではない)、文字列の結合、文字列内の指定位置の文字抽出、文字列の長さ、剰余、四捨五入、絶対値、切り上げ、切り下げ、平方根、三角関数、自然対数、対数、指数関数、リストの指定位置の値、リストの長さ、リストに指定された値が含まれるか、のブロックがある。その他の演算や関数には対応していないので、別途ユーザが定義ブロックを使用して作成する必要がある。

### 3.4 変数の操作

変数の操作ブロックには、値の代入、値の増減、変数レポーター、のブロックがある。

### 3.5 リストの操作

リストの操作ブロックには、値の追加、指定位置の値の削除、指定位置に値を挿入、指定位置の値を置換、のブロックがある。

### 3.6 制御ブロック

制御ブロックには、n 回繰り返し、無限繰り返し、条件が満たされるまで繰り返し、指定した（あるいはすべての）スクリプトの停止、指定秒数待つ、条件が成立するまで待つ、メッセージを送る、メッセージを送って待つ、条件分岐、2 方向条件分岐、のブロックがある。制御ブロックは、それ自身が構造化されていて、入れ子にすることもできる。

繰り返しブロックから脱出するブロックはないので、繰り返しの条件を工夫する必要がある。

### 3.7 イベントの通知

イベント発生の通知ブロックには、緑の旗がクリックされたとき、特定のキーが押されたとき、このスプライトがクリックされたとき、背景が特定の背景になったとき、メッセージを受け取ったとき、などのブロックがある。イベント発生時は、同時に複数のスクリプトに通知することができる。

### 3.8 ペンの操作

ペンの操作ブロックには、図形の消去、ペンの上げ下げ、ペンの色、濃さ、太さを指定したり、変化させるブロックがある。スプライトを移動しながら、ペンを上下させて図形を描画する。スプライトの画像をス

テージ上にスタンプするブロックもある。

### 3.9 タイマー

タイマーのブロックには、タイマーをリセット、タイマー値のレポーター、のブロックがある。タイマーはリセットされるとゼロになり、常時、リセット時からの経過時間を計測する。

### 3.10 対話処理

ステージ上に質問を表示したり、スプライトが質問を発したりして、その答えをキーボードからタイプするような対話処理のブロックには、～と聞いて待つ、答えのレポーター、のブロックがある。

### 3.11 キーの入力検知

キーの入力を検知するブロックには、特定のキーが押されたか、のブロックがある。

### 3.12 マウスのクリック検知

マウスのクリックを検知するブロックには、マウスが押されたか、のブロックがある。

### 3.13 スプライトの属性（動き）の変更

スプライトの属性を変更して、スプライトの動きを変更するブロックには、n 歩動かす、n 度右回転、n 度左回転、回転方向指定、マウスや特定のスプライトの方向に向ける、x,y 座標指定、マウスや特定のスプライトの位置に座標を変更する、x 座標の増減や指定、y 座標の増減や指定、もし端に着いたら跳ね返る、回転方法の指定、x 座標のレポーター、y 座標のレポーター、向きのレポーター、などのブロックがある。

### 3.14 スプライトの属性（見た目）の変更

スプライトの属性を変更して、スプライトの見た目を変更するブロックには、～と n 秒言う、～と言う、～と n 秒考える、～と考える、表示する、隠す、コスチュームを指定したコスチュームに変更する、次の番号のコスチュームに変更する、背景を指定した背景（あるいは前の背景、後の背景）に変更する、色（あるいは魚眼レンズ、渦巻き、ピクセル化、モザイク、明るさ、幽霊）の効果の増減や指定、画像効果をなくす、大きさの増減や指定、一番前に出す、重なりを 1 層下げる、コスチューム番号のレポーター、背景の名前のレポーター、大きさのレポーター、のブロックがある。

### 3.15 音の操作

音を鳴らすブロックには、指定した音を鳴らす、終

わるまで指定した音を鳴らす、すべての音を止める、指定したドラムを  $n$  拍鳴らす、 $n$  拍休む、指定した音階の音を  $n$  拍鳴らす、楽器の指定、音量の増減や指定、音量のレポーター、テンポの増減や指定、テンポのレポーター、のブロックがある。

### 3.16 クローンの生成

スプライトの複製(クローン)を作るブロックには、指定したスプライトのクローンを作る、クローンされたとき、このクローンを削除する、のブロックがある。

### 3.17 定義ブロック

ブロックをサブルーチンとみなし、ブロック(サブルーチン)名と仮引数名を定義するブロックがある。この定義ブロックは値を返さない。再帰呼び出しは可能であるが、再入可能ではない点に注意する必要がある。再帰呼び出しの例として、ヒルベルト曲線、シェルピンスキー曲線の描画や、クイックソートなどがある。また、長いスクリプトを、あるまとまった単位に分割し、それぞれを短い定義ブロックにする場合にも使用される。分割の結果、スクリプトのまとまりが小さくなるので、狭いスクリプトエリアの中でも、視認性がよくなり、修正操作も簡単になる。

### 3.18 リストデータのリンケージ

直接的にファイルの入出力を行うブロックはないが、手動でリスト内のデータを、大量にテキストファイルに入出力することができる。リスト内のデータは、プロジェクト内に記録され、プロジェクトを終了してもデータは保持される。そのため、他のアプリケーションで加工されたデータを Scratch に渡してリスト内に記録することができる。逆に Scratch で計算した結果を他のアプリケーションに渡して、図表化して表示することができる。プロジェクトを保存すると、リスト内のデータも保存される。応用例として、(1)MIDIの音楽データのノートオン情報から、数万の音符を抽出して Scratch に渡し、1時間以上のクラシック音楽を自動演奏する、(2)Scratch でシミュレーションして得られた大量の統計データをエクセルに渡してグラフ化する、などがある。

### 3.19 画像データのリンケージ

Scratch の背景、スプライト、コスチュームに使用する画像は、内蔵する画像ライブラリから選択したり、内蔵するペイントエディタで描画したり、パソコンの画像ファイルから読み出すことができる。この画像ファイルは、他のアプリケーションで作成したものや、インターネットからダウンロードしたものでもよい。

フォトショップやイラストレータで高度に加工された画像ファイルを使用することができる。

### 3.20 音データのリンケージ

Scratch に使用する音は、内蔵する音ライブラリから選択したり、パソコンの音楽ファイルから読み出したり、マイクから録音することができる。この音楽ファイルは、他のアプリケーションで作成したものや、インターネットからダウンロードしたものでもよい。応用例として、音声合成ツールで本物のオルゴールの音色を音階別に合成したものを Scratch に渡して自動演奏し、7 オクターブの音域を持つ仮想オルゴールを鳴らすことができる。

### 3.21 高度な作品作り

Scratch は他のアプリケーションとの間でデータを大量に交換することができることから、高度な作品を作ることができる。(1)人の音声を話すアニメーション、(2)本物の楽器の音色による長時間の音楽の自動演奏、(3)複雑な事象のシミュレーション解析、(4)高度なゲーム、(5)複雑な図形描画、(6)複雑なアルゴリズムの検証などである。初心者は、Scratch を習得した後も継続して高度な作品作りを楽しむことができる。

## 4. Scratch の WEB サイトの交流機能の利用

Scratch で作品を作っても、個人で鑑賞しているだけではつまらない。他のユーザと交流し、作品に対する感想や意見を交換したり、作品を改造したり、作品のコレクションを作ったりしたいと考えるユーザは多い。Scratch2.0 は WEB 上で動作するため、ユーザ間の交流機能が充実している。この交流機能を初心者がどう利用すべきかを考察する。

### 4.1 作品の共有

Scratch の作品(プロジェクト)は共有すると、インターネットに公開され、URL を付与される。その URL にアクセスすれば、誰でも(ユーザでなくても)その作品を鑑賞することができる。誰でも作品の中(背景、スプライト、スクリプト、画像データ、音データ)を参照することができる。初心者は、気に入った作品を参照して、アルゴリズムを模倣することができる。また、自分が共有した作品を参照したユーザ数、お気に入りに入れたユーザ数、好きを付けられたユーザ数、リミックスされた回数、コメント数などを管理することができる。作品のコメント欄には、その作品を鑑賞したユーザから、意見や感想などが寄せられるので、作品の不具合の修正や改良に役立てることができる。

## 4.2 作品のリミックス

作品は他のユーザがリミックス（改造）することができる。他のユーザの作品をリミックスすると、自分の作品としてコピーされ、作品の中を自由に変更できるようになる。リミックスした作品名は、最後尾に「remix」の文字が付加され、元の作品と区別される。初心者には、この機能を利用して、気に入った作品を自由に改造して、自分の作品にすることができる。

## 4.3 作品のスタジオ

作品をジャンル別、グループ別、用途別、目的別などに分けて、作品のコレクションを作ることができる。これはスタジオと呼ばれ、好きな作品をスタジオに入れることができる。スタジオに作品を追加できるのは、マネージャとキュレータのみに設定することができる。マネージャはスタジオを作ったユーザであり、キュレータはマネージャからスタジオに招待され、参加することを了承したユーザである。初心者には、この機能を利用して、初心者の同好会のようなグループを作り、メンバーに対して、自分が作った作品を紹介したり、コメント欄で感想や意見を交換することができる。キュレータになると、スタジオに作品やコメントが追加されたとき、自動的に通知メッセージが送られてくる。

## 4.4 注目のプロジェクト

MIT の Scratch チームは、1,460 万以上の作品の中から、毎日 1 作品を選び、WEB サイトのトップページに掲載している。この作品は注目のプロジェクトと呼ばれ、20 日間掲載される。注目のプロジェクトは、次の基準で選ばれる。

- (1) Scratch の革新的な使い方を示したプロジェクト
- (2) 共同作業やリミックスをうながすプロジェクト
- (3) コミュニティの多様性（性別、年齢、文化、国、関心など）を示すプロジェクト
- (4) 様々な人々が楽しめるプロジェクト

作品が注目のプロジェクトとしてトップページに掲載されると、その作品は世界中のユーザに注目され、多数の人々に参照されたり、お気に入りになれたり、好きを付けられたり、コメントが寄せられたり、リミックスされたりする。現在、注目のプロジェクトに選ばれた日本人は、筆者を含め 12 名しかいない。その作品のほとんどは英語で表記されている。5 月 21 日に東京大学本郷キャンパスで、Scratch day 2016 in Tokyo が開催され、Show & Tell のイベントにおいて、筆者が作成した注目のプロジェクトの作品の紹介と説明を行った。初心者には、注目のプロジェクトを参考にして、作品のレベルを向上していくことができる。

## 4.5 ディスカッションフォーラム

初心者には、日本語のディスカッションフォーラムを利用して、質問を發し、知識を持つユーザからの回答を得たり、自分の作品を宣伝することができる。

## 4.6 ユーザプロフィール

初心者には、ユーザプロフィールに、自己紹介、自分が取り組んでいることなどを記入し、自分がどのように Scratch の WEB サイトで活動しているかを他のユーザに知らせ、世界中のユーザと知り合うことができる。

## 5. いわき市における小中学生プログラミング体験教室の事例

福島県いわき市において、小中学生を対象に、Scratch プログラミング体験教室を実施した。テキストは筆者が作成し、例題は Scratch の WEB サイトに共有されている作品を使用した。1 回 4 時間完結で、5 月 7 日、8 日の 2 回、いわき市生涯学習プラザにおいて実施した。1 日目は 8 組、2 日目は 6 組の子共と、その保護者が参加した。パソコンを 1 人 1 台使用し、例題の演習を行った。ユーザアカウントは、教室用に、あらかじめ作成したものを使用した。実施内容は次のものである。

- (1) ゲーム、音楽、アニメーションなどの作品の鑑賞
- (2) 10 種類の種別毎の全ブロックの動作確認の演習
- (3) NHK E テレで放送された 5 つの例題の演習
- (4) 妖怪たたきゲームの演習
- (5) 自由作品の作成、共有および当日の教室用スタジオへの登録
- (6) いわきスクラッチクラブの紹介
- (7) Scratch 2.0 オフラインエディターの紹介

### 5.1 例題で使用した作品

例題で使用した作品は、7 種類の教材用スタジオに用途別に分類して登録し、順に演習していく方式を採用したため、多数の例題をスムーズに演習することができた。

妖怪たたきゲームは、完成版のプロジェクトから、主要スクリプトを除去したプロジェクトを与え、テキストを見ながらスクリプトを追加し、プロジェクトを完成させる例題である。このプロジェクトを作成するに当たり、使用する英字は、「time」、「point」、「set」、「end」のみ、日本語は「当たり!!」、「はずれ!!」のみであり、それぞれ 1 回しか入力しないので、子供でも簡単に作成することができる。今回は時間の都合上、妖怪たたきゲームの作成は割愛し、完成版のプロジェクトの解説と、ゲームの鑑賞のみを行った。

## 5.2 教室用スタジオ

教室用スタジオは、当日演習した作品を集めたスタジオである。そのスタジオの中に、当日の参加者が作成し、共有した作品を追加した。参加者が自分のユーザアカウントを作成後、そのスタジオを参照し、自分の作品をリミックスすることができる。

## 5.3 いわきスクラッチクラブのスタジオ

いわきスクラッチクラブのスタジオは、教室の参加者が継続して利用することができるスタジオであり、講師がマネージャ、参加者がキュレータになる。教室の終了後に、参加者が作った作品を追加したり、今後のステップアップ教室の連絡などに利用する。

## 5.4 オフラインエディター

Scratch2.0 オフラインエディターは、Scratch の WEB サイトが混み合って応答が遅いときに、オフラインでプロジェクトの編集を行うプログラミングツールである。WEB サイトのプロジェクトをパソコンにダウンロードして、オフラインエディターで編集することができる。オフラインエディターで編集したプロジェクトは WEB サイトにアップロードすることができる。

## 6. 今後の課題

Scratch2.0 の現状の問題点と今後の課題を考察する。

### 6.1 日本語への対応

Scratch2.0 の WEB サイトは英語表記されているページが多い。一部プロジェクトの作成に必要な編集画面は和訳されているが、ヘルプ、マニュアル、例題プロジェクトなどは、英語表記のままである。これらを和訳する必要がある。

筆者は Scratch1.4 Reference Guide を和訳して、教室の参加者に配布している。

### 6.2 楽器の種類と音色

Scratch2.0 では、楽器の種類は 20 種類しかなく、楽器の音色は本物の楽器とは異なっている。鳴らしたい音色がなければ、別途ユーザが音色を合成して作成しなければならない。筆者は、本物のオルゴールの音色を合成して作成し、作品で演奏用の音に使用している。

### 6.3 ユーザインターフェース

ボタン、チェックボックスなどのユーザインターフェースに使用するコントロールの画像は用意されていないので、別途ユーザがコントロールの画像を作成し

なければならない。筆者は、使用するボタンの画像を作成し、作品でボタンの表示に使用している。

## 6.4 標準関数

演算ブロックで用意されている関数以外の標準関数は用意されていないので、別途ユーザが定義ブロックを使用して作成しなければならない。筆者は、クイックソートの定義ブロックを作成し、作品でデータの並べ替えに使用している。

## 6.5 プロジェクトの情報の印刷

プロジェクトの情報を印刷する機能は用意されていない。このため、画面上で目視するしかないが、スクリプトエリアが小さいので見難いことがある。また、文字列検索機能がないので、変数の使用箇所を検索することが困難な場合がある。ある変数への代入をどのブロックで行っているかを検索することは、アルゴリズムの検証上、重要であるが、Scratch では目視で行うしかない。筆者は、必要に応じて、画面のハードコピーを取り、対応している。

## 6.6 配列の次元数

リストは 1 次元配列しか扱えないので、作品作りにおいては、工夫が必要である。筆者は、複数のリストの名前に番号を付け 2 次元配列のように使用している。

## 6.7 プロジェクトの復元

スクリプトの上書き、移動、複写、挿入をした場合の復元はできない。削除のみ復元できる。すべてを復元するには、プロジェクトをパソコンのファイルへバックアップする必要がある。筆者は、必要に応じて、プロジェクトのバックアップを行っている。

## 7. まとめ

Scratch を初心者のプログラミング教育に使用する利点や、高度な作品作りの可能性を考察してきた。しかし、一般には Scratch は子供向けのプログラミング機能に限定され、簡単なことしかできないと思われている。これは、Scratch に関する詳細な日本語マニュアルがないことが一因である。Scratch の世界に入ってきて、日本語マニュアルが見つからず、困っている人がいる。マニュアルやヘルプや例題プロジェクトなどを和訳し、正しく Scratch を理解してもらうことが必要である。そうすれば、子供だけでなく、大人も Scratch に関心を示すようになるであろう。

Scratch は作品を作る労力が、ソースコード記述型言語と比較して大幅に小さく、短時間に作品を完成することができる。そのため、作品作りが楽しくなり、も

っと面白い作品を作りたいという意欲が湧く。

Scratch を習得した世界中の子供たちが、WEB サイトにおいて多くの作品を共有している。その作品は質・量ともに驚くべきものがある。日本においてもScratch がより普及し、多くの初心者が自由に作品作りを楽しむようになることを期待している。

## 文 献

- [1] 阿部和広, “子供の創造的活動と プログラミング学習”, 情報処理学会誌, vol.57, no.4 pp.349-353, Apr.2016.
- [2] 阿部和広, “小学生からはじめるわくわくプログラミング”, 日経 BP 社, 東京, 2013.
- [3] N.ビルト, “アルゴリズムとデータ構造”, 浦昭二, 國府方久史共訳, 近代科学社, 東京, 1990.



図 1. Scratch2.0 の編集画面

ブロックパレット スクリプトエリア コメント

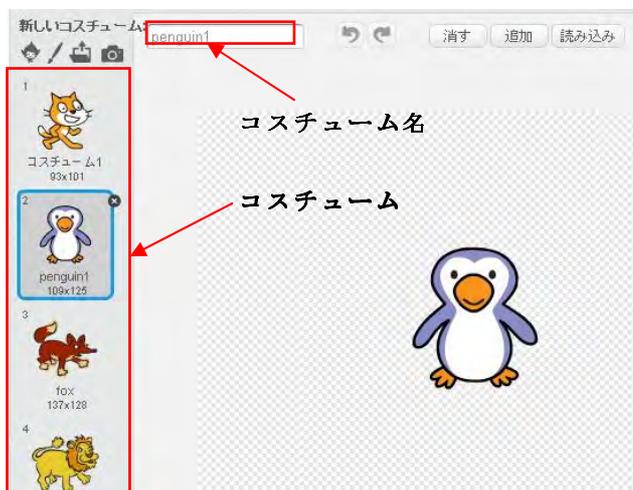


図 2. コスチュームの編集画面

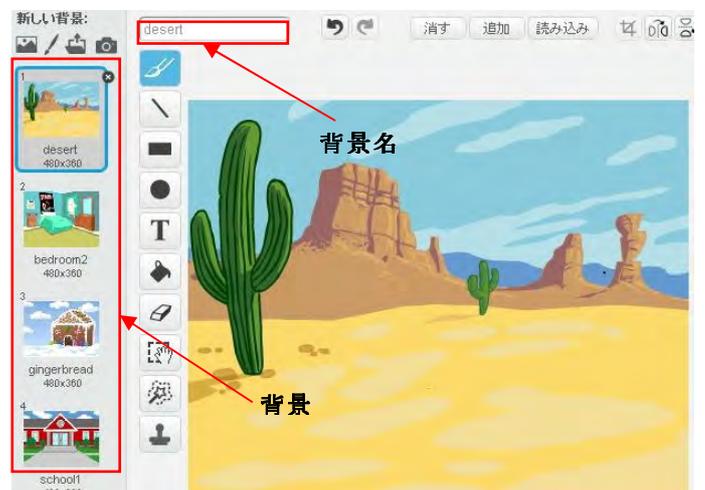


図 3. 背景の編集画面



図 4. 音の編集画面

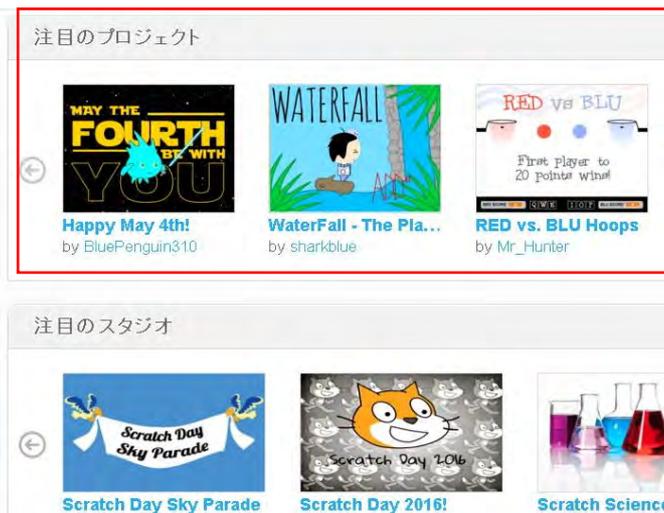


図 5. WEB サイトの注目のプロジェクト



図 6. 筆者が作成した注目のプロジェクト

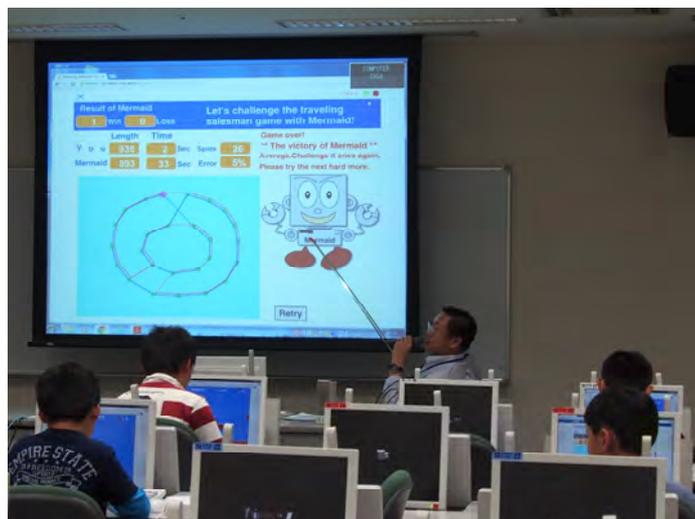


図 7. 小中学生プログラミング体験教室の風景



図 8. スタジオに集めたオルゴール作品の例