

小学校段階における Scratch と JavaScript を用いた
プログラミング教材開発と授業実践

2021年2月15日

三重大学大学院教育学研究科

教育科学専攻 理数・生活系教育領域

山守研究室

219M015 倉知孝拓

目次

第1章 はじめに	1
1.1 研究背景	1
1.2 研究目的	1
1.3 研究概要	2
第2章 小学校段階でのプログラミング教育の提案	4
2.1 小学校段階 6年間を見通したプログラミング教育の提案	4
2.1.1 パソコン操作能力について	4
2.1.2 年齢に適した題材や学習方法	5
2.1.3 学年段階ごとのプログラミング授業題材例と配当の仮説	5
第3章 他研究との比較	7
3.1 他研究と本研究の比較	7
3.2 SCRATCH を用いた研究	7
3.2.1 Scratch を用いた小学生向けお絵かきの授業実践	7
3.2.2 Scratch を用いた図形理解を深める授業実践	7
3.2.3 Scratch を用いた小学校プログラミング授業の実践～小学生を対象としたプログラ ミング教育の再考～	8
3.2.4 小学校2学年におけるプログラミング教育の実践～スクラッチ Jr を活用したアニメ ーション物語の作成～	9
3.2.5 プログラミングの体験形式がプログラミング学習の動機づけに与える効果	10
3.3 JAVASCRIPT を用いた研究	11
3.3.1 JavaScript を用いた中学生向けプログラミング教育	11
3.3.2 プログラミング言語を習得するための JavaScript を使用した WBT 教材の開発と実 践	11
3.4 本研究との比較	12
3.4.1 Scratch を用いた研究の比較	12
3.4.2 JavaScript を用いた研究の比較	12
第4章 Scratch を用いた「ひらがな練習ゲーム」	13
4.1 題材概要	13
4.2 ステージ構成とゲームの流れ	13
4.3 題材の詳細	14

4.3.1	ペン機能を持たせたねずみスプライト	14
4.3.2	ねずみスプライトのプログラム	14
4.3.2.1	お手本からはみ出た回数を数えるカウンター	14
4.3.2.2	ねずみにペン機能を持たせるプログラム	15
4.3.3	お手本文字の表示方法	16
4.3.4	結果発表に表示されるキャラクター	17
4.3.5	提出ボタンについて	19
4.3.6	文字選択画面の文字スプライト	19
4.4	今後の課題	20
4.5	本題材のまとめ	21
第6章 Scratch を用いた「魚釣りゲームプログラミング」		22
6.1	題材の概要	22
6.2	題材開発の経緯	22
6.3	指導の手順	22
6.4	題材の詳細	25
6.4.1	待機画面	25
6.4.2	用意したスプライトとプログラムの説明	25
6.4.3	結果発表画面	29
6.5	題材の工夫した点	31
6.6	授業で工夫する点	31
6.7	題材のまとめと課題	32
5章 Scratch を用いた「ローマ字タイピング練習ゲーム」		33
5.1	題材概要	33
5.2	開発の経緯	33
5.3	題材の詳細	34
5.3.1	スタート画面	34
5.3.2	ステージ選択画面	34
5.3.3	ゲーム画面	35
5.3.4	クリア画面	36
5.3.5	リスト機能を用いた問題の作成	38
5.3.6	出題プログラム	39
5.3.7	正誤判定をするためのプログラム	40
5.3.8	メッセージブロックによるプログラム間の連動	41
5.4	問題に使用した語と正答のファイル	42
5.5	本題材の今後の課題	44

第7章 Scratch を用いた「めいろゲームプログラミング」	45
7.1 実践概要	45
7.2 授業の流れ	45
7.3 TIGHTVNC の利用	49
7.4 実践で工夫した点	49
7.5 授業における児童の様子	50
7.6 まとめ	50
第8章 Scratch を用いた「一筆書きで図形描画プログラミング」	51
8.1 題材概要	51
8.2 題材の詳細	51
8.3 各図形を描く流れ	54
8.4 指導のポイント	58
8.5 まとめ	58
第9章 JavaScript を用いた「おみくじゲームプログラミング」	59
9.1 実践概要	59
9.2 授業の流れ	60
9.3 授業で工夫した点	70
9.4 授業における児童の様子	70
9.5 今後の課題とまとめ	70
第10章 Scratch を用いた「花模様を描くプログラミング」	72
10.1 実践概要	72
10.2 実施内容	72
10.3 配布したプログラムの説明	74
10.4 授業の流れ	78
10.5 授業で工夫した点	97
10.6 TA によるサポート	98
10.7 授業アンケート結果と考察	99
10.8 授業で分かった反省点と改善について	103
第11章 全体の考察	104
第12章 まとめ	106
12.1 本研究の成果	106
12.2 今後の課題	106

参考文献.....	108
謝辭.....	110

第1章 はじめに

1.1 研究背景

現在日本は産業、グローバル化等変化の激しい状況にある。超スマート社会(Society5.0)構想における、あらゆる物事の情報化という側面から見ても私たちの生活に ICT 機器の導入が進められている。そのことにより、ますます情報活用能力やプログラミング的思考力等の、変化の激しい時代を生き抜いていくための力を身に付けることが必要になると言われている。そのような経緯から 2020 年より小学校でプログラミング教育が実施されているが、未だに教育現場ではどのような活動すれば良いのか悩ましい問題がある。在籍する子どもは「パソコンを初めて触る子どもから、家庭でよく操作する子ども」「プログラミングは学校での授業で初体験という子どもから、習い事でやっている」等パソコン操作やプログラミングの経験について、児童それぞれで様ではない。そのため、パソコン操作に慣れつつ初学者にとってプログラミングを学びやすい方法で、授業が展開されることが求められる。

また、GIGA スクール構想の実現によって令和 5 年度までには小中全校において、児童生徒 1 人 1 台にコンピュータが割り当てられる予定である。このことから、教育は ICT 機器を取り入れたスタイルに変わっていくことが予想できる。そのためにも、より一層低学年の早い段階からパソコン操作へ馴染み、パソコンを学習へ有効利用するための努力が求められる。

これまで ICT 機器を授業で頻繁に扱っていなかった児童にとって、パソコンを用いたプログラミングの授業は興味を惹きやすい題材の 1 つと成り得る。子どもの適度なコンピュータの利用はプログラミング的思考の育成に繋がると考えられるため、学校以外の所でもコンピュータを利用しようという意欲を高めることが出来るような授業展開が理想である。

1.2 研究目的

(1) 教育現場の現状

すでに教育現場ではプログラミング教育が各教科・教育活動の中で行われているが、現状では多くの教師がどのような活動を通してどのように指導をすれば良いのか未だに戸惑いを感じている。また題材の開発に負担を感じている教師もいるだろう。実践をさせて頂いた学校の中には、利用が推奨されている Scratch の使い方をあまり分かっていない様子も見られた。そのため、プログラミング教育として開発を行ってきた授業題材を紹介し、どのようにプログラミング的思考力を育てていくのかその一例を示すことで、現場で働く教師の一助とするために研究を行った。

(2) 6 年間を見通したプログラミング教育

前述したように授業でプログラミングをする子どもの、パソコンやプログラミングへの慣れ方は一様ではない。よって小学校段階では、低中高それぞれの学年に適した題材によって、パソコン操作能力とプログラミング的思考力の両方を育成していかなくてはならない。

またパソコン操作以外にも、児童は学年段階ごとに考えたり取り組んだりできる物事の範囲が異なる。低学年である程抽象的な思考は難しいと言われている。そのため、出来る限り児童が経験したことがある具体的な遊びや活動を題材にして低中学年では扱っていくべきである。

(3) 用いるプログラミング言語

Scratch はスクリプトに対して与える命令コマンドがブロックになっており、1つひとつのブロックを繋げることでプログラミングを行うビジュアルプログラミング言語である。既に用意されている意味をもったブロックを順序立てて並べるだけという点において、複雑なプログラミング言語の動かし方や特徴に悩まされることはなくなる。また、作ったプログラムの動作を同じ画面で確認が出来るため、実行結果が明確に分かり抽象度は低く、低学年の児童でも比較的扱いやすいと考えられる。

プログラミング言語はコンパイラ等を必要とし、家庭で全員が行うには環境設定の観点から難しいものが多いと言える。しかし、今回提案する Scratch と JavaScript を用いたプログラミングは、大がかりな環境の整備を必要としないため児童に興味をもたせることが出来れば、主体的に家庭でコンピュータを利用してプログラミングに携わる足がかりとなる効果が期待できる。

以上のように、児童の発達段階やプログラミングに対する関心や理解度を踏まえたうえで、授業題材を各学年別に考案し当てはめる。児童にプログラミングに興味をもたせ、逐次処理や繰り返し処理、条件分岐処理等のプログラミングにおける要素を学び、最適な解を考えることでプログラミング的思考力を養わせることを目的とした。

1.3 研究概要

以上の背景と目的を踏まえて、各学年別にプログラミング教育に用いる題材を提案する。

まず低学年では第 1 学年向きにマウスを用いてひらがなを練習するツールを用いて、マウスの操作に慣れさせる。第 2 学年では魚釣りを題材に簡単なプログラミングの要素を使った考え方を身に付ける。次に中学年では第 3 学年にてローマ字タイピングゲームでキーボード操作に慣れながら、迷路ゲームプログラミングを用いて、迷路をクリアするためにブロックを繋いで逐次処理の考え方を身に付ける。第 4 学年では一筆書きプログラミングによって、プログラミング的思考力を育む。最後に高学年では、第 5 学年でテキストベースのプログラミング「おみくじゲーム」の作成に挑戦する。第 6 学年で幾何学模様を描く先行研究内容の発展として「花模様を描くプログラミング」に取り組むストーリーを考えた。以上で説明した各学年において提案する題材の詳細については、第 4 章以降にて述べていく。

本研究では、題材の開発だけでなく、授業実践も行っている。3年生向き「めいろゲームプログラミング」、5年生向き「おみくじゲームプログラミング」及び6年生向き「花模様を描くプログラミング」は、小学校現場で実践を行った。それらの実践の詳細は、題材の詳細を述べている各章の中で述べる。

なお、プログラミング能力の育成のために、文部科学省が使用を推奨している Scratch を用いて授業開発を行っているが、子どもの成長を見ながら JavaScript を用いて授業を行うことも想定している。JavaScript を用いた題材は、「おみくじゲームプログラミング」であり、第4章で詳細に述べる。

第2章 小学校段階でのプログラミング教育の提案

2.1 小学校段階 6年間を見通したプログラミング教育の提案

小学生に対してプログラミング教育が実施されているが、どの学年の児童にも適当であるということが出来る教材や学習内容はないだろう。教材を選ぶ際には少なくとも児童のその年齢の子どもに適した方法と学習内容でプログラミング的思考力を育むことを考えなければならない。また、児童にコンピュータの特性を理解させることもプログラミング教育の目的に位置づけられていることから、パソコンを使った授業は必然的に増える。そのため、担当児童のパソコン操作能力が備わっているかを把握し、授業中の作業に滞りのないように学習活動を決定しなくてはならない。

2.1.1 パソコン操作能力について

児童のパソコン操作については、文部科学省の平成 25～26 年度のアンケート調査によれば、国公立の 5 年生児童 3343 人へ行った質問で、「キーボード入力は得意であるか」という質問に対し肯定・やや肯定的な回答を示した児童は 61.2%であった。しかし、筆者はこれまでに複数の実践協力校で 1 年生から 6 年生までを授業者・TA の立場で参加させて頂いたが、低学年の児童はもちろん遅く、中高学年においてもキーの位置を覚えていないことや、ローマ字入力のアルファベットの適切な組み合わせも確実に暗記しているわけではないため、正確さと速度の観点でいえば一般的な速度よりやや遅いと感じる児童が多かった。次に、マウス操作に関しては低学年児童では左クリックで対象をドラッグしたまま、カーソルを移動させることの出来る子どもは殆どいなかった。マウスを使ってお絵かきをする授業実践に TA として参加した際も、児童の握るマウスの軌道は直線的で滑らかなカーブを描くことに困難そうな表情を浮かべていた。また、ドラッグによって描くのだが途中でクリックしている指を離してしまっていた。この場合の解決策は 2 つあり、1 つ目は手を握って補助をしながら作業を進めること。2 つ目は一気に大部分を描こうとするのではなく、少しずつ部分的に描いていく方法を勧めることであった。現場で教師として担任となれば、基本 1～2 人で多いとき 30～40 人の児童を見なければならない。その場合アドバイスとして 2 つ目の解決策をクラスの全児童に教えることが現実的である。しかし、ある程度のドラッグをする力加減と感覚を身に付ける必要があるため、マウスのコントロールトレーニングも必要である。授業実践へは十数名の TA が参加することが多く、その理由の大部分を占めている要素が、児童の使い慣れていないパソコンやタブレット端末の操作補助であると感じている。そのため、低学年段階からパソコンへ慣れるための活動を行うことは必須であると考えられる。

2.1.2 年齢に適した題材や学習方法

文部科学省は『子どもの発達段階ごとの特徴と重視すべき課題』で高学年は抽象的な思考の次元への適応が課題となる時期であると述べている。プログラミング教育ではアルゴリズムを考えたり理解したりして、実現するためにプログラムを作成する。その方法がパソコンを使った物であればやはり抽象度は具体物を使うよりも高くなる。さらにそのパソコンを使った方法でも、ビジュアルプログラミングであるのかテキストベースのプログラミングであるのかでは、また抽象度は変わってくる。テキストベースであれば、普段使う日本語によるコーディングではないため、初学者にとっては構造を把握するだけでも難しく感じるだろう。複数のソースファイルからなるプログラムもあり、複雑さや抽象度もます。よって、担当児童がどれ程の思考が可能であるかを日頃の授業・生活場面を観察して把握することで、適切な題材と学習方法を選択しなくてはならないと考える。

2.1.3 学年段階ごとのプログラミング授業題材例と配当の仮説

この章では①児童のパソコン操作能力②児童の発達段階の2つの視点からプログラミング教育で実施する題材や学習方法を設定すべきだと述べてきた。

パソコン操作能力の向上について、文部科学省は低学年でマウス操作、中学年でタッチタイプ能力を身に付けることを推奨している。このことを踏まえて、第1学年でマウスを使ったひらがな練習、第3学年でローマ字タイピング練習の題材を位置付ける。基本的にキーボード入力を行う際はローマ字入力であるため、第3学年の国語でローマ字を学習するタイミングで始めると良いだろう。

次にプログラミング教育に関する授業題材について、低学年の第2学年で魚釣りを題材にしたゲーム作りで、ゲーム画面での魚の動きを具体的にイメージながら繰り返しや条件分岐を組み合わせて考える。並行して具体物を使うことや、日々の学習内容の思考過程をふり返って考えることも良いだろう。例えば、筆算の計算の手順を1つひとつ言語化し、並べてフローチャートを作ってみる。くり上がりの有無によって分岐が生じることで、場合分けをして普段計算していることに子どもが気づく実践例もある。繰り返しや条件分岐の考え方に第2学年で気づき、活用していくことが出来る段階であると考えた。

次に中学年では、簡単なプログラムをScratchのようなビジュアルプログラミング言語で作ってみる活動である。私の研究している題材でいえば、第3学年で「めいろゲームプログラミング」で「前へ進む」「右へ向く」「左へ向く」の3つのブロックを組み合わせゴールまで辿り着かせる。第4学年ではそれまで身に付けた逐次・繰り返しを使って与えられた図形を短く簡潔なプログラムで、重複のないように解答を考えることを一筆書きプログラミングで学ぶ。

最後に高学年では、少しレベルを上げた課題に取り組んでも良いだろう。この学年段階までプログラミング的思考を身に付け育ててきた児童ならば、第5学年でJavaScriptのようなテキストベースのプログラミングに挑戦すると、抽象的な思考に働きかける機会とな

り、さらなる成長に結びついていこう。JavaScript プログラミング「おみくじゲーム」を作る授業実践では、ステップアップの課題で併用していくと良い。ただし全て打ち込んで完成させるのは難しいので、初期プログラムを配付して、部分的に書き換えさせる等の時間短縮・作業負担軽減の工夫を施すことが望ましい。またこれまでに実践例が挙げられている吉原健人氏のペン機能と回転ブロックを使った、幾何学模様を描く授業が第5学年で実施されていた。そこでさらに発展させた内容の「花模様を描こう」教科内容の延長線にある課題を第6学年に提案する。線を引いて屈折する回数を増やすことで、滑らかな線に近づけて変えていく考え方は、屈折回数が増えると切片を構成する線分の長さが短くなる関係をイメージする難しさ等を含めて、第5学年より上の学年に位置付けた。

以上の授業題材の学年への割り当てについて、実践を行ったものは児童からの感想やTAの所感、アンケート等を踏まえて考察を行った。実践が行うことが出来なかったものは、先行研究や参考資料等から再検討の必要性について考えていく。

第3章 他研究との比較

3.1 他研究と本研究の比較

これまでに小学校段階でプログラミング教育必修化に向け、扱うプログラミング言語や指導法、指導内容等の多くの研究が為されてきている。本項で、これまで行われてきた研究の概要を確認し、これから筆者が行うべき研究の方針を定めていく。それらの研究と筆者が行っている研究内容を比較し、未だ研究されていない側面からプログラミング教育の研究にアプローチできるように役立てていく。

3.2 Scratch を用いた研究

3.2.1 Scratch を用いた小学生向けお絵かきの授業実践

Scratch を用いた小学生に対するプログラミング授業実践である。2020 年より始まるプログラミング教育に対し指導方法や内容について戸惑いを見せる現場教員の一助とするため、小学校 1 年生向けに行った授業実践の報告を載せている。

パソコン操作に慣れさせることを主な目的として、パソコンを通したお絵かきを題材とした実践を紹介している。2 年間をかけそれぞれ吉原健人氏が Scratch を用いたお絵描きの授業実践(2016)で「水族館をつくろう!」、稲垣諒氏が Scratch コスチューム画像のサイズ均一化ツールの開発(2017)で「サバンナをかこう!」の 2 テーマで行った。水族館をつくる授業では、児童に魚やタコ等の海の生き物を描かせた後授業者のパソコンにスプライトを送り、それらを 1 つの Scratch のプログラムで動かすというものである。専用のボタンを作成し、海の生き物が整列して動いたりランダムに動いたりすることを切り替えることが出来る。移動している絵の大きさは意図的に大きくなったり小さくなったりするようにプログラムで設定してあり、子ども達が自分の絵が目立つ動きをする姿を見て盛り上がっていた。「サバンナをかこう!」では奥行きによって絵の大小が変わる姿を見て、もっと描きたいと思った児童が複数のサバンナの動物を描こうとする姿が見られたことから、子ども達のパソコンを使って絵を描き動かす魅力楽しさを感じさせることが出来たと考えられる。

3.2.2 Scratch を用いた図形理解を深める授業実践

2017 年に 5 年生向きには吉原健人氏が、プログラミング初歩を体験させることを主な目的として、「星を描こう!」をテーマに逐次処理と繰り返し処理によって、回転ブロックの角度とペン機能を使い星形の図形や幾何学模様を描く実践を行っている。45 分という短い時間で着手させるために、サンプルプログラムを作り配付しておき目標とするプログラムの形に近づけていくという工夫によって、多くの児童が星形を完成させ第 2 の課題である幾何学模様の作成に熱心に取り組んだ。直感的にプログラムを触る児童が多く、偶然出来た幾何学模様を見て盛り上がりプログラムの並びについて尋ね合って活発に活動してい

た。一方で、この時に意図的にプログラムを組んで試行錯誤しイメージした図形に近づけていこうとした子どもがどれ程の人数いたのかは分からない。吉原氏は作成したプログラムの中身について理解しているのかということについて、気がかりである。そのため、実際に学校現場で行う際には確実にプログラムの内容を確認しながら作るということも重要視して評価していく必要性が分かったようである。

3.2.3 Scratch を用いた小学校プログラミング授業の実践～小学生を対象としたプログラミング教育の再考～

2011年に発表された森秀樹氏らの Scratch を用いた小学生に対するプログラミング授業実践である。1980年代の後半はコンピュータの普及が進み、「Logo」等を利用したプログラミング教育が試みられていたが、その高度な内容より指導が難しくかつ子ども達の興味を惹くことが難しいことを理由に、2000年代にはプログラミングを通じた教育活動は、ほとんど行われなくなっていた。そのような背景があり、初学者向けに開発された Scratch を用いて、児童に対して適切にプログラミングを導入することを目的としている。具体的には(1)命令コマンド(ブロック)の基本的理解(2)プログラミングに対する興味を育むことが目的である。

実践は男女各 19 名(計 38 名)の 4 年生を対象として、26 時間に渡り行われた。26 時間は大きく分けて①Scratch コマンド探し②図形を描く③作品作り 1 回目④作品作り 2 回目の 4 つの過程で構成されている。①②は Scratch を探索的に進め練習課題である。③の作品づくりは、作品の画面イメージと目標を立て制作する。完成作品を Scratch サイトに挙げると、多くのコメントが寄せられたため児童へ紹介した。④の作品作りでは 4,5 名のグループを組ませ、センサボードを使った作品作りをする。センサを動かす過程で条件設定を行うため、不等号などの記号の説明も行っている。

完成した作品を総合し、評価を行った。作品の規模として 1 回目の作品は平均ブロック数 39.7 個に対し、2 回目の作品の平均ブロック数は 32.7 個と減少した。2 回目はセンサを用いるため 1 回目 비해条件を設定するための「調べる」「数」ブロックの使用数は増した。このことから、センサの扱いや特徴も理解していることが分かる。

児童に行ったアンケートによると、「スプライト制御」「繰り返し」「座標」「キー入力の判別処理」「条件分岐」「スプライト間の連携」について、それぞれ高い理解度が示された。理解度とブロックの利用度をまとめた表から、基本的にブロックの利用率が高いほどそのブロックに対する理解度が高い傾向があることが読み取れる。また、5 段階の「5：楽しかった～1：つまらなかった」の評価を付けさせると平均 4.78(SD 0.76)の高い評価を得た。合わせて行った難易度を調べる 5 段階アンケート「5：簡単だった～1：難しかった」では、平均 3.22(SD 1.27)と必ずしも簡単とはいえない内容であったようである。

以上の結果から(1)命令コマンド(ブロック)の基本的理解(2)プログラミングに対する興味を育むといった 2 観点では、Scratch を用いて目的を達成できたと言える。実践を終え

た後の課題として、「より少ない時間で行えるようにしていくこと」「他学年向けの内容も考えること」「学校教員だけでも取り組めるように研修や支援体制の検討」等が挙げられている。

3.2.4 小学校2学年におけるプログラミング教育の実践～スクラッチ Jr を活用したアニメーション物語の作成～

2018年に発表された田中良研氏らの研究である。活用したプログラミング言語はスクラッチ Jr と呼ばれる子どもの発達支援のために開発された低年齢(5～7歳)向けのアプリである。スクラッチ Jr は画面上の処理内容が記された命令コマンドを組み合わせることによって、対話形式の物語やゲームが作ることが出来るようになっている。他にも撮影した写真や録音した音声を取り込んで作品に使用することも出来る。

筆者は実践校の2年生10人を対象に国語「お話のさくしゃになろう」の単元で授業を行った。読者に伝わりやすいように試行錯誤しながら場面を順序立てて考えることがプログラミング的思考の習得に考えて、国語の科目を選択している。実践の流れとしては①課題「4コマからなる、ねこの休日を考えよう」の説明②ワークシートによる物語構成③iPadでスクラッチ Jr を通した物語作品の作成④作品の紹介と発表の4つの活動となっている。授業の狙いは3つ。1つ目は「試行錯誤」とし、最初から完璧なものを作ろうとするのではなく、少しずつ手直しをしていくことで完成へと近づけていくこと。2つ目は「コンピュータの仕組み」とし、コンピュータを思い通りに動かすためには考えたことを的確に伝える必要があることを知ること。3つ目は「プログラミング的思考」とし、スクラッチ Jr を使って物語を作成する活動を通してプログラミング的思考の「分解」と「順序立て」の考え方のイメージをつかむことができることとしている。

授業は2人1組のペアで作品作りを行う。ペアで活動することによって、物語構成について話し合ったりスクラッチ Jr の動かし方について分からない点について助言し合ったりするなど、積極的な話し合いを中心に活動が展開されていた。この点で、初学者に向ける学習形態のヒントがあると感じた。初めて動かすツールについて個人で試行錯誤しては、分からないことが多すぎて活動時間に限界がある。そのような時に、複数人で1つの目標に向かって取り組むことで、それぞれで分かった情報を共有し合うことで問題解決を迅速に行うことが出来るというメリットがある。状況ごとに応じた学習形態の比較はここでは検討されていなかったが、課題の難易度や学習者の熟達度によって学習形態を踏まえた授業計画を立てることの重要性が大きいだろう。

授業で作成されたワークシートには、スクリプトである猫がどこで何をするのかということや作品の見どころが、簡潔に書かれていたことから正確に物語の手順について説明する力が身に付いたようである。感想ではタブレット操作の難しさや自分の考えたとおりに猫スプライトを動かす難しさ、台詞を考えることへの苦労などが挙げられた。パソコンを使えば4コマに動きが生まれ表現の幅が広がる。一方でプログラムを組む苦労もあるとい

う本質的な部分も学ぶことが出来たのだろう。

3.2.5 プログラミングの体験形式がプログラミング学習の動機づけに与える効果

2017年に発表された岡崎善弘氏らによる研究である。プログラミング体験イベントを開催し、体験形式を3用意することで、参加した児童生徒の興味関心への効果を検証している。また、子どもがプログラミング教育の3段階(動機付け,継続的な学習,深化)を歩み続けるために、Eccles et al.(1983)の期待価値理論に注目しイベントを通して、今後もプログラミングは上手く出来そうだという期待と、プログラミングに対する価値を高めることが重要であると考えた。

動機付けを検証する3つの体験形式は以下の3つである。

- ①講義型：逐次的にプログラミングを教えるため、全員の進捗が揃っていることを確認してから次の教示が行われる。21名の小学生が参加した。
- ②協同学習：子どもが協同しながらプログラミング学習をする。21名の小学生が参加した。
- ③個別型：子どもが1人でプログラミング学習をする。20名の小学生が参加した。

体験イベントは62名の小学3,4年生が3日に分かれて参加しScratchを用いて行われた。Scratchの操作説明を行うチュートリアル(約20分)とシューティングゲーム作成の本課題(約40分)で構成されている。プログラム完成後は約20分間、予め用意されていた弾の速さや弾のオブジェクト、自機・敵機の移動速度等をカスタマイズする時間が設けられている。本課題の内容は自機と敵機へのプログラミングであった。自機へは方向キーと自機の連動、スペースキーを押すことで弾が発射される、敵機の3度触れるとゲームが終了する3つのプログラムを加えた。敵機へは画面上をランダムに動く、自機に触れるとダメージを与える、自機が発射した弾に当たると消える、消えると1秒後に再び現れるように4つのプログラムを加える課題であった。協同型と個別型に参加した児童へは、プログラムの作り方を見開き1頁のテキストにまとめて配付している。

アンケート回答の分析の結果、事前事後の成功期待・課題価値の変化の観点から、講義型と協同型が初学者にとっては望ましいということが示唆された。特に協同型では、自発的な受講生同士の助け合いで問題を解決することが、子どもの学習内容に対する意識に強く関わったと予測できる。学習者のレベルに合わせてどのような場で学ぶことが適切なのかも、考慮すべき条件となると岡崎氏の研究は示唆している。

3.3 JavaScript を用いた研究

3.3.1 JavaScript を用いた中学生向けプログラミング教育

2018年に発表された、中学生が使うことを目的として開発された JavaScript を用いたプログラミング教育教材の研究である。「じゃんけんゲームを作ろう」をテーマとして短いプログラムで動かすことを目指し、生徒の興味を惹き意欲的に取り組むとが出来る教材テーマとなっている。

生徒の作業内容は5段階に分けられている。

第1段階：Hello World!の文字列を出力させるプログラムをテキストエディタに打ち込ませる。

第2段階：用意したソースファイルを、乱数を用いて、発生した値に応じて「グー」「チョキ」「パー」の手を変えて表示するように書き変える。

第3段階：用意したソースファイルを、配列を用いた「じゃんけんゲーム」のプログラムに書き変える。

第4段階：第3段階のプログラムを、ボタンを押すことで関数が呼び出されるプログラムに書き変える。

第5段階：出す手それぞれのボタンの作成、じゃんけん結果の表示をするプログラムに書き変える。

と変数・配列・関数定義・関数呼び出しなどを理解させる説明までをしており、将来的にプログラミングを行う可能性を視野に入れた授業展開を想定している。中学生向けとして簡単すぎず、かつプログラミングの基礎的な記述を学習するための題材として提案している。

3.3.2 プログラミング言語を習得するための JavaScript を使用した WBT 教材の開発と実践

2007年に発表された山本芳人氏らの JavaScript を使用した実践である。目的は C 言語や Java の基本的な文法を習得するためとしている。大学1年生を対象としてアンケートを行い、持っているパソコンに大学で使うアプリケーションやコンパイラをインストールしていないことが分かった。山本氏が作成した JavaScript を使用した Web Based Training 教材を使用することで、自宅でもプログラミングを疑似体験できるようにしている。例えば C 言語の命令を入力し「実行する」ボタンをクリックすることで計算結果を表示する。コンパイル・実行を行わずに Web ブラウザのみで C 言語プログラムの実行結果を確認することが出来る。このように、環境整備が必要な言語の敷居を下げ学校外でもプログラミングの学習を可能にするために JavaScript を使用している例がある。

3.4 本研究との比較

3.4.1 Scratch を用いた研究の比較

紹介したプログラミング教育に関する実践研究においては、小学校低・中・高学年 6 年間でそれぞれの段階における題材を提案はされていなかった。パソコン操作能力やプログラミング的思考力等の、プログラミング教育で実施される内容に取り組み理解するための能力は、子どもの年齢やパソコンへの慣れである程度決まる。そのため、各低・中・高学年を担当する現場教師の一助となれるよう、初学者から経験者までが学ぶことが出来る内容に幅をもたせた複数の授業実践や授業題材を、学部生時代の研究から本研究までを含めて提案している。低学年では、主にツールを用いてパソコン操作能力の向上させることや、授業者が作ったプログラムで遊ぶことでプログラミングに対する基礎的な知識を身につける。中学年では、初歩的なプログラミング体験を通して逐次処理や繰り返し処理、条件分岐処理などのプログラムの流れを理解する。そして、操作能力・思考力が共に育ってきた高学年段階で、与えられた課題に対して自ら考えイメージしたことを表現するために、プログラムを組むことでプログラミング的思考力をさらに高めていくビジョンをもって題材の開発を行っていく。

3.4.2 JavaScript を用いた研究の比較

紹介したプログラミング教育に関する研究においては、小学生向けの JavaScript を用いた授業実践はなかった。JavaScript は Web ブラウザで操作を確認することができ、特別なソフトをインストールする必要がないため、学校に通常存在するパソコンの環境を変えずに実施することができる。また、テキストベースのプログラミングであるため Scratch よりも抽象的な構造理解が求められる。そのため、タイピング能力がある程度育ち、Scratch によってプログラミング基本的な思考方法が身についた段階の子どもに向けて題材を開発する方法としては適していると考えられる。紹介されていた JavaScript を用いた研究では、C 言語や Java の基本的文法を獲得することを目的としたり、配列・関数を扱っていたり等初めてコーディングを通してプログラミングを行う小学生にとっては難易度が高い。筆者の研究では、児童に予めプログラムを配って書き変えさせたり、フローチャートを用いて条件分岐の思考を説明したり、乱数の発生を抽選箱に例えてイメージさせるなどをして児童が理解をしやすいように工夫することで、テキストベースのプログラミング初体験の児童が多い中で授業実践を行った。あくまで最重要の目的はプログラミング的思考力の育成という点を意識しながら授業開発を行った。

第4章 Scratch を用いた「ひらがな練習ゲーム」

4.1 題材概要

本題材は小学1年生向けに開発した、マウス操作向上のためのひらがな練習をするツールである。マウスでドラッグして、画面上のひらがなをなぞって練習する。なぞる途中にお手本となる字からはみ出すことで変数がカウントされ、ズレが少なければ書き終えた後にアニメーション演出が表示されるようになっている。例えば、ひらがなの「あ」を練習した後にズレが少なければ、ありの絵が2パターンの動きを繰り返すアニメーションに設定してある。練習した字が使われている物や生き物が動く姿を見せることで、より上手に書こうという意欲を引き出すことに繋がる。また、書いた字の使用例を示す意味も含めている。

4.2 ステージ構成とゲームの流れ

ひらがな練習ゲームは4つのステージで構成されている(図4-1,(a)(b)(c)(d))。



図4-1 ひらがな練習ゲームステージ

ゲームの流れについては、初期画面の「もじをえらぶ」ボタンを押して文字選択画面へ移行する。練習したいひらがなを選んでクリックすると文字練習画面に移行する。ドラッグして文字を書き終えた後に画面右上にある「ていしゅつ」ボタンをクリックすると結果発表画面に移行する。結果発表画面で表示されるキャラクターは各文字 2 種類で、お手本からはみ出した回数によって変化する。例えば、「あ」の練習でははみ出した回数が 200 回以下であればありのアニメーションが映される。200 回を超えると猫が動きながらはみ出した回数を言って、結果発表が終わる。はたマークを押すことで再び初期画面に戻って、ゲームをやり直すまでが一連の流れである。

4.3 題材の詳細

4.3.1 ペン機能を持たせたねずみスプライト

ひらがな練習では、マウスをドラッグしてひらがなの文字の軌跡をなぞることで、文字を書くことが出来る。ペン機能を持たせたスプライトが必要であったことと、字を書く時にマウス位置の目安にすることを目的として作成した(図 4-2)。

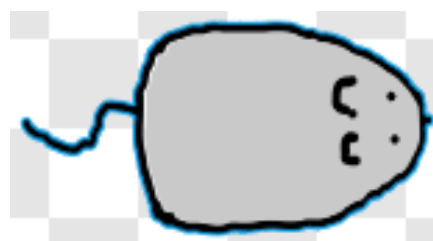


図 4-2 ペン機能を持ったねずみ

4.3.2 ねずみスプライトのプログラム

次に、ねずみスプライトに組み込んだプログラムを挙げる。

4.3.2.1 お手本からはみ出した回数を数えるカウンター

図 4-3 は書いた文字の軌跡がお手本からどれ程はみ出したのかを測るカウンターのブロック解である。ドラッグの最中に図 4-2 のねずみがお手本からはみ出すことでカウントされていく。

次ページで挙げる図 4-4 のプログラムで、ドラッグの最中にねずみがお手本の外側にある白色に触れることで「メッセージ 1」のやり取りが行われ、1 カウントされるようになっている。



図 4-3 お手本との比較カウンター

4.3.2.2 ねずみにペン機能を持たせるプログラム

「ずっと」繰り返しボタンの中身が主なペン機能を持たせるプログラムである。「マウスポインターを向ける」によってマウスが動く方向へねずみが向くようになる。「マウスポインターへ行く」によってマウスのポインターをねずみが追いかけるように動く。条件分岐「もし<マウスが押された>なら」によって、ドラッグ中に黒い線を引くことが出来るようになり、さらにその中の「もし<(白)色に触れた>なら」「メッセージ1を送って待つ」によって、白色に触れた判定を行い、図 4-4 のプログラムを動かしカウンターを乗せる。

「もし<提出ボタンに触れた>なら」によって、スプライト「ていしゅつ」ボタンにマウスポインターが重なったときに、「隠す」ブロックによってねずみスプライトが消えるようになっている。これは、文字を書き終えた後に結果発表画面へ移行しようと「ていしゅつ」ボタンを押そうとしても、ねずみスプライトが邪魔をして押すことが出来ないという問題が起きたため組み込んである。

また注意点として Scratch の仕様で、ドラッグして線を引く際、編集画面ではお手本のシルエット上に線が引けなくなっている。そのため、ゲームで遊ぶ時は画面右上のボタンによって全画面表示に切り替えなくてはならない。



図 4-4 ペン機能を持たせるプログラム

4.3.3 お手本文字の表示方法

図 4-5 は文字練習画面になった時に表示されるお手本となる文字のシルエットである。スプライト自体はコスチューム編集画面の「テキスト」で打ち込んだ文字である。ただし、そのスプライトをそのままお手本として使ってしまつと、ペン機能で書いた文字の線がお手本の文字に隠れてしまつて見えない問題が起こる(図 4-6)。そのため、図 4-7 のようにスプライトを「スタンプ」し画面にお手本と同じシルエットを映す。これはスプライトではないため、ペン機能によってそのままお手本の上に線を引いているように見せることができる。

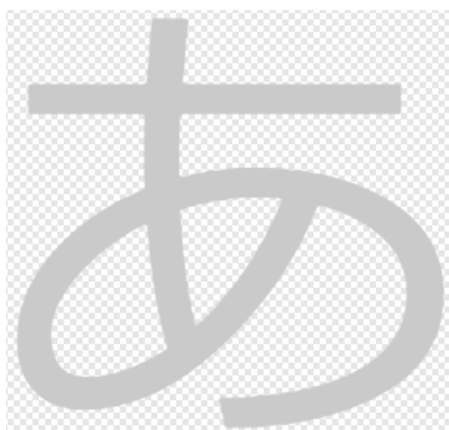


図 4-5 お手本文字スプライト

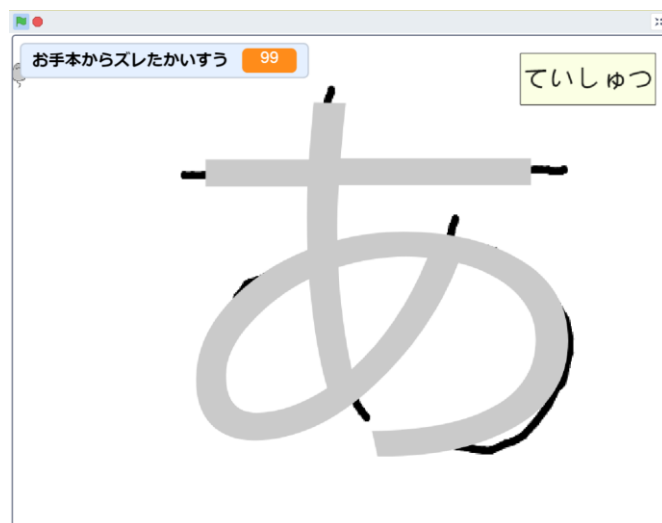


図 4-6 お手本文字のスプライトをそのまま使った場合



図 4-7 お手本文字のプログラム

4.3.4 結果発表に表示されるキャラクター

結果発表時に表示するキャラクターは各ステージ2種類である。1種類目は練習する文字が頭文字になる生き物や物のアニメーションである。ここでは例としてありのSpriteを示す(図4-8)。2種類目は各ステージ共通でデフォルトで用意されている猫のSpriteである(図4-9)。コスチュームは2つ用意しており、1秒置きにコスチュームを入れ替えることでアニメーションを表現している(図4-10)。



(a) 結果発表あり 1



(b) 結果発表あり 2

図4-8 お手本からはみ出した回数が200回以下だった場合に示されるキャラクター

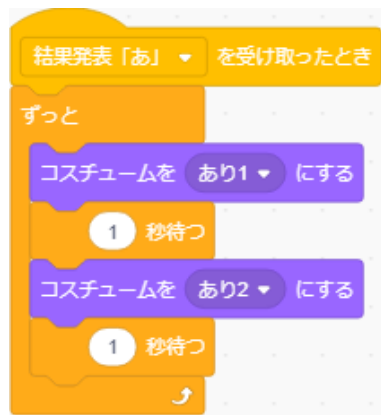


(a) 結果発表猫 1



(b) 結果発表猫 2

図4-9 お手本からはみ出した回数が200回を超えた場合に示されるキャラクター



(a) ありのコスチューム入れ替えプログラム



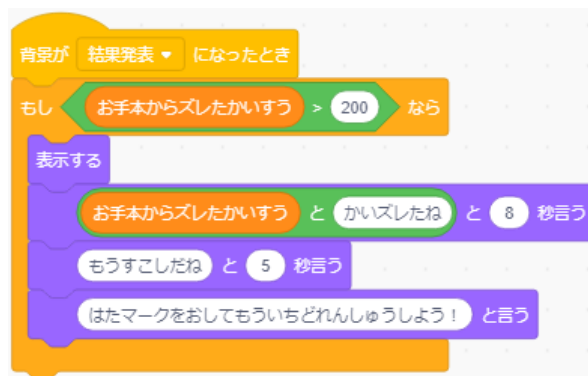
(b) 猫のコスチューム入れ替えプログラム

図 4-10 コスチューム入れ替えプログラム

また、各キャラクターにはお手本からはみ出た回数を判定して表示し、コメントするプログラムが備えられている(図 4-11)。



(a) ありのコメントプログラム



(b) ねこのコメントプログラム

図 4-11 キャラクターの条件付きコメントプログラム

4.3.5 提出ボタンについて

ひらがなの文字を書いた後結果発表画面に移行するために、文字練習画面右上には提出ボタンが表示してある。結果発表で扱われるキャラクターはひらがなごとに変わるため、それぞれの字に対して1つボタンを用意しておき、ボタンを押すとその文字を頭文字としたキャラクターのプログラムが動く。例えば図4-12はひらがな「あ」の練習画面に用意された提出ボタンである。「結果発表「あ」を送る」(図4-13)によってありのスプライトにメッセージが送られ、ありのプログラムが動く。文字ごとに提出ボタンのスプライトを用意し紐づけをする。文字ごとに提出ボタンを分けなければ、他に用意した他の文字に関連する生き物や物のスプライトも同時に動いて結果発表に表れてしまう。そのようにならないよう、各文字専用の提出ボタンと結果発表に移行するメッセージを設けている。

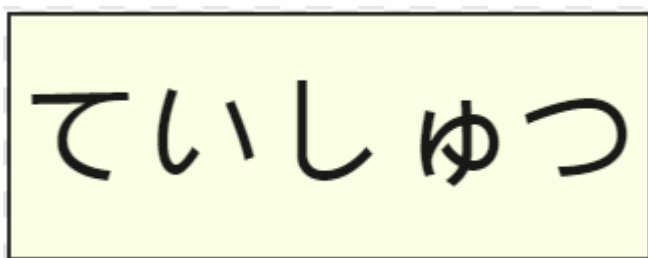


図 4-12 提出ボタンスプライト



図 4-13 結果発表へ移行するためのプログラム

4.3.6 文字選択画面の文字スプライト

文字選択画面には各文字のスプライトが置いてある(図4-14)。練習したい文字をクリックすることで、各文字専用の提出ボタンを表示させる(図4-15)。



図 4-14 文字選択「あ」スプライト

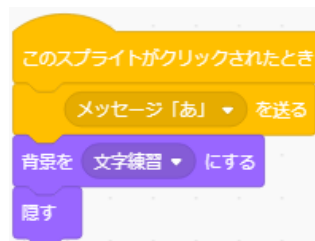


図 4-15 文字選択「あ」のプログラム

4.4 今後の課題

まず文字練習と結果発表における課題点がある。文字を最後まで書き上げなくても提出することができてしまうことである。その場合、お手本からはみ出した回数が少なくなり成功イラストが表示され、最後まで上手に書くことが出来た子どもとの区別がつかなくなり、子どものマウス操作能力の測定が正しく行うことが出来なくなってしまう。

次に変数カウンターについて、ねずみがお手本外側の白色の部分に触れている間ずっと数値は上昇していきすぎてしまい大きい数値になりやすい。そのため結果発表キャラクターの、お手本からはみ出した回数の判定条件の基準を 200 回に設定してある。理想としては、一度白色に触れた場合マウス・カウンターの動きを共に止め、もう一度ドラッグされたときに判定がなされるという造りで、白色に触れている間は連続ではなく一回一回白色に触れた回数を測りたい。

練習した出来上がりの文字についての課題点である。小学 1 年生対象ということでお手本の字の幅を大きくしてある。そのためお手本通りに書きカウンターの数値が低くても、角張った箇所が多い字であることもある。より直線やカーブを滑らかに書かせたいのであれば、お手本の字の幅を狭くしてはみ出さないように意識させることで改善されるだろう。ある程度マウス操作が上達すれば、様子を見てお手本の字の幅を狭めていくと良い。

4.3.5 でも述べたように、結果発表のキャラクターを練習文字ごとに用意したことで、提出ボタンを文字ごとに用意する必要があった。これによって、別の使い方に変更する際負担がかかってしまう。そのため、1つの提出ボタンと文字ごとのキャラクターのプログラムを結び付ける方法を考えるか、結果発表で使用するキャラクターを全文字で統一することで改善することが考えられる。

指導について、このゲームはマウス操作を練習することは出来るが、ただなぞるように伝えるだけでは早い上達は見込めない。低学年の子どもにおいては初めてマウスを握る子どもも多いため、マウスの使い方から別に指導する必要がある。右クリック左クリックの区別やドラッグの仕方等を、書画カメラを用いて手元を移したり実際に手を握ってマウスを扱う感覚を伝えてあげたり、別に教具を用意して示したりして指導することも試みると良いだろう。また、年齢の低い子どもはドラッグ中にクリックしている指を離してしまう子が多いように思う。長いドラッグにはある程度の腕・手先の筋力が必要であるため、慣れるまではお手本をなぞる時にも一度で一本の線を書くのではなく、何度かに分けて書いていく方法も試させたい。字の特徴を覚えさせるには一画書き終えるまでは止まらない方が良いが、子どものまずは子どもがマウスの使い方になれることを、ここでは優先させたい。

4.5 本題材のまとめ

本題材は小学 1 年生を対象として開発したものであるが、他学年での漢字や描画練習への応用ももちろん可能である。担当学年の子どもの実態に合った、字や絵の練習に使いマウス操作に慣れていくことを目指すために使うことが出来るだろう。

1 年生で使う場合には、50 音順に拘らずに書く練習をしていくと良いだろう。例えば直線のみで出来ている字から始めて、徐々にカーブがある字の練習に移行していくと良い。

本題材を使う活動の目的は「マウス操作の向上」であり、お手本となる字を表示しペンでなぞることが出来る仕様になっているため方法は示すことは出来た。他に考慮しなくてはならないことは、題材をどの時間にどれ程使うか、どのように指導するかによって上達スピードは変わるだろう。担当する子ども達はどれ程のことが出来るのかを把握し、相応の難易度にゲームの仕様や扱う文字・絵を変更することで効果が得られると考える。

第5章 Scratch を用いた「魚釣りゲームプログラミング」

5.1 題材の概要

本題材は制限時間内に釣り針を制御するボタンを、タイミングを計って押すことで釣り針が下降し、魚と釣り針を接触させ釣り上げる内容のゲームである。児童にとってのプログラミングの基本の学びやすさに重きをおいて考えたもので、児童が扱う範囲のプログラムを簡潔なものに留めてある。授業では、子どもが描いた魚が画面上で静止している状態で授業を始める。2つの動作を加えるようにプログラムを加えるものを課題として与える。1つ目は魚が画面上を横に往来する。そして2つ目は、落ちてきた針に魚のスク립トが触れると座標が釣り針と水面まで上昇していくプログラムに作り加えていくよう指導する。

5.2 題材開発の経緯

本題材は、プログラミングをこれから学ぼうとしている低学年の児童向けに開発したものである。開発理由は主に2つある。1つ目は題材である「魚釣りゲーム」の间歇性である。初めてプログラミング言語に触れ、手探りで試行錯誤しながら動作に慣れていく段階では、シンプルな構造設計がされたプログラムで考えることが望ましいと考える。魚釣りゲームは、子どもにとってもイメージがしやすく、糸を下して接触した場合釣り上げるといふ動きは、逐次処理と条件分岐、繰り返し処理の要素を絡めて考えやすい。これが、1つ目の理由である。

2つ目は、これまでの研究との関連性に注目をしたことである。2017年度に卒業した同研究室の吉原氏の研究で「すいぞくかんをつくろう！」という、1年生を対象としたScratchで魚を描かせ授業者のパソコンに集め、授業者が作ったプログラムで一斉に動かす授業実践があった。この単元で造形や鑑賞、マウス操作能力の向上等を目的として行った実践であった。この授業の流れで児童が描いた魚を、自らが作成したプログラムによって動かすことを経験することで、よりプログラミングの意味を実感し、創作意欲が向上する効果に期待が出来ると考えたためである。

以上の理由より、本題材の開発に着手した。

5.3 指導の手順

手順として、授業者が予め魚の初期位置を決めるプログラム(図5-1)を作成しておき、それに子どもが、魚が画面を左右に往来するようにプログラムを加えていく。授業では、待機画面から「はじめる」ボタンを押したゲーム画面(図5-2)を見ながら指導していく。まず繰り返しブロックの中に、魚の座標を進めるための「動かす」と画面端に到達した時に向きを変えるための「跳ね返る」ブロックを入れる(図5-3)。次に、釣り針に触れたときの動きをプログラムする。条件分岐ブロックの中に、「y座標を変える」と判定を水面の

座標にした「繰り返し」ブロックの中に入れ(図 5-4)、図 5-3 で作成したブロックの中に入れる。このゲームは、釣れた魚の数によってクリア画面でのコメントが変化するように設計してあるため、変数機能の「何匹を1ずつ変える」ブロックを加える。最後に「隠す」ブロックを追加し、釣り上げた魚を画面から消去する。以上が、子どもが授業者の指導を基に行う課題の内容である。



図 5-1 初期ブロック

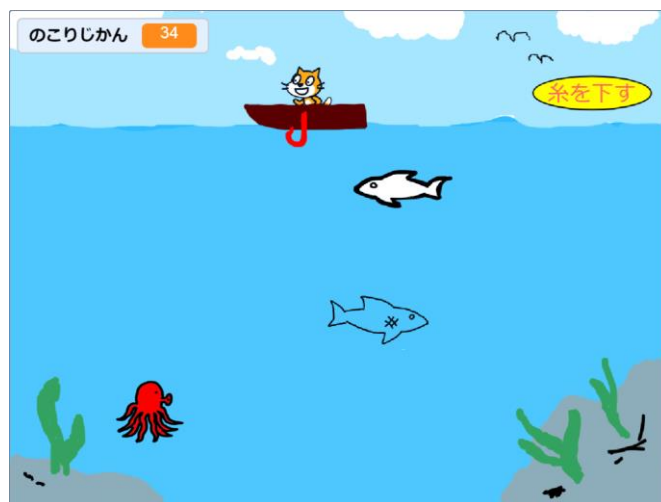


図 5-2 魚釣りゲーム画面

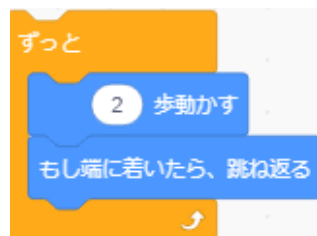


図 5-3 画面を左右に往来するためのプログラム



図 5-4 釣り上げるためのプログラム

早く課題を終えてしまった子どもについては、音機能をブロック「魚が釣れた時に音を鳴らす」動作を追加させることで、進捗状況の調整を行っていく。

最終的に児童が完成させるプログラム(図 5-5)を示す。

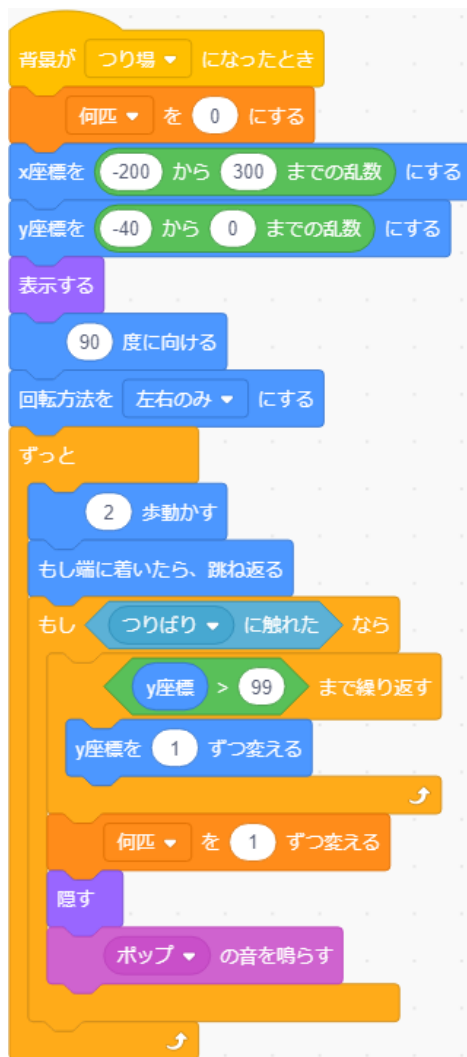


図 5-5 完成目標とするプログラム

5.4 題材の詳細

5.4.1 待機画面

魚釣りゲームは待機画面(図 5-6)よりスタートする。画面真ん中の「はじめる」ボタンをクリックすると、すぐにゲーム画面(図 5-2)に移行する。



図 5-6 魚釣りゲーム待機画面

5.4.2 用意したスプライトとプログラムの説明

まず、ゲーム画面を泳ぐ魚(海の生き物)は 3 匹用意した(図 5-7)。魚 1 と魚 2 については授業者が全児童のファイルに共通で配布したものである。魚 3 については、本来児童がそれぞれ描いた魚のスプライトを用意するが、今回はサンプルで作成したものを図で示す。



図 5-7 魚スプライト

扱うスプライトを描く時の注意点として、スプライトを描く枠の中心(図 5-8 内赤丸)にスプライトを置くことである。そのようにしなければ、ゲーム画面でも魚の中心がズレてしまい見た目上釣り針と接触していたとしても、ヒット判定とならない場合がある。児童にスプライトを別単元で描かせる際も、スプライト編集枠の中心に合わせて描いているかを授業者は伝え、確認しておく必要がある。

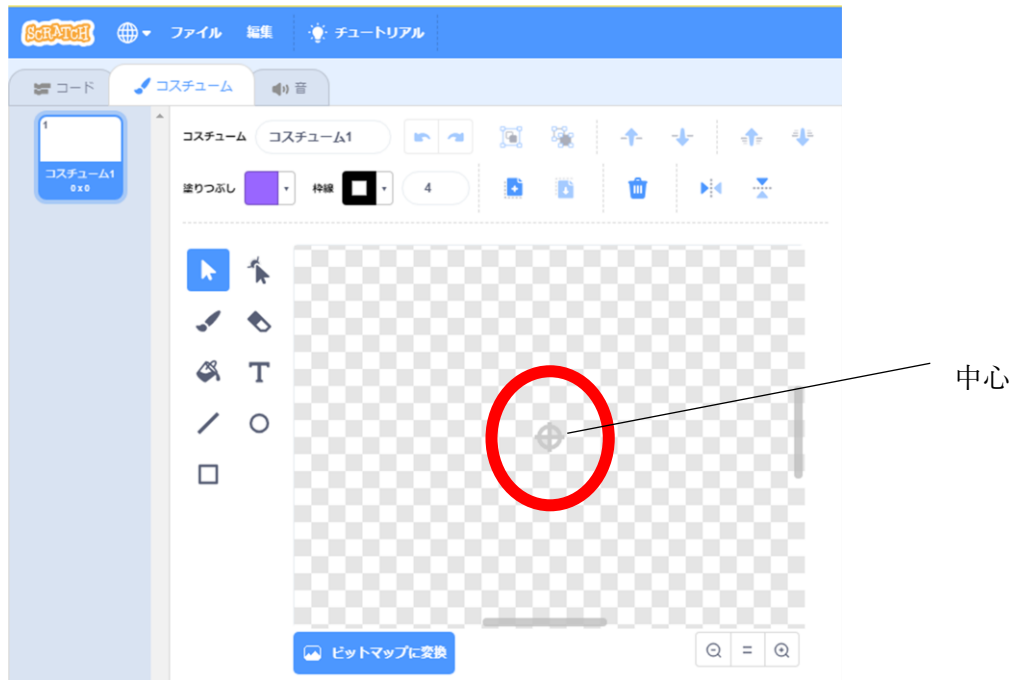


図 5-8 スプライト編集枠中心

各魚のスプライトに備えられているプログラムは図 5-1 のものである。「何匹を 0 にする」ブロックによって、釣れた魚の数をリセットする。次に、「x 座標を -200 から 300 までの乱数にする」「y 座標を 10 から 60 までの乱数にする」ブロックによって、ゲーム画面を開く度に画面上の x、y(縦,横)軸よりランダムに選び配置する。また、魚のスプライトごとに「y 座標を○から○までの乱数にする」の数字を異なるものにしてはいる。これは、3 匹の魚が重なることを防ぐために設定してあり、余裕をもって y 座標の範囲設定を指定することで、画面にバランスよく魚を散りばめることが出来る。一番下に繋げている「回転方法を左右のみにする」ブロックを入れてある理由は、魚が端に到達し跳ね返る際にスプライトが画面上を向いてしまい、y 軸方向にスプライトが動いて行ってしまふことを防ぐためである。



図 5-1 初期位置に配置するプログラム

魚釣りをしている猫のSpriteには、主にタイマー機能のブロックを備えてある。



図 5-9 つりねこのSprite

変数機能によって、「のこりじかん」ブロックを作成し、1 ずつ時間を減らすように設定する。初期設定では時間を 60 秒に設定してあるが、児童の遊ぶ様子を見て「のこりじかんを〇にする」ブロック内の数字を変更出来る。結果発表画面に移るための条件として、①「設定時間が経過した時」②「全ての魚を釣り上げた時」のいずれかを満たした際に設定した。

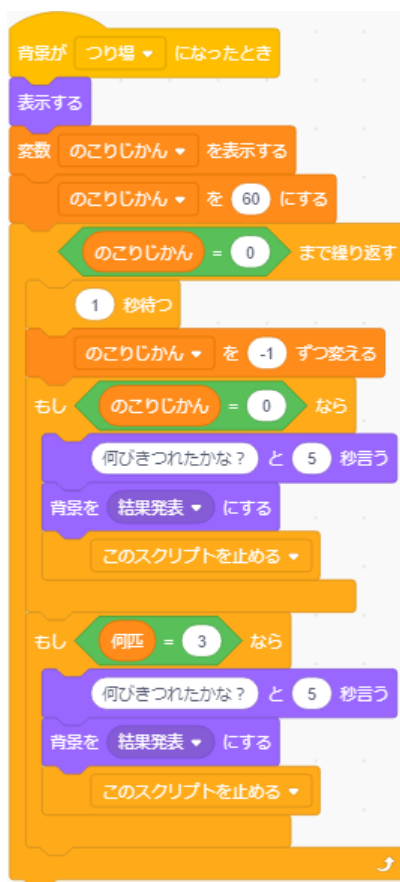


図 5-10 タイマー機能をもたせたつりねこプログラム

次に「糸を下す」ボタンについてのプログラムを説明する。図 5-11 を押すことで、メッセージ「糸を下す」を送り、別に用意したスプライト「つりばり」へと動きを繋げる。



図 5-11 糸を下すボタンスプライト

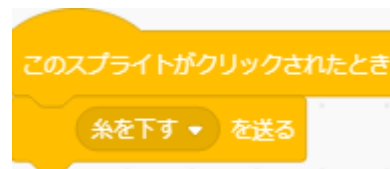


図 5-12 糸を下すメッセージ

上述した「糸を下す」メッセージを、つりばり(図 5-13)が受け取り、画面を下降する。下降時の条件と実行処理は、①「y座標が-170より小さくなった時、つりばりを引き上げる」②「スプライト魚1,2,3のいずれかに触れたとき引き上げる」③「①,②でなければ、下降し続ける」の3つ(図 5-14)に分岐する。

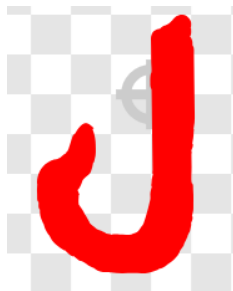


図 5-13 つりばりスプライト

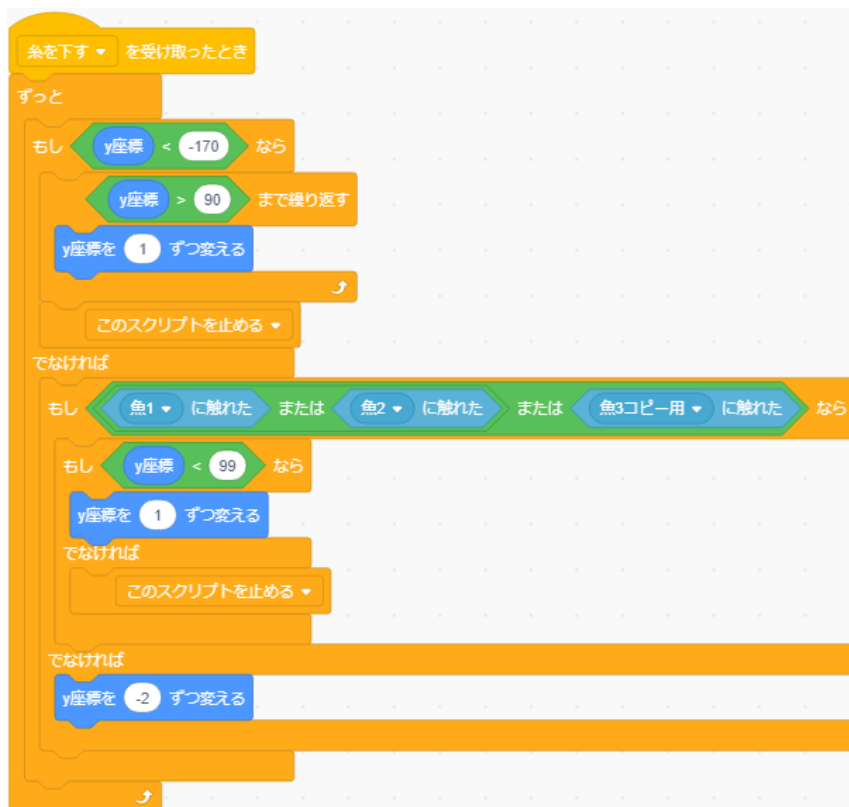


図 5-14 つりばりが下降するプログラム

5.4.3 結果発表画面

ゲームが終了すると、児童がゲームで釣り上げた魚の数とコメントが表示される画面に移る(図 5-15)。図は 3 匹釣れた時の画面を映している。

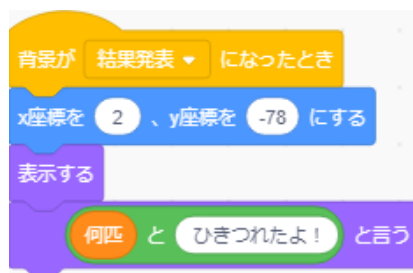


図 5-15 結果発表画面

画面中央の「すごいね!」の sprites と、猫の「〇ひきつれたよ!」が釣れた魚の数によって異なるようになっている。それぞれのコメントは、条件分岐ブロックによって分けられている。例えば、釣れた数が 0 匹であれば「がんばろうね!」、1 匹であれば「もうすこしだよ!」、2 匹であれば「やったね!」、3 匹すべて釣れたなら「すごいね!」と表示するように、それぞれプログラムしてある。



(a) Cat スプライト



(b) 結果を伝えるプログラム

図 5-16 Cat に備えられたブロック

がんばろうね！

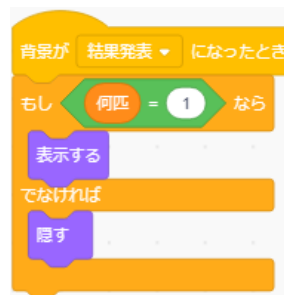
(a) 結果コメントスプライト 1



(b) 0匹の場合のプログラム

もうすこしだよ！

(c) 結果コメントスプライト 2



(d) 1匹の場合のプログラム

やったね！

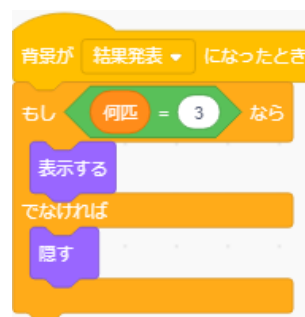
(e) 結果コメントスプライト 3



(f) 2匹の場合のプログラム

すごいね！

(g) 結果コメントスプライト 4



(h) 3匹の場合のプログラム

図 5-17 釣れた魚の数によって変わる結果のコメントとそのプログラム

5.5 題材の工夫した点

実際の教育現場の子どもは、家庭のパソコンの有無、学校外での Scratch 等を通したプログラミング経験の有無によって、マウス・キーボード操作への慣れやプログラミング、言語の特徴への理解度は異なる現状を、これまでの授業実践で見えてきた。そのため、「魚を動かす」「釣り針に触れたときの動き」のステップに分けて教えることが出来るようにし、子どもごとのペースで課題に取り組ませることが出来るようになっていく。また、一度に複数の処理内容について考えさせることで、プログラムの構造が理解しにくくなってしまいうため、プログラミングを学び始める子どもにとっても、スモールステップに分けることは有効であると考へた。

本題材は、考へ組んだプログラムによってゲームで遊ぶことが出来るため、身近にあるゲームや自動で動いている物が人の手によって動いているのだと伝えるには効果的であると考へる。授業内で児童が完成目標まで仕上げ遊んだ後に、ゲーム的要素を追加していきゲームらしさを増すよう改善させていく流れにも繋げられるだろう。例へば、今後このカウンター機能を使って、条件分岐を応用した課題に取り組ませることが出来る。変数の機能を用いてカウンターを作ることで、ポイントを付与し点を競い合うゲームも展開出来るだろう。

5.6 授業で工夫する点

より少ない時間数で課題に取り組ませるために、子どもが自分で描いた魚をスクリプトとして保存したファイルを予め作り配布しておく。また、多くのスクリプトの動きについて考へさせると、混乱してしまうため魚の動きに限定して考へさせることを伝えておく。そしてプログラムが長く複雑になりそうであれば、プログラムについて説明するときに、その都度その部分のみを示すことによっても考へたいことに集中させる。

黒板にゲーム画面と同じ背景を描いた模造紙と厚紙に貼った魚のスプライトを準備しておき、理想とする動きを演じて考へていく。そうすることで、魚の動きをより具体的にイメージし、クラスの子どもが共有出来るようになる。例へば、魚を模造紙の端に移動させていき「このままだと端にぶつかってしまいますね。ぶつかったらどのように動けば良いかな」「魚が反対を向いて進めばいいんじゃないかな」のようなやりとりを行うことで、一方的な指導でなく子どもの気づきを拾って、反応することが出来る。このように、パソコンに向かいながらではなく、具体物を使って考へ解決方法を知った上で、プログラミングを行うことで自分がすべきことのイメージをもつことが出来るだろう。また、手順を追って考へていくことで、どのような動作を繰り返し処理として行えば良いのか、どのような条件を用いて分岐処理を行えば良いのかを理解することが出来るため、1単位の動きを繰り返しや条件分岐ブロックの中に入れることの意味が理解しやすくなる。条件分岐については、フローチャートを用意して説明すると視覚的に認識できて状況が整理しやすくなるため、状況に応じて指導方法を選択すると良いだろう。

5.7 題材のまとめと課題

本題材はプログラミングを始めて間もない低学年の子ども向きに考えたもので、プログラミングの要素(逐次処理・繰り返し処理・条件分岐処理・実行)を簡単に学ぶことが出来る。低学年段階ではパソコン操作が拙い子どもが多いことから、子ども達がマウスを使って絵を描き、それを授業者作ったプログラムで動かす展開の授業がこれまでに実践されてきた。本題材を通して、自ら描いたスクリプトを動かすようプログラムする体験をすることは、プログラミングの意味と目的を実感する足がかりとなるだろう。また、既習単位との繋がりを意識して題材設定したことで、学習に対するモチベーションを保ち向上させる効果も期待できる。

魚釣りゲームは簡潔な設定ではあるが、少しプログラムの内容を変更することでゲームの難易度の調整やエフェクトの追加が可能である。例えば、魚の動く速さや大きさだけでも、釣ることが初期設定よりも難しくすることが出来る。そのため、興味をもち授業外の時間でも、自ら触る子どもが増えることもあるだろう。子どもの思い描いた動きに、間違いながらも試行錯誤して近づけていくことで少しずつプログラミングを学んでいって欲しい。

今後は、子どもが成長し誘導なしで課題に挑戦していくためには、より難易度の幅をもった課題を設定する必要がある。例えば、「魚の種類ごとにポイントを付与する」「得たポイントに応じてクリア画面を変える」というようにレベルを少し上げた課題を設け、それまでに培ってきたプログラミング的思考力を使って考えられるような題材に改善していきたい。

6章 Scratch を用いた「ローマ字タイピング練習ゲーム」

6.1 題材概要

小学校中学年向きに、タイピング練習ゲームの制作を行った。本題材は Scratch のリスト機能を利用して、問題の作成と出題を行うものである。キャラクターが話す言葉を解答フォームに入力する。入力した文字列と、予め作っておいた正答リストの文字列の照合を行い、正答の判定をする。時間制限を設け、正答した問題数に応じてクリア画面を変えるようにプログラムしてあるため、画面を見ればどの程度の正答率か一目で判断することが出来る。

題材の作成時はキーボードのアルファベットの位置を覚えることと関連して「ローマ字のタイピング練習」をテーマに作成を行った。しかし、Scratch のリスト機能はテキストエディタの文字列を一行ずつ読み取ってリストを作成するため、基となるメモ帳を書き換えることで漢字の読み書きの練習にも応用が可能である。よって、使用時期の学習範囲に合わせてテーマを変更すれば、より広い範囲でタイピング練習に使うことが出来るだろう。

6.2 開発の経緯

筆者はこれまでに授業者・TA として小学校で授業実践に参加する機会があったが、その時に児童の実態としてキーボード操作の能力に課題があると感じていた。年齢の低さからパソコンの操作に慣れていないことは当然であるが、この先パソコンを用いてプログラミングを通した授業を経験していくことを考えると、キーボードの操作に慣れていかななくてはならない。そこで、小学生の子どもでも楽しくタイピングを学んでいけるように、ゲーム式のタイピング練習ツールを開発した。

6.3 題材の詳細

6.3.1 スタート画面

初期画面(図 6-1)の黄色いスタートボタンを押すことで、ステージ選択画面に移るようになっている。

スタートボタンは、初期画面以外の画面では隠れるようになっている。また、初期画面に戻った場合に表示されるようになっている。

タイピングれんしゅう
ゲーム



ステージをえらぶ

図 6-1 スタート画面

6.3.2 ステージ選択画面

ローマ字タイピング練習を想定して作成した際には、ステージを6つ用意した。それぞれのステージで扱った問題は後述する。ステージ1では、「A,B,C,...」等のアルファベット26文字を打ち込む問題になっている。ステージ2では「てんぷら、まんが,...」等「そのまま英語でも使えるローマ字」の日本語を出題し、ローマ字に直して打ち込問題。ステージ3では「バナナ,オレンジ,...」等のように、「アクセントの位置が異なる語」を出題した。ステージ4では「ボール,ゲーム,...」等のように「ローマ字と英語では綴りや音が異なる語」を出題。ステージ5ではステージ2~4の問題を混合し10問出題した。最後にステージ6では「pasokon, zyugyou, ...」のように学校生活に関わる語のローマ字を出題した。

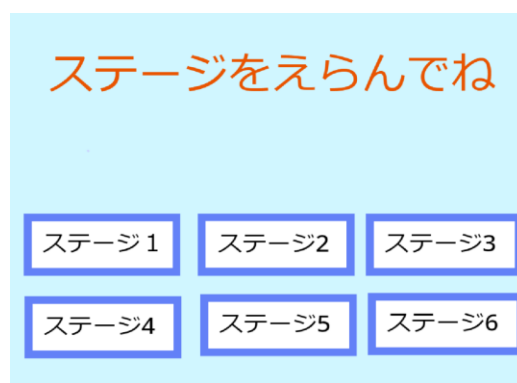


図 6-2 ステージ選択画面

6.3.3 ゲーム画面

図 6-2 でステージ選択を行った後、すぐにゲームが始まる。図 6-3 はステージ 1 の画面である。

画面右の魔女のスク립トが、問題リストに読み込まれた文字列を解答する度に出題する。画面下の入力フォームに解答し、Enter キーを押すことで正誤判定を行うようになっている。正解した場合は、画面右上の「せいかいすう」カウンターにポイントが加算される。



図 6-3 ゲーム画面

6.3.4 クリア画面

ゲームが終了し、結果発表を行う画面である。制限時間内に正答した問題数に応じて、クリア画面が変わるようになっている。

条件分岐処理ブロックを用いて、変数ブロックを作ってカウントした正答した問題数を条件として表示する画面を決定する。以下に例として、ステージ1とステージ2における正答数に応じてクリア画面を変えるプログラムを載せる(図6-4, ステージ1は左、ステージ2は右)。クリア画面は全5枚用意してあり、ステージ1の場合全26問を約5問刻みで画面が変わるようになっている。他ステージは問題数が異なるため、バランスの良い正答数の幅でクリア画面決定を行うように条件の数値を変更する必要がある。



図 6-4 クリア画面表示プログラム



(a) クリア画面 1



(b) クリア画面 2



(c) クリア画面 3



(d) クリア画面 4



(e) クリア画面 5

図 6-5 クリア画面 5 種

6.3.5 リスト機能を用いた問題の作成

Scratch のリスト機能を用いた、問題の作成の仕方を説明する。

リスト機能は Scratch のコード編集画面左の機能覧内の「変数」をクリックすることで選択することが出来る。そして、「リストを作る」をクリックしリスト名を決定する。次にゲーム画面に空のリストが表示される(図 6-6)。空リストにカーソルを合わせ右クリックすると「import」を選択出来るようになる(図 6-7)。

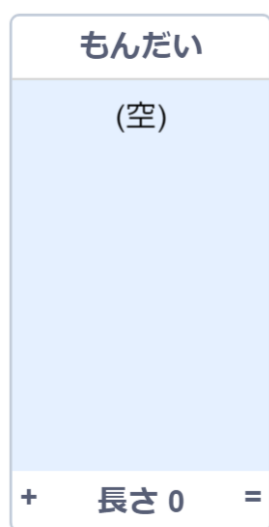


図 6-6 空リスト

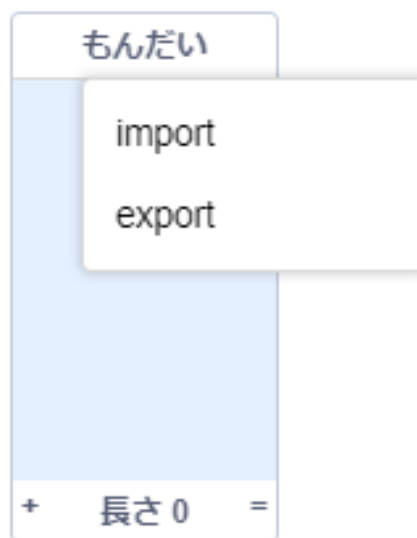


図 6-7 文字列の読み込み

次にエクスプローラーが開かれ、リストに読み込むファイルが選択できる(図 5-8)。

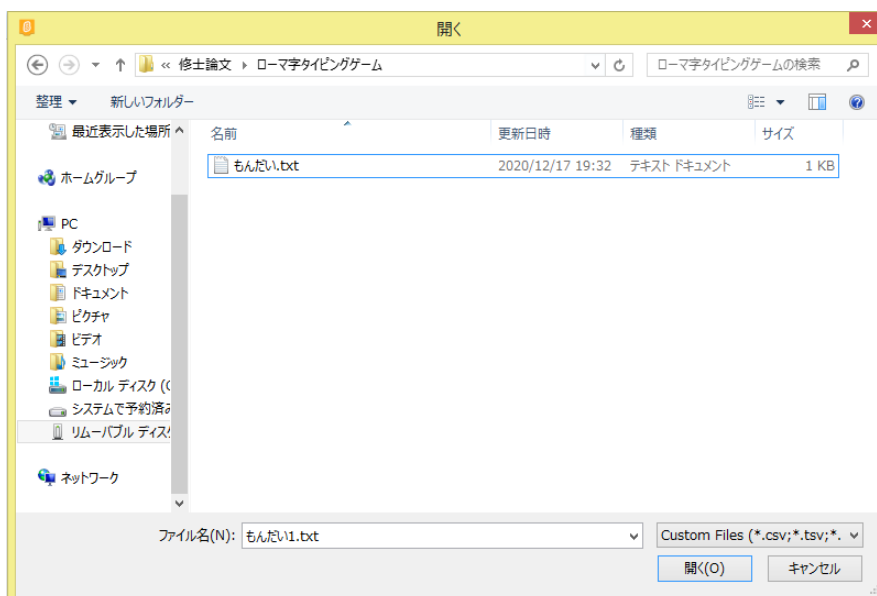


図 6-8 読み込むファイルの選択

ファイルを選択すると、空だったリストの中に、選択したファイルに配列されていた文字列が一行ごと読み込まれる。図 6-10 はステージ 1 と同じ問題をメモ帳にてファイル(図 6-9)を作成しておき、読み込んだものである。



図 6-9 文字列ファイル



図 6-10 リスト完成

6.3.6 出題プログラム

次に入力フォームに打ち込んで送信する度に、次の問題が出題されるためのプログラムを示す。魔女のsprayにプログラムしてある(図 6-11)。

一番上の変数ブロックを用意し、解答するごとに 1 加算されるように繰り返しブロックの中に入れてある。この 1 ずつ変化する変数が、問題リストに入っている 1 行ずつに対応している。例えば、図 6-10 では変数が「2」の場合 2 行目の「B」が出題される。



図 6-11 出題プログラム

繰り返しブロック内の「～と聞いて待つ」ブロックによって出題する語を尋ねる。このブロックによって尋ねることで、入力フォームが自動的に表示される(図 6-12)。また入力した答えも認識され、次に入力フォームに打ち込むまで「答え」として保持される。次項目で説明するがこの答えを、正答リストと比較して正誤判定を行う。



(a) 答えを尋ねるブロック



(b) 入力フォーム

図 6-12 出題と解答の仕組み

6.3.7 正誤判定をするためのプログラム

入力フォームに打ち込んだ語が正解であるかを判定するプログラムを示す(図 6-13)。

条件分岐ブロックの条件として、「答え(入力フォームに打ち込んだ語)=正答リストにある問題ごとの正答」を設定しておく。もし正解であった場合は正解数を加算し、不正解であった場合は正解数を変えない。



図 6-13 正誤判定プログラム

6.3.8 メッセージブロックによるプログラム間の連動

本題材では、スプライトやユーザーの1つひとつのアクションに繋がりがあある。例えば、①魔女のスプライトが問題を出す

②ユーザーが入力した答えを猫が答える

③正誤判定を行う

というように、1つの動作に次の動作が反応してゲームが進行していく。この仕組みを設定するために、「メッセージ」ブロック(図 6-14)を利用した。



図 6-14 「メッセージ」ブロック

使用例として、上記の①～③のプログラムでの使用方法を以下に示す。



(a) 魔女のスプライトが問題を出す



(b) ユーザーが入力した答えを猫が答える



(c) 正誤判定を行う

図 6-15 「メッセージ」ブロック使用例

6.4 問題に使用した語と正答のファイル

ゲームで使用した語を配列した①問題と②正答のファイルを、6 ステージ分示す。

ステージ 1,2 は出題した語と同じ文字列を打ち込む問題であるため、①問題と②正答のファイルの中身は同じである。

また、「てんぷら」のように「tempura」「tenpura」と訓令式とヘボン式 2 通りの表記の仕方がある語については、問題番号に合わせて改行をして答えを打ち込んだファイルを 2 つ用意してある。

(1) ステージ 1(アルファベット 26 字)

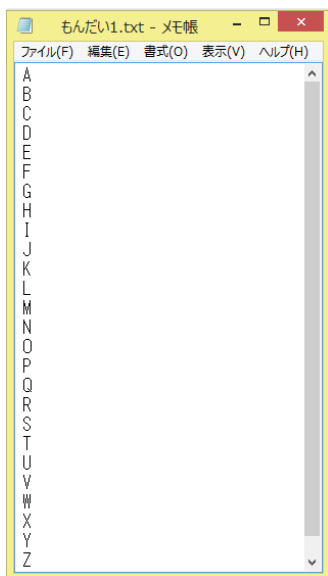


図 6-16 ステージ 1 問題・正答ファイルの中身

(2) ステージ 2(そのまま英語でも使えるローマ字)

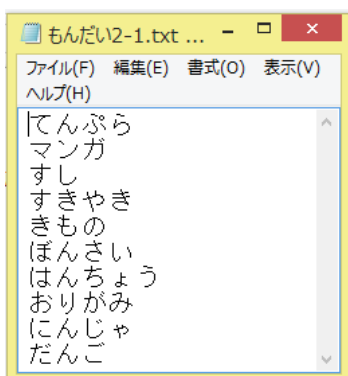


図 6-17 ステージ 2 問題ファイルの中身

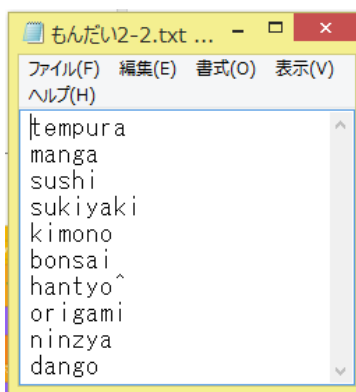


図 6-18 ステージ 2 正答ファイルの中身

(3) ステージ 3(アクセントの位置が異なる語)

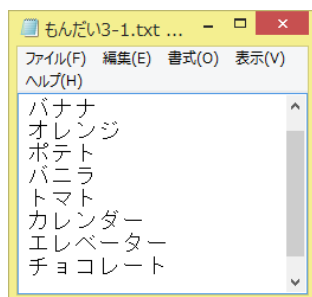


図 6-19 ステージ 3 問題ファイルの中身

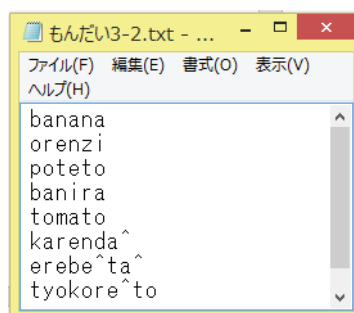
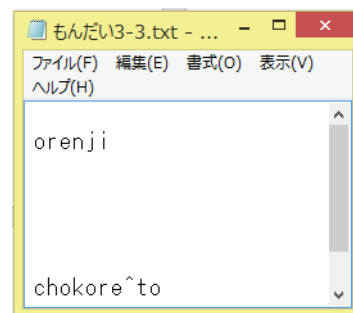


図 6-20 ステージ 3 正答ファイルの中身



(4) ステージ 4(ローマ字と英語では綴りや音が異なる語)

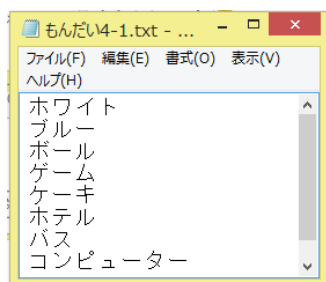


図 6-21 ステージ 4 問題ファイルの中身

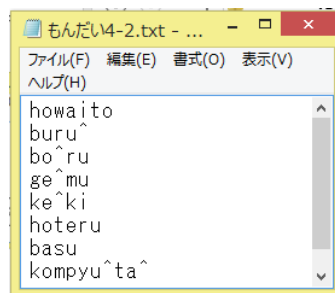
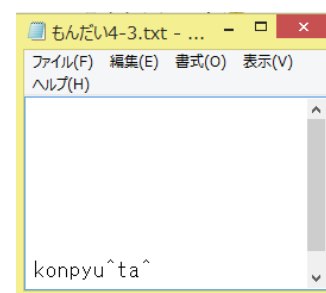


図 6-22 ステージ 4 正答ファイルの中身



(5) ステージ 5(ステージ 2~4 の混合問題)

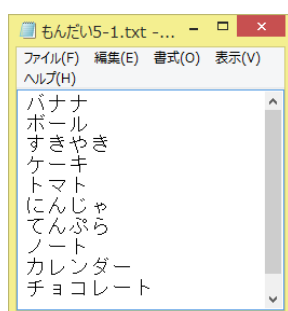


図 6-23 ステージ 5 問題ファイルの中身

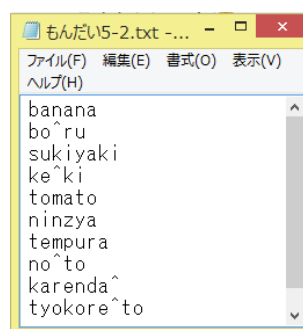
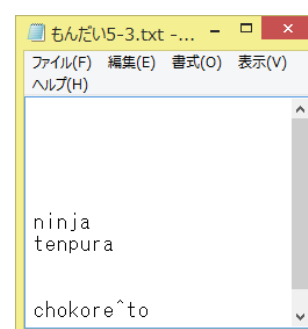


図 6-24 ステージ 5 正答ファイルの中身



(6) ステージ 6(学校生活に関わる語)

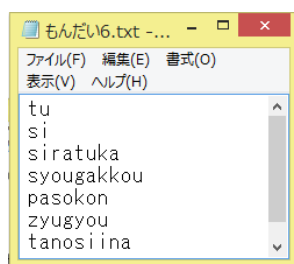


図 6-25 ステージ 6 問題・正答ファイルの中身

6.5 本題材の今後の課題

本題材はテキストエディタに配列した文字列を読み取ることで、問題・正答のリストを作成していた。この仕組みによって、覚えたローマ字や英単語のアウトプットの練習や漢字の読みや正誤問題への応用が可能である。用途の変更も、メモ帳を用意してリストとして読み込み、プログラムの問題数を変えれば適用できるため容易であると言える。しかし、6.4の図 6-18 等で示されているように表記の仕方が複数ある場合、正答ファイルも複数用意しなくてはならない。また、改行の数が適切でない時は正誤判定が正しく行われないう。よって、新たに正答ファイル 1 つで複数の表記の仕方に対応できる仕組みを考える余地が見いだされた。

次に問題の出題について、問題ファイルに配列した順番で出題されるようになっていた。しかし、繰り返し使って文字の綴りを覚えることを目的とするのであれば、問題ファイルの中から順番ランダムにして出題する仕組みの方が、繰り返していく内に出題順を覚えて機械的に答えるようになってしまうことがなくなるため、より記憶のために効果が得られる。そのため、ランダムに問題ファイルから出題する仕組みも加えられると良いだろう。

最後に児童が間違えた問題を記録する仕組みについても、今後検討したい。もし児童が苦手とする問題の傾向が分かれば、指導者としても個別に指導したり、繰り返し出題したりしていくことで克服していく手を打つことが出来るためである。子どもにとっては楽しく学べるツール、指導者としては子どもの学習進捗を確認する役割をもたせたツールとして今後も開発を行っていきたい。

(2) 画面の説明

図 7-2 から図 7-4 は迷路ゲームプログラムの画面説明や操作説明をするために用意したパワーポイントの画面である。

図 7-2 のスライドを用いて、迷路の画面にあるスクリプトやボタンの説明を行った。1つひとつの文章の内容と赤枠を対応させて説明しており、画面右の図で説明している箇所を赤い枠で囲う

ようにしてどこの部分について説明しているのかが分かりやすいように工夫した。

まず、画面右上にある旗をクリックすると新しく迷路が作られること。ここで、プログラムの間違いに気付いたときなどに間違っって旗を押してしまうと、途中の迷路が消えて新しい迷路に書き換わってしまうため、注意を促した。



次に、左上が猫のスタート位置になっており、ブロックを繋ぎ合わせて迷路の右下にあるゴールまでたどり着かせることがクリア条件であることを説明した。

最後に「Start」ボタンをクリックすると猫がさいしょの位置から動き、ブロックを繋いでいる途中に、どのような動きをするのかを確認するために使うためのボタンであることを説明した。

(3) 猫の動かし方の説明

図 7-3 を見せながら、「右むけ右」「左むけ左」「前に進む」の3種類のブロックを繋ぎ合わせて、猫を動かしていくことを説明した。

1. めいろのせつめい

- ・  マークをクリックするとめいろが新しく作られる
- ・  をがめんの右下にあるゴールまでうごかそう!!
- ・ **Start** クリックすると猫がさいしょのいちからうごく

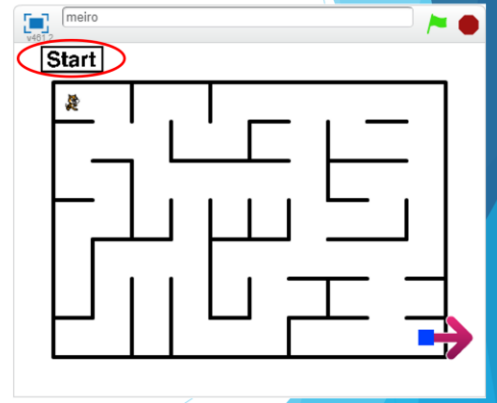




図 7-2 迷路の説明画面

2. ねこのうごかし方

- ▶ ・ 3つのブロックを組み合わせて
- ▶ ねこをうごかしていく

 ねこが『右むけ右』をするブロック

 ねこが『左むけ左』をするブロック

 ねこが『前にすすむ』ブロック



図 7-3 猫の動かし方の説明画面

(4) 猫の向きの説明

図 7-4 を見せて、「右むけ右」「左むけ左」のブロックを繋ぎ合わせて向きを変える時に、猫がどの向きを向いているのかを分かるように説明した。猫の体が向いている方向へ進むようになっており、「前に進む」をした時にどの向きに進むのかアニメーションを使って見せた。



図 7-24 猫の向きの説明画面

(5) 練習問題の設定

図 7-5 と同じ迷路図をホワイトボードに書き、紙に印刷した猫とブロックを使って練習問題を児童と考え、ホワイトボード動きの確認をした。ブロックを一枚貼るごとに猫も向きを変えたり進めたりして貼りなおして、ゴールまで進めた。簡単な迷路であるが、多くの児童がどのブロックを繋げれば良いかを声に出して答えていたり、頷いたりする姿を見て、練習問題を解く時点での理解度は十分に高いと感じた。



図 7-5 練習問題の図

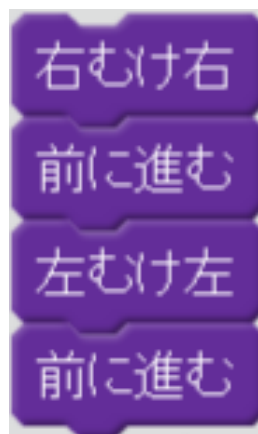


図 7-6 練習問題の答え

(6) 学習進捗別のステージ設定

用意した迷路の大きさは2つあり、図7-7は小さい方の迷路である(以下 meiro1)。練習問題を解いた後に、meiro1 をゴールしてクリアするためのプログラム作成に取り組ませた。取り組ませてから2分程で1人目がクリアした。その後5分以内には、9割強の児童が meiro1 をクリアしている状態であった。15分間は meiro1 を新しい迷路にして取り組ませる予定であったが、TA から児童の取り組みの様子を尋ねながら机間巡視をし

て家でプログラミングをやっている子どもや、何度もクリアをして飽き始めている児童についてはTA に大きい方の迷路(以下 meiro2)へ進ませるように指示を出した。

図7-8は meiro2 の画面である。迷路を児童に解かせ始めてから15分経った時点で、任意で meiro2 に移らせた。meiro1 に比べてゴールするためのブロック数多く、壁も多いためクリアするまでに多くの時間をかけていた。meiro2 を見た児童は広さや複雑さが増したことへの驚きの反応示していた。また、難易度が上がったことによりクリアする意欲が増した児童もいた。しかし、meiro2 に取り組ませてから7分程を経過した時点では8割程度の児童がクリアして2~3回目を解いていた。

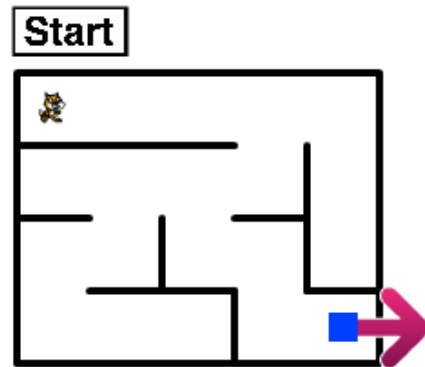


図 7-7 meiro1

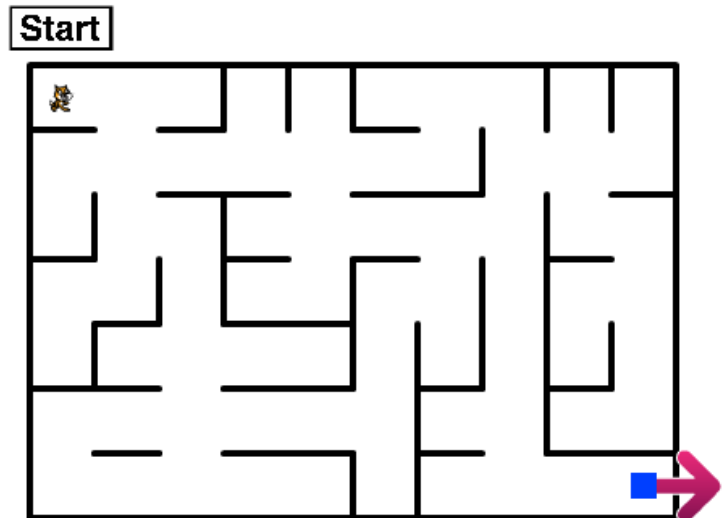


図 7-8 meiro2

7.3 tightVNC の利用

「めいろゲームプログラミング」では、児童の活動への取り組みの様子が一目で確認できるように、tightVNC を活用した(図 7-9)。児童が使用するタブレット端末の画面を、授業者が操作するパソコン画面に一齐に映し出したことで、タブレットの不具合で課題を進められない児童に気づき、すぐに対応することが出来た。また、授業者のパソコンの画面を見せることも出来るため、本題材の授業以外にも効果を発揮するツールである。

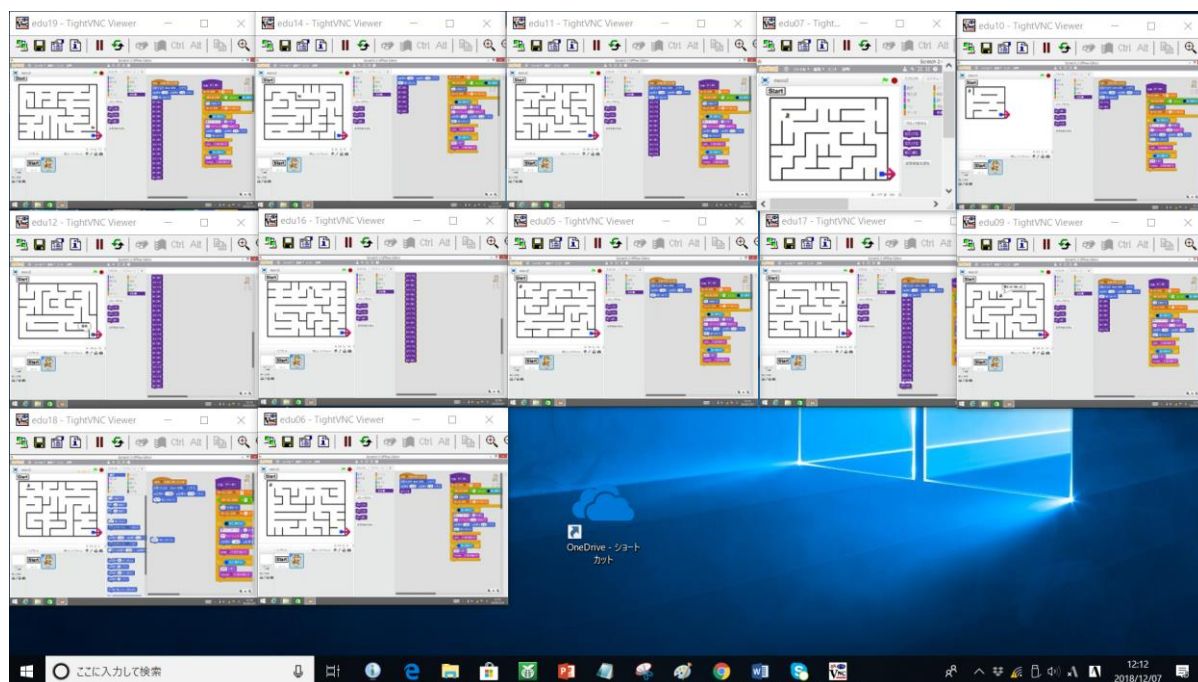


図 7-9 tightVNC 起動中の画面

7.4 実践で工夫した点

小学3年生が興味をもってプログラミングを行うことができるようなツールを考える際に、子どもたちの身近な迷路を作ってクリアするようなゲームが適していると考えた。扱うプログラムも逐次的に並べるだけであるため、操作や思考の何度も低いと考える。また、並べたブロック塊で同じ動きがあれば繰り返しを使ってプログラムをまとめる書き方を教えることも出来る。

最初は meiro2 だけを用いて授業を行う予定であったが、児童のプログラミングに対する理解度はそれぞれで異なるためクリアする早さに大きく差が出てしまうこと予想した。また、段階を踏んで難易度を上げることで抵抗を取り除くことを目的としてステージを難易度別の2つに分ける工夫をした。

練習問題を解く際に、パソコンの画面ではなくホワイトボードに迷路を書き、紙で猫とブロックを作って貼ることで、児童の反応に合わせて貼るブロックや猫の動きを変えるこ

とができた。このように説明することで、貼ったブロックと猫の動きを対応させながら確認し、児童と一緒に答え合わせができる。結果的にもし途中で間違っただとしても、猫の誤った動きを見たりすることで間違っただ考えに気付くことができる。一方的な説明ではなく児童と考えながら活動することで、分かっていない子どもも理解している子どもの考えに触れることができるため、苦手をなくす方法としても効果は高いだろう。

7.5 授業における児童の様子

授業で対象となった児童は12名であり、全児童が meiro2 までブロックを繋げてクリアすることが出来ていた。授業後に頂いた感想シートから、特に児童が難しく感じていたことが2点分かった。1点目は迷路のステージが大きくなるほど、繋げるブロック数が多くなるため、途中でどこまでの猫の動きをプログラムしたのか分からなくなってしまったことである。そして猫が壁に当たった時に、どこの部分が間違っているのかを見つけ出すことが難しかったようである。その場合の対応として、ブロックが長くなった場合繰り返しブロックでまとめることでブロック数が減り、間違えた部分を探しやすくなったとTAから報告を受けた。Scratch 経験者の子どもは初めから繰り返しブロックを使って取り組む様子が見られた。見やすいプログラムを工夫して作っている場合、間違えたときに対処しやすくする方法として紹介すると良いだろう。2点目は、猫が向いている向きから「右を向く」「左を向く」のどちらを繋げれば、意図した方向に向くのか考えることが難しかったようである。このように、順を追って次にどのような動きをすれば良いのか1つひとつ立ち止まって考えたことに、この題材の意義があったと考える。

7.6 まとめ

大小2つのステージを設けたが、どちらもほとんどの児童がクリアすることが出来ていたため、難易度としてはやや低いものの順序立ててゴールまでの道筋を考え、ブロックを繋げさせる目標は達成できた。

反省としては想定していたよりも meiro1 をクリアする児童が早く出たため、さらにステージの難度を上げて段階を増やすことも考えていきたい。例えば、障害物を置いて通ることが出来る道を絞ったり、通過ポイントを指定して必ず通らなくてはならない道を設けたりする何度調整が思いつくが、その場合はプログラムの仕様を大きく変更しなくてはならないだろう。

また練習問題をホワイトボードで説明する際、理解している子どもの反応に合わせ過ぎてしまったことを反省している。ほとんどの子どもが頷きや自身の考えを口にしてはいたが、静かに聞いている子どもに対して目線を送ったり、問いかけで理解を確認したりするべきであった。

第 8 章 Scratch を用いた「一筆書きで図形描画プログラミング」

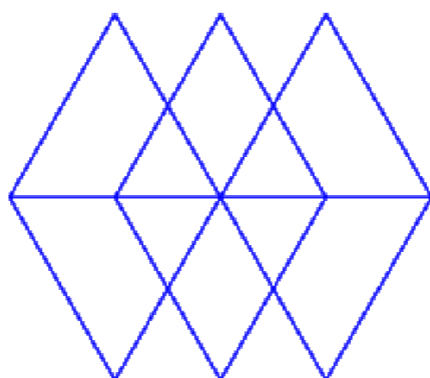
8.1 題材概要

本題材はプログラミング的思考力を育むため、小学 4 年生向けに開発したものである。児童に描かせたい図形を提示し、一筆書きするためのプログラムを作らせる。同じ図形を描くプログラムは書き方によっていくつもあるが、できる限り短く簡潔なプログラムを考えることが基本である。そのため、本題材では与えられた図形の最小単位を考え繰り返す書き方や、図形のどの箇所から描くことで一筆書きが出来るのかを試行錯誤しながら考えさせることを目指す。

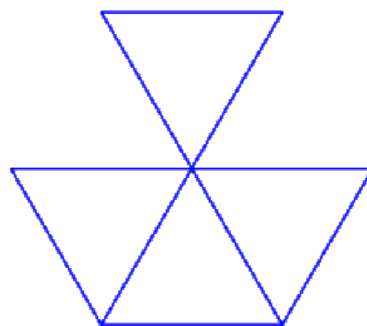
8.2 題材の詳細

8.2.1 用意した図形

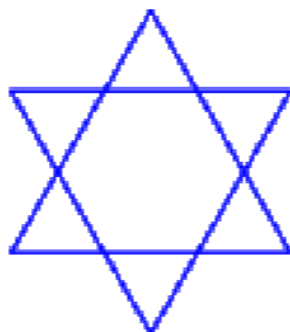
今回用意した図形は以下に示す 3 つ(図 8-1)である。以降説明のために左から図形 1.2.3 の名前を付けることとする。



(a)図形 1



(b)図形 2



(c)図形 3

図 8-1 用意した図形

図形 1,2 については全体の図形を構成している最小単位を繰り返す考え方が用いられている。図形 3 については、書き始める始点を考えることで短い手数で書き上げることが出来る。それぞれの図形のプログラムは以下の図 8-2。

```

    が押されたとき
    x座標を -100 、y座標を 0 にする
    90 度に向ける
    全部消す
    ペンを下ろす
    3 回繰り返す
        60 度回す
        100 歩動かす
        120 度回す
        100 歩動かす
        60 度回す
        100 歩動かす
        120 度回す
        100 歩動かす
        120 度回す
        50 歩動かす
    50 歩動かす
    
```

(a)図形 1 プログラム

```

    が押されたとき
    x座標を 50 、y座標を -50 にする
    -30 度に向ける
    全部消す
    ペンを下ろす
    3 回繰り返す
        200 歩動かす
        120 度回す
        100 歩動かす
        120 度回す
    60 度回す
    100 歩動かす
    
```

(b)図形 2 プログラム



図形3 プログラム
図 8-2 各図形のプログラム

8.3 各図形を描く流れ

(1) 図形 1 を描く流れ

1) 座標と向きを決める

図形1を描く始点と向きを決定するためのブロックである(図 8-3)。図形 1 はダイヤ型が3つ連なっている図形だが、一番左のダイヤ型から右にずれて書いていく。そのため、座標は画面のやや左に設定してある。ペン機能を持たせた猫のスプライトは 90 度(画面右)を向いてから描き始めていく。

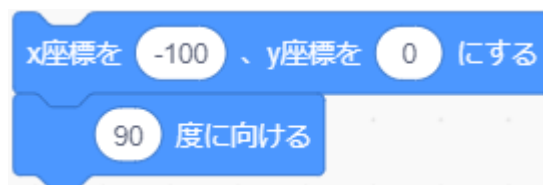


図 8-3 座標と向きの決定

2) 図形 1 の最小単位の書き順

図形 1 は図 8-4 の図形が最小単位となっている。赤い丸は始点を表しており、そこから反時計回りに 60 度回転し、赤矢印の番号順に進み菱形を描く。再び始点まで戻ってから 120 度時計回りに回転して 50 歩進む。ここまですべての最小単位を描く手順である。⑤まで書き終え同じ動作をあと 3 回繰り返し、50 歩進むことで、ダイヤ型が 3 つ連なった図形 1 を描くことができる。

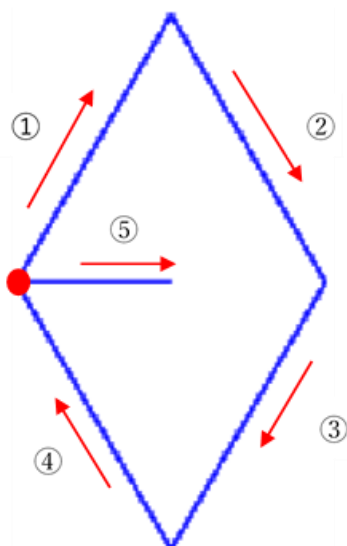


図 8-4 図形 1 の最小単位

(2) 図形 2 を描く流れ

19 座標と向きを決める

図形 2 を描く始点と向きを決定するためのブロックである(図 8-5)。図形 2 は正三角形が下行に 3 つ、上行真ん中に 1 つ描かれる図形である。一番右の三角形の辺から描いていく。そのため、座標は画面の右下に設定してある(図 8-6)。

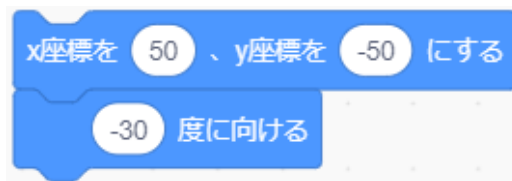


図 8-5 座標と向きの決定

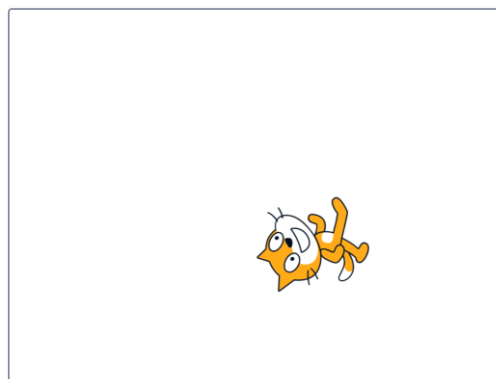


図 8-6 図形 2 を描く際の初期位置

2) 図形 2 の最小単位の書き順

図形 2 は図 8-7 の図形が最小単位となっている。赤い丸は始点を表している。-30 度(画面左上)を向き正三角形 2 つ分の辺を描き 120 度時計回りに回転し、正三角形 1 つ分の辺を描く。ここまですが最小単位である。この動作をあと 2 回繰り返すことで再び始点まで戻る。そこで反時計回りに 60 度回転して一辺を描けば完成である(図 8-8)。

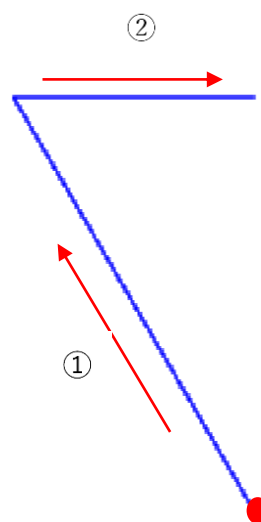


図 8-7 図形 2 の最小単位

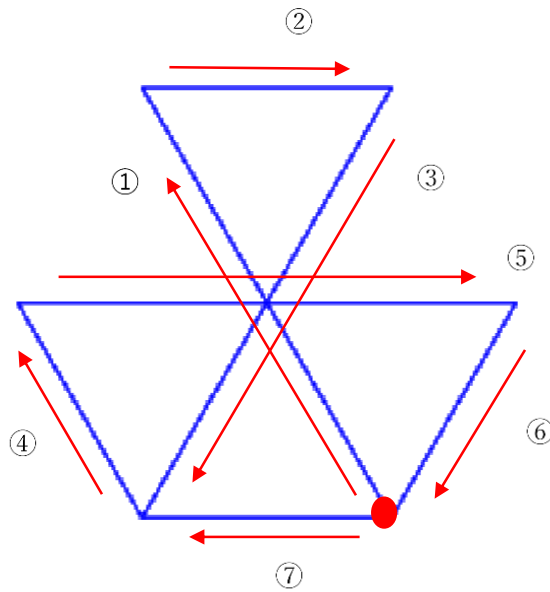


図 8-8 図形 2 全体の書き順

(2) 図形 2 を描く流れ

1) 座標と向きを決める

図形 3 を描く始点と向きを決定するためのブロックである(図 8-9)。図形 3 はダビデの星形である。ダビデの星形は上向き、下向きの三角形がそれぞれ重なって出来ている。2つの三角形の右下の交点を始点として描いていくため、-150 度(画面左下)に向けるために設定してある(図 8-10)。

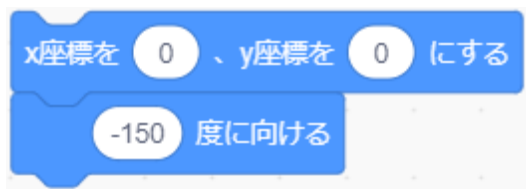


図 8-9 座標と向きの設定

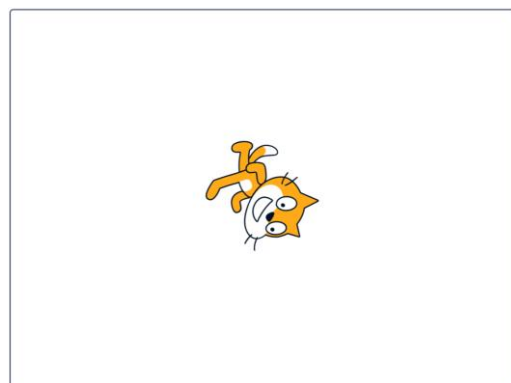


図 8-10 図形 3 を描く際の初期位置

2) 図形 2 の最小単位の書き順

図形 3 の描き順は図 8-11。赤い丸は始点を表している。ダビデの星形は上下向きが異なる三角形が 2 つ重なっている。よって 2 つに分けて描き順を示す。まずは図 8-11 に示したように赤点を始点にして時計回りに三角形を描く。始点に戻ってきたら、時計回りに 60 度回転して 2 つ目の三角形を描く。図 8-11,12 の赤丸(始点)は重なっている。下向きの三角形を描き終えたら上向きの三角形も時計回りに描いて完成である(図 8-12)。

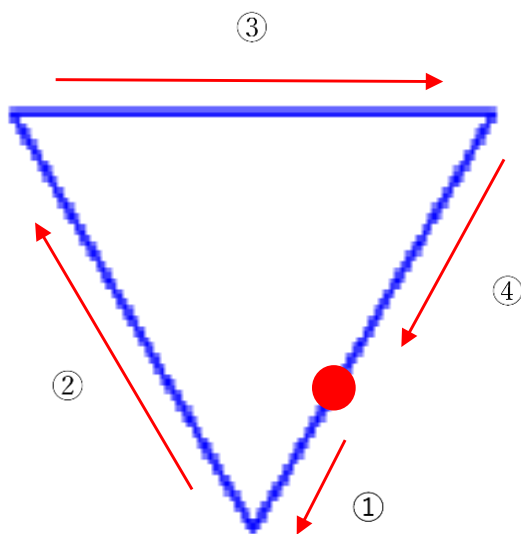


図 8-11 図形 3 の描き順前半

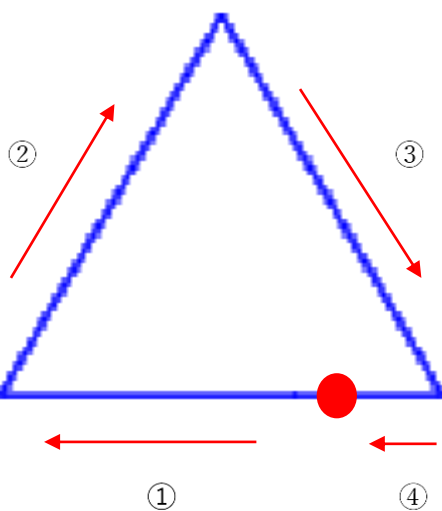


図 8-12 図形③の描き順後半

8.4 指導のポイント

描画する図形は同じ動作を繰り返し行ったり、描き始めの点を考えたりすることによって短く簡潔なプログラムで描くことが出来る。そのため、図形を観察し最小単位を見つけたり、ブロックの数が少なくすむ描き方に注目したりして取り組ませることがポイントになる。しかし、いきなり最小単位といっても児童にはイメージすることが難しいため、練習用の図形を用意しブロックを1つひとつ繋げて図形を描き、そのプログラムを観察してどのような動作を何度も行っているかを図形と見比べて考えてみると良いだろう。図形2のように「正三角形」という、知っている図形を繰り返し描くのではなく、正三角形2つ分の辺、1つ分の辺を交互に描く結果複数の正三角形が出来る図形もある。そのため知っている図形を繰り返すという考えに囚われないことも大切である。

8.5 まとめ

本題材は、ペン機能と回転ブロック、移動ブロックを用いて図形を描くという題材であるが、特に児童に身につけて欲しい考え方は「より最適な方法で描く」ということである。これは、プログラミング的思考力に通じた思考で与えられた課題達成のために、どのような動作が何回必要で、どのように組み合わせれば最も短く早くすることが出来るのかを考える力である。それを実感させるために、課題を達成するためにはいくつもの方法があり、その中で最も目的に沿った方法を選択する思考を養わせる活動として、本題材は適している。授業では、提示した図形をそれぞれの児童にプログラムを組んで描かせ、それらを共有して比較する活動をしてもらいたい。また、その際なぜその描き方にしたのかを説明することが出来ることが重要である。Scratchを用いた図形を描く授業では「ブロックを並べたらなぜか図形が出来た」という児童をよく見かけるが、偶然ではなく意図的に描かなくては力にならない。試しながら取り組むことは必要だが、考えを持って行うことが大切であることは伝えていかなくてはならないだろう。

第9章 JavaScript を用いた「おみくじゲームプログラミング」

9.1 実践概要

筆者が学士であった2017年度に行った小学5年生を対象とした、JavaScriptを用いて「おみくじゲーム」を作る実践(図9-1)である。対象児童は前年度にScratchを使ってプログラミングを経験していたことで、逐次処理や条件分岐についての理解があったため、コーディングベースのプログラミングにTA付きで挑戦してもらった。具体的には変数を用いて打ち出した数に応じて手を出す「じゃんけんゲーム」のグー・チョキ・パーの手を、おみくじの結果(大吉・凶・大凶)に出力割合を変更して書き換えさせる課題に取り組ませた。結果は全児童において課題を達成し、「タイピング自体は難しかったが、プログラミングへは楽しく興味をもって取り組むことが出来た」という旨の感想を聞くことが出来た。

```
<html>
<body>
氏名
<br>
○年
<br>
<script>
function omikuzipon() {
  var ransuu = Math.random() ;
  if(ransuu <= 0.15) {
    document.body.innerHTML += "だいきち";
  }
  else if(ransuu <= 0.95) {
    document.body.innerHTML += "きょう";
  }
  else {
    document.body.innerHTML += "だいきょう";
  }
  document.body.innerHTML += "<br>";
}
</script>
<input onclick="omikuzipon();" type="button" value="うらなう" />
<br>
</body>
</html>
```

図9-1 おみくじゲームプログラムソース

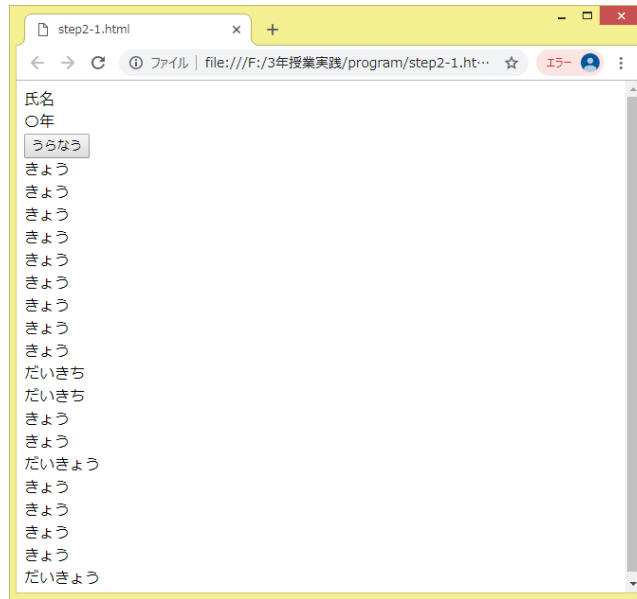


図 9-2 おみくじゲーム実行結果

9.2 授業の流れ

(1) HTML でブラウザに表示されることの説明

予め配っていただいた「じゃんけんプログラム」の説明を児童に行く前に、普段見ているウェブページ等も文章によるプログラムによって表示されるように作られていることを説明し、これから行うコーディングとブラウザで実行結果を見られることの関係性をつかみやすいように導入を行った。

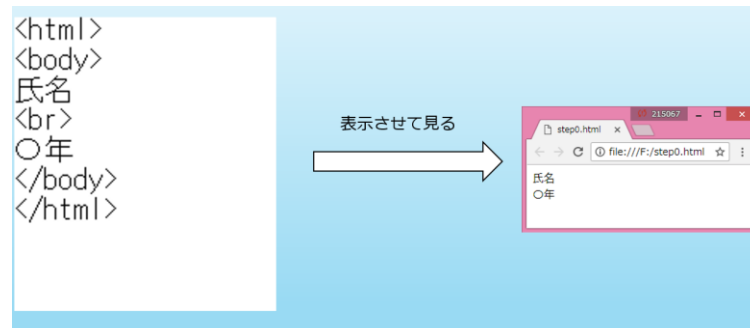


図 9-3 HTML でブラウザに表示されることの説明画像

図 9-3 を見せて<html>と</html>はホームページのもとになる文であることを示していること、<body>と</body>では含まれた文(氏名、
、○年)が表示されること、
は行を変えるタグであることを説明した。

(2) HTMLでは動的な出力ができないことの説明

ページを開くたびにじゃんけん
の「グー」「チョキ」「パー」が変
わって表示されるようにしたい
が、HTMLではプログラムを毎回
書き換えなければ同じ手が表示さ
れてしまうという問題点を説明
し、JavaScriptは動的な出力が可
能であることの違いを述べたう
えで、「じゃんけんプログラム」の説
明に進んだ。

図9-4の画像の例では<body>と
</body>の中に「グー」と書き入れることで、ブラウザに「グー」が表示されるよ
うになったが、「チョキ」「パー」が「グー」に変わり表示されるようにするには動的な
JavaScriptの文で書く必要があると説明した。

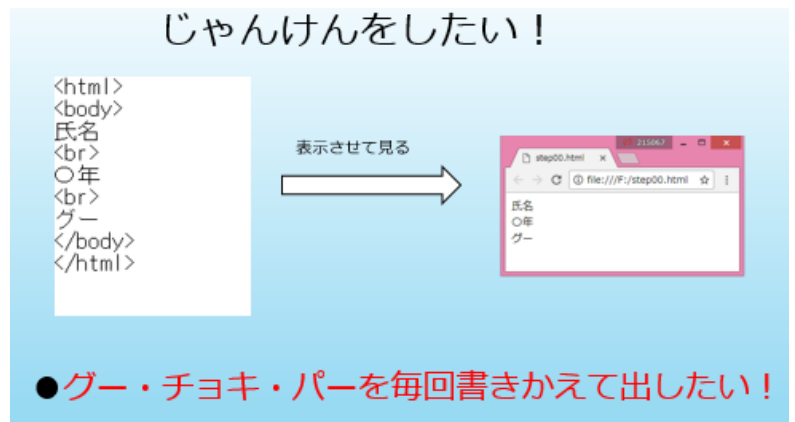


図9-4 HTMLでは動的な出力はできない説明画像

(3) 「じゃんけんプログラム」についての説明

図9-5は児童のパソコンに予め配っ
ておいたプログラム(画面左ウインド
ウ)と実行結果(画面右ウインドウ)であ
る。<script>と</script>タグが加わ
り、その中にコンピュータが「グー」
「チョキ」「パー」のどれかを出すプロ
グラムが書かれていることを説明し、
実行結果を隣のウインドウに表示し、
更新するごとに表示される手の種類が
変わる動的な出力が出来ることを実際
に児童に触らせて確かめさせた。その
後に、<script>と</script>タグの中に
書かれている文がどのような役割になっ
ているのかを1つずつ流れに沿って説明した。

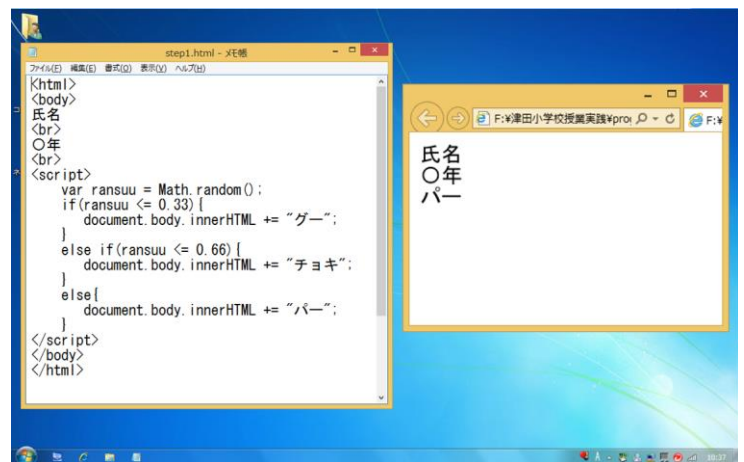


図9-5 JavaScriptで動的な出力が変わることの例

次に乱数についての説明を行った。図 9-6 を見せて script タグ中の 1 文目が乱数を 1 つ決定し変数である ransuu に代入する役割であることを説明した。今回の授業では、Math.random()関数を用いて 0.0 以上 1.0 の値が取り出されるようになっている。変数という表現に児童はあまり慣れていないことを予想し、説明時には ransuu を入れ物として捉えるように説明した。

図 9-7 のように、箱から数が出てきて ransuu という入れ物に入っていくアニメーションを作り、児童が視覚的に流れを捉えてイメージしやすいようにした。パワーポイントを用いた説明では、0.54 が取り出されたと仮定し、この後の説明にも利用した。

次に乱数を利用して、if 文で出す手の判定をする考え方を説明した。

図 9-7 の説明で取り出した 0.54 を用いて、ブラウザに表示される手の種類がどのように判定されるのかを図 9-8 で説明した。if 文をフロー図にして、取り出した乱数が「0.33 以下であるか」、「0.66 以下であるか」を判定して、表示される手が「グー」「チョキ」「パー」の 3 種類になるようになっている。

1 つ数を取り出す

① `var ransuu = Math.random();`
(1)

(1) 0 以上 1 より小さい数を 1 つ決める (1)

(2) ransuu という変数(入れ物)に(1)で決まった数を入れる (2)




図 9-6 乱数の説明

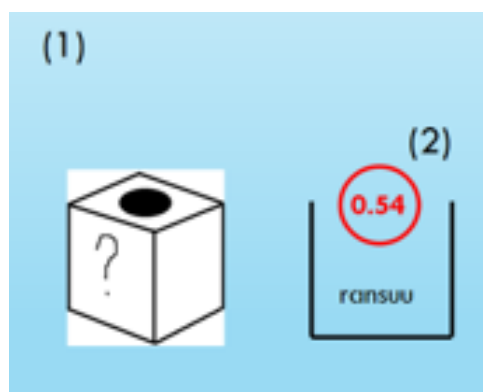


図 9-7 乱数に変数 ransuu に代入されるイメージ図

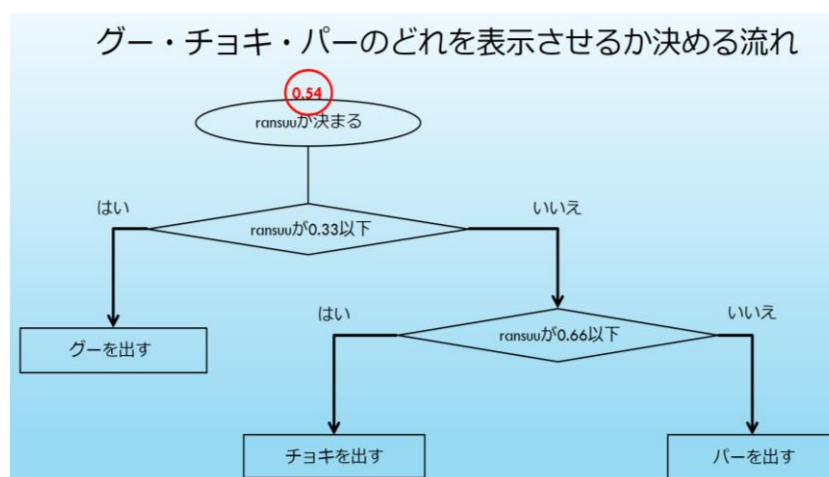


図 9-8 「グー」「チョキ」「パー」の判定の方法説明図

図 9-9 を用いて取り出した乱数が「0.33 以下であるか」、「0.66 以下であるか」の判定をする理由を説明した。3 種類の手が表示されるのはそれぞれ「グー」が 33%、「チョキ」が 33%、「パー」が 34%のほとんど同じ割合であり、それを実現するために 0.33 や 0.66 という数値を設定した。

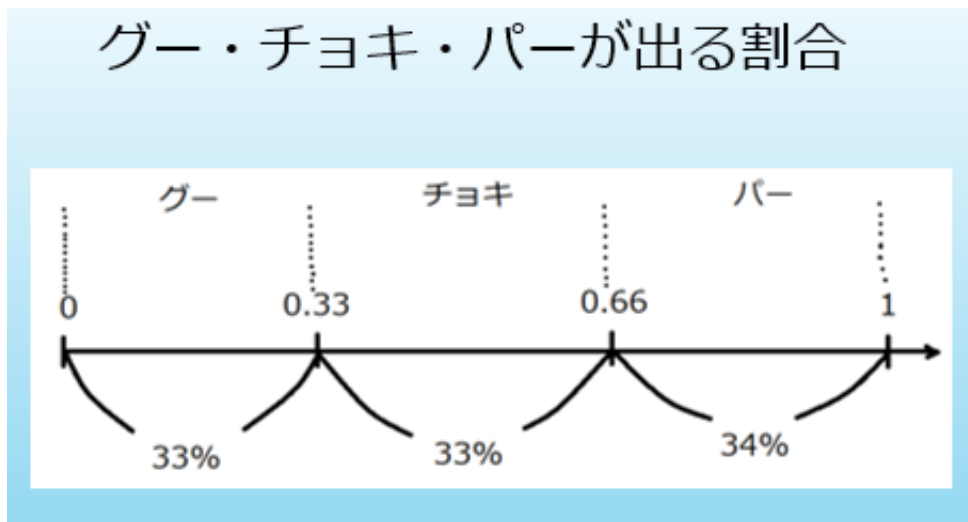


図 9-9 表示される手の割合

児童には上の数直線を区切ることで幅として考えることで、それぞれが表示される割合の大きさの関係が決まると説明した。例えば、「グー」をもっと表示されやすくしたいのならば、0.33 をより大きい数値にすることで数直線上の「グー」が占める幅も大きくなるため、if 文での判定で設定した数を書き換えればよい。つまり表示される割合を変えたいのならば、判定する数値を書き換えて幅を大きくしたり小さくしたりすれば良い。

図 9-10 を見せながら、乱数を取り出した後の流れをプログラムで書くと、`<script>`と`</script>`タグ中の 2 行目以降のようになり、3 行ずつに分け先にフロー図で説明したことと同じ内容であることを解説した。初めて文章でのプログラムを見る児童がほとんどであったため、「if」や条件 () 内)、「document.body.innerHTML += " ";」がそれぞれどのような意味をもっているのかを解説しながら説明した。

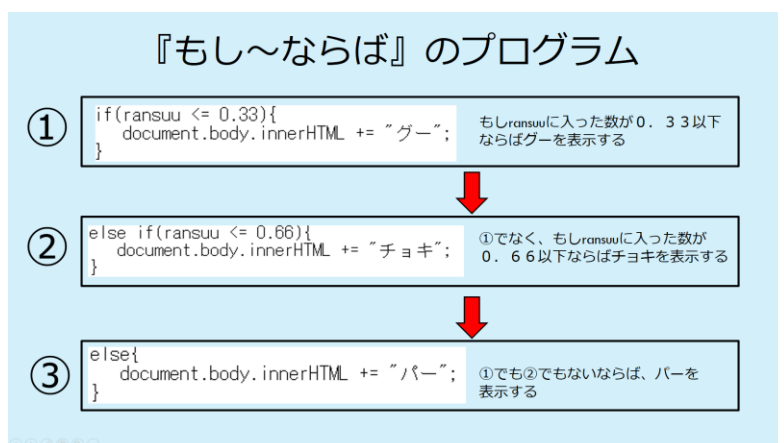


図 9-10 if 文の説明図

(4)「おみくじゲームプログラム」に書き換えさせる

「じゃんけんプログラム」の内容について説明した後に、配っておいた「じゃんけんプログラム」を「おみくじプログラム」に書き換えさせる課題に取り組みさせた。

問題の内容は図9-11に書かれているように、大吉・吉・凶の3種類がそれぞれ10%、70%、

20%の割合で表示されるようなプログラムに書き換えることである。ヒントとして、図9-9の図を付けて判定条件の数値を変え、数直線に占める各パラメータの幅を調整すれば良いということを書いておいた。

図9-11のページをホワイトボードに映していたところ、児童が図9-9の数直線に両手を当てその間隔を広げたり狭めたりする姿を確認できたことから、割合を幅によって視覚化したことで、くじが表示される割合の大小関係がイメージしやすいと感じていたのだと分かる。

図9-12は図9-11の問題の解答である。

文章内の線が引かれた部分が児童に書き換えさせる箇所になっていた。

もんだい1

じゃんけんプログラムを書きかえて、おみくじプログラムを作りなさい。

くじはだいきち・きち・きょうの3しゅるいで
だいきち：10%
きち：70%
きょう：20%
の割合で出るようにすること。

ヒント
0.33と0.66の数を書きかえて、
はばの大きさをかえる

0 0.33 0.66 1
グー チョキ パー
33% 33% 34%

図9-11 「おみくじプログラム」に書き換えさせる問題

もんだい1の答え

```
<html>  
<body>  
  氏名  
<br>  
  ○年  
<br>  
<script>  
  var ransuu = Math.random();  
  if(ransuu <= 0.1){  
    document.body.innerHTML += "だいきち";  
  }  
  else if(ransuu <= 0.8){  
    document.body.innerHTML += "きち";  
  }  
  else{  
    document.body.innerHTML += "きょう";  
  }  
</script>  
</body>  
</html>
```

図9-12 問題1の解答画像

(5) ボタン付きおみくじプログラムの説明

図 9-13 は児童のパソコンに予め配っておいた 2 つ目のプログラムである。

「<input onclick="omikuzipon();" type="button" value="うらなう" />」によって、ボタンを表示し、ボタンを押すことによって、関数「function omikuzipon(){}」の中にあるプログラムが動く仕組みになっていることを説明した。

ボタンをつけてみる

step2.html

```
<html>
<body>
氏名
<br>
〇年
<br>
<script>
function omikuzipon(){
var ransuu = Math.random() ;
if(ransuu <= 0.15){
document.body.innerHTML += "だいきち";
}
else if(ransuu <= 0.95){
document.body.innerHTML += "ぎょう";
}
else{
document.body.innerHTML += "だいぎょう";
}
document.body.innerHTML += "<br>";
}
</script>
<input onclick="omikuzipon();" type="button" value="うらなう" />
<br>
</body>
</html>
```

ボタンをおしたら
赤わく内のプログラムが動く

ボタンを表示させる

図 9-13 ボタン付きおみくじプログラムの説明図

(6) 問題2 提示

図 9-14 によって説明したボタン付きおみくじプログラムを見童達が問題の指示通りに書き換える課題に取り組ませた。元々のプログラムに、「吉」が表示される文の追加と、4種類のくじが表示される割合の調整ができれば課題クリアとなる。

ヒントとして、「else if(){ }」文を追加して、他の行と見比べて条件()内を書き入れていくことで正解へアプローチしていけることを説明した。

図 9-15 は問題 2 の解答を見童に発表したときの画像である。枠内で囲まれた文を追加し、if文の条件()内の数値を乱数の値が0.75 以下であるように設定できていれば正解となる。

見童の様子を机間巡視によって観察していたところ、紙に数直線を書き区切り線を入れ、幅

の大きさを書き入れていた姿を見て、問題文のパーセンテージに惑わされることなく 4 種類のくじそれぞれが表示される割合が、数直線の幅の大きさであることを理解できていたと見て取れた。

もんだい2

新しくおみくじプログラムを作りなさい。

くじはだいきち・きち・きょう・だいきょうの4しゅるいで
だいきち：15%
きち：60%
きょう：20%
だいきょう：5%
のあたりやすさで出るようにすること。

ヒント
else if ({ })の文を1つ追加する

図 9-14 おみくじプログラムを書き換える課題説明図

もんだい2の答え

```
<html>
<body>
  氏名
  <br>
  〇年
  <br>
  <script>
  function omikuzipon(){
    var ransuu = Math.random() ;
    if(ransuu <= 0.15){
      document.body.innerHTML += "だいきち";
    }
    else if(ransuu <= 0.75){
      document.body.innerHTML += "きち";
    }
    else if(ransuu <= 0.95){
      document.body.innerHTML += "きょう";
    }
    else{
      document.body.innerHTML += "だいきょう";
    }
    document.body.innerHTML += "<br>";
  }
  </script>
  <input onclick="omikuzipon();" type="button" value="うらなう" />
  <br>
</body>
</html>
```

図 9-15 問題 2 の解答図

図 9-16 は問題 2 の解説に用いたスライドである。多くの児童が紙に同じ図を書くすがたを確認することができた。中には、問題 2 の問題文をホワイトボードに映していたところに「じゃんけんの説明の時に使っていた割合の絵(図 9-9)が見たい」と言ってきた児童もいて、視覚化することの重要性を感じた。

この図を使うことを決めた理由が、問題文に書かれた数値をそのまま打ち込む児童が出てきてしまう可能性を考えたためである。例えば、

```

if(ransuu <= 0.15){
    document.body.innerHTML += "だいきち";
}
else if(ransuu <= 0.60){
    document.body.innerHTML += "きち";
}
else if(ransuu <= 0.20){
    document.body.innerHTML += "きょう";
}
else{
    document.body.innerHTML += "だいきょう";
}
document.body.innerHTML += "<br>";
}

```

のような誤りである。4 行目と 7 行目の下線を引いた数値のように誤った数値を書き入れてしまう間違いを予想していた。そこで、全体を 1 とした時の幅が占める割合がくじの出やすさであると説明すれば分かりやすいと考えての試みであったが、狙い通りの結果が得られた。

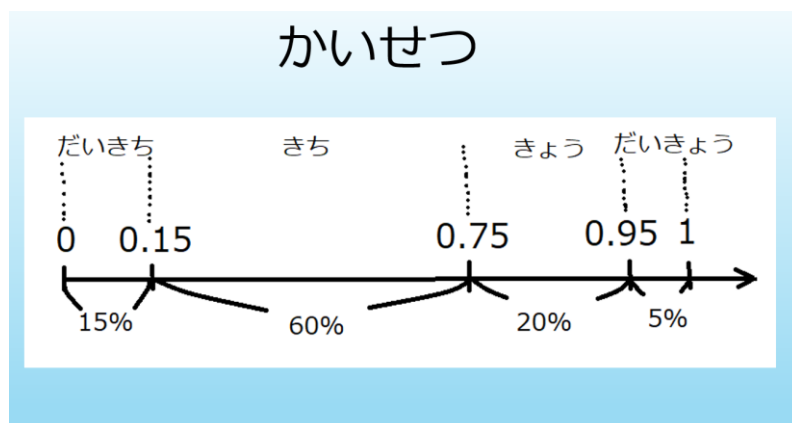


図 9-16 問題 2 の解説図

(7) 発展した「じゃんけんプログラム」の配布と説明

児童にプログラミングに興味をもってもらうことを狙いとして、実践で扱った「じゃんけんプログラム」が発展するとどのようなゲームになるのかを知ってもらうために、児童に予め配っておいたプログラムで遊ばせた後に、ソースコードを紙に印刷し配布した。発展した「じゃんけんプログラム」の内容は勝ち負けが表示されるプログラムである。以下にソースコードと実行結果の例を挙げる。

<ソースコード>

```
<html>
<body>
氏名
<br>
〇年
<br>
<script>
function jankenpon(anatanote) {
  var ransuu = Math.random();
  var keisankinote = Math.floor(ransuu * 3);
  var janken = ["グー", "チョキ", "パー"];
  document.body.innerHTML += "計算機の手は";
  document.body.innerHTML += janken[keisankinote];
  document.body.innerHTML += " : ";
  document.body.innerHTML += "あなたの手は";
  document.body.innerHTML += janken[anatanote];
  document.body.innerHTML += " : ";
  if(keisankinote == anatanote) {
    document.body.innerHTML += "あいこ";
  }
  else if(anatanote - keisankinote == 1 ||
    anatanote - keisankinote == -2) {
    document.body.innerHTML += "あなたの負け";
  }
  else {
    document.body.innerHTML += "あなたの勝ち";
  }
  document.body.innerHTML += "<br>";
}
```

```

}
</script>
<input onclick="jankenpon(0);" type="button" value="グー" />
<input onclick="jankenpon(1);" type="button" value="チョキ" />
<input onclick="jankenpon(2);" type="button" value="パー" />
<br>
</body>
</html>

```

<実行結果>

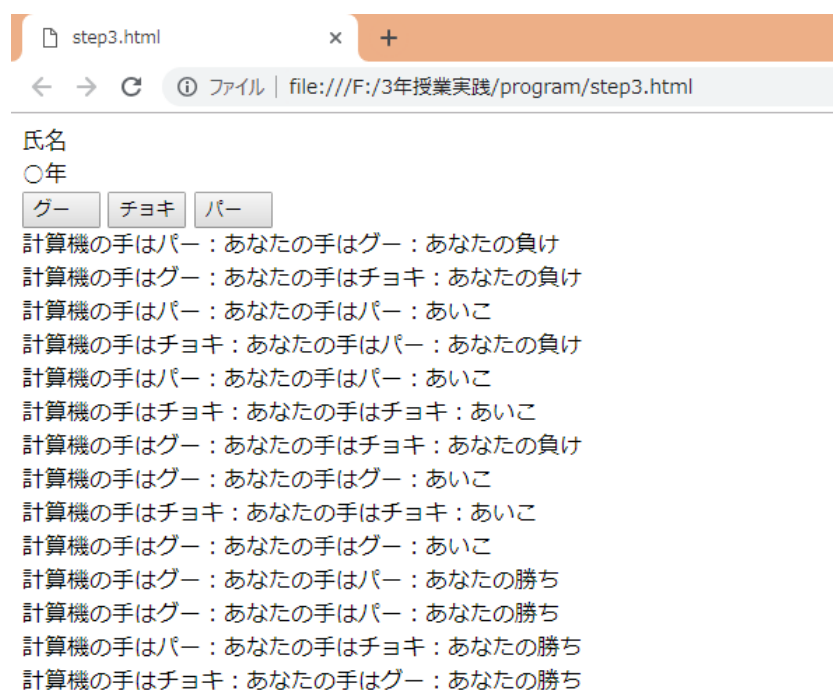


図 9-17 発展した「じゃんけんプログラム」実行結果の例

図 9-17 は児童に配付した発展したじゃんけんプログラムの実行結果である。手の種類ごとにボタンが用意されており、計算機の出した手とユーザーの選んだ手、そして勝敗が表示されるようになっている。

9.3 授業で工夫した点

今回の授業は児童が経験してきた Scratch のようにブロックを繋げてプログラムを作るのではなく、文章で打たれたプログラムに抵抗を感じる児童が出てくるのではないかと予想していた。そのため、できる限り役割をもっている一文や一つの塊を取り上げ説明し、それを図やアニメーションによって視覚し、理解させることができるように工夫した。実践の流れでも紹介したが、乱数を取り出した後に変数に代入されるアニメーションや if 文のフロー図、じゃんけんの手やどの種類のくじがでるのかは割合で決まっています、それは数直線に 0~1 の範囲をとった時にそれぞれが占める割合で考えることができるという方法の紹介などである。

2 つ目の工夫点は時間を短縮するための工夫である。授業は 1 時間完結であったため、何もない状態からプログラムコードを書いている時間内に終わらなかった。そこで予め書き換え前のプログラムを配っておき、課題を考える時間を確保することができたという点である。数値の入力やコピー&ペーストによって文章を追加するだけの状態のプログラムを配っておくことで、割合を調整するための計算やどのような文章を追加すればよいのか等について考えるための時間を多くとる必要があったため、この工夫は有効であったと考えられる。

9.4 授業における児童の様子

まずコーディングのプログラミングは初めてであったことから、文字列を間違えることなく打ち込むことに難しさを感じていた。また、英単語の文字列を入力することに慣れていないため、TA にキーの位置を教えてもらいながら書き上げていた。基本のプログラムを配布しておき、書き換えさせる方法によってタイプ量を減らしたことで正答を書き上げることが出来ていたのだと考える。また、パワーポイントでのプログラムや考え方を説明する時間を十分にもてたことが、児童の理解を助けただろう。

児童 1~3 人に対して TA1 人が付く体制であった。授業後児童の様子について尋ねたところ、助けた部分はタイピング部分で、プログラムの構造や課題の考え方は児童自身でヒントを踏まえて考えることが出来ていたと報告を頂いた。このことには前年度に 1 年間 Scratch を用いてアニメーション等の作品作りを経験していたことが、影響しているだろう。一通りの作品作りによって基本的なプログラミングの要素を学び、実際に使って作る体験をしていたからこそ、プログラムの構造は理解できたのだと考える。

9.5 今後の課題とまとめ

JavaScript は文字ベースによるプログラミングであるため、タイピングに慣れていない児童の場合、時間が掛かってしまう点である。これでは、プログラムの構造や課題に取り組む時間が減ってしまう可能性がある。実践では文中の数値を書き換えたり、コピー&ペーストして文章を追加したりと比較的簡単な作業を行わせるようにして児童の負担を少な

くすることで課題を考えることに専念させた。また、実践では TA にタイピングのサポートをして頂いたが、教育現場では大勢のサポートを付けることはできないため、タイピングやパソコン操作の基礎は低学年の内から学習しておくことが望ましい。そのために、校内での連携をとり発達段階にあわせたプログラミング教育向けの指導計画を立てるべきである。

2つ目の課題は、児童の作ったプログラムに誤りがあった場合に、エラー箇所を探すことに時間が掛かってしまい、全体に指示したり他の質問に応えたりする時間が減ってしまうという点である。この実践では、予め用意していたプログラムを書き換えさせるだけであったため、書き換えた箇所を主に確認すれば良かった。しかし、レベルが上がり児童一人ひとりが異なるプログラムを書く授業を展開するならば多くの児童が躓くことが予想される。この問題点については課題の設定を、使用する頻度の多い処理や文章を限定したり、雛形を用意しておき児童に選んで使わせたりするなどの改善方法がある。

時間短縮やスライドでの説明に工夫をして1時間で授業を終えることが出来た。教育現場で扱う際には、もっと時間数に余裕を持たせることが必要である。また、説明の工夫については、今回はスライドのみであったが実物を使った説明も効果的である。例えば、乱数の説明で抽選箱を用意して、そこから無作為に数が書かれたくじを引く動作に例えてみるなどである。抽象的な考え方であるからこそできる限り具体物を使って仕組みを理解出来るように伝える努力は必要である。

第 10 章 Scratch を用いた「花模様を描くプログラミング」

10.1 実践概要

2016 年より Scratch を使って図形描画させる授業実践に取り組んでいる。星や正多角形を描かせたり、多角形の基本図形をずらしながら描画する幾何学図形や、床模様等に使用されるタイルパターンを描かせたりしてきた。そして 2019 年度は、桜と菊の花模様テーマにして描かせた(図 10-1)。花びらの一片は屈折線分で近似して描画されるが、線分の本数を増やすほどに、花びらは滑らかな曲線に近付いていく(a1→a2→a3, b1→b2→b3)。最初に屈折線分の数の少ない花模様(a1, b1)を描くプログラムを児童に配布し、その屈折回数を増やし滑らかな花びらに変えていく課題を与えた。

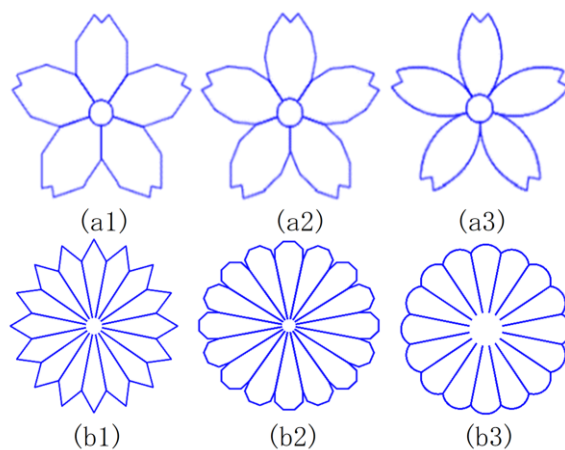


図 10-1 描かせた花模

10.2 実施内容

2019 年 9 月 4 日と 9 月 18 日の 2 日間で、津市立西が丘小学校 5 年生(全 135 名)を対象に、Windows 端末を用いて各クラス 1 回ずつ 45 分間の授業を行った。表 10-1 は授業実践当日の日程や参加して頂いた TA の人数等をまとめたものである。

表 10-1 授業実践の日程

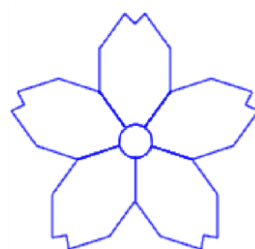
授業実践日	授業時間	TA
9 月 4 日(水)	1 回目 11:30~12:15 2 回目 13:35~14:20	情報教育 2 年 5 人+3 年 2 人 +4 年 6 人
9 月 18 日(水)	3 回目 11:30~12:15 4 回目 13:35~14:20	情報教育 2 年 5 人+4 年 4 人

プログラミング環境は Scratch Desktop を予め各端末にダウンロードし、それぞれに授業者が屈折回数の少ない花びらを描くプログラムファイル(図 10-2,10-3)を配布しておいた。また、授業者はパソコンとプロジェクターを 2 台ずつ準備していき、予め用意していたパワーポイント資料と児童と同じ Scratch の画面を映したパソコンそれぞれ 2 台を、TV には Scratch の画面・スクリーンにはパワーポイント資料を映して交互に見せて説明を行った。

10.3 配布したプログラムの説明



②



(b) (a)のプログラムで描かれる桜

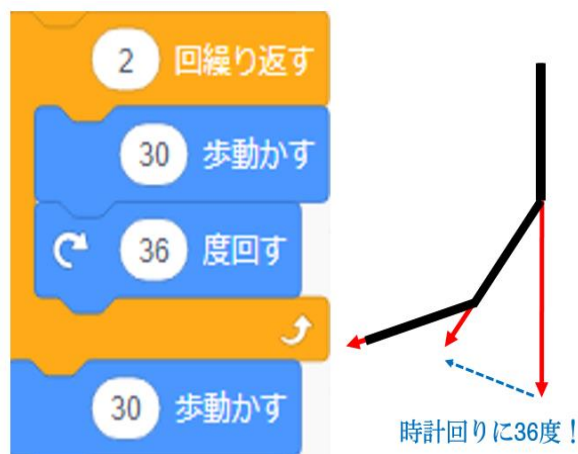
①

③

(a) 屈折回数の少ない桜の花模様を描くプログラム

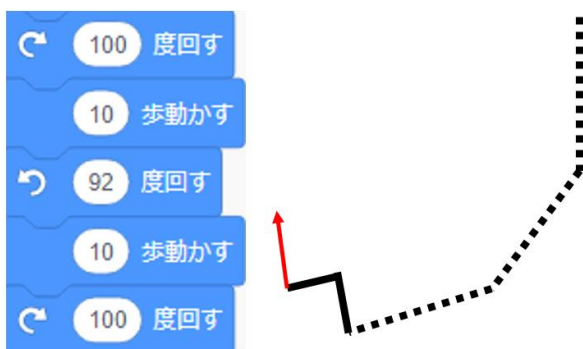
図 10-2 桜の花模様を描く配布したプログラム

図 10-2 は児童が使う端末に配布しておいた、屈折回数の少ない桜の花模様を描くプログラムである。①のプログラムで1枚の花びらを描くことが出来る。①は大きく3つの動きで花びらを描いている(図8-3)。まず、図 10-3(a)について説明する。繰り返しブロック内の30歩は1つの辺を描くことを表し、1辺描いた後に36度時計回りに回転する。それを2回繰り返した後の30歩は3本目の辺を描くことを表している。



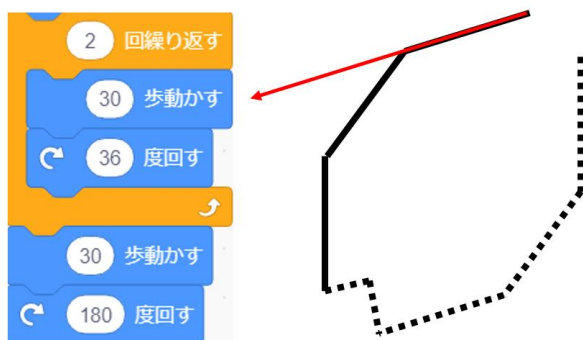
(a) 花びらの切片を描くプログラム

次に図 10-3(b)について説明する。(a)で描いた際に向いていた向きから100度時計回りに回転し、10歩動く。そこから反時計回りに92度回転し10歩動く。最後に100度時計回りに回転すると、赤矢印の向きを向いて次の(c)を描く処理に移る。図 10-3(c)についての説明をする。これは図 10-3(a)で行った処理と同様に3本の辺を作り、1枚の花びらを描き終える。最後に180度回転することで赤矢印の向きを向いて、次の2枚目、3枚目…5枚目を描くことが出来る。



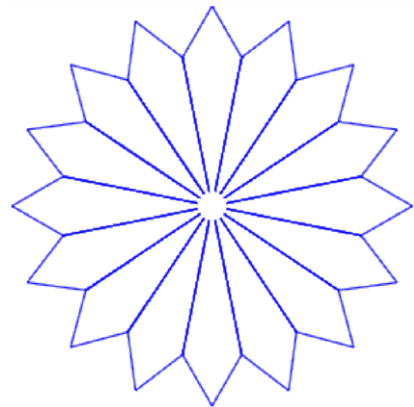
(b) 花びらの先端を描くプログラム

①の処理を②の繰り返しブロックで5回行うことで、5枚の花びらを付けた桜の花模様を描くことが出来る。最後に③のブロックにより、猫を画面左(-90度)へ向かせ1歩動かし5度を繰り返す処理を行う。これは、 $5 \text{度} \times 72 \text{回} = 360 \text{度}$ を意味しており花模様の中心にある円を描くためのブロックになっている。



(c) 花びらの切片を描くプログラム

図 10-3 花びら1枚を描くプログラム

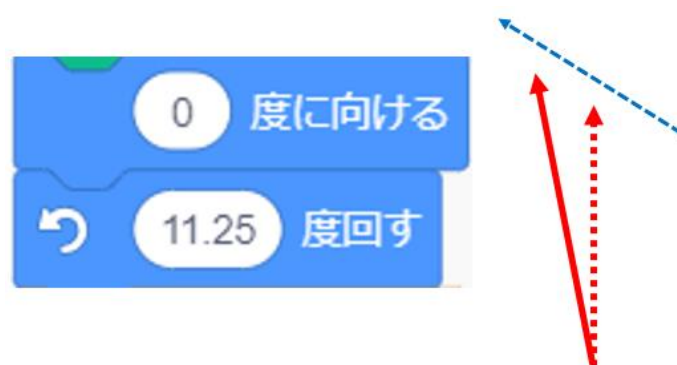


(b) (a)のプログラムで描かれる菊

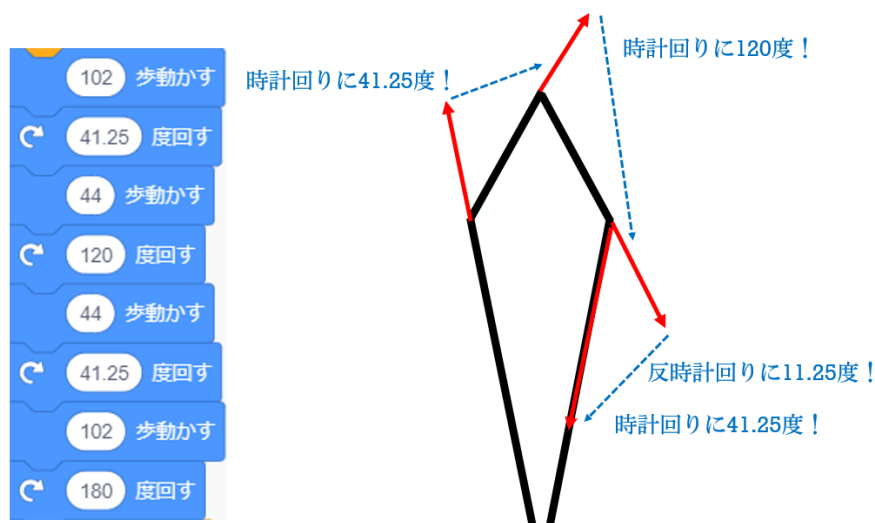
(a) 屈性回数の少ない菊の花模様を描くプログラム

図 10-4 菊の花模様を描く配布したプログラム

図 10-4 は児童が使う端末に配布しておいた、屈折回数の少ない菊の花模様を描くプログラムである。図 10-5 を参考にしながら説明する。まず、図 10-5(a)のブロックによって、画面上(0 度)の向きから反時計回りに 11.25 度回転する。花びらは全部で 16 枚あり、1 枚目の花びらは中心から見て真上にあるため、これを描くために角度を決めておく。次に図 10-5(b)のブロックによって 1 枚の花びらを描いていく。まず、120 歩動かすことで長い辺を作る。そして 41.25 度時計回りに回転し、44 歩動いて短い辺を作る。そこから 120 度時計回りに回転し 44 歩動くことで、先端が描かれる。さらに 41.25 度回転し 102 歩動くことで花びらが 1 枚描き終わる。桜のプログラムと同様に 180 度回転することで、次の 2 枚目、3 枚目…16 枚目を描く準備をしておく。以上の花びらを 1 枚描く動きを図 10-4③の繰り返しブロックで 16 回行うことで、16 枚の花びらを付けた菊の花模様を描くことが出来る。



(a) 花びらを描くための向きへ変えるプログラム



(b) 菊の花びらを 1 枚描くプログラム

図 10-5 花びらを 1 枚描くプログラム

10.4 授業の流れ

授業は次の(1)～(6)の手順で進めた。なお、多くの児童が Scratch 初体験であった。

(1) 授業で行うことの説明

1)本授業において、子ども達が取り組むことをについて伝え説明した(図 10-6)。授業前半では桜の花模様を描き、後半では菊の花模様を描くことを告げた。

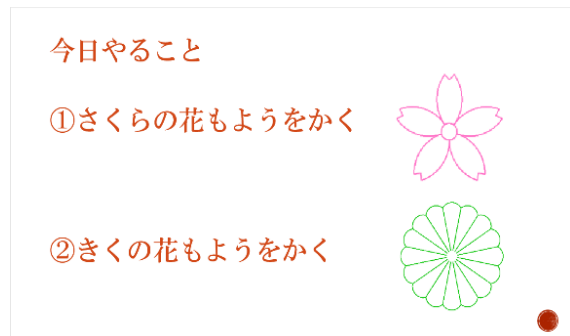


図 10-6 取り組むことの説明

2)花模様の描き方について、どのような考え方で描いていけば効率よく描けるのか説明した(図 10-7)。まず、1枚の花びらを描くためのプログラムを作る。そして、それを花びらの枚数分繰り返すことによって、花模様を描くことが出来る。まず配布したプログラムについての説明を行ってから、課題に取り組んでもらうことを伝えた。

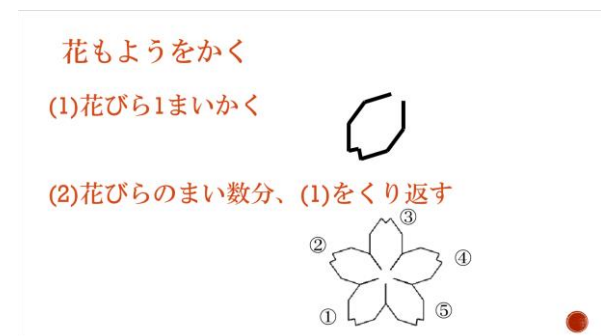


図 10-7 花模様の描き方

(2) 配布した桜の花びらを描くプログラムの説明

1)画面と猫の向きの説明をした(図 10-8)。Scratch の画面は、画面左が-90 度に設定されており、そこから時計回りの向きへ角度が決められている。よって、画面上が0度・右が90度・下が180度であることを確認した。「180 度に向ける」ブロックによって、画面右を向いていた猫が、画面下を向くことを、アニメーションで動きを見せ説明した。

180 度の向きにした理由として、花びらは左下のものを1枚目として描いていくためであることを伝え、猫が下を向いた後に花びらを1枚描いて示した。



図 10-8 画面と猫の向き説明

2)桜の花びらを1枚描くためのプログラムの説明をする(図 10-9)。まず児童に配布したプログラムの、花びらを1枚描く処理を行う部分のみを示す。次に、桜の花びら1枚を描く際は①片側の辺②花びらの先端③もう片側の辺を描く3つのブロック塊に分けられることを伝え、それぞれに該当するようにブロックを3つに分けて示す。次の過程で、3つのブロックがそれぞれどのような役割をもっているのかを説明をしていく。

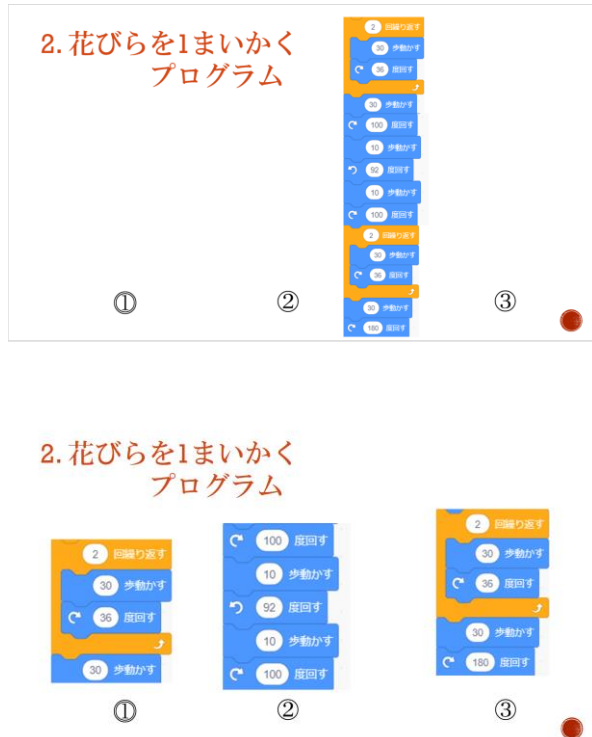
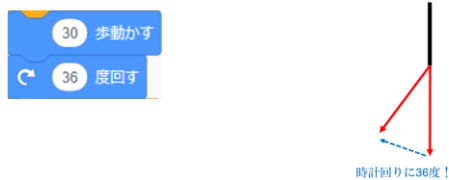


図 10-9 花びら1枚を描くための考え方

3)図 10-9 で分けた3つのプログラムブロックの処理内容を説明する。説明している箇所のブロックを表示しながら、花びらの辺や猫の向きの変化を1つひとつ矢印や実線、点線で示しながら説明した。

3. 花びらを1まいかく①



(a)

3. 花びらを1まいかく②



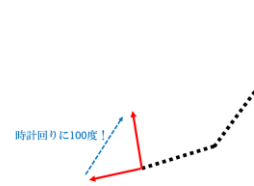
(b)

3. 花びらを1まいかく①



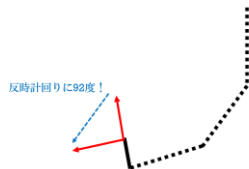
(c)

3. 花びらを1まいかく②



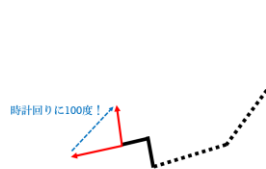
(d)

3. 花びらを1まいかく②



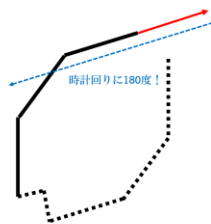
(e)

3. 花びらを1まいかく②



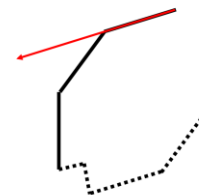
(f)

3. 花びらを1まいかく③



(g)

3. 花びらを1まいかく③



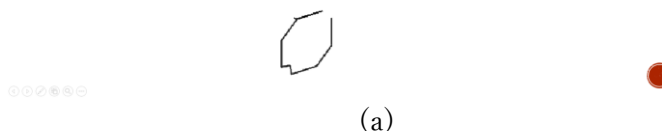
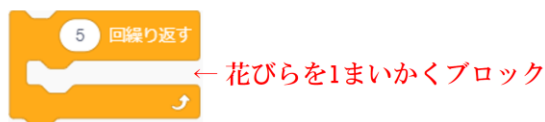
(h)

図 10-10 花びらを 1 枚描く過程のアニメーション

4) 図 10-11 の資料を使って繰り返しブロックの意味を説明した。今回は 1 枚の花びらがを描く処理を 1 単位としているので、繰り返しブロックの中に前項で説明したブロック塊が入ることを伝える。

また、繰り返しになるが、アニメーションでも 1 枚の花びらが繰り返し処理によって 5 枚描かれ花模様が完成することも示す。

3. 花びらのまい数分、くり返す



3. 花びらのまい数分、くり返す

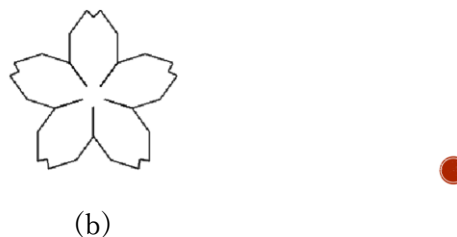
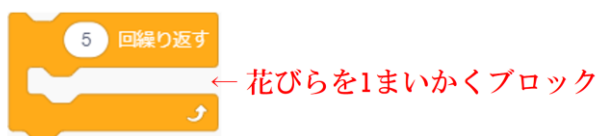


図 10-11 繰り返しブロックの意味の説明

(3) 課題 1(図 10-1(a2), (a3)を描くプログラムの作成) に取り組む

1) 図 10-12 のスライドを見せ、配布したプログラムが描く桜(左)と課題で描く桜(中央・右)のそれぞれで異なる部分について、気づいた子どもに答えさせた。反応として「右に行くにつれ、滑らかになっている」という旨の発言を数人から得た。

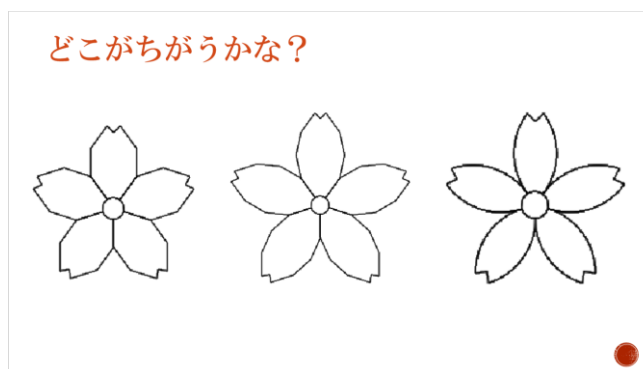


図 10-12 3 つの桜の花模様の違い

2)前項で分かった桜の花模様の違いはどのように生じるのかを、ヒントとして提示した(図 10-13)。

桜の花びらの場合、配付したプログラムは 1 つのカーブで 36 度回転を 2 回行うことで 72 度回転させていた。桜の課題では角度を変える回数を増やすことでより滑らかなカーブを作り出す動作をプログラムし改変していく。ヒントとして①カーブを描く途中で角度を変える回数を変えても、72 度回することは共通であること②角度を変えているポイントを丸で囲み、何回に増えているか気づかせた。角度を変える回数が分かったところで、1 回ごとに何度変えるのかを考えることに目を向けるよう？

度×?回=72度と書き表してある。72 度×3 回の計算式をプログラムで表現すれば良いのだと分かるように働きかけた。

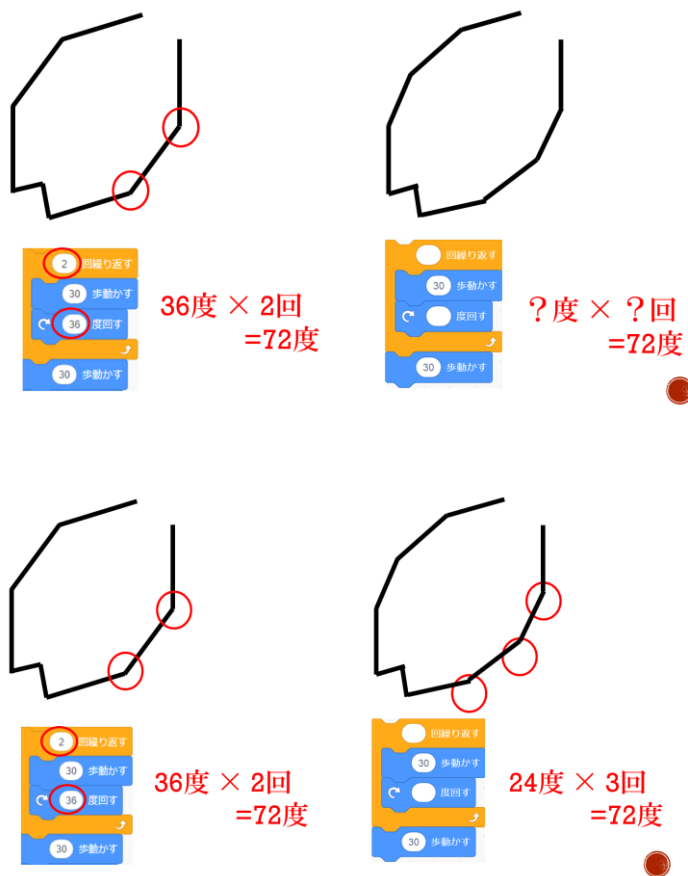
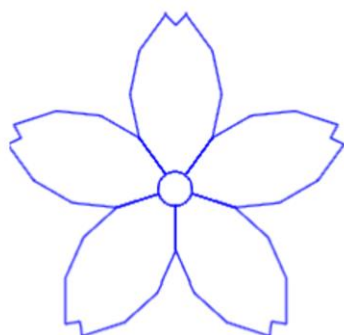


図 10-13 桜課題ヒント

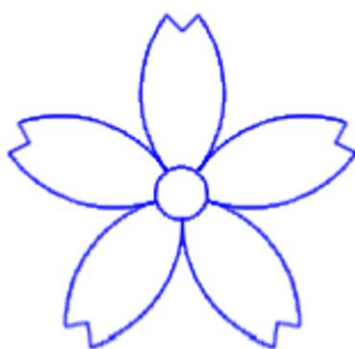
3)桜の花模様を描く問題の解答を発表した。児童に配ったプログラムの該当部分を表示し、 $(\text{回転する角度}) \times (\text{回数}) = 72$ となるように与えたヒントを確認してから、繰り返しブロックごと赤枠で囲い発表した。

さくらの花もよりの答え



(a)桜の課題1つ目解答

さくらの花もよりの答え



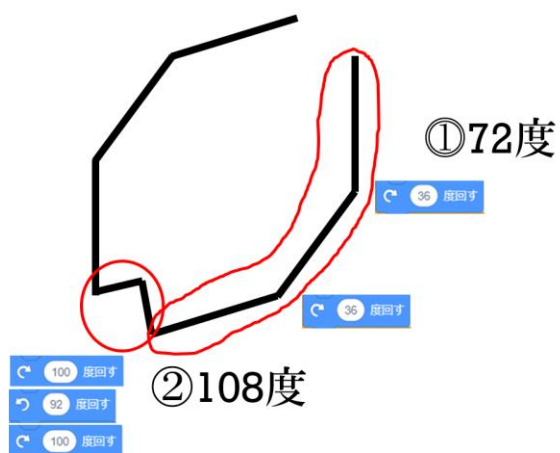
(b)桜の課題2つ目解答

図 10-14 桜の課題解答

4)花びら 1 枚を描き終えるまでに何度回転したのかを考えるために、アニメーションで、1 つひとつの回転ブロックを表示していき、回転ブロックに書かれた角度(図 10-15)に合わせて猫の向きが元々向いていた向きからどのように変わっていくのかを赤矢印で遷移を表し説明した(図 10-16)。

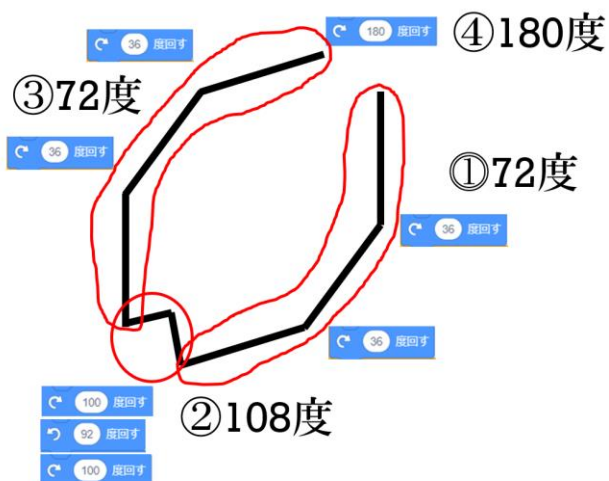
桜の場合花びらは 5 枚描くため、1 枚ごとに $360 \text{ 度} \div 5 = 72 \text{ 度}$ の角度を確保しなければ均等なバランスで描くことが出来ない。そのため、桜の花びら 1 枚を描くプログラム内の回転ブロックの角度全てを合計すると、元々向いていた画面下(180 度)から時計回りに 72 度回転するように予め計算し、決められたものであることを説明した。

1まいかきおわるまでに、何度回ったかな？



(c) 1枚描き終わるまでの回転角度の説明 3

1まいかきおわるまでに、何度回ったかな？

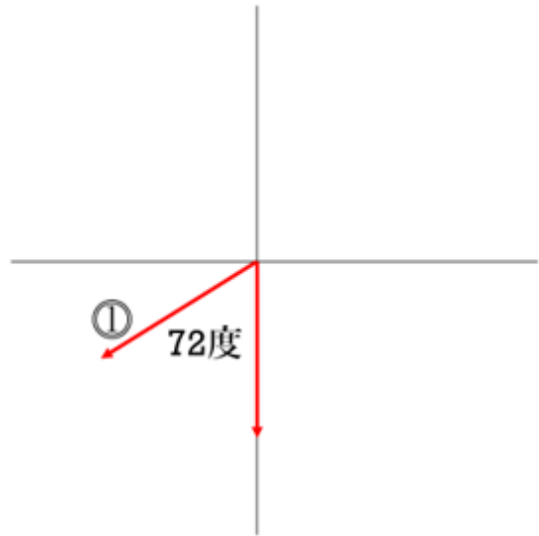


(d) 1枚描き終わるまでの回転角度の説明 4

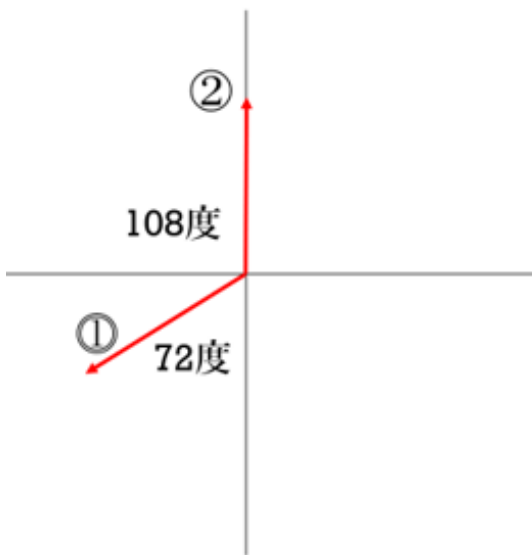
図 10-15 桜の花びら 1枚を描くプログラムの回転ブロック



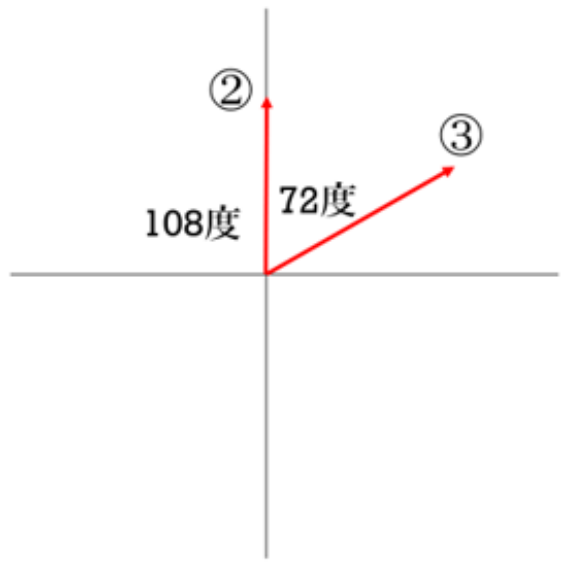
(a)



(b)



(c)



(d)

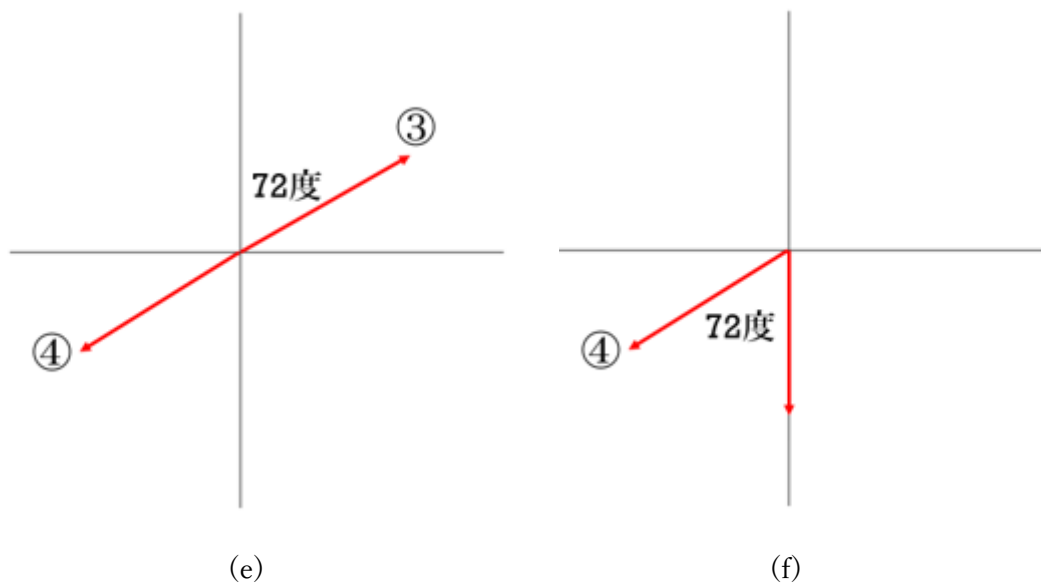


図 10-16 向きの遷移を表す赤矢印

(4) 配布した菊の花びらを描くプログラムの説明

1) 桜の花模様引き続き、菊の花模様を描いていく。

桜の花模様は左下の花びらから描いたが、菊の花の場合は鉛直上方向の 1 枚から描いていく。よってスライドで、どの花びらから描いていくのかを赤枠で囲い示した(図 10-17)。

また、鉛直上方向へ描くために回転ブロックによって画面上(0 度)の向きに変え、そこから反時計回りに 11.25 度回転させておく(図 10-18)。

きくの花もようをかこう

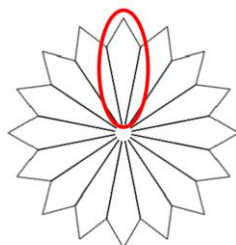


図 10-17 菊の花もようの 1 枚目

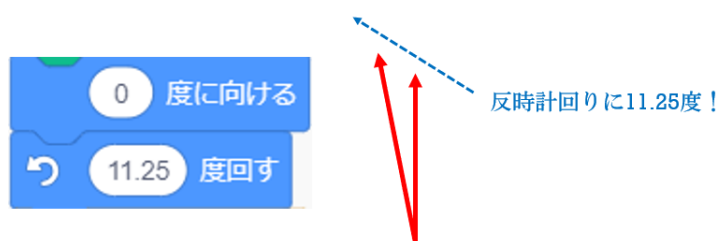


図 10-18 1 枚目を描き始める向きを決める回転ブロック

2) 菊の花びらを 1 枚描く過程を、アニメーションを用いて説明した。配ったプログラムの該当箇所を示した後に、1 つひとつの回転と線を引く動きを交互に見せていった(図 10-19)。

きくの花もようをかこう

時計回りに41.25度!

102	歩動かす
41.25	度回す
44	歩動かす
120	度回す
44	歩動かす
41.25	度回す
102	歩動かす
180	度回す

(a)

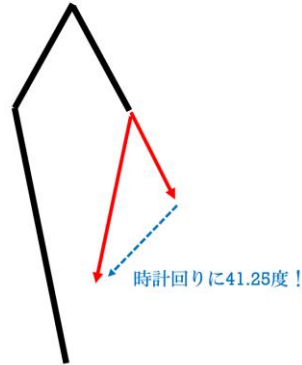
きくの花もようをかこう

時計回りに120度!

102	歩動かす
41.25	度回す
44	歩動かす
120	度回す
44	歩動かす
41.25	度回す
102	歩動かす
180	度回す

(b)

きくの花もようをかこう



(c)

きくの花もようをかこう



(d)

図 10-19 花びらを1枚描く過程のアニメーション

(5) 課題 2(図 10-1(b2), (b3)を描くプログラムの作成)

1) 図 10-20 のスライドを見せ、配布したプログラムが描く菊(左)と課題で描く菊(中央・右)のそれぞれで異なる部分について、気づいた子どもに答えさせた。1 度桜の花模様を描く課題で滑らかなカーブを描いているため、角度を変える回数を変えること

どこがちがうかな？

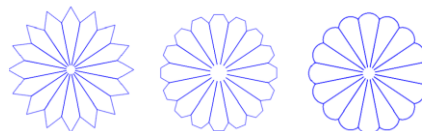


図 10-20 3つの菊の花模様の違い

とカーブが滑らかになっていることを結びつけて覚えられている子どもの反応も見られた。

2)課題で描く菊の花模様の違いはどのように生じるのかを、ヒントとして提示した(図10-21)。

①カーブを描く途中で角度を変える回数を変えても、120度回することは共通であること②角度を変えているポイントを丸で囲み、何回に増えているか気づかせた。角度を変える回数があったところで、1回ごとに何度変えるのかを考えることに目を向けるよう?度×?回=120度と書き表した。60度×2回の計算式をプログラムで表現すれば良いのだと分かるように、どの部分を書き変えるのかと共に示してから課題に取り組ませた。

菊の花びらの場合、配付したプログラムは1つのカーブで120度回転を1回行うことで120度回転させていた。課題では角度を変える回数を増やすこ

とにより滑らかなカーブを作り出す。菊の課題1つ目(図10-20中央の菊)は2回角度を変えているため、60度×2回=120度となるようにプログラムの数値を書き変える。次に2つ目菊課題は1度×120回=120度にプログラムの中身を書き変える。

桜の課題で繰り返しブロック内の数値を変更することが、この課題のポイントであることに気づいている児童が多かった。

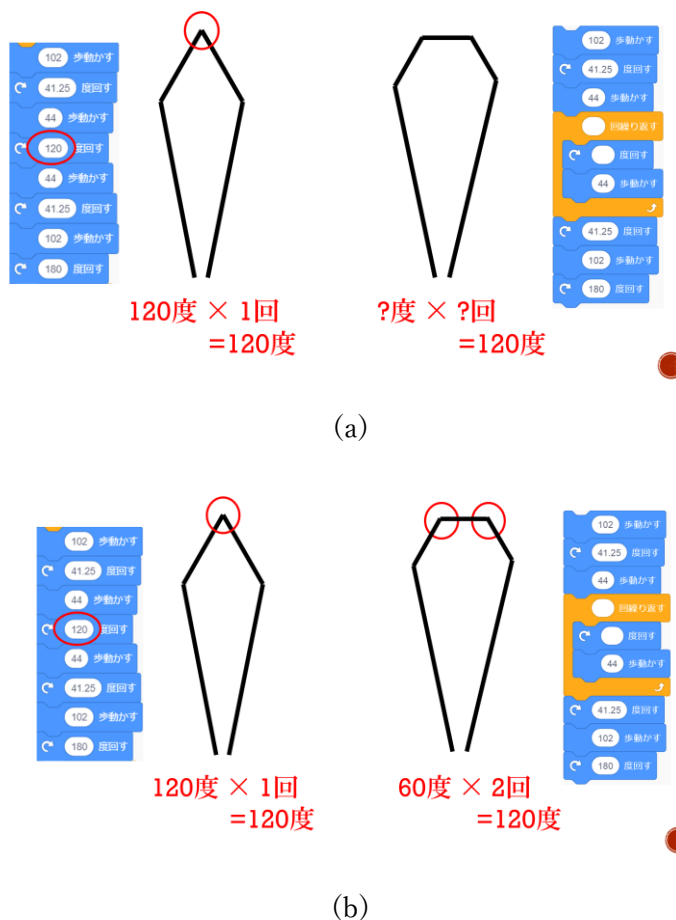
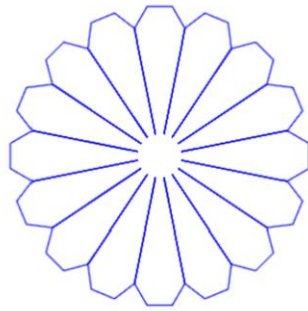


図10-21 菊課題ヒント

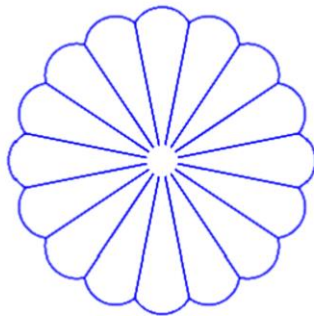
3) 菊の花模様を描く問題の解答を発表した(図 10-22)。児童に配ったプログラムの該当部分を表示し、(回転する角度)×(回数)=120 となるように与えたヒントを確認してから、繰り返しブロックごと赤枠で囲い発表した。

きくの花もよりの答え



(a) 菊の課題 1 つ目解答

きくの花もよりの答え



(b) 菊の課題 2 つ目解答

図 10-22 菊課題解答

4)花びら 1 枚を描き終えるまでに何度回転したのかを考えるために、アニメーションで、1 つひとつの回転ブロックを表示していき、回転ブロックに書かれた角度(図 10-23)に合わせて猫の向きが元々向いていた向きからどのように変わっていくのかを赤矢印で遷移を表し説明した。

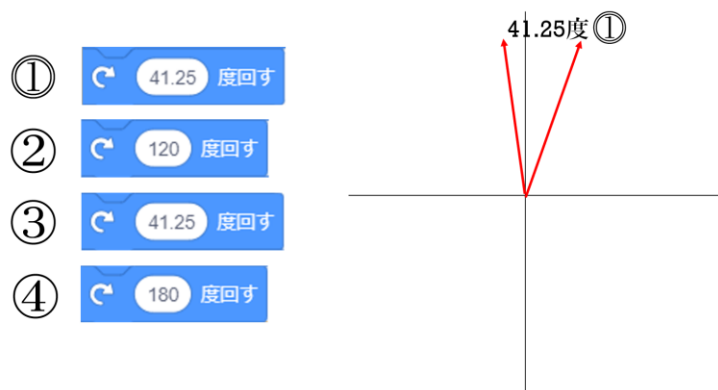
菊の場合花びらは 16 枚描くため、1 枚ごとに $360 \text{ 度} \div 16 = 22.5 \text{ 度}$ の角度を確保しなければ均等なバランスで描くことが出来ない。そのため、菊の花びら 1 枚を描くプログラム内の回転ブロックの角度全てを合計すると、元々向いていた画面左斜め上(-11.25 度)から時計回りに 22.5 度回転するように予め計算し、決められたものであることを説明した。

1まいかきおわるまでに、何度回ったかな？



(a)

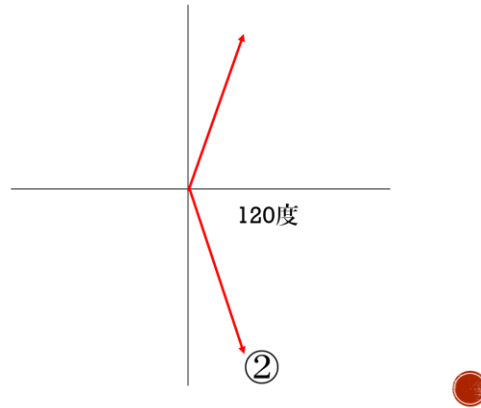
1まいかきおわるまでに、何度回ったかな？



(b)

1まいかきおわるまでに、何度回ったかな？

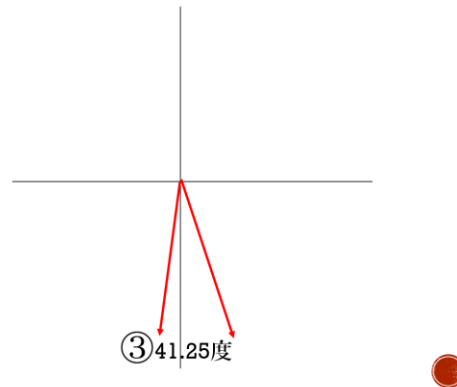
- ① 41.25 度回す
- ② 120 度回す
- ③ 41.25 度回す
- ④ 180 度回す



(c)

1まいかきおわるまでに、何度回ったかな？

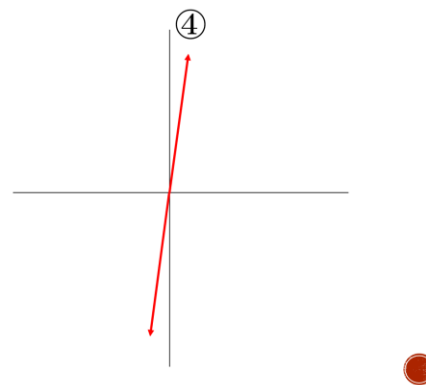
- ① 41.25 度回す
- ② 120 度回す
- ③ 41.25 度回す
- ④ 180 度回す



(d)

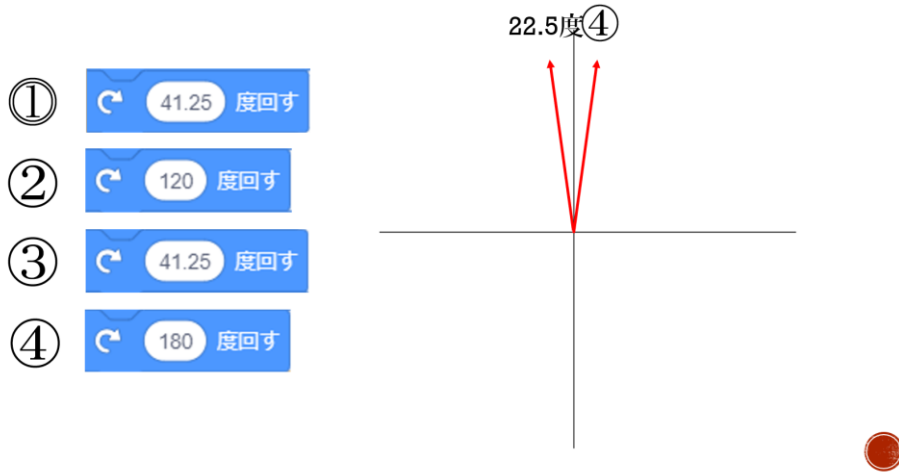
1まいかきおわるまでに、何度回ったかな？

- ① 41.25 度回す
- ② 120 度回す
- ③ 41.25 度回す
- ④ 180 度回す



(e)

1まいかきおわるまでに、何度回ったかな？



(f)

図 10-23 桜の花びら 1 枚を描くプログラムの回転ブロックと遷移の赤矢印

(6) クロージング
授業の最後に Scratch で授業者が描いた星やその他の幾何学模様を子ども達に紹介した。どれも本授業と同様にペン機能を用いて、繰り返しブロックや回転ブロックを組み合わせで予め設計した通りにプログラムしたものであり、正確で綺麗な模様を描くことも出来る Scratch を活用することの楽しさを伝え授業を締めくくった。

最後に

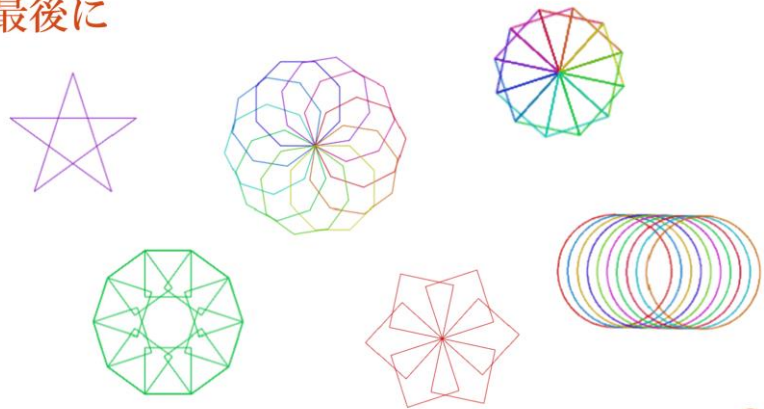


図 10-24 Scratch で描いた模様の紹介

10.5 授業で工夫した点

本授業は初めて Scratch を体験する児童が多かったため、予め桜と菊の花模様を描くプログラムを配付しておいた。このようにすることで、授業の課題の説明や取り組む時間を確保することができた。「〇度×〇回=〇度」のような1回ごとに回転させる角度と繰り返し回数との関係を理解し、それをプログラムで表すということを体験させるには今回の手立ては良かったと考える。

また、中には Scratch を日頃利用している子どもも在籍しており扱いの慣れによって生じる、進捗の早さに違いも出てくる。そのため、桜の花びらと菊の花びらを描く際に使う設計図(図 10-25,26)を配付し、子ども達が各自で課題を進めていくことが出来るようにしておいた。また、授業者の説明で理解できなかった児童に対して TA が説明する際にも、配付した設計図は役に立っていた。

児童同士で分からないことや気づいたことを共有することが出来るように、4,5人で1つのグループを組ませた。効果が見られた例として、課題で回転を何回行えば

①さくらの花びら

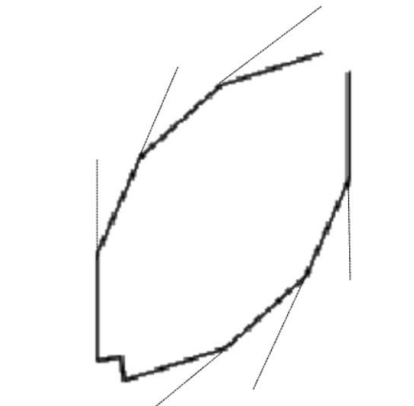
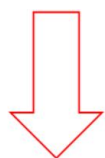
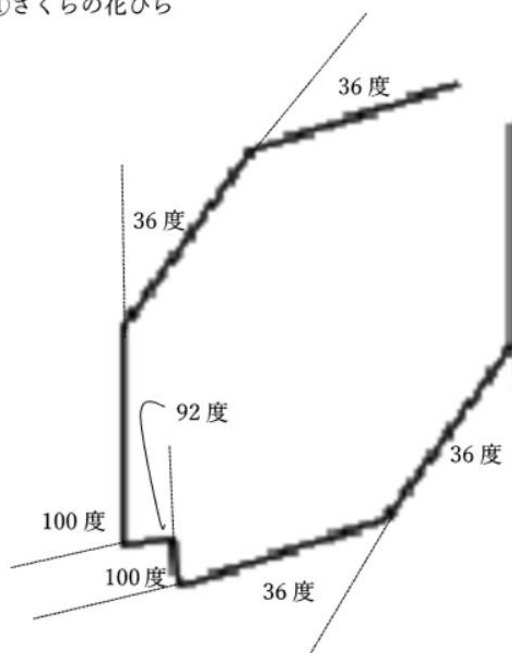


図 10-25 桜の花びら設計図

良いか分かるが、プログラムの扱い方が分からない子どもに対して、別の Scratch の扱いに慣れた子どもが教える姿があった。授業者や TA が直接全員の子どものに教えることが出来ない状況でも、子ども達だけで解決していく場面もあるため授業の内容や活動に応じて授業形態を変えていく工夫も必要である。

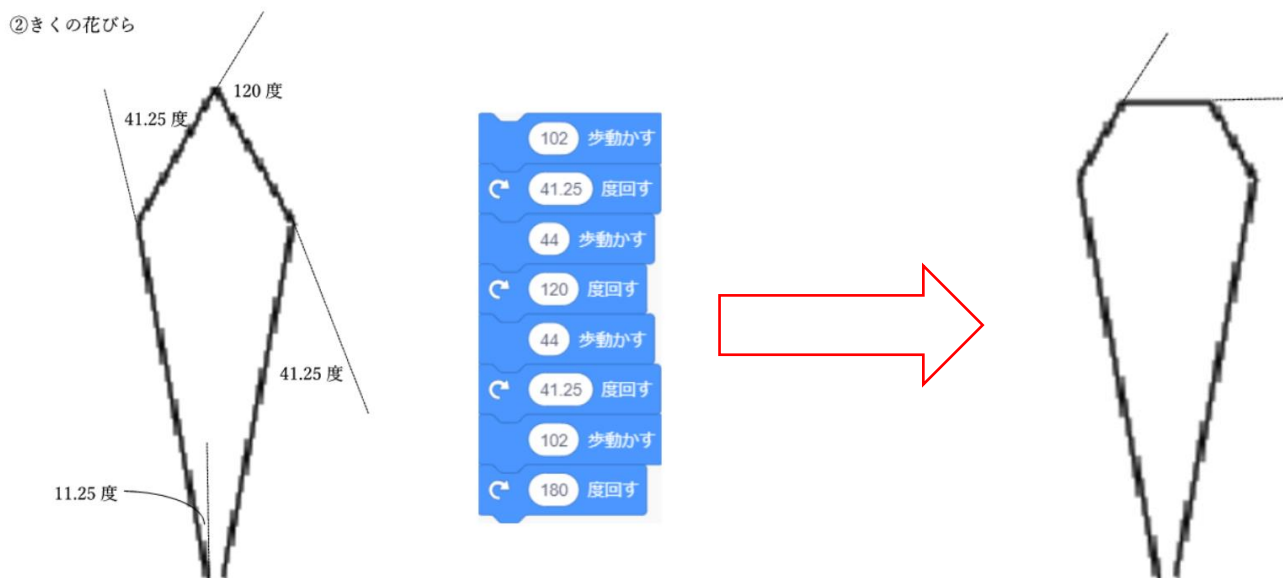


図 10-26 菊の花びら設計図

10.6 TA によるサポート

本実践では、児童のサポート役として授業者を除き 1 日目情報教育 2 年 5 人、3 年 2 人、4 年 6 人の計 13 人、2 日目情報教育 2 年 5 人、4 年 4 人の計 9 人に TA として参加してもらった。対して児童は 32~35 名のクラスであったため、児童 3~4 人に 1 人の TA が付いて課題に取り組んだ。実際に TA が補助した内容は、「児童に予め配っていたプログラムが崩れてしまった際にもとのプログラムへ修復する」「回転ブロック等に数値を打ち込む際の入力補助」「設計図をもとにどのようなプログラムに書き換えれば良いか、一緒に考え気づくために誘導質問をする」等である。中でも、タブレットのキーボードを用いて数値を打ち込む操作や、タッチ操作ミスによるトラブルに TA が対応する機会が多かった。特に授業内で扱うブロックの数値を変える作業は、全員が共通して行うため授業者が説明する機会を設けることで対応件数は減少すると考える。そのために、授業冒頭で簡単な図形を描くチュートリアルを導入する授業改善も視野に入れるべきである。

10.7 授業アンケート結果と考察

授業終了後に4クラス全135名に対して、授業のアンケートを書いて貰い集計を行った。以下に質問・回答項目と回答割合、それらを受けた考察を述べる。

(1) プログラミングはふだんどれくらいやっていますか。

回答の割合は①今回が初めて：66%②これまでに1,2回：22%③時々やったことがある：8%④かなりやったことがある：4%という結果であった。経験者と呼べるほどプログラミングをやっている児童は全体の1割程度で、ほとんどが初めて経験した様子であった。このことから、Scratchでのプログラムの作り方やブロックのもつ意味について説明する必要性が高かったことが推測できる。

(2) 今回の授業の内ようは分かりましたか。

回答の割合は①よく分かった：27%②まあ分かった：48%③あまり分からなかった：19%④全く分からなかった：6%であった。7割以上の児童が内容についての理解は出来ていたようである。しかし、理解できなかった児童については授業中に「分からない」と発言していた子どもの様子からブロックと猫の動きの繋がりを理解できていないようであった。つまり、そもそもブロックやその角度数値の意味を授業内で理解できず、ブロック塊で動く猫の動きも分からなくなってしまっていたのである。この状況を改善するには、子どもにブロックを触らせて、簡単な図形を描く体験をさせることが有効であると考えられる。例えば正三角形のような内角と外角が分かりやすい図形を、回転ブロックの数値を変えて繋いで描くことで、自分の手で理解しながら描くことが出来る。チュートリアルとしての簡単な図形を描く活動を取り入れることで、少しでも多くの児童が今回の内容を理解できるように改善すべきであると感じた。

(3) 今後、家でもプログラミングをしたいと思いませんか。

回答の割合は①そう思う：45%②少しそう思う：38%③あまりそうは思わない：10%④まったくそう思わない：7%であった。(2)の項目で内容に理解を示さなかった児童に、プログラミングに対する興味関心の低さを集計時に感じた。1時間の授業で完結させる中で、多少は児童とのやり取りで理解度を確認しながら進めるよう意識したが、難度の高い課題を2つこなすために一方的な説明が多くなってしまった。本来は前述したように自分でブロックを動かす中で気づきを得ていくことが望ましいだろう。そのような活動が無かったために、プログラミングの楽しさや魅力を伝えることが出来なかった児童が17%程いたと反省している。一方で自由感想欄に、「これまでプログラミングでゲームを作ったことはあったが、図形・模様が描けることは初めて知って面白かった」という旨の感想が数点あった。どの回答者も経験者であったが、教科で学ぶ内容をプログラミングでの活動に関連させることが出来るということが伝えられたようである。初学者だけでなく、経験者にとっても新たな気づきのある授業にもなり得るため、学校でのプログラミングの活動の特徴を今後も意識して授業を行いたい。

(4) 下のプログラムでかける花びらの絵を選んで、点を線でつないで下さい。

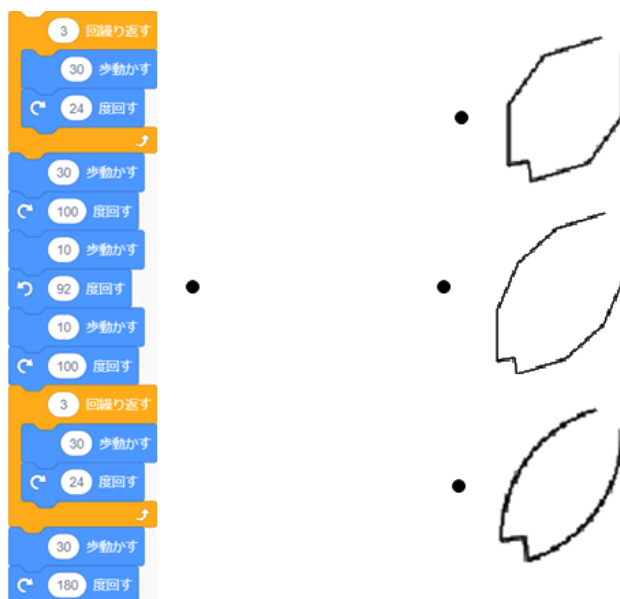


図 10-27 ブロックと描ける絵を問う質問

回答割合は78%の児童が正答していた。桜の花模様を描く課題の1つ目に出題した内容である。次の項目で示すが、与えた課題で4種類の桜・菊の花模様に挑戦させた結果最も描くことが出来ていたものが、この問いの正答である切片の屈折回数が3回の桜の花模様であった。短い活動の中で高い割合で正當に辿り着いたと感じている。誤答で最も多かったのは、一番上の予め児童に配付した花びらを選択した回答であった。恐らく繰り返しブロックの数値が「3」であるために、1つのカーブに3つの辺がある一番上の絵をえらんだのだと推測する。1つひとつのブロックの内容を追っていくと、正答することが出来るため、逐次処理の考え方をより伸ばして行く必要がある。

(5) 授業中にかけた絵を、囲んで下さい。

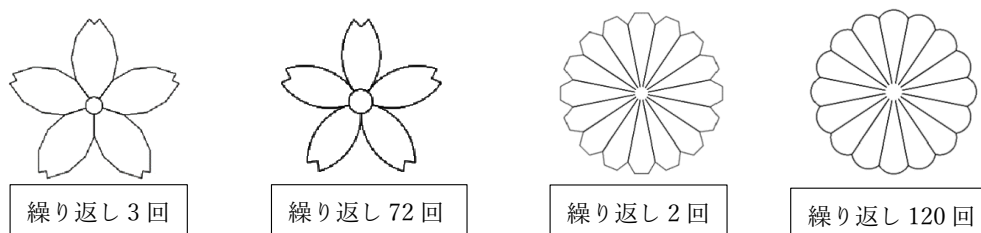


図 10-28 描けた花模様を問う質問

回答割合は桜の花模様「繰り返し 3 回」：76%、「繰り返し 72 回」：27%、菊の花模様「繰り返し 2 回」：42%、「繰り返し 120 回」：11%であった。桜と菊ともに繰り返し回数が増える 2 つ目の課題で正答できた児童が 1 つ目に比べて減っている。これは桜の課題で 2 つ、菊の課題で 2 つと桜菊それぞれ同じ時間内で 2 つの花模様を描かせたことで、時間が不足していたことも影響している。自由感想記入欄にも「もっとプログラミングをする時間があれば描けそうだった」という旨の感想が複数あった。また桜の課題の後に菊を描いたことで、2 つの課題の共通点にも少しずつ気づき始めた様子もあったため、十分に授業時間を確保して行うことでより学びを得られただろう。

(6) さくらの花びらをかくプログラムで、なぜ「〇度回す」の数字になっているのかが理かいましたか。

(7) きくの花びらをかくプログラムで、なぜ「〇度回す」の数字になっているのかが理かいましたか。

図 10-1(a2)は 70%、(a3)は 27%、(b2)は 42%、(b3)は 11%の児童が描くことができていた。(a3)(b3)を描けた児童が少ないのは、実際に取り組んでいた時間が短かったためと思われる。また、(a2)に比べ(b2)を描けた児童の割合が少ない要因の一つとして、(a1)から(a2)は数値の変更で修正できるのに対し、(b1)から

表 10-2 桜プログラムの屈折角度の質問

	36 度回す	100,92,100 度回す	180 度回す
①	37%	16%	48%
②	46%	33%	32%
③	10%	26%	10%
④	5%	21%	7%
未回答	2%	4%	3%

(b2)は「120 度回す」の付近にブロックを追加する必要があることに気付けなかったことが考えられる。(b2)では繰り返しブロック追加は必須ではないが、(b3)では繰り返しブロック追加が必須である。(b2)で繰り返しブロックが追加できていれば、(b3)では数値を返すだけで完成させることができる。

児童に予め配布しておいた桜と菊の花びらを描くプログラムに書かれている屈折角度を授業で説明した。その上で、アンケートでは、桜のプログラム場合「36 度回す」「100,92,100 度回す」「180 度回す」の数値の理解度を、①よく分かった②まあ分かった③あまり分からなかった④まったく分からなかった

表 10-3 菊プログラムの屈折角度の質問

	11.25 度 回す	41.25 度 回す	120 度 回す	180 度 回す
①	24%	25%	38%	51%
②	36%	33%	34%	24%
③	25%	26%	17%	14%
④	11%	12%	9%	7%
未回答	4%	4%	2%	4%

の 4 段階で記入させ、菊も同様に確認した。その結果を、表 1 と表 2 に示す。まず、桜の屈折角度について、「100,92,100 度回す」の質問は 3 つの角度を同時に聞いてしまったことで、児童を困惑させ①②と回答した児童の割合が他の 2 つの質問に比べて低くなったと思われる。また、菊の屈折角度について小数の理解度がやや低いのは、児童にとって小数の角度が慣れていない数値であったためと思われる。小学生は小数点をまだ習っていないため、16 片でなく 20 片の菊を用いると良かったかもしれないが、よく見慣れた 16 片の菊を用いて興味を惹かせた。

10.8 授業で分かった反省点と改善について

他頁でも述べたことであるが、授業実践の対象となった児童はプログラミングや Scratch を扱うことが初めてであった子どもが多い。そのため、扱うブロックの役割やそれによって動くスプライトの動きを理解できていなかった印象をもった。本実践は各クラスに 1 時間 (45 分) の短時間でプログラミングを体験し、逐次処理や繰り返し処理によって花模様を描くことが出来ることを目的としていた。よって、子ども達が実際にブロックを組み合わせ猫の動きを確認しながら扱いに慣れていく過程がなかった。本題材は高学年向けで、低学年より Scratch を通してプログラミングを行っているのであればチュートリアルは必要ないが、2020 年度のプログラミング教育が始まって間もない期間は、授業の初めの時間で児童に簡単なプログラムを作らせ、作ったとおりの動きをスプライトが実行するという場面を見せプログラミングについての感覚を慣らすべきだと感じた。今回の授業であれば、正三角形や正方形等の図形の特徴を確認しそれを踏まえた上で、回転ブロックの角度を設定し繰り返すことでプログラムと実行結果が結びつく感覚を養うべきである。授業者がブロックの紹介・指導し簡単な図形を描く経験をさせることで、その後の内容理解に大きく違いが出てくるだろう。

第 11 章 全体の考察

本研究では小学生向けに授業題材を複数開発し、実施する学年の目安を仮定し割り当てた。パソコン操作能力に関する題材については第 1 学年でマウス操作の練習、第 3 学年でキーボード入力を練習し中学年までにパソコン操作能力の基礎を習得することを目指したい。そのためには、調べ学習や資料作成等のパソコンを使った授業の冒頭や、タブレット端末が普及した時には朝学習の短い時間を使い、パソコンに触れる機会を確保していくことがまずは初段階で目指す姿だろう。また、ローマ字タイピング練習は第 3 学年に位置付けたが、プログラムの中身を変更してひらがなをタイピングで打ち込む練習は低学年から始めても良いだろう。またその時はアルファベットの配置に注目させておくと、キーボードに慣れ親しみ中学年でのローマ字入力の練習がしやすくなる。

次に第 2 学年に「魚釣りゲームプログラミング」、第 3 学年に「迷路ゲームプログラミング」を位置付けた仮説について考察をする。魚釣り題材のゲームでは具体的で簡潔な内容とし、逐次処理・繰り返し・条件分岐の考え方が使われる。また、迷路ゲームでは逐次処理の考え方が使われる。本研究では第 3 学年で「迷路ゲームプログラミング」の授業実践を行い、ほとんどの児童がブロックを繋げて迷路を、時間内に何度もクリアすることが出来ていた。しかし、必要になる考え方が基本的に逐次処理のみであるならば、迷路ゲームを第 2 学年、より多くのプログラミングの要素が考えに組み込まれる魚釣りゲームを第 3 学年の順に割り当てるのが望ましいかもしれない。また、文部科学省が示した「プログラミング教育を通じて目指す育成すべき資質・能力」の枠組みや海外の事例を参考に、ベネッセが 2018 年に提供したプログラミングで育成する資質・能力の評価規準では、低学年で順次処理、中学年で順次処理と繰り返し処理の組み合わせを使って考える力の育成を基準としている。以上のことを踏まえると第 2・3 学年に位置付けた題材の並びを今後再度検討する必要があるだろう。

第 4 学年では、それまであまり経験しなかった複数ある解・方法からより目的に合ったものを選択し実行する価値観を学び感じられるだろう。一筆書きは重複しない書き方であるため、手数が少なくなる利点がある。そのように考えれば、実生活においてルート選択へも適用することも考えられる。前述したベネッセの評価基準において、中学年の思考・判断・表現力等の「ふり返る」能力の観点では、「目的や意図したことに対しての手順を考察し、問題がある場合は理由を考えたり改善方法を考えたりすること」が挙げられている。この第 4 学年の段階でより適切な解に近づけるために、道筋を立てて考え振り返って評価を行うことを学習するのは妥当であると考えられる。

高学年では、それまで Scratch を用いてプログラミング的思考力を培い、普段のパソコンを使った学習でパソコン操作も上達していることが期待出来る。コーディングベースはビジュアルプログラミングよりは抽象的思考が必要となるが、過去に行った授業実践では、1 年間 Scratch で作品作り等の活動を行った子ども達が、JavaScript のプログラムの構

造を理解することが出来ていた。指導の過程でプログラミングに必要な考え方を具体的な例で示すなどの工夫を交えながら、抽象度の高いコーディングベースのプログラミングに挑戦する段階としても第5学年は適当であると考え。Scratch等の小学生向けプログラミングツールはブロックを移動させてプログラミングさせるものが多いが、プログラムは文字ベースで動いているものであることを教えるために、ブロック操作でないプログラミングを経験させた。幸いにもJavaScriptはWebブラウザがあれば動かすことができ、文字ベースのプログラミングツールとして適している。幼児が目の前に用意された積み木で遊ぶ段階から、素の部材を使って遊び出すステップに似て、Scratchのブロックで遊ぶ段階から、次のステップとして、素の部材を使わせることが必要である。その結果、Scratchのブロックの機能には無いことを実現しようとして新たな発想を生むことができる。小学生でもScratchのブロックの機能だけに固執させる必要は無く、触らせられるものは自由に触らせた方がよい。教師側の能力の問題で、Scratchのようなビジュアルプログラミング一択になっていっているが、教師側もJavaScriptのような児童が利用できるツールを、駆使できる能力を身に付けるべきである。本実践では、JavaScriptでも教師側の能力があれば、児童は授業に付いて来れることを示すことができたと思う。次に第5学年で算数の内容を応用し幾何学模様等を描く活動内容を発展させた花模様を描く実践を行った。第5学年で1時間の授業でTA付きの授業で、7,8割の児童が内容理解していたことを考えれば、授業時間数に余裕を持たせかつ子どもがScratchでの図形描画に慣れていれば実施可能であるだろう。しかし、より内容を理解した子どもの割合を増やすために、第5学年で吉原氏の実践のように星や他の幾何学模様の図形描画活動を行って、図形の性質を理解しScratchで図形を描く感覚に慣れさせてから、花模様を描く活動を行うと良いだろう。授業実践の子どもの様子やアンケート結果、これまでの実践から花模様を描く活動は、高学年が適切であったといえる。

開発した題材はプログラムや仕様を変更することで、別の使い方をすることが出来るものが多い。高学年の題材については、発展的な内容であるだけに難易度がやや高めであるため、子ども達に出来ることを把握して実施するのかを決めたり、内容変更して難易度を下げたりして対応しなくてはならない。

第 12 章 まとめ

12.1 本研究の成果

本研究では小学校の各段階で扱うに望ましい題材の内容提案と、プログラミング言語 Scratch と JavaScript を用いた授業実践を紹介した。Scratch はブロックの処理内容が一目で分かり、題材によって活動内容や課題の難易度の変更が可能である。本研究においては、「魚釣りゲーム」のように簡素なものから、「花模様を描こう」のようにこれまでに他研究で実施されてきた図形を描く題材の発展内容まで、難易度の幅を作ることが出来る。この点では、Scratch を用いてプログラミング教育を行うことの柔軟性を示すことが出来た。さらに、児童のパソコン操作能力とプログラミング的思考力の習熟度に合わせて JavaScript による教材で授業展開することで、より抽象的な形でプログラミング的思考力を発揮することが出来る。学習環境の簡易さも踏まえれば Scratch と JavaScript を児童の実態に応じて使い分けていくことが、児童のさらなる成長に繋がると考える。

これまで筆者・研究室単位で参加した授業実戦で分かった子どものパソコン操作の拙さという課題に対して、キーボード操作向上を目的としたタイピング練習ツールの開発も行った。プログラミング教育が必修化され、これから子ども達はパソコンに触れる機会が、従来よりも増えていく。しかし、当然初めてパソコンに触れるという子どもがいる。その状況の中で「パソコン操作を確実に育むということ」「プログラミング的思考力の習得」を目指して活動していかななくてはならない。そのように考えれば、低学年段階でパソコン操作の向上とプログラミングの初歩を学び、中学年段階で簡単なプログラムの作成、高学年で習熟度を見ながら高度なプログラミングへと挑戦していくという一連のビジョンをもっておくことは、闇雲に題材を開発して実践するよりも有意義な教育を可能にする。そして、筆者が学部生時代と本研究で取り組んできた実践題材や他研究の実践内容を、前述したビジョンと子ども達の実態を踏まえた上で当てはめていくことで、より高いプログラミング教育の効果が得られるだろう。またこれを実現するためには、教育現場で活動する教師間でプログラミングに関する知識や考え方を共有し、協力して小学校段階 6 年間を見通したプログラミング教育を児童に施していくことが不可欠である。

12.2 今後の課題

第 11 章全体の考察で、第 2,3 学年のように各学年に割り当てた授業題材で、並びを再検討しなくてはならないものがあることが分かった。題材の中には実践を行うことが出来なかったものがあるため、他実践との比較や資料等から実施するに適切な学年を検討せざるを得なかった題材がある。今後それらを学校現場で児童の実態を十分に把握したうえで、題材を用いて授業を行い割り当ての当てはまりが妥当であるのかを確認していきたい。

Scratch と JavaScript による授業題材の開発を行ってきたが、その多くは総合的な学習の

時間で実践をさせて頂いたものである。プログラミング教育は教科の中においても進められていくべきであるため、各教科の教育内容・活動の範囲で行うことが出来るプログラミング教育の検討も今後は進めていきたい。その方法に関しては、現場教員として準備感じる負担を出来るだけ軽減できるように、プラグド・プログラミングにこだわらずにアンプラグド・プログラミングによるものも視野に入れたい。パソコンを使わないことで興味を惹くことは難しいという意見はあるが、子ども達がプログラミングを意識する必要はないと考える。あくまで教師が順序立てて考える思考法を意識して活動を考えることが重要であるため、パソコンを使うことが最重要ではないからである。各教科の教育内容の本質的な魅力を伝えること、プログラミング的思考力の育成という目的達成のためにプラグドかアンプラグドによる授業展開を選択していくべきである。このような考えをもって、広い視野で各教科内容に基づいたプログラミング教育も推し進めていきたい。

教材開発と授業実践を行う中で、児童の学習活動に関する課題点にも気が付いた。プログラミングの初学者であれば、Scratch を扱う上でどのように慣れるまでの支援を行うのか。初学者と経験者が一緒に学習する場面では、どのような指導が効果を上げられるのか。本研究においては、それらの課題について学習支援の提案と検証ができなかったため、クラス単位ではなく児童個人単位にも着目し、学習上の課題と支援のためのツールや指導法、学習形態等の観点から研究を行いたい。

参考文献

- [1]文部科学省：小学校段階におけるプログラミング教育の在り方について（議論の取りまとめ、
https://www.mext.go.jp/b_menu/shingi/chukyo/chukyo3/053/siryu/_icsFiles/afieldfile/2016/07/08/1373901_12.pdf,(2021年2月10日現在).
- [2]阿部和広：小学生から始めるわくわくプログラミング,日経,BP社,2013.
- [3]倉本大資,阿部和広：小学生から始めるわくわくプログラミング2,日経,BP社,2016.
- [4]杉浦学：Scratchではじめよう！プログラミング入門,日経,BP社,2015.
- [5]Scratch サイト, <https://scratch.mit.edu/studios/1168062/>,(2021年2月10日現在).
- [6]倉知孝拓,山守一徳,岡本光弘：JavaScriptを用いた初等プログラミング教育の授業実践,2018年電子情報通信学会総合大会,D-15-4,2018.03.20.
- [7]文部科学省：子どもの発達段階ごとの特徴と重視すべき課題,
https://www.mext.go.jp/b_menu/shingi/chousa/shotou/053/gaiyou/attach/1283165.htm,
(2021年2月10日現在).
- [8]文部科学省：「GIGAスクール構想の実現」に関する補助事業の概要について,
https://www.mext.go.jp/content/20200219-mxt_syoto01-000003278_505.pdf,(2021年2月10日現在).
- [9]森秀樹,杉澤学,張海,前迫孝憲：Scratchを用いた小学校プログラミング教授業の実践～小学生を対象としたプログラミング教育の再考～,
『日本教育工学会論文誌34』,2011年34巻4号,p387-394,
https://www.jstage.jst.go.jp/article/jjet/34/4/34_KJ00007142887/_article/-char/ja/,(2021年2月10日現在).
- [10]山守一徳,吉原健人,大原敦子：Scratchを用いたお絵描きの授業実践,
情報処理学会第78回全国大会,1F-04,4-479,480,2016.3.10,
https://ipsj.ixsq.nii.ac.jp/ej/?action=pages_view_main&active_action=repository_view_main_item_detail&item_id=163107&item_no=1&page_id=13&block_id=8
(2021年2月10日現在).
- [11]稲垣諒,山守一徳：Scratch コスチューム画像のサイズ均一化ツールの開発,
情報処理学会第79回全国大会,1ZC-05,2017.3.16,
https://ipsj.ixsq.nii.ac.jp/ej/index.php?active_action=repository_view_main_item_detail&page_id=13&block_id=8&item_id=181714&item_no=1(2021年2月10日現在).
- [12]吉原健人,山守一徳,葛原孝紀：Scratchを用いた図形理解を深める授業実践,
情報処理学会第79回全国大会,1ZC-04,2017.3.16,
https://ipsj.ixsq.nii.ac.jp/ej/?action=pages_view_main&active_action=repository_view_main_item_detail&item_id=181713&item_no=1&page_id=13&block_id=8

(2021年2月10日現在).

[13] 田中良研,伊達寛幸,中田充: 小学校2学年におけるプログラミング教育の実践:
スクラッチ Jr を活用したアニメーション物語の作成,
山口大学教育学部附属教育実践総合センター,『教育実践総合センター研究紀要』,46
巻,p157-166,2018.09.30,
<http://petit.lib.yamaguchi-u.ac.jp/G0000006y2j2/metadata/B040046000017>

(2021年2月10日現在).

[14] 岡崎善弘,大角茂之,倉住友恵,三島知剛,阿部和広: プログラミングの体験形式がプログラ
ミング学習の動機づけに与える効果,『日本教育工学会論文誌』,2017年41巻2号
p169-175,
https://www.jstage.jst.go.jp/article/jjet/advpub/0/advpub_41047/_article/-char/ja/,(2021
年2月10日現在).

[15] 山本芳人, 広瀬啓雄: プログラミング言語を習得するための JavaScript を使用した
WBT教材の開発と実践,『日本教育工学会論文誌』,2008年,31巻,Suppl.号,p13-16,
https://www.jstage.jst.go.jp/article/jjet/31/Suppl./31_KJ00004964336/_article/-char/ja/,
(2021年2月10日現在).

[16]山守一徳: JavaScript を用いた中学生向けプログラミング教育,『三重大学教育学部研究
紀要』,三重大学教育学部研究紀要69巻,pp.23-30,2018.01.04,
[https://mie-](https://mie-u.repo.nii.ac.jp/?action=pages_view_main&active_action=repository_view_main_item_detail&item_id=11808&item_no=1&page_id=13&block_id=21)
[u.repo.nii.ac.jp/?action=pages_view_main&active_action=repository_view_main_item_deta](https://mie-u.repo.nii.ac.jp/?action=pages_view_main&active_action=repository_view_main_item_detail&item_id=11808&item_no=1&page_id=13&block_id=21)
[il&item_id=11808&item_no=1&page_id=13&block_id=21](https://mie-u.repo.nii.ac.jp/?action=pages_view_main&active_action=repository_view_main_item_detail&item_id=11808&item_no=1&page_id=13&block_id=21),(2021年2月10日現在).

[17]文部科学省: コンピュータを使用した情報活用能力を測定する初めての調査,
[https://www.mext.go.jp/component/a_menu/education/detail/_icsFiles/afieldfile/2015/0](https://www.mext.go.jp/component/a_menu/education/detail/_icsFiles/fieldfile/2015/03/24/1356195_1.pdf)
[3/24/1356195_1.pdf](https://www.mext.go.jp/component/a_menu/education/detail/_icsFiles/fieldfile/2015/03/24/1356195_1.pdf),(2021年2月10日現在).

[18] 倉知孝拓,山守一徳,三谷哲也: Scratch で花模様を描くプログラミング授業実践
情報処理学会第82回全国大会,5ZH-04,2020.3.6,
[https://ci.nii.ac.jp/lognavi?name=ipsj&lang=ja&id=http%3A%2F%2Fid.nii.ac.jp%2F1001](https://ci.nii.ac.jp/lognavi?name=ipsj&lang=ja&id=http%3A%2F%2Fid.nii.ac.jp%2F1001%2F00205922%2F&naid=170000183069)
[%2F00205922%2F&naid=170000183069](https://ci.nii.ac.jp/lognavi?name=ipsj&lang=ja&id=http%3A%2F%2Fid.nii.ac.jp%2F1001%2F00205922%2F&naid=170000183069),(2021年2月10日現在).

[19]倉知孝拓,山守一徳: 魚釣りゲームを題材にした Scratch プログラミング授業
情報処理学会第83回全国大会,5ZG-03,2021.3.19.

[20]ベネッセ教育情報サイト:「プログラミングで育成する 資質・能力の評価規準(試行版)」
徹底解説,
<https://benesse.jp/programming/beneprog/2018/07/13/standard/>,(2021年2月10日現在).

[21]文部科学省:「教育の情報化に関する手引」検討案第4章情報教育,
https://www.mext.go.jp/b_menu/shingi/chousa/shotou/056/shiryo/attach/1249662.htm,
(2021年2月10日現在).

謝辞

本研究を進めるにあたり、ご指導、ご助言、ご協力頂いた三重大学教育学部数学教育・情報教育コース情報教育専攻山守一徳教授に深く感謝いたします。また、共にこの修了研究活動に取り組み、協力やアドバイスをして頂いた先生方や山守研究室、教育学研究科同期の院生、難しい内容でもいつも丁寧に教えて下さった先輩方、授業実践に協力して下さった後輩達に心から感謝申し上げます。

また、津田小学校、南島西小学校、西が丘小学校、府中小学校の先生方に授業実践を行う上で多くのご協力を頂きました。ここに皆様への深い感謝を述べさせていただきます。