

未就学児を対象とした プログラミング教育に関する研究

渡辺 勇士

電気通信大学大学院情報理工学研究科
博士(工学)の学位申請論文

2021年3月25日

未就学児を対象とした プログラミング教育に関する研究

博士論文審査委員会

主査	中山 泰一	教授
委員	岩崎 英哉	教授
委員	小林 聡	教授
委員	寺田 実	准教授
委員	兼宗 進	教授

著作権所有者

渡辺 勇士

2021

Study on Programming Education for Preschoolers

Takeshi Watanabe

Abstract

Programming education is important in our information society. A programming-education curriculum that is consistent from elementary education to higher education has been proposed and is being developed. The start of programming education is becoming earlier and earlier; thus a curriculum that fosters interest of preschoolers through programming experience is required. How preschool children can understand programming and how they can use it freely to express their ideas must be determined.

This study focuses on 5-6 year old preschool children's understanding of programming and their expression through it. I analyzed the types of programs created by kindergarten children through lessons conducted for 1 year to reveal their understanding and expression. As a result, I quantified preschoolers' understanding of programming through the programs they created.

I describe the background of this thesis in Chapter 1 and discuss related work in Chapter 2. In Chapter 3, I describe the implementation environment and method of the programming lessons at the kindergarten where this study was conducted, how I collected the programs created by the children, and introduce an analysis method of the programs.

In Chapters 4, 5, and 6, I describe the children's understanding of programming based on the programming tasks for each lesson. In Chapter 4, I describe the children's understanding of programming to make drawings move. I prepared images that have a clear direction to move. Then I analyzed in which direction children make them move by programming. In Chapter 5, I describe how the children understand the repetition of picture changes from analyzing their programs. In

Chapter 6, I describe their understanding of programming random and rotational movements by analyzing their program.

In Chapter 7, I analyze how the children used the programming they learned to express themselves in the final lesson. In this lesson, children created programs to introduce the operetta they were going to perform in their kindergarten graduation ceremony for their parents.

In Chapter 8, I summarize this thesis. The main contribution of this study is that preschoolers' understanding of and expression using programming were quantitatively determined. Finally, I propose a programming education curriculum for preschoolers.

未就学児を対象とした プログラミング教育に関する研究

渡辺 勇士

和文要旨

これからの情報社会を生きる人々にとって、プログラミング教育は重要とされている。未来の産業競争力の育成という観点からも、また、空気のようになりつつある情報社会において、市民がその制度や、情報倫理に関する見識を持つために、教養の面からも全ての人が学ぶべきものになっている。

すでに初等中等教育において、プログラミングに関する学習が必修となっている。また、未就学児においても、情報機器に触れる機会が大いにある。そのため、就学前に自分でコンピュータにプログラミングをする体験をし、情報の原理への興味関心を醸成し、その特性を知る必要性が指摘されている。また、現在ではタブレット端末が普及し、教育用プログラミングツールが多く開発されている。未就学児の発達段階に合わせた、理想の未就学児のプログラミング教育について、研究することが可能である。

子どもを対象にしたプログラミング教育に関する研究は、1960年代後半から多くある。その中で、未就学児のプログラミングの理解についての研究は、多面的な研究が必要であり、十分ではない。特に今後さらに普及するタブレット端末を用いた、スクリーン上でのプログラミング体験の学びと、その理解については明らかにされる必要がある。

未就学児の学びを考慮すると、未就学児のプログラミング教育は具体的であり、未就学児自らが探求し、発見するものでなければならない。その上で、未就学児がプログラミングを用いて表現するカリキュラムでなければならない。本研究では、この問題意識のもとに「未就学児のプログラミングの理解」と「未就学児のプログラミングを使った表現」を明らかにすることを目的とした。

この目的を達成するために、年長児を対象にした、発達段階を考慮した1年間のカリキュラムを作成した。そして、実際に幼稚園において、年長児に1年間の

レッスンを実施し、園児の作ったプログラムを分析することで、未就学児のプログラミングの理解について分析した。また、カリキュラムに沿った課題の他に、園児に自由にプログラムを作成させる課題を用意し、そこで作られたプログラムを分析することで、未就学児にとってプログラミングがどのような表現のツールになるかを分析した。

第1章では、序論として、我が国における情報社会の広がり、プログラミング教育の現状、そして、未就学児の学びに合わせたプログラミング体験を考察した上で、本研究の目的について述べる。

第2章では関連研究を、子どもを対象にしたプログラミング教育の関連研究と、本研究で用いているビジュアルプログラミング言語 Viscuit（ビスケット）に関する研究に分けて論じ、本研究の立脚点について述べる。

第3章では、本研究の実践を行なった香川富士見丘幼稚園でのレッスン概要・実施環境・実施方法について述べ、園児の作ったプログラムの収集方法・分析方法について述べる。

第4章から第6章では、園児の「プログラミングの理解」について述べる。

第4章では、動きの方向の理解について述べる。園児が意図した方向に絵を動かすプログラムができているかどうかを、予め明確な動きの方向をもった絵を用意し、その絵に対して、園児がどのようなプログラムをするかを分析することで明らかにする。

第5章では、絵の変化の繰り返しの理解について述べる。園児のどれくらいの割合が繰り返し続けるプログラムを正確に作れているか、また、どのように、どれくらいの長さの変化の繰り返しを作っているのかを分析することで、園児の絵の変化の繰り返しのプログラムの理解について明らかにする。

第6章では、ランダムな動き、そして、回転の動きの理解について、予め用意した絵と自分で描いた絵に対して、園児の作ったプログラムを分析することで明らかにする。

そして第7章では、園児の「プログラミングを使った表現」について述べる。1年間のカリキュラムの中の最終レッスンにおいて、園児が学んだプログラミングをどのように活用して、表現をしていたのかについての分析をする。具体的には、園児は卒園式で保護者に対して演ずる予定であったオペレッタを紹介するプログラム作品を作成した。そのプログラムを絵とプログラムの関係を詳細に確認することを通して分析した。

第8章では、本研究の成果をまとめる。本研究の成果は年長児向けに作成した1年間のレッスンカリキュラムを、実際に年長児に実施し、そこで作られたプログラムを量的に分析し、年長児のプログラミングの理解と、プログラミングを用いた表現を示すことで、プログラミングが未就学児にとって表現のツールになることを明らかにしたことである。さらに、分析結果とカリキュラムを照らし合わせ、今後の未就学児を対象にしたプログラミング教育についての提言を行う。

目次

第1章	序論	1
1.1	本研究の背景	1
1.1.1	情報社会の広がりとは未就学児	1
1.1.2	日本におけるプログラミング教育	2
1.1.3	未就学児の学びとプログラミング体験	5
1.2	本研究の目的	8
1.3	研究の概要と本論文の構成	9
第2章	先行研究	11
2.1	はじめに	11
2.2	子どもを対象にしたプログラミング教育に関する研究	12
2.2.1	子どもを対象にしたプログラミングツールの研究	12
2.2.2	子どもは何を学ぶのかについての研究	18
2.2.3	未就学児のプログラミングについての研究	21
2.3	ビジュアルプログラミング言語 Viscuit（ビスケット）に関する研究	23
2.3.1	図形書き換え型言語に関する研究	23
2.3.2	Viscuit におけるプログラム	26
2.3.3	高度な Viscuit のプログラム	30
2.4	先行研究のまとめ	34
第3章	研究対象の概要	38
3.1	はじめに	38
3.2	香川富士見丘幼稚園	38
3.2.1	経緯	38
3.2.2	研究対象と方法	39
3.2.3	教室レイアウトとレッスンの進め方	39
3.3	カリキュラム内容	42
3.4	プログラムの収集と分析	47
3.5	研究対象の概要のまとめ	48
第4章	動きのプログラムにおける方向の理解についての分析	50
4.1	はじめに	50

4.2	レッスン内容と対象	50
4.3	方向に関する分析	51
4.3.1	レッスン1のプログラムの分析	51
4.3.2	レッスン2方向に関する分析	53
4.3.3	レッスン4における方向に関する分析	55
4.4	速度に関する分析	57
4.5	動きのプログラムにおける方向の理解についての分析のまとめ	60
第5章	繰り返し続けるプログラムの理解についての分析	61
5.1	はじめに	61
5.2	レッスンの内容と対象	61
5.3	練習課題の内容と分析	62
5.4	自由課題の内容と分析	66
5.5	繰り返し続けるプログラムの理解についての分析のまとめ	71
第6章	ランダムな動きと回転の動きについての理解の分析	74
6.1	はじめに	74
6.2	ランダムな動きの理解の分析	74
6.2.1	練習課題の内容と分析	75
6.2.2	自由課題の内容と分析	80
6.3	回転の動きの理解の分析	81
6.3.1	練習課題の内容と分析	81
6.3.2	自由課題の内容と分析	87
6.4	“ランダムな動き”と“回転の動き”についての理解の分析のまとめ	90
第7章	プログラムを使った表現の分析	93
7.1	はじめに	93
7.2	クラスA（にゃんきちいっかのだいぼうけん）	94
7.2.1	基本情報	94
7.2.2	直線の動き	96
7.2.3	ランダムな動き	97
7.2.4	絵の変化の繰り返し	98
7.2.5	回転の動き	99
7.2.6	衝突	99
7.2.7	園児のプログラミング表現の数と種類	100
7.3	クラスB（北風と太陽）	101
7.3.1	基本情報	101
7.3.2	直線の動き	104
7.3.3	ランダムな動き	104
7.3.4	回転の動き	107

7.3.5	“直線の動き”と“絵の変化”を組み合わせたプログラム表現	108
7.3.6	園児のプログラミング表現の数と種類	108
7.4	保育日誌	111
7.5	プログラムを使った表現の分析のまとめ	111
第8章	本論文の総括	114
8.1	本論文のまとめ	114
8.2	本研究の理論的な貢献	118
8.3	本研究の制限と課題	120
8.4	結言	121
	謝辞	123
	参考文献	125
	関連論文の印刷公表の方法および時期	134

目 次

1.1	Piaget の発達図式 ([30] を参考にし, 著者が作成)	5
1.2	佐伯の提唱する学びのドーナツ論 ([81] を参考に著者が作成)	7
2.1	Logo における床置き of タートル ([98] より引用)	13
2.2	Squeak Etoys の画面 ([99] から引用)	15
2.3	左: LEGO マインドストームの躯体, 右: ロボットを動かすプログラムを作る画面	15
2.4	キュベットでプログラムする児童の姿	16
2.5	Scratch の画面	16
2.6	ScratchJr の画面	17
2.7	コンピュータシミュレーションプロセス (CTP) ([78] を参考に著者が作成)	20
2.8	BITPICT の画面 ([17] より引用)	24
2.9	ChemTrains の画面 ([5] より引用)	24
2.10	KIDSIM の操作方法 ([2] より引用)	25
2.11	左: 図形書き換え型を採用した AgentSheets, 右: ブロック型の if-then ルールを採用した AgentSheets ([77] より引用)	26
2.12	Viscuit の製作画面	27
2.13	Viscuit における衝突のプログラムの例	29
2.14	上級者モードにおける Viscuit のインタラクションのプログラムの例	30
2.15	Viscuit の設定画面	32
2.16	グリッドモードの Viscuit	32
2.17	左: 猫にコインの画面, 右: 猫にコインのメガネの一部	32
2.18	左: 初期画面・右: 矢が放たれる確率の調整の一部	33
2.19	左: プレイ画面・右: ステージを変えるルンバを表示するメガネ	34
2.20	次のステージを表示するメガネ	34
2.21	具体化された抽象的な概念の CTP	36
2.22	CTP 単位でみた手続き型の言語プログラムと Viscuit のプログラムの違い	37
3.1	ビスケットランドを使った発表会の様子	40
3.2	会場のレイアウト図	41

3.3	園児を前に集めている様子	41
3.4	Viscuit におけるプログラムの例	44
4.1	左 : L1-P1 の画面, 右 : L1-P2 の画面	52
4.2	左 : L2-P1 の画面, 右 : L2-P2 の画面	54
4.3	左 : L4-P1 の画面, 右 : L4-P2 の画面	56
4.4	ヒストグラム / 左 : 横の世界の正答率, 右 : 縦の世界の正答率 . . .	57
4.5	L3-P1 の画面	58
4.6	L3-園児の速さのヒストグラム	59
5.1	L7-P1 における絵と正解のプログラム	63
5.2	L7-P2 における絵と正解のプログラム	63
5.3	L7-P3 における絵と正解のプログラム	63
5.4	L8-P1 における絵と正解のプログラム	64
5.5	L8-P2 における絵と正解のプログラム	64
5.6	L9-P1 における絵と正解のプログラム	64
5.7	L9-P2 における絵と正解のプログラム	65
5.8	L7-LLL3 のプログラム	68
5.9	L8-LLL4 のプログラム (左 : A 児, 右 : B 児)	70
5.10	L9 の自由課題 (左 : クラス A, 右 : クラス B)	70
5.11	L9-LLL5 のプログラム (C 児)	71
6.1	左 : L5-P1 の画面, 右 : L5-P2 の画面	75
6.2	左 : L6-P1 の画面, 右 : L6-P2 の画面	75
6.3	「カニ」っぽい動きの例	77
6.4	Viscuit における回転の機能のボタン	81
6.5	左 : L10-P1 の画面, 右 : L10-P2 の画面	82
6.6	左 : L11-P1 の画面, 右 : L11-P2 の画面	83
6.7	左 : L12-P1 の画面, 右 : L12-P2 の画面	83
6.8	回転の大きさの比較 : 左が回転半径 30 ポイント, 右が 100 ポイント	84
6.9	L10-P1 の大きさと傾き	85
6.10	L10-P2 りんごの回転の大きさと傾き	85
6.11	L10-P2 鉛筆の回転の大きさと傾き	85
6.12	L10-P2 はなの回転の大きさと傾き	86
6.13	L10-P2 星の回転の大きさと傾き	86
6.14	L11-P1 の回転の大きさと傾き	86
6.15	L11-P2 : 2 色の風車の大きさと傾き	87
6.16	L11-P2 : 4 色の風車の大きさと傾き	87
6.17	L11-P2 : 1 色の風車の大きさと傾き	87
6.18	L12-P2 の大きさと傾き	88

6.19	L10-F 回転半径の平均のヒストグラム	89
6.20	L11-F 回転半径の平均のヒストグラム	89
6.21	L12-F 回転半径の平均のヒストグラム	89
7.1	直線の動きの例	97
7.2	ランダムな動きが使われた例	98
7.3	猫の表情の変化の例	99
7.4	衝突を使ったプログラムの例	100
7.5	直線の動きにおける方向のマッチしている例	105
7.6	雲の口が変わるプログラムの例	106
7.7	表情の変化の例	106
7.8	紙芝居の例	107
7.9	回転する風の例	107
7.10	直線の動きと絵の変化で雪が舞っている姿を表している例	108
8.1	L1-12 のそれぞれのレッスンの体験モデル	118
8.2	本研究全体の体験モデル	119

表 目 次

2.1	様々な教育用プログラミングツールの比較	17
2.2	Viscuit とテキストベース言語, ブロックベース言語との違い	28
2.3	ScratchJr と Viscuit の違い	29
2.4	Viscuit のメガネで表現できること	31
2.5	Viscuit を含めた様々な教育用プログラミングツールの比較	36
3.1	レッスンの時間配分	40
3.2	2017 年度に実施したレッスンの内容	42
3.3	練習課題に用意した絵と自由課題のテーマ	43
3.4	カリキュラム内の Viscuit プログラムの多言語との比較	45
3.5	分析のために注目した json ファイルの情報	48
4.1	分析の対象となった園児の数	51
4.2	L1-P1 の分析結果	52
4.3	L1-P2 の分析結果	53
4.4	L2-P1 の分析結果	54
4.5	L2-P2 の分析結果	55
4.6	L4-P1 の分析結果	57
4.7	L4-P2 の分析結果	58
4.8	L4-自由課題の分析結果	59
4.9	L3-P1 の分析結果	59
5.1	L7-9 の練習課題の結果	65
5.2	L7-9 の二項検定の結果	65
5.3	L7-F の LLL 集計	67
5.4	平均メガネ数の分布 (L7-LLL2)	67
5.5	L8-F の LLL 集計	68
5.6	平均メガネ数の分布 (L8-LLL2)	69
5.7	平均メガネ数の分布	69
5.8	クラス A の結果 (L9-F)	70
5.9	クラス B の結果 (L9-F)	71

6.1	L5-6 の練習課題の結果	76
6.2	L5-P1 の練習課題の結果	77
6.3	カニの結果	78
6.4	エビの結果	78
6.5	L6-P1 の練習課題の結果	79
6.6	L6-P2 の練習課題の結果	80
6.7	L5, L6-F の結果	81
6.8	L7-9 の練習課題の結果	84
6.9	L10, L11, L12-F の結果	88
7.1	分析したプログラムの数	95
7.2	使われたテクニックの分類	95
7.3	モチーフとして使われた絵とプログラム	96
7.4	絵と方向	97
7.5	変化のモチーフと部分	98
7.6	回転のモチーフと動き	99
7.7	テクニックの種類と内訳	100
7.8	園児の分布	102
7.9	分析したプログラムの数	102
7.10	使われたテクニックの分類	103
7.11	モチーフとして使われた絵とプログラム	104
7.12	直線の動かされた絵と方向	105
7.13	変化のモチーフと部分	106
7.14	回転のモチーフと動き	107
7.15	Group details	108
7.16	園児の分布	110
8.1	レッスン内で採用した工夫	116

第1章

序論

1.1 本研究の背景

1.1.1 情報社会の広がりとう就学児

今日、我々の周りにはスマートフォンをはじめとする、情報機器が当たりまえのように普及している。2008年に発売されたiPhoneから、わずか10年でタブレット端末は「スマホ」と呼ばれるかたちで、若者から高齢者に至るまで普通のものになった。また一般的に販売される家電にも、様々なソフトウェアが搭載されるようになり、IoTも珍しいものではなくなっている。現代社会は情報社会として完成されつつある。

情報社会の中で暮らす現代人にとって、コンピュータをブラックボックスのままにしておくことは危険である[80][82]。コンピュータがどのような仕組みで動いているのかを知るためには、プログラミングをすることを通して、その性質を知ることが重要である[25][49]。プログラミング教育は重要視されており、すでに小学校、中学校、高校で必修化されている。その中でプログラミング教育のスタートは、就学前から徐々に始まるべきだという主張がある[67][109]。

一方で、未就学児が情報機器に接することに対して、警笛を鳴らす人々もいる。現在普及している端末やアプリケーションには、子どもの射幸性を過度に刺激するものも多い[36]。また、保護者がコンピュータサイエンスの基礎知識をもっていない場合が多く、どのアプリを触らせてよいのか、保護者が判断することが難しい場合もある[65]。このように様々な要因から、未就学児に情報機器に触らせることに対しての不信感が醸成されている。

しかし、未就学児と情報端末の関係において、年齢制限等を設けて未就学児を情報機器に触らせないことは不可能である。未就学児が情報機器に接触することが不可避であるならば、それらをブラックボックスとして享受するのではなく、少なくともそれらのサービスは、人間が作っているプログラムで動いている、という実感を持って触れるべきである。そのためには、未就学児においても、プログ

ラムを作ることを通して、情報に対する直感を育み、興味関心を醸成し、作り手として情報機器・サービスを批評できる心と目を育てるべきである。そして、早期からコンピュータを身近に感じることは、小学校、中学校、高等学校におけるプログラミング教育の準備となる。学校教育を通して、さらに深い知識を身につけ、情報社会に参加する態度を育てられると考えられる。

このような視点に立った時、未就学児における最適なプログラミング教育とは、どういったものなのかを考える必要がある。そして、最適なプログラミング教育をデザインするためには、未就学児がプログラミング体験を通して何を学ぶのか、そして、学んだものを適切に理解し、使うことができるのか、明らかになっている必要がある。

1.1.2 日本におけるプログラミング教育

職業訓練の中の、技能としてのプログラミング教育ではなく、非専門家も含めた「万人向けのプログラミング教育」のスタートは1989年に遡る[48]。1989年に中学校学習指導要領に「技術・家庭科」の「情報基礎」（選択領域）として、プログラミングに関する学習が明記された。これは1987年の教育課程審議会の答申において、情報化の進展に対応した領域の新設が必要である、という意見を反映させたものであった。現在、学校教育におけるプログラミングに関する指導については、中学校「技術・家庭科」においてプログラムによる計測・制御が必修となっている[62]。また、高等学校においては、共通教科「情報」の科目「情報の科学」において、プログラミング言語などにより簡単なアルゴリズムを表現し、自動実行させることなどが示されている[58]。

小学校においては、2016年4月19日の産業競争力会議において、初等中等教育でプログラミング教育を必修化するという方向が示された[100]。この決定後に行われた小学校段階におけるプログラミング教育のあり方についての議論では、「プログラミング的思考」という言葉が用いられ、それを学ぶ必要性が明示されている[59][60]。「プログラミング的思考」とは「自分が意図する一連の活動を実現するために、どのような動きの組合せが必要であり、一つ一つの動きに対応した記号を、どのように組み合わせたらいいのか、記号の組合せをどのように改善していけば、より意図した活動に近づくのか、といったことを論理的に考えていく力」と説明されている。その上で「プログラミング教育とは、子供たちに、コンピュータに意図した処理を行うよう指示することができるという体験をさせながら、将来どのような職業に就くとしても、時代を超えて普遍的に求められる力としての「プログラミング的思考」などを育むことであり、コーディング（プログラミング言語を用いた記述方法）を覚えることが目的ではない」と、示されている。

つまり「プログラミング的思考」とは、自分が意図した目的を達成するために、どのようにコンピュータに命令すればいいか、論理的に考える思考法である。そして、児童はプログラミングを通して、コンピュータで何らかの情報の処理を行

うことで「プログラミング的思考」を身につける。また「プログラミング的思考」とは、思考方法の名称であり、特定の言語の記法を覚えることではない。算数や理科での「プログラミング的思考」の育み方が例示された上で、この思考力は専門の教科を設置するのではなく、各教科で育むことが望ましいとされている。

学校教育外でも、今後訪れる Society 5.0 に備える手段として、プログラミング教育は注目されている [66]。文部科学省だけではなく、総務省、経済産業省にとっても、プログラミング教育は力を注ぐ領域になっている。

総務省は 2017 年より「若年層におけるプログラミング教育推進事業」として事業を行っている [92]。この事業において、2017 年は地域を対象にして、学校教育課程外でのプログラミング教育事業への支援が行われた。2018 年においては、障害を伴う児童を対象に、プログラミング教育の事業を募集し、障害を伴う児童がプログラミングを学ぶ機会を創出した。2019 年以降は、「地域 ICT クラブ普及推進事業（地域における IoT の学び推進事業）」として、地域で子どもたちが、そこに住む住民とモノづくりやデザイン等をテーマに、プログラミング等 ICT 活用スキルを学びあい、世代を超えて知識・経験を共有する機会を提供する事業への補助をしている [93]。

経済産業省は 2018 年より「未来の教室」実証事業とし、プログラミング教育を含む ICT を活用した、様々な新しい学びのあり方を模索する事業へ助成を行っている [42]。また、生徒 1 人 1 人にパソコンと高速ネットワークの支給を計画する GIGA スクール構想と関連する事業も行っている [56]。2020 年度は GIGA スクール構想によって学習インフラが学校において整った後に、そのインフラを使って実施すべき授業・コンテンツを提供する事業者に対して「先端的教育ソフトウェア導入実証事業（EdTech 補助金）」として、学校設置者と EdTech を提供する事業者に対して補助金を出している [43]。このように、小学校教育におけるプログラミング教育の領域では、学校教育の外の様々な団体をも巻き込んで、様々な取り組みがなされている。

小学校からプログラミングに取り組むことを前提に、2022 年、2024 年にそれぞれ改訂される中学校、高等学校の学習指導要領においても、プログラミングの取り扱いが小学校との連携を目的に変わりつつある [38]。また、入試科目として「情報」を位置付けることで、より多くの人に「情報」を学ぶ機会を創出させる動きもある [47]。

幼稚園においては、2017 年に幼稚園教育要領が新しくなった [61]。新しい教育要領解説には、5 歳児修了時までには育ってほしい具体的な姿が「幼児期の終わりまでに育ってほしい姿」として明確化されている [62]。その中には「保育活動をとおして、思考力の芽生え、数量・図形、文字等への関心・感覚、豊かな感性と表現」とある。この文中に「プログラミング」という言葉はでてきていない。しかし、野口はこれらを前述の小学校における「プログラミング的思考」につながる要素を孕んでいると指摘する。そして、保育の中のプログラミング的思考を「プログラミング的思考の基礎」と呼んでいる [68]。また、山崎らは幼稚園から小学校の連

携だけではなく、幼稚園から高等学校まで、段階的な情報のカリキュラムが必要であるとし、その提案をしている [109]。その中で、幼稚園においては、遊具型の Programmable Toy（ロボット）を使い、遊びの中で、動かしたい動きを意図して実現することで、未就学児がプログラミング的思考を育むことを提案している。

「情報教育課程の指針設計 -初等教育から高等教育まで」においても、入学前の段階でプログラミングを通してコンピュータの原理的なものに触れることは、情報学の興味・関心を育むためによいと書かれている [54][67]。そして、幼稚園教育要領にある直接的な体験を重視すること、という文面を踏まえた上で、未就学児の体験では「コンピュータそのものの特徴的な部分」を体験することを、目的とすべきだと述べられている。具体的には、絵やロボットを動かす経験や、文字を使用せずにブロックや絵の配置で実行指示をする体験が、例として挙げられている。活動の設計については、児童一人ずつが自分のものとして、実行指示を組み立てる機会を作ること、また、実行指示の原因が自分であることが明確にわかること、が重要と書かれている。

すでに独自にプログラミングのレッスンを行う、幼稚園が出てきている [111]。また、小学校でのプログラミング教育の必修化を受けて、民間の事業者が運営する小学生向けのプログラミング教室も増えている。その中で、未就学児向けのクラスを開講する教室も多い。アーテック社は年中・年長向けのプログラミングカリキュラムのフランチャイズ事業として「FirstSTEAM」の販売を 2020 年から開始している [34]。

このように日本において、1980 年代の後半から、子どもが習得すべき教養としての、プログラミング教育が認識されている。小学校においては 2010 年代中盤から、未来における ICT 人材の不足、また、IoT の普及による社会の大きな変化に対応する手段として、プログラミング教育が注目された。そして、2017 年に改定された小学校の学習指導要領では、プログラミングは小学生が学ぶべきものとして明記された。中学校・高等学校においては教科内での学びとして、すでに取り入れられており、今後は小中高と連携した情報教育が組み立てられる見通しである。また、就学前からのプログラミング教育の必要性も提案されている。

そして民間においては、未就学児を対象にしたプログラミング教育が始まりつつある。それにも関わらず、実際に未就学児がプログラムを作れるのか、また、未就学児にとってどのような学びのツールになるのかについての、量的な研究は少ない。重要性は認められているにもかかわらず、その量的な研究が少ない原因は下記が考えられる。

- 未就学児は発話から理解を調査することが難しい。
- 端末の操作が未就学児に難しい
- 一人一台の操作端末が用意できず、グループワークでのプログラミングだったため、個々人の理解に言及できない。

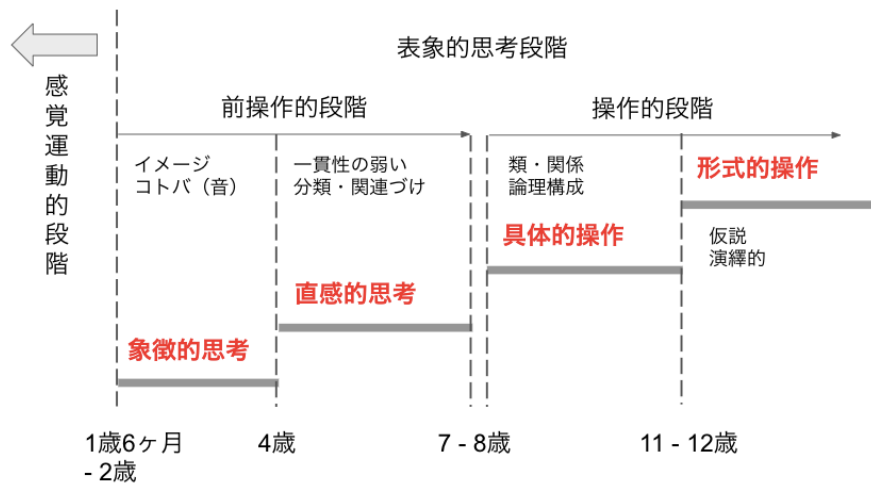


図 1.1: Piaget の発達図式 ([30] を参考にし, 著者が作成)

- 実施している幼稚園・保育園が少ない。

現在はこれらの否定的な要因は乗り越えられつつある。民間の企業・施設において、未就学児のプログラミング体験が可能になっている背景には、タブレット端末の普及と、教育用プログラミングツールの開発に要因がある。キーボードやマウスの操作を必要とせずに、コンピュータに命令をし、プログラミングを体験できる下地ができつつある [18][72][88]。必要性が論じられ、民間では未就学児のプログラミング教育に取り組む企業・施設も出てきている中、今こそ、研究的に未就学児の理想のプログラミング教育について考え、深く研究を進めていくタイミングである。

1.1.3 未就学児の学びとプログラミング体験

子どもの発達段階に関して、Piaget は図 1.1 のように示している。未就学児の年齢にあたる、3 歳から 6 歳を見てみると、「表象的思考段階」の前半である「前操作的段階」に当てはまる。「前操作的段階」では、子どもは論理を操作できないと言われている [30][86][110]。「表象的思考段階」前の「感覚運動的段階」では、何かを記憶するときに運動や動作が重要な要因になる。それに対して、「象徴的思考」期では、イメージを使って考え、記憶することができるようになる。例えば「積み木」を「電車」に見立てるなど、具体的なものを別の具体的なものに見立てて遊ぶことができるようになる。続く 4 歳から 7 歳が該当する「直感的思考」期には物事を分類したり、関連付けたりできるようになる。一方で、この時期は数や量といった概念が、置き方や見せ方の違いで同一視できなく、知覚的な目立った特徴に左右され、一貫した論理操作ができないと言われている。つまり、具体的に目に見えているものを優先に捉えるため、抽象的な絶対的な量などの把握が難しい。

7歳以降にあたる「表象的思考段階」の「操作的段階」に入ると、自己中心的な考え方から解放され、論理的に物事を考えられるようになると言われている。しかし、この「操作的段階」においても、7歳から11歳が該当する「具体的操作」期には、操作がおよぶ範囲は具体的な世界に限られる。続く11歳以降の「形式的操作」を獲得することによって、数学的に物事を扱えるようになると言われている。

発達には個人差があり、また、この段階はグラデーションがある。よって全ての学齢の児童の思考がこのモデルに当てはまるとは言えないが、未就学児を対象にしたとき、「象徴的思考」「直感的思考」に重きを置く必要がある。つまり、具体的に目に見えるものを中心に、学習環境を用意する必要がある。抽象的な数字の表現などは適さないと考えられる。

また幼児画の発達に関して、Lowenfeldは描画発達段階を以下のように分けている[26]。

1. なぐりがき時代：二歳～四歳
2. 前図式時代：四歳～七歳
3. 図式時代：七歳～九歳
4. 仲間づくり時代：十一歳～十三歳
5. 青春期：十三歳～十四歳

未就学児は「なぐりがき時代」と「前図式時代」にいると考えられる。「なぐりがき時代」においては、大人からその絵を見ても何の絵か判断するのは難しい。児童との対話と、大人の視点から、児童が何を描いたのかを読み取ろうとする努力が必要である。また、「前図式時代」においては、シンボル（記号・象徴）を使って目の前にないものを書き表すことができるようになる。これらの分類はPiagetの分類と重ね合わせても一致する部分がある。

幼稚園において、新しい教育要領が2017年に告示され、2018年より施行されている[61]。幼稚園における教育は「環境を通して行う教育」と位置づけられており、「環境を通して行う教育」とは、遊びを通しての総合的な指導の中で行われるものだとしている。幼稚園教育要領解説では、幼稚園児が就学前に目指す姿における「思考力の芽生え」には、以下の解説がある[62]。「思考力の芽生えは、領域「環境」などで示されているように、周囲の環境に好奇心をもって積極的に関わりながら、新たな発見をしたり、もっと面白くなる方法を考えたりする中で育まれていく」（中略）「幼児は、身近な事象に積極的に関わる中で、物の性質や仕組みなどを感じ取ったり、気付いたりするようになる。5歳児の後半になると、遊びや生活の中で、物の性質や仕組みなどを生かして、考えたり、予想したり、工夫したりするなど、身近な環境との多様な関わりを楽しむようになる」。

佐伯は、幼稚園教育要領の解釈としてではないが、このような環境との関わりの中での学びを以下のように説明している[81]。乳幼児はモノを使って遊ぶとき、そのモノを使って「どうすれば、どうなるか」を試している。そして、様々なモノの働き方を試みて「どうすれば、どうなるか」を探求している。この実践を経て、

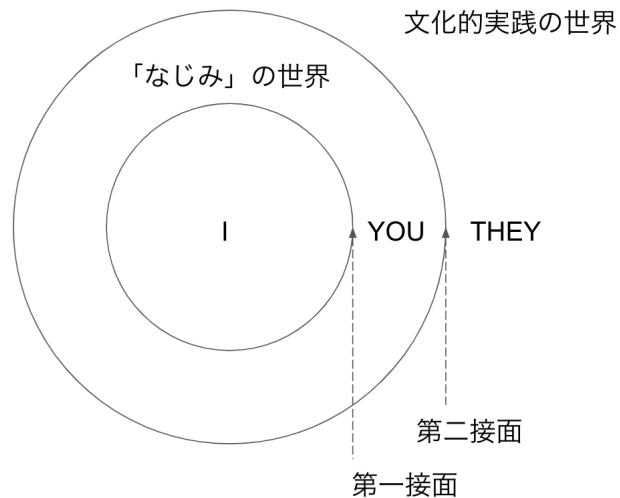


図 1.2: 佐伯の提唱する学びのドーナツ論 ([81] を参考に著者が作成)

その対象の性質を味わい，その対象についてわかるようになる。

また，佐伯は乳幼児が対象を十分探求し，「なじむ」まで熟達すると，今度はそれを誰かに「見て欲しくなる」という。例えば，モノが道具であれば，その操作に熟達すれば，乳幼児は直ちに何かおもしろいものを作ってみたくなる。そのとき，乳幼児は誰か他者に出来上がったものを見てもらったり，喜んでもらったりしてもらいたいことを希望し，期待する。ここから，子どもたちの社会や文化に向けての文化的実践が芽生えるという。

佐伯は文化的実践を，人間が自分たちの生活をより「よく」するためにする下記の4点を含む営みだと定義する [84]。

1. 「よい」とは本来どういうことなのかを探る（価値の発見）
2. 「よい」とする価値を共有しようとする（価値の共有）
3. 「よい」とされるものごとをつくり出す（価値の生産）
4. 「よい」とされるものごとを多く残したり，広めたりする技術を開発する（価値の普及）

文化的実践を定義した上で，これらの価値の発見，共有，生産，普及の活動の前提として，人がなにかを「わかる」活動があると佐伯はいう。「思考力の芽生え」とは，この文化的実践に重なる部分があると考えられる。

佐伯はこのモノを媒介にする学びについて，自身の「学びのドーナツ論」で説明をしている [81]。図 1.2 のドーナツにおいて，ドーナツの穴の部分に I（私），そしてドーナツの輪の部分に YOU（あなた），最後にドーナツの外側を THEY（彼・彼女ら）の世界だとみなす。そのとき，I と YOU の接面を第一界面とし，YOU と THEY の接面を第二界面とする。佐伯は第一界面の学びを経た後，第二界面の学びを経ることのプロセスが重要だという。

ここでYOUをモノと捉える。最初に、IはYOUとしてのモノを「なじむ」まで応答関係を繰り返し、展開させる。そうするとIにとって、モノが自分の体の延長のような「なじんだモノ」になる。佐伯はIとしての乳幼児は、モノと多様で豊かなYOU的関わりを経緯した上で「文化的実践」としてのTHEYと出会うべきだと指摘する。YOUを通り越して、自己(I)が外界(THEY)にむき出しになっていた場合、様々な世の中の規範や要請を「べきである」「ねばならぬ」として突きつけられる。自己(I)が学びや発達において、YOU的存在と出会わない場合、THEYに対して「あるべき自己」が「外(ないしは上)」から指示的・指令的に提示され、それに盲目的に従わされることになる。YOU的存在が介在することによって、自分の意味づけを持った上で、文化的実践として外界との関わりを持てるようになるという。

ここで、未就学児のプログラミング教育のあるべくデザインを、幼稚園教育要領や佐伯のドーナツ論から考える。そこではICTを使って児童自身が「どうすれば、どうなるか」を体験できる体験が必要だと考えられる。そして、そのプログラムを使って、他者に対して自分の価値観を表現できる活動である必要がある。表現とは他者に自分の考えを伝える、伝達を含んだ創作を意味する[26]。その表現が、情報社会という社会に対しての文化的実践になり、思考力を芽生えさせることができると考えられる。

また、発達段階から考えると、未就学児は数字を用いたプログラミングは不適切だと考えられる。その上で抽象的な思考を要しない、具体的かつ「コンピュータそのものの特徴的な部分」を体験できる、プログラミング教育が望ましい。

1.2 本研究の目的

本研究のリサーチクエスションは、未就学児のプログラミング教育を考えたときに、プログラミングが未就学児の表現のツールになるかどうかである。未就学児自らが情報社会の一員だと考えるためには、自己の価値観に照らし合わせた他者への、表現を通した文化的実践が必要である。それには、未就学児はプログラミング体験を通して、プログラミングを自己の延長のようになるまで理解することができるかどうかを明らかにする必要がある。その上で、未就学児が自分の価値観に照らし合わせ、プログラムを作ることができるのかどうかを明らかにする必要がある。

リサーチクエスションを踏まえた、本研究の目的は以下の2つである。

- 未就学児はプログラミングを理解できるか。
- 未就学児がプログラミングでどのような表現をするか。

未就学児のプログラミング教育を考える上で、未就学児がどの程度プログラミングを理解できるのかは、明らかにされるべきである。また、プログラミングの

理解だけではなく、それが児童の手に「なじみ」、どのように他者に対して、文化的実践としての「表現」をするのかが明らかになるべきである。未就学児のプログラムに対する理解の量的な指標と、実際に未就学児がプログラミングを駆使し、どのようなものを作るのかが明らかになっていることは、今後、未就学児における、評価指標を有するプログラミングカリキュラムを作成する上で、重要な参照になる。以上が本研究の目的である。

1.3 研究の概要と本論文の構成

本研究では次の4点を行った。

1. 未就学児の学びを考慮した、プログラミング教育カリキュラムの作成
2. カリキュラムにおける各課題に基づいた、未就学児の作ったプログラムの分析・評価
3. プログラムを用いた未就学児の表現の分析・評価
4. 1-3に基づき、本研究から得られた知見をもとに、未就学児におけるプログラミング教育を考察し、整理し、提言する

カリキュラムは第1.1.3項を考慮し、年長児全員が理解することを目指したカリキュラムを作成した。カリキュラムでは、数字での表現を用いずにプログラムを作成することができる、ビジュアルプログラミング言語 Viscuit（ビスケット）を用いた。Viscuit については第2.3.1項、第3.3節において詳しく述べる。

また、課題に基づいたプログラムの分析・評価に関しては、園児が作成したプログラムを量的に分析・評価した。本研究に際して著者は、園児の発話から園児の理解を測るのは難しいと考えた。そこでプログラム自体から理解を分析・評価する手法をとった。この手法は園児の理解を客観的・量的に測定できる手法としても有効だと考えた。それぞれのレッスンでは、課題として絵を与え、その絵が課題通りにプログラムされているかどうかを分析・評価した。その結果、Viscuit における“直線の動きの方向”“絵の変化の繰り返し”“ランダムな動き”“回転の動き”に対する、園児の理解について明らかにした。

教えられたプログラムができているかどうかだけでなく、園児がそれを使って、どのようにプログラミングを表現に活用するかを調べた。カリキュラムの最終レッスンに、プログラミングのテクニックを課題としない、表現課題を設定した。具体的には、園児が卒園式に演じる劇を、保護者に紹介するプログラムを作ること課題とした。本課題において、園児がプログラミングをどのように利用するかを分析・評価した。

本研究の成果は「未就学児のプログラミングの理解」と「未就学児のプログラミングを用いた表現」を明らかにしたことである。そして、実践を通じた本カリキュラムの効果と課題の提案である。カリキュラムを実際に実施したところ、園

児がプログラミングを理解していることがわかった一方、当初は考えの及ばなかった課題が浮き彫りになったり、教授法についても改善の余地があることがわかったりした。実践を踏まえた上で、未就学児のプログラミング教育のあるべき姿をもう一度考え直し、考察し、今後続く議論へと発展させる。

以下、第2章では、未就学児を含む子どもを対象とした、プログラミング教育に関する先行研究を述べる。第3章では、本研究の舞台である香川富士見丘幼稚園、および、プログラミングレッスンの実施環境・実施方法について説明し、園児の作ったプログラムの分析方法について述べる。第4章、第5章、第6章では、課題を伴ったレッスンについての、園児のプログラムの分析結果を述べる。第7章では、最終レッスンにおいて、園児が12回で学んだテクニックをどのように活用して、プログラムを使って表現しているのかについて述べる。第8章では、本論文の総括をおこなう。

第2章

先行研究

2.1 はじめに

本章では国内・海外を含めて未就学児に関わらず，子どもを対象にしたプログラミング教育の先行研究について述べる．先行研究を述べる上で，子どもとプログラミングに関わる研究を，2つの節に分けて述べる．

第2.2節では，子どもを対象にしたプログラミング教育に関する研究について，3つの項に分けて述べる．

第2.2.1項では，教育用プログラミングツールに関する研究について述べる．子ども向けのプログラミングツールの研究は1960年代後半に始まり，いまや様々なツールが存在する．それらのツールは，ロボットを用いるものや，スクリーン上でのみプログラムを作るものなど様々である．それらを比較，考察する．

第2.2.2項では，子どもがプログラミングを通して，何を学ぶのかについての議論について述べる．1960年代後半の子ども向けのプログラミングツールの開発時から，子どもはプログラミングを通して何を学ぶか，に関しての議論は様々ある．現在ではプログラミング教育の目的は，コンピュータサイエンス教育に置かれる場合が多い．そして，コンピュータサイエンスのどの概念を学ぶのか，また，どういう学びが必要なのか，そして，学んだあとに児童のどのような変化が求められているのか，という様々な議論を確認する．

第2.2.3項では，未就学児を対象にしたプログラミングの理解に関する研究を述べる．多くの研究では，未就学児はプログラミングをすることができ，また，プログラミング自体を楽しみ，非常に動機付けられている姿が明らかになっている．その中で，ロボットを使ったプログラミング教育の報告は多い．未就学児の中には，高度なプログラムも理解できる児童もいる，という報告もある．しかし，スクリーン上でプログラムを作るプログラミングに関しては研究は少ない．それらの様々な研究を本項では述べる．

第2.3節では，本研究で用いているビジュアルプログラミング言語 Viscuit（ビ

スケット)に関する先行研究について述べる。

本研究で使用している Viscuit は、図形書き換え型のプログラミング言語である。第 2.3.1 項では、図形書き換え型のプログラミング言語の歴史について述べる。図形書き換え型言語は 1990 年代から活発に議論がなされ、非専門家でもわかりやすく理解できるプログラミング言語として研究されてきた。どのような研究的な背景の上に Viscuit が開発されたかを述べる。

第 2.3.2 項では、Viscuit について詳しく述べる。Viscuit における命令方法、また、手続き型の言語との違い、そして、Viscuit でできるプログラミングについて説明する。

最後に、本章のまとめを行う。

2.2 子どもを対象にしたプログラミング教育に関する研究

2.2.1 子どもを対象にしたプログラミングツールの研究

子どもを対象にして、どのようにプログラミングに接してもらうかの議論は Papert から始まる [98]。Papert は「子どものためのコンピュータ」についてビジョンを描いた。Papert は教育において、コンピュータを人間の思考を助ける道具としてだけでなく、本質的に人間の思考過程を変える道具として捉えた。Papert は Piaget の構成主義を土台として、構築主義の学びの重要性を訴えた。Piaget の構成主義では、知識を個人個人の中に蓄積されていくものではなく、社会やコミュニティによって構成され、また、それらに依存しているものとした [102]。例えば、母国語を話すとき、それは教えられたから話せるのではなく、教えられてなくても覚えてしまうように、学習は教育によってだけでなく、環境との相互作用の上に起きる。学習が環境との相互作用で起こる中で、ある学ぶ対象自体が抽象的で難しい概念を持つとき、その対象を学ぶことは難しくなるという。一方、Papert の構築主義では、子どもたちを建築者にとらえる。そして、子どもは、建築者のように「自分の知っていること」を組み合わせる新たな知識を学んでいく、という。そして、Piaget がいう難しい概念は、対象を具体化するような材料がそのコミュニティに存在しないことが原因だと指摘する。その上で、コンピュータがその難しい抽象的な概念を具体化するツールになるという。子どもはその時点で知っていることと、コンピュータで具体化されたことを組み合わせ、従来は学ぶことが困難だと言われていたことも、学ぶことができるという。つまり、子どもが何かを学ぶとき、対象を具体化できるツールがそのコミュニティにあれば、子どもはその対象を既存の知っていることと組み合わせ、学ぶことができると訴える。

Papert は構築主義的に学ぶことにより、知らないものを無理に個人に取り込むのではなく、自分が知っていることの組み合わせによって、知識を発見する学び



図 2.1: Logo における床置きタートル ([98] より引用)

が可能になるといった。そして、この学びがプログラミングによって可能になると指摘した。なぜなら、プログラミングをすることで、抽象的な概念が具体的に捉えられるからである。このような考えから、Papert らは 1967 年にプログラミング言語 Logo を開発した。

Logo は学校教育、とくに数学教育を目的に作られた。プログラムは forward や right などのわかりやすい英単語で記述できた。また、Logo の持つ「タートルグラフィックス」では、タートルというオブジェクトにそれらの命令を与えることで、その軌跡で絵を描くことができた。床置きのロボットも開発され、スクリーンに絵を描く以外に、実際に床に軌跡を描くタイプのデバイスもある (図 2.1)。

しかし、Logo は、1970 年代を通して多くの学校で使われていたにも関わらず、当初の熱狂は長く続かなかったと報告されている [73]。その理由として、Logo 自体のもつ直感的ではないシンタックスと、プログラムをするときに正確性を求められた点が挙げられている。さらに、子どもへの教え方自体も十分に理解されないまま、現場の先生に活用されるに至った結果、子どもの興味を継続できなかったといわれている。

1980 年代後半から、コンピュータの専門家ではない、一般の人や、初学者が簡単にプログラムを作れるための言語として、手続き型ではない様々な言語が生まれている。例示プログラミングや図形書き換え型言語である。例示プログラミングとは、プログラムを記述するのではなく、コンピュータ上で実行したい操作をユーザーが記録、また、録画し、その操作をコンピュータが解釈し、一般化し、実行させる手法である [53]。図形書き換え型とは、書き換え型の言語の種類であり、ある図形、または、パターンを条件とし、その図形、パターンが現れた場合、その図形、パターンを別の図形、パターンに書き換える手法である。図形書き換え型言語については第 2.3.1 項で詳しく述べる。

兼宗は、例示プログラミングはプログラミング言語の文法を覚えなくてもよいことが利点だが、デメリットもあると指摘する [39]。そして、教育目的で例示プログラミングを採用するときのデメリットとして以下を挙げている。

- プログラムを記述できない
- プログラムの動作原理の理解につながらない

利用者はプログラミングをするとき、プログラムを明示的に記述することがないため、プログラムの構造を学習できない。そして、例示する行為では、計算機の内部でどのようなアルゴリズムが生まれ、処理が進んでいるのか見えない。そのため、プログラミングの理解が深まらない点を指摘している。

Repenning も、例示プログラミング、また、図形書き換え型のようなグラフィカルな簡単なインターフェイスの言語を批判している。これらを使うことは、簡単なプログラムはできるが、複雑なプログラムには通じないことを指摘している [77][85]。このように、図形書き換え型言語では「遊び」にはなるかもしれないが、汎用的な教育用のプログラミング言語としては不十分であるという指摘がある。1990 年代には様々な図形書き換え型言語が発表されている [5][17][75][91]。しかし、2000 年代を通してこの図形書き換え型の言語は、教育シーンにおいて主流にはならなかった。

一方で、前述した Papert の思想と Logo は、現在普及しているプログラミングツールに非常に大きい影響力を示した。1990 年代初頭に Resnick は、Logo におけるタートルを複数操作することを可能にすることによって、群れのようなシミュレーションをプログラムすることが可能になった StarLogo を開発した [74]。StarLogo では、それぞれのタートルが「感覚」のパラメータを持てるようになっており、生物の群のシミュレーションをすることができる。また、1997 年には Logo, Starlogo, Hypercard, Powerpoint に影響を受けた Squeak Etoys が発表された [40] (図 2.2)。Squeak Etoys は子ども向けのプログラミング学習言語である。この言語はスクリーン上で絵や図形を、ブロック型のプログラムで動かすことができる。また、音の録音などもでき、画像のファイルを取り込むこともできる。それによって、子どもは自分の知っている知識を組み合わせで構築主義的に学ぶことができる。そのようなマルチメディアな学びを促進する目的で開発された。Squeak Etoys はオブジェクト指向のビジュアルプログラミング言語であり、ユーザーは絵にブロックによる命令を重ねていくことでプログラムを作成する。

また 1996 年には LEGO を使ってロボットを作ることができる、マインドストームが発表される [45][46] (図 2.3)。マインドストームでは、画面上で命令のブロックをつなげてプログラムを作り、それをマインドストームに読み込ませ、ロボットを動かすことができる。マインドストームは LEGO ブロックを接続することができ、LEGO でブロック作品を作るように、ロボットを作成し、プログラムを作り、動かすことができる。モーターやセンサーなど、ロボットエンジニアリングに深く関わる知識を、子どもは遊びながら学ぶことができる。

その後、ロボットを使ったプログラミングのツールは他にも開発される。ロボットは形があり、具体的であるため子どもでも理解しやすい。そのため、未就学児のプログラミング教育ツールとしてロボットは頻繁に利用されている。CHERP や

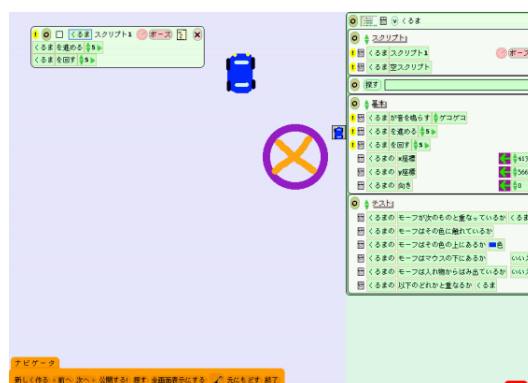


図 2.2: Squeak Etoys の画面 ([99] から引用)

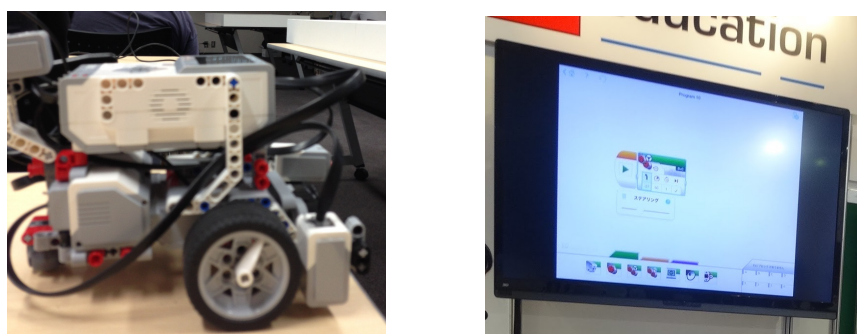


図 2.3: 左 : LEGO マインドストームの躯体, 右 : ロボットを動かすプログラムを作る画面

KIBO, また, キュベットなどが未就学児向けのプログラミング教育では使われている [14][72][97]. ロボットタイプのツールには, マインドストームのように画面でプログラムを作り, ロボットに読み込ませるものの他に, コンピュータを介さないでプログラムすることができるものもある. キュベットに関しては, キュベットに無線で繋がったデバイスに, 命令のブロックを具体的にはめ込み, できた命令を無線で通信することで, キュベットを動かす (図 2.4). ロボットを単純に動かすことと, いくつかの繰り返しの命令しかできないが, 非常に簡単にロボットをプログラミングで動かす体験が可能である.

スクリーン上でプログラムを実行するものにおいては, Squeak Etoys の後継となる Scratch が 2007 年に発表される [1][87]. Scratch では Squeak Etoys にあったブロックの命令から, 必要なものを取捨選択することによって, 子どもがより使いやすいプログラミング言語になっている. Scratch は 2020 年現在, 世界で最も使われている, 子ども向けのプログラミング言語の一つである. また, Scratch を 5-7 歳の子どもむけに改良した ScratchJr が開発されている [15][50][88] (図 2.6). ScratchJr は Scratch に対して, 命令のブロックがさらにわかりやすく, 少なくなっている. 数値表現を用いずに, ブロックの量で命令を組み立てることができる. また, カメラで写真を取り込んだり, 音を出したりして, 様々な要素をプログラム



図 2.4: キュベットでプログラムする児童の姿

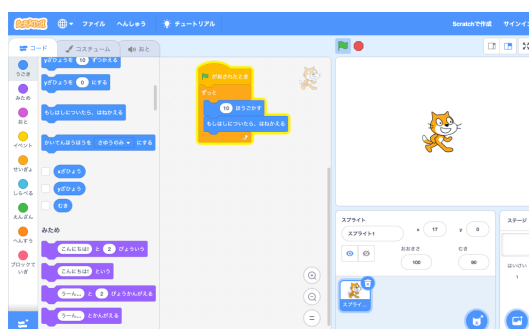


図 2.5: Scratch の画面

に入れることができる。ロボットや Scratch や ScratchJr については，子どもへのプログラミング教育の研究がある．それらの研究は第 2.2.3 項で述べる．

2003 年には，本研究で活用してる Viscuit が開発されている [21][22][23][24][25]．Viscuit については第 2.3.2 節で詳しく論じる．

他にも日本語でプログラミングをすることができるドリトル，アルゴリズムをゲーム感覚で学べるアルゴリズムや，欧米の非営利団体が提供する code.org，文字を使わず，絵の関係性でプログラムを作るスプリングンなど様々なツールがある [10][28][35][39]．また現在も進行形で，プログラミングを教えるためのツールは，世界中で開発され続けている [27][37]．

このように，1960 年代後半から始まった，子どもを対象にしたプログラミング教育に関して，現在では様々なツールが存在する．これらのプログラミングツールにおいて，その「表現」「具体性」「物理制約」「拡張性」に焦点をおき，まとめた表が表 2.1 である．プログラミングツールは，大分類として，文字で命令を入力する「テキスト型」とブロックの組み合わせで命令を入力する「ブロック型」とにわけた．また，それらをスクリーン上のみで操作するものとロボットを操作するものに分けた．それぞれのカラムにおいて「表現」はその言語でできる表現の幅を示す．表現できるものの幅が広いものを○，表現できるものが制限されているものを×で示した．△は○と×の間にあるものとした．「具体性」とは，プログラ



図 2.6: ScratchJr の画面

表 2.1: 様々な教育用プログラミングツールの比較

大分類	中分類	言語名	表現	具体性	物理制約	拡張性
テキスト型	スクリーン	Logo	○	×	なし	○
		ドリトル	○	×	なし	○
ブロック型	スクリーン	Squeak Etoys	○	△	なし	○
		Scratch	○	△	なし	○
		ScratchJr	△	○	なし	○
	ロボット	マインドストーム	△	△	あり	○
		キュベット	×	○	あり	×

ムを作るときに抽象的な思考を必要とするか、しないかを示す。本研究では、未就学児のプログラミング教育がテーマであるため、抽象的な命令が必要であるものを×、抽象的な命令を用いずに、具体的な命令でプログラミングができるものを○とした。△は○と×の間にあるものとした。「物理制約」とは、命令の実行が物理法則に影響を受けるかどうかである。「あり」または「なし」で示した。「拡張性」とは、他のメディアを取り込めたり、接続ができた；また、入力・出力を他のメディアでできるかどうかを示す。

この表をみると、ドリトルのように汎用言語に近い、または汎用言語を学ぶ前段階の教育言語は、表現力は高いが、抽象度が高い（具体性がない）ことがわかる。テキスト型の言語の場合、様々な表現ができるが、抽象度が高い。また、ブロック型の言語では、Squeak Etoys や Scratch のようにプログラミングも実行も、両方がスクリーン上で行われるものについては、表現できるものは多いが、プログラムを作るのに数値表現を必要とする。テキスト型の言語に比べて、ブロック型を使うので具体性は高いが、しかし、依然として抽象度は高い。ブロック型を未就学児向けにした ScratchJr は、ブロックが大きくなり、ブロックをつなげることで、数字を使わずにプログラムを作ることができるが、表現できる幅は少なくな

る。また、ロボットにブロック型言語で命令するタイプは、ロボットの上で行われる実行が物理制約を受けるため、誤差が生じる。物理制約があるものは、準備に時間がかかったり、用意することに労力が必要でもある。マインドストームに関しては、ロボット自体の制限があるので、表現できることは限られるが、センサーやモーターなどとの接続による拡張性は高い。キュベットに関しては非常に簡単で、具体的だが、表現の幅は限られている。

このようにみると、抽象度が高ければ高いほど幅の広い表現ができることがわかる。また、ロボットに関しては、物理制約がかならず伴う。できる表現が限られる場合、ユーザーが「どうしたら」「どうなる」の意味の探求が深くできない可能性がある。また、情報の原理を濁らずに体験するためには、物理制約がない方が良いと考えられる。未就学児に必要なプログラミングツールとは「表現」が○であり、かつ、「具体性」も○であり、そして、「物理制約」がないものが望ましいと考えられる。

以上が、主に小学生以下を対象にしたプログラミング教育のツールとその歴史である。

2.2.2 子どもは何を学ぶのかについての研究

Papert は、コンピュータが子どもに知識を与えるために使われるのではなく、子どもがコンピュータにプログラミングをする過程で学ぶべきだと説いた [98]。構築主義では、学ぶべきものが予め用意されているのではなく、子どもは自分の知っている知識を組み合わせることを通して、新しい知識を発見すべきだと言われている。例えば、子どもは Logo を使って図形が描きたかった場合、座標の数値を元にその数を実際に自分で歩いてみたり、実際に回転してみたりする。そして、コンピュータに実行させたいことを、自分の動作で確認し、自分の知っていることを組み合わせ、自分で発見していくことがよいとした。こうすることで子どもは、その子どもの得意、または、不得意が学ぶ対象によって決定づけられるのではなく、子ども自身がコンピュータを利用して、自分の特性を活かせる方向に成長することができる考えた。

その後、コンピュータを使った教育実践は盛んに行われ、様々な利点も明らかにされている [9]。子ども達がコンピュータを使って学ぶとき、それぞれが個で隔離されるのではなく、子たち同士が喋る時間が増えることが報告されている [64]。また、子どもが友達と一緒にコンピュータを使うときに、一人で使うときよりも熱心な態度を見せることが明らかになっている [29]。具体的に数学の教育についても、コンピューターの利用が有用なことが明らかになっている [8][16][96]。

現在では、プログラミング教育の意義として、コンピュータサイエンス (CS) 教育が挙げられる場合が多い [51]。情報機器の利用が生活の中に空気のように溶け込んでいる今、CS 教育を通して、情報サービスに批判的な視点を持つ人材の育成が必要とされている。また、批判的な視点をもった上で、新しい価値・技術を作

り出す人材が求められている．その中でコンピュータ科学者ではない人が，コンピュータ科学者のように考える思考法が，コンピューショナルシンキング（CT）として言われている [106]．CT という用語は 1960 年代からあるとされている．それを，2006 年に Wing が改めて「世界的に誰にでも適用できる姿勢とスキルのセットであり，学ばれ，使われることが望ましいものである」と提案し，改めて普及した．Wing は CT を，特定の問題をコンピュータを使って解決するために，コンピュータが実行できる形に表現する思考のプロセスだと定義している．この定義の中で，CT のキー概念は「抽象化」だと言われている．問題を解決するとき，解決方法をコンピュータが理解できるように，抽象化できる思考法を育むことの重要性が述べられている．近年の子ども向けのプログラミング教育では，この CT をどのように子どもに教えるかに，焦点が置かれる場合が多い．我が国のプログラミング教育における「プログラミング的思考」は，この概念を参考に行っている部分が多い [59][60]．

我が国ではプログラミング的思考は「順次」「分岐」「繰り返し」の 3 つの概念の組み合わせとして論じられる [90]．低学年児童に対する授業では，この 3 つの概念をコンピュータを使わずに体験する授業も行われている．それらはダンスを手順と繰り返しで表したり，天候の変化で洋服が変わることを考えることを通して「分岐」を学んだりする形で行われている [101]．またアメリカにおいて，CT 教育を小学校に普及する団体である CT4edu は，CT の基本的なコンセプトが「問題の分解」「アルゴリズム」「抽象化」「自動化」だとし，それらを教育の中で取り入れられるキットを公開している [107]．

一方で，上記の概念は命令を順次実行する形の手続き型の言語の考え方である．Repenning は CT を学ぶために，複雑で難解な手続き型のコーディングが必要であることに疑問を唱えている [78]．そして，Wing の定義に従い，CT の過程を 3 つのステージで説明している（図 2.7）．

1. 問題形成（抽象化）
2. 解決の表現（自動化）
3. 解決の実行と評価（分析）

この 3 つのステージを繰り返しながらコンピュータを使うことが，コンピューショナルシンキングプロセス（CTP）だと示している．このプロセスにおいて，問題形成と解決の表現は人間の能力により，問題の実行はコンピュータの能力によって行われる．そして，人間の能力と，コンピュータで可能なことを統合する，繰り返しのプロセスが CTP だと指摘する．また，それを助けるツールのことをコンピューショナルシンキングツール（CTT）と呼んでいる．その上で，優れた CTT とは，コンピュータに実行をさせるときに，できるだけ考える負荷が少なく，ユーザーを混乱させる偶発的な複雑性がないものだとする．Repenning はその CTT の 1 つの例として，自身が開発した AgentSheets をあげている [75][76][77][78]．AgentSheets はブロック型の命令と，if-then のルールベースのハイブリッドな命令形式を有す

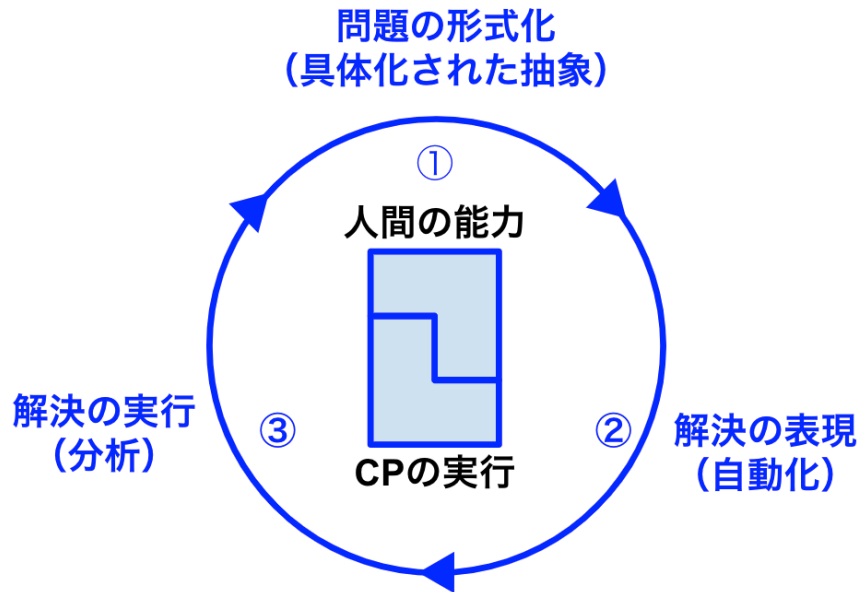


図 2.7: コンピュータシヨナルシンキングプロセス (CTP) ([78] を参考に著者が作成)

る言語である。その特徴について、手続き型のコードを書いて実行するよりも容易に、数個の命令で複雑なシミュレーションやゲームが作れる事実を指摘している。AgentSheets については次節で取り上げる。

つまり、CT 自体を学ぶためには、複雑なコーディングの能力の獲得が必要ではない。コンピュータ上で自分の実現したい内容を考え、それをコンピュータが理解できる程度の抽象的な命令にし、コンピュータで実行させ、結果を分析・評価する、このプロセスが実現すればよい、としている。CT を獲得するために、コーディングが前提ではないということは、我が国の教育指導要領とも一致することである。

一方で CT 自体に対する批判もある。Denning は Wing に触発された CT を新しい CT (New CT) と呼び、それまでの従来の CT とは異なるものだと指摘する [11]。New CT の普及は、1960 年から 1970 年代に広がった「Computer Science = Programming」という、CS を狭めて見る図式を「CS = New CT」に鞍替えしているだけであり、やはり CS という領域を狭く解釈することになっているという。Wing の New CT は単に存在していた CS を改めてまとめただけでなく、さらに CS の分野を過度に狭くしているという。さらに、そもそもの CT 自体は「アルゴリズム的思考」として自然科学の中に存在し、ある特定の CS の部分にのみ使われている概念ではなく、広く科学者が使っているものだと指摘している。

本研究で用いる Viscuit の開発者である原田は、CT を学ぶこととは違う視点で、プログラミングを体験する重要性を訴える [25][94]。原田は合同会社デジタルポケッ

トで作成した小学生向けの連続授業の中で、以下のことを伝えている。

1. コンピュータは命令の通り動くが、間違いはわからない
2. ものと情報の原理の違い
3. コンピュータと人間の得意なことと不得意なことの違い
4. 1つの命令は単純だが、数が増えることで複雑になる
5. 情報化社会を作る当事者は君たちだ

最初の4つに関しては、コンピュータの特徴である。そして最後の1つは子どもたちの、授業を受けたあとの態度に関するものである。つまり、学ぶ内容も大事だが、子どもたちがプログラミングを通して成功体験をし、これからのコンピュータは自分たちの世代が作っているのだ、という「想い」を芽生えさせることが重要だと指摘する。

このように、プログラミング教育で子どもは何を学ぶのかについての議論は多くある。その議論は具体的な学ぶべき内容を指摘するものから、その学ぶ過程を指摘するもの、また、学んだあとの状態を指摘するものなど様々である。

2.2.3 未就学児のプログラミングについての研究

未就学児のプログラミング教育において、プログラミングを教えるツールは様々ある。また、何を教えるべきかの考えも様々である。それぞれの角度の様々な研究が必要であり、今後も継続して明らかにされていくべきフィールドである。本項では、未就学児のプログラミング教育に関する研究に焦点を当て、現在までに判明している未就学児のプログラミングの理解について整理する。

未就学児が実際にプログラミングをすることができ、また、その際に非常に興味をしめし、意欲的にプログラミングに取り組んでいたことが、多く報告されている [4][9][50][63][71]。しかし、未就学児がどのような概念まで理解できるのか、また、どのようなプログラムを個々の児童が作成するのかについての研究は少ない。

Kazakoff は、コンピュータの中においても、日常の生活においても、順次処理が重要なことに着目した [41]。そして、幼稚園の中でロボットを用いたプログラミングの実践をすることによって、児童の順次処理の能力が向上するかどうかについて研究した。研究では、実際の幼稚園で、幼稚園の教諭の指導のもとで順次処理・並べ替えの能力が向上するかどうかの実験が実施された。実験の結果、プログラミングを体験したグループと、体験していないグループを比べたところ、プログラミングを体験したグループは、順次処理・並び替えの能力が向上したことを報告している。一方で、この報告のなかでは、児童が実際にどのようなプログラムを作ったのかについては述べられておらず、児童のプログラミングの理解については言及がされていない。

Anzoategui らは、4歳から5歳の児童21名をグループに分けて、キュベットを用いて任意の道順をプログラムさせる課題を実施した [4]。そして、児童のそれぞ

れのグループが、ゴールに辿りつくためにどのような戦略をとったか、また、プログラムを作るのにどれだけ時間がかかったかが報告されている。この研究では、児童はあらかじめ複数のプログラム例を見せられ、グループで議論し、提示された例からプログラムを選んで実行している。このため、それぞれの児童がどの程度プログラムを理解したのかは不明である。

一方で、Flannery らは未就学児のプログラミングへの理解に踏み込んで研究をしている [14]。Flannery らは 4.4 歳から 6.6 歳（平均 5.6 歳）の幼稚園児たちが作ったプログラムについて研究した。実験の中で、Flannery らは被験者である児童達を、Flavell の発達段階の分類に従ってチームに分けた [13]。その上で、児童にプログラミングを体験させたところ、それぞれの発達段階で明確にプログラミングに対する理解が違ったことが確認されている。彼らは CHERP を使っている [7][12]。実践の中で、前操作期の児童はしばしばプログラムを作ることに困難を見せが、一方で、具体的操作期の児童は、教えられたこと以外のことも、進んで自ら探求することができたことが報告されている。

また、Marina らの研究では、4-6 歳の児童を対象にし、6 つのレッスンで構成された一連の講座を実施した [6]。Flannery らと同じように CHERP を使用し、児童のプログラミングをする様子を記録し、彼らが作ったプログラムも合わせて児童の理解を分析した。この研究では多くの児童が、このカリキュラムの前半については高い理解を示したと報告されている。しかし、後半の難しい部分になると、理解を示す児童の数は減ったと報告されている。特に条件分岐の部分では困難を示す児童が多かったということである。

小学生を対象にした研究においても、条件分岐は難しいことが報告されている。Cecilia らはロボットプログラミングの体験を通した、未就学児と小学生の学習を比較した [52]。その研究の中で、5-6 歳の子供は「順次」、「繰り返し」、「パラメータ」を使うことはできたが、「条件分岐」には、理解に困難を示していたことが報告されている。また、Linda らはインターネット上で、Scratch を用いて作成されたプログラムについて調査している [89]。Scratch のブロックの中で、「衝突」「相互作用」「スコア」といったブロックの利用が、「見た目」「動き」「変化」のブロックに比べて、使われている頻度が低いことを報告している。「衝突」「相互作用」「スコア」といった動きは「条件分岐」を伴うものである。この研究では、オンライン上に保存されたプログラムをダウンロードし、分析しているため、プログラムを作った児童の年齢は明確にはなっていない。しかし、多くの人が「条件分岐」に難しさを感じていることがわかる。

ロボットを使わず、ソフトウェアのみを使った研究として、Papadakis らの研究がある [71][88]。Papadakis らは ScratchJr を使い、1 年間 13 回にわたって幼稚園でプログラミングの実践を実施した。この研究では、児童が作ったプログラムの分析が行われている。命令のブロックの中では、「動き」のブロックのうち、特に「右に動く」というブロックが一番使用されていると報告されている。また「ジャンプ」というブロックと「上に動く」「下に動く」のブロックを混同している様子

や、「右に曲がる」「左に曲がる」のブロックを混同している様子が見られたと報告されている。しかし、課題に対してどれくらいの児童が正しく動かせていたのか、具体的にどれくらいの割合の児童が思い通りに動かせていたのか、についての言及はない。

上記の先行研究をまとめると次のことが示されている。

- ロボットをつかった、未就学児が集団で取り組むプログラミングに対する理解の研究は多くある。
- 未就学児において、また、小学生においても、条件分岐は難しい場合が多い。

本項では、未就学児に対するプログラミング教育の研究を確認した。ロボットを使った研究が多い理由は、ロボットが具体的であり、未就学児の発達段階とあっているからだと考えられる。また、未就学児や小学生においては「条件分岐」という概念は抽象的であり、認識が難しいことがわかる。一方で、スクリーン上のみを使った未就学児のプログラミングの理解についての研究は少ないことがわかった。ロボットは具体的であるが、高価である場合も多く、準備についても労力を要する場合がある。今後のタブレット端末の普及を考慮すると、スクリーン上のみで命令と実行ができるプログラミングの理解についても、明らかになる必要がある。以上が、未就学児のプログラミングの理解に関する先行研究である。

2.3 ビジュアルプログラミング言語 Viscuit（ビスケツト）に関する研究

2.3.1 図形書き換え型言語に関する研究

1980年代から、グラフィカルなインターフェイスの可能性が指摘される中で、1991年に Furnas は BITPICT を発表している [3][17]。BITPICT ではビットの配置で命令を作成する。左の状態をビットの配置で指定し、その状態が画面上にあれば右の状態に変化する、というルールを作ることで命令する（図 2.8）。ルールで指定された変化は繰り返し画面上で実行される。また、ルール間に矛盾が生じた時のために、矢印の下に数値入力を行うことで、ルールに優先順位を作ることができる。

Furnas は 3 つの主張をしている。

1. グラフィカルなインターフェイスはプログラミングを学んだり、プログラムをすることを簡単にする
2. 絵全体を書き換える法則によってグラフィカルなコンピューティングが可能である
3. 1. と 2. は強く関連している。

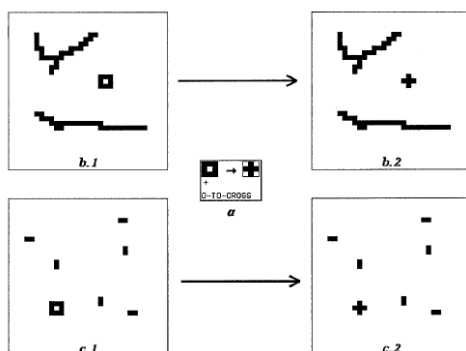


図 2.8: BITPICT の画面 ([17] より引用)

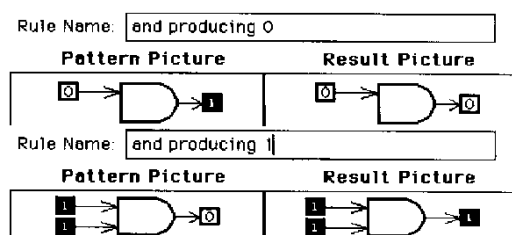


図 2.9: ChemTrains の画面 ([5] より引用)

1. については、初心者にとって、また、特定のタスクにあっては熟達者にとっても当てはまるという。そして、1. と 2. は人間が本質的、また、遺伝的に、絵によって推論できる特徴があるから関連を持つものだという。よって、人間が図形・絵をどのように認識しているかを理解し、図形・絵による推論・コンピューティングの可能性を広げることが重要だと述べている。そして、非センセンシャルな推論システムに関する理論的な研究が、コンピュータを直感的に操作することを実現する可能性がある」と指摘した。

また、Bell らは 1993 年に ChemTrains という図形書き換え型言語を発表している [5]。ChemTrains は限られたプログラミングの知識しか持たない人でも使える、プログラミング言語として設計された。ChemTrains は BITPICT に対して、ルールの適応の幅を増やすために変数が導入していることや、ビットの配置ではなく、絵自体の変化を採用しているところに違いがある (図 2.9)。ChemTrains も左 (前の状態) と右 (後の状態) を指定するルールを作成することで命令を作る。論文の中では複雑なアプリケーションが数個のルールで実装されることが報告されている。ChemTrains は、ユーザーのできる範囲を絵の振る舞いに限定し、ルールを絵に特化させることで、初学者でも簡単にプログラムが作れるようにした。また、この言語はプログラミング言語について習熟している人は対象にしていけないことを明示している。Bell は図形書き換え型に関して、具体的に図形・絵を用いることにより、抽象的な表現への変換を必要とせず、複雑な動作も命令できることを特徴だとして挙げている。

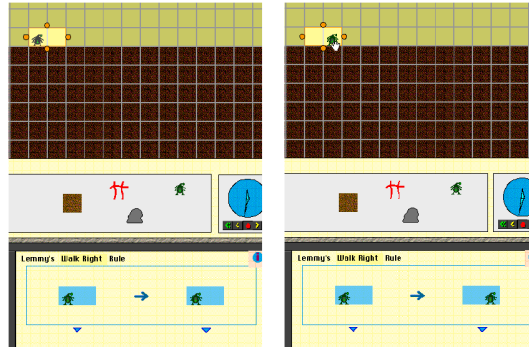


図 2.10: KIDSIM の操作方法 ([2] より引用)

1994 年に Smith らは、子ども向けの教育用プログラミング言語として KidSim を発表している [2][91]。KidSim は、プログラミングをしたことがない人でも理解しやすいように、汎用の言語としてではなく、シミュレーションに特化した言語として開発された (図 2.10)。Smith らはソフトウェアを知る一番良い方法はプログラムを書くことであるという。一方で、プログラミングの一番の問題は、多くの人にとって難しすぎるのだと指摘する。そして、Smith はその原因を文字入力・表現だとし、KidSim に図形書き換え型の命令と例示型プログラミングの概念を導入した [53]。

KidSim も左から右の状態に変わることを指定するルールを使い、絵の配置で命令をする。命令が実行されるステージは格子状になっており、画面上の任意の部分を選択し、選択した範囲の中で絵を動かすと、その行為がルールとして記録される。また、KidSim ではユーザーインタフェースの設計上で、人間の操作と実行がスクリーン上に可視化され、ユーザーに対して操作や実行が隠されているのではなく、情報を見せることで、コンピュータ上で何が起きているのかを理解しやすくすることを大切にしている。内部で起きていることを隠すのではなく、コンピュータが見ているものと人間が見ているものを同一のものにすることを目指している。KidSim は 1996 年に Stagecast Creator として販売されたが、現在は購入することができない [95]。

Repenning らが開発した AgentSheets[77] は、1991 年に図形書き換え型を採用する形で発表された。しかし、その後、図形書き換え型から、Before, After の状態の遷移を命令する形を維持しながら、ブロックを使った命令形式に変わっている。ある状態が存在した場合、その状態を指定された状態に変える、ルールベースのプログラミングと、ブロック型のプログラミングがハイブリッドに共存している形を採用している (図 2.11)。Repenning は図形書き換え型は短絡的な方法だと批判する。そして、基礎的なゲームやアニメーションを作るのには非常に効果的だが、より洗練されたゲームやシミュレーションを作ることを可能にするようなプラットフォームにはならないという。

Repenning も手続き型のプログラミングが難しいことを主張している。そして、第 2.2.2 項で述べた通り、CTP と CTT を提唱する。CTT においてはユーザーが

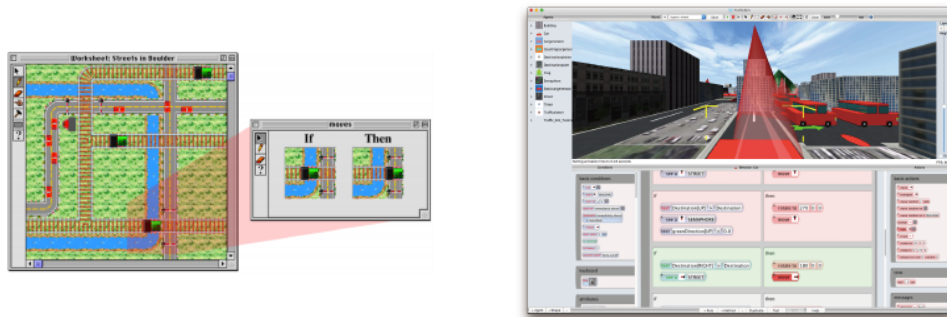


図 2.11: 左：図形書き換え型を採用した AgentSheets，右：ブロック型の if-then ルールを採用した AgentSheets ([77] より引用)

プログラムを作成するときに，必要ではない負荷，偶然的な複雑性に関してはできる限り削減すべきだと主張している。

他にも図形書き換え型を採用している言語として，1996 年に山本らが発表した Visulan や，2007 年に Kindborg が発表した ComiKit などがある [44][108]．また次節で示す Viscuit も図形書き換え型言語である。

このように図形書き換え型言語は，プログラミングの知識がない，一般の人にとって非常に理解することが簡単な記述形式であり，プログラミングが難しいと感じる人のために開発されてきた．その特性は具体的であり，直に触っているような操作感であり，複雑性を排除し直感的に理解ができることだった．一方で，汎用のプログラミング言語に比べ，複雑なプログラムを作ることが難しいという批判もあった。

2.3.2 Viscuit におけるプログラム

Viscuit は 2003 年に原田が開発したビジュアルプログラミング言語である [18]．Viscuit は KidSim を参照して開発されている．Viscuit は KidSim と似ている一方で，違う部分も多数ある．例えば，KidSim においては，命令は並べられた順番に実行されるが，Viscuit では命令は並べられた順番に関係なく，実行される．また，KidSim ではそれぞれの絵・ルールにはプロパティがあり，その中で絵の内部情報を数字で入力することができるが，Viscuit は絵の内部情報を指定することはできない．よって，Viscuit においては文字や数字を全く使わずに，命令を作ることが可能である。

KidSim を含めたそれまでの図形書き換え型言語に対して，Viscuit の新規性は絵を配置するグリッドがない点である．絵を配置するグリッドがないため，ユーザーは絵をステージ上，また，ルールの中のどこでも配置できる．また，グリッドに従わないため，絵の配置で作られた条件には「緩やかな書き換え」を適用しており，条件の発動と，結果が厳密にならない実装がされている．グリッドに従わ

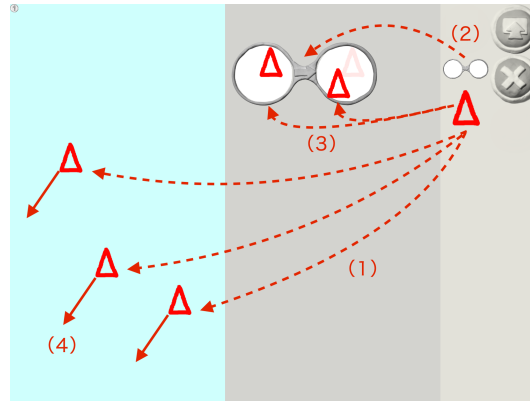


図 2.12: Viscuit の製作画面

ないので、絵の回転もできる点も、Viscuit よりも前の図形書き換え型言語に比べて新しい点である [21][22].

2003 年に開発されてから改良が重ねられ、現在はタブレットでも利用することができる. 文字入力を必要とせず、タブレット端末でも利用できるように、マウスの操作やキーボードの操作が難しい、未就学児もプログラミングを体験できる. 「前の状態」を「後の状態」に変える、メガネと呼ばれるツールの意味を覚えるだけで、プログラムを作ることができるため、プログラムを作るまでのハードルが非常に低い [25].

Viscuit では、図 2.12 のように、部品置き場（右端）にある絵をドラッグして（ドラッグすると自動的にコピーされる）、ステージ（左半分のエリア）に置き（1）、メガネと呼ばれるツールをメガネ置き場（グレーのエリア）に置き（2）、メガネの左右の丸それぞれにも絵を入れる（3）。左右のメガネ内での絵の配置の差分に基づいてステージに配置された絵が動く（4）。このように Viscuit では、プログラムの制作画面において、絵をメガネに配置して規則を作ることによってプログラムを作成する. Viscuit の授業やワークショップでは、メガネの役割を左と右の丸の間にある「矢印」で説明している. この「矢印」は、左の丸の状態から、右の丸の状態への変化を意味している、と子どもに説明している.

Viscuit と他の教育用プログラミングツールとの違いは、ユーザに厳密な数字の入力を求めず、具体的な配置から Viscuit がその座標を読み取り、スクリーン上で絵の動きを実行するところである. 図 3.4 はテキスト型の言語としてドリトル、ブロック型の言語として Scratch、そして、図形書き換え型言語として Viscuit の命令を比較したものである. 従来のプログラミング言語ではコンピュータ上で何かの命令を実行するときには、厳密な数字に落とし込む必要がある. 例えば、斜めに絵を動かしたいときに、座標面上で毎実行時に X、または、Y が何ピクセル移動するのかを指示する必要がある. また、絵の向かう向きを角度で指定し、1 回の移動を数値で表現する必要がある. その指示がないとコンピュータは動かない. 一方で、人間が絵を斜めに動かしたい場合、多くの場合は「斜め」であり、具体的

表 2.2: Viscuit とテキストベース言語、ブロックベース言語との違い

	テキストベース言語	ブロック型言語	図形書き換え型
	ドリトル	Scratch	Viscuit
プログラム	<pre> プログラム 1 かめた=かめ!作る。 2 かめた!135 みぎまわり。 3 「かめた!10 あるく。」!10 繰り返す。 </pre>		

な数値ではない。Viscuit は前の絵の位置から「斜め」方向に移動する配置を与えるだけで、内部で座標を計算し、スクリーンで実行をしてくれる。命令を数値化する、という抽象化の部分を Viscuit がやってくれるため、ユーザーはそこに認知的な負荷を感じずにプログラムを作ることができる。


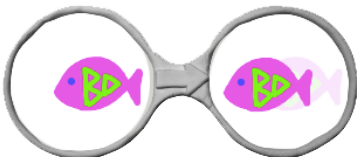

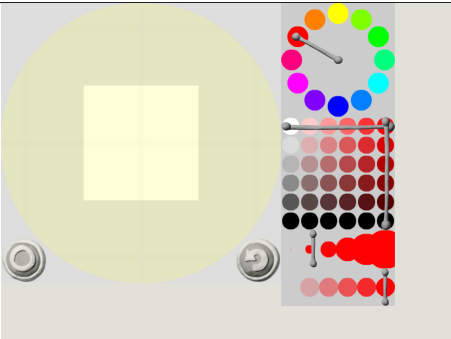
また、手続き型言語との違いとしては、命令が記述された順に実行されるのではなく、メガネで指定されたパターンがスクリーン上に見つかったときに実行される点が挙げられる。よって、ユーザーは順序立てて命令を考える必要はなく、スクリーン上で実現されてほしい変化を羅列するのみでよい。

Viscuit は表現力を落とさずにプログラム作成操作を簡単にしているところも特徴である。図 2.3 は例として ScratchJr をあげて、Viscuit の表現力の幅の広さを示す。上は絵を動かす命令の例である。ScratchJr ではブロックとして上下左右のブロックが与えられられており、それ以外の方向のブロックは用意されていない。斜めに動かすには縦方向のブロックと横方向のブロックを組み合わせ、階段のように動かすことになる。それに対して Viscuit はメガネの中の絵の配置だけで、360 度、どの方向にも動かすことができる。また、絵を配置するという動作の中で、絵のずらし具合によって移動量（速さ）を指示することができる。

また、図 2.3 において下の段は絵を描く際の色の選択画面である。ScratchJr においては色を選べる数が限られている。また、線の太さに関しても 5 段階に制限されている。一方で、Viscuit は色相環と明度彩度のバーの移動で、コンピュータで出せるすべての色を出すことができる。色を数値表現するのと同じ表現力を、Viscuit の描画画面は持っている。そして線の太さもグラデーションの中から選べる。このように、Viscuit は直感的な操作でプログラムを作れる上に、表現できるものの幅が広い特徴がある。

Viscuit において“衝突（条件分岐）”は、2 つ以上の絵をメガネの左に入れることで作ることができる。左右のメガネの丸には複数の絵を入れることができる。図 2.13 において、2 番目のメガネの左には「蕾のチューリップ」に「蝶々」が重なっているように絵が配置されている。「蕾」の絵と「蝶々」の絵はそれぞれ別に描かれている。これでステージ上で「蕾のチューリップに蝶々が重なったら」の意味になる。メガネの右には開いた「チューリップ」と「蝶々」が入っているので、こ

表 2.3: ScratchJr と Viscuit の違い

	ScratchJr	Viscuit
動きの表現		
色の表現		

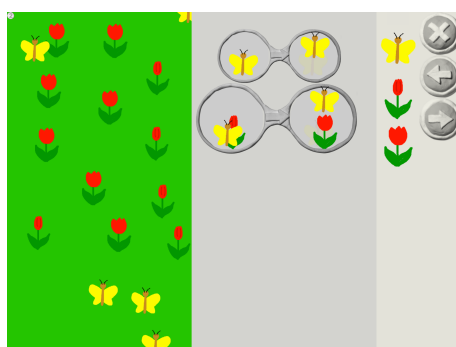


図 2.13: Viscuit における衝突のプログラムの例

のメガネの意味は「蕾のチューリップに蝶々が重なったら、チューリップは開いたチューリップになり、蝶々は上にいく」になる。ステージ上では「蕾のチューリップ」はそのままだと花は咲かないが、「蝶々」がメガネの左側のような配置になったときに、「咲いたチューリップ」の絵に変化する。また、この衝突が作動する絵の配置については厳密ではなく、メガネの左側の配置に近い配置になったときに作動するようになっている。

インタラクションを実装するための“指マーク”がある。ユーザーはこの“指マーク”を使うことによって、絵を「触った時に」という操作による条件分岐を作ることができる。図 2.14 はこの“指マーク”を使ったプログラムの例である。この例には「割れてない卵」「割れた卵」「ひよこ」の 3 つの絵が用意されている。メガネの左側には「割れてない卵」、そして、その絵に重ねるように“指マーク”が入っ

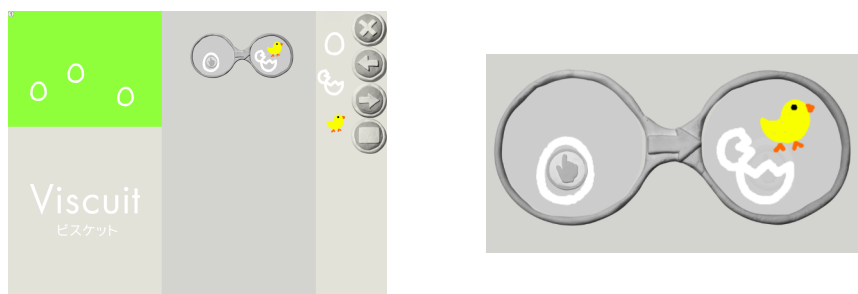


図 2.14: 上級者モードにおける Viscuit のインタラクションのプログラムの例

ている．これで「割れてない卵を触ったら」という命令になる．右側には「割れた卵」と「ひよこ」の2つの絵が入っている．これで、「割れてない卵を触ったら、卵は割れた卵に変化して、ひよこがでてくる」という命令になる．

以上から、Viscuit における、表現と意味をまとめると 2.4 のようになる．メガネにおいて「左右の絵が同じ」場合は、その絵が直線に移動する．また「左右の絵が違う」場合は、左の絵が右の絵に変化する．このとき、右の丸に絵が複数入っている場合は、左の絵が複数の絵に変化する．「左右の絵が同じ」で、絵に傾きが付いている場合、絵は回転する．また「左に複数の絵」が入っていた場合は、その左のメガネで指示された絵の配置になったときだけ、右のメガネの状態が起こる．最後に、「左に指マーク」が入っていた場合は、その場所を触ったときに、右のメガネで指示された絵が変わる．基本的にこれらの組み合わせでプログラムを作成する．よって、ユーザーは覚えることが少なく、また、これらの絵の配置でできることは直感的である．メガネの組み合わせでできることは全てが判明しているわけではなく、近年でもメガネの組み合わせでできることが発見され続けていると、開発者の原田はいう [25]．次項では、この単純なメガネの組み合わせで作られている高度な Viscuit のプログラムを確認する．

2.3.3 高度な Viscuit のプログラム

Viscuit では構文を使わず、一見、複雑なプログラムは作ることが難しく、プログラミングの入門のツールのように見える．しかし、非常に複雑なプログラムが Viscuit を使って作れることがわかっている．Viscuit の、Viscuit 以前の図形書き換え型の言語との違いはグリッドを用いない絵の配置と命令であった．一方で、Viscuit の高度な作品には頻繁にグリッドが活用されている．図 2.15 は Viscuit の設定を編集する画面である．ここでは、背景の色、ステージ上での実行の速度、絵が画面の端に到達したときに、画面上から消えるのか、または、つながっていて逆端から現れるのか、そして、グリッドの表示のオン・オフの切り替えなどの設定ができる．

図 2.16 はグリッドモードをオンにした状態の画面である．ステージ上にグリッ

表 2.4: Viscuit のメガネで表現できること

メガネ	絵の組合せ	意味
	左右の絵が同じ	直線の移動
	左右の絵が違う	絵の変化
	左右の絵が同じ（傾きあり）	絵の回転
	左に複数の絵	条件の指定
	左に指マーク	インタラクションの指定

ドが表示されているのがわかる。グリッドモードをオンにすると、絵を配置するときに、絵がグリッドに従って配置されるようになる。メガネに絵を入れるときも、グリッドが表示され、絵はグリッドに従って配置される。グリッドモードをオンにすると、グリッドに従った厳密な配置、厳密なメガネの実行ができるようになる。

合同会社デジタルポケットでは不定期に、Viscuit を使ったプログラミングコンテストを実施している。2018 年に開催された「パズル」をテーマにしたプログラミングコンテストでは、Viscuit で作られた様々なパズルが応募され、その作品はアプリ「パズルコレクション」としてリリースされている [19]。

それぞれのパズルの中では、新しいパズルが発明されたり、大量のメガネを使うことで複雑な画面の遷移や、映像表現が実装されたりしている。例えば、「猫にコイン」（図 2.17）はこのコンテストの特別賞を受賞したプログラムの一つである。このパズルは両替をテーマに作られており、落ちてくる硬貨を集めることで、より大きい金額の硬貨に両替をし、目標である金額を貯める内容になっている。このプログラムにメガネは 400 個以上使われている（図 2.17 右）。主に 5 つの硬貨が並んだ時に、1 つの硬貨になるメガネの実装に、多くのメガネが使われている。例えば、1 円が 5 枚で 5 円になる、というような場合の全ての並びが、1 円、10 円、100 円に作られている。それぞれ約 60 通りになるが、これは、アルゴリズムで書いたほうが楽なのか、それともパターンを列挙したほうが早いのかについては、どち

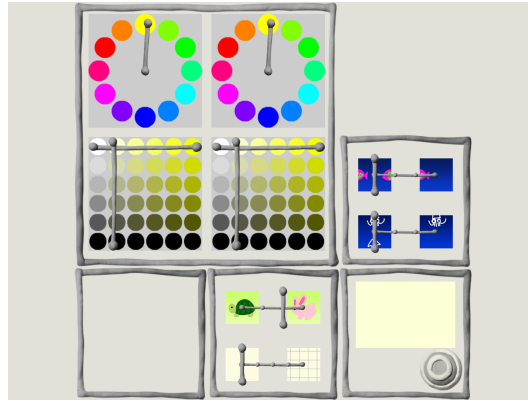


図 2.15: Viscuit の設定画面

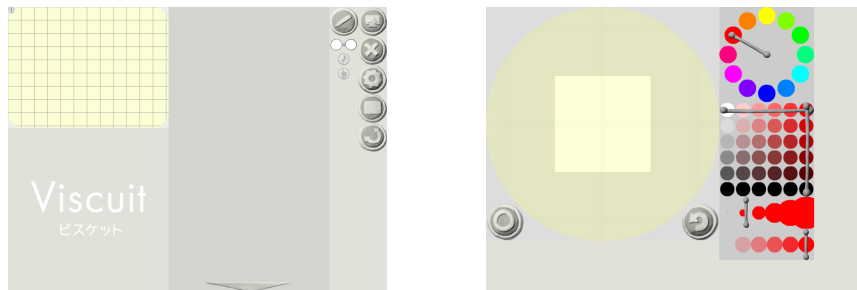


図 2.16: グリッドモードの Viscuit

らがよい手法であるとは言い切れない面がある．このプログラムから，ユーザーは少なくとも，複雑なアルゴリズムを知らなくても，1つ1つメガネを増やしていくだけで，複雑なゲームを作ることができることがわかる．

また，本研究で対象としている香川富士見丘幼稚園は，卒園した園児のうち，希望者が定期的に幼稚園に通い，Viscuit の作品作りを学んでいる．2015 年に，幼稚園において Viscuit を使ったプログラミングレッスンを開始した．その当時のレッスンを受講した年長は，現在，小学 5 年生になり，今も Viscuit を学び続けている．筆者は 2019 年に開催された第 4 回全国小中学生プログラミング大会に応募された，本幼稚園のプログラミングレッスンに通う小学生の作品を分析した [31][103]．そ



図 2.17: 左：猫にコインの画面，右：猫にコインのメガネの一部



図 2.18: 左：初期画面・右：矢が放たれる確率の調整の一部

の中で、児童が Viscuit を使って高度なプログラムを作っていることがわかった。

図 2.18 右は当時小学 4 年生だった児童が作ったパズルゲームのメガネの一部である。絵は 49 個、メガネは 156 個使われていた。このパズルはいわゆる落ちゲーで、同種の食べ物の絵が 3 つ並んでいるところをタッチすると得点になる。図 2.18 の左がゲームの画面である。黒いモンスターが画面左上から、徐々にパズル盤面に近づいてくる。パズルの盤面をモンスターに食べられると、ゲームオーバーとなる。よって、食べられる前に高得点を出し、クリアをしなければならない。

この児童のプログラムの特色は、メガネを大量に作ることで確率をコントロールしているところである。モンスターが進行するメガネの他に、モンスターが止まっているメガネを大量に作ることで、モンスターの速度を調整している。また、画面右下の弓矢が時々矢を放ち、その矢が画面左下の反射板にあたり、モンスターを攻撃し、モンスターの速度を調整するようになっている。図 2.18 の右を見ると、上から 2 番目が矢を放っているメガネだが、他のメガネは 1 つの絵がそのまま止まっていることを表している。この弓矢を発する仕組みは、全部で 25 個のメガネが使われている。そのうち 24 個が弓矢が変化しない命令になっているため、 $1/25$ の確率で弓矢が発せられるようになっている。このように任意の動作に対して、「実行されるメガネ」と「実行されない（変化のない）メガネ」の数で確率を調整している。

また、図 2.19 は別の当時小学 4 年生だった児童の作品である。このプログラム

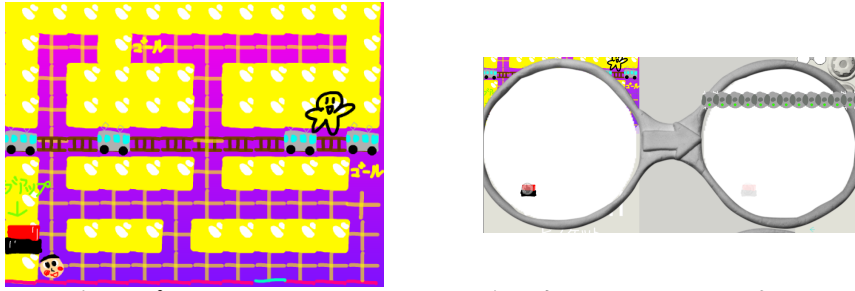


図 2.19: 左：プレイ画面・右：ステージを変えるルンバを表示するメガネ

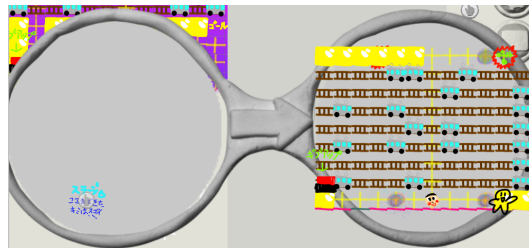


図 2.20: 次のステージを表示するメガネ

では絵が87個，メガネは119個使われていた．このプログラムは，電車でぶつからないようにゴールにたどり着く迷路ゲームである．このゲームは6ステージ用意されている．1ステージクリア，もしくはギブアップすると画面上部からルンバが横一列に並んで降りてきて，画面に配置された前のステージの絵を片付ける（図 2.19）．ルンバはどの絵にぶつかっても下に向かい，ぶつかった絵を削除するプログラムがされている．画面下部に並べられた横棒でステージを認識していて，掃除機が画面を掃除し終わると，画面下部の棒の色が変わり，次のステージにいくボタンが表示されるようになっている．そのボタンを押すと次のステージが表示される．図 2.20 で示すメガネがそのメガネである．メガネの左側が「次のステージに進むボタンを押したら」であり，右のメガネには出現するステージの絵が，その配置どおりに並べられて作られている．つまり，メガネの右側にステージを再現している．

このように，Viscuit を用いても複雑なパズルやゲームが作れることがわかって

2.4 先行研究のまとめ

本章では，子どもとプログラミングに関する研究を「プログラミングツール」，「学ぶ概念」，そして，「未就学児のプログラミングの理解」の3つの視点から確認した．また，図形書き換え型のプログラミング言語について，どのような研究的な歴史があるかを確認した上で，Viscuit について詳しく述べた．

「プログラミングツール」では Viscuit を含め，1960 年代後半から開発が始まっ

た様々なツールを論じた。ロボットを動かすもの、また、ブロックの配置と並びによって命令をするものなど様々なツールを確認した。

その上で「学ぶべき概念」では、コンピュータサイエンスをどのような切り口で考え、子どもに教えるべきなのかの議論について論じた。1960年代の子ども向けプログラミングツールの黎明期では、コンピュータは抽象的な概念を具体的にするために活用できると指摘されていた。一方でCTという概念が広く現在は普及し、人間が思考を抽象化することが重要視されていることを確認した。CTをプロセスととらえ、そのプロセスを通過できるのであれば、複雑なコーディングは必要ない、という考えもあった。CTとは別に、学んだあとの子ども達の未来に対する変化が重要である、という指摘も確認した。

そして、第1.1.3項では、現在報告されている未就学児のプログラミングの理解に関する研究を確認した。それらの研究では、ロボットを使ったプログラミングの実践におけるものが多かった。その中で、未就学児がプログラミングをすることができていたことを確認した上で、発達の段階によって、進度が大きく異なることが報告されていた。また、小学生に対する研究からも、条件分岐のプログラムは難しいことがわかってることを確認した。

その次に、図形書き換え型のプログラミング言語の歴史を確認した。図形書き換え型言語が、専門家のための複雑な言語を目指すのではなく、初学者や一般人が、どのように簡単にプログラミングを体験することができるかの試行錯誤の歴史であった。そして、それらの歴史を踏まえた上でViscuitについて確認した。Viscuitは文字も数字も使わず、メガネの左と右で、絵の前の状態と後の状態を指定することで、プログラムを作る非常に簡単なものであった。一方で、単純な作品に留まるのではなく、複雑なプログラムも作ることができることを確認した。

第2.2節を踏まえると、現在ではCS教育の重要性の上で、抽象的な思考がプログラミング教育で主張されている。一方で、その研究の歴史の原点であるPapertは、コンピュータによって、今まで学ぶことが難しかった概念を具体的に理解することができるようになり、子どもそれぞれが特性を活かして学ぶ未来を描いていた。ここで、図2.21はViscuitを使ったプログラミングのプロセスをReppeningのCTPに落とし込んだものである。Viscuitはこれまで抽象的で難しかった命令を、未就学児でもできる簡単な命令に変える。そこでは1、具体的な思考で命令の仕方を考えることになる。そして、2、タブレットを使って、指で命令をプログラムし、コンピュータ上で自動化させる。3、において実行された命令を児童は確認し、また、次の命令を考える。このようなプロセスを辿っていると考える。上記プロセスを数値表現を知らない未就学児でも体験することができる。

第1.1.3項でも確認したように、未就学児にとって、具体的な概念を用いるプログラミングが必須である。また、コンピュータの原理的な部分が体験できることは、未就学児にとって望まれている体験である。図2.5は先の図2.1にViscuitを加えたものである。Viscuitは具体性を持ちながら表現できるものが広い。そして、物理的制約がないため、純粋なコンピュータ上での実行を確認することができる。

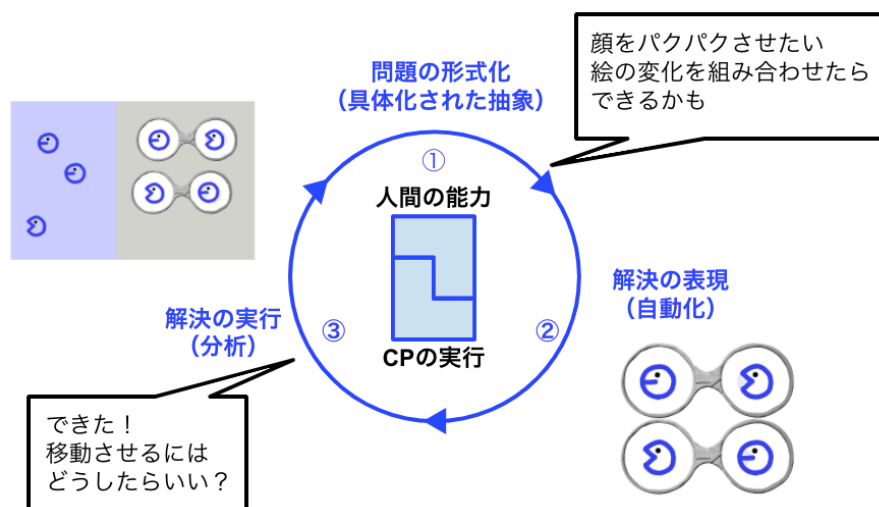


図 2.21: 具体化された抽象的な概念の CTP

表 2.5: Viscuit を含めた様々な教育用プログラミングツールの比較

大分類	中分類	言語名	表現	具体性	物理制約	拡張性
テキスト型	スクリーン	Logo	○	×	なし	○
		ドリトル	○	×	なし	○
ブロック型	スクリーン	Squeak Etoys	○	△	なし	○
		Scratch	○	△	なし	○
		ScratchJr	△	○	なし	○
	ロボット	LEGO WeDo	△	○	あり	○
		キューベット	×	○	あり	×
図形書き換え型	スクリーン	Viscuit	○	○	なし	×

一方で拡張性がないことがわかるが、本研究におけるカリキュラム作成においては支障はなかった。

情報の原理的な体験をするためには、スクリーンを用いたプログラミング体験の方が適している。また、物理的な制約がないということは、準備や用意がロボットに比べても容易である。今後さらにタブレット端末の普及することが予想され、タブレット端末上でのプログラミング体験に関する研究は必要である。

最後に Viscuit で作られた複雑な作品を見てきた。Viscuit におけるプログラミングの際の思考法と、手続き型のプログラムの思考法との間の、ユーザーの思考の違いについて考察する。図 2.22 は手続き型言語を使った CTP と Viscuit における CTP の違いの仮説である。手続き型言語の場合は、1つのプログラムの中で CTP が行われる。最終的には複数のサブルーチンを組み合わせる形になるが、その単位が大きい。一方で Viscuit のプログラムはそれぞれのメガネが完結しており、

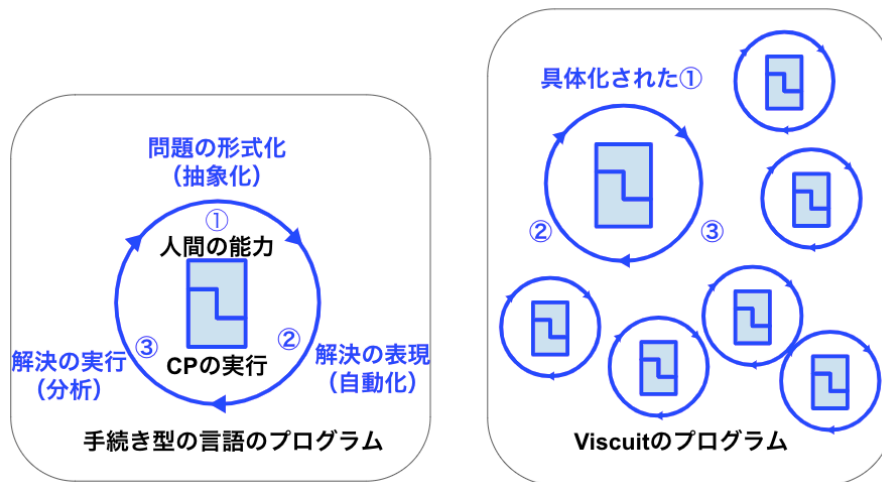


図 2.22: CTP 単位でみた手続型の言語プログラムと Viscuit のプログラムの違い

その完結したメガネの「組み合わせ」でスクリーン上で実行をさせる．これらの両者には利点と欠点がある．Viscuit 型の CTP では、複雑なものを作ることができるが、メガネの数が膨大になり、そこで描かれた絵に対してのみしか動かない．一方で、抽象的な思考はそれほど必要とせず、知っているメガネの使い方をただ組み合わせることで、コンピュータならではの面白いものを作ることができる．また、ライブラリが存在しないため、すべてを自分の手で作ることになる．そのため、作られたプログラムに関してブラックボックスになる部分はない．手続型の CTP では、抽象的で、どのような事例に対しても適用できるプログラムを、数学やアルゴリズムを利用して作ることができる．しかし、作成者は数学やアルゴリズムの理解を求められる．また、ライブラリを使う場合、そのライブラリが自分で作ったものでない場合は、プログラムの一部がブラックボックスになる場合がある．

Papert が指摘するようにコンピュータの誕生は、人間がそれまで学ぶことが難しかった概念を、具体的なものに変え、学びやすくする力ある．そして、Repening が指摘するように、人間の思考とコンピュータができることの組み合わせでする体験・過程が重要であった．日本学術会議が指摘するように、未就学児には原理的な体験が必要である．一方で、未就学児は数字を使って表現は難しい．ロボットに関しては、具体的で、プログラミングの体験は提供できるが、ロボット自体は物理的制約に縛られており、情報の原理的な体験としては濁っている可能性がある．Viscuit はスクリーン上で具体的に命令を実行することができる．そして、未就学児は自分が知っていることの組み合わせでプログラムを作ることができ、コンピュータの原理的な機能を体験することができる．また、Viscuit は具体的な命令だが、表現力が高いため、意味の探求が可能であり、未就学児の学びとしても適している．こういった観点で考えたとき、本研究で採用している Viscuit は、未就学児のプログラミングの体験には、最適なツールだと考えられる．Viscuit を使った

プログラミングで，未就学児が何を学んでいるかは，明らかにされるべきである．

第3章

研究対象の概要

3.1 はじめに

本研究は神奈川県茅ヶ崎市にある香川富士見丘幼稚園で実施した [32]。香川富士見丘幼稚園で本研究を行うようになった経緯，そして，研究の対象とした園児，研究方法について述べる。実施したレッスンについては，園児がストレスなくレッスンに取り組めるように，教室のレイアウトや，タブレットの使い方について工夫した。それら工夫を含めたレッスンの進め方についても詳しく述べる。

カリキュラムでは第2章で見てきた Viscuit を使用した。Viscuit を使うことで，園児は数の操作では難しくてできないようなプログラムを，スクリーン上で動かし，学ぶことができる。Viscuit を使ったカリキュラムについても詳しく述べる。

そして，保存された Viscuit のプログラムをどのように分析したかについて述べる。

最後に，本研究対象の概要をまとめる。

3.2 香川富士見丘幼稚園

3.2.1 経緯

著者が所属する合同会社デジタルポケットは，Viscuit を使った指導者育成講習「Viscuit ファシリテータ講習」を 2015 年より定期的に実施している [20]。この講習では Viscuit を使って，プログラミングの何を，どう，どのように教えるのかを伝えている。2015 年 4 月に，香川富士見丘幼稚園の園長先生がこの講習に参加した。その後，合同会社デジタルポケットが幼稚園でのプログラミングレッスンの実施の依頼を受けた。

合同会社デジタルポケットと幼稚園で打ち合わせを重ね，2015 年度中に実験的に，有志の参加者を募ったワークショップを数回実施した。2016 年度より，年長

の全クラスと、卒園した小学生のうちの希望者に、レッスンを行うようになった。2016年度は全てのレッスンを合同会社デジタルポケットが行った。2017年度より、幼稚園の教諭が年長のクラスのレッスンを担当するようになった。

放課後の希望者のレッスンについては、2017年度は2年生クラス、2018年度は3年生クラスも、2019年度は4年生のクラスも開講した。現在では1年生から5年生のクラスが開講され、全てのレッスンを幼稚園の教諭が担当している [103]。

実践の開始当初は、機材は合同会社デジタルポケットが持ち込んでいた。しかし、2016年に園がiPad miniを30台以上購入し、Wi-Fi環境も整えた。現在は、レッスンは全て園の設備を使って実施されている。

3.2.2 研究対象と方法

本研究では2017年度の年長（5，6歳）58名（29名2クラス）を分析の対象としている。各レッスンで作成され、保存されたプログラムを分析した。そして、授業者の教授方法とクラスの様子をビデオで撮影し、分析の参考にした。幼稚園を通して、全ての園児の保護者の方々には研究の承諾をいただいている。

プログラミングの理解について、著者は園児から文字や発話を通して回答を得ることは難しいと考えた。そこで、園児が作ったプログラム自体の分析を行った。プログラムの保存のされ方については、第3.4節で詳しく述べる。最終レッスンでは、園児の描いた絵も詳細に確認し、絵とプログラムを関連させて、それぞれの表現が効果的か、また、妥当かどうかを分析した。

幼稚園の作成した本レッスンに関する保育日誌についても、幼稚園の先生の視点からのプログラミングレッスンの見え方として、適宜参照した。

3.2.3 教室レイアウトとレッスンの進め方

図3.2がレッスンを行った会場のレイアウトである。教室を「教えるゾーン」(Z1)と「自由課題ゾーン」(Z2)に分けている。

授業者はレッスンの内容を教授するとき、園児をZ1に集める(図3.3)。その際、タブレットはZ2に固定して置き、Z1に持ってこさせないようにする。つまりZ1では、園児が指導者の話を聞くことしかできない状態にする。これによって、タブレットが目の前にあると集中ができない園児も、授業者の話を集中して聞くことができるようになる。

授業者は園児を、一つの操作法を教えるたびに、Z1とZ2を行ったり来たりさせる。この行ったり来たりの教授法は、その日の課題の「練習課題」の時間に行う。「自由課題」のときはZ2から移動させることはしない。「自由課題」のときは、隣の園児と話をし、お互いの作品を見合いながら、園児が発想を広げられるようにした。

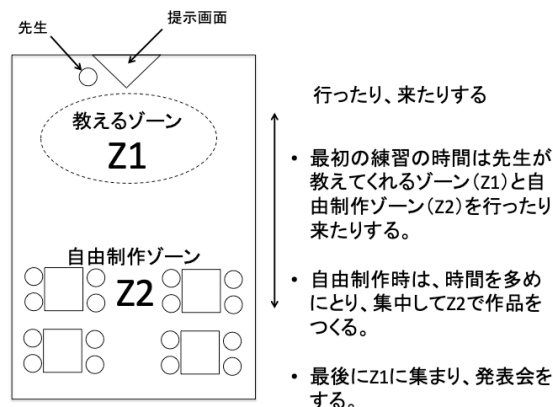


図 3.1: 会場のレイアウト図



図 3.2: 園児を前に集めている様子

場所の移動の回数は、教えることの難易度によって変えた。例えば、初回のレッスン (L1) では、最初に園児を Z1 に集め「絵を 3 つステージに置く」「メガネを出す」「メガネの両方に絵をいれる」のそれぞれの作業で園児を Z2 と先生がいる Z1 を行き来させた。こうすることによって、先生が伝えたいことの 1 つ 1 つを園児が聞き、実行できるようにした。この教授法は、合同会社デジタルポケットのファシリテータ講習の中で受講者に教えているものである。練習課題においてすでに学習した内容であれば、Z1 で先生が 1 つの絵の動きを見せたあとは「それぞれの絵にあった動きをつけてきてください」という指示しかない場合もあった。

1 レッソンは 40 分である (表 3.1)。前半に練習課題を 2 つ、または、3 つ実施し、その内容に基づく自由課題が続く。最後にグループで作成した作品の発表会を行う (図 3.1)。練習課題では、筆者らが用意した絵を用いてプログラムを制作させた。後半の自由課題では、園児自身に絵を描かせてプログラムを作成させた。

レッスンの実施方法は一斉授業よりもワークショップに近い形式をとった [24][33]。一方的に先生が教え、園児が決まった 1 つの答えに辿りつくのではなく、テーマ・課題から、園児自身の中にある発想や、創造性が引き出されることを目指して設計した。園児同士が相互に教えあい創発を促す仕掛けや、教えるのではなく園児が発見できるような仕掛けを取り入れた。

レッスンは著者、プログラミングを教える幼稚園の教諭、そして、各クラスの担

表 3.1: レッスンの時間配分

1 レッスン	時間	内容	場所
	5-10 分	練習 1	Z1 / Z2
	5-10 分	練習 2, 3	Z1 / Z2
	20 分	自由課題	Z2
	5 分	発表会	Z1

任の教諭の三人で基本的に行った。プログラミングを教える幼稚園の教諭は、園児を前に集め、集めた園児にタブレット端末でレッスン中にすべきことを見せ、何をそのレッスンですべきかを指示した。この教諭の指示でレッスンは進行した。著者は園児のプログラミングをしている様子の撮影と録画、また、レッスン中に起きる機材のトラブル、Viscuit のアプリケーションのトラブルに対処した（例えば、充電がなくなっている端末が出てきたり、園児が誤った操作をしたことにより画面が変わってしまったり、など）。担任の教諭は園児全体に目を配ったり、前の取り組みでトラブルがあった場合は、該当する園児をケアしたりする役割をした。

以上が主な役割分担だが、レッスン中は様々な園児から自分のプログラムについて「見て、見て」と声がかかり、声をかけられたら適宜それぞれが園児に対応した。また、授業者の教諭も、担任の教諭も、ともに Viscuit のファシリテータ講習を受講している。講習では「子どもの発見を奪わない」ことの重要性を教えており、園児がプログラミングに取り組んでいる最中に、授業者が答えを言わないことを指導している。園のレッスンでもそれを大事にしている。

また、各レッスンではその回によっては、映像を録画するためのアシスタントや、保育系大学からの幼稚園への実習生や、見学者が入ることがあった。

3.3 カリキュラム内容

2017 年のレッスンにおける実施カリキュラムは、著者と、授業者である幼稚園の教諭とで相談しながら作成した。2016 年に実施したものを基本的に踏襲している。

本カリキュラムでは Viscuit を採用している。採用している理由は、第 2.3 節で述べたように、具体的な命令だからである。そして、具体的なものにも関わらず、表現できる幅が広いからである。また、Viscuit 自体は他の知育アプリのように、アプリからのユーザーの操作への意味づけがない。一般的な知育アプリは、ユーザーの操作が正しいと、その正しかったことがわかるように音が鳴ったり、その操作を承認するような画面の変遷がおこる。一方で Viscuit はユーザーの操作に対する反応はなく、ユーザーはアプリからの意味づけに左右されずに、プログラムを作ることができる。これによって第 1.1.3 項で確認した、自分で探る活動が可能だと考えた。

1 年間で 13 回のレッスンを開催した（表 3.2）。2016 年にレッスンを実施し、園

表 3.2: 2017 年度に実施したレッスンの内容

番号	プログラムの使い方	テーマ	実施日
L1	直線の動き	絵を動かす	5/11
L2	直線の動き	絵を動かす	5/25
L3	直線の動き	速さを意識して動かす	6/8
L4	直線の動き	方向を意識して動かす	6/22
L5	ランダムな動き	ゆらゆら動かす	7/13
L6	ランダムな動き	ゆらゆら動かす	10/26
L7	絵を変化させる	2つの絵の変化	11/9
L8	絵を変化させる	2つ以上の絵の変化	11/30
L9	絵を変化させる	アニメーションにする	12/14
L10	絵を回転させる	絵を回転させる	1/11
L11	絵を回転させる	その場で絵を回転	1/18
L12	絵を回転させる	大きく絵を回転	1/25
L13	総合	自由につくる	2/8

児は毎回レベルを上げなくとも、練習の絵の見立てを変えるだけで十分楽しめることがわかった。よって、レッスンでは毎回レベルを上げるのではなく、難易度によっては繰り返し絵を変えて同じ内容のレッスンを実施した。例えば、5月11日と5月25日のレッスンの練習内容はほとんど変わらない。しかし、練習の絵が海のもので空のもので違うだけで、園児は十分に楽しんで練習をすることができた（表 3.3）。この方法は、重要な部分の学習の定着にも有効であると考えられる。また、この方法は、できるだけスモールステップで進めることによって、園児の中で分からない園児を出さないようにする意図もあった。今後、全ての子どもがプログラミングを理解する必要がでてくると考えられるため、特定の理解の早い園児を対象にするのではなく、全ての園児が理解できるカリキュラムを意識した。

レッスンでは、大きく分けて Viscuit の 4 つのテクニックを園児は学んだ。ここではレッスンの番号を表 3.2 を参照して表す。L1-L4 では“直線の動き”を学んだ。ここでは、絵を動かすことと、絵の持った方向性と一致させるように動かすこと、そして、動きの速さについて学んだ。図 3.4 左上の（1）の、魚が入っているメガネにおいて、右のメガネの丸に薄く魚の尻尾の部分が見えている。これが左の丸の中の魚の位置を右の丸の中で示したものである。このメガネで、薄い魚の位置よりも、濃い魚（これがユーザーが絵の置き場から直接配置した絵である）が左方向にずれた分だけ、絵がステージ上で 1 度の読み込みごとに動く。円の中における左側の絵の位置から、相対的に右側の絵がどれくらい離れているかが動きのスピードを決める。そのズレが大きくなればなるほどスピードは速くなる。

L5-L6 では“ランダムな動き”を学んだ。ここでは、同じ絵から始まるメガネを 2 つ以上作ることによって、絵をランダムに動かすことを学んだ。左の丸の中の絵

表 3.3: 練習課題に用意した絵と自由課題のテーマ

番号	練習 1 の絵	練習 2 の絵	練習 3 の絵	自由課題
L1	三角	海の生き物	-	海
L2	円	空のもの	-	空
L3	かたつむり・うさぎ	さかな	-	草原
L4	横に動くもの	縦に動くもの	-	横・縦の世界
L5	おばけ	カニ・エビ	-	七夕
L6	くらげ	海の生き物	-	ハロウィン
L7	りんご	電球	太陽と月	2 つの絵の変化
L8	種・芽・花	しりとり	-	3 つの絵の変化
L9	パクパク	パク	-	棒人間ダンス
L10	渦巻き	いろんなもの	-	ぐるぐるスーパー
L11	星雲	風車	-	オリジナル風車
L12	ロケット・星	流れ星	-	流れるなにか

が同一なメガネを複数作ることによって“ランダムな動き”を作ることができる。図 3.4 右上の (2) では「おばけ」の絵が入っているメガネが 2 つ表示されている。それぞれ右側のメガネの中の「おばけ」は、上と下にずれているメガネである。この場合は上に 50%，下に 50% の確率で動くことによって、「おばけ」はゆらゆら動く。どちらか 1 つのメガネが実行されるのではなく、同じ絵が左に入っているメガネはすべて等確率で実行される。左の丸の中の絵が、同一のメガネで始まるメガネを増やしていくと、増やしていくごとに、それぞれのメガネが実行される確率は下がる。

L7-L9 では“絵の変化の繰り返し”を学んだ。メガネの左と右に違う絵を入れることで“絵の変化”をプログラムできる。ここでは、絵を違う絵に変化させること、また、変化した後には元の絵に戻ることを通して、“絵の変化の繰り返し”を学んだ。その上で、3 つ以上の絵を用いた繰り返しを学んだ。図 3.4 左下の (3) のメガネは「口が閉じた顔」が「口が開いた顔」になる変化を繰り返す、という意味になる。ステージ上に左のメガネの中にある絵があった場合、その絵を右のメガネの絵に変更する。上が「閉じた口の絵があったら、開いた口の絵になる」、下が「開いた口の絵があったら、閉じた口の絵になる」のメガネである。この 2 つのメガネで、ステージ上では口を開けて、閉じてを繰り返すアニメーションのような動きになる。

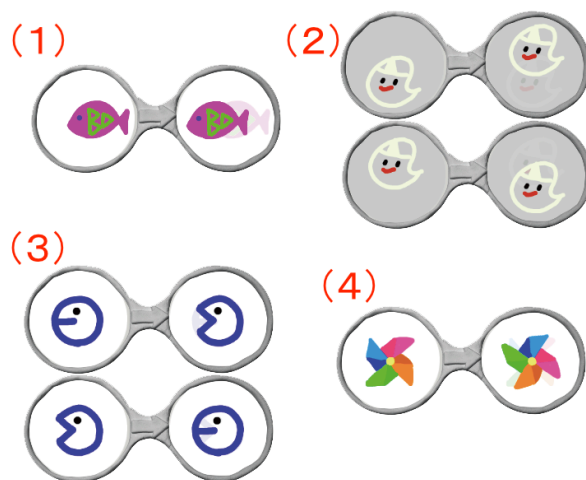


図 3.3: Viscuit におけるプログラムの例

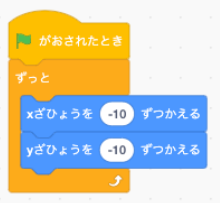


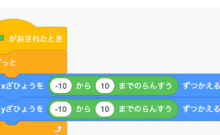

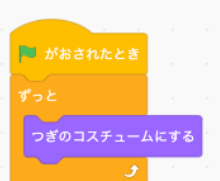
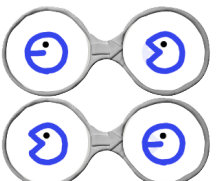



L10-L12 では“回転の動き”を学んだ．右側の絵を傾けることで，絵を回転させることができる．ここでは，メガネに絵を入れるときに，絵に傾きをつけることによって，遅い回転，早い回転，また，小さい回転，大きい回転などが作れることを学んだ．図 3.4 右下の（4）のメガネは，右の丸の中の「風車」が左の丸の中のものに比べて，右に傾いている．この場合，絵は右に回転する．傾きの大きさが絵の回転の速さになる．絵の中心が右と左で同一であれば，その場で回転する．絵の中心がずれている場合は，そのずれと傾きがステージ上で再現される．

最後のレッスンである L13 では，園児は学んだプログラミングテクニックを使って，卒園式で演じるオペレッタを紹介するプログラミング作品を，保護者のために作成した．

これらのレッスンの順番の決め方は下記のように決めた．まず最初に“直線の動き”はメガネ 1 つと絵が 1 つでできる命令であり，最初に実施した．その次に“直線の動き”を 2 個以上組み合わせることで実行できる“ランダムな動き”の実施とした．そして，メガネに同じ絵ではなく，違う絵を入れる，という新しい要素をもった“絵の変化の繰り返し”を実施した．最後に“回転の動き”を実施した．“回転の動き”はメガネ，絵ともに 1 つで作成することができるが，絵が回転することはインパクトが強く，“回転の動き”を“ランダムな動き”や“絵の変化の繰り返し”の前に実施すると，園児のモチベーションが“回転の動き”に引きづられる可能性が考えられたため，最後とした．

また，それぞれのレッスンにおけるテクニックは，ブロック型の言語と比べた場合，表 3.4 となる．Scratch と比べた場合，それぞれの命令には数字表現や，乱数といった抽象概念が使われることがわかる．“直線の動き”“ランダムな動き”では移動距離・場所を数値で指定する必要がある．また，“回転の動き”では角度で絵の回転する傾きを指定する必要がある．ScratchJr と比べた場合，“直線の動き”“回転の動き”に関しては，ブロックを増やすことで移動距離，回転の傾きを命令す

表 3.4: カリキュラム内の Viscuit プログラムの多言語との比較

	Scratch	ScratchJr	Viscuit
直線の動き			
ランダムの動き		実装なし	
変化の繰り返し		実装なし	
回転の動き			

ることができる。一方で、ブロックが増えることで、命令自体が長くなる。また、ScratchJr に関してはランダム、そして、絵自体を他の絵に変えるブロックの実装はない。

それぞれの練習課題に現れる絵には、課題に沿ったプログラムを作りたくなる絵を用意するように配慮した（表 3.3）。例えば、「ランダムの動き」では、練習 1 において「おばけ」の絵を用意した。これは、園児にそれまでのまっすぐ動く絵に対して、「おばけ」の絵はゆらゆら動かしたい、と思ってもらうためである。また、「絵の変化」のところでは、練習 1 において「りんご」の絵と「（芯だけになった）食べられたりんご」を用意した。これは、園児に「りんご」の絵を食べられた状態に変化させたい、と思わせるためである。

自由課題の部分では、園児達は自分でその課題に適したアイデアを考え、絵を描き、作品を作った（表 3.3）。練習課題において体験したことを理解していない限り、自分自身の作品を作ることが難しいと考えた。第 4 章、第 5 章、第 6 章の分



図 3.4: ビスケットランドを使った発表会の様子

析では練習課題と自由課題の両方のプログラムを分析する．自由課題では園児の自発性を尊重し，テーマは提示するが，そのテーマに沿わないプログラムを作っている園児がいても，それを禁じたり，咎めたりすることはしなかった．

先行研究では“条件分岐”は小学生でも難しいという報告があった [6]．著者は2016年のレッスン実施時に，小学1年生のクラスにおいて，Viscuitにおける“条件分岐”である“衝突”のレッスンを実施した．結果として，先行研究と同じように，多くの小学1年生が“衝突”の概念を理解することに困難を示していた．本レッスンでは，園児全員がプログラミングを楽しみと感じ，理解し，プログラムで表現ができるようになることを目指した．よって，今回の園児のレッスンには“衝突”の概念を教えるレッスンは含まなかった．

Viscuitには二つのモードが用意されている．“初心者モード”と“上級者モード”である．“初心者モード”では，メガネの右と左で絵の数が同じ場合のみ，メガネの命令が実行される．子どもや初心者にとって，絵が増えたり減ったりする現象は，物理現象とは異なるため難しい．例えば，物理法則ではボールは強く蹴れば遠くに飛ぶし，弱く蹴れば近くまでしか飛ばない．小さい子どもはこの法則でメガネが動いていると勘違いし，絵をメガネにたくさん入れると，何かすごいことが起こると勘違いする場合がある．また，そういった勘違いではなくても，偶然に1つの種類の絵を左に一つ，右に二つ入れたとする．そうすると絵は指数関数的に増える．その現象がおもしろくなってしまうと，左から右への変化がステージで作動しているということに集中できず，絵が増えることが面白くなってしまう場合がある．

それゆえ，“初心者モード”を利用した方が，メガネにおける絵の変化の役割について理解するのが簡単である．また，“初心者モード”では，自分のプログラムを保存することで，ある任意の閉じたグループ内で，画面に保存されたプログラムを一斉に表示させる“ビスケットランド”という機能が利用できる（図3.1）．幼稚園児を対象にした本カリキュラムでは，この機能を毎回，園児の自由課題の最後の発表会に利用した．

本研究で実施しているレッスンは、コンピュータの原理的な面として以下の3点を体験できている。これらは合同会社デジタルポケットが小学生向けに行っている、5つの授業の一番最初の授業で小学生が学ぶ部分である [94]。

1. プログラムを作ることによって絵が動く
2. 1つの命令でスクリーンにある複数の絵が動く
3. 間違ってもプログラムを作っても、コンピュータはその通り実行する。

上記のことはレッスン中に口頭では園児に伝えていない。しかし、体験を通して感じてもらえていると考えている。Viscuit は具体的な命令だが、2番は抽象化に通じる部分である。もし、園児が適切に Viscuit のプログラミングを理解していれば、上記の3つに照らし合わせれば下記のような行動をした場合、情報の原理的な面を理解したと考えられる。

1. メガネを作成できる
2. メガネの数は最小限になる
3. 絵に対して妥当な動きをつける

以上が、本研究で用いたカリキュラムである。

3.4 プログラムの収集と分析

Viscuit で作られたプログラムは、全て JSON 形式で作品ファイルとして保存される。保存は以下のタイミングで行われる。

- 「保存」ボタンを押したとき
- 画面の切り替えをするとき

各レッスンの練習課題は練習1, 2, または、3に対応した教材が Viscuit の画面上に用意されている。レッスンを進行する際は、その都度、教材を切り替えて園児はプログラムを作成した。切り替える操作は園児が行った。画面を切り替える操作をしたときに、園児のプログラムは保存がされるようになっている。自由課題であれば、園児が共有画面（ビスクランド）にプログラムを送るために「保存」ボタンを押したときに、保存がなされる。自由課題では保存をすると画面が新しくなるので、また新しい作品に取り掛かる。

園児は練習課題において、用意された絵を用いて1つのプログラムしか作ることができない。自由課題においては、それぞれのペースで複数のプログラムを作ることができる。園児の製作過程における画面のデータは取得していない。本研究では園児のプログラムの最終状態を分析している。

作品ファイルには、どの端末で作成されたかを示すタブレットのIDが付与されている。どのタブレットをどの園児が使っているかを予め記録し、そのIDを元に

表 3.5: 分析のために注目した json ファイルの情報

対象	わかること	データ
ファイル	プログラムが 作成された様子	ステージに置かれた絵の数 描かれた絵の数 使われた技の数
メガネ	ルールのタイプ	左右の絵の違い 回転しているかどうか 同じ絵からはじまるメガネの数
	スピード 方向	メガネの中で絵の置かれた座標

どの園児のプログラムなのかを判断した。園児には最初のレッスンから最後のレッスンまで、同じタブレットを使用させた。

表 3.5 は作品ファイルから取り出せる情報の種類である。L1-L4 では、作られたメガネの右と左の座標位置の違いから、速さと向きを調べ、分析を行った。L5-L6 では、同じ絵から始まるメガネが複数ある作品ファイルを抽出し、分析を行った。L7-L9 では、任意の A という絵から変化が始まったとき、最後に A の絵に戻るまでのメガネの数を作品ファイルごとに分析した。L10-L12 では、作られたメガネの右と左の座標位置の違いと、絵につけられた角度から、絵が回転する半径を分析した。L13 では、絵と、それぞれの絵に付けられたメガネとを関連づけ、意味付けを行い、分析をした。

3.5 研究対象の概要のまとめ

本章では研究対象の概要について詳しく述べた。香川富士見丘幼稚園での本研究にいたった経緯を確認した上で、研究における実践のデザインを確認した。対象は 58 人の年長児であり、インタビュー調査は行わず、園児の作ったプログラムから、園児のプログラミングの理解について測定した。

また、レッスンにおいては園児がストレスなくプログラミングに取り組み、自由に探求できるようにするため工夫をした。具体的にはレイアウトを教えるコーナーと作るコーナーに分けることや、ワークショップ的にレッスンを行うことにより、園児の発想を狭めるのではなく、どんな発想も否定しない運営をした。また、レッスンでは教え込むのではなく、園児の発見を促すようにした。

具体的なレッスン内容として 1 年間 13 回のレッスンを確認した。本研究では Viscuit を採用した。第 2 章で見てきた通り、Viscuit は具体的であり、抽象的な思考を用いなくてコンピュータならではの命令が作れるからである。また、それぞれのレッスンで習得した Viscuit のテクニックが、他の言語ではどのようなものになるのかを確認した。1 年間で園児は“直線の動き”“ランダムな動き”“絵の変化の

繰り返し”“回転の動き”の4つのテクニックを課題とした。

最後に、Viscuitのプログラムの保存のされ方と、本研究ではどのように作品ファイル进行分析したかを述べた。

第4章

動きのプログラムにおける方向の理解についての分析

4.1 はじめに

本章では園児の、絵の動きのプログラムにおける方向の理解について分析する。絵の動きの方向に焦点をおいたレッスンは第1回目のレッスン（L1）から第4回目（L4）のレッスンであった。

本章のリサーチクエスションは下記である。

- 園児は絵の持っている方向を意識し、意図して正しい方向に動かせるか
- 園児は自分で絵を描き、その絵を方向を意識して動かせるか
- 園児は絵を動かすとき、絵の動きの速さを制御できるか

上記を明らかにするために、練習課題においては絵自体の方向性がはっきりした絵を用意した上で、園児にそれらの絵に動きをつけさせた。絵のもっている方向と、園児の動かした方向が一致していれば、園児は意図して正しい方向に絵を動かしているといえる。また、園児に絵を描かせた上で、「縦に動かす」「横に動かす」という課題を出し、園児がその通りの方向に動かしたかを分析した。速さに関しては、動きが速い生き物と遅い生き物の絵を用意し、それぞれの絵に速さの差をつけて動かしているかどうかを分析した。

4.2 レッスン内容と対象

表4.1にそれぞれのレッスンの参加人数と、分析の対象になった園児の数を示す。練習課題（P）では、課題を逸脱したプログラムを作るケースが見られた。その場合はその園児の理解が判断できない。そのため、授業者の提示した課題を逸脱していない園児のみを分析の対象とした。

表 4.1: 分析の対象となった園児の数

L 番号	参加人数	P 番号	対象	分析数
L1	55 人	P1	三角	40 人
		P2	さかな 1	35 人
			かに	38 人
			いか	40 人
			さかな 2	30 人
L2	52 人	P1	丸	23 人
		P2	ロケット	38 人
			しずく	38 人
			ヘリコプタ	32 人
			とり	31 人
L3	57 人	P1	うさぎ・かたつむり	31 人
L4	56 人	P1	くるま	41 人
			くも	32 人
			さかな	32 人
		P2	ロケット	46 人
			ふうせん	40 人
			しずく	40 人

逸脱した例として、2 回以上保存している場合がある。これは画面の切り替えのボタンを押したなどの理由で、同じ課題に 2 回以上取り組んだ園児である。1 つの園児に 2 つのファイルがあった場合、一方が課題にマッチしていて、一方が間違っているという状況が見られたため、分析の対象外とした。また、1 つの絵にメガネが 2 個以上使われた場合も、分析の対象外とした。これは授業者に言われたことを超えて、プログラムの実験をしていた園児である。

L1-P1, L2-P1 を除き、練習では複数の絵が用意されている（図 4.1, 図 4.2, 図 4.3, 図 4.5）。全ての絵を動かす園児もいれば、その中からいくつかを選んで動かす園児もいた。

4.3 方向に関する分析

4.3.1 レッスン 1 のプログラムの分析

前述（表 3.3）のように、レッスンは 3 つのパート、または、4 つのパートで構成されている。L1 では P1（図 4.1 左）で、方向性のない、無機質な「三角」の絵で Viscuit で絵が動かせることを学ぶ。P2（図 4.1 右）では用意された方向性がはっきりしている海の生き物を動かす。自由課題では園児一人ひとりが「うみ」にい

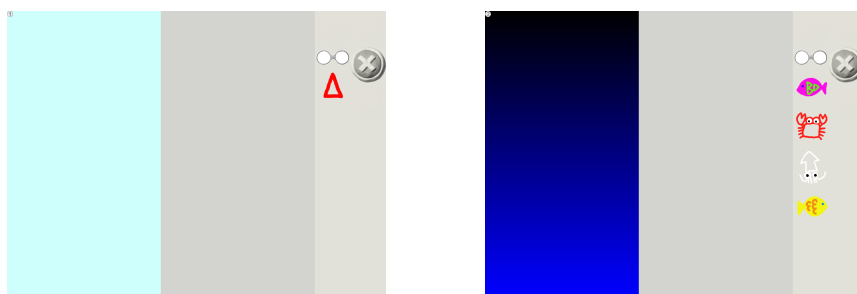


図 4.1: 左 : L1-P1 の画面, 右 : L1-P2 の画面

表 4.2: L1-P1 の分析結果

絵	方向	人数	割合	p 値
三角	上	8 人	20.00 %	0.58
	下	12 人	30.00 %	0.47
	右	12 人	30.00 %	0.47
	左	8 人	20.00 %	0.58

る生き物を考え、絵を描き、プログラムを作った。自由課題は自由度が高いため、分析の対象外とした。

P1 (図 4.1 左) は「動きをつけること」が課題である。表 4.2 を見ると、三角自体に方向性がないため、プログラムで動かす方向もほぼ均等に分かれていることがわかる。絵自体が方向性を持たない場合、園児が上下左右どの方向に動かすかには偏りが見られなかった。園児が絵をどの向きに動かすかの確率は均等だと仮定し、それぞれの絵の動きに二項検定を行った。その結果、それぞれの方向に有意な差は見られなかった。

P2 (図 4.1 右) では、4 つの海の生き物が用意されている。上からピンクの「さかな 1」は顔が左を向いているので左、「カニ」は横に動くので左右のどちらか、黄色い「さかな 2」は右に向いているので右である。これらの絵には講師側の意図が組み込まれている。一方で、「いか」については絵が持っている方向が明確ではないため、分析の対象外とした。

表 4.3 はそれぞれの課題に対して、動きがマッチしていた園児の数である。それぞれの方向から上下、または、左右に 45 度の角度までをマッチしていると考えた。p 値が 0.05 より小さいものを「*」、その中でも p 値が 0.01 より小さいものを「**」で表した。

ビデオで確認したところ、P1 の教えるところでは「三角を三つおく」「メガネをおく」「三角をメガネの両方に入れる」の手順を、ゾーンを行き来して作業をした。その際に、園児は次のことを授業者に習うために、走って前に行っていた。三角が動き出してからはいき来はせず、授業者の指示で、どのような規則で動きが決まるのかを園児たちは探索していた。園児達からは動いた喜びや、「わかった」と

表 4.3: L1-P2 の分析結果

絵	方向	人数	割合	p 値
さかな 1	上	10 人	28.57 %	0.696
	下	4 人	11.43 %	0.077
	左 (正解)	18 人	51.43 %	0.001 **
	右	3 人	8.57 %	0.029 *
かに	上	8 人	21.05 %	0.709
	下	2 人	5.26 %	0.002 **
	左 (正解の一部)	15 人	39.47 %	0.058
	右 (正解の一部)	13 人	34.21 %	0.192
	左右合計 (正解)	28 人	73.68 %	0.005 **
さかな 2	上	10 人	33.33 %	0.295
	下	2 人	6.67 %	0.019 *
	左	5 人	16.67 %	0.399
	右 (正解)	13 人	43.33 %	0.032 *

いう声がでていた。その後、授業者は園児を前に集め、メガネの機能を説明した。

P2の教える場面では、最初に授業者が「さかな 1」をタブレットで動かして見せた。そのとき「さかな 1」を下の向きに間違えて動かして園児に見せたところ、笑いとともに「ちがう」という指摘が園児からでた。その後「さかな 1」を頭の方角に動かすのを見せた後は、その他の生き物については授業者は動きのつけ方は見せず、「それぞれの生き物にあった動きをつけてください」という指示のみであった。

L1-P2において、一つ目の絵で見本を見せているが、他の絵はそれぞれ別の方向性を持っているため、園児の理解を測る上で影響になっているとは思われない。「さかな」「かに」「さかな 2」の絵について、正解の方角（「かに」については左右の合計）が有意に多く選ばれている。

4.3.2 レッスン 2 方向に関する分析

L2では、P1（図 4.2 左）で、方向性のない「丸」の絵で Viscuit で絵が動かせることを学ぶ。P2（図 4.2 右）では「そら」の世界観に存在する絵を動かす。それぞれの用意された絵の方向性は、はっきりしている。自由課題では園児一人ひとりが「そら」にいる生き物を考え、絵を描き、プログラムを作る。自由課題は自由度が高いため、分析の対象外とした。

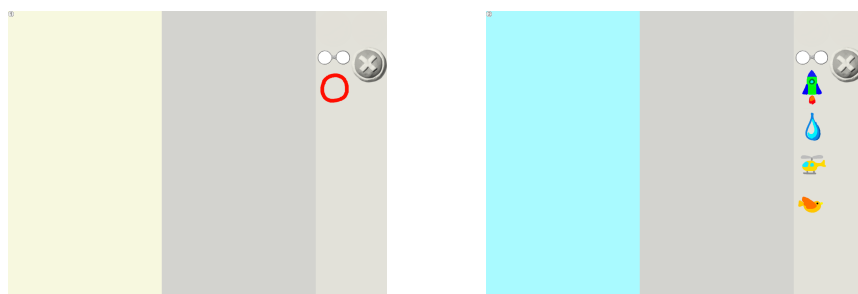


図 4.2: 左 : L2-P1 の画面, 右 : L2-P2 の画面

表 4.4: L2-P1 の分析結果

絵	方向	人数	割合	p 値
丸	上	8 人	34.78 %	0.33
	下	8 人	34.78 %	0.33
	左	3 人	13.04 %	0.23
	右	4 人	17.39 %	0.47

L1 と同じく P1 (図 4.2 左) は「動きをつけること」が課題である. 表 4.5 右を見ると, 丸自体に方向性がないため, プログラムで動かす方向に有意な偏りはないことがわかる.

P2 (図 4.2 右) では, 4 つの「そら」の世界観にある絵が用意されている. 上から「ロケット」は上を向いているので上, 「しずく」は下に落ちるので下, 「ヘリコプター」は上に飛ぶ, また, 左方向を向いているので上か左, 最後に「とり」は右に向いているので右である.

表 4.5 を見ると, 全ての絵について, 正解の方向 (「ヘリコプタ」については上左の合計) が有意に多く選ばれている. また, 間違った方向については, 比率が有意に低いことがわかる.

ビデオで確認したところ, P1 の教えるところでは, 授業者は「丸を三つおく」ところで一度園児を自分の座席に戻らせて作業をさせた. その後, 園児を教えるゾーンに集めた. そして「丸を動かすにはどうすればいいんだっけ?」と授業者が聞くと園児達は「メガネー」と答えている. その後「メガネをおく」「丸をメガネの左の丸にいれる」までを座席に戻って園児に作業をさせ, 教えるゾーンに再び園児を集めた. 授業者はメガネで絵を動かすところを見せる際, 丸の絵をメガネの右側に入れるときに「上に (前の絵よりずらして) 入れるとどうなるんだっけ?」「下にいれると?」「横にいれると?」と園児に質問をした. 園児はその都度「上」「下」「横」と答えていた. その確認が終わったあと, 園児たちは自分の座席に戻り, 自分のタブレットで丸の絵に方向をつけていた.

P1 と P2 の間に, 授業者はメガネを使った速さの換え方について, 少し話していた.

表 4.5: L2-P2 の分析結果

絵	方向	人数	割合	p 値
ロケット	上（正解）	30 人	78.95 %	< 0.001**
	下	2 人	5.26 %	0.002**
	左	6 人	15.79 %	0.2598
	右	0 人	0.00 %	< 0.001**
しずく	上	6 人	15.79 %	0.2598
	下（正解）	28 人	73.68 %	< 0.001**
	左	2 人	5.26 %	0.002**
	右	2 人	5.26 %	0.002**
ヘリコプ タ	上（正解の一部）	11 人	34.38 %	0.2234
	下	1 人	3.13 %	0.002**
	左（正解の一部）	17 人	53.13 %	< 0.001**
	右	3 人	9.38 %	0.04*
	上左合計（正解）	28 人	87.68 %	< 0.001**
とり	上	10 人	32.26 %	0.41
	下	3 人	9.68 %	0.06
	左	1 人	3.23 %	0.003**
	右（正解）	17 人	54.84 %	< 0.001**

P2の教える場面では、最初に授業者が世界観が「そら」であることを確認したあとに「ロケット」をタブレットで動かして見せた。そのとき「ロケット」を下の向きに間違えて動かして園児に見せたところ、笑いとともに「ちがう」という指摘が園児からでた。その後「ロケット」を上の方に動かすのを見せた後は、その他の絵については授業者は動きのつけ方は見せず、「それぞれの絵にあった動きをつけてください」という指示のみであった。

L2-P2において、L1-P2と同様に一つ目の絵で見本を見せているが、他の絵はそれぞれの別の方向性を持っているため、園児の理解を測る上で影響になっているとは思われない。

4.3.3 レッスン4における方向に関する分析

L4の内容は、方向にフォーカスをした練習課題と自由課題になっている。L1～L3では、「うみ」などの世界観で統一されていた絵の種類が、L4では方向で統一されている。P1（図4.3左）では「くるま」「くも」「さかな」であり、「横に動くも



図 4.3: 左 : L4-P1 の画面, 右 : L4-P2 の画面

の」の世界が用意されている。P2（図 4.3 右）では「ロケット」「ふうせん」「しずく」であり、「縦に動くもの」の世界が用意されている。

自由課題は 2 つ用意されている。1 つ目が「横の世界」。もう 1 つ目は「縦の世界」である。いままでの自由課題は世界観に合わせた絵を描き、動かすことが課題だったが、このレッスンでは動きの方向が課題になる。課題となる方向が明確なため、この L4 では自由課題も分析した。

また、ビデオで確認したところ、L4 においては、P1 の最初に 1 つだけ正しい動きを見せ、そのあとは「それぞれにあった動きをつけてください」という指示しかしていなかった。P2 においては一つのクラスでは 1 つだけ正しい動きを見せ、もう一つのクラスでは正しい動きも見せずに「それぞれにあった動きをつけてください」と指示をしていた。

L4-P1, P2 においては「よこ」「たて」と明言している。園児が絵の方向を考える上では影響がないとはいきれないが、園児たちの「よこ」と「たて」のプログラムの作り方の違いの理解は測れていると考える。これは L4 の自由課題においても同様である。

表 4.6, 表 4.7 をみると、全ての絵において、正解の方向が有意に多く選ばれている。「横の世界」の「くも」を除いて、その p 値はすべて 0.001 より低い。

自由課題では園児はそれぞれのペースでプログラムを作り、保存してビストランドに表示し、また次のプログラムに取り掛かる。よって、園児によっていくつプログラムを作るかは違った。表 4.8 は園児が作成したプログラム（園児の区別なく）全てについて課題にマッチしているプログラムの割合である。プログラム全体の本数は横の世界が 208 個、縦の世界が 224 個であった。メガネの左右に絵が入っていないものは対象外とした。横に動かす課題では、作られたプログラム全体の 7 割以上が横に動いていた。また、縦に動かす課題も同じように、7 割以上のプログラムが縦の動きになっていた。これらにおいて、正解の方向が有意に多く選ばれており、その p 値はいずれも 0.001 未満である。

園児一人ひとりに注目してみる。それぞれの園児において、作ったプログラムの中で課題にマッチしたプログラムの割合を出した。図 4.4 はその正答率に基づいてヒストグラムにしたものである。どちらの自由課題においても、100 % のところに一番大きい山が現れている。これらの園児はそれぞれ、全体の 4 割弱であった。

表 4.6: L4-P1 の分析結果

絵	方向	人数	割合	p 値
くるま	上	1 人	2.44 %	< 0.001 **
	下	2 人	4.88 %	0.002 **
	左	2 人	4.88 %	0.002 **
	右 (正解)	36 人	87.80 %	< 0.001 **
くも	上	5 人	15.63 %	0.307
	下	3 人	9.38 %	0.041 *
	左 (正解の一部)	7 人	21.88 %	0.839
	右 (正解の一部)	17 人	53.13 %	< 0.001 **
	左右合計 (正解)	24 人	75.00 %	0.007 **
さかな	上	0 人	0.00 %	< 0.001 **
	下	2 人	6.25 %	0.013 **
	左 (正解)	22 人	68.75 %	< 0.001 **
	右	8 人	25.00 %	1

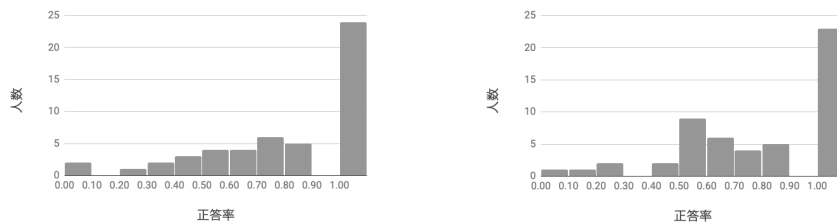


図 4.4: ヒストグラム／左：横の世界の正答率，右：縦の世界の正答率

横の世界においては 70 %～80 %に，縦の世界においては 50 %～60 %に小さい山が現れている。

4.4 速度に関する分析

実施日が前後するが，L3 (図 4.5) について述べる．L1, L2, L4 が動かす方向に焦点をおいた課題だったのに対して，L3 では速さの調整の仕方について学んだ．P1 (図 4.5) には「かたつむり」と「うさぎ」の絵が用意してあり，この 2 つを動かす．「かたつむり」は動きが遅い生き物，「うさぎ」は動きが速い生き物であるので，その差をスピードで表す．自由課題は「そうげん」をテーマに動くものを描き，動かす．自由課題は自由度が高いため，分析の対象外とした．

表 4.7: L4-P2 の分析結果

絵	方向	人数	割合	p 値
ロケット	上（正解）	38 人	82.61 %	$< 0.001^{**}$
	下	5 人	10.87 %	0.026 *
	左	0 人	0.00 %	$< 0.001^{**}$
	右	3 人	6.52 %	0.002 **
ふうせん	上（正解）	31 人	77.50 %	$< 0.001^{**}$
	下	1 人	2.50 %	$< 0.001^{**}$
	左	1 人	2.50 %	$< 0.001^{**}$
	右	7 人	17.50 %	0.361
しずく	上	4 人	10.00 %	0.028 *
	下（正解）	32 人	80.00 %	$< 0.001^{**}$
	左	1 人	2.50 %	$< 0.001^{**}$
	右	3 人	7.50 %	0.009 **

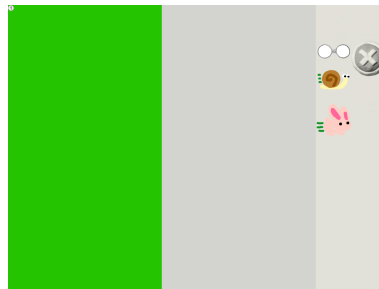


図 4.5: L3-P1 の画面

ビデオでは、授業者はまず「かたつむり」と「うさぎ」の絵を見せ「うさぎの絵はどう動く？」と尋ねた。そうすると「ぴよんぴよんする」と園児は答えていた。授業者が「かたつむりは？」と尋ねると、園児は「ゆっくり」と反応していた。その後、授業者が「かたつむり」を「うさぎ」より速く動かすと、園児から「ちがう」という反応が返ってきた。その後、授業者は「それぞれにあった動きをつけてください」と言って園児たちに課題に取り組ませていた。

授業者の指示した通りに、メガネを作っていた園児は31名だった。分析対象となった園児の中で「かたつむり」と「うさぎ」をそれぞれ遅く・速く動かしていたのは7割であった（表 4.9）。速さの差を2倍以上つけていた園児は5割だった。「かたつむり」と「うさぎ」のどちらを速くするかは、どちらかが速くなればどちらかが遅くなるため、どちらが速くなるかの確率は均等だと仮定し、2項検定をおこなった。p 値の表記については L1, L2, L4 と同じである。「うさぎ」を速く動かしている園児の数は有意に多いことがわかる。

図 4.6 は各園児が作ったメガネの「かたつむり」と「うさぎ」の速さの差のヒストグラムである。正の差は「うさぎ」の方が速いプログラムであり、負の差（黒

表 4.8: L4-自由課題の分析結果

世界	方向	数	割合	p 値
横の世界	上	31	14.03 %	< 0.001 **
	下	14	6.33 %	< 0.001 **
	右 (正解の一部)	86	38.91 %	< 0.001 **
	左 (正解の一部)	77	34.84 %	< 0.001 **
	左右合計 (正解)	163	73.75 %	< 0.001 **
縦の世界	上 (正解の一部)	129	54.66 %	< 0.001 **
	下 (正解の一部)	39	16.53 %	0.009 **
	上下合計 (正解)	168	71.19 %	< 0.001 **
	右	25	10.59 %	< 0.001 **
	左	31	13.14 %	< 0.001 **

表 4.9: L3-P1 の分析結果

速さ	人数	割合	p 値
うさぎが速い	24	75.00 %	0.003 **
2 倍以上速い	16	50.00 %	1

い枠線に囲まれたカラム)は「かたつむり」の方が速いプログラムを表す.「かたつむり」の方を速く動かしている園児が7名いるのがわかる.

また, L3における P2 の結果では, 上記 P1 に見られるよりも, 理解を見ることが難しかった. P2 では「細長い魚 (ダツ)」と「マンボウ」が用意されていた.「カタツムリ」と「うさぎ」ほど, 園児はその速さの違いを, 直感的に感じ取れなかったことが原因だと思われる.

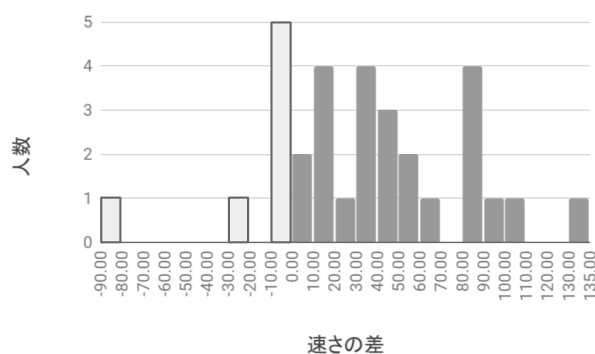


図 4.6: L3-園児の速さのヒストグラム

4.5 動きのプログラムにおける方向の理解についての分析のまとめ

L1において、園児がプログラムを作ること的成功し、また、非常に動機付けられている様子が明らかになった。これは先行研究で確認されていることと同じであった [4][9][50][63][71]。

L1-P2 から園児の作るプログラムにおいて、正解の動きが有意に多くなっている。よって、園児の多くは最初のレッスンから絵の進行方向を理解して、その方向に合うプログラムを作成しているのがわかった。

L2-P2 と L1-P2 を比べると、不正解についても有意に低いものが多くなっていることがわかる。よって、L1 に比べて、L2 のほうがより多くの園児が絵の進行方向に絵を動かしている。つまり、2 回行なったほうが園児の理解は進むと考えられる。

L4-P1, P2 においては、園児は絵から判断する以外に、授業者から「横の動き」、「縦の動き」と動きが示されている要因もあってか、特に縦の世界では多くの園児が絵の進行方向に絵を動かしているのがわかる。つまり、「横」と「縦」の動きのプログラムの作り方の違いに関しては、4 回目のレッスンに至って、大半の園児が理解している。Papadakis ら [71] の研究では、園児は上下左右のブロックを混同している様子が確認されたが、Viscuit においては具体的に絵の位置で方向を命令するためか、混同している様子は見られなかった。

L4 の自由課題においては、絵を用意せず、園児の描いた絵を用いて授業者が上下、または、左右で方向を絞ったプログラム制作を課題としていた。課題通りに横方向だけ、縦方向だけ作った園児はどちらも 4 割程度であった。また、回答者をヒストグラムにすると、すべてのプログラムが課題の方向に一致していた園児が一番多かった。そして、全く正答できなかった園児の割合は非常に低いことがわかった。よって、与えられた絵ではなく、自分の描いた絵においても、課題に沿った動きをつけている園児が一定数いたことはわかった。

以上のことから練習課題において、絵の方向がはっきりしている絵については、大半の園児が、絵の進行方向通りにプログラムできていることが明らかになった。一方で、自由課題において自分の絵を用いて、一定数の園児が与えられた方向通りに動かしている、という結果は得られたが、大半の園児が課題通りに動かしている様子は確認できなかった。

速さを課題としたレッスンでは、授業者の指示に従い、逸脱せずにプログラムを作った園児に関しては、75%の園児が、速さを使い分けられていることを確認することができた。一方で、逸脱したプログラムを作った園児に関しては本研究ではその理解度はわからなかった。

第5章

繰り返し続けるプログラムの理解についての分析

5.1 はじめに

本章では，園児がビスケットにおいて，繰り返して変化を続けるプログラムを理解し，そのプログラムを作成できるかどうかを分析した．絵の変化の繰り返しを題材にしたレッスンは第7回（L7）から第9回（L9）のレッスンである．

本章におけるリサーチクエスションは下記である．

- 園児は絵の変化の繰り返し続けるプログラムを理解できるのか
- 園児はどれくらいの長さの，繰り返し続けるプログラムを作るのか

上記を明らかにするために，ある A という絵が， $A \Rightarrow B$ のように B に変化したあとに， $B \Rightarrow A$ のように最初の絵に戻るプログラムが作られているかどうかを分析した．また，最初の絵から最後の絵に至るまで，何種類の絵を経ているかを見ることによって，園児の作る絵の繰り返しの長さを分析した．

5.2 レッスンの内容と対象

L7-9 では，Viscuit における“絵の変化”について学んだ．その上で“絵の変化”を組み合わせ“絵の変化の繰り返し”を学んだ．練習課題（P）では，時間を経て変化する様子を示す2つ以上の絵を用いて，園児はプログラムを作成した（図5.1，図5.2，図5.3，図5.4，図5.5，図5.6，図5.7）．自由課題（F）では繰り返させる絵自体も，園児が描いてプログラムを作成した．

L7 では A，B という，2つの任意の絵が交互に変化するプログラムを作成した（図5.1，図5.2，図5.3）．A と B の絵があるとき， $A \Rightarrow B$ （A を B に変化させる．

以下も同様), $B \Rightarrow A$ という命令があると A から B, B から A の変化が繰り返しおこる命令になる。

L8 では A, B, C, そして, D という 3 つ, または 4 つ以上の絵が順に変化し, その変化が繰り返すプログラムを作成した (図 5.4, 図 5.5)。例えば, $A \Rightarrow B$, $B \Rightarrow C$, $C \Rightarrow D$, $D \Rightarrow A$ という命令があると, 4 つの絵が順に変化し続ける繰り返しを作ることができる。

L9 では絵の繰り返しを活用して, ある任意のモチーフが動いて見えるようにするために, 繰り返しを使うことをテーマにした (図 5.6, 図 5.7)。例えば, まったく違った A と B の絵を使って, $A \Rightarrow B$, $B \Rightarrow A$ と繰り返しを作るのではなく, 口を開いた A1 と口を閉じた A2 の絵を $A1 \Rightarrow A2$, $A2 \Rightarrow A1$ と繰り返させると, 同一のモチーフが口をパクパクする動作を繰り返しているように見える。

それぞれのレッスンにおいて, 集計したプログラムのデータから, メガネの左右の絵が違うメガネを抽出して分析を行った。絵の変化が繰り返すようにプログラムが作れている場合であっても, 同じ命令が複数ある (例えば, $A \Rightarrow B$ という命令が重複している) 場合があった。これらの場合は, 園児が絵の変化を繰り返し続けさせるための必要なパターンを理解せずに, 適当に複数の命令を作ることによって “絵の変化の繰り返し” を, 作ることに成功している場合があると考えた。なぜなら, 繰り返しのパターンの抽出ができ, そのパターンを理解していれば, 無駄なメガネを作る理由がないと考えたからである。必要以上にメガネを作成している園児は, 望んだ結果が得られるまでランダムにメガネをたくさん作る, という戦略をとっていると考えた。よって, 繰り返しができているか, だけでなく, それぞれの繰り返し続けるプログラムが, いくつのメガネで作られているかも数えた。

5.3 練習課題の内容と分析

L7 の学習内容は「2 つの絵が交互に変化する繰り返し」であり, 園児がどうやって絵を変化させるかを学ぶ最初のレッスンだった。L7-P1, P2 では絵が変化することが自然に感じ取れる絵を準備した。

L7-P1 (図 5.1) では「りんご」の絵と「(芯だけが残っている) 食べられたりんご」を用意した。園児が「りんご」の絵が食べられる (「食べられたりんご」の絵になる) という変化を実現してみたいかなるようになっている。

L7-P2 (図 5.2) では「灯の消えている電球」と「灯のついている電球」を用意し, 園児は電気の点灯と消灯を繰り返させたいかなるようにした。この結果, 園児に $A \Rightarrow B$, $B \Rightarrow A$ の変化のパターンを自力で発見させるように促した。

最後の L7-P3 (図 5.3) では, 「太陽」と「月」の絵が用意されている。太陽の後に月がきて, 月の後に太陽が来るように, 園児は繰り返しのプログラムをこの二つの絵を使って作る。

L8 では「3 つ以上の絵が繰り返し変化し続ける組み合わせ」を作ることを目的と

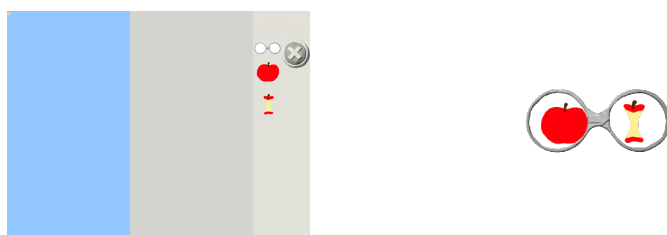


図 5.1: L7-P1 における絵と正解のプログラム

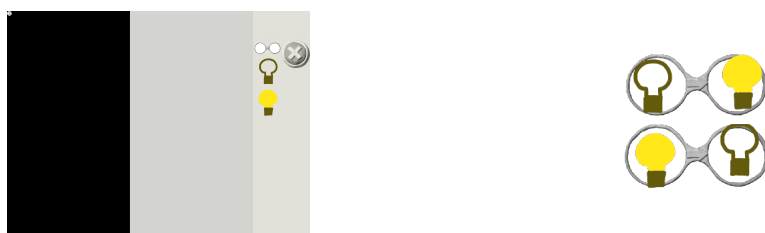


図 5.2: L7-P2 における絵と正解のプログラム

した．L8-P1 (図 5.4) では「たね」「芽」「花」の絵が用意されている．これらは植物が時間の経過に従って育つ様子を表している．L8-P2(図 5.5) では「こぶた」「たぬき」「きつね」「ねこ」の絵が用意されている．これらはしりとり遊びを参考にした．また、「こぶたぬきつねこ」という未就学児向けの歌もあり，園児にもよく知られている [104]．

L9 では「アニメーションを作る」ことを目的とした．L7，L8 では，園児は一つの絵を別の絵に変えることを学んでいた．このレッスンでは，園児は最初の絵に対して部分的に違う絵を描き，それらの絵を変化させていくことで，アニメーションを作ることを学んだ．L9-P1(図 5.6) では，園児は「口を開けている顔の絵」と「口を閉じている顔の絵」を与えられる．園児たちが二つの絵を使って，パクパクするアニメーションが作りたくなるようにした．L9-P2(図 5.7) では「口を閉じている顔の絵」だけが用意されている．園児は「口を開けた顔の絵」は自分で描かなくてはならないようにした．

それぞれの練習課題 (P) では，授業者が園児を集め，その課題に沿ったメガネを作るためにはどうしたらいいか問題を提議し，実際に操作する様子を見せた．そ

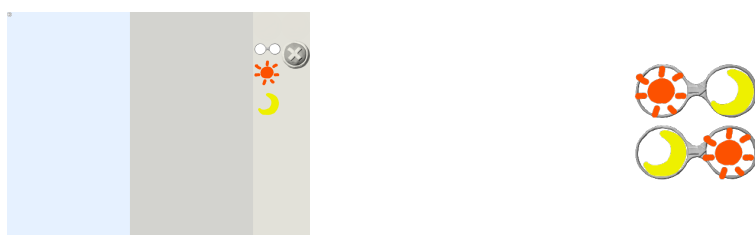


図 5.3: L7-P3 における絵と正解のプログラム

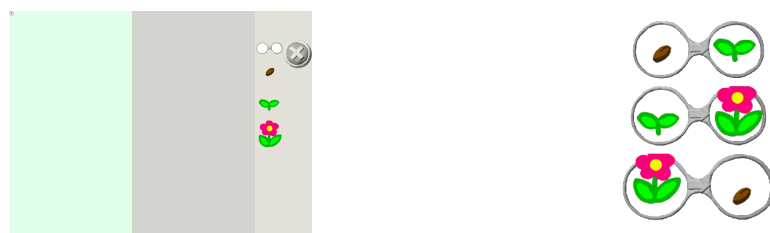


図 5.4: L8-P1 における絵と正解のプログラム

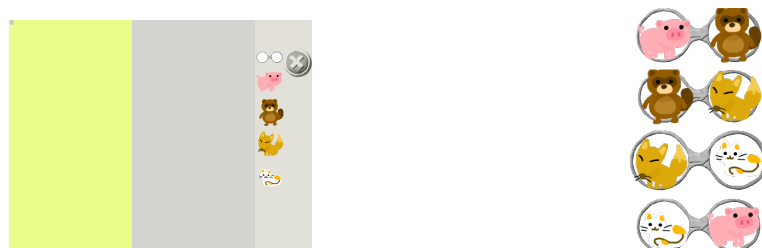


図 5.5: L8-P2 における絵と正解のプログラム

の時に授業者は、答えを見せるのではなく、答えの一步手前で操作を止めるようにした。

これらの課題で作られたプログラムを分析した結果が表 5.1 である。“N”は対象となるプログラムを保存した園児の数を表している。“正解 (A)”のカラムは、最小限のメガネの数でプログラムを作った園児の人数と割合を示している。図 5.1～図 5.7 において、最小限のメガネは図の右に示されているものである。また、“過剰 (B)”のカラムは絵の変化が繰り返されるメガネの組み合わせはできているが、余計なメガネが存在しているプログラムを作った園児の人数と割合である。“A+B”は“正解 (A)”と“過剰 (B)”の合計である。L7-P3 と L8-P1 においては、設定に不手際があったため、園児のプログラムは保存されていなかった。

全てのレッスンにおいては、80%を超える園児が、変化が繰り返されるメガネの組み合わせを含む、プログラムを作成していた (A+B)。一方で、必要最低限のメガネで課題を遂行していた園児の割合は、L9-P2 を除き 60%前後であった。

また、表 5.2 はそれぞれのレッスンにおける絵の変化の繰り返しが偶然に作成される確率と、それぞれのレッスンにおいて正しくプログラムを作成した園児の数

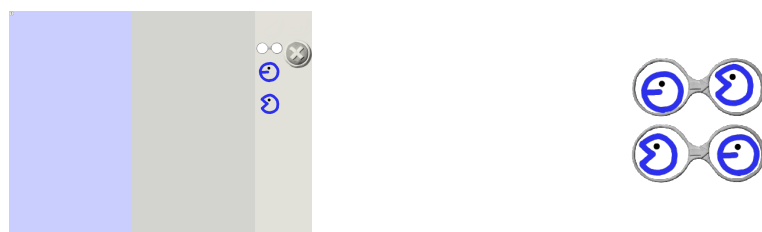


図 5.6: L9-P1 における絵と正解のプログラム

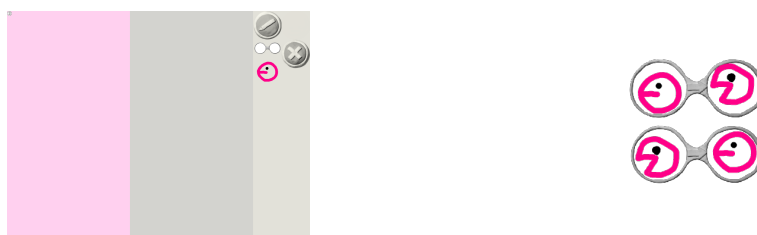


図 5.7: L9-P2 における絵と正解のプログラム

表 5.1: L7-9 の練習課題の結果

		N	正解 (A)(%)	過剰 (B)(%)	A+B(%)
L7	P1	52	33(63.46%)	9(17.31%)	42(80.77%)
	P2	53	31(58.49%)	13(24.53%)	44(83.02%)
L8	P2	56	34(60.71%)	11(19.64%)	45(80.36%)
L9	P1	49	28(57.14%)	18(32.65%)	44(89.79%)
	P2	46	24(52.17%)	14(30.43%)	38(82.61%)

をもとに 2 項検定を行った結果である．“確率”はそれぞれの繰り返しを作る際に，必要な絵を用いて組み合わせを作ったときに，正しく繰り返しの組み合わせができる確率を示している．例えば，2 つの絵の変化の繰り返しの場合，A の絵から始まる組み合わせを考えたとき，あり得る組み合わせは 8 パターンである（「 $A \Rightarrow B$ 」 $A \Rightarrow B$ 」「 $A \Rightarrow B$ $B \Rightarrow A$ 」「 $A \Rightarrow B$ $A \Rightarrow A$ 」「 $A \Rightarrow B$ $B \Rightarrow B$ 」「 $A \Rightarrow A$ $A \Rightarrow B$ 」「 $A \Rightarrow A$ $B \Rightarrow A$ 」「 $A \Rightarrow A$ $A \Rightarrow A$ 」「 $A \Rightarrow A$ $B \Rightarrow B$ 」）．あり得るプログラムの組み合わせは重複順列になる．それぞれの絵を選ぶ確率が同じだという帰無仮説に基づき，“正解 (A)”の人数に対して 2 項検定を行なった結果の p 値を“p 値”に示している．どのレッスンにおいても，プログラムを作成できた園児の数は有意に多い，ということがわかった．この結果から，園児たちは偶然に絵の繰り返しのプログラムを作った

表 5.2: L7-9 の二項検定の結果

		A/N	全場合	繰返の場合	確率	p 値
L7	P1	33/52	-	-	-	-
	P2	31/53	8	1	12.5%	2.742e-15
	P3	-	8	1	12.5%	-
L8	P1	-	243	4	1.64%	-
	P2	34/56	16384	36	0.22%	2.2e-16
L9	P1	28/49	8	1	12.5%	1.362e-13
	P2	24/46	8	1	12.5%	1.011e-10

のではないことがわかる．よって，園児は意図をもって絵の繰り返しのプログラムを作っていると考えられる．

それぞれの課題において，繰り返しを作ることができなかった園児のプログラムは，任意の最初の絵に最後の絵を戻すところで，間違っていることが多いことがわかった．L7-P2においては9名中， $A \Rightarrow B$ のメガネはできているのだが，逆方向のメガネが作れていないことに関する間違いが5名であった．他の4名の園児は，メガネ自体が不完全であったり，動くメガネを作っていたりした．L8-P2においては，11名中， $A \Rightarrow B$ ， $B \Rightarrow C$ ， $C \Rightarrow D$ まではできていて，DをAに戻せていない間違いが8名であった．他には，絵を1つのメガネで動かしている園児が3名であった．L9-P1においては，3名中， $A \Rightarrow B$ のメガネはできているのだが，逆方向のメガネが作れていないことに関する間違いが3名全員であった．L9-P2においては，8名中， $A \Rightarrow B$ のメガネはできているのだが，逆方向のメガネが作れていない間違いが7名であり，まったくルールを作っていない状態だった園児が1名であった．

5.4 自由課題の内容と分析

自由課題（F）では，園児によって作られるプログラムの数は違う．そこで，著者はそれぞれの園児が作ったプログラムの中で一番多くの絵を使って作られている，繰り返し続ける変化のプログラムに注目した．表5.3，表5.5，表5.8，表5.9はそれぞれL7，L8，L9の自由課題の結果を表している．それぞれのデータは，最長の変化の繰り返し続けるメガネの組み合わせ：Length of the Longest Loop (LLL)に基づいて分析した．

ここで $LLL \leq 1$ のメガネは，変化を繰り返すメガネの組み合わせを，一つも含まないプログラムを作った園児の数を表している．つまり，任意の絵AがBになる ($A \Rightarrow B$) ($LLL=0$) というメガネか，Aの位置が移動する ($A \Rightarrow A$) ($LLL=1$) というメガネしか作っていない園児を意味する．

“人数”の項目は，それぞれLLLのプログラムを作った園児の数を表している．“%”はそれぞれのLLLにおける園児の割合を示している．“平均”は園児がそのLLLのプログラムに使ったメガネの数の平均を表している．最後の“累積割合”は，L7，L8，L9における最長のLLLから $LLL=0$ に向かったの累積割合を表している．

また，表5.4，表5.6，表5.7はそれぞれのLLLのプログラムで，使われたメガネの数の平均の分布を表している．それぞれの繰り返しをLLLに従ってランダムに作る確率は，表5.2であった．これらの確率で，偶然にメガネの組合せで繰り返しが成立することがあるため，偶然は完全には取り去れない．しかし，それぞれの確率は小さく，特に $LLL \geq 3$ に関しては非常にその確率は小さい．よって，最低限に近いメガネの数で繰り返しを作成している園児は，意図をもって繰り返しを作っていると考えた．一方で，最低限以上のメガネでLLLを実装している園児に関しては，組み合わせを多く作った結果，偶然にそのLLLが達成できた可能性も

表 5.3: L7-F の LLL 集計

Length of Longest Loop	N = 53		
	≤ 1	2	3
人数	8	44	1
%	15.09%	83.02%	1.89%
平均	1.25	2.54	7
累積割合	100.0%	84.91%	1.89%

表 5.4: 平均メガネ数の分布 (L7-LLL2)

メガネ数 (平均)	N = 44
	人数 (%)
2.00-2.20	30 (68.18%)
2.20-2.40	2 (4.54%)
3.00-3.20	1 (2.27%)
3.20-3.40	5 (11.36%)
3.40-3.60	1 (2.27%)
4.00-4.20	4 (9.09%)
8.00-8.20	1 (2.27%)

あると考えた。このように、LLL に必要な最低限のメガネで作っている園児の数を、園児の理解を計る指標とした。

L7-F(自由課題)では、園児は絵を描き、練習で学んだ、2つのメガネを使った変化の繰り返し (LLL=2) の作成に取り組んだ。授業者は課題を「変化し続ける作品を作りましょう」と言って提示した。

表 5.3 に見られるように、83.02%の園児が LLL=2 のプログラムを作ることができている。LLL=2 のプログラムに使われているメガネの平均数は 2.54 であった。その分布が表 5.4 である。平均数 2.00-2.20 が一番割合が多く、68.18%の園児がこれにあたる。2.00-2.40 まで含めると 72.72%の園児が、平均して 3 個未満のメガネで LLL=2 のプログラムを作っている。このことから、LLL=2 を作った園児の多くが変化のメガネの意味を理解し、意図的に繰り返しのメガネの組み合わせを作れていたと考える。

LLL=3 を作っている園児も 1 人いた。LLL=3 のプログラムを作るのに必要な最低限のメガネは 3 である。しかし、このプログラムに使われたメガネは 7 である。このことから、メガネを多く出しているうちに、長さが 3 のプログラムができた可能性も考えられる。この園児の作っているメガネを図 5.8 に示す。同じ組み合わせのメガネがいくつか使われている。このことから、園児は無作為にメガネに違

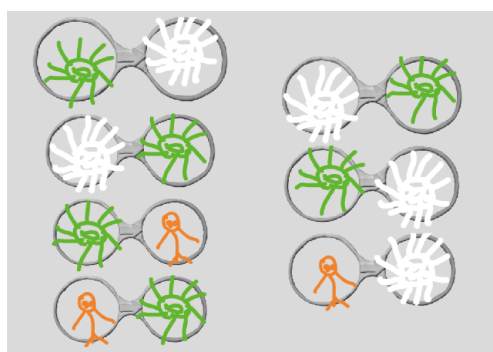


図 5.8: L7-LLL3 のプログラム

表 5.5: L8-F の LLL 集計

N = 56						
LLL	≤ 1	2	3	4	5	6
人数	16	13	23	2	1	1
%	28.57%	23.21%	41.07%	3.57%	1.79%	1.79%
平均	1.50	2.34	3.61	7	5	6
累積割合	100.0%	71.43%	48.22%	7.15%	3.58%	1.79%

う絵を入れる行為を続けることで、偶然に LLL=3 作することに成功している可能性がある。

また、 $LLL \leq 1$ の園児のプログラムに関して、プログラムを詳しく確認したところ、絵の変化自体に挑戦している園児は 8 名中 5 名であった。その中で 1 人は 3 つ以上の絵の繰り返しにチャレンジしていたが、最初の絵に戻すことができていなかった。3 名は 1 つも絵の変化をさせるプログラムを作っていなかった。前述のように、自由課題では課題を与えつつも、課題と逸脱したプログラムを作ることも許容している。

L8-F では園児は絵を 3 つ以上描き、 $LLL=3$ 以上のプログラムを作った。授業者は「3 つ以上の絵を使ってプログラムを作ってみましょう」と課題を提示していた。表 5.5 にある通り、 $LLL=2$ のプログラムについて、使われたメガネの平均は 2.34 である。その分布が表 5.6 である。平均数 2.00-2.20 が一番割合が多く、61.54% の園児がこれにあたる。2.40-2.60 まで含めると 84.62% の園児が、平均して 3 個未満のメガネ数でプログラムを作っている。 $LLL=3$ のプログラムについては、使われたメガネの平均数は 3.61 であった。その分布が表 5.7 である。平均数 3.00-3.20 が一番割合が多く、73.91% の園児がこれにあたる。これらにより、それぞれの LLL を作った園児の多くが、意図してプログラムを作っていると考える。

$LLL=5, 6$ については、LLL と一致する最小のメガネの数で、プログラムが作られている。このことから、3 を超える LLL についても、これらの園児はメガネによる変化の繰り返しの組み合わせを、理解していると考えられる。

表 5.6: 平均メガネ数の分布 (L8-LLL2)

N = 13	
メガネ数 (平均)	人数 (%)
2.00-2.20	8 (61.54%)
2.40-2.60	3 (23.08%)
3.00-3.20	1 (7.69%)
4.00-4.20	1 (7.69%)

表 5.7: 平均メガネ数の分布

N = 23	
メガネ数 (平均)	人数 (%)
3.00-3.20	17 (73.91%)
4.00-4.20	3 (13.04%)
5.00-5.20	2 (8.70%)
10.00-10.20	1 (4.35%)

一方で, LLL=4を作った2人(A児, B児)については, 最小限のメガネを超えたプログラムの作り方をしている. A児, B児が作ったメガネは図5.9の左と右である. 重複するメガネが複数あるのがわかる. L7の自由課題のLLL=3の園児と同じように, 組み合わせを増やすことで結果として, 変化が繰り返される組み合わせになった可能性が考えられる.

また, LLL \leq 1の園児のプログラムを確認したところ, 絵の変化自体に挑戦している園児は, 16名中11名であった. 11名の中8名は, 3つ以上の絵の繰り返しにチャレンジしていたが, 最後の絵を最初の絵に戻すメガネがなく, 繰り返しになっていなかった. 11名のうち3名は2つの絵の繰り返しを作ろうとしていたが, 最初の絵に戻すメガネが作られていなかった. 5名は絵の変化をさせるプログラムを作っていなかった.

L9の自由課題は, 棒人間のアニメーションを作ることを課題とした. ここで, Aクラスは, 初期画面に直立している棒人間の絵を一つ用意した. Bクラスは何も絵がない状態を初期画面とした(図5.10左, 右).

それぞれのクラスの結果が表5.8, 表5.9である. 表が示すように, どちらのクラスでも, 85%以上の園児がLLL=2以上の繰り返しを含むプログラムを作っている. LLL=2に関して, 使われているメガネの平均はクラスAで2.54, クラスBで2.41であり, L7, L8と大きな違いはない.

また, LLL=5について, それぞれのクラスで1人の園児が作成している. クラスBの園児はメガネ5個でLLL=5を完成させていた. 一方, クラスAの園児は



図 5.9: L8-LLL4 のプログラム (左 : A 児, 右 : B 児)

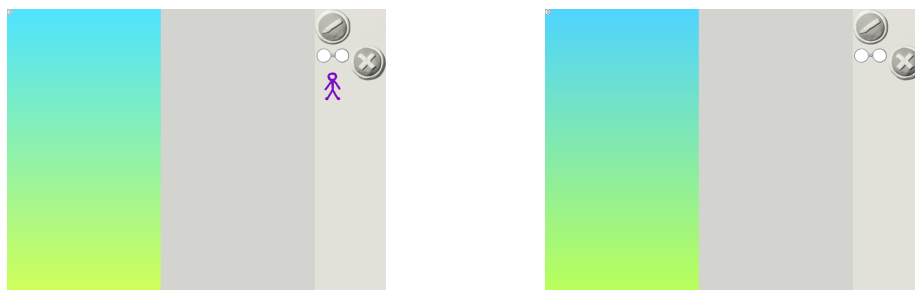


図 5.10: L9 の自由課題 (左 : クラス A, 右 : クラス B)

メガネを 9 個使っている。そのメガネを図 5.11 に示す。このプログラムには重複しているメガネはない。従って、いままでの L7, L8 で最小のメガネ以上のメガネを使っていた園児とは違う。この園児は意図的に 1 つの絵に対して、変化がランダムに起こるように、最小メガネ以上のメガネを使っている可能性が考えられる。これは L7, L8 では見られなかったパターンであった。

しかし、LLL=3 以上の長さを作った園児はクラス A において 16.39%, クラス B において 26.09%にとどまった。L8 (表 5.5) と比べると、長さ 3 以上のプログラムを作る園児の数は減っていることがわかる。

クラス A とクラス B の結果を表 5.8, 表 5.9 で比べると大差がないことがわかる。クラス A と B で自由課題の初期画面を変えたが、園児の作るプログラムの LLL に

表 5.8: クラス A の結果 (L9-F)

N = 26				
LLL	≤ 1	2	3	5
人数	4	18	3	1
%	15.38%	69.23%	11.54%	4.85%
平均	1.83	2.54	3.67	9
累積割合	100.0%	85.62%	16.39%	4.85%

表 5.9: クラス B の結果 (L9-F)

N = 23					
LLL	≤ 1	2	3	4	5
人数	3	14	4	1	1
%	13.04%	60.87%	17.39%	4.35 %	4.35%
平均	3.47	2.41	3.75	4	5
累積割合	100.0%	86.96%	26.09%	8.7%	4.35%

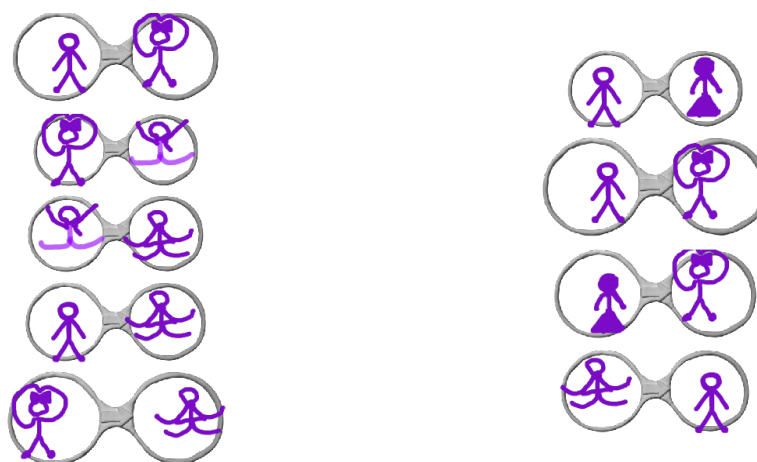


図 5.11: L9-LLL5 のプログラム (C 児)

は影響がなかったことがわかる。

“絵の変化の繰り返し”の含まれていない $LLL \leq 1$ の園児の数は、L8に比べると16名から7名と減っている。その内訳は、変化が繰り返し続けるプログラムにチャレンジしていない園児が2名と、絵が変化していくプログラムはできているのだが、最後の絵が最初に戻るというメガネがないために、繰り返しを作ることには失敗している園児が5名であった。

5.5 繰り返し続けるプログラムの理解についての分析のまとめ

練習課題 (L7-P2, L8-P2, L9-P1, L9-P2) において、80%を超える園児が与えられた絵を用い、変化が繰り返し続けるプログラムを作ることができていた。また、それぞれのプログラムが完成する組み合わせの確率から、プログラムを作成できた園児の数に対して2項検定を行ったところ、その数は有意に多いことがわかった。つまり、最小限のメガネで“絵の変化の繰り返し”のプログラムを作っている園児は、意図をもって作られていると考えることができた。一方で、最小限

のメガネ以上を使っている園児は、闇雲にメガネを出し、“絵の変化の繰り返し”になるまでそれ続けている可能性が、考えられた。

練習課題では正答を作る一手手前まで園児に見せていた。よって、園児は答えを一手手前まで写経のように写すことができたのであり、練習課題において80%以上の園児が課題をクリアできていたのは当然だとも考えられる。

20%の課題ができていなかった園児に関しては、任意のAという絵から別の絵に変化させた後に、最終的にAの絵に変化させるメガネが作られていないケースが一番多かった。メガネで具体的に絵を変化させることはわかるが、繰り返しを作るにあたり、最終的にAに戻すメガネを作る点が難しいことがわかった。

自由課題において、繰り返しを最低限のメガネで成立させる確率とLLLから、多くの園児が意図して繰り返しを作成していると考えることができた。一方で、園児それぞれのプログラムの数、プログラムの内容もバラバラであるため、個別の理解が見えた一方で、LLLやメガネの平均に関する俯瞰した結論を導き出すことはできなかった。

L7-Fにおいては、80%以上の園児が2つの絵を相互に変化させ続けるプログラムを作れていた。さらに、それらにおいて、3より小さい平均数でメガネを作っている園児の割合は、繰り返しを作った園児の72.72%であった。このことから、多くの園児は2つの絵を変化させ続けるプログラムを理解し、意図的に作っていると考えることができた。

L8の自由課題は3つ以上の絵を使った、変化し続けるプログラムだった。課題に沿ったプログラムを作った園児は48.22%であった。L7の自由課題に比べると、与えられた課題を遂行できた園児の数が減っていた。

L8において、LLL=3のプログラムを作った園児のメガネの数を調べたところ、4より小さい平均数でメガネを作っている園児の割合は、73.91%であった。このことから、LLL=3を作った園児の多くは、理解してプログラムを作っていたと考えることができた。

L9においては、最初に1つの絵を用意し、2つのクラスで差を検証した結果、大きな差がでなかった。よって、繰り返し続けるプログラムを作るにあたって、絵を1つ多く考えることが、プログラム制作の負担にはなっていないことがわかった。最小のメガネ数を超えたメガネを使ったプログラムであっても、L9では重複メガネは作らずに、ランダム性をプログラムに入れるために、メガネを多く使ったプログラムを作っている園児が確認できた(図5.11)。

重複するメガネを作成していた園児に関しては、それを理解してやっているのか、意図をせずに作ったものなのかは、プログラムを作っている様子を参照しなければ、判断は難しい。園児の理解を明確に判断するには、園児の制作している様子を観察し、時には「これはなに？」といった問いかけをし、園児が作っているものを把握する必要がある。

練習課題においても、自由課題においても、LLLが増えた場合、繰り返しが作ることのできていない園児は、任意の最後の絵を、任意の最初の絵に戻すことに

失敗している例が多く見られた。今回のレッスンでは明確に言葉にして「終わりの絵を最初の絵に戻す」ことを伝えてなかった。そのため、指導者は学習者に対し言葉にして「終わりの絵を最初の絵に戻す」ことを強調する必要が感じられた。その際、最後の絵で止まってしまっている間違いを見せ、最後の絵から最初の絵に戻すメガネを作ることによって、変化が循環し、繰り返しになることを画面で見せると、聴覚的にも視覚的にも園児の理解が進む可能性がある。これによって、LLLが増えた場合でも、学習者を効果的に指導できる可能性があると考えられる。

第6章

ランダム動きと回転動き についての理解の分析

6.1 はじめに

本章では、園児の作る“ランダム動き”のプログラム、そして、“回転動き”のプログラムについて分析した。“ランダム動き”に関しては第5回（L5）、第6回（L6）のレッスン、そして、“回転動き”のレッスンは第9回（L9）から第12回（L12）のレッスンで行われた。

本章のリサーチクエスションは下記である。

- 園児は“ランダム動き”のプログラムを作ることができるか
- 園児は“回転動き”のプログラムにおいて、回転の大きさを意図して使い分けられるか

上記を、園児が作ったプログラムから明らかにする。

6.2 ランダム動きの理解の分析

L5-6では、1つの絵に対してメガネを複数使うことによって、絵をランダムに動かすことを学んだ。練習課題（P）では、ゆらゆら、または、動いたり止まったりする動きを特徴とする絵を用いて、園児はプログラムを作成した（図6.1, 図6.2）。自由課題（F）では、ランダムに動かす絵自体も、園児が描いてプログラムを作成した。

L5-6でも、L1と同様に課題と逸脱したプログラムを作成していた園児が見られた。本章の分析では2つ以上プログラムを作っていた場合は、その中で課題通りに作られたものがあった場合は、課題を遂行したプログラムとして集計した。ま

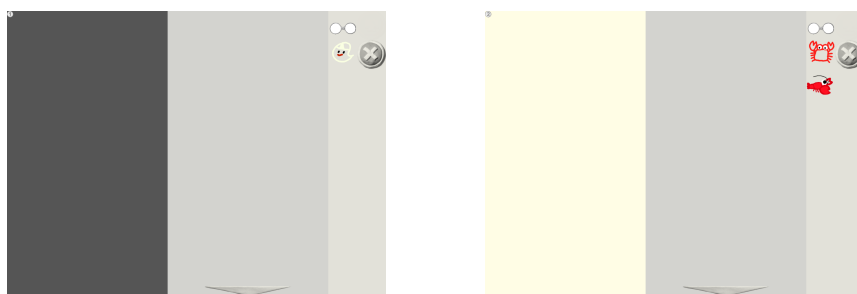


図 6.1: 左 : L5-P1 の画面, 右 : L5-P2 の画面

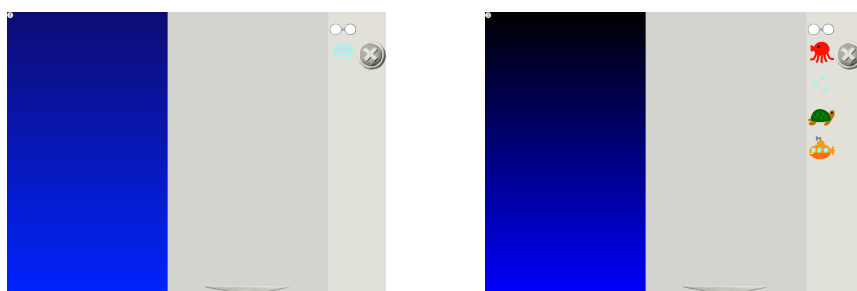


図 6.2: 左 : L6-P1 の画面, 右 : L6-P2 の画面

た, メガネの数に関しては, 本レッスン以降は2つ以上使うプログラムが求められるため, メガネをいくつ使っているかに関わらず, プログラムを分析した。

L5 と L6 の間には夏休みがあり, 3ヶ月以上期間が空いていた。そのため, 園児がViscuitを使ったプログラムの作り方を思い出すためにも, L5 と L6 の内容は見本の絵を変えただけで, 課題の内容としては同じものとした。

それぞれのレッスンにおいて, 集計したプログラムのデータから, 同じ絵から始まるメガネが複数あるプログラムを抽出し, 分析を行った。練習課題は動きによる課題, 自由課題については使うべきテクニックを基準に分析を行った。

6.2.1 練習課題の内容と分析

L5-P1, P2, また, L6-P1, P2 では絵が一方方向に動き続けると, 違和感を感じるような絵を準備した。L5-P1 (図6.1 左) では「おばけ」の絵を用意した。「おばけ」をランダムにゆらゆら動かしてみたい, と思わせるようにレッスンを実施した。L5-P2 (図6.1 右) では「カニ」と「エビ」の絵を用意した。「カニ」はL1にも登場しているが, ランダムの動きを覚えた上で, より「カニ」らしく動かすことを課題とした。L6-P1 (図6.2 左) では,「くらげ」の絵を用意し, それを受けてL6-P2 (図6.2 右) では,「タコ」「泡」「カメ」「潜水艦」の絵を用意し, それぞれの絵を, 海の中にランダムにゆらゆら動かすことを課題とした。

これらの課題で作られたプログラムを分析した結果が表6.1である。「N」はレッスンの参加園児の数を表している。「0 (%)」は「ランダムの動き」のプログラム

表 6.1: L5-6 の練習課題の結果

		N	0 (%)	>1(%)
L5	P1	51	4(7.84%)	47(92.16%)
	P2	51	12(23.53%)	39(76.47%)
L6	P1	52	1(1.92%)	51(98.08%)
	P2	52	20(38.46%)	32(61.54%)

を1つも作っていなかった園児の数, “>1 (%)” は“ランダム動き”のプログラムが1つ以上作れていた園児の数を表している。

L5, L6ともに, P1については大半の園児がランダムが作れていることがわかる。一方で, どちらのレッスンでも, P2ではランダムが作れた園児の数が減少している。

それぞれの練習課題で, 園児の作ったプログラムを授業者の教え方も確認し, 詳細に見る。L5-P1 (図 6.1 左) では, 授業者は最初にまっすぐ上に動くおぼけを見せた後に「でも, おぼけってこんな風にまっすぐ動く?」と, 園児に聞いていた。そうすると園児は「違う」と答えた。その後, 授業者は「おぼけはゆらゆら動くよね。そういう動きはメガネを2つ使うとできます。(上に動くメガネがある状態で) もう一個下に動くメガネを作ると, どうなるかやってみてください」と, 自分ではメガネは作らず, 園児たちに自分のタブレットでやってみるように促した。

表 6.2 は, 園児たちが「おぼけ」に対して作ったプログラムの分類である。この課題で園児に作ってもらいたかったメガネは「上に動く」メガネと「下に動く」メガネである。4人の園児がランダムのメガネを作っていなかった。34人の園児がメガネを2つ用い, その中で22人の園児が「上」と「下」のメガネを作っていた。「上」と「下」以外のメガネを作っていた園児の中では, 「左」に進むメガネを2つ作っていた園児が一番多かった。これはおぼけが「左」を向いているからだと考えられる。15人の園児がメガネを3つ以上出して, プログラムを制作していた。これらの園児の中で, 「上」「下」のメガネを含むプログラムを作成していたのは10人であった。これらの園児は「上」「下」のメガネに加え, さらに「上」「下」を追加する園児, また「右」「左」を追加する園児も見られた。5つメガネを使い, しかも, 上下を含まないメガネを作っている園児は「上」に動くメガネと「右」に動くメガネを1つずつと「左」に動くメガネを3つ作っていた。47人 (92.16%) の園児がメガネを2つ以上使っており, その中の32人 (60.38%) の園児が与えられた課題を遂行したプログラムを作っていたと言える。

L5-P2 (図 6.1 右) では, 「カニ」と「エビ」を, それぞれ“ランダム動き”を使って, 生き物のように動かすことを課題とした。ここでは, 園児が楽しみながら, ランダムのプログラムに親しみ, どうしたら生き物のようになるかを考えてもらうことを狙いとした。生き物のように動かすには「止まる」メガネが必要で

表 6.2: L5-P1 の練習課題の結果

メガネの数	人数 (%)	上下含む	上下含まない
1	4(7.84%)	-	-
2	32(62.75%)	22	10
3	6(11.76%)	3	3
4	5(9.80%)	4	1
5	4(7.84%)	3	1
合計	51	32(60.38%)	15 (50.94%)

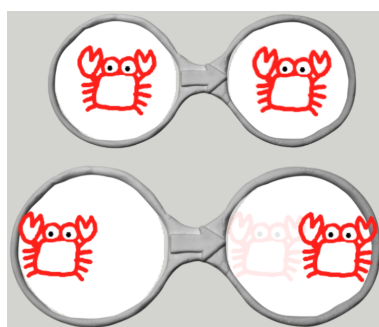


図 6.3: 「カニ」っぽい動きの例

ある．動きの中に時々「止まる」メガネが入ると，生き物の動きに見えてくる（図 6.3）．「止まる」というのは，メガネの右と左に絵を入れる時に，前の絵からずらさないように入れると「止まる」命令になる．一方で，絵を完全に前の位置に合わせることは難しいため（指を離す時に絵がずれてしまう場合がある），分析の際は絵のずれが座標位置において 5 ポイント以下のメガネを「止まる」メガネだとみなした．

授業者は自身が右に動いたり，左に動いたり，止まったりする様子を見せて「ゆらゆらのメガネを使って，カニやエビっぽい動きを作ってみてください」と言って，メガネの作り方は見せずに園児にプログラムを作らせた．

表 6.3, 表 6.4 は園児たちが「カニ」「エビ」に対して作ったプログラムの分類である．「カニ」については，48 人の園児が取り組んだ．10 人はメガネ 1 つで，カニをまっすぐに動かしていた．29 人の園児がメガネを 2 つ使って，カニをランダムに動かしていた．このうち，横方向のメガネだけで「カニ」を動かしていたのが 22 人，上下も加えて動かしていたのが 7 人だった．また，この 22 人中 13 人が右方向に動くメガネを 2 つ作っていた．そして，9 人は左右に行ったり来たりするメガネを作っており，9 人のうち 3 人は，動きが小さい「止まる」メガネを作っていた．メガネ 2 つでカニを動かしていた園児の中で「止まる」を作っていたもう一人の園児は，上方向に動く時が「止まる」動きであり，座標上は上に動くメガネを作っていたが，「右にいたり，止まったり」するメガネになっていた．また，メ

表 6.3: カニの結果

メガネの数	人数 (%)	横方向のみ	上下を含む	止まる含む
1	10(20.83%)	-	-	1
2	29(60.42%)	22	7	4
3	1(2.08%)	1	0	0
4	5(10.42%)	3	2	0
5	2(4.17%)	2	0	2
7	1(2.08%)	-	1	0
合計	48	28(58.33%)	10(20.83%)	7 (14.58%)

表 6.4: エビの結果

メガネの数	人数 (%)	横方向のみ	上下を含む	止まるを含む
1	19(45.24%)	-	-	0
2	20(47.62%)	14	6	1
3	2(4.76%)	0	2	0
4	1(2.38%)	1	0	0
合計	42	15 (35.71%)	8 (19.05%)	1 (2.38%)

ガネ5つを作っていた園児も「止まる」メガネを取り入れていた。

「エビ」については、取り組んだ園児の数がカニより少なく42人だった。19人の園児が、メガネを1つでまっすぐ動くように「エビ」を動かしていた。これは「カニ」と比べて多かった。動きの方向については、20人がメガネを2つ使ってランダムに「エビ」を動かしていた。この20人のうち、12人が右に動くメガネを2つ作っていた。そのうち1名が「止まる」メガネを作っていた。左右に動かしている園児は2名だった。

授業者は課題に取り組ませる際に、メガネを一切見せていなかった。課題を遂行している園児は、授業者の真似をしてこれらのメガネを作ったのではなく、自分で考えてこれらのメガネを作っていると考えられる。園児がメガネを複数使い、様々な方向に動かしていることから、「カニ」「エビ」を生き物っぽく動かすにはどうしたらいいか、試行錯誤していると言える。一方で「カニ」では20.83%、「エビ」では45.24%の園児が“ランダムの動き”を使っていない。このことから、用意した絵と求めている動きのプログラムの、うまくマッチしていなかった可能性が考えられる。L5-P2の課題は自由度が高い課題であったこともあり、この結果から園児の理解を明確には推測することはできなかった。

L6-P1 (6.2左) では、クラゲをゆらゆらと、ランダムに動かすことを課題とした。前回のレッスンから、3ヶ月以上間隔が空いていたということもあり、この回

表 6.5: L6-P1 の練習課題の結果

メガネの数	人数 (%)	上下含む	上下含まない	止まるを含む
1	1(1.92%)	-	-	0
2	17(32.69%)	12	5	2
3	9(17.31%)	2	7	0
4	16(30.77%)	12	4	3
5	6(11.54%)	6	0	3
6	3(5.77%)	3	0	1
合計	53	35 (66.04%)	16 (30.19%)	9 (16.98%)

はL5 でやった内容の復習とした。授業者は最初にクラゲを下に動かして見せた。その時点で園児から「ゆらゆら」という声をあげていた。その後、同じメガネを使って、授業者はクラゲを上の方に動かし、園児に「クラゲってこういう風にずっと上にいく？」と園児に尋ねた。園児は「上下、上下」という返答していた。授業者はもう一つメガネを出し「では、メガネを2つよりも多く使って、クラゲを泳がしてみてください」と言い、園児を課題に取り組ませた。

表 6.5 が L6-P1 の園児のプログラムの分析結果である。1 人の園児がメガネを1 つだけでプログラムを作っていた。2 つ以上のメガネを使って、ランダムにクラゲを動かしていた園児のプログラムの中で、上下の動きが含まれていたのは35 人(66.04%) だった。16 人(30.19%) の園児が、メガネは2 つ以上使えていたが、上下のメガネが含まれていなかった。園児が課題として上下のメガネを作ることを終え、他の実験をはじめ、メガネを改造してしまった可能性もある。また、クラゲの動きの中に「止まる」メガネを活用している園児は9 人(16.98%) いた。

L5-P1 よりも、課題となる絵をランダムに動かしている園児は多かった。2 回目の“ランダムの動き”のレッスンだったということと、クラゲの絵が“ランダムの動き”にうまくマッチしていたことが要因と考えられる。しかし、52 人(98.08%) がランダムのメガネを作っていた一方で、その中で授業者が出した課題を遂行していた園児は35 人(66.04%) だった。

L6-P2 では「タコ」「泡」「カメ」「潜水艦」と、様々な「うみ」の世界観にあるものを用意し、それらをランダムに動かす課題とした。授業者は特にメガネを見せることなく「メガネを2 つ以上使って絵を動かしてみてください」とだけ言い、園児を課題に取り組ませた。

表 6.6 は L6-P2 の園児の作ったプログラムの分析結果である。L6-P2 は複数の絵が用意されていたため、4 つのうちの、何種類を園児がランダムに動かしていたかを集計した。プログラムの中で、1 つの絵もランダムに動かさなかった園児が20 人(38.46%) いた。4 つの絵のうち1 つだけランダムに動かしていたのが15 人(28.85%)、2 つが12 人(23.08%)、3 つが4 人(7.69%)、そして、4 つ全てラン

表 6.6: L6-P2 の練習課題の結果

絵の種類	人数 (%)
0	20(38.46%)
1	15(28.85%)
2	12(23.08%)
3	4(7.69%)
4	1(1.92%)
計	52

ダムに動かしていた園児は1人(1.92%)であった。P1の結果と比べると、絵が増えたため課題の意図がぼやけ、園児はただ絵を動かすことに、没頭してしまった可能性が考えられる。

「泡」で19点、「タコ」で16点、「カメ」で10点、「潜水艦」で7点のゆらゆらさせているプログラムが確認できた。園児は絵自体に、ランダムに動かす必然性を感じなかった可能性も考えられた。これらの4つの絵の中では「泡」が一番ランダムに動く必然性が高い、と園児は感じたのだと思われる。

6.2.2 自由課題の内容と分析

L5の自由課題は七夕の日程が近かったため「七夕」とした。また、L6の自由課題はハロウィンが近かったため「ハロウィン」とした。それぞれテーマを設定した上で、その前の練習課題で学んだランダムの動きを使うことを提示し、自由課題に取り組ませた。

自由課題では、それぞれの園児について、作った全ての作品ファイルの中で、1つ以上“ランダムの動き”の作品ファイルを作った園児の数を数え、割合を出すことで園児の理解を分析した。また、それぞれの園児が作ったランダムの作品ファイルが、それぞれの園児が作った全体の作品ファイルの何パーセントを占めるかを出し、全体としての平均を出した。

表6.7は、L5-F、L6-Fの分析の結果である。L5-Fにおいて、保存された作品ファイルは全部で338個であった。1人あたりとしては平均で5.98個の作品ファイルを作っていた。この中でメガネを2つ以上使った作品ファイルを、1つでも作った園児は20人(35.71%)であった。また、それぞれの作品ファイル総数の中で“ランダムの動き”の作品ファイルが占める割合の平均は24.30%であった。

L5のP1では92.16%、P2では76.47%の園児が“ランダムの動き”の作品ファイルを作っていたのに対し、“ランダムの動き”の作品ファイルが少ないことがわかる。これはテーマが「七夕」であったことから、必ずしも“ランダムの動き”と作るプログラムが、結びつかなかったためだと考えられる。

表 6.7: L5, L6-F の結果

	N	平均	0 (%)	> 1(%)	ランダム割合
L5	56	5.98	36(64.29%)	20(35.71%)	24.30%
L6	52	5.19	17(32.69%)	35(67.31%)	45.66%



図 6.4: Viscuit における回転の機能のボタン

L6-F において、保存された作品ファイルは全部で 270 個であった。1 人あたりとしては平均で 5.19 個の作品ファイルを作っていた。この中でメガネを 2 つ以上使った作品ファイルを、1 つでも作った園児は 35 人（67.31%）であった。また、それぞれの作品ファイル総数の中で“ランダムの動き”の作品ファイルが占める割合の平均は 45.66%であった。

L5 の自由課題の結果から比べると、園児の“ランダムの動き”を使う割合は増えていた。一方で、第 4 章における“直線の動き”，第 5 章における“絵の変化の繰り返し”の自由課題と比べると、練習課題で学んだことを使っている割合が低かった。

6.3 回転の動きの理解の分析

6.3.1 練習課題の内容と分析

L10 からは、絵の向きを傾けることを可能にするボタンが追加された（図 6.4）。このボタンは初期状態は図 6.4 の左のようだが、タッチすると右のように凹んだボタンに表示が変わる。この状態になると、絵を、絵の中心を基準として、傾けることができるようになる。

L10（図 6.5）では、まず回転の動きに慣れるために、回転させること自体を目的とした。著者と、授業者である幼稚園の教諭は、園児にとって絵が回転すること自体が面白いと考えた。そこで L10 では、絵を傾けると回転するということに親しむために、必ずしも回転する必然性を持っていない絵も使い、レッスンを構成した。L10-P1 では渦をまわす「ぐるぐる」の絵と、L10-P2 では絵を回転させること自体の楽しさを感じてもらうために、必ずしも回転する必然性をもたない「りんご」「花」「鉛筆」「星」の絵を用意した。

授業者の園児に対する教え方を確認した。L10-P1 では、まず授業者は、用意されている絵が何の絵なのかを園児に尋ねた。すると園児は「ぐるぐる」と答えた。

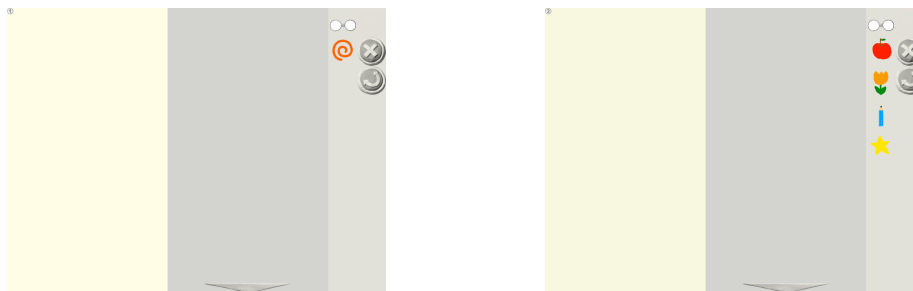


図 6.5: 左 : L10-P1 の画面, 右 L10-P2 の画面

その後、授業者は「ぐるぐる渦巻きがあったらどういう風に動かしたい？」と尋ね、園児たちは手で指を回転させる仕草を見せながら「ぐるぐるー」と元気よく答えた。授業者はその後、メガネの両方に「ぐるぐる渦巻き」を入れ、絵がまっすぐ動く様子を見せ、園児にこれでいいか尋ねた。園児は「だめー」と答えた。そして、授業者は画面に新しいボタン（図 6.4）が増えていることを指摘し、実際にボタンを押し、すでにメガネに入っている絵を傾けてみせ、舞台上で絵が回転している姿は見せずに、園児に席に戻ってメガネを作るように指示していた。園児が作成している最中に、授業者は再度園児を前に集めた。授業者は絵の傾き方によって、回転の仕方が変わることを説明した。その後、その説明を踏まえた上で、もう一度園児を席に戻らせてプログラムの作成に取り組ませた。

P2では、「りんご」の絵を故意にまっすぐ動かし、園児に回転のボタン（図 6.4）を押すことを指摘されてから、回転のボタンを押し、絵を回転させた。その後「えんぴつ」も回転させ、もう一度傾きが絵の回転の速さに関わっていることを教え、園児に P2 に取り組ませた。

L11（図 6.6）の学習内容は、中心を合わせて絵を回転させることである。メガネの右と左で、絵の中心を合わせた上で絵を傾けると、絵は移動をせずにその場で回転する。L11-P1 では、中心がはっきりした渦をまく「星雲」の絵を用意した。L11-P2 では 3 種類の「風車」の絵と「風車の棒」の絵を用意した。

L11-P1 では、園児から絵が「ブラックホールだ」という指摘を受け、絵を「ブラックホール」の見立てでレッスンを進めた。最初に故意にまっすぐに動かし、園児と一緒に回転を作る方法を確認した。また、回転させるときも故意に中心をずらし、少し大きく回転をさせ「これじゃ、なんかおかしいね」と園児に問いかけた。その時、園児から「真ん中に置く」という意見が出た。その後、中心に合わせ、絵が回転している様子を見せ、園児に席に戻ってメガネを作るように指示した。

L11-P2 では、まず画面が変わったあとに園児に「これなーんだ」と園児に聞くと、園児は「風車（かざぐるま）ー」と答えた。ステージに「棒」の絵を置き、その「棒」の先端に風車の絵を配置し、最初に故意に「風車」がまっすぐ動き、棒から離れていく様子を見せた。その後、中心を合わせて回転させ、傾きによって回転の速さが変わることを、そして、回転の速さで風の強弱が想像できることを話し、園児に課題に取り組ませていた。



図 6.6: 左 : L11-P1 の画面, 右 : L11-P2 の画面



図 6.7: 左 : L12-P1 の画面, 右 : L12-P2 の画面

L12 (図 6.7) の学習内容は、絵を大きく回転させることである。P1 では、いままではメガネに入れた絵を傾けていたが、ステージ上に絵を傾けて配置することで、一つの“直線の動き”のメガネでも、様々な方向に動かせることを学んだ。見本の絵は方向のはっきりしている「ロケット」と「流れ星」を用いた。P2 では「(軌跡が曲がっている) 流れ星」を用意し、軌跡に合わせて絵を傾けることで、大きな回転の作り方を学んだ。L12-P2 についてはステージに絵を傾けておくことが課題だったため、メガネに関する分析はしなかった。

L12-P2 では、最初に故意に「流れ星」をまっすぐ動かし、軌跡が曲がっているにも関わらず、まっすぐ進んでいるのが気持ち悪いことを園児と確認した。そして、その場で回転させても気持ちが悪いことを確認した。その後、回転させてない状態で少し左のメガネの中で絵をずらし、その後に少し傾けると大きな回転になることを伝え、実際にステージ上で流れ星が大きく回転している様子を見せ、園児に課題に取り組ませた。

これらの課題で作られたプログラムを、分析した結果が表 6.8 である。“N” は対象となる園児の数である。“回転” は回転を使っていた園児の数と割合を表している。“正解” は回転を使っていた園児の中でも、提示された課題を遂行していた園児の数と割合を表している。L10 においては、回転を使うことを主題と考えていたので、“正解” に値する値は記載はしなかった。

表 6.8 から、90%以上の園児が回転のテクニックを使えていることがわかる。L11 においては、座標面で 5 ポイント以下のズレで、中心を合わせて絵を回転させているメガネを「中心が合っているメガネ」として集計した。これは L5, 6 における

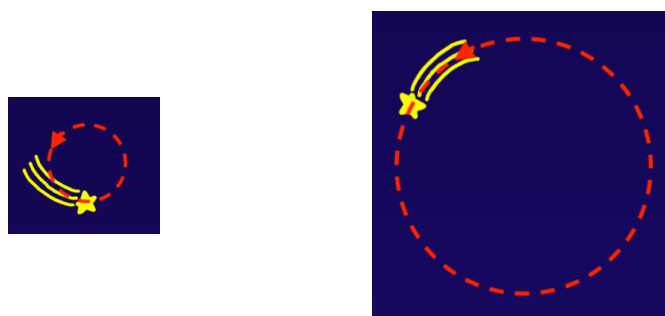


図 6.8: 回転の大きさの比較：左が回転半径 30 ポイント，右が 100 ポイント

表 6.8: L7-9 の練習課題の結果

		N	回転 (%)	正解 (%)
L10	P1	53	51(96,23%)	-
	P2	50	46(92.00%)	-
L11	P1	50	48(96.00%)	19(38.00%)
	P2	50	45(90.00%)	35(70.00%)
L12	P2	41	39(95.12%)	27(65.85%)

「止まる」メガネの判別の仕方と同じである．L10 と L11 における P1 は用意された絵は一個であったが，いずれのレッスンにおいても P2 においては，用意された絵が 4 個であったため，P2 に関しては，少なくとも 1 つの絵に課題を遂行している園児の数を数えた．P1 においては正解のメガネを作っている園児は 38%であった．一方で P2 においては，70%の園児が中心に近づけた回転を 1 つ以上作っていることがわかった．

L12-P2 においては正解を，回転の半径が，右と左のメガネの絵の移動よりも大きく，また，回転中心までの距離が 30 ポイント以上のものを正解とした．これは，授業者がみせる課題が，その通りに実行すると回転半径がおおよそ 30 ポイントになることを参考にした（図 6.8）．L12-P2 においては 65%の園児が課題を遂行できているのがわかった．

それぞれの練習課題において，園児がどのような回転の使い方をしていたかを確認する．L10-P1 において，図 6.9 は園児が作った回転の大きさ（回転円の半径）と回転の傾きを示している．回転の大きさのヒストグラムについては，園児の 1 人が半径 1450.38 のプログラムを作っていたため，そのデータは除いてグラフを作っている．ヒストグラムをみると回転の大きさに関しては，0 から 25.00 のところと 50.00 から 75.00 に山が来ているのがわかる．また，回転の傾きに関しては，偏りがなく絵を傾けていることがわかった．

L10-P2 では「りんご」「鉛筆」「はな」「ほし」の 3 つの絵が与えられた．図 6.10, 図 6.11, 図 6.12, 図 6.13 はそれぞれの回転の半径の大きさと，それぞれの絵につけ

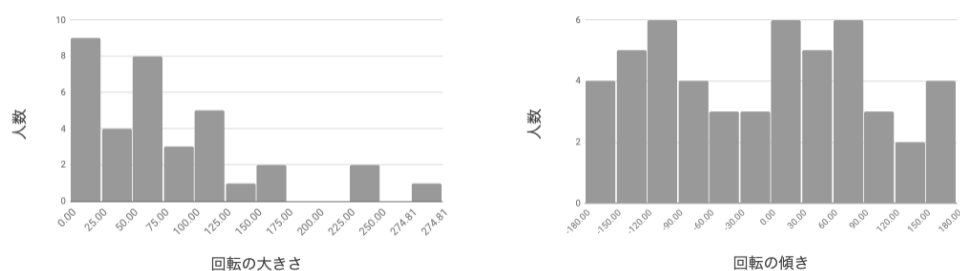


図 6.9: L10-P1 の大きさと傾き

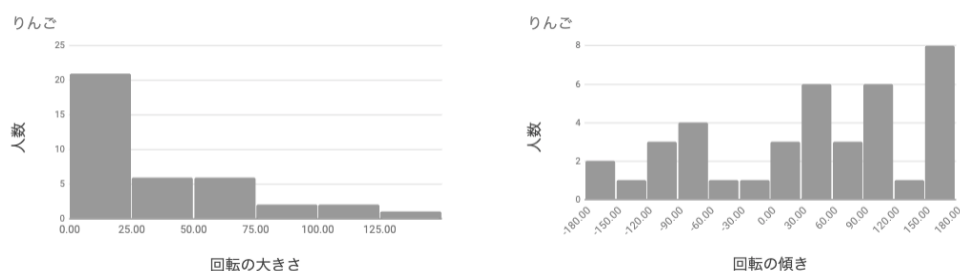


図 6.10: L10-P2 りんごの回転の大きさと傾き

られた傾きのヒストグラムである。「りんご」に関しては、園児の一人が回転半径 7876.44 のプログラムを作っていたため、そのデータは除いてグラフを作っている。回転の大きさについては、どのレッスンにおいても、0 から 25.00 に山がきており、回転は大きくはないことがわかった。回転の傾きについては、それぞれのヒストグラムを見ると、90 度に近い傾きが多いようにみて取れた。

L11-P1 は絵の中心を合わせて回転させる課題であった。図 6.14 は園児が作った回転の大きさ（回転円の半径）と、園児がどのくらい絵を傾けていたかのヒストグラムを示している。このヒストグラムでは、メモリを 10.00 刻みにした。回転の大きさに関しては、ヒストグラムをみると 0.00 から 10.00 のところに一番山が来ているのがわかる。L10 に比べると、中心に近づけることが意識されていることがわかる。回転の傾きに関しては、L10 と同じよう 90 度付近の偏りが多いようにみて取れた。

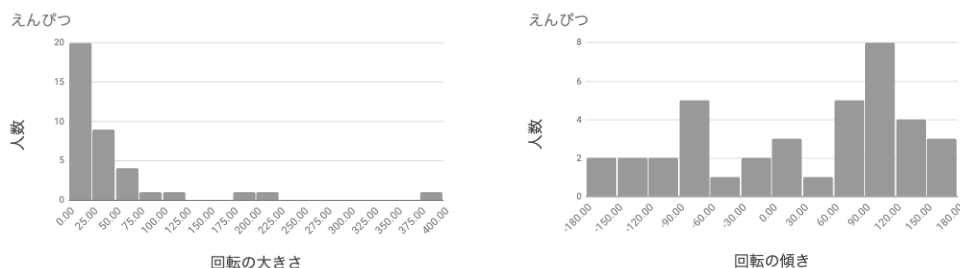


図 6.11: L10-P2 鉛筆の回転の大きさと傾き

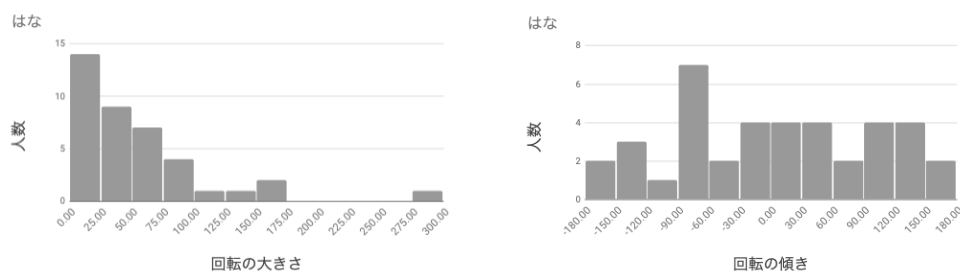


図 6.12: L10-P2 はなの回転の大きさと傾き

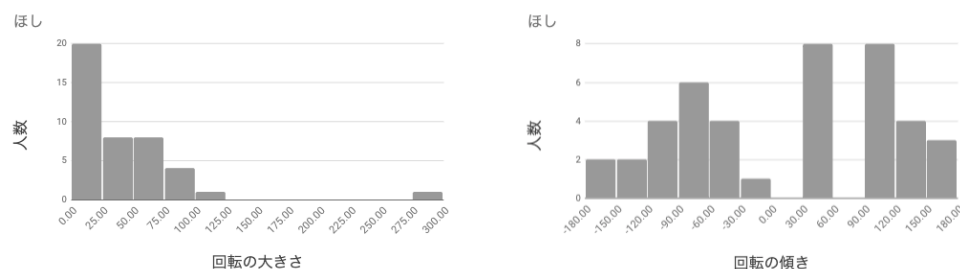


図 6.13: L10-P2 星の回転の大きさと傾き

L11-P2 では「棒」「4 色の風車」「2 色の風車」「1 色の風車」の 4 つの絵が与えられた。図 6.10, 図 6.11, 図 6.12, 図 6.13 それぞれの回転の半径の大きさと、それぞれの絵につけられた傾きのヒストグラムである。回転の大きさに関しては、ほとんどが 0.00 から 10.00 に収まっており、絵の中心の近くで回転しているのがわかった。回転の傾きに関しては特徴を導き出せなかった。

L12-P2 では「(軌跡がカーブしている) 星」の絵が与えられた。図 6.18 は園児が作ったプログラムの半径の大きさと、絵につけられた傾きのヒストグラムである。回転半径が 1269.33, 804.88 という園児がそれぞれ 1 名いたため、それらのデータは削除してグラフを作成した。回転の大きさについては、小さい山が 25.00-50.00 にある一方で、L10, L11 と比べると、100.00 から 125.00 のところを中心に山が来ているのがわかる。多くの園児が絵を大きく回転できていることがわかった。また、回転の傾きに関しては、L10, L11 と比べると、-30 度から 0 度の園児が多いこ

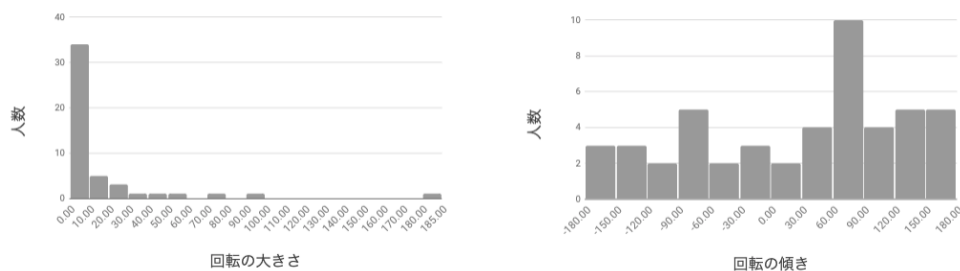


図 6.14: L11-P1 の回転の大きさと傾き

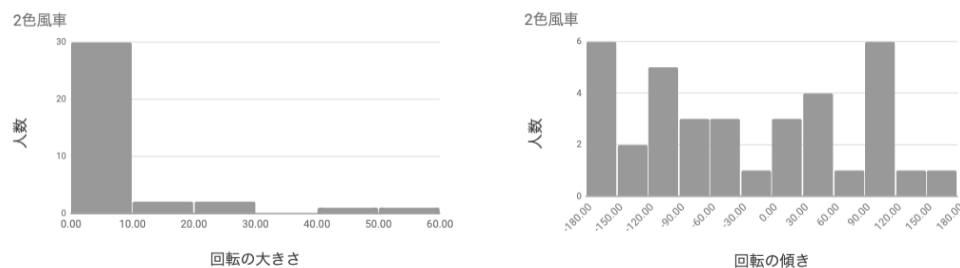


図 6.15: L11-P2 : 2 色の風車の大きさと傾き

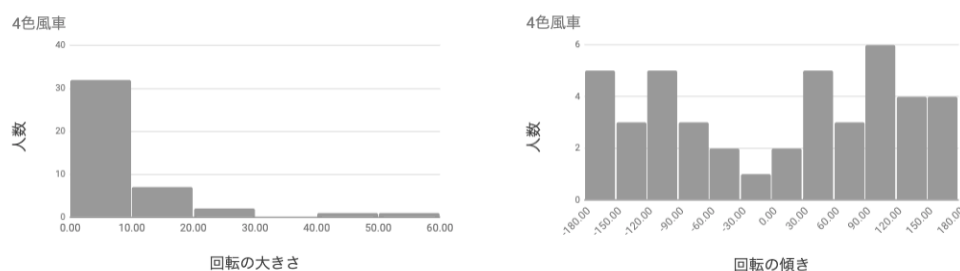


図 6.16: L11-P2 : 4 色の風車の大きさと傾き

とがわかる。これは流れ星が左を向いているためにマイナスの傾きであり、また、大きく回転させるためには、回転の角度を小さくしなければならないため、ここでも、一定数の園児が大きな回転を作れていることがわかった。

6.3.2 自由課題の内容と分析

L10の自由課題は、とにかく回転させることを楽しんでもらうことを意図し「ぐるぐるスーパー」とした。園児たちにも親しみのある、スーパーマーケットにあるものをいろいろ回転させよう、という趣旨である。L11ではL11-P2でやった風車を受けて、「オリジナルの風車を作ろう」とした。L12は夜空をテーマにして、P2の流れ星のから「(夜空に) 流れる〇〇」とした。それぞれの自由課題では、園児はP課題で学んだことを使うように、授業者から指示された。自由課題では、1つ

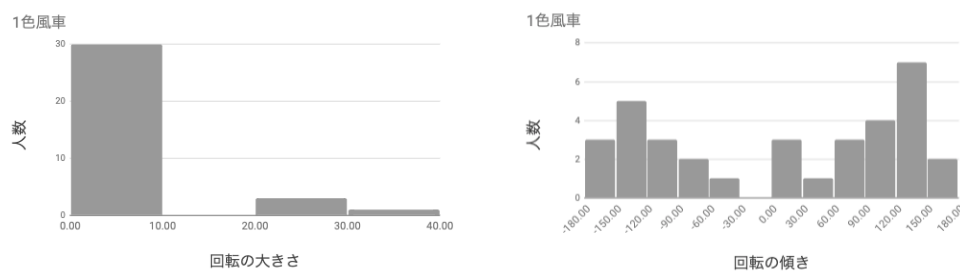


図 6.17: L11-P2 : 1 色の風車の大きさと傾き

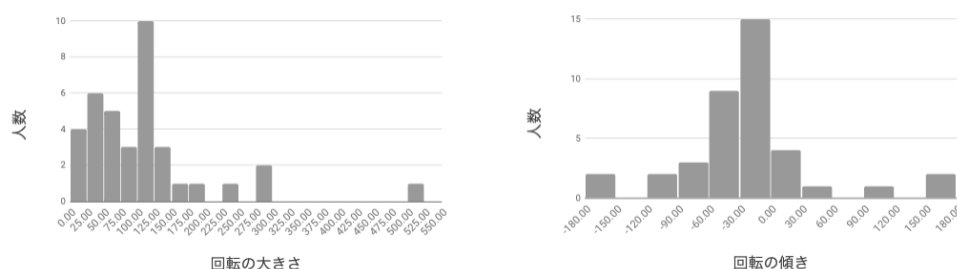


図 6.18: L12-P2 の大きさと傾き

表 6.9: L10, L11, L12-F の結果

	N	平均	回転あり (%)	回転の割合
L10	53	4.85	49(90.57%)	77.49%
L11	50	6.5	49(98.00%)	91.27%
L12	41	5.0	36(87.80%)	62.22%

以上回転の動きのプログラムを作った園児の数を数え、割合を出すことで園児の理解を分析した。

表 6.9 はそれぞれの自由課題における“回転の動き”を使った園児の割合と、一人一人の全作品ファイルの中に占める“回転の動き”を使った作品ファイルの割合である。L10-F では、保存された作品ファイルは全部で 257 個であった。1 人あたりとしては、平均 4.85 個の作品ファイルを作っていた。この中で、1 つでも“回転の動き”の作品ファイルを作った園児は 53 人中 48 人 (90.57%) であった。また、それぞれの作品ファイルの中で“回転の動き”の作品ファイルが占める割合の平均は、77.49% であった。

L11-F (自由課題) の分析すると、保存された作品ファイルは全部で 325 個であった。1 人あたりとしては、平均 6.5 個の作品ファイルを作っていた。この中で、1 つでも“回転の動き”の作品ファイルを作った園児は 50 人中 49 人 (98.00%) であった。また、それぞれの作品ファイルの中で“回転の動き”の作品ファイルが占める割合の平均は、91.27% であった。

L12-F (自由課題) の分析すると、保存された作品ファイルは全部で 205 個であった。1 人あたりとしては平均 5.0 個の作品ファイルを作っていた。この中で、1 つでも“回転の動き”の作品ファイルを作った園児は 41 人中 36 人 (87.80%) であった。また、それぞれの作品ファイルの中で、“回転の動き”の作品ファイルが占める割合の平均は、62.22% であった。作品ファイルに占める“回転の動き”の割合が減ったのは、P1 で絵を傾けてステージに置き、それをまっすぐ動かしたからだと思われる。また、流れ星自体も本来はまっすぐ飛ぶものであるため、園児の発想がそのようになったのではないかと考えられる。

図 6.19 は L10 において、それぞれの園児が作った個人個人の“回転の動き”の作

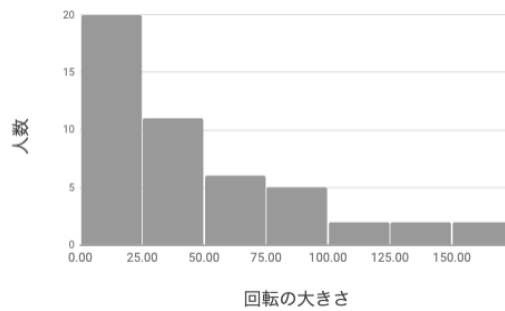


図 6.19: L10-F 回転半径の平均のヒストグラム

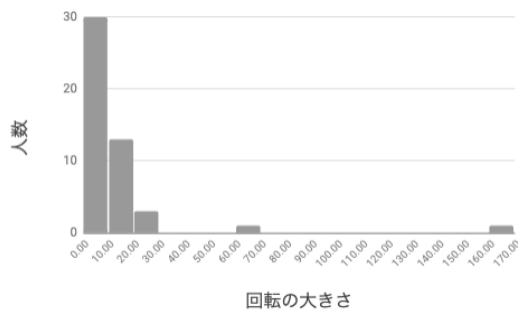


図 6.20: L11-F 回転半径の平均のヒストグラム

品ファイルの、円の大きさの平均をヒストグラムにしたものである。これを見ると、0.00 から 25.00 に一番山ができており、小さい回転のプログラムを園児は作っている傾向が見えた。

図 6.20 は L11 において、それぞれの園児が作った個人個人の“回転の動き”の作品ファイルの、円の大きさの平均をヒストグラムにしたものである。168.66 という大きい回転半径の園児もいたが、L10 にくらべて、より小さい“回転の動き”のプログラムを、園児は作っている傾向が見えた。

図 6.21 は L12 において、それぞれの園児が作った個人個人の“回転の動き”の作品ファイルの、円の大きさの平均をヒストグラムにしたものである。これを見ると、0.00 から 0.25 の回転半径のプログラムを作っている園児が多い一方で、L10、

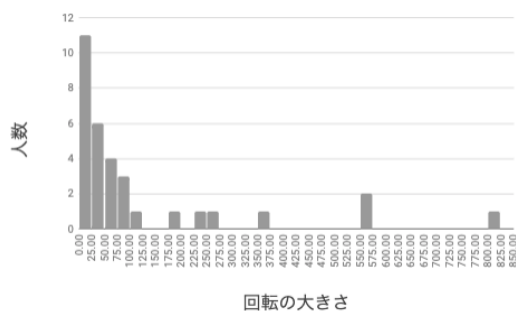


図 6.21: L12-F 回転半径の平均のヒストグラム

L11 にくらべて多くの園児が、回転の半径が大きいプログラムを作っていることが確認できた。

6.4 “ランダム動き”と“回転の動き”についての理解の分析のまとめ

“ランダム動き”については、L5-P1において、92.16%の園児がメガネを2つ以上使うプログラムを作っていた。一方で、課題である「おぼけ」が上下にゆらゆらするメガネを含むプログラムを作った園児は60.38%だった。第4章、第5章と同じように、課題が早く終わり、メガネを壊して遊んでいた園児がいるとも考えられるが、それを考慮したとしても、課題を遂行している園児は少なかった。

L5-P2について「カニ」「エビ」をそれぞれメガネを複数使って、生き物のように動かす課題においては、カニにおいて、取り組んだ園児の74%、エビにおいては54%の園児がメガネ複数使い、ランダムに動かしていた。一方で、メガネを複数使わずにプログラムを作っている園児が、「カニ」では20.83%、「エビ」では45.24%いた。これは絵をランダムに動かすということの楽しみよりも、まっすぐ動かす、また、速く動かす、または、ステージにたくさん置くということの楽しみのほうが強かった可能性がある。また、課題が自由度が高かったために、園児は何が求められているのかわからなかった可能性がある。一方で、数名の園児は絵を「止める」というテクニックに気づいた様子があった。

L6-P1においては、メガネ1つでプログラムを作っていた園児は1人だった。また、メガネを2つ以上使った園児のなかで、「くらげ」を上下に動かすメガネを含んだプログラムを作っていた園児は67.31%だった。ここでも、課題が早く終わり、メガネを壊して遊んでいた園児がいると考えられるが、一方で、2回目なのにもかかわらず、課題を遂行している園児は少なかった。

L6-P2に関しては、メガネを複数使う課題としては、自由課題を含めると5回目の課題だったのにも関わらず、20人（38.46%）の園児がメガネを複数使っていなかった。与えられた全ての絵に、“ランダム動き”をプログラムしていた園児は1名だけだった。ここまでの時点で、絵をランダムに動かすことの楽しさを、園児は感じられていない可能性が考えられた。また、用意された絵に関して、必ずしもランダムに動かなくても不自然でない絵だと、園児が受け取った可能性が考えられた。

自由課題においては、L5、L6ともに、メガネ2つ以上使ったプログラムを1つでも作った園児の割合は、それぞれ35.71%、67.31%であった。L5とL6を比べると、2回目のL6では、ランダム動きにチャレンジした園児は増えていることがわかる。しかし、大半が“ランダム動き”を理解し、プログラムしたとは言えない結果であった。また、それぞれの園児の作ったプログラムの総数の中で、“ランダム動き”を採用したプログラムの割合の全体平均を調べたところ、L5では24.30%、

L6では45.66%であった。この数字から、L6において、最低一つはランダムを作っている園児が67.31%いる一方で、それぞれの園児は、それよりも“直線の動き”を多く使っていることがわかった。

“直線の動き”よりも“ランダムの動き”のほうが、園児にとって難易度が高いと考えられる。L1-4では絵を複数用意しても、園児はそれぞれ方向通りに動かしていた。しかし、L5-6では園児は絵が複数になったときに、課題を遂行する割合が減っていた。授業者は1つの絵をランダムに動かす課題を増やし、園児の理解を導いた上で、用意された絵を増やす必要があると感じられた。また、自由課題に関しては、ランダムに動かす動機になるような課題を用意する必要性を感じられた。

“回転の動き”については、練習課題において、L10-P1, P2において、90%以上の園児が“回転の動き”のメガネを作れていた。P2において、絵が増えたのにも関わらず、“回転の動き”を使っている園児の比率が大きく変わらなかったのは、絵を回転させるということが、現実世界では難しいことであったため、園児たちは面白がり、そのモチベーションを引き出す題材になったからだと考えられる。

L10において、園児がどのような“回転の動き”のプログラムを作っているか分析したところ、回転半径は0-25がピークとなり、大きく回転させている園児は少なかった。また、傾ける角度についても、90度に傾けている園児が多い傾向が見えたが、それによって特別な特徴は引き出せなかった。

L11においては、中心を合わせる回転を園児は学んだ。P1においては、園児はL10と比べると、小さい回転半径で絵を動かしていることがわかった。絵の傾きについては、90度に傾けている園児が多い傾向が見えたが、それによって特別な特徴は引き出せなかった。L11-P2において多くの園児が、絵の中心に合わせて置く、その場での“回転の動き”のプログラムにチャレンジしている姿がわかった。一方で、5ポイント未満のずれの園児は70%であった。これは、タブレット上で絵の中心を合わせたとしても、指を離すときに絵がずれてしまっていることも考えられる。

L12においては、絵を大きく回転させることを学んだ。L12-P2においては、L10, L11と比べると、回転半径が大きくなっていることがわかった。また、L12-P2の絵の傾きを見ると、回転の角度が0に近いところにピークが来ており、園児は大きく絵を回転させることの理解が確認できた。

それぞれの自由課題において、多くの園児が回転を使っていた。L10においては、90%の園児が回転を使い、プログラムの中に占める回転のプログラムの数の平均は77.49%であった。L10のP1, P2で、多くの園児が回転を作っていることを考えると、園児は十分に回転を楽しみ、様々な絵を回転させようという意欲が感じられた。テーマが「ぐるぐるスーパー」と、園児の身近なスーパーマーケットを取り上げたこと、また、どんな回転でもよかったことから、園児が発想を広げやすかったのだと考えられる。

L11においては、98%の園児が回転を使っていた。また、それぞれのプログラムの中で、回転のプログラムが占める割合の平均は91.27%であった。作ったプロ

グラムのほとんどで園児は“回転の動き”を使っていたことがわかった。テーマが「風車」だったこともあり、テーマ自体が園児が回転を使うモチベーションに適切になっていたことがわかる。

L12においては、87%の園児が回転を使い、プログラムに占める割合の平均は62%だった。これはテーマが「流れ星」だったこともあり、直線で動かす園児が増えたのが原因だと考えられる。

それぞれの自由課題において、園児の作ったプログラムの回転半径の平均をヒストグラムにした。それぞれを比べてみると、L10に比べ、L11で作られたプログラムのほうが回転半径が小さいことがわかった。また、L12で作られたプログラムはL10, L11で作られたプログラムに比べて、回転半径が大きいプログラムが増えていることがわかった。これによって、一定数の園児は“回転の動き”のプログラムについて理解しており、自由課題においても、自分のプログラムに、練習課題で習得したテクニックを利用していることがわかった。

以上が“ランダムな動き”と“回転の動き”の分析のまとめである。

第7章

プログラムを使った表現の 分析

7.1 はじめに

本章では、13回目の最終レッスンで収集した、園児のプログラムの分析結果について述べる。最終レッスンでは、園児は卒園式で演じる劇（オペレッタ）の予告ビスケットランドを、保護者に見せるためにプログラムを作った。演じる劇はクラスによって違った。1つのクラスは「にゃんきちいっかのだいぼうけん」という演目を演じる予定であった。もう1つのクラスは「北風と太陽」という演目を演じる予定であった。

「にゃんきちいっかのだいぼうけん」のストーリーは、ある日ネコのお父さんが海岸で壺を釣り上げるところからはじまる [69]。その壺は綺麗に磨くと、表面に宝の場所が描かれている。ネコの家族はその宝を求めて海に冒険にでる。冒険にでると、ネコの家族は海賊に遭遇する。海賊はネコの家族に壺を渡すように要求するが、波に揺られて、壺を持ったネコの子どもが壺ごと海に落ちてしまう。ネコのお父さんは子どもを助けようと海に飛び込む。すると今度は、他のネコの兄弟が、お父さんとネコの子どもを助けようと海に飛び込む。それを見ていた海賊は感動し、海賊もネコの家族を助けようと飛び込む。最後は、全員が助かるが壺は海の中に消えてしまう。一方で、全員が改めて家族の絆を知る、というストーリーである。

「北風と太陽」のストーリーは、北風と太陽が旅人のマントを脱がせようとする物語である [105]。北風は風で旅人にマントを脱がせようとするが、旅人は寒いためますます身を固め、マントを脱がない。太陽は暖かくすることで、旅人のマントを脱がせようとする。旅人は暖かさにマントを脱いでしまう。

L1-L12は、それぞれのレッスンで、Viscuitのテクニクに基づいた課題が提示されていた。しかし、L13ではテクニクとしての課題の提示はなく、園児は自分たちが演じる演目を表現するためにViscuitを使った。L13では練習課題は用意は

せず、発表会の時間以外の時間をすべて制作に使った。

それぞれの節では「にゃんきちいっかのだいぼうけん」「北風と太陽」で園児が作った作品ファイルを分析する。そして、園児が与えられたテーマを表現するために、プログラミングをどのように活用するかを明らかにする。ここでいう表現とは他者に自分の考えを伝える、伝達を含んだ創作を意味する [26]。

園児は内容を表現するために、1つの絵に対して、1つ、または、複数のメガネを使っていた。そこで、L13では描かれた絵を基にし、その絵とメガネのセットを1つのプログラム表現として考えた。例えば、Aという絵が2つ以上のメガネを使って動きがつけられていた場合は、その絵とメガネをセットで1つの表現とカウントした。

プログラム表現を、1年間のレッスンで園児が習った“直線の動き”、“ランダム動き”、“絵の変化の繰り返し”、“回転の動き”の4つの学習内容で分類した。“絵の変化の繰り返し”については、一方向の変化のメガネも見られた。一方向の変化というのは $A \Rightarrow B$ という一方向の変化の命令のメガネである。L7からL9では $A \Rightarrow B$ に加え、 $B \Rightarrow A$ というメガネを作ることで、A, B, A, B, A, B,,, という変化が繰り返し起こる命令を学んだ。そのため、この一方向の変化は集計はしたが分析から除外した。

7.2 クラス A（にゃんきちいっかのだいぼうけん）

7.2.1 基本情報

分析の対象になった園児の数は28人であった。128個の作品ファイルが生成されていた。その中で作られたメガネの数は206個であった。

この206個について、収集した作品ファイルと実際のプログラムを付け合わせ、筆者が目視で以下の集計を行った。206個のメガネについて、それぞれを1つ1つの絵に対応させて集計した。そのときに、何も絵が入っていないメガネ、片側だけしか絵が入っていないメガネを省いた。その結果、画面上で絵を動かすために動作するメガネは196個であった。そのメガネを1つずつの絵に振り分けた結果、絵とメガネの組み合わせたプログラム表現は152個あった（表7.1）。

152個のプログラム表現を、“直線の動き”“ランダム動き”“絵の変化の繰り返し”“回転の動き”“衝突”の5つに従って分類した結果が表7.2である。“衝突”を除いて、これらのテクニックは通年のレッスンで教えられたものである。“衝突”に関しては、教えてないにも関わらず使っている園児がいた。

これらのテクニックについて、園児がどのテクニックを選ぶかに確率的な偏りがないという帰無仮説のもと、それぞれに二項検定を実施した。その結果、“直線の動き”と“回転の動き”は有意に多く選ばれていたのがわかった。“直線の動き”と“回転の動き”は、メガネが1つと絵が1つで作れるため、“絵の変化の繰り返し”よりも簡単なのだと思われた。その結果、園児はこれらのテクニックを“絵の変化

表 7.1: 分析したプログラムの数

項目	数
園児数	28
保存されたファイル (json)	128
ファイルの中のメガネの数 (rule)	206
有効なメガネの数	196
表現としてのプログラム	152

表 7.2: 使われたテクニックの分類

テクニック	人数	%	p-value
回転の動き	51	35.92%	$p < 0.05$
直線の動き	49	34.51%	$p < 0.05$
絵の変化の繰り返し	33	23.24%	$p > 0.05$
ランダムな動き	7	4.93%	$p < 0.05$
衝突	2	1.42%	Not executed

の繰り返し”よりも多く採用していたと考えられる。“衝突”に関しては、レッスン中で教えていないテクニックであるため、他の4つのテクニックと比べて、採用される確率は同じではないと考え、二項検定はしなかった。

“ランダムな動き”は適用された数が極端に少なかった。

“衝突”についてはレッスン中ではまったく触れていない。一方で、1人の園児がこのテクニックを使ってプログラム表現を作っていた。

152個のプログラム表現について、そのそれぞれにおいて、何の絵がモチーフとして使われているかを表したのが表 7.3 である。

モチーフとして採用された絵では「猫」が一番多く「?」「波」「壺」「船」が続いた。「?」は著者では何の絵かが判別できなかったものである。「その他」には描いた園児が2人以下の絵が集められている。「にゃんきちいっかのだいぼうけん」は、猫の一家が壺を頼りに、船で海へ宝を探しに旅にでるお話である。よって、登場するものがモチーフとして多く選ばれているのは、納得できる結果であった。

一方で、何を表している絵なのか判別できない絵が32個あった。これはプログラム表現全体の数152個のうちの21.05%に当たる。

表 7.3 には、総数の他にそれぞれの絵に対して、どのような動きのプログラム表現が何個あったかが示されている。表における“循環”は“絵の変化の繰り返し”を表し、“変化”は“絵の変化”の一方向で終わっているプログラムの数を表している。描かれた数が多かったものから順に「猫」は“絵の変化の繰り返し”が、「?」は“回転の動き”が、「波」は“回転の動き”が、「壺」は“直線の動き”、“絵の変化の繰り返し”と“回転の動き”が、そして「船」は“直線の動き”が多いのがわかる。それぞれ一番多くテクニックが採用された数字の横には「*」を表記した。

表 7.3: モチーフとして使われた絵とプログラム

絵	総数	直線	ランダム	循環	変化	回転	衝突
猫	38	12	1	13*	5	7	0
?	32	10	0	6	4	12*	0
波	21	4	1	0	0	16*	0
壺	13	4	1	4*	1	4*	0
船	12	6*	3	1	0	2	0
星	4	0	0	0	0	4*	0
キャンディ	3	0	0	1	0	2*	0
魚	3	3*	0	1	0	0	0
その他	28	11*	1	8	0	4	2

次に、それぞれのプログラム表現の詳細を“直線の動き”“ランダムの動き”“絵の変化の繰り返し”“回転”と分けて分析した。

7.2.2 直線の動き

“直線の動き”については全部で 49 個のプログラム表現があった。その中で、使われている絵が方向性を持っているかどうか、また、それぞれの絵がどの方向に動かされていたかをカウントしたのが表 7.4 である。

49 個の中で 32 個のプログラム表現が、絵自体が動きの方向を持たない絵を使っていた。そして、17 個が絵が動きの方向を持っている絵を使っていた。絵や絵の方向を持っているというのは「絵が（顔の向きなど）進行方向をもっている」または「他の絵との関係性で進行方向がわかる」ものを指す。ここで、上下左右のうちの一方向だけでなく、縦、または、横のどちらかに進む性質を持っている絵（例えば、船は左右どちらに進んでもよい場合があった）は、左右どちらかの方向性を持っている絵だとして、方向性がある絵だとカウントした。

図 7.1 は方向性を判断できる絵の例を示している。左の船の絵は、船が正面から描かれている。この場合、船は右か左に動くはずである。右のネコの絵は、ネコ単体であれば、どちらに動くのが最適かは判断しかねる。しかし、この場合は地平線が引かれているので、このネコは左右どちらかに動く必然性があると考えた。

絵が方向を持っていない絵というのは「？」の絵や、顔だけの絵である。それらの絵を動きの方向が推測できない絵としてカウントした。

その結果、方向性を持っている 17 個の絵は、全てその絵がもっている方向通りに動かされていた。

表 7.4: 絵と方向						
方向性	絵	Total	U	D	L	R
あり	船	6	0	0	6	0
	魚	4	0	0	1	3
	猫	3	0	0	1	2
	乗り物	2	0	0	1	1
	人	1	0	0	0	1
	波	1	0	0	1	0
なし	?	10	1	1	5	3
	猫	9	1	0	3	5
	壺	3	0	0	0	3
	身体の一部	2	0	0	0	2
	波	2	0	0	1	1
	その他	6	4	0	1	1

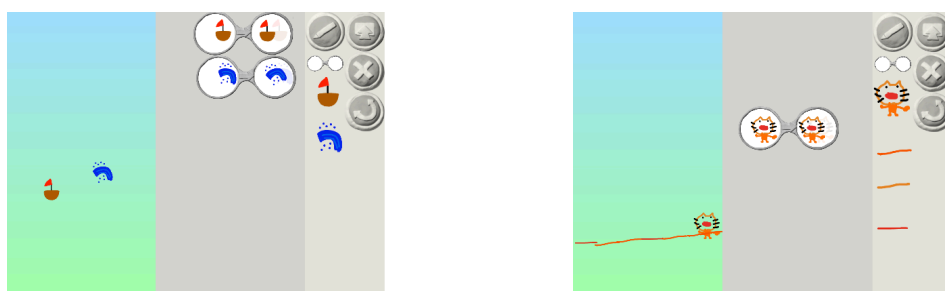


図 7.1: 直線の動きの例

7.2.3 ランダムな動き

“ランダムな動き”は前述のとおり，適用された絵が少なかった．全部で7個であった．詳細に一つ一つのプログラム表現を見てみると，7個中3個が船がゆらゆらと回転を伴って，波に揺られるように動いていた（図7.2）．他は猫が1個，波が1個，幽霊が1個，壺が1個だった．猫と幽霊は直線の動きを複数使ったランダムであった．そのほかは全て回転の動きが伴っていた．“ランダムな動き”をレッスン中で教えた当時は“直線の動き”を使った“ランダムな動き”のみを教えていた．12回のレッスンの中では“ランダムな動き”と“回転の動き”を組み合わせるプログラムは教えていないが，この表現を使っている園児がいることがわかった．船がゆらゆらしている表現は，ネコの家族が船で旅する様子を表そうとしているのだと考えられる．

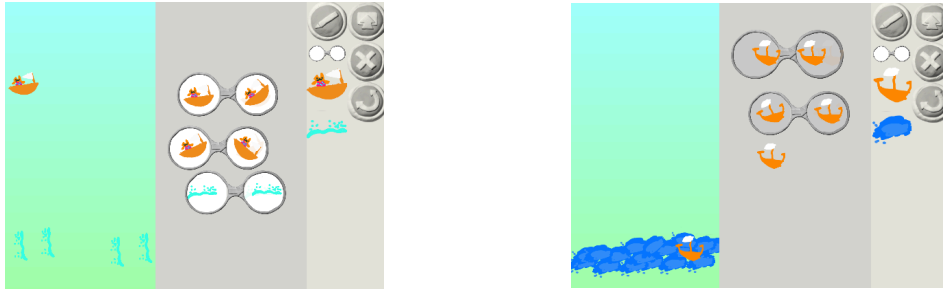


図 7.2: ランダムな動きが使われた例

表 7.5: 変化のモチーフと部分

絵	総計	変化する部分				
		表情	絵全体	色	口	その他
猫	13	10	2	1	0	0
?	6	0	5	1	0	0
壺	4	0	0	2	0	2
顔	2	0	0	0	2	0
その他	8	0	3	5	0	0

7.2.4 絵の変化の繰り返し

“絵の変化の繰り返し”は33個のプログラム表現があった。表7.5は“絵の変化の繰り返し”がどのような絵の、どの部分に適用されていたかを示している。“絵の変化の繰り返し”で一番使われたモチーフは猫であった。その次は「?」だった。また、猫をどのように変化させ、変化を繰り返させ続けていたのか確認したところ「猫」の「表情」が10個と一番多かった。

図7.3は「猫」が「表情」を変えているプログラム表現の例である。このように全く同じような絵を二つ描き、その部分を変えることによって「表情」を表現している。

また、壺に関して、壺の色が変化しているものが2つと、壺の形が変化しているものが1つ、また、壺の柄が変化しているものが1つであった。劇の中では、ネコの父親が汚れた壺を発見し、壺を磨くことでその表面に宝の地図を見つける。この壺の色の变化は、そのシーンを表現しているのだと考えられた。

L7-L9のレッスンではA, B, C, または、それ以上の絵を使って、 $A \Rightarrow B$, $B \Rightarrow A$ だけでなく、さらに長い“絵の変化の繰り返し”の作り方も教えていた。しかし、L13では3つの絵を使った“絵の変化の繰り返し”を作っていた園児は、1人だけであった。

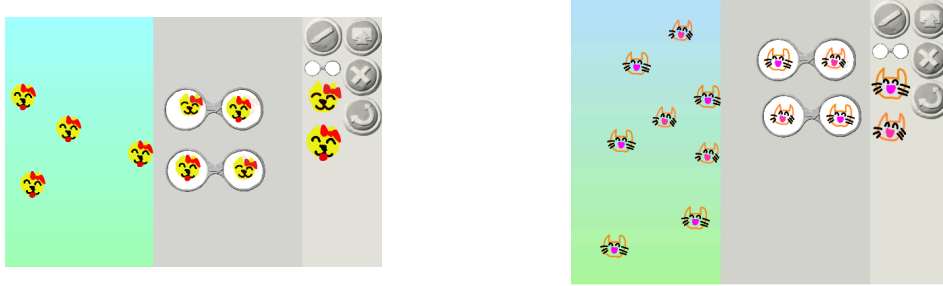


図 7.3: 猫の表情の変化の例

表 7.6: 回転のモチーフと動き

絵	総計	その場	小さく	大きく
波	16	6	4	6
?	12	6	3	3
猫	7	4	3	0
星	4	1	2	1
壺	4	4	0	0
キャンディ	2	1	1	0
船	2	1	0	1
その他	4	3	1	0

7.2.5 回転の動き

“回転の動き” のプログラムについては51個のプログラム表現があった（表7.6）。一番“回転の動き” が採用されていたのは「波」だった。また、Viscuitでは「大きい／小さい」回転を、絵のずらし方で作ることができるが、16個の「波」の回転の表現では、その「波」が大きく回転しているものが6個で一番多かった。この波は「にゃんきちいっかのだいぼうけん」中に出てくる。「にゃんきちいっか」が遭遇する大きい波を表していると受け取れる。

7.2.6 衝突

“衝突” のプログラムを使っているプログラム表現もあった。一つは、釣竿を持って移動するネコ（図7.4左）であり、もう一方は、船に乗った海賊（図7.4右）であった。これらのプログラムは2つの絵で作られており、2つはそれぞれ単独では動くことができない。2つの絵が図7.4のようにメガネの中に配置されたときだけ動く。これらのプログラムを作ったのは同じ園児であった。

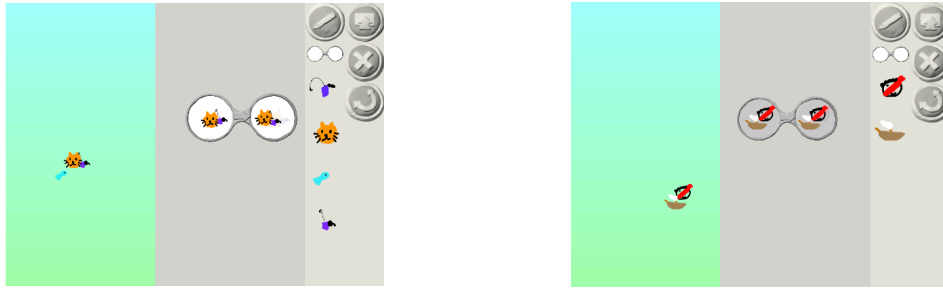


図 7.4: 衝突を使ったプログラムの例

表 7.7: テクニックの種類と内訳

テクニックの種類	合計	テクニック	人数
2	16 (57.14%)	M+R	8 (28.57%)
		R+CP	4 (14.29%)
		M+CP	2 (7.14%)
		RM+R	1 (3.57%)
		CP+Con	1 (3.57%)
3	6 (21.43%)	M+CP+R	5 (17.84%)
		M+CP+RM	1 (7.14%)
1	4 (14.29%)	M	1 (3.57%)
		RM	1 (3.57%)
		CP	2 (7.14%)
4	2 (7.14%)	M+CP+RM+R	2 (7.14%)

7.2.7 園児のプログラミング表現の数と種類

それぞれの園児の作ったプログラム表現について集計した。全ての園児は最低でも1つ以上のプログラム表現を作っていた。一番多く作っていた園児は、13個のプログラム表現を作っていた。作られたプログラム表現の平均は5.43個であった。

表 7.7 は園児たちがどのテクニックを選んだのかを示している。“M”は“直線の動き”，“RM”は“ランダム動き”，“CP”は“絵の変化の繰り返し”，“R”は“回転の動き”，そして，“Con”は“衝突”を表している。

16人の園児が2つのテクニックを使っており、特に“直線の動き”と“回転の動き”を使っていた。これは“直線の動き”と“回転の動き”が、絵が1つ、メガネが1つで作ることができる、一番簡単なプログラムだからだと思われる。4人の園児は1つのテクニックしか使っていなかった。2人の園児が教えられた全てのテクニックを使っていた。

これまでで、絵に対して、効果的にメガネが適応されているプログラム表現を確認した。しかしながら、作られたプログラム表現の数は、園児ごとに大きく違うのも確認した。よって、一部のViscuitのプログラミングが得意な子が、これら

の集計に大きく影響している可能性が考えられた。

そこで、先に確認してきた以下のプログラム表現を、園児が劇のシーンの再現として、効果的にプログラミングを使って表現した例とし、それらを作った園児の数を集計した。

- 動きの向きと絵の方向性がマッチしていた“直線の動き”：17点。
- 「猫」の表情の変化を表していた“絵の変化”：10点。
- 「船」のゆらゆらした動きを表していた“ランダムな動き”：3点。
- 「壺」の外観の変化を表していた“絵の変化”：2点。
- 「波」の回転を表していた“回転の動き”：16点。
- 2つの絵を使って条件づけられたプログラム：2点。

その結果、28人中24人（85.71%）の園児が、これらの表現に関わっていることがわかった（表7.8）。その中で、13人の園児が1つのテクニックで、10人の園児が2つ、1人の園児が3つの種類のテクニックを用いていた。

上記に該当するプログラム表現を作っていなかった、28人中の4人の園児に関して、彼らの作っていたプログラム表現を確認すると、どの園児も適切にプログラム表現は作っていた。彼らが作っていたものが、著者が選んだ上記の典型に当てはまるプログラム表現ではなかっただけであり、4人全員が劇になんらかの関係があるものを作っていた。

7.3 クラスB（北風と太陽）

7.3.1 基本情報

分析の対象になった園児の数は28人であった。28人が自由製作中に作成した作品ファイルは128個であった。その中で作られたメガネの数は225個であった。

この225個について、収集した作品ファイルと実際のプログラムを突き合わせ、幼稚園の教諭のアドバイスも受け、著者が目視で以下の集計を行った。225個のメガネについて、それぞれを1つ1つの絵に対応させて集計した。そのときに、何も絵が入っていないメガネ、片側だけしか絵が入っていないメガネを省いた。その結果、画面上で絵を動かすために動作するメガネは223個であった。そのメガネを1つずつの絵に振り分けた結果、絵とメガネの組み合わせでのプログラム表現は166個あった（表7.9）。

166個のプログラム表現を、“直線の動き”“ランダムな動き”“絵の変化の繰り返し”“回転の動き”の、4つに従って分類した結果が表7.10である。“直線の動き”と

表 7.8: 園児の分布

園児	種類	直線	ランダム	循環	回転	条件
A 児	1		1			2
B 児	2	2			1	
C 児	1	2				
D 児	1	3				
E 児	1				3	
F 児	2	1			1	
G 児	1	1				
H 児	1			1		
I 児	2	1			3	
J 児	2	1			1	
K 児	3		1	2	1	
L 児	1				1	
M 児	1	1				
N 児	2			3		
O 児	1			1		
P 児	1			1		
Q 児	1		1	2		
R 児	2	2		1		
S 児	2			1	1	
T 児	2		1		1	
U 児	2	1			1	
V 児	1				1	
W 児	1	1				
X 児	2	1			1	
合計		17	3	12	16	2

表 7.9: 分析したプログラムの数

項目	数
園児数	28
保存されたファイル (json)	137
ファイルの中のメガネの数 (rule)	225
有効なメガネの数	223
表現としてのプログラム	166

表 7.10: 使われたテクニックの分類

テクニック	人数	%	p-value
直線の動き	74	45.68%	$p < 0.05$
回転の動き	45	27.78%	$p < 0.05$
絵の変化の繰り返し	31	19.14%	$p < 0.05$
ランダムな動き	11	6.79%	$p < 0.05$
直線と変化の組み合わせ	1	0.62%	Not executed

“絵の変化の繰り返し”を組み合わせた“直線と変化の組み合わせ”を除いて、これらのテクニックは通年のレッスンで教えられたものである。“直線と変化の組み合わせ”に関しては、教えてないにも関わらず使っている園児がいた。どちらにもカウントできないので、1つの項目とした。

これらのテクニックについて、園児がどのテクニックを選ぶかに確率的な偏りがないという帰無仮説のもと、それぞれに二項検定を実施した。“直線と変化の組み合わせ”に関しては、レッスン中で教えていないテクニックであるため、他の4つのテクニックと比べて、採用される確率は同じではないと考え、二項検定はしなかった。

その結果“直線の動き”“回転の動き”は、有意に多く選ばれていたのがわかった。一方で、“絵の変化の繰り返し”“ランダムな動き”は有意に少なく選ばれていた。“直線の動き”と“回転の動き”については、メガネが1つと絵が1つで作れるため、“絵の変化の繰り返し”よりも簡単なことがその要因だと考えられた。“ランダムな動き”は適用された数が極端に少なかった。

166個のプログラム表現について、そのそれぞれにおいて何の絵がモチーフとして使われているかを表したのが表 7.11 である。モチーフとして採用された絵では「北風」が一番多かった。続いて「太陽」「花」「旅人」「？」が多かった。「？」は何の絵か判別できなかったものである。「その他」には描いた園児が2人以下の絵が集められている。演じられたオペレッタ「北風と太陽」では、「花」は「太陽」の仲間として、「雪」は「北風」の仲間として登場する。よって、それぞれに登場するものがモチーフとして多く選ばれているのは、納得できる結果であった。

一方で、何を表している絵なのか判別できない絵が20個あった。これは全体の数166個のうちの12.05%に当たる。

表 7.11 には、総数の他に、それぞれの絵に対して、どのような動きのプログラムが適用されたかが示されている。表における「循環」は“絵の変化の繰り返し”を表し、「変化」は“絵の変化”の一方向で終わっているプログラムの数を表している。描かれた数が多かったものから順に「北風」は“絵の変化の繰り返し”が、「太陽」は“直線の動き”が、「花」は“回転の動き”が、「旅人」は“直線の動き”が、「？」も“直線の動き”が、そして、「雪」も“直線の動き”が多いのがわかる。それぞれ

表 7.11: モチーフとして使われた絵とプログラム						
絵	総数	直線	ランダム	循環	変化	回転
北風	37	12	3	10*	1	12
太陽	27	8*	1	7	2	9
花	22	7*	1	4	0	10
旅人	21	14*	1	2	0	4
?	20	13*	1	0	0	6
雪	16	10*	2	1	1	2
パクパク	5	0	0	5	0	0
虫	3	0	1	0	0	2
その他	15	10	1	3	0	1

一番多くテクニックが採用された数字の横には「*」を表記した。

次に、それぞれのプログラム表現を“直線の動き”“ランダムの動き”“絵の変化の繰り返し”“回転の動き”と分けて分析した。

7.3.2 直線の動き

“直線の動き”については全部で74個のプログラム表現があった。その中で使われている絵が方向性を持っているかどうか、また、それぞれの絵がどの方向に動かされていたかをカウントしたのが表 7.12 である。

74個の中で、55個のプログラム表現が、絵自体が動きの方向を持たない絵を使っていた。そして19個が、絵が動きの方向を持っている絵を使っていた。絵や絵の方向を持っているというのは「絵が（顔の向きなど）進行方向をもっている」「他の絵との関係性で進行方向がわかる」ものを指す。また、同じ雪の絵でも、雪自体の絵と、キャラクターとしての雪の絵があった。そこで、雪自体の絵としては、下に動く必然性を持っていると考えた（図 7.5 左）。また、キャラクターとして描かれた雪は、絵自体に方向性が見つけられないため、方向性がない絵にカウントした。「?」の絵や、正面を向いた顔だけの絵を、動きの方向が推測できない絵としてカウントした。

その結果、方向性を持っている19個の絵のうち、15個の絵（78.95%）がその方向通りに動かされていた。

7.3.3 ランダムの動き

“ランダムの動き”は前述のとおり、適応された絵が少なかった。全部で11個であった。詳細に一つ一つのプログラムを見てみると、11個中5個が“回転の動き”と“ランダムの動き”を一緒に使っており、また、同じく11個中5個がメガネを3

表 7.12: 直線の動かされた絵と方向

方向性	絵	総数	上	下	左	右
あり	北風	11	1	1	9	0
	雪	7	2	5	0	0
	旅人	1	0	0	1	0
なし	?	13	1	1	7	4
	旅人	13	0	2	9	2
	太陽	8	2	3	1	2
	花	7	0	2	1	4
	雪 (キャラクタ)	3	1	0	1	1
	花と太陽	2	0	0	2	0
	猫	2	1	0	1	0
	その他	7	1	0	2	4

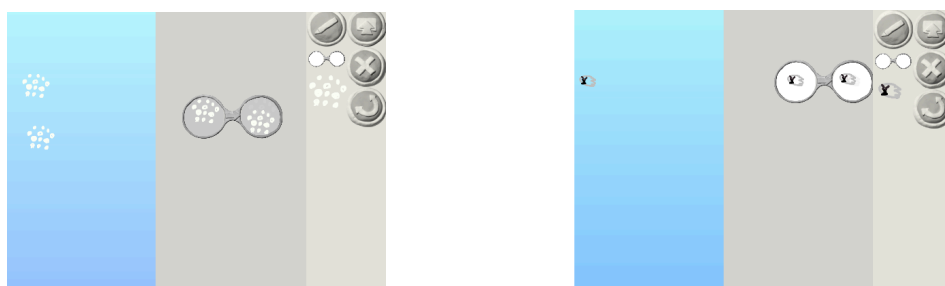


図 7.5: 直線の動きにおける方向のマッチしている例

つ以上使って、ランダムに絵を動かしていた。絵の内訳としては、「北風」が3個、「雪」が2個であった。そのほか、「?」「花」「顔」「太陽」「虫」「旅人」が1つずつであった。「ランダムの動き」をレッスン中で教えた当時は「直線の動き」をつかったランダムのみを教えていた。よって、12回のレッスンの中では「ランダムの動き」と「回転の動き」を組み合わせるプログラムは教えていないが、この表現を使っている園児がいることがわかった。「北風」や「雪」がランダムに動いている様子は、嵐の様子を表現しているとも考えられたが、シーンを効果的に表せていると言い切れる表現は見られなかった。

絵の変化

“絵の変化の繰り返し”は全部で32個だった。表 7.13 は“絵の変化の繰り返し”がどのような絵のどの部分に適用されていたかを示している。“絵の変化の繰り返し”で一番使われたモチーフは「北風」であった。その次は「太陽」だった。「北風」をどのように変化させ、繰り返させているかを確認したところ、「北風」の「口」が

表 7.13: 変化のモチーフと部分

絵	総計	変化する部分				
		口	表情	色	パーツ	その他
北風	10	6	0	2	0	2
太陽	7	0	4	2	0	1
パクパク	5	4	0	1	0	0
花	4	0	2	1	1	0
旅人	2	0	0	1	1	0
その他	4	0	0	0	0	4

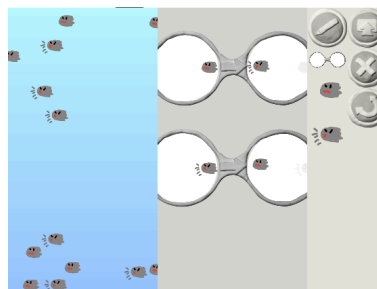


図 7.6: 雲の口が変わるプログラムの例

6 個と一番多かった。次に多かったのは「太陽」の「表情」の変化であった。「太陽」と「太陽の絵の顔の一部を変えた絵」を描き、繰り返し変化させていた。

図 7.6 は「北風」が「口」を変えているプログラム表現の例である。このように全く同じような絵を二つ描き、その部分を変えることによって「口」で風を吹いているシーンを表現している。図 7.7 は「太陽」「花」が表情を変えているプログラム表現の例である。

L7-L9 のレッスンでは A, B, C, または、それ以上の絵を使って、 $A \Rightarrow B$, $B \Rightarrow A$ だけでなく、さらに長い循環の作り方も教えていた。しかし、2 つの変化以上の長さの変化の繰り返しを作っている園児は、2 人だけだった。その内 1 人は、ストーリーの場面の絵を、5 個のメガネを使って紙芝居のように表現していた（図 7.8）



図 7.7: 表情の変化の例

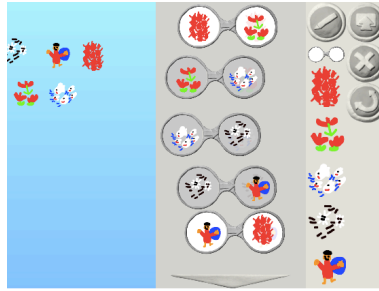


図 7.8: 紙芝居の例

表 7.14: 回転のモチーフと動き

絵	総計	その場	小さく	大きく
北風	11	5	5	1
花	10	9	0	1
太陽	9	9	0	0
?	6	6	0	0
旅人	4	3	1	0
雪	2	0	2	0
虫	2	0	2	0
木の葉	1	0	1	0

7.3.4 回転の動き

“回転の動き”のプログラムについては45個のプログラム表現があった(表7.14)。一番“回転の動き”が採用されていたのは「北風」だった。Viscuitでは「大きい／小さい」回転を、絵のずらし方で作ることができる。しかし、L13のクラスBにおいては、多くの絵がその場で回転していた。また、大きく回る北風は、L11の流れ星を大きく回したように、綺麗に回転しており、これは北風が吹き荒れている様子を表現していると捉られた(図7.9)。

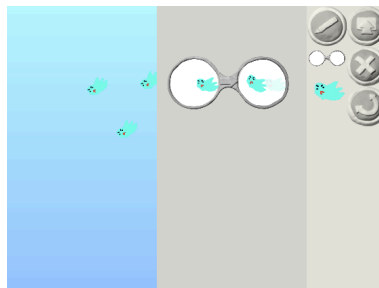


図 7.9: 回転する風の例

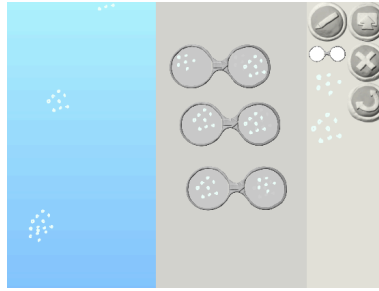


図 7.10: 直線の動きと絵の変化で雪が舞っている姿を表している例

表 7.15: Group details

テクニックの数	合計	テクニック	人数
2	18 (64.29%)	M+R	9 (32.14%)
		M+CP	6 (21.43%)
		R+CP	1 (3.57%)
		RM+R	1 (3.57%)
		CP+Mix	1 (3.57%)
3	5 (17.86%)	M+RM+R	2 (7.14%)
		M+R+CP	2 (7.14%)
		M+RM+CP	1 (3.57%)
1	4 (14.29%)	M	2 (7.14%)
		R	1 (3.57%)
		CP	1 (3.57%)
4	1 (3.57%)	M+CP+RM+R	1 (3.57%)

7.3.5 “直線の動き”と“絵の変化”を組み合わせたプログラム表現

“直線の動き”と“絵の変化”を両方使うことによって「絵が動いたり，変わったる」ように絵を動かしている園児が1名いた．このプログラムは雪の絵に対してつけられていた（図 7.10）．雪が舞っている姿を表しているのだと考えられた．“直線の動き”としては左下方向につけられており，雪の動きとして妥当なものだった．

7.3.6 園児のプログラミング表現の数と種類

それぞれの園児の作ったプログラム表現について集計した．全ての園児は最低でも1つ以上のプログラム表現を作っていた．一番多く作っていたのは13個作った園児であった．作られたプログラム表現の平均は5.43個であった．

表 7.15 は園児たちがどのテクニックを選んだのかを示している．“M”は“直線の動き”，“RM”は“回転の動き”，“CP”は“絵の変化”，“R”は“回転”，そして，

“Mix”は“直線の動き”と“絵の変化”を合わせて使っているものを表している。

17人の園児が2つのテクニックを使っており、特に“直線の動き”と“回転の動き”を使っていた。4人の園児は1つのテクニックしか使っていなかった。1人の園児が教えられた全てのテクニックを使っていた。これは“直線の動き”と“回転の動き”が、メガネ1つ、絵1つでできる、一番簡単なプログラムだからだと思われる。

クラスBにおいても、作られたプログラム表現の数は園児ごとに大きく違うことが確認できた。そこで、クラスAと同じように、先に確認した以下のプログラム表現を、園児が劇のシーンの再現として、効果的にメガネを使ってプログラムを使った例とし、それらを作った園児の数を集計した。

- 絵の方向性と動きの向きがあっていた“直線の動き”：15点。
- “北風”の口の形の変化を表した“絵の変化”：6点。
- “太陽”の表情の変化を表した“絵の変化”：4点。
- “花”の表情の変化を表した“絵の変化”：2点。
- 絵のストーリーを紙芝居のように表した“絵の変化”：1点
- 北風の大きい動きを表した“回転の動き”：1点。
- 絵の変化と動きを組み合わせた雪のプログラム：1点

その結果、28人中18人(64.28%)の園児が、これらの表現に関わっていることがわかった(表7.16)。また、これらの表現に関わっていない、10人の園児のプログラムを確認したところ、「北風と太陽」に関連があるプログラムを作らなかった園児が2名、あとの8名はなんらかの関連があるプログラムを作っていた。また、関連のないプログラム表現のみを作っていた2名のうち、1名は“直線の動き”と“絵の変化の繰り返し”のテクニックを、1名は“絵の変化の繰り返し”のテクニックのみを使っていた。課題に基づいた表現としては適切ではないが、プログラムとしては、有意に適用されることが少なかった“絵の変化の繰り返し”を使っており、Viscuitを理解していると考えられた。オペレッタに関連のあるプログラム表現を作っていた8名は“直線の動き”と“回転の動き”を使っている園児が多かった。そして、それらの絵自体の方向性がはっきりしないプログラムが多かった。つまり、これらの園児については、著者からは判断ができないが、園児たちは思い通りの方向に絵を動かしている可能性は考えられた。

表 7.16: 園児の分布

園児	種類	直線	循環	回転	混合
A 児	1		1		
B 児	1	1			
C 児	1	1			
D 児	1	1			
E 児	1	4			
F 児	1				1
G 児	1	1			
H 児	1		4		
I 児	2	1	1		
J 児	1	1			1
K 児	1		2		1
L 児	2	2	1		
M 児	1	1			
N 児	1	1			
O 児	2	1	1		
P 児	2		1	1	
Q 児	1		1		
R 児	1		1		
合計		15	13	1	1

7.4 保育日誌

授業者である幼稚園の教諭は、毎回のレッスンで保育日誌をつけている。この最後のレッスンに関しては、保育日誌を確認することで、幼稚園の先生から見た最終レッスンについても言及したい。保育日誌には以下が書かれていた。

- 今までで一番一人一人が違う動きを作って個性のあるビスケットランドになっていたように思う。
- テーマが自分たちが演じる劇だと伝えると大喜びをして夢中で作っていた。
- 保護者に「見せるもの」として作る意識もしていたように感じる。
- 役全員を描こうとする子どもや、自分の役をたくさん描く子どももいれば、「お話には出てこないけれど、お話の世界にはいるかもしれない」と、想像して自由に書く子どももいた。

この日誌から、園児は Viscuit によって想像力が刺激されている様子がうかがえた。また、他者に伝える表現をすることが園児のモチベーションを引き出している様子もうかがえた。

7.5 プログラムを使った表現の分析のまとめ

1 年を通じて、園児は Viscuit の使い方を学んだ。最後のレッスンでいままで学んできた“直線の動き”“ランダム動き”“絵の変化の繰り返し”“回転の動き”のプログラムを、卒園式で演じる自分の劇を他者に伝えるために使った。本章では、園児がどのようにこれらのテクニックを使うかを確認した。

それぞれのクラスで作られた表現としてのプログラムは 152 個と 166 個であった。作られたメガネの総数の中で、無効なメガネとしてカウントされなかったものは、それぞれのクラスで 10 個と 2 個であった。また、メガネをテクニックで分けた際、不完全の一方通行の“絵の変化”に関しては、それぞれのクラスで 10 個と 4 個であった。ここから、園児が全体としてメガネの使い方を習得し、無駄にメガネを作るのではなく、作る意図を持って必要なメガネを出している姿がうかがうことができた。

それぞれのクラスでのメガネの用途は、クラス A では“回転の動き”が一番多く、2 番目が“直線の動き”であった。それぞれ、有意に多く選ばれていたのがわかった。クラス B では“直線の動き”が一番多く、2 番目が“回転の動き”であり、これらも有意に多く選ばれていた。これらはメガネ 1 つ、絵も 1 つで作成することができ、一番初歩のテクニックだからだと考えられる。

また、どちらのクラスでも“ランダム動き”の利用が少ないことがわかった。これに関しては、L5 と L6 で、園児が十分にアイデアを表現できていなかったこと

に原因があると思われる。第6章で分析した結果が、このテクニックの採用率に関連があると考えられる。第1.1.3項で佐伯が指摘した「なじむ」ところまで、このテクニックを味わうことが、レッスンの中でできていなかったのだと考えられる。

“絵の変化の繰り返し”については、クラスAにおいては適用された数に統計的な有意はなかったが、クラスBでは有意に少なく選ばれていた。“直線の動き”“回転の動き”に比べて少なく選ばれる理由は、最低でも絵を2つ、メガネも2つ使うからであり、園児にとっては理解があったとしても、実際に自由な状況でプログラムをするには、より手軽に作れる“直線の動き”“回転の動き”を選ぶ傾向が見られた。

“ランダムな動き”が少なかった一方で、“ランダムな動き”が適用された中には、レッスン中に教えていない組み合わせを使って、プログラムを作っている園児が複数確認された。クラスAにおいて、船のゆらゆらに効果的に活用されている例が見られた。この組み合わせはクラスBにおいても見られた。しかし、クラスBにおいては、それが表現として効果的に作用している例は見られなかった。また、クラスAにおいては1名が、教えられていない“衝突”のテクニックを使っている園児もいた。先行研究において、Louiseは、具体的思考期の子どもは教えなくても探求していける姿が見られると報告していたが、その報告と一致するものである[14]。

クラスAでは、それぞれのテクニックについて、絵に対して一番多く適用されたものが、表現として効果的なものであったことを確認した。一方で、クラスBでは“ランダムな動き”と“回転の動き”に関しては、一番作られたものが効果的な表現だとは言えなかった。園児の選ぶテクニックの比率や、適用したテクニックの種類の数などは、クラスA、クラスBで同じであった。よって、選ばれた劇の題材が影響しているのではないかと考えられる。「にゃんきちいっかのだいぼうけん」は明確に移動する物語であるが、「北風と太陽」は移動ではなく、コミュニケーションと状態の変化の物語である。Viscuitで表現するものには移動や動きが伴ったものが適している可能性が考えられた。第6章の“ランダムな動き”のレッスンでも、自由製作時の課題の設定の仕方は園児のプログラムに影響する可能性が見られた。このことから、園児に自由にプログラミングをさせる場合は、制作環境の設定が大きく関わる可能性が見られた。また、ここから、先に“絵の変化の繰り返し”の適用が少ないことを述べたが、これに関しても、自由課題の設定によって、選ばれる比率が変わる可能性があると考えられる。

すべての園児が1つ以上のプログラム表現を作っていた。クラスAにおいては、85.7%の園児が効果的なプログラムに関わっていた。クラスBにおいては、64.6%の園児が効果的なプログラムに関わっていた。これらを平均すると、クラスA、クラスBで74.99%の園児が効果的なプログラムを作っていると言える。今回の研究対象の園児の中で、多くの園児がViscuitを使った表現ができていると言える。また一方で、これらに入っていない園児も、それぞれでプログラムを作れていることがわかった。保育日誌には、絵本のストーリーの外側の、大人が思いも寄らな

い世界観を作ろうとしている園児もいたことがわかった。より詳細に、園児の表現活動としてのプログラミングを見るときは、プログラムだけでなく、制作中に「なにを作ったの?」「なんで?」などのインタビューを並行して行う必要性も感じた。これは第5章「まとめ」でも明らかになったことである。

保育日誌には、今までで一番一人一人が違う動きを作って個性のあるビスケットランドになっていたように思う、と書かれていた。本カリキュラムの目標は、全ての園児がプログラミングを自分のものとして、自分なりの表現ができるようになることだった。園児のプログラム表現の分析と、保育日誌を確認することで、大半の園児自分のアイデアでプログラムをし、表現できていたことがわかった。一方で、プログラム分析だけで園児の意図を汲む限界もわかった。

第8章

本論文の総括

8.1 本論文のまとめ

第1章においては本研究の背景を確認し，リサーチクエスションと目的を確認した．情報社会の広がりにより，未就学児の頃から情報機器に触れる機会が増えてきている．その中で，早期からプログラミングを体験し，情報の原理的なものを知り，興味関心を育む必要があった．乳幼児の教育においては，具体的なものを使い，また，乳幼児自身が対象に触り，問いかける中で意味を見出す学びが必要であった．また，乳幼児は対象に意味を見出すと，それを使って他者に向かって表現したくなることがわかった．こういった乳幼児の学びのプロセスから，未就学児におけるプログラミング教育においても，未就学児がプログラミングの意味を探求し，表現する学びが必要であることを導いた．その上で，本研究のリサーチクエスションは，未就学児のプログラミング教育を考えたときに，プログラミングが未就学児の表現のツールになるかどうか，であった．このことを問題にいた上で，2つのことを明らかにすることが本研究の目的であった．

- 未就学児はプログラミングを理解できるか．
- 未就学児がプログラミングでどのような表現をするか．

いままでは情報機器・端末が高価であったため，これらの研究は困難がともなった．しかし，情報社会の広がりにより，タブレットは普及しており，また，タブレット上でプログラミングが可能なツールも開発されている．そして，すでに民間企業・施設において，未就学児に対してのプログラミング体験を実施するところもある．今こそ未就学児のプログラミングの理解と表現について明らかにされるべきであった．

第2章においては先行研究を確認した．はじめに子どもを対象にしたプログラミング教育ツール，子どもがプログラミングを通して学ぶこと，未就学児のプログラミングの理解について確認した．その上で，本研究で用いている Viscuit に関

する研究を確認した。現在の未就学児のプログラミング教育でわかっていることは、ロボットを使った研究が多いということである。今後、さらなるタブレットの普及により、未就学児のプログラミング体験が増えるのはスクリーン上での体験だと予測される。また、ロボットを使った研究が多いのは、未就学児の発達段階的に具体的な学びが必要であるからだと考えられた。そして、その上で、図形書き換え型言語である Viscuit は具体的な言語であり、それにも関わらず表現の幅が広いことを確認した。1960 年代後半に Papert はコンピュータの教育における意義は、抽象的なものが具体的になることであると指摘していた。現在のプログラミング教育では、抽象的な思考ができるようになることが、その意義として言われることが多い。しかし、具体的な命令によって、スクリーン上で様々なプログラミング体験ができる Viscuit は、未就学児にとっては望ましいプログラミング体験であることを確認した。

第 3 章では本研究の概要を述べた。研究を実践した幼稚園、レッスンの概要、カリキュラム、そして、プログラムの収集と分析方法について説明した。その中で、レッスンの実施方法やカリキュラムの作成において、未就学児の行動と理解を促すために取り入れた、様々な工夫について述べた。

第 4 章、第 5 章、第 6 章では、園児のプログラミングの量的な理解について、分析した結果を述べた。“直線の動き”、“絵の変化の繰り返し”、“回転の動き”に関しては、多くの園児がプログラムを作れており、また、与えられた絵に対して適切にプログラミングをしていることがわかった。一方で“ランダムな動き”については、一定の理解が見られるものの、自由にプログラミングをさせた場合は、その活用率が低かったことがわかった。

第 7 章では、1 年間行ったレッスンの最終レッスンにおいて、園児がどのようにプログラミングで表現をするのかを分析した。園児は表現をするために非常に動機付けられており、また、それぞれの園児がそれぞれの園児なりにプログラミングで表現をしていることがわかった。

ここで、第 4 章から第 7 章を踏まえた上で、第 3 章において挙げた、本実践における工夫について考察を行う（表 8.1）。「理解の促進」を目的にした工夫として、繰り返し同じプログラミングテクニックを扱い、進度を早めるのではなく、絵を変えることで何度も同じことを実施した。L1-L12 の中では、同じ内容を実施したことによって、プログラムを正確に作れている園児の数が増えている場合が多く見られた。よって、多くの園児の理解に、繰り返し実施することが効果をしめし、また、理解が早い園児も、飽きずにプログラミングに取り組むことができたと考えられる。

また、動きの方向や性質をもった絵を予め用意しておくことによって、園児は自然と絵に対してプログラムを作ることができていた。佐伯は、乳幼児は対象に共感することを通して、そのものの性質を考え、自分なりの考えをもって、対象に接するという [79]。この性質と照らし合わせると、園児はその絵が「どうしてもらいたいのか」を共感し、その絵の性質に合わせてプログラミングをすることに

表 8.1: レッスン内で採用した工夫

目的	工夫内容	効果
理解の促進	繰り返し同じことをやる	学習の定着が見られた
	絵をあらかじめ用意しておく	絵に共感する様子が見られた
	スペースを分ける	課題を明確に伝えられた
	発見を促す	それぞれの文化的実践につながった
表現の誘起	自由課題時間を多めにとる	なじませる時間になった
	各回に自由課題を入れる	なじませる時間になった
	グループ制作を採用	多元的に、表現の下地を醸成した
	ビスケットランドの活用	多元的に、表現の下地を醸成した
	各回に発表会を設ける	多元的に、表現の下地を醸成した
活動の保障	一人一台のタブレット	干渉や邪魔されることなく取り組めた

よって、プログラミングを学ぶことができたと考えられる。実際にそれぞれのレッスンの中では、絵の性質と違う動きを教諭が見せた時、園児から「ちがう」という明確な指摘が何度もあった。Pepart はプログラミングが抽象的なことを具体的にすることで学びやすくする、といった。本研究における実践では、対象が具体的に becoming になることによって、園児は対象に共感がしやすくなったと考えられる。そして、結果として、園児はより自然にプログラムを作る方法を学ぶことができたと考えられる。

「スペースを分ける」工夫によって、タブレットが前にあると集中できない園児や、落ち着いて座って話を聞くことができない園児も、教諭の話を聞く時間とプログラムを作る時間の切り替えができ、理解の向上につながった。レッスン中では極力、教えることを避け、自分で発見することを促した。これに関しては、客観的指標に基づいた効果は指摘できない。しかし、先の共感からの学びを考えると、教えた通りやるのではなく、共感に基づいて園児が自らプログラミングをする、というプロセスにできたのではないかと考えられる。

「表現の誘起」を目的にした工夫に関しては、プログラミングに対して園児が自分なりの意味や価値を見出すために、多く自由課題の時間をとった。これによって、L13においても、園児は自分なりのアイデアでプログラミングに臨むことができたのではないかと考えられる。また、園児を班に分け、ビスケットランドを利用し、それぞれが作ったプログラミング作品を見合うことができるようにした。レッスン中は園児同士が「見て、見て」と声をかけあったり、自分の作品がビスケットランド画面にでると喜ぶ姿が何度も確認できた。また、「見て、見て」と促された園児はその園児の作品を見、反応をしめし、影響を受けている様子も見られた。これによって、プログラミングの理解が高まったのみならず、園児同士に「こんなアイデアを出してもいいんだ」という雰囲気醸成できたと考えられる。

そして自分のアイデアを他者に対して、表現しやすい雰囲気を作ることができたのではないかと考えられる。

最後に「活動の保障」として、1人1台のタブレットを用意してレッスンに臨んだ。これによって、他者に干渉・邪魔されることなく、それぞれの園児がプログラム作りに集中してもらうことができたと考えられる。そして日本学術会議の示す指針の通り、自分のプログラムが動いている実感を与えることができたのではないかと考えられる。

また、第3.3項において、カリキュラムを説明した上で、本カリキュラムが実施された後の園児のプログラムの特徴を下記のように予測した。

1. メガネを作成できる
2. メガネの数は最小限になる
3. 絵に対して妥当な動きをつける

第4章、第5章、第6章、および、第7章を鑑みると、園児はメガネによって絵の性質に合わせたプログラムを作っていた。そして、L13においては、それぞれのプログラミング表現には無駄なメガネはほとんど確認できなかった。これらから、園児は、非常に初歩の段階の原理的なことだが、「コンピュータはプログラムで動く」「プログラムは命令された通りに動く」という情報の原理的なことを理解したといえる。

本研究を通して、目的と照らし合わせると結論は下記である。

- “直線の動き”，“絵の変化の繰り返し”，“回転の動き” に関しては，未就学児は理解し，適切にプログラミングをすることができる。
- 具体的なプログラミング言語を使うことによって，未就学児はプログラミングを使った表現をすることができる。

また，以上から，プログラミングが未就学児にとっての表現のツールになると結論できる。

本研究の各レッスンでは，園児たちは図8.1のように，CTPを繰り返したと考えられる。園児たちはプログラミングレッスンにより，具体的な絵の配置によってコンピュータに命令する体験を繰り返した。その過程でプログラミングが園児の手に「なじむ」道具になった。この繰り返しが可能であったのは，第2章の図2.22で見た通り，手続き型言語のプログラムとViscuitのプログラムのCTP単位の違いが原因と考えられる。Viscuitのプログラムは小さいCTPの単位でプログラムを作ることができるため，完成が早い。よって，完結したCTPを何度も体験できたのだと考えられる。また，そのCTPの繰り返しのなかで，園児たちは様々な与えられた絵に「共感」することによって，その絵を通したプログラムの世界を様々な角度から見たと考えられる。それゆえに，様々な活用の視点をもつことができ，L13において，それぞれの園児なりの表現が生まれたのだと考えられる。

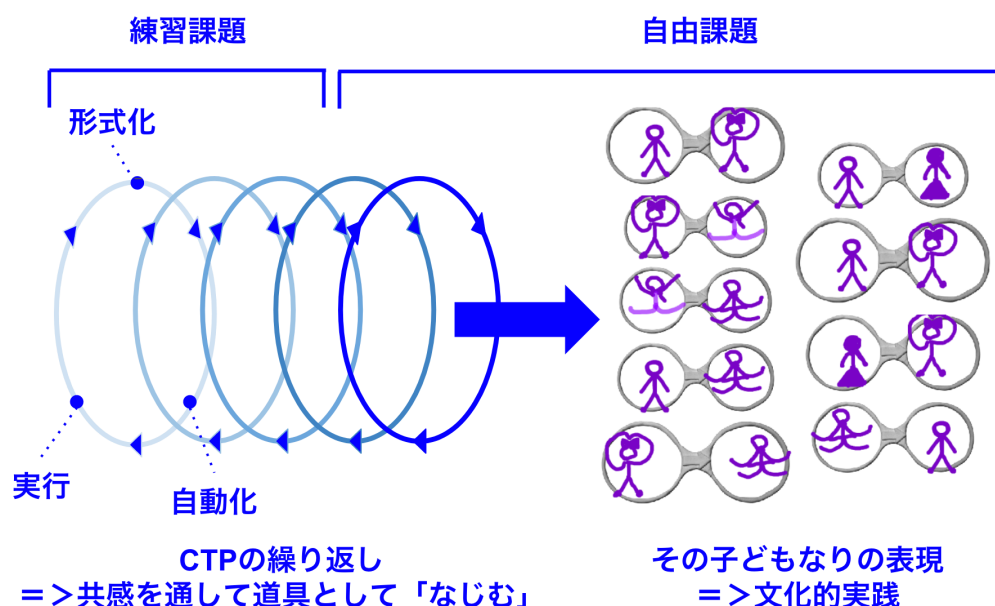


図 8.1: L1-12 のそれぞれのレッスンの体験モデル

図 8.1 が各レッスンで園児たちがコンピュータを使ったプロセスだとする。図 8.2 では、その各レッスンが YOU 的な、なじんだプログラミングとの関係を生成する役割を担っている。そして、その結果として、L13 における文化的実践的な表現がでてきている。一方で、図 8.1 で見られるとおり、各レッスンにも文化的実践的な、各テクニックに基づいた自由課題が内包されていた。このように、結果的に「なじむ～文化的実践」までの活動が入れ子になっていたカリキュラムデザインだと振り返ってみると考えられる。

未就学児はコンピュータの原理的なものを、具体的なプログラミングによって、体験し、学ぶことができると言える。また、プログラミングが表現のツールになることから、我が国において、幼稚園を含む未就学児の教育でもプログラミング実践が可能であることがわかった。

8.2 本研究の理論的な貢献

本研究では、Papert の指摘する、教育におけるコンピュータの意義は、抽象的で学びにくいものを具体的にすることだ、という指摘から図形書き換え型言語を見直した。また、未就学児における学びにおいては、具体的なものが好ましいという発達研究の知見からも、具体的である図形書き換え型言語の価値を再考した。さらに、未就学児における教育においては、未就学児自身が意味を探求し、発見できる体験が不可欠であることを論じた。未就学児のプログラミング教育を考えた時、この点を解決するために、図形書き換え型言語の発展の現在形である Viscuit について論じた。Viscuit は具体的であると同時に、非常に表現力が高い言語であ

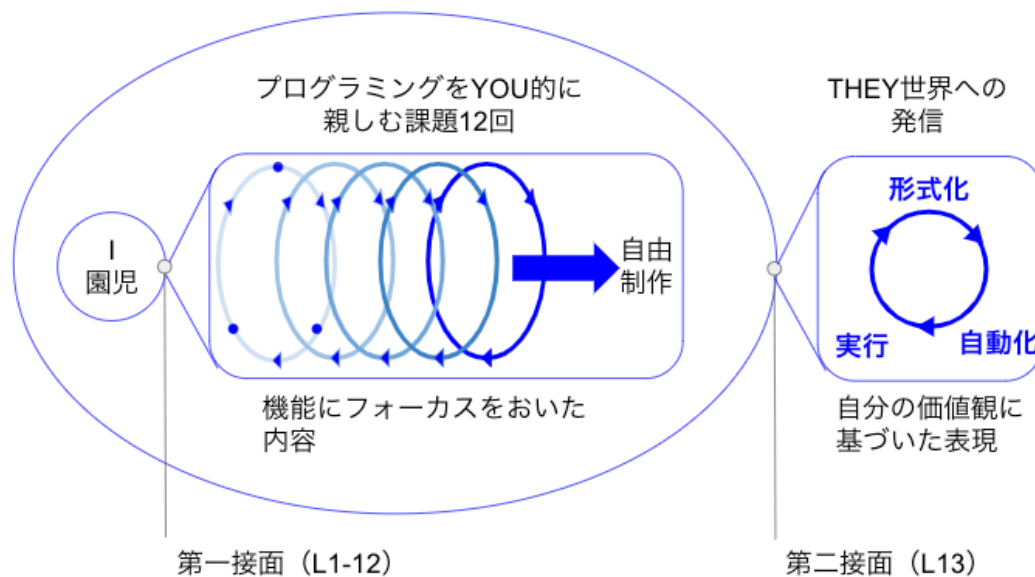


図 8.2: 本研究全体の体験モデル

る。そして、プログラミングを学ぶ意義についての議論の中から、コンピュータと人間の特性を相互に体験する CTP の理論を取り上げた。この具体的なプログラミングの見直しと、文化的実践の重要性から導き出した表現の重要性、そして、プログラミングにおけるプロセスの繰り返し（CTP）を、未就学児のプログラミング教育に適用したことが、本研究の第一の理論的貢献である。

また、先行研究ではロボットを使ったプログラミング実践が多いことも確認した。一方で、スクリーン上でのみの、未就学児のプログラミング実践・理解についての研究は少ないことを確認した。本研究を通して、未就学児は具体的に表現力の高いプログラミング言語を使うことで、プログラミングを理解し、表現することができることがわかった。また、その際に、iPad などの 1 つのタブレット・機材で、画面上の絵の見立てを変えることにより、様々な未就学児の共感が呼び起こせることがわかった。そして、様々な共感を伴った CTP を繰り返し体験することで、プログラミング自体が未就学児の手になじみ、表現のツールになることを確認することができた。これは物理的な制約を不可避とするロボットを使った学びでは実現し難いものだと考えられる。ロボットを用いない未就学児のプログラミング教育の、ロボットに対する利点を明らかにしたものだと考えられる。ソフトウェアを使った、スクリーン上での未就学児のプログラミング教育の利点を明らかにしたことが、本研究の第二の理論的な貢献である。

8.3 本研究の制限と課題

本研究では未就学児の発達段階と学びを考慮したカリキュラムを作成し、それを実践した。未就学児のプログラミングの理解と表現に焦点を当て、未就学児の

プログラミング学習のデザインとその実践による結果をもとに、未就学児の理解と表現を明らかにしたと言える。一方で、それぞれの園児がプログラミングの過程でどのような発達のプロセスを経たかについては明らかにできてはいない。図 8.1, 図 8.2 における, CTP の繰り返しの部分が, どのような繰り返しなのかを詳細に見る必要がある。佐伯は「発達」というものは「価値中立的」なものではなく, その団体, コミュニティ, グループに属する文化的価値づけの中にある, という。つまり, 未就学児の発達のプロセスにおいて, その子どもの発達は, 子どもが帰属する集団の潜在的文化的価値に必ず影響される。そして, それを抜きにして, 意図的に特定の価値観を押し付けることがあってはならない [83]。本研究においては, 園児の全体のプログラムから量的な理解と, 最終的にどのような表現をするのかは詳しく見てきた。一方で, 園児たちが帰属するグループの中で, そのグループの潜在的文化的価値を伴って起きているプロセスについては見ていない。

未就学児はプログラミングという新しい道具に接したときに, 大人が思いも寄らない未知なる未来の文化的営みを発見するはずである。大人がすでに築き上げてしまっている固定観念にとらわれない意味づけをしている過程・様子を詳細に見る必要がある。具体的には, レッスン内でのプログラムを完成させるまでの試行錯誤の過程や, ビスケットランドを使った自由課題における園児同士のコミュニケーションの影響などが, 詳細に研究されるべきである。本論文第 7 章の園児の姿は, 授業者である幼稚園教諭や, 著者の想像を超えるものであり, 未就学児がプログラミングに「なじんだ」ときに, どのような情報社会への参加を見せるかを垣間見せている。園児がどのようにプログラミングができるようになったのか, プログラミングがどのように未就学児の発達・思考に影響するかは今後さらに研究される必要がある。

関連して, 当初実践を行う前は以下のことは予測できなかったことであった。

1. 園児が描いた絵が何を示しているのかが判別できない
2. 最終プログラムで, 園児が遊んでいる場合がある
3. 最終プログラムだけだと判断ができない

1. に関しては, 当初, 初回のレッスンから園児のプログラムと表現を突き合わせていく計画を立てていたが, 最初の方のレッスンでは園児の絵の中で, 何の絵かわからない絵が大半であったため断念した。また, 2., 3. に関しては本研究の研究手法である, プログラムから園児の理解を測定することに関わることである。2. に関しては, 第 4 章, 第 6 章で見られたが, 練習課題中に時間が余り, できていたにも関わらずそのプログラムを壊したり, 絵に逆の動きをつけることで遊んでいる園児が見られた。上で述べた通り, それらの行為についても, 単純に理解が見られなかったのではなく, 園児らの「いたずら」的な表現の一部であった可能性も考えられた。園児がどのようにプログラミングをしているか詳細に観察する必要がある。3. に関しては, 第 5 章, 第 7 章で見られたが, 制作過程を見なければ園児の意図が推し量ることができないプログラムがあった。プログラムだけか

らの判断は、客観的な測定が可能であり、著者一人での分析が可能であったため、最初の一步としては最善だったと考えられるが、今後は参与観察も考慮した実験が必要だと考えられる。

学習を促すカリキュラムも、実施しての課題がわかった。

1. ランダムと繰り返しの教え方
2. 難しいプログラムも教えられるか
3. 年少児，年中児のカリキュラムの開発

1. に関しては第5章，第6章で見られた課題である。教え方を工夫することで改善される可能性が考えられる。2. に関しては，今回は先行研究からも条件分岐は入れなかった。全員が理解することを目指す上で，インタラクションの入ったプログラムも入れなかった。一方で，全員が理解する条件分岐の教え方，インタラクションの教え方は考える余地はあると考えられる。また，早期に子どもに情報社会に対しての文化的実践を促すという意味では，年少児，年中児のカリキュラムの開発も課題として挙げられる。

8.4 結言

本論文の最後に，未就学児を対象にしたプログラミング教育に提言を行う。

未就学児を対象にしたプログラミング教育において，未就学児の発達段階に適した教育内容であるべきである。その際，使用する言語は具体的なもので，しかも，表現力の幅が広いものが望ましい。なぜなら，それによって，未就学児はプログラムをする対象に共感し，そのものがもっている性質に合わせて自分から自ずとプログラミングという行動ができる。

また，カリキュラムにおいては，十分時間をとり，未就学児が自らプログラミングでできることの意味を探求し，児童自体にプログラミングが「なじむ」時間をかけるべきである。佐伯のドーナツ理論に従えば，「なじむ」時間を経ずに，授業者の意味を教えるだけでは，その授業者（社会）のもつ意味に盲目的に従うことになり，児童自身の，意味を勝ち取った上での社会への主体的な参加にならない。本研究の中では，絵を変えて何度も同じ課題を与えることにより，児童に様々な共感を演出し，その共感によって児童自身のプログラムに対する意味づけを引き出したと考えられる。

そして，必ずグループ制作の自由課題を取り込むことが望ましい。自分だけでなく，他者の表現も見ることによって，多元的な意味づけの体験を担保するべきである。これによって，児童が自分なりに表現をする下地を養うことができる。

最後に，未就学児にプログラミングで表現をさせることが重要である。第7章においてみられた様々な児童なりの表現があった。このように，未就学児にとってのプログラミングの意味は，大人の価値観の教え込みで与えられるものではな

い。「なじむ」体験を経て、未就学児は表現することによって、自分にとってプログラミングとはどういうものなのか考え、文化的実践としてのプログラミングができるようになる。また、文化的実践としての未就学児のプログラミングは、情報社会に対する価値観が固まった保育者、教育者、大人たちにも影響を与えるものになる。未就学児の情報社会への参加を受け入れることが、より豊かな情報社会を築くことにつながると考えられる。そのような、発達の視点をもった未就学児のプログラミング教育についての研究が今後もなされるべきである。

本研究において、未就学児のプログラミング教育の有用性と価値を全て明らかにできたわけではない。特にプログラミングの存在が未就学児の思考・発達にどのような影響を及ぼすかについては、発達研究として詳細に見ていく必要がある。また、本研究における実践内においても課題がさらに明らかになった。

本研究では、プログラミング言語に Viscuit を利用した。Viscuit は具体的で、にも関わらず表現の幅が広い言語であった。一方で Viscuit を用いないでも、他の言語で未就学児がプログラミングを理解し、表現できるのかも明らかにされるべきである。表現力が高い、具体的なプログラミング言語であれば、未就学児でも理解し、表現できると考えられる。今後の研究で未就学児のプログラミング教育について、さらにこういった教育が必要なのか、可能なのか、そしてその有用性を明らかにしていきたい。

謝辞

本研究を行うにあたり、多くの方の御協力と御指導を賜りました。御世話になったすべての方に心より感謝します。最初に本論文を審査していただき、ご指導を賜りました論文審査委員の本学大学院情報理工学研究科情報・ネットワーク工学専攻の中山泰一教授、岩崎英哉教授、小林聡教授、寺田実准教授、大阪電気通信大学工学部電子機械工学科の兼宗進教授に深く感謝します。また、博士後期課程在学中の3年間の主任指導教官をしていただいた、久野靖教授にも深く感謝します。

中山泰一先生には、私の博士後期課程在学中の主任指導教官、それに論文審査委員会の主査をお引き受けいただきました。博士論文を作成するにあたり、技術的なことはもとより、公私にわたり常に温かく見守り、助言を沢山頂きました。久野靖先生には、在学中、常に温かく見守り、また研究に対して適切なアドバイスを頂きました。安心して研究を進めることができました。心より感謝いたします。

兼宗進先生には、本論文についてとても親切なアドバイスをいただきました。また、本学大学院情報理工学研究科情報・ネットワーク工学専攻の赤池英夫助教、本学共通教育部の赤澤紀子特任准教授、そして、本学大学院情報理工学研究科情報・通信工学専攻角田博保元准教授にも、本論文について様々なアドバイスをいただきました。深く感謝します。

合同会社デジタルポケットの原田康德博士には、研究の随所でお力添えをいただきました。また、私の博士後期課程への進学の扉を開いていただきました。公私にわたり温かく見守って頂き、深く感謝いたします。また合同会社デジタルポケットの井上愉可里さん、青柳美紀さんにも多大なご協力をいただきました。深く感謝しております。

プログラミング学習実践にご協力くださった幼稚園の皆さま、そして、関係者の皆さまに感謝いたします。特に中山佑梨子先生には、本研究において中心的に園児の指導にあたってくださいました。鈴木由香里園長先生も、本研究に惜しみない協力をしていただき、まことに感謝しております。

久野研究室、中山研究室のみなさんにも、本論文作成にお力を貸していただきました。特に、渡邊景子先生、当時博士前期課程に在籍していた久保文乃さんは、研究の手伝いをしていただき、沢山のアドバイスやご協力をくださいました。感謝いたします。

再度、大学院に進学した私に、どんな時でも協力してくれた母親に深く感謝いたします。

研究に理解と協力をし，いつも温かく見守ってくれた妻に心より感謝します．

参考文献

- [1] 阿部和広: “わくわくプログラミング,” 日経 BP, ISBN-10: 4822285154, 2013.
- [2] Cypher, A., & Smith, D. C.: “KidSim: end user programming of simulations,” *Proc. the SIGCHI conference on Human factors in computing systems*, pp.27-34, 1995.
- [3] Anderson, M. and Furnas, G.: “Relating Two Image-Based Diagrammatic Reasoning Architectures,” *Diagrammatic Representation and Inference*, Vol.6170, pp.128-143, 2010.
- [4] Anzoàtegui, L.G.C., Pereira M.I.A.R., and Jarrín M.C.S.: “Cubetto for preschoolers: Computer programming code to code,” *Proc.2017 International Symposium on Computers in Education. IEEE*, 2017.
- [5] Bell, B. and Lewis, C.: “ChemTrains: A Language for Creating Behaving Pictures,” *Proc. IEEE Symposium on Visual Languages*, pp.188-195, 1993.
- [6] Bers, M. U., Flannery, L., Kazakoff, E. R., & Sullivan, A. (2014). Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Computers & Education*, 72, 145-157.
- [7] Bers, M. U.: “The TangibleK Robotics program: Applied computational thinking for young children,” *Early Childhood Research & Practice*, Vol.12, No.2, 2010.
- [8] Clements, D. & Nastasi, B.: “Electronic media and early childhood education,” *Early childhood environment rating scale*, pp.251-275, 1993.
- [9] Clements, D. & Sarama, J.: “Young Children and Technology: What Does the Research Say?,” *Young Children*, Vol.58, 2003.
- [10] Code.org: “コンピューターサイエンスを学ぼう。世界を変えよう。|Code.org,” <https://code.org/>(参照 2020-4-1).
- [11] Denning, P. J.: “The profession of IT Beyond computational thinking,” *Communications of the ACM*, Vol.52, No.6, pp.28-30, 2009.

- [12] DevTech Research Group: “DevTech Research Group,” <https://ase.tufts.edu/devtech/index.html> (参照 202-9-1) .
- [13] Flavell, J. H.: “Piaget’s legacy,” *Psychological Science*, Vol.7, No.4, pp.200-203, 1996.
- [14] Flannery, L. and Bers, M.: “Let’s Dance the “Robot Hokey-Pokey!” Children’s Programming Approaches and Achievement throughout Early Cognitive Development,” *JRTE*, Vol.46, No.1, pp.81-101, 2013.
- [15] Flannery, L. P., Silverman, B., Kazakoff, E. R., Bers, M. U., Bontá, P., & Resnick, M.: “Designing ScratchJr: Support for early childhood learning through computer programming,” *Proc. The 12th international conference on interaction design and children*, pp.1-10, 2013.
- [16] Fletcher-Flinn, C. M., & Gravatt, B.: “The efficacy of computer assisted instruction (CAI): A meta-analysis,” *Journal of educational computing research*, Vol.12, No.3, pp.219-241, 1995.
- [17] Furnas, G. W.: “New graphical reasoning models for understanding graphical interfaces,” *Proc. the SIGCHI Conference on Human Factors in Computing Systems*, pp.71-78. 1991.
- [18] 合同会社デジタルポケット: “ビスケット Viscuit | コンピュータは粘土だ !! ,” <https://www.viscuit.com> (参照 2020-3-1) .
- [19] 合同会社デジタルポケット: “ビスケットプログラミングコンテスト 2018 結果発表,” <https://www.viscuit.com/awardannouncement/> (参照 2020-9-1) .
- [20] 合同会社デジタルポケット: “ビスケットファシリテータ講習 - デジタルポケット,” http://www.digitalpocket.org/training_program, (参照 2018-10-1).
- [21] Harada, Y. and Potter, R.: “Fuzzy rewriting: Soft program semantics for children,” *Proc. In IEEE Symposium on Human Centric Computing Languages and Environments*, pp. 39-46). 2003.
- [22] Harada, Y. and Potter, R.: “Fuzzy Rewriting, End User Development,” *Human-Computer Interaction Series*, Vol.9, pp.251-267, 2006.
- [23] 原田康徳: “体験型ワークショップ用ソフトウェアの開発,” 情報処理学会第 50 回プログラミング・シンポジウム報告書, pp.163-168, 2009.

- [24] 原田康德, 渡辺勇士: “バスケットプログラミングワークショップ—なぜワークショップなのか—,” 情報処理, Vol.58, No.10, 891-893, 2017.
- [25] 原田康德, 渡辺勇士, 井上愉可里: “バスケットであそぼう,” 翔泳社, ISBN4798143057, pp.137, 2017.
- [26] 林健造: “異文化としての幼児画—あなたへのメッセージの読みとり方,” フレーベル館, ISBN457781126X, pp.10-12, 1996.
- [27] Heikkilä, M. and Mannila, L.: “Debugging in Programming as a Multimodal Practice in Early Childhood Education Setting,” *Multimodal Technologies and Interaction*, Vol.2, pp.42, 2018.
- [28] 一般社団法人電子情報技術産業協会: “アルゴリズム体験ゲーム・アルゴリズム | JEITA ソフトウェアで未来をつくる,” <https://home.jeita.or.jp/is/highschool/algo/>(参照 2020-4-1).
- [29] Ishigaki, E. H., Chiba, T., & Matsuda, S.: “Young children’s communication and self expression in the technological era,” *Early Child Development and Care*, Vol.119, Issue.1, pp.101-117, 1996.
- [30] 井原成男: “ウィニコットと移行対象の発達心理学,” 福村出版, ISBN-10 : 4571230443, pp64, 2009.
- [31] JJPC: “第4回全国小中学生プログラミング大会,” <http://jjpc.jp/>, (参照 2019-11-1).
- [32] 香川富士見丘幼稚園: “茅ヶ崎市 香川富士見丘幼稚園ホームページ,” <https://kagawa-fujimigaokayochien.rexw.jp/>
- [33] 笠井優, 原田康德, 大島久雄, 高宮由美子: “ヴィジュアル言語 Viscuit を利用した連続ワークショップ,” 日本デザイン学会研究発表大会概要集, Vol.56, pp.62-63, 2009.
- [34] 株式会社アーテック: “4歳からの体験型 STEAM 教室 — First STEAM,” <https://www.artec-kk.co.jp/school/fs/> (参照 2020-9-1) .
- [35] 株式会社しくみデザイン: “Springin’ (スプリングイン) | プログラミング×絵×音で誰でもクリエイター,” <https://www.springin.org/jp/>(参照 2020-4-1).
- [36] 加藤則子: “乳幼児と電子メディア・インターネットをめぐる諸問題” 公衆衛生, vol.84, No.9, pp.578-583. 2020.

- [37] Kanaki, K. and Kalogiannakis, M.: “Introducing fundamental object-oriented programming concepts in preschool education within the context of physical science courses,” *Educ Inform Tech*, Vol.23, pp.2673-2698, 2018.
- [38] 兼宗進: “小中高で必修化されるプログラミング教育とそれに向けた研究の紹介,” 通信ソサイエティマガジン, N0.50, pp.92-99, 2019.
- [39] 兼宗進: “教育利用を目的としたオブジェクト指向言語の研究,” 筑波大学大学院ビジネス科学研究科博士論文, 2003.
- [40] Kay, A: “Squeak etoys, children & learning,” http://www.vpri.org/pdf/rn2005001_learning.pdf, 2005.
- [41] Kazakoff, E.R., Bers, M.: “Programming in a Robotics Context in the Kindergarten Classroom: The Impact on Sequencing Skills,” *JEMH*, Vol.21, No.4, pp.371-391, 2012.
- [42] 経済産業省: “未来の教室～learning innovation～,” <https://www.learning-innovation.go.jp/> (参照 2020-3-1) .
- [43] 経済産業省: “EdTech 導入補助金 令和元年度補正” <https://edtech-hojo.jp/> (参照 2020-3-1) .
- [44] Kindborg, M., & McGee, K. (2007). Visual programming with analogical representations: Inspirations from a semiotic analysis of comics. *Journal of Visual Languages & Computing*, 18(2), 99-125.
- [45] Klassner, F.: “A case study of LEGO Mindstorms’ suitability for artificial intelligence and robotics courses at the college level,” *Proc. The 33rd SIGCSE technical symposium on Computer science education*, pp.8-12, 2002.
- [46] Knudsen, J. B., & Loukides, M.: “The unofficial guide to Lego Mindstorms robots (pp. I-XV),” O’Reilly, ISBN-10: 1565926927, 1999.
- [47] 教育新聞: “「情報」を大学入試科目に 共通テストへの導入も提言,” https://www.kyobun.co.jp/news/20180314_05/ (参照 2020-9-1) .
- [48] 国立教育政策研究所: “資質・能力を育成する教育課程の在り方に関する研究報告書 4 : ICT リテラシーと資質・能力,” pp.51-69, 2017.
- [49] 久野靖: “プログラミング教育／学習の理念・特質・目標,” *情報処理*, Vol.57, No.4, pp.340-343, 2016.

- [50] Leidl, K. D., Bers, M. U., and Mihm, C.: “Programming with ScratchJr: a review of the first year of user analytics,” Proc. *International Conference on Computational Thinking Education*, pp. 116-121. 2017.
- [51] Mannila, L., Dagiene, V., Demo, B., Grgurina, N., Mirolo, C., Rolandsson, L., & Settle, A.: “Computational thinking in K-9 education,” Proc. *The working group reports of the 2014 on innovation & technology in computer science education conference*, pp.1-29, 2014.
- [52] Martinez, C., Gomez, M. J., & Benotti, L.: “A comparison of preschool and elementary school children learning computer science concepts through a multilanguage robot programming platform,” Proc. *The 2015 ACM Conference on Innovation and Technology in Computer Science Education*, pp.159-164, 2015.
- [53] McDaniel, R. G., & Myers, B. A.: “Getting more out of programming-by-demonstration,” Proc. *the SIGCHI conference on Human Factors in Computing Systems*, pp.442-449. 1999.
- [54] 文部科学省: “情報教育の参照基準,” https://www.mext.go.jp/component/a_menu/education/detail/__icsFiles/afieldfile/2019/03/13/1413650_008_1.pdf (参照 2020-3-1) .
- [55] 文部科学省: “情報教育に関連する資料” https://www.mext.go.jp/b_menu/shingi/chukyo/chukyo3/059/siryo/__icsFiles/afieldfile/2015/11/11/1363276_08_1.pdf(参照 2020-11-10)
- [56] 文部科学省: “GIGA スクール構想の実現について,” https://www.mext.go.jp/a_menu/other/index_00001.htm(参照 2019-10-1).
- [57] 文部科学省: “中学校技術・家庭科（技術分野）内容「D 情報の技術」,” https://www.mext.go.jp/a_menu/shotou/zyouhou/detail/mext_00617.html(参照 2020-11-1).
- [58] 文部科学省: “高等学校情報科（各学科に共通する教科）,” https://www.mext.go.jp/a_menu/shotou/zyouhou/detail/1416746.htm(参照 2020-11-1).
- [59] 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（議論とりまとめ）,” https://www.mext.go.jp/b_menu/shingi/chousa/shotou/122/attach/1372525.htm（参照 2020-3-1） .
- [60] 文部科学省: “小学校学習指導要領,” https://www.mext.go.jp/content/1413522_001.pdf(参照 2018-10-1).

- [61] 文部科学省: “幼稚園教育要領,” http://www.mext.go.jp/a_menu/shotou/new-cs/youryou/you/index.htm(参照 2018-10-1).
- [62] 文部科学省: “幼稚園教育要領解説,” https://www.mext.go.jp/content/1384661_3_3.pdf(参照 2020-10-27).
- [63] Morgado, L., Cruz, M., and Kahn, K.: “ToonTalk in Kindergartens: Field Notes,” Proc. *ICTE2002*, 2003.
- [64] Muller, A. A., & Perlmutter, M.: “Preschool children’s problem-solving interactions at computers and jigsaw puzzles,” *Journal of Applied Developmental Psychology*, Vol.6, Issues.2-3, pp.173-186, 1985.
- [65] 内閣府: “低年齢層の子供の保護者向け普及啓発リーフレット全体版「スマホ時代の子育て」,” https://www8.cao.go.jp/youth/kankyouto/internet_use/h30/leaf/pdf/leaf-print.pdf (参照 2020-3-1) .
- [66] 内閣府: “Society 5.0 - 科学技術政策 - 内閣府” https://www8.cao.go.jp/cstp/society5_0/ (参照 2020-12-1) .
- [67] 日本学術会議情報学委員会情報学教育分科会: “報告 情報教育課程の設計指針 - 初等教育から高等教育まで,” <http://www.scj.go.jp/ja/info/kohyo/pdf/kohyo-24-h200925.pdf> (参照 2020-10-1) .
- [68] 野口聡, 堀田博史: “プログラミング的思考の基礎をつくる保育方法の分析,” 日本教育工学会研究報告集, Vol.18, No.1, pp.1-8, 2018.
- [69] 岡本一郎, 中沢正人: “にゃんきちいっかのだいぼうけん,” 金の星社, ISBN-10 : 4798150746, 2017.
- [70] 大阪電気通信大学兼宗研究室: “ドリトルではじめるプログラミング,” <https://dolittle-es.eplang.jp/index.html> (参照 2020-12-2)
- [71] Papadakis S.P., Kaloglannakis M.K.and Zaranis N.: “Developing fundamental Programming concepts and computational thinking with ScratchJr in preschool education: A case study,” *IJMLO*, Vol.10, No.3, pp.187-202, 2016.
- [72] PRIMO: “プログラミング脳を3歳から プリモトイズ キュベット,” <https://www.primotoys.jp/>(参照 2018-10-1).
- [73] Resnick, M.: “Lifelong Kindergarten: Cultivating Creativity through Projects, Passion, Peers, and Play,” The MIT Press, ISBN-10: 0262037297, 2017.

- [74] Resnick, M.: “Turtles, Termites, and Traffic Jams (Complex Adaptive Systems): Explorations in Massively Parallel Microworlds,” The MIT Press, ISBN-10: 0262680939, 1997.
- [75] Repenning, A.: “Agentsheets: A Tool for Building Domain-Oriented Dynamic, Visual Environments,” Ph.D. Dissertation, University of Colorado at Boulder, 1993.
- [76] Repenning, A. and T. Sumner, “Agentsheets: a medium for creating domain-oriented visual languages,” *Computer. IEEE*, vol.28, no.3, pp.17-25, 1995.
- [77] Repenning, A.: “Moving beyond syntax: Lessons from 20 years of blocks programming in AgentSheets,” *Journal of Visual Languages and Sentient Systems*, Vol.3, No.1, pp.68-89, 2017.
- [78] Repenning, A., Basawapatna, A. R., & Escherle, N. A.: “Principles of computational thinking tools,” *Emerging research, practice, and policy on computational thinking*, pp. 291-305, Springer, Cham, 2017.
- [79] 佐伯胖: “共感 育ち合う保育のなかで,” ミネルヴァ書房, pp.17-20, 2007.
- [80] 佐伯胖: “コンピュータと教育,” 岩波新書, pp.15-16, 1986.
- [81] 佐伯胖: “「子どもがケアする世界」をケアする,” ミネルヴァ書房, pp.15-21, 2017.
- [82] 佐伯胖: “新・コンピュータと教育,” 岩波新書, pp.14-19, 1997.
- [83] 佐伯胖: “幼児教育へのいざない 円熟した保育者になるために,” 東京大学出版会, pp.84-86, 2014.
- [84] 佐伯胖: “わかるということの意味,” 岩波書店, pp.195-196, 1995.
- [85] Schneider, K., & Repenning, A.: “Deceived by ease of use: using paradigmatic applications to build visual design environments,” *Proc. the 1st conference on Designing interactive systems: processes, practices, methods, & techniques*, pp. 177-188, 1995.
- [86] R・ケイス: “ピアジェを超えて 教科教育の基礎と技法,” サイエンス社, ISBN-10 : 478190372X, pp.3-8, 1984.
- [87] Scratch: “Scratch - Imagine, Program, Share,” <https://scratch.mit.edu/>, (参照 2019-10-1).

- [88] スクラッチジュニア: “スクラッチジュニア - ホーム,” <https://www.scratchjr.org/>(参照 2018-10-1).
- [89] Seiter, L., & Foreman, B.: “Modeling the learning progressions of computational thinking of primary grade students,” Proc. *The ninth annual international ACM conference on International computing education research*, pp.59-66, 2013.
- [90] 清水匠, 中川一史: “小学校プログラミング教育におけるプログラミング的思考「条件分岐」の種類の整理,” 日本STEM教育学会第1回年次大会, pp.2-5. 2018.
- [91] Smith, D.C., Cypher, A. and Spohrer, J.C.: “KidSim: programming agents without a programming language,” *Commun. ACM*, Vol.37, No.7, pp.54-67, 1994.
- [92] 総務省: “若年層に対するプログラミング教育の普及推進（平成 28 年度 29 年度）,” https://www.soumu.go.jp/main_sosiki/joho_tsusin/kyouiku_joho-ka/jakunensou.html（参照 2020-3-1）.
- [93] 総務省: “地域 ICT クラブについて,” https://www.soumu.go.jp/main_sosiki/joho_tsusin/kyouiku_joho-ka/IoT_learning.html（参照 2020-3-1）.
- [94] 総務省: “総務省 | 若年層に対するプログラミング教育の普及推進事業 | ビスケットによるプログラミング入門全国のビスケットファシリテータを活用したプログラミング教育普及モデル,” <https://www.soumu.go.jp/programming/011.html>（参照 202-9-1）.
- [95] Stagecast Software, Inc.: “STAGECAST,” <https://web.archive.org/web/20150803093937/http://www.stagecast.com:80/>（参照 2020-9-1）.
- [96] Steffe, L. P., & Wiegel, H. G.: “Cognitive play and mathematical learning in computer microworlds,” *Journal of Research in Childhood Education*, Vol.8, No.2, pp.117-131, 1994.
- [97] Sullivan, A. A., Bers, M. U., & Mihm, C.: “Imagining, playing, and coding with KIBO: using robotics to foster computational thinking in young children,” Proc. *CTE2017*, pp.110, 2017.
- [98] シーモア・パパート: “マインドストームー子供、コンピューター、そして強力なアイデア,” 未来社, ISBN4624400437, 1995.

- [99] Squeakland: “すくすくスクイーク 車の運転,” http://squeakland.jp/sqsqsqueak/drive_a_car.pdf (参照 2020-11-26) .
- [100] 首相官邸: “第 26 回産業競争力会議,” <https://www.kantei.go.jp/jp/singi/keizaisaisei/skkaigi/dai26/siryou.html> (参照 2020-3-1) .
- [101] 利根川裕太, 佐藤智: “先生のための小学校プログラミング教育がよくわかる本,” 翔泳社, ISBN-10 : 4798150746, 2017.
- [102] 上田信行, 中原淳 “プレイフル・ラーニング ワークショップの源流と学びの未来” 三省堂, ISBN-10 : 4385365644, 2012.
- [103] 渡辺勇士, 中山佑梨子, 原田康德, & 久野靖: “プログラミング言語ビスケッ トを学び続けた児童のプログラムの分析,” 情報処理学会第 61 回プログラミング・シンポジウム報告書, 2020.
- [104] ウィキペディア: “しりとる王国 - Wikipedia,” <https://ja.wikipedia.org/wiki/しりとる王国>, (参照 2019-10-1).
- [105] ウィキペディア: “北風と太陽 - Wikipedia,” <https://ja.wikipedia.org/wiki/北風と太陽>, (参照 2020-10-1).
- [106] Wing, J. M.: “Computational thinking,” *Communications of the ACM*, Vol.49, No.3, pp.33-35. 2006.
- [107] Yadav, A., Larimore, R., Rich, K., & Schwarz, C. (2019, March).: “Integrating computational thinking in elementary classrooms: Introducing a toolkit to support teachers,” In Society for Information Technology & Teacher Education International Conference. Association for the Advancement of Computing in Education (AACE). pp.347-350, 2019.
- [108] Yamamoto, K. (1996, February). Visulan: “A visual programming language for self-changing bitmap,” *Proc. International Conference on Visual Information Systems*, pp.88-96, 1996.
- [109] 山崎貞登, 山本利一, 田口浩継ほか: “小・中・高校を一貫した技術・情報教育の教科化に向けた構成内容と学習到達水準表の提案,” 上越教育大学研究紀要, Vol.36, No.2, pp.581-593, 2017.
- [110] 湯澤正通, 前田健一, 杉村伸一郎ほか: “教育・発達心理学（心理学研究の新世紀）,” ミネルヴァ書房, ISBN-10 : 4623061566, 2012.
- [111] 読売新聞オンライン: “[生活調べ隊] 未就学児にプログラミング…幼稚園や教室 必修化受け保護者ら関心,” <https://www.yomiuri.co.jp/kosodate/20190408-0YT1T50299/> (参照 2020-9-1) .

関連論文の印刷公表の方法および時期

第4章 動きのプログラムにおける方向の理解についての分析

- (1) 渡辺勇士, 中山佑梨子, 原田康德, 久野靖: “幼稚園児のビスケットプログラムにおける動きの方向の理解についての分析,” 情報処理学会論文誌教育とコンピュータ, Vol.6, No.1, pp.28-39, 2020.

第5章 繰り返し続けるプログラムの理解についての分析

- (1) 渡辺勇士, 中山佑梨子, 原田康德, 久野靖: “幼稚園児のビスケットプログラムにおける繰り返し続けるプログラムの理解の分析,” 情報処理学会論文誌教育とコンピュータ, Vol.7, No.1, pp.38-49, 2021.

第7章 プログラムを使った表現の分析

- (1) Watanabe, T., Nakayama, Y., Harada, Y., & Kuno, Y.: “Analyzing Viscuit Programs Crafted by Kindergarten Children.” *In Proceedings of the 2020 ACM Conference on International Computing Education Research*, pp.238-247. 2020.