



Title	初等教育におけるディジタルものづくり教育のデザイン
Author(s)	森, 秀樹
Citation	大阪大学, 2018, 博士論文
Version Type	VoR
URL	https://doi.org/10.18910/69668
rights	
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

<https://ir.library.osaka-u.ac.jp/>

The University of Osaka

初等教育におけるデジタルものづくり教育のデザイン

大阪大学大学院人間科学研究科
博士論文

森 秀樹

2018年2月23日

目次

第1章 はじめに	4
研究背景、研究目的および研究方法	
第2章 デジタルものづくり教育	8
2-1.初等教育におけるデジタルものづくり教育	
2-2.初等教育向けデジタルものづくり教育とツールの歴史	
2-3.学習理論からみたデジタルものづくりを通じた学び	
第3章 小学校におけるデジタルものづくり授業のデザイン	22
3-1.児童の興味にあわせたデジタルものづくりを可能とする授業デザイン	
3-2.活動の循環モデルによるデジタルものづくり授業のデザイン	
3-3.デジタルものづくり授業の運営	
第4章 Cricket を活用した授業デザインと評価	26
4-1.プログラマブルコンピュータ“Cricket”とプログラミング環境の日本語化	
4-2.授業デザイン例1：小学4年生向け授業	
4-3.授業デザイン例2：小学3年生向け授業	
4-4.授業デザイン例3：小学6年生向け授業	
第5章 Scratch を活用した授業デザインと評価	74
5-1.プログラミング環境 “Scratch”	
5-2.授業デザイン例1：小学4年生向け授業	
5-3.授業デザイン例2：小学4年、6年生向け理科授業	
第6章 デジタルものづくりツール「プログラマブルバッテリー」の開発	91
6-1.いつでもどこでもデジタルものづくりを目指したプログラマブルバッテリーの開発	
6-2.プログラマブルバッテリーの機能	
6-3.プログラマブルバッテリーを活用した小学校低学年向け授業のデザイン	
6-4.評価	
6-5.考察と今後の課題	
第7章 文系大学生向けデジタルものづくり授業実践との比較	96

- 7-1.文系大学生向けデジタルものづくり授業のデザイン
- 7-2.Cricket を活用したデジタルものづくり授業のデザインと実践
- 7-3.Scratch を活用したデジタルものづくり授業のデザインと実践
- 7-4.小学校におけるデジタルものづくり授業実践との比較

第8章 まとめ

117

- 8-1.初等教育におけるデジタルものづくり授業デザインへの提言
- 8-2.考察～初等教育におけるデジタルものづくり教育の可能性～
- 8-3.結論
- 8-4.デジタルものづくり授業のデザインと運営で気をつけるべき 21 の事柄

参考文献

135

第1章 はじめに

本研究の背景、研究目的と研究方法

本研究は、デジタルツールを活用した新しいものづくりを初等教育に導入するために、小学生が取り組むことができるデジタルものづくり授業をデザインし、授業実践を通じて児童の学びを評価した上で、効果的なデジタルものづくり授業のデザインを提言するものである。

ものづくりが学校教育において重視されるようになったきっかけは、1999年に制定された「ものづくり基盤技術振興基本法」である。同法には、ものづくりに関する教育・学習機会を充実させることの必要性が盛り込まれており、第16条では「国は、青少年をはじめ広く国民があらゆる機会を通じてものづくり基盤技術に対する関心と理解を深めるとともに、ものづくり基盤技術に関する能力を尊重する社会的気運が醸成されるよう、小学校、中学校等における技術に関する教育の充実をはじめとする学校教育及び社会教育におけるものづくり基盤技術に関する学習の振興、ものづくり基盤技術の重要性についての啓発並びにものづくり基盤技術に関する知識の普及に必要な施策を講ずるものとする」と記されている。そして、2008年1月の中央教育審議会の答申「幼稚園、小学校、中学校、高等学校及び特別支援学校の学習指導要領等の改善について」では、教科を横断して重視すべき項目として、「情報教育」「環境教育」「キャリア教育」「食育」「安全教育」「心身の成長についての正しい理解」とともに「ものづくり」を挙げており、2008年公示の学習指導要領に反映されたものの、小学校でのものづくりは生活や図画工作に加えて、理科の一部で取り扱われることに留まった。

現在、中学校では技術・家庭科の技術分野でのものづくりを扱っているが、小学校ではものづくりについて集中的に学ぶ機会がない。他方、諸外国では、英国において初等教育カリキュラムの中に、ものづくりを学ぶ科目として「Design and Technology」が置かれており、2013年に発表された新カリキュラムにはコンピュータとプログラミングを活用したものづくりが含まれている。米国でも2012年に1000カ所の学校へデジタルファブリケーション機器を設置した工作室を設ける計画を立てる(日本経済新聞 2012年11月14日付)など、新しいものづくり教育が世界的に拡がりつつある。

2010年代半ば以降はAIやIoTなど、新しい情報技術による産業発展を目指して、英国、ロシア、オーストラリア、フィンランドなど世界各地で小学校段階でのプログラミング教育の必修化がなされた(太田ら 2016)。このような流れを受けて、我が国でも学習指導要領の改定により2020年度から小学校でのプログラミング教育を必修化することとなり、広い意味でのデジタルものづくりが初等教育に取り入れられることとなった。

現実問題として、子どもたちを取り巻く「もの」について目を向けてみると、家電製品から玩具まで、子どもにとって身近なものが多くにコンピュータが埋め込まれ、モーター・センサーなど様々な技術が活用されている。いわゆるTVゲームのみならず、玩具売上ランキングで毎年上位にあがるTVヒーロー関連玩具をはじめ、多くの玩具が電池で動き、光り、音を鳴らす。現代の子どもは意識することなく、これらの技術に遊びの中で慣れ親しんでいる。しかしこれでは、技術の消費者として

の子どもを積極的に育てているものの、新しい技術を主体的に活用できる子どもを育んでいるとは言い難い。学校教育で取り扱われているものづくりに目を向けると、いわゆるアナログなものづくりがほとんどで、玩具をはじめとして、子どもを取り囲む「もの」のデジタル化が著しい現状を踏まえると、学校教育で子どもが取り組むものづくりとの間に大きな隔たりがあると言わざるを得ない(森ら2012)。

2000年以降、「ものづくり」のあり方も技術の進展とともに発展が見られる。新しいデジタルものづくりとして、Gershenfeld(2005)が提唱した「パーソナル・ファブリケーション(大量生産ではなく、個人が必要なものを自らつくるものづくり)」やそのための場所としてのファブラボが世界各国で拡がっている。従来はほとんどが業務利用に限られていた埋め込み型コンピュータや3Dプリンター、レーザーカッターなどのデジタルものづくり機器が、低価格かつ高度な専門知識がなくても利用可能なものとなり、ホビーストを中心に一般にその利用が拡がり、学校教育をはじめ教育現場での利用も可能となった。

そこで本研究では、コンピュータなどのツールやプログラミングなどの手段を活用した新しいデジタルものづくり教育が小学校で取り組まれる将来を想定して、小学校で実施することができるデジタルものづくり教育をデザインした。授業実践を通して、デジタルものづくり授業のデザイン方法と授業を通じた児童の学びとの関係を明らかにし、初等教育における新しい学びの手段としてデジタルものづくり教育と授業デザインを提言することを目的とした。研究の方法論として、実際に授業をデザインし、授業実践を通じて、児童の学びについて評価する教育実践研究の方法をとった。本研究は2002年から2017年までの16年間をかけて実践した奈良女子大学附属小学校、京都教育大学附属京都小学校、立命館小学校、東京都区立A小学校での1クラスあたり10時間から30時間を超える長期間に渡る授業と2時間から4時間で実施した大阪府、京都都、奈良県、東京都、神奈川県の公立小学校での授業の延べ700時間超の授業実践を基にしたものである。いずれの授業も筆者が各学校の教諭と協議のうえ、授業デザインを行い、小学校教諭と筆者のチームティーチングとして授業実践を行った。その上で授業後に、児童の作品や児童が作品づくりを通じた振り返りとしてまとめたテキストの分析、児童への事前事後の質問紙調査から、児童の学びについて評価を行っている。

本論文は8章で構成している。本章、第1章では研究背景、研究目的、研究方法について述べた。第2章ではデジタルものづくり教育とツールについての現状と歴史的背景、および学習理論からデジタルものづくり教育について論じる。続く第3章では小学校でのデジタルものづくり授業のデザイン方法について論じ、第4章から第6章では上述の小学校での実践と児童の学びの評価について論ずる。第7章は大学生を対象としたデジタルものづくり授業実践との比較を行い、第8章で初等教育におけるデジタルものづくり教育の意義とそれを取り入れるための授業デザイン方法について提言を行う。

なお、本研究におけるデジタルものづくりは「コンピュータなどのデジタルメディアやプログラミングなどの手段を用いて行う新しいものづくり」として、いわゆる物理的なものをつくるものづくりだけではなく、画面上で動くものづくりも含めて広い意味でデジタルものづくりとした。

第2章 ディジタルものづくり教育

2-1.初等教育におけるデジタルものづくり教育

本節では初等教育におけるデジタルものづくり教育の歴史について振り返る。デジタルものづくりのなかでも、最も長い歴史を持つコンピュータとプログラミングを活用した教育を中心に述べていきたい。

2-1-1.デジタルものづくり教育の草創期～1980年代におけるプログラミング教育～

日本では2020年度から必修化されるプログラミングが、学習指導要領に明記される最初の小学校での広義でのデジタルものづくりとなる。

しかし1980年代のコンピュータの小学校への導入期に一部の学校や研究指定校で、後に紹介するプログラミング言語「Logo」を用いた教育実践が行われている。日本における小学校での最初のLogo活用による教育実践は、1982年に自作のLogoを用いて富山県の小学校分校で行われた戸塚滝登による実践とされる(戸塚1995)。学習障害を持つ5年生の児童ともに行った同実践では、21時間の学習を通じて、Logoでのプログラミングによる図形描写により二等辺三角形や台形、平行四辺形についての図形の性質を児童が理解できるようになった。戸塚は1984年には、4年生を対象に国語、算数、図画工作の授業内で、教科を横断して10時間のLogoを用いた授業実践を行っている。また教科外のゆとりの時間を活用して、5年生を対象に行われた鈴木勢津子による実践では、児童が3人1組となり、コンピュータの操作に慣れることからはじめ、Logoの使い方を図形の作成などを通じて覚え、さらに下級生にLogoの使い方を伝えるためのテキストやカルタづくりへと学習を展開した(鈴木1989)。土橋永一はLogoを用いて児童の国語と算数学習のためのソフトウェアを開発し、自らの授業実践に用いた(土橋ほか1984)。このように教師がLogoを用いて開発した教材を活用した授業実践もコンピュータの学校導入初期には多く見られた。

米国でも同様に学校へのコンピュータ導入とあわせて、プログラミング教育が実施された。

Harel(1986, 1991)は小学4～5年生を対象に米国ボストンの公立校でLogoを用いた授業実践を行った。Instructional Software Design Projectと名付けられた同実践では、児童が下級生向けの学習ソフトウェアの作成を通じて、学習そのものについて学ぶことを目指した。また1980年代にはLogoの教育効果についても様々な議論が行われた(Solomon 1986)。当時、Logoでのプログラミングはプランニングや問題解決スキルの習得に役立つと考えられていた。しかしPea(1983)はLogoでプログラミングを学習した児童と、未学習の児童に対して、「教室をきれいにする方法」についての計画をさせたところ両者に差がないことが明らかにした。言い換えれば、Logoでの学習が児童の他の活動や学習に役立てられていない可能性を示したことになる。

このように1980年代にコンピュータによる新しい教育の可能性を求めて、多くの教師や研究者がプログラミング教育に取り組んだが、普及をするには一般的な教師には教材として取り扱いにくく、発展的な使い方までにたどり着くことが難しかったこともあり、日本、米国ともに、コンピュータの導入

期以降は小学校での Logo を活用した取り組みは激減した。

2-1-2. 2010 年代におけるプログラミング教育の再考

2010 年代に入り、新しい産業基盤としての AI や IoT の立ち上げを目指し、新しく身につけるべきスキルとしてのプログラミングの流行を背景に、世界各地で初等教育でのプログラミング教育の導入が進んでいる。ここでは 2013 年に発表された新ナショナルカリキュラムで初等教育でのプログラミングを必修化した英国の事例を紹介する。英国ではナショナルカリキュラムの改定に伴い、それまでコンピュータの活用を中心であった教科「ICT」を廃止し、新たに新教科「Computing」を導入した。Computing はコンピュータサイエンスを核に教科を構成しており、日本の小学校低学年(1-2 年)にあたる Key Stage 1 より、プログラミングやデバッグ、アルゴリズムの理解に関わる学習が行われる。さらに初等教育段階から、ものづくりを学ぶ教科「Design and Technology」が設置されていることも英国ナショナルカリキュラムの大きな特徴である。同教科の学習は「Design」「Make」「Evaluate」「Technical Knowledge」に大分されており、ものづくりの過程を意識した内容となっている。小学校中高学年(3-6 年)にあたる Key Stage 2 では、教科 Computing での学習を活かし、プログラム制御する作品づくりが行われる。このように英国ではナショナルカリキュラムレベルで、デジタルものづくり教育が推進されている(Department of Education, UK 2013)。プログラミング教育の必修化は英国にとどまらず、前述のように日本でも 2020 年度より小学校でのプログラミング教育が必修化されるほか、ロシア、オーストラリア、フィンランドなどでも小学校での必修化がなされている(太田ら 2016)。

また学校教育以外でも、日本では子ども向けプログラミング教室が 2010 年代に入り、急速に教室数を増やした。企業の社会貢献活動の一環として、プログラミング教育に関連する活動の拡がりも見られる。2001 年から CSK(現 SCSK)は社会貢献活動「CAMP」の一環として、小学生向けにプログラミング活動を含むデジタルものづくりワークショップを全国で開催している。ソフトバンクグループは「Pepper 社会貢献プログラム」として、2017 年より全国の小中学校へ人型ロボット「Pepper」の貸し出しと Pepper を使ったプログラミング学習教材を提供している。世界的にも子どもが無料でプログラミング学ぶことができる場として、Coder Dojo が 2011 年より世界各地に拡がり、2018 年 2 月現在、世界 85ヶ国で 1600 の道場が展開されている。米国で 2013 年に設立された NPO 「Code.org」は、すべての学校でのプログラミング教育の実現を目指して、プログラミングを 1 時間で体験できるプログラム「Hour of Code」をウェブサイトを通じて、世界に向けて提供している。この活動は Amazon、Facebook、Microsoft、Google などの企業と個人による支援で進められている。このように民間主導による小学生向けプログラミング教育の普及も多く見られるようになった(神谷 2015)。

2-1-3. メイカームープメント

2000 年代半ば以降、安価でプログラミングも容易なマイクロコントローラー「Arduino」の登場や 3D プリンターやレーザーカッターなどのデジタルものづくり機器が誰でも使いやすくなったことで、ホビースト中心に様々なものづくりを楽しむメイカームープメントが拡がった。また Gershenfeld(2005)が提唱したパーソナル・ファブリケーションやそのための場所としてのメイカースペースやファブラボが世界各国で拡がり、デジタルものづくりのための環境も整っていった。スタンフォード大学の Bilkstein(2013)は、メイカースペースやファ布拉ボを初等中等教育の学校内に設置し、運営するプロジェクト「Fab@School」を 2008 年に立ち上げた。米国カリフォルニア州を中心に、世界各地に広がりつつある学校内でのメイカースペースやファ布拉ボに関わる教職員や研究者を中心にコミュニティを立ち上げ、2011 年より定期的にカンファレンス「FabLearn」が開催されている。

2-2.初等教育向けディジタルものづくり教育とツールの歴史

本節では米国マサチューセッツ工科大学(MIT)における研究を中心に初等教育向けディジタルものづくりツール開発の歴史についてまとめる。小学生でも取り扱うことができるディジタルものづくりツール開発の歴史は、1960年代よりMITで研究が進められた教育用プログラミング言語「Logo」にさかのぼる。Papertらによって開発されたLogoは、コンピュータに床の上に置いた亀型ロボット(フロアタートル)をプログラムによって動かすことからはじまった(図2-2-1)。

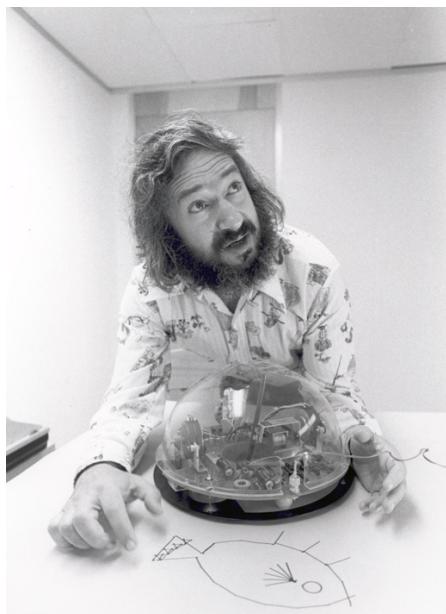


図2-2-1 フロアタートルとPapert (Photo by Matematicamente.it)

やがてフロアタートルは、1970年代末以降パーソナルコンピュータの普及とともに、コンピュータの画面上をプログラムによって動かすスクリーンタートル(図2-2-2)に移行して広まり、1990年代半ばには、米国の約3分の1の小学校にLogoが普及した(Resnick 1994)。Papert(1980)は、プログラミングを「コンピュータが分かる言葉で、コンピュータに考え方を教えること」と表現し、プログラミングを通じて子ども自身が考え方を学ぶことを目指した。また人が異国に住みながら生活のなかで自然にその国の言葉を覚えるように、子どもがLogo環境の中で算数や数学の概念を自然と学ぶことができることを指して、Logoを「Math Land」と表現した。Logoは子どもの思考を参考に設計され、「子どもがなにげなくおしゃべりしながら世界をモデル化したり、抽象化したりするという思考にできるだけ近いかたちのコンピュータ言語の開発を推進させた」(佐伯 1999)といわれている。

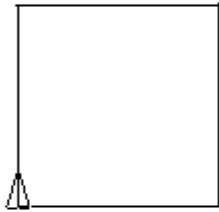


図 2-2-2 スクリーンタートル(Image by Leonardo Carneiro de Araujo)

例えば Logo では画面上のスクリーンタートルを動かすために以下のようなプログラムを作成する。

```
forward 10 (前へ 10 進む)  
right 90 (右へ 90 度まがる)  
forward 10 (前へ 10 進む)
```

先に説明したように 1980 年代から 1990 年代にかけて、コンピュータの小学校導入期には、米国だけでなく日本でも Logo を用いたプログラミング教育が熱心な教諭を中心に盛んに行われていた。しかし、しばらくするとプログラミングは授業で取り扱われなくなった。Resnick et al. (2009) は、この原因について、1)言語そのものの難しさ、2)図形を描くことなど特定の活動に限定されたこと、3)指導者による助言の問題を挙げている。

ソフトウェア上の Logo が各社から開発販売される一方で、1980 年代半ばには、MIT で LEGO® ブロックと Logo プログラミング言語を融合させた LEGO® /Logo が開発された。LEGO®/Logo は、スクリーンタートルのように画面上に留まらず、フロアタートルのように予めできあがつたものをプログラムで動かすのではなく、ブロックとモーター・センサーなどを使い、自分自身で動かすものをつくり、プログラムを作成して制御できることが特徴である。さらに、MIT は LEGO® /Logo の進化版として、コンピュータと有線で繋がっていた LEGO® /Logo から、より製作物に自由度を持たせられるようにコンピュータと繋がらずに、プログラムはコントロールユニットに送信し単独でプログラムを実行することができる Programmable Bricks を開発した(図 2-2-3)。Programmable Bricks は 1998 年に LEGO® 社より LEGO® MINDSTORMS として商品化されている。

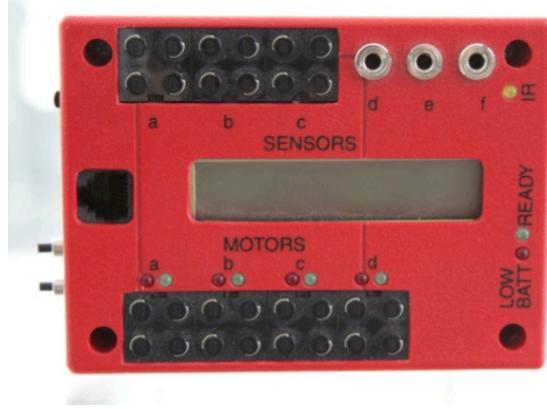


図 2-2-3 Programmable Bricks

また MIT では 1990 年代後半から、ロボットづくりに使用されることが多かった Programmable Bricks を小型で工作材料など様々な素材とも組み合わせることができ、男女を問わず作品づくりを楽しめるようにすることを目的に、Cricket の開発を進めた(図 2-2-4)。あわせて誰でも容易にプログラミングできるように Cricket のプログラミング環境として、ブロック型アイコンでプログラム作成可能なビジュアルプログラミング環境 LogoBlocks を開発した(図 2-2-5)。

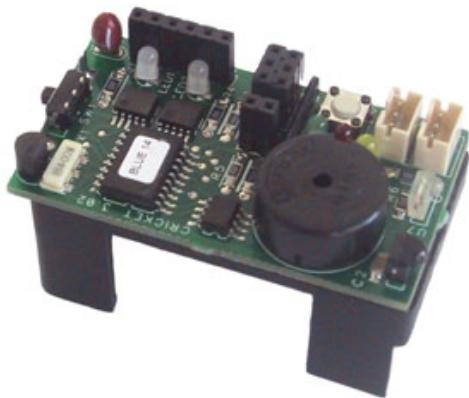


図 2-2-4 Cricket

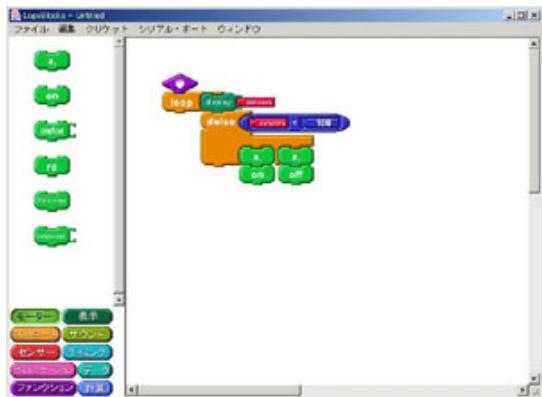


図 2-2-5 LogoBlocks

さらに 2000 年以降は LogoBlocks で用いられたビジュアルプログラミングの方法を活かし、子ども向けのビジュアルプログラミング環境「Scratch」の開発を進めた(図 2-2-6)。ハードウェアの制約がなく、コンピュータさえあれば誰でもシミュレーションやゲーム、インタラクティブなアニメーションなど、画面上で多様な作品づくりが可能な Scratch は、2007 年にインターネットで無償公開された。Scratch の特徴の一つは作品の共有機能である。Scratch でつぐられた作品はインターネット上の Scratch ウェブサイトで作品共有することができる。YouTube で共有された映像を視聴するように共有された作品を動かすことができるだけではなく、作品のプログラムを見ることもでき、さらにプログラムを改良して自分の作品として投稿することも可能である。Scratch は 40 カ国語以上の言語に対応しており、英語でつくられたプログラムを日本語でみるなど、言語を超えた作品共有が可能である。2018 年 2 月現在、Scratch ウェブサイト上に共有されている作品は 2800 万を超えていている。

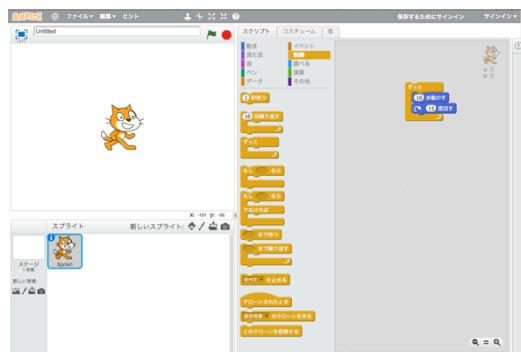


図 2-2-6 Scratch

ここまで、小学生が使用できるデジタルものづくりツールを半世紀にわたり研究してきた MIT が開発したツールについて述べてきた。プログラミング教育が世界的に広まりはじめた 2010 年代半ば以降は、世界各地で様々なツールが開発され、販売、提供されている。図 2-2-7 は英国 BBC が開発した教育用コンピュータ「Micro:bit」である。2016 年には英国すべての 11 歳、12 歳

の子どもに計 100 万個が配布された。Micro:bit のプログラミングはブラウザ上で動作するビジュアルプログラミング(図 2-2-8)や、JavaScript や Python でのテキストプログラミング、さらには Scratch を使ったプログラミングも可能である。Micro:bit は本体に埋め込まれた LED への出力、ボタンスイッチや加速度センサーからの入力、汎用ポートからモーターやセンサーなどの外部デバイスの制御も可能である。日本でも 2020 年までに 30 万台の配布を Micro:bit 財団が計画しており(Microbit 財団 2017)，日本語版のプログラミング環境も提供されている。



図 2-2-7 Micro:bit



図 2-2-8 ビジュアルプログラミング環境

2-3.学習理論からみたデジタルものづくりを通じた学び

Logo から Cricket, Scratch に至る MIT のデジタルものづくりツールの開発背景には、学習理論としてのコンストラクショニズム(Constructionism)がある。ここではコンストラクションズムを中心に本研究に関わる学習理論についてまとめる。

2-3-1.構成主義(Constructivism)

デジタルものづくりを取り入れた授業は、学習者が活動を通じて能動的に学ぶ場である。これは知識は教師から学習者へ一方向的に与えられるものではなく、学習者によって能動的に構成されるとする構成主義(Constructivism)を背景にしている。近年、構成主義は様々な学問分野や領域で研究がすすめられている。学習に関連する分野では、スイスの心理学者 Piaget による構成主義の影響が大きい。生物学に基礎おく Piaget は、主体(学習者)と客体(環境)との相互作用によって、人間は学習し発達していくとしている。Piaget は学習者と環境との相互作用において、学習者の認知構造を環境に押し当てて、環境を変化させて取り込もうとする活動を同化として、今度は反対に環境からの反作用を受けて、学習者の内的な認知構造が変化することを調節と呼んだ。この同化と調節の繰り返しによって、学習者の認知構造が量的あるいは質的に変化することを学習、発達としている(菅井 1993, 2000)。

2-3-2.コンストラクショニズム(Constructionism)

Piaget の共同研究者であった Papert は教育学習理論としてコンストラクショニズム(Constructionism)を提唱した。コンストラクショニズムでは、Piaget の構成主義における認知的な側面からの知識の構成から、ものづくりを介した知識の構成へと学習および教育方法の理論として広げている。つまりコンストラクショニズムは、ものづくりという意味でのコンストラクション(構成)と知識のコンストラクション(構成)の 2 つをかけた理論となっている。Piaget の構成主義と Papert のコンストラクショニズムは両者ともに日本語では同じように「構成主義」と訳されたり、英語表記でも Constructivism と Constructionism と類似しているため、同じように使用されることがあるが、本論文では区別のために Piaget の構成主義と Papert のコンストラクショニズムとして分ける。

コンストラクショニズムについて、Kafai と Resnick(1996)は「子どもたちは知識を得ているのではない。子どもたちは知識をつくりだしている。ロボットや詩、砂の城、コンピュータプログラムなど、自分にとって意味があり、それをもう一度自分自身が見直すことができ、他人と共有できるものづくりに積極的に関わっているときに、特に新しい知識はつくりだされる。」として、広く多様なものづくりをコンストラクショニズムに含めている。また Papert の研究を引き継ぐ Resnick はコンストラクショニズムを背景に、幼稚園で行われているような協同的で表現や活動を通じた学びに学び本来の可能性を見出し、それを一生涯続けるためのツールや学習環境の開発を目指して、MIT メディアラボに

Lifelong Kindergarten(生涯続く幼稚園)研究グループを創設した(Resnick 1998)。いわばテクノロジを活用した現代の Froebel による恩物であり, Montessori による教具であり, 就学以降の子どもから大人までが再び学び本来の姿を取り戻すためのツールと学習環境のあり方の研究でもある。本研究で活用する Cricket と Scratch は, Lifelong Kindergarten グループの研究成果である。

2-3-3.社会的構成主義

一方 Piaget の構成主義に対して, より学習における社会性や文化的側面を意識した社会的構成主義が 1990 年代以降広がった。Vygotsky の見直しから注目されるようになった社会的構成主義は, Piaget の構成主義が環境と学習者の直接的な相互作用であったのに対して, その中に仲間や教師などの媒介者や文化的, シンボリックな道具, 心理学的道具などの媒介が存在する環境による知識の構成を強調している(菅井 2000)。また社会的構成主義では, 知識の社会的な構成に目が向けられ, 発達を文化コミュニティへの参加形態の変容(Rogoff 2003), 状況に埋め込まれた活動のなかでの学習(Lave & Wenger 1991)など, 共同体のなかでの学習, 発達に目が向けられている。次に社会的構成主義の学習観のなかで, 本研究に関連する Vygotsky による最近接発達領域について触れたい。

2-3-4.最近接発達領域

Vygotsky による最近接発達領域は, 子どもが一人で問題解決できる現在の発達水準と, 他人との協同のなかで問題解決する場合に到達する水準との間の差異を指している。つまり一人ではできないが, 他人の助けや道具を使うことによって達成できる領域を示すものであり, 最近接発達領域は子どもの「明日の発達水準」を示すものである。協同的活動のなかでは, 周囲の子どもたちのやり方や考え方を見て学び, 模倣することでできないこともできるようになり, 子どもたちは自分一人ですることよりも多くのことを成し遂げることができる(柴田 2006)。この最近接発達領域に関して, Brown(1994)は理想的な学びの共同体としての教室について, 複合的な最近接発達領域(Multiple Zones of Proximal Development)という言葉を使い表現している。教室における共同体を構成する多様な児童, 教師が, 多様な活動を通じて, 複合的な最近接発達領域をつくりだし, 多様な学びを生み出す可能性を指摘している。

以上の学習理論に加え, 21 世紀の新しい思考方法として注目されている Computational Thinking と Creative Thinking(創造的思考)を本研究に関連するものとして挙げたい。

2-3-5.Computational Thinking

Wing(2006)は Computational Thinking をすべての子どもにとって, 読み書き計算に加えて必要な能力として提案している。Computational Thinking はコンピュータ科学者が考えるように考えること

を身につけること、すなわち抽象化や単純化などコンピュータが計算処理できるレベルで問題を捉えることで問題解決につながるように、論理的に思考することである。英国をはじめプログラミング教育を導入している多くの国では、児童の Computational Thinking スキル獲得を目的としてガイドラインや指標などを示している。日本でもプログラミング教育の導入にあたり、児童がプログラミングを通じて獲得するべき能力をプログラミング的思考として、「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」と定義した(文部科学省 2016)。いざれも広い意味で、コンピュータが考えられるレベルにまで、問題とその解決方法を明らかにしていく論理的思考を指している。

また Papert(1980)の著書「Mindstorms」でも Computational Thinking という言葉が使われており、Computational Thinking という着想は、Logo など初期の子ども向けプログラミング言語による 1980 年代のプログラミング教育や、さらには 1950 年代のコンピュータ草創期にも遡るという見解もある(Denning 2017)。

2-3-6.Creative Thinking と Creative Learning

変化の激しい 21 世紀には、既存の問題解決だけではなく、新たな発想を生み出す Creative Thinking(創造的思考)の重要性も唱えられている。Creative Learning(創造的な学び)は創造的思考を育む手段である。Resnick(2014)は Creative Learning を生み出すための 4 つの要素として Project, Peers, Passion, Play の 4 つの P を挙げている。以下では、Resnick (2014, 2017)による 4 つの P の定義を紹介する(表 2-3-1)。

表 2-3-1 4つのP

Project
人々は各々にとって意味のあるプロジェクトに積極的に関わっている際に、新しいアイデアを生み出したり、プロトタイプをデザインしたり、繰り返しそれらを洗練させたりしながら、最も学ぶことができる。
Peers
学びは社会的な活動として、人々がアイデアを共有し、プロジェクトで協働し、他の人の作品を活かして新しい作品をつくりながら、育っていく。
Passion
人々は自分が興味のあるプロジェクトに関わっている際に、より長く、より熱心に、挑戦と向き合いながら活動し、その過程を通じてより学ぶことができる。
Play
学びには新しいことに挑戦したり、いろいろなものをいじってみたり、どこに境界があるのか試してみたり、リスクをとってみたり(失敗することを恐れない)、これらを何度も繰り返すなど、プレイフルな(遊び心を持った)試みが含まれる。

以上が、直接的あるいは間接的に関連する学習理論である。これらを背景として、本研究ではデジタルものづくり授業のデザインと実践を行っている。

第3章 小学校におけるデジタルものづくり授業のデザイン

本章では、本研究でのデジタルものづくり授業デザインの前提とした児童個々の興味にあわせたものづくり活動と、活動の循環、授業運営に必須となる教師の立ち位置としてのファシリテーター的関わりについて述べる。

3-1.児童の興味にあわせたデジタルものづくりを可能とする授業デザイン

「Personally Meaningful Artifact (Kafai & Resnick 1996)」はコンストラクショニズムを支える重要な概念である。どのようなものづくりを通じても学びは生じるが、多くの学びが生じるためには児童にとって「個人的に意味を持つことができるもの」づくりに取り組むことが必要である。先に述べたように1980年代の初期のプログラミング教育が普及しなかった原因の一つは、図形を描くなど児童の活動を限定したことにある(Resnick et al. 2009)。本研究では、児童が興味にあわせて何をつくるかを決定できるように授業をデザインしている。ただしデジタルものづくりではプログラミングを含み、取り扱う機器の使い方をものづくりにあたって学習する必要がある。そこで本研究での授業は、プログラミングの練習などを通じて、デジタルものづくり機器の使い方に慣れることからはじめ、何をつくることができるかを児童が想像できるようにした上で、児童の興味にあわせたものづくり活動へ移行する授業の流れにしている。

3-2.活動の循環モデルによるデジタルものづくり授業のデザイン

上田(2005)は、ものづくりや表現などワークショップでの活動を「つくって」「かたって」「ふりかえる」という3つのフェーズを組み合わせてデザインするモデルを提案している。「つくった」作品について「かたる」ことによって、作り手としての自分と作品との関係性を意識することができ、さらに「ふりかえり」によって気づいた学びを他者に語り、言語化することによって、経験を再構成していくことができるとしている。この3つのフェーズを持つことで、つくることに没入するだけではなく、活動を内省する機会を設け、新たな創造につなげることができる。また振り返りは、Dewey(1938)の「自分自身の経験の反省的思考による再構成を通して、より高次な創造的知性が一元論的に生成される」など、歴史的にも活動や経験を通じた学びのなかで重要とされてきた。

Resnick(2007, 2017)はCreative Thinking Spiral, Creative Learning Spiralとして創造的な思考や創造的な学びを生み出すために、Imagine(想像する)-Create(つくる)-Play(遊ぶ)-Share(共有する)-Reflect(振り返る)という活動の循環が重要であるとしている(図3-2-1)。

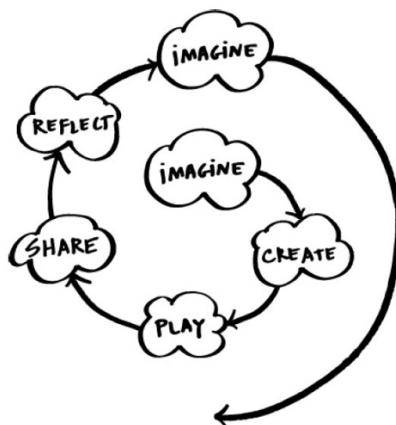


図3-2-1 Creative Thinking Spiral, Creative Learning Spiral [Resnick(2007)より引用]

本研究でのデジタルものづくり授業のデザインでは、1校時45分間という学校授業の時間的制限もあり、時間を区切り、活動を「計画する」-「作業する(つくる、試す)」-「結果をまとめる」-「共有する」-「振り返る」と明確化し、複数回に渡る授業では繰り返しこれらの活動が行われるようにしている(図3-2-2)。

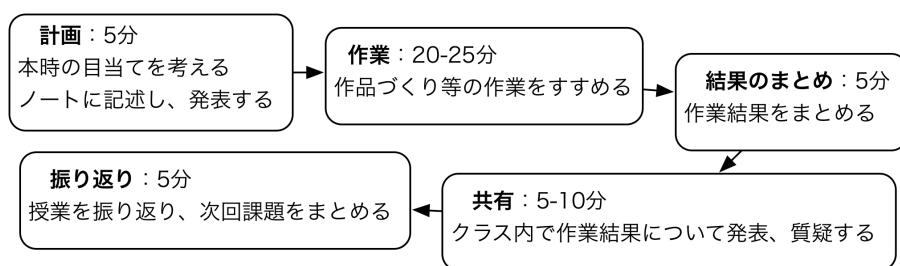


図3-2-2 授業における活動の循環例

3-3. デジタルものづくり授業の運営

デジタルものづくりを取り扱う授業では、教師は単に知識を伝達する教え手としての役割だけではなく、児童それぞれが取り組む多様なものづくりのなかで、児童とともに考え、児童のアイデアを引き出すファシリテーターとしての役割を担う必要がある。2000年以降、子ども向けのワークショップなどの学校外での学びの場でファシリテーターが広がっている。児童がそれぞれの興味にとって多様なデジタルものづくりを行っている教室では、児童の数だけ並行して様々な活動が進められる。本研究で実践した授業は、チームティーチングで運営をすることを前提にデザインしている。Logo を用いたプログラミング教育が定着しなかった原因の一つとして、Resnick ら(2009)は適切なタイミングで適切な助言を与えることができなかったことを挙げている。問題に行き詰まっている児童には解法のヒントを与え、さらに高度な作品づくりに取り組もうとする児童にはその手がかりを適切なタイミングで渡す必要がある。特に、ツールのトラブルには迅速に対応する必要がある。一方で時間をかけて考えている児童の活動を、児童が必要としない助言を与えることで遮らないよう、活動の状況を見守ることも必要である。

デジタルものづくりのためのツールなど、新しい教材を授業に用いる際に、十分な教材研究をする時間をとることができず、導入を思いとどまっている教師も多い。本研究では教材利用に関わる技術的支援を筆者が行ったため、教師の負担は減っているが、仮に教師が十分な教材研究と準備の時間がとれたとしても、多くの教師は授業中に教師でも即座に解決できない問題が発生することを懸念する向きがある。教師は児童とともに考える役割であると教師自身が認識することで、解決が難しい課題について、教師も入ってクラス全体で討議するなどの展開も可能となるはずである。このようにデジタルものづくり授業では、教え手としての教師から児童の学習を支援するファシリテーターとしての教師の役割が求められる。

第4章 Cricketを活用した授業デザイン

4-1. プログラムコンピュータ Cricket(クリケット)とプログラミング環境の日本語化

4-1-1. プログラムコンピュータ Cricket(クリケット)

本章で紹介する授業では、米国マサチューセッツ工科大学(MIT)メディアラボが開発した小型プログラマブルコンピュータ Cricket(クリケット)を用いた。図 4-1-1 は Cricket の製品版として開発された PICOCricket である。図 4-1-2 のように PICOCricket には、モーター、センサー、LED、音源を搭載したスピーカー等を接続することができ、コンピュータ上でプログラムすることで、これらを制御できる。ブロック型のインターフェースを持つプログラミング環境も開発されており、小学生でも容易にプログラミングが可能である(Resnick 2007)。センサーには、光センサー、音センサーのほか、抵抗値を測るセンサー、タッチセンサーが用意されている。本研究では Mori(2011)がプログラミング環境を日本語にローカライズしたバージョンを用いた(図 4-1-3)。



図 4-1-1 PICOCricket



図 4-1-2 PICOCricket とモーター、ライト、スピーカー

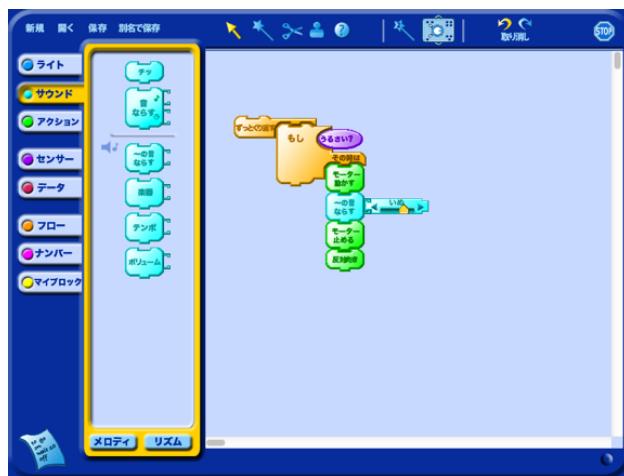


図 4-1-3 PICOCricket 用日本語版プログラミング環境

PICOCricket のプログラミングは、PICOBLOCKS を用いて行う。PICOBLOCKS はブロック型のコマンドを組み合わせることでプログラミングができる LogoBLOCKS の PICOCricket 版である。画面左タブ

より「ライト」「サウンド」「アクション」「センサー」「データ」「フロー」「ナンバー」「マイブロック」に分けられたコマンドのカテゴリを選択し、使用するコマンドブロックを選び、画面右のプログラミングエリアへドラッグする(図 4-1-4)。

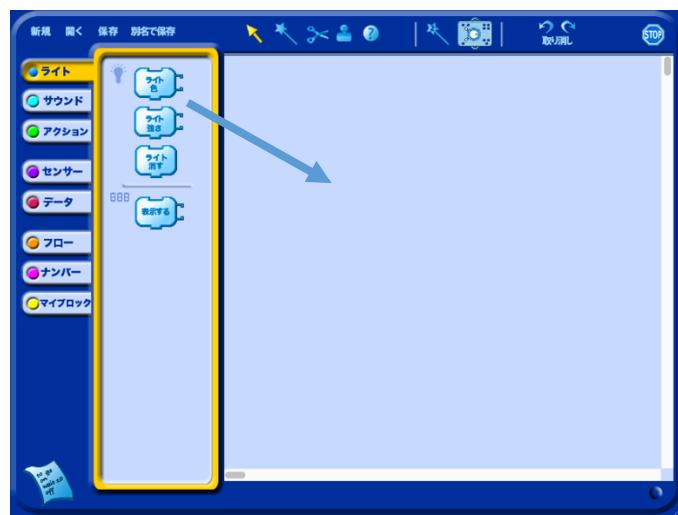


図 4-1-4 PICOBlocks のプログラミングエリア

プログラミングエリアに置かれたプログラムをダブルクリックすると、コンピュータに繋げられた送信ユニットを通じて PICOCricket へプログラムが送信される。一度、PICOCricket 側で受信したプログラムは、コンピュータ無しで PICOCricket 単体でプログラム実行できる。このため、児童が作成するおもちゃや作品などに容易に埋め込むことができる。図 4-1-5 は PICOCricket に接続されたモーターを動かすプログラムである。「モーター動かす」というコマンドブロック一つでモーターを動かすことができる。



図 4-1-5 モーターを動かすプログラム

コマンドブロックを繋げていくことで複雑なプログラムが作成できる。図 4-1-6 はモーターを 1 秒間動かすプログラムである。モーターを動かし始めた後, 10(1 秒)待って, モーターの動きを止める。このようにプログラムはコマンドブロックを上から順番に実行していく。

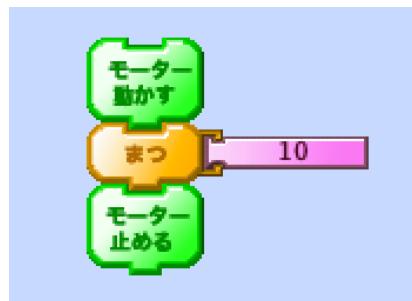


図 4-1-6 モーターを 1 秒間動かすプログラム

モーターの動きを制御する「アクション」ブロックの他にも, LED を制御する「ライト」ブロック, PICO Cricket から音をならす「サウンド」ブロック, 光や音センサーなどをつかうための「センサー」ブロック, PICO Cricket にセンサーからの返り値などを記録, 活用するための「データ」ブロック, 繰り返しや条件分岐を行うための「フロー」ブロック, 演算を行う「ナンバー」ブロック, プログラムの手続化を行う「マイブロック」の 8 カテゴリの 59 コマンドが利用できる。図 4-1-7 は, PICO Cricket に繋げられた LED をピンク色で点灯し, その後モーターが動き始め, 1 秒待って, 子猫の鳴き声をならし, モーターが止まり, ライトが消えるプログラムである。さらに, そのプログラムを繰り返して実行するためには図 4-1-8 のように, 繰り返しブロックを組み合わせることで可能になる(図 4-1-8 の場合は, 10 回繰り返す)。



図 4-1-7 ライト, モーター, サウンドを組み合わせたプログラム

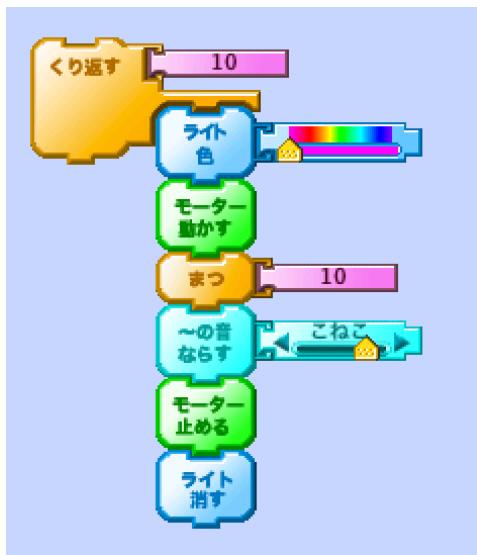


図 4-1-8 繰り返しを行うプログラム

図 4-1-9 のプログラムでは、「フロー」カテゴリの「もし～その時は」ブロックを使って条件分岐を行っている。もし音センサーが音を検知したら(うるさかったら), モーターが動きはじめ, スピーカーから犬の鳴き声をならす。その後モーターを停止させる。また次回モーター始動時の回転方向を逆にするように設定する。この一連の命令をずっと繰り返すプログラムとなっている。

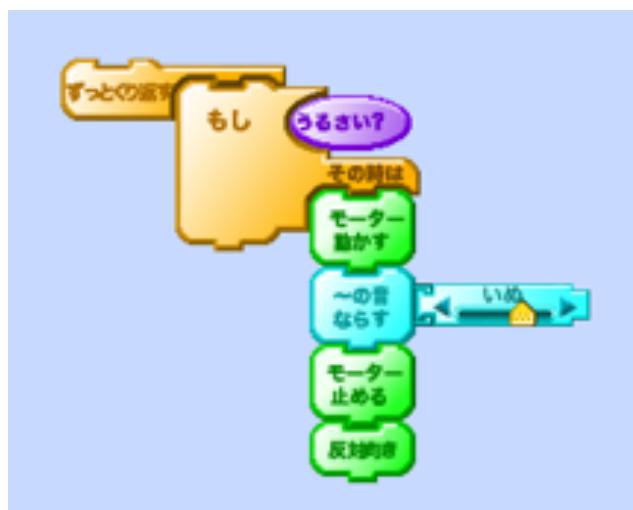


図 4-1-9 条件分岐するプログラム

4-1-2.プログラミング環境の日本語化

PICO Cricket 向けのプログラミング環境はコマンドに英語を用いていたが, 小学生が使用することを踏まえて, プログラミング環境の日本語化を行った。59 のコマンドについて表 4-1-1 のように日

本語の対訳を検討し、プログラミング環境に実装した。なお日本語化にあたっては小学校4年までに学習する漢字を利用することを前提とした。

表 4-1-1 英語コマンドと日本語訳

Category	Original	Japanese Translation	Category	Original	Japanese Translation
Lights	setlight color	ライト 色	Data	clear	消す
	setlight power	ライト 強さ		collect	集める
	light off	ライト 消す		rewind	巻き戻す
	display	表示する		next data	次のデータ
Sound	chirp	チツ	Flow	wait	まつ
	note	音 ならす		waituntil	～までまつ
	play sound	～の音ならす		forever	ずっとくり返す
	set inst	楽器		repeat	くり返す
	set tempo	テンポ		if then	もし その時は
	set volume	ボリューム		if then else	もし その時は その他は
	melody1	メ1		stopall	すべて止める
	rhythm1	リズム1		stop stack	スタック止める
Action	motor onfor	モーター～間動かす	Numbers	number	数字
	motor on	モーター動かす		+	+
	motor off	モーター止める		-	-
	reverse	反対向き		X	X
	this way	こっち		/	/
	that way	あっち		random(min, max)	乱数(最小 最大)
	set power	モーター強さ		>	>
Sensors	touch?	さわった?		<	<
	dark?	暗い?		=	=
	brightness	明るさ		and	かつ
	loud?	うるさい?		or	または
	loudness	音の大きさ		not	～でない
	connected?	つながった?	My Blocks	store in box1	覚える はこ1
	resistance	抵抗		box1	はこ1
	beamIR	赤外線 送る		store in box2	覚える はこ2
	IR	赤外線		box2	はこ2
	newIR?	新しい赤外線?			
	timer	タイマー			
	reset timer	タイマー 戻す			

英語と日本語との文法の違いから、どうしてもブロックでの表示と文法上の順序が異なってしまうコマンドについては、「～」を用いて対応した(図 4-1-10)。



図 4-1-10 英語と日本語コマンドブロック

以降、本論文では注釈がない限り、Cricket と記述しているものは PICOCricket を指している。

4-2.授業デザイン例1： 小学4年生向け授業

本節では小学4年生を対象とした Cricket を用いたデジタルものづくり授業のデザインと実践結果について報告する。本授業では「既存のおもちゃに新しい命を与える」ことをテーマに、電池で動く電車やぬいぐるみ等の既存玩具に Cricket とモーター・センサーを取りつけ、児童が新しいデジタルおもちゃの開発を行なった。また Cricket に接続用に用意されたモーター以外にも電源供給するため、モーター接続ケーブルをカスタマイズして、Cricket 用モーター以外の電動玩具にも電源供給がしやすいうようにした(図 4-2-1)。



図 4-2-1 接続ケーブル

4-2-1.授業デザイン

授業は一通りのコンピュータ操作に慣れている4年生を対象とし、「研究活動」をメタファーに大きく4つのフェーズで構成した。まず①授業で用いる製作ツールとしての Cricket を知る、②ツールに慣れるために「Cricket 研究」として各自で機能を試し、その後の研究課題を考える。その後、③グループもしくは個人で、今まで親しんできた玩具に新しい命を与える(新しい玩具を生み出す)ことをテーマに「おもちゃ研究」を行い、最後に④研究成果をまとめて、ポスター発表を行った(図 4-2-2)。

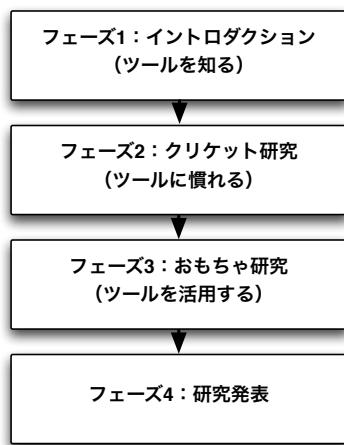


図 4-2-2 授業フェーズ

各時間の授業は大きく、図 4-2-3 のようにデザインした。最初に、①本時の目当てを考え、ノートに記述し発表する。また Cricket の使い方など教師から全体に向けた説明などがあればここで行う。次に、②各自の研究・製作作業を進める。そして、③本時の結果とふりかえりをまとめ、④クラス全員で集まり、本時の結果についての発表と質疑応答をする。またこの際に教師からのアドバイスなども付け加える。

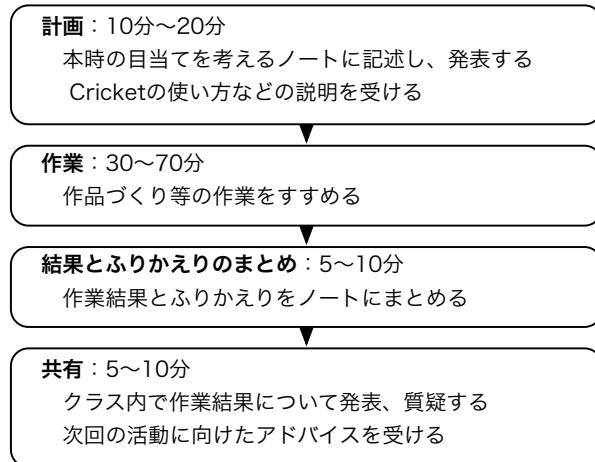


図 4-2-3 各回の流れ

4-2-2.授業実践

授業は奈良女子大学附属小学校 4 年生 1 クラス 39 名を対象に、同校が独自に実施している「情報」と「しごと」学習の時間を利用して、2011 年 11 月より 2012 年 5 月まで、各回 50 分から 100 分の授業(1 校時+休み時間、2 校時+休み時間を利用)として、計 16 回の授業を実施した(表 4-2-1)。

表 4-2-1 授業スケジュール

回	日付	内容	時間
1	2011年11月9日	イントロダクション（クリケットを試す）	50分
2	2011年11月24日	クリケット研究①（繰り返しをするプログラミング）	65分
3	2011年12月1日	クリケット研究②（音センサの制御、条件分岐するプログラミング）	75分
4	2011年12月8日	クリケット研究③（明るさセンサの制御）	85分
5	2011年12月15日	クリケット研究④（コマンドブロックの機能を発見する①）	100分
6	2011年12月22日	クリケット研究⑤（コマンドブロックの機能を発見する②、おもちゃとつなぐ）	95分
7	2012年1月13日	おもちゃ研究①（計画の発表と製作）	70分
8	2012年2月6日	おもちゃ研究②（デジタルおもちゃの製作）	100分
9	2012年2月8日	おもちゃ研究③（デジタルおもちゃの製作）	90分
10	2012年2月9日	おもちゃ研究④（中間発表の資料作成）	90分
11	2012年2月10日	おもちゃ研究⑤（中間発表）	50分
12	2012年2月16日	おもちゃ研究⑥（デジタルおもちゃの製作）	100分
13	2012年3月1日	おもちゃ研究⑦（デジタルおもちゃの製作）	90分
14	2012年3月15日	おもちゃ研究⑧（作品の仕上げ）	100分
15	2012年3月19日	おもちゃ研究⑨（研究レポートのまとめ）	80分
16	2012年5月12日	研究発表会	90分

授業のまとめとして、各自の研究成果をまとめる作業を春休みの課題とし、最後の研究発表会のみを5年次に実施した。教室は40台のPCと電子黒板を備えたコンピュータ教室(図4-2-4)を使い、担任教諭と筆者の2名で運営した。コンピュータ以外のCricketをはじめ、今回使用した機材(表4-2-2)は、授業の開始直前にセッティングし、授業後には授業準備室への片付けを行った。また準備と片付けの作業については、児童のなかで担当係を3~4名決め、その係を中心に児童が主体的に行った。



図 4-2-4 教室

表 4-2-2 使用機材(コンピュータ以外)

Cricket関連
Cricket本体、ビーマー（プログラム赤外線送信機）、シリアルケーブル Cricket用モータ、モータケーブル、モータボード（モータコントローラ） ライト(3色LED)、スピーカー(MIDI音源内蔵)、明るさセンサ、音センサ タッチスイッチ、ディスプレイ（数字表示用）、ケーブル（センサ等接続用） 接続ケーブル（電動玩具への電源供給用）
その他
レゴブロック（ギア、カム、シャフト、タイヤ等を含む） ワニ口クリップ、乾電池、【使用する玩具は各自で用意】

以下は児童の活動毎の様子を図示したものである。全体に向けて説明などを行う際は、図 4-2-5-1 のようにホワイトボードや電子黒板が見えるよう教室前方に児童が集まり、プログラミングを試してみるなど、全体で同じ活動をする際には図 4-2-5-2 のようになり、実際に作品づくりを行う際には、図 4-2-5-3 のように、児童は個々の活動にあつた場所で作業を行なった。

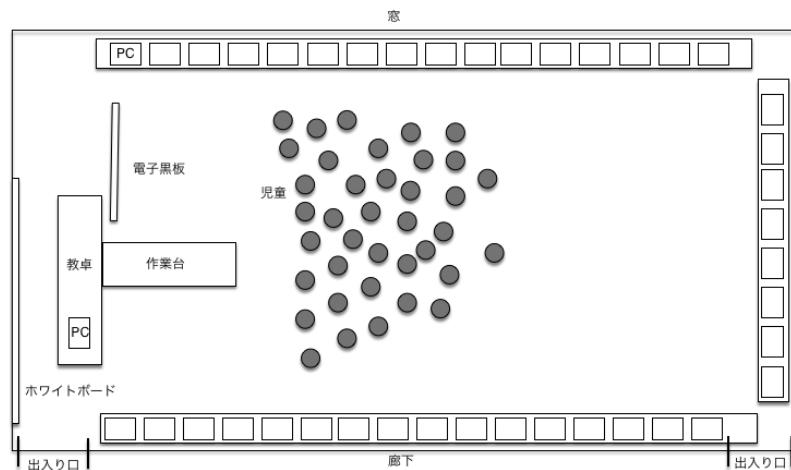


図 4-2-5-1 教室レイアウト(全体説明時)

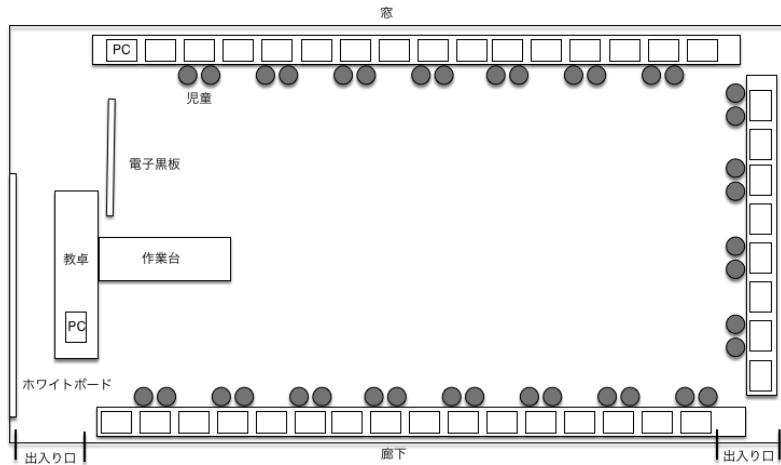


図 4-2-5-2 教室レイアウト(全体作業時)

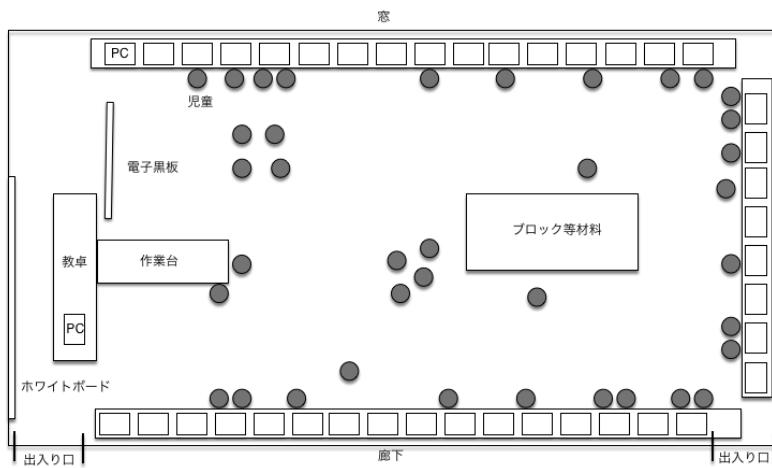


図 4-2-5-3 教室レイアウト(個別作業時)

4-2-2-1. イントロダクション

次に 4 つの授業フェーズについて説明する。まずはイントロダクションとして、第 1 回目の授業を児童が初めて触る Cricket に慣れることを目的に実施した。筆者からサンプル作品として、音に反応して動く作品を見せた後(図 4-2-6), モーター, LED ライト, スピーカーの Cricket への接続方法とそれらを制御する簡単なプログラム(図 4-2-7)の作成とプログラムのダウンロードと実行方法を説明した。その後 2 名 1 組に分かれて、実際に Cricket を試してみた(図 4-2-8, 図 4-2-9)。また本授業について保護者の理解を得るために、第 1 回目を授業参観日に実施し、保護者に児童の学習の活動の様子を見てもらうとともに、授業後に保護者向けの Cricket 体験会を行い、本授業のねらいを説明した。



図 4-2-6 Cricket を使ったサンプル作品の紹介



図 4-2-7 説明したプログラム



図 4-2-8 Cricket を試す①



図 4-2-9 Cricket を試す②

4-2-2-2.Cricket 研究

次に Cricket の機能をより深く理解するために、「Cricket 研究」として 2 名 1 組で活動を行った(図 4-2-10, 図 4-2-11)。計 5 回の授業で構成した本フェーズは Cricket の基本的な使い方の復習からはじめ、繰り返しや条件分岐など徐々にプログラミングの難易度をあげながら、表 4-2-3 のように新しいコマンドブロックの使い方を教師から説明した。その他のコマンドブロックについては、各グループで試しながら発見するよう促した(図 4-2-12)。結果、論理式の使い方や Cricket の入出力ポートの設定方法など、教師が説明していないコマンドブロックの使い方や機能を児童が多数発見した。これらの児童の発見は授業中に発表する機会を設け、クラス内で共有した。また、おもちゃづくりへの導入として、Cricket から専用モーター以外にも電池で動く玩具などへ電源供給ができるなどを紹介した。Cricket とセンサーやモーターを繋ぐケーブル(USB 規格)から、豆電球に直接電源がとれることに気づくなど、様々な発見をする児童の様子がみられた(図 4-2-13)。また係の児童による準備にも工夫が見られ、使用する Cricket 等の機材リストを予め配布するなど、授業が円滑に運営できるように工夫していた。



図 4-2-10 Cricket 研究の様子①



図 4-2-11 Cricket 研究の様子②

表 4-2-3 説明したコマンドブロック

回	日付	コマンドブロック等の説明
1	2011年11月9日	モータへの間動かす、～の音を鳴らす、ライト色、ライト消す
2	2011年11月24日	ずっとくり返す、くり返す、まつ、反対向き
3	2011年12月1日	もし、うるさい?、～までまつ、表示する、音の大きさ、不等式(>, <)
4	2011年12月8日	明るさ
5	2011年12月15日	(ヘルプ機能の使い方)
6	2011年12月22日	さわった?

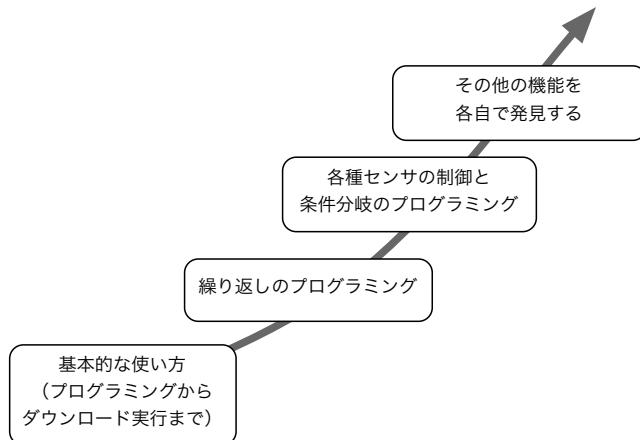


図 4-2-12 コマンドブロックの説明順序



図 4-2-13 豆電球を接続する様子

4-2-2-3.おもちゃ研究

Cricket の機能について、Cricket 研究のフェーズでハードウェアとソフトウェア面での理解を深めた後、既存のおもちゃと Cricket を組み合わせて新しいデジタルおもちゃを開発する「おもちゃ研究」のフェーズへ移った。電池で動く電車おもちゃやラジコンカー、木のおもちゃ、ぬいぐるみなど、Cricket と組み合わせたいおもちゃによって 10 のグループをつくり、9 回にわたって研究をすすめた(図 4-2-14、図 4-2-15)。



図 4-2-14 おもちゃ研究①



図 4-2-15 おもちゃ研究②

赤外線でコントロールするヘリコプター玩具に Cricket を組み合わせることを計画したグループには、新しい機能として Cricket の赤外線送受信機能の説明を行った。多くの児童が Cricket とモーターをどのように既存玩具に組み合わせるか、あるいは Cricket から電動の玩具や理科学習で使用した豆電球や LED にどのように電源供給をするかについて、悩んでいる様子が見られた。Cricket に児童が持っている豆電球や LED をつなぐ場合、Cricket 用のモーターボードから専用の接続ケーブルを用いて電源供給する必要がある。この場合、Cricket 用モーターのコマンドを使って制御する必要があり、児童にも説明をしたが、Cricket 用の LED を制御するコマンドで発光や発色を制御できるのではないかなど、興味を持って試す児童もみられた。また Cricket 研究フェーズで Cricket を構成する IC などのパーツに興味を持った児童が小型版の Cricket をつくることを考えるなど、児童の興味やおもちゃ研究に向けての課題も多岐にわたった。それぞれの児童が直面している課題について、個々に授業内でアドバイスを与えるとともに、全 9 回の授業では、5 回目に中間発表の機会を設けて、それまでの成果についてまとめ、クラス内での発表とアドバイスを交換しあう機会をつくった(図 4-2-16)。中間発表後は、おもちゃ製作の仕上げに向けて、Cricket とおもちゃとの組み合わせ方を再度工夫するとともに、プログラムも工夫していくように促した。最終回は研究レポートのまとめの時間として、①研究のあゆみ(今までの研究過程)、②研究の成果(何が達成できたのか)、③研究の振り返りと今後の課題について、模造紙にまとめをおこなった(図 4-2-17)。授業内で研究レポートのまとめが終わらなかつたため、研究レポートの仕上げを春休みの課題とした。また休み期間中も作品を見ることができるよう、最終回におもちゃ研究の成果である作品の写真、ビデオ、360 度アニメーション画像を撮影し、ウェブサイトにまとめた(図 4-2-18、図 4-2-19)。



図 4-2-16 中間発表



図 4-2-17 研究レポートの作成



図 4-2-18 作品ウェブサイト

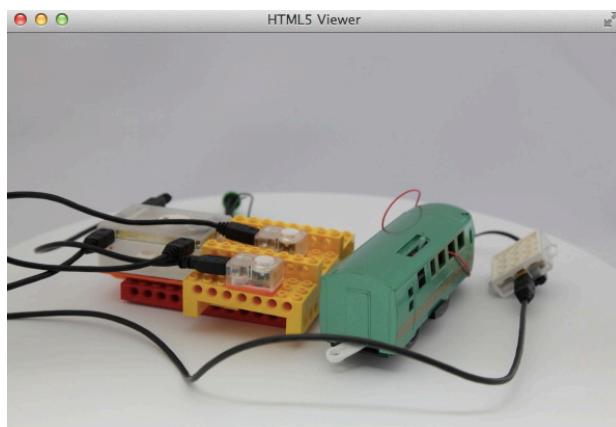


図 4-2-19 360 度アニメーション画面

4-2-2-4.研究発表会

本授業の最終回には保護者を招き、ポスター発表形式の研究発表会を実施した。発表会では熱心に自分たちのグループの研究発表をするとともに、他のグループの発表にも積極的に聞き入り、メモをとる児童の様子が見られた(図 4-2-20, 図 4-2-21)。発表会後の児童の振り返りからは、「今度発表する時にはもっとゆっくり短く切って発表したい」など、発表そのものへの感想や、「自分がさりげなくしていた工夫していたことも自分で分かってきました。」など、発表することからの気づきや、「同じようなテーマで進めている人と話し合って、どのようなところがだめだと考えるかきいてみたいです。」など、次のステップに向けて具体的な進め方を考えているものなどが見られた。さらには、「(ぼくの作品には)『暗い～ライトがつく』『スイッチを押される～止まる(車が)』機能がついていて、その車を見たら、○○さんがライトの機能はしん災の時に使えるのではないかといっていました。しかし、ぼくの頭の中には、別の考えがうかんでいました。ぼくの教室では昼間電気をつけるかつけないかで言い合いをしていました。(中略)ぼくの教室に、もし～暗いとライトをつけるというプログラ

ムをくみこんだら、この問題をかいけてできるのではないかという試みです。」のように、日常生活での課題の解決へとアイデアが拡がっている児童も見られた。



図 4-2-20 ポスター発表の様子①



図 4-2-21 ポスター発表の様子②

4-2-3. ディジタルおもちゃ作品

児童が本授業を通じて製作した作品の例を紹介する。図 4-2-22 は電池で動く電車のおもちゃである。Cricket と 2 つのスイッチを使ってコントローラをつくり、スイッチを押すと走行方向を前後に切り替えられるようにした。図 4-2-23 は LED が光るケーキのおもちゃである。夏休みの課題で製作した電子工作に Cricket を繋げ、Cricket の 4 つのポートに接続された LED がプログラムによつて、別々に点灯する。図 4-2-24 は Cricket に接続されたモーターを使って歩くぬいぐるみである。首につけられた LED も同時に光るようプログラムされている。図 4-2-25 は、だるま落としのおもちゃと Cricket を繋げた作品である。音センサーが反応すると、モーターに接続された木槌が回転し、だるまの下のコマを飛ばす。

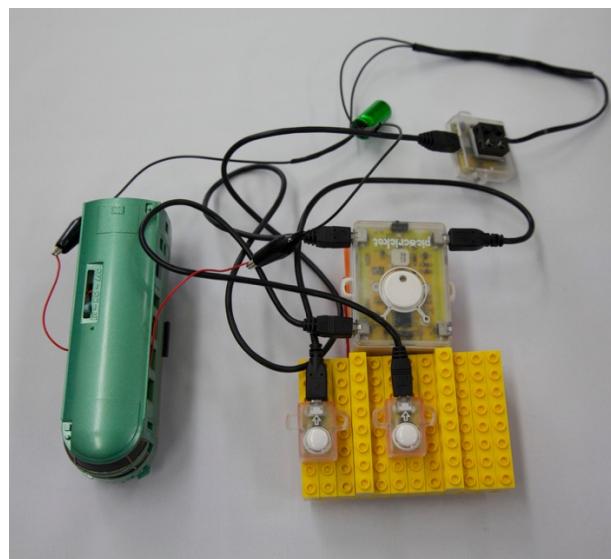


図 4-2-22 電池で動く電車おもちゃ

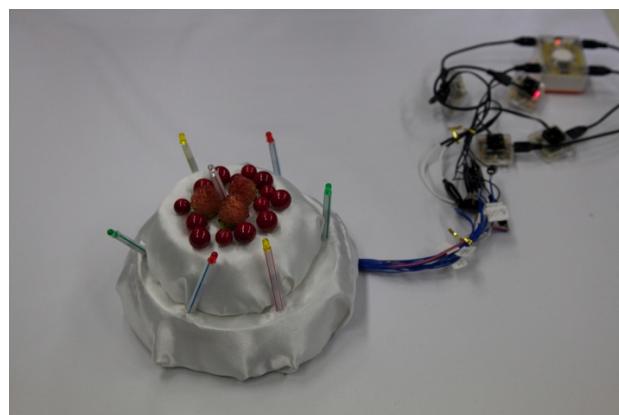


図 4-2-23 電子工作のケーキ

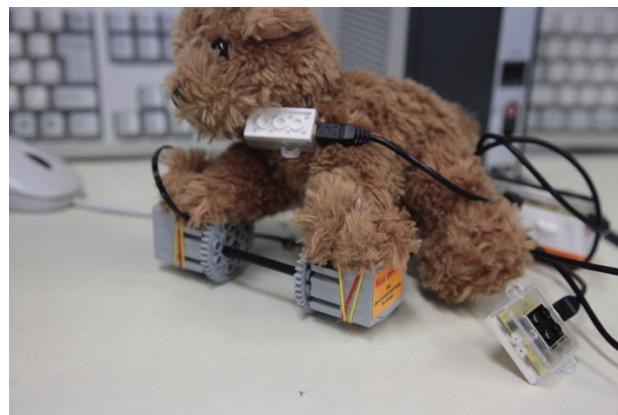


図 4-2-24 ぬいぐるみ

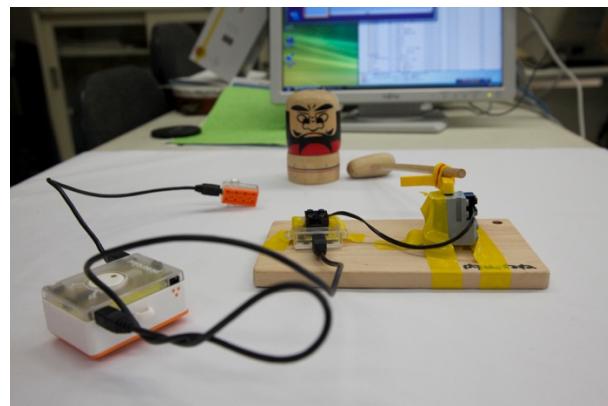


図 4-2-25 だるま落とし

次に児童がつくったプログラムの例を紹介する。図 4-2-26 はリモコン制御できるロボット玩具のプログラムである。2 つの Cricket を使い、送信側と受信側の Cricket に別々のプログラムをつくっている。送信側はボタン操作により送信する赤外線の種類を変え、受信側は送信側から送られてきた赤外線の種類によって、モーターの動きを変えるようプログラムされている。本実践と同様に小学校 4 年生を対象としてプログラミングを授業に取り入れた森ら(2011)の実践にみられたように、多くのプログラムで繰り返しや条件分岐といったプログラミングの概念が活用されていた。

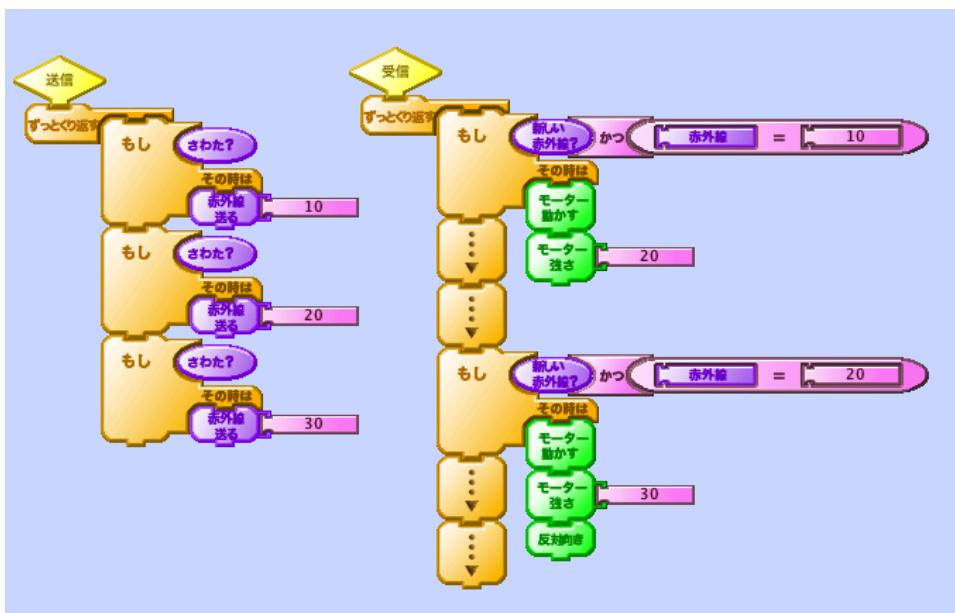


図 4-2-26 プログラム例

4-2-4.児童は何を学んだのか～研究発表会ポスターからの考察

ここでは児童が研究発表会用に作成した研究レポートを分析対象として実践の評価を試みる。研究レポートでは各児童が、①研究のあゆみ、②研究の成果、③振り返りと今後の課題の3項目についてまとめている。児童が本実践を通じて考えたこと、あるいは学んだと感じていることについて検討するため、「研究の成果」と「振り返りと今後の課題」に着目して分析を行う。

4-2-4-1.児童の研究レポートにおける頻出語群

研究レポートに書かれたテキストをデータ化し、テキストマイニング用ソフトウェア(Tiny Text Miner)を用いて形態素解析を行い、頻出語群について確認した。データ化を行った35名の研究レポートについて、3回以上の頻出語群のリストを名詞句、動詞句、形容詞句に分けて集計した(表4-2-4、表4-2-5、表4-2-6)。名詞句については「研究」(60回)、「プログラム」(59回)、「モーター」(51回)、「クリケット」(40回)とCricketを用いた研究という今回のテーマに関連した語句が上位にあがっている。次いで「赤外線」(29回)、「車」(29回)と各児童の研究テーマに関わる語句がみられる。一方で「課題」(11回)、「問題」(9回)、「方法」(8回)、「失敗」(7回)、「成功」(7回)など研究に関わる語句も多くみられた。動詞句では「する」(215回)が最も多く、「なる」(52回)、「ある」(37回)、「やる」(27回)と直接の活動に関わる語句が頻出語句としてあがっている。また「つける」(17回)、「かえる」(13回)など製作活動に関わる語句も多くみられた。形容詞句では「速い」(29回)、「重い」(7回)、「うるさい」(6回)が上位にみられ、モーターの制御や作品のバランスなど作品に直接関連する語句が多くみられた。このように頻出語群は、①Cricketに関わる語句(取扱いを含む)、②作品とその製作

に関わる語句、③研究に関わる語句に大きく分けることができた。さらに係り受けについて3回以上の頻度がみられたものを表4-2-7にまとめた。ここでもCricket、作品製作、研究に関する内容の係り受けが多いことが確認できた。

表4-2-4 頻出語群(名詞句)

名詞句									
語句	頻度	語句	頻度	語句	頻度	語句	頻度	語句	頻度
研究	60	課題	11	ヘリコプター	6	回転	4	ギア	3
プログラム	59	リモコン	11	図	6	他	4	コンピューター	3
モーター	51	自分	11	中	6	成果	4	電流	3
クリケット	40	何	11	たくさん	6	学習	4	左右	3
赤外線	29	スピード	10	後進	6	人	4	ドミノ	3
車	29	ブロック	9	実験	6	えんぴつ	4	線	3
それ	28	後ろ	9	そこ	5	電動	4	上	3
私	24	問題	9	手動	5	右	4	声	3
スイッチ	22	指令	8	はじめ	5	固定	4	巻き	3
センサー	19	バランス	8	ロボット	5	時間	4	ぞう	3
前	16	方法	8	だるま	5	左	4	理由	3
おもちゃ	16	ライト	8	タイヤ	5	トーマス	4	動き	3
コード	14	失敗	7	あと	5	意味	4	解決	3
最初	13	成功	7	力	5	導線	3	電池	3
今	13	目的	6	テーマ	5	一番	3	机	3
音	12	部品	6	いろいろ	5	サウンド	3		
これ	12	段階	6	場合	4	実行	3		
手	12	今回	6	今後	4	原因	3		
次	12	前進	6	コントローラー	4	進化	3		

表4-2-5 頻出語群(動詞句)

動詞句					
語句	頻度	語句	頻度	語句	頻度
する	215	行く	5	乗せる	3
なる	52	使える	5	もつ	3
ある	37	さぐる	4	たおれる	3
やる	27	光らせる	4	止める	3
つける	17	鳴る	4	まとめる	3
かえる	13	動かせる	4	かる	3
いく	12	言う	4	入る	3
さわる	9	おこる	4	さがす	3
つなぐ	9	ける	4	出す	3
つなげる	9	がんばる	4	取りつける	3
いう	8	走れる	4	ぶつかる	3
見る	8	ねじれる	4	入れる	3
変える	8	つく	4	まく	3
しれる	6	くりかえす	3	ふりかえる	3
ちがう	6	かむ	3		
いる	5	とれる	3		

表 4-2-6 頻出語群(形容詞句)

形容詞句	
語句	頻度
速い	29
重い	7
うるさい	6
ない	5
すごい	4
軽い	3

表 4-2-7 係り受け

係り受け	品詞	頻度	係り受け	品詞	頻度
する+する	動詞+動詞	7	プログラム+研究する	名詞+名詞	3
次+課題	名詞+名詞	6	前進+後進	名詞+名詞	3
さわる+センサー	動詞+名詞	5	クリケット+つなげる	名詞+動詞	3
研究+する	名詞+動詞	5	コード+ねじれる	名詞+動詞	3
プログラム+する	名詞+動詞	4	音+センサー	名詞+名詞	3
速い+走れる	形容詞+動詞	4	ヘリコプター+ラジコン	名詞+名詞	3
私+研究	名詞+名詞	4	速い+スピード	形容詞+名詞	3
前+後ろ	名詞+名詞	3	研究+成果	名詞+名詞	3
今+なる	名詞+動詞	3	モーター+する	名詞+動詞	3
プログラム+かえる	名詞+動詞	3			

4-2-4-2.研究レポート記述からの考察

次に上述の 3 項目 (Cricket, 作品製作, 研究)について、児童の学びに関わる記述を詳しくみてみる。児童にとって様々な学びや発見があったことがレポートの記述から分かる。以下にその一例を紹介する。まずは Cricket に関する記述からみていく。

このプログラムにすると、一度スイッチを押すと、前進します。そして、もう一度押すと後進することがわかりました。もう少し、プログラムを変えると、動かなくなるので、このプログラムでないと、プラレールは動かないということが、研究をしていて、わかつることができました。

Cricket を制御するプログラムについて、「もう少しプログラムを変えると動かなくなる」という表現からも、何度も試行錯誤を繰り返しながら、思った通りの動きをするプログラムに辿り着いたことが確認できる。

クリケットと初めてであったときは、おもちゃかなあ？と思っていました。だけど、予想とはずれていきました。なんでもできるようなものでした。

この記述からは、Cricket そのものに興味を持っていた様子がうかがえる。また「なんでもできるもの」という汎用性を持ったコンピュータとしての Cricket あるいはコンピュータそのものへの気づきが

みられる。

最初は、ただ、中を知って、どのような回路をつたってクリケットが出来たのかが知りたくて中を見て、研究をしていました。そして、クリケットの中が、虫のような部品、赤外線を受けたり送ったりするものと、スイッチがあって、やっと出来ているようなものということを知りました。

このように手元で Cricket を観察しながら、コンピュータとしての Cricket についてさらに深い発見する児童もいた。Cricket は開発コンセプトとして「Beyond Black Boxes」を挙げており(Resnick et al. 1996), 多くの製品の仕組みが見えなくなり分かりにくくなっている現状に対して、内部をできるだけ見えるように(ブラックボックス化しないように)設計されている。このことも児童が興味を深めた一因となったと考えられる。

また作品製作に関しても、個々の作品についての詳細な記述もみられた。

この研究での成果は手動でしか動かせなかつたおもちゃ(かつたんこ)が電動にすることによって、速いスピードで回るようになった。手動の場合では、約3秒に1回しか回りませんが、電動の場合、最も速いスピードで 0.9 秒に 1 回回ることができます。手動の場合と電動の場合での回るスピードをそれぞれ比較してみると、その差は 21 秒でした。電動は手動より 21 秒も速いスピードで回ることが分かった。

この児童は伝統的な木のおもちゃと Cricket を組み合わせた新しいおもちゃを研究しており、Cricket を使ってモーターで動かすことと、従来のように手で動かすこととの比較を行っている。

次の課題は、もっと色のバランスを考え、クリスマスツリーに取りつける。あと、真っ直い面から、見たら、反射する力があるから、目がいたい。だから、少しくつする。重さのバランスも、ちゃんと考える。

ここでは作品を全体的な視点から見て、次への課題を挙げている。また作品製作において、他の児童から直接的間接的に学んでいる様子が以下の記述から分かる。

はじめは、スイッチ一つでは3つの動きをすることはできないのではないかと思いました。でも、1つだけでも3つの動きができるので、T君の研究から進化させることができたなあと思いました。

とってもうれしかったことは、Mさんにモーターの速さを変えるを使わなければいいよと言われたので、そうしてみたら、より速くてうまくコントロールできるようになった。

ときどき、困っている時がありました。だけど、友だちの作品(研究)を見て「ここをこうすればいいのか」と思いやってみるとちゃんとできたので良かったです。次の課題は、失敗をおそれず、何度も同じことをやらないように自分の目的までがんばりたいです。

作品製作での課題解決には、他の児童からヒントを得ると同時に、教師からのアドバイスが直接課題解決に結びついたケースもみられた。一方で教師のアドバイスが届かずにつましく課題解決ができないまま、児童自身も問題を特定できないままになってしまっていたケースもみられた。

実験の時にもトラブルが、おこりました。それは、プラレールが動かないのです。それも、スイッチをかえても、モーターは動いているけれど、プラレールじたいが動きません。そこで、考えると、たぶん、指令を伝える電波を、何かが、じやましているか、それか中の、こうぞうが、変なことになっているかだと思います。だから、次ためしてみたいです。

Resnick et al. (2009)が、かつての教育用プログラミング言語「Logo」について、うまく学校現場へ浸透しなかった理由として、「プログラミングがうまくいかなかった際には適切な助言を与え、うまくいった際にさらに深い探求をうながすような授業の進め方がなされなかつた。」ことを挙げており、今回の授業でも作品製作やプログラミングの面で適切なタイミングで適切な助言を与えることができていなかつたことが確認できる。

研究については、研究の方法や研究活動そのものについて、様々な学びや発見があつたことが記されている。

プログラムが上手くいかない時は、声に出してプログラムを読み、意味が通じるか、通じないかで決まる。

このようにプログラミングの方法など、本授業での活動に直接関連した事柄に加えて、研究そのもののへの向き合い方についての気づきについての記述がみられた。

けつしてすぐにあきるのではなく、何がダメなのかをしらべ、分かるまで(せいこうするまで)研究をすることが分かりました。

新しい事をやって一番最初に、体験するのは失敗です。何回も失敗してやっと成功するのだなあということが、分かって失敗はわるくなくて、それをどういかすかが、問題なんだなあとと思いました

また今回の活動と身のまわりにあるものに結びつけて、日常生活へ発展させて考えている記述もみられた。

とくに、前まではできなかつたけれど今できるようになったことが増えたのが良かったです。これなら、もっと生活にやくだてることができると思います。例えば、エレベーターで下にスイッチをつけておいてしていされた重さをこえたらスイッチがおされてブザーが鳴るなどです。もっと生活につなげようと思います。

このように授業を通じた児童の学びは多岐に渡っており、各児童に応じた評価方法を構築することが課題である。

4-2-5.本節のまとめ

本実践では計16回のデジタルものづくり授業を通じて、小学4年生がCricketを用いて、順次や繰り返し、条件分岐を含むデジタルものづくりを行うことができた。また児童が振り返りとしてまとめたテキストを分析した結果、本授業を通じた学びが個人のものづくりや興味に応じて、多岐に渡って拡がっていることが確認できた。

4-3.授業デザイン例 2：小学3年生向け授業

本節では東京都区立A小学校3年生向けに授業をデザインし、実践したディジタルものづくり授業例について述べる。授業は大きく3つのフェーズで構成した。まずフェーズIとしてCricketに慣れることを目的に、Cricketのプログラミングを試しながら、2名1組のグループによる作品づくりを行なった。フェーズIIではモーターの回転を様々な動きに変える仕組みづくりを行い、2名1組のグループ及び個人で作品づくりに活かした。フェーズIIIでは本授業と並行して総合的な学習の時間で進めていた地域のお店調べを題材に、2~4名のグループを作り、地域のお店調べでの学習を活かした作品づくりを行った(図4-3-1)。

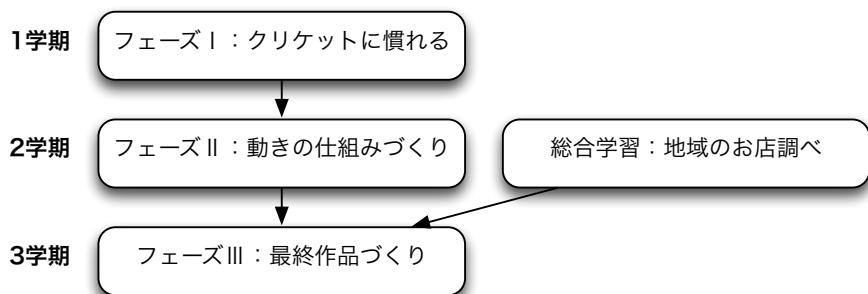


図4-3-1 授業の流れ

4-3-1.授業概要

授業は東京都公立A小学校の3年生1クラスを対象に、主に総合的な学習の時間を使って計17回34時間の授業として実施した。教室は普通教室ではなく、家庭科室や集会室などの特別教室を使い、ノート型コンピュータやCricketとモーター・ライト、スピーカーなどのパーツ、ブロック、工作材料を持ち込んで実施した(図4-3-2)。授業運営は、担任教諭を中心に筆者ら2~3名が授業支援に関わるチームティーチングとして行った。年間の授業スケジュールは表4-3-1の通りである。次に各フェーズでの授業について説明する。



図4-3-2 授業の様子

表 4-3-1 授業スケジュール

	回	日付	時間	概要
フェーズ I	1	2011/6/20	2	クリケットを試す
	2	2011/6/27	2	作品づくりと発表
	3	2011/7/2	2	作品づくりと発表
	4	2011/7/11	2	作品づくりと発表
フェーズ II	5	2011/11/21	2	動きのモジュールづくり
	6	2011/11/28	2	動きのモジュールを使った作品づくり①
	7	2011/12/5	2	動きのモジュールを使った作品づくり②
	8	2011/12/12	2	動きのモジュールを使った作品づくり③
	9	2011/12/19	2	作品発表
フェーズ III	10	2012/1/24	2	作品の計画と作品づくり①
	11	2012/1/31	2	作品づくり②
	12	2012/2/7	2	中間発表
	13	2012/2/14	2	作品づくり③
	14	2012/2/17	2	作品づくり④
	15	2012/2/20	2	作品づくり⑤と発表練習
	16	2012/2/21	2	発表準備、作品発表
	17	2012/2/28	2	ビデオレター原稿づくり、ビデオ撮影

4-3-1-1.フェーズ I : Cricket に慣れる

まず Cricket に慣れることを目標に, Cricket とモーター, スピーカー, ライトのみを取り扱った。Cricket と各パーツの繋ぎ方, コンピュータ上でのプログラミング, プログラムの Cricket へのダウンロード, プログラムの実行方法について説明と実演を行った。また児童が扱うコマンドは表 4-3-2 のように段階的に説明した。さらに Cricket を使った作品づくりに慣れるため, サンプルとビデオで作品例を紹介し, 2 回目から 4 回目の授業ではブロックと工作用材料を使った作品づくりに 2 名 1 組のグループで取り組んだ(図 4-3-3)。また毎回の授業終盤で作品発表を行った。授業後には児童が「発見したこと」, 「これからやってみたいこと」, 「分からぬこと・手伝いがひつようなこと」, 「必要なものやこと」について書き出してもらい, 次回授業の進め方の参考とした。

表 4-3-2 プログラムコマンド

	コマンド
第1回	モータ動かす、モーター止める、反対向き、まつ、ライト色、ライト消す、～の音鳴らす
第2回	ずっとくり返す、くり返す



図 4-3-3 作品づくりの様子

4-3-1-2.フェーズⅡ：動きの仕組みづくり

次に児童が製作する作品に単純なモーターの回転だけではなく、様々な動きを取り入れができるよう、モーターの回転を前後の動きなど、他の動きに変える動きの仕組みづくりに取り組んだ。ギアやカム、シャフトなど約100種類のパーツ(図4-3-4)を用い、動きの仕組みのサンプル(図4-3-5)と組立説明書を用意し、動きの仕組みを使った作品づくりを行った。いずれの児童も10分～15分でサンプルとして用意した動きの仕組みを組み立てることができた。さらにより複雑なモーター制御ができるよう、モーターの回転速度を制御する「モーター強さ」のコマンドと、繰り返しのコマンドを活用することで動きに変化が出ることを紹介した。本フェーズでは最初の2回を2名1組でのグループ作業とし、残りの3回を個人での活動とした。これはフェーズⅠからフェーズⅡのはじめまでの活動がグループによるものであったため、個人での活動によって個人のプログラミングや製作の理解を深めることを意図したためである。

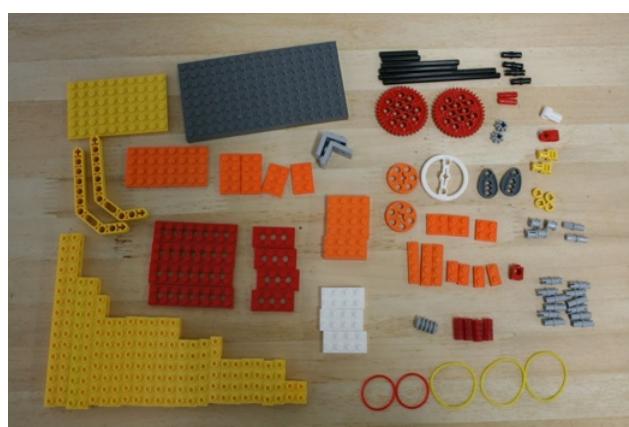


図 4-3-4 ブロック

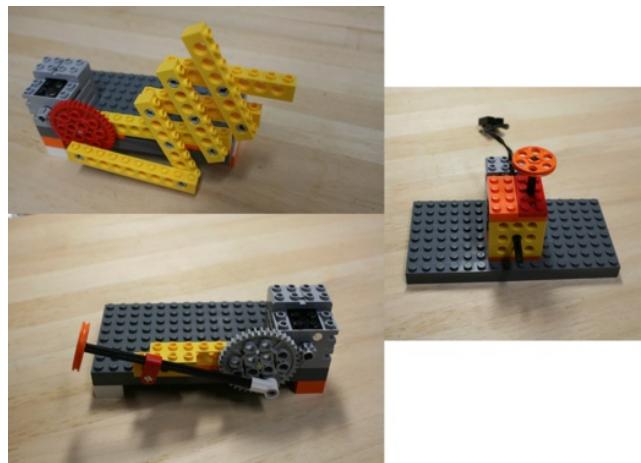


図 4-3-5 動きの仕組み

4-3-1-3.フェーズⅢ:最終作品づくり

総合的な学習の時間で並行して進めていた地域のお店に関する学習を活かし、児童が調べたお店を紹介する作品づくりにグループで取り組んだ(図 4-3-6)。希望する児童には新しくセンサーの使い方と条件分岐を活用したプログラミングの方法、メロディーの作り方について説明を加えた。計 4 回の作品づくりの半ばで中間発表を行い、製作の経過を発表するとともに、他グループの児童からアドバイスやフィードバックを受ける機会を設けた。その後、作品の仕上げと作品発表を行い、最後に題材としたお店の方に完成した作品を見せるため、作品とメッセージをビデオ撮影した(図 4-3-7)。

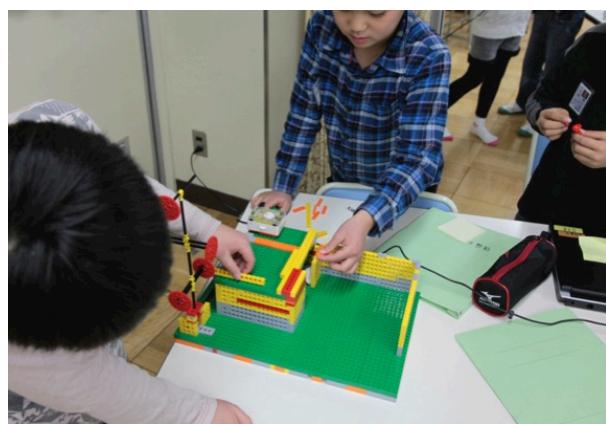


図 4-3-6 作品づくりの様子



図 4-3-7 ビデオ撮影の様子

4-3-2.作品

ここでは本授業実践の成果として、フェーズⅢにて計 7 グループに分かれて児童が製作した作品とその経過について紹介する。複数回にまたがって作品づくりを行う場合、グループによって進め方が異なるため進捗が把握しにくい。そこで本授業では児童の製作過程を残せるよう、360 度全周囲から撮影可能な Ortery 社の PhotoCapture 360 を用いて、毎回の授業後に作品撮影を行った。次に週毎の作品とプログラムの変化に着目して、各グループの作品製作過程を紹介する。

4-3-2-1.グループ A

左右に動きながら光る移動式のお店をつくった(図 4-3-8)。プーリーを使い、モーターの動きを車輪に伝えている。第 1 回目の作品づくりで土台となるモーターと車輪部分をつくった。続く 2 回目で作品の仕組み部分はほぼ完成し、残りの時間で細部の飾りなどのデザインの工夫を行った。プログラムは 2 回目の作品づくりの時間ではライトの点灯のみを試していたが、3 回目の際にモーターを動かすプログラムを追加し、完成させた(図 4-3-9)。

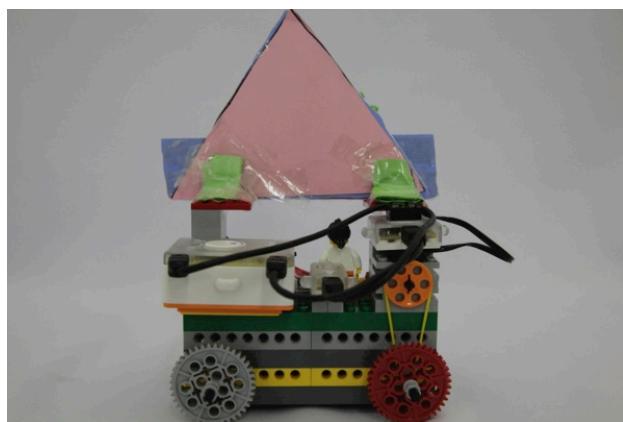


図 4-3-8 グループ A 作品

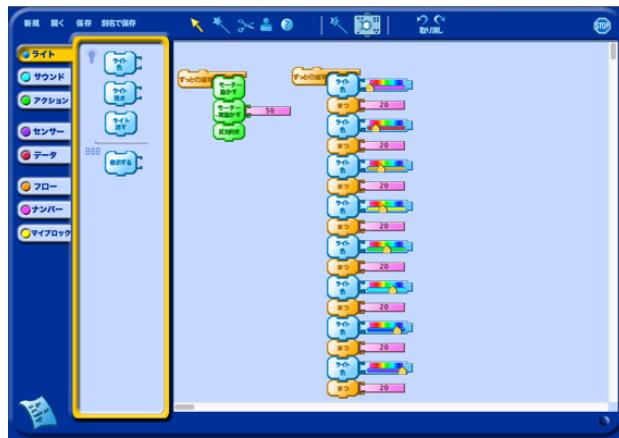


図 4-3-9 グループ A プログラム

4-3-2-2. グループ B

3 つの Cricket を使い、プログラムで動いて光り、音楽の鳴る看板を製作した(図 4-3-10)。2 回目の作品づくりで看板のパネルと上部のライトを組立て、続く 3 回目でモーターで動くキャラクタをついた。その後は、背面に Cricket を置く台をつくるなど工夫を行っていた。プログラムはライトの点灯パターンからつくりはじめ、3 回目の作品づくりの際に音楽を加え、最終的には 3 つの Cricket で音楽とライトの点灯、モーターの動き、ライトの点灯ができるようにプログラムを完成させた(図 4-3-11)。



図 4-3-10 グループ B 作品

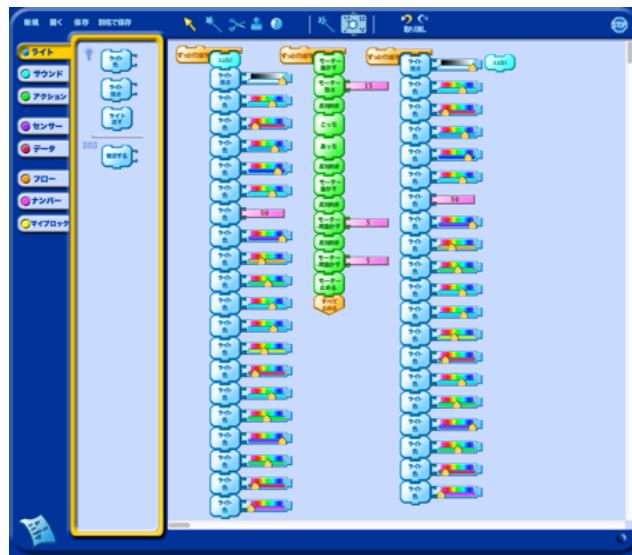


図 4-3-11 グループ B プログラム

4-3-2-3. グループ C

近隣の子どもも施設のミニチュアモデルを製作した(図 4-3-12)。2 つの Cricket を用いて、モニュメントの塔が回転し、建物内が光り、音楽を奏でるようにした。塔の回転にはフェーズⅡで学習したモーターの回転を 90 度変換する動きの仕組みを活用した。2 回目の作品づくりの時間に塔と建物内のライトを完成させた。その後は音楽のプログラムを追加し、モニュメントや建物の入り口などを追加して仕上げた。建物づくりを先行して進め、プログラムは 3 回目の製作時間にライトの点灯パターンとモーターを一定時間毎に反転させる命令を組立て、4 回目と 5 回目の製作時間に音楽と数字をディスプレイに表示させ、カウントダウンをするプログラムを追加し完成させた(図 4-3-13)。

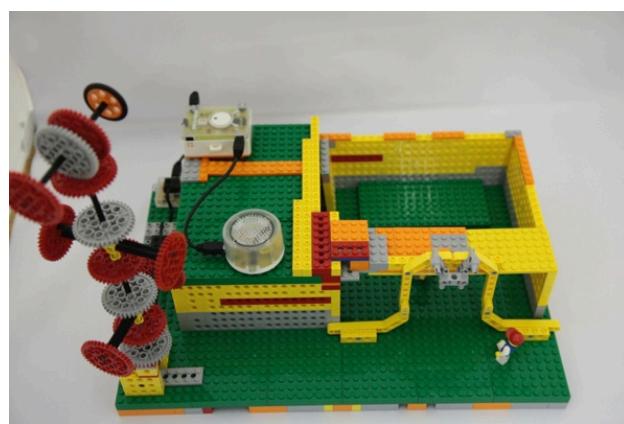


図 4-3-12 グループ C 作品

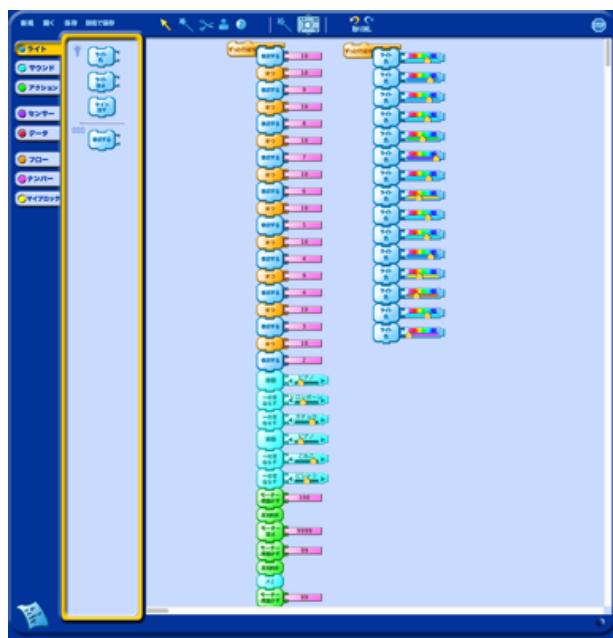


図 4-3-13 グループ C プログラム

4-3-2-4. グループ D

音センサーを使い、自動でドアが開閉し、店内に音楽が流れるお店を製作した(図 4-3-14)。ギアとモーターを使って、お店の入口ドアが自動で開くようにした。2回目の作品づくりの時間より、プログラムと仕組みづくりに分かれて作業を進めていた。仕組みづくりは2回目までブロックを使ったお店の看板づくりに集中していたが、3回目には自動ドアの仕組みづくりを行い、その後店内の飾り付けなどを行い仕上げた。プログラムは音楽から作りはじめ、3回目の製作の時間には明るさセンサーを使い、明るさに反応するプログラムを完成させた。その後、明るさセンサーを音センサーに変更して、音に反応して動きはじめるよう最終のプログラムを完成させた(図 4-3-15)。

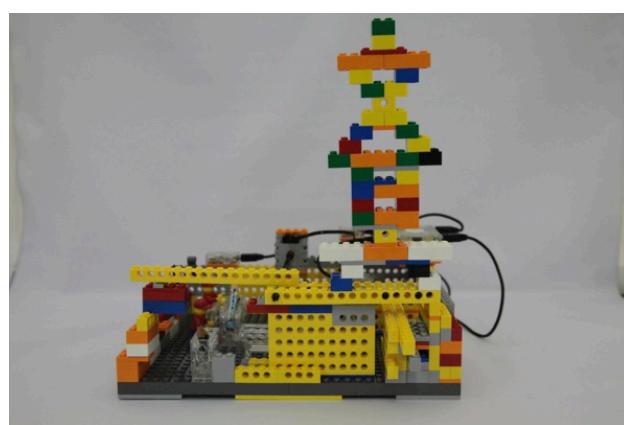


図 4-3-14 グループ D 作品

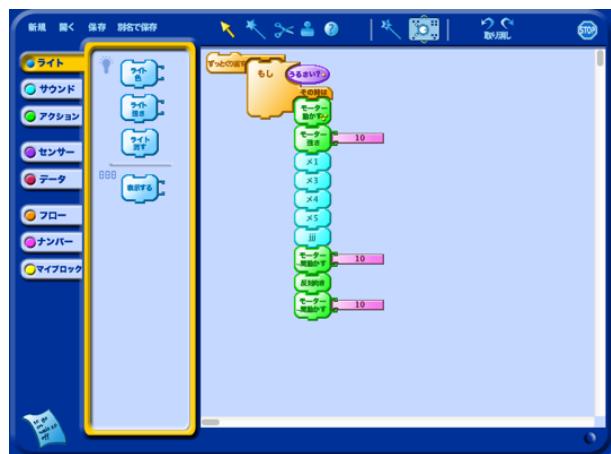


図 4-3-15 グループ D プログラム

4-3-2-5. グループ E

モーターの動きをプーリーで伝え、音に反応して動く電車を作成した(図 4-3-16)。3 回目の作品づくりの時間まで電車の自動扉にこだわって進めていたが、うまく完成することができず、4 回目からは電車そのものをつくり、動くようにした。最終的にはボール紙で本体を覆う電車の外観を作成し、作品に被せて完成させた。プログラムも当初は電車の自動扉を動かすモーターの動きと扉が閉まる際の音楽づくりを行っていたが、4 回目の製作の時間から電車そのものに作品づくりを変更したため、電車を動かすためのモーター制御のプログラムと電車が走る音づくりに変更し、最終的に音に反応して前後に動くプログラムを完成させた(図 4-3-17)。

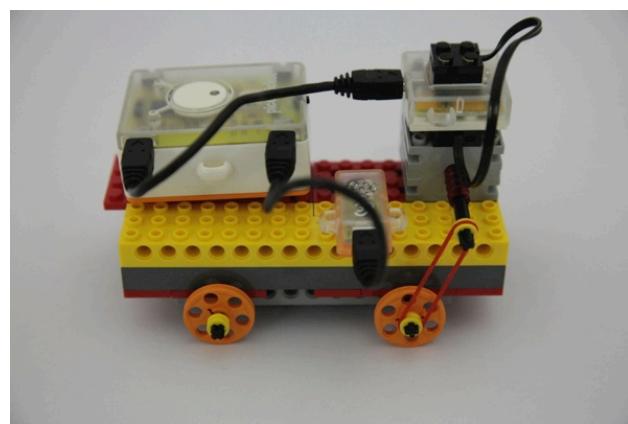


図 4-3-16 グループ E 作品

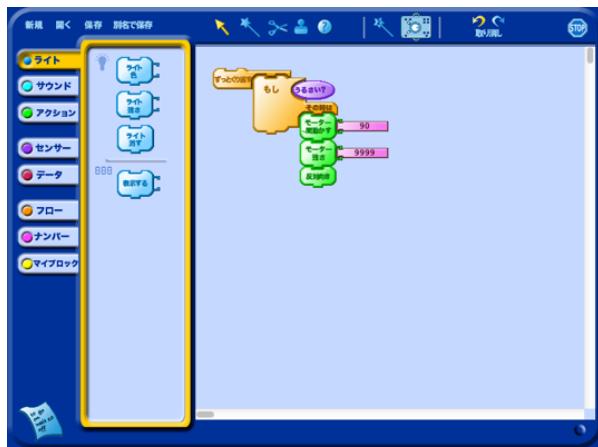


図 4-3-17 グループ E プログラム

4-3-2-6. グループ F

Cricket で動く車を置いた自動車販売店を作成した(図 4-3-18)。2 回目の作品づくりの時間に PRIMARY で動く車をつくり、モーターとライトを繰り返し制御するプログラムも完成させた(図 4-3-19)。グループ F は 2 回目の製作の時間に Cricket に関する部分は完成させてしまったため、その後の製作の時間は店内の装飾づくりに費やした。



図 4-3-18 グループ F 作品

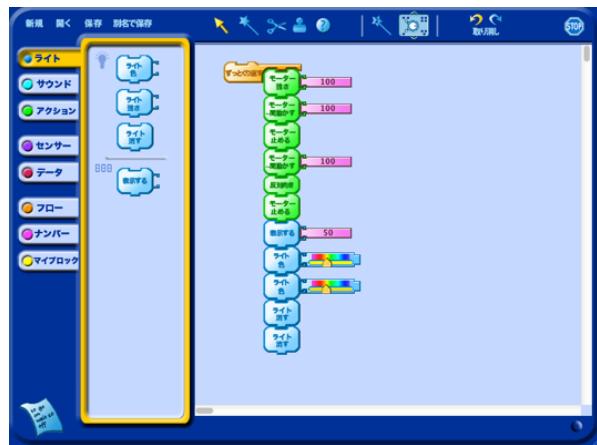


図 4-3-19 グループ F プログラム

4-3-2-7. グループ G

2 つの Cricket を使ってライトが点灯し、回転するお店の看板を作成した(図 4-3-20)。2 回目の作品づくりの時間で回転する看板をつくり、3 回目に光る看板をつくった。さらに車型の看板など、他の看板もつくりたが最終作品には加えなかった。回転する看板はモーターのシャフトに直接つないでまわしている。プログラムはほぼ 1 回目の製作で完成し、その後はメロディの追加やライトの点灯色の調整のみを行った(図 4-3-21)。

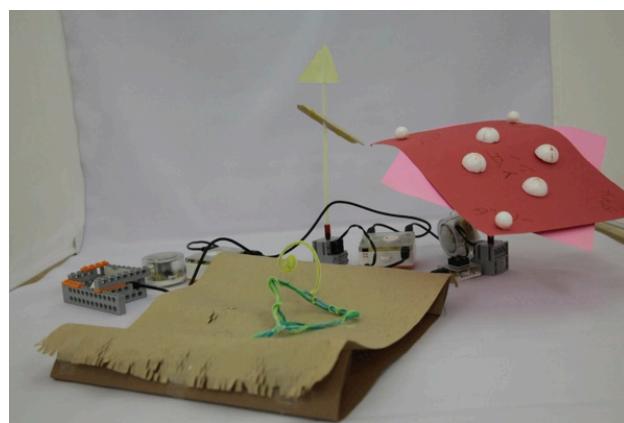


図 4-3-20 グループ G 作品

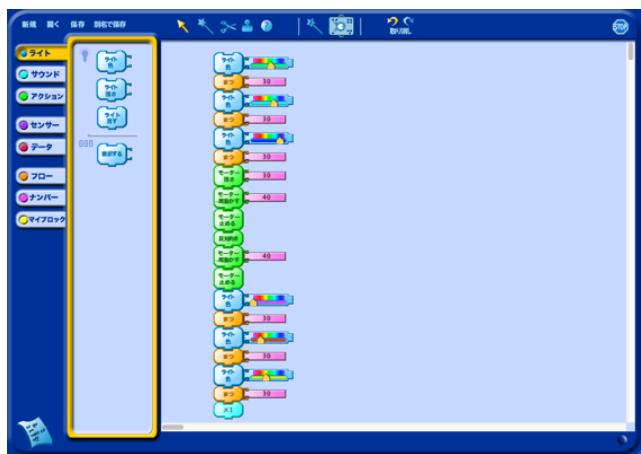


図 4-3-21 グループ G プログラム

4-3-3. 製作過程とプログラムに関する考察

各グループの製作過程を比べてみると、Cricket を使ったデジタルものづくりではグループによって作品製作の過程や作品製作にかかる時間が多様であることが分かる。これはグループによって作品づくりに伴う課題が異なること、あるいはプログラムから製作するか、外観や仕組みから製作するかなどの作業順序やグループ内のメンバーでの作業分担に起因していると考えられる。途中で自動扉から電車へと製作するそのものを変更したグループ E を除く 6 グループは、ある程度作品が完成してからは大きく変更しようとはせずに、作品にプログラムの追加や細かな飾り付けなどの小さな修正に留まっている様子が見られた。児童が作成したプログラムに目を向けてみると 7 グループ中 6 グループで繰り返しを使ったプログラムがつくられていた。さらに 2 グループはフェーズ III で追加説明した条件分岐を活用したプログラムを作成していた。プログラムの大きさをコマンド数で確認すると、1 つの Cricket あたり平均 13.9 コマンド(標準偏差 5.9)となり、まとまったプログラムを作成したことが確認できた一方で、動作には支障がないものの「モーター止まる」を 2 回重ねるなど、不要なコマンドブロックを使っているケースも見られた。

4-3-4. 本節のまとめ

計 34 時間の本授業実践を通じて、小学 3 年生の児童が無理なく Cricket を用いたデジタルものづくりを行うことができた。特にフェーズ II において動きの仕組みの学習を入れたことで、フェーズ I で自由に製作した当初の作品よりも、より仕組みを意識した作品がつくられていた。ブーリー や音楽の活用、センサーの活用など、グループ間で影響し合う様子も見られた。フェーズ III での児童の製作過程からは多くのグループが 3 回目の製作の時間(6 校時)で、おおよそ作品を完成できたが製作進度はグループにより異なることが分かった。これはグループもしくは児童によりものづくり過程における課題や興味が異なるためであると考えられる。授業という時間的の制約がある環

境のなかで、進度の異なる児童グループに対して、さらに作品を発展させるために、どのタイミングで、どのような助言をすべきかについて検討が必要である。あわせて、動作上問題のないものも含めて、正確なプログラムづくりのための支援についても検討すべき課題である。

4-4.授業デザイン例 3： 小学 6 年生向け授業

本節では大阪府公立 I 小学校の 6 年生向けにデザイン、実施した Cricket を活用したデジタルものづくり授業について報告する。授業は 2009 年 10 月に 2 クラス計 56 名の児童を対象に実践した。2 校時の授業を 2 日間実施し、休み時間は授業時間に含めて計 200 分間で授業を構成した。作品づくりのテーマは「何かに反応するロボット」として、児童は 2 人 1 組のグループで Cricket を用いたロボットづくりを行った。

4-4-1.授業の流れ

表 4-4-1 に 2 日間の授業の流れを示す。1 日目は Cricket を使った 2 つのサンプル作品が動く様子を見せながら、どのような仕組みとプログラムで動いているか紹介することからはじめた。次に LED、モーター、スピーカーを制御するプログラムの作成方法についての説明した後、児童は 2 人 1 組のグループに分かれ、プログラミングの練習を行った。その後、音センサーの使い方や条件分岐をするプログラムの作成方法を追加で説明し、グループ毎に 60 分間をロボット作成に費やした。2 日目は光センサーとタッチセンサーの使い方について紹介することからはじめ、50 分間を使ってロボット作成を行った。その後、各グループによる作品発表を行い、最後に 2 日間の活動について、感想をまとめて授業を終了した。

表 4-4-1 授業の流れ

1日目	導入	5分
	授業の紹介、Cricketの紹介	
	Cricketを使った2つの作品例の紹介	
	作品例がどのような仕組み、プログラムで動いているか説明	
	説明: Cricketへのプログラミング方法 その1	10分
	モータ、LED、スピーカーとCricketのつなぎ方	
	モータ、LED、スピーカーを制御するプログラミング	
	コンピュータ上で作成したプログラムのCricketへの転送と実行方法	
	Cricketを試してみる	15分
	最初のプログラムをつくって動かしてみる	
	説明: Cricketへのプログラミング方法 その2	5分
	音センサーの使い方	
	音センサーを活用するために、条件分岐を行うプログラミング	
	作品づくり	65分
	各グループでのロボットづくり	
2日目	説明: 光・タッチセンサーの使い方	5分
	光とタッチセンサーの使い方、プログラミング	
	作品づくり	50分
	各グループでロボットづくりと仕上げ	
	発表	40分
	グループ毎に作品紹介を行う	
	振り返り	5分
	各自の活動について文章でまとめる	

本授業は、(1)導入と説明(Cricket の使い方とプログラミングの方法)、(2)プログラミングの練習、(3)作品づくり、(4)発表、(5)振り返りの 5 つのパートで構成している。またプログラミングに際し、児童に説明したコマンドは表 4-4-2 の通りである。これらのコマンドはモーターや LED、スピーカー

一の制御方法として説明し、その後、繰り返しの方法や条件分岐のためのコマンドを追加で説明した。

表 4-4-2 説明したコマンド

1) 説明: Cricketへのプログラミング方法 その1 ライト色、ライト消す、まつ、～の音ならす、モータ～間動かす、ずっと、くり返す
2) 説明: Cricketへのプログラミング方法 その2 もし～その時、もし～その時は～その他は、メロディ、リズム
3) 説明: 光・タッチセンサーの使い方 さわった？、暗い？、～でない

作品づくりのパートでは児童はコンピュータで Cricket をプログラミングし(図 4-4-1)，様々な材料を用いて外観のデザインや動きの機構づくりを行った(図 4-4-2)。児童は作品づくりのなかで何度も繰り返し、試行と改良を行なった。



図 4-4-1 コンピュータ上のプログラミングの様子



図 4-4-2 グループでの作品づくりの様子

4-4-2. 機材、材料、道具

本授業では 2 名 1 組でグループをつくり、グループ毎にノート型コンピュータと Cricket 一式 (Cricket 本体、モーター、LED、スピーカー、音・光・タッチセンサー、各種ケーブル)を使用した。

また様々なサイズの LEGO®ブロックやギア、カム、ブーリーなどに加え、約 30 種類の工作用材料、空き箱やペットボトルなどの廃材、紙コップや割り箸などの身の回りのものを材料として、ハサミやテープ等の工具とともに用意した（表 4-4-3）。

表 4-4-3 本授業で使用した機材、材料、道具

機材	ノートPC、プロジェクター、スクリーン、PICOCricket一式（本体、モーター、LED、スピーカー、音・光・タッチセンサー）	
材料	ブロック	ギア、カム、ブーリー、シャフト、接続パーツ、20種類の形状が異なるブロック
工作用	ペットボトル、牛乳パック、空き箱、段ボール、発砲スチロール、輪ゴム、針金、紙皿、紙コップ、割り箸、ストロー、毛糸 結束バンド、風船、ビニールひも、洗濯ばさみ、麻ひも、つり糸、シール、軍手、クリップ、ゴムひも、わた、羊毛、ばらん、おりがみ、色画用紙、アルミホイル、紙袋、ビニール袋	
道具	セロハンテープ、ビニールテープ、両面テープ、養生テープ、のり、ボンド、はさみ、ホチキス、カラーマーカー	

4-4-3. 場のデザイン

本授業では通常教室を使用し、図 4-4-3 のようにレイアウトした。

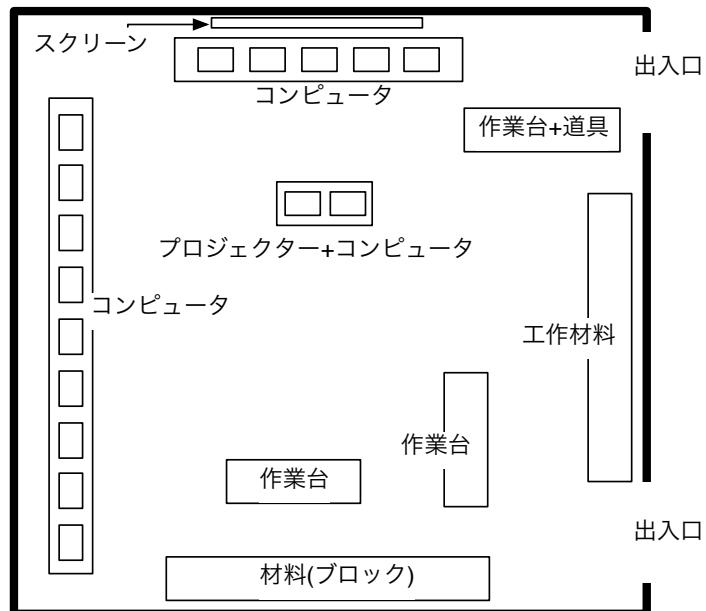


図 4-4-3 教室レイアウト

各グループが同時にプログラミングできるように、グループ数分のコンピュータを別々の机の上に配置した（図 4-4-4）。また椅子は児童が自由に動けるように、あえて置かないようにした。

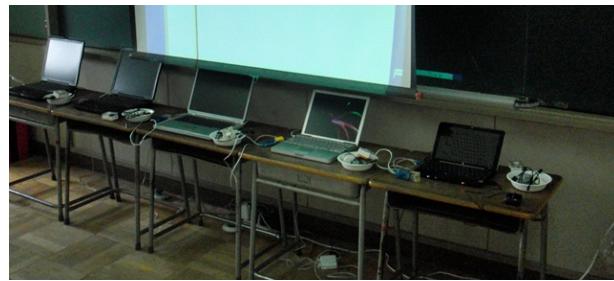


図 4-4-4 コンピューター

LEGO®ブロックと工作材料は置き場を分けた。さらに種類毎に分けておくことで児童が選びやすいように配置した(図 4-4-5, 4-4-6)。



図 4-4-5 工作材料



図 4-4-6 ブロック

ハサミやペン、テープなどのツールを置いた作業台をつくり、組み立てや糊付け、色づけなどの作業スペースとした(図 4-4-7)。



図 4-4-7 作業台

4-4-4.結果と考察

4-4-4-1.児童の作品

本授業で児童が作成した作品例を紹介する。図 4-4-8 のロボットはプログラムで動く車である。LED ライトが黄色から青色にかわると、車は音を出しながら動きはじめる。タッチセンサーが作動すると、LED が黄色で点灯し、Cricket は短いメロディーを演奏する。その 2 秒後に、Cricket が LED を青色にして点滅させ、短いメロディーが再び演奏された後、車が動きはじめる(図 4-4-9, 左)。

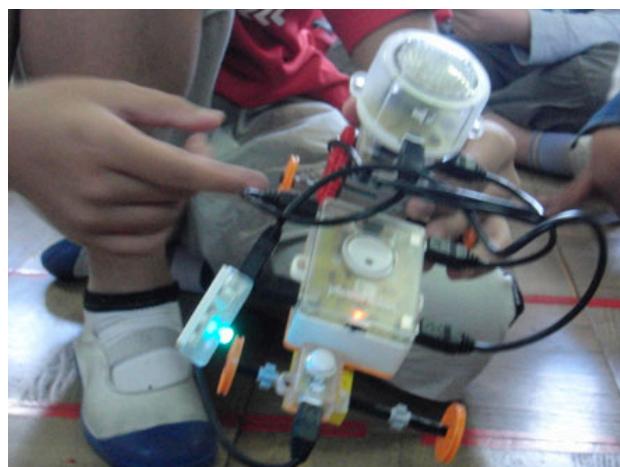


図 4-4-8 プログラムで動く車

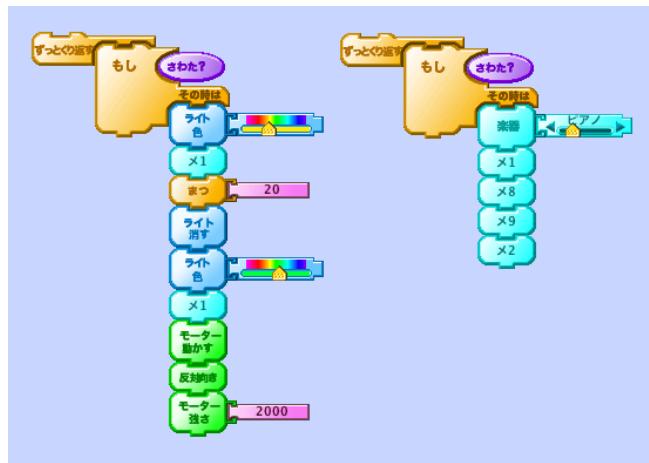


図 4-4-9 児童が作成したプログラム

図 4-4-10 は Cricket と牛乳パックで作られた電子オルゴールである。児童がタッチセンサーに触ると、4 つのメロディーが演奏される(図 4-4-9 右)。



図 4-4-10 電子オルゴール

4-4-4-2.児童によるプログラム

児童のプログラミングに関する理解を確認するため、授業で児童が作成した 26 のプログラムを検証した。プログラムのサイズをコマンドブロック数で数えると平均が 11.3 ブロック(SD 7.88)となつた。6~10 のコマンドブロックを使用しているグループが最も多く、コマンドブロックを最も多く使用したグループのコマンドブロック数は 42 であった(図 4-4-11)。

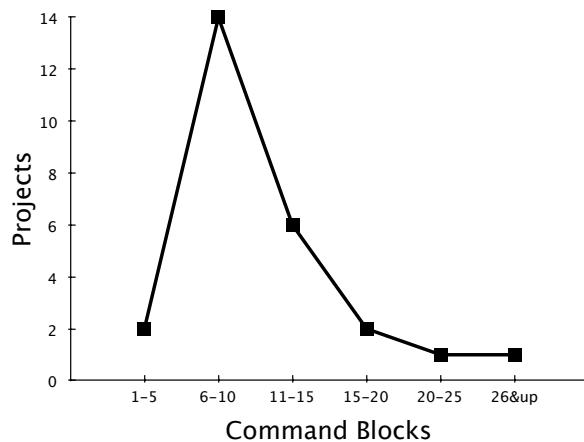


図 4-4-11 コマンドブロック使用数の分布

各グループのコマンド使用状況を確認すると、繰り返しに関わるコマンドブロック（「くり返す」「ずっとくり返す」）は、1組を除き25組（利用率：96.2%）で利用されていた。また条件分岐に関わるコマンドブロック（「もし～なら」）は21組（利用率：84.6%）で活用されていた。このように本授業を通じてプログラミングに関する重要な概念である順次、繰り返し、条件分岐について多くの児童が活用できることことが分かった。

表 4-4-4 コマンドブロックの利用率

カテゴリ	コマンドブロック	利用率	カテゴリ	コマンドブロック	利用率
フロー	ずっとくり返す	96.2%	サウンド	楽器	15.4%
ライト	ライト 色	88.5%	サウンド	リズム	15.4%
フロー	もし その時は	84.6%	ライト	表示する	11.5%
ライト	ライト 消す	61.5%	サウンド	音 ならす	11.5%
アクション	モーター動かす	57.7%	サウンド	テンポ	11.5%
センサー	うるさい？	57.7%	サウンド	ボリューム	11.5%
サウンド	～の音ならす	50.0%	アクション	反対向き	7.7%
アクション	モーター～間動かす	46.2%	フロー	くり返す	7.7%
サウンド	メロディ	42.3%	サウンド	チッ	3.8%
アクション	モーター止める	34.6%	アクション	こっちは	3.8%
フロー	まつ	34.6%	アクション	あっち	3.8%
アクション	モーター強さ	26.9%	センサー	暗い？	3.8%
センサー	さわった？	26.9%	センサー	タイマー 戻す	3.8%
ライト	ライト 強さ	19.2%	フロー	～までまつ	3.8%

一方で前述のように児童が使用したコマンドブロックの数は限られており、表 4-4-4 に見られるように使用されたコマンドの種類も限られていたことがわかる。

4-4-4-3.事前事後アンケートの結果

本授業では児童の「プログラミング」「ものづくり」「コンピュータを使ったものづくり」への興味関心

について、授業前後で質問紙調査を実施した(5件法／5:とても興味がある～1:全く興味がない)。表4-4-5は授業前後の質問における平均および標準偏差の結果である。t検定を行ったところ、「プログラミング」と「コンピュータを使ったものづくり」について児童の興味関心の向上に1%水準での有意差が見られた(プログラミング $t(51)=3.487$, $p < .01$, コンピュータを使ったものづくり $t(51)=2.771$, $p < .01$)。

表4-4-5 プログラミング、ものづくり、コンピュータを使ったものづくりへの興味

	授業前 (N=27)		授業後 (N=26)		**
	M	SD	M	SD	
プログラミング	2.81	1.24	3.96	1.15	**
ものづくり	3.89	1.15	4.15	0.92	
コンピュータを使ったものづくり	3.11	1.34	4.04	1.08	**

4-4-5.本節のまとめ

Cricketと日本語化したプログラミング環境を使用することにより、小学6年生が4校時でコンピュータを埋め込んだプログラムで動く作品づくりを行うことができた。また25グループ中21グループの作品には、順次、繰り返しと条件分岐を含むプログラムが活用されていた。授業前後の質問紙調査では、本授業を通じてプログラミングとコンピュータを使ったものづくりへの関心の向上がみられた。

第5章 Scratchを活用した小学校授業のデザイン

5-1. プログラミング環境 Scratch(スクラッチ)

Scratch は MIT メディアラボが開発したプログラミング環境である。インターネット上で無償配布されている。ブロック型コマンドを組み合わせることで、視覚的にプログラミングできる。コマンドは多言語に対応しており、日本語でのプログラミングも可能である(図 5-1-1)。作品は Scratch の「共有」ボタンを押すことで、作品公開サーバへ自動的にアップロードされる。

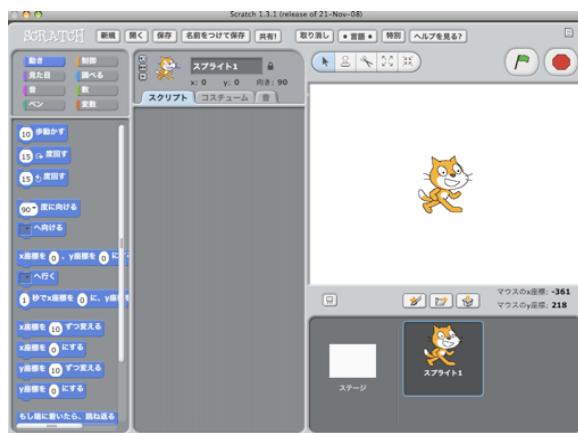


図 5-1-1 Scratch 1.3.1

プログラミングは画面上の各スプライトに対して行う。図 5-1-1 のネコを動かす場合、画面左上から「動き」カテゴリを選択。左下に表示されるコマンド一覧よりブロックを選択し、中央のプログラミングエリアにドラッグする。ブロックを繋げることで複数のコマンドを順に実行できる。プログラムの実行には、実行したいブロックをダブルクリックする。画面右上のプログラム実行ウィンドウは 480x360 ドットとなっている。また、フルスクリーンでもプログラムを実行できる。

図 5-1-2 のプログラムではネコが 10 歩動き、右に 15 度回転することを 10 回繰り返す。繰り返し関連するブロックは、繰り返し対象のブロックを中にはめ込むなど、ブロック形状と機能がリンクしているため、エラーが発生しにくい。



図 5-1-2 Scratch プログラム例

本章での授業実践で利用した Scratch1.3.1 では「動き」「制御」「見た目」「調べる」「音」「数」「ペン」「変数」の 8 カテゴリに 115 種類のブロックが用意されている。「動き」に関するブロックには、画面上のスプライトを動かすためのコマンドの他、XY 座標に関するコマンドが含まれる。「制御」には繰り返しや条件分岐、「見た目」にはアニメーションや画像効果、「調べる」にはセンサーの値やキ

一入力の判別処理、「音」には効果音や音ファイルの再生、音階等を指定した音の再生、「数」には四則演算記号や等号不等号記号、乱数、「ペン」にはLogoのタートルグラフィックに対応するペン操作、「変数」には変数やリストに関するブロックが含まれる(図 5-1-3)。また内蔵のペイントエディタでスプライトを描くことや外部画像ファイルを読み込んで利用することも可能である。

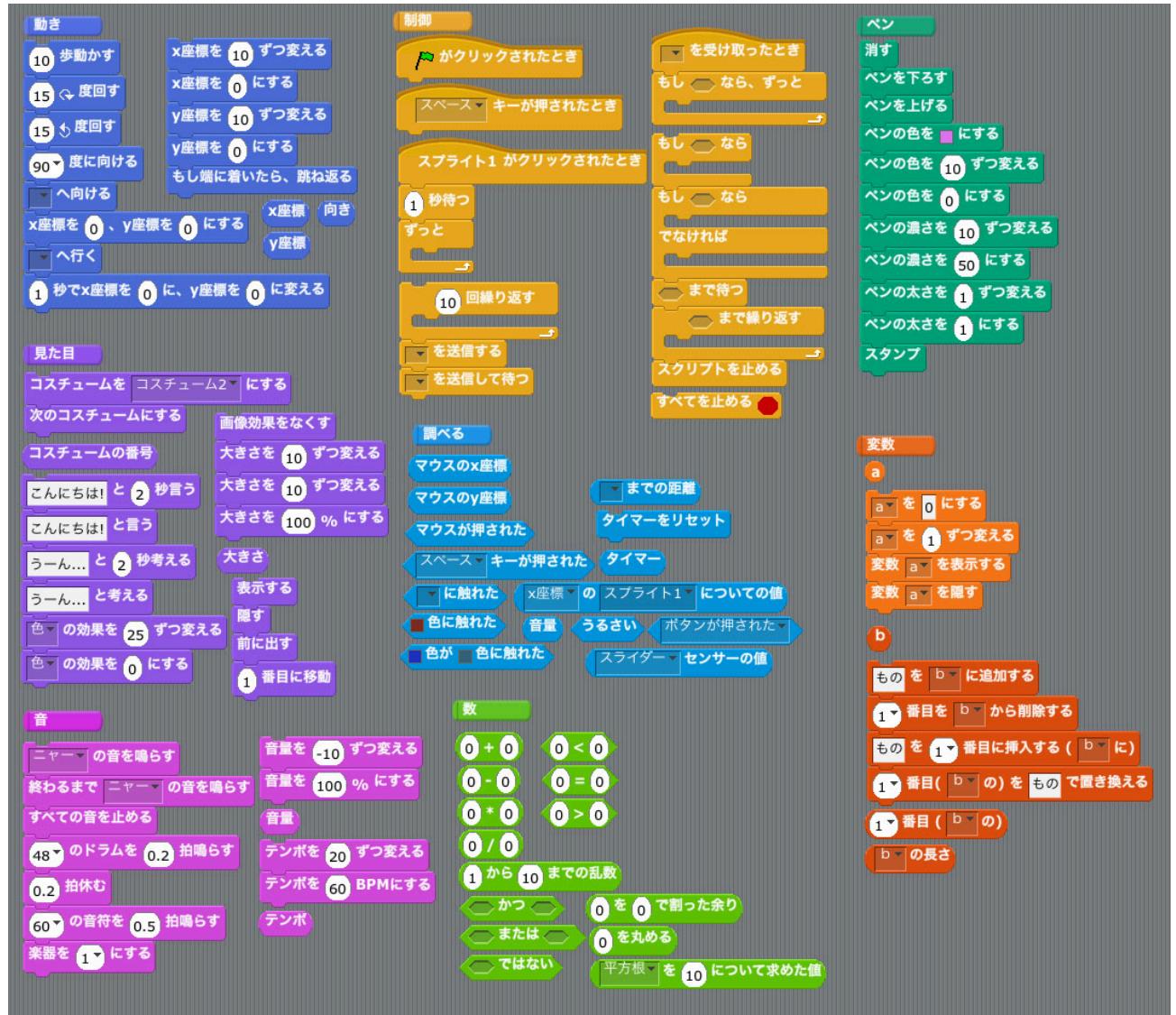


図 5-1-3 Scratch1.3.1 ブロックリスト

Scratchはキーボードやマウスからの入力を受け付けるほか、光センサー、音センサー、タッチスイッチ、スライダーおよび4つの抵抗センサー用のポートが備えられている専用のセンサーボードを用いることで、外部からの情報を取り込むことができる(図 5-1-4)。

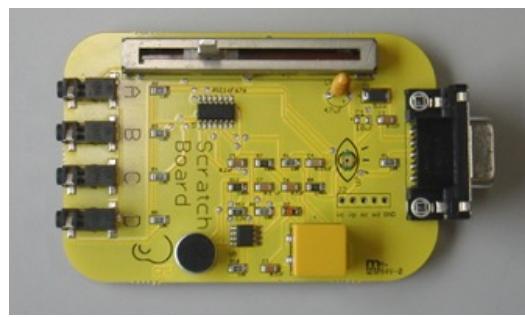


図 5-1-4 センサー ボード

5-2. 授業デザイン例1：小学4年生向け授業

本実践は奈良女子大学附属小学校の4年生1クラス38名(男子19名、女子19名)を対象に、2008年6月より2009年3月まで計26時間に渡り行われた。授業は同校のコンピュータ室(児童用40台、教師用1台のWindowsマシンを設置)で行い、指導は情報担当教諭と担任教諭が児童の学習活動全般を支援し、筆者がScratchに関する技術的支援をするチームティーチングとして行った。またScratchによるプログラミングやインターネット上への作品公開の作業に無理がないよう、マウス操作や、文字入力、電子ファイルの保存・管理、インターネット閲覧など、コンピュータの基本操作を学習した4年生を実践の対象とした。

5-2-1. 授業デザインの方針

授業はScratchの特長である直感的操作を活かし、探索的にすすめるプログラミングの練習課題からはじめた。続いてプログラミングの知識を活かす応用課題として、同じくScratchの特長である作品の多様性を活かし、ゲームやインタラクティブなアニメーションなど、児童がテーマを自由に設定し、児童の興味・関心と結びつくような作品づくりに取り組むこととした。さらに作品づくりの経験を活かし、複雑で高度なプログラミングと作品づくりを行うため、2回目の作品制作の時間を設けた。また各時間に発表時間を設け、クラス内で作品や作業について共有するとともに、インターネット上の作品共有を試みた。

5-2-2-1. 授業フェーズ

26時間の授業は「Scratchコマンド探し」「図形を描く」と2回の作品制作「作品づくり①」「作品づくり②」の大きく4つのフェーズとした(図5-2-1)。



図5-2-1 授業フェーズ

プログラミングから実行までの基本操作からはじめ、スプライトを動かすための基本コマンド(画面上を～歩動かす、～度回す等)、繰り返し、描画のためのペン操作、座標、条件分岐、スプライト間連携(スプライト間でのメッセージ送受信)、センサーの値、等号不等号に関するコマンドについて、各フェーズの中で順次使い方を説明した(図5-2-2)。またフェーズ1から3は個人作業、フェー

ズ4はグループ作業とした。

フェーズ1	フェーズ3
～歩動かす、～度回す ～回繰り返す、ずっと（繰り返す） ～秒待つ	旗がクリックされたとき ～キーが押されたとき ～まで待つ、もし～ならずっと
フェーズ2	フェーズ4
ペンを下ろす、ペンを上げる X座標を～Y座標を～にする	～センサの値、<、>、=

*フェーズ1については英語コマンドを説明

図 5-2-2 児童への説明(コマンド)

5-2-2.授業の流れ

図 5-2-3 のように、45 分間の授業では最初の 5 分間で各児童がその日の作業について計画を立てる。続いて 20-25 分間の作業後、ノートに作業結果をまとめ(5 分)、クラス内での発表と質疑応答による共有(5-10 分)、最後に振り返り(5 分)をまとめる流れで授業を実践した。作業中に児童の様子を確認しながら適宜アドバイスするとともに、ノート記述から次時の児童へのアドバイスを検討した。また教師から新しいコマンドブロック等を説明する場合には、計画の前後で説明を加えた。

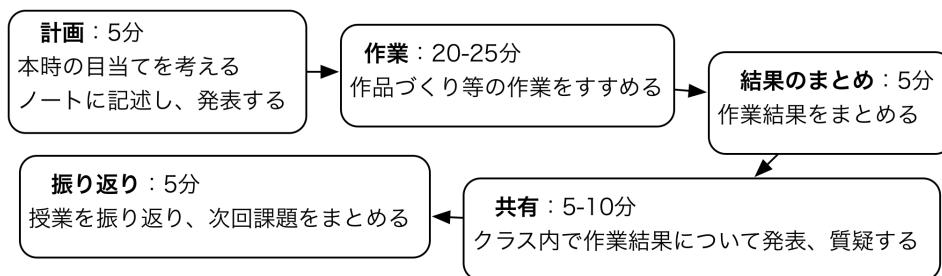


図 5-2-3 1 時間の授業の流れ

5-2-3.各フェーズでの活動

5-2-3-1.フェーズ 1: コマンド探し

Scratch の起動からプログラミング、実行までの方法を説明した後、Scratch に慣れることからはじめた。当初利用していた Scratch1.2 では日本語コマンドブロックが利用できなかった(フェーズ 2 以降では日本語が扱える 1.3 および 1.3.1 を利用)ため、英語で書かれたブロックの意味を探ることを課題にコマンドブロックを試し、その機能をノートに記述した。

5-2-3-2.フェーズ 2: 図形を描く

Scratch はコマンドブロックの組み合わせでプログラミングが可能なため、導入の敷居が低い。反面、単に組み合わせるだけでプログラムが完成してしまうため、偶然ではなく、自分が意図した動作をするためのプログラミングへの意識が向きにくい。そこで次に目的を明確にするため、指定さ

れた図形を作成する課題に取り組んだ。

図形を描くため、移動するスプライトの軌跡を描くペンコマンドを紹介した後、児童は正三角形、正方形、円、長方形、星形等の図形作成課題に挑戦した。図形課題を進めていく間、スプライトを240度回転させるために、120度回転させるコマンドを2つ利用するなど、無駄なコマンドが多く見られた。そこで実行中のコマンドブロックをハイライトする「ステップ実行」機能を紹介し、プログラム実行中に、どの部分が実行されているか確かめながら、無駄のないプログラミングを目指すことを促した。その後、さらに複雑なスプライト制御のため、座標に関するブロックを紹介した。

5-2-3-3. フェーズ 3: 作品づくり①

インタラクティブな作品づくりのため、キー入力の判別処理や条件分岐に関わるブロックについて説明した。また、あるスプライトの動作終了後に、別のスプライトの動作を開始するようなスプライト間連携のブロックについて説明を追加した。作品づくりは画面イメージと機能、作業目標について各回の計画を立てることからはじめ、次に制作に移った。作品計画は隨時変更しても構わないとしたが、各時間で目標をたて、作業後に結果を振り返ることとした。作品完成後に、クラス内で作品発表を行った。Scratch ウェブサイト上にクラス用ギャラリをつくり、インターネット上でも作品を展示したところ、児童の作品に外国から英語でコメントが寄せられ、授業で紹介した。

5-2-3-4. フェーズ 4: 作品づくり②

4~5名のグループ作業として、センサーボードを使った作品づくりを行った。センサーの値や条件設定のための等号不等号のブロックを紹介した後、作品の計画、制作、発表の流れで進めた。1つのコンピュータを使って作業をすすめるグループや、別々のコンピュータで作業を進めてからスプライトの書き出し・読み込み機能を使って統合するグループが見られた。

5-2-4. 評価

ここでは児童の作品におけるプログラムに着目し、本実践の評価を行う。また授業後のアンケート結果からプログラミングへの理解や興味について評価する。

5-2-4-1. プログラムの大きさはどのくらいか

作品づくり①では、個人による38作品、作品づくり②ではグループによる9作品が完成した。作成したプログラムの規模を確認するため、プログラムにおけるステップ数の代わりに、コマンドブロック数について調べた。なおプログラミングエリアには置かれても直接プログラムに使われていないブロックについては数から除外した。作品づくり①におけるコマンドブロック数の平均は39.7個(SD:47.6)であった(表5-2-1)。

表 5-2-1 コマンドブロック・スプライト数

ブロック	作品づくり①		作品づくり②	
	M	SD	M	SD
ブロック	39.7	47.6	29.5	11.9
スプライト	5.2	4.6	4.0	6.1

図 5-2-4 は児童が使用したコマンドブロック数の分布である。10 個以下のコマンドブロックしか使
用しなかった児童は全体の 15.8%にとどまっていた。また 60 個以上使用している児童も 15.8%い
た。作品づくり②ではブロック数の平均は 29.5 個(SD 11.9)と、作品づくり①に比べて少なくなっ
て
いる。スプライト数については作品づくり①および②で、各々平均 5.2 個(SD:4.6), 平均 4.0 個(SD
6.1)となった。

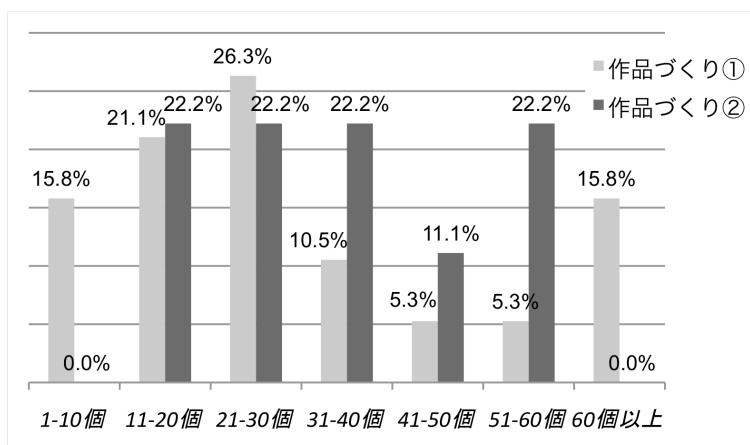


図 5-2-4 ブロック数の分布

5-2-4-2.どのようなコマンドブロックが使われたか

次に使用コマンドブロックをカテゴリ別に見てみる(表 5-2-2)。

表 5-2-2 カテゴリ別平均使用ブロック数と全体比

	作品づくり①			作品づくり②		
	M	SD	全体比	M	SD	全体比
動き	18.5	27.2	46.6%	3.9	2.7	11.9%
制御	13.9	25.4	35.1%	12.0	8.0	36.7%
見た目	4.3	4.2	10.9%	5.8	5.6	17.7%
調べる	0.1	0.5	0.3%	6.0	4.2	18.4%
音	0.0	0.0	0.0%	0.1	0.3	0.3%
数	0.1	0.6	0.3%	3.7	5.1	11.2%
ペン	2.3	6.3	5.8%	1.2	3.3	3.7%
変数	0.0	0.0	0.0%	0.0	0.0	0.0%

作品づくり①と作品づくり②の全作品で使用されたコマンドブロックを Scratch の 8 つの命令カテゴリを用いて集計した。作品づくり①では「動き」が全体の 46.6% を占め、次いで「制御」35.1%、「見た目」10.9% となっている。対して、作品づくり②では、「動き」は平均 3.9 個と作品づくり①の平均 18.5 個に比べて少ない。代わりに「制御」が 36.7% と最も多く、「調べる」18.4%、「見た目」17.7%、「動き」11.9%、「数」11.2% となっている。「調べる」および「数」ブロックの使用増はセンサーボード利用のためと考えられる。また「音」ブロックの使用が少ないとことについては、コンピュータにスピーカーが繋がっていないかったためと考えられる。「変数」が作品づくり①②ともに利用が極端に少ない理由としては、授業中に説明しなかったことに加え、他ブロックに比べ、利用の難易度が高いことが考えられる。画面上でのスプライト制御や繰り返しなどを含む作品づくりは、作品に合わせずあえて利用しなかった児童を除けば、ほぼ全員が達成できた。キー入力の判別処理や条件分岐についても、9 割近い児童が取り組めた。作品づくり②では、図 5-2-5 のように同じ構造を持つプログラムをスプライト内や複数のスプライトで利用した作品が見られた。

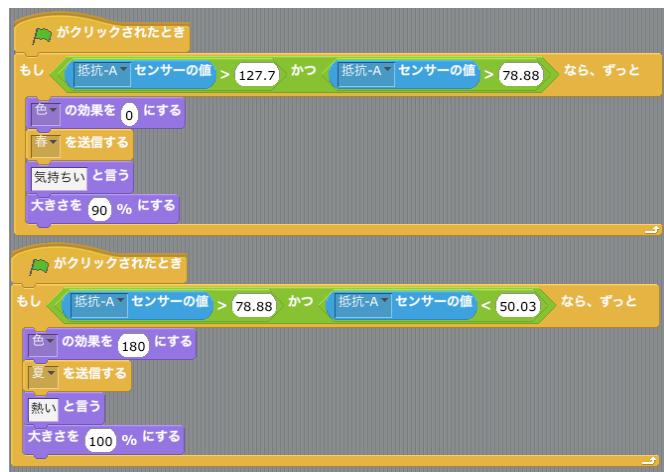


図 5-2-5 同一構造プログラムの利用

既に作品づくりの経験があったこと、センサーボードを使った作品であったこと、グループ作品であったことなどの理由が考えられるが、作品づくり②における平均利用ブロック数の減少からも、作品づくり①でいろいろ試した段階を経て、児童がより効率的なプログラミングを達成したことがうかがえる。

5-2-4-3. ブロックの理解度とその利用

授業後に、主要なブロックの理解度について、アンケート調査を行った(N=36)。プログラミングの理解度を検証するため、スプライト制御(～歩動かす、～度回す)、繰り返し(ずっと、～回繰り返す)、座標(X 座標を～Y 座標を～にする)、キー入力の判別処理(～キーが押された時、旗がクリックされた時)、条件分岐(もし～ずっと)、スプライト間連携(～を受け取ったとき、～を送信する)の 6 項目について、「5:理解できた」から「1:分からなかった」までの 5 段階で評価してもらった。また、児童の自己評価と照らし合わせるため、該当するブロックの作品内での利用率についても調べた(表 5-2-5)。

表 5-2-5 ブロック理解度と利用率

	理解度		利用率	
	M	SD	作品づくり①	作品づくり②
スプライト制御	4.97	0.17	86.8%	22.2%
繰り返し	4.94	0.24	68.4%	100.0%
座標	3.97	1.19	23.7%	66.7%
キー入力の判別処理	4.82	0.72	89.5%	77.8%
条件分岐	4.32	1.09	7.9%	88.9%
スプライト間連携	3.65	1.57	23.7%	11.1%

理解度はいずれも高い結果となった。特にスプライト制御と繰り返しについては、ほとんどの児童が 5 段階中の 5 と最高評価をしており、利用率からも作品づくり①と②の間で差があるものの実際に使えていたことが確認できる。次いで、キー入力の判別処理についても理解度と利用率が高い。条件分岐については、作品づくり②では利用率が高いが、個人で制作した作品づくり①ではあまり利用されなかつたこともあり、理解度に若干ばらつきが見られる。座標についても同様の傾向が見られる。またスプライト間連携については、6 項目中、理解度評価と利用率ともに最も低い結果となった。次に児童の作品例から、具体的なコマンドブロックの利用について見ていくたい。

5-2-4-4. スプライト制御と繰り返しによる作品例

図 5-2-6 は作品づくり①の作品である。自動車レースや雪の降る動き、海中の生き物の様子、自分の名前を書く作品など、様々な作品が完成したなかで、同作品は森林破壊をテーマとした。人間がのこぎりで森林の木を伐採しているとサメが飛び出して、メッセージが表示される。



図 5-2-6 作品画面

図 5-2-7 左のプログラムでは、のこぎりの動きをプラス、マイナス方向にスプライトを動かし、それを繰り返すことで表現している。図 5-2-7 右では鋸で切られた木が倒れる。木の動きは少しずつ回転するようにスプライトを制御することで表現した。また冒頭に 5 秒待つブロックを使い、倒れるタイミングを調整している。



図 5-2-7 繰り返しによるのこぎり(左)と木(右)の動き

5-2-4-5. 条件分岐による作品例

図 5-2-8 はセンサーボードを使った作品づくり②の作品である。スライダを使ってキャラクターの動きを制御する作品や音に反応する作品など、よりインタラクティブな作品が目立った。図 5-2-8 の作品は、明るさに反応して画面上のスプライトの色が変化する。



図 5-2-8 作品画面

プログラムには明るさセンサーの値が「0 より大きく、かつ、10 より小さい」場合は、色の効果を 0 にするという条件分岐が使われている(図 5-2-9)。



図 5-2-9 条件分岐による色変化

5-2-4-6.プログラミングへの興味

授業後に Scratch を使ったプログラミングへの興味について、楽しさの観点からアンケートを行った(N=36)。5 段階[5:楽しかった～1:つまらなかった]で評価をしてもらったところ、平均 4.78(SD 0.17)の高い評価を得た。あわせて簡単だったか？という質問をしたところ(5 段階[5:簡単だった～1:難しかった])、平均 3.22(SD 1.27)と必ずしも簡単とは思っていなかったことが明らかになった(表 5-2-6)。児童の感想からも「スクラッチの初めの学習では苦戦していましたが友だちや先生から教えてもらってだんだんとわかってきてスクラッチを楽しむことができました」など、最初の時点での難しさを感じている児童がいたことも確認できた。

表 5-2-6 授業後アンケートの結果

	M	SD
スクラッチを使ったプログラミングは楽しかったですか？	4.78	0.76
スクラッチを使ったプログラミングは簡単でしたか？	3.22	1.27

5-2-5.本節のまとめ

26 時間の授業を通じて、練習課題からはじめ、画面上でスプライトを動かすなどの制御や繰り返し命令を含めた作品をつくることができた。また、条件分岐やキー入力の判別処理といったより複雑なプログラミングを含めた作品づくりにも 8 割を超える児童が取り組むことができた。「楽しかったか」という質問への評価も高く、プログラミングを活用したデジタルものづくりが Scratch を用いることで小学 4 年生が無理なく達成できた。

今後は、児童のプログラミングを通じた学習過程やその効果についての検証を課題としていきたい。26 時間という時間数をプログラミング教育に当てるのは難しいため、時間数を減らした実践が可能なようにフェーズ毎の内容や時間配分についても精査したい。あわせて今回は 4 年生を対象としたが、5~6 年生を対象とした場合や 4~6 年生を対象に 3 年間に分けて実践する場合などについて検証していきたい。また、本実践は筆者が授業運営をサポートする形で実践を進めたが、教師だけでも実践できるような研修の実施等、支援体制の構築が課題である。

5-3. 授業デザイン例 2：小学 4 年生、6 年生向け授業

5-3-1.はじめに

プログラミング教育を総合的な学習の時間など、教科外学習だけに収めてしまうと、小学校においては依然としてプログラミング教育が一過性のものとなってしまう危険性を持つ。そこで本実践では児童が習得したプログラミングの知識を、発展的学習として教科学習の中で活かすことを試みた。つまりワードプロセッサーを活用した作文、プレゼンテーションツールを活用した発表学習、あるいはインターネットを活用した調べ学習と同様に、プログラミングを通常の教科学習の中で活かすことを目的に授業デザインと実践を行った。

5-3-2.Scratch を活用した理科学習

本実践では、教科学習のなかでも理科の中でプログラミングを活用することを試みた。具体的には各種実験の際に Scratch とセンサーボードを用いて、児童が独自の実験道具づくりを行った。ここでは小学 4 年の熱单元、小学 6 年の電気单元の実践を報告する。

5-3-2-1.小学 4 年理科：熱单元でのプログラミング活用

「もののあたたまり方」に関する授業では金属の熱の伝わり方についての学習に取り組んでいた。多くの児童がろうそくやサーモテープなどを使って実験道具をつくる中で、一部の児童が Scratch をツールとして選択し、Scratch 及びセンサーボード、熱センサーを使った実験道具づくりに取り組んだ。図 5-3-1 の作品ではセンサーボードの 4 つの抵抗センサーポートに温度によって抵抗値が変化するサーミスタを繋げている。銅棒の端に熱源(アルコールランプ)を置き、熱源から徐々に離した 4 カ所にセンサーを置き、各センサー値を画面上で数値確認できるようにしたほか、銅棒に繋いだセンサーと同じように、画面上 4 カ所に四角形の温度表示スプライトが、温度が高くなるのに従い、青から黄緑、オレンジ、赤に変化する。

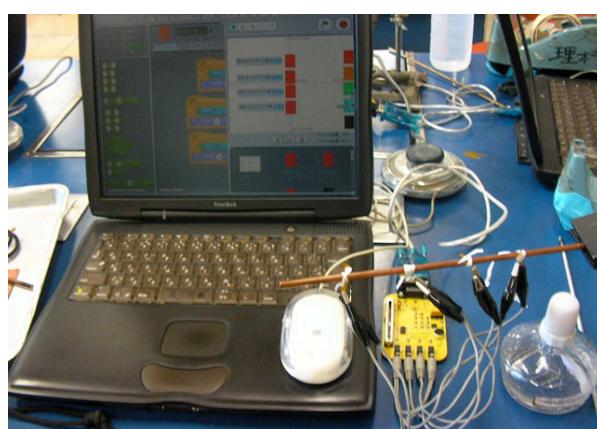


図 5-3-1 金属の温まり方を調べるツール

プログラムにはセンサーからの値が「～より大きく、かつ、～より小さい」などの条件分岐を使った。これにより、例えばセンサー値が 500 の場合はオレンジ、500 を超えると赤になるなど、値とあわせ色でも確認できる(図 5-3-2)。



図 5-3-2 条件分岐プログラム

5-3-2-2.小学 6 年理科:電気単元でのプログラミング活用

小学 6 年の発電・蓄電と電気の利用の単元での活用を試みた。一部の児童がより正確な発電時のハンドル回転数の計測を目的にツールとして Scratch を選択した。図 5-3-3 は児童らがつくった装置である。発電機の本体とハンドルに抵抗センサーを活用した接触スイッチを取り付け、接触により何回まわしたかをカウントする。

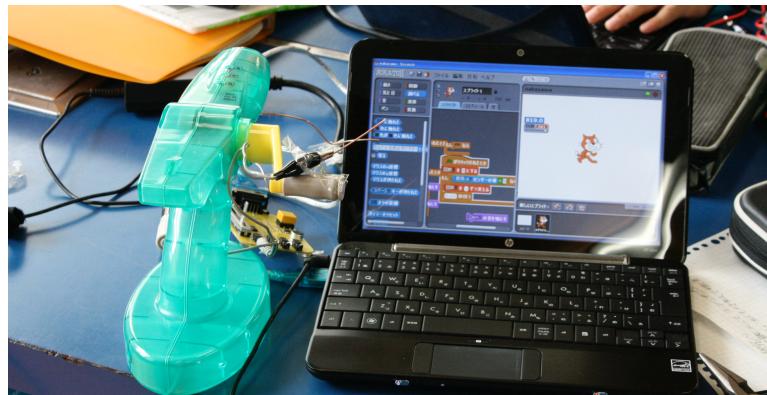


図 5-3-3 ハンドル回転数を調べるツール

プログラムでは、スイッチが接触したことを抵抗センサーの値で読み取り、その場合に回転数を示す変数を 1 ずつ増やす仕組みとした。また同速度でまわすために、メトロノームのようにリズム発生するプログラムを加えた。(図 5-3-4)



図 5-3-4 回転数の計測とリズムプログラム

5-3-3. 考察 例示の観点から

Resnick et al. (2009)によれば、指導者による助言の果たす役割は大きい。電気単元での活用では、指導者から下記のプログラムを児童への助言として例示した(図 5-3-5)。



図 5-3-5 例示したプログラム

児童が作成した図 5-3-4 のプログラムと比較すると、例示を参考にしながらもタイミングの調整や新しい機能の追加など工夫が多く見られる。さらに例示との差を詳細に見るため、電気単元で実験道具を製作した全 5 グループのプログラムについて確認した。例示されたプログラムと同様のコマンドは各グループで 2~4 個のみ使用されていた(表 5-3-1)。また例示と同様に利用されているコマンドの内容を見てみると「(変数)を 0 にする」、「(変数)を 1 ずつ変える」、「もし～なら、ずっと」と、今回の実験道具を作成する際には、重要な変数と条件分岐に関わるコマンドであったことが確認

できた。

表 5-3-1 例示プログラムとのコマンド比較

グループ	総コマンド数	例示同様	例示と異なる
A	20	3(15%)	17(85%)
B	18	4(22%)	14(78%)
C	18	3(17%)	15(83%)
D	17	3(18%)	14(82%)
E	12	2(17%)	10(83%)

5-3-4.本節のまとめ

本実践では理科の中でプログラミング学習を活かす実践を行った。今回の実践では、児童に対する例示がうまく機能し、多様なプログラムが作成された一方で、全グループが変数コマンドを用いるという偏りが生じたことも否めない。この点からは児童が例示されたプログラムの範囲から細かな変更はしているものの、大きな変更をしていなかったとも考えられる。児童が試したいこととプログラミングを含むデジタルものづくりの知識、さらには使えるツールおよび教科における学習目標とをいかに兼ね合わせて、授業デザインすることができるかが教科でのデジタルものづくり活用の大きな課題である。

第6章 ディジタルものづくりツール「プログラマブルバッテリー」の開発

6-1.いつでもどこでもデジタルものづくりを目指したプログラマブルバッテリーの開発

デジタルものづくり授業実践を行うための障壁の一つは準備である。そこで、いつでもどこでも誰でも、自分の興味の持った材料でデジタルものづくりを楽しめることを目的に「プログラマブルバッテリー」(以下 PB)の開発を行った。PB はコンピュータでプログラムをする必要がなく、誰でも簡単にモーターなどの出力デバイスを制御できるツールである(森 2014)。本章では PB とプログラミング教育の小学校導入のための授業実践について報告する。

6-2.プログラマブルバッテリーの機能

PB はモーターや LED、各種スイッチなどを接続することができ、PC を用いることなく、本体のボタン操作のみで、LED やモーターなどの出力デバイスのオンとオフのパターンを記録させることができるデバイスである(図 6-1)。具体的な操作は以下の通りである。①記録スイッチを押して、記録モードにする(本体の赤色 LED が点灯する)。②出力スイッチを押して、出力デバイスのオンとオフのパターンを記録する(例えば、1 秒オンー0.5 秒オフー0.5 秒オンー1 秒オフなど)。③記録スイッチを再度押すことで記録モードを停止する(赤色 LED が消灯する)。④再生ボタンを押すことで、記録したオンとオフのパターンを繰り返し再生する。また記録モード以外の状態では、出力スイッチを押すことで、パターンを記録すること無しに、接続された出力デバイスのオンとオフを切り替えることができる。



図 6-1 プログラマブルバッテリー

PB は DC 3V を出力する仕様となっており、これに対応するものであれば、LED やモーター、ブザーなどのパーツから、電池で動く玩具まで様々なものを接続し、出力のパターンを制御することが可能である。また本体上の出力スイッチの代わりに、任意のスイッチを接続して利用することもできる。PB の特長は、PC 無しで簡易なプログラミングができることである。これにより、どこでも(PC を持ち込めない場所でも、屋外でも)、誰でも(PC 操作に慣れていない児童や未就学児でも)、さらに時間を選ばずに短時間でも(いつでも)LED やモーターなどを活用したものづくりが可能となる。

改良版の PB(図 6-2)は最高 DC 9V まで出力できるようにすることで、様々なモーターでの利用を可能にした。また低学年児童にも使いやすいようボタンの大きさを変更している。



図 6-2 改良版プログラマブルバッテリー

6-3. プログラマブルバッテリーを活用した小学校低学年向け授業のデザイン

次に PB を活用した小学校 1 年および 2 年生を対象としたプログラミング授業例について紹介する。2017 年 2 月に東京都大田区立小学校で 2 校時を使って実施した本授業は、プログラミングの経験がない児童が無理なく取り組めるよう大きく 3 つのパートで構成した。

①体を動かす

児童自身がコンピュータになったつもりで、「右手をあげる」「手をたたく」などの紙に書かれた命令を順番に実行した(図 6-3)。

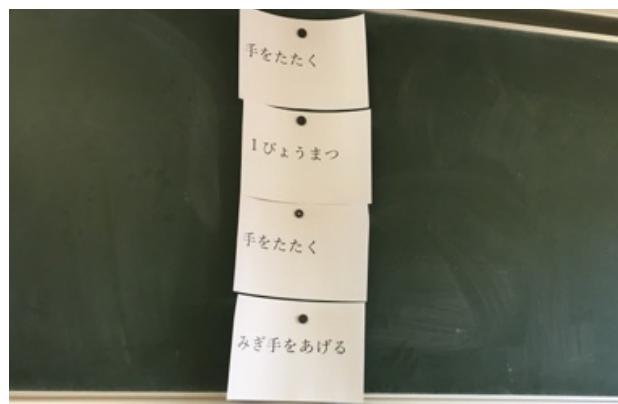


図 6-3 人間プログラミング

②ものを動かす

次に PB を使って、モーターの制御と LEGO®ブロックを組み合わせた作品づくりを行った(図 6-4, 6-5)。



図 6-4 PB を使った作品づくり①

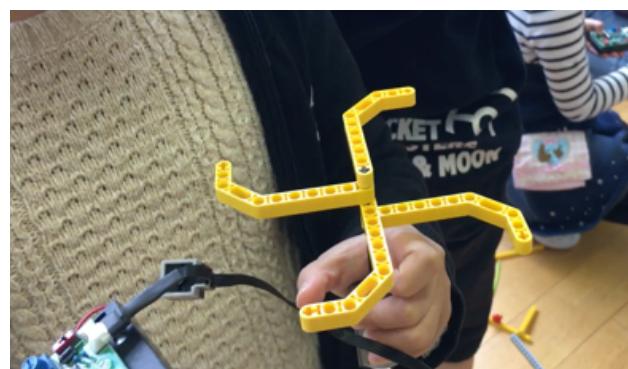


図 6-5 PB を使った作品づくり②

③コンピュータでのプログラミングを体験する

最後に Scratch を使い画面上のキャラクタを動かす簡単なプログラミングを行なった(図 6-6)。

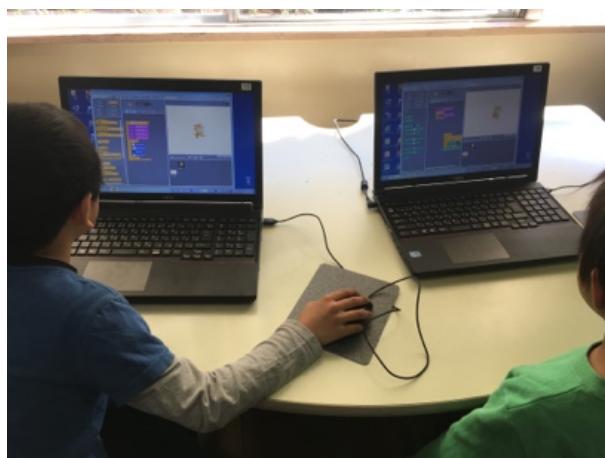


図 6-6 Scratch でのプログラミング体験

6-4.評価

授業後に PB の難易度について、5(むずかしい)から 1(かんたん)の 5 件法で質問した結果、平均 1.99(SD 1.25, N=70)となり、小学校低学年でも簡単に使用できることがわかった。また作品に対する満足度(5:まんぞくできた-1:まんぞくできなかった)は、平均 4.72(SD 0.70, N=70)と高い評価となった。児童のモーター制御記録時間は平均 21.49 秒(SD 19.10, N=67)とばらつきが生じた。

6-5.本章のまとめと今後の課題

PB を使って簡単に低学年でもコンピュータを埋め込んだものづくりをすることができた。また、もののづくりそのものを児童が楽しむことができた。Scratch を使ったプログラミング体験の事前に PB を用いることで、Scratch を用いたプログラミングに児童がよりスムーズに取り組むことができる様子も見受けられた。今後は PB で、よりデバッグがしやすいように、児童が記録したパターンを表示するなどの改良を行うほか、中高学年向けの授業デザインや PB だけを用いた授業デザインについても検討を行いたい。

第7章 文系大学生向けデジタルものづくり授業実践との比較

7-1.文系大学生向けディジタルものづくり授業のデザイン

本章では、Cricket と Scratch を用いて行った文系大学生向けのディジタルものづくり授業のデザインと授業実践結果について報告する。

1980 年代後半以降、家庭用 TV ゲーム機や携帯型ゲーム機は急速に普及した。2000 年代からは、端末の高性能化により携帯電話でも様々なディジタルゲームが遊べるようになった。まさに現代の大学生は生まれた時からディジタルゲームに触れてきた世代であるが、それらをつくる機会は小学校から高等学校を通じて、また大学でも特に文系学生にはほとんどない。中学校の技術・家庭科の技術分野で「プログラムによる計測・制御」の単元や、高等学校での教科「情報」でプログラミングを体験する機会を持つ学生もいるが、多くの学生がプログラミングを体験しないまま大学生となり、特に文系学生はそのままプログラミングを学ぶ機会がないまま卒業していくのが実状である。

小学校でのプログラミング教育の必修化以降、小学校から中学校、高等学校さらには大学まで、教養としてのプログラミングあるいはコンピュータサイエンスをいかに体系的に学習できるようにしていくかは、今後の課題である。本実践では小学生でも利用可能な Cricket や Scratch を用いて、ディジタルものづくりやプログラミングの経験がない大学生がそれらを体験することができる授業をデザインした。また本章の最後に、小学校における授業と文系学生を対象とした大学生向けの授業実践結果の比較を試みる。

7-2.Cricket を活用したディジタルものづくり授業のデザインと実践

本節では私立 A 大学の文系学生を対象とした情報科学科目「制御プログラミング」(2単位)として、2004 年度より 2013 年度まで実施した Cricket を用いたディジタルものづくり授業について報告する。学生は小学校での授業実践と同様に、4 日間の集中講義を通して、Cricket を用いた制御プログラミングの演習と応用課題として Cricket を活用した作品製作に取り組んだ(表 7-2-1)。

表 7-2-1 授業スケジュール

	時間	内容
1日目	①	イントロダクション
	②	プログラミング演習（モータ、ライト、サウンド）
	③	作品づくり(ロボットコンテスト)
2日目	④	プログラミング演習（センサ）
	⑤⑥	作品づくり(センサに反応するモノづくり)
	⑦	プログラミング演習（計算、変数、データ、コミュニケーション）
3日目	⑧	作品企画書づくり
	⑨⑩⑪	ファイナルプロジェクト製作
4日目	⑫⑬	ファイナルプロジェクト製作、プレゼンテーション準備
	⑭	作品プレゼンテーション
	⑮	まとめ

7-2-1.授業概要

授業は 1 日目を 3 時間とし、2 日目より 4 日目最終日までを 4 時間で実施した。受講者は 15 名を定員とし、授業運営は筆者と学生アシスタント 1 名で行った。教室はレイアウトが自由な多目的教室を使用し、受講者の作業スペースと作品製作で使用する各種素材やツールを置くスペースを設けた。また初日に実施するロボットコンテスト用のコースも教室内に設けた(図 7-2-1, 図 7-2-2)。



図 7-2-1 教室

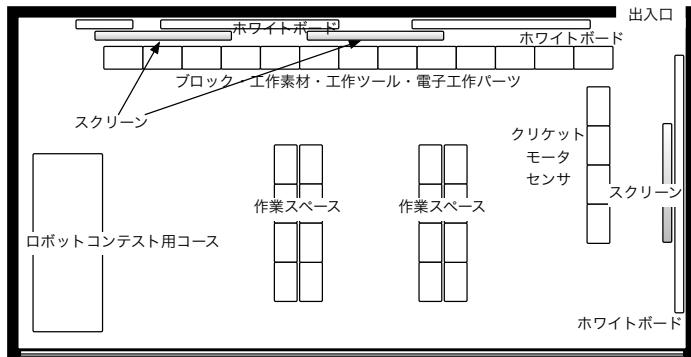


図 7-2-2 教室レイアウト

本授業で使用した機材、素材は表 7-2-2 の通りである。Cricket とノート型コンピュータのほか、作品製作用に様々な工作用素材や、動きの仕組み作りに活用できるギア等のブロックとあわせて、Cricket の抵抗センサーに接続できる可変抵抗器等の電子パーツ、はさみ等の工作ツールを用意し、教室内で自由に使えるようにした。

表 7-2-2 機材・素材一覧

クリケット	クリケット、モータ、モータボード、LEDライト、スピーカ、タッチスイッチ、音センサ、光センサ、抵抗センサ、ケーブル
工作素材	フェルト、綿、毛糸、ペットボトル、牛乳パック、空き箱、段ボール、発砲スチロール、輪ゴム、針金、紙皿、紙コップ、割り箸、ストロー、毛糸、結束バンド、風船、ビニールひも、洗濯ばさみ、麻ひも、つり糸、シール、軍手、クリップ、ゴムひも、羊毛、ばらん、おりがみ、色画用紙、アルミホイル、紙袋、ビニール袋 等
ブロック	ギア、カム、ブーリー、シャフト、コネクタ 等
電子パーツ	スイッチ、CDS、サーミスタ、可変抵抗器 等
工作ツール	はさみ、カッター、ペンチ、ビニールテープ、両面テープ、のり、ボンド、ホチキス、カラーマーカ 等
その他機材	ノート型コンピュータ、プロジェクタ

7-2-2.授業の流れ

次に 4 日間の授業の流れについて、順を追って説明する。

7-2-2-1.イントロダクション

最初に授業概要と 4 日間の流れを説明後、Cricket を使った作品例をビデオで紹介した。その際、Rusk et al. (2008) が指摘するように特定のサンプル作品のイメージだけに引きずられないよう、複数のサンプル作品を紹介している。

7-2-2-2.プログラミング演習(モーター, ライト, サウンド)

基本操作として Cricket とモーター, ライト, スピーカの接続と, コンピュータ上でのプログラミングから Cricket へのダウンロード, プログラム実行までの手順を説明し, その後出力に関する制御プログラミングの演習として, モーター, ライト, サウンドの制御に取り組んだ。プログラミング環境は日本語化されたものではなく, 英語版を使用した。モーターを制御する「motor onfor」「motor on」「motor off」, ライトを制御する「setlight color」「light off」, サウンドを制御する「play sound」, および繰り返しの制御に関わる「forever」「repeat」, タイミングを制御する「wait」の基本コマンドについて解説し, 受講者は表 7-2-3 の課題 1 から 4 に取り組んだ。さらに, これらの課題終了後に, 4 つの入出力ポートを別々に制御するポート指定の方法について説明を加え, 残りの課題に取り組んだ。課題 10 は早く進んだ受講者の応用課題として, モーター・ライト・サウンドを組み合わせた制御を課題としている。

表 7-2-3 プログラム練習問題(モーター, ライト, サウンド編)

No.	課題	No.	課題
1	ライトを3秒間緑色に点灯させる。その後、1秒間黄色に点灯し、さらに3秒間赤く点灯する。	6	ポート①と②のライトが同時にピンク色で点灯する。その後、ポート①のライトは、2秒後に消え、さらにその1秒後から2秒間黄色く点灯する。ポート②のライトは、1秒後に赤くなり、その後4秒後に消灯する。
2	ライトを3秒間緑色に点灯させる。その後、少しの間ライトを点滅させる。(緑色で)さらにその後、赤く点灯する。	7	色々な音を鳴らしてみよう
3	モータを3秒動かす。その後、反対向きに3秒動かす。	8	メロディをつくってみよう
4	モータを低速で3秒動かす。その後、高速で3秒動かす。	9	リズムをつくってみよう
5	ポート①のライトが点灯する。その後3秒後にポート②のライトが点灯する。(何色でも)	10	音の鳴る信号機をつくってください

7-2-2-3.ロボットコンテスト

次に, 出力に関するプログラム制御を活用する作品製作として, 車型ロボットづくりに取り組んだ。受講者は 2 名 1 組で, 図 7-2-3 のスタート地点から 2 つのゲートをくぐり, ゴール地点までコース上に設置されたボールを運ぶロボットを製作する(図 7-2-4)。グループ分けにも Cricket を用い, センサーと赤外線コミュニケーション通信を活用したパートナー探しゲームを行い, 以降で紹介する機能についても簡単に紹介した(図 7-2-5)。

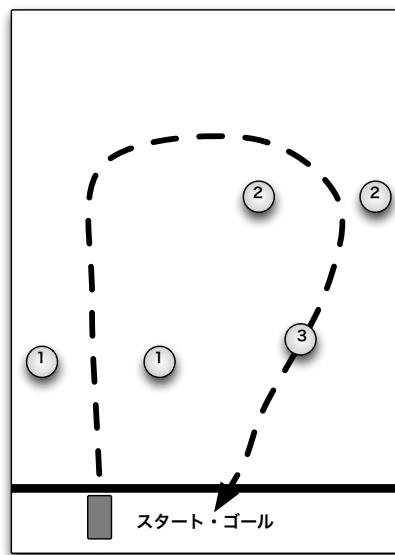


図 7-2-3 コースレイアウト



図 7-2-4 テスト走行

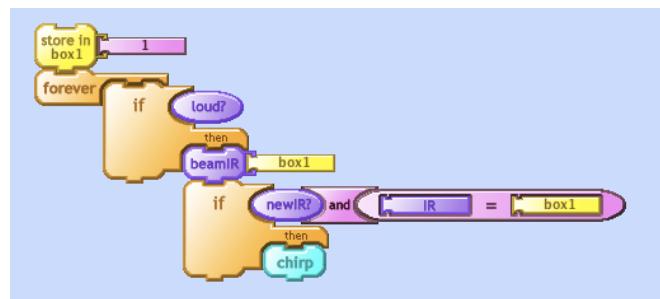


図 7-2-5 グループ分けプログラム

図 7-2-6 は受講者が製作した車型ロボットである。2 つのモーターとタイヤにキャスターを組み合わせ、三輪車型ロボットを製作した。ボールを運びやすいよう前方の機構を工夫している。図 7-2-7 は、2 つのモーターに直結する後輪と前輪をベルトで繋いだ 4 輪車型ロボットである。プログラムは、コース課題をクリアできるよう、Cricket に接続した 2 つのモーターに対して、個別に駆動時間やパワー、回転方向を細かく制御している(図 7-2-8, 図 7-2-9)。

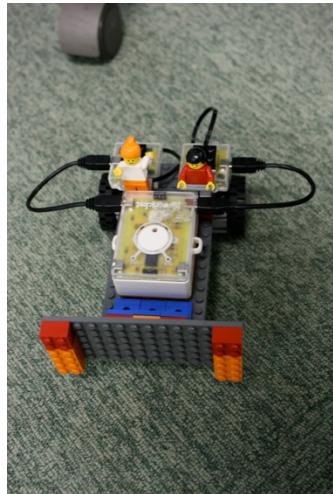


図 7-2-6 三輪車型ロボット

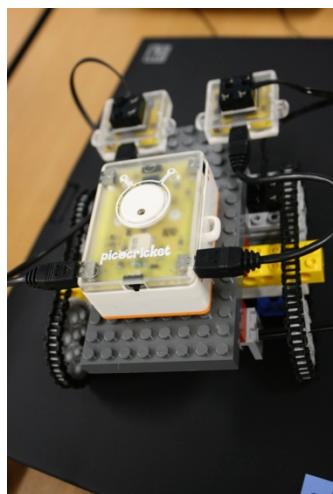


図 7-2-7 四輪車型ロボット

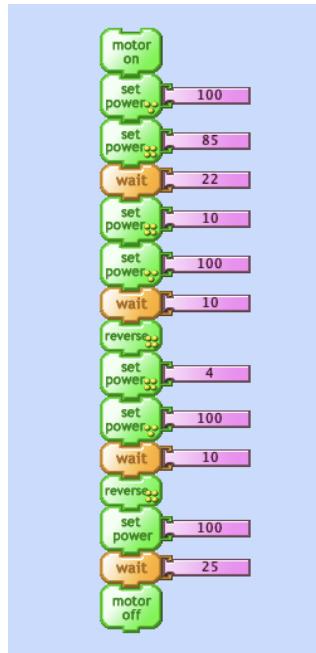


図 7-2-8 プログラム

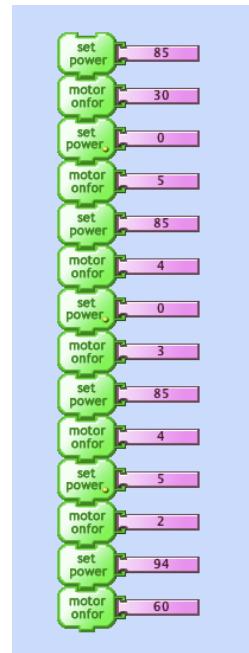


図 7-2-9 プログラム

7-2-2-4.プログラミング演習(センサー)

第2日目にはセンサーとスイッチの制御を含む入出力のあるプログラム作成に取り組んだ。条件分岐のための「if then」「if then else」コマンドについての説明後、表7-2-4の練習問題を行った。問題1から3は、「touch?」「loud?」「dark?」の判別条件が予め設定されたコマンドを利用する課題で、問題4から6では、センサーからの値を利用し、「=」「>」「<」「and」「or」を使った論理式を用いる課題としている。最後に応用課題として、実際のサンプル作品を制御しているプログラムをその動作から想像してプログラムを再現する。サンプル作品は、2つのスイッチとライトを用い、1つのスイッチを押すとライトが1つ点灯し、もう1つのスイッチを押すと別のライトだけが点灯する仕組みとした。図7-2-10は、サンプル作品を制御するプログラムである。

表7-2-4 プログラム練習問題(センサー編)

No.	課題
1	スイッチを押すと、モータが3秒動く。
2	うるさいと、犬が吠える。
3	暗くなるとライトが点灯する。明るくなるとライトが消える。
4	明るさの値が30よりも小さくなると、ライトが3秒間点灯する。
5	明るさの値が、20-30の時にライトが点灯する。それ以外は、ライトは消えている。
6	音センサーの値にあわせて、モータのスピードが変わる。
7	サンプルと同じプログラムをつくってみよう

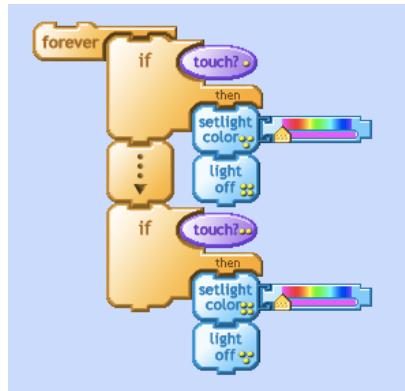


図7-2-10 サンプル作品のプログラム

7-2-2-5.センサーに反応するものづくり

次にセンサー・スイッチのプログラム制御の応用として、センサーに反応するものづくりをテーマに個人での作品製作を行った。音に反応してライトやモーター、サウンドが出力される作品(図7-2-11, 7-2-12)や、センサー値を利用したおみくじ(図7-2-13)などの作品がつくられた。

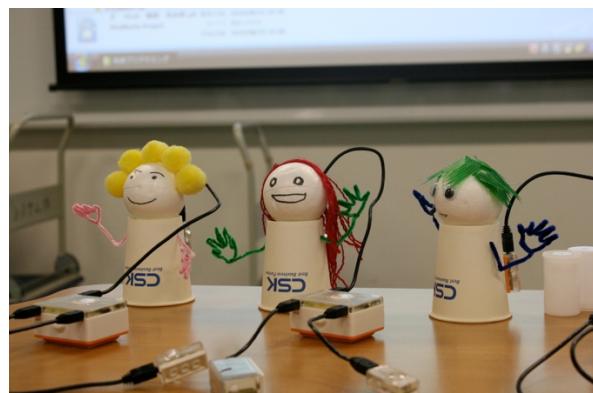


図 7-2-11 音に反応する人形

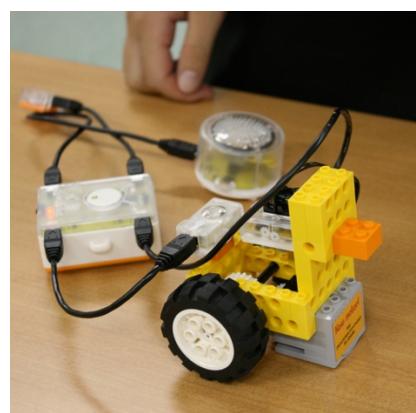


図 7-2-12 音に反応するアヒル

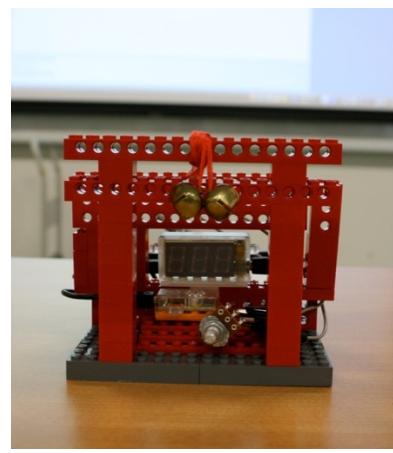


図 7-2-13 おみくじロボット

7-2-2-6. プログラミング演習(計算, 変数, データ, コミュニケーション)

3回目のプログラミング演習は、四則演算コマンドを用いた計算や変数の取り扱い、Cricketに記録可能な200の数値データの取り扱い、および内部タイマーの利用、赤外線通信コミュニケーションの制御について、表7-2-5の練習問題に取り組みながら学習した。また、その他の機能として、プログラムを手続きとしてまとめた「My Blocks」や乱数、スタックの停止方法については、作品製作のなかで隨時説明した。

表7-2-5 プログラム練習問題(計算, 変数, データ, コミュニケーション)

No.	課題	No.	課題
1	明るさセンサーの半分の値を表示する	6	0から999までタイマーを表示する
2	1から10まで順番にディスプレイに表示する	7	タイマーが500を超えるとブザーが鳴る
3	モータの速度が3秒ごとに10ずつ速くなる。(0から100まで変わる)	8	クリケット①のスイッチを押すと、クリケット②のライトが点灯する。
4	教室内の音の大きさの変化を30秒間記録して、グラフ化する。	9	クリケット①のセンサの値を、クリケット②のディスプレイに表示する。
5	上で収集したデータをモータの速度に当てはめてみる	10	クリケット同士が向き合うと、コオロギの鳴き声がする

7-2-2-7. 作品企画書づくり

車型ロボット製作、センサーに反応するものづくりの2回の作品製作に続いて、ファイナルプロジェクトとしての作品製作を行った。最終作品では、より深い作品製作に取り組めるよう、企画書作成からはじめる。企画に当たっては、①製作背景と目的として、製作するものが必要とされている状況、②作品概要、③作品スケッチ(作品が使われている状況などを含む)の3点をまとめた。企画書作成後にはプレゼンテーションを行い、他の受講者からアドバイスを受ける機会を設けた(図7-2-14)。



図7-2-14 企画プレゼンテーション

7-2-2-8. ファイナルプロジェクト製作・プレゼンテーション準備

5時間の授業時間を使って、ファイナルプロジェクトの製作と作品プレゼンテーションの準備を行った。予め用意した素材のほか、受講者も必要な素材を持ち込み、作品を完成させた(図7-2-15, 7-2-16)。



図 7-2-15 作品製作の様子 図 7-2-16 作品製作の様子

7-2-2-9.作品プレゼンテーション

最後に受講者は、自らの作品コンセプトの説明とともに、作品のデモと作品を制御するプログラムを他の受講者に向けて解説した。プレゼンテーション時間は1作品あたりおよそ10分間に設定し、発表の他、他受講者からの質疑も含めた(図7-2-17、図7-2-18)。

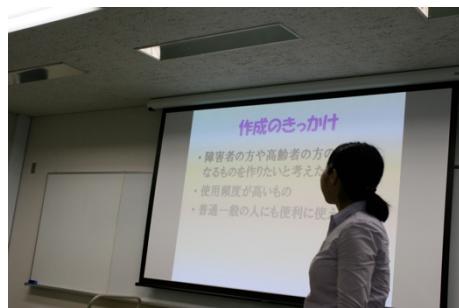


図 7-2-17 作品プレゼンテーション

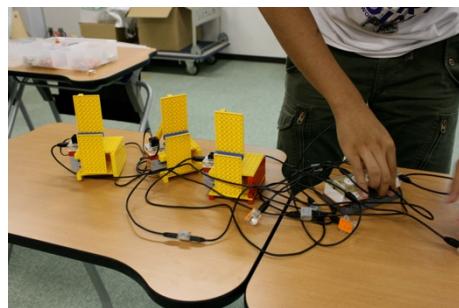


図 7-2-18 作品プレゼンテーション

7-2-3.結果と考察

7-2-3-1.受講者の作品

本授業で受講者がつくった作品例を紹介する。図7-2-19は明るさセンサーを活用したゲームである。指定された数値に近づけるようアイテムを選び、明るさを調整する。図7-2-20は点灯した色

に対応するボタンを押すゲームである。抵抗センサーに取りつけられたボリュームを回転させることで難易度が調節できる。図 7-2-21 は、図 7-2-20 の作品を制御するプログラムである。2 台の Cricket を使い、1 台を難易度設定に、もう 1 台をライト各色の表示とボタン入力の判定に用いている。

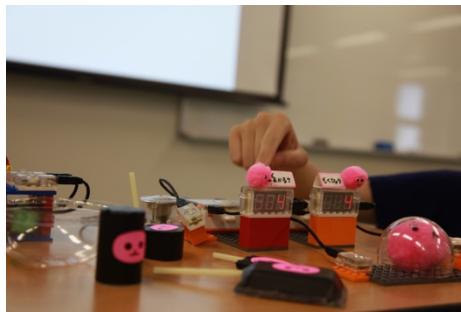


図 7-2-19 明るさゲーム

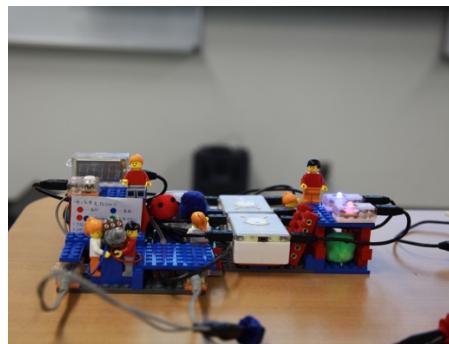


図 7-2-20 ボタン押しゲーム

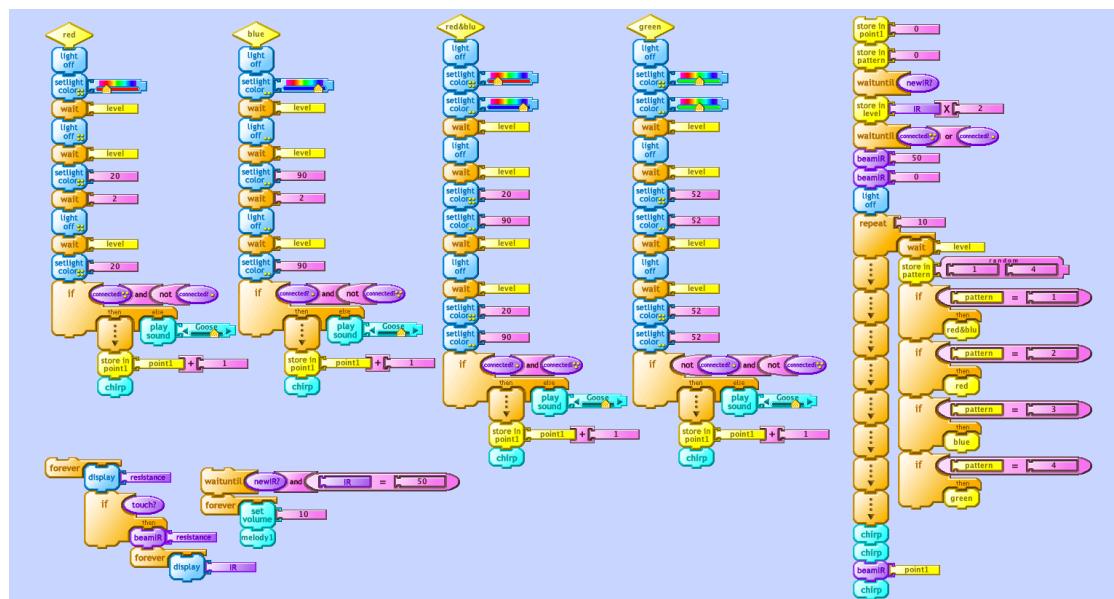


図 7-2-21 プログラム(ボタン押しゲーム)

7-2-3-2. プログラム

次に 2010 年度および 2011 年度受講者がファイナルプロジェクトとして製作した 15 作品について、プログラムコマンド別の利用数と利用率について確認する(表 7-2-6)。モータ、ライト、サウンドの制御に関わるコマンドは、作品の性質上使われていないコマンドが見られるものの、全作品で条件分岐に関わる「if then」もしくは「if then else」コマンドが利用されていた。また、変数の取り扱いに関わる「store in box」「box」も 15 作品中 13 作品で利用されている。赤外線コミュニケーション通信に関わる「newIR?」「beamIR」「IR」についても 3 分の 2 以上の作品で利用されており、条件分岐と変数の取り扱いとあわせて、ある程度の理解がなされたと考えられる。一方、データやタイマーに関するコマンドの利用は殆ど見られず、プログラミング演習での紹介方法などが課題である。

表 7-2-6 各コマンドの利用数と利用率(15 作品)

カテゴリ	コマンド	利用数	利用率	カテゴリ	コマンド	利用数	利用率
Lights	setlight color	6	40.0%	Data	clear	0	0.0%
	setlight power	3	20.0%		collect	0	0.0%
	light off	5	33.3%		rewind	0	0.0%
	display	10	66.7%		next data	0	0.0%
Sound	chirp	5	33.3%	Flow	wait	11	73.3%
	note	1	6.7%		waituntil	5	33.3%
	play sound	7	46.7%		forever	15	100.0%
	set inst	0	0.0%		repeat	3	20.0%
	set tempo	3	20.0%		if then	14	93.3%
	set volume	5	33.3%		if then else	11	73.3%
	melody	5	33.3%		stopall	0	0.0%
	rhythm	0	0.0%		stop stack	4	26.7%
Action	motor onfor	5	33.3%	Numbers	number	15	100.0%
	motor on	4	26.7%		+	8	53.3%
	motor off	3	20.0%		-	6	40.0%
	reverse	0	0.0%		X	5	33.3%
	this way	3	20.0%		/	2	13.3%
	that way	3	20.0%		random(min max)	3	20.0%
	set power	7	46.7%		>	6	40.0%
Sensors	touch?	8	53.3%		<	6	40.0%
	dark?	3	20.0%		=	12	80.0%
	brightness	4	26.7%		and	13	86.7%
	loud?	4	26.7%		or	3	20.0%
	loudness	5	33.3%		not	4	26.7%
	connected?	5	33.3%	My Blocks	my block	10	66.7%
	resistance	6	40.0%		store in box	13	86.7%
	beamIR	10	66.7%		box	13	86.7%
	IR	10	66.7%				
	newIR?	11	73.3%				
	timer	1	6.7%				
	reset timer	1	6.7%				

7-2-3-4. 受講者の感想

「最初は難しい印象を持ったが、触ってみると意外に簡単だった。」「1 日目はなかなか理解できなかつたが、2 日目から段々理解できて、どうすればどのような動きになるか分かつてきただ。」等、普

プログラミング演習と作品製作を通じて理解が進んだ様子がうかがえた。一方、「最初はすごく楽しかったが、2日目から大変さを思い知らされた。」「自分がイメージしたものを作成しようとすると難しい。」等、より複雑な作品製作のために壁にぶつかった様子もうかがえた。

7-2-4.本節のまとめ

本実践を通じて、タイマーとデータ機能を除いたほぼ全てのコマンドを活用してプログラミングを行い、Cricket を活用した作品づくりを文系学生でも行うことができた。さらにより複雑なプログラミングと作品製作を可能とするような授業デザインと受講者の支援方法について検討を進めたい。

7-3.Scratchを活用したデジタルものづくり授業のデザインと実践

次に本節では、Scratch を用いた文系大学生向けのデジタルものづくり授業実践について報告する。本実践はプログラミングの入門を目的とし、Scratch の開発コンセプトである初心者にとっての入りやすさと奥深さ、作品づくりの多様性に着目して、演習型授業をデザインし、実践した。授業は2009年12月にK女子大学の学生を対象に、3日間の集中講義として行われた。受講者は、いずれも文系学生75名(1年23名、2年33名、3年17名、4年2名)であった。事前のプログラミング経験については、回答があった67名中、「ある」が3名(4.5%)、「ない」が64名(95.5%)であり、ほとんどの学生は本授業で初めてプログラミングを体験した。最後に作品とプログラミングへの理解と興味に関するアンケートを元に評価を行った。

7-3-1.授業デザイン

授業は以下の流れで構成した。(図 7-3-1)

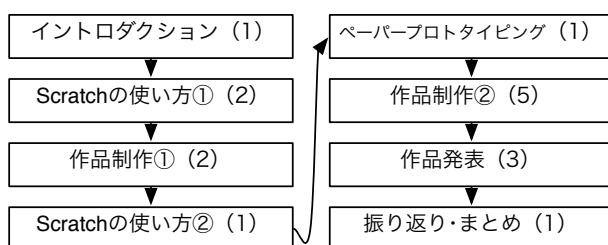


図 7-3-1 授業の流れ(括弧内は1コマ90分の授業時間数)

まずは、よりScratchの導入への敷居をより低くするために、イントロダクションの後、簡単なコマンドからはじめ、徐々に難しいコマンドへと移っていくプログラミング練習を行った。続いて、多様な作品づくりを目的にテーマを定めない1回目の作品制作、より複雑な2回目の作品制作のためのペーパープロトタイピング(企画書づくり)、作品制作(2回目)、作品発表の順で進めた。

I .イントロダクション

授業は集中講義3日間の内容とScratchを紹介することからはじめた。サンプルプログラムを見せ、Scratchで制作可能な作品のイメージ作りを行った。Rusk et al. (2008)が指摘するように、複数のサンプル作品を提示することにより、作品制作のイメージ拡大を狙った。

II .Scratch の使い方①

起動方法とプログラミングから実行までの基本的な使い方についての説明の後、各ブロックの機能説明と練習問題の取り組みを繰り返した(図 7-3-2)。

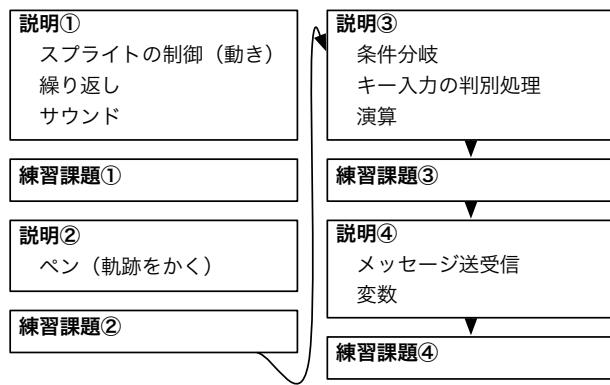


図 7-3-2 Scratch の使い方①の流れ

画面上のスプライトの動かし方(初期画面ではネコのスプライトが表示されている)についての説明の後、繰り返しとサウンドに関するブロックについて説明した。その後、次の 11 間の練習問題に取り組んだ。

- (1) ネコを 10 歩動かす。
その後ニヤーと鳴いて、さらに 10 歩進む。
- (2) ネコがニヤーと 10 回鳴く
- (3) ネコを 15 度ずつ右まわりに 1 回転させる。
その後左回りに 1 回転させる。
- (4) サンプルと同じプログラムをつくってみよう

4 番目の問題は、ネコが画面を左右に一定間隔に移動繰り返す様子を見て、実行されているプログラムを考える課題とした。

次にスプライトが移動する際に軌跡をかく「ペン」機能に関連するブロックを説明した。その後、スプライト制御、繰り返しを活用する練習問題として、図形描画課題に取り組んだ。

- (5)図形を描いてみよう
(正方形、正三角形、長方形、正六角形、円、星)
- (6)あなたの名前の一文字を書いてください

ここまででは Scratch に慣れることを重視し、画面上を動かすなど、出力のみのプログラミングに取り組んだ。続いて、キーボードからのキー入力の判別処理や条件分岐など入出力のあるプログラミングに取り組んだ。関連するブロックについて説明後、次の練習問題に取り組んだ。

- (7)スペースキーを押すとネコが 10 歩進む
- (8)ネコにマウスカーソルが触るとニヤーと鳴く。触れていないときはぐるぐるまわる。
- (9)もしマウスの X 座標が 170 か 200 の間なら、ネコが 10 歩動く。

最後に、スプライト間連携のためのメッセージ送受信と変数の取扱いを説明し、練習問題に取り組んだ。

- (10)(ネコを 2 匹用意する)ネコ 1 が 10 歩動いた後、ネコ 2 が 10 歩動く
- (11)ネコにマウスカーソルが触れた回数を数える

III.作品制作①

次に個人作品の制作に進んだ。制作テーマは特に設けず自由とした。1 回目は短めに設定し、2 回目を本格的な作品制作のため長く設定した。また制作後、他の受講者の作品を観る時間を設けた。

IV.Scratch の使い方②

2 回目の作品制作に向けて、明るさや音量、抵抗などの情報を取り込めるセンサーボードの使い方を説明した。また、サンプルプログラムをつくりながらセンサーボードの機能を試した。

V.ペーパープロトタイピング

2 回目の作品制作にむけて、作品をより深く練り込むための仕様書を作成した。ストーリーボード型のフォーマット用紙を用意し、画面遷移とできる限り詳しい機能について書き込んだ。

VI.作品制作②

2 回目の作品制作は、個人制作もしくは 2 名から 4 名 1 組のグループ制作とした。また作品制作①と同様に、テーマは特に設けなかった。制作終盤には、作品発表用にタイトルと作品解説を記載するシートを配り、制作作業についてまとめた。

VII.作品発表

各作品について、5 分間の発表を行った。作品の解説、デモストレーションの他、制作にまつわるエピソード等を発表した。また発表者以外の学生は、最も優れた作品への投票をするとともに、作

品の感想をコメントシートに記入した。また、授業後には他受講者からの感想を閲覧できるようにした。

VIII.振り返りとまとめ

2回目の作品制作での担当作業についてまとめ(グループ作品の場合), 3日間の授業の感想をまとめた。

7-3-2.結果と考察

7-3-2-1.受講者の作品

作品制作①は、個人制作であったため 75 作品が完成した。様々なストーリーのアニメーションが 52 作品 (69%) , マウスクリック等で音や絵が変化し、ストーリーが展開するインタラクティブアニメーションが 23 作品(31%)となった。また作品制作②では全 35 作品中、個人作品が 10 作品、グループ作品が 25 作品となった。作品はゲームが 13 作品、アニメーションが 12 作品、インタラクティブアニメーションが 6 作品、楽器やミュージックビデオなど音楽に関連する作品が 4 作品となり、作品の種類も内容もさらに多岐に渡り、プログラムもより複雑なものとなった。各作品により重視する点も異なり、スプライトの描画にこだわった作品もあれば、音楽との連携やゲームの難易度、あるいはストーリー展開など、各作品で特徴が異なった。

7-3-2-2.受講者アンケートの結果

I .Scratch プログラミングの理解

プログラミングの理解を確認するため、各コマンドブロックの使い方について、「5:よく分かる」から「1:まったく分からぬ」の 5 段階で評価してもらった。項目は Q1:スプライト制御, Q2:座標, Q3:繰り返し, Q4:キー入力の判別処理, Q5:条件分岐, Q6:メッセージ送受信, Q7:変数について質問した(表 7-3-1)。

表 7-3-1 コマンドブロック理解度

項目		M	SD
Q1	～歩動かす、～度回す	4.42	0.77
Q2	X座標を～にする、Y座標を～にする	4.13	1.09
Q3	ずっと、～回くりかえす	4.34	0.76
Q4	～キーが押されたとき	4.09	1.02
Q5	もし～なら	3.77	0.97
Q6	～を送る、～を受け取ったとき	3.38	1.30
Q7	変数	2.42	1.18

スプライト制御と繰り返しの理解度が高く、次いで座標、キー入力、条件分岐、メッセージ送受信、変数の順という結果となった。特に変数については理解度が低く、今後カリキュラムの修正が必要と考えられる。

II.作品制作・プログラミング・作品制作への興味

プログラミングへの興味について、他のコンピュータを使った作品制作への興味とあわせて5段階(5:とても興味がある～1:まったく興味がない)で質問した。表2は授業前後の平均と標準偏差を示したものである。プログラミングと作品制作をテーマとした授業であるため、授業前の興味も高いが、授業後にはさらに向上が見られた。また、*t*検定の結果からも有意傾向が見られた(プログラミング/両側検定 $t(129)=1.731$, $0.1 < p < .05$, 作品制作/両側検定: $t(129)=1.867$, $0.1 < p < .05$)。

表 7-3-2 プログラミング、作品制作への興味

項目	事前(n=67)		事後(n=64)	
	M	SD	M	SD
プログラミング	3.87	0.92	4.13	3.87
コンピュータを使った作品制作	4.09	0.87	4.36	4.09

III.感想記述

「最初は難しい印象を持ったが、Scratchを触ってみると意外に簡単だった。」や「プログラムを組むと聞いて興味はあったが難しそうで不安だったが、実際は思ったより分かりやすかった。」等、Scratchの入りやすさについての感想が見られた。また、「1日目はなかなか理解できなかつたが、2日目から段々理解できて、どうすればどのような動きになるか分かつてきただ。」等、Scratchを使いながら、1日目の練習課題と作品制作を通じて、プログラミングへの理解が進んだ様子がうかがえた。一方、「最初はすごく楽しかったが、2日目から大変さを思い知らされた。」「自分がイメージしたものを作ると難しい。」等、より複雑な作品制作のためのプログラミングの技術的な壁にぶつかった様子もうかがえた。また、「みんなの作品を見て、1つのソフトでもこんなに沢山のものができるんだなと想いました。」や「アニメやゲームなど表現の幅がとても広がって色々な欲が出てきた」等、多様な作品づくりへの可能性についても受講者自身が気がついた様子がうかがえる。

IV.コンピュータ・インタラクティブメディアの理解

プログラミングの理解、興味とあわせてプログラミング活動を通じたコンピュータの仕組みとディジタルゲームなどのインタラクティブメディアの仕組みの理解について、5段階(5:よく分かる～1:まったく分からない)で質問した。表7-3-3は授業前後の平均と標準偏差を示したものである。*t*検定の結果、両条件の平均の差はいずれも有意であった(コンピュータの仕組み/両側検定: $t(127)=3.730$, $p < .01$, インタラクティブメディアの仕組み/両側検定: $t(125)=4.472$, $p < .01$)。

ずれも授業前の平均が低いが、この結果からも授業後には若干ではあるが理解度の向上も確認できた。

表 7-3-3 コンピュータ・インタラクティブメディアの理解度

項目	事前(n=67)		事後(n=64)	
	M	SD	M	SD
コンピュータの仕組み	2.60	1.10	3.27	0.95 **
インタラクティブメディアの仕組み	2.31	1.10	3.09	0.89 **

** $p < .01$ 水準

7-3-3.本節のまとめ

2回の作品づくりと段階的なプログラミング練習を通じて、スプライト制御、座標、繰り返し、条件分岐などのプログラミングに必要な知識を理解することができた。また2回の作品づくりを通じてより複雑で多様な作品がつくられた。アンケート結果からは、プログラミングの活動を通じて、コンピュータを使った作品制作やプログラミングへの興味やコンピュータやインタラクティブメディアの仕組みへの理解を高めることができたことも確認できた。今後は変数の取り扱い等、さらに複雑なプログラミングを可能とするような練習課題やさらに多様な作品づくりを可能とする授業デザイン、受講者支援の方法について検討を進めたい。

7-4. 小学校におけるデジタルものづくり授業実践との比較

最後に小学校における Cricket と Scratch を活用した授業実践との比較を行いたい。本章での文系大学生向け授業は、いずれも小学校でのデジタルものづくり授業と同じように「導入」「プログラミング演習(ツールの使い方に慣れる)」「作品づくり」「発表」「振り返り」の流れで実施した。またツール以外に使用する材料も、LEGO®ブロックや工作材料など、小学校授業と同じ材料を用意した。授業時間に関しても、90 分 × 15 回と第 4 章と第 5 章で紹介した小学校 4 年生を対象とした授業実践とほぼ同じ長さであった。

まずプログラミングやコンピュータを使ったものづくりへの興味関心は、小学生と大学生ともに、授業後に向上が見られる結果となった。表 7-4-1 は Scratch を用いたデジタルものづくり授業後の小学 4 年生(N=36)と大学生(N=64)に、プログラミングに関する理解度の確認のため Scratch のコマンドに関する理解度(「5:よく分かる」から「1:まったく分からない」の 5 段階)を質問した結果である。自己評価のため単純に比較することはできないが、小学生も大学生と同様に Scratch を使ったプログラミングについて授業後に理解できたと感じたことが分かる。

表 7-4-1 Scratch 理解度アンケート比較

項目	小学生		大学生	
	M	SD	M	SD
～歩動かす、～度回す	4.97	0.17	4.42	0.77
X 座標を～にする、Y 座標を～にする	3.97	1.19	4.13	1.09
ずっと、～回くりかえす	4.94	0.24	4.34	0.76
～キーが押されたとき	4.82	0.72	4.09	1.02
もし～なら	4.32	1.09	3.77	0.97
～を送る、～を受け取ったとき	3.65	1.57	3.38	1.30

授業を通じて製作された小学生と大学生の作品を比較することも難しい。大学生の授業ではプログラミングの練習課題として変数まで取り扱ったが、小学生の授業では児童全員が取り組む課題としては取り扱わず、必要な児童だけへの説明に留めた。これは筆者が一部の児童へ変数の説明をした際に児童全員が理解することは難しいと判断したためであるが、変数の取り扱いにより大学生の作品はアイデアも拡がり、より複雑なものとなっている。またプログラミングに関する知識とあわせて、ものづくりの経験の違いが小学生と大学生との作品のアイデアに違いをもたらしていたとも考えられる。今後は年齢や経験、既有知識の差がデジタルものづくりに与える影響について明らかにし、小学校においても学年差を考慮したデジタルものづくり授業のデザインに役立てていこう。

第8章 まとめ

8-1.初等教育におけるデジタルものづくり授業デザインへの提言

本研究での授業デザインと授業実践から、初等教育において児童の多様な学びを育むデジタルものづくり授業デザインのために重要な要素を表 8-1-1 にまとめる。デジタルものづくりを扱う授業では、伝統的な教授中心の授業とは異なる授業デザインが必要である。通常の教科学習ではカリキュラムに従い、教師が予め計画した通りに遂行できるよう授業がデザインされる。予め設定された時間内に、決められた学習内容が行われるよう、学習目標と教授学習方法について教師が設定し、綿密に指導計画が立てられる。一方、本論文で紹介してきたデジタルものづくり授業では、児童が自らのものづくり活動を決めるることを通じて、児童の学習はデザインされる。学習の内容や方法、目標、時間についても児童が決めなくてはならない。また評価についても児童が自己評価できることが求められる。そのため児童が自らの活動を決定しつつ、それに関わり必要な知識を習得し、限られた時間で活動に区切りがつけられるような授業デザインが必要である。

デジタルものづくり授業の運営における教師の主な役割は、児童の活動支援者となることである。全ての活動とそれに伴う学習を児童に任せるのでなく、児童がものづくりに行き詰った際や、児童のものづくりがさらに発展する可能性がある場合には、児童が必要とする助言を必要なタイミングで提供する必要がある。また授業で用いる機材や材料、教室の選定などの学習環境デザインもデジタルものづくり授業で教師が担う大きな役割である。

表 8-1-1 デジタルものづくり授業と伝統的授業の違い

	伝統的授業	デジタルものづくり授業
教師の主役割	知識伝達	活動支援
指導計画	教師が予め作成	教師が予め作成、児童の活動状況により随時変更
学習内容	一様:カリキュラムによって設定	多様:教師が設定するものを元に児童が決定
学習目標	教師が設定	教師が設定するものと児童が設定するものが存在
学習時間	固定的:カリキュラム配当時間で決定	個別・流動的:児童の活動によって決定
学習方法	講義中心:教師が予め設定	活動中心:教師の助言のもとに児童が決定
評価方法	絶対評価(テスト等), 觀察評価	絶対評価(作品), 觀察評価と児童による自己評価

図 8-1-1は従来の授業とデジタルものづくり授業との授業デザインアプローチを比較したものである。多くの授業では同一の学習目標に向かって児童全員が到達できるように授業をデザインする。一方で本論文で取り扱ったデジタルものづくり授業では、児童全員が到達すべき学習目標としてプログラミングやツールの使い方の習得があるが、その先のデジタルものづくり活動では個々の児童によって学習目標が異なる。

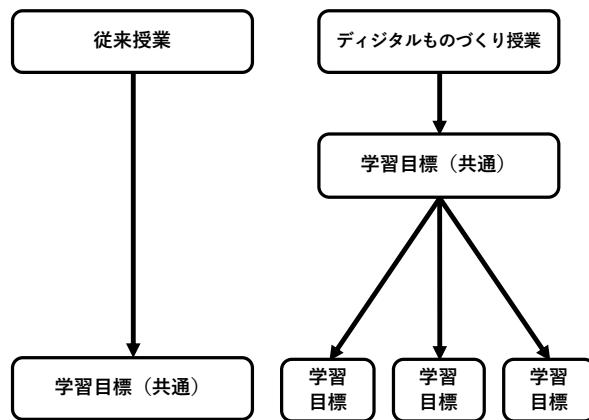


図 8-1-1 従来授業とディジタルものづくり授業のデザイン比較

さらに詳しいディジタルものづくり授業のデザインと運営方法について図 8-1-2 に示す。先に述べたようにディジタルものづくりで用いるツールの使い方(プログラミングを含む)に関しては、全ての児童が学ぶことができるよう実施する。以降、児童がものづくりを行う場面では、各々の児童が、1)活動の目標を立て、2)ものづくりし、3)授業中の発表の機会や非公式の情報共有の機会を通じて、他の児童からの新しいアイデアを得て、4)現在のものづくりの課題や問題について考えるために活動の振り返りを行うとともに、次の活動の目標を考える。教師は授業内で行う項目だけは決めておくものの、実際に行う内容については児童に委ねる。

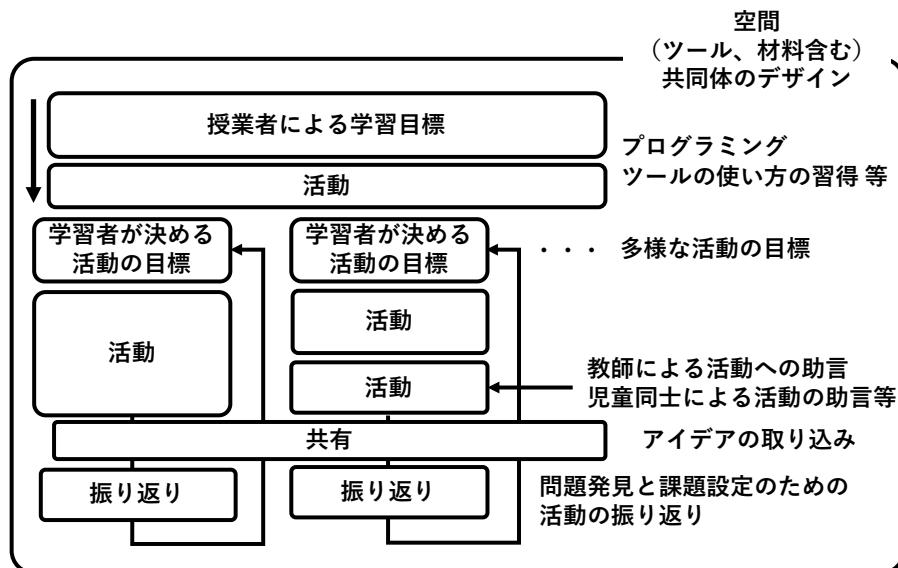


図 8-1-2 デジタルものづくりの授業デザイン構造

このように授業者によって設定された学習目標のあとに、児童が個々の学習目標が設定できるよう2段階で構造化する授業デザインが、児童の自律的で多様な学びを生み出すための小学校におけるデジタルものづくり教育には必要であることを本研究の結論として提言する。

8-2. 考察～初等教育におけるデジタルものづくり教育の可能性～

本節では、本研究で取り扱ったデジタルものづくり教育について、多角的観点から初等教育での実施における可能性を考察する。

8-2-1. ノンフォーマル学習としてのデジタルものづくり教育

まずデジタルものづくり教育をフォーマル学習とインフォーマル学習の観点から検討したい。フォーマル学習は「組織化され、構造化された環境において発生し、明らかに（目標設定、時間、リソースの観点から）学習としてデザインされている学習」（OECD 2011）とされている。この定義によれば通常の教科学習など小学校で行われている授業はフォーマル学習にあたる。インフォーマル学習は「仕事、家庭生活、余暇に関連した日常の活動の結果としての学習」（OECD 2011）とされており、児童を例にとれば学校外での遊びや日常生活のなかでの学びがあてはまる。さらにOECD(2011)では、フォーマル学習とインフォーマル学習の間にあたるノンフォーマル学習を定義している。ノンフォーマル学習では「学習（学習目標、学習時間、もしくは学習支援の観点から）としては明確にデザインされていないが、計画された活動に埋め込まれた学習」であるとしている。山内ら(2013)は、近年、新しい学びの場として子ども向けに科学館や博物館など社会教育関連施設で実施されることが多くなったワークショップはノンフォーマル学習にあたるとしている。本研究で行ってきたデジタルものづくり教育は、学校教育の一環として授業時間は計画されているが、児童の学習そのものは完全にデザインされたものではなく、児童が個々にものづくりを通じて目標を設定していく必要がある。その点では、デジタルものづくりは学校教育というフォーマル学習のなかで、ノンフォーマル学習を経験する機会になっているといえる。

8-2-2. カリキュラムのなかでのデジタルものづくり教育

次に学校カリキュラムのなかでのデジタルものづくり教育について考えたい。本研究での主たる授業実践は奈良女子大学附属小学校で実施した。同校は大正自由教育運動のなかではじまった児童中心主義教育のカリキュラムを現在まで続けている学校である。児童の学びに重きが置かれるようになった現在と違い、教授が教育の中心であった時代に、教育を学習と捉える考え方は当時、「発想の転換無しには理解できない主張であった」（奈良女子大学文学部附属小学校 1998）。当時の同校主事であった木下竹次による学習原論には、同校の目指す教育としての自律的学習者の育成について記されている。そのなかで「学習とは、学習者自らが教師指導の下にある整理された環境の中にあって、自ら機会を求め、自ら刺激を与え、自ら目的と方法を定め、社会的自我の向上と社会文化の創造を図っていく作用である」（木下 1923）と示している。また同校では、教科の枠組みを超えた合科学習を大正期から続けており、学習法としても独自学習、相互学習、第二次独自学習と繋がる学習の流れを提唱し、実践を続けている（奈良女子大学附属小学校 2015）。学

習を児童個人の独自学習からはじめ、相互学習で他の児童とともに学ぶことで学習を相互に補い、さらに独自学習に戻り、自らの学びを深める学習の流れを常に授業に取り入れている。本研究での授業実践で行ったデジタルものづくりのなかで、児童が自ら学習の目的と方法を考え、さらに教科を超えた広い視点で、他の児童とともにお互いに影響をしながら学ぶことは、長年に渡る児童の学びを重んじてきた同校のカリキュラムのなかでこそ、無理なく実施できたともいえる。しかしながら、奈良女子大学附属小学校のように児童中心主義に基づくカリキュラムに基づき運営されている小学校は限られる。現状の小学校カリキュラムでは、本研究で実践してきたようなデジタルものづくりは総合的な学習の時間での実施が最適である。小学校における総合的な学習の時間は「横断的・総合的な学習や探究的な学習を通して、自ら課題を見付け、自ら学び、自ら考え、主体的に判断し、よりよく問題を解決する資質や能力を育成するとともに、学び方やものの考え方を身に付ける、問題の解決や探究活動に主体的、創造的、協同的に取り組む態度を育て、自己の生き方を考えることができるようとする」(文部科学省 2008)ことを目指している。しかしカリキュラム開発は多忙をきわめる学校と教師に委ねられ、結果的に本来の目的とはかけ離れ单なる活動のための時間として終わってしまうことも少なくない。最も困難であると考えられるカリキュラム開発について、デジタルものづくりに基盤を置けば、授業で用いる教材や授業の内容など、大枠を決定することができる。その上で、教師が児童にあわせてカリキュラムや個々の授業を開発できる面においても、デジタルものづくり教育には期待できる。総合的な学習の時間で目指す自律的で能動的な学習者の育成にデジタルものづくりが活用できると考える。

8-2-3.成人学習の導入としてのデジタルものづくり教育

生涯学習、成人学習におけるアンドラゴジーと自己調整学習は、初等教育におけるデジタルものづくりの可能性を別の角度から位置付けるキーワードである。生涯学習社会と言われる現代に対応するため、学校教育においても、総合的な学習の時間の設置など児童生徒の生涯学習への移行を目指した施策がなされてきた(加澤 2004)。デジタルものづくり教育における自律的、能動的学習者の育成は、生涯学習社会のための成人学習理論としてのアンドラゴジーにも通じる。Knowles(2013)はアンドラゴジーにおける成人学習のモデルとして、1) 知る必要性、2) 学習者の自己概念、3) 学習者の経験の役割、4) 学習へのレディネス、5) 学習への方向づけ、6) 動機付けを挙げている。また成人教育者の役割として、学校教育において教えられることに慣れてきた「依存的学習者から自己調整学習者への移行」を支援することであるとしている。自己調整学習は「自らの学習を動機づけ、維持し、効果的に行うプロセス」(自己調整学習研究会 2012)である。近年、特に大学教育において成人学習への移行のため、自己調整学習者を育てる試みが盛んに行われている(Nilson 2017)。デジタルものづくりは、児童にとって身近なコンピュータで動くものづくりを題材とすることで、学習への動機付けを容易にし、ものづくりを通じてその維持と効果的に実施

するための過程についても学習する機会にもなり得る。デジタルものづくりは成人教育を見据えた小学校段階での教育方法ともなり得ると考えられる。

8-2-4. デジタルものづくり教育と学校教育のコンフリクト

デジタルものづくり教育の可能性として、あらためて学校教育のなかでデジタルものづくりを実施する難しさについても触れたい。プログラミング教育を含むデジタルものづくり教育を小学校で実施する場合に注意すべきことは、本研究でも使用してきた Cricket や Scratch をはじめ MIT メディアラボで開発されてきた一連のツールは、Lifelong Kindergarten をコンセプトに開発されていることである。すなわち、従来の教授主義の学校教育とは対極的に、学習者による自主的な学びのために設計されている。幼稚園で子どもたちが自分の興味にあわせて表現や創作活動をしながら、協働的に生じる学びをテクノロジを用いることで生涯続けることが、Lifelong Kindergarten 研究グループのテーマである。Cricket や Scratch には、子どもたちが手を動かし、Tinkering しながら(いじくりながら)試したくなるような設計がされている。例えば、プログラミング環境のなかでブロック型につくられたコマンドは、コマンドのカテゴリ毎に机の引き出しにしまわれているように配置されている。またコマンドブロックの形状も機能ごとに異なっており、様々なコマンドブロックを出して、繋げて、試してみたくなるように設計されている。しかし Cricket や Scratch を使って教授中心の授業を行おうとすると、自ら手を動かしていじりながら試してみたいという児童を制止しながら、授業進行をしなくてはならなくなる。プログラミング活動に限らずデジタルものづくりを進めていく場合に、ぶつかることが多い授業進行への壁である。筆者も授業実践のなかで度々このような状況に出会った。あまりにも児童を制止する機会が多くなりすぎると、児童を教師の命令通りに動くようにプログラミングしているような錯覚に陥ることもある。このようなツールの持つ特性を教師が理解した上で、授業をデザインし、運営をしていくこともデジタルものづくり教育には必要となる。

1980 年代の初期のプログラミング教育がうまくいかなかった原因の一つは、プログラムによる図形描画に活動を限定してしまったことがある。Scratch など、近年のツールは児童の多様な活動を可能にするよう設計されている。しかし多様な活動が可能なゆえに、授業進行のために児童の活動を制限してしまえば、2020 年度から必修化される小学校でのプログラミング教育も、初期のプログラミング教育と同様の結果を招きかねない。

児童がデジタルものづくりに楽しそうに取り組む姿を見て、単に遊んでいるように見えるという教師の声を聞くことがある。確かに本研究で使用したツールや授業のための用意した学習環境は、児童が楽しみながらものづくりに取り組めるようデザインされているが、それは児童の動機付けのために他ならない。その動機を維持し、ものづくりを介した学習を続けていくための目標の設定と、その達成に向けて振り返りを含めた活動の循環を促すための教師による支援と授業デザインが必要である。

8-2-5.多様な学びを育むディジタルものづくり教育導入の意義

ディジタルものづくりはプログラミングなどスキル習得に限定すれば、教授中心の授業デザインと授業運営でも実施可能である。しかし、それでは本研究で明らかにしてきた児童の多様な学びを育む可能性を持つディジタルものづくり教育を十分に活用できていない。本節であらためて多角的にディジタルものづくりを見てきたように、ディジタルものづくり教育はフォーマル学習としての従来の学校教育とは異なり、成人群にも通じる自律的学習者を育む教育となり得る。児童中心主義の教育観のもと学校カリキュラムの構築と運営がされている一部の小学校では、多様な学びを育むディジタルものづくり教育をそのまま活用できるが、多くの学校と教師にとっては教授中心の授業と異なるディジタルものづくりを授業に導入することは大きな挑戦である。しかし、もう一度総合的な学習の時間の導入目的である「横断的・総合的な学習や探究的な学習を通して、自ら課題を見付け、自ら学び、自ら考え、主体的に判断し、よりよく問題を解決する資質や能力を育成するとともに、学び方やものの考え方を身に付け、問題の解決や探究活動に主体的、創造的、協同的に取り組む態度を育て、自己の生き方を考えることができるようとする」(文部科学省 2008)に立ち返ってみれば、ディジタルものづくり教育は総合的な学習の時間が目指す本来の目的を達成するための手段になるともいえる。

8-3.結論

本研究ではデジタルものづくり授業をデザインし、2002 年より延べ 700 時間の小学校でのデジタルものづくり授業実践を行い、児童の学びの評価を行った。小学 6 年生はブロック型コマンドによる組み合わせを用いるプログラミング環境を使うことで、4 時間の授業を通じて、プログラミングの基本的な概念となる順次、繰り返し、条件分岐を活用したデジタルものづくりが行えることが明らかになった。また小学 4 年生でも約 30 時間の授業を通じて、同様のプログラミングに関する概念の活用したデジタルものづくりが可能であることも分かった。

授業を通じて児童が作品製作についてまとめたレポートのテキストについて形態素解析と個々の記述を確認したところ、デジタルものづくりを通じた児童の学びは、プログラミングに関する概念獲得に留まらず、デザインやメカニズム、コンピュータの仕組み、ものづくりの方法、学びそのものなど多岐に渡ることも明らかになった。この結果から、小学校におけるデジタルものづくり教育にはプログラミングのための知識など特定の知識獲得と、児童それぞれの興味を背景とした多様な学びを育むことが可能であることが分かった。

プログラミングなどの特定の知識獲得に限定すれば、教授中心の授業デザインでもデジタルものづくり教育は実現可能である。しかしデジタルものづくり教育は、総合的な学習の時間が本来目指す自律的学習者を育むための学習であることも本研究を通じて明らかにした。

そこでデジタルものづくりを通じて、児童の自律的で多様な学びを生み出すための授業デザインとして、プログラミングなど基本的な知識は全ての児童が共通に学び、ものづくり活動では児童それぞれが自らの活動と学びを決めることができるよう、共通の学習目標と児童個々の学習目標の 2 段階の学習目標を設定し授業を構造化する授業デザインを、初等教育におけるデジタルものづくり授業のデザインとして提言した。

8-4. デジタルものづくり授業のデザインと運営で気をつけるべき 21 の事柄

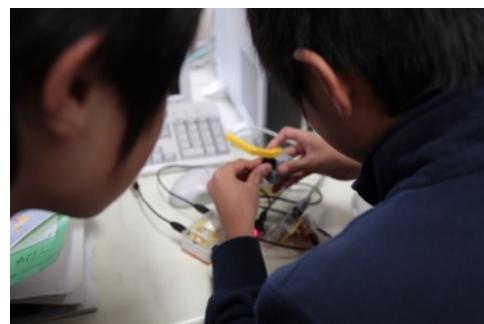
最後に小学校でのデジタルものづくり授業デザインと授業運営において筆者が実施している項目をあげたい。コンピュータの草創期に Papert と Solomon(1971)はコンピュータが普及し、教育に活用される近未来を見据えて書いた「Twenty things to do with computer」を示した。ここではデジタルものづくり授業を行う 21 世紀の小学校で、教師が授業デザインや授業運営で気をつけるべき 21 の事柄を紹介したい。



以下はデジタルものづくり授業のデザインにあたって気をつけるべきことである。

1.児童を専門家にする

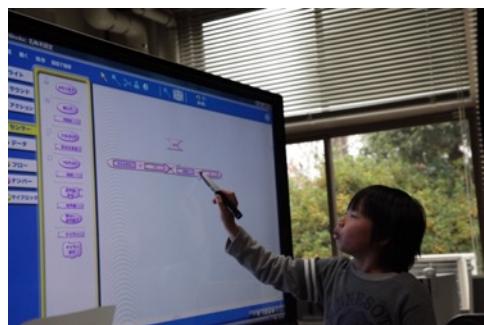
デジタルものづくりを通じて学ぶためには、受動的ではなく自律的、能動的な学習者になる必要がある。発明家やデザイナー、研究者など、児童に専門家のつもりで活動することを促すことで、単にものづくりを楽しむだけでなく、自分にとって意味のある目的を見出しやすくなる。また児童が目的を持つことで活動の中で迷ってしまった際に立ち返りやすくなる。



2.児童を先生にする。先生も学び手になる。

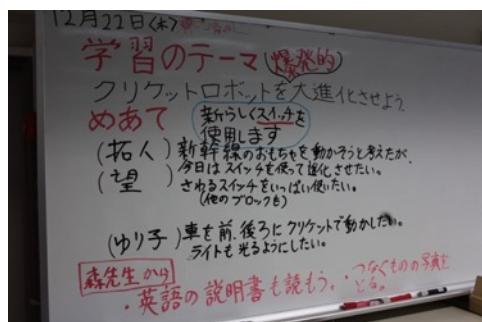
児童を自律的で能動的な学習者にする別の方法は、児童に教え手(小さな先生)となってもらうことである。デジタルものづくり活動のなかでは、児童の人数だけ様々な問題が発生する。自ら問題を解決することは、児童にとっての学びとなるが、一人で解決できない問題に直面した際には、

教師を含めて熟達者による適切なタイミングでの適切な助言が必要となる。一人の教師が対応できる児童の数には限りがあるため、児童が積極的に他の児童を教えることは有効な手段になる。また教師も積極的に児童から学ぶ姿勢を見せてることで、教室に教え手と学び手がフレキシブルに入れ替わる文化をつくることができる。



3.授業での活動のめあてを書き出す

Creative Learning Spiral (Resnick 2007) は、デジタルものづくりから学ぶための有効な方法である。時間の制約がある授業の中で、Creative Learning Spiral(創造的学びの循環)を実現するためには、Imagine(想像する) – Create(つくる) – Play(遊ぶ, ためす) – Share(共有する) – Reflect(振り返る)のそれぞれの活動を明確にする必要がある。活動のめあてを書き出すことで授業内で活動を方向付けるとともに”Imagine”的なきつかけをつくることができる。



4.協働作業の機会を取り入れる

ペアやグループでの協働作業に取り組むことで、お互いに「つくる」作業を通じて学び合う機会をつくる。また個人での作業と協働作業を繰り返すことで、児童が自分で何もせずにグループに依存してしまうことを防ぐとともに、個人での学びとグループでの学びの相互作用を生み出す。



5.発見したことを共有する

児童が授業のなかで発見したことを共有するために、短い発表の時間を設ける。また発表の時間以外にもインフォーマルな共有の機会ができるように、活動中にもほかの児童の活動を見にいくことができるようとする。児童はインフォーマルなアイデア共有を常に積極的に行っており、作品づくりをしていると近くで作業している児童の作品が似ていることがよくある。



6.自分の作品を発表する

児童が製作した作品を発表する時間を設ける。毎回の授業での短い発表のほかに、複数回に渡って実施する授業では発表と発表のための準備をする回を設ける。発表することで自分の考えをまとめ、他の児童から意見をもらうことで新しい発見へつながる。



7.授業での活動の振り返りをする

授業の終盤に2回に分けて、活動の振り返りを行う。クラス内での共有のために、今日の授業での活動を振り返り、自分の言葉でまとめる。その後、他の児童の発表を聞き、次の時間に試したいことを含めて、もう一度活動の振り返りをまとめる。



8.自分で自分を評価する

他の授業と異なり、自分で作品をつくる授業では学習目標を一律に決めるのは難しい。そこで自分で目標をたて、毎回の授業での振り返りや発表の時間などで、自分で自分の活動を評価する機会をつくることが重要である。



9.間違いを有効に活用できるようにする

プログラミングにおいて、デバッグは学びのための最も強力な手段である(Papert 1980)。多くの児童は間違うことに恐怖心があるため、教師は授業のはじめに、この授業では間違えることは悪くなく、間違えから、いかに自分で答えを見つけるかが大事であることを伝える何度も必要がある。



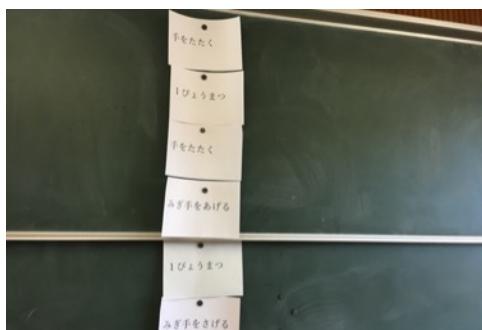
10.再度挑戦できる機会をつくる

より児童が試行錯誤から学ぶ機会をつくるために、より多くの製作時間を確保する。作品を根本的に変えないと新しいアイデアを試すことができない場合もあるため、2回以上の製作する機会を設けることで、前回の製作の経験や他の児童から得たアイデアを新しく試しやすくなる。



11.頭だけで考えずに身体を使って考える

プログラミングをする場合、頭を使って考えることが中心になる。頭だけでなく身体を使いながら考えることは、児童の学びにとって重要である(Papert 1980)。授業の冒頭でコンピュータを使わずに、児童がコンピュータになったつもりで手を挙げるなどの命令に従って体を動かしてみることもスムーズな導入になる。また、動きがうまくいかない場合は、自分の体で動きをまねしてみることも問題と解決方法の発見に繋がる。



次にデジタルものづくり授業の運営において気をつけるべき事項を挙げる。

12.児童自身が授業の準備する

準備された環境ではなく、児童自身が授業の準備に関わることで、授業にも児童が主体的にデジタルものづくりに関わることができる。作品づくりのための材料を集めて、授業を持ってきたり、児童の中で係を決めて、授業の準備を教師とともにを行うことも有益である。



13.児童に片付けをお願いする

ロボットづくりなど、沢山のものを使う場合、教師が全てを片付けるのは大変である。児童に自分でつくった作品を分解して片付けてもらうことで、児童にとっては製作過程を振り返る機会になる。つくる際だけでなく、分解をしながら発見をすることも多い。



14.授業進行にメリハリをつける

児童はつくる活動に夢中になっていると、話を聞くなどの活動に移るのが難しくなる。そこで児童にも予め教師が話をする際には聞くように約束する。それでも児童を次の活動に移すことは難しいが、決して児童をプログラムするように無理矢理命令に従わせるのではなく、教師は児童がそれだけつくる活動に夢中になっていることを覚悟して授業運営する必要がある。



15.真似をするように促す

児童が真似をすることや真似をされることを嫌がる場合がある。いいアイデアは積極的に真似をして、自分の作品に取り入れることを勧める。その際は新しいアイデアを加えることや、元のアイデアを作り出した児童のことをきちんと尊重するように促す。



16.教えない教師になる

ディジタルものづくり授業では、教師が児童にすべてを教えることはできない。教師がすべてを教えることを諦め、分からることは児童と一緒に探求することを心がけることで、教師自身も新しい教材を取り扱う恐怖から解放される。

17.児童を放つておく

児童の興味は多岐に渡る。教師にとっては遠まわりに見えて、児童にとっては必要な過程である場合もある。児童に教えすぎないためにも、教師は児童から少し距離を置き見守ることが必要である。ただし操作などの技術的な問題はすぐにサポートすることも必要である。



18.児童の素の学びの姿を見つめる

教科学習では、学習塾での補完的な知識の学びもあり、児童の学びそのものに対する素の姿をみることが難しい。デジタルものづくりでは児童の活動に向かい合う姿を知り、学びへの向き合い方を見る機会にもなる（ただし、学校外でプログラミングやロボット作り体験ができる教室も増えていくので注意してみる必要がある）。



19.身近なものと関連づけて説明する

家電からおもちゃや、ゲームまで児童の身の回りにある多くのものにコンピュータが埋め込まれプログラムで制御されている。身近なものの仕組みを考えることでも、コンピュータとプログラミングに関する理解が深まる。



デジタルものづくりは、作品づくりという目的の実現や、そのための問題解決、さらにものづくりの過程で興味を持った事柄を深く学ぶこときっかけを提供するものである。2020年度からの小学校でのプログラミング教育の必修化にともない、広い意味でのデジタルものづくりを全ての児童が体験できることになる。一方で授業方法によっては、単に知識や技能としてのプログラミング教育に留まってしまう可能性も否めない。そこで最後に以下の2つの事柄を追加したい。

20. デジタルものづくりやプログラミングは学びに役立つことを伝える

児童や保護者、教師に中にはスキルとしてのプログラミングの習得のために、プログラミング教育が必要であると思っている人も多い。プログラミングは学び方の学びに役立つことや、プログラミング以外のものづくりを通じても、それは身につくことを伝える(ある児童からは、将来はお花屋になるのでプログラミングは私には必要ないと言わされたことがある)。

21. ほかの授業で活用できることを考える

児童と教師の双方にとって、プログラミングやものづくり授業での体験は、ほかの授業でも活用できる。具体的にどのように活用できるかを考えることで、普段の授業にも新しい学びを導入することにつながる。

参考文献

- 荒木貴之(2008)「ロボットが教室にやってくる」, 教育出版
- Blikstein, P. (2013)Digital Fabrication and 'Making' in Education: The Democratization of Invention. In J. Walter-Herrmann & C. Büching (Eds.), FabLabs: Of Machines, Makers and Inventors. Bielefeld: Transcript Publishers.
- Brown, A. (1994) The Advancement of Learning, Educational Researcher, Vol 23, No.8 pp.4-12
- Brennan, K. and Resnick, M.(2012) New Frameworks for Studying and Assessing the Development of Computational Thinking. Proceedings of the 2012 annual meeting of the American Educational Research Association.
- CAMP,「こどもたちの「共に創る力」を育む SCSK グループの社会貢献活動」,
<http://www.camp-k.com> (最終アクセス日:2018年2月10日)
- Cole. M.(2002)「文化心理学 発達・認知・活動への文化-歴史的アプローチ」, 天野清訳, 新曜社
- Coder Dojo Japan,「子どものためのプログラミング道場」,
<https://coderdojo.jp> (最終アクセス日:2018年2月10日)
- Code.Org, 「Anybody can learn」, <https://code.org> (最終アクセス日:2018年2月10日)
- Csikszentmihalyi, M. (2000)「楽しみの社会学」, 今村浩明訳, 新思索社
- Denning, P.(2017) Remaining Trouble Spots with Computational Thinking, Communications of the ACM, Vol. 60 No. 6, pp.33-39
- Department of Education, UK (2013) National curriculum in England: computing programmes of study,
<https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study> (最終アクセス日:2018年2月10日)
- Department of Education, UK (2013) National curriculum in England: design and technology programmes of study,
<https://www.gov.uk/government/publications/national-curriculum-in-england-design-and-technology-programmes-of-study> (最終アクセス日:2018年2月10日)
- 土橋永一,北尾文孝,室屋 正俊,鈴木 勢津子(1988)「ロゴと子どもと先生」,ロゴジャパン, 京都
- Gershenfeld, N.(2005)Fab: The Coming Revolution on Your Desktop – from Personal Computers to Personal Fabrication ,Basic Books, New York
- Dewey, J. (1938) Experience and Education, 市村尚久訳『経験と教育』講談社
- Harel, I.(1986)Kids as Software Desiners: An assessment of children's concepts of learning and teaching. The proceedings of LOGO 86, pp.150-151

- Harel, I.(1991)Children Designers, Ablex Publishing Corporation, NJ.
- Illich, I(1977)「脱学校の社会」, 東洋, 小澤周三訳, 東京創元社
- 自己調整学習研究会(2012)「自己調整学習 理論と実践の新たな展開へ」,北大路書房
- Kafai, Y. and Resnick, M.(1996)“Constructionism in Practice: Designing, Thinking, and Learning in A Digital World”, LEA.
- 神谷加代(2015)「子どもにプログラミングを学ばせるべき 6 つの理由」,インプレス
- Kay, A. C. and Goldberg, A. (1977) Personal dynamic media. Computer. 10(3), pp. 31-41.
- 加澤恒雄(2004)「ペタゴジーからアンドラゴジーへ -教育の社会学的・実践的研究-」, 大学教育出版
- Knowles, M.(2013)「成人学習者とは何か 見過ごされてきた人たち」,掘薰夫, 三輪建二監訳, 凤書房
- 久保田賢一 (2000) 「構成主義のパラダイムと学習環境デザイン」,関西大学出版部
- Lave, J. and Wenger,E.(1991)Situated Learning: Legitimate Peripheral Participation. Cambridge University Press(佐伯胖訳(1993)「状況に埋め込まれた学習 ー正統的周辺参加」,産業図書)
- Micro:bit 財団, 「Micro:bit 財団は 2020 年までに BBC micro:bit を 30 万人の日本の子どもたちに届けることを目指す」,
- <http://microbit.org/ja/2017-08-03-japan-launch/>(最終アクセス日 :2018 年 2 月 10 日)
- 文部科学省(2016)「小学校段階におけるプログラミング教育の在り方について(議論の取りまとめ),
http://www.mext.go.jp/b_menu/shingi/chukyo/chukyo3/053/siryo/_icsFiles/afieldfile/2016/07/08/1373901_12.pdf (最終アクセス日 :2018 年 2 月 10 日)
- 文部科学省(2016)「小学校学習指導要領」,
http://www.mext.go.jp/component/a_menu/education/micro_detail/_icsFiles/afieldfile/2017/05/12/1384661_4_2.pdf (最終アクセス日 :2018 年 2 月 10 日)
- Martinez,S. & Stager, G.(2013) Invent to Learn, Constructing Modern Knowledge Press. (阿部和広監訳(2015)「作ることで学ぶ」, オライリージャパン)
- 美馬のゆり, 山内祐平(2005)「未来の学びをデザインする」, 東京大学出版
- 茂木一司 編集代表(2010)「協同と表現のワークショップ」, 東信堂
- 森秀樹(2007)「クリケットを使ったワークショップ型授業のための教員支援」,日本教育工学会研究報告集 JSET07-4, pp.215-218
- 森秀樹,杉澤学(2008)「Cricket と Scratch を使った小学校情報教育の試み」,第 24 回日本教育工学会大会発表論文集:489-490
- 森秀樹,杉澤学,張海,前迫孝憲(2009)「Scratch を用いた小学校情報授業のデザインと実践」,第 25 回日本教育工学会大会発表論文集:671-672

- 森秀樹, 杉澤学, 前迫孝憲(2010)「プログラミングを活用した小学校理科学習」, 日本教育工学会第 26 回全国大会講演論文集, pp.737-738
- 森秀樹(2010)「Scratch を用いた文系大学生向けプログラミング教育」, 日本教育工学会論文誌 34(Suppl.), pp.141-144
- Mori, H.(2011) Practical Study on Computer-Embedded Monodukuri Workshops for Elementary School Students. *The Journal of Information and Systems in Education* Vol.9(1) ,pp.25-34
- 森秀樹, 杉澤学, 張海, 前迫孝憲(2011)「Scratch を用いた小学校プログラミング授業の実践～小学生を対象としたプログラミング教育の再考～」, 日本教育工学会論文誌 34(4), pp.387-394
- 森秀樹・杉澤学・前迫孝憲(2012)「デジタルおもちゃづくりを取り入れた小学校ものづくり授業」, 日本教育工学会第 28 回全国大会講演論文集, pp.871-872
- 森秀樹(2014)「いつでもどこでも誰でもデジタルものづくりツール 「Programmable Battery」の開発」, 日本教育工学会第 30 回全国大会講演論文集, pp. 877-878
- Mori, H.(2016) What did students learn in programming workshops? – Comparison of students' reports from two programming workshops in Japan -. *FabLearn 2016, Proceedings of the 6th Annual Conference on Creativity and Fabrication in Education*, pp.103-106
- Mori, H.(2017) The Programmable Battery: A tool to make computational making more simple, playful, and meaningful. *IDC '17: Proceedings of the 2017 Conference on Interaction Design and Children*, pp.515-519
- Mori, H.(2017) Twenty-One Things to Do with Scratch in the Twenty-First Century Classroom. *Scratch2017BDX : Opening, Inspiring, Connecting*, Jul 2017, Bordeaux, France. pp.
- 森秀樹(2017)「いつでもどこでも誰でもデジタルものづくりツール「Programmable Battery」の開発(2)」, 日本教育工学会第 33 回全国大会講演論文集 pp.757-758
- 森政弘(1999)「ロボコン博士のもの作り遊論」, オーム社
- 中野民夫(2001)「ワークショップー新しい学びと創造の場」, 岩波書店
- 中野民夫(2003)「ファシリテーション革命」, 岩波書店
- 奈良女子大学文学部附属小学校学習研究会編(1998)「総合的な学習の提案」, 明治図書
- 奈良女子大学附属小学校学習研究会(2015)「自律的学ぶ子どもを育てる奈良の学習法『話す力, 書く力, つなぐ力』を育てる」, 明治図書
- Nilson, L.(2017)「学生を自己調整学習者に育てる」, 美馬のゆり, 伊藤崇達監訳, 北大路書房
- OECD(2011)「学習成果の認証と評価—働くための知識・スキル・能力の可視化」山形大学教育企画室監訳, 松田岳士訳, 明石書店
- 太田剛, 森本容介, 加藤浩(2016)「諸外国のプログラミング教育を含む情報教育カリキュラムに関する調査—英国, オーストラリア, 米国を中心として—」日本教育工学会論文誌 40(3), pp. 197-208

- Papert, S. and Solomon, C. (1971) Twenty things to do with a computer, Artificial Intelligence Memo No. 248, A.I. Laboratory, Massachusetts Institute of Technology
- Papert, S.(1971) Teaching children thinking,LOGO Memo No.2, A.I. Laboratory, Massachusetts Institute of Technology
- Papert, S.(1980) Mindstorms: Children, Computers, and Powerful Ideas. Basic Books, New York
- Papert, S.(2005) You Can't Think about Thinking without Thinking about Thinking about Something, Contemporary Issues in Technology and Teacher Education, v5 n2-3 pp.366-367
- Pea,R.(1983) Logo Programming and Problem Solving,Technical Report No. 12.
- Resnick, M. and Silverman, B.(2005) Some Reflections on Designing Construction Kits for Kids. Proceedings of Interaction Design and Children conference:117-122
- Resnick, M.(1994) Turtles, Termites, and Traffic Jams: Explorations in Massively Parallel Microworlds. A Bradford Book
- Resnick, M. et al.(1996) Beyond Black Boxes: Bringing Transparency and Aesthetics Back to Scientific Instruments. Proposal to the National Science Foundation
- Resnick, M.(1998) Technologies for Lifelong Kindergarten. Educational Technology Research & Development, vol. 46, no. 4
- Resnick, M.(2007) Sowing the Seeds for a More Creative Society. Learning & Leading with Technology December/January 2007-08:18-22
- Resnick, M. et al(2009)Programming for All. Communications of the ACM Vol.52 No.11:60-67
- Resnick, M. (2014) Give P's a Chance: Projects, Peers, Passion, Play. Constructionism and Creativity conference, opening keynote. Vienna.
- Resnick, M. (2017) Lifelong Kindergarten: Cultivating Creativity through Projects, Passions, Peers, and Play. MIT Press.
- Rogoff, B.(2003)The Cultural Nature of Human Development. Oxford University Press(當眞千賀子訳(2006)「文化的営みとしての発達」,新曜社)
- Rusk, N. et al. (2005) Rethinking Robotics: Engaging Girls in Creative Engineering. Proposal to the National Science Foundation.
- Rusk, N. et al(2008)New Pathways into Robotics: Strategies for Broadening Participation. Journal of Science Education and Technology Volume 17(1):59-69
- 佐伯胖(1999)「マルチメディアと教育」,太郎次郎社
- Schön,D.(2001)「専門家の知恵——反省的実践家は行為しながら考える」,佐藤学,秋田喜代美訳,ゆみる出版
- 柴田義松(2006)「ヴィゴツキー入門」,寺子屋新書

ソフトバンク「Pepper 社会貢献プログラム スクールチャレンジ」,
https://www.softbank.jp/corp/csr/next_generation/pepper/school/
(最終アクセス日:2018年2月10日)

Solomon,C.(1988)「子供の学習とコンピュータ 新しい教育理論によるCAIへのアプローチ」岡本敏雄,赤堀侃司,横山節雄監訳,パソコンメディア

菅井勝雄(1993)「教育工学 一構成主義の学習論と出あう」,教育学研究,第60巻3号,pp.27-37

菅井勝雄(2000)「構成主義」「社会的構成主義」『日本教育工学事典』,実教出版

菅井勝雄,赤堀侃司,野嶋栄一郎(2002)「情報教育論 一教育工学のアプローチ」,日本放送出版協会

鈴木勢津子(1989)「考える力をはぐくむコンピュータ教育」,啓学出版

田中耕治(1999)「総合学習の可能性を問う 奈良女子大学文学部附属小学校の「しごと」実践に学ぶ」,ミネルヴァ書房

戸塚滝登(1995)「コンピュータ教育の銀河」,晚成書房

上田信行(2005)「経験のパブリッシング」『国立民族学博物館を活用した異文化理解教育のプログラム開発』国立民族学博物館調査報告

上田信行,中原淳(2013)「プレイフル・ラーニング」,三省堂

Vygotsky, L. S.(1978) *Mind in Society: The Development of Higher Psychological Processes*, Harvard University Press

Vygotsky, L. S.(2003)「発達の最近接領域の理論 教授・学習過程における子どもの発達」,土井捷三・神谷栄司訳,三学出版

Wing, J. (2006) Computational Thinking. Communications of the ACM. vol. 49, no. 3, pp. 33-35

山内祐平,森玲奈,安斎勇樹(2013)「ワークショップデザイン論 -創ることで学ぶ-」,慶應義塾大学出版

謝辞

授業実践に参加いただいたすべての先生、児童の皆さんに感謝申し上げます。

特に奈良女子大学附属小学校の杉澤学教諭には、まだ小学校での授業実績がなかった 2002 年にお声がけをいただき、以来 10 年以上に渡り、Cricket と Scratch を活用した授業をともに考え、実践していくなかで貴重な助言をいただきました。Cricket と Scratch の開発者である MIT メディアラボの Mitchel Resnick 教授には、研究員として MIT メディアラボに在籍以来、筆者の授業実践やワークショップに貴重な意見をいただきました。またコンピュータクラブハウスの立ち上げなどに関わり、沢山の実践経験をお持ちの Natalie Rusk 博士には、実践のなかでの子どもたちの活動の意味づけを考えるヒントをいただきました。Cricket を使ったデジタルものづくり授業は、筆者が所属していた株式会社 CSK(現 SCSK)の社会貢献活動「CAMP」で実施してきたワークショップのデザインが元になっています。2001 年の立ち上げから、CAMP のメンバーともに試行錯誤をしてきたことが小学校での授業デザインにも活きています。また CAMP の活動立ち上げから、同志社女子大学の上田信行教授にはワークショップ、コンストラクショニズムと学び、Logo からはじまった MIT での子どもの学びのためのツール開発の歴史など、沢山のことを一緒にプロジェクトをさせていただきました。さらにこの分野を学術的に深めていく面白さを教えていただき、当時会社員であった私に学会発表の際のご指導をいただきましたなど、上田先生との出会いがなければ、本研究ははじめられませんでした。学校外で実施してきた Cricket を使ったワークショップを小学校授業で実践するというアイデアは、当時 CAMP プロジェクトの責任者であった田村拓氏と北川美宏氏によるものです。2002 年から 2008 年まで本研究は CAMP プロジェクトの一環として進めました。また 2005 年からは大阪大学大学院人間科学研究科の教育工学研究室に社会人大学院生として在籍し、菅井勝雄教授と前迫孝憲教授にご指導いただきました。これらの大変な下地があつて、2008 年に前迫教授のもと、教育工学研究室の助教となってから本格的に小学校でのデジタルもののづくり教育について、研究を進めることができました。前迫教授から自由な研究の機会と時間をいただいたことで、プログラミング教育が必修化となる前に、本研究をじっくりとすすめることができました。また同時に教育工学研究室の皆さんからの多くのアドバイスをいただき、それまでの教育実践から教育実践研究へと大きく前進させることができました。前迫教授、西森年寿准教授、齊藤貴浩教授には、都度ご指導をいただきましたことに感謝申し上げます。齊藤教授には博士論文の仕上げに際しまして、細部に渡りご指導いただきました。そして最後になりますが、本研究は家族の支えなしには進めることはできませんでした。家族にも感謝の気持ちを伝えたいと思います。

ブロック玩具で繰り返しつくったテレビアニメや特撮ヒーロー一番組にでてきたロボット、手に傷をつくりながら何度もつくった竹とんぼ、教育テレビを見ながらわくわくしてつくった空き箱や廃材を使った工作、幼い頃に自分自身がつくって、家族や友達にみせて楽しんだものづくりが、本研究をはじ

めるに至った原体験と個人的なきっかけです。MIT メディアラボが開発した Cricket をはじめてみた 1998 年に、昔親しんだものづくりにコンピュータを埋め込むことで新しいものづくりができるることに興奮を覚えました。その後、幸運にも 2001 年 3 月に京都けいはんなに開設した大川センターで、当時の株式会社 CSK の社会貢献活動として、子どもを対象としたワークショップの研究開発プロジェクト「CAMP」に計画段階から関わる機会を得ました。そこで最初に行ったのが Cricket を使った動くおもちゃづくりのワークショップでした。小学校での総合的な学習の時間の導入時期とも重なり、また新しいものづくりや情報、サイエンスに関わる研究授業の一環として、CAMP で実施してきた Cricket ワークショップを学校授業向けにデザインし、実践したのが本研究のはじまりです。

ものづくりを通じた学びについては Papert(1980)の幼少期にギアを組み合わせたものづくりからの学びのような体験は私自身にはありません。もちろん、幼少期のものづくり体験のなかで何かしらの学びがあったとは思います。しかし Papert が自身のギアとの出会いを指して言う「Object to think with : 考えるための道具」となるようなものづくりの体験ではありませんでした。Papert はギアを考えるための道具として算数や数学を考える際にも活用していました。私自身がものづくりを通じた学びを明確に意識できたのは大学卒業後、株式会社セガでソフトウェア(子ども向けのエデュテインメントソフトウェア)の開発に携わった機会でした。その時のものづくりの体験そのものが、今でも研究や仕事をすすめる上で考えるための道具となっています。

本論文ではデジタルものづくりを通じた学びのための授業デザインについて述べてきました。多様なものづくりを可能とするデジタルものづくりの体験を通じて、一人でも多くの児童にとってデジタルものづくりが考えるための道具となるきっかけになればと思っています。Kay と Goldberg (1977)は、「コンピュータはあらゆるメディアとなり得るメタメディア」であると提言しました。コンピュータを活用したデジタルものづくりは、様々な人々にとって様々な考えるための道具となるものとなり得るのではないかと思っています。Papert(1980)がギアとの出会いを “I felt in love with gears” と表現したように、考えるための道具になるためには感情は大事な要素です。その点では、コンピュータに囲まれて育ってきた現代の子どもたちにとって、デジタルものづくりは自然に興味を持ち感情面に訴える活動となると考えています。

もちろん、ものづくりは子どもたちにとって活動の一つに過ぎません。学校内外のあらゆる活動が児童にとって考えるための道具になる可能性を持っています。野球やサッカーなどのスポーツ活動やピアノや書道などの習い事、日常の遊びの場合もあるかもしれません。また子どもたちが一日の大半の時間を過ごす学校は、考えるための道具との出会いの場として最適な場でもあります。私自身はプログラミング教育など、デジタルものづくり授業を小学校で実施することをきっかけに、多様

な授業が増え、より多くの授業が子どもたちにとって、学校が考えるための道具との出会いの場となることを望んでいます。初等教育において、デジタルものづくり教育を実施する最大の効果は何か。大きな問い合わせありがとうございます、現時点での私自身の答えは、子どもたちが自ら夢中になれる活動のなかでの学び方を学ぶことです。

博士論文を書き終えた今、論文を書くことを通じて自分自身が博士論文をまとめることを通じて何を学んだのか、次の活動のための振り返りをはじめたいと思います。

2018年2月23日