小学校「総合的な学習の時間」での
プログラミングカリキュラムの開発と授業実践

古田貴久・奥木芳明・高秀涼佳
蓮見龍希・渡邊　晶

群馬大学教育実践研究　別刷
第38号　207～214頁　2021

群馬大学共同教育学部　附属教育実践センター

群馬大学教育実践研究　第38号　207～214頁　2021

小学校「総合的な学習の時間」での プログラミングカリキュラムの開発と授業実践

古田貴久[1)]・奥木芳明[2)]・高秀涼佳[1)]
蓮見龍希[1)]・渡邊晶[1)]

1）群馬大学共同教育学部技術教育講座
2）長野原町立北軽井沢小学校

Development and practice of computer programming curriculums for the period for integrated studies in elementary school

Takahisa FURUTA[1)], Yoshiaki OKUGI[2)], Suzuka TAKAHIDE[1)]
Tatsuki HASUMI[1)], Akira WATANABE[1)]

1）Department of Technology Education, Cooperative Faculty of Education, Gunma University
2）Kitakaruizawa Elementary School, Naganohara

キーワード：カリキュラム開発、プログラミング、小学校、総合的な学習の時間
Keywords : curriculum development, computer programming, elementary school, the period for integrated studies

（2020年10月30日受理）

1．はじめに

小学校でのプログラミング指導が必修化されて1年が経つ。小学校でのプログラミング指導については、1980年代はLogoなどを活用した教育実践が行われてきたが（鈴木，1989)、やがてすたれて2010年ごろにはほとんど行われなくなった。その潮目が変わったのは、マサチューセッツ工科大学のResnickらが開発したScratchに代表される、直感的にプログラムを書くことのできるプログラミング言語の登場であろう（森，杉澤，張，前迫，2011)。日本でもドリトルやビスケットといった、子ども向けプログラミング言語が開発・提供され、それらの言語を使った授業実践が行われてきた。

しかしながら、小学校での授業実践を容易にしない原因の1つに、いまだプログラミング指導の教材が不十分であることが挙げられる（坂巻，福島，2017)。確かに小学校でのプログラミング授業実践は数多くなされているし、それらの授業で使われた児童用の手元資料や、液晶プロジェクタで投影された提示資料や指導案もWebページで公開されている。しかしながら、そのほとんどはその学校での全体的な教育指導計画の一部として、具体的で特定の狙いを効率的に達成するように構成されている。各学校の実態を前提としているため、他の学校で同じ内容を実施するには、あらかじめ教員が相当資料を読み込んで、書かれていない事項を補足しながら指導していかなくてはならない。だが、筆者らの交流のある数名の小学校教員によると、実際にプログラミングを指導する立場としては、いわゆる教科書の「指導書」のような資料が要望されるようである。

本論文では、小学校の「総合的な学習の時間」にお

いて、各教科でプログラミングを取り入れた授業を行うための基礎となる知識・技能を指導することを狙いとし、ScratchとMicro:bitを対象とした、それぞれ全６時間のプログラミング入門のカリキュラムを開発した。カリキュラムは指導案と児童用手元資料（マニュアル）から構成されるが、マニュアルは児童が一人でマニュアルを読みながらでもプログラミングを習得できることを目標とした。図１はマニュアルのページ例である。本論文では、カリキュラムの説明と、2020年10月に群馬県内の小学校で実施した授業実践の結果、および、授業実践を通じて得られた、授業を行う上で特に重要なポイントについて述べる。なお、指導案とマニュアルは改訂して近日中に公開する予定である。

２．カリキュラムの概要

１回45分の授業時間は、どの時間も前半の30分はマニュアルを使いながら一斉指導方式で解説し、後半の15分は各自の自由改造を基本構造とした。自由改造では、どのような改造が考えられるかマニュアルに例を数個示しておいて、児童はその例を見ながら、自分自身のアイデアを加えて作品を作り上げる展開とした。なおこの自由改造の時間は、前半の一斉授業が何らかの理由で伸びたり逆に短時間で終わったときに、授業時間の調整を行うためでもある。実習の授業では、児童・生徒によって進度が大きく異なるので、このような授業時間内に収める工夫が欠かせない。

Scratchでは、最初の２時間で、くり返しと条件判断など情報処理の基本的な流れをカバーした。それに続く３時間は、スプライトを追加して、条件判断の対象を増やし、変数を作成するなどの発展的な内容とした。最後の１時間は、フィジカルコンピューティング（コンピュータがキーボードとマウス以外のインタフェースを持つことを強く意識させること、すなわち、人間のフィジカル（身体的）な動作に反応したコンピュータの処理（O'Sullivan, Igoe, 2004）の体験とした。

Micro:bitでは、最初の２時間は順次、くり返し、センサと条件判断のプログラミングを行った。この際、MakeCodeエディタのシミュレータだけでなく、積極的に実機にダウンロードさせて、実機を揺さぶったり手をかざして当たる光の明るさを変えさせて、プログラムしたとおりに動くことを確かめさせた。続く３時間は、段ボールを材料とした工作と組み合わせた作品制作を行い、最後の１時間を作品の発表会とした。

図１．本研究で作成した児童用マニュアルの例（左：Scratcth、右：Micro:bit）

3．Scratchの授業

全６時間の指導計画を構成した（表１）。各時間の主な内容は以下の通りである。合わせて、それぞれの時間で、実施上特に留意が必要なことを述べる。

１時間目　まずScratchのプログラム開発環境各部の名称を紹介した。次に、「10歩動かす」を使ってスプライトを動かしたら、その下に「１秒待つ」と「15度回す」をつなげ、プログラムが上から順に実行されること、「ずっと」で無限ループすることを確認していった。次いで、「ペン」を使って軌跡を残すプログラムを作り、線の色や太さを変えながらさまざまな図形を描かせた。最後に、正方形や正三角形などを描かせた。正三角形は外角の発想が必要なので、大学生でも初めてプログラムを書いたときは間違えがちである。子どもに達成感を持たせるために、正三角形の描画は多角形をいくつか描いてからがよいであろう。

２時間目　まず「座標」について説明した。座標は部分的に既習事項であるが、スプライトを画面上で動かすとき、「ｘ座標を10ずつ変える」など座標に言及するブロックを使うので、ｘ-ｙ座標について触れておく必要がある。なお、負の数も既習事項ではないが必要なので、「マイナスの記号がついていると反対側に動く」と説明した。次に、右向き矢印キーが押されるとスプライトが右に移動する条件分岐するプログラムを作り、左向き矢印キーが押されたら左に移動するコードを追加して、左右に動かせるようにした。

指導上の留意点として３つ挙げたい。１点目は、「動き」パレットには「ｘ座標を10ずつ変える」と「ｘ座標を10にする」というブロックがあることである。これらは形状がとてもよく似ているので、「ｘ座標を10にする」を使っている児童がときどき見られ、プログラムを実行してもスプライトが動かない原因になる。２点目は、「もし……なら」のブロックを２つ組み合わせるとき、これら２つのブロックを順次に並べるのではなく、入れ子にする児童がいることである。プログラムがどのように実行されるかを理解すればこのようなミスは起きないが、そうでない児童はあまり気にしないで実行する。

３点目は、「矢印キー」と言われてもわからない子どもがいることである。マニュアルではキーボードの全体図を示し、「右向き矢印キーはこれ」と赤い丸で囲って示したが、周囲の友達に教わっている子どもが散見された。

ここまでの２時間で、Scratchプログラミングの導入は終了であり、簡単なプログラムなら書けるようになる。

３時間目　Scratchで作れる作品の幅を広げることを狙いに、シューティングゲームを作りながら、複数のスプライトの間でやり取りを行うプログラミングを

表１．Scratchプログラミングの学習指導計画（全６時間）

校時	学習の目標
1	・Scratchのプログラム環境について知る【知識・技能】 ・Scratchのスプライトを移動させることができる【思考・判断・表現】 ・スプライトが残した軌跡で図形を描くことができる【思考・判断・表現】
2	・Scratchで条件分岐について知る【知識・技能】 ・スプライトなどの見た目の変え方を知る【知識・技能】 ・押したキーによってスプライトの位置や見た目を変えるプログラムを作る【思考・判断・表現】
3	・Scratchで複数のスプライトを使う方法を知る【知識・技能】 ・画面の端に着くまで、他のスプライトと接触するまでなど、条件付き反復を知る【知識・技能】 ・マトあてゲームを改造してオリジナルの設定を作り、他の作品と比べる【思考・判断・表現】
4	・Scratchで変数を使う方法を知る【知識・技能】 ・値で条件分岐する方法を知る【知識・技能】 ・プログラムをコピーして効率的にプログラムを作る方法を知る【知識・技能】 ・変数や値を用いた改造をする【思考・判断・表現】
5	・これまで学んだScratchでの順次処理、条件分岐などを使ってプログラムを作る【知識・技能】 ・課題で示された、プログラムの改造の方向性を理解して実践する【思考・判断・表現】
6	・デモ作品を体験し、これまでに学んだScratchで、双方向性のシステムが作れることを知る【知識・技能】 ・自分もプログラミングを使って、何か動かすものを作りたいと思う【主体的に学習に取り組む態度】

扱った。まず、ネコのスプライトに前時に作った矢印キーで左右に動かすプログラムをつけた。次に、ボールのスプライトを追加する手順を説明し、スペースキーが押されたらボールが発射されるプログラムをボールにつけた。なお、「スペースキー」と言われてもわからない子どもがいるので、これもマニュアルに図示した。ボールのプログラムを徐々に改良していって、ネコのいる位置から発射されて、画面の上端に達したら消滅するようにした。次いで、ボールの当たるマトのスプライトを追加して、ボールが当たったら消滅し、1秒たったらまた画面に現れるようにした。その次に、ステージの背景の変え方を説明した。課題として、スプライトや背景を好きに変えたり増やしたりして、自分の世界観でゲームを作らせた。

4時間目　本時の狙いは変数の導入である。ステージにアルファベットABCが位置に出現するので、マウスでクリックしてアルファベットを消すというゲームである。1つのアルファベットにつけたプログラムを、ドラッグ＆ドロップで他のスプライトに複製する手順を教えて、プログラミングの効率化を体験させた。ゲームでは正解するとスコアが増えていくが、間違えるとライフが減っていき、0になるとゲームオーバーである。この部分は数値の条件判断である。

5時間目　児童のオリジナリティを引き出すことを狙いとして、マニュアルでは簡単な場面設定を行ったあと、何通りかの改造の仕方を提案した。児童は適当な改造を選び、独自のアイデアを加えながら作品作りを行った。

6時間目　フィジカルコンピューティングの体験を狙いとして、MakeyMakeyとMicro:bitを使った作品それぞれ2つずつを、児童をグループに分けて一人ずつ順に体験させた。

4．Micro:bitの授業

全6時間の指導計画を構成した（表2）。

1時間目　MakeCodeエディタのプログラム開発環境を紹介し、LED画面にハートを点滅させるプログラムと、押されたボタンによってLED画面に異なるアイコンを表示するプログラムを作った。1時間目の狙いは開発環境（プログラムのエディタ、シミュレータ、および実機へのダウンロード）に慣れることと、「ずっと」のくり返し（無限ループ）、LED画面にパターンを表示させる方法、および、ボタン入力で分岐するプログラミング知識と技能を身に着けることにある。

2時間目　センサを使うプログラミングの知識と技能の習得を目的として、揺さぶられたらLED画面にランダムに数を表示するサイコロのシミュレータと、周囲の明るさによって表示されるアイコンを変えるプログラムを指導した。

3時間目から5時間目　Micro:bitと、段ボールをハサミやカッターで適当な形状に切り出して、テープやのりでとめた工作品とを組み合わせた作品作りである。今回は3タイプの制作品を用意し、児童にランダムにどれかの制作品を割り当てた。そして、作品ごとに異なるテーブル（作業台）に児童を集めて、それぞれの作品の作り方を説明した資料を使いながら、テーブルごとのTTが指導した。また、児童には自分なり

表2． Micro:bitプログラミングの学習指導計画（全6時間）

校時	学習目標・活動
1	・Micro:bitの特徴と各部の名称を知る【知識・技能】 ・MakeCodeエディタとシュミレータの使い方を知る【知識・技能】 ・LEDディスプレイのプログラミングができる【知識・技能】
2	・Micro:bitのさまざまなセンサについて知る【知識・技能】 ・条件判断文を使って処理を分岐させることができる【思考・判断・表現】
3～5	・製作品の例から1つを選び、創意を加えながら、プログラミングと工作で製作する ・図工的作品をプログラミングで制御することを知る【思考・判断・表現】 ・コンピュータに制御される図工的作品を製作する【知識・技能】 ・より創作的な作品を作るための工夫を凝らす【主体的に学習に取り組む態度】
6	・自分の作品のよさや独自性を友達に説明することができる【思考・判断・表現】 ・友達が作品作りで行った工夫やよさを見出して伝えることができる【思考・判断・表現】 ・コンピュータを使ったより豊かな作品を作っていきたいと思う【主体的に学習に取り組む態度】

の創意を加えることを推奨し、各TTが相談に乗った。

2020年の実践で用意した３つの制作品について説明する。

もぐらたたきゲーム：LED画面に１～３の数がランダムに現れる。その番号のモグラを叩く。弾型に切り取った段ボールに顔を描いたモグラにアルミ箔の電極がつけてあり、電極に触るとMicro:bitのタッチセンサが反応する（図２）。

数の暗記ゲーム：LED画面に１～３の数が６回ランダムに現れる。そのなかで最も表示された回数の多かった番号のボタン（アルミ箔を球のように丸めたもの）にタッチすると正解である。

イライラ棒：ぐにゃぐにゃに折り曲げた50㎝ほどの針金に沿って、直径２㎝ほどの輪っかにした針金を、スタート地点からゴール地点まで一度も接触させずに移動できれば勝ちである。

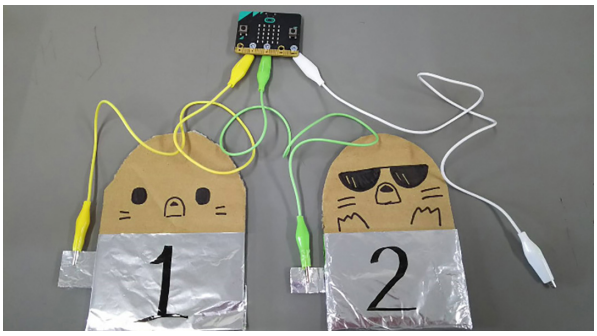

図２．もぐらたたきゲーム

６時間目　最初の10分で同じ作品を作った児童同士で互いの作品を紹介しあった。３～５時間目の作品作りでは、各児童は自分の作品を作り上げることに集中していたので、同じテーブルであっても他の児童がどのような作品を作っていたかを、あまりよく知っているわけではない。そのため、同じタイプの作品が、人によってどのように作られたかを比べるよう促した。後半の20分では、児童を半分に分けて、他のタイプの作品を全員が体験する時間とした。すなわち、半分の児童が他のタイプの作品を体験している間、残りの半分の児童は自分たちのタイプの作品を説明する。体験する児童と説明する児童を入れ替えて、同じことを実施することで、全員が、自分たちの作品の説明と、他のタイプの作品を体験した。

５．授業評価

６時間目の最後に、ScratchとMicro:bitそれぞれを受講した児童らに質問紙を用いたアンケートを行った。回答者はScratchの授業を受けた５年生12名とMicro:bitの授業を受けた６年生12名である。

質問項目は、「プログラミングが好きになった」、「プログラミングが得意になると思う」など授業を通じてプログラミングに対する意識を問う15項目、プログラミングで使った「10歩動かす」などのブロックに対する理解度を問う項目（Scratchは13項目、Micro:bitは９項目）、および、「プログラムを書いたこと」、「友達とプログラムを見せ合ったこと」など授業における６個程度の場面についての評価である。

評価は、それぞれの項目について「とてもそう思う」「だいたいそう思う」「どちらともいえない」「あまりそう思わない」「全然そう思わない」、またはそれに準ずる表現で提示し、１つずつ選んでマルをつけてもらった。なお、ブロックの理解度については、「理解」とはブロックの働きが分かって、自分で使えることであると補足を加えて、５段階でどの程度理解したと思うかを答えてもらった。ブロックの理解度は児童の主観的な評価であるが、森ら（2011）にならった設問である。

児童から得られた回答を、「とてもそう思う」から「全然そう思わない」を５～１に置き換えて、項目ごとに平均値と標準偏差を求めた。主な項目の結果を表３に示す。また、Scratchのブロックに対する理解度の自己評価結果を、平均値と標準偏差を森ら（2011）の結果とともに表４に示す。

Scratch、Micro:bitともに、「プログラミングは楽しかった」、「プログラミングが好きになった」が平均4.6～5.0であり（５点満点）、プログラミングの授業を楽しんで受けていたと言える。また、「プログラミングは将来の役に立つと思う」と「もっと勉強して、いろいろプログラムを作ってみたい」が、平均4.4～4.7であり、より発展的な内容を学ぶことに意欲を示している。

今回作成した児童用のプログラミングマニュアル（手元資料）については、「マニュアルがわかりやすかった」の平均が4.4～4.6であることから、児童にとって適切な難易度であったと言える。

表3．児童による授業評価の結果

質問項目	Scratch	Micro:bit
プログラミングは楽しかった	5.0(0.0)	4.7(0.8)
プログラミングが好きになった	4.8(0.4)	4.6(0.8)
プログラミングは難しくなかった	3.1(1.4)	3.4(0.8)
難しい言葉がいっぱいあった	3.3(1.4)	3.3(1.2)
自分のアイデア(考え)が思いつかなかった	2.7(1.4)	3.2(1.6)
マニュアルがわかりやすかった	4.6(0.8)	4.4(1.1)
プログラミングはめんどうだと思った	1.4(0.8)	1.2(0.4)
プログラミングは将来の役に立つと思う	4.6(0.8)	4.7(0.6)
もっと勉強して、いろいろプログラムを作ってみたい	4.4(0.9)	4.5(1.2)

評価は５～１点。値は平均値、カッコ内は標準偏差

表4．プログラミングで使われたブロックの理解度

ブロックの種類	本研究(Scratch)	森，杉澤，張，前迫(2011)
スプライト制御	4.96(0.20)	4.97(0.17)
くり返し	4.92(0.28)	4.94(0.24)
座標	4.83(0.55)	3.97(1.19)
キー入力の判別処理	4.79(0.71)	4.82(0.72)
条件分岐	5.00(0.00)	4.32(1.09)

どちらも理解度は５～１点の自己評価。
値は平均値、カッコ内は標準偏差。

森ら（2011）が実施した授業後のアンケートでは、「プログラミングは楽しかったですか（５段階［５：楽しかった～１：つまらなかった］)」が平均4.78（*SD*：0.76)、「プログラミングは簡単でしたか（５段階［５：簡単だった～１：難しかった］)」が平均3.22（*SD*：1.27）であった。本研究でも、５段階でScratchが「楽しかった」が平均5.0（*SD*：0.0）であり、「難しくなかった」が平均3.1（*SD*：1.4）であり、児童はプログラミングがとても楽しかったと感じている一方で、必ずしも簡単であるとは感じていないという点で、本研究と森ら（2011）での評価の傾向は一致する。

６．考察

今回の実践を通じて以下のような知見を得た。いくつかは既によく知られているが、あらためてここで述べる。

６．１．ScratchとMicro:bitの対比

Scratchはパソコン上で完結したプログラミング環境であるのに対して、Micro:bitは実機にプログラムを転送して動かせるところに特徴がある。

筆者らの印象であるが、プログラムの書きやすさではScratchの方が優れている。Micro:bitは、ブロックを組み合わせてプログラムを作ることよりも、文字ベースのプログラミング言語Pythonの学習に主眼があって、「回数を指定したくり返し」のブロックなどに見られるように、ブロックの仕様はあまり洗練されていないように感じる。

それに対して、Micro:bitは加速度センサや光センサなど、さまざまなセンサを搭載していることが魅力である。とくに、加速度センサがあるので、実機を揺らしたり（Shake)、基板の向きを変えたりといった児童の身体的動作に反応するプログラムが作れるが、これらの作例は児童の興味をとくによく掻き立てるようである。また、電気が導通するとオンになる端子がついているので、理科の実験などでも使えるだろう。

以上から、プログラミングを初歩から系統的に学習させたいならScratchが適しているが、コンピュータを物理的世界と相互作用させたり、コンピュータが我々の暮らしに組み込まれている様を体験的に理解させるにはMicro:bitが適していると言える。

６．２．例示作品の構想と新しいブロックの導入について

基本（本カリキュラムの１・２時間目）だけでも小学校におけるプログラミング授業の基礎固めという目標は達成できるので、図工的な作品作りは必須ではない。しかしながら、例示作品は教師の腕の見せ所である。

ここでは、例示品の構想に関して２点述べたい。

１つ目は、Micro:bitの３～５時間目で児童に３種類の作品を提示したが、これらを３名の大学生が個別に構想したものであり、結果的にバリエーションがあまりなかった。すなわち、どの作品も端子の導通に反応するタッチセンサを使い、加速度センサや光センサなどMicro:bitの特長を生かせなかったことがある。

同時に、モグラたたきと数の暗記ゲームは、どちらもLED画面に表示された数に応じてアルミ箔の電極にタッチするゲームであり、類似性が高い。

Scratchの６時間目に、フィジカルコンピューティングの作品としてMicro:bitを使った素振りカウンタを体験させたが、児童はこのデモに大変興味を示した。例示作品は、様々なセンサを搭載しているMicro:bitの強みをもっとよく生かして、かつ、例示作品を構想している時点で第三者の意見を聞いて、内容的に類似した作品群にならないようにするべきであろう。なお、事前に第三者の点検を受けることは、他のことについても有用である。教育実習の模擬授業と同じで、気兼ねなく厳しい意見も言ってくれる、信頼できる第三者に使ってもらって、その批判に耳を傾けるべきである。

２つ目は、このカリキュラムでは、できるだけ新規なブロックをできるだけ増やさず、いかに既習事項を組み合わせて作品を作っていくかに重点を置いた。新規なブロックを増やすと、そのブロックについて説明して、実際に使ってみて働きを確認させる必要がある。だが、後日あまり使うことのないブロックを、わざわざ時間をとって学習させることに意義が見出せない。

例示作品や課題の提案者は自分の構想に愛着があるので、その作品の実現のために新しいブロックを導入することを当然と考える。だが、あくまで授業として作品作りを行っているという観点からは間違っている。学習者の知識状態を前提として、今あるモノや知識では何が不足なのかをしっかり検討すべきである。その上で、そのブロックや例示作品を導入することの教育的コンピュータ科学的で客観的な理由を示せないなら（「自分は大事だと思うんです」でなく）、その例示作品は教材として不適格である。

６．３．キーボードからの文字入力について

キーボードからの文字入力は、２つの理由から、とくに作業時間がかかるものである。１つは、小学生の場合、そもそもキーボード入力に慣れていないことがある。さらに、ローマ字で日本語を表すことにも慣れていないので、十分な作業時間を取らなくてはならない。プログラミングの授業でシューティングなどのゲームを作らせることはよくあるが、そこでゲーム性を高める手段の１つは得点である。すなわち、何らかのアクションが成功すると得点できるという設定でプログラムを作らせるというものである。また、Scratchの場合、スプライトに何かメッセージを言わせることがあるが、そのときもメッセージの文を入力しなくてはならない。

２つ目の理由は、全角と半角（ANK）文字の切り替えである。変数名やメッセージに日本語（漢字、ひらがななど）を入力するときは全角で入力するが、ScratchもMicro:bitもブロックの引数の値（例えば、「10歩動かす」の10など）は半角で入力しないと、処理系が正しい値に解釈しない。したがって、何を入力するかに応じて、適切に全角と半角を切り替えつつ入力していかなくてはならないのだが、この判別と、実際に正しく切り替えることは小学生にはとても容易とは言えない。

以上から、キーボード入力は極力少なくすること、また、キーボードからの入力は半角に統一して、メッセージはローマ字または英語で入力させることを検討すべきである。理由は、ローマ字は３年生で学習することと、ライフ、スタート、フィニッシュなどの英語は、スペルは知らなくても、言葉自体は児童にもなじみがあるからである。５，６年生であればscoreに「スコア」とよみ仮名を振っておけば、変数の意味は分かるであろう。

６．４．工作を伴う授業の展開について

もし工作を伴う授業を行うのであれば、図画工作の単元と連携させて年間指導計画に位置付けることを検討すべきであろう。また、児童の人数が多くなった場合は、プログラミングはパソコン室、工作は図工室のように、部屋を別にすることが適当だろう。

授業時間については、本実践では、段ボールから形を切り出して、のりやテープでとめていって構造物を作りながら、デコレーションを加えていった。小学校の授業時間は45分であるが、このような工作は、35分ほどあれば「もぐら」を３匹作ってアルミホイルの電極をつけたり、イライラ棒を設置する台座を作ることができるので、２時間あれば作品作りはできると考えられる。

しかしながら、工作を行う場合、授業時間内で準備と後片付けをしなくてはならない。とくに、後片付け

は、作りかけの作品を壊さないように注意しながら格納場所に運び、残った材料やハサミやのりをしまい、作業台の上に何も残っていない状態に戻すので、場合によっては10分ほどかかることがある。これは授業時間が45分であることを考えると、見過ごすことのできない長さである。

時間割編成上、容易ではないことは理解しているが、工作を伴うプログラミングの授業は、連続した２時間で実施することを検討すべきであろう。授業時間が連続していれば、児童が前回の授業で何をしたか思い出す必要がないこと、後片付けが２回でなく１回で済むので、作業時間をより多く確保できることが挙げられる。

２つめの製作品の例示についてであるが、工作を指導するためには、指導者がその作品に精通していなくてはならない。そのためには、指導者が授業を実施する前に、例示する製作品を自分で作った経験が必要であろう。すなわち、自分で作ってみることで、制作を児童任せにせず、児童の足並みを揃えながら指導していくことが容易になる。授業時間が限られている場合は、このことは特に重要である。なぜなら、進行をあまり児童任せにすると、児童は構造物を作り上げることよりもデコレーションに凝り始めて、時間内に完成しない恐れが高くなるからである。また、実際に自分で作ってみることで、どのような工具や材料が必要か、何は児童に家から持ってきてもらうか、制作中に安全を確保するには何に注意するか、などの見当がつけられることも挙げられる。

文献・資料

MATHRAX［久世祥三＋坂本茉里子］(2019) プログラム×工作でつくるmicro:bit．オーム社

Micro:bit Educational Foundation (ND) micro:bitホームページ．https://microbit.org/

Microsoft (2020) MakeCodeホームページ．https://makecode.microbit.org/

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．日本教育工学論文誌，**34** (4)，387-394.

O'Sullivan, D. & Igoe, T. (2004) Physical Computing. Thomson

坂巻若菜、福島健介 (2017) 授業実践から考える小学校におけるプログラミング教育の課題・方向性．2017 PC Conference, 151-154.

サヌキテックネット (2017) micro:bit Lab．へようこそ．https://sanuki-tech.net/micro-bit/

スイッチサイエンスエデュケーション編集部 (2019) micro:bitではじめるプログラミング．親子で学べるプログラミングとエレクトロニクス．オライリー・ジャパン

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

高松基弘 (2018) micro:bitであそぼう！たのしい電子工作＆プログラミング．技術評論社

(ふるた　たかひさ・おくぎ　よしあき・たかひで　すずか　はすみ　たつき・わたなべ　あきら)