

## 幼児期プログラミング教育用教材の分析

柴田 雅博\*

**要旨** 2017年から2018年にかけて文部科学省により小学校、中学校、高等学校の学習指導要領が改訂され、問題解決のための論理的思考（Logical Thinking）力の修得に大きく舵が切られた。2020年度より小学校でのプログラミング教育必修化が始まり、中・高・大へと順々にプログラミング教育推進が進んでいくことになる。就学前の幼児教育においても2017年の新幼稚園教育要領にプログラミング的思考の修得を意識した文言が追加されているが、具体的な指導方法は明記されていない。本稿では、就学前の幼児期におけるプログラミング教育を想定し、現在販売・配信されているプログラミング教育を銘打った教材や知育玩具を分類し、幼児期のプログラミング教育教材のあり方を検討する。

**キーワード** プログラミング教育、論理的思考、プログラミング的思考、幼児教育

### 1. はじめに

2017年から2018年にかけて文部科学省により小学校、中学校、高等学校の学習指導要領が改訂された。今回の改定で大きく変わった点として、問題解決のための論理的思考（Logical Thinking）力の修得に大きく舵を切ったことがある。国際団体ATC21sによって定められたデジタル時代に必要となるリテラシー的スキルである21世紀型スキル<sup>[1]</sup>の一つが問題解決能力であり、論理的思考は問題解決能力のために重要なスキルとして定められており、文部科学省でもこの21世紀型スキルの育成に力を入れ

ている。また、論理的思考力を修得するためにプログラミング教育を取り入れ、プログラミング的思考を鍛えることで論理的思考力の獲得を目標と定めている。

また、昨今の急速なAI技術の発展に伴い、近い将来シンギュラリティが起これば、AIやロボットに大半の仕事を取られるといった話も広まっている。また、現在インターネットが社会インフラとして定着しており、今後、5GによりAI技術やIoTが社会や生活の基盤となっていくとも予想される。その一方、現在の日本は他の先進国と比べてAIやネットワーク技術において立ち遅れている感があり、今後の日本の

\* 福岡県立大学人間社会学部・講師

技術発展を担うAI人材の育成が急がれ、経済産業省などはそちらの観点からSTEAM教育に力を入れ、プログラミング教育を推進している。

このように、文部科学省、経済産業省の両省が連携して、学校教育では、初等、中等から高等教育に至るまで一貫してプログラミングの思考スキルを国語や数学のような基本リテラシーとして修得すべきという方向に動いている。そのような世の中の情勢の中で、就学前の幼児期についてもプログラミング教育へのニーズが生まれている。特に2020年度より小学校でプログラミング教育の必修化が始まり、幼小接続のために、幼稚園や保育園でもプログラミング教育を取り入れるところが現れ、また幼児向けプログラミングスクールや体験会のようなイベントも流行している。

同様に、書店や玩具店でもプログラミング教育を謳った書籍や知育玩具が多く販売されており、ICT関連の展示会においても多くのプログラミング教材が展覧されている。本稿では、幼児期のプログラミング教育を対象に、幼稚園や保育園、あるいは家庭にこれらのプログラミング教育用教材や知育玩具を導入することを考え、教育タイプや導入運用面での課題などを分析し、就学前幼児期のプログラミング教育用教材のあり方を検討する。

## 2. プログラミング教育とは

まずプログラミング教育とはどういうものであるかを紹介する。

### 2.1. プログラミング教育の目的

よく誤解されているが、プログラミング教育

は職業プログラマーの育成を目指すものではない。アプリケーション開発やコーディング能力の修得は教育の目的ではなく、あくまでも問題解決能力としての論理的思考力を身につけるための手段という位置づけである。

プログラミング教育と共に現在のホットワードがSTEM教育あるいはSTEAM教育である。STEMとは、科学 (Science), 技術 (Technology), 工学 (Engineering), 数学 (Mathematics) の4つの頭文字を取ったもので、これらを横断的、実践的、体験的に学ぶことにより、21世紀型スキルの中心である主体性、創造性、判断力、問題解決力を養うことを目的としている。またSTEM教育によりICT化グローバル化した社会に適応した人材の育成につなげるとして文部科学省・経済産業省・総務省が連携して推進している<sup>[2]</sup>。最近ではSTEMに芸術あるいはリベラルアーツ (Arts) を加え、STEAM教育としても紹介されている。

科学、技術、工学、数学は共にプログラミングと密接に関係がある。これらはプログラミングおよび情報科学の基盤となる知識やスキルであり、また応用や実践の場でもある。また、芸術としてはプロジェクションマッピングやVR絵画のようなICT技術を用いた新しい芸術が生み出され、リベラルアーツとしては人文科学とコンピュータ研究会のように人文科学の調査研究に積極的に情報科学的手法を取り入れると言った取り組みも進んでいる。

プログラミング教育はこれらSTEAM教育の一環として行われるもので、後に示すが、必ずしもコーディングを伴ういわゆるプログラミングを行うものだけではなく、様々なプログラミング的思考能力の育成が含まれている。

## 2.2. プログラミング教育必修化

文部科学省が2017年から2018年にかけて改定した小・中・高の各学習指導要領（新学習指導要領）<sup>[3][4][5]</sup>の中で論理的思考やプログラミング的思考を身につけるため、教育に本格的にプログラミングを取り入れる方針が発表された。この新学習指導要領の下で、2020年度より小学校でのプログラミング教育の必修化が開始された。また、2021年度からは中学校では教科「技術・家庭科」の中で制御・通信のプログラミングを取り入れることとなり、プログラミング教育の全面実施が開始予定である。高等学校では、すでに必修となっている教科「情報」に関して、これまで「社会と情報」と「情報の科学」のどちらか選択であったものを、2022年度からは必修科目「情報Ⅰ」と選択科目「情報Ⅱ」の二つに再編され、「情報Ⅰ」、「情報Ⅱ」ともに単元の中にプログラミングが含まれており、これにより高等学校でもプログラミング教育が必修化されることになる。

なお、今年度よりすでに開始されている小学校のプログラミング教育の必修化であるが、これは新たな専門教科を立ち上げるわけではなく、既存教科（「国語」、「算数」、「生活」など）のそれぞれにプログラミング的思考に対する単元を取り入れるという形を取っている。専門教科を立ち上げることなく既存教科の中でプログラミング教育を行うということで、授業計画を立てる小学校教員にとってはかなりハードルの高いものと予想される。しかし、コンピュータで実際にプログラムを組んでいくことだけがプログラミング教育ではない。たとえば、文献[6]は小学校低学年向けのプログラミングドリル教材であるが、この中の一例として、三色団子の完成品を見せて、団子と餡子をどの順番

に串に挿し、餡子を塗っていくかを考えさせるといったものがある。順序問題や論理パズル、繰り返しからルールを見つけるなど、既存の教材の延長でも論理的に施行するためのプログラミング的思考を鍛えることができる。

大学には小・中・高のような学習指導要領はないが、2019年の国立大学の改革方針により文部科学省では文系理系に係わらず、全学部でAIに関する教育を受けられるよう教育標準カリキュラムの開発に着手し、カリキュラム策定後はこれを公立大学、私立大学へも適用する予定である。

一方、幼児教育では、2017年の新幼稚園教育要領<sup>[7]</sup>において、新たに「幼児期の終わりまでに育ってほしい姿」として「思考力の芽生え」や「数量・図形、文字等への関心・感覚」などが示され、直接プログラミング教育と明記はされていないが、プログラミング的思考を意識した文言が追加されている。ただし、文献[8]で大久保らが指摘しているように、具体的にどのような指導を行うのか指示されているわけではない。大久保らは、新幼稚園教育要領の指導計画の作成上の留意事項に「幼児期は直接的な体験が重要であることを踏まえコンピュータなど情報機器を活用する際には幼稚園生活では得難い体験の補完幼児の体験との関連を考慮すること」とあることから、園内で直接コンピュータを使わせるのではなく、体験や遊びの中で論理的思考の素地を育成することにより小学校教育へ接続するものとしている。

## 2.3. プログラミング的思考の育成

プログラミング的思考とは問題をどのように解決するのかを考えることである。プログラミング的思考は、数学、論理学、統計学をはじめ

として、人文科学、社会科学、教養や常識的知識など、様々な能力を合わせた統合的技術であり、特定の一分野だけを鍛えるのではなく、横断的、総合的に修得することによって問題解決に活かすことができるものである。

太田らは諸外国のプログラミング教育および情報教育カリキュラムを調査した<sup>[9]</sup>。その中で英国の教科「コンピューティング」におけるコンピューショナルシンキングの概念を表1に示す。ここに挙がっているいくつかの能力を組み合わせて問題解決に取り組むことがプログラミング的思考であり、これらを鍛えていくのがプログラミング教育である。正確に言うとコンピューショナルシンキングとプログラミング的思考は完全に同じものではないが、非常に関連が深いものであるため、ここでは太田らの分類を基に、プログラミング的思考に必要な個々の能力を解説する。

### 2.3.1. 抽象化

抽象化は複雑な要素が絡み合った自然現象や社会現象などから、問題の本質に関係する部分だけを抽出し、単純なモデルに置き換えることである。たとえばサイコロを振るという行為を考えたとき、物理的に考えれば、手に持ったサイコロの初期状態、手を振る動作における角度や力の強さ、サイコロを放つ手の形状や角度や初速度、サイコロが落ちるまでの風の方向や

強さ、床や机の摩擦係数などが複雑に関連して最終的にある目が出る。しかし、それらをすべて計算機で計算することはとてもできないため、通常はこれを1～6のどれかの目が出る確率現象としてモデル化する。問題を解決するためには、問題を取りまく様々に複雑に絡み合う事象や環境の中から、問題解決に必要な部分だけを抜き出し、不必要となる部分は削除するということが重要である。余計な部分を取り除き、問題をできるだけシンプルな形態に作り替えることで問題の本質部分だけに取り組むことができるのである。これが抽象化である。特にプログラミングにおいては、グラフ化、数式化した数学モデルに問題を置き換えることが問題解決の前提となる。複雑な環境から問題の本質を見抜き、いかにシンプルな形にモデル化できるかが抽象化スキルである。

### 2.3.2. デコンポジション

デコンポジション (decomposition) 即ち分解は、大きな問題を小さな部分問題に切り分けることである。複雑に絡み合った問題を複雑なまま解くというのは難しいが、問題をいくつかの部分に切り分けることにより、個々の小問題は単純化され解きやすくなる。たとえば、グラフ理論における最短経路問題では、迷路上に構成されるグラフ構造においてスタートノードからゴールノードまで最短距離 (コスト最小) を

表1 英国の教科コンピューティングでのコンピューショナルシンキングの概念

(太田剛 他, 2016)

概念	概要
抽象化	問題を単純化するため、重要な部分は残し、不要な詳細は削除する。
デコンポジション	問題や事象をいくつかの部分に、理解や解決できるように分解する。
アルゴリズム的思考	問題を解決するための明確な手順で、同様の問題に共通して利用できるものである。
評価	アルゴリズム、システムの手段などの解決方法が正しいか、確認する過程である。
一般化	類似性からパターンを見つけて、それを予測、規則の作成、問題解決に使用する。

求める問題である。これを動的計画法では各ノードの入力枝のうち最小コストを持つ枝を選ぶという単純な問題に置き換えて解く。また、数学の証明問題には、複雑な証明をそのまま解くのではなくいくつかの補題に分け、補題を証明することにより本題の証明を図ることがある。このように、そのままでは手がつけられない問題を小さな部分問題に分けるということは問題解決手段として非常に有効である。ビジネスにおけるプロジェクトマネジメントなどでもプロジェクトの大目標を達成するために中間目標を立て、マイルストーンを置きながら全体計画を立てていくが、これもデコンポジションである。

### 2.3.3. アルゴリズム的思考

アルゴリズムとは問題の解くための手順のことである。アルゴリズム的思考とはこのアルゴリズムを構築することである。アルゴリズム的思考は最もコンピューショナルな能力といえる。コンピュータプログラミングにおける制御は、繰り返しと条件分岐の2つに集約される。問題の解き方は必ずしもコンピュータプログラミング的なものだけではないが、アルゴリズムの多くは繰り返しと条件分岐から成るプログラミングを想定した形式で作られている。つまり、アルゴリズム的思考は、問題をプログラミングで解ける形に置き換える能力と言ってもよい。

アルゴリズム的思考はある意味デコンポジションの逆ともいえる。アルゴリズム的思考とは小さな断片を使って大きな物を構成することにより複雑な挙動を実現する力である。小さな断片については既に解法が既知であり、どのような振り舞いをするのかを理解していることが前提である。このような断片だけから全体の完

成図を想像し、断片同士を組み合わせながら解法を求めていくことが重要となる。何をしたらどうなるのかという因果関係を想像し、適切な既存アルゴリズムを組み合わせ、自分が取り組むべき新しい問題の解決を図るのである。また、次に示す「評価」で問題解決に失敗したときに、失敗を分析し、どこが間違いだったのかを特定し、修正するといったトライアル&エラーで問題に挑む姿勢も重要となる。

### 2.3.4. 評価

評価とは、組み立てたアルゴリズムや解法が正しい解答を導き出せるかどうかを評価する能力である。プログラムの評価としては、主に3つの観点がある。

1つ目は正解に辿り着くことができるかというものである。当然ながら正しい答えに辿り着くことが目的であるから、正しく解答できないプログラムは評価外である。ただ、そもそも問題に正しい解答がないという場合もある。抽象化が間違っていて、定式化した問題そのものが間違っているという可能性もあるし、芸術のようにどれが正解か決められないという問題もある。評価は単にプログラムの間違いというだけでなく、問題設定などの評価でもある。

2つ目が停止するかどうかという観点である。繰り返し処理はプログラム制御の基本であるが、繰り返し処理には必ず停止することを保証することが求められる。繰り返し処理では永遠に止まらないという状態（無限ループ）に陥ることがあり、こうなると永遠に計算し続けるが解答が求められないという状態になる。極限問題のように解法が正しくても無限ループに陥る場合もあるため<sup>1)</sup>、いかに停止性を保証することも重要である。

3つ目は効率性である。同じ正解を導き出せ

るものでも、アルゴリズムによって高速に計算できるものから、無駄な計算を行って計算結果が出るまで非常に遅いというものがある。このアルゴリズムの効率性は等比級数的に効いてくる場合もあり、単順にスペックの高い計算機を使用すれば解決するという話ではないことも多い。いかに効率的に解答を導くかという観点が重要である。効率化はビジネスにおいては単純に時間的な話だけでなく、プロジェクト実行に掛かる予算や人的資源や配分などの効率化という視点が必要となる。

### 2.3.5. 一般化

一般化は個々の事例や連続した事象の中から、挙動や振る舞いに一定の法則性や類似性を見いだせないか観察し、見つけた法則性や類似性を問題解決に活かすことである。一般化は（共に逸話であるが）アルキメデスが風呂に入っていてアルキメデスの原理を発見し、ニュートンが落ちたリンゴから万有引力の法則を発見したように、科学的なアプローチとして古今東西の科学者が行なってきたことであり、古典論理学でいうところの帰納的アプローチに等しい。一般化スキルを鍛えるためには、観察力を養い推論能力を養うことが重要である。

また、統計分析における疑似相関や教師あり機械学習における過学習のように、帰納的に見つけた法則性や関連性がサンプルの少なさや偏りにより、本質から乖離しないよう注意深く検証することも重要である。

### 3. 幼児期プログラミング教育に対する教材

小学校のプログラミング教育必修化に伴い、ここ数年小学生のためのプログラミング教材、教育者対象のプログラミング教育のための事例

集や教材集、知育玩具類などが多数販売されている。また、民間のプログラミング教室やプログラミング体験会などのイベントも開催されている。同様に、小学校受験や幼小接続を見込んで就学前の幼児期のプログラミング教材も出ており、幼少期のプログラミング教材や知育玩具はちょっとした流行といってもいいだろう。

本稿では、特に就学前幼児を対象にいくつかのプログラミング教材や知育玩具を分析し、幼児期にふさわしいプログラミング教育のあり方について考察する。

プログラミング教育は、パソコン、タブレット、その他教育用コンピュータ端末などを使って行うこともできるが、それらがなければプログラミング教育ができないわけではない。コンピュータ端末を用いないアンプラグドな教材を使ってもプログラミング教育はできる。今回試してみたいいくつかの教材から、現在、配信・販売されている幼児期プログラミング教材を2つの観点により分類する。

#### 3.1. 動力による分類

1つ目の観点は動力に対する分類である。その分類を表2に示す。教材は大きくは電池やコンセントなどの電源を必要とするものとしなないものとに分かれる。また、電気を使う教材については、アプリケーションとしてPC上で使うもの、プログラミング学修教材として作られたマイコンボード、電子玩具として販売されているものに分かれる。

アプリケーションはコンピュータやスマートフォン、タブレットなどの一般のコンピュータ端末を使用するため、専用の機械を購入する必要はない。ビジュアルプログラミングなど直接プログラミングを取り扱うアプリケーションも

表2 幼児期のプログラミング教材の分類（動力による分類）

動力	概要
アプリケーション	コンピュータやタブレットなどのアプリケーション
マイコンボード	プログラミングができる電子基板教材
電子玩具	電子制御でき、電池などを必要とする玩具
アンブラグド	電池など電源を必要としない玩具

あれば、絵本作りやアニメーション作りなど情報教育と絡めたもの、パズルなどのゲーム形式で目的達成に挑むものなどがある。親の所有する端末などでも利用できるため比較的手軽に導入できるが、幼稚園等で園として用意するには金額面や管理面でハードルが高い。また、幼児にコンピュータ端末を使わせたくない親や教諭もいるという課題もある。

マイコンボードは、プログラミングを学ぶための教材として作られた小型の電子基板で、パソコン等につないでプログラミングを行うことができるというものである。有名なものとして英国BBCが主体として作った「micro: bit」<sup>2)</sup>がある。ボード自体は数千円と比較的安く、小学校用の教材として広まりつつあるが、就学前の幼児が扱うにはやや難易度が高いと思われる。

電子玩具は動力として電池やコンセントなどの電源を必要とし、電気で動く玩具である。電子制御やコンピュータチップ、センサーなどにより、利用者(幼児)が玩具の動作をコントロールできるものがプログラミング教育用玩具として多く販売されている。たとえば「キューベット」<sup>3)</sup>は木製の車の玩具で、専用のボードに前に進む、右に曲がる、左に曲がるなどのコーディング用ブロックをはめ込むことで車の動作をプログラミングすることができるというものである。また、「Ozobot」<sup>4)</sup>はセンサーにより床(机など)に描かれた線を辿って移動する車

の玩具で、線の色の組み合わせで、スピードの上げ下げ、左右に曲がるなどの動作制御が行なえるというものである。電子玩具の価格は材質やできること、センサーやコンピュータの有無などに応じて数千円から数万円まで幅広く、数万円のは家庭で購入するにはややハードルが高い。

最後が電源を必要としないアンブラグドな教材である。主には、積み木やブロックなど組み立てることを目的としたもの、ボールを使って高いところから低いところまで転がしていくような位置エネルギーを使ったもの、カードやボードゲームなど、テキストやノートなどがある。これらについては、プログラミング教育を謳いながらも、既存の玩具と同様かそれに近いものなので、導入への抵抗は低いと考えられる。

### 3.2. 目的タイプによる分類

2つ目の分類が目的タイプによる分類である。これは各教材に設定されている教育目的のタイプによって分類したもので、その結果を表3に示す。次にそれぞれの目的タイプについて説明する。

#### 3.2.1. プログラミング型

プログラミング型教材は、実際にパソコンなどを使って、コンピュータプログラムを書くものである。表2の分類における「アプリケーション」や「マイコンボード」を用いて行うこ

表3 幼児期のプログラミング教材の分類（目的タイプによる分類）

目的タイプ	概要
プログラミング型	コンピュータで実際にプログラミングを行う。
物作り	ブロックや積木など、物体を構築する。
お話作り	断片的な手がかりからストーリーを作る。
目的達成型	目的を設定し、その目的を達成するための工夫を行う。
ボードゲーム・カードゲーム	ゲームに勝つための戦略を考える。

とが主流である。プログラムを書くと言っても、C言語やPythonなどのCUI型のプログラミングではなく、Scratch<sup>5)</sup>を代表とするビジュアルプログラミング言語であることが多い。ビジュアルプログラミングは、テキストエディタにコマンド形式のプログラムを入力するのではなく、「〇回繰り返す」、「〇〇ならば△△を、そうでないなら××を実行」といった制御ブロックを組み合わせて、視覚的にプログラミングができるもので、スペルミスなどのsyntax errorのデバッグで、子ども達がプログラムを実行する手前で挫折しないようにできている。Scratchの対象は主に小学生以上であるため、就学前幼児期にはより直観的なViscuit<sup>6)</sup>などを用いることもある。プログラミング型の教材は、プログラムを組むことを目的としており直接的なプログラミング教育アプローチである。ただし、プログラミングである以上、幼児にいきなりこれを使って遊ばせるのにはやや無理があり、基本的な操作やプログラミングについてのレクチャーが必要となる。そのため、実際はプログラミングスクールや小学校の授業の中で講師や教員をつけての講習で使用されていることが多く、子ども一人や親子と一緒に日常的にプログラミングを行う習慣が付くかといふとなかなか難しいと感じる。

### 3.2.2. 物作り

物作り形の教材は、積み木やブロックなどを使って目的のものを組み立てるというものである。完成品をイメージしながら適切な部品を選び、つなぎ合わせていくという行為は空間認識能力を高めると共に、2.3.3節で示したアルゴリズム的思考を育てるのに有効である。積み木であれば重心を保ちながら下から上に積み木を積み上げていかないと途中で崩れてしまい、ブロックであれば接続を考えながらパーツを組み立てていかないと、パーツ同士の幅が合わずに接続できなかつたり、上下左右のバランスが悪くなつたりする。また、車輪パーツを使って自動車や列車を作るような場合には左右の高さを合わせないと斜めに傾いた自動車が出来上がつたり、接続部が弱いとブロックの重さに耐えきれずすぐに壊れたりするかもしれない。ブロックの中には、磁石の入った四角形や三角形の板を磁力でくっつけるといった方式のものも販売されている。また、ブロックとコンピュータ機器を組み合わせたものもある。Legoは以前からLego Education<sup>7)</sup>としてレゴブロックで組み立てたものをプログラミングで動かすことのできる製品を販売している。代表的なものがLegoマインドストームであり、本体となるインテリジェントブロックにプログラムを転送し、ブロックで作成した部品を動か



すというものである。LegoマインドストームはScratchなどを使うこともできるが、Pythonを使ってかなり本格的なプログラミング制御も可能であり、就学前の幼児に使わせるにはまだ難しい。幼児用にデュプロ®プログラミングトレインセットも用意されており、こちらはレールをつなげて線路を作り、その上を列車が走るというもので、Ozobotのように色ブロックにより停止、進行方向の反転、汽笛などの列車の動作制御を行うものである。「学研のニューブロックプログラミング」<sup>8)</sup>はブロックを組み立てて自動車を作り、それをコンピュータ制御で動かすというもので、コクピットパーツに制御用のクルーパーツを乗せ、載せたクルーの順番に車輪パーツが動作する。「Osmo」<sup>9)</sup>はiPadにカメラ用のパーツを設置し、カメラに写されたプレイエリアの物体をコンピュータに取り込むことができるというものである。作成したタングラムをiPadのカメラで映し、その形状を読み取ると言った遊びができる。Osmoでは、ほかにも命令ブロックを組み合わせたプログラミング制御やアルファベットブロックで単語作りなどいろいろなことができる。

### 3.2.3. お話作り

お話作りは、与えられた手がかりからストーリーを作成したり、場面を正しい順番に並べたりするというもので、創造性などの情操教育と共に、因果関係の理解、文章構成能力を育てることができる。「プログラミン」<sup>10)</sup>は文部科学省の提供するWeb上で動くプログラミングアプリケーションで、キャラクターをビジュアルプログラミングで動作させ、ストーリーやアニメーションを作るというものである。Webアプリケーションとして提供されており、お手本

も充実してコンピュータアプリケーションとして提供されている教材としては非常に使いやすいのだが、Flash Playerの提供・サポートの終了と共に、2020年12月でサービスを終了した。iPadを使ったものとしては「ピクエのつくるえほん」<sup>11)</sup>がある。これはキャラクタや背景を挿入し、地の文やセリフをつけデジタル絵本を作るというもので、BGMを入れたりナレーション音声を入れたりといったこともできる。また、プリントアウトして紙の絵本として製本することもできる。「100てんキッズ お話づくり絵カード」<sup>12)</sup>は場面が入ったカードを組み合わせて正しい時系列に並べストーリーを作らせるというものである。カードの裏には正解のストーリーが書かれているが、それを使わずに独自のストーリーを考えさせるといったこともできる。「ローリーズ・ストーリー・キューブス」<sup>13)</sup>は、サイコロ状の六面体キューブに絵が描かれており、最大9個のキューブを振って出た目の絵を組み合わせるとストーリーを作っていくというもので、与えられた断片からストーリーを構成する力を鍛えることができる。

### 3.2.4. 目的達成型

目的達成型は、特定の目的を決め、それを達成するためのプロセスを組み立てていくというものである。目的達成のためにその解決方法を模索するというのはプログラミングそのものである。たとえば、数独やクロスワードパズルといったアナログゲームあるいはコンピュータゲームとして、パズルゲームが多く作られているが、それらも目的達成型玩具の一つといえる。パズルゲーム以外のものとしては、ボールを転がしゴールにたどり着くタイプのものが多い。くもん出版の「NEWくみくみスロープ」<sup>14)</sup>はブロック形式のパーツを組み合わせるとコース

を作り、高い位置のスタート地点から低い位置にあるゴールまでボールを転がすというもので、立体的な空間認識能力やスロープの組み合わせ、ボールの転がり方（下り坂を転がることはできるが上り坂を作ると転がることはできないなど）を駆使してコースを作る必要がある。「ころがスイッチ ドラえもん」<sup>15)</sup>も同様にボールをスタートからゴールまで到達させるようコースを作ることを目的とする。基本的には高い位置から低い位置にボールを転がしていくのだが、この製品では「ひみつ道具スイッチ」と呼ばれる特殊パーツにより、ボールの動作を制御することができるのが特徴である。たとえば「どこでもドア」はボールにより片方のスイッチが押されたらもう片方のドアが開く仕組みで、あらかじめドアに収納された別のボールが転がり始めることでボールが移動したように見せることができる。「エスパー帽子」は1つ目のボールが通過するとコースを変え、2つ目のボールを別のコースに送り出すことができる。このようにギミックを工夫してゴールへのルートを組み立てることができる。

### 3.2.5. ボードゲーム・カードゲーム

アンラダグなSTEAM教育教材としてボードゲームやカードゲームも注目されている。ゲームはシンプルなルールの下で、目的を達成したり勝利を目指したりするもので、すでに標本化が終わった状態から問題解決に挑むことができ、また遊びながら攻略法を練っていくなど、プログラミング的思考を育てるのに非常によい教材である。実際、プログラミングを学習する際にはゲーム製作を題材にすることも多い。ボードゲームやカードゲームは様々なものが販売されている。ただし、プログラミング的思考を鍛えることを目的とするなら双六のよう

な運の要素を含まないゲームの方がよいと考えられる。たとえば、ペグソリティアは古くからあるゲームで、隣にある駒を飛び越えたら飛び越えた駒を取り除くというのを繰り返して、盤面の駒を1つ残してすべて消していくというものである。「ROBOT Turtles」<sup>16)</sup>はGoogleのプログラマーが開発したボードゲームとして有名で、カードを使って亀を移動させ、宝石までたどり着くことを目的とするものである。こちらはプログラミング教育を意識して開発されており、非常にプログラミングに近い形のゲームデザインである。

### 3.3. 考察

3.1節、3.2節に見られたように、プログラミング教育用の教材や知育玩具として様々な物が開発、販売されている。この中で、就学前幼児期のためのプログラミング教材としてはどのようなものが相応しいだろうか。まず幼稚園や保育園の中でパソコンやタブレットを用意し、アプリケーションを使わせるのは、導入費や維持費、管理面の手間などを考えるとまだまだ難しいだろう。園長や教諭としても、園内にICT端末を導入し、園児に使わせることに精神的な抵抗がある人も多い。また、小学校であれば授業やクラブ活動でビジュアルプログラミングの講習を行うこともできるが、幼稚園や保育園ではイベントとしてゲスト講師を呼んで講習会や体験会を開くといったことはできるかもしれないが、外部講師を招くのは単発あるいは2～3回の短期でしかできず、長期的にプログラミング教育を施すということは難しい。持続可能性の面でも、教諭が園児に直接指導できないものを導入するのは難しい。

それを考慮すると、幼稚園や保育園に置くも

のとしては、園内に玩具を置いておいて、子ども達が好きに遊べるような教材がよいと考えられる。積み木やブロックなどはすでに置いているところも多いので、そこから派生した3.2.4で示したようなプログラミング的思考を意識した知育玩具を導入するのがよいのではないかと考えられる。またカード式のお話作り玩具や文献[6]のようなドリル形式の教材は、教諭が勉強の時間に利用しやすい教材であると言える。

一方、家庭においては、自宅にあるPCやタブレットを親子で一緒に使うのであればアプリケーション型のプログラミング教育教材を利用することができる。ViscuitやScratchを使うためには親が操作を勉強しておく必要があり、チュートリアルもあるので慣れれば使えるようになると思われるが、プログラミングスクールや体験会への参加が必要になるかもしれない、ややハードルは高い。Osomoのようにプログラミングを意識せずに使えるものもあるがやや金額が高い。「ピッケのつくるえほん」のようなお話作りや情操教育教材、パズルゲームなど、探せば多種多様なものが見つかるが、ピンからキリまであるので適切なものを選ぶのが大変である。

ボードゲームやカードゲームなどアンプラグドな教材や玩具は親子で一緒に遊ぶのにはよいが、小学生以上を対象にしているものが多く6歳未満の子どもが対象年齢になっているものは意外と少ない。ルールがシンプルで、かつ子どもの興味が持続し飽きさせないものを選ぶ必要がある。ローリーズ・ストーリー・キューブスはルールがシンプルで親子のコミュニケーションも図れるため、今回見た玩具の中では導入しやすいものであると考えられる。

#### 4. おわりに

本稿では、就学前の幼児期にプログラミング教育を行う際に用いるプログラミング教材について、既存のプログラミング教材や知育玩具を分析しタイプ別に分類した。動力による分類では、パソコンやタブレット上で動く「アプリケーション」、プログラミング教育用に開発された「マイコンボード」、電池で動く「電子玩具」、動力を必要としない「アンプラグド」に分かれる。また目的タイプ別による分類では「プログラミング型」、「物作り」、「お話作り」、「目的達成型」、「ボードゲーム・カードゲーム」に分かれる。幼稚園や保育園に導入するものとしては、導入・維持コストや教諭の負担を考え、子どもが好き勝手に遊べるような、積み木やブロックそれに類した物作りや目的達成型の玩具を置くのがよいと考えられる。家庭に導入する場合には、アプリケーション型の教材であれば自宅のPCやタブレットにインストールしたりWeb上で利用できたりするため、比較的手軽に導入できる。また、お話作りは因果関係の把握や構成能力を高めると共に、情操教育や親子のコミュニケーションにとっても有効であると考えられる。一方、本格的なプログラミング型教材の導入は、講師やインストラクターを必要とする場合が多く学修の持続可能性の面でもややハードルは高い。

今回は、著者が教材や玩具を触って感じたものを主軸に幼児期のプログラミング教材のあり方を考えたが、実際に幼稚園や保育園の教諭、子どもや親の意見も調査する必要があり、今後の課題である。

## 謝辞

本研究はJSPS科研費 JP19K03060の助成を受けたものです。

## 注

- 1) ただし、実際問題としては、計算機の場合、通常は丸め誤差により途中で微小計算が不能になって停止する。
- 2) <https://microbit.org/>
- 3) <https://www.primotoys.jp/>
- 4) <https://www.ozobot.jp/>
- 5) <https://scratch.mit.edu/>
- 6) <https://www.viscuit.com/>
- 7) <https://education.lego.com/ja-jp/>
- 8) <https://www.gakkensf.co.jp/nbsl/nbpg/>
- 9) <https://www.playosmo.com/ja/>
- 10) <https://www.mext.go.jp/programin/>
- 11) <https://www.pekay.jp/pkla/ipad>
- 12) <https://www.gentosha-edu.co.jp/book/b341539.html>
- 13) <http://www.rorysstorycubes-japan.com/>
- 14) <https://www.kumonshuppan.com/kumontoy/kumontoy-syousai/?code=54474>
- 15) <https://toy.bandai.co.jp/series/doraemon/corogaswitch/>
- 16) <http://www.robotturtles.com/>

## 参考文献

- [1] P.グリフィン, B.マクゴー, E.ケア, 三宅なほみ (監訳), 益川弘如 (訳), 望月俊男 (訳)『21世紀型スキル: 学びと評価の新たなカタチ』, 北大路書房, 2014.
- [2] 新たな時代を豊かに生きる力の育成に関する省内

タスクフォース, 『Society 5.0に向けた人材育成～社会が変わる、学びが変わる～』, 文部科学省, 2018.

- [3] 文部科学省『小学校学習指導要領』, 2019.
- [4] 文部科学省『中学校学習指導要領』, 2019.
- [5] 文部科学省『高等学校学習指導要領』, 2019.
- [6] 石戸奈々子 (監修)『7さいまでに楽しくおぼえる論理的思考力を育てるプログラミングれんしゅうちょう』, 学研, 2019.
- [7] 文部科学省『幼稚園教育要領』, 2017.
- [8] 大久保淳子, 坂無淳, 柴田雅博「就学前のプログラミング的思考の育成カリキュラムの開発」, 国際幼児教育学会第41回大会, オンライン開催, 2020.
- [9] 太田剛, 森本陽介, 加藤浩「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—」, 日本教育工学学会論文誌, 40(3), 197-208, 2016.