Vol.2015-CE-131 No.6
2015/10/10

小学生向けプログラミング学習教材「まなべー」の開発と教育効果

間辺 広樹[1]　並木 美太郎[2]　兼宗 進[3]

概要：画面上のロボットに動作を指示する形の小学生向けプログラミング学習教材「まなべー」を開発した．児童らが無理なく取組めるように，使える命令と制御構造を制限し，ゲーム性を持たせた入門用教材である．日本語の簡単な命令が書かれたカードを並べるだけで短時間に，『コンピュータはプログラム通りにしか動かないこと』や『確認しながら進めるプログラムの作り方』を学べるようにした．小学生を対象とした科学イベントと，高等学校の情報科の授業で行った実験結果から、教材の適用範囲と活用法を考察する．

キーワード：コンピュータサイエンスアンプラグド, プログラミング教育，トレース，デバッグ

The Development and Educational Effect of Manabē, A Programming Learning System for Elementary School Students

Manabe Hiroki[1]　Mitaro Namiki[2]　Kanemune Susumu[3]

Abstract: We developped a programming learning material for elementary school students called 'Manabē'. Its learning style is to order to a robot some movements. On Manabē, students can program on computer screen by ordering cards written with simple Japanese words. Moreover, they can play like a game. We restricted the instructions and control structures so that students can learn programming easily. We tested it on a scientific event and an informatics lesson in a high school. This paper reports the results of these tests and how we can use this in schools.

Keywords: Computer Science Unplugged, Programming Education, Trace, Debug

1. はじめに

情報技術が社会や生活に溢れている現在，コンピュータやネットワークの可能性を引き出し，よりよい社会や生活を創造的に生み出すために，プログラミング教育の重要性は増加している．

平成 25 年 6 月には世界最先端 IT 国家創造宣言 [1] が閣議決定された．その中で，IT 利活用の裾野拡大を推進するための基盤の強化として，「初等・中等教育段階におけるプログラミングに関する教育の充実，IT を活用して多様化する課題に創造的に取り組む力を育成，発達段階に応じた指導案や教材の蓄積」を求めている．プログラミング教育とその教材の充実は，今や国家レベルの課題である．

一口にプログラミング教育と言っても「作品作り」「コンピュータの仕組みの理解」「アルゴリズムの実現」「言語の理解」など力点は様々である．そのため，授業の目的や学習者の年齢等に適した教材が選べるように，選択肢を増やすことが必要である．

これまで小学校の授業や児童向けプログラミング教室などでは，自分の作りたいものを自由に作る「作品作り」に重きを置いた実践が多く報告されてきた [2][3][4]．それに伴って，児童のアイデアを引き出し，それを容易に実現できるプログラミング言語が選択されてきた．Viscuit[5]，Scratch[6]，ドリトル [7] がその代表である [8]．

本研究では主に小学生に対して『プログラム通りにしか

[1] 神奈川県立柏陽高等学校
[2] 東京農工大学
[3] 大阪電気通信大学

ⓒ 2015 Information Processing Society of Japan

動かないことコンピュータの仕組み』と『確認しながら進めるプログラムの作り方』を学ばせることを重視した教材を提案する．

小学生でもコンピュータの仕組みを学べるように工夫されたコンピュータサイエンスアンプラグド [9](以下，CS アンプラグドと記す) には「プログラミング言語」という学習項目がある．これは，与えられた図形を人の言葉だけで伝えるという活動を通して，プログラミング言語の基礎概念を学ばせるものである．この学習には，「実際のコンピュータでは」として，前述したドリトルで確認する活動が紹介されている．ドリトルは日本語で入力できることから，この学習に相応しいプログラミング言語である．しかし，日本語入力のおぼつかない児童にとっては，敷居が高い．本研究の狙いは，その部分の補完である．

そこで，単純な日本語が描かれたカード（タイル）を並べるだけでロボットを動かす仕組みを提案する．複数のコースを用意し，児童にコースに合わせたロボットの動作を考えさせる．途中で誤りがあっても動作してしまうように作れば，学習者にとっては予測しなかった動きになるため，『プログラム通りにしか動かない』ことへの意識が高まると考えられる．また，ロボットの動きをコントロールするために，学習者は頭の中でロボットの動きをイメージする必要があるが，動きが複雑になると，頭の中だけで考えることに限界がある．そこで，適宜トレースしたり，動作確認をしながらプログラムを作る作り方を学ぶことが期待できる．トレース指導の重要さは江木ら [10] が指摘しているが，児童・生徒を対象としたプログラム作りで，その点を意識させることはこれまで議論されてこなかった．

命令カードを並べてロボットを動かす仕組みには，アルゴロジック [11] が知られているが，アルゴロジックは，元々高校生向けに開発されたものであるから，小学生には難しい．また，プログラムが間違っていた場合に，間違えたことは教えてくれるが，それによってロボットがどのような振る舞いをしてしまうのかは示さない．

以上を踏まえ、本研究にて教材を開発し，小学生と高校生を対象とした実験授業で教材の効果と適用範囲を明らかにする．

2. プログラミング学習教材「まなべー」の仕様

本研究で扱う教材は，プログタミング学習の導入段階で使うものであるから，プログラミング嫌いを作らないためにも楽しさや達成感を得られるような，課題解決型のゲームに仕立てることとした．そこで，以下の 3 点を指針として開発することとした．

(1) 楽しみながら学習できる (達成感を得られる)
(2) コンピュータの仕組みを学べる
(3) プログラムの作り方を学べる

また，作品作りを行うわけではないので，多くの機能は必要ない．そこで，命令は「1 まえ」「3 まえ」「みぎ」「ひだり」「ひろう」の 5 種類に制限する．また，制御構造は順次構造のみを用いることとする．そこで，開発したのがプログラミング学習教材「まなべー」である (図 1)．

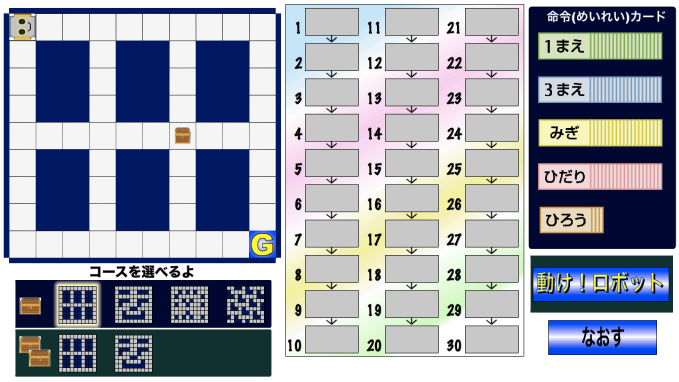


図 1 まなべーの画面とコース 1
Fig. 1 Screen Shot of Manabē

まなべーを起動すると，画面左側に「窓」のような形をしたコースが表示される．これをコース 1 と呼ぶこととする．その左上にロボット (の顔) が配置されている．「1 まえ」「3 まえ」「みぎ」「ひだり」「ひろう」という命令がそれぞれカードの形になって画面右側に置かれている．ロボットは白い道の上を歩くことはできるが，着色された部分 (紺色) は壁となるので，歩くことができない．そこで，カードを画面中央の連番が示されたプログラム記述部分の枠へとドラッグ操作で並べていく．これが手順を書くことと同等の操作になる．この「プログラミング」によって，ロボットの動作と歩く道順が決まる．コースの途中には宝箱が置かれているので，その位置に来たらロボットに宝箱を拾わせる命令「ひろう」を入れる必要がある．プログラムを実行するには，画面の下部にある「動け！ロボット」と書かれたボタンを押す．ボタンを押すと，カードの並び通りにロボットは動く．「1 まえ」や「3 まえ」は，ロボットが向いている方向に軌跡を描きながら枠の数だけ移動する．「みぎ」や「ひだり」は，その方向へ 90 度回転する動きとなる．プログラムが正しく作られていれば，無事にゴールし，「おめでとう！任務完了だ」と書かれた吹き出しが表示される．ただし，ゴール時に宝箱を持っていないと「おい！宝箱はどうした？」と表示される．

プログラムの実行時は，カードの追加・修正やコースの変更はできない．プログラムの実行が完了すると，画面右下に「なおす」ボタンが表示される．これを押せば，プログラムの編集やコースの変更が可能になる．この実行 (「動け！ロボット」) と編集 (「なおす」) を繰り返して，ロボットに任務を完了させる．

プログラムの失敗例と成功例を図 9 に示す．失敗例ではプログラムの 7 番で「みぎ」と「ひだり」を間違えている

ⓒ 2015 Information Processing Society of Japan

ので，ロボットはゴールと反対の方向に移動した．また，壁に衝突しているので，ロボットの顔が歪んでいる．成功例ではその 2 点を修正したので，宝箱を持参したロボットが無事にゴールしている．

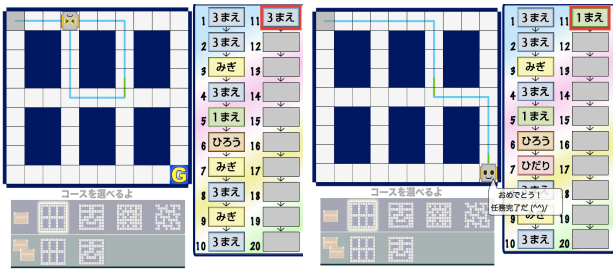


図 2 成功例 (左) と失敗例 (右)
Fig. 2 Each example of Success(left) and Failure(right)

コースはコース 1 の他に，コース 2，コース 3，コース 4 を合わせて 4 通り用意した (図 3)．



図 3 コース 2 (左), コース 3 (中央), コース 4 (右)
Fig. 3 Course2(left), Course3(center) and Course4(right)

以上が，プログラミング学習教材「まなべー」の概要である．「まなべー」は Flash で制作されているため，通常の Web ブラウザと Flash プレーヤーさえあればパソコンでも，タブレット端末でも動作する．画面の縦横比は，科学イベントで用いることを想定したため，その際に使うタブレット端末の縦横比に合わせた．

3. 実証実験

3.1 実験 1：小学生対象の科学イベント

実験 1 は，情報オリンピック日本委員会と (株) 富士通が開催している富士通キッズイベント 2015「夢をかたちにするしくみ」で行われた．同イベントは，2008 年より CS アンプラグドの学習をイベント向けにアレンジし，CS アンプラグドの実践経験が豊富な教員等を講師に迎えながら，実験的な学習活動を実施してきた [12]．2015 年は前述した CS アンプラグドの「プログラミング言語」が対象の学習であった．Web ページやメールマガジンにて参加者を募り，保護者同伴のもとで実施された．午前の部に 32 名，午後の部に 35 名の合わせて 67 名の小学生が参加した．

会場は 5 人〜6 人の子供が一緒に作業するように机が並べられ，それぞれの机には補助講師が 2 人ずつ配置された．全体の流れはメイン講師がコントロールした．各机には子供の数だけタブレット端末が置かれ，描いた絵を一斉に表示するシステムも作られた．以上の環境を踏まえて，45 分間の学習として実施された (表 1)．

表 1 富士通キッズイベントの学習の流れ
Table 1 Lesson Plan of Fujitsu Kids Event

活動	内容	教材等	時間
1	導入	メイン講師の説明	3 分
2	言葉の指示で絵を描こう	CS アンプラグドの活用	12 分
3	ロボットに命令してみよう	特製ロボットによる寸劇	15 分
4	コンピュータで確かめよう	まなべーの活用	10 分
5	まとめ	メイン講師の説明	5 分

まず，活動 1 では，「導入」としてメイン講師が「自分がコンピュータになって考える体験を通してコンピュータに命令すること（プログラミング）について考える」という学習内容を説明した．

活動 2 では，CS アンプラグドの「プログラミング言語」をアレンジして「言葉の指示で絵を描こう」という活動を行った．これは，会場から選ばれた 1 人の子供が、例示された絵について「真ん中に四角を描きます」などと言葉で指示 (これがプログラムに相当する) して，絵を見ていない他の子供にタブレット端末に描かせるという活動であった．作られた作品は，収集をして中央のモニターに表示し，「正確に伝えることの大切さ」などを学ばせた．(図 4)．

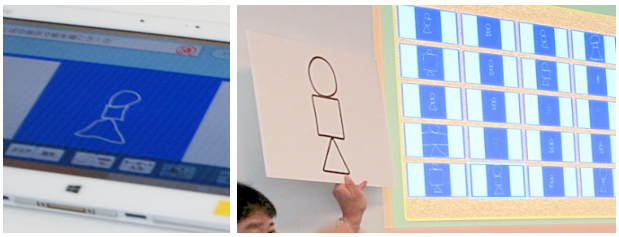


図 4 「言葉の指示で絵を描こう」の作品例 (左) と全員の作品 (右)
Fig. 4 Students' Drawing Sample

活動 3 では，「ロボットに命令してみよう」という活動を行った．これは，メイン講師や会場から選ばれた子供が，人形のロボットに命令して動作させる．プログラミングの概念をコンピュータの画面の中だけでなく，現実社会と関連を意識させることが狙いである．活動は，イベントらしく寸劇風に仕立て上げた (図 5)．

活動 4 では，「コンピュータで確かめよう」という活動を行った．活動 3 で用いた命令で動作するロボットを，まなべーを使い，1 人 1 台のタブレット端末で使う活動であった (図 6)．使える時間が限られていたため，活動 4 では 10 分間という短い時間にいくつのコースを攻略できるか，が課題となった．

活動 5 では，「まとめ」としてメイン講師が，それまでの活動が「実際に使われている情報技術とどのように結びつ

ⓒ 2015 Information Processing Society of Japan



図 5 子供の言葉で動く「ロボット」
Fig. 5 Controled 'Robot' by A Child's Voice



図 6 まなべーを用いたイベントでの活動
Fig. 6 Activity on Manabē

くのか」を説明し，実際に使われているプログラミング言語を紹介した．

以上が，科学イベントとして企画された「ロボットに動きを教えよう」であった．最後に，児童と保護者に対して「楽しさ」や「満足度」について，アンケート調査を行った．

3.2 実験２：高等学校情報科の授業

高等学校での実験は，高等学校を含めた他校種での利用価値を考察するためである．高校生にとってはまなべーは難易度の低い教材であるが，それでもプログラミング的な考え方の乏しい生徒には，馴れるまでの間，難しさを感じる可能性もある．その場合，教材としての利用価値があると言える．また，高等学校で利用できれば中学校でも利用できる可能性がある．

実験は，H 高校で「情報の科学」を学ぶ 1 年生 39 名を対象とし，パソコンを 1 人 1 台使って行った．H 高校は理系進学希望者が多く，論理的な学習を好む生徒が多い高校である．中学校の技術家庭の授業を中心に，25 名がプログラミング経験者であったが，その中の 11 名はそれを「プログラミング」と認識できてなく，経験というより体験というレベルの学習歴であると推測された．

授業は 50 分で実施した．プログラミング経験に関する簡単なアンケート調査を行った後，教師が一人一人にまなべーの実行 f ファイルを配布し，4 種類のコースそれぞれにゴールを目指すように指示した．

実験に際して，生徒にはプリントを配布し，コース毎に「完了までに要した時間」と「使用した命令カードの枚数」を記録させた．また，プログラミング作りの難しさについて，5 段階 (1. 超簡単！　2. 簡単　3. 少し難しい　4. かなり難しい　5. 超難しい！) の中から 1 つを選択させた．また，すべて終わったところで全体の『難しさ』をコースと同じように 5 段階から，また，『楽しさ』を 4 段階 (1. とても楽しかった　2. 少し楽しかった　3. あまり楽しくなかった　4. つまらなかった) から選択させた．最後に，「ロボットを正しく動かすために，何に気を付ければよいか」を自由記述させた．

開始当初はどの生徒もカードをゆっくりと動かていたが，時間の経過と共に速くなった．途中までの動作を確認しながら慎重にカードを並べる生徒と，一度に何枚ものカードを並べる生徒もいた．一度に多くのカードを並べる場合，途中の命令カードを間違えるとロボットが予期しない振る舞いをする．その際に，誤り箇所が特定できず，戸惑う生徒もいた．高校生には簡単な課題であることから，実験前は真面目に取り組むかどうかが危惧されたが，どの生徒も真剣に課題に向かっていた (図 7)．また，生徒同士でロボットを動かしたコースを見せ合ったり，カードの枚数を少なくするための命令カードの使い方など，生徒同士で議論し合う光景が教室内の複数箇所で確認された．



図 7 まなべーを実験的に活用した高等学校情報科の授業
Fig. 7 Trial of Using Manabē on Informatics Lesson in a High Schooll

4. 実験結果

4.1 科学イベントの実践から

イベントの中の活動 4「コンピュータで確かめよう」の様子を示す．メイン講師が 1 分程度でまなべーの使い方を説明すると，児童はすぐに操作に取り掛かった．高等学校と同様に，慎重に確かめながら進める児童と，あまり確かめもせずに一度に多くのカードを並べる児童に分かれた．ロボットが無事にゴールして喜んだり，おかしな動き方をしたことに慌てるなど，イベント会場らしくにぎやかな雰囲気になった．「ロボットがかわいそうだから」と，壁にぶつからないように気をつけてカードを並べる児童もいた．活

ⓒ 2015 Information Processing Society of Japan

動 4 は 10 分間と限られていたため，多くのコースを攻略できず，終了の合図に不満を口にした児童もいた．また，その後の休憩時間に続きをやりたいと申し出た児童も多かった．イベント後のアンケートでも，「楽しかったし，勉強になった」と答える保護者，児童が多く，「ゲームを通して学習できたのでわかりやすかった」「いろいろなプログラミング言語を学びたいと思いました」などと肯定的に授業や教材を受け止める声が多かった．

思考過程の分析は操作画面の録画によって行った．録画できない端末もあったが，36 名中 31 名分について，各児童の操作の一部を記録することができた．その 31 名を分析の対象とした．学年と性別は表 2 の通りである．

表 2 分析の対象となった児童 31 名の構成
Table 2 Information of the Students for Evaluation

学年	男子	女子	合計
4 年生	6 名	4 名	10 名
5 年生	3 名	5 名	8 名
6 年生	10 名	3 名	13 名
合計	19 名	12 名	31 名

活動は個人差が大きく，コース 1 しかできなかった児童もいれば，すぐに要領を得てすべてのコースをゴールした児童もいた．一度に配置したカードの枚数が多過ぎて，誤り個所を特定できず，修正を繰り返す児童もいた．その一例を図 8 に示す．ある児童は一度に 9 枚のカードを配置して一気にゴール目指した．しかし，コース中央で宝箱に向かうための「ひだり」が抜けているため，そのまま直進してコースを外れた (図 8 左)．児童はそのことに気付き，途中に「ひだり」を入れたが，今度は曲がる位置を間違えているため行き過ぎて壁にぶつかった (図 8 右)．

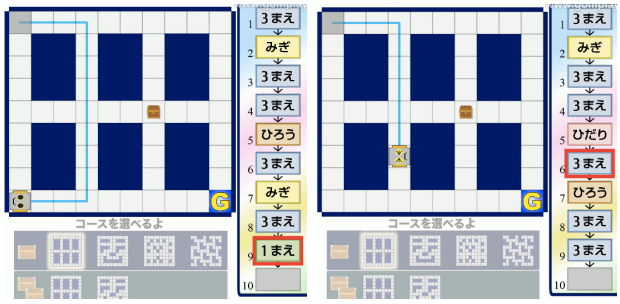


図 8 修正を繰り返してもゴールに辿り着けない実行例
Fig. 8 An Example of the Execution Not to Reach the Goal by Repeating Repairments

このように，児童らは実行と修正を繰り返してコース毎にゴールを目指した．中には，途中で挑戦をあきらめてしまう生徒もいた．そこで，コース毎の「挑戦者数」「ゴール者数」「あきらめ者数」を確認できた範囲で表 3 に示す．

次に，コース 1 を実行した各個人の結果の記録について，学年と性別で分けたものを図 9 に示す．「児童 ID」「学年」「性別」「実行回数」「実行時間」「最大配置数」という項目を表示した．

表 3 コース毎のゴール者数 (n=31)
Table 3 Number of the 'Finished' Students on Each Course

	コース 1	コース 2	コース 3	コース 4
挑戦者数	31 名	24 名	6 名	5 名
ゴール者数	28 名	20 名	5 名	4 名
あきらめ者数	3 名	4 名	1 名	1 名

「実行回数」は，途中の確認も含めて「動け！ロボット」(実行) のボタンを押して，ロボット動かした回数である．「実行時間」は，最初の命令カードを動かしてから，最後の「動け！ロボット」(実行) のボタンを押すまでの時間である．「最大配置数」は，一度の実行で配置した命令カードの枚数の最大値を示している．例えば，児童 01 は，ゴールまでに 3 回「動け！ロボット」のボタンを押してロボットを動作させている．その中で，1 回目は「5 枚」のカードを配置し，ひろう位置と動作の正しさを確認している．2 回目はそれを踏まえて「5 枚」のカードを追加し，合計 10 枚にして一気にゴールを狙っている．しかし，1 マス手前で止まってしまったため，3 回目は修正のために命令カードを「1 枚」追加して無事にゴールできている．従って，1 回目と 2 回目に配置した「5 枚」の 5 が最大配置数となる．31 名の中で，児童 10 と児童 16 は，コース 1 をゴールすることができなかった (表 4)．

表 4 児童 01 の動作
Table 4 Manipulation of a Student, Jidou01

	実行時枚数	配置枚数	動作の詳細
1 回目	5 枚	5 枚	「ひろう」で確認
2 回目	10 枚	5 枚	「ゴール」を狙うが 1 マス手前で止まる
3 回目	11 枚	1 枚	1 枚追加して「ゴール」

ここで，背景色をつけた「最大配置数が 8 以上の児童」は，学年が上がる毎に増えている傾向と実行時間が長くなっている傾向が見える．また，ゴールできなかった 2 名も最大配置数が 10 と高くなった．

4.2 高等学校の実践から

高等学校の実践では，全体的な印象について (1) 難しさと (2) 楽しさに関する印象をそれぞれ選択肢から選んで記録させた．また，4 種類のコース毎に (1) 確認も含めロボットを動作させた回数（以下，「実行回数」と記す），(3) ゴールまでに要した時間（以下，「実行時間」と記す），(4) ゴール時に使用したカードの枚数（以下，「カード枚数」と記す），(5) 難しさ（以下，「難しさの自覚」と記す）をプリントに記録させた．また，本節ではそこで得られたデータの集計結果を示す．

全体的な「難しさ」の印象を表 5 に示す．小学生を想定した教材であることから，51.3 %の生徒が「超簡単！・簡

ⓒ 2015 Information Processing Society of Japan

児童ID	学年	性別	実行回数	実行時間	最大配置数
児童01	6	M	3	1分 42秒	5
児童02	6	M	2	1分 45秒	10
児童03	6	M	3	2分 5秒	4
児童04	6	M	2	2分 36秒	10
児童05	6	M	3	3分 0秒	8
児童06	6	M	3	3分 15秒	13
児童07	6	M	5	4分 20秒	5
児童08	6	M	5	6分 54秒	11
児童09	6	M	8	8分 44秒	13
児童10	6	M	4		10
児童11	6	F	1	1分 5秒	10
児童12	6	F	4	2分 17秒	4
児童13	6	F	3	2分 28秒	5
児童14	5	M	3	3分 15秒	5
児童15	5	M	6	5分 50秒	5
児童16	5	M	4		10
児童17	5	F	2	1分 5秒	10
児童18	5	F	3	3分 6秒	5
児童19	5	F	4	3分 19秒	5
児童20	5	F	3	4分 29秒	10
児童21	5	F	6	5分 45秒	9
児童22	4	M	3	2分 20秒	5
児童23	4	M	3	2分 47秒	9
児童24	4	M	4	2分 57秒	4
児童25	4	M	5	3分 0秒	5
児童26	4	M	5	3分 12秒	5
児童27	4	M	3	3分 15秒	8
児童28	4	F	2	1分 49秒	10
児童29	4	F	5	2分 57秒	4
児童30	4	F	6	3分 40秒	4
児童31	4	F	8	3分 55秒	4

図 9 コース 1 を用いた児童毎の活動状況
Fig. 9 Personal Data about the Use of Course1

単」を選んでいるが，「少し難しい・かなり難しい」と感じた生徒も約半数の 48.8 %と高い比率を示した．

表 5 高校生による「動け！ロボット」の難しさの印象 (n=39)
Table 5 Difficulty of the Whole of the Activity

選択肢	回答	人数	割合
1	超簡単！	2 名	5.1 %
2	簡単	18 名	46.2 %
3	少し難しい	18 名	46.2 %
4	かなり難しい	1 名	2.6 %
5	超難しい！	0 名	0.0 %

次に，コース毎の実行回数，実行時間，難しさの自覚の集計結果を示す (表 6)．

この結果から，コース 1 は実行回数が少なく，実行時間が短い．このことから，コース 1 は易しいが，コース 2 以降で難易度が増していると言える．難しさの自覚でも同様の傾向を示している．そこで，「実行回数」「難しさの自覚」「実行時間」との間の関係を調べたところ，「難しさの自覚」が強い程，「実行時間」が長いという傾向が読み取れた (図 10)．

「ロボットを正しく動かすために，何に気を付ければよいか」の自由記述からは，以下のような答えがそれぞれ複数あった．

(1) 左右の向きに気を付ける
(2) 最短距離を判断する
(3) 動きを確認しながらプログラムを作る

表 6 コース毎の実行回数，実行時間，難しさの自覚
Table 6 The Number of Executions, Execution time and the Difficulty of Each Course

	コース 1	コース 2	コース 3	コース 4
実行回数平均	1.9 回	2.6 回	2.3 回	2.8 回
実行時間平均	1 分 29 秒	3 分 7 秒	2 分 9 秒	2 分 46 秒
1. 超簡単！	17 名 (43.6 %)	1 名 (2.6 %)	4 名 (10.3 %)	2 名 (5.1 %)
2. 簡単	20 名 (51.3 %)	16 名 (41.0 %)	23 名 (59.0 %)	10 名 (25.6 %)
3. 少し難しい	2 名 (0.0 %)	18 名 (46.2 %)	10 名 (25.6 %)	20 名 (51.3 %)
4. かなり難しい	0 名 (0.0 %)	4 名 (10.3 %)	2 名 (5.1 %)	7 名 (18.0 %)
5. 超難しい！	0 名 (0.0 %)	0 名 (0.0 %)	0 名 (0.0 %)	0 名 (0.0 %)

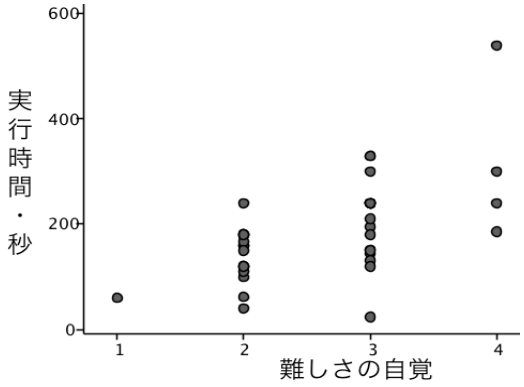


図 10 コース 2 における「難しさの自覚」と「実行時間」の関係
Fig. 10 Relationship between 'Difficulty' and 'Execution Time' on Course2

表 7 高校生による「動け！ロボット」の楽しさの印象 (n=39)
Table 7 Impression of Whole of the Activity

選択肢	回答	人数	割合
1	とても楽しかった	13 名	33.3 %
2	少し楽しかった	21 名	53.9 %
3	あまり楽しくなかった	5 名	12.8 %
4	つまらなかった	0 名	0.0 %

(4) マスの数を間違えない
(5) ミスをしない
(6) 拾うを忘れない
(7) ロボットの立場になって考える

4.3 実行時間の比較

小学生と高校生のコース 1 における「実行時間」を比較した (図 11)．横軸が秒数であるため，左側程早くゴールできたことを意味する．明らかに小学生よりも高校生の方が早くゴールできているが，重なっている部分も多く，小学生の上位 25 %程 (8 名) は，高校生の平均レベルと同等の活動ができていると言える．

ⓒ 2015 Information Processing Society of Japan

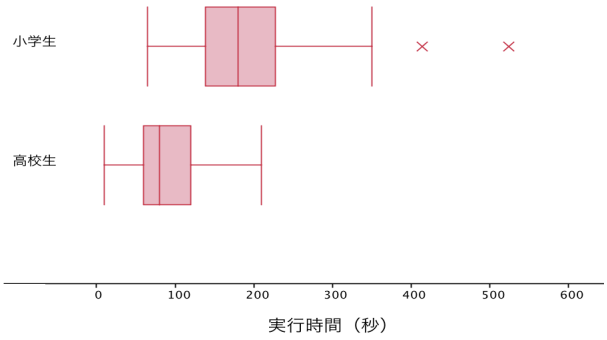


図 11 コース 1 に要した「実行時間」の小学生と高校生の比較
Fig. 11 Comaparison of 'Executing Time' Between Elementary Students and High School Students

5. 考察

小学生が扱う教材であるため，2 章で開発の指針を示した．本章では，これらの観点が満たされたかどうかを検証する．その上で，高等学校での実践を踏まえて，教材の適用範囲と入門言語としての役割を考察する．

5.1 小学校での活用

児童・生徒向けの教材として「楽しさ」を求めた．小学生にとってはイベントの雰囲気やイベント後のアンケート結果などから，楽しみながら学習できる教材であることがわかった．これは，コース毎にゴールしたときの達成感があることが大きいと考えられる．頭の中でコースを作り，それを命令カードと対応させたことは，プログラミング的思考である．生徒はゴールを目指して慎重に確認していた．これは，プログラムのトレースをしていたと言える．また，命令カードやその順番が間違えていてロボットが児童の予想と異なる方向へ行ってしまうことも多かった．そのために命令カードを交換したり，順番を変えたりした．これはプログラミングのデバッグをしていたと言える．これらのことから，本教材は当初指針とした 3 つの観点が満たされたと考える．

本教材は，CS アンプラグドの「プログラミング言語」の学習と，ドリトルによるプログラミングとの間を補完する目的で作ったものである．今回はドリトルは使わなかったが，まなべーの「３まえ」「みぎ」という命令は，ドリトルの「カメ太！　３０　歩く.」「カメ太！　９０　右回り.」という命令に対応する．まなべーでプログラミングの考え方を身に付けた児童が，日本語入力できるようになれば，ドリトルへとスムーズに学習を発展させることが可能であると考えられる．

本教材は，動作環境を整える上でも手間はかからない．Web ブラウザと Flash プレーヤーがあれば，ローカルでもネットワーク経由でも動作する．また，使い方は単純であり，教師は 2〜3 枚の命令カードを使って，ロボットを動かす方法と，その後の修正方法などを 1 分程度説明すれば，児童・生徒は理解できる．その後，複数のコースが用意されていることを示せば，児童・生徒は「自分で」解き始める．そのため，プログラミング言語の知識や経験がない教師が担当しても，気軽に授業を行うことができ，小学生の早い段階からプログラミング教育を可能にする．また，タブレット端末で動作するため，国家が提唱する「プログラミング教育」と「タブレット端末の利用」を同時に満たす教材である．

以上より，本教材が小学校の「コンピュータの仕組みを理解する」ための入門教材として適切であると考える．ただし，今回実践したのは 4 年生から 6 年生の高学年である．1 年生から 3 年生までの低学年に対して使える教材であるかどうかの検証は今後の課題である．

5.2 教材の適用範囲

高等学校での実践結果を踏まえて教材の適用範囲を考察する．高校生には易しすぎる教材であることから，我々は生徒が「楽しさ」や「難しさ」を感じることはないと予想していた．しかし，実際には 9 割近くの生徒が「とても楽しかった」「少し楽しかった」と回答し (表 7)，半数近くの生徒が「少し難しい」「かなり難しい」を選択した (表 5)．

この結論には，複数の要因が考えられる．まず，ロボットに仕事をさせるという設定が，ゲーム性をもっていたことと達成感を得られる仕組みになっていたことである．また，ロボットの立場になって方向を考えるといった思考は，多くの生徒は初めての経験であり，それが知的な楽しみとして感じられたとも考えられる．

これらのことから，制御構造しか持たない教材でも，高校生のプログラミング教育の導入段階としては，一定の利用価値があることがわかった．ただし，現段階で利用できる時間や機会は僅かである．今後は，他の制御構造を取り入れたり，ロボットにメソッドを定義して自立させるなどの機能を追加すれば，より高校生が利用するに適した教材になると考えられる．また，高校生で一定の活動ができたということは，中学生でも利用できると考えられる．今後は，校種に応じた実践が可能になるように，機能を増やしていきたい．

5.3 入門言語としての役割

まなべーはエラーメッセージを出さない．プログラムに誤りがあるかどうかは，ロボットの振る舞いによって学習者自身が判断する．誤りはすべては「論理エラー」であり，アルゴリズムの誤りである．この誤りは学習者本人がデバッグを行って取り除く必要がある．まなべーの場合は，ロボットが進んだ距離か，曲がる方向の認識の違いが誤りの主な原因となる．

ⓒ 2015 Information Processing Society of Japan

この種の誤りは，一度に多くのカードを置いていく学習者に多かった．図 9 に示した個人の記録からも，カードの最大配置数の大きな生徒は，実行時間が長くなる傾向があった．一度に多くのカードを置いた場合，児童 11 のようにミスがなければ，実行時間は短くなる．しかし，そのような生徒の多くは途中でミスをしていた．その場合，誤り個所を特定し，修正しなければいけない．誤りがすぐに見つからなかったり，修正と実行を繰り返すことで，実行時間は長くなる．特に，複数の誤りが入っていた場合，その修正はさらに難しくなる．児童 10 と児童 16 はその状況を改善することが出来なかったために，途中であきらめた．小学生は高学年になるほど一度に並べるカードの数が増えていたが，それは枚数を増やしても大丈夫だろう，という自信の表れであるが，結果として実行時間も増えてしまった．高校生の実践でも操作記録は取れていないが，「難しさの自覚」が強い程「実行時間」が長かった．これも，難しいと生徒は，一度にカードを置く枚数が多かったためだろうと推測する (図 10)．

これらのことから，一度に多くのカードを並べるという行為はリスクが大きく，誤り個所が特定できない場合，学習を進めることができなかったり，難しさを感じさせる要因となる．学習者自身はわかったつもりになったところで，そのように行動してしまうと考えられるが，多少時間や手間が掛かっても，きちんと確認しながらプログラムを作り込んでいく姿勢を持つが，プログラミングの学習を円滑に進めるためには必要である．カードを 1 枚置くたびに確認をするか，曲がる箇所や宝箱を拾う箇所などの「要所」で確認させることの指導の徹底が，学習の達成には必要であると考えられる．

通常のプログラミング言語では，制御構造として条件分岐や反復構造が入る．これらを組合せるプログラミング学習では，確認作業の必要性は更に高まる．従って，プログラミング学習の入り口として順次構造だけを学んでいる過程で，このような慎重なプログラミングの姿勢は徹底して教授することが，その後の学習には効果的である．まなべーは，専門知識を持たない教員でも指導が可能である，と説明したが，活動の前後に確認やデバッグの指導を徹底すれば，教育的な意義は更に高まる．また，それがまなべーの入門言語としての最も大きな役割と考える．

6. おわりに

プログラミング言語の入門用教材としてまなべーを開発し，小学生と高校生を対象とした 2 通りの実践を行った．CS アンプラグドの「プログラミング学習」を，コンピュータで体験するために，タイピングのおぼつかない小学生に対しても，学習を可能にすることを確認した．また，小学校から高等学校まで，その発育段階に応じて利用価値があることを示した．特に，活動の中で難しさを感じさせないようにするには，トレースやデバッグを適切に行わせることが重要であり，そこに教材としての価値があることを示した．以上より，本教材をプログラミング教育におけるコンピュータの仕組みを理解させるための教材として提案する．

ただし，本実践は小学生の高学年を対象としたものであり，より低年齢の層に対する検証はできていない．また，教材としては順次構造しか持っていないため，より高度な学習をしたいという要求に応えられるものにはなっていない．これらの検証や機能の追加は今後の課題である．

謝辞

本研究は，科学研究費補助金 (基盤研究 (C) 25350214，奨励研究 15H00219) の補助を受けています．

参考文献

[1] 世界最先端 IT 国家創造宣言, 首相官邸, (2014). http://www.kantei.go.jp/jp/singi/it2/kettei/pdf/20140624/siryou1.pdf
[2] 原田康徳, 勝沼奈緒実, 久野靖: 公立小学校の課外活動における非専門家によるプログラミング教育, 情報処理学会論文誌 55(8), pp.1765-1777(2014).
[3] 阿部和広: 津田子どもの創造的活動と ICT 活用, 情報処理 56(4), pp.350–354 (2015).
[4] 兼宗進, 中谷多哉子, 御手洗理英, 福井眞吾, 久野靖: 初中等教育におけるオブジェクト指向プログラミングの実践と評価, 情報処理学会論文誌, Vol.44, No.SIG13, pp.58–71 (2003).
[5] VISCUIT: http://www.viscuit.com/
[6] Scratch: https://scratch.mit.edu/
[7] ドリトル: http://dolittle.eplang.jp/
[8] 兼宗進, 阿部和広, 原田康徳: プログラミングが好きになる言語環境, 情報処理 50(10), pp.986–995, (2009).
[9] 兼宗進監訳: コンピュータを使わない情報教育アンプラグドコンピュータサイエンス, イーテキスト研究所 (2007).
[10] 江木鶴子, 竹内章: プログラミング初心者に トレースを指導する〜デバッグ支援システムの開発と評価〜, 日本教育工学会論文誌 32(4),pp.369–3Sl(2009).
[11] アルゴロジック: http://home.jeita.or.jp/is/highschool/algo/
[12] 石塚丈晴, 兼宗進, 堀田龍也: 小学生に対するアンプラグドコンピュータサイエンス指導プログラムの実践と評価, 情報処理学会論文誌, 教育とコンピュータ Vol1 No.2, pp.19–27(2015).

ⓒ 2015 Information Processing Society of Japan