

# 小学生向け図形描画ロボットプログラミングシステムの開発と評価 drawing robot programming system for elementary school students

小野寺峻真<sup>†</sup> 山口高平<sup>‡</sup>  
Ryoma Onodera<sup>†</sup> Takahira Yamaguchi<sup>‡††</sup>

<sup>†</sup> 慶應義塾大学 理工学部

<sup>‡</sup> 慶應義塾大学 理工学部

<sup>‡</sup> Faculty of Science and Technology, Keio University.

<sup>†</sup> Faculty of Science and Technology, Keio University.

## 要旨

本研究ではプログラミングにおける情報の構造化を体験させることによる論理的思考やロボットプログラミングを行うことによる実社会におけるプログラムの存在に気づかせるということを重点においた小学生向け情報処理教育用のシステムを提案する。生徒らが実際に図形描画ロボット「ナノローバー」をプログラミングできるプログラミング環境を開発し小学5年生の児童を対象に模擬的な授業を行い、生徒らへのアンケートと担当の小学校の教員へのヒヤリングによって評価を行った。

## 1. はじめに

現在、日本では2020年度より小学校でプログラミング教育が必修化され、情報教育の推進に力が入れている。しかしながら、文部科学省の「小学校プログラミング教育の手引（第二版）」[1]ではプログラミング教育は算数や理科などの既存の教科の中で主に行うこととされており、十分な時間が取れているとは言えない状況である。また、算数の授業の中で行われるプログラミング教育の中ではScratchが普及されているが、画面上だけでのプログラムの実行となっており、実社会の多くの場面でプログラムが利用されている認識を高めることができていないのが現状である。

このような背景から、本研究ではロボットを用いたプログラミングを行うことで、論理的思考、社会におけるコンピュータの存在、プログラミングと教科との関連という3点に着目した小学生向け情報処理教育用のシステムを開発し、それを算数の学習の中で実践することを目的とする。小学5年生を対象に本システムを用いて実際に小学校で行われているような算数の授業でのプログラミングによる多角形の作図に取り組んでもらい、システムの有用性を検証する。

## 2. 提案システム

本システムではもともと用意されてある命令を組み合わせるという活動だけでなく、組み合わせることのできた動きを一つの動きとして捉えさせるという情報の構造化を体験させることで児童たちの「プログラミング的思考」をより養うことができるのではないかと考えた。

また、その上で本実験ではタブレット端末やノートPCの画面上での処理だけではなく、児童の書いたプログラムによってロボットを動作させる。そうすることで、プログラムが実際に現実世界に対して影響を与えるという体験をさせ、自身の身の回りの機械もこのような情報技術によって動いているのではないかという気づきを与えることができるのではないかと考えた。

システム概要図を以下の図1に示す。

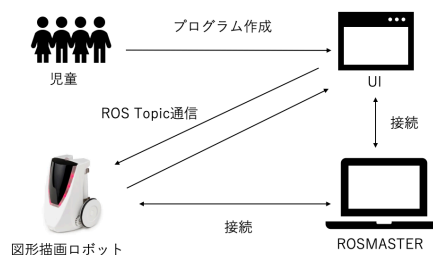


図1 システム概要図

本システムは児童が操作するナノローバープログラミング環境, 作成したプログラムに従って動作する図形描画ロボット, ROS を動作させナノローバープログラミング環境を立ち上げるための仮想マシンの3つによって成り立っている. 各機器同士の通信は WiFi によって行う. 児童は仮想マシンによって立ち上げられたナノローバープログラミング環境にそれぞれのタブレット端末からアクセスし, プログラムを作成する. プログラムを実行すると, 作成したプログラムに沿って ROS Topic 通信を用いて事前に設定されている図形描画ロボットの動作の Topic に値を Publish し, 動作を実行させる. 一つ一つの動作が終了するたび, 図形描画ロボットから動作終了の通知を出し, それを受け取り次第次の動作を行わせ, 図形を描画させる. 児童はプログラムの実行によって描画された図形を確認し, プログラミングを進めていく

### 3. 図形描画ロボット

本研究では, 児童がプログラミングによって図形を描画するロボットとしてヴイストン株式会社 [10] から発売されているナノローバー [9] というロボットを使用した. 以下の図2にナノローバーの外観を示す.



図2 ナノローバーの外観

ナノローバーは Arduino IDE と呼ばれる開発環境を用いてプログラムを開発し, それを本体の基盤に書き込むことで開発したプログラムによって動作させることができる. 書き込むプログラム内で Publisher, Subscriber を立ち上げ, ROS を立ち上げている PC と rosserial というライブラリーを用いて通信することで ROS による制御を行える. 本実験で扱うのは小学5年生の算数科の多角形の作図なので, 最低限必要とされるナノローバーの機能は「直進」, 「方向転換」, 「ペンの上下」である. それらを実行する為の Subscriber に加え, 児童により詳細な動きを実感してもらう為, 速度変更用の Subscriber も用意した. その上で動作が終了したことを知らせる為の Publisher を立ち上げる.

### 4. プログラミング環境

本プログラミング環境を開発するにあたって, 本研究では多角形の作図の授業に特化した最低限の機能を備えつつ, 前述の情報の構造化の機能を備えさせなければならなかった. そして, 今回の研究にて開発するプログラミング環境のユーザーとなるのは小学生であるため, あまり複雑な構造にはしないべきだと考えた. 例えばプログラムを作っている途中で関数を作成する際, 画面遷移などが発生し元の作っているプログラムが画面上で見えなくなることが発生した場合, 自身が編集しているのは何なのかということがわからなくなる可能性がある.

また, 開発を行うにあたって昨年度などに scratch での多角形の作図の授業を行なっているという杉並区立浜田山小学校の先生にヒアリングを行い, 小学校で使用しているソフトなどを見せていただくとともに, 小学生の理解力についてのヒアリングを行い, 開発の参考とした. ヒアリングの結果, 色やアイコンなど直感的に判別できるものがあるとわかりやすいこと, 関数や変数といった言葉はわかりづらい, 一方向の逐次実行は理解できることなどがわかった. これらの事項を踏まえた上で, 処理の流れを視覚的に捉えられるよう, フローチャートに近い形で表せるようプログラミング環境を開発した.

本環境はタブレット端末で操作することを想定しており, タブレット端末のブラウザアプリからアクセスをして使用する. そのために本システムは apache を用いて配備することで同一ネットワーク内で

あれば外部のネットワークに接続せずアクセスすることができる。また、ユーザーが触れるインターフェースは `html,css,JavaScript` を用いて作成しており、実装には `Bootstrap` というフロントエンド用のフレームワーク、`SortableJs` という要素のソートを行うためのライブラリーを用いている。

以下の図3にプログラミング環境の初期画面を示す。

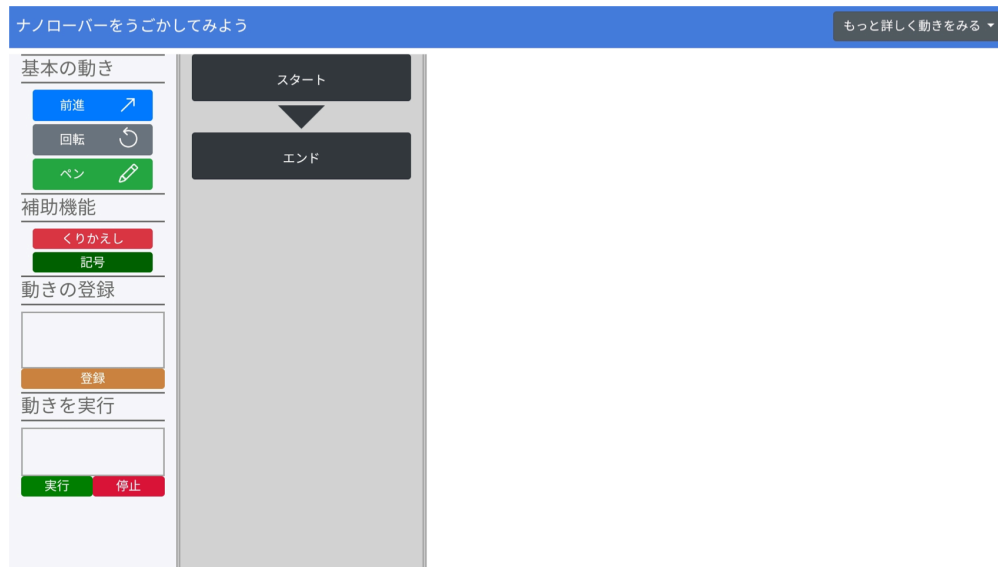


図3 プログラミング環境の初期画面

ユーザーは画面左側にあるボタン列にあるボタンをタッチし、「スタート」、「エンド」の間に各カードを表示、それらを並べていくことでプログラミングを行う。見出し「基本の動き」の下にある「前進」、「回転」、「ペン」のボタンはナノローバーの基本的な3つの動きを示している。また、「基本の動き」以外にも反復構造を構成する為の「くりかえし」、変数の宣言の為の「記号」ボタンを見出し「補助機能」の下に配置しており、「基本の動き」と同じようにボタンをタッチすることでカードを表示する。

「前進」カードではナノローバーを前にどれだけ動かすか自由入力にて `cm` 単位で指定する。当初はナノローバー内の処理と合わせて `mm` 単位での指定であったが、杉並区立浜田山小学校の先生に実験を行う前に意見を頂いたところ、小学生にとっては `cm` の方が直感的に大きさのイメージがしやすいのではないかとの意見を頂いた為、単位は `cm` とした。「回転」カードではナノローバーを左右どちらに何度回転させるか指定する。方向はプルダウンで、何度かは自由入力で指定する。「ペン」カードではナノローバーに装着されているペンをあげるかさげるかをプルダウンにて指定する。

次に、補助機能について述べる。補助機能ではナノローバーの動き以外でプログラミング的な役割を果たすカードを配置している。「くりかえし」カードは2つの部分に別れており、「くりかえし始め」にてくりかえしの回数を指定し、実行時に「くりかえし始め」と「くりかえし終わり」の間に入っている内容を指定回数分繰り返す。「記号」カードはプログラミングにおける変数の機能を簡易的に再現したものである。ユーザーはもともと用意されている4つの記号、「○」、「×」、「△」、「□」をプルダウンより選択し、その記号に自由入力欄に入力した値を保存することができる。またそれらの「記号」を使用する際には、各カードの自由入力欄にそれぞれの記号を入力することで使用することができる。また、各カードの自由入力欄では四則演算を行うことができる。加算ならば数値や記号同士の間「+」、減算ならば「-」、乗算ならば「\*」、除算ならば「/」を入力することで実行時に値を計算する。この四則演算は「記号」を交えて行うこともできる。

次に「動きの登録」機能について説明する。画面左の見出し「動きの登録」の下に配置されている「登録」ボタンを押すと以下の図4のようなウィンドウが画面上部に表示される。この画面で作成したい動きの名前を自由入力で、判別のために関数に設定したい色をプルダウンより指定し、登録ボタンを押すことで動きの登録をすることができる。

図4 「動きの登録」用ウィンドウ

登録を行うと、登録した動きのカードを表示する為のボタンが「登録」ボタンの上の枠に追加されるとともに、編集する為のフローがもともとあるメインのフローの右に追加される。2つ目以降の動きを登録するとボタンは1つ目の下、フローは1つ目の右に順に追加されていく。表示できる上限を超えた場合はそれぞれ上下、左右にスクロールすることで登録した全ての動きを表示することができる。また、どのフローを編集するかは各フローの上部をタッチすると変更できる。また、「記号」機能を用いて引数を表現することで関数のように使用することもできる。

続けて、「基本の動き」の詳細な内容を表示する機能について説明する。画面右上の「もっと詳しい動きをみる」ボタンをタッチすると、ボタンの下に「前進」と「回転」ボタンが表示される。これらのボタンを押すことで対応した「基本の動き」をより詳細に表したフローが表示中のフローの右に新たに表示される。この時に表示されるフローはナノローバーが「前進」または「回転」の動作を実行する時に行う処理をこのプログラミング環境上で表したものとなっている。詳細フローはフロー内のカードの並べ替えを行うことはできないが、設定されている値を変更することはできる。例えばこの「前進」の詳細フローであれば、動く際の速度を変更することができる。変更した場合は、下部の「設定を反映する」ボタンを押すことで実際に「前進」を行う際の速度を変更することができる。

最後に画面左下の「実行」ボタンを押した時の処理について説明する。「実行」ボタンを押すと一番左のメインのフローの内容に応じて「基本の動き」の順番とそのカードに入力されている値を保存したリストを生成する。メインフロー内に「くりかえし」や登録した動きのカードがあった場合はこの時点で処理する。このリストと入力されていた値に応じて、ナノローバーが立ち上げた Topic → ROS の Message を配信する。Message の配信には javascript で ROS との通信を行える roslibjs というライブラリーを使用している。対応する Topic へと Message を送ると一時待機し、ナノローバー側が配信する終了通知を購読し次第、次の Message を配信するという仕組みになっている。途中で「停止」ボタンが押された場合は終了通知を購読するのを終了し、現在ナノローバーが行っている処理で動作は終了となる。

## 5. 実験と評価

### 5.1. 実験内容

本研究で開発したシステムを用い、実際に小学生に使用してもらうことでシステムについての実験を行なった。概要は以下の表1の通りである。

表1 実験概要

実施対象	杉並区立浜田山小学校の5年生9人
実施場所	杉並区立浜田山小学校校長室
実施日	2020年1月14日,20日,21日
実施時間	12:55~13:40
実施内容	1. ユーザーインターフェースについての説明 2. 児童に正多角形を作図してもらう 3. 「動きの登録」機能の説明 4. 「動きの登録」機能を使って作図してもらう 5. 「基本の動き」の詳細と「記号」機能についての説明 6. 正方形と正三角形を複合した図形の作図

1,2回目の実験はそれぞれ14日と20日に実施し、対象者はそれぞれ4人と5人で、実施内容については上の表の1~5番を実施したが、5番については十分な時間が取れなかった。そこで、1回目の



実験の対象者に対して1月21日に3回目の実験を行なった。実施内容は3～4について復習した後、5番を改めて説明を行い、理解度を確かめるために6番の内容を実施した。

また、理解度を確認するため、実験後に以下の図5のようなアンケートを行った。また、実験の終了後、協力していただいた杉並区立浜田山小学校の先生にインタビューを行なった。インタビューの内容は「実験時の児童の反応」、「動きの登録」、「記号」の機能、「ナノローバーを用いることの意義」についての3点である。

<p>実験後アンケート (1,2回目)</p> <p>質問1 あなたはScratchなどを使ってプログラミングをしたことがありますか? 1. はい (ツールの名称) 2. いいえ</p> <p>質問2 質問1ではいと答えた方に質問です。今まで使ったことのあるツールと比べて今回の実験で使ったツールは使いやすかったですか?理由と一緒に教えてください 1. はい 2. いいえ</p> <p>質問3 今回の実験で使った「動きを登録する」機能について理解できましたか? 1. とても理解できた 2. 理解できた 3. あまり理解できなかった 4. 理解できなかった</p> <p>質問4 今回の実験を通じてナノローバーが前に進むときや回転するときどうやって進んでいるかわかりましたか? 1. とてもよくわかった 2. わかった 3. あまりわからなかった 4. 全然わからなかった</p> <p>質問5 最後に今回の実験に関して感想を聞かせてください</p>	<p>実験後アンケート (3回目)</p> <p>質問1 今回の実験で使った「動きを登録する」機能について理解できましたか? 1. とても理解できた 2. 理解できた 3. あまり理解できなかった 4. 理解できなかった</p> <p>質問2 今回の実験で使った「記号」機能について理解することはできましたか。また、この機能について意見などがあれば教えてください 1. とても理解できた 2. 理解できた 3. あまり理解できなかった 4. 理解できなかった</p> <p>質問3 最後にやった複雑な図形を描くプログラムについて理解できましたか? 1. とても理解できた 2. 理解できた 3. あまり理解できなかった 4. 理解できなかった</p> <p>質問4 最後に今回の実験に関して感想を聞かせてください</p>
--	--

図5 実験後アンケート

## 5.2. アンケート結果と評価

実験のアンケート結果について述べる。質問2について使いやすかったと回答したのは9人中7人だった。使いやすいと感じた理由としてはアイコンや色の使い分けなどの視覚面や機能が少ない分かりやすさなどの意見が挙げられており、使いにくいと感じた理由としては「記号」の機能のわかりにくさや通信の不安定さなどが挙げられていた。実際に授業で使用するとなった場合は今回よりも多くのナノローバーを制御して行かなければいけないため、より安定して接続できるシステムを構築することが必要となると考えられる。質問3,4では9人全員が理解できた、とても理解できたを選択しており高い理解度を得ることができた。3回目の実験の質問1を見ると全員が「とても理解できた」と回答していた。3回目の実験では前回の実験の復習から入ったので、複数回説明を受け体験することで理解が深まったのではないかと考えられる。1,2回目の実験では十分な時間が取れないと思われていたが、3回目では「記号」機能についてしっかりと説明する時間が取れたため、「記号」機能についての質問を追加した。結果としては4人中2人が「あまり理解できなかった」を選択しており、他の項目に比べ児童の理解度が非常に低い結果になった。今回「記号」機能を使う必要性が出てくるのが引数付きの関数を作成する場合のみであり、ほとんどの児童がそこについての理解ができておらず、そこまでの理解を得るのに時間が足りなかったのではないかと考えられる。また、「記号」という表現についても非常に曖昧な表現であり、そこが児童の理解の妨げになってしまっていたのではないかと考えられる。質問3については全員が「理解できた」を選択しており、問題を理解することはできたと考えられる。

次に、1,2回目のアンケートの質問5と3回目のアンケートの質問4、実験の感想について見ていく。感想からは児童たちが非常に楽しんでプログラミングを行なっていたというがわかった。学んでいく中で児童が楽しさを感じているのはモチベーションの維持にも繋がるため、本システムは児童の興味を惹きモチベーションを維持させる効果があったと言えるだろう。また、ロボットの正確さや仕組みについて言及する回答も複数あり、本システムにおけるロボットを使うことを通じて機械の正確さなどに意識を向けるという狙いがうまく働いていたと言える。今回のシステムは多角形の作図に特化したシステムとなっているため、プログラミングの自由度は非常に低くなってしまっている。今後はこのような意見を参考にし、システムの自由度を広げることが課題として挙げられる。

## 5.3. インタビュー結果と評価

続けて先生へのインタビュー結果について述べる。まず、児童の反応についてだが、先生側から見ても関心の持続というものを感じたようであった。また、昨年度行ったScratchによる正多角形の作図と比較しても児童たちが意欲的にプログラミングに向かっていったと意見をもらった。これらの効果について、

ナノローバーがあることによって児童たちの中で目的と手段が明確されたからではないかとの意見を頂き、本システムを通じ、児童がプログラミングについてより関心を持ってくれたと言える。次に「動きの登録」、「記号」機能についてだが、児童へのアンケート結果でもそうであったように「記号」機能を有効に使うことが難しそうだということを先生側も感じたということがわかった。今回の実験で言うと多角形の外角の性質という規則性を理解していないとこれらの機能を有効に使うことができないということである。しかしながら逆にいうとこれらを有効に活用することができるかということによって児童が多角形の性質をちゃんと理解しているかどうかということを知ることができるのではないかと考えられる。先生へのインタビューでは授業の中でしっかりと教えれば児童たちもこの規則性にたどり着くということであったので、例えば「正  $n$  角形を作図する関数」をプログラミングできるかどうかということを知算数の授業に取り込むことはプログラミングによってその教科の学びを確かなものにするのに繋がり、非常に有効だと考えられる。このときに考えられるのは正多角形の辺の数を引数とする関数であるが、現状のシステムでは引数をとる関数を表すには登録した動きの中で記号を用いて角度などを表現し、メインのフローの中で「記号」に値を入れてから登録した動きを呼び出さなければならず、非常にわかりにくくなってしまっている。このような引数をとる関数を簡素化することもシステムの自由度を拡張するには必要になると考えられる。最後に、ナノローバーを用いることの意義についてだが、ナノローバーを用いることで本研究の目的である社会の中のコンピューターの存在に気づかせるということへの効果があるのではないかとの意見を頂いた。画面上だけでなく、児童に実際にプログラムによって現実世界のものを動かさせるということによって、機械、ロボットがプログラムによって動いているということを実感させることができたのではないかと考えられる。

#### 5.4. 結果と評価まとめ

最後に上記の結果と評価をまとめたものを以下の表2,3に示す。

表2 評価点まとめ

評価点
ロボットを動かすことでプログラムが現実世界へ与える影響を考えるきっかけになっていた。
「多角形の作図」においては関数の作成が理解を確かめる課題になり、教科の学びをより確かなものにすると言える
本システムは児童たちの積極性、主体性のきっかけになっていた。

表3 課題点まとめ

課題点
再利用のある関数について教えるには教科に対する深い理解とそのための課題設定、プログラミング環境の改善が必要。
多角形の作図に特化しており、自由度を拡張する必要がある
システムの安定性を高める必要がある

#### 6. 終わりに

本研究では小学校におけるプログラミング教育を支援する目的で論理的思考、社会におけるコンピューターの存在、プログラミングと教科との関連という3点に着目したシステムを提案、評価を行った。結果的に児童の積極性、関心に効果を与え、重点をおいた3点に対しても有効に働き、情報の構造化を用いた課題を設定することで児童の科目に対する理解度をより深めることができるのではないかと考えられる。

今後の課題としては本システムでできるプログラミングの自由度を広げることでより汎用的なシステムにすることが挙げられる。そのためにも扱うロボットもナノローバーに限らず、例えば Anki の Cozmo や Vector などセンサーをもつロボットを用いることで幅を広げていくことが必要である。

#### 参考文献

- [1] 文部科学省, 小学校プログラミング教育の手引き (第二版), 文部科学省, 2018, [https://www.mext.go.jp/component/a\\_menu/education/micro\\_detail/\\_icsFiles/afieldfile/2018/11/06/1403162\\_02\\_1.pdf](https://www.mext.go.jp/component/a_menu/education/micro_detail/_icsFiles/afieldfile/2018/11/06/1403162_02_1.pdf)