

ViPPER - ロボットを使用した初等教育向けビジュアルプログラミング -

瀬古 俊一[†] 山岸 真弓^{††} 中西 健太^{††}
伊与田 康弘^{††} 永井 敏裕^{††}
服部 隆志^{††} 萩野 達也^{††}

ViPPER - Visual Programming for Primary Education with a Robot -

SHUNICHI SEKO,[†] MAYUMI YAMAGISHI,^{††} KENTA NAKANISHI,^{††}
YASUHIRO IYODA,^{††} TOMOHIRO NAGAI,^{††} TAKASHI HATTORI^{††}
and TATSUYA HAGINO^{††}

1. 概要

本研究は、小学生でもプログラムの動きが簡単に理解できるビジュアルプログラミング環境を提供する。携帯ゲーム機上でロボットの動きをプログラミングし、記述したプログラムの流れをアニメーションで表現しながらロボットに制御データを送信する。これにより、ロボットの動きとプログラムの流れを見比べられるためプログラミングを理解しやすくなる。結果、小学生でも簡単に理解できるビジュアルプログラミング環境が実現される。

2. 特徴

ビジュアルプログラミングを利用することで、命令を組み合わせるだけで簡単にプログラムを作成できる。文法エラーを絶対出さないようにすることで、書いたプログラムが必ず実行できるようにする。これらの機能があるため、小学生でも簡単にプログラムを書ける環境を目指す。

プログラムの実行結果としてロボットを使用する。実世界上のロボットが動くため、プログラミングした結果が画面内で動く環境よりも小学生に親しみを持たれやすくなっている。プログラミングは携帯ゲーム

機上で行うため、PC上でプログラミングするよりも小学生に抵抗感なく受け入れられる。また、ロボットと携帯ゲーム機は持ち運び可能なため、みんなで持ち寄って遊びながらプログラミングを学ぶことが可能である。

本環境は、実行過程のアニメーションを行うことによってプログラムが実行されていく様子を可視化できる。命令が実行されていく順番を明示的に見ることができるので、プログラムがどのように実行されていくかの理解を深められる。また、繰り返しや条件分岐の動きなども見ることができ、特に繰り返しに関しては残り回数がカウントダウンされていくため、制御構文が理解しやすい。

このアニメーション時に、実行過程と同期してロボットも動作する。そのため、画面を見ながらロボットの動きを追うことでプログラムの動きとソースコードとの対比ができる。これにより MindStorms¹⁾ や TensorBorg²⁾ といった従来のロボットの動きを制御するビジュアルプログラミング環境では難しかった実行過程とソースコードとの対応付けの手助けが可能となる。したがって、従来のものよりも、プログラミングの理解を深めることができる。

以上の特長を有することで、プログラミングへの導入の敷居を下げ、小学生にも分かりやすいプログラミング教育を行うことができる。

3. システム

携帯ゲーム機である PSP 上にビジュアルプログラ

[†] 慶應義塾大学大学院政策・メディア研究科
Graduate School of Media and Governance, Keio University

^{††} 慶應義塾大学環境情報学部
Faculty of Environmental Information, Keio University

ミング環境を構築し、それを用いてロボットを制御するプログラムを書く。ロボットには無線 LAN モジュールが搭載されており、無線 LAN を用いて制御データを PSP からロボットに送ることでロボットを動かす。

3.1 ロボットの仕様

ロボットの形状は左右に 1 つずつ無限軌道を搭載した車型である (図 1)。それぞれの無限軌道には独立したモータがついており、無線 LAN を使ってデータを送ることでモータを制御する。無線 LAN を用いているため、電波が届く位置であればどこからでもロボットを動かすことが可能である。また、複数のセンサを搭載しており、センシングの結果によって分岐するプログラムを書くことが可能である。

3.2 言語の仕様

PSP 上に実装してあるビジュアルプログラミング環境は、BASIC 風のビジュアル言語である (図 2)。プログラムの実行は Start から順番に矢印をたどっていき End で終わる。Start と End の間にあらかじめ用意されている命令をパネルとして挿入していくことでプログラミングしていく。各命令は引数を 1 つ持つので、ロボットが前進する距離や旋回する角度などを細かく指定できる。

繰り返しや条件分岐は入れ子可能であり、書いたプログラムは自動インデントされる。自動インデントとは、入れ子の中のベースラインが自動的に一段右にずれることによって繰り返しや条件分岐などの範囲を分かりやすくすることである。このように、手続き型言語の表記に似せることによって、将来手続き型言語を習うときに知識が繋がりがやすくなる。また、条件分岐は通常の条件式による分岐のほか、ロボットに搭載されているセンサの状態による分岐も可能である。例えば赤外線センサを用いて、前に障害物があるかないかで異なる振る舞いを行わせることができる。

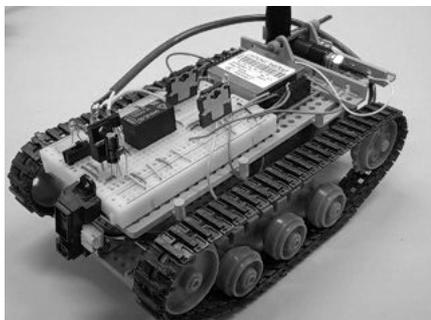


図 1 ロボット
Fig. 1 Robot



図 2 プログラミング画面
Fig. 2 Programming image

4. 今 後

4.1 機能の追加

現在の環境では関数を定義できないので、手続き型言語の機能としては不十分である。関数を定義できるようにし、関数を使った際の実行順序もアニメーションで表示することができれば、さらにプログラミングの理解を深めることができる。

モータの制御は、前進・旋回・後退といったあらかじめ定義されている命令しか使用できない。このモータの制御を自由に設定できる命令を実装すると、斜めに前進・その場で回転などといった新しい動きをロボットにさせることができる。モータ制御の自由化を実装すれば、自分で新しい動きを作り出すといった考える力を身につけさせることができる。

4.2 方 針

今後は、まだ小学生に利用してもらっていないので、実際に使ってもらいフィードバックを受けて初等教育で扱ってもらえるよう改良していく。また、ゲーム性を取り入れるなどして、より楽しく継続してプログラミングの勉強をしてもらえるような仕組みも取り入れていく。

参 考 文 献

- 1) F. K. Scott: "LEGO MindStorms: Not Just for K-12 Anymore", Robotics in education, Part 1, 2003, vol. 10, no 2 (20 ref.), pp. 12-18 [7 page(s) (article)]
- 2) BANDAI: "TensorBorg", available at http://www.roboken.channel.or.jp/tensor_pub/
- 3) 文部科学省: "学習指導要領における情報教育の改善内容", 文部科学省資料, 2002.
- 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", 情報処理学会誌: プログラミング vol.44 No. SIG13(PRO18), 2003.