

競争型知識共有プラットフォームを用いた ロボットプログラミングによる学習支援システムの検討

A Study of Learning Support System by Robot Programming using Competitive Knowledge Sharing Platform

前田 新太郎^{*1}, 古池 謙人^{*2}, 東本 崇仁^{*1}

Shintaro MAEDA^{*1}, Kento KOIKE^{*2}, Takahito TOMOTO^{*1}

^{*1}東京工芸大学工学部

^{*1}Faculty of Engineering, Tokyo Polytechnic University

^{*2}東京工芸大学大学院工学研究科

^{*2}Graduate School of Engineering, Tokyo Polytechnic University

Email: front4.shintaro@gmail.com

あらまし:プログラムのコードをより良くするためには他人のコードを学ぶ活動が重要である。本研究では、他者のコードから自身のコードを洗練する活動の支援を目的とし、コードのスコア化、ランキング掲載、コード共有が行える知識共有プラットフォーム、段階的なコードの提示機能を開発した。題材としてロボットプログラミングを扱う。これはコードの共有のためには同じ問題設定である必要があることと、振舞いを可視化しやすいためである。

キーワード: ロボットプログラミング, 知識の共有, 学習支援システム

1. はじめに

プログラミングにおいては、自身の書いたプログラムのコードをより良いものへ近づけるために、他人のコードから学ぶ活動が重要である。しかし、通常のプログラミング演習などでは、問題の通りに動くコードが構築できれば良いとされることが多く、それを洗練する機会は少ない。よって本研究では、他者のコードから学び、自身のコードを洗練する活動の促進を支援する。そのために、まず、複数人に対して、場面に合わせてロボットを操作するコードの構築を行わせる。次に、各々が構築したコードをサーバ上で実行し、スコア化する。このスコアをランキングとして掲載することで、学習者同士で競い合わせる。さらに自身のコードとスコアの近いコードのみが自身に共有される。このようにして、知識共有が行えるプラットフォームを開発する。

2. システム概要

他者のコードを理解するためには、そのコードがどのように振る舞うかを知ることが重要である。本研究では、ロボットの動きとしてコードの振る舞いを可視化する。これにより、自身と他者のコード間の差異をロボットの動きの違いとして認識することで、他者のコード理解や互いの知識共有を促進する。

開発したシステム画面の例を図1に示す。本システムは、コードを用いてロボットを操作し、課題に対するスコアを集計する。与えられる課題では、ロボットに畑の上で作物の種植えと育った作物の収穫を繰り返させ、作物の収穫数とそれに要するロボットの操作回数でスコアを決定する。また、1問につき、複数の畑の配置を提示している。よって学習者が高いスコアを獲得するためには、複数の畑に対応

する汎用的なコードの構築が求められる。よって学習者が高いスコアを獲得するためには、複数の畑に対応する汎用的なコードの構築が求められる。

本システムのロボットの関数リストを図2に示す。まずロボットを動かすために、番号1~4の移動関数を用意した。学習者はこの関数を利用して畑を巡るように移動するプログラムを構築する。Planting 関数はロボットの現在地点に畑が存在する場合、その畑に種を植える関数で、Harvest 関数はロボットの現在地点に育った作物が存在すれば収穫を行う関数である。これらの関数は呼び出す度にコストがかかるようになっており、本システムでは「使用回数」と呼んでいる。

また本システムは、多人数での利用を想定しており、各学習者の構築したコードをサーバ上でスコア化し、ランキング形式で掲載する。これにより、自身のスコアを向上させる動機付け⁽¹⁾の創発を狙っている。さらに、学習者が提出したコードのスコアと近いスコアのコードを閲覧可能にする。これにより知識共有において、自分とレベル差のある人のコー



図1 開発したシステム画面の例

ドが理解困難であるケースや、上位のコードをただ真似してしまうケースの解消が期待できる。

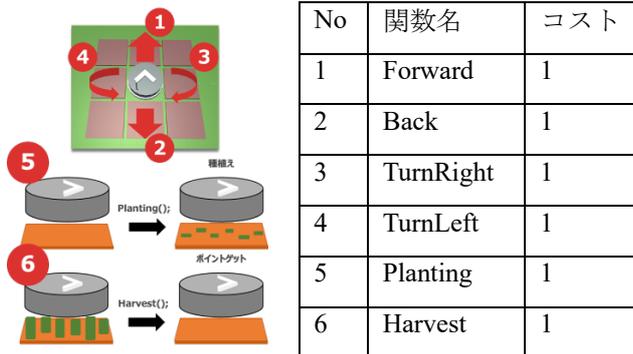


図2 ロボットの関数リスト

3. コードの評価指標

本システムの評価指標は「収穫数」、「処理時間」、「合計スコア」の3つが存在する。まず、「収穫数」とは、2章で紹介した、作物の収穫によるスコアである。作物は、ロボットが何かしらの行動を行う度に成長して行き、成長度によって獲得できるスコアが増加する。2つ目の「処理時間」は、ロボットを動かせることに必要な関数の使用回数を用いて擬似的に評価をしている。つまり、ロボットが何かしらの行動を取る度にコストがかかるので、学習者は少ない手数で作物を収穫することが求められる。最後の「合計スコア」は、「収穫数」から「処理時間」を引いた値で、コードの効率の良さの基準となる。これら3つの要素で算出されたスコアは、データベースに保存され、評価指標別にランキングとして掲載される。

4. 予備評価実験

4.1 予備評価実験の目的と手順

開発したシステムの学習効果を検証するために工学部大学生の5名に対して評価実験を実施した。実験は、チュートリアル→事前テスト→システム利用学習→事後テスト→アンケートという流れで行った。

まず、本システムの活動に慣れてもらうために、チュートリアルとして簡単な問題を提示し被験者と一緒に解決した。

事前・事後テストでは、本システムを用いて実施するが、ランキング昨日とコード共有機能の利用を制限する。つまり、コードの実行とスコア計算のみができる状態の本システムを利用してテストを実施する。一方システム利用学習では、機能制限のない状態で本システムを用いて学習をしてもらう。

最後にアンケートによって本システムや、本システムでの学習活動に対する印象を調査する。

4.2 テスト結果

事前・事後テストにより、ランキングとコード共

有機能を搭載したシステムの学習効果を調査する。

表1にコードの構文エラーやシステム特有の水たまりによるクラッシュ判定を除いた収穫ポイントの平均値を示す。

テスト結果から本システムのランキングとコード共有機能により学習効果があったと示唆される。

表1 事前・事後テストの結果

	事前テスト	事後テスト	差
問題1	153.0	501.6	348.6
問題2	10.4	109.5	99.1
問題3	97.8	655.2	557.4

4.3 アンケート結果

6件法によるアンケート結果を表2に示す。アンケート結果から、プログラミングにおいて効率的なプログラムの構築とより良いプログラムへ改良することの重要性が高く評価されたとともに、システムによる効率的なプログラムの構築とより良いプログラムへ改良する支援も行えたことが示唆された。また、自身と他者のプログラムを比較することは学習へ繋がることが高く評価されたとともに、コード共有機能による自身と他者とのコードの比較が支援できていることが示唆された。

表2 アンケート結果

No	質問項目	平均
1	効率的なプログラムの構築はプログラミングにおいて重要だと思うか	6.0
2	より良いプログラムへ改良することはプログラミングにおいて重要だと思うか	5.8
3	他者のプログラムと自身のプログラムを比較することは学習へ繋がると思うか	5.4
4	本システムの利用により、効率的なプログラムの構築に繋がると思うか	5.2
5	本システムの利用により、より良いプログラムの構築に繋がると思うか	4.8
6	本システムのコード共有機能で自身と他者のプログラムの比較に繋がると思うか	5.0

5. おわりに

本稿では、通常のプログラミング演習における問題点に着目し、コードのスコア化とランキングを活用した知識共有プラットフォームの開発と予備評価実験を行った。

参考文献

- (1) 長谷川忍: “認知的スキル学習支援におけるゲーミフィケーションの役割”, 教育システム情報学会誌, Vol.36, No.1, pp.9-16 (2019)