2ZF-06

入学前教育におけるプログラミング課題の履歴を活用した学習環境の試行

小山田圭吾† 市川尚† 富澤浩樹† 阿部昭博†
岩手県立大学大学院 ソフトウェア情報学研究科†

1．はじめに

本学部では早期合格者に対してeラーニングによる入学前教育を実施しており，プログラミングの課題も含まれている．筆者らは，作成されたプログラムを学習者間でチェックすることを支援するオンライン上の学習環境を開発してきた[1]．その際に，学習者の課題をチェックしてフィードバックを行う生徒をチューターとしている．このシステムは学習者が提出した課題の質向上にある程度寄与していたが，チューターによるチェックに不備が見られた．井垣ら[2]は，効果的な指導を行うには，様々な要因から発生する学習者の遅延や停滞を把握してそれに応じた指導が重要だとして，学習者のコーディング過程を分析・可視化することにより，助けを必要としている学習者の発見に役立ったことを報告している．

本研究では，チューターが効果的な指導を行う支援をすることを目的に，学習者のプログラミング履歴を記録し，チューターが履歴を確認しながらチェックできるシステムを試作した．オンライン上で行われる入学前教育で実践した結果について述べる．

2．システム構成・開発

2.1. システム構成

学習ツールはGoogle Blocklyを使用した．基本的には入学前教育参加者のみで学習を進める方針とし，教員などはシステムの運用や状況に応じて支援を行う．生徒の情報はMoodleから取得する．

2.2. システム開発

システムはブラウザ上で動作し，Javascriptと PHP，MySQL で開発した．機能は以下の通りである．

(1)プロセス記録機能

学習者がプログラムを作成した際に，どのような作業を行ったか記録する機能である．プロセスの記録はブロックを掴んで配置するごとに行われる．

(2)チェック機能

チューターが学習者の課題をチェックする機能である．学習者が取り組んだプログラミング課題とその作業履歴，チェック項目が提示され，それらを見ながらチェックを行う（図1）．作業履歴は画面下にあるシークバーとボタンを使いながら確認する．

(3)各課題とシステムのヘルプ

入学前教育の Moodle コース上に各課題で注目すべき点をチェック項目として提示した．また，Blockly の使い方やプログラムの考え方，チェック項目の説明などを記載したページを用意した．

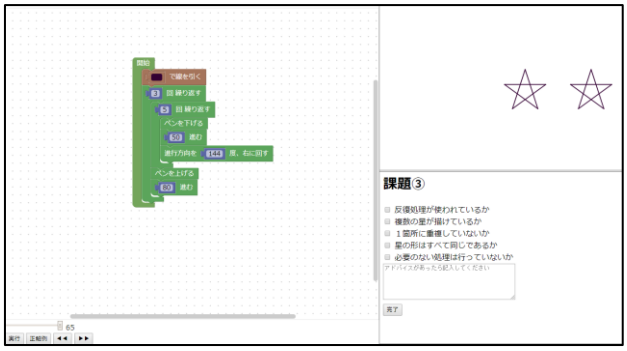

図1 チューター画面

3．結果

3.1. 入学前教育における実践

入学前教育参加者 20 名に事前テストとアンケートを実施し，それを基にチューターを選抜した．課題は 11 月 22 日～12 月 19 日の期間に学習者がチェック項目を満たすまで提出とチェックを繰り返すという手順で取り組ませた．チューターには，学習者に直接答えを教えないなどの注意を促した．課題終了後，事後テストとアンケートを実施した．

課題は制御構造の基本となる順次，分岐，反復をそれぞれ課題1～3，それらを組み合わせた課題4を用意した（表1）．テストは，課題に対応したBlocklyのプログラムを提示し，どんな動きをするか答える問題を10点満点で出題した．アンケートは，システム利用に関する質問を7件法と自由記述で構成した．

本実践は，学習者10名（プログラム未経験者3名），チューター10名で進め，2019年1月7日までに集まったデータについて分析した．

表1 課題内容

	課題内容
課題1	数字を2回入力して変数に格納し，2つの数字で四則演算を行った結果を，和・差・積・商の順で表示するプログラムを作りなさい
課題2	分岐処理の課題：数字を2回入力して変数に格納し，分岐処理を使って大きい方の数字を出力するプログラムを作りなさい．また，同じ数字が入力された場合，そのことを表示するようにしなさい
課題3	反復処理を使って星型を複数描くプログラムを作りなさい（一部が重なるのは良いが，1箇所に重なってはいけない）
課題4	数字を10回入力し，奇数が入力された回数を出力するプログラムを作りなさい

3.2. チューターのチェックについて

チューターのチェックは全体で61回行われた．チューターのチェックの妥当性を確認した結果を表2

Trial of learning environment utilizing the history of programming tasks in pre-admission education
Keigo Oyamada†, Hisashi Ichikawa†, Hiroki Tomizawa†, Akihiro Abe†
†Graduate School of Software and Information Science, Iwate Prefectural University

に示す．課題 1 は全ての課題のチェックが行われており，不備も見られなかったが，それ以外は若干の不備が見られた．課題 2～4 の未チェックの課題は同じ学習者の作品であり，提出が遅れたためにチェックがされていない．課題 2 の未完成の課題は指摘を受けているのに気付かず，次の課題に進んでいた．課題 4 の未チェックの課題 1 件と未完成の課題 2 件は，期限内に終了することができず期限内に提出できなかった学習者のものだった．また，集計時点で提出されていた課題の最終確認をした結果，提出されたすべての課題に問題は見られなかった．

表 2　チェック結果

	合格者数	未チェック	未完成	最終確認
課題①	5→8→10	0	0	10
課題②	4→6→8	1	1	8
課題③	9	1	0	9
課題④	5→6→6	2	2	6
※課題1は1回目のチェックで5人が合格であり，3回目以降のチェックで10人全員が合格になったことを示している				

3．3．　学習者のプログラミングの作業履歴

学習者の作業の履歴から得られた課題ごとの特徴を述べる．筆者が閲覧して調べた結果である．

課題 1: Blockly の使い方を確かめるような作業をしている学習者が 6 名見られた．また，変数の中身を確認する作業を行っている学習者が 6 名見られた．

課題 2：10 名全員が変数を 2 つ最初に用意してから分岐処理について考えていた．出力結果を 3 種類作成する必要があるが，躓いた学習者が 6 名いた．未経験の学習者の内 2 名は，提示したサンプルを最初に再現し，動きを確認してから課題に取り組んだ．

課題 3：星を 1 個描いてから数を増やすという順序で考えた学習者が 8 名いた．星を描くための直線を 5 回引く処理は 10 名全員ができた．線を引く処理や星を描く処理に躓いたと思われる学習者が 8 名いた．

課題 4：10 名中 8 名の学習者は反復処理と分岐処理を同時に使用する点に躓かず課題を進めていた．躓きの原因のほとんどは変数の扱いであり，奇数をカウントするという処理に躓いた学習者が 6 名見られた．提出が遅れた学習者 2 名もここに躓いていた．

3．4．　課題にかけた時間

学習者の作業時間と作業回数を課題ごとに集計した．作業回数はブロックを動かして配置した時を一回とした．長時間の放置が明らかであった学習者 1 名を除き，9 名の総作業時間の平均は 97 分 56 秒，作業回数の平均は 600 回であった．学習者や課題でばらつきが見られた．課題ごとの作業時間の平均を見ると，課題 4 が最も時間がかかっていた．

3．5．　テスト結果

事前テストは 20 名全員が受験し，事後テストはチューター役を除いた学習者 10 名が受験した．事前テストの 20 人全員の平均は 7．3 点，学習者 10 名の平均は 4．6 点であった．学習者 10 名の事後テストの平均は 7．8 点と上がっていた．個人の点数を見ると事前テストと結果が変わらない学習者が 2 名いたが，残り 8 名は点数が上がっていた．

3．6．　チューターへのアンケート結果

チューター10 名中 9 名が作業履歴を確認しており，その内，課題 1 は 9 名，課題 2～4 は 7 名が確認していた．作業履歴がチェックするに当たって役に立ったかを質問したところ，履歴を確認した 9 名全員がそう思うと回答していた．役立った場面について質問したところ，課題 1 は必要のない処理を行っておりどこで間違えたのかを見つけるため，課題 2 は条件分岐がしっかり行えているかの確認や正しい値が表示されない原因の調査，課題 3 と 4 では課題の試行錯誤を確認したというような記述が見られた．チューターが学習者のプロセスを見て気づいたことを質問したところ，作業の回数などから苦労しているのが伝わった，課題をやっていくうちに力が付いているのが分かったなどの記述が見られた．チューターが作業履歴を確認する際に注目した点があるか質問したところ，変数の扱いに注目した，課題 3 では星を 1 つ描ければスムーズに進められるかもしれないと思ってみていた，などの回答があった．

4．考察

本システムにより学習者のプログラミングの作業履歴を取得することができ，プログラミングの状況をより具体的に把握することができた．履歴を取得して提示することは，チューターからも肯定的な意見が得られ，効果的な指導を行うための支援にある程度寄与していたと考えられる．しかし，学習者によっては 200 を超える作業回数になり，チューターがそれらを確認するのは時間がかかる．そのため，履歴を自動的に分析して，必要なところを残しながら短縮して見られるようにすることなど，履歴を確認しやすくする必要がある．

5．おわりに

本稿ではプログラム学習環境の履歴を取得し，そこから得られた考察を述べた．今後は，取得できるようになったデータを活用し，チューターがよりよく指導できる（あるいは学習者の学習をより効果的にする）仕組みについて検討する必要がある．

参考文献

1) 小山田圭吾ほか：オンライン上での相互チェックを取り入れた入学前教育におけるプログラミング学習環境の開発，第 80 回全国大会講演論文集，pp．617-618（2018）．
2) 井垣宏ほか：プログラミング演習における進捗状況把握のためのコーディング過程可視化システム C3PV の提案，情報処理学会論文 vol54，No．1，pp．330-339（2013）