

大学における初学者に対するプログラミング教育環境の構築と実践

Practice of Programming Education using a Developed Learning Environment for Beginners of University Students

布施 泉^{*1}, 中原 敬広^{*2}, 岡部 成玄^{*1}
 Izumi FUSE^{*1}, Takahiro NAKAHARA^{*2}, Shigeto OKABE^{*1}
^{*1}北海道大学
^{*1}Hokkaido University
^{*2}合同会社三玄舎
^{*2}Sangensha LLC.
 Email: ifuse@iic.hokudai.ac.jp

あらまし：本稿では、大学の初学者を対象としたプログラミング教育の実践について報告する。初学者に特化した Moodle ベースの学習環境を構築し、Ruby 言語を用いたプログラミングの授業を行った。初学者用に、一画面でプログラムの入力、実行・出力を行うとともに、教師用資料の閲覧ができる環境であり、自宅からも同じ環境で利用できる。学習者の記録から、本学習環境は効果的に利用されたと考えている。
 キーワード：プログラミング, Ruby, Moodle, 初学者

1. はじめに

著者らは、大学の一般教育における初学者のための授業用プログラミング環境として、Moodle ベースの学習環境を構築している⁽¹⁾。本稿は、当該学習環境の評価を行うために、当該システムの利用有無による授業の課題進捗状況と、学習者の作業記録および授業内外での当該学習システムの利用履歴を確認し、本学習環境の分析を行うことを目的とする。

2. Moodle を用いたプログラミング学習環境

本学習環境は、初学者が大学の一般教育の授業としてプログラミング教育を受けることを前提として設計している。本学習環境の利用者は、大学入学後の一般情報教育の中で簡単なプログラミング (Scratch) は体験済みで、プログラミングに興味を持っているが、本格的なプログラミングの経験はない。本学習環境の設計方針を以下に示す。

1. 授業での利用が大前提である。自宅での予習復習においても、同じ学習環境で利用可能であること
2. 教師が資料提示を安直にできること
3. 学習者の集中力を保持するために、一画面でプログラムの編集・実行ができ、教師の提示資料が確認できること
4. プログラミング言語は汎用的なものであり、十分に学習が進んだのちは、本環境から各自の環境への移行がスムーズにできること

以上の要件から、ウェブインターフェースで、学習者管理ができる Moodle プラグインとして環境を構築することとした。また、プログラミング言語として、Ruby と Python が利用可能である。

本稿の授業では、Ruby を用いた。図 1 に Moodle 上のリンクをクリックした際の初期場面 (学習者毎のフォルダが表示) を、図 2 にプログラムファイルを開いた際の画面を表示する。

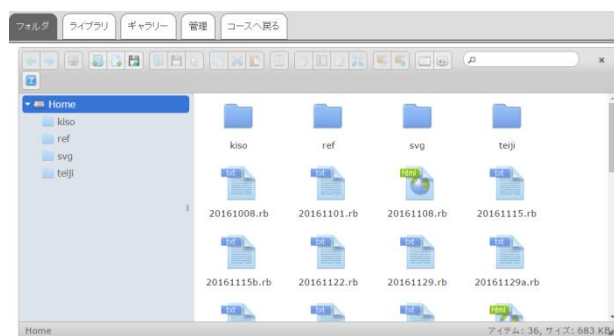


図 1 プログラミング学習環境画面



図 2 プログラムファイルを開いた際の画面

図 2 に示す通り、画面右上には学習者が取りこんだライブラリ資料を選択表示できる。授業時は、ここに教師が指示した資料を表示する。ライブラリにプログラムファイルを指定し、そのまま実行することもできる。その場合には、別タブに実行結果が表示される。自らが作成するプログラムは、図 2 の左側に配置され、それを上部の「保存して実行」ボタンを押下することで、右下の領域に実行結果を表示することができるようにした。

本学習環境では、学習者は画面遷移なしに一画面内で教師資料を閲覧し、内容を確認ながら、プログ

ラムを作成・実行できるため、プログラムの内容に集中して取り組むことができると考える。

3. 実践結果

図3は本環境の利用有無(2015年度は利用無, 2016年度は利用)による課題進捗であり, 第1週から第12週の授業で提出した基礎課題数を順次カウントした。受講者は3グループに分かれて授業を受けており, 図に記載の初学者グループの人数は, 各年度20名前後である。図3には第2週, 5週, 8週, 11週のみ明示しているが, 左から順に週数を表す(初回はガイダンスである)。授業課題は11の基礎課題と応用課題から構成される。図3の横軸は, 授業の受講者全員が11題の基礎課題を全て提出すると右端の11に達するように規格化している。図3に示される通り, 本環境を利用した方が, 第3週目以降, 毎週1課題程度, 満遍なく提出しており, ほぼ11週で基礎課題の提出が終了していることが確認された。一方, 2015年度の本環境を利用していない場合には, 各週で進捗にバラツキが見受けられる。

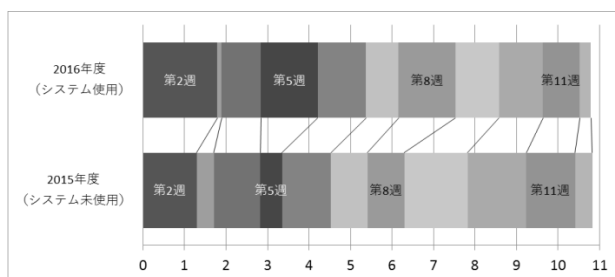


図3 各授業週における基礎課題の提出進捗状況

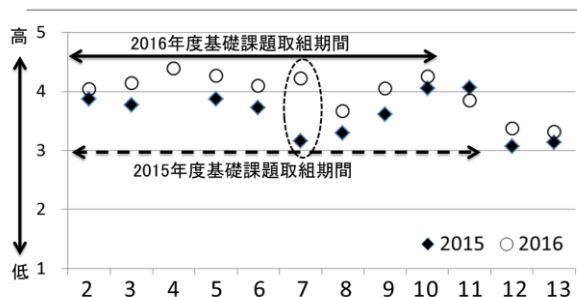


図4 各授業週における授業進捗の満足度についての学習者の自己評価 (5段階: 横軸は授業週数)

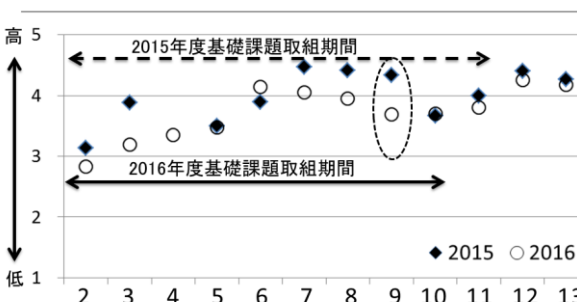


図5 各授業週における授業の難易度についての学習者の自己評価 (5段階: 横軸は授業週数)

図4, 図5は, 学習者に毎週授業終了時に記載させている作業記録から, 授業進捗に関する満足度と授業の難易度(5段階評価)を比較したものである。2015年度に比べ, 2016年度は満足度が平均的に高く, 難易度はやや難しい程度で授業週数によるバラツキが少ないことが確認できる。

4. 応用課題

基礎課題終了後は, 応用課題を学習者に取り組ませる。応用課題では, SVG, Ruby/Tk などを用い, グラフィカルな結果を描画する課題や文字の前後のつながりなど基礎課題で扱った内容を少し高度にした課題など, 多種の課題群を用意し, 学習者に選択させ数に制限なく解かせることとしている。この際, 基礎課題で用いていた本学習環境の利用有無についても自由とした。本 Moodle プラグインで Tk を利用し, ブラウザ上でグラフィカルな出力をレンダリングする機能を用意するのは困難であったため, Tk を用いたプログラムを作成する場合には, 学習者のローカル PC でプログラムの作成と実行を行わせたところ, 実行に特段の困難はなく, 学習者は好みに応じ, スムーズに環境と課題を選択することができた。約 1/3 はローカル PC で Tk を用いた応用課題を選択し, 2/3 は本環境をそのまま使い続けた。

5. 考察

前章の結果から, 本システムの利用はスムーズに行われ, 図3より第6週から第9週程度までは, 前年度に比して1週間程度早めのスピードで基礎課題が提出されていることがわかる。学習者の自己評価においても, 授業進捗の満足度と難易度は週数によるバラツキが少なく, 本学習環境を用いることで, 学習者はスムーズにプログラミングを行うことができたと考える。応用課題の実施においては, 1/3の学習者は, 本環境ではなく, ローカル PC 環境で Ruby を問題なく実行できていた。よって, 目的の一つである「学習が進んだのちは, 本環境から各自の環境への移行がスムーズにできること」についても達成できたと考える。これらの結果と考察から, 本学習環境は授業での初学者用のプログラミング学習環境として効果的に利用できることが示されたと考える。

6. まとめ

Moodle を用いた大学初学者用のプログラミング学習環境を開発し, 実際の授業で利用した。プログラムの作成・実行を一画面で設計したことにより, 課題が効率的に実行され, 学習者が大きなバラツキなく満足して利用していたことが確認された。

参考文献

- (1) 布施泉, 中原敬広, 岡部成玄: “授業での利用を前提とした初学者用プログラミング学習環境の開発 — Ruby 言語の Moodle 環境での学習支援 —”, 大学 ICT 推進協議会年次大会, FE22 (2016)