

1ZM-01

誤りから学ぶプログラミング復習コンテンツの制作

石川悠真[†] 土肥紳一[‡]

東京電機大学大学院システムデザイン工学研究科[†] 東京電機大学システムデザイン工学部[‡]

1. はじめに

東京電機大学システムデザイン工学部デザイン工学科では1年次後期に初めてのプログラミング講義として「コンピュータプログラミング I」を受講する。この科目は必修科目となっている。毎年受講生の半数が初回授業アンケートにてプログラミングが初めてと回答している[1]。本研究では、初学者が陥る誤りを調査及び想定し、そのサンプルコードと改良後のサンプルコードをコンテンツとして提示する。そして、学習者がそのコンテンツを理解し、プログラミング学習に繋げることを目的とする。つまり、ユーザがその誤りを知ることで、プログラミング学習に対してどう影響するかを調査する。そのためのコンテンツを試作し、使用状況を調査する。

2. Processing について

Processing は、イメージ、アニメーション、そしてインタラクションを生み出すソフトウェアを書くためにあるプログラミング言語である[2]。画面に楕円を描くとき、ソースコードを1行書くだけでできる。視覚的なフィードバックが即座に得られ、スケッチブックに絵を描く感覚でプログラミングできるように設計されている。Processing は、Java で作られており、オブジェクト指向プログラミングの基礎も学習できる。この言語を使用して、「コンピュータプログラミング I」で扱っている[3]。

3. 誤りについて

誤りは「実行結果は正しいが、ソースコードは誤っているもの」とする[4]。目的に対しての実行結果は正しいが、達成するための過程が誤っていればそれは誤りであると定義した。2022年度の「コンピュータプログラミング I」の中間試験で出題されたある問題の一つで、一部の解答者は、実行結果が正しく見えたが、その中のソースコードが条件式の部分で誤りがあったため、減点されたという事例があった。「これも正解ではないのですか」という質問を受けたが、確かに条件式の計算部分で誤りは起きているため、それは不正解となった。良いソースコードを書くためには、あえて悪い記述の弊害を知識として得ることが必要と考えている。そのうえで変更し強い構造を知ること、悪しき構造の具体的な課題を認識し、設計改善が可能となる[5]。ソースコードを記述している人が、実行結果がうまくいっているようにみえるために、ソースコード上の誤りに気付にくい事例を集めてコンテンツにしようと考えた。

4. コンテンツの試作

4.1 SharePoint による制作

授業は、教科書の例を入力しながら進行する。誤りのコンテンツを制作する上で、ソースコードでどういう誤りが起こり得るか想定した。想定した理由としては、実例として集めるにはあまりに数が乏しかったからだ。そのため、2022年度の「コンピュータプログラミング I」の中間試験の実例のほかには、教科書の内容と照らし合わせながら想定した。教科書、過去に出題された試験を基に誤りの定義に沿ったものを Microsoft の SharePoint にて制作した。SharePoint を利用した理由としては、

本学がライセンス契約を結んでいることで利用できること、コンテンツの管理とコミュニティで共有することができることで先行研究として利用されていた[6]。公開したコンテンツは11個である。また、レスポンスデザインに対応しており、スマートフォンでも閲覧できる。トップページにQRコードを載せているため、デスクトップから閲覧した場合、スマートフォンでそのQRコードを読み込むことで即座にスマートフォンでの閲覧を可能にする。コンテンツは4つのカテゴリに大きく分けた。分けた理由は実行結果ごとに分けた方が閲覧時に見分けがつきやすいと考えたからだ。「1-1」からは単円のみを表示している。「2-1」からは2つの円を表示している。「3-1」からは、3つ以上の円を様々な方法で表示している。「4-1」からは総集編として円の表示を踏まえたゲームを表示している。SharePoint にて作成したトップページの一部の様子を図1に示す。



図1 SharePoint で作成したトップページの一部

4.2 円の表示の意図

単円の表示に付随して本研究の意図の解説とこれから出るコンテンツの大きな解説を載せた。本研究では、円の表示の実行結果が主であり、それに絞った理由を載せている。幾何学的な単純さ、座標系の理解、パラメータと変数の使用という3点で円の描画に絞った結論を述べた。幾何学的な単純さについて、円は中心点と半径という2つのパラメータだけで定義される単純な形であり、ソースコードを書く際に必要となる計算が最小限に抑えられ、基本的な描画関数にすることで、誤りのコンテンツに集中できると考えた。座標系の理解について、円を描画する際その中心の座標を指定する必要があり、これにより Processing の座標系について学べると考えた。パラメータと変数の使用について、その大きさ、色、位置といった属性を変

Proposal for Programming Learning Content from Errors Based on Instructional Design

[†]Yuma Ishikawa, Graduate School of System Design and Technology, Tokyo Denki University

[‡]Shinichi Dohi, School of System Design and Technology, Tokyo Denki University

数や関数のパラメータとして扱うことができる。他の 2 つの上記と並び、変数の基本的な使用方法を用いることで誤りのソースコードに集中できると考えた。

4.3 単円の表示

単円の表示について実行結果とソースコード、解説を載せている。これらは円の色を除いて同じ実行結果が出力される。その中で、ソースコードを複雑化しながら、どういった誤りが潜んでいるかを載せた。その内、変数名について作成したコンテンツを図 2 に示す。誤りについて、改善後のソースコード、解説は折り畳み式となっており、最初は閉じられて表示される。



図 2 変数名についてのコンテンツ

4.4 2つの円の表示や様々な誤り

2 つの円の表示について実行結果とソースコード、解説を載せている。これらの意図として、「1-1」からは単円の表示についての誤りを載せたが、円が増えた場合も基本的な考えは変わらないことを示すためである。しかし、図形が増えたことによって、様々な表現方法があるということを示した。

「1-1」からは単円、「2-1」からは 2 つの円について触れた。そこで、「3-1」からは、円の数を指定せず、増えた場合でも考え方は同じであることを示した。特に「3-3」では、for 文での誤りをコンテンツとして提示し、2022 年度の「コンピュータプログラミング I」の中間試験の実例を題材として利用した。

4.5 総集編としてのゲーム

これまでの表題では、円を表示するプログラムに対しての誤りを載せた。「4-1」からは、学習者に対してこれまでの誤りを総集編という形で、学習者がある場で遊べるゲームを制作した。これらのソースコードには、「1-1」から「3-3」までに出題された誤りを含みながら制作し、誤りの場合も改良後の場合も、実行画面では同じように見えている。

5. 使用状況

Microsoft SharePoint には分析機能がある。これは、過去 7 日、30 日、90 日に分けて重複しない閲覧者数、サイトの訪問数、ユーザあたりの平均時間が記載される。ほかには、過去 7 日間で人気のあるコンテンツとして、サイトページ名、コンテンツページに対する重複しない閲覧者数がある。使用状況に関する分析情報として、デバイスごと、時間ごとで表記される。これらは

csv ファイルで保存することができる。また、公開しているコンテンツは、東京電機大学のアカウントを持っている方のみ閲覧できる。2024 年 1 月初旬の使用状況分析を図 3 に示す。

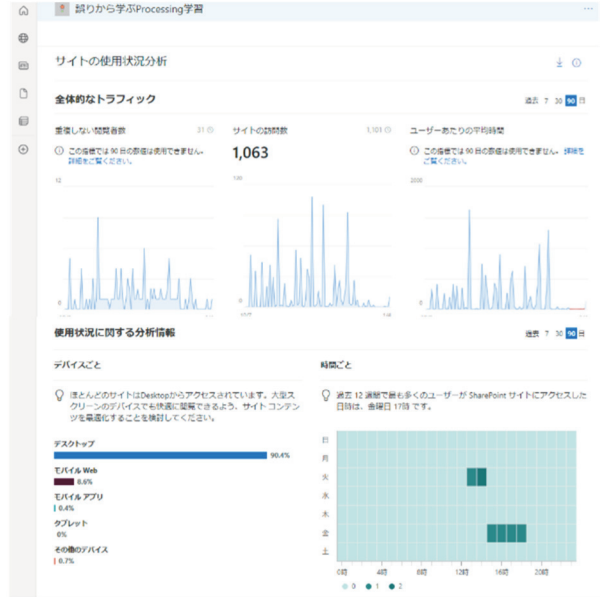


図 3 2024 年 1 月初旬の段階の使用状況

評価方法は、個々のコンテンツ内に Microsoft Forms を用いて、理解度、見やすさ、納得したかどうかの質問項目を用意した。最後に、誤りの全体を通した理解度調査とアンケート調査を用意した。ここで、学習者にモチベーションへの影響を調査する。学習者への負荷を考慮し、自由記述は最後のアンケート調査以外は設けず、全て5段階評価、理解度調査については 4 つの選択式を用いた。また、それぞれアンケート調査は一度のみの回答とし、理解度調査については何度も回答可とした。

6. まとめと今後の課題

本研究では、Processing 言語における誤りのあるソースコードを用いて、サンプルコードと改良後のサンプルコードをコンテンツとして提供することで、学習者がコンテンツを理解し、プログラミング学習に繋がるか調査することを目的とした。誤りとはあくまで「実行結果は正しいが、ソースコードは誤っているもの」とした定義に沿う。本論文では、コンテンツの試作公開、使用状況を述べた。

今後の課題として、使用者の数が少ないため、評価に十分なデータが取れないことが一番の課題である。5 で述べたように、受講生がサイトを訪問した時に、訪問時間を延ばし、公開しているコンテンツに行くような工夫を考えていく。

参考文献

- [1] 「コンピュータプログラミング I」の Web サイト、<https://dohi.chiba.dendai.ac.jp/~dohi/computer-programming-1/ad/>, 2024 年 1 月 9 日閲覧
- [2] Cassy Reas, Ben Fry, Processing をはじめよう, オライリージャパン, 2016
- [3] 土肥 紳一, Processing によるオブジェクト指向プログラミング入門教育の実践, 情報教育シンポジウム論文集, pp.134-141, 2018
- [4] 石川 悠真, 土肥 紳一, インストラクショナルデザインに基づいた誤りから学ぶプログラミング学習コンテンツの提案, 情報処理学会第 85 回全国大会論文集(4), pp.665-666, 2023
- [5] 仙場大也, 良いコード/悪いコードで学ぶ設計入門, 技術評論社, 2022
- [6] 廣川 大樹, 土肥 紳一, Processing 言語による手続型プログラミング復習コンテンツの試作, 情報処理学会第 85 回全国大会論文集(4), pp.681-682, 2023