

分岐を学ぶために効果的な教材を考える —お絵かきプログラミングによる入門科目を振り返って—

米屋 美雪¹ 内田 奈津子^{2,a)}

概要: プログラミング入門においては、プログラミングの概念を学び、原理を理解することが重要である。プログラミングの基礎知識には、制御構造があるが、プログラミング初学者にとっては難しく、躓きやすい。初学者の学習を助けるための工夫は様々あるが、著者らは、絵を描く教材によりプログラミングを学んだ。しかし、制御構造の分岐に関しては、演習問題では理解することができたが、活用するまでには至らなかった。本論文では、学習者の視点でプログラミング入門科目を振り返り、初学者が分岐を学ぶための効果的な教材を検討して提案する。

キーワード: プログラミング入門, 制御構造, 教材, お絵かき教材, 学習者視点

Review of effective teaching materials for learning Selection: Reflections on an Introductory Programming Course with Drawings

1. はじめに

時代の変化とともに、プログラミングとプログラミング的思考の重要性は年々高まっている。2020年からは小学校でプログラミング教育が必修となり、プログラミングは誰もが学ぶ、誰もが使うツールとなった [1]。

文部科学省の「超スマート社会における情報教育の在り方に関する調査研究」[2]によると、大学教育においても一般・共通教育として情報学教育が広く実施されている。しかし、内容や演習量にばらつきがあるほか、企業ではIT人材の不足が叫ばれており、満足な教育は行き届いていない。そこで、満足な情報教育を施すために、有効なプログラミング入門教材の確立が必要であると考えた。

著者はプログラミング初学者であり、大学の講義において絵を描く教材を使ってプログラミングを学んできた。その中で、抽象化は比較的容易に理解できたが、制御構造の理解に苦労した。

また、履修者の中でも様々なつまづきがあり [3]、実際

に制御構造を使った方が簡便に記述できるロジックを制御構造を使わずに記載している者が多かった。

著者の所属する大学は文系大学であるため、当大学に属する学生のだれもが理解できる教材を確立できれば情報処理に長けていなかったり情報学に興味をもちがなかったりする人でも理解できる、すなわち社会のより多くの人がプログラミングを理解できる教材に近づくであろう。

プログラミングの学び方は複数存在するが、本論文では、お絵かき教材でプログラミングを学んだ体験をもとに学習者の視点で、分析し、課題点を明らかにする。そして、その結果から、誰もが活用できる制御構造、特に分岐を学ぶための効果的な教材を提案する。

2. 先行研究

大岩は、[4]でプログラミング初学者について以下のように分析している。

初心者はプログラムを作ろうとしても、何から始めてよいか分らない。今作ろうとしているプログラムが何をしようとしているか、それを実現するために何が必要かを確認したうえで制作作業を始めれば、自分のやっている作業が全体の中でどのような位置づけになるかがわかる。うまくプロ

¹ フェリス女学院大学文学部コミュニケーション学科
Ferris University, Yokohama, 245-8650, Japan

² フェリス女学院大学
Ferris University, Yokohama, 245-8650, Japan

a) uchida@ferris.ac.jp

グラムが作れない人はプログラムの全体像を把握せずに、自分が作れそうな所から作業を始めるので、途中で行き詰まってしまったり、それまでの作業が無駄になったりする。

この分析は、プログラミング初学者の現状や躓く箇所を明らかにしているが、あくまでプログラミング上級者の視点から検討したものである。著者はプログラミング初学者であり、だからこそ真にわかりやすくレベルにあった教材の提案ができると考える。

3. どのように学んできたか

3.1 全体の構成

2020 年度後期に開講された、フェリス女学院大学国際交流学部 国際交流専門科目「情報発信と世界」にて学んだ。著者は、文学部の所属であるが、本講義は他学部解放科目であり、また1年生から4年生まで履修可能であることから、履修した。

本講義の到達目標を、以下に示す。

- (1) プログラムの仕組みを学び、簡単なプログラムが作れるようになる。
- (2) プログラムを通して、アルゴリズムの概念を学び、問題解決の基礎を身に付ける。
- (3) PBL(Project based Learning) の実践を通して、課題解決能力、コミュニケーション能力などを身につける。

講義は全 15 回で構成されている。今年度はオンライン授業であったため、初回は環境設定を行い、前半の第 2 回から第 7 回まではプログラミングの基礎知識を、後半の 8 回から 15 回までは前半で学んだことを基にグループに分かれ、一つの絵を作成した。表 1 に具体的な講義過程を示す。

3.2 授業の特徴

授業の特徴としては、お絵かき教材を用いること、講義と実践で学んだ後、後半のグループワークで再度実行する点があげられる。お絵かき教材では、視覚的にプログラミング的思考、図面設計、デザイン開発を学ぶ。後半のグループワークによって、理解度の再確認と活用度が図れると同時に、教え合いによる更なる理解力の向上が見込める。また、チームで行うデザイン開発の過程で、開発プロセスのモデルも学ぶことができる。

3.3 テキスト内容・進め方

テキストは、解説と演習問題で構成されている。それぞれの項目ごとの説明を受けたのち、演習問題に取り組んだ。演習問題は複数問あり、教員から指定される場合と学生の各々が選択する場合がある。また、授業で扱わなかった演習問題は次回までの課題として、1~2 問自由に選択して解

表 1 15 回のカリキュラム内容

Table 1 Contents of the 15-class

回数	内容
1 回目	プログラミングの導入
2 回目	分岐と繰り返し
3 回目	制御構造と配列
4 回目	手続き・関数と抽象化
5 回目	2 次元配列と画像
6 回目	プログラミングまとめ
7 回目	グループ作成・課題とゴール設定
8 回目	アイデア出し・プレスト・仕様決め
9 回目	デザイン・設計・分担・制作
10 回目	試作・中間ドキュメント作成
11 回目	中間発表・仕様見直し
12 回目	制作・単体テスト
13 回目	コードレビュー
14 回目	制作・統合テスト
15 回目	最終成果物(まとめ)

表 2 取り上げるテキストの例題

項目	内容
1.	前回演習問題の解説
1.1 演習 3a	四則演算を試す
1.2 演習 3b	剰余演算
1.3 演習 3c	円錐の体積
1.4 演習 4	さまざまな絵
2.	基本的な制御構造
2.1	実行の流れと制御構造
2.2	枝分かれと if 文
演習 1	
2.3	繰り返しと while 文
2.4 例題	プレーキのかかるボール
演習 2	
2.5	計数ループ
2.6	制御構造の組み合わせ
演習 4	
本日の課題 2A	
次回までの課題 2B	

答する。今回取り上げるテキストの例題について、表 2 に示す。

表 2 の 2.1 では基本的な制御構造である接続、分岐、反復の流れ図で示し、以降それぞれの項目で if 文、while 文、計数ループ、制御構造の中に制御構造が入ったプログラムを解説した。例題及び練習問題の全体は、表 A.1 に記載し、表 3 に、分岐を開発するうえで重要だと考える一部を抜粋する。

授業の進め方は概ねテキスト通りであるが、時間と学生の特性の都合上数学的な問題は飛ばし、教員が指定した演習 1c を授業内で行った。演習 1c のプログラムコードを、図 5 以下に示す。次回までの課題では演習 2 から 4 の問題を自由に 2 題選択し取り組んだ。

表 3 分岐テキスト内容抜粋

項目	問題 (作成するプログラム)
2.2 例題	絶対値の算出
演習 1	絶対値計算プログラムの好きなバージョンを動かす。動いたら、枝分かれを用いて、次の動作をする Ruby プログラムを作成する。
演習 1c	実数を 1 つ受け取り、それが正なら「positive」、負なら「negative」、零なら「zero」という文字列を返すプログラム
2.4 例題	転がってきたボールが止まる絵 (図 1) を描くプログラム
2.5 例題	固定回数ボールが描かれるプログラム (図 2)
演習 3	例題のプログラムを動かし、図 (図 3) のようなものを作る (a e)
2.6 例題	はね返りと特定番号で色が変わるボールを描く (図 4) プログラム



図 1 プレーキのかかるボール



図 2 固定回数の反復



図 3 演習 3(a) (e) の問題

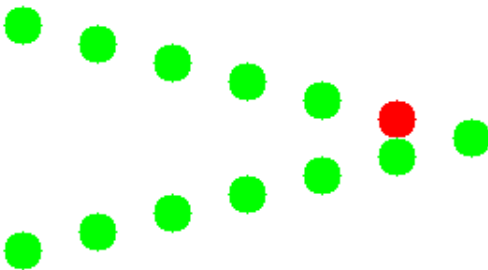


図 4 はね返りと特定番号で色変化

4. 分析の手法

提案にあたって、実体験のほかに、次の 3 つのデータを活用する。

```
def nzp(x)
  if x < 0
    result = 'negative'
  else
    if x == 0
      result = 'zero'
    else
      result = 'positive'
    end
  end
  return result
end
```

図 5 演習 1c のコード

(1) 2018, 2019 年度の履修者を対象とした分岐を学んだ際の授業内アンケート

- 枝分かれや繰り返しの動き方が納得できたか
- 枝分かれと繰り返しのどちらが難しいか、またそれはなぜか
- 制御構造の組み合わせができるようになったか

(2) 2018 年度から 2020 年度にかけての中間課題におけるループと分岐の活用度

(3) 2018 年度から 2020 年度に当授業を履修し、現段階でフェリス学院大学に在籍している学生を対象として実施したアンケート

(3) のアンケートについて、以下に示す。調査期間：2020 年 2 月 8 日から 10 日

調査対象者数：28 名

回答者数：14 名

〈項目〉

- テキストはどの程度読み込んだか
- 演習問題はどの程度取り組んだか
- なぜ最終課題で分岐を活用しなかったのか

以上の 3 項目を分析することによって、何が原因で分岐を使えていないのか探る。

5. 調査結果・分析

5.1 (1) の調査結果・分析

授業ではネガティブ・ポジティブを例題に授業を進めた。その際のアンケートの回答を、以下にまとめて示す。

- 回答者 10 名のうち、枝分かれや繰り返しが理解できたという回答したのは 8 名だった。
 - 回答者 12 名のうち、繰り返しよりも枝分かれの方が難しいと回答したのは 4 名だった。
 - 回答者 24 名のうち、制御構造の組み合わせができるようになったと回答したのは 13 名だった。
- 以上より、授業直後の段階では 8 割もの人が分岐やルー

表 4 年度別履修者数と課題提出数

年度	履修者数	課題提出数
2018	17	14
2019	16	16
2020	7	5
合計	40	35

プを理解したと思っていたこと、分岐よりもループの方が難しいと感じていたこと、半数は制御構造の組み合わせができるようになったと思っていたことがわかる。

5.2 (2) の調査結果・分析

まず初めに、中間課題の提出状況について、表 5 に示す。

提出された課題 35 件のうち、ループを使ったものは 27 件、分岐を使ったものは 1 件だった。5.1 では分岐を理解したと考えている人が多かったが、実際に用いている人は、ほとんどいないことがわかった。

5.3 (3) の調査結果・分析

(3) の調査結果について、一部は回答を基に著者が分類したものを以下に示す。

- 回答者 14 名の中でテキストを読み込んでいる人は 7 名、一度はすべて読んでいる人は 4 名、すべては読んでいない人は 3 名であった。
- 分岐が活用されなかった理由として、有効回答者 13 名の中では、理解ができていない人が 7 名、理解はしたが活用できない人が 4 名、使う必要がないと判断した人が 2 名という結果だった。
- 回答者 14 名の中で十分に演習課題を提出できなかった人は 2 名だった。
- 十分に演習課題を提出できなかった 2 名は、分岐の活用に関して、理解ができていないと回答した。

これらの結果により、1 度は理解したと思っても理解できていなかったり、活用できない人が多いことがわかった。また、演習課題に取り組まないという理解ができないことが結果として提示された。

5.4 著者の実体験

著者の経験として、演習 1c に取り組んだ時には分岐が理解できたように感じた。しかしながら、その後はループを学び、2 回目に関した分岐の問題は 2.6 の例題の分岐とループが混ざった問題であったため、分岐とループが頭の中で混ざってしまい、当日中の理解ができなかった。

次回までの課題ではループ単体の問題である演習 2a と演習 3a を選択したため分岐の理解は遅れ、中間課題時に理解、活用できるようになった。

著者が作成した分岐を用いたプログラムを図 5 に示す。

出力された結果を図 8 に示す。

また、当分析を行う上で、著者の場合は規則的に色を変

```
def hometown1
  fillrect(1600, 550, 3200, 1100, 149, 204, 207, 0)
  11.times do |i|
    fillrect(100+300*i, 550, 200, 100, 235, 97, 90, 0)
  end
  6.times do |i|
    if i % 2 == 1
      town(200+600*i, 200, 200, 222, 107, 55, 43, 64, 117, 74, 49, 36, 213, 235, 225, 0)
      town(-200+600*i, 900, 200, 222, 107, 55, 43, 64, 117, 74, 49, 36, 213, 235, 225, 0)
    else
      town(200+600*i, 200, 200, 183, 176, 172, 240, 195, 208, 102, 148, 108, 250, 238, 0, 0)
      town(-200+600*i, 900, 200, 183, 176, 172, 240, 195, 208, 102, 148, 108, 250, 238, 0, 0)
    end
  end
  writeimagef("hometown3.ppm")
end
```

図 6 中間課題のコードの一部



図 7 出力した絵

える図面を作成したため、分岐を用いたが、そもそも規則的な変化がある図面を描かなければ分岐を使う発想に至らないだろうと考えた。

6. 考察

分岐の授業終了直後では、理解できたと考えている人が多いが、実際に理解・活用できている人が少ない理由としては以下の事柄があげられると考える。

- 演習 1c が解けたことで、理解していると誤認した
- その後の分岐の演習回数が少なく、ループと混同したまま時間が過ぎ、忘れてしまった
- 分岐単体の絵の問題がないため、分岐を用いた図面を作成する発想がなく、結果として理解しても活用できないといった状況を引き起こした

7. 今後の学習に向けての提案

第一に、分岐に関する演習問題の増量が必要である。演習 1c を 1 問解いたのちに、分岐とループが複合した 2.6 例題を解くのではなく、もう一問分岐単体の演習問題を解いてから 2.6 に進むべきである。また、演習 1c は、分岐の仕組みはよくわかるものの、絵に直接結びつく課題でなく、2.6 の例題は、変化が一箇所のみであることから、分岐を活用した絵の想像が難しい。

したがって、規則的な変化が視覚的に分かりやすく、今後の図面作成の参考になるような絵を描く例題の追加を提案する。

理解するうえでは例題に連続性や反復要素があった方が定着することから、2.5 例題や演習 3 のような形状で 2 色、3 色を規則的に並べる問題であるとよいと考える。

第二に、現在のテキストの分岐を習ってからループを学



図 8 2色, 3色を規則的に並べる分岐の問題の例

表 5 提案する授業の進め方

項目	授業内容
1.	実行の流れと制御構造
2.	繰り返しと while 文
2. 例題	ブレーキのかかるボール (図 1)
3.	計数ループ
3. 例題	固定回数描かれるボール (図 2)
4.	枝分かれと if 文
4. 例題	実数を 1 つ受け取り, それ为正なら「positive」, 負なら「negative」, 零なら「zero」という文字列を返すプログラム
4. 例題 2	入力する数字によって出力される絵が変わるプログラム (図 8)
5.	制御構造の組み合わせ
5. 例題	3 で描いたボールが奇数回の時と偶数回のときで色が変わるようにする

ぶ構成を逆にし, 先にループを習ってから分岐を学ぶ構成へ変更することを提案する。ループの中に分岐をいれることも多いので, 先にループの知識を固めてから分岐に取り組んだ方が後の中間課題や最終課題での活用に効果的であると考える。以上二つを統合するとこのような授業の流れになる。

8. まとめ

本論文では, 著者や他学生の理解度や理解までの過程を分析し, 授業で取り扱う分岐の問題の少なさと, 分岐を使って絵を描く問題がないことが最終的に分岐が理解できない, 活用できないという状況を作り出していることがわかった。可視化した課題点を例題をふやす, 授業構成を変えることで改善し新たな授業内容の提案をした。

しかし, 制限条件として, 理解できない根本の原因が条件式などの基本的な算数, 数学の考え方がわからないなどプログラミング学習以前の事柄にあった場合, 十分な理解を促すことは難しいということも考えられる。

プログラミングは, プログラムを正しく完成させることだけがゴールではなく, プログラム作成を通じてプログラミング的思考を身につけることも重要な要素である。その点において, お絵かきプログラミングは, プログラムや数値を誤った際に間違いがわかりやすく, 出力結果が絵にあらわれるため, 正解に近づいていく過程を視覚的に捉えることができる。また, 図面作成によって設計についても学べるため, これからの社会に必要な要素を効率よく学べ [5], 今後の更なる活用が期待される優れた教材だと言え

るだろう。

本論文が, 誰もが理解できるプログラミング教育の確立の足掛かりになることを期待している。

参考文献

- [1] 首相官邸:世界最先端 IT 国家創造宣言 (2013 年 6 月 14 日)(オンライン), 入手先 (<http://www.kantei.go.jp/jp/singi/it2/kettei/pdf/20130614/siryou5.pdf>) (参照 2021-02-10).
- [2] 文部科学省:「超スマート社会における情報教育の在り方に関する調査研究」報告書 (オンライン), 入手先 (http://www.mext.go.jp/a_menu/koutou/itaku/1386892.htm) (参照 2021-02-10).
- [3] 永井 絵梨奈, 伊地知 咲希, 内田 奈津子:初学者がなぜループ処理でつまづくのか?—文系女子大におけるプログラミング入門の経験から—, 2020 年度 情報処理学会関西支部 支部大会 講演論文集 (2020).
- [4] 大岩元:プログラミング教育の社会的意義—設計から始めるプログラミング教育—, 情報教育シンポジウム 2016 論文集, pp.122-128(2016).
- [5] 伊地知咲希, 永井絵梨奈, 内田奈津子:プロジェクトにおける活動分析 —お絵かきプログラミングによる入門教育と PBL —, 2020 年度 情報処理学会関西支部 支部大会 講演論文集 (2020).

付 録

A.1 演習問題一覧

表 A.1 項目ごとの演習問題一覧

項目	問題 (作成するプログラム)
2.2 例題	絶対値の算出
演習 1	絶対値計算プログラムの好きなバージョンを動かす。動いたら、枝分かれを用いて、次の動作をする Ruby プログラムを作成する。
演習 1a	2 つの実数を受け取り、大きい方を返す
演習 1b	3 ないし 4 の実数を受け取り、最大のものを返す
演習 1c	実数を受け取り正か負か零かによって異なる単語を返す
2.4 例題	転がってきたボールが止まる絵 (付図 1) を描くプログラム
演習 2	2.4 例題をそのまま動かし、次のように直す
演習 2a	斜めに動かす
演習 2b	水平方向にはブレーキがかかるが、縦方向には一定の速さで動き続ける
演習 2c	水平方向には一定の速さで動き続けるが、縦方には徐々に早くなる (落ちる)
演習 2d	画像の下端に来たら跳ね返る
演習 2e	好きなボールの軌跡を描く
2.5 例題	固定回数ボールが描かれる (付図 2) プログラム
演習 3	例題のプログラムを動かし、図 (付図 3) のようなものを作る (a e)
2.6 例題	はね返りと特定番号で色が変わるボールを描く (付図 4) プログラム
演習 4	はね返りの例題をそのまま動かし、次のことを実行する。
演習 4a	間隔が一定でなく徐々に長く/短くなる
演習 4b	1 回だけ色を変えるかわりに「1 つおき」や「2 つおき」に変える
演習 4c	X 方向だけでなく Y 方向も跳ね返る
演習 4d	画像の上/左端でも跳ね返る
演習 4e	ボールが 1 つでなく 2 つになる
演習 4e	好きな跳ね返りの絵を作成する