

プログラミング入門をプロジェクトで やってみた

内田奈津子

フェリス女学院大学

プログラミング授業のハードルを下げるには？

プログラミング教育は、次期学習指導要領で小学校から高校まで各段階での実施が決まっている。これからの社会の構成員には、単にソフトウェアを使うだけでなく、ソフトウェアを作りその原理を知るといった形の知識や技能が求められている。

しかし、高等教育においては定められたカリキュラムはなく、文系大学では、プログラミングの授業を用意することは簡単ではない。

一方で、学生の卒業後の進路にはSEなども含まれており、プログラム作成だけでなく、グループ活動、仕様策定、進捗管理など、プロジェクトマネジメント（以下、PM）に相当する能力が求められる。現代社会においては、社会基盤としてITスキルはもとよりこれらの力はSE以外のさまざまな職場でも有効なはずである。

このような考えから、学生にプロジェクトを体験させることを過去に多く試みてきたが、情報技術にかかわるプロジェクトであっても、十分なプログラミング能力を持たないため、自ら考え何かを形作るというプロジェクトは単独ではできなかった。

このような経緯から、初学者の入門としてのプログラミングをプロジェクトとともに学べる授業デザインを考えた¹⁾。これにより、プログラミングを学ぶことのハードルを低くし、社会での実践に必要なスキルが身に付くようなデザインをした。

提案する授業の構成を表-1に示す。

授業では、次の4つを到達目標として設定した。

(1) プログラミング

例題を打ち込んで動かす体験にとどまるのではなく、考えたことをコードとして実現できる。

(2) PM

グループワークを経験し、有効なコミュニケーション、実践的なレポートやプレゼンテーションなどのスキルを身に付ける。

(3) 要求分析

目的にかなった開発のために「何が重要か」「どのような機能を実装するか」のとりまとめ作業を体験し、そのスキルを身に付ける。

(4) ITスキル

コードの編集やデバッグ、グループワークのための情報共有、成果物の作成などに各種のITツールを実際に活用しながら身に付ける。

筆者の勤務校ではプログラミングにそれほど時間を割くことができないため、90分15回（2単位）でプログラミング入門から始めてプロジェクトまでを扱う形とした。

表-1 15週の構成

週	内容
1週目	導入
2・3週目	分岐と繰り返し
4・5週目	制御構造と配列
6・7週目	手続き/関数と抽象化
8・9週目	2次元配列と画像
10週目	プログラミングまとめ
11週目	プロジェクト①グループ作成と構想
12週目	プロジェクト②デザイン決定
13週目	プロジェクト③コーディング
14週目	プロジェクト④統合
15週目	プレゼンテーションとまとめ

本稿では、2017年前期(4～7月)に2つの授業(Aクラス8名、Bクラス5名。文学部と国際交流学部)の2～4年次生)で実施した結果を紹介する。

この授業では前半で演習を取り入れたRuby言語のプログラミング入門を行い、ビットマップ画像が作成できるところまで学ぶ。続いて後半で数名ずつのグループでプロジェクトを組み、各グループで話し合っ絵の作品(実際には動画作品となった)を構想・設計・制作する。

特に工夫した点は、次の2つである。

- 必要最小限の内容しか教えない
- 課題設定をし、学んだ知識と技能をグループワークで実践する

プログラミングだけ学ぶのであれば、オンライン教材での自学も可能であるが、教室に集まって学ぶ価値として、グループワークを取り入れた。

前半：プログラミング入門

この授業は初めてプログラムを学ぶ学生が対象であり、後半のプロジェクト部分までに「自分で思った

表-2 プログラミングを学ぶ際の指針(抜粋)

授業の進め方	よくない点
	<ul style="list-style-type: none"> ・教科書に書いてあるたくさんの知識をそのまま覚えることを求められる。 ・教科書に載っているような練習問題は「どんな問題がある」「どうやって解く」をドリルなどで繰り返し練習させられ覚えさせられる。 ・試験のときはその「覚えた」やり方で短時間に多くの問題を解くことを求められる。
改善点	<ul style="list-style-type: none"> ・学生は「プログラミングの最小規則」を理解し、それらの規則をさまざまな方法で自ら繰り返し練習する必要がある。 ・学生は自分のスキルや興味に応じて取り組む問題を選択する必要がある、結果としてより良いモチベーションが得られる。 ・学生は「正しい」プログラムを書くことに多種多様があることを理解し、解決策は1つではないことを理解する。
教科書	よくない点
	<ul style="list-style-type: none"> ・テキストにはプログラミング言語の規則(書き方、機能)が逐一解説されていて、授業でもそれを順番に学んでいく。 ・演習問題が「プログラムを書くこと」でなくその説明している内容の知識を覚えたかどうかの確認問題になっている。 ・プログラムの例題が少ししか登場せず、その少しの例題を懇切丁寧に説明している。
改善点	<ul style="list-style-type: none"> ・最小の経験則が説明され、学生はさまざまな演習を通してそれらのルールを適用することを推奨する。 ・さまざまな困難を伴う多くの練習問題が用意されていて、学生は自分のレベルや興味に応じて解決するための問題を選択。 ・多くのトピックとサンプルコードが提供され、学生は興味を引くトピックを選択。

ようにコードを書く」力を養う必要がある。その指針として、文献2)で挙げられているものを採用した(表-2)。テキストも文献2)の著者に執筆いただいた。

内容構成は表-1の前半(1～10回)となる。

各回は「説明→演習→時間内の課題→次回までの課題」の順で実施されるようになっている。各回の課題は、非常に平易なもの(例題を少し手直しすれば提出できる)から高度なものまで用意されており、学生は自分のレベルに応じて選んで解答する。

これにより、レベルに応じた学びを得ることができ、「例題をコピーして動かす」のではない、自分なりのコードを書くことが可能となる。

大部分の学生が、レベルの高低はあるがそれぞれ自分なりの絵を生成するコードが書け、後半のプロジェクトの分担をこなせるようになった。

後半：グループによるプロジェクト

後半部分は主としてプロジェクトを体験し、その力をつけることが目的となる。「どのような作品を作るか」を決めるプロセスで要求分析を体験する内容がそこには含まれる。また、グループ活動のためにITツール群の活用も多く必要となる。

プロジェクトで行うグループ作業の内容と構成を表-3に示す。これらの活動を通して、PMについて多くのことを体験的に学ぶ。

授業では学生が最低限のプログラミングスキルを身に付け、実際に作品を制作する形をとったことから、「自分たちに可能な範囲でどのような作品を作るか」という要求分析の体験も行うことができた。加えて、プロジェクトに最初から自分たちで取り組

表-3 プロジェクトの流れ

Step	内容
1	グループを作り、リーダーと各メンバの役割を決定する
2	テーマ(主人公)とアニメーションの構造を決定する
3	作成時の各メンバの役割を決定する
4	スケジュールと統合計画を決定する
5	プログラムを1つずつ作成する
6	各部品を統合し、欠陥を修正する
7	プレゼンテーションを作成してプレゼンテーションを行う



むことで、スケジュールや仕様の策定、制作体制などにおいて、「誰かに作ってもらう」「提案だけする」のでは得られないリアルな学びが得られた。

授業の様子

2017 年前期（4～7月）開講の A・B クラスは、外国人客員教員が英語で行った。

テキストやその他の資料は LMS (Moodle) で配布し、それを使って（授業ごとの個人の理解についての議論）、授業の課題、学生の宿題を収集した。

前半のプログラミングの部分は、テキストに基づいて授業を行った。当初は 5 回の予定であったが、授業が英語で行われ、さらに教員間の若干の認識のずれから 10 回をあてることとなった。

11 回目以降のプロジェクトでは、クラス A を 2 つの小グループ (A1 と A2)、クラス B は 1 つのグループ (B1) に分けた。最初にプロジェクトにおけるタスクと目標設定について説明し、最終のプレゼンテーションの要件を提示した(図-1)。

グループ内で、図案を出し合い、図面の構成を議論し、要件を定義した。また、各自の役割分担とスケジュールを考えプロジェクトに取り組んだ。

Schedule

Final presentation of the project inclass

July 24, 2017

What to present

- The output or the drawing
- A discussion of the role of members in the group in creating the drawing
- Schedule, a discussion of the time spent in the development of the drawing project
- Review of the code, wherein you will discuss how you implemented the drawing
- Challenges in the development of the project and how it is being answered

図-1 授業時の提示資料の一部



図-2 学生の作品の一部^{☆1}

^{☆1} 課題として作成した作品は、許諾のためにここに掲載できないため、学生本人に依頼して同水準のものに変更

なにができたのか？

図-2 は、各グループの成果として作られたアニメーションの静止画である。

グループ B1 の実際のプレゼン資料を元にプロセスを説明する(図-3)。

最初に学生たちは、メインキャラクタを決め、イメージを起草し、統合する背景を決定した。

ついで図-4 のようなマトリックスのメモを作り、作業を進めていった。

こうしたプロセスを通し、各グループで 5～10 枚の絵をプログラムし、アニメーションに仕上げた。各グループの状況を以下に示す。

グループ A1 (3名) :

基本の絵(パンダの顔)を各自が作り、その中で一番良いものを選んだ。パンダの顔は簡潔であるため、この方法はメリットもあったが、複雑な画像には適していないことを認識し、受講生はそのメリットとデメリットを理解した。その後、各メンバに異なる役割を割り当て、作業を進めた。

複数のコードを結合するのが難しかったと学生たちは述べている。この解決のために、コード内のコ

Proceeding

1. Discussion
2. Deciding our design
3. Drawing
4. Discussion
5. Programming
6. Discussion
7. Adjustment




図-3 グループ B1 のプロセス (発表資料より)

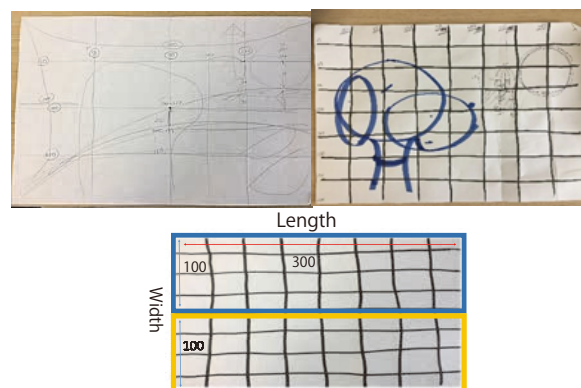


図-4 作業時のメモ

メント方法とその効果について学んだ。

グループ A2 (4名) :

メンバー間にスキルの差があり、熟練したメンバーが複雑な部分を担当した。その後、各自がほとんど独立して作業を行った。グループ活動のオーバヘッドが小さかった。他方で、統合は大部分が1人に委ねられ、他メンバーは、リーダーに完成したプログラムを送るだけであったため、リーダーの負担が大きくなった。また、グループ内の協力関係が弱かったためか、プレゼンテーションの準備が難しかった。

グループ B1 (5名) :

3つの部分に分割し、分担して作業を進めた。ユニークな点は、メンバーが3つのパートに分かれ、3人でメインキャラクターを作成した点である。

グループ内コミュニケーションが良く、プロジェクト進行がうまくいった。一方、初期段階で失敗が

あった。描画の出力のみを保存し、コードの保存を忘れ、再度プログラムすることとなったが、それが学びとなり後半の活動にプラスの影響を与えた。

まとめ

半期でプログラミングとプロジェクトを行うという欲張りなカリキュラムであったが、どのグループも期限内にプロジェクトを完了でき、学生たちは当初の目的を理解したと総合的に評価できる。

多くの授業が1人でやる作業が中心で、受講生同士が協力して共同作品を完成させるということがない。プロジェクトの課題を短いアニメーションにしたことの効果は大きかった。絵を描くために必要な最低限のプログラミングを学び、それを活用してアニメーションを作成するという非常に簡潔な課題ではあるが、動画作成プロセスを通じて共同のプロジェクトの進め方を実践的に学んでいる。

授業アンケートの回答を表-4に示す。

“当初難しいこともあったが、最終的には達成感が得られた”と述べており、今後の励みとなるに違いない。一方で、プログラミングについては、サンプルプログラムに頼り過ぎであったとか、応用は難しいと言っている。次につながる学びの機会の工夫は、今後の課題の1つである。

参考文献

- 1) Uchida, N., Montenegro, C., Kuno, Y. and Kakehi, K. : Importance of Course Content and Classroom Design in Project-Based Learning in Programming Courses - Case Study of a Liberal Arts College, Ferris University in Japan -, Philippine eLearning Society (PeLS) , 3rd International Congress on eLearning 2017, Hanoi, Vietnam (Oct. 2017).
- 2) 久野 靖: 何のためにプログラミングを学ぶの? そしてどのように?, <http://lecture.ecc.u-tokyo.ac.jp/~ckuno/is16/sprosym-2016.pdf> (2017年9月1日閲覧). (2017年10月11日受付)

表-4 授業アンケート回答

項目	学生の回答
プログラミング	プログラムを動かすための命令を、Rubyの言葉を使って作成し動かすことができました。条件の付け方や繰り返しのやり方などのコードを学習し理解しましたが、実際に応用する能力はまだ身に付いていないので、今後も、自主学習に励みたいと思います。
	Rubyの基本は書けるようになったと思います。しかし、サンプルに頼りすぎの場合もありました。
ITスキル	GoogleドライブとLINEによる情報共有を行い、スケジュール調整が厳しいときでもスムーズにほかのメンバーの進捗状況を把握できました。
	データ共有と保存については、すべてクラウド上で行いました。授業時間外で作業をしたときなども進行状況をクラウド上で知らせることができ追加コードの挿入もスムーズに行えたと思います。ICT機器を利用したグループワークは、初めてだったのですが、クラスポータルやクラウド上に保存してある情報はいつでもスマートフォンやPCを使って見ることができたので、予習や復習が行いやすかったです。
その他	ほかの授業ではあまり扱わない専門用語の英語が分からず、最初は苦しみましたが、だんだんと分かるようになってきて、最後までやりきることができたので、今は達成感でいっぱいです。
	課題を行うときに、先生のコードを写してプログラムを動かしたり、先生のコードをもとに応用ということはできたのですが、方法を自ら考えてコードを書くということが、なかなかできませんでした。この授業では、問題解決型の学習が多く、分からないときは、先生に尋ね、アドバイスをもらうことができたのですが、もう少し自分で考えてみるという努力が必要だったかなと感じています。
	自分が担当したところは、ひたすら丸を重ねる作業で「名前をつけて保存」の名前も勝手に決めていましたが、全体で合わせるときに名前を合わせなければいけないことに気づき、いろいろなプロセスを踏んで1つの動画ができあがっていることを実感しました。

内田奈津子 (正会員) uchida@ferris.ac.jp

フェリス女学院大学情報センター講師、電気通信大学情報理工学研究科在籍中。情報教育を中心に、学内システムの運用管理に従事。

