第一工業大学研究報告
第33号 (2021) pp.13-24

メタ学習の支援による プログラミングの自己学習能力の向上を目的とした 大学授業の検討

渋沢良太

鹿児島県霧島市国分中央1-10-2 第一工業大学 工学部 情報電子システム工学科

E-Mail: r-shibusawa@daiichi-koudai.ac.jp

A Study of University Classes for Improving Self-Learning Ability of Programming by Supporting Meta-Learning

Ryota SHIBUSAWA

Department of Informatics and Electronics ,1-10-2, KokubuChuo, Kirishima, Kagoshima, 899-4395, Japan

E-Mail: r-shibusawa@daiichi-koudai.ac.jp

Abstract: In response to the demands of today's society, programming classes are now being expanded in elementary schools, junior high schools, high schools, universities, and other educational institutions around the world. However, the progress and changes in information technology are dizzying, and many of the skills acquired in educational institutions become obsolete quickly. Therefore, it is almost impossible to use the programming skills acquired in four years of university without updating them after graduation. With this in mind, this paper proposes a guideline for the content of classes that aim to help students acquire the ability to continuously update their programming skills. In addition, we will report on the practical examples of classes in Introduction to Programming I and Introduction to Programming II, which are offered in the first and second semesters of the first year of the Department of Information and Electronic Systems Engineering, respectively, and on the survey of students' educational effects by questionnaire.

Key words: Meta-learning, Programming learning, Metacognition, Computer science education

1. はじめに

現在の社会において，情報技術は企業等の組織の活動，個人の生活にとって欠かせないものとなっており，その重要性は年々増している．プログラミングのスキルを修得した者がごくわずかであった時代には，企業で行うプログラミングは，顧客に販売するシステムの開発が多くの割合を占めていた．プログラミング言語やその開発環境，動作環境の進化に伴い，近年では，業務の効率化，改善のために，顧客に販売するシステム開発以外を主な業務とする者もプログラミングを活用するように変化してきている．データドリブンマーケティング，CRM(Customer Relationship Management)，D2C(Direct to Consumer)等，近年社会でのニーズが特に高いデータサイエンス，EC(Electronic Commerce)の実践においても，プログラミング及びそれに関する技術は欠かせない．

このような時代の要請を受け，小学校，中学校，高等学校，大学等の教育機関において，プログラミングの授業が拡充されつつある．しかし，情報技術の進展と変化は目まぐるしく，教育機関で修得したスキルの多くは陳腐化してしまうのも早い．したがって，大学の4年間で修得したプログラミングのスキルを，卒業後にアップデートすることなく一生涯役立てることはほぼ不可能である．そのため，学生が社会人として企業等に勤務し，プログラミングのスキルを継続して企業活動に活かすためには，十分なスキルを修得済みであることに加え，そのスキルを継続してアップデートする能力を修得している必要がある．

上記の問題意識のもと，本論文で述べる学習支援手法では，大学のプログラミングの授業において，プログラミングの知識やスキルを学生が修得することだけを目的としない．それに加えて，学生がプログラミングのスキルを自ら継

続してアップデートする能力を修得することを目的とし，そのために学生に学習させるべき内容についての一つの指針を提案する．また，本学の情報電子システム工学科1年次の前期，後期にそれぞれ開講されているプログラミング入門I，プログラミング入門IIにおける，授業の実践例，アンケートによる学生の教育効果の調査について報告する．

2. メタ学習させる内容についての指針

学習についての学習のことを，本論文ではメタ学習(Meta-learning)と呼ぶ．筆者は，学習者が学習内容そのものを学習するだけではなく，メタ学習することによって自身の能力を卒業後も継続してアップデートする能力を修得できると考えている．本章では，そのようなメタ学習において，学生に学習させるべき内容についての指針を示す．

2. 1. プログラミングで修得すべき能力と学習方法についての学習

プログラミングの能力には，図1に示すように，大きく分けて知識と実践能力(スキル)の2種類があると考えられる．プログラミングの文法や，制御構造，オブジェクト指向等の概念を学ぶのは，プログラミングの“知識”の学習である．“知識”だけを修得していても，学生自身が開発したいソフトウェアや，顧客の要望を満たすソフトウェアは開発できない．

それを実現するために必要な，もう一つの種類の能力が“スキル”である．教科書を見て，そこに書かれているコードを同じように打ち込んで実行するだけでは，“知識”の学習に留まってしまう．学生が“スキル”を修得するためには，自らが実現したいと考えたソフトウェア，ないしその一部の機能をどのようにしたら実現できるかについて，学生自らが考え，必要なことを学び，問題解決することが必要である．何かの資格を取りたいと考えている学生が多いが，資格取得のための学習によって修得できることは，多くの場合“知識”であり，“スキル”はほとんど含まれないことに注意すべきである．資格取得だけで学習を満足していては，ソフトウェアを開発する能力は一向に身につかない．

教員が一方的にプログラミングの知識やテクニックについて説明する形式の授業では，学生が“スキル”を修得することはできない．教員があえて丁寧に教えずに，学生自らが手を動かして考える時間を作ることが重要である．2011年から2019年までの間MIT MediaLabで所長を務めたJoi Ito氏は，MIT MediaLabにおける教育の方針として“Learn through construction rather than instruction.”[1]を掲げていた．この方針は，学生がプログラミングのスキルを修得するためにも重要な方針である．

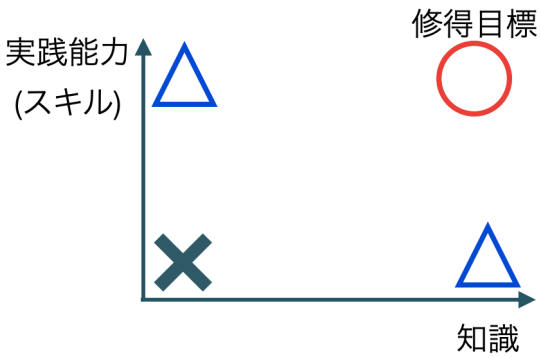

図1　プログラミング学習の修得目標

上記の内容について学生に認識させ，教員が学生に対して話すプログラミングの知識は最低限にして，学生自らが手を動かしてプログラムを構築し，問題解決させることが望ましい．また，上記の方針で授業を展開すると，教科書には書かれてない，学生のオリジナルのプログラムが，学習のアウトカムとして生成される．このアウトカムは，後に学生が学習のディブリーフィングをする際や，企業などに提出するポートフォリオの作成時にも利用することが可能である．そのため，そのアウトカムは学習管理システムにアップロードしておく等，学生が必要な時に，いつでも見られるようにすることが望ましい．特にゲーム開発会社等のソフトウェア開発会社では，自社に就職を希望する学生に，これまでに開発したソフトウェアおよびそれらのポートフォリオを提出させる機会が多い．

2. 2. 大学における学習と研究の違いの学習

“学習”は，書籍やインターネット，大学の授業等で学ぶ活動である．一方，“研究”は，教科書に載っていない，誰にも発表されていない新しい知見を生み出す活動である．

大学における“学習”は，いわば“研究”の部分集合とみなせる活動であり，インプットする活動である．高等学校までの学習は，大学受験のための学習が主な内容であった．その過程においてはテストも実施されるが，テストはアウトプットする活動ではなく，学習内容を正しくインプットできているかを検証する活動とみなすことができる．

一方，“研究”はアウトプットすることが主な活動である．論文や学会で発表するばかりが研究ではなく，実は日常生活において生きるた

めの工夫として，個々人はそれぞれのより良い行動選択についての研究を無意識のうちに実践しているはずである．このように“学習”と“研究”は別の概念とみなせるのであるから，“学習”ができたとしても“研究”ができるとは限らない．企業が行っている事業活動も，より利益を生み出すための研究とみなせる．従って，社会に出てからより重要視されるのは，インプットする活動よりもアウトプットする活動，すなわち研究である．

上記のように，学生にインプットする活動とアウトプットする活動の違いを認識させ，“学習”にとどまらず，社会においてより重要視されるアウトプットする能力を身につけることを目標にさせることが望ましい．

2. 3. 理系と文系の意味の学習

日本の社会においては，理系，文系という学問，能力の区分けが良く用いられている．この区分が，学生の能力を向上させる妨げとなることがある．例えば，「自分は理系だから」といって，文系とみなせそうな学問に苦手意識を持ったり，反対に「自分は文系だから」といって，理系とみなせそうな学問に苦手意識をもったりしてしまう．

理系と文系は本来受験のテスト内容の区分である．なぜその区分を大学で学ぶ学問や職業の区分として適用する必要があるのだろうか？また，その区分は学問や職業を適切に区分できるのであろうか？図2に示すように，大学以降で学ぶことには理系でも文系でもあるとみなせるものや，理系でも文系でもないとみなせるものも数多くある．また，理系，文系とみなせるものよりも，その両者ともみなせないものの方が圧倒的に多い．従って，“理系”，“文系”という区分は，学問や職業を適切に区分できないのである．またこの区分にとらわれてしまうと，

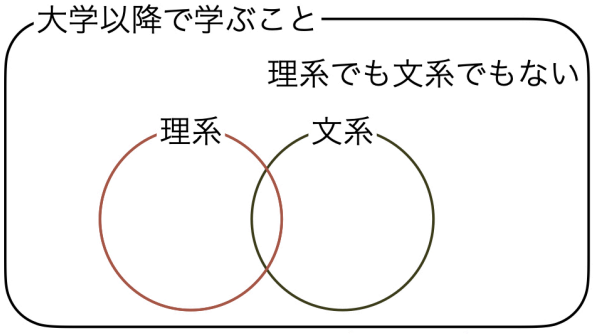

図2 大学以降で学ぶ内容

その区分の外に出ないように学問や職業の内容が制限され，新しい学問や職業を生むイノベーションを阻害してしまう恐れがある．

大学生の多くは，入学時点において自分自身が理系か文系のどちらかであるという先入観を持ってしまっている．上記の内容について学習させ，その先入観を取り払った学習態度を身に着けさせることが望ましい．

また，学習の目標として資格取得を設定する学生も多いが，資格取得のための学習は，学習内容の範囲を制限してしまう恐れがあることを認識する必要がある．それは，資格試験が出題範囲を明確に定めてしまうためである．資格試験の出題範囲に収まった学習を行うことで，果たして目まぐるしく変化する社会のニーズに答えられる能力を身につけられるだろうか？学生には，大学での学習は，新しい知見をアウトプットするための学習として認識させるようにすることが望ましい．その上で，制限された学習ではあるが，企業などへの能力のPRとして資格取得の学習を行うことは悪くはないであろう．しかし，学習内容を資格試験の範囲に制限してしまうことは，学生の能力向上の阻害要因になりうる．同様に，授業という枠によって，学習の範囲を学生が狭めてしまう恐れもある．学生に対しては，このように資格や授業によって無意識に学習の範囲を制限してしまっていることを認識させ，資格や授業の枠にとらわれない学習を目指すようにさせるのが望ましい．

2. 4. 学習のモチベーションについての学習

2. 4. 1. 学習の継続過程におけるモチベーションと能力の向上の関係についての学習

まず，学生が積極的に学習しない理由として最も考えられることは，「やる気が起きないから」である．これに関して，「やる気が起きないからやらない」という考えは間違っており，「やり始めないとやる気は起きない」ということを学生に認識させる必要がある．何事も始めてみないと，その物事に対しての先入観によって人は楽しいかそうでないかを判断してしまう．学生が一見楽しくなさそうに思えることでも，実際に行ってみると楽しくなることも多くあるのである．

乳幼児は，様々な能力を短期間で急激に修得していく．乳幼児は先入観を持たずに，感覚器官を通じて知覚されるもの全てに興味を持ち，何事も体験しようとする．これによって，楽しいと感じることを発見していくことができ，楽しみながら様々な能力を獲得していく．学生もこのような経験を通して成長してきているた

め，原点に立ち返って何事にも興味を持ち，学んでみようとする姿勢になることが，大学で学ぶ態度して重要になる．

次に，学生が学習を継続しない理由として最も考えられることは，「学習を初めてみたが，面白くなかったため」や，「難しくなって行き詰まったため」である．これに関しては，図3を用いて学生に理解させると良い．ある対象について学習を始めたばかりの時は，その対象で用いられる最低限の概念やルールを覚える必要があるため，あまり面白いとは感じない．しかし，それでも学習を継続してその概念やルールを覚えて活用できるようになると，急激に楽しく感じられるようになる．その後，能力が伸び悩むと楽しく感じられなくなる期間も生じるが，それでも学習を継続して能力が向上すると，また違う楽しさが生じるといった過程が繰り返される．このことは，どの学生も自身の人生を振り返させると思い当てる節があるはずである．例えば，トランプ等のゲームをする際，そのゲームのルールをまず覚える必要があるため，ルールを覚えるまではあまり面白いとは感じず，その後，図3のような過程で能力と楽しさが遷移していたであろう．

上記のことを学生に伝え，何事も先入観を持たず学び始めてみる，学習を継続すると楽しさと能力は長期的には向上するが，短期的には低下することもあることを認識させることが望ましい．そして，局所的なつまらなさに陥り，学習をやめてしまうことがないよう認識させることが望ましい．

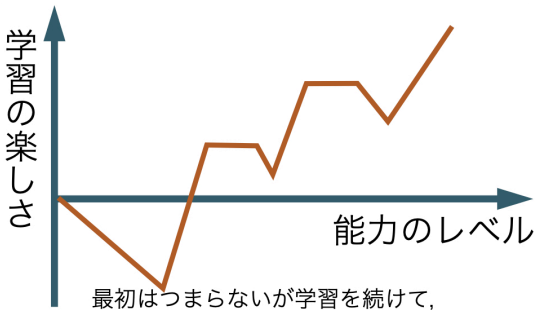

図3　学習の楽しさと能力の向上の関係

2. 4. 2. 学習モチベーションを向上させる方法

プログラミングを学習したいと考えている学生には，「プログラミングは役立ちそう」といった考えや，「すごいことが実現できる」といった憧れを持っている学生も多くいる．これらの考えは正しく，学生の学習モチベーションの源泉として認識し，学習モチベーションの向上に活用することが望ましい．

プロラミングが社会にどのように役立っているのか，プログラミングを使って実現されている最先端の研究や製品，サービス等について，学生が知らなさそうなことを適宜紹介することは，学習モチベーションの向上に繋がると考えられる．そのため毎回の授業の中で5〜10分間程度は時間を取り，このような事例を紹介することで，学生が今学習していることを継続することで，どのようなことが実現できるようになるかを認識させることが望ましい．

2. 5. 対象の学習と対象外の学習との関係性の学習

学生に対して，プログラミングの学習と，それ以外の学習との関係性について学ばせるのが望ましい．これによって，他の授業科目の学習のモチベーションを高めることが期待できる．

プログラミングの学習を進めることによって，高度なソフトウェアを実現する能力の一部が修得できるであろう．しかし，ソフトウェアを利用する人間に対して，そのソフトウェアのサービスをシステムとして提供する際には，センサや電子回路，ネットワークや機構等，優れたハードウェアの実現も必要になることが多い．従って，そのような優れたシステムを開発できるようになるためには，プログラミングだけでなく，ハードウェアについても学習が必要となる．また，プログラミングによって実現するソフトウェアは，コンピュータというハードウェア上で動作するため，ソフトウエアの性能を高めるためにはコンピュータアーキテクチャ，それを実現する基盤技術となっている物理学，およびコンパイラ，インタプリタ，仮想マシンを理解する必要がある．また，アプリケーションソフトウェアは，ミドルウェア，オペレーティング・システムによって提供された機能を利用して動作するため，ソフトウェアの性能，安全性を高めるためにはそれらの基盤ソフトウェアについての理解も必要になる．

また，プログラミングによって，特定の機能を実現したい時，プログラミングの文法を学習するだけではその機能は実現できず，アルゴリズムや数学によって数式を導出する必要がある．すなわち，プログラミングの言語仕様を学ぶことと，処理の実現方法を学ぶことは一致しないのである．例えば，3章でも紹介しているプログラミング入門Iの授業で課題として出している，x人の中で同じ誕生日である2人がいる確率を計算するソフトウェアを実現しようとした時，その計算式を導出できなければ，いくらプ

ログラミングの言語仕様を学んでも，そのようなソフトウェアを開発することはできない．

プログラミングは，数学における計算とある種の同型性を有している．最も簡単な例では，プログラムはコンピュータに対する記号列によって構成されており，その記号列中の記号を，タイプミスによって別の誤った記号に変えてしまうと，多くの場合目的の動作をしなくなる．数学においても，数式は記号列によって構成されており，数式中の記号を別の記号に誤って変えてしまうと，多くの場合数式が全く別の意味を持つものになってしまう．従って，数学の学習で計算能力を高めること，公理を正しく認識して，推論規則を正しく適用して定理を演繹する能力を修得することと，プログラミングにおいて目的の処理を正しく記述できるようになる能力を修得することは表裏一体なのである．数学が得意な者が，プログラミングも得意であることが多い理由には，このような性質も関係しているであろう．数学は基礎科学であり，「何の役に立つか分からない」といった印象を持つ学生も多くいると思われるが，このように数学を学習することは，プログラミングを学習することでもあることを学生に認識させることが望ましい．

また，プログラミングはソフトウェアを実現する手段であり，どのようなソフトウェアを開発すると利益をあげられるのか，社会の役に立つか等は，プログラミング以外の社会科学等の学問分野の学習によって理解を深められる．例えば，イノベーションの手法，経営戦略論や組織論等の経営学，マーケティング，社会課題の認識等が有用であろう．

以上を学生に認識させることで，優れたソフトウェアを開発できるようになりたいと考えている学生に対して，数学，アルゴリズム，コンピュータ・アーキテクチャ，ネットワーク，オペレーティング・システム，電子回路等の各種ハードウェア，社会科学等，プログラミングに関わりのある授業科目の学習意欲を高められると考えられる．

一般的に，ある学問分野について理解を深めるとき，その学問分野だけを学ぶだけでは不十分であり，対象の学問分野とそれ以外の学問分野との関係を学ぶことによって，対象の学問分野についての理解を深めることができる．また，対象の学問分野は，その学問分野の中身のみによって規定されているのではなく，他の学問分野との関係性によって規定されている側面もある．そのような関係性を学ぶことによって，対象の学問分野がそれ以外の学問分野，および社会的な課題の解決においてどのような役割を果たすのかを理解できるようになる．そして複数の学問分野を学ぶことによって，システムを実現する知識，スキルを修得できるようになるのである．

2.6. **Second-order**メタ学習

メタ学習，すなわち学習についての学習をFirst-orderメタ学習とした時，本論文では，メタ学習についての学習をSecond-orderメタ学習と呼ぶ．すなわち，この節で述べているSecond-orderメタ学習は，学生に対してメタ学習を学習させることを示している．

学生に対してメタ学習という考え方を紹介し，学生自身の学習をより良いものへと導くために，学習方法を工夫したり，自らの学習に対するモチベーションをコントロールする方法を学ぶことの重要性を認識させることが望ましい．大学の授業で学ぶ内容は忘れやすいと言われているが，学習方法はそれよりも忘れにくいと考えられる．学習内容を忘れてしまったとしても，優れた学習方法を修得できていれば，また必要な時に必要な内容を再度学んで修得することが可能であると期待できる．

3. プログラミング入門**I**の実践

プログラミング入門Iは，本学情報電子システム工学科のカリキュラム改定によって，2020年4月から本学情報電子システム工学科の1年生向け授業として前期に開講されている．本章では，2020年度前期のプログラミング入門Iの実践例を述べる．

3.1. 授業の概要

Scratch3.0[30]を用いて，プログラミングの学習及びメタ学習の支援を行った．履修登録者数は47名，そのうち単位を取得できた者は39名であった．COVID-19の感染予防対策を試行錯誤する時期において授業が開講され，第1〜2回，第11〜15回はZoom，第3回から10回は対面授業の形式で授業を実施した．

第1回から10回までの授業では[2]の文献を教科書として用い，各学生にそれぞれのペースで教科書の内容を参考にしてシューティングゲームを開発させた．本授業では，プログラミングに関する高度な知識の学習よりも，2章で述べたメタ学習に重点を置いたため，プログラミング言語，教科書の文献に平易なものを採用した．また，プログラミングの学習の初学者は，文法エラーで躓くことが多く，文法エラーばかりに気を取られていると，プログラミングの全体像，

プログラミングのメタ学習について理解する前に挫折してしまうことが多い．そのため，完全に文字列を入力するプログラミング言語ではなく，用意されたブロックを組み合わせることにより，文法エラーを減らせるScratchをプログラミング言語として採用した．教科書の内容に物足りない学生に対しては，より高度なプログラム開発の課題を与え，個々の学生のレベルに応じて学べるようにした．

第11回から13回までは，学生に個人またはグループでオリジナルのプログラム，およびその説明資料をプレゼンテーションソフトで作成させ，第14回，第15回の授業で他の学生の前で発表させた．このオリジナルのプログラムのテーマは自由とし，ゲームやアート，計算など何でも良いが，Scratch 3.0を使用することとした．また，チームで共同してプログラミングを行う経験も重要であるため，チームで課題に取り組んでも良いものとした．そして，ただプログラムを作るだけでなく，それがどのようなものなのか，どのように作られているか，工夫したことについて他の学生に説明する資料を作成させ，発表させるようにした．これにより学生は，自らが課題を通して学んだことを振り返り，整理することができる．また，最終課題で作成したプログラムの説明資料，およびそのプログラムのソースコードはMoodle上で履修登録者全員で

表1 プログラミング入門Iの授業の課題

回	目標	課題
1	Scratch3.0の基本的な操作方法の理解	繰り返しドラムの音を鳴らすプログラムの開発
2	画像のスプライトへの反映方法，キーボードでスプライトを動かす方法の理解	シューティングゲームの自機を左右キーで移動させるプログラムの開発
3	スプライトのアニメーションの理解．変数，繰り返し処理による計算の理解．	自機がロケット噴射するアニメーションの実装．(早く終了した人向け：x人の中に，同じ誕生日である2人が存在する確率を求めるプログラムの開発．)
4	座標指定によりスプライドを移動させる方法の理解．条件分岐を用いた計算の理解．	回転しながら移動する敵キャラのプログラム開発．(早く終了した人向け：x年がうるう年か否かを判定するプログラムの開発．)
5	乱数，スプライトの複製方法の理解．並列処理，リスト，スタティック変数，インスタンス変数の理解．	乱数を使ってランダムな位置から繰り返し動く複数体の敵キャラのプログラム開発．(早く終了した人向け：かえるの歌のメロディを，3匹のかえるが追従して演奏するプログラムの開発．)
6	当たり判定方法，メッセージパッシング，マルチスレッドプログラミングの理解．機械翻訳，音声合成の機能の理解．	自機から弾丸を発射し，敵キャラに当たったら敵キャラを消滅させるプログラム開発．(早く終了した人向け：機械翻訳，音声合成による，指定した言語から別の言語への翻訳，発声プログラムの開発．)
7	効果音とBGMの再生方法，画像効果の使い方，関数の定義と使い方，タイマーの使い方の理解	ゲーム開始とともにBGMが再生され，弾丸が敵キャラに当たると衝突音とともに爆破アニメーションが再生されるプログラムの開発．(早く終了した人向け：タイマーを使った，一時停止・再開機能付きストップウォッチの開発．)
8	コメントアウト，デバッグの方法の理解．プログラムの状態設計，状態遷移図の理解	ゲームのスコアと残機数を記録し，表示するプログラムの開発．
9	クラウド変数，三角関数の利用方法の理解	ハイスコアを記録して永続化するプログラム，敵キャラを三角関数を使って移動させるプログラムの開発．
10	複雑なアニメーション，再帰プログラム，フラクタルの理解	複数の異なる種類の敵キャラ，ボスキャラを動作させるプログラム，シューティングゲームの完成版の開発．(早く終了した人向け：再帰処理によるSierpinskiのギャスケットを描くプログラムの開発．)
11-13	作りたいアプリケーションを実現する方法の理解	オリジナルのアプリケーションの開発
14-15	自らが作成したソフトを分かりやすく説明する方法の理解	オリジナルのアプリケーションの発表

表2 プログラミングを活用した研究・製品の事例紹介

No.	研究内容
1	ICTによる農業支援（薬剤散布ドローン[3]，アスパラガス自動収穫ロボット[4]）
2	リハビリ支援ロボット（ロボットスーツHAL[5]）
3	没入型HMDを使った野外での道案内（DreamWalker[6]）
4	パーキンソン病患者等，手が不自由な人の食事を支援する震えの低減スプーン（Lifeware Steady[7]，スプーンと床の平行を保つジンバルスプーン（Lifeware Level[8]）
5	視覚障がい者の屋外歩行における，障がいしゃ及び周囲の人への衝突危険通知（BBeep[9]）
6	ニューラルネットワークを用いた無発声インタラクション[Silent Speech](Glove talkII[10], SottoVoce[11], Derma[12], Speech synthesis from neural decoding[13])
7	デジタルファブリケーション（3Dスキャナ，CAD，3Dプリンタ），食品の3D printing（FoodFab[14]）
8	Augmented Reality, Mixed Realityの応用（紙によるソフトウェアコントローラ[15]，スマートホームプログラミング[16]，CGオブジェクトを手でつかめるAR[17]）
9	人間拡張（Human Augmentation）に関する研究（3,4本目の擬似的な手を使った共同作業[18]，身体のバランスを安定化させる擬似的な尾による[19]，指搭載型顕微鏡[20]）
10	Swarmロボットによるインタラクション（2次元の形状表現[21],[22]，3次元の形状表現[23]，自己変形[24]）
11	Playware（電子機器を搭載した光るゴムボール[25]，プロジェクションマッピングを使った車椅子ユーザ，非車椅子ユーザがともに遊べるアイスホッケー[26]）
12	Ambient interface（プログラマブルな窓[27]，目元だけ日陰にする車のサンバイザー[28]）

共有し，他の学習者のプログラムのソースコードを参考にして学べるようにした．

本授業では，学生自らに手を動かしてプログラミングさせるため，毎回の授業で表1に示す課題に取り組ませた．そして，個々の学生に実習の成果物として，Scratch3.0で開発したプログラムの共有URLを毎回Moodleに提出させた．毎回の授業の開始15分ほどは，筆者が2章で示した内容の講義，および表2に示す，プログラミングが活用された，世界で行われている先端的な研究の紹介を行った．その後，残り1時間15分間ほどを使って，学生に実習を行わせた．その際，教員が学生全員に対して一方的に教科書の内容の細部について説明することは行わず，個々の学生から分からないことがあると申告があった時に，その学生に対して理解の支援を行うようにした．

3. 2. 授業のアウトカム

最終課題の成果として，学生らが提出したプログラムの概要を，紙面の都合上種類ごとにまとめたものの一部を表3に示す．本授業を通して学生らはこれらのプログラを各自開発できるようになった．全体的にプログラムの内容のオリ

表3 プログラミング入門Iの最終課題の成果概要

No.	概要
1	迷路・アクションゲーム．スタートからゴールまで，キャラクタを操作させ，途中に出現する敵キャラ等を避けながら移動する．壁を超えて移動などはできないようにしてあり，敵キャラに衝突すると最初からやり直しになる．敵キャラの形状，動きは多様であり，衝突判定が正確に行われていた．
2	サッカーのPK．ゴールキーパーがランダムに移動する．プレイヤーはPKのキッカーとなり，ゴールキーパーにボールを取られないようにボールをシュートする．
3	麻雀の役判定プログラム
4	シューティングゲーム，捕獲ゲーム．授業で扱ったものと同じものは存在せず，出現するお化けを魔法使いが魔法で排除するものや，くまが餌を捕獲するもの等が提出されていた．FPSのようなシューティングゲームにし，シュートのターゲットカーソルの移動を滑らかにする等の細かい工夫も見られた．
5	ビデオモーションを使って，楽器を演奏するプログラム
6	マリオのようなアクションゲーム

ジナリティ，質は高く，デザインに凝ったものも多くみられた．最終課題は各自が作成したいプログラムを構想し，それを実現するものであったため，その実現にあたって教科書には載っていない問題解決が求められことも多くあり，各学生は様々な工夫を凝らして問題解決に取り組み，デバッグにも多くの時間を費やしていた．

プレゼンテーションの質も全体的に高く，何をするプログラムなのかについての概要，状態遷移図を使用したプログラムの内容について分かりやすく説明したものが多く見られた．

また，学生の感想として，「作りたいものを作るのは難しかったが，できた時にはかなりの達成感があった．」，「見た目，動きを思い通りにするのに多くの時間を要した」等が多くみられた．

4. プログラミング入門**II**の実践

プログラミング入門IIは，2020年9月から本学情報電子システム工学科の1年生向けの授業として後期に開講されている．本章では，2020年度後期のプログラミング入門IIの実践例を述べる．

4. 1. 授業の概要

Python3.8.6[31]およびそれに付随するIDLEを用いて，プログラミングの学習及びメタ学習の支援を行った．履修登録者数は44名，そのうち単位を取得できた者は38名であった．プログラミング入門IIを履修した44名のうち，43名はプログラミング入門Iも履修していた．本講義では，より詳細で幅広い機能を学習がプログラミングできるようにさせるため，Pythonを言語として採用した．また本講義は，2年次以降に開講されているC，Java，Pythonのプログラミングの授業へと学習を発展させる前段階としても位置づけられる．COVID-19の感染予防対策のため，第1〜15回まで全てZoomで授業を実施した．そのため，プログラミングの学習は，履修者が各自所有するPCに動作環境をインストールして行わせた．

本講義の授業内容を表4に示す．第1回から第10回までの授業では，[29]の文献を教科書として用い，筆者がその文献の内容を説明しながら，コーディングを示し，学習者に各自のPCで同様にコーディングしながら学習を行わせた．多くの学生にとって，ゲームは最も慣れ親しんでいるソフトウエアの一つであるため，簡単なゲーム開発を題材とした学習を扱うようにした．本講義では，Scratchより複雑なPythonを使用しているため，プログラミング入門Iのように教科書の内容の詳細を説明せずに自習できる学習は多くないと考えられた．そのたため，筆者が第10回までの毎回の授業において，教科書の内容の説明，注意点を述べるようにした．本講義における履修者は全員ノートPCを購入しており，Windows OSのものとMac OSの者がいたが，オンライン授業でほぼ全員問題なくプログラミングの実行環境を準備できた．一部，動作環境の準備に時間がかかった学生もいたが，そのような学生には最初Google colaboratory[32]を使って学習させ，動作環境のインストールが必要な第7回の授業までに動作環境の準備を終えるようにさせた．

表4 プログラミング入門IIの授業内容

回	内容
1	Pythonの概要，各自のPCにPython3.8.6をインストール．
2	Pythonプログラムの実行方法，Pythonの基本的な文法，日付に関するライブラリ等基本的なライブラリの使用方法．
3	変数と計算式，リスト
4	条件分岐と繰り返し
5	関数の定義と呼び出し，importの使い方
6	CUIのすごろくゲームの開発
7	GUIの基礎1(画像，テキスト表示とイベント・ドリブン型処理によるおみくじプログラムの開発)
8	GUIの基礎2(テキスト入力，チェックボックス，ボタンを使った診断ソフトの開発)
9	リアルタイム処理を使った迷路ゲームの開発
10	Pygameライブラリの使い方
11-13	課題制作(オリジナルのアプリケーションの開発)
14-15	課題発表(オリジナルのアプリケーションの発表)

プログラミング入門Iと同様に，第11回から13回までは，学生に個人またはグループでオリジナルのプログラム，およびその説明資料をプレゼンテーションソフトで作成させ，第14回，第15回の授業で他の学生に対してZoomで発表させた．このオリジナルのプログラムのテーマは自由とし，ゲームやアート，計算など何でも良いが，Pythonを使用することとした．

4. 2. 授業のアウトカム

最終課題の成果として，学生らが提出したプログラムの概要を，種類ごとにまとめたものの一部を表5に示す．本授業を通して学生らはこれ

表5 プログラミング入門IIの最終課題の成果概要

No.	概要
1	CUIによるオセロゲーム．コンピュータと対戦が可能.
2	OpenCV[33]を使って，しきい値を指定して画像を2値化するプログラム.
3	落ちてくる玉を弾き返し，ブロックを崩していくゲーム.
4	Ren'Py[34]を使ったノベルゲーム
5	GUI電卓．10進数の四則演算の他，10進数からn進数への変換，例外を使ったエラーの補足を実装したものがあった.
6	マリオのようなアクションゲーム
7	3目並べ，4目並べ．コンピュータと対戦が可能.
8	○Xゲーム．コンピュータと対戦が可能.
9	一時停止，再開機能がついたストップウォッチ
10	なぞなぞゲーム
11	トランプの「戦争」[35]を再現したゲーム．コンピュータと対戦が可能.
12	脱出ゲーム

らのプログラを各自開発できるようになった．プログラミング入門Iの最終課題と同様に，各学生は様々な工夫を凝らして問題解決に取り組み，デバッグにも多くの時間を費やしていた．プログラミング入門Iよりも，プログラミングの難易度が高かったが，プレゼンテーション，開発した成果物の質ともに，予想していたより高かった．

また，多くの学生はオリジナルのプログラム開発において，プログラミングの実装方法についてインターネットを通して様々なWebサイトを参考にしており，目的のソフトウェアを実現するために必要なことの調査能力，検索能力も養われたと思われる．またその際，「優れたプログラミング能力を持っている人が，世の中に沢山いることに気づいた」と感想を述べた学生もいた．能力の高い人を見つけ，その人がどのような知識，スキルを持っているかを知ることは，学習の高い目標にもなると考えられる．一方で，学生自身がそのような人より現時点で能力が低いため，学生自身が自身を失わないようにフォローすることも重要である．そのため，学生に対して誰もが最初から優れたプログラミングをできるわけではなく，学習の継続によってそのようなプログラミングが可能になることを補足した．

また，多くの学生は「時間の都合上今回の成果物までの機能にしたが，本当はここまで実現したかった」という意見も述べていた．今回の課題への取り組みによって，プログラミングの学習を継続するモチベーションが生まれたものと思われる．

5. 学生アンケート

本学で実施している授業評価アンケートとは別に，プログラミング入門IIの終了時に，プログラミング入門IとIIを通した授業について，学生に対してアンケート調査を行い，28名から回答を得た．学生から本音の回答を得るため，学生たちには完全に無記名でアンケートに答えてもらった．

「プログラミング入門IとIIを受講して，プログラミングを学習する方法を修得できましたか？」という問いに対する回答を図4に示す．「かなり修得できた」が12名，「少し修得できた」が10名，「どちらとも言えない」が5名，「あまり修得できなかった」が1名であった．アンケート回答者の約8割は，メタ学習によって学習する方法を修得できたと感じていた．

「プログラミング入門IとIIを受講して，プログラミングの学習意欲は向上したと思いますか？」に対する回答を図5に示す．「かなり向上した」が10名，「少し向上した」が14名，「かわらなかった」が4名であった．このことから，本講義は，本講義終了後の学生の学習の継続意欲の向上に一定の効果があったと言える．

「プログラミング入門IとIIを受講して，プログラミングの能力は向上しましたか？」に対する回答を，図6に示す．「かなり向上した」が9名，「少し向上した」が16名，「変わらなかった」が3名であった．このことから，本講義はメタ学習のみならず，プログラミングの学習そのものに対しても，一定の効果があったと言える．

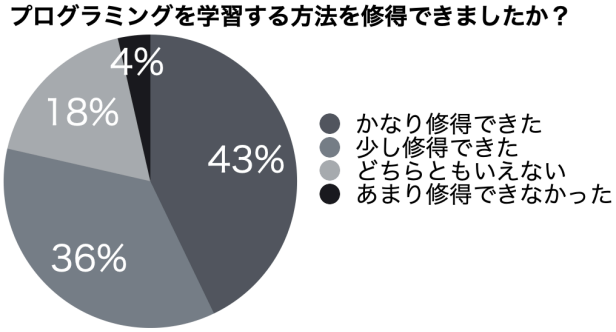

図4 学習方法の修得についてのアンケート結果

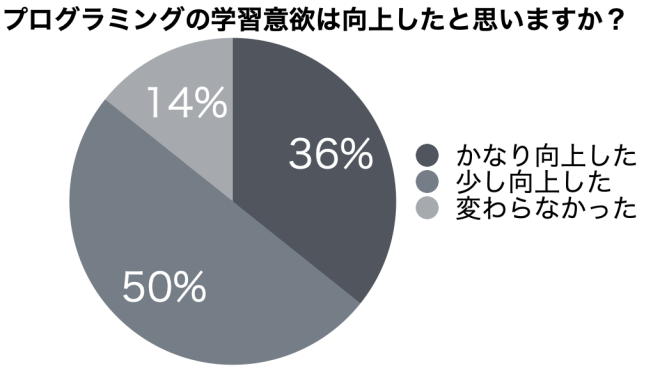

図5 学習意欲の向上についてのアンケート結果

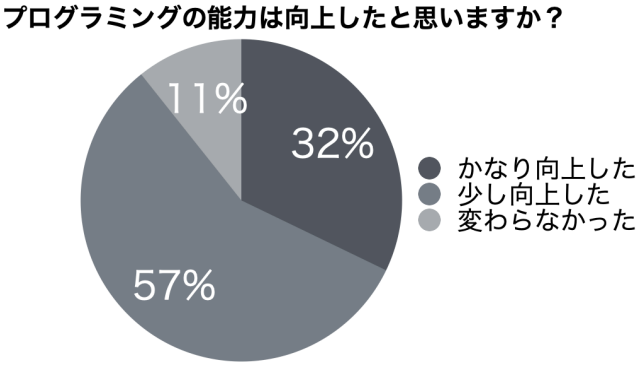

図6 能力の向上についてのアンケート結果

6. おわりに

本論文では，プログラミングのスキルを，学生が自ら継続してアップデートする能力を修得することを目的とした授業の一つの設計指針を提案し，その実践例を述べた．この設計指針，筆者による授業の実践内容の完成度は未だ不十分であり，教育効果の十分な測定とともに，今後継続して改善していく．本論文で述べたメタ学習の支援は，プログラミング以外の学問の学習支援にも適用し得る．

COVID-19の感染拡大に伴い，大学が遠隔授業を実践するようになった．この経験は，大学における学習，教育を見直す良い契機ともなった．世界トップクラスの大学や様々な企業が，インターネットを通して誰でも無料あるいは低価格で閲覧し，学ぶことができる幅広い分野，レベルの学習コンテンツを提供している．これらのコンテンツには，非常に質が高く，図や動画を活用して分かりやすいものが多く存在している．これらを使えば，大学に通わずとも初級レベルから上級レベルまで非常に質の高い学習をすることも可能である．このような状況下において，大学で学習，教育する上で重要なことは何であろうか．

授業で使用可能な質の高い外部の学習コンテンツは，積極的に授業で活用すべきである．それにより，教員による授業の資料作成，説明の負担を減らすことができる．その分，教員はより重要なことに力を注ぐことができる．筆者は，教員は授業においては個々の学生のメンタリング，授業外では研究に力を注ぐことが重要であると考える．

優れた学習コンテンツが存在していたとしても，それを学生が正しく，完全に理解できるとは限らない．そのため，教員には学生が理解できていること，できていないことを把握して，学習コンテンツや関連する内容について学生が理解できるように補足することが求められる．また，学生に学習のモチベーションがなければ，優れた学習コンテンツは活用できない．学生の学習モチベーションが高まるように，教員は学生に働きかけることが望まれる．このような学生のメンタリングにおいて，本論文で述べたメタ学習を支援する授業の設計指針が役立てられると考えている．

また，教員が研究によって出した成果は，学生にとって世界中で唯一の学習コンテンツである．実際に，査読付きの論文誌や国際会議で発表される研究成果には，査読によって一定の新規性，信頼性が保証されるため，世界中の他のどの学習コンテンツを見てもその研究成果については述べられていない．従って教員が研究に力を注ぎ成果をあげることで，大学独自の学習コンテンツが形成され，学生が大学に通って学ぶ意味が生じる．周知の事実であるかもしれないが，教員の研究成果を授業内容の主体とし，世界中の良質な学習コンテンツを併用して授業を実践することが，やはり理想的な大学の授業のあり方ではないだろうか．

参考文献

[1] Joi Ito, “The problem with tech people who want to solve problems”, https://www.vox.com/recode/2019/6/26/18758776/joi-ito-mit-media-lab-resisting-reduction-exorcist-kara-swisher-recode-decode-podcast-interview, 2021年6月7日参照.
[2] 杉浦学，”Scratchではじめよう！プログラミング入門3.0版”，日経BP，2019年11月.
[3] 株式会社Nileworks, “農業用ドローンNile-T20”, https://www.nileworks.co.jp/, (2021年6月8日参照).
[4] inaho株式会社，”AIを使った自動野菜収穫ロボットinaho", https://inaho.co/, (2021年6月8日参照).
[5] Toshiyasu OGATA, Hiroshi ABE, Kazuhiro SAMURA, Omi HAMADA, Masani NONAKA, Mitsutoshi IWAASA, Toshio HIGASHI, Hiroyuki FUKUDA, Etsuji SHIOTA, Yoshio TSUBOI, Tooru

INOUE, Hybrid Assistive Limb (HAL) Rehabilitation in Patients with Acute Hemorrhagic Stroke, Neurologia medico-chirurgica, 2015, Vol. 55, No. 12, p. 901-906.
[6] Jackie (Junrui) Yang, Christian Holz, Eyal Ofek, and Andrew D. Wilson. 2019. DreamWalker: Substituting Real-World Walking Experiences with a Virtual Reality. In Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology (UIST '19). Association for Computing Machinery, New York, NY, USA, 1093–1107. DOI:https://doi.org/10.1145/3332165.3347875
[7] Lift Labs of Verily Life Sciences LLC, "Lifeware Steady", https://www.liftware.com/steady/, (2021年6月8日参照).
[8] Lift Labs of Verily Life Sciences LLC, "Lifeware Level", https://www.liftware.com/level/, (2021年6月8日参照).
[9] Seita Kayukawa, Keita Higuchi, João Guerreiro, Shigeo Morishima, Yoichi Sato, Kris Kitani, and Chieko Asakawa. 2019. BBeep: A Sonic Collision Avoidance System for Blind Travellers and Nearby Pedestrians. Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems. Association for Computing Machinery, New York, NY, USA, Paper 52, 1–12. DOI:https://doi.org/10.1145/3290605.3300282
[10] S. S. Fels and G. E. Hinton, "Glove-talk II - a neural-network interface which maps gestures to parallel formant speech synthesizer controls," in IEEE Transactions on Neural Networks, vol. 8, no. 5, pp. 977-984, Sept. 1997, doi: 10.1109/72.623199.
[11] Naoki Kimura, Michinari Kono, and Jun Rekimoto. 2019. SottoVoce: An Ultrasound Imaging-Based Silent Speech Interaction Using Deep Neural Networks. Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems. Association for Computing Machinery, New York, NY, USA, Paper 146, 1–11. DOI:https://doi.org/10.1145/3290605.3300376
[12] 暦本 純一, 西村 悠, "Derma: 皮膚運動計測によるサイレントスピーチインタラクション", 情報処理学会 インタラクション2020論文集", pp.11-20, 2020年3月.
[13] Anumanchipalli, G.K., Chartier, J. & Chang, E.F. Speech synthesis from neural decoding of spoken sentences. Nature 568, 493–498 (2019). https://doi.org/10.1038/s41586-019-1119-1
[14] Ying-Ju Lin, Parinya Punpongsanon, Xin Wen, Daisuke Iwai, Kosuke Sato, Marianna Obrist, and Stefanie Mueller. 2020. FoodFab: Creating Food Perception Illusions using Food 3D Printing. In Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI '20). Association for Computing Machinery, New York, NY, USA, 1–13. DOI:https://doi.org/10.1145/3313831.3376421
[15] Vincent Becker, Sandro Kalbermatter, Simon Mayer, and Gábor Sörös. 2019. Tailored Controls: Creating Personalized Tangible User Interfaces from Paper. In Proceedings of the 2019 ACM International Conference on Interactive Surfaces and Spaces (ISS '19). Association for Computing Machinery, New York, NY, USA, 289–301. DOI:https://doi.org/10.1145/3343055.3359700
[16] R Suzuki, K Masai, M Sugimoto, ReallifeEngine: A Mixed Reality-Based Visual Programming System for SmartHomes. ICAT-EGVE, 105-112, 2019.
[17] Jing Qian, Jiaju Ma, Xiangyu Li, Benjamin Attal, Haoming Lai, James Tompkin, John F. Hughes, and Jeff Huang. 2019. Portal-ble: Intuitive Free-hand Manipulation in Unbounded Smartphone-based Augmented Reality. In Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology (UIST '19). Association for Computing Machinery, New York, NY, USA, 133–145. DOI:https://doi.org/10.1145/3332165.3347904
[18] MHD Yamen Saraiji, Tomoya Sasaki, Reo Matsumura, Kouta Minamizawa, and Masahiko Inami. 2018. Fusion: full body surrogacy for collaborative communication. In ACM SIGGRAPH 2018 Emerging Technologies (SIGGRAPH '18). Association for Computing Machinery, New York, NY, USA, Article 7, 1–2. DOI:https://doi.org/10.1145/3214907.3214912
[19] Junichi Nabeshima, MHD Yamen Saraiji, and Kouta Minamizawa. 2019. Arque: artificial biomimicry-inspired tail for extending innate body functions. In ACM SIGGRAPH 2019 Posters (SIGGRAPH '19). Association for Computing Machinery, New York, NY, USA, Article 52, 1–2. DOI:https://doi.org/10.1145/3306214.3338573
[20] Noriyasu Obushi, Sohei Wakisaka, Shunichi Kasahara, Atsushi Hiyama, and Masahiko Inami. 2019. MagniFinger: magnified perception by a fingertip probe microscope. In ACM SIGGRAPH 2019 Emerging Technologies (SIGGRAPH '19). Association for Computing Machinery, New York, NY, USA, Article 17, 1–2. DOI:https://doi.org/10.1145/3305367.3327994
[21] Michael Rubenstein*, Alejandro Cornejo, Radhika Nagpal, Programmable self-assembly in a thousand-robot swarm, Science 15 Aug 2014, Vol.

345, Issue 6198, pp. 795-799, DOI: 10.1126/science.1254295
[22] Mathieu Le Goc, Lawrence H. Kim, Ali Parsaei, Jean-Daniel Fekete, Pierre Dragicevic, and Sean Follmer. 2016. Zooids: Building Blocks for Swarm User Interfaces. In Proceedings of the 29th Annual Symposium on User Interface Software and Technology (UIST '16). Association for Computing Machinery, New York, NY, USA, 97–109. DOI:https://doi.org/10.1145/2984511.2984547
[23] MIT CSAIL, Self-transforming robot blocks jump, spin, flip, and identify each other, https://news.mit.edu/2019/self-transforming-robot-blocks-jump-spin-flip-identify-each-other-1030, (2021年6月8日参照)
[24] Ryo Suzuki, Clement Zheng, Yasuaki Kakehi, Tom Yeh, Ellen Yi-Luen Do, Mark D. Gross, and Daniel Leithinger. 2019. ShapeBots: Shape-changing Swarm Robots. In Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology (UIST '19). Association for Computing Machinery, New York, NY, USA, 493–505. DOI:https://doi.org/10.1145/3332165.3347911
[25] 出田修，佐藤俊樹，間宮暖子，芝崎郁，中村潤，児玉幸子，小池英樹，“跳ね星：電子機器を組み込んだデジタルスポーツ用ゴムボールの開発”, WISS2008, 2008年.
[26] Roland Graf, Sun Young Park, Emma Shpiz, and Hun Seok Kim. 2019. IGYM: A Wheelchair-Accessible Interactive Floor Projection System for Co-located Physical Play. In Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems (CHI EA '19). Association for Computing Machinery, New York, NY, USA, Paper LBW1615, 1–6. DOI:https://doi.org/10.1145/3290607.3312792
[27] Jun Rekimoto, Squama: Modular Visibility Control of Walls and Windows for Programmable Physical Architectures, Advanced Visual Interfaces (AVI 2012), 2012.
[28] Bosch Mobility Solutions USA, Virtual Visor, https://www.bosch-mobility-solutions.us/us/highlights/connected-mobility/virtual-visor/, (2021年6月8日参照).
[29] 廣瀬 豪, Pythonでつくる ゲーム開発 入門講座，ソーテック社，2019年7月.
[30] Scratch 3.0, https://scratch.mit.edu/, (2021年6月10日参照).
[31] Python 3.8.6, https://www.python.org/downloads/,(2021年6月10日参照).
[32] Google colaboratory, https://colab.research.google.com/, (2021年6月10日参照).
[33] OpenCV, https://opencv.org/, (2021年6月10日参照).
[34] Ren'Py, https://www.renpy.org/,(2021年6月10日参照).
[35] 戦争(トランプゲーム), https://ja.wikipedia.org/wiki/戦争_(トランプゲーム), (2021年6月10日参照).