

文系学生を対象にしたプログラミング体験授業の実践報告

小川真里江^{*1}

Email: ogawa@mejiro.ac.jp

^{*1}: 目白大学情報教育センター

◎Key Words

一般教養教育, 3D 仮想空間, ドローン

1. はじめに

政府が目指すこれからの社会 Society5.0⁽¹⁾ では、新たな社会を支える人材として「自ら課題を発見し解決手法を模索する、探究的な活動を通じて身につく能力・資質が重要である」ことを挙げている。近年、人工知能 (AI) や IoT などの情報技術が発展する社会的背景から STEAM (Science, Technology, Engineering, Art, and Mathematics) 教育⁽²⁾ が重要視されるなか、大学においても文理の枠にとらわれることなく分野横断的な学びが求められている。しかしその一方で、文系学生にとっては、数学や化学などの分野に苦手意識や挫折した学習者も多いため、数式・計算・論理の抽象性などを前面に出すことなく、体験をとおして興味や関心を引き出すなどの工夫が欠かせない。

このような学びの導入として、実際に手を動かしながら取り組めるプログラミングは有効な手段の一つである。プログラミングを学ぶことは単に技術を習得するだけでなく、コンピュータへ命令するための論理的思考力や、プログラミングをとおして『わかる』ということが何なのかを感じとる力を育むことができると考えられる。このことを踏まえ、本研究では学習者が『わかる』という感覚を実体験によってより強く得られるように、仮想空間や現実空間を用いたプログラミング言語の学びを提案する。

アメリカの国立訓練研究所が発表した「ラーニングピラミッド⁽³⁾」では、学習形態の違いによる学習定着率は「講義」が 5% に対して「自ら体験する」が 75% に達するとし、体験的な学びの重要性を示している。また、日本の義務教育においてもプログラミング学習がはじまり、その学びでは「実際の体験を通じて、プログラミングに留まらず現代技術の背後にある思考力を涵養する教育⁽⁴⁾」が重要視されている。

本報告は、実体験を含むプログラミング学習を教材とした授業の実践報告である。

2. 学習ツール

プログラミングでは、一般的にエディタ上でソースコードの入力やデバッグ、および、動作確認は主に文字ベースでおこなわれる。このため、はじめてプログラミングを学ぶ者にとっては、プログラムコードの意味をイメージ化することが難しい側面がある。そこで、本学習ではプログラミングの過程のうち特に動作確認に焦点をあて、単なる文字情報による確認に留まらず、3D 仮想空間と現実空間でプログラムの動作を確認する方法を採用する。3D 仮想空間では、オブジェクト化したドローンをプログラ

ムで動かす体験をする。現実空間では、実物のドローンをプログラムの指示に従って動かせることを体験する。これらの空間の利用は、道具への興味を持たせると共に、プログラムコードを視覚的かつ直感的に『わかった』という感覚を獲得することを支援する。

3D 仮想空間にはフリーソフトの OpenSimulator を用いる。この空間は、1 辺が 255px の平野な土地、および、海と空で構築されている。また、この空間上では、自分の分身となるアバターが指示に従ってオブジェクト作りや動作確認をおこなう。さらに、球体や立方体などの 3D オブジェクトを変形させ組み合わせることで、ユーザがイメージした複雑なオブジェクトも作成できる (図 1)。

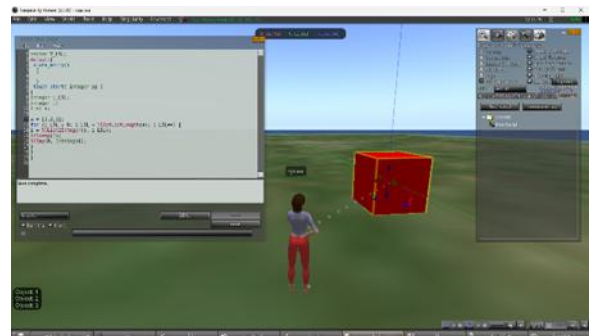


図 1 3D 仮想空間 OpenSimulator

個々のオブジェクトにはプログラムを埋め込むことで、直感的に指示内容の動作確認が可能となる。OpenSimulator を使ったプログラミングは、本来スクリプト言語に位置付けられるリンデンスクリプトが、実空間で使うドローンのプログラミングには Python が必要になる。本学習ではプログラミングが初めての学習者や苦手意識をもった学習者を想定しているため、複数の言語を一度に学ぶことに対する複雑さを軽減する必要がある。そのため、学習する言語は Python のみとして、3D オブジェクトのプログラミングは、Python で書かれたソースコードをリンデンスクリプトへ変換する翻訳機を使用する対策を講じた。この翻訳機は、オブジェクトに対して設定したいイベント (たとえば、オブジェクトに触れたときに動作する処理) を指定し、Python で書かれたソースコードを翻訳機を使ってリンデンスクリプトに変換される。変換されたソースコードは、ユーザが一切手を加える必要はなく、オブジェクトに貼り付けるだけで動作させることができるように工夫がなされている。

現実空間で使用するモノとして位置付けているドロー

ンには、教育向けに DJI と Ryze Tech が共同開発した tello EDU⁶⁾ (以下、tello) を用いる (図 2)。



図2 ドローンを飛行させている様子

このドローンは Python や Scratch を用いたプログラミングができる。また、手のひらサイズの小型で軽量 (約 87 g) であるため、屋内の比較的狭い場所でも飛ばすことができる利点がある。ドローンでのプログラミング学習に Python を使用した理由の一つには、通常ドローンを飛行させるプログラムには通信やコマンドによる複雑な命令が必要になるが、それらのコードをラッピングし一つの部品のように扱うことが可能で、学習者のレベルに合わせた工夫ができる。また、tello には機体の小型のカメラが付いているため、ラッピングを活用することで、学習者は簡単なコードで写真を撮ることも可能である。さらに、画像認識などの AI 技術とリンクする部品を作成し提供することで、より発展的な学びに広げる可能性も有している。

その他、Python は、① 初心者でも文法が理解しやすい、② インタープリター型言語であるため動作確認が容易、③ 開発するためのツールが無償で数多く提供されているなどが挙げられ、体験的な学びをサポートするのに適している。

3. Python を用いた学習

プログラミング学習においては、Microsoft が無償提供する Visual Studio Code⁶⁾ (以下、VSCoDe) をエディタとして活用する。VSCoDe は、Pythonをはじめ、JavaScript などのさまざまな言語に対応しており、拡張機能も利用できる。また、初心者にも使いやすい仕様で軽量かつ高速なため、一般的な PC にも容易にインストールできる特徴がある。

初心者がプログラミング学習で挫折することなく学びを継続するためには、プログラムを作るための道具を簡単に使えるようにすることだけでなく、学習を進めるなかで成功体験を積み重ねていくことも重要である。このために、作成したプログラムをすばやく 3D 仮想空間やドローンで動作確認をおこなうことができるような工夫も取り入れている。

3D 仮想空間やドローンを用いた体験的なプログラミング学習であっても、Python の文法や構文を使った基礎的な書き方を理解しておく必要がある。そのためには、① 基本構文である変数やデータの型、演算子の使い方、② 条件分岐である if、ループの for の基本的な使い方、③ オリジナルのプログラムを作成する、というプロセスで学

びを進めていく。ただし、作成したプログラムの動作確認では、単に手順どおり入力して結果の文字が表示されるだけでなく可能な限りシミュレーターを利用する。たとえば、言語学習の初歩的な段階で「Hello World!」を表示させる事例が見られるが、これをアバターがオブジェクトに触ると表示させられるようなことを 3D 仮想空間でおこなう。単純なことではあるが、より『わかる』プログラミング学習につながるのではないであろうか。さらに、学びを進めていくにあたっては、3D 仮想空間から実空間へ移行するときの連続性を配慮している。たとえば「オブジェクトが時計回りに 1m ずつ周回する」という移動を取り入れたテーマを与え、はじめのステップでは 3D 仮想空間でシミュレーション、次のステップでは、同様の動きを現実空間上の目の前にあるドローンで飛行を確認する。この体験は、プログラミング学習における『わかる』感覚が、ドローンの飛行時に感じる音や風力などの臨場感と結びつくことで、記憶に残る生きた学びとなり、実感を伴う経験値と知識が結びつくことで学習が深まると考えている。

4. 授業の概要

この試みは、文系の 2 年生から 4 年生の学習者を対象とした一般教養の中の情報選択科目の一つとしておこなった。対象はプログラミングを学ぶことが初めて、あるいは、過去に挫折や苦手意識を持つ経験のある学習者としている。

進め方は、大きく二部構成となっており、前半に 3D 仮想空間を用いた基礎的なプログラミングの練習、後半にドローンを用いた実習をおこなった。

授業の到達目標は、プログラマーなど専門職を育成することを目的としているのではなく、経験値の少ない学習者でも自作したプログラムを使って実際にモノを動かす体験をととして『わかる』や『できる』という自己効力感を高めることにある。

具体的には、前半に用いる 3D 仮想空間では、一人で一つの空間を使って自作したプログラムでモノに指示を与えて動かす一連の流れをシミュレーションすることによって、視覚的にプログラムの動作を確認する。初心者にとっては、記述したプログラムコードを視覚的な動きで確認できることで、単に文字だけを追う作業による単調さからくる飽きが軽減され、取り組みやすくなることに期待が寄せられる。一方、後半に用いるドローンの実習では、3D 仮想空間とは違い目の前でドローンを飛ばすことで、プログラムの動作をよりリアルに感じ取ることができる。前半と後半を通じたこの直感的な学びのプロセスは、学びのハードルを下げることや躓きから来る意欲の低下を防ぐことへの効果が期待される。

授業手順は、はじめに学習者自身のノート PC の環境を整え、次に前半に用いる 3D 仮想空間の操作練習をしながら変数・構文・リスト・関数などの基礎的なプログラミングの知識を習得する。次に、後半で用いるドローンの操作練習をしながら前半で学んだ内容を使い、ドローン飛行の課題に取り組む流れになっている。また、最後には自由課題として、自ら考えたドローンの飛行ルートに基づきプログラミングをおこない、実際に動作確認までをおこなう。なお、ドローンを使つての動作確認にあたっては、

ドローンを使いたい学習者のノート PC と一対一の関係で接続する必要があるが、予算の関係などから学習者全員分を準備することはできないため、課題プログラムが作り終えたものから用意された数台のドローンを順番に使っておこなっている。

5. アンケート分析結果

毎回の授業終了時には、課題の提出と共に授業内容の難易度、興味関心などを択一式、コメントを自由記述式で答えてもらっている。図3は2024年後期に実施された授業でのアンケートから、プログラミングの難易度および興味に着目して、難易度と興味を軸にその頻度を表したものである。回答は難易度、興味共に5段階評価としている。縦軸の頻度は、全回答件数に対する割合を表している。全体的には難易度に関わらず興味を持って学んでいる様子が伺われる。

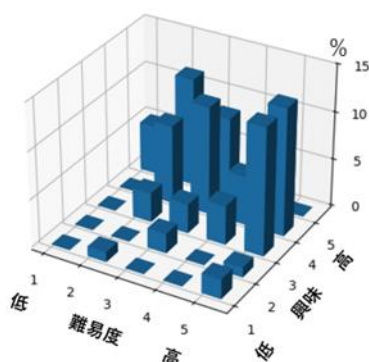


図3 難易度と興味

特徴的なことは、難易度が高いと感じながらも興味を持って取り組んでいるケースが多く認められることである。一般的には難しいと感じると興味を失うケースが多いが、今回の試みでは興味を失うことなく学習を継続できていると推察される。

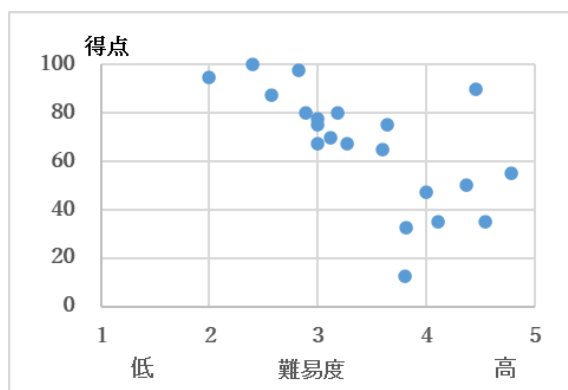


図4 難易度と小テストの得点

図4は実機のドローンを飛行させる直前の授業におこなったpythonのプログラミングに関する小テストとプログラミングに対する難易度の関係を表している。横軸は毎回の授業で尋ねた難易度についての学習者一人ひとりの平均値である。この図では一般的な傾向が認められ、難

しく感じている学習者ほど、その得点は低い傾向にある。

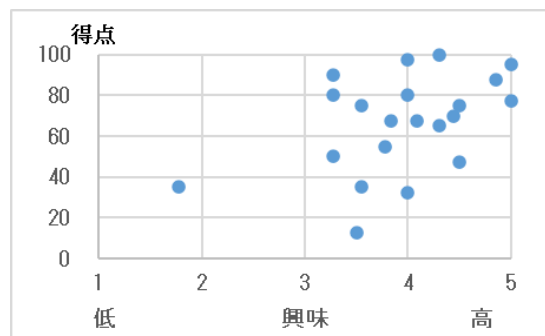


図5 興味と小テストの得点

図5は図4と同様に縦軸に小テストの得点、横軸に興味と定めその関係を表している。難易度の関係とは異なり、興味の得点の間には強い相関は認められない。小テストの点数が低い学習者も興味を持って継続して取り組んでいることが伺われる。課題解決のために自らプログラムコードを生成する力が養われていなくとも、興味を失うことなく取り組めていることは、幅広く学習するという観点に立てば、単に単位獲得のためだけに授業に参加する態度と比較すると、意味があるように思える。

今後、課題の達成状況や自由記述と対比させることで、一人ひとりの学びの特性を把握したい。また、2025年前期におこなった授業についても分析を進めたい。

参考文献

- (1) 内閣府：“Society 5.0とは”，https://www8.cao.go.jp/cstp/society5_0/index.html, (2025年6月29日参照)。
- (2) 文部科学省：“STEAM教育等の各教科等横断的な学習の推進”，https://www.mext.go.jp/a_menu/shotou/news/mext_01592.html, (2025年6月29日参照)。
- (3) 東京都教育委員会，“平成25年度教育研究院研究報告書（高等学校・情報）”，<https://www.kyoiku-kensyu.metro.tokyo.lg.jp/09seika/reports/files/kenkyuin/kou/joho/h25kou-joho.pdf>, 東京都教育庁指導部指導企画課, (2025年6月30日参照)。
- (4) 本田敏明，赤間文香：“プログラミング教育について-情報リテラシーの涵養かICT人材の育成か-”，茨城大学教育学部紀要（教育科学），66号，pp.367-386（2017）。
- (5) tello EDU：<https://www.ryzerobotics.com/jp/tello-edu>, (2025年6月20日参照)。
- (6) Visual Studio Code：<https://azure.microsoft.com/ja-jp/products/visual-studio-code>, (2025年6月20日参照)。