

Puppy による 60 分プログラミング体験コースの実践

秋信 有花^{1,a)} 倉光 君郎^{1,b)}

概要: 近年では、中高生が気軽にプログラミングを体験する機会が増えている。手軽に参加可能なプログラミング体験コースは、中高生がプログラミングに興味を持つきっかけとなり得る貴重な機会である。しかし、このような体験コースでは時間の制約が多く、プログラミングに触れられる時間は多くとも 60 分程度と限られている。このような時間的な制限の中、プログラミングの楽しさや継続的な学習意欲を高める体験を提供するのは大きなチャレンジといえる。

本稿では、Puppy を用いた 60 分のプログラミング体験コースの実践を報告する。アンケートにより、プログラミングを勉強してみたいという学生は、19.6%から 44.2%に増加し、継続的な学習意欲を高める体験コースの実践ができた。

The 60 Minutes Programming Introduction with Puppy

YUKA AKINOBU^{1,a)} KIMIO KURAMITSU^{1,b)}

Abstract: In recent years, there have been an increasing number of opportunities for junior and senior high school students to experience programming in a casual manner. Easy-to-attend programming courses are a valuable opportunity to get junior and senior high school students interested in programming. However, these hands-on courses have many time constraints, and the amount of time they can be used for programming is limited to about 60 minutes at the most. It is a great challenge to provide an experience that raises the enjoyment of programming and motivates students to continue learning under such time constraints.

In this paper, we report on the practice of a 60-minute programming experience course using Puppy. According to the questionnaire, the number of students who wanted to study programming increased from 19.6% to 44.2%, and we were able to implement a hands-on course that motivated them to continue learning.

1. はじめに

近年、ネットワーク技術や人工知能の発展により、我々の生活と情報技術は身近なものとなってきている。また、COVID-19 流行もあり、今後、社会生活のオンライン化、デジタル化がますます進んでいくと予想される。そのような社会的な背景のもと、プログラミングに関心を持ち、プログラミングを体験してみたいという中学生・高校生の数も着実に増えている。

現在、情報系学科主催のオープンキャンパス、高大連携

の大学体験講義などの様々な場面で、中学生・高校生に対してプログラミング体験の機会が提供されている。このような体験の機会は、中学生・高校生が手軽に参加しやすいという利点があるが、プログラミングに触れられる時間が多くとも 60 分程度と限られている。このような時間的な制限の中、プログラミングの楽しさや継続的な学習意欲を高める体験を提供するのは大きなチャレンジといえる。

本稿は、Puppy の初期プロトタイプシステムを用いた 60 分プログラミング体験コースの実践を報告する。

まず、Puppy は、中学生・高校生向けのプログラミング教育の機会を広げるため、新たに開発を進めている Web プログラミング環境である。Puppy の特徴は、物理シミュレーションと統合されたプログラム可視化にある。物理シミュレーションによって実世界の物体の動きを描画することで、抽象的なプログラムを直感的に視覚化できるように

¹ 日本女子大学大学院理学研究科数理・物性構造科学専攻
Graduate School of Science Division of Mathematical and
Physical Sciences, Japan Women's University, 2-8-1 Mejiro-
dai, Bunkyo-ku, Tokyo 112-8681, Japan

a) m1616003ay@ug.jwu.ac.jp

b) kuramitsuk@fc.jwu.ac.jp

なっている。また、独自の Python ベースの簡易プログラミング言語も設計し、テキストベースのコーディングを学び、より本格的なプログラミングにステップアップしやすくなっている。

本研究では、Puppy プロタイプシステムを用いて、高校生及び大学1年生を対象に60分プログラミング体験を実施した。特に、大学1年生は家政学部児童学科が対象であったが、プログラミングを勉強してみたい学生は、19.6%から44.2%に増加し、逆に、「勉強したいけど、私には難しそう」と感じていた学生は、60.9%から39.5%に減少した。Puppy は、プログラミングの楽しさを伝え、継続的な学習意欲を高める体験コースの実践ができることが確認された。

本稿の残りは、以下の通り構成される。2節では、Puppy のコンセプトを紹介する。3節では、Puppy を紹介する。4節では、Puppy によるプログラミング体験コースの報告を行う。5節では、関連研究を簡単に述べる。6節では、本論文を総括し、展望をまとめる。

2. Puppy コンセプト

Puppy は、中学生・高校生向けの新しいプログラミング入門環境として企画され、2019年度から開発が行われている。まず、簡単にPuppyの開発コンセプトを紹介していきたい。

2.1 物理シミュレーションを用いた可視化

プログラムは、目に見える実世界の現象に比べると、抽象的な概念であり、子どもたちに関心を持たせることが難しい。そのような中、ロボットを用いた制御プログラミングは、小学生から高校生まで定番のプログラミング入門教材となっている。ロボット制御を用いると、抽象的なプログラム処理がロボットの実際の動作として直感的に理解できる。実際に、ロボットの動作をみることで達成感も大きい。しかし、ロボットを用意するのはスペースや設備の面でコストも大きく、大学等のプログラミング体験コースでロボットを用いても、在宅で続けにくい。

Puppy のアイデアは、物理シミュレーションを用いることで、仮想的な実世界をパソコン(タブレット)内に実現し、プログラムの動作と融合して可視化することである。設備やスペースを気にすることなくプログラミング体験を継続的な学習につなげていけるようにする。

2.2 コーディング支援

近年の入門プログラミング言語は、キーボードの操作に不慣れた初学者に対して、Scratchのようなブロックベースのプログラミング環境を提供することが増えている。Puppy では、「テキストはプログラミングの基礎」と考え、テキストベースのコーディングを採用している。テキスト

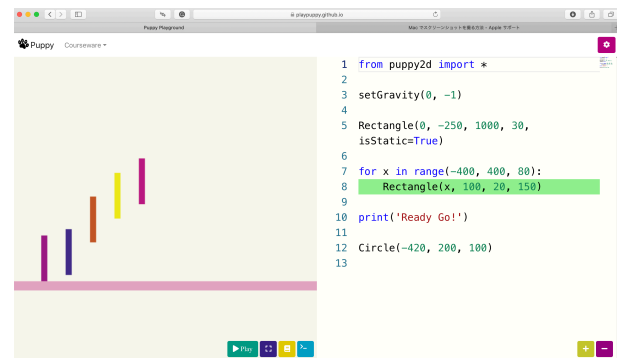


図1 Puppy メイン画面

の入力を採用する代わりに、初学者向けの新しいコーディング支援を検討していく計画である。現在、次のような支援を検討し、試作や実験を行っている。

- エディタによる支援
- 言語や構文の工夫による支援
- 自然言語処理技法を用いた支援 [1]

3. Puppy の紹介

本節では、4節で述べる体験コースで用いたバージョンをベースにしてPuppyの機能を紹介する。なお、Puppyのプロタイプシステムは、以下のリンクから試すことができる。

<https://playpuppy.github.io/>

なお、実装に興味を持たれた読者は、[2]などを参考にしてください。

3.1 Puppy メイン画面

Puppy は、Webブラウザ上でユーザインタフェースを提供する。図1は、Puppyメイン画面のスクリーンショットである。ユーザは、右側のエディタ部にコードを書き、同じ画面の左側でプログラムの実行を可視化することができる。エディタ上の緑マーカーは、実行中の行を示している。

Puppyの特徴は、物理シミュレーションと統合されたプログラムの可視化にある。図2は、ドミノ倒しプログラムの実行を一定間隔でキャプチャし並べて表示したものである。物体は、重力にしたがい落下運動をし、ボールと衝突すると、ドミノを順番に倒していく様子がわかる。このように物理シミュレーションを導入することで、少ないコード量で自然な動きが再現でき、さらに可視化された物体運動からプログラムを直感的に理解できるようになっている。

3.2 Puppy プログラミング

もう少し、Puppyプログラミングをみていこう。

Puppyでは、本格的なPythonプログラミングへステップアップしやすいように、Pythonの簡易版としてPuppy言語を独自に設計している。Puppy言語は、中学校や高等

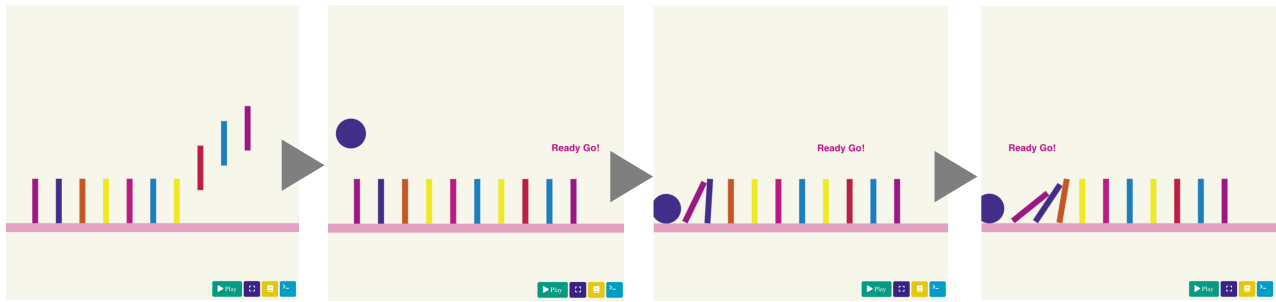


図 2 物理シミュレーションと統合されたプログラムの可視化

学校の限られた授業時間内でもプログラミングが学べるように言語機能が制限されている。繰り返しやリストなどの機能は備えており、情報オリンピック等のアルゴリズム問題を解くことができるよう配慮している。

物理シミュレーションによる可視化は、`puppy2d` モジュールをインポートすることで実現できる。モジュールとして提供することで、Python 文法を拡張することなく統合されている。次のように、`Circle` オブジェクトを作成すると、物体として画面に表示される。

```
Circle(0, 100, 100, fillStyle='red', restitution=1.0)
```

なお、`restitution=1.0` は、跳ね返り係数が 1.0 であることを与えている。物理属性が与えられることで、物体は表示されるだけでなく、物理シミュレーションによって動きが実現される。また、物体属性は、座標を含め、メソッドやプロパティを通して操作できる。このような操作と可視化を組み合わせると、プログラミングを学ぶことができる。

次は、図 2 で示したドミノ倒しプログラムの全ソースコードである。物体オブジェクトや物理属性を組み合わせることで、少ないコード量ながらも自然な動きが再現できる。

```
from puppy2d import *

setGravity(0, -1)

Rectangle(0, -250, 1000, 30, isStatic=True)

for i in range(-400, 400, 80):
    Rectangle(i, 100, 20, 150)

Circle(-420, 200, 100)
```

さらに、`Puppy` では、マウスイベントの他、キーボードイベント、衝突イベント、タイマーイベントを与えることができる。これらのイベントを組み合わせることで、簡単なゲームやアート作品を作成できるようになっている。

3.3 ユーザ支援機能

`Puppy` は、テキストベースのコーディングを採用している代わりに、初学者向けのコーディング機能を提供している。`Puppy` では、シンタックスハイライトや入力候補の表

示機能などの一般的な統合開発環境で提供されている機能だけではなく、初学者向けの独自の支援機能を提供している。図 3 は、全角文字の検出、また型エラーを検出した際のエディタである。`Puppy` の実行環境上では、全角文字のハイライトやコンパイルエラーをエラーログだけでなく色の变化でフィードバックしている。エラーをよりわかりやすく伝えることで、テキストベースのコーディングの困難さを減らしている。特に、半角と全角の区別ができていない初学者は多く、間違っ全角を使ってしまったことによるエラーは多い [3]。そのため、`Puppy` の実行環境上では、事前に全角の検出を行っている。

また、`Puppy` は、学習支援機能として `Puppy` コースウェアを提供している。`Puppy` コースウェアは、`Puppy` を使用して授業を行う際の教材として使用することができる。図 4 は、コースウェア機能を表示させた `Puppy` の画面である。画面の左側のノードブックで授業内容を確認しながら、右側のエディタ部でコードの記述ができるようになっている。現在の `Puppy` コースウェアには、我々が作成した教材例や `Puppy` のコード例が用意されている。今後は、ユーザ (教員) 自らコースウェアやノートブックを簡単に編集できるようにしていく予定である。さらに、`Puppy` は、Google アカウントによるログイン機能も実装されており、ログインしたユーザのコンパイルエラーはログとして記録されるようになっている。

4. 実践・ユーザ評価

我々は、高校生及び大学 1 年生を対象に、`Puppy` を使用した 60 分程度のプログラミング体験コースを行った。本節では、最新の実践例である全学共通情報リテラシーの授業内でおこなったプログラミング体験コースとそのアンケート結果について報告する。

4.1 概要

我々は、日本女子大学の全学共通情報リテラシー科目である基礎情報処理において、`Puppy` を用いたプログラミング体験コースを実施した。基礎情報処理では、インターネットや文書作成、表計算処理を身につけたのち、最終回の一コマを活用してプログラミングの体験を行った。実施

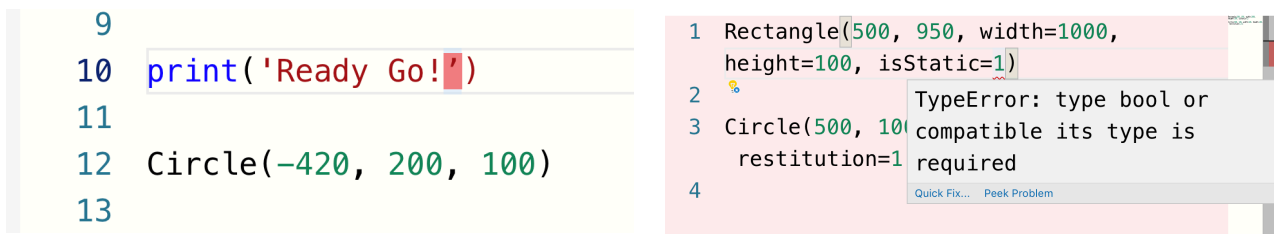


図 3 全角文字の検出 (左) と型エラー (右)



図 4 コースウェア機能

した授業の概要は以下の通りである。

- 基礎情報処理 (第 14 回)
- 実施日: 2020 年 1 月 22 日
- 授業時間: 13:20 - 15:00 (100 分間)
- 対象: 家政学部児童学科 1 年生
- 人数: 46 名
- 授業形態: ハンズオン形式 (1 人 1 台 PC を使用)

4.2 体験コースの設計

日本女子大学の授業は 1 コマ 100 分となっており、そのうち、60 分間を実際の体験コースとした。具体的には次のような流れで実施した (カッコ内は所要時間)。

- 講師の紹介 (5 分)
- "The Hour of Code is Here" *1 の視聴 (5 分)
- 授業前アンケート、環境準備 (15 分)
- プログラミング体験 (60 分)
- 授業後アンケート (10 分)
- クロージング (5 分)

プログラミング体験は、Puppy60 分コースとして用意した入門コースウェアにしたがって進めた。内容は、まず数値や文字列の基本的な操作を学び、Puppy の特徴である物体記述を試してみる。そのあと、繰り返しの構文を学んで高度なプログラミングを体験できるようになっている。最後には、理解度を確認するため、課題「ドミノ倒し」が用意されている。図 5 は、Puppy60 分コース*2 のユニットを示している。

Puppy コースの課題「ドミノ倒し」は、スクラッチから

*1 www.youtube.com/watch?v=FC5FbmsH4fw

*2 playpuppy.github.io/playground/?course=course/PuppyCourse#0

- (1) 「ようこそ」
 - Puppy の紹介と簡単な操作説明
- (2) 「Hello World」
 - print 文を使って好きな文字列を表示してみよう
- (3) 「数と計算」
 - 演算記法を使って自由に四則演算を試してみよう
- (4) 「円や四角を表示させてみよう」
 - 自由に物体オブジェクトを配置してみよう
- (5) 「弾むボール」
 - 重力や跳ね返り係数などのプロパティを設定し、物体に好きな動きを追加してみよう
- (6) 「繰り返し」
 - for 文を使って物体オブジェクトをたくさん表示してみよう
- (7) 課題「ドミノ倒し」
 - ドミノ倒しプログラムを作成してみよう

図 5 Puppy 60 分コース

プログラムを書く応用問題となっている。まず、受講生にはドミノ倒しプログラム (図 2) の動きを動画で見てもらい、同様の動きを自力で再現してもらった。ドミノ倒しプログラムを完成させるためには、以下の 3 つのポイントが理解できている必要がある。

- プログラムは上から順に実行される
- 繰り返しの構文をうまく用いることで簡潔にプログラムを書くことができる
- プロパティを追加することで、動きを変えることができる

4.3 授業アンケート

授業アンケートは、体験コースの前後 2 回行った。体験コースを通して、受講生のプログラミングの印象がどのように変化したのかを知るため、体験前と体験後に同様の質問も含まれている。以下は、アンケートの概要である。

授業前アンケート

- A: 普段のパソコンの利用頻度
- B: タイピングの習熟度
- C: これまでのプログラミング経験 (自由記述)
- D: (体験前の) プログラミングの印象

授業後アンケート

- E: 課題の達成度 (満足度)
- F: プログラミングの理解
- G: (体験後の) プログラミングの印象
- H: 授業の感想 (自由記述)

表 1 プログラミングを体験したことのある学生

	学生 A	学生 B	学生 C	学生 D	学生 E
体験した言語	-	-	-	-	Python
体験時期 や機会	高校	中学	高校	高校	父が SE
体験時間	1 時間	3 時間	50 分	2 時間	15 時間

表 2 プログラミングに対する印象の変化 (自由記述)

1	思ったより簡単で楽しかった
2	もっと意味の分からない記号や暗号を使うものなのかなど 思っていたけど、考えていたよりも簡単にできて、 少しだけ身近に感じられることができた
3	プログラミングは専門的なものだと思っていたが、 初心者でも楽しむことの出来るものであることが分かった

具体的な質問項目は、アンケート結果と共に説明する。

4.4 授業前アンケート結果

授業前アンケートの有効回答数は 46 であった。

図 6 は、『A : 普段のパソコンの利用頻度』と『B : タイピングの習熟度』について図示したものである。受講生は、毎日 (24%)、1 週間に数回 (15%)、1 ヶ月に数回 (25%)、1 学期に数回 (36%) 程度使っていた。タイピングに関しては得意でないと感じている学生が多いことがわかった。

『C : これまでのプログラミング経験』は、5 名の受講生が経験ありと回答した。表 1 は、その経験についてまとめたものである。学生 E をのぞいてほとんど未経験といえた。

『D : 体験前のプログラミングの印象』は、「勉強したいけど、私には難しそう」と回答した受講生が 60.9% と最も多かった。「勉強してみたい」と意欲的な回答を示した受講生は 19.6% とどまった。また、受講生のうちの 30.4% は「勉強しないと時代に取り残されそう」と回答した。

授業前アンケートの結果を総合すると、受講生の特徴は以下にまとめられる。

- パソコンの利用状況は様々であり、タイピングについては苦手意識を持っている学生が多い
- プログラミング経験のない学生がほとんどである
- プログラミングに対して「難しそう」などネガティブな印象を持っている学生が多い

4.5 授業後アンケート結果

授業前アンケートの有効回答数は 43 であった。

図 7 は、『E : 課題の達成度について』の回答結果を図示したものである。受講生の 8 割以上が限られた時間内でドミノ倒しプログラムを記述できたことがわかる。

『F : プログラミングの理解度』は、以下の項目について選択式で回答を得た。

- (1) プログラミングを学ぶと、数式を書いて計算できる
- (2) プログラミングは、数式だけでなく、文字や画像などを扱うことができる
- (3) プログラムは、文法どおりに書かないと、正しくうごかない
- (4) プログラミングは、数学だけでなく英語力 (国語力) も重要である
- (5) プログラムは、上から順番に実行される
- (6) プログラミングは、繰り返し処理を簡単に実現できる

表 3 今回の授業を経験してさらに学びたいこと (自由記述)

1	迷路みたいなものを作ってみたいと思った
2	ゲームの作り方を知りたいと思った
3	キャラクターなどどのように プログラミングで動いているか学びたいです

- (7) プログラミングは、どちらかと言えば、料理に似ている
- (8) もう少し練習すればいろいろできるようになりそう

図 8 は、上記の各項目についての回答結果をグラフにまとめたものである。この回答結果から、プログラミングの基本について受講生の理解が進んでいることが確認できた。

『G : 体験後のプログラミングの印象』の質問では、多くの受講生のプログラミングに対する印象の変化がみられた。授業前アンケートでは、「勉強してみたい」と回答した受講生は 19.6% とどまったが、授業後アンケートでは 44.2% まで増加した。逆に、「勉強したいけど、私には難しそう」と回答した受講生は、60.9% から 39.5% まで減少した。

図 9 は、授業前アンケートと授業後アンケートのプログラミングの印象の変化を図示したものである。このグラフは、左にいくほどポジティブな印象、右にいくほどネガティブな印象を持っていることを表している。

また、表 2 は、「その他、授業を受けてプログラミングに関して印象が変わったことがあったら教えてください」の自由記述形式の回答を抜粋したものである。この質問では、30 人の受講生から回答を得た。受講生からは「思ったより簡単」「初心者でも楽しむことができる」など、プログラミングに対して好印象な声が多く寄せられた。

表 3、表 4 は、『H : 授業の感想』のそれぞれの回答を抜粋したものである。表 3 は、「今回の授業を経験してさらに学びたいことがあったら書いてください」の自由記述形式の回答を抜粋したものである。この質問では、11 人の受講生から回答を得た。それぞれの回答から、ビジュアル要素やゲーム要素のあるものに興味や関心を示すことがわかった。表 4 は、「Puppy の良かった点や悪かった点、気づいたことがあったら教えてください」の自由記述形式の回答を抜粋したものである。この質問では、25 人の受講生から回答を得た。Puppy の良かった点としては、「色がかわいい」などの Puppy 特有の UI に関する点や「実行結果が目に見えてわかりやすい」「プログラムがわかりやすい」などの Puppy コンセプトに関する点をあげてもらった。

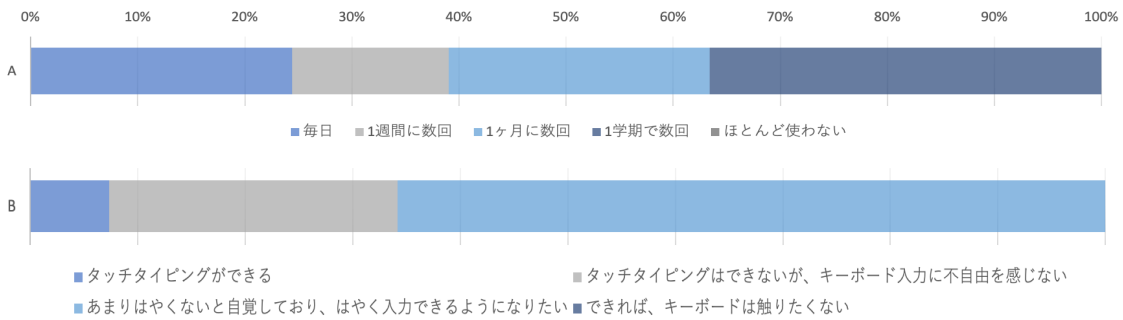


図 6 パソコンの利用頻度とタイピングの習熟度

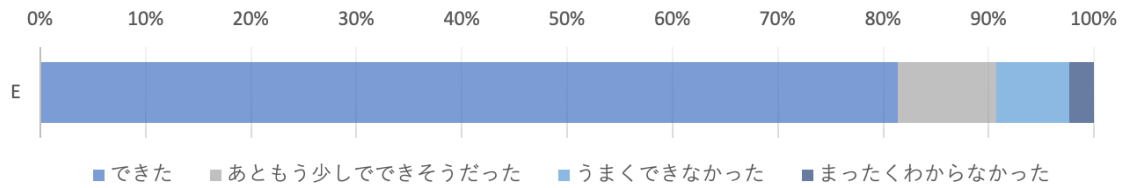


図 7 ドミノ倒しプログラムの達成度 (自己申告)

表 4 Puppy の良かった点や悪かった点, 気づいたこと (自由記述)

1	カラフルでかわいかった
2	実行結果が画像や図としてわかりやすく表示されていて、目に見えてわかるのが楽しかったです
3	わかりやすいプログラムでよかった
4	例題のような感じで、操作の仕方や構文の書き方が書かれていたのが分かりやすくてありがたかった
5	前のページに戻したときに書いていたものが残るともっと良くなると感じました
6	固定とかの英単語が打ちにくかったです

5. 関連研究

プログラミング教育の重要性が世界的に高まっている中、子どもたちがプログラミングをスムーズに学ぶことができるように、様々なタイプの支援が研究されている。

まず、LEGO Mindstorm[4] や Micro.bit[5] などのハードウェアを提供することで、プログラミング学習をスムーズに進める取り組みは人気を博している。この利点は、実世界の物理的な物体をプログラムで操ることで、より直感的なプログラム理解が高まることが期待できる点がある。しかし、ハードウェアを用いたプログラミング学習は、その場にハードウェアがあることが前提となるため、自宅などで気軽に取り組むことは難しい。そのため、継続的な学習という観点からみたときには課題が残るといえるだろう。

プログラミングの障壁となるのは、キーボード操作に慣れない学生が多い点である。近年、Scratch[6], [7] や Blockly[8] に代表されるブロックベースの言語がプログラミング教育に用いられている。これらのアプローチは、短期的な体験コースだけでなく、継続的な学習という点でも

プログラミングの概念をよく学べることが実証されている。しかし、学生らが一歩先のレベルのプログラミング、すなわち C/C++ や Python などの実務レベルのプログラミングに興味を持った場合のことを考えると、テキストベースの環境に慣れておく必要があるのも事実であろう。

6. むすびに

Puppy は、物理シミュレーションによる統合された Web プログラミング環境である。Puppy 言語と呼ばれる簡易化された Python を備え、物理シミュレーション上で可視化された物体の動作から、プログラムを学ぶことができる。

本稿は、Puppy の 60 分体験コースについての実践報告を行った。「プログラミングを勉強してみたい」と感じている学生は、19.6%から 44.2%に増加し、逆に、「勉強したいけど、私には難しそう」と感じている学生は、60.9%から 39.5%に減少した。Puppy は、プログラミングの楽しさを伝え、継続的な学習意欲を高める体験コースの実践ができることが確認された。

Puppy の開発プロジェクトは 2019 年に始まったばかりである。Puppy は、プログラミング教育を研究するプラットフォームとしてオープンソース公開し、研究者や教育者に提供してゆく予定である。また、高等学校の教員がすぐに利用しやすいコースウェアの開発を進め、多くの若者がプログラミングを学びやすい環境を整えてゆきたい。

謝辞 Puppy の最初にユーザ評価に参加していただいた、有意義なユーザ評価をいただいた日本女子大学理学部サマースクールの受講生、神奈川県立厚木高等学校の受講生、日本女子大学家政学部児童学科の受講生に感謝し

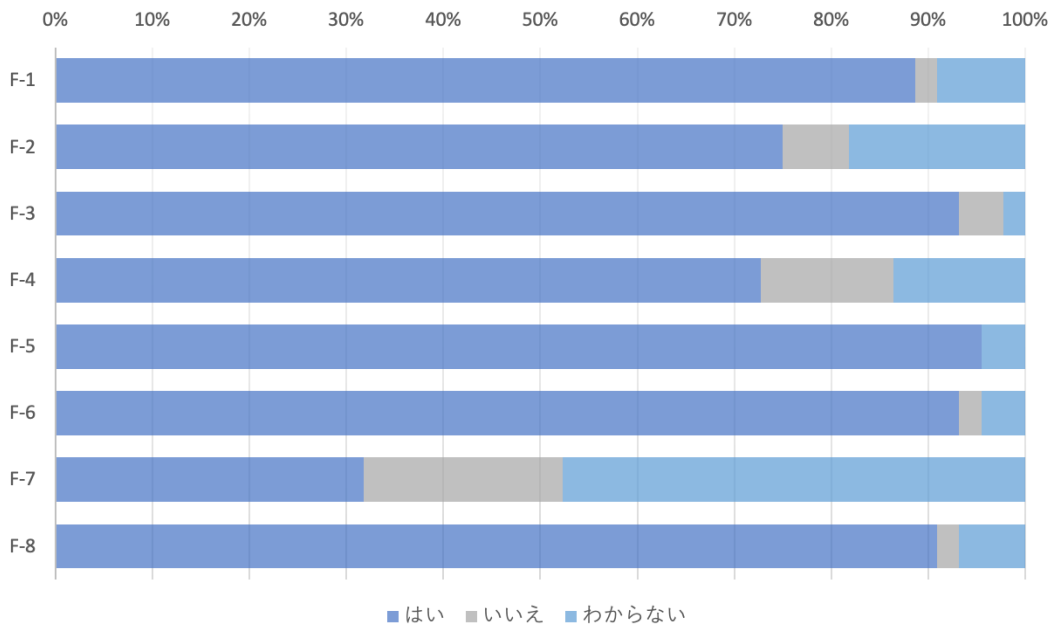


図 8 プログラミングの理解度

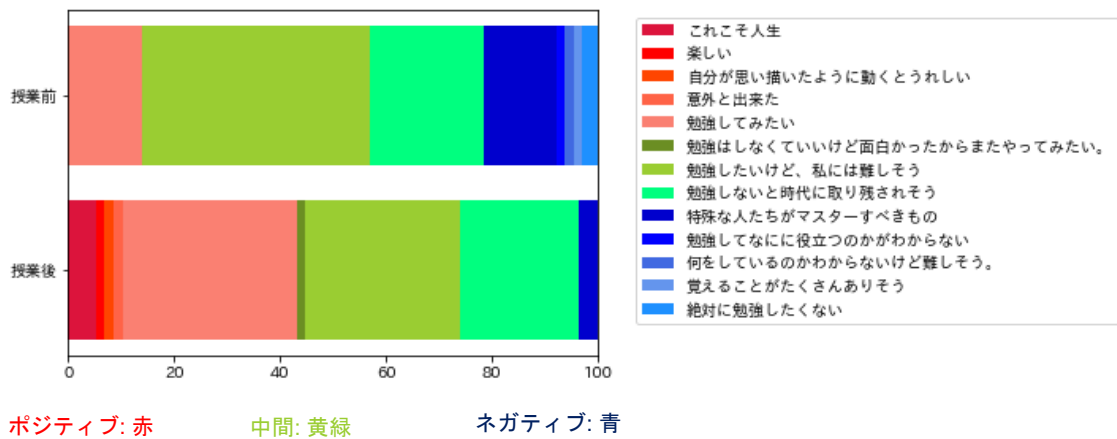


図 9 プログラミングに対する印象の変化

ます。

参考文献

- [1] 秋信有花, 倉光君郎: 自然言語記述からの近似コード生成を用いた初学者プログラミング支援, *xSIG2020* (2020).
- [2] 多田 拓, 秋信有花, 坂根万琴, 倉光君郎: Puppy: プログラムの直感的理解を目指した新しいプログラミング入門環境, *情報処理学会インタラクション 2020 論文集*, pp. 258-263 (2020).
- [3] 河村一樹: 一般情報教育におけるプログラミング教育のあり方について, *技術報告 16*, 東京国際大学商学部 (2011).
- [4] Lawhead, P. B., Duncan, M. E., Bland, C. G., Goldweber, M., Schep, M., Barnes, D. J. and Hollingsworth, R. G.: A Road Map for Teaching Introductory Programming Using LEGO&Copy; Mindstorms Robots, *Working Group Reports from ITiCSE on Innovation and Technology in Computer Science Education*, ITiCSE-WGR '02, New York, NY, USA, ACM, pp. 191-201 (online), DOI: 10.1145/782941.783002 (2002).
- [5] Siever, B. and Rogers, M. P.: Micro: Bit Magic: Engaging K-12, CS1/2, and Non-majors with IoT & Embedded, *Proceedings of the 50th ACM Technical Symposium on Computer Science Education, SIGCSE '19*, New York, NY, USA, ACM, pp. 1237-1238 (online), DOI: 10.1145/3287324.3287527 (2019).
- [6] Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B. and Kafai, Y.: Scratch: Programming for All, *Commun. ACM*, Vol. 52, No. 11, pp. 60-67 (online), DOI: 10.1145/1592761.1592779 (2009).
- [7] Maloney, J. H., Peppler, K., Kafai, Y., Resnick, M. and Rusk, N.: Programming by Choice: Urban Youth Learning Programming with Scratch, *Proceedings of the 39th SIGCSE Technical Symposium on Computer Science Education, SIGCSE '08*, New York, NY, USA, ACM, pp. 367-371 (online), DOI: 10.1145/1352135.1352260 (2008).
- [8] Fraser, N.: Ten things we've learned from Blockly, *2015 IEEE Blocks and Beyond Workshop (Blocks and Beyond)*, pp. 49-50 (2015).