

ゲーム感覚で学べるプログラミング初心者向け 学習支援インターフェース

An Interface That Supports Beginners in Learning Programming Like a Game

榊原 亜実

Ami Sakakibara

法政大学情報科学部デジタルメディア学科

E-mail: ami.sakakibara.2m@stu.hosei.ac.jp

Abstract

Computing skills become necessary for all students in a digital age. In educational dealing with introductory programming, they often use visual programming languages that allow them to design programs by placing visual objects. Also, previous study has shown that block-based programming languages are more effective than text-based ones in introductory learning. For those who are new to programming, Scratch is intuitively usable and is relatively easy to understand among block-based visual programming languages. However, visual programming languages that can be converted to text-based programming languages such as Blockly are not suitable for beginners because they are often for developers. As a result, we can see that beginners need a programming language environment whose level is intermediate between those of Scratch and Blockly. In this study, the purpose is to propose a visual programming interface that enables beginners to enjoy learning programming and improve their coding skills, while using a block-based language. This paper presents a user interface and an assistant chat bot in a block-based programming environment, and shows the result of an experiment on the evaluation of the user interface and the chat bot. The experiment measured the time that participants spent for each task, and conducted pre- and post-experiment questionnaire surveys. This paper also presents the points to be improved and discussions on future research.

1. はじめに

全世界でコンピュータ科学に興味を持つ学生を増やし、IT 人材を確保するために、プログラミング教育が積極的に行われている。入門的なプログラミング学習の場では、テキスト以外の視覚的なオブジェクトを配置することによってプログラムを設計できるビジュアルプログラミング言語を用いることが多い。入門的学習の場において、テキストベースのプログラミング言語よりも、ブロックベースのビジュアルプログラミング言語のほうが有効であることがこれまでの研究で示されている [1]。

プログラミング初心者にとって、Scratch [2]や Blockly Games [3]は直感的に扱えてわかりやすいが、実際にプログラムを書く能力の向上には繋がりにくい。また、Blockly

[4]は実用的であるがため、複雑なプログラムを扱う人の開発環境には適しているが、初心者が扱うには難易度が高い。以上の問題点から、初心者がプログラミングを学ぶには、これらの中間レベルのプログラミング環境が求められていることが分かる。

本研究では、プログラミング初心者がブロックベースのビジュアルプログラミング言語で学習する上で、プログラミング概念を理解し、楽しみながらコーディングスキルの向上させるための初心者向けビジュアルプログラミングインターフェースを提案することを目的とする。楽しく学べる工夫として、ゲームのチュートリアルのように、段階を踏みながら自分で学んでいける仕組みを構築する。また、学習を補助する存在としてアシスタントチャットボットを追加し、先に進めないという事態を未然に防げるようにする。このインターフェースを評価するために評価実験を行い、設定した段階を達成するまでの時間の計測と事前・事後アンケート調査を行った。最後に、改良すべき点と、今後の研究についての議論を述べる。

2. 関連研究

峰内ら [5]は、チャットボットを用いた教育に関する研究として、学生にとってなじみのある LINE を利用して気軽に予習復習ができるツールを開発した。期待される効果として、学習機会を増やすことができる、失敗を恐れず練習ができる、自身の考えをまとめることができる、などを挙げている。

ブロックベースの言語を用いたプログラミング教育のための研究として、Scratch にロボホン専用のブロックを組み入れることで、簡単操作でロボホンを動作させることができるようにした製品がある [6]。実際の動きをイメージし、試行錯誤を繰り返すことで、論理的思考力を育むことができる。また、作成したプログラムは、スタディノートで簡単にクラスや学年、学校間で共有することができる。

ゲーム感覚で学ぶことに関する研究として、ゲームベースラーニング(以下、GBL)と呼ばれる、特定の学習成果をもたらすゲームを使った学習があげられる。Fujimoto ら [7]は、GBL とオープンエデュケーション(開かれた教育)の研究動向について述べている。

3. 提案手法

本研究では、ゲームのチュートリアルのように自分で取り組みながらプログラミングを楽しく学習していくこ

とのできる仕組みを構築する。具体的には、最初に見えるブロックを制限しておき、学習者の到達レベルによって使えるブロックを増やしていくといった仕組みにする(図1)。このような仕組みをここでは機能解除と呼ぶ。

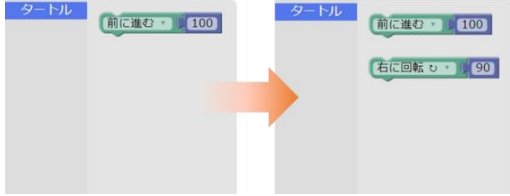


図 1 ある段階に達すると、表示ブロックを増やす

加えて、初心者が自身の力で学んでいくことを助けるためのUIとして、アシスタントチャットボットを提案する。具体的には、ブロックの使い方を詳しくわかりやすく説明してくれるヘルプ機能と、簡単な会話をやり取りできる会話機能をもつチャットボットを構築する。

開発したインターフェースの画面を図2に示す。Blocklyのタートルグラフィックス①をベースに、チャットボットを開くためのボタン②を設置した。ボタン②を押すと画面右にウィンドウ③が出現し、会話機能(3.2.2節)としてのチャットボットがユーザーに話しかけるようにしている。画面上側には、現在ユーザーの所持するポイント④を表記した。ブロックタブ⑤とブロック⑥は、Blocklyのタートルグラフィックスのブロックタブやブロックを用いた。各段階によって使えるブロックを制限するため、それぞれの表示する数を変更した。

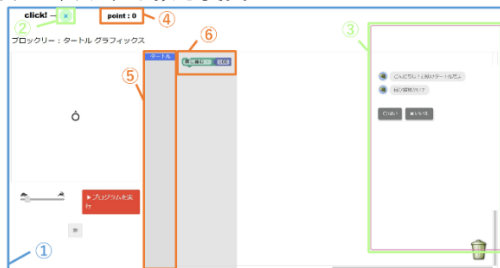


図 2 画面スクリーンショット(初期状態)

3.1. 機能解除

機能解除は、ある一定のレベルに達した際にページ遷移をして、ユーザーの使用できるブロックを増やすためのものである。レベルの判定をするために、解除されるたびにポイントを加算する。機能解除には4段階(表1)あり、段階が進むごとに表示されるブロックが増える。表1の「目標」は次の段階に進むために達成する必要がある目標、「pt」は目標達成時に追加ポイントが付与された後のポイントのことである。また、これらについてはチャットボットの会話機能で質問できるようにしている。ポイントは常に画面に表示されていて(図2④)、機能解除の段階が一つ上がるたびに100ptずつ加算されていく。

初期状態である第1段階のブロックは、「前(後ろ)に進む」ブロックのみであり、ブロックを1つ以上使って実行すると目標達成し、ポイントが加算されて新たなブロッ

クが増える。目標達成の判定は、プログラムを実行してブロックを読み込むときにされる。目標達成となると、現在のポイントに100pt加算され、ページを再読み込みすることにより、ブロックが増えたあとのページが表示される。

2段階目では、使えるブロックとして「右(左)に回転」ブロックが前の状態から1つ増え、合計2種類のブロックが置かれる。3段階目では、そこに更に「繰り返す」ブロックが増える。4段階目に到達することが目標であるため、4段階目では使えるブロックが大幅に増え、自由に組み合わせることができる。

表 1 機能解除の段階

段階	目標	pt
1	まっすぐな線を引いてみよう	100
2	好きな正多角形を描いてみよう	200
3	繰り返すブロックを1つ使って正多角形もしくは幾何学模様のような図形を描いてみよう	300
4	(自由に使えるようになるため目標設定はなし)	-

3.2. チャットボット

チャットボットとユーザーがやりとりするにあたって、前提としてユーザーからチャットボットに送れるメッセージの形式は選択式にする。

3.2.1. ヘルプ機能

使い方が知りたいブロックを右クリックしたときに展開されるメニューから「ヘルプ」を選択すると、チャットボットのウィンドウが立ち上がり、そのブロックについて使い方から活用方法まで様々なことを聞くことができる。たとえば「前に進む」というブロックの場合、まっすぐな線を引けることを図で示し、関連するブロックとして他のブロックを登場させ、組み合わせると作れるものの完成図も一緒に示す。ブロックの組み合わせ例を見て、同じものを作ろうとした場合、答えをすぐに教えるのではなく、工程ごとに手順を示していくようにして、答えにたどりつくための思考力を磨くことができるようにする。

3.2.2. 会話機能

会話機能はヘルプ機能のようにブロックに関する質問ができるのではなく、使い方やインターフェースに関する質問に答えるための機能である。ボタン(図2②)をクリックすると、会話機能としてチャットボットが起動する。この機能は、初めてこのページを見て、何をしたらいいのかわからなくなってしまったユーザーのためにあって、楽しく会話しながら疑問を解消することができる。

最初の会話は決まっていて、質問があるかを尋ねるようにしている。質問がない場合は会話を終了させ、ある場合は具体的な質問内容を特定するためにユーザーにいくつかの選択肢を与えながら進める。最初に質問できる内容は、タートルグラフィックスについて、使い方について、「point」について、ブロックについての4つに分かれている。そのうちの1つを選択すると、それについて質問を繰り返しながら掘り下げて説明する。もしここで、ユーザーがブロックについての疑問を解消しようとしているのであれば、ブロックについて質問できる選択肢を最初に選ぶと、自然にヘルプ機能を使うよう誘導される。

たとえば、「使い方について」を選択すると、画面のUIのスクリーンショットとともに、UIのパーツそれぞれについて番号が割り振られており、その中でわからないパーツについて質問することができる。特定のパーツの番号を選択し説明を聞いた後に、先ほどの質問に戻って他のパーツについても聞けるようにしている。

4. 実装

チャットボットの実装には、BotUI という JavaScript フレームワークを採用した。ボタン(図 2②)横には、最初にクリックするよう促すための文字を表記し、すべきことがわからないユーザーが真っ先にチャットボットに気づけるようにした。チャットボットの立ち上げはボタン以外に、右クリックメニューからもできるようにした。

機能解除は、ページを再読み込みすることによって、ポイントやブロックが追加されるようにした。再読み込みした直後にダイアログボックスを表示して、達成したポイントやブロックが増えたということを示した。表 1 の 3 段階目を達成し、4 段階目に突入すると、ダイアログボックスに加えて、最終段階に達成したことを示すポップアップウィンドウが開くようにした。

5. 実験

5.1. 実験の目的と参加者

本研究で開発した環境を使って学習することによって、プログラミング学習に対するモチベーションが上がるかどうか、また、経験の有無によって生じる差を小さくすることができるかどうかを調べた。集まった参加者は 5 名の中学生(全員男性、平均年齢 14.3 歳)で、そのうち 3 名はプログラミング未経験者であった。

5.2. 評価方法

開発した環境を使用して生じたプログラミング学習に対するモチベーションの変化の有無についてはアンケート調査、プログラミング経験の有無がどれだけ影響していたかは各段階でかかった時間の計測によって調べた。

5.2.1. 各段階でかかった時間の計測

実験中の参加者を動画に撮影し、次の段階になるまでにかかった時間を測定した。カメラは学習者の手元と画面が写るように斜め後ろに設置した。

5.2.2. アンケート調査

実験の事前・事後にアンケート調査を行った。その両方に、5 段階のリッカート尺度を用いた以下の設問を含めた。

- ・ プログラミングが得意だと思うか。
 - ・ 今後、プログラミングを学ぶ機会があったら、楽しみだと思うか。
 - ・ 今後、プログラミングを学ぶ機会があったら、難しいと思うか。
 - ・ 今後、プログラミングを積極的に学びたいと思うか。
- 事後調査には、開発環境に関する以下の設問も含めた。
- ・ チャットボットを使ってみたか。
 - ・ チャットボットはわかりやすいものであったか。
 - ・ 目標を達成すると、達成感を感じたか。

加えて、事後調査では、使っていて難しかったと思うところを自由記述してもらった。

5.3. 実験結果

実験をするにあたって、各参加者に番号をふった。参加者番号 2, 4 は、プログラミング経験者であり、その他の参加者は未経験者である。

5.3.1. 時間計測・観察の結果と考察

各段階を達成するまでにかかった時間を表 2 に示す。また、参加者がチャットボットの各機能を使った回数を表 3 に示す。これらは、撮影した動画を利用して測定した(5.2.1 節)。結果から以下が推察される。

参加者 1, 2, 3, 5 の各段階の時間を比較すると、1 段階目を達成するまでの時間に比べて、2 段階目に時間を取られていた。一方、参加者 4 は 1 段階目の時間のほうが短かった。これは、プログラムを実行する順序が明示されていなかったためであると考えられる。なぜなら、動画を観察したところ、参加者 4 はブロックをワークスペースに置く前に実行ボタンを押して、ブロックを置いたあとに実行ボタンをしばらく押さずにいたためである。また、参加者 4 はプログラミング経験者であり、2 段階目以降の時間は特にかかっていたりはなかった。

参加者 1, 2, 3 で 2 段階目よりも 3 段階目の時間のほうが短かったのは、達成するための目標設定が 2 段階目とそれほど変わりがなかったためであると考えられる。反対に、時間がかかっている参加者 4, 5 は、観察の結果、繰り返すブロックの中に他のブロックを 1 つしか入れることができないと思い込んでいたように見える。その誤解を解消するために、繰り返すブロックのヘルプ機能で 1 つ以上のブロックを入れることができるという説明をしていたが、そのヘルプ機能を使った参加者はいなかった(表 3「繰」列)。

表 2 各段階の達成にかかった時間とその合計

参加者番号	1段階	2段階	3段階	合計
1	0:01:19	0:16:27	0:02:34	0:20:20
2	0:01:31	0:06:16	0:04:22	0:12:09
3	0:05:39	0:34:18	0:06:59	0:46:56
4	0:13:20	0:00:52	0:04:00	0:18:12
5	0:02:57	0:22:56	0:40:57	1:06:50
平均	0:04:57	0:16:10	0:11:46	0:32:53

表 3 チャットボットの機能が使われた回数

参加者番号	会話機能								ヘルプ機能				
	TG	使い方							pt	block	前 右 繰		
		①	②	③	④	⑤	⑥	⑦⑧					
1			2	1				1		1			
2	1			1	1			1		3		1	
3	2		4	3	1	1	1	2					
4	1		2	1				2				1	
5		1	4	1				1	1			1	
合計	4	1	12	7	2	1	2	7	0	4	2	1	0

5.3.2. アンケート調査結果

5.2.2 節で述べた各評価項目について事前アンケート調査と事後アンケート調査の結果を表 4 に示す。指標は、

5: とてもそう思う, 4: まあまあそう思う, 3: どちらともいえない, 2: あまりそう思わない, 1: まったくそう思わない, の5段階である。

表 4 事前・事後調査の回答スコア

参加者	得意		楽しみ		難しそう		学びたい	
	事前	事後	事前	事後	事前	事後	事前	事後
1	3	4	4	5	4	4	5	5
2	3	4	4	5	3	2	3	5
3	2	2	4	3	3	5	3	2
4	4	4	4	5	4	4	3	3
5	2	1	5	4	4	5	5	4
平均	2.8	3.0	4.2	4.4	3.6	4.0	3.8	3.8
分散	0.7	2.0	0.2	0.8	0.3	1.5	1.2	1.7

事前・事後のアンケート調査のスコアについて, 対応のある t 検定を行った。まず, プログラミングが得意だと思うかという項目では, 統計的に有意な差はなかった ($p = 0.62$)。また, プログラミングを学ぶ機会があったら楽しみだと思えるか, 難しそうだと思うかという 2 項目についても, 有意な差はなかった (それぞれ $p = 0.70$, $p = 0.48$)。学習意欲についても, 有意な差はなかった ($p = 1.00$)。よって, 本研究で開発したインターフェースによる心理的な影響は認められなかったといえる。

開発環境に関する設問では, 全員がチャットボットの存在に気づき, それを使っていたことがわかった。「チャットボットはわかりやすいものであったか。」の項目について, 4 人がとてもわかりやすかったと回答し, 残りの 1 人もまあまあわかりやすかったと回答していた。また, 「目標を達成すると, 達成感を感じたか。」という項目については, 5 人中 4 人が達成感を感じていた。また, 開発環境について, 使っていて難しく感じたところを自由記述してもらった。参加者 1 から順に, 以下に示す。

- チャットボットに方法を教えてもらうまで, 何をしたらいいかわからなかった。
- ブロックを繰り返させるのに少し時間がかかった。
- 3 段階目の目標の「幾何学模様」がわからなかった。
- 最後の段階までに色々なブロックを組み合わせると多角形を作ったのが大変だった。
- 多角形を作ることが難しかった。

6. 議論

実験結果から, 実験の前後で心理的な影響はなかったが, 個人間の感じ方やかかった時間には違いがみられた。このことから分かる通り, 個人によって感じ方やかかった時間に, 差が大きく出てしまうことが課題である。特に初心者同士で比べた際に, 大きく差が開いた原因として, 繰り返すブロックの使い方の説明が, 特定の人にとっては充分ではなかったことが挙げられる。繰り返すブロックのヘルプ機能に気づくことができれば, 正しい使い方が理解できる仕組みにしているが, 実験で繰り返すブロックのヘルプ機能を使った人はいなかった。それに対して, 会話機能は比較的使われていたことから, 会話機能にヘルプ機能の内容を移行したほうがよいと考えることもできるかもしれないが, ブロックの種類が増えたときに選択肢が膨大になってしまうため, 別の機能を設けるべきである。ヘルプ機能を全く使わなかった参加者がいた

ことから, 右クリックメニューにも問題があると考えられる。Blockly のインターフェースを改造することで, 使い方を知りたいブロックを置くとヘルプ機能に移行するようにもできたかもしれない。

結果として, 参加者が自力で最後まで達成できたという点ではプログラミング入門者向けのインターフェースが開発できたといえる。また, 実験参加者の中でも経験者に焦点を当てて考えると, Scratch と Blockly の中間レベルのインターフェースが開発できたといえる。しかし, 繰り返すブロックが正しく使えていた人があまりいなかったことを踏まえると, 論理的思考力の向上が図れるインターフェースを開発できたとは言い難い。

7. おわりに

ゲーム感覚で学べるプログラミング入門者向け学習支援インターフェースを作成した。本研究では, ある段階を達成すると新たな機能が解除されるような仕組みと, アシスタントチャットボットを導入した。

ユーザー評価から, 機能解除やチャットボットの会話機能の有用性は示されたが, ヘルプ機能としてのチャットボットは有効でないことが示された。この問題は, Blockly のインターフェースを改造するなど, ヘルプ機能を立ち上げるための操作を単純にすることで解決できると考えられる。本研究により, 繰り返すブロックを論理的に理解させるための機能が必要であることが示唆された。

文 献

- [1] D. Weintrop and U. Wilensky, "Comparing Block-Based and Text-Based Programming," *ACM Transactions on Computing Education*, vol. 18, no. 1, pp. 3:1-3:25, 2017.
- [2] M. Resnick, et al., "Scratch: Programming for All," *Communications of the ACM*, vol. 52, no. 11, pp. 60-67, 2009.
- [3] Google, "Blockly Games," [Online]. Available: <https://blockly.games/>. [Accessed Jan. 2020].
- [4] Google, "Blockly," [Online]. Available: <https://developers.google.com/blockly/>. [Accessed Jan. 2020].
- [5] 峰内暁世, 松葉龍一, 戸田真志, 鈴木克明, "チャットボットを利用した学びの促しを支援するツールの開発," 大学 ICT 推進協議会, 2017.
- [6] シャープマーケティングジャパン株式会社ビジネスソリューション社, "プログラミング学習支援ツール," [Online]. Available: <https://www.sharp-sbs.co.jp/education/programming/>. [Accessed Jan. 2020].
- [7] T. Fujimoto, K. Shigeta and Y. Fukuyama, "The Research Trends in Game-Based Learning and Open Education," *Educational Technology Research*, vol. 39, no. 1, pp. 15-23, 2016.