

# プログラミングコースにおける評価の可能性診断のためのブロックベースのテキストプログラミング環境の必要性

金容天<sup>1</sup> 金子美<sup>1</sup> 金龍<sup>2</sup> 李元揆<sup>1, a)</sup>

**概要:** 人工知能技術の進歩により、プログラミング教育の重要性及び必要性が強調されており、海外の多くの国々でもプログラミング教育の重要性が強調されている。しかしながら、初心者がプログラミングを学習することには多くの困難が伴う。そのような理由から、スクラッチ、エントリー、パーソンズ・プロブレムズのようなプログラミング環境が開発された。プログラミング教育における重要なプロセスの一つは評価であり、評価は学習目標を達成するための助けとなる。本論文では、スクラッチのようなブロックベースのプログラミング環境を利用し、Pythonのコマンドを組み合わせた、追加、修正、削除できる評価を含むプログラミング環境の必要性を活用して、高等学生を対象に授業を行った。また、使用したプログラミング環境が初心者のプログラミングプロセスの評価に役立つことを確認した。本研究の意義は、ブロックベースのテキストプログラミング環境のプログラミングプロセスの評価可能性を検証したことにある。

**キーワード:** ブロックベースのテキストプログラミング環境、Python プログラミング、評価

## The Necessity of a Block-based Text Programming Environment for Possibility Diagnosis of Assessments in Programming Process

KIM YONGCHEON<sup>1</sup> KIM JAMEE<sup>1</sup> KIM YONG<sup>2</sup> LEE WONGYU<sup>1, a)</sup>

Received: August 1, 2023, Accepted: August 1, 2023

**Abstract:** With the advancement of artificial intelligence technology, the importance and necessity of programming education are being emphasized, and many countries around the world are also highlighting the importance of programming education. However, beginners face many difficulties in learning programming. For this reason, programming environments like Scratch, Entry, and Parsons Problems have been developed. One of the crucial steps in programming education is evaluation, as it aids in achieving learning goals. In this study, we utilized the need for a programming environment that includes evaluation and allows for the combination, addition, modification, and deletion of Python commands based on blocks like Scratch, and conducted classes for high school students. We then verified whether the used programming environment could assist in evaluating beginners' programming process. The significance of this study lies in diagnosing the potential for evaluating the programming process in a block-based text programming environment.

**Keywords:** Block-based Text Programming Environment, Python Programming, Assessment

### 1. はじめに

人工知能技術の発展により、プログラミング教育の重要性が強調されている。イギリスは国家教育カリキュラム

(National Curriculum in England)を通じてコーディングを必須とし、中国とフィンランドもプログラミング教育を必須にしている[1][2]。また、日本も2025年から大学入学共通テストに「情報」科目を設け、プログラミング、データサイエンスなどに関連する問題を出題する[3][4]。全世界的にプログラミング教育が行われているが、初心者が専門家向けのテ

<sup>1</sup> Korea University, Seoul 02841, South Korea

<sup>2</sup> Korea National Open University, Seoul 03087, South Korea

a) Correspondence : lee@inc.korea.ac.kr

キストベースのプログラミング言語を使用するのは容易ではない。テキストプログラミング言語を使用する過程で、誤字による文法のエラーは初心者がプログラミングに興味を失う一因となる。このような理由から、初心者向けのプログラミング教育では、ブロックベースのプログラミング言語が多く使用されるイギリスでも Key Stage 1、2 段階の 3～6 年生向けのフィジカルコンピューティング教育ではブロックベースの Micro:bit を使用し、Key Stage 3 段階の 7～9 年生ではテキストベースのプログラミング言語を使用している。

ブロックベースのプログラミング言語は、コマンドをマウスでドラッグアンドドロップして組み合わせる方式を使用するため、テキスト入力によるエラーが発生しないという利点がある。しかし、産業現場で使用することはできないという欠点があるため、韓国のエントリー、米国の Blockly のようにブロックベースのコマンドを組み合わせるとテキストコマンドに変換するハイブリッドプログラミング環境が使用されることもある。これは専門家が使用するテキストプログラミングの経験が得られるという利点があるが、単純に変換するだけの機能は初心者のプログラミング能力を向上させることはできない。このような問題を解決するために、Python コマンドが記述されたブロックを組み合わせるプログラミング能力を向上させるために Parsons' Problem が開発されたのである。

プログラミング教育で重要な一過程は評価である。評価は学習目標を達成するための助けとなるからだ[5]。2000 年代初頭のプログラミング教育と評価はテキストベースで行われた。学習者はプログラミング学習過程でテキストコマンドの使用に苦勞し、評価過程でも完全な理解なしに問題解決を試みるが多かった[6]。初心者向けのプログラミング教育では Scratch、Blockly、Micro:bit、Parsons' Problems などのブロックベースのプログラミング環境を使用することと同様に、評価過程でも初心者向けの環境が考慮されるべきである。つまり、初心者のプログラミング過程を評価するためには、テキストベースではなく Parsons' Problem のようなブロックベースのテキストプログラミング環境の開発が必要である。

この論文では、Scratch のようなブロックベースのプログラミング環境で Python コマンドを組み合わせ、追加、修正、削除できるプログラミング環境の必要性を利用し、高校生を対象に授業を行った。そして、使用されたプログラミング環境が初心者のプログラミングの過程を評価するのに役立つことが確認された。本研究は、ブロックベースのテキストプログラミング環境におけるプログラミング過程の評価可能性を診断することを目的としている。"

## 2. ブロックベースのプログラミング環境

### 2.1 スクラッチ

スクラッチは MIT メディアラボが開発されたプログラミング言語である。テキストベースのプログラミング言語はコマンドを視覚的に提供しないため、初心者がテキストをタイピングする難しさがある。スクラッチは利用可能なコマンドをブロックとして提供し、ユーザーはマウスでドラッグアンドドロップしてコマンドを組み合わせるプログラムを作る。動作や音声などのアニメーション機能も提供するため、ゲームデザイン活動を通じて問題解決能力を向上させるのに効果的である[7]。しかし、スクラッチを活用したプログラミング授業では評価まで結びつけるのは難しい。正しく動作しないプログラムを生徒たちに提供し、間違っただ部分を修正したり、必要なブロックを追加する形で評価項目を構成することは可能だが、教師が生徒たちの提出したプログラムを一つずつ全て実行しなければならないという制約がある。

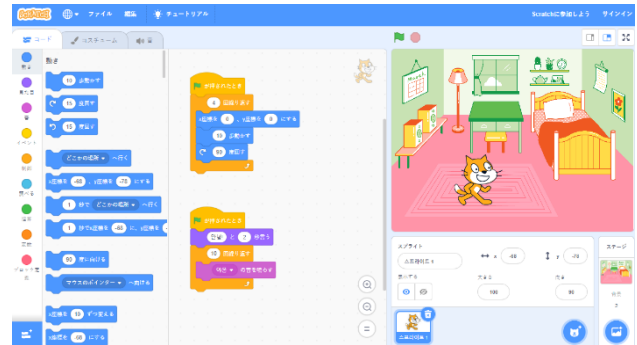


図 1 スクラッチプログラミング環境

Figure 1 Scratch Programming Environment

### 2.2 エントリー

エントリーは韓国のネイバーコネク ト財団が運営するブロックベースのプログラミング環境である[8]。スクラッチと同様に、ブロックをマウスで組み合わせるプログラムを作ることができる。

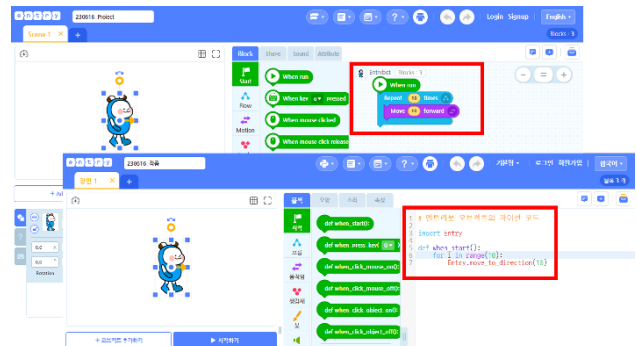


図 2 エントリープログラミング環境

Figure 2 Entry Programming Environment





図 6 ChatGPT のエラーに対するフィードバック  
Figure 6 Feedback on errors in ChatGPT

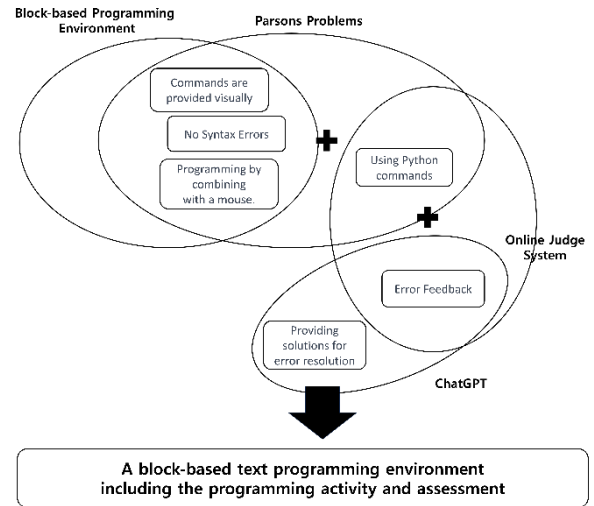


図 7 ブロックベースのテキストプログラミング環境の構造  
Figure 7 Block-based text programming environment structure

## 4. プログラミング活動と評価を含む環境

### 4.1 Online Judge System

この研究で使用した、プログラミング活動と評価を同時に行うことのできるプログラミング環境の構造は、図 7 のようになっている。まず、視覚的にコマンドを提供し、文法のエラーがなく、テキストのコマンドをタイピングせずにマウスでコマンドを組み合わせるプログラミングするブロックベースのプログラミング環境の利点を持っている。次に、Parsons' Problems やオンライン採点システムと同様に、Python のプログラミングコマンドを使用する。そして最後に、文法のエラーがある場合は、オンライン採点システムのようにエラーのフィードバックを提供する。

### 4.2 Online Judge System

この研究では、Parsons' Problems のような Python のコマンドを使用しながら、Scratch や Entry のようにプロジェクトを作成できるプログラミング環境を使用して授業を進行した。

授業は、高校 1、2 年生の 19 人を対象に、1 日に 3 時間ずつ、2 日間行われた。使用したプログラミング環境は、Parsons' Problems の限界であるプロジェクト管理の難しさ、コマンドの順序のみを配置可能な形式を解決したもので、ユーザーは自分が実行したプロジェクトを確認でき、コマンドの順序だけでなく、編集、削除、追加の機能も提供する。

学習者はプログラミング活動を行い、同じ環境で評価まで行うことができる。評価プロセスでエラーが発生する場合、オンライン採点システムのようにエラーフィードバ

クが提供される。答えが正しい場合は、答えと 100%一致するというメッセージが表示され、答えが間違っている場合は、答えと一致する割合が表示される。

授業は 4 つのタイプで行われました。

まず、真似である。学習者はプログラムを作るために必要なコマンドを提供された順番通りにそのまま移動させればプログラムが完成する。このプロセスを通じて、学習者は Python のコマンドの形を習得することができ、ブロックベースのプログラミング環境の一つの利点である文法のエラーがないため、プログラミングの失敗を経験しないという利点がある。

次に、配置である。真似と同じコマンドがブロックとして提供されるが、順序がランダムに配置されているため、学習者は正しい順序も組み合わせる必要がある。このプロセスは Parsons' Problems と同じ形式で、学習者は Python のコマンドを読むことができ、問題解決手順であるアルゴリズムに集中することができる。Parsons' Problems との違いは、問題解決に必要なでないコマンドも一部提供して難易度を上げることができるという点です。学習者は問題解決に必要なコマンドだけを使用してプログラムを作る必要がある。

第三に、修正である。一つのプログラムを作るために使用するテキストコマンドを初心者がすべて入力するのは容易ではない。そこで、大部分のコマンドを提供するが、文法のエラーが発生する可能性のある部分のコマンドを除外して提供し、ユーザーがコマンドの一部だけを修正する活動である。この活動を通じて、学習者はエラーメッセージを読んで問題を解決する練習をすることができる。

最後に、追加である。プログラムを完成するために必要な基本のコマンドを一部欠落させた状態でコマンドを提供し、学習者は必要なコマンドを追加する形である。このプロセスを通じて、学習者は Python のコマンドの一行を直接

追加する経験をし、自分でプログラミングができるという自信を得ることができる。

上記の4つのタイプで高校1、2年生の19人を対象に授業した後、配置、修正、追加の形で評価を行った。一つのプラットフォームで授業と評価が行える環境を使用することで、活動後の評価まで自然につながる授業を進行することができた。

## 5. 結論および提言

本研究の目的は、ブロックベースのテキストプログラミング環境でのプログラミングプロセスの評価の可能性を診断することである。そのために、ブロックベースのプログラミング環境でテキストベースのプログラミングが可能な環境を使用し、19人の高校生を対象に授業を行った。その結果、以下のような示唆が得られた。

一つ目は、プログラミング能力を向上させるためには、コマンドを組み合わせることを含めて初心者が直接コマンドを追加、修正、削除するプロセスが必要であるということである。Parsons Problemの場合、テキストプログラミングコマンドをブロックとして利用できるという利点があるが、ユーザーが直接コマンドを修正または追加する活動が提供されていないため、プログラミング活動が行われたとは言えない。これは、学習者が直接コマンドを書く活動がない場合、「本当のプログラミング」を経験していないという研究結果[14]と一致している。

二つ目は、初心者レベルに適した段階的なプログラミング学習が必要であるということである。初心者が一度に完璧なプログラムを作成することは困難である。エラーのある部分を見つけて正しく修正するためには、提供されたコマンドをそのまま追って組み合わせながらテキストコマンドを学ぶ必要がある。その後、コマンドの一部を修正したり追加したりすることでプログラミングに対する自信を増す必要がある。なぜなら、文法のエラーによる失敗はプログラミングに対する自信を失う結果となるからである[15][16]。

三つ目は、学習者にエラーへの対応能力を育てる必要があるということである。つまり、初心者がプログラムを作成する際にエラーを経験しないようにガイドするのではなく、エラーが発生した場合に、エラーの位置と原因を理解し、自分が望むプログラムを作成できる力を育てる必要がある。

今後の研究では、ブロックベースのテキストプログラミング環境を使用して評価を行い、Online Judge Systemのような自動採点や、ChatGPTのようなエラー解決のフィードバックを提供する環境の開発が必要とされている。

Funding: This research was funded by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT), grand number No. 2021R1A2C2013735.

## 参考文献

- [1] DFE, U. K. “National curriculum in England: computing programmes of study”. Retrieved July, 16, 2014.
- [2] Wu, L., Looi, C. K., Multisilta, J., How, M. L., Choi, H., Hsu, T. C., & Tuomi, P. “Teacher’s perceptions and readiness to teach coding skills: A comparative study between Finland, Mainland China, Singapore, Taiwan, and South Korea”, *The Asia-Pacific Education Researcher*, Vol. 29, pp. 21-34, 2020.
- [3] 岩間徳兼, 田村志穂美, 板東信幸, 飯田直弘, & 池田文人. “新科目「情報 I」に対する高等学校等の反応: 進路指導担当者の回答から”, *高等教育ジャーナル: 高等教育と生涯学習*, Vol. 30, pp. 17-28, 2023.
- [4] 赤澤紀子. “大学入試における教科「情報」の出題の調査分析”, *電気通信大学紀要*, Vol. 32, pp. 54-61.2020.
- [5] Kim, M. J., Lee, W. G., & Kim, J. M. “Presenting the Development Direction Through the Analysis of Tool used to Measure Computational Thinking”, *The Korean Association of Computer Education*, Vol. 20, pp. 17-25, 2017
- [6] Gomes, A., & Mendes, A. J. “Learning to Program - Difficulties and Solutions”, *International Conference on Engineering Education – ICEE 2007*, pp. 283-287, 2007
- [7] Erol, O., & Çırak, N. S. “The effect of a programming tool scratch on the problem-solving skills of middle school students”, *Education and Information Technologies*, Vol. 27, pp. 4065-4086, 2022.
- [8] Gim, N. G. “Development of life skills program for primary school students: Focus on entry programming”, *Computers*, Vol.10, 2021.
- [9] Ericson, B., McCall, A., & Cunningham, K. “Investigating the affect and effect of adaptive parsons problems”, *Proceedings of the 19th Koli Calling International Conference on Computing Education Research*, pp. 1-10, 2019.
- [10] Kim, S., Park, J., Jeon, S., & Seo, D. “Web-Based Online Judge System for Online Programming Education”, *2022 IEEE International Conference on Consumer Electronics (ICCE)*, pp. 1-3, 2022.
- [11] Surameery, N. M. S., & Shakor, M. Y. “Use chat gpt to solve programming bugs”, *International Journal of Information Technology & Computer Engineering*, Vol. 3, pp. 17-22, 2023
- [12] Unal, A., & Topu, F. B. “Effects of teaching a computer programming language via hybrid interface on anxiety, cognitive load level and achievement of high school students”, *Education and Information Technologies*, Vol. 26, pp. 5291-5309, 2021
- [13] Lytle, N., Dong, Y., Cateté, V., Milliken, A., Isvik, A., & Barnes, “Position: Scaffolded coding activities afforded by block-based environments”, *2019 IEEE Blocks and Beyond Workshop*, pp. 5-7, 2019.
- [14] K. Powers, S. Ecott, & L.M. Hirshfield. “Through the looking glass: teaching CS0 with Alice”, *ACM SIGCSE Bulletin*, Vol. 39, pp. 213–217. 2007.
- [15] Hu, Y., Chen, C. H., & Su, C. Y. “Exploring the effectiveness and moderators of block-based visual programming on student learning: A meta-analysis”, *Journal of Educational Computing Research*, Vol. 58, pp. 1467-1493, 2021.
- [16] T. Jenkins. “On the difficulty of learning to program”, *Proceedings of the 3rd Annual Conference of the LTSN Centre for Information and Computer Sciences* pp. 53-58, 2002.



著者紹介



**金容天**  
高麗大学校 大学院 コンピュータ  
科学部



**金子美**（正会員）  
高麗大学校教育大学院 コンピュー  
タ教育専攻



**金龍**  
放送通信大学校 eラーニング学科



**李元揆**（正会員）  
高麗大学校 大学院 コンピュータ  
科学部