

ブラウザ上で動作する DNCL 学習環境 WaPEN と BlocklyPEN の改良

岡本 勇紀, 中西 通雄

Yuki Okamoto, Michio Nakanishi

大阪工業大学情報科学部コンピュータ科学科

Dept. of Computer Science, Faculty of Information Science and Technology, Osaka Institute of Technology

Email: nakanishi.michio@gmail.com

あらまし : 大学入試センター試験科目の情報関係基礎では、擬似言語の DNCL が用いられている。DNCL を Web ブラウザ上で実行できる環境には、WaPEN や BlocklyPEN などがある。本研究では、WaPEN で関数を定義できるように機能拡張し、BlocklyPEN にブロックの追加やインターフェイスの改善などを行った。大学生 6 人に評価してもらい、BlocklyPEN がプログラミング初学者にも使いやすい実行環境になっていると考える。今後は BlocklyPEN がより使いやすくなるよう、デバッグ機能の追加などを行う必要がある。

キーワード : DNCL, WaPEN, BlocklyPEN, ブロックプログラミング

1. はじめに

大学入試センター試験科目の情報関係基礎では、DNCL という日本語表記の疑似プログラミング言語を用いる⁽¹⁾。DNCL の実行環境として PEN が作成されたが、JRE をインストールしなければならない。そこで、Web ブラウザ上で動作する WaPEN や BlocklyPEN などが作成された。WaPEN は、PEN と同様に、キーボード入力によって DNCL を記述し実行できる環境である⁽²⁾。入力支援ボタンを押すことで対応した文が挿入されるため、キーボード入力の手間を減らすことができる。BlocklyPEN は、ブロックで DNCL プログラムを作成し実行できる環境である⁽³⁾。Google が開発した Blockly と WaPEN を統合し、改良したものである。ブロックでプログラミングすることで、プログラムの構造が見やすくなるほか、キーボード入力の手間や構文エラーを減らすことができる。これにより、BlocklyPEN はプログラミング初学者でも簡単にプログラミングできる実行環境となっている。

本研究では、WaPEN と BlocklyPEN に対して機能の拡張や見た目の改善を行い、初学者にとって使いやすい環境になるように改善した。

2. 研究概要

2.1 関数の導入

PEN は関数に対応していたが、WaPEN は関数に未対応であったので、WaPEN に関数の構文を追加した。また、BlocklyPEN も関数に対応させた (図 1)。



図 1 関数定義ブロック

2.2 画面インターフェイスの改良

2017 年度に作成された BlocklyPEN は、画面に余白が多いためブロックを組む画面が狭く、プログラ

ムを作成しにくかった。そこで、画面の構図の見直しやフォントサイズの調整などを行った。(図 2)。また、ブロックを DNCL に変換したものを表示する領域を設けた。これにより、実際に DNCL をキーボード入力した場合の様子を確認することができる。

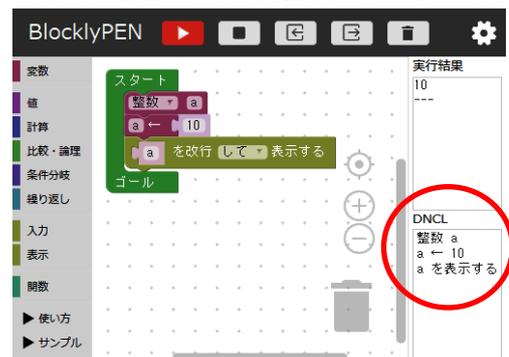


図 2 BlocklyPEN のメイン画面

2.3 学習環境の設定

環境設定の変更を簡単に行える画面を作成した (図 3)。WaPEN では、変数の宣言が必要かどうか設定するには、WaPEN のファイルを編集する必要があった。そこで、BlocklyPEN では設定画面を設け、設定を簡単に変更できるようにした。

また、学習者のレベルに合わせて使用できるブロックを制限できる、学習者レベルという設定を設けた。これにより、学習者のレベルに対応したブロックのみを表示させることが可能になる。学習者にとって余計な情報を与えることがなくなり、勉強しやすい環境にすることができる。レベルを増やすことで、様々な学習レベルに対応することができる。

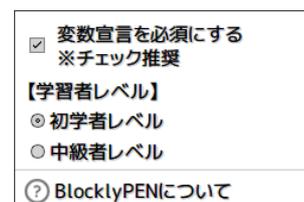


図 3 環境設定画面

2.4 言語仕様の拡張

本研究でブロックをいくつか追加した。ある程度学習した人が情報関係基礎の問題などを解く際に、ブロックを組む手間を減らすことができる(図4)。



図4 複合代入演算ブロック

なお、「a *= 10」は、WaPEN では書けないので、「a = a * (10)」の形に変換して WaPEN に渡している。

2.5 その他

そのほか、BlocklyPEN の使い方やブロックの種類などが記載された解説ページの作成、ブロックやバグの修正などを行った。

3. 評価と考察

プログラミング経験者として、本学部生の3回生1人と他大学の文系学生1人、およびプログラミング経験のない大学生4人の合計6人に、WaPEN とBlocklyPEN の両方を使用して3つのプログラムを作成してもらい、その後、質問に回答してもらった(表1)。プログラムは、代入、演算、条件分岐、繰り返しのいずれかを使用する、3行から7行程度のものであった。

表1 質問と肯定的な回答数

質問	WaPEN	BlocklyPEN
プログラミングしやすい	3	5
プログラムの構造が見やすい	4	6
コンパイルエラーが少ない	3	4
文字入力が少ないと良い	1	6

表1から、BlocklyPEN は、ブロックプログラミングの利点を活かすことができていると、初学者にとって、キーボード入力でのプログラミングする実行環境よりも使いやすい実行環境になっていると考えられる。しかし、他大学のプログラミング経験者1人のみ、「プログラミングしやすい」の項目で、WaPEN の方が高評価であった。これは、被験者用のプログラムが簡単で、ブロックプログラミングの利点を体感しにくかったためと考えられる。

筆者は、平成30年度の情報関係基礎第3問のプログラムを、WaPEN とBlocklyPEN の両方を使って作成した。その結果、日本語入力の少なさ、プログラムのコピーのしやすさ、値を変更する際の手間の少なさを3つの点で、BlocklyPEN の方が優れていると感じた。

以上から、ある程度複雑なプログラムを作成し、ブロックプログラミングの利点を体感することができれば、BlocklyPEN でプログラミングしやすいと言える。

4. 課題

本研究を進める中で、被験者による評価の最中や、機能を追加したことにより、新たな問題が浮き彫りとなった。

4.1 デバッグ機能の追加

BlocklyPEN に、ステップ実行機能や変数の値を表示する機能を追加すべきだと考えている。

被験者に繰り返し文を使ったプログラムを作成してもらった。しかし、プログラムには誤りが含まれており、無限ループが発生した。そこで、無限ループが発生する理由を口頭で説明したが、被験者はプログラムの流れと変数の値を同時に覚えることが難しく、理解するのに時間を要した。プログラムをステップ実行し、変数の値が変移する様子を見ながら説明することで、被験者の理解を早めることができたと考えられる。

Blockly はブロックを光らせる機能を有しており、WaPEN はステップ実行が可能である。したがって、BlocklyPEN にステップ実行機能を追加することは可能だと考えられる。変数の値を表示する機能も追加することで、プログラムの流れを確認しやすくなり、プログラミング的思考を理解する助けになると考えている。図5は、デバッグ機能を追加した際、変数の値を表示する画面のイメージ図である。



図5 変数の値を表示する画面

4.2 追加したブロックの評価

図1や図4のブロックを追加したが、被験者に使用してもらった機会はなかった。もし、BlocklyPEN を数回以上の授業で使用してもらえれば、追加したブロックの有用性を検証する必要がある。

5. 結論

本研究では、WaPEN に関数の構文を追加したほか、BlocklyPEN を改善して、6人の大学生による評価を行った。その結果、BlocklyPEN は初学者にとって使いやすい実行環境であると言える。

今後は、BlocklyPEN にデバッグ機能などを追加し、より使いやすい実行環境となるように改善する必要がある。

謝辞

本研究の一部はJSPS 科研費 JP17K01088 の助成をうけた。

参考文献

- (1) 独立行政法人大学入試センター: "センター試験用手順記述標準言語(DNCL)の説明", http://www.dnc.ac.jp/sp/albums/abm.php?f=abm00004841.pdf&n=H23_dncl.pdf (2011.1)
- (2) 中西渉: "DNCL の Web 上での実行環境の開発", 情報教育シンポジウム, pp.160-162 (2017.8)
- (3) 岩崎みのり: "ビジュアルブロック化した DNCL の Web ブラウザ上での実行環境", 大阪工業大学情報科学部卒業論文 (2018.1)