

電子情報通信学会「著作権規程」の基本方針より

電子的利用については、著作者本人ならびに所属機関が著作者の著作物の全文を著作者の研究室や所属機関のホームページもしくはプレプリントサーバに掲載する場合、一定条件の下で出版社版 PDF もしくは早期公開版 PDF の掲載を許諾します。

※掲載条件等、詳細については「利用申請基準」を御覧ください。

本会出版物に掲載された論文等の著作物の利用申請基準より

条件 A : 権利表示 (例 copyrightc2013 IEICE)

条件 B : 出版社版 PDF(紙版をスキャンで作成したもの含) の掲載。著者最終版は不可。

条件 C : 出所の明示 (例 著作者名、書名 (題号)、雑誌名、巻、号、頁、発行年など)

条件 D : 著作者の了解

条件 E : IEICE Transactions Online トップページへのリンク

上記、公開基準に従い出版社版 PDF を公開いたします。

なお、IEICE Transactions Online トップページは下記になります。

<https://search.ieice.org/>

ビジュアル型言語とテキスト型言語の学習時の生体情報の比較について

梅澤 克之[†] 中澤 真^{††} 平澤 茂一^{†††}

[†] 湘南工科大学 〒251-8511 神奈川県藤沢市辻堂西海岸 1-1-25

^{††} 会津大学短期大学部 〒965-0003 福島県会津若松市一箕町大字八幡門田 1-1

^{†††} 早稲田大学 〒169-8555 東京都新宿区大久保 3-4-1

E-mail: †umezawa@info.shonan-it.ac.jp, ††nakazawa@jc.u-aizu.ac.jp, †††hira@waseda.jp

あらまし 近年、プログラミングの入門としてビジュアル型のプログラミング言語（以降、ビジュアル型言語と呼ぶ）が使われるようになってきている。その後はC言語やJava言語などのテキスト型プログラミング言語（以降、テキスト型言語と呼ぶ）に移行していくことになる。しかしシームレスな移行方法は確立されていない。本研究プロジェクトでは、ビジュアル型言語からテキスト型言語への移行の方法論を確立することを目的とする。本研究が確立されると、プログラミング言語の初学者がビジュアル型言語による学習から始めて、その後、シームレスかつ自発的にテキスト型言語の学習に遷移できるようになることが期待できる。我々は以前の研究で、ビジュアル型言語とテキスト型言語の中間的な特徴を持つ中間コンテンツの提案を行った。そしてこの中間コンテンツによる学習を両言語の学習の間に挟むことによって、テキスト型言語の理解度が上がることを示した。本研究では、ビジュアル型言語とテキスト型言語の学習の際に、学習者の生体情報を計測し、両言語の学習の際の生体情報に何らかの差異があるか否かを確認する。

キーワード ビジュアル型プログラミング言語, テキスト型プログラミング言語, 学習分析, 生体情報

About the Comparison of Biometric Information during Learning of Visual- and Text-based Programming Language

Katsuyuki UMEZAWA[†], Makoto NAKAZAWA^{††}, and Shigeichi HIRASAWA^{†††}

[†] Shonan Institute of Technology Tsujido-Nishikaigan 1-1-25, Fujisawa, Kanagawa 251-8511, Japan

^{††} Junior College of Aizu Monden 1-1, Yahata, Ikki-Machi, Aizuwakamatsu, Fukushima 965-0003, Japan

^{†††} Waseda University Okubo 3-4-1, Shinjuku, Tokyo 169-8555, Japan

E-mail: †umezawa@info.shonan-it.ac.jp, ††nakazawa@jc.u-aizu.ac.jp, †††hira@waseda.jp

Abstract Beginners in learning programming learn visual-based programming languages such as Scratch, while experts use text-based programming languages such as C and Java. However, a seamless transition from a visual-based programming language to a text-based programming language has not been established. Our research project aims to establish a methodology for the transition between the two types of languages. In our previous research, we proposed an intermediate content with intermediate features between visual-based language and text-based language. And we showed that the comprehension of the text-based programming language is improved by inserting the intermediate content learning between the learning of both types of languages. In this study, we measure the biometric information of the learner during the learning of the visual-based programming language and the text-based programming language, and confirm whether there is any difference in the biometric information during the learning of both types of programming languages.

Key words Visual-based Programming, Text-based Programming, Learning Analytics, Biometric Information.

1. はじめに

近年、プログラミングの入門としてビジュアル型のプログラミング言語（以降、ビジュアル型言語と呼ぶ）が使われるよう

になってきている。その後はC言語やJava言語などのテキスト型プログラミング言語（以降、テキスト型言語と呼ぶ）に移行していくことになる。しかしシームレスな移行方法は確立されていない。

我々は、ビジュアル型言語からテキスト型言語への移行の方法論を確立することを目的とする研究プロジェクトを開始している。具体的には、ビジュアル型言語とテキスト型言語の学習の利点を有し、両者の差異を埋める教育コンテンツ（中間型言語とよぶ）を検討・試作し、実証実験を通して、学習結果だけでなく学習中の学習状態を評価する。これにより、その中間型言語の有効性を評価し、今後の初等・中等プログラミング教育に役立つ教育コンテンツ（中間型言語）を完成させる。ビジュアル型言語とテキスト型言語の溝を埋める想定の中間型言語の評価に関しては、今までは、学習後のアンケートや成績など、学習効果の結果のみに着目しており、理解できるようになったか否かを評価するのみであったが、これらの評価方法では、今回提案する中間型言語の効果を正確に測ることが出来ない。我々の研究プロジェクトでは、学習後の従来の評価方法に加えて、学習中の脳波や視線、表情などの生体情報および学習中の学習履歴を用いて学習状態を計測し、中間型言語がビジュアル型言語とテキスト型言語の中間的な役割を果たし、スムーズな移行に貢献しているかを分析・評価する。本研究が確立されると、プログラミング言語の初学者がビジュアル型言語による学習から始めて、その後、シームレスかつ自発的にテキスト型言語の学習に遷移できるようになることが期待できる。

我々は以前の研究 [1] で、上述の研究プロジェクトの一部として、まず中間コンテンツの提案に着目し、ビジュアル型言語の学習とテキスト型言語の学習の間に、我々が提案する中間コンテンツを用いた学習を挟む方がその後のテキスト型言語の理解度が高まるということを実証実験を通して示した。さらに、提案する中間コンテンツが、ビジュアル型言語とテキスト型言語の中間に位置づけられる特徴を有することをアンケートにより評価した。

本研究では、ビジュアル型言語とテキスト型言語の学習の際に、学習者の生体情報（脳波、心拍、表情）を計測し、両言語の学習の際の生体情報に何らかの差異があるか否かの分析を行う。脳波を目的変数、心拍と 10 種類の表情を説明変数として、重回帰分析を行ったところ、説明変数の係数の正負の符号に差異が見られた。

2. 従来研究

2.1 ビジュアル型言語について

ビジュアル型言語は、ブロックベースの命令型言語、またはフローベースの関数型言語の 2 つの大きなカテゴリに分類される。

Mason ら [2] は、可能な限り類似するように設計された 2 つの環境のいずれかでいくつかの単純な問題をプログラムするような数百人規模の実験を行い、2 つのカテゴリの相対的なメリットを評価する実証研究を行った。

Robinson ら [3] は、ブロック型言語 (Scratch) からテキスト型言語への移行について研究を行った。Scratch の魅力的な機能の 1 つは、学習者が構文エラーを回避して簡単に理解して使用できることである。Scratch はプログラミングロジックの学習に重点を置いており、学習者は論理的に考える方法を学ぶ

が、プログラムの作成方法を学べるものではないと述べている。

2.2 ビジュアル型言語とテキスト型言語の比較研究

Mladenović ら [4] は、プログラミングの基本概念の 1 つであるループについて、学生の誤解に関する調査を 207 人の小学生を対象に実施した。生徒たちは、ブロック型言語 (Scratch) とテキスト型言語 (Logo および Python) の 3 つのプログラミング言語を学んだ。ブロック型言語では学生のループに関する誤解が最小限に抑えられることがわかった。ネストされたループなどタスクが複雑になるにつれて、違いがより明確になった。この研究では、構文の問題が解消されるためプログラミングの初学者向けにはビジュアル型言語を使用する正当性を主張している。ただしこの研究では両言語のギャップを埋める事には言及していない。

Navarro-Prieto ら [5] は、1 つの実験を行って、ビジュアル型言語がテキスト型言語よりも理解しやすい理由を画像処理の心理学の観点で説明している。画像処理は意味情報へのより迅速なアクセスを容易にするため、ビジュアル型言語は、テキスト型言語よりもデータフロー関係を迅速に心的表現を構築できるという仮説を検証した。テキスト型言語のプログラマーは最初に制御フローの心的表現にアクセスし、次にデータフローの心的表現にアクセスすることが分かった。

Xu ら [6] は、既存の学術データベースを調査するという方法で、ブロック型言語のプログラミング環境とテキスト型言語のプログラミング環境が、学生の認知的と感情的な学習成果の両方に及ぼす全体的な影響を調べた。初心者プログラマーのためのブロック型言語の有用性 (utility) と有効性 (efficacy) の統計的優位性を示すことはできなかったが、ハイブリッド型の更なる調査の必要性を述べている。

2.3 ハイブリッド型言語の提案

Daskalov ら [7] は、初学者向けのテキスト型言語とビジュアル型言語のハイブリッド型言語を使用するための環境を提案している。ブロック型ではなくフロー型のビジュアル言語とテキスト型言語のハイブリッド環境であり、初学者のプログラミングトレーニングに適していると主張している。

Weintrop [8] は、テキスト型言語とビジュアル型言語とハイブリッド型言語の比較を行った。結論として、ハイブリッド型言語が両方の祖先の特徴を示している一方で、特定の領域でブロック型言語とテキスト型言語よりも優れていることを明らかにした。

また、Weintrop [9] は、PencilCode を使用して、ブロック型言語、ハイブリッド型言語、テキスト型言語を使用して移行のスキルの影響を比較した。3 つグループの学習者がそれぞれの言語で 5 週間の学習をして第 6 週目以降はすべてグループが Java を使用したテキストベースに移行した。ブロック型言語の学習者の方がテキスト型言語の学習者よりもループや変数を理解する姿勢が優れていた。ブロック型言語の欠点は、大規模で複雑なプログラムを構築するのが難しいことが分かった。また、ハイブリッド型は、テキスト型とブロック型を切り替えて使う必要があったため、公正な比較が出来なかった。

Alrubaye ら [10] は、ブロックのみの表示にする、テキスト

のみの表示にもできる，ブロックの領域とテキストの領域を並べても表示できる，というようなツールを作成して移行を評価した．ブロックベースからテキストベースの学習アプローチと比較した場合，平均してハイブリッドベースのアプローチは，プログラミングの基礎，暗記，移行の容易さに関する学生の理解を30%以上向上させることがわかった．

2.4 ビジュアル型言語とテキスト型言語間のギャップに関する研究

Tóth ら [11] は，ビジュアル型言語とテキスト型言語間のギャップの存在を指摘した．この研究ではビジュアル型言語 (MIT App Inventor 2) から Java Bridge Code Generator を知識伝達のメディエーターとして使用して，テキスト型言語 (Android Studio) に移行する方法を検証した．この論文では Java Bridge Code Generator が，ビジュアル型言語とテキスト型言語の間のギャップを埋めるのに役立ったと主張している．

Weintrop ら [12] はビジュアル型言語から入ったが学習者とテキスト型言語から入った学習者がテキスト型言語に移行した後の知識移転の変化を実験で評価し，両者に有意差が無い事を示している．移行ではなくテキスト型言語のスキルを修得後の比較を行った．

我々は，以前の研究 [1] で，ビジュアル型言語の学習とテキスト型言語の学習の間に，我々が提案する中間コンテンツを用いた学習を挟む方がその後のテキスト型言語の理解度が高まるということを実証実験を通して示した．さらに，提案する中間コンテンツが，ビジュアル型言語とテキスト型言語の中間に位置づけられる特徴を有することをアンケートにより評価した．

また，我々は，脳波計を用いて学習者の学習状況を把握する方法の研究を推進中である．我々は，課題遂行中の脳波情報を取得し，難しい課題に取り組むと β/α の値が高くなることを示した [13] [14]．この研究の過程で，ビジュアル型言語 (Scratch) の課題に取り組んでいる際の脳波とテキスト型言語 (C 言語) に取り組んでいるときの脳波に差異があることを確認した．具体的には，ビジュアル型言語に関しては課題の難易度が高くなっても β/α の値が高くなることはなかった．これによりビジュアル型言語とテキスト型言語の学習過程には異なる思考が行われている可能性が示唆された [15]．

3. 本研究プロジェクトの概要

本研究プロジェクト全体の目標としては，ビジュアル型言語とテキスト型言語の学習の利点を有し，両者の差異を埋める教育コンテンツ (中間型言語とよぶ) を検討・試作し，実証実験を通して，学習結果だけでなく学習中の学習状態を評価する．これにより，その中間型言語の有効性を評価し，今後の初等・中等プログラミング教育に役立つ教育コンテンツ (中間型言語) を完成させる．まず，ビジュアル型言語とテキスト型言語の溝を埋める想定の中間型言語の検討・施策し，それを用いて実証実験を行う．評価に関しては，今までは，学習後のアンケートや成績など，学習効果の結果のみに着目しており，理解できるようになったか否かを評価するのみであった．これらの評価方法では，今回提案する中間型言語の効果を正確に測ることが出来

ない．本研究では，学習後の従来の評価方法に加えて，学習中の脳波や視線，表情などの生体情報および学習中の学習履歴を用いて学習状態を計測し，中間型言語がビジュアル型言語とテキスト型言語の中間的な役割を果たし，スムーズな移行に貢献しているかを分析・評価する．本研究の全体概要を図1に示す．

3.1 プログラミング間の差異を埋めるプログラミング学習コンテンツの開発

現状のビジュアル型言語は見た目が楽しい，すぐ実行して動作する，文法エラーが存在しないなどの特徴がある．これに対してテキスト型言語は，文字ばかりであり，1文字でも間違えれば動作せず，グラフィカルなことをやろうとするととても手間がかかる，というような特徴がある．提案する中間型言語は，シンプル (テキスト型言語特有の追加知識不要)，素早いフィードバック (実行結果がすぐわかる)，文法エラーが起りにくく，論理エラーの箇所がわかり易い，というような特徴を持つものを試作する．

3.2 学習時の生体情報 (脳波や視線) を取得する実証実験

3.1 で作成した中間型言語，およびビジュアル型言語，テキスト型言語のそれぞれを用いて，学習時に脳波や視線等を取得する実証実験を実施する．具体的には，大学のプログラミング実習等の授業，あるいは付属高校の生徒，研究室大学生を対象に，Scratch，プログラミン，Blockly 等のビジュアル型言語と Java，JavaScript，C 言語等のテキスト型言語を用いた実証実験を実施する．なお，実験参加者に関しては経験者と初学者の両方を実験対象とすることで，より効果的に各言語の差異を明らかにできると考える．

3.3 各言語学習時の差異の分析・抽出し，学習プロセスの溝の明確化

上記 3.2 の実験結果を用いて各言語の学習時の学習状態の差異を分析する．学習時の編集履歴と脳波を計測することによって，学習者がどのような課題でつまづいているのか，あるいは逆にどのようなときに学習意欲が高まっているのか等を明らかにした上で，学習者の編集プロセスとその際の脳波によって学習者を分類し，比較・分析を行っていく．

4. 実験方法

4.1 実験参加者

今回の実験の参加者は，湘南工科大学の3年生と4年生の8名である．彼らは同じ大学の同じ学部でプログラミング関連のコースを数年間学んだ経験を有する．彼らのプログラミングスキルは同等である．

4.2 実験に使用する Web サービス

ビジュアル型言語のプログラミングを行う際は，Google Blockly [16] (図2参照) を利用した．また，テキスト型言語のプログラミングを行う際は，Paiza [17] (図3参照) を利用した．

4.3 実験に使用する問題

以下の6種類の実験を行った．

実験 V1: マウスによるテキスト入力 (ビジュアル型)

実験 V2: 問題 A (ビジュアル型)

実験 V3: 問題 B (ビジュアル型)

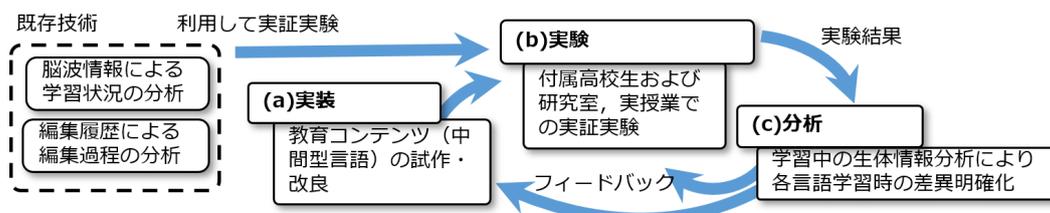


図 1 本研究プロジェクトの全体概要

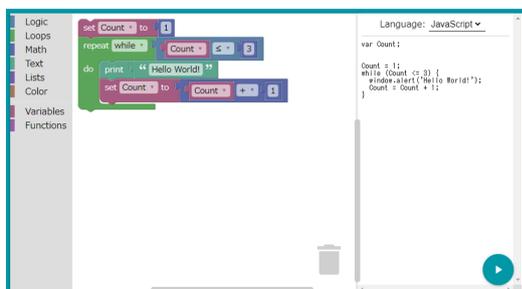


図 2 Google Blockly の画面

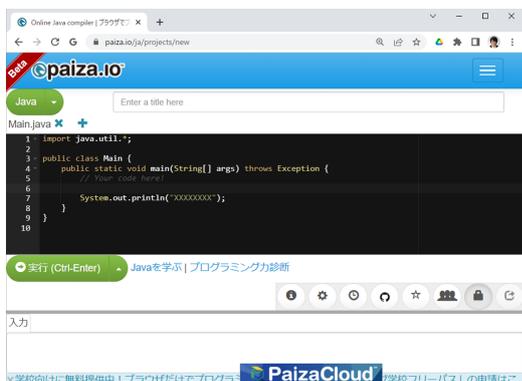


図 3 Paiza.io の画面

Connector とは、NeuroSky 社が提供する MindWave Mobile 2 との通信機能を提供するドライバである。また、この脳波計で取得できる脳波の種類は表 1 に示す 8 種類であり、各値は単位のない 4 バイトの浮動小数値である。

今回は従来研究 [18] に則り、低 α 波、高 α 波、低 β 波、高 β 波の 4 種類を使用する。また今後は低 α 波を α_l 、高 α 波を α_h 、低 β 波を β_l 、高 β 波を β_h と示す。今回は既存研究 [19] より β/α の組み合わせ 4 種類 (β_l/α_l , β_h/α_l , β_l/α_h , β_h/α_h) と、さらに低周波と高周波の平均として $(\beta_l + \beta_h)/(\alpha_l + \alpha_h)$ を加えた 5 種類の β/α の数値を使用する。今後は $(\beta_l + \beta_h)/(\alpha_l + \alpha_h)$ を β_{l+h}/α_{l+h} と示す。

表 1 取得できる脳波の種類

種類	周波数 (Hz)
δ 波	0.5–2.75
θ 波	3.5–6.75
低 α 波 (α_l)	7.5–9.25
高 α 波 (α_h)	10–11.75
低 β 波 (β_l)	13–16.75
高 β 波 (β_h)	18–29.75
低 γ 波	31–39.75
中 γ 波	41–49.75

実験 T1: キーボードによるテキスト入力 (テキスト型)

実験 T2: 問題 A (テキスト型)

実験 T3: 問題 B (テキスト型)

ここで、実験 V1 と T1 で用いた入力文、実験 V2 と T2 で用いた問題 A、実験 V3 と T3 で用いた問題 B を図 4 に示す。

テキスト入力文	問題A	問題B
君が代は 千代に八千代に さざれ石の いわおとなりて こけのむすまで	30÷7の商と余り を求めよ。 (答えの小数点以下は切り捨て)	*を4回繰り返し て表示せよ。

図 4 テキスト入力文と問題 A, 問題 B

4.4 実験に使用する生体機器

4.4.1 簡易脳波計

脳波計測には、NeuroSky 社の MindWave Mobile 2 を使用する。脳波のデータはこのヘッドセットと ThinkGear Connector 間を Bluetooth で接続し、ログ収集アプリが ThinkGear Connector と TCP/IP 通信を行うことで収集する。ThinkGear

4.4.2 心拍計

心拍の計測にはユニオンツール株式会社の“myBeat”ウェアラブル心拍センサ WHS-1 を使用する。myBeat は心拍、加速度、体表温の 3 種類の生体データを計測できる。専用の USB 受信機と付属の専用ソフトを使用し計測が可能である。今回は計測した数値のうち心拍 (HR : Heart Rate) を使用する。

4.4.3 表情判定

表情判定を行うには表情の録画と表情の分析が必要である。表情の録画には、Logicool 社製の C920n の Web カメラを使用する。表情の分析には、撮影した映像を解析する (株) シーエーシー 社の心 sensor (Affectiva 社製の感情認識エンジンを搭載) を使用する。このアプリケーションは撮影された映像またはリアルタイムの映像から人の顔を認識し、感情 (怒り、軽蔑、嫌悪、恐れ、喜び、悲しみ、驚き、真顔、表情の豊かさ、感情価など) を数値化可能である。また、瞬きや顔の向き、顔のパーツの座標などを数値化可能である。今回は前述の数値化された 10 種類の感情を使用する。これらの感情の中で感情価以外は 0 から 100 の間の数値で表され、感情価は -100 から 100 の数値で表される。今回は怒り、軽蔑、嫌悪、恐れ、悲しみ、驚きを負の感情とし、喜びを正の感情とする。

5. 実験結果

4.1節で示した8名の実験参加者それぞれについて、V1～V3, T1～T3の計6種類の実験を行った。それぞれの実験で取得した生体情報は、5種類の脳波 (β_l/α_l , β_h/α_l , β_l/α_h , β_h/α_h , β_{l+h}/α_{l+h}) と1種類の心拍数 x_1 , 10種類の表情 (怒り x_2 , 軽蔑 x_3 , 嫌悪 x_4 , 恐れ x_5 , 喜び x_6 , 悲しみ x_7 , 驚き x_8 , 真顔 x_9 , 表情の豊かさ x_{10} , 感情価 x_{11}) である。以降, β_l/α_l は y_1 , β_h/α_l は y_2 , β_l/α_h は y_3 , β_h/α_h は y_4 , β_{l+h}/α_{l+h} は y_5 とする。

6. 分析

従来研究 [20] でも示されている通り, 2つの実験の差に着目して分析を行う。2つの実験の差とは, 例えば個々の実験結果に着目するのではなく, V1とV2の実験結果の差に着目する。具体的には, V2-V1, V3-V1, T2-T1, T3-T1に着目する。そうすると, ビジュアル型については実験参加者8名についてV2-V1とV3-V1の計16セットの実験結果が得られたことになる。テキスト型についても同様に16セットの実験結果が得られた。

この16セットの実験結果を使って5種類の脳波を目的変数, 心拍数と10種類の表情を説明変数とする重回帰分析を行った。

6.1 全説明変数を用いた重回帰分析

まずは, 11種類の説明変数をすべて使って目的変数である脳波を説明する重回帰式を求める。ビジュアル型の結果を式(1)～(5)示す。また, テキスト型の結果を式(6)～(10)に示す。また, それぞれの重回帰式の決定係数(寄与率)を表2に示す。一般的な目安としては決定係数(寄与率)が0.5や0.6以上であれば, 有用な回帰式が得られたと考えられる。よってすべての目的変数について有用な回帰式が得られたといえる。

表2 全説明変数を用いた重回帰式の決定係数(寄与率)

式番号	決定係数(寄与率)
(1)	0.950
(2)	0.951
(3)	0.901
(4)	0.939
(5)	0.987
(6)	0.978
(7)	0.730
(8)	0.976
(9)	0.853
(10)	0.879

6.2 重回帰式の差異

全説明変数を用いた重回帰式(1)～(10)を見ると, 説明変数 x_1 , x_3 , x_4 , x_6 , x_7 , x_9 の係数の符号がビジュアル型とテキスト型で正負が反転している。このようにビジュアル型言語とテキスト型言語の学習時の生体情報の違いを観測できた。

7. まとめと今後の課題

本研究では, 脳波を目的変数, 心拍数と表情を説明変数とした重回帰分析を行った。その結果, 一部の説明変数の係数の正負が, ビジュアル型言語とテキスト型言語で反転した。つまり, ビジュアル言語とテキスト言語の学習時の生体情報の違いを観察することができたと言える。

今後の課題としては, 我々の研究プロジェクトの全体目標の達成に向けて, 参加者を増やしての実験および分析, 学習結果の分析に加えて, 学習中の脳波などの生体情報を用いた学習状態の分析, 実授業への適用などを行っていく予定である。

今後の課題としては, より多くの実験参加者と測定対象の生体情報の種類を増やして, さらなる実験を行いたいと考えている。また最終的には, ビジュアル型言語とテキスト型言語の間のギャップを埋める中間言語を作成して評価したいと考えている。

研究倫理について

今回の実験は湘南工科大学研究倫理委員会の承認を得ている。また実験参加者から実験参加に関する署名を得ている。

謝辞

本研究は JSPS 科研費 JP22H01055, JP21K18535, JP20K03082 の助成を受けたものです。また, 本研究の一部は, 早稲田理工研特別勘定 1010000175806 NTT 包括協定共同研究, および, 経営情報学会「ICTと教育」研究部会の助成を受けたものです。本研究成果の一部は早稲田大学理工総研プロジェクト研究「次世代 e-learning に関する研究」の一環として行われたものです。

文献

- [1] 梅澤克之, 石田昂大, 中澤真, 平澤茂一. ビジュアル型言語からテキスト型言語への移行のための中間コンテンツの提案と評価. 電子情報通信学会 (IEICE) 技術報告, pp. 1-7, 2022.
- [2] Dave Mason and Kruti Dave. Block-based versus flow-based programming for naive programmers. *2017 IEEE Blocks and Beyond Workshop (B&B)*.
- [3] William Robinson. From scratch to patch: Easing the block-stext transition. *In Proceedings of the 11th Workshop in Primary and Secondary Computing Education (ACM)*, pp. 96-99, 2016.
- [4] Monika Mladenović, Ivica Boljat, and Žana Žanko. Comparing loops misconceptions in block-based and text-based programming languages at the k-12 level. *Education and Information Technologies 23(4)*, pp. 1483-1500, 2018.
- [5] Raquel Navarro-Prieto and Jose J. Ca nas. Are visual programming languages better? the role of imagery in program comprehension. *International Journal of Human-Computer Studies*, pp. 799-829, 2001.
- [6] Zhen Xu, Albert D. Ritzhaupt, Fengchun Tian, and Karthikeyan Umamathy. Block-based versus text-based programming environments on novice student learning outcomes: A meta-analysis study. *Computer Science Education*, Vol. 29, pp. 177-204, 2001.
- [7] Rumén Daskalov, George Pashev, and Silvia Gaftandzhieva. Hybrid visual programming language environment for programming training. *TEM Journal*, Vol. 10, pp. 981-986, 2021.

ビジュアル型の全説明変数を用いた重回帰式

$$y_1 = -0.008x_1 - 0.729x_2 - 1.171x_3 - 0.000x_4 + 37.027x_5 - 0.192x_6 - 0.076x_7 - 1.636x_8 - 0.171x_9 - 0.078x_{10} + 0.106x_{11} + 0.180 \quad (1)$$

$$y_2 = -0.033x_1 - 1.389x_2 - 2.087x_3 - 0.611x_4 + 48.561x_5 - 0.659x_6 - 1.627x_7 + 10.971x_8 - 0.703x_9 - 0.136x_{10} + 0.074x_{11} + 0.527 \quad (2)$$

$$y_3 = +0.002x_1 - 0.472x_2 - 0.807x_3 - 0.616x_4 + 21.685x_5 - 0.098x_6 - 0.389x_7 - 0.584x_8 - 0.129x_9 - 0.030x_{10} + 0.039x_{11} + 0.287 \quad (3)$$

$$y_4 = -0.007x_1 - 1.170x_2 - 1.301x_3 - 0.737x_4 + 32.062x_5 - 0.571x_6 - 1.343x_7 + 5.860x_8 - 0.634x_9 - 0.086x_{10} + 0.015x_{11} + 0.372 \quad (4)$$

$$y_5 = -0.000x_1 - 0.414x_2 - 0.560x_3 - 0.272x_4 + 14.017x_5 - 0.234x_6 - 0.516x_7 + 2.380x_8 - 0.239x_9 - 0.035x_{10} + 0.017x_{11} + 0.144 \quad (5)$$

テキスト型の全説明変数を用いた重回帰式

$$y_1 = +0.015x_1 - 2.427x_2 + 0.170x_3 + 2.624x_4 + 2.271x_5 + 0.348x_6 + 1.181x_7 - 1.685x_8 + 0.436x_9 - 0.005x_{10} + 0.186x_{11} + 0.209 \quad (6)$$

$$y_2 = +0.033x_1 - 2.375x_2 + 0.345x_3 + 2.990x_4 + 4.833x_5 + 0.350x_6 + 0.741x_7 - 1.542x_8 + 0.282x_9 - 0.068x_{10} + 0.171x_{11} + 0.400 \quad (7)$$

$$y_3 = +0.014x_1 - 0.463x_2 + 0.433x_3 + 0.764x_4 - 1.768x_5 + 0.231x_6 + 0.755x_7 + 0.239x_8 + 0.299x_9 + 0.010x_{10} + 0.064x_{11} + 0.251 \quad (8)$$

$$y_4 = +0.038x_1 - 0.723x_2 + 0.801x_3 + 2.879x_4 + 1.928x_5 + 0.245x_6 + 0.825x_7 + 0.030x_8 + 0.230x_9 - 0.071x_{10} + 0.141x_{11} + 0.414 \quad (9)$$

$$y_5 = +0.018x_1 - 0.498x_2 + 0.386x_3 + 1.112x_4 - 0.835x_5 + 0.046x_6 + 0.444x_7 + 0.365x_8 + 0.094x_9 - 0.020x_{10} + 0.075x_{11} + 0.194 \quad (10)$$

- [8] David Weintrop and Uri Wilensky. Between a block and a typeface: Designing and evaluating hybrid programming environments. *IDC '17: Proceedings of the 2017 Conference on Interaction Design and Children*, pp. 183–192, 2017.
- [9] David Weintrop. Blocks, text, and the space between: The role of representations in novice programming environments. In *2015 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, pp. 301–302, 2015.
- [10] Hussein Alrubaye, Stephanie Ludi, and Mohamed Wiem Mkaouer. Comparison of block-based and hybrid-based environments in transferring programming skills to text-based environments. *Proceedings of the 29th Annual International Conference on Computer Science and Software Engineering*, pp. 100–109, 2019.
- [11] Tomáš Tóth and Gabriela Lovászová. Mediation of knowledge transfer in the transition from visual to textual programming. *Informatics in Education*, Vol. 20, pp. 489–511, 2021.
- [12] David Weintrop and Uri Wilensky. Transitioning from introductory block-based and text-based environments to professional programming languages in high school computer science classrooms. *Computers & Education*, Vol. 142, , 2019.
- [13] Katsuyuki Umezawa, Tomohiko Saito, Takashi Ishida, Makoto Nakazawa, and Shigeichi Hirasawa. Learning state estimation method by browsing history and brain waves during programming language learning. *Trends and Advances in Information Systems and Technologies. World-CIST'18 2018. Advances in Intelligent Systems and Computing*, Springer, Cham., Vol. 746, , 2018.
- [14] Katsuyuki Umezawa, Tomohiko Saito, Takashi Ishida, Makoto Nakazawa, and Shigeichi Hirasawa. Learning-state-estimation method using browsing history and electroencephalogram in e-learning of programming language and its evaluation. *Proceeding of the International Workshop on Higher Education Learning Methodologies and Technologies Online (HELMeTO 2020)*, pp. 22–25, 2020.
- [15] Katsuyuki Umezawa, Yutaka Ishii, Makoto Nakazawa, Michiko Nakano, Manabu Kobayashi, and Shigeichi Hirasawa. Comparison experiment of learning state between visual programming language and text programming language. *2021 IEEE International Conference on Engineering, Technology & Education (TALE)*, pp. 01–05, 2021.
- [16] Blockly games. <https://blockly.games/>. Accessed: 17 January 2023.
- [17] Paiza learning. <https://paiza.jp/works>. Accessed: 18 December 2022.
- [18] Kouji Yoshida, Yuuta Sakamoto, Isao Miyaji, and Kunihiko Yamada. Analysis comparison of brain waves at the learning status by simple electroencephalography. In *Advances in Knowledge-Based and Intelligent Information and Engineering Systems 2012*, pp. 1817–1826, 2012.
- [19] Katsuyuki Umezawa, Tomohiko Saito, Takashi Ishida, Makoto Nakazawa, and Shigeichi Hirasawa. An electroencephalogram-based method for judging the difficulty of a task given to a learner. *Proceeding of the IEEE 17th International Conference on Advanced Learning Technologies (IEEE ICALT 2017)*, pp. 384–386, 2017.
- [20] 梅澤克之, 中澤真, 中野美知子, 平澤茂一. プログラミング学習時の脳波と心拍と表情の関係について. 電子情報通信学会 (IEICE) 技術報告, pp. 14–19, 2022.