

# 初学者向けプログラミング教育に関する意識調査 — GUIとCUIのどちらがよいか? —

飯尾 淳<sup>1,a)</sup>

**概要:** 大学の1年生レベルを想定した、初学者向けプログラミング教育においてはCUI型のプログラミングを題材とすべきか、あるいはGUI型のプログラミングを題材に用いるべきか、それぞれメリットとデメリットがあり判断が難しい。本研究では、どちらが望ましいかについての意識調査を、既にIT産業の現場で活躍している、あるいはIT関連の教育を実施している専門家群と、このような教育を受講する立場である学生群の、2つのグループに対して実施した。本稿ではその結果について考察する。

**キーワード:** 初学者, プログラミング教育, GUI, CUI

## Survey on Programming Education for Beginners — Which is Better: GUI or CUI? —

JUN IIO<sup>1,a)</sup>

**Abstract:** It is difficult to determine which is better for entry-level programming education, in choosing a programming model for beginners between Character User Interface (CUI) and Graphical User Interface (GUI). This study has conducted the awareness survey among two groups; participants in one group are professionals and participants in the other group are students. The results of the survey are reported and discussed in this paper.

**Keywords:** Beginners, Programming education, GUI, CUI

### 1. 背景

大学において、プログラミング経験の乏しい、初学者の学生に対するプログラミング演習を担当すると、予想外のところで躓く学生が多いことに気づく。情報技術に対するごく初歩的な概念や技法は高等学校の教科「情報」で学んでいるはずであるにもかかわらず、さらには、ひととおりの前提知識を講義で事前に教えておいたはずなのに、演習を実施すると様々なトラブル状況に陥る。

また、「エディタで編集してCUI(Character User Interface)でコマンドを投入する」というやり方にはじめて触れる学生が多く、「資料で示したとおりにプロンプト記号から打ち込んでしまう」、あるいは、「作業ディレクトリ(current working directory)が分からなくなってしまう」、「エディタでプログラムを打ち込み終えて、最後に保存す

るのを忘れている」など、プログラミング以前のトラブルで躓いてしまうというケースが頻発する。

これらのトラブルを避けるためには、プログラミング以前の、OSの操作や基本概念といった基礎的な知識を事前に時間をかけて教えることが必要となるが、学生に求める前提知識に関してレベル差があるために基礎的事項に時間を十分にとることができないという制約もあり、プログラミング演習のなかで並行して指導していかざるを得ないという事情もある。また、プログラミング演習を実施するにあたり、教育すべき本質はどのようにプログラムを作成するか、プログラムによっていかにして問題を解くかということであって、ソースコードの編集方法やコンパイルの仕方といった話題はあくまで付随的なものと考えられるが、それらをきちんと理解していないとソフトウェアを作成することができないという矛盾も孕んでいる。これは、どのような環境を用いてプログラムを作成させるのか?という選択にも通じる問題である。

このような背景を鑑みて、本論文では、初学者に対する

<sup>1</sup> 中央大学  
Chuo University, Hachioji, Tokyo 192-0393, Japan  
<sup>a)</sup> iiojun@tamacc.chuo-u.ac.jp

```

1  #!/System/Library/Frameworks/JavaScriptCore.
  framework/Versions/A/Resources/jsc
2
3  var stdin, stdout;
4
5  // stdin
6  stdin = this.readline();
7
8  // process
9  stdout = stdin.split("").reverse().join("");
10
11 // stdout
12 print(stdout);

```

図 1 タイプ A : CUI 型プログラム (コマンドとして実装され、実行される)

Fig. 1 Type-A: CUI-type program, which is implemented and runs as a command

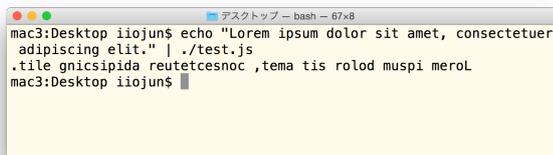


図 2 CUI 型プログラムの実行

Fig. 2 Running the CUI-type program

プログラミング教育、とくに演習を交えた実践的な入門教育を提供するうえで、どのような方法を選択すべきかを検討する。

## 2. 問題設定と調査方法

本研究では、問題を単純化するために、プログラミング演習の入口を GUI (Graphical User Interface) 型のプログラミング環境で実施すべきか、あるいは旧来の CUI 型のプログラミング環境で実施すべきかの 2 択を考える。それらに対し、実際に現場で活躍している IT 技術者または IT 関連教育の従事者というプロフェッショナルな方々はどのように考えるか、また、教育を受ける立場の学生としてはどのように考えるかについての意見聴取を実施した。

なお、本問題設定は、プログラミング環境として CUI がよいか GUI がよいかを問うものではなく、プログラム課題として CUI ベースのプログラムがより理解しやすいのか、あるいは GUI ベースのものの方が理解しやすいのかを問うものである。操作対象としては文字列の反転という共通課題を設定したが、この課題設定に対しても、「より直感的なものの方がよいのではないか」という指摘が専門家のコメントとしてあったことを付記しておく\*1。

### 2.1 問題の設定

題材としては、「処理対象として与えた文字列を反転さ

\*1 課題を共通化するのではなく、GUI の課題はよりグラフィカルにしたもののほうがよいのではないかと指摘を、査読者からも頂戴した。

```

1 <html>
2 <head>
3   <meta charset="utf-8">
4   <title>情報システムのテンプレート</title>
5   <script type="text/javascript">
6     function do_action () {
7
8       var in_field = document.getElementById("in");
9       var out_field = document.getElementById("out");
10
11       out_field.value = in_field.value.split("").reverse().join("");
12
13     };
14   </script>
15 </head>
16 <body>
17   <label for="inputtext">1. 入力欄にデータを入力する:</label><br>
18   <textarea id="in" name="in" cols="60" rows="4"
19     placeholder="入力する文字列をここに書き込む"></textarea><br>
20   ↓<br>
21   <label for="go">2. 「実行」ボタンを押す:</label>
22   <button type="button" id="go" name="go" onclick="do_action()"
23     <font color="blue" size="+2">実行!</font></button><br>
24   ↓<br>
25   <label for="outputtext">3. 出力欄に処理結果が出力される:</label><br>
26   <textarea id="out" name="out" cols="60" rows="4"
27     placeholder="出力文字列が書き込まれる"></textarea><br>
28 </body>
29 </html>

```

図 3 タイプ B : GUI 型プログラム (HTML ソースコードの一部として実装され、ブラウザ上で実行される)

Fig. 3 Type-B: GUI-type program, which is implemented as a part of HTML source code and runs on a Web-browser



図 4 GUI 型プログラムの実行

Fig. 4 Running the GUI-type program

せた文字列を出力する」という、ごく単純なプログラムを考える。

図 1 は、コマンドとして実装される CUI 型 (コマンド型) のプログラム例である。JavaScript で書かれており、実行環境としては、OS X 上の JavaScriptCore\*2 を利用した。

プログラムのファイル名を「test.js」として保存し、さらに実行属性が与えられていることを前提として図 2 のように実行することができる。便宜上、こちらのプログラムをタイプ A と呼ぶことにする。

他方、図 3 は、同じ処理を HTML に埋め込んで実装したものである。本来、JavaScript は、ウェブブラウザで実行するために作られた処理系であり、位置づけとしてはこ

\*2 OS X に付属する JavaScript インタプリタ「jsc」をコマンドラインから起動する。

ちらが素直なものであるとも考えられる．ここでは，こちらのタイプのプログラムを，GUI 型（ブラウザ型），タイプ B と呼ぶことにする．

タイプ B のプログラムを実行した状況を，図 4 に示す．

タイプ A およびタイプ B のいずれにしても，プログラムの本質的な部分は「.split("").reverse().join("")」で表される，「文字列を配列に分割したのちに反転させ，再び文字列化する」という 1 行だけである．しかし，両者のプログラムを比較すると，タイプ A，CUI 型のプログラムはより抽象的でありシンプルな印象を与える．対してタイプ B では GUI 部品を用意するための HTML コードが長々と記述されており，タイプ A と比較してやや煩雑な印象は否めない．

タイプ A とタイプ B ではいずれも処理内容は同じであるが，ごく簡単な情報システムのモデルとして考えた場合，入力と出力をどう表現しているかが大きく違っている．タイプ A，コマンド型のプログラムでは，入出力の部分が「標準入出力」に抽象化されていて，理解している者にとってはシンプルでよいが，とっつきにくいという性質を持つ．一方のタイプ B，ブラウザ型のプログラムでは，入力と出力が，それぞれ HTML 中にテキストエリア (textarea) として表現されていて具体的で分かりやすいという性質を持つ．しかし，そのためプログラムの記述としては，HTML 部分も含め，前述のとおり煩雑な表現となっている．

はたして，プログラミング初心者にとっては，どちらのタイプが，より理解しやすいものであろうか．

## 2.2 調査方法

このような問題設定に対して，質問紙調査で意見を求めることとした．意見を求めた対象は，「1. IT 産業の現場で業務に従事している技術者もしくは IT 関連の教育に従事している教育者を中心とする専門家」と，「2. プログラミング演習を受講する立場である学生」という 2 つの集団である．

### 2.2.1 専門家を対象とした調査

専門家向けの質問紙では，以下の 3 問に対する回答を求めた．

- (1) 初学者向け教育には，どちらのプログラムが優れていると考えるか？
- (2) IT に関する自分の熟練度はどのくらいか？
- (3) プログラミング教育に関する意見（任意回答）

なお，(1) ではタイプ A とタイプ B のどちらが望ましいかについて「タイプ A が望ましい (1)」から「タイプ B が望ましい (10)」までの 10 段階での評価を求め，(2) では熟練度のレベルをやはり 10 段階（「ほぼ初心者 (1)」から「経験豊富 (10)」まで）での評価を求めた．最後のプログラミング教育に関する意見は自由記述形式で，かつ，任意回答とした．



図 5 アンケートフォームのスクリーンショット

Fig. 5 A screenshot of the questionnaire

**プログラミング教育に関するアンケート**

質問 1：あなたは，CUI プログラム（コマンド型：タイプ A）と GUI プログラム（ブラウザ型：タイプ B）のどちらが分かりやすいと考えますか？

コマンド型 (タイプ A) が分かりやすい | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | ブラウザ型 (タイプ B) が分かりやすい

質問 2：初心者は，CUI プログラム（コマンド型：タイプ A）と GUI プログラム（ブラウザ型：タイプ B）のどちらを最初に学ぶべきと考えますか？

コマンド型 (タイプ A) を先に学ぶべき | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | ブラウザ型 (タイプ B) を先に学ぶべき

質問 3：あなたは，提示したプログラムの構造や実行の様子をどれだけ理解できていますか？

ほとんど理解できていない | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | ほぼ理解できている

履修している講義に○を付けてください  
「ネットワーク技術 (1) (2)」 / 「情報システム設計」「情報システム開発」

図 6 学生に配布したアンケートフォーム

Fig. 6 A sheet of the questionnaire which delivered to the students

これらの質問と回答用紙は Google ドキュメントのフォームとして作成 (図 5) し，SNS の繋がりによる人的ネットワークを活用して回答を依頼<sup>\*3</sup>した．なお，質問への回答を求めるにあたり，専門家であるか否かの条件を制限として課すことはしなかったが，後述の結果を見る限り回答者の IT リテラシレベルは概ね高く，示唆的な意見を収集することができた．

### 2.2.2 学生を対象とした調査

受講生側として，学生がどう考えるかについても同様に質問紙を用意して意識調査を実施した．学生側に提示した主たる質問は，次の 3 項目である．

- (1) CUI プログラムと GUI プログラムのどちらが分かりやすいと考えるか？

<sup>\*3</sup> 問題設定やプログラムの説明に関しては，Facebook のノート機能を使用し，そこから Google ドキュメントのフォームへハイパーリンクで誘導することによって回答記入を促した．

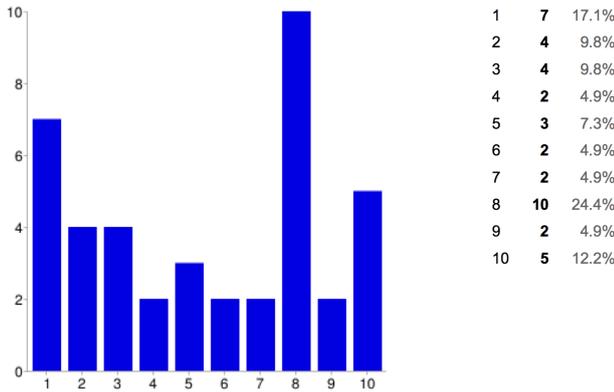


図 7 CUI と GUI のどちらが優れているか (専門家)

Fig. 7 Which is suitable: CUI vs GUI? (professionals)

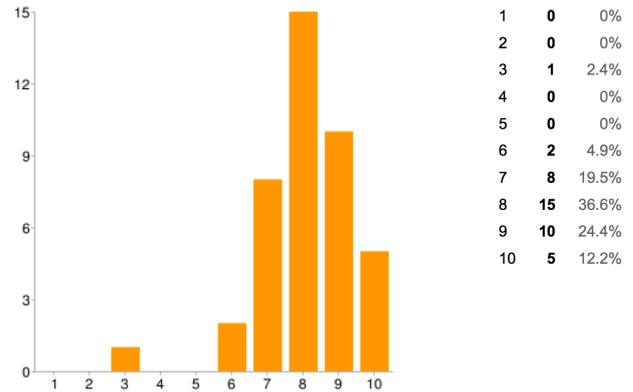


図 8 IT リテラシのレベル (専門家)

Fig. 8 Level of IT-literacy (professionals)

- (2) CUI プログラムと GUI プログラムのどちらを先に学ばせたいか?
- (3) 提示したプログラムの構造や実行の様子をどれだけ理解できているか?

なお、上記以外に受講している講義についても質問したが、この質問は回答の重複を避けるためにのみ用いており本質的ではないため省略する。それぞれの回答は、専門家向けのものと同様に 10 段階での評価とした。(1) および (2) については「CUI(1)」から「GUI(10)」まで、10 段階の相対評価である。また、最後の質問は「ほとんど理解できていない(1)」から「ほぼ理解できている(10)」までの 10 段階評価である。

学生向け調査はオフラインで実施した。すなわち、プログラムの解説まで含めて記載したアンケート用紙(図 6)を作成し、複数のプログラミング関連講義<sup>\*4</sup>を受講している学生を対象に、質問への回答を求めた。

### 3. 調査結果と考察

前述の方法で収集した意識の結果と分析について、本節で述べる。

#### 3.1 専門家の意識

ネット上での問題提起と回答の依頼は 2014 年 12 月 6 日に開始し、12 月 15 日までの 10 日間で 41 件の回答が寄せられた。

特筆すべきは、任意回答としていた「プログラミング教育に関する意見」に 36 件<sup>\*5</sup>もの意見が寄せられた<sup>\*6</sup>点である。各意見は平均して 158 文字の長さであった。

<sup>\*4</sup> 対象としたプログラミング関連講義は 2 年生以上が受講する専門科目であり、その中には必修科目も含まれる。

<sup>\*5</sup> 無回答 4 件、無効(ノイズと思われる意味不明な文字) 1 件であった。

<sup>\*6</sup> さらに、回答を依頼した SNS での記事に対しても 31 件のコメントが付いており、そのなかでも建設的かつ有効な意見がいくつか寄せられた。

#### 3.1.1 回答者の立場

図 7 は、初学者を対象としたプログラミング演習環境としてタイプ A の CUI 型がよいかタイプ B の GUI 型がよいかを 10 段階で評価した結果である。専門家の意見としては、CUI がよいとする意見(1~3)と、GUI がよいとする意見(8~10)の両極端に分かれた。

また、図 8 は、回答者自身の IT リテラシレベルを問うたものである。レベル 8 を中心に分布しており、インターネットを介して匿名<sup>\*7</sup>で寄せられた回答といえども今回集められた回答者の意見は専門家の意見としてほぼ信頼できることがわかる。

#### 3.1.2 寄せられた意見(主なもの)

任意回答で寄せられた意見には、いくつかのパターンが見出された。以下、簡単にまとめる。

まず、CUI が適しているか GUI が適しているかは「教育の目的による」という指摘である。ゴールとして高い能力を持つ情報処理技術者を育成したいのか、教養として最小限のプログラミング能力を身につけさせたいのかによって立場は異なるであろうという意見が寄せられた。前者を目指すのであればコンピュータの動作原理、基礎的な知識からしっかりと叩き込むために CUI をベースに教育すべきであるという意見もあった。

また、具体的なプログラミング環境を推薦する意見も多く寄せられた。Processing<sup>\*8</sup>を用いて見た目にはわかりやすいものがよいのではないかと、GUI を突き詰めてビジュアルプログラミングを利用すべき、思い切って MIT Scratch[1] のレベルまで落としてみたらどうか、Arduino<sup>\*9</sup>や Raspberry Pi<sup>\*10</sup>のような目に見えてわかるものがよい、などの意見がある。

<sup>\*7</sup> 最初に SNS で回答を募ったという経緯から、人的ネットワーク上に存在する回答者であることは容易に想像でき、また任意回答の内容から本人を推測できるケースもある。そのため、完全な匿名ではなく、いわば半匿名の回答者であるといえる。

<sup>\*8</sup> <https://processing.org>

<sup>\*9</sup> <http://www.arduino.cc>

<sup>\*10</sup> <https://www.raspberrypi.org>

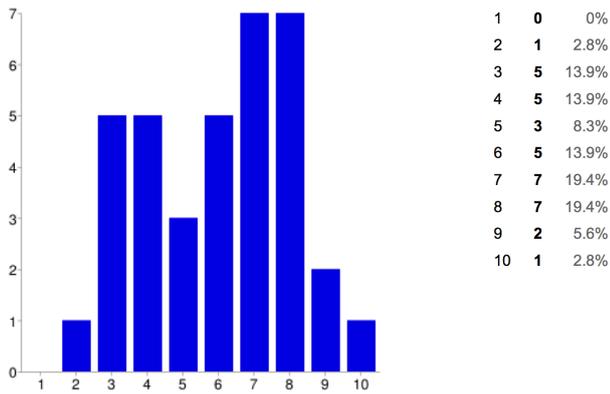


図 9 CUI と GUI のどちらが分かりやすいか (学生)

Fig. 9 Which is easy to understand: CUI vs GUI? (students)

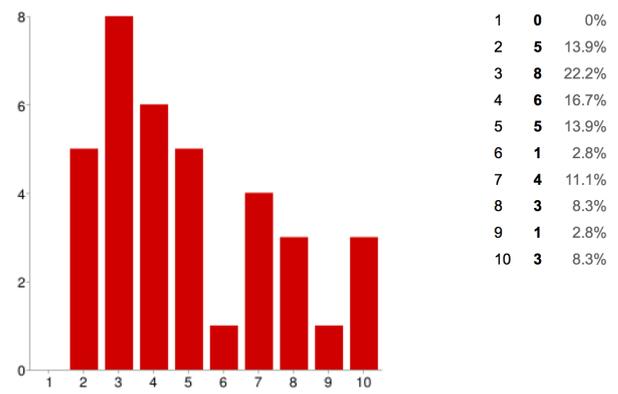


図 11 問題に関する理解度 (学生)

Fig. 11 Level of understanding (students)

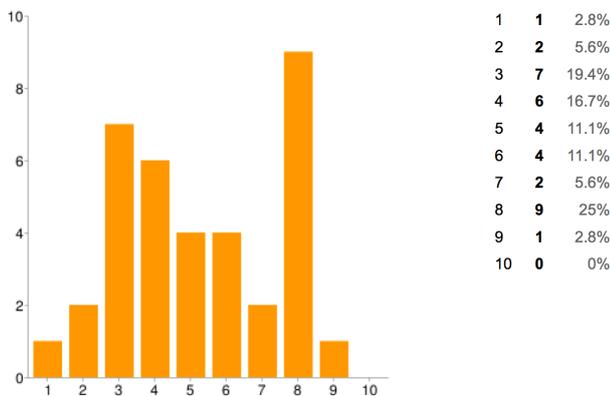


図 10 CUI と GUI のどちらを先に学ぶべきか (学生)

Fig. 10 Which does beginner study first: CUI vs GUI? (students)

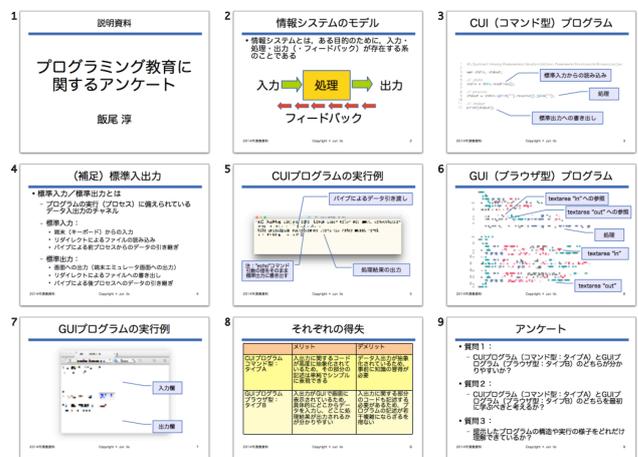


図 12 問題設定の説明資料

Fig. 12 Materials for illustration on the problem

さらに、CUI を利用しようが GUI を利用しようが、とにかく学生の興味を惹くことが重要であり、何を対象としてどのようなアルゴリズムで処理するかを意識させることが大切であるという意見も、複数、寄せられた。今回提示した例でもまだ複雑すぎる、パズルを解くタイプのものがよい、結果がすぐにフィードバックされるもの、ゲームがよい、といった意見が目立った。

なお、とにかくたくさん経験させるために CUI も GUI も両方経験させる、という意見と、初学者の混乱を避けるために CUI なら CUI、GUI なら GUI と割り切って教え方をデザインし、CUI と GUI を行き来しないようにしたほうがよい、という相反する意見の対立<sup>\*11</sup>もあった。

### 3.2 学生の意識

続いて、受講する立場である学生の意識を示す。学生からの回答は、同じく 2014 年 12 月上旬に実施されたプログラミング関連の講義時間の最後<sup>\*12</sup>を用いて集めた。有効回

\*11 それぞれの意見は独立して寄せられたものであり、どちらかの意見に対して投げかけられた意見というわけではない。回答者の立場によって正反対の意見が寄せられた例として興味深い。

\*12 図 12 はその際に使用した説明資料である。

答は 36 件、対象となった学生はほぼ 2 年生が中心であり、若干名の 3, 4 年生が混じっている。

図 9 および図 10 は、学生による意識の分布である。専門家の意識では両極端に分かれていたが、一転してどっちつかずの回答分布となっている。どちらを先に学ぶべきかという質問について、わずかに 2 つのピークが見られる程度である。

なお、どちらが分かりやすいかという質問に対する回答と、どちらを先に学ぶべきかという質問に対する回答、それぞれの分布に関し、相関計数は 0.356 ( $p$  値 = 0.033) とそれほど高い相関は見られなかった。

図 11 は、この問題に対する理解度の分布である。残念なことに、あまり理解できていないという結果が多いという状況になった。質問紙の配布と記入、回収に際して、十分に時間をとり問題の説明を実施したにもかかわらず、多くの回答者が理解できていないと考えた。図 8 で示された専門家による意識の高さと比べると、対照的な結果となっている。

### 3.3 考察

教育のゴールが不明確であったなど、問題設定が若干甘かったこともあり、専門家の意識としては、初学者に CUI 型のプログラミング環境を薦める立場と GUI 型のそれを薦める立場の両極端に分かれた。したがって本調査の範囲では、残念ながら結論は得られなかった。しかし、自由記述によるコメントとして、本問題に対する示唆に富む意見を集約できた点は非常に有意義であった。

一方、学生の意識調査は、そもそも問題を十分に理解できていない回答者が多く信頼できない結果となった。問題を適切に理解できていないため確信を持って判断することができず、その結果としてどちらでもない(4~7)という回答に集中したと考えられる。この点は反省点として捉え、次回、同様の調査を実施する際にはより多くの時間をかけて説明したうえで回答を求めようとするべきであろう。

また、CUI プログラムと GUI プログラムの違いをより明確にするために、今回のような文字列反転処理という共通課題ではなく、CUI プログラムは文字列操作とする一方で GUI プログラムはグラフィカルな処理を行うものとする。プログラムの位置付けを明確に分離することも検討すべきかもしれない。ただし、そのような違いを加える際には、学習に対する有効性の判断に対して、対象とする処理の面白さが影響してしまう可能性があり、課題設定には十分な配慮を行うべきである。

## 4. 関連研究

ユーザインタフェース評価の歴史において、CUI と GUI の比較は古くからなされてきた。たとえば、既に 90 年代において Gaff[2] は図書館データベースにおいて CUI と GUI の比較を実施して GUI の優位性を唱えており、Rauterberg はビギナーとエキスパートを対象として CUI と GUI の操作比較を実施している。また、近年でもグラフィカルなソフトウェアを対象として CUI と GUI の比較を Feizi ら [4] が検証しており、Fellmann ら [5] は VR のシステムに対して同様の検証を実施した。

このように、個別のアプリケーションに関して CUI と GUI のどちらが適しているかといった検証はしばしば行われてきたが、初学者がプログラミングを学ぶ際に CUI と GUI のどちらを薦めるべきであるかとの議論はあまり行われてきていない。なお、Pears ら [6] は初学者向けプログラミング教育に関して綿密なサーベイの結果を報告しているが、そのなかで、プログラミング環境に関する議論がごくわずかではあるが言及 [7] されている。

神谷 [8] は GUI 環境の考慮はプログラミングの学習には余計なものであるとし、CUI 環境によるプログラミング教育を論じている。ここで、単なる CUI プログラムだけではなく、外部にランプ点滅の制御回路を接続し、実感を伴うプログラミング演習を提供している点は、Raspberry Pi や

Arduino を使うとよいのではないかと本研究で集積された専門家の意見に通じるものがあり、興味深い。

## 5. まとめと今後の課題

初学者向けプログラミング演習の環境として、CUI 型のプログラミングを教えるべきか、あるいは GUI 型のプログラミングを教えるべきかとの問いに対して、具体的かつごく簡単なプログラム事例を示したうえで専門家および学生に対して質問紙による意識調査を実施した。

専門家の意見は CUI 派と GUI 派に二分され、また、前提条件や教育の最終目的によるなどの指摘もあったため、どちらが適しているかという問題に対する明確な解は得ることができなかったが、自由回答による示唆に富む意見を集めることができた。また、学生を対象にした調査では、そもそも十分に問題を理解できなかった回答者が多かったとの不備があり信頼性の高い回答を集めることができなかった。また、問題設定にも再考の余地があることも今後の検討材料である。

今回の調査では有効回答数もそれほど多く集めることができなかったため、より明確な条件設定を行ったうえで、さらに多くの回答者から意見を集めることが、今後の課題として残されている。

謝辞 本研究における質問紙調査に協力して下さった全ての回答者および査読時に有用な指摘をくださった査読者の方々に感謝します。

## 参考文献

- [1] Resnick, Mitchel, *et al.*, "Scratch: programming for all." *Communications of the ACM* 52.11 (2009): 60-67.
- [2] Gaff, Margaret, "GUI vs CUI: Individual personality types and the experience of learning to use library databases," *The Australian Library Journal*, 43.3 (1994): 183-191.
- [3] Rauterberg, Matthias, "An empirical comparison of menu-selection (CUI) and desktop (GUI) computer programs carried out by beginners and experts," *Behaviour & Information Technology*, 11.4 (1992): 227-236
- [4] Feizi, Andisheh, and Chui Yin Wong. "Usability of user interface styles for learning a graphical software application." *Computer & Information Science (ICCIS)*, 2012 International Conference on. Vol. 2. IEEE, 2012.
- [5] Fellmann, Tom, and Manolya Kavakli. "A command line interface versus a graphical user interface in coding VR systems." *Proceedings of the Second IASTED International Conference on Human Computer Interaction*. ACTA Press, 2007.
- [6] Pears, Arnold, *et al.* "A survey of literature on the teaching of introductory programming." *ACM SIGCSE Bulletin* 39.4 (2007): 204-223.
- [7] Gross, Paul and Powers, Kris. "Evaluating assessments of novice programming environments." In *Proceedings of the first international workshop on Computing education research (ICER '05)*. ACM, New York, NY, USA, 99-110.
- [8] 神谷達夫, "CUI 環境でのプログラミング教育," *京都創成大学紀要*, 6. (2006): 77-86.