

プログラミング学習における GUI ベースから CUI ベースへのスムーズな移行教育 Smooth Transition from GUI-based to CUI-based Education in Programming Learning

恐神 正博[†] 大熊 一正^{††} クリストファー ピロット[‡]
Masahiro Osogami Kazumasa Ohkuma Christopher Piroto

1. はじめに

慢性的な ICT 技術者不足に対し、小・中・高校での学校教育において、プログラミングの必修化が行われている。一方、組込み系をはじめとする ICT 技術者の育成においては、継続的なプログラミング学習の必要性も叫ばれ[1]、その中で GUI の環境を用いたプログラミングの導入教育から、CUI 環境を用いた継続的なプログラミング学習へのスムーズな移行教育が模索されている[2]。本研究では、GUI ベースの導入教育から CUI ベースでのプログラミング学習へのスムーズな移行教育法について、大学 1 年生に対し 4 年間にわたり調査した結果の報告を行う。その結果、事前に GUI によるプログラミングの経験があることによって、CUI でのプログラミングを学習する際、特に変換仕様、演算子、条件分岐、ネストなどの概念について、学生の理解度が高まることが分かった。さらに、GUI ベースから、CUI ベースのプログラミングへの移行方法として、3 種類の方法を試行し、GUI ベースの環境のみで一通り学習した後に CUI ベースの環境に移行する方法が最適であり、GUI ベースと CUI ベースの環境を同時に利用しつつ移行教育を行う場合は、プログラミング概念の理解度の低下につながる事が分かった。

2. GUI ベースおよび CUI ベースにおける学習

GUI ベースから CUI ベースへの学習へとスムーズに移行させるために、それぞれの授業コンテンツを GUI ベースと CUI ベースとで対応させた。ここでは、GUI ベースおよび CUI ベースのそれぞれの学習環境と、対応させた各授業コンテンツについて説明を行う。

2.1 学習環境

GUI ベースでの学習環境には、プログラミングの導入教育において世界中で広く利用されている Scratch[3]を用いた。この理由としては、フリーウェアとして公開されており誰でも無料で利用できる点や、機能ごとのブロックの組み合わせで、プログラムを簡単に作ることができ、またそのプログラムの構造を視覚的に理解しやすい点があげられる。このため初学者にとって 1 つの難関となるプログラミング言語の文法習得の段階を踏むことなく、文法習得よりも重要とされるアルゴリズムの直感的理解を促せる。一方で、CUI ベースの学習環境としては、組込み系で広く用いられている C 言語を用いている。コンパイラには Microsoft 社の C コンパイラを、エディタにはフリーウェアであるサクラエディタ[4]を用いた。C コンパイラのみ市販のものを使った理由として、福井工業大学ではすべての入学生にノート PC を準備させており、その設定の中に Microsoft 社の C コンパイラも含まれているためである。

[†] 福井工業大学 経営情報学科
^{††} 岡山理科大学 教育推進機構
[‡] 福井工業大学 基盤教育機構

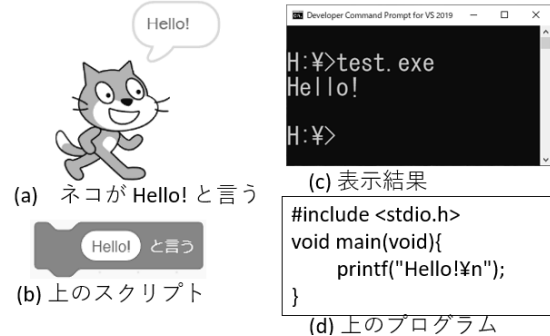


図 1 GUI ベースと CUI ベースでの対比の例

2.2 各授業におけるコンテンツ

各授業としては、GUI と CUI とで内容が対応するように、それぞれ、「1.表示」、「2 変換仕様」、「3 変数」、「4 キーボードからの入力」、「5 演算子」、「6 条件分岐 (if)」、「7 条件分岐 (if-else)」、「8 反復」、「9 ネスト」、の順になるように行った。図 1 には「1 表示」における GUI および CUI の例を示している。

3. 調査結果

調査は、本学の大学 1 年生に対し、筆者が担当する 1 年時の必修科目であるプログラミングの授業において行った。初年度 (2016 年) は、GUI ベースの事前学習の効果を確認 (比較) するため、CUI ベースの内容のみを行った。2 年目 (2017 年) は、前期に GUI ベースの授業、後期に CUI ベースの授業を実施した (これは、GUI を使った授業を一通り終えてから、CUI を使った授業へ切り替えた場合となる)。3 年目 (2018 年) は、GUI ベースの授業と CUI ベースの授業を毎週交互に対応した内容で実施した。最後の 4 年目 (2019 年) は、毎回の授業を、前半と後半の 2 つに分けて実施し、その前半で GUI ベースの授業を、後半で前半に対応する CUI を使った授業を実施した。また、CUI ベースでの授業の際、理解度を 4 から 1 までの 4 段階の整数値で回答させた。

表 1 に、初年度に実施した事前の GUI ベースでの学習を行わず、最初から CUI ベースでの学習を行った場合の各授業における理解度の結果を示す。なお、表 1-4 の番号は、2.2 節で説明した各授業の内容 (コンテンツ) の番号と対応させている。ここで、各個人ごとに、すべての回の理解度の平均値と各回ごとの理解度の値について、差の検定を行った。なお、これらの結果については、正規性検定から、データが正規分布であることが棄却されたため、マン・ホイットニーの U-検定の結果を示している。また、有意水準 5% で有意差が得られた値について左上に * を付している。

表 1 において、* を付した有意差が得られた値を見ると、番号 1, 3, 4, 6, 8 は全体の平均がプラスに、番号 2, 5, 7, 9 ではマイナスにそれぞれ変化しており、差がマイナスとなった内容 (2: 変換仕様, 5: 演算子, 7 条件分岐 (if-else), 9: ネスト) に

表1 1年目(2016年)におけるCUIベースでの各授業での理解度(Cのみ)

番号	1	2	3	4	5	6	7	8	9
理解度の全体の平均値との差 (全体の平均値: 3.079)	*0.285	*-0.487	*0.143	*0.136	*-0.283	*0.263	*-0.180	*0.234	*-0.404
標本数(学生数)	35	35	39	34	37	37	35	31	32

*: $p < 0.05$

表2 2年目(2017年)におけるCUIベースでの各授業での理解度(半期ごと、ScratchとC)

番号	1	2	3	4	5	6	7	8	9
理解度の全体の平均値との差 (全体の平均値: 3.406)	*0.039	*0.142	*-0.067	*-0.231	*0.035	*-0.162	*0.084	*-0.007	*0.104
標本数(学生数)	28	24	12	22	25	22	29	30	32

*: $p < 0.05$

表3 3年目(2018年)におけるCUIベースでの各授業での理解度(隔週で、ScratchとC)

番号	1	2	3	4	5	6	7	8	9
理解度の全体の平均値との差 (全体の平均値: 3.208)	*-0.013	*-0.139	N/A	N/A	*0.098	*0.106	*0.064	*-0.101	*0.292
標本数(学生数)	21	25	0	0	27	28	26	23	28

*: $p < 0.05$

表4 4年目(2019年)におけるCUIベースでの各授業での理解度(毎時、ScratchとC)

番号	1	2	3	4	5	6	7	8	9
理解度の全体の平均値との差 (全体の平均値: 3.436)	*-0.058	*0.095	*0.127	N/A	N/A	*0.147	N/A	*-0.280	N/A
標本数(学生数)	26	25	25	0	0	22	0	19	0

*: $p < 0.05$

ついて、CUI ベースでのプログラミング学習における躓きやすいポイントと推察されることが分かった。

続いて、最初から CUI ベースで学習を行うのではなく、GUI ベースでの学習を経て、プログラムの考え方等がある程理解してから CUI ベースでの学習を行った場合に理解度がどのように変わるかについて、2 年目から 4 年目にかけて調査を行った。その結果が表 2-4 に示したものである。

表 2-4 を見ると、表 1 から得た躓きやすいポイントである番号の 2 に注目すると、表 2 および表 4 において、負の値から正の値に転じている。これは事前の GUI ベースでの学習がその後の CUI ベースでの学習に対し、効果があったことを示している。表 3 の場合は負の値のままであったが、2/3 の割合で、事前の GUI ベースの学習効果があったことが確認できる。また、番号 5,7,9 の場合には、表 2 および表 3 において負の値から正の値に転じている。なお、表 2-4 における N/A の表記については、その回のアンケート調査が出来なかったことにより、値が得られなかったことを示している。これらのことから、躓きやすいポイントである番号 5,7,9 においても事前の GUI ベースでの学習の効果が確認できた。一方で、番号 8 (反復) については、表 1 での値は正の値であったにもかかわらず、表 2-4 では負の値に転じている。これは、CUI ベースのみでの学習の方が、事前に GUI ベースでの学習を行うよりも理解度が上がることを示している。

これらの結果から、プログラミング学習において、事前の GUI ベースでの学習を行った場合には、特に変換仕様、演算子、条件分岐(if-else)、ネストについての理解度が向上することが分かった。ただし、すべての場合に理解度が向上するわけではなく、一部(例えば、反復処理など)については、事前の GUI ベースでの学習を行わず、いきなり

CUI ベースでの学習を行った方がいい場合もあることが分かった。

4. おわりに

慢性的な ICT 技術者不足に対する ICT 技術者の育成において、継続的なプログラミング学習の必要性が叫ばれており、そのための、プログラミング教育における GUI ベースから CUI ベースへのスムーズな移行教育法について考察した。

その結果、CUI ベースでのプログラミングを行う前に、GUI ベースでのプログラミング教育を行うことで、変換仕様、演算子、ネストなど、特定の内容について、理解度を高めることができることが確かめられた。一方で、反復処理など CUI ベースでのみでの学習を行った方が、理解度が向上する場合があります、すべての授業内容について効果が上がるわけではないことも明らかとなった。

今後については、アルゴリズムの定量的な理解度における GUI ベース、CUI ベースの両方の比較検証を行うなど、更なる GUI ベースから CUI ベースへの効果的な移行教育について研究を進めていきたい。

謝辞

本研究の一部は科研費(24501221, 16K01138, 22K02926)の助成を受けたものである。

参考文献

- [1] 内閣府, 平成 25 年度年次経済財政報告, 第 3 章 第 1 節 2013.7.
- [2] 政府 CIO ポータル, 世界最先端 IT 国家創造宣言, 2013.6.
- [3] MIT Media Lab, "Scratch 公式サイト," <http://scratch.mit.edu/> (参照日 2023 年 5 月 20 日).
- [4] サクラエディタ公式サイト, <https://sakura-editor.github.io/> (参照日 2023 年 5 月 20 日).