

ビジュアルプログラミングツールを用いたプログラミング導入教育の試み A Trial of Introductory Computer Programming Education using Visual Programming Tool

松村 寿枝[†] 内田 眞司[†] 西野 貴之[†] 清水忠昭[‡]
Toshie Matsumura[†] Shinji Uchida[†] Takayuki Nishino[†] Tadaaki Shimizu[‡]

1. はじめに

現在、使用する言語の違いはあるが高専の情報系学科では、プログラミング教育が行われている。奈良高専情報工学科でも 2~4 年生でプログラミング教育を行っている。学生はプログラミング教育に対してどのような意識を持っているのか本学学生の授業アンケートの結果を見ると、プログラミングの意義は理解しているが、プログラミングに苦手意識を持つ学生が一定数いるのも事実である。これは他の高専や大学等でも同様である^{[1][2]}。

一般的にプログラミング教育は、プログラミング言語の文法を教え、種々の例題を解いた後で、いくつかの応用問題を解く形式である場合が多い。これに対して本校では、複数教員によるプログラミング教育、習熟度を利用した教育などいろいろな取り組みをこれまでにやってきた^{[3][4]}。しかし、苦手意識を持つ学生は一定数存在するままである。そこで本研究では学生の苦手意識を改善し、プログラミングの理解を深めることを目的とし、プログラミング教育の導入にビジュアルプログラミングを用いた講義を行う。これまでにビジュアルプログラミングを用いたプログラミング教育としては、Squeak や Scratch^{[5][6]}などを使用した研究が報告されている^{[7][8]}。Squeak や Scratch は子供を主な対象としているため、変数を数値型・文字型と限定することで変数を気にすることなく、マウスの操作のみでプログラムを行うことができる^{[10][11]}。そのため、Java や C のようなテキスト記述型の言語で生じる文法的なエラーが生じにくく、初学者にとって扱いやすいプログラミング方法であり、入門段階では成功を取めている^{[7][8]}。また、松澤らの先行研究では、ビジュアルプログラミングとテキスト記述型の言語併用による講義では、苦手意識を持つ学生ほどビジュアル言語の選択率が高かったことが報告されている^[9]。また、杉浦らは、プログラミングで実践すべき教育内容は、「課題は簡単であっても、学習者が自らアルゴリズムを組み立てることを実践し、それを実行することが可能なプログラミング言語で記述し、その正しさを検証すること」としている^[7]。筆者らはこの考え方を取り入れ、ビジュアルプログラミングを本校のプログラミング I の講義の導入教育に用いることとした。ビジュアルプログラミングを導入するにあたって、従来の講義形態はそのままにし、アルゴリズムを考える際に使用するフローチャートを書くことと同じようにプログラミングできるもの、なるべくビジュアルプログラミングからテキスト記述型言語へのシームレスな移行が可能なツールを検討した。特にプログラミング初学者は、アルゴリズムの分析や詳細化の段階を無視し、フ

ローチャートを書くよう指示があっても十分にアルゴリズムを詳細化しないままプログラミングを始める傾向がみられるとの報告もある^[7]。そこで本研究では、フローチャートを書くことを重視し、フローチャートを書くのと同様な操作でプログラミングできるツールとしてヴィストン株式会社製の「加速度センサプログラマ」と専用ソフトウェアの「ビュートビルダーG」を用いる^[12]。プログラミング I の導入として、上記のツールを使った講義を行い、学生のアンケート結果をまとめ、その効果を検証する。

2. 講義概要

本校では、プログラミング I を 2 年生で、プログラミング II を 3 年生で、プログラミング III を 4 年生と 3 年間プログラミングの講義を行っている。各学年は、例年 40 名前後で、いずれも必修科目である。プログラミング I ~ III は 45 分の講義を 2 コマ、年間 30 週の講義である。使用言語は、プログラミング I, II では Java, プログラミング III では C 言語である。いずれの講義も 2 名の教員が担当し、1 人は、主担当として、講義資料の作成、演習・課題の作成、講義の進行、演習中の質疑応答を行っている。もう 1 名の教員はサポート担当として、演習中の学生からの質疑応答に対応している。プログラミング I, II では、上記の主担当、サポート担当は 15 週で交代する。プログラミング III では主担当、サポート担当は 30 週で固定している。また、カリキュラムは、毎年同じであり、講義内容および使用する教科書は同じであるが、教員は年度ごとに異なるため、教員が作成する配布資料や演習・課題や試験問題などは年度によって異なっている。定期試験については、2 名で相談をしたうえで、問題を作成している。配点については 2 名とも同じ点数とし、それらを合計し、課題の提出状況等を勘案して、シラバスに記載通りに成績評価としている。

2.1 講義詳細

プログラミング I は通年科目で、前期の 15 週でプログラムの書き方、入出力、変数、演算子、順次、条件分岐、反復を学ぶ。後期の 15 週でデバッグの方法、データ型、配列、文字列処理を学ぶ。講義は、主担当が作成した授業資料を用いて文法的事項を説明、その後、演習・課題の問題を説明し、残りの時間に演習・課題を行う形である。ビジュアルプログラミングを実施した年は、当日提出する演習と締め切りを 1 週間とする課題を設定した。演習は、当日中に提出できる簡単なものとし、その日習った文法的事項の定着を図るための比較的簡単な問題である。課題は、締め切りが長く、1 週間かけて解くことができるので、演習よりも熟考が必要な発展問題としている。

[†] 奈良工業高等専門学校 National Institute of Technology, Nara college

[‡] 鳥取大学大学院 Tottori University graduate school

前期には、前期末試験、後期には後期中間試験および学年末試験を 80 分で実施している。この試験では、与えられた問題(仕様)に対してプログラムを作成してそのソースコードを提出するものを中心としている。

この前期の 15 週中の 3 週分を用いて、プログラムの構造である「順次」、「条件分岐」、「反復」を学習するのにビジュアルプログラミングを使用した。また、講義の際には文法的事項をテキストで説明するだけでなく、フローチャートも使用して、処理の流れやプログラムの構造の理解を促すような講義資料を作成した。図 1 に講義資料の一部を示す。順次処理についての説明と実際のプログラム例を示し、説明の後に「ビュートビルダーG」を用い、課題を解いてソースコードを提出する。使用したビジュアルプログラミングのツールの詳細は 2.2 で述べる。3 週の講義の後には、順次、条件分岐、反復について Java を用いた演習・課題を実施した。

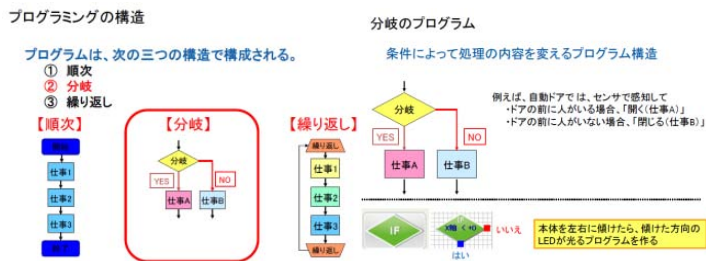


図 1 説明資料の例

2.2 ビジュアルプログラミング

ビジュアルプログラミングは本校のプログラミングの講義で使用している Java, C 言語のようなテキスト記述型の言語でなく、視覚的な操作でプログラミングを行うマウス操作でプログラミングを行うものである。いろいろな種類のビジュアルプログラミング言語がつかわれているが、本研究では、ビジュアルプログラミングのツールとしてヴィストン株式会社製のプログラミング教材「加速度センサプログラマ」と付属の専用ソフトウェアの「ビュートビルダーG」を使用した。図 2 にプログラム作成例(左図)を示す。フローチャートを書くのと同様にそれぞれの命令ブロックをマウス操作によりつなぎ合わせることでプログラミングを行うことができる。出来たプログラムは、USB ケーブルを接続、加速度センサ(右図)に転送し、動かすことができる。加速度センサは、昨今スマートフォンやゲーム機などに組み込まれており、学生にもなじみのあるものである。また、自分が作成したプログラムの通りに加速度センサの反応・動作することを体験することで学生のモチベーションをあげることを期待している。

フローチャートについては、学生はプログラミング I の講義以外の専門科目でフローチャートを学んでおり、簡単なフローチャートなら書くことができる。そのため、フローチャートを書くのと同じようにプログラミングをすることができるものを採用した。

3. 結果

3.1.1 アンケート結果

前期の講義が終わってから、学生にアンケートを実施した。アンケート項目は、5 段階評価とした。アンケート項目は、以下のとおりである。

1. 加速度センサプログラマで使用したビジュアルプログラミングはわかりやすいとおもいましたか?
2. ビジュアルプログラミングのソフトウェアは使いやすかったですか?
3. プログラムの構造(順次、分岐、繰り返し)についてこれを使うことで理解できましたか?
4. 特に順次、分岐、繰り返しのどの部分についてこれをつかうことで理解が進んだと思いますか?(複数回答可)
5. 加速度センサとビジュアルプログラミングを今後も使った方がいいと思いますか?
6. 授業の時間についてはどうでしたか?
7. プログラミングの学習に役に立ったと思いますか?
8. 感想があれば、自由に聞かせてください。

アンケート項目のうち 1, 3, 4 の結果を表 1~3 に示す。表において「全体」は、欠席の 3 名をのぞいた 40 名全員の集計結果である。また、前期末試験の試験点数から上位 25%、下位 25%の結果を「上位 25%」、「下位 25%」としてまとめている。なお、講義の受講者は 43 名(アンケート回答は 40 名)で、上位 25%は 11 名、下位 25%も 11 名であるが、受講者 43 名中成績順に集計すると下位 25%の中に欠席していた 2 名が入るため、下位 25%のみ 9 名での集計結果となっている。

表 1 の加速度センサを用いたビジュアルプログラミングのわかりやすさについては、「わかりやすい」と「まあわかりやすい」をあわせると全体で約 75%となり大部分の学生が「わかりやすい」としている。これを成績別の結果で比較すると下位の学生の 88.9%のほうが上位の学生の 72.7%よりも「わかりやすい」、「まあわかりやすい」と評価している。また、上位者は「ふつう」と評価したものも多く、下位者のほうが加速度センサプログラミングに対する評価が高かったと考えられる。

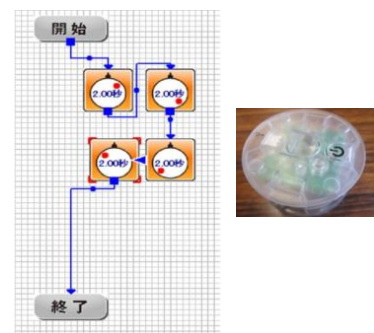


図 2 プログラムの例(順次:左)と加速度センサ(右)

表 2 に理解度(項目 3)の結果を示す。「できた」と「まあできた」をあわせると全体では、72.5%の学生がプログラムの構造を理解できたとしているが、成績別では、上位者で 63.6%にとどまるのに対して、下位者では 72.5%で理解できたとしている。

プログラムの構造のどの部分で理解ができたのかを訪ねた項目 4 の結果を表 3 に示す。下位者では、「3 つとも」が最も高く、次に「繰り返し」、「分岐」の順になっており「順次」を選択した回答者はいない。一方、上位者では「3 つとも」と「分岐」が同じであり、次に「繰り返し」、「順次」の順になっている。上位者は、アルゴリズムで間違いやすい「分岐」を理解できており、一方下位者では、3 つの構造による理解に差はないことが推測される。

アンケート項目 5.7 の結果を表 4.5 に示す。表 4 より全体では、このビジュアルプログラミングを 60.0%の学生が使用したほうがよいとしており、下位者では、66.7%の学生が来年度以降も使用したほうがよいとしているのに対し、上位者では、約半分である 36.4%にとどまっている。この結果とアンケート項目の 7 の結果を表 5 からみると、全体では、60.0%の学生が役立つとしており、下位者では、表 4 と同様に 66.7%の学生が役立つとしている。それに対して、上位者では、54.4%である。この結果から下位者にとってビジュアルプログラミングは役に立ち来年度以降も実施したほうがよいものとしているのに対し、上位者は、半数以上で理解に役立つとはしているが、来年度以降の使用には否定的な評価であると考えてよい。上位者の自由記述には、「簡単すぎるのもっと Java の演習をした方がよい」、「プログラミングを視覚的に理解することができるので、プログラミングの初心者向き」、「プログラミングの授業には不要」、「文字でのプログラミングがよい」との記述がみられた。プログラミングがある程度のレベルに達している学生にとっては、有用性は認めるが、プログラミングの講義としての評価は高くなかったようである。一方、下位者の自由記述には「文法だけではない根本的な構造等をわかりやすく理解できたのでよかった。視覚的に楽しく学習できた」という好意的なコメントも見られた。この結果は、先行研究でも指摘されている通り、「初学者にとっては学びやすく入門に適している。」、「苦手意識を持つ学生ほどテキスト記述式よりもビジュアルプログラミングを選ぶ」傾向があることと一致している。

以上の結果から、本研究目的であるプログラミングに苦手意識を持ちやすいと思われる成績下位者に対しては、評価が高く、苦手意識を改善させ、理解を促すものになったのではないかと考える。

3.1.2 点数との関係

ビジュアルプログラミングを使用した場合と使用しない場合の前期末試験の点数を比較した。なお、同一教科書を使用し、同じカリキュラムで講義を行っている 2012 年から 2014 年の 3 年間を比較した。比較は、同じ教員 2 名がビジュアルプログラミングを実施した年と実施していない年、教員 2 名のうち 1 名が別の教員で未実施の年である。ビジュアルプログラミングを未実施で教員が同じ場合、平均点 67.50 (SD 27.33) 点、ビジュアルプログラミングを未実施で教員が異なる場合、平均点 55.20 (SD 22.52) 点、ビジュアルプログラミングを実施で教員が同じ場合、平均

57.99 (SD 18.58) 点である。それぞれ t 検定(有意水準 0.05)で両側検定を行った結果、教員が異なる場合に有意な差があった。また、実施の有無についても有意な差が得られた。教員の違い、ビジュアルプログラミングの実施の有無ではいずれも有意な差が得られ、教員の違い、実施の有無は試験の点数に影響があることがわかる。これは、教員が異なることで、教え方、試験問題の難易度に違いが出るためこのような結果になったと考える。また、同じ教員でもビジュアルプログラミングの有無で、点数に違いがあり、逆に実施した場合に平均点が低下したことについては、実施した試験は Java でコーディングするものでありビジュアルプログラミングによる評価にはなっていないことも要因と考えられる。また、3 年間の比較では、試験問題が異なるうえ難易度をそろえることをしていない等の問題があったため、点数で単純に比較ができなかった。このため単純に試験の点数でビジュアルプログラミングの評価はできていない。更に教員が異なる場合での実施の有無の比較もできていないので、今後検討が必要である。

3.1.3 講義の構成とツール

講義に使ったツールの加速度センサプログラムの使いやすさと講義演習等の時間についてアンケート結果をまとめる。表 6 に使いやすさについて尋ねた設問 2 の結果を、表 7 に、時間について尋ねた設問 6 の結果を示す。

全体は、「使いやすい」と「まあ使いやすい」を合わせると 60%であり、評価は高い。特に下位者と上位者を比べると上位者で 27.3%しか「使いやすい」と評価していないのに対し、下位者の 44.4%は「使いやすい」としている。また上位者の 18.2%は「少し使いにくい」との評価であるのに対して、下位者は 0.0%である。テキスト記述型の Java のコーディングが得意な者、あるいは慣れていていると思われる上位者には使いにくい、一方 Java のコーディング初学者には評価が高かったことがわかる。初学者にはよいが、上級者には物足りない結果であったといえる。

表 7 から時間については下位者で 22.2%が「すくなくかった」、55.6%が「ちょうどいい」というのに対して上位者では下位者の半数に近い 9.1%と 27.2%である。一方、「少し少なかった」「少し多い」「多い」は、下位者よりも上位者のほうが増えている。この結果から、下位者では、「ちょうどいい」あるいは「少ない」としているのに対し、上位者では「ちょうどいい」、「少し多い」と感じているようである。このことは、成績上位者あるいはプログラミングに慣れていている者に対して講義内容がものたりないと感じていることにつながっているのではないかと考えられる。

4. まとめ

本研究では、学生の苦手意識を改善し、プログラミングの理解を深めることを目的とし、ビジュアルプログラミングを用いたプログラミング導入を行った。アンケートの結果を受講者全体と成績上位 25%、成績下位 25%に分類し、比較した。上位者では、ビジュアルプログラミングに評価が低いものも見られたが、下位者では高評価のものが多かった。プログラミング構造の理解についても、上位者では、3 つとも、分岐の評価が高く、下位者では 3 つとも理解が高くなっていることが分かった。また、教員別、ビジュ

アルプログラミングの講義の有無で点数を比較したところ有意な差がえられた。しかし、問題の難易度の設定や Java のみの試験であったため、ビジュアルプログラミングによる効果の検証には至らなかった。

以上より、プログラミングが苦手な学生にはビジュアルプログラミングの評価が高く、研究目的である苦手意識の改善にはつながると考えられる。一方、得意な学生の評価はそれほど高くないことも分かった。これは先行研究でも指摘されているように初学者への導入としては、本研究が一定の効果があったといえる。一方、ビジュアルプログラミングからテキスト記述型の言語へのシームレスな移行は、不十分なままである。この点については、アルゴリズムの学習をビジュアルプログラミングとは別に行うこと、マウス操作によって作成したコードをテキストに置き換えるなどの演習を十分に行うか、あるいはそれを行うことができるツールの導入を行うことが必要と考えている。また、これまでのカリキュラムを変更しない範囲での導入であったため、ビジュアルプログラミングの実施が3回のみであったなど講義時間が少なかったことも今後検討していきたいことである。更に、来年度以降の利用、時間について下位学生と上位学生で異なる回答であるため、習熟度別の課題を設定するなどの対策が必要であると考えている。

表3 理解箇所(設問4) [単位: %]

	全体	下位 25%	上位 25%
順次	10.0	0.0	7.1
分岐	22.0	10.0	35.7
繰り返し	20.0	20.0	21.4
3つすべての理解に役立った	40.0	60.0	35.7
理解には役立たなかった	6.0	0.0	0.0
未回答	2.0	10.0	0.0

表4 今後使用したほうがよいか(設問5) [単位: %]

	全体	下位 25%	上位 25%
ぜひつかったほうがいい	25.0	44.4	0.0
どちらかといえばつかったほうがいい	35.0	22.2	36.4
どちらともいえない	22.5	33.3	36.4
あまりつかわないほうがいい	12.5	0.0	18.2
つかわないほうがいい	5.0	0.0	9.1
未回答	0.0	0.0	0.0

表1 わかりやすさ(設問1) [単位: %]

	全体	下位 25%	上位 25%
わかりやすい	60.0	55.6	63.6
まあまあわかりやすい	15.0	33.3	9.1
ふつう	17.5	11.1	27.3
すこしわかりにくい	2.5	0.0	0.0
わかりにくい	5.0	0.0	0.0
未回答	0.0	0.0	0.0

表5 理解の役に立ったか(設問7) [単位: %]

	全体	下位 25%	上位 25%
役立った	35.0	55.6	27.3
まあ役立った	25.0	11.1	27.3
ふつう(どちらともいえない)	22.5	22.2	36.4
あまり役に立たなかった	15.0	0.0	9.1
役に立たなかった	0.0	0.0	0.0
未回答	2.5	11.1	0.0

表2 プログラムの構造理解(設問3) [単位: %]

	全体	下位 25%	上位 25%
できた	37.5	55.6	45.5
まあまあできた	35.0	33.3	18.2
ふつう(どちらともいえない)	17.5	11.1	18.2
あまり理解できない	5.0	0.0	9.1
理解できない	2.5	0.0	0.0
未回答	2.5	0.0	9.1

表6 使いやすさ(設問2) [単位: %]

	全体	下位 25%	上位 25%
つかいやすい	42.5	44.4	27.3
まあまあつかいやすい	17.5	22.2	18.2
ふつう	30.0	33.3	36.4
すこしつかいにくい	7.5	0.0	18.2
つかいにくい	2.5	0.0	0.0
未回答	0.0	0.0	0.0

表 7 講義時間(設問 6) [単位: %]

	全体	下位 25%	上位 25%
すくなかった	12.5	22.2	9.1
すこしすくなかった	22.5	0.0	18.2
ちょうどいい	37.5	55.6	27.3
すこし多い	12.5	11.1	18.2
多い	12.5	11.1	18.2
未回答	2.5	0.0	9.1

謝辞

アンケート等にご協力いただいた本校学生の皆様に感謝します。また、データ整理や分析をおこなっていただいた研究支援員の河村絵美さんにも感謝します。

参考文献

- [1] 大木真, 永田和生, “プログラミング導入学習の改善”, 熊本高等専門学校 研究紀要, Vol.5, No.1(2014).
- [2] 角田博保, “Column プログラミングは楽しいですか?”, 情報処理, Vol.52, No.12 (2011).
- [3] 内田眞司, 松村寿枝, 西野貴之, 松尾賢一, 本間啓道, 山口賢一, “複数教員によるプログラミング教育, 第 28 回高専情報処理教育研究発表会(2008)
- [4] 内田眞司, 松村寿枝, 西野貴之, “習熟度を導入したプログラミング教育の試み”, 情報教育研究集会講演論文集 2008 年度, (2008)
- [5] ようこそ スクイークランドへ, <http://squeakland.jp/>
(参照: 2015 年 4 月 3 日)
- [6] Scratch-想像, プログラム, 共有, <https://scratch.mit.edu/>
(参照: 2015 年 4 月 3 日)
- [7] 杉浦学, 松澤芳昭, 岡田健, 大岩元, “アルゴリズム構築能力育成の導入教育: 実作業による概念理解に基づくアルゴリズム構築体験とその効果”, 情報処理学会論文誌, Vol.49, No.10(2008)
- [8] 森秀樹, 杉澤学, 張海, 前迫孝憲, “Scratch を用いた小学校プログラミング授業の実践”, 日本教育工学論文誌, 34, 4, (2011)
- [9] 松澤芳昭, 酒井三四郎, “ビジュアル型プログラミングとテキスト記述型言語の併用によるプログラミング入門教育の試みと成果”, 情報処理学会研究報告, Vol.2013-CE-119, No.2(2013)
- [10] 松村寿枝, 内田眞司, 近藤勝也, “スクイークを用いた小中学生のためのプログラミング講座 - 奈良高専情報工学科公開講座報告 -”, 奈良高専研究紀要, 45(2009)
- [11] 松村寿枝, 内田眞司, 山口智浩, “Squeak を用いた小中学生のためのプログラミング講座 - 奈良高専情報工学科公開講座 -”, 高専教育第 34 号, (2011)
- [12] 加速度センサプログラマーヴィストン株式会社, http://vstone.co.jp/products/vs_gs001/index.html
(参照: 2015 年 4 月 3 日)