

動画を用いたプログラミング自学自習の試み

伊藤 恵¹ 杉本 識吏¹ 大場 みち子¹ 下郡 啓夫²

概要：高等教育機関におけるプログラミング教育では対面授業と e-learning を併用したブレンディッド学習が使われるようになってきている。しかし、情報系大学におけるプログラミング教育は必修科目となっている場合が多く、大人数授業となるため、学習者個々の状況に応じた対応が難しく、e-learning 等を用いた自習に多く依存しがちである。自習によるプログラミング学習では、プログラムの作り方を学ぶために例題プログラムを見て、それを真似てプログラム作成をすることが多いが、出来上がったプログラムを十分には理解せずに書き写すだけになり、プログラムは完成しても作り方は学べていないことが多い。そこで我々はプログラムの完成形ではなく、プログラムが作られていく過程を学習者に示すことによって、表面的ではないプログラミングの理解と学習を支援できるのではないかと考えた。具体的には、プログラミングを行っている過程の動画を収集し、学習者に応じた動画を閲覧させた後に、実際のプログラミングを行わせる。実際にいくつかのプログラミング課題について動画の収集とそれを閲覧させた上でプログラムを作成させる実験を行ったため、その結果と考察を述べる。

1. はじめに

高等教育機関におけるプログラミング教育では対面授業と e-learning を併用したブレンディッド学習が使われるようになってきている [1]。しかし、情報系大学におけるプログラミング教育は必修科目となっている場合が多く、大人数授業となるため、学習者個々の状況に応じた対応が難しく、e-learning 等を用いた自習に多く依存しがちである。自習によるプログラミング学習では、プログラムの作り方を学ぶために例題プログラムを見て、それを真似てプログラム作成をすることが多いが、出来上がったプログラムを十分には理解せずに書き写すだけになり、プログラムは完成しても作り方は学べていないことが多い。

そこで我々はプログラムの完成形ではなく、プログラムが作られていく過程を学習者に示すことによって、表面的ではないプログラミングの理解と学習を支援できるのではないかと考えた。具体的には、プログラミングを行っている過程の動画を収集し、学習者に応じた動画を閲覧させた後に、実際のプログラミングを行わせる。実際にいくつかのプログラミング課題について動画の収集とそれを見せた上でプログラムを作成させる実験を行ったため、その結果と考察を述べる。

2. 先行事例

2.1 プログラミング教育

情報系の高等教育機関では必修、大人数授業。提示された課題に応じて学習者がプログラムを作成し、教師がそれを採点。学習者はプログラムの作り方を座学講義または自習により学ぶ。教師によるライブコーディングを除き、プログラムの一部あるいは完成品を見て、そこから作り方を学ぶことが多いが、単に完成したプログラムを書き写すだけで、何故その作り方が良いのか十分な理解ができず、表面的な理解に留まってしまうこともある。また大人数授業であるために学習者の理解度に応じた対応が十分にはできないことが多い。

2.2 e-learning を用いたプログラミング教育

プログラミングに関する知識を問うものや、プログラムの穴埋めをするものなどの e-learning 教材のほか、アルゴリズムを学ぶための支援システムや、学習者が作成したプログラムを予め用意された自動採点プログラムにより採点するものなどがある [2], [3]。スライドと共に教師自身の映像が閲覧できることや、概念を分かりやすく示すためにアニメーションが使用されることもあるが、プログラムがどう作られていくかという過程が分かりやすく示されることは少ない。

¹ 公立はこだて未来大学
Future University Hakodate

² 函館工業高等専門学校
National Institute of Technology, Hakodate College

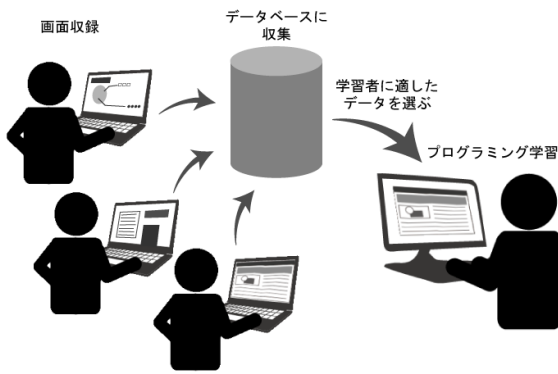


図 1 動画を用いたプログラミング学習

3. 動画を用いたプログラミング自学自習

我々は、プログラミングを行っている過程の動画を予め収集しておき、学習者の理解度に応じた動画を提示することで、プログラミングに対する表面的でない理解と学習を支援することを目指す(図1)。

本稿ではプログラミングを動画収録される被験者を「作成者」、動画を見てプログラミングを学ぶ被験者を「学習者」と呼ぶこととする。また、動画はプログラミングを行っている間のすべての作業が記録されるように、プログラミング作業中のコンピュータ画面すべてを動画収録することとする。

3.1 作成者と学習者のスキル

プログラミングは概して個々のスキル差が大きくなりがちであり、作成者と学習者のスキル差が大きいと、本研究で目指す学習の効果はほとんど得られないと考えられる。作成者と学習者のスキルがほぼ同程度か、もしくは作成者の方が学習者よりもやや高い程度が、この学習方法の効果が強いと考えられる。

3.2 問題からプログラムへの落とし込み

一般にプログラミングの課題は主に文章で問題が説明され、それを満たすプログラムを作成するものが多い。このような課題を解く過程は、i) 問題文の理解、ii) 理解した問題から作成すべきプログラムのアルゴリズムへの変換、iii) プログラミング言語での表現の3段階に分けられると考えられる。ごく簡単な課題の場合、ii) 理解した問題から作成すべきプログラムのアルゴリズムへの変換の比率は小さいが、課題が難しくなるに従ってその比率は大きくなり、i) 問題文の理解や iii) プログラミング言語での表現は分かっている、ii) 理解した問題から作成すべきプログラムのアルゴリズムへの変換が出来ないために課題を解けないことが増えてくると考えられる。

表 1 完成プログラムと作成過程動画からの学び(仮説)

	完成プログラムからの学び	作成過程動画からの学び
問題文の理解	×	△
理解した問題からアルゴリズムへの変換	△	○
プログラミング言語での表現	○	○

プログラミングを行っている過程の動画を見る場合、直接的に得られるのはプログラミング言語での表現だが、その表現(つまりプログラム)が作られている過程を見ることで、理解した問題をどうアルゴリズムに変換したかや、場合によっては問題文をどう理解したかも垣間見れる場合があると考えられ、完成プログラムそのものではなく、作られていく過程の動画を見るからこそ学べる要素が大きいと考えられる(表1)。

3.3 プログラム作成スタイル

プログラムを作る際の作成スタイルも様々ある。プログラムをほとんど書き上げてから初めてコンパイルと実行をするスタイルや、少しずつ書いてコンパイルおよび実行を繰り返すスタイル、あるいは前半と後半でこの両者を組み合わせるスタイルなどである。プログラミングを行っている過程の動画には、このような作成スタイルがそのまま収録されるため、作成スタイルに関する気づきや学びも得られる可能性がある。

4. 学習実験

動画を見ることによるプログラミング学習について、2種類の実験を行った。実験1は、そもそもプログラミング動画を見ることで何らかの学習効果がありそうかどうかを確認する実験である。実験2は、プログラミング動画を見ることによる学習と単に完成プログラムを見るだけによる学習との学習効果の違いを確認するための実験である。以下の4.1~4.3節で、対象としたプログラム課題と被験者、プログラミング動画の収録、動画を用いた自学自習に関して、実験1と実験2の双方について述べる。

4.1 対象としたプログラム課題と被験者

著者ら所属大学では学部1年次と2年次にプログラミングの授業があるが、プログラミングの様子を動画収録される「作成者」としては、学部1年生から修士2年生まで幅広く対象とした。また、動画を見てプログラミングを学ぶ「学習者」としては、プログラミングを学んでいる最中の学部1年生のほか、プログラミングが不得意な学部2~4年生も対象とした。

著者ら所属大学において学部全員が必修科目として受講するC言語のプログラミングを対象とし、繰り返し、配

表 2 実験 1 の動画データ (繰り返しの課題)

番号	作成時間	削除文字数
a ₁	6'38"	9
a ₂	7'08"	16
a ₃	5'32"	10
a ₄	21'00"	47
a ₅	3'38"	21
a ₆	29'09"	172
a ₇	13'03"	32
a ₈	6'38"	9
a ₉	7'08"	16
平均	7'31"	36.9

列, 関数, 構造体などに関する比較的簡単なプログラミングの課題を扱うこととした。

実験 1 では繰り返し, 関数, 配列に関する課題各 1 問計 3 問を用意し, 作成者には課題 3 問を解いてもらい, それらから学習者に合わせた動画を選定した上で, 動画と同じ課題を解いてもらうこととした。実験 2 では二重ループと構造体に関する課題を複数用意し, 作成者と学習者は類似するが異なる課題各 1 問ずつを解くこととした。

4.2 プログラミング動画の収録

プログラミング過程の動画収録には, Mac OS X に付属する QuickTime の画面収録機能を利用した。また, プログラミングの授業で用いているものと同じだけ同じプログラミング環境になるよう, 可能な限り授業と同じ教室で, 授業と同じ端末を使用した。エディタおよびコンパイラとして授業で用いているものと同じ Emacs と gcc を使用した。

実験 1 では, C 言語のプログラミング授業を受講中の学部 1 年生 8 名と受講済みの学部 4 年生 4 名の計 12 名に繰り返し, 関数, 配列に関する課題各 1 問計 3 問を解いてもらい, その様子を画面収録した。収録時間 30 分で課題を 3 問解いてもらったところ, 収録時間内に各課題のプログラムを完成できた人数は繰り返しの課題が 9 名, 関数が 4 名, 配列が 3 名であったことから, この実験では完成した人数が最も多い繰り返しの課題の動画のみを使用することとした。繰り返しの課題プログラムを完成できた 9 名の動画で課題プログラム作成時間のほか, プログラムを円滑に作成したかどうかの目安として作成途中の削除文字数を計測したところ, 表 2 の通りであった。これらの動画を学習者に見せる際, 習得スキルが低いかプログラミングが苦手な学習者には, 作成時間が長かったり, 削除文字数がある程度多い動画を, 習得スキルがより高くプログラミングが苦手でない学習者には, 作成時間が短く, 削除文字数も少ない動画を見せることを想定した。

実験 2 では, C 言語のプログラミング授業を受講済みの学部 3 年生 1 名, 学部 4 年生 12 名の計 13 名に二重ループ

表 3 実験 2 の実験種類別学習者数

	完成プログラムを見て 類似課題を解く	動画を見て 類似課題を解く
二重ループ課題	9 名	16 名
構造体課題	7 名	16 名
計 (重複除く)	16 名	16 名

を用いる課題と構造体を用いる課題各 1 問計 2 問を解いてもらい, その様子を画面収録した。1 問につき 30 分の収録時間で課題 2 問を計 1 時間で解いてもらったところ, 収録時間内に二重ループの課題を完成できたのは 2 名, 構造体の課題を完成できたのは 4 名であった。完成までの所要時間やプログラム作成中の削除文字数等を踏まえ, 二重ループと構造体のそれぞれの課題に関して, プログラミングが得意な作成者の動画と不得意な作成者の動画を選定した。また, 選定した 30 分間の動画から画面の動きが全くない部分など, 学習に不要と思われる部分を除去し, 約 10 分間の動画教材として利用することとした。

4.3 動画を用いた自学自習

実験 1 ではプログラミングが不得意であると自称する学部 4 年生 2 名および修士 1 年生 1 名にプログラミングの動画 30 分を見せた後, 動画で見せたものと全く同じ課題を制限時間 20 分で解いてもらった。動画を見る際にはメモは取って良いが丸写しはしないことを条件とした。制限時間内に課題のプログラムを完成できたのは 3 名中 2 名であり, 完成した 2 名の平均所要時間は 13 分 12 秒だった。実験後のアンケートにより 3 名とも「動画を見ることで解きやすくなった」, 「動画を見たことでプログラムを読む手順がわかった」と答えた。また, 修士 1 年生 1 名は過去 2 年間プログラミングを行っていなかったが, 動画をみることでプログラミングを思い出せたとの意見を述べていた。

実験 2 では学部 1 年生 14 名, 学部 2 年生 2 名の計 16 名に学習者として実験に参加してもらった。16 名に対する事前アンケートにより, 相対的にループが苦手であるとする 9 名を A グループ, 相対的に構造体が苦手であるとする 7 名を B グループとした。この実験では動画で見るプログラムと課題で解くプログラムは類似するが異なる問題とした。また, 課題を解く前に動画を見ることと, 動画は見ずに完成済みのプログラムだけを見ることの差異を確認するため, 完成プログラムを見てから類似課題を解くものとプログラミング動画を見て類似課題を解くものの 2 種類の実験とした。完成プログラムを見てから類似問題を解くのは A グループ, B グループそれぞれ苦手とする課題のみ, 動画を見て類似問題を解くのはどちらの種類の課題も解いてもらうこととした。実験種類別の各課題ごとの学習者の人数は表 3 の通りである。

学習者 16 名中, 相対的にループが苦手な者 9 名には二重ループの課題を, 相対的に構造体が苦手な者 7 名には構

表 4 実験 2 の結果 (個人単位)

グループ	課題 1	課題 2	課題 3
A	-	-	-
A	-	-	-
A	-	-	-
A	-	26'16"	-
A	-	25'01"	-
B	-	-	30'16"
A	-	-	25'18"
A	-	-	21'08"
A	-	-	17'11"
B	-	-	15'21"
B	-	-	11'19"
B	-	-	7'20"
A	-	11'46"	7'03"
B	18'40"	-	10'13"
B	16'09"	19'45"	12'33"
B	15'59"	30'25"	10'18"
平均	16'59"	20'08"	16'02"

造体の課題を割り当て、二重ループまたは構造体を用いる完成プログラムを 10 分間見てから類似の課題を制限時間 30 分で解いてもらったところ、制限時間以内にプログラムを完成できたのは 16 名中 3 名で、3 名の平均所要時間は 16 分 56 秒であった。完成したのは 3 名とも構造体の課題に取り組んだ学習者であった。

学習者 16 名全員に二重ループのプログラミング動画を 10 分間見てから類似の課題を制限時間 30 分で解いてもらったところ、制限時間内にプログラムを完成できたのは 16 名中 6 名で、6 名の平均所要時間は 20 分 8 秒であった。

学習者 16 名全員に構造体のプログラミング動画を 10 分間見てから類似の課題を制限時間 30 分で解いてもらったところ、制限時間内にプログラムを完成できたのは 16 名中 10 名で、10 名の平均所要時間は 16 分 2 秒であった。

5. 評価と考察

2 つの実験結果を踏まえて、評価と考察を述べる。

実験 1 では比較実験等を行っていないほか学習者の人数も少ないが一定の学習効果はありそうであることは確認できた。実験後に学習者 3 名に対して、動画を見て課題が解きやすくなったか、動画を見てプログラムを読む手順が分かったか等の項目について「はい」「いいえ」の 2 択でアンケート調査を行ったところ、いずれも「はい」との回答が得られた。

実験 2 全体を通して、学習者 16 名は相対的に苦手と自己申告した種類の課題を完成プログラムを見てから解き、次にそうでない種類の課題を動画を見てから解き、最後に相対的に苦手と自己申告した種類の課題を動画を見てから解いた。つまり、限定的な実験ではあるものの、最初の課題の結果と最後の課題の結果を比較することで完成プロ

ラムを見てからプログラミングすることと、動画を見てからプログラミングすることの差を見ることができる。実験 2 の学習者 16 名の 3 つの課題のプログラム作成時間は表 4 の通りである。時間はその課題のプログラムの完成までの所要時間で、表中の - 記号はプログラムを完成できなかったことを示している。学習者各自が同じ種類の課題を解いた課題 1 と課題 3 の結果を比較すると、課題 1 も課題 3 も未完成の学習者が 5 名、課題 1 は未完成だが課題 3 は完成した学習者 8 名、どちらも完成していて課題 1 より課題 3 の方が早く完成した学習者 3 名となっており、課題 1 よりも課題 3 の方が出来が悪くなっている学習者は居ない。

実験後に学習者 16 名に対して、動画を見る前後のプログラミングの違いや動画を見ることによる学びについてアンケート調査を行った (図 2)。各項目について「はい」「どちらかと言えばはい」「どちらかと言えばいいえ」「いいえ」の 4 択で回答してもらったところ、多くのアンケート項目について 7 割前後の学習者が動画を見ることへの肯定的な回答をしていた。「動画を見ることで構造体に対する理解は深まる」との項目については否定的な回答が 5 割を超えていた。また「動画を見ることで忘れていたプログラミングを思い出した」との項目については肯定的な回答は 5 割を超えているが他の項目よりは低かった。実験 1 では授業でプログラミングを学んだ後しばらくプログラミングをしていなかった学習者が対象であったが、実験 2 ではプログラミングを学んでから間もない 1,2 年生が学習者だったため、この項目への肯定的な回答がさほど多くはなかったものと考えられる。

この実験ではどの学習者も、完成プログラムのみを見てからのプログラミングを先に行い、動画を見てからのプログラミングを後に行っているため、同じ課題ではないものの後で行ったプログラミングの方が出来が良くなる可能性があるため、どこまでが動画を見たことによる効果であるかはこの実験結果だけからでは明らかではないが、アンケート結果を踏まえると一定の効果はあると考えられる。

実験 2 の結果からは全体的に構造体の課題の出来が良いように見られるが、二重ループと構造体の用意した課題自体に難易度の差があった可能性があるほか、事前アンケートでどちらかと言えば構造体が苦手と答えた学習者がどちらかと言えばループが苦手と答えた学習者よりも平均的にプログラミングスキルが高かった可能性がある。

6. おわりに

プログラミングの自学自習において、完成したプログラムを見ることでなく、プログラムが作られていく過程の動画を見ることによって、表面的でないプログラミングの理解と学習を支援することを目指し、実際にプログラミング動画の収集とそれらを用いた学習実験を行った。限定的な実験ではあるものの、動画を見ることによる学びは相応に

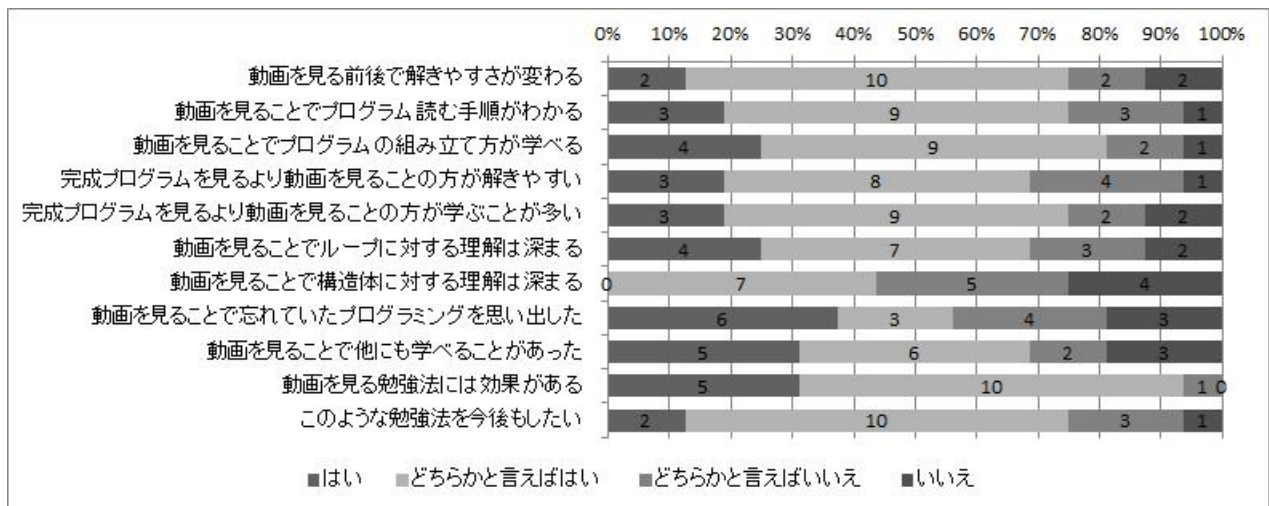


図 2 実験 2 学習者アンケート結果

得られる可能性があることが分かった。引き続き、動画の収集方法、動画を用いた学習方法等を検討し、学習実践を重ねることで本手法によるプログラミング学習の可能性を実証していく。

謝辞 本研究は JSPS 科研費 25381285 の助成を受けたものです。

参考文献

- [1] 中尾茂子, 安達一寿, 北原俊一, 新行内康慈: ブレンディッドラーニングによるプログラミング学習の実践と評価, 教育情報研究: 日本教育情報学会学会誌, Vol. 22, No. 3, pp. 47-56 (2007).
- [2] 新開純子, 宮地功: ブレンド型授業によるプログラミング教育の効果, 教育システム情報学会誌, Vol. 28, No. 2, pp. 151-162 (2011).
- [3] 伊藤恵, 美馬義亮, 大西昭夫: コース管理システムと授業固有の課題チェック機能の Web サービスによる連携, 情報処理学会論文誌, Vol. 52, No. 12, pp. 3121-3134 (2011).