

体験型デジタル教材を教師が容易に作成可能なプログラミング環境の開発

1245126 筒井 まや乃 【ソフトウェア検証・解析学研究室】

A Programming Environment That Allows Teachers to Easily Create Digital Learning Materials

1245126 TSUTSUI, Mayano 【Software Verification and Analysis Lab】

1 はじめに

近年、情報化やグローバル化にともなって、日本ではそれらの社会情勢に対応できる人材の育成を目的とした教育の改革が進められている。例えば、子供の問題解決能力を育てる目的で、小学校ではプログラミング教育の全面実施や高等教育では教科「情報I」のカリキュラムの新設が行われている。また、PCや端末が普及し利用される社会になったことで、生徒が利用可能な端末を支給する学校が増えている[1]。このような教育の変化に伴ってデジタル教材の需要が増加しており、多くのデジタル教材が作成されている。これらデジタル教材は生徒自身が体験することで物事への理解を促すような分かりやすさを重視した体験型デジタル教材が多い。例えば、ドリルのように問題を解くことで知識の定着をはかる教材、シミュレータのように自身で操作することで起きる変化を観測する教材が体験型デジタル教材として挙げられる[2]。しかし、教師が生徒に学ばせたい項目のデジタル教材が利用可能な教材の中に含まれているとは限らない。そこで教師が自身の授業に応じたデジタル教材を作成することを目的としたシステムが必要であると考えられる。

本研究では、体験型デジタル教材を教師が容易に作成可能なプログラミング環境「SlideBlock」の開発を行い、評価を行った。

2 提案開発環境

本研究で提案する体験型デジタル教材開発環境の実行画面は図1のようになっている。

本研究の目的は、教師が容易に扱うことができ、目的に応じたデジタル教材の開発ができる教材開発環境を実現することである。しかし、既存のデジタル教材作成環境は、教科書の内容に書き込みを行うソフト、文章作成ソフト、スライド作成ソフトのように特定のデジタル教材作成に特化したものが多い。自由度が高い開発環境となるとプログラミング言語を用いて開発する必要が出てくるが、プログラミング言語を学ぶことは学習コストが高くなるため、容易に扱うことができるという本研究の目的から外れる。そこで提案開発環境ではデジタル教材開発を行うのに自由度が高く、学習コストを少な

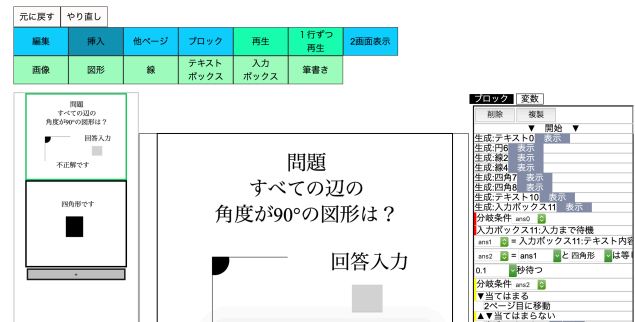


図1 提案するデジタル教材開発環境

くするため、ビジュアルプログラミング環境を採用した。また、提案開発環境の画面構成を多くの教師が利用したことがあるスライド作成ソフトに似せることで、操作方法を理解しやすいようにしている。例えば、スライド作成ソフトでは左側の画面で複数のスライドの管理を行い、右の画面内で管理している1枚のスライドの編集を行えるが、提案するデジタル教材開発環境でも同じような画面構成を採用している。これによって、右画面は1枚のデザイン画面の編集画面であることを強調し、1枚のデザイン画面に対して1つのプログラム編集画面がセットになっていることを理解しやすくしている。

操作方法がスライド作成ソフトに類似しているとともに、教材を構成する各画面（デザイン画面とプログラムの組）をそれぞれスライドのように管理・編集できる点が、既存のビジュアルプログラミング環境とは異なる提案開発環境の特徴である。このような方式にすることで、画面単位での拡張や改変が容易であり、また、作成済みの教材の一部を再利用することも容易である。テンプレートとなるデザイン画面とプログラムの組を共有すればさらに教材作成時間を短縮できると考えられる。

上記の特徴の利点として、表示内容を切り替えながら進むデジタル教材の作成が容易である点も挙げられる。例えば、複数のクイズ画面とどのクイズに回答するか選択するメニュー画面からなる教材を考えると、Scratch[3]等のプログラミング環境では画面上の各オブジェクトの表示・非表示の切り替えによって表示内容を変えること

になり、プログラムが複雑化し、モジュール化もしにくい。提案開発環境では、各画面をそれぞれ作成し、各画面のプログラム中に他の画面への移動命令を記述するだけで作成できる。また、このようなデジタル教材は授業中や授業ごとに利用する教材をその場ですぐに切り替えて使い分けができるため、複数の教材を利用しやすくなる。

3 被験者による評価

提案開発環境を使うことでデジタル教材がより作成しやすくなるか評価するために実験を行った。実験は既存のビジュアルプログラミング環境「Scratch」[3]と提案開発環境「SlideBlock」で同じようなデジタル教材を作成してもらい比較実験である。被験者として教職経験者計5名に実験に協力してもらいアンケート調査や口頭での聞き取りを行った。

実験方法は、まず先に使用するプログラミング環境の操作方法の説明を20分から30分行った後に、15分間でデジタル教材の作成を行ってもらった。被験者のうち3人には、先にScratch、次にSlideBlockで教材作成を行ってもらった。残りの2人には逆の順番で教材作成を行ってもらった。各プログラミング環境でデジタル教材の作成が終わった時点でアンケート調査を行った。

各アンケート調査は2つの設問に分けて行った。まず設問1はどちらが容易に教材を作成できたのか比べるため、システムユーザビリティスケール(SUS)を用いて5段階評価を行った。次に設問2では口頭またはアンケートで「作成したデジタル教材の良い点悪い点」「作成に利用した開発環境の良い点悪い点」を調査した。比較アンケートでは上記の設問1と同じ質問でScratchとSlideBlockどちらの方がよりそう思えるのか調査を行った。

4 結果・考察

ScratchとSlideBlockのSUS評価値を表1に示す。

各SUS評価の値からSlideBlockはどの値においてもScratchを下回っており、Scratchより容易に利用できるとは言えない結果となった。

設問2のアンケート調査では、Scratchを利用した場合の意見としては、色で分けられていることで覚えやすく操作方法が複雑ではないため扱いやすいといった意見や、キャラクターが動く教材が作成されるため子供向けなら作成できそうといった意見、簡単にプログラムを構築できる反面、簡単にしすぎて自由度が足りないといった意見があった。次にSlideBlockへの意見としてはメニュー欄が階層構造になっているため覚えるまで扱うのが難しいといった意見や、普段使っているスライド作成ソフトに近いので操作方法が分かりやすいといった意見、変数操作の結果をデザイン画面に表示しやすいのでプログラムの構造が理解しやすいといった意見があった。

表1 各開発環境でのSUS評価値

項目	Scratch	SlideBlock
頻繁に利用したい	2.8	1.6
利用する必要はない	2.8	2.2
容易に使える	3.2	1.6
扱うにはサポートが必要である	2	1.8
うまく機能が統一されている	3	2
操作の際に機能を 探すことが困難である	2.2	1.6
使用方法について、素早く学べた	2.6	1.8
使い方が非常に難しい	3	2.2
デジタル教材を 作成できる自信がある	2	2
使いこなすには、 もっと学ぶ必要がある	1	0.6
SUS評価値	61.5	43.5

SlideBlockのSUS評価値が低い結果になった理由としては、操作説明時間が短く慣れ親しむ時間が少ないため、色や形で認識可能なScratchのプログラムの方が多少時間が経った後も覚えやすく、それに対してSlideBlockはプログラムを選択するメニュー欄が階層構造であるため、プログラムの形だけではなくどのメニュー欄を開くのかといった操作の方法まで覚えなければならないことが考えられる。つまり、初めて使う人から見た使いやすさの点ではSlideBlockはScratchに及ばなかったと考えられる。

一方、2節で述べた既存の教材開発環境に対する提案教材開発環境の利点は、設問2のアンケート調査結果では5人中4人が良かった点として挙げていることから、その点では提案開発環境の目的は達成できたといえる。

5 まとめ

本研究では、体験型デジタル教材を教師が容易に作成可能なプログラミング環境の開発を行い、評価を行った。比較アンケートでは既存の開発環境の方がより扱いやすいと評価を受けたが、提案開発環境の利点として想定した点を有用と評価する意見も得ることができた。

今後の課題として、今回の評価で調査することができなかつたより長い期間用いた際の調査や、指摘されたユーザーインターフェイスの改善などが挙げられる。

参考文献

- [1] 文部科学省, “GIGA スクール構想の実現”, https://www.mext.go.jp/a_menu/other/1421443_00002.htm, 2022/1/19.
- [2] 大森 康朝ほか, “教材作成支援ソフト「デジきょ」の改良と試用”, 情報処理学会第80回全国大会講演論文集, pp.863-864, 2018/3/13.
- [3] “Scratch”, <https://scratch.mit.edu/>, 2022/1/19.