

## キーボード操作の時系列情報を活用したコーディング学習支援手法の提案 Proposal of learning method for coding: utilizing time sequence of keyboard operation

星野 裕樹<sup>†</sup>      納富 一宏<sup>†</sup>      西村 広光<sup>†</sup>      示野 浩士<sup>†</sup>  
Yuki Hoshino   Kazuhiro Notomi   Hiromitsu Nishimura   Hiroshi Shimeno

### 1. はじめに

情報化が進んでいる現在、教育機関においてプログラミングの授業が実施されていることもあり、ウェブブラウザ上で動作するプログラミング学習環境の構築や学習状況をリアルタイムにフィードバックするシステムなど、多くのICT活用事例が報告されている<sup>[1]-[4]</sup>。

神奈川工科大学の情報学部においても、プログラミングの授業を実施しており、座学と演習の時間を分けることでプログラムの知識と技術の両方を学べるようにしている。しかし、演習時間において学生の指導を行っているとき、コーディングに関する技術が不十分である学生が多く見られる。そのため、学生が記述したプログラムは、インデントが適切に付与されていないことによる可読性の低下や、条件や関数の終わりを示すコードの記述忘れによってエラーが起こることがある。コーディング技術は授業内で知識として教えるだけでなく、学習者自身が普段から意識してプログラミングを行わなければ、身に付けることは困難である。コーディング技術の学習には、学習者がプログラムに対して他人のレビューを受けたり、ライブコーディングのように他人のコーディングを見て学んだりすることが重要であると考えられる。

Adobe社の商用製品である「Captivate」<sup>[5]</sup>を使用することで、コーディング中の操作を記録し、補足説明を加えるなどしてFlashビデオ形式でのコンテンツ作成ができる。その他にも、デスクトップキャプチャソフトで録画した動画をYouTubeなどで配信するといった方法もある。しかし、専用ソフトのインストールが必要、動画形式によるプログラムの見づらさ、プログラムをテキストデータとして扱えないなど、いくつかの欠点がある。

そこで本研究では、コーディング時のキーボード操作と時間情報の記録・再生ができ、Web上でコーディング技術を学ぶことのできる教育コンテンツの提案と実装方法について示すと共に、利用者である学生からの意見収集とアンケートによる分析結果について報告する。

## 2. システム実装

### 2.1 システム設計

本システムは、①Scanner、②Recorder、③Manager、および④Playerの4つのモジュールから構成される。①Scannerはテキスト編集エリアにおけるキータイピングやマウス操作、テキスト更新などのイベントを監視し、イベントが発生する度に時間情報の取得を行って、各情報をバッファ（メモリ）に格納する。②Recorderはイベント情報および時間情報をバッファからファイルに逐次出力して記録する。また、テキスト編集エリアにおけるソースプログラムを最終形態として出力する。③Managerは全体を管理・制御する。④PlayerはXML形式にて記録された情報に

基づいて、イベントを時間情報通りに実行する。全体のシステム構成を図1に示す。

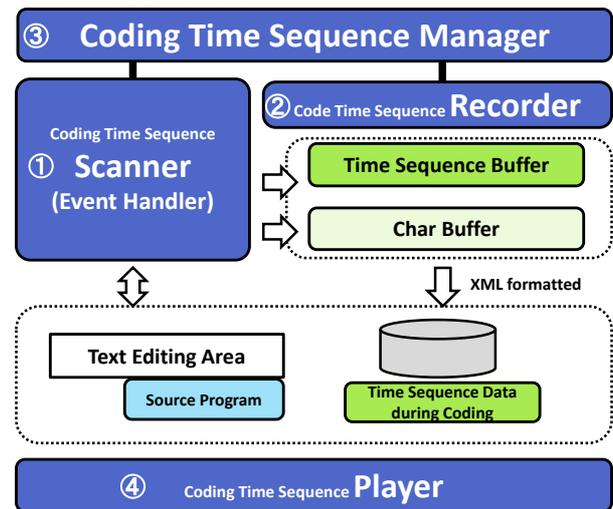


図1 システム構成

本提案システムはWebブラウザ上で動作する。実装には、JavaScript、jQuery<sup>[6]</sup>、およびCodeMirror<sup>[7]</sup>を用いている。また、イベントおよび時間情報の書き込み/読み込みといったサーバサイドの処理にはPHPを用いている。

CodeMirrorとは、JavaScriptで作られたエディタであり、本システムの記録/再生ページにて使用している。オートインデントや行番号表示など、エディタとして必要な機能が一通り揃っており、別途アドオンを読み込むことで、自動補完などの機能追加を行うこともできる。また、対応言語やテーマを変更するなど、エディタの機能から外観まで自分の好みにカスタマイズすることができる。

### 2.2 システム利用

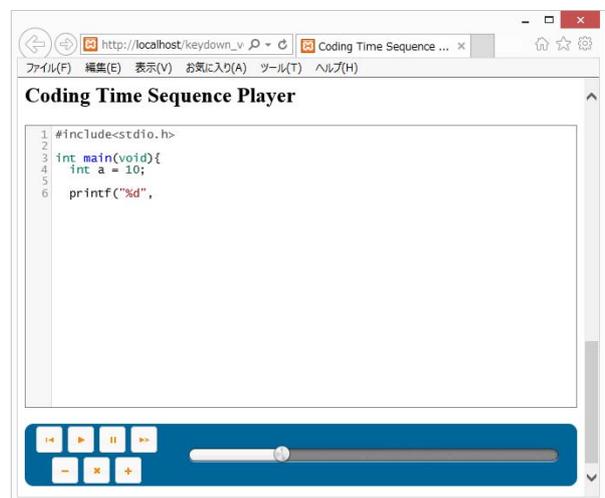


図2 プログラミング動作再生ページ

<sup>†</sup> 神奈川工科大学 Kanagawa Institute of Technology

コーディング動作を記録する場合は、記録ページにて記録開始ボタンを押し、エディタへプログラムを打ち込むことで記録ができる。記録停止ボタンを押すことでサーバへ記録した情報が保存される。

コーディング動作を再生する場合は、図 2 に示す再生ページにて記録データを読み込み、再生ボタンを押すことでプログラムが打ち込まれていく。下部にあるボタンにはそれぞれ処理が割り当てられており、一時停止、再生速度調整、1コマずつ再生といったことも行える。

### 3. システムの評価

#### 3.1 評価方法・結果

被験者として、本学の学生 18 人にデスクトップキャプチャソフト「Camtasia Studio」<sup>[8]</sup>、および「Captivate」で録画したビデオ 2 種類と提案システム上で再生するコーディング動作を記録したコンテンツをそれぞれ視聴してもらう。また、視聴後には提案システム上でコーディングを行ってもらい、記録したものを後で再生して再現度の確認をもらう。試用後にアンケートを行うことでデータを収集する。アンケートの結果から、コーディング技術の学習における提案システムの使い易さと不満点の確認をし、システム改善に役立てる。アンケートの設問項目を表 1 に、集計結果を図 3 に示す。なお、提案システムを「Web システム」と表記している。

表 1 アンケート設問項目

No.	設問項目
1	3 種類のコンテンツを視聴して、どれが一番見易かったですか？
2	授業で実際に使用する場合、どれが一番良いと思いますか？
3	Web システム上でプログラミングを記録して再生したものと、実際に打ち込んでいる時を比較した場合、再現度の差はどのくらいありましたか？
4	本システムで他人のプログラミングを視聴できるようにした場合、学習に役立つと思いますか？

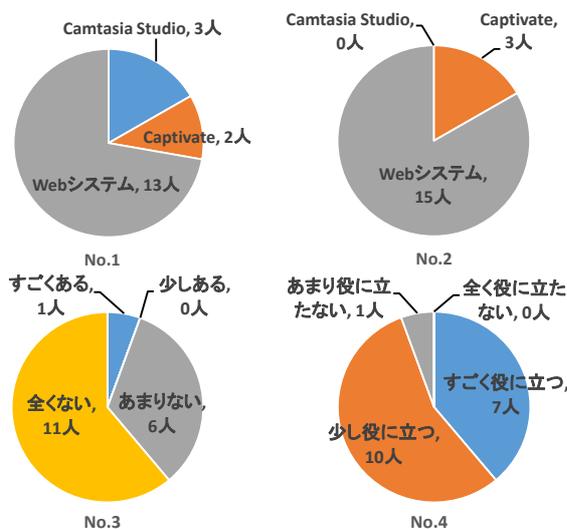


図 3 アンケート結果

#### 3.2 考察

本システムでは、Web ブラウザ上でコーディング動作を記録し、動画で録画した場合と遜色ない形で後から再現することで、自身のコーディングを客観的に見て問題点を発見したり、他人のコーディング過程を視聴して真似たりすることで学習に役立てることを目指している。

設問 3 では、実際にプログラムを打ち込んでいる時との差が「あまりない」、「全くない」と答える学生が 17 人と多く見られた。また、自由記述欄には「自分の打つ速さをリアルに実感できて良い」といったことが書かれていたことから、コーディング動作の再現度については十分であると考えられる。

設問 1 と 2 では、3 種類のコンテンツを比較した場合、本システムで再生されたものが見易くて良いと答えた学生が 13 人いることから、動画で録画した場合と同じことができるのであれば、本システムの方が学習し易いのではないかと考える。現在は再生する機能しかなく、逆再生やシークバーを操作して指定した時刻に飛ぶなどといったことができない。そのため、自由記述欄に「途中から再生できるようにしてほしい」といった意見が書かれていた。今後、逆再生や指定時刻からの再生といった機能の追加を予定している。

#### 4. まとめ

本稿では、コーディング学習を目的とした学習環境の提案と評価について述べた。学生を対象にアンケート調査（被験者 18 人）を行った結果、再生時の見易さ、およびコーディング動作の再現度が十分であることが確認できた。

今後、提案システム上での再生だけでなく、リアルタイムに複数の人と情報共有できるようにし、ライブコーディングやペアプログラミングといった使用方法も検討する。また、現在はコーディングの記録・再生しかできないため、作成したプログラムの実行結果を確認できるようにし、教員によるフィードバック、デバッグのアドバイス表示など、学習支援対象の拡大を行い、実用化を目指す。

#### 参考文献

- [1] 那須野 薫, 上野山 勝也, 松尾 豊, “次世代プログラミング学習サイト構築の試み”, 人工知能学会 全国大会論文集 27, pp.1-3 (2013).
- [2] 畝川 みなみ, 宇野 健, “プログラミング演習における学習状況のフィードバックシステムの開発”, 情報処理学会 第 76 回全国大会講演論文集, pp.549-551 (2014).
- [3] 長瀧 寛之, 北村 英純, 永井 孝幸, 都倉 信樹, “遠隔ペアプログラミング支援システム SATORI の開発とプログラミング教育への適用”, 公益社団法人日本工学教育協会 工学・工業教育研究講演会講演論文集 平成 16 年度, pp.445-446 (2004).
- [4] 梶並 知記, 安田 光, 井上 亮文, 市村 哲, “プログラムコーディング過程を記録した動画教材の作成作業を支援するインタフェース”, 情報処理学会マルチメディア、分散協調とモバイルシンポジウム 2011 論文集, pp.1035-1042 (2011).
- [5] “Adobe Captivate 8”:  
<http://www.adobe.com/jp/products/captivate.html> (参照 2015.06.22).
- [6] “jQuery Terminal Emulator”:  
<http://terminal.jcubic.pl/> (参照 2015.06.22).
- [7] “CodeMirror”:  
<https://codemirror.net/> (参照 2015.06.22).
- [8] “Camtasia Studio”:  
<https://www.techsmith.co.jp/camtasia.html> (参照 2015.06.22).