

クラウドを活用するグラフィックプログラミング教育

矢吹 太郎 Taro YABUKI

佐久田 博司 Hiroshi SAKUTA

概要:CG を題材としたプログラミング教育には、その成果を静止画あるいは動画で見られるという特徴がある。本稿では、これらの成果物をクラウド(画像・動画共有サービス)上で公開する方法を提案する。成果物をクラウド上で公開することには、(1) サーバを用意する必要がない、(2) 情報発信技術を習得できる、(3) 互いの成果物を評価し合うことによって学習者のモチベーションが高まることが期待できる、などの利点がある。公開形式を整えることによって、教師は成果物やそれに対する学習者同士の評価を簡単に取得できる。そのための具体的な公開形式と評価結果の取得方法、筆者らが実際に行った演習授業における試行結果を示す。

キーワード:CG／プログラミング教育／Processing／クラウド／画像共有サービス／動画共有サービス

1. 序論

クラウド、インターネット上で展開されるさまざまなサービスによって、個人が簡単に情報を発信できるようになっている。そのような状況に教育の現場も対応させようというのは自然な考えであろう。本稿では、プログラミング教育を現状に対応させる一手法を提案する。プログラミング教育の成果物のクラウド上での公開である。そのことが、教師や学習者にどのような利益をもたらすかを議論する。

題材はグラフィックプログラミングとする。グラフィックプログラミングの成果物である静止画や動画は、後述する Flickr や YouTube などのサービスによって、クラウド上で容易に公開できるからである。

グラフィックプログラミングのための言語・開発環境は Processing^[1]を利用する。プログラミング言語 Java の簡易版とも言える Processing は、容易に習得できるものであるため、学習者はプログラミング言語自体について苦勞することなく、グラフィックプログラミングの本質を学べるからである。

本稿では、クラウドという用語を「自前の計算資源を必要としないサービス」という意味で用いる。クラウドを、仮想マシン群を提供する Infrastructure as a Service (IaaS) と、ソフトウェアの動作プラットフォームを提供する Platform as a Service (PaaS)、具体的なサービスを提供する Software as a Service (SaaS)に分類する方式においては、本稿で述べるクラウドは SaaS に近い。

2. Processing を利用する CG 教育

本章では、Processing とは何か、Processing を利用す

る演習のスケジュール、演習の成果物を公開するクラウドについて説明する。

2.1. Processing

Processing は Java をベースにしたプログラミング言語・開発環境である。Java のすべての機能のほかに、ウェブカメラやマウスなどの各種入力機器を簡単に扱えるようになっており、デザイナーやアーティストによく利用されている。

Processing は一般的なプログラミング教育の入門時において利用しやすい開発環境である。プログラミング言語自体についての知識(Java の場合はクラスやメソッド main()の定義など)が必要なく、言語自体について詳しく学ばなくても、ある程度のアプリケーションを作成できるからである^[2]。

Processing はまた、グラフィックプログラミング教育の入門時においても利用しやすい開発環境である。あらかじめ用意されている基本的なグラフィック要素の描画、座標変換、カメラの操作などのための機能は、OpenGL のような標準的なライブラリと類似する形式になっているため、Processing で学ぶ作法が、後に本格的にグラフィックプログラミングを行う際にも生かされるからである^[3]。

筆者らは、青山学院大学理工学部情報テクノロジー学科の2年生後期に開講される演習科目「情報総合プログラミング実習 1」(受講者約 80 名)において、Processing を利用している。この演習は、C 言語についての学習を一通り修了した学生を対象に開講されるものであるが、Java のようなオブジェクト指向プログラミング言語と、グラフィックプログラミングについて学ぶことが目標の一部になっているため、先に述べたような理由から、Processing が向いているのである。

2.2. 演習スケジュール

90 分×2 コマ×15 週の演習のうち、90 分×4 コマを Processing を利用する演習に割り当てる。各コマの学習内容と課題は以下のとおりである。

1. **CG の基礎:**座標軸、色と塗り、2 次元図形、座標変換(平行移動と回転)を学ぶ。
課題:静止画を Flickr(2.4 節)にアップロードする。
2. **アニメーション:**アニメーションの作り方、ベクトルの操作、物体運動のシミュレーション、動画のキャプチャ方法、YouTube(2.4 節)へのアップロード方法を学ぶ。

課題:(1) アニメーション動画を作成し、YouTube にアップロードする。(2) 他の学習者の第 1 回の成果物を Flickr 上で閲覧・評価し、よいものを 3 件報

告する。

3. **3次元CG**: カメラの概念, 座標系, 座標変換, 変換のためのスタックの操作方法を学ぶ。

課題: (1) 3次元のアニメーション動画を作成し, YouTube にアップロードする。 (2) 他の学習者の第2回の成果物を YouTube 上で閲覧・評価し, よいものを3件報告する。

4. **インタラクション**: マウスの活用法, 操作履歴の取得法を学ぶ。

課題: (1) インタラクションを行うアプリケーションを作成, その動作する様子をキャプチャし, YouTube にアップロードする。 (2) 他の学習者の第3, 4回の成果物を閲覧・評価し, よいものを3件ずつ報告する。 (3) Processing を用いる演習や, 成果物をクラウド上で公開することについての感想文を書く。

2.3. 成果物の公開場所としてのクラウド

大学等の演習においては, 学習者は学習の成果をレポートとして提出するのが一般的である。レポートの提出方法としてよく採用されているのは, 紙面による提出や電子ファイルのメールでの送信, 授業管理システム (Course Management System, CMS) へのアップロードなどである。

本稿では, レポートの一部である成果物をクラウド上で公開することを提案する。この方法には, 以下に述べるような利点がある。

2.3.1. システム構築が容易

CMS のようなレポートを提出するためのシステムを用意する必要はない。もちろん, すでに CMS を利用している場合に, その利用を止める必要はない。筆者らの演習では最後の感想文の提出には CMS を利用している。

2.3.2. 情報発信技術の習得

情報をウェブ上で公開する技術 (情報発信技術) は, 今日の学生が身につけるべき情報リテラシーに不可欠なものである。演習科目の中で画像共有サイトや動画共有サイトを利用することによって, 学習者は静止画や動画の作成方法やそれらの公開方法を学べる。

2.3.3. 学習者間での閲覧・評価

クラウド上で公開された成果物は, 学習者同士が互いに閲覧し合える。これによって生まれる競争意識などによって, 学習のモチベーションが向上することが期待できる。また, 世界に向けて公開していること, 同じ課題に取り組んだ多くの学習者に見られるということが, ウェブ等からの剽窃を抑制する効果もあるだろう (ある学習者が参考にしたウェブサイトは, 他の学習者も閲覧している可能性が大きい)。

クラウド上で公開された成果物は, 学習者同士が互いに評価することもできる。クラウド上のサービスの多くには, そのための仕組みがあらかじめ用意されている (3.3 節)。学習者は, そのような仕組みの存在によって

学習のモチベーションを向上されることが期待される。教師は, どの学習者がどのくらい評価されたかや, 誰が誰を評価したかという情報を簡単に取得し (3.4 節), 教育方針や最終評価に反映させられる。

2.4. Flickr と YouTube

CG 教育の成果物の公開場所として, 静止画共有サービスである Flickr (<http://www.flickr.com/>) と動画共有サービスである YouTube (<http://www.youtube.com/>) を利用する。これら以外のサービスでも, 以下の要素を満たしていれば, 本稿で提案する手法のために利用できる。

- コンテンツに検索のためのキーワード (タグ) をつけられる。
- コンテンツを評価する仕組みが用意されている。
- Application Program Interface (API) が用意されている。

2.4.1. 提供される API

Flickr^[4] と YouTube^[5] の API のうちで, 本稿で提案する手法で利用するものを以下に挙げる (利用方法は 3.4 節で述べる)。

- Flickr の画像検索 (`flickr.photos.search`, <http://www.flickr.com/services/api/flickr.photos.search.html>): タグを指定して, 演習の成果物としてアップロードされた画像を一括取得する (取得できる画像の上限は 4,000 枚だが, 通常の運用ではこの制限は問題にならない)
- Flickr の画像への評価の取得 (`flickr.photos.getFavorites`, <http://www.flickr.com/services/api/flickr.photos.getFavorites.html>): 学習者同士の評価結果 (評価された回数と, 誰が誰を評価したかという情報) を取得する
- YouTube Data API の動画検索 (http://code.google.com/intl/ja/apis/youtube/reference.html#Searching_for_videos): タグを指定して, 演習の成果物としてアップロードされた動画と, その動画が再生・評価された回数を一括取得する

3. 手法

本章では, 成果物をクラウド上で公開し, それらを利用する方法を説明する。

3.1. 準備

教師は, 学習者が成果物を検索するためのタグと, 成果物を特定するためのタイトルの付け方を決め, 周知させる。タグは他で使われていないユニークなものにする必要がある。タイトルは学生番号などにするとよい。

学習者は, 各サービスを利用するために, ユーザ登録を行う^{注1, 2}。

3.1. 成果物の公開

学習者は以下の手順で成果物を公開する。

1. 静止画あるいは動画を作成する^{注3}。

2. ファイルをアップロードする。
3. タイトルを教師によって決められた形式に変更する (学生番号が望ましい)。
4. 教師によって決められたタグを付ける。
5. タグを使った検索で、自分がアップロードした成果物が現れることを確認する (Flickr では、登録が反映されるまでに時間がかかる場合がある^{注2)})。
6. 成果物についての説明を記述する (剽窃を疑われないために、参考にしたものをすべて記述するというルールを定めておくとよい)。

3.2. 成果物の閲覧

本節では、クラウド上で公開された成果物の閲覧方法を説明する。

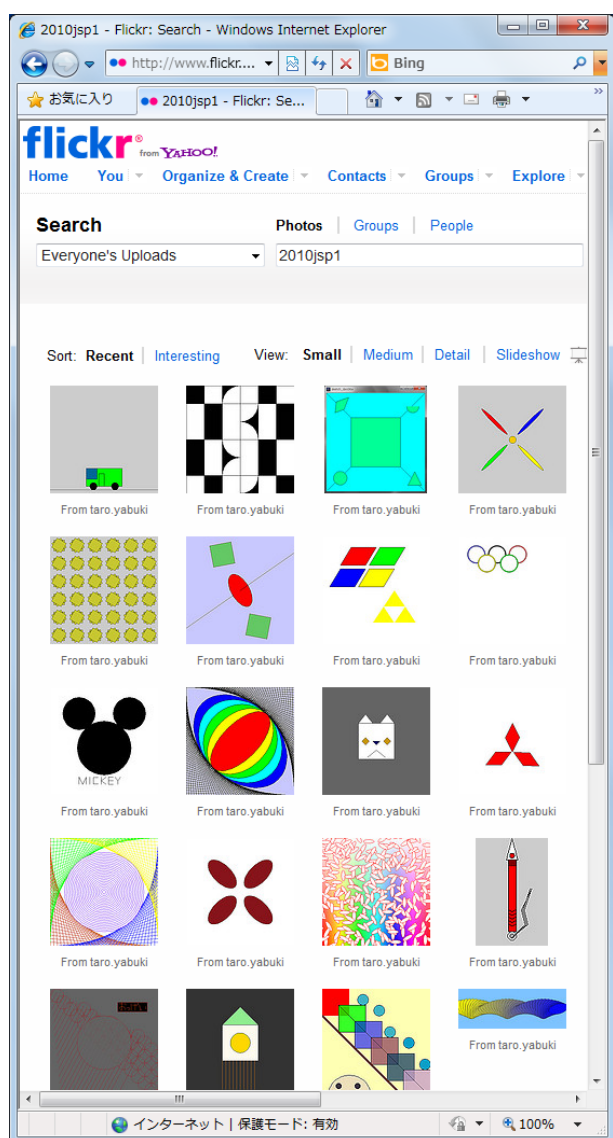


図1 Flickrで画像を検索し、一覧表示させる例

(この例に現れる画像は、CMSに蓄積された画像を筆者がまとめてアップロードしたものである。つまり、アップロード方法が本文の説明とは異なるが、その差異は本質的なものではない。図2のYouTubeの各動画は、学習者が自分でアップロードしたものである。)

3.2.1. 静止画(Flickr)

Flickr にアップロードされた静止画は、それらに付けられたタグを使って検索し、一覧表示させられる(図1)。

3.2.2. 動画(YouTube)

YouTube にアップロードされた動画は、それらに付けられたタグを使って検索し、一覧表示させられる(図2)。

3.3. 成果物の評価

本節では、Flickr や YouTube 上で、学習者同士が互いの成果物を評価し合う方法を説明する。

3.3.1. 静止画(Flickr)

検索結果を一覧表示した図1のサムネイル画像をクリックすると、対象画像だけを表示するページになる。画像を高く評価する際には、そのページの「☆Favorite」をクリックする。



図2 YouTubeで動画を検索し、一覧表示させる例

3.3.2. 動画(YouTube)

検索結果を一覧表示した図2のサムネイル画像をクリックすると、対象動画の再生画面になる。その画面の上の

インターフェースを使って動画を評価する。YouTube には 2 種類の評価方法がある。「評価する」と「お気に入り」に追加である(厳密に言えば、「評価する」の反対を意味する「評価しない」もある)。「評価する」は 1 クリックで手軽だが、誰が誰を評価したかというデータも取りたい場合には、「お気に入りに追加」を使うとよい。どちらを利用するかは事前に教師が決めておかなければならない。本稿では、「評価する」を利用する。

3.4. 評価結果の取得

成果物がクラウド上でどのように評価されたかは、ウェブブラウザ上で目で見確認することができる。しかし、成果物が多い場合には、そのような方法は現実的ではなくなる。そこで、2.4.1 項で紹介した API を利用して、評価結果を一括取得する方法を紹介する^{注 4}。

3.4.1. 静止画(Flickr)

Flickr では、以下の手順で画像への評価(評価回数と誰が評価したか)を取得できる。

1. 指定したタグの付いた画像を API ですべて取得する。
2. 一つ一つの画像に対して、「☆Favorite」を押した人のユーザ ID を API ですべて取得する。

この手順を実行し、結果を TSV(タブ区切りテキスト)形式で出力する PHP スクリプトは図 3 のようになる。

```
<?php
///?tag=値」で指定した画像の情報を TSV 形式で取得するスクリプト
$api_key = 'Flickr で取得した API_KEY をここに記入する';

if (!isset($_GET['tag'])) die();
$tag = urlencode($_GET['tag']);

header('Content-type: text/tab-separated-values; charset=UTF-8');
header("Content-Disposition: inline; filename=$tag.tsv");

function getFan($photo_id, $page, $people) {
    global $api_key;
    $result = simplexml_load_file("http://api.flickr.com/services/rest/
    .?method=flickr.photos.getFavorites&api_key=$api_key
    .&photo_id=$photo_id&per_page=50&page=$page");
    foreach ($result->photo->person as $person) {
        $people[] = $person['nsid'];
    }
    if ($page < $result->photo['pages'])
        return getFan($photo_id, $page + 1, $people);
    else
        return $people;
}

$page = 1;
do {
    $result = simplexml_load_file("http://api.flickr.com/services/rest/
    .?method=flickr.photos.search&api_key=$api_key&tags=$tag"
    .&sort=interestingness-desc&per_page=500&page=" . ($page++));
    $pages = $result->photos['pages'];
    foreach ($result->photos->photo as $photo) {
        $photo_id = $photo['id'];
        $owner = $photo['owner'];
        $title = $photo['title'];
        $fan = getFan($photo_id, 1, array());
        echo "$photo_id\t";
        echo "$owner\t";
        echo "$title\t";
        echo count($fan)."\t";
        echo implode(':', $fan)."\n";
    }
} while ($page <= $pages);
```

図 3 Flickr で特定のタグを付けられた画像への評価を調べる PHP スクリプト

3.4.2. 動画(YouTube)

YouTube では、指定したタグの付いた動画を API で検索すると、その動画が評価された回数(「評価する」が押された回数)も同時に取得できる^{注 5}。その結果を TSV 形式で出力する PHP スクリプトは図 4 のようになる。

```
<?php
///?tag=値」で指定した画像の情報を TSV 形式で取得するスクリプト
if (!isset($_GET['tag'])) die();
$tag = urlencode($_GET['tag']);

//header('Content-type: text/plain; charset=UTF-8');
header('Content-type: text/tab-separated-values; charset=UTF-8');
header("Content-Disposition: inline; filename=$tag.tsv");

require_once 'HTTP/Request.php';
$http_request = new HTTP_Request();

for ($startIndex = 1; true; $startIndex+=50) {
    $http_request->setUrl("http://gdata.youtube.com/feeds/api/videos?vq="
    . "$tag&alt=json&orderby=published&start-index={$startIndex}&max-results=50");
    $http_request->sendRequest();
    $results = json_decode($http_request->getResponseBody(), true);
    //print_r($results);
    if (!isset($results['feed']['entry'])) break;

    foreach ($results['feed']['entry'] as $entry) {
        //print_r($entry);
        $keyword = $entry['media$group']['media$keywords']['$t'];
        if (strcmp($keyword, $tag) == 0) {
            $average = $entry['gd$rating']['average'];
            $numRaters = $entry['gd$rating']['numRaters'];
            $likeCount = $average == 0 ? 0 : round($numRaters * ($average - 1) /
4.);

            echo $entry['media$group']['media$player'][0]['url'], "\t";
            echo $entry['title']['$t'], "\t";
            echo $entry['author'][0]['name']['$t'], "\t";
            echo "$average\t";
            echo "$numRaters\t";
            echo "$likeCount\t";
            echo $entry['yt$statistics']['favoriteCount'], "\t";
            echo $entry['yt$statistics']['viewCount'], "\t";
            echo $entry['media$group']['media$description']['$t'], "\n";
        }
    }
}
```

図 4 YouTube で特定のタグを付けられた動画への評価を調べる PHP スクリプト

4. 結果

2.2 節で述べた課題の成果物をクラウド上で公開し、学習者同士が評価し合った結果を示す。

4.1. 学習者同士の評価

評価された数のヒストグラムは図 5~8 のようになる。一部の成果物だけが評価され、大部分の成果物はほとんど評価されていないことがわかる。

4.2. 学習者同士の閲覧状況

動画の評価には、一覧表示で済む静止画と比べて評価者への負担が大きい。その負担を考慮して、2.2 節で述べたように、筆者らは学習者に、よいもの 3 件を報告することだけを要求し、すべての動画を見ることは要求しなかった。そのような条件のもとでは、各動画が再生された回数は図 9~11 のようになった。再生回数が少ない動画がかなりあることがわかる(学習者は約 80 人いるため、仮に全員が再生していれば、再生回数は 80 を超えるはずである)。

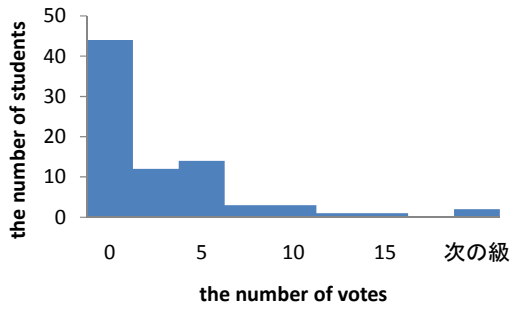


図5 課題1(CGの基礎)の成果物(80件)の評価数のヒストグラム(平均 2.2, 標準偏差 4.0)

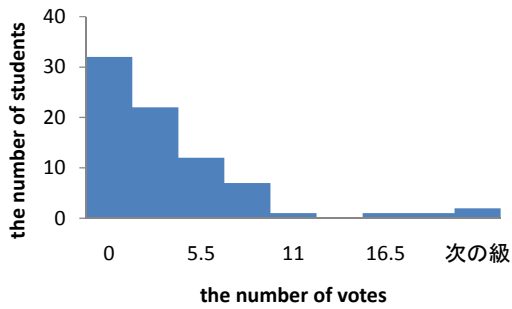


図6 課題2(アニメーション)の成果物(78件)の評価数のヒストグラム(平均 2.7, 標準偏差 4.5)

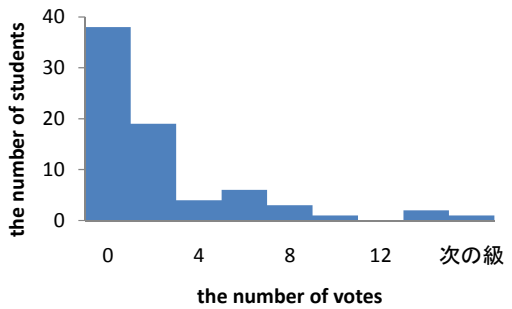


図7 課題3(3次元CG)の成果物(74件)の評価数のヒストグラム(平均 1.9, 標準偏差 3.4)

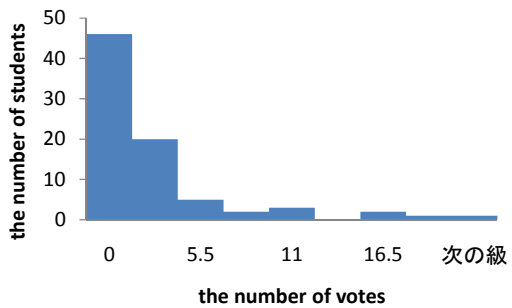


図8 課題3(3次元CG)の成果物(80件)の評価数のヒストグラム(平均 2.1, 標準偏差 4.4)

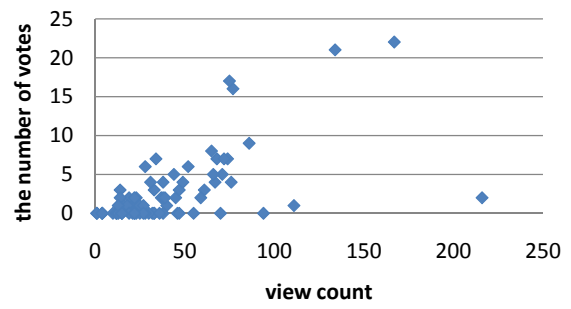


図9 課題2の成果物の再生回数と評価

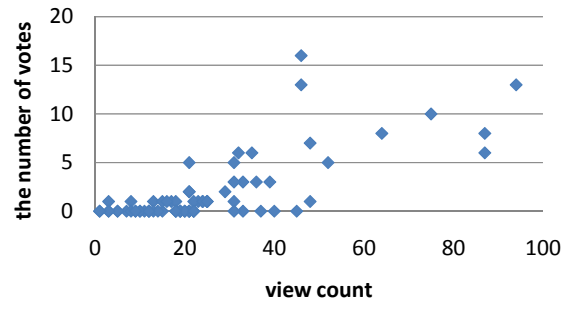


図10 課題3の成果物の再生回数と評価

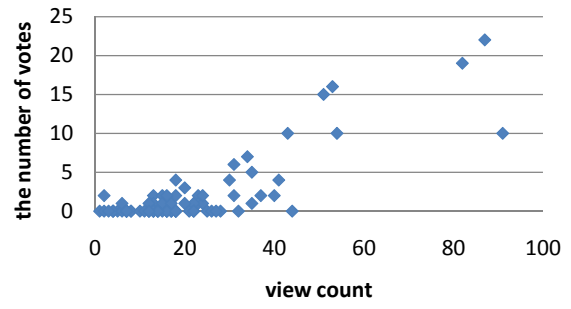


図11 課題4の成果物の再生回数と評価

4.3. 学習者からの感想

本稿で提案する手法に対して、学習者からは以下のような感想が聞かれた。

- Processing はシンプルでわかりやすかったプログラミングへの理解が深まった。
- 動画をキャプチャする方法がわかってよかった。
- YouTube へのアップロード方法を学べてよかった。
- 他人の作品を見られるという形態は面白い。
- 他人と評価し合うことで向上心が生まれる。
- 何をしているものなのかわからない動画があった。詳しい説明を書かせるべき。
- 友達同士で評価し合ったり、本質とは関係の無い見た目だけで評価が決まる恐れがある。
- 学習者同士の評価を最終成績に反映させるなら、もっと公平性を高める工夫が必要だ。
- ウェブで公開されているもののコピーがあった。
- 利用する素材の著作権が心配である。

5. 考察

本章では、本手法の利点と課題を考察する。

5.1. 利点

5.1.1. 必要な計算機資源

クラウドを利用する本手法には、成果物をアップロードするための CMS を用意する必要がないという利点がある。そのため、計算機資源を持たない教師でも、本手法を採用することができる。教師が管理する計算機と比べて、情報の消失や停電などの事故に遭遇する危険が小さいことも本手法の利点である。

5.1.2. 情報発信技術の習得

本手法を通じて、学習者は静止画や動画などのコンテンツの作り方や、クラウド上での公開方法といった、今日の情報リテラシーにおいて重要な情報発信技術を習得できる。このことは 4.3 節で紹介した学習者の感想でも確認できる。

5.1.3. 学習者のモチベーションの向上

クラウド上で公開された、他の学習者の成果物を閲覧したり評価したりすることは、学習者のモチベーションの向上に大きく貢献する。このことは 4.3 節で紹介した学習者の感想でも確認できる。

5.1.4. 教師のモチベーション向上

今日の大学は、教育の内容や成果が、研究成果に比べて外からは見えにくくなっている。本稿で述べたような方法によって、教育の内容と成果が世界に向けて公開されるようになれば、教師の責任感は増し、それに伴ってよりよい教育を行おうというモチベーションも向上することが期待される。

5.2. 課題

本手法には以下のような課題がある。

- 他の学習者の動画を見て評価することは、評価者にとってかなりの負担になる。評価対象を振り分けて、効率よく評価する仕組みが必要である。
- Flickr や YouTube には、閲覧回数の高いものや評価の高いものから順に表示させる機能がある（先に表示されたものはまた評価されるという生のフィードバックがかかる。4.2 節の結果でも、そのような傾向が見て取れる）。このような機能は不公平を生む原因になるため、本手法を採用する際には利用すべきではない。
- レポートを公開することに伴うリスクは、組織内だけで処理する場合に比べて大きい。しかし、そのリスクの大きさと、それをチェックする目の数の多さを意識すれば、剽窃はしにくくなるであろう。とはいえ、参考にしたものはすべてコメントとして記載するなどのルールを定める必要はある。

6. 結論

CG を題材としたプログラミング教育において、その成果物をクラウド上のサービスで公開し、学習者同士が互

いに評価し合う方法を提案した。教師は、サービスが提供する API を利用して、評価結果を自動的に取得できる。本手法によって、教師の負担は軽減され、学習者の情報発信技術、学習のモチベーションは向上する。

計算資源を手元に置く時代は終わり、クラウドを利用する時代になりつつある。本稿では提案したのは、静止画・動画共有サービスというクラウドの一形態の活用法である。クラウドには他にもさまざまな形態があり、それらを活用することで、従来は実現困難だった教育手法が、容易に実現できるようになるだろう。

注

- 注1 大学の演習室ではユーザ登録に失敗するという事例が報告されている。そのような事態に備えて、あらかじめ自宅などでユーザ登録を済ませておくとういだろう。
- 注2 Flickr は、画像を 5 枚アップロードして数日経ってからでないと、タグを指定しての検索結果にコンテンツが掲載されない仕様になっている (<http://www.flickr.com/help/tags/#2855CT>)。そのため、演習の早い段階で、学習者は 5 枚以上の画像をアップロードしておくことが望ましい。これとは別に、成果物が検索にかからないようにする設定をオフにしておくことも必須である。
- 注3 Windows 環境では、静止画の作成には組み込みの Snipping Tool、動画の作成には Microsoft Expression Encoder 4 Screen Capture を使うのが簡単である。
- 注4 スクリプトの実行環境は、筆者らがウェブ上で公開する予定である。そのため、本稿で紹介する手法を実践しようとする教師が、これらのスクリプトの実行環境を用意する必要はない。本稿にスクリプトを掲載しているのは、提案手法の再現性を担保するためである。
- 注5 誰が評価したかはわからない。学習者以外が評価している可能性もある。そのような可能性を排除したい場合は、「評価する」ではなく「お気に入り追加」を利用するとよい。

参考文献

- [1] Processing.org. <http://processing.org/>
- [2] 田中孝太郎, 前川峻志. *Built with Processing*. ビー・エヌ・エヌ新社, 2010.
- [3] 菊池誠. Processing によるプログラミング教育. 情報処理, Vol. 52, No. 2, pp. 213–215.
- [4] Yahoo! Inc. The App Garden. <http://www.flickr.com/services/api/>
- [5] Google, Inc. YouTube の API とツール. <http://code.google.com/intl/ja/apis/youtube/overview.html>

著者紹介

やぶき たろう : 青山学院大学理工学部情報テクノロジー学科, 252-5258, 神奈川県相模原市中央区淵野辺 5-10-1 (taro.yabuki@unfindable.net)

さくた ひろし : 青山学院大学理工学部情報テクノロジー学科, 252-5258, 神奈川県相模原市中央区淵野辺 5-10-1