

contents

[コラム]

リフレッシュするのは受講者だけではなく—
ジョーシン 2014 秋での議論を受けて—
…鈴木 貢

[解説]

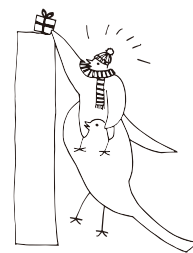
プログラミング学習環境の自作
PenFlowchart によるプログラミング学習
…中西 渉

[解説]

情報化社会における創造力をどう育
むか？『ビスケット塾』の試み
…小林桂子 渡辺勇士

 Column

リフレッシュするのは受講者だけではなく—ジョーシン 2014 秋での議論を受けて—



去る 2014 年 10 月 25 日に、テーマを「情報教科教員としてのスキルアップ—教員免許更新講習などを活用して」として、「高校教科『情報』シンポジウム 2014 秋」（愛称 ジョーシン）が開催された。シンポジウムでは、8 月 4 日から 6 日にかけて本会が実施した教員免許更新講習の報告に始まり、情報教育を担う教員の生涯教育について多面的な議論が活発に行われた。実施の経緯については、ぺた語義の解説¹⁾やコラム²⁾を、シンポジウムの内容については本会の初中等教育委員会(<http://sigps.eplang.jp/>)の本シンポジウムの Web ページをご覧ください。ここではシンポジウムでの議論を通じて考えたことを書かせていただきたい。

小職にとって教員免許更新講習（以下更新講習と略）は数年に 1 度の「おしごと」で、上から目線で一方通行の大学生向け授業のサブセットと、確実に合格点が出るテストでごまかしていたが、シンポジウムを聴講して、改めなければならぬと確信した。講師としてお手伝いいただいた先生から、講習を通して先生自身もリフレッシュしたという感想があったが、小職のようなりエゾンではなく、自分が興味を抱ける内容を、分かりやすさに重点を置いて指導されていたことが資料集から読み取れる。

現行の更新講習で情報は「教科指導、生徒指導その他教育の充実に関する事項」の選択領域であるが、「免許状更新講習における選択必修領域の導入について」では、2016 年以降は「教育の情報化（情報通信技術を利用した指導）および情報教育（情報モラルを含む）等」の選択必修となる。^{たが} 箍がきつくなるとするのは小職だけであろうか。

2013 年 6 月の閣議決定「世界最先端 IT 国家設立宣言」や、上記の選択必修化が示すように、情報教育のプレゼンスはますます高まっている。しかし、ぺた語義で繰り返し指摘されているように、初等中等学校でのお寒い情報教育は、情報系学部・学科が優秀な学生を確保する際の障害の 1 つと考えられる。確かに、更新講習は、社会への小窓に過ぎないかもしれないが、個々の大学教員の興味を社会に問う機会の 1 つと捉え、受講者と「萌え」を共有することは、講師も自分のリフレッシュにもつながるのではないか。

最後に、2015 年に開催される 2 つの高校教科「情報」シンポジウム（5 月中旬 大阪工業大学うめきたナレッジセンターと、10 月下旬 早稲田大学（予定））のご案内で締めくりたい。詳しくは上記の初中等教育委員会の Web ページをご覧ください。

参考文献

- 1) 中野由章, 中山泰一: 高等学校情報科教員の現状—その問題点と我々にできること—, 情報処理, Vol.55, No.8, pp.872-875 (Aug. 2014).
- 2) 松原 仁: 教員免許更新講習, 情報処理, Vol.55, No.12, p.1399 (Dec. 2014).

鈴木 貢 (高校教科「情報」シンポジウム 2014 秋 実行委員長/島根大学)

ロゴデザイン ● 中田 恵 ページデザイン・イラスト ● 久野 未結

プログラミング学習環境の自作 PenFlowchart によるプログラミング学習

中西 渉

名古屋高等学校

開発に至る経緯

筆者の勤務校である名古屋高等学校は、学年 12 クラス規模の私立男子校である。教科「情報」の授業が始まった当初は、プログラミング教育に適したソフトウェアを模索していた。これは、情報教室の端末を Linux にしたため、Excel の VBA などは使えず、最初は OpenOffice.org Calc のマクロを用いたが、手順が面倒であることなどから生徒の反応が非常に悪かったためである。

筆者は 2006 年に行われた「教育用プログラミング言語に関するワークショップ 2006」に参加し、PEN¹⁾ という初学者向けプログラミング学習環境の存在を知った。PEN の実行画面を図-1 に示す。使われているプログラミング言語は、大学入試センター試験の情報関係基礎の出題に用いている手順記述言語 DNCL を拡張したものである。日本語ベースということでそのままでは入力が面倒になるため、構文や命令を簡単に挿入できる入力支援ボタンが用意されている。Java で開発されているので、実行環境の OS を問わないことがありがたかった。さっそく 2006 年度から勤務校の授業は PEN に切り替えた。PEN は実行中の変数の値表示や、実行速度の調整、1 行ずつの実行など、動作確認やデバッグに適した機能が充実しており、生徒たちがプログラムを作って実行することのハードルは下がった。

しかしプログラムが複雑になるにつれて、エラーに悩む生徒が増えてきた。PEN では本来なら、入

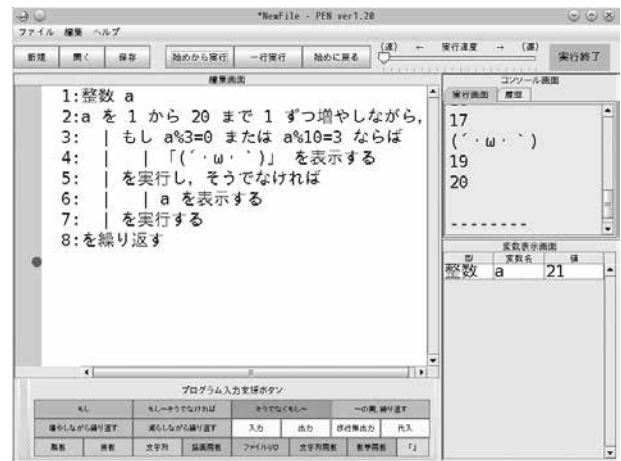


図-1 PEN の実行画面

力支援ボタンを正しく使っていれば構文エラーは発生しないはずである。しかし彼らは図-2 のような正しいプログラムを、わざわざ図-3 のように書きなおしてしまうことがある。ほかのプログラミング言語を知っている我々は「もし～を実行する」が「IF～END IF」のブロック構造に相当することが直感的に分かるが、生徒は日本語としての自然さを優先してこのようにしてしまふと考えられる。もちろん実行時にはエラーメッセージが表示されるのだが、そのメッセージを見て「『を実行する』を削除したのがいけなかった」と気付くことができる者はそもそもその行を削除したりしないのではないだろうか。

そこで構造を保ったプログラムを生成する仕組みがあればこのような問題がなくなると考え、フローチャートを作ることで PEN のプログラムを生成する PenFlowchart を 2011 年に開発した²⁾。図-4 の

もし $a=0$ ならば
| 「ゼロ」を表示する
を実行する

図-2 正しいプログラム

もし $a=0$ ならば
「ゼロ」を表示する

図-3 間違ったプログラム

ように、画面上部のパーツをフローチャートにドラッグ&ドロップすることで、PEN用のコードを生成することができる。実行やデバッグはPENの機能をそのまま利用する。

コードからフローチャートに書き戻す機能も実装したが、これは松澤芳昭氏らのBlock Editor³⁾がJavaのコードからブロックに変換する機能を持っていることに影響を受けたものである。なお、フローチャートからコードへの変換は自動的に行われるが、逆は明示的にボタンを押さないと変換しない。これは書いている途中のコードを解析しても構文エラーになるからである。

作成したフローチャートはEPSまたはPNGフォーマットで画像として出力できるため、プリントやスライド、テスト問題を作るときに活用している。

アルゴリズムの学習において、フローチャートが最善のものだと考えているわけではない。PADのような構造化チャートを使うことも検討したが、高校の教科書はすべてフローチャートを用いているのでそれは適切でないと考えた。また、フローチャートは自由に線がひけるためスパゲティ化してしまうという批判があるが、PenFlowchartではパーツの加除しかできないので構造は保たれる。

PenFlowchartを用いることによって、前述したようなブロック破壊による構文エラーはなくなり、実習はスムーズに進むようになった。図によるプログラミングを行うことで、フローチャートは分かるがコードが分からない生徒を生むことになるのを危惧したが、期末テストの得点を分析したところ、そ

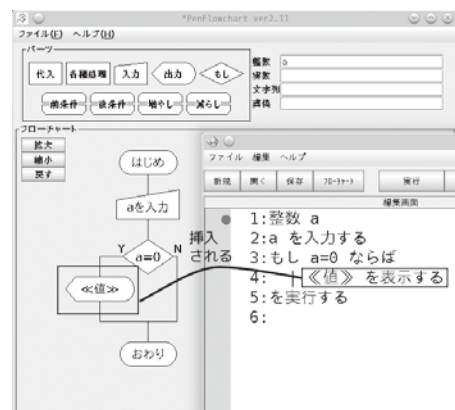
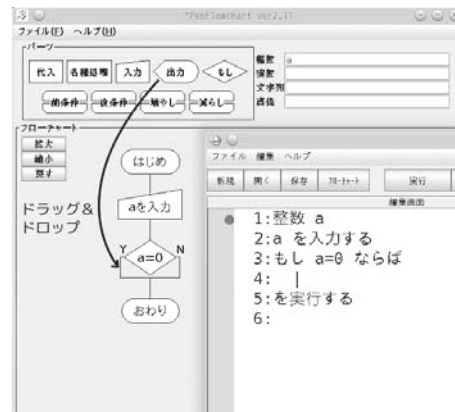


図-4 PenFlowchart の実行画面

れ以前よりもフローチャートとコードの関係についての理解がよくなってきているという結果が得られた⁴⁾。

当初予想していなかった副作用もある。ほぼ毎時間の授業で、課題のプログラムをメールで提出させていたのだが、PENの頃は隣の生徒の画面を見ながら、入力支援ボタンを使わずに手で書き写す生徒もいた（必然的に構文エラーのまま提出する者も多い）。その理由は、前述したブロック構造を理解しない者が入力支援ボタンを使うと、予想外の形で構文が挿入されてしまうため、かえって面倒を感じることにあるらしい。しかしPenFlowchartでその通りのプログラムを作るには結局フローチャートを自分で完成させなくてはいけない。その結果、教えてもらいながらも自力でパーツを配置してフローチャートを作成するようになった。PenFlowchartにおけるパーツの追加は、PENにおける入力支援ボタンに対応しているため、この訓練がブロック構造の意識につながったのではないかと推測する。

いくつかの大学や専門学校でも PenFlowchart を

	DNCL	BASIC
構文	《変数》を入力する	INPUT《変数》
定義	<pre><INPUT : "を入力する"> InputStat(): {} {Ident() <INPUT>}</pre>	<pre><INPUT : " INPUT"> InputStat(): {} {<INPUT> Ident() }</pre>

図-5 構文定義ファイル

使用していただいた。学生や教員からの要望を教へていただくことで、機能追加や改善を行った。たとえばすでに作ったプログラムをループに入れて繰り返し実行したいというとき、以前は外側のループの中ですべてを作りなおさなくてははいけなかったの、カット & ペーストでパーツを移動できるようにしたことなどである。

DNCL 以外の言語

筆者の授業は自作プリントを中心に行ってきたので DNCL でも支障なかったが、新課程の情報の教科書で DNCL を使っているものは皆無であるため、PenFlowchart を使うなら教科書は使えないという状況が今後も続くことになる。そこで他言語への移植を試みた。

まず最初に C++ のコードを生成するものを作ったが、授業で使うとなるとコンパイラなどを用意して使い方を教へなくてははいけないので、とても現実的に使えるようなものではなかった。また、コードからフローチャートを生成するには構文の解釈が必要となるが、C++ のパーサを書くのは筆者には困難であるため断念した。

続いて、BASIC 版に着手した。これは、PEN は構文解析部分を JavaCC で作っているの、図-5 のように書き換えれば容易に BASIC 版になるからである。その「翻訳」さえ行えば、すでにある DNCL の実行環境を利用することで、プログラムの実行やデバッグを容易に行うことができる。

このようにして BASIC 用の PenFlowchart を作成したが、実はこのような BASIC を用いている

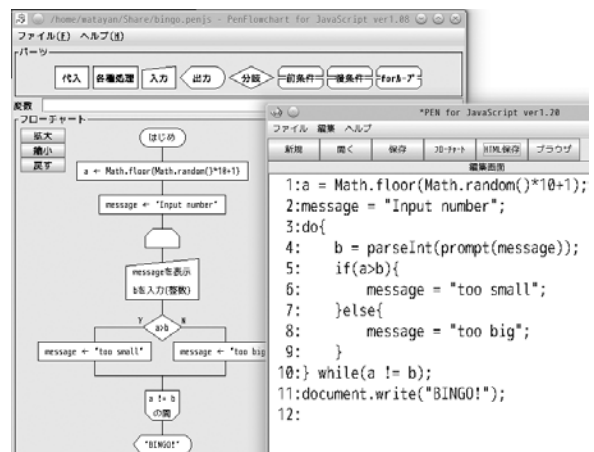


図-6 PenFlowchart for JavaScript の実行画面

高校の「情報」の教科書はない。実教出版の「情報の科学」では表計算の VBA を用いているが、セルを配列とみなすという面白いアイディアを採用しているため、教科書のプログラムをこの BASIC 版で実行することはできない。そこで JavaScript 版を作ることにした。これにより JavaScript のプログラムを扱う日本文教出版や東京書籍の教科書に対応できる。

JavaScript の実行環境を内製することは（筆者にとっては）容易ではないので、PEN からプログラム実行に関する機能を取り去り、生成されたプログラムを埋め込んだ HTML ファイルを出力し、ブラウザに読み込ませて実行する機能を追加した。実行画面を図-6 に示す。プログラムを書き換えたときには再度 HTML ファイルを出力してブラウザでリロードしなくてはならないなど、余計な手数が増えることで生徒が混乱することを心配したが、今年度（2014 年度）の授業で使ってみた様子では手順として覚えてしまえば大したことではないと捉えているようである。

自由な環境

勤務校では今後の授業で JavaScript 版を使っていくことになるが、開発についてはいろいろ迷いがある。たとえば、フローチャートが足枷になっていると感じられることがある。JavaScript で普通に

用いられる else if などをフローチャートで表現するのは難しいため、正しい構文を使ってもフローチャートへの変換ができないことがあるのだ。ソースコードを自由に書けるようになった者にとっては、これが不自由な制限に思われることが予想される。高校の授業で教科書通りのプログラムを扱うにはフローチャートが使えるものでいいかもしれないが、それ以上のことを望む場合にはフローチャートを使えないようにしたものを用意すべきであろう。フローチャートを使えなくした環境でも HTML のタグを自動的に埋め込んだり、PEN の入力支援ボタンを使ったりできるという点で、テキストエディタでの作業よりも楽になると考えられる。

フローチャートを使わない環境を作ることは、すでにある機能をただ停止すればいいので容易である。PenFlowchart は GPL (GNU General Public License) で公開しているので、ほかの利用者がそのような変更を加え、それを再配布することも自由である。これは PEN が GPL で公開していたことに由来し、それにより PenFlowchart やほかの言語版を作ることができた。

多くの情報教室で授業支援ツールが用いられているように、授業で使う環境やツールはカスタマイズが必要になることが多い。したがって、そこで用いられるソフトウェアは自由であることが望ましいと考えている。ソースコードが手に入るなら、それを

ほんの数行変更することで、自分の望む環境が得られることもある。勤務校の情報教室の端末を Linux にしたのは、そのような自由を求めてのことであった。授業で使うソフトウェアをゼロから作り上げることは難しいにしても、既存ソフトウェアのカスタマイズや機能の追加・削除を現場の教員が行うことは不可能ではないし、他人のソースコードを読んで理解するのはいい勉強にもなる。そのように自分で手を動かして不満を解消するような教員が増えることを願っているし、筆者自身もそうありたい。

なお、PenFlowchart とその他言語版は筆者のサイト <http://watayan.net/prog/> で配布している。

参考文献

- 1) 西田知博, 原田 章, 中村亮太, 宮本友介, 松浦敏雄: 初学者用プログラミング学習環境 PEN の実装と評価, 情報処理学会論文誌, Vol.48, No.8 (Aug. 2007).
- 2) 中西 渉: Penflowchart の開発, 情報処理学会研究報告 コンピュータと教育(CE), Vol.2012-CE-113, Vol.13 (Jan. 2012).
- 3) 松澤芳昭, 酒井三四郎: ビジュアル型言語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 コンピュータと教育 (CE), Vol.2013-CE-119, Vol.2 (Mar. 2013).
- 4) 中西 渉, 辰己丈夫, 西田知博: Penflowchart によるプログラミング導入教育の評価, 情報処理学会研究報告 コンピュータと教育(CE), Vol.2012-CE-121, Vol.9 (Oct. 2013).

(2014年11月30日受付)

中西 渉 (正会員) watayan@meigaku.ac.jp
名古屋高等学校教諭。担当教科は情報と数学。コンピュータと教育研究会運営委員。初等中等教育委員。

情報化社会における創造力をどう育むか？ 『ビスケット塾』の試み

小林桂子

特定非営利活動法人デジタルポケット

渡辺勇士

特定非営利活動法人デジタルポケット

「プログラミングで科学と芸術をつなぐ」

この言葉は、プログラミングスクール「ビスケット塾」を説明するリーフレットに使用しているものである。筆者らは、これからコンピュータを学ぶ子供たちに、プログラミングの技術やコンピュータの知識を身に付けることだけでなく、プログラミングを媒介に物事を捉えることが必要だと考えた。世界を「科学の目」で観察し、「芸術の心」で創作することが今後ますます重要になるだろう。これからの社会を担う子供たちにそういった経験を積んでもらいたい、そんな意図が込められている。

「ビスケット塾」の設立とその目的

「ビスケット塾」は、2013年3月にサービスを開始した。「プログラミングで思い通りに絵や動きをコントロールし、作品化する」ことを重視したレッスンを実施している。

運営する特定非営利活動法人デジタルポケットは、「情報化社会に生きるすべての人々の情報リテラシーの向上」を目的に2011年11月に設立された団体である。ここで言う情報リテラシーは、単なるデジタル機器を使いこなす能力ではなく、プログラミングなどのコンピュータの本質に触れて創作ができる能力を指している。

現代の社会は、コンピュータを使って創造性を発揮できる人材を求めており、その能力を伸ばす

場として、2000年代、多くの大学や専門学校で情報のデザインや表現を扱う新しい学部・学科が立ち上げられている。デジタルポケットは、それまでワークショップで活用していたプログラミング言語「Viscuit」を利用し、小学生を主な対象にした、継続したプログラミングのレッスンを通じて「コンピュータ上でクリエイティビティを発揮する能力」を育もうとした。

「ビスケット塾」は小学生以上を対象とし、毎週60分または隔週120分のレッスンを実施する。これまで実施した「ワークショップ」の手法を応用し、複数の受講生が相互にコミュニケーションをとりながら、制作に取り組む(図-1)。

なぜ Viscuit を使うのか？

「ビスケット塾」で利用している言語「Viscuit」は、



図-1 「ビスケット塾」レッスンの様子



図-2 未就学児童がプログラミングを楽しむ様子

NTT コミュニケーション科学基礎研究所原田康徳氏が開発したビジュアルプログラミング言語である。Viscuit は、絵を組み合わせてプログラムを作るため、タイピング技術や、算数、英語の知識を必要としない。また、Viscuit のユニークな描画インタフェースで、絵が苦手な子供も取り組みやすい。そのため、筆者らが実施しているワークショップでは、4 歳以上の未就学児童が、非常に短い時間で十分に論理を理解し、自由に絵を描き、プログラミングを楽しむ姿を見ることができる(図-2)。

また、Viscuit は、レッスン内容に応じて環境を調整することができる。アニメーションづくり・ゲームづくり・絵本づくり・作曲といった、レッスンの目的と難易度に合わせてインタフェースを設定でき、レッスンを段階的に進めることができる。

「楽しくつくる」ために —ワークショップのカリキュラム化

筆者らは、開発者の原田氏の協力のもと、法人設立時より数多くのプログラミングワークショップを実施してきた。

ワークショップを実施する際には、「参加者それぞれが自分なりに理解し、楽しめる」ことを目的としていたため、参加者は、1 回のワークショップを受け、「楽しかった」だけで、プログラミングに興味を持たないことも多かった。では、興味を持ってもらうために何ができるか。そこで、このワークショップに段階を設け、カリキュラム化し継続する

動きを作る	「海の世界」を作る しゃくとり虫, 歩く人を作る ゆらゆらする動きを作る
衝突判定	ぶつかったら絵が変わる ロボット大戦 卵を産むニワトリ 風邪がうつる様子を作る
インタラクション	ボタンを押す 動きをコントロールする シューティングゲームを作る
表現方法	遠近感のある風景を作る 動く抽象画を作る
音	動きとリズムを作る オルゴールを作る(作曲)
絵本	ルールに従って場面を作る みんなで1つの絵本を作る オリジナル絵本を作る

表-1 カリキュラムの例

ことで目的に近づけるのではないかと考えた。

小学校に併設される放課後子供教室¹⁾などで、複数回のワークショップを同じ児童が受講する機会を作り、どのような変化があるか実験をしてみたところ、継続してワークショップを受講すると、プログラミングの理解度が上がるだけでなく、表現の多様さ・精緻さも向上することが分かり、「ビスケット塾」のサービスへとつながった。

一方、継続するためには「面白い」「やってみたい」と受講生が思うようなコンテンツが必要である。筆者らは、ワークショップの経験から、段階的にプログラムの複雑さを上げながら、作品の課題を考え、原田氏の監修のもと、カリキュラム化を行った。表-1 はカリキュラムの例である。

「楽しく理解する」ために —設計図をつくる

Viscuit のプログラミングは、操作が非常に直感的なため、「動作する理由」が分からなくても、適当に絵を配置すれば動いてしまう。ワークショップの場合はそれでもよかったが、レッスンとして課題に取り組むためには、プログラミングの前に受講生が構想を考え、形にするプロセスが必要であると考えた。

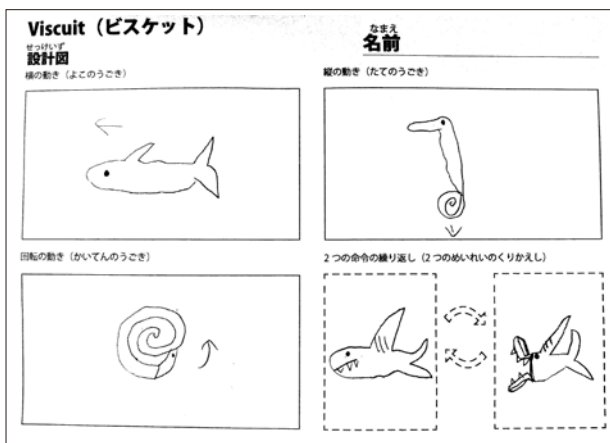


図-3 設計図の例

そのために、受講生が、これからどのような絵をどのように動かすつもりなのかを記述する「設計図」用紙を用意した。この「設計図」にはレッスンの内容に合わせてさまざまなフォーマットがある。

「設計図」を描くことで、「やろうと思ったことが実現できたか」を講師と受講生が相互に確認できる(図-3)。

ビスケット塾のレッスンの流れ

「ビスケット塾」のレッスンは、受講生自身が考え、手を動かして作ることを重視しているので、レッスン時間の大部分を制作にあてている(60分のレッスンの場合、制作時間が40分程度)(図-4, 5)。

- 1) 課題を示し、例題で練習をする
- 2) 設計図を描く
- 3) プログラミングして作品を制作する
- 4) 自分が、何をどのように作ったかを発表会で説明する

「継続したレッスン」を実施して

レッスンを継続して受講した子供たちには、受講し始めた頃と比較して、共通して以下のような様子の変化が見られた。

- 作品制作を通じて、自分の考えに自信を持つ。
- レッスンに集中できるようになる。
- 苦手なスキル(絵を描く、アイデアを出す、説



図-4 発表会の様子

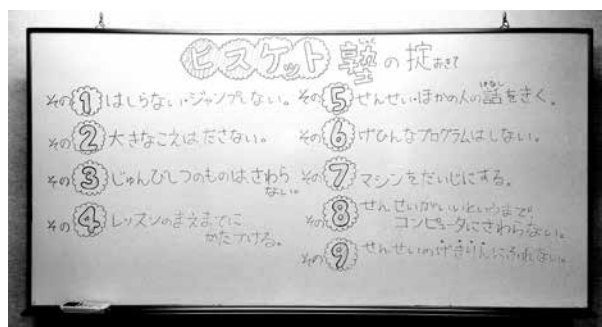


図-5 受講生と一緒に作った「ビスケット塾のおきて」

明する)にも取り組もうとする。

- ほかの人の作品を観察し、良いところを自発的に取り入れ、絵やプログラムが高度になる。
- 自分だけが面白いものよりも、人が見て面白いと思う作品を目指す。
- 自分のやり方(アイデアの出し方、絵のテイスト、プログラムの作り方)ができてくる。
- 「できない」ことにくやしさを感じる。

また、2014年10月には、プログラミングスクール TENTO²⁾と合同で「プレゼン大会」を実施した。通常のレッスンでは、受講生は、多くの人の前で発表する機会はなかったので、作品制作に加えて発表も練習して臨んだ。筆者らの心配をよそに、小学校1・2年生を含めた受講生たちが、非常に堂々と自分の作品を説明し、質問にもきちんと回答する姿に驚かされた(図-6, 7)。

今後の課題と展望

「ビスケット塾」を約1年半運営してきて、課題も

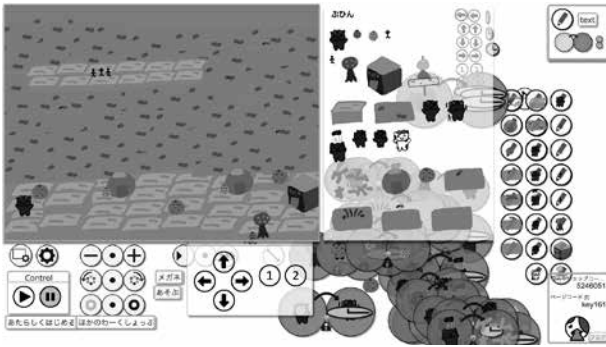


図-6 プレゼン大会「ビスケット部門最優秀賞」を受賞した作品「ニャ島のコンビニ」。奥行き感のある画面，ゲームのルールと難易度，キャラクターの造形のバランスが良い

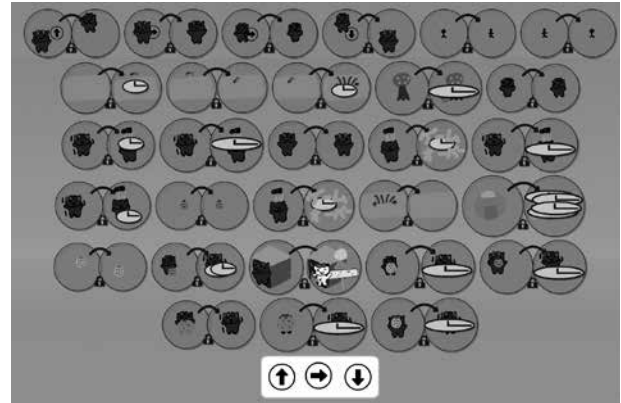


図7 「ニャ島のコンビニ」のプログラム（メガネ）。キャラクターの動作が精密に作られているのが分かる

いくつか見えてきた。

- 最初に紹介した「プログラミングで科学と芸術をつなぐ」という言葉は、ICTにかかわったことのある保護者は理解できるが、そうでない保護者にとっては意味が分からない。この塾の目的を別の言葉や方法でも示す必要がある。
- 特に低学年の場合、課題が複雑になると、設計図を作るのが難しくなり、拒否反応を示す受講生もいる。
- Viscuitで思い通りのものが作れるようになった後はどうするか。同じスペースで実施しているプログラミングスクールTENTOに編入し、大人が使うものと同じテキストの言語に移行できる、というルートを作ったが、まだ移行した受講生はいない。どのような進路があるのか、子供の様子をよく観察して考える必要がある。

一方、当初は小学校低学年の子供が受講生になると想定してレッスンを実施していたが、4・5年生の受講が増えてきている。高学年の子供にも十分に楽しく取り組める内容であることが分かった。

「ビスケット塾」を見学・体験した方々は皆、「これは実際に見て体験してみないと、良さが分からない

いですね」と言う。興味を持った方は、ぜひ見学にいらしてほしい。筆者らは、情報化社会を生きるすべての子供たちに、「ビスケット塾」のカリキュラムを通じて「コンピュータを思い通りに動かして、自分の思い通りのものをつくる」経験をしてほしいと考えている。本稿がそのひろがりのきっかけになることを願っている。

参考文献

- 1) 原田康徳, 勝沼奈緒実, 久野 靖: 公立小学校の課外活動における非専門家によるプログラミング教育, 情報処理学会論文誌, Vol.55, No.8, pp.1765-1777 (Aug. 2014).
- 2) 草野真一: プログラミングスクールTENTOの冒険, 情報処理, Vol.54, No.9, pp.948-951 (Sep. 2013).

(2014年11月30日受付)

小林桂子 koba@digitalpocket.org

特定非営利活動法人デジタルポケット代表理事。岐阜県立国際情報科学芸術アカデミー (IAMAS) を卒業後、同校に勤務しながら行ったデジタルメディアを活用したワークショップをきっかけにデジタルポケットを設立。

渡辺勇士 watanabe@digitalpocket.org

特定非営利活動法人デジタルポケット理事。明治大学卒業後、繊維商社に勤務するが演劇のワークショップとの出会いをきっかけに退職。ワークショップデザイナープログラム履修をきっかけに青山学院大学社会情報学研究科にて修士号を取得。

デジタルポケット Web サイト: <http://www.digitalpocket.org>