

# 生徒・児童によるプログラミング —ビスケットの挑戦—



原田康徳

NTT コミュニケーション科学基礎研究所

## デジタル機器を使用するすべての人に プログラミングを教えよ

コンピュータは普通ではない性質を持っている。我々が普通に使っている鍋とかハサミといった道具とも違うし、人間の助手や幼児やチンパンジーとも違う。この性質は、一般人が普通に生活して成長してゆく過程のさまざまな経験から自然に理解できるようなものではない。他の経験から一切類推できない。だから普通ではないのである。

一方、コンピュータはその利便性から、どんどん生活の中に入り込んできている。iPadのような「使いやすいコンピュータ」は、普通ではない性質を隠しながら、一方で便利な性質はできるだけ残すように、非常に苦勞して設計されている。時代は、このままコンピュータの基本を隠しながら流れようとしている。このままでよいのだろうか。

子どもの頃に積み木で遊んだ人は多いであろう。積み木を高く積んでは、崩れるのを楽しむ。このような遊びのおかげで、力学に関する直観が身に付く。もし、子どもの頃に積み木遊びを一切せずに大人になったらどうなるだろう。荷物をどれくらい積み重ねると崩れるか、といった直観は持っていない。その人に物置の管理をさせる場合「荷物は4段までしか積んではいけません。しかし小さな荷物なら5段でもいいです」といったルールを大量に覚えて使うしかないだろう。基本的な直観を持たずに大量のルールで行動を規制する、現在のデジタル機器を使うために大量のルールが必要なのはこれが理由である。

コンピュータの本質を理解するには、プログラミングを経験するのが近道と考える。プログラミングの過程で生じるさまざまなことの積み重ねで、コンピュータの特異な性質を理解していくのである。ただし、ここでは役に立つものを作るためのプログラミングではなく、積み木遊びと同じレベルのプログラミング遊びでよい。プログラミングで遊びながら、コンピュータが持っている普通ではない性質に対して直観が身に付くのではないだろうか。

プログラミング遊びは、従来のプログラミング教育とも違う。高校で物理を選択しないような専門家を目指さない人も、子どもの頃に積み木遊びくらいはやっている。このレベルの場を提供するのがプログラミング遊びである。

プログラミング遊びを実践している子どもたちがいる。墨田区の緑小学校の放課後事業「みどりっ子クラブ」の子どもたちである。2010年の春からビスケットを使い始め、2011年には非常に安定した活動に進化している。プログラミング遊びの効果が具体的にどのように現れているのかは、今後の調査研究を待たなくてはならないが、本稿では、具体的にどのように活動が進められ、日ごろどうということが起こっているかを紹介する。

## ビスケット

ビスケット<sup>1)</sup>は、プログラミングの楽しさと可能性を数学や英語を使わずに誰でも直観的に理解でき

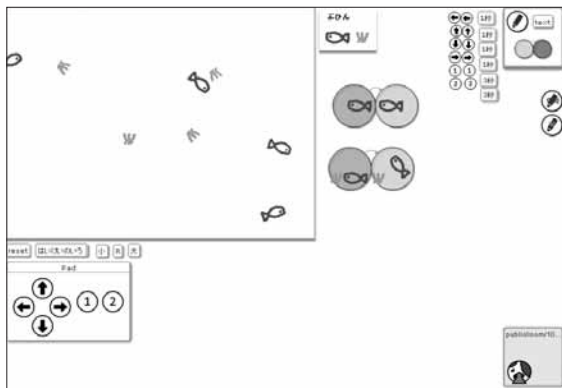


図-1 ビスケットのプログラムと実行例

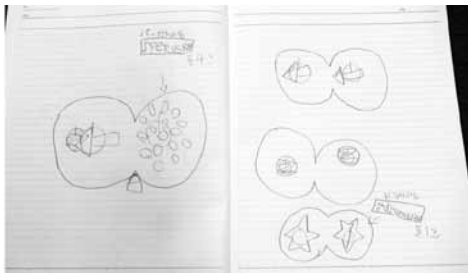


図-2 制作ノート

るように設計されたビジュアルプログラミング言語である。絵の配置の書き換え言語で、マッチングと生成に柔軟性を導入した点が特徴である。

図-1 はビスケットのプログラムと実行例である。図中の円が2つ並んだ記号が書き換えルールである。実行の各ステップで、左の円(条件側)にある図形の配置を右の円(アクション側)に書き換える。上のルールは魚がまっすぐ進むという意味になる。下のルールは魚が海藻にぶつくとよける、という意味になる。これを実行すると、魚は海藻をよけながら泳ぐ。ルールの条件側、アクション側それぞれに、アイコンを入れることで、センサや音生成などの制御も可能となる。

## みどりっ子クラブ

みどりっ子クラブは2002年に緑小学校で生まれ、2007年からは、放課後子ども教室の活動を、ほぼ毎日実施している。2010年からパソコン室を活動の場に加えるということで、Webで簡単に使用できるタイプ練習とビスケットを実施した。参加希望者が殺到し、スタッフが少人数では対応できなくなるなどがあり、さまざまな工夫をこらしてきた。現

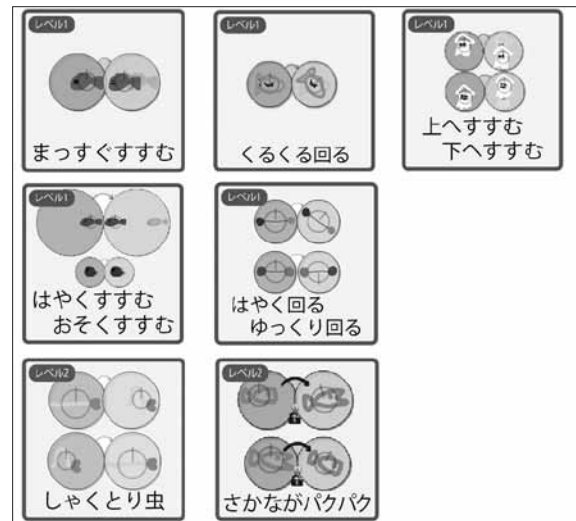


図-3 技マップ・技シール

在では、子どもたちの間で秩序ができあがり、実施の工夫もより高いレベルへの誘導にシフトしている。

この活動はさまざまな工夫によって日々変化している。これまでに実施した工夫を紹介することが、活動を理解する近道となろう。

### ●制作ノート

制作ノート(図-2)は児童に1人1冊配られる。ここには、自分が作成した作品の記録、待ち時間の間の作品の設計図(後述)、絵コンテ、キャラクタ設計などを書いていく。また、スタッフからのコメントの返答にも使われている。

### ●時間制限

児童ごとに割り当てられたIDでログインし、制作を開始してから、設定された時間(30分)までしか制作できないようなシステムを導入。順番待ちなどの交通整理が非常にスムーズになった。

### ●技マップ・技シール

全体を見ていると見過ごされがちであるが、いつまでも最初のレベルから進歩しない児童が見られた。本人が、そのレベルにとどまって遊んでいても十分楽しいということでもあるが、本人がプログラミングの奥深さに気がついていないということでもある。そこで、ビスケットで遊ぶ上で知っておいてほしいことを、小さな技として命名し、技マップにより全体像を把握できるように、また、習得した技はシールとして制作ノートに貼るようにした(図-3)。

技の例を示すと、「まっすぐ進む」、「くるくる回

る]、「ゆっくり／速く」のような単位である。それぞれ習得には非常に短時間の説明ですむようにし、子ども同士の「教え合い」をうながすよう考慮してある。

### ● 教え合いのうながし

子ども同士で、技を教え合うようにうながしている。現在では、新1年生に2年生が競って教え合う、学年の壁を越えて、低学年が高学年を教えるといった様子が見られるようになった。

教え合いは、プログラミング遊びにおいて非常に重要な要素であると考えている。スタッフの省力化という実質的な利点もあるが、それ以上に、この遊びが、教師から習うものではなく、自分たちが伝搬してゆく知識なのだということを意識づけさせたいからである。

### ● 特別編ワークショップ

平日の活動を補完する形で、年に数回、休日に数時間の集中プログラムを行う。参加人数分のPCを確保し、制限時間をなくした。これまでに、ゲーム作り、絵本作り、音遊び、お話し作りといった内容が実施された。すべての児童が参加できるわけではないが、これに参加した児童たちが中心になって平日の活動のレベルアップが期待できる。また、特別編は、システムの機能アップと同時に行われることが多かったため、新しい機能の使い方を児童たちに最初に伝搬するという意味もあった。

### ● ビスケット検定

本人の技の理解をより正確にするために、検定を行っている。基本的な技1つ1つを、大人のスタッフが確認しながら行っている。新しく参加した児童は非常に熱心に検定を受けている。自分では知っているつもりでも、基本的な部分での理解が正確ではなかった児童がおり、本人の確認にも役に立ったようである。ただし、検定が持つ強い意味（検定に合格することを目標としてしまう）があまり全面に出してしまわないよう、あくまでもプログラミング遊びをより豊かにするために行っているということには気をつけている。

### ● ハカセ問題

検定を軽くクリアしてしまった児童のために、動

きを見せてそれと同じ動きを作らせる問題を出している。技を2つ組み合わせる程度の問題は、簡単にクリアされてしまう。一方、それぞれの構成要素は単純な技であっても、プログラムの量が多く整合性を持たせるために根気が必要な問題は、興味を持たせ続けるのは難しいようである。ハカセ問題は筆者本人が現地に行かずに、スタッフとのメールによるやりとりの遠隔教育でもある。それだけに、興味を持たせるのが難しい問題は、特別編として直接教える方がよいのかもしれない。

### ● 技認定

児童が自ら技を発明したケースがあったので、それを新しい技として認定した。ビスケットはプログラミング言語であるから、開発者に知られていない使われ方がたくさんある。児童が発明した技は非常に基本的であり、すでに知られていた技と双対の関係にある美しい発見であった(図-4はすでに知られていた技「うずまき」。棒はまっすぐ進み、三角は回転しながら棒を生成する。図-5は小4の男子が発見した技「ウェーブ」。棒は回転する。三角はまっすぐ進みながら棒を生成する。図-6は実際にその男子が制作した作品。この技を応用した作品を何度も作って完成度を高めている)。

## プログラミング遊びで何が獲得されるか

プログラミング遊びの試みは非常に成功している。楽しく遊ぶというレベルはもちろんのこと、きわめて高度なプログラミングを自発的に行っている点も見逃せない。彼らはこの遊びからどのようなことを獲得することが予想されるだろうか。

### □ プログラミング遊びは普通の遊びと同列である

みどりっ子クラブの特徴でもあるが、さまざまな遊び(グラウンドで一輪車に乗る)を自由にやってよい。走り回って汗をかいた児童がビスケットを30分だけやって、またほかの遊びに行く、という光景が何度も見られている。子どもに対するコンピュータ使用への反論として「子どもが外で遊ばなくなる」といった意見

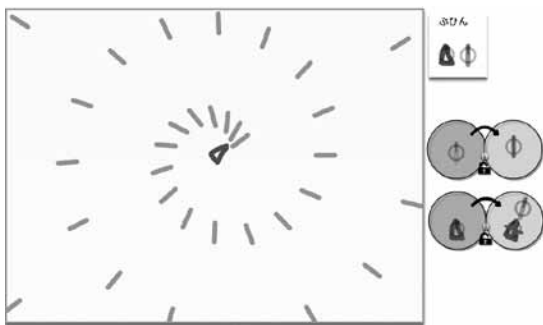


図-4 うずまき

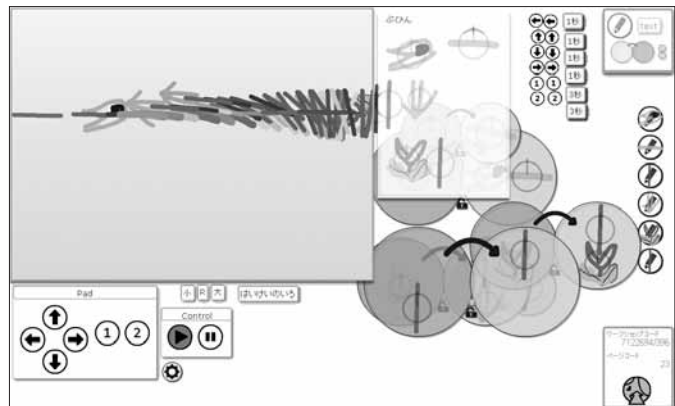


図-6 ウェーブによる作品 (小4男)

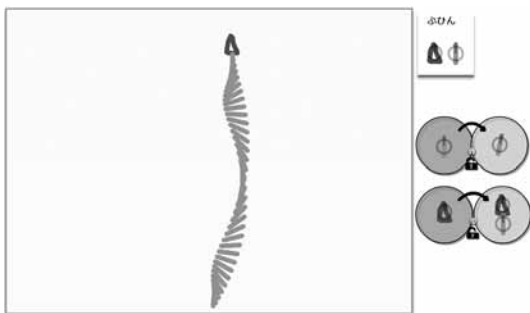


図-5 ウェーブ

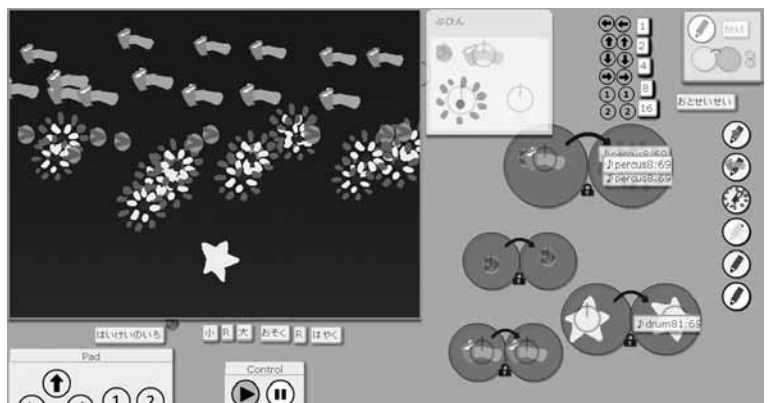


図-7 花火の例 (小2女)

をよく聞かす、まったくの杞憂である。子どもはもっとバランスを持っており、信用してよい。ただし、活動全体でコミュニケーションを重視していることが重要である。

1人で外で遊ぶのはつまらないが、1人でコンピュータで遊ぶのは面白いので、コミュニケーションのない状態が続くと本当に外で遊ばなくなる。

## □ コンピュータに対する直観

### ● 小さな変化の積み重ね

ビケットのプログラムは複数の「微小変化」によって定義され、それらの変化を連続して適用することで全体の実行が決まる。ルールベースのプログラミング言語はもちろん、手続き型プログラミング言語であっても、結局は各ステップが微小変化でしか記述できないので、「微小変化の積み重ね」がコンピュータの基本能力と言ってもよい。

10枚の連続した絵を描いて、10コマからなるアニメーションを「微小変化」によって表現することもできる。これは、繊細な表現が自由にできる一方で、作った通りにしか動かないため、できた作品は飽きやすい。一方で、少ない個数の「微小変化」による指定でも、複雑で楽しい動きができることもある。こ

れができるのは、コンピュータの本質を活かした表現が身に付いたことである。

花火が打ち上がって、弾とぶつかることで花火が出る(図-7)。プログラムは単純であるがそれぞれの花火の打ち上げの高さと、弾とぶつかるタイミングの違いで、花火の発火がよいバランスでバラバラになっている。

### ● リソースは無限に増える

「うずまき」「ウェーブ」の技にあったように、絵が増えるということを簡単に表現できる。実際にコンピュータウイルスの感染も恐ろしい早さである。増え方にも、定数なのか指数なのかいろいろある。こういった増え方の速さの直観を普段の生活から得るのはきわめて困難である。ものの腐り方、インフルエンザの感染、ねずみ講、チェーンメールなど、コンピュータ以外でもこういった現象は見られるのであるが、その直観を持った人は驚くほど少ない。プログラミング遊びはその直観を獲得できる貴重な経験になっている。

増える例は、ミサイルが連続して発射される技として教えているが、さまざまな作品の表現に使われている。指数関数的な増加は、最終的な作品の中に使われることは少ない(すぐ破たんするので)が、制作過程で偶然に遭遇することは多い。チェーンメールやネズミ講の授業では、この現象を例に説明するとすぐに理解できるのではないか。

他の現場であるが、中1の男子がリソースが増える・減るをうまくバランスさせるゲームを作った。これはまさに生体系シミュレーションであるが、彼はこれが面白い遊びであると偶然に発見した。

### ●自分がすべてを制御するということ

すべて自分で制御しなければならないし、間違えはすべて自分が原因である。ビスケツトで無茶をしても、使う側が怪我や病気にはなりにくい、玩具としては非常に安全な無茶装置である。コンピュータでの無茶を推奨し、それがすべて自分に降りかかるということも体験させている。

## □ものづくりの基本

### ●設計図による制作

コンピュータが混んでいるときに、待ち時間の間制作ノートの上に作りたい作品の設計図を書かせたところ、思いのほか上手くいった。コンピュータで直接作品を制作することは、フィードバックがすぐに得られるという点で非常に優れているのであるが、欠点もある。ビスケツトは少しの操作で振る舞いが大きく変化するので、適当に操作して偶然面白いプログラムができてしまう。プログラミング入門としてはよい性質ではあるが、より高いレベルの作品へは、きちんと設計図による構想が重要である。特別編などで時間があるときに、直接コンピュータに向かう前に、紙を積極的に使って、紙の上でプログラムの構想やプログラム本体の制作をうながしている。頭の中で動きを想像しながらものを作ることは重要で訓練になる。

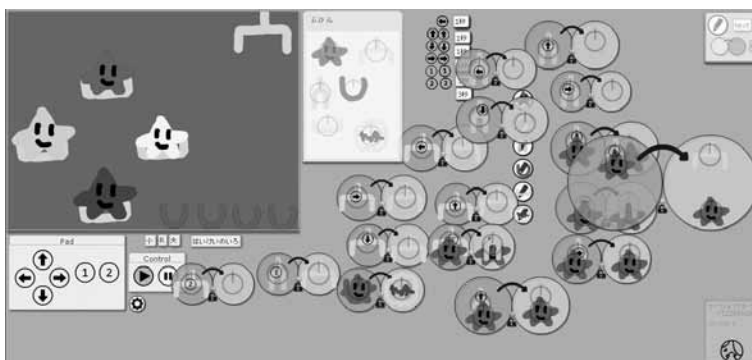


図-8 UFO キャッチャー (小3女), 機能ごとにプログラムが整理して配置されている

### ●問題を分解して考える能力

大きなシステムは、一度に作ることはできない。自分が手に負える小さな問題に分解し、それらを順に正確に制作して積み重ねていくことが必要である。ビスケツト自身にはそれを直接支援する仕組みはあえて用意していないので、あまり考えないで作ると画面がぐちゃぐちゃになる。きちんと大きなものを作る子どもは、画面がよく整理されており、問題を上手に分解して制作していることがよく分かる(図-8)。これもものづくりの基本である。

## これから

プログラミング遊びを子どもたちに伝える試みはまだ始まったばかりであるが、この魅力が広く伝わるにはまだ非常に多くの時間がかかるように思われる。その障壁の最大の原因は、プログラミング遊びを知っている大人が少ないということである。大多数の大人が知っているコンピュータはプログラミング遊びとは程遠いものである。普及のためには大人の協力者を増やすことが近道なのかもしれない。

### 参考文献

1) Viscuit, <http://www.viscuit.com/>

(2011年8月10日受付)

原田康徳 harada.yasunori@lab.ntt.co.jp

NTTコミュニケーション科学基礎研究所。IPA 未踏IT人材発掘・育成事業プロジェクトマネージャ。1992年北海道大学大学院情報工学専攻博士後期課程修了。博士(工学)。ワークショップデザイナー。