

## 子供も大人もプログラミングで本当のコンピュータに触れよう

合同会社デジタルポケット 原田 康徳

### 1. あらまし

コンピュータが持っている本当の能力はプログラムを書けばどんなものにも変身できるということです。しかし、プログラミングは難しかったので、その視点は一般人には注目されず、生活や仕事を便利に豊かにする装置という扱いでコンピュータが普及しました。そして変身できるというコンピュータの本当の魅力は、プログラムがかける人たちの秘密になってしまいました。ところが、プログラミングの研究が進んで、(応用は限られますが)誰でも簡単にプログラムを作れるようになってきました。

となると、これをいろいろなことに活用しない手はないと思いませんか？

### 2. プログラミング言語の発展

簡単に、これまでプログラミング言語がどのように発展してきたか、この先どうなるかをお話しましょう。

コンピュータの一番下の部分では簡単な数の計算だけができます。足し算なら2つの数を足すなど。複雑な計算をコンピュータにさせるためには、小さな計算をコンピュータがわかるように組み合わせます。たとえば、10個の数の合計を求めたければ、順番に2つずつ足していくように教えます。これがプログラムです。このプログラムを作ることをプログラミングと言います。

ところが、コンピュータにやらせたいことがどんどん複雑になるにしたがって、2つの数の足し算に分解するようなことを人間がわざわざやりたくはなくなります。その分解作業にはある程度パ

ターンがあります。コンピュータはいろんな作業を自動化してくれますが、プログラムを作る作業も自動化させられないかということになります。これがプログラミング言語の発明です。たとえば、少し高度なプログラミング言語では「全部の合計を求める」という命令ができて、それをコンピュータが直接理解できる2つの数の足し算の命令に自動的に分解して動きます。

プログラミング言語はコンピュータにやらせたいことが決まるとそれに応じて設計できます。たとえば、コンピュータに数学的な計算をさせたいとすると、プログラムは数式に近い形で書けるのが便利です。それでFORTRANという言語が発明されました。それから、コンピュータに人間のようを考えさせたいと思うと、今度は数ではなくa、bといった記号を扱いたくなります。それでLISPという言語が発明されました。その先は、コンピュータの用途が広がるにつれて、プログラミング言語は何万と誕生し、今も沢山生まれています。

私が開発したビスケット (Viscuit) は、画面上の絵を動かすことを目的としたプログラミング言語です。たとえば、絵を斜め上に動かしたいとき、今までのプログラミング言語だと動く方向や速さを数で指定しなければなりません。「右上43度の方向に速さ8ドット毎秒で動け」のように、細かく指定しないとコンピュータは動いてくれません。ビスケットでは絵の動かし方を数ではなく、絵を使ってコンピュータに指示します。メガネという仕組みで、絵の並びの元と後をコンピュータに教えると、コンピュータは配置の差分をとって動く方向と速さを計算します。内部では正確な数で指令されていますが、見た目は絵の並びだけ

で、大体こんな感じで動いて、という指示で済むのです。

さらにビスケットは曖昧な計算をします。たとえば、「玉ねぎが安かったら買ってきて」というお使いを頼んだとき「安いというのは1つ40円より少ないということでしょうか」と返されてしまうのが、今までのコンピュータでしょう。ビスケットは「大体安ければ買います」という動きをします。これは、絶対に間違えられては困る場合には使えませんが、「玉ねぎがほしいよ」という気持ちが伝わればよい程度であれば簡単な方がよいですね。そんなプログラミングにビスケットは向いています。今まで窮屈で使いにくかった応用でも、この曖昧な動きのおかげで劇的に使いやすくなります。プログラミングの幅を広げたということです。

ビスケットに限らず、この先もこうやって今までコンピュータで作ることが面倒だった領域にプログラミング言語が広がって、作るのが簡単になっていきます。つまり、そろそろ一般の人も使える時代がきたということです。昨今のプログラミング教育ブームはプログラミングが簡単になったから起きたわけではありませんが、プログラミングが一般の人に注目を浴びるタイミングに間に合ったのはよかったと思っています。

### 3. ビスケットってどんなもの

ビスケットはすでに述べたように絵を動かすことに特化したプログラミング言語ですが、もう一つ教育用ということを強く意識しています。たとえば、幼児に対して「子供には木のおもちゃを与えたい」というこだわりを持つ方がいらっしゃいますが、私のビスケットへのこだわりもそれに似ています。「コンピュータはこういうものだよ」というメッセージが強く入っています。コンピュータにおける積み木はどうあるべきか、という話かもしれません。

まず、大人が仕事のために作ったコンピュータをそのまま小さくしたものではありません。積み木には仕事の効率をあげるなんてことは求められ

ていませんよね。積んでは崩したり、稚拙な形を何かに見立てたり、といった遊びに「効率」という言葉はそぐいません。

例をあげると、ビスケットでは自分で絵を描くことができますが、描いた絵を複製することができません。他のツールだと絵を複製して一部書き換えて新しい絵を作るといった作業が簡単にできますが、ビスケットではあえてできないようにしています。同じ部分でももう一度描かなければなりません。もう一度描いた方がアニメーションとして味がでるのはもちろんですが、それ以上に私が込めたメッセージは、この程度の作業の節約はコンピュータにとって全く本質ではないですよ、もっと広い視点でコンピュータを使いましょう、ということなのです。操作が複雑になって、少しだけ作業が楽できて、作品の質は落ちる。「少しだけ作業が楽できる」というコンピュータの呪縛にとらわれ過ぎています。あくまでもプログラミングの楽しさ・可能性・すごさがコンピュータにとって一番重要なことという原点を大切にしているのです。大人が知っているコンピュータとはちょっと方向性が違うのです。

### 4. ビスケットの応用例

ビスケットは2003年に開発されて、様々な方々に使っていただきました。ここでは、ビスケット自身の使い方とともにご紹介したいと思います。

#### 4.1 ビスケットランド

ビスケットの入門用のワークショップです。自分で絵を描いたのち、メガネという命令で絵を動かします。

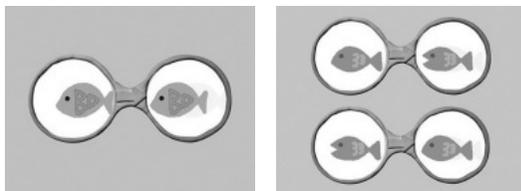
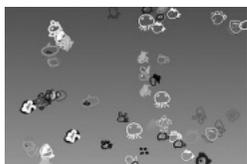


図1 魚が横に進むメガネ 図2 魚がパクパクするメガネ

メガネは左にあるものを右に変えるという意味で、時間変化を表現しています。ビスケットはこのメガネと絵を組み合わせてプログラムを作って

いきます。図1のメガネは左右の魚がずれて置かれていますので、そのずれた方向に魚が進んでいきます。ずれが大きいと魚の速度も速くなります。図2の2つのメガネは2つの絵が交互に切り替わったアニメーションになります。絵を3つ以上描いた場合は、メガネも増やせばコマ数の多いアニメーションが作れます。このようにして自分で好きなように動かすことができたなら、作品を大きな画面に送ります。そこには仲間が作った作品がまとめられて、みんなで動かした世界ができます。



ビスケットランド(図3)は、海、草原といった場所を指定したテーマの他に、赤い色、丸い形、四季といった様々なテーマで実施することができます。一人一人の作品が大きく目立たないので、絵が苦手な子でもコンプレックスを感じることなく参加でき、画面から互いにゆるく影響し合うという特徴があります。4歳児くらいからシニアまで楽しんで参加できます。

#### 4.2 感染のシミュレーション

情報が拡散する様子を体験します。まず2種類の棒人間を描き、1つは画面に沢山いれて横に動かす、もう1つは画面に1つだけいれて縦にゆらゆら動かす、というプログラムを作ります(図4)。

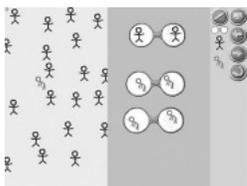


図4 感染のシミュレーション

ここまで全員ができたところで、問題を出します。「横に動いている人は健康です。ゆらゆら動いている人は風邪を引いています。ここで、健康

な人が風邪を引いた人とぶつかった時に風邪がうつるといふプログラムを作ってください」。これは少しずつヒントを出していき、最終的には小学一年生でも完成できるようです。正解のプログラムは図5です。「風邪を引いた人と健康な人がぶ

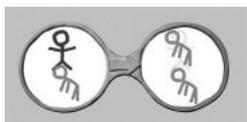


図5 風邪がうつるメガネ

つかったら、2人とも風邪を引いた人になる」と読みます。メガネの左側に絵を2つ以上入れる

と、画面上で絵がそのように並んで置かれると、その並びをメガネの右のように置き換える、という動き方をします。この絵の配置の判定を曖昧にしているのです、ビスケットが曖昧な処理をできるようにします。

このプログラムを走らせると、最初は1人だった風邪を引いた人が、ぶつかるごとに1人ずつ増えていき、最終的には全員に風邪がうつってしまうという動きをします。これは情報の拡散の例として次のような説明をしています。「ものは他人に渡すと自分の手元から無くなります。ものの原理は移動です。それに対して情報は他人に教えても自分は忘れません。情報の原理は複製です。ここで風邪を引いた人を情報を知っている人、健康な人を情報を知らない人と置き換えてみてください。情報を知っている人は最初1人だけだったが、知らない人に教えていくと、情報はどんどん広がっていきます。情報は原理的に広がるということです」。

この後、情報リテラシーの解説に繋げてもよいですし、シミュレーションの重要さという話に繋げてもよいでしょう。指数関数的に拡散する様子というのは、直感的にはわかりにくいので、このように自分でプログラムを作って動かす体験ができることは、プログラミングの特徴でもあります。

作業自体は小学一年生くらいでも全員が終わらせることはできますが、その背景にある理屈の話が理解できるのはもう少し上の学年でしょう。しかし、積み木をどのようにしたら崩れやすいかという遊びと同じと考えると(重心の概念はもっと上の学年であるように)、情報が拡散する遊びは小さいうちからやっていたよいのかもしれない。

#### 4.3 ゲームづくり

ビスケットでは画面タッチに反応するプログラムも、タッチ自体を絵と同等に扱うことで作ることが



図6 たまごをさわると、たまごが割れて、ひよこが出てくる

ことができます。たとえば、図6の例では、メガネの左側にたまごと「さわると」というアイコンが入れられており、「たまごを

さわると、たまごが割れてひよこがでてくる」というプログラムになります。メガネを増やすことでいろいろな応用が可能になります。たとえば、「たまごの右をさわると、たまごが右に動く」「生まれたひよこは横に進む」「ひよこをさわるとにわとりになる」「にわとりをさわるとたまごを生む」。

このようにメガネを1つ増やすごとに、複雑さが少しずつ増してゆきます。単純な機能を組み合わせることで複雑になっていくというのは、コンピュータのプログラミングの基本的な性質です。

#### 4.4 ルール型プログラミング

ビケットは命令の実行順序を指定しません。人間はコンピュータにやってほしいことを伝えて、コンピュータはそれに対してどのように実行するかを自動的に考えてくれます。具体的にそれを示す例をご紹介しますと思います。

図7はおちげーのプログラムです。画面にはボールの生成器（下向きの三角）と空白を示す絵が並んでいます。メガネは「生成器の下に空白があると、ボールが生成される」「ボールの下に空白

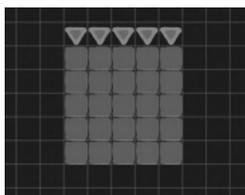


図7 おちげーの空の盤面

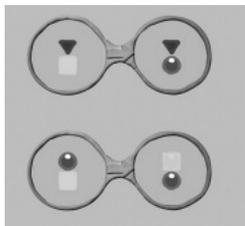


図8 ボールが生成される。ボールが落ちる

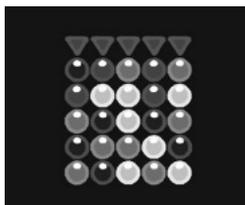


図9 ボールがランダムに埋まっているところ

次に、同じ色のボールが図10のように並んでいて、それをタッチするとボールが消える、というメガネを作ります。消えるというのはボールを空白に置き換える、ということです。これを動かしてみると、ボールが消えたところに空白ができ、その空白に上のボールが落ちてきて、その分だけ新しくボールが生成され

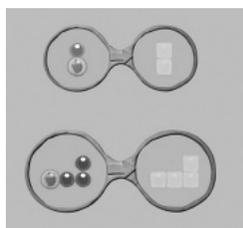


図10 ボールを消すメガネ

ます。重要なことは、ボールを消すというプログラムを追加しただけで、ボールが落ちて生成されるということも自動的に呼び出されるようになったということです。広く知られている手順型のプログラミングではこんなに簡単にはいきません。すべての状態についてもれなく処理を呼び出すように注意深く作らなければ、どこかで不具合（バグ）が出ます。逆に人間が細かい処理を書かないでコンピュータに自動化させるということは、バグの少ないプログラムを作ることでもあるのです。

#### 5. プログラミングで獲得する能力

ここで、プログラミング教育が論理的な思考の育成に役立つかどうか、という点にも触れなければなりません。ビケットはコンピュータに面倒な処理は任せて、作りたいものを直接表現しやすい、という特徴があります。そのため、素直にビケットで遊んでいるだけでは論理的な思考の育成には繋がらないと考えています。もちろん、大きなプログラムをいろんな依存関係を考えて作っていく場合には、論理的に物事を考えなければどこかで破綻します。これは作文でも作曲でもどんな創造的な活動においても言えることで、特にプログラミングだけの特徴ではありません。一方で、手順を細かく指示しなければ動かない従来型の言語やツールでは、最初に少し動かすだけでもある程度の論理が必要になります。これがプログラミングが論理的思考と関係があると思われる理由でもあります。しかし、これはプログラミング一般の話ではなく、古い言語・ツールに限定された話なのです。

プログラミング教育でどのような能力がつかか、といった難しい話は横に置いておいて、まずはプログラミングでコンピュータの本当の魅力に1人でも多くの方に触れていただけたらと思います。