コンピュータと教育 50－7
（1998．11．13）

プログラミングを通した論理的思考の外在化の実験

上田信行
甲南女子大学

古堅真彦
（財）国際メディア研究財団

本発表は「プログラミングを学ぶ」ことの教育的意義を検討するために行ってきた実験的ワークショップの実践報告である．プログラミング活動における内省的認知の重要性に注目した「Reflective Design」とJava言語を使ってreactiveな作品づくりを可能にした「Playful Design」の2つのワークショップを紹介する．これからの情報教育に必要な学習環境デザインへの一つのアプローチとして提案したい．

An Experiment of Externalization of Logical Thinking through Computer Programming

UEDA, Nobuyuki
Konan Women's University
nueda@kcn.or.jp

FURUKATA, Masahiko
International Media Research Foundation
furu@imrf.or.jp

This paper is a report of an experimental workshop designed to examine the educational significance of learning computer programming. We will present two workshops called "Reflective Design" and "Playful Design." "Reflective Design" focuses on learner's reflective cognition and algorithmic thinking, and "Playful Design" focuses on reactive programming using Java. We want to propose these workshops as one powerful approach to designing learning environments for information education in Japan.

■情報教育への新しいアプローチ

今，子どもたちの学習・情報環境は大きく変わろうとしている．一人ひとりの子どもが，自分の思いや発見を人に伝えることに喜びを感じ，人との交流を通して自分を見つめ，世界を知る感動を味わうという新しい経験が，インターネットやマルチメディアによって活性化されつつある．まさに，新しい学び手たちのコミュニティ(Brown & Campione,1994)が学校の枠を越えて，生まれつつある．今回の我々の研究は，このような子どもたちにとって，我が国の情報教育は何を目指すべきか，どのようにアプローチすべきかという問いを，「プログラミングを学ぶ」という文脈の中で考えていこうとするものである．

■Reflective Design & Playful Design

本発表では２つのワークショップを紹介したい．「プログラミングを学ぶ」ことの重要性は何かと考えると，それはアルゴリズムの理解と言えるであろう．その状況に最もふさわしいアルゴリズムを考え，プログラム化していく．

「Reflective Design」はこのプロセスに内在するプランニングと内省的認知を重視したメタ認知的学習環境デザインへのアプローチである．「Playful Design」は，Java言語を使ってマウスに反応するreactiveな作品をつくり，その作品をweb上で公開するというものであり，即興的なパフォーマンスを重んじた学習環境である．

Workshop 1

Reflective Design

本ワークショップは，プログラミング教育で大切だとされているアルゴリズムの理解を深める学びの場をどうデザインすればいいか，そして，その中で参加者たちは何を学んでいくのかを吟味したものである．

アルゴリズムというのは，ある課題をコンピュータに実行させるにはどのように手順を記述していけばいいかという論理的思考（道筋）のことであり，プログラミングとは，プログラマーの頭の中にあるアルゴリズムをコンピュータが理解できるように正確に手続きとして記述していくことである．

アルゴリズムを考え，それをプログラム化するプロセスにおいては，プランニング能力に加えて内省やモニタリングといった自分の認知過程を理解し，コントロールする意識的で意図的なメタ認知的スキルが要求される．しかし，現在の教育システムの中では，子どもたちが自らの学びのプロセスを振り返ったり，問題を深く吟味したり，新しいアイディアを創出するといった内省的認知(reflective cognition)の機会が少ないように思われる．

この内省的認知に注目してこれからのプログラミング教育を考えようというのが筆者らの視点であり，研究の動機である．

■研究の目的

本研究の目的は，プログラミング教育で重要だとされるアルゴリズムの認識が，参加者たちの共同探求過程によってどのように深まるかを明らかにすることにある．そのためにコンピュータ・プログラミング・ワークショップを実際にデザインし，実施，評価した．

今回の研究は，アルゴリズムをプログラミングに変換する過程で重要なことは何かを参加者たち自身が共同作業の中でどう気づくのかに焦点をあてた．

■研究の方法

参加者は中学３年生の男子６人であり，ワークショップは1997年８月と11月に行った．

・ワークショップ課題(8月)

アトリエ１　自然言語で図形を記述する．

アトリエ２　与えられた最小限の文法で図形をプログラムし，それを実行する．

・ワークショップ課題(11月)

アトリエ３　自然言語で図形を記述する．

◇アトリエ１

アトリエ１ではまず，２人１組になってあらかじめ用意した図１のような図形の１つを受け取る．参加者は，その図形をできるだけ的確に日本語で記述する．つまりプログラミングを行うのである．そしてその図形を見たことがない他のグループが，記述された日本語だけを読んで図形を再現する．このようなことを数回繰り返した．最初は比較的単純な図形からはじめて順次複雑にしていった．

アトリエ１のワークショップ・デザインで心がけたことは，できるだけ既存のプログラミング言語の機種やOSなどに依存する部分を排除しプログラミングの中のアルゴリズムの要素を浮き上がらせるようにしたことである．

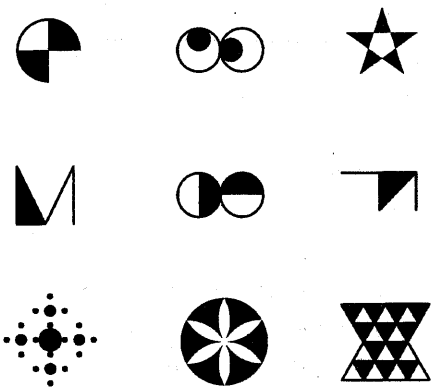


図１
アトリエ１の課題

◇アトリエ２

アトリエ２は現在のコンピュータの仕組みを実際に自分の身体を使って体験するものである．まず，アトリエ１のように２人１組になって図２のような図形の１つを受け取る．それとともに図３に記されたようなあらかじめ決められた簡単な文法も受けとる．参加者はその文法を使用し図形をプログラム化する．

そしてプログラムは，変数，インタープリタ，描き込みなどのコンピュータの各機能になりきった残りの４人によって実行される．実行結果はプログラマーの目の前のモニターディスプレイに表示される．

つまりアトリエ２は参加者自身がコンピュータの内部の各機能になって他の参加者が書いたプログラムを実行するものである．

この簡単な文法は筆者（古堅）がつくったものであり，既存のプログラミング言語を非常に簡素化した仕組みだとも言える．それを使い，既存のコンピュータのプログラムの実行の仕組みを身体を使って理解することを試みた．

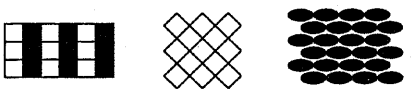


図２
アトリエ２の課題

☆（ ）という名前の変数を初期値（ ）で用意する
☆変数（ ）の値を（ ）にする
☆変数（ ）の値を（ ）だけ増やす
☆変数（ ）の値を（ ）だけ減らす
☆ペンで（ ， ）と（ ， ）を線で結ぶ
☆（ ， ）と（ ， ）で囲まれたところを四角で塗りつぶす
☆（ ， ）と（ ， ）で囲まれたところを丸で塗りつぶす
☆（ ， ）と（ ， ）で囲まれたところを四角で囲む
☆（ ， ）と（ ， ）で囲まれたところを丸で囲む
☆（ ）の間は以下をくりかえす
　｜　｜
☆もし（ ）だったら以下をする
　｜　｜
　その他だったら以下をする
　｜　｜

図３
アトリエ２の文法

◇アトリエ３

アトリエ３は課題としてはアトリエ１と同じものである．しかし，アトリエ２を体験し，さらにある程度期間をおくことにより，同様の問題に対して接し方がどのように変化するかを検討した．また図４のように，問題そのものもアトリエ１で与えたものよりも複雑にした．

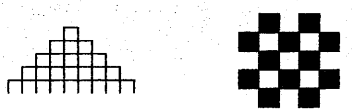


図４
アトリエ３の課題

■結果と考察

結果として，アトリエ１はゲーム性があるからか参加者は集中して課題にのめり込んでいった．そして興味深い結果もたくさん現れた．例えば，最初は「目が２つ並んでいる」などの「あいまい」と思われる表記が目立ったがそれを実行してみてうまくいかないことを体験すると，次からは「なぜうまくいかないのか，どのように表現すれば相手に的確に伝わるのか」を考えるようになった．また，ほかの参加者もその状況をみて，あいまいではない表現とはどのようなものなのかを自発的に考えるようになった．物事を伝える際にexplicit（明示的）に表現することがいかに大切であるかを参加者が身をもって体験したことで我々が目的とした初期的な段階はクリアしたといえる．また，この前日に我々がワークショップのスタッフに対して行った結果では，ある一人が「消す」という概念を考案した．受け取った図形とプログラムは図５のようなものである．するとこれは次から全体に広がり「消す」はそのコミュニティの「共通語」になった．このようにして「言語」はコミュニティのメンバーによって作られていく．

1　円をかきます
2　円の左右を２等分する直径をかきます
　　円の上下を２等分する直径をかきます
　　そうすると円の中に十字ができたと思います
　　同時に十字によって分けられた４つの扇型もあると思います
3　右上の扇型を塗りつぶしてください
4　塗りつぶした扇型に向かい合う扇型を塗りつぶしてください
5　塗りつぶした２つの扇型にはさまれた扇型で
　　右下の方の弧の部分を消す

90度，口を開けたバックマン？

図５
「消す」が含まれるプログラム

アトリエ２では日本語での簡単な文法を与え，それをヒントとしてアルゴリズムを構築することを試みた．アトリエ２はアトリエ１に比べてゲーム性よりもその複雑さの印象が強いからか，または自分が行うべき動きが直感的に認識できないためか最初は参加者の積極的な行動があまり感じられなかった．特にコンピュータの各機能の動きを理解するまでにかなりの時間を要した．しかし，最終的にはスムーズな動きが再現され，コンピュータの機能を実行した４人はそれぞれの機能を理解したようである．また，プログラムが一行ごとに実行され画面上に図形がじわじわとできていく様子はいつもコンピュータ上でしかプログラミングを行ったことがない我々自身も非常に興味を引かれるものであった（図６）．

本来はアトリエ１をある期間続けて行い，自分たちのプログラミング言語をつくり上げ，それをもってアトリエ２のような実行環境にのぞむべきでありそれが理想的なかたちである．しかし今回は１日という時間的な制約があったためこのプロセスを簡略化せざるを得なかった．時間をかけて「言語づくりのコミュニティ」が立ち現れてくるようなワークショップをデザインするのが我々の次の課題である．

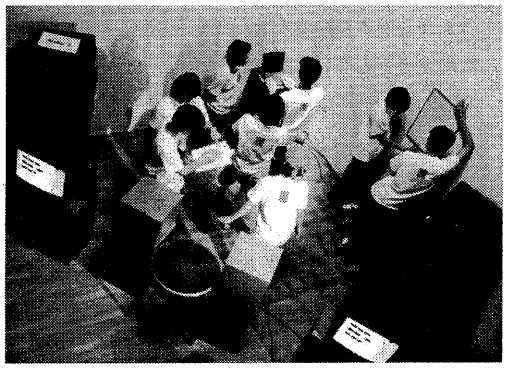
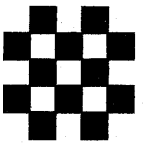


図６
アトリエ２の風景

アトリエ３においては非常に興味深い結果が得られた．プログラマーがプログラムを実行側に渡して彼らがそのプログラムで図形を描き始める．すると描画途中で明らかに描かれる図形が違っている場合，プログラマーはそれがプログラミングの間違いなのかそれとも描画している人間の間違いなのかを真剣に考え始めた．



1) まず，2cm×10cmの横にのびた太い黒い長方形を２本縦に並べる．（2cmの間隔で平行になるように）

2) 次は2cm×10cmの縦にのびた太い黒い長方形２本横に並べる．（同上）

3) この２つは重ねるものとする（○×ゲームの間隔で重ねる）

4) 1)の線に2)の線をのせて色が重なるところを白くする．（ 1) と2)を重ねてできる間隔が正方形になるようにする．）

5) そうすると４隅が白の正方形の10×10cmの大きな正方形ができる．

確かめ： 白の正方形の合計は13個，黒の正方形の合計は12個存在する．しかし書かない．

図７
アトリエ３の図形とプログラム

そして，プログラマーが，一意性を持たないプログラミング，つまりexplicitness(明示性)の度合いが低いプログラムを書いたことに気づくと積極的に，まず実行側の作業を止め，そしてそのあいまいなところを直し，再び実行側にプログラムを渡すという場面が見られた．つまり，「デバッグ」の概念が自然に発生したのである．すなわち，他者が実行するプロセスをメタレベルで見ることによって内省という作業が可視化できたのである．そして，このことが参加者のメタ認知を促進させていく．

アトリエ１ではこのデバッグの作業が自然に発生することはなかった．あくまで「この部分が違う解釈を生んだのだ」という考察段階に留まっていた．しかし，アトリエ３では間違っている部分を書き直しそれをもう一度実行させるというプログラマー側の積極的な試みがあった．これは何度も同じワークショップを繰り返すことによって参加者のメタ認知的スキル（自分の認知プロセスを知りそれをコントロールする能力）が高まったからだと考えられる．このことは非常に興味深いことである．これは，まさに人間だけでつくり上げたコンピュータ，そしてプログラミング環境といってもいい．

プログラミング教育では自分の思考過程を振り返り(reflect)，コントロールするメタ認知的スキルが重要になってくる．Reflective Designでは極力コンピュータの使用は避け，自分の身体自身も含めた機能を熟知している道具のみでワークショップを行った．このようなワークショップを行うことはプログラミングを含めたアルゴリズミックな感性を磨くこと，また上記のようなメタ認知的スキルが育つことに効果が期待できると考える．

Workshop 2

Playful Design

Reflective Designは論理的に記述するとはどのようなことかということを考える機会を与えることが目的であった．そのため機能の中身がブラックボックス化されているコンピュータを使うのではなく，出来るだけ機能の中身を熟知している普段使っている紙や鉛筆などの道具や日本語などの言語や身体そのものでワークショップを行う必要があった．それらを使い論理的な記述方法を考えるようにした．

しかし，これだけでは論理的思考そのものの本質は学べるが一回の実行があまりにも遅すぎる．また，参加者達だけで問題設定をすることが困難であり，このワークショップで得たものを実用に移す作業が参加者だけでは困難である．

そこで，この考え方を踏襲し実際にスピーディーに確実に論理的思考を具現化させ，また参加者が自ら問題を設定しやすいようにするために，コンピュータと既存のプログラミング言語の構造を使用して実際に画面上に絵を構築していくということを試みた．

前回は，使っている機能の内部がすべてわかっている上でじっくりと熟考(reflection)して，論理的思考能力をはぐくむということに重点をおいた．これに対し，今回は使う機能は多少ブラックボックス化されてはいるものの，直感的な文法を使って記述すると確実な答えが即座に返ってくる．そのようなことを何度も繰り返し，「おもしろさ」という点からどのように論理的思考をはぐくむことができるかということを試みようとした．このような意図からワークショップを「Playful Design」と名付けた．

■研究の目的

研究の目的はReflective Designの時と同様で「論理的思考の外在化」と「アルゴリズミックな感性を磨く」ことにある．また，今回もワークショップの最小単位を二人にして「協調作業」によってどのような効果が現れるかも検討した．さらに今回は最初にまず，プログラミングのエキスパートが参加者の第1グループにプログラミングの方法を指導し，次にその第1グループが参加者の第2グループに教えるという，横だけではなく縦の協調作業も加えることによってどのような効果が現れるかということも試みた．

■研究の方法

1）Programming Base for Java

このワークショップに先がけ，プログラミング言語のJavaを利用した制作環境であるProgramming Base for Javaというものを開発した．

これはプログラミングという環境の中の，機種やOSに依存した部分をできるだけ排除し，可能なかぎりプログラミングの問題の中の「アルゴリズム」というところだけを浮き立たせるようにしたものである．プログラミングの初心者でも比較的容易にプログラミングの特にアルゴリズムの構築という作業に没入できるような仕組みにした．

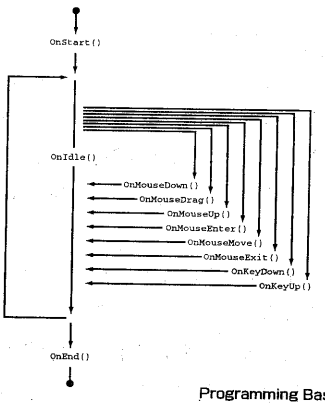


図8
Programming Baseの構造図

Programming Base for Javaは図8のような構造を持つ．つまり，まずアプリケーションが立ち上がったときにはOnStart，アプリケーションが立ち上がっている間はOnIdle，そしてアプリケーションが終了するときはOnEndの中身が実行される．さらにマウスやキーボードなどのいわゆる「イベント」がおきたときにそれぞれOnMouseDown（マウスボタンが押された瞬間），OnMouseDrag（マウスがドラッグされている間），OnMouseUP（マウスボタンが放された瞬間），OnMouseEnter（マウスが表示領域に入ってきた瞬間），OnMouseMove（マウスが動いている間），

OnMouseExit（マウスが表示領域から出ていった瞬間），OnKeyDown（キーボードからキーが押された瞬間），OnKeyUp（キーボードからキーが放された瞬間）の中身が実行される．それぞれの中身はJava言語の文法に則って書かれる．

例えばOnMouseDownにgrf.drawLine(100, 100, 200, 200);と書くだけでマウスのボタンが押された瞬間に画面上の(100, 100)から(200,200)まで線が引かれるという行為が作れる．また，それぞれはイベントの要素（マウスの位置や押されたキーの値）を値として持っており，これによって例えばOnIdleにgrf.drawLine (0, 0, mouseX, mouseY);と書くと常に画面の原点からマウスまでの線が書かれるような仕組みが作れる．例としてこのProgramming Base for Javaで作成した作品と実際のプログラムを図9と図10に示す．

2）ワークショップの流れ

参加者は高校生（1年生）が4名（男子2名，女子2名），大学生が2名，大学院生が1名，我々研究者が3名であり，1998年の8月から9月にかけて行った．

内容としては，1]2日間のアトリエ活動，2] 高校生たちだけの作業（e-mailと電話による我々との相談を含む），3] Web上での展覧会＋アトリエでの作品発表会＋新規参加者を対象にした作品作りから構成されている．

1]では最初1日かけてProgramming Base for Javaはどのようなものか，これを使うとどのようなことができるかということを高校生1名に説明し，我々研究者と共に簡単な作品作りを行った．2日目は，1日目に教えた高校生と，我々研究者と大学院生が新しく参加したメンバーをガイドし，2チームに別れて作品をつくりあげていった．

2] では1週間後に高校生たちだけが集まり，作品づくりを行った．その時，我々研究者に電話とe-mailを通して相談がなされた．

3] に関しては，Web上での展覧会とアトリエでの作品発表会を1ヶ月後に行った．また，それに付随して最初の2日間にアトリエ活動を経験した参加者が新たに加わった参加者に指導をし，できた作品をWeb上の作品群に加えるということも試みた．

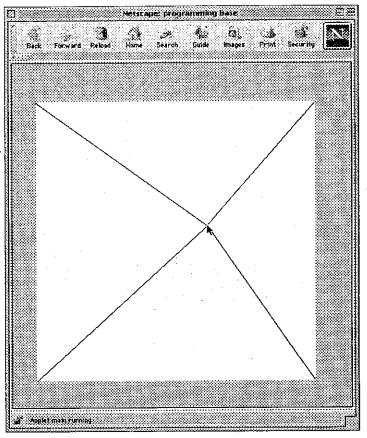


図9
Programming Baseの例(イメージ)

```
//base.java
import java.applet.*;
import java.awt.*;

public class base extends Applet {
   Graphics   grf;
   Image      offs;
   int        monitorWidth, monitorHeight;

   public void onStart(){   }
   void onIdle(int mouseX, int mouseY){
      //ペンの色を白にする
      grf.setColor(Color.white);
      //画面全体を白で塗りつぶす
      grf.fillRect(0, 0, monitorWidth, monitorHeight);
      //ペンの色を緑にする
      grf.setColor(Color.green);

      //マウスの位置から画面の四隅まで線をひく
      grf.drawLine(mouseX, mouseY,0, 0);
      grf.drawLine(mouseX, mouseY,monitorWidth, 0);
      grf.drawLine(mouseX, mouseY,monitorWidth, monitorHeight);
      grf.drawLine(mouseX, mouseY,0, monitorHeight);
   }
   public void onEnd(){   }
   void onMouseDown(int mouseX, int mouseY){   }
   void onMouseDrag(int mouseX, int mouseY){   }
   void onMouseUp(int mouseX, int mouseY){   }
   void onMouseEnter(int mouseX, int mouseY){   }
   void onMouseMove(int mouseX, int mouseY){   }
   void onMouseExit(int mouseX, int mouseY){   }
   void onKeyDown(int theKey){   }
   void onKeyUp(int theKey){   }

   public void init(Image off, int monitorWidth, int monitorHeight){
      offs = off;                  //don't touch here
      grf = offs.getGraphics();   //don't touch here
      this.monitorWidth = monitorWidth;
      this.monitorHeight = monitorHeight;
   }
}
```

図10
Programming Baseのサンプル(コード)

■研究の結果と考察

ワークショップでは最初にどのような作品がつくれるかのデモンストレーションを行った．この作品

群が参加者をひきつけ，このようなプログラムがほんとうにつくれるのかという驚きと，早くやってみたいという気持ちが重なって，すぐに作品づくりに入っていった．最初にJavaプログラミングに必要な最小のメソッド（関数）を説明し，簡単な作品をまず仕上げるようにガイドした．

作品はアトリエで６つ完成し，1週間後の高校生だけの作業で2つ完成した．完成作品はすぐにWebサイト上にアップロードした．

<http://www.kcn.or.jp/~nueda/playfulDesign/index.html>

また，最後のアトリエでも新規参加者が２つの作品を完成させ，それも同サイトにアップロードした．

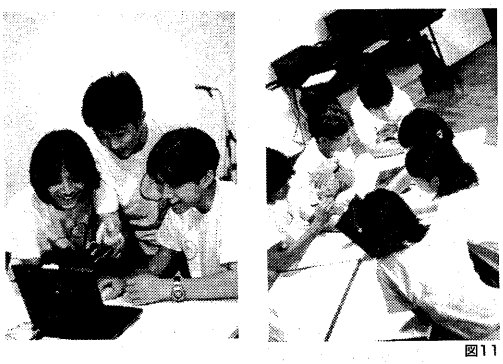


図11
Playful Designの風景

今回のプロジェクトで特筆すべき点は，ワークショップが活性化した一つの要因に，Javaとインターネットが果たした役割が大きかったことである．さまざまなプログラミング言語の中からJavaを選んだ理由としては，１）作った作品を即時公開できるということと，２）制作者どうし，または制作者と指導者のネットワークを介した作業が可能であるということがあげられる．

１）はJavaで作った作品はインターネットを介して世界中の人々に見せることができるということである．他の言語の場合は，作品を他の人に見せるときは作品そのものをその人のところに持っていかなくてはならない．しかし，Javaであれば作品をサーバーと呼ばれるコンピュータに置いておきさえすれば世界中からインターネットを介してその作品を閲覧することが可能である．

このことも参加者にとって，本気で作品をつくる（閉じられた世界ではなく，オーディエンスを意識した）大きな要因になったようである．ただ単にプログラミングができるようになるというだけではなく，制作物を非常に多くの人にすぐにみてもらえるという環境が，参加者をつくり手としての姿勢に向かわせたようである．２）もインターネットによって実現できたことである．このワークショップは学校などの閉じられた環境ではなく，指導者と参加者，また参加者どうしが遠隔地に存在する環境で行った．そのため2]の段階ではどうしても電話を含めたいわゆるネットワークに頼らなければならない．電話だけだとプログラミングのコードの説明をするのに非常に困難を要するが，インターネットを介し，容量が小さくさらにすぐに閲覧できるJavaを使うことが遠隔地での指導や相談という面に大きな利点をもたらした．

また，今回新たに試みた参加者が新規参加者をガイドするということにおいても興味深い結果が得られた．参加者にはあらかじめ「次の日に来る参加者には君たち自身が教えなくてはいけない」ということを伝えておいた．筆者（古堅）はプログラミングの講座を他の場所でも行ったことはあるが，その時は自分で作品を作ったらそれを提出して終了というものがほとんどであった．この場合，参加者は少しぐらい原理的に納得いかない点があったとしても，結果的に作品がうまくできているのでなんとなくその疑問点を放置しておく傾向が見られた．しかし今回はそのような疑問は次の日に自分自身に降り掛かってくるということがわかっているので作品そのものが出来上がったとしても原理的な疑問点を可能な限り追究するという場面がいたるところで見られた．この「責任」というものを指導者側から意識的に与えることも参加者に積極性を促す要素となるようである．

■参加者の作品

参加者を観察していると，最初の作品が完成するとすぐに，次につくりたい作品のイメージがどんどんと浮かんでくるようであった．どうアプローチすべきかのアルゴリズムを考えてはプログラムを書き，

何回もデバッギングを繰り返してイメージ通りの作品をつくることに挑戦していった．Programming Baseがプログラムの構造がはっきりとわかるようにデザインされているため，自分が書いたプログラムをチェックし，デバッギングしやすかったのだと思う．作業環境にエキスパートがいるだけで，あまり細かいところを気にせずに，つくりたい作品のイメージに集中してプログラミング出来たようである．

ここでは紙面の関係上，ソースコードを載せることができなかったが作品のいくつかのスナップショットを図12に示す．

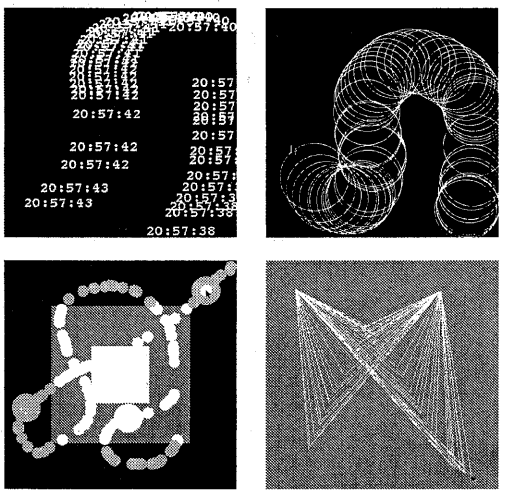


図12
Playful Designの作品

■新しい「学び」が情報教育に出会うとき

2つのワークショップを通して我々が子どもたちに経験してほしかったのは，従来の知識伝達を中心としたインストラクション型の「学び」ではなく，子どもたちが自分にとって意味のあるデザイン活動（プログラミング）に没頭することによって作品をつくりあげるというコンストラクション型の「学び」であった．ワークショップに参加した子どもたちはプログラミング活動と共に立ち現れてきた「学びのコミュニティ」を強烈に感じ，その中に共有，分散しているリソース（エキスパートやパートナーなど）をフルに活かして作品作りを行った．特に，Playful Designでは，筆者の一人がJavaによるProgramming Baseを開発し，子どもたちと共にワークショップを進め，作品を作り上げていったところにこのプロジェクトのダイナミズムがあったように思える．

今回のワークショップを通して子どもたちは，自分たちの伝えたいメッセージを論理的に組み立て，表現し，仲間と共に考えていくことの楽しさ難しさを自らの体験を通して学んでくれたと思う．このような「学びのコミュニティ」を，これから本格的に始まる情報教育で実現することが出来ればすばらしいことだと考える．

今回のプロジェクトを行う上でラーニング・デザインの岸本菜穂美さんにはとてもお世話になった．ここに記して感謝する．

■Readings that inspired us

- Brown, A.L. & Campione, J. C.(1994). Guided Discovery in a Community of Learners. In K. McGilly, ed., *Classroom Lessons*. The MIT Press.
- Brown, J. S.,Collins, A., & Duguid, P. (1989). Situated knowledge and the culture of learning. *Educational Researcher*, 18(1), 32-42.
- Harel, I & Papert, S. (Eds.). (1991). *Constructionism*. Norwood, NJ: Ablex.
- Kafai, Y.B. (1995). *Minds in play*. Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.
- Kafai, Y.B. & Resnick, M. (1996). *Constructionism in practice: Designing, thinking, and learning in a digital world*. NJ: Lawrence Erlbaum Associates.
- Lave, J. and Wenger, E. (1991). *Situated Learning - Legitimate peripheral participation-*, Cambridge University Press.
- Rogoff, B., & Lave, J. (Ed.). (1984). *Everyday Cognition: Its Development in Social Context*. Cambridge, MA: Harvard University Press.
- Vygotsky, L.S. (1978). *Mind in society: The development of higher psychological processes*. Cambridge, MA: Harvard University Press.