

Programming Baseを使用したデザイン教育の研究と実践

デザイン教育におけるプログラミングのあり方の提案

Study and Status Report: Programming Base in Design and Educational Environment.

— A proposal for programming syllabus incorporated in curriculum for design.

古堅真彦

Furukata Masahiko

財団法人国際メディア研究財団

Abstract : Regarding designs on computers, issues about why and how programming technology will be incorporated to creative process has been discussed more these several years. Researches on user interface for designers to program are in its preliminary stage, but programming technique seems to have high potential to expand the design possibilities because the
Key Words : design, programming, algorithm

technique allows designers to fully manipulate the digital media. Programming activities naturally employs logical thinking and designers who learned the skills may reflect their creative process and develop new one. This is a report on an experiment of programming education and a proposal of its course ware for the design majored students.

■はじめに

デザインワークにプログラミングという制作手法をとり入れるべきかどうかという議論が数年前から盛んに行われている。今回はそのような状況の中で筆者が行っているデザイン分野における新しいプログラミング教育およびそれを通じたデザイン教育のあり方を提案する。

■背景

プログラミングという制作手段はユーザーインターフェイスという面ではまだまだ発展途上段階であり、「コンピュータに熟達した人のみが行える制作手法」という印象が強い。しかし、手法そのものをとらえてみればコンピュータを根底から操作できる道具指向な制作手段であり、「モノを作ること」が目的であるデザインワークに非常に有効な制作手段といえる。

また、プログラミングには常にアルゴリズム的な感性というものが要求される。つまり、ある目的を達成するとき、そのためにはどのような行程がもっとも効率が良いかということを見極める力が必要とされる。このアルゴリズム的な感性を育むということは、「機能」や「効率」など「アルゴリズム」に非常に関係深い要素が常に要求されるデザインワークのスキルアップにもつながるのではないかと考えられる。

このようなことから筆者はデザイン分野においてコンピュータは有効な道具であり、さらにプログラミングはそのデザインワークにおける思考および制作段階の有効な手段だと考える。

しかし、先述のとおり、一般のデザイナーが簡単にプログラミングするという状況は現段階では困難なものがある。

筆者は普段プログラミングという方法でデザインワークや研究を行っている。そこでそのスキルを活かし、デザイナーができるだけ直感的に、さらにアルゴリズム的な感性をはたらかすことができるようなプログラミングの環境作りを目指した。

Programming Baseとはそのようなプログラミング環境の名称である。

■Programming Base

具体的にはProgramming Baseとは最初の単純作業レベルのことはあらかじめソースコードとして用意しておいて、そのソースコード群に書き足すことによって制作をしていく、という

環境であり、プログラミング環境でのテンプレートとすることができる。

Programming Base開発の経緯としては、Programming Baseという具象にする以前から個人的にプログラミング用のテンプレートを用意し、使用していた。しかしあくまで個人的な環境で作っていたため制作の際にプログラマーとしてのアプローチとデザイナーとしてのアプローチをよく混同していた。つまりテンプレートの根底の部分を書き直しているのか、テンプレートを利用してコンテンツを作っているのかがあいまいになっていた。そこで、テンプレートを他者と共有することでこの問題を解消しようと試みた。つまり、自分もそのテンプレートの一利用者になることにより、テンプレートをデザインしているのか、それともテンプレートを使用しているのかを明確に分けるように考えた。このようにすると、このテンプレートのデザイン分野での位置付けを明確にでき、一般のデザイナーもより簡単にプログラマーではなくデザイナーとしてプログラミングが行えるのではないかと考えた。

■Programming Baseの構造

ここでProgramming Baseの構造について触れておく。

Programming Base自体はCまたはJava言語で記述するプログラミング環境である。構造としては基本的に**いつ、どうする**という記述で成り立っている。例えばあらかじめ用意されたOnMouseDownという領域に

```
g.drawLine(100, 100, 200, 300);
```

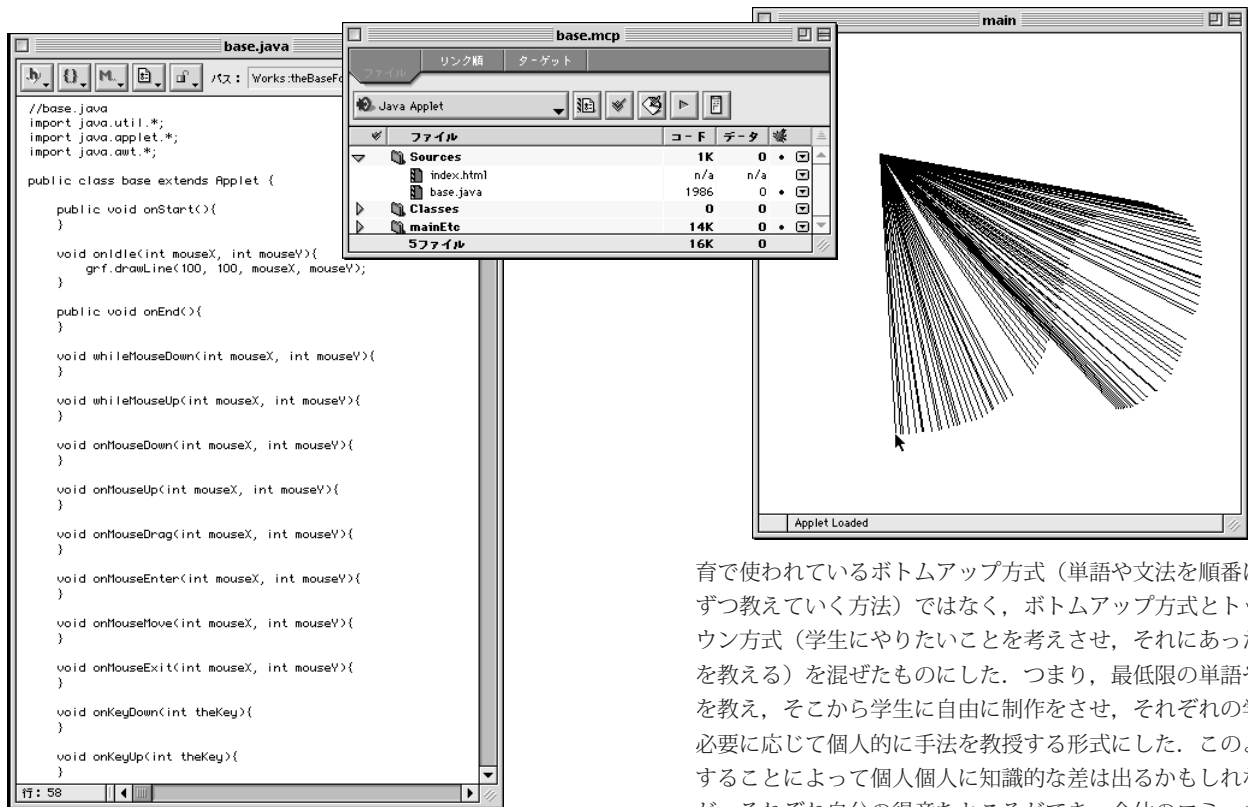
と書けば、マウスのボタンが押されたときに画面上の(100, 100)という位置から(200, 300)という位置まで線を引くというものが出来る。このOnMouseDownというのが**いつ**であり、

「g.drawLine(100, 100, 200, 300);」というのが**どうする**である。**いつ**にはこれ以外に「OnStart (最初に)」「OnIdle (いつも)」「OnEnd (最後に)」「OnMouseDown (マウスボタンが放されたら)」「OnKeyDown (キーが押されたら)」などがある。**どうする**にはCまたはJava言語をそのまま記述する。

このほかの特徴としては例えばOnIdleの領域に

```
g.drawLine(100, 100, mouseX, mouseY);
```

と書くと(100, 100)の位置から現在のマウスの位置まで線が引きつけられるといういわゆる**インタラクティブ性**をあらかじめ



図：Programming Base

じめ内部の構造に含んでいるというものがあげられる(図)。

Programming Baseの利用者は基本的にこれらのことが一つにまとまったファイルを書き換えるだけ作品を作ることが可能である。

■制作概念と教育カリキュラム

Programming Baseはあくまでデザイナーが対象なので、コンピュータの内部の機種に依存する部分ではできるだけ意識せずに、デザイナーが持つアルゴリズム的思考をそのまま記述できるようにした。つまり導入部分は紙と鉛筆というメタファーで表記できるような仕組み作りを心がけた。ただし、先にも書いたように筆者はコンピュータやプログラミングは「アルゴリズム的思考を外在化できる道具」だと考えているのでProgramming Baseもプログラミング本来がもつ「アルゴリズムの記述」という側面は一般のプログラミング環境と変わらないようにしている。また、Programming Base自体はCやJava言語で記述していくプログラミング環境である。そこを入り口にしてさらに技術的な側面からプログラミングを習得するという場面が出てくることも大いに考えられる。そこでProgramming Base自体のソースコードはできるだけオープンなものにし、構造もできるだけシンプルなものを目指した。つまり、初心者は今までのメタファーを使い直感的にプログラミングができて、プログラミングの概念を習得したものはその特徴を活かしたデザインワークができて、そこからプログラミングの技術面での内部の構造に興味を持ったものは比較的容易に中身を理解できるという環境を目指した。

また教育カリキュラムに関しては、あくまでデザイナーが対象なので、最初の技術習得段階の教育にはそれほど長い時間をとるべきものではないと考えたので、既存のプログラミング教

育で使われているボトムアップ方式(単語や文法を順番に1つずつ教えていく方法)ではなく、ボトムアップ方式とトップダウン方式(学生にやりたいことを考えさせ、それにあった手法を教える)を混ぜたものにした。つまり、最低限の単語や文法を教え、そこから学生に自由に制作をさせ、それぞれの学生の必要に応じて個人的に手法を教授する形式にした。このようにすることによって個人個人に知識的な差は出るかもしれないが、それぞれ自分の得意なところができ、全体のコミュニティの中での自分の位置づけも明確になるのではないかと考えた。

■考察

このような環境をもって、武蔵野美術大学、多摩美術大学、国際メディア研究財団や他にさまざまな場所でのワークショップを行った。それぞれの場合において人数、対象、年齢が異なったため、単一の教育カリキュラムではなかなかうまくいかず、その場に応じたカリキュラムが必要なことを痛感した。しかし、どの場面でも今までのプログラミング教育でよく使われていたボトムアップ方式だけでは受講者の「やる気」を喚起することはできず、デザイナーに限らず「モノを作る」という明確な目標を設定してそこを目指して個人(またはグループ)が進んでいき、そのために必要な概念や技術を習得するというカリキュラム構成のほうが結果として有機的に学習できるのだということを実感した。

また、できるだけオープンな環境作りを目指し、Programming Baseの使用者でも簡単にProgramming Baseに機能を追加できるような仕組みを考えたのだが、その仕組みを理解させるためにはどのような教育カリキュラムが適切かはまだ研究途中であり、今後の課題である。

■参考文献

- 1) 上田信行, 古堅真彦, 岸本草穂美. プレイフル・デザイン. 1998. 第5回日本視聴覚・放送教育学会1998年度大会発表論文集, p.22-23
- 2) 上田信行, 古堅真彦. プログラミングを通じた論理的思考の外在化の実験. 1998. 情報処理学会研究報告98-CE-50, p.49-56.
- 3) writing space design 98/99. 1999. 武蔵野美術大学視覚伝達デザイン学科