マルチメディア教材ツールとしての"HyperCard Lessons"の制作にいたるまで

Progess up to "HyperCard Lessons" as an authoring tool for Multimedia software

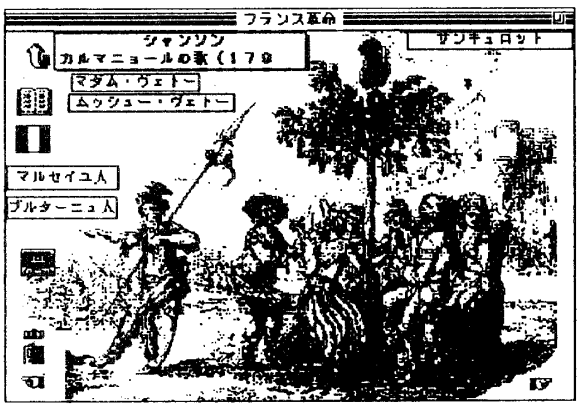
中田　平
NAKATA, Hitoshi
金城学院大学
Kinjo Gakuin University

＜概要＞　筆者は1994年4月に"HyperCard Lessons"（ハイパーカードレッスン）というスタックウエアーをCD-ROMで開発した。これは『ハイパーフランス革命』で始まった筆者の教材開発が、『ハイパー科学館』、『MacBASIC』という教材開発を経て、HyperCardを本格的に学習教材として定着させようという試みにまで到達した結果である。その意味で、筆者にとってマルチメディア教材開発についての一種の総決算とも言えるソフトウエアーである。

＜キーワード＞　HyperCard　マルチメディア　教材開発　言語教育　CAI　BASIC

1.　『ハイパーフランス革命』

フランス革命200周年記念の1989年、筆者は「一般教育演習」という科目で「フランス革命」をテーマに取り上げ、200年前の事件の今日における意味を学生と一緒に考えた。『ハイパーフランス革命』と名付けたHyperCardスタックは、この演習科目の成果として誕生したものである。製作に当たっては、マッキントッシュのハイパーカードの内在的な能力をフルに引きだすことを目指した。グラフィック（絵や写真やビデオ）とサウンド（シャンソンや効果音）とテキスト（事件や社会の動きについての解説）を素材にして、フランス革命の展開の通時的追跡を軸に、重要人物の紹介、資料、統計、その他、当時のシャンソンまで織り混ぜて、体験的に理解できるようにと考えた。



『ハイパーフランス革命』の一枚のカード

フランス革命を立体的に把握するために、スタック全体を４つの部分に分け、それぞれ、１）年代史、２）登場人物、３）資料、４）シャンソンとした。編年構成の年代史を軸に、革命の登場人物を絡ませ、年代記をはみ出す資料を別にまとめ、最後に革命当時のはやり歌を収録した。結局、スタックは全部で３．５メガほどに膨れ上がったのである。

このスタックは翌年のアップルコンピュータの第2回スタックウエアコンテストでグランプリを獲得した。

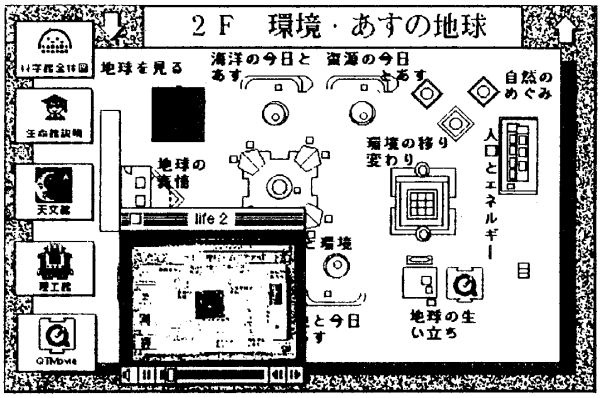
2.『ハイパー科学館』

1991年の秋から、筆者は名古屋市科学館の経営係の依頼で科学の博物館という素材をマルチ

メディアで見せることに取りかかった。「ハイパー科学館」は名古屋市科学館を紹介するためのハイパーカードスタックであるが、そもそも制作の動機となったのは、名古屋市科学館の経営係の業務の必要性からである。それは、普通考えられるような、科学館の展示物そのもののプレゼンテーションの重要性よりは、現代のイベントホールに共通のマネージメントサイドの問題、つまり臨時案内業務の方々の職員教育、あるいは社員研修といった深刻なマネージメントの問題があるからだ。

実は、名古屋市の科学館に限らず、博物館、美術館、イベントホール、その他、現在、集客環境の拡大しつつある時代にあって、そのイベント会場の直感的把握をどういう手段で行うかが、極めて深刻な問題になっていると言っていいからだ。

さらに、名古屋市科学館の小学校、中学校の生徒の入館者のために、引率の教員の下見にも使用する。そのためには、全部の建物の展示物の内容をくまなく提示する必要があり（展覧的網羅性）、と同時に、限りのある時間のなかで有効に観覧する計画を立てるために、重要な展示物やイベントをかいつまんでピックアップする必要もある（要点の把握）。そうした矛盾した2つの要求に答えることが重要な責務だった。こうした観点から『ハイパー科学館』は作られたものである。名古屋市科学館のエントランスに設置された『ハイパー科学館』は、こうした種類のイベント会場のプレゼンテーションツールとしてある種のモデルになると思われる。



『ハイパー科学館』の一枚のカード

3.『MacBASIC』

筆者自身を含めて4人の開発プロジェクトチームで、1993年2月にHyperCardを利用してBASICをマッキントッシュ上で実現するCAIソフトウエアー『MacBASIC』を作ることに成功した。「情報基礎」の教科書はコンピュータ言語の見本としてBASICを選んでいる。確かに、かつてBASICは初心者にとっての唯一のコンピュータ言語であったために、中学生にコンピュータ・プログラミングを教えるには格好の入門プログラムであることは事実であろう。しかし、今日ではコンピュータ言語の選択はコンピュータに何をさせるかによって決定されるもので、決して一義的なものではない。コンピュータのOSがBASICで書かれているわけでもない以上、BASICはたんなる実例に過ぎないという点を強調しなくてはならない。したがって、「情報基礎」がプログラミング言語としてBASICを選んだのは、たんにコンピュータの働きを知るための選択だと考えるのが妥当である。

われわれは、BASICを使ってBASICをそのまま教えることが教師にとっても生徒にとっても「教育的」であるとは考えなかった。われわれは、コンピュータに命令して結果を得るというプログラミングの構造を、視覚的に、あるいは別の言葉で言うとオブジェクト・オリエンティッドなかたちで教えたいと考えた。そのために、HyperCardを使って、HyperTalkでBASICのコマンドを受けてコンパイルするという回りくどい方法をとったのである。そのことによって、オブジェクト指向のプログラミングの可能性に道を開くことができるのではないかと考えたからである。

われわれがMacBASICと命名したプログラムで実現したかったことは、「情報基礎」のなかの「プログラムの作成」あるいは「プログラムの活用」という項目に則って、コンピュータ・プログラミングの基本構造を中学生にわかりやすく教えることである。MacBASICはHyperCardという優れたアプリケーションを使って作られている。HyperCardは、それ自体がHyperTalkというプログラミング言語をもっているため、MacBASICは、HyperTalkというプログラミング言語によってBASICというプログラミング言語

を教えるという、いわば教育実験を試みているわけである。HyperCardはここでは一種のメタ言語の役割を演じているといえる。MacBASICによって、生徒はBASICからHyperTalkというよりフレクシブルな言語へ無理なく入ることができるだろう。生徒たちが、プログラミングの楽しさに目覚め、ハイパーカードに取り組む楽しさを自覚すること、それがわれわれの目的であった。

MacBASICでプログラムを打ち込んでいる画面

このソフトは、マッキントッシュを利用する情報科学の基礎教育になくてはならない言語教育ソフトであるという確信がある。しかし、このソフトのコンセプトを考えた筆者自身、このソフトの制作は、HyperCardというコンピュータ言語の一面しか示さない、一種の必要悪のようなものだという思いがあったことも否定できない。

"HyperCard Lessons"（ハイパーカードレッスン）という教材を世に出すことによって、筆者は『MacBASIC』というソフトと一対のHyperCard教材を完成したという満足感を味わうことができたのである。

4.『HyperCard Lessons』

筆者は1994年4月に"HyperCard Lessons"をCD-ROMで発売した。HyperCardのようなオブジェクトをもつコンピュータ言語の学習プログラムは決して組み立てやすいものではない。しかし、HyperCardにも学習者に無理なく教えることができる標準的な順序をうちたてる必要がある。"HyperCard Lessons"が目指したのはそういう教材である。

HyperCard Lessonsの構成は次のようになっている。

レッスンは12のレッスンと6つのプラクティスでできている。

イントロダクション

HyperCard の基礎

◆新規スタックの作り方について—小手調べ

—Lesson 1　マウスで絵を描こう

—Lesson 2　ボタンの役割

—Lesson 3　フィールドの役割

★Practice 1　絵葉書を描こう→この練習はカードを絵葉書にみたてて絵を描く

—Lesson 4　メッセージボックス

—Lesson 5　ボタンのスクリプト

★Practice 2　ボタンやカードを使ったアニメーション・スタック

—Lesson 6　フィールドのテクニック

★Practice 3　文書キャビネット→文書を格納するスタック

—Lesson 7　サウンド

★Practice 4　データベース作り→データベースを作るスタック

—Lesson 8　カラー PICT

—Lesson 9　リソース

★Practice 5　Colorizing HyperCard を使ってカラーのスタックを作る

—Lesson 10　QuickTime

★Practice 6　QuickTime のスタックを作る

—Lesson 11　スタック・スクリプト

—Lesson 12　コマンドとファンクション

ここでは、時間の制約もあるため、学習マニュアルのなかから「Lesson 9 リソースとはなにか？」の一部だけを引用したい。

Lesson 9 リソースとはなにか？

HyperCardのリソースの種類はアイコン、カーソル、ピクチャー、サウンド、XCMD、XFNDです。そのなかでXCMD、XFNDはHyperTalk以外の言語、PascalやCといった言語で書かれた外部コマンド、外部ファンクションです。XCMD、XFNDは『ハイパーカードレッスン』の範囲を超えていますから、ここでは取り扱いません。しかし、たとえばHyperCard自身にもPictureなどのXCMDが内蔵されています。

つまり、HyperCardの限界を超えたプログラミングをおこないたいと考えるユーザーにとって、PascalやCといった言語で書いたプログラムを実行するパスをはじめから予定していたという意味で、HyperCardがいかに拡張性の高いプログラムであるかがよくわかります。

◆サウンド・リソース

サウンド・リソースの作り方は、audioスタックを利用することがもっとも簡単です。audioスタックを利用すればマイクから直接音声入力をし、Saveすればそれを自分のスタックにサウンド・リソースとして保存することができます。また、Editを使えば入力したサウンドの編集もできます。

◆カーソル変更

マッキントッシュの標準カーソルはIビーム<Ibeam>、十字<cross>、プラス<plus>、時計<watch>、手<hand>、矢印<arrow>、ビジー<busy>、なし<none>があります。下の絵では<none>のかわりに<蛙>になっています。

カーソル変更は簡単ではありません。その右にあるResEditというアプリケーションを使ってシステムのリソースを書き変える必要があるからです。ResEditはマッキントッシュのユーザーにとって不可欠のものですが、その取扱には十分注意してください。

◆ピクチャー・リソース

カラーのピクチャーを表示したい場合、すでに説明したようなHyperCardに内蔵のPictureコマンドを使ってピクチャーをファイルのままで利用する方法があります。しかし、それ以外にもリソースとしてスタックに組み込んでColorizing HyperCardのようなXCMDを使って表示する方法があります。ところで、カラーのPICTはペイント系のソフトなら何でも描けますが、通常はPICTファイルになります。Photoshopは直接PICTリソースとして保存しますので、Photoshopを使ってリソースにして、リソースムーバーでスタックのリソースに組み込むことができます。

このカードのスクリプトには次のスクリプトが書いてあります。

```
On opencard
  lock screen
  colorizeHC "new","cap","250,100"
  unlock screen
End openCard
on closeCard
  lock screen
  colorizeHC "Erase"
  if the result is not empty then answer the result
  unlock screen
end closeCard
```

colorizeHC "new"とcolorizeHC "Erase"で表示と消去ができるのです。

◆アイコン・リソースとアイコンの作り方

アイコンリソースを新しく作るためには、編集メニューの一番下にあるアイコン編集ツールを使います。

このカードの「アイコン編集」のボタンも同じことができます。アイコン編集は難しくありません。まず、ビットマップの絵を準備しておきます。アイコン編集メニューから「切り取り」を選択して切り取りたい絵の上でクリックします。すると、アイコンの絵の原形として取り込まれます。それから「特殊機能」のテクニックを利用して好きなアイコンに仕上げてください。

次のカードは左の大きな絵を22のアイコンにしたものです。大きな絵でもアイコンにすることができます。

◆リソースムーバー

HyperCard2.0のフルセットのなかにPower Toolsというスタックがあります。これは、非常に強力なユーティリティーですが、そのなかにリソースムーバーというカードがあります。

リソースムーバーを使えば、スタック間でリソースをコピーしたり、削除したり、リソースの名前やID番号を変えたりできます。また"snd "（サウンド）リソースを再生したり、アイコンやカーソルやピクチュアのリソースを表示したりもできます。

"HyperCard Lessons"は初心者から開発レベルまで無理なく順序よく教える教材として位置付けられている。この教材はHyperCardの学習のスタンダードを目指したものであるが、それに成功したかどうかはもちろんユーザーの率直な感想に待たなくてはならない。