「物理法則を利用した動的形態のプログラミング言語の開発」

開発者：古堅真彦

1．要約

本プロジェクトでは，コンピュータ画面上のオブジェクトの動きを，主に物理法則（ニュートン力学）によって制御する専用のプログラミング言語の仕様策定とその実行環境の試作を行った．
本プロジェクトによる成果はウェブデザイナーやコンピュータを駆使するアーティストなどいわゆるコンテンツクリエイターと呼ばれる人たちによって活用されることを目的とする．

2．背景および目的

昨今のコンピュータの普及率や処理能力の向上により，コンピュータがモノ作りのツールとしてではなく，メディアそのものとして活用されるシーンが増えてきている．単なるパソコン上のウェブページ画面だけではなく，携帯電話や PDA などその領域は拡大化，複雑化している．そうした中，そのコンテンツの表現手法に「動き」を多用する場面が増えてきている．コンピュータの特徴を考えるとこのような状況は当然であり，この傾向はさらに増加するだろう．そういった際にこの「動き」という要素に注目してこれを操る具体的な仕組みが必要になってくる．
現在のそれは複数の静止画を連続させることによって動きを見せる，いわゆるパラパラアニメーションの方式が主流である．この方式の長所としては「制作者が確実に動きを作り上げることができる」，「緻密な動きをつくることができる」，「目で確認しながら動きを制作できる」などが挙げられる．反面，短所としては「閲覧者の介入が困難（インタラクティブ性に乏しい）」，「作り上げた後の変更が困難」，「言語での制御が困難（専用のプログラミング言語がない）」などが挙げられる．現在ウェブページを代表的にコンピュータ画面をメディアとして扱う動くコンテンツの制作環境はパラパラアニメーション方式を採用したものがほとんどである．そこで，これらの状況を踏まえて，本プロジェクトでは
・パラパラアニメーション方式等の従来の方式にしばられない
・物理法則を知らなくても物理的な動きを表現できる
コンピュータ独自の画面上の動きを制御するプログラミング言語とその実行環境を試作することを目的とする．

3．プロジェクト概要

本プロジェクトでは以下の４点を行った．
・言語仕様調査
・言語仕様開発
・プロトタイプソフトウェア開発
・言語仕様書作成
現在，静止画を作成するためのプログラミング言語は多く存在するが，動き，すなわち時間軸を考慮したプログラミング言語はほとんど存在しない．本プロジェクトはまず，プログラミング言語の現状調査とそこから「動き」との関連性の調査，そしてそれを基に，時間軸をロジカルにとらえ，そこに，動きを表現する機構を言語体系（仕様）として開発し [motionExpress-language]，それを実際に実行，閲覧できるプロトタイプを試作した [motionExpress-viewer]．また，これら２点と言語を編集する機能を併せ持った開発閲覧統合プロトタイプソフトウェア [motionExpress-language software] を試作し，言語仕様の検証を行い，最終的に言語仕様書を作成した．
本プロジェクトを要約するならば「モノ作り指向の，画面上の動きを制御するプログラミング言語とその実行環境の開発」ということができる．数学や物理，また技術的スキルとしてのプログラミング言語を未理解なクリエイターが論理的な（アルゴリズミックな）感性を駆使して画面上に親近感のある動きをプログラミングの手法を駆使して構築できるシステムの開発のプロジェクトである．
本プロジェクトのベースになる研究開発は数年前より開発者が行っている．本プロジェクトではその一部を実現させる．

4．開発内容

本プロジェクトで行った「言語仕様調査」，「言語仕様開発」，「プロトタイプソフトウェア開発」，「言語仕様書作成」についてそれぞれの内容を記述する．

4．1 言語仕様調査

実際の開発に先駆けて，現状の言語仕様調査を行った．まず，原田 PM と懇談し，本プロジェクトに関する既存のプログラミング言語の特徴と本プロジェクトへの取り込み方について議論した．

調査対象は大きく分けて

・コンパイルの方法

・動きを扱う言語仕様

・ソースコード内の時間軸と実行時の時間軸との関連性

の3点であり，これらについて調査を行った．

4．2 言語仕様開発

言語仕様は motionExpress-language と名付けた．ここではその仕様内容について記述する．motionExpress-language 内の記述は大きく分けて

・タイミング

・base

・point

という3つのカテゴリーに分けられる．

タイミング

「タイミング」は時間軸を管理する構文である．あらかじめ決められた構文にそって描画や画面内のオブジェクトを動かす時間を決める．あらかじめ内部に語彙が用意されていて，その語彙に引数を指定することで，タイミングを決定する．語彙は 14 個が用意されている．

例えば，以下のように記述すると

```
point p;
[setup]{
      p = new point();
}

[paint]{
      drawString("motionExpress", p);
}

[time, 3, 10]{
      p.pushTo(mouse);
}
```

[setup]..... 最初，点 p を用意して，

[time, 3, 10] ..... 起動後3秒から10秒までその点を mouse の方向に押す．

[paint] ..... その点の場所に “motionExpress” と描画する

という結果が得られる．

base

base は各タイミング内での作業を管理する構文である．描画領域の属性とオブジェクトの描画がその主な機能になる．内容はプロパティとメソッドから成っている．プロパティはマウスカーソルの位置や画面の大きさなど 10 個から成っており，メソッドは主に描画用のメソッドの 35 個から成っている．メソッドはそれぞれ引数を持っておりそれぞれの属性に応じた描画が可能である．

point

motionExpress-language では動きを点で制御し，その点を基準とした描画で動きを表現するという形態になっている．その点をクラス化したものが point である．

内容はプロパティとメソッドから成っている．プロパティは座標値など８個から成っており，メソッドはその座標を「押す（push）」などの 23 個から成っている．メソッドはそれぞれ引数を持っておりそれぞれの属性に応じた描画が可能である．

４．３ プロトタイプソフトウェア開発

言語仕様を検証するために試作したプロトタイプソフトウェアは大きく分けて

・プログラミング実行，閲覧環境（motionExpress-viewer）

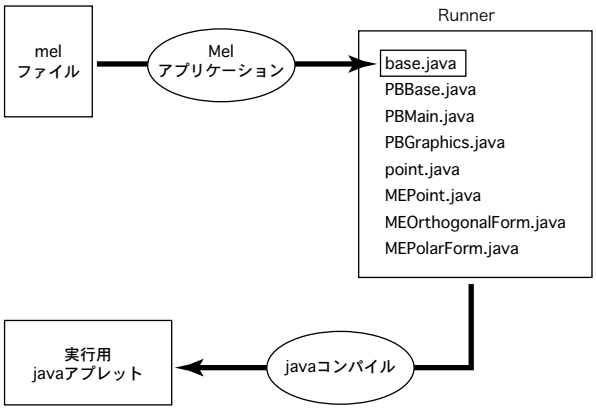
・テキスト編集，デバッグ，実行，閲覧等開発を統合的に扱うソフトウェア（motionExpress-language software）

の２種類である．それぞれについて記述する．

プログラミング実行，閲覧環境（motionExpress-viewer）

motionExpress-language に則って記述された言語は .mel という拡張子がついたテキストファイルとして保存される（mel ファイル）．mel ファイルを実行可能形式に変換するのが motionExpress-viewer である．motionExpress-viewer は具体的には図１のような形態になっている．

まず，mel ファイルを java application である Mel(.class) がその内容を分析解釈し，java のテキストファイル（base.java）に変換する．base.java ファイルは mel ファイル上のタイミングや描画，動きなどの記述を java の記述に対応させて変換したものである．base.java ファイルは既に開発済みの PBBase クラスを拡張したもので，java 内部の煩雑な処理はこの PBBase が担っているため，base.java は mel ファイルの記述のほとんどを１対１に変換できる仕様になっている．



（図1）

また，動きを管理する point であるが，これは java 内部に同名の point クラスが存在し，それと直結する形態になっている．このため .mel ファイル内の point 関連の記述はそのまま base.java 内での point クラスの記述になる．

base.java ファイルは PBBase.java ファイルを含めた java ファイル群 Runner［motionExpress-viewer 実行環境ソフトウェア（＊開発済）］とともにクラスファイルにコンパイルされ java applet となる．

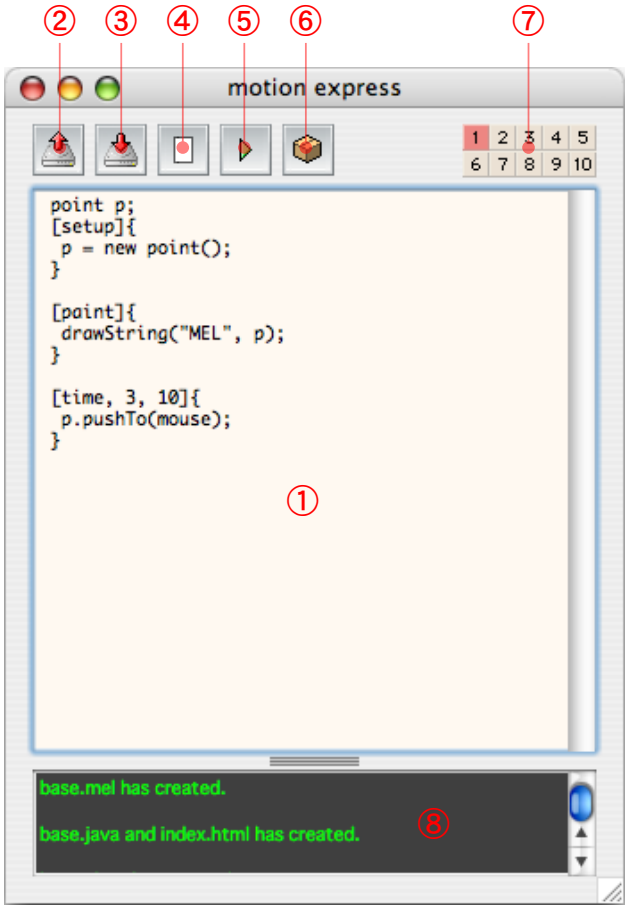
テキスト編集，デバッグ，実行，閲覧等開発を統合的に扱うソフトウェア（motionExpress-language software）

motionExpress-language と motionExpress-viewer を開発環境としてシームレスにつなぐソフトウェアが motionExpress-language software である．図２のようなユーザーインターフェイスを持っている．

一般的機能を有するテキストエディタ上（①）でソースコード開発を行い，そのソースコードはテキストファイル（.mel ファイル）として保存，読み込み（②，③）が出来る．

プレイボタン（⑤）とエクスポートボタン（⑥）には，motionExpress-viewer，つまり書き込んだソースコードを Mel(.

class) に引き渡して base.java ファイル化し，さらにそれを含めた全体を java コンパイルで java applet 化する機能が埋め込まれている．プレイボタンにはその後 applet viewer が自動的に起動し，作成した java applet を閲覧する機能，エクスポートボタンにはあらかじめ設定されたディレクトリに作成した java applet をコピーする機能が埋め込まれている．
ちなみに④は new（ソースコード初期化）ボタン，⑦はワークスペース選択ボタン，⑧はメーッセージコンソールである．



（図2）

これらの開発環境についての詳細はウェブサイト http://www.motionExpress.net/mel/jp/ 上に掲載している．

5．今後の課題，展望

本プロジェクトでは画面上の動きを管理する独自言語仕様とそれを実行するプロトタイプが出来た．ただ，今回試作したものは java バージョンである．ウェブデザインの実情やコンテンツクリエイターの現状を踏まえると Flash バージョンが必要になってくる．今回のプロジェクトで java のプログラミング言語に則ったソースコードに変換する技術ができたので，それを基に汎用性を持つ実製品として，Flash バージョンを開発するのが今後の課題である．また，本成果は実際に制作現場で使ってもらうことが大前提であるので，専用ウェブウページの充実などをはかり，機能とともにさらなる周知の充実をはかりたいと考えている．

6．開発者

本プロジェクトの主たる部分はこのプロジェクトの開発者である古堅真彦が行った．言語仕様書や言語仕様に則った作例のウェブサイト上への掲載を開発補助の大田暁雄君に依頼した．