# ＨＳＰプログラミングによる
# アニメーションスクリーンセーバーの作成

久留美　晶子　，　近藤　智子
名古屋文理大学 情報文化学部 情報文化学科 はせがわ研究室

平成15年2月4日　提出

**要旨：**

現在、様々なスクリーンセーバーがエンターテイメントを目的に開発されている。私たちは、見ているだけでも楽しいスクリーンセーバーを作りたいと思い、プログラミング言語HSPを使って、Microsoft Windows 上で動作するオリジナルスクリーンセーバーを作成した。今回作成したスクリーンセーバー「Small Marine World」は、海の中をアニメーションで表現したものである。

スクリーンセーバーの作成にあたって、まず最初に、図形が移動するスクリーンセーバーを作成し、それをもとに、魚が移動するものを作った。そして、岩、タコ、海藻、サメなどのオブジェクトを増やしていき、これらに動きをつけていった。タコや海藻が自然な動きに見えるようにするため、アニメーション表示する画像の順番を変えて、試行錯誤を繰り返した。次に、魚の数を設定する画面を作り、最後にプレビューのミニ画面を作った。

HSP 言語を用いれば、Window 定義を詳細に記述しなくても、Windows アプリケーションの作成ができ、オブジェクト指向などの概念を用いなくても、アニメーションを含めたマルチメディアアプリケーションが作成できる。また、HSP はメニューからスクリーンセーバー作成ボタンを選ぶだけで実行形式の１つとしてスクリーンセーバーを選択でき、簡単にWindowsのスクリーンセーバーが作成できる。Windowsプログラミングは、難しいというイメージがあるが、HSPを利用すれば個人でも容易に作成することができる。

今回作成したスクリーンセーバーは、アニメーションを楽しむだけではなく、リラクゼーション効果も期待できる。また、魚の個数も設定できるので、実行のバリエーションが楽しめる。

本稿では、作成したスクリーンセーバーとその開発手順について紹介し、HSP による Windows プログラミング､アニメーションの実現､およびスクリーンセーバーの作成についても報告する。

## １．はじめに

スクリーンセーバーは、コンピュータの操作を一定時間何もしないでおくと、画面を保護するために働く機能である。もともとはCRT（Cathode Ray Tube）ディスプレイの焼きつき防止のために開発されたものである。現在では液晶ディスプレイなどが利用されているため、画面の焼きつきはほとんどなくなってきたが、現在でもエンターテイメントの１つとして様々なスクリーンセーバーが利用されている。例えば、Windows に標準で数種類のスクリーンセーバーが用意されている他、様々なオリジナルスクリーンセーバーが商品として、または個人プログラマによる趣味として数多く開発されている。中には、リラクゼーション効果や、ゲーム感覚のものなども

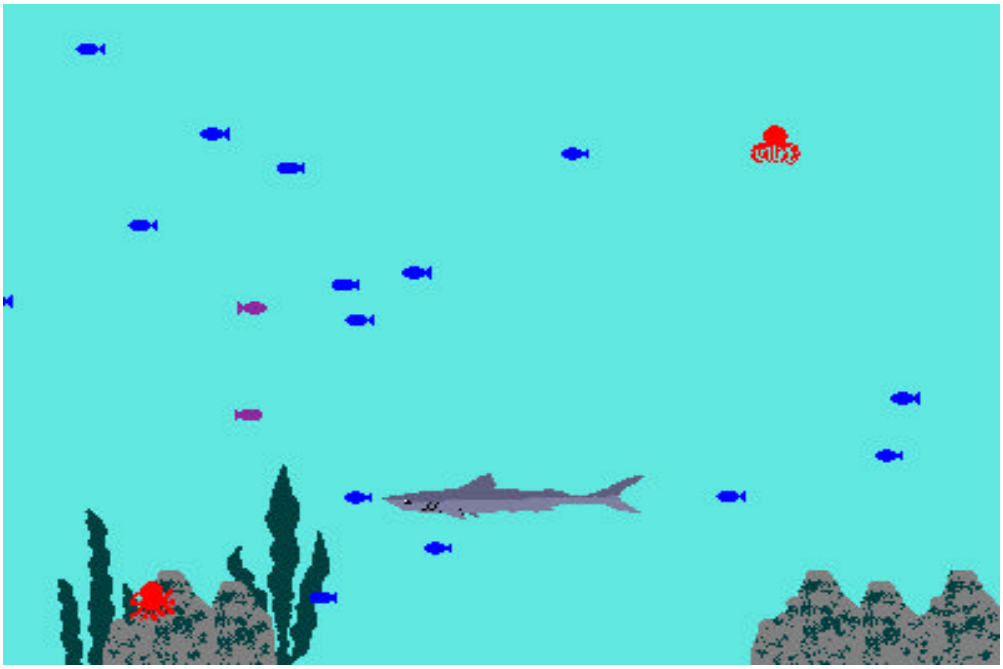

**図１．オリジナルスクリーンセーバー「Small Marine World」完成図**

ある。

そこで、見ているだけでも楽しいスクリーンセーバーを作りたいと思い、プログラミング言語 HSP（Hot Soup Processor）[1) 2)] を使って Windows 上で動作するオリジナルスクリーンセーバー「Small Marine World」を作成した（**図１**）。作品は、海の中の様子をイラスト的なアニメーションで表現したものである。

## ２．Windows スクリーンセーバーの開発

### ２．１．Windows スクリーンセーバーの構成

Windows のスクリーンセーバーは、３つのモードとコマンド解析の４つの部分からできている（**図２**）。

「コマンド解析」は、スクリーンセーバーが起動すると動き、これがどのモードで起動されたかを判別する。どのモードで起動したかによって、次に示す①コンフィグ画面、②プレビュー（ミニ画面）、③フルスクリーン（本体部分）のどれかに振り分けられる。

①コンフィグ画面

これはスクリーンセーバーの各種設定を行うモードである。Windows の「コントロールパネル」にある「画面のプロパティ」の「スクリーンセーバー」タブでスクリーンセーバー設定画面が表示される。設定画面でスクリーンセーバーのプログラムが選択され、設定ボタンが押された場合に、このモードで動く。

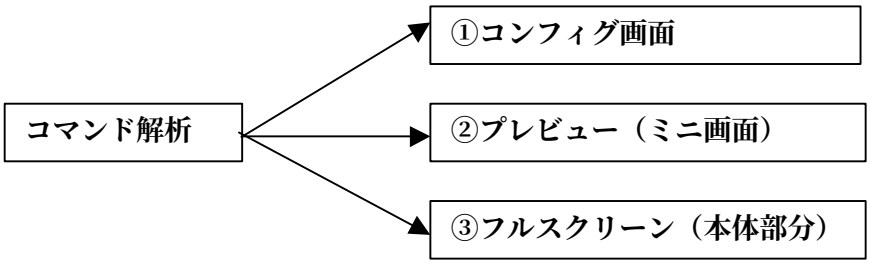


**図２．Windows スクリーンセーバーの構成**

②プレビュー（ミニ画面）

Windows のスクリーンセーバー設定画面の中に表示されるミニ画面のことである。小さな画面にフルスクリーンモードの画面のミニチュアまたはスクリーンセーバーのイメージを説明するオリジナルミニ画面が表示される。ミニ画面の画面表示やアニメーション表示の処理はここに記述する。

③フルスクリーン（本体部分）

スクリーンセーバーとして動く時には、フルスクリーンとして記述された部分が動く。

## ２．２．開発に用いる言語について

Windows プログラミングに適した言語として、一般に Visual Basic や Visual C++などが提供されている。しかし、Visual C++では、Windows の Window 定義を詳細に記述しなければならず、オブジェクト指向などの概念を理解していなければならない。Visual Basic では、Windows のフォームは容易にデザインできるが、スクリーンセーバーやアニメーションの実現には一定のスキルが要求される。それらに比べ、HSP はメニューからスクリーンセーバー作成ボタンを選ぶだけで、簡単に Windows 用のスクリーンセーバーが作成でき、アニメーションの作成も可能である。そのため、今回は HSP で作成することにした。

## ２．３．HSP による Windows プログラミング

HSP は、Windows アプリケーションを開発するためのスクリプト言語である。

Windows 上での画像表示、メニューシステムなどが用意されており、テキストによるスクリプトを書くだけで、容易に構築することができる。HSP では、Windows のウィンドウや、グラフィックやサウンドのようなマルチメディアコンテンツを扱う。また、プログラミングスタイルは、細部ではＣ言語と同様な表現も取り入れながら、全体では BASIC に似た表現（goto 文によるサブルーチンの実現など）を用いており、構造化プログラミングやオブジェクト指向の概念を知らなくても、Windows のマルチメディアアプリケーションが作成できる点が特徴である。HSP では、エディタやコンパイル機能をもつ実行形式として Windows アプリケーションの他に、Windows スクリーンセーバーの形式をメニューから選択することによりファイル出力ができる。

## ３．HSP によるアニメーションの実現

アニメーションの動きは、物体の移動と変形から成り立っている。

## ３．１．物体の移動

物体を移動させるには、まず、１枚の画像を表示して、それを消去し、次に、座標を変えて再表示する。これを繰り返すことで、物体が移動しているように見える。今いる座標と次の座標との距離を変えることで、物体の速度を変化させることができる。HSP では、プログラム中にラベルを設定し、goto 文で反復することによって物体の移動処理を記述できる。

## ３．２．物体の変形

物体を変形させるには、形が少しずつ異なった同じサイズの画像を連続して表示する。HSP では、1 コマごとの画像をすべて並べて１枚の画像データ（キャラクターパターン画像）とし、並べた画像を連続して表示させることができる。プログラミングのときに、並べた画像１コマ分のサイズを指定する。これで、アニメーション表示する画像のサイズが決まる。また、画像の順序を入れ替えて表示させることもできる。

## ４．「Small Marine World」の作成

HSP を用いて作成したオリジナルアニメーションスクリーンセーバー「Small Marine World」について以下に説明する。

## ４．１．開発手順について

スクリーンセーバーの開発にあたっては、まず Windows スクリーンセーバーとしての構成部（**図２**）を作成し、次にグラフィックデ

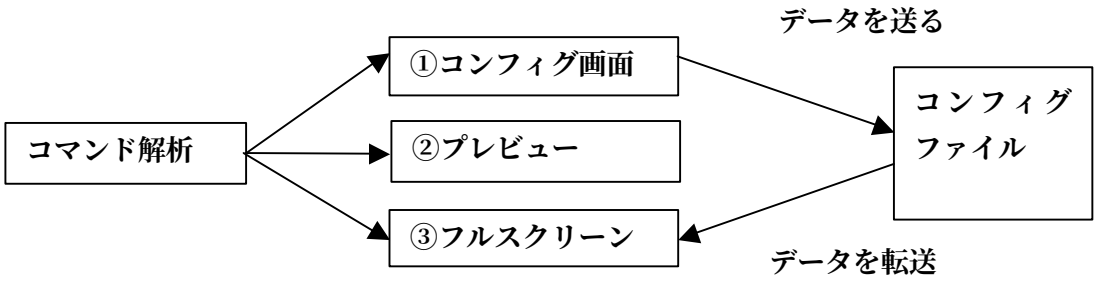


**図３．コンフィグデータのファイル渡し**

```
buffer 2
picload "sakanaL.bmp"
buffer 3
picload "sakanaR.bmp"
dim x,m :dim y,m
dim dx,m :dim dy,m

repeat m
    rnd x.cnt,640 :rnd y.cnt,480
    rnd dx.cnt,3 :rnd dy.cnt,3
    dx.cnt-=8
    dy.cnt=dy.cnt-8
loop
*main
boxf 0,0,ax,ay
repeat m
    x.cnt+=dx.cnt
    if (x.cnt<-50) | (x.cnt>ax) :dx.cnt=-dx.cnt
    y.cnt+=dy.cnt
    if (y.cnt<-50) | (y.cnt>ay) :dy.cnt=-dy.cnt
    pos x.cnt,y.cnt
    if dx.cnt<0 {
                gcopy 2,0,0,32,32
    }
    else {
                gcopy 3,0,0,32,32
    }
loop
goto *main
```

**図４．魚のアニメーションプログラム**

sakanaR　sakanaL

図５．魚の画像

```
buffer 6
picload "kaisou20.bmp"
dim anm1,8
anm1=4,0,3,1,3,0,4,2
i=i+1\8
pos -10,ay-150 :gcopy 6,anm1.i*150,0,150,200
```

図６．海藻のアニメーションプログラム

図７．150×200 ドットのパターンを５つ並べた海藻の画像

```
buffer 5
picload "tako5.bmp"
pt=pt+1\3
pos 600,takoy :gcopy 5,pt*40,0,40,40
```

図８．タコのアニメーションプログラム

図９．40×40 ドットのパターンを
３つ並べたタコの画像

ータを作成しながらアニメーション部分を徐々に追加していく方法をとった。大規模なソフトウェア開発では、ウォーターフォールモデルに基づく商品開発や、オブジェクト指向を取り入れたデザインパターンの利用などが行われるが、個人的な楽しみで作られるスクリーンセーバーのような小規模のソフトウェアでは、このようなスパイラル型の開発の方が適している[3)]。

### ４．２．「Small Marine World」開発の実際

最初に、文献１）の解説を参考に画面上をボールが移動するスクリーンセーバーを作成した。それをもとに、ペイントを用いて描いて作成した魚の画像データ（BMP 形式）を読み込み、魚が移動するアニメーションを作った。さらに、逆向きの魚の画像を追加し、進行方向にあわせて魚の向きが変わるようにプログラムを変更した。次に、岩、タコ、海藻、サメの画像を増やしていき、それらに動きをつけていった。タコや海藻が自然な動きに見えるようにするため、アニメーション表示する画像の順番を変えて、試行錯誤を繰り返した。

次に、魚の数を設定する画面を作った。コンフィグ画面で設定された魚の個数は、図３のようにファイル渡しによってフルスクリーンモードのプログラムに反映され、指定の数だけ魚が表示されるようにした。

最後に、プレビューのミニ画面を作った。フルスクリーン用の画像データをそのまま用いてプレビュー窓に縮小表示すると、画像にひずみが生じる。そのため、フルスクリーン用の画像データを修正して、あらかじめ 152×112 ドットの画像を作った。さらにプログ

ラムでは、念のため、プレビュー窓のサイズを取得し、一度、仮想画面に画像を読み込み、gzoom 命令でプレビュー窓のサイズに合わせてズームコピー（サイズを変更しながらコピー）した[4]。

これらの制作日程の記録を、**表１**に示す。

また、以下に（１）移動する物体、（２）変形する物体、（３）変形しながら移動する物体、の３パターンのアニメーションの実現について実際のプログラムに沿って説明する。

（１）魚のアニメーション（移動する物体）

**図４**の５行目以降の変数 m は、魚の数を表している。５～６行目の dim 命令で魚の座標の変数 x,y および dx,dy を魚の数だけ配列として用意している。次に、９～１０行目の rnd 命令で各魚の初期座標を乱数によって配列変数 x,y に設定し、同様に dx,dy にはそれぞれの魚の移動量の値を設定する。ラベル＊main から始まるメインルーチンでは、まず、boxf 命令で画面全体をクリアする。repeat～loop 命令中で配列変数の番号をカウントし、それぞれの魚の座標に移動の計算をして、魚の画像（**図５**）の表示処理をしている。goto 命令でラベル「*main」に戻り、メインルーチンの処理を繰り返す。

（２）海藻のアニメーション（変形する物体）

**図６**に示すようにウィンドウ ID６（HSP では、０～３１までの ID 番号を付けて、最大３２個のウィンドウ画面を作れる）の buffer にキャラクターパターン画像を取り込み、アニメーションさせる順序を表す。パターン番号を配列変数 anm１に入れておく。

メインルーチン内では、配列番号をカウントし、８回で１ループさせている（**図６**）。配列

**表１． 開発の記録**

| | |
|---|---|
| ～4 月 | Windows スクリーンセーバーの開発法について調査 |
| 4/16 | HSP プログラミング開始 |
| 5/22 | ボールが移動するスクリーンセーバーを作成 |
| 6/19 | 図形が移動していくスクリーンセーバーを作成 |
| 9/30 | 図形のプログラムを元に魚のスクリーンセーバー作成開始 |
| 10/2 | 違う色の魚の画像を追加・左右に進む魚ができる |
| 10/23 | 岩の画像を追加 |
| 10/29 | 魚の個数設定の画面を作成 |
| 11/6 | タコの画像を追加 |
| 11/7 | タコに動きをつける |
| 11/11 | 海藻の画像を追加 |
| 11/15 | 海藻の動きを修正 |
| 11/26 | サメの画像を追加 |
| 11/28 | サメの動きを修正・サメと魚の速度を変化 |
| 12/5 | コンフィグ画面を修正・スクリーンセーバーを起動させると内部エラー発生 |
| 12/10 | 魚の個数設定のプログラムを修正 |
| 12/12 | プレビューのミニ画面を作成・エラーが発生・タコの動きと速度を修正 |
| 12/17 | プレビューのミニ画面用の画像を作成 |
| 12/19 | プレビューのミニ画面ができる |
| 12/20 | 完成 |

変数の値から、ID６のコピー元位置を算定しgcopyによってID０（メイン画面）にパターン画像（**図７**）をコピーする。この方法で海藻の動きを表現した。

（３）タコのアニメーション（変形しながら移動する物体）

前述の方法を組み合わせることで、変形しながら移動するアニメーションを実現できる。

実際のプログラム（**図８**）では、足を動かしながら移動するタコのアニメーションを作成した。**図８**に示すように、まず、ウィンドウID５にパターン画像（**図９**）を読み込んでおいて、メイン画面（ID０）に、gcopy 命令で４０×４０ドットのパターンを順番にコピーしてアニメーションさせている。**図８**の３行目の計算式で、コピーするパターン番号(変数 pt）を、０,１,２,０,１,２…とループさせている。

## ５．まとめ

「Small Marine World」を開発し、アニメーションスクリーンセーバーとして完成させた。Windows プログラミングは、難しいというイメージがあるが、HSP を利用すれば個人でも容易に作成することができる。今回作成したスクリーンセーバーは、海の中をアニメーションで表現したもので、リラクゼーション効果も期待できる。また、魚の個数も設定できるので、個人でも楽しめる。今後、サウンドや色の設定などができるようにして、より楽しめるものにしたい。

## [参考文献]

１）おにたま、悠黒喧史、奥山喜正：「HSP Windows95/98/2000 プログラミング入門」, 株式会社秀和システム,(2001)

２）おにたま、悠黒喧史、うすあじ：「HSP 2.55 Windows95/98/2000/Me/XP スクリプトプログラミング逆引きテクニック」,株式会社秀和システム,(2001)

３）Mint（経営情報研究会）：「図解でわかるソフトウェア開発のすべて」,株式会社日本実業出版社,(2000)

４）ホームページ
http://www.geocities.co.jp/SiliconValley-SanJose/1578/title.htm