本稿は Linux Japan 誌 1998 年 11 月号に掲載された記事に補筆修正したものです．

スクリーンセーバーで Go! 2

スクリーンセーバーで楽しんでいますか？ というのもちょっと変かもしれませんね．普通，しばらく席を離れている間にスクリーンセーバーが立ち上がるので，本人は見ていない筈だからです．それでも， 戻ってきた時に『ああっ，動いている，焼付け防止のために働いてくれてありがとうマルチちゃん』と感謝しつつ，しばらく画面を眺めるくらいの余裕が欲しいですね．でも，何度も同じキャラクターを見ていると飽きが来てしまうのも人間の常で，しかたないことと思います．マルチちゃんをもっとリアルな画像にしたいとか，xfishtank で猫や犬を泳がせたいとかいう気持ちになる人もいるでしょう．靴も二足を交替で使うと一足の 2 倍以上長持ちします．というわけで，アニメーションモジュールも沢山あればあるほど飽きが来ない筈です?... トニカク，そんな飽きっぽい貴方のために，簡単なアニメーション（静止画が動きまわる程度ですが）の自作方法を紹介したいと思います．

静止画をランダムに表示するには

リスト 1　静止画をスクリーンのランダムな位置に表示させるスクリプト．

```
#!/bin/bash

IMAGE=$1
IW=‘identify $1 |awk ’{print $2}’|tr x+ ’  ’ \
                          |awk ’{print $1}’‘
IH=‘identify $1 |awk ’{print $2}’|tr x+ ’  ’ \
                          |awk ’{print $2}’‘
SW=‘xwininfo -root |awk ’/geom/{print $2}’ \
               |tr x+ ’  ’ |awk ’{print $1}’‘
SH=‘xwininfo -root |awk ’/geom/{print $2}’ \
               |tr x+ ’  ’ |awk ’{print $2}’‘

X=$[$RANDOM % $[$SW-$IW]]
Y=$[$RANDOM % $[$SH-$IH]]

xloadimage -geometry "$IW"x"$IH"+"$X"+"$Y" \
                            -quiet $IMAGE
```

ちょっと話が戻りますが，前回 xloadimage などの静止画表示ツールは表示する位置を変化させなければスクリーンセーバーとしては失格と言いました．それでは余りに酷なので，救いの手を差し延べましょう．UNIX 使いなら,『ランダムに位置を変えて表示するシェルスクリプトが作れるはずだ』と考えるでしょう．もちろんで表示位置を指定するオプションが使えることが前提です．例えば，xloadimage は X 標準のオプション `-geometry w×h±x±y` を理解しますから，スクリーンサイズ（=ルートウィンドウの大きさ）を得るために xwininfo，画像の大きさを知るために ImageMagick の identify を併用して，ちょっと無理矢理ですが，**リスト 1** の内容のシェルスクリプト（名前を xload.sh とします `chmod +x xload.sh` を忘れないでください）を作って，

```
xload.sh  imagefile
```

と実行すれば，ランダムな位置に *imagefile* を表示します．こんな意地っ張りのスクリプトを組むのも (^^;, UNIX ならではの楽しみですね．

データを入れ換える

さて今回のテーマに戻りますが，いきなり一から作るのも大変なので，簡単にできるところから始めましょう．画像データを入れ換えるというのは，最も簡単な方法です．なにしろ，ほとんど X のプログラムを知らなくてもいいのですから．そんな超初心者のために，最初からユーザーカスタマイズできるように考えられているものもあります．xfishtank が将にその例でして，つくづく良くできたプログラムなあと感心します．

xfishtank

xfishtank では部分的に僅かに異なる 2 つの画像を組み合わせることができます． 2 つの画像を交互に表示しつつ (2 frame animation)，スクリーン内を動き回るよう出来ているのです．もちろん同じ画像を使っても構いません，その場合には，静止画像がスクリーンを動き回ることになります．

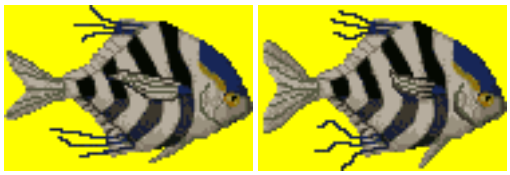

図 1　fish 画像形式　*.h　には 2 枚のフレームが入っています．

画像の形式は独自形式のもので，FISH(拡張子 .h) 形式と呼ばれます． それらは，xfishtank を展開したディレクトリのサブディレクトリ **fishmap/** にあります．まず，ここに自分の好みの画像を FISH 形式に変換して

置かなければなりません．また，ルートディレクトリには，実行ファイル内に取り込む画像のリスト **FishList** がありますから，項目を書き換える必要があります．この２つの作業が全てで，あとは make 一発です．簡単ですね．

では，早速やってみましょう．適当な作業ディレクトリ（ここでは/tmp　とします）で TrueColor の X サーバーでも動くようパッチの当たった xfishtank-2.1tp.tar.gz を伸長します．

```
tar zxvf xfishtank-2.1tp.tar.gz
```

xfishtank-2.1tp に移って，ls を実行した結果を示します．

```
FishList            bubbles.h      medcut.c
Imakefile           compact.h      medcut.h
Makefile            fishmaps/      pcfshtofish/
README              fishtogif/     rasttofish/
README.Linux        gifread.c      read.c
README.TrueColor    giftofish/     vroot.h
README.Why.2.1tp    gofish/        xfish.c
Version             makeh.c
```

FishList と fishmap/ が確認できますね．続いて画像の作成です．まずサブディレクトリ fishmap/ に移ってください．そして xfishtank で泳がせたい画像を GIF 形式で持ってきます．ただし，**同じ大きさと同じ背景色（左上隅の色をそれと見なします）**のものです．と言われても困るかもしれませんので，X に標準配布の『目玉親父』をつかいましょう．/usr/include/X11/pixmaps

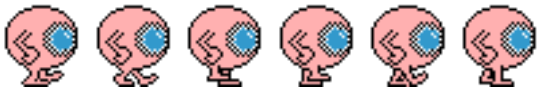

図 2　X に標準配布の目玉親父．

にある，eye1.xpm と eye2.xpm をコピーします．そして GIF 形式への変換を行います．ImageMagick の “convert” を使って

```
convert eye1.xpm eye1.gif87
```

などとして eye1.gif87 と eye2.gif87 を作ります．さらに FISH 形式への変換をします．xfishtank に付属している変換ツール **giftofish** を使って

```
../giftofish eye1.gif87 eye2.gif87
```

とすると，eye.h が出来ます．giftofish は サブディレクトリ giftofish/ で予め作成しておきましょう（xmkmf -a; make で出来ます）．

次に，FishList の編集です．第 1 行が画像の数，2 行目以降に名前を記します．ここではたった 1 つ eye を登録してみましょう．

```
1
eye
```

となります．これで準備万端，

```
make clean
xmkmf -a
make
```

しましょう．目玉親父がうようよ動きまわる xfishtank の出来上がりです．この方法で筆者は Linux の産みの親 Linus 氏と，マスコットのペンギンを含んだ xfishpeng を作成してネットに公開したところ．... どうも，『センス悪ー』らしく反響はありませんでした（T.T）

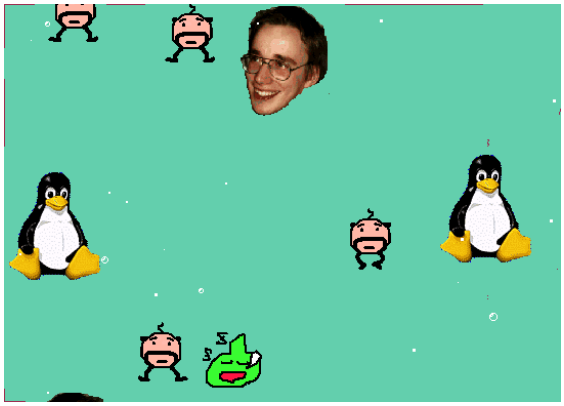

図 3　xfishpeng の実行画面：人気はイマイチでした．

画像を扱うツール：ImageMagick

好みの画像を xfishtank に取り入れるためには，まず形式を GIF（バージョンの違いがあって，古い gif87 に対応してます）で持ってこなければなりません．GIF 形式はネットワーク上で最も一般的な形式ですから心配ないでしょうが，もし他の形式で入手した場合には**変換ツール**が必要となります．画像といえば **xv** がインストールされていると思いますが，これは現在シェアウェアとなっていて，継続使用するには送金が必要ですから，ここでは取り上げません．

筆者は **ImageMagick** の **convert**（そのままの名前ですね) を愛用しています．ただし，JPEG 画像は，読み込みに失敗することがあるので，時々 **djpeg, cjpeg** のお世話になります．ImageMagick はバイナリパッケージがありますから，当座はそれを利用しましょう．**display, animate, import, identify, combine** 等（これまたマンマの名前で，判りやすいっていうかなんていうか）の便利なツールが手に入れられます．

万能ツール display　display を除いた他のツールはコマンドライン上で使うためのものです．ところが display は画像をみるだけではなく，ツール全体の GUI となっています．porsche.xpm を題材にして，少し細かく使い方を見てみましょう．作業用のディレクトリを作成して，そこにコピーすることから始まります．

```
mkdir temp
cd temp
cp /usr/include/X11/pixmaps/porcshe.xpm ./
```

次に，display で画像をスクリーン上に開きます．**画像の上で左ボタンをクリックする**とメインメニューが現れますから，File をクリックしてください，ファイルメニューが現れます．上から 5 番目 Save... をクリックすると，今度はファイル閲覧ダイアログが現れますから，Format をクリックして，すると画像フォーマット選択ダイアログが現れますから，gif87 を選択して，Select をクリックして，再びファイル閲覧ダイアログで File name が porsche.gif87 であることを確認して Save を押します，すると...
あーシンド．あとは直観でできますからどうぞご自由に．コマンドラインなら，たった一行

```
convert porsche.xpm porsche.gif87
```

だけなのに．同じことがコマンドラインで済む場合には，GUI したくない理由お判りいただけましたか?

折角 display は素晴らしいという説明をしようとしていたところなのに，つい愚痴が出てしまいました．何でも GUI という風潮に常々疑問を抱いているので，GUI の面倒な部分をわざと誇張しましたが，画像を扱う場合は，変換や加工の結果を確認しながら作業したいので，GUI は不可欠でしょう．コマンドライン推奨派の筆者も，これだけは認めざるを得ません．例えば，Image Effect などは，効果の具合をその場で確認しながら（気にいらなければ Undo して）作業しないと話になりませんね．

もっとも，これらの効果の程度は数値で与えることになっていますから，コマンドライン上で

```
convert -swirl 40 porsche.xpm porsche.gif87
```

などと実行することができます（実は display が裏で convert を呼び出して実行させている）から，定型業務をバッチ処理的にこなすことも可能です．

import は使える　スクリーン上の画像をファイルに収めるためのツールは，**スクリーンダンプ**と呼ばれますが，結構画像変換に使えます．というのも，ある画像の一部分を取り出したいときに有効だからです（例え

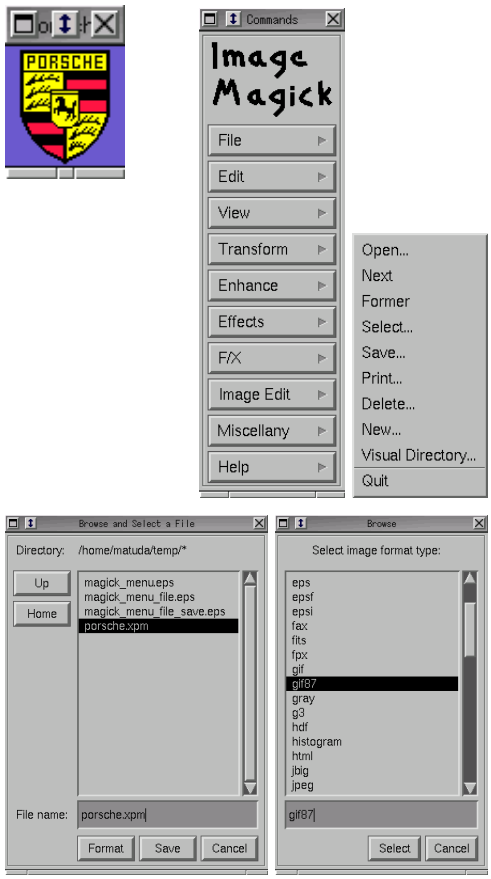

図 4　display の GUI を使って画像を変換する場合に次々現れるメニュー．porsche.xpm，メインメニュー，ファイルサブメニュー，ファイル閲覧ダイアログ，画像フォーマットと選択ダイアログ... ふー．

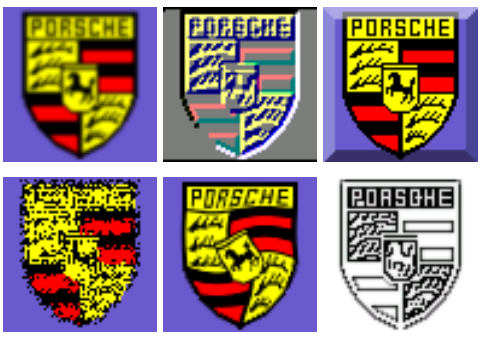

図 5　ImageMagick で用意されているイメージエフェクトの一部．左上から blur，emboss，raise，spread，swirl，chacoal．

ばアニメーションの特定のオブジェクトを狙い打ちするのは難しいです）．画像全体を表示させておいて，収めたい部分だけを指定すればいいのです．大抵のスクリーンダンプ（import，xwd，xgrabsc，xwpick など）は，窓指定と矩形領域指定（xwd は名前通り窓指定のみ）の２通りをサポートしてます．矩形領域の１つの頂点でマウス左ボタンをクリックしたのち，ドラッグ（つまりボタンを押したままカーソルを移動させる）してもう一方の頂点上でボタンを離すと，矩形領域が指定されます．ImageMagick に含まれる import は 16bit 色にも対応（xgrabsc と xwpick は 8bit まで），ボーダー取り込み，フレーム取り込み，実行開始の遅延など，機能が豊富で大変便利です．また，ImageMagick をソースからコンパイルすると楽しいオマケの付いていることが判ります．例えば，サブディレクトリ script/ には wish を用いた GUI フロントエンド xsnap があります．display の grab に比べて，取り込む事前にオプション設定ができる点が多少便利でしょう，起動も display よりやや速いです．

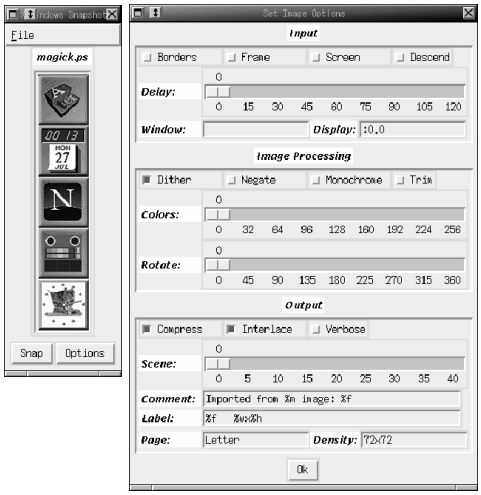

図 6　import に対する wish を使ったフロントエンド xsnap．取り込み直後の確認画面（左），オプション設定ダイアログ（右）．

ハウツー自作プログラムミング

以上，紹介した画像変換ツールを使って，GIF 形式に画像を揃えて自分のお好みの xfishtank が作成できるようになった筈です．すると今度は xfishtank の動きがしっくりこない，自分の思うようにスクリーン上で画像を動かしたいと思うようになります．xfishtank では速さだけはオプション `-r rate` で変えられますが，動き方を変えることはできません．したがって，これ以上は**自分でプログラミングするしかありません**．もちろんそれはとても大変な作業です．しかし，筆者は中学生の頃,『原理も判らず，五球スーパーラジオをキットで組み立てた』タイプの人間です．理屈はあとで結構，なによりも自分で作ってみて動く楽しみを味わってもらいたいのです．別にプロのプログラマーになるわけではないのですから，Linux という玩具でとことん楽しむには,『見よう見まねプログラミング』で十分と言い切ってしまいましょう．

そうは言っても，Linux に触っている限りは

- C 言語でプログラミングの経験が多少ある，常々したいと思っている．

くらいの読者の皆さんを想定します．それプラス X Window System （X ライブラリ）と 画像ファイル（Xpm ライブラリ）についての知識は必要になりますが，その都度説明していくという方針で進みます．

XPM ライブラり

リスト 2　XPM ファイルの構造

```
/* XPM */
static char * plaid[] = {
/* plaid pixmap
 * width height ncolors chars_per_pixel */
"22 22 4 2 ",
"   c red   m white   s s_red ",
"Y  c yellow   m black   s s_yellow ",
"+  c yellow   m white   s s_yellow ",
"x  c black   m black   s s_black ",
/* pixels */
"x   x   x x x   x   x x x x x x + x x x x x ",
"  x   x   x   x   x   x x x x x x x x x x x ",
"x   x   x x x   x   x x x x x x + x x x x x ",
"  x   x   x   x   x   x x x x x x x x x x x ",
"x   x   x x x   x   x x x x x x + x x x x x ",
"Y Y Y Y Y x Y Y Y Y Y + x + x + x + x + x + ",
"x   x   x x x   x   x x x x x x + x x x x x ",
"  x   x   x   x   x   x x x x x x x x x x x ",
"x   x   x x x   x   x x x x x x + x x x x x ",
"  x   x   x   x   x   x x x x x x x x x x x ",
"x   x   x x x   x   x x x x x x + x x x x x ",
"          x           x   x   x Y x   x   x ",
"          x             x   x   Y   x   x   ",
"          x           x   x   x Y x   x   x ",
"          x             x   x   Y   x   x   ",
"          x           x   x   x Y x   x   x ",
"x x x x x x x x x x x x x x x x x x x x x x ",
"          x           x   x   x Y x   x   x ",
"          x             x   x   Y   x   x   ",
"          x           x   x   x Y x   x   x ",
"          x             x   x   Y   x   x   ",
"          x           x   x   x Y x   x   x "
} ;
```

X で標準のカラー画像形式はなんでしょうか? X の基本ライブラリでは白黒のビットマップを XBM 形式

で扱う関数群がありますから，白黒画像は XBM が基本で間違いありません．ところがカラー画像は XImage 構造体が定義されているだけで，既定の保存形式はないのです（XWD は古くからあった形式ですが，xwd, xwud 以外のツールのデータとして使われる例を知りません）．XBM 形式に類似している XPM(X PixMap) 形式が事実上の標準となっていて，libXpm ライブラリを GROUPE BULL というフランスに本拠を置く団体が保守・開発しています [1]W[3]．XPM の特徴は，Xlib て定義されている XBM を扱う関数と類似の関数群で扱えることにあります．X の教科書で XBM に関する記述があったら，それは関数名をちょっと変えて XPM にもそのままあてはまります．つぎに，なんといってもテキストファイルであることでしょう．もともと，C 言語のソースファイルに`#inculde` して使えるように考えられていますから当然ですね．したがって，mule などのエディタで編集が可能ですし，なにより内部構造が非常に単純ですから，日曜プログラマーでも十分手に負える範囲です．

ちょっと，データ構造を見てみましょう．**リスト 2** は，/usr/include/X11/pixmaps にある Plaid.xpm の中身（実は，xpm の配布に含まれる解説書の例題）です．データの型は文字列の配列 char *name[ ]={" "," ",....}; となっていて，４つのパートに分れています．最初の 1 行は，4 つの 10 進数 "width height ncolors cpp" が記されていて，画像の幅，高さ，色数，1 画素を表す文字数を表しています．この例では，22 × 22 の大きさで，色は 4 色，2 文字で 1 つの画素を表現していることになります．続いては色の定義で，文字列と色の対応を 1 行づつ記述していきます．1 行目は

```
"   c red m white s s_red "
```

となっていますが，これは空白文字 2 つ（1 つではありません）を "red" に対応させるという意味です．カラーの場合にはここまでで十分ですが，XPM では X サーバーが mono や grayscale である場合にも表示できるように考えられています．m は mono（他にも g は grayscale）の場合の色を指定します．s は固定の色ではなくシンボリックな色名を表します．コメント行 `/* pixels */` 行以降がデータとなります．ピクセルの幅分の文字配列（この例では，2 × 22 で 44 文字です）が，ピクセルの高さ数に等しい行数含まれます（この例では 22 行）．4 つ目のパートは拡張機能を記述することになっています．Plaid.xpm には該当する部分がありません．以上，言を労さずとも，ほとんど一目瞭然ですね．

透明色　画像データは基本的に矩形です．すなわちプログラム上必ず矩形領域として扱うことになります．ところが，X のクライアントでも emiclock や oneko のように任意の外形を持っているものがあります．これは矩形領域内で新しく描画したい部分を限定し，その他の部分は背景画像が透けて見えるようにする，いわゆる**クリッピング**という方法を用いた結果です．また，

図 7　クリッピング：原図（左）でマスク（中）された部分だけ塗り，残りの背景はそのまま．すなわち，背景が透けて見える（右）．

X ではシェイプ拡張という名前でも呼ばれます．クリッピングやシェイプ拡張については後々のお楽しみとしましょう．XPM ではこの透明部分を表す色が定義されていて 'None' あるいは 'none' と記すことになっています．例えば，

```
grep None /usr/include/X11/pixmaps/*.xpm
```

とすると，どの画像に 'None' が含まれているか判りますから，それをスクリーンに表示しますと，あれれ，透明になってない．そうです，display も xli(xloadimage) も xv さえも 'None' を正しく扱いません．画像ビューアーではないのですが，tgif は期待通りの表示をします（あまり有名ではありませんが，Motif を使ったファイルマネージャー xkhfm も大丈夫です）．

後で説明しますが，Xpm ライブラリを用いるとクリッピングが容易に実現します．したがって，複雑な外形の画像が簡単にスクリーン上に描けますから，楽しみにしていてください．

仮想ルートについて

今回は紙面が少なくなってしまいましたので，X でスクリーンセーバー用のグラフィックを描くプログラムについての紹介は無理です．残りの紙面を使って予備知識として，**仮想ルート**の考え方を説明します．

前回紹介したアニメーションはスクリーンセーバー専用というわけではなく，例えば

```
xfishtank
```

と起動すると，それまでのスクリーン画像が消えて全体が海の中になり，魚が泳ぎまわります．ただし，アプリケーションの窓は前面に居残ります．`Ctrl-C` で終了さ

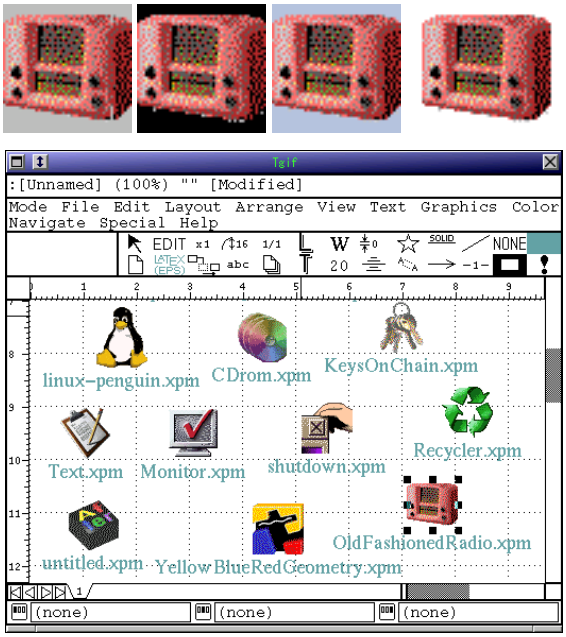

図 8　代表的画像ビューアの，透明色の定義 'None' のサポート状況．左上から display，xli，xv は未サポート（xloadimage はそもそも表示できません）．tgif の BrowseXPixmap はサポートしています（下図）．

せると，スクリーンは元の画像に戻ります．このように，xfishtank はデフォルトの設定がスクリーン一杯に描画することになっています．では，xscreensaver に含まれる bubbles をオプションをなしで起動するとどうなるでしょうか？ 小さな窓が現れるだけです．これではスクリーンセーバーにはなりません，したがってスクリーン一杯に描く指定が必要です．そこで，XScreenSaver に記述例があるように，オプション "`-root`" が必要になるのです．では

```
bubbles -root
```

で起動しましょう，今度はアプリケーションの窓の背景としてスクリーン一杯に泡がでました．`Ctrl-C` で終了します．あれ，背景に泡が残ったままとなってしまいます．でも，これは正しい動作なのです．なぜって，それはルートウィンドウに描画する（書き換えてしまう）というオプションを付けたからです．では，ルートウィンドウを書き換えないようにスクリーン一杯に起動するにはどうすればいいのでしょうか？ 例えば

```
bubbles -geometry 1024x768
```

のようにスクリーンサイズを指定する方法があります．しかし，これではウィンドウマネージャーによって装飾を施されてしまいます．結論は，bubbles の場合に

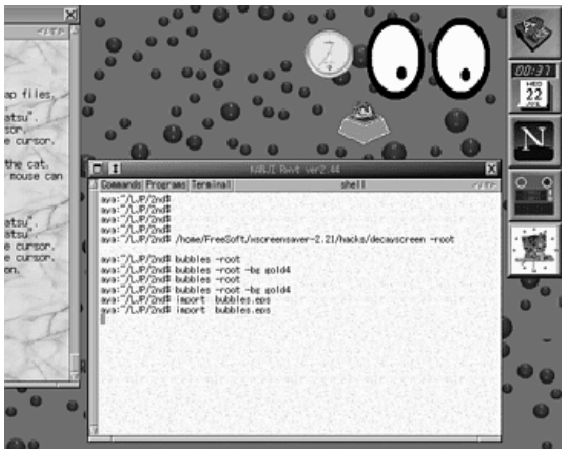

図 9　bubbles -root で起動して終了すると，ルートウィンドウに泡が残ります．画面右上には，シェイプ拡張を使って矩形以外の外形を持つ，xeyes，pclock，oneko を動かしました．

は不可能なのです．では xfishtank はこの問題をどう解決しているのでしょうか．答は『仮想ルートウィンドウ』です．ウィンドウマネージャーによる装飾を無視したスクリーン一杯のウィンドウでスクリーンを覆い，その上に描画しているのです．もちろん，ルートウィンドウは元のまま保存されます．

ちょっとごちゃごちゃしましたから，まとめましょう．スクリーンセーバーとして機能させるため，スクリーン一杯に描くには，ルートウィンドウを直接書き換えるのが手っ取り速いのですが，それでは，終了後に跡が残ります．跡を残さないようにスクリーン一杯に描くには，『仮想ルートウィンドウ』を設ければいいということです．

次回予告編

というわけで，プログラムを紹介するつもりが，『たった 1 球を投げるのに 30 分をかける巨人の星』のような展開になってしまいました．1 球の重み，イヤ，1 つのプログラムを書くに当たっては，相応の予備知識が必要なので我慢してください．次回は必ず魔球を投げる，もとい，アニメーションプログラムの原型を紹介します．

プログラムのインストール情報

- xfishtank はソースが必要です，しかも Linux でコンパイルに失敗しないように，patch があたったものが RedHat の SRPMS にあります．Linux Japan の付録 CD-ROM を活用しましょう．

- xpm3.4 ライブラリのソースも xfishtank と同様です.
- ImageMagick は画像ライブラリがきちんとインストールされていないとコンパイルに失敗しますので，本文中にも述べましたが，バイナリ配布を利用しましょう．RedHat-4.* の RPMS が使いやすいでしょう．Slackware な方はちょっと面倒です（実は筆者は Slack 派）．基本的には変換ユティリティ rpm2targz を用いて *.tgz を作成し，/ から展開すればその rpm はほとんどの場合 OK です．ところが，ImageMagick は依存関係のあるライブラリファイルが RedHat 版では 4 つもあります．libtiff と libjpeg と libpng と zlib です．これらも

```
rpm → tgz： rpm2targz *.rpm

tgz → 展開： / で tar zxvf *.tgz
```

の手順でインストールしましょう．ライブラリを更新した場合には，

```
/sbin/ldconfig
```

を忘れないでください．さらに，RedHat-5.0 から持ってこようとすると，ld-linux.so.2，limb.so.6，libc.so.6 等の基本的なライブラリを要求される場合があります．これはかなり危険ですから，慎重に（上書きだけはしない．元に戻せるよう古いライブラリのコピーを保管しておく等の対策をして）インストールしましょう.

参考文献

[1] Xpm の開発グループ
http://koala.ilog.fr/lehors/xpm.html