ServantWare

Animation Kit Help

The ServantWare Animation Kit is comprised of four types of animation: cell, cursor, curtain and scroll. The INVERT Color Wheel tool helps identify colors for cell animation. The source code for the samples is encrypted. By registering you can gain the key to the encryption scheme and quickly add animation to your applications.

Registration Instructions

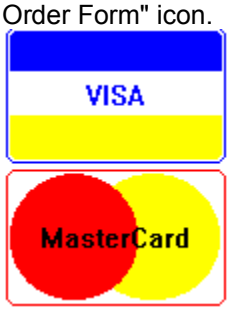
- Registering
- Source Code

Sample Information

- CELL.EXE
- COLORS.EXE
- CURSOR.EXE
- CURTAIN.EXE
- SCROLL.EXE

Registering the Animation Kit

Note: This information can be printed by opening ORDERFRM.WRI or double-clicking on the "Animation Order Form" icon.



The Animation Kit can be registered with your VISA or MasterCard by calling 800-444-5457

You can also mail a check or money order for $20.00 (US) to:

ServantWare
1426 Brookfield
Ann Arbor, MI 48103
U.S.A.

Please make checks payable to: ServantWare

By registering, you will receive a registration code to decrypt the source code for all examples. Technical support is also available via CompuServe [76636,1166] for registered users.

Trial Use License
ASP Ombudsman Statement
Warranty
Author Information

Trial Use License

ServantWare hereby grants you a limited license to use this software for evaluation purposes for a period not to exceed thirty (30) days. If you intend to continue using this software (and/or it's documentation) after the thirty (30) day evaluation period, you MUST make a registration payment to ServantWare. Using this software after the thirty (30) day evaluation period, without registering the software is a violation of the terms of this limited license.

ASP Ombudsman Statement

This program is produced by ServantWare, a member of the Association of Shareware Professionals (ASP). ASP wants to make sure that the shareware principle works for you. If you are unable to resolve a shareware-related problem with an ASP member by contacting the member directly, ASP may be able to help. The ASP Ombudsman can help you resolve a dispute or problem with an ASP member, but does not provide technical support for members' products. Please write to:

ASP Ombudsman
545 Grover Road
Muskegon, MI 49442-9427

or send a Compuserve message via CompuServe Mail to ASP Ombudsman 70007,3536.

Warranty Information

THE ANIMATION KIT PROVIDED HEREUNDER (COLLECTIVELY REFERRED TO AS "SOFTWARE") IS PROVIDED AS IS WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL SERVANTWARE OR ITS SUPPLIERS BE LIABLE FOR ANY DAMAGES WHATSOEVER INCLUDING DIRECT, INDIRECT, INCIDENTAL, CONSEQUENTIAL, LOSS OF BUSINESS PROFITS OR SPECIAL DAMAGES, EVEN IF SERVANTWARE HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. SOME STATES DO NOT ALLOW THE EXCLUSION OR LIMITATION OF LIABILITY FOR CONSEQUENTIAL OR INCIDENTAL DAMAGES SO THE FOREGOING LIMITATION MAY NOT APPLY.

Author Information

ServantWare is a software company owned and operated by Paul Ligeski. At ServantWare we strive to uphold the character quality that is part of our name: service. This concept is fundamental to our approach to product development, production and marketing.

Please feel free to contact me at any time if you have any questions, comments or suggestions. I can be reached at:

Paul Ligeski
President and CEO
ServantWare
1426 Brookfield
Ann Arbor, MI 48103
USA

Voice: (313) 741-1711 (evenings)
CompuServe: 76636,1166
Internet: 76636,1166@compuserve.com

Animation Kit Decryption Utility (DECRYPT.EXE)

This utility decrypts the source code for the Animation Kit into readable form. Executing the program requires a password and a starting directory. It should only be used in conjunction with the "Help\About Animation Kit..." menu item on each of the sample executables registering.

The samples were compiled with MSC v6.0a.

Cell Animation (CELL.EXE)

Cell animation is the process of simulating movement through an erase/move/redraw animation procedure. CELL.EXE moves a camel around the screen through cell animation.

Instructions
To change the direction of the camel, select the "Direction\Horizontal" and "Direction\ Vertical" menu items. Each menu item is a toggle. To get the camel to move diagonally, select both menu items.

To change the speed of the camel, select the "Speed" menu item. Enter the number of horzontal and vertical pixels the camel should move for each time interval. NOTE: values higher than 25 will destroy the animation effect.

Description
A timer event calls MoveBitmap to erase the bitmap, calculate its new position and redraw it. The bitmap is erased by drawing the same bitmap on top of itself. This is accomplished by using the INVERT flag to turn all the image and dead space pixels in the bitmap to the background color. Once the bitmap is erased, the new position is calculated. By storing more parameters such as vertical direction and speed, the animation can become more robust by bouncing the bitmap off walls or wrapping the bitmap around the window. After the new position is calculated, the bitmap is redrawn using the same BitBlt call that erased the bitmap. Windows executes these three steps so quickly that it looks like the bitmap moved.

Related Topics
COLORS.EXE

INVERT

A global RGB value (0x00990066) used in the BitBlt call to display bitmaps.

MoveBitmap

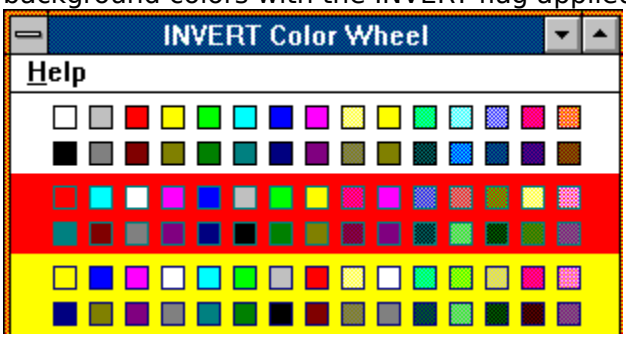
This routine can be found in the file ANIMATE\CELL\CELL.C

INVERT Color Wheel (COLORS.EXE)

The INVERT Color Wheel assists in creating a bitmap useful for animation on colored backgrounds. COLORS.EXE does not do animation itself.

Instructions

The INVERT color wheel displays the Windows 3.1 Image Editor pallete on a variety of solid background colors with the INVERT flag applied.



To use the wheel, choose the color of the bitmap you wish to animate in the white area and choose the background for the client area of your application (for this example, we will choose red on yellow). Find the (red) square in the (yellow) background and match that square's position in the pallete on the white background (light purple). You can look at the CAMEL.BMP for another example.

Description

Since every bitmap is rectangular, there is "dead space" around the image. For example, a bitmap of a round ball has dead space in each corner. The most common solution to insuring that the bitmap looks like a ball is to make the dead space the same color as the background. If the SRCCOPY flag for BitBlt is used and two bitmaps overlap, the dead space in the second bitmap overwrites the image in the first. The INVERT flag gives priority to the image over the dead space in the BitBlt draw, allowing both bitmaps to be fully displayed. The one drawback with INVERT occurs when your background is a color other than white. To make a red ball appear on a blue background, the ball in the bitmap resource must be green.

Related Topics:

CELL.EXE

Windows 3.1 Image Editor

This program, part of the Windows SDK, is in your SDK directory as IMAGEDIT.EXE. You can access it by double-clicking on the "Image Editor" icon in your Program Manager group labeled "Software Development Kit 3.1".

CAMEL.BMP

This image is in the file CELL\CAMEL.BMP

Cursor Animation (CURSOR.EXE)

Cursor animation is the process of simulating movement by changing the display what the user controls on the screen. CURSOR.EXE changes the cursor from a man running right to a man running left and vice versa.

Instructions
Move the cursor in the window. When the mouse moves left the cursor turns into a man running left. When the mouse moves right the cursor turns into a man running right.

Description
The WM_MOUSEMOVE event returns the current position of the mouse. Parsing the long parameter reveals that the low word contains the horizontal position and the high word holds the vertical position. By comparing the current position to the old position you can determine which direction the mouse moved and set the cursor appropriately.

NOTE: When you register the window class, set hCursor to NULL. Otherwise the cursor will flicker.

WM_MOUSEMOVE

This case statement can be found in the MainWndProc routine of the file CURSOR\CURSOR.C

Curtain Animation (CURTAIN.EXE)

Curtain Animation is the process of simulating movement by gradually revealing more and more of an image. CURTAIN.EXE reveals a skyline

Instructions
To change the direction of the movement, select "Center", "Left", "Right", "Up", or "Down" under the "Open" menu item. The screen will repaint moving in the selected direction.

To change the speed of the movement, select the "Delay" menu item. Enter a delay factor in milliseconds. As a rule, a value of 1000 equals one second. NOTE: Values higher than 1000 will destroy the animation effect.

Description
The process involves drawing the strip, moving to the edge of that strip and drawing the next strip. Each strip is three pixels wide and drawn using the BitBlt API call and the SRCCOPY flag.

Opening the bitmap left calls OpenLeft which draws columns starting at the left edge of the image until the width of the bitmap is reached. Conversely, opening right calls OpenRight draws columns starting with the width of the bitmap until the left edge is reached. Similarly, OpenDown draws rows starting at the top edge until the height of the bitmap is reached while OpenUp draws rows starting the with height of the image until the top edge is reached. OpenCenter draws columns starting at the center and calculates the new position of the new strips on either side of what is currently drawn.

OpenLeft

This routine can be found in the file CURTAIN\CURTAIN.C

OpenRight

This routine can be found in the file CURTAIN\CURTAIN.C

OpenDown

This routine can be found in the file CURTAIN\CURTAIN.C

OpenUp

This routine can be found in the file CURTAIN\CURTAIN.C

OpenCenter

This routine can be found in the file CURTAIN\CURTAIN.C

Scroll Animation (SCROLL.EXE)

Scroll animation is the process of simulating movement through moving the background. SCROLL.EXE moves text in a "movie credits" form by moving the background that the text is displayed on.

Instructions
To start the scrolling select the "Start" menu item.

To change the speed of the camel, select the "Speed" menu item. Enter the number of horzontal and vertical pixels the camel should move for each time interval. NOTE: values higher than 25 will destroy the animation effect.

Description
The Scroll routine draws the text to scroll in a predefined rectangle. Next the ScrollWindow API call is envoked with the horizontal direction, vertical direction, and (a second) rectangle defining the complete range of movement for the text. If the horizontal direction parameter is positive, the text will move rightwhile a negative value moves the text left. In the same way, if the vertical direction parameter is positive, the text will move down while a negative value moves the text up. Once the text hits the edge of the rectangle, it will disappear.

Scroll

This routine can be found in the file SCROLL\SCROLL.C

Text to scroll

The text to be scrolled is in the STRINGTABLE section of SCROLL\SCROLL.RC