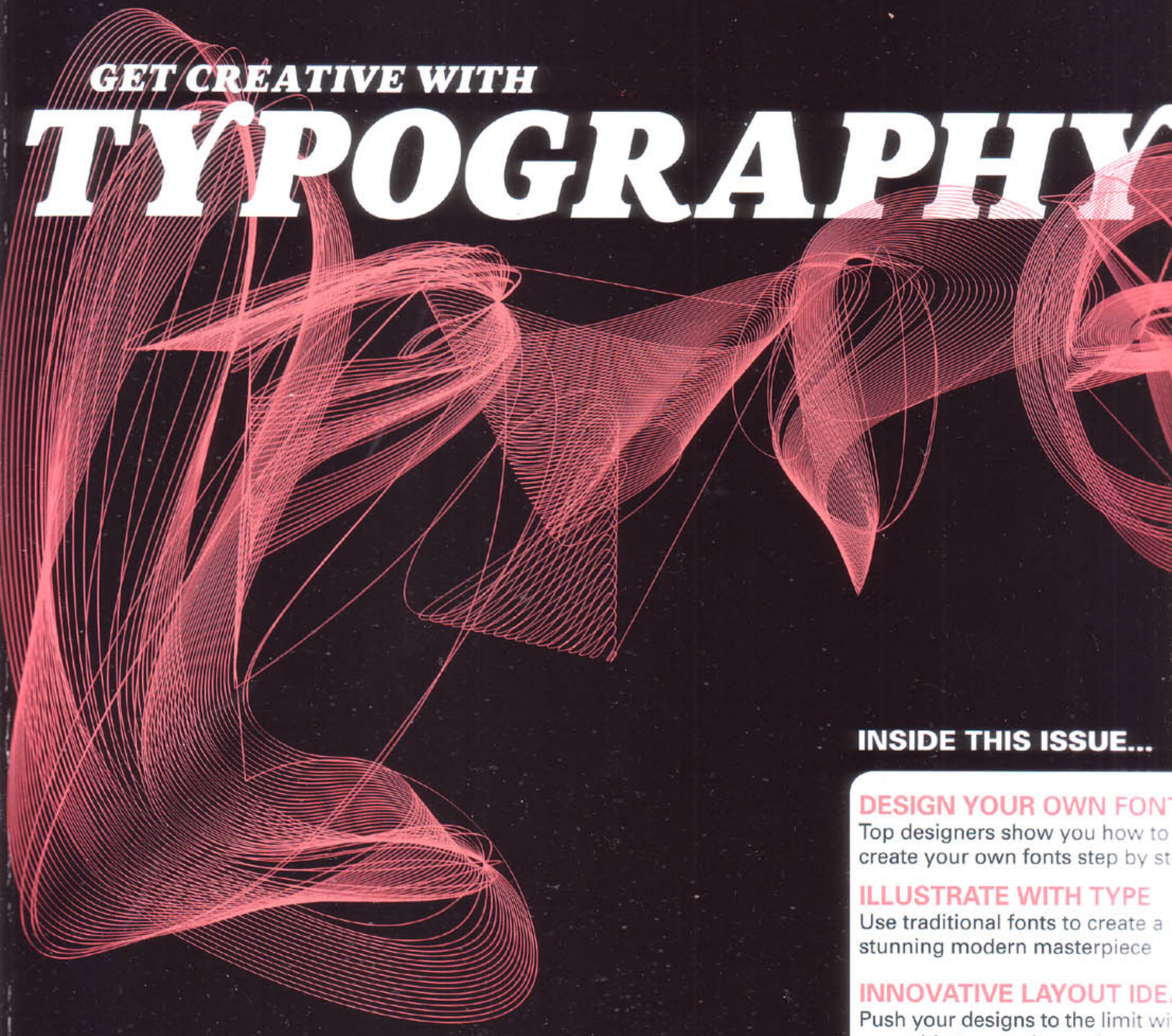
WIN THE ENTIRE MONOTYPE LIBRARY WORTH £2,999! • PLUS 2633 FREE FONTS ON THE

COMPUTER arts projects

THE IN-DEPTH GUIDE FOR DIGITAL CREATIVES



GET CREATIVE WITH TYPOGRAPHY

INSIDE THIS ISSUE...

DESIGN YOUR OWN FONT
Top designers show you how to create your own fonts step by st

ILLUSTRATE WITH TYPE
Use traditional fonts to create a stunning modern masterpiece

INNOVATIVE LAYOUT IDE
Push your designs to the limit wi our guide to experimental typogr

ISSUE 66 XMAS 2004 £6
9 771740 900004 13
UK edition · Printed in the UK

COMPUTER arts projects

Subscriptions – page 56
Future Publishing Ltd, FREEPOST, BS4900, Somerton, Somerset, TA11 6BR
[t] Customer hotline: 0870 444 8455
[f] 01458 271 146
[e] ca.subs@futurenet.co.uk

Editorial [t] 01225 442 244
Rob Carney associate editor
[e] robert.carney@futurenet.co.uk
Ashley Young production editor
[e] caproduction@futurenet.co.uk
Sarah Robson acting art editor
[e] caart@futurenet.co.uk
Johann Chan designer
[e] johann.chan@futurenet.co.uk
Matt Gallimore senior CD editor
[e] matt.gallimore@futurenet.co.uk
Simon Holland CD editor
[e] simon.holland@futurenet.co.uk
Sarah Williams software copyright co-ordinator

CD problems [t] 01225 822 743
Technical support: ca.support@futurenet.co.uk

Contributors
Cover typeface: Jeff Knowles, Research Studios
Contributors: Jason Arber, Max Asare, Dylan Channon, Clusta Design Consultancy, Justin Dike, Silas Dilworth, Christopher Kenworthy, Jeff Knowles, Meglotorch, Mark Porfoid, Xavier Robitaille, Evan Roth, Thomas Schostok, Jason Smith, Jerome Turner.

Printed at: BGP

Contact
Computer Arts Projects, Future Publishing, 30 Monmouth Street, Bath BA1 2BW
[t] 01225 442 244 [f] 01225 732 295
[i] ISDN: 01225 789 293
[w] www.computerarts.co.uk

Advertising [t] 01225 442 244
Lee Haines key account manager
[e] lee.haines@futurenet.co.uk
George Lucas senior sales executive
[e] george.lucas@futurenet.co.uk
Melissa Watkins classified enquiries
[e] melissa.watkins@futurenet.co.uk
Rosa Smith advertising manager
[e] rosa.smith@futurenet.co.uk

Production
Mark Anson production coordinator
[e] mark.anson@futurenet.co.uk

International licensing
Simon Wear international director
[e] simon.wear@futurenet.co.uk
[t] +44 (0) 1225 442 244

Management
Kelley Corten publisher
Dom Beaven publishing director
Gillian Carson senior editor
Roddy Llewellyn senior art editor

Distributed by
Marketforce (UK) Ltd
5th Floor, Low Rise Building, Kings Reach Tower, Stamford Street, London, SE1 9LS
[t] +44 (0) 20 7633 3300

Jason Comber trade marketing manager
[e] jason.comber@futurenet.co.uk

The Future Network
Roger Parry non-executive chairman
Greg Ingham chief executive
John Bowman group finance director
Robert Price managing director

Welcome

Typography is a subject that each and every designer will have to embrace at some point in their career, and it's often left until the last minute by those not so experienced. Throughout this issue, we bring you expert guidance on how to get experimental and creative with typography – taking your design skills further and learning from the best in the business.

We've got everything from creating your own original and stylish font (p18), to taking a boring system font and creating a stunning illustration (p24). We also have features on experimental typography, including such seminal designers as Bruno Maag and Jeff Knowles, along with a host of unmissable tips and techniques on creating dynamic type effects in *Flash*, motion graphics and film titles. Whatever your typographic interest, we've got it covered.

Turn to the contents over the page to see the full listing, and have a look below to get an idea of the best new, and old, faces in typography. Let me know what you think of the issue.

Rob Carney, Associate Editor
rob.carney@futurenet.co.uk

Contributing creatives

COVER TYPEFACE

Jeff Knowles, Research Studios

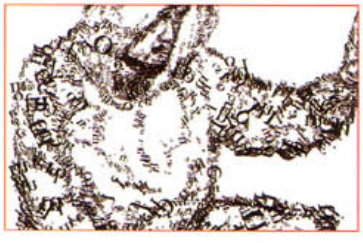
Jeff Knowles' Spiral typeface was originally commissioned for Made in Clerkenwell – an open studio event showcasing the work of many different types of creatives. Jeff, Senior Designer at Research Studios, talks about the typeface in our Experimental Type feature, beginning on page 12.

[w] www.researchstudios.com

Silas Dilworth

Silas Dilworth is designer and copywriter at established foundry T.26. This month, he brings together a panel of world-famous designers to judge which fonts you should, and shouldn't, be using. Turn to page 80 to see the results.

[w] www.t26.com
[w] www.silasdilworth.com



ni9e

ni9e is a collaboration between Evan Roth and Max Asare. This month, the team brings you an outstanding tutorial on creating imagery using digital type in *Flash*. These techniques will show you how to create some of the amazing interactive typographic images found at the website below...

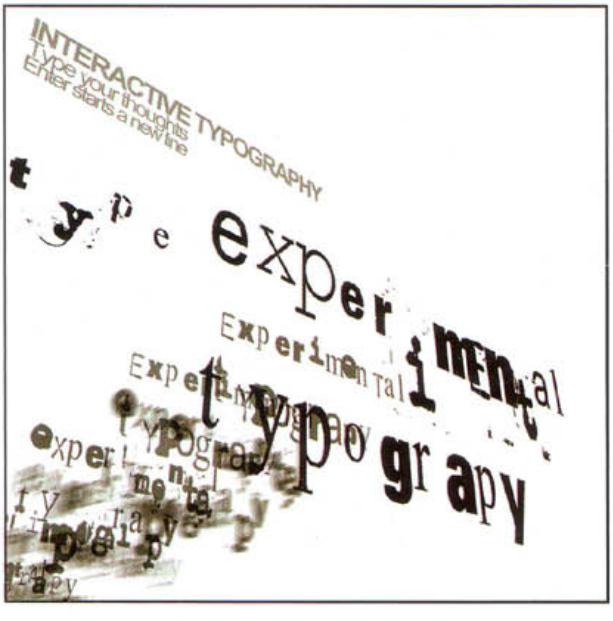
[w] www.ni9e.com

Tell us what topics you'd like to see covered in-depth in *Computer Arts Projects*. Contact us on: **[e] rob.carney@futurenet.co.uk**

We're always on the lookout for new artists as well as established creatives – so if you're interested in contributing to the magazine, please send some examples of your work to **[e] johann.chan@futurenet.co.uk**

Create imagery using digital type

Computer programming and human interaction offer entirely new possibilities for design and communication with type. The creators of ni9e.com, Evan Roth and Max Asare, offer an in-depth look at their personal experiments in digital type...



With backgrounds in architecture, ni9e.com began as a space on the web where we could experiment with ideas free from budgets and committees, enabling us to explore notions of randomness and fluidity. Turning away from clients and embracing experiments such as the ones described on the following pages enabled us to create work that lies between the worlds of art and design. With no previous training in computer programming or typography, *Flash* offered an inviting platform to experiment in both. Typographic Illustration and Typoactive are presented here as two such studies in type, code and interaction.

Typographic Illustration (http://ni9e.com/typo_illus.html) is a drawing technique that uses text to create imagery. Drawing on data gathered from expressive and imperfect strokes of the hand, images are revealed over time. Image, text, and music are used together to communicate content. The underlying goal of this piece (main image, pictured here) was to illustrate lyricists visually using their own words. This project is deconstructed to demonstrate how relatively simple code is used to create tools for building compelling imagery.

Typoactive (http://ni9e.com/typoactive.html) is an interactive animated type tool. This project (pictured left) represents the user's keystrokes as an impermanent and active series of random type styles, scales, and placements. Design characteristics of specific typefaces blur into one another as relationships between very different character forms are questioned. The Typoactive project is explained from start to finish in Part 3 of this tutorial.

Expertise provided by Evan Roth and Max Asare of **[w]** www.ni9e.com.

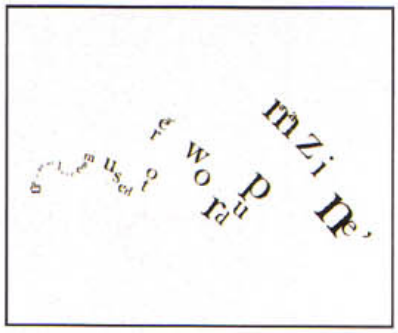
All the components needed to complete this tutorial can be found on this month's cover CD in the Software\Tutorials\Tutorial files\Tutorial files\All\Illustration folder.

Part 1: Drawing a typographic illustration

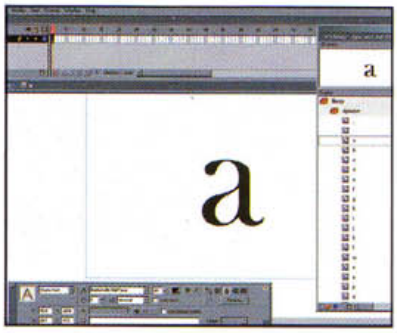
Begin your illustration with the development of a drawing tool, then trace over an image...

Tool creation

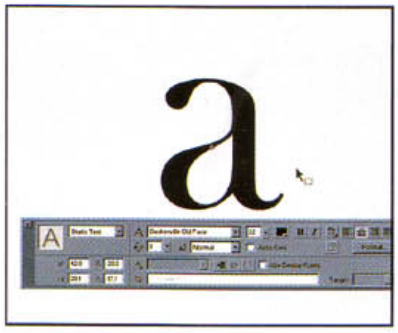
This illustration technique is not possible without first building this simple drawing file. New-media artists such as John Maeda, Amit Pitaru and Joshua Davis are personally influential in stressing the importance of artists designing their own tools. Designing and building new and interesting tools will lead to the creation of more innovative and unique projects.



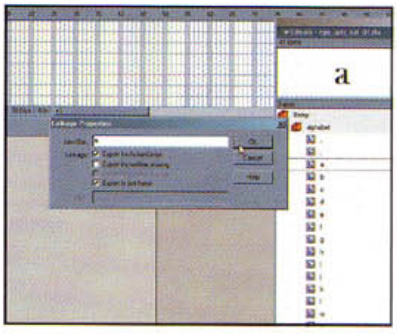
1 The first step in producing this style of text-based illustration is to create a custom drawing tool. This tool places the characters on screen with natural drawing strokes rather than individually positioning each letter by hand.



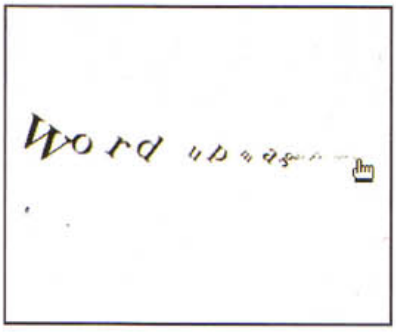
2 Open a new *Flash* file and create a movie clip for each letter in the alphabet as well as any punctuation and/or numbers desired. Centre each character visually within the different clips and set them in the same typeface and font size.



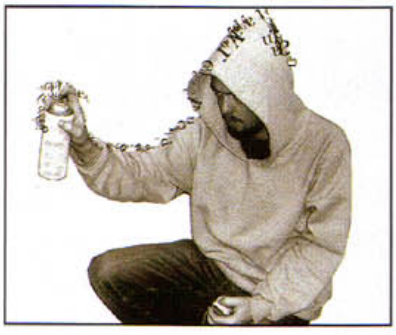
3 The choice of typeface has a considerable impact on the final image. The fluid curves of serif-based fonts often add to the freehand sketched look of the drawing, but feel free to experiment. Note: the typeface is changeable after the image is drawn, so this initial design decision is not binding.



4 Set the linkage on all of the clips to their own letter. To do this, right-click the movie clip (Ctrl-click on the Mac) in the library and go to Linkage. Make sure you check 'Export for ActionScript' and name the clip in the Identifier box.



5 For specific code examples see the typo_illus_draw.fla included on the cover CD. The basic premise, however, is that when the mouse is pressed, a movie clip is created at the mouse position and scaled, based on the speed of the user's movement. This data is saved in the .swf for later use.



6 If illustration isn't your strong point, bring in an image as an underlay to use for tracing (there's no shame in this). Create a greyscale image high in contrast to make areas of light and darkness easy to evaluate.

7 Bring the underlay image into a movie clip and set its visibility equal to a true/false variable controlled from the keyboard. This enables the user to quickly turn the image on and off while tracing, giving immediate feedback as to what the illustration will look like after the underlay is removed.

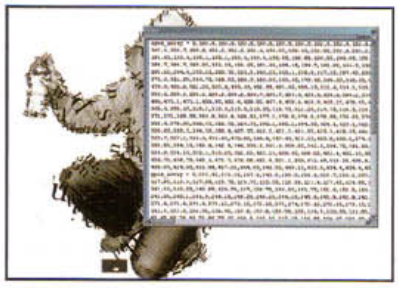
8 Take advantage of the built-in functionality of the Flash Player by using the right-click Zoom-In feature to magnify areas of detail. For regions such as the eyes and mouth, you might need to zoom in two or three times to reach the appropriate level of detail.

9 Continue to trace over the entire image. If you intend to play the illustration back as an animation in *Flash*, be conscious of the order in which the image is drawn, because this is the same way it will animate. Also, the more characters you draw, the more memory intensive the final .swf is.

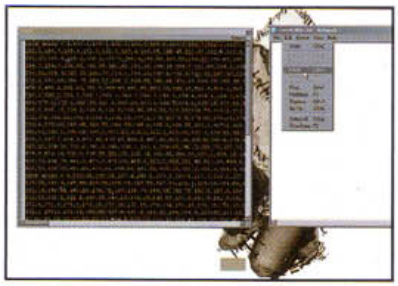
Drawing a typographic illustration continued...

Capturing movement

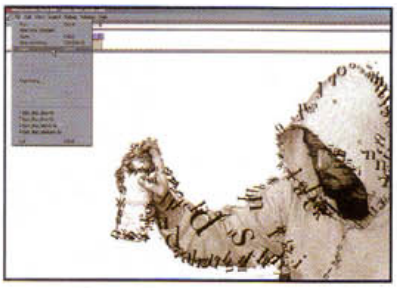
Scaling the size of the characters based on the speed of the mouse movement is one of the most important elements of the drawing tool. This feature expresses the motions of the user visually. The data is recorded from human gestures rather than algorithmic equations, which gives the end product an imperfect yet expressive quality.



10 When the tracing of the underlaying image is complete, click a button in the drawing file to pop up a trace window containing the x, y, and scale information for each character on the screen. To view the trace window, the drawing must be done within a 'test movie' of the *Flash* authoring environment.



11 There are a lot of numbers in the trace window, but don't be intimidated. Select all the text and copy and paste it into a blank text file. You should save a raw set of the drawing data for safe keeping.



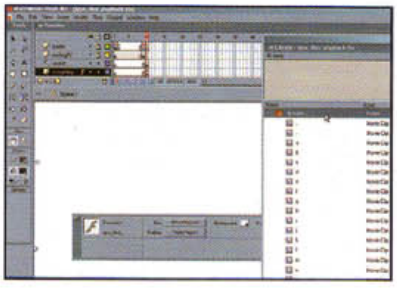
12 Once the data is saved from the drawing file, this part of the process is complete. When the drawing .fla file is closed, the image is temporarily lost, but have confidence that it resides somewhere in that long list of numbers saved in the text file.

Part 2: Using code to create a typographic image

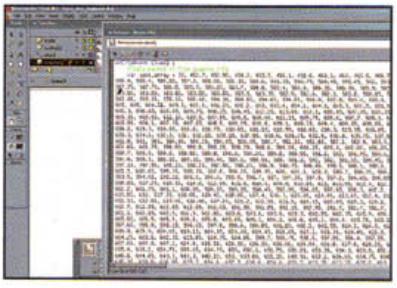
Another way of making an image with type is to re-create it from code...

CPU vs. human speed

It's entirely possible to re-draw the illustration in real time as it was drawn originally. The examples at ni9e.com, however, redraw the characters in a linear fashion: animating as fast as the computer can draw clips to the screen. This linear and rigid timing plays against the fluid and irregular gestures of the hand, as captured in the scaling of the letters.

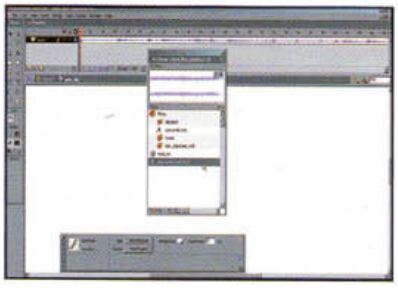


1 Create a new *Flash* file for re-creating the drawn data. Include the same exact alphabet in the library, and retain the same pixel dimensions for the .swf width and height. Make sure to embed any fonts that are used within this file.



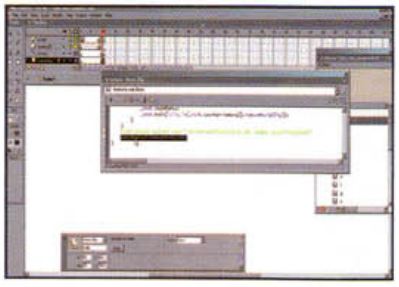
2 Copy and paste the data saved in the text file into an onClipEvent (load) command of the new playback file. Make a series of arrays that store the x position, y position and scale information as created from the drawing file.

3 After the data arrays are in place, create a string array containing ' ' text to be spelled out as the characters are re-drawn. In this example, this array has been set equal to text from this tutorial, so the image is made from the words that you're reading here.



4 Audio is an important consideration. The examples on the ni9e.com site use the lyrics from the audio as the text that draws the image (that is, drawing lyricists in their own words). Import the audio from a .mp3 file and set it to streaming.

5 On every frame of the animation, advance one letter in the text array and create the corresponding movie clip from the library. Set the clip's position and scale equal to the data in the arrays created from the drawing file.

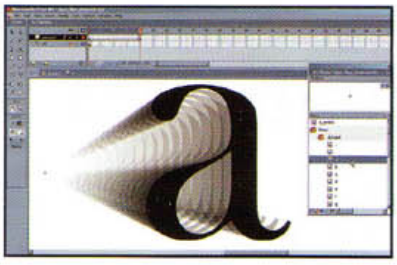


6 Experiment with the speed of the animation and the limits of the CPU processor. Creating two to five letters every frame can greatly increase the visual speed of the drawing. Using the built-in setInterval() function gives greater control over playback speed.

Using code to create a typographic image continued...

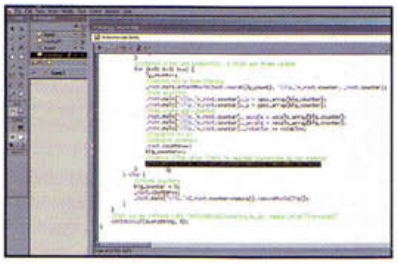
Image and text

In any text-based illustration technique, two methods are available to communicate an idea: 1) characters that read as an image, and 2) characters that read as text. The main focus of this project is in their reading as an image; however, eliminating rotation of the characters and limiting the length of the text to one word or phrase can help in the legibility of the letters as words.

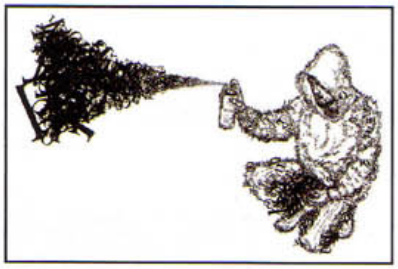


7 Experiment with simple changes to the letter clips in the library. Small modifications, such as basic tween animations and changes in placement, scale, and colour, can have dramatic effects on the final animation. The same goes for substituting images, symbols, and icons for the characters.

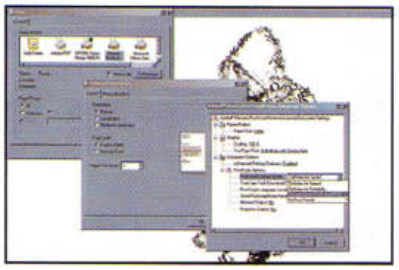
8 Changes in rotation of the character clips will also impact the resulting illustration. You may find that rotating each character by one degree from its neighbours often adds more visual movement. Leaving all of the letters in their upright positions tends to vertically elongate the final image.



9 To keep the animation in a looping cycle, delete earlier clips before looping and creating new ones. We often animate the illustration to 90 per cent completion before deleting characters, creating a long string of text that grows on one end while deleting on the other.



10 The .swf is now complete. Besides audio (which will stream) the files size should be somewhere between 50-120kb depending on the font. Putting the audio in its own .swf for web publication might help download and playback speeds.



11 To create a high-resolution typographic illustration suitable for printing, print the .swf to an .eps file. Make sure you have a generic PostScript printer set up and select 'Print to file'. In the Advanced tab of the Printing Preferences, select EPS from the PostScript Output Option box.

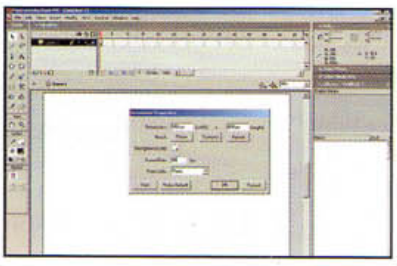
12 Transferring typographic illustration animations to video is also possible using capture software such as *SnagIt*. Set the frame rate in the playback .swf file and the frame rate in the capture software to the same number. Open the .swf and hit record in *SnagIt*. This creates a .mov file of the animation.

Part 3: Typoactive

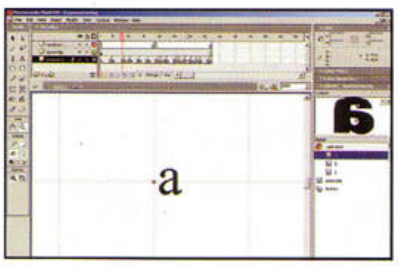
Try out this typographic experiment in interactivity and randomness...

Typeface selection

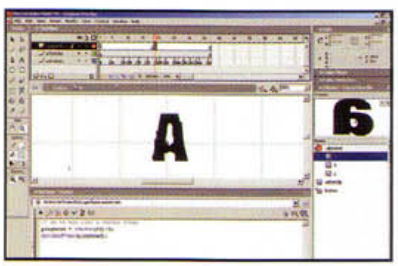
One theme explored was the blurring of distinctions between various seemingly incongruous typefaces. Clean and classic fonts are contrasted with grungy, slightly illegible type, adding to the overall idea of randomness and tension.



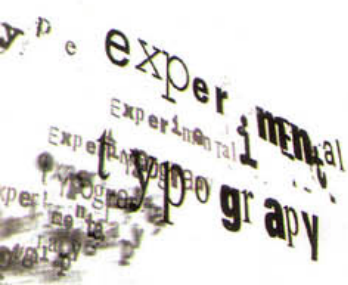
1 This section is again made from just a still image. The main technique for creating movement in this scene is produced using *Photoshop*. We take a still of a video camera and open it in *Photoshop*. We then select a very thin slice of the image, and copy and paste this into a new layer. Using the Transform tool, we scale this thin slice, which stretches the image from that particular point.



2 Open a new *Flash* file and set the frame speed to 50fps. Smoothness and speed of the animation are of the essence, so a frame rate of 24 or greater is necessary to achieve the proper fluidity desired.



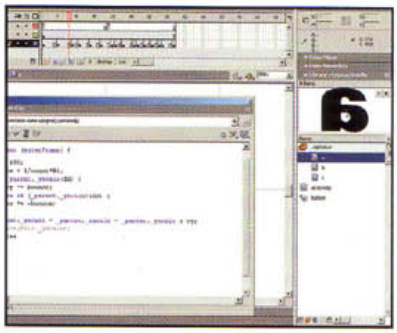
3 Create an animation for the letter 'a'. To achieve a random, indeterminate look to the movie clip, the duration, scale, and location of each typeface should differ from frame to frame. Due to the high frame rate, tweened animation is not necessary.



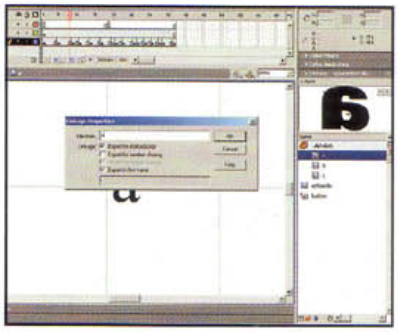
Typoactive continued...

Dynamic scripting

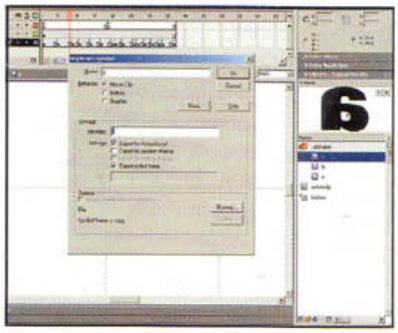
The key aspect for finding the right feel of motion is the script that controls the scale of the movie clip. This element was crucial to the overall piece and is covered in more depth in the .fla included with the CD. Influential dynamic scripting in typographic projects include work by Yugo Nakamura and Erik Natzke, among others.



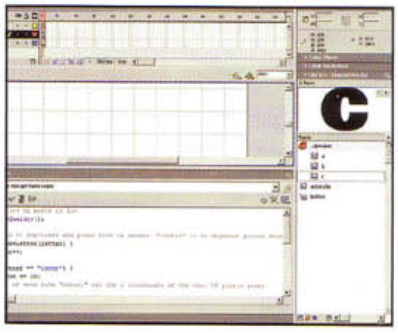
4 For an added degree of variation, add several frame actions throughout the movie clip that tell the playhead to move to a random frame. This is necessary because, while the keyframes for our various letter movie clips are identical in the .fla, the intent is that they all appear distinct from each other on playback in the .swf.



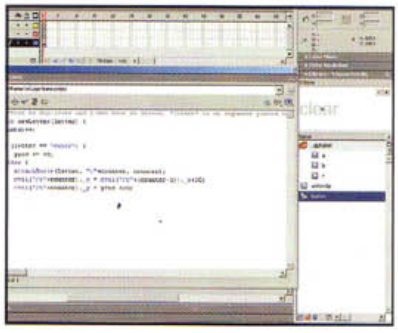
5 Now add an action clip to dynamically control the scale of the letter. The scale of the movie clip will constantly oscillate between values larger than or smaller than 150 per cent. Also, an action is needed to remove the movie clips when the user hits the clear button.



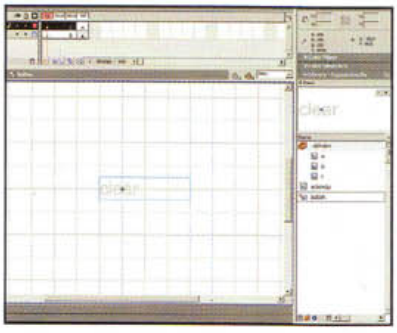
6 Make a movie clip for each letter in the alphabet, as well as any punctuation and/or numbers desired. The easiest way to do this is to duplicate the library instance, leaving the keyframes the same, and changing the letter in each text box.



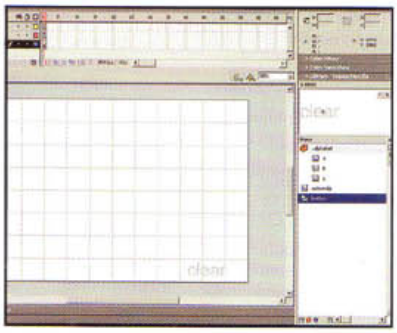
7 At this point, the main timeline scripts are added to make the final product. In the first frame, turn the quality of the .swf to low to ensure the fastest possible playback. The lag on the processor will grow as the number of movie clips on the screen increases.



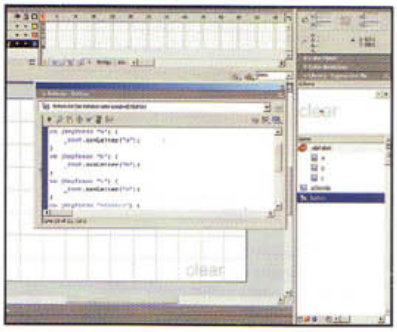
8 Declare a new function to add text to the screen. The function receives the last letter pressed from a button, attaches the corresponding letter movie clip, and places the letter on screen.



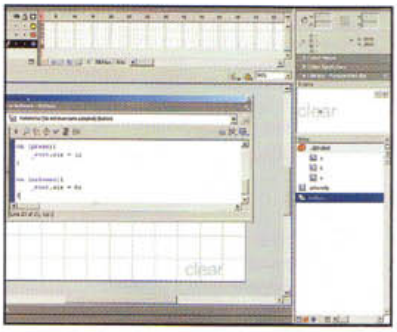
9 A method is needed to detect the user's keystrokes. To do this, create a button that will also double as the clear button for the user. For specific code examples, see typoactive.fla included on the cover CD.



10 The next step is to add the button to the main scene. This object controls the bulk of the movie and is the only object in our main timeline. This example places the button inconspicuously in the lower-right corner of the movie.



11 On the button, add actions to call the new function when a key press is detected. This action will pass the letter pressed to the function and add the appropriate letter movie clip to the screen. This is required for each letter.



12 As a final step, add an action to the button to clear all of the movie clips from the screen. At this point, the .swf is complete and ready for publication to the web. ca-p