

Typesetting with Flash MX

M.D. Rowland

Abstract

A common misnomer states that there are only two fonts on the web: Times New Roman and Arial. While it is true that web designers often select their typefaces from a set of de facto standards, there are many sophisticated ways to practice the art of typography on web projects. Web designers must not ignore type. Even though there are significantly fewer years of historical examples to draw on, digital type should be approached with the same careful attention to detail as traditional type. This paper will strive to answer the question: how can web designers using Macromedia Flash produce legible, credible, and beautiful type in their projects?

Introduction

Before the invention of the personal computer, type was set by experts. A typesetter was a tradesman who's only concern was the proper treatment of the letters and numbers on the page. For better or worse, the trade of typesetting has all but disappeared, and most type is now set by amateurs on personal computers. This paradigm shift has fostered some of the most atrocious typography disasters in the history of the craft. Why? For a simple reason: in the same way that an architect would be ill-equipped to install an intricate Spanish tile floor, most designers have virtually no capability to properly set type. To complicate the matter even further, type is no longer confined to the printed page; it now travels at the speed of the ether from servers all over the world to millions upon millions of desktop computers in every home and school. Type must now exist in two worlds, one of ink and one of pixels. With shifting paradigms, expanding media demands, and ever-increasing technological innovations and complications, how can any one designer be expected to master the skill of typesetting in a digital world?

While this paper will touch on many topics that apply to traditional typography as well as generic web design, the main focus will be on increasing the level of typographic awareness for those web designers using Macromedia Flash as a component of their skill set.

There are many convincing arguments to be made both for and against using Flash to develop content for the world wide web. Regardless of which side you happen to agree with, one thing is for certain; there is a general lack of sensitivity to the treatment of type in Flash-based projects. Be it the fault of inexperienced designers, or a lack of typesetting control and integration within the program, the result is an obvious typography disaster. Since users have no control over the interface and functionality of the Macromedia Flash application, focus will be on increasing the user's skills when interacting with the program's type features.

Theoretical Considerations

Attributes of Type

A careful distinction must be made before this topic is approached further. The word "type" carries some connotative and denotative ambiguity. In order to ensure clarity, the phrase "attributes of type" will be used to refer to the typographic elements of a project and the typographic decisions made by the designer, i.e.: font choices, size of copy, weight of subheadings, etc. and will not refer to specific properties of typefaces such as x-height, slope, ornateness, etc. With that said, when setting type, designers must consider, and ultimately balance, several attributes:

Aesthetic Appeal

Type should be pretty. If it isn't pretty, it's not going to inspire anyone to want to read it. Whenever possible, type should be inviting, exciting, and beautiful.

Legibility

Type must be legible. After you've convinced someone to read it, you have to deliver it in a way that isn't painful to endure over the long haul. Aesthetic appeal will capture readers, but legibility will keep them.

Usability

Usability refers to a user's ability to interact with your text and incorporate it into their computing environment. This is a unique concern of electronic text. Some factors that affect usability are selectability, browser-side control of type size rendering, <alt> tags for rasterized type, and clearly defined hyperlinks within text fields.

Content

Robert Bringhurst wrote "The typographer's one essential task is to interpret and communicate the text. Its tone, its tempo, its logical structure, its physical size, all determine the possibilities of its typographic form" (1999). Put simply, the typographic decisions you make must portray the text in an authentic way, making sure to give the reader no false pretenses.

These four attributes constantly collide with each other and demand consideration from the designer. Too much bias in either direction may produce type that is functional but not interesting, or perhaps beautiful but unreadable—or even worse—totally misrepresentative of its text. A balance must be achieved between form and function, experimentation and convention, pursuit of beauty and pursuit of truth.

Technical Considerations

Jaggies and Fuzzies

Flash makes it easy to display a wide variety of typefaces on anyone's computer without installing fonts or consuming excessive bandwidth. Flash can embed a font within a shockwave movie file and display that font's characters on a client machine regardless of whether the client has the font installed or not. This is a nice feature for designers who are looking to break out of the "Times New Roman/Arial" rut.

Unfortunately, embedding a font within a shockwave movie mandates that the font be rendered with an anti-alias. This raises the question, "To Embed, or Not to Embed?" Which is, in this scenario, the same as asking "To Anti-Alias or Not to Anti-Alias?"

The anti-alias dilemma is a hot topic among professionals who interact with digital type such as web & multimedia designers, and software developers. To properly display a font on-screen, the font's vectors must either be smoothed with an anti-alias or remain jagged due to the limitations of the screen's resolution. Which is better, the jagged edges, or the smooth halo around each letter?

Let's consider the attributes of type and what effect anti-alias has on them:

Anti-Alias: Effects on Aesthetics

While ultimately a matter of personal preference, most users and experts would agree that a well-rendered anti-alias is generally more visually appealing than a jagged vector outline. In any case, here are conflicting industry opinions:

Nicholas Negroponte (1994) has long called for all text to be anti-aliased. In the article "Aliasing: the blind spot of the computer industry," he says:

What puzzles me the most is that we seem to have educated an entire generation of computer scientists who don't fully understand this simple phenomenon, and we seem to have trained the public to take it for granted. Perhaps it's time to make [orthochromatic] graphics a violation of Occupational Safety and Hazards Administration minimum standards for display quality. Or, perhaps the Environmental Protection Agency can declare this condition to be visual pollution. The point is that it must stop. (1994).

The November 13, 2000, issue of the Independent takes the opposite extreme about the anti-aliased rendering of type in Mac OS X:

It looked like someone had smudged the screen with margarine. Other people can bear [anti-aliasing], but I hate it; only by choosing a tiny size of a non-aliased font could I begin to write without feeling uncomfortable (Independent, 2000).

The debate rages on, but for now let's simplify the issue. The anti-alias rendering technology used by Flash is ugly. anti-aliased letters are supposed to have a subtle 'halo' that smooths the jagged edges of the vector outlines. In Flash, particularly at small and medium sizes, the subtle halo is more like a disturbing aura of blurriness.

At larger sizes, the anti-alias rendering in Flash is quite pleasing. For this reason, it's perfectly acceptable to embed fonts that will be used as headings or display faces.

Anti-Alias: Effects on Legibility

In his recent study *Legibility and Comprehension of Onscreen Type*, Dr. Scott Chandler wrote extensively about the effects of anti-aliasing on legibility of onscreen type. Chandler's research suggests that at smaller sizes, serifed faces are more legible when rendered with an anti-alias, while sans-serif faces are slightly more legible when rendered without an anti-alias. The research also indicates that at smaller sizes, a non-anti-aliased sans-serif face is more legible on screen than either an anti-aliased sans serif or

serif face (Chandler, 2002). While this statement makes sweeping generalizations about a complicated topic with many variables, it is consistent with Dr. Chandler's findings.

Dr. Chandler's research methods were exhaustively accurate and precise; however, he did not use the anti-alias rendering technology inside of Flash as a part of his study. It is this author's opinion (which is shared by many industry professionals) that the anti-alias rendering technology used by Flash is vastly inferior to the anti-alias rendering engines within Adobe Photoshop, Adobe Acrobat, and other pieces of software. After reviewing samples of anti-alias rendering of Times New Roman generated by Flash MX, Chandler had the following post hoc comments to add concerning the anti-alias rendering of type within Flash:

Without doing a comprehensive analysis such as that conducted for my dissertation, several attributes of Flash anti-aliasing are obvious. The anti-alias rendering appears inferior to other rendering techniques. The rendering itself seems awkward and unprofessional. Most rendering technologies work hard to consistently align type to a grid, that isn't evident from this font sample. Although this can be the result of badly drawn glyphs, the misalignment to the baseline and variance of x-height across glyphs is an indication that the rendering is poor. A font like TrueType Times New Roman is almost certainly well hinted, making this sort of rendering inexcusable. (Chandler, 2003)

Anti-Alias: Effects on Content

The decision to render type with or without an anti-alias is largely independent of the task of picking a font that accurately reflects the content. Decisions made about this attribute may be largely overruled by decisions made in other steps, for example; to ensure maximum legibility and comprehension, you may decide against embedding type. This forces you to use a universal font that will be available on a vast majority of all client machines. These universal fonts may not be an ideal choice for complementing the text, but picking a satisfactory candidate from what's available is still possible.

Pixel-Based Fonts

Because of Flash's anti-alias rendering limitations on smaller type sizes, many type foundries are designing fonts specifically for use within Flash. These pixel-based TrueType fonts are intended to be used only at one size and dramatically reduce the negative aspects of Flash anti-alias rendering at smaller type sizes when properly used. Visit www.miniml.com for examples.

Summary

Critics of Flash have been quick to point out the shortcomings of the appearance, and lack of legibility of small and medium sized anti-aliased type rendered by Flash. For this reason, when setting smaller type sizes, it is almost always better to use a standard web font as a device font within Flash and allow the client machine to render the type. When setting static text at small sizes, you should select a universal font such as Arial, Times New Roman, or Verdana and check the *use device fonts* box under the text properties in the property inspector. If you're setting dynamic text at smaller sizes, you should once again select a universal font and then click on *Character* and select *No Characters*. Pixel-based fonts

are an effective alternative to using device fonts for smaller type sizes.

Leading, Kerning, Tracking & Other Type Controls

Macromedia has made a decision within Flash to abandon much of the traditional typographic terminology that has been used by the industry since the days of Gutenberg. In addition, the type control interfaces within Flash are dissimilar to any other graphics application.

First, there are no precise kerning controls. This is a huge omission and severely cripples the amount of typographic control provided to designers by Flash. There is an “auto kerning” feature, but it appears to have little or no effect on most passages of type.

Another typographic shortcoming is the lack of paragraph controls. Flash includes a rudimentary leading control called “line spacing,” and some indentation and margin controls under *Format* inside text properties, but is weak at controlling hyphenation, justification, and composition. The controls are functional but far from perfect. The Flash application could be greatly improved by the integration of advanced type controls and formatting features.

Lack of Professional Type Controls:

Effects on Attributes of Type

It's easy to see how the lack of professional type controls has a negative effect on the attributes of type set with Flash. More intelligent paragraph composition would ultimately translate into prettier passages of type. The lack of precision kerning controls severely undermines legibility.

Selectable Text

Another important technical consideration is the inclusion of selectable text. Selectable text may seem like a trivial issue, but it's actually quite important. Users visiting your site are mostly concerned with content which makes the ability to select and copy/paste that content into other applications an important usability feature. It's easy enough to turn on selectable text and it has no significant technical drawbacks, so always make an effort to make your body copy and important text selectable.

Independently Formatted Printable Objects

An often overlooked but high-powered feature of Flash MX is the ability to customize printable content independently of your onscreen content. Printing web pages is an issue that's plagued designers from the very beginning of the world wide web—so much that many have all but given up on the idea of creating web pages that work both on-screen and in print.

The problem is one of formatting: computer screens are wide. Paper is tall. Web sites are designed for wide computer screens and have to be fitted on to tall pieces of paper with conflicting aspect ratios.

Flash has an innovative feature which allows designers to include independently formatted content specifically for printing purposes. With a small amount of effort, a designer can repurpose an entire web site's content to a portrait aspect ratio for the printed page. This allows designers final control over how their web pages will

print, taking away print formatting decisions from the end-user's web browser and eliminating the conflicting aspect ratio problem.

Independently Formatted Printable Objects:

Effects on Attributes of Type

Independently formatted printable objects are a key usability concern. Different users may choose to interact with your text in different ways. Giving them both high-quality on-screen and print options makes your text more usable. Less apparent are the effects on aesthetics and legibility. Not having to concern oneself with dual-formatting a single web page that works on screen and in print leads to a wider array of design options. Separating onscreen and print versions of content could potentially have positive impacts on your text's aesthetic appeal and legibility for both versions.

Conclusion

Typesetting under the best of circumstances is tough. Factoring in confusing font rendering technologies, lack of professional type controls, and quirky usability features makes it all the more difficult. Daunting as it may be, properly setting type inside of your Flash projects is imperative to ensure effective communication of your ideas.

References

- Brighurst, R. (1999), *The elements of typographic style* (2nd ed.), Vancouver, B.C., Canada: Hartley and Marks.
- Chandler, Scott B. (2002). *Legibility and Comprehension of Onscreen Type*. Unpublished Dissertation.
- Chandler, Scott B. (2003). *Flash MX Anti-Alias Rendering, e-mail response*. Retrieved February 3, 2003.
- Independent. (2000). *Return of the Mac, new and improved*. Retrieved November 19, 2000, from <http://www.independent.co.uk/news/Digital/Features/2000-11/mac131100.shtml>
- Negroponete, N. (1994). *Aliasing: The blind spot of the computer industry*. Retrieved May 20, from http://www.wired.com/wired/archive/2.01/negroponete_pr.html