

# The Phillip Newsletter 2007

phillipkerman.com LLC's report to Clients, Colleagues, and Prospects

www.phillipkerman.com/newsletter

news@phillipkerman.com

Volume 11 Issue 1

## Now with no trans-fat, same 9 pt font, and the October Surprise revealed on page 3

**Projects:** *You'd think programming a Sudoku game would make me a good player, but you'd be wrong.*\_\_\_\_\_

It feels as though I've only completed two projects this year—but those two were doozies. I programmed a couple of two-player games, produced by **Adveractive** and **Carbonated Games**, that you can play with your IM buddies in **MSN Messenger**: **Sudoku Too** and a forthcoming **jigsaw** game. You'd think I'd have some sort of rock-star status (or at least reality-show-contestant status) because these games are available right inside an instant messaging network with 180 million regular users. At least I'm popular among the little old ladies playing jigsaw anyway. I wonder now whether having my name listed in the credits may attract deranged Sudoku players upset when they lose a game. (Seriously, the fact everyone involved got credit is a really great thing that should be more commonplace.) Something like 10-20 million users regularly play these games, of which there are currently only a couple dozen. Despite this global reach, it's amazing how the requirement to use Windows and the MSN Messenger client (not something like Trillian) excludes so many of my Flash buddies. IM me at **sudoku9000@hotmail.com** and I'll give you a tour of the games.

These games posed some serious technical challenges (besides making games that are fun to play). The development kit **Microsoft** makes available provides a simple interface to send arbitrary XML strings to the other client. Alas, there are no callbacks so keeping the two players synchronized is tricky. In addition, the message frequency can't exceed 120 per minute and each message can't exceed 1500 bytes. I'm particularly proud that I managed to allow players to save their jigsaw games (with up to 200 dynamically drawn pieces on a 1200x1200 field) within the 2000 byte per-player limit for data storage. All these restrictions make sense in order to keep their network running smoothly but also make programming challenging.

I did a few micro sites for **hotpepper.com** including **amaa.com/portfolio/** which turned into a bit of a showdown between Flash and HTML as two nearly identical versions are available. While the Flash one is cooler and arguably easier or harder to use, I am preloading all the thumbnails so the apparent delays are much shorter in the Flash version.

We also built a really nice usable timeline for **HP** (**tinyurl.com/ra88q**). I think it really shows that we spent the right amount of time getting it to "feel" right. Amusingly, it's as if someone is trying to goad me with the headline above the timeline reading "a more extensive text timeline is also available".

### Editorial: Skills for the future

I get tired of hearing how a math and science education is the key to success in the future. Of course those skills are important—especially if that's your personal interest or if you're considering the diversity of skills in your country's workforce. However, I think people sometimes place such a great value on technology that they overlook the foundation on which so much of our success relies—namely, communication skills. Language skills, writing, and verbal communication are arguably *more* important than math and science. At a minimum, I'd say if you're a better communicator you'll have more success at whatever you do. For example, the most creative scientists will have less success if they can't compose a sentence to communicate their ideas. I'd even add visual communication to the list of what's important for today and the future.

If you were under a rock in 1977 (or perhaps more likely not born yet) and out of touch in the 90s you may not be aware Portland has an **NBA** team. And now they have a nice website that I got to work on through **Zaaz.com**. Check out the part I programmed: **nba.com/blazers/team**...complete with alpha channel video, RSS feeds, and BitmapData for the cross fades.

Speaking of towns that used to have great basketball teams, I went down to Los Angeles to teach a two-day workshop: "*Unleashing the ActionScripter Within*" through **richmediainstitute.com**. It was pretty fun. Check out the videos of me teaching!

I converted a "session planner" that I had originally built for online subscription use at **soccerspecific.com** to run offline as a **Zinc** projector. I also added some customized printing options. The core feature (to let soccer coaches layout practice routines) is pretty simple—basically, just a bunch of drag and drop. However, with advanced drawing tools and little touches such as an undo-stack, it happens to be implemented really well.

**Jeff Faulkner** hired me to program a build-to-own sunglass tool he created for Oakley. I built a template used for each of the sunglass builders linked from: **oakley.com/custom/eyewear/**

And now you've reached the place in every newsletter where I make sure to list every project I worked on. Be certain, however, that if the project appears in the following short list it's not as though the project was unimportant. They sure seemed important while I was working these projects!

**thinkmh.com** had me build this crash-test video player for **Thomas Built Buses**: **tinyurl.com/zohg4**

**overlandagency.com** had me help build a small framework.

**smashingideas.com** had me build a few custom presentation tools for their clients... neat stuff, like letting the presenter pause to make permanent edits to the content.

**creativdepartment.com** let me build a cool prototype using a (soon to be cliché) parallax effect. It was fun because we used Flash 8 filters to add realism.

I think a lot of people will read this and simply say "no duh". Perhaps I'm overreacting and everyone already knows what I'm saying. It just seems like I hear it all the time. Of course I don't want my saying communication skills are important to imply that I think math and science are not important. (In fact when I hear fewer students are going into math, science, and—gasp—programming, it does concern me.) Ultimately it's most important to study what interests you. Find something you like to do. It's easy to find someone who values what you have to offer (and is willing to pay for it) if it's something you really care about too.

**Editorial update:** in 2003 *The Phillip Newsletter* predicted that blogs would turn out to be a fad. The jury is still out on that one while *The Phillip Newsletter* continues to be printed.

### A better hit test

Sure Flash 9 has some crazy collision detection options... but for those of us back on earth, Flash 8's `BitmapData.hitTest()` is pretty awesome. Recall how the old `MovieClip.hitTest()` can tell you if any one *point* intersects a particular clip but it can't help you determine whether two odd shaped clips are touching. `BitmapData.hitTest()` is tricky to use, but the basic approach involves making two 32-bit bitmaps (say, based on your two clips using `BitmapData.draw()`). (See the online version of this newsletter for a link to a code sample.)

### Using setTimeout within classes

I suppose you could probably edit the intrinsic classes to get around the fact that `setTimeout()` generates a compile-time error in AS2. Here's another way:

```
global["setTimeout"] ( function(){  
    //code  
}, delay);
```

### Reviews ...*Too many new tools to review them specifically*

---

#### Edward R. Tufte (tufte.com)

The granddaddy of information design recently published his fourth book (in a promised 5-book set) *Beautiful Evidence*, so I figured I'd go to his 1-day course as a refresher from 9 years ago when I last attended. The new book and the course are great, as expected. For a Tufte fan like me both included some repeated information—but it was great to hear, for the first time in person, his description of “sparklines” as well as his PowerPoint diatribe. I'll admit that listening or reading his overtly opinionated work is entertaining but can make me feel like challenging him—if for no other reason than to pick a fight. But when one of the other students in his course began to question his contention that a single 11"x17" sheet of paper can hold the information equivalent to 250 PowerPoint slides (a fact anyone with a calculator could deduce) I decided it's obviously best not to mess with him.

Tufte is a mindset and so much of his work is really valuable. (I'd go as far to say his influence has significantly helped my career.) I have often felt that he was a little out of his element when discussing information design for the computer screen—perhaps because of his disdain for anything less than photographic quality resolution. But the thing that struck me in his course was that he effectively applied all his teachings to today's technologies while at the same time not pandering to whatever happens to be hot right now. (Speaking of pandering to what's hot right now... see the column to the right.)

#### Flex 2

I don't feel quite authorized to do a thorough review of Flex 2 yet—I'm lucky to get Flash player 8 projects. But, man, I can't wait to get into the Flex Builder 2 workspace. (I'll miss you PrimalScript, sorry.) Even if it's just to author class files—I'm ready. It's well thought out and feels “real”. The fact the command line compiler and SDK is entirely free is cool and if that's what it takes to make the Flash player go farther—great. But for me, the work environment of Flex Builder 2 is what's attractive. Sure, it's fun to dream about the new capabilities of Flash player 9—but you can access that from the Flash IDE (and today even). I am also looking forward to the new component architecture in Flash but ever since Smart Clips I think the implementations have been a little half-baked. I have high hopes for the *next* version because of the super-human programmers involved including [gskinner.com](http://gskinner.com) and [metaliq.com](http://metaliq.com). (Then again, the folks involved in Flash have never exactly been hacks either.)

### Pre rendering filters

You can create some nice effects by using a filter and repeatedly modifying its properties (say, its `blurx`) as you keep applying it to a clip. That usually works fine, but an option that can improve the performance is to pre-render the steps of your filter animation as bitmaps, store them in an array, and then display those bitmaps very quickly. If you have to render the bitmaps when the user clicks, you'll notice an initial delay (as you render all the bitmaps). However, the animation will play faster so it might be worth that cost. Sure it can take some RAM... but just remember: bitmaps display fast; vectors download fast.

### Use onResize

More of a kick in the pants than a “tip”. And sure, it's tons more work. But when appropriate you can make your application way more useful and professional by defining layout rules to accommodate any screen size. Sample code is available online.

### AJAX

What's there to review? I suppose I could review one of the countless available frameworks. And there are tons of great sites that, because they happen to implement one or more of the initials in AJAX, mark the beginning of *another* new era of something. Seriously, I'm a tad defensive because often I see examples that cite AJAX to spite Flash... “see what I built... and I didn't use Flash”. Well, if you have to tell a story (like how hard something was or that you did it without resorting to Flash—as if that's cheating) then it doesn't matter. What everyone judges projects on is the end result and its effectiveness. The truth, however, is there are some killer web apps that—by the feat of some AJAX geniuses (and good design)—are pretty darn useful! One thing that's great about AJAX (besides all the masochists who get the stuff working in most browsers) is that there's a concerted effort to standardize and at least share lessons learned. See the Yahoo! UI Library (YUI) for an example. (Granted, much of that library is nothing more than putting names on what we've done for years... but that's exactly what a design pattern should do.)

Back to Flash getting AJAX'd (also known as the Ajaxination of Flash)... I do think “we” (or Adobe) need to be careful not to let AJAX-heads banish Flash to the standing of simply a media format. I'm not sure how to best prevent this (the Flex-AJAX Bridge is a good start). I do know it will take more than simply all of us knowing what Flash can do... and understanding the relative strengths of various solutions. (Since when does the “better” technology necessarily win anyway?) One suggestion is to avoid AJAX vs. Flash showdowns. Sometimes Flash would win—say, when comparing Yahoo! maps to Google's... but Flash would certainly lose in some cases such as [jamjar.adobe.com](http://jamjar.adobe.com) or [broadmoor.com](http://broadmoor.com) vs. almost anything comparable. But in these cases you'd be crediting or blaming the technology based on the specific implementation. Maybe AJAX's popularity is just payback for when people used Flash for the sake of using Flash. It was wrong then and it's wrong with AJAX too. I realize this is more of an editorial than a review. The worst part is that I'm treating AJAX as a specific technology when, in fact, it's simply a technique.