

The Phillip Newsletter 2008

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

www.phillipkerman.com/newsletter

news@phillipkerman.com

Volume 12 Issue 1

The Secret is Out: The Phillip Newsletter Contains Hidden Puzzles!

Projects: *Fewer projects than usual, but bigger projects.*

Every time I start this newsletter I think "oh no, I didn't do all that many projects last year" but then after I review the year I realize the list isn't so bad. (Try it for yourself: list all your accomplishments over the last 12 months—it's a great exercise.)

If you're tired of hearing about the two-player Sudoku and Jigsaw games I've been working on (including four variations of **Jigsaw Too** on MSN), just imagine having to work on it for multiple years! I'm currently busy adding a really killer feature to the jigsaw game. (Can you guess what it is?). I am a bit surprised that the work remains challenging, interesting, and fun—but here's where my advice comes in: if you're going to work on something for a long time, make sure it's something interesting!

I'm about to start a **third game title** (and maybe even a fourth one) for MSN Messenger to be released in early 2008.

Multi-user applications are deceptively challenging to build. In the case of the MSN Messenger games, doing anything beyond turn-based games is particularly difficult because the environment behaves as peer to peer. It was almost a dream coming back to **Flash Media Server** to add some powerful features to an **online classroom** application. I added a messaging tool that lets teachers easily filter an onslaught of student questions plus I built a survey tool for teachers to build pop quizzes on the fly. The good thing about this project was how easy it was to see its value. Compared to an online game which obviously has value, the business model here was much easier to comprehend.

Editorial: *Living for a Living*

Reading delivers more words per minute than listening to someone speak (though I realize it's different). When I prepared my ignite presentation for MAX (see Projects, above) I tried taking 15 years of self-employment experience and condensing it down to a funny 5 minute talk. In the end, I could only safely cover about 7 points. But because I've spent time collecting my thoughts, I figured I could share more here.

The presentation highlighted such revelations as "Dumb & Dumber" which means you only need to know more than your clients—that you don't need to first study everything before making the leap to self-employment. That having a client who appreciates what you offer is critical—don't try to be something you're not just because you think your client wants that. That it's people who hire people, so increasing your opportunities to be face to face with people increases your business. I threw in the geo-offset trick which is a way to get past any typecasting your current prospects may have of you. For example, if you're a production artist and want to be an art director, it's easier to change your environment (move to a new location or just change the people you hang with) and then define your past yourself instead of trying to convince the people who already know you that you've grown.

The two lessons I shared were: 1, you really need to be direct and tell clients the truth even if it's not what they want to hear; and 2, resist getting too busy in order to plan for that next gig and keep from getting burned out too. That all took 5 minutes to deliver in the live version—but probably only took you 1 minute to read. You didn't get to hear my jokes unless you were there. The presentation was supposed to be funny but I was also serious when suggesting to include downtime as a way of being more efficient.

I helped **creativdepartment.com** rebuild the portfolio portion of their website. The clothesline metaphor they designed came out pretty nice.

Because I like being complete here, let me mention I also did one small job each for **mile7.com** and **periscopic.com**

I've written 4 book titles (not this year, but total). If you count revisions it's actually a total of 9 books. The one title that's worth revising with every edition of Flash is **Sams Teach Yourself Flash in 24 Hours**. The CS3 edition took a lot more work than usual. I think the reason is that the user base has necessarily split into either Flash developer or Flash graphic and animation producer.

I didn't do a ton of conference presentations this past year. The most exotic was **Flash on the Beach** in Brighton UK. The biggest was undoubtedly the **National Association of Broadcasters (NAB)**. The most Canadian was **FITC Toronto**. The two with the most similar vibe were Portland based **Webvisions** (webvisionsevent.com) and Ohio University's **Spring BR** (in Athens). In the "attending only" category I went to **Flex360** in San Jose (a really great conference by the way) and **Adobe MAX** (wouldn't miss it). Wait, that's a lie! I got a gig to speak at the **MAX Ignite** event. You create 20 slides that automatically advance every 15 seconds. Basically cram as much information into the smallest space as not to waste the audience's short attention span.

Naturally, I have still more opinions of how to be an effective self-employee. So here are a few more. First, there's a simple negotiation technique I first heard about from **Tom Jackson** (careervictory.com) that you might not think works—but it can: always make the other person say the price first. Sure, I get clients who really want me to bid first but when I know how much a client wants to pay, I also can see if their expectations are reasonable. So, when the client says "how much?" you say "what's the budget?" It doesn't have to be evasive either.

In case you ever feel like you're charging too much, consider two things. First, the number of days you have left to live is limited (see **phillipkerman.com/deathday** to calculate it). And, second, the dreaded "contractor multiplier" can make you think you're making more than you are. It fools you into counting your pay multiple times: chat to a prospect about a gig, get the gig, do the work/track your hours, send a bill, finally get paid. Each stage feels like you made money—but these are the same dollars repeated.

More tips: don't play hide-and-peek with your clients. With Caller ID and email it's actually pretty easy to do—but don't do it. Falling off the face of the earth is the quickest way to upset a client. Also, don't play hide-and-peek with your bugs. Be sure you're the one who informs the client about a bug. I'm assuming you want to do good work, so don't expect the client will never notice. You don't have to fix bugs as you find them—in fact, I'd recommend just listing bugs then take your time to analyze and fix them... all in good time. Finally, a great scope-creep deterrent is to accept every change from your client but keep saying "great, we'll add that to the list for version 2".

Tech Tips

Clearer implicit getter/setters:

I've always preferred explicit methods for getting/setting a class's private vars, but implicit getter/setters have some advantages (like being able to do `myInstance.myVar++`). Still I've avoided them because I felt it was hiding what would otherwise be public var declarations up at the top with all the other vars. I found a really easy way to resolve this issue while snooping at some code from **John C. Bland II** (johncblandii.com)—it's really just a formatting trick that makes it easier to read. Put your implicit getter/setters on one line each... up with the other vars... like:

```
private var _myVar:String;
public function get aVar():String { return _myVar }
//and so on
```

The "don't dock" key:

Love the new CS3 interface but hate how dock-happy it gets? Simply hold Ctrl while dragging a panel to prevent it docking.

Localization tip:

Roughly 80% of MSN Messenger users are outside the US—so obviously, all the text in the games I built needs to get translated. An interesting oversight I made initially was to hardwire where, within a sentence, to insert the user's name. For example, I looked at something like "Phillip has won the game" and built it dynamically: `<username><has_won_the_game_suffix>`. The problem was that while the translators could change "has won the game" I hardwired the position for the current user's name at the beginning. In some languages it needed to read (the equivalent of) "The person who won is Phillip". The solution turned out to let the translators edit: "[username] has won the game". (They would simply move the token "[username]" verbatim to the correct position and my code would find and replace that known token.) In retrospect it seems like a simple thing... but I overlooked it.

How to speed up the CS3 install:

This is a really great tip... if only it existed. I decided Adobe's installer situation bothers free loving people, students, and educators from within. Doing what people really expect never can temper bad reviews.

Reviews

AIR (adobe.com/go/air)

The idea of desktop apps—even connected desktop apps—is not particularly new... but I am excited about AIR for a couple reasons: it's drawing developers to the Flash Platform and it makes distribution easier through its seamless install. I have some concerns such as the fact I see a bigger movement away from the desktop than towards it. But there are some great use-cases for desktop apps... and maybe I'm not allowing for enough blurring of the line between desktop and web. As a user, I happen to like using a browser because of its known security model and distributed nature (nearly everything stays remote). I understand AIR apps overcome the apparent crisis that people are crashing or accidentally closing their browser. But if that's such a big deal, I don't know why browsers couldn't be changed to run in separate processes and include a "pin" option. Like I said, I'm happy for anything that increases demand for the kinds of applications I like to build (and desktop apps are more fun to build). I guess I'm just trying to dampen some of the baseless excitement for AIR. I'm okay with valid benefits, but it's only a matter of time before any hype-driven sparkle dulls and people see AIR at its true worth. It's not magic... but rather a way you can use your web skills (which, let's be fair, means Flash and Flex unless you're satisfied with the "app" qualities of HTML or like doing things the hard way with AJAX) to build desktop apps that extend the capabilities of Flash (mainly file access, drag and drop, windowing, and internet connection sensing).

Tortoise SVN (tortoisesvn.net)

Little did I know just how right **Joey Lott** (person13.com) was when I heard him say something like "working without version control is masochistic". It reminded me of a recommendation I got (in 1992) to use Debabelizer. That was an indispensable batch processing and palette creation tool (and beautiful example of arguably the worst UI in software history). You may not know Debabelizer, but likely you already know how version control lets you maintain every change in your project so that you can revert any time. Part of my resistance came from a paranoia that the software would mangle my data. Another issue was that SVN (a popular, if not the de facto, version control software) is a command line tool. Yeah, I know what I'm missing but I also need to "see" what I'm doing. Anyway, Tortoise gives you context menus from the file system and an intuitive interface for version control maneuvers (like rolling back or seeing the differences in two files). Tortoise does all this magic then simply sends requests to the geeky command line program.

Chumby (chumby.com)

When I got my hands on the alpha prototype (though **FITC.ca**) and mentioned it to some colleagues I hadn't expected to be such a target of envy! This bean-bag-looking always-on wi-fi internet-appliance with touch-screen and accelerometer is hot and it's not even out yet! I now have a production model (more RAM, faster processor, much improved UI, and Flashlite 3 meaning it supports video) to write a quick start article. Chumby has put the fun back into programming! Sure it's a technical challenge making stuff work on modest hardware—but it's *like* developing for mobile, just easier because this hardware isn't quite so low end and more fun because you can apply your Flash skills to make useful apps... such as a stereo remote that grabs album art from the internet (check out phillipkerman.com/chumby/). I do think that because it's not portable people will need to think of creative applications and, ultimately, its success rides on a killer app emerging. I expect to build an inexpensive home automation system. Regular consumers can download, install, and share applications via chumby.com.

Twitter (twitter.com)

I suppose it's a bit odd using a paper newsletter to tell you about twitter (though compared to a blog post, the thoughts here have developed over time and will remain in print longer). Anyway, this site is nothing more than a way to broadcast "what I'm doing now". I find it uniquely useful however. The group of folks whom I follow include some people who, while they're perfectly friendly acquaintances (or even friends), they are not people I'd likely go through the trouble of corresponding with through instant messaging (or that really old technology you may have heard of called "email"). The thing is, I can still casually follow what they're up to and actually learn useful information! For example, while attending a conference I heard one of the other sessions was really interesting so I jumped ship. Another time, I found someone who was working on a similar problem that I had and followed up with a DM (direct message). This site (there are other similar ones) could use some filtering settings—but there's an open API so many alternative interfaces are available. It's also sort of weird when you consider everyone follows a different set of friends and they may or may not overlap each other. The point though: it's not just a fad, it's a unique cross between IM, email, and blogs... and it *can* be useful. Here's a prediction: a standard presence data format will emerge allowing all such sites to easily share data.