

just for you

The White Wall

Overcoming Programmers Block

by **Paul Looney**

In the beginning there is the void, and the void covers the screen.

Readers of revUp have probably all encountered the programmers' version of writers' block.

You have a masterpiece in your mind. You can picture the world's adulation when it ships. Knighthood. A condo on Maui. Bronze statues of you and your computer in public places around the world. You are ready to write computer poetry - this time you might even do all of the code in rhyming iambic pentameter.

But now it is just you and the void, the white wall, the null screen of death, the light at the end of the endless tunnel...

Veteran revUp reader that you are, you probably have found many ways to wrestle the void. I'm sure other readers, and, of course, [the Editor](#), would like to hear your experiences and solutions. Meanwhile I'll offer some of mine...

To understand Programmers' Block it helps to understand flies.

In a moment we will discuss flies. But first there is an interesting story my Grandfather told on himself. It has almost nothing to do with our topic today but it is a funny tale and, as you'll see, it is hard to work into a conversation. Besides, it, too, concerns flies.

My Grandfather lived on a farm in the US Midwest. The nearest city, Davenport, Iowa, was 20 miles away so the family only went there once a year to get provisions they could not grow on the farm. Most shopping was done from catalogs.

One winter Grandfather noticed "Fly Killer - Guaranteed" in one of the catalogs. Just the thing for a farm full of horses, cows, pigs, and chickens. It was priced right and "Guaranteed".

Some time later the package arrived and, as he tells it, seemed a bit light. Indeed there were only two small pieces of hardware and a set of instructions. The instructions said "Put fly on (toy) anvil, hit with (toy) hammer - Guaranteed".

At this point you know nothing more about flies than when you started reading this article, but that is about to change.

Flies are fast. When they sense danger, they can get airborne in about 1/125th of a second (faster than most prize fighters can punch). It takes a faster shutter speed on a camera to freeze a flying fly, than a speeding race car. Armed with a swatter you don't have much chance - unless you know this: Flies have compound eyes. They are like a pair of highly fragmented mirrors. They aren't much good for what we call seeing, but they detect motion well. Any rapid, hostile motion and the eye signals the brain to do what it is programmed to do - flee.

The problem is, for the fly, anyway: the little brain can only process a small amount of information (movement from left, fly right; movement from right, fly left). If you come at the fly with a swatter in each hand (even slowly) the fly will just sit there trying desperately to determine which way to flee - until the end.

Programmers' Block is similar, too many choices, too much to do, too many ways to begin.

A nice thing about X Cards, like Rev, is the ability to just jump in anywhere, play around with an interface, revise it a dozen times, and, eventually, write some code and all is well. I prefer a little more "system". My favorite adage is from the wonderful world of carpentry: measure twice, cut once. I'm about to give you some specific tips. I realize circumstances, style, and experience may dictate exceptions, but, in the recurring words of the great Jan Schenkel, "I hope this helps."

Let's attack the void straight on. We will need a stack, so making one is a good beginning. We now have something on the screen. We have begun!

Before the euphoria of plastering a window on the void dissipates, we must confront the most important decision in the project:

The name

I can not overrate the importance of the name. If you can't name the project, you don't understand it, you can't program it! Without a name, the project will suck countless, wasted hours of dead-end coding from your wrist-weary hands.

I'm not talking about a "marketing" name, which is clever, concise, unique, and profitable. Those names are hard - and many good programmers never do them well (I am raising my own hand at this point). But you don't need a marketing name, you need a working name. "Untitled" is not a working name.

If the name eludes you now, read on a bit. If you don't have a name by step five, don't proceed without one.

Who

Who will use your masterpiece? Not just which company. Who in the company. Not just position, but person.

In software, features and stability are reciprocals. If you audience is "computer savvy" you can add more features because you have to spend less of your time (and budget) handling potential errors. If you are designing a product to be used by "management", trim features (even over their objections) and adds lots of error handling!

If you have not named it yet, pause here and think a bit on the scope of the project, the features. Are you creating a Bowie knife, one for the Swiss Army, or something that slices cheese? If you know this, naming is easy.

Why

Forget features.

What benefits do the users get from using this? Why would they use it? Do they want it? Will they actually use it? Does it save enough time, money, and/or aggravation to be worth programming?

If you don't have a name yet, try something radical: ask the prospective users to name it. In the process, they will be telling you exactly what they expect it to do.

How

From the first mockup onward watch users "use" the program. Have them show you how to use it. Watch them. Listen to them. They will tell you what is right and wrong.

Ask questions, then shut up. Your answers don't count.

Success requires bravely and repeatedly asking questions when you really do not want to hear the answers!

The Window

We have been selling a business system for over 20 years - 15 versions to date. You can tell every major version by glancing at it - the window size has evolved with each generation. We started with a window that filled the 512 pixel width of the Mac Plus screen, the current window is 1024 pixels wide. For a business system, we need all of the space we can reasonably get - and a 1024 width is the largest that will fit on most monitors shipped in the last decade. If you are creating a palette, your concern will be making it as compact as possible.

Don't forget "white space".

I've had the good fortune of having a number of highly qualified designers beat into my head the importance of white space. This is the part of the window without fields, buttons, or graphics - the background. It is critical. About one third of the window (more if possible) should be white space. Try a quick demo: create a small stack, put a button in it that touches all four edges of the stack. Looks "uncomfortable" doesn't it? Now enlarge the stack or shrink the button until there is about 30% white space. Much better.

Never let buttons or fields touch the edge of the window!

"White space" does not need to be "white".

White space, more than anything except typefaces, adds elegance to a stack.

About Typeface

I could probably write a hundred pages, or more, on type and type selection - but, if you've read this far, you already know this is not that kind of article. I use Verdana. It is available on OS X and Windows. It looks good at small sizes and large. It is reasonably compact.

The main negative with Verdana is the weird way it handles numbers. Non-proportional fonts have like-size numbers, text, and spaces. Proportional fonts have variable width text but generally all the numbers are one or two spaces wide (so you can make a calendar or calculator with them). But with Verdana, the numbers are about 1.2 spaces wide???

In Rev, you set the typeface for the stack and all buttons and fields in that stack inherit it - very convenient.

The Secret Magic Grid

The most important dimension on the window is... the leading! The leading establishes the proportions of all window objects. Proportion is a significant part of beauty.

Some designers have a natural sense of proportion. For the rest of us there is a mathematics of proportion going back at least as far as Euclid. Users are not generally aware of this but they see properly proportioned stacks as elegant, or just "right".

A grid is one quick and easy way of establishing proportion - and getting beyond the white wall. In my youth I would start a stack by putting a 12 x 12 pixel grid on the background. I aligned all fields and buttons to this grid. The margins were always 12, 24, or 36 pixels. All buttons and fields were some multiple of 12 wide. It soon occurred to me that the grid was actually the field leading - a field with lines 12 pixels apart (9 pt. type).

As screen resolution has increased I've moved to 15 pixel leading. My margins are 15 or 30 pixels. Buttons are 15 pixels tall. Objects are 15 pixels apart. Etc. The proper type size for 15 pixel leading would be 11.4 - so I use 11 pt.

Thanks to Rev's Size and Position panel in the Property Inspector, I no longer need the grid. If the top of an object is 62, I know it is two pixels too low. If the left of an object is 89, I know it is 1 pixel too far left. Object widths are easy to figure: start with 60, go up to 75, 90, 105, 120, etc. or down to 45, 30, 15, etc.

I look forward to the convergence of TV and Computer - when all screens will have a resolution of 1080 x 1920. I hope to talk about that soon!

Meanwhile, we have measured enough. A final thought then time to cut.

As important as planning is, and it is, all the good that comes of it can be lost by "analysis paralysis". Programers intent on shipping perfect product, seldom ship product. Give the customer/client honest value - not perfection. They will reward you with more suggestions, bug notes, enhancements, refinements, and improvements than you could possibly have thought to add on your own. By the way, we charge our customers to be beta testers. They understand that it is a privilege, and they value the invitation.

Meanwhile, Editor Heather, and I, would like to hear from you.

Paul Looney

► **main** menu

on **revUp** today

- The Dawning of a Brand New Array
- Localizing Your Software
- The White Wall
- FileMaker Pro to Revolution Conversion Procedure

what's **NEW**

