

Why Do Current Graphical User Interfaces Not Work Naturally & How They Can Be Fixed?

By Warren M Myers

User interface design, a part of the broader field of ergonomics, has been a challenging field to work in since man first tried making a tool for somebody else. Consider the lowly garden trowel. A trowel is simply a piece of wide metal connected to a handle, whereby its wielder may move small amount of earth to place seeds or seedlings in a garden.

Early trowels had hammered-metal scoops on the front, narrowing down to a thin, round spike that entered a similarly crude, small wooden club that more or less fitted the hand of the user. In the intervening centuries since the inception of that first trowel, engineers and designers have spent countless years trying to get the trowel to move more dirt, not disturb the other plants, not bend, be light, and be comfortable. Recent models utilize special alloys and shaping to deter the blade from deflecting under use, and have swooping, cushion-grip, non-slip handles that make using the tool, if not a delight, than at least comfortable. These new editions of the old standby discourage hand fatigue, and encourage people to spend more time in their gardens digging little holes without getting tired.

Computers have used myriad methods to communicate with the user ever since the first electronic computer, ENIAC, was built with flashing lights and hand-operated toggles to enter simple programs. Along with the amazing technological leaps made in the past 70 years, have come far broader acceptance and use of these machines in daily life. Thousands of people per year go into professional careers in which they will spend over 70% of their time staring at a computer, waiting for feedback on some process they have run, developing new tools, and even relaxing by playing games.

The first major leap in interface design came when some enterprising soul decided to hook up a teletype to a computer, so that the results of a program's run could be seen as text on paper. Humans have used written words and symbols for thousands of years to communicate with one another. Certainly having the computer return its work in the form of these easily recognizable symbols and words is a good idea.

The next big jump came when another brave soul decided to connect a dumb screen to a distant mainframe and have the results of the current job displayed in front of him as little dots of light in those familiar shapes we call words. Shortly after this, several people with far too much time on their hands came up with ways of drawing nifty little shapes on the screen, using either text or directly affecting each pixel to show a graphic.

Then, in the 1970's a small group of researchers at Xerox's Palo Alto Research Center (PARC) dreamed-up a way for the computer to show everything it was doing as pictures, and the user interacted by both typing (the old standby) and by selecting these pictures and 'clicking' on them with his mouse. Here was a huge break from the past. No longer was the operator tied completely to a keyboard, now he could move this little box with buttons around, and the computer mimicked his movements on the desk with movements of a cursor on the screen.

Apple Computer quickly snapped-up these researchers' ideas, and debuted the Apple Lisa and Macintosh in the early 1980's. Shortly thereafter, Microsoft showcased its new product, Windows, on IBM PC hardware. While the personal computer revolution had begun in the mid 1970's, the era of Graphical User Interfaces, or GUIs, really brought computers to the masses. No longer was there a requirement to learn archaic typed commands, like `du`, `ls`, `dir`, or `cp`. Rather, the user could open a 'folder' and 'view' the contents of the folder with little pictures – icons – next to them instead. GUIs fast became the most popular method of interacting with these machines by home users and office workers alike. Even engineering departments began seriously working with GUIs at this point, developing computer-aided drafting and design (CADD)

software that allowed the engineers to visualize what they wanted to build virtually, rather than through the age-old process of drawing blueprints, building little models, and then full-size mock-ups – all before the real object could be built.

At the same time, there was another growth spurt going on in the home entertainment industry. Gaming consoles, like Atari and Nintendo, became very popular. However, these machines didn't give feedback to their users in the same way 'typical' computers do, they gave feedback by your little plumber being attacked by a mutant turtle, or getting to the end of the race first, and a victory song playing.

But all of these impressive interface improvements have still been oriented around a simple, recurring theme: they're all square.

Squares – and rectangles – have been used as a primary interface shape for millennia. Just think of all the things you use on a daily basis that are rectangular: books, magazines, televisions, computer monitors, spatulas, soda boxes, parking slots – everything around us seems to be square. Squares stack nicely, look good on bookshelves, and are easy to fit into storage spaces.

But, there are a lot of things that don't do so well when shaped like a square: wheels, eggs, clothing, telephones, bolts, cups – just to name a few. Each of these has curves, points, or is round. They're not square. Sure, they may have squarish segments, but they're distinctly not square in their overall design.

The interface design on myriad 'real-world' items does not limit itself to just one shape. Where it is appropriate, other shapes are brought in for their inherent improvements over other modalities of thinking. It would seem strange to make compact discs triangular, because triangles are not efficient in their ratio of usable area to perimeter and manufacturing costs.

So, why have computer interface designers not put much apparent effort into coming up with alternative interface

shapes? One primary reason is that people expect monitors to be square. They think of a computer as a gigantic, electronic notebook, ledger sheet, newspaper, drawing pad, etc. Users have come to expect small-scale smoothing, such as done by the buttons and window corners in Mac OS X, Windows XP, etc. The buttons are rounded, the windows don't look sharp and forbidding, and overall the feel is a more comfortable, calm look.

Of course, the built-in assumptions by programmers, hardware designers, and users that the screen is a big Cartesian coordinate system, and huge piece of graph paper has reduced interface designers to limiting their thought processes to simple geometric shapes: rectangles, squares, and maybe a stray slant or two. The limit of roundness has been shown in buttons, window corners, and icons. A few enterprising souls have developed funky interfaces for programs like Winamp, where downloading and installing 'skins' rearrange the core components to some themed idea. But all of these attempts still run on a grid.

The time has come to investigate and try new graphical user interfaces. A grid shouldn't limit the GUI of the future. I think the time has come where people want a new, more efficient, different way to interact and control these tools. In nature, the strongest shape is the arch, or circle. Eggs are round because, a) it's hard to lay something that's square (ouch), and b) the shape distributes weight evenly all over the entire shell of the egg, rather than on a couple points of failure. Carrying this thought over to the digital world, I believe that the most logical shape for a main interface to exist in is the circle.

Throw out your built-in biases towards rectangular screens. What if screens were round? Instead of using the Cartesian x-y system, they could use polar coordinates. Rings would naturally exist in a circular environment. An entire system can be developed wherein the main interface, the buttons that allow you into different segments and levels of the system are based on concentric rings. The middle circle will bring you to the top level of the system. Festooned about the center circle can be wedges that describe the different segments of the

system. One such system might have the following wedges: human resources, accounting, manufacturing resources, and purchasing. Select the HR wedge to move into employee hiring, status, reviews, etc. Click over to purchasing, and make sure that the orders you've placed are en route, prepare new orders, anything you think of doing now in such a department.

After selecting one of these primary wedges, the center circle will still take you back to the previous level of detail. The wedges have been renamed to handle the different sub-sectors of that wing of the organization. Continued selection of these sectors will take you deeper into the system, and escaping will be a simple process of selecting the center choice multiple times. Perhaps there would also be a 'home screen' choice that will bring you back out to the top level.

This new interaction can be extended to any application running, or even the Operating System itself. A device akin to Alt-Tab or Cmd-Tab would switch which control rings were visible on the selection screen. The OS might have wedges for applications, documents, search, and tools. Under tools would be things like user management, hardware/application installation and removal, and other system management tasks. The documents sector could have sub-wedges for music, movies, documents, pictures, databases, etc. These categories can be modified by the user, or the system, based on how given files have been tagged. Under applications could be categories such as games, business, development, internet, etc.

Some programs have already started utilizing inventive interfaces on their own, but there has not yet been a concerted effort put forth to bring these ideas to the mainstream. For computer and graphics card manufacturers, there should be a push to support things like this with a small second screen just to hold the control for the active application, separating the work from the control. Perhaps this could be done with a little bulb off the corner of your current monitor where the control interface can reside, while the program actually runs on the main screen.

I think this new paradigm can fully embrace this new control interface while at the same time keeping the traditional workspace we've grown accustomed to on the main screen. In fact, by moving the control system to a separate device, even more screen real estate will be available to view whatever document, webpage, drawing, etc we're interested in seeing. Instead of having cluttered menu and tool bars, they can all be shoved off onto a separate screen while the real work is done on the big monitor.

I originally developed this idea of ring control in 2000 as a mental exercise in 'thinking outside the box'. I thought outside the box. I abandoned the box. Interface designers need to do the same thing, and come up with truly new and innovative ways of interacting with our computers. I would gladly add on a small screen to run the control aspects of the programs I work with on a regular basis in order to have a cleaner environment in which to work. Both computer users and designers have become complacent in our thinking. New ideas must be tried to improve efficiency. Not every idea will be a good one, but we need to try all sorts to find those good ones.

[Warren M. Myers studied at Elon University in North Carolina, and is now helping Sigma Xi in Research Triangle Park NC move its IT department to do utility programming and systems administration. He began programming in the early 90s in BASIC, and learned C++ through working on a finite element analysis program with a friend who wanted to try out some ideas he had for working in a non-Fortran environment.]

Source: Ubiquity Volume 6, Issue (July 19 - 26 2005)
<http://www.acm.org/ubiquity>