(Approx. 1015 words)

## Tiles – The Other GUI

By Dick Maybach, Brookdale Computer User Group

www.bcug.com

n2nd (at) att.net

All of us, whether our operating system is Windows, OS-X, or Linux, use a graphical user interface that can be called WIMP, for its defining features: Windows, Icons, Menus, and Pointer. There is another system, called tiles, that some prefer because they can open, close, resize, and move between windows using only the keyboard. They argue that continually moving their hands to the mouse slows their work. The first window screen I saw, years ago, was a tiled one running on Unix.

When we use a particular method for a long time, we begin to think of it as the only method, which of course is seldom true. If you’d like to gain some perspective by experimenting with tiles, you can try Regolith Linux, <https://regolith-linux.org/>. Be warned, however, that even though it’s just Ubuntu with a different user interface, it’s probably very different from anything you’ve ever used, and brace yourself for some adventure.

Regardless of whether you ever use it, knowing a bit about tiles will give you some perspective on user interfaces. The figure shows the desktop that results when you first boot Regolith and hit the key combination <Win><Shift><?>, where <Win> is the Windows key on a PC.



Figure 1. Initial Regolith Desktop with its Help Panel.

This key combination brings up the help panel, shown on the right in Figure 1. You can dismiss it with the same key combination you used to call it. (You’ll be referring to this often when you first begin using Regolith.) The first application opened occupies the full screen, as shown in Figure 2, where the browser is looking at the Regolith website.



Figure 2. One Application Open.

When the second application opens, it splits the screen with the first one; you can choose whether the split is horizontal or vertical. Figure 3 shows the result of doing two splits. With Firefox open, I first split the screen vertically so the file manager occupied its bottom half. Then I split the file manager tile horizontally to give its right half to the LibreOffice word processor. Finally, I increased the sizes of the browser and the LibreOffice tiles.



Figure 3. Three Applications Open.

Just as with a WIMP system, the layout is dynamic. In Figure 4, I’ve rearranged the tiles, with the terminal on the left, Firefox in the center, and the file manager on the right.



Figure 4. Rearranged Tiles.

In actual use, you will often need a more complex layout. Figure 5 shows one for managing a website. The large tiles at the upper left and right show the HTML code that generates a home page and the same page as a user would see it, respectively. Thus, as you make changes to the code, you can see their effect immediately. Between these two tiles are two smaller ones for file browsers, making it convenient to copy files on the PC. Although the tiles are smaller, they are large enough for the application. At the bottom is a tile with a terminal for command-line operations, which also can be small, as command-line commands are usually concise. The last tile is for GIMP, and is really just a placeholder, as the view would be toggled to full screen, as shown in Figure 6, to do graphics work.



Figure 5. Website Managing Example with Six Tiles.



Figure 6. GIMP Expanded to Full-screen Mode.

Note that the desktop is never visible, as the tiles always entirely cover it. Also, as new tiles are always added by splitting an existing one, you do have to plan more than with a WIMP scheme. The latter is much more ad hoc since you can place windows wherever you like, without regard for other windows. Although you can confuse yourself if something vital is covered, which can’t happen with tiling. Tiling is quite flexible once you become used to it, as you can resize and move tiles, and enlarge them to full screen if needed. What takes practice is that managing the tiles is done from the keyboard rather than with the mouse. At first, this is quite slow, as you have to refer constantly to the help screen, but after you learn the common commands, your hands leave the keyboard only when you have to use the mouse within an application. This brings the speed of the command line to a graphical environment.

Working in an application is the same whether you manage the interface using a WIMP or a tiled strategy; you use the keyboard and mouse identically. I do find that with tiles, I’m more likely to work in full-screen mode, as the size of my application window is likely to be small when tiled.

Tiling window managers are available for most operating systems, <https://en.wikipedia.org/wiki/Tiling_window_manager>. However, because they are so different from the WIMP approach that most of us use, your early experiences are likely to be frustrating. A safe way to experiment is to install Regolith on a live memory stick or a virtual machine. (Ubuntu users can install Regolith as an alternate desktop; see their website.) Yes, it’s Ubuntu Linux, but the main differences between Linux and Windows lie in their windows managers; and these disappear once you begin working on an application. Put another way, the differences between Regolith and Windows with a tiling user interface are quite small. Finally, if you are tempted to change your Microsoft windows manager to a tiling one, read the caveats in the above reference before you do, as there are significant issues.

While it’s interesting to investigate tiling, few people are likely to use it. In the WIMP approach, most common operations are self-evident. If you see something, moving the cursor over it and clicking starts the operation. Typing <Win><arrow-key> to select a tile, or <Win><r> to begin resizing one is less obvious. But if you spend many hours each day on your PC and especially if you use the command line frequently, using a tiled interface could make you more efficient.