Virtualization is one of the dark arts of tech geeks. We don’t fully understand how it works, but we wield it to create amazing, magical things. Like server redundancy (which is a very magical thing, trust me).

For the average computer user, virtualization is a far less useful thing, but it does serve one key purpose: running old software in old operating systems. (For my primer on operating systems, check out the February 2017 issue on The Metropolitan’s Website.)

Yet the process of installing and running a virtual machine can be very tricky. I’d like to provide a simple guide to doing so, with the end goal of running two very old games that will no longer work on modern Windows: Maxis’ SimPark, and a fan-made Mario DOS game (available at <http://www.wieringsoftware.nl>), both childhood favorites of mine.

## First, There Might be an Easier Way

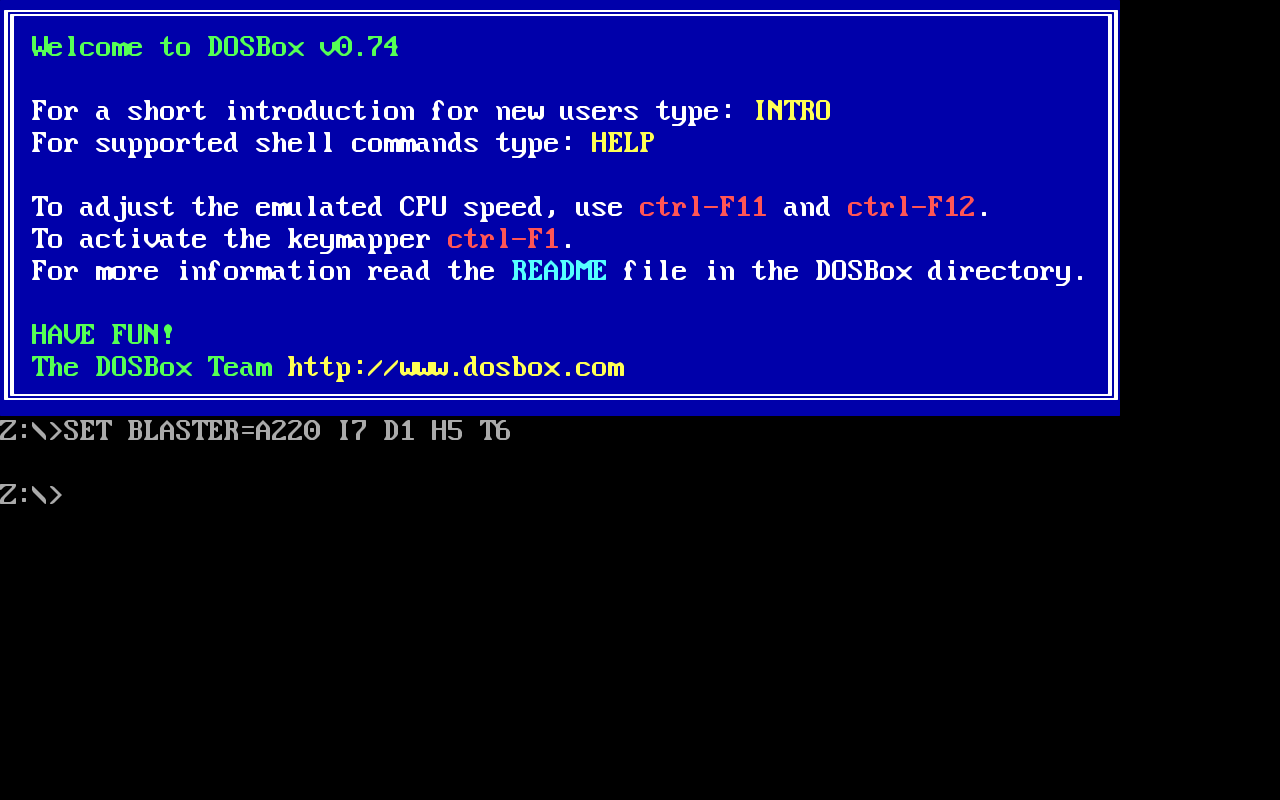
It deserves to be mentioned that a lot of ancient software can run with a great deal of success using *emulators.* There isn’t actually much of a difference when we speak of a “virtualizer” versus an “emulator. The distinction boils down to whether most of the emulation is occurring on hardware (then it’s a virtualizer) or on software (in which case, it’s an emulator).

Many people have already heard of emulators, and might be scared away by a stigma of illegality surrounding them. This is partially earned. While emulator software itself is legal in the United States, the typical use of an emulator is to run copyrighted software, whose distribution and possession may, in some instances, violate various copyright laws.

What I’m concerned with in this article is running old CDs and fan games. Assuming you got the software legally in the first place, it is 100 percent legal to it using an emulator.

With that out of the way, the main emulator for running old Windows games is *DOSBox* (available at dosbox.com). In fact, this is such a popular solution that many of the old games sold on Steam (steam.com) and Good Old Games (gog.com) actually come with DOSBox. After all, why should old studios bother rewriting their games when something like DOSBox can instantly allow them to be run on a modern system instead?

Installing DOSBox is trivial (just download and run the installation wizard), but getting it to work with your old software is another matter.



Caption: After starting DOSBox . . . now what? [Note: if possible, print include this image in the print version.]

In some cases, it’s as simple as opening the .exe file you want to run with DOSBox. I could do this with my DOS Mario game, but it ran much slower than I expected. There’s a simple explanation for this: DOSBox defaults to emulating a much, much older CPU than the game expects. To correct this, I increased DOSBox’s cycle speed with the shortcut Ctrl+F12 (I could also have lowered it with Ctrl+F11). For the Mario game, I simply needed to get it to around 15,000 (from the default of 3,000), and it was able to run smoothly.



Caption: A properly running knock-off Mario DOS game. [Note: if possible, print include this image in the print version.]

Another emulator that is still being developed is Win3mu (win3mu.com), specifically designed to run Windows 3 applications. In practice, there aren’t many of those, but it’s worth keeping an eye on as well.

## When You Need a Full Operating System

DOSBox isn’t perfect. In fact, it’s quite limited: it only runs games designed for MS-DOS, one of Microsoft’s operating systems from 1981 to 2000. In theory, games made through 2000 would work with DOSBox. Most games actually switched from being developed for DOS to being developed for the more advanced Windows platform beginning with Windows 95. If the game you want to run came out after 95 (like the 1996 SimPark), odds are DOSBox won’t run it.

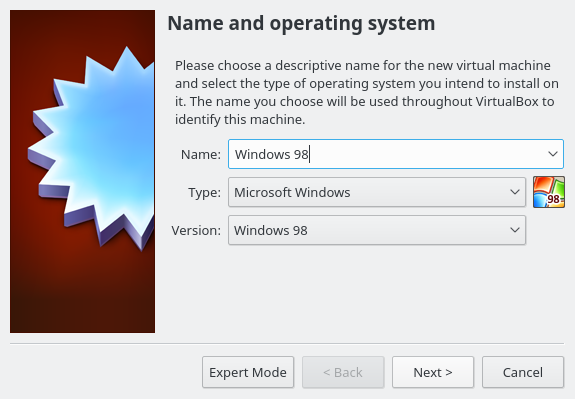
As a result, a full Windows 9x virtual machine is needed. There are two popular virtualizers out there: VirtualBox and VMWare. VMWare’s paid offering, VMWare Workstation, is probably the best option, but it costs money. If you already happen to have a subscription (some classes at MetroState will include them for the course), go for it. If not, it’s a choice between VMWare’s free offering, VMWare Player, and Oracle’s free and open-source offering, VirtualBox (virtualbox.org). Given a choice between the two, I generally prefer VirtualBox. This is because it includes custom scaling options (which are a must on high-DPI screens like Apple’s Retina displays), is slightly faster in many situations, and makes it easier to move over to VMWare than the other way around.

One limitation is that it can be difficult to install Windows 3 or Windows 95 on VirtaulBox. Windows 98 is generally compatible with all programs written for those older operating systems, however, so this isn’t really a problem as long as we decide to go with Windows 98.

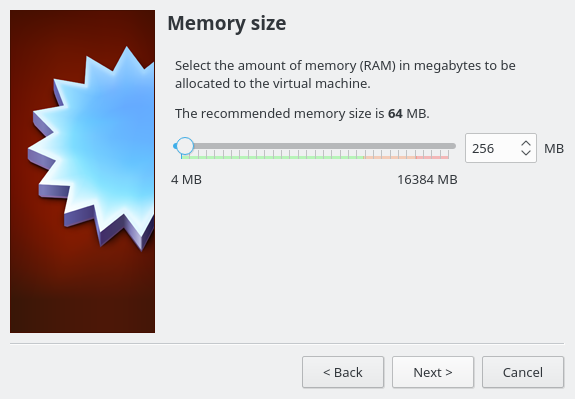
Do note, as well, that while my demonstration will use Windows 98, for most games Windows 2000 and Windows XP will also work just fine, and will still perform very well on most modern computers.

Once you’ve installed your virtualizer (I’ll use VirtualBox for my demonstration), you should go ahead and procure a copy of Windows 98—maybe you still have a CD lying around somewhere or maybe you can buy a copy off of eBay. Go ahead and create a new virtual machine and give it a reasonable amount of RAM. For Windows 98, this should ***never be more***than 512MB (if you use more, Windows 98 will likely fail to boot—I usually use 256MB). When asked to create a hard drive, do so, giving it at least a couple gigabytes of space. The OS itself takes up around 250MB, but you almost certainly have hundreds of gigabytes to spare, and we could be installing quite a few applications into the virtual machine.

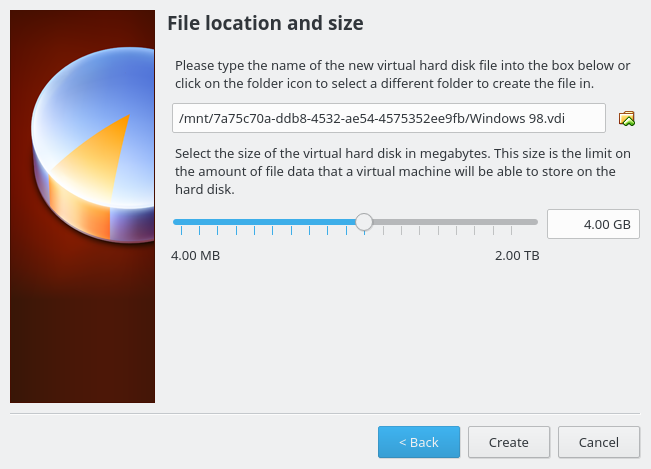
Finally, depending on your virtualizer, you will either be asked to insert an installation disk or you will do so once you start the virtual machine for the first time.



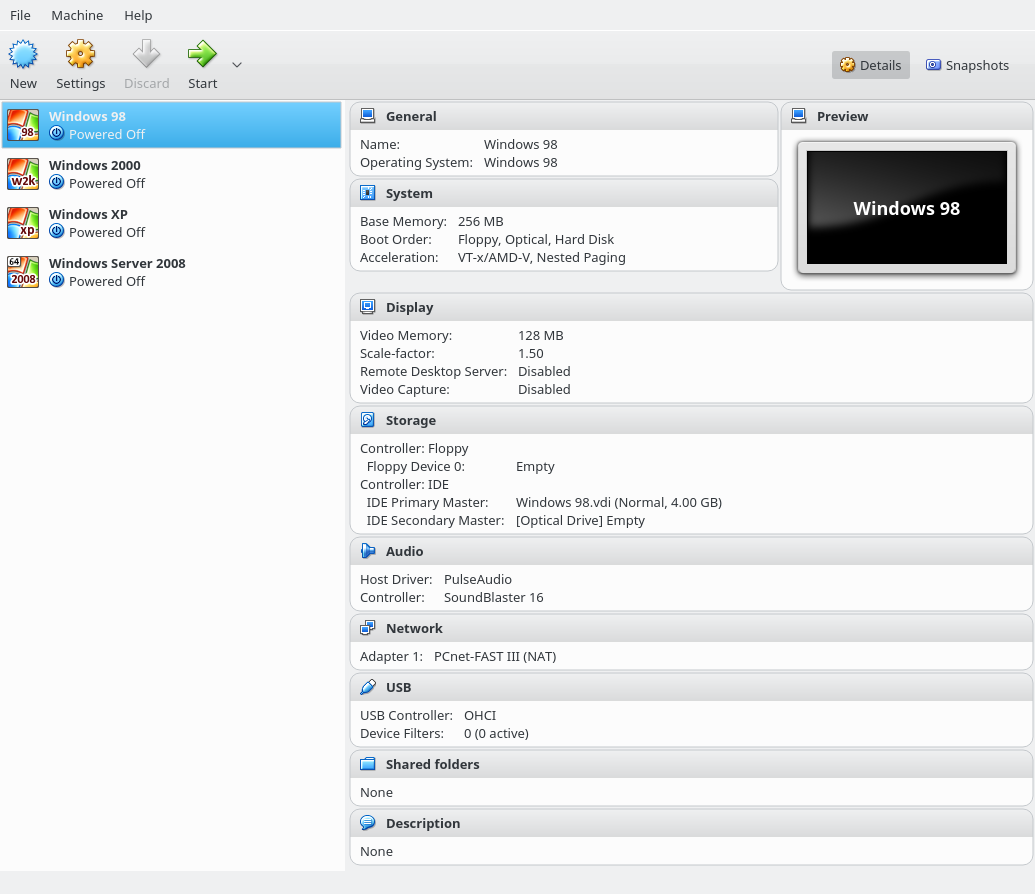
Step 1: Create the Virtual Machine



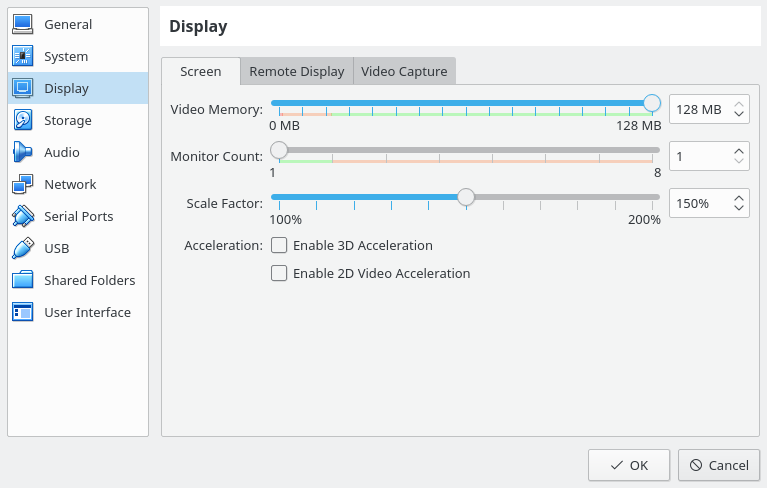
Step 2: Set the memory to somewhere between 64MB and 512MB. 256MB is a good medium.



Step 3: Create a hard drive image of somewhere between 256MB and 20GB. 4GB is a good medium.



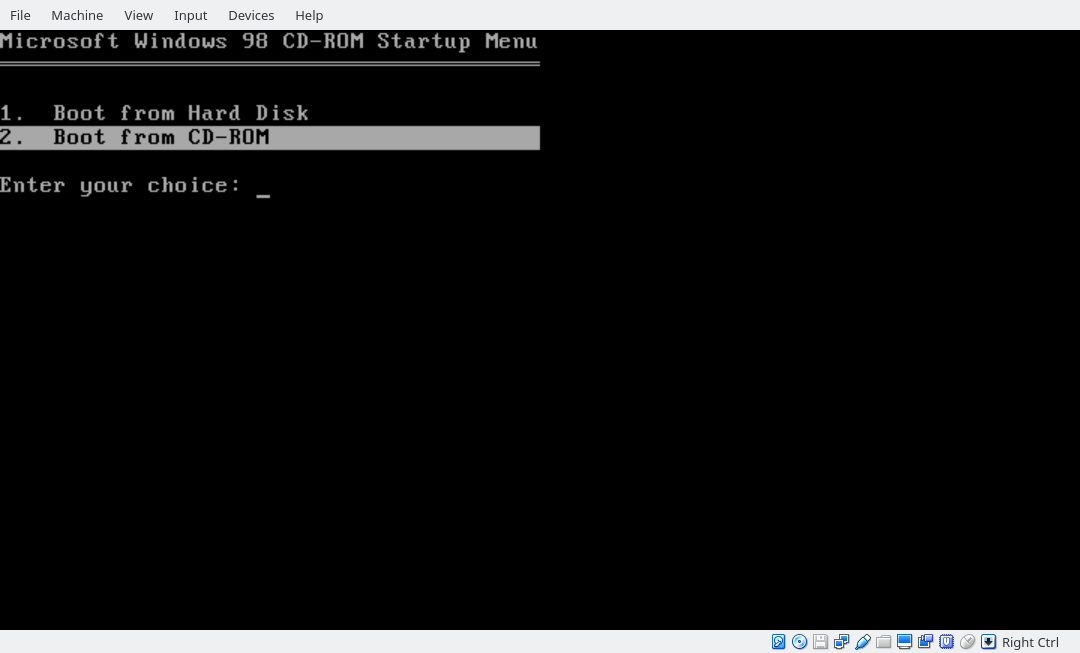
Step 4: After all of that, it should look like this.



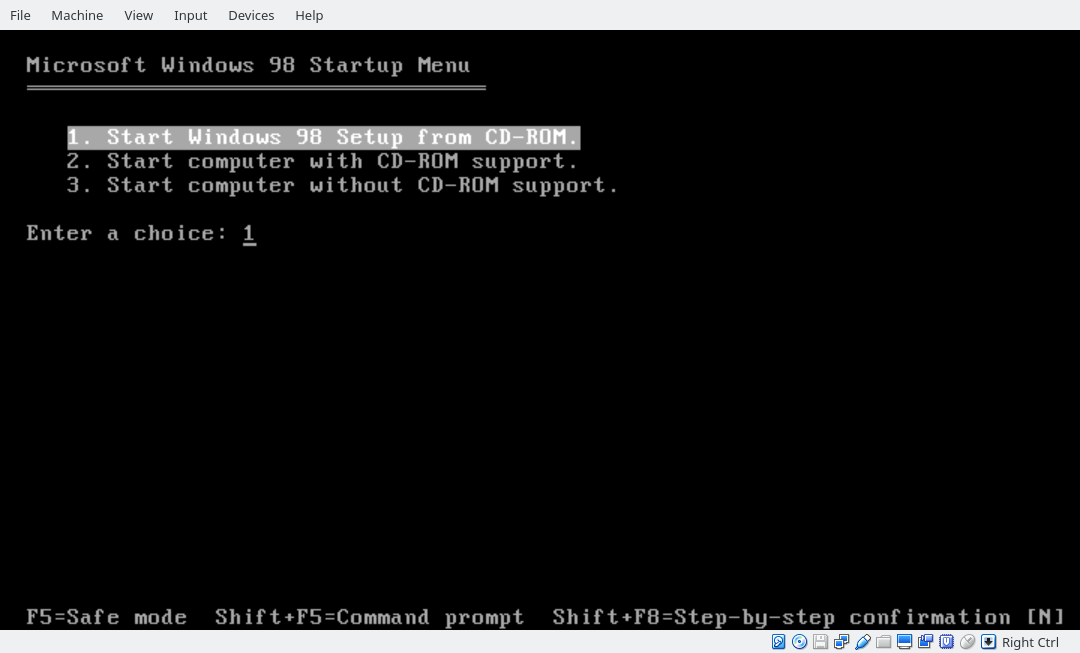
Step 5: If you want, you may also go into settings and adjust the zoom factor. For my display, 1.5 is helpful.

Now, we’re ready to start the operating system installation itself. 1) Start up the virtual machine you just created, 2) select the installation media, if necessary, and 3) if asked, make sure to boot from CD. You’ll then start working through the installation wizard.

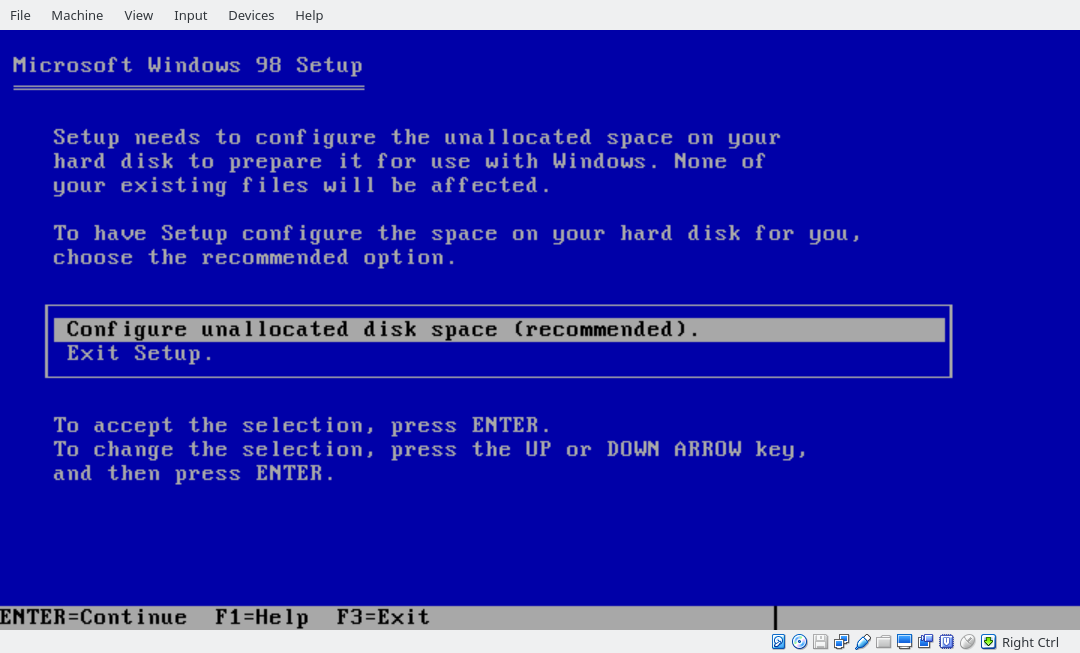
The installation wizard is pretty straight forward. For the first part, you will configure your disk by choosing “Configure unallocated disk space,” and, when asked, should *enable* support for large disks. The system will restart, you’ll need to boot from CD-ROM a second time, and the setup wizard will format your disk. Finally, it will perform a system check and start the proper Windows 98 installation.



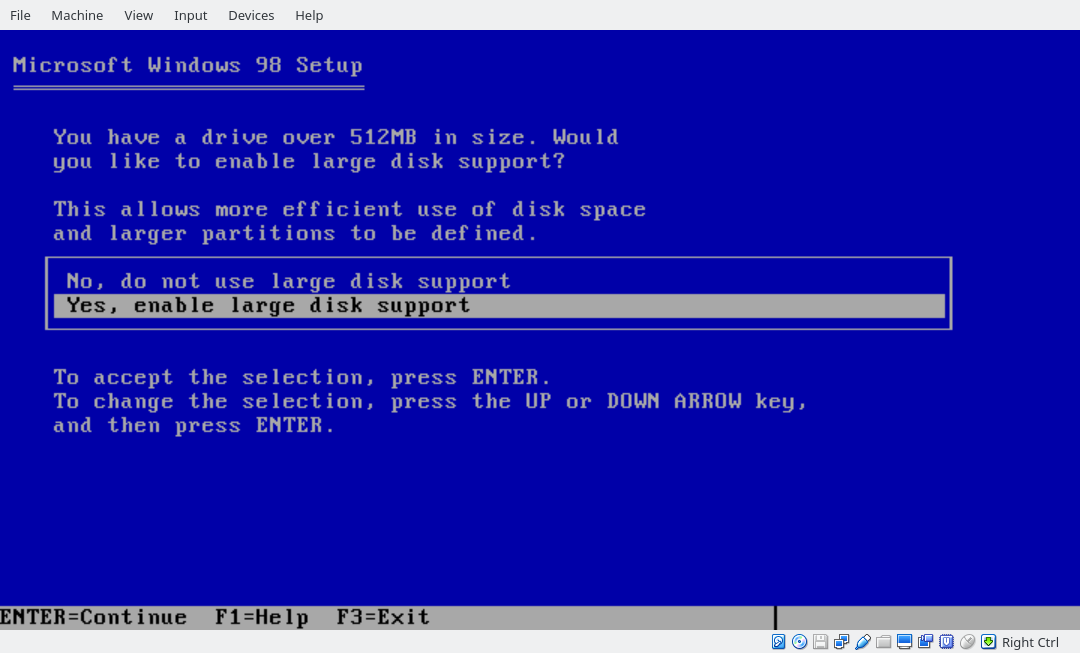
Step 1: Boot from CD



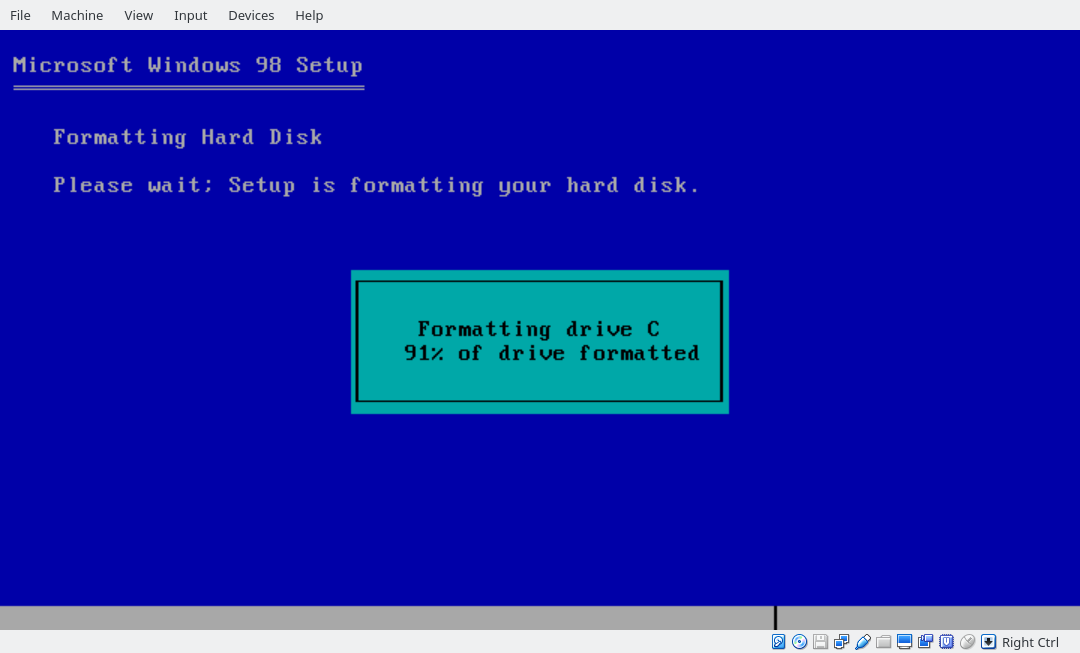
Step 2: Start the setup.



Step 3: Configure your disk.



Step 4: Enable large disk support.



Step 5: Restart, boot from CD-ROM again, and wait for the setup wizard to format your virtual disk image.

In the next stage, we’ll be using the fancier Windows 98 Setup. You will start by choosing where to install Windows (back in these days, everything was installed on top of MS-DOS; that included Windows itself) and how much you want to install. You’ll wait a while, click “next” a bunch of times, and eventually the virtual machine will restart again. This time, you want to boot to the hard disk instead of the CD-ROM. You’ll get an opportunity to enter your name, a Windows product key and then set the time. After a few more restarts, you should have a desktop.



Step 1: Click “Continue.” Click “Next” to install to C:\WINDOWS.



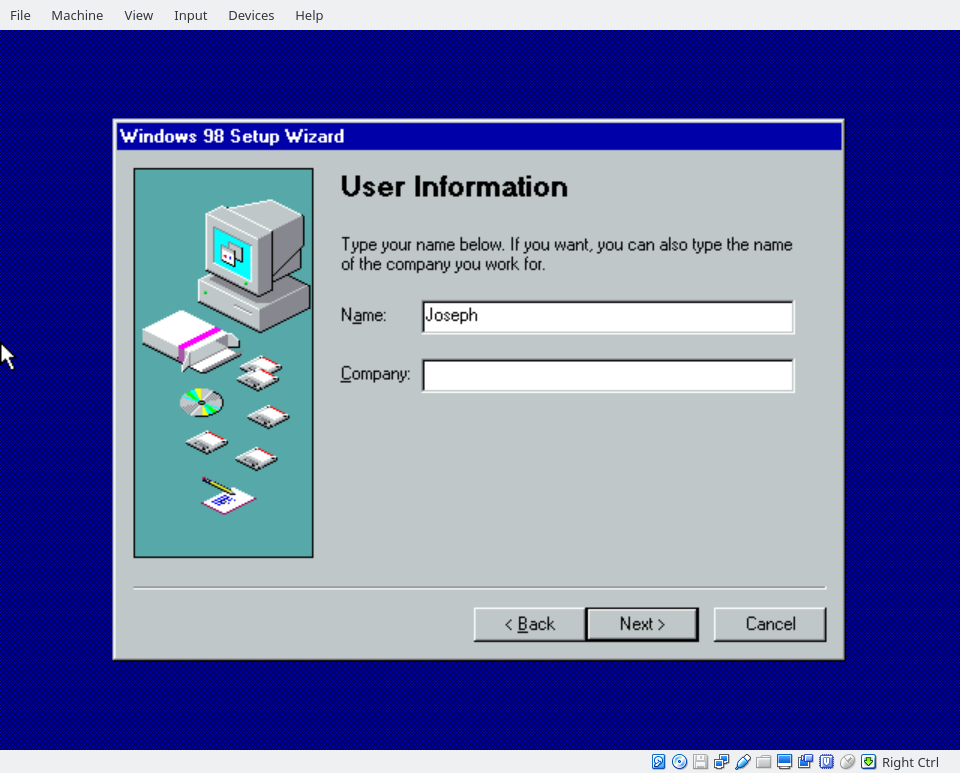
Step 2: “Typical” is fine here, but you can also go into “advanced” and install some of the cooler stuff if you feel like it. Click “Next” a bunch of times.



Step 3: Wait. “Windows Just Got Better.” I feel like they’ve used that line a lot.



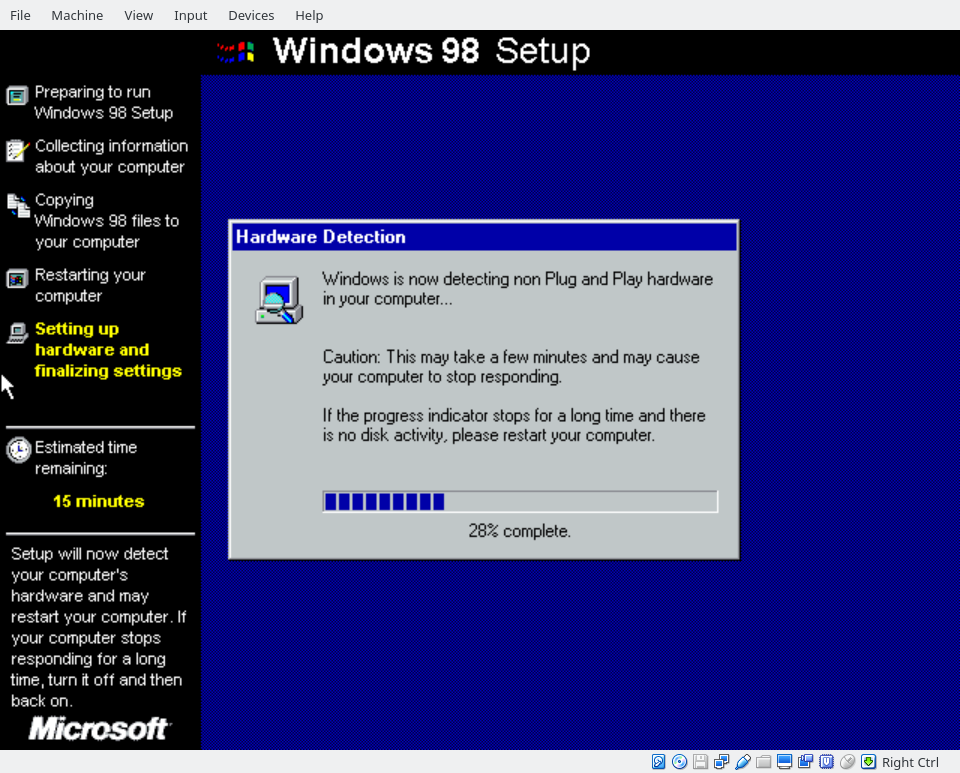
“Full-motion, full-screen multimedia titles”? This truly is the future! [Note: if we could get this one into the print edition just for the humor value, I’d love that.]



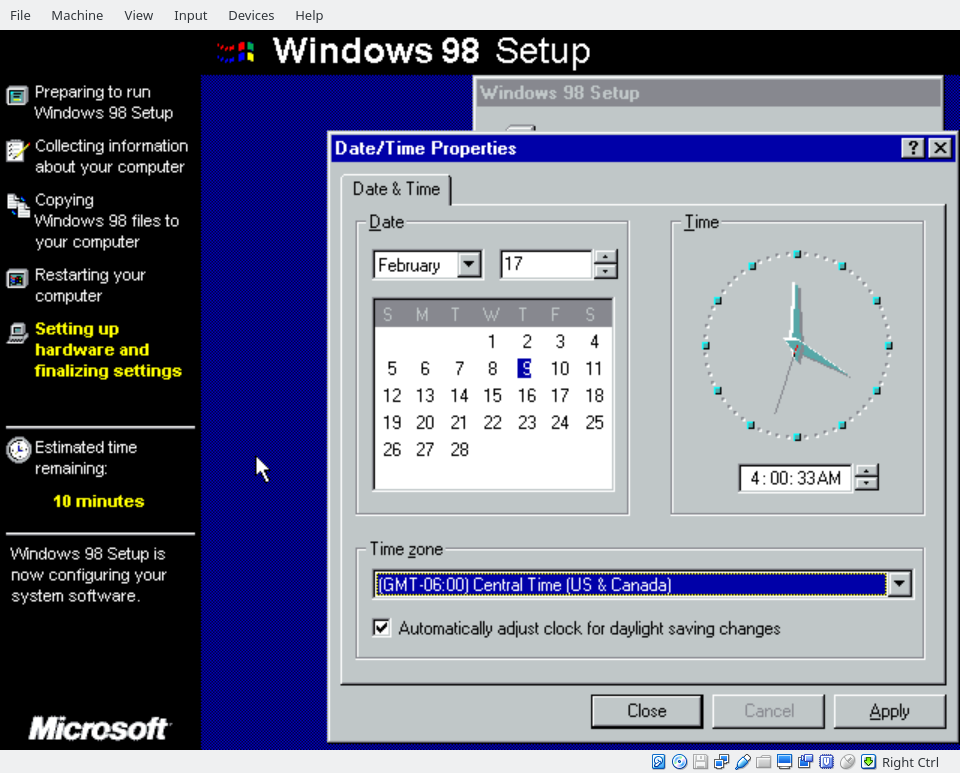
Step 4: You will reboot again. Choose to boot to hard disk this time, and enter a name once prompted. Click “Next,” accept the EULA, and click “Next” again.



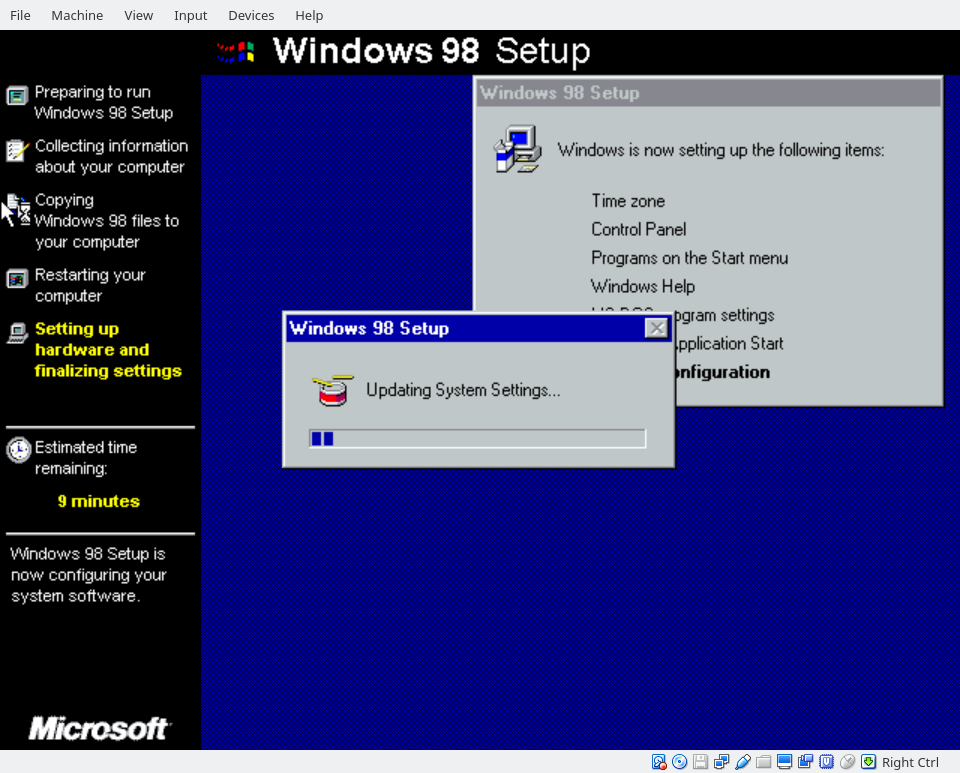
Step 5: Enter your product key. Hit “Next” and “Finish.”



Step 6: Wait some more.



Step 7: You can adjust your time zone if desired, but the time itself should be correct—it will be whatever time your main system clock is set to.

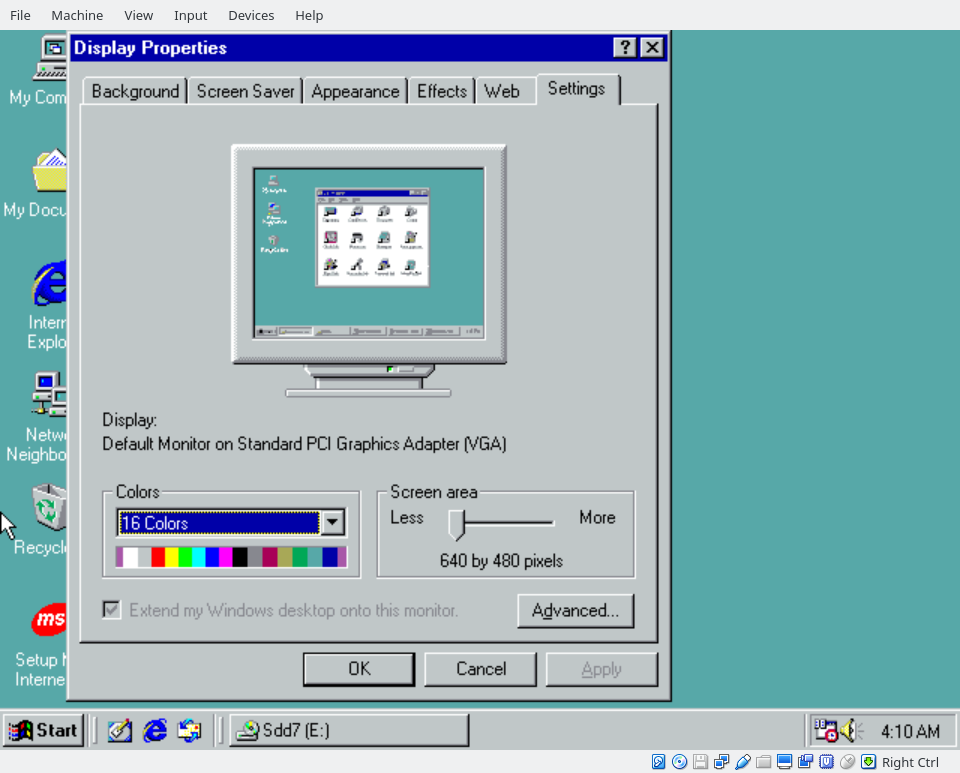


Step 8: Waiting is fun, right? At the end, it will restart one more time. Humorously, it also blue-screened after rebooting, though in Windows 98 you can often choose to ignore these errors.

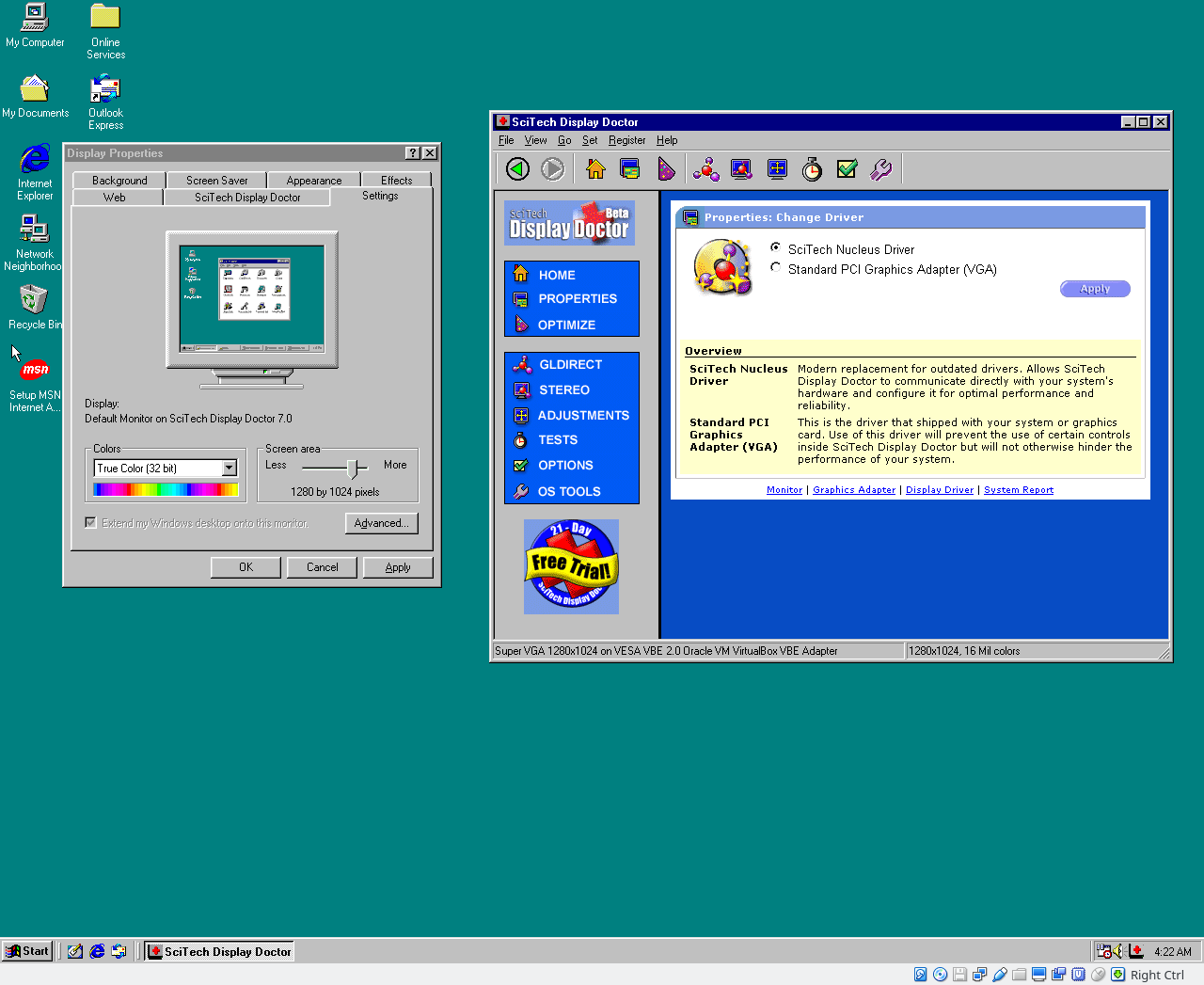


Step 9: Finally, a desktop!

As old as Windows 98 is, proper support for it tends to be limited. Right now, our desktop is only 640x480 pixels. It’s only 16-bit high color—whereas we should be able to display full 24-bit true color. There are some options. Scitech Display Driver is an abandoned driver that can successfully achieve high resolutions and 24-bit color. You can find it online, but I can’t speak to exactly how legal doing so would be. The laws surrounding “abandonware” (as in, you can’t find copies and the company shows no interest in future sales) are murky.



Well, that’s kind of disappointing.



The fix is easy, but tedious: find the Scitech Display Driver online, add it as a disk to your Virtual Machine, install the driver, and then choose the “SciTech Nuclear Driver” in the SciTech window.



After our display driver fix, SimPark runs great! As simple as popping in the original CD. [Note: if possible, print include this image in the print version.]