What even is a graphics card?

Clarissa Littler

If you're a gamer, you may be familiar with the idea of a graphics card. The graphics card is that really expensive brick of fans and metal that you plug into your motherboard in order to make your games look awesomer. But it's also an extra chip in your laptop or smartphone that's a lot less flashy. And what exactly does it do? It makes the graphics! Okay, article done, thanks for reading!

Wait, no. That was silly. Let's go through what exactly it does, but let’s work *backwards*.

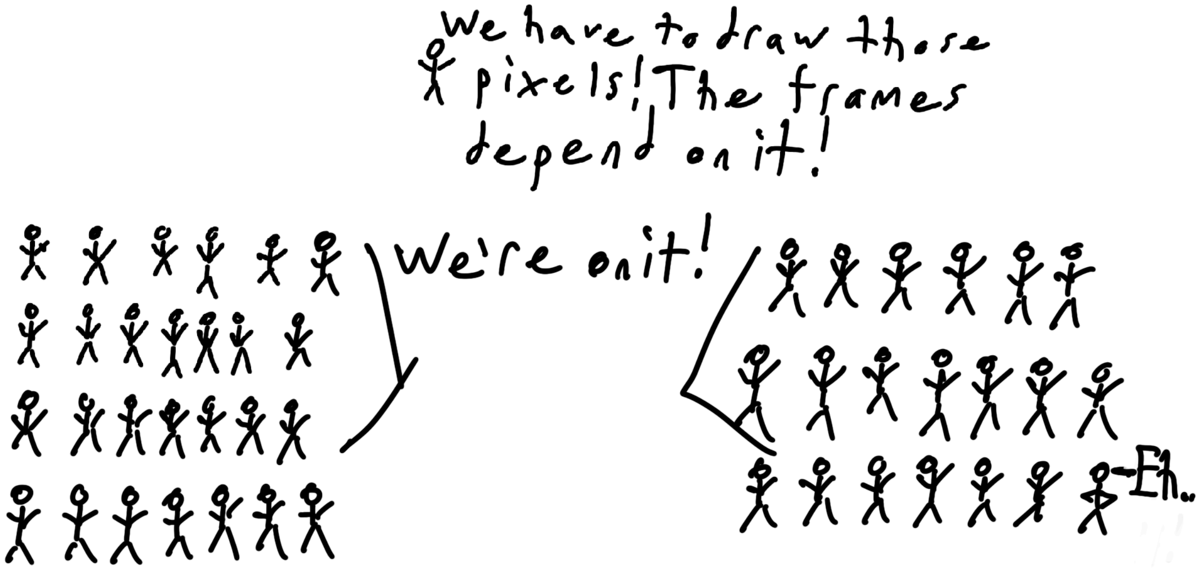
You're staring at your screen. Maybe it's your phone screen, or a laptop, or the 4k monitor on a sick gaming desktop. Either way, the screen is made up of millions of squares called *pixels*. These pixels can each display one of millions, or even billions on nicer monitors, of possible colors. Well, that's a subtle lie. It creates all of these many colors by mixing red, green, and blue light together.

How do all these pixels get their colors on the screen? All the data for them was sent to the monitor from the *frame buffer* of the video RAM. What's the frame buffer? It’s like a flipbook: where there's a stack of paper and you flick through them so fast it looks like a smooth animation. Except the frame buffer is a flipbook where the page **behind** the page you can currently see is getting drawn while you're looking at the one on top of it.

The "page" gets flipped 60 times per second, or 120 times a second for really fancy monitors, and the computer has to start drawing the next page.

Next, let's talk about how each pixel in the new "page" gets drawn. Each pixel needs to know what color it will be for this new frame. These colors are calculated independently on a pixel-by-pixel basis by the *fragment shaders*, which are the last phase of computation by the graphics card.

Before we get any further lets stop and think about the mind-blowing implication of this: we need to do calculations on millions of pixels one by one and we need to do that in less than *17 milliseconds*. That'd be **impossible** if you actually had to do a pixel at a time. That's what makes your graphics card special! You don't have to do these calculations one at a time but rather hundreds or even thousands at a time. How can a graphics card do this but the CPU in your computer can't? Basically, it's because the graphics card's many tiny processors are a lot more specialized than the CPU that runs your operating system. They're only good for a small number of operations: the ones they need to do to render graphics!



Caption: The author’s rendition of OpenGL giving commands to all the shaders

So how do the fragment shaders calculate the color of every pixel? Well, the fragment shaders are given a *bunch* of tiny triangles to work with. These tiny triangles either have a color at each vertex corner *or* some data about which part of a pre-made texture they cover. The fragment shader can take this data and "fill in" the appropriate colors to the interior of the triangle.

Caption: Fragment shaders *interpolating* between colors in a triangle

Where do the tiny triangles come from? They're put into place by the *vertex shaders*. The vertex shaders take in the 3D models of your game, which even when they look smooth are made up of a ton of triangles and squares, and then perform all the transformations to rotate, scale, and place them in the "3D space" of the game to figure out which triangles are visible and which ones are blocked by other solid objects.

We're almost to the beginning now! The code that loads up the 3D models, textures, and tells the shaders what to do and how to transform the graphics all comes from the game. But how does a game communicate with the graphics card? Through specialized APIs like OpenGL, Metal, or Vulkan! These are libraries of code that every game engine, like Unity, Godot, or Game Maker, uses under the hood to render graphics.

So that's a little summary of how a graphics card works, but it's not even remotely the end of the story. There's also the story of GPGPU, general-purpose GPU programming, where people use graphics cards to do all sorts of cool things unrelated to graphics. There's also the newest cool thing in graphics that only a few graphics cards can do: real-time ray tracing! We'll talk about all of these in future articles.