

Microspeak: Real estate and Airspace

 devblogs.microsoft.com/oldnewthing/20241126-00

November 26, 2024



Remember, Microspeak is not merely for jargon exclusive to Microsoft, but it's jargon that you need to know.

Today's Microspeak terms are *real estate* and *airspace* (one word, rather than the more natural *air space*).

The term *real estate* is rather commonly used in the industry. It is a jargon term to refer to space on the screen.

Do we really want to spend that much real estate on a status bar? Can we make the status bar appear only when there is something that wants to report status?

As the saying goes, "They're not making any more!" The pixels available on the screen is a limited resource, and if you decide to occupy some of it with a status bar, then somebody else loses their space. You have to decide how to assign space to each screen element to provide maximum value.

Building on the idea of *real estate* is *airspace*: Your program controls a certain amount of real estate, and different UI elements can occupy space on this land: A button here, an edit control there, and when two controls overlap, you have to decide which one is considered to be "on top" and which is "underneath". These types of negotiations are usually handled by whatever UI framework your application uses.

Things get more complicated when you are mixing multiple UI frameworks within a single window. For example, you might be working mostly in classic Win32, but you want to use a media player control that uses a DirectX swap chain or a compositor from the Microsoft.UI.Composition namespace. Without some way to coordinate between the two UI frameworks, you will have them both trying to render into the same pixels, or one of them will simply "take priority" over the other in a way that you don't want.

This is known as the "airspace problem".

Solutions (or the lack of solutions) for this problem vary depending on the frameworks involved. But at least now you know what the term means.