How can I reposition my window so it isn't covered by the touch keyboard?

devblogs.microsoft.com/oldnewthing/20150615-00

June 15, 2015

Raymond Chen

Last week, we saw how we could rearrange our window contents to avoid having the touch keyboard cover the edit control. But what if the touch keyboard covers the entire window? No amount of rearranging will help. We'll have to move our window.

Let's make these changes to the `DoLayout` function:

```
void
DoLayout(HWND hwnd, int cx, int cy, bool isKeyboardShowing = false)
{
  ...
  if (g_hwndChild) {
    ...
      if (IntersectRect(&rc, &rcEdit, &rcKeyboardClient)) {
        if (rcKeyboardClient.top > 50) {
          cyEdit = min(rcKeyboardClient.top, cy);
        } else if (isKeyboardShowing) {
          // need to reposition the entire window, ugh.
          int dyAdjust = 50 - rcKeyboardClient.top;
          RECT rcWindow;
          GetWindowRect(hwnd, &rcWindow);
          SetWindowPos(hwnd, nullptr,
            rcWindow.left, rcWindow.top - dyAdjust, 0, 0,
            SWP_NOZORDER | SWP_NOACTIVATE | SWP_NOSIZE);
          return;
        }
      }
      ...
}
```

If there are at least 50 pixels of the edit control visible, then we consider that good enough. If not, and if we are responding to the keyboard showing, then we take matters into our own hands and *move our window* so that there are 50 pixels of the edit control visible. I didn't bother adding a check to make sure we never moved beyond the top of the work area; I'll leave that as an exercise for the reader, seeing as it's more typing that tends to distract from the point of the article.

Raymond Chen

Follow