**Font Rendering**

Render strings as a set of glyph quads. The quads are indexing into a texture atlas (font atlas) containing all the glyphs being used for the font; for now that's just the ASCII set.

**Building a Font Atlas**

Need to sort by some measure; try to sort descending by each glyph's larger side

**Future Work - Handling Unicode**

If we do use Unicode, it'll be UTF-16, probably little endian (so BOM is necessary for external strings, at least). Each atlas can store about 256 characters, which splits (most) fonts into 256 pages. As uncompressed 8-bit greyscale bitmaps @ 512x512, a full font would need 64 MB to store all characters; evidently, we should only load the page atlases we need. To get a character, you'd look at the least significant half (wchar\_t isn't always 16 bits) of a character to get the page resource to load, pull that page, then index into the page by the most significant half to get the needed glyph.