Lecture notes 03.02.2020

- we'll step through all the code and learn what exactly it does
- how does glm represent angles?
- camera frustrum manipulation and all the values associated with it
- if near clip is too close floating point errors will happen – screw up the calculations
- adding small numbers to very large numbers can fuck up and either add to much or nothing at all because of the floating point standard
- we move stuff around in imaginary space, the camera has its own model space
- how can we arrive at the model, view, and projection matrices
- `uniform` in the shaders means we use the same mvp_matrix for all points
- we simulate all the stuff in 3D and then display all in 2D
- vertex shader is called after the vertex shader is done computing all its shit
- we retrieve the "address" of the mvp_matrix because we actually don't know where we should send the data
- y-x-z is a common rotation encoding, it is useful because it is often used