Task 2 Cube rendering

First, we setup the cube with an array of 3d vectors and its Model Matrix. The Model matrix starts as an Identity Matrix, which means that it is right at the center of the world (so no transformations are applied to the cube). Each vector in the array resembles the position of a vertex.

Second, I setup the view matrix, which is the position of the camera for our scene. I place it at position (0,0, -2) and let it look at the center (0,0,0) and define the up vector to be the y axis (0,1,0). I place it at z -2 so I have some space between the camera and the cube.

Third, we define the projection matrix, by setting the FOV to 90 degrees and min and max depth as well as the screen aspect ratio.

Fourth, we apply some transformations to the model matrix of the cube – at first a rotation and then a translation.

Then we log all the necessary data, and, in the end, we calculate all the screen positions. Therefore, we take the precalculated MVP matrix and multiply it to each vertex of the cube – which gives us the normalized pixel coordinate. This value then needs to be transformed to pixel space. In this task I deliberately ignored the clipping and a screen offset. That's the reason why we also can receive negative pixel coordinates after rotating the cube.