

CS 405 Project Step 1

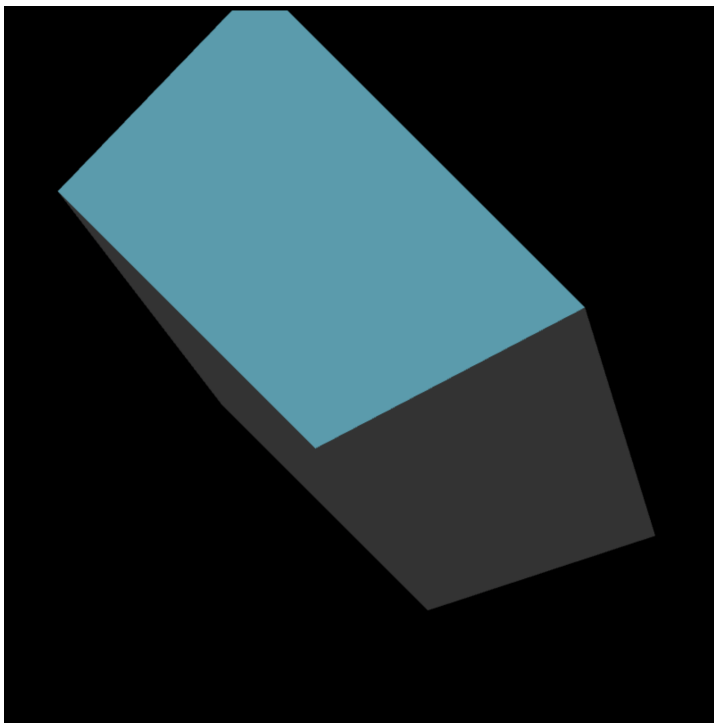
Gorkem Yar 27970

Task 1:

For this task, I used chatgpt 3.5 as specified in the project description. I gave the prompt to chatgpt in order to create the transformation matrix. The result generated by the chatgpt is in the form of Float32Array.

```
javascript Copy code  
  
const transformationMatrix = new Float32Array([  
    0.35355, -0.61237, 0.70711, 0.24495,  
    0.61237, 0.35355, 0.70711, -0.20495,  
    -0.70711, 0.70711, 0, 0,  
    0, 0, 0, 1  
]);
```

The object created using this transformation is given below.



My link is <https://chat.openai.com/share/f874c18f-b238-4d2a-adde-2df6a7696668>

Task 2:

For this task, I used the given `createScaleMatrix`, `createTranslationMatrix`, `createRotationMatrix_X`, `createRotationMatrix_Y`, `createRotationMatrix_Z`, and `multiplyMatrices` functions.

Firstly, I created the necessary matrix with the given parameters in the prompt. Then, I started to multiply the matrices. My order for multiplication is given below.

Translation*Scale*Rotation_Z*Rotation_Y*Rotation_X

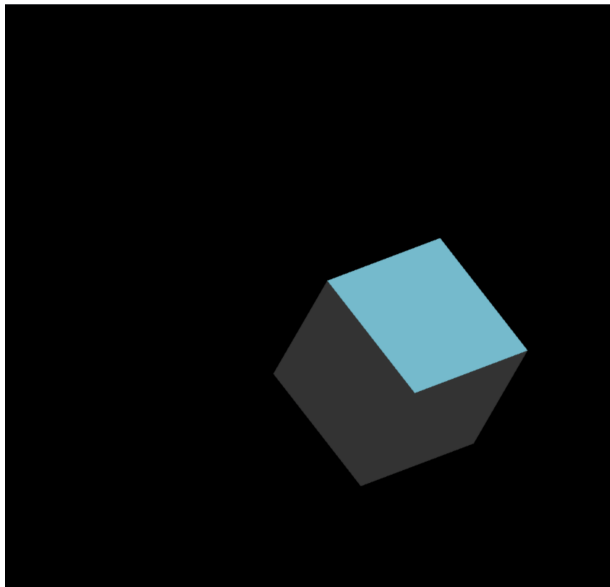
My order is the same as the lecture slides.

$$M = (\text{translation})(\text{shear})(\text{scaling})(\text{rotation})$$

The transformation starts with rotation and then continues with scale and translation. The resulting matrix is:

```
const transformation_matrix = new Float32Array(  
  [0.1767766922712326, -0.2866116404032102, 0.3695994506163247, 0.30000001192092896,  
   0.3061862071122423, 0.3695994506163247, 0.14016502888836202, -0.25, -0.7071067690849304,  
   0.3535533845424652, 0.6123724142244846, 0,  
   0, 0, 0, 1]);
```

Using this transformation the resulting image is given below.



The transformation in task1 differs from task2 because of two reasons. The first reason is the order of the transformations is different in the chatgpt version. The second reason is that the chatgpt 3.5 can make mistakes in the matrix multiplication operation. So, the transformation matrices differ which leads to different images.

Task 3:

As specified, I used chatgpt for this task as well. I asked chatgpt how to make the transformation animation smoothly. It advised me to use linear interpolation to make transition continuous. The code we generated is:

```
function getPeriodicMovement(startTime) {
  // Total duration of each cycle of the animation (in milliseconds)
  const cycleDuration = 10000; // 10 seconds

  // Time elapsed since the animation started
  const elapsedTime = (Date.now() - startTime) % cycleDuration;

  // Calculate the progress in the current cycle (between 0 and 1)
  const progressInCycle = elapsedTime / cycleDuration;

  // Define your initial and final transformation matrices for the object
  const initialTransform = createIdentityMatrix();
  const finalTransform = getModelViewMatrix();

  let interpolatedTransform;

  // Check if we are in the first 5 seconds (forward transformation)
  if (progressInCycle < 0.5) {
    // Interpolate forward from initial to final transformation
    interpolatedTransform = interpolateTransforms(initialTransform, finalTransform, progressInCycle * 2);
  } else {
    // Interpolate backward from final to initial transformation for the next 5 seconds
    const progressInBackwardCycle = (progressInCycle - 0.5) * 2;
    interpolatedTransform = interpolateTransforms(finalTransform, initialTransform, progressInBackwardCycle);
  }

  return interpolatedTransform;
}
```

```
// Example transformation matrix interpolation function
function interpolateTransforms(startTransform, endTransform, progress) {
  // You need to implement your own interpolation logic here
  // This can involve interpolating translation, rotation, and scale components
  // For a simple linear interpolation, you can do something like:
  const resultTransform = [];
  for (let i = 0; i < startTransform.length; i++) {
    const interpolatedValue = startTransform[i] + (endTransform[i] - startTransform[i]) * progress;
    resultTransform.push(interpolatedValue);
  }
  return resultTransform;
}
```

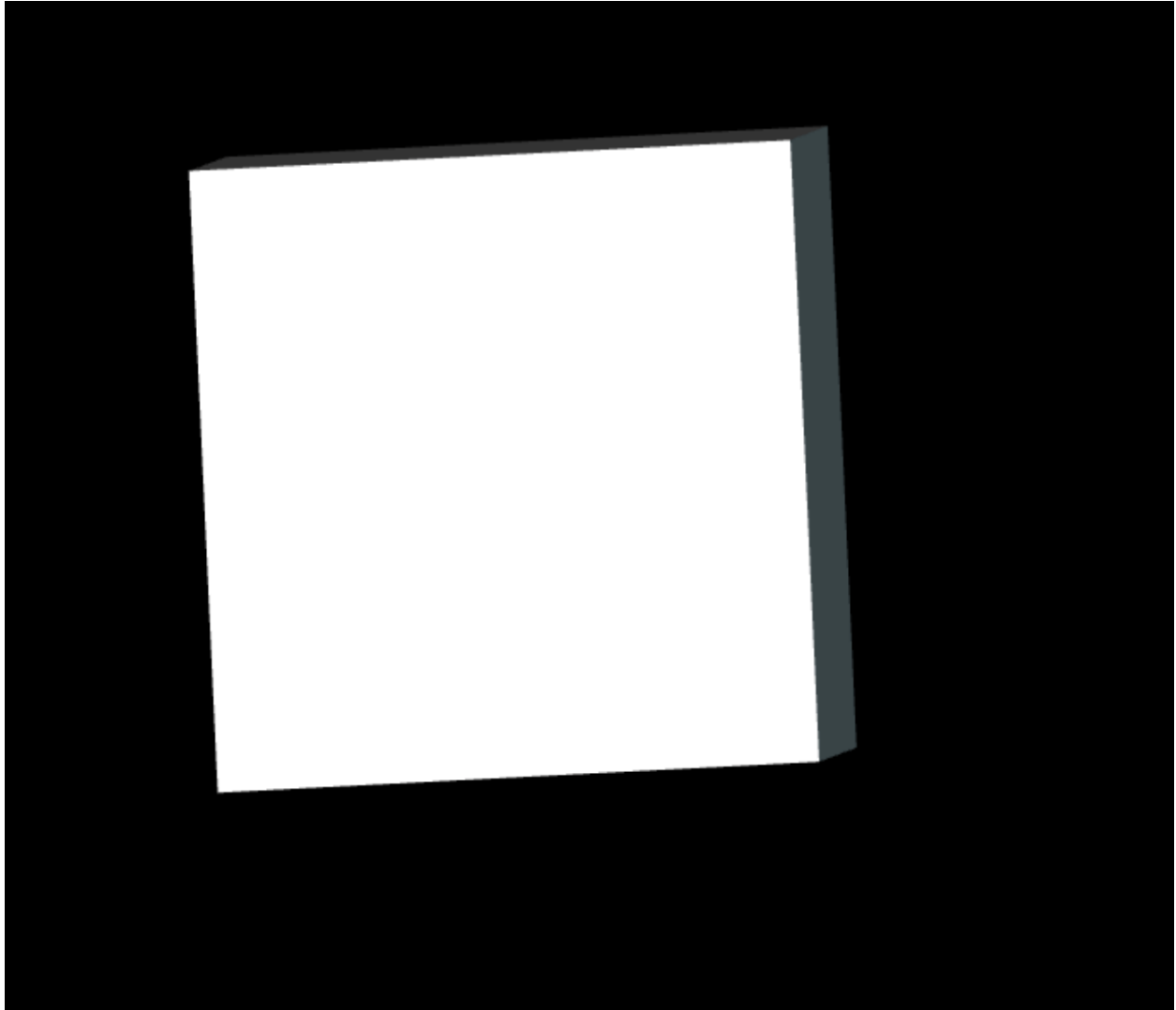
You, yesterday • project initialization

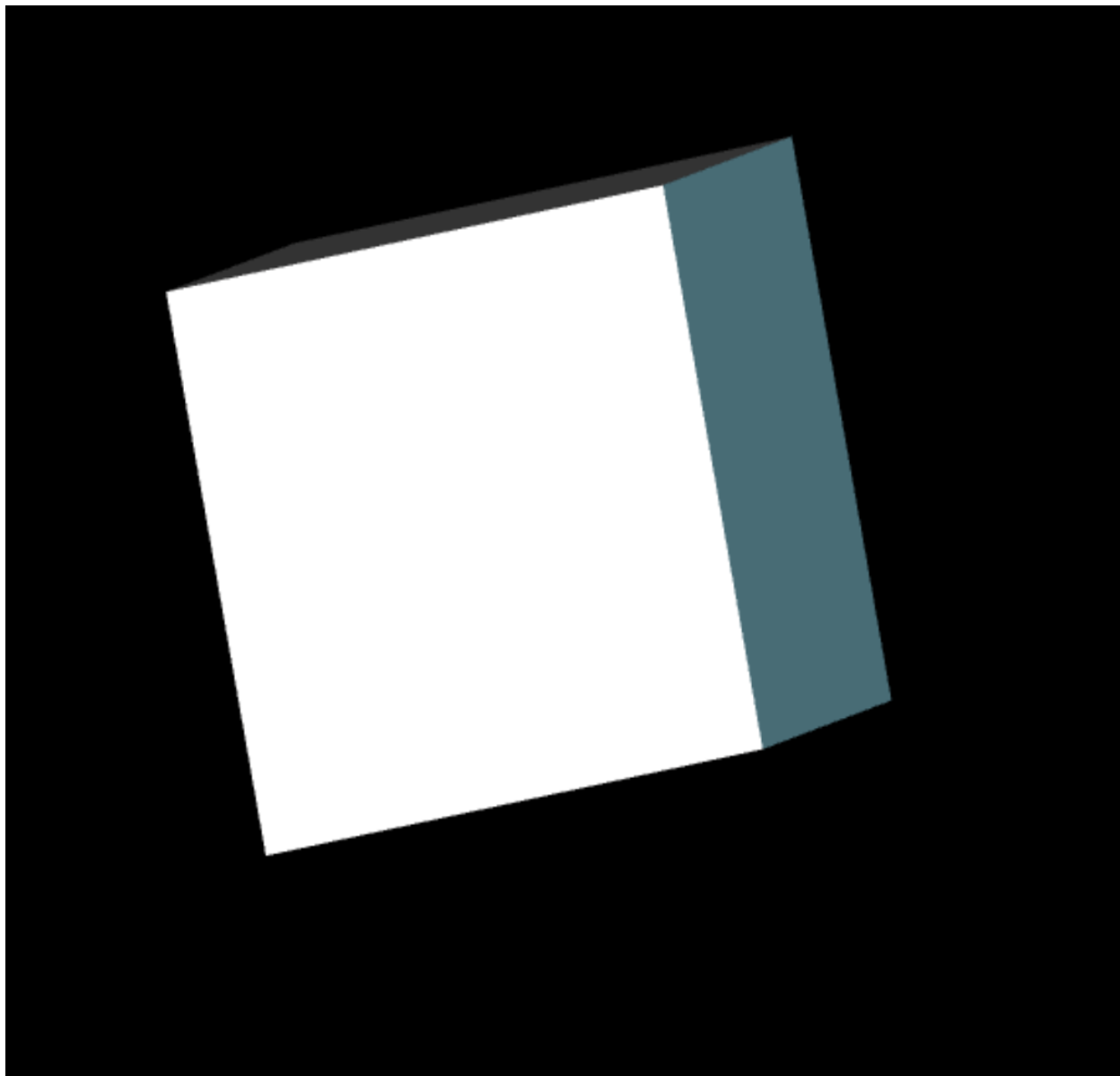
interpolateTransforms function calculates the transformation matrix with the time dependency. Firstly, it decides the initial and final transformation matrices. Later, using the time-variant it calculates the interpolation matrix and returns the result matrix.

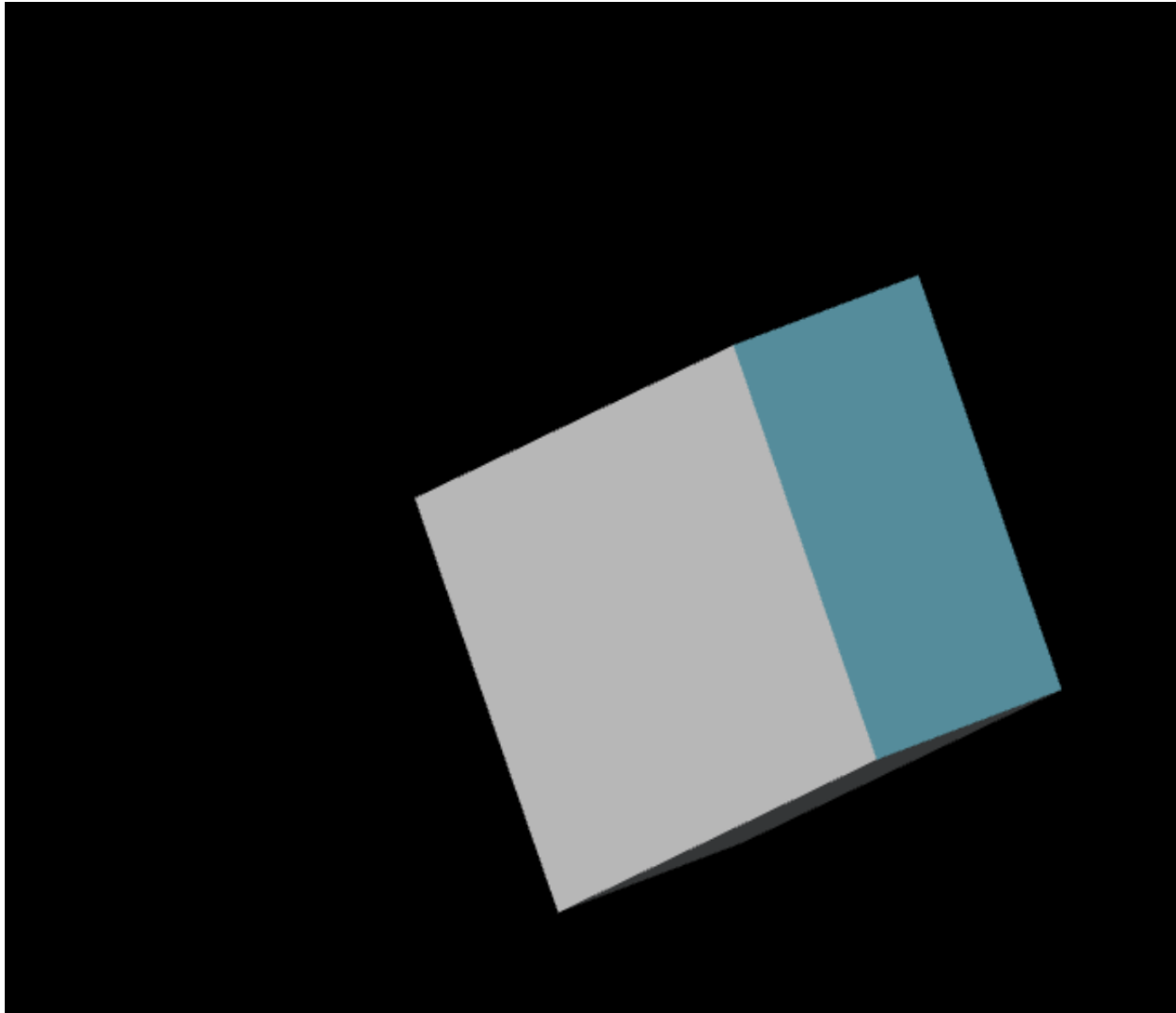
The chatgpt conversation link:

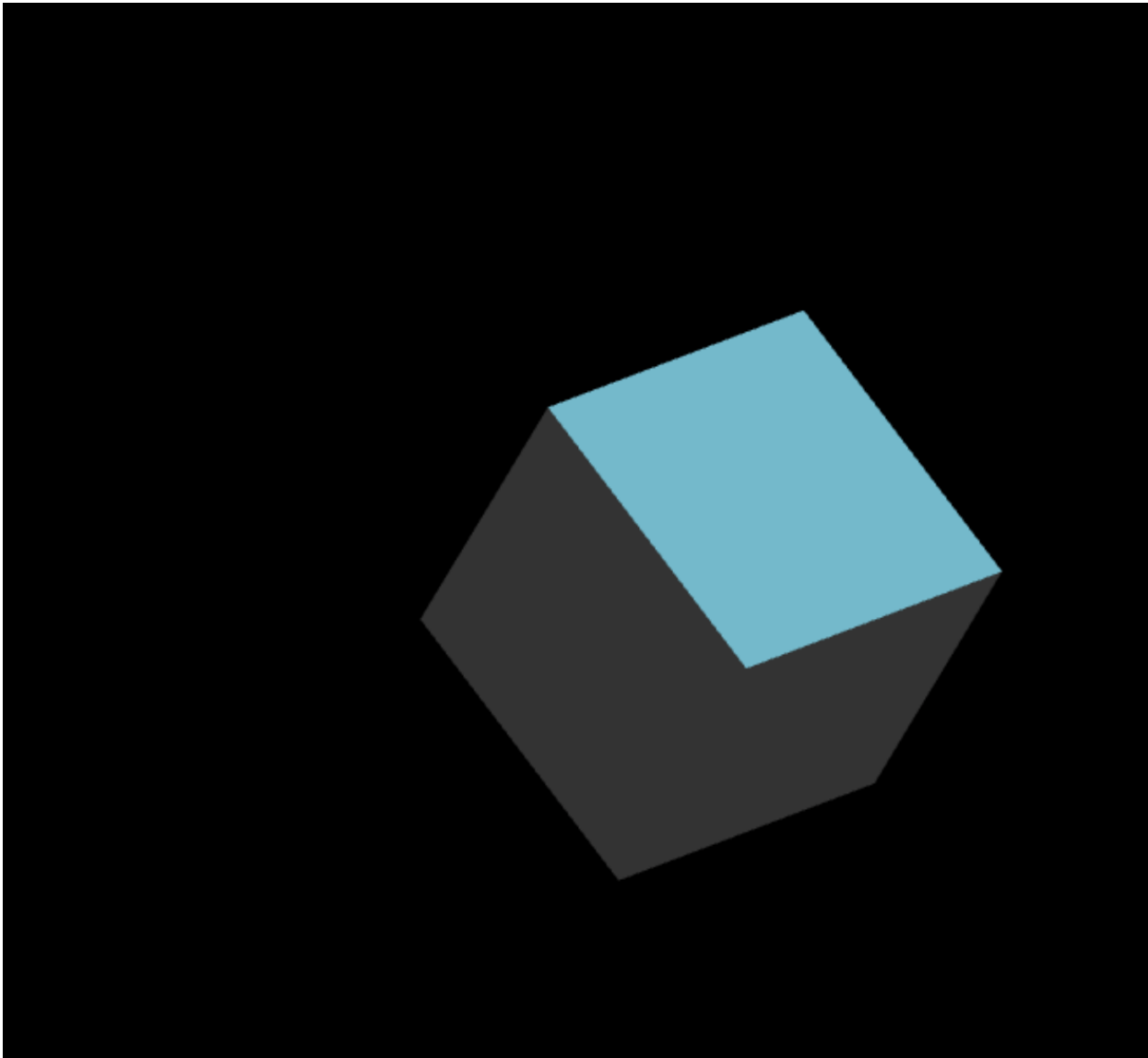
<https://chat.openai.com/share/c2d3970e-4c08-4059-aa8b-b971775a07dc>

Some images

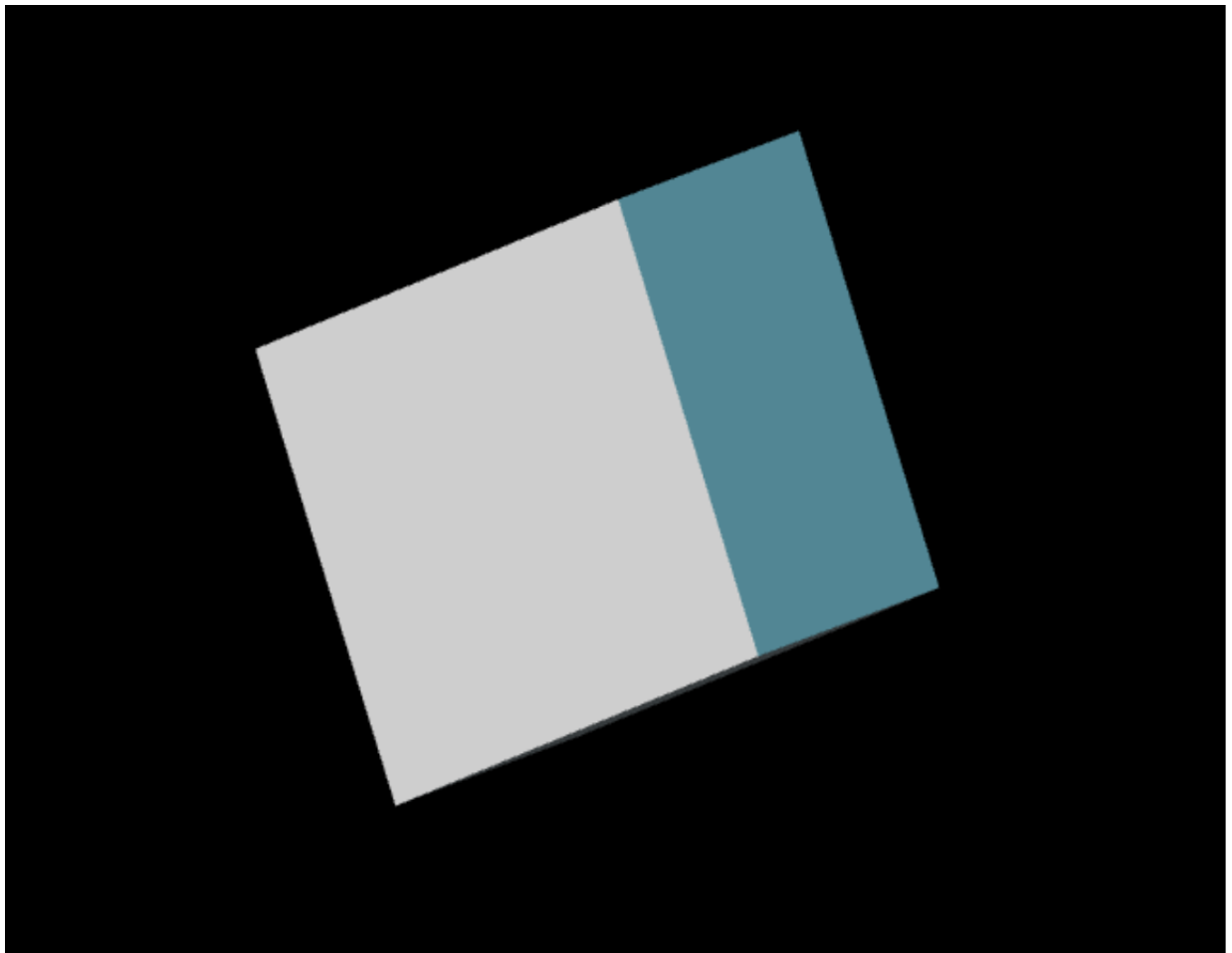








Returning back



Original form:

