Project 1 Report

The project consists of three different tasks.

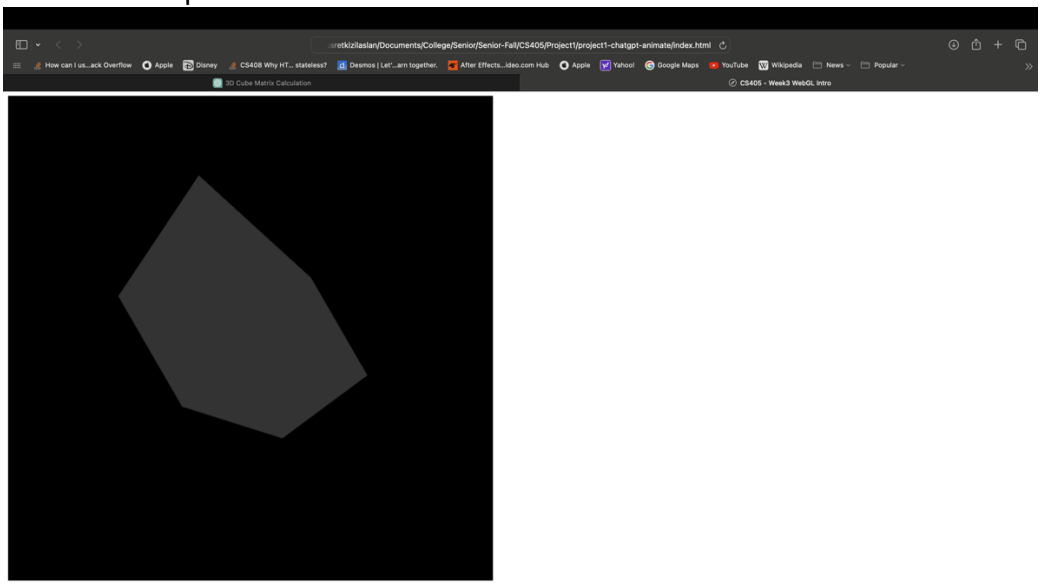
Task 1

First task was prompting ChatGPT to handle matrix transformations. I have prompted GPT3.5 with the given transformation-prompt.txt.

The matrix output is like below:

```javascript
const transformationMatrix = new Float32Array([
    0.433, -0.25, 0.25, 0.3,
    0.25, 0.433, -0.25, -0.125,
    -0.25, 0.25, 0.433, 0,
    0, 0, 0, 1
]);
```

The visual output is like below:



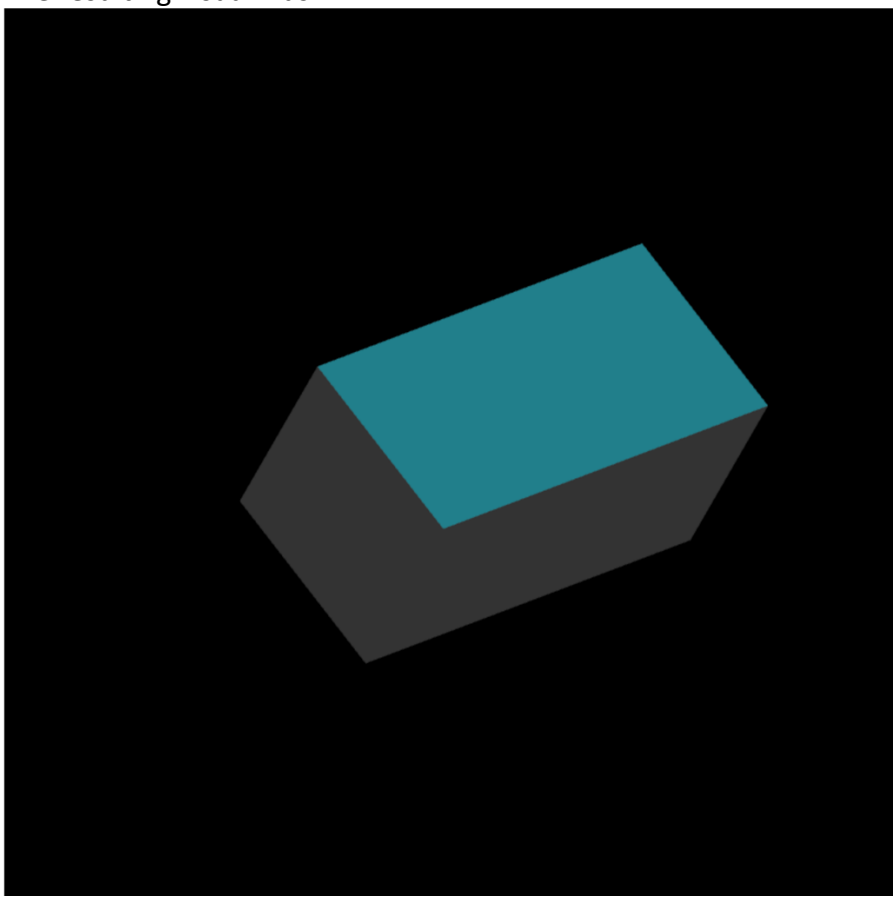
Task 2

Second task was to handle the matrix transformations by our own calculations. There were five transformations at total. Necessary function for transformations were provided. The transformations should be handled in order. First translation, then scaling, later x-y-z rotations.

The resulting matrix was:

0.1767766922712326, -0.2866116464138031, 0.7391989231109619, 0.12468592077493668
0.3061861991882324, 0.36959946155548096, 0.28033003211021423, -0.00054399791406468
-0.3535533845424652, 0.1767766922712326, 0.6123723983764648, -0.15026019513607025
0, 0, 0, 1

The resulting visual was:



The results of ChatGPT and mine are different. ChatGPT handles the transformations in wrong order, therefore the result of GPT is not correct.

Task 3

For this task, a transformation animation was generated. This animation was generated by ChatGPT. The animation translates the matrix from identity matrix to the matrix that I have calculated at Task 2. In first 5 seconds, the matrix smoothly transforms into target matrix. Then the transformed matrix translated into the initial matrix in 5 seconds. The code of this process is below:

```
/**
 *
 * @TASK3 Ask CHAT-GPT to animate the transformation calculated in
 * task2 infinitely with a period of 10 seconds.
 * First 5 seconds, the cube should transform from its initial
 * position to the target position.
 * The next 5 seconds, the cube should return to its initial position.
 *
 */


function getPeriodicMovement(startTime) {
    const elapsed = (Date.now() - startTime) / 1000; // Convert milliseconds to seconds
    const period = 10; // Total period of the animation in seconds

    // Calculate the phase of the animation (values between 0 and 1)
    const phase = (elapsed % period) / period;

    // Interpolate between the initial and target transformations based on the phase


    let targetMatrix = getModelViewMatrix();
    const initialMatrix = createIdentityMatrix();
    // Linear interpolation
    const interpolatedMatrix = new Float32Array(16);

    if (phase < 0.5) {
        for (let i = 0; i < 16; i++) {
            interpolatedMatrix[i] = initialMatrix[i] + (targetMatrix[i] - initialMatrix[i]) * (phase % 0.5) * 2;
        }
    }
    else {
        for (let i = 0; i < 16; i++) {
            interpolatedMatrix[i] = targetMatrix[i] + (initialMatrix[i] - targetMatrix[i]) * (phase % 0.5) * 2;
        }
    }

    return interpolatedMatrix;
}
```

The phase of the timelapse is calculated. If phase value is below 0.5, it means that the matrix transformation to the target matrix is not completed. If it is bigger than 0.5, then it is time to go back to the initial position. In for loops, points of the interpolatedMatrix is calculated for every iteration. The point values changes or every phase value. Therefore, the transformations appear smoothly.

The URL of the ChatGPT conversation is like below:
https://chat.openai.com/share/5e173fdb-aab7-4ae3-9624-92300b22876f