

CS405

2024-2025 FALL

Project 1 Report

CS405 Project 1: 3D Animations using ChatGPT

Merve Bilgi

29117

For this project, I worked on creating and animating a 3D cube object in WebGL using transformations. I used ChatGPT for tas 1 and 3 as it is described in the pdf. I used it to calculate the model-view matrix and apply it to a cube. Then, I animated the object using a set of transformations. I uploaded the work to GitHub as required.

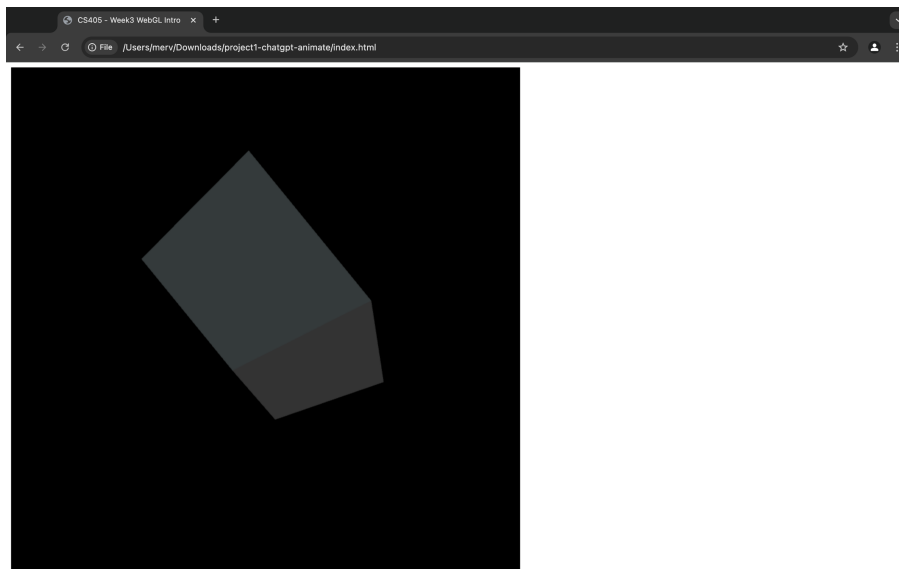
Github Link: <https://github.com/xmrvx/CS405-PROJECT1>

Task 1: Using ChatGPT for the Transformation Matrix

In the first task, I had to apply several transformations to a 3D cube. The transformations involved moving (translating) the cube, scaling it, and rotating it around different axes. Specifically, the transformations were:

- Translate: Move the cube by 0.3 units on the x-axis and -0.25 units on the y-axis.
- Scale: Scale the cube by 0.5 along the x and y axes.
- Rotate: Rotate the cube by 30 degrees around the x-axis, 45 degrees around the y-axis, and 60 degrees around the z-axis.

I used ChatGPT to calculate the final transformation matrix. I copied the prompt that given by you into ChatGPT, and it gave me the matrix in `Float32Array` format, which I then inserted into the `getChatGPTModelViewMatrix()` function inside `utils.js`. After running the code, the cube was rendered with all the transformations applied correctly.

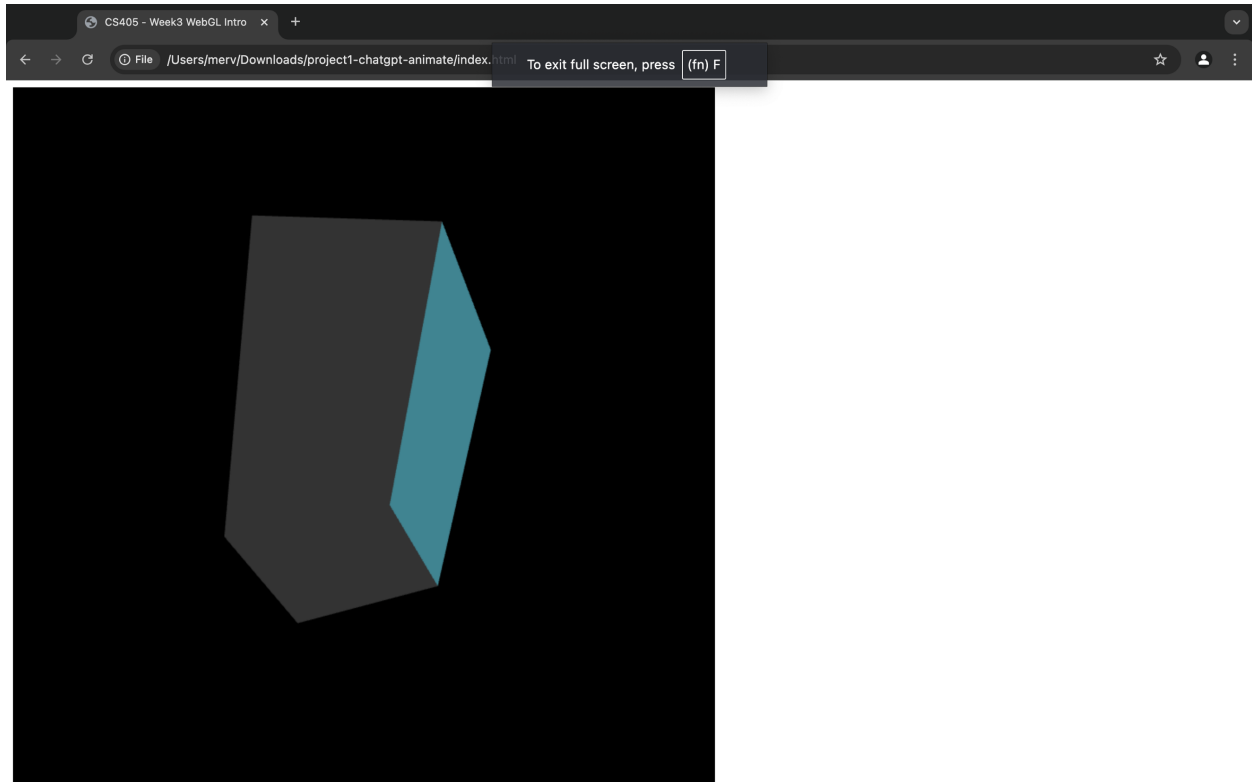


ChatGpt Link: <https://chatgpt.com/share/671292bd-ddc8-8009-9978-c60fb51caf20>

Task 2: Manually Creating the Transformation Matrix

In Task 2, I manually created the same transformation matrix by editing the `getModelViewMatrix()` function in `utils.js`. This method performed the same sequence of transformations as in Task 1, but I calculated it manually without ChatGPT. After running both versions, I noticed a difference in the cube's appearance.

Task 2 cube:



This difference could be because of a few things:

- **Floating-Point Precision:** WebGL or JavaScript might handle numbers a bit differently when calculating with ChatGPT versus doing it manually. These small differences in how numbers are processed can slightly change how the cube looks.
- **Transformation Order:** Another reason could be the order in which I applied the transformations. Even though the same moves (translation, scaling, and rotation) were used, doing them in a different order can make the cube look different.
- **WebGL Settings:** WebGL might also have some default settings that are applied automatically when rendering, which could be another reason why the cube looks slightly different between the two methods.

and both versions display the cube with the required modifications.

Task 3: Animating the Cube

For the final task, I needed to animate the cube using the transformations from Task 2. The cube had to move in a loop: transitioning to its transformed state in 5 seconds and then returning to its original position over the next 5 seconds, creating a continuous 10-second animation cycle.

To do this, I asked ChatGPT for help again. I used its response to modify the `getPeriodicMovement()` function in `utils.js`. After running the code, the cube moved smoothly in and out of its transformed state as expected.

Chatgpt link: <https://chatgpt.com/share/67129299-33c4-8009-beac-816597bdef4c>

