

## Computer Graphics Project Report

-Sai Durga Naresh Ponthangi

I created a webpage for my 3D rendering final project. The webpage includes a canvas element with an id of "myWebGLCanvas" that displays the 3D object I rendered. I added the ability to shade the object with either Gouraud or Phong shading, or display it in wireframe mode. To allow the user to choose the shading mode and shape of the object, I added two drop-down menus.

To handle the rendering and updating of the 3D object, I used several JavaScript files, including "renderer.js," "shapes.js," "MV.js," and "webgl-utils.js." These files allowed me to manage the rendering process and update the object on the canvas in real-time.

In addition to the shading options, I added input boxes for the vertex and fragment shaders. The user can enter their own code into these boxes, which allows for customization of the shading and rendering process. To manage user interface elements like color pickers and file input types, I utilized jQuery and jQuery UI.

Implementing these features can pose several challenges. Loading a file containing vertices for the object can be difficult as it requires parsing the file and converting it into the appropriate data structure for rendering. The rendering methods, including wireframe, phong shading, and gouraud shading, may require significant processing power, which could lead to performance issues on older hardware or slower devices. Additionally, implementing the various controls for color and position selection requires a deep understanding of WebGL and 3D graphics programming.

During the implementation, one may encounter bugs related to incorrect variable names or values, leading to unexpected behavior or errors. Proper testing and debugging are essential to ensure the webpage functions as intended. One lesson learned is the importance of modularity and organization when dealing with complex JavaScript files to make them easier to manage and debug. Another lesson is the significance of performance optimization for real-time 3D rendering.

Implementing this code has taught me several valuable lessons. Firstly, I gained a deeper understanding of 3D graphics programming and WebGL technology. I also learned the importance of testing and optimizing web applications to ensure that they work across different browsers and devices. Additionally, I learned the importance of proper input validation to prevent security vulnerabilities and unexpected behavior.

Managing multiple JavaScript files also taught me the importance of organizing code and using proper design patterns to make it more maintainable and easier to debug. Lastly, the use of

third-party libraries like jQuery and jQuery UI helped me understand the benefits of using pre-existing code to simplify the development process and improve user experience.

This is my working interface:

