Report

Problem Definition:

This program reads a .txt file containing vertex and face data, renders triangles using barycentric coordinates, and generates a ppm image file. The program interpolates colors across the triangles to produce a smooth gradient effect.

Description of Classes/Data Structures:

1. **Vertex**:
    - A structure to store the (x, y) coordinates of a vertex.
    - Example: Vertex { int x, y; }.
2. **Face**:
    - A structure to store a triangle defined by three vertex indices (v1, v2, v3) and the RGB colors for each vertex.
    - Example: Face { int v1, v2, v3; int colors[9]; }.
3. **Dynamic Arrays**:
    - The program uses dynamically allocated arrays to store vertices and faces.
    - Example: Vertex* vertices = new Vertex[numVertices];.

Methods and Steps:

1. **Reading the Input File**:
    - The readInputFile function reads the input .txt file and extracts:
        - Image size (width and height).
        - Vertex list (coordinates of the vertices).
        - Face list (triangles defined by vertex indices and their colors).
2. **Computing Barycentric Coordinates**:
    - The computeBarycentricCoordinates function calculates the barycentric coordinates (alpha, beta, gamma) for a pixel with respect to a triangle.

- These coordinates determine whether the pixel lies inside the triangle.

3. **Interpolating Colors**:
    - The interpolateColor function uses barycentric coordinates to interpolate the colors of the triangle's vertices and determine the pixel's color.
4. **Rendering Triangles**:
    - The renderTriangle function iterates over all pixels in the image and uses barycentric coordinates to determine which triangle covers each pixel.
    - If a pixel lies inside a triangle, its color is set using interpolated colors.
5. **Writing the Output Image**:
    - The writePPMFile function writes the rendered image to a .ppm file in the P3 format.
6. **Memory Management**:
    - The program dynamically allocates memory for vertices, faces, and the image.
    - Memory is freed at the end of the program to avoid memory leaks.