3D Object Transformation and Projection Using Homogeneous Coordinates in MATLAB

For this project, I created a MATLAB program that allows for scaling and rotation of a 3D object using homogeneous coordinates and basic transformations. Here's how I approached it:

1. **Reading the Data**:
   I started by reading in the vertex data from the house.txt file and the edge connectivity data from edges.txt. I also loaded the center point from the point.txt file. The vertex data provided the x, y, and z coordinates of each point, while the edges defined which vertices were connected to form the wireframe of the object.
2. **Converting to Homogeneous Coordinates**:
   I converted the vertices into homogeneous coordinates by adding an additional coordinate (1) to each vertex. This enabled the use of transformation matrices for the required scaling and rotation operations.
3. **User Input for Transformations**:
   The program then prompted the user for the scaling amounts along the x, y, and z axes, as well as the rotation angles about the x, y, and z axes. Additionally, the user was asked to input a z0 value, which represents the offset for the projection plane parallel to the x-y plane.
4. **Centering the Object**:
   Since the object could be centered around an arbitrary point, I first translated the object so that this center point was moved to the origin. This step was necessary to ensure correct scaling and rotation about the origin, and it involved subtracting the center point from each vertex.
5. **Applying Scaling and Rotation**:
   After translating the object to the origin, I applied the scaling and rotation transformations. The program first scaled the object along the specified axes and then rotated it about the x, y, and z axes in sequence (first x, then y, then z). Each transformation was applied using matrix multiplication with the appropriate transformation matrices.
6. **Projecting the Object**:
   After applying the transformations, I projected the object onto a plane parallel to the x-y plane, with an offset defined by the user's input for z0. This involved dividing the transformed vertices by their homogeneous coordinate and applying the projection matrix to visualize the object on the 2D plane.

7. **Plotting the Result**:
   Once the projection was complete, the program used MATLAB's plot function to display the object. Each edge of the wireframe was drawn by connecting the corresponding vertices as defined in the edges.txt file. I used the hold on command to plot multiple edges, and axis equal ensured that the aspect ratio was correct.
8. **Testing with Different Projections**:
   I used the program to generate various types of projections (right, isometric, perspective with one and two vanishing points, and oblique) of the house object by modifying the input parameters. The results were saved and included as outputs for submission.

This MATLAB program provides a flexible way to scale, rotate, and project 3D objects defined in text files, allowing for visualization from different angles and perspectives. The program works for any object as long as the vertex and edge data is provided in the same format, making it versatile for different models.