**Project Proposal**

**Leading Question**

Given a set of vertices and edges:

* Can the desired 3D mesh be represented in 2D space? (rendering algorithm)
* Can the shortest path be found from point a to point b? (pathfinding)
* What is the minimum number of colors one can use to color a mesh? (graph coloring)

**Dataset Acquisition and Processing**

Datasets will be simple to come by since they are 3D models. Blender3D will be used to import files in any supported format, and export them as a wavefront (.obj) file format. From there, the only data collected from the file will be vertex locations in 3D space, normals in 3D space, edges and their corresponding vertices, and potentially color data.

Once data is collected from the OBJ file, the vertices will be mapped to 2D space using Isometric Projection (<https://en.wikipedia.org/wiki/Isometric_projection>). This can be done in Python quite easily.

Should any errors occur in the models, they can be cleaned using Blender 3D.

**Graph Algorithms**

**Data Algorithms**

Rendering - Develop an algorithm that takes in a set of 3D points, maps them to a 2D space, and then plots them in a PNG image. Then, using the edges and faces information, draw the edges and color the faces.

Graph coloring - Develop an algorithm that colors the vertices of the mesh using a minimum number of colors. Input will be a set of 2D points, and the output will be a PNG image with the vertices colored appropriately.

**Traversal Algorithms**

The traversal algorithm will be A\*, a greedy BFS algorithm. The heuristic chosen will be linear distance to the desired point. Input will be a set of 2D points and the output will be a PNG image with the appropriate edges colored representing the path chosen by the algorithm.

**Timeline**

Week 1: Data collection and process

Each team member will collect at least two 3D models of their choice with no duplicates between the team. Once collected, the models will be processed using Blender 3D and Python.

Week 2: Rendering and graph coloring

Team members will tackle the development of the rendering and graph coloring algorithms. The first issue will be to address the rendering of all the vertices and edges in 2D space in a PNG. After that, tasks can be split between rendering faces and a graph coloring algorithm.

Week 3: A\* Development and Testing

The A\* algorithm will be implemented once the overall structure of the graph is completed. Team members will also write test cases to thoroughly test the code in various aspects. Past tests will be improved upon and cleaned.

Week 4: Off (Fall break)

Week 5: Zazz

Polish off any components and add face lighting if time is available.