**Assignment 2 Report**

Nirmal Chaudhari

Sathurshan Arulmohan

Mithun Paramathasan

# Discovering the Project

1. Describe the technical debt associated with the starter code (generator and visualizer).

Generator:

* In the DotGen.java, the Vertex generator was creating 4 vertices for every coordinate in the canvas. This caused vertices to overlap and cover each other. When it came to generating the colours for the segment, this acted as a technical debt since it was uncertain which vertex would show up on top. Additionally, when it came to generating the segments, the vertices overlapping created unnecessary segments and noise to be created within the segment lists.
* Also, the DotGen.java was creating another vertex list just to attribute colors to each vertex element. This was a significant technical debt since it was a waste in memory space, and created unnecessary new lists that would affect with the code readability.

Visualizer:

* In the GraphicsRenderer.java, the Vertex visualizer was treating the vertex data structure as a list which isn’t really necessary since we aren’t using any list features. Instead, it would be better to represent it as a HashSet since index aren’t necessary. This limits the scope of available methods, and makes the data structure more focused.

1. Reflect on the difficulties in controlling the evolution of both parts if staying at the immutable data structure level.

The problem with the immutable data structure is that it is memory costly to make changes to the data structure. For example, the given code had to create a new set of vertices just to add color properties. This means that in future development, we would need to know all the properties before making vertices and segments, or we would need to re-create the whole set with the new properties. This makes development difficult as it is not always the case that we would know all the properties of vertices and segments before creating them, and we would want to reduce memory cost. The design of the data structure should be mutable so that it allows the developer to make changes in the future. Later in the project, we would need to change the segments to create different polygons. With the current immutable structure, we would need to re-create the whole mesh instead of editing a small part of it.