This steps are mainly done manually, although eventually some automated process may be implemented. The main suggestion is that after generating a new tissue, always use the `GeneratedTissueVisualization.ipynb` to check if everything is okay.

When generating external boundaries

- The method we use to generate boundaries through Voronoi tesselations assumes a vertex at infinity (labelled as -1). We can exclude this vertex from the calculations but it leads to some large spikes in the external boundary. In theory it is possible to regularize the vertices of the external boundary by applying a energy minimization to their associated areas prior to the simulation of the form

  $$E = \sum_B K(A - A_0)^2$$

  But I haven't implemented this yet. Thus far, the best way is to manually change the vertices positions.

- There is also the issue that sometimes, vertices in the external boundary are not recognized as such, and vertices in the bulk are included in the boundary. It only happens to a small enough number of vertices, that we just have to manually include or remove the faulty choices in `boundaries.txt`.

Following theses steps would ensure that all appropriate vertices are at the boundary and are near equilibrium.

When generating internal boundaries

There are a couple of issues with the internal boundaries to be addressed.

In principle, there should be no issue with having vertices with only two neighbours (associated with a single cell); But given our implementation of topological rearrangements, which are a core part of the closure, we require all vertices to have two neighbours. To deal with this:

- Delete edges (can't leave blank edges on `ridges.txt` file)
- Delete vertices from the cell regions in the `centers.txt` file and from the wound boundary in `boundaries.txt`
- Leave blank cells (for removed cells to give way to the wound).
- Leave blank vertices (we can't delete the vertices from the `vertices.txt` file)
- Add new edges for remaining vertices.

There should be a way to automate this, but I haven't implemented it yet, so we just do it manually. It does take a bit of work. This issue is also not present for wounds of 1 and 2 cell-sizes.