chaos Create your world.

Mesh file task

You are given some meshes in JSON format:

```
{
    "geometry_object": {
      "vertices": [
            7.695038, 0.07024442, -0.1745732,
            7.721254, 0.2733468, -0.2023404,
            ......
      ],

      "triangles": [
            0,1,2,0,3,1,....
      ]
    }
}
```

The mesh is represented by two arrays:

- Vertices - each 3 floats represent one vertex in space
- Triangles - each 3 integers represent the indices of vertices of a triangle. In the example above triangle 1:
    - Vertices - 0, 1 and 2
    - Edges - e0(v[1]-v[0]) and e1(v[2]-v[0])
    - Geometric normal can be calculated by the cross product of e0 and e1

Mandatory tasks:

1. Parse the given meshes in JSON format
2. Calculate smooth vertex normals. https://en.wikipedia.org/wiki/Vertex_normal
3. Calculate some mesh statistics. The calculations here must be carried out using **all** available threads
    a. Area of the smallest (non zero) triangle
    b. Area of the biggest triangle
    c. Average triangle area

Optional tasks:

- Simple graphical user interface - a window with some buttons to load JSON file, execute the calculations and some labels with the calculated results.
- Determine if the given mesh is closed. Let's say that closed means that each edge has at least two neighbors.
- Generate a new mesh by subdividing each triangle into 4 smaller ones
- Pick some random point in space. Determine if it is inside the given mesh.

Some general requirements and recommendations:

- C++
- JSON parsing can be done with external library such as RapidJSON, TinyJSON, JsonCPP. Qt JSON, nlohmannJson.
- GUI can use libraries such as
    - Qt
    - wxWidgets
    - ImGUI
- Geometry related tasks should be written in code.
- Solutions's performance is a priority
- Multiplatform code is a bonus
- Clear, concise and reasonably commented code is a bonus
- If the project uses any external libs - their versions should be noted. Ideally their source should be publicly available.
- Ideally some solution files should be provided - Visual Studio, CMake or something similar