Grading Structure of the "Second" Assignment (Modeling)

Open World RPG Game Concept

You will be given several tasks(quests) every week. They are divided into two main categories, "***main***" and "***optional***". The main type is what you need to proceed onto the (some of the) tasks of the following weeks, and optional type is some "side quests".

Each task has a maximum mark. If you try it, you will be given a mark ranging from 0 to that maximum mark.

Final Boss

At Week 12, we will announce several BIG tasks options, and you can choose one (and only one) to do. Of course, this task will have more marks comparing to previous tasks.

Final Grading

We will gather and grade your submissions on IVLE, but we will arrange a time to meet you personally during the reading week also. To discuss and understand what you have done.

About Optional Tasks

For main tasks, we expect you to try all of them. However, for optional tasks, we expect you only to try %80 of them. For example, if there are a total of 200 marks for all the optional quests (we are still adjusting that), we pick the optional questions from you (or you can choose them by yourself) the optional tasks you did that the maximum mark worth 160.

For example, if we have 5 optional tasks with 40 marks each. And a student did all of them and the marks are 20, 40, 35, 40, 30 respectively. Then his mark for optional task will be 145 out of 160 (ignoring the 20).

We will have a list of all main and optional tasks at the end of the semester for you to check.

Lab 2 Tasks

Implement `enext()`, `sym()` (20 marks, main)

As mentioned in the lecture notes, they could be separate functions outside of the class.

Implement `org()`, `dest()` (20 marks, main)

As mentioned in the lecture notes, they should be member functions of the class.

Implement `fnext()` (80 marks, main)

After reading an OBJ file, link up all the triangles by setting up the `fnlist`. In order to prove your implementation is right, you could implement one of the followings to show that your code works.

Compute the Number of Components (20 marks, optional)

An OBJ file may contain a few pieces of surfaces, and the number of pieces is the same as the definition of the number of components in a graph. You may simply achieve this by BFS or DFS like a graph. However, by being in the same component we only mean that their triangles are connected by `fnlist`. Namely, two triangles can be in two different components if they intersect each other but they are not connected through some vertices and edges.

Implement `orientTriangles()` (20 marks, optional)

Two triangles are in the same orientation if their version 0 normal vectors are in the same side of the surface by the right hand rule. If they are not in the same orientation, you can rectify that by "flipping" one of the triangle. By flipping a triangle, it can be done by exchanging two entries of the vertex indices in the `flist` (and the corresponding entries in the `fnlist`). You can orient all the triangles in the same surface by a BFS/DFS traversal with flipping.

However, there is a possibility that meshes are NOT orientable, namely, you cannot flip all the triangles to face the same side of the surface. The simplest example is the Möbius Strip. For such cases, your `orientTriangles()` will return FALSE and leave the mesh "unoriented".

Compute Vertex Normal Vectors for Smooth Shading (10 marks, optional)

OpenGL can render the Gouraud and Phong shadings by averaged vertex normal vectors. Create a new vector array inside your class and name it `vnlist`. It stores all the vertex normal vectors that is the average of those of surrounding triangles. And add a keyboard function in your `main.cpp` so that you can choose to toggle flat or smooth shadings.

Visualize boundary edges (10 marks, optional)

Color the edges shared by only exactly one triangle by red.

Implementing Selection of Triangle by User Marquee (20 marks, optional)

Select a portion of your mesh by dragging a rectangle in your windows. Set up a bit vector (or bitset) `tselect` in your class to record the selected triangles.