Lab2 Half-edge Data Structure

Description

You are supposed to finish the edge collapse algorithm with the half-edge data structure.

Framework

We provide the environment configuration on the Visual Studio 2022(v143) on DEBUG x64 pattern

As for Linux users, please help yourself.

- geometry
  - half-edge.h(.cpp)

    the half-edge data structure definition
  - Mesh.h(.cpp)

    encapsulate the Mesh structure
  - Model.h(.cpp)

    encapsulate the Model structure   You are supposed to finish the functions(`collapseEdge` `randomCollapse` `fromMesh`) under this class
- imgui

  the library files of imgui
- manager
  - InputManager.h(.cpp)

    encapsulate some mouse/keyboard event and action solution
  - Key.h

    the Macro-Def of event and action
- utils
  - Camera.h(.cpp)

    encapsulate the moving of camera
  - glad.c

    external library GLAD
  - GUI.h

    manage the GUI widget on the left top of the window
  - global.h

    define some global variable
  - Utils.h(.cpp)

    define some callback and commonly-used functions
  - Shader.h(.cpp)

    encapsulate the Shader structure

- main.cpp
  the entry of the procedure, providing the mainloop for rendering

Instruction

In this lab, you should

- implement two functions in `src/geometry/Model.cpp`. The places where you should fill your code in are marked with `\\TODO:`
  - In function `randomCollapse`(callback function). You need to implement an algorithm that reduces the number of faces in certain order, ensuring that each call to the function itself reduces the number of face and and calls the function `collapseEdge` to truly collapse the face.
  - In function `collapseEdge`, you should implement the algorithm to truly collapse the edge.
  - In function `fromMesh`, you should fill in the `opposite` attribute

We provide the visualization of the loaded model.

Attention! Adding some extra members in the `Model` class is admitted

Submission

You should submit your **code** along with **a report**.

Please **clarify the special condition you meet(including the specific, half-edge .etc. Better for drawing a diagram)** in your report.

Please submit on canvas before deadline.

Pack your program so that it can run on **BARE computers**, which means you should provide glfw & glad & glm environment and .sln or CMakeLists files.

In this lab, windows users may not bother to provide the environment. But this **should be a must** for linux/macos users.

Providing an executable is strongly recommended.

Plagiarism will not be tolerated.