# CS3242 3D Modelling & Animation

## Assignment 1 Submission Report


- Name: Yang Shiyuan
- Matrix No: A0214269A


### Instructions for application testing

1. Unzip the zip file `A0214269A_Assignment1.zip`.
2. Navigate to the folder `./x64/Release/`.
3. Locate the application `Lab 01.exe` file.
4. Once the application is loaded, key in `cat.obj` to load the model.
5. In the main window:
    1. Press `S` to toggle the smooth shading option.
    2. Press `L` to perform loop subdivision. There are 2 variants of loop subdivision to choose from:
        1. Press `1` before `L` to perform loop subdivision with Warran Beta Version.
        2. Press `2` before `L` to perform loop subdivision with Origin Beta Version.
    3. Press `C` to toggle on coloring for different components.
    4. Press `R` to read in another model.
        1. To read a model with ease, it is best to place the model in the same folder with `Lab 01.exe` application.
            1. Currently, only `.obj` and `.off` files can be loaded.
        2. If the above method is employed, you can simply enter the `filename` in the application in order to load the model.
    5. Press `B` to perform barycentric subdivision.
    6. Press `E` to toggle on the edges which are the boundary edges of the model.
    7. Press `W` to toggle on wireframe.
    8. Press `P` to toggle on the model in its polygon form.
    9. Press `O` to write the current model on the screen to a file.
        1. When specifying the filename, please remember to include the file extension. Currently only `.obj` and `.off` files can be saved.
        2. The newly written file will be located in the same folder as where the `Lab 01.exe` is located.
    10. Press `V` to toggle on the model in its vertex form.
    11. Press `Q` to exit the application.
6. Several models have been loaded in the same folder as the `Lab 01.exe` application. They are ready for your usage.
    1. `cat.obj`
    2. `pikachu.obj`
    3. `deer.obj`
    4. `teapot.obj`
    5. `teddy.obj`
    6. `cube.off`
    7. `cubes.off`

## Instructions for project development

1. Unzip the zip file `A0214269A_Assignment1.zip`.
2. Double click on `Lab 01.sln` to open the Visual Studio Solution.
    1. Please ensure that you have the latest version of Visual Studio before loading up the `Lab 01.sln` file.
3. Press `Ctrl + Alt + L` to open the solution explorer to explore the project.
4. To test, press the play button `Local Windows Debugger`. Ensure that the following settings are configured as such:
    1. Configuration: `Release`
    2. Platform `x86`
    3. ▪ ☑ Build Option
5. **Please ensure that you have the `eigen` C++ library installed in your local machine. This library is required for the project to run**
    1. Refer to the following link to install `eigen` C++ library in your local machine:
        1. https://www.youtube.com/watch?v=6mMjv-tA5Jk&t=270s&ab_channel=FromScratch

## Task Completion

| S/N | Task Details | Nature | Implementation |
|---|---|---|---|
| 1 | Computing Normal Vectors | Main | `operationLib.cpp` -> `generateFaceNormals()` |
| 2 | Computing Angle Statistics | Main | `operationLib.cpp` -> `generateStatistics()` |
| 3 | Write an OBJ File | Main | `main.cpp` -> `writeFile()` |
| 4 | Read some other type of file other than OBJ | Optional | `main.cpp` -> `readOffFile()` |
| 5 | Implement enext(), sym() | Main | `mesh.cpp` -> `enext()` & `sym()` |
| 6 | Implement org(), dest() | Main | `mesh.cpp` -> `org()` & `dest()` |
| 7 | Implement fnext() | Main | `mesh.cpp` -> `setupAdjList()` |
| 8 | Compute the Number of Components | Optional | `mesh.cpp` -> `processNumOfComponents()` |
| 9 | Implement orientTriangles() | Optional | `mesh.cpp` -> `orientTriangles()` |
| 10 | Compute Vertex Normal Vectors for Smooth Shading | Optional | `mesh.cpp` -> `generateVertexNormals()` |
| 11 | Visualize boundary edges | Optional | `mesh.cpp` -> `drawEdge()` |

## Boss Conquest

**Theme:** `Subdivision`

| S/N | Task Details | Status | Implementation |
|---|---|---|---|

| S/N | Task Details | Status | Implementation |
|---|---|---|---|
| 1 | Barycentric Subdivision | Completed | `subdivisionLoop.cpp` & `mesh.cpp` -> `barycentricSubdivide()` |
| 2 | Loop Subdivision | Failed | `subdivisionLoop.cpp` & `mesh.cpp` -> `loopSubdivide()` |

**Boss Update:**

- Couldn't finish loop subdivision as I could not trace the error that causes the new vertex created during the subdivision to be outside the mesh.
- Hence any new object created after the loop subdivision will appear like a `forzen` creature. 😩

## Reference

- Barycentric Subdivision Reference: https://ncatlab.org/nlab/show/subdivision
- Loop Subdivision Reference: http://www.cs.cmu.edu/afs/cs/academic/class/15462-s14/www/lec_slides/Subdivision.pdf
- Off File Format: https://en.wikipedia.org/wiki/OFF_(file_format)

## Reflection

- I feel that this assignment is probably the most challenging one I have ever done. There was a lot of difficulty in the implementation of the `fnext` list as there are many steps leading up to the creation of the list, especially when we need to make sense of the raw vertices and faces, then to place them in the correct order.
- Though the assignment is difficult to implement, I feel that the most interesting part is to see our work in action visually.