CSC4140 Assignment V

Computer Graphics

April 5, 2022

Geometry

Student ID: 119010434

Student Name: Zhang Qihang

This assignment represents my own work in accordance with University regulations.

Signature:

1 Overview

I have implemented the calculation of Bezier curves and surfaces in this assignment. I also implemented operations on meshes by means of halfedge data structures. These operations include flip curves, split curves, and upsample a graph. The most interesting part of this assignment is the learning of the halfedge data structure, which is very powerful and intuitive, and can be very effective in solving graph manipulation problems after understanding the operations on halfedge, edge, vertex, and face.

2 Task1

De Casteljau's algorithm is a recursive method to evaluate polynomials in Bézier curves. The positions of the points at each moment need to be calculated separately, resulting in a curve.

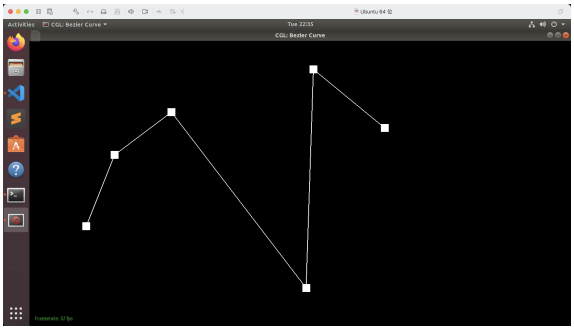

Figure 1: Bezier curve with 6 points

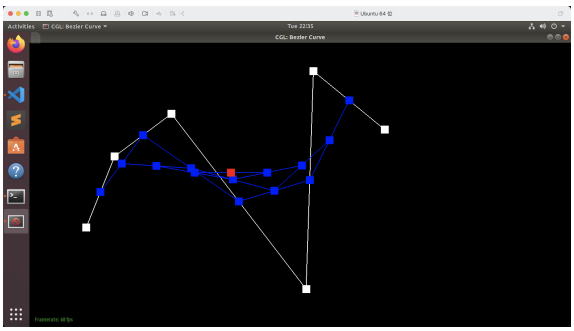

Figure 2: Bezier curve with 6 points step through

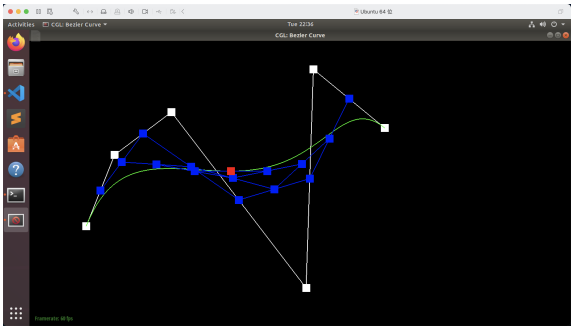

Figure 3: Bezier curve with 6 points with curve

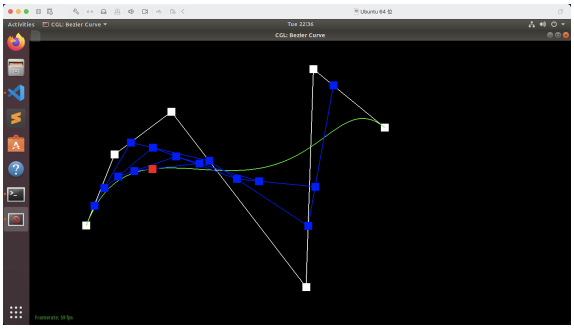

Figure 4: Bezier curve with 6 points with curve moving

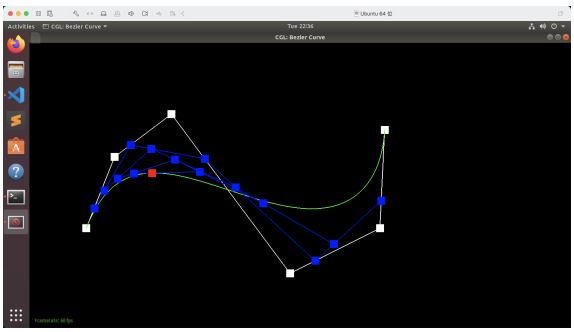

Figure 5: Bezier curve with 6 points with control point moving

3 Task2

A surface can be obtained by joining multiple curves, but when forming a Bessel surface, it is necessary to use all the corresponding points of all Bessel curves at moment t as all control points of a Bessel curve to form a Bessel curve, and eventually, a Bessel surface will be formed on the sum of all the curves corresponding to t.

3.1 Implementation

The points at uv need to get all the corresponding Bessel control points at u first, while these points can form a Bessel curve, and then get the final desired point by v. Evaluate1D is able to find the Bessel curve formed by a group of points corresponding to the points at moment t.

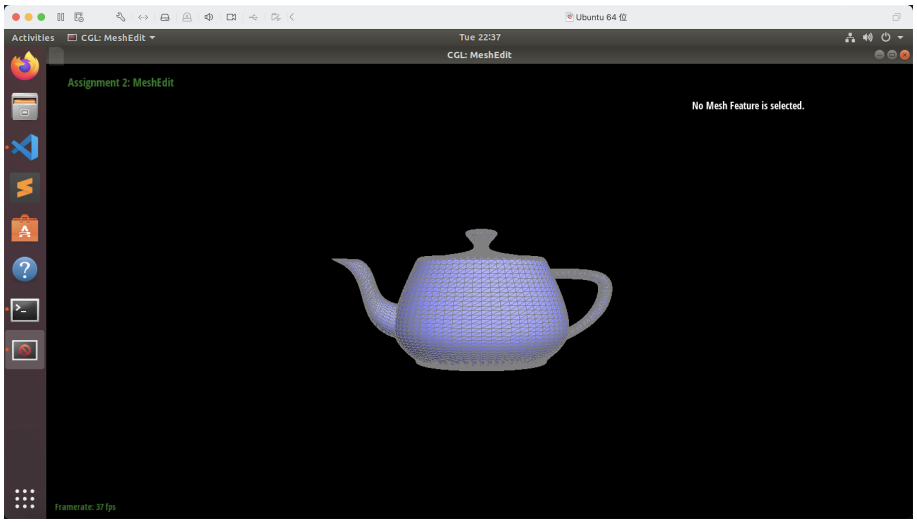

Figure 6: /teapot.bez

4 Task3

First, the area can be calculated quickly by vector cross method, and all the faces near a vertex can be traversed sequentially according to the twin and next functions of the halfedge data structure. According to this method, the area-weighted normal of the vertex is obtained.

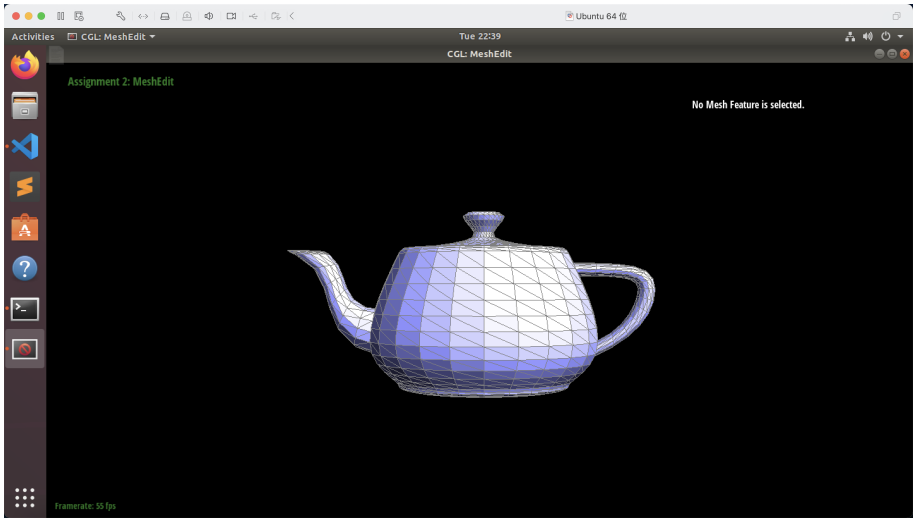

Figure 7: teapot.dae flat shading without vertex normals

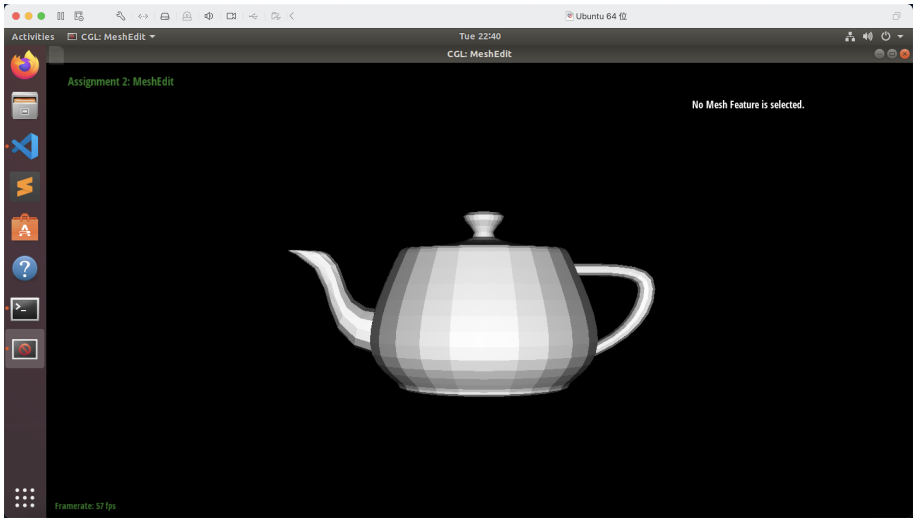

Figure 8: teapot.dae Phong shading without vertex normals

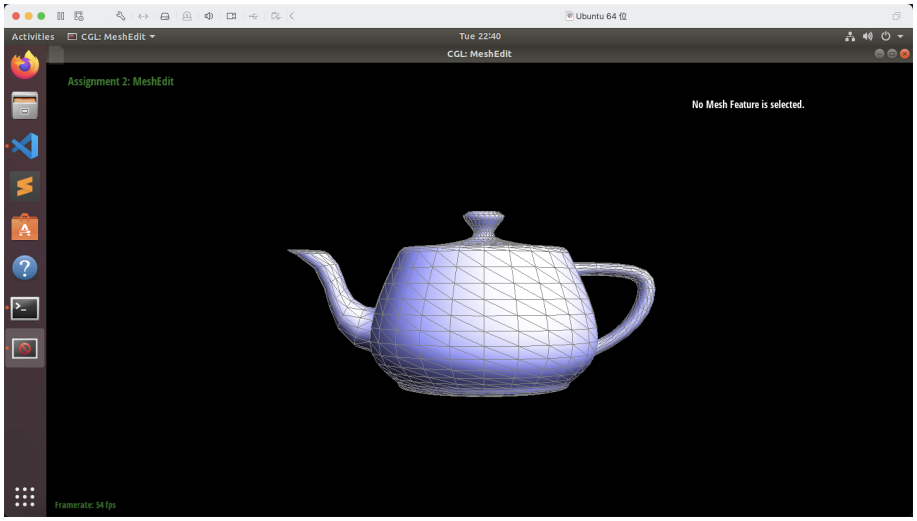

Figure 9: teapot.dae flat shading with vertex normals

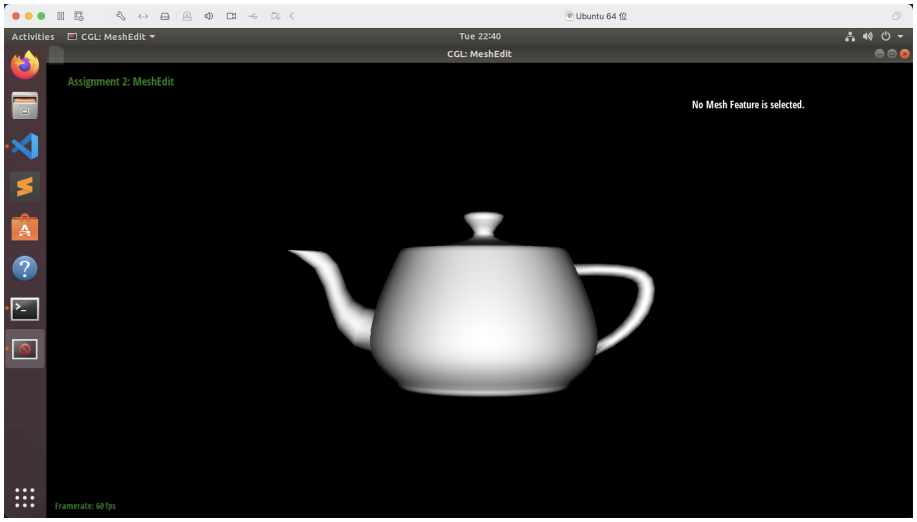

Figure 10: teapot.dae Phong shading with vertex normals

5 Task4

I use picture to explain how to implemented edge flip operation.

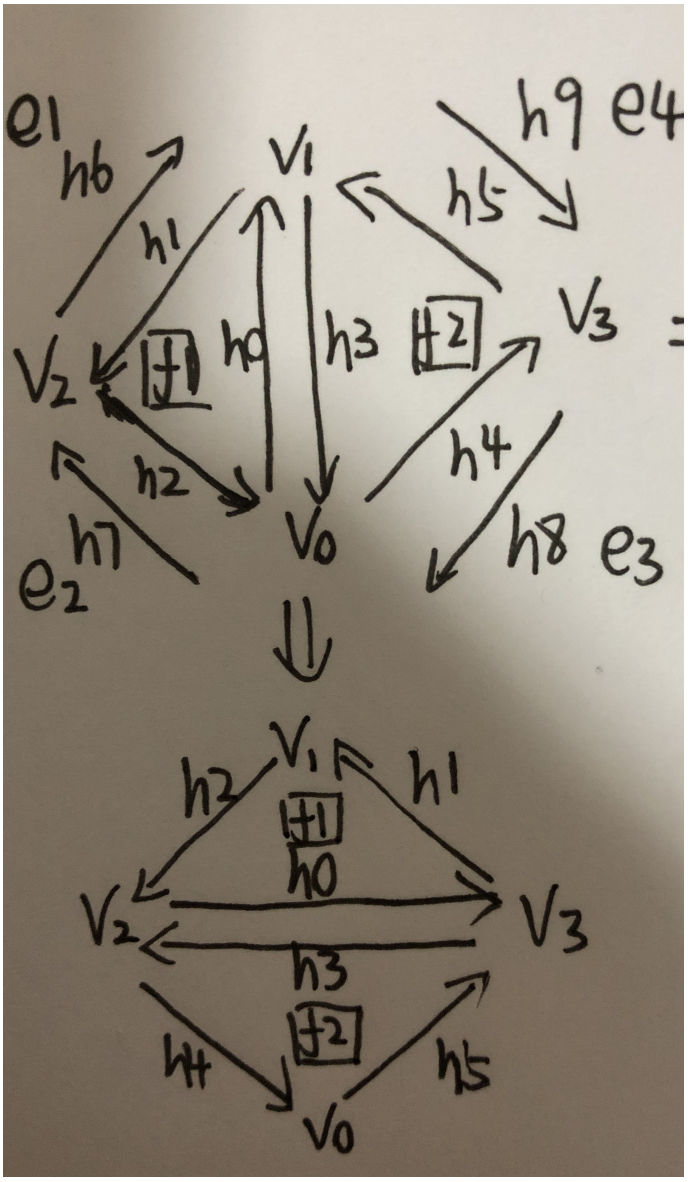

Figure 11: How to Edge Flip

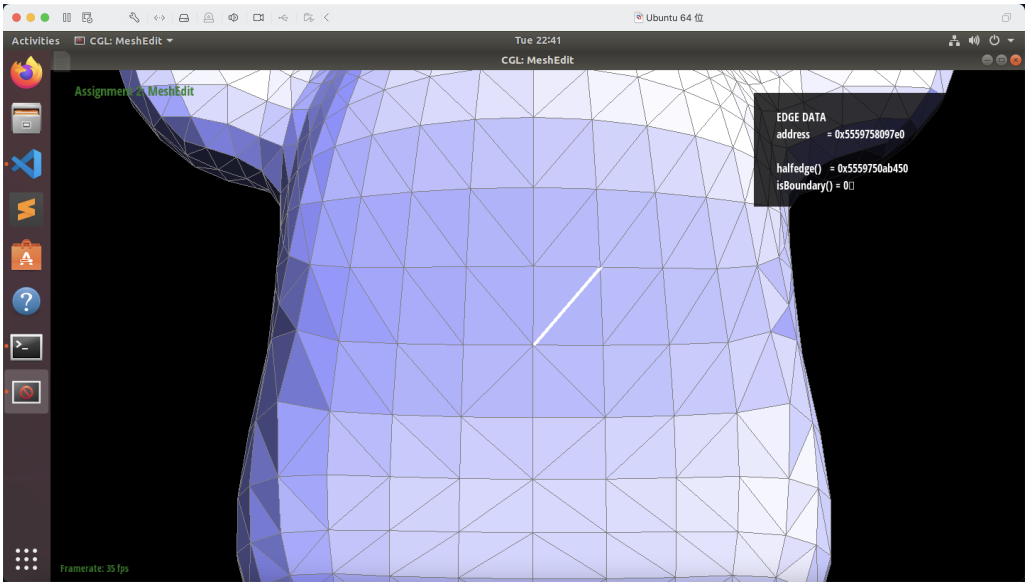

Figure 12: Before Edge Flip

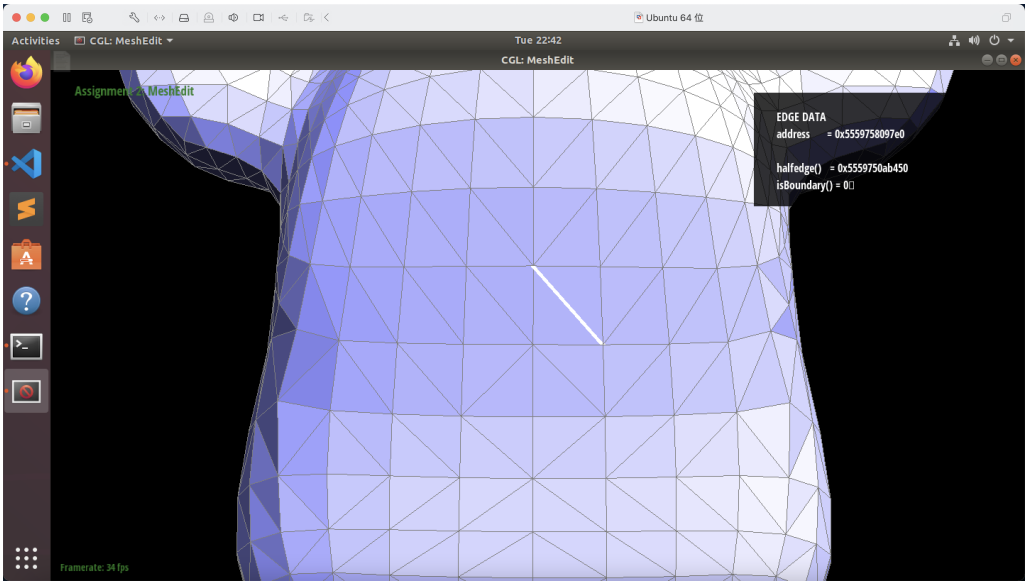

Figure 13: After Edge Flip

6 Task5

I use picture to explain how to implemented edge split operation.

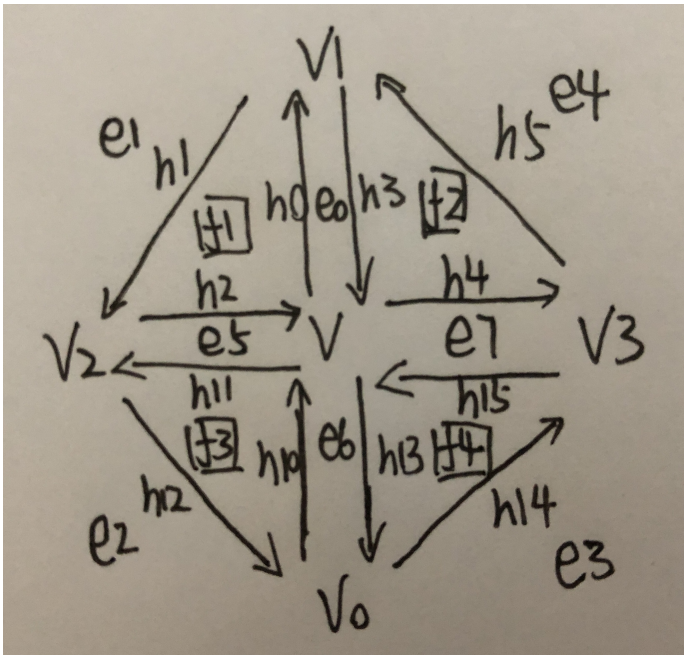

Figure 14: How to Edge Split

6.1 Extra

I use picture to explain how to implemented edge split operation with boundary face.

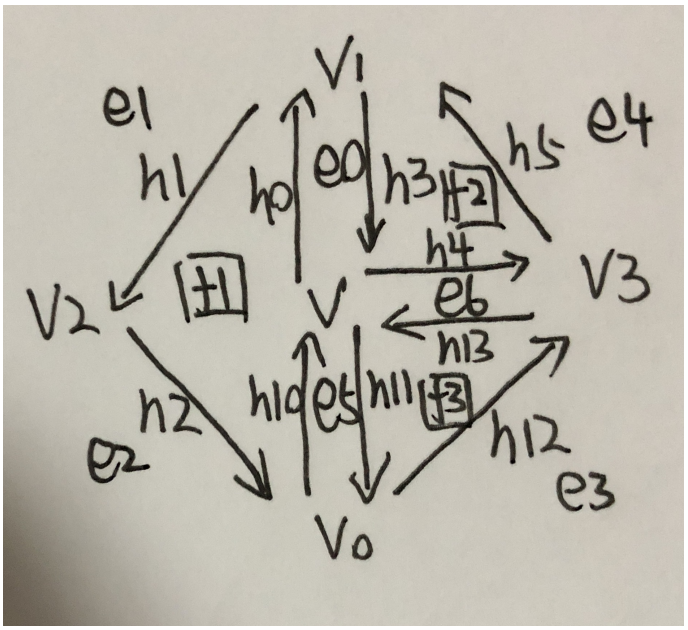

Figure 15: How to Edge Split with Boundary

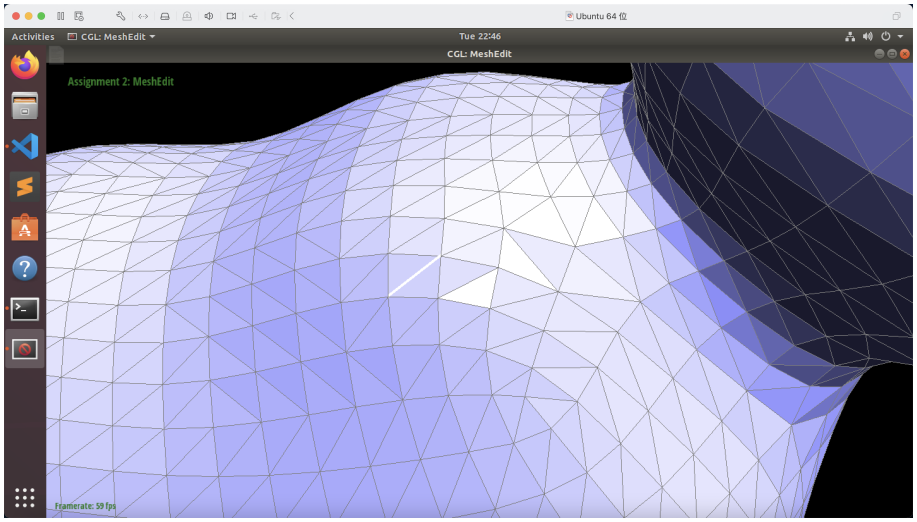

Figure 16: Cow before Split

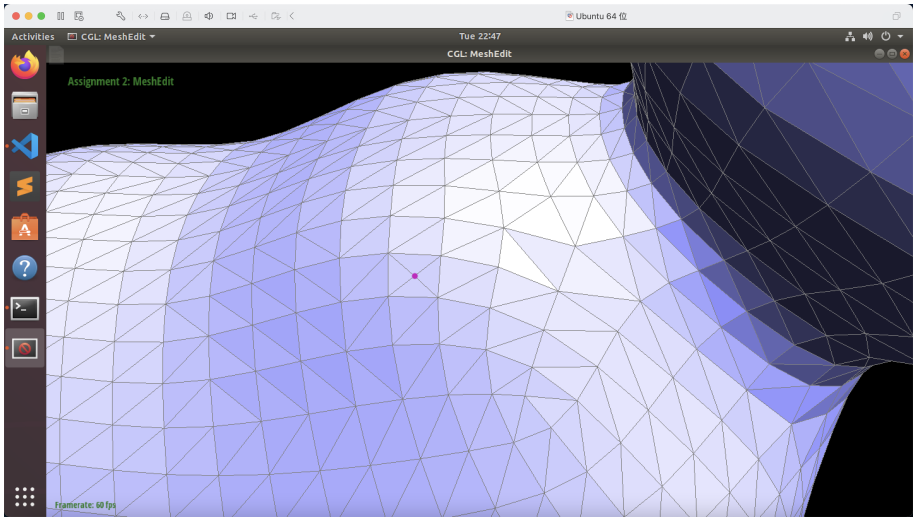

Figure 17: Cow after Split

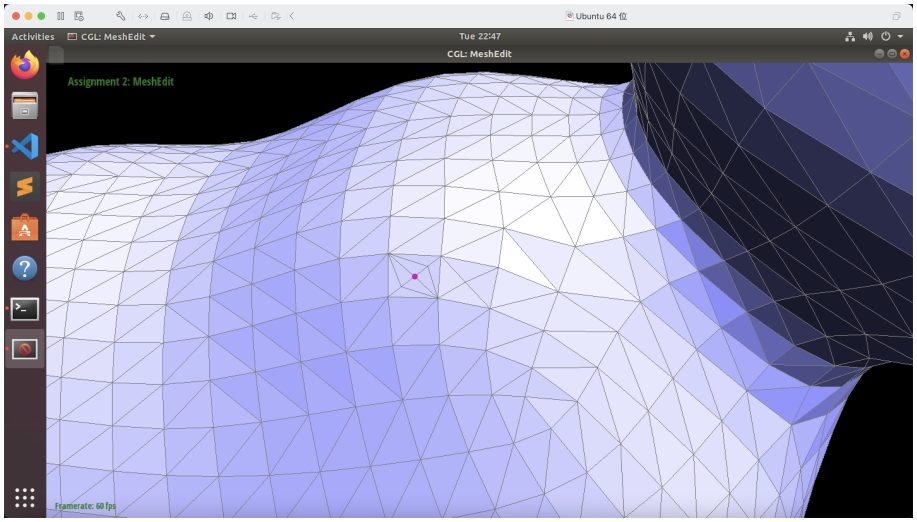

Figure 18: Cow after some Split and Flip

7 Task6

First: Compute the position of every new added vertex using 3/8*(A+B)+1/8*(C+D).

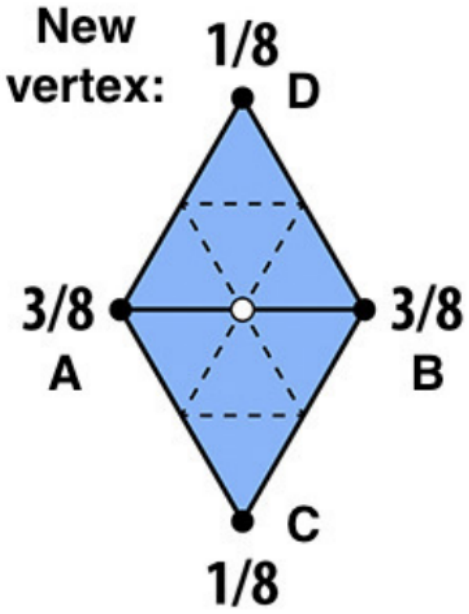

Figure 19: New Vertex

Second: Update the position of every existing vertex using (1 - n * u) * original$_position$ + $u * original_neighbor_position_sum$.

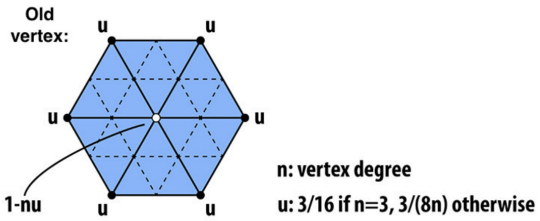

Figure 20: Update Vertex

Third: Split all old edge to more edges

Fourth: Flip new added edge which only connected to one new vertex.

7.1 After loop subdivision

Sharp corners become smooth but sunken.

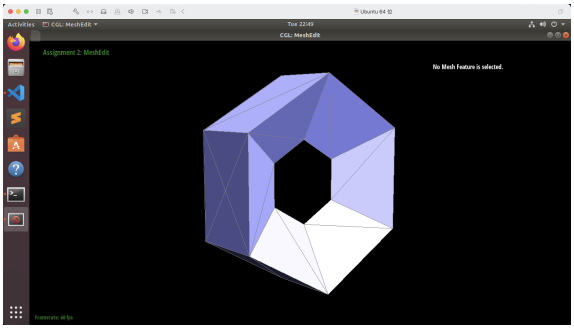

Figure 21: loop Subdivsion

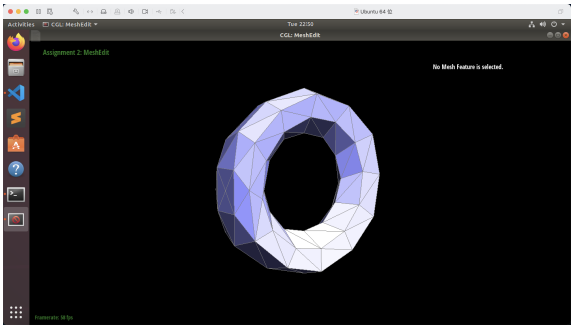

Figure 22: loop Subdivsion

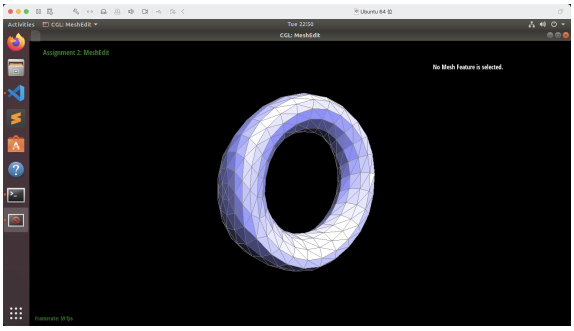

Figure 23: loop Subdivsion

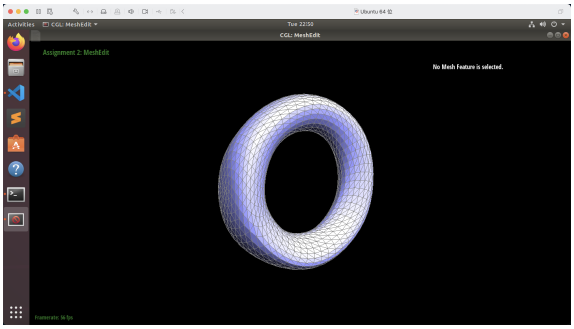

Figure 24: loop Subdivsion

We can reduce this effect by pre-splitting some edges.

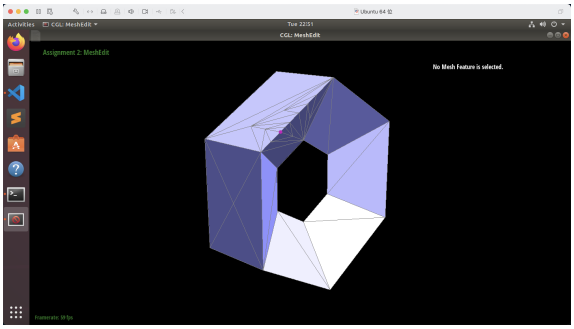

Figure 25: loop Subdivision with pre-spliting

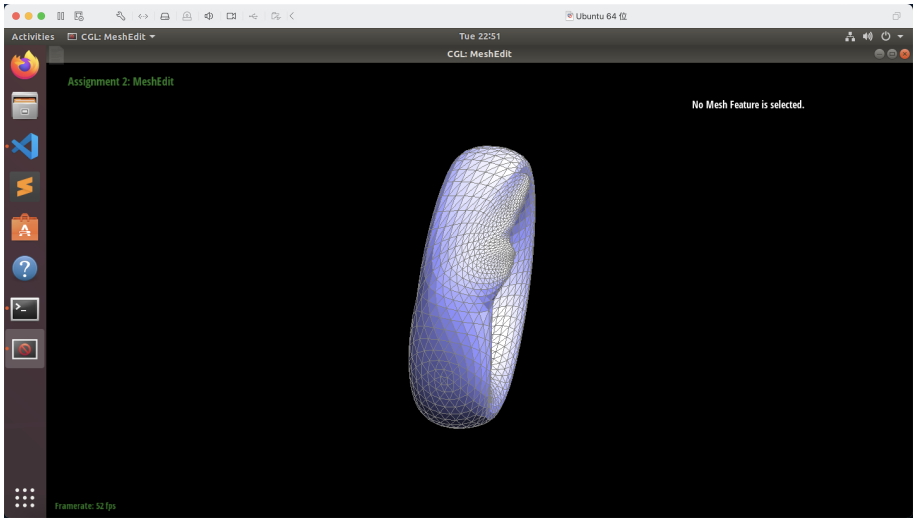

Figure 26: loop Subdivision with pre-spliting

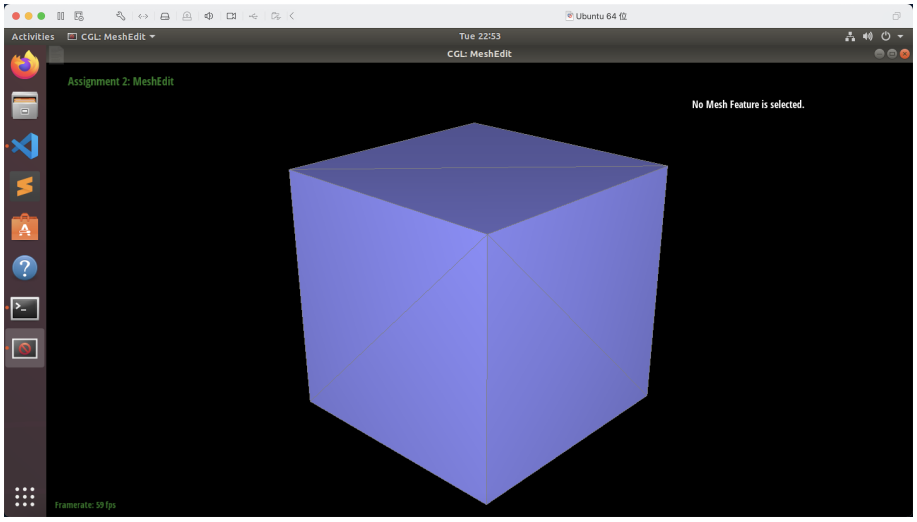

Figure 27: cube.dae

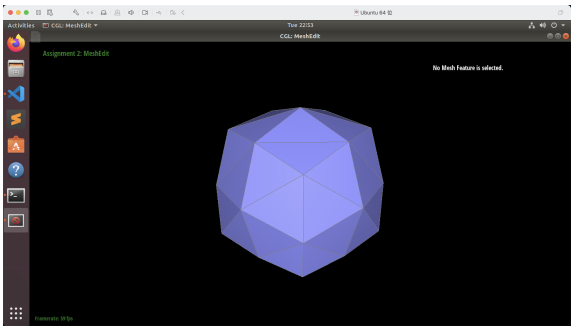

Figure 28: cube.dae

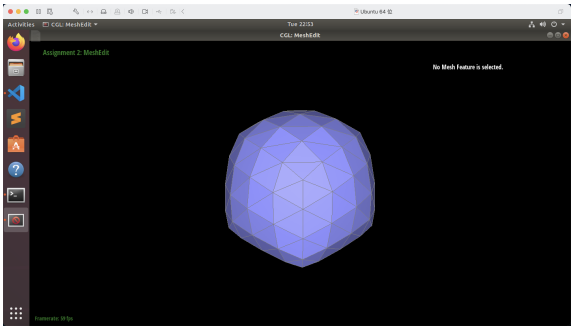

Figure 29: cube.dae

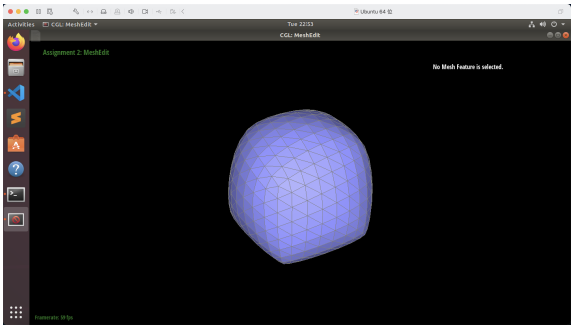

Figure 30: cube.dae

7.2 Pre-process the cube with edge flips and splits so that the cube subdivides symmetrically

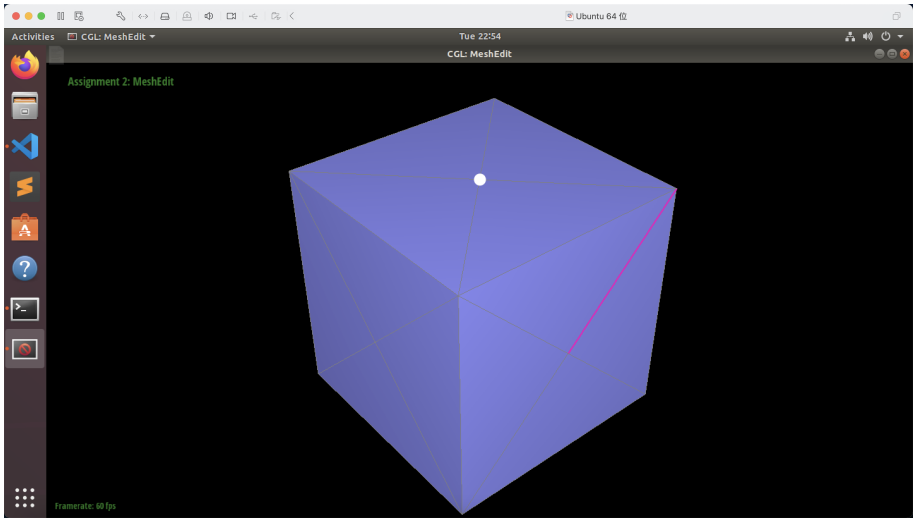

Figure 31: cube.dae with Pre-process

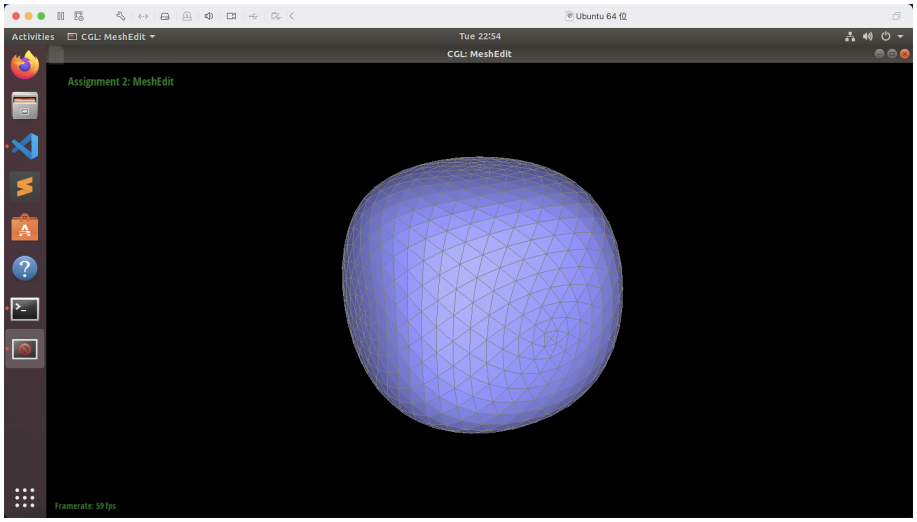

Figure 32: cube.dae with Pre-process

Because the number of triangle faces is too small and the area of each triangle is very large, it will eventually lead to subdivision along the traces of the triangles in the upsample and eventually lead to errors, which can be avoided if the large triangles are split in advance(Pre-process).

8 Extra

1.Implemented edge split operation with boundary face 2.After the implementation of the edge split, we can support meshes upsample with boundary.