Computer Graphics Assignment Report 2

Manideep Nizam

April 2019

1 Problem Statement

1.a 3D Rendering Using MVC Architecture:

Render a 3D scene using more than two objects in the scene. The ob- jects will be rendered using their surface meshes given in .ply file format. Datasets are available at: https://people.sc.fsu.edu/ jburkardt/data/ ply/ply.html Larger resolution meshes are available at: http://visionair.ge.imati.cnr.it/ontologies/shapes/viewmodels.jsp

1. Write a parser for the ply files and define the mesh as "model" in your implementation. Your scene must contain multiple models which could be from different ply files. Each object can be normalized to be contained inside the unit cube (i.e. x,y,z range of the cube is in range [0,1]). Each mesh must be rendered with the following options (one at a time): (a) wireframe mesh, where the triangles are not filled; (b) filled mesh, where all the trian- gles are filled with one single colour.

2. Implement object-centric affine transformations, like translation, rotation, scaling. Associate keyboard and mouse inputs for the transformations, where translation and rotation are governed by mouse inputs. e.g. left mouse button may be used for translation and right for rotation; "+" and "-" keys may be used for scaling. The mouse button inputs for both "press down", "drag" and "release" can be used for covering all aspects of translation and rotation. In order to pick an object in the scene, a combination of keyboard and mouse button inputs must be used. The rotations must be implemented using quaternions.

3. Compute vertex normals by averaging the normals of the triangles sharing the vertex. The vertex normals must be unit vectors. Color the vertex using the normal vector, where (R,G,B) correspond to (x,y,z)- coordinates of the normal vector.

4. Implement a splat for each triangle in the mesh, where a splat is a filled circle contained in the triangle, placed at the incircle of the triangle in the mesh. The circle must be coloured using the normal coordinates, (R,G,B) corresponding to the (x-,y-,z-) coordinates of the normal vector. [There can be different display options for the mesh – wireframe , filled triangles, just vertices without the edges, and as splats.]

5. Implement lighting using a point light source. Draw the light source as a relatively small sphere or using GL_POINTS in the scene. Enable lighting as well the point light source, and using appropriate mouse and keyboard actions, move the light.

6. Depending on your code design, the mouse and keyboard interactions and feedback must be handled between a "view" or a "controller" object. The code for the scene display must be in the "view" object.

2 APPROACH

2.a MVC ARCHITECTURE

I used three classes Model,View,Controller and a Main.

Model is independent of View and Controller, Contains all the information of model including light position and light color.

View is independent of model,controller and contains shader,lightshader information.Model is rendered using draw function in view which takes model object as argument.

Controller contains information of all the models.when we use a particular model all its information is loaded in controller and operations like translation,scaling and rotation are done. For quaternion rotation it uses trackball function from trackball.cpp.

2.b Model:

Model contains all the information of model like Vertices, Triangles, Normals, TriangleNormals.It contains information of a unit circle in xy plane. It also contains information of SplatVertices , SplatTriangles , SplatColor .It contains Vertexarrayobject and vertexbufferobject information of both model and light.

Each model is given a id in the constructor,which is used to render all the models without overlapping.

Initially Vertices, Triangles are readied from .ply file,normalizes to -1 to 1 and then trianglenormals and normals are computed.Then, this information is loaded in to VAO,VBO[3].

Then a unit circle is computed by the use of genetateCircleVertices(), generateCirleTriangles() functions which constructs a unit regular polygon, which is a approximation to circle.(The more number of sides the better circle).

Then Incetres and inradius of all the triangles are computed.

Now, splatvertices are generated using generateSplatVertices() function which constructs vertices of incircles of all the triangles using the circlevertices which are already computed.

we construct a incircle of a particular triangle by scaling the unit circle in xy plane by the inradius of triangle, then rotating it by the angle made by normal of triangle and normal of circle along the axis which is cross product of these two normals and then transclating it to incenter of triangle.In this way vertices of all incircles are computed.

SplatTriangles,triangles of incircles are computed with the help of circletriangles.Splatcolor is the normal of the corresponding triangle.Now, this data is loaded in CVAO,CVBO[3].

My light is model dependent and each model has a light source.Each light source is independent of the other.

2.c VIEW:

View contains information of the shader programme and adding,compiling,creating shaders are done in view.

View contains a function draw which takes a model as argument and binds VAO,CVAO depending on the rendering id and renders the object.

2.d CONTROLLER:

Controller has all models of our scene, Depending upon the current model, model information is loaded into it and continuously update models.

Rotation is implemented using Quaternions, which make use of trackball function from trackball.cpp.

3 ANSWERS

3.a

We can optimise the I/O operations by using instancing. With the help of instancing we can draw same object many times with single render call.In instancing instead of using glDrawArray, we use glDrawArrayInstanced which has one additional arguement of number of times same object is to be drawn, in the similar way glDrawElementsInstanced is used instead of glDrawElements.

3.b

We may test out light by observing the intensity of light as the object moves towards or away from the light source. If the object moves towards the light source the intensity should increase and if the object moves away from the light source the intensity should decrease.

And as color of a pixel is the product of object's color and if the light is of a single color(R or G or B) we should see only that color.

Also, By observing shadows and by comparing with the expected output of ray tracing,etc.

3.c

Splat on Vertices can be done in the same way splat on triangles in done. But, radius should be taken such that no two circles overlap. So taking the half of minimum distance from the corresponding vertex to all its adjacent vertices can work. Normal of the circle will be normal of that of vertex and color of the circle will be the unit normal of vertex.

4 REFERENCES

4.a

Opengl:-https://learnopengl.com

4.b

GLFW input guide:-https://www.glfw.org/docs/latest/input_guide.html

4.c

Quaternion:-https://www.google.com/url?sa=t&source=web&rct=j&url=
http://web.cse.ohio-state.edu/~shen.94/781/Site/Slides_files/trackball.
pdf&ved=2ahUKEwiP4ZqEocDgAhWXiHAKHSykDLwQFjAAegQIARAB&usg=AOvVaw01HYBm82T0o0cm8A_
_RWru&cshid=1550320314951

4.d

Quaternion:-https://www.cosc.brocku.ca/Offerings/3P98/course/OpenGL/glut-3.7/progs/examples/trackball.c

4.e

Instancing-http://www.informit.com/articles/article.aspx?p=2033340&seqNum=5

4.f

PRIMER I, PRIMER II by Ankita Victor.