Computer Graphics

Assignment-3+4

Ayush Saraswat (IMT2016123)

- Assignment 3

  Rendering textures and implementing texture maps on different geometrical models.
- Assignment 4

  Implement a scene graph and animate the 4 objects used in the previous problem and vary speed of movement.

Mappings were as follows-

1.) **Planar**: We ignore the z coordinate of the points and map x and y coordinates of the model to the texture.

2.) **Cylindrical**: We map the y coordinate of the models to the y coordinate of the texture but for x coordinate mapping we use angle θ which goes around the curved surface of the cylinder.

3.) **Spherical**: We map y coordinate to angle φ which goes around the curved surface in the y-z plane and the x coordinate to angle θ which is similar to that in the cylindrical texture map and goes around the x-z plane.

-This was applied on 3 geometrical objects and one arbitrary object

.

-Fixed lighting was done to give effects of global illumination.(Assignment 3)

-**Parser** : This was used to read ply files and return the number of faces, number of vertices, the vertices, the element index array. This information was then passed to the shaders.

-**Shader class** : This was used to create, add and compile shaders. It returns a shader ID which can be further used to refer to the created shader object.

-**Model Class** : It has an array of vertices, an array of indexes, transformation matrices, vertex array identifier, vertex buffer identifier and an index buffer identifier.
selection of the model to check and store whether the model has been clicked or not.
A children vector enables us to implement the Scene graph here.
Splatting, updating the model and its children recursively is also implemented here.
There are methods to handle any user input and perform transformations by modifying the transformation matrices.
Texture mapping and texture selection was also implemented here itself.

-**View Class** : This class corresponds to View in MVC, it handles the creation of windows and initialization of various OpenGL requirements.

-**Controller Class** : This class corresponds to the Controller in MVC, it has a view and a renderer object. It handles any interaction between the user and the View/Renderer, displaying of the scene and also creates a Shader object to compile shader programs. It also loads and binds textures to the GPU.

- Animation(Assignment 4) :Object A is the root of the scene graph tree, Object B is a child of object A and revolves around it , Object C is a child of object B and it follows it .Object D is a child of Object C and it does a translation on top of it. The motion is controlled using an integer which determines the type of motion the object follows. Lights can be called as a child to every model and hence will follow and take the same transformation as the parent(model) it is illuminating.