Computer Graphics

Assignment – 5

Name: Ishan Mehta

Roll No: 2019309

COMMENTS ARE WRITTEN IN THE CODES

Question 1)

First, we will load the image (“pattern.png”) using the below function.

```
stbi_load("pattern.png", &width, &height, &nrChannels, 0);
```

The texture is a property of the material and thus image data is stored as a variable in the material object along with its width, height and number of color channels.

Now when the ray hits the sphere at a point P. We calculate the texture coordinates U, V from the point P(X,Y,Z) as –

$$U = \frac{\left(tan^{-1}\left(\frac{Y}{X}\right) + \pi\right)}{2 * pi}$$

$$V = \frac{\cos^{-1}\left(\frac{Z}{Radius}\right)}{\pi}$$

U and V are lie between 0 and 1. Then they are scaled to [0, texture_width] and [0, text_height] resp.

The colour which corresponds to the pixel of texture at row U and column V is applied to the point P.

The reference of the object intersected, the values U and V, the normal at the point of intersection are all saved as parameters in the ray Object.

Using this ray and these values we find the shading for the intersected point and Material::shade function returns the colour of the pixel of the texture correspond to point P using U and V as explained above.

Question 2)

ImplicitSurface is derived from the object class.

It has a function float value(Vector3d P), which returns the implicit function value for the point P(X,Y,Z)

It also has a function getNormal(Vector3d P) which return the normalized normal vector at point P of the implicit surface. This is required for the shading of the surface.

The normal is computed as

$$N = \frac{\partial F(X,Y,Z)}{\partial X}, \frac{\partial F(X,Y,Z)}{\partial Y}, \frac{\partial F(X,Y,Z)}{\partial Z}$$

Where F(X,Y,Z) is the implicit function given to us.

The ‘intersect’ function uses the ray marching algorithm. It takes in input a ray defined as O + tD. Where O is the origin of the ray and D is the direction of the ray and t is the parameter.

In the ray marching algorithm we first set the value of the parameter t to 0 and then increment it by dt (which is 0.1) till t_max(which is 50). This geometrically means that we are moving along the line in small steps (small steps are defined by dt).

After each step when we land on some point P on the line, we check what is the value of value(P) if it is very close to 0 then that means we are very close to the surface and can consider this as an intersection. And then we paint this pixel of the screen using normal of the surface and the material.

If this never happens in the loop and t reaches t_max then this means there is no intersection and we paint the corresponding pixel with the background color.