CS-AD-216: Foundations of Computer Graphics

Assignment 9, Due: December 12

Problem 1 (10 points)**.**
Modify the ray tracing program so that most of the computation is done in the fragment shader. The idea is to draw a square with corners $(-1,-1)$ and $(1,1)$. Each fragment then corresponds to a pixel and the corresponding fragment shader can then do the necessary computations to figure out the color of the pixel. Move the entire scene set up and the relevant functions to the fragment shader. You will need to use C-style structs in the fragment shader since GLSL ES 2.0 does not support C++ style classes. Please have a look at `http://math.hws.edu/eck/cs424/notes2013/19_GLSL.html` to learn how to use these. Your program should display an animated scene with at least two lights and interesting shadows and reflections. Please have a look at the code in the directory `'Ray Tracer (Animated)'`.

Further modify the code to add *anti-aliasing*. This can be done by doing recursive ray tracing by shooting multiple random rays through a pixel and taking the average of computed colors.

Bonus (5 points): Make at least one of the spheres a billiard ball by putting the texture `billiard_ball_texture.jpg` on it. This ball should be spinning slowly along an axis passing through its center so that the texture is not static. The ball should look similar to ball shown in `billiard_ball.jpg` (ignoring the reflection of the area lights).

Problem 2 (10 points)**.**
Use billboarding to add a number of copies of the trees given by the `tree1.jpg` and `tree2.jpg` to the scene you created in Assignment 8. Note that the trees should always be facing the camera and only rotate around their trunk. The white background in the pictures should be discared. You can use the keyword `discard` in the fragment shader to discard a fragment.

Problem 3 (10 points)**.**
The goal of this exercise is to display a pear. The attached files include an obj file, an mtl file, a diffuse map, a normal map and a specular map. The specular map is just another texture which is used exactly like the diffuse map and provides the shininess value at each point. The image is black and white i.e. R,G,B values at each pixel are the same. Thus you can use any one of the components to compute the shininess value. The value that you get will be between 0 and 1. You will need to multiply it with a large number (e.g. 300) to get the shininess value.

The obj file contains three objects, each beginning with a line starting with `usemtl` describing the material from the mtl file to be used. You will need to separate these into three obj files and convert each into javascript notation using the script `obj2json.html`. All three objects use the same images for the diffuse and normal maps.

The mtl file indicates that the specular map is to be used only for the first object (pear skin). The shininess values for the other two objects, pear flower and pear tail, are 0 and

45.09 respectively. You can ignore the other details in the mtl file.

Please provide a trackball interface.