Assignment#2 GLSL Shaders

20131329 신선우 (전기전자컴퓨터공학부/학생)

**1. Introduction**

This assignment is making GLSL Phong and cartoon shaders.

1. Phong shader with interactive parameters change.

2. Cartoon shader with silhouette and cartoon rendering with interactive parameters change.

3. Keyboard interaction with previous assignment code.

I used lecture notes code for Phong shading and made cartoon shading with part of Phong code.

**2. Method**

- createGLSLProgram

This function is already implemented in textfile.cpp. So, I just used this function to making glslprogram. This program automatically load, compile, link shader.

- Phong Shading

- Phong Vertex shader

There is only one changed part with the lecture note’s code. This code caused type error.

 Changed one.

This code is changed code because of bug. Previous code caused data type error.

So, I moved the calculation part inside of the length function.

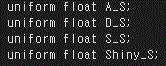
- Phong Fragment shader

Also, in Phong fragment shader, I fixed some type error. For example, the vec4 n was vec3 n. So, I added the 4th matrix parameter in normal and type conversion vec3 to vec4(normal, 0); Then, it works properly.

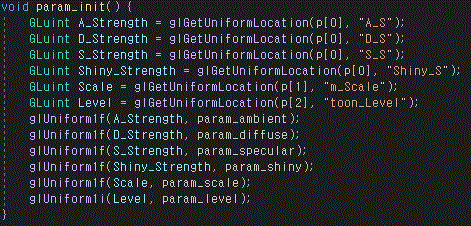
There were several errors like this.



I added some parameter which passed by main.cpp.



Those parameters are performed as ambient, diffusion, specular, shininess parameter.



All parameters used in shaders are passed by glGetUniformLocation and glUniform functions, in main.cpp.

Those functions change shader’s parameters interactively.



This line presents a diffuse color. The D\_S parameter is the diffusion parameter which can changes 0.1 to 1. It makes diffusion color darker and brighter. 





Those codes represent how to use Ambient, Specular, Shiny parameter.

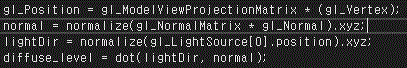


Also, I changed attenuation value 1 to 8. The default value was 1.

Because, if the value is 1 then all of the object color is too dark.

- Toon Vertex shader

The important part is finding the level of light.



So, I used code which already used in Phong vertex shader, gl\_Position and normal.

The lightDir is light direction which is a normalized light source position.

Also, the level was calculated by the dot product which represents the value of ‘Lambert’s cosine law’ value.

This parameter is passed to Toon fragment shader. 

- Toon Fragment shader

To interactive level scale, the shader gets an int toon\_level value.

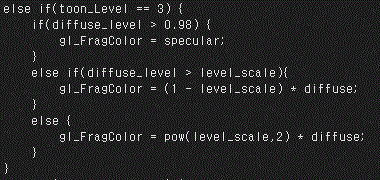
In Toon Fragment shader, the only diffuse and specular are calculated.

Because the object’s color is represented by the level of ‘Lambert’s cosine law’ value.

Also, I make a float variable which is level\_scale. This variable is 1/(toon\_level - 1)

The difference of toon\_level was assigned by if statements.

This is the code of toon\_level3.

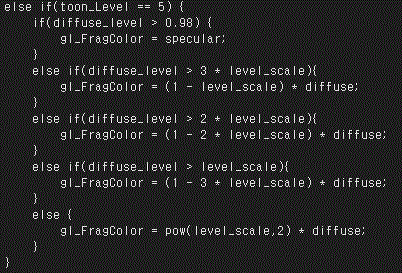


The toon\_level 3 represents the level of color is 3.

So, I set every first color is specular which diffuse\_level is bigger than 0.98.

Also, the lowest value is the power of level\_scale multiply original diffuse color.

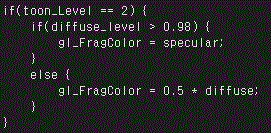
This is the code of toon\_level5.



The difference of level is made by multiply of level\_scale.

Also, the strength of diffuse color is made by multiply of level\_scale.

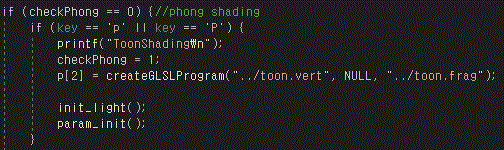
The exception one is level2



There is only specular and diffuse color part. So, I set diffuse strength value 0.5 as default.

- Keyboard interaction

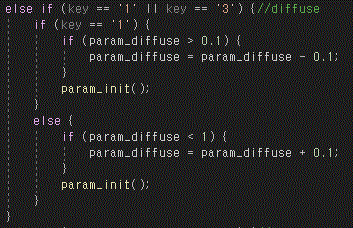
The previous assignment there was keyboard interaction. So, I can easily implement this function.



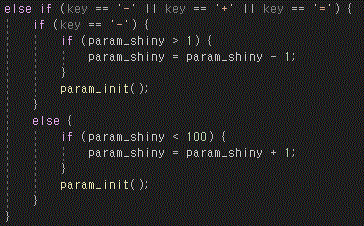
The checkPhong is a checking bit to check whether the current state is Phong shading or not.

If the checkPhong is 0 then the current state is 0. If press ‘p’ then the state is switched to toon shading.

The init\_light(); and param\_init(); is the function of loading parameters. To render.



This is the part of Phong shading’s diffuse parameter change. If the ‘1’ is pressed then the parameter is decreasing and applied by param\_init() function. If the ‘3’ is pressed then the parameter is increasing.

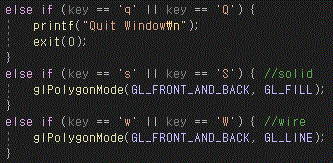


This is the part of Phong shading’s shiny parameter change. I added one more key ‘=’. Because, in my keyboard, the key of the left side of backspace, doesn’t work if there is only ‘-‘ and ‘+’ if statement.

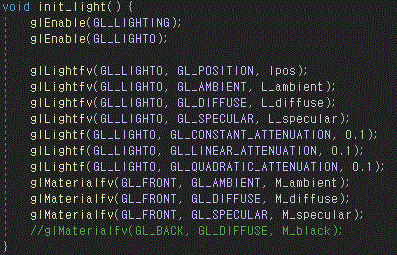
So, to increase human interaction intuition, I added ‘=’ key to that part.

If ‘-‘ is pressed then the parameter is decreasing. And the object’s shininess is increasing.

Also, I added a solid and wireframe mode changing option. And quit the window option to make convenience.



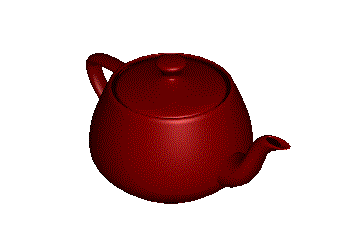
This code is init\_light() code.



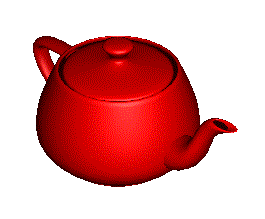
This code passes the parameter of the position of light and color of light and attenuation factors.

**3. Result**

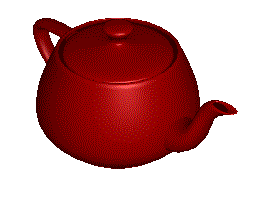
The default state of Phong shading with teapot.



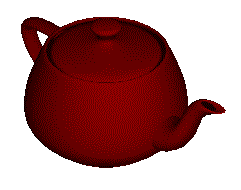
Low and High diffusion parameter



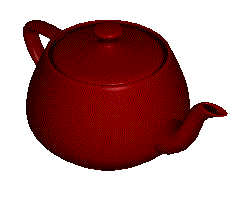
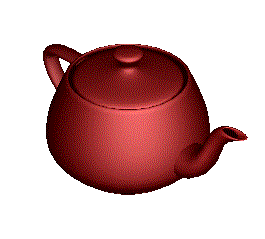
Low and high ambient parameter



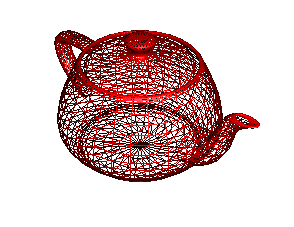
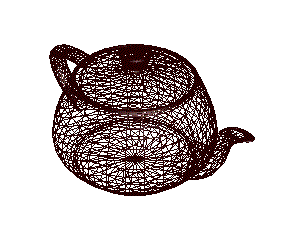
Low and high specular parameter



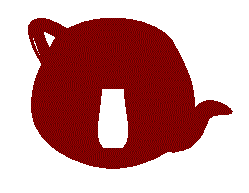
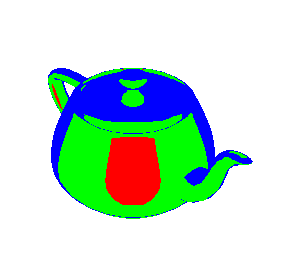
Low and High Shininess parameter



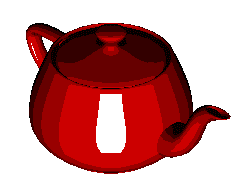
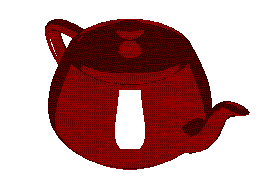
The wireframe with low diffusion parameter and high diffusion parameter.



Toon shading of test image and level2



Toon shading of level 3 and level6



**4. Conclusion**

The most of the task was successfully implemented but the silhouette rendering was failed.

I can’t figure out how to make multi-pass rendering at the same time.

Also, I didn’t implement trackball in the last assignment. At this assignment, also, I didn’t implement the function. Because of the lack of time.