

ICG HW3 Report

110550108 施柏江

1. How I implemented the shading/effects:

How I get the vector L, N, V, R, H:

L 是指向光源的向量，所以只要將光源的位置減掉物體的位置即可取得。N 是物體的法向量，所以只要將從 vertex shader 傳入的法向量正歸化即可取得。V 是指向觀測者視線的向量，所以只要將相機位置減掉物體的位置即可取得。R 是 L 的反射光，可以利用 $\text{reflect}(L, N)$ 取得。H 是入射光和視線正中間的向量，將 L 和 V 相加並正歸化即可取得。

Blinn-Phong shading:

先將 material 和 light 各自的 ambient, diffuse, specular 存入 K_a , K_d , K_s , L_a , L_d , L_s 。再利用 L_a , K_a 和 texture 得到 ambient; L_d , K_d , texture 和 L 與 N 的內積值得到 diffuse; L_s , K_s , shininess 和 N 與 H 的內積值得到 specular。最後將 ambient, diffuse 和 specular 相加得到最終的顏色。

Gouraud shading:

Ambient, diffuse 和 specular 的算法與 Blinn-Phong shading 大致相同，由於顏色是要經過內插法取得，需在 vertex shader 計算完成後再傳到 fragment shader 著色。

Flat shading:

Ambient, diffuse 和 specular 的算法與 Blinn-Phong shading 大致相同，但是法向量是在 geometry shader 中計算每個三角形的法向量再傳入 fragment shader。

Toon shading:

先利用 K_s , L_s , L , N , V , R 計算出 specular，如果 L 和 N 的內積絕對值 < 0.3 ，將顏色設置為深棕色；如果 L 和 N 的內積絕對值 ≥ 0.3 且 specular 的值大於 0.01 ，將顏色設置為淺棕色；其他情況則設置為正常的棕色。

Border effect:

如果 V 和 N 的內積絕對值 < 0.3 ，我將顏色設為亮米白色，在邊緣製造出亮亮的效果，否則就將顏色設為原本 texture 的顏色。

Dissolve effect:

我利用 uniform 傳入 dissolvePos，它會隨著時間而增加，並將一個變數 dissolve 初始化為 1，如果物體的 $position.x < dissolvePos$ ，就將 dissolve 設為 0，最後再將物體 texture 顏色的 alpha 乘上 dissolve，製造出鹿從尾巴到頭隨著時間逐漸消失的效果。

2. The problems I met and how I solved them:

一開始在 `createShader` 時發生了無法打開 `shader` 的錯誤，以為是 `sample code` 給的程式碼需要修改，後來才發現是 `shader files` 的路徑忘記在前面加上 `shaders/` 了。

在寫 `Flat shading` 時，由於是第一次寫到 `geometry shader`，我對於以 `struct` 形式的 `input` 和 `output` 以及如何在 `for` 迴圈裡正確的 `assign` 數值還不太熟悉，於是重新回去看講義模仿它的寫法才成功完成。

在寫 `Toon shading` 時，一開始我照著 `spec` 上給的方式去設置顏色，發現深色的部分比 `demo` 裡的少很多，以為是 `L` 或 `N` 計算錯誤，後來在討論區才發現可以更改深色的條件，夾角不一定要 > 90 度。

在寫 `Dissolve effect` 時，我想調整顏色的 `alpha` 值來達到消失的效果，但是發現不管怎麼調 `alpha` 物體看起來都長得一樣，在網路上搜尋了一下發現只要加入 `glEnable(GL_BLEND)`和 `glBlendFunc(GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA)`後，就能達到我想要的效果了。