

## HW3 – Shading

0816039

李品慈

- Explain how you implement the above shading/effects.

```
normal = mat3(transpose(inverse(M))) * in_normal;
uv= texcoord;
fragPos = vec3(M * vec4(in_position, 1.0));
worldPos = vec4(in_position, 1.0);
```

首先，先將 phong 跟 toon shading 的 vertex shader 與從 main 傳入的資料做計算：

1. 把 Normal 和 normal matrix 相乘，以避免 scale 後 normal 指向錯誤
2.  $Uv = texcoord$
3. 這裡我加入了一個 fragPos 來記錄現在 fragment 的 position 以利往後 fragment shader 使用
4.  $worldPos = in\_position, 1.0$

```
// add in phong compnents
vec4 ambient = vec4(La, 1.0) * vec4(Ka, 1.0) * albedo;

vec3 L = normalize(WorldLightPos - fragPos);
vec3 N = normalize(normal);
float diff = max(dot(L, N), 0.0); // dot always >= 0
vec4 diffuse = vec4(Ld, 1.0) * vec4(Kd, 1.0) * albedo * diff;

vec3 V = normalize(WorldCamPos - fragPos);
vec3 R = reflect(-L, N);
float spec = pow(max(dot(V, R), 0.0), gloss); // dot always >= 0
vec4 specular = vec4(Ls, 1.0) * vec4(Ks, 1.0) * albedo * spec;
```

在 phong shading 的部份我們必須要計算三個分別的 component。

1. ambient: 相對簡單，只需將  $La * Ka * model\ color$
2. diffuse: 需要先計算光源和 fragment 的 vector(L)，以及與 N 同樣進行 normalization，以利後續在內積的計算。在計算完畢後內積 L 和 N，這裡取和 0 的 max 是為了保持值  $\geq 0$  並且  $< 0$  時 diff 皆設為 0。最後將  $Ld * Kd * diff * model\ color$  即為 model 的 diffuse。
3. specular: 需要先計算視線和 fragment 的 vector(V)，以及利用剛剛算出的 L 和 N 找到反射的 vector(R)。計算完畢後內積 V 和 R，一樣取和 0 的 max。最後將  $Ls * Ks * spec * model\ color$  即為 model 的 specular。

```

vec3 L = normalize(WorldLightPos - fragPos);
vec3 N = normalize(normal);
float level = max(dot(L, N), 0.0); // dot always >= 0

float intensity;

if(level > 0.75) intensity = 0.8;
else if(level > 0.30) intensity = 0.6;
else intensity = 0.4;

```

```

color = vec4(Kd, 1.0) * albedo * intensity + edge_color;

```

Toon shader 相對簡單，在這裡會依 L 和 N 之間的夾角來決定 intensity，我依照 spec 上的設定將 model 分為三個不同 intensity 來顯示。而這裡的 intensity 只要在最後計算 color 時與 model color 相乘即可。

```

float edge_intensity = 0;
if(EdgeFlag == true) edge_intensity = 1 - max(dot(V, N), 0.0);
vec4 edge_color = vec4(1, 1, 1, 1) * pow(edge_intensity, 3);

```

最後則是 edge effect，在 phong 和 toon shading 中實作方式是相同的。依照教給的提示，在這裡我設定 intensity 為  $1 - V \cdot N$ ，實現離邊緣越近，值越大的特性，而為了讓 edge 更明顯一點，我將 pow 中的 5 改為 3，讓 edge 的範圍稍微擴大一些。

### ● Describe the problems you met and how you solved them.

這次作業相較上次簡單許多，也由於上次的經驗，這次對 shader 間的資料傳輸更清楚，也更容易了解現在手頭有什麼資料，或是能用來做什麼，所以沒有遇到什麼大問題。

唯一一個問題只有一開始在實作 phong 的時候，因為誤解 worldPos 是 fragment 的 position，所以 specular 等等的顏色不會因為旋轉而改變，後來參考了助教提供的 reference 網站，根據上面的介紹加入 fragPos 後問題就解決了。