

# CG\_HW1 作業報告

## 1. 程式碼介紹

### 1.1 讀取 .obj 和 .mtl 檔案

- 先用 HW1 的方法讀取 .obj 檔案
- 針對 usemtl 來讀取 .mtl 檔案，並分別將不同名稱的 material 存在 unordered\_map 中。
- 不同材質的 f 新增一 subMesh，對應到 mtlMap 的 material 中，然後依照 HW1 的方法，紀錄 PTN 頂點(也要將多邊形分割成多個三角形)。
- 這邊 maxVertex 因為考慮到負值，所以改為 -FLT\_MAX。

### 1.2 填充 initialize 和 release&新增 Render 和 CreateBuffer

- 這邊參考 HW1 寫法，但 iboId 要另外處理。
- **CreateBuffer** 這裡和 HW1 不同的地方是，因為 iboId 是 subMesh 的成員，所以需要遞迴做處理。
- **Render** 會在 main.cpp 中使用，做 bind buffer 和 shader uniform 的宣告。

### 1.3 main.cpp 的 RenderSceneCB 和 Vertex Shader

- **RenderSceneCB** 這裡要先將資料傳到 vertex shader 中，所以要 uniform 宣告 worldMatrix, normalMatrix, MVP。
- 這邊有一件很重要的事，就是需要調用 **CreateBuffer**(我一開始忘記，結果超級坐牢)。
- **Vertex Shader** 要做的是把所有資料轉到世界座標系(vertexPosition 和 vertexNormal)，之後要在 **Fragment Shader** 計算打光。輸出的東西是 iPosWorld 和 iNormalWorld。
- 可以參考 shader 的 vis\_normal.vs。

### 1.4 Fragment Shader 和 Model debug

- 需要先確定 subMesh 和 material 的讀檔沒有問題，所以 **Vertex Shader** 處理完以後，在 **Fragment Shader** 這邊可以先讓 **FragColor** 賦值一個 vec4(0.0, 0.0, 0.0, 1.0)的值，以確定沒有問題(第一次寫的時候，subMesh 有問題，ColorCube 缺一塊，只好重寫)。
- 重寫以後，沒有問題以後，再開始寫 direction light、point light 和 spotlight。

### 1.5 Direction light

- 參考 Shading 的 `gouraud_shading_demo.vs` 寫 ambient light 和 direction light。
- 把 Vertex Shader 傳進去的 `iPosWorld` 和 `iNormalWorld` 做 **normalize**。
- 用投影片的 Diffuse 和 Specular 公式計算，Specular 先用 `vE` 和 `light direction` 算出 `vH` 再繼續計算。
- `main.cpp` 傳入 **uniform** 參數。

### 1.6 Point light

- 參考 Shading 的 `gouraud_shading_demo.vs` 寫 Point light。
- 計算用距離計算出光的強度(**radiance**)，帶入 Diffuse 和 Specular 公式算出。
- `main.cpp` 傳入 **uniform** 參數。

### 1.7 Spotlight

- 這邊可以參考 `main.cpp` 前面定義 `parameters` 的地方，**light.h** 定義出 spotlight 需要的其他參數 (`direction`、`cutoffStartInDegree`、`totalWidthInDegree`)。
- **shaderprog.cpp** 這邊要新增 `loc` 參數和傳進 **uniform** 要叫的名字。
- 照 spotlight 的線性衰減方式計算 **intensityFactor**，計算的 `diffuse` 和 `specular` 乘以這個衰減的 `factor`，相加後回傳。
- 最後將所有光線相加後，賦值到 **FragColor**。

### 1.8 Dynamically Load Object

- 更改 **LoadObjects** 和 **ProcessKeysCB**，當 `mesh!=nullptr` 時候，刪除並 new 新 object(這裡也要記得 **CreateBuffer**，超級重要!!!!)。
- **ProcessKeysCB** 設定 `p(prev)`、`n(next)`，用於切換上/下一個模型。

## 2. 結果(截圖)

