

HW3 Texture Mapping

105072123 黃海茵

老師說這次不需要把畫面左右分割，然後光源只需要 directional light，所以我就只留下 per pixel 那塊，這部分都是用上次的 code。

● TODO

```
// [TODO] Bind the image to texture
// Hint: glGenTextures, glBindTexture, glTexImage2D, glGenerateMipmap
glGenTextures(1, &tex);
glBindTexture(GL_TEXTURE_2D, tex);
glTexImage2D(GL_TEXTURE_2D, 0, GL_RGBA, width, height, 0, GL_RGBA, GL_UNSIGNED_BYTE, data);
glGenerateMipmap(GL_TEXTURE_2D);
```

在 opengl 中要使用 texture 的話，要先用 glGenTextures，使用方法是 glGenTextures(GLsizei n, GLuint *textures)，n 指定生產的紋理物件的數量，textures 是一個非負整數陣列，用了存儲 n 個紋理物件的 ID。
再呼叫 glBindTexture，使用方法是 glBindTexture(GLenum target, GLuint texture)，target 為將紋理物件和目標 GL_TEXTURE_2D, GL_TEXTURE_3D, GL_TEXTURE_2D_ARRAY 或 GL_TEXTURE_CUBE_MAP 進行綁定，texture 為進行綁定的紋理的控制碼。

再呼叫 glTexImage2D，使用方法是 glTexImage2D (GLenum target, GLint level, GLenum internalFormat, GLsizei width, GLsizei height, GLint border, GLenum format, GLenum type const void * pixels)。

然後因為要實作 mipmap 所以呼叫 glGenerateMipmap 來實現。以上四個 function 只要填入對應的資料即可。原本 GL_RGBA 我是填 GL_RGB，發現圖片會變得很奇怪，看了討論區發現有人有一樣的問題，才知道要改成 RGBA 才能正常顯示。

```
// [TODO] Bind texture and modify texture filtering & wrapping mode
// Hint: glActiveTexture, glBindTexture, glTexParameterf
glActiveTexture(GL_TEXTURE0);
glBindTexture(GL_TEXTURE_2D, models[cur_idx].shapes[i].material.diffuseTexture);
if (mag_linear) {
    glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_LINEAR);
}
else {
    glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_NEAREST);
}
if (min_linear) {
    glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_LINEAR_MIPMAP_LINEAR);
}
else {
    glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_NEAREST);
}
if (addr_repeat) {
    glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_REPEAT);
    glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_REPEAT);
}
else {
    glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_MIRRORED_REPEAT);
    glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_MIRRORED_REPEAT);
}
```

在這個 TODO 中，首先用 `glActiveTexture` 設置啟動的紋理單元（texture unit）。每一個紋理單元有多個紋理目標（texture targets）選擇（`GL_TEXTURE_1D, 2D, 3D` or `CUBE_MAP` 之一）。然後 `glBindTexture` 在上一個 TODO 說明過了，這裡就不再解釋一次。

最後根據 `mag_linear`, `min_linear` 與 `addr_repeat` 三個變數來控制切換的 texture filtering，並使用 `glTexParameterf()` 函數來確定如何把紋理象素映射成圖元。其中對 `mag_linear` 有 linear 與 nearest 兩種，`min_linear` 分為 nearest 與 `linear_mipmap_linear`。對 `addr_repeat` 來說則分為 repeat 和 mirror 兩種。

```
// [TODO] Get uniform location of texture
iLocTex = glGetUniformLocation(program, "tex");
```

用法是 `glGetUniformLocation (GLuint program, const GLchar *name)`，`program` 指定要查詢的程式物件，`name` 是要查詢其位置的統一變數的名稱。

```
// [TODO] sampling from texture
// Hint: texture
vec4 texColor = vec4(texture(tex, texCoord).rgb, 1.0);
fragColor = fragColor * texColor;
```

為了將 texture 上色，我們利用 `texture()` 這個 function，第一個參數要放 texture sampler，第二個參數放對應的 texture coordinate。再來考量前面光線的影響，即可算出所需要的 Fragment Color。