CG HW3

一、各個texture mode的差異討論

1. Magnification filtering:

按F來調整linear或nearest。

此種實作方式是因為texture的image較小，而相對來說貼圖區域較大，因此需要放大texture。

Nearest Linear

Nearest Linear

Nearest Linear

2. Minification filtering:

按G來調整linear或nearest。

與Magnification filtering相反，此種實作方式是因為texture的image較大，而相對來說貼圖區域較小，因此需要縮小texture。

Nearest Linear

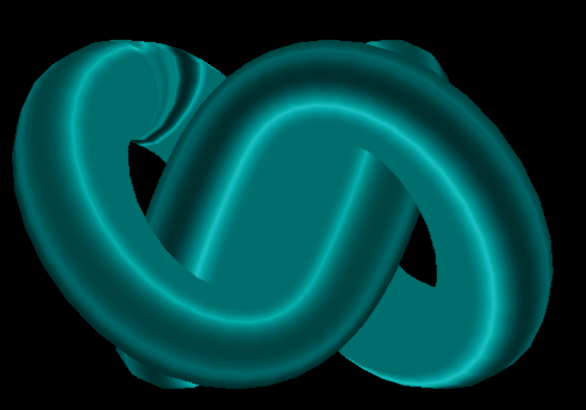
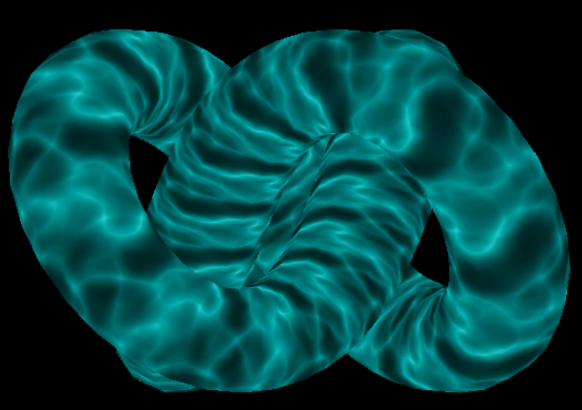
在Magnification filtering、Minification filtering這兩種實作中，nearest在輪廓上可以看出鋸齒狀，而linear模式下呈現的效果與細節處理都比nearest來的平滑，這是因為nearest會選擇將texture貼上最臨近的pixel顏色，所以在pixel之間的轉換較不自然、顆粒感較重，而linear根據鄰近的四個pixel顏色值，使用線性內插計算最終顯示的顏色，因此這樣的實作方式看起來比較平滑，跟真實的物體比較接近。

3. repeat/clamp to edge:按H來調整。

在這種實作中，repeat代表重複的鋪設texture，而clamp to edge代表將texture的邊緣進行拉伸來完成鋪設。

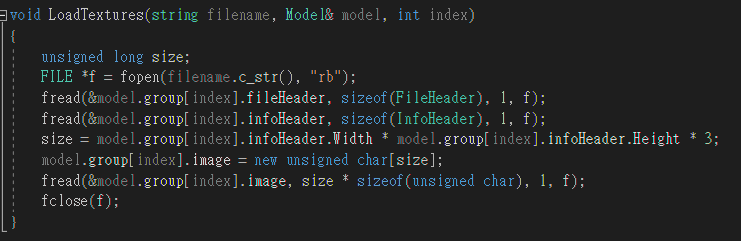
在repeat模式下，我們會限制texture座標，當超出時選擇重複貼上texture，因此產生如圖片中不斷repeat的視覺效果；在clamp to edge時，超出時會選擇重複顯示texture邊緣，所以看起來有拉伸感。

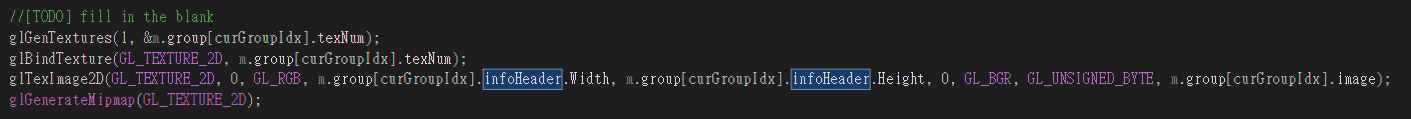
Repeat clamp to edge



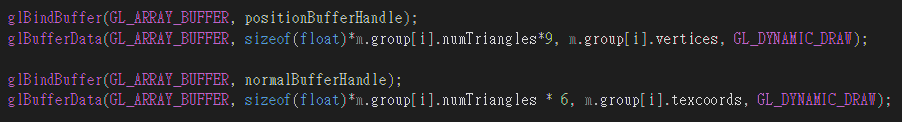
二、load texture過程

首先，在LoadTextures( )這個function中，我們會將texture models讀到memory中。



接下來，利用TraverseColorModel ( )這個function，變數texName即為texture名稱，將剛剛讀到memory中的files使用glGenTextures( )，產生一個在m.group[curGroupIdx].texNum位置的texture memory；glBindTexture( )把 texture image(GL\_TEXTURE\_2D) 與該model bind起來成為model的texture，最後利用glGenerateMipmap( )來generate mipmap。

最後，是如何把memory中的texture搬運到GPU的呢，在這裡我們可以利用draw Model()這個function中的glBindBuffer()、glBufferData()這兩個functions，glBindBuffer()是綁定buffer，glBufferData()會接著將參數資料填入指定buffer，而m.group[i].texcoords就是要傳遞的資料。



因此，在GPU開始執行shader.vert時，會把剛剛提到的texcoords以out.vec2的型態傳給shader.frag進行texture處理，就可以完成這次作業的全部流程。