

計算機圖學-HW1

劉士弘 103060004

• Loading model

1. translation
2. scaling

① Loading model:

首先，一開始的圖為彩色的三角形，雖然我們有讀到obj檔，但因為顯示的矩陣並沒有更改，所以圖形也不會改變。

之後我們會發現在`onDisplay`這個function中，有個`glVertexAttributePointer`這個函式，需要把傳入的第四個參數，改成讀入obj檔的vertices和color的座標，並且把下方的`glDrawArrays`的函式中的第三個參數，傳入總共vertices的數量 $((OBJ \rightarrow numtriangles) * 3)$ 即可得到尚未normalized的圖案。

② Translation:

之後，我們需要用到兩個array來存讀到obj檔案的vertices和color，所以`traverseColorModel`這個function中原本的for迴圈裡，把每一次讀到的三角形的三個點的(x,y,z)座標都存下來，然後去計算(x,y,z)個別的max和min，之後將加除以2，得到中心點座標，再用for迴圈跑每個vertices，把每個vertices的(x,y,z)座標去減掉中心點座標，即可讓圖片移動到正中心。

③ Scaling:

因為在translation時，我們有把所有vertices中的max和min找出來，之後我們只需要把max和min相減之後除以2，就可以得到scaling，但需要注意的是，因為(x,y,z)相減出來得到的值並不會相同，若分開除以不同的scale，會造成deformation，所以我拿最大的值出來當scale，即可完成normalization的最後步驟。

EX:

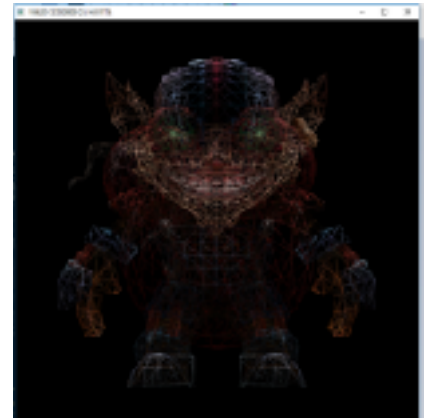
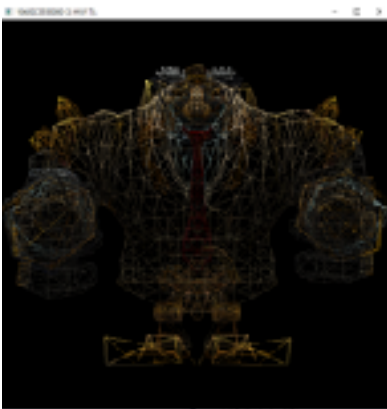


- **Wireframe/solid mode**

- ① **Wireframe:**

因為我們一開始跑出來的圖案，就是solid的圖，所以我們必須想辦法讓它變成wireframe的形式，因為需要用到鍵盤控制，所以我們第一步先到onKeyboard中去設定W是我們要切換模式的按鍵，先define W 為0X77，之後再利用flag控制是要填滿或是線條狀，若要填滿，則使用glPolygonMode(GL_FRONT_AND_BACK, GL_FILL)，並讓flag = 0，若要為線條狀，則使用glPolygonMode(GL_FRONT_AND_BACK, GL_LINE)，並讓flag=1，這樣即可用W鍵切換模式。

EX:



• Keyboard/Mouse

① h - to show help menu:

首先也是到OnKeyboard這個function中，之後define h為0X68，再把想要秀的資訊print出來即可。

② z/x - switch different models :

也是在OnKeyboard這個function中，define x為0X78，z為0X7A，之後利用一個flag，在按Z時+1，按X時-1，之後再利用mod決定要秀出哪一張圖，之後叫重新呼叫loadOBJModel即可。

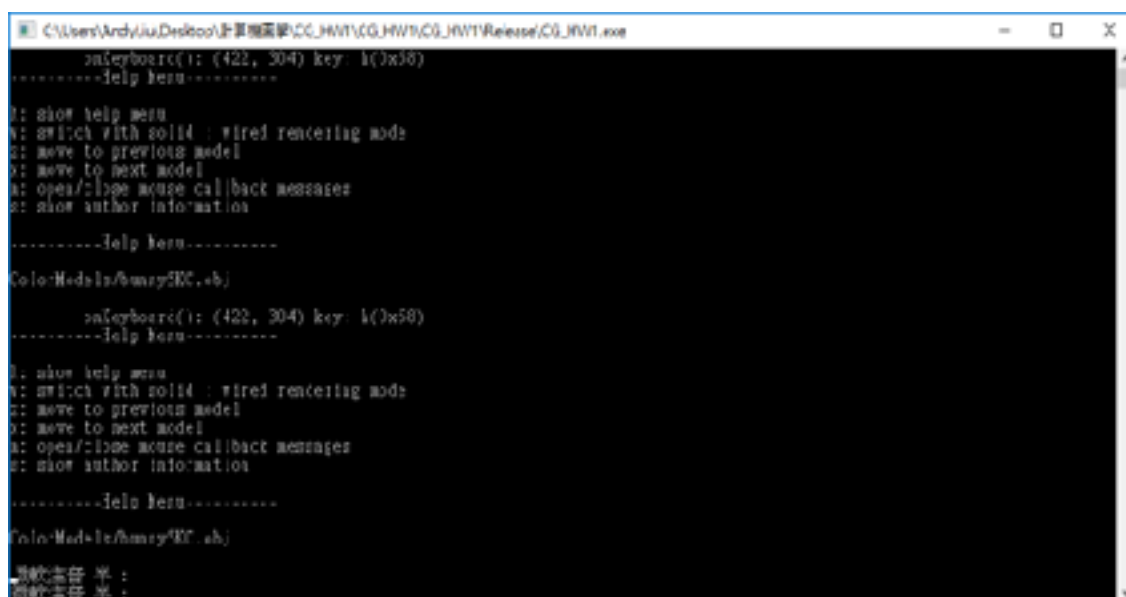
③ w - switch solid/wireframe mode :

在第二部分已經講過實作方法。

④ c - color filter function :

首先，先在main.cpp中宣告一個GLint RGBS用來當傳入的location，之後，需要透過RGBS = glGetUniformLocation(p,"rgbcolor")來得到在shader宣告的uniform的location，之後進到OnKeyboard，當按下C時，就會利用glUniformli(RGBS,c)來更改shader裡面的值，並依照不同的c值，print出不同顏色。

EX:



```
C:\Users\Andy\OneDrive\Desktop\計算機圖學\CG_HW1\CG_HW1\Release\CG_HW1.exe
onKeyboard(i: (422, 304) key: h(0x68)
-----help menu-----
h: show help menu
w: switch with solid : wired rendering mode
z: move to previous model
x: move to next model
m: open/close mouse callback messages
e: show author information
-----help menu-----
Color-Model: /w/wireframe
onKeyboard(i: (422, 304) key: h(0x58)
-----help menu-----
h: show help menu
w: switch with solid : wired rendering mode
z: move to previous model
x: move to next model
m: open/close mouse callback messages
e: show author information
-----help menu-----
Color-Model: /w/wireframe
系統主窗 半:
系統主窗 半:
```

- **color filter function**

- ① **RGB and normal filter:**

在上面提到，會利用`glUniformli(GBS,c)`來更改`shader`裡面的值，而此處的`c`就是一個`int`，每按一次`c`鍵就會+1，之後在`shader`裡面，利用`mod`的方法，來決定要顯示什麼顏色，如果是紅色，則把`gl_FragColor = vec4(vv3color.r,0,0,1.0)`，綠色和藍色就以此類推，只是放在`vec4`的位置不一樣而已。

EX:



- **How to operate my program**

- ① 開啟sln檔案後，compile完成後，按h即有指令列表。

- **problems you met Other efforts you have done**

- ① 一開始在讀檔時，因為沒有考慮到OBJ->numvertices的個數並不是真正總vertices的數量，所以畫出來的徒刑並不完整，後來考慮到後才解決這個問題。
 - ② 因為剛嘅使接觸opengl，所以對shader不太了解，並且也不知道uniform如何使用，之後上網查詢後，才解決uniform傳輸的問題，並成功讓model有RGB三種顏色。