Computer Graphics – HW5 : Views && Perspective

14 數字媒體

14331075 鞏持茵

**1. Introduction**

So far we have learn how to move the objects in the virtual world, but it seems that we always observe the virtual world in a fix position. Hence, let’s move around and look around on this world! In this homework, you are required to manipulate the view/camera in OpenGL and to know the differences and correlation between this homework and homework6.

**2. Tasks**

**2.1 Pre-requirement**

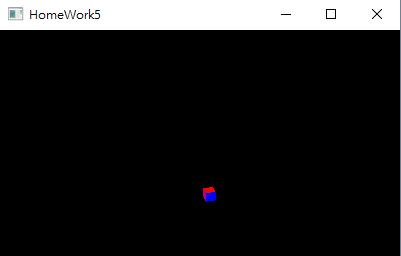
* Language: Only C/C++ is accepted in this homework.
* Libraries: Only OpenGL and freeGLUT/GLFW can be used.
* OS/IDE: Any OS/IDE can be used.
* The cube you’ve drawn in homework4 is needed. Each face of the cube should be colored differently. The size of the cube should be 0.1x0.1x0.1.

**2.2 Projections**

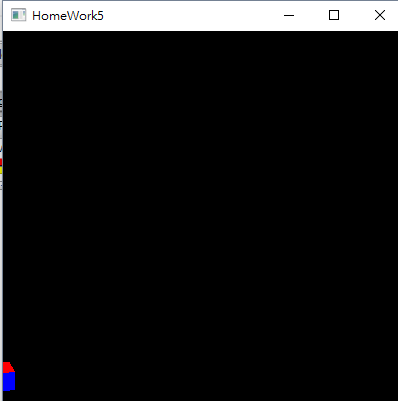
Projection define how a virtual 3D world projects into a 2D screen such that we can see it. There are 2 kinds of projections are used in OpenGL: perspective projection and orthographic projection.

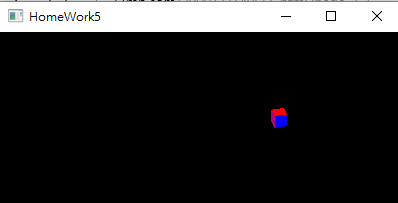
* Place the cube in (0.5,-0.5,-1.5) and create a orthographic projection. Show what you get in the report. Try different parameters(i.e. left, right, bottom, top, near, far), and write down how these parameters affect result in the report.







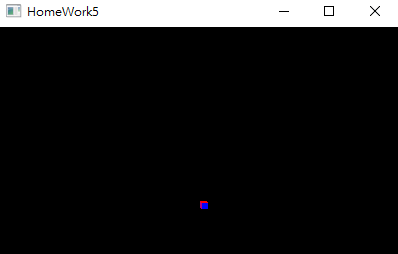




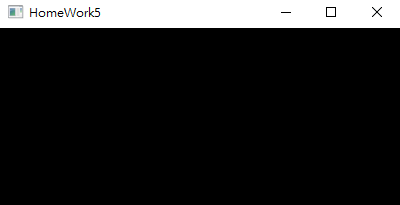
當使參數left變正數之後，取的映射範圍向右移，因此立方體向左移。當right值變小，則立方體向右移。Near和far的參數則必須要取在特定數值，不然會顯示不出圖形。而在正交投影的部分，不管他far取多遠，投影出來的大小依舊會是一樣的，不會有遠近分別。

* Place the cube in (0.5,-0.5,-1.5) and create a perspective projection. Show what you get in the report. Try different parameters, and write down how these parameters affect result in the report.

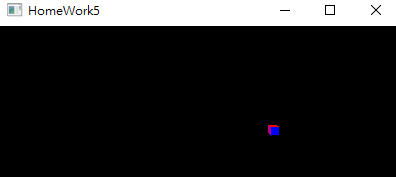












* Use exactly same parameters for 2 projections and discuss the differences of 2 results. Post the result in your report and discuss why.

使用一樣參數的情況下，透視投影的立方體比正交投影的要小，不過因為立方體很小的緣故，所以其實並不是可以很好的看出區別。而位移方向的部分和正交投影是一樣的趨勢。透視映射的時候可以看出矩形各面的大小有根據距離而有所不同，根據z軸比較遠的部分較近的部分還要小

。而我原本使用的函數則是向下延伸的，感覺有點奇怪。

**2.3 View Changing**

You are required to change the view of the camera to observe the cube.

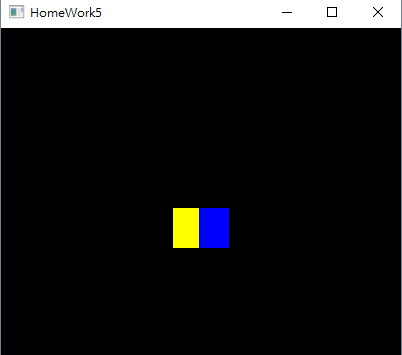
* Place the cube in (0,0,0), use the perspective projection. Import glu.h, try gluLookat(...) function. Let your camera circle around the cube, but looking at the cube all the time.
* Show the results and write down your main algrithm in the report.
* In practice, the camera setting is actually stored in a matrix called View matrix. The transformations is also stored in a matrix called Model matrix. However, in OpenGL, these two matrices is actually one matrix called ModelView matrix. Discuss why it is so in the report. (Hints: Is there an actual camera object in virtual world? If there is not, how to pretend we have one?)

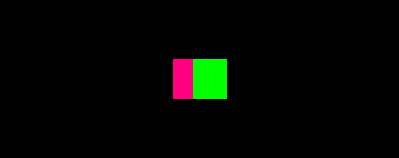
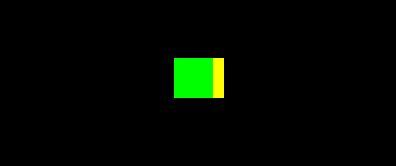
Hints: You may want to change your camera position in a circle. Try:

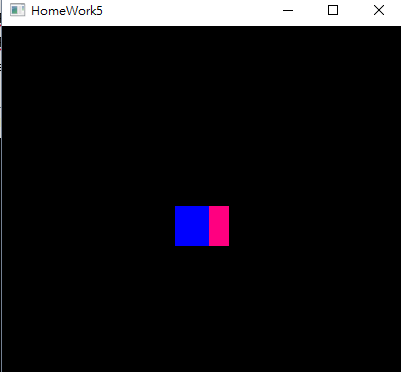
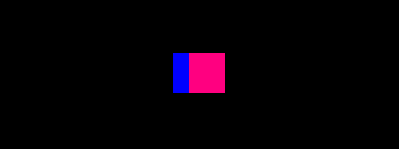
camPosX=sin(clock()/1000.0)\*Radius;

camPosZ=cos(clock()/1000.0)\*Radius;

Since sin(x)^2+cos(x)^2=1, that make sure we form a circle in XoZ plane.







和直接的透視投影其實沒有很大的區別，最大的不同就是使用函數並將視角的部分用變數來進行更改，以達到旋轉改變視角的目的。所以在這個函數裡，矩形會在xz平面上旋轉，雖然都是旋轉的樣子但我們這次是改變視角來達成目的，而上次則是改變立方體本身。Radius的部分因為我不是很確定實際到底要用怎樣的數，所以我就隨意的設了0.8。在虛擬世界並沒有一個相機對象，所以我們使用gluLookAt來設一個視角模擬相機，就可以自己決定要從哪個角度看我們的圖形，而將兩個矩陣合成一個應該只是希望節省運算而已吧。