**CS3570 Introduction to Multimedia**

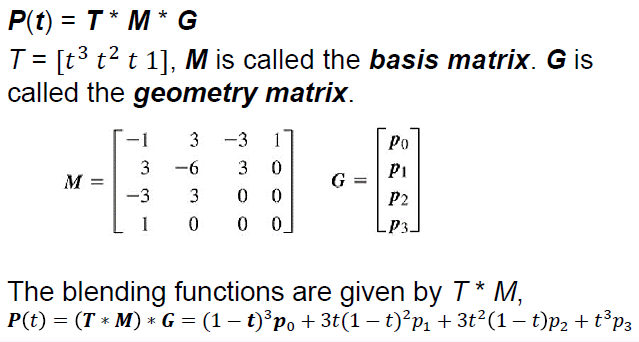
**Homework #4 Report**

103062234 張克齊

**PartI. Bezier Curve**

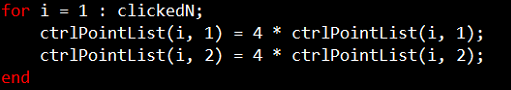
**[實作方法]**

在助教提供的code中幫我們寫好了標點的方法，因此主要實作的就只剩Bezier Curve的演算法；在spec中有提到，我們實作的方式為一次取4個點(3rd-order)，要注意的是起始點也是終點，並且每次取的最後一點也是下一次取的第一個點，因此若要取10個點，則需要在圖中標9個點，順序會是1234, 4567, 7891，這樣我們就可以依據下圖的公式進行實作。



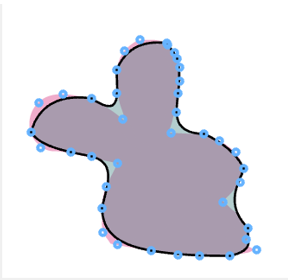
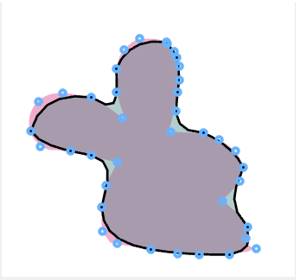
運算Bezier的部分我寫在myBezier.m中，要傳入的有標得點的X和Y座標、標的點數以及LoD(圖中的t)，算法如同圖中最後一行，要小心的只有當最後一段要回到起始點時的判斷，這邊因為題目是要求39個點以及99個點，也就是最後一段都會是XXX1，因此我沒有用mod進行處理而是判斷最後一段，這樣就能達到Bezier curve的效果。

(b)小題的意思是要比較直接放大的結果跟放大每個點(x, y)再去做Bezier得到的結果有甚麼差異，因此可以利用以Nearest neighbor interpolation放大四倍，並利用下圖的方式將點放大四倍後再丟入myBezier()中，得到的結果覆蓋在放大的圖上，而最終我們可以透過line()畫線比較兩者間的差異。

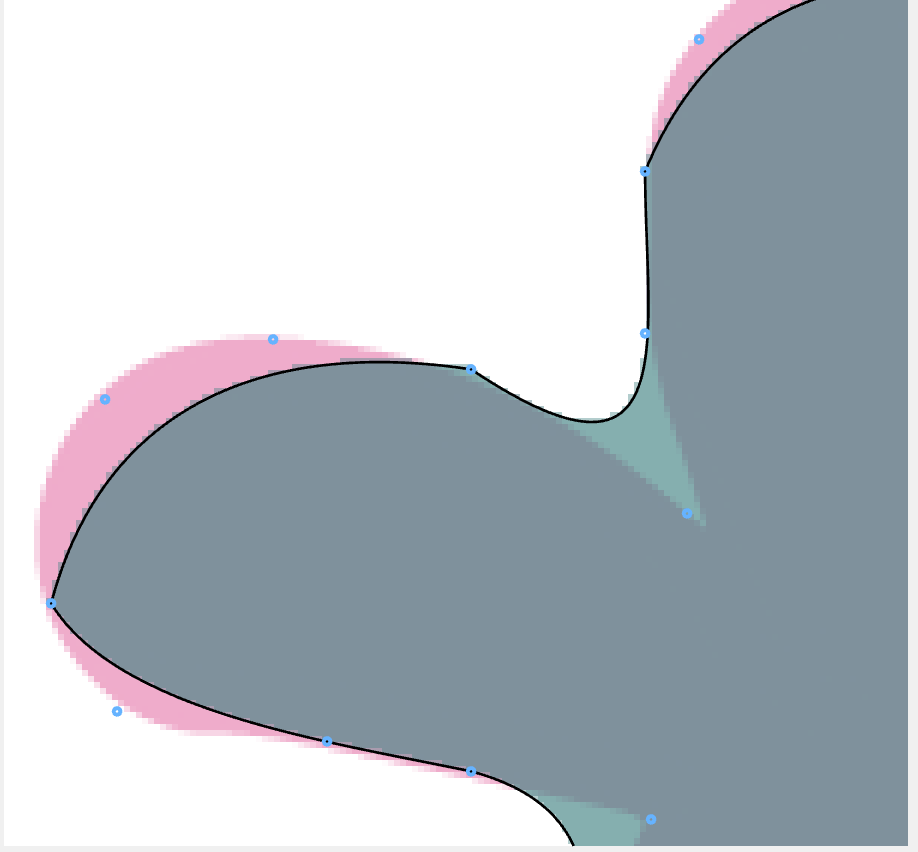
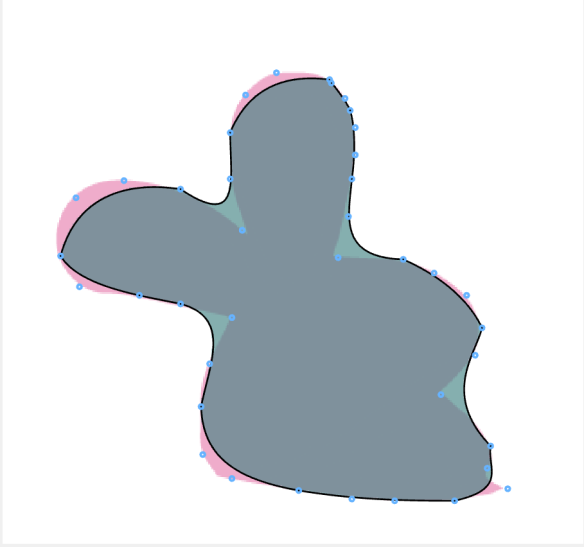


[結果圖]

**40 points**

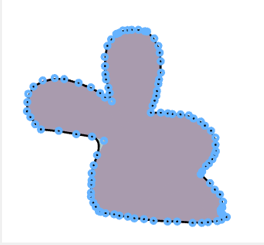
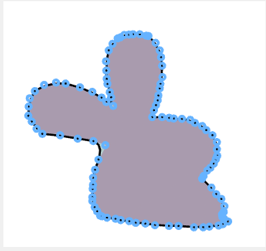


Original LoD Low LoD High

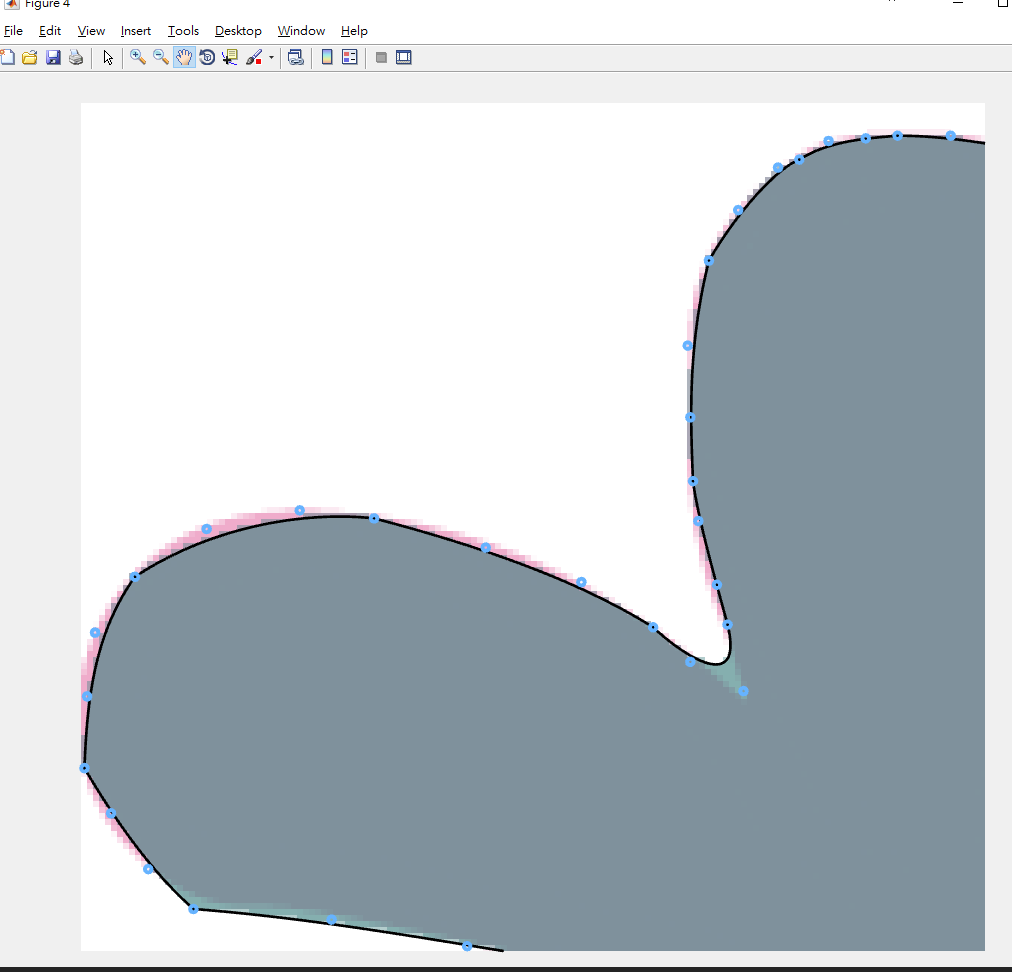
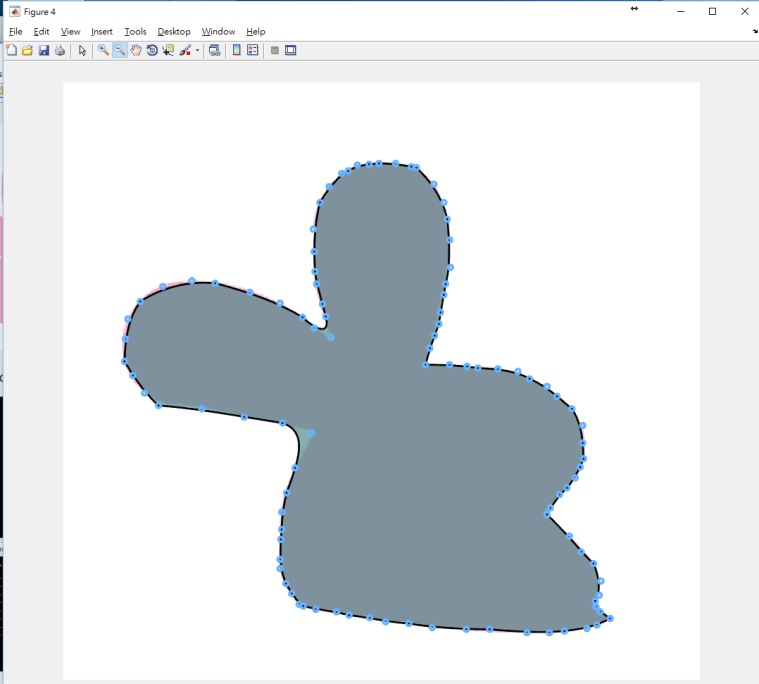


Resized by Nearest Neighbor(NN)

**100 points**



Original LoD Low LoD High

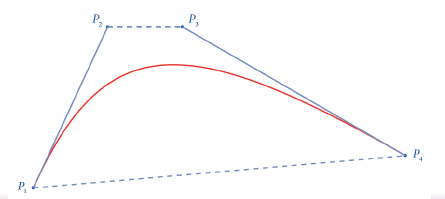


Resized by Nearest Neighbor(NN)

[比較與結論]

1. Sampling rate的比較:

因為在實作Bezier curve中，取的四個點中只會保留起點和終點，中間兩點會被調整彎曲，如下圖所示；因此在取的點中(假設等距離)，當點數愈多代表愈密集，彎曲程度就比較小，更能貼近圖片的外圍，如上面結果圖比較可明顯發現。



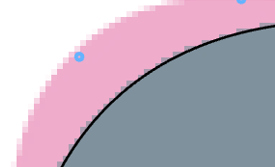
2. Levels of details(LoD)的比較:

LoD的意義可以代表著在取的四個點中，要把他轉化成多少個點，像是LoD = {0, 0.2, 0.4, 0.6, 0.8, 1}，就是把4個點轉換成6個等分的點，其中0跟1是起始跟終點，因此我們也可以推測出當LoD的間隔愈小(轉換的點愈多)，出來的曲線會更加平滑，從取40 points的兩張結果可以明顯發現Low比較不平滑。



3. (b)的比較：

從結果圖來看，我們可以發現直接放大的會有很明顯的顆粒階梯感，那是因為我們是以pixel為單位放大4倍，因此在做內插時會使得某些不在line上的pixel對應到原本圖片line上的pixel，因此我們也了解到應該是要對點放大重新計算Bezier curve而非放大圖片才能得到比較精確的結果。

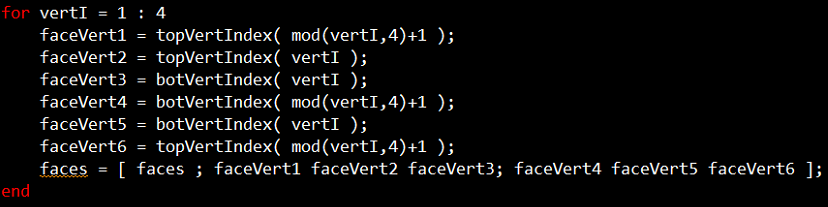
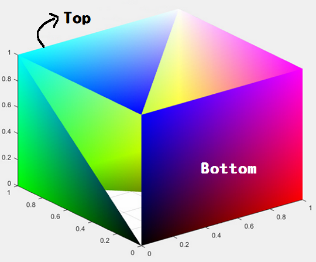


**PartII. 3D Models**

**[實作方法]**

1. RGB cube:

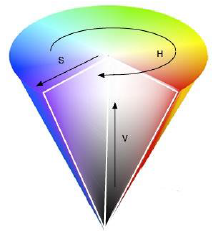
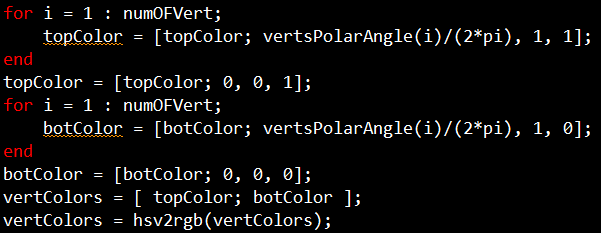
在執行助教附的code時，會發現有些三角形沒有被畫到，如下圖所示；這就讓我知道是face的地方寫的不完全，並且從圖中可以觀察到缺的規律是2個Bottom的vertex和1個Top的vertex圍成的三角形，所以我只要在同一個for迴圈加入這樣的face就能得到完整的RGB cube。



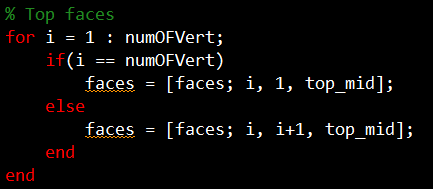
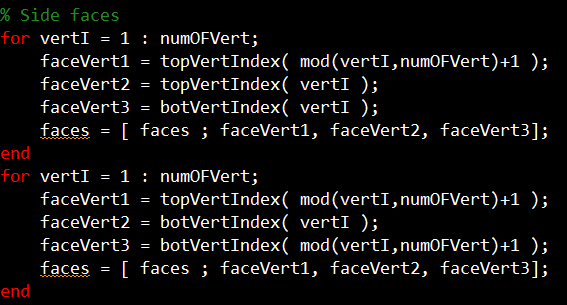
2. HSV cylinder:

一開始先用supplement教的方法做出角度及XY座標並存起來，這邊取的一個圓外圍的點數為60個，也就是每6度一個點；接著我先定義Top為上面的圓，Bottom為下面的圓，而vertex的index規律定義為1~60是上面圓的外圍點，61為上面圓的圓心，62~121為下面圓的外圍點，122為下面圓的圓心，這樣的設定能讓下面計算的for迴圈都是從1到60；我使用兩個變數和代表上下圓的圓心，並記錄每一點的座標。

再來我們要來定義HSV的color，這邊可以參考第二章講義提到的HSV color model，如下圖所示。H、S、V的值範圍都是0~1，圖中箭頭方向都是從0指向1，因此我們可以實作出每個vertex對應的(H, S, V)，可對應到下面的code。最後，因為螢幕的顯示是RGB(0~255)，因此我們可以透過matlab的函式hsv2rgb()將其轉至RGB color model。

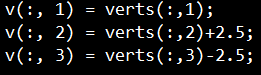
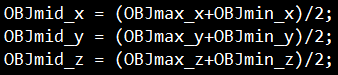
 

最後就是跟RGB cube一樣要找出每一個要畫的face，我的順序是Top到Side到Bottom，方式跟做RGB cube一樣，只要記得判斷Top跟Bottom是否繞了一圈就好，如下圖所示。這樣，就能完成HSV cylinder。

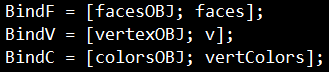
 

3. 平移與結合:

首先考慮平移，對於HSV cylinder來說，因為我們本來的中心就已經是(0, 0, 0)，所以我們可以直接將所有vertex平移到想要的位置，如下左圖，代表將圓柱平移到中心為(0, 2.5, -2.5)；但對於讀進來的OBJ我們就不能直接這樣做，原因是他的中心並非是原點，所以處理時要先分別找出他的x, y, z軸的最大最小值相加除以2，找的方式為跑過所有值並開變數刷新存取，這樣才可以得到我們OBJ當前中心位置，就能透過跟剛剛一樣的平移方式將其移至原點，如下右圖所示。

至於結合的部分，照著助教的提示就是把圓柱和OBJ的vertices, colors以及faces結合在一個matrix中，其中要特別注意faces的index問題，因為在分開時兩者的編號都是從1開始，但結合後我們應該讓加在後面的index做平移，我是將OBJ加在後面，所以是，這樣就能透過下圖的方式結合將兩個物體一起顯示。



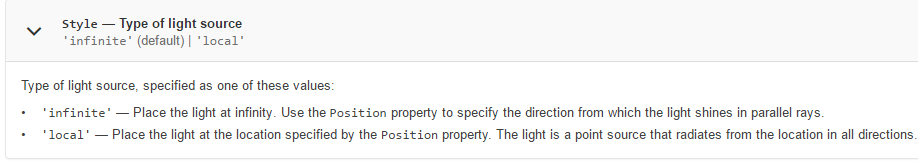
4. Lighting(每個figure可以給他一個新的光源)

(1) 光的種類:

這邊要實作兩種光源position light(point light)和directional light，方法是依據matlab官方文件，只要在light()函式中加入’Style’參數，其中local為position light而infinite為directional light，這樣就能打出兩種光並進行比較。



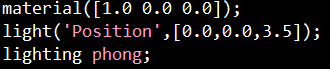




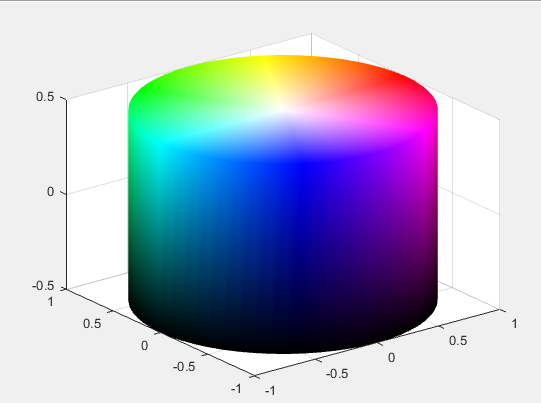
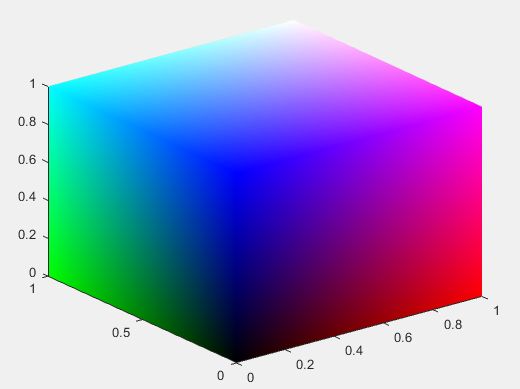
(2) 光的三元素:

這邊先討論作法，只要運用官方文件提到的material([ka ks kd])函式，就可以給題目要求不同的值來比較結果。

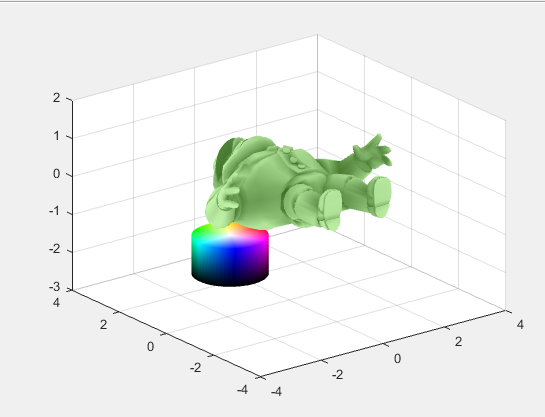




[結果圖]

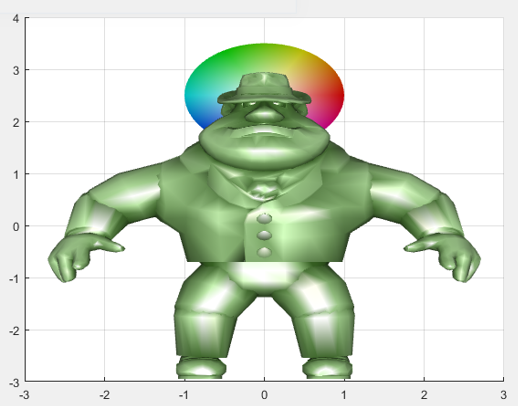
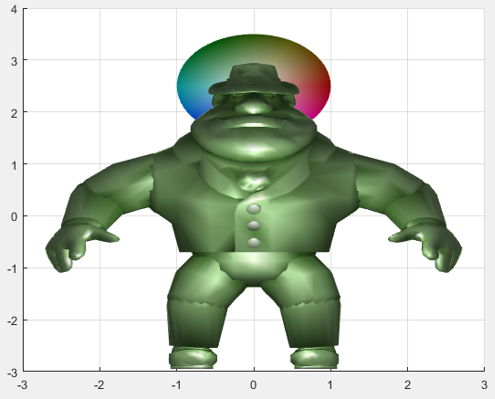


RGB Cube HSV Cylinder



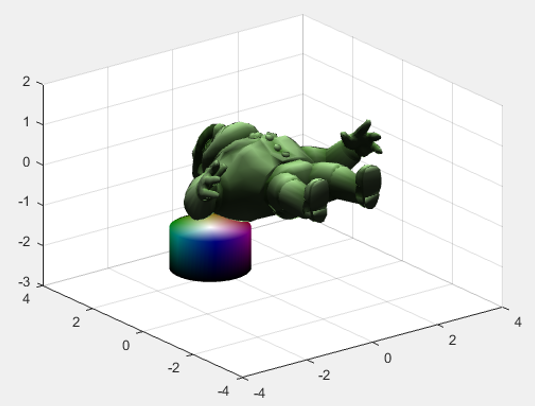
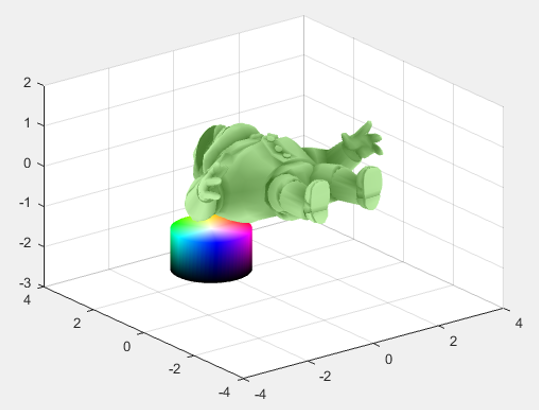
Original combination

Light Source

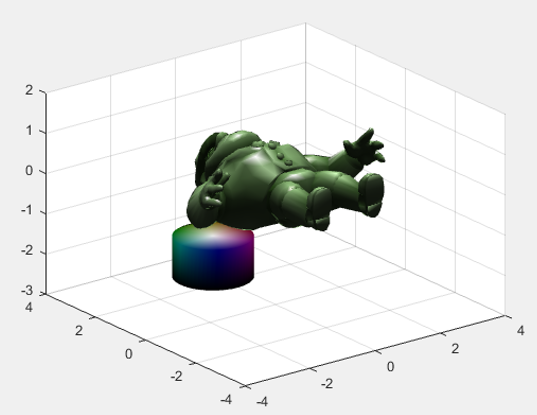
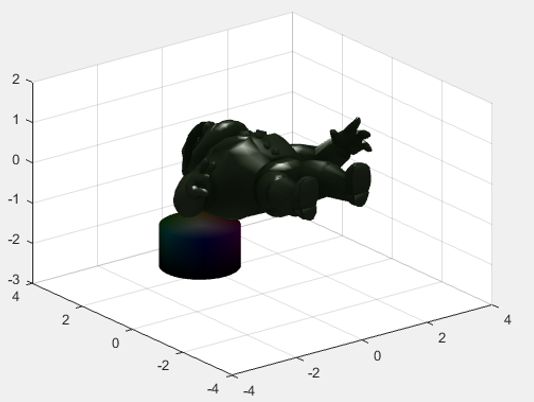


Position(Point) light Directional light

[ka, kd, ks]



[1.0, 0.0, 0.0] [0.1, 1.0, 0.0]

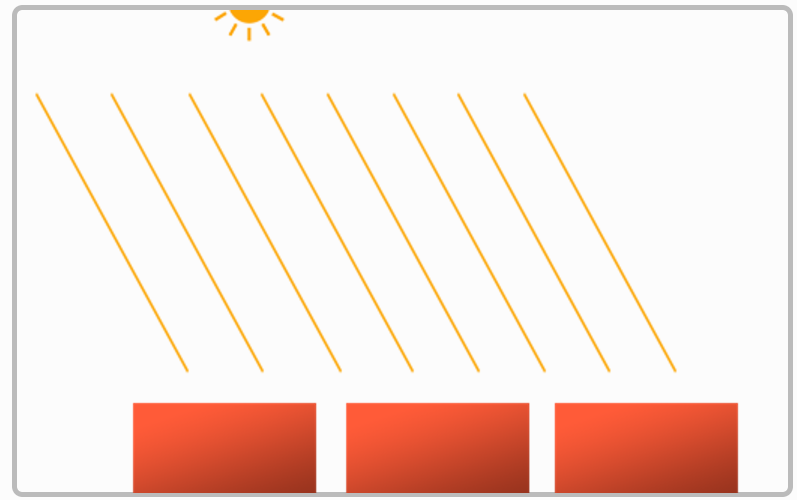
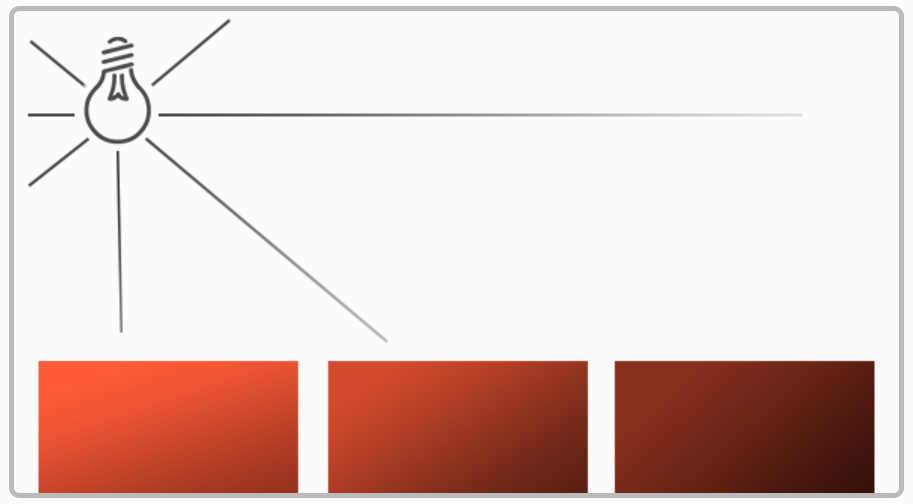


[0.1, 0.1, 1.0] [0.0, 0.8, 1.0]

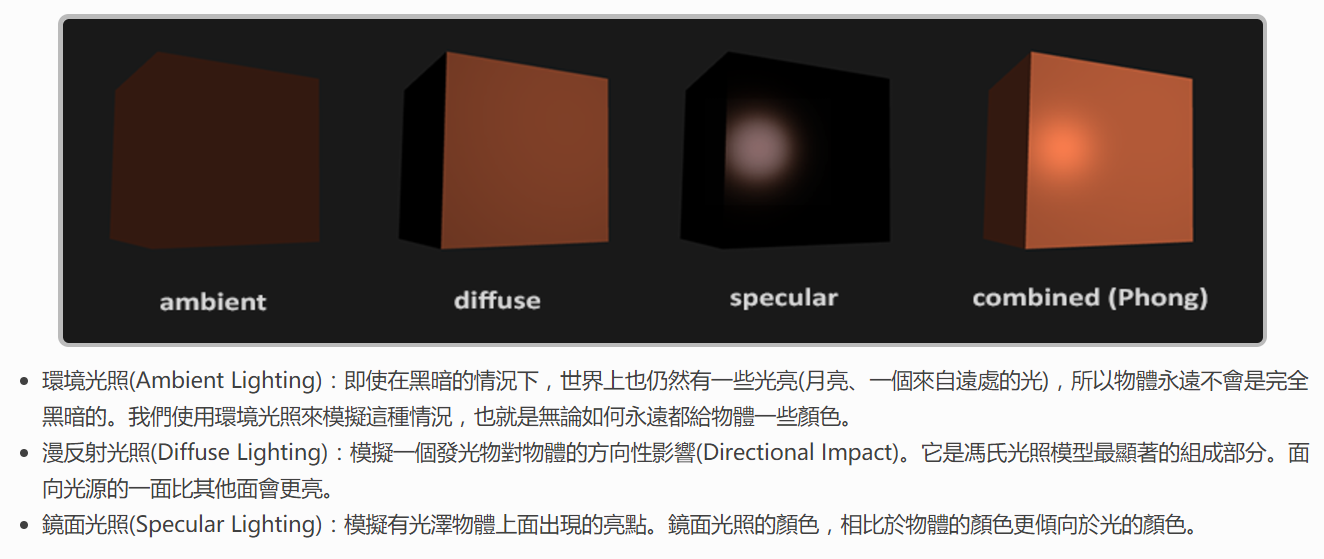
[比較與結論]

1. 討論不同種類光的結果:

Position light又稱為point light，代表著一個有位置的光源向所有方向發光，且光的強度會隨著距離而減低(變暗)，可以想成燈泡的效果，如下左圖；Directional light代表著假設光源很遠，那會變成一道道平行光，並來自於同一方向，最簡單的聯想為太陽，如下右圖。從結果圖可以顯示兩者的差異，我們可以很明顯發現若為position light，物體會有陰影，而平行光則是每個角度都會發亮。



2. 光的三元素討論:



Phong lighting model主要是由上述三元素所組成，他們的概念如同參考網站上所述，這邊分析題目中給的四組職所對應的結果圖

I. (ka , kd , ks) = (1.0, 0.0, 0.0):

代表只存在環境光，透過環境光可以使我們看到物體的每個地方，也就是無論光源在哪裡整個物體都有光打到。

II. (ka , kd , ks) = (0.1, 1.0, 0.0):

代表主要受到反射光的影響，所以我們看到的物體會只有光源打到物體反射的部分，因此會有陰影的感覺。

III. (ka , kd , ks) = (0.1, 0.1, 1.0):

代表主要為鏡面光，也就是只會顯示光源反射到我們眼睛的光，產生像是光澤的效果，其他部分若設為0則會呈現黑色。

IV. (ka , kd , ks) = (0.1, 0.8, 1.0):

代表受到反射光和鏡面光的影響，因此會呈現II跟III結合的效果，也就是有陰影及光澤的結果圖。

*參考網站: https://learnopengl-cn.readthedocs.io/zh/latest/02%20Lighting/02%20Basic%20Lighting/*