

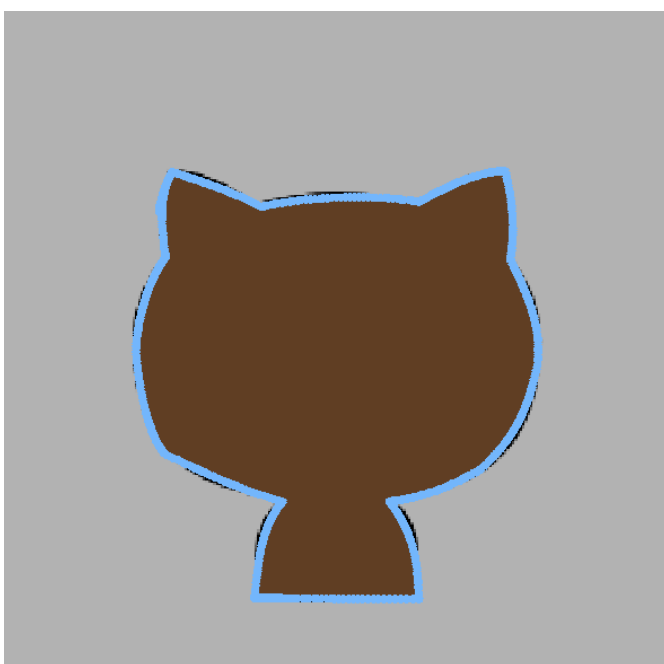
# Homework #2 Report

102062111 林致民

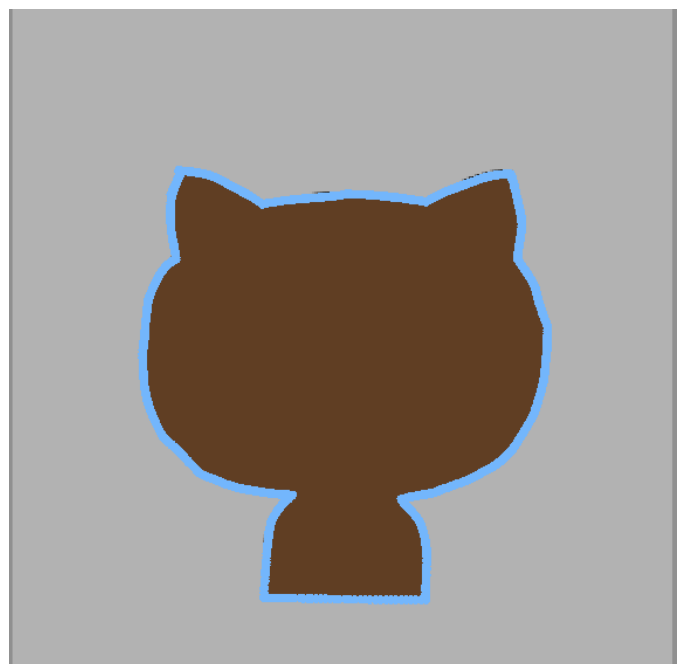
## Part 1(a):

第一題要我們畫的是貝茲曲線，題目的要求是，我們要用貝茲曲線把那個貓圈起來（目前沒有對貓過敏），每四個點做一次貝茲曲線的Equation，例如有 1 2 3 4 5 6 7 8 9 個點，那麼分別對(1, 2, 3, 4) & (4, 5, 6, 7) & (7, 8, 9, 1) 各做一次。那要怎麼計算呢？首先，先把矩陣M建立出來  $M = \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$ ，然後使用  $sample = linspace(0, 1, LOD)$  來控制我的level of detail，也就是每一個線段他取樣的頻率，再來要找出矩陣  $T = [t^3, t^2, t^1, 1]$ ，當然這個t我是用syms t暫時用變數取代，看sample的當前數字是什麼，把它帶進去便是，最後把三個矩陣乘起來  $(T * M) * G$ ，可是這個矩陣在matlab中乘完，我發現到他會是用分數來表示，所以為了從分數轉換成小數，就使用了 `double(vpa(result))`。

然後根據用滑鼠點選圖片取樣點個數的不同，圍出來效果的差別：可以明顯發現點的比較多的可以圍的比較完整：D

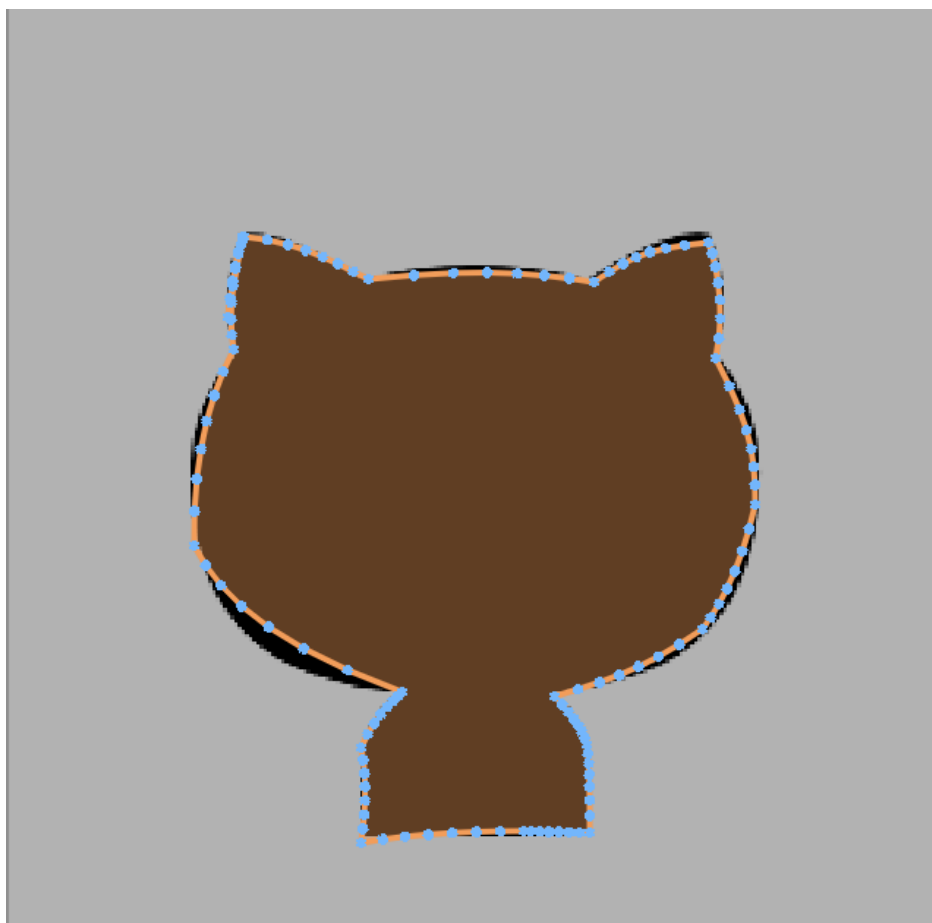


點了43下



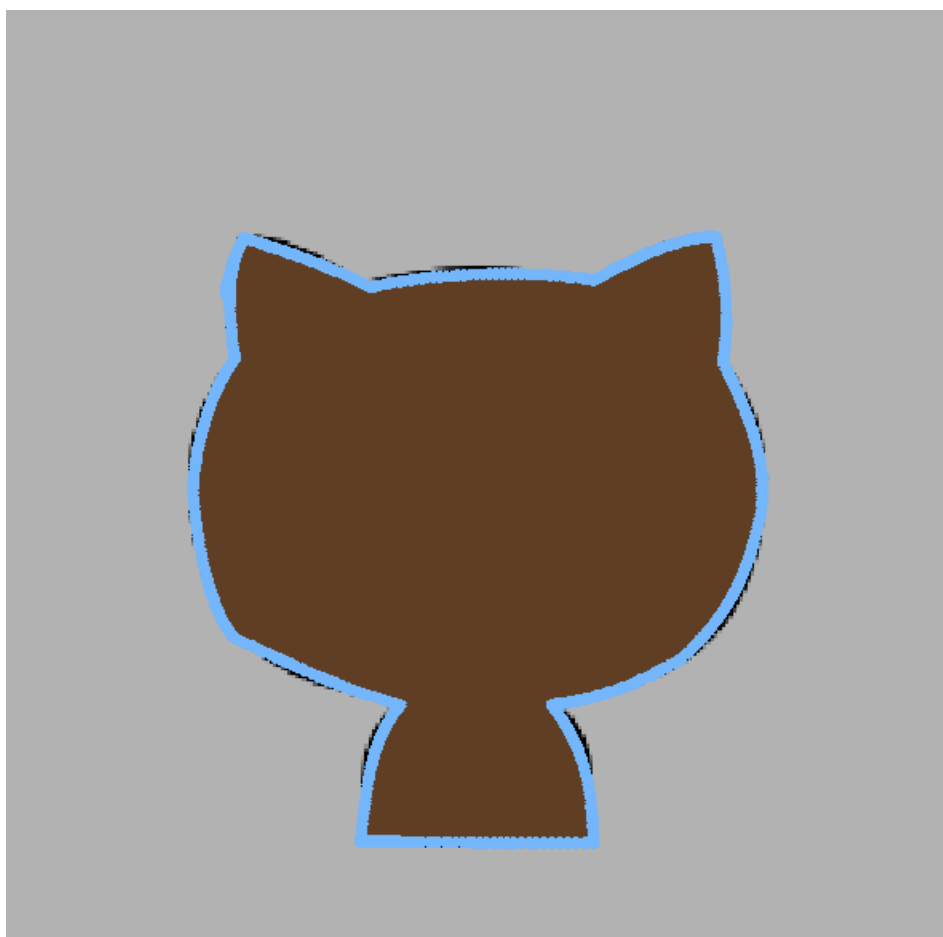
點了97下

根據一個線段取樣的個數（L O D）：



LOD = 7

稀疏

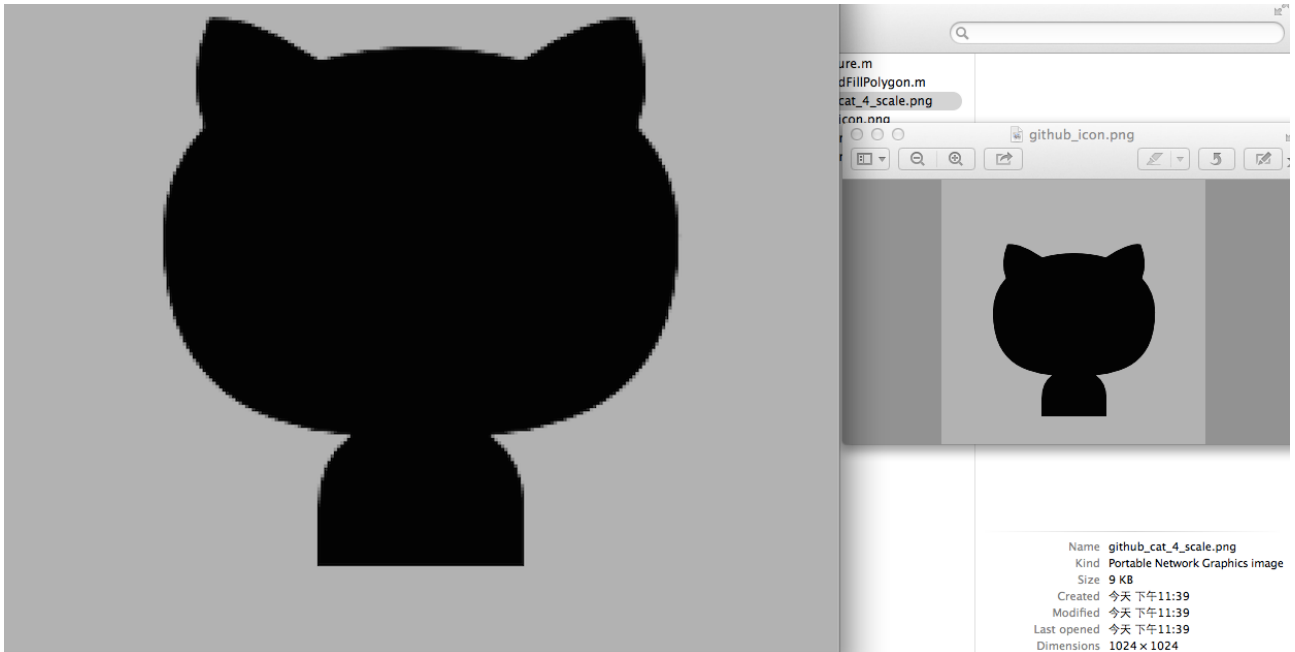


LOD = 40

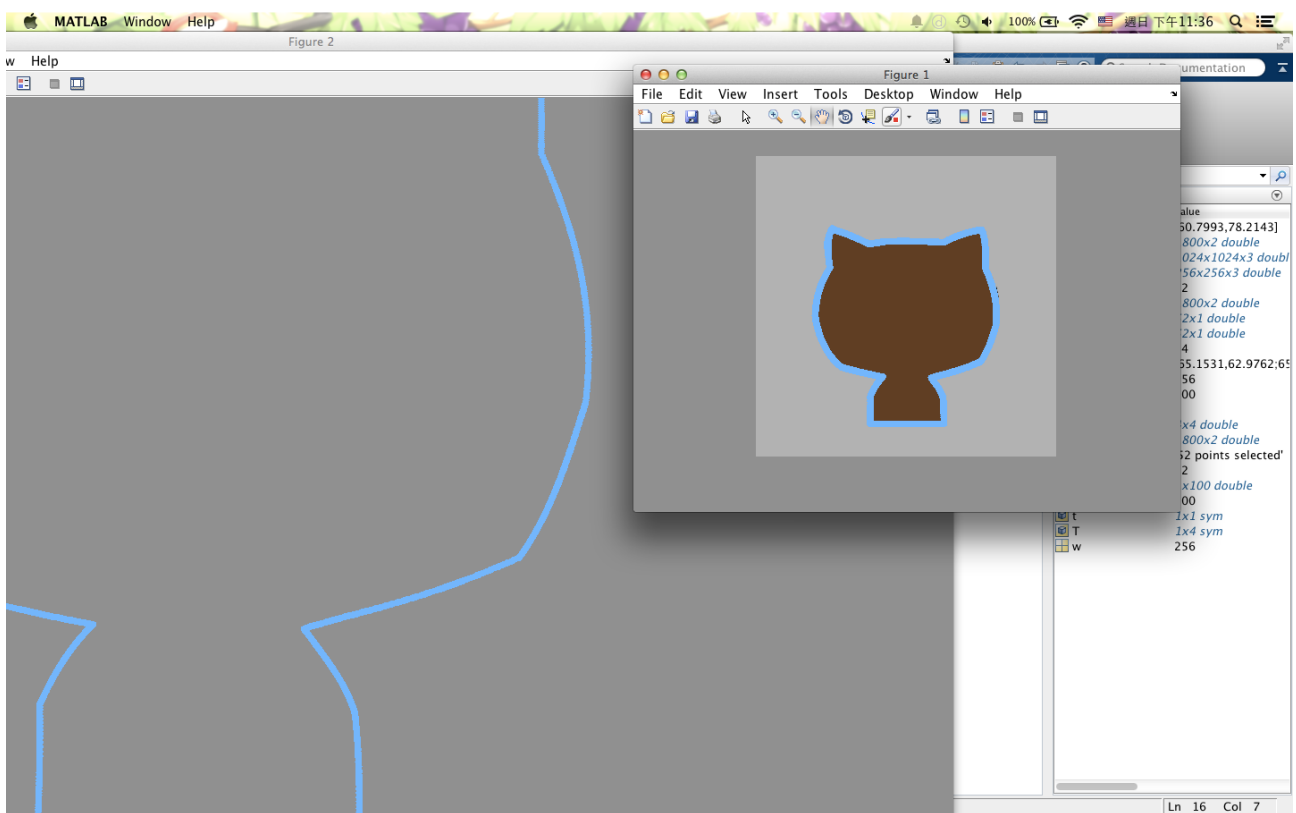
密集

part 1(b)

針對向量圖還有點陣圖放大四倍做比較：



點陣圖使用  $N \times N$  interpolation 的 `imresize()` 邊緣會有明顯鋸齒狀

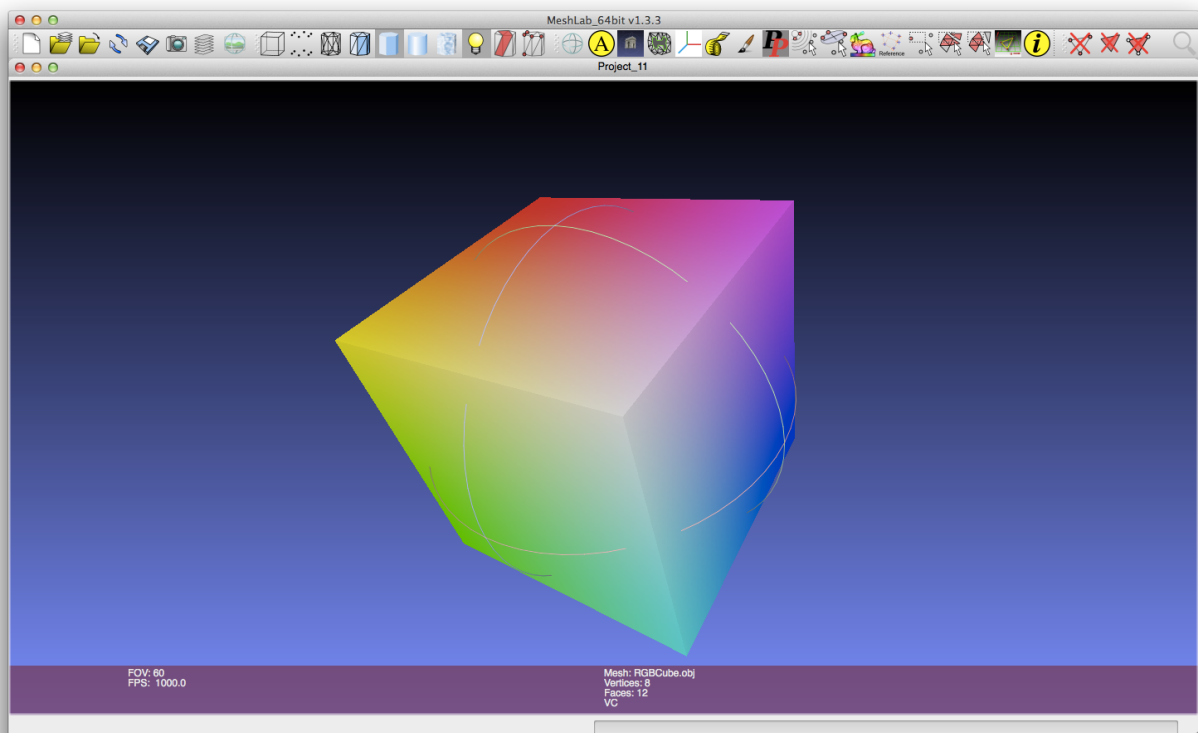


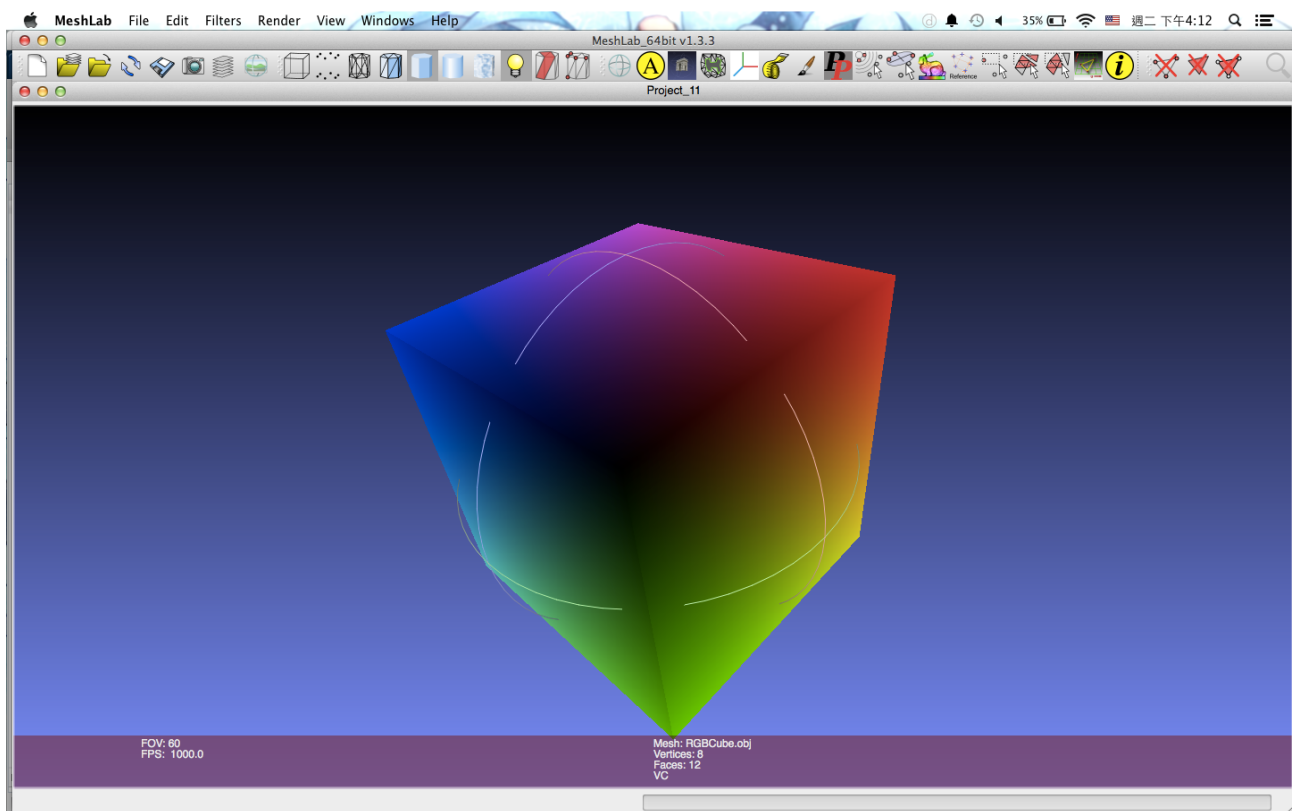
直接把向量的坐標點放大四倍的話，看不出像是Bitmap的鋸齒狀，誤差上用肉眼比較看不出來。

向量圖的放大比較方便，我只要把算好的結果（所有坐標點），全部乘以四就可以了，而Bitmap 的放大直接call `imresize(catImage, 4, 'nearest')` 這個API。

## Part 2(a)

這題的要求只是把正方體做好並上色而已，由於這學期圖學那門課剛好有學到，所以做這個比較沒壓力就是了。我的做法就只是把所有的點補齊，然後mapping index到正方體八個腳的坐標點，並且讓定義角點的顏色(rgb表示)最大值，(1 0 0), (0 1 0), (1 1 0)... 等。然後face的部分則是把一個面切成很多個三角形，GPU在繪製圖形不管幾邊型，都是以三角形的方式處理，四邊形就用兩個三角形組成，五邊形就用三個三角形組成，原型的話就用多個三角形繞著中心點來畫三角形，看要取樣到『多邊』形才會讓我們的眼睛看起來像是圓形，這個在(b)的部分會談到。這題只要把剩下缺的三角形補完，然後把八個角的顏色最大值定義好，就可以了，以下是結果：





不過有個奇怪的狀況是，如果有些三角形以順時針，有些三角形以逆時針的方式繪製的話，整個方塊的顏色會看起來不太對勁，方塊的顏色會偏黑，目前不知道是怎麼回事，不過我有把Camera伸進去正常的方塊的內部裡面看，顏色差不多就是長那樣（暗暗的）。

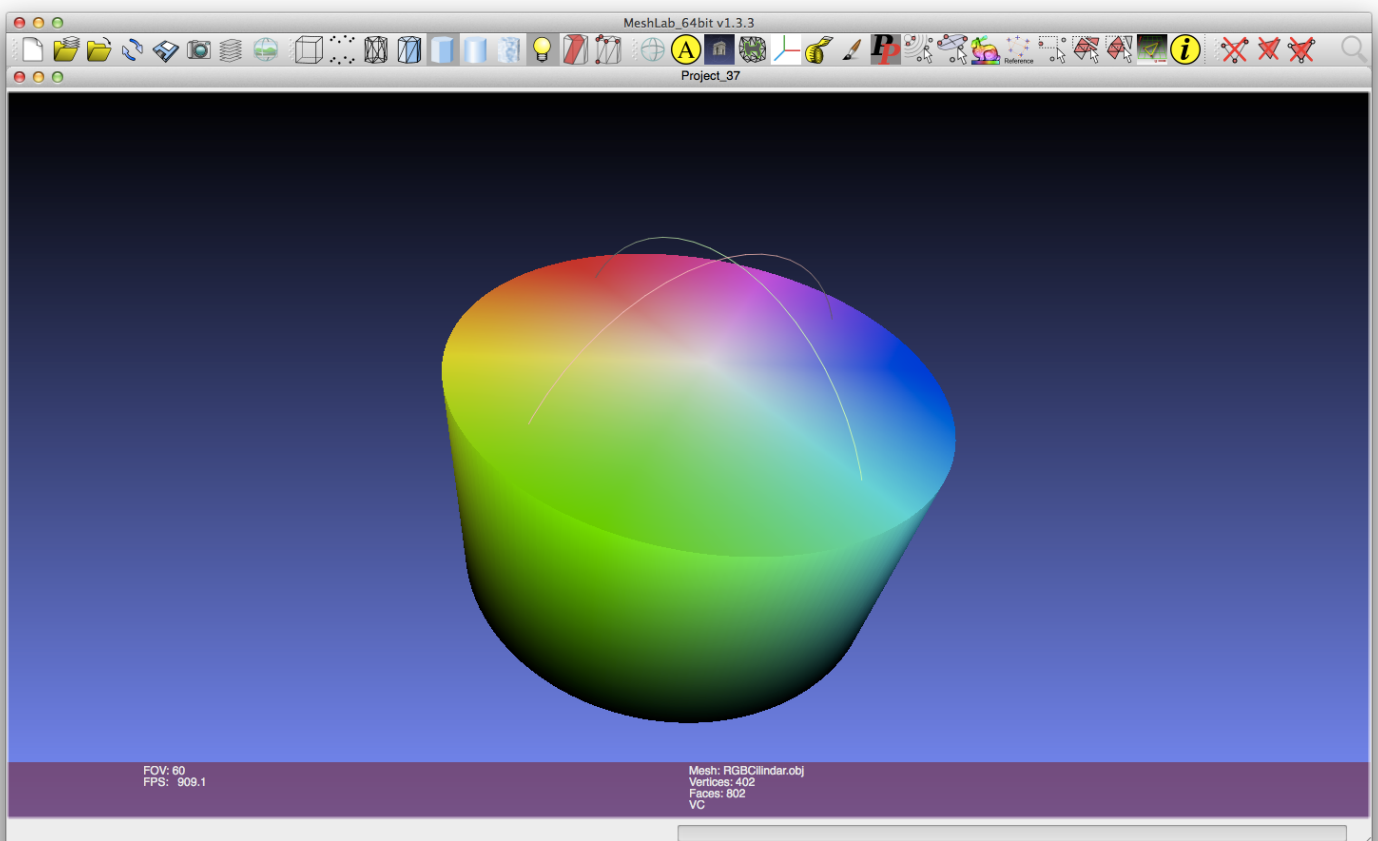
## Part 2(b)

這一題有點有趣，要我們畫出HSV顏色表示的圓柱圖形，我的做法是，在(a)有提到的，把圓形切成很多個三角形，然後以中心點為所有邊的三角形頂點，在頂面和底面個弄出一個三角形。然後圓柱的邊邊則是利用『兩個三角形可以組成四邊形』的特性，把這個四邊形弄得很細很細，讓肉眼看不太出來有邊的存在，把圓柱的邊畫出來。顏色的部分則是用以下方式定義：

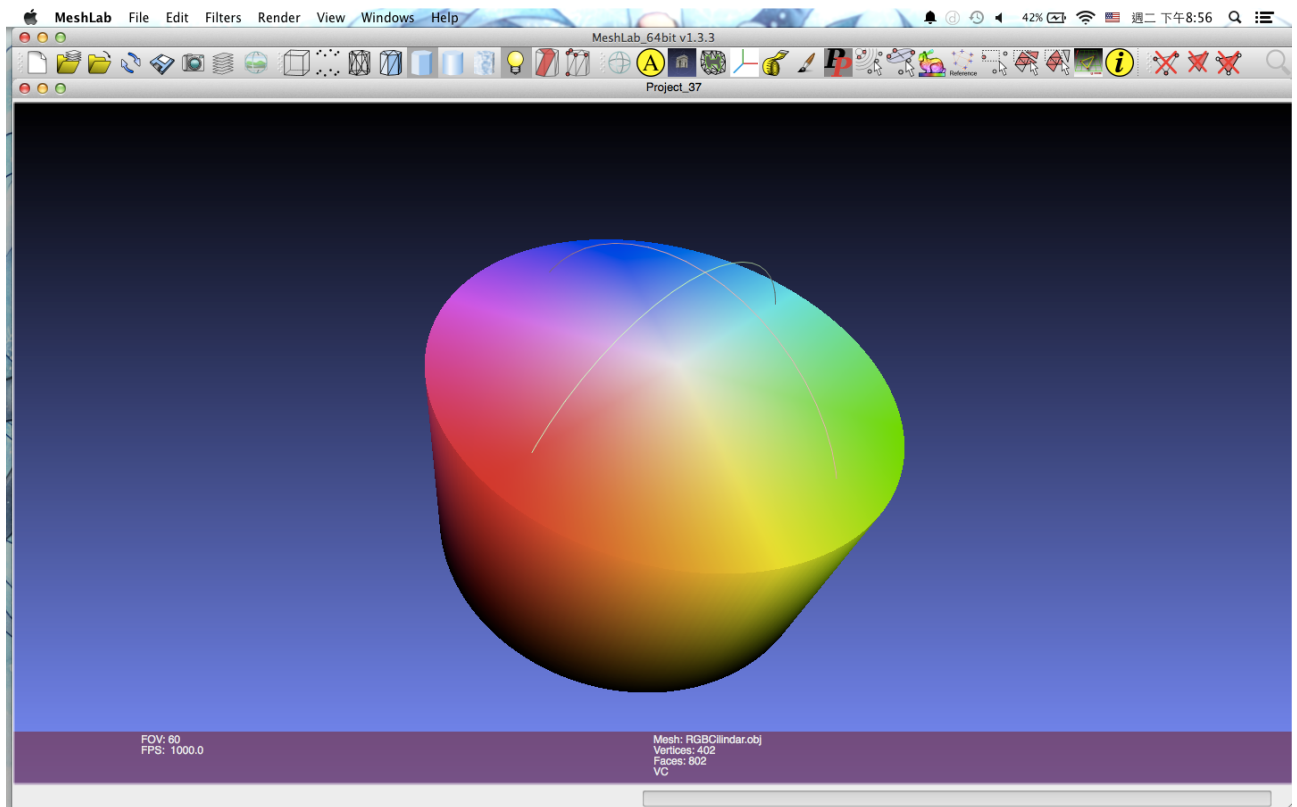
1. 由於坐標點只存在於頂面、底面上而已，我只要定義這兩個遍的坐標就可以了，並且跟(a)一樣，坐標和index都對應上去。

2. 以頂面、底面來看，中心點的顏色飽和度最低，最接近『超多』多邊形（圓形）的邊邊顏色飽和度比較高，所以只要定義中心點飽和度 = 0，邊邊飽和度 = 1就可以了。
3. 底面的亮度為0，頂面的亮度為1，所以任何一點出現在頂面的坐標點，就直接把它的亮度定義為1，反之則定義為0。
4. 顏色則和頂面還是底面無關，邊邊坐標點的定義就是以極坐標轉換平面坐標，顏色則是讓0~1切成『邊數 + 1』等份的方式定義。
5. 從HSV的顏色轉換成RGB，API : `hsv2rgb(colors)`

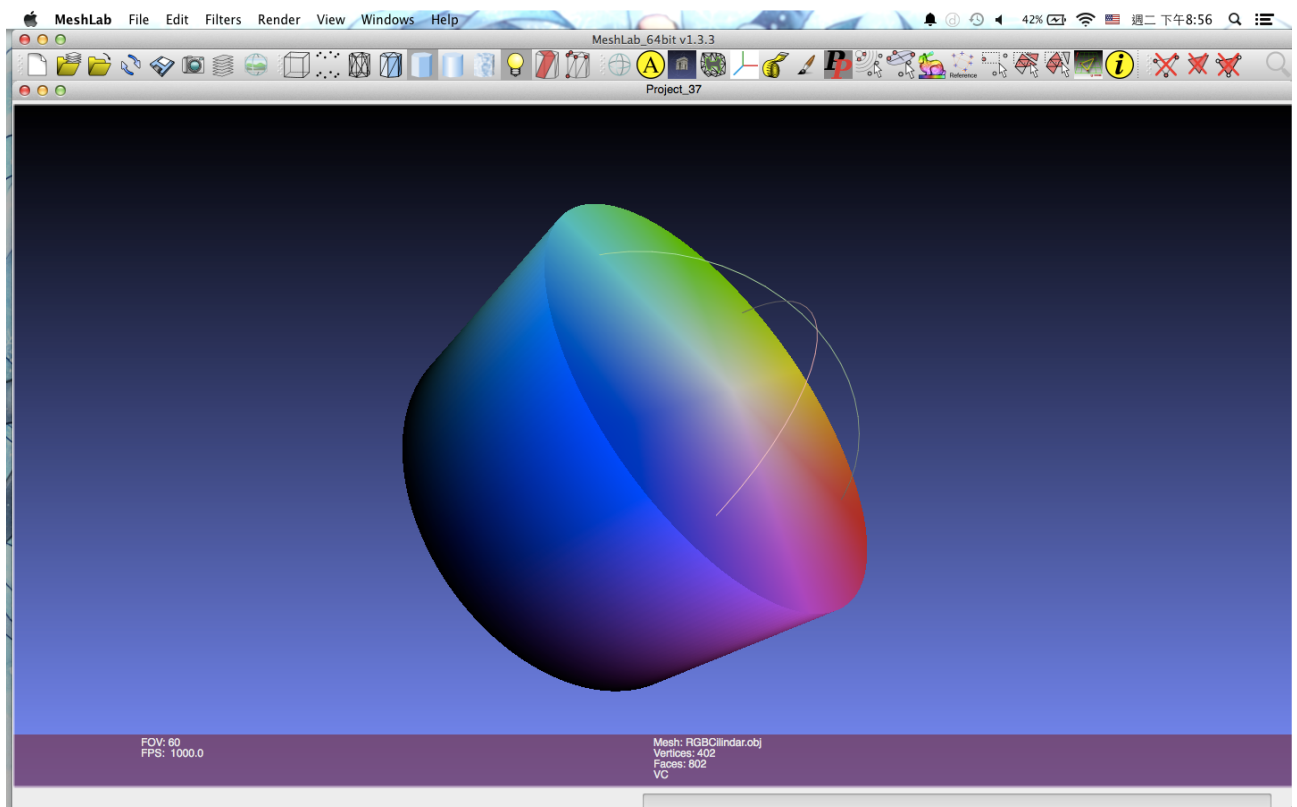
以下是畫出來的結果：



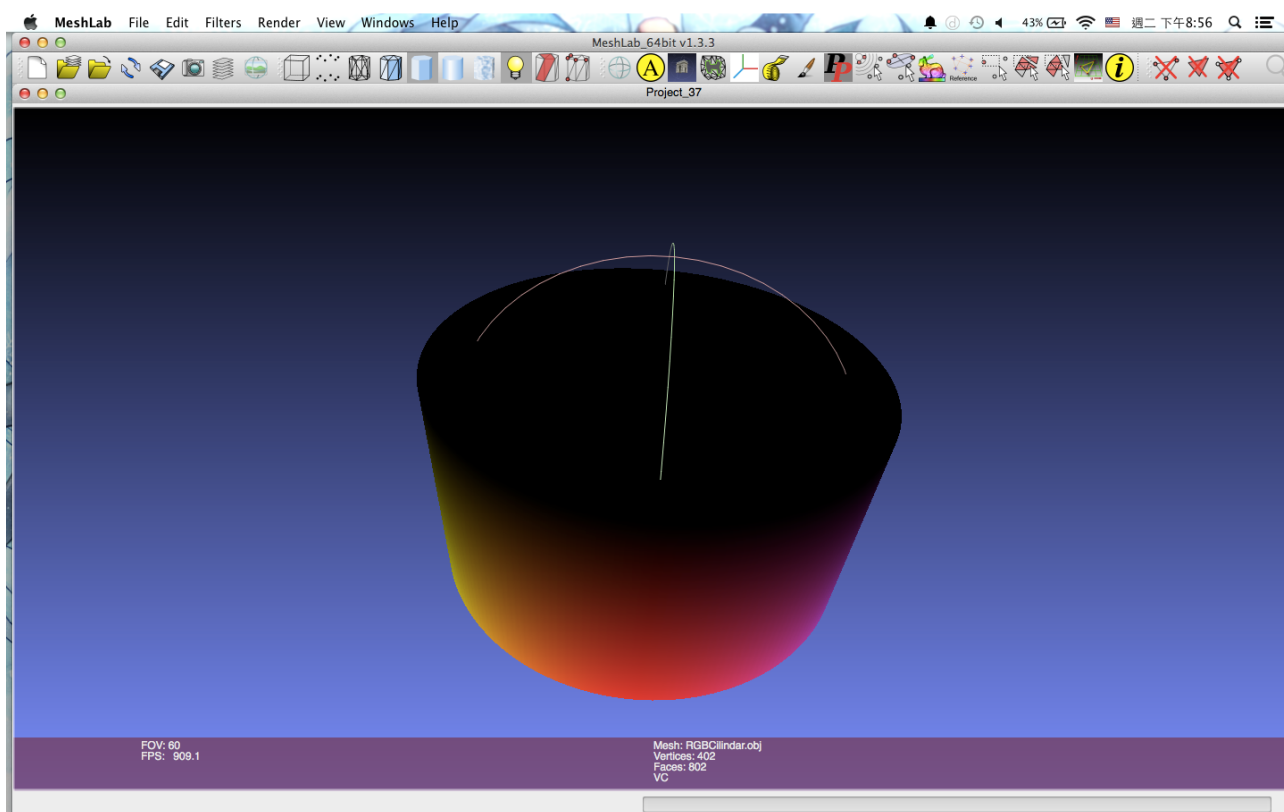
不同的角度看圓柱



不同的角度看圆柱







底面確確實實是黑色

不過.obj file有個好處是，可以直接把檔案打開，然後看做出來的model對不對，以下是debug方式：

