

## Assignment 2

### Bezier Curve

404410030 資工二 鄭光宇

1. 方法概述：De Casteljau 演算法在只有 4 個控制點時，要產生一條貝茲曲線需要下面幾個步驟：
  - (1) 假設控制點編號為  $p_0, p_1, p_2, p_3$ ，  
 $p_0$  與  $p_1$  之間取中點，產生  $p_{01}$ ，  
 $p_1$  與  $p_2$  之間取中點，產生  $p_{12}$ ，  
 $p_2$  與  $p_3$  之間取中點，產生  $p_{23}$ 。
  - (2) 之後使用  $p_{01}$  與  $p_{12}$  產生  $p_{012}$ ，  
 $p_{12}$  與  $p_{23}$  產生  $p_{123}$ ，  
 $p_{012}$  與  $p_{123}$  的中點  $p_{0123}$  即為在貝茲曲線上的一點。
  - (3) 將  $p_0, p_{01}, p_{012}, p_{0123}$  做為左邊一組控制點， $p_{0123}, p_{123}, p_{23}, p_3$  為右邊一組控制點，兩邊用以上的方式遞迴下去，即可畫出整條貝茲曲線。
2. 程式實做方法：**bezier\_c.m** 為實做以上方法的自定義函數，遞迴的順序為 infix。輸入四個控制點

$$(\text{例如：} P = \begin{bmatrix} P_{0x} & P_{0y} \\ P_{1x} & P_{1y} \\ P_{2x} & P_{2y} \\ P_{3x} & P_{3y} \end{bmatrix})$$

與遞迴深度；輸出一個含有貝茲曲線上點的  $N \times 2$  矩陣，每一列有平面上 X,Y 座標，陣列的內容以從  $P_0$  出發到  $P_3$  的順序排列。

**main.m** 是這次作業要執行的檔案，它會隨機挑平面上四個點做為控制點，之後反覆呼叫 **bezier\_c** 函數產生平面上在貝茲曲線上的點，使用 **plot** 用點連接近似描出貝茲曲線，用 **scatter** 將曲線上的點標示出來，並且以呼叫 **bezier\_c** 時的遞迴深度為序，將結果輸出到 **image** 資料夾。最後近似的曲線會被存在 **image/Final.png**，不同遞迴深度產生的結果會被存到 **image/illustrate\_\*.jpg**，\*為呼叫時使用的遞迴深度。

3. 程式使用方法：執行 **main.m**，結果的圖片在 **image** 資料夾中。
4. 結果討論：如同預期，遞迴深度愈深，產生的點愈多，可以近似出更精確的貝茲曲線，但是遞迴深度太深時，程式執行的效率較差。
5. 問題討論：實做過程中沒有遇到太大的困難，唯一之前沒有學過的，是如何重疊兩張繪圖，重疊兩張以上繪圖的方式是 **hold on**，疊圖完後，**hold off** 就可以了。