臺灣二００六年國際科學展覽會

科　　別：電腦科學

作 品 名 稱：西爾平斯基船帆與掛毯圖形應用於數位圖形與數位音樂創作

學校 / 作者　：國立臺中第一高級中學　林自均

作者簡介

我叫林自均，興趣是電腦、作曲(MIDI)、打鼓、看書…等等，對於程式設計(C 語言、BASIC)、動畫製作(Flash)、音樂編輯(Cakewalk)都很有興趣，也有一些基礎，從小就覺得電腦很好玩，而資訊技能對於我的未來應該很有幫助。由於父母反對填鴨式的教學方法，管教的方式開放，讓我有了更多自我發揮的空間和自己的想法，不會輕易的拘泥於形式。在小學、國中時擔任資訊股長，負責班上網頁與畢業光碟，進入高中後很幸運的參加進階的資訊班，對 C 語言也越來越有把握，這都得歸功於老師的指導有方，最近參加「全球華人音樂創作比賽」、「宏碁數位創作獎」競賽，雖然最後結果尚未揭曉，但由於數位作品--「公車驚豔」初審入圍，使我得以受邀參與「宏碁數位創意營」，從不斷的資訊科技的參賽和研習活動中，我學習到許多珍貴的經驗與知識，程式設計、動畫製作與音樂編輯更是把我的生活點綴得多采多姿，這一切的一切實在讓我收穫太多了。

中文摘要

西爾平斯基船帆(Sierpinski Gasket)與西爾平斯基掛毯(Sierpinski Carpet)都屬於碎形(fractals)圖形的一種，可以利用迭代運算系統 IFS(Iterated Function Systems)碼來產生，代入迭代運算方程式後，經由多次的運算，可以得到重覆的圖形。本研究中，我將提出一些作法，找出西爾平斯基船帆與掛毯圖形其遞迴關係式，進而討論出其數位圖形之規律性及所涵蓋的內容與性質，著重在推廣西爾平斯基船帆與掛毯圖形的概念，將一段音樂曲取出，把它們看成反覆隨機迭代點，利用程式經由多次的插值運算，計算出各段音符。最後加入基因演算法來解決音符長短的問題，把製造好的音符染色體放置到交配池中，以隨機的方式在交配池中選取其中之一個染色體進行交配的動作，此二音符染色體會交換彼此的基因，產生下一代新的代表音符長短之染色體，應用於數位音樂創作，而衍生的西爾平斯基船帆與掛毯圖形新穎應用與創新的結果，希望能提供數位音樂創作的多樣性，更進而可以找出「好聽的音樂」與數學的直接關聯性。

關鍵詞：西爾平斯基、數位圖形、數位音樂

Digital image and music generator using Sierpinski Gasket and Carpet concepts

‘Sierpinski Gasket’ and ‘Sierpinski Carpet’ are two graphics that belong to fractals. They can be produced by IFS (Iterated Function Systems). By iterative computation of many times, we can obtain the similar graphics. In my research, there are some methods to generate Sierpinski Gasket, Sierpinski Carpet, and the iterative algorithms. In addition, I would discuss the regularity and the content as well as the properties of those digital patterns. At last, the advanced application of Sierpinski Gasket and Sierpinski Carpet to digital music pieces was presented. The program took a note of several measure of music as the beginning point, and made the IFS calculations for each new note in each measure. But there was no difference in beats if you just make the IFS iteration. So I changed the beats with genetic crossover method. In this research, the expression of the DNA to each beat of note was adopted. The same way, it took a note as a beginning point. And the system obtained the new DNA from the old notes for new ones randomly. That would make a piece of brand new music. What I want to do in this research is improve the multiformity of music and find what the relationship is of ‘good music’ and mathematical algorithms.

Keywords: Sierpinski, Digital patterns, Digital music

壹、前言

數學家曼德布洛(Mandelbrot)在 1960~70 年代發展出碎形幾何學[1-3]，碎形(fractals)會展現自我相似性，不管把碎形圖形放大或縮小，它和原本的圖形都長得很相似，取一小部份來看，就像整體一樣複雜，而圖形就在越來越小的尺度裡不斷重複[4]。

波蘭數學家西爾平斯基(Sierpinski)於 1916 年提出了西爾平斯基船帆圖形，這裡先介紹它的其中一種畫法(見圖 1)：

首先畫出一個任意三角形，再將三角形每一邊的中點連線得到四個小三角形，去掉中央的三角形。接下來所有步驟都和前面一樣：分成四個小三角形，去掉中央的三角形。我們可以發現，作一次這個步驟之後，剩下的許多小三角形總和面積都是原來的 3/4。如此不停的重複地作下去，按照極限的觀念，剩下的三角形總和面積將是

$$\lim_{n \to \infty} (\frac{3}{4})^n = 0$$

後來西爾平斯基又提出西爾平斯基掛毯圖形(見圖 2)：

首先畫出一個任意方形。再將方形每一邊取三等分，將三等分點連線，得到九個小方形，去掉中央的方形。接下來所有步驟也是都和前面一樣：分成九個小方形，去掉中央的方形。我們可以發現，作一次這個步驟之後，剩下的小方形總和面積都是原來的 8/9。如此不停的重複地下去，按照極限的觀念，剩下的小方形總和面積將是

$$\lim_{n \to \infty} (\frac{8}{9})^n = 0$$

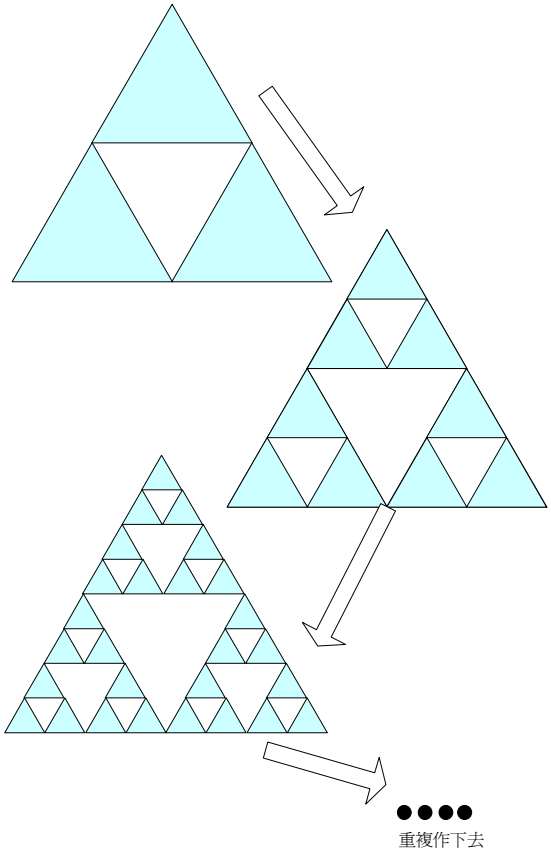


圖 1 西爾平斯基船帆圖形

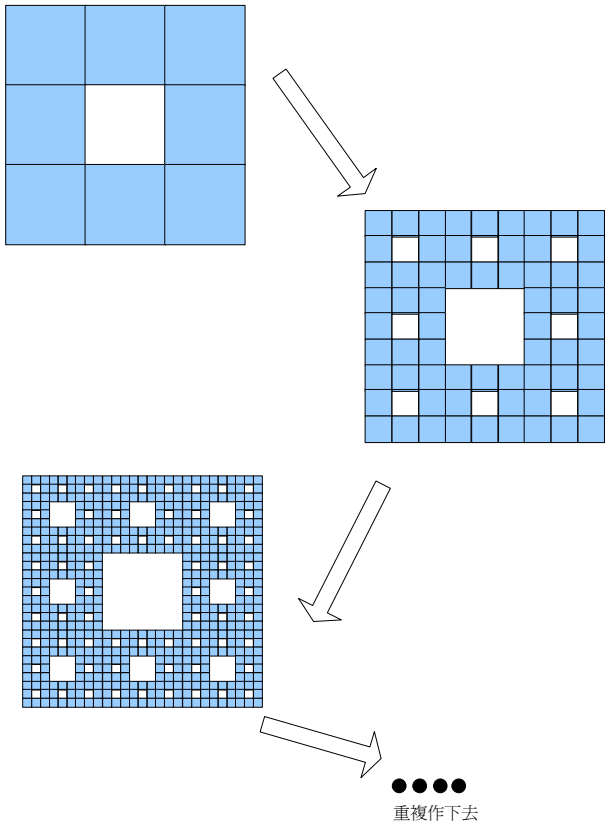


圖 2 西爾平斯基掛毯圖形

碎形圖形有很多的產生方式，以上是一種方法，還有一個方法是用迭代運算系統IFS(Iterated Function Systems)碼來產生，代入迭代運算方程式後，經由多次的運算，可以得到重覆的圖形。利用電腦繪圖技巧，可以將這些碎形呈現出來。

數學家曼德布洛提出「英國的海岸線有多長？」的文章。文章中提出一個想法：當使用越小的量尺來量度海岸線長度時，所量出的計量會越大。換句話說，海岸線的長度是按照量尺的大小而定的。所以說，任何海岸線的長度在某種意義下皆可視爲無限大。

羅勃・伊斯在「一條線有多長？」書中[5]，提出了生活中意想不到的數學謎題，例如「多瑙河有多長？／一條線有多長？有幾種不同答案？／碎形是什麼？能產生哪些奇妙的圖像？／數字中也藏有驚人的碎形？／碎形如何讓網路圖片傳遞更快？／學

會碎形，有可能大賺一票？／邊界無限長，面積也會無限大嗎？」等等有趣的碎形圖形觀念，並推衍到利用碎形結構來不斷重複、循環呈現樂曲，讓音樂家創作出碎形音樂與碎形變奏樂曲。書中提出「爲什麼有些聲音聽不到？／耳朵怎麼分辨出「難聽」與「悅耳」？／如何奏出好聽的組合音？／以噪音剋制噪音，真的有效？／和諧音的規則是用榔頭敲出來的？／十二音是怎麼來的？／史上最早的音階系統是什麼？／世上真有魔鬼音？／荒腔走板的歌聲也有可能是天籟美聲？」等碎形音樂的問題。所謂碎形音樂，就是將樂曲組成自我相似性的音樂旋律，比如說將原來的某音樂曲找出兩段碎形結構，其一爲主旋律，另一爲副旋律，整個音樂的旋律係以主旋律爲主，但在適當處接上副旋律，不斷重複、循環呈現的結果，別有一番風味。

是否存在有一個演算法，可以使得某一段莫扎特的樂曲，例如莫扎特第 24 號奏鳴曲，輸入後輸出爲第 25 號奏鳴曲？或是出現另一位作曲家如貝多芬的曲風？雖然說來匪夷所思，無可否認的，這是一個有趣的想法。我初次接觸碎形，不禁讚嘆產生碎形的容易程度，像西爾平斯基船帆與西爾平斯基掛毯，利用迭代運算系統來產生，任取一點出發，只要反覆隨機取三角形頂點，經由多次的插值運算，代入迭代運算方程式後，就能描繪出重覆的圖形。竟然規則那麼簡單，而反覆迭代後竟能有如此美妙的結構，令人十分驚訝，這也引起我進一步去一窺究竟的動機。我同時想到，推廣西爾平斯基船帆與掛毯圖形的概念，應用於數位音樂創作，一定可以衍生新穎應用與創新的結果。

貳、研究方法或過程

一、 研究方法

假設多邊形頂點、中心點或圓形等圖案特徵點爲各反覆隨機迭代點 Pm={ P1, P2, P3…..Pn}。

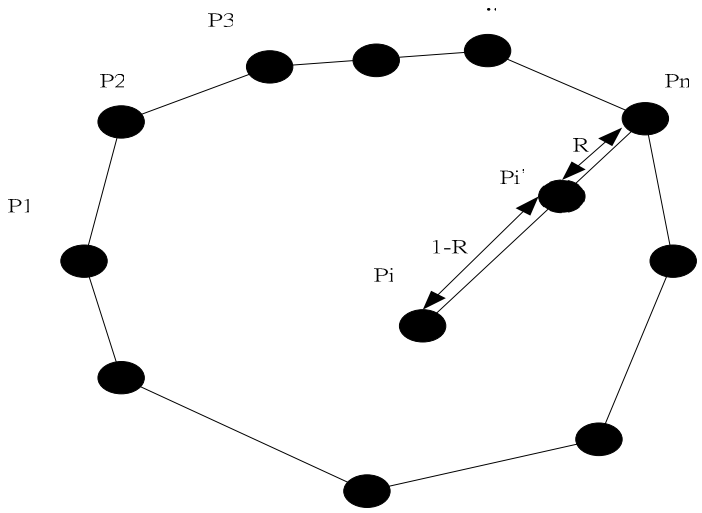


圖 3 反覆隨機迭代點 Pn 與新迭代點 Pi’

利用迭代運算系統，任取一點 Pi 出發，反覆隨機與各迭代點作插值運算，取適當的插值運算係數 R，得到新迭代點 Pi’ (見圖 3)：

$$Pi' = R \cdot Pi + (1 - R) \cdot Pn$$

其中，R 爲新迭代點 Pi’ 到各迭代點之距離與 Pi 到各迭代點之距離之比值。

我們發現：

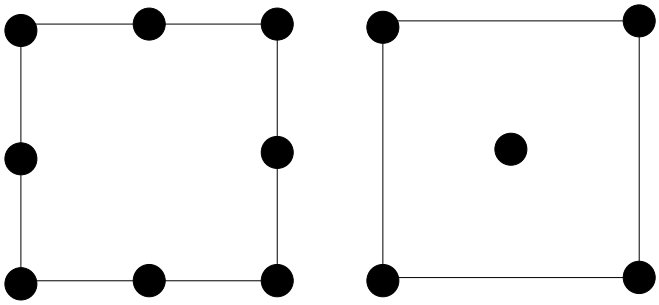
1. 當反覆隨機迭代點爲三角形三頂點，插值運算係數爲 0.5，可以得到 Sierpinski Gasket 圖案。
2. 當反覆隨機迭代點爲正方形頂點與各邊中點(見圖 4（A）)，插值運算係數爲 1/3，可以得到 Sierpinski Carpet 圖案。
3. 當反覆隨機迭代點爲五邊形頂點，插值運算係數爲 3/8，可以得到 Sierpinski

Pentagon 圖案。

4. 當反覆隨機迭代點爲六邊形頂點，插值運算係數爲 1/3，可以得到 Sierpinski Hexagon 圖案。
5. 當反覆隨機迭代點爲正方形頂點，插值運算係數爲 1/3，可以得到 Cantor Square 圖案。
6. 當反覆隨機迭代點爲正方形頂點與內部中心點(見圖 4（B）)，插值運算係數爲 1/3，可以得到 Box Fractal 圖案。

另外，我們還可以做一些進階的實驗：

1. 改變插值運算係數，當 R 值較小時，各重複圖案距離較遠。當 R 值較大時，圖案會擠在一起，呈現混亂的結果。
2. 當反覆隨機迭代點爲多邊形或圓形圖案，只要反覆隨機取多邊形圖案頂點或圓形的各個點，取適當插值運算係數，經由多次的插值運算，就能產生出有趣的圖形。
3. 將一段音樂曲取出，把它們看成反覆隨機迭代點，利用程式經由多次的插值運算，可計算出各段音符。



（A）　　（B）

圖 4（A） Sierpinski Carpet 迭代點 （B）Box Fractal 迭代點

另外，由於如果直接把音符的高低做迭代運算，那麼運算出來的音符常常會變得平平的，變化不大。於是本研究就將各個音符高低的差距拉大。

具體的作法爲：

1.先將最高和最低的音符(分別設爲 Max 及 Min)找出來，再找出它們的平均值(設爲 Aver)。

2.把各個音符(n[i])和平均值的距離(n[i]-Aver)找出來。

3.將差距加大爲(n[i]-Aver)/R，其中 R 爲迭代運算係數。

4.即可得到各個新的迭代頂點爲(n[i]-Aver)/R + Aver。

經過迭代運算後的 MID 檔[6]之音符只有音高的變化，並沒有長短的不同，所以我利用基因觀念之交配(Crossover)的方法[7]來解決長短的問題。這個方法是把製造好的音符染色體放置到交配池中，以隨機的方式在交配池中選取其中之一個染色體進行交配的動作，此二音符染色體會交換彼此的基因，產生下一代新的代表音符長短之染色體。

要做基因交配，首先要製造每一個音符長短的基因，我的定義是：把一拍分成八等份，利用 8 個 x 或 y 的組合來表示，而以 y 的數量多寡來代表音符的長短。

例如說：

1/8 的音符，表示成 xxxxxxxy

1/2 的音符，表示成 xxxxyyyy

第二個步驟，要把交配池的隨機取兩個基因來交配，交配的方法是把各個基因的染色體(x 和 y)各自隨機取得(見圖 5)，成爲一個新的基因。

把以上這兩個例子隨機取出新的染色體，可能是：

1/8 的音符，表示成 xxxxxxxy

1/4 的音符，表示成 xxxxxxyy

3/8 的音符，表示成 xxxxxyyy

1/2 的音符，表示成 xxxxyyyy

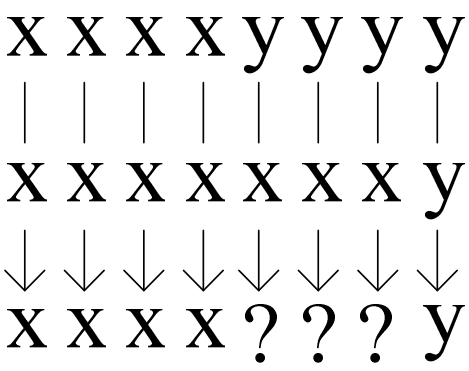


圖 5 把各個基因的 x 和 y 各自隨機取得

也許會有人說：「如果我在它兩個基因的範圍之間隨便挑一個基因，那它的效果不是和交配法一樣嗎？」其實是不一樣的。以機率來說，交配後的音符，它的長短介於它的父母之間的機率，會大於其他的機率。如圖 5 的例子，出現 1/4、3/8 音符的機率爲 3/8，而出現 1/8、1/2 音符的機率爲 1/8，就像兒女的身高，介於他們父母的身高之間的機率，會比其他情況的機率來得高。因此我認爲在這個地方使用基因交配法是比較合理的一個作法。

二、 研究過程

在本研究的軟硬體需求上，使用以下設備及軟體：

1.個人電腦一台(CPU：PentiumⅢ 500，記憶體：128MB ，顯示卡：S3 savage 3D/M)，用來執行系統

2.圖像印出裝置

3.程式軟體：Borland C++ Builder 6.0

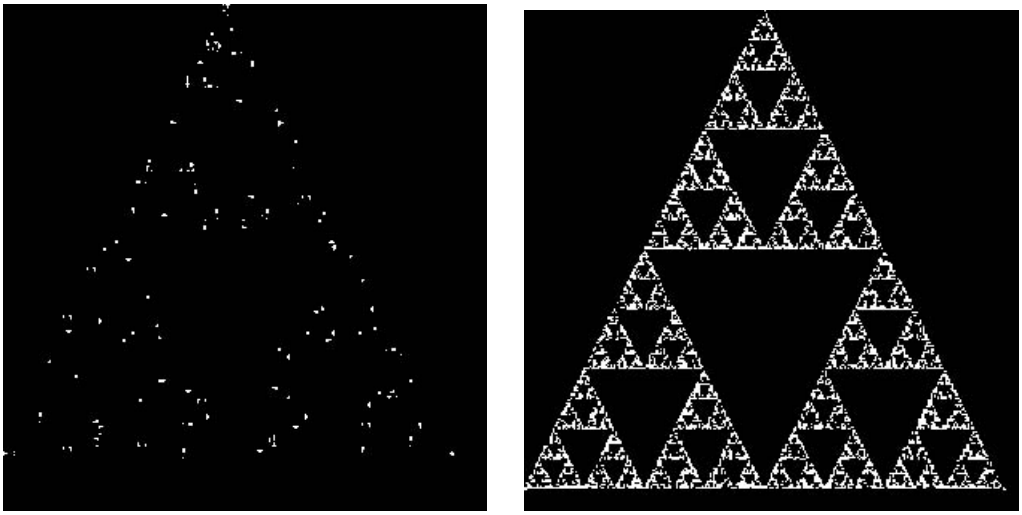
4.數位音樂創作軟體：Cakewalk 8.0

首先利用 Borland C++ Builder 撰寫出西爾平斯基船帆圖形描繪程式，利用迭代運算系統，任取一點出發，發現最後圖形與起始點無關。進一步過展到多邊形，甚至圓形圖案，只要反覆隨機取多邊形或圓形圖案頂點，經由多次的插值運算，就能產生出有趣的圖形。

最後，推廣西爾平斯基船帆與掛毯圖形的概念，應用於數位音樂創作，將一段音樂曲取出，把它們看成多邊形或圓形圖案頂點，利用程式經由多次的插值運算與基因交配法，計算出碎形結構的各段音符，讓樂曲組成自我相似性的音樂旋律，重複循環呈現，輸出 MIDI 檔，如果還要加上副旋律和節奏，再利用數位音樂創作軟體 Cakewalk 進行編曲。

參、研究結果與討論

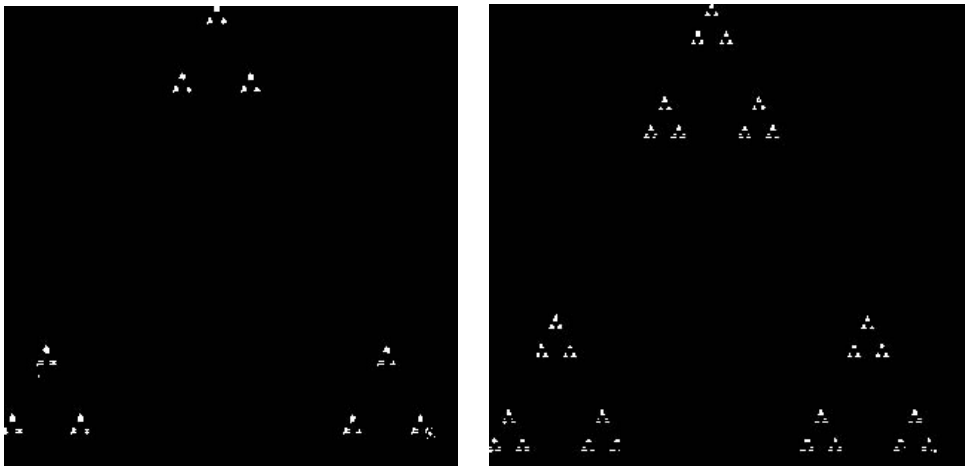
利用 Borland C++ Builder 撰寫出西爾平斯基船帆圖形描繪程式，當迭代運算次數少時，圖案並不明顯(見圖 6（A）)。當迭代運算次數高時，圖案即明顯呈現(見圖 6（B）)。



（A）　　（B）

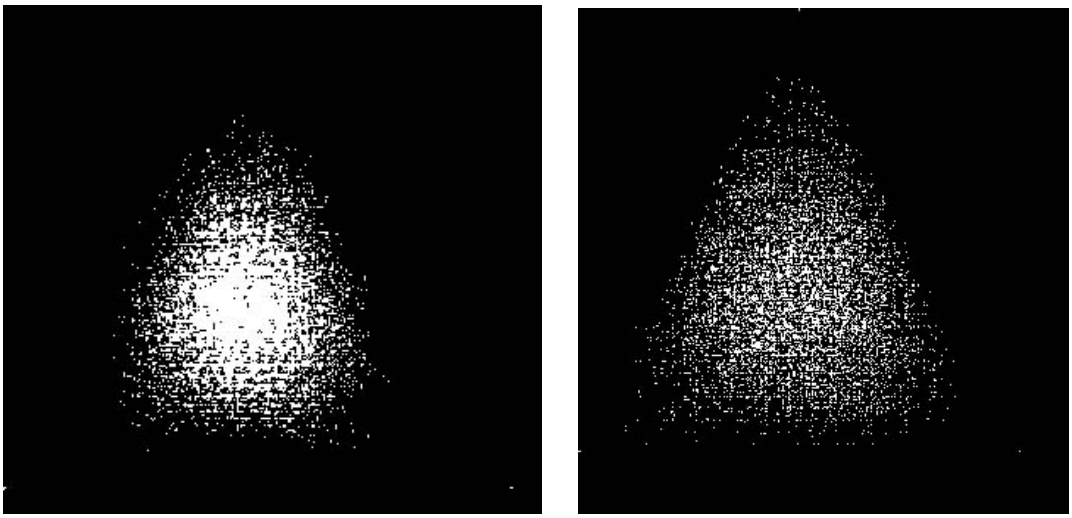
圖 6 西爾平斯基船帆圖形經（A）100 次 （B）10000 次迭代運算後(R=0.5)

三角形頂點，經由多次的插值運算，代入迭代運算方程式後，其描繪出的圖形如圖 7 (R<0.5) 與圖 8(R > 0.5)所示。當 R 值小於 0.5 時，各重複圖案距離較遠，彼此不相干涉。當 R 值大於 0.5 時，圖案呈現混亂的結果。



（A）　　（B）

圖 7 三角形頂點經由多次的插值運算之圖形（A）R=0.2 （B）R=0.3)



（A）　　（B）

圖 8 三角形頂點經由多次的插值運算之圖形(A）R=0.9（B） R=0.8

接下去的迭代運算皆訂爲 10000 次，執行西爾平斯基掛毯圖形描繪程式，當 R=0.33 時，圖案即明顯呈現(見圖 9)。正方形頂點與各邊中點，經由 10000 次的插值運算，當 R 值小於 0.33 時，其描繪出的圖形如圖 10（A），當 R 值大於 0.33 時，圖案呈現混亂的結果，如圖 10（B）。

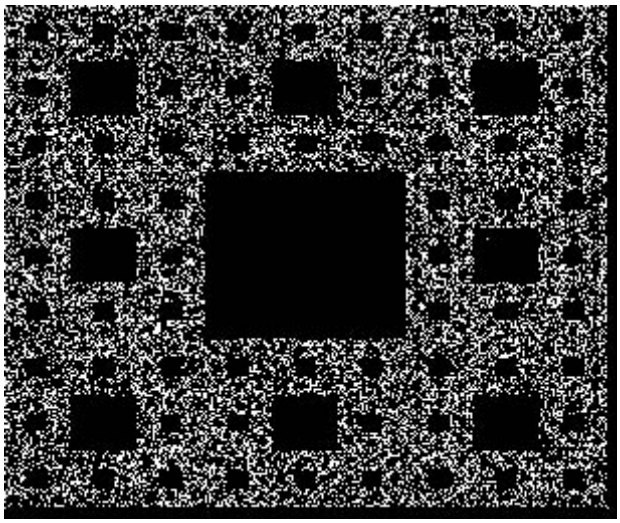
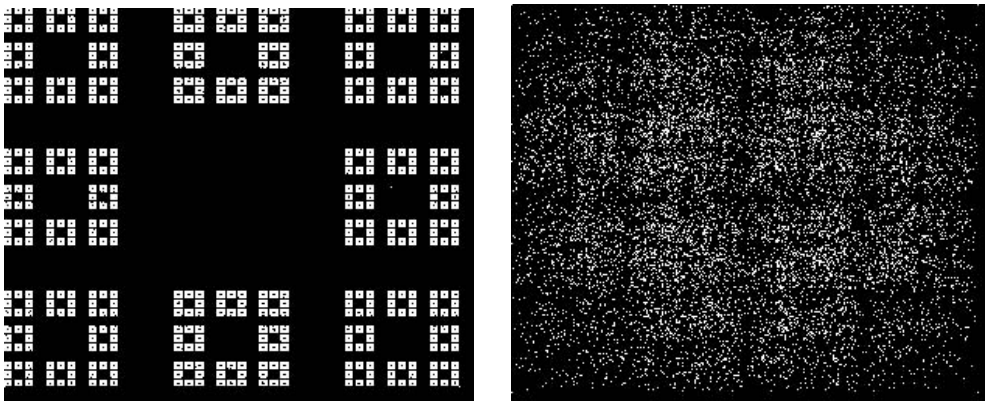


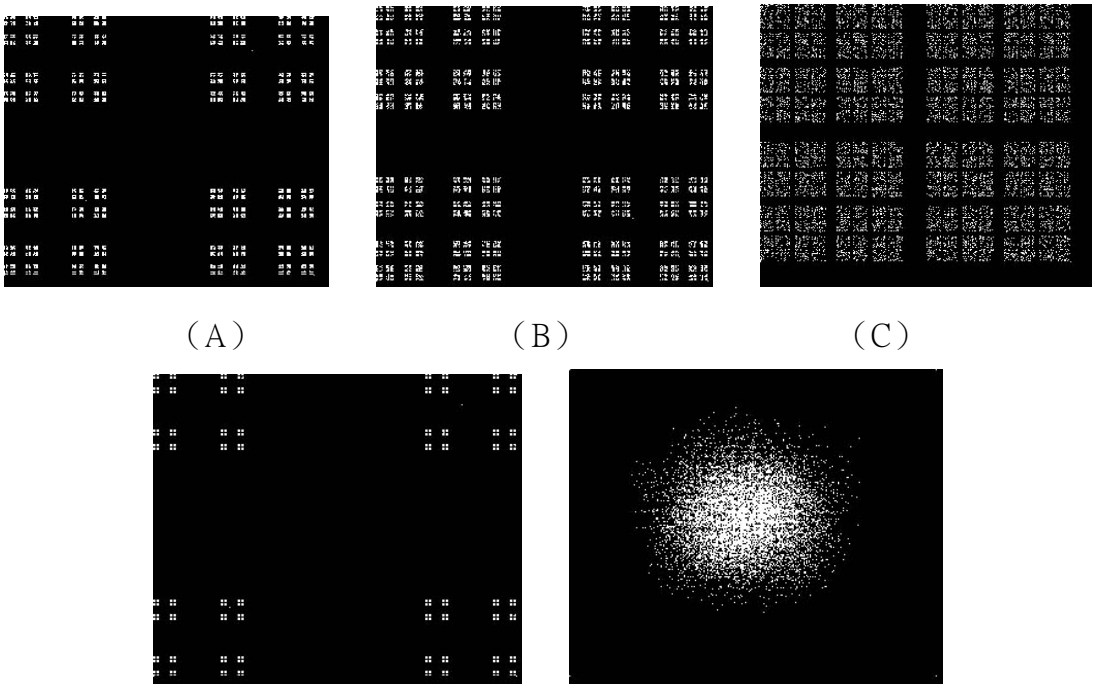
圖 9 西爾平斯基掛毯圖形(R=0.33)



（A）　　（B）

圖 10 正方形頂點與各邊中點經由多次的插值運算之圖形 （A）R=0.25　（B）R=0.5

正方形頂點，經由 10000 次的插值運算，當 R 值等於 0.33 時，其描繪出的圖形如圖 11（A），當 R 值大於 0.33 時，其描繪出的圖形如圖 11（B），與圖 11（C），當 R 值小於 0.33 時，其描繪出的圖形如圖 11（D）。當 R 值甚大時，圖案呈現混亂的結果，如圖 11（E）。

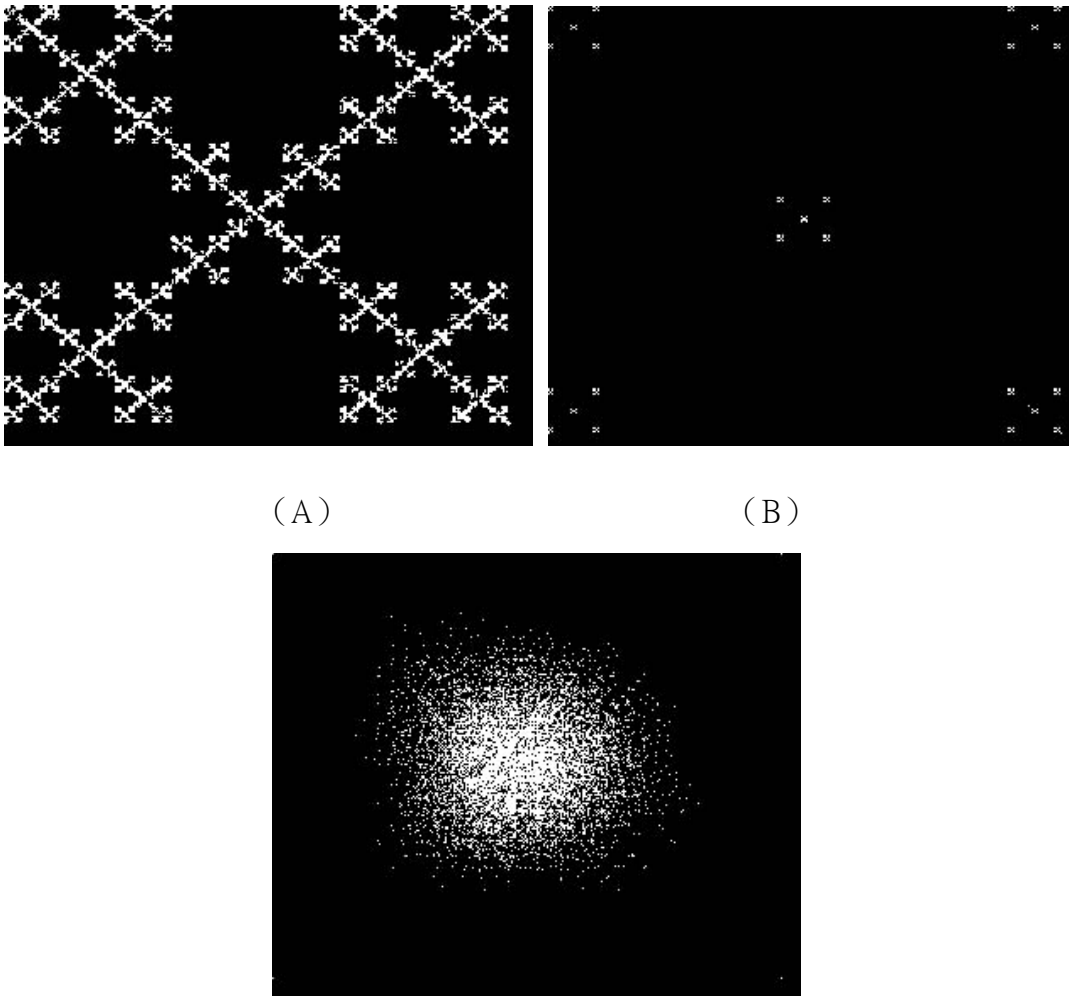


（A）　　（B）　　（C）

（D）　　（E）

圖 11 正方形頂點經由多次的插值運算之圖形（A）R=0.33　（B）R=0.4 （C）R=0.46 （D） R=0.25 （E） R=0.9

正方形頂點加上內部中心點，經由 10000 次的插值運算，當 R 值等於 0.33 時，其描繪出的圖形如圖 12（A），當 R 值小於 0.33 時，其描繪出的圖形如圖 12（B），當 R 值大於 0.33 時，圖案呈現混亂的結果，如圖 12（C）。

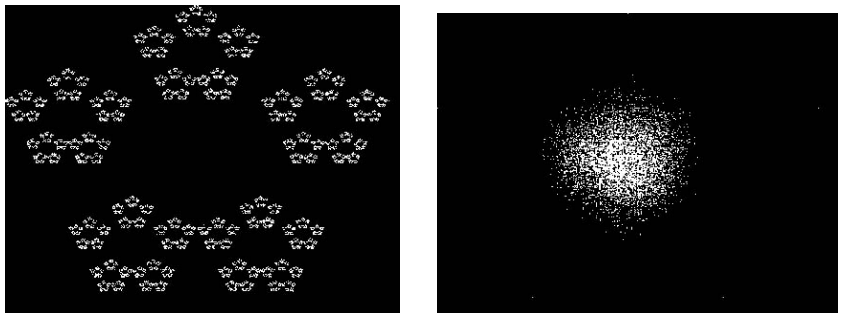


（A）　　（B）

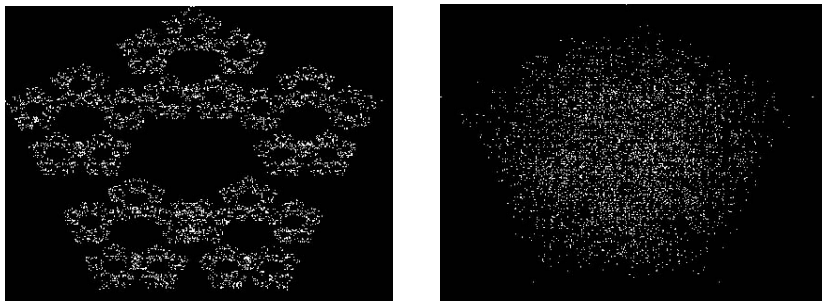
（C）

圖 12 正方形頂點加上內部中心點，經由多次的插值運算之圖形（A）R=0.33　（B）R=0.1　（C）R=0.9

正五邊形反覆隨機取多邊形頂點，經由多次的插值運算，所產生出有趣的圖形如圖 13。



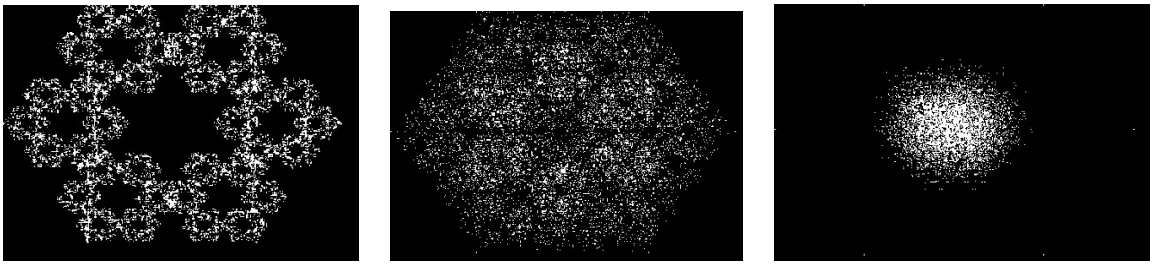
（A）　　（B）



（C）　　（D）

圖 13 正五邊形頂點經由多次的插值運算之圖形 （A）R=0.33 （B）R=0.9 （C）R=0.4 （D） R=0.6

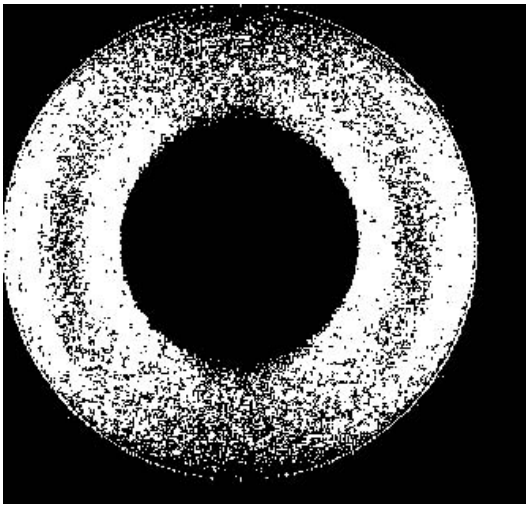
正六邊形反覆隨機取多邊形頂點，經由多次的插值運算，所產生出有趣的圖形如圖 14。



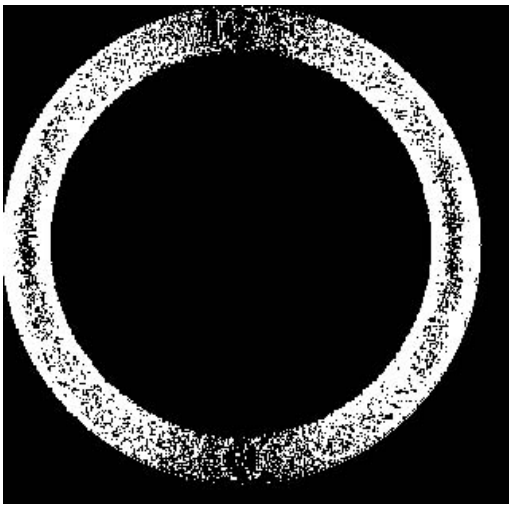
（A）　　（B）　　（C）

圖 14 正六邊形頂點經由多次的插值運算之圖形(A)R=0.375 (B)R=0.5 （C）R=0.9

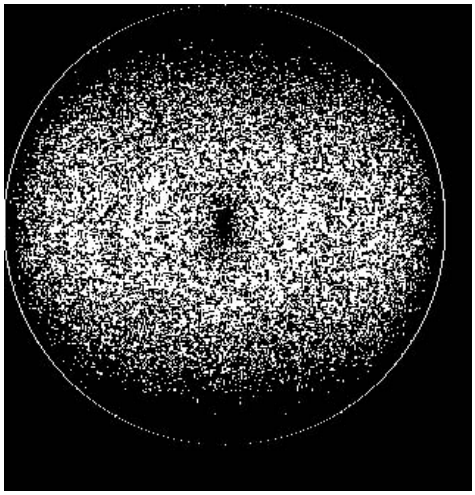
進一步過展到圓形圖案，反覆隨機取圓形圖案各點，經由多次的插值運算，所產生出有趣的圖形如圖 15。



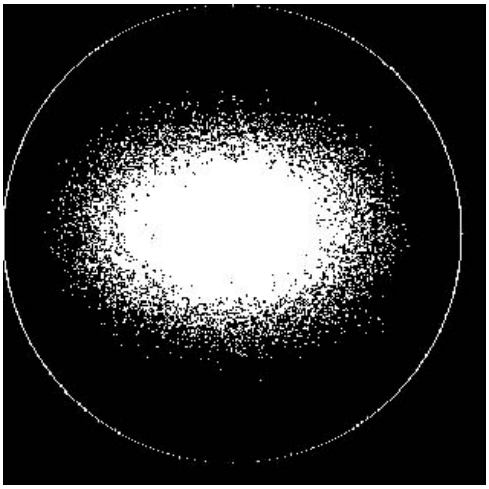
（A）



（B）



（C）



（D）

圖 15 以圓形爲頂點經 10000 次迭代運算後圖形（A）R=0.25（B）R=0.1（C）R=0.5（D） R=0.8

推廣西爾平斯基船帆與掛毯圖形的概念，應用於數位音樂創作，而衍生的碎形音樂創新，將一段音樂曲取出，利用程式經由多次的插值運算，計算出音符的數值，再利用數位音樂創作軟體 Cakewalk 4.0 進行編曲，如圖 16。

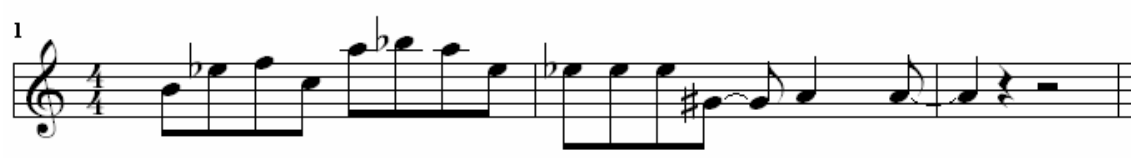


圖 16　推廣西爾平斯基船帆與掛毯圖形的概念而衍生的碎形音樂

爲了避免音樂單調，在主旋律之外，又加入副旋律與伴奏，讓整體變奏音樂內涵豐富起來(見圖 17)。

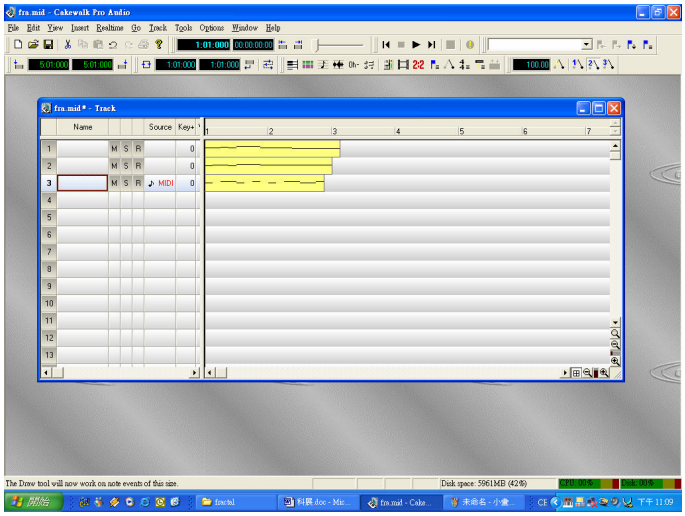


圖 17　在主旋律(Track 1)之外，又加入副旋律(Track 2)與伴奏(Track 3)

此我利用用 MIDI 編曲，並透過 Windows 的 MIDI 功能來播放音樂，在此討論 MIDI 檔格式如下：MIDI 是數位樂器介面 Musical Instrument Digital Interface 的縮寫，它是一種電子樂器之間的通訊協定標準。MID 檔案是將每一聲道的每一個音符的資訊通通以文字記錄下來。就是因爲 MID 檔案只是記錄音符，而不是記錄某一時間的波形，所以 MID 檔案不像 WAV 檔那樣的龐大，如果只是純音樂的話，大家就比較常在網路上使用 MIDI。

MID 檔主要是由兩大部份節區資料所組合而成：表頭資料節區 Mthd、音軌資料節區 Mtrk 節區。在表頭資料 Mthd 節區的內容包括節區名稱、節區資料大小、MIDI 檔類型、音軌節區數目、時間基準。而在 Mtrk 音軌節區包含節區名稱、節區資料大小、節區資料[6]。

而我在前面所提到的「將音符高低、長短改變」，在程式中就是讀取一個 MIDI FORMAT 0 檔，經過迭代法、基因演算法[7]之後，再把它寫入另一個 MIDI 檔。因爲我要做的碎形音樂只是單音，所以在這裡要更動的東西不多，我只需要用到 Mtrk 音軌節區的節區資料的一小部份而已。

一個音符的樣子是長這樣：

ABCA

其中「A」是表示音高，「B」是表示音量，「C」是表示長度，它們都是以 ASCII 碼的數值運作的。例如一個高音 Do，音量 100，半拍的音符，它就表示成：

Hd < H

其中「H」的 ASCII 碼爲 72，代表高音 Do，「d」的 ASCII 碼爲 100，代表音量 100，「<」的 ASCII 碼爲 60，代表長度爲半拍。本研究透過 MID 檔讀取四小節的樂曲，利用多次的插值運算與基因交配，藉此產生下一代新的碎形音樂，再寫入到新的 MID 音樂檔案中，並進行播放音樂。

本研究整合碎形圖形及碎形音樂產生程式，其畫面如圖 18 所示。而碎形音樂部份的處理流程如圖 19 所示。

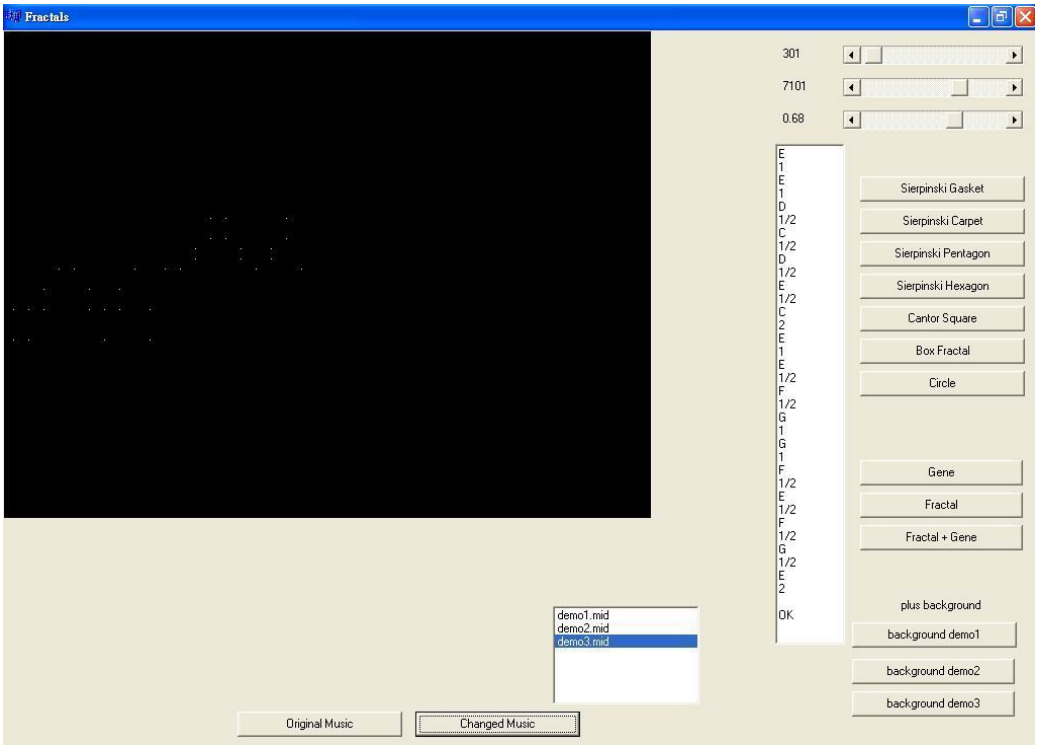


圖 18 碎形圖形及碎形音樂產生程式之畫面

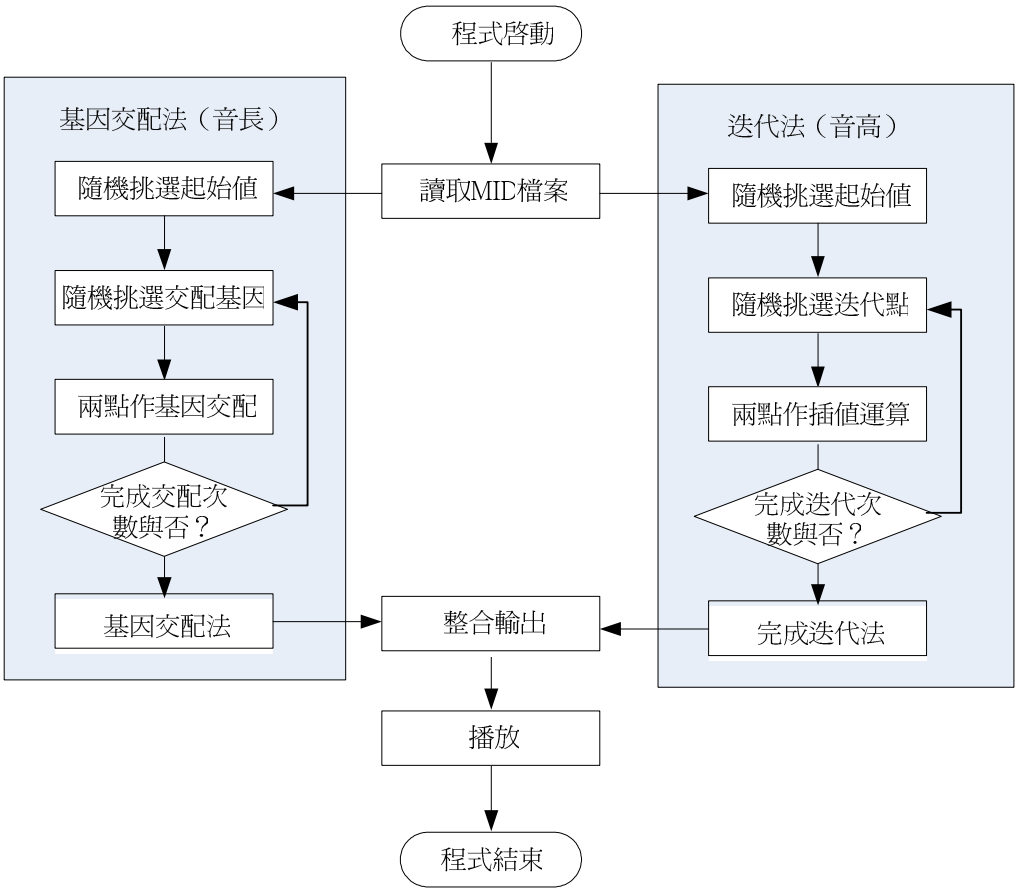


圖 19 碎形音樂產生程式流程

表 1 本研究之碎形音樂的試聽結果

試聽結果	難聽	普通	好聽
旋律 1	21%	40%	39%
旋律 2	12%	47%	41%
旋律 3	17%	32%	51%

利用設計出來的程式和處理流程，挑選三段旋律作爲實驗素材，產生碎形音樂，經由一百人次的實驗與試聽之後，結果所得數據如表 1 所示。我們在試聽過程中逐一地將每首碎形音樂進行是否悅耳的判斷，由表中的數據可以得知，對於本研究所實驗的旋律，利用我們所設計出的碎形音樂技術重新編曲，結果「好聽」的程度皆大於「難聽」的程度。至於「旋律 1」的悅耳程度較低，根據分析，我認爲是音符的長短、音高的變化量不足，而音符數目又太少，導致迭代點不足，而造成較爲難聽的結果。

目前電腦已被廣泛地運用到音樂創作上，從而增加了許多音效及使用的素材，呈現出多樣性、範圍廣、重視音響表現等面貌。本研究推廣西爾平斯基船帆與掛毯圖形的概念，應用於數位音樂創作，而衍生的西爾平斯基船帆與掛毯圖形新穎應用與碎形音樂創新的結果。爲了避免音樂單調，在主旋律之外，又加入副旋律與伴奏，讓整體變奏音樂內涵豐富起來，而又不失整體的和諧性。也許在未來有了電腦碎形音樂軟體的輔助，會使作曲家更節省人力，更容易完成數位音樂創作。

肆、結論與應用

目前電腦已被廣泛地運用到音樂創作上，從而增加了許多音效及使用的素材，呈現出多樣性、範圍廣、重視音響表現等面貌。本研究推廣西爾平斯基船帆與掛毯圖形的概念，應用於數位音樂創作，而衍生的西爾平斯基船帆與掛毯圖形新穎應用與碎形音樂創新的結果。爲了避免音樂單調，在主旋律之外，又加入副旋律與伴奏，讓整體變奏音樂內涵豐富起來，而又不失整體的和諧性。也許在未來有了電腦碎形音樂軟體的輔助，會使作曲家更節省人力，更容易完成數位音樂創作。再更進一步，我希望未來能就此發現一條路，可以找出「好聽的音樂」與數學的直接關聯性。

伍、參考文獻

1. 吳文成，碎形，Fractal http://alumni.nctu.edu.tw/~sinner/ complex/ fractals/index.html
2. Lawrence H. Riddle, Sierpinski Gasket, http://ecademy.agnesscott.edu/ ~lriddle/ ifs/siertri/siertri.htm
3. Stewart, Ian. Four Encounters with Sierpinski's Gasket, The Mathematical Intelligencer, 17, No. 1 (1995), 52-64.
4. 廖思善，碎形的魅力，科學月刊 363（2000 年 3 月號），p.222。
5. 羅勃・伊斯(譯者：蔡承志)，一條線有多長？生活中意想不到的 116 個數學謎題，三言社出版,2005 年 6 月
6. 林宸生，數位信號--影像與聲音處理，全華書局，民國 86 年 6 月
7. Goldberg, D, E. , Genetic Algorithm in Search, Optimization, and Machine Learning, Addison-Wesley , New York, 1989.

附錄(程式碼)

```
//---------------------------------------------------------------------------
#include<math.h>


#include <vcl.h>
#pragma hdrstop

#include "Unit1.h"
//---------------------------------------------------------------------------
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;
//---------------------------------------------------------------------------
__fastcall TForm1::TForm1(TComponent* Owner)
    : TForm(Owner)
{
}
//---------------------------------------------------------------------------

void __fastcall TForm1::ScrollBar1Change(TObject *Sender)
{
 Label1->Caption=(ScrollBar1->Position)*100+1;
}
//---------------------------------------------------------------------------
void __fastcall TForm1::ScrollBar2Change(TObject *Sender)
{
 Label2->Caption=(ScrollBar2->Position)*100+1;
}
//---------------------------------------------------------------------------
void __fastcall TForm1::ScrollBar3Change(TObject *Sender)
{
 Label3->Caption=(ScrollBar3->Position)*0.01+0.01;
}
//---------------------------------------------------------------------------
void __fastcall TForm1::Button1Click(TObject *Sender)
{

    int x[3];
```

```
int y[3];
int r;
float i, j, R ;
int a =      250;
int b =      250;

x[0]=250;
  y[0]=0;
  x[1]=0;
  y[1]=400;
  x[2]=500;
  y[2]=400;

for(i=0;i<3;i++){
          x[(int)i]=x[(int)i]+10;
          y[(int)i]=y[(int)i]+10;
}

for (i=1;i<=640;i++){
          for ( j = 1; j <= 480; j++)
                    Canvas->Pixels[i][j] = clBlack;
}

for(i=0;i<3;i++)
          Canvas->Pixels[x[(int)i]][y[(int)i]] = clYellow;

randomize();

i=(ScrollBar1->Position)*100+1;
j=i+(ScrollBar2->Position)*100+1;
R=(ScrollBar3->Position)*0.01+0.01;

for (;i<=j;i++){
          r=rand()%3;

          a = a * R + x[r] * (1-R) ;
          b = b * R + y[r] * (1-R) ;

          Canvas->Pixels[a][b] = clWhite;
}
```

```
}
//---------------------------------------------------------------------------

void __fastcall TForm1::Button2Click(TObject *Sender)
{
    int x[8];
    int y[8];
    int r;
    float i, j, R ;
    int a=250;
    int b=250;

     x[0]=0;
     y[0]=0;
     x[1]=0;
     y[1]=250;
     x[2]=300;
     y[2]=0;
     x[3]=300;
     y[3]=250;
     x[4]=150;
     y[4]=0;
     x[5]=0;
     y[5]=125;
     x[6]=150;
     y[6]=250;
     x[7]=300;
     y[7]=125;

    for(i=0;i<8;i++){
            x[(int)i]=x[(int)i]+10;
            y[(int)i]=y[(int)i]+10;
    }

    for (i=1;i<=640;i++){
              for ( j = 1; j <= 480; j++)
                        Canvas->Pixels[i][j] = clBlack;
    }

    for(i=0;i<8;i++)
```

```
            Canvas->Pixels[x[(int)i]][y[(int)i]] = clYellow;


    randomize();

    i=(ScrollBar1->Position)*100+1;
    j=i+(ScrollBar2->Position)*100+1;
    R=(ScrollBar3->Position)*0.01+0.01;

    for (; i <= j; i++){
            r=rand()%8;

            a = a * R + x[r] * (1-R) ;
            b = b * R + y[r] * (1-R) ;

            Canvas->Pixels[a][b] = clWhite;
    }
}
//---------------------------------------------------------------------------

void __fastcall TForm1::Button3Click(TObject *Sender)
{
    int x[5];
    int y[5];
    int r;
    float i, j, R ;
    int a =     250;
    int b =     250;

     x[0]=200;
     y[0]=0;
     x[1]=0;
     y[1]=100;
     x[2]=400;
     y[2]=100;
     x[3]=100;
     y[3]=300;
     x[4]=300;
     y[4]=300;
```

```
    for(i=0;i<5;i++){
            x[(int)i]=x[(int)i]+10;
            y[(int)i]=y[(int)i]+10;
    }

    for ( i = 1; i <= 640; i++){
            for ( j = 1; j <= 480; j++)
                    Canvas->Pixels[i][j] = clBlack;
    }

    for(i=0;i<5;i++)
            Canvas->Pixels[x[(int)i]][y[(int)i]] = clYellow;


    randomize();

    i=(ScrollBar1->Position)*100+1;
    j=i+(ScrollBar2->Position)*100+1;
    R=(ScrollBar3->Position)*0.01+0.01;

    for (; i <= j; i++){
             r=rand()%5;

             a = a * R + x[r] * (1-R) ;
             b = b * R + y[r] * (1-R) ;

             Canvas->Pixels[a][b] = clWhite;
    }
}
//---------------------------------------------------------------------------

void __fastcall TForm1::Button4Click(TObject *Sender)
{
    int x[6];
    int y[6];
    int r;
    float i, j, R ;
    int a =      250;
    int b =      250;
```

```
  x[0]=100;          y[0]=0;
  x[1]=300;          y[1]=0;
  x[2]=0;            y[2]=140;
  x[3]=400;          y[3]=140;
  x[4]=100;          y[4]=280;
  x[5]=300;      y[5]=280;

 for(i=0;i<5;i++){
        x[(int)i]=x[(int)i]+10;
        y[(int)i]=y[(int)i]+10;
 }

 for ( i = 1; i <= 640; i++){
      for ( j = 1; j <= 480; j++)
            Canvas->Pixels[i][j] = clBlack;
 }

 for(i=0;i<6;i++)
      Canvas->Pixels[x[(int)i]][y[(int)i]] = clYellow;


 randomize();

 i=(ScrollBar1->Position)*100+1;
 j=i+(ScrollBar2->Position)*100+1;
 R=(ScrollBar3->Position)*0.01+0.01;

 for (; i <= j; i++){
        r=rand()%6;

        a = a * R + x[r] * (1-R) ;
        b = b * R + y[r] * (1-R) ;

        Canvas->Pixels[a][b] = clWhite;
 }
}
//---------------------------------------------------------------------------

void __fastcall TForm1::Button5Click(TObject *Sender)
{
```

```
int x[4];
int y[4];
int r;
float i, j, R ;
int a =        250;
int b =      250;

x[0]=0;
  y[0]=0;
  x[1]=0;
  y[1]=250;
  x[2]=300;
  y[2]=0;
x[3]=300;
  y[3]=250;

for(i=0;i<4;i++){
x[(int)i]=x[(int)i]+10;
y[(int)i]=y[(int)i]+10;
}

for ( i = 1; i <= 640; i++){
        for ( j = 1; j <= 480; j++)
                Canvas->Pixels[i][j] = clBlack;
}

for(i=0;i<4;i++)
          Canvas->Pixels[x[0]][y[0]] = clYellow;

randomize();

i=(ScrollBar1->Position)*100+1;
j=i+(ScrollBar2->Position)*100+1;
R=(ScrollBar3->Position)*0.01+0.01;

for (; i <= j; i++){
          r=rand()%4;

          a = a * R + x[r] * (1-R) ;
```

```
            b = b * R + y[r] * (1-R) ;

            Canvas->Pixels[a][b] = clWhite;
    }
}
//---------------------------------------------------------------------------

void __fastcall TForm1::Button6Click(TObject *Sender)
{
    int x[5];
    int y[5];
    int r;
    float i, j, R ;
    int a =     250;
    int b =     250;

     x[0]=0;
     y[0]=0;
     x[1]=0;
     y[1]=250;
     x[2]=300;
     y[2]=0;
     x[3]=300;
     y[3]=250;
     x[4]=150;
     y[4]=125;


    for(i=0;i<5;i++){
            x[(int)i]=x[(int)i]+10;
            y[(int)i]=y[(int)i]+10;
    }

    for ( i = 1; i <= 640; i++){
        for ( j = 1; j <= 480; j++)
                Canvas->Pixels[i][j] = clBlack;
    }

    for(i=0;i<5;i++)
        Canvas->Pixels[x[(int)i]][y[(int)i]] = clYellow;
```

```
    randomize();

    i=(ScrollBar1->Position)*100+1;
    j=i+(ScrollBar2->Position)*100+1;
    R=(ScrollBar3->Position)*0.01+0.01;

    for (; i <= j; i++){
            r=rand()%5;

            a = a * R + x[r] * (1-R) ;
            b = b * R + y[r] * (1-R) ;

            Canvas->Pixels[a][b] = clWhite;
    }
}
//---------------------------------------------------------------------------

void __fastcall TForm1::Button7Click(TObject *Sender)
{
    int x;
    int y;
    int r;
    float i, j, R ;
    int rd=150;
    int a =     250;
    int b =     250;

    for ( i = 1; i <= 640; i++){
        for ( j = 1; j <= 480; j++)
            Canvas->Pixels[i][j] = clBlack;
    }


    for(i=0;i<=2*rd;i++){

        x=(rd*rd)-(rd-i)*(rd-i);
        x=rd+sqrt((double)x);
        y=i;
```

```
        Canvas->Pixels[x][y] = clYellow;

}

for(i=0;i<=2*rd;i++){

        x=(rd*rd)-(rd-i)*(rd-i);
        x=rd-sqrt((double)x);
        y=i;

        Canvas->Pixels[x][y] = clYellow;

}

randomize();

i=(ScrollBar1->Position)*100+1;
j=i+(ScrollBar2->Position)*100+1;
R=(ScrollBar3->Position)*0.01+0.01;

for(;i<j;i++){

          r=rand()%2;

          if(r==0){

                  r=rand()%(2*rd+1);

                  x=(rd*rd)-(rd-r)*(rd-r);
                  x=rd+sqrt((double)x);
                  y=r;

                     a=a*R+x*(1-R);
                     b=b*R+y*(1-R);

                     Canvas->Pixels[a][b] = clWhite;
            }

          else{
```

```
                        r=rand()%(2*rd+1);

                        x=(rd*rd)-(rd-r)*(rd-r);
                        x=rd-sqrt((double)x);
                        y=r;

                        a=a*R+x*(1-R);
                        b=b*R+y*(1-R);
                        Canvas->Pixels[a][b] = clWhite;

                }
        }

}
//---------------------------------------------------------------------------

void __fastcall TForm1::Button8Click(TObject *Sender)
{

  mciSendString("play bg1.mid",NULL,1024,0);

}
//---------------------------------------------------------------------------

void __fastcall TForm1::Button9Click(TObject *Sender)
{
mciSendString("play ori.mid",NULL,1024,0);
/* MediaPlayer1->FileName="ori.mid";
     MediaPlayer1->Open();      */
}
//---------------------------------------------------------------------------

void __fastcall TForm1::Button10Click(TObject *Sender)
{
mciSendString("play fra.mid",NULL,1024,0);
/* MediaPlayer1->FileName="fra.mid";
     MediaPlayer1->Open();    */
}
//---------------------------------------------------------------------------
```

```
void __fastcall TForm1::Button11Click(TObject *Sender)
{
    // char    *str=Label4->Caption.c_str();
      const int size=1024;
      char buffer[size];
      char aa[32],bb[32];
      int p,b[4];
      int r,r2;
      int k=32;
      int sum,kk[4];
      char xy[32][4];

      memset(buffer,0,size);

      HFILE file1=_lopen(FileListBox1->FileName.c_str(),OF_READ);
      if (file1!=NULL) _lread(file1,buffer,size);

/*    if(OpenDialog1->Execute())
        HFILE file1=_lopen(OpenDialog1->FileName.c_str(),OF_READ);
      if(file1!=NULL)
      _lread(file1,buffer,size);          */

      HFILE file2=_lopen("ori.mid",OF_WRITE);
      _lwrite(file2,buffer,size);

      HFILE file3=_lopen("fra.mid",OF_WRITE);

      for(int i=0;i<k;i++){
            aa[i]=buffer[70+6*i];
            bb[i]=buffer[72+6*i];

            if(bb[i]==60)
                        k=k-1;
            if(bb[i]==90)
                        k=k-2;
            if(bb[i]==120)
                        k=k-3;
      }

      Memo1->Lines->Add(" ");
```

```
Memo1->Lines->Add(" ");
Memo1->Lines->Add(" ");
Memo1->Lines->Add(" ");

for(int i=0;i<k;i++){
        switch(aa[i]){
                case 72:Memo1->Lines->Add("C"); break;
                case 73:Memo1->Lines->Add("C#");break;
                case 74:Memo1->Lines->Add("D"); break;
                case 75:Memo1->Lines->Add("D#");break;
                case 76:Memo1->Lines->Add("E"); break;
                case 77:Memo1->Lines->Add("F"); break;
                case 78:Memo1->Lines->Add("F#");break;
                case 79:Memo1->Lines->Add("G"); break;
                case 80:Memo1->Lines->Add("G#");break;
                case 81:Memo1->Lines->Add("A"); break;
                case 82:Memo1->Lines->Add("A#");break;
                case 83:Memo1->Lines->Add("B"); break;
                case 84:Memo1->Lines->Add("+C");break;
        }

        switch(bb[i]){
                        case 30:Memo1->Lines->Add("1/2");break;
                        case 60:Memo1->Lines->Add("1");break;
                        case 90:Memo1->Lines->Add("3/2");break;
                        case 120:Memo1->Lines->Add("2");break;
        }
}

for (int i=1;i<=640;i++){
        for (int j=1;j<=480;j++)
                Canvas->Pixels[i][j]=clBlack;
}

int j=0;
sum=0;
for(int i=0;i<k;i++){
       sum=sum+bb[i];
       if((sum%240)==0){
                        kk[j]=i+1;
```

```
                j++;
        }
}

float i=(ScrollBar1->Position)*100+1;

randomize();

//change the beats
for(int e=0;e<k;e++){
    switch(bb[e]){
        case 30: xy[e][0]='x';xy[e][1]='x';xy[e][2]='x';xy[e][3]='y'; break;
        case 60: xy[e][0]='x';xy[e][1]='x';xy[e][2]='y';xy[e][3]='y'; break;
        case 90: xy[e][0]='x';xy[e][1]='y';xy[e][2]='y';xy[e][3]='y'; break;
        case 120:xy[e][0]='y';xy[e][1]='y';xy[e][2]='y';xy[e][3]='y';break;
    }
}

for(int e=0;e<4;e++){
      if(e==0)
            r=rand()%kk[e];
      else
            r=kk[e-1] + rand()%(kk[e]-kk[e-1]);

      for(int q=0;q<4;q++)
          b[q]=xy[r][q];

      for(float q=0;q<i;q++){
          if(e==0)
                r=rand()%kk[e];
          else
                r=kk[e-1] + rand()%(kk[e]-kk[e-1]);

          for(int h=0;h<4;h++){
                r2=rand()%2;

                if(r2==0)
                      b[h]=xy[r][h];
          }
      }
```

```
        if(e==0){
                for(int q=0;q<kk[e];q++){
                        for(int h=0;h<4;h++){
                                r2=rand()%2;

                                if(r2==0)
                                        b[h]=xy[r][h];
                        }
                        bb[q]=0;

                        for(int w=0;w<4;w++){
                                if(b[w]=='y')
                                        bb[q]=bb[q]+30;
                        }
                }
        }

        else{
                for(int q=0;q<(kk[e]-kk[e-1]);q++){
                        for(int h=0;h<4;h++){
                                r2=rand()%2;

                                if(r2==0)
                                        b[h]=xy[r][h];
                        }
                        bb[kk[e-1]+q]=0;

                        for(int w=0;w<4;w++){
                                if(b[w]=='y')
                                        bb[kk[e-1]+q]=bb[kk[e-1]+q]+30;
                        }
                }
        }
}

for(int e=0;e<k;e++)
        buffer[72+6*e]=bb[e];
```

```
        _lwrite(file3,buffer,size);
        _lclose(file1);
        _lclose(file2);
        _lclose(file3);

        Memo1->Lines->Add(" ");
        Memo1->Lines->Add("OK");
}
//---------------------------------------------------------------------------

void __fastcall TForm1::Button12Click(TObject *Sender)
{

        //char    *str=Label4->Caption.c_str();
        const int size=1024;
        char buffer[size];
        double R;
        char    aa[32],bb[32];
        double max,min,aver;
        int p,b[4];
        int r;
        int sum,kk[4];
        int k=32;

        memset(buffer,0,size);

        HFILE file1=_lopen(FileListBox1->FileName.c_str(),OF_READ);
        if (file1!=NULL) _lread(file1,buffer,size);

        /*HFILE file1=_lopen(str,OF_READ);
        if (file1!=NULL)
                _lread(file1,buffer,size);*/

        HFILE file2=_lopen("ori.mid",OF_WRITE);
        _lwrite(file2,buffer,size);

        HFILE file3=_lopen("fra.mid",OF_WRITE);

        for(int i=0;i<k;i++){
```

```
        aa[i]=buffer[70+6*i];
        bb[i]=buffer[72+6*i];

        if(bb[i]==60)
                k=k-1;
        if(bb[i]==90)
                k=k-2;
        if(bb[i]==120)
                k=k-3;
}

Memo1->Lines->Add(" ");
Memo1->Lines->Add(" ");
Memo1->Lines->Add(" ");
Memo1->Lines->Add(" ");

for(int i=0;i<k;i++){
        switch(aa[i]){
                case 72:Memo1->Lines->Add("C"); break;
                case 73:Memo1->Lines->Add("C#");break;
                case 74:Memo1->Lines->Add("D"); break;
                case 75:Memo1->Lines->Add("D#");break;
                case 76:Memo1->Lines->Add("E"); break;
                case 77:Memo1->Lines->Add("F"); break;
                case 78:Memo1->Lines->Add("F#");break;
                case 79:Memo1->Lines->Add("G"); break;
                case 80:Memo1->Lines->Add("G#");break;
                case 81:Memo1->Lines->Add("A"); break;
                case 82:Memo1->Lines->Add("A#");break;
                case 83:Memo1->Lines->Add("B"); break;
                case 84:Memo1->Lines->Add("+C");break;
        }

        switch(bb[i]){
                case 30:Memo1->Lines->Add("1/2");break;
                case 60:Memo1->Lines->Add("1");break;
                case 90:Memo1->Lines->Add("3/2");break;
                case 120:Memo1->Lines->Add("2");break;
        }
}
```

```
for (int i=1;i<=640;i++){
        for (int j=1;j<=480;j++)
                Canvas->Pixels[i][j]=clBlack;
}

R=(ScrollBar3->Position)*0.01+0.01;

max=0;
min=127;
for(int i=0;i<k;i++){
      if(aa[i]>max)
                max=aa[i];
      if(aa[i]<min)
                min=aa[i];
}
aver=(max+min)/2;

for(int i=0;i<k;i++)
        Canvas->Pixels[i*15+10][240-(aa[i]-aver)*10] = clYellow;

for(int i=0;i<k;i++){
        if(aa[i]>aver)
                aa[i]=aver + (aa[i]-aver)/R;
        if(aa[i]<aver)
                aa[i]=aver - (aver-aa[i])/R;
}

for(int i=0;i<k;i++)
        Canvas->Pixels[i*15+10][240-(aa[i]-aver)*10] = clWhite;

int j=0;
sum=0;
for(int i=0;i<k;i++){
      sum=sum+bb[i];
      if((sum%240)==0){
                kk[j]=i+1;
                j++;
      }
}
```

```
float i=(ScrollBar1->Position)*100+1;

//change the melody
for(int e=0;e<4;e++){
	if(e==0)
		r=rand()%kk[e];
	else
		r=kk[e-1] + rand()%(kk[e]-kk[e-1]);

	p=aa[r];

	for(float q=0;q<i;q++){
		if(e==0)
			r=rand()%kk[e];
		else
			r=kk[e-1] + rand()%(kk[e]-kk[e-1]);

		p=p*R + aa[r]*(1-R);
	}

	if(e==0){
		for(int g=0;g<kk[e];g++){
			r=rand()%kk[e];

			p=p*R + aa[r]*(1-R);

			buffer[70+6*g]=p;
			buffer[73+6*g]=p;
		}
	}
	else{
		for(int g=0;g<kk[e]-kk[e-1];g++){
			r=kk[e-1] + rand()%(kk[e]-kk[e-1]);

			p=p*R + aa[r]*(1-R);

			buffer[70+6*(g+kk[e-1])]=p;
			buffer[73+6*(g+kk[e-1])]=p;
		}
```

```
            }
        }

        _lwrite(file3,buffer,size);
        _lclose(file1);
        _lclose(file2);
        _lclose(file3);

        Memo1->Lines->Add(" ");
        Memo1->Lines->Add("OK");
}
//---------------------------------------------------------------------------

void __fastcall TForm1::Button13Click(TObject *Sender)
{

        //char    *str=Label4->Caption.c_str();
        //check the address of a midi file
        const int size=1024;
        char buffer[size];
        double R;
        char    aa[32],bb[32];
        double max,min,aver;
        int p,b[4];
        int r,r2;
        int sum,kk[4];
        char xy[32][4];
        int k=32;

        memset(buffer,0,size);

        HFILE file1=_lopen(FileListBox1->FileName.c_str(),OF_READ);
        if (file1!=NULL) _lread(file1,buffer,size);

        /*HFILE file1=_lopen(str,OF_READ);
        //read a midi file
        if (file1!=NULL)
                _lread(file1,buffer,size);*/
        HFILE file2=_lopen("ori.mid",OF_WRITE);
        _lwrite(file2,buffer,size);
```

```
HFILE file3=_lopen("fra.mid",OF_WRITE);

for(int i=0;i<k;i++){
	aa[i]=buffer[70+6*i];
	bb[i]=buffer[72+6*i];

	if(bb[i]==60)
		k=k-1;
	if(bb[i]==90)
		k=k-2;
	if(bb[i]==120)
		k=k-3;
}
//check the number of the notes

Memo1->Lines->Add(" ");
Memo1->Lines->Add(" ");
Memo1->Lines->Add(" ");
Memo1->Lines->Add(" ");

for(int i=0;i<k;i++){
	switch(aa[i]){
		case 72:Memo1->Lines->Add("C"); break;
		case 73:Memo1->Lines->Add("C#");break;
		case 74:Memo1->Lines->Add("D"); break;
		case 75:Memo1->Lines->Add("D#");break;
		case 76:Memo1->Lines->Add("E"); break;
		case 77:Memo1->Lines->Add("F"); break;
		case 78:Memo1->Lines->Add("F#");break;
		case 79:Memo1->Lines->Add("G"); break;
		case 80:Memo1->Lines->Add("G#");break;
		case 81:Memo1->Lines->Add("A"); break;
		case 82:Memo1->Lines->Add("A#");break;
		case 83:Memo1->Lines->Add("B"); break;
		case 84:Memo1->Lines->Add("+C");break;
	}
	//show the notes of the midi file

	switch(bb[i]){
```

```
                case 30:Memo1->Lines->Add("1/2");break;
                case 60:Memo1->Lines->Add("1");break;
                case 90:Memo1->Lines->Add("3/2");break;
                case 120:Memo1->Lines->Add("2");break;
        }
        //show the length of the notes
}

for (int i=1;i<=640;i++){
        for (int j=1;j<=480;j++)
                Canvas->Pixels[i][j]=clBlack;
}
//clean up a piece of room for drawing

max=0;
min=127;
for(int i=0;i<k;i++){
      if(aa[i]>max)
                max=aa[i];
      if(aa[i]<min)
                min=aa[i];
}
aver=(max+min)/2;
//find the highest and the lowest notes, and average them

for(int i=0;i<k;i++)
        Canvas->Pixels[i*15+10][240-(aa[i]-aver)*10] = clYellow;
//draw the notes on the dark space

R=(ScrollBar3->Position)*0.01+0.01;

for(int i=0;i<k;i++){
        if(aa[i]>aver)
                aa[i]=aver + (aa[i]-aver)/R;
        if(aa[i]<aver)
                aa[i]=aver - (aver-aa[i])/R;
}
//make the gap of each notes bigger

for(int i=0;i<k;i++)
```

```
        Canvas->Pixels[i*15+10][240-(aa[i]-aver)*10] = clWhite;
//draw the notes which are changed

int j=0;
sum=0;
for(int i=0;i<k;i++){
      sum=sum+bb[i];
      if((sum%240)==0){
                  kk[j]=i+1;
                  j++;
      }
}
//confirm how many notes are there in each measures

float i=(ScrollBar1->Position)*100+1;

randomize();

//change the beats
for(int e=0;e<k;e++){
    switch(bb[e]){
        case 30: xy[e][0]='x';xy[e][1]='x';xy[e][2]='x';xy[e][3]='y'; break;
        case 60: xy[e][0]='x';xy[e][1]='x';xy[e][2]='y';xy[e][3]='y'; break;
        case 90: xy[e][0]='x';xy[e][1]='y';xy[e][2]='y';xy[e][3]='y'; break;
        case 120:xy[e][0]='y';xy[e][1]='y';xy[e][2]='y';xy[e][3]='y';break;
    }
}
//write the genetic expression of the beats of each notes

for(int e=0;e<4;e++){
      if(e==0)
                  r=rand()%kk[e];
      else
                  r=kk[e-1] + rand()%(kk[e]-kk[e-1]);

      for(int q=0;q<4;q++)
            b[q]=xy[r][q];
      //choose a random beats for beginning

      for(float q=0;q<i;q++){
```

```
        if(e==0)
                r=rand()%kk[e];
        else
                r=kk[e-1] + rand()%(kk[e]-kk[e-1]);

        for(int h=0;h<4;h++){
                r2=rand()%2;

                if(r2==0)
                        b[h]=xy[r][h];
        }
}
//do the "genetic algorithm" for times

if(e==0){
        for(int q=0;q<kk[e];q++){
                for(int h=0;h<4;h++){
                        r2=rand()%2;

                        if(r2==0)
                                b[h]=xy[r][h];
                }
                bb[q]=0;

                for(int w=0;w<4;w++){
                        if(b[w]=='y')
                                bb[q]=bb[q]+30;
                }
        }
}

else{
        for(int q=0;q<(kk[e]-kk[e-1]);q++){
                for(int h=0;h<4;h++){
                        r2=rand()%2;

                        if(r2==0)
                                b[h]=xy[r][h];
                }
                bb[kk[e-1]+q]=0;
```

```
                    for(int w=0;w<4;w++){
                        if(b[w]=='y')
                            bb[kk[e-1]+q]=bb[kk[e-1]+q]+30;
                    }
                }
        }
        //do the "genetic algorithm" for several times for each notes
}

for(int e=0;e<k;e++)
        buffer[72+6*e]=bb[e];
//write the new beats into the midi file

randomize();

//change the melody
for(int e=0;e<4;e++){
        if(e==0)
                r=rand()%kk[e];
        else
                r=kk[e-1] + rand()%(kk[e]-kk[e-1]);

        p=aa[r];
        //choose a note for begining

        for(float q=0;q<i;q++){
                if(e==0)
                        r=rand()%kk[e];
                else
                        r=kk[e-1] + rand()%(kk[e]-kk[e-1]);

                p=p*R + aa[r]*(1-R);
        }
        //do the "Iterated Function Systems" for times

        if(e==0){
                for(int g=0;g<kk[e];g++){
                        r=rand()%kk[e];
```

```
                        p=p*R + aa[r]*(1-R);

                        buffer[70+6*g]=p;
                        buffer[73+6*g]=p;
                }
        }
        else{
                for(int g=0;g<kk[e]-kk[e-1];g++){
                        r=kk[e-1] + rand()%(kk[e]-kk[e-1]);

                        p=p*R + aa[r]*(1-R);

                        buffer[70+6*(g+kk[e-1])]=p;
                        buffer[73+6*(g+kk[e-1])]=p;
                }
        }
        //do the "Iterated Function Systems" for each notes
    }

    _lwrite(file3,buffer,size);
    //write the changed notes into the new file

    _lclose(file1);
    _lclose(file2);
    _lclose(file3);

    Memo1->Lines->Add(" ");
    Memo1->Lines->Add("OK");
}
//---------------------------------------------------------------------------


void __fastcall TForm1::Button14Click(TObject *Sender)
{
  mciSendString("play bg2.mid",NULL,1024,0);
}
//---------------------------------------------------------------------------

void __fastcall TForm1::Button15Click(TObject *Sender)
{
```

```
  mciSendString("play bg3.mid",NULL,1024,0);
}
//---------------------------------------------------------------------------
```

評語

1. 寫作程式以產生 Sierpinski Gasket 與 Sierpinski Carpet 的各種樣態，This is neat work。
2. 將 Sierpinski 的概念應用於現有好聽的樂曲(用鋼琴彈奏)以產生新的音樂。
3. 改進方向(1)可以思考如何判斷所產生的新樂曲，如何判斷是否悅耳。(2)有無其他應用。