Chapter11

玩聲音，數位音訊的無限可能

/ 林宜徵

程式與音樂是一個乍聽沒有關聯，但實際上已結合許久的應用領域。本章將介紹數位音訊原理、以程式處理音訊，以及運用圖像式工具編輯 MIDI 訊號和製作簡單合成器。本章共有四小節，包括 MIDI 訊號讀取分析、PureData 音樂編輯程式與 MIDI 創作、音訊原理與頻譜檢視和 PureData 合成器製作。學習完本章後，讀者將能撰寫程式製作 wav 音訊，也能以 PureData 工具進行簡易音樂編輯與創作。

緒論

數位音樂一般來說包含兩種資料型態，一為音訊（如 .wav 或 .aiff 等），一為 MIDI 音訊。若要做任何數位相關的音樂處理，無論是創作或分析，皆無法離開此二種資料型態的範疇。本章分別介紹兩種音樂數位訊號（音波與 MIDI），每一主題段落內之內容又區分為：(一) 用基礎程式語言和套件讀取基本數位訊號的格式和訊息；(二) 以高階的圖像化音樂編輯程式指導如何編排和應用這些數位訊號。本章內容主軸從音樂資料的分析到運用高階的圖像化程式實作較為複雜的練習作業。若無高階的圖像化程式輔助，僅用一般程式語言處理的話，音樂的製作過程會相當冗長費時與無法掌握實作成果，因此在業界，採用數位音樂工作站或圖像化程式是最普遍的做法。以下是本章後續內容的簡要清單。一般音樂製作人在創作編曲時多半仰賴 MIDI 訊號的編輯，MIDI 相對屬於需音樂背景知識方能掌握之訊號；而音訊處理則是屬於聲響工程或合成器開發之工程人員所熟悉的領域，需工程數學背景知識方能掌握。讀者僅需知道此二種格式的特性以及能以程式讀取或以軟體作基礎編輯運用即可，不需詳記相關算式或訊號格式。

11-1 簡介 MIDI 與 MIDI 訊號讀取分析

自從 MIDI 訊號問世後，音樂產業中音樂的製作流程可說是受到極大的衝擊和影響，過去寫譜、印譜、排練、演出、錄音的音樂生產流程，轉變為 MIDI 訊號作曲、編曲、部分聲部（如主奏樂器）錄棚、混音（於電腦上透過音訊部分頻率音量的調整平衡）、最後過帶輸出。流程上大幅減少對於演奏者錄製音樂的需求，影響產業甚鉅！

音樂資料類型概述與基本 MIDI 訊號讀取分析

- Digital Audio Workstation 簡介
- 簡介 MIDI 與 MIDI 訊號讀取分析

古代創作音樂的方式

- 作曲家創作（可經過出版商出版）
- 演奏家練習
- 演奏家演出

現代數位音樂創作音樂製作方式與流程

- 作曲（主旋律）
- 編曲（配器）、伴奏等 loop
- 錄棚（主唱、真實樂器演奏）
- 混音
- 過帶（輸出）
- 出版發行
- 上述流程除了錄棚外，其餘可由 Digital Audio Workstation (DAW) 完成。

聲音編輯的兩種基本方式：MIDI 與音訊

音樂數位介面（Musical Instrument Digital Interface，簡稱 MIDI）是一個工業標準的電子通訊協定。MIDI 並未帶有任何音訊訊息，僅有音符開始、結束之時間點、力度、音色、拍號等資料。

- 簡介 MIDI 與 MIDI 訊號讀取分析

上網搜尋 : MIDI to Text online converter

試試看：http://flashmusicgames.com/midi/mid2txt.php

上網搜尋任一 MIDI sequencer

試試看：https://onlinesequencer.net/import

Play with Chrome Music Lab:

試試看：https://musiclab.chromeexperiments.com/Song-Maker/

- Music21

Music21 為一款由美國麻省理工學院製作之處理 MIDI 訊號之套件，其有完整功能可協助分析或編輯音樂 MIDI 資料。

Music21 實作方式

1. 打開 Google Colab(須註冊 Google）

https://colab.research.google.com/notebooks/welcome.ipynb

2. 在 Google Colab 內新增一個 Python3 之記事本。

圖 11-1 新增記事本

將以下程式碼複製貼上，然後按左側三角執行。

E11-2-1.py

```
01 !pip3 install music21
02 from music21 import *
03 environment.set('autoDownload','allow' )
04 s = converter.parseURL('http://yuan.yocjh.kh.edu.tw/midi/td.mid')
05 s.show('text')
```

圖 11-2

程式碼解說

安裝 Music 21 之環境。

```
01 !pip3 install music21
```

把 Music21 載入使用。

```
02 from music21 import *
```

Music21 自動下載 URL 資料設定。

```
03 environment.set('autoDownload','allow' )
```

自動下載 URL 的 MIDI 檔案並轉成 Music21 之格式。

```
04 s = converter.parseURL('http://yuan.yocjh.kh.edu.tw/midi/td.mid')
```

以文字顯現 MIDI 檔案之內容。

```
05 s.show('text')
```

Digital Audio Workstation (DAW)

課後可自行嘗試：下載任一免費或試用版本之 DAW: Garage Band, Pro Tools, Reaper, Cubase 等，在其官網或 youtube 觀看教學影片，並自行嘗試使用 piano roll 作點旋律。

小結

運用 Colab 雲端服務，可以讓創建 Python 環境簡單一致化，不用擔心每台電腦程式環境不一致的問題。

讀者僅需知道如何在參考 Music21（或其他類似第三方套件）的官方說明下能正確讀取 MIDI 檔案即可，不需熟記套件的指令。

透過此節，讀者先從運用他人的 MIDI 讀取網站，到實際運用第三方套件撰寫程式碼讀取 MIDI，了解 MIDI 訊號提供的內容後，再用編曲軟體以 MIDI 訊號編曲，將更對 MIDI 訊號更熟悉且對加強學習樂趣。讀者可以嘗試同時輸入兩三首樂曲的旋律，或是用 piano roll 畫上簡單的圖形（也可運用不同的力度在 DAW 上為圖形『著色』），聽聽看，會很有趣的！

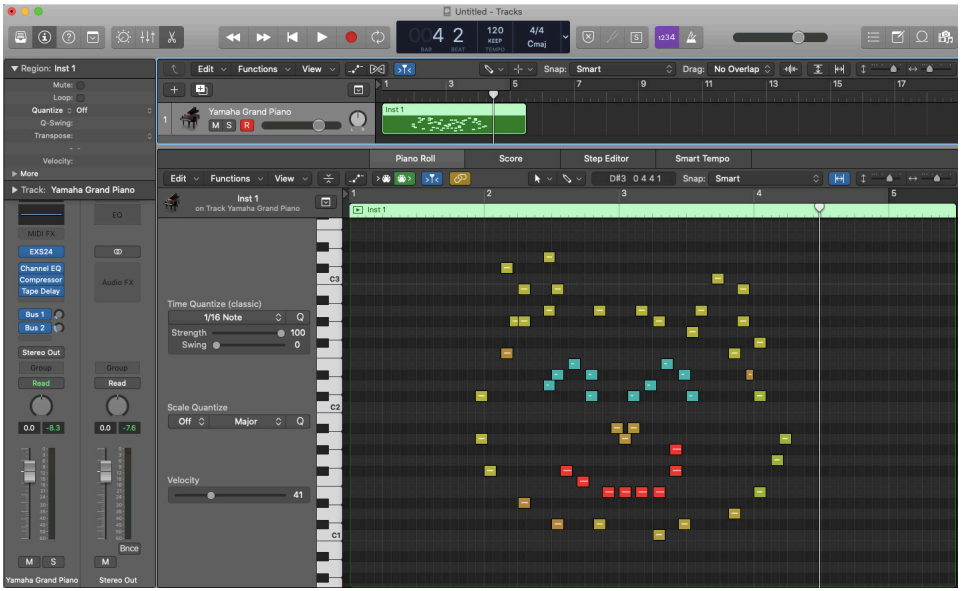

圖 11-3 用 piano roll 畫圖著色做音樂。

隨堂練習

11.1.1 . 請問以下對 MIDI 訊號之敘述，何者為非？

(A) 是一個工業標準的電子通訊協定。

(B) 不需經過 DAW 等軟體的音源設定，讀取後可直接由電腦產生聲響，作為跨平台使用，非常方便。

(C) 有音符開始、結束之時間點、力度、音色、拍號等資料。

(D) 在 DAW 上作曲，用此種訊號創作很方便。

參考解答：B，MIDI 訊號和音訊最大的差異就是 MIDI 不帶有任何音訊在裡面，因此在未設定音源和過帶的狀況下， MIDI 訊號無法產生任何聲音。

11.1.2 . 以下關於 Digital Audio Workstation 的敘述何者為非？

(A) 可用來錄音。

(B) 可用來創作。

(C) 僅能讀取 MIDI 檔案，無法讀取聲音訊號。

(D) 有了 DAW 後，音樂市場對於演奏者的需求減少了。

參考解答：C，DAW 可用來處理 MIDI 和音訊檔案格式，且此二種檔案皆能再其上做效果器或混音等處理。

11-2Pure Data（免費版）圖像化音樂編輯程式與 MIDI 創作

Pure Data 歷史、下載

Pure Data（或稱作 PD）是米勒・帕克特在 90 年代為創造交互的計算機音樂和多媒體作品而開發的視覺化程式設計語言。雖然帕克特是 PD 的主要作者，但是它是一個多數開發者一起開發新擴展的開放原始碼項目。它以一個類似於 BSD 許可證下發行，可運行在 GNU/Linux、Mac OS X、iOS、Android 和 Windows。Max/MSP 屬於 Pure Data 商業化的版本，讀者可自行斟酌是否採用。

Pure Data 下載

請至 https://puredata.info/ 下載對應 OS 之版本→下載後解壓縮→打開解壓縮後的檔案。

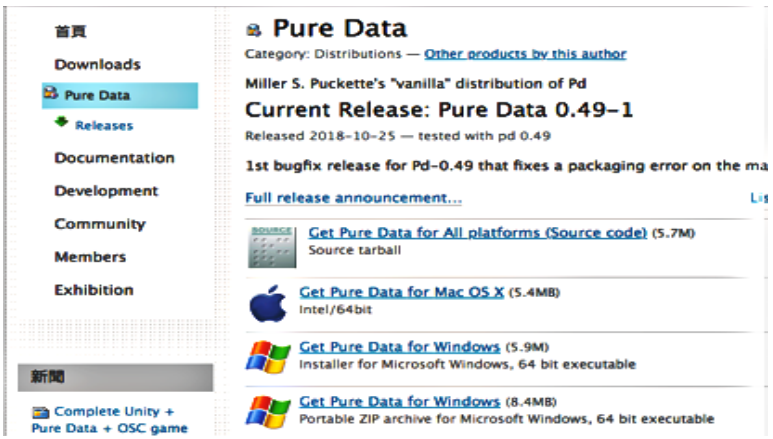

圖 11-4

PD 資料型態介紹

1. PD 有自己獨特的資料型態，包括： Object、Message、 Number、Symbol 和 Comment。
2. 可用右邊的快捷鍵建立各種資料型態之物件。
3. 將這些物件連接起來，行程 Patch，便可執行許多數學運算、MIDI、甚至音訊等運算。

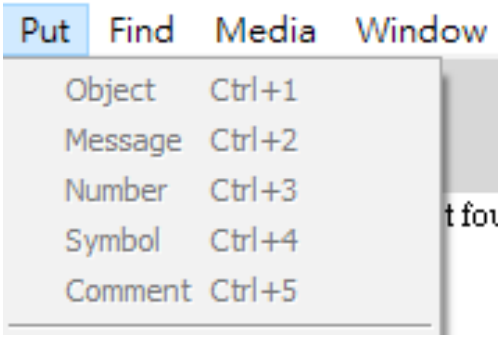

圖 11-5

Object

最主要資料型態，屬於傳統程式語言之 function 類別，本身具有邏輯運算的功能，當輸入正確的 API （與參數）時，會出現帶有粗黑短線條之黑色框，粗黑短線條為訊號輸入和輸出之接口，上端接口為輸入端，下為輸出端 。所有 Object List 可查詢：http://blazicek.net/list_of_pure_data_objects.html。

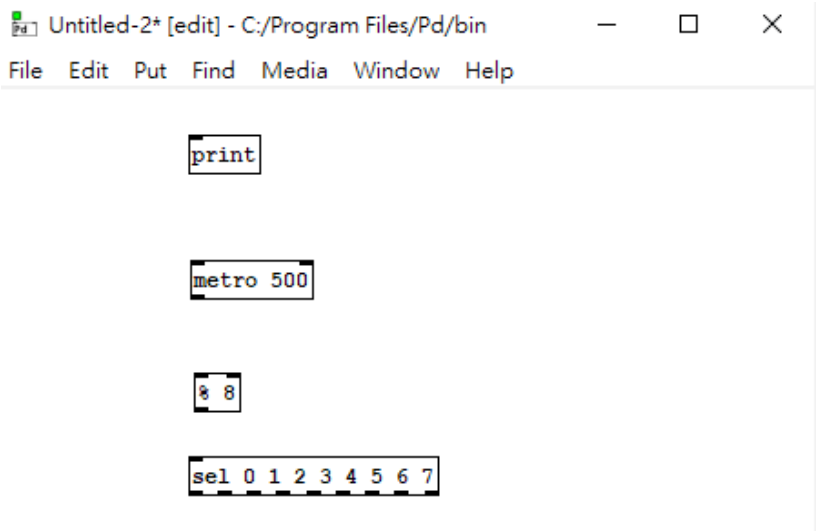

圖 11-6

Message

Message 屬於靜態字串類別，旗標狀，左側上下各有粗線接口，上為輸入端、下為輸出端。

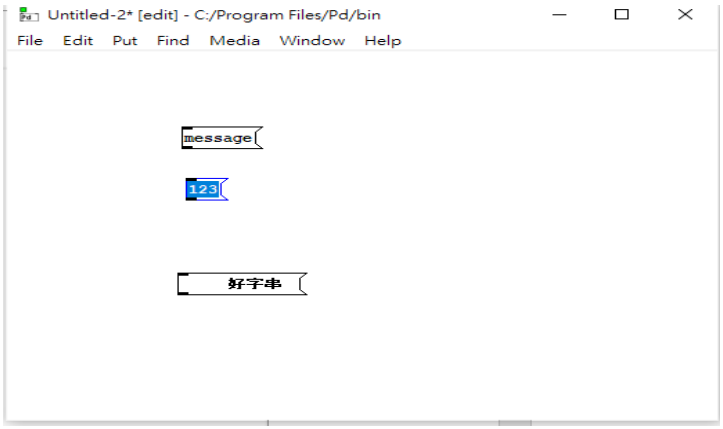

圖 11-7

Number

Number 屬於數字類別，可有負數、浮點值，右上方有缺角之框格狀，左側上下各有粗線接口，上為輸入端，下為輸出端，在 edit mode 中只能呈現預設值 0，切換到 play mode 時（crl+e or cmd+e) ，以滑鼠左鍵點按於該數字框格上下拖曳後，可更改數字。

以 message 輸入連接浮點數字後，在 edit mode 點按 message 框格後，number 即變成為可拖曳更改之浮點值（圖中最右）

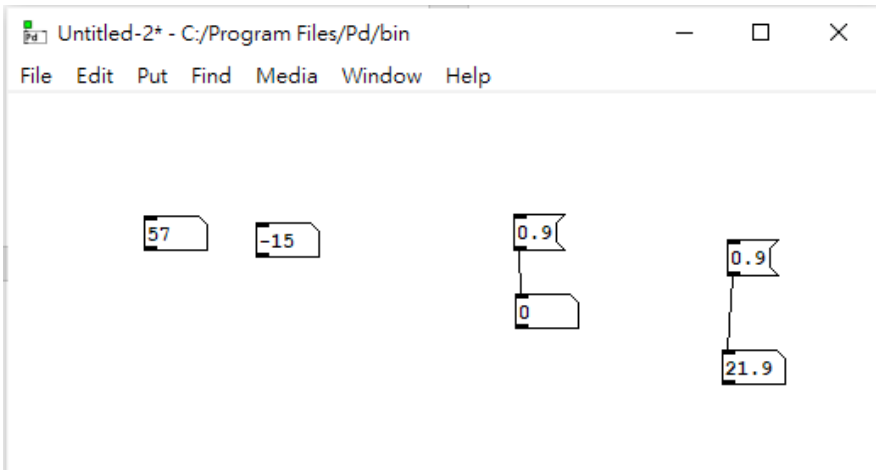

圖 11-8

Comment

1.Comment 可用來加註文字。

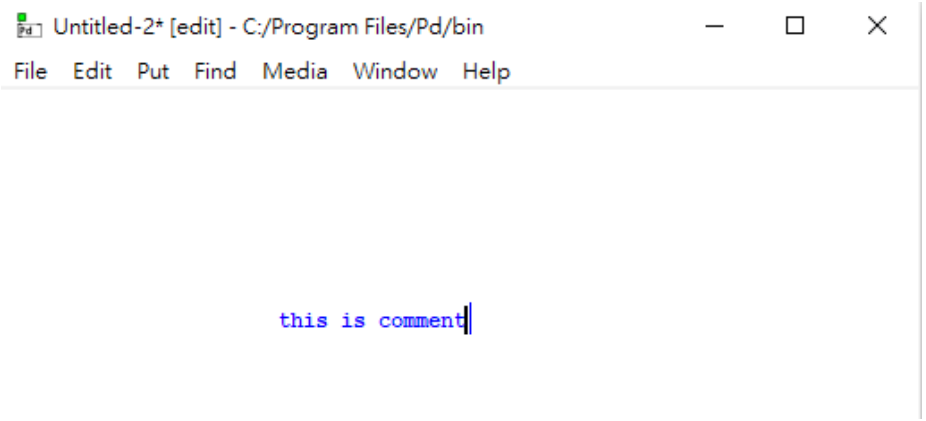

圖 11-9

Symbol

屬於 String，具有被快取（cached）或實際上永久儲存於 PD 的雜湊表中，可以被附上一些資料。PD 中有許多資訊是靠 Symbol 做辨認的，因此 Symbol 是一種非常重要的資料型態。一般來說初學 PD 較少使用 Symbol ，因此在此不多詳述，如有興趣了解請見以下資料連結： https://puredata.info/community/pdwiki/symbol

Help 功能

僅須將滑鼠以右鍵點按上述物件便能看到關於該物件的 *Help* 解釋。

PD 實作（旋律產生器）

在 Log 視窗勾選右上方的 DSP 來開啟音訊。依照下方圖示產生各種物件，並以滑鼠左鍵點按個物件的輸出或輸入端，以產生線條連結個物件。

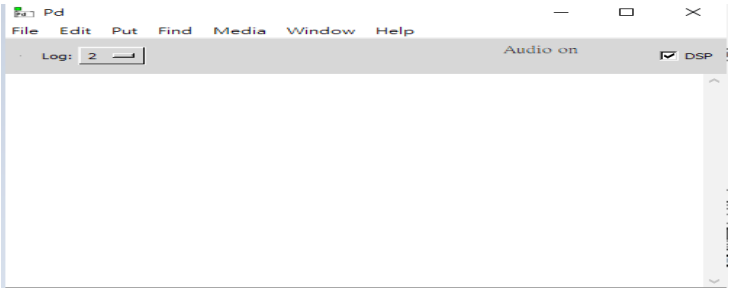

圖 11-10

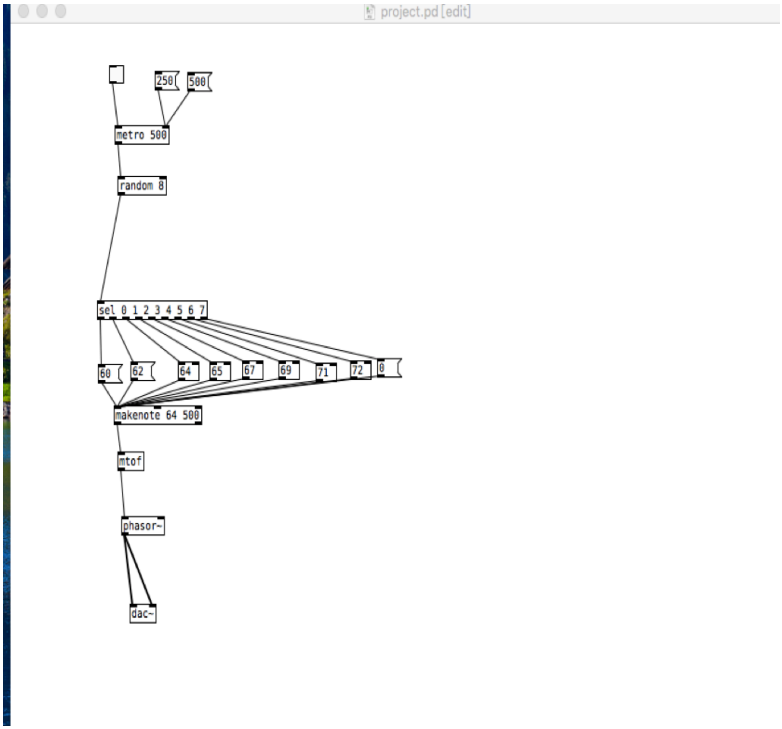

圖 11-11

小結

透過圖像式程式，同學們製作了第一個音樂的 AI 小工具，當我們撰寫程式，讓電腦產生類似人類的思考或判斷，就是人工智慧的範疇，AI 和我們的生活關係越來越密切，但終究是一個工具，如何設計工具和使用工具，最終也是由人的智慧去掌握的！

隨堂練習

11.2.1 . 請問以下關於 Pure Data 的敘述，何者為是？

(A) 是一種為創造交互的計算機音樂和多媒體作品而開發的視覺化程式設計語言。

(B) 為米勒・帕克特獨立開發的軟體 。

(C) 不支援 Mac OS。

(D) 無法處理 MIDI 訊號 。

參考解答：A

11.2.2. 請問以下 PD 的資料型態，何者為非？

(A) Object

(B) Message

(C) Comment

(D) Function

參考解答：D，PD 有自己獨特的資料型態，包括 : Object, Message, Number, Symbol, 和 Comment 。

11.2.3. 請問對於 PD 的操作方式，何者為是？

(A) 當需要使用註解時，可使用旗標狀的 Message 資料型態。

(B) 在 edit mode 下即可執行並更改 Number 資料的數值。

(C) 如要產生音訊，須在 Log 視窗勾選右上方的 DSP 來開啟音訊。

(D) Pure Data 為一款付費軟體 。

參考解答：C，初學者常忽略須在 Log 視窗勾選右上方的 DSP 方能開啟音訊

11.2.4. 請用 PD 撰寫一個程式，能同時產生二個旋律。

1. 二組旋律皆為五聲音階組成之旋律，而其中一組的音階高於 第二組一個八度。

2. 請用 orc~ 物件取代 phasor~ 物件，使音色較為悅耳。

3. 調整二個旋律的音量，使一組較大一組較小聲。

4. 利用音量為 0，設計一個二個旋律結束的開關。

5. 想想看還可加入什麼變化？

（提示：將課堂的練習再新增另一組，並結合在一起。五聲音階由 CDEGA 組成。）

參考解答：見附檔 project.pd

11-3 數位音訊原理 (基本音波撰寫) 與頻譜檢視

數位聲音訊號原理

聲音是一種波動，當演奏樂器、拍打一扇門或者敲擊桌面時，聲音的振動會引起介質 —— 空氣分子有節奏的振動，使周圍的空氣產生疏密變化，形成疏密相間的縱波，這就產生了聲波，這種現象會一直延續到振動消失為止。

一般的聲音總是包含一定的頻率範圍。人耳可以聽到的聲音的頻率範圍在 20 到 2 萬赫茲（Hz）之間（每秒震動循環次數）。高於這個範圍的波動稱為超音波，而低於這一範圍的稱為次聲波。

類比音訊與數位音訊

圖像有類比圖像（例如傳統底片式相片）與數位圖像的分別，類比圖像所有的線條為連續性，當一條斜線無限放大，還是斜線，但數位則是不連續性，當一條斜線被放大解讀，則會看到不連續之階梯狀線條。

類比音訊就比如日常生活的聲音，所有音波是連續的，但數位音訊則是指採取某些不連續的點以儲存入電腦中。

數位音訊採樣頻率

要將音波資料儲存入電腦等數位裝置，需要將一段音波的細分成許多點，將各點的聲壓數值依序儲存起來。細分成多少點，稱之為取樣頻率，有鑒於人類可聽頻率最高為兩萬赫茲，一個波循環，至少需有兩個取樣點，因此能聽頻率最高所需的取樣為 2 萬 乘以 2(取樣點）＝ 4 萬取樣點，依電腦的位元數，則為建議為每秒至少需取 44100 個取樣點以上，方能將人類能聽頻率全數收錄。

現行 CD 採樣頻率為 44100 Hz，其他如搭配影像或高音值聲音作品可將採樣頻率提高到 48000 ~ 192000 HZ 之間。採樣頻率越高，所需儲存空間則越多。

頻譜檢視與分析

用不同樂器演奏同樣的音高，聲音頻率雖然相同，但由於波形不同，形成不同的音色。各式各樣的波形，只要是循環波，便可以被分解為不同頻率、不同強度的正弦波的疊加。這種變換（或分解）的過程，稱為傅立葉變換。

透過傅立葉轉換成頻譜，我們可看到一個循環音波被分解成由低至高頻的多個正弦波之分布情形，以理解音色的特性。例如：聲音渾厚者的歌聲與聲音纖細者的歌聲，雖唱同樣音高，頻譜分析下，聲音渾厚者之低頻分佈會較多，聲音纖細者低頻分佈會較少。

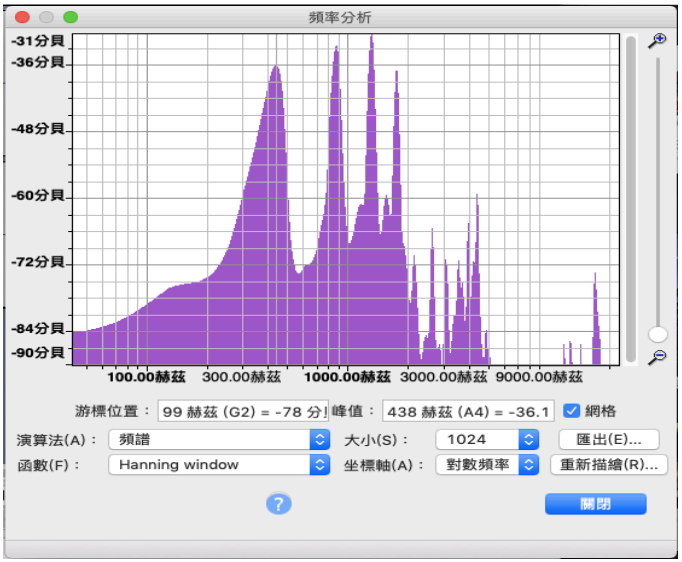

圖 11-12 頻譜範例

此為筆者人聲哼唱 標準音 A 音（440 Hz) 的頻譜分析，可見頻譜分佈並非僅於 440 赫茲，還有 440 的泛音與其他非泛音之雜訊頻率（沙啞聲）。越清亮渾厚的好歌喉產生的聲音，頻譜分佈會分佈在泛音列之上，非泛音列之雜訊頻率會較少。

下載 Audacity 與分析頻譜實作

Audacity 是一套免費的音頻分析與編輯軟體，可致其官方網站下載：https://www.audacityteam.org/download/。下載安裝並執行此軟體後，點按錄音鍵，便可用電腦的麥克風開始錄音，請哼唱一段小曲。

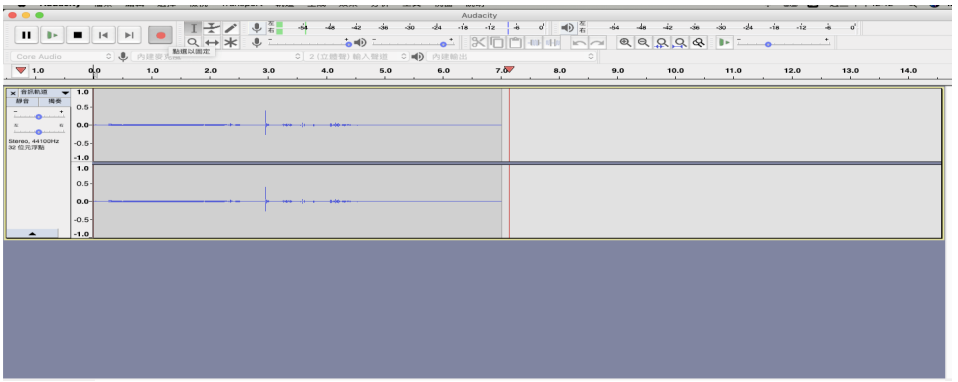

圖 11-13

點按上方紅色錄音鈕開始錄音，結束錄音請按停止鈕。

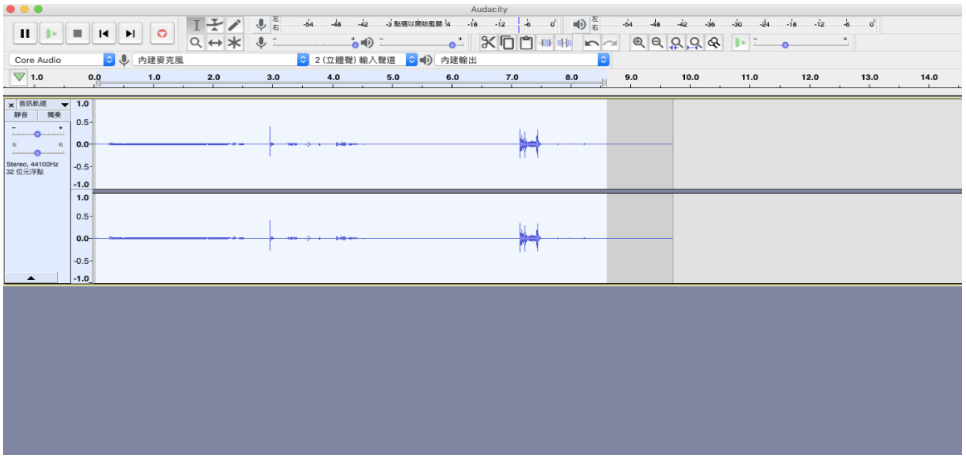

圖 11-14

點按停止鈕後，以滑鼠長按左鍵一邊選取想要分析的段落，被選取之段落會反白。選取後，於上方工具列中點按『分析 —> 描繪頻譜』。

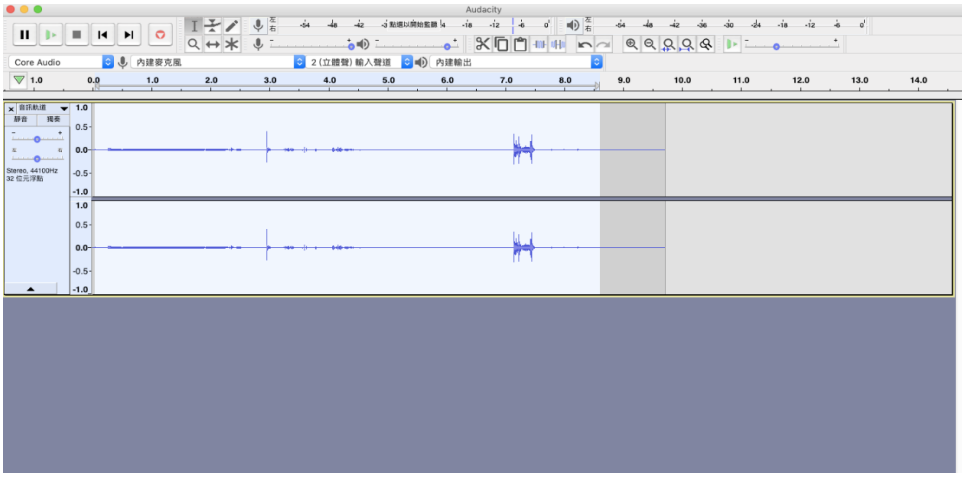

圖 11-16

完成之頻譜分析。

基本 sine wave 音訊程式撰寫

了解音訊原理後，接下來實際用 Python 來寫一個 sine wave 音頻訊號。

打開 Google Colab (須註冊 Google 帳號）。

https://colab.research.google.com/notebooks/welcome.ipynb

在 Google Colab 內新增一個 Python3 之記事本。

將以下程式碼複製貼上。

E11-3-1.py

```
01  import numpy as np
02  from scipy.io.wavfile import write
03  from scipy.io.wavfile import read
04  from google.colab import files
05  #sample rate per seconds
06  sps = 44100
07  #frequency
08  freq_hz = 440.0
09  # duration in seconds
10  duration_s = 4.0
11  amplitude = 0.3
12  each_sample_number = np.arange(duration_s * sps)
13  waveform = np.sin(2*np.pi * each_sample_number * freq_hz / sps)
14  waveform_volume = amplitude * waveform
15  waveform_intergers = np.int16(waveform_volume * 32767)
16  #32767 為振幅定值
17  write('sinewave.wav', sps, waveform_intergers)
18  read('sinewave.wav')
19  files.download('sinewave.wav')
```

01 行至 04 行為環境設定：scipy 套件內有一個 wavfile 可執行 wav 檔案的讀寫。

Sine wave 的數學式為：y(t) = A * sin(2*pi*f*t + phi)

先設定基本參數，sps （sample rate per seconds） = 44100；

欲產生的 sine wave 頻率 freq_hz = 440.0，時長 duration_s = 4.0（四 秒），振幅值 (A) 為 0-1，在此設定 0.3，amplitude = 0.3。

由於在數位處理時，必須給電腦每一個取樣的時間值 (t)， 因此對於每一次波的循環 t = 1/sps 為時 4 秒的 sine wave 將會有 4 * sps 的取樣點 (each_sample_number)，此外之後方便運算，將這些取樣點以 np.arange 方式計算。

套上公式：

```
waveform = np.sin(2*np.pi * each_sample_number * freq_hz /sps)
```

執行程式碼，便會自動下載一個 sinewave.wav 的檔案。

小結

	正弦波（sine wave） 是聲音的最基本單位，透過正弦波的組合，可組合成各種聲音，而這也是數位音訊的基本原理。

透過這節，讀者了解正弦波和聲音的關係，也能運用程式撰寫基本的正弦波，並轉成音訊檔。這些音訊的計算，屬於音訊處理的領域，為工程領域，並廣泛運用於聲響工程，如音響、合成器、聲音或音樂特徵擷取等專業之中。雖然音訊處理屬於艱深的工程領域，但讀者藉由本節課，可以對此數位聲響的運作原理有基本概念，進而對於音訊的多種格式，如 wav, mp3, aiff 等或是耳機或揚聲器響應頻率等基本生活相關知識，也能更了解與應用。

隨堂練習

11.3.1. 請問以下關於人耳可聽到的頻率，何者為非？

(A) 人耳可聽到的頻率平均為 20~20000Hz 之間 。

(B) 人耳對於各種中音頻的敏銳度較高音頻和低音頻為優。

(C) 人耳可以同時聽到多種音頻，並藉由頻率的差異辨認方向。

(D) 人耳可聽到的頻率範圍會隨年紀增加而遞增。

參考解答：D，範圍應隨年紀增加而遞減。

11.3.2. 請問以下關於數位音訊的敘述何者為是？

(A) 為連續的訊號。

(B) 揚聲器可以直接接收和讀取數位音訊。

(C) 依據人耳可聽範圍，數位音訊的採樣頻率至少應為 40000Hz 以上。

(D) 聲波最基礎的單位是方形波。

參考解答：C

11.3.3. 請問以下關於聲波的敘述何者為非？

(A) 不同樂器演奏相同音高，音色不同的原因是在於其波型不同。

(B) 一把好的木製樂器通常高頻的振幅（力度）較大，低頻較小。

(C) 當我們觀察聲波的圖形時， 須透過傅立葉轉換，方能計算出頻譜。

(D) 泛音是指與基音呈現倍率的頻率。

參考解答：B，通常好的木製樂器，通常低頻較為渾厚，聲音較具共鳴。

11.3.4. 請撰寫一個程式能產生白噪音（注意音量不要太大聲，振幅建議為 0.1，且揚聲器或耳機須先調整到最小聲，以免傷耳與損害到揚聲設備）提示：若把各種隨機的頻率疊加在一起，便能產生白噪音的效果。

E11-3-2.py

```
01  import numpy as np
02  from scipy.io.wavfile import write
03  from scipy.io.wavfile import read
04  from google.colab import files
05  import random
07  #sample rate per seconds
08  sps = 44100
10  # duration in seconds
11  duration_s = 4.0
13  waveform = np.arange (duration_s * sps)
14  each_sample_number = np.arange (duration_s * sps)
16  i = 0
17  while i <=500:
18    #frequency
19    freq_hz = 40000.0 * random.random()
20    waveform += np.sin(2*np.pi * each_sample_number * freq_hz / sps)
21    i+=1
22  amplitude = 0.1
23  waveform_volume = amplitude * waveform
24  waveform_intergers = np.int16(waveform_volume * 32767) #32767 為振幅定值
25  write('whitenoise.wav', sps, waveform_intergers)
26  read('whitenoise.wav')
27  files.download('whitenoise.wav')
```

請試著找出生活中的各種聲音，用頻譜分析後，找找看哪些聲音和白噪音類似？（提示：如風聲、海浪聲）

11-4 Pure Data 合成器製作

音訊疊加式合成器

一般日常聽到的器樂演奏之聲音，含有具循環性樂音頻率和非循環性的噪音頻率（例如擦弦、氣聲、或敲擊物件的噪音）所組成。可由正弦波的組合來模擬循環性樂音頻率的部分。循環性的樂音通常為某音的基音頻率以及該基因的倍數頻率（又稱泛音）所疊加合成，音訊上僅要將各取樣點的數值相加，便可製作。

PD 實作（音訊疊加式合成器）

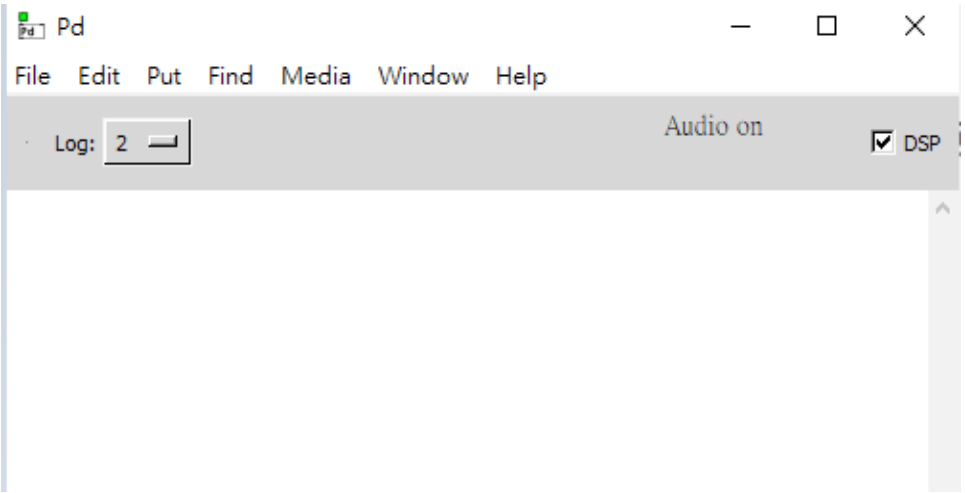

圖 11-18

在 *Log* 視窗勾選右上方的 *DSP* 來開啟音訊，依照上方圖示產生各種物件，並以滑鼠左鍵點按個物件的輸出或輸入端，以產生線條連結個物件。

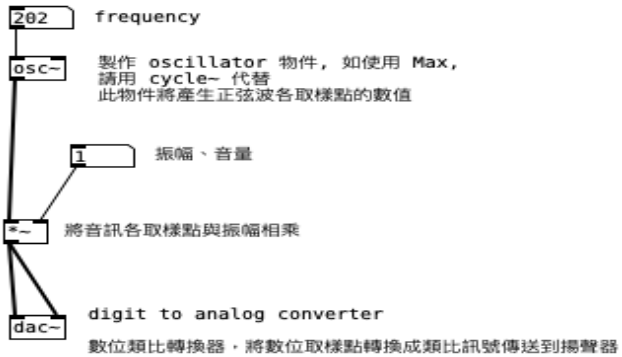

圖 11-19 單一正弦波

首先依照圖示製作一個單一的正弦波，運用 crtl + e 或是 comment + e 切換編輯 / 執行模式，在執行模式中，可用滑鼠左鍵拖曳調整 frequency 和振幅的數值，frequency 須在人耳可聽範圍方能被聽見，震幅則須介於 0~1 之間的浮點數。

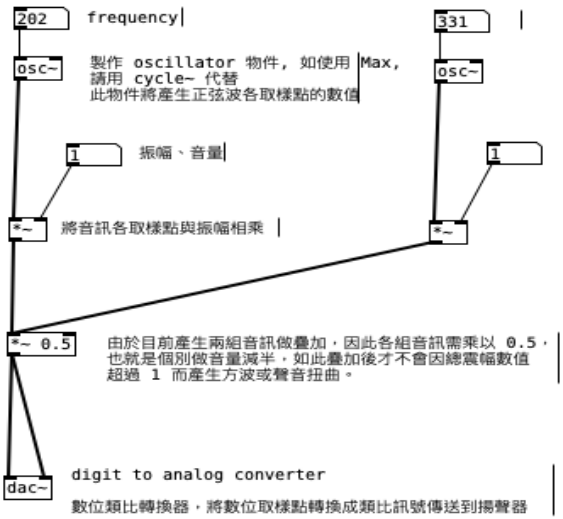

圖 11-20 第二組正弦波

可用滑鼠圈選欲重製的物件，按 crtl + c 以及 crtl + v 複製貼上，並拖曳到要放置的位置，以產生另一組正弦波。

由於兩組正弦波之振幅最高數值震幅為 1 的情況下，疊加後振幅為 2 ，因此需再將疊加後的振幅減半（乘以 0.5）使最高振幅不超過 1。

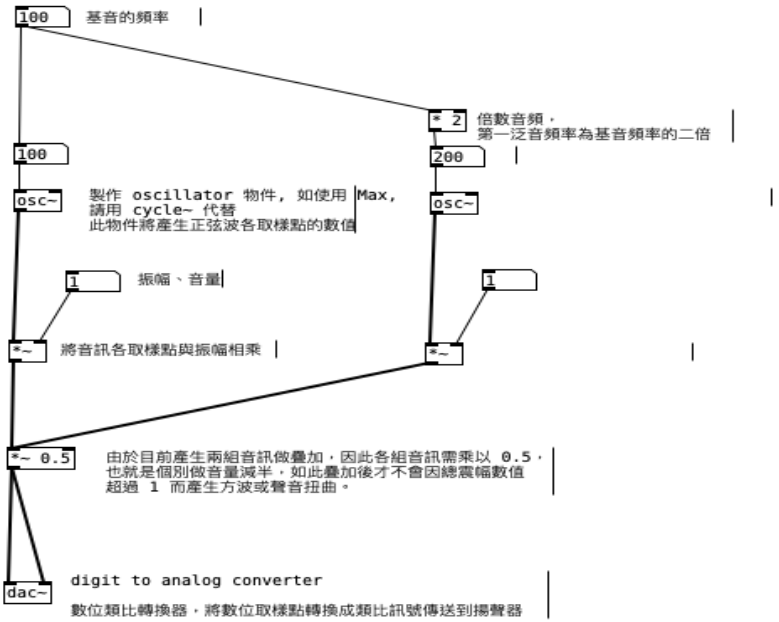

圖 11-21 使第二個正弦波與第一個正弦波為倍數關係

在上方加入一個數字物件作為基音的頻率，並連至第一個正弦波的頻率數字物件上（則第一個正弦波之頻率數字物件僅會重現基音頻率的數值）。

第二個正弦波上方加入相乘物件，在此因為要做第一泛音，其頻率為基音之兩倍，故乘以二，並將基音頻率物件連接到此相乘物件。

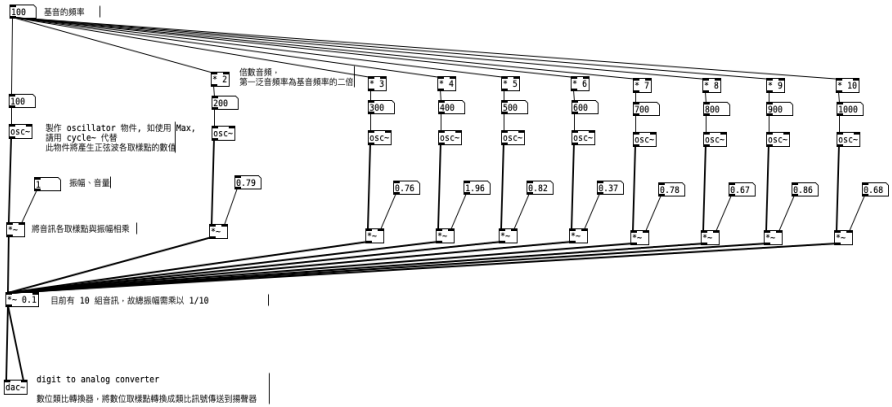

圖 11-22 製作其他九組泛音正弦波

- . 繼續製作其他九組泛音正弦波，並調整倍數值。
- 由於目前有 10 組正弦波，故將總振幅數值改乘以 0.1 (1/10)。
- 在執行模式下，對各組振幅數值之物件，按住 shift 鍵，並以滑鼠左鍵點按上下拖曳，便能產生浮點值（小數點），滑鼠的位置偏右一點，可調整小數點的位數。

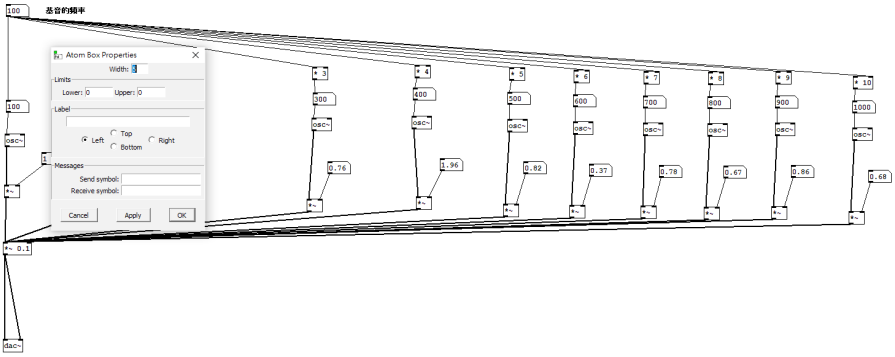

圖 11-23 設定數字物件的範圍

- 以滑鼠右鍵點按振幅數字物件，便會出現對話方塊，在方塊中設定數字物件的值範圍，在此設定為 1，如此在執行模式下，更易操縱。

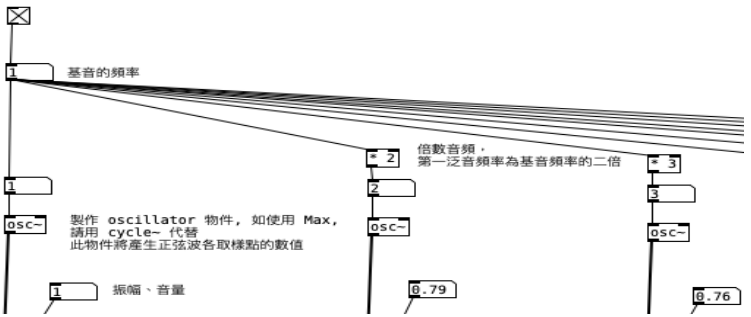

圖 11-24 總開關設置

- 在基音頻率上方新增一物件，並打上 toggle，此時便會自動產生方塊狀的開關物件。
- 將開關物件連接到基音頻率。
- 在執行模式下，可開啟開關（空格狀）或關閉開關（有 X 字狀）。
- 調整基音頻率，此時所有正弦波之數值將被連動，產生樂音。
- 透過調整各正弦波之振幅值，以改變音色，基音的振幅需高過其他泛音之振幅，如此，此總音波感知頻率才會維持在基音。

小結

由於音訊處理以 Python 程式語言撰寫相當不容易，且合成器製作與應用需要能即時播放的系統，因此許多數位音樂家或工程師在設計合成器時，會選擇 PD 或是 Max/MSP 的圖像式程式（但程式本身是以 C 語言為基礎作成的）創作，如此便可以越過訊號處理的計算知識直接進行聲響上的變化與選擇。除了圖像式語言，也有接近傳統程式語言的聲響程式，例如 SuperCollider, CSound

或 ChucK，而 Max/MSP 也能銜接 Pro Tools 等 DAW，對於有興趣鑽研數位音樂的讀者可深入研究這些程式工具。

參考資料

1. Google Colab 官方網站與使用說明：https://colab.research.google.com/
2. Music21 官方網站與說明：http://web.mit.edu/music21/
3. Pure Data 的操作方式，可參考以下連結：https://tuftsdev.github.io/MusicAppsOnTheIpad/readings/reading1.pdf
4. Pure Data 官方網站：https://puredata.info/
5. Audacity 官方網站與說明：https://www.audacityteam.org/
6. 資料引用：https://zh.wikipedia.org/wiki/%E5%A3%B0%E9%9F%B3