走音救星：把歌詞轉成唱歌的軟體

組員：陳羿豐　指導教授：蔡淳仁

一、　動機與目的

我很喜歡唱歌，也很喜歡聽同學唱歌，然而同學總是說他們不會唱，於是我就想著，能不能錄下同學們說話的聲音，然後用剪接軟體把說話轉成唱歌。我認為這是做得到的，因為我曾經用 Audacity 和 Praat 把 Google 翻譯的語音轉成歌聲，只是受限於我能取得的工具，當時花了非常久才完成。於是我決定開發只要唸出歌詞，就可以合成出歌聲的 App。

這個專題原本的名稱為「開發把朗讀變唱歌的軟體」，但是我覺得名字不夠吸引人，所以就換名稱。也許同學是怕走音才不敢唱歌吧。

二、　現有相關研究比較

市面上已經有多種合成歌聲的軟體：

1. Praat：是語音分析軟體，可用於分析音高、音量及共振峰等語音參數。[1] Praat 還提供操縱語音音高的功能，雖然合成歌聲不是其設計目的，但可用來合成歌聲。
2. VOCALOID：專業的歌聲合成軟體，初音未來就是用這個軟體創造的。然而音色只能由廠商提供，使用者不能創造音色檔。
3. UTAU：以剪接音檔來合成歌聲的軟體，可自行錄製音色。

以上的軟體中，只有 Praat 有開放原始碼，並且提供研究論文，因此本專題參考 Praat 合成語音的方法來實作。

三、　原理

Praat 合成歌聲的方式是 TD-PSOLA (Time Domain Pitch Synchronous Overlap and Add)，這個方法會把現有的錄音檔轉為歌聲。以下列出其步驟：

1. 找出錄音檔(或音色檔)在各個時間的周期
2. 把聲音按照周期切成多個區塊
3. 改變這些區塊的間距，並將區塊中互相重疊的部分相加，即可以改變聲音檔的周期。由於音高是周期的倒數，因此可以改變音高。
4. 第 3 步改變區塊的間距，會導致聲音的長度改變，為了使音長不變，可以內插區塊，或者刪除多餘的區塊。

圖 1 是 TD-PSOLA 的運作方式。每個區塊用不同的顏色標記，在改變區塊間距時，相同的顏色表示重複的區塊。

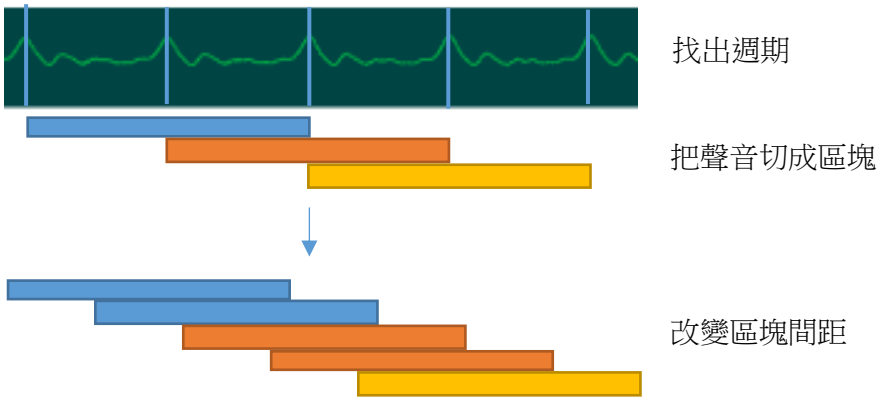


圖 1、PSOLA 的示意圖

每個區塊實際上包含兩個周期的聲波，因為如果區塊只有一個周期，則在區塊邊界上會有斷點產生，合成結果會有雜音。在取出區塊時，為了消除斷點，還要把區塊乘上漢明窗，把邊緣處柔化。

把聲音按照周期切割的原因是，可以假設人聲是由聲帶振動產生的脈衝，經由共鳴器官得到的結果，脈衝的頻率是音高，而共鳴器官的脈衝響應就是區塊。

為了做出 TD-PSOLA，必須要能夠偵測聲音的周期，然而偵測聲音的周期(或音高)是很難的問題。Praat 的論文提到，可以用自相關函數來取得周期和音高。[3]自相關函數就是自身與延遲後訊號的相似度，由公式(1)計算出。計算聲音的自相關函數後，可尋找其最大值，通常這就是周期。

$$\mathrm{ac}(\tau) = \sum_{i=0}^{N-1-i} f(i)f(i+\tau) \qquad (1)$$

由公式(1)計算的自相關函數，在週期很大的時候會失真，因為累加的項數減少了。Praat 的論文提出的方法是，把自相關函數除以窗函數的自相關函數，就可以修正。[3]

即使做了以上的修正，有時候還是會受到雜訊的影響，使得最大值的位置不是周期，這時可以計算每個局部最大值的位置作為音高的機率，然後用維特比演算法取得最佳音高序列。不是所有的聲音都有音高，比如說錄音的開頭和結尾、語氣停頓的地方、還有子音，因此使用維特比演算法還要把「無音高」狀態加進來考慮。當音量很小，或者自相關函數很小時，就有可能是子音，在這些情況下，要把「無音高」的機率提高。[3]

在合成歌聲時，我假設錄音檔裡只有子音和母音，母音有音高，可以用 TD-PSOLA 處理，但是子音沒有音高，無法以音高切割聲音。我的處理方法是直接把子音部分疊加到合成波形上，因為子音不會改變長度。萬一子音部分太長，超過音符長度，才會嘗試縮短子音，這時會把聲音切成長度 0.008~0.012 秒的區

塊，然後刪除部分的區塊，直到長度滿足要求。

四、　系統實作

本專題所開發的 App 是網頁，以 JavaScript 來撰寫，這樣就不需要發佈到 App Store，而且也省下編譯的麻煩。

程式的流程如圖 2 所示。使用此程式需要先錄下唸歌詞的聲音，然後從錄音檔裡面圈選出每個單字的範圍，最後輸入歌詞的音高和音符。

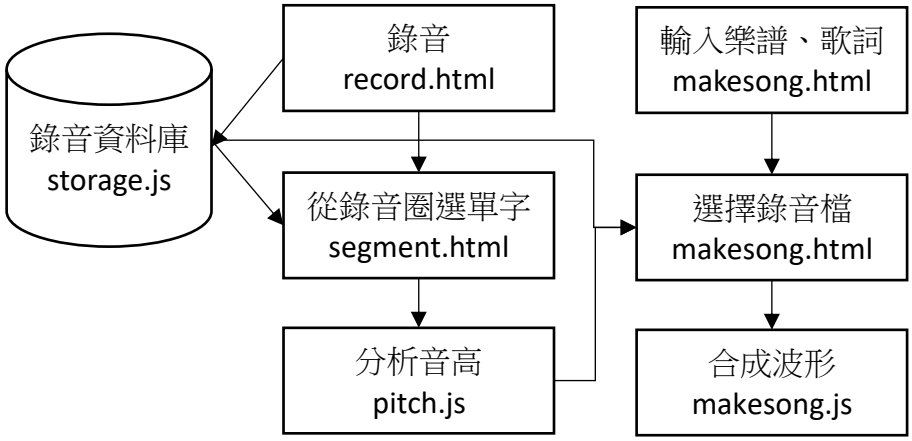


圖 2、系統架構圖

程式由主畫面和數個子程式構成。主畫面提供五個功能：

- Record in browser：錄音程式
- Change pitch by changing playback speed：變更播放速度，以改變音高
- Detect pitch：偵測音高程式的展示
- Segment each word：圈選單字的程式
- Make song：輸入音符和歌詞後，就可以合成歌聲的程式

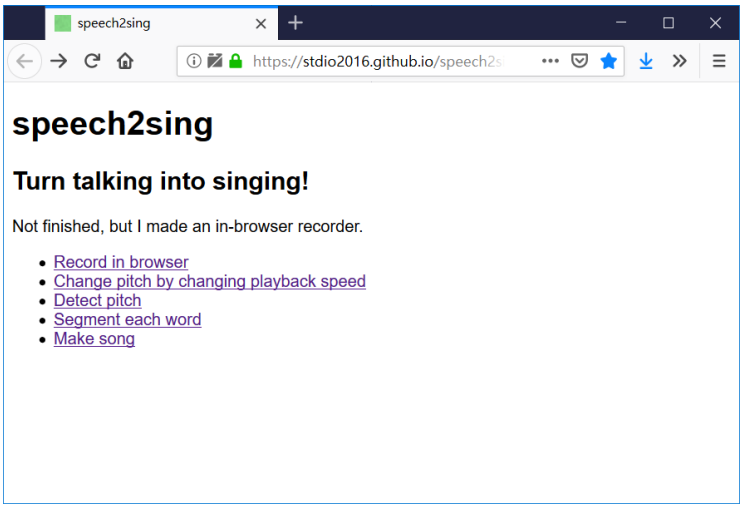


圖 3、程式主畫面

以下說明各個子程式的介面

- Record in browser：錄音程式

網頁名稱為 record.html，如圖 4，UI 的程式為 js/record.js。

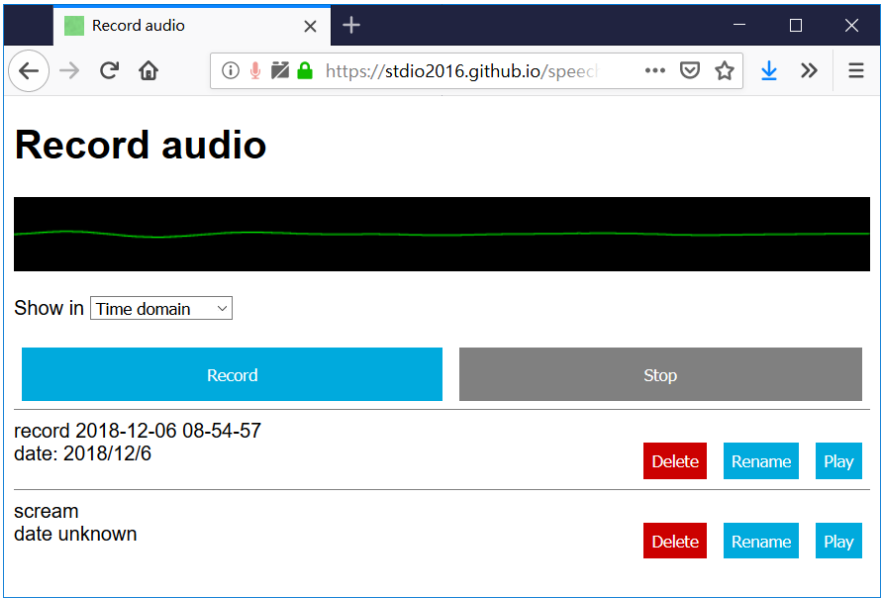


圖 4、Record in browser 的介面

啟動時會詢問「是否要分享麥克風」(訊息會因為瀏覽器而有不同)，請按下分享，這樣程式才能夠使用麥克風錄音。按下 Record 鍵可以錄音，再按 Stop 可以結束錄音，這時系統會詢問錄音檔的名稱，輸入後就可以存檔。如果不指定名稱，則預設以現在時間命名。

已儲存的錄音檔會列出來，可以播放錄音、更改檔名或刪除錄音。

錄音和播放的聲音可以視覺化顯示，提供 Time domain (時間域)、Frequency domain (頻率域) 和 Autocorrelation (自相關) 輸出，可從 Show in 下拉式選單選取。Time domain 顯示波形，Frequency domain 顯示 0~5000Hz 的頻譜圖，Autocorrelation 顯示正規化後的自相關函數。

- Change pitch by changing playback speed：變更播放速度，以改變音高

網頁名稱為 changepitch.html，如圖 5，UI 的程式內嵌於網頁裡。

這個程式可以用重新取樣的方式來改變音高，使用方法是：調整 pitch 拉桿，然後在下方的檔案列表中按下 Play 按鈕，就可以播放。pitch 拉桿在正中央時，聲音不變，拉桿越往左，聲音越低，速度也變慢，拉桿越往右，聲音越高，速度也變快。用重新取樣改變音高的特點是，音長也會一起改變，且音長和音高成反比。

這個程式沒有在流程圖上，功能也非常侷限，因為我寫這個程式只是為了測試瀏覽器的 API 功能，在實作出合成器之前，我可以先知道調整音高後聲音會變怎樣。

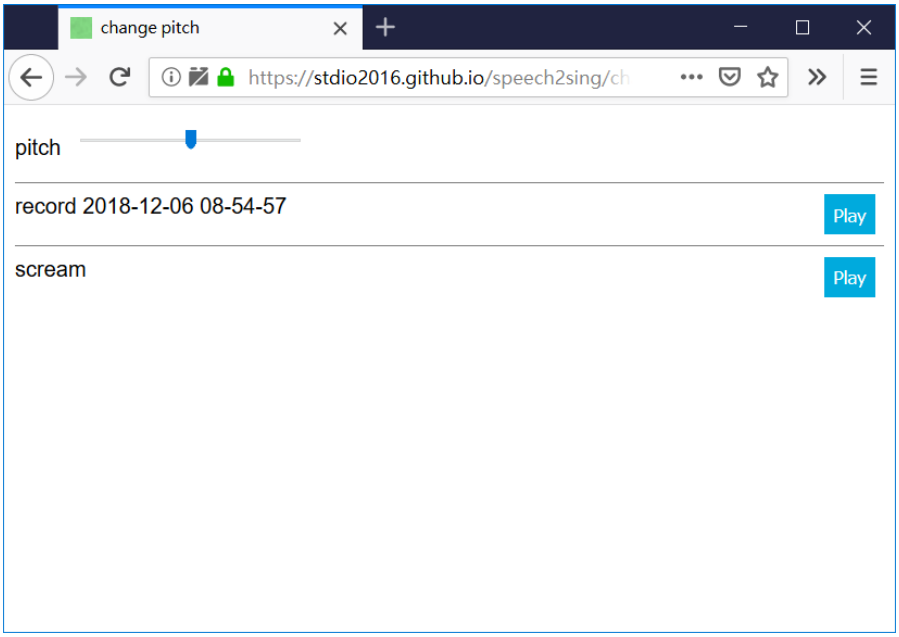


圖 5、Change pitch by changing playback speed 的介面

- Detect pitch：偵測音高程式的展示

網頁名稱為 pitch.html，如圖 6，UI 的程式為 js/pitch_ui.js。

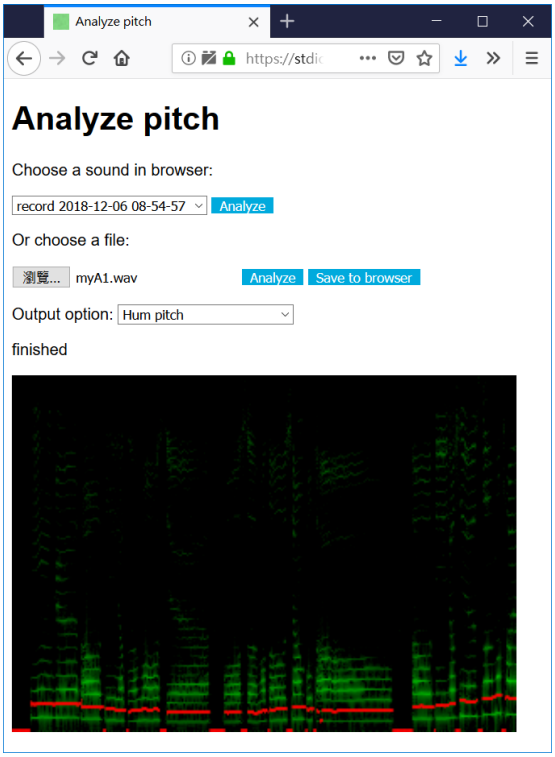


圖 6、Detect pitch 的介面

網頁中的標題是「Analyze pitch」不是「Detect pitch」，因為我不清楚哪個名稱比較貼切。

從 Choose a sound in browser 選單選擇一個錄音，再按下第一個 Analyze 鍵，即

可分析音高。分析需要一些時間，會在畫面下方顯示進度。

分析完會顯示時頻譜，用綠色表示，橫軸是時間軸，顯示大約前 10 秒的範圍，縱軸是頻率，範圍是 0 ~ 5000Hz。音高以打紅點呈現，橫軸與時頻譜一致，縱軸則是 0~2500Hz，由於偵測音高的間隔是 0.01 秒，因此紅點非常密集，看起來像是曲線圖。音高以實際值的兩倍呈現，因為通常人聲的音高在 80~300Hz 之間，會很靠近圖案的下方，不易觀察變化。

也可以分析電腦或者 Android 手機上的音檔。只要按下 Or choose a file 的瀏覽鍵，就可以選擇裝置上的檔案，再按下第二個 Analyze 鍵以分析音高。iOS 由於系統限制，無法在瀏覽器裡選擇音檔。Save to browser 鍵可以把檔案存入本程式的資料庫。

除了輸出時頻圖外，還可以播放聲音。Output option 下拉式選單可以設定輸出的效果：

1. Hum pitch：哼出音高，音色是正弦波
2. Correct pitch：把音高調整到 C 大調
3. Robotic：把音調全部變成一樣的音，因為像機器人說話一樣沒有起伏，所以我稱為 Robotic。
4. Octave up/down：升降八度，不改變音色
5. To female/male voice：把聲音轉成女生/男生的聲音
6. Harmonic effect：自動產生三度和音，只能支援 C 大調或 a 小調
7. Helium effect：模擬吸入氦氣之後的聲音
8. Sulfur hexafluoride effect：模擬吸入六氟化硫後的聲音。六氟化硫比空氣密度大，會使聲音變得低沉。
9. Mute：不輸出聲音

以上大多數效果的合成方法是 TD-PSOLA，也就是專題程式用來合成歌聲的方法，不過有些效果器不如預期，例如 To female voice 對我的聲音處理後，聽起來像我還沒變聲的聲音，不像女生的聲音。

這個程式是做為測試用，並不在系統流程圖上。不過如果要使用裝置上已有的檔案作為音色，就需要在這裡把檔案存入資料庫。

- Segment each word：圈選單字的程式

網頁名稱為 segment.html，如圖 7，UI 的程式為 js/segment.js。

在使用錄音程式儲存錄音之後，需要把錄音裡面的每個字的範圍框出來，因為合成器需要每個字的聲音。使用方法是，從 Choose a sound in browser 選單選取錄音，按下 Open 鍵以載入音檔。載入需要一些時間，等到讀取完後，黑色長條處就會顯示聲音的波形圖。

這時畫面會出現紅線和藍線，這兩條線之間的範圍代表一個字的範圍，可以拖

曳這兩條線來調整範圍。當確認框選範圍是正確的以後，就可以按下 Add segment，新增這個區間，程式會詢問這個區間的名稱，建議在這裡輸入這個區間所代表的字，就可以用歌詞找錄音了。

為了方便找出一個字的開頭和結束，程式提供播放範圍的功能。按下 Play 可以播放目前可見波形的範圍，按下 Play selected 則可以播放選取區間，也就是紅線和藍線之間的範圍。

如果聲音太長，不足以顯示完整波形，可以左右拖曳波形圖。還可以按下 Zoom in 放大波形圖，Zoom out 縮小波形圖。

已經新增的區間如果發現有問題，也可以修改，只要在下方的列表處找到需要更改的區間，然後按下 Edit 鍵，紅線和藍線就會自動更新成這個區間，並且波形圖的上面會顯示正在編輯的區間名稱。這時可以再度拖曳這兩條線，確認沒有問題後按下 Confirm 鍵。

當錄音裡所有的字都已經框選完以後，就可以按下 Save 鍵存檔。如果沒有按下 Save 就離開的話，程式不會自動存檔。

可以修改已經儲存的範圍。在載入音檔之後，如果發現已儲存的區間，會全部列出來，這時就可以用 Edit 更改區間。

這個程式會順便分析音高，並把音高資訊存進資料庫，這樣在合成歌聲的階段就不用每次合成都重新計算音高。

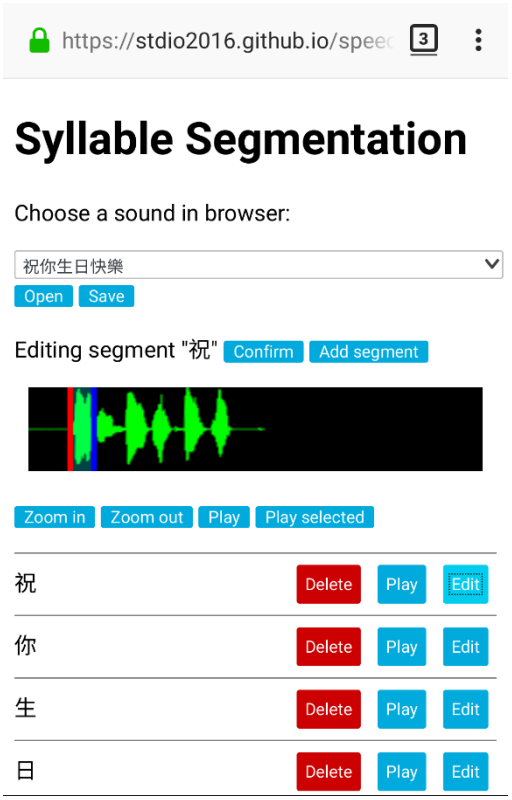


圖 7、Segment each word 的介面(手機畫面)

- Make song：輸入音符和歌詞後，就可以合成歌聲的程式

網頁名稱為 makesong.html，如圖 8，UI 的程式為 js/makesong.js。

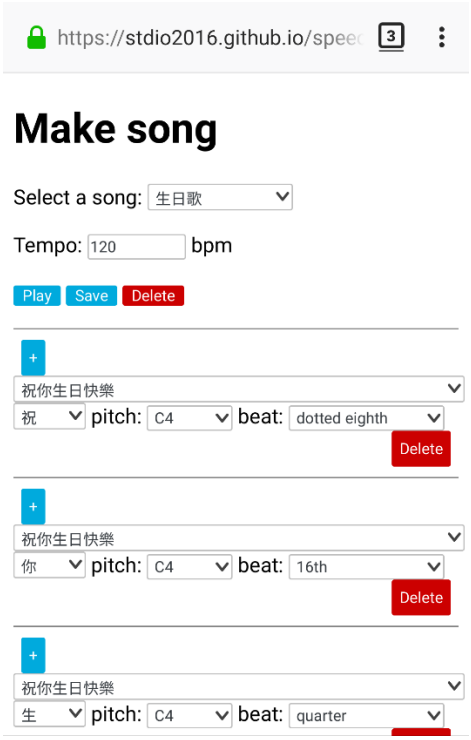


圖 8、Make song 的介面(手機畫面)

在這裡可以輸入樂譜，合成出歌聲，不過由於開發的困難度，我設計的介面不是音樂製作軟體常見的 Piano roll 形式，而是使用下拉式選單，選擇音色檔、音高以及音長。

首先在 Tempo 欄位輸入一首歌的節拍速率，本程式以四分音符為一拍，輸入的單位是每分鐘的拍子數，也就是 bpm(beats per minute)。

接下來，按下頁面最下方的 Add note 按鈕新增一個音符(沒有在圖 8，因為畫面太長被截掉了)，就會出現如圖 9 的介面。

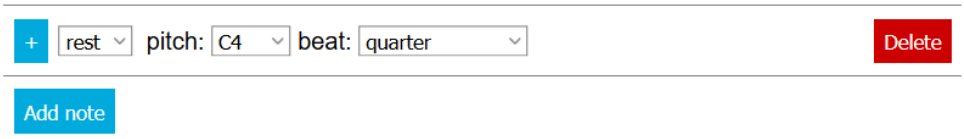


圖 9、音符編輯介面

最左邊的加號是在這個音符的上面插入音符，最右邊的 Delete 是刪除音符。

Pitch 選單是音高選取，使用的音高記號是科學音高記號法：英文字母 CDEFGAB 加上升記號，再加上八度的編號。八度編號是 4 表示為中央八度，如 C4 是中央 Do，A4 是中央 La。

Beat 選單是音長選取，可選擇音符的種類。目前提供 16th(十六分音符)、

eighth(八分音符)、dotted eighth(附點八分音符)、quarter(四分音符)、dotted quarter(附點四分音符)、half(二分音符)、dotted half(附點二分音符)和 whole(全音符)。

加號右邊的選單是錄音檔選擇，當選取錄音檔之後，這個選單的右邊會再出現一個選單，如圖 10 (檔案存在本地端資料庫而不是雲端，所以檔名不會和這裡的一樣，只是做參考)。出現的選單可以選擇此錄音檔裡面的單字，這個選單就可以輸入歌詞。

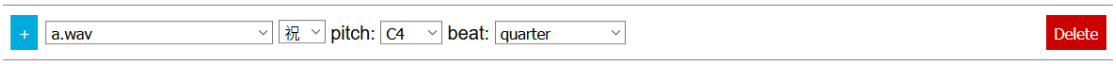


圖 10、音符編輯介面，選擇錄音檔

錄音檔選單還有特別的選項，rest 和 tied。Rest 是休止符，因為沒有聲音，所以不能選擇歌詞。Tied 表示這個音符要跟上一個音符連在一起，歌詞沿用上一個音符，如果音高相同，則相當於樂譜中的連結線，如果音高不同，則相當於圓滑線。程式會對圓滑線作圓滑處理，音高以線性內插計算出來，所以音高不會突然的跳動。

輸入完音符後，按下畫面上方的 Play 可以合成並播放歌聲，如果滿意，按下 Save 就可以存檔。存檔時會詢問歌名，預設是作曲日期。

如果不想要某一首歌，可以在畫面最上面的 song 選單選擇歌曲，然後按下 Delete，為了避免誤刪，系統會再度確認你是否要刪除。

專題還寫了一些工具程式，以下介紹

- js/pitch.js 和 js/pitchworker.js

分析音高的程式。在原理部分有提到偵測音高的方法，但是我發現，偵測音高需要的計算量非常大，以至於成為效能的瓶頸，所以我把偵測音高的部分用 Web Worker 來平行化。Web Worker 是瀏覽器提供的 API，可用來背景執行程式，開啟多個 Web Worker 就可以平行化。我平行化的部分是計算自相關，因為每一秒的聲音檔要計算 100 次的自相關函數。

- js/storage.js

處理儲存的功能。我使用瀏覽器的 Indexed DB 來儲存資料，如錄音檔、歌曲等。Indexed DB 是一種資料庫的 API，可在本地端存放資料，並且利用非同步操作來改善效能。然而非同步操作也造成程式相當難以撰寫，因此我把處理資料庫的部分獨立出來。

- js/forSafari.js

用來解決在 Safari 瀏覽器裡無法播放聲音的問題。

- lib/fft.js

我撰寫的快速傅立葉轉換程式，有對實數資料做優化。在計算時頻圖以及自相關函數時會用到。

- css/style.css

用來設定介面的外觀。

- lib/alertbox.js 和 css/alertbox.css

用來顯示錯誤訊息，以及顯示對話框。

- lib/IndexedDB-getAll-shim.js

這不是我寫的程式，我用它來解決在 Edge 和 Safari 瀏覽器裡 Indexed DB 的相容性問題。

- lib/audio-recorder-polyfill 內的檔案

這是其他人寫的程式，但是我發現和我的程式的架構不相容，所以有做修改。用來解決在 Edge 和 Safari 瀏覽器裡錄音功能的相容性問題。

五、 成果

程式已放到網路上，網址為 https://stdio2016.github.io/speech2sing/index.html

原始碼放在 https://github.com/stdio2016/speech2sing/裡。

本程式可以支援電腦和 Android 手機的 Firefox、Chrome 和 Edge。

以下是用專題的程式產生的結果，測試音檔及輸出結果都在 sound 資料夾裡。為了展示錄音功能，所有的測試音檔都是在手機上，由這個專題程式錄製我的聲音。由於程式是以即時方式輸出，因此不會產生音檔，輸出音檔是我錄製電腦發出的聲音。

測試音檔 1：人生短短.ogg

使用 Detect pitch 程式，Output option 設為 To female voice，輸出的結果為「人生短短_female.wav」。

圖 11、人生短短.ogg 的時頻圖

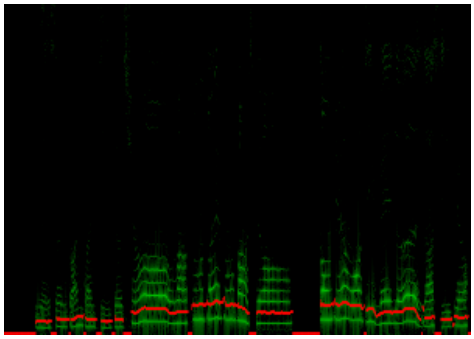
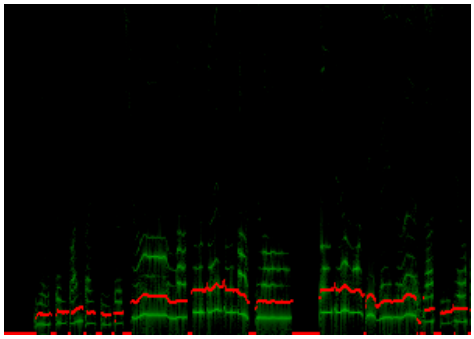


圖 12、人生短短_female.wav 的時頻圖



測試音檔 2：祝你生日快樂.ogg

用「祝你生日快樂.ogg」來製作生日歌

表 1、每個字在錄音檔「祝你生日快樂.ogg」的位置

歌詞	開始	結束
祝	0.6386667s	1s
你	0.9786667s	1.3213333s
生	1.3213333s	1.832s
日	1.8746667s	2.1853333s
快	2.3453333s	2.7173333s
樂	2.8446667s	3.08s

合成結果為「生日歌.wav」，BPM 設為 120。

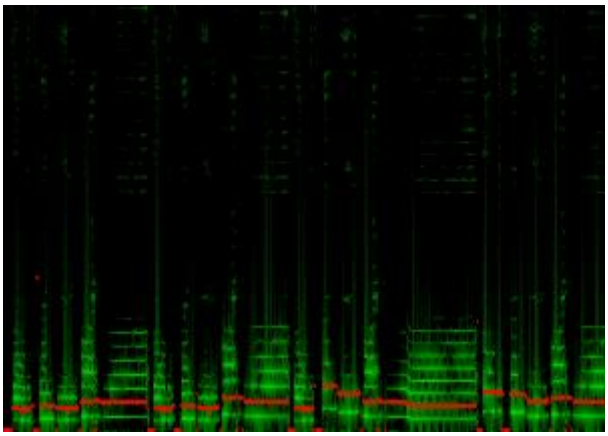


圖 13、生日歌.wav 的時頻圖

測試音檔 3：能不能給我一首歌的時間.ogg

合成出周杰倫的「給我一首歌的時間」的某一段副歌。

表 2、每個字在錄音檔「能不能給我一首歌的時間.ogg」的位置

歌詞	開始	結束
能	0.5266667s	0.8293333s
不	0.875s	1.092s
給	1.4653333s	1.704s
我	1.704s	2.0093333s
一	2.0933333s	2.2546667s

首	2.2813333s	2.6853333s
歌	2.7546667s	2.9426667s
的	2.964s	3.16s
時	3.2893333s	3.6026667s
間	3.6773333s	3.9826667s

表 3、「給我一首歌的時間」的某一段副歌的譜

錄音檔	歌詞	音高	音長
能不能給我一首歌的時間	能	G#3	eighth
能不能給我一首歌的時間	不	F4	eighth
能不能給我一首歌的時間	能	F4	eighth
能不能給我一首歌的時間	給	F4	eighth
能不能給我一首歌的時間	我	D#4	eighth
能不能給我一首歌的時間	一	D#4	eighth
tied		C#4	eighth
能不能給我一首歌的時間	首	C#4	quarter
能不能給我一首歌的時間	歌	C#4	eighth
能不能給我一首歌的時間	的	D#4	eighth
能不能給我一首歌的時間	時	F4	quarter
能不能給我一首歌的時間	間	D#4	dotted quarter

BPM 設為 120，合成結果為「給我一首歌的時間.wav」。

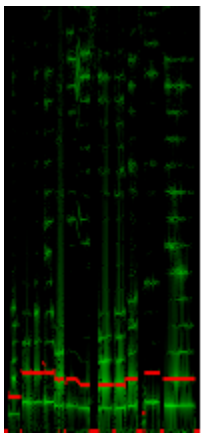


圖 14、給我一首歌的時間.wav 的時頻圖

由於現在瀏覽器基於安全性的限制，如果要在本地端執行這個程式，需要使用本地端的網頁伺服器，然後連上 localhost。

六、 結論與心得

我在這個專題寫出的程式，合成聲音的效果不怎麼好，如測試音檔 2 的長音，音量變化很奇怪。這是因為程式沒有音量處理，音量只由錄音決定，而講話持續在最大音量的比例較少，唱歌持續在最大音量的比例較大。另外，合成的聲音，偶而會在單字的開頭有雜音。我認為是因為程式誤判子音和母音的位置，導致程式嘗試延長子音。

測試音檔 3 的「我」這個字破掉了，因為錄音程式在我唸「我」這個字的時候當了一下，結果錄音就不連續了。

就算有這些問題，我還是認為已經達到我娛樂的目的，至少這個 App 可以合成出同學的歌聲。希望未來能透過類神經網路，來改進音質，不過到那時候，應該也不是開發網頁程式了吧。

我花了很久的時間在尋找資訊專題的題目，還在指導教授的面前拿捏不定，後來才把大一大二時玩音樂遇到的問題當成專題。然而一開始並不順利，因為我的功課太重了，每天被作業追著跑，沒有時間研究專題。一直到這個學期，我修了互動式音訊處理導論之後，才有辦法繼續追趕進度。這門課教的東西，有些是我已經研究過的東西，但更多的是我沒碰過的主題，例如隱藏式馬可夫模型，還有拍速預測等，這些使我能夠理解相關研究論文。

即使是這學期，也感覺有很多狀況，使我沒辦法做更多改進。因為推甄研究所，花了我許久的時間來做備審資料，導致專題競賽初賽時沒有做出我認為可以展示的成果，還得在初賽結束後的晚上補上合成器。我覺得，能夠做出這種程式，已經是很神奇的事了，因為我發現，系上沒有一個教授是做聲音處理相關的，指導教授能給我的幫助有限，專題也只有我一個人，報告和海報需要我來做。這個專題只有我一個人，其實是因為我太晚才決定題目，其他人已經找好組員了。

至於把資訊工程專題當成互動式音訊處理導論的專題……本來我想要做的是，改進課堂上提到的哼唱式歌曲查詢，但是我發現到已經接近期末了，專題卻還沒有完成，在時間壓力下，只好把資訊專題當成互動式音訊的專題。

七、 參考文獻

[1] Paul Boersma & David Weenink (2017): Praat: doing phonetics by computer [Computer program]. Version 6.0.23, retrieved 11 November 2017 from http://www.praat.org/

[2] 吳銘冠、陳嘉平（2013）。基於時域上基週同步疊加法之歌聲合成系統。載於 Proceedings of the 25th Conference on Computational Linguistics and Speech Processing (ROCLING 2013) (頁 76-89)。中華民國計算語言學學會

[3] Paul Boarsma (1993): Accurate short-term analysis of the fundamental frequency and the harmonics-to-noise ratio of a sampled sound. IFA Proceedings 17: 97-110.