

## 國立臺灣海洋大學 107 年度專題執行成果報告

計畫名稱：建構 AI 生成音樂之個人化使用者介面系統

學生姓名：游俊弘

就讀系所：資訊工程學系

學生學號：00557103

指導教授：莊鈞翔

研究期間：108 年 7 月 1 日至 108 年 9 月 20 日底止

中華民國 108 年 9 月 20 日

## 目錄

Chapter 1 介紹 .....	3
1-1、摘要 .....	3
1-2、前言 .....	3
1-3、研究目的.....	3
Chapter 2 研究環境與方法 .....	4
2-1、使用對象.....	4
2-2、操作環境(後端/前端).....	5
2-3、方法 .....	5
2-3-1、架構圖.....	5
2-3-2、前端技術.....	6
2-3-3、後端技術.....	10
Chapter 3 實驗結果 .....	12
3-1、訓練資料.....	12
3-2、網頁介面.....	14
3-3、實際操作.....	19
3-3-1 初始設定.....	19
3-3-2 實際使用.....	23
3-4、程式碼架構.....	23
Chapter 4 討論.....	24
4-1 成效討論 .....	24
4-2 優缺點 .....	24
4-3 未來展望 .....	25
Chapter 5 結論.....	26
5-1 結論與建議.....	26
5-2 成果自評 .....	26
5-3 指導教授評語.....	27
Chapter 6 參考文獻.....	28

## Chapter 1 介紹

### 1-1、摘要

音樂在我們的生活無所不在，每個人都希望有一首屬於自己的歌曲，面臨 AI 時代，在看到 Google Brain 團隊所研發的 Magenta 套件，除了用來自動生成旋律、和弦與節奏，運用樂理方式組合成一首完整的歌曲，以此作為出發點，本研究則是運用監督式學習卷積神經網路(Convolutional Neural Network, CNN)為基底所開發出來的 MuseGan，了解其中的演算法，透過改變其參數，已達成生成出良好的訓練模型，以完成一首歌為目的，並且製作方便使用者編輯與使用的網頁。

### 1-2、前言

在資訊科技迅速發展的時代中，許多我們的生活體驗都允許加入智慧化系統，如此能夠善用科技進步所帶來的利益，也減少許多無謂時間、金錢成本的浪費。運用樂理方式組合成一首完整的歌曲。再來此研究自動化且客觀的用來編輯該新創音樂，生成更貼近使用者感受的音樂，以達到個人化的概念。其中，也搭配良好的使用者介面(User Interface, UI)，讓使用者能夠從生成出來的音樂，用大幅縮短的時間來完成一首屬於自己的歌。

然而，AI 音樂生成技術有很多種，例如：監督式學習遞迴神經網

路(Recurrent Neural Network, RNN)，半監督式強化學習(Reinforcement Learning, RL)，Q-learning、Target Q-network、Reward RNN 演算法來實作、運用長短期記憶(Long Short Term Memory Network, LSTM)，但這些在學習時間依賴性為佳，但生成音樂注重的是每小節之間的和諧，CNN 可以很好學習這種局部模。

### 1-3、研究目的

由於目前的音樂自動生成系統，大多偏向從無到有直接生成音樂，並直接控制模型參數，而鮮少針對個人喜好與個人化生成音樂，本研究欲開發 AI 音樂自動生成系統，包括訓練模型、AI 生成模型；使用者可以透過 UI 介面，其功能包括輸入 midi 檔案、調整參數、播放以及存檔的功能，輸入已經訓練好的 midi 檔案，MIDI 檔本身即包含多音軌（以及樂器資訊），能更方便，加以編輯完成一首歌曲，自動化且客觀的用來表達新創音樂，並且返回模組再次重新生成更貼近使用者感受的音樂，以達到個人化的概念。

## Chapter 2 研究環境與方法

### 2-1、使用對象

1. 對於音樂製作
2. 有相關興趣者

## 2-2、操作環境

後端:linux 18.04 Server

前端:使用 javascript 與 jquery，架設在 github 上，使用

Google 網頁 或 Safari 網頁

## 2-3、方法

### 2-3-1、架構圖

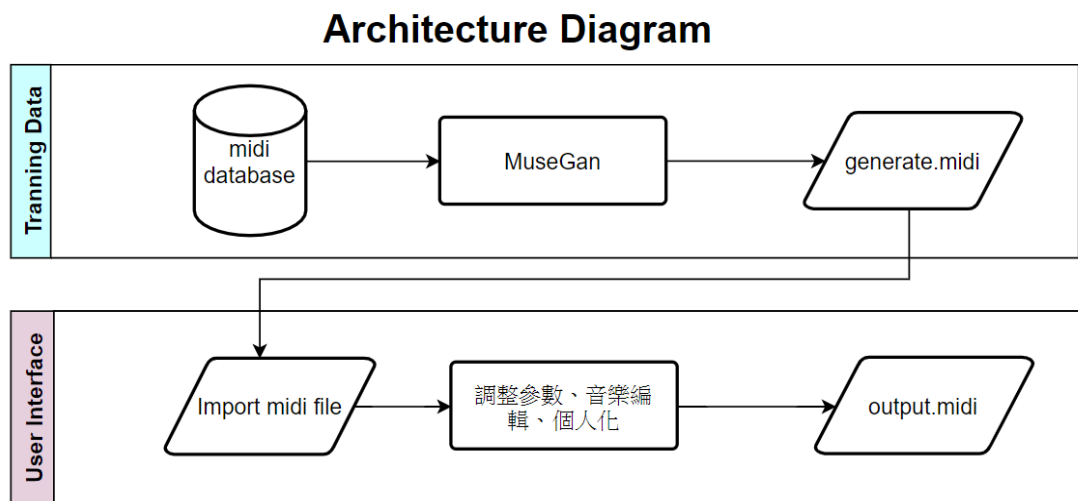


圖 1 架構圖

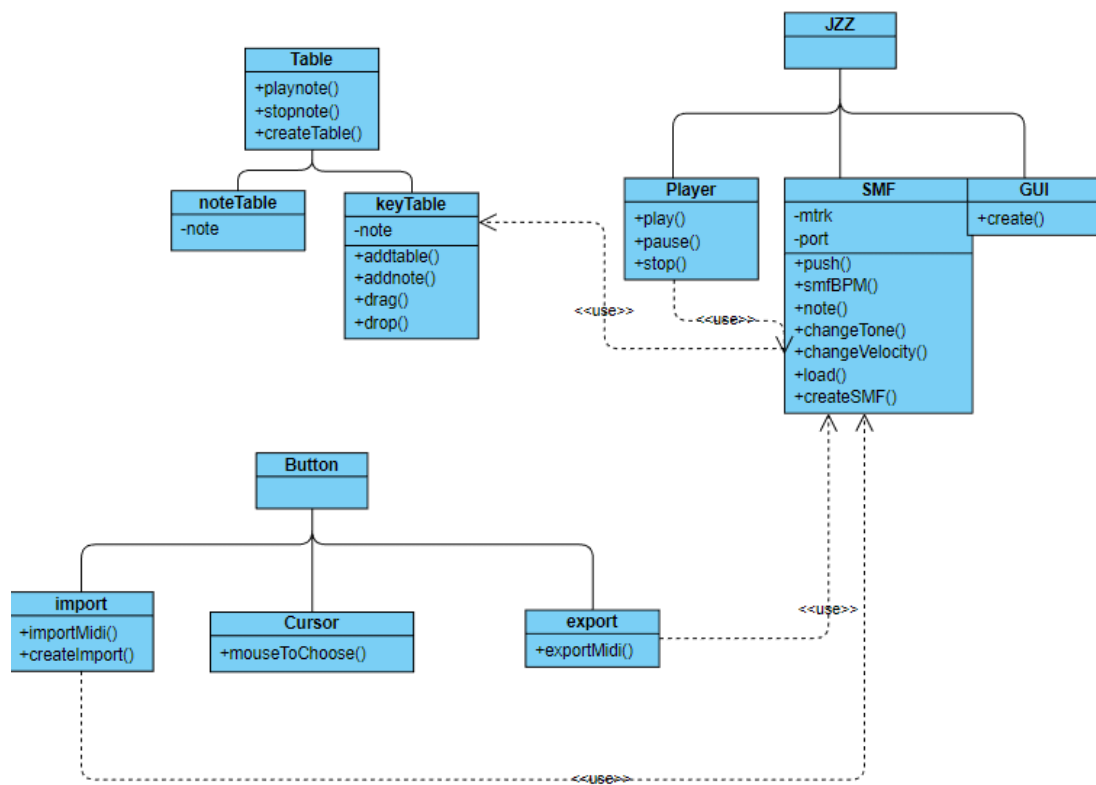


圖 2 UML

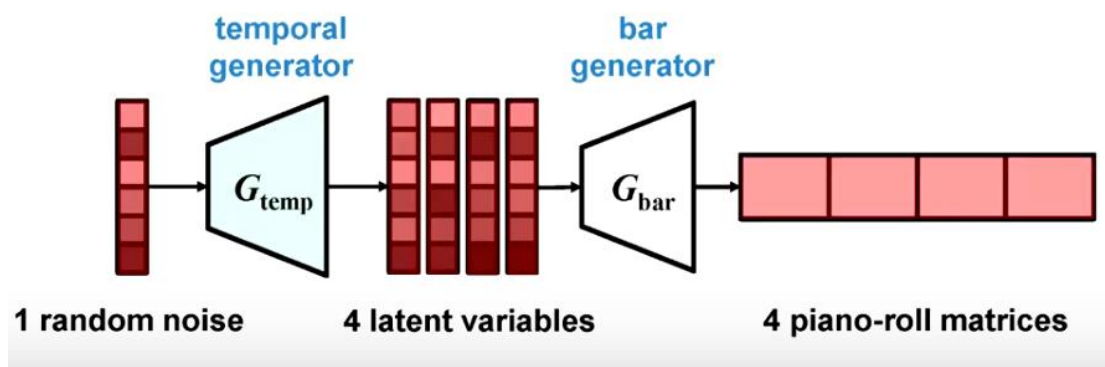


圖 3 MuseGAN 架構

## 2-3-2、前端技術

在講解技術前，我們先對於音樂的幾個名詞解釋一下：

### 1. Tick:

在 MIDI 編曲(Sequence)當中，這是比一拍更小的單位。假設一拍若能分割成 120 個 Tick，則 60 ticks 就代表半拍，30 ticks 代表 1/4 拍(16 分音符)。

一般 MIDI 編曲器中，您可以自由指定每一拍可分割成幾個 Tick，這也就是所謂的解析度(Resolution)。分割得越細，MIDI 音樂就能表現出越大的細膩度。

單位是以 TPQN 來表示，也就是"Ticks Per Quarter Note"(每一四分音符幾個 Tick)的意思。舉例而言，若一台 MIDI 編曲機的規格寫出"Resolution: 480 TPQN"，就表示它可以將一拍分割成 480 個 Tick。在有的 MIDI 編曲器材(Sequencer)中，會以"Clock"來表示"Tick"。

## 2. Track:

一個音軌，就是一道所儲存的訊號。這個所儲存的訊號，可以是聲音訊號、MIDI 訊號、或者是影像(video)訊號。

在聲音(Audio)的領域中，聲道(Channel)與音軌(Track)在英文中是截然不同的觀念，但台灣地區的使用者一律都習慣稱呼為音軌，導致在做更深入的應

用時發生觀念上的混淆。簡單來說，聲道(Channel)是訊號傳輸的通道，而音軌(Track)則是一道所儲存的訊號。因此在混音座上，您不會看到"Track"的字眼，因為"Track"的概念只出現在具有錄音功能的器材。

在 MIDI 的領域中，頻道(Channel)與軌(Track)也是不同的東西。一個 MIDI 軌指的就是一道所儲存的 MIDI 訊號，而頻道指的是 MIDI 訊號所用來傳送通道。

在編寫網頁時，我們使用 javascript 及 jquery 來呈現我們的網站，將其架設在 github 上，這樣在開發者身上，除了 git 是免費且開源，在專案版本控制上也有良好的系統管理。

我們使用 JAZZ-SOFT 音樂套件，做出此網頁的音樂能播放的一個基礎，其內容主要分為兩大 Documentation，一是 JZZ.js，它是 midi library for Node.js 以及 web browsers，另一個則是 Jazz-Plugin，它是屬於一個在網頁瀏覽器比較低層次的 midi support。還有使用 SweetAlert 彈出視窗模組達到匯入匯出時的動畫彈出效果，我們也可以把它簡稱為 swal，SweetAlert 有很多參數可以設定，比起放入 3 個字串當作參數，我們可以直接加入一個物件當作參數，



讓設定更具有彈性。

在編輯頁面中，主要就是音符的表格，將表格物件化，降低了原本表格的複雜度，再來就是滑鼠的觸發事件(ClickEvent)，以下是我們設計的 pseudocode 如下圖 4:

```
DO:
  增加ClickEvent

case 編輯音符:
  (當左鍵按下時)
    播放對應音符並儲存至tempnote
    若該格為空增加音符物件
    當該格為當前最後一小節則增加小節數
    當鼠標為左右拖曳時可改動音符長度
  (當拖曳時)
    更改音符長度
  (當放開時)
    停止播放對應音符
    修改音符長度 24tick為單位
    修改格子長度 46px為單位
  (當右鍵按下時)
    清除該格之物件

case 選擇範圍:
  (當左鍵按下時)
    決定起始位置
  (當右鍵按下時)
    決定結束位置
```

圖 4 ClickEvent pseudocode

在匯入 midi 檔時，我們需要注意，要清除當前播放音檔，新建音符表格，將新的檔案製作音源，其 pseudocode 如下圖 5:

```

function createImport(midi檔案)
    清除目前音符表格
    建立新音符表格
    new JZZ.MIDI.SMF(midi檔案)//建立新的SMF(音源)，放midi檔案進入
    紀錄每個track音符數
    for
        將音符存入
    end
end
end

```

圖 5 createImport pseudocode

### 2-3-3、後端技術

我們安裝的 server 是 ubuntu 18.04 版本，安裝內容：

1. CUDA version 10.0
2. Nvidia-driver version 418.07.00
3. cuDNN7\_7.6.3.30
4. tensorflow gpu 版

在生成音樂這方面，我們使用的是 museGAN，所使用的的 traning data 是流行音樂，所以勢必會有很多的音軌 (multitracks)，舉例：一首歌可能有 3 個 tracks，可能就會有鼓、鋼琴、吉他 3 個不同的 tracks，其 Generator 必須確保在生成時仍必須保持它原有每個 track 不同的特性，其 Discriminator 則是一個 CNN 的架構。

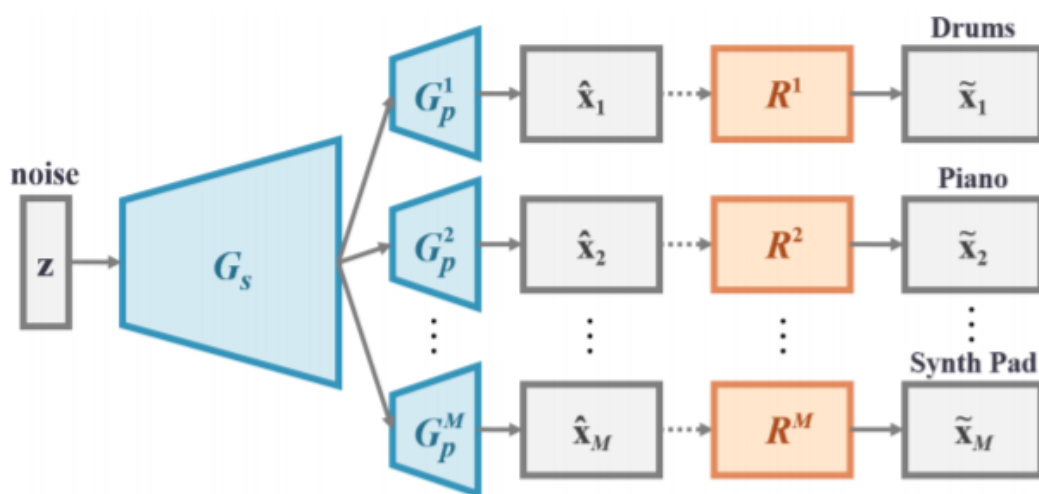
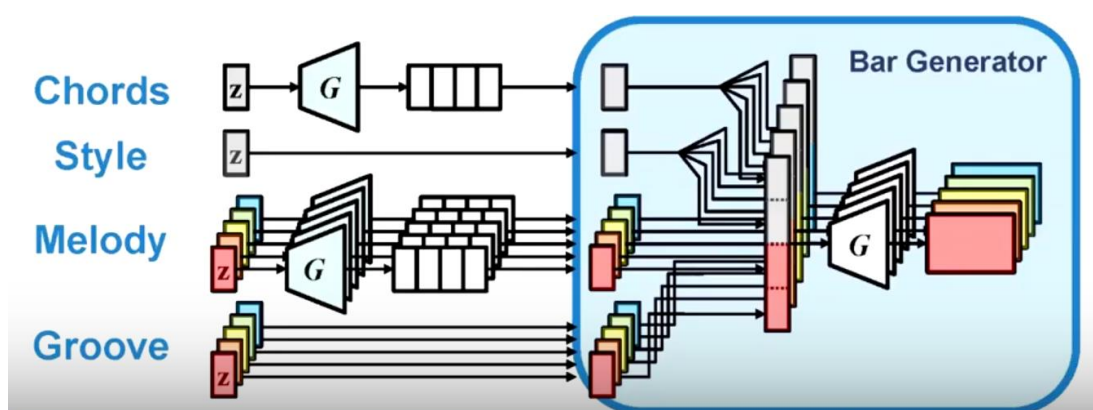


圖 6 museGAN Generator 架構



Generator		Time	
		Dependent	Independent
Track	Dependent	Melody	Groove
	Independent	Chords	Style

圖 7 以音樂上的 4 種特性，對應其與時間跟音軌的關係

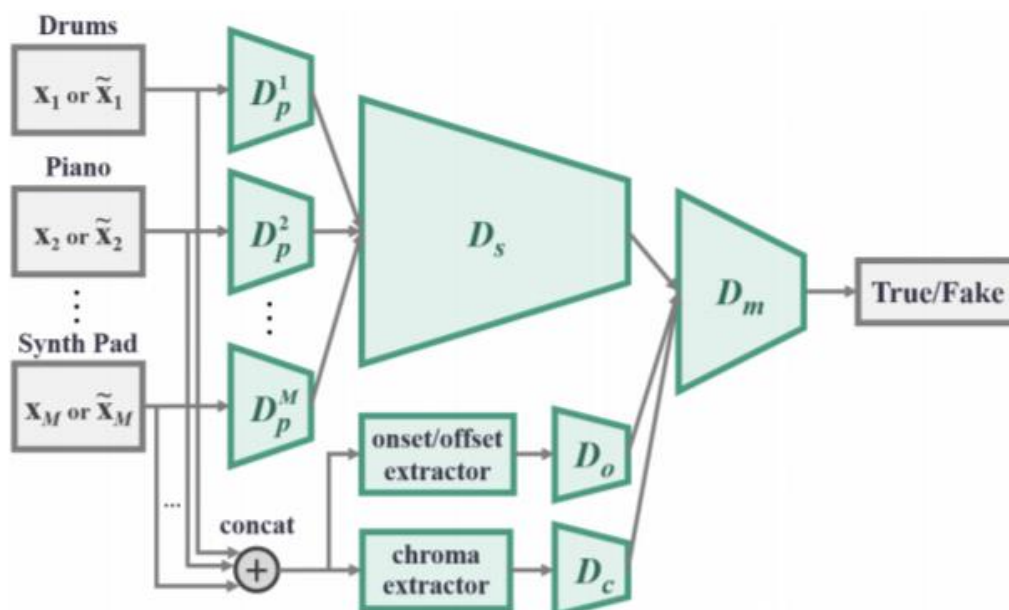


圖 8 Discriminator 架構

## Chapter 3 實驗結果

### 3-1、訓練資料

1. 圖 9~圖 11 為剛開始訓練，啟動環境的部分。

```

musegan.train INFO Using parameters:
{'beat_resolution': 12,
 'condition_track_idx': None,
 'data_shape': [4, 48, 84, 5],
 'is_accompaniment': False,
 'is_conditional': False,
 'latent_dim': 128,
 'nets': {'discriminator': 'default', 'generator': 'default'},
 'use_binary_neurons': False}
musegan.train INFO Using configurations:
{'adam': {'beta1': 0.5, 'beta2': 0.9},
 'batch_size': 64,
 'colormap': [[1.0, 0.0, 0.0],
               [1.0, 0.5, 0.0],
               [0.0, 1.0, 0.0],
               [0.0, 0.0, 1.0],
               [0.0, 0.5, 1.0]],
 'config': './exp/experiment_001/config.yaml',
 'data_filename': 'train_x_lpd_5_phr',
 'data_source': 'sa',
 'eval_dir': '/home/chang/musegan/exp/experiment_001/eval',
 'evaluate_steps': 100,
 'exp_dir': '/home/chang/musegan/exp/experiment_001',
 'gan_loss_type': 'wasserstein',
 'gpu': '0',
 'initial_learning_rate': 0.001,
 'learning_rate_schedule': {'end': 50000, 'end_value': 0.0, 'start': 45000},

```

圖 9

```
chang@cnelab: ~/musegan x kevin@kevin-pc: /mnt/c/Users/user x + v
tensorflow INFO Create CheckpointSaverHook.
tensorflow INFO Graph was finalized.
2019-09-11 09:49:39.614554: I tensorflow/core/platform/cpu_feature_guard.cc:141] Your CPU supports instructions that this TensorFlow binary was not compiled to use: AVX2 AVX512F FMA
2019-09-11 09:49:40.597919: I tensorflow/compiler/xla/service/service.cc:150] XLA service 0x219fc50 executing computations on platform CUDA. Devices:
2019-09-11 09:49:40.597979: I tensorflow/compiler/xla/service/service.cc:158] StreamExecutor device (0): TITAN V, Compute Capability 7.0
2019-09-11 09:49:40.621101: I tensorflow/core/platform/profile_utils/cpu_utils.cc:94] CPU Frequency: 3600000000 Hz
2019-09-11 09:49:40.622139: I tensorflow/compiler/xla/service/service.cc:150] XLA service 0x21c7600 executing computations on platform Host. Devices:
2019-09-11 09:49:40.622181: I tensorflow/compiler/xla/service/service.cc:158] StreamExecutor device (0): <undefined>, <undefined>
2019-09-11 09:49:40.622623: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1433] Found device 0 with properties: name: TITAN V major: 7 minor: 0 memoryClockRate(GHz): 1.455 pciBusID: 0000:65:00.0 totalMemory: 11.75GiB freeMemory: 11.34GiB
2019-09-11 09:49:40.622655: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1512] Adding visible gpu devices: 0
2019-09-11 09:49:40.630792: I tensorflow/core/common_runtime/gpu/gpu_device.cc:984] Device interconnect StreamExecutor with strength 1 edge matrix:
2019-09-11 09:49:40.630838: I tensorflow/core/common_runtime/gpu/gpu_device.cc:990] 0
2019-09-11 09:49:40.630852: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1003] 0: N
2019-09-11 09:49:40.631236: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1115] Created TensorFlow device (/job:localhost/replica:0/task:0/device:GPU:0 with 11033 MB memory) -> physical GPU (device: 0, name: TITAN V, pci bus id: 0000:65:00.0, compute capability: 7.0)
tensorflow INFO Running local_init_op.
tensorflow INFO Done running local_init_op.
```

圖 10

```
chang@cnelab: ~/musegan x kevin@kevin-pc: /mnt/c/Users/user x + v
tensorflow INFO Create CheckpointSaverHook.
tensorflow INFO Graph was finalized.
2019-09-11 09:49:39.614554: I tensorflow/core/platform/cpu_feature_guard.cc:141] Your CPU supports instructions that this TensorFlow binary was not compiled to use: AVX2 AVX512F FMA
2019-09-11 09:49:40.597919: I tensorflow/compiler/xla/service/service.cc:150] XLA service 0x219fc50 executing computations on platform CUDA. Devices:
2019-09-11 09:49:40.597979: I tensorflow/compiler/xla/service/service.cc:158] StreamExecutor device (0): TITAN V, Compute Capability 7.0
2019-09-11 09:49:40.621101: I tensorflow/core/platform/profile_utils/cpu_utils.cc:94] CPU Frequency: 3600000000 Hz
2019-09-11 09:49:40.622139: I tensorflow/compiler/xla/service/service.cc:150] XLA service 0x21c7600 executing computations on platform Host. Devices:
2019-09-11 09:49:40.622181: I tensorflow/compiler/xla/service/service.cc:158] StreamExecutor device (0): <undefined>, <undefined>
2019-09-11 09:49:40.622623: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1433] Found device 0 with properties: name: TITAN V major: 7 minor: 0 memoryClockRate(GHz): 1.455 pciBusID: 0000:65:00.0 totalMemory: 11.75GiB freeMemory: 11.34GiB
2019-09-11 09:49:40.622655: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1512] Adding visible gpu devices: 0
2019-09-11 09:49:40.630792: I tensorflow/core/common_runtime/gpu/gpu_device.cc:984] Device interconnect StreamExecutor with strength 1 edge matrix:
2019-09-11 09:49:40.630838: I tensorflow/core/common_runtime/gpu/gpu_device.cc:990] 0
2019-09-11 09:49:40.630852: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1003] 0: N
2019-09-11 09:49:40.631236: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1115] Created TensorFlow device (/job:localhost/replica:0/task:0/device:GPU:0 with 11033 MB memory) -> physical GPU (device: 0, name: TITAN V, pci bus id: 0000:65:00.0, compute capability: 7.0)
tensorflow INFO Running local_init_op.
tensorflow INFO Done running local_init_op.
```

圖 11

2. 圖 12~圖 13 為訓練(training)的部分

```
chang@cnelab: ~ x + v
precated and will be removed in a future version.
Instructions for updating:
Use standard file APIs to check for files with this prefix.
tensorflow INFO Restoring parameters from /home/chang/musegan/exp/experiment_001/model/model.ckpt-0
tensorflow WARNING From /home/chang/.local/share/virtualenvs/musegan-MRDqdHRU/lib/python3.6/site-packages/tensorflow/python/training/saver.py:1070: get_checkpoint_mtimes (from tensorflow.python.training.checkpoint_management) is deprecated and will be removed in a future version.
Instructions for updating:
Use standard file utilities to get mtimes.
tensorflow INFO Running local_init_op.
tensorflow INFO Done running local_init_op.
tensorflow INFO Saving checkpoints for 0 into /home/chang/musegan/exp/experiment_001/model/model.ckpt.
2019-09-11 09:53:36.662525: I tensorflow/stream_executor/dso_loader.cc:152] successfully opened CUDA library libcublas.so.10.0 locally
musegan.train INFO step=100, gen_loss=-9.1822E+01, dis_loss=-2.1054E+02
musegan.train INFO Running sampler
musegan.train INFO Running evaluation
musegan.train INFO step=200, gen_loss=-4.5389E+01, dis_loss=-1.7783E+02
musegan.train INFO Running sampler
musegan.train INFO Running evaluation
musegan.train INFO step=300, gen_loss= 1.7158E+01, dis_loss=-1.4479E+02
musegan.train INFO Running sampler
musegan.train INFO Running evaluation
musegan.train INFO step=400, gen_loss=-3.8843E+00, dis_loss=-1.1460E+02
musegan.train INFO Running sampler
musegan.train INFO Running evaluation
```

圖 12

訓練完成後的資料如下圖：

exp002	2019/4/10 下午 0...	檔案資料夾	
output	2019/4/5 上午 12...	檔案資料夾	
output 2	2019/4/9 下午 08...	檔案資料夾	
test_002_hard	2019/4/4 下午 11...	檔案資料夾	
test01_hard_mid	2019/3/30 下午 1...	檔案資料夾	
test01_mid	2019/3/30 下午 1...	檔案資料夾	
.DS_Store	2019/9/11 下午 0...	DS_STORE 檔案	9 KB
step.log	2019/4/1 上午 10...	文字文件	18 KB
test_002_hard.tar	2019/4/2 下午 04...	WinRAR 壓縮檔	290 KB
test01_default_mid.tar	2019/3/30 下午 1...	WinRAR 壓縮檔	490 KB
test01_hard_mid.tar	2019/3/30 下午 1...	WinRAR 壓縮檔	330 KB

圖 13 museGAN 生成好的 midi

### 3-2、網頁介面

網頁編輯部分可以進行許多功能，除了最基本音樂播放暫

停與停止，還有許多功能：（如圖 14~圖 20）。

#### 1. 播放速度:可以調整速度 bpm

2. 音色:有不同的音色可供調整
3. 功能選取:分為編輯音符、段落選取
  - a. 編輯音符:左鍵新增音符，右鍵取消音符
  - b. 段落選取:左鍵起始播放，右鍵為結束點
4. 跟隨模式:隨著音樂進度條的移動，分成手動跟隨、自動跟隨與分頁跟隨
5. 音量控制:控制音量大小
6. 匯入範例音檔及匯入本地端音檔
7. 上傳到資料庫，可以新增到範例音檔
8. 匯出至使用者指定資料夾



圖 14 網頁首頁，下放有三個按鈕由左至右分別為首頁、幫助、編輯



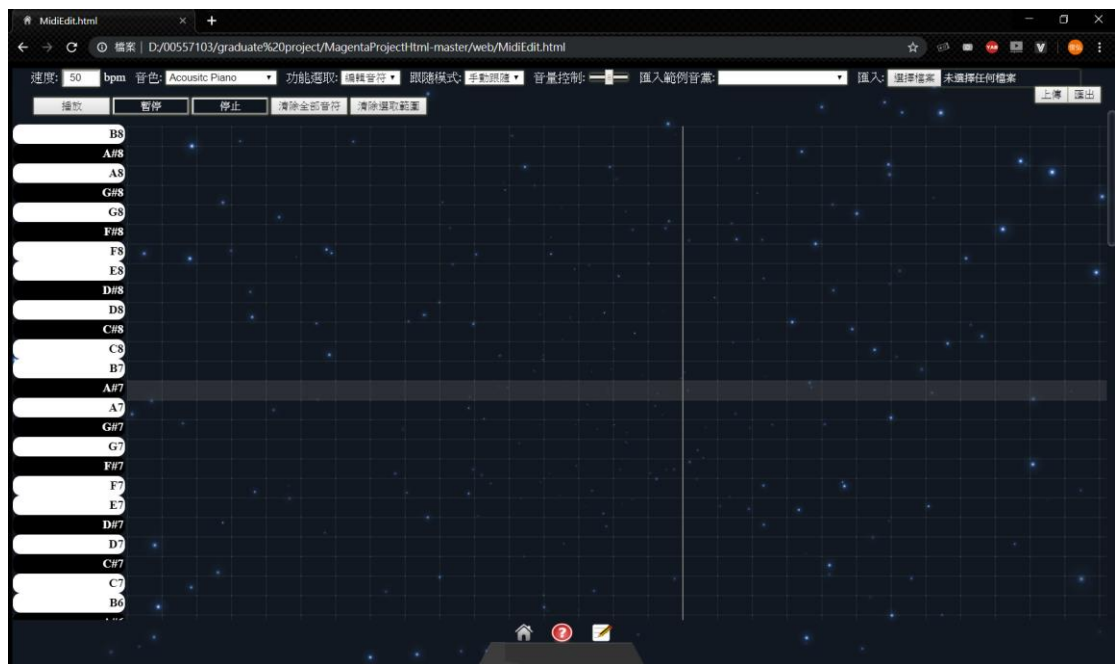


圖 15 編輯頁面，可執行許多編輯功能



圖 16 幫助頁面，讓使用者可以使用鍵盤操作，更加方便





圖 17 速度(BPM)、音色、音量、功能選取(編輯音符、段落選取)、  
跟隨模式、撥放功能、匯入匯出

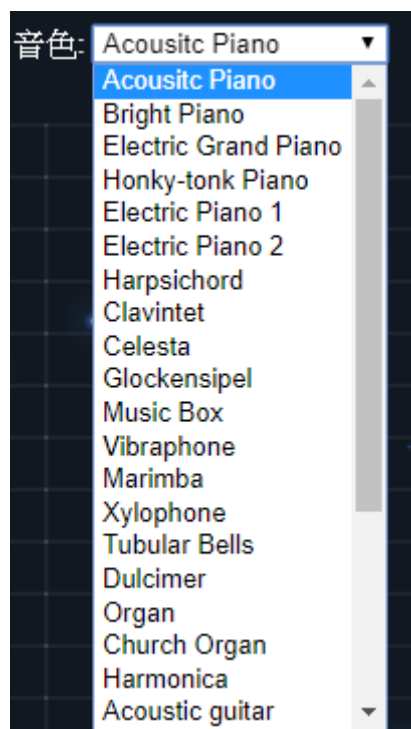


圖 18 音色選取，有不同的音色可以選取

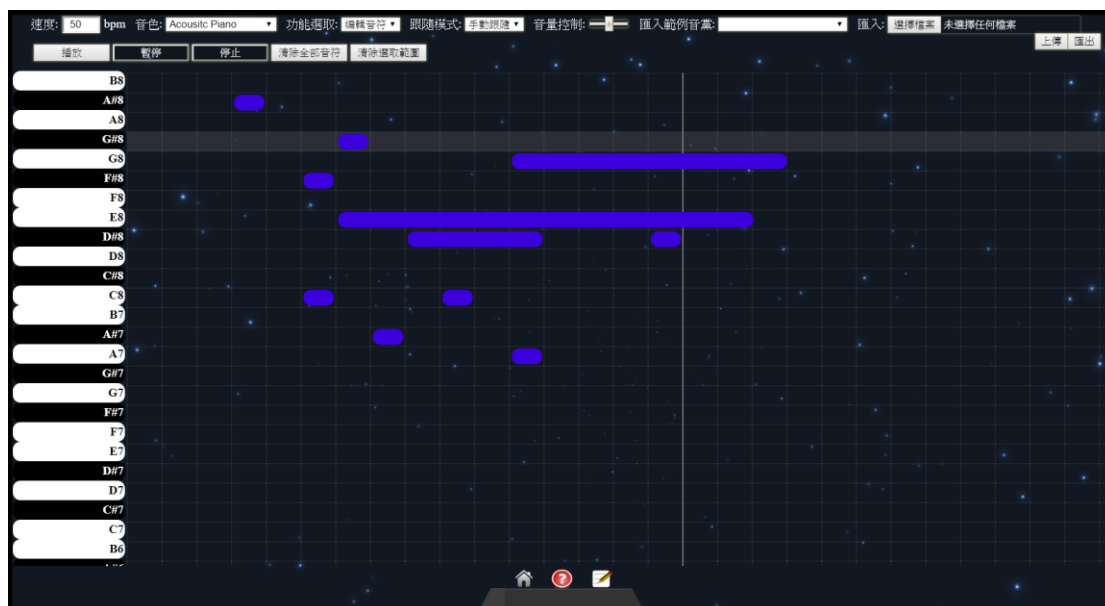


圖 19 編輯情況，音符的延長、音符的拖曳

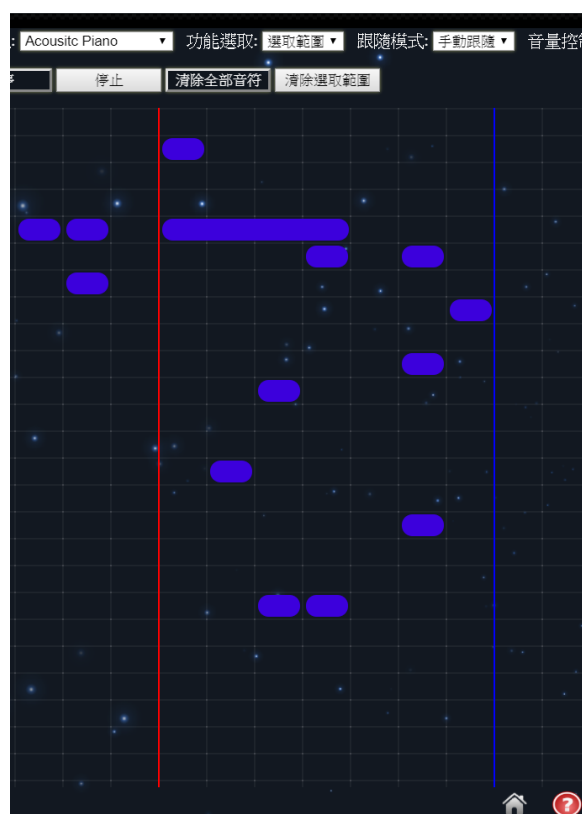


圖 20 範圍選取，可以指定播放區塊(紅線至藍線)

### 3-3、實際操作

#### 3-3-1 初始設定

一、進入網頁編輯區時，可以選擇匯入 museGAN 生成的音樂或者自行編輯。(圖 21~圖 30)

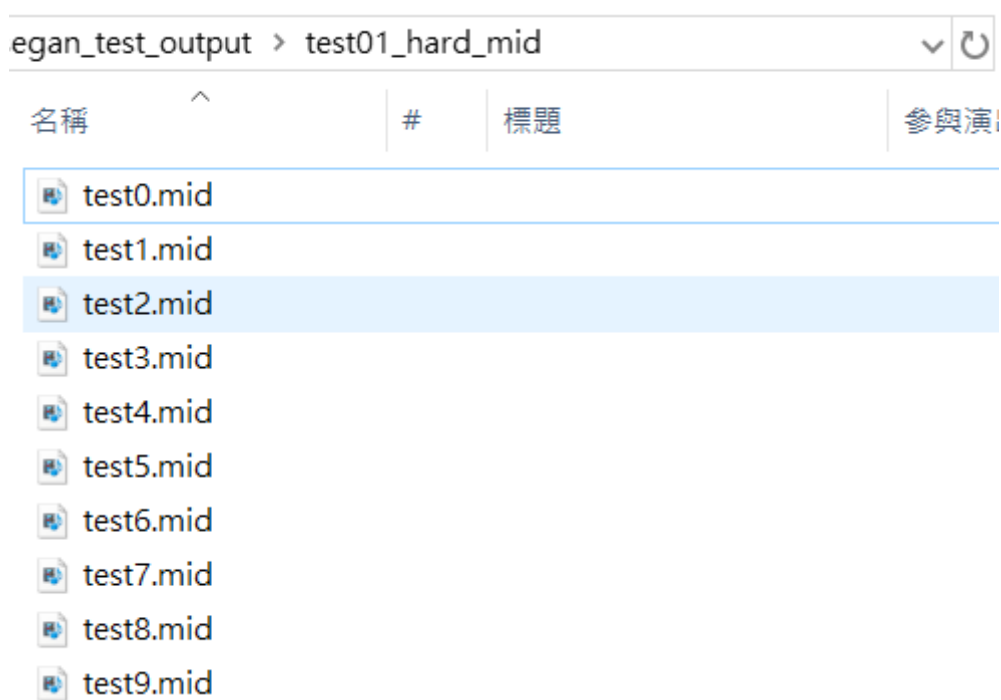


圖 21 生成完的 midi 檔案

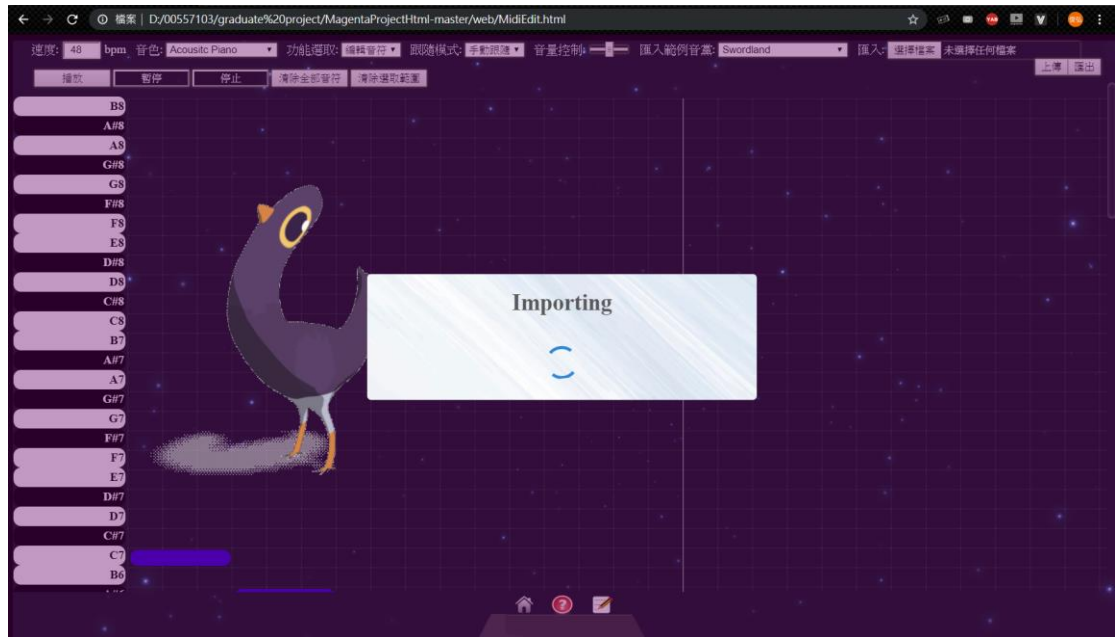


圖 22 匯入 museGAN 生成好的 midi 檔案中

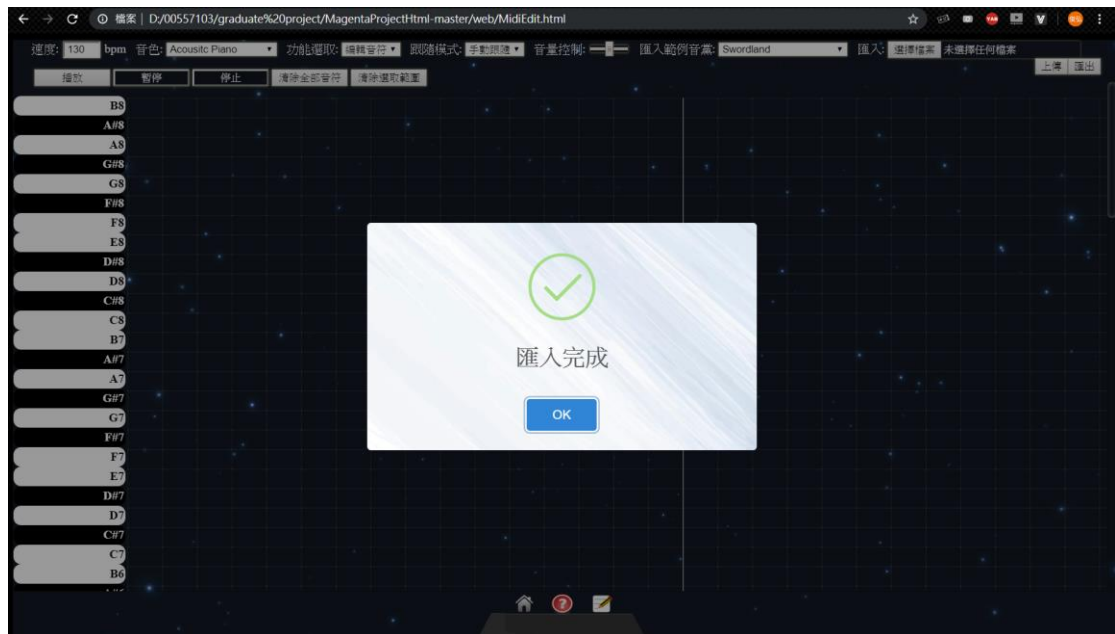


圖 23 匯入完成

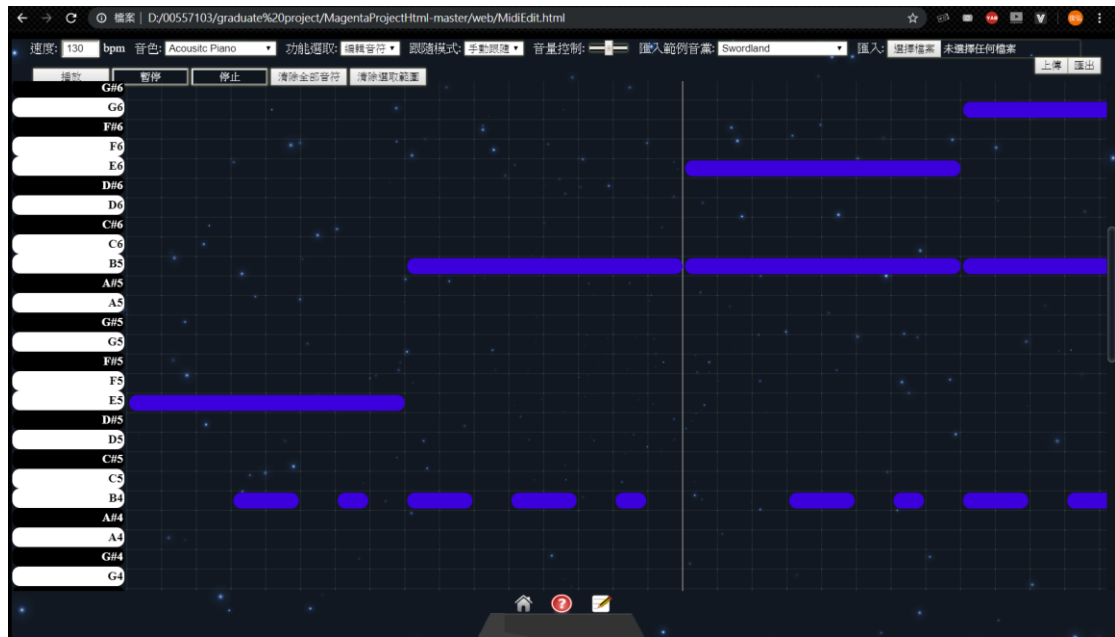


圖 24 匯入狀況，顯示在編輯頁面上

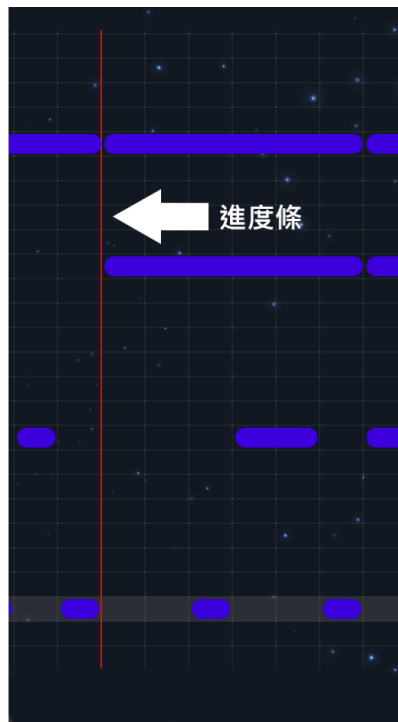


圖 25 進度條，在按下播放鍵後會有進度條(紅色線)隨音樂節拍進行播放

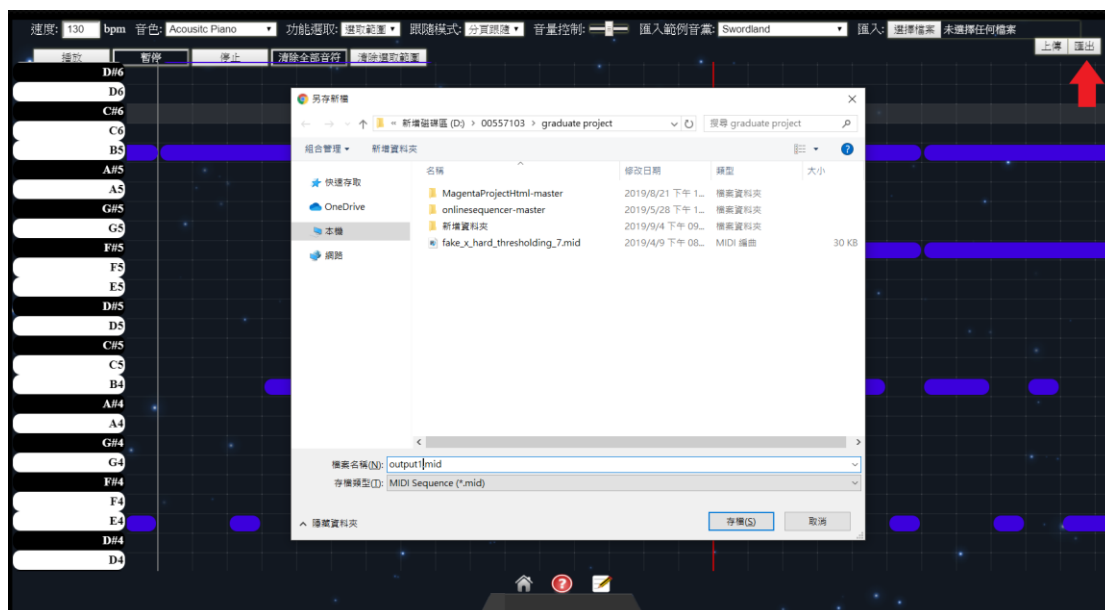


圖 26 匯出 midi 檔案

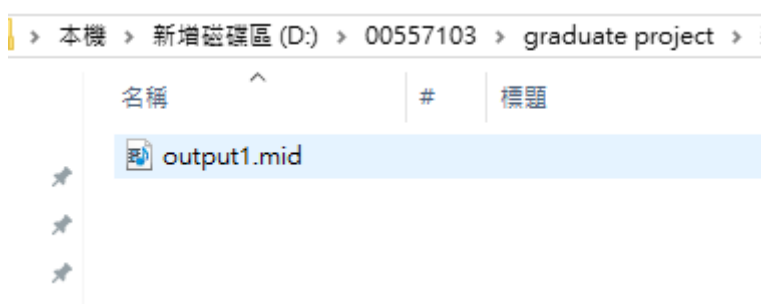


圖 27 匯出狀況，下載在使用者設定的路徑中

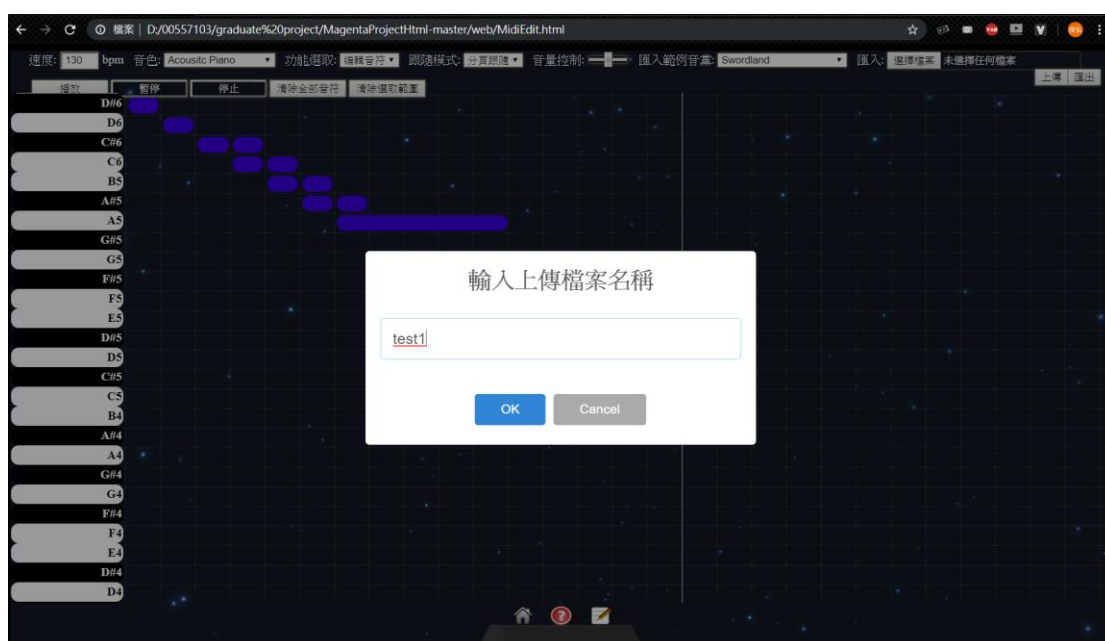


圖 28 上傳至 firebase 資料庫(ex:檔案名稱 test1)

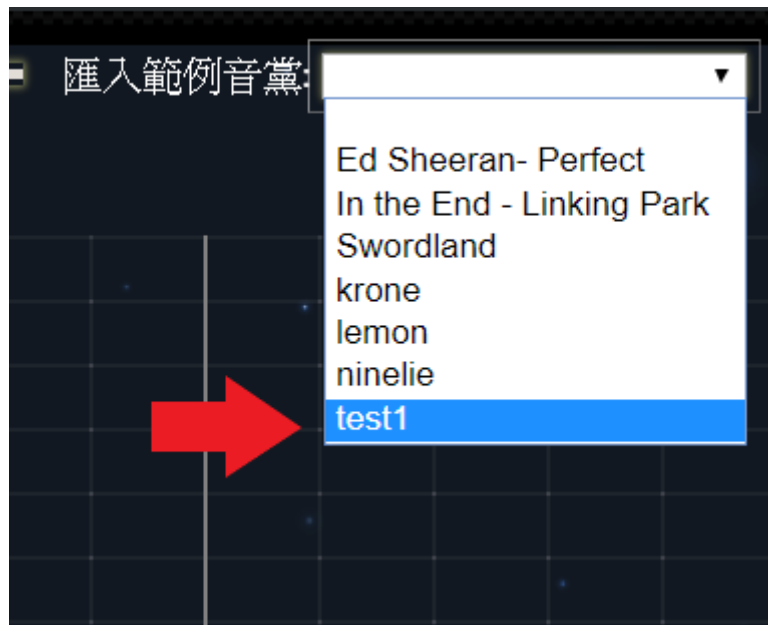


圖 29 上傳至 firebase 資料庫的音檔重整之後會出現在範例音檔中

### 3-3-2 實際使用

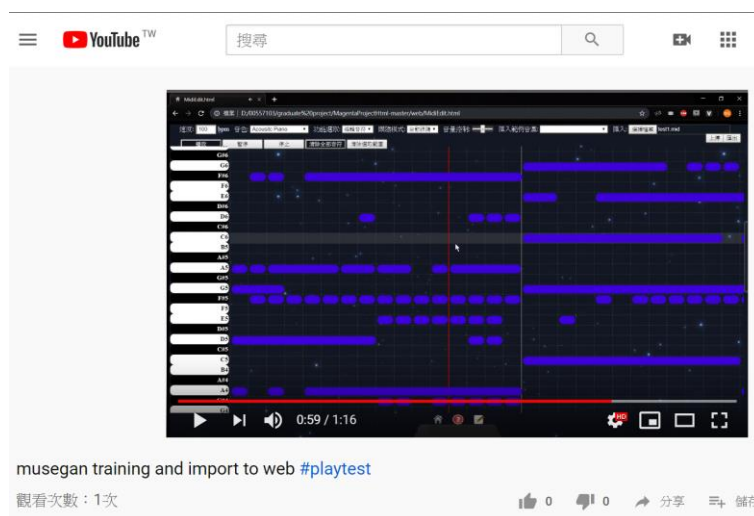


圖 30

以生成的 midi 檔案匯入進行播放，網址如下：

<https://www.youtube.com/watch?v=xmX62Y0UfXE>

### 3-4、程式碼架構(圖 31~圖 32)

HTML				
主畫面、幫助、編輯頁面				
 help.html	2019/7/28 上午 0...	Chrome HTML D...	1 KB	
 index.html	2019/7/28 上午 0...	Chrome HTML D...	2 KB	
 MidiEdit.html	2019/7/28 上午 0...	Chrome HTML D...	5 KB	

圖 31 html

JavaScript				
編輯及物件化				
 MidiEdit.js	2019/7/28 上午 0...	JavaScript 指令檔	23 KB	
 object.js	2019/7/28 上午 0...	JavaScript 指令檔	2 KB	
網頁背景、音樂控制及外掛套件				
 back.js	2019/7/28 上午 0...	JavaScript 指令檔	3 KB	
 bar.js	2019/7/28 上午 0...	JavaScript 指令檔	1 KB	
 controller.js	2019/7/28 上午 0...	JavaScript 指令檔	8 KB	
 playmidiscript.js	2019/7/28 上午 0...	JavaScript 指令檔	4 KB	
 property.js	2019/7/28 上午 0...	JavaScript 指令檔	3 KB	
 receiver.js	2019/7/28 上午 0...	JavaScript 指令檔	7 KB	
 synth.min.js	2019/7/28 上午 0...	JavaScript 指令檔	63 KB	
 synth.min.js.map	2019/7/28 上午 0...	Linker Address M...	137 KB	
 worker.js	2019/7/28 上午 0...	JavaScript 指令檔	1 KB	

圖 32 JavaScript

## Chapter 4 討論

### 4-1 成效討論

本研究成果，MuseGAN 在生成音樂的時候，需要花費很長時間，



需要 10 幾個小時以上，所以成效不佳。

在後端與前端的連結上，因為後端生成的音樂完成後，沒有辦法直接傳到前端，需要手動匯入，後端與前端沒有完全整合好。

因為是用網頁製作，所以再匯入音檔的時候如果遇到多音軌或比較大的音樂檔案會匯入得比較久，可能會造成 RAM 使用過度的情形。

#### 4-2 優缺點

優點：

- 因為是網頁，所以在使用上比起音樂軟體更加方便
- 直觀操作，方便使用
- 編輯成本低

缺點：

- 功能比真正的音樂編輯軟體少了許多
- 進度條因為是跟音樂不同方式設計的，所以會造成些微的不同步
- 對於匯入一般的歌曲(4 分鐘以內)是使用正常的，但對於匯入大的檔案需要花費更多時間等待，會造成 RAM 使用過度。
- 缺乏後端與前端的完整性

-需要花費長時間訓練生成音樂

#### 4-3 未來展望

短期目標：

-增加更多功能。

中期目標：

-若要改善後端與前端的連結上，可以使用 Ruby on Rails(RoR)技術，可以將後端的資料傳給前端網頁。

長期目標：

-要改善花費長時間生成音樂，若要即時達到作曲編輯的效果，就必須研究其演算法，改善其複雜的數學公式。  
-可以開發成 APP 供使用者使用

### Chapter 5 結論

#### 5-1 結論與建議

目前所呈現出來的部分仍有些缺陷需要在未來解決。對生成音樂以及編輯音樂，應該要再想辦法使角度更全面，要讓它可以成為即時作曲的系統。另外，目前只有網頁的形式，希望可以讓他的相容性更好，也可以開發成 APP，讓使用者可以隨手一支手機，創作一首歌曲。

#### 5-2 成果自評

當初構思題目時想做有關人工智慧的題目，而後來發現在音樂生成這一塊其實也有很多人在嘗試，而自己也想嘗試看看，雖然是使用別人已經寫好的 museGAN 來生成音樂，但也因為這樣閱讀了許多 paper，了解到想要運用 AI 生成音樂確實需要花費許多功夫。這過程我們確實也遇到滿多技術的問題與挑戰，而也在教授指導之下試著去克服、試著去突破它。經過這次的專題，我們更瞭解到自主學習的能力培養重要性，有太多情況與東西是我們無法在課堂上有時間去學習到的。如何在意見分歧時整合大家的想法、如何切入重點探討問題，以及在時間、成本下該如何解決問題。我們同時也學到不同以往的思考方式，過去學期型的專題大部分都是思考自己怎麼樣去處理會比較好寫；但在這專題時，需要進行角色轉換，「如果我是使用者，我會覺得……」，這個切入點可以幫助我們對自己開發的專題更為嚴謹與更加符合大眾需求，這些都是未來我們進入業界後都需要面臨與面對的，趁這個機會體會到，並非壞事。

### 5-3 指導教授評語

音樂合成技術結合人工智慧發展快速，本專題除了有效運用 deep learning 在核心技術生成音樂外，在大暑計畫中更發揮學生們在過去所培育的前端技術，將所生成的音樂片段得以視覺化，建

立了方便編輯的介面，並克服匯入、顯示與編輯等問題，讓此平台得以有效且直覺式的呈現，相信學生在這樣的經驗中磨練出解決問題的能力，期待可將此平台發揮至更多時間序列資料上，並且整合更多的資料庫將此平台的效益擴大。

## Chapter 6 參考文獻

1. <https://www.w3schools.com/>
2. <https://salu133445.github.io/bmusegan/pdf/bmusegan-ismir2018-paper.pdf>
3. <https://salu133445.github.io/musegan/pdf/musegan-aaai2018-paper.pdf>
4. <https://salu133445.github.io/musegan/pdf/musegan-ismir2017-lbd-paper.pdf>
5. <https://salu133445.github.io/musegan/model>
6. <https://jazz-soft.net/>
7. <https://sweetalert2.github.io/>
8. <https://medium.com/@afun/ubuntu-16-04>
9. <https://askubuntu.com/questions/841876/how-to-disable-nouveau-kernel-driver>
10. <http://blog.davidou.org/archives/1387>
11. <https://vimsky.com/article/3872.html>
12. <https://www.rolandtaiwan.com.tw/glossary/t.htm>