

Multimedia Computing Environment hw2

BY B04902073 資工三 洪崇凱

註：Word 文件使用 pdf2doc.com 轉檔服務，格式與 pdf 如有落差還請見諒

1 開發工具

1. 使用框架

- Python 3.4.0
- Gstreamer 1.0
- Gtk 3.0

2. 依賴套件

- gizeh
- moviepy
- mido
- intervaltree

2 功能構想

我在 YouTube 找尋背景音樂時看到一類影片，是將 MIDI 編輯器的播放過程錄製下來，作為電子鋼琴樂的播放畫面。除了讓一般的音樂影片增加視覺化的趣味，欲使用此樂譜的人也能直接預覽樂譜的品質。



我想將這項功能擷取出來：使用者能開啓 MIDI 檔案並預覽、儲存成播放影片。由於應用程式是呼叫自己寫的函式庫，甚至可以使用 shell 大量轉換多個 MIDI 檔案；同時使用者對畫面有極大的控制度。

3 環境建構

3.1 Debain Linux

以下設定在 Ubuntu 16.04.4-desktop-amd64 作業系統測試成功。

1. 目錄下打開終端機，安裝依賴套件

```
$ sudo apt install gstreamer1.0-plugins-bad python3-pip ffmpeg libffi-dev  
$ pip3 install --user gizeh moviepy mido intervaltree
```

3.2 Windows

以下設定在 Windows 7 sp1 64-bits 作業系統測試成功。

1. 安裝 Python 3.4 ^[1]
2. 安裝 PyGObject for Windows ^[2]
 - a) 在 GNOME libraries 選單中選擇以下套件
 - Base packages
 - Gst-plugins
 - Gst-plugins-extra
 - Gst-plugins-more
 - Gstreamer
 - GTK+
 - JSON-glib
 - b) 在 non-GNOME libraries 不需選擇任何套件
 - c) 在 development packages 不需選擇任何套件
3. 開啓 cmd.exe 為依賴套件安裝做前置準備

```
> python -m pip install --upgrade pip  
> pip install requests pycparser
```
4. 從 Unofficial Windows Binaries for Python Extension Packages ^[3] 下載已編譯的 wheel 檔案
 - cffi-1.11.5-cp34-cp34m-win_amd64.whl
 - moviepy-0.2.3.4-py2.py3-none-any.whl
5. 再次開啓 cmd.exe 安裝依賴套件

```
> pip install cffi-1.11.5-cp34-cp34m-win_amd64.whl  
> pip install moviepy-0.2.3.4-py2.py3-none-any.whl  
> pip install gizeh mido intervaltree
```

6. 開啓 C:\Python34\Lib\site-packages\cairocffi__init__.py

找到第 41 行並更改

```
cairo = dlopen(ffi, 'cairo', 'cairo-2')
```

成為

```
cairo = dlopen(ffi, 'cairo', 'cairo-2', 'cairo-gobject-2')
```

註: Windows 未內建支援 webm 的播放器, 可下載 VLC media player。

4 實際操作

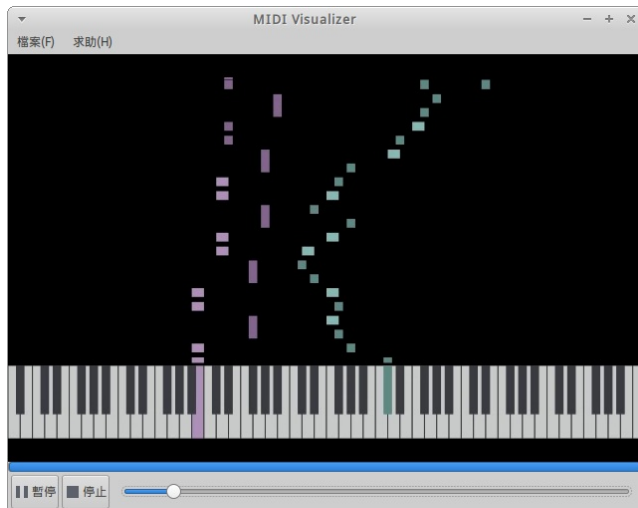
執行 main.py, 即有應用程式視窗出現:



開啓檔案選擇選單列的「檔案」→「開啓」, 之後需等待程式將 MIDI 解析並生成影片, 下方進度條後指示目前進度, 以範例檔案皆需大約 1 分鐘。



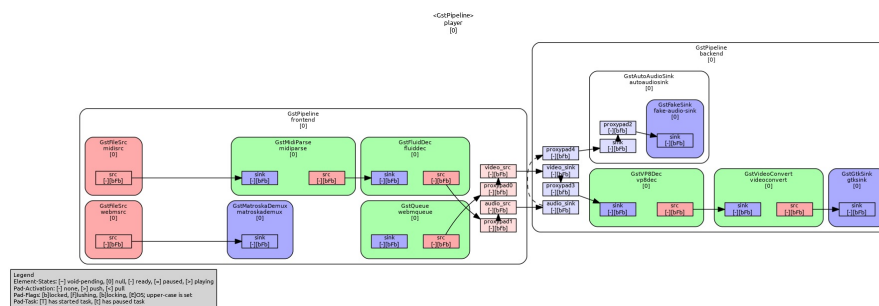
之後即可下方播放列預覽影片內容, 或者選擇「檔案」→「儲存/另存新檔」儲存所錄製的影片。



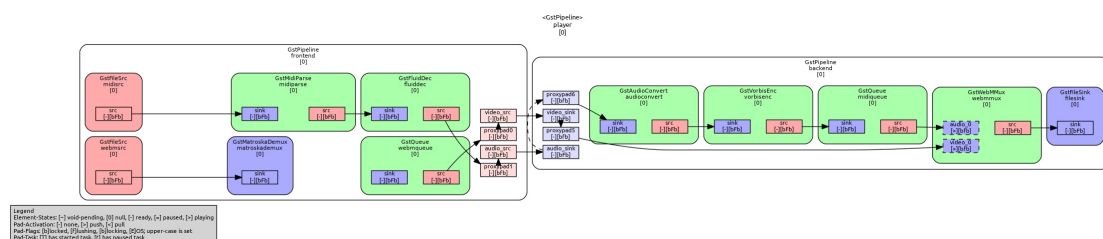
作業檔案中，midi/ 資料夾有兩個範例檔案，其輸出結果放置於 doc/output/ 資料夾。

5 細節解釋

受到依賴套件 moviepy 功能限制，僅能逐張渲染畫面並接為暫存影片。開啓時產生的檔案僅有畫面，沒有音檔。是透過 Gstreamer 同步播放兩者達成預覽效果。



儲存時臨時改動 Gstreamer pipeline 架構，將連結目標由播放後端改至儲存後端。由於前端不變，可以預期成果相同於播放時的預覽畫面。



6 遭遇問題

1. 由於依賴套件僅將 MIDI 做低階轉換，將其生成資料結構的程式碼仍是自己製作。遇到怪異的 MIDI 訊息只得放棄處理。但其他軟體似乎皆能正確處理異常訊息，或許需要找到更多相關文件閱讀。
2. 因為逐張渲染畫面，生成影片的效率極低，需要將近輸出影片一半的時間。我認為更好的作法是直接編寫 Gstreamer 插件，達成即時串流生成影片，但開發難度大幅提升而未採用。
3. Windows 下應用程式無法播放預覽，已經確認是在上圖播放 pipeline 中的後端中，其中一個套件錯誤引起，目前猜測是 autoaudiosink 引起。但不影響最重要的儲存功能。

7 作業心得

這次作業間接使用到 `cairo` 這個開源繪圖程式。雖然在 Linux 類作業系統中受到廣泛使用與支援，在 Windows 上建立環境卻成為難事。幸好有前次作業的嘗試經驗而順利建構。但 Windows 的建構第 6 步應該是 `cairoffi` 未能正確找尋 `PyGObjects for Windows` 動態函式庫引起，之後需要發送 `pull request` 以修正問題。

編寫解析檔案的功能時，碰巧遇到 `intervaltree` 上游依賴套件 `sortedcontainers` 改版為 2.0，造成 `intervaltree` 查詢功能失效。幸好當天即有網友發送 `pull request` 修正問題，主作者也在昨天 (2018/05/28) 就接受並納入主版本，目前已能下載到正常運作版本。

8 補充資料

8.1 參考資源

1. Python GStreamer Tutorial
from <https://brettviren.github.io/pygst-tutorial-org/pygst-tutorial.pdf>
2. PyGObject API Reference
from <https://lazka.github.io/pgi-docs/>
3. List of Elements and Plugins
from <https://gstreamer.freedesktop.org/documentation/plugins.html>

8.2 標注連結

- [1] <https://www.python.org/downloads/release/python-340/>
- [2] <https://sourceforge.net/projects/pygobjectwin32/>
- [3] <https://www.lfd.uci.edu/~gohlke/pythonlibs>