

Toccata

IDE para el trabajo con Lilypond

Javier Salazar Loyola

Diciembre 2010

Perspectiva Global del Problema

Creación y Edición de Partituras Musicales

Constántemente se ven los músicos en la necesidad de crear partituras.

- Trabajos Propios
- Transcripciones
- Arreglos

Antiguamente Las partituras eran hechas a mano, por artistas que dedicaban su vida a ello: Creación de prensas

Actualidad Varios “buenos” programas comerciales del estilo WYSIWYG

- Sibelius (Sibelius Software Ltd)
- Finale (MakeMusic! Inc)

Ambas opciones son bastante caras: sus resultados tienen el problema del formato

- En internet se encuentran buenas fuentes de partituras clásicas o modernas
- Sin embargo, están incompletos muchas veces, o están demasiado completos: hay que extraer las partes individuales de las partituras de las grandes sinfonías

Lilypond es un lenguaje de marcado del estilo de \LaTeX , especializado en edición de partituras y fácilmente integrable con éste último: esta misma presentación ha sido creada de ese modo: Los primeros compases de “Los Pollitos Dicen...”.



Los Primeros compases del himno de Israel (nótese el cambio de tonalidad como única diferencia)



Complejidad Lilypond es un lenguaje lleno de detalles. El código para el himno de Israel:

```
\relative c' {\key c \minor \time 4/4 c4 d  
ees f g2 g}
```

El usuario objetivo, el músico, generalmente no es programador

Solución una IDE que simplifique el uso del programa: Esta es la solución que busca Toccata.

Toccatà

Porqué Toccatà

Toccatà se refiere a varios tipos de piezas en música:

- Puede ser parte de las *suites* barrocas
- Es el preludio en algunas piezas (especialmente en las de Bach)
- Se dice que Bach en su Toccatà y Fuga en Re menor, BWV 565, la compuso para probar los órganos donde tocar.

El tema de la fuga:



Python y PyQt4

Por qué este Lenguaje y esta Librería

- Multiplataforma
- PyQt4 puede ser poderoso y estético
- Rápido desarrollo, al carecer de compilación (y no tener mucho tiempo de carga)
- Lilypond está repartido entre Python, C++ y Scheme

- MDI (Multiple Document Interface), para la edición de proyectos grandes
- Resaltado de texto (incluidos fragmentos en Scheme)
- Generación de partitura con un botón
- Soporte (básico, en estado embrionario) para visualización de PDF
- Guardado automático en archivo .log del registro de mensajes, además de su despliegue.
- Compilación no detiene al programa (multihebra)

- Poco configurable. Cambiar de lector implica cambiar el código.
- La interfaz MDI puede mejorarse.
- Aún puede hacerse más para poder integrar Lilypond-Book.
- La interfaz puede hacerse más amigable con el uso de algunas herramientas WYSIWYG o con la inclusión de módulos de sonido.

- Lilypond es un lenguaje poderoso y de resultados admirables. Una IDE debe poder adecuarse a todas esas herramientas de la mejor forma posible.
- PyQt4 provee una gran cantidad de herramientas para el desarrollo, y si la velocidad del lenguaje interpretado es poca, puede traducirse de inmediato a C++.
- Si se deja de lado la necesidad de velocidad de ejecución extrema, Python puede proveer todo lo que uno necesita.
- El proyecto Lilypond lleva siendo trabajado desde 1995, es ridículo pensar que se puede alcanzar trabajando sólo unos meses, por lo que queda mucho trabajo que realizar.

“Sin música, la vida sería un error”

(F.W. Nietzsche)

“En verdad, si no fuera por la música, habría más razones para volverse loco”

(P. I. Tchaikovsky)

“La música constituye una revelación más alta que ninguna filosofía”

(L. van Beethoven)