

Toccata

IDE para el trabajo con Lilypond

Javier Salazar Loyola Cristobal Troncoso

Junio 2009

Perspectiva Global del Problema

Creación y Edición de Partituras Musicales

Se ven los músicos en la necesidad de crear y editar partituras. Constantemente se ve la necesidad de tener una buena herramienta de edición para los que crean sus propias partituras o editan otras para su propia comodidad en la lectura.

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

Posible Solución en Grabado de Partituras

Lilypond es un lenguaje interpretado tipo \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)



- Uno de los principales inconvenientes con Lilypond es su complejidad: demasiados detalles, a veces muy engorrosos, y el usuario final de este lenguaje es músico: no un programador
- Posible solución: implementación de una IDE para el trabajo con el lenguaje: ese es el trabajo a desarrollar este proyecto, en estado embrionario

Tocatta

Porqué Tocatta

Una toccata puede referirse a varios tipos de piezas (no a mini conciertos de bandas locales)

- Por un parte, se la puede encontrar en las *Suite*, como parte de ellas
- Puede ser una pieza que sirve de preludio a otra mayor
- Bach las utilizaba para probar los órganos en los que iba a tocar (cf. Tocatta y Fuga en d (re menor), BWV565)

El tema de la fuga:



Tocata

La Razón de ser del Nombre

Para Bach, una toccata era para probar el instrumento, para ayudar a que funcionara bien.

Ese es el objetivo de este programa: ayudar al desarrollador a que la partitura quede bien hecha.

El inicio de la Toccata:



Python y PyQt4

Porqué Python y Porqué PyQt4

- Porque no son Java
- Porque ambos están pensados para ser multiplataforma
- Porque PyQt4 es estético y poderoso a la vez (algo en lo que falla Swing)
- Porque con la sintaxis de Python y PyQt4 pueden hacerse cosas muy interesantes (ejemplo de ello es el manejo de eventos por medio `QtCore.SIGNAL` y `QtCore.SLOT`)
- Porque Lilypond tiene como principales códigos para trabajar a C++, Python y Scheme

Toccata - IDE para el desarrollo de Lilypond

Primera Ventana completamente Desarrollada: Embrión

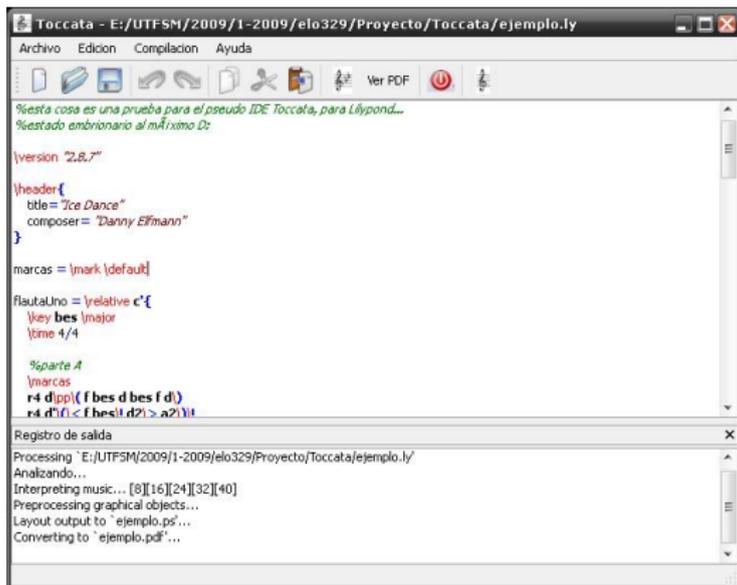


Figura: Ventana Principal

Algunos Casos de Uso

Resaltado de Sintaxis - I

Nombre Resaltar Sintaxis

Propósito Implementar colores y distintos tipos de fuentes según sea pertinente

Actores Objeto que resalta (class Highlighter)

Pre-condiciones El objeto editor de texto tiene texto cargado

Evento Carga de un nuevo archivo “.ly” o ingreso de nuevo texto al archivo cargado

Post-condiciones El objeto editor muestra el texto en formato enriquecido, con la sintaxis resaltada

Tipo Automático: gatillado por eventos

Curso Normal de eventos

- 1 Carga el archivo o se modifica el ya cargado
- 2 El objeto encargado de resaltar el texto lo hace
- 3 El texto se muestra formateado de acuerdo a ciertas reglas básicas

Curso Alternativo De no poderse generar el resaltado por algún error desconocido (Durante las pruebas se depuraron todos los que fueron posibles), el programa sigue corriendo, pero sin el resaltado de texto

Requerimientos No Funcionales Ninguno explícito

Algunos Casos de Uso

Compilación con Lilypond - I

Nombre Compilar Archivo

Propósito Generar el archivo PDF y PS con la partitura para imprimir

Actores Lilypond, Usuario

Pre-condiciones El archivo debe estar guardado y el directorio debe tener permiso de escritura

Evento Usuario lo solicita explícitamente

Post-condiciones Los archivos PDF y PS quedan listos para ver, y el usuario puede ver en la ventana de registro los mensajes del programa (útiles en caso de fallo en la compilación)

Tipo Manual: gatillado por el usuario

Curso Normal de eventos

- 1 Se crea un proceso hijo con el programa Lilypond
- 2 Lilypond intenta compilar el archivo
- 3 El resultado se muestra en el Registro de Salida

Curso Alternativo Si está instalado Lilypond, no debiera haber ningún curso alternativo. El proceso hijo compila o no, el objetivo es que el resultado se vea en la ventana de registro, de modo de poder corregir de inmediato los errores que pudieran generarse.

Requerimientos No Funcionales Ninguno explícito

Algunos Casos de Uso

Ver archivo PDF o PS generado - I

Nombre Ver Partitura Generada

Propósito Invocar a un editor que muestre la partitura generada

Actores Visor de PDF o PS

Pre-condiciones El archivo debe estar compilado y los visores deben existir

Evento Usuario lo solicita explícitamente

Post-condiciones Los archivos son visibles mediante visores de PDF y PS

Tipo Manual: gatillado por el usuario

Curso Normal de eventos

- 1 Carga el visor con el archivo PDF o PS generado
- 2 El visor queda disponible para el usuario

Curso Alternativo Dependiendo del visor, puede que éste no abra en caso de que el archivo no exista, lo que puede pasar si es que aún no se compila.

Requerimientos No Funcionales El visor no se cierra al cerrar el programa principal.

Diagrama UML para el Caso de la Compilación o algo parecido, al menos

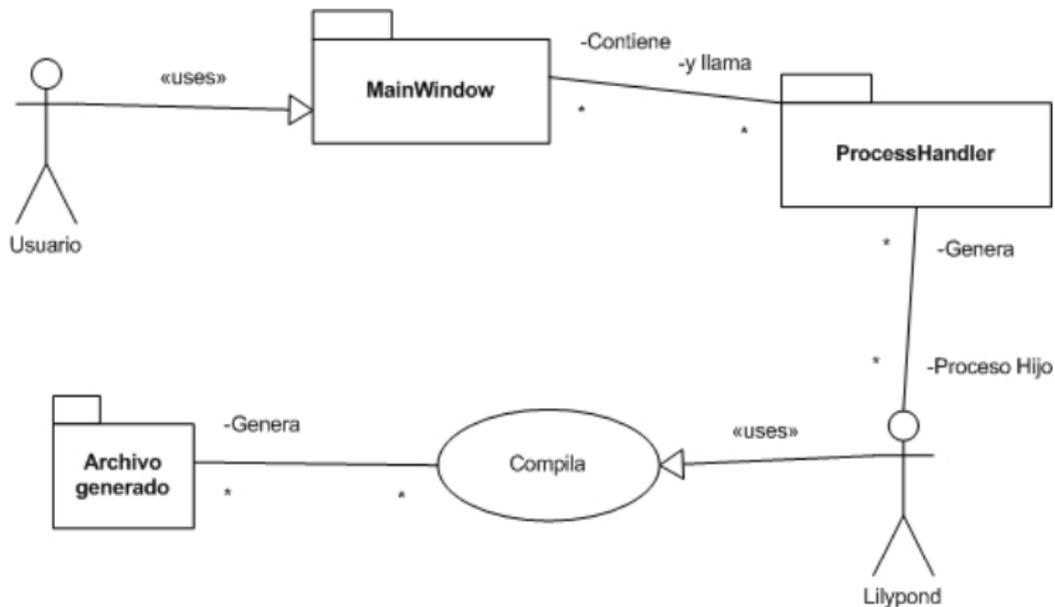


Figura: Diagrama para el caso de la Compilación

Conclusiones

Problemas durante el desarrollo

Muchos problemas surgieron durante el desarrollo de este proyecto, como por ejemplo:

- Aprender PyQt4 de la forma más eficiente posible
- Aprender a usar subprocessos y obtener la salida de éstos
- Obtener el resaltado de sintaxis adecuado
- Priorizar objetivos en aras de cumplir plazos
- Aún queda demasiado desarrollo por hacer

Debido a la infinita gama de posibilidades en la notación musical, el desarrollo a seguir desde ahora podría enfocarse a:

- Permitir la configuración del programa (visores preferidos, banderas de Lilypond, colores del resaltado, etc)
- Integrar las funcionalidades del lenguaje Lilypond con el programa (generación de plantillas básicas, etc)
- Implementar una interfaz MDI (MultiDocument Interface)
- El objetivo principal: crear una interfaz gráfica que permita el ingreso de las notas, articulaciones, expresiones, etc en estilo WYSIWYG, pero con resultados en Lilypond.

“Sin música, la vida sería un error”

(F.W. Nietzsche)