

Toolbox para Aplicações Musicais na Internet

Márcio de Oliveira Costa^{1,2}, Jônatas Manzolli^{1,3}

¹Núcleo Interdisciplinar de Comunicação Sonora (NICS)

²Instituto de Computação

³Instituto de Artes, Departamento de Música

Universidade Estadual de Campinas (UNICAMP)

“Cidade Universitária Zeferino Vaz”, Rua da Reitoria 165,

Caixa Postal 6166, 13081-970, Campinas, SP, BRAZIL

{marcio, jonatas}@nics.unicamp.br

Abstract. *A toolbox for multi-platform applications using the MIDI protocol is presented here. The aim was to develop a set of fundamental routines to be used as Internet musical tools. This was done using the JAVA2 SDK 1.3 language, its sound manipulation library and multi-treading components. The Java Virtual Machine, Java Sound Classes and the Software Synthesizer were the basic Java resources explored in this research. The toolbox performance was evaluated for different platforms (Spark Station, Mac and PC) and OS (Solaris, Windows and Linux).*

Resumo. *Um toolbox para aplicações multiplataforma utilizando o protocolo MIDI é aqui apresentado. O objetivo do trabalho é desenvolver rotinas fundamentais para serem utilizadas como meio de interação musical na Internet. A pesquisa foi realizada com a versão da linguagem Java (JAVA2 SDK 1.3), que possui uma vasta biblioteca para manipulação sonora e recursos de multi-treading. Foram exploradas as possibilidades oferecidas pelo JAVA2 - a Java Virtual Machine, as novas classes para manipulação sonora e o sintetizador software. O desempenho do toolbox foi avaliado frente a diferentes plataformas (Sparc, Mac e PC) e sistemas operacionais (Solaris, Windows e Linux).*

1. Introdução

A utilização da World Wide Web como um novo meio de expressão musical tem sido discutida recentemente em diversos contextos. Seja sobre a distribuição em larga escala de produções sem o aval de gravadoras, produtores e compositores, seja sobre a sua utilização como fonte de pesquisa musicológica ou, em última instância, dimensionando-a como uma nova mídia para inserção da produção musical como estudado por (Burk)1998, Helmuth (2000) e Yamagishi & Setoh (1998). Dentro desta última abordagem, apresentamos nesta artigo a implementação de uma ferramenta para a utilização criativa-musical da Internet. Desenvolvemos um toolbox em JAVA2 que possibilita *composição musical compartilhada por multi-usuários* utilizando recursos cliente-servidor conforme proposto por Burk (2000). Nas próximas seções há uma breve apresentação do paradigma multiplataforma utilizando a *Java Virtual Machine*, das estruturas musicais implementadas e dos resultados experimentais utilizando o toolbox em três plataformas e sistemas operacionais diferentes. No texto que se segue, todavia, há conceitos derivados de Orientação a Objetos como divisão em classes, o uso de métodos, herança e polimorfismo que o leitor interessado encontra em Ritchey (1999). Utilizamos também, no desenvolvimento do controle sonoro em tempo real, a implementação de *threads* descrito em Tanenbaum (1999).

2. Composição Musical Compartilhada

Um aplicativo on-line permite que a criação musical se propague em larga escala. Vários usuários utilizando o mesmo aplicativo têm a percepção da composição produzida localmente, mas a máquina servidora congrega a música composta por todos os que utilizam o programa o que gera questionamentos sobre a autoria das composições e da própria latência inerente à Internet que influencia a percepção do *tempo musical*. Há dificuldades técnicas como a grande quantidade de arquiteturas de computadores e sistemas operacionais que dificultam o desenvolvimento de aplicativos que atendam ao maior número possível de usuários. Trabalhando-se com som há a agravante de uma grande gama de placas disponíveis no mercado e nem todas são compatíveis e têm qualidade sonora. Com a popularização da WWW ganhamos um novo meio de comunicação que pode ser explorado musicalmente, mas entre outras questões colocamos: *Como unificar esta diversidade de plataformas e sistemas operacionais? Como criar um aplicativo musical que possa ser executado e compartilhado nesta diversidade?* A pesquisa aqui apresentada, procura responder esta duas questões.

2.1. A Abordagem Java

Ambientes de composição musical importantes como Max não são multiplataforma, o que prejudica o seu uso generalizado e global. Todavia o desenvolvimento do jMax em Java possibilitará a sua utilização em várias plataformas como apresentado em Déchelle et all (2000). Utilizando-se a linguagem Java cada camada do aplicativo possui um único representante. Todo o projeto é desenvolvido tendo em vista um sistema neutro, a *Java Virtual Machine*. A Interface Gráfica (GUI) é padronizada e “idêntica” para todos os sistemas operacionais. A camada do programa é feita usando o paradigma de orientação a objetos criando um sistema muito estruturado, facilmente reaproveitado e expandido por heranças e polimorfismo. Na camada de som, o novo kit de desenvolvimento da Sun (vide Sun Microsystems, Inc. (2000), presente na nova Virtual Machine, evita a instalação de *plugins* e gera mensagens MIDI em tempo real. Este kit possui também um sistema de software-síntese compatível com a especificação “*General MIDI*, um padrão mínimo de qualidade sonora. Atualmente existem implementações da Java Virtual Machine versão 1.3 para windows (95,98,ME,NT,2000), Mac OS X, Unix (Solaris) e Linux, o que atinge praticamente todos os computadores e foram escolhidas cada uma destas plataformas para aplicar os testes do toolbox.

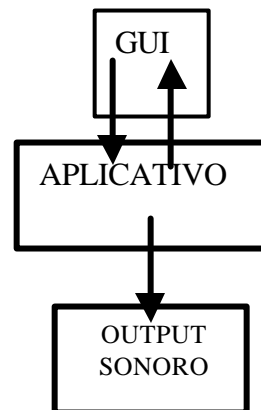


Figura 2.0 diagrama de uma aplicação multiplataforma utilizando a Java Virtual Machine.

3. Toolbox para Aplicações Musicais

O objetivo da pesquisa foi transformar o ToolBox num conjunto de ferramentas musicais para interação em tempo real, contendo desde eventos simples como notas (com uma duração preestabelecida), passando por acordes, arpejos, seqüências etc, e todo o desenvolvimento do software deveria ocorrer de forma eficiente e inteiramente transparente.

3.1 Organização das Classes Desenvolvidas

Temos uma classe principal que cuida do controle em tempo real das notas tocadas chamada NoteCollector (um coletor de notas). Uma instância dessa classe é usada na classe Tvoz, que acrescenta outras funcionalidades ao pacote do Java (javax.sound.midi). Separar o controle (NoteCollector) das estruturas criadas (Tvoz) permite que usuários avançados modifiquem o controle em tempo real ou expandam sua capacidade sem se preocupar com o tipo de evento musical utilizado.

3.2 Coletor de Notas

O coletor de notas trata-se de uma *Thread* que é disparada automaticamente numa base de tempo pré-estabelecida, nest caso 20ms. A cada NoteO, segue o número de vezes que a base de tempo deve permanecer ativada. No tempo especificado a *thread* libera esse comando para o sintetizador Java, que processa o comando e gera a saída sonora. O coletor de notas na verdade age como um disparador de eventos. Trata-se de uma *thread* que desperta a cada base de tempo. Quando desperta, percorre uma lista que contém em cada posição Note Number, Velocity e o tempo de disparar cada mensagem (Note Off, evento com Velocity igual a zero).

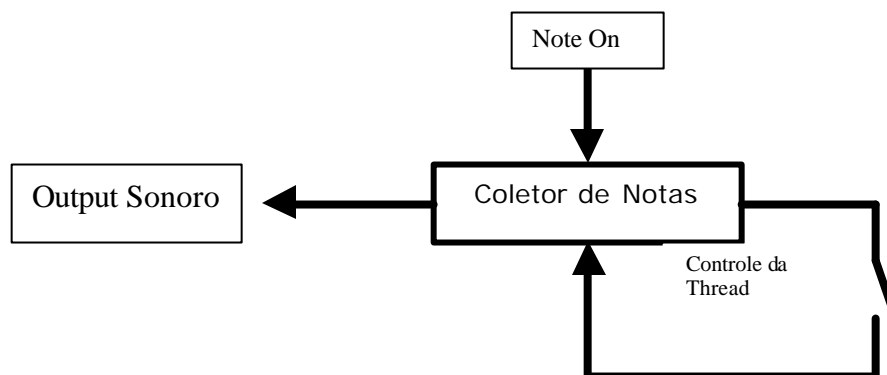


Figura 4.0 Diagrama do Coletor de Notas controlado pela thread desenvolvido para controlar o fluxo de eventos MIDI em tempo real.

3.3 Estruturas Musicais

3.3.1 Delayed Note On

Esse método permite criar um evento Note On para ser executado num tempo futuro e especificar a duração do mesmo. O evento Note On é então enviado para o coletor de notas como mostra as linha de código abaixo:

```
public void DNoteOn(int n, int v, int d, int t)
{
    collector.Note(n,v,d);
    collector.Note(n,0,d+t); }
```

3.3.2 Acordes

Esse método determina a nota fundamental de um acorde de 4 notas, controla o seu tipo dentro de uma tabela de acordes possíveis e indica sua inversão. Veja abaixo a chamada do método:

```
public void PlayAcorde(int notaBase,int vel,int tempo, int tipo, int rev)
```

3.3.3 Arpejos

Analogamente ao método acima, este permite tocar um arpejo de 04 notas determinando a inversão e a articulação (variando de 0 a 100%). Veja abaixo a chamada do método:

```
public void PlayArpejo (int notaBase, int vel, int tempo, int tipo, int rev,int Art)
```

3.3.4 Combinação de Eventos

Trata-se de um método que gera uma sequência de arpejos diferentes com durações cada vez mais distantes e todos feitos a partir da mesma nota fundamental. Temos um número par de arpejos escolhido aleatoriamente (ex. 2, 4, 6) e seus parâmetros (articulação e inversões). Este foi um dos testes usados para verificar se o toolbox respeitava a duração entre as notas e para testar se as estruturas sonoras programadas soariam da maneira esperada utilizando-se muitas chamadas de arpejos que sobrecarregam a thread.

4. Resultados e Aplicações

4.1 Ambiente de Teste

O ambiente de teste consistiu de um programa em modo texto que executava acordes, notas e arpejos nas máquinas alvo. Abaixo apresentamos os resultados com comentários e observações para cada caso.

Plataforma	Sistema Operacional	Memória RAM	Desempenho
Sparc 20	Solaris 7	32	Satisfatório
K6 133	Win2K	64	Ineficiente
K6 133	Linux	64	Excelente
PII Clerom 600	Linux	32	Excelente
P MMX 266	Win98	32	Excelente

TABELA 1.0 Apresenta as diferentes plataformas onde o toolbox foi testado e o respectivo desempenho.

Esses testes foram realizados apenas com o programa em Java operando no sistema e criando-se dinamicamente vários acordes, arpejos e notas. Chamamos de insatisfatório o sistema que não manteve o "tempo" entre as notas (atrasos irregulares, o sistema todo manteve-se instável e o sintetizador midi "ficou mudo"). Satisfatório quando 90% das vezes não houve problemas; notas e arpejos tocaram normalmente. Algumas vezes percebeu-se um atraso na resposta, mas isso foi raro. Excelente quando não foi percebido nenhum atraso nem mesmo valendo-se de instrumentos para medir o tempo entre as notas.

4.2. Estúdio Virtual

Com um dos resultados obtidos esperamos rescrever alguns aplicativos já desenvolvidos no NICS em outras linguagens e disponibilizá-los juntos num conjunto de Applets, servindo assim como embrião do Estúdio Virtual.

4.3. Rabisco

Implementamos também um ambiente para desenvolvimento da criatividade sonora chamado Rabisco, que associa sons a desenhos feitos com o mouse utilizando-se uma GUI. Trata-se de um projeto em conjunto com o PGL (Partnership in Global Learning) envolvendo pesquisadores do Brasil, México, Estados Unidos e Canadá e um total de 30.000 alunos da escola fundamental na sua primeira fase. Uma versão alfa do Rabisco encontra-se disponível em <http://www.nics.unicamp.br/rabisco>.

5. Conclusão

Com a Internet aparecem novas possibilidades de interação entre músicos, pois usuários em pontos diferentes podem compartilhar informação e contribuir simultaneamente na criação de uma obra através da WWW. Com este objetivo, desenvolvemos um aplicativo para Internet que permite criação musical em tempo real e nos preparamos para a implementação do Estúdio Virtual que permitirá um nível de interação maior, permitindo composições conjuntas.

Tecnicamente, percebemos que a configuração aceitável para o aplicativo desenvolvido necessita principalmente de uma boa quantidade de memória RAM. Acreditamos, pelos testes realizados, que 64MB sejam suficientes. Quanto ao processador, qualquer um com o desempenho pelo menos igual a um Pentium MMX 233 é suficiente. Uma configuração inferior necessitaria de uma carta de som de boa qualidade com controle MIDI via hardware para melhorar o desempenho. O toolbox realmente facilita o desenvolvimento e reaproveitamento de código como comprovamos no desenvolvimento do Rabisco, praticamente não houve dificuldades com o controle sonoro.

Referências

- Burk, Phil (2000) "Jammin' on the Web - a new Client/Server Architecture for Multi-User Musical Performance" Proceedings of the International Computer Music Conference 2000:117-120
- Burk, Phil (1998) "JSyn - A real - time Synthesis API for Java" Proceedings of the international Computer Music Conference 98:252-255.
- Déchelle, Schnell, Borghesi, Orio (2000) "The jMax environment: an overview of new features" Proceedings of the International Computer Music Conference 2000:252-257.
- Helmuth, M. (2000) "Sound Exchange and Performance on Internet2" Proceedings of the International Computer Music Conference 2000:121-124.
- Ritchey, Tim (1999) "Programming with Java!". New Riders Editor.
- Tanenbaum, (1999). Sistemas Operacionais "Projeto e Implementação" 2ª ed Woodhull Bookman ISBN 85-7307-530-9
- Sun Microsystems, Inc. (2000) "Java™ Sound API Programmer's Guide". Documentação completa e oficial da versão de JDK usada.
- Yamagishi & Setoh (1998) "Variations for the WWW - Network Music by Max and the WWW", Proceedings of the International Computer Music Conference 98:510-513.