

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

CARLO SULZBACH SARTORI
HYAGO SALLET

TOCADOR DE MÚSICA EM JAVA

Prof. Marcelo Soares Pimenta

INF01120 - Técnicas de Construção de Programas

Semestre 2014/1

INTRODUÇÃO

O programa em questão, denominado *Music Composer*, é um tocador de música a partir de texto estruturado e não estruturado. Foi desenvolvido como trabalho prático final para a disciplina de Técnicas de Construção de Programas, utilizando-se a linguagem de programação Java.

Em especial, para os efeitos de sons, foi usada a biblioteca *JFugue*, que implementa boa parte da funcionalidade necessária para o programa, como tocar notas baseada em uma *string* de entrada. Para o restante, foram usadas bibliotecas, classes e métodos genéricos da linguagem escolhida.

A construção se deu com a aplicação dos conceitos vistos ao longo do semestre na disciplina, como aqueles referentes à qualidade de *software*, modularidade e convenções de programação. Assim, foi feito um programa que procura seguir todos os conceitos para se adequar a um mundo exigente de *softwares*, onde as qualidades internas e externas são importantes juntas e onde manter um código legível e de fácil manutenção reduz muitos custos.

REQUISITOS

Parte Funcional:

1. Tocar música a partir de uma entrada textual do usuário:
Definição básica do programa, ele mapeia o texto de entrada e toca uma música baseado na entrada.
2. A entrada em questão poderá ser via arquivo ou digitação direta:
O usuário pode carregar a entrada de um arquivo texto, ou digitar diretamente na caixa de texto.
3. Opção para salvar o texto atual em arquivo:
Além de ler de um arquivo, é possível salvar o texto atual em disco também.
4. Permitirá *pausar/parar* a música a partir de um botão:
A música pode ser pausada, ou parada, neste último caso, pressionar o botão de execução faria com que a música comesse do início, e não do ponto onde foi parada.
5. Seleção do instrumento que tocará a música:
Uma lista com os instrumentos permite que se escolha qual deles tocará a música como um todo.
6. Seleção da velocidade com que a música será tocada:
Um campo permite que seja selecionada a velocidade (em BPM) com que a música irá tocar.
7. Mapeamento de notas em arquivo texto:
O mapeamento das notas para o modo iniciante não fica no código, mas sim em um arquivo texto, permitindo que o usuário o modifique.
8. Restauração do arquivo de mapeamento a partir do programa:
O arquivo de mapeamento pode ser restaurado a partir do programa, evitando possíveis erros no qual o usuário (desavisado) acaba por excluir o arquivo do sistema.
9. Dois modos de uso: *Iniciante* e *Avançado*:
Um permite texto corrido, sem formatação alguma, o outro permite formações mais complexas, mas que exigem maior conhecimento. Abrange uma gama maior de usuários.

Parte não funcional:

1. Interface simples:
Poucos botões, com pouca abertura para ambiguidade. Cada objeto faz uma operação única definida.

2. Informar ao usuário todos possíveis erros:

Todos os erros que acontecem são informados ao usuário, em alguns casos com uma sugestão de como contornar o problema para que o programa volte a funcionar normalmente.

3. Janela de instruções:

Uma janela de instruções com explicações sobre o programa.

4. Exemplos definidos:

Um conjunto definido de exemplos para que o usuário possa entender, na prática, o funcionamento do programa.

PARTE FUNCIONAL

Classe *MainWindow*: Responsável pela janela principal do programa. Ela mantém toda a interface feita com o usuário e a parte funcional do sistema, através dos botões e dispositivos de entrada e saída do *Music Composer*.

Classe *TextFile*: Uma importante classe para o programa, ela mantém o Tipo Abstrato de Dado "Arquivo Texto", com as operações necessárias para sua manipulação, leitura e escrita, criação e outros. É responsável pelas ações de salvar e ler de um arquivo texto, bem como pela criação do arquivo de mapeamento, quando este ainda não existe no sistema.

Classe *MusicalThread*: Implementa *threads* para as músicas, de forma que uma música terá uma *thread* só dela dentro da JVM, permitindo que outras operações ocorram em paralelo a sua execução.

Classe *InstructionWindow*: Implementa a janela de instruções para o *Music Composer*. Uma classe simples, mas que foi criada para facilitar a manutenção e melhorar a modularidade do código.

Classe *InfoMessages*: Classe que contém todas as mensagens informadas pelo programa ao usuário final. É ainda a classe que exibe as mensagens, criando a pequena janela onde elas serão mostradas ao usuário.

Classe *Music*: Serve para instanciar objetos do tipo *music*, os quais são representações do Tipo Abstrato de Dado Música no universo do *Music Composer*. Uma música contém uma melodia, uma velocidade e um instrumento que a toca. Mas também conta com métodos para normalizar a entrada (*InputParser*), para tocar a música, para pausar a música, entre outros.

Classe *MusicalNotes*: O módulo *MusicalNotes* tem como único objetivo a definição das notas musicais, isto é, das construções básicas válidas no *Music Composer*. Como por exemplo, a definição das notas(caracteres) que serão SOL, DÓ, RÉ, MI, as durações, os tempos e assim por diante.

Além disso, a fim de tornar melhor a experiência do usuário com o software, a *MusicalNotes* permite que o mapeamento das notas musicais seja feito da maneira externa ao programa(via arquivo texto). Dessa forma, o usuário pode escolher quais caracteres ele deseja para as notas musicais, durações ou tempos.

Classe *MusicalMap*: O módulo *MusicalMap* está diretamente relacionado ao *MusicalNotes*. O papel dele é construir, com base nas definições das notas musicais do *MusicalNotes*, um Mapa Musical onde cada caractere está associado a uma Nota, Duração ou Tempo musical.

Além do *Map* implementado pelo *MusicalMap* tornar claro o mapeamento das notas musicais, a implementação é bastante eficiente em termos de custo computacional. O mapa musical nada mais é do que um array de strings onde cada célula possui a nota, duração ou tempo correspondentes a um caractere que é dado pela índice da célula (código ASCII do caractere). Por isso, o acesso é direto e, portanto, o mapeamento é feito em tempo constante $O(1)$.

Classe *InputParser*: Este módulo se comporta como uma espécie de interface entre o mundo externo e o *software*. Basicamente, dado uma string qualquer como entrada o *InputParser* se comunica diretamente com o *MusicalMap* transformando a entrada em uma string válida que será de fato utilizada pra representar a melodia a ser tocada.

Além de transformar cada caractere da entrada em uma nota válida o *InputParser* normaliza a entrada de forma a manter a consistência com a biblioteca de áudio *JFugue*.

INTERFACE COM O USUÁRIO

O *Music Composer* mantém uma interface simples, mas que atende as necessidades dos usuários que farão uso do sistema. Estes são desde pessoas sem qualquer conhecimento do universo musical, até semi-profissionais com profundo conhecimento da música.

Há duas opções diferenciadas na tela principal, uma para os usuários *iniciantes na música* e outra para os mais *avançados*. Para ajudar, o programa conta com exemplos e uma janela de instruções, explicando brevemente sobre cada modo, e do programa como um todo.

Em qualquer um dos dois modos a funcionalidade é a mesma, uma música será executada com base na entrada textual fornecida pelo usuário. Em especial, a entrada pode ser via digitação direta, ou via leitura de um arquivo texto. Para a execução da música, ainda é possível selecionar o instrumento e a velocidade com que ela será tocada.

A simplicidade da interface foi mantida de tal forma que o *Music Composer* tem apenas duas janelas, descontando as de erros, que são apresentadas ao usuário. A principal, em torno da qual todo o programa gira, e a janela de instruções, citada anteriormente. Abaixo há uma explicação mais detalhada sobre cada uma e sobre seu *layout*.

Janela Principal

A janela principal do programa conta com todos os botões necessários para o funcionamento básico do programa, sem a necessidade de exigir que o usuário utilize dos menus disponíveis no topo dela. A interface tenta deixar tudo claro para um uso lógico e fácil, e para isso todos os botões, além de contarem com textos significativos, contam com textos de ajuda, que são mostrados ao se deixar o mouse sobre cada um deles.

O campo de texto em tamanho mediano faz com que o programa não ocupe um espaço desnecessário na tela, mas permite que mesmo textos grandes sejam digitados sem causar dificuldade na leitura posterior. Além disso, seu tamanho é expansível na medida em que barras de rolagem são inseridas caso o texto continue além do limite visível para a direita, ou para baixo.

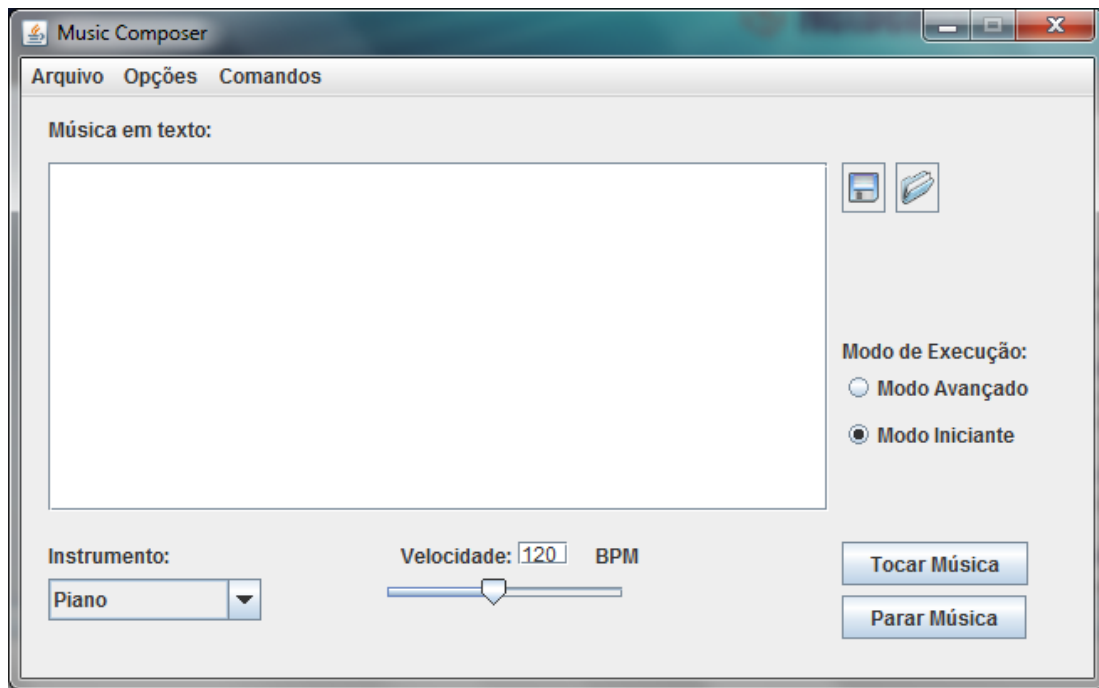


Imagem 1 - Janela principal do Music Composer

No lado direito da tela estão concentrados diferentes botões. Na parte superior está o botão de *Salvar* e *Abrir* arquivos texto. Um pouco mais abaixo estão os de seleção do modo de execução do programa. Na parte inferior estão os dois botões de controle da música, *Tocar Música* e *Parar Música*. Sempre que *Tocar Música* é pressionado, ele é desabilitado até que a música termine, ou ser parada pelo usuário, impedindo assim o erro em que quando o botão era clicado novamente, a música reiniciava.

Na parte inferior da tela está o seletor de instrumento, uma lista exaustiva ordenada alfabeticamente com as opções disponíveis de instrumentos. O número de opções, seguindo a biblioteca *JFugue* é 128 instrumentos, porém apenas alguns destes foram selecionados e colocados neste lista. Ao lado do seletor está a barra de rolagem para definir a velocidade em *beats* por minuto com que a música será tocada. O valor pode ser modificado entrando com um número na caixa de texto, ou simplesmente rolando a barra.

Os menus acima permitem ações simples como: salvar arquivo, abrir arquivo, sair e limpar a caixa de texto. Mas também oferecem algumas mais complexas, tal como criar o arquivo de mapeamento de notas, abrir janela de instruções e carregar exemplos.

Janela de Instruções

A janela de instruções, acessada através do menu *Comandos -> Instruções*, exibe textos que podem ajudar os menos familiarizados com o programa. Em sua parte central ela conta com o texto sendo mostrado no momento, e na parte inferior conta com três botões para navegar entre: *Instruções (gerais)*, *Instruções do modo iniciante* e *Instruções do modo avançado*. Ao se pressionar cada um desses botões, ele será desabilitado para que o usuário tenha conhecimento de onde se encontra, e o texto mostrado será modificado para o respectivo texto do botão selecionado.

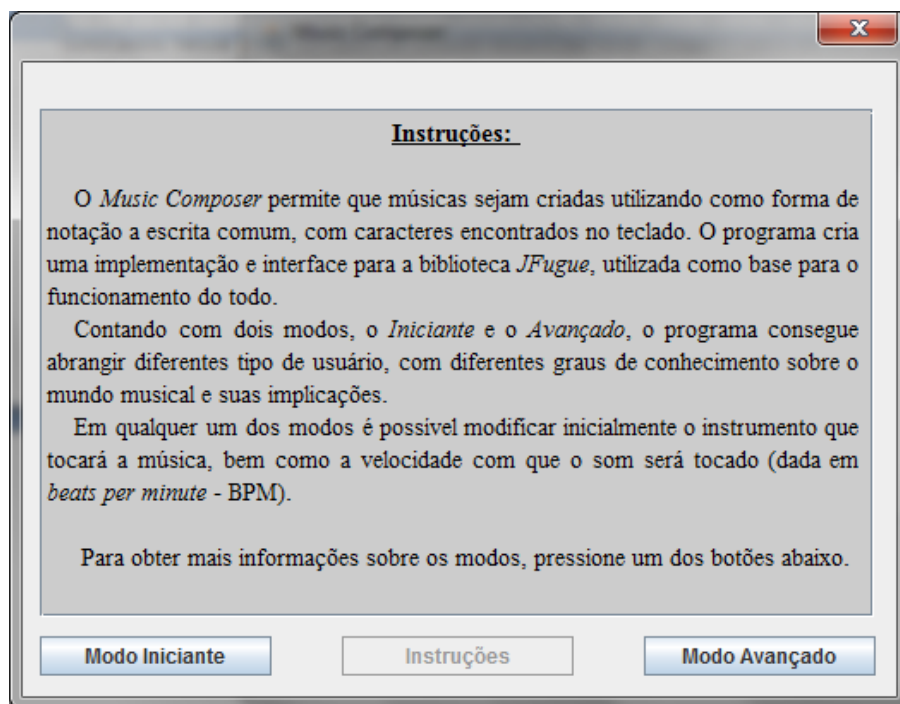


Imagem 2 - Janela de Instruções

As instruções para o modo iniciante contam com o mapeamento de cada nota, mostrando os caracteres disponíveis e qual nota, oitava ou duração eles representam.

Já para o modo avançado, é mostrado um pouco sobre a biblioteca utilizada, a *JFugue*, e um *link* de um manual é disponibilizado para aqueles que querem ler mais sobre o modo de uso completo da biblioteca, bem como seus possíveis comandos para uso no modo avançado do programa.