

Nome: Cauê Scotti (00338597), Galeano Folle (00334948), Luca Boni (00337818) e Pedro Roewer (00330062) | Data: 14/08/2023

High UFRGS Musical: road to INFINITY - Etapa 2

1. Mudanças em Relação à Etapa Anterior:

O grupo ponderou a retirada de algumas funcionalidades, como efeitos aplicados aos sons, distorção e efeitos de modularização do áudio. Também optou deixar a aplicação dessas funcionalidades como uma possível extensão a ser explorada.

O slider de volume teve seu design redesenhado, passando de um botão giratório para um slider convencional.

Houveram adições de atributos em algumas classes de modo a permitir maior legibilidade e desempenho (pois não é preciso calcular a cada nova iteração determinados valores).

Métodos também foram adicionados, para garantir que cada função seja responsável por uma tarefa, aumentando assim a legibilidade do código.

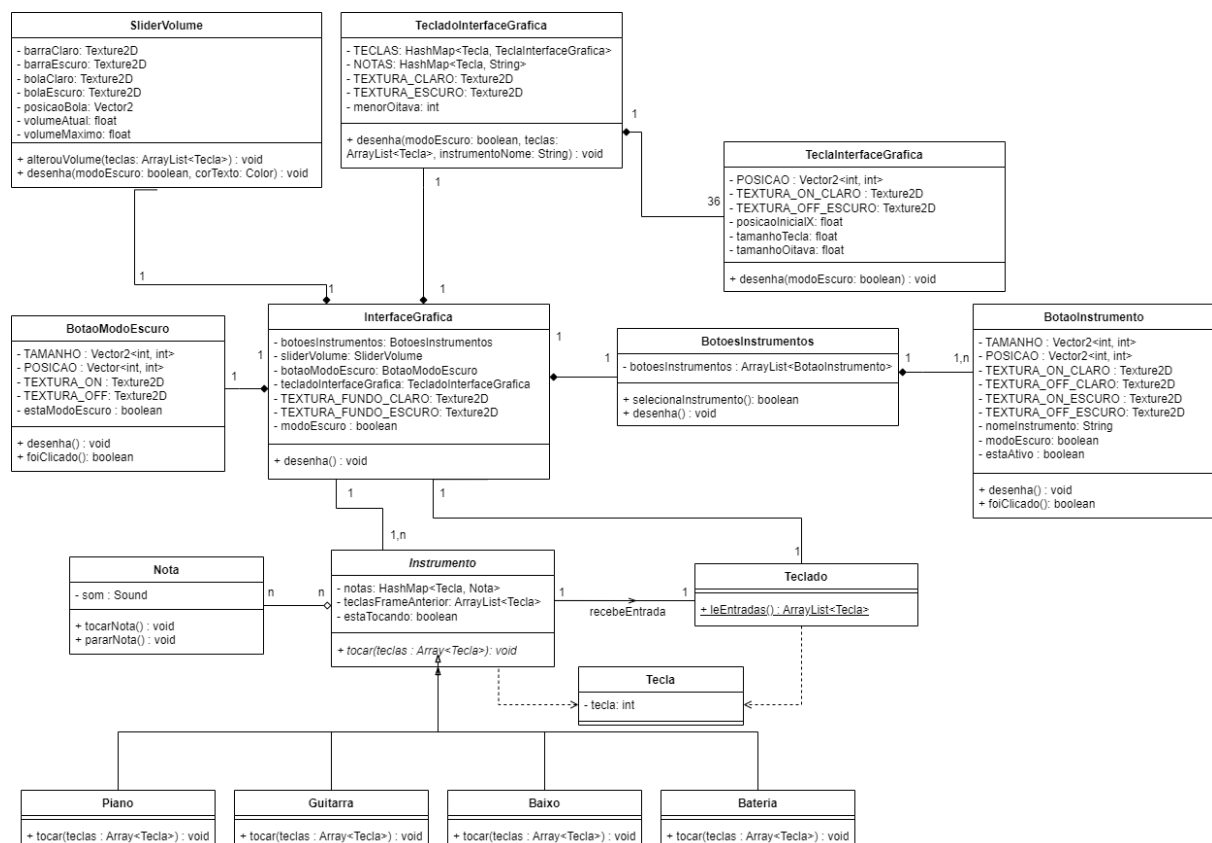


Imagem 1: diagrama de classes reformulado.

2. Código

A fim de simplificar a implementação, o código pode ser dividido em três partes principais: implementação gráfica, classes de instrumentos e classes de leitura de teclado.

2.1. Leitura de teclado

Três classes são utilizadas na tarefa de reconhecer e tratar as teclas pressionadas pelo usuário: Tecla, Instrumento e Teclado. Os métodos utilizados para a leitura, assim como na implementação gráfica, foram importados da biblioteca Jaylib.

O método `Jaylib.isKeyDown` identifica todas as entradas do teclado em um determinado momento, retornando todas as teclas pressionadas naquele instante. Esse retorno é fornecido por um valor inteiro correspondente ao código ASCII do caractere. No entanto, a biblioteca Jaylib foi programada para reconhecer caracteres no padrão americano. Portanto, a classe Tecla é utilizada para realizar essa conversão por meio de um switch case. Dessa forma, a leitura do teclado pode ser feita de maneira correta e consistente.

A classe Teclado está relacionada com a classe Tecla mencionada anteriormente. Através de um método estático chamado `leEntradas()`, a classe Teclado retorna um `ArrayList` de Tecla. Esse array contém todos os caracteres pressionados em um instante, ou seja, proporciona um mapeamento completo do teclado do usuário após a conversão realizada pela classe Tecla.

Conforme descrito na seção 2.1 "Instrumentos", há um arquivo de texto que relaciona cada caractere a uma nota, a qual é armazenada em um arquivo de áudio. A classe abstrata Instrumento é responsável por receber, em seu construtor, um arquivo de texto (txt) e estabelecer uma relação entre todas as teclas e suas respectivas notas por meio de um `HashMap`.

2.2. Instrumentos

O programa possui uma classe abstrata chamada Instrumento, que serve como base para as classes filhas: Baixo, Bateria, Guitarra e Piano. Essa classe possui um construtor que procura nos arquivos do projeto um mapeamento de teclas e notas. O arquivo contém em cada linha um caractere, seguido de um separador ('@'), e um nome de arquivo de áudio. Dessa forma, estabelece-se a relação entre o teclado e os sons correspondentes. Cada classe filha utiliza o construtor `super()` e implementa o método `tocar()`, que recebe uma lista de teclas (`ArrayList<Tecla>`), e faz tocar as notas correspondentes. Cada instrumento faz isso de uma forma diferente:

A Bateria toca até o fim os arquivos de áudio relacionados a uma tecla, não importando se o usuário deixa de pressionar o teclado, simulando de fato uma bateria (uma vez que um instrumentista bate com a baqueta em um tambor, o som permanece tocando).

A Guitarra e o Baixo têm uma implementação similar: o usuário primeiro escolhe as notas a serem tocadas, e pressiona a tecla <Enter>, que “libera” o som, simulando um toque nas cordas. Quando o usuário para de pressionar <Enter>, o toque continua, porém quando remove os dedos das teclas, o som para.

O Piano, por sua vez, tem a abordagem mais simples: a nota toca enquanto o usuário mantém pressionada a tecla correspondente.

2.3 Implementação gráfica

A interface gráfica consiste no uso da biblioteca Raylib, que consiste em uma janela na qual podemos colocar objetos e implementar a lógica através destes objetos. A interface

é inicializada com parâmetros como: taxa de FPS, tamanho da tela e nome da aba, os botões presentes dentro da interface agem como seletores do instrumento a ser tocado, utilizando a captura do mouse como parâmetro, e o display das notas representa os arquivos de áudio que estão em execução no momento, todas elas são inicializadas pela função `inicializaJanela()`.

A interface também tem como componente principal a implementação de uma interface de áudio completamente suportada pela Raylib, utilizando parâmetros como valor de áudio, sendo o slider de áudio a representação visual deste valor dentro da aplicação, que é atualizado a cada passagem do loop de execução.

Outra feature interessante implementada pelo grupo foi a interação do modo claro/escuro com o período atual do dia, usamos uma função que retorna o horário atual e mantém o padrão claro das 6h até às 18h.

Toda a interface mantém-se no loop de execução da janela da Raylib até que a condição `WindowShouldClose()` seja dada como verdadeira, isso é atingido pela tecla ESCAPE ou pelo encerramento da janela manualmente.

A classe `InterfaceGrafica` concentra a maior parte dos componentes visuais da aplicação, aspectos como a textura do fundo claro/escuro são carregados nesta classe através da função `Jaylib.LoadTexture()` e a classe `Texture2D`, outros componentes importantes como slider, modo escuro, teclado e botões para cada instrumento também encontram-se dentro da classe. A classe invoca os métodos de desenho de seus componentes que renderizam na tela.

O slider de volume é composto basicamente por texturas do fundo e do indicador que correspondem a cada modo escuro/claro, o componente basicamente responde às teclas direcionais, que fazem o indicador subir ou descer, para aumentar ou diminuir o volume, usando a função `alterouVolume()` e `Jaylib.SetMasterVolume()` para controlar o volume e `Jaylib.DrawTextEx()` para o desenho das texturas.

O botão modo escuro reage ao clique do mouse e troca as texturas carregadas na função `Jaylib.DrawTextEx()` com base no seu estado atual e horário do início da aplicação.

O Teclado serve como display das notas sendo tocadas e responde de acordo com a distribuição das notas dada pelo instrumento que está sendo tocado no momento.

3. Teste

Os testes foram realizados utilizando o JUnit. A maior parte da implementação do programa é feita por meio de funções de execução e recepção de entradas de periféricos. Tendo isso em vista, a maioria dos testes tinha como foco verificar variáveis booleanas que indicam se as ações executadas por cada uma das funções foram concluídas com sucesso.

3.1 Bateria, Baixo, Guitarra e Piano

Os testes desses instrumentos consistem em fornecer um array de notas para serem tocadas à função "tocar" de cada instrumento e verificar se todos os booleanos que indicam se a execução foi realizada corretamente estão definidos como TRUE.

3.2 Nota

O teste realizado na classe Nota consiste em fornecer um nome de nota e instrumento, e, ao tentar tocar a nota, testa se está efetivamente sendo tocada.

3.3 Tecla

O teste da classe Tecla cria instâncias de Tecla, e testa as conversões do layout ABNT2 para o americano (utilizado pela Raylib), além de testar se o método equals() retorna verdadeiro quando o caractere relacionado é igual para dois objetos Tecla diferentes.

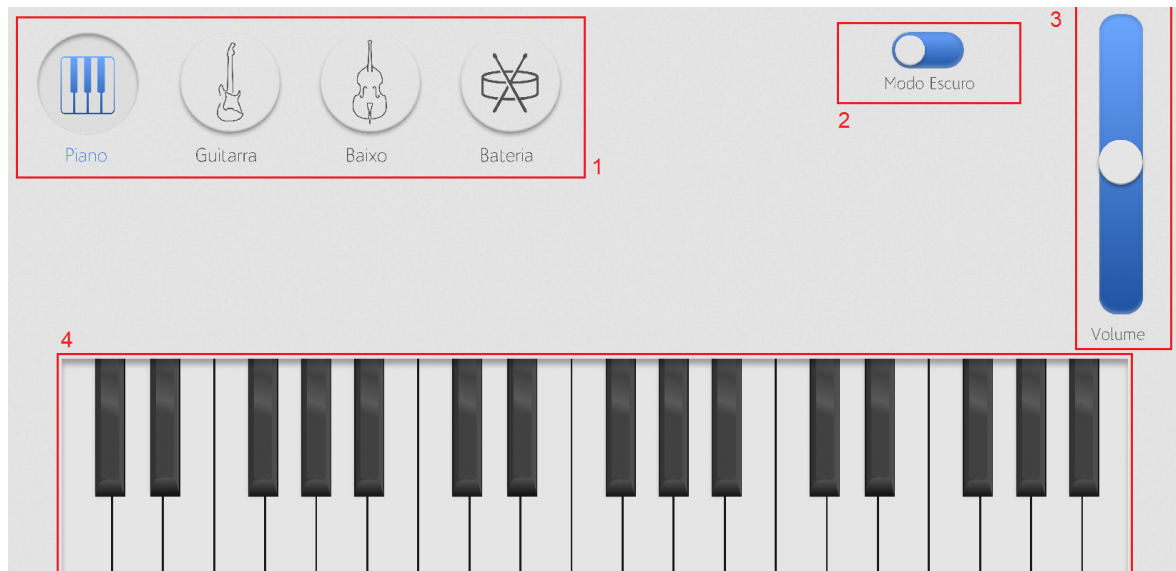
3.4 Experiência do grupo

O grupo enfrentou certa dificuldade na elaboração dos testes, principalmente devido ao entendimento limitado dos membros sobre a interação entre as funções de testes do JUnit e as funções da Jaylib.

4. Executável

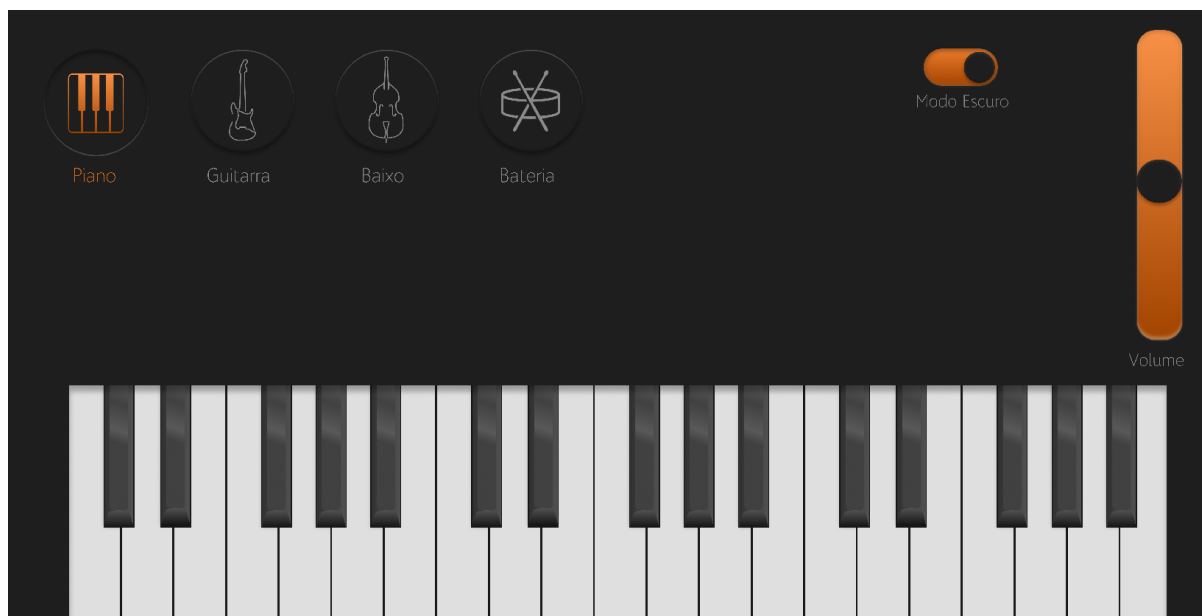
4.1. Interface

Para iniciar o aplicativo, o usuário deve clicar duas vezes, ou selecionar e pressionar <Enter> no arquivo HighUFRGSMusical.jar.



Interface no modo claro

1. Seletor dos instrumentos: Nessa seção da interface é possível escolher o instrumento que será tocado. Para isso, basta apenas clicar no ícone desejado e o programa mudará imediatamente para o instrumento selecionado.
2. Seletor do modo escuro: O seletor do modo escuro faz justamente o que o nome explicita: permite ao usuário alternar entre o modo claro e escuro, através de um clique do mouse no botão. Sua versão escura pode ser conferida em seguida.
3. Slider do volume: O volume da aplicação pode ser alterado através dos direcionais do teclado. O volume atual é mostrado por este slider no canto direito.
4. Teclado: O teclado mostra as teclas que são pressionadas, se estão relacionadas a uma nota (a classe Bateria não mostra nenhuma nota, visto que são sons sem tom).



Interface no modo escuro

A única alteração na interface em relação à etapa 1 foi o slider de volume. A ideia inicial era criar um “knob” para exibi-lo. Entretanto, devido a alguns problemas relacionados à biblioteca, essa mudança mostrou-se necessária.

4.2. Mapeamento das notas e forma de tocar

Instrumentos de corda (guitarra e baixo) são tocados segurando o botão correspondente a sua nota e apertando “ENTER”, simulando o ato de tocar as cordas do instrumento. As teclas e notas associadas no baixo e na guitarra podem ser conferidas nas tabelas abaixo:

Guitarra					
a	E2	w	D#3	4	D4
s	F2	e	E3	5	D#4
d	F#2	r	F3	6	E4
f	G2	t	F#3	7	F4
g	G#2	y	G3	8	F#4
h	A2	u	G#3	9	G4
j	A#2	i	A3	0	G#4
k	B2	o	A#3	-	A4
l	C3	1	B3	=	A#4

Baixo					
a	E1	w	D#2	4	D3
s	F1	e	E2	5	D#3
d	F#1	r	F2	6	E3
f	G1	t	F#2	7	F3
g	G#1	y	G2	8	F#3
h	A1	u	G#2	9	G3
j	A#1	i	A2	0	G#3
k	B1	o	A#2	-	A3
l	C2	1	B2	=	A#3

ç	C#3	2	C4	[B4
q	D3	3	C#4		

ç	C#2	2	C3	[B3
q	D2	3	C#3		

O piano foi pensado de maneira a simular da melhor forma um piano real. Para tocá-lo, é necessário apenas apertar a tecla correspondente a sua nota, como segue:

Piano									
q	C3	o	D4	b	E5	5	F#3	f	C#5
w	D3	p	E4	n	F5	6	G#3	g	D#5
e	E3	'	F4	m	G5	7	A#3	j	F#5
r	F3	\	G4	,	A5	9	C#4	k	G#5
t	G3	z	A4	.	B5	0	D#4	l	A#5
y	A3	x	B4	;	C5	=	F#4		
u	B3	c	C5	2	C#3	a	G#4		
i	C4	v	D5	3	D#3	s	A#4		

Para tocar a bateria é necessário apenas pressionar as teclas correspondentes. A diferença com o piano é que o som de cada tecla continua até terminar, enquanto que no piano o som é encerrado quando a tecla deixa de ser pressionada.

Bateria					
v	Caixa	u	Tom	4	Hihat aberto
espaço	Bumbo	y	Prato de ataque	8	Prato de condução
t	Tom	3	Hihat fechado	9	Surdo

O usuário pode ainda alterar o mapeamento das teclas e notas. Para isso, basta ir para o arquivo *mapeamento<instrumento>.txt*, na pasta *assets*, e alterar as linhas. O funcionamento detalhado de cada instrumento pode ser lido na seção 2.2.