

Ex. 5 (Outros sons)

Na função `tone()` em `PlayThatTuneDeluxe.java`, usamos uma senoide para criar o vetor `a[]`. Podemos usar algoritmos diferentes para criar sons com "timbres" diferentes. Por exemplo, podemos usar o algoritmo de Karplus e Strong, descrito brevemente a seguir.

Suponha que queremos gerar um som com tonalidade (frequência) `hz` e duração `d`. Crie um vetor `a[]` com capacidade para `N + 1` doubles, onde `N` é a parte inteira de $(44100 * d)$. Seja `p` a parte inteira de $44100/hz$. Preencha as primeiras `p` posições de `a[]` com valores aleatórios sorteados no intervalo `[-1, 1]`. Para preencher as posições `p` até `N` do vetor `a[]`, adote o seguinte algoritmo:

```
for (int i = p, j = 0; i <= N ; i++, j++)  
    a[i] = .997*(a[j] + a[j+1])/2;
```

O vetor `a[]` produzido dessa forma contém o som que podemos agora enviar para a saída de áudio usando `StdAudio.play(a)`. Esse som terá a tonalidade dada por `hz` e duração dada por `d`.

Uma outra coisa interessante é que podemos alterar o modo como `a[i]` é calculado para que o seu som fique diferente! Você pode testar com:

```
a[i] = .997*(19*a[j] + a[j+1])/20;
```

ou

```
a[i] = .997*(a[j] + a[j+3])/2;
```

ou

```
a[i] = .997*(9*a[j] + a[j+5])/10;
```

ou

```
a[i] = .997*(8*a[j] + a[j+2] + a[j+8])/10;
```

etc.

Nesse EP, vocês terão que criar uma versão melhorada de `PlayThatTuneDeluxe.java` (cujo nome será **`PlayThatChordDeluxe.java`**) que utiliza o algoritmo de Karplus–Strong ao invés do método da senoide! Para isso, escreva uma função de assinatura

```
public static double[] wave(double hz, double d)
```

que devolve o vetor `a[]` obtido pelo processo do algoritmo Karplus–Strong e use-a no lugar do `tone()`, do `PlayThatTuneDeluxe`.

Seu **PlayThatChordDeluxe.java** deve receber toda a entrada na entrada padrão. Cada linha de entrada (não vazia) deverá ser constituída de um certo número (positivo) de inteiros e um real. Os inteiros indicam a tonalidade das notas a serem tocadas simultaneamente. O real indica a duração dessas notas.

Por exemplo, os arquivos `elise.txt` ou `looney.txt` poderiam ser dados como entrada para seu programa `PlayThatChordDeluxe.java`. Nesses casos, apenas uma nota seria tocada em cada momento (misturadas com pausas, ou seja, momentos de silêncio). Isto é, o comportamento de `PlayThatChordDeluxe.java` seria igual ao comportamento de `PlayThatTuneDeluxe.java` nesses casos.

Por outro lado, seu `PlayThatChordDeluxe.java` deve também aceitar entradas como **Ascale2.txt** dado abaixo. Nesse caso, várias notas são tocadas simultaneamente.

Além disso, para o timbre do som ficar melhor, troque, **obrigatoriamente**, a linha

```
double hz = 440 * Math.pow(2, pitch / 12.0);
```

do `PlayThatChord` por

```
double hz = 220 * Math.pow(2, pitch / 12.0);
```

Experimente executar o `PlayThatChordDeluxe` usando como entrada os vários exemplos de S&W.

Seu programa também deve receber um argumento **OPCIONAL** da linha de comando. Se não houver esse argumento (ou seja, `if(args.length < 1)`), execute o programa usando o método normal de calcular `a[i]` (que é $a[i] = .997 * (a[j] + a[j + 1])/2$). Se ele estiver presente (`if(args.length >= 1)`), você deve executar seu programa usando um outro método de calcular `a[i]`, de sua escolha (experimente testar com vários e coloque aquele que você mais gostou piscando)

Faça, também, um relatório (cujo nome deve ser **relatorio.<ext>** onde `<ext>` pode ser `pdf` ou `txt`) contando seus experimentos com os métodos diferentes de calcular `a[i]` e sobre outras coisas que você julgar interessante!

NÃO SE ESQUEÇA: Garanta que seu programa compile com `javac`introduc `PlayThatChordDeluxe.java` e que ele execute com `javaintroc`s `PlayThatChordDeluxe`.

Sugestão. Para ler uma linha da entrada padrão, use `StdIn.readLine()`. Se `s` é do tipo `String`, faça

```
s = s.trim();
```

para jogar fora os espaços no começo e no fim de `s`. Faça

```
String[] t = s.split("\\s+");
```

para "decompor `s`" nos strings sem espaço que compõe `s` (e são separados por espaços).

Observação. Você pode achar interessante visitar

https://en.wikipedia.org/wiki/Karplus%E2%80%93Strong_string_synthesis

<http://introcs.cs.princeton.edu/java/assignments/guitar.html>

<http://introcs.cs.princeton.edu/java/21function/PlayThatTuneDeluxe.java.html>