10 PLAY "A4B4P5"

RUN:LIST 40-50 :ABCP

MSX MÚSICA

JOSÉ MAURÍCIO O. BUSSAB

30 40 50 GOTO

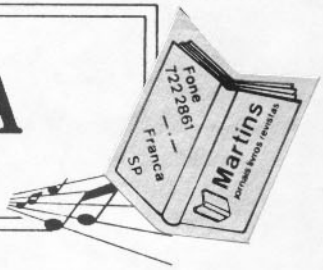
INTERVAL ON:NOTA

PRINT "TESTANDO 1,2,3,..."

SOUND INTERVAL

McGraw-Hill

MSX-MÚSICA



MSX - MÚSICA

José Maurício de Oliveira Bussab

McGraw-Hill
São Paulo
Rua Tabapuã, 1.105, Itaim Bibi
CEP 04533
(011) 881-8604 e (011) 881-8528

Rio de Janeiro • *Lisboa* • *Porto* • *Bogotá* • *Buenos Aires* • *Guatemala* • *Madrid* • *México* • *New York* • *Panamá* • *San Juan* • *Santiago*

Auckland • Hamburg • Kuala Lumpur • London • Milan • Montreal • New Delhi • Paris • Singapore • Sydney • Tokyo • Toronto

Editor: Milton Mira de Assumpção Filho
Coordenadora de Revisão: Daisy Pereira Daniel
Supervisor de Produção: Neder Roberto O. Campos
Capa: Informática Graphyco Visual
R. Dr. Emílio Ribas, 136 – São Paulo

Dados de Catalogação na Publicação (CIP) Internacional
(Câmara Brasileira do Livro, SP, Brasil)

Bussab, José Maurício de Oliveira.
B986m MSX-música / José Maurício de Oliveira Bussab. -- São Paulo: McGraw-Hill, 1987.

1. MSX (Computadores) – Programação 2. Música por computador I. Título

87-1509

CDD-781.610285
-001.642

Índices para catálogo sistemático:

1. MSX: Computadores: Programação: Processamento de dados 001.642
2. Música por computador 781.610285

SUMÁRIO

INTRODUÇÃO

O QUE É E A QUEM SE DESTINA ESTE LIVRO?

O assassino é o mordomo.

A melhor maneira de estragar um livro de mistério é começá-lo pelo final.

Como este não é um livro de mistério...

No final o leitor verá que ele é uma introdução ao uso do MSX na área de música, com ênfase maior na produção de som.

Pronto. Já contamos o final.

O leitor não pode reclamar que foi despistado e que pensava que o assassino fosse o maquinista do trem.

E quem é esse leitor?

O livro foi escrito para os programadores BASIC-MSX que têm interesse em música, em física da música, ou para músicos amadores e profissionais que já programam em BASIC-MSX e querem ver aplicações que unam as duas áreas de interesse.

Um conhecimento de BASIC elementar é bem-vindo.

A razão disso é muito simples: existem muitos outros livros de introdução ao BASIC-MSX. Seria pouco eficiente repetir aqui a informação que pode ser encontrada tão facilmente em outro lugar. Preferimos concentrar nossos esforços nas áreas diretamente pertinentes ao tema do livro e pressupor que o leitor já esteja familiarizado com os

comandos mais elementares do BASIC (FOR-NEXT, GOSUB, READ-DATA etc.), sua função e como utilizá-los na confecção de programas simples.

Mesmo assim tentamos apresentar os programas da forma mais clara possível, usando o menor número possível de atalhos e comandos obscuros do BASIC. O importante é que os métodos e algoritmos utilizados fiquem claros para o leitor.

As pistas devem levar o leitor ao elemento do crime e não despistá-lo. Neste livro fizemos o possível para que o suspeito fosse sempre o criminoso.

Quem já conhece um pouco de música e usa os comandos PLAY e SOUND, mas gostaria de ver aplicações que não fossem efeitos sonoros e pequenas melodias, pode pular direto para o Capítulo 4.

Quem está interessado só na física da música e suas aplicações no MSX (freqüência, timbre, envoltória...) pode pular para o Capítulo 3.

POR QUE O MSX?

Por que escolhemos o MSX como equipamento para trabalharmos?

Porque ele é uma máquina relativamente barata e cômoda para testes, idéias e experimentos em música ligada à Computação.

A primeira vez que tive um MSX na minha frente foi numa tarde em 1985. Abri o manual, liguei a máquina e dois minutos depois eu estava freneticamente tentando fazer com que ele produzisse sons de bateria eletrônica.

Consegui fazê-lo após uma hora. As funções de som no MSX são de fácil acesso e razoavelmente completas. Sabendo o que se quer e como, é fácil fazer.

O resultado dessa e de muitas outras tardes (e noites) veio junto com o lançamento do EXPERT: o cartucho "TOQUE!" e a sonorização de diversos outros programas.

O material presente neste livro pretende ser suficiente para que você faça, pelo menos, seu próprio programa de simulação de instrumento.

Resolvemos incluir os projetos de composição por computador e composição assistida por computador (seção 8.3) principalmente para desmistificar esse tema. Na região onde a computação e a música se tocam nenhuma polêmica é tão grande quanto a

que ocorre nas áreas que se relacionam com a criação artística. A discussão aí tende a continuar esquentando, enquanto nas outras áreas como produção de som, edição de partituras e até mesmo na educação musical assistida por computador ela já esfria.

Os fractals musicais (seção 8.3.2) são um caso à parte. A idéia de usar a teoria dos fractals em música e não em arte visual é bem nova e promete dar muito o que falar. O MSX se torna particularmente interessante por ter três canais de som, o que permite tocar os fractals de primeira, segunda e terceira ordem ao mesmo tempo.

A palavra de ordem é **experimentar**. Todo o material aqui apresentado é apenas um ponto de partida. O assunto não está encerrado com o final do livro.

O assassino é o mordomo, que fugiu.

Agradecemos de todo o coração a Dona Leila, Sr. Zé Hugo, Dr. Santiago, Dona Lady e Srta. Herta pelo apoio e paciência.

CAPÍTULO 1

SOM, MÚSICA E O COMANDO PLAY

1.1. O SOM E A MÚSICA

Vivemos em um mundo de sons. Estamos constantemente sendo bombardeados por uma avalanche de sons misturados: vozes, músicas em elevadores, buzinas, mugidos de gado, o mar quebrando nas pedras...

Dentro deste universo de sons em que vivemos, alguns conjuntos de sons são propositadamente organizados de forma artística ou como diversão. Esses conjuntos muito particulares e organizados de sons foram por nós batizados de "música". O indivíduo que organiza conjuntos de sons é chamado de "compositor" ou, quando ele trabalha ou modifica um organização de sons já existente, de "arranjador". O indivíduo que produz sons de acordo com a organização estabelecida pelo compositor é o "executante", o "cantor", o "instrumentista" etc...

O conjunto de sons de um engarrafamento de trânsito não é, a princípio, música. Isso simplesmente porque ninguém (nenhum compositor) organizou os sons desse engarrafamento em particular segundo qualquer forma artística de organização e disse "isto é música".

Isso poderia ser feito. Um compositor poderia preparar uma partitura para buzinas, guinchos de pneus e ruídos de motor. Isto seria MÚSICA. Outro compositor poderia gravar o ruído de um congestionamento e depois trabalhar sobre a fita gravada, remontando pedaços, remixando, até criar uma forma organizada de sons que ele pudesse chamar de música.

Mesmo que um compositor simplesmente gravasse um congestionamento, deixasse o conjunto de sons intocado e dissesse "isto é música", teríamos de concordar

com ele. Ele escolheu uma determinada forma de organização. Voltaremos a esse ponto mais tarde.

A idéia de organizar sons normalmente chamados de "barulho" em formas musicais não é nova. Muitos compositores já organizaram sons de vídeo-game, campainhas, máquinas de escrever, bigornas e até mesmo o completo silêncio em formas musicais.

A fala humana não-intencionalmente musical, como o discurso de um político, foi objeto de reorganização por diversos compositores.

Por outro lado, uma goteira pingando sobre as teclas de um piano não produz necessariamente música. Não importa que os sons estejam sendo produzidos por um instrumento musical. A intenção não é a de produzir música. Agora, se um compositor dadaísta espera um dia de chuva, vai até um teatro, coloca um piano sobre o palco, faz um buraco no teto deste teatro, exatamente sobre as teclas do piano, e apresenta os sons gerados pelas gotas caindo sobre as teclas como sendo música, aí sim, isto é música.

Composição é somente aquilo que resulta de um trabalho de organização. Os sons de um engarrafamento podem ser música e os sons de um piano podem ser ruído.

O ponto que se faz importante aqui é que não existem sons naturalmente musicais e sons não-musicais.

Todo som pode ser musical ou não-musical.

Todo o som deve ser tratado, estudado e classificado com a mesma hierarquia. Devemos estudar o som de uma flauta e o som de um canhão de igual maneira, com igual reverência e com os mesmos pesos e medidas. Ambos são sons de igual calibre e podem eventualmente ser usados como elementos de uma composição musical. Basta escutar a "1812" de Tchaikovsky para verificar a veracidade dessa afirmação.

Vamos definir então o primeiro conceito necessário para o estudo do som: o conceito de **evento sonoro.**

1.2. EVENTO SONORO

Consideraremos **evento sonoro** como sendo a menor unidade possível de som.

Como exemplos de eventos sonoros temos: uma nota tocada por um piano, uma vogal falada, uma granada explodindo, um prato de bateria sendo percutido etc.

Todo evento sonoro é classificado da mesma forma.

> Todo evento sonoro é composto pelas mesmas quatro características ou "parâmetros": duração, afinação, intensidade e timbre.

A **duração** é o tempo que o evento sonoro dura.

A **afinação** é a nota que efetivamente está sendo tocada. Um Dó, por exemplo, tem uma afinação diferente de um Ré. Alguns eventos sonoros são de afinação pouco definida, como, por exemplo, o prato de uma bateria ou a explosão de uma granada.

A **intensidade** é o volume do som. Uma nota tranqüila de uma flauta doce tem uma intensidade menor que o som de um jato 747 decolando.

O **timbre** é a característica mais difícil de ser definida. Pode-se dizer que o timbre é a "cor" do som. Uma saxofone tocando um Dó com uma certa intensidade durante um certo tempo, e uma flauta tocando o mesmo Dó com a mesma intensidade durante o mesmo tempo soam diferentes. Ambos são eventos sonoros diferentes em timbre, apenas. Costuma-se qualificar o timbre de um instrumento com adjetivos arbitrários. Pode-se dizer, por exemplo, que um saxofone tem um som áspero e uma flauta tem um som suave ou doce.

Qualquer evento sonoro é composto desses quatro e apenas desses quatro parâmetros seja ele um som considerado musical ou não. Essas quatro características independem umas das outras. Elas não estão ligadas de maneira alguma. Por exemplo, um som de alta intensidade não tem necessariamente uma alta afinação e vice-versa.

O que é uma partitura musical?

Uma partitura musical é simplesmente um registro ordenado de eventos sonoros. É um documento que permite ao instrumentista reproduzir os eventos sonoros especificados pelo compositor ou arranjador, numa ordem estabelecida.

> Uma partitura é um sistema de notação de eventos sonoros.

O BASIC do MSX também tem seu sistema de notação de eventos sonoros. Esse sistema tinha de existir, ou não seria possível "explicar" ao micro que eventos sonoros produzir e em que ordem.

Ora, se um evento sonoro é determinado por quatro parâmetros, qualquer sistema de notação de eventos sonoros tem de permitir uma simbologia para cada um desses parâmetros.

> Tanto uma partitura quanto o BASIC-MSX têm uma maneira de especificarmos afinação, duração, intensidade e timbre.

Vamos, a seguir, estudar os parâmetros separadamente e verificar como eles são especificados nos dois sistemas.

1.2.1. AFINAÇÃO

Na realidade, a afinação é relacionada com a freqüência da nota, mas veremos isso mais tarde, em detalhes.

Vamos ver, por enquanto, como a **música** trata a afinação.

Vamos considerar a afinação como sendo, simplesmente, o **nome** da nota que está sendo tocada. É o "Dó", "Mi" etc.

Quantas afinações diferentes existem segundo a tradição musical?

A tradição musical identifica apenas doze diferentes notas musicais: Dó, Dó#, Ré, Ré#, Mi, Fá, Fá#, Sol, Sol#, Lá, Lá# e Si. Essas doze notas, arrumadas nessa ordem crescente, são chamadas de **escala cromática.**

O símbolo # indica **sustenido.** "Dó#" é pronunciado **Dó sustenido.**

Aqui esbarramos num problema muito sério da teoria musical: "o problema das sete notas".

Você deve estar pensando: mas eu sempre pensei que fossem só sete notas: Dó, Ré, Mi, Fá, Sol, Lá e Si.

Isso não é verdade. As "sete notas musicais" nada mais são do que um conjunto particular de sete notas muito usadas na Idade Média e que receberam seus nomes naquela época. As outras cinco notas sustenidas não receberam nenhum nome especial, apenas o sobrenome de "sustenido", simplesmente por serem de pouca importância na época.

A tradição, no entanto, persiste. As cinco notas banidas ainda levam o mesmo nome da nota anterior da escala acrescida de "sustenido". Hoje em dia ninguém as considera "menos importantes" que as outras. Todas as doze notas juntas compõem uma escala musical regularmente espaçada e simétrica. Não dê importância à nomenclatura medieval: aceite-a.

Então não existem as notas Mi# e Si#? Na atual escala cromática, conhecida como **temperada** e que já está sendo usada há alguns séculos, elas têm a mesma afinação, respectivamente, das notas Fá e Dó.

Cada uma dessas cinco notas sustenidas podem ter um segundo nome. Em vez de darmos o nome da nota anterior acrescido de "sustenido", podemos dar o nome da próxima nota, acrescida de **"bemol"** (simboliza-se **"b"**). Por exemplo: Lá# (Lá sustenido) é o mesmo que Sib (Si bemol).

Na escala cromática. Dób é igual a Si e Fáb é igual a Mi.

A "distância" de afinação entre quaisquer duas notas consecutivas da escala cromática é a mesma e é chamada de **meio-tom**. Entre o Dó e o Dó# temos uma distância de meio-tom. Entre o Mi e o Fá também temos meio-tom. As distâncias entre o Dó e o Dó# e entre o Mi e o Fá soam exatamente iguais. É por isso que dissemos anteriormente que as doze notas da escala cromática formam uma escala equidistante e simétrica.

Temos na Figura 1.1, um teclado de piano com a disposição dessas notas e sua notação tradicional e de cifras.

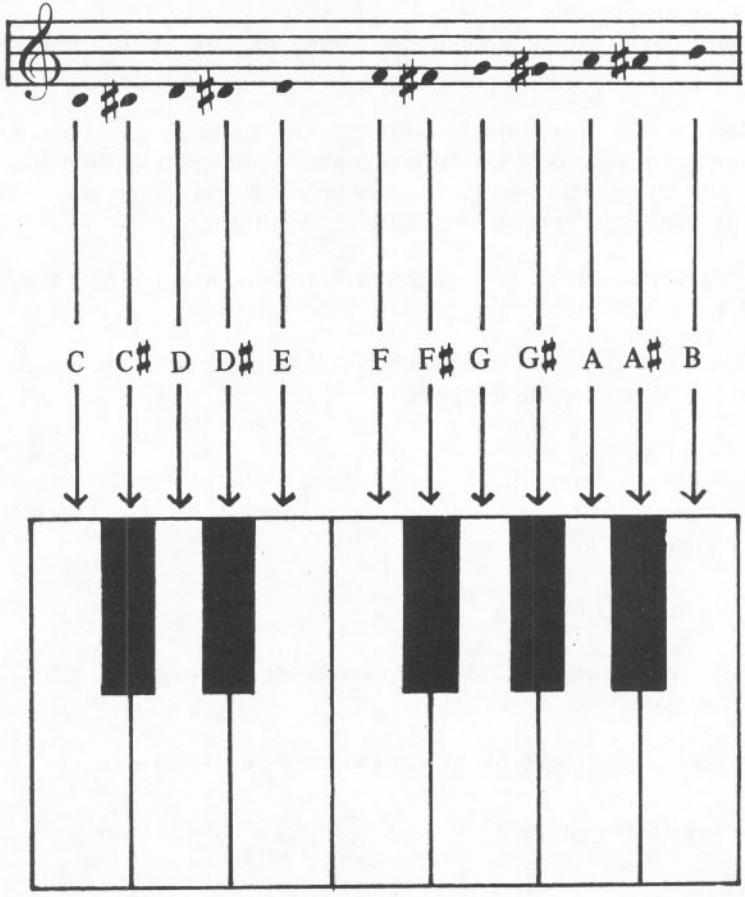


Figura 1.1 Doze notas musicais

Numa partitura tradicional, a afinação de uma nota é indicada pela altura vertical que a nota ocupa dentro do pentagrama (as cinco linhas da pauta). Temos as doze notas musicais no pentagrama da Figura 1.1. A posição de cada nota no pentagrama está indicada na figura por um ponto escuro.

A técnica de notação tradicional, no pentagrama, é bastante adequada para papel pautado e lápis e a nomenclatura usual de notas (Dó, Ré etc.) é bastante adequada para a linguagem falada. No entanto, para diversos outros fins, existe uma outra notação que é mais adequada. Estamos falando da **notação cifrada** ou **cifras.**

Nessa notação cifrada, temos um código de letras para cada nota. O código está também na Figura 1.1. Esta é a notação normalmente utilizada em música popular para uma rápida comunicação. Ela não tem os inconvenientes de representação computacional das outras duas notações, pois cada nota pode ser representada por uma simples letra, ocasionalmente seguida do símbolo # ou b.

Essa notação é a mais simples para máquinas de escrever ou, no caso, para computadores.

Existem vários programas editores de música em notação tradicional (pentagramas), inclusive para o MSX. Infelizmente, o processamento gráfico envolvido é tão grande que uma linguagem não particularmente musical, como o BASIC-MSX, não pode se dar ao luxo de conter um editor gráfico de partituras.

Para produzir notas de uma determinada afinação no MSX, você pode usar o comando **PLAY** e a notação cifrada.

Verifique que afinações você quer produzir e use as cifras correspondentes. O comando deve ser escrito da seguinte maneira:

PLAY *"cifras"*

Por exemplo: para tocar a seqüência de notas Dó, Mi, Sol, Dó, Fá, Lá, Dó você deve executar o comando:

```
PLAY "CEGCFAC"
```

Para simbolizar o sustenido você pode usar tanto o caractere "#" como "+". O bemol é simbolizado com "–".

Por exemplo, a instrução para tocar Dó, Ré#, Fá#, Lá seria:

```
PLAY "CD#F#A"
```

Ou, usando bemóis para todas as notas (você pode usar bemóis em apenas uma delas, ou várias, como quiser):

PLAY "CE-G-A"

Toda a discussão sobre o número de notas existentes na música atual não levou as oitavas em conta.

Ora, é claro que a escala cromática não pára no Si. Depois do Si vem um novo Dó e assim por diante. Antes do Dó da escala existe um outro Si. Na verdade, a escala cromática se repete indefinidamente para cima e para baixo. Ela se espelha infinitamente por todo o espectro possível de som. Sempre existirá um Dó mais agudo ou mais grave.

Se você tem um piano, você pode tocar sempre o próximo Dó. Se você está no último Dó do piano, compre uma flauta, depois um flautim, depois um apito de ultra-som.

Para diferenciar um Dó do outro, dizemos que eles estão em **oitavas diferentes.**

Isso se aplica a qualquer nota, não somente ao Dó. Escolha uma nota qualquer. A próxima nota igual a esta, mais aguda, estará uma **oitava acima.** A próxima nota mais grave, uma **oitava abaixo.**

Similarmente, usamos expressões como "duas oitavas acima", "três abaixo" etc.

O MSX tem a capacidade de reproduzir uma gama de afinações que cobre oito oitavas. Como cada oitava contém doze notas, temos um total de 96 notas diferentes no MSX.

O MSX numera essas oitavas de 1 a 8. A oitava mais grave que o MSX pode soar leva o número 1 e a mais aguda leva o número 8.

Na Figura 1.2 podemos ver um teclado fictício de piano com todas as possíveis 96 notas do MSX, com sua notação tradicional e o número de oitavas.

Dentro de um comando PLAY, especificamos o número da oitava através da letra "O" seguida do número.

> Todas as notas que vierem após a especificação de oitava serão tocadas na oitava indicada, até nova especificação.

Se nenhuma oitava for especificada, será usada a de número 4. Todos os nossos exemplos de PLAY até agora foram tocados, portanto, nesta oitava.

Se quisermos tocar Dó Mi Sol nas oitavas 3, 4 e 5 (um arpejo de Dó Maior) podemos executar o seguinte comando PLAY:

Figura 1.2 Notas do MSX e suas oitavas

```
PLAY "O3CEGO4CEGO5CEG"
```

Por enquanto, vamos deixar a afinação de lado e passar para a notação de duração.

1.2.2. DURAÇÃO

A duração é um intervalo de tempo. É o tempo entre o início e o final do evento sonoro. Poderíamos medir esse tempo em termos de segundos. Um maestro poderia dizer ao primeiro violino: toque um Si por 4.56 segundos.

Essa não é, no entanto, a maneira pela qual os músicos representam a duração de um evento sonoro.

A duração de uma nota é representada, em uma partitura, por meio de uma convenção de sinais que já dura alguns séculos. Nesse tipo de notação usual, não se especifica a duração em termos absolutos. Os símbolos contidos em uma partitura jamais dizem para um músico: "toque uma nota tal durante tantos segundos". Uma partitura diz ao músico: "toque uma nota longa" ou "toque uma nota com duração igual a metade da duração de uma nota longa" ou "um quarto da duração" e assim por diante. Os símbolos e as durações representadas por eles estão na Figura 1.3.

Note que esta notação representa a duração **relativa entre as notas.** A partir da tabela da Figura 1.3 podemos deduzir não só as relações entre a semibreve e as outras figuras mas entre as figuras entre si. Por exemplo: qual a relação entre a duração da colcheia e a da mínima? Ora, se duas mínimas equivalem a uma semibreve e oito colcheias equivalem a uma semibreve, então quatro colcheias equivalem a uma mínima.

O que é importante é que na notação tradicional da partitura, não se exprime tempo absoluto mas tempo **relativo.** Cada figura exprime um tempo que não tem sentido isolado mas somente em conjunto com as outras. Por isso uma partitura pode ser tocada mais lenta ou mais rapidamente. Quando uma partitura é tocada em uma velocidade diferente, a **relação** entre as durações das notas não muda.

A notação de durações é conhecida habitualmente pelos músicos como notação rítmica. Uma combinação de diversas notas de diferentes durações sempre denota um ritmo ou **padrão rítmico.**

Podemos representar um padrão rítmico combinando vários símbolos de

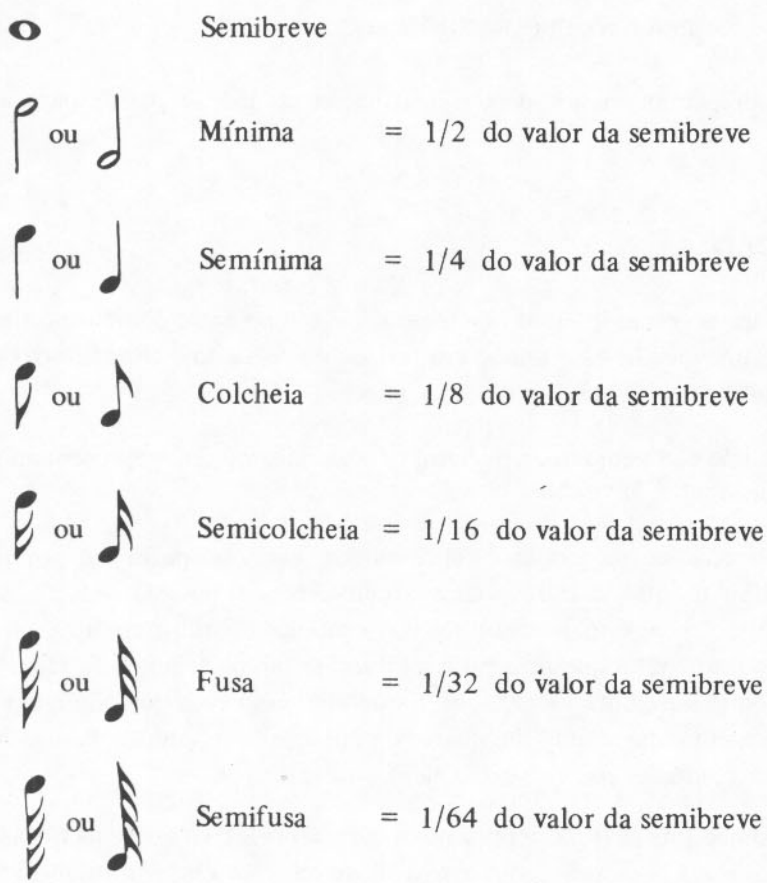


Figura 1.3 Figuras rítmicas

duração. Veja o padrão rítmico da Figura 1.4, por exemplo. Nela estão quatro figuras rítmicas: uma semibreve seguida de duas semínimas e uma mínima.

Figura 1.4 Padrão rítmico

Qual a duração que cada uma dessas quatro figuras representa?

Em termos de duração relativa à semibreve, as semínimas valem um quarto da duração desta e a mínima vale metade.

Vamos supor que a primeira figura (a semibreve) durasse um segundo. A segunda figura (a semínima) duraria um quarto de segundo, pois ela vale **sempre** um quarto do que vale a semibreve. A terceira figura também duraria um quarto de segundo. A quarta (a mínima) duraria meio segundo, pois ela **sempre** vale metade da semibreve.

Vamos supor, por outro lado, que resolvéssemos fazer a semibreve durar dois segundos. A duração das outras três figuras seria, respectivamente: meio segundo, meio segundo e um segundo.

É claro que um músico, para tocar, não fica pensando no valor das durações em termos de segundos. O que ele pode fazer é, por exemplo, bater com o pé uma marcação fixa de tempo e pensar: o "TOC-TOC-TOC" do meu pé está tocando uma porção de semínimas, uma após a outra.

Tendo uma marcação rítmica fixa no pé, ele pode bater com a mão o padrão rítmico, usando o pé (as semínimas constantes) como guia.

Vamos supor que o músico tenha de tocar uma semibreve com a mão. Ele sabe que cada semibreve tem uma duração igual à duração de quatro semínimas. Se ele está batendo com o pé uma porção de semínimas e a semibreve vale quatro semínimas, ele sabe que, para tocar uma semibreve com a mão, terá de tocar durante um tempo igual a quatro batidas do seu pé (as semínimas).

Um exemplo concreto de padrão rítmico está na Figura 1.5. Nela temos o padrão rítmico do "Marcha Soldado".

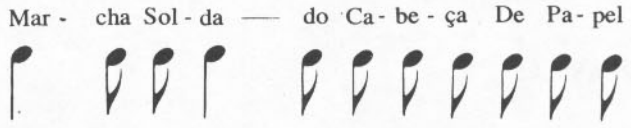


Figura 1.5 Padrão rítmico do "Marcha Soldado"

Nesse exemplo, temos uma melodia bem conhecida. Todas as figuras são colcheias, exceto duas que são semínimas. Cante a melodia em voz alta. Note que as

durações das notas correspondentes às sílabas "MARcha" e "solDAdo" valem o dobro das outras, as colcheias.

Mas como a notação tradicional de afinação se combina com esta de duração?

Muito simples. Vimos que, no pentagrama, a afinação da nota era indicada pela posição vertical que ela ocupava no pentagrama. Pois bem, em conjunto com isso, a **figura** vai indicar a **duração.**

O "Marcha Soldado" completo, em notação tradicional, com a notação de afinação, está na Figura 1.6.

Figura 1.6 Marcha completo

A notação de duração no MSX segue aproximadamente a mesma filosofia da notação tradicional. Para indicar a duração de uma determinada nota, você deve colocar, após a nota, um número que indicará a duração.

Na Figura 1.7 temos uma tabela completa dos valores e das figuras que eles representam.

Exemplo: queremos escrever um Lá com duração de semínima. Já vimos que o Lá é representado por A. A semínima é representada por 4. Para isto, devemos então executar a instrução.

```
PLAY "A4"
```

O "Marcha Soldado" no MSX fica (supondo-se oitava 3):

```
PLAY "O3G4G8E8C4C8E8G8G8G8E8D8"
```

Se não especificarmos a duração das notas, é assumida uma duração de 4 (semínima). Todos os exemplos de PLAY vistos até agora tinham esse valor.

Figura	Nº MSX
Semibreve	1
Mínima	2
Semínima	4
Colcheia	8
Semicolcheia	16
Fusa	32
Semifusa	64

Figura 1.7 Tabela das durações MSX

Existe uma maneira simples de memorizar os números de cada figura. O número que representa a figura também representa quantas figuras são necessárias para completar uma semibreve. Por exemplo: o número que representa a semicolcheia é 16 porque são necessárias 16 semicolcheias para completar uma semibreve (veja novamente a Figura 1.3).

Se, numa dada melodia que você pretende escrever no MSX, existe uma predominância de uma determinada figura de uma certa duração, utilize o seguinte recurso: coloque, antes de escrever as notas, a letra L seguida do número dessa "duração da maioria". Se você fizer isso, você não precisará especificar a duração dessas notas. Você ainda precisará especificar, é claro, a duração das notas que não pertencem a essa maioria.

No nosso "Marcha Soldado" temos uma predominância de colcheias (duração número 8). Podemos reescrever a melodia assim:

```
PLAY "O3L8G4GECC4EGGGED"
```

Note que as únicas durações especificadas foram as que não são colcheias.

Escrever a divisão rítmica de uma dada melodia na notação habitual de partituras não é uma tarefa trivial. Também não é trivial o contrário, ou seja, ler uma dada divisão rítmica numa partitura e tocá-la com precisão. Essas tarefas são chamadas, respectivamente, de "Ditado Rítmico" e "Solfejo Rítmico". Elas tomam boa parte do tempo de estudo do músico, e sua análise estaria fora do escopo do livro.

Como dissemos anteriormente, a partitura exprime a relação de durações entre as diversas notas e não as durações absolutas.

Suponhamos que haja centenas de notas em uma partitura. As durações relativas de todas elas já estão especificadas e basta que apenas UMA das durações absolutas das figuras seja especificada para que todas as outras também o sejam.

Foi isso o que fizemos acima, quando dissemos "a semibreve durará um segundo".

Numa partitura tradicional, o valor absoluto da duração de uma figura é indicado colocando-se no alto da partitura uma marcação como a da Figura 1.8.

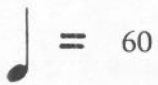


Figura 1.8 Andamento

A figura mostra uma semínima sendo igualada ao número 60. Isto significa que, nesta partitura, a semínima vale "1/60 de minuto" ou um segundo. Se o número fosse igual a 80, a semínima valeria 1/80 de minuto ou 0,75 segundos.

Esta marcação é conhecida como **marcação de tempo** ou **andamento.**

Ora, se a semínima vale um segundo, podemos deduzir quanto valem todas as outras figuras rítmicas: a semibreve valerá 4 segundos (a semínima vale sempre um quarto dela), a mínima valerá 2 segundos etc.

Na verdade, esta marcação, que aparece no alto das partituras, normalmente é usada em conjunto com um aparelho chamado **metrônomo.** Este aparelho é uma espécie de "pé automático". Ele faz uma porção de ruídos semelhantes a estalidos, igualmente espaçados. A duração do intervalo entre os estalos é regulável por um marcador. Sob o marcador existem números escritos. Se o instrumentista vai iniciar o estudo de uma peça que tem uma marcação de tempo como a apresentada na figura anterior, ele regula o metrônomo para o número correspondente à marcação da partitura. Ele sabe que as batidas do aparelho serão figuras iguais à figura que está sendo igualada ao número. No exemplo da figura, o instrumentista regularia o metrônomo para 60 e saberia que cada batida deste estaria representando uma semínima.

Se ele quisesse tocar uma semínima, bastaria ele tocar uma duração igual à batida do metrônomo. Se quisesse tocar uma mínima, tocaria uma duração igual a duas batidas do metrônomo etc.

No BASIC-MSX, a marcação de tempo se faz colocando o número do tempo que deve ser igualado a uma semínima no início da melodia, precedido pela letra T. Por exemplo, a marcação "semínima = 60" é representada pela marcação T60.

Nosso "Marcha Soldado", com uma marcação de tempo de 80, ficaria:

```
PLAY "T80O3L8G4GECC4EGGGED"
```

O BASIC-MSX aceita andamentos de 32 a 255. Se nenhum for especificado, é usado 120. Todos os exemplos até agora, portanto, tinham andamento 120.

1.2.3. TIMBRE

Como já dissemos antes, o timbre é a "cor" do som. É difícil definir o conceito de timbre sem abusar de termos técnicos mas pode-se dizer que é o timbre que dá a

identificação principal de um instrumento, que nos faz dizer "é um som de piano" ou "é um som de flauta".

Ao contrário dos outros parâmetros, o timbre está normalmente relacionado com o instrumento e não com a peça de composição em si.

Os outros indicam uma duração, uma afinação e uma intensidade que não depende do instrumento. O timbre é a assinatura do instrumento sob esses outros três parâmetros.

Cada instrumento acústico tem um timbre fixo com pequenas variações possíveis. Um saxofone, por exemplo, já tem o timbre característico do saxofone. O saxofonista pode, no entanto, fazer com que o timbre varie um pouco variando a pressão que sua boca faz sobre a palheta. Isso faz com que o timbre fique mais ou menos "estridente".

Pelo fato de cada instrumento acústico ter seu próprio timbre, as partituras tradicionais não têm uma notação sistemática para esse parâmetro, que é normalmente deixado a cargo da interpretação do instrumentista. Há exceções para essa regra, como, por exemplo, a notação de "pizzicato" para o violino e o texto de uma peça a ser cantada, que não deixam de ser, a rigor, apenas variações de timbre. Uma notação sistemática, no entanto, inexiste.

Por outro lado, os sintetizadores e outros instrumentos eletrônicos têm um bom controle desse parâmetro. E isso não deixa de ser lógico, já que nenhum sintetizador tem um "som próprio" derivado de sua constituição física, como os instrumentos acústicos têm, ficando todo o controle do som a cargo do músico.

Como veremos em detalhes mais tarde, o timbre depende de uma característica do som denominada **forma de onda**.

Nos sintetizadores, a forma de onda e, portanto, o timbre, é controlada de diversas maneiras, que variam de máquina para máquina (síntese aditiva, subtrativa, FM, PD e mais uma porção de outras que não vêm ao caso). O que importa é que em qualquer sistema eletrônico de síntese de som, o músico dispõe de controle preciso e minucioso sobre a forma de onda.

Por uma inexplicável falha, os projetistas do PSG não incluíram nenhuma maneira de se controlar a forma de onda. Ela é fixa e aproximadamente quadrada.

Existe uma maneira de se contornar ligeiramente este problema. De qualquer

modo, o timbre não pode ser controlado com facilidade e muito menos versatilidade no MSX.

Essa maneira complicada de se variar um pouco o timbre do MSX será vista bem mais tarde.

Por enquanto ficamos por aqui, lamentando esse fato.

1.2.4. INTENSIDADE

A intensidade é o volume do som. Quanto mais alto o som, maior sua intensidade.

Cada instrumento acústico tem sua própria gama de intensidades. Alguns têm uma gama de intensidade mais alta, sendo difícil fazer soar uma nota de volume baixo, como o trompete, por exemplo. Outros têm uma gama de intensidade baixa, como a harpa.

O controle preciso de intensidade nos intrumentos acústicos é uma arte. Os instrumentistas passam anos estudando a variação da intensidade de seus próprios instrumentos, pois disso depende muito a expressividade da peça musical.

Em uma partitura tradicional, a notação de intensidade é feita em termos de um conjunto de letras derivado do italiano. É habitual reconhecer os seis níveis de intensidade que estão na Figura 1.9. Na figura temos as abreviaturas e os nomes, em ordem crescente de intensidade.

pp	Pianíssimo
p	Piano
mp	Mezzo Piano
mf	Mezzo Forte
f	Forte
ff	Fortíssimo

Figura 1.9 Notação tradicional de intensidade

Note que "piano", aqui, não está indicando o instrumento piano. A palavra "piano" significa, em italiano, "suave", "tranqüilo". O instrumento conhecido por piano se chamava, originalmente, "pianoforte", exatamente por ser capaz de reproduzir tanto as intensidades "piano" quanto as intensidades "forte", característica rara nos instrumentos da época de sua invenção.

A letra correspondente à intensidade fica acima do pentagrama e ela vale até segunda ordem. A partitura de nosso "Marcha Soldado", começando "piano" e terminando "forte", está na Figura 1.10.

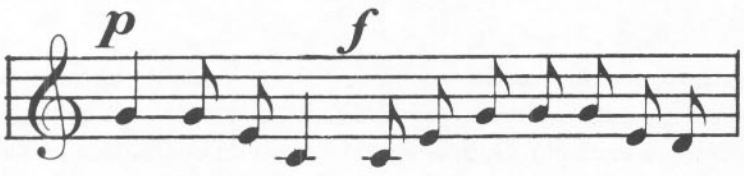


Figura 1.10 Marcha Soldado com intensidades notadas

Essa notação é bastante flexível. Em algumas partituras deste século, é possível encontrarmos ppp (*pianissíssimo*) e até um ou outro ffff (como será que se diz isso? *fortississíssimo*?), para denotar, respectivamente, um som praticamente inaudível e outro ensurdecedor. Mas essas aberrações são raras. A notação habitual está na Figura 1.10.

Essa notação habitual é bastante subjetiva e não poderia ser usada diretamente num computador.

O MSX admite intensidades (volumes) **numeradas de 0 a 15.** Para indicar uma dada intensidade é usada a letra V antes do número que designa a intensidade.

A relação entre as intensidades tradicionais e as do MSX é muito vaga. Pode-se dizer que ela é a que está presente na Figura 1.11.

Notação Tradicional		Notação MSX
pp	Pianíssimo	V1 e V2
p	Piano	V3 a V5
mp	Mezzo Piano	V6 a V8
mf	Mezzo Forte	V9 a V11
f	Forte	V12 e V13
ff	Fortíssimo	V14 e V15

Figura 1.11 Intensidades no MSX

Colocando o "piano" e o "forte" nas mesmas posições da partitura tradicional da Figura 1.10, nosso "Marcha Soldado" fica:

```
PLAY "T80V5O3L8G4GECC4V12EGGGED"
```

O V5 indica "piano" e o V12 indica "forte".

1.3. SONS SIMULTÂNEOS

Um gerador de sons sozinho não tem a menor graça. Raras são as composições, mesmo para instrumentos acústicos, que são feitas para um único instrumento melódico. A afirmação não se aplica ao piano nem ao violão, que não são melódicos, são harmônicos, pois são capazes de fazer soar várias notas ao mesmo tempo.

O compositor tem de ser muito competente para ser capaz de escrever uma peça solo interessante para um instrumento melódico. Para compensar a ausência de notas simultâneas, peças solo são escritas de modo a usar toda a capacidade de timbres e intensidades do instrumento e, geralmente, todo o virtuosismo do instrumentista.

Isto é o que ocorre na música erudita. Na música popular, a existência de peças solo é ainda mais rara. Às vezes pode-se encontrar, por exemplo, um solo de guitarra de Jimi Hendrix ou um solo de bombardino de Hermeto Pascoal, sem qualquer acompanhamento, mas esse fato é raro.

A maioria esmagadora das peças musicais é feita com dois ou mais sons simultâneos.

Na notação tradicional de partituras, duas ou mais frases musicais ocorrendo simultaneamente são representadas com dois pentagramas, um sobre o outro. A indicação de que eles são simultâneos é feita colocando-se uma barra vertical unindo estes pentagramas. Na Figura 1.12 temos uma pequena peça a três vozes em notação tradicional.

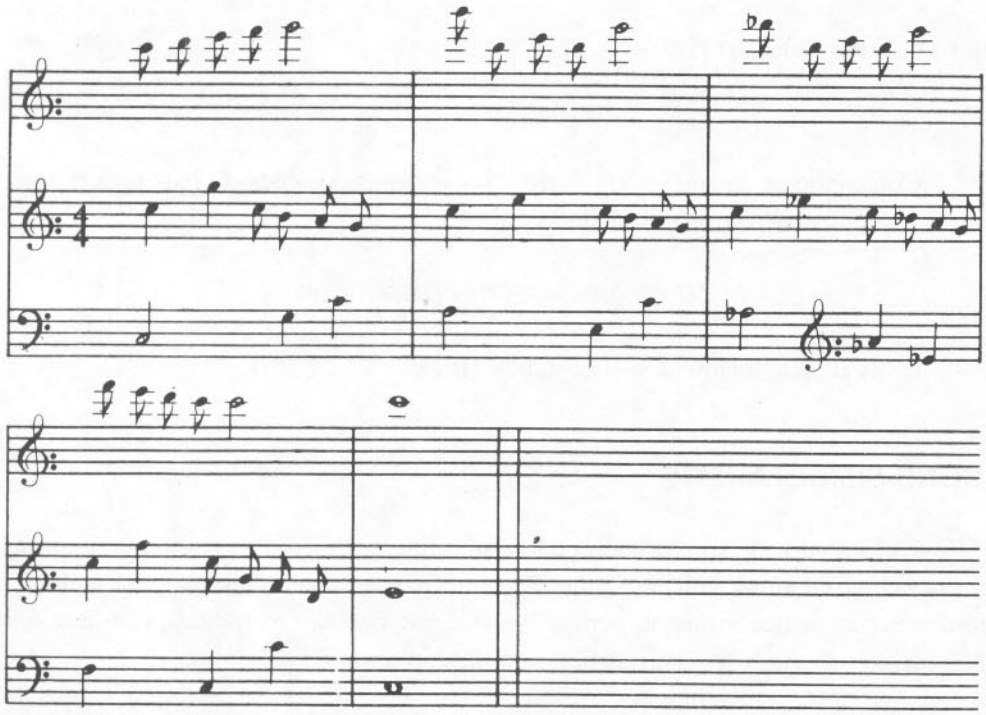


Figura 1.12 Peça musical em notação tradicional

Não se preocupe com a fração 4/4 e com as barras verticais. Elas são apenas elementos auxiliares de notação que são chamados, respectivamente, de **fórmula de compasso** e **barras de compasso.** A função da fórmula é indicar quantos tempos tem cada compasso. A barra de compasso separa um compasso do outro. Esses elementos estão sempre presentes em uma partitura, mas não são necessários no BASIC-MSX (nem existe maneira de indicá-los). Voltaremos a falar deles no nosso projeto 2 (programador de ritmos).

Até agora tratamos dos sons no MSX como se fosse possível apenas produzi-los individualmente. Se isso fosse verdade, não teríamos condições de escrever um programa para executar a peça da Figura 1.12. Na verdade, nesse computador, temos a capacidade de produzir até três eventos sonoros simultâneos.

> A existência, no MSX, de três geradores de som, é a principal razão para fazer deste computador uma boa máquina para servir de introdução à computação na música.

O MSX é um dos poucos computadores disponíveis no mercado que apresenta esse recurso fundamental. Outros computadores, como os das famílias Apple, IBM-PC e TRS-80 dispõem, em seu hardware não-modificado, de um único gerador de sons de volume fixo, tornando quase impossível a geração de mais de um som de cada vez. Quando isso é feito, nesses computadores, é usada tal quantidade de truques e artifícios que nenhum dos dois sons é ouvido de maneira clara.

Felizmente, dispomos de três geradores de som independentes no MSX.

Para gerar sons simultâneos, usando a instrução PLAY, basta escrever as duas ou três frases musicais que deverão ser tocadas simultaneamente, em notação MSX, e separá-las por vírgulas. Assim:

PLAY "*frase 1*", "*frase 2*", "*frase 3*"

Vamos ver agora como criar um programa para executar a melodia da Figura 1.12 nas três vozes.

1.4. USANDO O PLAY EM UM PROGRAMA

Inicialmente vamos usar apenas um canal e programar a melodia do alto (a primeira voz).

O programa resultante é o MELOD1.

Programa-exemplo MELOD1

```
10 PLAY "T8ØL8"
20 PLAY "O5CDEFG2"
30 PLAY "ADCDG2"
40 PLAY "A-DCDG2"
50 PLAY "FEDCC2"
60 PLAY "C1"
```

A linha melódica foi transformada em instruções PLAY e cada PLAY foi colocado em uma linha do BASIC.

Para facilitar a comparação com a partitura original, e apenas por isso, cada instrução PLAY foi escrita de forma a conter um compasso. Isso não faz a menor diferença no resultado do programa. Distribuímos as notas por seis PLAYs, mas poderíamos usar quantos quiséssemos (até mesmo um único PLAY).

Execute o programa e note como o BASIC indica que o programa terminou antes da música parar de tocar. O cursor aparece na tela com o prompt "OK" mas a música continua tocando por alguns segundos.

O BASIC tem uma "pilha de sons" que é usada em conjunto com o PLAY. OS comandos vão sendo guardados e executados à medida que vão sendo necessários. Isso libera o BASIC para passar para a linha seguinte à do PLAY, mesmo sem esse comando ter sido terminado.

Completemos agora as duas vozes que faltam. O resultado é o programa MELOD2.

Programa-exemplo MELOD2

```
400 PLAY "T8ØL8","T8ØL8","T8ØL8"
410 PLAY "O5CDEFG2","O4C4G4C03BAG","O2C2G403C4"
420 PLAY "ADCDG2","O4C4E4C03BAG","O2A2E403C4"
430 PLAY "A-DCDG2","O4C4E-4C03B-A-G","O2A-203A-4E-4"
440 PLAY "FEDCC2","O4C4F4C03GFD","O2F2C403C4"
450 PLAY "C1","O3E1","O2C1"
```

Cada instrução PLAY contém, agora, as três vozes de cada compasso.

Você continua podendo colocar quantas notas quiser no comando PLAY. Agora, no entanto, há um cuidado extra a tomar:

> Se uma voz de um comando PLAY tiver uma duração total menor do que as outras, ela fica em silêncio do final de sua melodia até o final das outras.

O BASIC não "adianta" notas do próximo PLAY para completar a voz.

Por isso, **confira muito bem a soma das durações das três vozes.** É fácil omitir uma nota por engano.

A terceira versão do programa (MELOD3) mostra como o PLAY pode ser usado em conjunto com outros comandos do BASIC. Os PLAYs foram substituídos por um único PLAY (linha 120), dentro de um loop FOR-NEXT. Os comandos para os PLAYs estão dentro dos DATAs e são lidos pela instrução READ (linha 110).

Programa-exemplo MELOD3

```
100 FOR N=1 TO 6
110  READ A$,B$,C$
120   PLAY A$,B$,C$
130 NEXT N
140 END
300 DATA "T80L8","T80L8","T80L8"
310 DATA "05CDEFG2","04C4G4C03BAG","02C2G403C4"
320 DATA "ADCDG2","04C4E4C03BAG","02A2E403C4"
330 DATA "A-DCDG2","04C4E-4C03B-A-G","02A-203A-4E-4"
340 DATA "FEDCC2","04C4F4C03GFD","02F2C403C4"
350 DATA "C1","03E1","02C1"
```

CAPÍTULO 2

MAIS COMANDOS PARA O PLAY

Além dos comandos apresentados até agora, a instrução PLAY oferece mais para o controle dos parâmetros dos eventos sonoros. Vamos analisar os comandos que faltam.

2.1. AFINAÇÃO: MAIS COMANDOS

Além de especificarmos a afinação com a notação de cifras, podemos especificar a nota por seu "número de série".

Como já vimos, o MSX é capaz de tocar as doze notas em oito oitavas diferentes. O total de notas que podem ser tocadas é 8 vezes 12 : 96 notas.

Essas noventa e seis notas podem ser identificadas também por um número de série que varia de 0 (indicando o Dó mais grave, oitava 1) até 95 (o Si mais agudo, oitava 8).

Para fazer soar a nota que tem o número desejado, você deve colocar a letra N antes do número.

Exemplo: queremos tocar a nota Lá # na oitava 4. Esta nota pode ser tocada a partir da intrução PLAY:

```
PLAY "O4A#"

ou

PLAY "N46"
```

Se você quiser saber o número de série de uma nota basta usar a Figura 1.2 ou calculá-lo com a seguinte fórmula:

N = 12 x (Oitava – 1) + Nota

"Oitava" é o número da oitava e "Nota" é o número da nota segundo a seguinte tabela:

C = 0	D# = 3	F# = 6	A = 9
C# = 1	E = 4	G = 7	A# = 10
D = 2	F = 5	G# = 8	B = 11

Pode parecer confuso usar o número da nota em vez de uma indicação de oitava e afinação mas isso é muito útil no caso de programas que calculam como, por exemplo, um compositor automático.

Nesse tipo de programa, as notas são calculadas pelo programa e o resultado é, usualmente, um número. É mais fácil usar a instrução PLAY com o número de série diretamente do que transformar o número em nota e oitava e usar a notação de cifras.

Voltaremos a esse comando quando falarmos em composição automática.

2.2. DURAÇÃO: MAIS COMANDOS

Se, após a duração de uma determinada nota, colocarmos um ponto (.), estaremos fazendo com que a duração seja aumentada em metade.

Por exemplo, temos um Lá com a duração de semínima (A4). Se colocarmos um ponto após o "4" (A4.), estaremos aumentando a duração da semínima. O aumento é igual à metade da duração da semínima ou: uma colcheia (duração 8). Isso é chamado de **pontuar uma figura**. A semínima, seguida do ponto, é chamada de **semínima pontuada**. Como já vimos, a semínima comum tem a duração de um quarto de semibreve e a colcheia tem a duração de um oitavo de semibreve. A semínima pontuada tem a duração igual à soma dessas duas, ou três oitavos de semibreve (1/4 + 1/8 = 3/8).

Para pontuar uma figura na notação tradicional, o procedimento é exatamente o mesmo: após a figura, colocamos um ponto, como a nota da Figura 2.1.

Figura 2.1 Semicolcheia pontuada em notação tradicional

Na figura temos uma semicolcheia pontuada, que tem a mesma duração de uma semicolcheia e uma fusa somadas.

A notação de duração não inclui somente a duração dos sons, ela inclui também a duração dos silêncios ou **pausas**.

Freqüentemente é necessário indicar quanto tempo um certo instrumento deve ficar em silêncio ou qual a duração de uma certa pausa entre as notas que um determinado instrumento deve tocar.

O mesmo princípio de notação de duração de notas se aplica aqui. Na notação tradicional, a cada figura rítmica corresponde uma figura de pausa. Veja a Figura 2.2.

Se encontramos uma pausa em um partitura, no meio de outras notas, isso indica um tempo que deve ser esperado, em silêncio.

A duração desse tempo é indicada pela figura. Uma pausa de semínima indica que o tempo a ser esperado é o tempo correspondente ao tempo de execução de uma semínima.

Na instrução PLAY, o comando para a pausa é o comando R. O comando deve ser seguido do número que indica a figura que tem a duração daquela pausa.

Uma pausa de semínima, por exemplo, é indicada, no MSX, por R4.

Nosso velho "Marcha Soldado" poderia ter duas pausas de colcheia no meio dele:

```
PLAY "T80V5O3L8GR8GECR8CV12EGGGED"
```

Ache essas pausas tanto na instrução acima como ouvindo a música.

Figura	Pausa Correspondente	Nº MSX
Semibreve		1
Mínima		2
Semínima		4
Colcheia		8
Semicolcheia		16
Fusa		32
Semifusa		64

Figura 2.2 Pausas na notação tradicional.

2.3. INTENSIDADE: MAIS COMANDOS

Na realidade, existem apenas mais dois comandos ainda não mencionados: os comandos S e M. O comando S, seguido de um número de 0 a 15, controla a forma da envoltória, e o comando M, seguido de um número de 0 a 65535, controla o seu período.

Ambos são responsáveis pela variação da intensidade com o tempo (Envoltória, Capítulo 6). Esse é um tópico bem complexo e mais à frente há uma seção inteira dedicada a ele.

2.4. COLOCANDO VARIÁVEIS DO BASIC DENTRO DO PLAY

Dentro do PLAY podemos substituir pedaços de comandos ou comandos inteiros pelos valores presentes em variáveis. Essa substituição pode ser numérica ou alfanumérica.

Vejamos primeiro a substituição numérica.

> Qualquer número dentro de um PLAY pode ser substituído por um nome de variável do BASIC.

Isso é feito substituindo-se o número por um sinal de igual, seguido do nome da variável e de um ponto-e-vírgula.

Vamos supor que temos uma variável AA do BASIC que vale 58. As duas instruções a seguir são equivalentes:

```
PLAY "N58"
```

ou

```
PLAY "N=AA;"
```

Ambas as instruções acima têm o mesmo efeito: fazem com que a nota número 58 (lá# na oitava 5) seja tocada.

Isso funcionaria para qualquer lugar em qualquer comando onde um número pudesse ser escrito e não somente para o comando N.

> Lembre-se sempre do ponto-e-vírgula após o nome da variável.

Outra maneira de usar as variáveis do BASIC dentro do PLAY é com a substituição alfanumérica. Ela é feita com o auxílio do comando X.

> O comando X faz com que os comandos presentes dentro de uma variável string sejam lidos.

Variáveis string, no BASIC, são aquelas que são seguidas de um cifrão ($) e que

podem guardar valores alfanuméricos. Se uma variável dessas contiver um conjunto de comandos de PLAY, basta colocar, dentro do PLAY, a letra X, seguida de um sinal de igual (=), o nome da variável e um ponto-e-vírgula (;), para que os comandos presentes na variável string sejam executados.

Por exemplo, no pequeno programa abaixo, temos uma variável string, C$, que contém o string "FAC". O comando X, dentro do PLAY, contém uma referência a essa variável:

```
10 C$="FAC"
20 PLAY"CEGX=C$;GBDC"
```

O PLAY desse pequeno programa tem o mesmo efeito de:

```
PLAY"GEGFACGBDC"
```

Esse uso de variáveis com conjuntos de comandos pode economizar tempo e espaço. A variável contém uma espécie de "sub-rotina musical". O mesmo conjunto de comandos, que pode ser bastante grande, pode ser reutilizado várias vezes, em diversos PLAYs diferentes.

A instrução abaixo também tem o mesmo efeito:

```
PLAY"CEG"+C$+"GBDC"
```

A diferença está na velocidade. Essa última forma exige que a operação de concatenação (a operação de "soma" de strings) seja realizada antes da execução das notas, consumindo tempo. Na maioria dos casos, no entanto, a diferença entre um formato e outro é muito pequena. Esse atraso é raramente perceptível.

2.5. PROGRAMANDO DOIS CÂNONES

Vamos programar mais duas peças musicais. A forma musical das duas é a mesma: ambas são cânones de oito frases. Na Figura 2.3 temos a primeira voz de cada um dos cânones.

Um cânone é um bom exempo de uso imprescindível de melodias simultâneas. Um cânone bem conhecido é o Frère Jacques. Uma das vozes começa a cantar sozinha a melodia. Um pouco depois, outra voz começa a cantar a mesma melodia um pouco

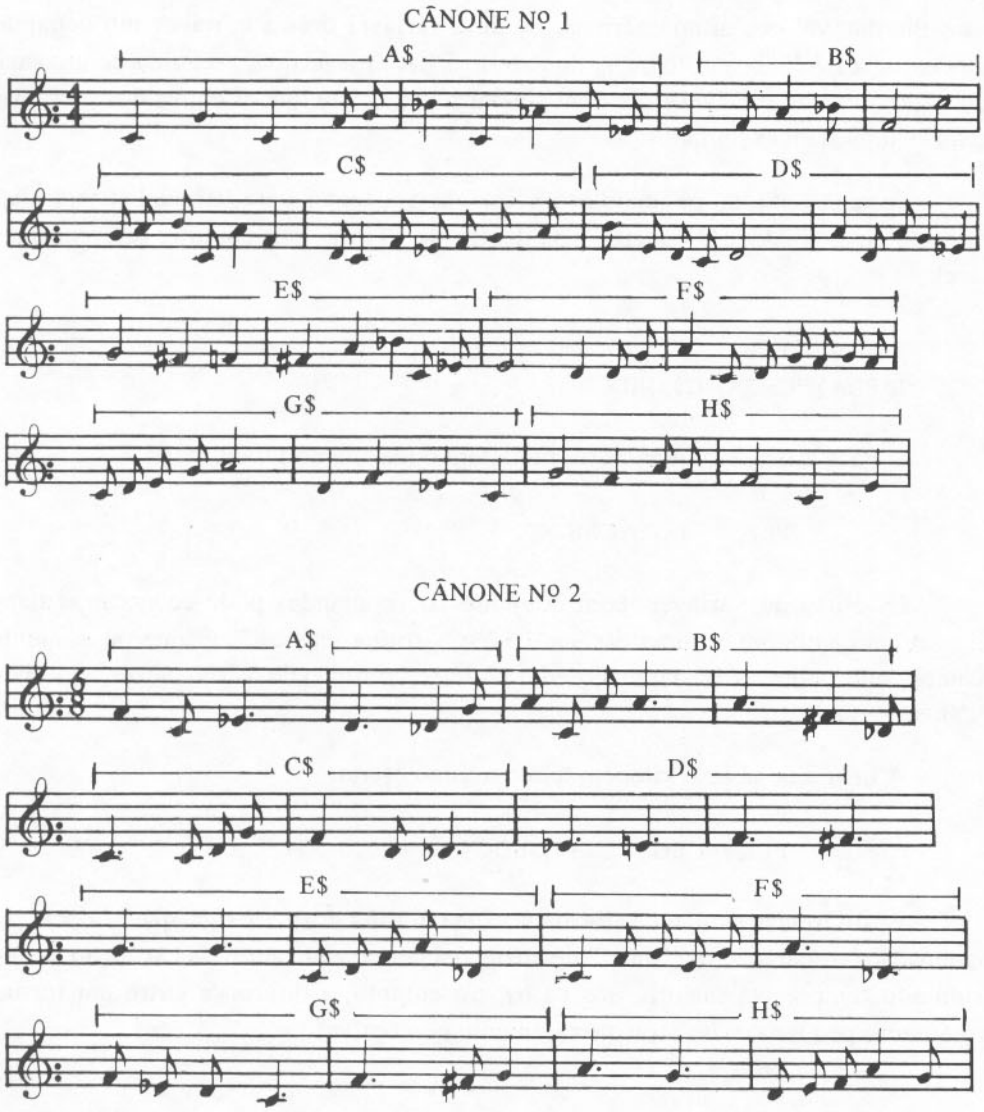


Figura 2.3 Dois cânones

atrasada. Depois outra voz entra, cantando a mesma melodia. O efeito é muito bonito. Além disso, é a forma musical onde identificamos mais facilmente cada uma das três vozes diferentes.

Os programas exemplo CANON1 e CANON2 são a transcrição dos cânones da Figura. 2.3.

Programa-exemplo CANON1

```
10 CLEAR 1ØØØ
40 PLAY "L8Ø3","L8Ø4","L8Ø5"
60 A$= "C4  G4  C4  F G B-4 C4  A-4GE- "
70 B$= "E2      F A4  B-F2      C2     "
80 C$= "G A B C A4  F4  D C4  F E-FGA- "
90 D$= "B E D C D2      A4  D A G4 E-4 "
100 E$="G2      F+4 F4  F+4 A4  B-4 CE-"
110 F$="E2      D4  D G A4  C D G F GF "
120 G$="C D E G A2      D4  F4  E-4 C4 "
130 H$="G2      F4  A G F2      C4  E-4"
200 PLAY A$+B$+C$+D$+E$+F$+G$+H$
300 PLAY "R1","R1","R1"
350 PLAY A$
360 PLAY B$,A$
365 FOR N=1 TO 4
370   PLAY C$,B$,A$
380   PLAY D$,C$,B$
390   PLAY E$,D$,C$
400   PLAY F$,E$,D$
410   PLAY G$,F$,E$
420   PLAY H$,G$,F$
430   PLAY A$,H$,G$
440   PLAY B$,A$,H$
450 NEXT N
600 END
```

Programa-exemplo CANON2

```
10 CLEAR 1ØØØ
40 PLAY "L8Ø3","L8Ø4","L8Ø5"
60 A$= "F4   C  E-4.     D4.      D-4 G  "
70 B$= "A  C A  A4.      A4.      F+4 D- "
80 C$= "C4.      C  D  G F4     D D-4.   "
90 D$= "E-4.     E4.      F4.      F+4.   "
100 E$="G4.      G4.      C  D  F A D-4  "
110 F$="C4   F  G  E  G A4.      D-4.   "
120 G$="F E- D  C4.      F4.      F+ G4  "
130 H$="A4.      G4.      D  E F  A4  G  "
200 PLAY A$+B$+C$+D$+E$+F$+G$+H$
300 PLAY "R1","R1","R1"
350 PLAY A$
360 PLAY B$,A$
365 FOR N=1 TO 4
370   PLAY C$,B$,A$
380   PLAY D$,C$,B$
390   PLAY E$,D$,C$
400   PLAY F$,E$,D$
410   PLAY G$,F$,E$
420   PLAY H$,G$,F$
430   PLAY A$,H$,G$
440   PLAY B$,A$,H$
450 NEXT N
600 END
```

Preste atenção principalmente no encaminhamento das vozes. Tente perceber a superposição da melodia.

Ambos os programas têm a mesma estrutura. Cada uma das oito frases está numa variável alfanumérica diferente (A$ a H$).

A melodia é apresentada inicialmente por inteiro, numa única voz, para que ela

fique bem gravada (linha 200). Depois há uma pausa (linha 300) e o cânone propriamente dito se inicia. A primeira voz apresenta a primeira frase da melodia (linha 350) e a segunda voz entra logo em seguida (linha 360). Depois entra a terceira (linha 370). As três passam a tocar a mesma melodia, defasada de uma frase.

Na linha 400, por exemplo, temos a primeira voz tocando a frase F e a segunda voz tocando a frase E. Na linha 410, temos a **segunda** voz tocando a frase F. A primeira já está tocando a G. Tente perceber isso, tanto ouvindo como observando a listagem do programa.

Note que cada canal foi programado para uma oitava diferente (oitavas 2, 3 e 4).

No programa CANON2 note o uso das semínimas pontuadas.

CAPÍTULO 3

O PSG E O COMANDO SOUND

Até agora aceitamos o fato do MSX ser capaz de produzir sons, mas não entramos em detalhes de como ele é capaz de fazer isso, qual é o hardware ou a eletrônica envolvida.

Na verdade, a peça responsável pelo som, nesse computador, é um pequeno "chip" ou circuito integrado conhecido pela sigla PSG (*Programmable Sound Generator* – Gerador de Som Programável).

Todas as características de geração de som do MSX (três canais, forma de onda fixa, volume variável etc.) são, na realidade, características do PSG.

Sempre que um som tem de ser gerado, o processador central do MSX (o Z-80) manda uma mensagem ao PSG, que cuida dessa tarefa.

Uma das vantagens de se utilizar um circuito externo inteligente de som é a de que o processador central fica livre da tarefa de supervisionar a sonorização. Em alguns outros computadores, se você quer que um Dó seja tocado durante um segundo ou minuto, durante todo o segundo ou minuto que a nota estiver sendo tocada, o processador estará incapacitado para realizar qualquer outra tarefa. No MSX isso não acontece.

Quando você executa uma nota, por exemplo, na instrução PLAY "C", o BASIC do MSX entende que você quer tocar um Dó e passa essa informação ao PSG.

A instrução não é passada da maneira como nós a vemos, no formato do PLAY. Na verdade, o BASIC decodifica o PLAY em instruções de nível mais baixo. Ele tem de informar detalhadamente ao PSG como fazer para produzir um Dó, na oitava que queremos.

Esse conjunto de instruções do PSG é o que aprenderemos a usar para termos domínio total sobre o som sem a intermediação do PLAY. Isso nos dará uma versatilidade muito maior na produção de sons.

O PSG é, como seu nome diz, um gerador **programável**, o que implica numa maneira de programá-lo. Isso se faz por meio dos seus **registradores.**

O PSG contém, dentro dele, um conjunto de 16 registradores, numerados de 0 a 15, que funcionam como se fossem 16 posições de memória de um byte (8 bits) cada. Cada um desses registradores tem uma função específica dentro do processo de programação e geração de som.

É através desses registradores que o PLAY trabalha e é através deles que trabalharemos, agora, diretamente.

A instrução que permite a comunicação direta do usuário com os registradores do PSG, sem a intervenção do PLAY, é a instrução SOUND, que tem o seguinte formato:

SOUND *registrador, conteúdo*

No lugar de *registrador* devemos colocar o número do registrador (de 0 a 15) e no lugar de *conteúdo* devemos colocar o valor que desejamos que fique no registrador que tem este número. O conteúdo pode variar de 0 a 255, como já seria de se esperar, já que cada registador tem oito bits, mas alguns dos registradores têm certas restrições aos possíveis valores.

Para colocarmos, por exemplo, o conteúdo 34 no registrador 2, usamos a seguinte instrução:

```
SOUND 2,34
```

A lista dos registradores com suas funções e os limites dos valores que eles podem assumir está na Tabela 3.1

Antes de passarmos para uma discussão detalhada do funcionamento dos registradores, vamos analisar o tópico dos "pares de registradores".

Tabela 3.1 Registradores do PSG, sua função e faixa de valores possíveis. Os registradores marcados com * podem assumir valor 16, no caso de envoltória automática.

REGISTRADOR	FUNÇÃO	VALORES POSSÍVEIS
0	período canal A (ajuste fino)	0 a 255
1	período canal A (ajuste grosso)	0 a 15
2	período canal B (ajuste fino)	0 a 255
3	período canal B (ajuste grosso)	0 a 15
4	período canal C (ajuste fino)	0 a 255
5	período canal C (ajuste grosso)	0 a 15
6	período do ruído	0 a 31
7	mixer	0 a 63
8	volume canal A	0 a 15*
9	voume canal B	0 a 15*
10	volume canal C	0 a 15*
11	período da envoltória automática (ajuste grosso)	0 a 255
12	período da envoltória automática (ajuste fino)	0 a 255
13	tipo da envoltória automática	0 a 15

3.1. PARES DE REGISTRADORES

Alguns dos registradores trabalham, na realidade, em "pares". Por exemplo: os registradores 4 e 5, trabalhando em dupla, programam o período do canal C. Não se preocupe, por enquanto, com o que seja "período".

Esses pares existem nos casos onde a característica desejada (como o período do canal C, no exemplo acima) deve poder assumir valores maiores do que 255.

Ora, um byte está limitado a um valor máximo de 255. Para que uma dada característica possa assumir um valor maior que 255, precisamos de um outro byte para completar o valor. Essa é a razão dos pares.

Os pares estão indicados, na Tabela 3.1, por duplas de "ajuste grosso" com "ajuste fino". Esses pares são conhecidos também, respectivamente, como "mais significativo" e "menos significativo" ou "MSB" e "LSB".

Vejamos como temos de partir dos ajustes grosso e fino para chegar ao valor desejado e vice-versa.

Para combinar os valores de grosso e fino e chegar ao valor final usaremos a seguinte expressão:

Valor = Fino + 256xGrosso

Voltando ao exemplo acima, vamos supor que temos um valor de 44 no registrador 4 (ajuste fino do período do canal C) e um de 2 no registrador 5 (ajuste grosso do período do canal C). Que período total o valor desses registradores indica?

Usando a expressão acima chegamos ao resultado 556 (o resultado de 44 + 256x2). Este é o período do canal C se esses valores estiverem programados.

Qual o valor máximo para o período do canal C? Para responder a essa pergunta, tudo o que temos a fazer é calcular o valor total do período quando os dois registradores (grosso e fino) estiverem com seu valor máximo.

O valor máximo do registrador 4 (fino) é 255 e o valor máximo do registrador 5 (grosso) é 15. O período total máximo será 4095 (o resultado de 255 + 15x256).

Vamos fazer o contrário agora: dado um valor, como fazer para encontrar o valor individual de cada registrador que compõe o par?

Os valores para cada um dos registradores (grosso e fino) são dados pelas expressões abaixo:

```
Grosso = Valor \ 256
Fino = Valor MOD 256
```

A barra invertida(\ ) significa "divisão inteira" no BASIC. É a divisão que despreza a parte fracionária. O operador MOD é o operador "resto". O resultado de "Valor MOD 256" é o resto da divisão de Valor por 256.

Então, continuando com o mesmo exemplo acima, vamos supor que nós quiséssemos programar o período do canal C com o valor de 2000. Como faríamos para descobrir que valores colocar nos registradores grosso e fino?

O valor do registrador grosso deve ser o resultado da divisão inteira de 2000 por 256: 7. O valor do registrador fino é o resto dessa mesma divisão: 208.

Então, colocando o valor de 208 no registrador 4 e o valor 7 no registrador 5 estaremos, na verdade, obtendo um período de 2000 no canal C.

Vamos passar agora à discussão da função e programação dos registradores.

Como estaremos atuando diretamente no PSG, agora, precisaremos nos aprofundar um pouco mais nos fundamentos do som. Vamos ver alguns conceitos bem simples utilizados no estudo do som, do ponto de vista da Física.

CAPÍTULO 4

FREQÜÊNCIA

Até agora, quando falamos em som, falamos em evento sonoro. Esse conceito é muito útil mas não está ligado, de forma alguma, à maneira como o som é produzido, transmitido nem compreendido por nossos ouvidos. Daqui para a frente, estaremos envolvidos com características da produção de som que exigem que façamos uma pausa para verificar este ponto.

Em primeiro lugar, vejamos como é criado um som.

Som é vibração. Todo som é produzido por uma vibração muito rápida de alguma coisa.

Uma corda de violão, por exemplo, vibra muito rapidamente (qualquer coisa da ordem de 400 vezes por segundo) para que seu som seja ouvido. A **freqüência** é o número de oscilações por segundo (ou por unidade de tempo) de uma dada vibração. Ela indica o número de vezes, por segundo, que uma certa coisa está indo e voltando.

Uma unidade de medida para a freqüência é o **Hertz** ou oscilações por segundo. A corda mencionada acima estaria vibrando a uma freqüência de 400 Hertz (abrevia-se 400 Hz).

O **período** de uma vibração é o **tempo** que o vai-e-vem demora. Se temos uma certa freqüência de 0,5 Hz (meia vez por segundo) temos um período de 2 segundos (o vai-e-vem vai demorar dois segundos). As relações seguintes permitem que você saiba o período a partir da freqüência e vice-versa:

$$T = 1/F \qquad e \qquad F = 1/T$$

T representa o período e F, a freqüência. Um é sempre o inverso do outro.

Sendo o período um intervalo de tempo, ele tem a mesma unidade que o tempo (segundos, milissegundos, microssegundos, horas...).

A afinação tem uma relação direta com a freqüência (e com o período) de oscilação de um som.

4.1. FREQÜÊNCIA DAS NOTAS MUSICAIS

Se você tiver um piano ou outro instrumento de teclado em casa, procure o Lá central desse instrumento. Este Lá tem uma freqüência de 440 Hz e por isso é usualmente conhecido pelos músicos como o **Lá 440**. Mas note bem: não é qualquer Lá, em qualquer oitava, que tem a freqüência de 440 Hz, mas apenas o Lá central. Na instrução PLAY, ele é o Lá da oitava número 4.

Isso é uma convenção. Convencionou-se, mundialmente, considerar a freqüência de 440 Hz como sendo o Lá central. Todos os instrumentos, a partir daí, passaram a ser afinados segundo essa freqüência de referência. Se você comprar um diapasão Lá para ajudá-lo a afinar seu instrumento, estará, na realidade, comprando uma ferramenta que foi calibrada com toda a precisão para produzir uma freqüência de 440 Hz quando percutida.

De vez em quando, as pessoas responsáveis por esses "standards", que eu não sei bem quem são (regentes alemães? físicos americanos?), resolvem mudar o Lá central, por alguma razão de construção dos instrumentos. Se não me engano, o Lá central, em 1982, estava cotado em 443 Hz. Isso não faz uma diferença tão grande, afinal de contas, já que qualquer grupo de músicos, antes de começar a tocar, afina os respectivos instrumentos de acordo com um padrão (um dos instrumentos, um diapasão ou um afinador eletrônico). Todos têm, portanto, seus instrumentos emitindo freqüências baseadas na mesma freqüência de referência, seja ela o Lá 440 ou o Lá 443 ou qualquer outra. Além disso, a diferença de afinação do Lá 440 para o Lá 443 é imperceptível para um ouvido não-técnico. Sendo assim, podemos considerar o Lá como sendo 440, por tradição.

Assim como o Lá central tem a freqüência absoluta e universal de 440 Hz, todas as outras notas correspondem a freqüências exatas e absolutas.

Em primeiro lugar, vejamos como fazer para, sabendo a freqüência absoluta de uma determinada nota, descobrir a freqüência da mesma nota, em outra oitava.

A regra é simples:

> Para subir uma oitava, multiplique a freqüência por dois. Para descer uma oitava, divida por dois.

O Lá uma oitava acima do Lá central (freqüência 440) tem freqüência igual a 880 Hz. O Lá uma oitava acima deste tem freqüência 1760 Hz. O Lá uma oitava abaixo do Lá central tem freqüência igual a 220 Hz. E por aí vai...

Com essa simples regra, podemos descobrir todas as freqüências das oitavas de uma nota com freqüência conhecida.

E as outras notas, qual a freqüência delas? Para descobrir isto, teremos de usar um pouco de matemática.

A fórmula geral para as oitavas de uma nota é:

$$N_{a+n} = 2^n N_a$$

Onde n é o número de oitavas a subir (n tem valor negativo se forem oitavas a descer). N_a é a nota básica e N_{a+n} é a nota, n oitavas acima ou abaixo, conforme o sinal. Um exemplo: a freqüência do Lá duas oitavas acima do Lá central é

$$N_{a+2} = 2^2 \times 440 = 4 \times 440 = 1760 \text{ Hz}$$

Vamos refletir um pouco: o que significa subir uma oitava? Significa dobrar a freqüência, mas significa também subir 12 meios-tons. Vamos reescrever a fórmula geral acima para levar isto em conta:

$$F_{12n} = 2^n F_0$$

Onde F_0 indica a freqüência básica, n indica o número de oitavas acima e F_{12n} indica a freqüência da nota n oitavas ou n vezes doze meios-tons acima.

Qual a relação entre a freqüência de uma nota e a de outra, um meio-tom acima? Vamos reformular a pergunta para: **qual a relação entre uma nota e outra, um doze-avos de oitava acima**? Basta utilizar a fórmula acima com n igual a 1/12.

$$F_1 = 2^{1/12} F_0$$

Esta resposta é muito importante. Ela indica que as freqüências de quaisquer duas notas adjacentes da escala cromática temperada tem uma proporção constante. A freqüência da nota, dividida pela freqüência da nota imediatamente anterior (meio-tom

abaixo) é **SEMPRE** igual a

$$2^{1/12}$$

que é igual à raiz décima-segunda de dois, que é igual a 1.05946.

Já temos, então, a regra para o meio-tom adjacente:

> Para subir meio-tom, multiplique a freqüência por K. Para descer meio-tom, divida por K. K é uma constante igual a $2^{1/12}$ ou 1.05946.

Com isto, podemos completar o cálculo das freqüências que faltam. Vamos descobrir a freqüência do Lá # imediatamente acima do Lá Central. Essa freqüência será igual a K vezes a freqüência do Lá Central (440 Hz). Ela é aproximadamente igual a 466 Hz.

O próximo programa-exemplo (FREQ1) ajuda a calcular freqüências de notas. Insira uma freqüência básica e um número de notas. O resultado do programa é uma série de freqüências que são as correspondentes à escala cromática iniciada naquela nota.

Programa-exemplo FREQ1

```
50 CL=1789770! ' clock
60 C=CL/16
100 PP=2^(1/12)
150 PRINT"A PROPORCAO ENTRE UMA NOTA E OUTRA"
160 PRINT"MEIO TOM ACIMA E' : ";PP
300 INPUT"FREQUENCIA INICIAL";FQ
320 INPUT"NUMERO DE NOTAS";NU
390 FR=FQ
400 FOR N=1 TO NU
410   PRINT"NOTA";N;
420   PRINT USING" FREQ = #####  ";FR;
580   FR=FR*PP
590   PRINT
600 NEXT N
```

Se, por exemplo, você colocar como freqüência básica o valor de 440 Hz e um número e notas iguais a 4, você terá como resultado as freqüências das seguintes notas, nesta ordem: o Lá Central, o Lá# imediatamente acima deste, o Si e o Dó.

4.2. A FREQÜÊNCIA E O PSG

O PSG do MSX recebe o período (e não a freqüência) como dado de entrada para a afinação de um som. O período não é informado em segundos nem em milissegundos. Ele é informado numa unidade bastante incomum, que é igual a dezesseis vezes o período do clock interno do micro.

Todo micro tem um componente eletrônico vital: o cristal ou clock. Ele fornece, ininterruptamente, uma seqüência de pulsos, igualmente espaçados no tempo, para que o microprocessador fique sincronizado. Esse é o famoso "cristal-de-tantos-megahertz". O chip PSG usa os pulsos do mesmo cristal.

Cada marca de micro tem um cristal diferente, com uma freqüência diferente. Segundo os manuais dos micros MSX nacionais, temos as seguintes freqüências de cristal:

HOT-BIT: FC = 1789770 Hz
EXPERT: FC = 1788055 Hz

Na verdade, o cristal tem uma freqüência que é igual ao dobro da apresentada acima. O PSG usa a freqüência do clock dividida por dois.

O período do cristal será, então, 1/FC. O período-base do PSG, como já informamos antes, é igual a 16 vezes o período deste clock, ou 16/FC.

Vamos chamar de F a freqüência em Hertz que desejamos obter. O período em segundos dessa freqüência é 1/F. O período que devemos informar ao MSX (T_{msx}) é o período em segundos que desejamos obter, dividido pelo período-base, ou seja:

$$T_{msx} = \frac{FC}{16 \times F}$$

Qual é o período que devemos informar ao MSX (HOT-BIT) para que este produza um Lá central (F = 440 Hz)?

$$T_{msx} = \frac{1789770}{16 \times 440} = 254$$

Para o EXPERT, este valor seria o mesmo após o arredondamento, pois a diferença entre as freqüências dos clocks é muito pequena.

O programa-exemplo FREQ2 é semelhante ao programa FREQ, mas produz também os valores de T_{msx}.

Programa-exemplo FREQ2

```
50 CL=1789770! ' clock
60 C=CL/16
100 PP=2^(1/12)
150 PRINT"A PROPORCAO ENTRE UMA NOTA E OUTRA"
160 PRINT"MEIO TOM ACIMA E' : ";PP
300 INPUT"FREQUENCIA INICIAL";FQ
320 INPUT"NUMERO DE NOTAS";NU
390 FR=FQ
400 FOR N=1 TO NU
410   PRINT"NOTA";N;
420   PRINT USING" FREQ = #####  ";FR;
430   PM=C/FR
440   PRINT USING"T MSX = #####";PM;
470   SOUND 0,PM MOD 256
480   SOUND 1,PM \ 256
500   SOUND 8,10
550   FOR I=0 TO 100:NEXT I ' ESPERA
560   SOUND 8,0
580   FR=FR*PP
590   PRINT
600 NEXT N
```

Lembre-se de alterar a linha que contém o valor da freqüência do clock se o seu computador for um EXPERT. Se for um outro MSX, vasculhe o manual para encontrar o valor correto do clock (lembre-se de dividir por dois).

Já descobrimos o valor de período para que o MSX produza um Lá central

(254). Para que o canal A do PSG produza este Lá, temos de programar o par de registradores 0 e 1 com este período. Lembre-se de como colocar um valor num par de registradores, como mencionado acima.

Além disso, temos de programar o volume desse canal (registrador 8) para algum valor entre 1 e 15, para que o som possa ser ouvido.

Execute as seguintes instruções para programar o Lá central no canal A:

```
SOUND 0, 254 MOD 256
SOUND 1, 254 / 256
```

e esta instrução para colocar o volume do canal A em 10:

```
SOUND 8,10
```

Para desligar o som, zere o volume:

```
SOUND 8,0
```

Note como o som não pára enquanto o volume não for zerado. Isso é uma conseqüência da geração externa de som. Você pode programar o PSG para tocar uma freqüência e trabalhar no BASIC enquanto o som fica sendo constantemente produzido. O processador central nem se importa se o PSG está ou não produzindo um som. Isso é de grande valia na economia do tempo precioso do processador.

Para comparar a afinação do Lá central obtida desta maneira e a afinação do Lá central segundo o comando PLAY, execute novamente as três primeiras instruções mas não execute a última, que corta o som. No lugar dela, execute:

```
PLAY"O4A1"
```

que é o comando PLAY para gerar o mesmo Lá central por uma semibreve. Ambas as afinações devem soar absolutamente iguais, pois o BASIC-MSX também usa o Lá central como sendo 440 Hz. Se isso não ocorrer, verifique a freqüência do clock no manual do MSX que você está usando.

Como já mencionamos, certas orquestras usam o Lá central como tendo uma freqüência de 443 Hz. Repita os cálculos acima para gerar essa freqüência e compare com o Lá central (440) do PLAY. Note como a diferença entre eles é pequena.

O próximo programa-exemplo é um programa que simula, no teclado do MSX, um teclado tradicional de piano, com uma oitava e uma terça (16 notas).

Programa-exemplo PIANO1

```
50 CLS
100 DIM FR(16)
150 FOR N=Ø TO 16
160   READ FR(N)
170 NEXT
180 DATA 214,2Ø2,19Ø,18Ø,17Ø
190 DATA 16Ø,151,143,135,127
200 DATA 12Ø,113,1Ø7,1Ø1,95
210 DATA 9Ø,85
250 TC$="AWSEDFTGYHUJKOLPÇ"
300 SOUND 8,Ø
310 SOUND 1,Ø
350 PRINT"PIANO :"
355 PRINT
360 PRINT" W E   T Y U   O P"
365 PRINT"A S D F G H J K L Ç"
370 PRINT:PRINT
380 PRINT"USE A BARRA DE ESPACO PARA PARAR O SOM"
390 PRINT"USE ESC PARA TERMINAR"
400 A$=INKEY$
410 IF A$="" THEN 4ØØ
450 N=INSTR(TC$,A$)-1
470 IF N>=Ø THEN 1ØØØ
500 IF A$=" " THEN SOUND 8,Ø
510 IF A$=CHR$(27) THEN END
550 GOTO 4ØØ
1000 SOUND 8,1Ø
1010 SOUND Ø,FR(N)
1020 GOTO 4ØØ
```

O programa não usa o PLAY para a produção do som, usa apenas o controle direto de afinação do PSG através do comando SOUND. Por essa razão o som é contínuo. Veremos a flexibilidade que esse controle sobre o tempo de execução da nota pode nos dar mais tarde, quando falarmos de envoltória.

Os períodos utilizados para as notas (os 16 valores dentro dos DATAs das linhas 180 a 200) foram calculados para o HOT-BIT, usando o programa-exemplo FREQ2. Se você tiver outro MSX, você pode querer recalcular as freqüências mas isso não é tão necessário, a menos que você queira conferir a afinação com um diapasão. Os clocks dos outros sistemas não são, em geral, tão diferentes assim. Mesmo assim, a afinação relativa entre as notas permanece perfeitamente correta. O que muda, ligeiramente, é a afinação em relação a outros instrumentos.

Já sabemos como fazer para programar a freqüência de um canal. Como o MSX dispõe de três canais independentes, podemos programar uma freqüência diferente em cada um deles.

É isso o que usaremos para criar uma pequena variação de timbre.

CAPÍTULO 5

TIMBRE E HARMÔNICOS

Como já mencionamos de passagem, o parâmetro timbre, de um evento sonoro, depende muito da forma de onda da nota. Mas o que é esta forma de onda?

A forma de onda é a maneira como o som oscila. Já vimos que, para que um som seja produzido, alguma coisa deve estar oscilando muito rapidamente. Quando dizemos "alguma coisa que oscila", queremos dizer um objeto mesmo, como a corda de um violão, o papelão de um alto-falante, o vidro de uma garrafa que se quebra e até suas próprias cordas vocais.

O objeto que produz o som sempre tem uma maneira peculiar de oscilar. Uma corda de violão pode, por exemplo, seguir pacientemente até um lado e imediatamente iniciar o caminho de volta, no mesmo passo em que ela foi.

Outro objeto (a palheta de um saxofone, por exemplo) pode demorar o mesmo tempo que a corda de violão, para ir de um lado ao outro, executando a oscilação de uma forma diferente de um vai-e-vem regular. Esse segundo objeto poderia, por exemplo, saltar bruscamente até um lado, esperar um tempo e depois saltar da mesma maneira para o outro lado.

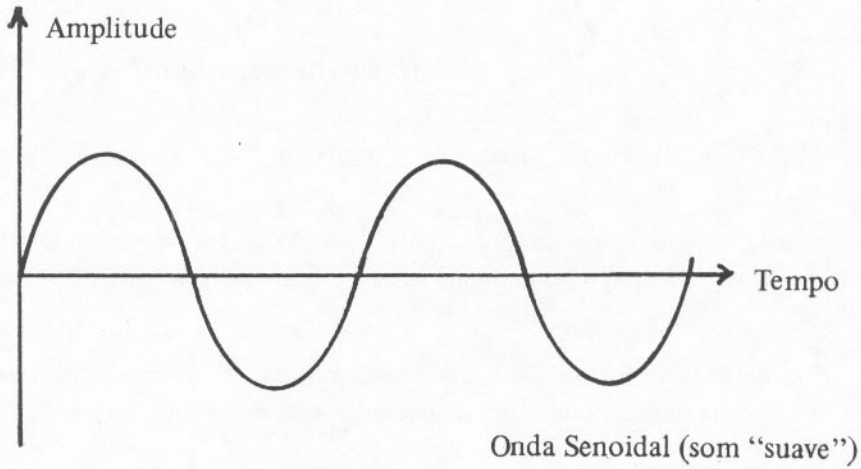
Por incrível que pareça, nosso ouvido é capaz de distinguir os dois tipos de movimento, mesmo estando esses dois movimentos ocorrendo 400 vezes por segundo. A corda que vai-e-vem regularmente e o objeto que vai-e-vem com saltos bruscos produzem sons de **timbres diferentes e de formas de onda diferentes.**

Nosso ouvido não identifica o tipo de vai-e-vem, não identifica um vai-e-vem brusco ou contínuo. Quando chamados a qualificar os dois tipos de sons, diremos qualquer coisa como: "ah... o segundo som é mais áspero".

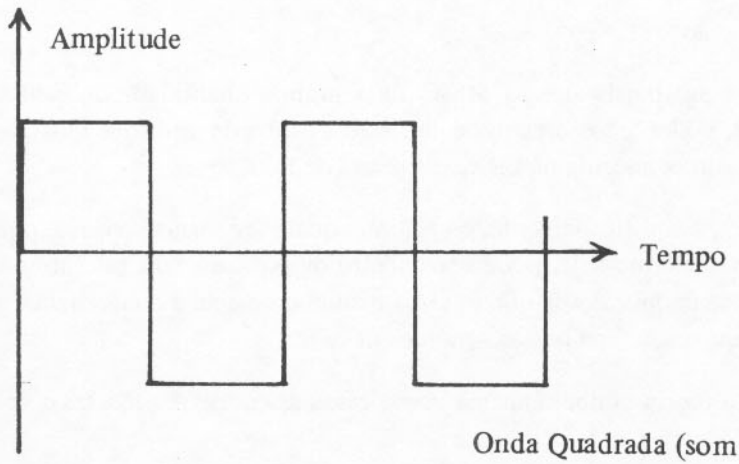
Esse é o efeito que a forma de onda produz: o timbre.

Os dois timbres descritos acima poderiam corresponder, respectivamente, a uma forma de onda senoidal e a uma quadrada.

Na Figura 5.1 estão algumas formas de onda comuns, com alguns comentários.



Onda Senoidal (som "suave")



Onda Quadrada (som "muito áspero")

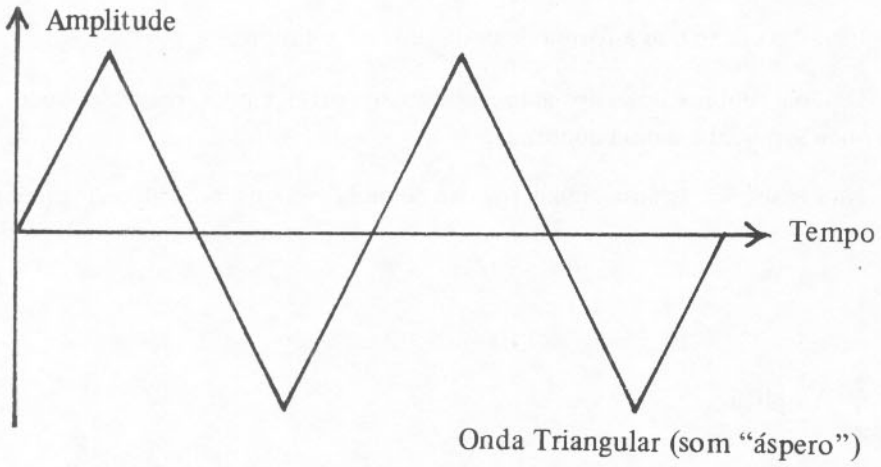


Figura 5.1 Formas de onda

É usual sentirmos um timbre áspero quando a forma de onda é cheia de movimentos bruscos. A forma de onda mais suave, a senoidal, é a forma de onda normal da flauta doce.

A variação da forma de onda proporciona sons com "personalidade". Construindo formas de onda, estamos, na realidade, construindo sons "cheios", "quentes", "etéreos"...

Mas, como já vimos:

> O PSG do MSX não prevê o controle da forma de onda.

O que fazer então?

Se tivéssemos, no hardware do MSX, uma grande quantidade de canais com forma de onda senoidal, poderíamos criar qualquer outra forma de onda que quiséssemos, segundo uma técnica muito conhecida na Física: as séries de Fourier.

Essa teoria diz, simplificadamente, o seguinte: qualquer forma de onda, por mais esquisita e cheia de detalhes que seja, pode ser construída por uma soma de um número muito grande de ondas senoidais, desde que se saiba exatamente qual a freqüência e qual o volume (ou amplitude) que cada uma dessas senóides deve ter.

É lógico que a teoria também ensina como calcular essas freqüências e volumes corretos.

Se você, por exemplo, quer uma onda quadrada e só dispõe de geradores de ondas senoidais, tudo bem! Basta você saber quais as freqüências corretas e quais volumes você deve programar nos seus geradores de onda senoidal, somar todas as ondas que eles produzem e pronto! Você tem uma onda aproximadamente quadrada. E quanto mais geradores senoidais você usar, mais próxima sua onda estará da onda quadrada.

A teoria também nos ensina a calcular as freqüências e os volumes para obter a onda quadrada desejada no exemplo do parágrafo acima. O procedimento, no entanto, é trabalhoso e usa técnicas de cálculo avançado. Vamos ver apenas os resultados, a título de exemplo. As senóides a somar devem ser: a senóide com a afinação desejada e com um terço do volume final desejado (chamada de fundamental), a senóide com o triplo da freqüência mas com um terço do volume da fundamental, a senoide com o quíntuplo da freqüência e um quinto do volume da fundamental e assim por diante (7,9,11,13...).

Somando todas essas ondas senoidais obteremos uma onda quadrada.

A onda senoidal que tem a freqüência básica é chamada de **fundamental.** As outras têm sempre freqüências que são múltiplos inteiros da fundamental. Estas são chamadas de **harmônicos.**

Portanto, o processo da série de Fourier pode ser escrito como: **qualquer onda é uma soma de harmônicos.**

Esse processo de sintetizar uma forma de onda através da soma de harmônicos é também chamado de **síntese por soma.**

Os harmônicos são numerados. O primeiro harmônico é a própria fundamental. O segundo harmônico é a onda com o dobro da freqüência da fundamental. O terceiro harmônico é a onda com o triplo da freqüência da fundamental, e assim por diante. Podemos dizer, então, que uma onda quadrada pode ser obtida pela soma dos harmônicos ímpares (veja que no exemplo da onda quadrada, lá atrás, só usamos as freqüências múltiplas de números ímpares).

Se você toca órgão, deve estar reconhecendo esse método de variação de timbre. Os órgãos mais antigos usam exatamente esse método com uma nomenclatura diferente. Os diversos geradores senoidais, cada qual com um dos harmônicos, são chamados de registros. Experimente preparar o órgão apenas para emitir o som fundamental, sem nenhum harmônico. Esse é o som de uma senóide. Experimente agora colocar harmônicos ímpares e veja como o som vai ficando cada vez mais áspero.

No MSX não temos um número grande de geradores senoidais. Longe disso. Temos apenas três geradores de onda quase quadrada.

O máximo que podemos fazer é um arremedo de série de Fourier. Podemos somar três ondas de freqüências diferentes, programando cada canal para produzir uma delas.

É isso que o programa PIANO2 faz. Ele é semelhante ao programa PIANO1 anterior. A diferença é que ele usa os três canais e pede a relação de freqüências entre a fundamental e a harmônica a ser tocada no canal B e entre a fundamental e a harmônica a ser tocada no canal C.

Além disso, o programa pede também os volumes dos três canais.

Não pense nesse gerador de onda como sendo um que usa séries de Fourier. Ele não é. Ele é uma maneira de se contornar o problema do timbre fixo do MSX através da execução simultânea de três freqüências diferentes, de uma maneira inspirada nas séries de Fourier.

Já vimos como controlar o parâmetro afinação e conseguimos alguma variação no parâmetro timbre. Vamos agora controlar o parâmetro intensidade através do envelope.

Programa-exemplo PIANO2

```
100 DIM FR(16)
150 FOR N=Ø TO 16
160   READ FR(N)
170 NEXT
180 DATA 214,202,190,180,170
190 DATA 160,151,143,135,127
200 DATA 120,113,107,101,95
210 DATA 90,85
220 TC$="AWSEDFTGYHUJKOLPÇ"
240 SOUND 8,Ø:SOUND 9,Ø:SOUND 10,Ø
250 SOUND 1,Ø:SOUND 3,Ø:SOUND 5,Ø
255 CLS
260 INPUT"VOLUME DA FUNDAMENTAL";V0
265 INPUT"VOLUME DA HARM. 1";V1
270 INPUT"VOLUME DA HARM. 2";V2
275 INPUT"RELACAO FUND/HARM.1";R1
280 INPUT"RELACAO FUND/HARM.2";R2
350 PRINT"PIANO :"
355 PRINT
360 PRINT" W E   T Y U   O P"
365 PRINT"A S D F G H J K L Ç"
370 PRINT:PRINT
380 PRINT"USE A BARRA DE ESPACO PARA PARAR O SOM"
390 PRINT"USE ESC PARA TERMINAR"
400 A$=INKEY$
410 IF A$="" THEN 4ØØ
450 N=INSTR(TC$,A$)-1
470 IF N>=Ø THEN 1ØØØ
500 IF A$=" " THEN SOUND 8,Ø:SOUND 9,Ø:SOUND 10,Ø
510 IF A$<>CHR$(27) THEN 4ØØ
520 INPUT"REPROGRAMAR";X$
530 IF LEFT$(X$,1)="S" OR LEFT$(X$,1)="s" THEN 240
550 END
1000 SOUND 8,V0:SOUND 9,V1:SOUND 10,V2
1010 SOUND Ø,FR(N):SOUND 2,FR(N)/R1:SOUND 4,FR(N)/R2
1020 GOTO 4ØØ
```

CAPÍTULO 6

O ENVELOPE OU ENVOLTÓRIA

O volume de um som pode variar (normalmente varia) nos instrumentos não sintetizados. Se tocarmos uma corda de violão podemos perceber como o som vai morrendo com o tempo. Se não encostarmos na corda, uma vez percutida, o som da corda vai diminuindo de intensidade até morrer.

Assim como o som da corda do violão tem esse comportamento característico de morrer depois de um certo tempo, cada instrumento tem seu modo característico de variar de volume em função do tempo.

> A maneira como um determinado som (ou evento sonoro) muda de volume com o tempo é chamada de envoltória ou envelope.

A envoltória, mais precisamente, é a função volume x tempo ou **intensidade x tempo.**

Essa definição de envoltória ainda não é precisa, pois pode causar a confusão que os manuais do MSX normalmente causam. É importante notar, portanto, o seguinte:

> A envoltória não é a forma de onda.

Como já vimos, a forma de onda é, também, a variação da intensidade com o tempo. O tempo em que a intensidade varia é da ordem de milissegundos (milésimos de segundo). Na envoltória, ao contrário, consideramos variações de intensidade em segundos ou décimos de segundo.

Vamos esclarecer esse ponto. Para produzir um som, como já vimos, precisamos vibrar alguma coisa muito rapidamente. Uma corda, por exemplo, vai e volta uma vez a

cada 2 milissegundos, aproximadamente, quando o Lá central é tocado. A corda, vibrando rapidamente, transmitirá essas vibrações para o ar que, por sua vez, vibrará a cada 2 milissegundos também. A intensidade do som carregado pelo ar varia, portanto, do mínimo ao máximo, uma vez a cada 2 milissegundos para que o Lá possa ser ouvido. Isto é a FORMA DE ONDA.

Esse Lá vai morrendo, depois de alguns segundos, pois a corda vai deixando de vibrar. O som, de uma maneira geral, vai diminuindo de intensidade durante alguns segundos. Isto é a ENVOLTÓRIA.

A Figura 6.1 mostra como a intensidade varia com o tempo no exemplo acima.

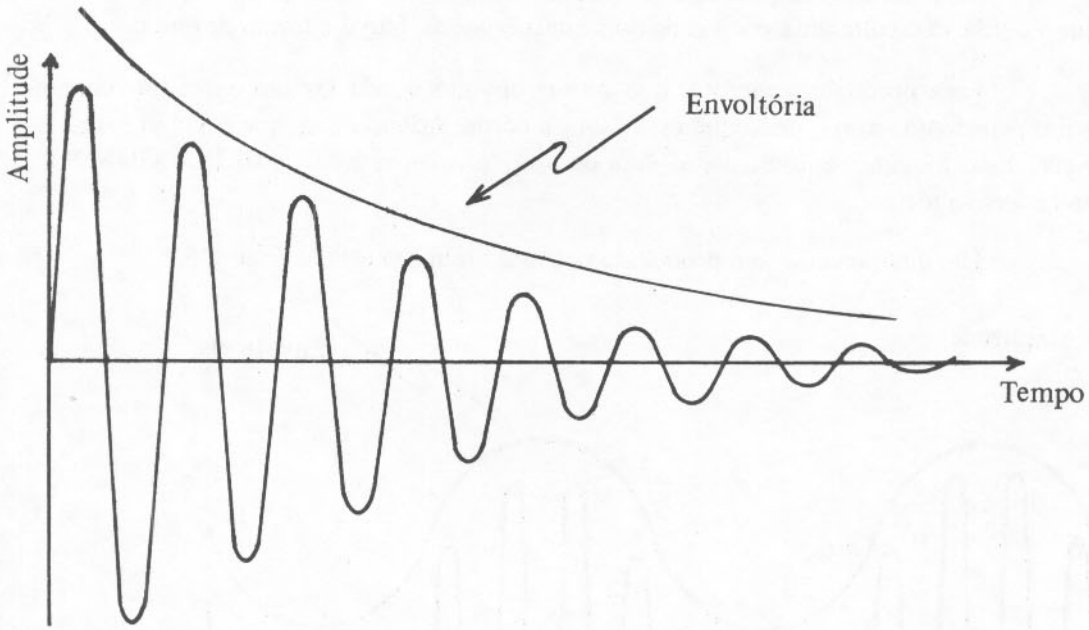


Figura 6.1 Envoltória da corda de violão

Poderiámos dizer também que a envoltória é a variação da **amplitude** da vibração com o tempo.

No início da vibração da nossa corda, se movimento é AMPLO: a corda vai longe e volta a cada dois milissegundos. Conforme o tempo vai passando, o vai-e-vem da

corda fica cada vez menos amplo até cessar por completo. O volume do som vai, portanto, caindo, mas a freqüência de vibração da corda permanece sempre a mesma: a corda vai e volta sempre com o mesmo período, apesar de ir cada vez menos longe. A nota é sempre a mesma (Lá), porque a freqüência da nota é a própria afinação.

Um outro tipo de envoltória muito comum e que causa bastante confusão entre forma de onda e envoltória é o **vibrato.**

É bastante difícil explicar o que é o vibrato apenas por meio de palavras. Ele é um recurso muito usado por cantores, organistas e violinistas para deixar uma nota mais "expressiva": é o "flutuar" de uma nota. A afinação permanece a mesma mas o volume varia periodicamente com o tempo, aumentando e diminuindo alternadamente.

Se um violinista produz um Lá central, seu arco está, na realidade, fazendo com que a corda vá e volte uma vez a cada dois milissegundos. Isto é a forma de onda.

Para produzir o vibrato, que é uma envoltória, ele faz um outro movimento, muito mais lento, com o dedo que está sobre a corda, fazendo com que o volume do som oscile. Esse movimento é um vai-e-vem de aproximadamente 200 MILISSEGUNDOS. Isso é a envoltória.

Um diagrama do som produzido por esse violinista está na Figura 6.2.

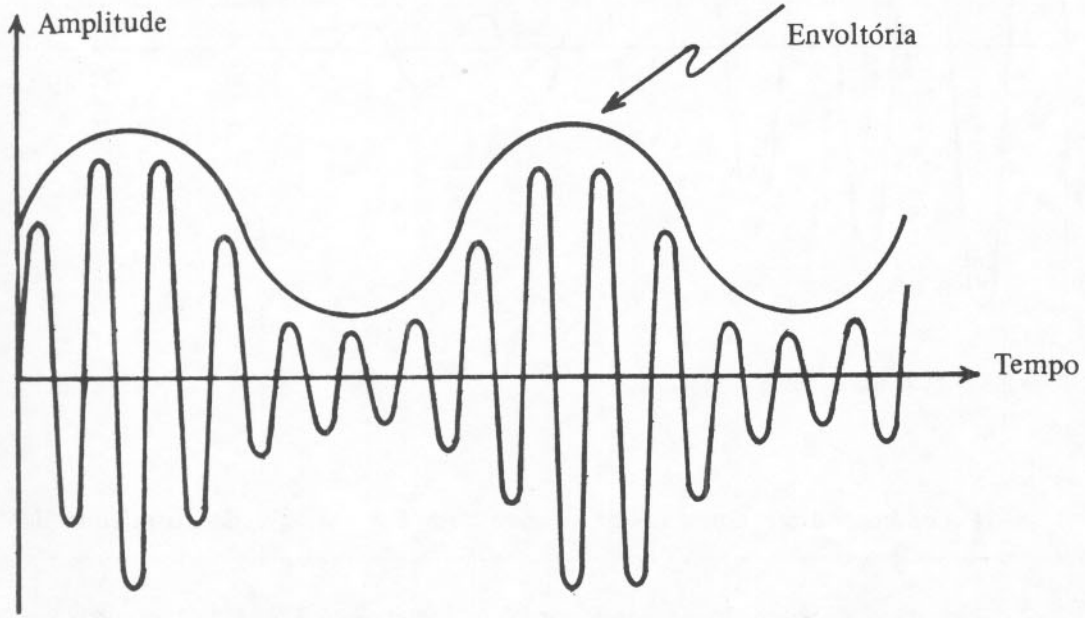


Figura 6.2 Envoltória do vibrato

Quando se fala em envoltória e esta é representada por meio de um gráfico, é costume omitir a forma de onda, já que sua representação apenas complicaria o gráfico. As envoltórias dos dois exemplos apresentados (o som de uma corda de violão morrendo e o violino executando um vibrato) são, respectivamente, estas:

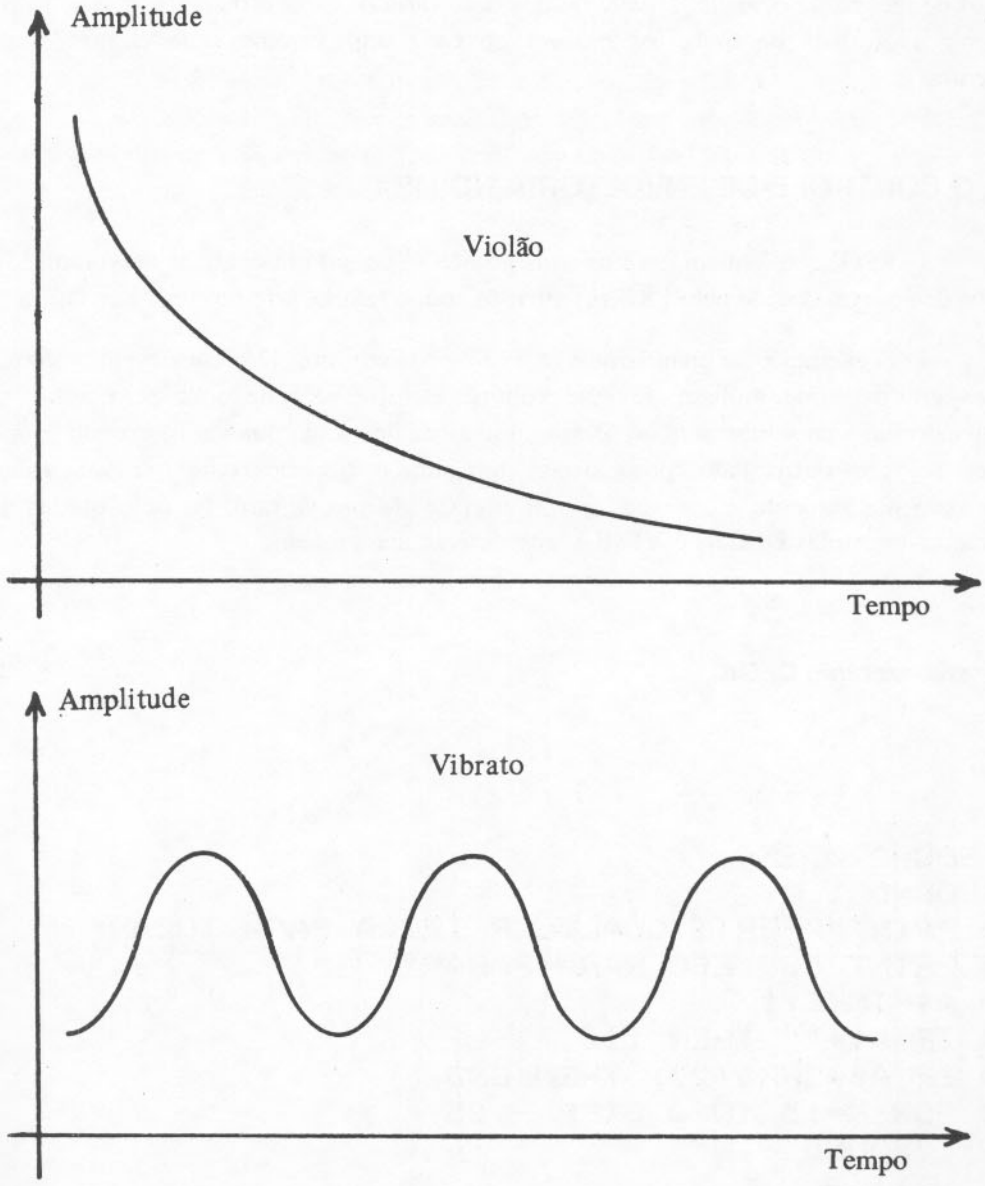


Figura 6.3 Envoltórias de violão e vibrato

Quando você estiver vendo um gráfico de envoltória, lembre-se, portanto, de que ele nada tem a ver com a forma de onda do evento sonoro (o timbre do evento sonoro). Podemos ter dois eventos sonoros de mesma forma de onda, com envoltórias diferentes (por exemplo, violino com e sem vibrato) ou dois sons de mesma envoltória e formas de onda diferentes, como por exemplo: um cantor e um violinista usando um vibrato de mesma velocidade. O volume dos dois varia da mesma maneira, mas se formos observar as formas de onda (os timbres de cada um) veremos que elas são muito diferentes.

6.1 O CONTROLE DE ENVOLTÓRIA NO MSX

O PSG, que é quem gera os sons no MSX, possui um controle de volume. Se o volume do som produzido pelo PSG for variado com o tempo, teremos uma envoltória.

Por exemplo, se mandarmos o PSG produzir um Dó com certo volume e formos gradualmente diminuindo esse volume até que ele chegue a zero, estaremos produzindo um som semelhante ao som de um corda de violão que vai morrendo pouco a pouco. Se, por outro lado, produzirmos um som e fizermos com que seu volume alternadamente aumente e diminua, estaremos fazendo um vibrato. Isso é o que os dois programas-exemplos DECAI e VIBRA, respectivamente, fazem.

Programa-exemplo DECAI

```
50 SOUND 0,254
60 SOUND 1,0
100 PRINT"APERTE QUALQUER TECLA PARA TOCAR"
105 PRINT "OU ESC PARA PARAR"
110 A$=INKEY$
120 IF A$="" THEN 110
130 IF A$=CHR$(27) THEN END
200 FOR N=15 TO 0 STEP -.05
210   SOUND 8,N
220 NEXT N
230 GOTO 110
```

Programa-exemplo VIBRA

```
50 SOUND Ø,254
60 SOUND 1,Ø
100 PRINT"APERTE QUALQUER TECLA PARA TOCAR"
105 PRINT "E ESC PARA PARAR"
110 A$=INKEY$
120 IF A$="" THEN 11Ø
150 A$=INKEY$
160 IF A$=CHR$(27) THEN 4ØØ
200 FOR N=11 TO 15 STEP .1
210   SOUND 8,N
220 NEXT N
230 FOR N=15 TO 11 STEP -.1
240   SOUND 8,N
250 NEXT N
300 GOTO 15Ø
400 SOUND 8,Ø
410 END
```

No programa DECAI, o loop das linhas 200 a 220 é que faz o volume cair, de 15 a 0, com uma velocidade de 0.05 por iteração. No programa VIBRA, o par de loops das linhas 160 a 250 faz o volume subir e depois descer entre os volumes 11 e 15, com uma velocidade de 0.1.

Se você mudar as constantes dos loops, estará mudando a envoltória. Experimente aumentar e diminuir as velocidades e limites inferior e superior.

Lembre-se da função dos registradores do PSG. Os registradores de números 0 e 1 controlam a afinação do canal A. Nas linhas 50 e 60, em ambos os programas, o canal A está sendo programado para emitir um Lá 440.

O próximo programa-exemplo (SOPRO) gera um som que tem uma envoltória mais complexa, característica dos instrumentos de sopro. O som, no momento em que é criado, tem um volume alto (no momento em que o instrumentista sopra), que rapidamente cai e se mantém constante (enquanto o instrumentista mantém o ar fluindo). Após um

certo tempo, ele morre rapidamente (quando o instrumentista para de soprar). A forma da envoltória é a seguinte:

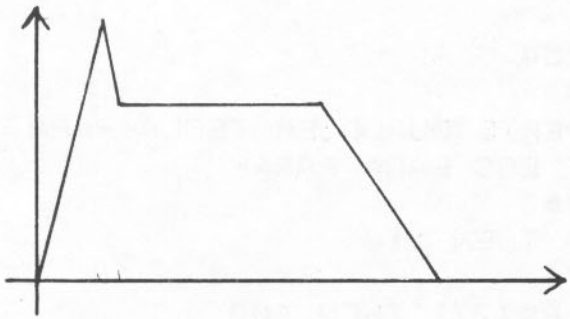


Figura 6.4 Envoltória de instrumento de sopro

Programa-exemplo SOPRO

```
50 SOUND Ø,254
60 SOUND 1,Ø
100 PRINT"APERTE QUALQUER TECLA PARA TOCAR"
105 PRINT "OU ESC PARA PARAR"
110 A$=INKEY$
120 IF A$="" THEN 11Ø
160 IF A$=CHR$(27) THEN 4ØØ
200 FOR N=15 TO 8 STEP -.2
210   SOUND 8,N
220 NEXT N
230 FOR N=1  TO 2ØØ
240   SOUND 8,8
250 NEXT N
260 FOR N=12 TO Ø STEP -.5
270   SOUND 8,N
280 NEXT N
300 GOTO 11Ø
400 SOUND 8,Ø
410 END
```

As linhas 200 a 220 fazem com que o som se inicie com um volume 15 e caia até 8. Nas linhas 230 a 250 ele se mantém constante em 8 por um tempo e, então, cai até 0, controlado pelas linhas 260 a 280.

6.2. A REPRESENTAÇÃO ADSR

No último programa-exemplo (SOPRO) vimos um som característico dos instrumentos de sopro. Ele é um bom exemplo de um evento sonoro que pode ser representado por uma envoltória ADSR.

Esta sigla é uma abreviatura de "Attack-Decay-Sustain-Release". A tradução literal seria "Ataque-Decaimento-Sustentação-Soltura". No entanto, no Brasil, só a tradução **ataque** é usada pelos músicos. As outras três mantêm-se em inglês.

As envoltórias ADSR são todas aquelas que podem ser divididas em quatro etapas: uma primeira onde o volume parte do zero e chega a um pico (o *ataque*), uma segunda onde o volume cai desse pico até um nível intermediário (o *decay*), uma terceira onde o volume permanece constante ou cai um pouco (o *sustain*) e uma quarta onde o volume cai desse patamar constante até o zero (o *release*).

Na verdade, o *sustain* e o *release* são um pouco mais complicados que isto em um sintetizador real: ambos estão condicionados ao tempo em que a tecla do sintetizador está acionada. Mas, no nosso caso, vamos usar a versão simplificada de ADSR apresentada acima.

Uma representação gráfica de uma envoltória ADSR está na Figura 6.5.

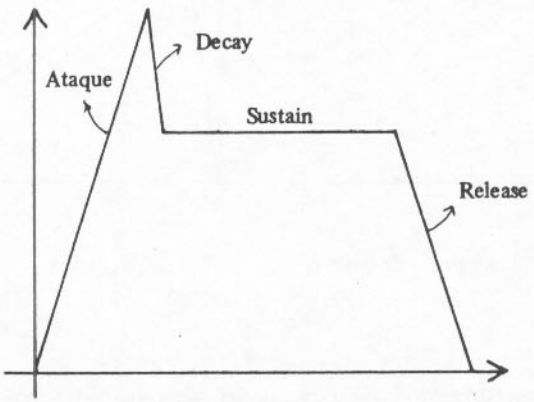


Figura 6.5 Envoltória ADSR

Nem todas as envoltórias podem ser representadas apenas com os quatro parâmetros das envoltórias ADSR. Como exemplo de uma envoltória não-ADSR temos o vibrato. O vibrato não pode ser representado como um simples "subir-e-depois-descer". Ele é uma envoltória cíclica, impossível de ser representada por quatro segmentos de reta.

No entanto, a maioria dos instrumentos acústicos tem envoltórias que podem ser aproximadamente representadas pelo modelo ADSR.

A envoltória da corda do violão, que já estudamos, pode ser expressa, em termos de ADSR, da seguinte maneira: *ataque* instantâneo (o volume vai de zero até o pico instantaneamente), *decay* bem lento, sem *sustain* nem *release*.

A envoltória ADSR do instrumento de sopro que acabamos de ver é a seguinte: *ataque* instantâneo, *decay* rápido, *sustain* razoavelmente longo e *release* rápido.

Os instrumentos de percussão são, geralmente, representados da seguinte maneira: *ataque* rápido, *decay* rápido, sem *sustain* e *release* rápido ou lento, dependendo do instrumento. Uma caixa de bateria tem um release rápido, o som morre rapidamente. Já um tímpano ou um prato de bateria tem um *release* bem lento, o som demora a morrer (Figura 6.6).

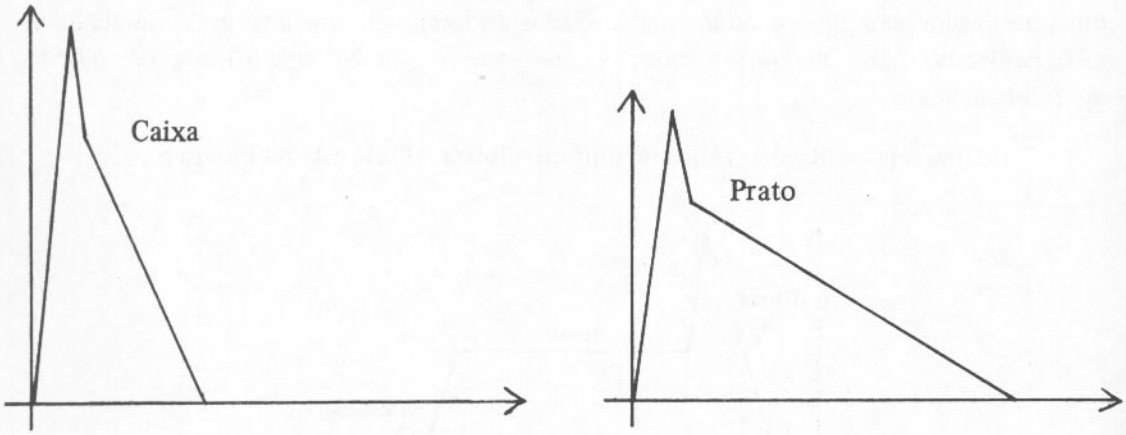


Figura 6.6 Envoltórias de caixa e prato de bateria

O caso dos instrumentos de percussão será tratado mais longamente quando falarmos do programador de ritmos.

O próximo programa-exemplo é exatamente um controlador de envoltória ADSR. Para cada um dos quatro parâmetros (*Ataque, Decay, Sustain* e *Release*), dois números são necessários: a velocidade do parâmetro e a sua duração.

Não basta simplesmente dizer, por exemplo, que o *sustain* "demora pouco" ou que ele "sobe rápido". Os dois números são necessários.

A velocidade indica a "inclinação" da reta. Quanto maior o número, mais inclinada a reta.

A duração indica por quanto tempo o som estará nesse parâmetro. Quanto maior o número, mais tempo o som fica dentro do parâmetro.

Exemplo: quanto maior a duração do ataque, mais tempo ele demorará. A **rapidez** com que o som sobe durante esse tempo de ataque especificado, no entanto, é especificada pela velocidade do ataque.

Programa-exemplo ADSR

```
60 DIM VE(3),TE(3)
130 SOUND 8,0:SOUND 0,0:SOUND 1,3
150 ON INTERVAL=4 GOSUB 5000
430 VE(1)=-VE(1):VE(2)=-VE(2):VE(3)=-VE(3)
450 CLS:PRINT"SIMULADOR DE ENVOLTORIA"
460 PRINT
500 INPUT"VELOC. DE SUBIDA DO ATAQUE";VE(0)
510 INPUT"TEMPO DE ATAQUE";TE(0)
520 INPUT"VELOC. DE DESCIDA DO DECAY";VE(1)
525 VE(1)=-VE(1)
530 INPUT"TEMPO DE DECAY";TE(1)
540 INPUT"VELOC. DE DESCIDA DO SUSTAIN";VE(2)
545 VE(2)=-VE(2)
550 INPUT"TEMPO DE SUSTAIN";TE(2)
560 INPUT"VELOC. DE DESCIDA DO RELEASE";VE(3)
565 VE(3)=-VE(3)
570 INPUT"TEMPO DE RELEASE";TE(3)
600 V=0   ' VOLUME INICIAL = 0
650 NI=0 ' NIVEL = 0 (ATAQUE)
```

```
670 T=0  ' TEMPO DESTE NIVEL = 0
700 TT=0 ' TEMPO TOTAL = 0
1000 SCREEN 2,0,0:OPEN "GRP:" FOR OUTPUT AS #1
1010 PSET(10,150)
1020 T=T+1
1025 TT=TT+1
1030 V=V+VE(NI)
1040 IF V>15 THEN V=15
1050 IF V<0 THEN V=0
1060 IF T<TE(NI) THEN 1100
1070   T=0
1080   NI=NI+1
1090   IF NI<4 THEN IF TE(NI)=0 THEN 1080
1100 LINE -(10+TT*5,150-V*10)
1110 IF NI<4 THEN 1020
1200 T=0
1210 NI=0
1220 V=0
1230 TT=0
1800 PSET(10,160):PRINT#1,"APERTE QUALQUER TECLA"
1810 IF INKEY$="" THEN 1810
2000 INTERVAL ON
2005 IF NI<4 THEN 2005
2010 INTERVAL OFF
2020 SOUND 8,0
2050 CLOSE 1
2500 INPUT"REPETIR O SOM";A$
2510 IF LEFT$(A$,1)="S" OR LEFT$(A$,1)="s" THEN 600
2600 INPUT"PROGRAMAR NOVA ENVOLTORIA";A$
2610 IF LEFT$(A$,1)="S" OR LEFT$(A$,1)="s" THEN 430
2700 CLS:END
5000 INTERVAL OFF
5003 IF NI>3 THEN 5200
5005 SOUND 8,V
5010 LINE(9+TT*5,149-V*10)-(11+TT*5,151-V*10),1,BF
5030 TT=TT+1
5040 T=T+1
5050 V=V+VE(NI)
5060 IF V>15 THEN V=15
5070 IF V<0 THEN V=0
5100 IF T<TE(NI) THEN 5200
5120   T=0
```

```
5130   NI=NI+1
5150   IF NI<4 THEN IF TE(NI)=Ø THEN 513Ø
5200 INTERVAL ON
5210 RETURN
```

O programa tem um funcionamento muito simples. Basta introduzir os valores de velocidade e duração para cada um dos quatro parâmetros (A, D, S e R).

O programa tem outra característica interessante: o gráfico da envoltória é mostrado após a programação e conforme o som vai sendo produzido.

Para construir a envoltória, o programa lança mão de um recurso excelente do BASIC-MSX: a instrução INTERVAL.

O comando da linha 150 faz com que a cada 1/15 de segundo a sub-rotina 5000 seja executada. Para ativar esse processamento periódico, a instrução INTERVAL ON tem de ser executada, como na linha 2010. Executando INTERVAL OFF, você estará desligando esse efeito.

A rotina 5000 é exatamente a que processa o envelope, aumentando ou diminuindo o som. Ela sabe em que parâmetro do envelope estamos por meio do valor da variável NI (nível). Essa variável assume os valores: 0 (para *ataque*), 1 (para *decay*), 2 (para *sustain*), 3 (para *release*) e 4 (para nenhum som). Os valores de velocidade e tempo para cada parâmetro estão nos vetores VE e TE, respectivamente.

Na Figura 6.7 estão os valores de velocidade e duração para cada um dos parâmetros das envoltórias já vistas: corda, sopro e percussão.

Lembre-se, no entanto, que os sons que você vai ouvir com esse programa são sons que têm uma envoltória, e somente uma envoltória, semelhante ao som do instrumento. Isso garante um som apenas vagamente semelhante ao som original, pois apenas a envoltória e a duração do evento sonoro estão sendo controladas. Os outros dois parâmetros (timbre e afinação) não estão sendo controlados aqui. Você aprendeu a controlá-los nas seções anteriores. Todos os controles serão unidos num só programa no nosso primeiro projeto, logo mais à frente.

Corda:

Ataque: V=15, T=1
Decay: V=1, T=15
Sustain: V=0, T=0
Release: V=0, T=0

Sopro:

Ataque: V=15, T=1
Decay: V=2, T=3
Sustain: V=0, T=20
Release: V=1, T=9

Percussão:

Ataque: V=15, T=1
Decay: V=3, T=2
Sustain: V=0.5, T=20
Release: V=0, T=0

Figura 6.7 Valores para Corda, Sopro, Percussão

6.3 AS ENVOLTÓRIAS PRÉ-PROGRAMADAS DO MSX

Até agora, construímos nossas envoltórias alterando gradualmente e diretamente o registrador de controle de volume. Quando o controle de volume não é alterado, temos uma envoltória de volume constante.

Essa é a maneira como o PSG trabalha normalmente: o volume permanece constante até que um novo valor de volume seja programado.

No entanto, o PSG já tem, embutido, um controlador de envoltória. Se esse controle de envoltória for ativado, o volume será variado automaticamente. Não é necessário que o usuário varie o controle de volume por meio do programa.

Você deve estar imaginando por que então complicamos tanto o assunto, fazendo um controlador de envoltória ADSR, que age diretamente sobre o controle de volume. Por que não usamos logo o controle automático de volume?

A resposta é:

O controle automático de envoltória do PSG é simples demais para uso em Música.

Além dos tipos de envoltória serem muito simples, apenas uma pode ser selecionada de cada vez. Todos os três canais têm de ter, obrigatoriamente, a mesma envoltória automática.

Todos esses problemas podem ser contornados com o controle explícito da envoltória.

De qualquer modo, vejamos como fazer para utilizar a envoltória pré-programada, que pode servir como envoltória de "emergência" quando não estamos com paciência ou tempo suficiente para desenvolver a envoltória manual ou para aplicações onde a simplicidade desse controle seja adequada.

Em primeiro lugar, vejamos como selecionar uma dada envoltória dentre as oito oferecidas pelo PSG. O registrador que seleciona o tipo de envoltória é o 13. Dependendo do valor que for colocado neste registrador teremos um tipo. Na Figura 6.8 temos todos os oito tipos de envoltória oferecidos.

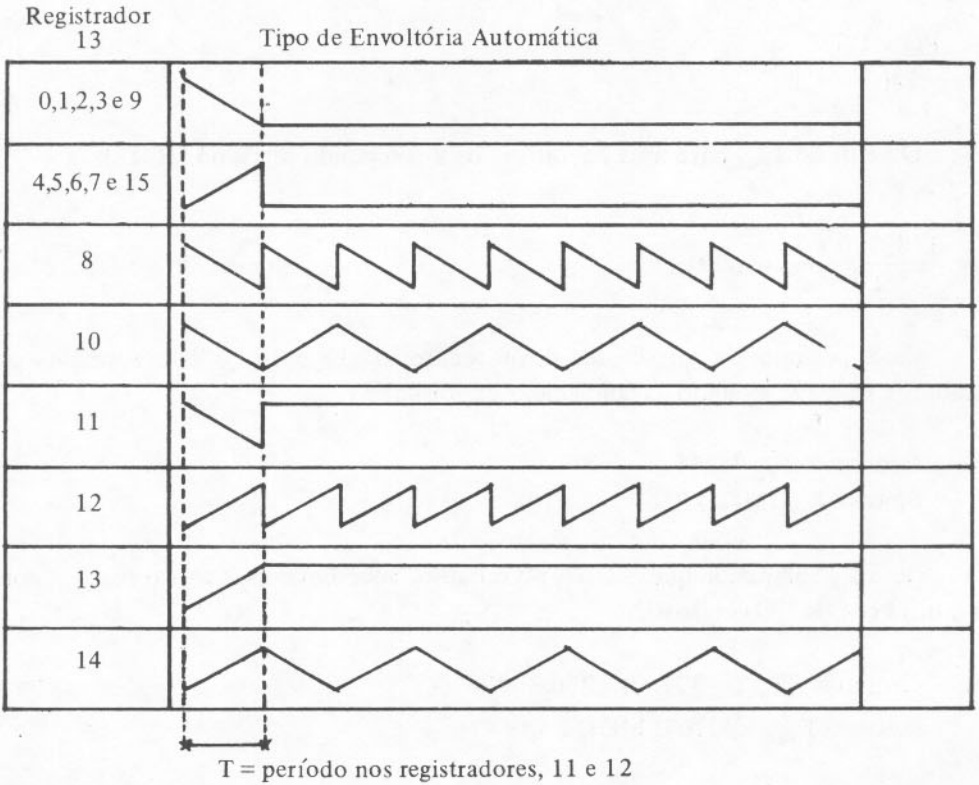


Figura 6.8 Envoltórias automáticas do MSX

Como podemos perceber, nenhuma das envoltórias seria adequada para reproduzir as que já geramos manualmente.

As envoltórias cíclicas (valor no registrador = 10 ou 14) se prestam para simular um vibrato.

Depois de selecionado o formato da envoltória temos de especificar a duração, assim como no programa-exemplo ADSR. No gerador do PSG temos, no entanto, formatos de envoltória com apenas um segmento de reta ou com vários segmentos de mesmo comprimento (as duas envoltórias cíclicas). Apenas uma duração deve, portanto, ser especificada. A duração é especificada nos registradores 11 e 12.

O registrador 11 é o registrador de ajuste grosso e o 12 é o registrador de ajuste fino. Cada um pode receber um valor de 0 a 255.

A unidade de tempo é 256 vezes o período do clock. Sendo T o período da envoltória em segundos e T_{msx} o período para os registradores do PSG, temos:

$$T_{msx} = \frac{FC \times T}{256}$$

O valor de T_{msx} para uma envoltória de um segundo seria, no HOT-BIT:

$$T_{msx} = \frac{1789770 \times 1}{256} = 6991$$

Esse período da envoltória deve ser repartido entre o par formado pelos registradores 11 e 12, segundo as conhecidas expressões:

Grosso = $T_{msx} \backslash 256$
Fino = T_{msx} MOD 256

Se T_{msx} for maior que 32767, no entanto, você deve usar as expressões abaixo **ou terá um erro de "Overflow".**

Grosso = $(T_{msx} - 32767) \backslash 256 + 128$
Fino = $(T_{msx} - 32767)$ MOD 256

Além de programar o tipo de envoltória e a sua duração, temos de programar a

freqüência do canal e colocar o volume em 16. Esse último valor é obrigatório.

> Se o volume do canal tiver um valor menor que 16, a envoltória não será automática.

O valor 16 no registrador de volume é quem indica que estamos usando uma envoltória automática e não o controle manual de volume.

O próximo programa-exemplo (ENVPSG) permite que você experimente as envoltórias pré-fabricadas do PSG. Para utilizá-lo, você deve introduzir os valores de tipo de envoltória (0 a 15) e a duração da envoltória (0 a 65535).

Programa-exemplo ENVPSG

```
50 SOUND Ø,254
60 SOUND 1,Ø
150 INPUT"NUMERO DA ENVOLTORIA (Ø A 15)";EN
160 INPUT"PERIODO DA ENVOLT. (Ø A 65535)";PE
170 IF PE>32767 THEN 23Ø
180 SOUND 11,PE MOD 256
190 SOUND 12,PE \ 256
200 GOTO 28Ø
230 SOUND 11,(PE-32767) MOD 256
240 SOUND 12,(PE-32767) \ 256 + 128
280 SOUND 13,EN
290 SOUND 8,16
300 PRINT"APERTE QUALQUER TECLA PARA PARAR"
310 IF INKEY$=""THEN 31Ø
320 SOUND 8,Ø
400 INPUT"OUTRA ENVOLTORIA";A$
450 IF LEFT$(A$,1)="S" OR LEFT$(A$,1)="s" THEN 15Ø
600 END
```

As linhas 50 e 60 programam a freqüência para o Lá central. O período é programado nas linhas 180 e 190 ou nas linhas 230 e 240, conforme ele for maior ou menor que 32767. A linha 280 programa o tipo do envelope, e a linha 290 habilita a execução do envelope.

CAPÍTULO 7

RUÍDO

Se você leu a parte referente ao PSG no manual do seu MSX, deve ter visto que ele tem um gerador de ruído, chamado, por vezes, de gerador de *noise* ou rumor.

Em primeiro lugar, vejamos qual é a definição de ruído.

> Consideramos "ruído" como sendo o som que não tem uma afinação precisa.

Todo o som que não pode ser caracterizado como sendo um Lá ou Dó ou qualquer nota definida é tecnicamente definido como sendo um ruído.

Mais tecnicamente um pouco: qualquer forma de onda que não tenha um período de oscilação identificável é um ruído.

Veja a Figura 7.1. Nela temos a representação de uma oscilação. Compare essa figura com qualquer uma das formas de onda da Figura 5.1.

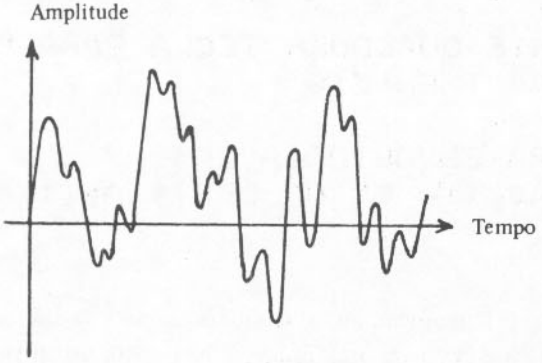


Figura 7.1 Ruído

Note como, no ruído da Figura 7.1, você não é capaz de identificar o vai-e-vem constante que caracteriza o som afinado.

Para que um som seja afinado, ele tem de ter uma freqüência precisa. Para que ele tenha uma freqüência precisa, deve ser possível identificar o vai-e-vem, e o tempo de sua oscilação deve poder ser medido.

Como exemplo de eventos sonoros cotidianos que contêm ruído ou que são exclusivamente ruído temos: batida na madeira, sons de motor, o som do vento etc.

Pense agora no vento soprando por uma fresta de janela, como nos filmes de terror. Quanto mais forte ou mais fraco o vento, podemos perceber um certo tipo de "afinação" no ruído. Quanto mais forte o vento, mais "agudo" fica o ruído. Apesar de não podermos identificar qualquer nota precisa nesse ruído, podemos comparar ruídos e dizer: "este ruído é mais agudo (ou mais grave) que aquele outro".

E agora, como explicamos a aparente afinação de um som que, por definição, não pode ter afinação?

Em primeiro lugar, lembre-se de que um timbre é visto, pela análise de Fourier, como uma soma organizada de freqüências diferentes. O que acontece com o ruído, segundo esse ponto de vista, é que ele contém, na verdade, uma soma descontrolada de inúmeras freqüências diferentes.

Mesmo dentro das freqüências descontroladas, geralmente podemos especificar qual o espectro ou, em outras palavras, qual a faixa de variação dessas freqüências descontroladas. Essa faixa pode abranger freqüências mais altas ou mais baixas o que poderá fazer com que percebamos uma espécie de afinação no ruído.

Nenhum ruído é afinado, ou deixaria de ser ruído. Pode-se falar, no entanto, em "freqüência do ruído" como uma forma simplificada de se dizer "faixa média de freqüências que compõem o ruído".

É isso o que acontece com o vento que sopra na janela, por exemplo. Da mesma forma, podemos dizer que o som de um vidro quebrando é mais agudo do que o som de um trovão. Estamos, na verdade, falando de uma média de freqüências componentes.

Para que alguém pode querer, conscientemente, colocar ruído em música?

O problema está, mais uma vez, em considerarmos o ruído como sendo não-musical.

O ruído pode perfeitamente ser musical. Toda percussão tem componentes de

ruído, isso quando não é completamente composta dele.

Não só a percussão mas outros instrumentos musicais têm componentes de ruído. Uma flauta, no momento exato em que é soprada, tem componentes de ruído que no instante seguinte deixam de existir. É a intensidade do primeiro sopro que produz essas componentes.

A rigor, qualquer som produzido naturalmente contém ruído, em maior ou menor quantidade.

Em um sistema de produção de eventos sonoros não pode faltar a capacidade de gerar ruído e controlar sua faixa de freqüência.

O PSG é capaz de ambos e o acesso a essa característica é conseguido através dos registradores 6 e 7. Em primeiro lugar deve-se selecionar a freqüência do ruído com o registrador 6, depois habilitar a saída de ruído com o valor correto no registrador 7 (chamado de registrador mixer). Por fim, temos de programar o volume no registrador correspondente ao canal. Vejamos em detalhes.

A faixa de freqüências é escolhida da mesma forma que a afinação de um canal: através do período medido na unidade padrão do PSG (16 vezes o período do clock).

A expressão para o cálculo desse período é idêntica à da afinação:

$$TR_{msx} = \frac{FC}{16 \times FR}$$

Onde TR_{msx} é o período do ruído, FC é a freqüência do clock e FR é a freqüência de ruído desejada.

> O período do ruído é apenas um para os três canais e ele pode ter valores apenas na faixa 0 a 31.

Se quisermos, por exemplo, uma freqüência média de ruído (FR) da ordem de 7000 Hz devemos programar o MSX (aqui um HOT-BIT com FC = 1789770) com o valor:

$$TR_{msx} = \frac{1789770}{16 \times 7000} = 15{,}98$$

ou, aproximadamente, 16.

Para programar esse valor de freqüência de ruído, executamos a instrução SOUND.

```
SOUND 6,16
```

Mas isso não garante que o ruído seja produzido. Precisamos ainda programar o mixer e o volume.

"Mixer" é a palavra usada para designar o registrador número 7, cuja tradução literal é "misturador". O que ele faz é controlar como o som afinado deve ser misturado com o ruído, para cada um dos três canais.

Como já vimos, cada canal pode produzir um som afinado, cuja freqüência é controlada por um par de registradores. Por exemplo: o som afinado do canal B é controlado pelo par formado pelos registradores 2 e 3. Chamaremos, aqui, o som afinado de **"nota"**.

Vimos também que cada canal é capaz de emitir, além da nota, um ruído, que tem uma afinação única para os três canais, controlada pelo registrador 6.

O papel do mixer é programar cada canal separadamente para a produção de nota, ruído, ambos ou nenhum som.

Ele é programado segundo os valores mostrados na Figura 7.2

```
+1  — DESLIGA NOTA A
+2  — DESLIGA NOTA B
+4  — DESLIGA NOTA C
+8  — DESLIGA RUÍDO A
+16 — DESLIGA RUÍDO B
+32 — DESLIGA RUÍDO C
```

Figura 7.2 Programação do Mixer

Não falamos no mixer até agora porque quando o MSX é ligado, o mixer é imediatamente programado para a produção de notas e somente notas. Vejamos qual seria o valor a colocar no registrador 7 para que isso acontecesse.

Para que o ruído fosse desligado teríamos de somar os seguintes valores: 8

(desliga ruído A), 16 (desliga ruído B) e 32 (desliga ruído C). O total é 56.

A instrução abaixo, portanto, habilita apenas a produção de notas, nos três canais:

```
SOUND 7,56
```

A instrução para habilitar apenas ruído, nos três canais, seria:

```
SOUND 7,7
(1 + 2 + 4 = 7)
```

A instrução para habilitar notas e ruído nos três canais, seria:

```
SOUND 7,0
```

Vamos supor que o mixer já está programado para a produção de ruído no canal escolhido, com ou sem produção de notas, como desejado. Falta ainda um pequeno detalhe: o controle de volume no canal em que queremos o ruído.

Já vimos como fazê-lo. Basta colocar o valor de volume (0 a 15) no registrador 8, 9 ou 10, conforme o canal.

Um exemplo completo: programar o PSG para produzir um ruído (e apenas ruído) no canal B (e apenas neste canal), com freqüência média de 5000 Hz e volume 13.

Em primeiro lugar, o valor do período será (para o HOT-BIT) 1789770/(16x5000), que é aproximadamente igual a 22.

O valor a colocar no mixer para a produção de ruído apenas no canal B é: 1 (desliga nota A) + 2 (desliga nota B) + 4 (desliga nota C) + 8 (desliga ruído A) + 32 (desliga ruído C) que é igual a 47.

A seqüência de comandos SOUND será:

```
SOUND 6,22 (programa freq. ruído)
SOUND 7,47 (programa mixer para ruído em B)
SOUND 9,13 (programa volume canal B)
```

Para silenciar o canal B:

SOUND 9,0

Uma última nota interessante sobre o controle de ruído do PSG:

> O ruído não pode ser controlado pela instrução PLAY. Só a instrução SOUND é capaz disso.

Vejamos o programa-exemplo RUÍDO onde simulamos o som do vento entrando por uma fresta.

Programa-exemplo RUIDO

```
10 R=RND(-TIME)
100 PRINT"APERTE QUALQUER TECLA PARA PARAR"
110 FR=Ø
150 IF INKEY$<>"" THEN 1ØØØ
160 PRINT"FREQUENCIA DE RUIDO =";FR
170 SOUND 7,47
180 SOUND 9,13
250 IF FR<16 THEN 4ØØ
260 F2=INT(RND(8)*14)
270 FOR N=FR TO F2 STEP -.Ø1
280   SOUND 6,N
290 NEXT N
300 GOTO 5ØØ
400 F2=INT(RND(8)*14+16)
410 FOR N=FR TO F2 STEP .Ø1
420   SOUND 6,N
430 NEXT N
500 FR=F2
510 GOTO 15Ø
1000 SOUND 7,56
1010 SOUND 9,Ø
1020 END
```

Esse programa faz um vai-e-vem com a freqüência de ruído, caminhando para uma freqüência alta quando a variável FR indica uma freqüência baixa e vice-versa.

O mixer é programado com o valor 47 na linha 170, habilitando apenas a produção de ruído no canal B. Quando o programa é abandonado, a linha 1000 volta a habilitar a produção de notas para os três canais.

Isso encerra a parte referente a ruído e também a parte genérica sobre o PSG. Com o controle de envelope, o controle de afinação, o controle parcial de timbre e o controle de ruído, vamos partir para nosso primeiro projeto: o sintetizador.

CAPÍTULO 8

PROJETOS

8.1. O MSX COMO INSTRUMENTO MUSICAL

Nosso micro pode servir como um instrumento musical para execução em tempo real. É claro que ele jamais vai ser tão poderoso como um instrumento construído exclusivamente para esse fim, mas ele serve como bancada de teste e de aprendizado na geração de som.

Uma vantagem muito grande de se usar o micro como sintetizador é que você, ao construir um programa de simulação desse instrumento, terá de se preocupar com todos os detalhes de construção de um sintetizador, detalhes esses que poderiam lhe passar desapercebidos no dia-a-dia. Dentre esses detalhes de construção estão: flexibilidade (como fazer para que esse instrumento possa variar seu timbre, envelope etc.), resposta (como fazer para que o som seja produzido assim que o instrumentista pressione uma tecla), recursos (como colocar o controle de oitava, portamento etc.).

Além de todos esses problemas, existe um problema prático para o uso do MSX como instrumento: o teclado. O teclado do micro tende a imitar um teclado de máquina de escrever e não um teclado de piano. Torna-se difícil, portanto, simular um teclado musical nele. Você jamais será um virtuose num MSX.

Outro problema sério de construção física do micro: os controles. Como fazer para simular os controles de um sintetizador?

Um instrumento real sempre vai ganhar essa parada. É sempre muito mais fácil e preciso girar um botão ou ligar uma chave num sintetizador do que executar uma seqüência de comandos de teclas em um micro.

Poderíamos incorporar no nosso projeto uma tela que simulasse o console de um sintetizador mas preferimos não fazê-lo.

> Toda a parte de processamento visual e interface com o usuário, mais elaborados, deste e dos outros projetos, fica a cargo do leitor.

A parte referente à comunicação com o usuário e os controles estão no nível mais simples possível.

Tomamos essa decisão por várias razões, mas a principal delas é a do tamanho dos programas.

Por experiências anteriores, sabemos que a parte de interface com o usuário, em programas desse tipo, toma cerca de **80%** do programa em BASIC. Apenas os restantes 20% são efetivamente ligados ao processamento do som. Um exemplo: no programa de simulação de instrumento TOQUE!, por nós desenvolvido, apenas 22% das linhas do programa BASIC são dedicadas a funções realmente essenciais de geração e controle do som.

As versões dos programas com os outros 70% de processamento visual e interface com o usuário estão disponíveis em fita.

O primeiro projeto é o sintetizador SINT.

Programa-exemplo SINT

```
60 DIM VE(3),TE(3),FR(16),RE%(16,5)
70 FOR N=0 TO 16:READ FR(N):NEXT N
80 DATA 855,807,762,719,679,641,605,571,539,508,480,453,428,404,381,360,339
90 TC$="AWSEDFTGYHUJKOLPÇ"
100 RT=1
130 SOUND 8,0:SOUND 9,0:SOUND 10,0
140 SOUND 0,0:SOUND 1,0:SOUND 2,0:SOUND 3,0:SOUND 4,0:SOUND 5,0
150 ON INTERVAL=6 GOSUB 5000
430 VE(1)=-VE(1):VE(2)=-VE(2):VE(3)=-VE(3)
450 CLS:PRINT"SINTETIZADOR"
460 PRINT
500 INPUT"VELOC. DE SUBIDA DO ATAQUE";VE(0)
510 INPUT"TEMPO DE ATAQUE";TE(0)
520 INPUT"VELOC. DE DESCIDA DO DECAY";VE(1)
525 VE(1)=-VE(1)
530 INPUT"TEMPO DE DECAY";TE(1)
540 INPUT"VELOC. DE DESCIDA DO SUSTAIN";VE(2)
545 VE(2)=-VE(2)
550 INPUT"TEMPO DE SUSTAIN";TE(2)
560 INPUT"VELOC. DE DESCIDA DO RELEASE";VE(3)
565 VE(3)=-VE(3)
570 INPUT"TEMPO DE RELEASE";TE(3)
600 V=0  ' VOLUME INICIAL = 0
```

```
650 NI=Ø ' NIVEL = Ø (ATAQUE)
670 T=Ø  ' TEMPO DESTE NIVEL = Ø
700 TT=Ø ' TEMPO TOTAL = Ø
800 INPUT"RELACAO VOLUME FUND/HARM 1";V1
805 INPUT"RELACAO FREQ FUND/HARM 1";R1
810 INPUT"RELACAO VOLUME FUND/HARM 2";V2
815 INPUT"RELACAO FREQ FUND/HARM 2";R2
830 INPUT"RELACAO DE TOM";RT
840 INPUT"RUIDO (S/N)";A$
850 IF LEFT$(A$,1)="S" OR LEFT$(A$,1)="s" THEN 880
860 SOUND 7,56
870 GOTO 930
880 SOUND 7,48
890 INPUT"PERIODO DO RUIDO (Ø-31)";TR
900 SOUND 6,TR
930 FOR N=Ø TO 16
950   RE%(N,Ø)=FR(N)\RT MOD 256
955   RE%(N,1)=FR(N)\RT \ 256
960   RE%(N,2)=FR(N)\RT\R1 MOD 256
965   RE%(N,3)=FR(N)\RT\R1 \ 256
970   RE%(N,4)=FR(N)\RT\R2 MOD 256
975   RE%(N,5)=FR(N)\RT\R2 \ 256
990 NEXT N
1000 SCREEN 2,Ø,Ø:OPEN "GRP:" FOR OUTPUT AS #1
1010 PSET(1Ø,75)
1020 T=T+1
1025 TT=TT+1
1030 V=V+VE(NI)
1040 IF V>15 THEN V=15
1050 IF V<Ø THEN V=Ø
1060 IF T<TE(NI) THEN 11ØØ
1070   T=Ø
1080   NI=NI+1
1090   IF NI<4 THEN IF TE(NI)=Ø THEN 1Ø8Ø
1100 LINE -(1Ø+TT*2,75-V*5)
1110 IF NI<4 THEN 1020
1200 T=Ø
1210 NI=4 ' PARA NAO GERAR SOM
1220 V=Ø
1230 TT=Ø
1300 PSET(1Ø,8Ø):PRINT#1,"TECLADO :"
1310 PSET(1Ø,95):PRINT#1,"  W E   T Y U   O P"
1320 PSET(1Ø,1Ø3):PRINT#1," A S D F G H J K L Ç"
1330 PSET(2Ø,123):PRINT#1,"ESC PARA TERMINAR"
1500 INTERVAL ON
1800 A$=INKEY$
1810 IF A$="" THEN INTERVAL ON:GOTO 18ØØ
1820 INTERVAL OFF
1830 N=INSTR(TC$,A$)-1
1850 IF N>=Ø THEN 19ØØ
1860 IF A$=CHR$(27) THEN 2ØØØ
1870 INTERVAL ON
1880 GOTO 18ØØ
1900 SOUND 8,Ø:SOUND 9,Ø:SOUND 1Ø,Ø
1950 SOUND Ø,RE%(N,Ø)
1955 SOUND 1,RE%(N,1)
1960 SOUND 2,RE%(N,2)
1965 SOUND 3,RE%(N,3)
1970 SOUND 4,RE%(N,4)
1975 SOUND 5,RE%(N,5)
```

```
1980 SOUND 8,VE(0):SOUND 9,VE(0)/V1:SOUND 10,VE(0)/V2
1985 T=0:V=0:NI=0 ' PODE PRODUZIR SOM
1995 GOTO 1800
2000 INTERVAL OFF
2020 SOUND 8,0:SOUND 9,0:SOUND 10,0
2050 CLOSE 1
2500 INPUT"REPETIR O SOM";A$
2510 IF LEFT$(A$,1)="S" OR LEFT$(A$,1)="s" THEN 600
2600 INPUT"PROGRAMAR NOVO SOM";A$
2610 IF LEFT$(A$,1)="S" OR LEFT$(A$,1)="s" THEN 430
2700 CLS:END
5000 INTERVAL OFF
5003 IF NI>3 THEN 5300
5040 T=T+1
5050 V=V+VE(NI)
5060 IF V>15 THEN V=15
5070 IF V<0 THEN V=0
5080 SOUND 8,V:SOUND 9,V/V1:SOUND 10,V/V2
5100 IF T<TE(NI) THEN 5200
5120   T=0
5130   NI=NI+1
5150   IF NI<4 THEN IF TE(NI)=0 THEN 5130
5200 INTERVAL ON
5210 RETURN
5300 SOUND 8,0:SOUND 9,0:SOUND 10,0
5310 INTERVAL ON
5320 RETURN
```

Esse programa é simplesmente a união das idéias apresentadas sobre síntese de som no MSX. Ele compreende os programas PIANO2 e ADSR com controles de ruído e oitava.

As perguntas feitas inicialmente são as mesmas do ADSR, referentes aos quatro parâmetros da envoltória.

As perguntas seguintes são referenes à freqüência e ao volume **relativos** das ondas a serem somadas. A freqüência e o volume de cada harmônico serão proporcionais à freqüência e ao volume da fundamental.

A pergunta seguinte, "RELAÇÃO DE TOM" refere-se a uma constante para afinar ou mudar de tom **todas** as freqüências tocadas. Todas as freqüências serão multiplicadas por essa constante. Você pode usar esse parâmetro para subir ou descer uma oitava, dando-lhe valores que são potências de 2.

Em seguida é perguntado se você deseja ruído no som final. Se você responder que sim, terá de informar o período do ruído.

As perguntas poderiam ser respondidas assim (entre parênteses estão alguns comentários):

VELOCIDADE DE SUBIDA DO ATAQUE? 15
TEMPO DE SUBIDA DO ATAQUE?1

(*ataque* rápido)

VELOCIDADE DE DESCIDA DO DECAY? 3

TEMPO DE DESCIDA DO DECAY? 1

(*decay* rápido)

VELOCIDADE DE DESCIDA DO SUSTAIN? 1
TEMPO DE DESCIDA DO SUSTAIN? 10

(*sustain* lento)

VELOCIDADE DE DESCIDA DO RELEASE? 2
TEMPO DE DESCIDA DO RELEASE? 1

(*release* rápido)

RELAÇÃO VOLUME FUND/HARM 1? 1

(volume do harmônico 1 = volume da fundamental)

RELAÇÃO FREQ FUND/HARM 1 ? 2

(a freqüência do harmônico a ser tocado no canal B é igual a 2 x freqüência da fundamental, é o segundo harmônico.)

RELAÇÃO VOLUME FUND/HARM 2 ? 2

(volume do harmônico 2 = 2 x volume da fundamental)

RELAÇÃO FREQ FUND/HARM 2 ? 4

(harmônico 2 é o quarto harmônico)

RELAÇÃO DE TOM ? 2

(o teclado vai estar uma oitava acima da afinação original)

RUÍDO (S/N) ? S
FREQÜÊNCIA DO RUÍDO ? 13

As dúvidas nesse projeto podem ser esclarecidas com facilidade. Basta você olhar as listagens dos programas PIANO2 e ADSR, que são os módulos que compõem este programa.

8.2. O MSX COMO PROGRAMADOR DE RITMOS

Nos capítulos anteriores, já mencionamos algumas características normalmente atribuídas aos instrumentos de percussão. Vamos falar um pouco mais sobre eles.

O que é um instrumento de percussão?

Como o próprio nome já diz, são os instrumentos cujo som é produzido pelo percutir ou chocar de um objeto com outro.

Um tamborim, por exemplo, soa quando a baqueta bate (percute) contra o couro.

Se formos levar a definição ao pé da letra, poderemos concluir que o piano também é um instrumento dessa categoria, pois nele há uma percussão de um martelo contra uma corda, e que o baixo elétrico também pode se comportar dessa maneira, quando o baixista funk bate na corda com o lado do polegar.

A definição não é, no entanto, interpretada tão literalmente. São usualmente considerados instrumentos de percussão, a bateria, o pandeiro, o xilofone e todos os outros onde o percutir é o ato mais característico da execução.

O ato de percutir alguma coisa tem um efeito interessante na envoltória do som produzido. A intensidade cresce muito rapidamente até um pico como efeito da batida (*ataque* instantâneo) e imediatamente cai um pouco quando o sistema "se recupera do choque" (*decay* rápido e curto). O *sustain* e o *release* variam de instrumento para instrumento.

A envoltória de um instrumento de percussão é, portanto, semelhante à da Figura 8.1.

Além da envoltória característica, todo instrumento de percussão tem ruído, de maior ou menor freqüência.

Nosso próximo projeto é o mais extenso desse livro. Isto se deve ao fato de ele ser um programador de ritmos de três vozes. Um programa desse tipo **sempre** vai exigir um editor que, por sua vez, vai exigir uma interface com o usuário de maior complexidade (movimentação de cursor, atualização de tela, comandos, checagem de erros etc.).

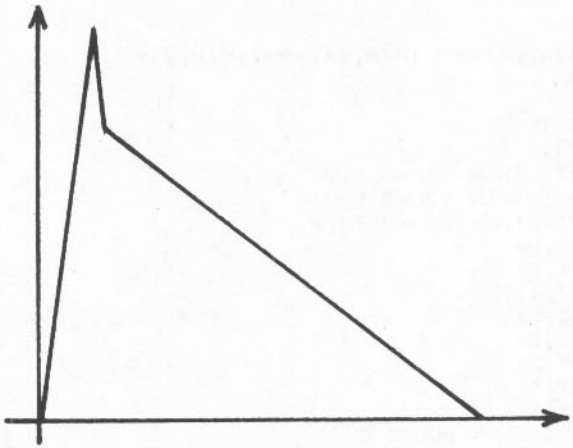


Figura 8.1 Envoltória da percussão

O programa leva o nome RITMO

Programa-exemplo RITMO

```
10 CLEAR 5000:DEFINT A-Z
20 DIM IT(63,2),FF(63,2),FG(63,2),AR(63),MX(63),VL(63,2),PR(5)
30 DIM PF(5),PG(5),PD(5),PM(5),CB$(10),MP(10)
100 MP=0:IT$=" QWERT"
110 FOR N=1 TO 5
120  READ PR(N),PF(N),PG(N),PD(N),PM(N)
130 NEXT N
150 FOR N=0 TO 10
160  READ CB$(N),MP(N)
170 NEXT N
510 CP=0:TP=8
520 GOSUB 6000
700 GOSUB 2000
710 INTERVAL ON
1000 GOTO 1000
1997 '--------
1998 ' COMPILA
1999 '--------
2000 FOR N=0 TO MP:FOR VZ=0 TO 2:VL(N,VZ)=0:NEXT VZ:NEXT N
2003 LOCATE 0,39:PRINT"AGUARDE...";
2005 FOR N=0 TO MP
2010  FOR VZ=0 TO 2
2020   IF IT(N,VZ)=0 THEN 2200
2030   V=12:K=0
2040    FF(N+K,VZ)=PF(IT(N,VZ))
2045    FG(N+K,VZ)=PG(IT(N,VZ))
2050    AR(N+K)=PR(IT(N,VZ))
2055    VL(N+K,VZ)=V
2080    V=V-PD(IT(N,VZ)):K=K+1
2090    IF V>0 AND K+N<=MP THEN 2040
```

```
2200  NEXT VZ
2220  MX(N)=PM(IT(N,Ø))+PM(IT(N,1))+PM(IT(N,2))
2300 NEXT N:RETURN
2997 '-------
2998 ' TOCA UM PONTO
2999 '-------
3000 IF VL(PT,Ø)=12 THEN SOUND 8,Ø
3002 IF VL(PT,1)=12 THEN SOUND 9,Ø
3004 IF VL(PT,2)=12 THEN SOUND 1Ø,Ø
3008 SOUND Ø,FF(PT,Ø)
3010 SOUND 1,FG(PT,Ø)
3020 SOUND 2,FF(PT,1)
3030 SOUND 3,FG(PT,1)
3040 SOUND 4,FF(PT,2)
3050 SOUND 5,FG(PT,2)
3100 SOUND 6,AR(PT)
3110 SOUND 7,MX(PT)
3115 IF VL(PT,Ø)=12 THEN SOUND 8,15
3120 IF VL(PT,1)=12 THEN SOUND 9,15
3125 IF VL(PT,2)=12 THEN SOUND 1Ø,15
3150 SOUND 8,VL(PT,Ø)
3160 SOUND 9,VL(PT,1)
3170 SOUND 1Ø,VL(PT,2)
3200 PT=PT+1:IF PT>FF THEN PT=II
3210 RETURN
3500 A$=INKEY$:IF A$="" THEN 3500
3510 I=INSTR(IT$,A$):IF I<2 THEN 3500
3520 MP=1Ø:IT(Ø,Ø)=I-1:GOSUB 2000
3530 ON INTERVAL=1Ø GOSUB 3000
3540 PT=Ø
3550 INTERVAL ON
3560 IF PT>9 THEN INTERVAL OFF:IT(Ø,Ø)=Ø:GOTO 3500
3570 GOTO 3560
3997 '--------
3998 ' DATA
3999 '--------
4000 DATA 5,20,Ø,6,Ø
4010 DATA 3,16,Ø,2,Ø
4020 DATA 20,200,2,3,Ø
4030 DATA 1Ø,50,2,3,Ø
4040 DATA 1,250,5,5,Ø
4100 DATA "!...:...:...:...!...:...:...:...",63
4110 DATA "!...:...:...:...!...:...:...",47
4120 DATA "!...:...:...:...!...:...:...:...",63
4130 DATA "!...:...:...:...:...",39
4140 DATA "!...:...:...:...!...:...:...",47
4150 DATA "!...:...:...:...:...:...:...",55
4160 DATA "!...:...:...:...!...:...:...:...",63
4170 DATA "!.....:.....:.....",35
4180 DATA "!...:...:...:...:...",39
4190 DATA "!.:.:.:.:.:.:.:.:.:.:.:.",43
4200 DATA "!.....:.....:.....:.....",47
4997 '------------
4998 ' IMPRIME A TELA DO EDITOR. CP E' A FORMULA DE COMPASSO
4999 '------------
5000 CLS:MP=MP(CP)
5010 PRINT:PRINT"    ";CB$(CP)
5020 FOR VZ=Ø TO 2
5030  PRINT HEX$(VZ+1);"!!";
5040  FOR N=Ø TO MP\2
```

```
5050   PRINT MID$(IT$,IT(N,VZ)+1,1);
5060  NEXT N
5070  PRINT
5080 NEXT VZ
5100 PRINT:PRINT:PRINT"    ";CB$(CP)
5120 FOR VZ=0 TO 2
5130  PRINT HEX$(VZ+1);"  ";
5140  FOR N=MP\2+1 TO MP
5150   PRINT MID$(IT$,IT(N,VZ)+1,1);
5160  NEXT N
5170  PRINT"!!"
5180 NEXT VZ
5200  GOSUB 5500
5290 RETURN
5497 '---------
5498 ' IMPRIME MARCAS
5499 '---------
5500 LOCATE 0,5:PRINT STRING$(39,32);:LOCATE 0,11:PRINT STRING$(39,32);
5550 IF II<=MP\2 THEN LOCATE 3+II,5 ELSE LOCATE 2+II-MP\2,11
5560 PRINT">";
5570 IF FF<=MP\2 THEN LOCATE 3+FF,5 ELSE LOCATE 2+FF-MP\2,11
5580 PRINT"<";
5590 RETURN
5997 '--------
5998 ' EDITOR
5999 '--------
6000 X=3:Y=2:PA=0:VO=0:CU$=CHR$(220):PM=20:PS=0
6010 GOSUB 5000:GOSUB 7000
6090 CG$=MID$(IT$,IT(PA,VO)+1,1)
6100 A$=INKEY$
6120 IF A$<>"" THEN 6200
6130  PS=PS+1
6140  LOCATE X,Y
6145  IF PS=PM\2 THEN PRINT CG$; ELSE IF PS>PM THEN PRINT CU$;:PS=0
6150  GOTO 6100
6200 LOCATE X,Y:PRINT CG$;
6220 I=INSTR(IT$,A$)
6230 IF I=0 THEN 6300
6240  IT(PA,VO)=I-1:GOTO 6090
6300 IF ASC(A$)<>28 THEN 6320
6303  X=X+1:PA=PA+1
6305  IF PA>MP THEN X=3:PA=0:Y=2+VO ELSE IF PA=MP\2+1 THEN X=3:Y=Y+6
6310  GOTO 6090
6320 IF ASC(A$)<>29 THEN 6340
6323  X=X-1:PA=PA-1
6325  IF PA<0 THEN X=3+MP\2:PA=MP:Y=Y+6 ELSE IF PA=MP\2 THEN X=3+MP\2:Y=Y-6
6330  GOTO 6090
6340 IF ASC(A$)<>31 THEN 6360
6343  Y=Y+1:VO=VO+1
6345  IF VO>2 THEN VO=0:IF Y=5 THEN Y=8:PA=PA+MP\2+1 ELSE Y=2:PA=PA-MP\2-1
6350  GOTO 6090
6360 IF ASC(A$)<>30 THEN 6380
6363  Y=Y-1:VO=VO-1
6365  IF VO<0 THEN VO=2:IF Y=1 THEN Y=10:PA=PA+MP\2+1 ELSE Y=4:PA=PA-MP\2-1
6370  GOTO 6090
6380 IF A$<>"X" THEN 6500
6390  GOSUB 2000
6400  ON INTERVAL=TP GOSUB 3000
6410  CLS:PRINT"PARA INICIAR APERTE A BARRA DE ESPACO."
6415  PRINT"PARA TERMINAR APERTE-A NOVAMENTE"
```

```
6420  PT=0
6430  IF INKEY$<>" " THEN 6430
6440  INTERVAL ON
6450  IF INKEY$<>" " THEN 6450
6460   INTERVAL OFF:SOUND 8,0:SOUND 9,0:SOUND 10,0
6470  GOTO 6000
6500 IF A$<>"I" THEN 6550
6510  II=PA:GOSUB 5500:GOTO 6090
6550 IF A$<>"F" THEN 6600
6560  FF=PA:GOSUB 5500:GOTO 6090
6600 IF A$<>"C" THEN 6650
6610  LOCATE 0,15:INPUT"FORMULA DE COMPASSO (2 A 12)";CP
6615  CP=CP-2:IF CP<0 OR CP>10 THEN BEEP:GOTO 6610
6620  MP=MP(CP):GOTO 6000
6650 IF A$<>"V" THEN 6700
6660  LOCATE 0,15:INPUT"VELOCIDADE (1 A 20)";TP
6665  TP=26-TP:IF TP<6 OR TP>25 THEN BEEP:GOTO 6660
6670  GOSUB 7000:GOTO 6090
6700 IF A$<>CHR$(27) THEN 6800
6710  CLS
6720  INPUT"VOCE TEM CERTEZA DE QUE QUER ABANDONAR  O PROGRAMA";A$
6730  IF LEFT$(A$,1)="S" THEN END
6740  GOTO 6000
6800 BEEP:GOTO 6090
6900 GOTO 6900
6997 '--------
6998 ' REIMPRIME LINHAS INFERIORES
6999 '--------
7000 LOCATE 0,15:PRINT STRING$(39,32);
7010 LOCATE 0,12:PRINT STRING$(39,32);
7020 LOCATE 0,12:PRINT"Velocidade =";26-TP;"     Compasso =";CP+2
7030 LOCATE 0,16:PRINT"INSTRUMENTOS : Q W E R T"
7040 LOCATE 0,17:PRINT"COMANDOS : <I>nicio, <F>im, <C>ompasso,"
7050 LOCATE 0,18:PRINT"             <V>elocidede, e<X>ecuta,"
7060 LOCATE 0,19:PRINT"             <ESC> para terminar"
7100 RETURN
```

O programa permite que o usuário utilize cinco instrumentos, designados pelas letras Q, W, E, R e T, em qualquer uma das três vozes, designadas pelos números 1, 2 e 3.

Tudo o que o usuário tem a fazer é "escrever a partitura" movendo o cursor por meio das setas e digitando a letra correspondente ao instrumento no lugar correto.

As letras podem ser apagadas com a barra de espaços.

No alto da tela estão as seguintes marcas: "!", que indica o primeiro tempo de um compasso; ":", que indica um tempo; e ".", que indica um quarto ou um sexto de tempo, dependendo da fórmula de compasso usada.

O usuário dispõe dos seguintes comandos:

I (marca Inicial)

Esse comando faz com que a marca inicial (indicada por ">" seja colocada na posição do cursor. As marcas inicial e final indicam onde se inicia e onde termina a repetição. A parte do ritmo que se repete é **apenas** aquela que está entre essas marcas.

Recomenda-se que a marca inicial seja colocada **sempre no início de um compasso (sinalizado por "!").** De preferência, coloque na primeira posição da tela (onde ela inicialmente está).

Se as marcas não forem preparadas, nenhum ritmo será tocado.

F (marca Final)

Esse comando faz com que a marca final (indicada por "<") seja colocada sob o cursor.

Recomenda-se que a marca final seja colocada **na posição imediatamente anterior ao início de um compasso** (na última fração de tempo do compasso anterior). De preferência, coloque na última posição da tela.

X (eXecuta)
Executa o ritmo editado.

V (Velocidade de execução)
Muda o andamento. O novo valor pode ser qualquer número de 1 a 20.

C (fórmula de Compasso)
Muda a fórmula de compasso. Dela dependem quantos tempos tem cada compasso e também quantos compassos ficam disponíveis para edição.

As fórmulas possíveis podem ser qualquer uma entre 2 e 12. Você pode, se achar interessante, construir um ritmo de 11 por 4 (e por que não?). Basta usar o valor 11 aqui.

Os valores comuns em música são: 2, 3, 4, 6 e 8. Os outros são raramente encontrados.

> Nas fórmulas de compasso usuais, indicadas acima, o usuário dispõe de quatro compassos (dois na linha de cima e dois na linha de baixo). Nas outras, estão disponíveis apenas dois compassos (um em cada linha).

ESC – abandona o programa.

Dentre todos os projetos, este é o que tem o funcionamento interno mais complexo.

Como já mencionamos antes, conforme a apresentação de um programa vai ficando mais e mais profissional, a parte referente ao processamento visual e comandos vai aumentando e, conseqüentemente, maior vai ficando o programa.

As linhas realmente responsáveis pelo controle e execução do som de bateria no programa RITMO são as linhas 2000 a 3570 (um quarto do programa). Todo o resto refere-se à inicialização e controle do editor de tela.

Sobre o editor (sub-rotina 6000), basta dizermos que ele prepara um ritmo que é guardado dentro da matriz IT(63,2). Cada posição da matriz guarda um número que representa um instrumento codificado. O primeiro índice da matriz indica o **ponto**, nome que estamos usando para indicar a menor fração de tempo musical que nosso programador de ritmos pode representar. No compasso 4/4, por exemplo, um ponto vale uma semicolcheia. O segundo índice indica o número do canal (0, 1 ou 2 para indicar A, B ou C, respectivamente) onde o instrumento deverá ser tocado.

O instrumento é codificado da seguinte maneira: Q=1, W=2, E=3, R=4, T=5. Se o valor de uma dada posição for 0, isso indica que nenhum som deverá ser **iniciado** neste canal, nesta fração de tempo.

Exemplo: O valor 3 na posição IT(41,1) indica que no ponto número 41 (a décima semicolcheia do terceiro compasso, se a fórmula for 4/4), no canal B, deverá se iniciar a produção do som do instrumento tipo 3, que corresponde ao indicado pela letra E, quando você programa o ritmo. Esta é a maneira pela qual o editor vê o ritmo escrito na tela. Esta estrutura de armazenagem seria, no entanto, extremamente ineficiente para a produção de som em tempo real.

A rotina que produz o som (3000 – TOCA UM PONTO) não pode perder tempo investigando que instrumento deve ser tocado e calculando os valores a colocar nos

diversos registradores do PSG. Para isso existe a rotina COMPILA (4000). Ela serve de ponte de ligação entre a estrutura de armazenagem do editor e a estrutura para execução em tempo real utilizada pela rotina 3000. Essa última estrutura é composta de cinco matrizes: FF(63,2) (freqüência fina de cada canal em cada ponto), FG(63,2) (freqüência grossa de cada canal em cada ponto), AR(63) (afinação do ruído em cada ponto), MX(63) (valor do mixer em cada ponto) e VL(63,2) (volume de cada canal em cada ponto). É fácil perceber que cada uma dessas matrizes corresponde a um registrador do PSG. Com essa estrutura de armazenagem, o trabalho da rotina 3000 fica bastante simplificado. Tudo o que ela tem a fazer quando for chamada a tocar os instrumentos de um ponto é colocar em cada registrador do PSG o valor da matriz correspondente.

A rotina COMPILA percorre a matriz IT e, para cada ponto, preenche o ponto correspondente de cada um das cinco matrizes de registradores do PSG acima apresentadas. Para saber como traduzir o código do instrumento na freqüência, ruído e demais parâmetros que indicam o som real do instrumento, a rotina utiliza os **vetores de descrição dos instrumentos**: PF(5) (freqüência fina de cada um dos cinco tipos de instrumento), PG(5) (freqüência grossa), PR(5) (afinação do ruído), PM(5) (valor do *mixer*) e PD(5) (*decay*). Esse último não corresponde a um registrador do PSG. Ele é a velocidade de *decay* e indica, indiretamente, durante quantos pontos o som deve ser soado e de quanto deve diminuir o volume de ponto para ponto.

Vamos voltar ao exemplo acima, do instrumento E no ponto 41, no canal B. Vejamos como a rotina COMPILA faz para preencher as matrizes de registradores do PSG correspondentes ao ponto 41.

O instrumento E é indicado na matriz IT como sendo tipo 3. Os valores dos vetores de descrição do instrumento número 3 são:

PR(3) (freq. ruído) = 20
PF(3) (afin. fina) = 200
PG(3) (afin. grossa) = 2
PD(3) (decay) = 3
PM(3) (mixer) = 0 (ruído/nota nos 3 canais)

Os dados acima estão no DATA da linha 4020.

O instrumento tipo 3 tem, portanto, ruído e som afinado ao mesmo tempo, tanto a freqüência do ruído como a da nota são baixas e o *decay* não é nem muito rápido nem muito lento.

A rotina COMPILA passa diretamente as constantes que estão nos vetores PR(3), PF(3), PG(3) e PM(3), respectivamente para as posições AR(41), FF(41,1), FG(41,1) e MX(41) (linhas 2040, 2045, 2050 e 2200). Os índices são 41 e 1 porque o ponto é o 41 e o canal é B.

O volume, que é guardado na posição VL(41,1) é inicialmente colocado em 12 (linhas 2030 e 1055).

O volume é decrementado do valor da velocidade de *decay* (ele fica valendo 9, portanto) e, se ele for maior que zero, todas as posições das matrizes serão repetidas no próximo ponto (42), exceto o volume (matriz VL), que recebe o novo valor decrementado.

O volume vai sendo decrementado e o "som vai sendo copiado" nos pontos seguintes, até que o valor do volume (variável V) chegue a zero. É assim que a envoltória é conseguida neste programa. Quando a rotina 3000 for executar os pontos em seqüência, ela executará o mesmo som durante vários pontos, mas com volume cada vez menor, o que dá a sensação de *decay*.

O ataque característico da percussão é conseguido nas linhas 3115 a 3125. No primeiro ponto de cada som, o valor do volume é 12. Essas três linhas fazem com que o valor 15 de volume seja tocado por um breve instante, antes do valor 12. Isso produz um pico de ataque.

8.3 O MSX COMO COMPOSITOR

Como já dissemos antes, música é um conjunto organizado de eventos sonoros. O tipo de organização e de sons usados é o que vai diferenciar um tipo de música de outro.

É por isso que podemos ter uma orquestra sinfônica tocando diferentes tipos de música. O conjunto de sons possíveis é sempre o mesmo: as notas que os diversos instrumentos da orquestra podem tocar. A **organização** desses sons é que muda, por exemplo, de Bach para Stravinsky.

Por outro lado, podemos ter instrumentações diferentes (conjuntos de sons diferentes) para aproximadamente a mesma organização. O Hino Nacional cantado por uma cantora de música popular e o Hino Nacional tocado por uma banda de música são, ambos, a mesma música, pois a organização não varia.

A regra não vale sempre mas podemos dizer que

o que caracteriza uma composição musical e o ato de compor é a maneira como os sons estão organizados e não os sons em si.

O que é a organização dos sons, afinal de contas? É "que som vem antes, depois e junto de outro" ou, simplificando bastante: "a ordem das notas".

Chegamos, então, a uma conclusão aparentemente óbvia e simplista: "compor é o ato de colocar notas em seqüência". Partindo disso podemos dizer que um bom compositor é o que sabe colocar as notas em uma boa seqüência.

Mas o que é uma "boa seqüência"?

A resposta é: ninguém sabe ao certo.

Não podemos responder essa pergunta com uma resposta clássica, que foi usada durante séculos: "uma boa seqüência é aquela que é agradável ao ouvido". Essa resposta, além de subjetiva demais, nega todo o trabalho do compositor Schoenberg, que passou a vida inteira tentando fazer com que essa resposta clássica fosse esquecida. Na opinião desse compositor, que revolucionou a música no início deste século, uma composição não deve ser necessariamente agradável ao ouvido, mas deve ter, obrigatoriamente, uma organização muito sólida. Em outras palavras, o ouvinte não deve perceber beleza, e sim organização. Foi a partir desse princípio que ele fundou seu método de composição, o dodecafonismo, e compôs diversas peças que não podem jamais ser qualificadas como "belas", em respeito a sua memória.

Então, se não estamos procurando "seqüências agradáveis de notas", o que estamos procurando como ponto e partida para organizar nossos sons e, portanto, compor?

Organização faz pensar em regras. Existem, então, regras para se organizar bem um conjunto de sons? Sim, elas existem, regras escritas e não-escritas, regras que são passíveis de serem colocadas claramente no papel e regras que não o são.

A maioria dos compositores usa regras que não podem ser expressas claramente porque nem os próprios compositores sabem exatamente que regras são essas. Eles compõem usando, muitas vezes, a "intuição" e vão criando, assim, um estilo próprio com regras próprias. Essas regras são reais e existem dentro da cabeça dos compositores, quer eles queiram, quer não, quer elas sejam explícitas e conscientes ou implícitas e inconscientes.

Algumas dessas regras, de tempos em tempos, tornam-se um consenso ou um estilo de um grupo de compositores. Pode acontecer, nesse caso, de algumas delas serem

explicitamente formuladas e colocadas no papel. Temos, então, as regras de harmonia, contraponto, as regras de composição dodecafônica e muitas outras.

Elas não são leis, nem, por outro lado, restrições absurdas a serem combatidas e também não são a garantia de que, quem as seguir à risca terá como resultado uma peça aceitável de composição musical. Elas são, simplesmente, a expressão de algumas características de um estilo particular de música. Se, por exemplo, seguirmos as regras de harmonia tradicional para uma determinada composição, teremos como resultado uma composição que tem características dos compositores que usavam a harmonia tradicional (mesmo sem o saber), como por exemplo, Bach.

Mas essas regras não são leis. Bach escreveu centenas de corais nos quais as regras da harmonia tradicional estão presentes de ponta a ponta, mas em *todos* eles há, pelo menos uma vez, em algum ponto, uma violação de alguma regra.

Mais uma vez, então, as regras representam apenas uma fração das regras de organização das notas, aquela fração que pode ou foi colocada no papel. As outras regras que fazem o gênio dos grandes compositores não podem ou não foram codificadas claramente.

E o computador, ele é capaz de compor? Sim, já que existem regras para compor. O computador é absolutamente incapaz de criar, hoje em dia, e provavelmente jamais poderá fazê-lo. O que ele faz, e faz bem, é: *seguir regras*.

O computador chegará a ser tão bom compositor quando um ser humano? Somente se *todas* as regras de composição forem explicadas. Isso se tais regras *puderem* ser explicitadas e traduzidas em uma linguagem adequada e precisa para o computador.

Para que, então, programar um computador para compor?

Para uma coisa, pelo menos: como incentivo para que as regras de composição sejam explicitadas ou descobertas. Se a composição que resulta de um programa de computador é ruim, a culpa não é do computador. Ele é uma máquina muito obediente mas incapaz de criar. Para melhorar a composição que o computador produz, o compositor tem de se esforçar para descobrir como ele próprio compõe. E o beneficiário desse esforço não é o computador nem o público (se é que algum público vai sair de casa para ouvir uma composição totalmente produzida por um computador), mas o próprio compositor e outros compositores, que terão novas regras explicitadas, novas teorias de composição. O método de composição desse compositor (pelo menos a parte explicitável desse método) pode agora ser ensinado a novos compositores.

Muitas críticas são feitas à arte feita por computador e até mesmo à arte

auxiliada por computador, aquela onde o processo de criação é totalmente executado pelo artista mas o trabalho artístico final necessita de tamanho volume de dados que um computador tem de ser usado. A crítica mais severa é justamente a de que o artista é obrigado a explicitar um método até então inconsciente. Esta crítica toma como ponto de partida a idéia de que arte é um processo inconsciente e que arte consciente não é arte.

Pessoalmente, não acreditamos na total inconsciência da arte. Por outro lado, não acreditamos que algum compositor seria tolo o bastante para apresentar numa audição uma peça completamente produzida (e não apenas auxiliada) por um *seu* programa de composição. Isso porque este tipo de programa, como já dissemos, serve apenas como emulador ou simulador de uma parte dos processos de composição desse mesmo artista, jamais substituindo-o. A principal finalidade do programa compositor é a de estudar a composição em si.

Uma ressalva se faz necessária. Estamos discutindo a validade do **programa compositor** (como nosso próximo projeto: COMP) e não da composição assistida por computador (conhecida pela sigla CAC). Nessa última, a composição é preparada por um ser humano, mas ele necessita do computador para um pesado e tedioso desenvolvimento matemático. É o caso de nosso quarto projeto: os fractals musicais. Neste, as simetrias envolvidas são tantas que se faz quase necessário o uso do computador.

Vamos então passar a ensinar o computador a compor.

Retomando a definição de composição, "compor é organizar sons segundo certas regras" ou "compor é usar regras para saber que nota deve vir após, antes e junto com outra nota". Como, então, fazer o computador escolher as notas baseado nas regras?

Bem, supondo que uma determinada nota já esteja escolhida, temos de escolher a próxima. E, para escolher uma "boa" nota temos de usar as regras de composição. Como fazer o computador escolher *uma* nota, a partir das notas anteriores e das regras?

Propomos dois métodos para a escolha da "próxima nota". O primeiro método consiste em criar uma porção de notas completamente ao acaso e escolher a melhor delas. A melhor será a que melhor se adaptar às regras. O segundo método é usar as regras para escolher uma nota apenas, uma nota que já se adapte perfeitamente às regras.

Se fôssemos comprar um terno, teríamos dois caminhos a seguir: o primeiro seria entrar numa loja e escolher, dentre os ternos expostos na loja, o melhor, segundo nossas regras inconscientes e muito particulares do que seria um bom terno. O segundo método seria ir a um alfaiate e encomendar um terno perfeito.

São esses os dois métodos que propomos aqui: o primeiro, de preparar várias

notas ao acaso e escolher a melhor, será chamado de "avaliação", o segundo, de preparar logo de uma vez a nota perfeita, segundo as regras, será chamado de "dedução".

8.3.1 COMPOSIÇÃO POR AVALIAÇÃO

O programa aqui incluído para ilustrar esse método foi chamado de COMP.

Programa-exemplo COMP

```
10 DEFINT A-Z:CLS
30 DIM NA(2),NU(2),DM(2),NE(9),CC(9),MU(63,2),RT(63)
40 R=RND(-TIME)
50 N$="C C#D D#E F F#G G#A A#B "
100 NU(0)=RND(1)*15+20:NU(1)=NU(0)-3-INT(RND(1)*2)
105 NU(2)=NU(1)-3-INT(RND(1)*2)
110 GOSUB 2000
115 PRINT"PREPARANDO A COMPOSICAO..."
117 PRINT
120 FOR NN=0 TO 63
130 MU(NN,0)=NU(0):MU(NN,1)=NU(1):MU(NN,2)=NU(2)
140 PRINT NN;"-";
142 FOR N=0 TO 2
144  PRINT NU(N);MID$(N$,(NU(N) MOD 12)*2+1,2);
146 NEXT N
148 PRINT"  TEMPO :";RT(NN)
150 VZ=INT(RND(1)*3)
200  GOSUB 1000:GOSUB 1100
210  DM(VZ)=SGN(NE(MX)-NU(VZ))
220  NU(VZ)=NE(MX)
230  IF RND(1)<.5 AND VZ<2 THEN VZ=VZ+1:GOTO 200
300 NEXT NN
310 PRINT
320 PRINT"PRONTO PARA TOCAR."
330 PRINT"APERTE QUALQUER TECLA."
340 IF INKEY$="" THEN 340
390 PLAY "T120","T120","T120"
400 FOR K=0 TO 1:IN=0:GOSUB 2500:NEXT
450 FOR K=0 TO 1:IN=8:GOSUB 2500:NEXT
455 FOR K=0 TO 1:IN=0:GOSUB 2500:NEXT
460 FOR K=0 TO 1:IN=16:GOSUB 2500:NEXT
580 FOR K=0 TO 1:IN=24:GOSUB 2500:NEXT
585 FOR K=0 TO 1:IN=32:GOSUB 2500:NEXT
590 FOR K=0 TO 3:IN=0:GOSUB 2500:NEXT
595 FOR K=0 TO 1:IN=24:GOSUB 2500:NEXT
600 FOR K=0 TO 1:IN=40:GOSUB 2500:NEXT
```

```
620 FOR K=Ø TO 1:IN=48:GOSUB 2500:NEXT
710 FOR K=Ø TO 1:IN=56:GOSUB 2500:NEXT
790 FOR K=Ø TO 1:IN=Ø:GOSUB 2500:NEXT
900 END
997 '-----------
998 ' GERA DEZ POSSIVEIS NOTAS COM VALORES DE 2Ø A 6Ø
999 '-----------
1000 FOR N=Ø TO 9:NE(N)=INT(4Ø*RND(1)+2Ø):NEXT N:RETURN
1097 '-----------
1098 ' CRIA O CONCEITO PARA CADA UMA DAS NOTAS DE NE().VZ E' A VOZ.
1099 '-----------
1100 FOR N=Ø TO 9
1110  CC(N)=Ø
1120  IT=ABS(NE(N)-NU(VZ))
1125  DM=SGN(NE(N)-NU(VZ))
1130  IF IT=5 OR IT=7 THEN CC(N)=CC(N)+1Ø:GOTO 12ØØ
1140  IF IT=2 THEN CC(N)=CC(N)+15:GOTO 12ØØ
1150  IF IT=3 OR IT=4 THEN CC(N)=CC(N)+7:GOTO 12ØØ
1160  IF IT=1 OR IT=8 OR IT=9 OR IT=Ø THEN CC(N)=CC(N)+5:GOTO 12ØØ
1200  IF DM<>DM(VZ) THEN CC(N)=CC(N)+5
1220  FOR K=Ø TO 2
1230   IF K=VZ THEN 13ØØ
1240   IH=ABS(NE(N)-NU(K))
1250   IF IH=7 OR IH=3 OR IH=4 THEN CC(N)=CC(N)+5:GOTO 13ØØ
1260   IF IH=8 OR IH=9 OR IH=5 THEN CC(N)=CC(N)+3:GOTO 13ØØ
1270   IF IH MOD 12=1 OR IH MOD 12=2 THEN CC(N)=CC(N)-1Ø:GOTO 13ØØ
1280   IF IH MOD 12=1Ø OR IH MOD 12=11 THEN CC(N)=CC(N)-8:GOTO 13ØØ
1300  NEXT K
1310  IF VZ<2 THEN IF NE(N)>NU(VZ+1) THEN CC(N)=CC(N)+8:GOTO 14ØØ
1320  IF VZ>Ø THEN IF NE(N)<NU(VZ-1) THEN CC(N)=CC(N)+8:GOTO 14ØØ
1400  IF VZ=2 AND NE(N)<4Ø THEN CC(N)=CC(N)+2
1450 NEXT N
1500 MX=Ø:CX=Ø
1510 FOR N=Ø TO 9
1520  IF CC(N)>CX THEN CX=CC(N):MX=N
1530 NEXT N
1600 RETURN
1997 '---------
1998 ' CRIA PADRAO RITMICO
1999 '---------
2000 FOR N=Ø TO 63
2010  RT(N)=4
2020 NEXT N
2030 FOR N=Ø TO 56 STEP 4
2035  IF RND(1)<.3 THEN 22ØØ
2040  X=INT(RND(1)*4+N)
2050  Y=INT(RND(1)*4+N):IF Y=X THEN 2Ø5Ø
2060  Z=INT(RND(1)*4+N):IF Z=Y OR Z=X THEN 2Ø6Ø
2100  RT(X)=RT(X)/2
2110  RT(Y)=RT(Y)*2
2120  RT(Z)=RT(Z)*2
```

```
2200 NEXT N
2250 RETURN
2497 '---------
2498 ' TOCA AS NOTAS DE IN A IN+7
2499 '---------
2500 FOR NN=IN TO IN+7
2510  PLAY "L=RT(NN);","L=RT(NN);","L=RT(NN);"
2520  X=MU(NN,0):Y=MU(NN,1):Z=MU(NN,1)
2530  PLAY"N=X;","N=Y;","N=Z;"
2540 NEXT NN
2550 RETURN
```

Nesse método, escolhemos dez notas ao acaso e usamos as regras para avaliar as notas uma a uma. A nota que tiver, após o processo de avaliação, obtido a pontuação mais alta será a nota que, efetivamente, será incluída na composição. As outras serão descartadas.

O número de notas escolhidas ao acaso não é tão importante. Nesse programa usamos dez. Quanto maior esse número, maior é o número de escolhas (maior a chance de se encontrar uma boa nota) mas maior é, também, o tempo de processamento. O número dez é um bom meio termo. Mude-o, se achar conveniente.

Temos, em primeiro lugar, de colocar as três notas iniciais de cada voz, sem usar as regras.

A primeira nota da primeira voz é escolhida totalmente ao acaso (qualquer uma com número de série entre 20 e 35). Já estamos usando uma regra, por mais banal que ela pareça. Consideramos, como primeira regra de composição que "a primeira nota de uma música não é importante". O que importa são as notas que seguem e suas relações com essa primeira nota. A regra que estamos usando diz, na verdade, que "não importa em que **tom** estamos tocando uma música, ela sempre será a mesma música". Uma mesma música pode começar em qualquer nota, dependendo do tom em que for tocada. Para nós isso não é importante. Para os hindus, por exemplo, é.

Já está escolhida a primeira nota da primeira voz. A primeira nota da segunda voz é a nota **uma terça maior ou menor (3 ou 4 meios-tons) acima daquela. A primeira nota da terceira voz tem também essa distância com a segunda voz.** Esse processo está nas linhas 100 e 105 do programa.

Isso faz com que nossa composição comece num acorde maior, menor, diminuto ou aumentado.

Escolhida a primeira nota de cada voz, vamos escolher a próxima, segundo as seguintes regras:

REGRAS DE MELODIA (linhas 1130 a 1200): IT é o **intervalo melódico** (a distância entre uma nota e a nota anterior da **mesma** voz). Ele é calculado na linha 1120.

Se IT é igual a 5 meios-tons (quarta justa) ou 7 meios-tons (quinta justa), some 10 pontos (linha 1130).

Se IT é igual a 2 meios-tons (segunda maior) some 15 pontos (linha 1140).

SE IT é igual a 3 meios-tons (terça menor) ou 4 meios-tons (terça maior), some 7 pontos (linha 1150).

Se IT é igual a 1 meio-tom (segunda menor), 8 meios-tons (sexta menor), 9 meios-tons (sexta maior) ou é igual (uníssono), some 5 pontos (linha 1160).

Se houve movimento contrário (a voz está subindo agora, mas tinha descido na última nota ou vice-versa), some 5 pontos (linhas 1125 a 1200).

REGRAS DE HARMONIA (linhas 1220 a 1300): IH é o **intervalo harmônico** (a distância entre uma nota e outra de uma voz diferente). Ele é calculado na linha 1240.

Se IH é igual a 7 meios-tons (quinta justa), 3 meios-tons (terça menor) ou 4 meios-tons (terça maior), some 5 pontos (linha 1250).

Se IH é igual a 8 meios-tons (sexta menor), 9 meios-tons (sexta maior) ou 5 meios-tons (quarta justa), some 3 pontos (linha 1260).

Se IH é igual a 1 meio-tom (segunda menor) ou 2 meios-tons (segunda maior), mesmo que as notas estejam em oitavas diferentes (o MOD tem essa função), **subtraia** 10 pontos (linha 1270).

Se IH é igual a 10 meios-tons (sétima menor) ou 11 meios-tons (sétima maior), mesmo que as notas estejam em oitavas diferentes, **subtraia** 10 pontos (linha 1280).

CRUZAMENTOS (linhas 1310 e 1320): há cruzamento quando uma voz ultrapassa a outra. Por exemplo: a primeira voz está tocando a nota 40 e a segunda está tocando 50. Se a primeira voz tocar uma nota com valor maior que 50, ela fará um movimento de **cruzar** a segunda voz.

Se a voz não cruzou uma outra voz, some 8. Se não cruzou nenhuma, ambas as linhas darão pontuação, totalizando 16 pontos.

REGRA DE TESSITURA (linha 1400): se a nota da terceira voz não for maior ou igual a 40, some 2 pontos.

As regras usadas são bastante tradicionais, exceto por uma ou outra. Mude-as!! Encha sua composição com intervalos de quarta aumentada, cruzamentos, meios-tons simultâneos. Ou, ao contrário, não permita intervalos harmônicos de terça...

As regras estão aí para serem mudadas.

Outra sugestão: mude o método de criação de divisão rítmica. O método usado (linhas 2000-2250) permite apenas alguns padrões diferentes, sempre compostos de uma mínima, uma semínima e duas colcheias (não necessariamente nessa ordem).

8.3.2. COMPOSIÇÃO POR DEDUÇÃO: FRACTALS

A composição por dedução ou composição assistida por computador (CAC) é a mais usada. Ela, na verdade, utiliza o computador apenas para auxiliar o músico em tarefas de cálculo que seriam tediosas em demasia caso ele não fosse usado.

O exemplo de CAC apresentado aqui é o método de composição por **fractals**, também chamado de **mudança de escala**.

Veja a partitura da Figura 8.2. Nela temos uma pequena frase de quatro semibreves.

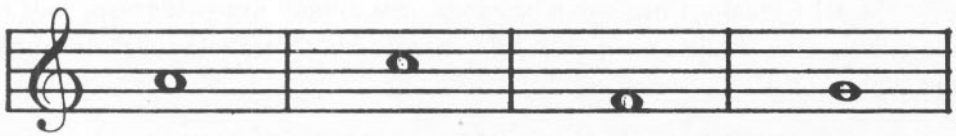


Figura 8.2 Fractal de primeira ordem

A filosofia do método que vamos usar diz:

Em cada parte vai o todo.

Vamos chamar a frase original (a da Figura 8.2) de **fractal de primeira ordem.**

O fractal de **segunda ordem** terá a mesma frase em uma das vozes, como no fractal de primeira ordem. Até aí, nenhuma mudança. Na segunda voz, no entanto, teremos quatro vezes essa frase, **uma frase sobre cada nota**, como na Figura 8.3.

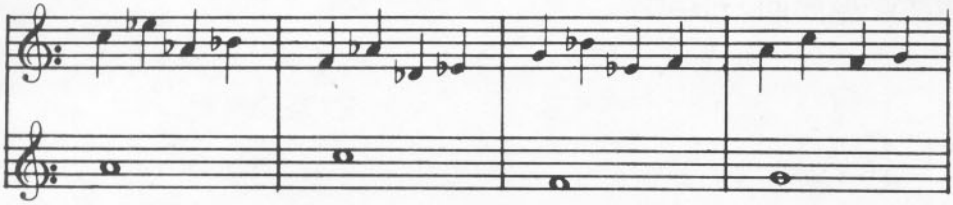


Figura 8.3 Fractal de segunda ordem

Sobre cada nota da frase original vai um cópia da frase, em escala reduzida. A regra para a redução de escala é que a nova frase deve preencher exatamente a duração da nota sobre a qual está, nem mais, nem menos.

A distância harmônica entre a primeira nota da nova frase e a nota de baixo deve ser igual à distância melódica entre a nota de baixo e a nota seguinte na frase original.

Na Figura 8.4 está o fractal de terceira ordem.

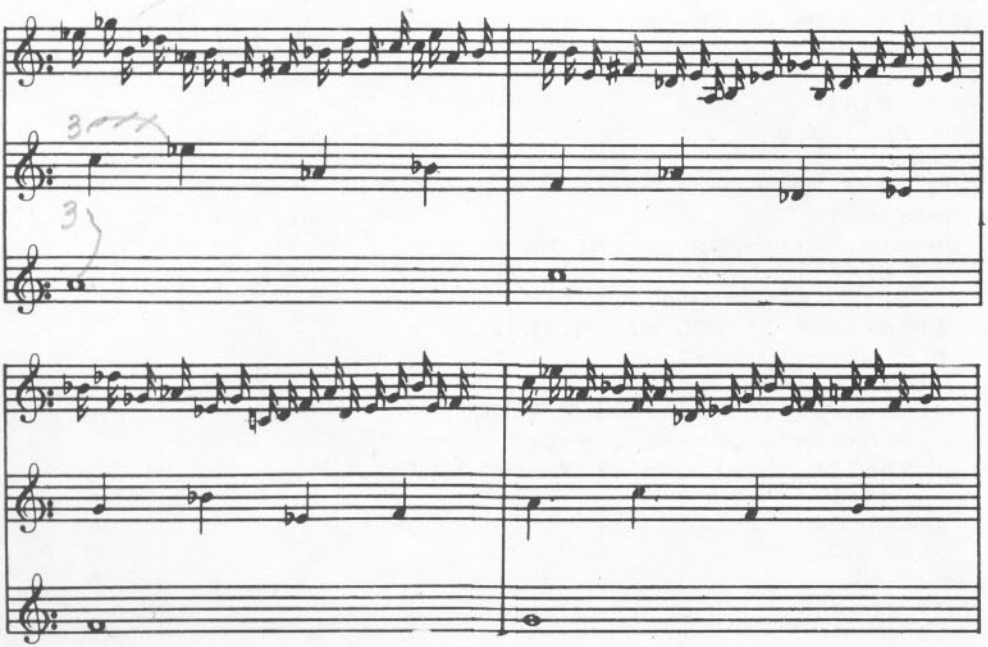


Figura 8.4 Fractal de terceira ordem

O programa FRAC serve exatamente para calcular e executar fractals de terceira ordem

Programa-exemplo FRAC

```
10 DEFINT A-Z
15 A=RND(-TIME)
20 CLS
30 PRINT"FRACTALS MUSICAIS"
35 PRINT
40 INPUT"NUMERO DE NOTAS";NI
50 DIM N1(NI-1),T(NI-1)
70 FOR N=0 TO NI-1
75   N1(N)=0
77   PRINT"NOTA";N+1;:INPUT N1(N)
80   IF N1(N)=0 THEN N1(N)=RND(3)*20+20
90   PRINT N1(N)
100  PLAY"L1N=N1(N);"
110  T(N)=0
120  PRINT"NO. DE TEMPOS";:INPUT T(N)
130  IF T(N)=0 THEN T(N)=RND(4)*2+1
140  PRINT T(N)
200 NEXT N
210 PRINT
215 INPUT"VALOR DA MENOR FIGURA (1-64)";VF
220 FOR N=0 TO NI-1:FOR I=1 TO T(N):PLAY"L2N=N1(N);":NEXT I:NEXT N
230 PLAY"L=VF;","L=VF;","L=VF;"
240 PLAY"R1R1"
250 SCREEN 2,0,0
260 X=10
310 FOR NA=0 TO NI-1
315  A=N1(NA)
320  DA=N1((NA+1) MOD NI)-N1(NA)
350  FOR NB=0 TO NI-1
360   B=N1(NB)+DA
370   DB=N1((NB+1) MOD NI)-N1(NB)
380   FOR NC=0 TO NI-1
420    C=N1(NC)+DA+DB
500    FOR NV=1 TO T(NA)*T(NB)*T(NC)
510     PLAY"N=A;","N=B;","N=C;"
530     PSET(X,96-A):PSET(X,96-B):PSET(X,96-C)
540     X=X+1
570    NEXT NV
600   NEXT NC
650  NEXT NB
700 NEXT NA
1000 CLS
```

```
1010 INPUT"REPETIR";A$
1020 IF LEFT$(A$,1)="S" OR LEFT$(A$,1)="s" THEN 220
1030 INPUT"OUTRO FRACTAL";A$
1040 IF LEFT$(A$,1)="S" OR LEFT$(A$,1)="s" THEN RUN
1050 END
```

Para criar o fractal você deve introduzir os seguintes valores: o número de notas da frase original, o número de série e a duração de cada nota da frase e o valor do menor tempo.

O número de notas deve ser baixo. Tente não passar de doze.

O número de notas de um fractal de terceira ordem é o cubo do número de notas original.

Se você colocar 16 notas no seu fractal original, você terá 4096 notas na composição final.

A duração de cada nota é indicada pelo número de notas de menor tempo que ela ocupa. É aconselhável não abusar de valores maiores que um. Não use ritmos muito variados.

Use, para o valor do menor tempo, figuras da ordem de 8 ou 16. Não use figuras muito lentas nem muito rápidas.

O programa executa a frase original, lentamente, por um tempo e depois começa a tocar o fractal de terceira ordem.

Aqui vão alguns exemplos de frases básicas. O tempo para cada uma das notas está entre parênteses, após seu número de série. O valor da menor figura pode ser 8 (lento) ou 16 (rápido).

CONTATOS IMEDIATOS

Número de notas: 5
Notas: 42 (1), 44 (1), 40 (1), 28 (1), 35 (2)

ATIREI O PAU NO GATO

Número de notas: 6
Notas: 37 (3), 35 (1), 34 (1), 32 (1), 34 (1), 35 (1)

FRACTAL CANTATA Nº 3

Número de notas: 12
Notas: 40 (1), 42 (1), 40 (1), 47 (1), 45 (1), 47 (1), 40 (1), 42 (1), 40 (1), 35 (1), 33 (1), 35 (1)

O programa não é extenso e sua parte realmente importante vai das linhas 310 a 700. Nessas linhas estão os três *loops* aninhados que calculam o fractal de terceira ordem.

O número de notas do tema inicial (o fractal de primeira ordem) é guardado na variável NI.

Cada um dos três loops percorre o tema, calculando as notas dos três canais e guardando-as nas variáveis A, B e C.

A linha 530 coloca as notas dos três canais na tela, sob a forma de pontos. Quanto mais grave o som, mais "baixo" estará o ponto do canal correspondente. Esta é uma forma de se perceber a simetria visualmente, além de ouvi-la.

CAPÍTULO 9

CONSIDERAÇÕES FINAIS

Muitos outros tópicos e projetos poderiam ser concretizados ou discutidos detalhadamente aqui, mas o espaço disponível, infelizmente, não é infinito.

Abaixo segue uma lista de projetos e idéias de maior ou menor dificuldade, que merecem uma consideração:

SINTETIZADOR POLIFÔNICO

Transforme o programa SINT num sintetizador polifônico. Permita que o usuário execute mais de uma nota de cada vez, com a pena de perder a variação de timbre, uma vez que os três canais não mais poderão ser usados para a síntese aditiva.

COMPOSITORES ALEATÓRIOS

Se você conhece um pouquinho de estatística, seria interessante que você produzisse melodias aleatórias usando diferentes distribuições. Os resultados são surpreendentes.

COMPOSITOR POR AMOSTRAGEM

Construa um programa que toque uma melodia a partir de uma tabela de probabilidades. O usuário entra com uma melodia inicial (a amostra) e o programa cria uma tabela que contém o número de vezes que cada nota ocorreu. A partir daí, uma melodia é criada, com a mesma probabilidade de ocorrência das notas. Se você conhece a teoria das cadeias de Markov, pode usá-la também. Assim você estaria criando melodias cuja probabilidade é dependente das últimas duas ou três notas produzidas.

DODECAFONISMO AUXILIADO POR COMPUTADOR

Um programa útil para quem faz música dodecafônica seria um que, a partir da série original, calcula a invertida, a retrógrada, e a invertida retrógrada e transpõe as quatro séries nos doze tons.

EDITOR DE PARTITURAS

Reprograme os caracteres do MSX para que eles mostrem símbolos musicais. Faça um editor de tela que contenha características que facilitem o trabalho do músico (por exemplo: divisão automática de compassos, espaçamento proporcional de notas, inserção e deleção inteligentes...) e, se você tiver paciência suficiente, crie rotinas para imprimir o resultado numa impressora gráfica. Este projeto não é para quem tem nervos fracos.

HARMONIZADOR

Construa um programa que harmonize uma dada melodia. O programa deve completar os dois canais que faltam usando uma teoria de sua escolha (harmonia funcional, harmonia jazzística tipo Berklee, bitonalismo...). O compositor poderia ser por avaliação ou dedutivo. O primeiro é mais fácil de construir.

ACORDES DE VIOLÃO OU PIANO

Construa um programa que indique todas as inversões possíveis para um dado acorde, no violão ou no piano. Use a capacidade gráfica do micro.

Como dissemos anteriormente, o MSX é excelente para este tipo de experimento. Invente outras aplicações. Não pretendemos terminar o livro com um chavão do tipo "o limite é a sua imaginação", pois somos suficientemente realistas. Existem outros fatores limitantes, é claro, como, por exemplo: o número de canais disponíveis, a qualidade do som, a velocidade do BASIC...

De qualquer maneira, o MSX tem, disparado, a melhor relação preço/performance do mercado nacional, na área de Computação com Música. Esperamos que este livro tenha mostrado um pouco das potencialidades pouco exploradas desta máquina.

Composição e Arte-Final:
JAG Composição Editorial e Artes Gráficas Ltda.
Praça F. Roosevelt, 208 - 8º andar
Tel. (011) 255-5694 - São Paulo

Impressão e Acabamento
GRÁFICA E EDITORA FCA
AV. HUMBERTO DE ALENCAR CASTELO BRANCO, 3972 - TEL.: 419-0200
SAO BERNARDO DO CAMPO - CEP 09700 - SP

José Maurício O. Bussab é analista de sistemas da Código Informática. É também autor do conhecido software TOQUE! e de trilhas sonoras de diversos jogos para a linha MSX. Neste livro ele mostra como usar o BASIC-MSX na execução e composição de música. As idéias apresentadas são ilustradas com programas-exemplos e aplicadas na construção de quatro projetos completos: um sintetizador, um programador de ritmos e dois compositores automáticos.

0-07-450057-0

McGraw Hill