

“Enquanto eles riem” para Clarinete e computador rodando MAX/MSP – Escrita e Estratégias interativas

Cristiano Figueiró, Anselmo Guerra de Almeida

Laboratório de Pesquisa Sonora Emac/UFG – Campus II - Samambaia - Caixa

Postal 131 Goiânia - Goiás - Brasil - Cep 74001-970

figocris@yahoo.com.br, anselmo@musica.ufg.br

Resumo

Esse artigo expõe o processo de composição da peça “Enquanto eles riem” para Clarinete e Computador rodando MAX/Msp, tanto no âmbito da técnica de programação empregada, quanto da escrita instrumental e alguns aspectos poéticos de construção da peça como por exemplo a intervenção do performer no processo de composição e a interatividade do performer com o computador. O primeiro trecho do artigo expõe os elementos de escrita instrumental e de poética composicional. Na segunda parte do artigo é expostas as estratégias de programação em MAX no sentido de vencer os desafios de interatividade na composição.

Abstract

This article exposes the compositional process of the piece “Enquanto eles riem” to Clarinet and Computer with MAX/Msp, in the field of the programming technique, how in the instrumental writing and some poetic aspects of construction of the piece, for example, the performer intervention in the composition process and the performer interactivity with the computer. The first part of the article exposes elements of instrumental writing and poetic composition. The second part of the article exposes some MAX programming techniques in order to win the challenge of interactivity in the composition.

Enquanto eles riem...., para Clarinete e Computador

Esta peça se tornou um desafio ao longo do processo, por abranger áreas distintas como a escrita e performance tradicional e o universo da produção sonora digital com a idiomatismo dialético da música eletroacústica. Procurou-se não transformar este contraste num elemento castrador de uma linguagem ou outra. Mas, ao contrário, um limite bem estreito onde se possa explorar, de uma maneira consciente e, ao mesmo tempo, lúdica, as

desta seção, que se contrapõe as notas da série que são desenvolvidas, já com algumas mudanças que estavam presentes na seção A.



Fig.3: compasso 16

A entidade de base é o motivo da construção da segunda parte da seção B, que se consiste apenas de transposições deste acorde, ajustadas numa figuração rítmica quase constante (acelerando) e que faz um contraste com a dinâmica empregada. O que resulta num gestual interessante para realização de processamento com o MAX.

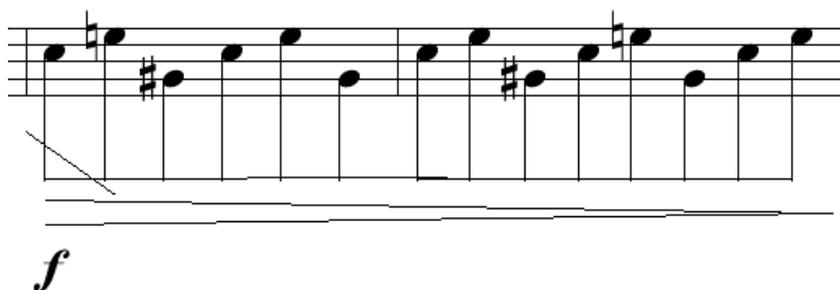


Fig.4: compasso 29-30, figura rítmica de aceleração em contraponto com a dinâmica que vai diminuindo.

A terceira parte da seção B é uma seção mais lenta, em que foi usada a entidade de base transposta como elemento básico do desenvolvimento melódico e harmônico. A partir dessa transposição, chegou-se a escala de lá menor harmônica, que na sua configuração possui três transposições da entidade de base.



Fig.5: compasso 44-46, constr. melódica em lá menor harmônica.

A quarta parte da seção B é um retorno variado à segunda parte, com caráter de cadência virtuosística e serve de ponte para o retorno à seção A que volta, com variações, no compasso 73. Esta nova seção (A'), se caracteriza por um uso mais efetivo do computador e das células rítmico-melódicas, propostas pelo performer, que acabaram sendo incorporadas na composição enquanto células mesmo e não como órgãos estruturais da composição, o que implicaria em um trabalho composicional mais profundo. Como exemplo, foi escolhido um trecho em que, após uma variação melódica da série, foi acrescentada uma célula melódica de "risada", em que o performer simula uma risada com o Clarinete. Estes elementos implicam uma diferença na execução, pois o mais importante não são as notas e sim o caráter específico do trecho.

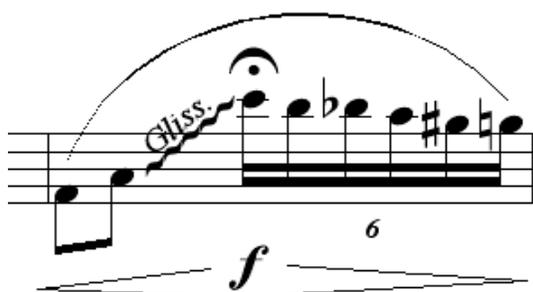


Fig.6: compasso 91, Clarinete "rindo"

No compasso 97 começa uma seção que é uma ponte para a entrada da seção B'. Nesta ponte podemos notar como foi pensada a condução vocal, pois o Clarinete, apesar de ser um instrumento melódico, com sua sustentação de notas permite uma escrita pseudo-polífonica.



Fig.7: compasso 98-101, escrita a duas vozes.

A seção B' começa no compasso 116 e combina o uso da entidade de base interpolada com a série inteira adaptada ao desenho melódico característico da parte lenta da seção B.



Fig.8: compasso 118-119, interpolação da série com a entidade de base e o desenho melódico

O final da peça consiste de uma pequena coda de cinco compassos em que o último compasso usa um artifício técnico da Clarineta, em que o performer fica assoprando o instrumento sem emitir notas definidas, apenas o ruído do sopro. Isso cria uma suspensão cênica enquanto o computador tece seus últimos comentários. Em relação a interatividade do performer com o computador, podemos dizer que é uma interatividade calcada na condução da partitura, ou seja, o desafio de interatividade é conseguir fazer o computador acompanhar a performance do músico, guiado pela partitura.

Max/Msp – implementações

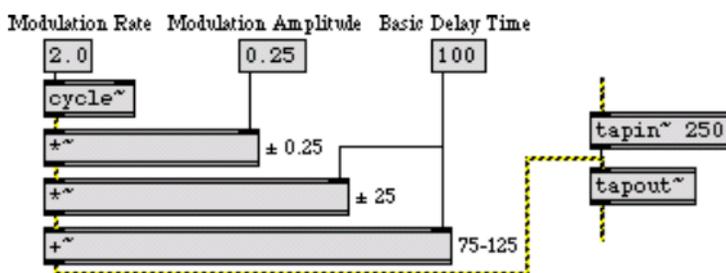
Nesta parte vamos analisar os trabalhos desenvolvidos no ambiente MAX/MSP criados a partir das premissas de interatividade musical que se fizeram necessárias durante a construção dessa peça.

Tendo em vista o pensamento composicional que guiou o processo, e o tipo de interação desejada, procurou-se desenvolver um sistema que possibilitasse que a performance da peça não dependesse de outro performer humano além do clarinetista. Aqui vamos expôr dois patches de MAX desenvolvidos nesse sentido.

Interface

Este trabalho se trata de uma interface de performance baseada no tutorial MSP número 30 do MAX. Este *patch* possibilita a captura de áudio e processamento em tempo real, no qual a idéia básica do efeito é acrescentar um oscilador como variação de tempo de *delay*. O que ocasiona um *delay* limpo quando você posiciona a frequência do oscilador em zero e um suave efeito *flanging* com a frequência abaixo de 0.5Hz, variando a densidade do efeito conforme o índice de modulação.

A principal estrutura para o entendimento deste tutorial é a operação de modulação do oscilador(*cycle~*) controlando os parâmetros de *delay* (*tapout~*)(fig. 1). Os objetos *tapin~* e *tapout~* atuam em conjunto criando linhas de *delay*. O sinal original entra no *inlet* esquerdo de *tapin~* que tem como argumento o tamanho da linha de *delay* descrito em milisegundos e que vai ser lido por *tapout~* que tem como argumento o tempo em milisegundos que leva para acionar a primeira linha de *tapin~*. Neste caso, além da saída de *tapin~*, é somada a entrada de *tapout~* os dados de um oscilador (*cycle~*) que realiza a modulação do sinal de áudio. A saída de *tapout~* toma dois rumos: o primeiro vai direto para a saída de áudio e o segundo é escalado por um multiplicador variável (de 0 a 1) e reconectado a entrada de *tapin~* o que causa um efeito que em pedais de guitarra se conhece por *feedback* e se trata de uma repetição contínua das linhas de *delay*.



Modulating the delay time with a low-frequency oscillator

Fig.9 : núcleo básico do efeito flanger.

O trabalho de programação em si, além da análise do tutorial, foi a criação de uma interface de uso em performance. Uma interface é um dispositivo físico ou lógico que faz a adaptação entre dois sistemas, que no caso foram de um lado o *patch* do tutorial do MSP e de outro um performer com a partitura da composição e a possibilidade de improvisar otimizando os recursos do tutorial, ou outro *patch* controlador do sistema que ao acompanhar a performance, liga e desliga essa interface. A construção de uma interface de uso do programa faz parte do processo de aprendizado de uma linguagem. No caso do MAX, escrever interfaces, torna-se básico na aprendizagem, pois toda a estrutura da linguagem é gráfica e voltada para a performance.

Nesse caso foram acrescentados objetos gráficos de controle como botões e sliders verticais, horizontais e em forma de caixa de desenho (**pictslider** - onde o controle é feito com o mouse e valor controlado pelas quatro arestas da caixa). As ligações da interface com o *patch* do tutorial foram feitas através do objeto **patcher** que realiza a conexão e o encapsulamento de um *patch* com outro. Alguns artifícios foram adicionados, no sentido de ampliar as possibilidades interativas do performer com o computador e de facilitar a manipulação do programa durante a performance. Foram acrescentadas envoltórias de segmentos lineares (**line**) ao controle de volume, causando assim um efeito de *fade in* e *fade out*. Os controles de tempo de *delay* e espacialização do sinal processado respectivamente são controlados pelo objeto **drunk**, que é um gerador numérico randômico e que necessita de um comando *bang* para se acionar, que, no caso, é fornecido por **metro**, um metrônomo gerador de *bangs*. O fato de ter alguns parâmetros controlados por gerador independente, que produz parâmetros randômicos durante a performance, empurra o performer a interagir de maneira mais intensa com a resposta do computador.

uma peça que inclui improvisos do performer e alguns pontos sincronizados com o acionamento de *samples* pelo computador. Ou seja, o computador espera pelo performer os pontos certos de sincronia, que são uma sequência de notas. Os espaços entre os pontos de sincronia são usados para improviso do performer que deve interagir com o universo sonoro que ele mesmo acionou.

O processo do *patch* é muito simples. O interesse vem da forma prática e econômica da organização das ações e a portabilidade, podendo fazer parte de um complexo de outros elementos ou simplesmente mudarmos a função do mapeamento dos dados que, como vimos, podem ser muitas. Os dados de performance podem ser mapeados para o acionamento de *samples* que estejam na memória do programa, controle de parâmetros de síntese ou processamento, ou nível composicional.

A organização deste programa pode ser dividida em quatro partes: reconhecimento de alturas, transferência dos dados da performance para dados MIDI, acompanhamento e acionamento de *samples*. A parte do reconhecimento de alturas foi resolvida com o uso do objeto **fiddle~**, escrito por Miller Puckete (1999). Este objeto é um algoritmo que fornece alguns dados sobre o reconhecimento da performance que está na sua entrada de áudio, como altura e amplitude. No caso, foi usado apenas o reconhecimento de altura, que é endereçado para o objeto **midiformat**, que transforma um valor de altura em dado MIDI.

Uma vez que o programa faz o reconhecimento de altura da performance e transforma isso em dado MIDI, podemos enviar este dado para o objeto **follow**, que é um objeto em que podemos sincronizar performances MIDI. Ou seja, o objeto tem uma sequência MIDI na sua memória e consegue acompanhar uma outra sequência que fica sendo mandada a ele com base na parte gravada. A cada nota que é reconhecida, o objeto manda uma mensagem que aciona um processo, que no caso é o *playback* de *samples*, e que poderia ser uma série de ações possibilitando diferentes níveis de interatividade, como, por exemplo, o acionamento de estruturas sonoras baseadas no histórico do comportamento da performance, sequências randômicas com parâmetros variáveis, processamentos variados do sinal de entrada, etc....

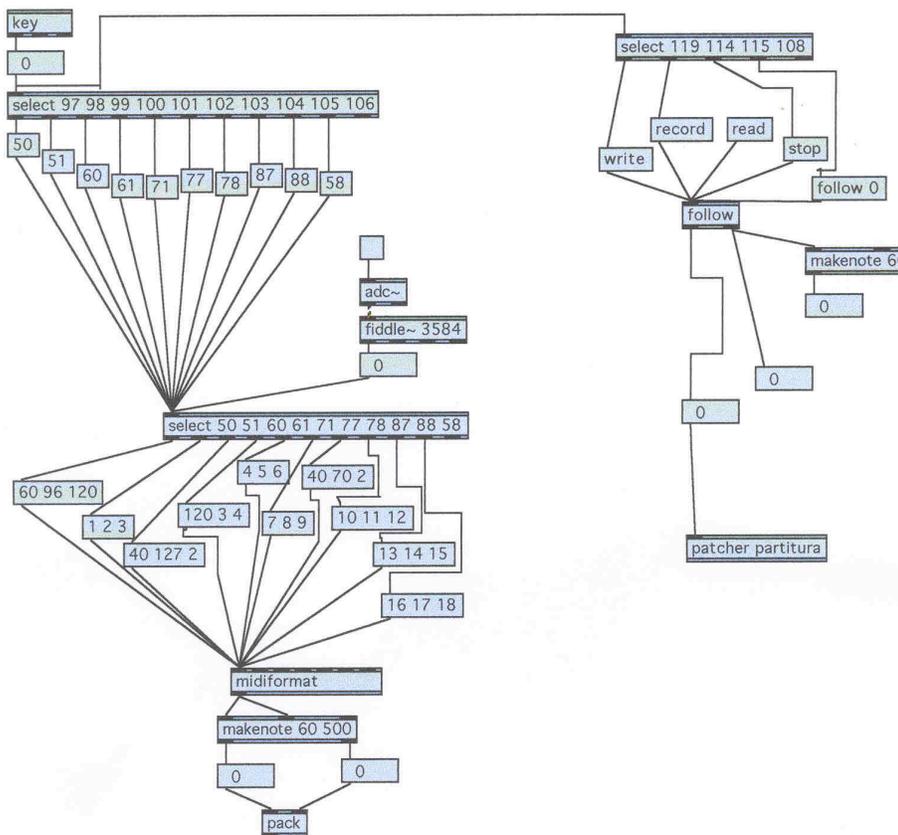


Fig.11: *patch* para MAX que realiza acompanhamento interativo (*Score – Follower*).

Conclusão

O *score-follower* desenvolvido ainda é uma versão instável, ou seja, ainda não é uma ferramenta com a qual podemos contar sem o ônus de erros. Mas consiste de uma estrutura básica de um *score-follower*, pronto para desenvolvimento e implementação em composições. Os principais desenvolvimentos são o trabalho, direto no algoritmo do objeto **fiddle~**, no sentido de estabelecer diferenças entre os timbres dos instrumentos, pois um dos maiores problemas de leitura deste objeto é o fato dele reconhecer muitos harmônicos de cada timbre, enquanto que o interessante para um *score-follower* seria o reconhecimento apenas do harmônico fundamental de cada nota, pois para cada timbre ou espectro harmônico diferente ele se comporta de maneira diferente. Outro passo é o de distribuir o

processamento, pois, neste caso, o mesmo programa reconhece as alturas e realiza o acompanhamento acionando os *samples*. Se conseguirmos deslocar o reconhecimento para um processador e o acionamento do acompanhamento para outro processador, podemos chegar mais perto de uma situação estável. Poderemos também ampliar a lógica deste *patch*, pois poderá ser incluído, por exemplo, a análise de comportamento da performance de maneira que o *patch* responda de maneiras diferentes de acordo com cada performance. A saída para a instabilidade desse *patch* é a performance auxiliada por um performer ao computador, acionando os *samples* e a interface de processamento manualmente.

Podemos fazer uma síntese dos processos composicionais de escrita e de programação que mais foram utilizados: serialização livre de alturas, idiomatismo instrumental como ponto estrutural nas composições, contraste entre narrativas diversas, colagens, granulações, interação com intérprete no nível da performance e da composição, estruturação eletroacústica provinda da escrita instrumental.

Os *patches* de MAX/MSP, pela própria característica de programação do ambiente MAX, são automaticamente células que podem ser incorporadas a novos projetos, ou ter detalhes alterados para servir a novas funções. Por vezes, o mesmo *patch*, com suas possibilidades bem exploradas, pode nos fornecer inúmeras respostas, muitas que nem tínhamos pensado originalmente.

Para finalizar, podemos concluir que o processo composicional de pesquisa serviu para o aprofundamento da busca por uma identidade artística, através da ampliação de técnicas de programação e da reflexão ao redor do fazer musical contemporâneo e suas relações com as novas tecnologias. Identidade artística essa que entendemos como uma constante busca dinâmica da expressão pessoal através do universo que engloba nossa individualidade. Enquanto etapa de formação artística para um compositor, pode-se dizer que esta pesquisa serviu para a ampliação do repertório gestual e idiomático através de possibilidades de ambientes computacionais voltados para a composição, escrita instrumental e combinação de linguagens.

Referências Bibliográficas

- ALMEIDA, Anselmo Guerra de. **A arte como modelo de interatividade**. In: Anais da Anppom, 1999.
- CAESAR, Rodolfo. **Novas interfaces e a produção eletroacústica**. In: Anais do SBCM, Brasília, 1997.
- CORREA, James. **Jardim dos caminhos que se bifurcam**: Processos composicionais. Dissertação de Mestrado, Programa de pós graduação em música, UFRGS, 2003.
- ECO, Umberto. **Obra Aberta**. Perspectiva, São Paulo, 1992.
- FERRAZ, Silvio. **Criação musical com suporte tecnológico**. In: Anais da Anppom, 1999.
- Composição e pesquisa: A categoria compositor-pesquisador ou o compositor que se perdeu num tubo de ensaio**. In: Anais da Anppom, 2000.
- NYMAN, Michael. **Experimental music – Cage and beyond**. Schirmer Books, New York, 1974.
- OPCODE Systems. **MAX Reference Manual**. Pablo alto, 1995.
- ROADS, Curtis. **Computer music tutorial**. MIT Press, Massachussets, 1996.
- TOOP, Richard. **György Ligeti**. Londres, Phaidon Press, 1999.
- WINKLER, Todd. **Composing Interactive Music – Techniques and ideas using Max**. MIT Press, Massachussets, 1993.
- XENAKIS, Iannis. **Determinacy and indeterminacy**. In: Organised Sound, Vol. 1, Nº 3, December 1996. Cambridge University Press, Cambridge, 1996.
- ZAMPRONHA, Edson. **Notação, Representação e Composição – Um novo paradigma da escritura musical**. São Paulo, Annablume/Fapesp, 2000.