Invenção para Clarinete e PD

Jair Teixeira Filho
IA - Unicamp (mestrando)
e-mail: jairzinhoteixeira@gmail.com

José Eduardo Fornari Novo Júnior
IA - Unicamp (Professor) e NICS - Unicamp (pesquisador)
e-mail: tutifornari@gmail.com
web: www.nics.unicamp.br

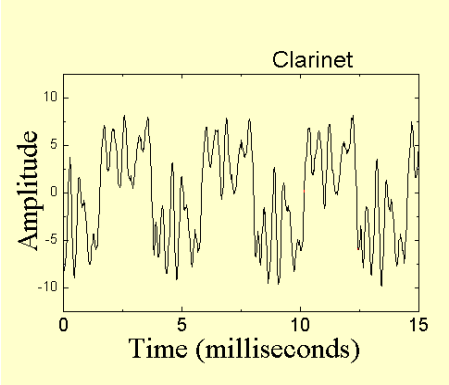
RESUMO: *Este artigo descreve o projeto de criação de uma peça musical intitulada "Invenção para Clarinete e PD". Esta é uma experiência de criação em música computacional interativa para um clarinete, em Bb (Si Bemol), e um modelo computacional projetado em Pure Data (PD), o ambiente de programação de processamento em tempo-real de dados de audio, imagem e controle. O objetivo é trazer conceitos eletroacústicos de criação e interação entre instrumentos musicais e computacionais, criando efetivamente uma música computacional interativa.*

1. INTRODUÇÃO

As mudanças estéticas e técnicas dos elementos musicais utilizados na produção musical a partir da década de 1940 trouxeram a música eletroacústica para um plano de maior importância ao revelar suas contribuições ao pensamento, à criação e às práticas musicais (MIRANDA, 2008). Com base nas afirmações de IAZZETA (1997) acerca de práticas e metodologias da música eletroacústica, o objetivo macro deste projeto é relacionar sons e procedimentos eletroacústicos com sons e procedimentos instrumentais ampliando a gama de sons e timbres, transformando a dramaticidade do fazer musical.

Os materiais escolhidos foram o clarinete em Bb (Si Bemol) e o software Pure Data. O clarinete foi escolhido, primeiramente, por ser instrumento de domínio deste primeiro autor; posteriormente, por se tratar de um instrumento de características acústicas peculiares. Ele é exemplo de um instrumento de tubo cilíndrico fechado em uma das extremidades. Se diferencia do saxofone e do oboé, que possuem tubos cônicos, pelo modo de ressonância normal, em que a pressão na extremidade fechada (boquilha) é muito próxima à pressão da primeira abertura (chave) ou da campana, no caso de o tubo todo estar fechado. Essas

condições resultam na presença de harmônicos "estranhos"[1] no som do instrumento, como na figura abaixo, onde é mostrado um trecho de 15 milisegundos da forma de onda típica de uma nota gerada num clarinete (fonte: http://www.phy.mtu.edu/~suits/clarinet.html). Esta característica poderá adicionar mais "informações acústicas", ampliando o leque de resultados sonoros processados.



A escolha do software Pure Data foi um critério do professor da disciplina[2]. O Pure Data se destaca pela sua capacidade de processamento em tempo real, interface relativamente simples e por sua multiplataforma, o que amplia as possibilidades em sala de aula uma vez que os alunos fazem diferentes escolhas sobre que plataforma operacional querem utilizar (iOS, Linux, Windows, etc).

O Pure Data, conhecido pela abreviação PD, é um ambiente de programação em tempo real de audio, video e processamento gráfico. Foi desenvolvido por Miller Puckette a partir de outro ambiente de programação, o Max, desenvolvido em conjunto com o "*Institut de Recherche et Coordination Acoustique/Musique" (IRCAM)*. O ambiente gráfico da programação utiliza a metáfora do fluxograma tradicional na organização dos objetos que, conectados uns aos outros, recebem e transmitem as informações. Na complexidade de sua programação o PD também agrega alguns comandos textuais como símbolos matemáticos, expressões condicionais e de comparação que tornam um pouco mais inteligível e intuitiva a programação aos iniciantes. O PD é um software livre e tem uma grande quantidade de bibliotecas disponibilizadas pela comunidade de desenvolvedores.

---

[1] http://www.phy.mtu.edu/~suits/clarinet.html Consultado em 02/05/2012.

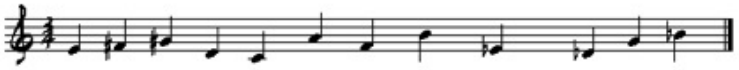
[2] Tópicos Especiais em Música e Tecnologia - Programa de Pós-Graduação em Música do Instituto de Artes - Unicamp.

2 MÉTODO

Como partes do conceito inicial de formatação do projeto foram escolhidos, como base, alguns exemplos da bibliotecas do PD Extended, que também está disponível na página da internet[3] do desenvolvedor, sendo estes:

02.VideoSphere.pd
07_corde3D.pd
40_i3D.pd
E01.spectrum.pd
E08.phase.mod.pd
G09.pitchshift.pd
H09.ssb.modulation.pd
H15.phaser.pd
I08.pvoc.reverb.pd

A partir destes exemplos foi concebido o projeto que consistia de três partes. A primeira parte consistia na execução ao clarinete de uma série dodecafônica previamente composta, qual seja:



Gradativamente seriam adicionados intervalos de 5ª, 8ª, 10ª, 12ª, 14ª e 16ª, todos superpostos. Isto se daria utilizando o objeto de multiplicação matemática  , que multiplica os números que *entram* no *inlet* da esquerda pelo valor atribuído dentro da caixa do objeto. Ao resultado sonoro desta primeira parte seriam adicionados efeitos de *reverb*, como no exemplo I08.pvoc.reverb.pd, de oscilação de frequencia, como no exemplo G09.pitchshift.pd, de *vibrato* como no exemplo H09.ssb.modulation.pd e de *phase* como no exemplo E08.phase.mod.pd.

No parágrafo acima o uso dos verbos no pretérito denuncia o abandono dessa primeira idéia. Isto se deu após as primeiras tentativas de implementação dessa primeira parte do *patch*, quando pudemos observar as dificuldades técnicas para a implementação desta idéia e a demanda de tempo necessário para tanto, o que inviabilizou a confecção desta primeira parte. Passamos então a começar a peça por aquela que seria a segunda parte.

[3] http://puredata.info/

Nesta segunda parte som e imagem interagem num processo de modulação de parâmetros gráficos pelos parâmetros sonoros. Isto se dá pela utilização de objetos de programação do PD que "traduzem" os sinais sonoros em argumentos que são utilizados como moduladores dos parâmetros gráficos. A interação sonora entre instrumentista e máquina ocorre pela realimentação do som através do microfone, que capta novamente os sons que foram gerados pelo clarinete, processados pelo PD e que saíram pelos auto-falantes. A interação gráfica ocorre quando da visualização das imagens pelo instrumentista, que reage segundo as mudanças que ocorrem no ambiente gráfico da peça.

A terceira parte consistia da gravação da primeira parte e sua posterior execução juntamente com a improvisação livre ao clarinete. A isso, somariam-se ainda todos os elementos apresentados na primeira parte que "apareceriam" e "desapareceriam" aleatoriamente no decorrer desta terceira e última parte da peça. O *fade out* tanto do clarinete quanto do PD conduziria ao fim da peça. Por motivos de escolha estética esta terceira parte também foi excluída, ficando a peça com uma única parte.

Foi adotada a improvisação livre como modelo de produção sonora e interação com o PD, e não mais a série dodecafônica.

A parte gráfica tomou sua porção na peça tendo como principal *sub-patch* o modelo 07_corde3D.pd, também do PD Extended. Este patch é de criação de Cyrille Henry[4] e foi incorporado como *sub-patch* gráfico. Basicamente, ele consiste de uma fileira de esferas que se movimentam correlacionadamente segundo massa, gravidade, elasticidade e forças próprias. Nele foram incorporados parâmetros de modulação em tempo real nos eixos de movimentação *x, y* e *z d*o plano cartesiano; isto também aconteceu com o tamanho de uma das esperas e com os lados das figuras geométricas. As cores das esferas, por sua vez, foram mudadas, mas permanecem estáticas até o momento final da peça, quando elas, uma a uma, se tornam brancas e, com num *fade out*, se tornam pretas, conduzindo ao final da peça. Também foram adicionadas esferas ao conjunto inicial do *patch*.

Na parte sonora, as freqüências produzidas pelo clarinete foram moldadas e transformadas por alguns objetos de programação, cada um deles com suas peculiaridades, e que foram escolhidos segundo a proposta inicial do projeto. O resultado é um som espacializado, ramdômico e de frequencias mais agudas.

[4] http://www.chnry.net/ch/?lang=fr

2.1 Objetos do Pure Data Extended

Passaremos agora a uma sucinta explicação dos objetos utilizados. Dois objetos muito utilizados e de conceito simples são  e  ; *"bang"* e *"toggle"*, respectivamente. O *"bang"* origina um impulso, um início, um *start* quando *clicado* e é normalmente usado para iniciar eventos. O *toggle* funciona como uma chave de liga/desliga, podendo ser acionado por um *"bang"* ou por caixas numéricas com os devidos parâmetros, geralmente 1 para ligar e 0 para desligar. O  envia um *"bang"* automaticamente assim que o *patch* é carregado. Ele é usado para carregar informações necessárias para o funcionamento do *patch* automaticamente. Não aceita argumento de nenhum tipo.

Há objetos destinados a fazer ligações "sem fio" entre os objetos e que podem, inclusive, ser endereçados de um *sub-patch* a outro. Isso facilita a visualização e a clareza da construção dos *patches*. Esses objetos chamam-se *send* e *recieve*[5]. São estes:  e  .

O objeto  estima a altura e a amplitude de um som, ambos continuamente como um fluxo de eventos. Opcionalmente, ele nos dá como resultados uma lista de picos senoidais usados para fazer a determinação do *pitch*. Os argumentos que são suportados por este objeto especificam, por exemplo, o tamanho da "janela" de análise, a polifonia máxima, o número de picos no espectro a considerar e o número de picos na saída "crua".

Já o objeto  capta o sinal de áudio e o transmite em sinal RMS[6] como sinal numérico da amplitude em dB (decibel). A saída é nivelada por baixo em zero.

O PD possui um objeto capaz de gerar "rampas" cujos níveis de intensidade e de tempo são determinados por mensagens enviadas a ele. O  gera "rampas" lineares cujos níveis e tempo são determinados por mensagens enviadas a ele. O ~ depois do nome do objeto indica que se trata de um objeto de áudio.

Os objetos que captam os sinais do microfone em tempo real e os convertem de sinal sonoro para sinal digital, e vice-versa, são os objetos  e  .

O objeto  faz a média das informações que "entram" no *inlet* da esquerda de acordo com os parâmetros oferecidos pelo *inlet* da direita. No caso, foi usado o parâmetro de média de 10 amostras para melhor filtrar as informações vindas do  .

---

[5] Do inglês, enviar e receber, respectivamente.

[6] Do inglês, Root Mean Square, ou potência média quadrática.

O snapshot~ foi utilizado como parâmetro de movimentação das esferas. Este objeto capta um sinal de áudio e o converte para um valor de controle. É como se fotografássemos os sinais de entrada com alguma periodicidade.

Foram amplamente usados os objetos delay e pipe, ambos com a função de armazenar uma informação recebida e envia-la depois de um tempo pré-estabelecido na sua caixa de mensagens. Desde o "disparo" para o início da peça, passando pelos *fades* no plano de fundo e nas cores das esferas, até as instruções para o fim da peça, esses objetos foram de fundamental importância.

O freeverb~ é um objeto que foi escolhido para se chegar os resultados estéticos idealizados desde a confecção do projeto inicial. A espacialização e a idéia de continuidade e fusão que este efeito sonoro nos dão foram determinantes na escolha deste objeto.

Os objetos catch~ e throw~ são receptores e preceptores de sinal de áudio "sem fio". Com eles é possível fazer ligações de sinais de áudio sem a utilização das "cordas", utilizadas para ligar um objeto a outro. Com eles também é possível enviar e receber esses sinais de um *sub-patch* a outro. Uma característica importante desses objetos é que eles somam o barramento do sinal, ou seja, eles adaptam os sinais enviados a eles de forma a não *clipar,* não permitir que o sinal exceda os parâmetros de distorção em dB.

Alguns objetos transformam o sinal MIDI em freqüência (em Hertz), como é o caso do objeto, mtof que enviou o sinal ao oscilador osc~ que, por sua vez, emite uma onda coseno.

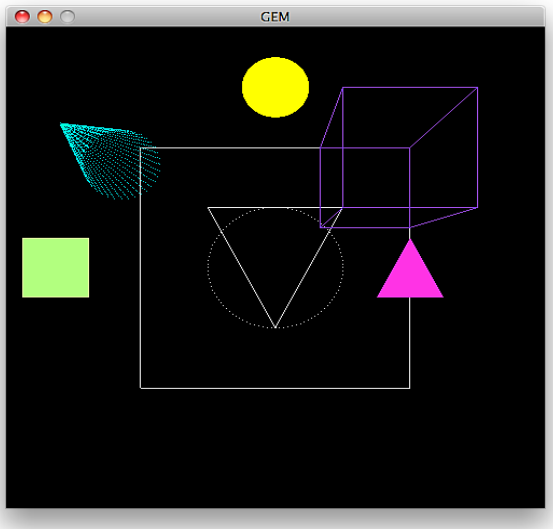
O metro envia uma série de *"bangs"* em uma constante de tempo cuja medida é feita em milisegundos. Seu *inlet* aceita argumentos como liga/desliga e a taxa de *"bangs" por milisegundo.*

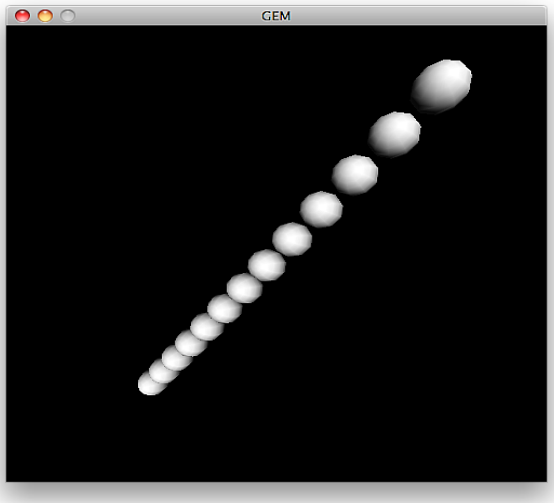
Um objeto muito interessante é o trigger. Ele envia os sinais de seu *inlet* de forma sequencial, da direita para a esquerda e de acordo com os argumentos de criação como *"float", "bang", symbol", "list", pointer", anything",* ficando assim: t f b s l p a. O objeto unpack faz algo parecido, listando os argumentos e distribuindo-os.

Na parte gráfica, um do principais objetos utilizados foi o gemwin, que controla a janela gráfica gerenciamento por meio de envio de mensagens de controle da janela. Depois da janela

criada e renderizada, o objeto gemhead é que conecta os objetos que serão criados com a janela principal. Qualquer objeto gráfico pode ser conectado depois dele e receber comandos de renderização. Alguns desses objetos gráficos são:

circle sphere square square cube triangle cone





No *patch* de Cyrille Henry encontramos alguns outros objetos de programação de extremacomplexidade. É o caso do mass3D e do link3D, que foram utilizados na criação do exemplo 07_corde3D.pd, como mostra a figura a seguir.

Esses objetos tratam da criação, do posicionamento, da massa, do peso, das forças aplicadas às massas e da interação entre as esferas. São usados também os objetos rotateXYZ e translateXYZ que tratam da rotação e translação das esferas.

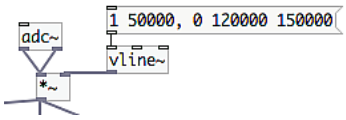
O world_light é um objeto utilizado no *patch.* Ele que cria um ponto móvel de luz com o qual é possível obter efeitos de iluminação e sombreamento.

Um importante efeito estético foi introduzido à peça com o objeto color, que, como o nome já diz, atribui cores aos objetos gráficos.

3. IMPLEMENTAÇÃO

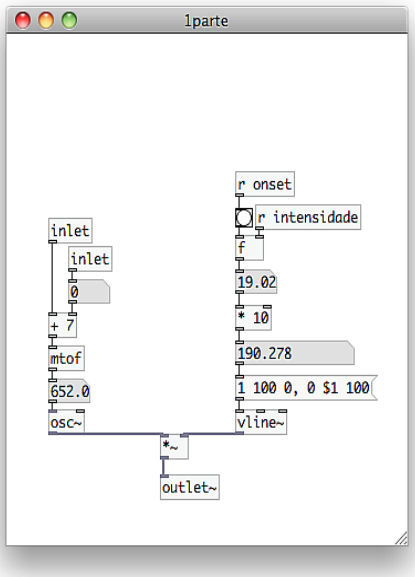
A implementação do projeto se deu em duas grandes partes: sonora e gráfica. A primeira a ser implementada foi a sonora, primeiramente porque seria através dela que a parte gráfica seria estimulada, por estar mais "palpável" diante do que havíamos proposto no projeto inicial e também por termos alcançado um resultado acima do esperado num curto espaço de tempo.

O som estéreo é captado pelo microfone por comando do adc~ e os dois *outlets* são conectados ao *~. Este recebe um argumento do vline~ para que aumente e diminua o volume em dB, primeiro para que não haja, quando da "abertura" do "DSP", um estalo típico; também usamos essas "rampas" de volume para obtenção de um efeito artístico, pondo começo e fim graduais à peça. Vários tempos foram testados até chegarmos a um bom termo. Essa rampa vai de 0 a 1 em 50.000 milisegundos, permanece em 1 por 150.000 milisegundos e cai para 0 novamente em 120.000 milisegundos.



Após sair do *~ o sinal entra no fiddle, do qual utilizamos os dois *outlets* da esquerda para modulação dos parâmetros. Do primeiro *outlet* as informações entram no *subpatch* "1parte". Antes da informação ser transformada de MIDI para freqüência, testamos alguns valores de soma até chegarmos a um resultado sonoro que nos agradasse. Depois, a informação entra no oscilador para depois somar-se à outra parte do *subpatch*, que foi construída a partir do segundo *outlet* do mesmo fiddle. Nesta outra parte, o principal objeto é

o float, que armazena a informação até que um “bang” como argumento no *inlet* da direita indique a mudança ou não para uma nova informação. Esse argumento, no caso, foi a envoltória da onda sonora, obtida com o objeto env~; desses resultados obtivemos melhor resultado com a média das informações. Depois de algumas experimentações chegamos ao valor de 10 para a multiplicação que se segue, pois as freqüências ficavam muito graves com valores menores e muito agudas com valores maiores. O vline~ trouxe uma espacialidade que buscávamos para a saída desses parâmetros de som. Este *patch* inteiro sofre alteração dos argumentos numéricos -5, 3, 7 e 12, que são os intervalos musicais que pretendíamos acrescentar ao som original (fundamental) do clarinete. Este é o *subpatch*:

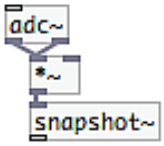


Um dos principais efeitos buscados foi o da espacialidade, da continuidade, algo que nos desce uma sensação de infinitude. O uso do freeverb~ veio de encontro e foi além das expectativas. No entanto, achamos um melhor resultado com o uso de argumento que modifica o “tamanho da sala”, ou seja, altera os parâmetros de reverberação. Depois disto, passamos à incorporação do *subpatch* gráfico.

As maiores mudanças feitas ao *patch* de Cyrille Henry (que foi encorporado ao projeto como *subpatch*) se resumem à movimentação e à interação criadas entre a fileira de esferas e o som da clarinete.

Primeiramente pesquisamos em quais esferas os impulsos que modificariam sua órbita melhor resultariam. Testamos uma a uma, todas as 13 esferas e então resolvemos que apenas

uma esfera recebendo os impulsos não seria de total agrado. Resolvemos então manter a segunda e a sétima esferas como receptoras deste impulso. Na segunda esfera é o eixo cartesiano X que recebe o impulso. Já na segunda esfera é o eixo Z que o recebe. Os sinais deste impulso vem do conjunto:



Aqui no sinal é “fotografado” e enviado a um 

, que “distribui” o sinal às duas caixas de mensagem que farão os 

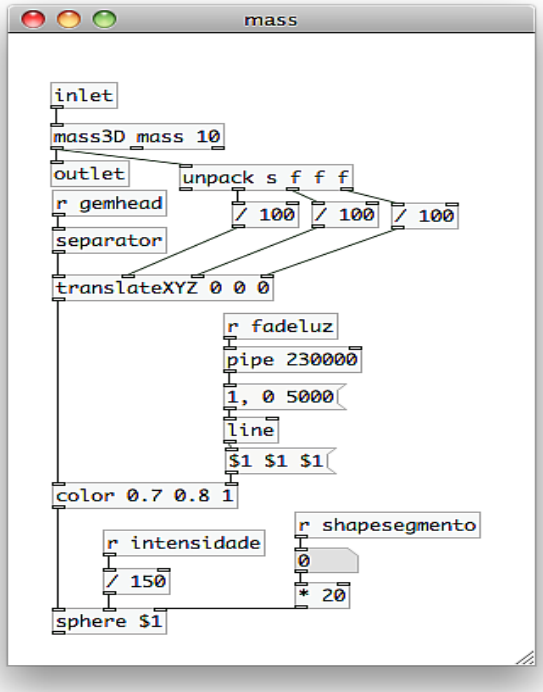
enviarem os sinais para que as esferas oscilem.

Os segmentos das esferas também modulam, ou seja, a esfera tem mais ou menos segmentos de acordo com as informações recebidas pelo *inlet* da direita, que passam pelo 

com um parâmetro de 10 amostras. Como esses números eram muito altos, não se percebia a mudança nos segmentos das esferas. Aplicamos então o objeto matemático de divisão por 15 e, posteriormente, o objeto 

que ignora as casas decimais, expedindo somente números inteiros ao se receptor. O número de segmentos das esferas foi alterado em busca de uma melhor visualização das possibilidades de configuração e do potencial da parte gráfica do PD. Alterando as camadas de preenchimento podemos visualizar a precisão e a qualidade do das janelas gráficas do GEM.

A oitava esfera tem uma participação diferente quanto ao número de camadas; ela teve o seu número de segmentos multiplicado por 20, enquanto as outras chegavam a sete. Chegamos a esses números devido ao seu tamanho variar muito, chegando a ficar muito maior do que o dobro das outras esferas, e, como queríamos uma esfera perfeita, aumentamos consideravelmente o número dos seus segmentos. Eis o *subpatch* da sétima esfera, já com as implementações de intensidade (para variação de tamanho), de segmentos e de *fadeout* de luz:

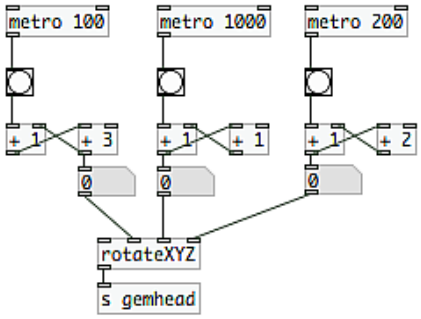


As cores das esferas foram alteradas uma a uma para melhor visualização das características de cada esfera e também como conceito artístico. Ao final da peça, uma a uma, cada esfera deveria escurecer ao ponto de se confundir com o plano de fundo. Obtivemos êxito com essa lógica específica em um *patch* separado, mas como não conseguimos implementa-la no *subpatch* a solução encontrada foi fazer com que cada esfera voltasse a ser branca para então escurecer e alcançar o efeito final desejado.

O *fade out* do plano de fundo tomou a forma final nos últimos dias antes do recital. O esmaecimento foi concebido somente para os 5.000 milisegundos finais da peça. Porém, durante as experimentações com a caixa numérica de argumentos um erro nos apresentou uma nova possibilidade, com a qual o conceito estético muito se identificou e a acabamos por adota-la. Ao argumento de cor do gemwin usamos uma "rampa" de 240.000 milisegundos, que são acionados por um delay, 1.000 milisegundos após o início da peça. Encontramos certa dificuldade nesta etapa pois o objeto loadbang não carregava o argumento 1 para que o fundo ficasse branco logo no carregamento do *patch*. A solução encontrada foi ligar o "*bang"* que dá

início à janela do GEM diretamente ao argumento 1, que torna o fundo branco e dá início ao *fadeout* do plano de fundo.

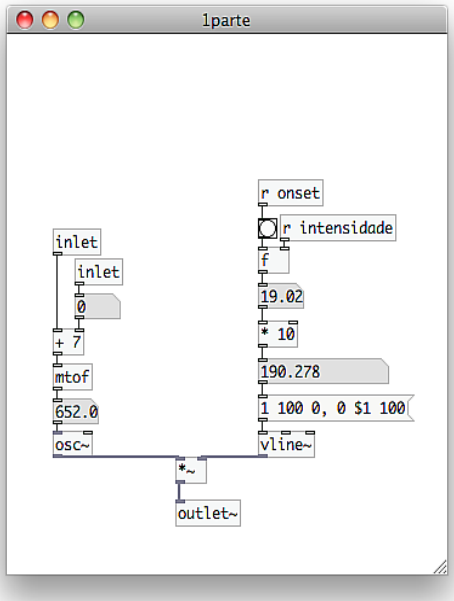
Para que a fileira de esferas tivesse uma rotação contínua, criamos uma espécie de metrônomo que altera os argumentos do rotateXYZ. Essa foi uma solução, ainda que funcional, não muito adequada, pois a soma dos valores tende ao infinito, podendo sobrecarregar o processamento. Este também foi um dos motivos pelos quais resolvemos restringir a peça a um período de tempo de aproximadamente 5 minutos. Este é o "metrônomo":



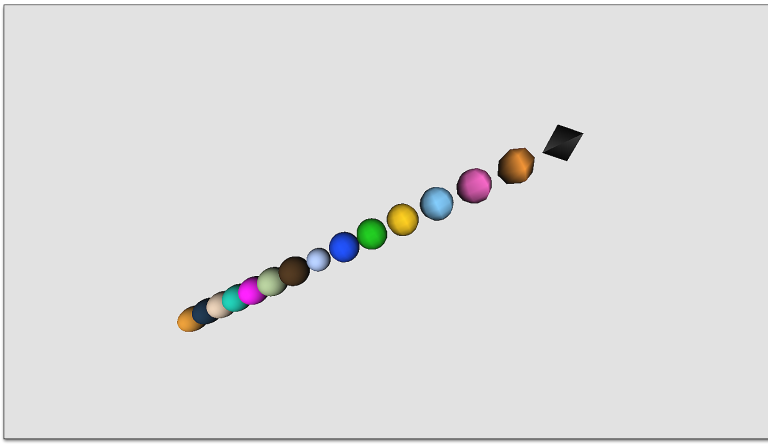
4. RESULTADOS

O funcionamento básico do *patch* pode ser descrito pela captação dos sons do microfone e o uso deste para gerar novos sons e interferir artisticamente na parte gráfica.

Depois de captados, os sons passam pelo fiddle. Foram usadas as duas saídas da esquerda para a molulação e criação dos sons. Do primeiro *outlet* as informações saem e são transformadas de MIDI para frequencia, depois de adicionado o valor 7, numa operação aritimética por meio de um objeto específico para isso. Depois disso a informação é transformada em sinal de áudio e enviada à saída. Nesta mesma saída, há o uso de alguns argumentos como o uso de um vline~ que dá um caráter menos impactante aos movimentos sonoros. Esses argumentos foram modificados a partir do segundo *outlet* do mesmo fiddle, para o qual foi criado um *sub patch* intitulado "1parte", mostrado na figura abaixo.



Deste segundo *outlet* também usou-se um sinal para interferir na movimentação da fileira de esferas através de um snapshot~, impulsionando algumas delas, escolhidas segundo sua posição na fileira. Do oscilador que recebe os sinais do microfone derivamos também uma saída de sinal de áudio para um objeto snapshot~, cuja saída foi enviada ao sinal para as esferas escolhidas. Depois de muitas experimentações, foram escolhidas duas esferas, a segunda e a sétima, pelo efeito visual que causavam. À segunda esfera, resolvemos impor um impulso em seu eixo de rotação X, enquanto que na sétima esfera, impusemos o mesmo impulso no eixo Z, neste, porém com o dobro de intensidade. Uma das infinitas visualizações (isso devido ao seu caráter de interação em “tempo real”) é como se vê na figura abaixo



O resultado estético foi sendo mudado ao longo do desenvolvimento da programação, porém, com ótimos resultados estéticos. O som adquiriu a espacialidade e a fusão entre os sons externos, captados, e os que foram captados e transformados, e que nos agradou ainda mais pela riqueza trimbrística dessa fusão com o clarinete.

Como finalização do projeto foi feito um recital com a participação dos alunos da disciplina com o objetivo de expor artisticamente os trabalhos realizados ao longo do semestre. Nossa participação teve alguns percalços devido à imprevisibilidade da situação, mais especificamente, pelo uso de um projetor com a resolução diferente da usada ao longo do desenvolvimento.

O *patch* funcionava num monitor de 15 polegadas com resolução de 1440x900, no qual não houve nenhum problema de visualização, ao contrário, possibilitando expirimentações e excelente resultado estético. Porém, quando conectado ao projetor com resolução diferente, o *patch* apresentou um deslocamento à esquerda, comprometendo a visualização de seu funcionamento. Alguns conceitos puderam ser visualizados como a rotação do eixo de esferas e as cores aplicadas a elas. Também o *fade out*, do mais claro para o mais escuro, do plano de fundo, como num anoitecer.

A tentativa de se utilizar o argumento *fullscreen* seria uma possibilidade, mas foi descartada devido à sua dificuldade de operação, que impossibilita o retorno ao modo de edição, forçando o fechamento do *patch* e sua nova abertura. Em oportunidade futura, este recurso será utilizado, podendo minimizar tal problema.

Uma nova gravação foi feita, posteriormente ao recital, desta vez no mesmo computador e monitor onde o *patch* foi desenvolvido, garantindo o seu funcionamento tal qual desejado. Desta vez, os resultados foram mais próximos aos esperados, estética e funcionalmente. Ambos os resultados podem ser vistos em
http://www.youtube.com/playlist?list=PL72BACAD45F188F25&feature=view_all

5. CONCLUSÕES

O objetivo deste projeto foi a implementação do processo de criação de uma peça musical que contemple a interação entre homem e máquina, instrumento convencional e computacional. No caso, Clarinete em Bb e o ambiente de programação em tempo real Pure Data Extended.

Ao longo do processo alguns conceitos foram revistos, muitos outros aprendidos, e muitas ideias foram colocadas à prova, umas com sucesso, outras não.

Um caso de sucesso do projeto foi a confecção de uma programação de interação sonora com muita espacialidade e fusão de sons e timbres. A realimentação, o *fade out* e a sensação de continuidade que este resultado sonoro nos trouxe reflete nosso ideal de busca infinita de conhecimento. Como se todo o espaço sideral representasse aquilo que pode ser aprendido pelo homem. É a metáfora da busca infinita pela aprendizagem e pelo conhecimento.

No aspecto gráfico, algumas ideias puderam ser aplicadas, como a coloração das esferas, que também representam o conhecimento, e que, como foi feito durante a implementação, carece de muitos acréscimos. Embora os *fade outs* de luz nas esferas não tenham obtido o resultado desejado, o acaso nos trouxe uma solução inesperada e que se revelou interessante: o retorno à cor branca antes de se tornar completamente escura e fundir-se ao fundo preto.

Melhorias serão necessárias futuramente. É o caso da rotação e translação da fileira de esferas, da resolução do monitor/projetor, do possível uso do *fullscreen*, parâmetros de modulação nas cores, melhorias na modulação do tamanho da(s) esfera(s), utilização de outras formas geométricas com modulações próprias, adição de intervalos musicais mais precisos e a possibilidade de gravação e reprodução de uma ou mais partes da peça.

Há muito o que aprender. Há muito o que melhorar. A interação entre homem e máquina através da música aponta para um futuro mais inteligente e belo. E entre 0 e 1, DÓ e SI, tudo é possível!

"O que está em jogo não é a transmissão daquilo que se inventa,
mas antes, a transmissão do poder de inventar."

Juan David Nasio

Referências Bibliográficas

DALBEM, C. M. *Estudo e descrição da linguagem Pure Data.,* Instituto de Informática da UFRGS. 2010.

FUGUEIRÓ, C.; KROGER, P. *Elementos e idéias para uma metodologia de estruturação de eventos temporais em Pure data,* XVII Congresso da Associação Nacional de Pesquisa e Pós-Graduação em Música (ANPPOM) São Paulo. 2007.

http://www.chnry.net/ch/?lang=fr (acessado em junho de 2012)

http://www.phy.mtu.edu/~suits/clarinet.html. (consultado em 02/05/2012)

http://puredata.info/ (acessado em junho de 2012)

MIRANDA, P. A. *Performer e meios eletrônicos: aspectos da interatividade na música eletroacústica mista.* Departamento de Música e Artes Cênicas da Universidade Federal de Uberlândia. 2010.

OLIVEIRA, L. C. de; FURLANETE, F.; GOLDEMBERG, R.; MANZOLLI, J. *Modelo Empírico da Sonoridade da Clarineta Aplicado como Ferramenta Composicional*, XVI Congresso da Associação Nacional de Pesquisa e Pós-Graduação em Música (ANPPOM), Brasília. 2006.