

Revista Música Hodie, Goiânia - V.13, 362p., n.1, 2013

PRIMEIRA IMPRESSÃO

Altar ou a Resposta dos Deuses (2010), para temple bell e eletrônica em tempo real (Pure Data)

Celso Cintra (UFU, Uberlândia, MG, Brasil)
celsocintra@demac.ufu.br

Altar or The Answer from the Gods (2010), for temple bell and live-electronics (Pure Data)

Altar ou a Resposta dos Deuses surgiu de uma encomenda do percussionista e colega professor da Universidade Federal de Uberlândia, Prof. Dr. Cesar Traldi, para quem a peça é dedicada. O nome da peça é derivado da utilização religiosa do instrumento. Trata-se de um *temple bell*, que também é comumente conhecido como *altar temple bell*. Em seu uso religioso no budismo ele é colocado em um altar e é tocado durante as orações. A ideia da peça é a de que ao orar se estabelece um contato com os deuses e suas resposta nem sempre são aquelas que gostaríamos de ouvir.

A peça deve durar entre 4 e 8 min, estabelecida pelo intérprete no momento da execução. Toda a estrutura formal foi programada em *Pure Data* e ao definir sua duração total, o *patch* calcula proporcionalmente as durações das seções. Dividida em três seções, as sonoridades processadas pelo computador ficam cada vez mais próximas das ações empreendidas pelo intérprete. A única instrução musical dada ao intérprete, que deve ser repetida em todas as execuções, é a de que ele deve atacar forte o instrumento e esperar a resposta do computador, que grava e reproduz este ataque após 13 segundos, tempo aproximado de extinção total do som do instrumento, gerando expectativa tanto no intérprete quanto na plateia. A partir daí toda a música é uma improvisação livre do intérprete em diálogo com as repostas geradas pelo computador. Estas respostas são processamentos em tempo real das sonoridades emitidas pelo intérprete, de maneira que não há, nesta peça, nenhum som estranho previamente gravado antes de sua execução.

Quanto à textura, *Altar ou a Resposta dos Deuses* possui basicamente duas camadas sonoras. Uma das camadas é estável a cada execução, formada a partir de um primeiro ataque, que é repetido primeiro como um eco, elevado em sua altura em 1Hz, depois de forma inversa e prolongada, o que o torna mais grave devido a extensão da duração deste som inverso, que é proporcional à duração da peça, e logo em seguida, quando chega ao seu início, ele é repetido em sua forma original, também prolongado e conseqüentemente transposto ao grave, proporcionalmente à duração da música. A outra camada é dinâmica e, em conjunto com a camada estável descrita acima, define também a estrutura formal da peça. Em termos sonoros, esta camada pode variar bastante a cada nova apresentação, pois depende dos sons executados ao instrumento pelo intérprete durante sua improvisação livre e do processamento do computador aplicado a estes sons que são gravados pelo programa. Esta camada dinâmica se inicia simultaneamente à resposta do computador ao ataque inicial do músico, e dura até a seção áurea da duração total.

Quando o intérprete começa sua improvisação, após esperar a resposta de seu primeiro ataque pelo computador, um *subpatch* começa a gravar 10 arquivos de áudio de 10 segundos cada ininterruptamente. Esses arquivos são então disparados, com transformações em sua velocidade de execução, por meio de processos randômicos. A este *subpatch* de gra-

vação de arquivos é sobreposto outro que chamei de “chuva” de *delays*. Ele se inicia na seção áurea da seção áurea, e termina simultaneamente com o *subpatch* gravador e reproduzidor de arquivos acima. Doze geradores de *delay*, cada um com uma duração diferente, são acionados randomicamente por um metrônomo de 1 bpm (ou 1000 ms). A partir daí cada duração de *delay* muda a duração do metrônomo, mudando assim a velocidade de acionamento randômico dos *delays*.

Com o término dos dois *subpatches* acima, inicia-se um terceiro *subpatch* que durará até o fim da música. Neste último *subpatch* é utilizado objeto *fiddle* do *Pure Data*. Ele é utilizado para analisar o som que o está alimentando identificando e decompondo seus cinco primeiros parciais que foram programados para soar como senoides. Estas senoides foram tratadas de forma que sua amplitude é inversamente proporcional à amplitude do som captado. Daí que quanto mais forte a dinâmica empregada pelo músico, menos se ouve determinados parciais e vice-versa. Cabe ao músico equilibrar sua execução com as respostas do processamento, podendo assim controlar com maior precisão o som resultante da combinação do instrumento com as senoides emitidas pelo computador.

Celso Luiz de Araujo Cintra - Compositor. Doutor em Música pela ECA/USP. Mestre em Música pelo Instituto de Artes da UNESP, onde concluiu o Bacharelado em Composição e Regência e foi aluno de Edson Zampronha, Flo Menezes e Edmundo Villani-Côrtes. Pesquisa a obra do musicólogo francês Alain Daniélou, cujo trabalho se baseia no estudo de diversas músicas orientais, principalmente a música indiana. Atualmente é professor do Curso de Música da Universidade Federal de Uberlândia - UFU, onde ministra as disciplinas Teoria da Música, Harmonia e Filosofia da Música e é coordenador dos laboratórios: LABMUL - Laboratório de Ensino e Pesquisa em Multimídia; LASON - Laboratório de Ensino e Pesquisa em Produção Sonora ligados ao NUMUT - Núcleo de Música e Tecnologia.

Altar ou A Resposta dos Deuses

para *temple bell* e eletrônica em tempo real (Pure Data).

Dedicada a Cesar Traldi

Celso Cintra
14072010

Altar ou A Resposta dos Deuses é uma peça de improvisação livre, em que o músico interage com os processamentos sonoros gerados pelo computador.

Instruções:

I. É necessário para esta peça três *patches*, que deverão permanecer na mesma pasta: *altar_count.pd*; *altar_inicial.pd* e *altar_rec.pd*;

II. Durante o decorrer da peça serão gerados 11 arquivos *.wav*, por isso apague todos os arquivos *.wav* produtos de execuções anteriores que porventura estejam na pasta dos *patches*;

III. A peça deve durar entre 4 (quatro) e 8 (oito) minutos, o músico decide;

IV. Abra o *patch* *altar_inicial.pd*;

V. Siga as instruções do *patch*, descritas abaixo:

1 Inclua a duração desejada para a música em minutos e segundos: aqui você deve entrar com um número para os minutos (de 4 a 8) e pressionar a tecla [enter], e um número para os segundos (de 0 a 59) e pressionar a tecla [enter].

- *Atenção: mesmo que o número zero já apareça no lugar dos segundos é necessário digitá-lo e pressionar a tecla [enter], para que o programa faça os cálculos necessários para a música;*

2 Ligue o áudio do Pd com o mouse no objeto *toggle*, de forma que apareça um "x" em seu centro;

- 3 Assim que estiver preparado, pressione a barra de [espaço] para iniciar a contagem regressiva;
- 4 A contagem regressiva em segundos começa no número (- 4) e vai até zero. No zero, ataque forte no *temple bell* e deixe soar;
- 5 Aguarde o computador responder;
- 6 Assim que ouvir o computador responder, comece a improvisar livremente, buscando interagir com as sonoridades que surgirão no decorrer da peça;
 - *Todos os sons que surgirão serão derivados dos sons que você fizer, divirta-se à vontade!*
- 7 Observe o tempo decorrido da música que aparece no cronômetro logo abaixo no *patch*, fique atento para que a música dure o quanto você digitou acima;
 - *Assim que o cronômetro chega no tempo previsto ele congela a contagem e desliga o patch, porém o microfone continua aberto;*
- 8 Ao terminar a música desligue o áudio do Pd (item 2 acima) com o mouse no objeto *toggle*, de forma que o "x" de seu centro desapareça, para evitar sonoridades indesejadas do microfone.
- 9 Caso vá fazer uma nova execução da peça, feche o *patch* e apague da pasta todos os arquivos *.wav* que foram gerados (conforme item II acima). Abra novamente o *patch* e siga todas as instruções a partir do item 1.

Uberlândia, 14 de julho de 2010.

O *patch* em Pure Data para esta música está disponível para *download* em:

<<http://www.numut.iarte.ufu.br/downloads-partituras-e-patches>>