

# *Live coding: um algoritmo gerador de sonoridade tonal no A Study in Keith (2009), de Andrew Sorensen*

Guilherme Martins Lunhani  
Orientador: Prof. Dr. Luiz Eduardo Castelões<sup>1</sup>

Instituto de Artes e Design – Universidade Federal de Juiz de Fora  
Juiz de Fora, MG <gcravista@gmail.com>

1 de abril de 2016

# Sumário

## Preâmbulo

O tema da improvisação de códigos, como um universo de conceitos, surgiu de uma experiência com um código em linguagem *python* (<<https://www.python.org>>), útil para gerar um mapa de termos, como ilustrado na figura abaixo.

# Preâmbulo



Figura : Nuvem de palavras do ??), 1º Congresso Internacional de Live Coding. Fonte: autor.

# Introdução

Abordagem do conhecimento das artes computacionais no *Livecoding*:

- Universo de conceitos

## Objetivo

- Investigar um Universo de Conceitos sobre a *improvisação de códigos (live coding)*;
- Investigar um Espaço de Conceitos, historicamente restrito, sobre a improvisação de códigos;
- Investigar um método de análise/criação para uma improvisação de códigos;
- Investigar um Espaço Conceitual de uma improvisação de códigos, *A Study in Keith* (2009) de Andrew Sorensen, e seu algoritmo musical em um ciclo de transformação.

## Definições de base da Improvisação de códigos

*Live coding* é uma técnica artística de improvisação. Pode ser empregada em muitos contextos diferentes de performance: dança, música, imagens em movimento e mesmo tecelagem. Eu concentrei minha atenção no lado musical, que parece ser o mais proeminente (??, p. 117)

## Definições de base da Improvisação de códigos

A definição denota a aplicação em qualquer outra área, não apenas como metáfora, mas como estratégia de gerenciamento de fluxos criativos, assistidos por computador



# Tecelagem

O grupo *Weaving codes* foi formado para a (...) *investigação de padrões a partir das perspectivas de tecelagem e música, e através do desenvolvimento de uma linguagem de computador e código para descrever a construção de tecidos (??)*

# Tecelagem

```
(twist 3 4 5 14 15 16)
(weave-forward 3)
(twist 4 15)
(weave-forward 1)
(twist 4 8 11 15)
```

```
(repeat 2
  (weave-back 4)
  (twist 8 11)
  (weave-forward 2)
  (twist 9 10)
  (weave-forward 2)
  (twist 9 10)
  (weave-back 2)
  (twist 9 10)
  (weave-back 2)
  (twist 8 11)
  (weave-forward 4))
```

## Tecelagem

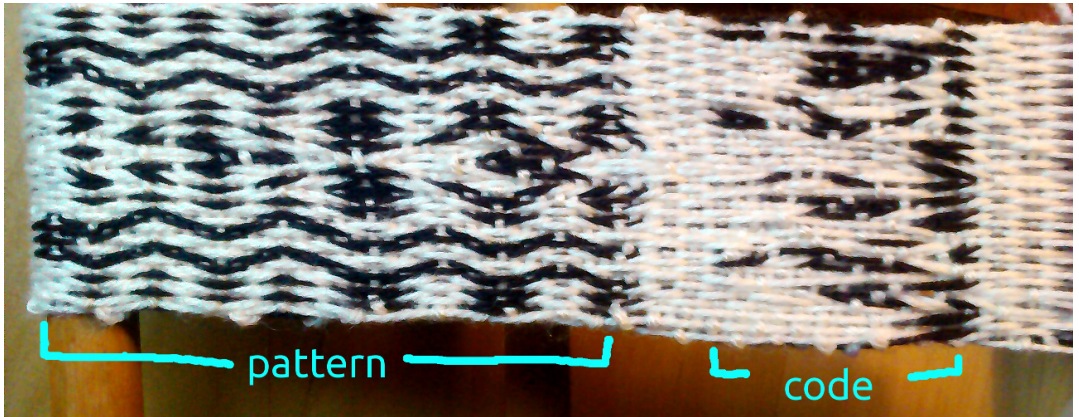


Figura : Tecido resultante da prática *Weaving code*. Fonte: ??).

# Tecelagem

- Apresentação de dois importantes personagens para o *Livecoding*: Alex McLean e Dave Griffiths (*Slub*)
- Observação de possíveis rituais de *livecoding*, como apresentação artística informal/formal.

## Tecelagem



Figura : Performance no Foam Kernow. **Fonte:** ??).

A banda *Slub* começou em 2000, como uma colaboração entre Adrian Ward e Alex McLean. A premissa do duo era utilizar a atividade de programação para realização de uma Música Eletrônica de Dança<sup>1</sup>. Sua primeira reunião foi em 2001, no *Paradiso club* em Amsterdã, durante o festival *Sonic Arts*. Em 2005 Griffiths se juntou ao duo durante o festival *Sonar*, o que abriu espaço para o desenvolvimento de uma estética de videogames (??, p. 138–140).

## Live coding como dança

- Dança como fim de uma improvisação de códigos → Algorave
- Dança como meio (coreografia) de uma improvisação de códigos → Kate Sichio

# Algorave

*Algorave 'começou como uma piada', de acordo com Alex McLean, um pesquisador de música computacional e um dos três de uma banda chamada Slub, que têm improvisado códigos por 13 anos. Ele veio com um termo enquanto conduzia uma gig em Nottingham com seu amigo Nick Collins (que tocava "datapop" sob o nome Sick Lincoln) no final de 2011. 'Nós sintonizamos em uma estação pirata tocando happy hardcore, e nós pensamos que seria bom programar alguma música rave.' Deste então, McLean organizou oito algoraves informais no mundo. (??)*



## Algorave

*Algorave não é sustentado exclusivamente por live coders, mas estes têm mantido uma forte presença em todos os eventos até agora. É assim talvez porque a tradição do live coding de projetar telas motiva todo o esforço; onde algoritmos não estão visíveis por períodos de tempo durante uma algorave, se corre o risco das coisas parecerem muito como um evento de música eletrônica padrão. (??, p. 356)*

Recorte histórico da Música Eletrônica para Dançar (??):

- 1992: Charles Ames disponibiliza o *Cybernetic Composer*;
- 1994: o duo *Koan*, formado pelos DJs Daniel Roeth e William Grey, realizam adaptações para entretenimento com base no *ambient music* de Brian ??);
- 1997: *Aphex Twin* (Richard David James) cria em o termo *live club algorithm*;
- 1999: o protocolo para edição audiovisual ao vivo *bbcutf* (??) é incluído nos *opcodes* do *CSound*;
- 2000: o então duo *Slub* realizam performances, autodenominadas *generative techno*;
- 2001 é identificada a utilização de redes neurais para composição de padrões semelhantes ao *drum'n'bass*;
- 2004: é fundado o TOPLAP em uma casa noturna de Hamburgo.

## Algorave



Figura : Performance do duo Canute (Karlesruhe, 2015) Fonte: ??).

onte: ??).



Figura : Performance de Alexandra Cárdenas (Londres, 2013) **Fonte: ??).**

## Algorave

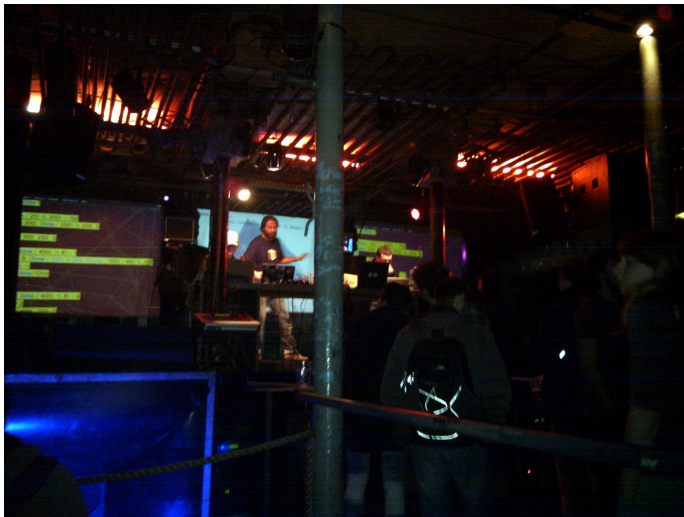


Figura : Performance do duo Mico Rex (Londres, 2013) **Fonte: ??).**

# Coreografia

Kate Sichio

- *Hacking the Body/Hacking Choreography*;
- *Hacking Coreography v.01 e v.02*);
- *Hacking The Body 2.0*;

# Coreografia

Para ??, cap. 1, p. 3), do ponto de vista computacional, a partitura coreográfica é mais próxima do código escrito em uma linguagem de computador do que a partitura musical tradicional.

# Coreografia

- Assimilação do pensamento algorítmico na Dança → *Sensibilidades Computacionais* (??, ?? apud ??, ??, cap. 1, p. 3;p. 31)
- *mecanismos de generalização e abstração, representação da coreografia e dança como computação* (??, cap. 1, p. 2–4) → estratégias elaboradas por coreógrafos como Merce Cunningham, Trisha Brown, Bill T. Jones, e William Forsythe.



## Coreografia

*Esta sensibilidade computacional é presente em dois níveis em um trabalho destes coreógrafos. Primeiramente, em seus processos coreográficos – os sistemas, métodos, e notação através dos quais os coreógrafos criam a dança. Segundo, no trabalho ele mesmo, finalizado, que aparece no palco e é interpretado pelo observador. As primeiras invenções e proclamações de Cunningham – a democracia do espaço do palco, e a redescoberta do que está atrás do dançarino como ponto de origem do movimento – pode ser interpretado como generalizações do tipo; qualquer ponto do palco é a “frente”, e conectado por um conjunto de articulações pode ser pensado como um membro. O que eram constantes, uma vez especificados em uma descrição rígida, se tornam variáveis em uma estrutura generativa.(??, cap. 1, p. 2–4)*

# Hacking Coreography

Duas experiências com uma Partitura de Eventos do artista Alison Knowles (mais especificamente a peça de performance #8, de 1965)

# Hacking Coreography

Divida uma variedade de objetos em dois grupos.

Cada grupo é rotulado com "tudo".

Estes grupos podem incluir diversas pessoas.

Existe uma terceira divisão do palco, objetos vazios, rotulados com "nada".

Cada um dos objetos é "alguma coisa".

Um executante combina e ativa os objetos das seguintes maneiras para qualquer duração desejada de tempo :

- \* "alguma coisa" com "tudo"

- \* "alguma coisa" com "nada"

- \* "alguma coisa" com "alguma coisa"

- \* "tudo" com "tudo"

- \* "tudo" com "nada"

- \* "nada" com "nada"

## Hacking Coreography v.01

*Depois que a partitura foi completada, contudo, ela foi hackeada. Isso significa que o executante tenta de alguma forma contornar as instruções originais. Isto foi feito sem preparações prévias e a audiência assistiu isso se desdobrar enquanto era realizada. Nesta primeira performance, o papel e os rótulos foram rasgados para criar novas palavras e categorias (...) Então ao invés de “nada”[Nothing], foram formados dois grupos, “não”[No] e “coisa”[Thing]. (??, p. 31)*

## Hacking Coreography v.02 I

```
/Dance/  
set up()  
{  
  dance a centre, right  
  dance b centre, left  
}  
  
movement()  
{  
  move1 (dance a = rotate) (dance b = jump)  
  move2 (dance a = brush) (dance b = lie down)  
  move3 (dance a = push) (dance b = run)  
  move4 (dance a = step) (dance b = kneel)  
}  
  
coreography()  
{  
  if (dancer a = rotate right 180)  
  then both jump = 2 feet to 1  
  if (dancer b = travels)  
  then brush = right foot  
}
```

## Hacking Coreography v.02 II

```
run(){  
  move1  
  move4  
  move4  
  move1  
  move2  
  move3  
  move1  
  move2  
  move3  
  move4  
}
```

```
/hack/  
{  
  if (dancer a = kneel)  
    dancer a = kneel  
  if (dancer a = rotate)  
    dancer b = rotate opposite direction  
}
```

## Hacking The Body I

*Esta peça é uma exploração de eletrônica codificada ao vivo e movimentos improvisados. Uma dançarina veste uma peça de atuadores hápticos. Estes atuadores são programados em tempo-real via OSC<sup>2</sup> para 'zunar' sobre os lados direito e esquerdo da dançarina para indicar qual lado do corpo a dançarina deve mover. A partitura é codificada ao vivo pela coreógrafa enquanto a dançarina responde por uma retroalimentação háptica. Esta peça explora o live coding de corpos, e movimento como saída, ao invés de saídas sonoras ou visuais como encontrado em muitas execuções de live coding*

## Hacking The Body II



**Figura :** Dançarina (anônima) controlada por Kate Sicchio (2015) através de uma codificação improvisada. **Fonte:** <<https://www.youtube.com/watch?v=uAq4BAbvRS4>>.



# Música computacional

- FooBarBaz (Aspecto histórico, em um evento nacional)
- Magno Calliman (Aspecto performático, em um evento nacional)
- Supercopair (Aspecto telemático, com pesquisadores brasileiros)

# Foo Bar Baz I

FooBarBaz é um grupo de improvisação de códigos formado por Gilson Beck, Renato Fabbri, Ricardo Fabbri e Vilson Vieira. Sua primeira apresentação foi durante o Festival Contato 2011. Os membros são ativos em um laboratório virtual conhecido como *labMacambira*(<http://labmacambira.sourceforge.net/>).

## Foo Bar Baz II



Figura : Festival Contato, 2011. Fonte: ??).

# Foo Bar Baz

```
["samples/fx/s20.wav"] => Foo.name;  
[0.] => Foo.prop;  
[.25, .15] => Foo.rate;  
[2., 1., 1., 4.] => Foo.du;  
[.8] => Foo.gain;
```

# Foo Bar Baz

Funcionamento → *grade temporal* (TimeGrid) → Graham Coleman  
(<http://www.dtic.upf.edu/~gcoleman/>).

# Foo Bar Baz I

*//basic timing operations abbreviated*

```
public class TimeGrid {
```

```
    dur beat;
```

```
    dur meas;
```

```
    dur sect;
```

```
    int nbeat;
```

```
    int nmeas;
```

*//phase and magnitude of offset*

```
    float measPhase;
```

```
    dur measOffset;
```

```
    ...
```

*//sync to beat*

```
    fun void sync() {
```

```
        beat - (now % beat) => now;
```

```
    }
```

```
    fun void sync(dur T) {
```

## Foo Bar Baz II

```
    T - (now % T) => now;
}

//how long to sync to this duration
fun dur syncDur(dur T) {
    return (T - (now % T));
}

//minimum time
fun dur tmin(dur a, dur b) {
    return (a < b) ? a : b;
}
...
}
```

# screenBashing I

Performance de *screenBashing* de Magno Caliman (ver 9), realizada durante o XIII ENCUN<sup>3</sup>.



## screenBashing II

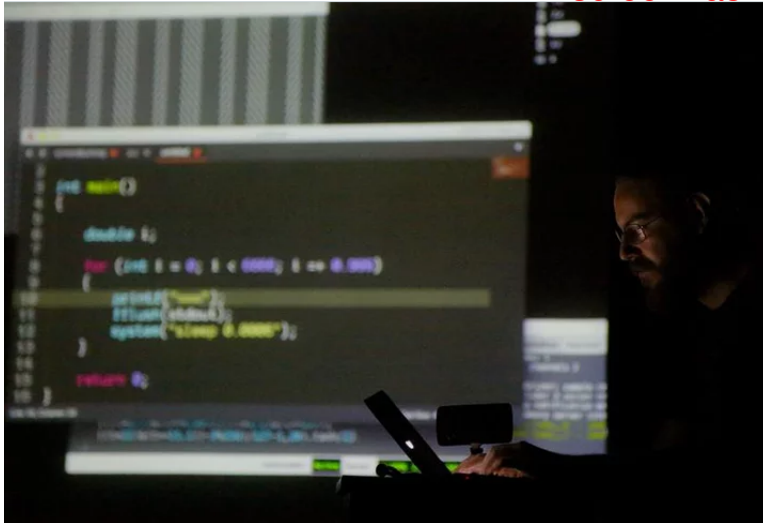


Figura : Performance de *screenBashing*. Fonte: <https://vimeo.com/148626379>.

## screenBashing

*Não acho totalmente adequado considerar como puramente um improviso. (...) acredito que se imaginarmos um continuum, uma linha onde de um lado vc tem uma improvisação totalmente livre (impossível de se alcançar, claro) e do outro uma composição 100% determinada (tão impossível quanto), acredito que screenBashing está posicionada mais à direita. . .*

# screenBashing I

## Código-fonte:

```
# include <stdio.h>

int main()
{
    double i;

    for (int i=0; i<; i < 6666; i+=0.999)
    {
        printf("\\\\ \\ \\ \\ // \\");
        fflush(stdout)
        system("sleep 0.0006")
    }
}
```

## Execução:

## screenBashing II

```
# gcc - compilador
# a.c - arquivo em linguagem c
# -o - escreve o resultado em um outro arquivo
# a - arquivo binario alvo
\$\> gcc a.c -o a
magno.c: In function 'main':
magno.c:7:12: error: conflicting types for 'i'
```

Resultado:

A 10x10 grid of 100 slanted, stylized characters resembling 'A' or 'V'. The characters are arranged in a regular pattern, with each character slanted to the right. The grid is composed of 10 rows and 10 columns.

## screenBashing

- O código sonoro foi pouco exposto
- Valorização ideológica. → repetição dos algarismos seis e nove → *black metal* abstrato (?)
- uma segunda notação (Capítulo 3), mais enxuta e sem erros, permite apontar a questão ideológica

## Código-fonte:

```
# include <stdio.h>
int main(){
    int i=0;
    for (;;i++){
        printf("=====  _____");
        fflush(stdout);
        // 30 frames por segundo
        // ou um limite aproximado da
        // percepcao humana (1/30)
        system("sleep 0.033");
    }
}
```

## Resultado:

```
=====  _____=====  _____=====  _____=====  _____=====  _____
 _____=====  _____=====  _____=====  _____=====  _____
__=====  _____=====  _____=====  _____=====  _____=====  _____
===  _____=====  _____=====  _____=====  _____=====  _____
_____=====  _____=====  _____=====  _____=====  _____=====  _____
_=====  _____=====  _____=====  _____=====  _____=====  _____
=  _____=====  _____=====  _____=====  _____=====  _____=====  _____
_____=====  _____=====  _____=====  _____=====  _____=====  _____
```

## Telepresença e espaços virtuais

- Uma rede local, com computadores diferentes, mas com os improvisadores fisicamente próximos;
- Uma rede remota, privada, que comunica um conjunto de pessoas fisicamente distantes ??, p. 152–153)
- O navegador de *internet* se torna o ambiente virtual de criação musical (??).

# Telepresença e espaços virtuais

*live coding sessions* → sessões de improvisações que ocorrem em encontros, simpósios e *workshops*, ricamente documentadas (<https://supercollider.github.io/archive>).



## Telepresença e espaços virtuais

Performance telepresencial em 2014, por Ben Swift, Henry Gardner e Andrew Sorensen, realizada entre dois intérpretes-programadores localizados na Alemanha e Estados Unidos usando um servidor SSH localizado na Austrália (??, p. 152–153)

# Telepresença e espaços virtuais

*Supercopair* (<<https://github.com/deusanyjunior/atom-supercopair>>) → ambiente cooperativo de improvisação de códigos.

# Telepresença e espaços virtuais

Execuções remotas em navegadores de *internet*, com auxílio da biblioteca *WebAudio API* (<<https://dvcs.w3.org/hg/audio/raw-file/tip/webaudio/specification.html>>).

# Telepresença e espaços virtuais

- *Gibber* (??) (<<http://gibber.mat.ucsb.edu/>>);
- *Wavepot* (<<http://www.wavepot.com>>);
- *Vivace* (??) (<<http://void.cc/freakcoding>>)
- *Termpot* (??) (<<http://jahpd.github.io/termpot>>)

# Definições Históricas da Improvisação de códigos

Exemplos proto-históricos → possuem similaridades com o conjunto de regras práticas publicadas por (??)

# Definições Históricas da Improvisação de códigos

- Pietro Grossi
- Baía de São Francisco

# Definições Históricas da Improvisação de códigos

Evento Histórico: Live Algorithm Programming and Temporary Organization for its Promotion

## Pietro Grossi – Computer Concerto

*Grossi começou a se interessar por música computacional durante a primeira metade dos anos 60, quando ele quando ele organizou um programa de rádio centrado em torno da "música inovadora" em geral (??). Contudo a primeira experiência de Grossi com um computador foi em Milão, no Centro de Pesquisa Elétrica da Olivetti-General. Aqui, auxiliado por alguns técnicos internos e engenheiros, ele conseguiu compor e gravar alguns de seus primeiros trabalhos em música computacional. Eles foram, em sua maior parte, transcrições de música clássica ocidental. Contudo, houve algumas exceções, por exemplo, uma faixa chamada Mixed Paganini (??, p. 126)*



## Pietro Grossi – DCMP (*Digital Computer Music Program*)

*Um dos mais importantes aspectos do trabalho de Grossi foi que todas intervenções eram instantâneas: o operador não tinha que esperar pelo computador terminar todas operações requisitadas, e depois ouvir os resultados. Cálculos de dados e reprodução sonoras eram simultâneos. **Esta simultaneidade não era comum no campo da Computer Music daquele tempo, e Grossi deliberadamente escolheu trabalhar desta forma, perdendo muito no lado da qualidade sonora. Seu desejo era poder escutar os sons resultantes imediatamente** (??, p. 126)*

# Pietro Grossi – DCMP (*Digital Computer Music Program*)

“preguiçoso” (*pigro, lazy*) → *reflexividade*

## Pietro Grossi – DCMP (*Digital Computer Music Program*)

*habilidade de um programa manipular como dados algo que representa o estado do programa durante sua própria execução, o mecanismo para codificação de estados de execução é chamado reificação.(??, p. 1).*

## Pietro Grossi – DCMP (*Digital Computer Music Program*)

*Naquele tempo, os recursos de cálculo eram escassos e, para obter a reprodução em tempo-real citada, era necessário pedir por pouca quantidade de dados. (...) A síntese de timbres necessita de uma quantidade imensa de dados, e então a escolha foi descartá-la temporariamente, e todos os sons eram reproduzidos com o timbre de uma onda quadrada. Esta forma de onda era gerada por extração do estado binário do pin de saída da placa mãe que controla o programa. Essa saída tinha um único bit, e então a onda sonora gerada era o resultado desta mudança do estado binário. Desta forma, o computador não emprega quaisquer recursos para calcular a síntese sonora, economizando-os para o processo de produção musical. Grossi não estava interessado na qualidade da saída sonora em sua primeira fase em Pisa. O que importava particularmente era a capacidade em trabalhar em tempo real, ou, em outras palavras, para ter a escolha de escutar imediatamente ao que ele escreveu no teclado do terminal de vídeo (??, ?? apud ??, ??).*

## Pietro Grossi – TAU2-TAUMUS (*Terminale Audio 2<sup>a</sup> versione*)

Grossi foi capaz de implementar melhorias de timbre, digitalmente controladas por doze vozes, com síntese analógica.

## Pietro Grossi – TAU2

*Estas doze vozes eram divididas em três grupos, compostos de quatro canais cada. O operador poderia atribuir um timbre diferente para cada grupo, que era modulado usando síntese aditiva com sete sobretons. Cada sobretom era controlado individualmente pelo programa.*

## Pietro Grossi – TELETAU

O TAU2-TAUMUS sofreu uma considerável modificação, sendo que era possível controlar o sistema digital-analógico remotamente. O novo programa foi batizado de TELETAU ??, p. 128–129)

## Pietro Grossi – TELETAU

*Grossi fez sua primeira experiência do tipo durante uma conferência de tecnologia em Rimini em 1970, onde o músico reproduzia algumas de suas composições, bem como sons randômicos, empregando um terminal de vídeo conectado pelo telefone para o computador da CNR em Pisa. A RAI, empresa de radiodifusão italiana, emprestou suas pontes de rádio [Comunicação entre duas antenas] para enviar sinais sonoros entre Pisa e Rimini. É como se fosse o primeiro experimento de telemática musical no mundo (??, p. 129).*



# Baía de São Francisco

- The League of Automatic Composers
- The Hub
- Ron Kuivilla

# Baía de São Francisco

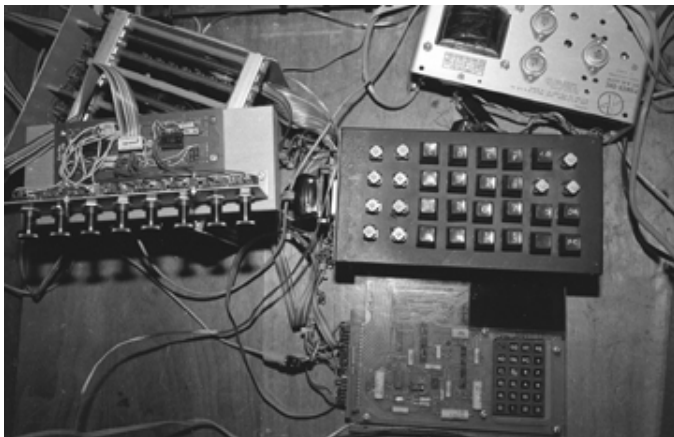
*Mills College* em Oakland.

## Baía de São Francisco

*Com o florescimento da indústria de computadores pessoais na Baía de São Francisco, o acesso às novas tecnologias e pessoas que desenvolveram elas era talvez o melhor no mundo. (...) Esta também é a cultura que deu ao mundo a música “New Age”, uma versão aguada e comercializada das músicas com base em modos e drones que Terry Riley, Pauline Oliveros, e LaMonte Young inventaram durante os anos cinquenta e sessenta. Mas a música feita na Costa Oeste também incluíam improvisações barulhentas e livre de restrições, que sobraram das revoluções contra-culturais dos anos 60(??).*

# The League of Automatic Composers

- Segunda metade da década de setenta: Jim Horton começa a adquirir micro-controladores KIM-1 (*Keyboard Input Monitor*, <http://www.6502.org/trainers/buildkim/kim.htm>), com interesses musicais.
- David Behrman e John Bischoff, Rich Gold, Cathy Morton, Paul Robinson, e Paul Kalbach.



**Figura :** Sistema de música computacional de John Bischof *circa* 1980. Foto: Eva Shoshanny<sup>5</sup>. **Fonte: ??).**

## The League of Automatic Composers

Na primavera de 1979, montamos uma série quinzenal regular de apresentações informais sob os auspícios da *Bay Center for the Performing Arts*. Todos outros domingos à tarde passávamos algumas horas configurando nossa rede de KIMs na sala *Finnish Hall*, na Berkeley, e deixávamos a rede tocando, com retoques aqui e ali, por uma ou duas horas. Os membros da audiência poderiam ir e vir como quisessem, fazer perguntas, ou simplesmente sentar e ouvir. Este foi um evento comunitário de tipos como outros compositores aparecendo, tocando ou compartilhando circuitos eletrônicos que tinham projetado e construído. Um interesse na construção de instrumentos eletrônicos de todos os tipos parecia estar "no ar". Os eventos da sala *Finn Hall* foram feitos para uma cena com paisagens sonoras geradas por computador misturado com os sons de grupos de dança folclórica ensaiando no andar de cima e as reuniões ocasionais do Partido Comunista na sala de trás do edifício velho venerável. A série durou cerca de 5 meses que eu me lembre.(??)

# The League of Automatic Composers

- Tim Perkis
- Sam Ashley, Kenneth Atchley, Ben Azarm, Barbara Golden, Jay Cloidt e Brian Reinbolt.

# The League of Automatic Composers

“rede de composições”



## The League of Automatic Composers

*Os membros da liga geralmente adaptavam composições solo para usar dentro da banda. Estes solos eram desenvolvidos independentemente por cada compositor, e eram tipicamente baseados em esquemas de algoritmos de um tipo ou outro. Existiam características de improvisação diferentes para muitas delas, como bem as músicas eram diferentes em detalhes. Teorias matemáticas, sistemas de afinação experimentais, algoritmos de inteligência artificial, projetos de instrumentos de improvisação, e performance interativa eram algumas das áreas exploradas nestes trabalhos (...). Os solos tocavam simultaneamente no cenário de grupo, se tornando “sub”-composições que interagem, cada uma enviando e recebendo dados pertinentes para o funcionamento musical. (??, 12º parágrafo).*

1986:

- Bischoff e Perkis, após a saída de Horton
- Chris Brown e Mark Trayle
- *THE NETWORK MUSE - Automatic Music Band Festival.*
- Scott Greham-Lancaster e Richard Zvonar
- Phil Burk, Larry Polansky e Phil Stone.

## The Hub

- Nome simbólico de um sistema musical centralizado • (...) *um pequeno microcontrolador como caixa de correio, para postar dados usados no controle de seus sistemas individuais, que eram então acessados por outro intérprete, para usar de qualquer maneira e em qualquer tempo que escolher.*

## The Hub

*The Hub originalmente surgiu como uma maneira de limpar uma bagunça. (...) Toda vez que nós ensaiamos, um conjunto complicado de conexões ad-hoc entre computadores tinham de ser feitas. Isso criou um sistema com um comportamento rico e variado, mas sujeito a falhas, e trazer outros jogadores ficava difícil. Mais tarde, procuramos uma maneira de abrir o processo, para torná-lo mais fácil para os outros músicos tocarem no contexto de rede. O objetivo era criar uma nova maneira para pessoas fazerem música juntos. A solução bateu no ponto da facilidade de uso, e fornecimento de uma interface de usuário padrão, de modo que os jogadores poderiam conectar praticamente qualquer tipo de computador. The Hub é um pequeno computador dedicado a passar mensagens entre os jogadores. Ele serve como uma memória comum, mantendo informações sobre a atividade de cada jogador que seja acessível para os computadores de outros jogadores (??, seção 2.1).*

# The Hub

1987

- Nick Collins e Phil Niblock: curadoria para uma performance telemática entre a *Experimental Media* e *The Clocktower* em Nova York.
- Divididos em dois trios, se comunicam entre os dois espaços:
- John Bischoff, Tim Perkis, Mark Trayle;
- Chris Brown, Scot Gresham-Lancaster, e Phil Stone;
- Dois *Hub* intercomunicáveis.
- Cada *Hub* era um sistema centralizado para cada trio.
- “Simple Degradation”;
- “Borrowing and Stealing”;
- “Vague Notions”;
- performance de um sexteto acusticamente divorciado mas informacionalmente ligado (??, seção 2.2).

- 1985: performance de *Water Surfaces*, na STEIM (*STudio for Electro-Instrumental Music*), em Amsterdã ??);
- 2007: reconstrução da peça “*TOPLAP001 - A prehistory of live coding*” ([http://toplap.org/wiki/TOPLAP\\_CDs](http://toplap.org/wiki/TOPLAP_CDs).);

## Water Surfaces

*Esta obra usou programação FORTH ao vivo; Curtis ??) testemunhou e relatou a performance de Ron Kuivila feita na STEIM em Amsterdã, em 1985; a performance original termina com a quebra do sistema. . .*

## Water Surfaces

*Ronald Kuivila programou um computador Apple II no palco para criar sons densos, rodopiantes e métricos, disposto em camadas e dobravam sobre si. Considerando o equipamento usado, os sons eram surpreendentemente grandes em escala. Kuivila teve problemas em controlar a peça devido q problemas sistêmicos. Ele finalmente entrou em dificuldades técnicas e finalizou a performance (??, p. 47).*



## Water Surfaces

*Eu vi o software FORTH de Ron Kuivila quebrar e queimar no palco em Amsterdã em 1985, mas antes disso, não fez uma música muito interessante. A performance consistiu de digitação(??)*

## *Live Algorithm Programming and Temporary Organization for its Promotion*

- Manifesto elaborado por **????**;
- DJ, Música algorítmica, Música de Processos;
- Regras práticas do *livecoding* publicadas por artistas-programadores ingleses;

## LAPTOP

*O Livecoding permite a exploração de espaços algorítmicos abstratos como uma improvisação intelectual. Como uma atividade intelectual, pode ser colaborativa. Codificação e teorização podem ser atos sociais. Se existe um público, revelar, provocar e desafiar eles com uma matemática complexa se faz com a esperança de que sigam, ou até mesmo participem da expedição. Estas questões são, de certa forma, independentes do computador, quando a valorização e exploração do algoritmo é o que importa. Outro experimento mental pode ser encarado com um DJ ao vivo codificando e escrevendo uma lista de instruções para o seu set (feito com o iTunes, mas aparelhos reais funcionam igualmente bem). Eles passam ao HDJ [ Headphone Disk Jockey ] de acordo com este conjunto de instruções, mas no meio do caminho modificam a lista. A lista está em um retroprojeter para que o público possa acompanhar a tomada de decisão e tentar obter um melhor acesso ao processo de pensamento do compositor (??, p. 245).*

- Música como um Processo Gradual<sup>6</sup>
- A ligação conceitual do *live coding* com a Música de Processos e a Música Generativa é relativa ao uso de algoritmos mas não ao resultado sonoro como processo de escuta.
- Música Minimalista de Processos é apresentada por ??, p. 128) → Música de Algoritmos Simples, um processo determinístico que age sobre focos de quadros temporais.
- Música Generativa é sensível às circunstâncias, isso quer dizer que irá reagir diferentemente dependendo das suas condições iniciais, onde ocorre e assim por diante.

## LAPTOP

*Na codificação ao vivo a performance é o processo de desenvolvimento de software, em vez de seu resultado. O trabalho não é gerado por um programa acabado, mas através de sua jornada de desenvolvimento do nada para um algoritmo complexo, gerando mudanças contínuas da forma musical ou visual ao longo do caminho. Isto contrasta com a arte generativa popularizada pela música geradora de Brian ??). (...) O resultado segue mais ou menos o mesmo estilo, com apenas algumas permutações, dando uma idéia das qualidades da peça. Isto é bem ilustrado pelo nosso estudo de caso de um artista-programador, que executa seu programa poucas vezes não para produzir novas obras, mas para obter diferentes perspectivas sobre o mesmo trabalho.*

## TOPLAP I

Permutação na ordem das letras do acrônimo LAPTOP dá origem ao acrônimo TOPLAP. ??, p. 246) e ??) apontam que este acrônimo dinâmico; isto quer dizer que as primeira, terceira e quinta letras possuem diversos significados

(Temporary|Transnational|Terrestrial|Transdimensional)  
Organization  
for the  
(Promotion|Proliferation|Permanence|Purity)  
of  
Live  
(Algorithm|Audio|Art|Artistic)  
Programming

Figura : Definição do significado de TOPLAP. Fonte: ??).

## TOPLAP II

*A organização TOPLAP ([www.toplap.org](http://www.toplap.org)), cuja sigla possui diversas interpretações, uma sendo Organização Temporária para a Proliferação da Programação de Algoritmos Ao Vivo, foi criada para promover e explorar o live coding. TOPLAP nasceu em um bar enfumaçada em Hamburgo à uma da manhã em 15 de Fevereiro de 2004 (??, p. 246).*

## Manifesto Lubeck04

Escrito em um ônibus, é mais conhecido como “*Show us your screens*”, e prescreve algumas regras práticas do *live coding* (??, p. 247).



## Manifesto Lubeck04 I

Exigimos:

- Acesso à mente do intérprete, para todo o instrumento humano.
  - Obscurantismo é perigoso. Mostre-nos suas telas.
  - Programas são instrumentos que podem modificar eles mesmos.
  - O programa será transcendido - Língua Artificial é o caminho.
  - O código deve ser visto assim como ouvido, códigos subjacentes visualizados bem como seu resultado visual.
  - Codificação ao vivo não é sobre ferramentas. Algoritmos são pensamentos. Motosserras são ferramentas. É por isso que às vezes algoritmos são mais difíceis de perceber do que motosserras.
- Reconhecemos contínuos de interação e profundidade, mas preferimos:
- Introspecção dos algoritmos.
  - A externalização hábil de algoritmo como exibição expressiva/impressiva de destreza mental.
  - Sem *backup* (minidisc, DVD, safety net computer).

## Manifesto Lubeck04 II

Nós reconhecemos que:

- Não é necessário para uma audiência leiga compreender o código para apreciar, tal como não é necessário saber como tocar guitarra para apreciar uma performance de guitarra.
- Codificação ao vivo pode ser acompanhada por uma impressionante exibição de destreza manual e a glorificação da interface de digitação.
- Performance envolve contínuos de interação, cobrindo talvez o âmbito dos controles, no que diz respeito ao parâmetro espaço da obra de arte, ou conteúdo gestual, particularmente direcionado para o detalhe expressivo. Enquanto desvios na tradicional taxa de reflexos táteis da expressividade, na música instrumental, não são aproximadas no código, por que repetir o passado? Sem dúvida, a escrita de código e expressão do pensamento irá desenvolver suas próprias nuances e costumes.

## Manifesto Lubeck04

- “Obscurantismo é perigoso. Mostre-nos suas telas”
- “Algoritmos são pensamentos, motosserras são ferramentas”

## Mostre-nos suas telas

- “*Using Contemporary Technology in Live Performance; the Dilemma of the Performer*” (??)
- Crítica ao sétimo dos questionamentos sugeridos para uma performance de improvisação ao vivo com computadores ??.

## Mostre-nos suas telas I

1. Causa-e-efeito é importante, pelo menos para o observador/audiência em uma sala de concerto.
2. Corolário: Mágica na performance é bom. Muita mágica é fatal! (chato).
3. Um componente visual é essencial para a audiência, tal como existe um aparato visual de entrada para parâmetros e gestos.
4. Sutileza é importante. Grandes gestos são facilmente visíveis de longe, o que é bom, mas eles são movimentos de desenho animado se comparados à execução de um instrumento musical.
5. Esforço é importante. Neste sentido, nós estamos em desvantagem de desempenho na performance musical com o computador.
6. Improvisação no palco é bom, mas “mimar” o aparato no palco não é improvisação, é edição. É provavelmente mais apropriado fazer isso no estúdio antes do concerto, ou se durante o concerto, com o console no meio ou atrás da sala de concerto.

## Mostre-nos suas telas II

7. Pessoas que representam devem representar. Um concerto de música de computador não é uma desculpa/oportunidade para um programador(a) se sentar no palco. Sua presença melhora ou impede o desempenho da representação?

# Algorithms are Thoughts, Chainsaws are Tools

Vídeo de Stephen ??), publicado no Vimeo, em 27 de fevereiro de 2010, como um *Coffee-Table Movie*. É uma análise pessoal da performance de *Strange Places* de Andrew Sorensen.

# Algorithms are Thoughts, Chainsaws are Tools

Algoritmo como pensamento é um espaço conceitual abstrato



## Algorithms are Thoughts, Chainsaws are Tools

*A noção de partitura não se aplica aqui, é como não fosse possível aplicá-lo ao músico de jazz ou tocador de bluegrass. (...). Levanta a questão, para mim, se, em uma sessão de livecoding \*feita\*, consiste simplismente no ato de digitar em um programa existente, seria tão convincente – eu acho que isso pode definitivamente ter pontos de interesse. Ou qual seria o análogo do livecoding para uma performance não-improvisada de música?*

## Algorithms are Thoughts, Chainsaws are Tools

*O que torna o livecoding diferente, e pode a performance de música tradicional imitar isso? Para responder esta questão, parece importante notar que as formas nas quais a música improvisada muitas vezes apela para alguma noção de autenticidade ou gênio. Enquanto o livecoding ele mesmo à noção de virtuosismo de código, “autenticidade” parece fora de lugar aqui. Se música improvisada sugere expressão, o livecoding sugere um conjunto de restrições na expressão, descrevendo os parâmetros através dos quais a máquina [midi] ganha expressão*

# Metodologia de análise de uma Improvisação musical de códigos

- Taxonomia de tipos de criatividade → o *Quadro Conceitual de Sistemas Criativos* (*Creative System Frameworks*, ou CSF);
- Discretização (amostragem) de processos criativos contínuos;
- Modelo de estruturação, e análise, de improvisações de códigos musicais. capítulo.

# Criatividade

- Criatividade (??);
- Criatividade, códigos e imagens mentais (??)
- Comportamento Criativo por Bricolagem;
- Quadro Conceitual de sistemas criativos
- O modelo de improvisação

# Paradoxo da Criatividade

- Paradoxo da Criatividade: “Capacidade que o falante de uma língua tem de criar novos enunciados sem que os tenha ouvido ou dito anteriormente.”  
(<http://www.priberam.pt/dlpo/criatividade>);

## Paradoxo Criatividade

*O ponto inicial de Boden para o desenvolvimento de sua explicação, é a observação de que o conceito de criatividade contém um paradoxo. Por definição, criatividade cria, i.e., produz alguma coisa nova. Mas se estamos comprometidos com uma abordagem mecanicista do mundo – nenhum milagre é permitido – iremos acreditar que tudo o que ocorre é, em princípio, previsível. Iremos acreditar também que qualquer coisa nova deve ser construída de componentes existentes. Isso implica que nada pode ser intrinsecamente novo.(??, p. 2)*

## Paradoxo da Criatividade

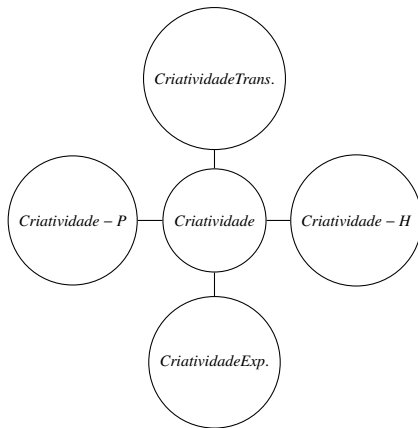
Valorização de distinções → *Para justificar a classificação de uma idéia criativa... alguém deve identificar os princípios generativos com respeito ao que é impossível* (??, ?? apud ??, ??, p. 51; p. 3)

# Taxonomia da Criatividade

*A distinção é entre o sentido de criar um conceito que nunca foi criado antes [criatividade-P], e um conceito que nunca foi criado antes por um criador específico [criatividade-H]. (...) no trabalho de Boden, existe uma distinção entre criatividade exploradora e transformacional, que será relevante aqui, e então merece alguma explicação. **Boden concebe o processo de criatividade como uma identificação e/ou localização de novos objetos conceituais em um espaço conceitual.***



# Taxonomia da Criatividade



**Figura :** Classificação da criatividade : 1) criatividade-psicológica/criatividade-histórica; 2) criatividade exploradora/criatividade transformacional. **Fonte:** autor com base em ??).

## Taxonomia da Criatividade

*De fato, na primeira edição [1990], ela não oferece uma explicação do número de diferentes tipos de criatividade que ela identificou. Parece que sua intenção era distinguir os dois tipos notados, espaços conceituais devem ter uma característica generativa. E isso certamente é a interpretação comum. Ainda no sumário 'em uma casca de noz' de sua teoria, foi adicionado um prólogo à segunda edição (??), e em ??), ela coloca que sua explicação distingue três principais formas de criatividade, sendo exploração, transformação e combinação. É somada à incerteza a observação que somente a definição forte [generalizadora] da definição possui o poder de resolver o paradoxo da criatividade*

# Taxonomia da Criatividade

- *comportamento combinatorial*, imagens poéticas, analogias – onde duas idéias novas associadas compartilham heranças de estruturas conceituais –, piadas (??, p. 348)
- *comportamento explorador* de algo → de espaços conceituais estruturados;
- *comportamento transformacional* → a exploração resulta em estruturas ("idéias") que não são novidades, mas inesperadas. Considerado por Boden como a “criatividade genuína”.

# Taxonomia da Criatividade

- Wiggins → classificação exploradora/transformacional ao enunciar quatro axiomas dos Sistemas Criativos ??, p. 3–4)
- Utilização de heurísticas e mapas para identificar conceitos valiosos dentre de um espaço conceitual existente (exploração guiada).

# Definição de Criatividade

Axiomas abrem espaço para uma definição generalizadora:

**Tabela :** Definições formais de criatividade por ??, p. 451)

Criatividade	"O estudo e suporte, através de meios e métodos computacionais, do comportamento exibido por sistemas naturais e artificiais, que podem ser considerados criativos se exibidos em humanos."
Computação criativa	"O estudo e suporte, através de meios e métodos computacionais, do comportamento exibido por sistemas naturais e artificiais, que são considerados criativos".
Sistemas criativos	"Uma coleção de processos, naturais ou automáticos, que são capazes de alcançarem ou simularem comportamentos que em humanos seriam considerados criativos"
Comportamento Criativo	"Um ou mais dos comportamentos exibidos por um sistema criativo"

# Criatividade, códigos e imagens mentais

- Comportamento criativo como o processo de estruturação uma *imagem mental*.
- Imagem mental é um símbolo, e seu significado é governado por regras gramaticais e sociais (códigos).
- Hierarquia de códigos simbólicos, visuais e gramaticais – Teoria da Codificação Dual (??, ?? apud ??, ??, p. 25–29):

## Teoria da Codificação Dual

*Seu [Paivio] argumento não é que existem dois códigos, mas sim que existe uma hierarquia de códigos, que se ramificam no topo em códigos lingüísticos discretos e códigos de percepção contínua, que Paivio nomeia como logogens e imagens respectivamente. Esta divisão é apresentada em seus processos simultâneos; os seres humanos são capazes de compreender a linguagem ao mesmo tempo que podem ver imagens. (...), é fácil para seres humanos ler um texto enquanto simultaneamente e ouve um discurso. **A explicação oferecida pela teoria da Codificação Dual é que existem sistemas de símbolos distintos, mas integrados, para linguagens e figuras.***

## Teoria da Codificação Dual

Para McLean, a teoria da Codificação Dual é suportada pela Neuropsicologia, onde *De maneira muito ampla, o hemisfério esquerdo é especializado para linguagem, e o direito para tarefas visuais-espaciais.*(??, ?? apud ??, ??, p. 128–129), o que por outro lado, é admitido como uma perigosa generalização. Para exemplificar a formação de imagens (conceitos, lembranças, sentidos) durante a leitura e escrita de texto, apresentaremos um pequeno exemplo de código que formaliza, em uma linguagem de programação, um simples algoritmo musical.



# Teoria da Codificação Dual e Dimensões cognitivas da Notação para linguagens de programação.

- Tabela auxiliar, usada para fins de criação de linguagens, para a verificação de dimensões cognitivas acessadas durante a atividade de programação.
- As dimensões cognitivas das linguagens visuais são usadas para separar os materiais (o que é representado como material sonoro), da maneira como serão manipulados (o como é representado o material sonoro), durante um fluxo temporal.

# Dimensões cognitivas da Notação para linguagens de programação. I

**Tabela :** Dimensões cognitivas da Notação para linguagens de programação. **Fonte:** (??, ?? apud ??, ??).

Dimensão	Significado
Abstração	Disponibilidade de mecanismos de abstração
Dependências escondidas	Invisibilidade de ligações importantes entre entidades.
Compromisso prematuro	Restrição na ordem de execução das coisas.
Notação secundária	Notação diversa da sintaxe formal.
Viscosidade	Resistência à mudança.
Proximidade de mapeamento	Proximidade de representação para o domínio-alvo.
Consistência	Semânticas similares são expressadas em formas sintáticas similares.

# Dimensões cognitivas da Notação para linguagens de programação. II

<b>Dispersividade</b>	Prolixidade da linguagem.
<b>Tendência ao erro</b>	Probabilidade de erros.
<b>Operações mentais difíceis</b>	Demanda de recursos cognitivos.
<b>Provisoriedade</b>	Grau de compromisso com ações e marcos.
<b>Função de expressividade</b>	medida em que o efeito de um componente pode ser inferida.

## Comportamento criativo por bricolagem

No começo o programador tem um conceito meio-formado que só atinge consistência interna através do processo de ser expresso como um algoritmo. O laço interno é onde o programador elabora o objetivo de suas imaginações, e o laço externo é onde essa trajetória está fundamentada na pragmática do que elas realmente têm que fazer. Através deste processo ambos algoritmos e conceitos são desenvolvidos até que o programador sinta que um se aplica com o outro, ou de outra forma julga o processo criativo finalizado.

## Comportamento criativo por bricolagem

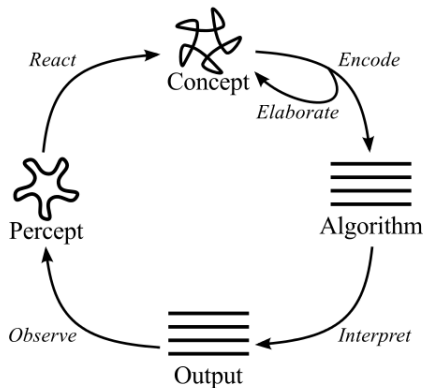


Figure 6.2: The process of action and reaction in bricolage programming

**Figura :** Modelo de bricolagem para o processo criativo realizado por um artista-programador.

**Fonte:** ??, p. 122).

# Comportamento criativo por bricolagem

Exemplo de um processo durante a elaboração de uma *estratégia transversal*

## Comportamento criativo por bricolagem

- Imagem mental: um *loop* sincopado, mas bastante regular, descrito em um compasso. Em uma “partitura-mental”, estruturamos o primeiro tempo com um baixo, que volta a tocar na segunda semicolcheia do terceiro tempo. No Segundo tempo, silêncio. No quarto tempo uma caixa aberta.

## Comportamento criativo por bricolagem

Snare drum

bass3

The image shows a musical score for two instruments: Snare drum and bass3. The Snare drum part is written on a five-line staff with a common time signature (C). It features a sequence of notes: a half note on the first line, a dotted quarter note on the second line, a quarter note on the third line, a dotted quarter note on the fourth line, and a quarter note on the fifth line. The bass3 part is written on a five-line staff with a common time signature (C). It features a sequence of notes: a half note on the first line, a dotted quarter note on the second line, a quarter note on the third line, a dotted quarter note on the fourth line, and a quarter note on the fifth line. The score is divided into two measures by a vertical line, and each measure ends with a double bar line and repeat dots. The entire score is enclosed in a rectangular box.



## Comportamento criativo por bricolagem I

```
-- Eletronic Dance Music, BPM = 120  
-- tempo 1 - baixo (bass)  
-- tempo 2 - silencio (silence)  
-- tempo 3 - silencio + baixo  
-- tempo 4 - caixa (sn e sn:4)  
d1 \$ (sound "bass3 silence [silence bass3] sn:4")
```

## Comportamento criativo por bricolagem

House bass

Snare drum

bass3

The image shows a musical score for three instruments: House bass, Snare drum, and bass3. The score is written in common time (C) and consists of two measures. The House bass part is in the bass clef and features a dotted half note on the second line of the staff in both measures. The Snare drum part is in the treble clef and features a sequence of notes: a quarter rest, an eighth note, a quarter note, and an eighth note, followed by a quarter rest, an eighth note, a quarter note, and an eighth note. The bass3 part is in the bass clef and features a dotted half note on the second line of the staff in both measures. The score is written on three staves, with the House bass staff at the top, the Snare drum staff in the middle, and the bass3 staff at the bottom. The staves are connected by a brace on the left and a bar line on the right.

## Comportamento criativo por bricolagem

```
-- Eletronic Dance Music, BPM = 120  
-- tempo 1 - baixo (bass)  
-- tempo 2 - silencio (silence)  
-- tempo 3 - silencio + baixo  
-- tempo 4 - caixa (sn e sn:4)  
d1 \$ (sound "bass3 silence [silence bass3] sn:4")
```

## Comportamento criativo por bricolagem

House bass

Snare drum

bass3

The image displays a musical score for three instruments: House bass, Snare drum, and bass3. The score is written in 4/4 time, indicated by the 'C' time signature. The House bass part uses a bass clef and features a sequence of notes and rests across four measures. The Snare drum part uses a drum clef and shows a rhythmic pattern of snare hits and rests. The bass3 part uses a bass clef and consists of a simple line of notes and rests. The score is organized into four measures, with a double bar line at the end of the fourth measure.

## Comportamento criativo por bricolagem

```
-- Eletronic Dance Music, BPM = 120  
-- com uma caixa seca no segundo tempo  
-- e uma caixa aberta no quarto tempo  
-- A cada 4 tempos, o ritmo diminui pela metade  
-- e depois volta ao normal.  
d1 \$ every 4 (density 0.5) (sound "bass3 sn [silence house] sn:4")
```

## Comportamento criativo por bricolagem

- Para ??, p. 130), programar “no momento”, a partir de um arquivo de texto em branco, com uma imagem mental do resultado sonoro (ou visual), caracteriza o processo criativo por bricolagem

## Quadro Conceitual de sistemas criativos

O universo,  $\mathcal{U}$ , é um espaço multidimensional, no qual dimensões são capazes de representar qualquer coisa, e todos os possíveis conceitos distintos correspondentes àqueles pontos em  $\mathcal{U}$  (...). Para tornar a proposta um espaço-tipo possível, permitirei que  $\mathcal{U}$  contenha todos os conceitos abstratos, bem como os concretos, e que é possível representar os artefatos tanto completos e incompletos (??, p. 451)

## Quadro Conceitual de sistemas criativos

Espaços conceituais  $C$ , finitos ou infinitos são definidos como restrições de um universo  $\mathcal{U}$ , caracterizando um conjunto não-determinístico de conhecimentos, representações, e conceitos



## Quadro Conceitual de sistemas criativos: axiomas

??, p. 452)

- $\mathcal{U}$ : um espaço multidimensional, cujas dimensões são capazes de representar qualquer coisa, e todos possíveis conceitos distintos correspondentes com distintos pontos em  $\mathcal{U}$ .

## Quadro Conceitual de sistemas criativos: axiomas

- Universalidade:  $\mathcal{U}$  pode conter tanto conceitos bem definidos (completos), parcialmente definidos (incompletos), e o mais incompleto dos conceitos. Os dois primeiros são representados pela letra  $c$ , enquanto o último é representado por  $T$ .
- Não-identidade dos conceitos: dois conceitos em  $\mathcal{U}$  são mutuamente diferentes entre si ( $c_1 \neq c_2$ ), e não são um Universo. Esta correção restringe a recursividade de conceitos, o que daria ênfase ao comportamento explorador e anularia a importância do comportamento transformacional.

## Quadro Conceitual de sistemas criativos: axiomas

- Inclusão Universal 1: *espaços conceituais*  $C$  são subconjuntos não-estritos do conjunto  $\mathcal{U}$ .
- Inclusão Universal 2: conceito vazio,  $T$ , em qualquer espaços conceitual  $C$ .

## Quadro Conceitual de sistemas criativos

$$\mathcal{U}_{livecoding} = [C_{Tecelagem}, C_{Audiovisual}, C_{Dana}, C_{Msica}, \dots, T]$$

# Quadro Conceitual de sistemas criativos

**Tabela :** Definições formais do Universo de possibilidades de **??**), ou Universo de Conceitos por **??**).

Representação	Nome	Significado
$\mathcal{C}$	Conceito	Uma instância de um conceito, abstrato ou concreto (??).
$\mathcal{U}$	Universo de Conceitos	Superconjunto não restrito de conceitos. (??). "Um universo de todos conceitos possíveis" (??)
$\mathcal{L}$	Linguagem	Linguagem utilizada para expressar regras.
$\mathcal{A}$	Alfabeto	Alfabeto da linguagem que contém caracteres apropriados para expressão das regras
$\mathcal{R}$	Regras de validação	Validam os conceitos em um universo, se apropriados ou não para o espaço trabalhado.

# Quadro Conceitual de sistemas criativos

**Tabela :** Definições formais do Universo de possibilidades de **??**), ou Universo de Conceitos por **??**).

$[[.]]$	Função de interpretação	"Uma função parcial de $\mathcal{L}$ para funções que resultam em números reais entre $[0, 1]$ (...) 0.5 [ou maior] significa uma verdade booleana e menos que 0.5 significa uma falsidade booleana; a necessidade disso para valores reais se tornará clara abaixo.
$[[\mathcal{R}]]$	Regras de validação	Uma função que interpreta $\mathcal{R}$ , resultando em uma função indicando aderência ao conceito em $\mathcal{R}$
$\mathcal{C} = [[\mathcal{R}]](\mathcal{U})$	Espaço Conceitual	Todos espaços conceituais são um subconjunto não-estrito de $\mathcal{U}$ . Um subconjunto contido em $\mathcal{U}$ . Uma função que interpreta $\mathcal{R}$ , resultando em uma função que indica aderência ao conceito em $\mathcal{R}$ .
$\mathcal{T}$	Regras de detecção	Regras definidas dentro de $\mathcal{L}$ para definir estratégias transversais para localizar conceitos dentro de $\mathcal{U}$ .

# Quadro Conceitual de sistemas criativos

**Tabela :** Definições formais do Universo de possibilidades de **??**), ou Universo de Conceitos por **??**).

$\mathcal{E}$	Regras de qualidade	Conjunto de regras que permitem-nos avaliar qualquer conceito que nós encontramos em $\mathcal{C}$ e determinar sua qualidade, de acordo com critérios que nós considerarmos apropriados. Regras definidas dentro de $\mathcal{L}$ para avaliar a qualidade ou a desejabilidade do conceito $c$ .
$\langle\langle\langle \mathcal{R}, \mathcal{T}, \mathcal{E} \rangle\rangle\rangle$	Função de interpretação	Uma regra necessária para definir o espaço conceitual, “independentemente da ordem, mas também, ficcionalmente, enumerá-los em uma ordem particular, sob o controle de $\mathcal{T}$ – isto é crucial para a simulação de um comportamento criativo de um $\mathcal{T}$ particular ( <b>??</b> ). Uma função que interpreta a estratégia transversal $\mathcal{T}$ , informada por $\mathcal{R}$ e $\mathcal{E}$ . Opera sobre um subconjunto ordenado de $\mathcal{U}$ (do qual tem acesso randômico) e resulta em outro subconjunto ordenado de $\mathcal{U}$ .

## O modelo de improvisação

*um esboço para uma teoria geral da improvisação integrada com preceitos da Psicologia Cognitiva (...) teoria do comportamento de improvisação na música (??, p. 2).*



## O modelo de improvisação

- Improvisação particionada em diferentes sequências pode ser parcialmente mapeada em categorias, como blocos sonoros, referentes conceituais e normas estilísticas, conjuntos de objetivos e processos.

# O modelo de improvisação

**Tabela :** Definições formais do Modelo de improvisação de Jeff ??), segundo ??, p. 2).

Representação	Significado
$E'$	Um bloco de eventos sonoros
$K'$	Uma seqüência de blocos de eventos E, onde um bloco de eventos não se sobrepõe com o seguinte
$I'$	Uma improvisação, particionada por interrupções em um número de K seqüências
$R'$	Um referente opcional, tal como uma partitura ou uma norma estilística

# O modelo de improvisação

**Tabela :** Definições formais do Modelo de improvisação de Jeff ??), segundo ??, p. 2).

$G'$	Um conjunto de objetivos
$M'$	Uma memória de longo prazo
$O'$	Um conjunto de objetos
$F'$	Um conjunto de características dos objetos
$P'$	Um conjunto de processos

## Estudo de caso

*Para fundamentar a discussão em música, considere uma peça de jazz, onde jazz é um conceito e uma composição particular é uma instância de um conceito. O musicista, explorando os limites do jazz, encontra então uma peça para além das regras usuais do jazz. Através deste processo, os limites do gênero musical podem ser redefinidos em algum grau, ou se a peça está em um novo terreno particularmente fértil, um novo sub-gênero de jazz emerge. Contudo uma peça de música que não quebra limites, de alguma forma pode ser considerada não-criativa (??, p. 117).*

## Estudo de caso

A *Study in Keith* é uma performance de programação ao vivo por Andrew Sorensen, inspirado nos concertos *Sun Bear* de Keith Jarret. Toda a música que você ouve é gerada a partir do código do programa que é escrito e manipulado em *tempo-real* durante a performance. O trabalho foi executado usando o ambiente de desenvolvimento [em linguagem] Scheme [chamado] Impromptu (<http://impromptu.moso.com.au>). Não é Keith, mas inspirado por Keith (??).

## Estudo de caso

A *Study In Keith* é um trabalho para piano solo (NI's Akoustik Piano), inspirado nos concertos *Sun Bear* de Keith Jarrett. **Note que não existe som para os dois primeiros 2 minutos da performance, enquanto estruturas iniciais são construídas.** Não é bem Keith, mas inspirado por Keith (??)

## Estudo de caso

Observação e análise de um comportamento criativo musical, de um improvisador-programador, que escreve uma programação-partitura. Mais especificamente, analisamos o contexto musical de uma simples sequência de blocos sonoros, gerada por uma função de interpretação , cujo referente opcional direto, são os Concertos *Sun Bear* de Keith Jarret.

## Concertos *Sun Bear*

Originalmente dez LPs de improvisações de Keith Jarrett no Japão produzidos pela *ECM Records* (1976 e 1978)



## Concertos *Sun Bear*

Terceiro dos concertos de improvisação que incluem o *Solo Concerts: Bremen/Lausanne* (1973) e *The Köln Concert* (1975).

## Concertos *Sun Bear*

Improvisação contínua, variando entre 31 a 43 minutos cada. Para cada dia, duas sessões de improvisação, em cidades diferentes.

- Kyoto, 5 de novembro (<<https://www.youtube.com/watch?v=T2TflQNxhjc>>);
- Osaka, 8 de novembro (<<https://www.youtube.com/watch?v=FC4iZ1wMoU8>>);
- Nagoya, 12 de novembro (<<https://www.youtube.com/watch?v=3a7ezm3D1jA>>);
- Tokyo, 14 de novembro (<<https://www.youtube.com/watch?v=ZH8VljjhPQ4>>);
- Sapporo, 18 de Novembro (<[https://www.youtube.com/watch?v=BqYBT\\_HoG4M](https://www.youtube.com/watch?v=BqYBT_HoG4M)>).

## Kyoto Parte 1

O tema de *Kyoto Parte 1* é repetido por Keith Jarrett no fim de *Kyoto Parte 2*. Então podemos considerar o todo deste concerto como uma grande Suíte (??, p. 129).

## Kyoto Parte 1

Revisto por Richard S. Ginnel <<http://www.mcana.org/formembersatlarge.html>>: [–]  
Este pacote gigantesco – um conjunto de dez LPs agora comprimidos em uma caixa robusta de seis [embalagens de] CDs – foi ridicularizado uma vez como uma última viagem de ego, provavelmente por muitos que não tomaram um tempo para ouvir tudo. (...) Ainda assim, o milagre é como esta caixa é consistentemente muito boa. **Na abertura de Kyoto, a meditação direcionada para o *gospel*** está em plena atuação, ao nível de suas melhores performances solo em Bremen e Koln, **e os concertos Osaka e Nagoya possuem citações de primeira linha, geralmente do tipo *folk***, mesmo profundas, idéias líricas.

## Kyoto Parte 1

Piano

The musical score is for a piano piece in 3/4 time. It consists of three measures. The first two measures are marked *pp* (pianissimo) and the third measure is marked *f* (forte). The melody is in the right hand, and the bass line is in the left hand. The key signature has one flat (B-flat). The first two measures are marked with a repeat sign, and the third measure is marked with a repeat sign and a first ending bracket.

Figura : Transcrição do motivo gerador do disco Kyoto, parte 1. **Fonte:** autor.

## Kyoto Parte 1

4

The musical score is written for piano in G-flat major (one flat) and 4/4 time. It consists of two systems, each with a treble and bass staff. The first system (measures 1-2) features a treble staff with a half-note melody (B-flat, A-flat, G-flat, F-flat) and a bass staff with a half-note accompaniment (B-flat, A-flat). The second system (measures 3-4) features a treble staff with a half-note melody (B-flat, A-flat, G-flat, F-flat) and a bass staff with a half-note accompaniment (B-flat, A-flat). The score includes dynamic markings: *pp* (pianissimo) at the beginning of the first system, *f* (forte) at the beginning of the second system, and *p* (piano) at the beginning of the third system. A repeat sign is present at the end of the first system and the beginning of the second system. A double bar line separates the first and second systems.

*pp* *f* *p*

## Kyoto Parte 1

8

*f* *p* *f* *p* *mp* *mp*

## Kyoto Parte 1

Uwe ??):

Na verdade, a abertura não é realmente improvisada - ela é baseada em uma música chamada "Song Of The Heart"

(<<https://www.youtube.com/watch?v=JgyRoQPDwM8>>). Eu tenho interesse especial em transcrever os primeiros 10, 11 minutos (que são simplesmente fenomenais). Por fim, eu usei a Reprise que Keith jogou no final da Parte II.



## Kyoto Parte 1

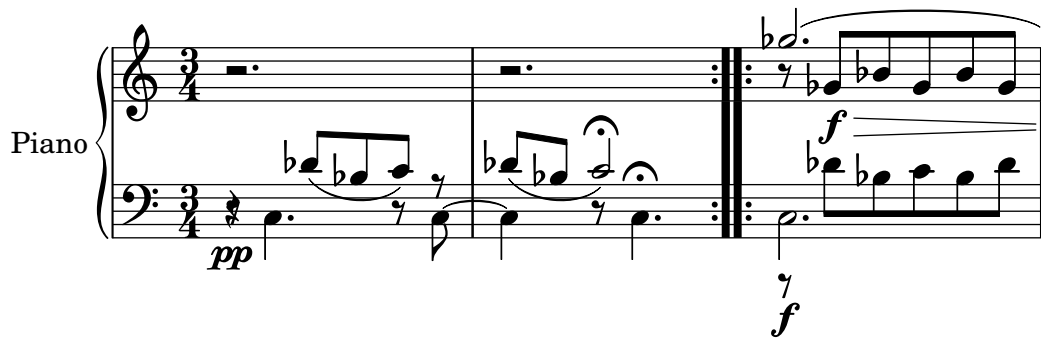


Figura : Transcrição do motivo gerador do disco Kyoto, parte 1. **Fonte:** autor.

## Kyoto Parte 1

4

pp

f

pp

## Kyoto Parte 1

8

The musical score for 'Kyoto Parte 1' consists of three measures. The first measure (measure 8) begins with a treble clef and a bass clef. The treble staff contains a half note G4, followed by eighth notes A4, B4, C5, and D5. The bass staff contains a half note G3, followed by eighth notes F3, E3, D3, and C3. The dynamic starts at *f* and crescendos to *pp*. The second measure (measure 9) continues the treble staff with a half note G4, followed by eighth notes A4, B4, C5, and D5. The bass staff contains a half note G3, followed by eighth notes F3, E3, D3, and C3. The dynamic starts at *f* and decrescendos to *pp*. The third measure (measure 10) continues the treble staff with a half note G4, followed by eighth notes A4, B4, C5, and D5. The bass staff contains a half note G3, followed by eighth notes F3, E3, D3, and C3. The dynamic starts at *pp*.

## Kyoto Parte 1

$$\begin{aligned} C : I^7 &\Rightarrow IV^7 \Rightarrow I^7 ; \\ &\Rightarrow I^7 \Rightarrow IV^7 \Rightarrow IV^7 ; \\ &\Rightarrow I^7 \Rightarrow I^7 \Rightarrow V^7 ; \\ &\Rightarrow IV^7 \Rightarrow I^7 \Rightarrow I^7 . \end{aligned}$$

## Kyoto Parte 1 – tempos 4 a 7

$$C : I^7 \Rightarrow IV^7 \Rightarrow IV^7 \Rightarrow I^7$$

## Kyoto Parte 1 – Song of the heart

$C : (bV^7/V)/IV \Rightarrow IV^7$ , ou  $C : bII^7/IV \Rightarrow IV^7$  (sequência do segundo grau bemol da subdominante para a subdominante do tom)

## Kyoto Parte 1 – Song of the heart

$$C : bV^7 \Rightarrow IV^7$$

## Kyoto Parte 1 – Song of the heart

$$C : bV \Rightarrow IV_4^6 \Rightarrow I^7.$$



## Kyoto Parte 1 – Song of the heart

O baixo pedal pode sugerir um 11 grau aumentado no primeiro acorde da sequência

$C : {}_{11\#add}bV)$

## Kyoto Parte 1 – Song of the heart

$$C : {}_{11}\#addbV \Rightarrow {}_5IV^{4-5} \Rightarrow I^{7-8}.$$

## Kyoto Parte 1 – Song of the heart

The image displays a musical score for a piano accompaniment, divided into four measures. The score is written on a grand staff with a treble and bass clef. The key signature is one flat (B-flat). The notes are as follows:

- Measure 1:** Treble clef has a whole note B-flat. Bass clef has a whole note B-flat, a quarter note B-flat, and a half note (B-flat) tied to the next measure.
- Measure 2:** Treble clef has a whole note B-flat, a quarter note B-flat, and a half note (B-flat) tied to the next measure. Bass clef has a whole note B-flat, a quarter note B-flat, and a half note (B-flat) tied to the next measure.
- Measure 3:** Treble clef has a whole note B-flat, a quarter note B-flat, and a half note (B-flat) tied to the next measure. Bass clef has a whole note B-flat, a quarter note B-flat, and a half note (B-flat) tied to the next measure.
- Measure 4:** Treble clef has a whole note B-flat, a quarter note B-flat, and a half note (B-flat) tied to the next measure. Bass clef has a whole note B-flat, a quarter note B-flat, and a half note (B-flat) tied to the next measure.

Harmonic analysis is provided in red text above and below the staff:

- Measure 1:** Treble:  $\hat{b}2$  (above B-flat),  $b5$  (below B-flat). Bass:  $\hat{1}$  (above B-flat),  $\hat{b}2$  (above B-flat),  $\hat{7}$  (above B-flat),  $\hat{1}$  (above B-flat).
- Measure 2:** Treble:  $\hat{7}$  (above B-flat),  $\hat{4}$  (above B-flat),  $\hat{6}$  (above B-flat). Bass:  $\hat{b}2$  (above B-flat),  $\hat{1}$  (above B-flat).
- Measure 3:** Treble:  $\hat{4}$  (above B-flat),  $\hat{6}$  (above B-flat),  $\hat{3}$  (above B-flat). Bass:  $\hat{b}7$  (above B-flat),  $\hat{1}$  (above B-flat).
- Measure 4:** Treble:  $\hat{5}$  (above B-flat). Bass:  $\hat{b}7$  (above B-flat),  $\hat{1}$  (above B-flat).

Chord symbols are written in red below the staff:

- Measure 1:** C:
- Measure 2:**  $bV/(add11+)$
- Measure 3:** IV64
- Measure 4:** I7

Colored arcs indicate voice leading or phrasing: a green arc in the bass of Measure 1, and blue and purple arcs in the treble and bass of Measures 2, 3, and 4.

# NI-Akoustik Piano

*Native Instruments*: empresa de tecnologias para áudio

## NI-Akoustik Piano

O *Akoustic Piano* é uma extensão (*plugin*) VST, que emula diferentes pianos acústicos. (<<http://www.native-instruments.com/en/products/komplete/keys/definitive-piano-collection/>>)

# NI-Akoustik Piano

$\mathcal{R}_{ask}^1$ : *A Study in Keith*  $\rightarrow$  *Disklavier Sessions* (??).

## NI-Akoustik Piano



**Figura :** Piano Disklavier de armário, com a parte interna exposta para exibir a placa-mãe.

**Fonte:** wikimedia.org

## NI-Akoustik Piano

Em *Disklavier Sessions* os programas escritos em tempo-real por Ben e Andrew geram um fluxo de dados de notas que é enviado para ser executado em um piano disklavier mecanizado. Assim como as alturas das notas, toda a performance do piano deve ser codificada na informação gerada pelo programa e enviada para o piano disklavier.



# NI-Akoustik Piano

Programação do piano no ambiente *Impromptu*.

# Impromptu

Linguagem e ambiente de programação OSX para compositores, artistas sonoros, VJ's e artistas gráficos com um interesse em programação ao vivo ou interativa. Impromptu é um ambiente de linguagem Scheme, um membro da família das languages Lisp.

## Impromptu

Considere a analogia da partitura musical tradicional. A partitura provê uma especificação estática da intenção – um programa de domínio estático. Musicistas, representam o domínio do processo, executam ações requeridas para realizar ou reificar a partitura. Finalmente, as ações no domínio do processo resultam em ondas sonoras que são percebidas por uma audiência humana como música. Este estágio final é o nosso domínio real de trabalho. Agora considere um domínio de programação dinâmica no qual o compositor concebe e descreve uma partitura em *tempo-real*. Nós geralmente chamamos este tipo de composição de improvisação. **Na improvisação o(a) musicista é envolvido em um circuito-fechado retroalimentado que envolve premeditação, movendo para ação casual e finalmente para reação, refinamento e reflexão (??, p. 823).**

# Scheme

Expressão vazia:

()

## Scheme I

Átomos, extraídos de sentenças abstratas, como listas de átomos:

$A \rightarrow ( A )$

$AB \rightarrow ( A B )$

$ABA \rightarrow ( A B A )$

$ABAC \rightarrow ( A B A C )$

# Scheme

Átomos também podem ser eles mesmos outras expressões:

```
;; A = A  
;; B = AB  
;; C = BAB
```

```
;; ABA ->  
( A ( A B ) A )  
;; ABAC ->  
( A ( A B ) A ( B A B ) )  
( A ( A B ) A (( A B ) A ( A B )))
```

```
;; A = 1  
;; B = +  
;; C = 2
```

```
;; ABA ->  
( 1 + 1 ) ;; = 2
```

```
;; ABAC ->  
( 1 + 1 2 ) ;; = 4
```

# Scheme

Notação chamada *notação prefixada*, inventada por Jan Łukasiewicz em 1924, para simplificar a lógica proposicional.

`;;A = 1`

`;;B = +`

`;;C = 2`

`;; ABA -> BAA`

`( + 1 1 ) ;; = 2`

`;;ABAC -> BAAC ->`

`( + 1 1 2 );; = 4`

# Scheme

```
;; define A = 1  
(define A 1)  
  
;; define B = 2  
(define B 2)  
  
;; divisao na forma (lambda argumentos operacao)  
(define divide ;; define nome da funcao  
  (lambda (a b) ;; argumentos da funcao (calcula lambda)  
    (/ a b)) ;; o que faz a funcao  
)  
  
;; execucao descritiva  
(divide A B)
```



## Exemplo musical para o Scheme I

Dois performers se apresentam no palco. Um violinista, em pé e parado, com seu arco preparado. Outro senta-se atrás do brilho da tela do *laptop*. Uma projeção da tela do *laptop* é projetada acima do palco, e mostra uma página em branco, com um simples cursor piscando. O musicista-programador começa a digitar

```
( play-sound ( now ) synth c3 soft minute)
```

## Exemplo musical para o Scheme

a expressão é avaliada, e lampeja no retroprojeto para exibir a ação do executante. O músico-programador, ouve o material temático fornecido pelo violinista e começa a delinear um processo generativo Markoviano para acompanhar o violino:

## Exemplo musical para o Scheme

```
( define chords
  ( lambda ( beat chord duration )
    ( for-each ( lambda ( pitch )
                  ( play synthj pitch soft duration ))
              chord )
    ( schedule (* metro * ( + beat duration )) chords
              (+ beat duration )
              ( random ( assoc chord (( Cmin7 Dmin7 )
                                       ( Dmin7 Cmin7 ))))
              duration )))

( chords (* metro * get-beat 4) Cmin7 4)
```

## Exemplo musical para o Scheme

A função *chords* cria um laço temporal, gerando uma sequência interminável de acordes de quatro tempos. Depois de poucos momentos de reflexão, o musicista-programador começa a modificar a função *chords* para suportar uma progressão de acordes mais variada, com uma razão aleatória [em função] da recursão temporal

## Exemplo musical para o Scheme

```
( define chords
  ( lambda ( beat chord duration )
    ( for-each ( lambda ( pitch )
                  ( play dls (+ 60 pitch) soft duration ))
              chord )
    ( schedule (* metro * ( + beat duration )) chords
              (+ beat duration )
              ( random ( assoc chord (( Cmin7 Dmin7 Bbmaj )
                                       ( Bbmaj Cmin7 )
                                       ( Dmin7 Cmin7 )))
                    ( random (3 6))))))

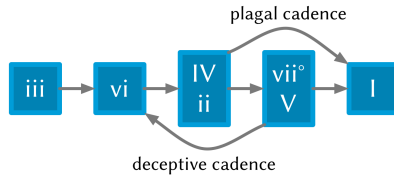
( chords (* metro * get-beat 4) Cmin7 4)
```

## Exemplo musical para o Scheme

Bloco de código: são utilizadas os seguintes movimentos harmônicos:

$I^{7+} \Rightarrow ii^7 \Rightarrow IV^7/IV$ , e  $IV^7/IV \Rightarrow I^7$  e  $ii^7 \Rightarrow I^7$ .

## Exemplo musical para o Scheme



	I	ii	iii	IV	V	vi	vii°
I	0.0	0.0	0.25	0.25	0.25	0.25	0.0
ii	0.0	0.0	0.0	0.0	0.5	0.0	0.5
iii	0.0	0.0	0.0	0.0	0.0	1.0	0.0
IV	0.25	0.25	0.0	0.0	0.25	0.0	0.25
V	0.5	0.0	0.0	0.0	0.0	0.5	0.0
vi	0.0	0.5	0.0	0.5	0.0	0.0	0.0
vii°	0.5	0.0	0.0	0.0	0.5	0.0	0.0

## Analise: Definição de instrumento

Primeiros eventos musicais gerados a partir das primeiras estruturas válidas de código. **Fonte:** (??).



## Analise: Definição de instrumento

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;  
;; Andrew Sorensen andrew@moso.com.au  
(define piano (au:make-node "aumu" "NaDd" "-NI-"))  
(au:connect-node piano 0 *au:output-node* 0)  
(au:update-graph)  
  
(au:load-preset piano "/tmp/convert_grand.aupreset")
```

## Analise: Definição de instrumento

Em 0' 52'' Sorensen define um tempo base. Em seguida, Sorensen apaga o código para então iniciar definições de notas (0' 54'').

## Definição de uma função auxiliar

```
(define *metro* (make-metro 110))
```

## Definição de uma sequência de blocos

Até 1' 07'', uma rotina auxiliar é definida como um laço iterativo. Porém não encontramos sua especificação no código-fonte do Extempore.

# Definição de uma sequência de blocos

```
(pc:cb-for-each-p chords piano  
  (pc:make-chord 50 70 2 (pc:diatonic 0 '- degree))  
  dur)
```

## Definição de uma sequência de blocos

Internamente, existe uma rotina que será o cerne de execução de uma nota, acompanhada de uma lista de 4 parâmetros (50, 70, 2):

## Definição de uma nota

```
(pc:make-chord 50 70 2 (pc:diatonic 0 '- degree))
```

## Definição de uma nota

Cria uma lista do “número” [com] alturas entre limites “menor” e “maior” do *pc* dado. Uma divisão dos limites, pelo número de elementos requisitados, decompõem a seleção em extensões diferentes, do qual cada altura é selecionada. *make-chord* tenta selecionar alturas para todos os graus do *pc*. É possível, para os elementos de um acorde resultarem em -1, se não existir nenhum *pc* para a extensão dada. [É] não-determinístico (i.e., resultados variam com o tempo). Argumento 1: limite menor (inclusivo). Argumento 2: Limite maior (exclusivo). Argumento 3: Número de alturas no acorde. Argumento 4: *pitch class* (??).



## Definição de blocos

Em 1' 08'', a função *chords* surge no fluxo audiovisual, sem nenhum processo de escrita. Este comportamento caracteriza a utilização de, ou uma cópia/cola de texto, ou de uma execução de um macro do editor de texto usado. Macros são pequenos programas no editor que auxiliam o processo de produção do código. De qualquer forma é importante salientar que o código é preparado (??).

## Algoritmo que define os acordes

```
(define chords
  (lambda (time)
    (for-each (lambda (p)
      (play-note (*metro* time) piano p 80 (*metro* 'dur dur)))
      (pc:make-chord 50 70 2 (pc:diatonic 0 (quote -) degree)))
    (callback (*metro* (+ time (* .5 dur))) chords (+ time dur))))

(chords (*metro* 'get-beat 4.0) 'i 3.0)
```

## Estratégia transversal

Este código inicial é então modificado, e finalizado em 1' 57'', momento em que é possível ouvir uma figura musical (uma classe de objeto  $O_{ask}^0$ ), duas díades, um intervalo de quarta justa entre Sol 2 (MIDI 55) e Dó 3 (MIDI 60). entre Mi bemol 2 (MIDI 51) e Dó 3 (MIDI 60).

## Estratégia transversal

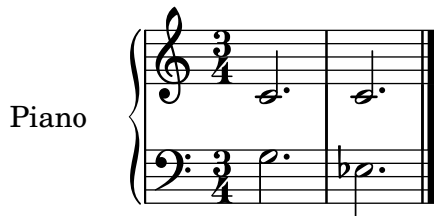
```
(define chords
  (lambda (time degree dur)
    (if (member degree '(i)) (set! dur 3.0))
    (for-each (lambda (p)
      (play-note (*metro* time) piano p
        (+ 50 (* 20 (cos (* pi time)))))
        (*metro* 'dur dur)))
      (pc:make-chord 50 70 2 (pc:diatonic 0 (quote -) degree)))
    (callback (*metro*) (+ time (* .5 dur))) chords (+ time dur)
      (random (assoc degree '((i vii)
        (vii i))))
      dur))

(chords (*metro* 'get-beat 4.0) 'i 3.0)
```

# Estratégia transversal

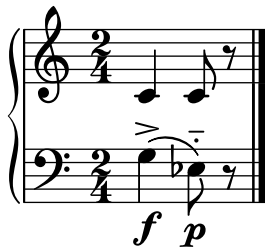
Duas transcrições: 110 BPM e 35–40 BPM

## Estratégia transversal



## Estratégia transversal

Piano

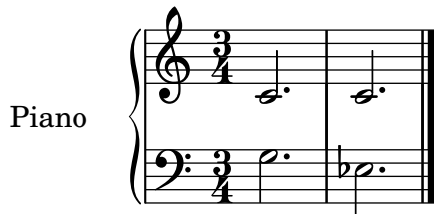


# Estratégia transversal

Neumas: • *bipunctum* • *clivis*



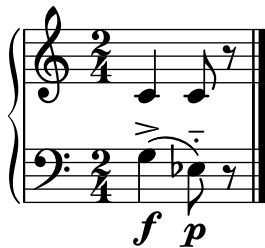
## Estratégia transversal



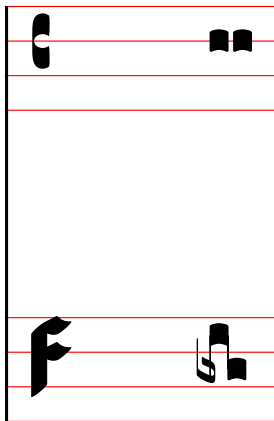
**Figura :** Transcrição literal e perceptiva do primeiro evento em *A Study in Keith*. **Fonte:** autor.

## Estratégia transversal

Piano



## Estratégia transversal



Um conjunto de características é definido pelo momento de execução do evento,  $\mathcal{F}'^0_{ask}$ , o grau,  $\mathcal{F}'^1_{ask}$ , e a duração deste evento,  $\mathcal{F}'^2_{ask}$ .

## Regra de qualidade $\mathcal{R}_{ask}$ .

```
( ... (... (callback (*metro*) (+ time (* .5 dur))) chords (+ time dur)
        (random (assoc degree '((i vii)
                                (vii i)))))
dur))
```

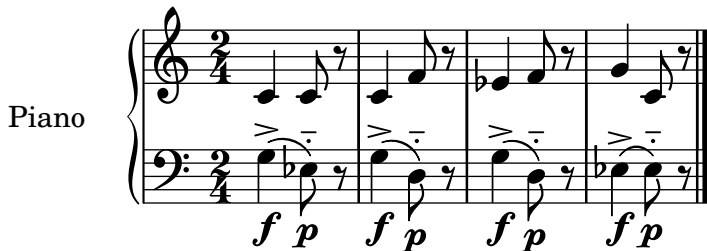
## Primeira sequência de blocos de eventos

O primeiro bloco de eventos  $\mathcal{E}_{ask}^{\prime 0}$  apresenta um contraponto de primeira espécie, aticulado em tempos fortes e fracos, de acordo com um movimento cadencial  $i \Rightarrow vii$ .

## Primeira sequência de blocos de eventos

Uma característica  $\mathcal{F}_{ask}^{'0}$  do algoritmo é sua direcionalidade em um âmbito de quinta em um número de compassos pares (4 nesse caso).

## Primeira sequência de blocos de eventos



**Figura :** Primeiros eventos musicais gerados a partir das primeiras estruturas válidas de código. **Fonte:** autor.



## Transcrição de neumas do primeiro bloco

Na mão direita:

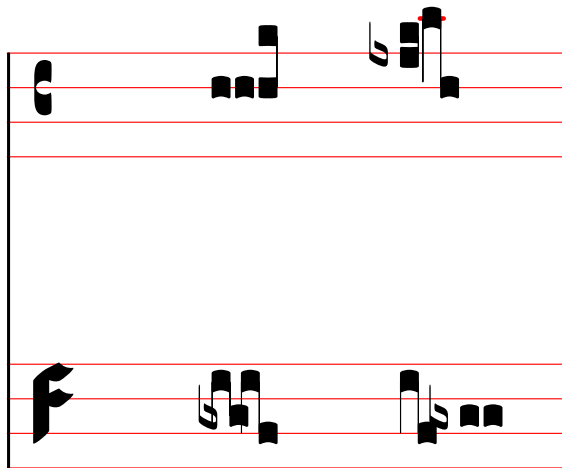
- um *bipunctum*;
- dois *podatus*;
- uma *clivis*.

## Transcrição de neumas do primeiro bloco

E na mão esquerda,

- três *clivis*
- um bipunctum.

## Transcrição de neumas do primeiro bloco



## Transcrição de neumas do primeiro bloco

Uma notação schenkeriana expõe um movimento plagal estrutural na mão direita ( $\hat{1}$ - $\hat{4}$ - $\hat{1}$ ), com um *porrectus* (um acento agudo, um grave e um agudo) estrutural na mão esquerda ( $\hat{5}$ - $\hat{2}$ - $\hat{3}$ ).

## Transcrição de neumas do primeiro bloco

The image displays a musical score for a Gregorian chant, consisting of a single system with two staves. The top staff is in treble clef, and the bottom staff is in bass clef. The melody is written on the top staff, starting with a half note on G4, followed by a half note on A4, then a half note on Bb4, and finally a half note on A4. A slur covers the last three notes. The bottom staff provides a harmonic accompaniment, starting with a half note on G3, followed by a half note on F3, and finally a half note on E3. A slur covers the last two notes. Above the top staff, there are three red neume symbols: a red '1' with a caret (^) above it, a red '4' with a caret (^) above it, and a red '1' with a caret (^) above it. Below the bottom staff, there are three red neume symbols: a red '5' with a caret (^) above it, a red '2' with a caret (^) above it, and a red '3' with a caret (^) above it. The entire score is enclosed in a black rectangular box.

## Segundo Bloco

Piano

The musical score is for a piano piece in 2/4 time. It consists of six measures. The melody in the treble clef is as follows: Measure 1: C4 quarter, D4 quarter; Measure 2: C4 quarter, D4 quarter; Measure 3: C4 quarter, D4 quarter; Measure 4: C4 quarter, D4 quarter; Measure 5: C4 quarter, B3 quarter; Measure 6: C4 quarter, D4 quarter. The bass line consists of eighth notes and quarter notes. The dynamics are marked as *f* (forte) and *p* (piano) alternating every measure. The piece ends with a double bar line.

Figura : Segundo bloco de eventos musicais. **Fonte:** autor.

Que pode ser reescrito como neumas na mão direita:

## Transcrição de neumas do segundo bloco

- cinco *podatus*;
- uma *clivis* na mão direita.

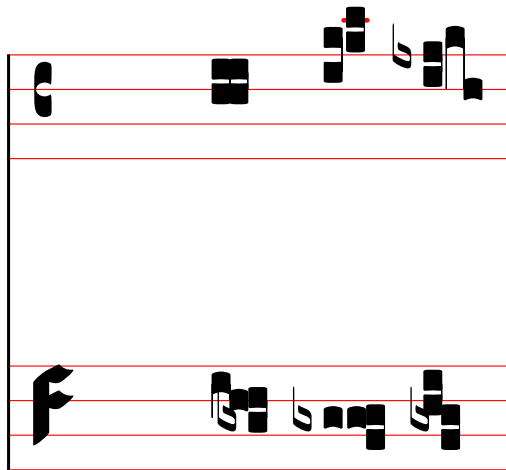


## Transcrição de neumas do segundo bloco

E na mão esquerda:

- uma *clivis*;
- um *podatus*;
- um *bipunctus*;
- três *podatus*.

## Transcrição de neumas do segundo bloco



## Transcrição de neumas do segundo bloco

The image displays a musical score for a three-measure phrase. The notation is written on a grand staff with a treble and bass clef. The key signature is C major, indicated by the absence of sharps or flats. The time signature is not explicitly shown but is implied to be common time (C). The notes are as follows:

- Measure 1: Treble clef has a half note C4 (middle C). Bass clef has a half note C3 (two ledger lines below). Above the treble staff is a red '1' with a caret (^). Below the bass staff is a red '5' with a caret (^).
- Measure 2: Treble clef has a half note E4, which is beamed to a quarter note D5. Bass clef has a half note E3. Above the treble staff is a red '5' with a caret (^). Below the bass staff is a red '2' with a caret (^).
- Measure 3: Treble clef has a half note C5. Bass clef has a half note G3. Above the treble staff is a red '1' with a caret (^). Below the bass staff is a red '3' with a caret (^).

Below the staff, the chords for each measure are labeled in red: **C:i64** for the first measure, **vii43** for the second measure, and **i6** for the third measure. A thick black line connects the bottom of the first and third measures, and another thick black line connects the bottom of the second and third measures, indicating the harmonic structure.

## Terceiro Bloco

$\mathcal{E}_{ask}^{'0}$  e  $\mathcal{E}_{ask}^{'1} \rightarrow$  Sorensen realiza uma transformação na estratégia transversal

# Primeira transformação da estratégia transversal I

```
(define chords
  (lambda (time degree dur)
    (if (member degree '(i)) (set! dur 3.0))
    (for-each (lambda (p)
      (let* (dur1 (* dur (random '(0.5 1))))
        (dur2 (- dur dur1)))
      (play-note (*metro* time) piano p
        (+ 50 (* 20 (cos (* pi time)))))
      (*metro* 'dur dur1))
      (if (> dur2 0)
        (play-note (*metro* (+ time dur1)) piano
          (pc:relative p (random '(-2 -1 1 2))
            (pc:scale 0 'aeolian))
          (+ 50 (* 20 (cos (* pi (+ time dur1))))))
          (*metro* 'dur dur2))))
      (pc:make-chord 50 70 2 (pc:diatonic 0 (quote -) degree)))
    (callback (*metro*) (+ time (* .5 dur)) chords (+ time dur)
      (random (assoc degree '((i vii)
        (v i)))))
```

## Primeira transformação da estratégia transversal II

```
(random (list 1 2 3))))
```

```
(chords (*metro* 'get-beat 4.0) 'i 3.0)
```

# Primeira transformação da estratégia transversal I

Transformação na acentuação

## Transcrição do terceiro bloco

Piano

5

Figura : Terceiro bloco de eventos musicais. **Fonte:** autor.



## Transcrição de neumas do terceiro bloco

Na mão direita:

- *bipunctus*;
- um *clivis*;
- um *porrectus*;
- um *clivis*;
- um *torculus*;
- um *clivis*;
- um *torculus*;
- dois *clivis*; e
- um *clivis subpunctum* .

## Transcrição de neumas do terceiro bloco

E na mão esquerda:

- um *bipunctus*;
- um *clivis*;
- um *podatus*;
- um *porrectus*;
- um *torculus*;
- um *podatus*;
- um *torculus*;
- um *climatus*;
- e um *clivis*.

## Transcrição de neumas do terceiro bloco



## Transcrição de neumas do terceiro bloco

A notação schenkeriana sugere um *torculus* estrutural na mão direita ( $\hat{1}-\hat{2}-\hat{1}$ ), onde existe um tetracorde decendente ( $\hat{5}-\hat{4}-\hat{3}-\hat{2}$ ). Na mão esquerda, destacamos o ponto mais grave da tessitura aponta para um modalismo ( $\hat{5}-\hat{7}b-\hat{3}$ ), bem como alterações nas estruturas rítmicas do trecho citado.

## Transcrição de neumas do terceiro bloco

## Laço iterativo modificado

```
(for-each (lambda (p)
  (let* (dur1 (* dur (random '(0.5 1))))
    (dur2 (- dur dur1)))
  (play-note (*metro* time) piano p
    (+ 50 (* 20 (cos (* pi time)))))
  (*metro* 'dur dur1))
(if (> dur2 0)
  (play-note (*metro* (+ time dur1)) piano
    (pc:relative p (random '(-2 -1 1 2))
      (pc:scale 0 'aeolian))
    (+ 50 (* 20 (cos (* pi (+ time dur1)))))
    (*metro* 'dur dur2)))
(pc:make-chord 50 70 2 (pc:diatonic 0 (quote -) degree)))
```

## Transcrição de neumas do terceiro bloco

Esta estratégia modifica o laço iterativo interno de cada altura da díade.

# Futuro?

$\mathcal{K}_{ask}^1 \rightarrow$  excede um objetivo deste documento.



## Conclusão

*“Wtf is livecoding”*

([<http://lurk.org/groups/livecode/messages/topic/ofAxZpxsKFpDRLnoA48Bh>](http://lurk.org/groups/livecode/messages/topic/ofAxZpxsKFpDRLnoA48Bh))

## Conclusão

*Live coding* celebra a efemeridade da própria definição

# Live coding

<INSERT DEFINITION HERE>

Figura : Definição de *live coding*: “Insira a definição aqui”. Fonte: ??).

## Conclusão

*A Study in Keith* de Andrew ??) ( (<<https://vimeo.com/2433947>>)

## Conclusão

Um caso particular que foge da maioria dos exemplos disponíveis.

## Conclusão

Envolve a manutenção de uma tradição musical tonal através de um interessante esforço de *replicação do estilo*.

## Conclusão

*A Study in Keith* de 2009, segundo um Quadro Conceitual de Sistemas Criativos.

# Referências I

1