

Fragments in Curved Air

Game generativo musical: estratégia generativa e interatividade

Eduardo Omine

Faculdade de Arquitetura e Urbanismo
Universidade de São Paulo
São Paulo, SP, Brasil
eduardo.omine@gmail.com

Daniela Kutschat Hanns

Faculdade de Arquitetura e Urbanismo
Universidade de São Paulo
São Paulo, SP, Brasil
dk.hanns@usp.br

Resumo— Este artigo apresenta o processo de criação do *game* musical generativo “Fragments in Curved Air” desenvolvido pelo autor e exibido na exposição de arte eletrônica FILE (Festival Internacional de Linguagem Eletrônica) realizada em São Paulo em 2011. Arte generativa é qualquer prática artística na qual o artista utiliza um sistema, que executado com algum grau de autonomia, contribui ou resulta em uma obra. O objetivo do *game* desenvolvido é investigar o potencial da estratégia generativa e da computação em trabalhos interativos, explorando a imprevisibilidade na produção de imagens e na composição musical. No *game*, o jogador navega em um espaço tridimensional, interferindo nos parâmetros visuais e musicais desse espaço. A trilha sonora é gerada em tempo real pelo *game*, seguindo um conjunto de regras que utiliza números aleatórios gerados por computador e *inputs* do jogador para selecionar notas de escalas musicais. Apesar da ausência de metas ou de um sistema de pontuação no *game*, se constatou seu potencial de imersão durante sua exibição.

Palavras-chave— Arte generativa, computação, interatividade, Processing

I. INTRODUÇÃO

Este artigo apresenta o processo de criação do *game* musical generativo “Fragments in Curved Air” desenvolvido pelo autor e exibido na exposição de arte eletrônica FILE (Festival Internacional de Arte Eletrônica) realizada em São Paulo em 2011.

Arte generativa, segundo Galanter (2003), é qualquer prática artística na qual o artista utiliza um sistema, que executado com algum grau de autonomia, contribui ou resulta em uma obra [1]. Ou seja, na arte generativa o artista não cria necessariamente um produto, mas sim um processo, implementado por meio de instruções escritas, programas de computador, máquinas, ou outros mecanismos. Esse processo possui autonomia em relação ao artista, e é seu resultado que consiste na obra propriamente dita. Para McCormack et al. (2012), processo deve ser considerado o “meio primário de expressão criativa” em arte generativa. [2].

Um exemplo de arte generativa é a peça musical “In C” (1964) do compositor e músico norte-americano Terry Riley [3]. Esta obra consiste em uma partitura com 53 frases musicais curtas, acompanhado de um texto com instruções para a performance. Ao contrário de uma peça musical na qual os

músicos leem a partitura linearmente, em “In C” os músicos intérpretes possuem autonomia para criar suas próprias combinações das frases musicais fornecidas pelo compositor. Desse modo, se obtém uma composição diferente a cada vez que a peça é realizada. A sonoridade obtida é complexa, resultante da sobreposição de frases musicais independentes. Esse princípio de composição musical apresentado por Riley foi uma inspiração para o desenvolvimento do *game* apresentado neste artigo.

Para Bense (1971), sistemas de estética generativa buscam uma “descrição numérica e operacional de características de estruturas estéticas”. Dessa forma, a estética generativa permite a “produção metódica de estados estéticos” [4]. O primeiro passo nessa produção estética é a descrição de uma informação estética em termos abstratos. A partir dessa abstração, se divide o processo de produção em etapas claramente formuladas e distintas. Ao se repetir as mesmas etapas com parâmetros diferentes, é possível obter diferentes instâncias concretas a partir das mesmas abstrações. Com a introdução de números aleatórios a esse processo, se torna possível obter resultados inesperados.

A possibilidade de observar resultados improváveis emergirem da interação entre simples regras é um dos fatores que motiva a utilização de computadores na prática de arte generativa. Burton destaca que “programas de computador são construídos a partir de operações lógicas determinísticas”, mas mesmo as mais simples operações podem ter consequências inesperadas quando executadas com *feedback* e interatividade [5].

Como acrescenta Tarbell, mesmo algoritmos simples podem resultar em formas complexas [6]. Em sistemas emergentes, não é possível determinar com antecedência o resultado dos processos em progresso. A possibilidade de algoritmos induzirem fenômenos com qualidades emergentes para gerar formas não-imaginadas é uma das características dos softwares que podem ser exploradas para a criação artística.

Por fim, o conceito filosófico de rizoma, proposto por Deleuze e Guattari [7], fornece fundamentos para uma prática artística dinâmica como a proposta pela arte generativa. Um rizoma se caracteriza por não apresentar uma hierarquia vertical, e sim uma rede de inter-relações horizontais entre seus pontos. Um rizoma é “sempre desmontável, conectável, reversível, modificável, com múltiplas entradas e saídas”. Ou

seja, um rizoma não apresenta uma forma definida e definitiva, ela é mutável em função das relações dinâmicas entre seus pontos. Da mesma forma, uma obra de arte generativa não apresenta uma forma final e definitiva, sua forma é resultado das relações dinâmicas entre as partes de um sistema proposto pelo artista.

II. OBJETIVO

O objetivo do *game* desenvolvido é investigar o potencial da estratégia generativa e da computação em trabalhos interativos. Por meio de um sistema computacional regido por algoritmos e alimentado por números aleatórios e *inputs* da audiência, se procurou explorar a imprevisibilidade na produção de imagens e na composição musical. Imagens e sons eram gerados programaticamente em tempo real, resultando em uma experiência audiovisual em constante mudança.

III. MÉTODO

Foi desenvolvido um *game* com a plataforma Processing [8], criada por Ben Fry e Casey Reas em 2001. Processing utiliza a linguagem de programação Java, possui código aberto, e é distribuído gratuitamente pela internet. Este *game* foi exibido em um festival de arte eletrônica em 2011, ao lado de outras obras de outros artistas. Durante o período da exposição, o autor observou, de maneira não-estruturada, a interação do público com o trabalho exposto.

IV. DESENVOLVIMENTO

O *game* foi executado em um computador, conectado a um *joystick* de Xbox 360, uma televisão de 50 polegadas, e um par de fones de ouvido (Figura 1).

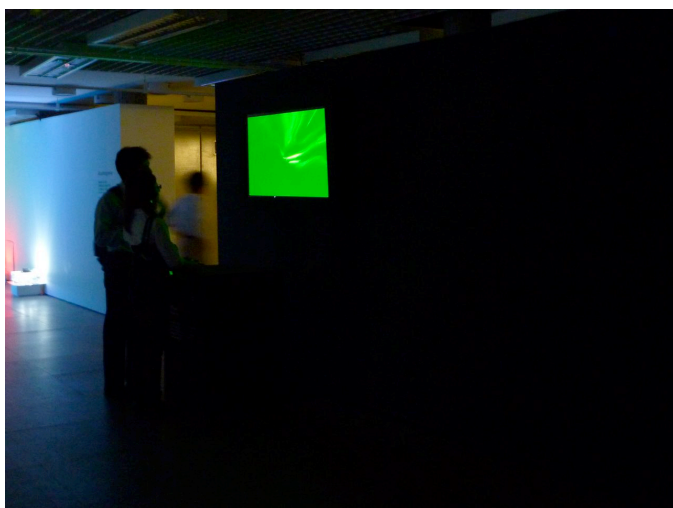


Fig. 1. Foto do game no espaço de exposição.

A. Som e imagem

O *game* se desdobra a partir de sua trilha sonora, gerada programaticamente em tempo real. A trilha é composta de 8 faixas, cada faixa correspondente a um instrumento (Tabela I). Seis faixas formam a base rítmica da trilha. As 2 faixas restantes são ativadas somente com a interação do jogador.

TABLE I. COMPOSIÇÃO DA TRILHA SONORA

Faixa	Instrumento	Grupo
1	Bumbo (bateria)	Base rítmica
2	Chimbau (bateria)	Base rítmica
3	Caixa (bateria)	Base rítmica
4	Baixo	Base rítmica
5	Sintetizador A	Base rítmica
6	Sintetizador B	Base rítmica
7	Arpejo A	Ativada com a interação
8	Arpejo B	Ativada com a interação

Cada faixa da base rítmica possui um padrão musical independente das outras faixas. Os padrões musicais consistem em sequências de até 16 notas, correspondentes à divisão do compasso musical em 16 partes iguais (semicolcheias). A cada 4 repetições do compasso, uma nota de cada padrão é selecionada e alterada aleatoriamente. As notas são escolhidas a partir de uma escala musical, que pode ser alterada pelo jogador.

Os elementos visuais do *game* consistem de formas geométricas tridimensionais primitivas, como cubos, cilindros, esferas e tetraedros (Figura 2).

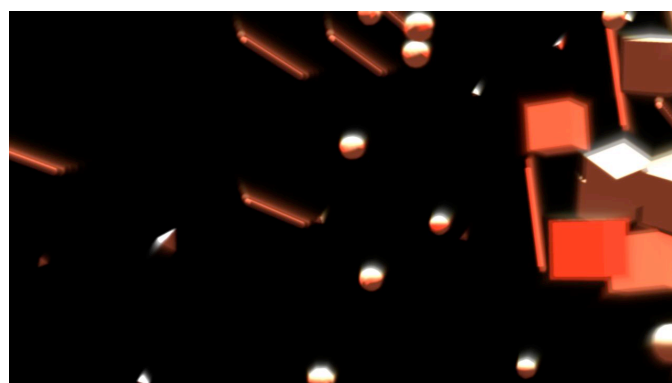


Fig. 2. As diferentes formas geométricas.

Cada grupo de formas pulsa de acordo com as notas musicais produzidas por uma determinada faixa musical (Tabela II). As formas geométricas se expandem ou encolhem conforme a intensidade do som produzido pela faixa sonora correspondente.

TABLE II. RELAÇÕES ENTRE INSTRUMENTOS E FORMAS

Faixa	Instrumento	Formas
1	Bumbo (bateria)	Cubos
2	Chimbau (bateria)	Tetraedros
3	Caixa (bateria)	Esferas
4	Baixo	—
5	Sintetizador A	Cilindros A
6	Sintetizador B	Cilindros B
7	Arpejo A	—
8	Arpejo B	—

B. Interações

As formas geométricas tridimensionais formam um espaço virtual por onde o jogador pode navegar, controlando uma câmera em primeira pessoa com o *joystick*. O controle da

câmera segue uma convenção estabelecida por outros jogos em primeira pessoa, na qual o controle direcional esquerdo do *joystick* manipula a posição da câmera, e o controle direcional direito manipula a direção da câmera.

O ato de navegar nesse espaço tridimensional utilizando os controles direcionais do *joystick* provoca uma distorção visual no cenário (Figura 3), além de ativar as 2 faixas sonoras interativas (arpejos A e B). A utilização do controle esquerdo ativa o arpejo A, enquanto o controle direito ativa o arpejo B.

Com o controle esquerdo, um movimento no eixo X provoca a repetição de uma mesma nota musical no arpejo A. Já um movimento para cima no eixo Y dispara uma sequência de notas ascendentes, e um movimento para baixo no eixo Y dispara uma sequência de notas descendentes. Com o controle direito, é o eixo X que dispara sequências ascendentes ou descendentes de notas no arpejo B, e o eixo Y que provoca a repetição de uma mesma nota musical.

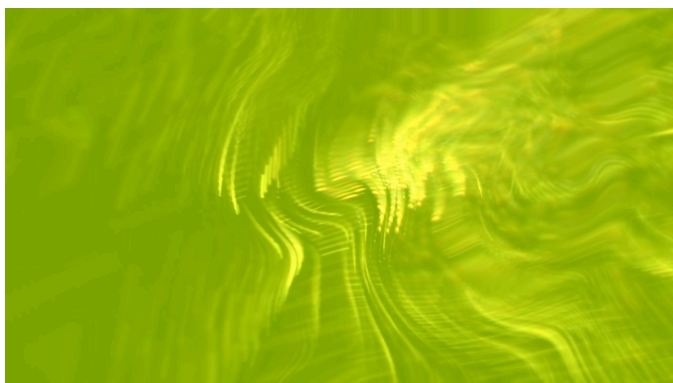


Fig. 3. Distorção visual provocada pela interação.

Pressionando os botões A, B, X, Y, LB, e RB do *joystick*, o jogador altera a escala musical utilizada pelo *game*, alternando entre 4 escalas disponíveis (escalas menor, pentatônica menor, pentatônica japonesa, e lídia).

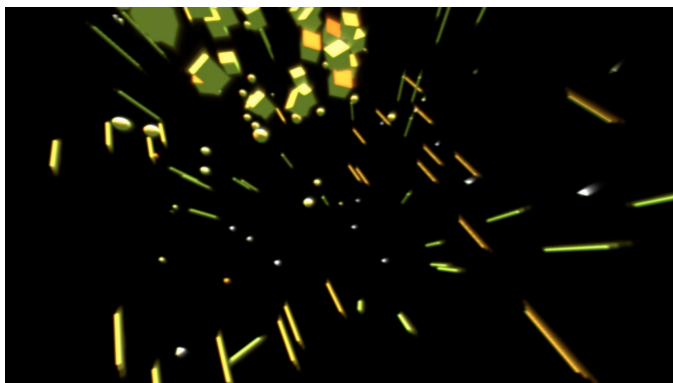


Fig. 4. Zoom ilustra aceleração do tempo musical.

Ao pressionar os botões LT e RT, o jogador altera simultaneamente o zoom da câmera e o tempo da música. Ao aproximar o zoom, fechando o campo de visão da câmera, o tempo musical é reduzido (a música se torna mais lenta), e ao afastar o zoom, ampliando o campo de visão, o tempo é acelerado (a música se torna mais rápida) (Figura 4).

C. Programação

Para habilitar a comunicação entre o *game* e o *joystick*, foi utilizada a biblioteca ProControll [9]. O controle de câmera em primeira pessoa foi implementado com a biblioteca OCD [10].

Na programação do áudio do *game* foram utilizadas as bibliotecas Minim [11] e Beads [12]. Enquanto a biblioteca Minim fornece ferramentas gerais de controle de áudio, a biblioteca Beads oferece classes voltadas especificamente para a composição musical por meio de programação.

Foram utilizadas classes utilitárias da biblioteca Toxiclibs [13] para manipulação de cores e de vetores espaciais (coordenadas x, y, e z).

O efeito de distorção visual é resultado da combinação dos seguintes filtros, implementados com *fragment shaders* em OpenGL: *blur*, *accumulation buffer*, e *displacement map*.

D. Gameplay

O *game* não apresenta objetivos específicos ou um sistema de pontuação. O formato de *game* foi utilizado para construir uma experiência interativa e artística com ênfase em seus aspectos visuais e musicais.

V. CONSIDERAÇÕES FINAIS

Por ter sido exibida em uma exposição coletiva, observou-se interferência de imagens e sons produzidos por obras vizinhas, o que pode ter tornado a interação com “Fragments in Curved Air” mais difícil ou menos atrativa para o público. A presença de elementos de isolamento físico, como paredes ou cortinas escuras, reduziria essas interferências e permitiria uma experiência interativa mais imersiva.

Dentro do público geral da exposição, observou-se que o público infantil foi o que interagiu de forma mais intensa, explorando as possibilidades dos controles de “Fragments in Curved Air”. Para ampliar o alcance do *game* para um público maior, recomenda-se em trabalhos futuros a adoção de elementos audiovisuais menos abstratos, e/ou mecânicas de *gameplay* mais tradicionais, com objetivos ou tarefas específicas. *Games* musicais como “Moondust” (1983) de Jaron Lanier [14], e “Rez” (2001) de Tetsuya Mizuguchi [15], são referências de trabalhos que equilibram abstração e figuração em suas linguagens visuais.

O autor espera explorar as relações entre arte generativa, computação, música, interatividade, e *games* com mais profundidade em trabalhos futuros.

REFERÊNCIAS

- [1] P. Galanter. “What is Generative Art? Complexity theory as a context for art theory.” In: GA2003–6th Generative Art Conference. 2003.
- [2] J. McCormack, O. Bown, A. Dorin, J. McCabe, G. Monro e M. Whitelaw, “Ten Questions Concerning Generative Computer Art,” Leonardo, San Francisco, 2012. No prelo.
- [3] T. Riley. “In C”. 1964. Peça musical
- [4] M. Bense. “The projects of generative aesthetics.” In: J. Reichardt. “Cybernetics, art and ideas.” Londres: Studio Vista, 1971.
- [5] E. Burton. “Sodaconstructor.” In: B. Fry e C. Reas. “Processing: a programming handbook for visual designers and artists.” Cambridge: MIT Press, 2007.

- [6] J. Tarbell. "Fractal.Invaders and Substrate." In: B. Fry e C. Reas. "Processing: a programming handbook for visual designers and artists." Cambridge: MIT Press, 2007.
- [7] G. Deleuze e F. Guattari. "Mil platôs. Volume 1." São Paulo: Editora 34, 1980.
- [8] Processing. Disponível em <<http://processing.org/>>. Acesso em 23 jul. 2013.
- [9] ProControll. Disponível em <<http://creativecomputing.cc/p5libs/procontroll/>>. Acesso em 23 jul. 2013.
- [10] OCD. Disponível em <<http://gdsstudios.com/processing/libraries/ocd/>>. Acesso em 23 jul. 2013.
- [11] Minim. Disponível em <<http://code.compartmental.net/tools/minim/>>. Acesso em 23 jul. 2013.
- [12] Beads. Disponível em <<http://www.beadsproject.net/>>. Acesso em 23 jul. 2013.
- [13] Toxiclibs. Disponível em <<http://toxiclibs.org/>>. Acesso em 23 jul. 2013.
- [14] J. Lanier. "Moondust." 1983. Videogame para a plataforma Commodore 64.
- [15] T. Mizuguchi. "Rez." 2001. Videogame para a plataforma Playstation 2.