

**UNIVERSIDADE FEDERAL DE MINAS GERAIS  
ESCOLA DE BELAS ARTES  
ARTES VISUAIS: HABILITAÇÃO EM CINEMA DE ANIMAÇÃO**

Airton Forato Alonso Filho

**MODELAGEM PROCEDURAL**

Belo Horizonte  
2013

Airton Forato Alonso Filho

## **MODELAGEM PROCEDURAL**

Trabalho de conclusão de curso apresentado como requisito parcial para aquisição do grau de Bacharéu em Artes Visuais, com Habilitação em Cinema de Animação.

Orientador: Prof. Virgílio Carlo de Menezes Vasconcelos

Belo Horizonte  
UFMG / Escola de Belas Artes  
2013

## Agradecimentos

Dedico a presente monografia ao meu filho, que me inspira a ver o novo de novo e de novo.

Agradeço a minha esposa e aos meus pais, pela paciência e por tanto terem me ensinado.

Agradeço ao professor Arttur Espindula por ter me introduzido ao 3D com tamanha paixão, bem como ao meu orientador Virgílio Vasconcelos por ter a mim propiciado o espaço e o suporte necessário para o desenvolvimento do presente trabalho.

*“máquina de pensar não funciona. O fato é secundário para nós. Tampouco funcionam os aparelhos de moto-contínuo cujos desenhos dão mistério às páginas das mais efusivas enciclopédias; tampouco funcionam as teorias metafísicas e teológicas que costumam declarar quem somos e o que é o mundo. Sua pública e famosa inutilidade não diminui seu interesse. Pode ser o caso (penso eu) da inútil máquina de pensar. [...] Como instrumento de indagação filosófica, a máquina de pensar é absurda. Mas não o seria como instrumento literário e poético.”*

(Jorge Luis Borges)

## Resumo

Este trabalho busca, através de uma breve introdução teórica e demonstração prática, apresentar ao leitor o conceito de modelagem procedural aplicada à técnica de animação 3D digital. A modelagem procedural é uma técnica que gera formas geométricas digitais, valendo-se da automatização de procedimentos em nome de um modelo final dotado de maior complexidade, gerado mais rapidamente, quando comparada às técnicas de modelagem poligonal vértice a vértice, com base em parâmetros ou algoritmos aplicados por computador a modelos primários criados pelo artista. Para tal, far-se-á uso do sistema de programação visual ICE (*Interactive Creative Environment*), recurso presente em softwares de modelagem para animação 3D digital produzidos pela empresa norte-americana Autodesk, como instrumento de demonstração, uma vez que este é dotado de interface gráfica amigável e intuitiva para artistas, e propícia à demonstração em mídia impressa, além de dispensar conhecimentos profundos nas áreas de programação, física, geometria, aritmética ou correlatas. A título de convenção, os modelos tridimensionais virtuais serão divididos em duas categorias intituladas: naturais (que representam aquilo que foi gerado sem intervenção direta do homem, também conhecidos pelo termo *organic modeling*) e trabalhados (aqueles criados por seres humanos, também conhecidos pelo termo *hard surfaces modeling*).

**Palavras-chave:**

Animação. Animação 3D Digital. Modelagem Procedural. Graphic Nodes. ICE.

## Abstract

This work intends to, through a brief theoretical introduction and a practical demonstration, introduce to the reader the concept of procedural modeling applied to the technique of digital 3D animation. Procedural modeling is a technique which generates digital geometrical shapes, using procedural automations seeking a more complex final model, more promptly generated based in parameters (patterns) or algorithms applied by computer to models made by the artist. With this goal, we'll use the visual programming system named ICE (Interactive Creative Environment) as a demonstrational tool, since it has a friendly graphical interface, favorable to demonstration in printed media, besides its exemption of deep knowledge about programming, physics, geometry, arithmetic or correlated fields. By convention, the final models will be divided in two distinct categories, which in Portuguese will be called: *naturais* (what represents everything generated without a direct human intervention, and in English is known as organic modeling) and *trabalhados* (the ones made by humans, in English known as hard surfaces).

**Keywords:**

Animation. Digital 3D Animation. Procedural Modeling. Graphic Nodes. ICE.

## Lista de ilustrações

FIGURA 1: Exemplo de modelagem procedural a partir da técnica <i>spin</i> . À direita, há meia seção transversal de um pão. À esquerda, suas extrusões ao redor de um eixo. Fonte: o autor. ....	11
FIGURA 2: Imagem resultante de processos de modelagem procedural. O modelo foi gerado com a técnica <i>spin</i> e suas cópias foram criadas com a técnica de <i>Geometry Instancing</i> .. Fonte: o autor. ....	12
FIGURA 3: Fotograma extraído de uma cena do filme Carros 2, em que é possível ver os personagens em primeiro plano, uma representação da <i>St. Paul's Cathedral</i> de Londres em segundo plano e prédios da cidade no plano de fundo. Fonte: Revista 3D World, Edição 145, p. 39. ....	14
FIGURA 4: Estrutura rizomática gerada proceduralmente. Fonte: o autor.....	16
FIGURA 5: Fotografia de uma espécie de couve-flor, em que é possível identificar padrões geométricos descritos pelo padrão Phi. Fonte: < <a href="http://tinylittleanthill.wordpress.com/tag/romanesco-cauliflower">http://tinylittleanthill.wordpress.com/tag/romanesco-cauliflower</a> > acesso em 30/01/2013..	18
FIGURA 6: Exemplos de estruturas semelhantes a árvores simpodiais de Aono e Kunni [2], geradas usando sistemas – L. Fonte: LINDENMAYER, Aristid; PRUSINKIEWICZ, Przemyslaw, p. 59. Tradução nossa.....	21
FIGURA 7: Exemplificação visual do sistema de quadros estruturais de Minsky. Fonte: < <a href="http://web.media.mit.edu/~minsky/papers/Frames/frames.html">http://web.media.mit.edu/~minsky/papers/Frames/frames.html</a> > acesso em 30/01/2013...	22
FIGURA 8: Duas possíveis representações da mesma cena, sendo à esquerda uma realizada por meio de <i>graphic nodes</i> e à direita outra realizada por representação virtual de modelos poligonais tridimensionais. Fonte: o autor.....	25
FIGURA 9: Demonstração de uma das possíveis maneiras de se gerar uma malha poligonal básica através dos menus do <i>software autodesk Softimage</i> . Fonte: o Autor. ....	26
FIGURA 10: Demonstração de uma das possíveis maneiras de gerar uma <i>point cloud</i> vazia através dos menus do <i>software autodesk Softimage</i> . Fonte: o Autor. ....	27
FIGURA 11: Demonstração de uma das possíveis maneiras de gerar uma ICE tree com simulação através dos menus do <i>software autodesk Softimage</i> . Fonte: o Autor. ....	27
FIGURA 12: Demonstração de uma das possíveis maneiras de acessar o <i>node Emit from geometry</i> usando a ferramenta de busca presente no editor <i>ICE</i> do <i>software autodesk Softimage</i> . Fonte: o Autor. ....	28

FIGURA 13: Demonstração de uma das possíveis maneiras de acessar o <i>node Get data</i> usando a ferramenta de busca presente no editor ICE do <i>software autodesk Softimage</i> .	
Fonte: o Autor. ....	28
FIGURA 14: Demonstração de uma das possíveis maneiras de acessar atrelar um dado ao <i>node Get data</i> usando o menu <i>Explorer</i> presente no editor ICE do <i>software autodesk Softimage</i> .	
Fonte: o Autor. ....	29
FIGURA 15: Demonstração de uma conexão bem sucedida entre <i>nodes</i> utilizando ICE no <i>software autodesk Softimage</i> .	
Fonte: o Autor. ....	30
FIGURA 16: Demonstração de uma emissão simples de partículas utilizando <i>graphic nodes</i> utilizando em ICE no <i>software autodesk Softimage</i> .	
Fonte: o Autor. ....	31
FIGURA 17: Demonstração de uma emissão simples de partículas utilizando representação virtual de polígonos tridimensionais.	
Fonte: o Autor. ....	32
FIGURA 18: Demonstração de uma emissão simples de partículas utilizando representação virtual de polígonos tridimensionais e geometria de instância.	
Fonte: o Autor. ....	33
FIGURA 19: Demonstração de uma emissão simples de partículas utilizando representação virtual de polígonos tridimensionais e geometria de instancia, além de partículas com tamanhos e orientações randomizadas.	
Fonte: o Autor. ....	34
FIGURA 20: Apresentação de um menu de configuração do <i>node Emit from geometry</i> .	
Fonte: o Autor. ....	35
Figura 21: Apresentação de um menu de configuração do <i>node Turbulize around Value</i> .	
Fonte: o Autor. ....	36
FIGURA 22: <i>Compound</i> criado pelo autor com finalidade de representar artisticamente raízes valendo se de algoritmo de colonização de espaço.	
Fonte: o Autor. ....	37
FIGURA 23: Detalhe de <i>FG experiment</i> .	
Fonte: < <a href="http://www.si-community.com/community/viewtopic.php?f=24&amp;t=447&amp;start=40">http://www.si-community.com/community/viewtopic.php?f=24&amp;t=447&amp;start=40</a> > acesso em 30/01/2013.	39
FIGURA 24: <i>Knit Strands</i> .	
Fonte:< <a href="http://www.matkovic.com/anto/works_knit_strand_09june2012.html">http://www.matkovic.com/anto/works_knit_strand_09june2012.html</a> > acesso em 30/01/2013.	40

## **Quadros**

Quadro 1 - Tipos de dados em graphic nodes do ICE.....	25
--	----

## Sumário

1	Introdução.....	11
2	Modelagem procedural e uso de padrões de repetição.....	17
2.1	Pi .....	17
2.2	Phi .....	18
2.3	Sistema de Lindenmayer .....	19
2.4	Sistema de quadros estruturais.....	20
3	<i>Interactive Creative Environment (ICE)</i> .....	22
3.1	Tipos de dados .....	23
3.2	Exemplo de uso do <i>ICE</i> , com representação por <i>graphic nodes</i> .....	24
3.3	Exemplos de compounds criados em <i>ICE</i> .....	37
3.3.1	Rizoma.....	37
3.3.2	<i>Kristinka Hair</i> .....	39
3.3.3	<i>Knit Strands</i> .....	40
3.3.4	Ruffle.....	41
	Considerações finais.....	42
	Glossário .....	43
	Referencias bibliograficas .....	46

## 1 Introdução

Vista de perto, a casa da chácara de Triste-le-Roy possuía muitas inúteis simetrias e repetições maníacas: a uma Diana glacial em nicho lóbrego correspondia em outro segundo nicho outra Diana; uma sacada refletia-se em outra sacada; duplas escalinatas abriam-se em dupla balaustrada. Um Hermes de duas caras projetava uma sombra monstruosa. Lönnrot rodeou a casa como rodeara a chácara<sup>1</sup>.

Na técnica de animação 3D digital, a modelagem procedural pode ser definida como aquela na qual o artista responsável pela criação dos modelos, ou modelador, se vale da automação de procedimentos com base em parâmetros ou algoritmos previamente definidos. É muito raro encontrar dentro do cinema de animação um filme criado com técnica de animação 3D digital, onde não haja a automação de procedimentos aliada à síntese de geometria, ainda que por vezes não nos demos conta disto.

Dentre as técnicas mais comuns pode-se citar a modelagem por simetria, ou espelhada, onde o modelador trabalha uma metade do modelo e outra é gerada por automação, e o *spin* (do inglês, 'girar'), em que meia seção transversal, por meio de procedimento automatizado, é extrudada repetidas vezes em torno de um eixo, como pode ser visto na FIG. 1:

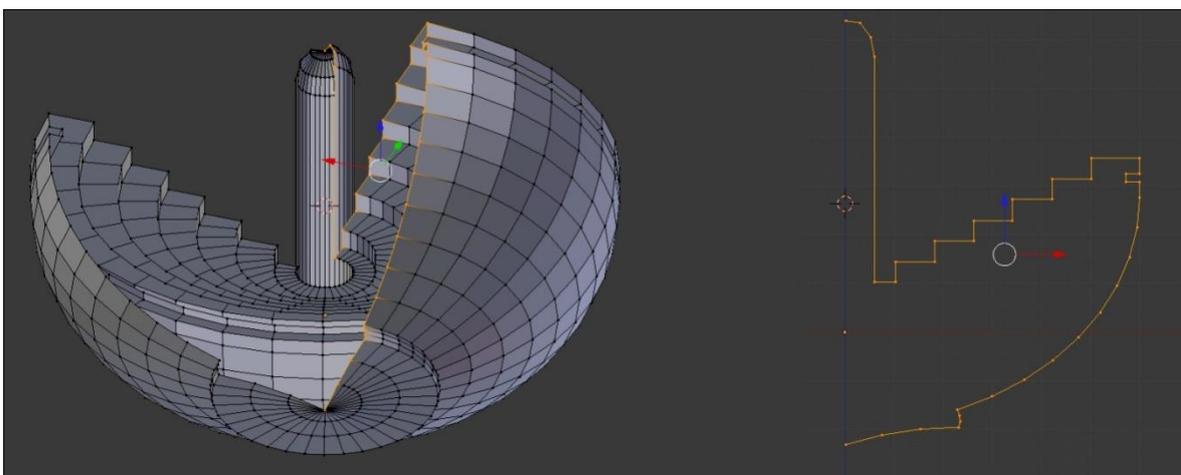


FIGURA 1: Exemplo de modelagem procedural a partir da técnica *spin*. À direita, há meia seção transversal um plano. À esquerda, suas extrusões ao redor de um eixo. Fonte: o autor.

<sup>1</sup>BORGES, Jorge Luis. *Obras completas*. São Paulo: Globo, 1998. V. 1, p. 563.

Na FIG. 2, apresenta –se o modelo resultante do processo ilustrado na FIG. 1.

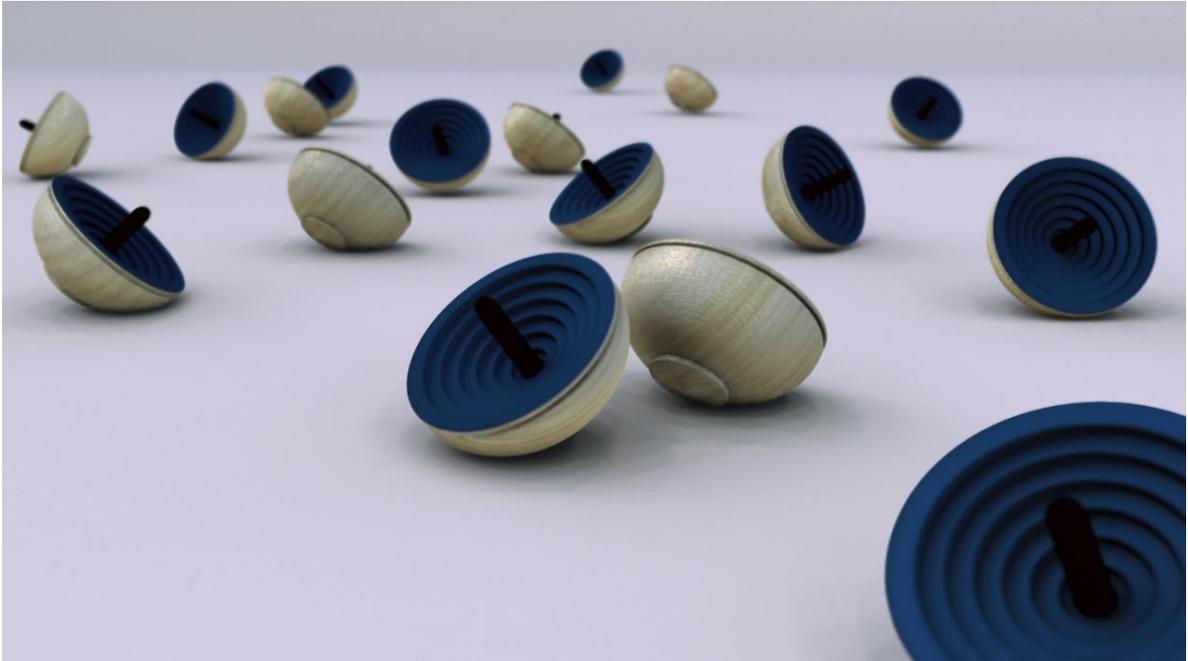


FIGURA 2: Imagem resultante de processos de modelagem procedural. O modelo foi gerado com a técnica *spin* e suas cópias foram criadas com a técnica de *Geometry Instancing*. Fonte: o autor.

A modelagem procedural possui diversas vantagens em relação à modelagem poligonal<sup>2</sup>. Dentre elas, podem-se listar: a capacidade de criar rapidamente modelos de formas complexas a partir de parâmetros simples e de gerar modelos diversos com base em padrões comuns, a fácil alteração dos modelos finais a partir da modificação de seus parâmetros iniciais, o fácil armazenamento de diversos modelos e o melhor aproveitamento tanto da CPU quanto da GPU.

Como afirma Pat Hanrahan:

[...]um conceito relacionado é a complexidade de Kolmogorev, ou o menor programa capaz de gerar uma dada função. Muito poucas linhas de código podem produzir lindas imagens. A cena de demonstração claramente ilustra essa ideia. Neste, a programadores são dadas limitações de que o tamanho dos arquivos contendo ambos códigos e dados (modelos, texturas, sons) devem caber em menos de 64KB. Deste arquivo emerge uma animação

<sup>2</sup>Por modelagem poligonal, entende-se aquela que figuras tridimensionais digitais são geradas por sucessivos procedimentos de extrusões, divisões e reposicionamento de polígonos, arestas ou vértices, sem que estes procedimentos sejam automatizados.

ricamente detalhada. (EBERT; MUSGRAVE; PEACHEY; PERLIN; WORLEY, 2003, p. v, tradução nossa.)

Há, no entanto de se salientar um viés do emprego de técnicas de modelagem procedural no que tange modelos orgânicos: a perda de controle do modelador quando comparadas às técnicas de modelagem poligonal. Deve-se, portanto, avaliar com atenção onde elas podem ou não ser utilizadas, sem prejuízo da proposta visual em que os modelos serão usados.

Para tal, é importante a análise de alguns critérios básicos, como o grau de autossimilaridade, o número e o padrão de repetições de um elemento bem como a importância deste na narrativa. O conceito de autossimilaridade foi definido por Mandelbrot (LINDENMAYER; PRUSINKIEWICZ, 1996, p. V, tradução nossa): “[...] quando cada pedaço de uma forma for geometricamente similar ao inteiro, ambos, a forma e a cascata que o gerou são chamados de autossimilares.”

Um exemplo do emprego de modelagem procedural e a ideia de autossimilaridade podem ser observados na imagem a seguir (FIG. 3), extraída de um fotograma do filme “Carros 2” (EUA, 2011, direção John Lasseter) que, em relação à sua composição, é dividida em três planos, e cada um destes possui um diferente grau de importância narrativa e há para cada plano diferente grau de automação de procedimentos. Sobre o emprego de técnicas de modelagem procedural no processo de construção dos cenários do filme Carros 2, Halstead, supervisor de cenários do filme, declarou em entrevista à edição 145 da revista 3D World:

Distante das paisagens, os designers de cenários preencheram a cidade de Londres com construções procedurais. “você deve achar algumas referências de carros nas construções procedurais, explica Hastead, “mas em geral, o departamento de arte concebeu estilos de edificações que capturaram o olhar e o sentimento de Londres. (Revista 3D World, Edição 145, p. 39, tradução nossa)



FIGURA 3: Fotograma extraído de uma cena do filme Carros 2, em que é possível ver os personagens em primeiro plano, uma representação da St. Paul's Cathedral de Londres em segundo plano e prédios da cidade no plano de fundo. Fonte: 3D WORLD .Londres: Barbara Robertson, 2000 - Mensal. ISSN 1470-4382.Edição 145, p. 39.

A partir da entrevista de Halstead, é possível deduzir que a equipe de artistas responsável pela criação dos modelos dos cenários utilizou técnicas de modelagem procedural de acordo com o critério de hierarquia visual presente na imagem. No primeiro plano, observam-se os personagens e neles as possibilidades de automação de procedimentos são limitadas no que concerne à modelagem. Isto se dá devido à baixa autossimilaridade dos modelos e possivelmente ao fato de que estes são compostos por elementos que reunidos visam externar características psicológicas dos personagens.

Em segundo plano, está um prédio londrino, a St. Paul's Cathedral<sup>3</sup>, e, devido a seu alto grau de autossimilaridade, há mais espaço para o emprego de técnicas procedurais. Porém, visto que este cumpre papel importante na contextualização do espaço no qual se desenrolam as ações, é essencial à narrativa e deve-se buscar um reconhecimento imediato do mesmo para que o público saiba que a ação ocorre na cidade de Londres. Para tal, é necessário que o modelo atenda com precisão os padrões de construção particulares do edifício original, sendo portanto mais interessante que o modelo seja dividido em partes e que na

<sup>3</sup>Mais informações em:< <http://www.londondrum.com/cityguide/st-pauls-cathedral.php>> acesso em 30/01/2013.

modelagem de cada uma faça-se emprego de procedimentos genéricos de modelagem procedural. Desenvolver procedimento capaz de gerar um modelo tão específico exigiria muito tempo de programação e o resultado seria pouco útil, visto que este possuiria aplicação única.

Já no terceiro plano, há um panorama da cidade de Londres e neste há apenas a preocupação de que o modelo represente uma cidade e que esta remeta ao padrão arquitetônico Londrino. Tal panorama pode ser gerado a partir da automatização de procedimentos, em que cópias de modelos tridimensionais de edifícios são distribuídas pela paisagem a partir de um mapa de influência. O processo de distribuição de cópias de modelos tridimensionais básicos também é conhecido pelo termo *Geometry Instancing*, e o mapa de influência (*weight map*, no termo em inglês) consiste em uma imagem bidimensional em que as cores de seus pixels são mapeadas a diferentes modelos tridimensionais e regiões da paisagem.

Existem diversos programas dotados de ferramentas para modelagem procedural, dentre os quais podem ser citados Blender, Maya, Cinema 4d, CityEngine, Softimage, Vue e Houdini. Cada um destes oferece, porém, grandes diferenças no método de trabalho, o que possivelmente se dá devido ao perfil de seus usuários finais e às aplicações a que cada programa se propõe. Algumas das diferenças são: a presença ou não de interface de programação visual (*logic nodes*), diferentes linguagens de programação utilizadas na criação de *scripts*, a integração de ferramentas procedurais em meio às ditas convencionais e seus métodos de renderização.

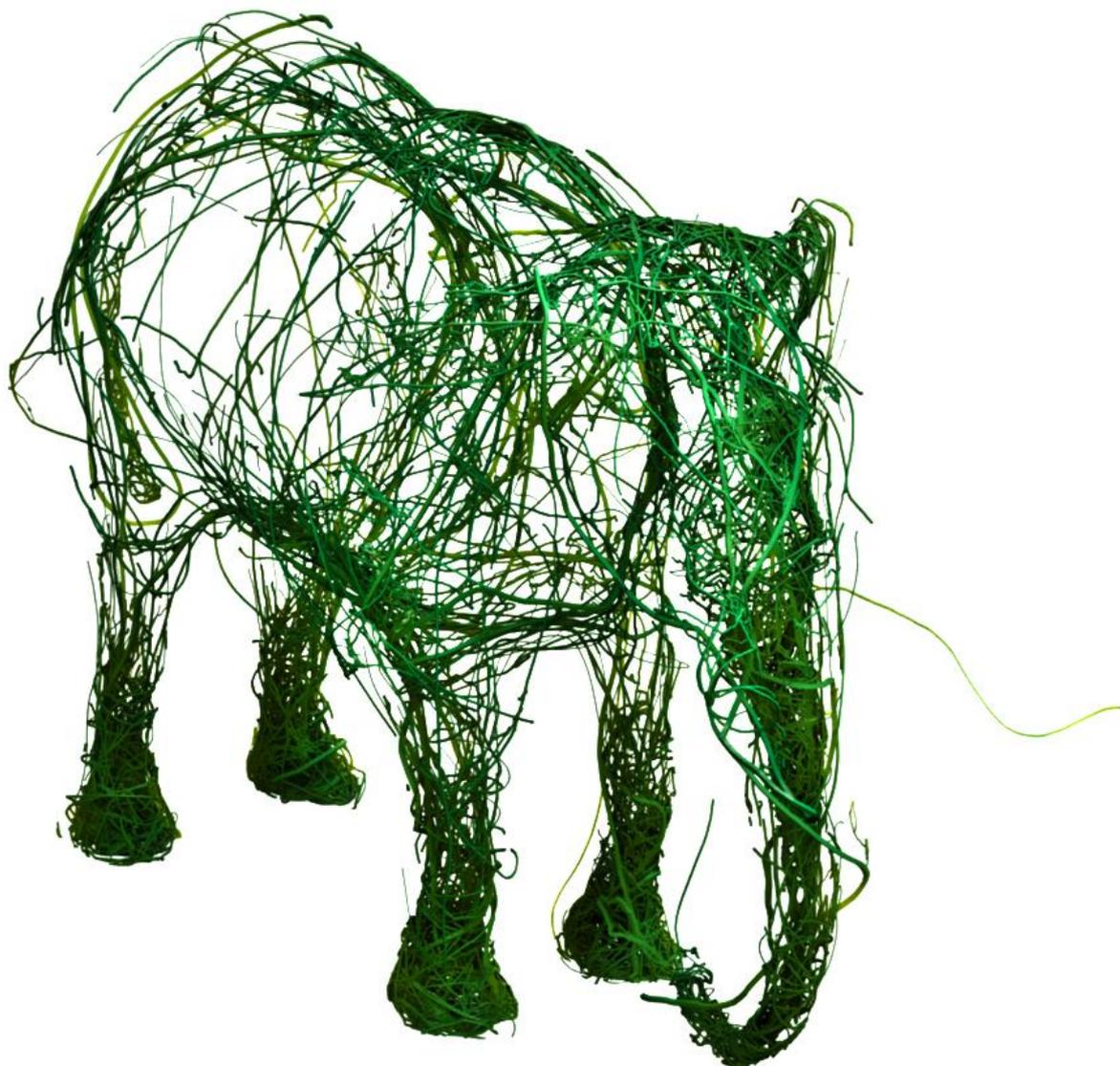


FIGURA 4: Estrutura rizomática gerada proceduralmente. Fonte: o autor.

Como exemplo de possibilidade visual resultante de técnicas de modelagem procedural, é possível ver na FIG. 4 o modelo de um elefante composto a partir de estrutura rizomática.

## 2 Modelagem procedural e uso de padrões de repetição

Em uma certa enciclopédia chinesa que se intitula Empório Celestial de Conhecimento Benevolente, está escrito, em suas remotas páginas, que os animais se dividem em: a) pertencentes ao imperador; b) embalsamados; c) domesticados; d) leitões; e) sereias; f) fabulosos; g) cães em liberdade; h) incluídos na presente classificação; i) que se agitam como loucos; j) inumeráveis; k) desenhados com um pincel muito fino de pêlo de camelo; l) et Cetera; m) que acabam de quebrar a bilha; n) que de longe parecem moscas<sup>4</sup>.

As técnicas de modelagem procedural possuem como princípio a repetição de padrões para a geração de formas, como as citadas modelagem por simetria e *spin*. Para gerar as formas tridimensionais e suas repetições, são necessários algoritmos e conhecimentos matemáticos para identificar e reproduzir padrões que produzem a sensação de verossimilhança nas imagens criadas digitalmente. Entre os padrões de arranjo matemáticos usados pelos algoritmos de programas de criação digital de imagens tridimensionais, podemos citar:

### 2.1 Pi

O primeiro e talvez mais importante padrão matematicamente descrito encontrado na natureza foi o Pi. Com este, pode-se descrever, dentre outros, a constante gravitacional, os números áureos bem como o teorema de Fibonacci.

A descoberta de Pi é incerta. Um dos registros mais antigos de Pi é um papiro egípcio atribuído a Ahmes que data de cerca de 1650 antes da Era Comum:

Físicos notaram a ubiquidade do Pi na natureza Pi é obvio nos discos da lua e do sol. A dupla espiral do DNA é revolucionada em torno do Pi. Pi se esconde no arco-íris, e se senta na pupila do olho e quando uma gota de chuva cai na água Pi emerge nos anéis que se propagam. Pi pode ser encontrado nas ondas, ondulações e espectros de todos os tipos e portanto o Pi ocorre em cores e musica. PI recentemente se mostrou presente na teoria de cordas. (PRESTON, 1992, p. 5, tradução nossa)

De acordo Richard Preston (1992), Ferdinand Liberman, matemático alemão, comprovou em efeito, em 1882, que Pi é um número transcendental, ou

---

<sup>4</sup>BORGES, Jorge Luis. *Outras inquisições*. trad. Davi Arrigucci Jr., São Paulo: Companhia das Letras, 2007.

seja, este não poderia ser representado sem a omissão de decimais, em papel, ainda que este tivesse o tamanho do universo.

## 2.2 Phi

Outro número recorrente na natureza é Phi, introduzido na Europa ocidental pelo matemático Leonardo Fibonacci, em 1902, através de seu livro intitulado *Líber Abaci*. Phi é encontrado na natureza, na distribuição de sementes, folhas, pétalas e sépalas em algumas plantas.

Phi é também amplamente empregado na indústria e no design. Os cortes do papel industrializado, bem como cartões de crédito e vários visores de aparelhos eletrônicos respeitam o padrão dos retângulos áureos de Fibonacci.



FIGURA 5: Fotografia de uma espécie de couve-flor, em que é possível identificar padrões geométricos descritos pelo padrão Phi. Fonte: <<http://tinylittleanthill.wordpress.com/tag/romanesco-cauliflower>>. acesso em 30/01/2013.

Como é possível identificar na FIG 5, os padrões geométricos descritos pelo padrão Phi podem ser encontrados em diversas estruturas da natureza, como em espécies vegetais.

## 2.3 Sistema de Lindenmayer

A beleza das plantas atraiu a atenção dos matemáticos por séculos. Traços geométricos conspícuos tais como a simetria bilateral das folhas, a simetria rotacional das flores, e os arranjos helicoidais das escalas em pinhas tem sido estudados extensivamente. (LINDENMAYER; PRUSINKIEWICZ, 1996, p. V, tradução nossa.)

Com o trecho acima, Lindenmayer e Prusinkiewicz iniciam seu livro intitulado *The Algorithmic beauty of plants*, onde descrevem o chamado *L-system*, hoje o sistema com mais ampla utilização na modelagem procedural de plantas e formas naturais, e que foi desenvolvido pelo biólogo húngaro Aristid Lindenmayer em 1968.

O “*L system*” não é um padrão propriamente dito, mas um código gramatical composto de letras e símbolos com os quais se compõem linhas que descrevem o comportamento padrão de um organismo, a partir das quais é possível a automação de procedimentos capazes de gerar modelos dotados de alto grau de verossimilhança. O sistema L foi predecessor de diversos outros sistemas, como exemplificado na FIG. 5.

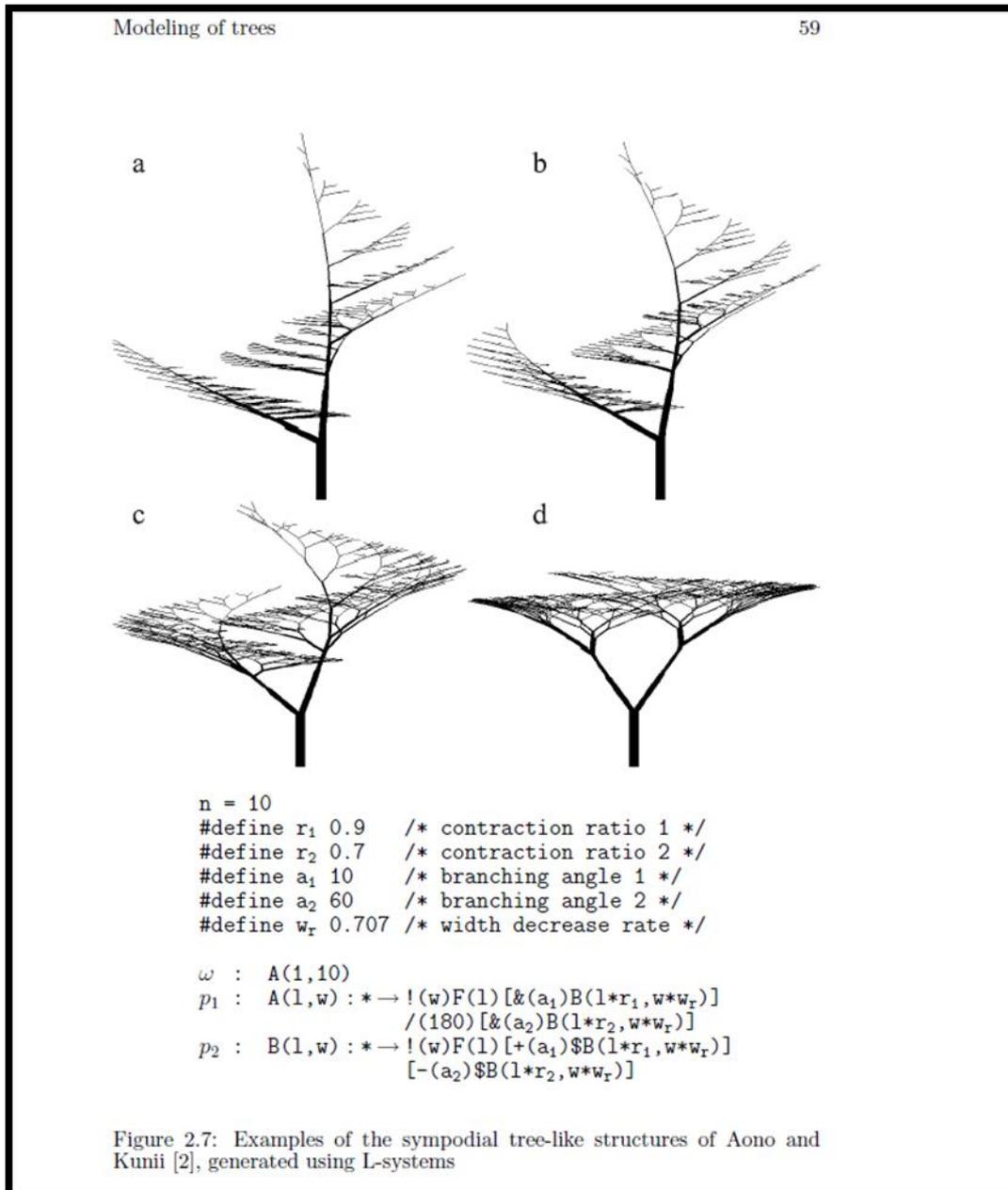


FIGURA 6: Exemplos de estruturas semelhantes a árvores simpodiais de Aono e Kunii [2], geradas usando sistemas – L. Fonte: LINDENMAYER, Aristid; PRUSINKIEWICZ, Przemyslaw, p. 59, tradução nossa.

## 2.4 Sistema de quadros estruturais

Apesar dos vários padrões de arranjo matemáticos existentes, talvez seja mais interessante ao artista visual conhecer a teoria de Marvin Minsky sobre a “representação em quadros estruturais” descrita em sua obra intitulada *A framework for representing knowledge*, publicado em 1974. Nela, o autor busca analisar certos

fenômenos da inteligência humana de modo que seja possível uma comunicação mais efetiva e econômica. A teoria de quadros estruturais alega que quando nos deparamos com uma situação nova, buscamos em nossa memória um quadro, que é uma “estrutura de dados utilizada para a representação de uma situação [ou objeto] estereotipada” (MINSKY, 1974, p.1, tradução nossa.). A estrutura deste, caso necessário, se adapta de modo a incluir neste a nova situação encontrada.

Pode-se pensar em um “quadro” como uma rede de nós e relações. Os “níveis mais altos” de um “frame” são fixos, e representam coisas que são sempre verdadeiras sobre uma suposta situação. Os níveis mais baixos tem muitos terminais – “compartimentos”, que devem ser preenchidos por exemplos específicos ou dados. Cada terminal pode especificar condições às quais as suas atribuições devem corresponder. (As atribuições em si mesmas são normalmente “quadros secundários menores). Condições simples são especificadas por marcadores que podem requerer que uma atribuição de terminal seja uma pessoa, um objeto de valor suficiente, ou um apontamento para um “quadro secundário” de um certo tipo. Condições mais complexas podem especificar relações entre as coisas atribuídas a vários terminais. (MINSKY, 1974, p.1, tradução nossa.)

Desta forma, tudo aquilo que é visto pela primeira vez é categorizado com base em requisitos conhecidos e, caso possível, armazenado em um quadro já existente, como exemplificado na FIG 6.

Padrões tais quais os acima citados podem servir de auxílio para artistas visuais, modeladores e animadores na criação de modelos complexos e verossímeis. Para tal, é possível o uso de softwares de modelagem e animação 3d, alguns deles previamente apresentados neste trabalho. A título de ilustração, será apresentado o sistema de programação visual ICE, ou Ambiente de Criação Interativa<sup>5</sup>.

---

<sup>5</sup>Disponível em:

[http://download.autodesk.com/global/docs/softimage2013/en\\_us/userguide/index.html#20/01/2013](http://download.autodesk.com/global/docs/softimage2013/en_us/userguide/index.html#20/01/2013)>  
acesso em 30/01/2013.

Tradução nossa.

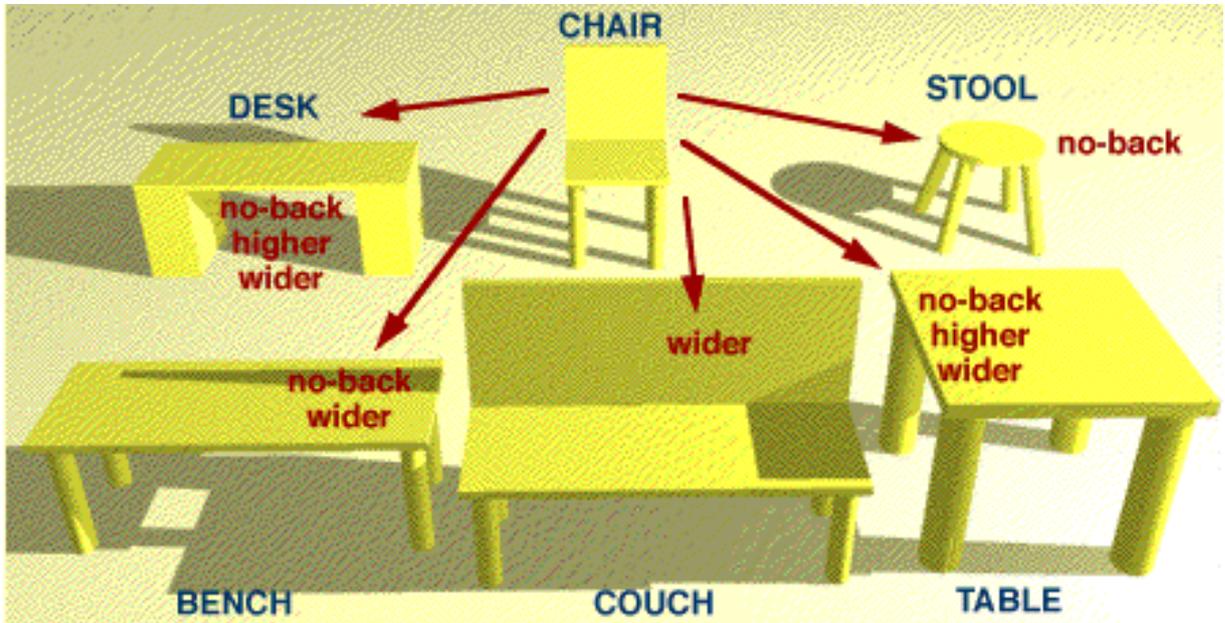


FIGURA 7: Exemplificação visual do sistema de quadros estruturais de Minsky. Fonte: <<http://web.media.mit.edu/~minsky/papers/Frames/frames.html>> acesso em 30/01/2013.

### 3 Interactive Creative Environment (ICE)

O *Interactive Creative Environment* (Ambiente de Criação Interativa, em português) ou simplesmente ICE é um sistema de programação visual de propriedade da Autodesk que pode ser encontrado em diversos produtos desta, dentre eles Maya, Softimage, 3D Max e AutoCAD. O uso do ICE dispensa o conhecimento de linguagens de programação tais como C++ ou Python uma vez que este baseia-se em *graphic nodes*<sup>6</sup>, que combinados criam uma *ICE tree*<sup>7</sup> que descreve o processo de automação de um procedimento. Devido à natureza visual e amigável dos *graphic nodes*, o ICE pode ser eficiente para demonstrar os princípios de modelagem procedural para artistas visuais.

É importante ressaltar, entretanto, que o ICE não é o único sistema amigável para a criação de modelos procedurais. Alguns programas, como o *E-on*

<sup>6</sup>Representação visual de dados, modificadores e funções lógicas para a criação de procedimentos

<sup>7</sup>Conjunto interligado de *graphic nodes* que modifica dados e informações de entrada para, por exemplo, criar modelos procedurais

*Vue*, possuem interface amigável e em outros pode se obter resultados similares através da elaboração de *scripts*.

### 3.1 Tipos de dados

Cada *graphic node* pode receber informações de entrada para efetuar alguma operação e produzir informações de saída resultante dos parâmetros informados. Tanto a entrada e saída dos *graphic nodes* aceitam determinados tipos de dados. O Quadro 1, traduzido e comentado a partir do manual do usuário do software Autodesk Softimage, traz algumas informações básicas sobre estes tipos de dados.

Quadro 1 - Tipos de dados em *graphic nodes* do ICE<sup>8</sup>

Cor	Tipo	Descrição
	<i>Polymorphic</i>	Aceita vários tipos de dados.
	<i>Boolean</i>	Um valor booleano: verdadeiro ou falso
	<i>Integer</i>	Um número positivo ou negativo sem frações decimais, por exemplo, 7, -2, ou 0.
	<i>Scalar</i>	Um número real representado como um valor decimal por exemplo, 3.14. Internamente este é um valor de ponto flutuante de precisão única <sup>9</sup> .
	<i>Color</i>	Quatro valores que descrevem uma cor em seus canais de vermelho, verde, azul e alfa.
	<i>2D Vector</i>	Um vetor bidimensional [x, y] cujas entradas são números reais, como por exemplo, coordenada UV
	<i>3D Vector</i>	Um vetor tridimensional [x, y,z] cujas entradas são números reais, como por exemplo, posição, velocidade, ou força.
	<i>4D Vector</i>	Um vetor quadridimensional [w, x, y, z] cujas entradas são números reais.
	<i>Quaternion</i>	Um quaternion <sup>10</sup> [x, y, z, w]. Quaternions são geralmente usados na representação de uma orientação. Quaternions podem facilmente ser misturados e interpolados, e é uma possível solução quando frente a problemas de <i>gimbal-lock</i> <sup>11</sup> .
	<i>Rotation</i>	Uma rotação representada por um eixo vetor [x, y, z] e um ângulo apresentado em graus.
	<i>3x3 Matrix</i>	Uma matriz 3 por 3 cujas entradas são números reais. Matrizes 3 por 3 são normalmente usadas para representar rotação e escalonamento.

<sup>8</sup> Fonte:< [http://download.autodesk.com/global/docs/softimage2013/en\\_us/userguide/index.html](http://download.autodesk.com/global/docs/softimage2013/en_us/userguide/index.html)> acesso em 30/01/2013.

<sup>9</sup> O cálculo de números reais é um problema a parte para os cálculos computacionais, já que há o risco de que de tal cálculo resulte um número transcendental (dotado de dízimas infinitesimais). Apenas em 1985 houve a padronização dos formatos de representação para números de ponto flutuante (IEEE 754), padronização tal que foi revisada em 2008 (IEEE 754-2008). Este padrão é popularmente conhecido como 32 bits.

<sup>10</sup> *Quaternion*, definido por Hamilton este em 1843 e reformulado por Gibbs em 1901.

<sup>11</sup> A perda de um grau de liberdade de rotação.

	<i>4x4 Matrix</i>	Uma matriz 4 por 4 cujas entradas são números reais. Matrizes 4 por 4 são normalmente usadas para representar transformações.(escalonamento, rotação e translação).
	<i>Shape</i>	Uma forma geométrica primitiva, ou uma referência um objeto na cena.Esse tipo de dados é usado para determinar o formato de partículas.
	<i>Geometry</i>	Uma referência a um objeto geométrico na cena, como uma engrenagem poligonal, curva NURBS, superfície NURBS, ou nuvem de pontos. O usuário pode fracionar a superfície de uma geometria a fim de gerar locais de superfícies para emitir partículas.
	<i>Surface Location</i>	Um local na superfície de um objeto geométrico. A localidade é atrelada à superfície do objeto de forma que mesmo que o objeto sofra transformação ou deformação, esta se move com o objeto e se mantém na mesma posição relativa.
	<i>String</i>	Uma linha de atributos. Estas podem ser usadas por verticilos padrão para armazenar arquivos de <i>path</i> , por exemplo.
	<i>Topology</i>	A topologia de um objeto de malha poligonal. É usado para modelagem em ICE.
	<i>Execution</i>	Não é um tipo de dados no sentido convencional. O usuário conecta portas <i>Execution</i> tais como a saída de um <i>Set Data</i> em um <i>Execute</i> ou nodes raiz para controlar a ordem de execuções em uma árvore.
	<i>Reference</i>	Também não é um tipo de dado no sentido convencional. Esse é uma referência a um objeto, Parâmetro ou atributo na cena, expresso como linha de atributos. O usuário pode entrelaçar referências como descrito em <i>Daisy-chaining References</i> <sup>12</sup> .

### 3.2 Exemplo de uso do ICE, com representação por *graphic nodes*

Para uma melhor compreensão acerca de modelagem procedural, representamos uma cena de exemplo e demonstramos como se deu seu processo de criação com o uso do ICE, integrado ao software Autodesk XSI, e seus *graphic nodes*. A FIG. 8 possui duas representações da mesma cena. À direita, faz-se a demonstração através de polígonos tridimensionais representados virtualmente. À esquerda, a cena é retratada por *graphic nodes* que juntos compõem o que se chama *ICE tree*.

<sup>12</sup> Disponível em: < [http://download.autodesk.com/global/docs/softimage2013/en\\_us/userguide/index.html](http://download.autodesk.com/global/docs/softimage2013/en_us/userguide/index.html)> acesso em 30/01/2013.

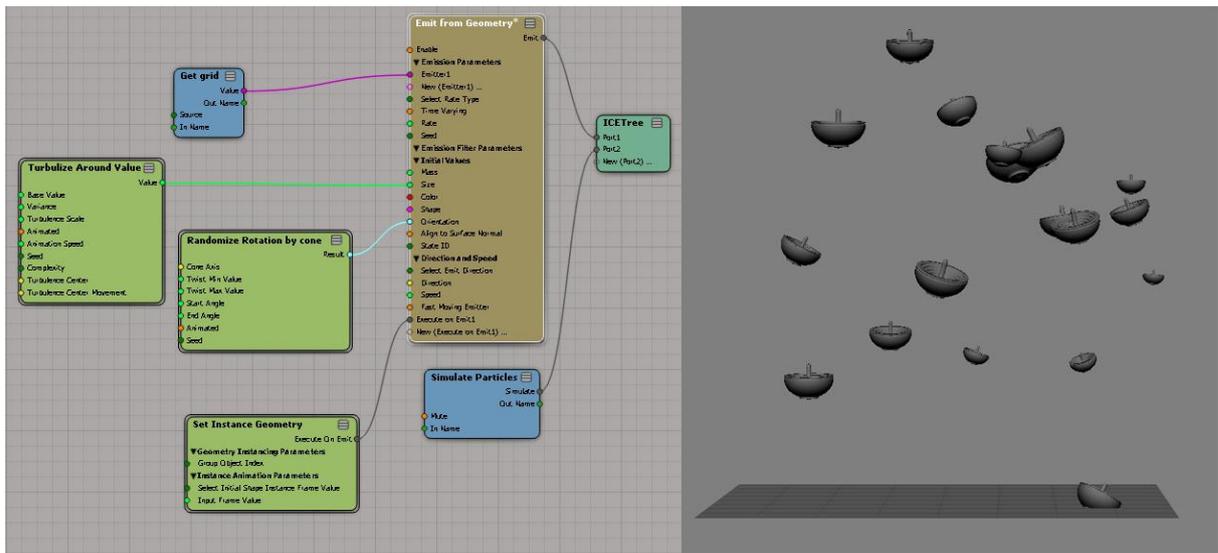


FIGURA 8: Duas possíveis representações da mesma cena, sendo à esquerda uma realizada por meio de *graphic nodes* e à direita outra realizada por representação virtual de modelos poligonais tridimensionais. Fonte: o autor.

À primeira vista, estas representações não parecem ter muito em comum e tampouco fica clara a relação entre elas. Por isso ilustraremos a seguir a criação da cena, apresentando em paralelo as representações poligonais e de *graphic nodes* referentes a cada estágio, salvo o processo de modelagem poligonal da geometria de instância<sup>13</sup>, já que isto fugiria ao tema central do presente trabalho.

Primeiramente, para tal resultado modelou-se uma geometria de instância e outra que serve como base para a emissão das partículas. Neste caso, a geometria ligada à emissão é uma malha poligonal simples, que representa a forma de um quadrado e que foi gerada através dos menus de modelagem do Softimage, como mostrado a seguir.

<sup>13</sup>Geometria que dá forma às partículas, que a princípio são pontos e, como tal, não podem ser visualizadas.

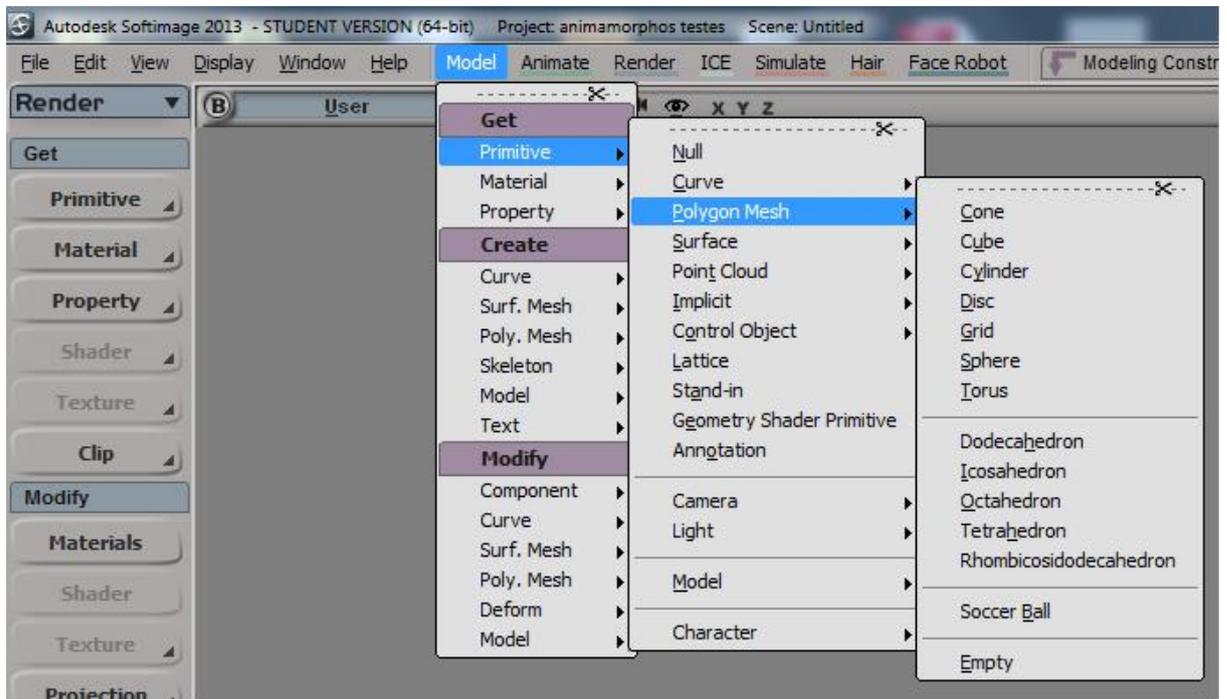


FIGURA 9: Demonstração de uma das possíveis maneiras de se gerar uma malha poligonal básica através dos menus do software Autodesk Softimage. Fonte: o Autor.

Com a malha já presente, cria-se uma *point cloud* vazia, isto é, a descrição do espaço virtual no qual se dá a modelagem ou simulação. Uma *point cloud* vazia não tem por si um volume. Este é calculado a cada quadro com base no espaço virtual ocupado pelas partículas. Pode-se imaginar a *point cloud* como uma caixa que se expande a fim de sempre comportar dentro de si as partículas. A *point cloud* vazia foi criada também pelo uso dos mesmos menus, como pode ser visto na FIG. 10.

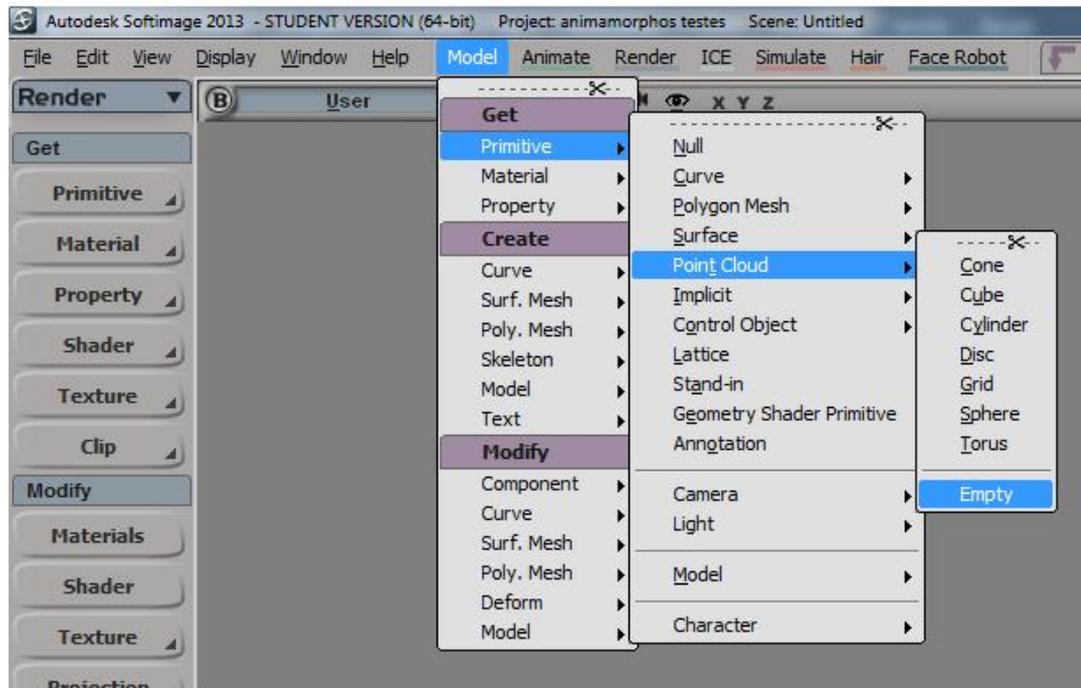


FIGURA10: Demonstração de uma das possíveis maneiras de gerar uma point cloud vazia através dos menus do software Softimage. Fonte: o Autor.

Uma vez criada a *point cloud*, acessa-se o editor do ICE através das teclas de atalho [Alt]+[9], e neste deve ser acessado o menu *Create* e selecionada a opção *Simulated ICE Tree*, como indicado pela Figura 11.

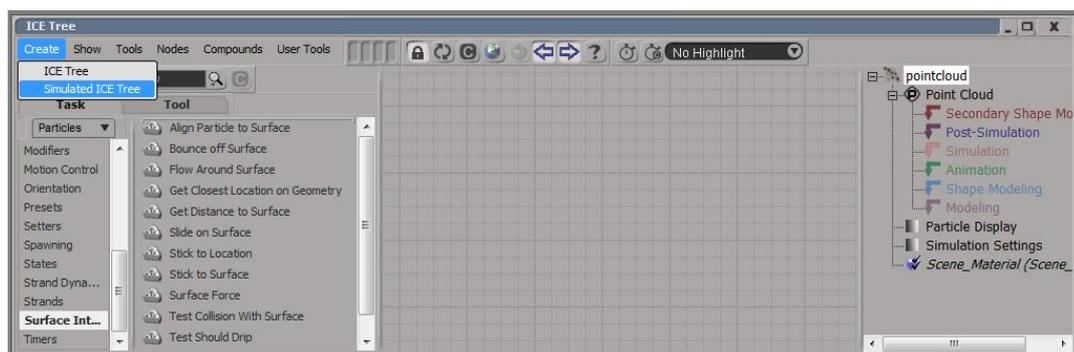


FIGURA 11: Demonstração de uma das possíveis maneiras de gerar uma ICE tree com simulação através dos menus do software Softimage. Fonte: o Autor.

Uma vez criada uma *ICE tree* simulada, o passo seguinte foi criar uma emissão a partir de geometria, e para tal utiliza-se a barra de busca que se encontra à esquerda do editor ICE. Uma vez localizado o *node* desejado, arrasta-se o mesmo para a área da *ICE tree*, clicando no *node* com o botão esquerdo e arrastando-o.

Com o *node* posicionado, cria-se então uma conexão entre as portas de execução. Nota-se, no entanto, que o *node* a princípio aparece em cor vermelha, indicando um erro. Isto ocorre porque ainda não foi definida a geometria base para emissão, como indicado pela FIG 12.

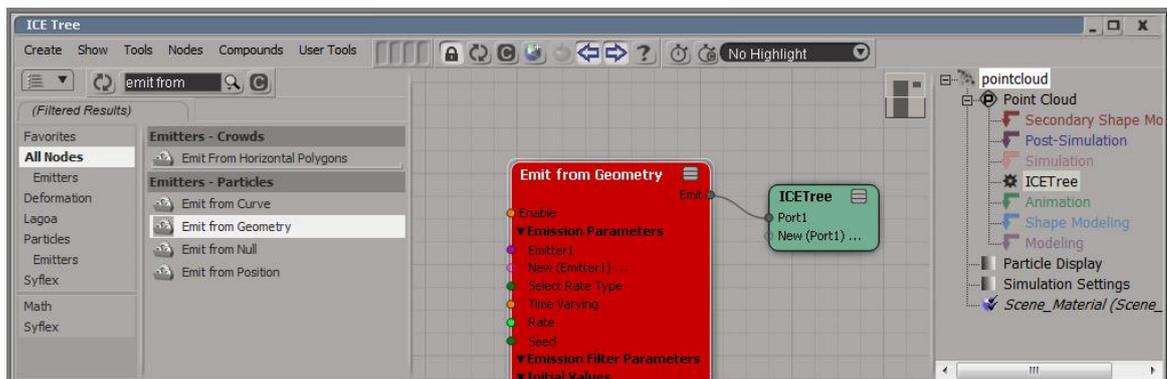


FIGURA 12: Demonstração de uma das possíveis maneiras de acessar o *node* Emit from geometry usando a ferramenta de busca presente no editor ICE do software Softimage. Fonte: o Autor.

Para solucionar isto, adiciona-se, da mesma forma como com o *node* *Emit from geometry*, um segundo *node* chamado *get data* e é criada uma ligação ente sua porta *Value* e a porta *Emitter* do *node* *Emit from geometry*, como demonstrado pela FIG 13.

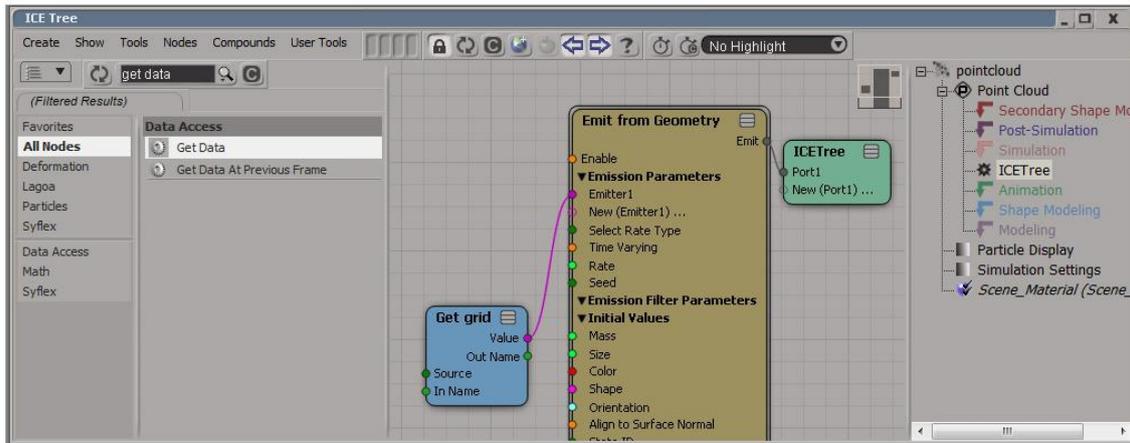


FIGURA 13: Demonstração de uma das possíveis maneiras de acessar o node Get data usando a ferramenta de busca presente no editor ICE do software Softimage. Fonte: o Autor.

Este *node* também aparece inicialmente em vermelho, em sinal de erro. Isto ocorre porque ainda não foi definida referência a nenhuma espécie de dados. Através do *node get data*, pode-se acessar quase qualquer dado relacionado a um objeto, como visto na FIG 14.

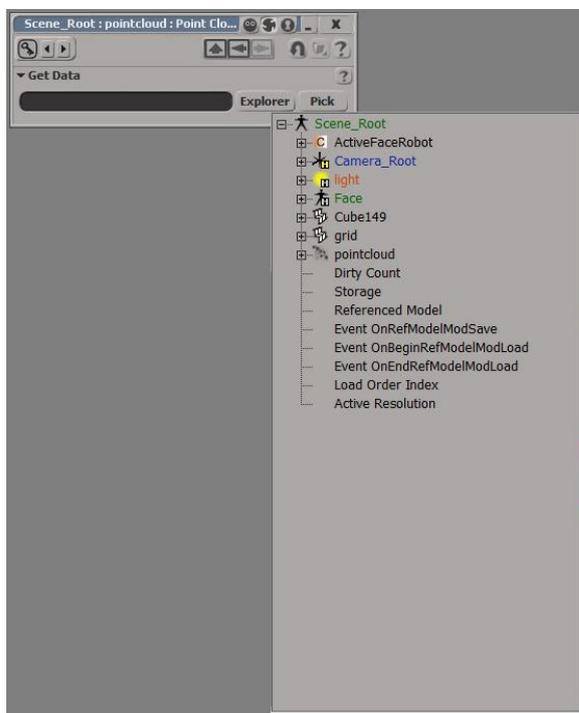


FIGURA 14: Demonstração de uma das possíveis maneiras de atrelar um dado ao *node Get data* usando o menu *Explorer* presente no editor ICE do software Autodesk Softimage. Fonte: o Autor.

Na situação apresentada, o dado buscado foi simplesmente o nome (referência) da geometria de emissão. Uma forma de determinar tal referência é ao acessar o menu inicial do *node get data* através de duplo clique, e selecionando então o nome da geometria no menu oculto do botão Explorer. Feito isso, observa-se que ambos os *nodes* não exibem mais a mensagem de erro, como pode ser visto na FIG. 15.

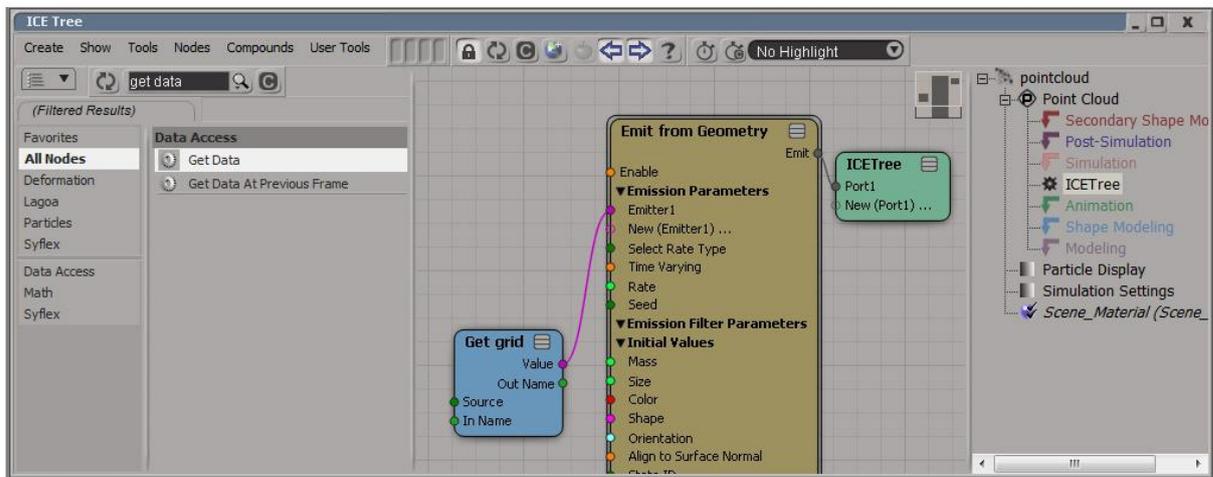


FIGURA 15: Demonstração de uma conexão bem sucedida entre *nodes* utilizando *ICE* no software Autodesk Softimage. Fonte: o Autor.

Realizado este processo, as partículas são geradas, mas permanecem porém imóveis atadas à malha, pois para que estas se movimentem é necessário o uso de um terceiro *node* chamado *simulate particles*, o qual pode ser acessado da forma ilustrada para os *nodes* *Emit from geometry* e *Get data*. Uma vez adicionado, a *ICE tree* *Simulate particles* deve ser ligada à alguma porta do tipo *execute*. No caso da *ICE tree* por nós representada, a conexão foi criada com a porta dois da *ICE tree*. As FIG. 15 e 16 ilustram este estágio.

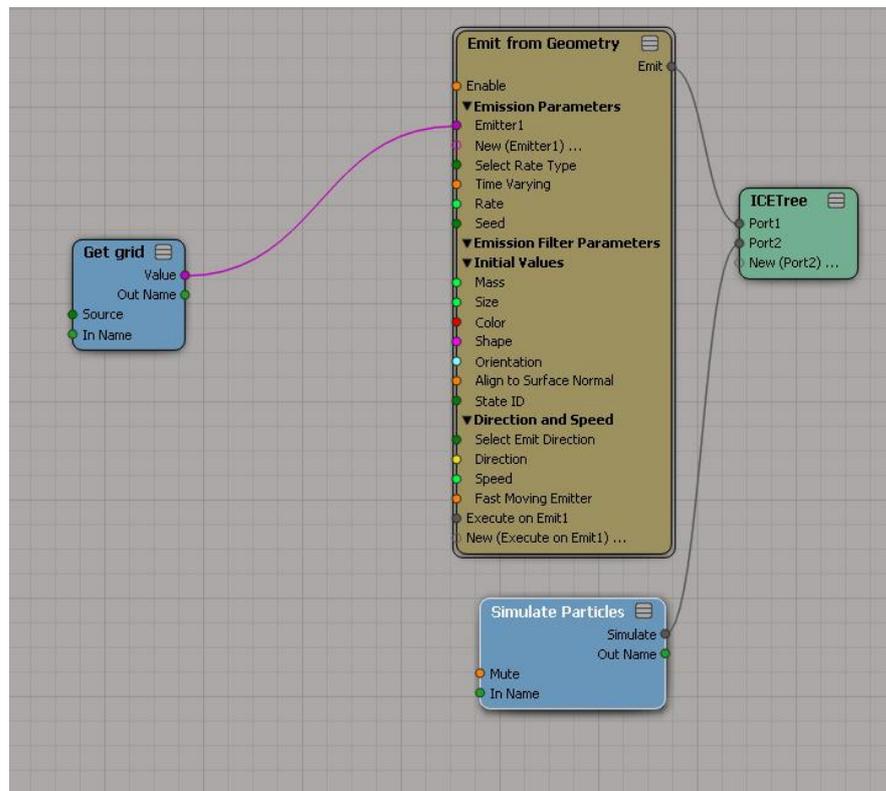


FIGURA 16: Demonstração de uma emissão simples de partículas utilizando *graphic nodes* utilizando em ICE no software Autodesk Softimage. Fonte: o Autor.

O próximo passo foi o acréscimo e a conexão de mais dois *nodes*, sendo *set instance geometry* ligado à porta *Execute on emission* do *node Emit from geometry* e o *randomize rotation by cone* que foi ligado à porta *orientation*, também do *node Emit from geometry*. Deve-se selecionar, então, a referência do *node* para o *node Set instance geometry*. No caso da cena demonstrada, apresenta-se apenas uma geometria de instância e por isso tal referência pode ser definida da mesma forma citada para o *node Get data*, através do menu oculto do botão *Explorer*. Definida a geometria de instância, a cena é geometricamente representada nas FIG. 17 e 18.

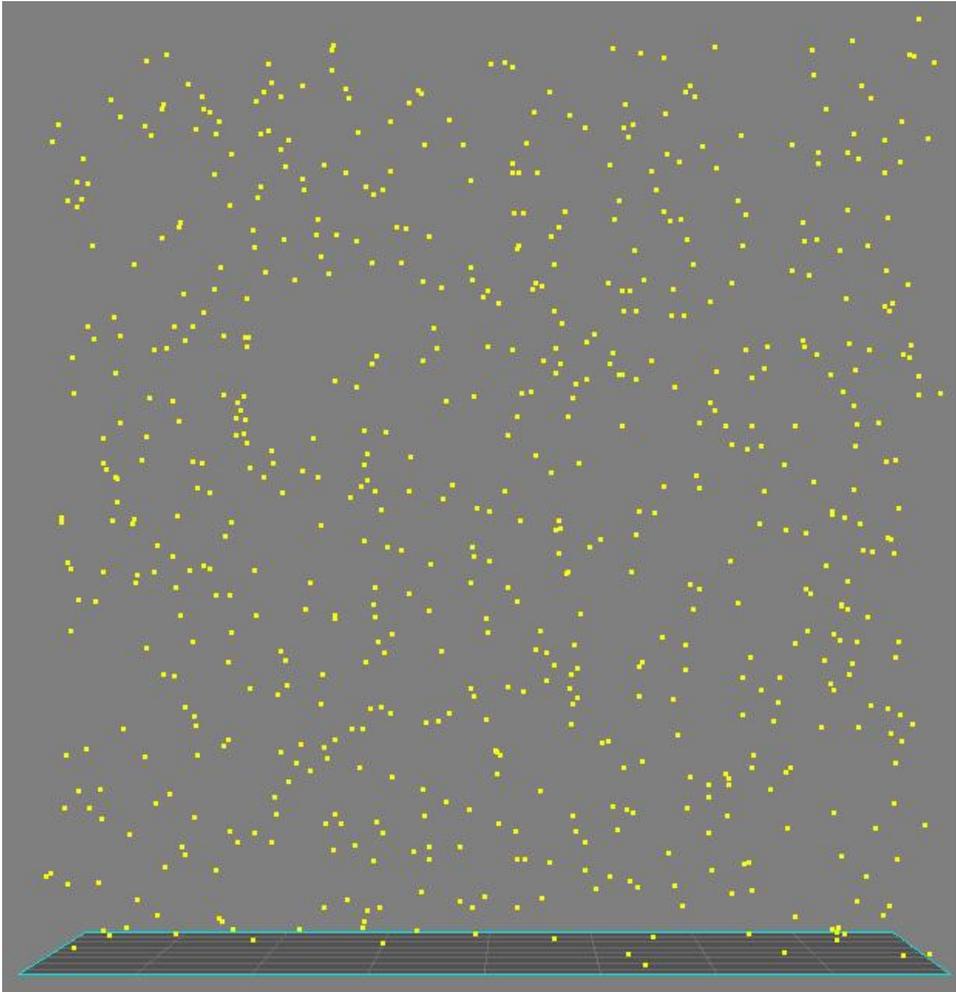


FIGURA 17: Demonstração de uma emissão simples de partículas utilizando representação virtual de polígonos tridimensionais. Fonte: o Autor.

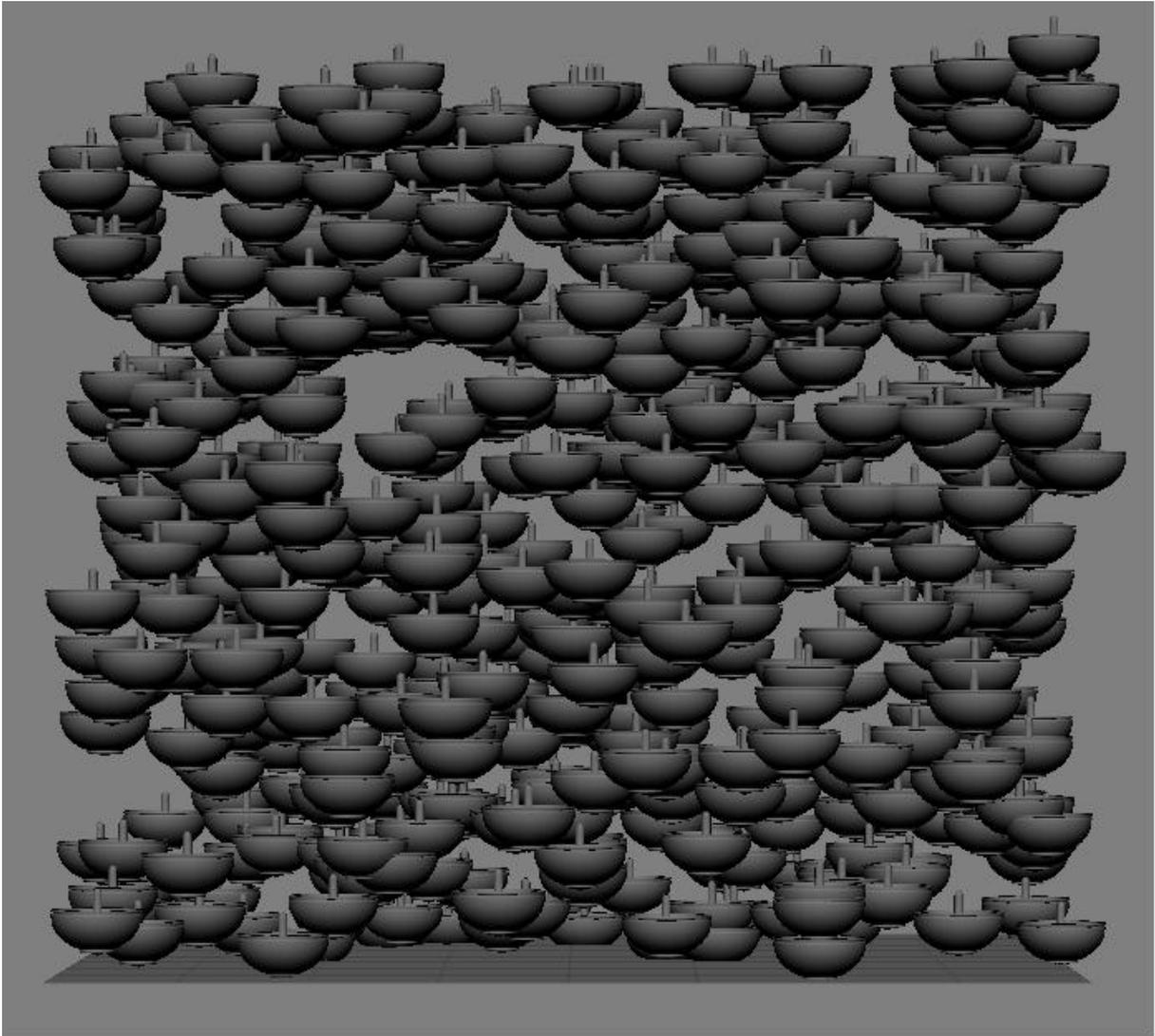


FIGURA 18: Demonstração de uma emissão simples de partículas utilizando a representação virtual de polígonos tridimensionais e geometria de instância. Fonte: o Autor.

Adicionou-se em seguida um terceiro *node*, chamado *turbulize around value*, o qual foi ligado à porta *size* do *node Emit from Geometry*, a fim de criar partículas de diferentes tamanhos. A princípio, a variação de tamanhos foi excessiva para a cena, como pode ser visto na FIG. 19.

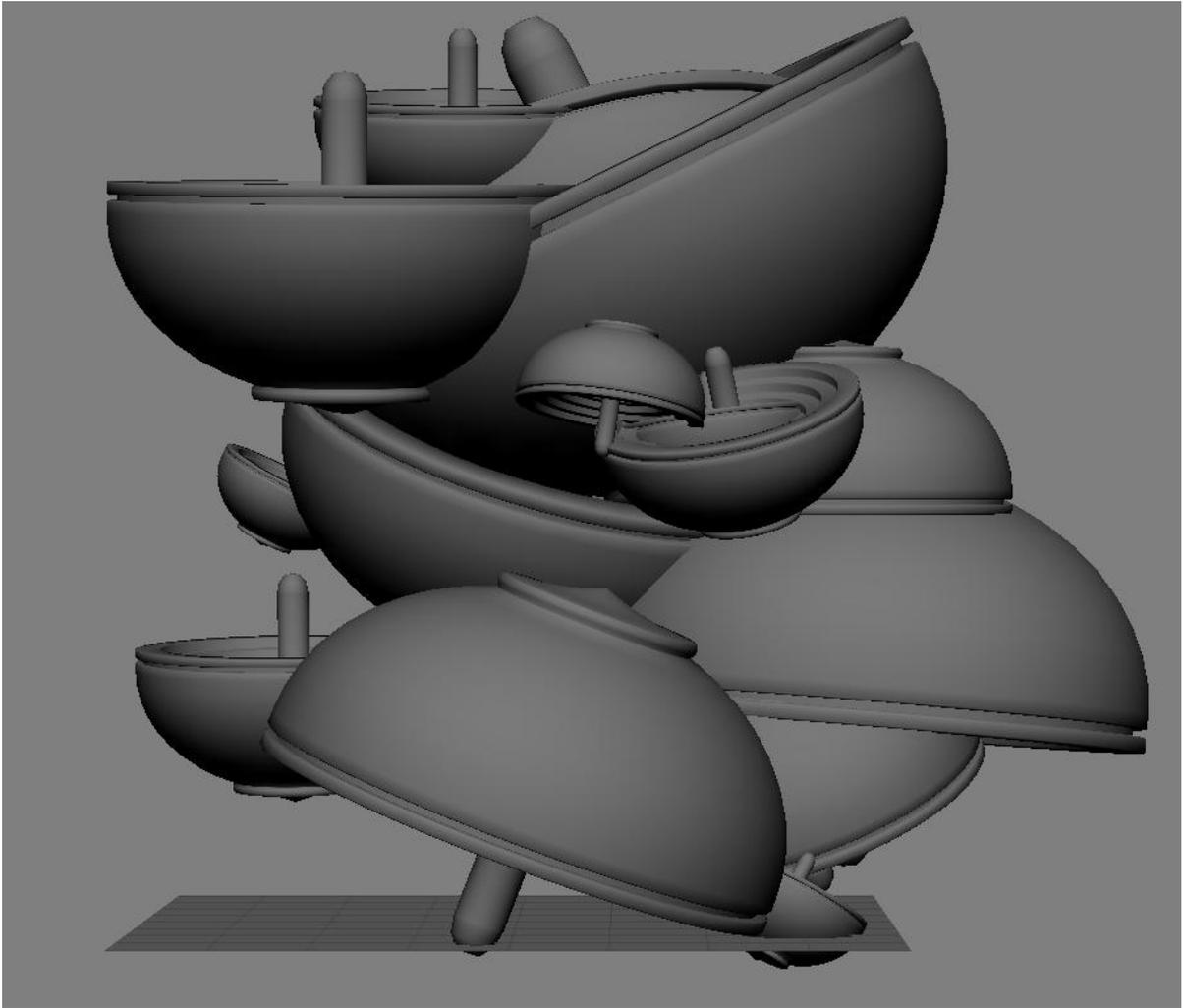


Figura 19: Demonstração de uma emissão simples de partículas utilizando representação virtual de polígonos tridimensionais e geometria de instancia, além de partículas com tamanhos e orientações randomizadas. Fonte: o Autor.

Foram configurados, então, os valores do *nodes Emit from geometry*, bem como do *turbulize around value*, acessando seus menus a partir de duplo clique e então atribuindo a eles os valores representados nas FIG. 20 e 21.

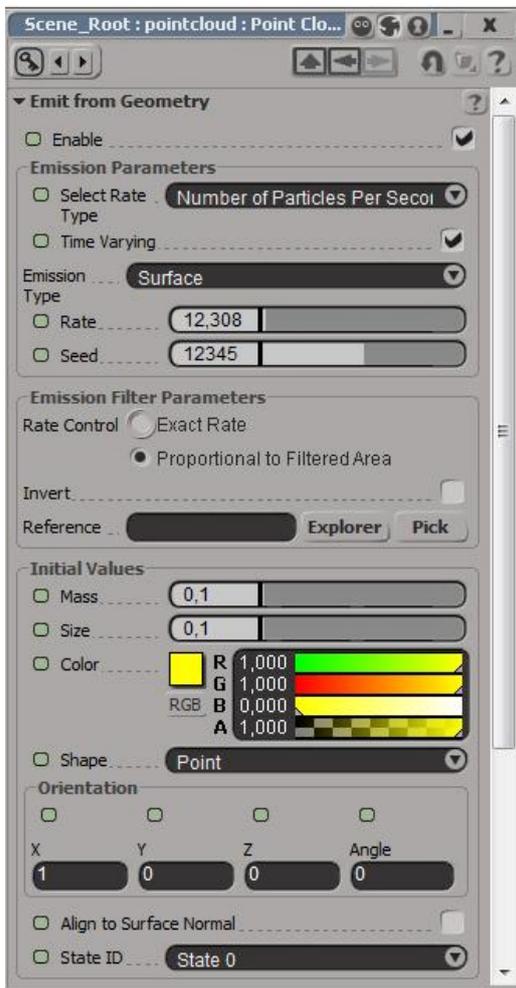


FIGURA 20: Apresentação de um menu de configuração do *node Emit from geometry*.  
Fonte: o Autor.

Após concluir tais configurações, foi possível produzir a imagem resultante apresentada na FIG 2, indicada no início deste trabalho.

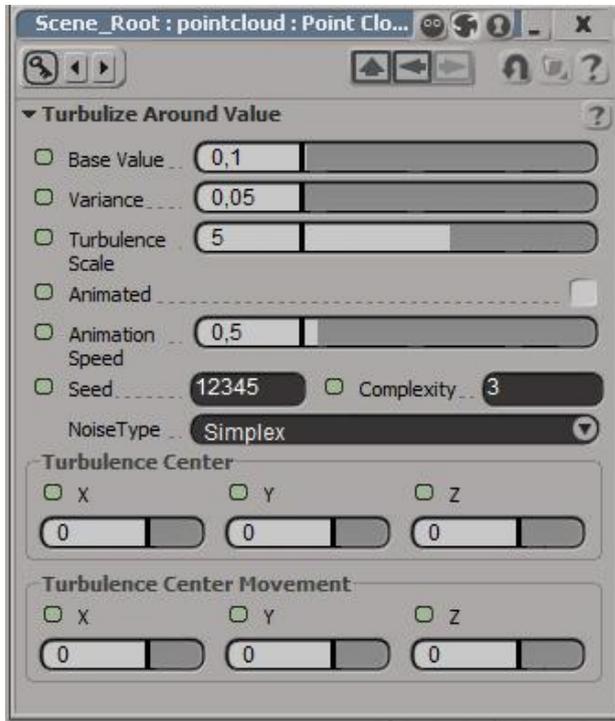


FIGURA 21: Apresentação de um menu de configuração do *node Turbulize around Value*. Fonte: o Autor.

### 3.3 Exemplos de *compounds* criados em *ICE*

A fim de melhor ilustrar algumas possibilidades do uso do *ICE* aliado à síntese de geometria, optou-se por citar alguns *compounds* criados em *ICE* capazes de gerar modelos a partir da automação de procedimento.

#### 3.3.1 Rizoma

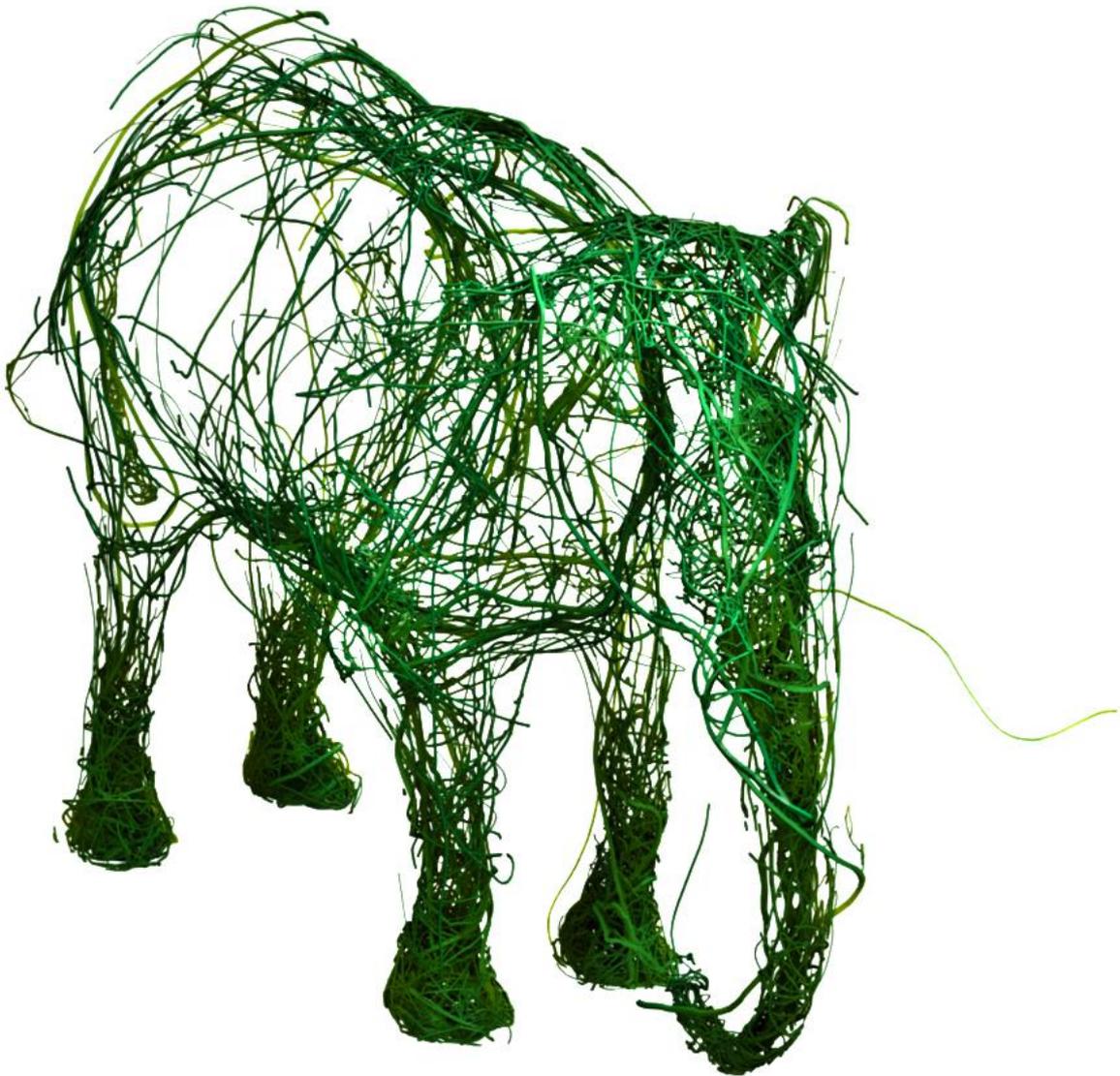


FIGURA 22: *Compound* criado pelo autor com finalidade de representar artisticamente raízes valendo-se de algoritmo de colonização de espaço. Fonte: o Autor.

A fim de obter maior controle sobre o modelo final, “Rizoma” trabalha a partir de múltiplas malhas geométricas como base para emissão de partículas. Os

parâmetros iniciais relacionados à emissão são compartilhados por todas as partículas emitidas a partir destas malhas, simplificando assim o controle do artista sobre o processo de emissão. As partículas emitidas movem-se dentro de uma malha geométrica a qual fornece dados a elas a fim de guiá-las. As partículas que se movem para longe desta malha guia são extintas, para maior controle sobre o modelo final e também para evitar que a CPU se ocupe com cálculos desnecessários.

À medida que as partículas se deslocam, estas geram um arranjo de pontos posicionados com base em dados obtidos a partir da descrição de suas trajetórias. Estes pontos são então ligados por linhas dando origem ao que chamamos de *strand trails*.

A partir de informações das posições de pontos aleatórios dispostos nestes *strands* gera-se nova emissão. Todas aquelas partículas provenientes desta emissão secundária se movimentam também com base na descrita malha guia e compartilham parâmetros comuns, diferentes, no entanto, daqueles parâmetros das partículas emitidas a priori.

Com base nas posições destas novas partículas é gerado ainda um processo terciário de emissão que se dá do mesmo modo que o segundo.

As trajetórias das partículas podem ainda ser influenciadas por gravidade e tropismos.

### 3.3.2 Kristinka Hair

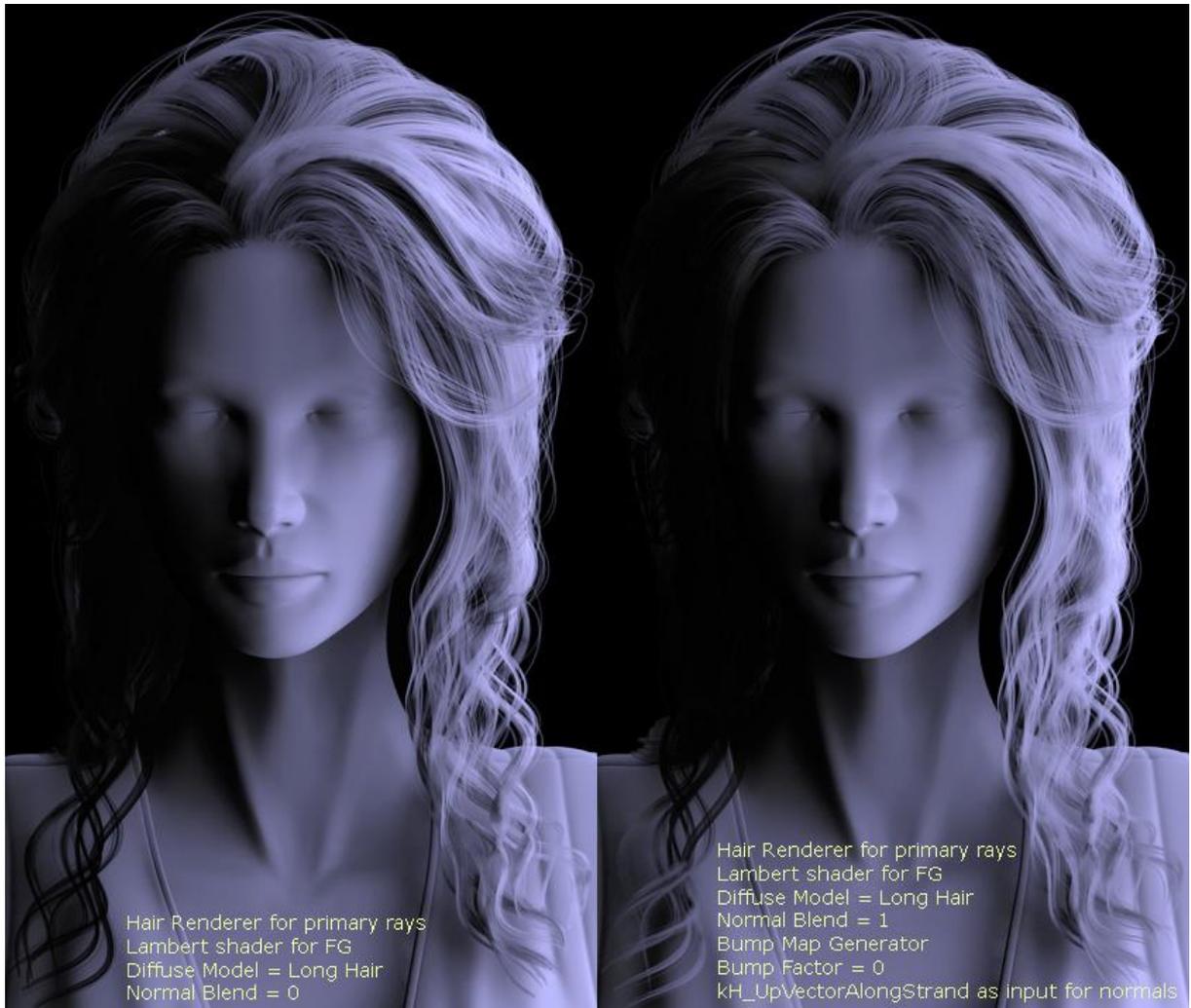


FIGURA 23: Detalhe de *FG experiment*. Fonte: <<http://www.si-community.com/community/viewtopic.php?f=24&t=447&start=40>> acesso em 30/01/2013.

De autoria de Mihail Dzhurev o “*Kristinka Hair*” baseia-se na criação de *strands* a partir da trajetória de partículas emitidas tendo como base uma malha poligonal. Estas partículas são guiadas, a princípio, pelas coordenadas UV de NURBS ou curvas. Os pontos que compõem os *strands* utilizados para a criação do modelo são rearranjados através de equações, criando assim ondulações e mechas distintas.

“*Kristinka Hair*” não é um *compound* propriamente dito, mas sim um conjunto destes, cabendo ao usuário fazer opção pelo uso daqueles que melhor se enquadrem às suas necessidades.

Tendo em vista a complexidade dos *compounds* que constituem “*Kristinka Hair*”, não será feita neste trabalho análise aprofundada destes.

### 3.3.3 Knit Strands



FIGURA 24: *Knit Strands*. Fonte: <  
[http://www.matkovic.com/anto/works\\_knit\\_strand\\_09june2012.html](http://www.matkovic.com/anto/works_knit_strand_09june2012.html)> acesso em 30/01/2013.

Também de autoria de Mihail Dzhurev, este *compound* baseia-se na geração de *strands* que se mesclam de acordo com padrões de trançado pré-definidos pelo usuário. Ocorre ainda a extrusão de uma malha poligonal ao longo destes *strands*, gerando geometria passível de modelagem não procedural, bem como a geração de *bump maps* ou *displacement maps*.

### 3.3.4 Ruffle

Criado por Jonah Friedman com o intuito de simular penas, este *compound* não é baseado em *strands*, mas sim em partículas substituídas por geometria de instância.

O número de penas é determinado por um cálculo que leva em conta um parâmetro estabelecido pelo usuário, a área da malha poligonal a partir da qual as partículas são emitidas, e os atributos das *bounding boxes* das partículas instanciadas.

A orientação, bem como a curvatura das penas, é calculada com base em “penas guia”, invisíveis quando renderizadas, distribuídas pelo usuário a seu critério ao longo da malha emissora. Para tal cálculo, “*Ruffle*” encontra a “pena guia” mais próxima de cada partícula, observa sua orientação e curvatura, assim como das “penas guia” circundantes. Munido destes dados, o *compound* calcula para cada partícula a orientação e a curvatura a serem atribuídas a cada amostra de geometria de instância.

## Considerações finais

Concluída esta pesquisa, observa-se que a automação de procedimentos aliada à síntese de geometria se mostra favorável à produção de filmes de animação 3D digital, quando há, nestes, elementos dotados de autossimilaridade, replicados de acordo com determinado padrão, ou com modificações parciais.

Para melhor aproveitamento das técnicas de automação de procedimento, é importante que haja um planejamento prévio, e por isto, foram listados de maneira abreviada alguns padrões, bem como autores cujas obras, se estudadas, podem auxiliar neste planejamento.

É este, no entanto, um trabalho de caráter introdutório, que esperamos sirva de entrada a aqueles que pelo tema se interessem. Ressalta-se ainda que existe a possibilidade de se realizar síntese de geometria por meio de automação de procedimentos valendo-se de diferentes processos, que não o uso de *logic nodes*. Esperamos que a leitura deste trabalho propicie ao leitor maior entendimento a respeito da modelagem procedural, de forma que possa mais rapidamente inseri-la em sua rotina de trabalho.

## Glossário

**Algoritmo:** Uma sequencia finita de instruções.

**Ângulo:** Região de um plano delimitada pela interseção de duas semirretas.

**Aresta:** Segmento de reta que demarca a intercessão de dois planos

**Booleano:** Valor binário significando verdadeiro ou falso.

**Bounding Box:** Quando aplicado a técnica 3d digital Caixa que abrange o volume de um modelo tridimensional digital.

**Bump Map:** Imagem utilizada com intuito de alterar as normais de uma malha poligonal a fim de simular relevo sem que seja alterada sua geometria.

**Compound:** Arranjo de logic nodes que compõe um algoritmo.

**Coordenadas UV:** Coordenadas de um modelo tridimensional virtual planificado.

**CPU:** *Central Processing Unit*, que traduzindo- se para o português é unidade central de processamento. É a parte do hardware de um computador popularmente conhecida como processador. A CPU é responsável pelos cálculos realizados por um computador.

**Displacement Map:** Imagem utilizada para alterar uma malha poligonal a partir da conversão de informação de cores RGB em dados escalares. Quando renderizada cada face é movida na direção de sua normal de acordo com o valor escalar obtido a partir da imagem.

**Estrutura Rizomática:** Estrutura ramificada

**Extrusão:** A expressão extrusão tem origem na industria metalúrgica, onde há diferentes formas de extrusão de perfis metálicos, sendo que nesta a definição de extrusão é o forçamento de um material através de de uma matriz a fim que este ao final possua dada seção transversal.

Na modelagem em 3d digital a extrusão se refere a criação de uma cópia e uma dada seção transversal.

**Geometry instancing:** traduzido como geometria de instância, e em 3d digital se refere a geometrias repetidas de acordo com padrão estabelecido a fim de substituir partículas ou outras geometrias.

**GPU:** *Graphics Processing Unite*, que traduzindo- se para o português é Unidade Gráfica de Processamento. É a parte do hardware de um computador popularmente conhecida como placa de vídeo. A GPU pode auxiliar ou ser responsável pelo calculo das representações visuais a serem realizadas por computador.

**Hard Surfaces Modeling:** Representação em 3d digital de modelo dotado de arestas salientes e impassível de deformação.

**Logic Nodes:** traduzido como verticilos lógicos. O termo verticilo é utilizado na biologia para definir um ponto em um caule onde insere- se outro órgão. Verticilos são popularmente conhecidos como nós.

São *Logic Nodes* os elementos básicos da programação visual.

**Mapa de Influência:** tradução nossa para o termo inglês *weight map*.

**NURBS:** *Non Uniform Rational Basis Spiline*, que pode ser traduzido para o português como Spiline de base racional não uniforme. É um modelo matemático utilizado na representação de superfícies em 3d virtual

**Organic modeling:** Representação em 3d digital de modelo dotado de poucas ou nenhuma aresta saliente e passível de deformação.

**Point Cloud:** um conjunto de Vértices em um sistema de coordenadas tridimensional.

**Polígono:** Figura plana delimitada por segmentos de retas.

**Renderização:** quando aplicado a técnicas de modelagem animação ou texturização de modelos criados em 3d digital, é o calculo da imagem final a ser exibida.

**Script:** Arquivos de texto criados a partir de linguagem de programação com função de controlar uma ou mais ações de um Software.

**Spin:** Do verbo da língua inglesa “Spin” que pode ser traduzido como girar. Na modelagem 3d digital se refere usualmente a extrusão de meia seção transversal de um modelo ao redor de um eixo.

**Topologia:** Disposição lógica de elementos.

**Vértice:** ponto de interseção de segmentos de retas.

**Weigth map:** traduzido como mapa de influência, onde cores são utilizadas para modular intensidade.

## Referências Bibliográficas

BORGES, Jorge Luis. *Outras inquisições*. trad. Davi Arrigucci Jr., São Paulo: Companhia das Letras, 2007.

BORGES, Jorge Luis. *Obras completas*. São Paulo: Globo, 1998. 708 p.

EBERT, David S.; MUSGRAVE, F. Kenton; PEACHEY Darwyn; PERLIN Ken; WORLEY, Steven. *Texturing & Modeling: a procedural approach*. Estados Unidos: Elsevier Science, 2003.

LINDENMAYER, Aristid; PRUSINKIEWICZ, Przemyslaw. *The algorithmic beauty of plants*. New York: Springer – Verlag, 1996.

MINSKY, Marvin. *A framework for representing knowledge*. MIT-AI Laboratory Memo 306, June, 1974.

PRESTON, Richard. *The New Yorker*. New York, 1992.

3D World. Londres: editora 3D Worl, n. 145, Agos. 2011.

UNIVERSITY OF SURREY <<http://www.maths.surrey.ac.uk/hosted-sites/R.Knott/Fibonacci/propsOfPhi.html>> acesso em 30/01/2013.

UNIVERSIDADE TÉCNICA DE LISBOA

<<https://dspace.ist.utl.pt/bitstream/2295/1096526/1/>> acesso em 30/01/2013.

BIT A BIT

<<http://bitabitdicas.blogspot.com.br/2009/01/unidade-de-ponto-flutuante-x87-fpu.html>> acesso em 30/01/2013.

MIT MIDIA LAB <<http://web.media.mit.edu/~minsky/papers/Frames/frames.html>> acesso em 30/01/2013.

AUTODESK SOFTIMAGE USER GUIDE

<[http://download.autodesk.com/global/docs/softimage2013/en\\_us/userguide/index.html](http://download.autodesk.com/global/docs/softimage2013/en_us/userguide/index.html)> acesso em 30/01/2013.