

Universidade Federal de Minas Gerais
Instituto de Ciências Exatas
Departamento de Ciência da Computação

**Geração Procedural e Visualização de Terrenos e
Planetas em Tempo-Real**

por

Fábio Markus Nunes Miranda

Proposta de Projeto Orientado em Computação II

Apresentado como proposta de trabalho na disciplina de Projeto
Orientado em Computação II do Curso de Bacharelado em Ciência da
Computação da UFMG

Prof. Dr. Luiz Chaimowicz
Orientador

Carlúcio Cordeiro
Co-Orientador

Assinatura do aluno:

Assinatura do orientador:

Assinatura do co-orientador:

Belo Horizonte – MG

2009 / 1º semestre

1 Introdução

A geração procedural de modelos é uma área da Ciência da Computação que propõe que modelos gráficos tridimensionais (representação em polígonos de algum objeto) possam ser gerados através de rotinas e algoritmos. Tal técnica vem se tornando bastante popular nos últimos tempos, tendo em vista que, com o crescimento da indústria do entretenimento, há uma necessidade de se construir modelos cada vez maiores e com um grande nível de detalhe. A técnica de geração procedural vem então como uma alternativa à utilização do trabalho de artistas e modeladores na criação de modelos tridimensionais.

2 Objetivo

O objetivo deste trabalho é continuar com o desenvolvimento do arcabouço proposto na disciplina Projeto Orientado em Computação I. O trabalho feito anteriormente envolve a parte de visualização dos terrenos e também a sua geração. Foi possível, ao final da disciplina de POC I, ter uma maior familiaridade com o assunto e também visualizar alguns terrenos (Figuras 1 e 2).

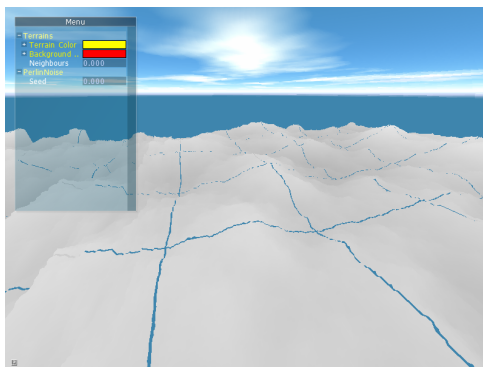


Figura 1: Terreno gerado no POC I.

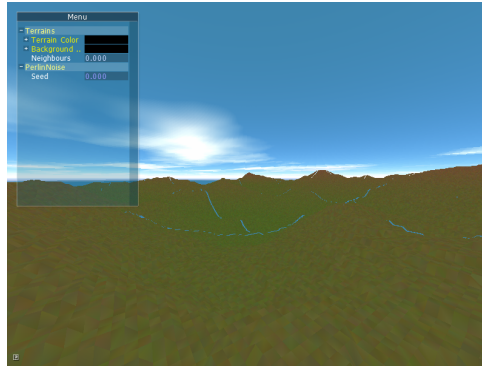


Figura 2: Terreno gerado no POC I.

Os terrenos gerados em POC I possuem uma série de parâmetros para alterar a sua aparência e também permitem uma navegação contínua, ou seja, o usuário pode andar livremente e novos terrenos são gerados, de acordo com a demanda.

A proposta para a disciplina de POC II é adaptar o que já foi implementado e permitir também a geração e visualização de terrenos esféricos, que possam representar planetas. Há uma série de problemas a serem abordados ao longo do trabalho, como a questão de nível de detalhe (LOD), já que os terrenos terão uma escala muito maior.

Não será implementada aqui uma *engine* gráfica completa. Essas disponibilizam para o desenvolvedor diversas funcionalidades, como animação de modelos, exportação de modelos, partículas, tratamento de entrada e saída de controles, tratamento de rede, etc. Como essas funcionalidades fogem do escopo desse trabalho, busca-se implementar uma ferramenta que seja independente da *engine*. O trabalho poderá ser tratado então como um "componente" de um aplicativo gráfico, e assim acoplado a qualquer *engine* gráfica.

3 Referencial Teórico

Vários trabalhos publicados abordam a geração procedural de modelos. Alguns destes trabalhos abordam a geração de cidades ([1] e [2]), outros abordam a geração de terrenos realistas (em tempo-real, como os trabalhos [3] e [4], ou não, como o *MojoWorld*[5]), ou então a geração de árvores [6]. A principal referência na área é o livro *Texturing and Modeling: A Procedural Approach* [7], em que é explicada a geração procedural de diversos tipos de modelos.

Algumas técnicas largamente utilizadas na geração procedural são: Sistemas de Lindenmayer (*l-System*)[8], que, através de uma gramática, modela o crescimento de plantas; geometrias fractais [9]; e também ruído de Perlin (*Perlin Noise*[10]).

A *engine* gráfica *I-Novae*[11] apresenta uma proposta semelhante ao deste trabalho. Porém, a *I-Novae* é uma solução independente, que se propõe a oferecer uma série de funcionalidades esperadas de uma *engine*. Como este trabalho não propõe uma *engine*, ele se torna muito mais flexível, podendo ser integrado a qualquer outra *engine* já existente.

4 Metodologia

4.1 Tipo de Pesquisa

O trabalho proposto é uma pesquisa de natureza aplicada, pois busca aplicar um conhecimento teórico (técnicas de geração procedural) e obter um resultado prático na forma de uma ferramenta e tem um objetivo exploratório. A pesquisa se dá em laboratório, pois se trata de um ambiente controlado.

4.2 Procedimento Metodológico

O primeiro passo do trabalho será a adaptação do que já foi feito, na disciplina de POC I, para a visualização de terrenos esféricos. Um estudo será feito e, caso necessário, será implementado um algoritmo de nível de detalhe, provavelmente o *Chunked LOD*[12]. A seguir, o trabalho será acoplado a uma *engine* gráfica.

O último passo será a adição de efeitos visuais, como textura, sombra e efeitos atmosféricos.

A lista abaixo apresenta os passos já abordados na disciplina de POC I:

- Etapa 1: Estudo das soluções já existentes.
- Etapa 2: Início do desenvolvimento.

- Etapa 3: Geração de terrenos planos com base em mapas de altura.
- Etapa 4: Geração de terrenos planos proceduralmente.

As quatro etapas foram concluídas. Dessa forma, as seguintes etapas serão abordadas ao longo da disciplina de POC II:

- Etapa 1: Adaptação para a visualização de terrenos esféricos - O arcabouço desenvolvido em POC I será modificado para que seja possível mapear as posições do terreno em uma esfera.
- Etapa 2: Implementação de um algoritmo de nível de detalhe - Como a complexidade do terreno tende a aumentar (maior número de polígonos), poderá ser necessário a implementação de um algoritmo de nível de detalhe.
- Etapa 3: Acoplamento a uma *engine* gráfica.
- Etapa 4: Adição de efeitos visuais.

4.3 Cronograma

Baseado nas etapas descritas anteriormente, o seguinte cronograma foi montado:

	MAR				ABR				MAI				JUN			
Etapa 1																
Etapa 2																
Etapa 3																
Etapa 4																

Figura 3: Cronograma para a disciplina de POC II.

5 Resultados Esperados

Ao final do trabalho, espera-se que seja possível:

- Visualização de um planeta em tempo-real e gerado proceduralmente.
- Acoplamento a uma *engine* gráfica.

Referências

- 1 Yoav I H Parish and Pascal Müller. Procedural modelling of cities. In *in Proc. ACM SIGGRAPH, (Los Angeles, 2001) ACM Press*, pages 301–308.
- 2 Stefan Greuter, Jeremy Parker, Nigel Stewart, and Geoff Leach. Real-time procedural generation of ‘pseudo infinite’ cities. In *GRAPHITE '03: Proceedings of the 1st international conference on Computer graphics and interactive techniques in Australasia and South East Asia*, pages 87–ff, New York, NY, USA, 2003. ACM.
- 3 Jacob Olsen. Realtime procedural terrain generation. In *Department of Mathematics And Computer Science (IMADA).*, 2004.
- 4 Lukas Zimmerli and Paul Verschure. Delivering environmental presence through procedural virtual environments. In *PRESENCE 2007, The 10th Annual International Workshop on Presence*, 2007.
- 5 Mojoworld generator. Disponível em: <http://www.mojoworld.org/>. Acessado em: 26 mar. 2009.
- 6 Speedtree | idv, inc. Disponível em: <http://www.speedtree.com/>. Acessado em: 26 mar. 2009.
- 7 David S. Ebert, F. Kenton Musgrave, Darwyn Peachey, Ken Perlin, and Steven Worley. *Texturing and Modeling: A Procedural Approach*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2002.
- 8 Przemyslaw Prusinkiewicz and Aristid Lindenmayer. *The algorithmic beauty of plants*. Springer-Verlag New York, Inc., New York, NY, USA, 1996.
- 9 Benoit B. Mandelbrot. *The Fractal Geometry of Nature*. W. H. Freeman, August 1982.
- 10 Ken perlin’s homepage. Disponível em: <http://mrl.nyu.edu/~perlin/>. Acessado em: 26 mar. 2009.
- 11 Infinity. Disponível em: <http://www.infinity-universe.com/Infinity/>. Acessado em: 26 mar. 2009.

12 Chunked lod. Disponível em: <http://tulrich.com/geekstuff/chunklod.html>.
Acessado em: 26 mar. 2009.