

Geração Procedural de Grandes Terrenos em Tempo-Real

Fábio Markus Nunes Miranda
Orientador: Prof. Luiz Chaimowicz
Co-Orientador: Carlúcio Cordeiro

Departamento de Ciência da Computação
Universidade Federal de Minas Gerais

Apresentação parcial - POC II

Sumário

1 Motivação

2 Metodologia

3 Proposta

4 Cronograma

5 Bibliografia

Motivação

- Atualmente, há uma necessidade de se criar modelos 3D cada vez maiores e com grande nível de detalhe.
- Porém, quanto maior e mais detalhado o modelo, mais tempo terá que ser gasto por um modelador para fazê-lo.

O que é geração procedural?

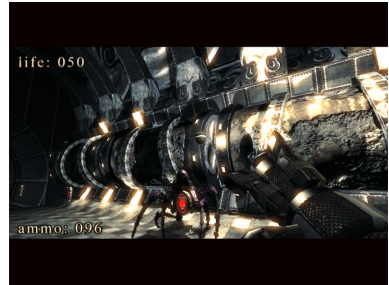
- Geração procedural é um termo genérico para descrever algoritmos que determinam características de efeitos ou modelos.
- Há diversos tipos de técnicas e algoritmos, cada um aplicado a uma determinada área:
 - L-System: geração de árvores e cidades.
 - Fractais e Perlin Noise: geração de terrenos e texturas

Vantagens da geração procedural

- Flexibilidade: alterando os parâmetros do algoritmo, é possível gerar um grande número de modelos.
- Espaço: não há necessidade de um grande espaço em disco, já que tudo será ditado por algoritmos.

Exemplos

- **.kkrieger**
Praticamente tudo gerado
proceduralmente
- Elite (1984)
- SpeedTree



Exemplos

- .kkrieger
- Elite (1984)
Oito galáxias, 256 planetas.
- SpeedTree



Exemplos

- .kkrieger
- Elite (1984)
- **SpeedTree**
Árvores geradas proceduralmente.



Sumário

1 Motivação

2 Metodologia

3 Proposta

4 Cronograma

5 Bibliografia

Metodologia

- Livro *Texturing and Modeling: A Procedural Approach*.
- Estudo das melhores formas de reduzir o gasto com memória através de estruturas de dados do OpenGL.
- Implementação do arcabouço.

Sumário

1 Motivação

2 Metodologia

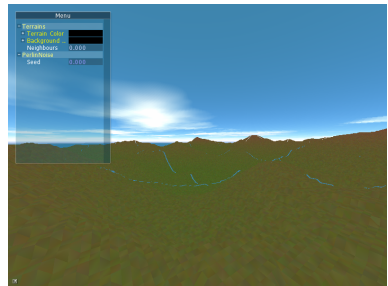
3 Proposta

4 Cronograma

5 Bibliografia

Trabalho anterior

- Ao final de POC I, foi possível gerar terrenos na CPU.
- O terreno era dividido em quadrados, que eram gerados proceduralmente a medida que era necessário.

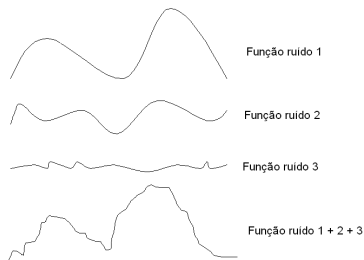


Proposta

- O objetivo deste trabalho é expandir o que foi implementado em POC I, dessa vez para terrenos maiores.
- Para isso, buscou-se transferir a geração dos terrenos da CPU para a GPU (placas de vídeo).

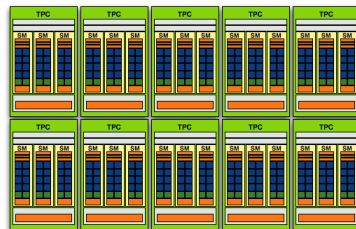
Ruído de Perlin

- O ruído é usado para simular estruturas naturais, como núvens, texturas de árvores, e terrenos.



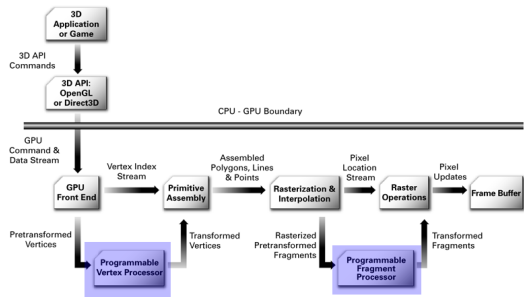
Geração na GPU

- A natureza facilmente paralelizável das aplicações gráficas fez com que as GPUs fossem desenvolvidas com um número muito maior de unidades de processamento do que as CPUs.
- Como o terreno vai ser gerado (e desenhado) na GPU, não haverá perda na transferência de dados do terreno entre CPU e GPU.



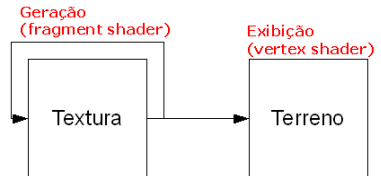
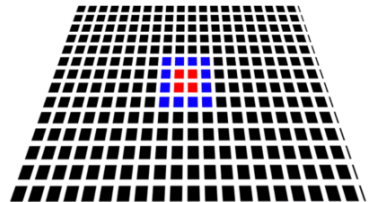
Geração na GPU

- A geração será feita através de conjuntos de instruções (shaders) executados na GPU.
- O *pipeline* gráfico das GPUs executa dois tipos de shaders:
 - Vertex shader: executado para cada vértice da cena.
 - Fragment shader: executado para cada fragmento dos polígonos (a saída do shader é uma cor, salva em uma textura).



Geração na GPU

- Fragment shader: calcular o novo terreno procedural e salva o resultado em uma textura.
- Vertex shader: após o calculo do terreno, irá consultar oque foi gerado, e ajustar a altura dos vértices do terreno.
- Para evitar efeitos indesejáveis do terreno sendo criado a medida que o usuário se move, o terreno é criado em passos, cada um sendo exibido quando está pronto.



Sumário

1 Motivação

2 Metodologia

3 Proposta

4 Cronograma

5 Bibliografia

Planejamento

- Etapa 1: Adaptação para a visualização de grandes terrenos na GPU.
- Etapa 2: Implementação de um algoritmo de nível de detalhe.
- Etapa 3: Acoplamento a uma *engine* gráfica.
- Etapa 4: Adição de efeitos visuais.

Sumário

1 Motivação

2 Metodologia

3 Proposta

4 Cronograma

5 Bibliografia



Benoit B. Mandelbrot.
The Fractal Geometry of Nature.
W. H. Freeman, August 1982.



Przemyslaw Prusinkiewicz and Aristid Lindenmayer.
The algorithmic beauty of plants.
Springer-Verlag New York, Inc., New York, NY, USA, 1996.



[geo-spatial data acquisition home.](#)
Disponível em: <http://emrl.byu.edu/gsda/>. Acessado em: 23 nov. 2008.



[Pixar animation studios.](#)
Disponível em: <http://www.pixar.com/>. Acessado em: 23 nov. 2008.



[Ian bell's elite pages.](#)
Disponível em: <http://www.iancgbell.clara.net/elite/>. Acessado em: 23 nov. 2008.



[Procedural content generation.](#)
Disponível em: <http://lukehalliwell.wordpress.com/2008/08/05/procedural-content-generation/>.
Acessado em: 23 nov. 2008.



[Acmc projects ,cg rendering of coral at the university of queensland.](#)
Disponível em: http://www.acmc.uq.edu.au/Projects/CG_Rendering.html. Acessado em: 23 nov. 2008.



[Object oriented framework development.](#)
Disponível em: <http://www.acm.org/crossroads/xrds7-4/frameworks.html>. Acessado em: 23 nov. 2008.



[Glfw - an opengl framework.](#)
Disponível em: <http://glfw.sourceforge.net/>. Acessado em: 23 nov. 2008.



[Anttweakbar gui library to tweak parameters of opengl and directx applications.](#)

Disponível em: <http://www.antisphere.com/Wiki/tools:anttweakbar>. Acessado em: 23 nov. 2008.



[Devil - a full featured cross-platform image library.](#)

Disponível em: <http://openil.sourceforge.net/>. Acessado em: 23 nov. 2008.



[Gamedev.net - 'slope lighting' terrain.](#)

Disponível em: <http://www.gamedev.net/reference/articles/article1436.asp>. Acessado em: 23 nov. 2008.



[Stefan Greuter and Jeremy Parker.](#)

Undiscovered worlds - towards a framework for real-time.

In In Proc. of the Fifth Intern. Digital Arts and Culture Conference. Press, 2003.



[Yoav I H Parish and Pascal Müller.](#)

Procedural modelling of cities.

In in Proc. ACM SIGGRAPH, (Los Angeles, 2001) ACM Press, pages 301–308, 2001.



[George Kelly and Hugh McCabe.](#)

Citygen: An interactive system for procedural city generation.

In Game Design & Technology Workshop, 2006.



[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.



[Jacob Olsen.](#)

Realtime procedural terrain generation.

In Department of Mathematics And Computer Science (IMADA)., 2004.



[Lukas Zimmerli and Paul Verschure.](#)

Delivering environmental presence through procedural virtual environments.

In *PRESENCE 2007, The 10th Annual International Workshop on Presence*, 2007.



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.



Farès Belhadj.

Terrain modeling: a constrained fractal model.

In *AFRIGRAPH '07: Proceedings of the 5th international conference on Computer graphics, virtual reality, visualisation and interaction in Africa*, pages 197–204, New York, NY, USA, 2007. ACM.



Speedtree — idv, inc.

Disponível em: <http://www.speedtree.com/>. Acessado em: 23 nov. 2008.



Mojoworld generator.

Disponível em: <http://www.mojoworld.org/>. Acessado em: 23 nov. 2008.



Infinity.

Disponível em: <http://www.infinity-universe.com/Infinity/>. Acessado em: 23 nov. 2008.



The official spore and spore creature creator site.

Disponível em: <http://www.spore.com/>. Acessado em: 23 nov. 2008.



Ken perlin's homepage.

Disponível em: <http://mrl.nyu.edu/~perlin/>. Acessado em: 23 nov. 2008.

Dúvidas?