

Geração Procedural de Terrenos para Jogos

João Carlos Becker

26 de Janeiro de 2019

Contextualização

- Criar conteúdo para jogos manualmente exige muito tempo de trabalho
- Conteúdo criado manualmente exige persistência do próprio conteúdo

Contextualização



Figura 1: Editor de mapas do Warcraft III

it.wikipedia.org/wiki/World_Editor_di_Warcraft_III

Contextualização

- Conteúdo criado proceduralmente exige persistência do método (Algoritmo)
- Quanto maior o mundo virtual, tecnicamente mais tempo os jogadores irão explorar [Bevilacqua, 2009]

Contextualização



Figura 2: Mapa de *Far Cry 5*, [Carrier, 2018]

Contextualização



Figura 3: Mundo de **Minecraft**

Objetivo

- Mostrar um método capaz de gerar um terreno para jogos
- Terrenos com relevos **naturais**

Exemplo



Figura 4: Erosão Fractal

imagem de paulbourke.net/fractals/googleearth/

1. Requisitos
2. Técnicas
3. Contribuições da UFFS
4. Jogos com Conteúdo Procedural

Requisitos

Requisitos

- Determinismo, $y = f(x, z)$
- Aleatoriedad

Requisitos

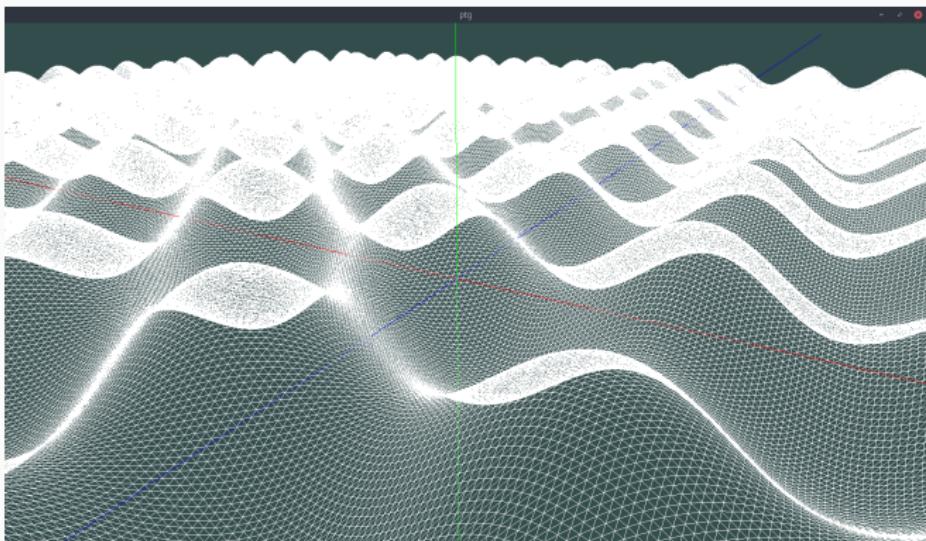


Figura 5: $y = \sin(x) + \sin(z)$

Técnicas

Geração de Relevo

- Ruído de Perlin

Geração de Relevo

- Ruído de Perlin
- Divisões estocásticas

Geração de Relevo

- Ruído de Perlin
- Divisões estocásticas
- Falhas geológicas

Geração de Relevo

- Ruído de Perlin
- Divisões estocásticas
- Falhas geológicas
- Deposição de sedimentos

Geração de Relevo

- Ruído de Perlin
- Divisões estocásticas
- Falhas geológicas
- Deposição de sedimentos
- Disposição do ponto médio

Geração de Relevo

- Ruído de Perlin
- Divisões estocásticas
- Falhas geológicas
- Deposição de sedimentos
- Disposição do ponto médio

Definição

- Ponto: $\textit{Point} \in \{\mathbb{R}^3 \vee \mathbb{R}^2 \vee \mathbb{R}\}$
- $-1 \leq \textit{noise}(\textit{Point}) \leq 1$

Noise

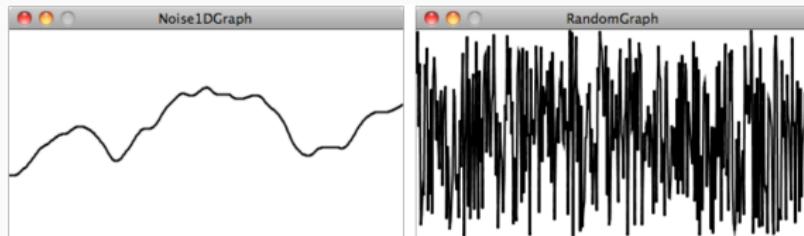


Figura 6: Noise vs Random, por [Shiffman, 2012].

Noise

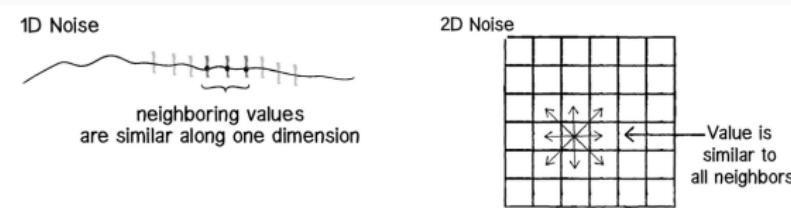


Figura 7: 1d and 2d Noise, por [Shiffman, 2012].

- Complexidade para n dimensões: $\mathcal{O}(2^n)$

- Quantidade de oitavas: $\theta \in \mathbb{N}$

Definição

$$\text{perlinNoise}(\text{Point}, \theta) = \sum_{t=0}^{t=\theta} \frac{\text{Noise}(\text{Point} \cdot 2^t)}{2^t}$$

Ruído de Perlin

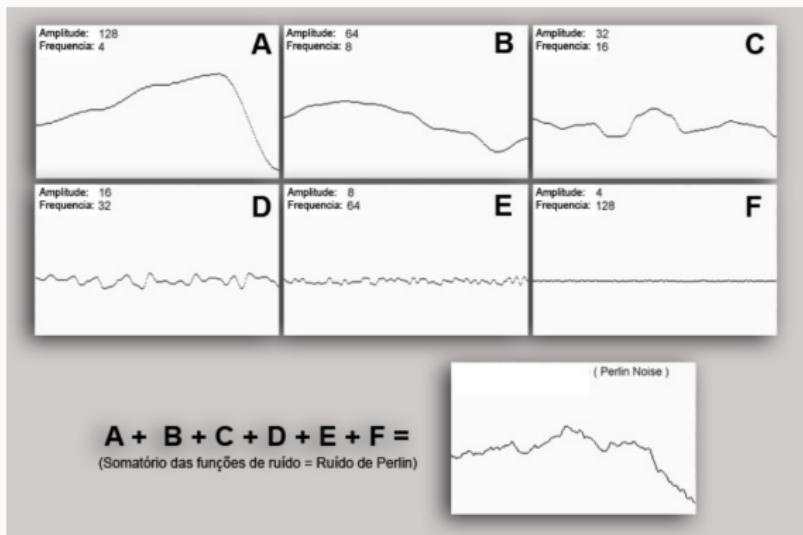


Figura 8: Ruído de Perlin 1D, por [Elias, 2000].

Ruído de Perlin

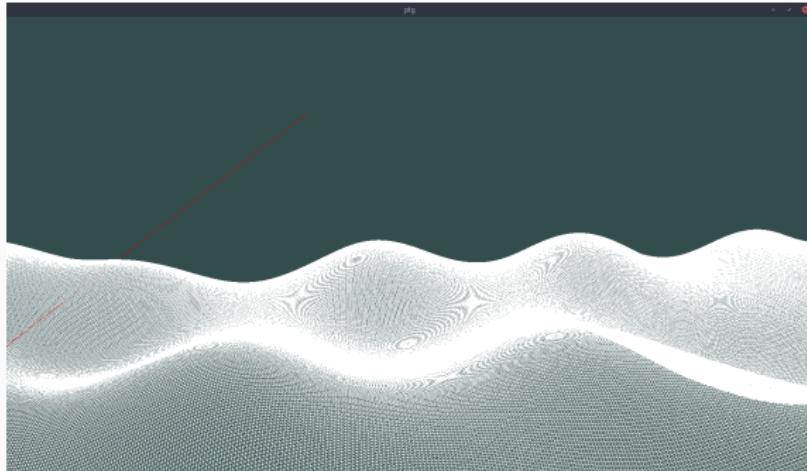


Figura 9: $\theta = 1$.

Ruído de Perlin

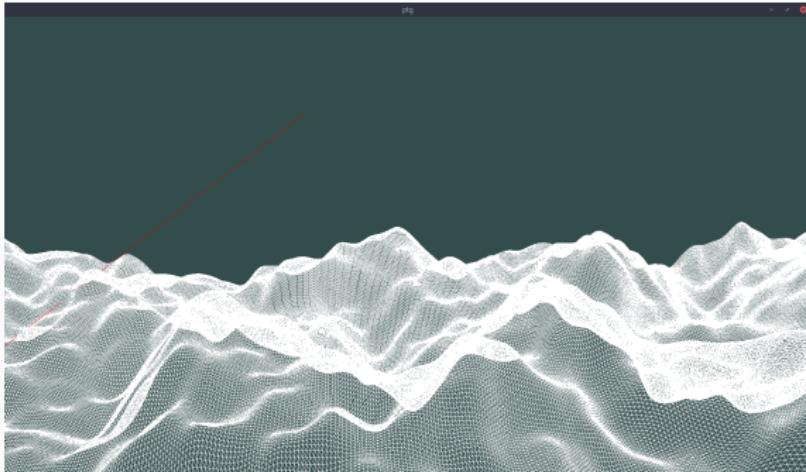


Figura 10: $\theta = 4$.

Ruído de Perlin

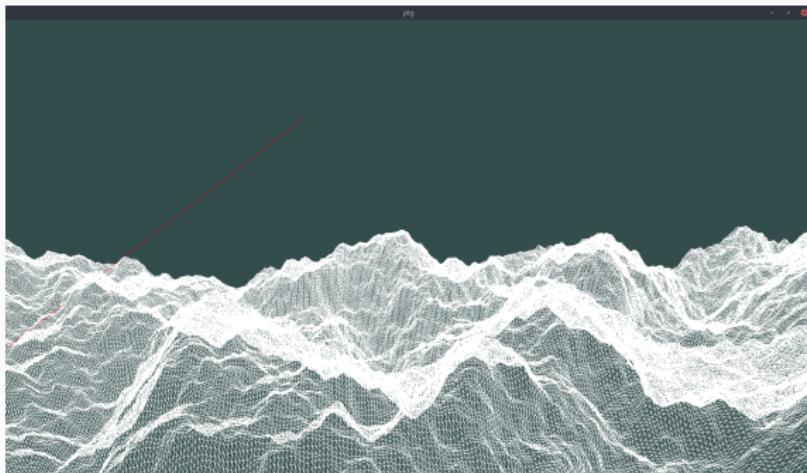


Figura 11: $\theta = 16$.

Caso de borda

$$\sum_{t=0}^{t=\theta} \frac{\text{Noise}(\text{Point} \cdot 2^t)}{2^t}$$

$$\text{Noise}(\text{Point}) = 1$$

$$\sum_{t=0}^{t=\theta} \frac{1}{2^t} = 2^{-\theta}(2^{\theta+1} - 1)$$

Ruído de Perlin

$$\max = \lim_{\theta \rightarrow \infty} 2^{-\theta}(2^{\theta+1} - 1)$$

Ruído de Perlin

$$\begin{aligned} \max &= \lim_{\theta \rightarrow \infty} 2^{-\theta}(2^{\theta+1} - 1) \\ &= 2 \end{aligned}$$

Ruído de Perlin



Figura 12: Limitações

Contribuições da UFFS

Geração Procedural de Cânions Baseado em Ruído de Perlin

- A partir de uma classificação de características de cânions;
- As características mais comuns foram selecionadas.

Geração Procedural de Cânions Baseado em Ruído de Perlin

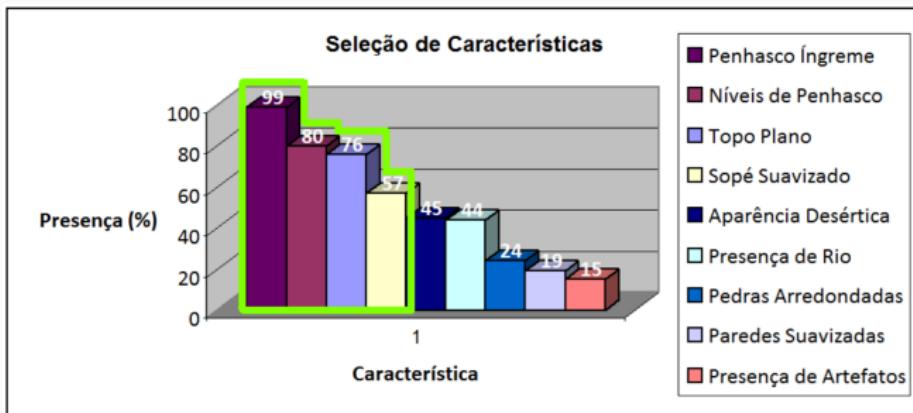


Figura 13: Frequência de características nos cânions [de Souza, 2016].

Geração Procedural de Cânions Baseado em Ruído de Perlin

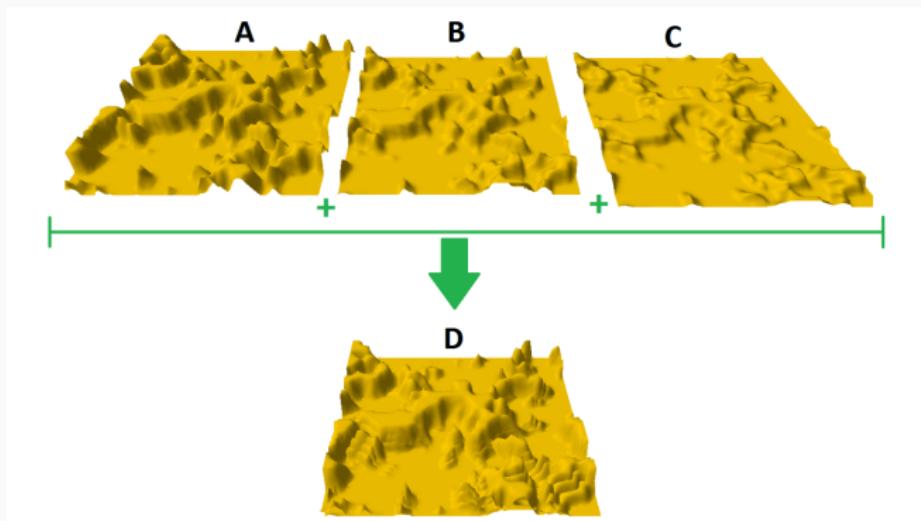


Figura 14: Soma das camadas para gerar Cânions

Geração Procedural de Cânions Baseado em Ruído de Perlin

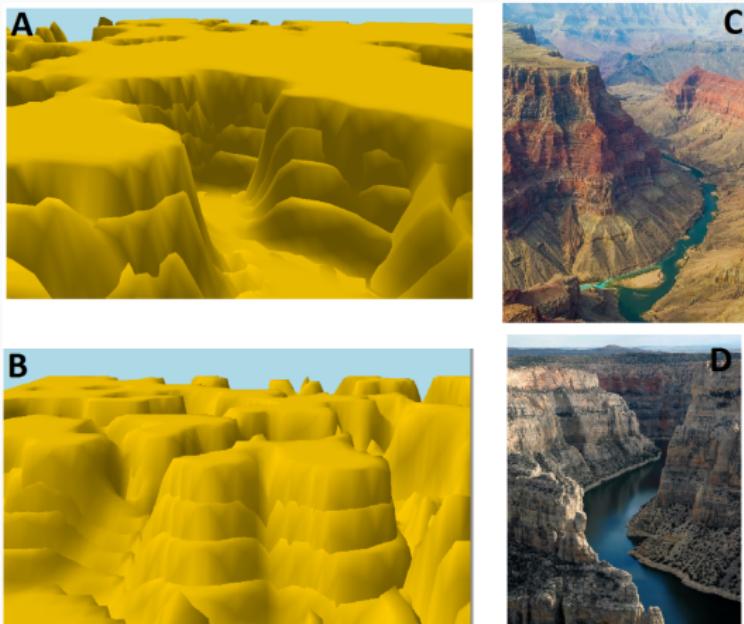


Figura 15: Resultado Final

Geração de Terrenos com Biomas Distintos para Jogos

- Implementação de relevo para 5 Biomas
 - Planícies
 - Montanhas
 - Vales
 - Deserto
 - Cânions
- Distribuição dos Biomas
- Suavização das fronteiras

Geração de Terrenos com Biomas Distintos para Jogos

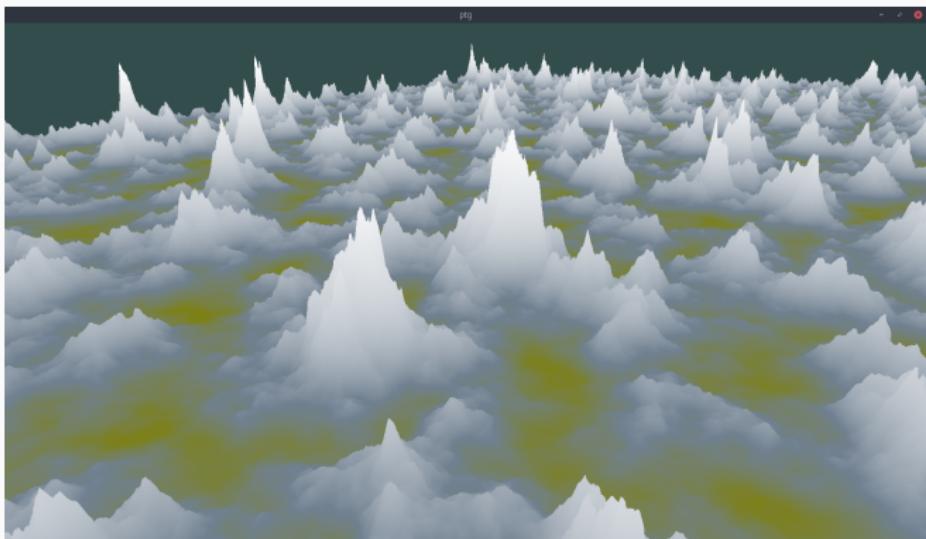


Figura 16: Montanhas, $h = 1.5^{h' \cdot 7} \cdot 5$

Geração de Terrenos com Biomas Distintos para Jogos

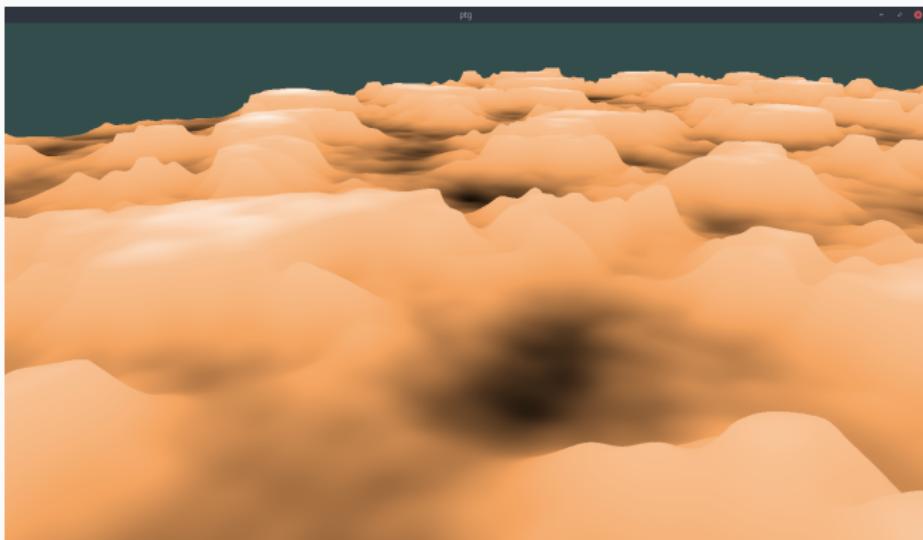


Figura 17: Cânions, $h = (1.5^{clamp(h', -0.3, 0.3) \cdot 14}) \cdot 5 + h' \cdot 8$

Geração de Terrenos com Biomas Distintos para Jogos

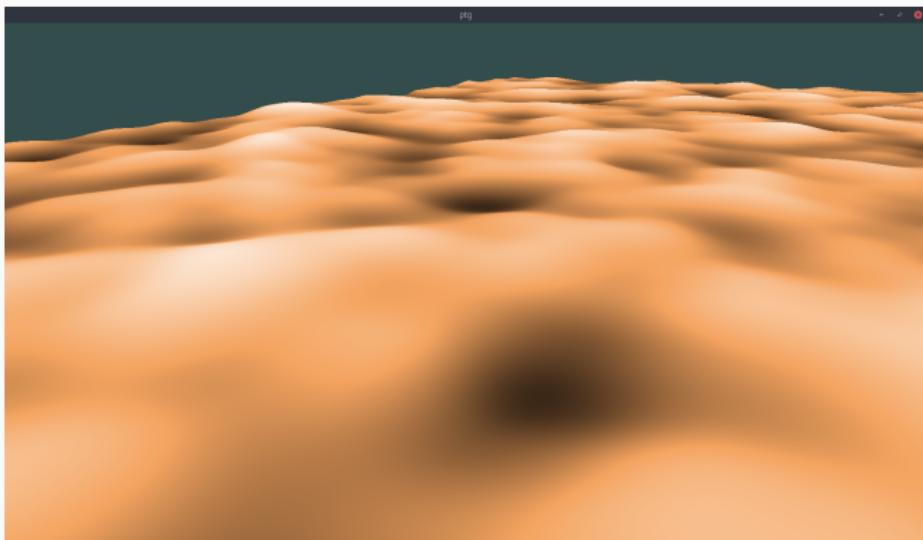


Figura 18: Deserto, $h = h' \cdot 16$

Geração de Terrenos com Biomas Distintos para Jogos

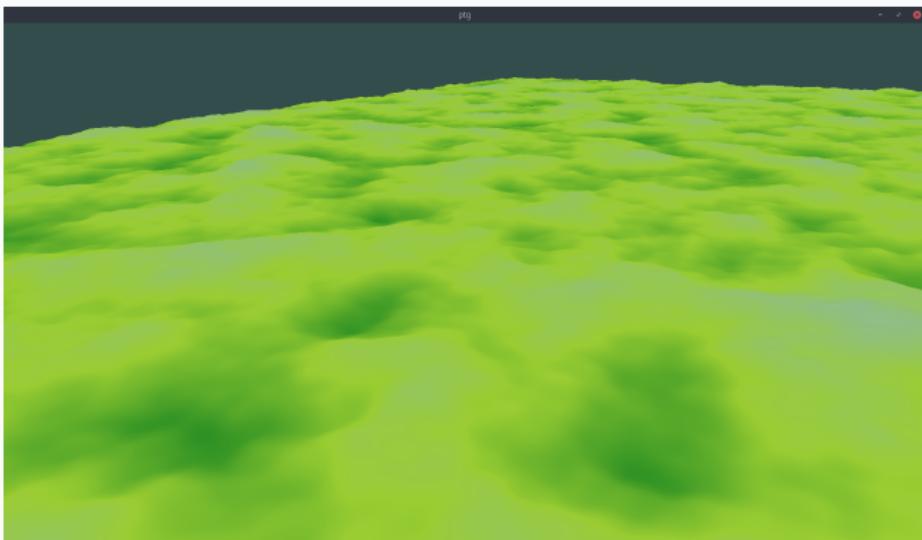


Figura 19: Planícies, $h = h' \cdot 10$

Geração de Terrenos com Biomas Distintos para Jogos

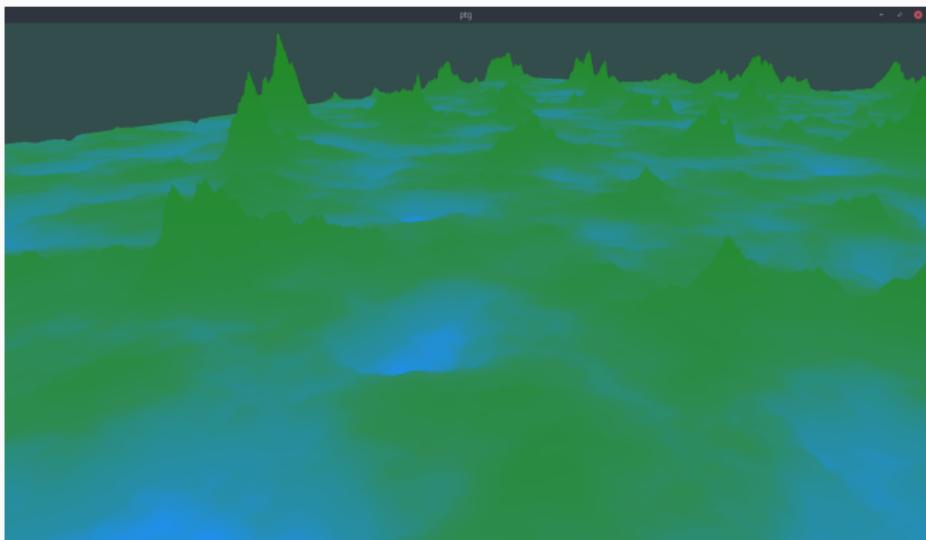


Figura 20: Vales, $h = (\text{clamp}(h', -0.5, 1.0) \cdot 5)^3$

Geração de Terrenos com Biomas Distintos para Jogos



Figura 21: Distribuição de Biomas

Geração de Terrenos com Biomas Distintos para Jogos

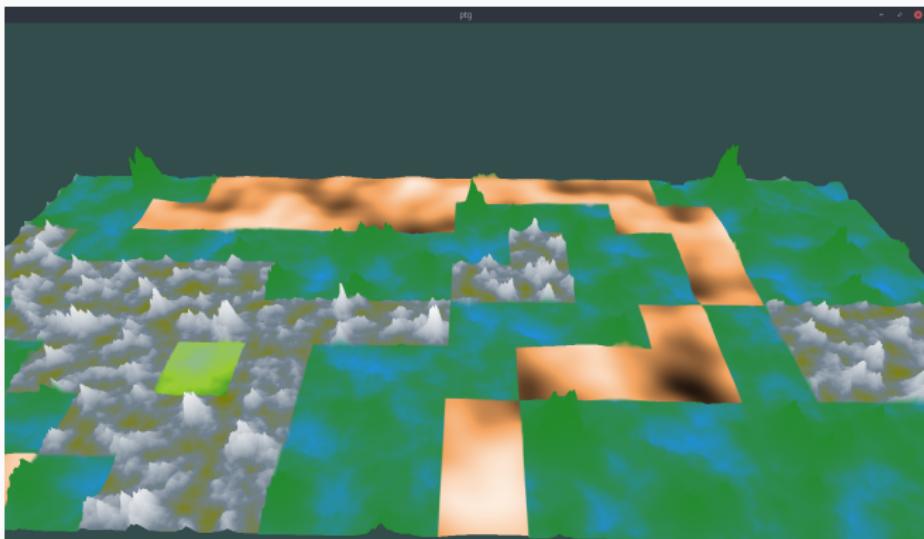


Figura 22: Distribuição de Biomas

Geração de Terrenos com Biomas Distintos para Jogos

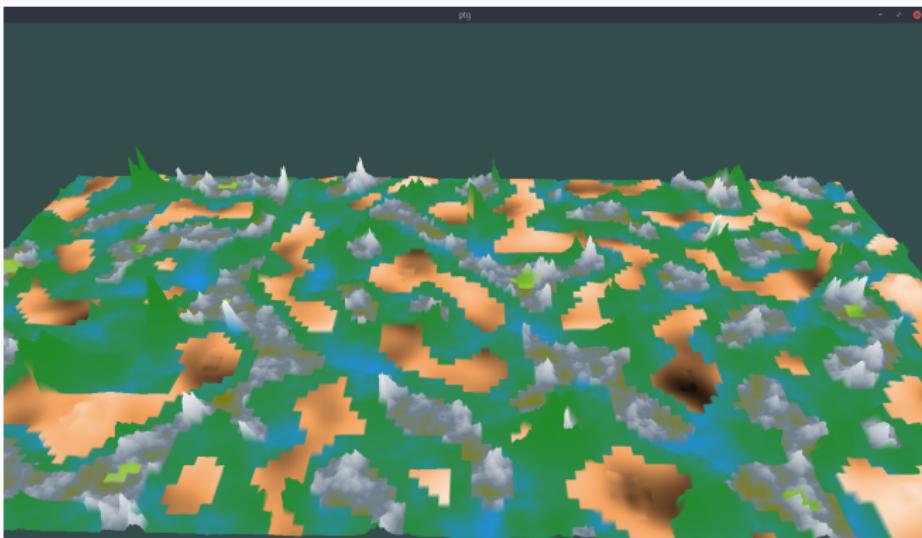


Figura 23: Frequência de Biomas = 8

Geração de Terrenos com Biomas Distintos para Jogos

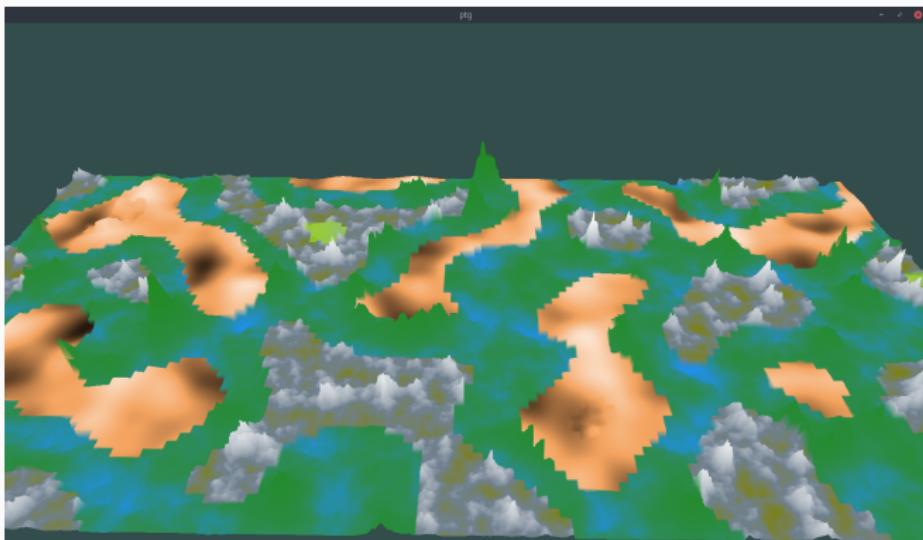


Figura 24: Frequência de Biomas = 16

Geração de Terrenos com Biomas Distintos para Jogos

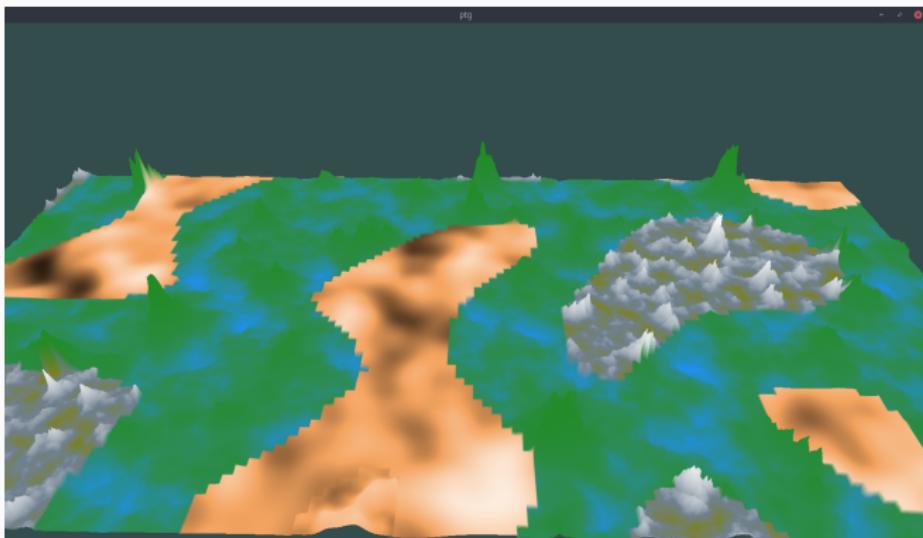


Figura 25: Frequência de Biomas = 32

Geração de Terrenos com Biomas Distintos para Jogos

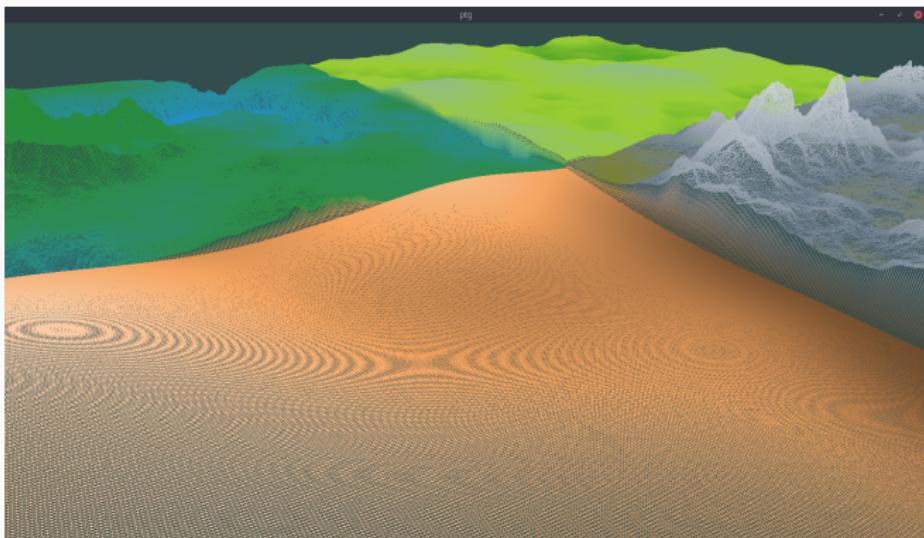


Figura 26: Fronteiras descontínuas

Geração de Terrenos com Biomas Distintos para Jogos

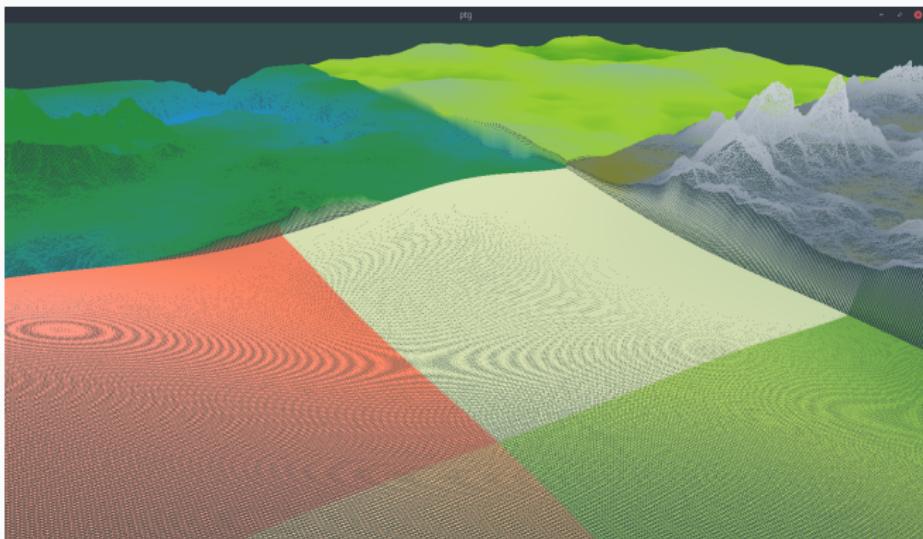


Figura 27: Casos de fronteiras

Geração de Terrenos com Biomas Distintos para Jogos

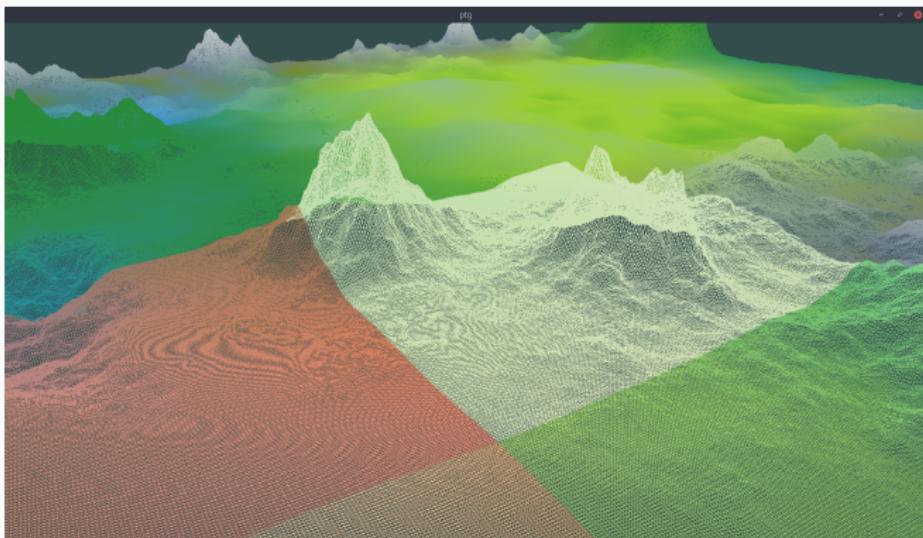


Figura 28: Com interpolação

Geração de Terrenos com Biomas Distintos para Jogos

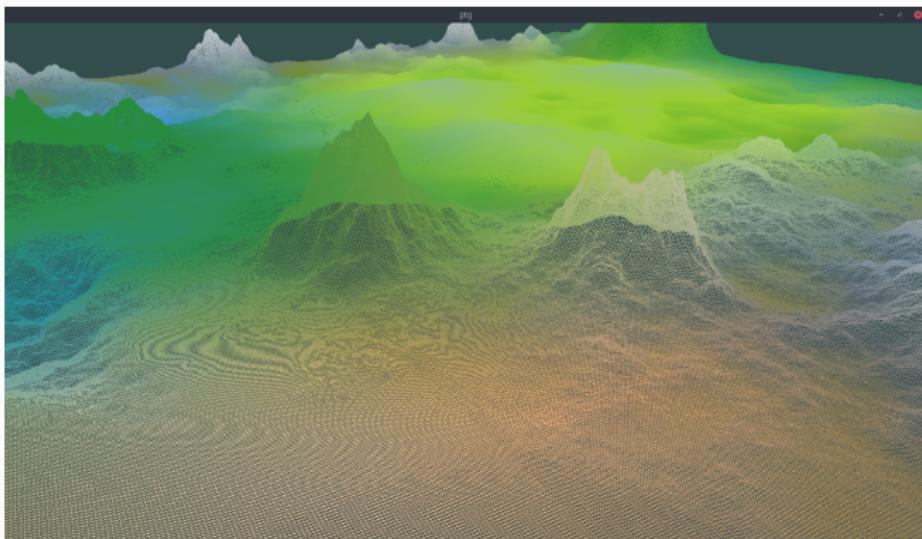


Figura 29: Com interpolação

Gerador de costas



Figura 30: Costa linear.

Gerador de costas

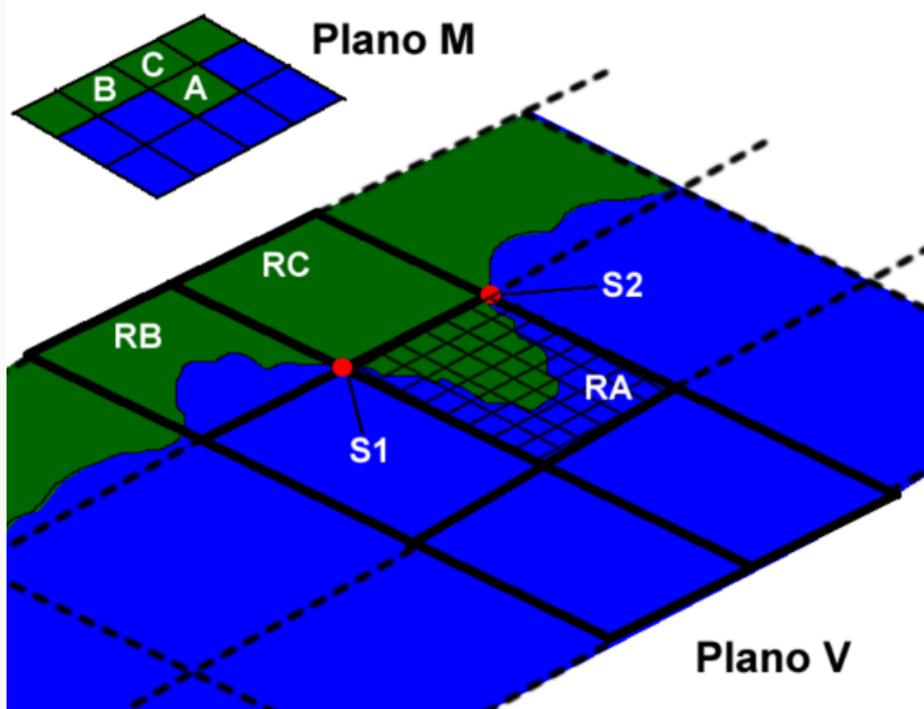


Figura 31: Método.

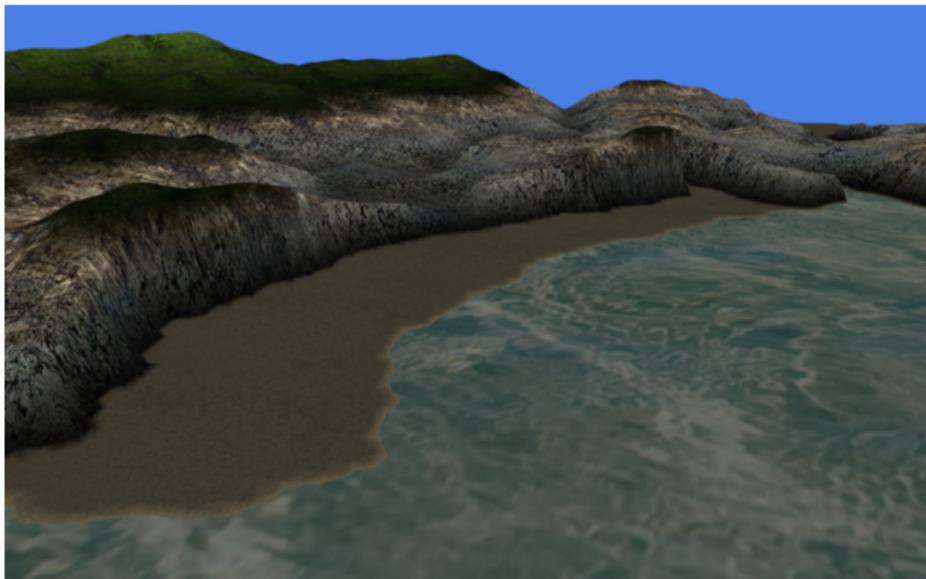


Figura 32: Costa orgânica.

Jogos com Conteúdo Procedural

- Gera planetas esféricos com relevo procedural
- Vegetação Procedural

Spore

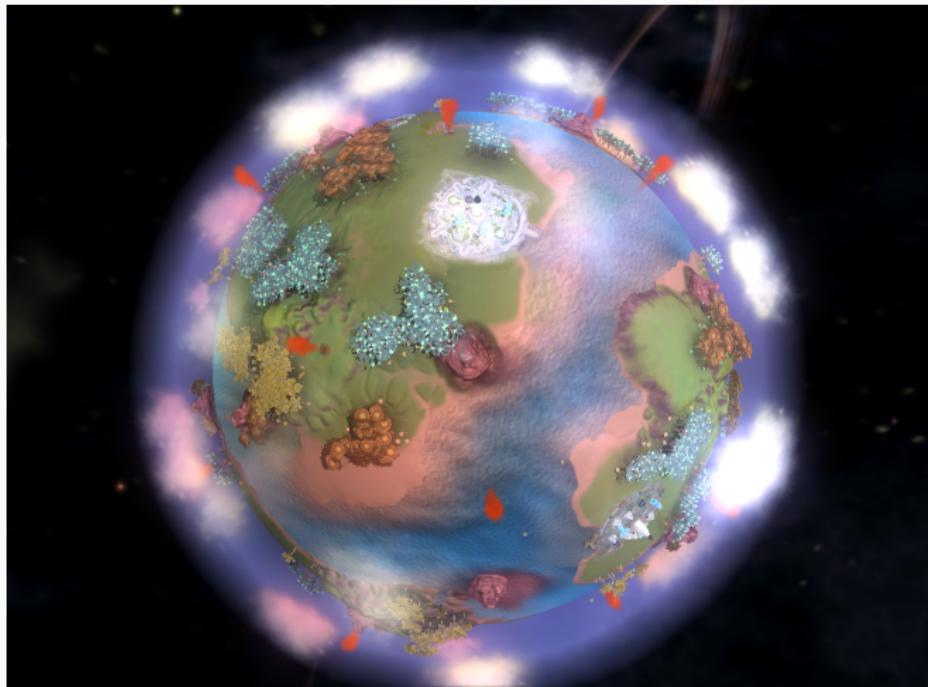


Figura 33: Spore, de spore.fandom.com/wiki/Planet

No man's sky



Figura 34: de wccftech.com/no-mans-sky-xbox-one-release-june-29/

Referências i

-  Bevilacqua, F. (2009).
Ferramenta para geração em tempo real de bordas de mapas virtuais pseudo-infinitos para jogos 3d.
Technical report, Universidade de Santa Maria, Brazil.
-  Carrier, E. (2018).
Farcry 5 procedural world generation.
-  de Souza, G. A. (2016).
Algoritmo de geração procedural de cânions baseado em ruído de perlin.
Technical report, Curso de Ciência da Computação. Universidade Federal da Fronteira Sul, Chapecó.

Referencias ii

-  Elias, H. (2000).
Perlin noise.
from The good-looking textured light-sourced bouncy fun smart and stretchy.
-  Shiffman, D. (2012).
The Nature of Code: Simulating Natural Systems with Processing.
Daniel Shiffman.

Dúvidas?