

Geração de terrenos



Mayara Marques da Rosa

Sumário

- Introdução
- Representação do terreno
- Construção da malha
- Câmera/Observador
- Ruído
- Ruído de Perlin
- Textura
- Resultados
- Referências

Introdução

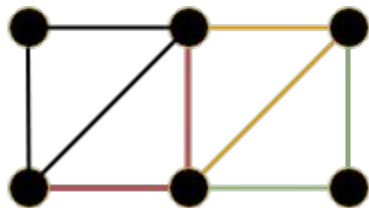
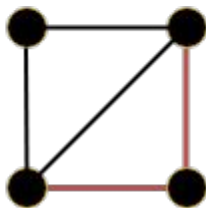
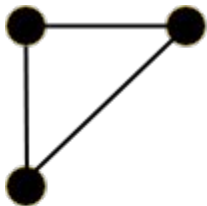
- Aplicações
- Motivação

Representação do terreno

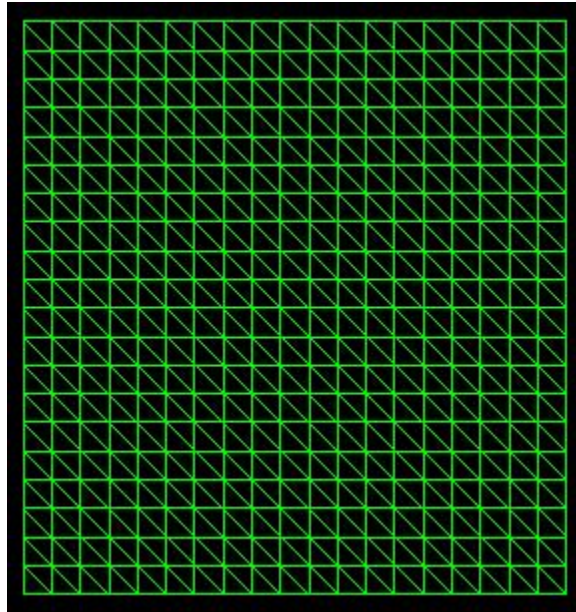
- Malha poligonal

Construção da Malha

- Triangle Strip

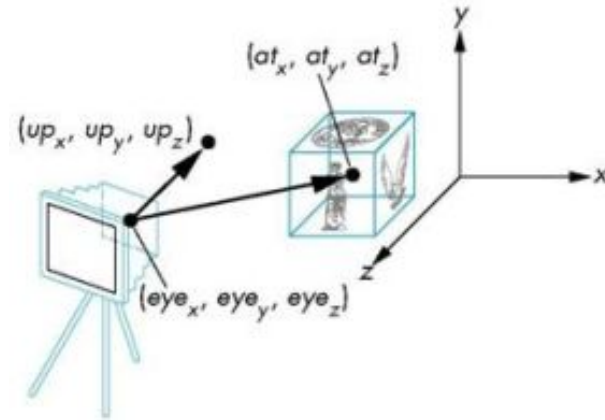
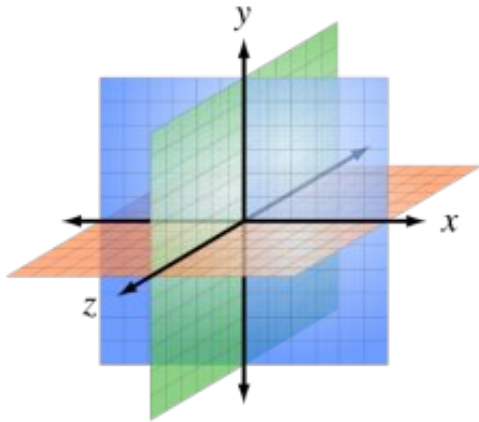


Construção da Malha



Câmera / Observador

- GluLookat



Imagens disponíveis em: <http://www.land-of-kain.de/docs/jogl/>

Câmera / Observador

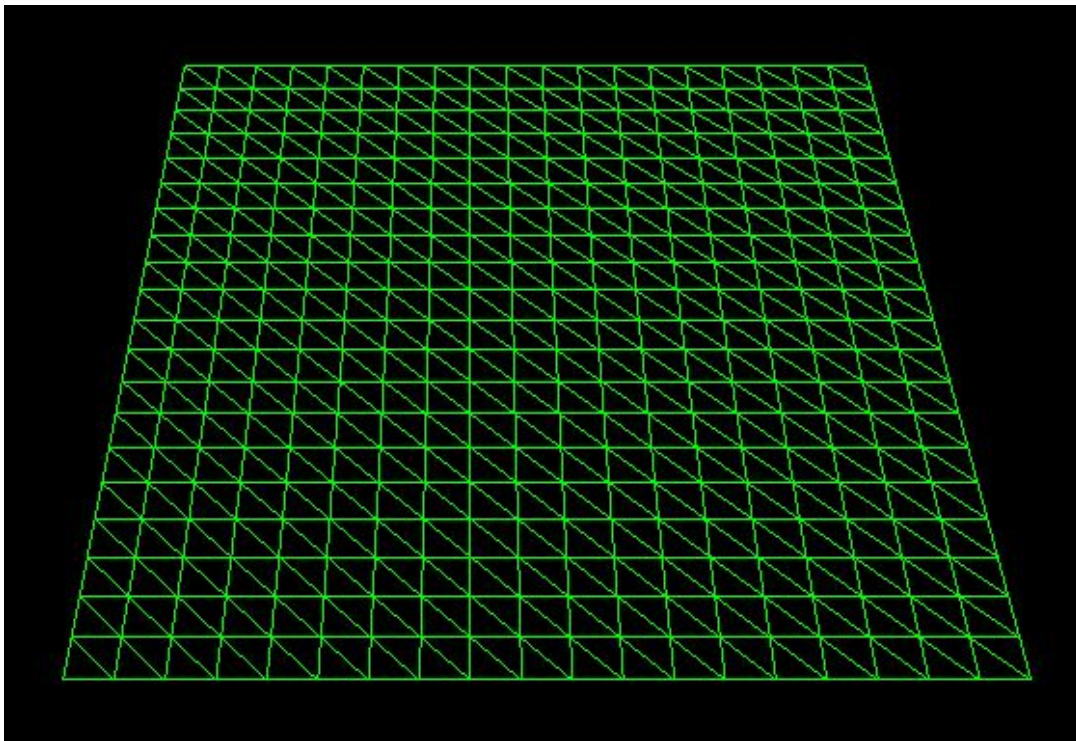


Imagem da malha após reposicionamento da câmera

Ruído

- Utilizar coordenada **Z**
- Definir pontos em **Z** de maneira **pseudo-aleatória** em um intervalo

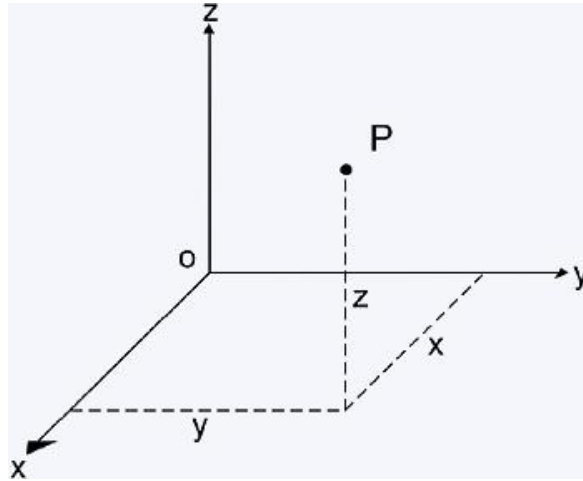
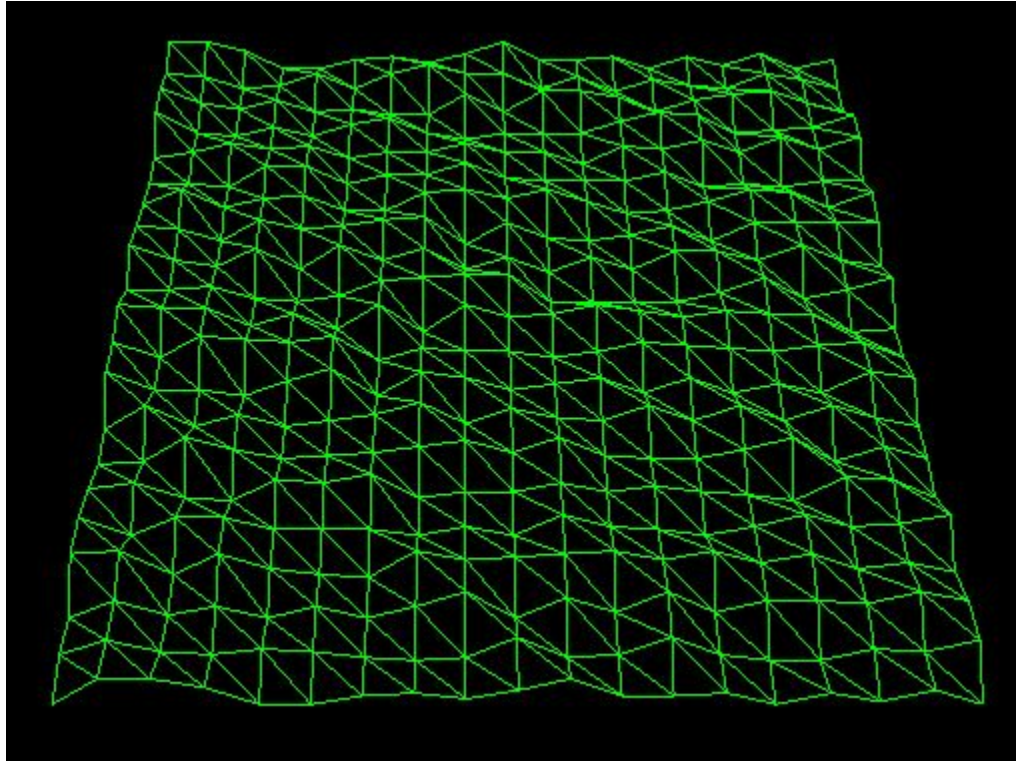


Imagem disponível em: http://efisica.if.usp.br/mecanica/universitario/cinematica/pos_coord_cartes/

Ruído



Perlin Noise

- Desenvolvido por Ken Perlin no anos 80
- Permite definir valores aleatórios suaves

Perlin Noise

1. Definir a grade
2. Produto escalar entre gradiente e a distância de vetores
3. Interpolação linear dos valores do produto escalar

Retorno: valor entre 0 e 1

Perlin Noise

- Obter um ponto x, y e seus vizinhos mais próximos na grade

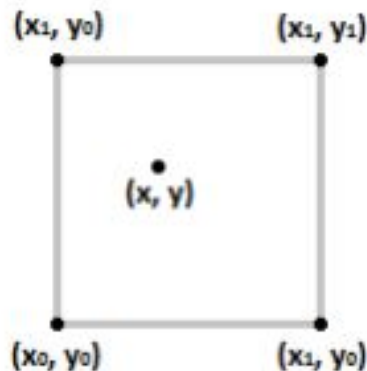


Imagem disponível em: <https://bcc.ime.usp.br/tccs/2013/rec/gustavo/monografia.pdf>

Perlin Noise

- Obter vetor de distância entre ponto x, y e seus vizinhos na grade

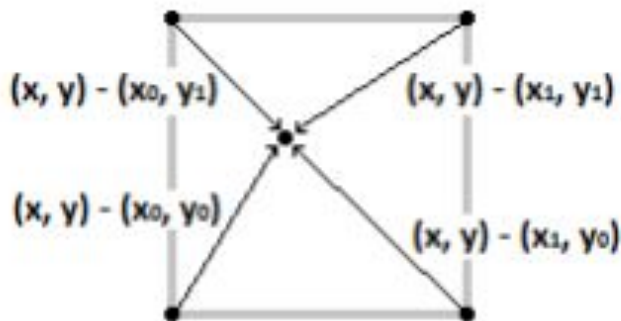


Imagem disponível em: <https://bcc.ime.usp.br/tccs/2013/rec/gustavo/monografia.pdf>

Perlin Noise

- Calcular o produto escalar entre o gradiente neste no ponto e o vetor de distância obtido

$$v1 = \text{gradiente}(x_0, y_0) \cdot ((x, y) - (x_0, y_0))$$

$$v2 = \text{gradiente}(x_1, y_0) \cdot ((x, y) - (x_1, y_0))$$

$$v3 = \text{gradiente}(x_0, y_1) \cdot ((x, y) - (x_0, y_1))$$

$$v4 = \text{gradiente}(x_1, y_1) \cdot ((x, y) - (x_1, y_1))$$

Perlin Noise

- Interpolação os pontos

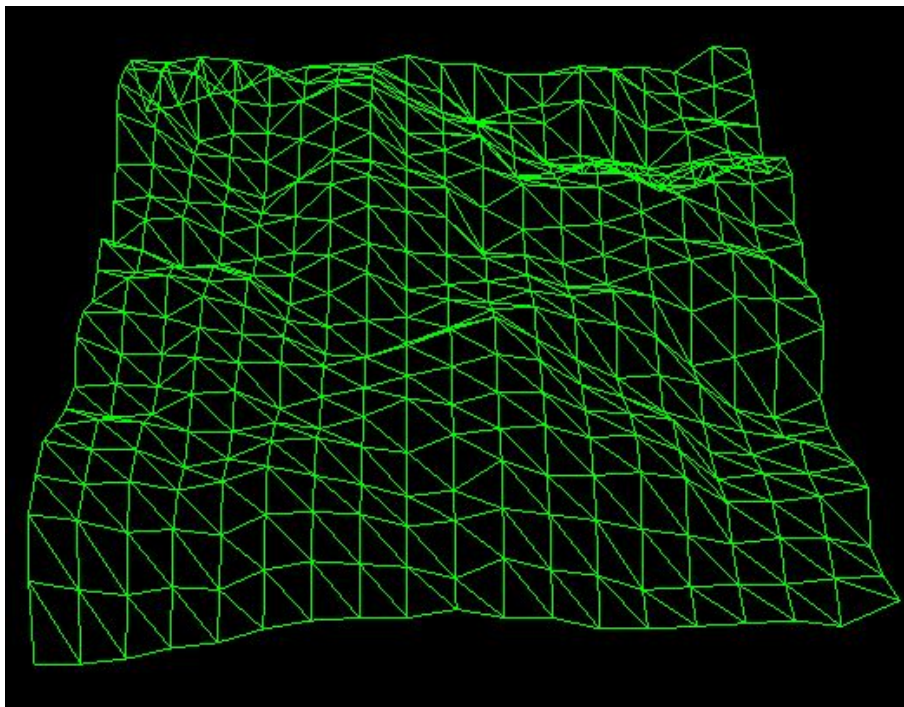
$$\text{interpolar}(v1, v2, x_{grid}) = x1$$

$$\text{interpolar}(v3, v4, x_{grid}) = x2$$

$$\text{interpolar}(x1, x2, y_{grid}) = v_{final}$$

Imagem disponível em: <https://bcc.ime.usp.br/tccs/2013/rec/gustavo/monografia.pdf>

Perlin Noise



Textura

- Mapeamento de textura
 - Idéia: “Aplicar” uma imagem sobre a malha

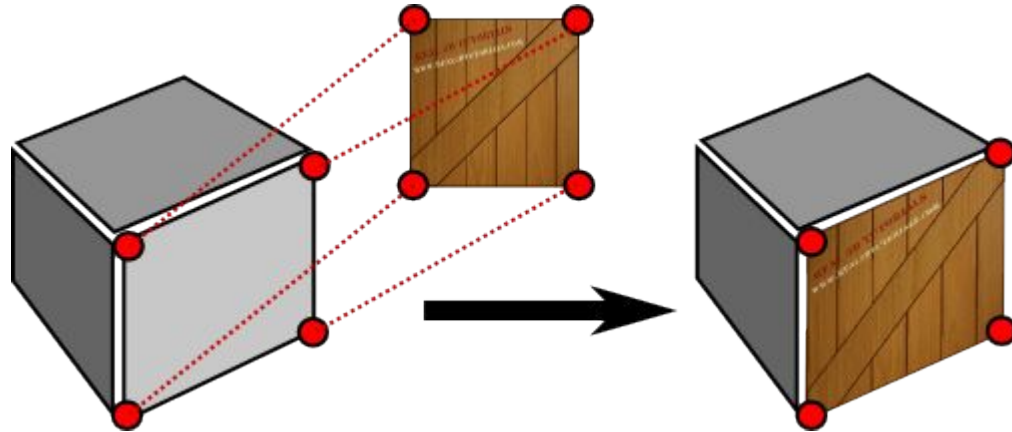


Imagem disponível em: <http://www.real3dtutorials.com/tut00005.php>

Textura

- Correspondência de vértices

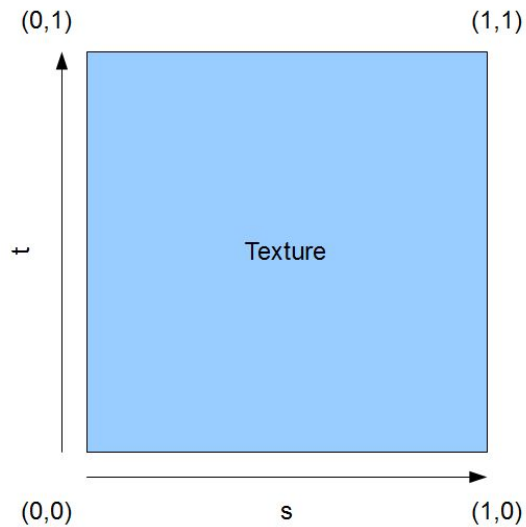


Imagem disponível em: <http://www.learnopengles.com/android-lesson-four-introducing-basic-texturing/>

Textura



Imagens disponíveis em: <https://opengameart.org/content/terrain-textures-pack-from-stunt-rally-23>

Textura

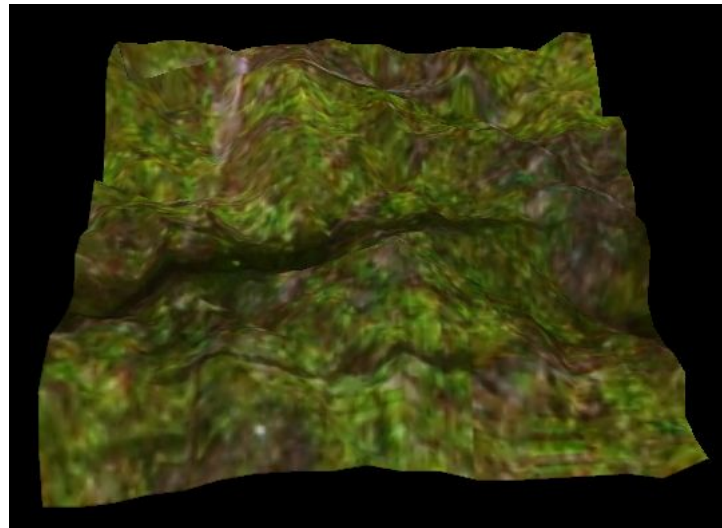
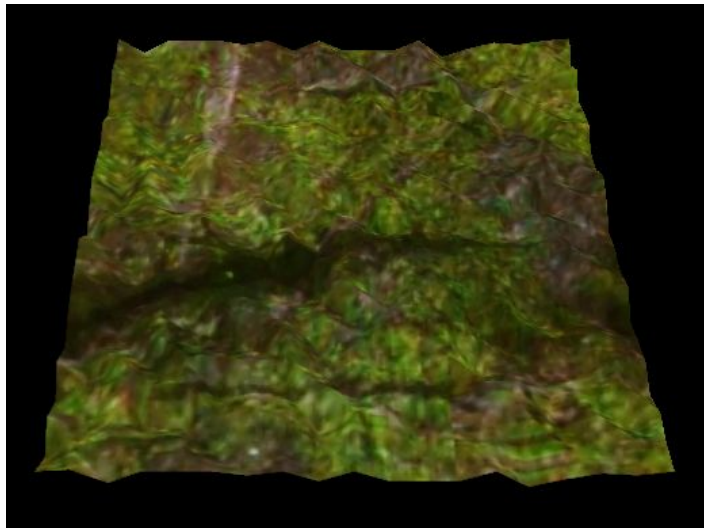
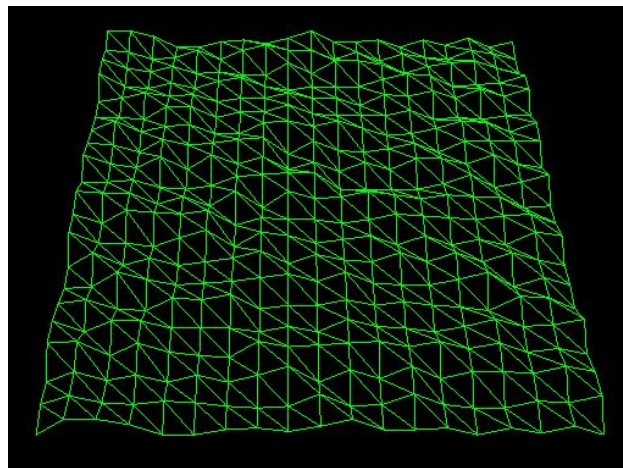
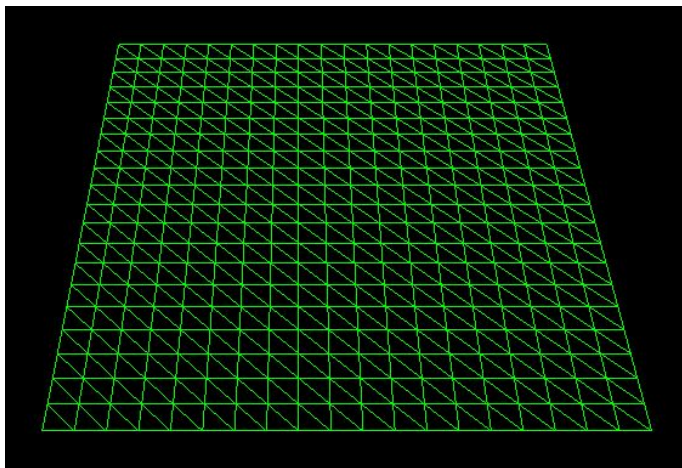
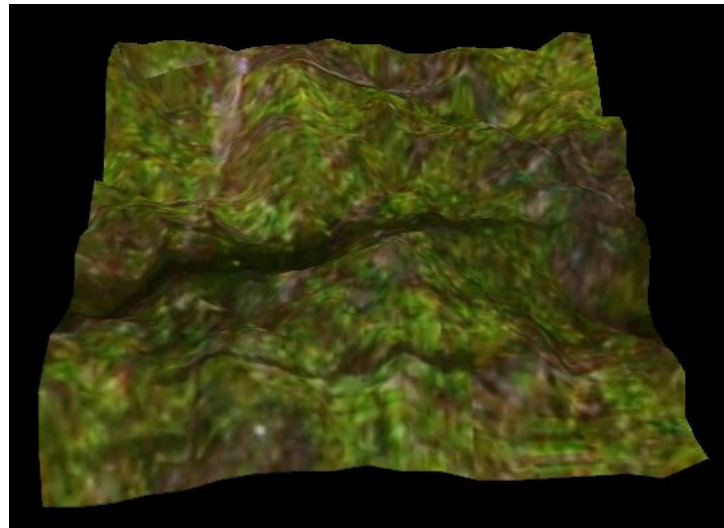
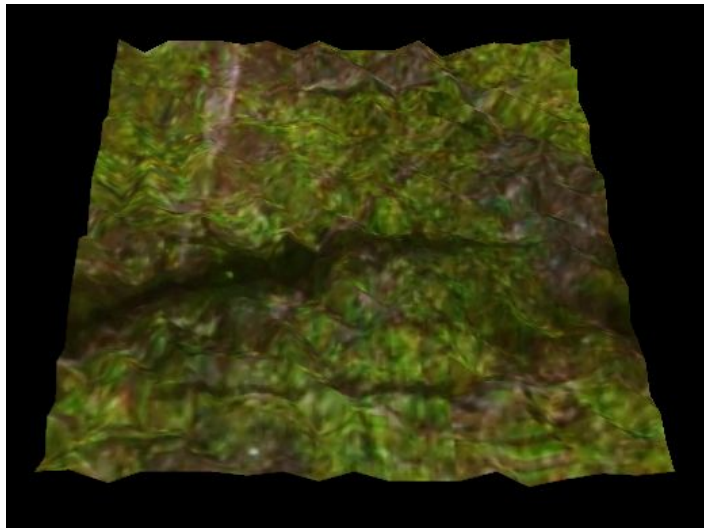


Imagem da malha após aplicação da textura

Resultados



Resultados



Referências

- https://www.youtube.com/timedtext_video?ref=player&v=ELpZW62HGVs
- <https://www.youtube.com/watch?v=Qf4dIN99e2w&list=PLRqwX-V7Uu6bgPNQAdxQZpJuJCjeOr7VD>
- <https://www.youtube.com/watch?v=MJ3bvCkHJtE&t=884s>
- <https://bcc.ime.usp.br/tccs/2013/rec/gustavo/monografia.pdf>
- <http://www.inf.pucrs.br/~manssour/CG/index.html>
- <https://open.gl/textures>

Dúvidas

