

# PUC/RS

# Computação Gráfica

Augusto Weiland

<augusto.weiland@acad.pucrs.br>

2014/1

# Trabalho 1

- o Desenvolvimento de um visualizador de Terrenos 3D com a aplicação de dois algoritmos de otimização, sendo um deles Frustum Culling.
- o O projeto foi desenvolvido em C++ com a biblioteca OpenGL/GLUT/SOIL.

# Controles

## o Teclas Normais:

- o o : Liga/Desliga Algoritmo de Frustum Culling e de Corte de Visão;
- o w : Move a câmera para frente;
- o a : Move a câmera para a esquerda;
- o d : Move a câmera para a direita;
- o s : Move a câmera para trás;
- o r : Subir no mapa;
- o f : Descer no mapa;
- o + : Aumenta 10 unidades de velocidade;
- o - : Reduz 10 unidades de velocidade;
- o esc : Fecha a Janela.

## o Teclas Especiais:

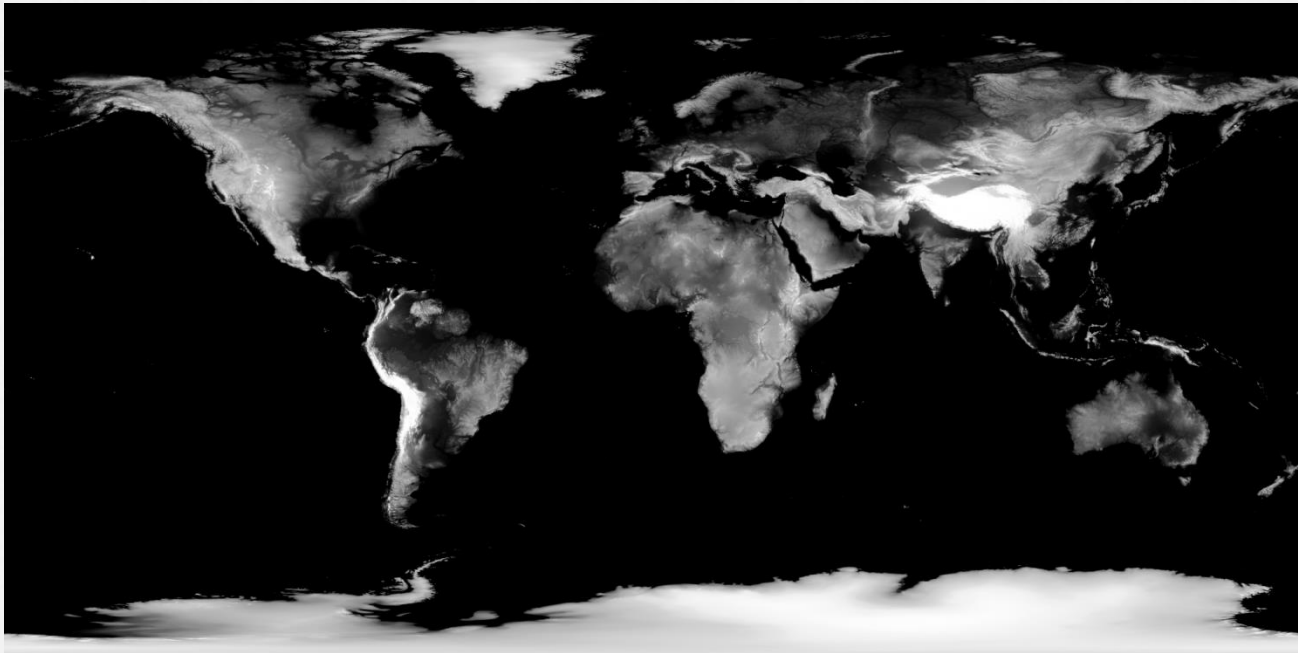
- o Seta para cima : Fullscreen;
- o Seta para baixo : Retorna a janela normal sem Fullscreen.



# Inicializando Algoritmo

- o O sistema carrega uma imagem em escala de cinza, a qual servirá de modelo para o mapa gerado em tela. Por padrão “world.png”.
- o Também por padrão o sistema é iniciado com velocidade 100 e os algoritmos de otimização ativados.

# Mapa “world.png”



# Algoritmo de Frustum Culling

- o Este algoritmo se beneficia da posição atual da câmera do OpenGL, extraíndo suas coordenadas, as quais são utilizadas na renderização do mapa (em tempo real), desta maneira os pontos que não estão dentro do campo de visão desta câmera não serão renderizados.

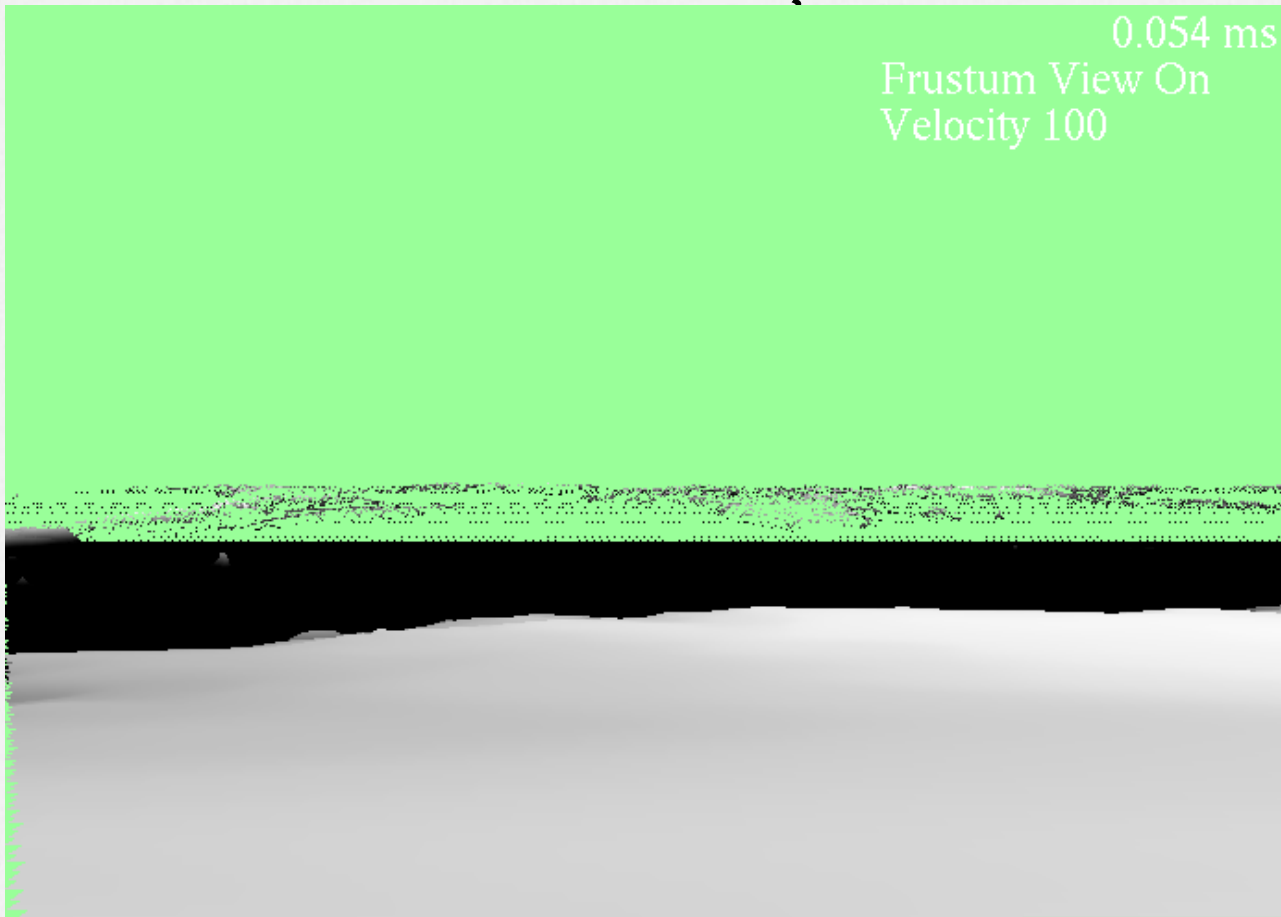


# Algoritmo de Corte de Visão

- o Este algoritmo gera uma visão simplificada dos pontos a uma certa distância do campo de visão da câmera, diferentemente do que as próprias configurações da câmera poderiam realizar, cortando estes pontos. Desta maneira renderizam-se todos os pontos a frente da câmera até seu limite, porém após a distância “x” da atual eles são simplificados.

# Tela inicial da Aplicação

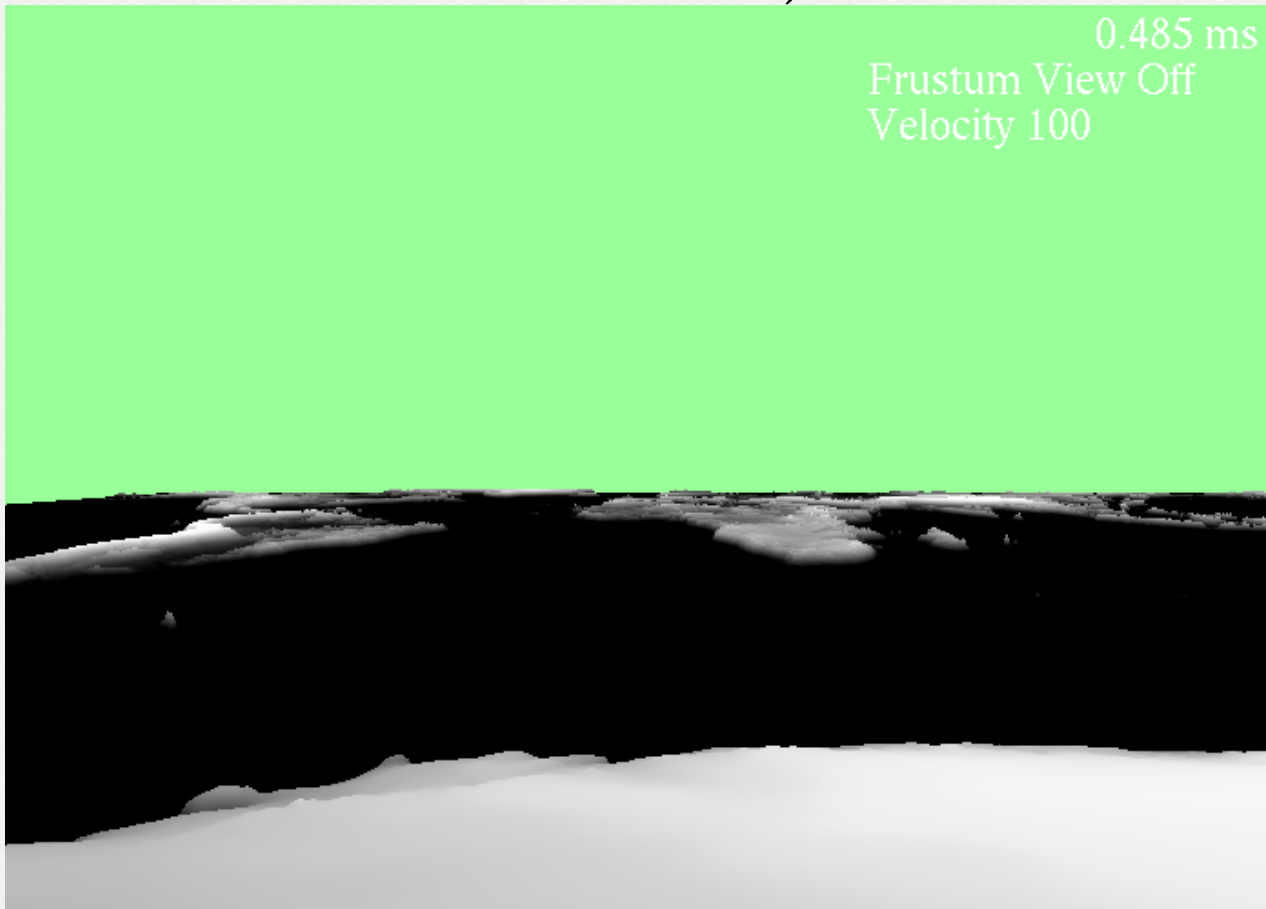
## Com Otimizações





# Tela inicial da Aplicação

## Sem Otimizações



# Tabela de Comparação de Tempo de Renderização

	Com Otimização	Sem Otimização
Média Mínima	0.06 ms	0.47 ms
Média Máxima	0.08 ms	0.49 ms
Média Total	0.07 ms	0.48 ms

# PUC/RS

# Computação Gráfica

Augusto Weiand

[augusto.weiand@acad.pucrs.br](mailto:augusto.weiand@acad.pucrs.br)

Fontes: <https://www.github.com/aweinand>

2014/1