

# Lista 2 - Piramide Usando OpenGL - SDL2

Macártur de Sousa Carvalho

April 18, 2014

## 1 Solução Proposta

A solução proposta nesta lista três é aprender a utilizar renderização de malhas triangulares em um programa utilizando SDL2 e OPENGGL , se a utilização de um shader.O arquivo a ser lido como amostra é um coelho no formato .off.Este formato é simples de ser lido e utilizado em programa feito na mão.

## 2 Estrutura de Dados Proposta

As estruturas de dados utilizadas para fazer está solução foram:

- Mesh(Abstração de malha triangular);

Contendo os atributos: Xmin,Ymin,Zmin,Xmax,Ymax,Zmax,vetor de pontos,vetor de faces,inteiro indicando se tem textura, inteiro indicando se tem cor, deltaX,deltaY,deltaZ, e um nFaces e nDots indicando a quantidade de pontos a serem armazenados no Mesh.

- Dot(Abstração de pontos):

Contendo os atributos: x,y,z que armazena as cordendas dos pontos; red,green,blue para armazenar os valores das cores nos pontos; alpha para armazenar o valor do ponto é opaco ou transparente;

- Face(Abstração de faces do objeto):

Contendo os atributos: vetor de indices que armazena os 3 ou quatro indices passados na leitura do arquivo;

### 3 A malha Poligonal

A malha poligonal é uma coleção de faces (onde cada face é um conjunto de vértices) que definem um objeto tridimensional nos campos da computação gráfica e da modelagem tridimensional. As faces geralmente são constituídas de triângulos (também chamada de malha triangular) ou quadriláteros, uma vez estas formas simplificam o processo de renderização, no entanto também podem ser compostas por formas geométricas mais complexas.

### 4 Principais formatos de arquivos de malhas 3D

Para a criação de arquivos de formatos de arquivos de malha 3d existem varias extensões possíveis tais como: 3DS(3D Studio), BLEN( BLENDER ), DAE (COLLADA), DXF (AutoCAD), FBX (Autodesk exchange), LOW (Lightwave), OBJ, OFF, PLY, PTS, PTX, SC1(Sculptris), SCL(Pro/engineer), SKP(Google sketchup), STL, TRI, V3D, WRL (VRML), X3D, X3DV, XSI(SoftImage), ZTL(Zbrush), XYZ.

Todos eles possuem uma maneira diferente de ser lido, para esta lista será citado a leitura do STL e do OBJ.

#### STL

O início da linha contém: solid name, onde name é opcional.

O arquivo continua com a leitura de triângulos iniciando, com facet normal indicando o vetor normal, com valores  $n_i, n_j, n_k$ , em seguida faz a leitura de todos os vértices do vetor.

Por fim fecha-se o loop de captura do vetor, o facet e o solid.

Exemplo:

Com name:

solid name

facet normal  $n_i$   $n_j$   $n_k$

outer loop

vertex  $v_{1x}$   $v_{1y}$   $v_{1z}$

vertex  $v_{2x}$   $v_{2y}$   $v_{2z}$

vertex  $v_{3x}$   $v_{3y}$   $v_{3z}$

endloop

```

endfacet
endsolid name
Sem name:
solid
facet normal ni nj nk
outer loop
vertex v1x v1y v1z
vertex v2x v2y v2z
vertex v3x v3y v3z
endloop
endfacet
endsolid

```

## OBJ

Neste arquivo linhas iniciadas com '#' são comentários.

Primeiramente é feito a leitura dos vertex, com 4 parametros, onde o quarto é opcional e definido com default em 1.0.

Exemplo de linha: v 0.123 0.234 0.345 1.0

O outro valor lido a coordenadas das texturas , que recebem 3 parametros entre 0 e 1 e o ultimo parametro é opcional e por default vale 0;

Exemplo de linha: vt 0.5000 1 0

Em segui é feito a leitura de vetores normais, tendo como magnitude unitaria

Exemplo de linha: vn 0.707 0.000 0.707

Em seguida é capturado parametros de espaços do vertex.

Exemplo de linha: vp 0.310000 3.210000 2.100000

E por fim contem os valores das faces definidos como:

f v1/ vt1 / vn1 ,

a utilização de textura é opcional podendo ser retirada ficando

f v1 // vn1

Exemplo de Arquivo:

# List of Vertices, with (x,y,z[,w]) coordinates, w is optional and defaults to 1.0.

v 0.123 0.234 0.345 1.0

v ...

...

```

    # Texture coordinates, in (u ,v [,w]) coordinates, these will vary between
0 and 1, w is optional and default to 0.
    vt 0.500 1 [0]
    vt ...
    ...
    # Normals in (x,y,z) form; normals might not be unit.
    vn 0.707 0.000 0.707
    vn ...
    ...
    # Parameter space vertices in ( u [,v] [,w] ) form; free form geometry
statement ( see below )
    vp 0.310000 3.210000 2.100000
    vp ...
    ...
    # Face Definitions (see below)
    f 1 2 3
    f 3/1 4/2 5/3
    f 6/4/1 3/5/3 7/6/5
    f ...
    ...

```

## 5 Principais Lições Aprendidas

Nesta lista foi possível entender melhor o funcionamento de uma malha triangular utilizando opengl, e como realizar a leitura de arquivos que utilizam formatos que armazenam malhas 3d.

## 6 Principais Dificuldades Encontradas

As principais dificuldades encontradas foram ler todos os pontos de maneira que todos sejam armazenados de forma correta nas estruturas e a função que mais foi difícil de ser utilizada foi a de redimensionar, pois deve-se entender bem a teoria para se aplicar os cálculos nas imagens. Sabendo todos as funções que pode-se implementar fica bem fácil de se realizar a renderização dos arquivos.

## 7 Imagem solução proposta

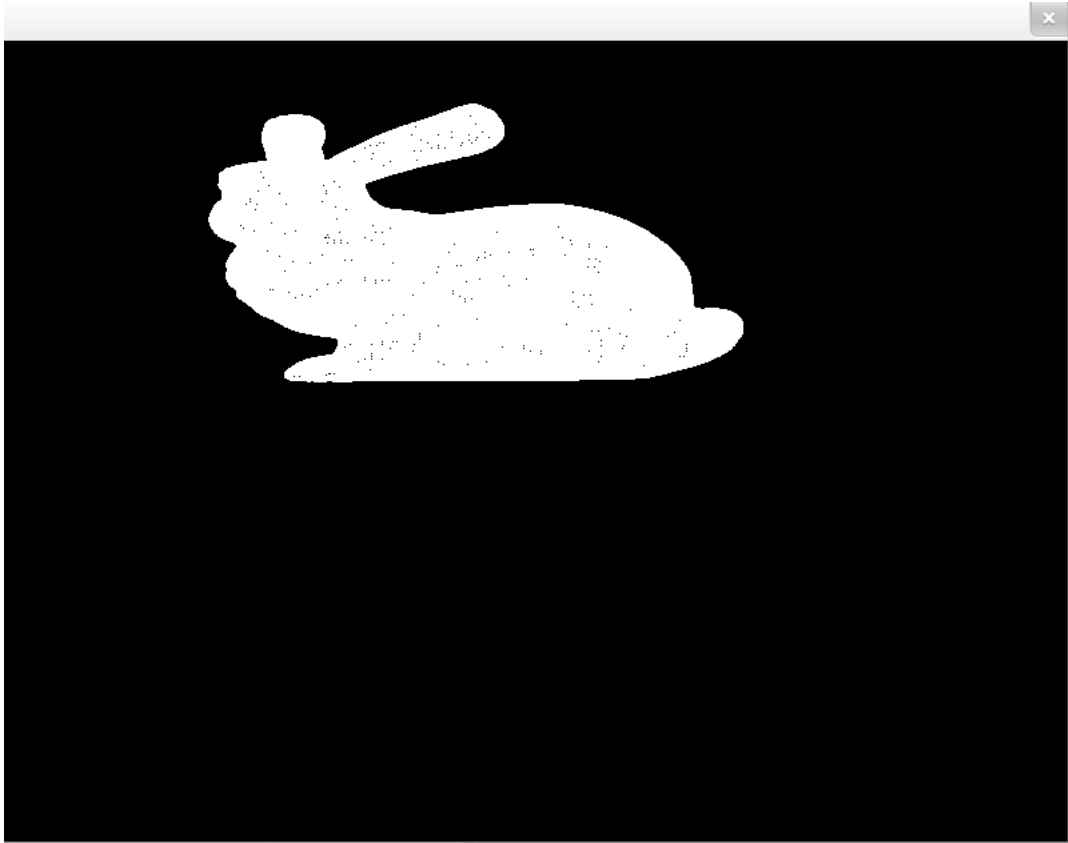


Figure 1: Figura utilizando Linhas(Vertices) de um coelho

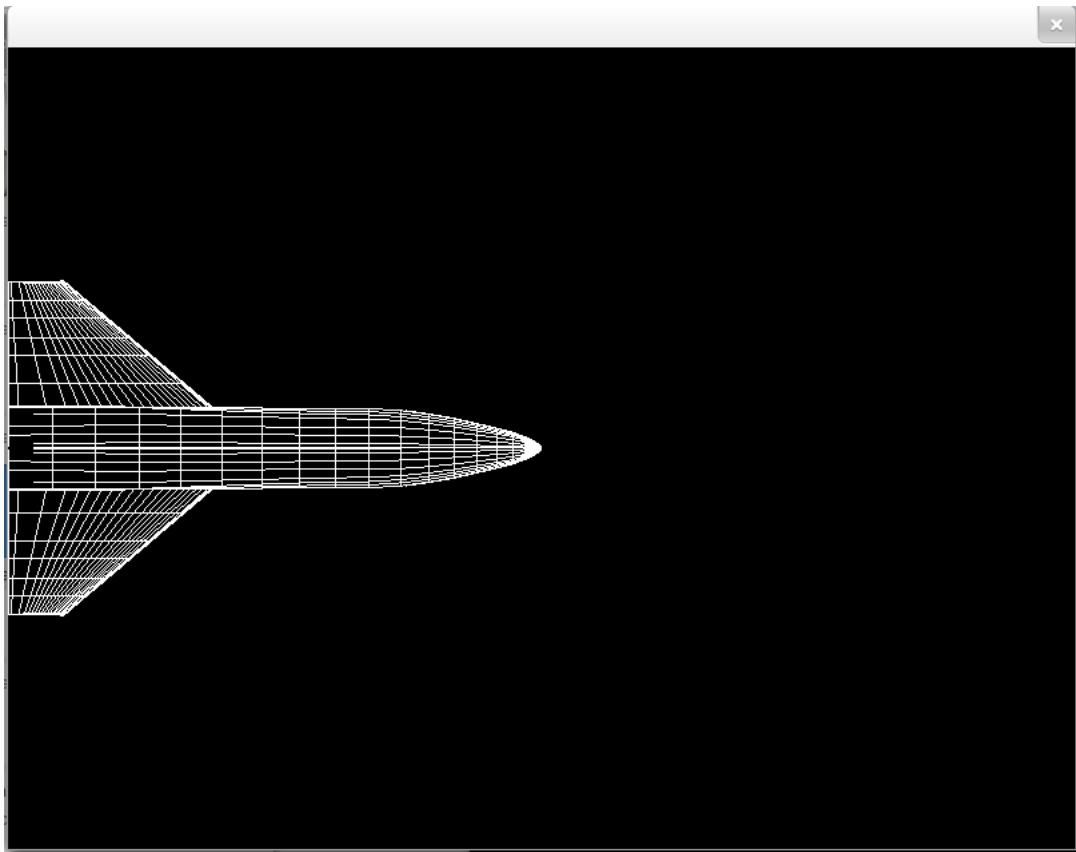


Figure 2: Figura utilizando linhas para formar uma nave espacial

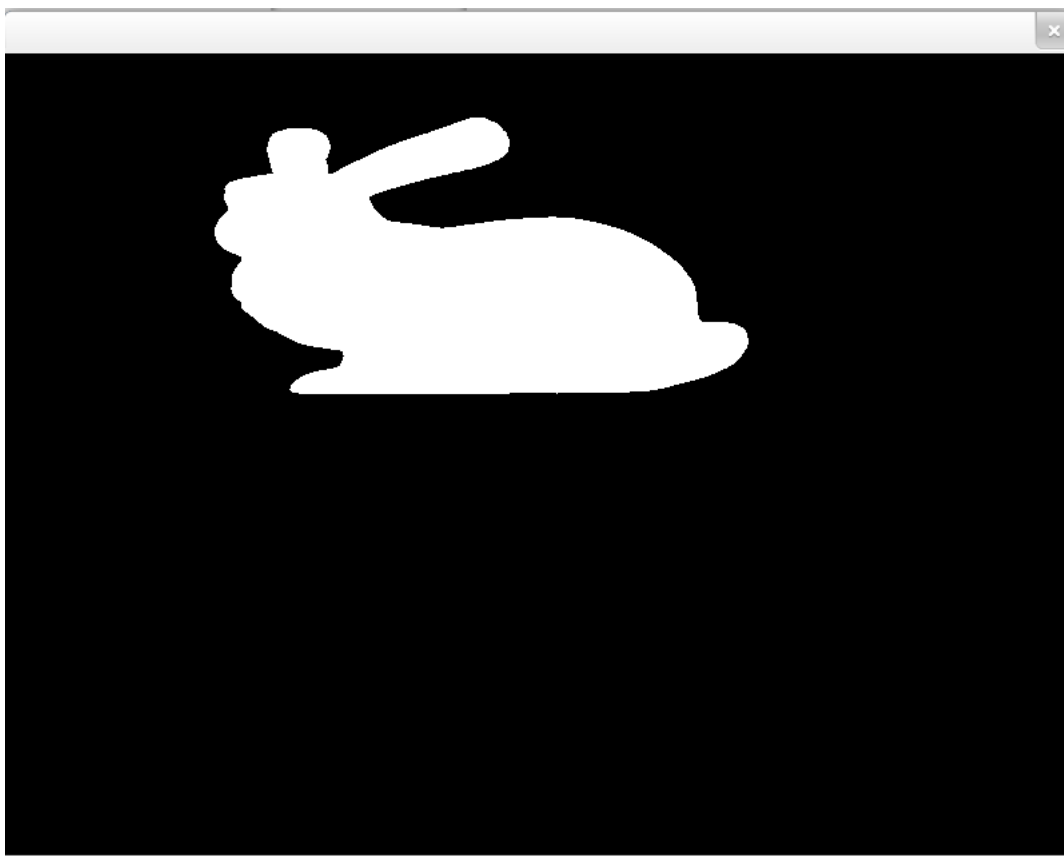


Figure 3: Figura utilizando poligonos para formar uma coelho