Universidade Federal de Santa Maria
Departamento de Eletrônica e Computação
Prof. Cesar Tadeu Pozzer
Disciplina: Computação Gráfica Avançada
pozzer@inf.ufsm.br
03/09/2008

Animação de Personagens

1. Animação por *key-framing*

Técnica muito simples de animação. Consiste em criar diferentes versões da geometria do modelo e depois realizar uma interpolação para a obtenção de posições intermediárias. A interpolação pode ser linear ou de grau maior.

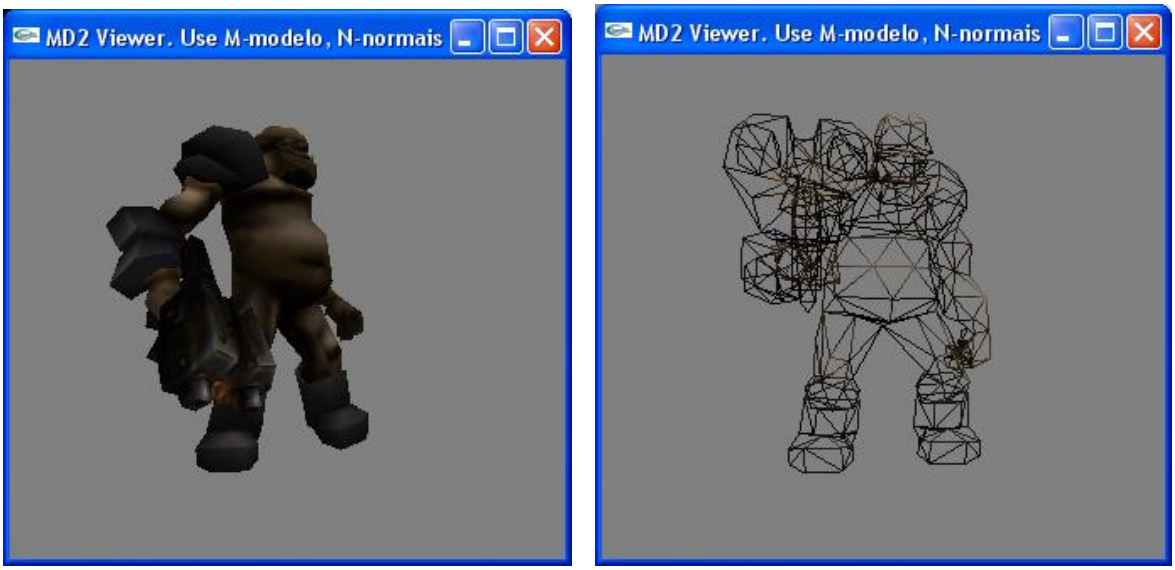
$$f(\alpha) = (1-\alpha)v_i + \alpha v_{i+1}, \quad 0 \le \alpha \le 1$$

O número de quadros (*frames*) vai depender da qualidade da animação e número de animações possíveis. Cada frame guarda a posição de cada vértice do modelo em uma dada configuração. Neste sistema de animação, não se tem informação topológica da geometria do modelo animado.

Existem outras formas mais complexas de animação, como no caso de cinemática inversa ou animação por esqueleto. Para maiores detalhes, consulte [1].

2. Formato MD2

O formato MD2 (http://tfc.duke.free.fr/coding/md2-specs-en.html) foi criado para ser utilizado no jogo Quake 2 (http://www.quake2.com/). Faz uso da técnica de *key-framing*. Cada modelo neste formato é geralmente é representado por dois arquivos: o arquivo do personagem e o arquivo da arma. Deste modo, é possível renderizar o personagem com ou sem a arma, ou somente renderizar a arma. A animação da arma segue a animação do modelo, logo o modelo que representa a arma tem o mesmo número de frames que o arquivo do modelo. Na seguinte figura são apresentados exemplos de um modelo MD2 visualizado com textura e em formato de wireframe. Foi utilizado o programa MD2 Viewer (ver demo gl_md2viewer).



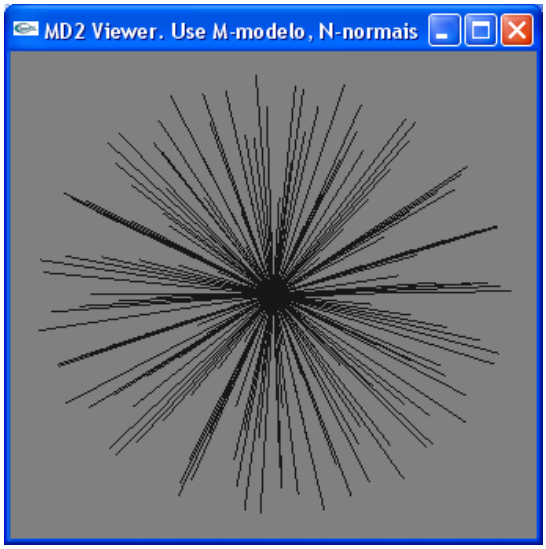
O número de frames do formato MD2 é limitado em 512. O jogo Quake define um padrão de 198 quadros, assim distribuídos:

Animação	Quadro inicial	Quadro final
STAND	0	39
RUN	40	45
ATTACK	46	53
PAIN_A	54	57
PAIN_B	58	61
PAIN_C	62	65
JUMP	66	71
FLIP	72	83
SALUTE	84	94
FALLBACK	95	111
WAVE	112	122
POINT	123	134
CROUCH_STAND	135	153
CROUCH_WALK	154	159
CROUCH_ATTACK	160	168
CROUCH_PAIN	169	172
CROUCH_DEATH	173	177
DEATH_FALLBACK	178	183
DEATH_FALLFORWARD	184	189
DEATH_FALLBACKSLOW	190	197
BOOM	198	198

Deste modo, para renderizar um personagem correndo deve-se renderizar os quadros entre 40 e 45. O formato também define em 4096 o número máximo de triângulos e 15000 o número de vértices.

2.1. Vetor de normais pré-calculadas

Para reduzir a quantidade de informação presente no arquivo, ao invés de se utilizar um vetor normal para cada vértice (3 float * 4 bytes = 12 bytes), utiliza-se um índice do tipo char para indexar em uma matriz pré-calculada de 162 normais normalizadas (ver figura). Deste modo gasta-se apenas 1 byte para a normal de cada vértice. Este conjunto de normais, apesar de não ser tão preciso, gera resultados muito satisfatórios na iluminação, visto que os modelos possuem poucas faces. Pequenos erros nos vetores normais não vão ser perceptíveis na visualização do modelo.




```
{ -0.525731f,  0.000000f,  0.850651f },
{ -0.442863f,  0.238856f,  0.864188f },
{ -0.295242f,  0.000000f,  0.955423f },
{ -0.309017f,  0.500000f,  0.809017f },
{ -0.162460f,  0.262866f,  0.951056f },
{  0.000000f,  0.000000f,  1.000000f },
{  0.000000f,  0.850651f,  0.525731f },
{ -0.147621f,  0.716567f,  0.681718f },
{  0.147621f,  0.716567f,  0.681718f },
{  0.000000f,  0.525731f,  0.850651f },
{  0.309017f,  0.500000f,  0.809017f },
{  0.525731f,  0.000000f,  0.850651f },
{  0.295242f,  0.000000f,  0.955423f },
{  0.442863f,  0.238856f,  0.864188f },
{  0.162460f,  0.262866f,  0.951056f },
{ -0.681718f,  0.147621f,  0.716567f },
{ -0.809017f,  0.309017f,  0.500000f },
{ -0.587785f,  0.425325f,  0.688191f },
...
```

O vetor de inteiros `lightnormals[]` do arquivo MD2 tem dimensão igual a num_frames * num_faces.
O vetor de normais `vertexNormals[]` contém as 162 normais pré-calculadas.

2.4. Renderização do Modelo

O número de vértices do modelo é dado pela variável `num_xyz`. Desta forma, o número de vértices e normais do arquivo são dados por `num_xyz*num_frames`. Para renderizar o modelo, após carregar os dados do arquivo, deve-se inicialmente cria-se o vetor de vértices em função da animação, parâmetro de interpolação e quadro renderizado anteriormente.

```
for( i = 0; i < num_xyz ; i++ )
{
   vert[i][0] = curr_vlist[i][0] + *interpol * (next_vlist[i][0] - curr_vlist[i][0]);
   vert[i][1] = curr_vlist[i][1] + *interpol * (next_vlist[i][1] - curr_vlist[i][1]);
   vert[i][2] = curr_vlist[i][2] + *interpol * (next_vlist[i][2] - curr_vlist[i][2]);
}
```

O vetor de inteiros apontado por **`ptricmds`** está organizado da seguinte forma: o primeiro inteiro contém o número de vértices por cada primitiva. Se o valor for positivo, a primitiva é um **`GL_TRIANGLE_STRIP`**, caso contrário um **`GL_TRIANGLE_FAN`**. Na seqüência, cada grupo de 3 índices está associado a cada vértice. Os dois primeiros são as coordenadas de textura (valor float) e o terceiro o índice do vértice e da normal. No geral, o numero de gl_commands em um arquivo MD2 varia entre 170 e 300.

```
while( i = *(ptricmds++) ) //while (i!=0)
{
   if( i < 0 )
   {
      glBegin( GL_TRIANGLE_FAN );
      i = -i;
   }
   else
   {
      glBegin( GL_TRIANGLE_STRIP );
   }

   // ptricmds[0]: coordenada de textura s
   // ptricmds[1]: coordenada de textura s
   // ptricmds[2]: índice do vértice e normal
   for( ; i > 0; i--, ptricmds += 3 )
   {
      glTexCoord2f( ((float*)ptricmds)[0], ((float*)ptricmds)[1] );

      glNormal3fv( vertexNormals[ lightnormals[ ptricmds[2] ] ] );

      glVertex3fv( vertices[ ptricmds[2] ] );
   }
   glEnd();
}
```

Vetor de índices das normais pré-calculadas. Lido do arquivo MD2

2.3. Geração do arquivo

Pode-se utilizar o programa 3D Studio Max, com plugin específico para exportar modelos 3D animados neste formato. Um exemplo de plugin é o QTIP [http://www.qtipplugin.com/].

3. Referências

[1] Alan H. Watt. **3D Computer Graphics, A**ddison Wesley; 3 edition, 1999.