MD2

Nathan Cournia

nathan@cournia.com

Clemson University

Outline

- Overview
- Features
- Loading
- The Bad
- The Good
- Creating
- Uber’s MD2

Overview

- Animated mesh format
- Created by id Software’s for Quake 2 engine in 1997
- Popular in demo scene

Features

- Simple binary format
- GL optimizations
- Vertex animation
- Texture (skin) information
- Indexed format
- Hundreds of free models available
- Central repository for models (http://www.polycount.com)

Data Format

- Binary Format
- Little Endian
- Data type sizes:
  - `char` 1 byte
  - `short` 2 bytes
  - `int` 4 bytes
  - `float` 4 bytes

Loading: Header

```
struct md2_header {
  char magic[ 4 ]; //Magic number. Always IDP2.
  int version;     //MD2 format version. Always 8.

  int skin_width;  //Width of texture (ex. 256).
  int skin_height; //Height of texture (ex. 256).
  int frame_size;  //Size of each frame in bytes.

  // . . .
```

Loading: Header Cont.

```
int num_skins;  //Number of textures.
int num_xyz;    //Number of vertexes in a frame.
int num_st;     //Number of texture coords in a frame.
int num_triangles; //Number of triangles in a frame.
int num_glcmds; //Number of dwords (4 bytes) in list.
int num_frames; //Number of frames in file.

// . . .
```

Loading: Header Cont.

```
  //offsets in bytes from start of file
  int offset_skins;     //To list of textures.
  int offset_st;        //To texture coords.
  int offset_triangles; //To triangle data.
  int offset_frames;    //To frame data.
  int offset_glcmds;    //To OpenGL specific command data.
  int offset_eof;       //Size of the file (in bytes).
}; //end of md2_header
```

Loading: Triangles

- Triangle vertex relationships do not change over frames.
- Only need 1 list of triangles.
- Triangles reference changing vertexes.

Loading: Triangles Cont.

- Move to header.offset_triangles
- Read in header.num_triangles md2_triangles

```
struct md2_triangle {
  short vertex_index[ 3 ];
  short st_index[ 3 ];
};
```

Loading: Key Frames

- Standard MD2 format contains 198 key frames
- Certain frame ranges contain animation sequences
- Examples:
  - stand: 0-39
  - taunt: 95-111
  - crouch pain: 169-172
- Linear interpolation is used between frames

Loading: Key Frames Cont.

- Move to header.offset_frames
- Read in header.num_frames md2_frames

```
struct md2_frame
{
  float scale[ 3 ];     //Scaling factor
  float translate[ 3 ]; //Translation factor
  char name[ 16 ]; //Name of the frame, (ex. death202)
  md2_vertex vertexes[ header.num_xyz ];
};
```

Loading: Key Frames Cont.: Vertexes

- md2_vertex vertexes[header.num_xyz] is a variable sized array of vertexes.

```
struct md2_vertex
{
  ubyte vertex[ 3 ]; //Compressed x, y, z vertex point.
  ubyte normal_index; //Normal index.
};
```

Loading: Key Frames Cont.: Vertexes

- Vertex data is in compressed format.
- Uncompress vertex with:

```
vertex * md2_frame.scale + md2_frame.translate
```

- `md2_vertex.normal_index` is an index into the Quake 2 engine’s normal table.

Loading: Texture Coordinates

- Texture coordinates do not change over frames.
- Move to `header.offset_st`
- Read in `header.num_xyz` `md2_sts`

```
struct md2_st
{
  short s;
  short t;
};
```

Loading: Texture Coords Cont.

- s, t coordinates are in pixels.
- Need to normalize to the range [0-1] to be useful for OpenGL.

```
u = 1.0 - s / header.skin_width;
v = s / header.skin_height;
```

Loading: Textures

- A single image contains all texture information for the entire model
- At `header.offset_skins` is a list of textures that can be used by the model
- This field is rarely defined (even in id Software's standard models)

Loading: Textures Cont.

- A valid skin for a player model must have a texture and icon file. ex: grunt.pcx and grunt_i.pcx
- For player models, the following psuedo code is a good way to search for valid textures

```
chdir( model_dir );
foreach file in *.pcx {
    if( exists( *_i.pcx ) ) {
    return *.pcx as a valid texture
  }
}
```

The Bad

- Slew of bad models on the net
  - Wrong Normals
  - Edges shared by more than 1 triangle
  - Leaky models
- Vertex animation
- No skeleton

The Good

- No skeleton
- Lots of models
- Community still active
- Easily extended for non-standard animation and multiple texturing passes

Creating

- Standard tools can be used with plug-ins
  - 3D Studio Max
  - Lightwave
  - Maya
  - MilkShape3D

Uber's MD2 Support

- Full MD2 support is in Uber.
- md2 object is a child of `mesh3`
- Loading up an md2 file is as simple as:

```
md2 model( "grunt.md2" );
if( !model ) {
  std::cerr << model.get_error( ) << std::endl;
}
md2.play( "stand" );
md2.update( 0.0 );
```

Uber’s MD2 Support Cont.

- Updating and rendering a MD2 model is as simple as:

```
while( 1 ) {
  //tell the model how much time has elapsed
  //since last frame
  model.update( uber.video.get_elapsed( ) );

  //render the model
  model.render( );
}
```

Uber’s MD2 Support Cont.

- Of course, Uber’s MD2 object is capable of much more
- See demos/meshviewer/md2viewer and Uber’s API docs for more info

Questions?