

EuSDL_MD2 file loader
Steve Allen
Released under LGPL.

This document is a first run release so its not "official" yet. Please bare with me as this will be improved upon along with future releases of the library.

This lib is for loading MD2 files into euphoria sequences for simpler rendering. I noticed that we don't have any 3D loaders yet so I thought I would create one for the very popular MD2 format. Also included is a loader for RAW triangles. To use the library you need Mark Akita's SDL_GL package.

I mainly wrote this to learn more about OpenGL so the library is far from optimized. I like to comment a lot when I am learning so the code is very well commented. This release does not support MD2 textures as I haven't completely figured them out yet. I am still working on this.

The three main routines are as follows:

`load_MD2_raw(sequence md2file)` -- This will load the file completely as raw information. It is still formatted into a Euphoria sequence though. Sequence md2file is the MD2 filename.

`load_MD2_formatted(sequence md2file)` -- This will load the file into a logical Euphoria sequence format. It is the simplest way of representing the data. The source code for this function is commented the most. It is pretty inefficient due to my attempt at keeping all of the data formatted.

`load_MD2_slim(sequence md2file)` -- This will create a formatted sequence like `_formatted` but will remove unnecessary data. This creates a leaner object that is less resource hungry. If you are going to use any of these routines in a real program this is the one to use.

All of the functions will return 1 of 2 sequences.

No Errors: Will return a sequence with the following structure:

```
MD2_object = {  
    0, 0, 0, 0, 0, 0, 0, 0, 0,  
    " ", -- Texture path & name (64 characters)  
    {}, -- Texture coordinate data (variable)  
    {}, -- Triangles (variable)  
    {}, -- Frames (variable)  
    {} -- GL commands (variable)  
}
```

Each part of the MD2 object can be accessed by the following pre-built constants:

```
MD2OBJ_SKIN_WIDTH = 1
MD2OBJ_SKIN_HEIGHT = 2
MD2OBJ_FRAME_SIZE = 3
MD2OBJ_NUM_SKINS = 4
MD2OBJ_NUM_VERTICES = 5
MD2OBJ_NUM_ST = 6
MD2OBJ_NUM_TRIS = 7
MD2OBJ_NUM_GLCMDS = 8
MD2OBJ_NUM_FRAMES = 9
```



header info....

```
MD2OBJ_TEXTURE = 10
MD2OBJ_TEXTURE_COORDS = 11
MD2OBJ_TRIANGLES = 12
MD2OBJ_FRAMES = 13
MD2OBJ_GLCMDS = 14
```

```
MD2OBJ_FRAME_SCALE = 1
MD2OBJ_FRAME_TRANSLATE = 2
MD2OBJ_FRAME_NAME = 3
MD2OBJ_FRAME_VERTICES = 4
```

Frames will have this format:

```
{ {SCALE1, SCALE2, SCALE3},
  {TRANS1, TRANS2, TRANS3},
  "FRAME NAME",
  {
    (for each vertex)
    {
      {X, Z, Y}, (all uncompressed)
      INDEX
    },
    {
      {X, Z, Y},
      INDEX,
    },
    ...
  }
}
```

Hence,

```
myMD2 = load_MD2_formatted("psychodeathman.MD2")
```

```
myMD2[ MD2OBJ_FRAMES][1] <-- would point to the 1st frame's data.
```

Errors: Will return a 2 element sequence with the following format:
`{-1, "Error Message"}`

The included “game.exw” shows how to load and render a model in this format. It is not the prettiest file in the world though and the texture code is botched up somehow. Consider this first release a WIP. Also, some of the code from the example program is completely stripped out of Mark Akita's demos in his awesome SDL_GL package.

Please provide ANY comments/suggestions/bugs/etc. This is a living library.

Thanks,

Steve A.