M3 Engineering Bay by SoodaTH july 2011

Aim of this text is to teach you reading m3 file in binary form. I am self-teaching myself on this topic and my assumptions may (and will be) wrong at times.

About used files,
Our test file is bunker.m3
Hex editor is 010 Editor
bunker.m3 was parsed with M3 Template 0.96.bt to produce „Template Results" which I observe in my writings.
Here first lines in hex editor look like this:
line1: 33 33 44 4D F0 EE 01 00 A6 02 00 00 01 00 00 00
line2: 01 00 00 00

I suspect user at sc2mapster.com (NiNtoxicated01) decided out of blue to call this block of highlighted HEX code „struct FileHeader fHead". Line1 and line2 make together complete „struct". Line1 and 2 could be together also called „carrot" (a eatable vegetable) but instead was named fancy „struct FileHeader fHead". Really matters to programs what does these HEX numbers like 33 and 33 at start stand for.

Best to teach is giving example and then commenting what each element stands for. At Volcore's web-blog „struct FileHeader fHead" is defined like this:

```
struct M3Header {
fourcc header_tag
uint32 tagindex_offset
uint32 tagindex_size
uint32 unknown1
uint32 unknown2
}
```

This is nice way to visualize struct (really this is just another way of unlimited ways to imagine how these Line1 and line2 could look like on paper). We are getting somewhere look at this:

example1: 33 33 44 4D

Volcore names it `fourcc` while NiNtoxicated01 names it: char fType[4], they both tell same thing that data in example1 has data type of `fourcc` (stands for "four character code").

In 010 Editor char fType[4] is generic name for example1 and it branches to:
char fType[0] = 33
char fType[1] = 33
char fType[2] = 44
char fType[3] = 4D
You get 4 char fTypes each having their unique index. char fType[4] has number 4 in it because it holds inside itslef 4 values. This is just a decoration, un-needed info to you.

To continue our observation „struct FileHeader fHead" `fourcc header_tag`, the `header_tag` (example1) part in soul holds value: 33DM
To get it we need conversion table (go to www.asciitable.com and look table):
HEX 33 in decimal is 51 and as char(acter): 3
HEX 33 in decimal is 51 and as char(acter): 3
HEX 44 in decimal is 68 and as char(acter): D
HEX 4D in decimal is 77 and as char(acter): M

Other parts in „struct FileHeader fHead“ work same way. I write down what points to what:

example6 start (How to understand given examples?)
line1: A6 02 00 00
<HEX code>
`uint32 tagindex_size` = ULONG nTags
data type human given name (Volcore's) = NiNtoxicated01's given name
example6 end

example2 start
line1: 33 33 44 4D F0 EE 01 00 A6 02 00 00 01 00 00 00
line2: 01 00 00 00
struct FileHeader fHead

=

```
struct M3Header {
...
...
}
```

example2 end

example3 start
line1: F0 EE 01 00
`uint32 tagindex_offset` = ULONG ofsTags
example3 end

Volcore writes: „The tagindex is the aforementioned list of tags, and starts at `tagindex_offset` and has `tagindex_size` elements. “

example4 start
line1: A6 02 00 00
`uint32 tagindex_size` = ULONG nTags
example4 end

example5 start
line1: 01 00 00 00
line2: 01 00 00 00
`uint32 unknown1 & uint32 unknown2` = struct HeadRef mref
example5 end

All examples can be considered as types which values are stored as HEX numbers. HEX file has structure, that is why we know first 4 HEX numbers make up 33DM and is considered „`header_tag`“. Though m3 is considered dynamic, file header is static (consider it as a starting point).

We have reached end of our observation. I did not cover every header struct element because I don't know myself also what they are. Consider this text as from noob to noob in a good way. I would like to thanks and give credit to: Volcore and NiNtoxicated01 for their efforts and knowledge sharing.

Useful links (continues on page 3):
http://volcore.limbicsoft.com/2010/02/starcraft-2-model-format-pt-1.html

http://volcore.limbicsoft.com/2010/03/starcraft-2-model-format-pt-2.html
http://code.google.com/p/libm3/
http://www.asciitable.com/
http://www.sweetscape.com/010editor/
http://code.google.com/p/libm3/downloads/detail?name=M3_Template_0.96.zip&can=2&q=