thinEdge M15 file format specification

Official documentation on model format used in ThinBASIC TBGL

Petr Schreiber
1.9.2009

Table of contents

About the file format

thinEdge M15 file format is an ASCII based container for geometry data. It can represent any 3D geometry using polygons, which have assigned texture, colour, normal and layer attributes. Files of this type can be generated using thinEdge modeller or other tools you can find on http://psch.thinbasic.com/. On this website you will also obtain the latest file format specification.

M15 file format was created for purpose of storing geometry data. At time of writing of this document files can be manipulated using mentioned tools and also via TBGL module included in thinBASIC interpreter which can be found at http://www.thinbasic.com/.

All programs meant as M15 generator, importer or exporter must support the format in scale specified in this document, otherwise they are considered incomplete implementations.

Description of the file format

M15 file format is an ASCII file, with line oriented storage of data. Data present on line are identified (with exception of first line in file) by token. The lines are delimited using CR and LF characters (ASCII code 13, 10).
The first line of file is reserved for version identification.

```
'thinEdge datafile {1.0.0.6}
```

```
'thinEdge datafile {1.0.0.7}
```

This file format currently exists in versions 1.0.0.6 and 1.0.0.7. The second one differs from first one only in including information about normal vectors.

The second line of file must contain information on used vertices, which is represented by identifier followed by comma and number of vertices.

```
VERTEXNUM,numberOfVertices
```

Sample for model with 1000 vertices:

```
VERTEXNUM,1000
```

Specified number of vertices can be equal or higher to real number of vertices in file.

The next line contains list of user defined layers used in model, in case there are some present. In case all vertices are in the same general layer, this line must not be present in file.

```
LAYERS,LayerName1,LayerName2,...,LayerNameN
```

Sample for layers “Door” and “Window”:

```
LAYERS,Door,Window
```

The layer names must be unique.

The next line must contain list of textures used in model. In case model does not use any texture, this line must not be present in file.

```
TEXLIST,TextureName1,TextureName2,...,TextureNameN
```

Sample for used textures a.bmp, b.tga:

```
TEXLIST,a.bmp,b.tga
```

Texture names must be without any path specifier, and can contain only names of BMP and TGA files. This limitation might me changed in future. Texture names must be unique.
Implementation of M15 importer should presume textures are situated in Textures directory.
Mentioned tokens must be on separate lines, and must be in following order:

```
VERTEXNUM, …
LAYERS, …
TEXLIST, …
```

The last two are optional as explained in previous paragraphs.

To define geometry M15 uses token POLY, which has 12 or 15 parameters, depending on file format version.

Syntax for single line in thinEdge M15 1.0.0.6 is:

```
POLY,x,y,z,stopFlag,textureID,U,V,R,G,B,layerID
```

Syntax for single line in thinEdge M15 1.0.0.7 is:

```
POLY,x,y,z,stopFlag,textureID,U,V,R,G,B,layerID,normalX,normalY,normalZ
```

The following table explains the meaning of the parameters. It is recommended to handle floating point parameters using double precision variables and integer parameters using 32 bit integers.

Parameter	**Type**	**Description**	**Present from version**
x	floating point	X coordinate of the vertex	1.0.0.6
y	floating point	Y coordinate of the vertex	1.0.0.6
z	floating point	Z coordinate of the vertex	1.0.0.6
stopFlag	integer	Polygon stop flag Can take only value of 0 or 1	1.0.0.6
textureID	integer	Positive integer number – order of texture in TEXLIST + 1 **Attention:** No texture is marked as 1, first as 2, …	1.0.0.6
U	floating point	Texture U mapping coordinate. Sometimes referenced as texX	1.0.0.6
V	floating point	Texture V mapping coordinate. Sometimes referenced as texY	1.0.0.6
R	byte	Red component of vertex diffuse color	1.0.0.6
G	byte	Green component of vertex diffuse color	1.0.0.6
B	byte	Blue component of vertex diffuse color	1.0.0.6
layerID	integer	Positive integer number – value 0 means general layer, 1 first user defined layer, … **Attention:** All vertices in same polygon must belong to same layer	1.0.0.6
normalX	floating point	X coordinate of vertex normal	1.0.0.7
normalY	floating point	Y coordinate of vertex normal	1.0.0.7
normalZ	floating point	Z coordinate of vertex normal	1.0.0.7

Sample files

Here are two sample M15 files, for better understanding of the format.

Triangle with no user layers, no texture and different color in each vertex

```
'thinEdge datafile {1.0.0.6}
VERTEXNUM,3
POLY,-1.000000, 0.000000, 0.000000, 0, 1,-1.000000, 0.000000,255,   0,   0,0
POLY, 1.000000, 0.000000, 0.000000, 0, 1, 1.000000, 0.000000,   0,255,   0,0
POLY, 0.000000, 1.000000, 0.000000, 1, 1, 0.000000, 1.000000,   0,   0,255,0
```

Remark: As you can see, UV coordinates are specified although it is not needed by specification.

Remark: There is no obligation to put redundant whitespace or precision in file. Model could look like this in ideal case:

```
'thinEdge datafile {1.0.0.6}
VERTEXNUM,3
POLY,-1,0,0,0,1,0,0,255,0,0,0
POLY,1,0,0,0,1,0,0,0,255,0,0
POLY,0,1,0,1,1,0,0,0,0,255,0
```

As you can see, without use of UVs, whitespaces and unimportant decimal places file will be much smaller. M15 generally provide very good compression ratio when packed by ZIP, but it is a good practice to minimize the size like in the example.

Model with user layers and textures, 2 triangles

```
'thinEdge datafile {1.0.0.6}
VERTEXNUM,6
LAYERS,Triangle1,Triangle2
TEXLIST,01.bmp,02.bmp
POLY,-1.000000, 0.000000, 0.000000, 0, 2,-1.000000, 0.000000,255,   0,   0,1
POLY, 1.000000, 0.000000, 0.000000, 0, 2, 1.000000, 0.000000,   0,255,   0,1
POLY, 0.000000, 1.000000, 0.000000, 1, 2, 0.000000, 1.000000,   0,   0,255,1
POLY,-1.000000, 0.000000,-1.000000, 0, 3,-1.000000, 0.000000,255,   0,   0,2
POLY, 1.000000, 0.000000,-1.000000, 0, 3, 1.000000, 0.000000,   0,255,   0,2
POLY, 0.000000, 1.000000,-1.000000, 1, 3, 0.000000, 1.000000,   0,   0,255,2
```

Remark: Identifiers were marked with color, so you can see which number refers to which layer / texture. You can also watch layer / texture numbering “paradox”.

Remark: Here the same, but more space optimized:

```
'thinEdge datafile {1.0.0.6}
VERTEXNUM,6
LAYERS,Triangle1,Triangle2
TEXLIST,01.bmp,02.bmp
POLY,-1,0,0,0,2,-1,0,255,0,0,1
POLY,1,0,0,0,2,1,0,0,255,0,1
POLY,0,1,0,1,2,0,1,0,0,255,1
POLY,-1,0,-1,0,3,-1,0,255,0,0,2
POLY,1,0,-1,0,3,1,0,0,255,0,2
POLY,0,1,-1,1,3,0,1,0,0,255,2
```

The files in 1.0.0.7 formats differ from 1.0.0.6 only in addition of the normal vector information.

Triangle with normals (format v1.0.0.7)

```
'thinEdge datafile {1.0.0.7}
VERTEXNUM,3
POLY,-1, 0, 0, 0, 1, 0, 0, 255,  0,  0,0, 0, 0, 1
POLY, 1, 0, 0, 0, 1, 0, 0,   0,255,  0,0, 0, 0, 1
POLY, 0, 1, 0, 1, 1, 0, 0,   0,  0,255,0, 0, 0, 1
```

M15 file generators

There are multiple approaches to obtain models in M15 file format.

Free converters

To generate models in version 1.0.0.6 and 1.0.0.7 of the formats you can use basically any application able to produce OBJ and MTL files. OBJ to M15 tool for the conversion, available free at http://psch.thinbasic.com/tools.html, will help you to automate the creation of M15 file.

This file format converter has been extensively tested with following programs.

Free modelers

Blender3D

Website: http://www.blender.org/
Blender M15 Exporter script by Michael Hartlef needed. You can get it at
http://psch.thinbasic.com/tools.html

ThinEdge

To load and edit files in version 1.0.0.6 of the format, you can use ThinEdge modeller. It is free, and can be downloaded from http://psch.thinbasic.com/thinEdge.html.

Wings 3D

Website: http://www.wings3d.com/
OBJ to M15 tool required for the file conversion.

Shareware programs:

Milkshape 3D

Website: http://chumbalum.swissquake.ch/
OBJ to M15 tool required for the file conversion.

Commercial programs

Rhinoceros

Website: http://www.rhino3d.com/
OBJ to M15 tool required for the file conversion.

Z-Brush

Website: http://www.pixologic.com/
OBJ to M15 tool required for the file conversion.

Future of the format

It is very probable there will be some additions to the format in the future, such as removing the limitation on supported image file formats for textures.
Watch http://psch.thinbasic.com/ for any news.