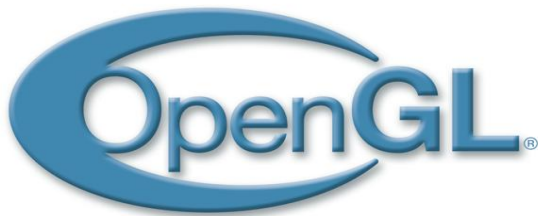


بسم الله الرحمن الرحيم و الصلاة و السلام على أشرف المرسلين نبينا محمد رسول الله صلى الله عليه و سلم و السلام عليكم يا إخوتي في الله

أقد لكم هذه الوثيقة المتواضعة و هي بداية لموضوع استيراد ملف الماكس فأتمنى لكم التوفيق

و أدعوا لي بخالص دعائكم بالرحمة من الله عز و جل



ملف الماكس:

يمثل برنامج الماكس أحد أهم البرامج المتفوقة في عالم التصميم الثلاثية الأبعاد , ففقر ترغب أحيانا لو أشركت ملف الماكس مع برنامجك الرسومي , قد تبحث عن مكتبات تحميل ملف الماكس و لكن الغيرة على كونك مبرمج ألعاب لن تترتاح في ذلك إذ أنك تريد أن تبرمج شيفرات تحميل ملف الماكس حسب ما تشاء و أن تتلاعب بها كيفما تريد و هذا الموضوع كتبته لذلك , إذن شد الرحال و للننطلق إلى نهاية الشعاع.

لدا الماكس إمتدادات عديدة و من أهمها 3DS

جدول لأهم الإمتدادات :

الإمتداد	المحرر أو المستعمل	الموقع الرسمي له
3DMF	3D Meta File, QuickDraw3D	http://www.apple.com
3DO	Jedi Knight	http://www.lucasarts.com
3DS	3D Studio Max: Binary-based format	http://www.discreet.com
ACT	Genesis 3D Engine	http://www.genesis3D.com
ASE	3D Studio Max: Essentially a textbased version of 3ds	http://www.discreet.com
ASC	3D Studio Max: Simple ASCII-based format containing only a minimum of data	http://www.discreet.com
B3D	Bryce 3D	http://www.corel.com
BDF	Okino	http://www.okino.com
BLEND	Blender	http://www.blender.nl
CAR	Carrara	http://www.eovia.com/carrara
COB	Calgari TrueSpace	http://www.calgari.com
DMO	Duke Nukem 3D	http://www.3drealms.com
DXF	Autodesk Autocad	http://www.autodesk.com
HRC	SoftImage 3D	http://www.softimage.com
INC	POV-Ray	http://www.povray.org
KF2	Max Payne animations and poses	http://www.maxpayne.com
KFS	Max Payne mesh and material information	http://www.maxpayne.com
LWLO	Lightwave layered file object	http://www.newtek.com
LWO	Lightwave object file	http://www.newtek.com
LWS	Lightwave scene file	http://www.newtek.com

MA	Maya ASCII-based format	http://www.aliaswavefront.com
MB	Maya	http://www.aliaswavefront.com
MAX	3D Studio Max	http://www.discreet.com
MD2	id Software's Quake II	http://www.idsoftware.com
MD3	id Software's Quake III	http://www.idsoftware.com
MDL	id Software's Quake	http://www.idsoftware.com
MDL	Valve's Half-Life	http://www.valvesoftware.com
MDL	Serious Sam	http://www.croteam.com
MS3D	Milkshape 3D format	http://www.swissquake.ch/chumbalum-soft
OBJ	Alias Wavefront	http://www.aliaswavefront.com
PSK	Unreal/Unreal Tournament Mes	http://www.unrealtournament.com/
PZ3	Poser	http://www.curioslabs.com
RAW	Raw Triangles	(لا يوجد)
RDS	Ray Dream Studio	http://www.metacreations.com
RIB	Renderman File	http://www.renderman.com
SCN	Calgari TrueSpace	http://www.calgari.com
SKD	Max Payne skin data file	http://www.croteam.com
SKN	The Sims	http://thesims.ea.com
SLP	Pro/Engineer	http://www.ptc.com
STL	Stereo-Lithography Format	http://www.sdsc.edu/tmfStl-specs/stl.html
VRML	Virtual Reality Modeling Language	http://www.web3d.org
X	Microsoft's Direct X format	http://www.microsoft.com
XGL	Various CAD Programs	http://www.xglspec.com

بنية ملف الماكس من الداخل:

لقد صمم ملف الماكس من الداخل بكشل هرمي أو بالأحرى كالشجرة تماماً فهو يحتوي على سلسلة من البيانات و تلك البيانات إعتبرها أوراق الشجرة , و كل ورقة لها دور خاص , حاوية على بيانات قد تكون رؤوس أو إحداثيات إكساء أو أوجه أو .. إلخ.

تخيل أن هناك نملة تريد الوصول إلى ورقة ما من الشجرة فبداية تبدأ من الجذع ثم تتسلق ساق الشجرة و ثم تعبر عبر الفروع لغاية وصولها إلى الورقة. هكذا سنكون نحن تماماً كما تفعل النملة.

الآن لنقل أن كل ورقة هي بحد ذاتها قطعة و كل فرع من فروع الشجرة هي قطعة أبطا و الساق و الجذع هما قطعتان أبطا , و الشجرة بأكملها هي الملف .

مفهوم القطع Chunks :

القطعة هي عبارة عن تركيب لبيانات و هذه البيانات تختلف من قطعة لأخرى و قد تحتوي قطعة ما على كم قليل من البيانات و قد تكون هناك قطع تحتوي على كم هائل من البيانات و قد تكون بشكل رهيب . ماذا تحتوي هذه القطع ؟ تحتوي كل قطعة على التالي:

- المعرف من الذي يمثل هوية القطعة كالرؤوس أو الخامات أو الإكساء أو المفتاح المبروز أو... إلخ.
- طول القطعة و أحيانا يكون مجموعة لقطع مشتقة من القطعة الأم بحجم 4 بايت (4 بايت هي حجم العدد الذي يمثل طول القطعة و ليس القطعة نفسها).
- بيانات القطعة و المتمثلة في إحداثيات الرؤوس و إحداثيات الإكساء و ... إلخ.

هذا الجدول يمثل التركيب العام للقطعة :

Offset	Length	
0	2	المعرف: Chunk identifier
2	4	الطول: Chunk length: chunk data + sub-chunks(6+n+m)
6	n	البيانات: Data
6+n	m	قطع جزئية: Sub-chunks

أنظر إلى الجزء الرابع من الجدول (القطع الجزئية) فالقطعة الأم تحتوي على العديد من القطع الجزئية و أيضا بعض القطع (و التي هي أيضا تكون عبارة عن قطعة جزئية من القطعة الأم) لديها قطع جزئية أيضا.

هذا المخطط التتابعي يمثل الشجرة في ملف الماكس:

```

MAIN3DS (0x4D4D)
|
+--EDIT3DS (0x3D3D)
|
|   +--EDIT_MATERIAL (0xAFFF)
|   |
|   |   +--MAT_NAME01 (0xA000)
|   |
|   +--EDIT_CONFIG1 (0x0100)
|   +--EDIT_CONFIG2 (0x3E3D)
|   +--EDIT_VIEW_P1 (0x7012)
|   |
|   |   +--TOP (0x0001)
|   |   +--BOTTOM (0x0002)
|   |   +--LEFT (0x0003)
|   |   +--RIGHT (0x0004)
|   |   +--FRONT (0x0005)
|   |   +--BACK (0x0006)
|   |   +--USER (0x0007)
|   |   +--CAMERA (0xFFFF)
|   |   +--LIGHT (0x0009)
|   |   +--DISABLED (0x0010)
|   |   +--BOGUS (0x0011)
|   |
|   +--EDIT_VIEW_P2 (0x7011)
|   |
|   |   +--TOP (0x0001)
|   |   +--BOTTOM (0x0002)
|   |   +--LEFT (0x0003)
|   |   +--RIGHT (0x0004)
|   |   +--FRONT (0x0005)
|   |   +--BACK (0x0006)
|   |   +--USER (0x0007)
|   |   +--CAMERA (0xFFFF)
|   |   +--LIGHT (0x0009)
|   |   +--DISABLED (0x0010)
|   |   +--BOGUS (0x0011)
|   |
|   +--EDIT_VIEW_P3 (0x7020)
|   +--EDIT_VIEW1 (0x7001)
|   +--EDIT_BACKGR (0x1200)
|   +--EDIT_AMBIENT (0x2100)
|   +--EDIT_OBJECT (0x4000)

```


قد لا يكون هذا المخطط كاملاً و لكن هو جزء من مخطط هائل , و لأن ملف الماكس يحتوي على كم هائل من القطع سنكتفي في هذه الوثيقة على

```
| +--EDIT_OBJECT (0x4000)
| |
| | +--OBJ_TRIMESH (0x4100)    TRIANGULAR MESH
| | |
| | | +--TRI_VERTEXL (0x4110)
| | | +--TRI_VERTEXOPTIONS (0x4111)
| | | +--TRI_MAPPINGCOORS (0x4140)
| | | +--TRI_MAPPINGSTANDARD (0x4170)
| | | +--TRI_FACEL1 (0x4120)
```

لتكون البداية إن شاء الله.

لو أنك أردت قراءة القطعة ذات التعريف 4110 فعليك قراءة القطعة الأم (0x4D4D) MAIN3DS ثم و التي بدورها لديها تفرع واحد وهي القطعة المتفرعة (0x3D3D) EDIT3DS و هاته الأخيرة لديها الكثير من القطع نختار واحدة منها و لتكن EDIT_OBJECT (0x4000) و هذه الأخيرة لديها الكثير أيضاً و نختار واحدة منها و هي

OBJ TRIMESH (0x4100)

ثم

TRI_VERTEXL (0x4110)

هذا الجدول يمثل تعريف بنيوي لما نحن بصدده :

MAIN CHUNK	
Identifier	0x4d4d
Length	0 + sub-chunks length
Chunk father	ليس لديه أم
Sub chunks	3D EDITOR CHUNK
Data	لا يحتوي بيانات
3D EDITOR CHUNK	
Identifier	0x3D3D
Length	0 + sub-chunks length
Chunk father	MAIN CHUNK
Sub chunks	OBJECT BLOCK, MATERIAL BLOCK, KEYFRAMER CHUNK
Data	لا يحتوي بيانات
OBJECT BLOCK	
Identifier	0x4000
Length	Object name length + sub-chunks length

Chunk father	3D EDITOR CHUNK
Sub chunks	TRIANGULAR MESH, LIGHT, CAMERA
Data	Object name
TRIANGULAR MESH	
Identifier	0x4100
Length	0 + sub-chunks length
Chunk father	OBJECT BLOCK
Sub chunks	VERTICES LIST, FACES DESCRIPTION, MAPPING COORDINATES LIST
Data	لا يحتوي بيانات
VERTICES LIST	
Identifier	0x4110
Length	varying + sub-chunks length
Chunk father	TRIANGULAR MESH
Sub chunks	ليس لديه قطع متفرعة منه
Data	Vertices number (unsigned short) Vertices list: x1,y1,z1,x2,y2,z2 etc. (for each vertex: 3*float)
FACES DESCRIPTION	
Identifier	0x4120
Length	varying + sub-chunks length
Chunk father	TRIANGULAR MESH
Sub chunks	FACES MATERIAL
Data	Polygons number (unsigned short) Polygons list: a1,b1,c1,a2,b2,c2 etc. (for each point: 3*unsigned short) Face flag: face options, sides visibility etc. (unsigned short)
MAPPING COORDINATES LIST	
Identifier	0x4140
Length	varying + sub-chunks length
Chunk father	TRIANGULAR MESH
Sub chunks	SMOOTHING GROUP LIST
Data	Vertices number (unsigned short) Mapping coordinates list: u1,v1,u2,v2 etc. (for each vertex: 2*float)

خطة العمل :

نحن بحاجة إلى مراعات الترتيب لتحميل البيانات من الملف بأي خطأ سيؤدي إلى التضارب و الإختلاط في البيانات .

تحميل الملف و إعداد المتغيرات أو بنيات (شكل مصفوفة) .

استعمال حلقة يعاد تنفيذها حتى نهاية الملف .

قراءة معرف و طول القطعة في كل حلقة .

فحص نوع القطعة .

يحتوي ملف الماكس على الكثير من القطع لتي قد لا تستعملها و لتجاهلها يكفي أن نفقر عليها عن طريق تحريك مؤشر الملف الحالي إلى موقع جديد يمثل القطعة الموالية لها وذلك عن طريق إضافة طول القطعة المراد تجاهلها مع الطول الحالي (الموقع الحالي للمؤشر) (ملاحظة : في حالة تجاهل قطعة ما و كان لهذه القطعة قطع فرعية فسيتم تجاهلها أيضا).

إذا ما تحقق الشرط (فحص نوع القطعة) نقوم بقراءة البيانات من هذه القطعة .

تفصيل الخطة:

نقوم بتعريف عدد كاف لعدد الرؤوس و المضلعات المحتملة.

```
#define MAX_VERTICES 8000
#define MAX_POLYGONS 8000
```

الإعلان عن بنية تمثّل سجل إحداثيات الرؤوس

```
typedef struct{
    float x,y,z;
    float norm[3];
}vertex_type;
```

الإعلان عن بنية تمثّل سجل للمضلعات

```
typedef struct{
    int a,b,c;
    float norm[3];
}polygon_type;
```

بنية لإحداثيات الإكساء

```
typedef struct{
    float u,v;
}mapcoord_type;
```

الإعلان عن بنية شاملة تمثّل الكائن أو الشكل أو الجسم

```
typedef struct {
    char name[20];
    int vertices_qty;
    int polygons_qty;
    vertex_type vertex[MAX_VERTICES];
    polygon_type polygon[MAX_POLYGONS];
```

```
mapcoord_type mapcoord[MAX_VERTICES];

int id_texture;

} obj_type, *obj_type_ptr;
```

ملاحظة : بإمكانك الإستغناء عن كل هذا و ذلك في ما يخص التعريفات و الإستعانة بالفئة vector من المكتبة STL.

------()-----

الانتقال إلى شيفرة دالة تحميل البيانات من ملف الماكس. هذه الدالة ستستقبل بارمترين فالأول هو مؤشر نحو بنية بيانات الكائن و الثاني هو إسم الملف الذي سيفتح. هذه الدالة تم تصميمها لترجع الرقم 1 في حالة نجاح العملية و الرقم 0 في حالة الإخفاق.

```
Char Load3DS (obj_type_ptr p_object, char *p_filename)
{
    int i ;
    FILE *l_file ;
    unsigned short l_chunk_id ;
    unsigned int l_chunk_length ;
    unsigned char l_char ;
    unsigned short l_qty ;
    unsigned short l_face_flags ;
```

المتغير الأول سنستعمله كعداد و المتغير الثاني هو مؤشر إلى ملف الماكس الذي سيفتح و الثالث هو تعريف القطعة و المتغير الرابع هو طول القطعة و الخامس مستخرج اسم الكائن (أو بالأحرى اسم الجسم) و السادس يمثل عدد الرؤوس أو المضلعات أو الأوجه أو .. إلخ. و المتغير السابع غرضه أن نحتفظ بمختلف المعلومات عن المضلع الحالي (مرئي أو غير مرئي .. إلخ) الذي يستعمله محرر الماكس لتصيير المشهد فهذا المتغير لا يهمنا هنا لأننا قلنا أنه خاص بالمحرر و لكن سنستعمله لغرض أن يتساير برنامجنا مع بيانات ملف الماكس (قلنا سابقا أن أي خلل في معرفة البنية الداخلية في ملف الماكس سيؤدي إلى تضارب في البيانات).

شيفرة فتح الملف

```
if ((l_file=fopen (p_filename, "rb"))== NULL) return 0;
```

إستعمال الحلقة يمكننا من عدم الخروج من ملف الماكس حتى قراءته كله

```
while (ftell (l_file) < filelength (fileno (l_file)))
{
```

الدالة ftell تستخبر عن موقع مؤشر ملف الماكس الحالي أي موقعه (إرجاع عدد البايتات من بداية الملف إلى الموقع الحالي لمؤشر ملف الماكس) .

الدالة filelength هي من أحد دوال المكتبةunistd.h ترجع عدد بايتات ملف الماكس.

fileno هذه الدالة ترجع واصف لمقبض ملف الماكس.

```
fread (&l_chunk_id, 2, 1, l_file);
fread (&l_chunk_length, 4, 1, l_file);
```

الشيفرة الأولى قراءة عنصر واحد من ملف الماكس بحجم 2 بايت و تحميلها في المتغير l_chunk_id و الذي يمثل تعريف

القطعة

الشفيرة الثانية بقراءة عنصر واحد بحجم 4 بايت و تحميلها في المتغير l_chunk_length و الذي يمثل طول القطعة.

```
switch (l_chunk_id)
{
    case 0x4d4d:
        break;
```

شفيرة الفحص :

ندخل إلى التعريف الأم في الحقيقة في تعريف الأم لا يوجد به أي بيانات فقط يتم المرور عليه لتجنب الخلط كما قنا سابقا فنحن كل ما يهمنا هو أي يمشي مؤشر ملف الماكس بشكل جيد من دون أخطاء و من دون أن يترك أي قطعة .

```
case 0x3d3d:
    break;
```

و كذلك في ما يخص القطعة المتفرعة من القطعة الأم فهي لا تحتوي على البيانات .

```
case 0x4000:
    i=0;
    do
    {
        fread (&l_char, 1, 1, l_file);
        p_object->name[i]=l_char;
        i++;
    }while(l_char != '\0' && i<20);
    break;
```

هنا في هذه القطعة توجد بيانات مهمة و هي إسم الكائن أو المجسم فنقوم بتخزينه في المتغير نايم (طول الاسم و ضعناه لكي لا يتعدى 20 حرفا و بإمكانك تغيير ذلك و بدون تقييد).

```
case 0x4100:
    break;
```

هذه القطعة لا تحتوي على بيانات فقط لأنها أم أيضا و لديها قطع متفرعة هي :

VERTICES LIST و FACES DESCRIPTION و MAPPING COORDINATES LIST

```
case 0x4110:
    fread (&l_qty, sizeof (unsigned short), 1, l_file);
    p_object->vertices_qty = l_qty;
    printf("Number of vertices: %d\n",l_qty);
    for (i=0; i<l_qty; i++)
    {
        fread (&p_object->vertex[i].x, sizeof(float), 1, l_file);
        fread (&p_object->vertex[i].y, sizeof(float), 1, l_file);
        fread (&p_object->vertex[i].z, sizeof(float), 1, l_file);
    }
    break;
```

الدخول في تعريف الرؤوس هذا يدل على أن القطعة الحالية تحتوي بيانات خاصة عن الرؤوس. فبداية نقرأ عدد الرؤوس ونحتفظ بها في المسجل vertices_qty.

و بعدها حلقة تتكرر على حسب عدد الرؤوس و من ثم حفظ كل متغير في سجل البنية.

```
case 0x4120:
    fread (&l_qty, sizeof (unsigned short), 1, l_file);
    p_object->polygons_qty = l_qty;
    printf("Number of polygons: %d\n",l_qty);
    for (i=0; i<l_qty; i++)
    {
        fread (&p_object->polygon[i].a, sizeof (unsigned short), 1, l_file);
        fread (&p_object->polygon[i].b, sizeof (unsigned short), 1, l_file);
        fread (&p_object->polygon[i].c, sizeof (unsigned short), 1, l_file);
        fread (&l_face_flags, sizeof (unsigned short), 1, l_file);
    }
    break;
```

الدخول إلى القطعة الخاصة بأوجه المضلعات , فنقرأ عدد أوجه المضلعا (وجه المضلع هو بحد ذاته مضلع و هو مركب من ثلاثة رؤوس بشكل مثلثي إضافة إلى بعض خواص وجه هذا المضلع المثلثي).

```
case 0x4140:
    fread (&l_qty, sizeof (unsigned short), 1, l_file);
    for (i=0; i<l_qty; i++)
    {
        fread (&p_object->mapcoord[i].u, sizeof (float), 1, l_file);
        fread (&p_object->mapcoord[i].v, sizeof (float), 1, l_file);
    }
    break;
```

الدخول إلى القطعة ذات التعريف 4140 (قائمة إحداثيات التخطيط), و قراءة أول متغير من هذه القطعة و هو عدد أزواج إحداثيات التخطيط.

```
default:
    fseek(l_file, l_chunk_length-6, SEEK_CUR);
}
}
```

حين الدخول إلى التعريف (default:) ديفولت فإنه يدل على أن القطعة الحالية ليست من القطع التي خصصناها.

الدالة fseek تقوم بتحريك المؤشر إلى موضع جديد من الملف (إلى الأمام بطبيعة الحال)

هذه الدالة تقوم بتحريك المؤشر ابتداء من موقع المؤشر الحالي نحو الأمام بحجم طول القطعة الحالية ناقص حجم تعريف القطعة و الذي يساوي 2 و العدد اذي يمثل حجم القطعة الذي هو بحجم 4 .

```
fclose (l_file); // Closes the file stream
return (1); // Returns ok
}
```

و في النهاية نقوم بغلق ملف الماكس و إرجاع القيمة 1 دلالة على نجاح العملية.

على فكرة مازال هنالك الكثير من القطع مثل الخامات و التحريك و التضليل و الكاميرا و الإنارة و .. إلخ , هذا يتطلب فهم لبنية ملف الماكس جيدا , سيتم دعم هذه الوثيقة بإصدارات متجددة و في كل مرة سأحاول إضافة قطعة جديدة إن شاء الله و هذه الوثيقة مرخصة للجميع من حيث الإضافة أو النشر أو النقد و التحليل .

وصلة المشروع كاملة

http://www.spacesimulator.net/tutorials/source_code/spacesimulator.net-tutorial4.zip

و هذا ناتج المشروع



مواقع ذات صلة

<http://www.spacesimulator.net>

<http://www.morrowland.com>

<http://www.the-labs.com/Blender/3DS-details.html>

<http://www.martinreddy.net/gfx/3d/>