

C3ds

a code resource
to read 3ds files

Introduction

C3ds is a program written in C++ to make easier the construction of applications that need to import 3d models using the well-known 3ds file format.

Code features

The code should be totally OS-Independent since it was programmed exclusively using the C++ standard library, so it can run either under Windows either under Linux, MacOS or any operating system under which exists a compiler that provides a support for the Standard C++ .

The whole structure is Object Oriented, classes concerning reading of 3ds files are two:

- C3dsSFileFormat
 - C3dsFileLoader

The former, C3dsSFileFormat contains information regarding chunks and some data structure used in 3ds files, the latter, C3dsloader, is a derivation of the previous so englobes the same information about data structures used to parse the file.

The visualization system is even Object Oriented and based, in this actual shape, on Glut, an absolutely OS and Hardware independent platform built on top of OpenGL. This part should too be good even for various Operating Systems (and anyway, if this is not exactly true, minimal changes are required to make a conversion).

To run the demo it's necessary install the DLL shared with the glut distribution package that you can find at the site www.opengl.org

With this intention I programmed a little Wrapper to build generic applications inspired to the excellent GlutMaster.

How it works

The mechanism is very simple. The "kernel" of the application is the member function `ParseChunk()` that calls recursively itself every time that the reading of a chunk is required and while there are still chunks available. When a chunk is found and recognized the routine calls the specific (private) function to extract information from file. This routine itself calls a virtual function that informs the user application of the actual chunk contents.

In this way you can create your own class inheriting from `C3dsFileLoader` and specializing it as you require, something like this:

```
class MyClass : public C3dsFileLoader
```

`MyClass` has simply to manage various information elaborated by `C3dsFileLoader` that are communicated transparently through the virtual functions which name starts with "User".

Here an example:

```
class MyClass : public C3dsFileLoader
{
protected:
void User3dVert(float x, float y, float z);
};
```

`User3dVert` is used roughly as a "callback" function, infact it is called every time that a Vertex is found. The same is valid for other virtual functions (which name starts with the word "User").

How can I use it?

The usage is very simple owing to the OOP design. It's simply necessary inherit the class of the new application from `C3dsFileLoader`, as above. Example:

```

class C3DModel : public C3dsFileLoader
{
public:
    C3DModel() : C3dsFileLoader() {}
protected:
    void        User3dVert(float x, float y, float z);
    void        User3dFace(unsigned short A, unsigned short B,
                          unsigned short C, unsigned short Flags);
    void        UserCamera(float px, float py, float pz,
                          float tx, float ty, float tz,
                          float Bank, float Lens);
    void        UserMapVertex(float U, float V);
    void        UserTransformMatrix(const Transform3dsMatrix &Transform,
                                    const Translate3dsMatrix &Translate);
    void        UserChunkObj();
    void        UserEndOfFile();
};

```

It's important that the constructor of the proper class calls the one of C3dsFileLoaders, so that initializations are made automatically at the start. To create your own program you simply have to override the functions which name starts with the "User" word inserting your specific code. For example it's possible use an "User" function to add a vertex in a STL vector:

```

typedef struct _Vertice
{
float x, y, z;
} Vertice;

class C3DModel : public C3dsFileLoader
{
public:
    C3DModel() : C3dsFileLoader() {}
private:
    vector <vertice>Vertici;
protected:
    void User3dVert(float x, float y, float z);
};

void C3DModel::User3dVert(float x, float y, float z)
{
    Vertice v;
    v.x = x; v.y = y; v.z = z;
    Vertici.push_back(v);
}

```

The same speech is valid for other functions which name starts with "User", classes are customizable as you need.

For a complete overview i suggest you to examine the source code related to this document, all will appear more clear than how it can be now.

The demo

If you compile and link sources and execute the object file you can see an implementation of the C3dsFileLoader class. It is the 3d model of a rotating object. It's possible choose a scale factor to zoom it (with "+" and "-" keys). For the future (if i shall find the time) i'm going to add a direct management and usage of viewports or cameras if present in the file (they are always read but not used by this demo).

Conclusion

I wrote this document and related source code without any particular pretension but with a purpose only: try to be useful to myself and others. For this reason i encourage you to improve the contents of my code (and this manual!) if you find a bug (that surely exists) or you just wish to add a new feature.

In this package you can find even a Gnu/Linux version created with Anjuta. I hope that my work can be useful. In next versions, if i shall find the opportunity, i'll program new functionalities, meanwhile i wish you a good programming!

Andrea Ingegneri

Home Page: <http://www.tsrevolution.com/>

This manual was written using Kile and L^AT_EX

Disclaimer

C3ds is a set of C++ sources, released following the GNU LGPL (Lesser General Public License). Please, visit <http://www.gnu.org> to obtain more information.

This document is released following the GNU FDL (Free Document License). To make easier the modification of this document it's possible to use the .tex source present in this package.

C3ds is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.