

C3ds

una risorsa di codice
per leggere i files 3ds

Introduzione

C3ds è un programma in C++ scritto per facilitare la realizzazione di applicazioni che necessitano di importare modelli tridimensionali tramite il noto formato 3ds.

La licenza adottata per C3ds è la GNU LGPL che permette di inserire il codice sorgente C++ del programma all'interno di applicazioni anche commerciali.

Alcune caratteristiche del codice

Il codice dovrebbe essere totalmente OS-Independent essendo stato realizzato esclusivamente ricorrendo alla libreria standard C++, diciamo quindi che può andare bene sia sotto Windows sia sotto Linux, MacOS o qualsiasi sistema operativo sotto il quale esista un compilatore che offra qualche supporto al C++ Standard.

Il tutto è Object Oriented, le classi riguardanti la lettura dei files 3ds veri e propri sono due:

- C3dsFileFormat
 - C3dsFileLoader

La prima contiene informazioni riguardanti i chunk ed alcune strutture dati a impiego esclusivo dei files 3ds, C3dsloader è una derivazione di questa classe e quindi ingloba al suo interno le stesse strutture dati che sono adottate per elaborare il file.

Il sistema di visualizzazione è anch'esso Object Oriented e basato, in questa sua forma attuale, su Glut, una piattaforma assolutamente OS independent costruita su OpenGL. Anche questa parte dovrebbe andare bene per altri sistemi operativi (se comunque ciò non fosse, poiché ho provato su pochi sistemi, ritengo che sarebbero necessarie pochissime modifiche per rendere compatibile il codice). Per fare funzionare il dimostrativo è necessario installare i DLL di glut reperibili dal sito www.opengl.org. A questo proposito ho realizzato un semplice Wrapper per costruire le applicazioni ispirandomi all'ottimo GlutMaster.

Nota: alcune parti del codice sono realizzate in modo rigido, senza ricorrere alla definizione di strutture, questo particolarmente durante la lettura dei chunk. Si è voluto fare così perché il formato 3ds ormai è sufficientemente obsoleto e si ritiene che i chunk rimarranno come sono ancora per svariato tempo.

Come funziona

Il funzionamento è fondamentalmente semplice. Il nucleo dell'applicazione è la funzione membro `ParseChunk()` che richiama se stessa ricorsivamente ogni volta che viene terminata la lettura di un `Chunk`, fino a che questi non sono terminati. Quando un chunk viene trovato e riconosciuto la routine richiama l'apposita funzione (privata) per l'estrazione delle informazioni dal file. La routine a sua volta richiama una funzione virtuale che si occupa di fornire all'utente il contenuto del chunk.

In questo modo la classe è ereditabile ed è quindi possibile realizzare un qualsiasi tipo di programma che ne faccia impiego tramite polimorfismo. In termini pratici è semplicemente necessario derivare la propria classe da `C3dsFileLoader`, in questo modo.

```
class MiaClasse : public C3dsFileLoader
```

Mia classe si dovrà occupare di gestire le informazioni che vengono elaborate tramite `C3dsFileLoader` e che gli vengono comunicate attraverso le funzioni virtuali `User`.

Faccio un esempio:

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

`User3dVert` viene utilizzata grossomodo come una funzione callback, viene richiamata ogni volta che si presenta un Vertice. Lo stesso vale per tutte le altre funzioni virtuali (che iniziano sempre per `User`).

Come si usa?

L'impiego è davvero semplice grazie alla struttura OOP. È necessario derivare la classe della propria applicazione da `C3dsFileLoader`, come visto nel paragrafo precedente.

Ecco un esempio:

```

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();
};

```

È importante che il costruttore della propria classe richiami a sua volta quello di C3dsFileLoader, in modo che le varie allocazioni avvengano automaticamente. Non occorre fare altro che definire le funzioni che iniziano per User inserendo ciò che si desidera. Per esempio è possibile fare che User aggiunga il vertice che gli viene segnalato in una lista:

```

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);
}

```

Lo stesso discorso è valido per tutte le altre funzioni che iniziano per User, sono adoperabili come meglio si crede.

Per ulteriori dettagli vi consiglio di dare un'occhiata al codice allegato a questo documento, vi apparirà tutto più chiaro di quanto non possa sembrare ora.

Il dimostrativo

Se compilate e linkate i sorgenti ed eseguite l'oggetto potrete vedere un esempio di implementazione pratica della classe C3dsFileLoader. Si tratta di un modello 3d in rotazione. È possibile scegliere un fattore di scala per ingrandirlo o rimpicciolirlo. In futuro vedrò di inserire la gestione diretta delle inquadrature se presenti nel file (i cui chunk per il momento vengono letti ma non adoperati).

Il tutto dovrebbe essere compatibile con un gran numero di sistemi operativi non essendo effettuata alcuna chiamata a funzioni specifiche di sistema. L'API grafica impiegata è OpenGL mentre il sistema per la gestione delle interfacce utente è Glut. È possibile reperire informazioni e librerie al sito www.opengl.org

Conclusioni

Ho realizzato questo documento e i sorgenti allegati senza alcuna pretesa particolare ma con uno scopo specifico: cercare di essere di aiuto agli altri e a me stesso. Per questo motivo vi invito a migliorare il contenuto dei miei sorgenti nel caso troviate un bug (sicuramente ne esistono) o vogliate introdurre delle migliorie. Spero davvero che C3ds possa essere utile.

In questa distribuzione è disponibile una versione per gnu/linux scritta con Anjuta. Nelle prossime versioni, se si presenterà l'occasione saranno inserire nuove funzionalità, nel frattempo auguro buon lavoro!

Andrea Ingegneri

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

Documento realizzato con Kile e L^AT_EX

Disclaimer

C3ds è un insieme di sorgenti C++, rilasciati secondo le specifiche della LGPL (Lesser General Public License). È possibile ottenere informazioni dettagliate a tale proposito al sito www.gnu.org

Questo documento invece segue le norme della GNU FDL (Free Document License). Per facilitare l'opera di miglioramento del presente testo, nella distribuzione è inserito il relativo sorgente .tex

C3ds è distribuito nella speranza che possa essere utile, ma senza alcuna garanzia. L'autore o altre persone legate allo sviluppo di C3ds non sono da ritenersi responsabili delle conseguenze derivanti da un suo qualunque uso.