

## **Blood 2 Goodies**

A cura di **Claudio Todeschini**  
[keiserxol@galactica.it](mailto:keiserxol@galactica.it)

Carissimi, è con estremo piacere che vi presento un paio di gadget per lo sparattutto della Monolith/GT Interactive, ovvero i tool per l'editing dei livelli e dei modelli del gioco, nonché l'INTERO codice sorgente.

- I tool di editing sono quelli necessari per la realizzazione di livelli aggiuntivi, nuovi modelli e personaggi. Per utilizzare tali strumenti è necessario possedere il gioco originale, dato che non sono ovviamente inclusi i suoi 'mattoni' costituenti, ovvero i suoni, i modelli, le texture, ecc.

- Il codice del gioco può interessare invece agli 'smanettoni', per andarsi a guardare come è stato realizzato il LithTech Engine, il motore grafico del gioco, farsi un'idea di come si realizza un titolo di questo calibro, nonché provare magari a realizzare qualcosa. Giusto per curiosità, il LithTech è lo stesso engine che muove anche Shogo: Mobile Armor Division.

**Nota bene:** non è sufficiente compilare il codice per riottenere l'intero Blood 2 completo e funzionante (cosa per la quale servono sia un minimo di competenza che alcune librerie presenti solo nel DirectX SDK). Sono necessari, ovviamente, tutti gli elementi veri e propri che lo costituiscono: texture, modelli, suoni, mappe, ecc., e che per ovvi motivi di spazio e copyright non possiamo includere.

### **Link Utili:**

Sito ufficiale del gioco

<http://www.the-chosen.com>

Comunità online

<http://www.planetblood.com/>

### **Blood 2 Tools**

Il pacchetto è composto da due utility, DEdit e ModelEdit.

#### **DEdit**

Il primo di essi serve a creare il mondo di gioco, ovvero livelli e mappe. Vi accorgete, lanciando DEdit, che le directory di lavoro sono vuote. Questo perché dovrete, in prima istanza, estrarre dal gioco originale le cosiddette risorse, ovvero tutti i suoi componenti singoli: texture, suoni, modelli, ecc. Come in quasi tutti gli sparattutto in soggettiva, tali oggetti si trovano all'interno di un unico grosso file (in Doom erano i famosi WAD), che hanno estensione .REZ. Nella cartella BIN si trova anche il programma 'LithRez.exe', che funziona a linea di comando, e il cui scopo è esattamente quello di 'decompattare' il grosso file .rez, restituendo i singoli componenti che contiene.

Utilizzando questo programma dovrete quindi spacchettare in bell'ordine il file 'BLOOD2.REZ', 'BLOOD2P.REZ' e 'BLOOD2P2.REZ' in una sottocartella del gioco, chiamata a sua volta BLOOD2. Mi raccomando di decomprimere tutto nell'esatto ordine in cui sono elencati i file: questo perché al loro interno ci sono alcuni oggetti con lo stesso nome, ed è necessario che quelli contenuti negli ultimi file sovrascrivano quelli presenti nel primo.

A questo punto potete cancellare i file .rez originali e lanciare il gioco con la stringa di comando 'client.exe -rez BLOOD2'; quando creerete i vostri add-on, che avrete cura di riporre in una cartella (sempre all'interno di quella del gioco) di nome diverso, ad esempio ADDON, basterà aggiungerla alla riga di comando precedente. Nel nostro esempio:

```
client.exe -rez BLOOD2 -rez ADDON
```

Al termine del lavoro, testato e funzionante, potete riconvertire la directory in un altro file .REZ e distribuirla, mandarla a noi di TGM (per l'angolo dei wads), ecc.!!!

La realizzazione di un livello procede per tre passi ben distinti:

+ *reperire i file che necessari*

Per realizzare il livello farete uso di modelli, suoni, sprite e texture. Ad eccezione degli sprite, tali risorse non sono realizzabili con DEdit, ma devono essere importate dal gioco stesso (vedi sezione precedente); alcune di esse possono, all'interno dell'editor, essere modificate.

+ *disegnare il livello*

Dovrete definire la geometria della mappa e la dislocazione dei singoli oggetti. La geometria è composta dalle varie superfici e dai solidi che realizzate. I giocatori si muovono all'interno di questa geometria e incontrano gli oggetti che voi inserite al suo interno.

+ *compilarlo e testarlo*

Prima di poter lanciare un livello con il LithTech dovrete generare un file .dat a partire da quello .ed utilizzato dall'editor. In un file .dat, molta dell'informazione necessaria per visualizzare la mappa è precalcolata (ad esempio, gli effetti di luce). Questo dovrebbe consentire al LithTech di offrire il massimo frame rate possibile.

#### **ModelEdit**

Questo programma serve per modificare le impostazioni dei file dei modelli, che hanno estensione .abc. Per crearne di nuovi avrete bisogno di programmi di modeling tridimensionale come 3D Studio.

## **Extra**

Nella cartella BIN c'è una sottodirectory chiamata PLUGINS, contenente delle estensioni per altri programmi commerciali. Ce n'è uno per creare modelli per il LithTech utilizzando 3Dstudio Max; un altro, sempre per 3DSMax, per usare le texture del LithTech (estensione .dtx); c'è poi un convertitore a linea di comando da formato 3Dstudio (non Max) ad .abc (modelli LithTech); infine, un plug-in per Photoshop che permette di salvare e caricare immagini in formato .dtx, le texture di LithTech.

## **Blood 2 Source Code**

### *Compilazione del codice.*

Dopo aver compilato i vostri file CSHELL.DLL, CRES.DLL, e OBJECT.LTO, dovete metterli in una nuova directory, cui darete il nome del vostro progetto. La sezione SRES per Blood2 è un file 'fasullo', che non dovete ricompilare; utilizzate invece il file SRES.DLL che si trova all'interno del file REZ di Blood2.

### *Debug del codice*

Per aiutarvi nella fase di debug, è stata inserito il renderer de ILithTech SOFTDIB.REN. Si trova nella cartella in cui avete installato il pacchetto del codice sorgente. Copiate banalmente questo file .ren nella directory in cui si trova il file eseguibile del gioco, 'client.exe', e vi sarà reso disponibile dal gioco stesso.

### *Directory*

#### APPHEADERS

Questa cartella contiene gli header per il LithTech. Deve essere aggiunta nella sezione INCLUDE del compilatore, nelle impostazioni del progetto.

#### CLIENTRES

Questa cartella contiene il codice per il file cres.dll file. Viene usato per la localizzazione. È una buona idea memorizzare qui l'eventuale testo che andate ad inserire, così che il codice del gioco non debba essere modificato quando si localizzano le stringhe di testo, ma solo questa sezione.

#### CLIENTSHELLDLL

Qui sono invece contenuti i sorgenti del file cshell.dll file. Si tratta del lato client del gioco: HUD, effetti speciali, e tutta la roba che riguarda il client.

#### MISC

Al suo interno di trovano file necessari per compilare e linkare tra loro diversi progetti. Questa directory deve essere aggiunta alle sezioni LIBRARY e INCLUDE (o attraverso le impostazioni del progetto o del compilatore).

#### OBJECTDLL

In questa cartella c'è il codice del file object.lto, relativo al lato server. Contiene inoltre tutte le definizioni di oggetti che potete caricare con DEdit

#### SHARED

In questa cartella sono contenuti gli header condivisi tra client e server; dovrebbe essere aggiunta alla sezione INCLUDE.

#### BLOOD2SERV

È il codice del server dedicato. Sembra incredibile, ma è davvero lui. Forse alla Monolith sono impazziti, o forse gli sta davvero a cuore la comunità dei MOD!!!

Nota: per compilare il codice Lobby per DirectPlay nella Clientshell dovete possedere il file DPLobby.h, parte del SDK di DirectX, che non può essere distribuito, ma solo ottenuto da Microsoft medesima.