

# USING MESH

(written by Guardian)

*...by moonlight we ride*

*ten thousand side by side... (Manowar – Battle Hymn, 1982)*

## 1 - INTRODUZIONE

Le Mesh sono strutture per rappresentare in memoria geometrie complicate, generalmente superfici convesse chiuse di forma qualsiasi, sotto forma di insiemi di vertici e delle relazioni che intercorrono tra essi: ogni modello od oggetto in un mondo 3D può essere rappresentato mediante una Mesh.

In questo tutorial non ci occuperemo della creazione delle Mesh ma “semplicemente” del loro utilizzo e di come vengono “tradotte” in buffer di dati dalle DirectX<sup>1</sup>. In questo tutorial verranno utilizzate mesh non ottimizzate, il che significa che la geometria del modello verrà descritta “per esteso” in termini di vertici, con il risultato che alcuni dati potranno apparire duplicati e ridondanti.

Le funzioni presentate andranno modificate se si desidera adottare una gestione avanzata che impieghi mesh ottimizzate.

Le seguenti sezioni riguarderanno quindi il caricamento di una mesh a partire da un file .x<sup>2</sup>, la visualizzazione su schermo ed infine una sezione che spiega in modo piuttosto approfondito come accedere agli elementi della Mesh.

A corredo del tutorial viene fornita una classe C++ che incorpora tutte le funzioni qui presentate, la medesima classe seppur espansa in alcune delle sue funzionalità viene utilizzata per effettuare il loading e la gestione dei modelli nel demo **ASTEROID**<sup>3</sup> che viene utilizzato come “terreno di test” per i nostri tutorial su DirectX.

Per una migliore comprensione si consiglia di seguire il tutorial facendo riferimento al codice sorgente accluso; del codice verranno commentate solamente le parti ritenute più importanti.

## 2 - COME CARICARE UNA MESH

Questa operazione viene eseguita nel costruttore della classe. La prima cosa che ci occorre è un puntatore ad un buffer, in modo da riempirlo con i dati relativi a materiali e texture. Dichiariamo quindi un puntatore ad una interfaccia **ID3DXBuffer**.

```
LPD3DXBUFFER pD3DXMtrlBuffer;
```

LPD3DXBUFFER è definito infatti come: typedef struct ID3DXBuffer \*LPD3DXBUFFER negli header files.

Carichiamo quindi la mesh ricorrendo alla funzione:

```
HRESULT D3DXLoadMeshFromX(
    LPSTR pFilename,
    DWORD Options,
    LPDIRECT3DDEVICE8 pDevice,
```

<sup>1</sup> Per una definizione più accurata di Mesh e della relativa teoria fare riferimento a: Computer Graphics: Principles and Practice in C 2/e di James D. Foley, Andries van Dam, Steven K. Feiner, John F. Hughes.

<sup>2</sup> Il formato di file proprietario di microsoft di cui viene fornita documentazione nell'SDK delle DirectX8 a cui si rimanda per maggiori informazioni.

<sup>3</sup> Nome del progetto soggetto a continui cambiamenti.