

Introduzione a Blender

“Blender is the free open source 3D content creation suite, available for all major operating systems under the GNU General Public License”, ovvero “Blender è la suite libera e open source per la creazione di contenuti 3D, disponibile per i più importanti sistemi operativi, sotto licenza GNU GPL”.

Questa affermazione si trova nella prima pagina del sito ufficiale di Blender (www.blender.org) e racchiude tutti gli spunti di cui faremo uso durante questa introduzione al manuale.

1

Contenuti 3D

Questo concetto indica in modo generico tutti i prodotti che si possono ottenere con un software di grafica tridimensionale. La storia dei contenuti 3D inizia circa negli anni Sessanta, quando William Fetter, un designer della Boeing, conia il termine *computer graphic*. A quel tempo i concetti di computer e software grafico erano decisamente diversi da quelli che abbiamo al giorno d'oggi: anche se i contenuti 2D (planimetrie, schemi e disegni tecnici) potevano essere in qualche modo rappresentabili, non era ancora giunto il momento per i contenuti 3D.

Nei due decenni successivi, grazie ai rapidi tempi di sviluppo delle tecnologie informatiche, numerosi eventi portarono i computer a essere in grado di presentare qualcosa di più interessante di qualche linea immobile sullo schermo, suscitando così un interesse al di fuori della ristretta cerchia di ricercatori e ingegneri, all'interno della quale tutto questo era nato. Verso la fine degli anni Ottanta alcuni progettisti compresero le potenzialità di questo mezzo e si adoperarono per utilizzarlo con scopi diversi dalla progettazione o visualizzazione. È in questo periodo che avviene il primo contatto tra la computer grafica e il mondo dell'intrattenimento.

Gli anni Novanta segnarono infine la svolta definitiva nell'impiego della computer grafica, con la produzione dei primi lungometraggi animati completamente in 3D, la comparsa dei primi videogiochi e l'impiego massiccio del computer per la generazione di effetti speciali nelle pellicole cinematografiche.

Al giorno d'oggi comprendiamo facilmente il concetto di computer grafica, o contenuto 3D, in quanto ne siamo continuamente a contatto (film di animazione, visualizzazioni di progetti e prototipi di prodotti, pubblicità). Questo contatto prolungato ha portato tra l'altro alla nascita di un atteggiamento di sufficienza nei confronti della materia. L'affermazione "fatto al computer" è piuttosto emblematica e costituisce uno spunto per riflettere su cosa può o non può fare un computer senza input umano.

Grazie alla computer grafica, nell'ultimo decennio sono state raccontate nuove storie, si sono esplorati nuovi mondi e raggiunti livelli sempre più elevati di complessità e realismo. Le industrie del cinema e della multimedialità sono in pieno sviluppo, e Blender è uno degli strumenti che rende tutto ciò possibile.

Libero e open source

Tralasciando momentaneamente le implicazioni artistiche relative alla computer grafica, torniamo alla definizione iniziale di Blender, che viene definito come software libero e *open source*.

Capiamo innanzitutto il significato di *open source*; per farlo, è necessario avere un'idea di base del modo in cui viene creato un programma per computer. Non si tratta di un processo estremamente complicato: una persona scrive un programma, ovvero una serie di istruzioni per un calcolatore, affinché esegua un compito preciso, per esempio richiedere l'inserimento di due valori numerici da parte dell'utente e farne la somma. Per fare questo utilizza una sintassi e un lessico propri di uno specifico *linguaggio di programmazione*. A questo punto le istruzioni devono essere rese comprensibili al computer che le deve eseguire, cioè devono essere "tradotte" in linguaggio macchina. L'operazione di traduzione viene svolta da uno speciale programma detto *compilatore*.

Dal momento in cui esce dal compilatore, il programma sarà “leggibile” da un computer, ma non più da un essere umano. Poiché il processo di compilazione non è reversibile, se un essere umano volesse conoscere il funzionamento di un programma dovrebbe leggerne la versione scritta con il linguaggio di programmazione, ossia il codice sorgente. I motivi per cui un altro essere umano voglia leggere un programma scritto da qualcun altro sono molteplici e spaziano dal desiderio di contribuire allo sviluppo e al miglioramento del software, alla ricerca di problemi strutturali nel codice, che possono renderlo instabile o pericoloso per il sistema che lo ospita.

Se i sorgenti non vengono resi disponibili da chi distribuisce il software, non sarà possibile capirne esattamente le funzionalità, che possono essere facilmente limitate dall'autore e attivabili, per esempio con l'inserimento di un numero di serie da acquistare insieme al software. È quindi chiaro che lo scopo di rendere un software chiuso è di tipo commerciale. I software di questo tipo vengono chiamati “proprietary” e la loro resa economica proviene direttamente dal numero di copie vendute.

La controparte di questo modello è costituita dai software *open source*, il cui codice sorgente è aperto, visibile, e la cui resa deriva in parte dalla vendita di copie del software, ma soprattutto dall'assistenza e dal supporto forniti all'utente dopo l'acquisto.

Questo confronto fa sembrare il modello del software proprietario ben poco interessante rispetto all'*open source*, ma è necessario precisare, riguardo quest'ultimo, che la possibilità di leggere il codice sorgente di un programma non comporta automaticamente il diritto di copiarlo, modificarlo o ridistribuirlo.

Questi aspetti sono caratteristici di un terzo modello, concepito e sostenuto da Richard Stallman, uno sviluppatore software divenuto attivista informatico, che nel 1983 lanciò il progetto GNU (*GNU is Not Unix*) per la creazione di un sistema operativo il cui codice sorgente fosse aperto (a differenza dei sistemi esistenti allora, che erano “blindati”), dando vita a quello che è chiamato software libero. Il software libero viene distribuito al pubblico con una licenza che concede all'utente vari diritti:

- eseguire il programma, per qualsiasi scopo;
- studiare il funzionamento del programma e, se lo desidera, applicargli modifiche (af-finché ciò avvenga è necessario poter accedere al codice sorgente del programma);
- distribuire copie del software;
- modificare il programma e ridistribuirlo, a patto di ridistribuire anche il codice sorgente modificato.

Queste sono le cosiddette quattro libertà fondamentali del software libero e sono espresse in un testo legale, che va allegato al software che si desidera rilasciare in questo modo. Il nome di questo testo è GPL (*General Public Licence*). Un software può essere *open source*, ma non necessariamente libero (rilasciato sotto GPL), mentre viceversa, per essere libero deve essere *open source*.

Se il principio di utilità economica del software *open source* si basa ancora (parzialmente) sulla vendita di copie (e sulla possibilità di vietare la copia non autorizzata del codice sorgente), il software libero azzerava teoricamente questa possibilità, puntando tutti i fattori di guadagno su aspetti come l'assistenza, la formazione, le donazioni (se il software è veramente libero, diventa sensato fare una donazione per supportarne lo sviluppo, in quanto i benefici ricadranno sul numero di persone più alto possibile); il software verrà quindi reso disponibile gratuitamente, di solito su Internet. Blender è un software libero,

quindi è gratuito. La gratuità è importante, ma non fondamentale: ciò che conta è la libertà.

La lingua inglese

La prima frase del capitolo è stata presentata in lingua originale per anticipare un argomento importante: l'uso di una lingua piuttosto che un'altra per definire i termini tecnici. Abbiamo tentato di preservare la terminologia inglese, fornendo le appropriate traduzioni in italiano ove possibile, consapevoli del fatto che la maggior parte della documentazione disponibile sul software, nel momento in cui si decidesse di approfondirne l'uso, è in inglese o comunque fa uso della terminologia inglese.

Blender 2.7

La versione di Blender scelta come riferimento per questa guida è la 2.7. Questa versione rappresenta un'evoluzione della famosa serie 2.5 (su cui si basa la precedente edizione di questo manuale). Durante gli ultimi anni lo sviluppo di Blender ha conosciuto una notevole accelerazione e numerose funzionalità sono state aggiunte o migliorate. Ci teniamo a elencare le rincipsi.

- *Cycles*: un nuovo motore di rendering fotorealistico e interattivo.
- Nuovi strumenti di modellazione.
- Aggiornamento e rifinitura dell'interfaccia.
- Strumenti per *motion tracking*.
- Miglioramenti della funzionalità del sistema di *compositing*.

In questo manuale tratteremo una buona parte di questi argomenti, seguendo il più possibile il percorso di sviluppo e creazione di un contenuto tridimensionale.