

# BLENDER MAGAZINE

ITALIA

Tutorial  
sul Retopo

Le novità  
nella SVN

Le news dal  
Peach Project

I motori  
di render

I render  
architettonici

## Blender Magazine Italia

Responsabili  
Luca Pinciani (Sinistar) e Anfeo

Collaboratori

Davide Vercelli (UncleZeiv)  
Irezumi  
YotaVGA  
Paolo Acampora (Pkcrime)

Grafica  
Luca Pinciani (Sinistar) e Anfeo

Copertina  
Immagine di Paolo Acampora (Pkcrime)  
Grafica di Luca Pinciani (Sinistar)  
Sito web:  
<http://blendermagazine.altervista.org>

Credits:  
Andrea Fiore (waaf), Anto, Antonio  
Pezzella (tonypezz), Dreamer, Ize, waz

Siti di riferimento:  
[www.blender.it](http://www.blender.it), [www.kino3d.com](http://www.kino3d.com),  
[www.blender.org](http://www.blender.org).  
Programmi usati: Blender 2.45,  
TheGimp 2.4, Scribus 1.3.3, Pdftk,  
Open Office

## In questo numero...

News dal Peach. . . . . pag 3  
News. . . . . pag 6

## Articoli

Making of Teddy . . . . . pag 10  
Motori di Rendering. . . . . pag 22  
Novità in SVN . . . . . pag 29

## Tutorial

Rendering Architettonici. . . . . pag 29  
Buchi con il Retopo. . . . . pag 41  
Blender Phyton For Dummies . . Pag 45

## Rubriche

Dove si trova il tasto SubSurf?. . Pag.48  
Comunicazioni dallo Staff. . . . .Pag.51  
Gallery. . . . . pag.52

## Editoriale

Negli ultimi anni Blender ha acquisito sempre maggiore importanza e notorietà.

Ciò perché, se prima solo con costosi programmi di grafica 3d c'era la possibilità di realizzare lavori di altissimo livello tecnico, oggi tale opportunità è alla portata anche di Blender, ormai potente quasi quanto i più conosciuti e utilizzati software 3d. Per questo motivo, esso ha una sempre maggiore diffusione in ogni ambito: amatoriale, commerciale e professionale.

È ovvio che la condivisione di esperienze e lo scambio di informazioni relative all'utilizzo di Blender è di fondamentale importanza per raggiungere una più approfondita e consapevole conoscenza del software e della grafica.

In Italia esistono due comunità importanti e molto attive. I forum principali sono [blender.it](http://blender.it) e [Kino3d](http://Kino3d), entrambi con numerosi utenti più o meno esperti. All'interno di questi, le richieste di consigli, suggerimenti e commenti sono accolte e accordate in modo spontaneamente collaborativo.

È proprio in questo ambiente che nasce Blender Magazine Italia: la prima rivista italiana su Blender. Nei diversi articoli di questo primo numero, sono sottoposti all'attenzione dei lettori diversi argomenti. Tra questi, un interessante approfondimento sui motori di rendering e una dettagliata analisi per i render architettonici; inoltre vengono presentate le novità nella SVN e del Peach Open Movie; vi si trovano poi un tutorial sull'utilizzo del Retopo e una galleria di immagini degli utenti italiani di Blender. E altro ancora...

Saremo lieti di ascoltare eventuali critiche o suggerimenti e vi invitiamo a partecipare con articoli, tutorial, immagini, approfondimenti ai prossimi numeri di Blender Magazine Italia.



# Peach Open Movie News



## ...Notizie dal Peach Team

Il 27 luglio 2007, Ton Roosendaal ha aperto il blog ufficiale del progetto Peach:  
<http://peach.blender.org>  
In questa sezione, raccoglieremo e metteremo in evidenza le notizie più importanti.  
(c) Copyright Blender Foundation

### Il Team ►

Nel progetto, sono impegnati alcuni tra i più importanti artisti che utilizzano Blender:

In dettaglio:

Sacha "Sago" Goedegebure  
(olandese): regista

Andy Goralczyk  
(tedesco) direttore artistico

Enrico "EnV" Valenza (italiano!)  
Artista (modellazione e texturing)

Nathan "Cessen" Vegdahl  
(americano): artista (rigging)

William Reynish  
(danese) artista (animazione)

Brecht van Lommel  
(belga) direttore tecnico

Campbell "Ideasman42" Barton  
(australiano) direttore tecnico

Ton Roosendaal  
(olandese) produttore



Alcune foto dei membri del Peach Team

## Il Cortometraggio e lo Studio ►

Il titolo del cortometraggio è "Big Buck Bunny", ed è stato scelto di chiamare col nome di Buck il coniglio protagonista. L'intero cortometraggio, sarà circa lungo 7 minuti. Ci saranno quattro personaggi: un coniglio, due scoiattoli ed un cincillà.



Per la realizzazione del Peach Open Movie, il Team lavorerà in un ufficio in Olanda.

A fine luglio, Ton ha registrato un video per mostrare ai curiosi lo studio, ancora in parte in allestimento.

Eccolo:

<http://www.blender.org/bf/blender-i.MOV>



## Caffé per il Peach ►

Prologo...

In seguito ad una discussione su kino3d.com, la comunità italiana di Blender invia, al Peach Team, un suggerimento per correggere le arealights di Blender.

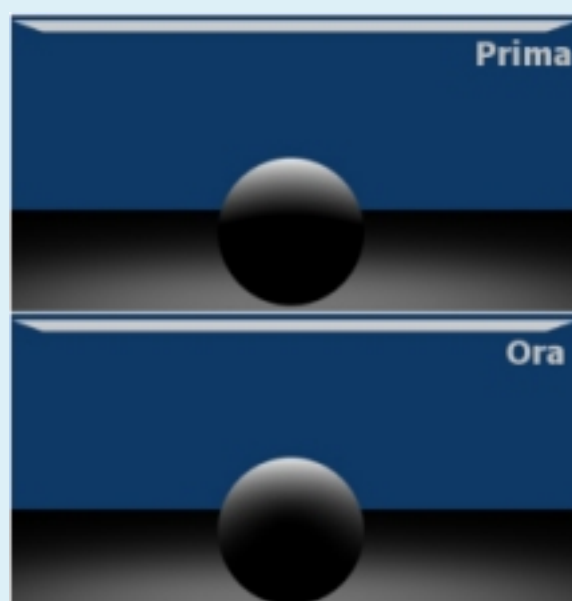
Due giorni dopo arriva la risposta del Team: "e in cambio la comunità italiana inizia una raccolta fondi per farci avere una macchina per l'espresso?".

Due mesi dopo...

Il postino suona alla porta del Blender Institute, recapitando un grosso pacco. Quando Ton apre il pacco è sorpreso del contenuto: un'originale macchina italiana per l'espresso!



...Così, poco tempo dopo, Ton ha mantenuto l'impegno di correggere le arealights di Blender.



Prima la luce non seguiva bene la forma della sfera, mentre ora, con le nuove area lights, essa è chiaramente illuminata in modo migliore.



## Tree Generator

In Novembre, Campbell ha scritto uno script python per Blender per la creazione di alberi. Lo script costruisce una mesh a sezione circolare a partire da curve. Ha numerosi parametri ed è piuttosto stabile e veloce. Per provarlo, è necessario utilizzare una versione SVN di Blender.



Caratteristiche principali:

- UV mapping
- Crescita di ramoscelli da rami robusti
- Riempire ramoscelli in una zona circoscritta
- Crescita delle foglie
- Curve per creare varianti in ogni albero
- Per l'animazione, automaticamente crea un'armatura e i relativi pesi

## Animazione

All'inizio di dicembre è stata finalmente pubblicato un breve video di 20 secondi che mostra alcuni test di animazione dei protagonisti del cortometraggio.

Potete vederne alcuni screenshot qui sotto, oppure l'animazione completa all'indirizzo:

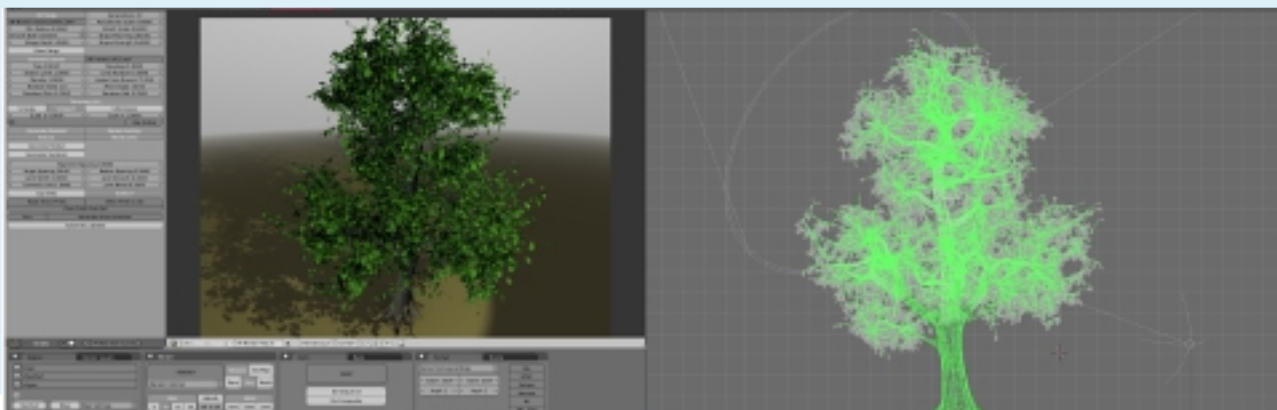
<http://www.youtube.com/watch?v=Tw8WvxZV6GA>

Sicuramente interessante! e successivamente ne verranno pubblicati altri con maggior frequenza.@

Sinistar



(c) copyright Blender Foundation  
peach.blender.org



# NEWS

## Kino3d Contest: Il Lombrico Killer ▶

Il 21 gennaio si è concluso il contest indetto su kino3d da tema:

### IL LOMBRICO KILLER

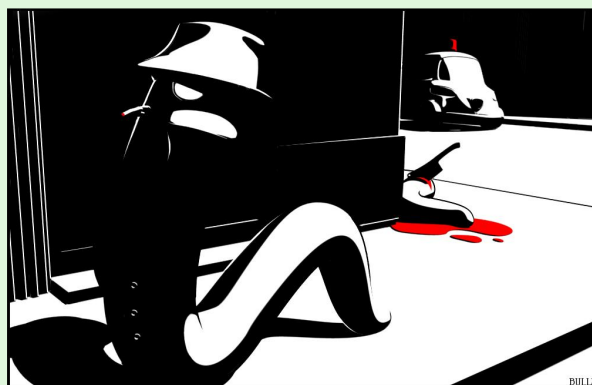
Le modalità di partecipazione erano molto libere, le opere potevano essere eseguite con qualunque mezzo, anche se molti hanno preferito utilizzare Blender. Come da titolo, bisognava realizzare un lombrico-killer o comunque qualcosa che centrasse con il tema.

La partecipazione è stata notevole e di alto livello. Anche la giuria non è stata da meno, infatti, direttamente dal Peach team, ENV in collaborazione con Sago e @ndy hanno scelto i vincitori:

migliore idea: Kaboom con "Il Lombrico"



miglior tecnica: Bullx ”:

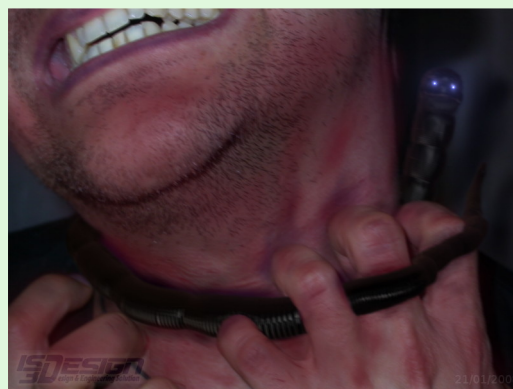


Menzioni speciali per lo stile a:

Gillan



Deskplus



Selidor



Al prossimo contest e per chi si fosse perso la discussione:  
<http://www.kino3d.com/forum/viewtopic.php?t=6176>

mentre qui sono raccolte tutte le immagini partecipanti:  
[http://www.kino3d.com/forum/album\\_cat.php?cat\\_id=5](http://www.kino3d.com/forum/album_cat.php?cat_id=5)

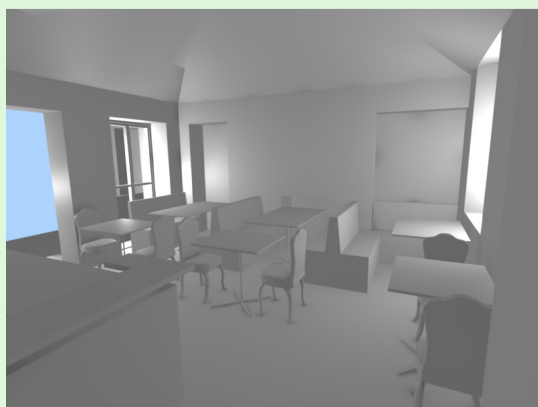
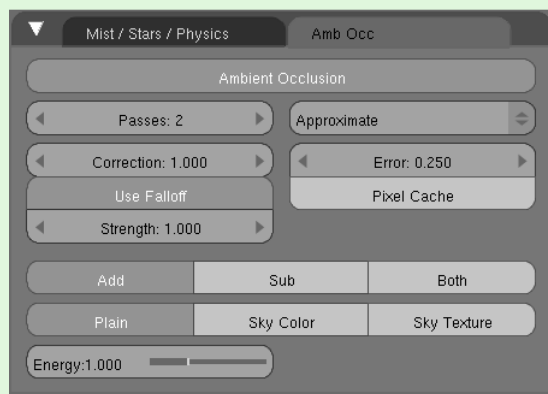


## Approximate Ambient Occlusion

Una nuova opzione sarà implementata nel pannello dell' Ambient Occlusion. Si tratta dell' Approximate AO, che permetterà di fare rendering più veloci a scapito però della correttezza fisica dell'illuminazione. Questo include vantaggi e svantaggi quindi ho fatto alcuni test con un interno.

NB: i test sono stati eseguiti con un portatile fornito di un AMD 2000xp e 512mb di ram.

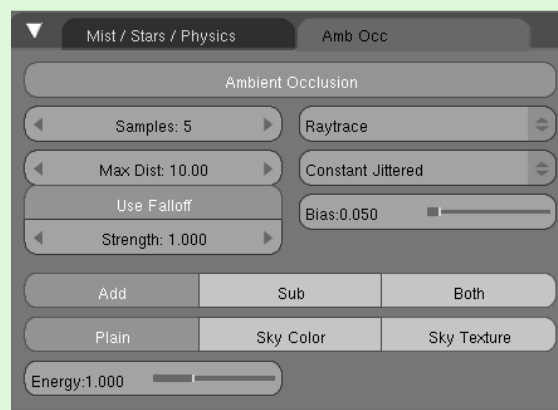
In questo render ho utilizzato l'Approximate AO, i parametri sono questi:



Render1: 4.34 min

Si può notare un'assenza quasi totale di rumore (tipica dell'AO) ma ci sono delle imperfezioni in alcuni punti. Come tempi posso dire che si avvicinano a quelli di Yafray che erano di 5 minuti e rotti.

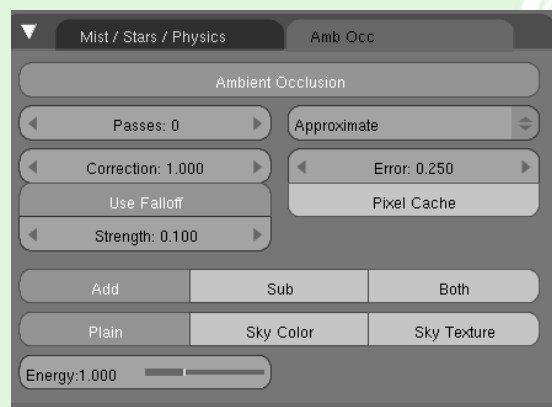
Questo è un render con l'AO normale con questi parametri:



Render2: 2.12 min

Pazzesco ma i tempi sono molto inferiori, questo è dovuto ad un basso numero di Samples, infatti si nota principalmente il rumore che caratterizza questi rendering.

Un'altra prova con l'Approximate AO, questa volta ho settato i parametri in questo modo:

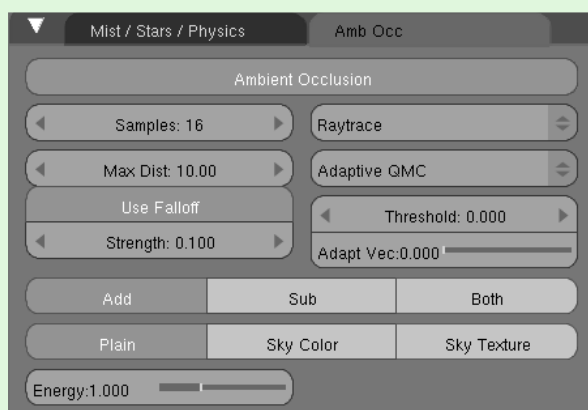




Render3: 3.29 min

I tempi sono diminuiti ancora e penso che con un settaggio più accurato sia possibile abbassarli ulteriormente. Si nota ancora qualche artefatto, ma per il resto è passabile.

Infine ho fatto un redering con l'AO, settato in Adaptive mode (vedere articolo su SVN per approfondimenti) e ho aumentato i Samples a 16.



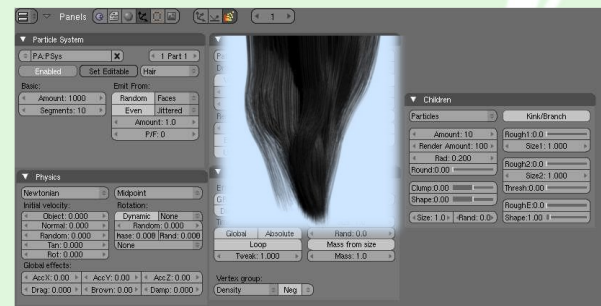
Render4:17.38 min

I tempi sono molto ma il rumore è sparito completamente e l'illuminazione è più fisicamente corretta.

In conclusione si può dire questa nuova feature è una manna dal cielo per le animazioni, ma potrebbe essere utile anche a quelli che non amano le lunghe attese per i loro rendering e per i loro clienti.

## Nuove Particelle

La tanto attesa versione del Particles System di Jahka pare matura per un inserimento in SVN. Molte le novità per quelli che avevano provato le vecchie versioni che, per chi non lo sapesse, prevedevano un sistema di controllo della fisica e uno strumento (chiamato Particles Mode) che corrisponde ad un vero e proprio parrucchiere digitale, dove è possibile allungare, accorciare, tagliare, spostare, gonfiare e aggiungere



particelle sotto forma di capelli. Una delle novità di questa versione è la possibilità di lavorare su poche particelle, alleggerendo il sistema e renderizzando alla fine il numero di "children" settati.

La maggior parte delle novità riguardano appunto la creazione di capelli, è infatti possibile selezionare direttamente un'opzione Hairs e automaticamente avremo il sistema predisposto per la creazione di capelli o Fur. Sono stati migliorati anche i materiali quindi sarà possibile ottenere meravigliose chiome in minor tempo.

Molto presto pubblicheremo un tutorial al riguardo, per ora bisogna accontentarsi della documentazione in inglese.





Interessantissimo tutorial scritto da Lucio Boscolo Mezzopan (loscotec) che tratta la modellazione, l'animazione, lo shading e lo scripting, tutto incentrato su un veicolo elettrico a 3 ruote chiamato e-Snake.

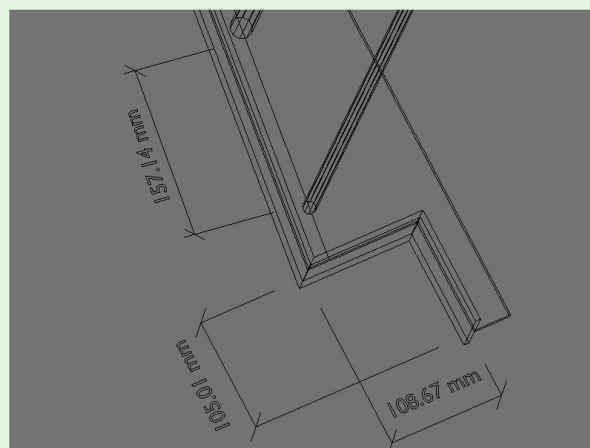
La parola tutorial è diminutiva in quanto si tratta di una vera e propria guida di 56 pagine, adatta ai novizi e imperdibile anche per gli utenti più esperti.

Molto interessante la parte dello scripting che è stato usato per il calcolo della posizione del corpo del triciclo, la rotazione delle ruote, l'inclinazione dello sterzo e il rollio della parte anteriore, partendo semplicemente da un empty che si trova nel punto di contatto a terra della ruota anteriore, al quale viene impostato un constraint per fargli seguire un percorso percorso.

Il sito da cui potete scaricare il PDF è il seguente:

<http://loscotec.interfree.it/>

Buona lettura.

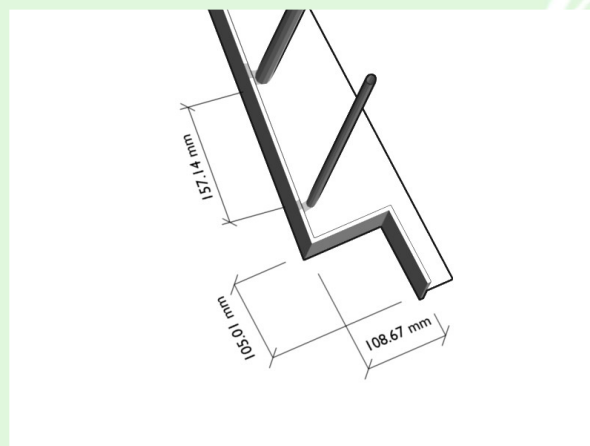


Per chi non lo conoscesse è uno script dallo sviluppo tutto italiano per l'integrazione di strumenti Cad in Blender. L'autore di questo progetto si chiama JAPY (Jefferson Air Project Yea!) e in tre anni è riuscito a creare e perfezionare uno strumento che ha aiutato moltissime persone. Con l'ultima release, sono state implementate le tanto agognate quote che faranno la felicità di molti utenti.

Per provare l'ultima versione:

<http://blender-archi.tuxfamily.org/Image:PROCAD24.zip>

Mentre non dimenticate di tenere d'occhio la discussione su blender.it: <http://www.blender.it/forum/viewtopic.php?t=3171&postdays=0&postorder=asc&start=0>



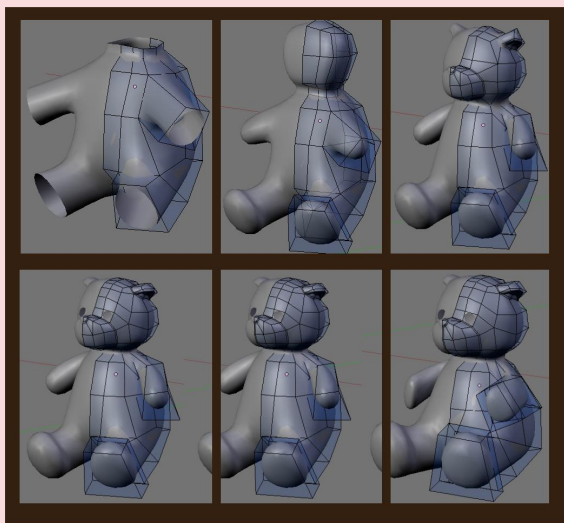
L'idea per quest'immagine è nata da una lettera minatoria inviata a Stephen King: "Finisci la saga della Torre Nera altrimenti l'orsacchiotto muore!"

Legare e imbavagliare un pupazzo è una cosa che non farei mai, non sono crudele abbastanza, ma la realtà virtuale, è tutta un'altra cosa!

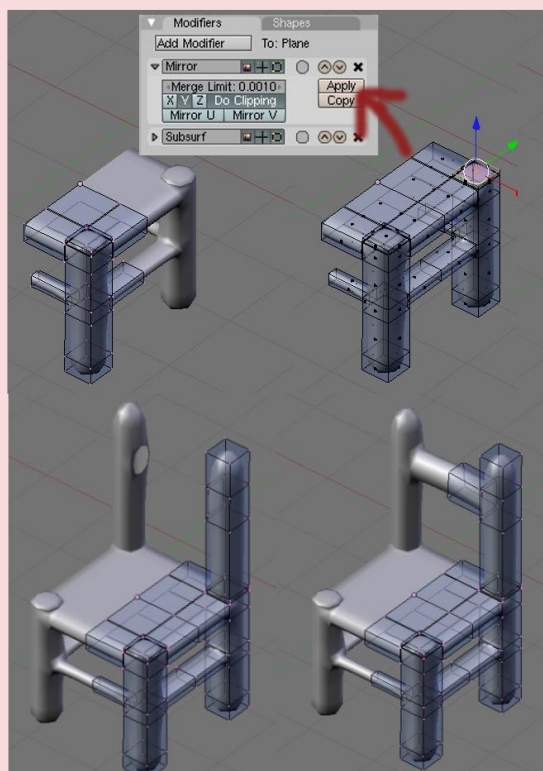


## Modellazione

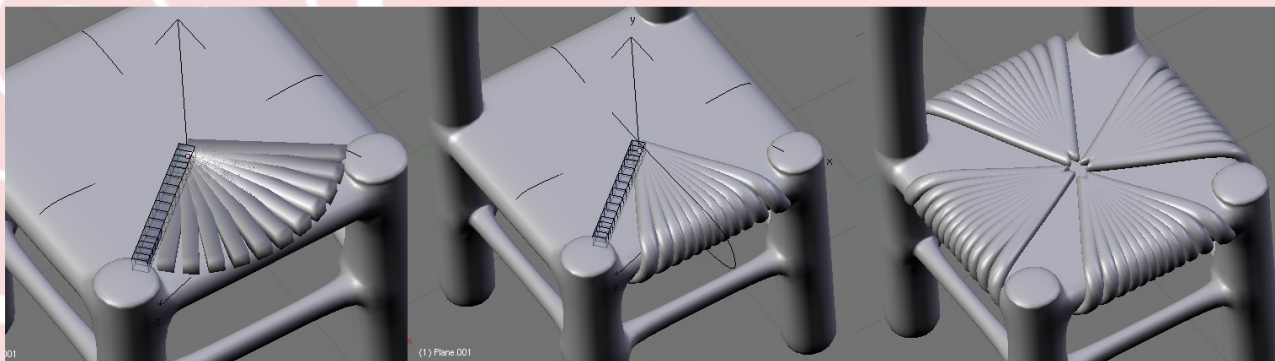
L'**orsetto** è un oggetto abbastanza facile: si presta benissimo al box-modelling



Per la **sedia**, sono partito da un quarto di sgabello, ho applicato il modificatore mirror e estruso lo schienale.



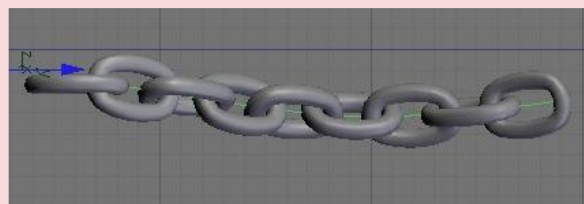
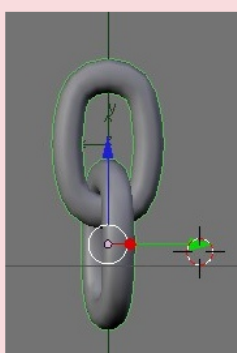




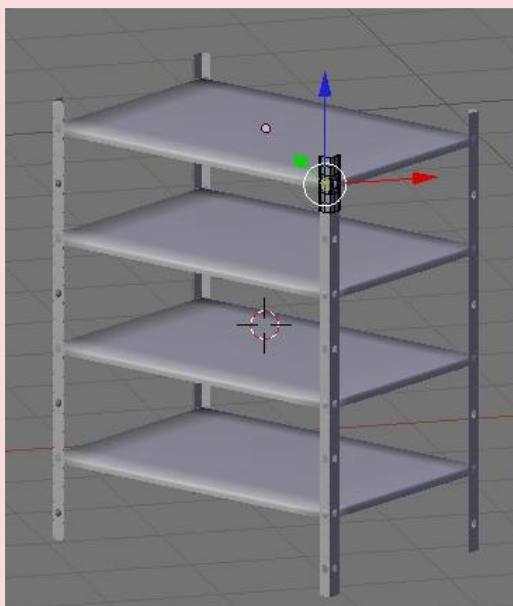
Per fare i **vimini** ho usato un parallelepipedo molto suddiviso, con un modificatore array.

Per ottenere un array a ventaglio basta abilitare soltanto "object offset" e immettere il nome di un oggetto ruotato (nel mio caso un empty). Con il modificatore curve ho fatto in modo che seguisse una curva aderente alla sedia.

Della **catena** basta creare un solo anello e duplicarlo con array. Bisogna usare "Object Offset" e immettere il nome di un empty leggermente spostato verso l'alto e ruotato di 90 gradi, in questo modo ogni anello sarà più in alto e perpendicolare a quello precedente.



Alla catena così ottenuta possiamo applicare un modificatore curva, in modo da poterla deformare a piacimento e, magari, avvolgerla attorno al pupazzo (grazie ZanQdo!).

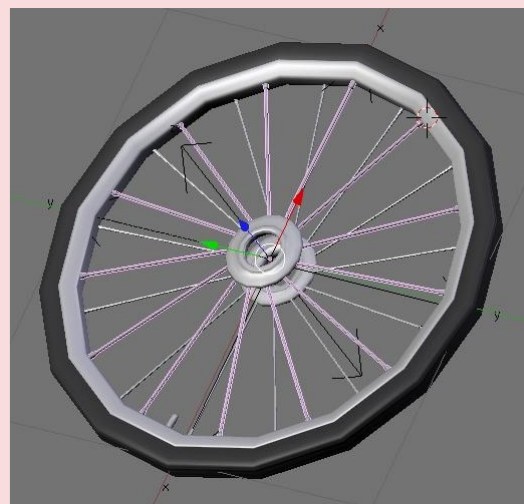


Sempre grazie ad array, è bastato modellare una "L" forata perchè diventasse un piede degli **scaffali**; attivando merge le copie si fonderanno in un unica mesh. Con un modificatore mirror attivo sia sull'asse x sia su y, i piedi sono diventati 4.

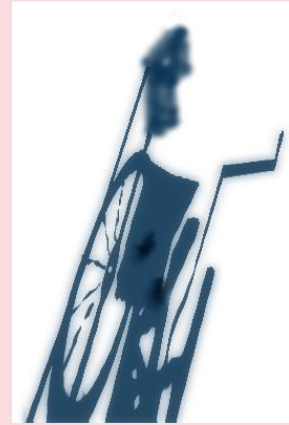


Un'ultima menzione alla **ruota** di bicicletta: anche nei raggi c'è lo zampino di un array con offset su un empty ruotato.

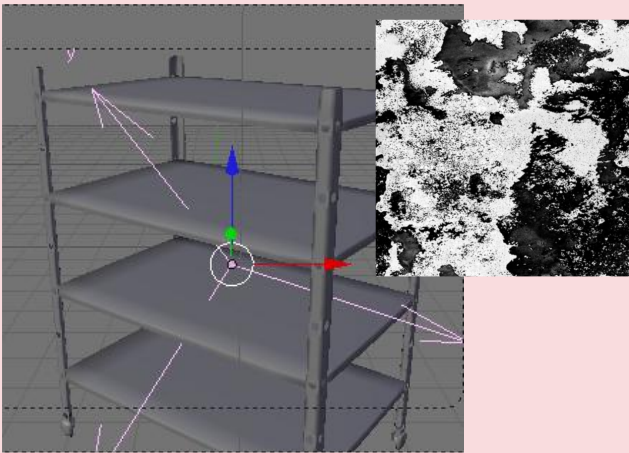
I modificatori mirror e array mi hanno permesso di modellare tutte le cianfrusaglie del mio scantinato in meno di due giorni.



Le ombre sul pavimento sono una texture. Per generarla mi sono basato su un bake del full render di una scena analoga, a cui però sono stati tolti tutti i materiali e aggiunta una luce generatrice. Mi piaceva l'idea che le ombre si allungassero verso lo spettatore, ma nella scena finale la fonte di luce è vicina e in alto, questo vuol dire che avrei dovuto disegnarle più corte.



Le texture dei muri sono prese da [www.mayang.com](http://www.mayang.com). Le ho applicate sui canali Nor, e in alcuni casi su Col, con multiply a opacità molto ridotta.



La texture degli scaffali, ha come "object offset" un empty trasversale, in questo modo ho ottenuto rapidamente uno sporco non ripetitivo con un solo materiale e una sola mappa, senza UVmappare.

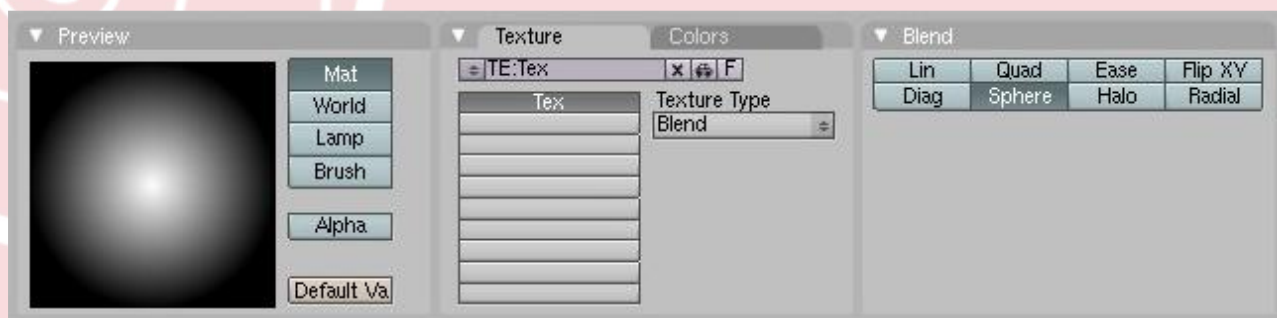


Per ottenere un effetto di (fake) trasparenza sulle sbarre e gli assi alle finestre, ho usato il vecchio trucco della texture a gradiente sferico con Nor come input.

È una tecnica molto utile, quindi vale la pena di illustrarla partendo da un caso generico.



A un qualsiasi oggetto, aggiungiamo una texture blend di tipo Sphere.



Nella scheda Map Input selezioniamo Nor e disattiviamo gli assi X e Y. Significa che la texture sarà visibile solo sulle superfici che non guardiamo frontalmente.

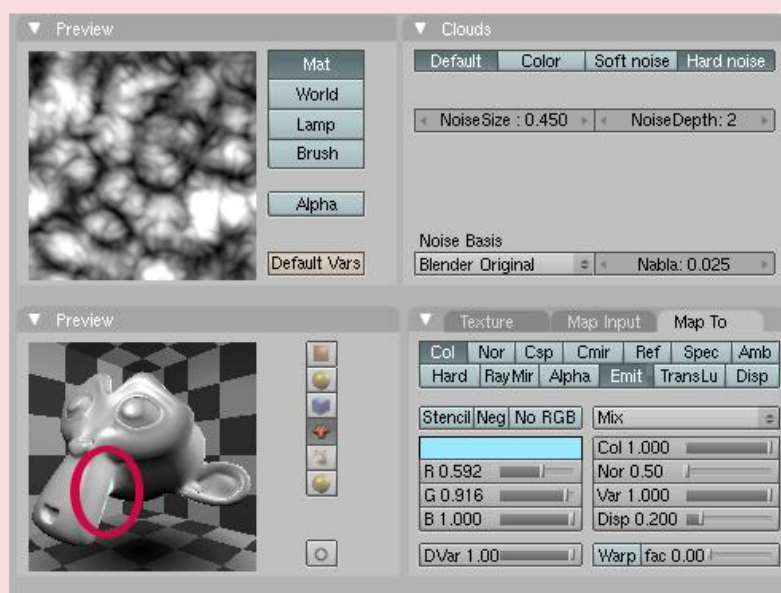


Nella scheda Map To scegliamo Emit invece di Col, perchè vogliamo che l'oggetto sia luminoso sui bordi.



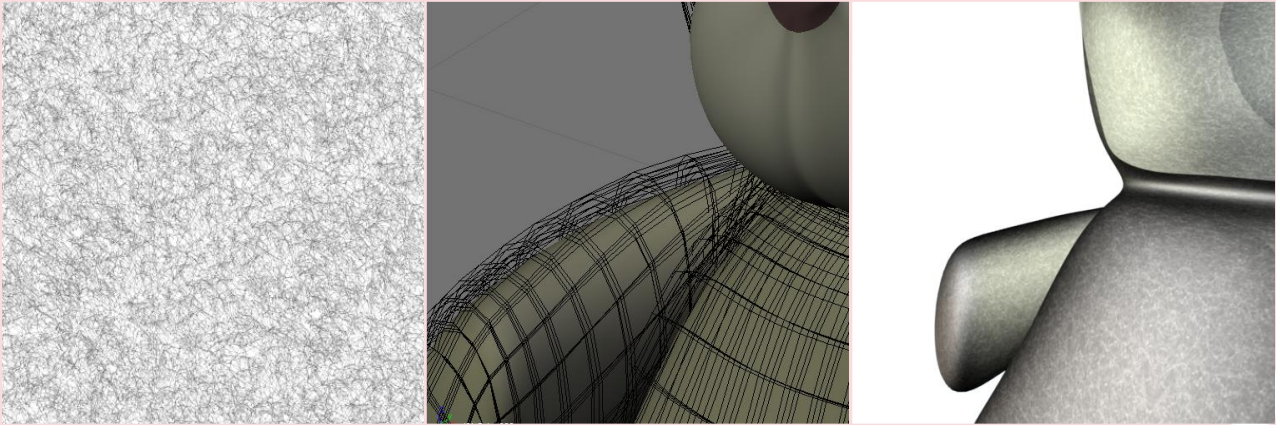
Volendo si può giocare con il valore di luminosità della texture per regolare la luminosità del bordo, mentre cambiando il contrasto si regola la transizione dal centro al bordo.

Nel mio esempio però, volevo una trasparenza più irregolare, quindi invece di usare il gradiente direttamente come valore di emit, l'ho usato come mascherina attivando "Stencil" sulla scheda "Map To", e disattivando tutti gli altri valori; sul canale superiore ho aggiunto una texture cloud, con "Map To" Emit e Col, scegliendo una colorazione azzurra, come la luce del cielo.



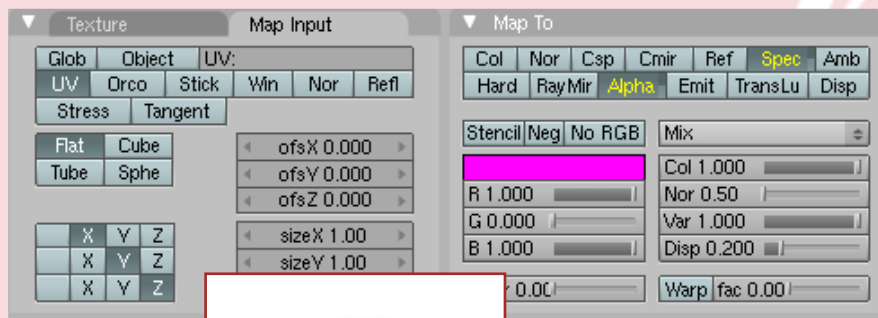
Tra le novità già presenti nell'SVN di blender c'è un livello di controllo incredibile sul pelo particellare, ma con la versione 2.45 che ho usato ho preferito ricorrere al trucco delle mesh sovrapposte, combinando gli strand con una pelliccia fake.

Si tratta di applicare una texture raffigurante dei peli, una sorta di tappeto, su più mesh concentriche (ottenute espandendo una copia con Alt+S) che ingabbino il personaggio.



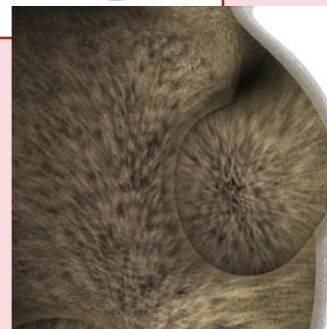
poiché la texture è seamless, ho potuto usare come input l'UV layout di default, quello con tutte le facce sovrapposte.

Selezionando Alpha e Spec in "Map To", con Dvar impostato a 0 e Ztransp selezionato in "Links and Pipeline", le "gabbie" diventeranno trasparenti nei punti colorati dalla texture.



Dare un materiale scuro alla mesh di base aiuta ad ottenere un effetto migliore: la parte scura, che si intravede sotto la gabbia, verrebbe ad essere la base del pelo, che si schiarisce man mano fino alla gabbia più esterna.

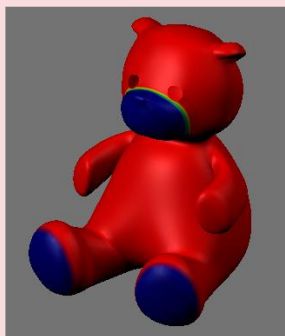
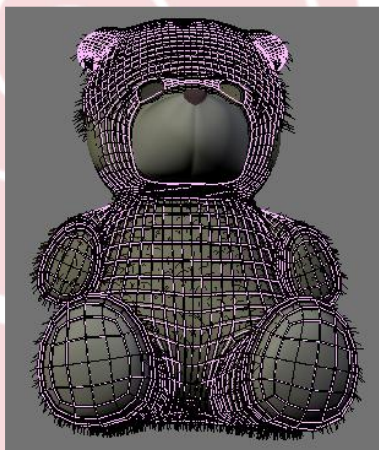
Nel mio esempio ci sono quattro "gabbie" (o "shells", come si dice in inglese, questo metodo non l'ho inventato mica io!).



Per rendere il pelo un po' più voluminoso, e soprattutto fare in modo che si veda sui bordi, ho comunque dovuto aggiungere delle particelle.

Peli così corti non richiedono chissà quale setup di pettinatura, ho usato dei vertex group, uno in fondo alla scheda particles per non far crescere peli sul muso, e uno nella scheda particle motions per regolare le transizioni tra zone coperte da pelo e zone che non lo sono (o che sono coperte da bavaglio e catena), per il resto li ho semplicemente lasciati crescere!

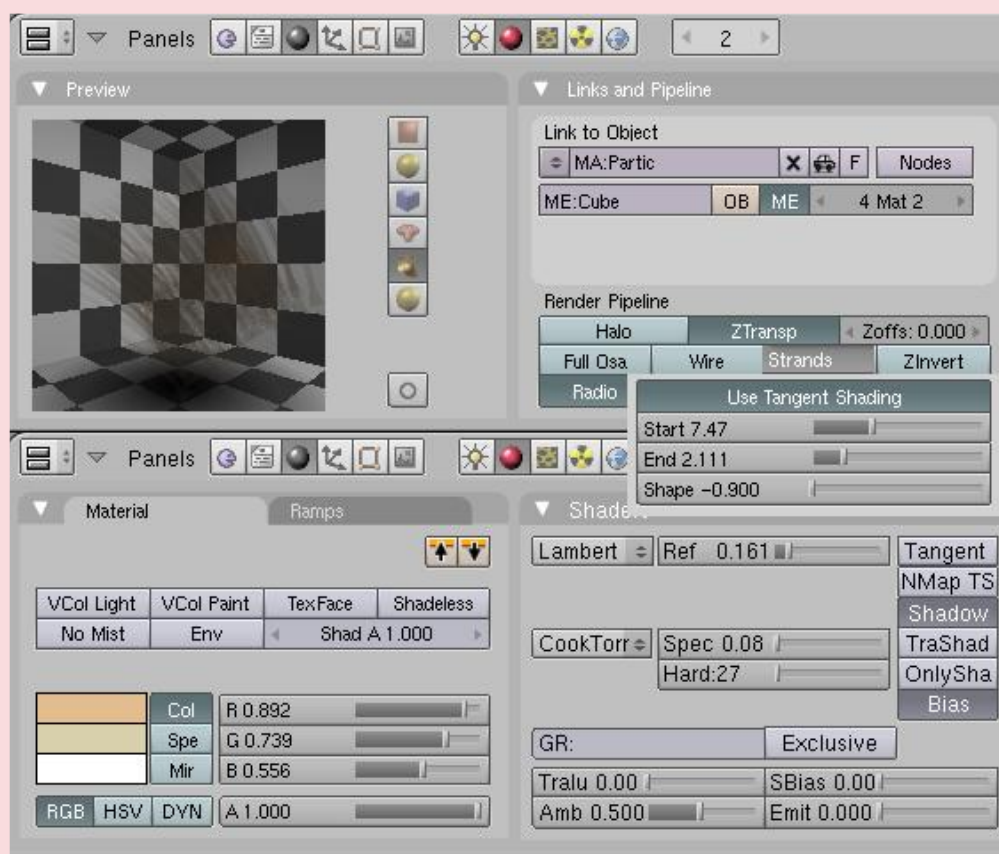
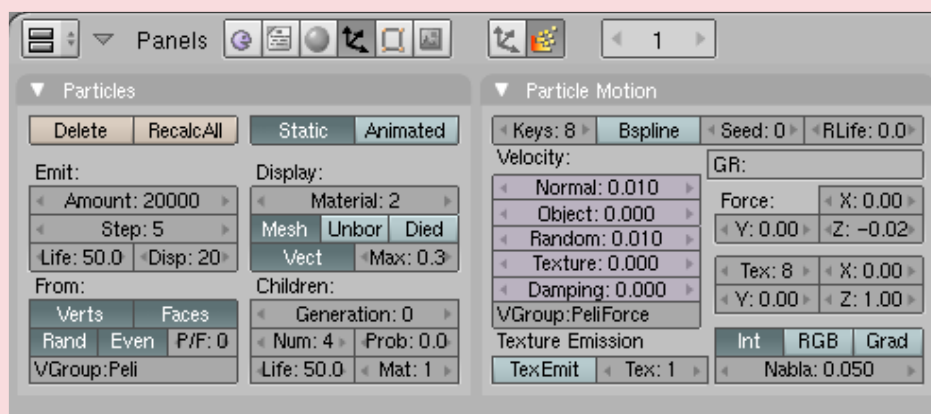




Il setup molto semplice delle particelle.

Niente di troppo originale, ho impostato 0,3 come lunghezza massima nella scheda "Particles", aggiunto un po' di casualità (Random:0,01) e gravità (Force Z:-0,02) nella scheda "Particle Motion".

Inoltre ho scelto che le particelle vengano visualizzate con il materiale numero due dell'oggetto (Material: 2, alla voce Display della scheda Particles).



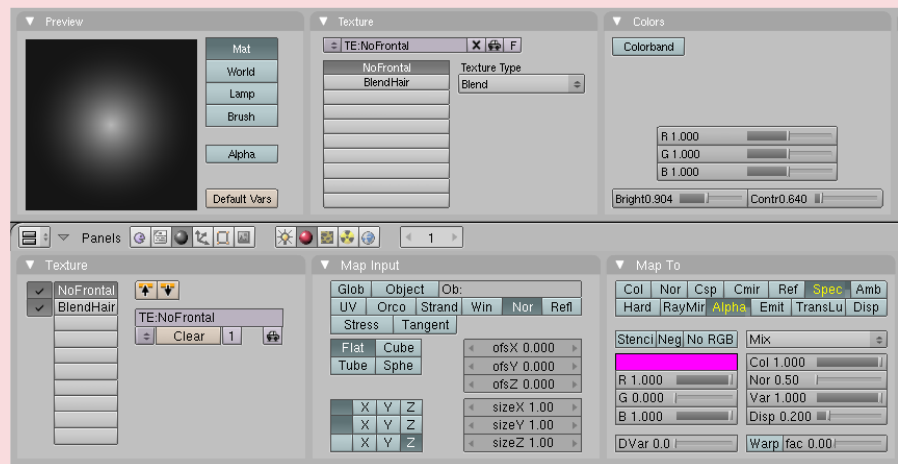
Il materiale numero due controlla il pelo particellare e il modo in cui questo si integra col "pelo texture".

Premendo sul tasto "Strands", in "Links and Materials", compare il pannello "Tangent Shading": il valore "Start" influenza la dimensione del pelo alla base, "End" quella sulla punta, "Shape" quanto la forma debba essere arrotondata (a -0,9 è molto appuntita).

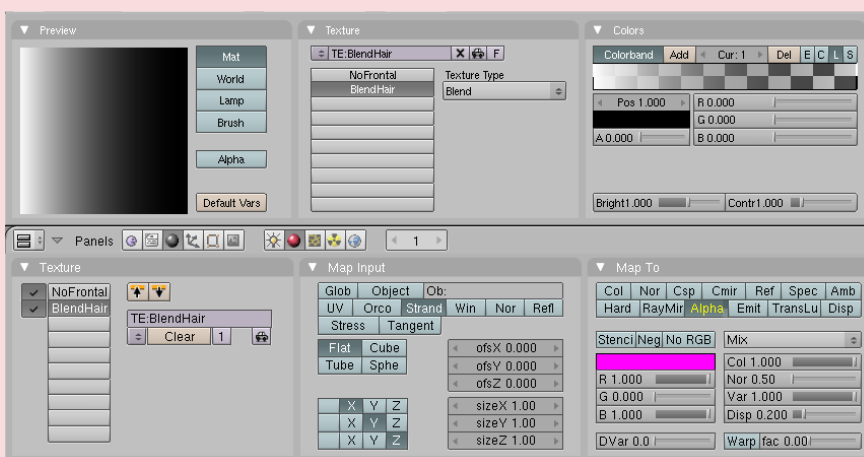
Il colore è quello che vogliamo dare alla pelliccia (lo stesso degli shell) mentre in Shader

ho dovuto impostare un valore di riflessione molto basso, poiché le strand ricevono un po' troppa luce rispetto agli altri oggetti.

A gestire la trasparenza ci pensano due texture, la prima è una blend di tipo Halo, che ho chiamato NoFrontal, serve a non far vedere i peli che sono esattamente frontali alla camera: infatti nella vista frontale gli shell con la texture se la cavano benissimo, mentre sui lati c'è bisogno di vedere del pelo reale.



I valori di luminosità e contrasto regolano ancora una volta rispettivamente visibilità e transizione; come nell'esempio di prima sulla trasparenza, lasciamo solo l'asse Z in "Map Input", mentre per fare in modo che la texture agisca sulla trasparenza selezioniamo "Alpha" e "Spec" in "Map To", ricordandoci di portare Dvar a 0.



L'altra texture è sempre una blend, stavolta lineare, il cui compito è rendere i peli ancora più sottili della loro geometria.

Per questo il "Map Input" sarà "Strand", e "Map To" invece Alpha.

Ricapitolando:



Mesh di base



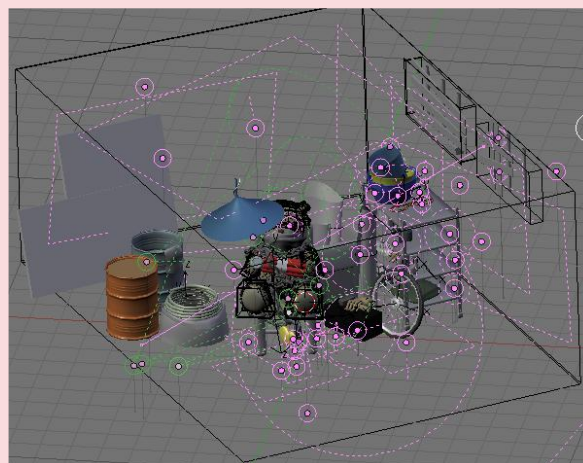
Mesh+gabbie con texture alpha



Particelle visibili solo quando non sono frontali



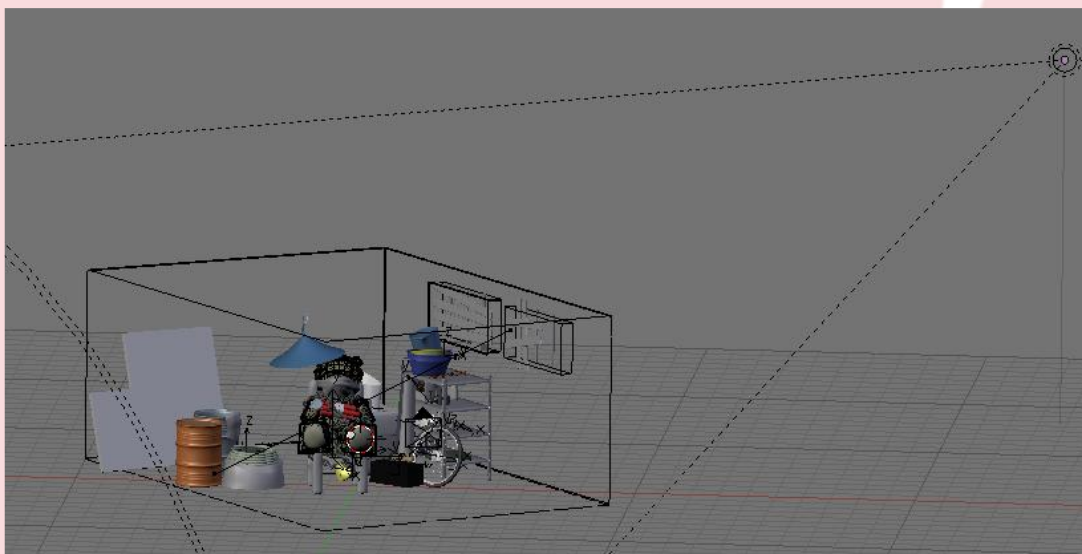
Risultato finale (tutto insieme!)



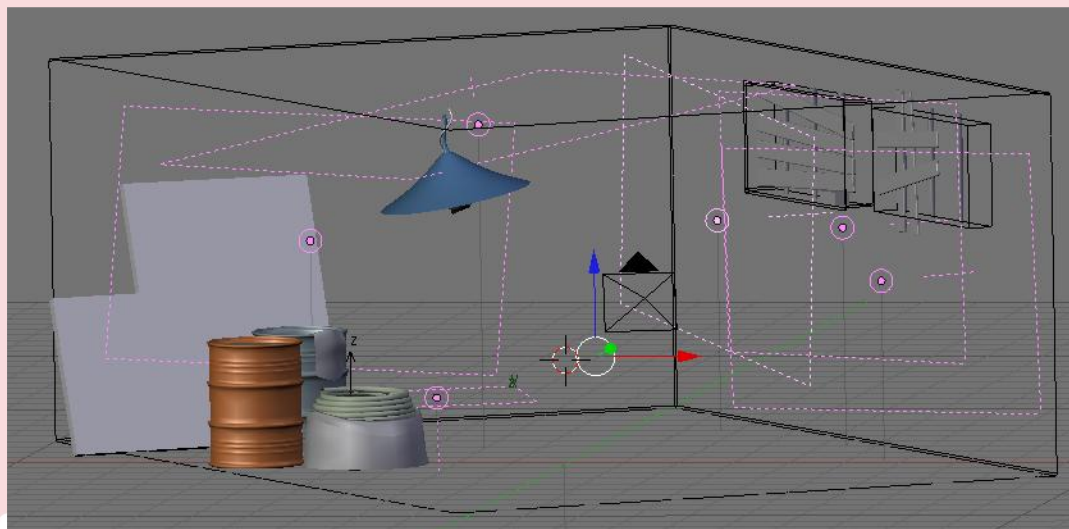
### Illuminazione

Non proprio il mio forte! Ho cercato di tenere un'atmosfera molto tenebrosa, poi però ho aggiunto molte luci per far risaltare alcuni oggetti.

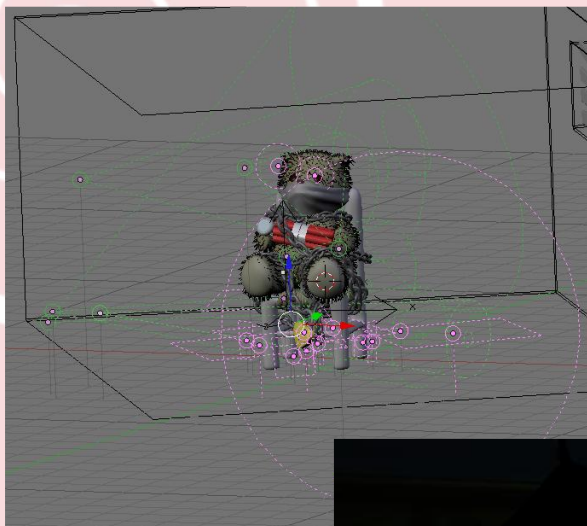
La luce principale è una grossa spot, che illumina attraverso le finestre; "clip start" e "clip end" (la regione nella quale le ombre vengono calcolate e utilizzate) coincidono con la lunghezza della stanza. La buffershadow di questa lampada è l'unica fonte di ombre nella scena.



In prossimità di ogni superficie ampia, c'è la classica luce di "fake radiosity", nel mio caso area light a bassissima energia e distanza,



con specular disattivato e il colore delle superfici vicine.



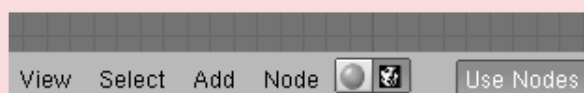
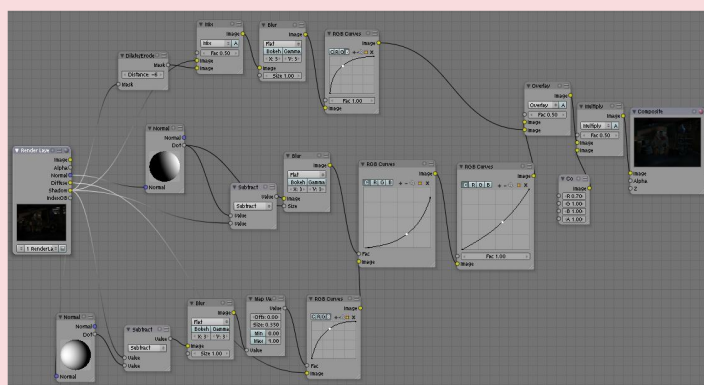
Altre luci fanno risaltare il centro della scena, sono tutte luci a bassa intensità, con attivata l'opzione sphere, che le rende inefficaci al di fuori della loro distanza.

Il risultato, sarà anche tenebroso, ma è certamente poco incoraggiante... Sigh!



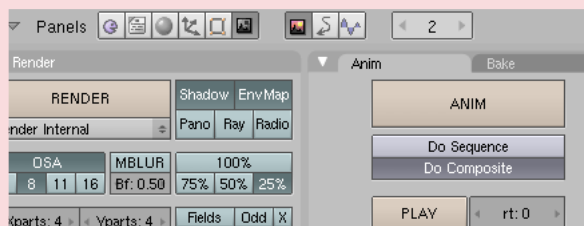
## Compositing ►

Niente paura, c'è il compositor, che permette di editare un render (o una qualsiasi immagine) attraverso uno schema di nodi invece che usando un programma di fotoritocco. I nodi possono usare anche informazioni 3d, come le normali, o l'UVmap.



Per usare i nodi compositi, bisogna scegliere "node editor" nell'header della finestra e attivare i pulsanti in figura.

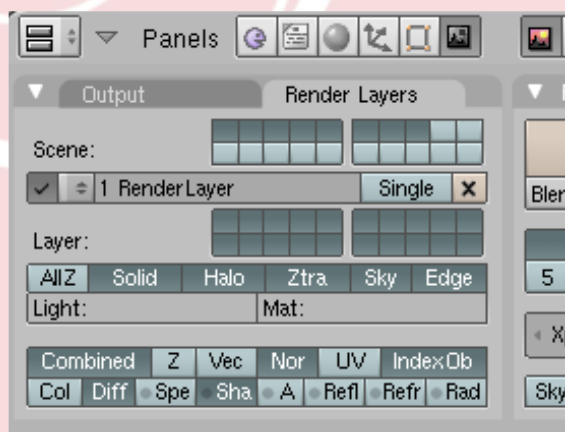
Inoltre bisogna attivare "Do Composite" nella scheda "Anim" del pannello "Render".



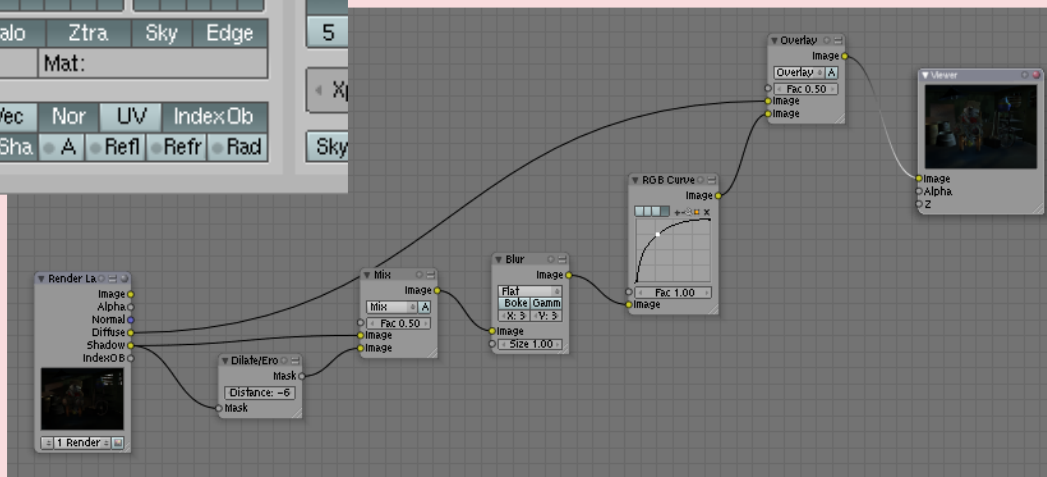
A lato c'è il mio schema di nodi... fa un po' paura, ed è molto caotico, ma proverò a spiegare un paio di passaggi.



Non tutti gli output (pass) del render sono disponibili di default, per abilitarli bisogna selezionarli nella scheda "Render Layers", pannello "Render"

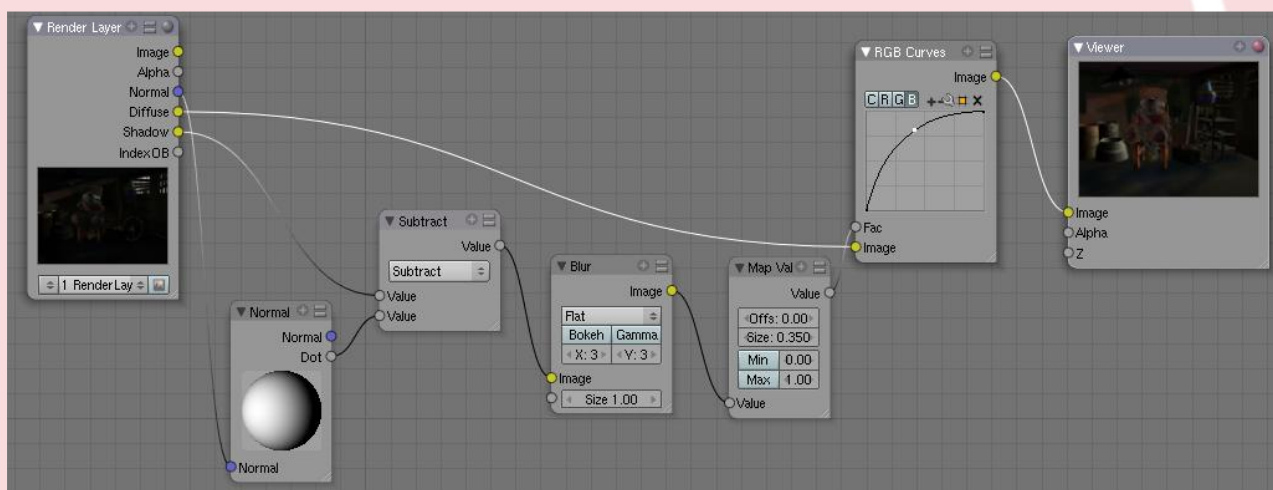


Poichè volevo che le ombre fossero poco nette e con una sfumatura celeste, quello che ho chiesto al compositor è prendere il canale delle ombre, eroderlo, sovrapporlo a sé stesso con opacità 50% (Fac 0,5), sfumare il tutto con un blur, alzare la curva del blu con "RGB curves", e infine un overlay sulla scena senza le ombre (diffuse pass).

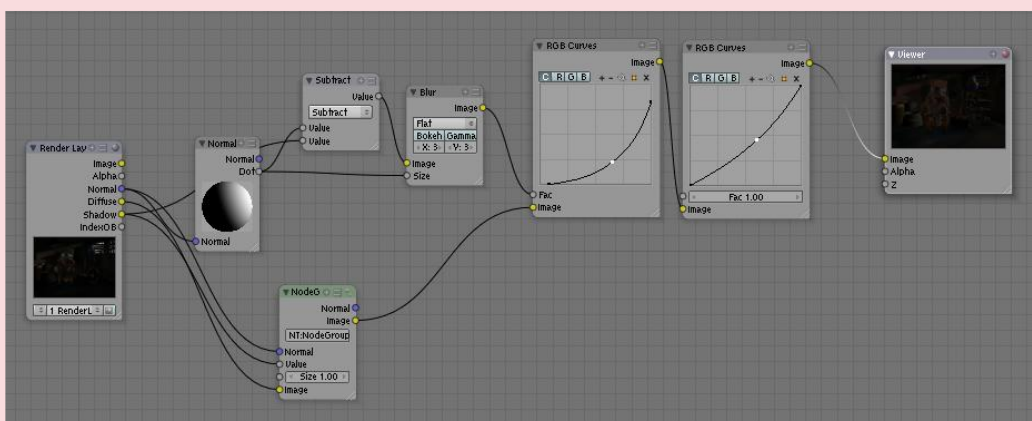


Questa parte di schema invece dice di applicare una correzione RGB sul diffuse, in corrispondenza delle zone d'ombra, ma non sulle facce che

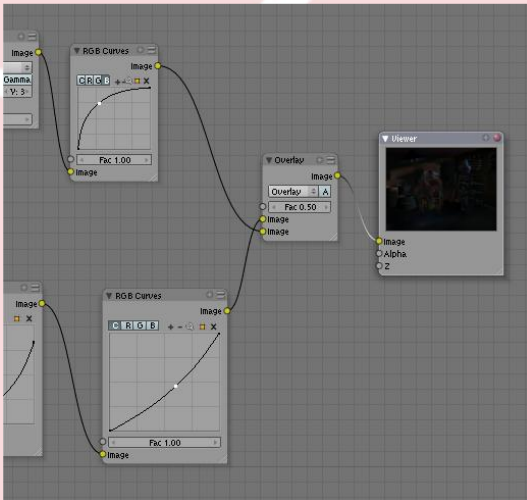
guardano nella direzione del nodo normal, ecco perchè l'input (Fac) è "shadow meno normal", poi sfocato e smorzato.



Quanto ottenuto da questa sequenza, lo do in pasto a uno schema simile, che scurisce le facce opposte.



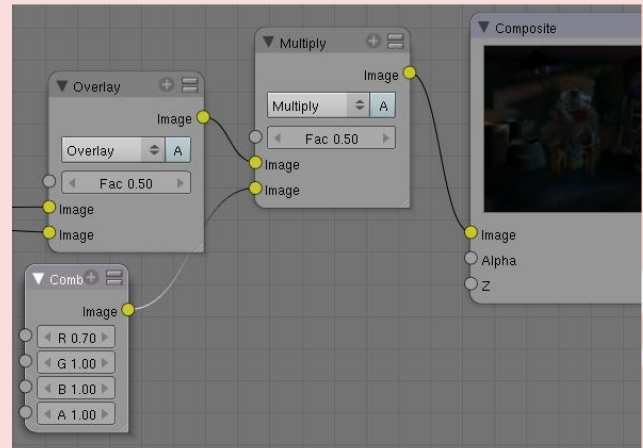
Il risultato l'ho usato al posto del diff mostrato nel primo schema parziale, per applicarvi le ombre.



Nel mio caso ho moltiplicato per Rosso=0,7 , Verde=1 , Blu=1, cioè, per ogni pixel, blu e verde restano uguali (moltiplicare per uno non cambia i numeri), mentre la componente di rosso scende al 70% di quello che era (occhio che in questi casi bisogna considerare il campo del colore da 0 a 1, non da 0 a 255).



L'ultimo passaggio prima dell'output definitivo (nodo Composite) è un Multiply, inserito perchè l'immagine aveva troppo "rosso". Quando c'è sovrabbondanza di un canale, la cosa più sbrigativa da fare è moltiplicare per un colore che abbia una frazione del canale di troppo, e 1 sugli altri canali.



Rispetto a prima, direi che è un notevole passo avanti, adesso ci vuole una bella passata di Gimp per dare qualche ritocco.

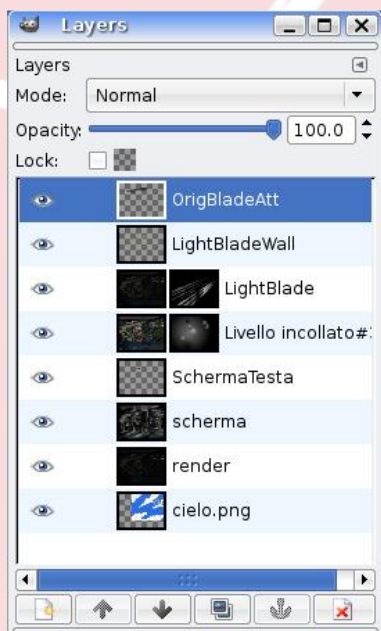
## Post-processing ►

*Due righe in difesa del post-pro*

Spennellare allegramente su un'immagine renderizzata è spesso considerato un po' come imbrogliare. Il fatto è che non è sempre facile indovinare le luci e i materiali che portano al risultato desiderato, ed è un peccato che un'immagine ne subisca le spese non liberando tutto il suo potenziale. Anche se dopo questo processo il risultato non è più animabile, non dimentichiamo che quasi tutto quello che facciamo col

fotoritocco è replicabile nel compositor (al costo di aggiungere degli oggetti che ci facciano da "maschere") o apportando modifiche minori alla scena, quindi, se proprio vogliamo che il nostro lavoro sia puro rendering, possiamo considerare il post-pro alla stregua di un lavoro di concept, utile a capire cosa dovremmo cambiare o aggiungere nel rimettere mano all'ambiente tridimensionale. In effetti sarebbe stato meglio fare delle prove con Gimp/Photoshop/Altro prima di settare luci, materiali e compositor, ma ormai...





aveva lasciato  $\alpha=0$  in corrispondenza delle finestre, quindi

Utilizzare più livelli lasciando inalterato il render di partenza può tornare utile per ripensamenti e verifiche; il primo livello è quello più in basso, e nel mio caso si tratta di un cielo.

Al secondo livello c'è il render, il buon blender

nessun problema di integrazione.

Terzo livello: un duplicato del render, leggermente sfocato e con modalità "schermo" (Mode: Screen) dà più luce e colore a tutta la scena. Il quarto livello fa la stessa cosa, ma ho cancellato tutto lasciando solo la sommità della testa, che volevo far risaltare.

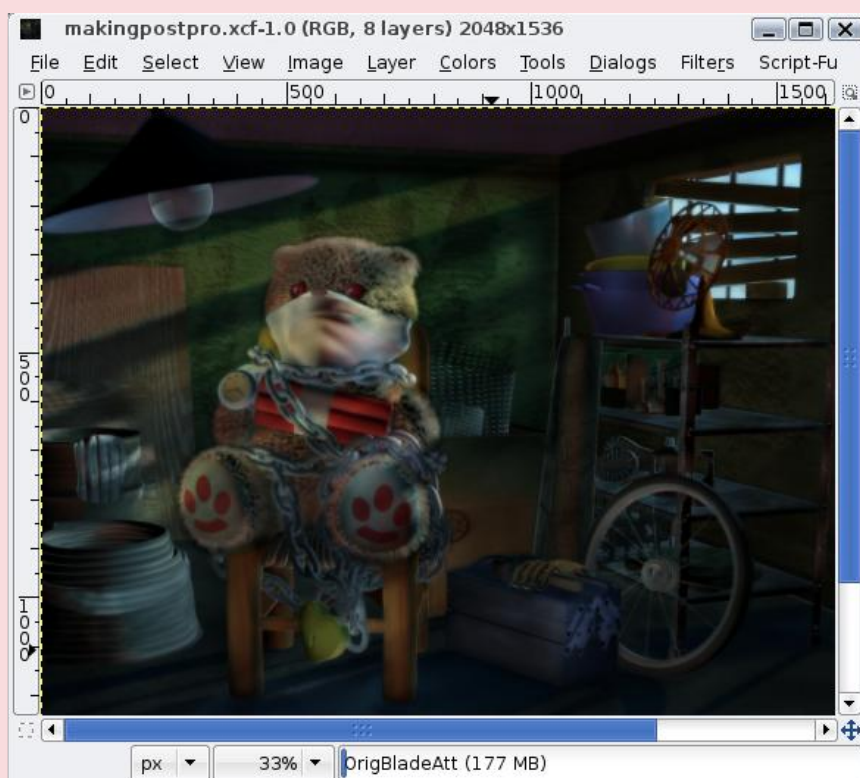
Al quinto ho "schermato" ulteriormente altre zone da porre in risalto, per farlo ho aggiunto una maschera tutta nera (che quindi inibiva l'intero livello) andando poi a schiarirla con dei gradienti sferici da bianco a trasparente. Questo crea transizioni molto graduali; una miniatura della maschera è visibile a lato di quella del livello. Stessa cosa con il sesto livello, dove la maschera a strisce crea delle "lame" di luce sul modello.

Il settimo livello aggiunge strisce di luce anche sul muro, mentre l'ottavo ritocca un po' la striscia originale, conformandola ai colori del resto dell'immagine.

Ritocchi minori ne ho apportati col pennello, ma ho già finto di saperne troppo, quindi meglio fermarsi qui... spero che quanto ho scritto possa tornare utile, grazie infinite per la lettura, a presto!

Per chiarimenti, critiche, refusi e segnalazioni su quest'articolo, oltre che alla rivista potete scrivere all'indirizzo

[studioamigdala@libero.it](mailto:studioamigdala@libero.it)



Inizia questa rubrica che ha lo scopo di informare sui renderer e sulle tecniche di rendering in maniera da poter scegliere lo strumento migliore di volta in volta per ogni lavoro.

Come sarà via via sempre più chiaro man mano si andrà avanti con gli articoli non esiste un renderer migliore degli altri, e non esiste una semplice regola che permetta di scegliere il renderer per il proprio lavoro. L'unico modo per fare una scelta corretta è quello di conoscere tutte le variabili in gioco e fare una scelta cosciente.

Questa rubrica tratterà sia le tecniche di rendering sia i renderer interfacciabili con blender. Il primo aspetto è fondamentale per apprezzare e capire correttamente le recensioni che verranno fatte. Questo primo articolo, in particolare, sarà un'esposizione di tutti i termini utili e dei concetti fondamentali per capire in seguito le tecniche di rendering.



immagine di Antonio Pezzella

# Motori di Rendering

Il grafico che costruisce una scena 3D si occupa di descrivere una scena. Posiziona i vertici od i punti di controllo, sceglie i materiali, muove le armature, posiziona le luci e via dicendo. La scena in questo modo viene descritta, ma non rappresentata. Per fare un paragone, il progetto di un palazzo ed il palazzo stesso non sono la stessa cosa. Si può descrivere perfettamente un come andrà costruito un edificio, fin nei minimi dettagli, ma se ne avrà sempre una descrizione.

Ovviamente possiamo realizzare un'immagine che sia fedele, in tutto, alla descrizione data, così come dei muratori potrebbero costruire il palazzo dal progetto. Questo compito è svolto da un programma chiamato renderer, che compie un'operazione chiamata rendering, che non è altro che la produzione dell'immagine a

partire dalla sua descrizione.

La trasformazione della descrizione della scena in un'immagine è un'operazione complessa, che per questo motivo può essere fatta in uno svariato numero di modi e con uno svariato numero di tecniche, alcune delle quali più veloci, altre più lente, alcune più precise, altre meno. Inoltre un renderer potrebbe non essere in grado di comprendere o di rappresentare determinate descrizioni.



## FOTOREALISTICO OD ARTISTICO?

Sebbene i renderer svolgano tutti lo stesso compito (per quanto alcuni meglio ed altri peggio), c'è una differenza fondamentale sul risultato finale, lo stile con cui l'immagine viene riprodotta.

Molti renderer cercano di essere fotorealistici, ovvero cercano di produrre un'immagine che sia indistinguibile da una fotografia. Questa è una caratteristica molto utile in certi scopi, ma non è sempre ciò che si vuole.

In alcuni casi si vuole riprodurre lo stile di un bozzetto, o di un dipinto, o di un disegno, o qualche altro stile non realistico. I renderer che permettono simili rendering applicano delle tecniche non fotorealistiche.

La maggior parte dei renderer sono principalmente fotorealistici, pur permettendo, in alcuni casi, di eseguire alcune tecniche non fotorealistiche. I renderer che invece sono studiati appositamente per produrre simili immagini sono in genere detti non fotorealistici, ma sono la minoranza.



Fig.1a - Renderer Fotorealistico

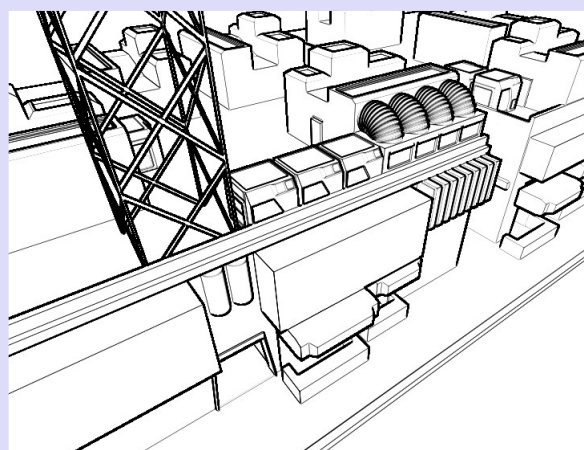


Fig.1b - Renderer Cartoon

## ILLUMINAZIONE GLOBALE



Fig.2a - Illuminazione Locale -Motore Interno



Fig.2b - Illuminazione Globale YafRay

Una variabile fondamentale, che accomuna specialmente i renderer fotorealistici, è il modo in cui viene gestita l'illuminazione.

Occorre definire due concetti per poter chiarire meglio in che modo i renderer possono differire nell'illuminazione. Usando il modello geometrico i raggi partono dalle fonti luminose ed urtano sui vari oggetti. Quando urtano su un oggetto una parte del raggio viene assorbita dall'oggetto stesso, un'altra invece viene riflessa in una qualche direzione. Una parte di ciò che viene assorbito dai sensori che rilevano l'immagine (ad esempio dagli occhi) viene utilizzata poi per realizzare l'immagine.

Qualora si considerino soltanto i raggi che compiono un percorso dalla sorgente luminosa all'oggetto e quindi alla camera si parla di illuminazione locale o diretta. Alcuni renderer sono in grado di mostrare solo questa, e così si dirà che quei renderer supportano solo l'illuminazione locale.

Più spesso alcuni renderer supportano solo alcuni percorsi di raggi, oltre quelli dell'illuminazione locale, magari per simulare i riflessi, o le rifrazioni, o le ombre. Quando un renderer considererà in qualche modo tutti i percorsi possibili parleremo d'illuminazione globale.

## RAGGI, PATCH E...

A questo punto è necessario cominciare ad illustrare qualcosa di tecnico. È indispensabile avere alcune conoscenze sul funzionamento interno dei renderer, per poter comprendere quali differenze ci sono. Al fine di comprendere le tecniche e le recensioni che verranno pubblicate successivamente si espongono ora alcuni concetti comuni a molte di queste.

Internamente un renderer può svolgere il suo compito in una moltitudine di modi diversi. Sebbene gli algoritmi di rendering utilizzati siano quasi sempre diversi anche tra renderer molto simili, questi si possono in genere classificare in poche categorie. Va detto però che alcune tecniche in diversi casi uniscono algoritmi che appartengono a più di una categoria, così è necessario comprendere che un renderer non per forza ricade in una sola delle categorie qui esposte. Inoltre non verranno esposte categorie rare, il cui utilizzo è sporadico.

In coda ad ogni tecnica esposta verranno offerti degli esempi. Si tratta solo di esempi e la lista non vuole essere autorevole o completa.



Fig.2b - Rasterizzatore - da Car Arena by Ize Design

Ray tracer: sono i renderer che più risultano naturali, in genere, poiché funzionano nel modo in cui noi pensiamo che funzioni la luce. Lanciano raggi e vedono dove sbattono e come. Spesso i raggi vengono originati dalle fonti luminose (come ci è naturale pensare) o più spesso dalle camere (dove arrivano e vengono assorbiti, ad esempio l'occhio). Si possono anche distinguere in deterministici (in cui il raggio va sempre nella stessa direzione) e stocastici (in cui il raggio cambia direzione in maniera pressoché casuale).

Esempi: blender internal, yafray, yaf(a)ray, mental ray®, indigo®, sunflow®... In realtà la quasi totalità dei renderer fotorealistici usano in qualche modo il ray tracing.

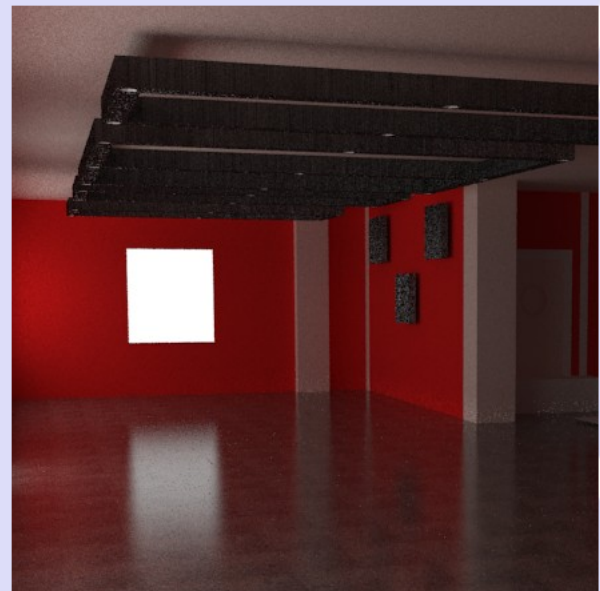


Fig.2a - Raytracer - Yaf[a]Ray

Rasterizzatori: sono i renderer probabilmente più semplici. Disegnano l'oggetto immediatamente, senza considerare l'interazione con altri oggetti né altro. Questi renderer sono meno comuni nella computer grafica fotorealistica, vengono usati specialmente per i giochi, ma comunque esistono alcune eccezioni in proposito, non così rare, generalmente rappresentate da renderer molto veloci.

Esempi: blender se disabilitato il ray tracing e la radiosity, lo scanline di 3d studio max, i motori grafici generalmente usati dai giochi.



Renderer basati sulla radiosity: In questi renderer l'intera scena viene divisa in piccoli elementi, chiamati patch, in cui viene ricorsivamente calcolata la radianza, ovvero l'intensità luminosa su ogni elemento. In pratica viene calcolata l'illuminazione locale tenendo in considerazione le fonti luminose, e per ogni elemento viene memorizzata l'illuminazione. Questi elementi diventano, a loro volta, fonti luminose, così viene ripetuto ancora il processo, e via dicendo fino a quando si ottiene un buon risultato. Questa è comunque una tecnica di supporto, poiché è necessario sia stimare il valore della radianza di una patch (che prima si faceva con tecniche molto complesse, ormai sempre più spesso con ray tracer stocastici) sia per mostrare la scena una volta finita.

Esempi: blender internal, povray. I renderer che usano la radiosity oggi giorno sono davvero pochi, più comuni renderer che usano tecniche ibride che discendono dalla radiosity.

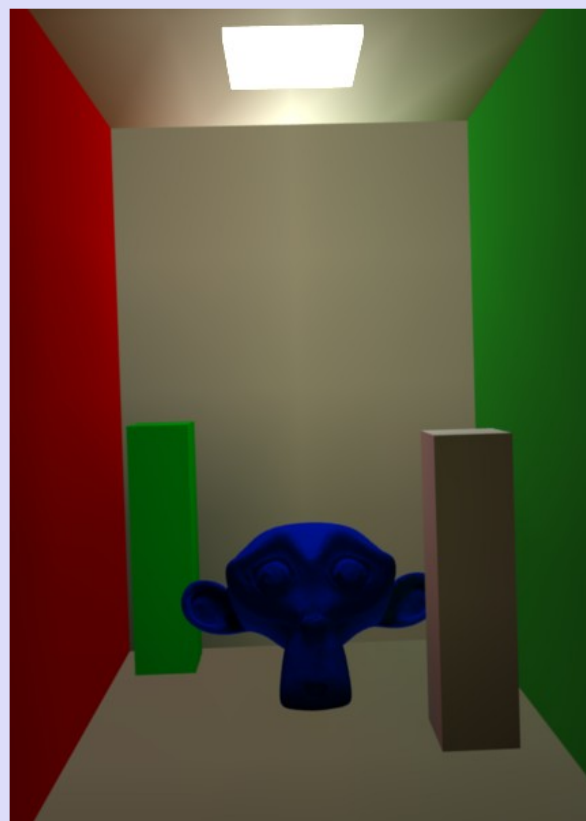


Fig.1c - Radiosity

Renderer che usano i micropoligoni: questa è una tecnica molto ricercata ed i renderer che la usano sono relativamente pochi, ma sono talmente importanti che non era possibile evitare di illustrarla. L'intera scena viene divisa in micropoligoni, che altro non sono che poligoni talmente piccoli che è possibile compiere delle ottimizzazioni notevoli su di loro. Ogni micropoligono rappresenta un elemento dell'immagine. Questi poi vengono disegnati molto velocemente con un processo simile a quello dei rasterizzatori.

Storicamente i renderer a micropoligoni creavano mappe per qualunque interazione tra gli oggetti, poiché supportavano solo l'illuminazione locale. Ultimamente molti renderer di questo tipo contengono un ray tracer ed alcuni persino qualche tecnica per il calcolo dell'illuminazione globale. Comunque la produzione di mappe per l'interazione tra gli oggetti rimane molto diffusa per questioni prestazionali.

Esempi: photorealistic renderman®, aqsis, pixie, 3delight®... Quasi tutti i renderer a micropoligoni usano l'algoritmo REYES®

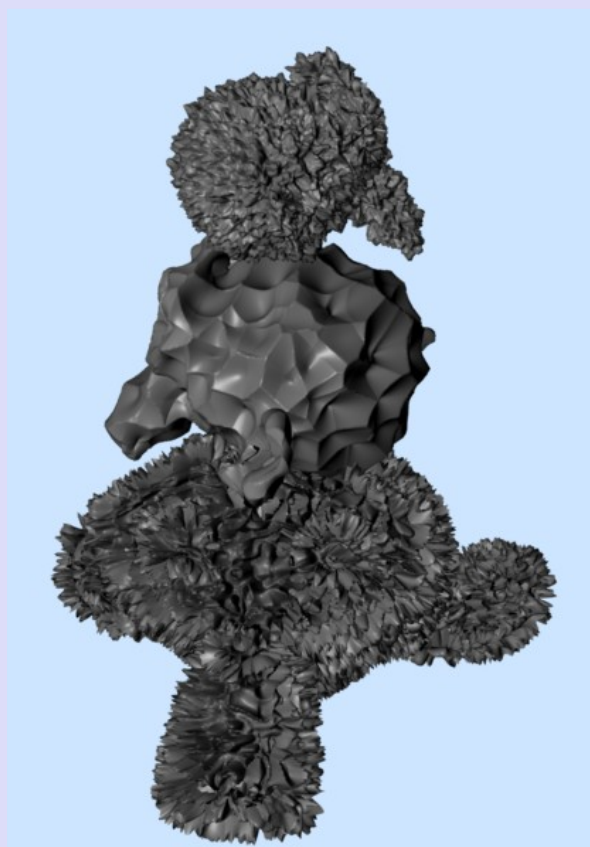


Fig.2d - Renderer con micropoligoni eseguito con Q-Dune

# SHADER PROGRAMMABILI

Come detto in precedenza dei renderer possono essere più flessibili di altri. Fino a quando però un renderer si basa su un numero ben definito di tecniche l'utente potrà sempre avere qualcosa che non è possibile rappresentare.

Alcuni renderer offrono la possibilità di usare shader programmabili, ovvero programmi che permettono di personalizzare il rendering.

Shader farebbe pensare che questi programmi descrivano un materiale, ma in realtà possono descrivere

qualunque cosa (il nome comunque è nato per i materiali).

Uno shader descrive come fare qualcosa. Non essendo un'opzione, ma uno script, è in grado di personalizzare completamente il processo di rendering per quanto concerne quella parte. Perciò in molti casi è conveniente avere diversi tipi di shader che permettano di personalizzare il processo di rendering in diversi punti.

## MATERIALI

I materiali possono essere di tanti tipo diversi in un renderer, ed in alcuni casi possono anche essere programmabili come abbiamo visto prima. In questo paragrafo si esporranno i più importanti al fine della descrizione degli algoritmi e se ne tenterà una classificazione.

**Materiali diffusi:** tendono a diffondere la luce abbastanza uniformemente in tutte le direzioni. Il più importante è quello perfettamente diffuso, che grossomodo (ma non esattamente) appare abbastanza simile ad un muro stuccato. Consiste in un materiale che riflette in tutte le direzioni ugualmente. I materiali diffusi che invece non sono sufficientemente vicini a quest'ultimo vengono detti glossy (patinati o lucidi), o, talvolta, semidiffusi.

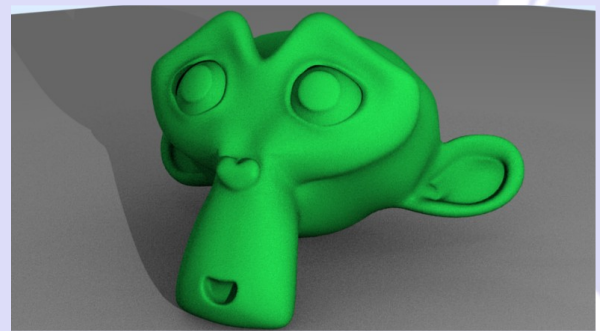


Fig.3a - Materiale diffuso

**Materiali speculari:** questi includono sia quelli che riflettono nella stessa emisfera dove cade il raggio (come gli specchi), sia quelli che possono riflettere anche indietro, causando la rifrazione (come il vetro). Anche qui risultano importanti quelli perfettamente speculari, che quindi invece di riflettere un raggio lungo diverse direzioni lo rifanno lungo soltanto una.



Fig.3b - Materiale Speculare

**Materiali retro-speculari:** sono rari nella realtà, un esempio sono i catarifrangenti. La loro caratteristica è quella di riflettere indietro i raggi, in grossomodo la stessa direzione, ma opposti in verso, del raggio che li genera.





## CAUSTICHE

Fig.4a - Esempio di Caustiche

Immagine di Cretox [www.inventati.org/cherudek/](http://www.inventati.org/cherudek/)

Particolarmente importanti nella valutazione di un renderer, poiché non sempre correttamente rappresentati ma molto visibili in certe scene correttamente renderizzate, sono quei raggi che rimbalzano da una luce attraverso una o più superfici speculari per poi arrivare su una superficie diffusa. L'illuminazione che si crea sul materiale diffuso viene chiamata caustica.

## TECNICHE IBRIDE

Qua verranno esposte le tecniche ibride più comuni, mentre le altre verranno esposte quando necessario.

Photon mapping: consiste nel memorizzare in una mappa una parte delle informazioni sulla radianza. In questo assomiglia alla radiosity ma ci sono diverse differenze. La principale è che il processo di campionamento della radianza non avviene con le patch, ma viene salvata una mappa di punti campionati con l'emissione di alcuni raggi. Quindi si tratta di una tecnica ibrida tra ray tracing e radiosity. Inoltre non è detto che si debbano salvare tutte le informazioni. Si può scegliere quali punti memorizzare a seconda di cosa si vuole rappresentare, ad esempio alcuni renderer creano le caustic map, che memorizzano solo le caustiche.



Fig.4a - Mappa di Fotoni -YafRay

Ambient occlusion: tecnica che approssima tutti i materiali come diffusi e precalcola l'occlusione (per le ombre) degli oggetti. È una tecnica che dà una buona approssimazione della luce in scene senza molti oggetti con materiali troppo distanti da quelli diffusi, ed è particolarmente veloce, ma reagisce male negli altri casi. Tuttavia i casi in cui funziona bene sono una grande percentuale.

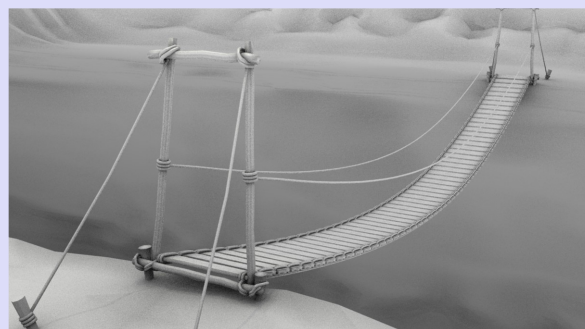


Fig.4b - Ambient Occlusion

Irradiance caching: vengono salvati dei campioni nella scena, così se un raggio cade vicino ad un altro che è stato già valutato non bisogna ricalcolare nulla. Vengono usate delle stime che riescono, in molti casi, a capire se l'errore dovuto a riutilizzare vecchi campioni è troppo elevato e così bisogna calcolare un nuovo campione o se è basso e così si può riutilizzare il vecchio. Viene spesso combinato col photon mapping.

I renderer fotorealistici si distinguono anche per una serie di effetti che possono supportare. Qua sono esposti quelli più importanti:

## OLTRE I MATERIALI

Media di partecipazione: la luce non sbatte necessariamente solo sulle superfici, e non si genera per forza solo sulle superfici. Un materiale, come ad esempio l'aria, può dare collisioni coi raggi potenzialmente in ogni punto, e del fuoco, ad esempio, può produrre luce in ogni punto. Effetti volumetrici di ogni tipo possono venire simulati coi media di partecipazione.

Subsurface scattering: un raggio che entra dentro un oggetto non è detto che esca collidendo solo lungo le superfici, può collidere anche all'interno. Questo si può simulare coi media di partecipazione, ma è generalmente proibitivamente lento, perciò molti renderer supportano delle approssimazioni utili a velocizzare il processo.

Tone mapping: in genere il dispositivo che mostra l'immagine non è in grado di visualizzare tutti i colori renderizzati. Spesso se si visualizza un'immagine i valori risulteranno falsati, in molti casi perché i valori troppo scuri risulteranno neri e quelli troppo chiari bianchi. Per trasformare lo spazio di colori di un'immagine in qualcosa visualizzabile abbastanza correttamente si può applicare un filtro, chiamato tone mapping.

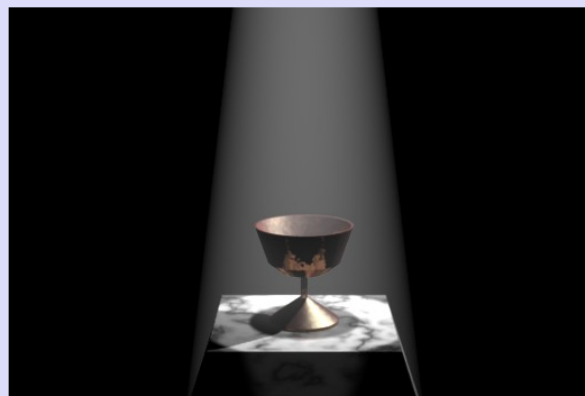


Fig.5a - Luce volumetrica

Media non costanti: normalmente i raggi viaggiano dritti tra una collisione e l'altra. Questo non è vero nel caso in cui l'indice di rifrazione cambia nello spazio. Alcuni renderer permettono di deviare i raggi nel vuoto, e questo può essere utile nel caso in cui si vogliano simulare effetti come i miraggi o le fata morgana.

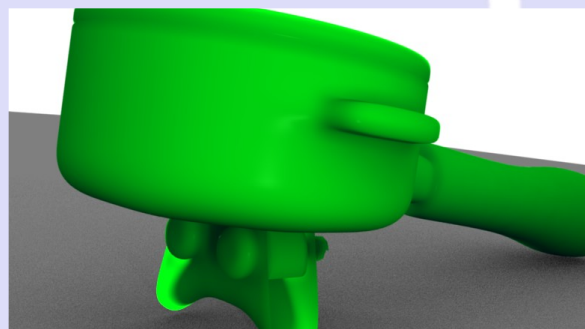


Fig.5b - SubSurface Scattering

## CONCLUSIONI

Da questa breve introduzione risulta probabilmente maggiormente chiaro che la scelta di un renderer per i propri lavori non è banale. Ci saranno spesso pro e contro per molti dei renderer a disposizione e la scelta è in genere molto personale. Proprio per questo gli articoli che seguiranno cercheranno di dare tutti gli elementi per poter prendere una decisione informata e cosciente sul problema.

## APPROFONDIMENTI

Per approfondimenti è possibile visitare i seguenti collegamenti:

[http://en.wikipedia.org/wiki/Rendering\\_\(computer\\_graphics\)](http://en.wikipedia.org/wiki/Rendering_(computer_graphics)) (ed i vari collegamenti)  
<http://graphics.pixar.com/Reyes/> (spiegazione dettagliata di come reyes funziona)  
[http://en.wikipedia.org/wiki/Rendering\\_equation](http://en.wikipedia.org/wiki/Rendering_equation) (un'equazione fondamentale che descrive il modo in cui la luce si muove nella scena)

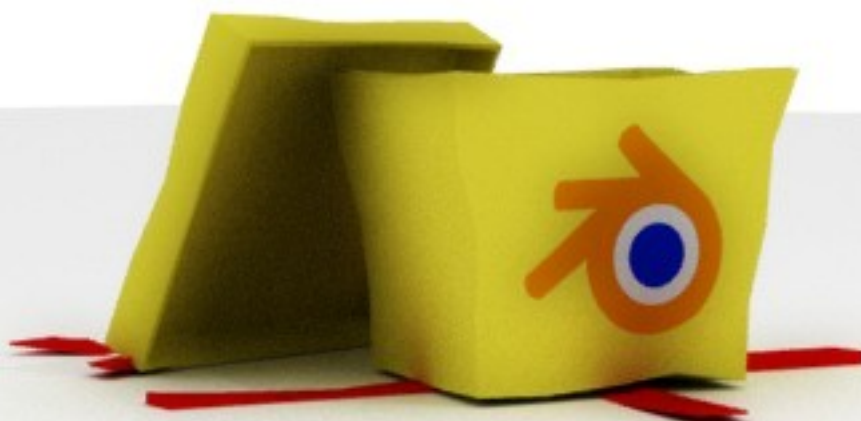
Tutti i marchi sono registrati ai loro legittimi proprietari. @



La nuova versione di Blender si prospetta piena di succose novità che molti utenti aspettavano, altri avevano richiesto e altri ancora resteranno semplicemente sorpresi dai passi in avanti che questo software sta facendo.

Molti di questi nuovi tools si devono al Peach Team, che in questo periodo, capitanati da Ton, sta portando avanti la realizzazione di un cortometraggio Open Source.

# Novità Per Blender



## Texture in Edit Mode

Grazie ad una richiesta diretta del Peach Team, è ora possibile visualizzare in Edit mode le texture sulla mesh, cosa che prima era possibile solo in object mode. Questo permetterà di poter aggiungere particolari in maniera più precisa facendo riferimento direttamente alla texture.

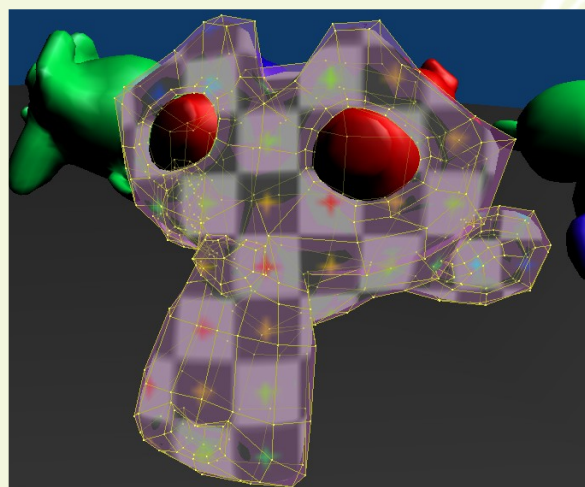


Fig1a. Un esempio di 3d view in Edit Mode con Texture applicata e il Viewport Shading in modalità Texture.

## MeshDeform Modifier

Il modificatore MeshDeform è un altro regalo dal Peach Team. Questo modificatore è simile al Lattice, solo che invece di far riferimento ad un oggetto "Lattice", fa riferimento ad una mesh qualsiasi.

L'utilità di questo modificatore è quella di aiutare l'animazione di personaggi corpulenti, creando un involucro esterno molto lowpoly della mesh principale e durante l'animazione basterà muovere i vertici di tale involucro che, deformandosi, deformerà la struttura della mesh del. Senza il MeshDeform sarebbe stato necessario un complesso sistema di ossa come quello per la simulazione dei muscoli.

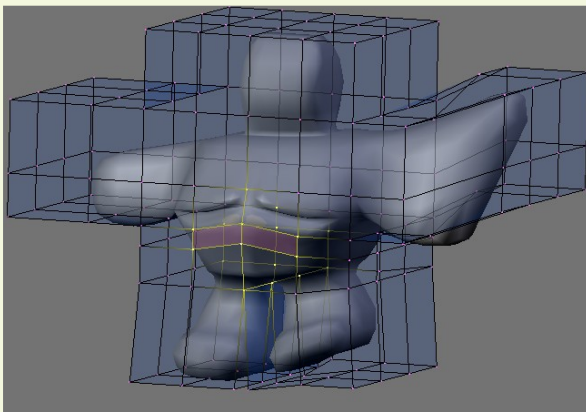


Fig1b. La gabbia che ricopre una mesh più complessa e ne controlla le deformazioni, questa è la potenza del MeshDeform.

## Bone Heat

Rimanendo in campo di animazione, voglio segnalare un utile strumento che potrebbe far risparmiare ore di lavoro quando si ha a che fare con Armature e Weight Paint. Si tratta dello skinning automatico, chiamato Bone Heat.

Dalla prossima versione di Blender, sarà possibile assegnare in maniera automatica i pesi dei vertici ai gruppi creati quando si imparenta un Armature ad una Mesh.

La procedura è molto semplice, creato il nostro personaggio e la sua armatura li imparentiamo, e quando ci viene richiesto come procedere con lo skinning, basta scegliere "Create From Bone Heat" e tutti i gruppi di vertici creati dai nomi delle ossa, avranno la giusta pesatura per poter iniziare ad animare il nostro personaggio quasi

immediatamente. In oltre dal pannello dei modifier, rendiamo l'armatura un modificatore e attiviamo Quaternion per ottenere delle pieghe più realistiche soprattutto per gomiti e ginocchia.

C'è da dire che per poter usare al meglio questo nuovo strumento è necessario avere una Mesh molto pulita e con vertici non troppo vicini. In alternativa, se proprio non riuscite a farlo funzionare, fare un Remove Double dovrebbe risolvere il problema.

## Color Range for Weight Paint

Una nuova opzione appare nel pannello "User Preferences" nella sezione "System and OpenGL". Si tratta del "Color Range for Weight Paint", dove è possibile personalizzare i colori che distinguono il peso dei vertici. Di default Blender utilizza un sistema cromatico che va dai colori freddi a quelli caldi per simulare il minore e maggiore peso dei vertici, più precisamente si parte da Weight 0 che equivale al colore Blu, per poi passare al Verde, il Giallo, l'Arancione e il Rosso che ha Weight 1.

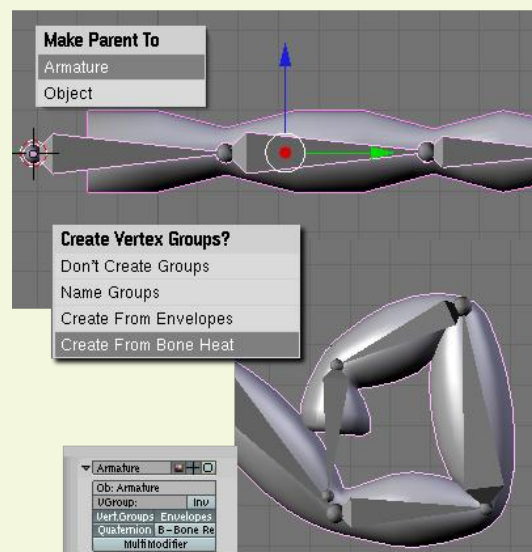


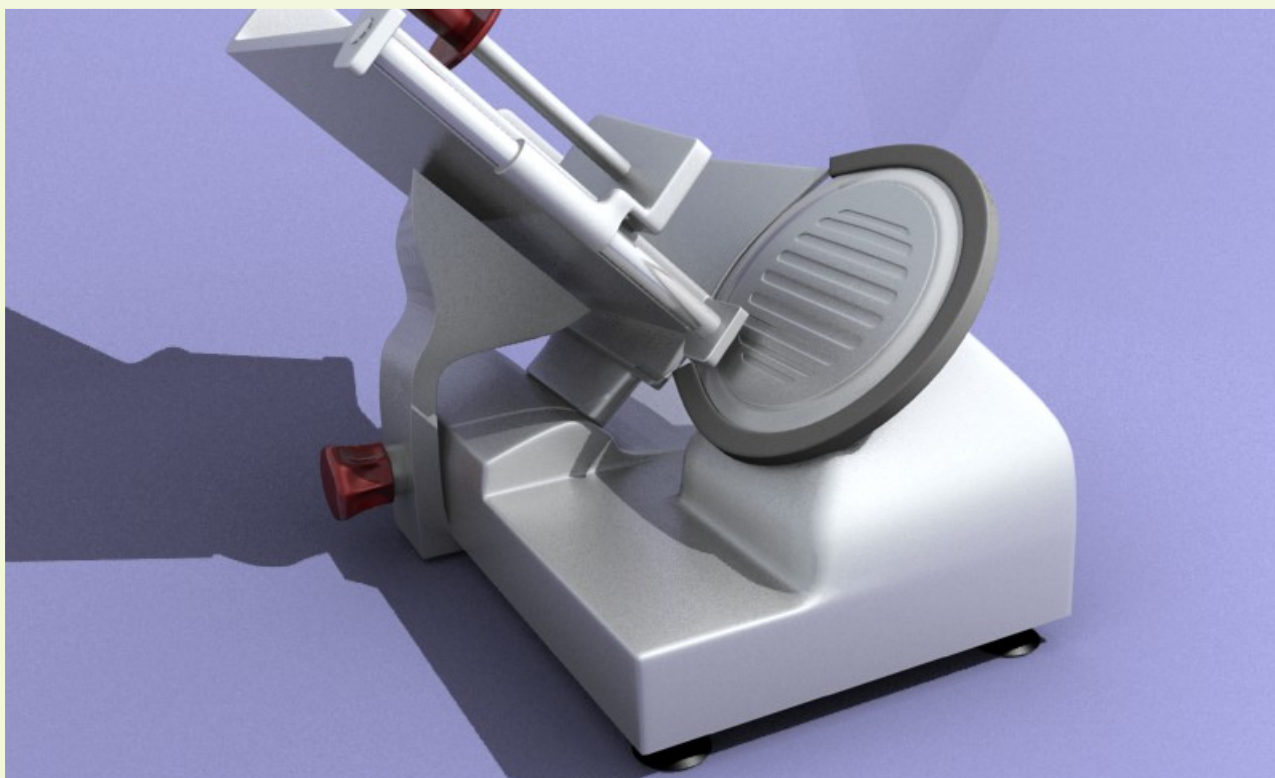
Fig1c. Due passaggi e il nostro modello è pronto per l'animazione con il Bone Heat.



La novità più importante è l'attesissima introduzione del sampling QMC (Quasi Montecarlo) nel motore di rendering interno. Per noi profani va detto semplicemente che questo tipo di campionatura permette al motore interno di avvicinarsi sempre di più ai motori di rendering commerciali, ma anche al suo rivale sul campo YafRay.

Per la precisione, sarà possibile scegliere tra Adaptive Sampling e Constant Sampling. La differenza principale è che mentre il Constant Sampling esegue un campionamento identico per tutto il rendering, l'Adaptive Sampling userà un campionamento variabile che si adatta al diverso tipo di illuminazione. La soglia di campionamento può essere decisa mediante il parametro Threshold disponibile nelle nuove funzioni introdotte e nei parametri del Ambient Occlusion, che altrimenti impiegherebbero tempi biblici per il calcolo del rendering.

## Le Meraviglie del QMC



### Ombre Sfocate ►

Finalmente anche le altre Light potranno proiettare ombre sfocate come attualmente fa l'Area Light. I parametri sono abbastanza semplici, possiamo scegliere il tipo di campionamento (Adaptive o Constant), la dimensione della sfocatura (Soft Size), la precisione di campionamento (maggiore sarà il parametro Samples e maggiore sarà la precisione), ed in fine il parametro Threshold, che permette di controllare il livello di Adaptive Sampling per le ombre. C'è da aggiungere che i tempi di rendering aumentano parecchio giocando con questi parametri,

soprattutto aumentando il livello di samples e usando il Constant Sampling.

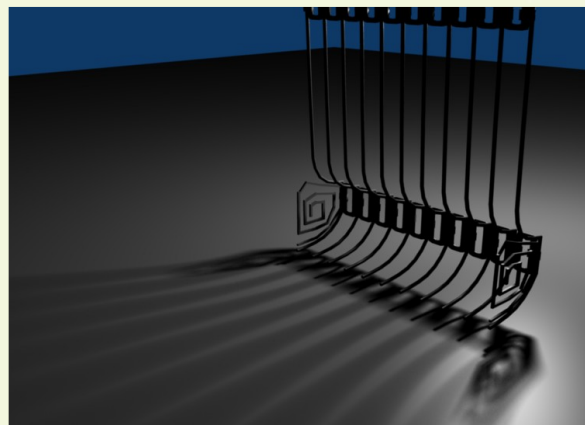


Fig2a. Una Lamp con Samples a 16 e Soft Size a 1. Per avere una sfumatura più grande aumentate il Soft Size.

## Riflessioni Sfocate

Le riflessioni sfocate (BRDF), prerogativa di Yafray (tramite settaggio da XML) e del nuovo Yaf[a]Ray, saranno disponibili anche nel motore interno di Blender. Questo tipo di riflessioni sono molto più realistiche e permettono di simulare meglio i materiali riflettenti come l'acciaio lavorato o più semplicemente la grande maggioranza di materiali che pur essendo opachi hanno delle riflessioni e rifrazioni (per i materiali trasparenti).

Anche in questo caso i parametri da settare non sono molti, se qualcuno ha già usato il pannello "Mirror Transp" noterà dei piccoli cambiamenti. Il principale è il parametro "Gloss", settato di default a 1 (cioè riflessione totale). Per valori inferiori a 1 avremo riflessioni sempre più sfuocate, in oltre, come per le ombre, possiamo settare il valore di campionamento e la sua soglia (Samples e Threshold), ricordandoci che valori alti vogliono dire alti tempi di rendering. Il parametro Depth indica il numero delle riflessioni interne, un valore che è bene aumentare solo per primi piani o superfici trasparenti, in quanto anche in questo caso i tempi di rendering aumentano.

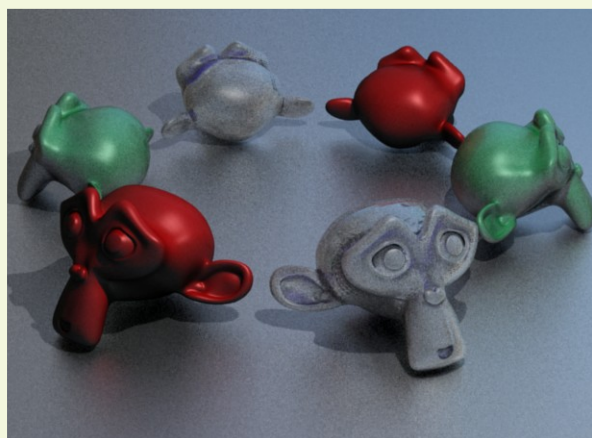
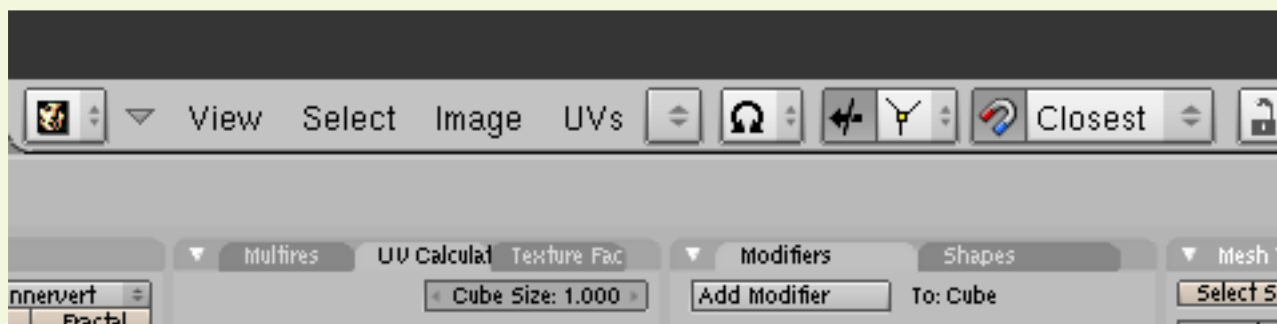


Fig2b (in alto). Suzanne in posa per mostrarci la potenza delle BRDF.

Fig2c (in Basso). Il pannello dell' UV/Image Editor rinnovato e arricchito con nuovi strumenti, tra cui il tanto desiderato Snap magnetico.



In fine c'è un parametro, "Max Dist", che serve a calcolare le riflessioni a seconda delle dimensioni della scena ed seguito da un menù a tendina dove è possibile scegliere tra "Fade to Sky Color" e "Fade to Material Color", che stanno ad indicare se la scena è ambientata in un esterno (Sky Color) o un interno (Material Color).

## UV Map

La modalità "UV Face Select" è stata eliminata, sarà ora possibile utilizzare il normale "Edit Mode" per selezionare i Seams Edge e unvrappare, senza dover necessariamente passare in un'altra modalità. Questo non è l'unica novità per gli unvrappatori accaniti, infatti è stato notevolmente migliorato l'UV/Image Editor. Tra le nuove opzioni, c'è l'attesissimo Snap calamitato e un cursore simile a quello della 3d view, con possibilità di eseguire Snap su di esso con la combinazione SHIT+S. Anche la modalità di selezione è stata migliorata, sono infatti possibile quattro scelte diverse. Deselezionando il pulsante "Mesh Indipendant Selection", sarà possibile avere la selezione di Facce e Vertici in comune con quelle dell'Edit Mode. Attivando il pulsante "Mesh Indipendant Selection", possiamo scegliere se selezionare i vertici attaccati (Shared Location), staccati (Disable) oppure in coppia con gli altri vertici (Shared Vertex).



## Normal Baking

Il Normal Bake è stato notevolmente migliorato, ora c'è la possibilità di scegliere quattro modalità diverse di cottura:

Camera, quella classica, le normali sono calcolate in base alla Camera, questo significa che non è possibile usare il Bake per animazioni, ma solo per pose statiche.

World: se l'oggetto viene animato o

deformato, il Bake non è più valido. Object: le normali vengono calcolate sull'oggetto e quindi il Bake rimane valido per trasformazioni, ma non per deformazioni.

Tangent, con il quale il Bake rimane invariato sia per deformazioni che trasformazioni.

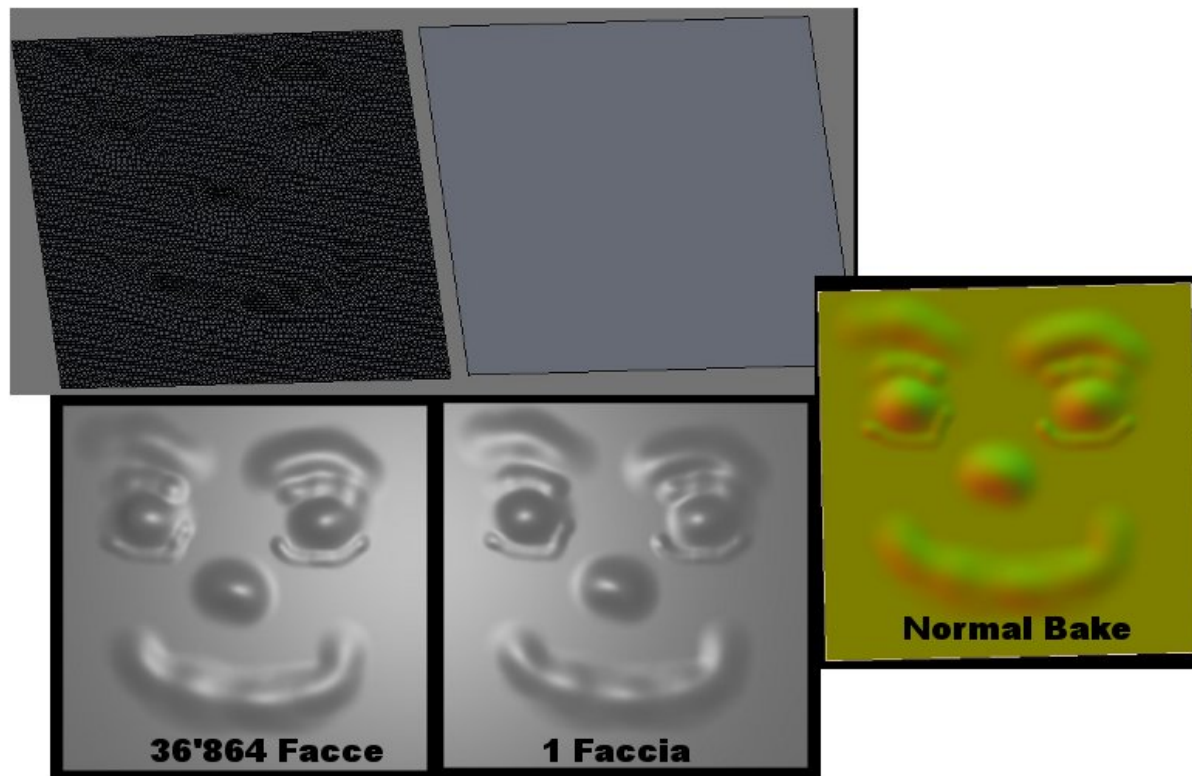


Fig2d. Come potete vedere in figura n alto, le due mesh a confronto, quella a sinistra, realizzata mediante Sculpt Tool e dalla quale è stata ricavata la Normal Map tramite il Bake. Quella a destra ha una faccia, ma con la Normal Map applicata, non si nota la differenza, tranne che per i tempi di rendering.

## Conclusioni

Come già detto, molte di queste modifiche sono state implementate su richiesta del Peach Team, quindi non ci dovremo meravigliare se sono tutti miglioramenti fatti per migliorare Blender per la realizzazione di animazioni computerizzate. Questo non vuol dire che non si stia lavorando in altre direzioni, per esempio, è stato appena formato un team sull'onda del Peach Team, per realizzare un videogioco open source. Quindi è probabile che tra qualche mese ci siano nuovi interessanti sviluppi per il nostro amato frullatore.

Per provare tutte queste nuove funzioni, è possibile scaricare l'ultima SVN di Blender dal sito: [www.graphical.org](http://www.graphical.org). Sono disponibili versioni già compilate per sistemi Windows, Linux e MacOS.

Per gli smanettoni di Linux, sono disponibili i sorgenti da compilare sul sito [www.blender.org](http://www.blender.org) .@

Anfeo

Questa rubrica tratterà passo passo tutte le fasi di lavoro che Designer/Architetti/Geometri o comunque professionisti del settore, utilizzano per passare da una piantina ad un Rendering di resa fotorealistica. Il tutto adattato per il funzionamento su Blender.

Partiamo dal presupposto che Blender non è un CAD, ma un modellatore poligonale. La differenza sostanziale, sta nel fatto che gli mancano strumenti parametrici, snap complessi e unità di misura. Quindi per ora scordiamoci di progettare la casetta dei nostri sogni direttamente in Blender, senza usare qualche piccola scorciatoia.

Vedremo infatti che niente è impossibile utilizzando gli script giusti, un po' di ingegno e soprattutto prendendo spunto dal flusso di lavoro di chi usa le accoppiate CAD+3dMax+VRay.



Tutorial

# Rendering Architettonici 1° parte Modellazione

Verranno proposte diverse soluzioni per lo stesso problema, sta all'utente scegliere quale sia la più adatta. Soprattutto per la scelta del motore di Rendering, che determinerà il compromesso finale tra Resa e Tempi.

In questa parte tratteremo le fasi iniziali, importazione e modellazione. Questa non vuole essere una guida alla modellazione, molte cose si daranno per scontate, verranno solo mostrate alcune soluzioni a problemi comuni che si riscontrano in questi progetti.



## Partenza

Bisogna realizzare il rendering fotorealistico di un edificio, vi viene consegnato il progetto, che altro non è che un file CAD che contiene le piante dei piani dell'edificio, o addirittura la sua struttura 3d.

Assicuratevi di aver capito le aspettative del cliente, normalmente nel progetto che vi viene dato non ci sono riferimenti per il tipo di arredamento esterno, quindi bisogna chiedere bene che tipo finestre, ringhiere e quanto altro vuole per il suo Rendering.

Prima di tutto bisogna cercare di importare il file in Blender. L'estensione più comune è quella .DXF di Autodesk, ma se riuscire a reperire dei file .OBJ o .3ds, potreste avere le cose più facilitate in seguito.

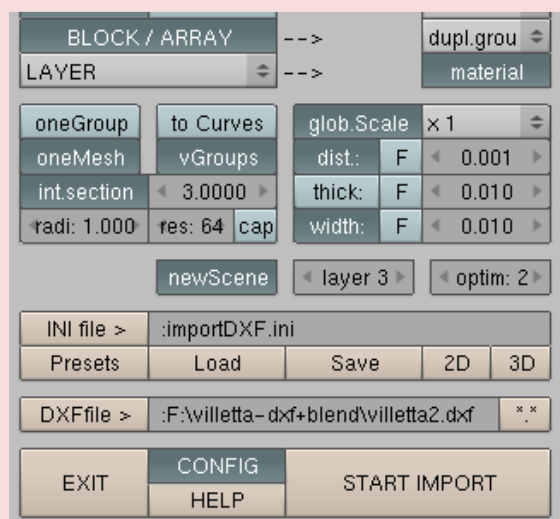
Esaminiamo il file in questione e a questo punto ci sono diverse strade che si possono seguire:

1. Utilizzare i modelli 3d del file, rattoppando qualche faccia mancante, rovesciando normali ecc... In pratica ripuliamo la mesh che durante l'importazione ha subito qualche errore di interpretazione.
2. Utilizziamo gli snap di Blender e cerchiamo di ricostruire il modello 3d, avendo come riferimento quello importato, che presenta troppi errori di Topologia che risulterebbero nel rendering finale.
3. Non ci viene dato nessun modello 3d, solo le viste prospettiche e le piante dei piani. A questo punto ci tocca un lavoro completo di rimodellazione, ma senza un oggetto 3d come riferimento, ma solo linee e curve.

Siccome per i primi due punti basta un semplice lavoro di riparazione e copiatura, ci occupiamo del terzo punto, ovvero costruire l'edificio usando come riferimento le linee del modello importato. Potete utilizzare il file "villetta2.dxf" in allegato alla rivista.

### 1. Importazione

Iniziamo con l'importare tutte le viste in un unico file di Blender. Andiamo su File>Import>Autodesk DXF (.dxf), poi, trattandosi di un 2d, clicchiamo sul pulsante 2d per settare in automatico l'importer e clicchiamo su Start Import.



### 2. Pulizia

Ci ritroviamo con le nostre viste 2d, uniamole con CTRL+J in un unica mesh, andiamo in Edit Mode, selezioniamo tutto e facciamo un RemDouble per eliminare i vertici doppi.

In oltre eliminiamo testi, quote e tutto quello che non ci interessa.

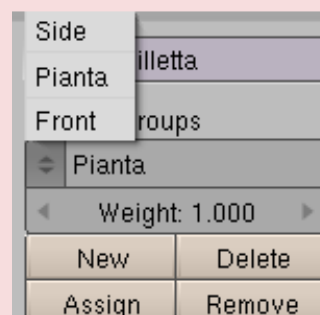
Potreste essere costretti a dover ridurre le dimensioni delle viste importate.



### 3. Ordine

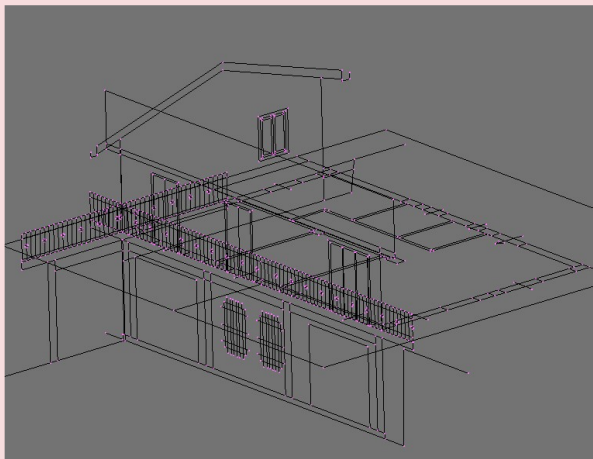
Abbiamo quindi una mesh divisa in 3 viste (che possono anche essere di più a seconda dei piani e della complessità del progetto). Selezioniamo una vista alla volta e assegniamo un gruppo di vertici ciascuno, in questo modo avremo la vita più semplice in seguito.

Facciamo questo perché al momento non è possibile usare lo Snap in Object mode (vedremo in seguito meglio), quindi dobbiamo lavorare in Edit Mode nello stesso Oggetto.



#### 4. Preparare la scena ▶

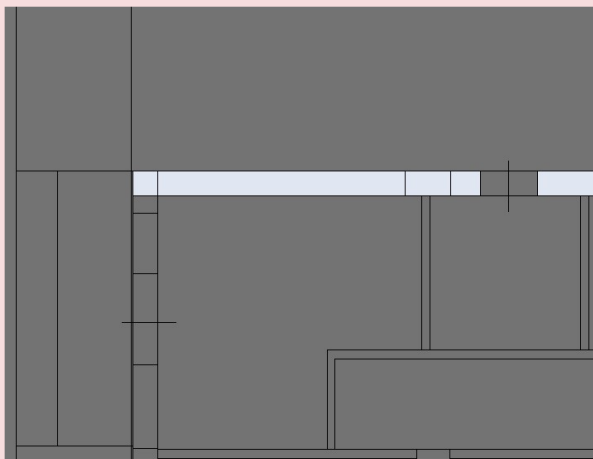
Selezioniamo, spostiamo e ruotiamo le viste in modo da farci un'idea di massima dell'ingombro totale. Attivando lo Snap (pulsate a forma di calamita) e tenendo premuto il tasto CTRL mentre spostiamo, i vertici si agganceranno automaticamente a quelli più vicini (se è selezionato Closet) o si agganceranno tramite il centro (se è selezionato Center).



#### 5. Mura ▶

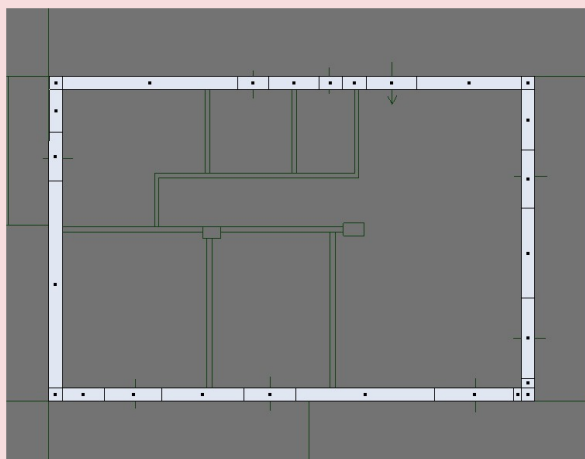
Andiamo quindi in vista Top dove avremo la pianta del piano che useremo come ricalco per costruire le Mura.

Aggiungiamo un oggetto Plane (sempre in Edit Mode) e assegniamogli un nuovo Gruppo di Vertici, chiamiamolo pure Mura. Ora scaliamo e spostiamo i vertici del nostro Plane e adattiamolo al primo pezzo del perimetro delle mura.



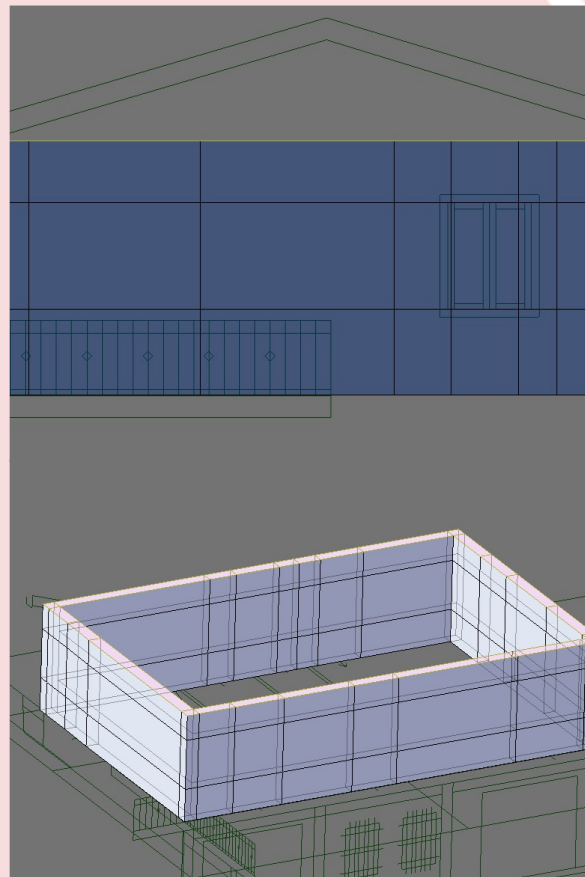
#### 6. Estrudere lungo il perimetro ▶

Lavorando solo con gli edge ed estrudendoli di volta in volta, seguiamo tutto il perimetro delle mura. In questa fase le strade possono essere molte, possiamo ricalcare fedelmente tutto il perimetro lasciando degli edge a segnare le porte e le finestre, oppure lasciare degli spazi vuoti, noi useremo il primo metodo.



#### 7. Estrusioni ▶

Finito il perimetro estrudiamo le mura in diversi momenti facendo riferimento alle altezze delle finestre e delle porte, in modo tale da avere a disposizione delle facce che riproducano tali fessure.



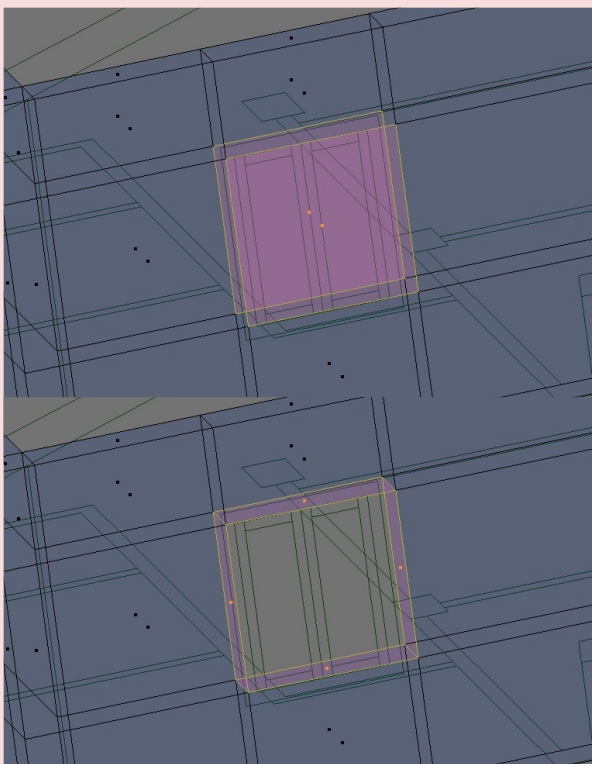


## Consiglio:

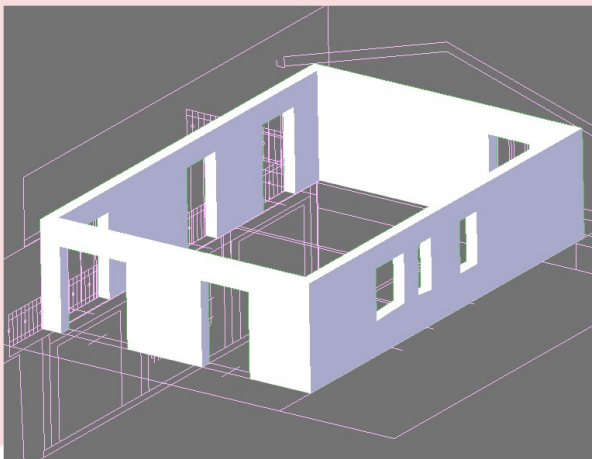
A questo punto possiamo anche separare le mura dalle viste rendendole un oggetto indipendente, basta premere il tasto P. Volendo è possibile mantenere questa unione per avere più precisione durante la lavorazione (e usare lo snap), il tutto dipende dal livello di complessità e dalla precisione che vogliamo ottenere.

## 8. Porte e Finestre

Selezioniamo le facce parallele (esterne e interne) in corrispondenza delle finestre e premiamo il tasto F, selezionando "Skin Face/Edge Loop". Ricordo che funziona solo per un gruppo di facce alla volta.



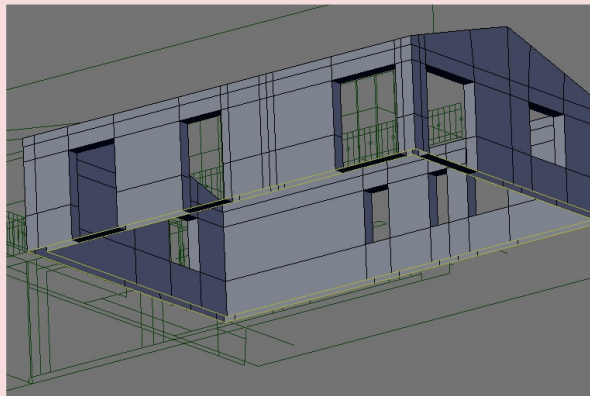
Facciamo lo stesso anche per le altre finestre e porte.



## 9. Pian Terreno

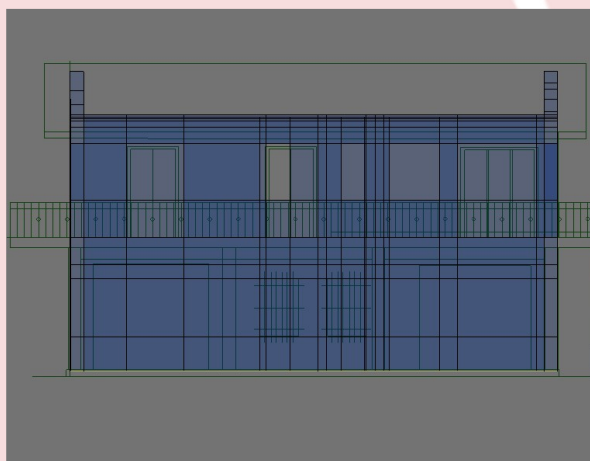
Estrudiamo un livello per il tetto, rendendolo a punta e alla fine abbiamo ottenuto il primo piano.

Non resta che fare il pian terreno, quindi prima di tutto cancelliamo tutte le facce del fondo, tranne quelle delle porte, quindi selezioniamo gli Edge esterni ed interni ed estrudiamo verso il basso.



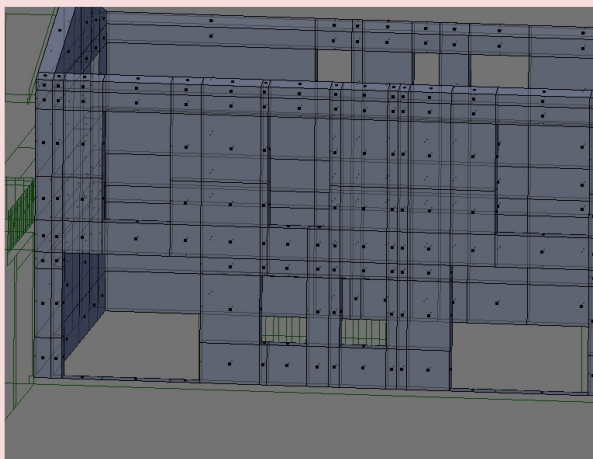
## 10. Estrudere in basso

Dobbiamo estrudere a diverse altezze, per avere una corrispondenza con le finestre e le porte del garage.



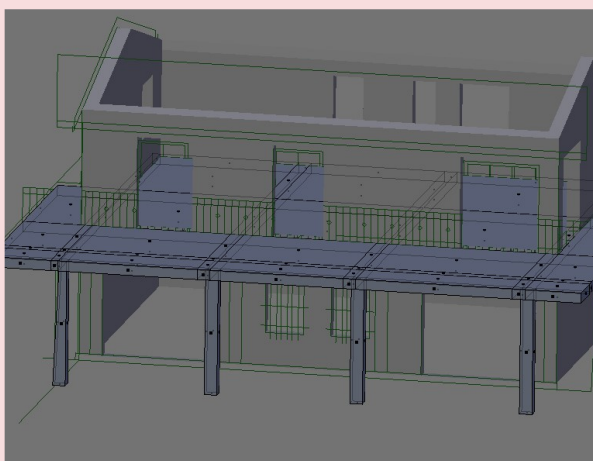
## 11. Garage e Finestre

Aggiungiamo degli edge loop (CTRL+R) verticali per ricalcare le porte del garage e con la tecnica di prima le foriamo. Se quando aggiungiamo degli edge loop la mesh forma dei triangoli o facce strane, è probabile che ci siano delle facce interne da qualche parte, quindi trovattele ed eliminatele.



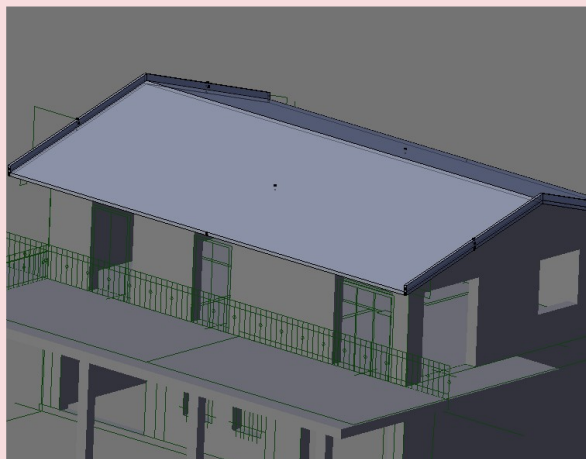
## 12. Pavimento

Per i pavimenti basta un semplice parallelepipedo che farà anche da balcone e vialetto. Estrudiamo i piloni aggiungendo dei loop in corrispondenza e i pavimenti sono a posto.



## 13. Tetto

Per il tetto le cose sono un tantino più complicate, ma basta seguire le linee e modellarlo di conseguenza. L'importante è lasciare lo spazio per le tegole.



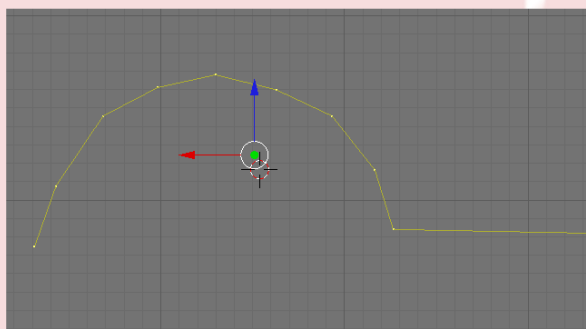
## Coppi

Per ora vedremo solo la realizzazione dei coppi del tetto, non perché siano difficili da fare, ma perché saper utilizzare l'Array può essere utile in molte occasioni.

## 14. Profilo

*N.B. Avere delle immagini di riferimento è sempre molto utile, ci aiutano a darci un'idea di quello che stiamo facendo.*

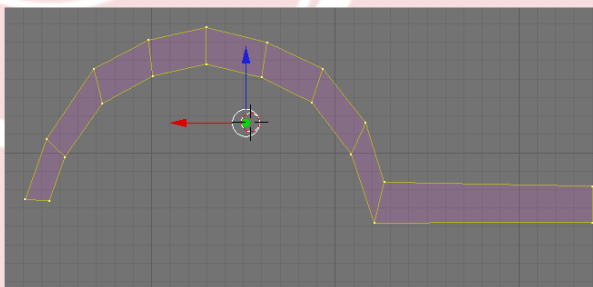
Partiamo dalla vista frontale sventriamo un cubo e teniamoci solo un vertice, dal quale ricaveremo una linea tipo questa.





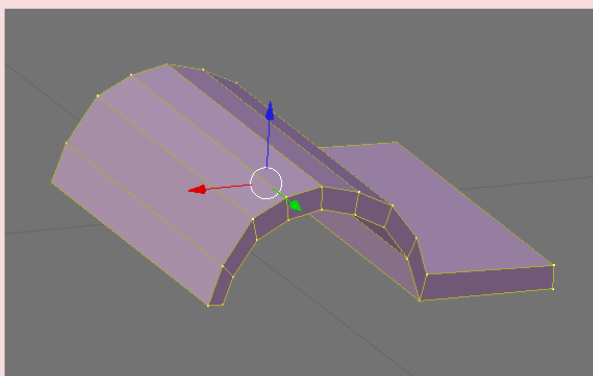
## 15. Spessore al Profilo

Selezioniamo tutto ed estrudiamo, scaliamo un po' e otteniamo una superficie del genere.



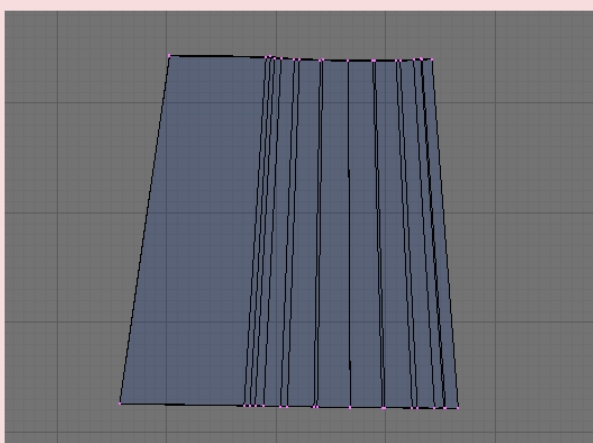
## 16. Estrudere il coppo

Riselezioniamo tutto, passiamo in vista Top e riestrudiamo le facce ottenendo il nostro bel coppo.



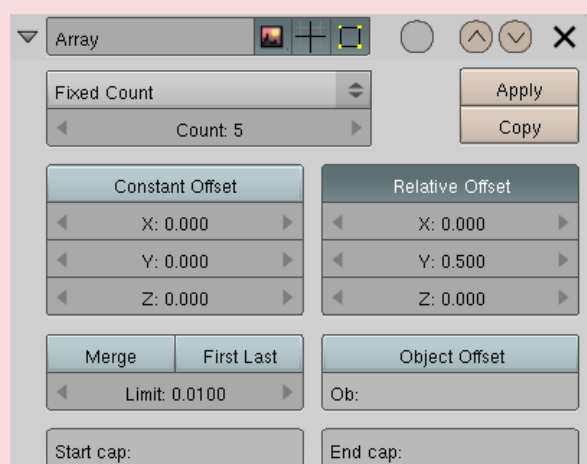
## 17. Dare una forma

Sempre in vista dall'alto, scaliamo un'estremità creando una specie di trapezio.



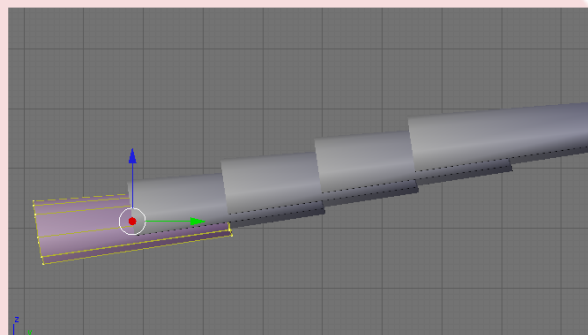
## 18. Array

Diamo un Set Smooth per arrotondare le normali. Non serve applicare un SubSurf a meno che non si facciano dei primi piani delle tegole, in questo modo risparmiamo sui poligoni. Aggiungiamo un modificatore Array e settiamolo in questo modo o comunque in modo che si accavalli con quelli successivi.



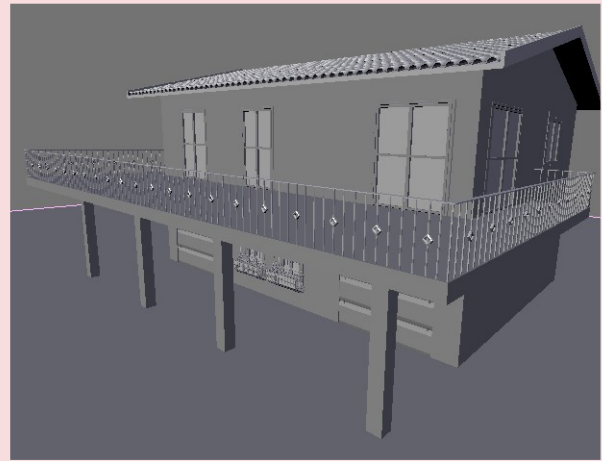
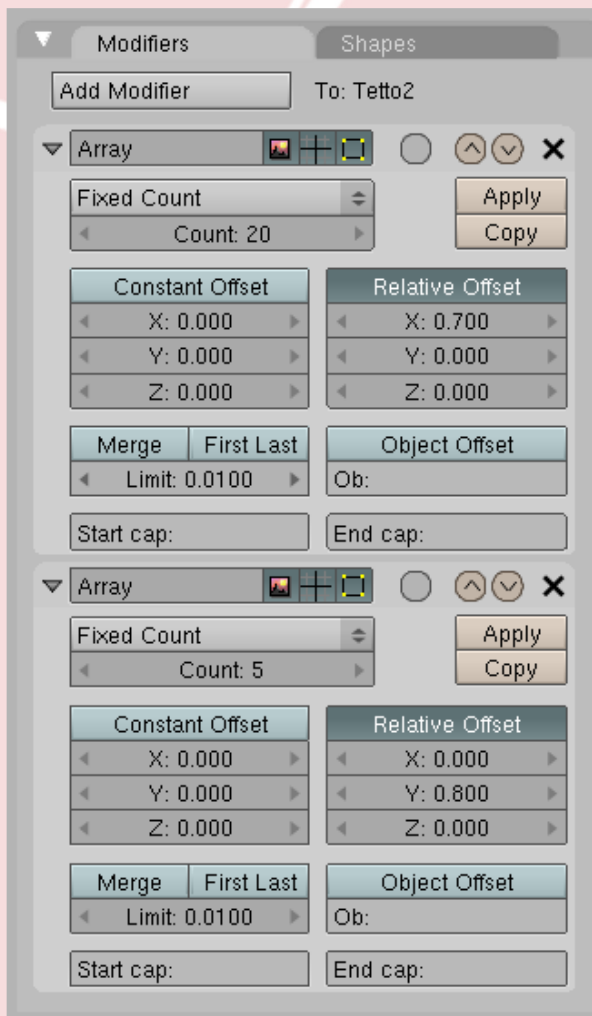
## 19. Accavallamento

Dobbiamo ruotare (in Edit Mode) il coppo, per fare in modo che l'accavallamento sia più preciso.



## 20. Un altro Array ►

Aggiungiamo un altro Array, ma questa volta settato per duplicare i coppi lateralmente.



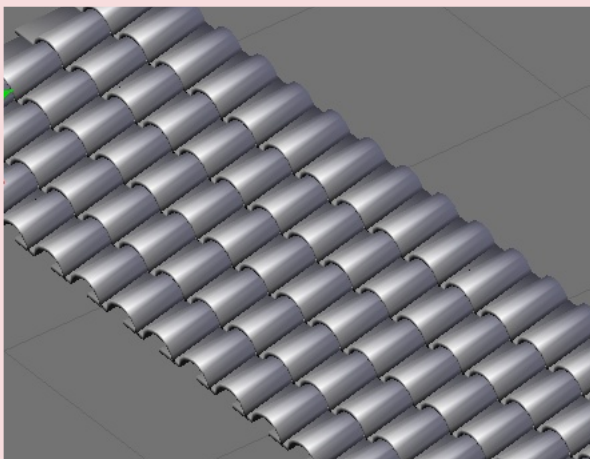
## Conclusioni ►

Crearsi una libreria personale di oggetti architettonici può far molto comodo e soprattutto risparmiare tempo per i progetti futuri. In rete si trovano moltissimi modelli gratuiti o a pagamento, ma è sempre bene archiviare in un file separato tutti gli oggetti che si creano, dando un nome comprensibile e rinominandoli per categoria. Vi ritroverete in poco tempo con tutto il necessario per poter dedicare più tempo all'edificio e al rendering.@

Anfeo

## 21. Ultime Regolazioni ►

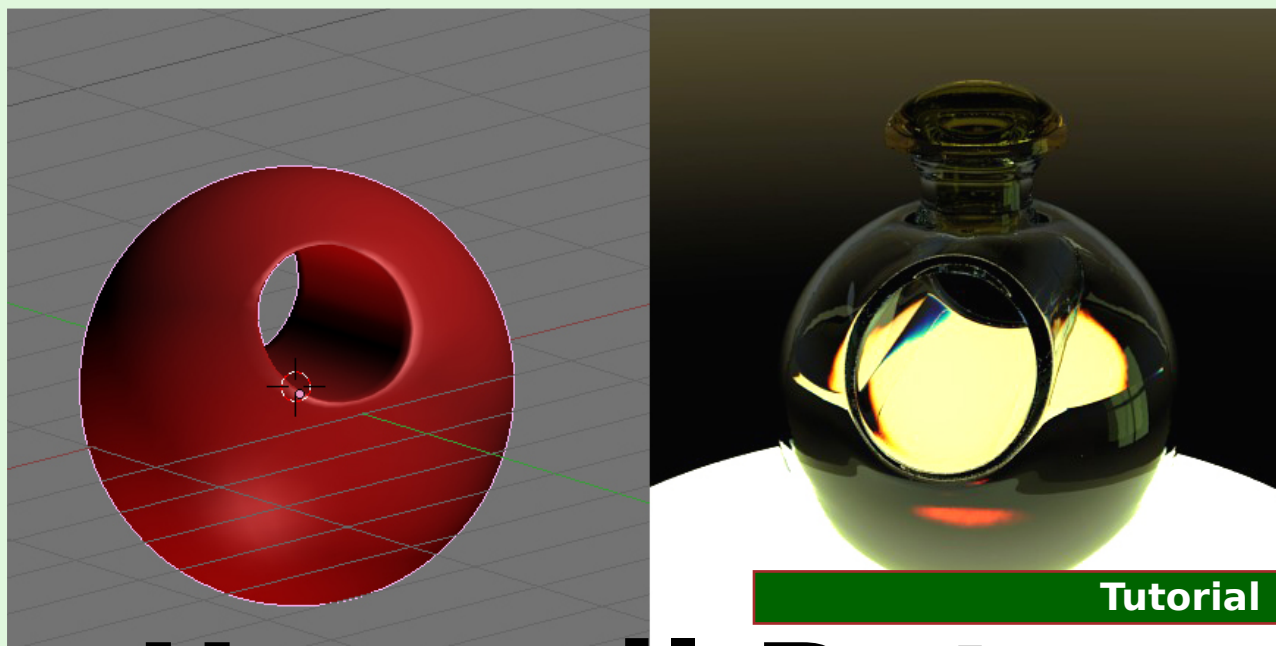
Il risultato finale va poi adattato alla pendenza del tetto, basta quindi un po' d'occhio e il gioco è fatto.





A partire dalla release 2.45 di Blender abbiamo a disposizione un potente tool che trova svariate applicazioni in fase di modellazione. Mi riferisco a "retopo" presente nel pannello di edit in "edit mode".

Esso ci permette ad esempio di ridisegnare la topologia di mesh complesse con altre più semplici allo scopo di rendere animabili degli oggetti la cui animazione risulterebbe impossibile ma anche di praticare fori o aperture su mesh non piane (esempio un cilindro o una sfera).



# Usare il Retopo

Nel primo caso, prendiamo ad esempio la fig. 1; come si può notare la mesh presenta un numero di vertici esagerato e se provassimo ad applicare il modificatore "decimate" il risultato sarebbe deludente.

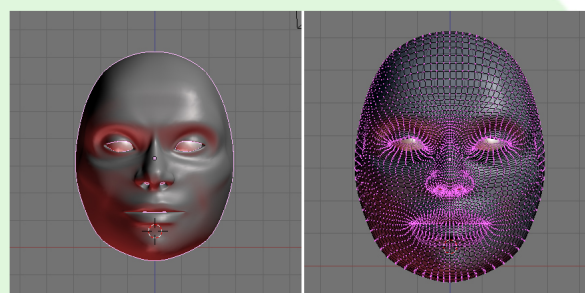


Fig.1

Aggiungiamo in "front view" un nuovo piano, scaliamolo, ruotiamolo e posizioniamolo di fronte al nostro oggetto, in un punto in cui la mesh sia sufficientemente ampia e premiamo il tasto "retopo" (fig.2).

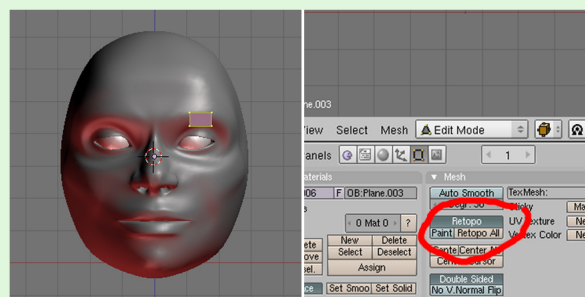


Fig.2

A questo punto il nuovo piano aderirà alla mesh sottostante (fig.3) e potremo cominciare ad estrarre gli edges o i singoli vertici ricreando una topologia consono al modello che stiamo "ricalcando". I nuovi piani edges o vertici aderiranno automaticamente all'oggetto e potremo aggiungere edge loops a piacimento e così via. Un modo pratico e veloce per aggiungere vertici e quello di tenere premuto CTRL e premere LMB.

Attivando "paint"(fig.4) abbiamo a disposizione un'altra potente funzione che risulta molto utile specialmente se si usa comunemente una tavoletta grafica ovvero la possibilità di disegnare direttamente sul modello scegliendo tra "pen" "line" o "ellipse".

Si disegna una griglia nel punto desiderato, rispettando il numero di vertici della mesh fin qui disegnata e si preme "invio" (fig.5-6).A lavoro finito sarà sufficiente disattivare retopo ed avremo una copia del modello originale con un numero limitato di vertici, pronta per un eventuale animazione o per il game engine.

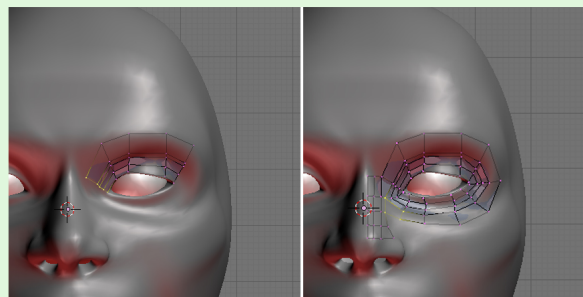


Fig.5

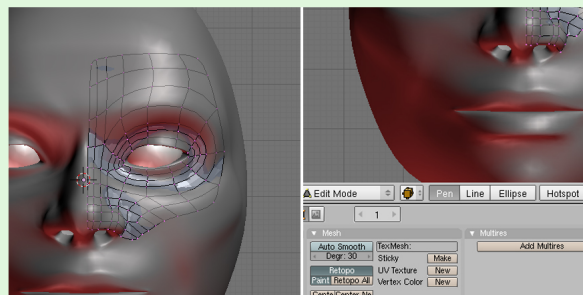


Fig.4

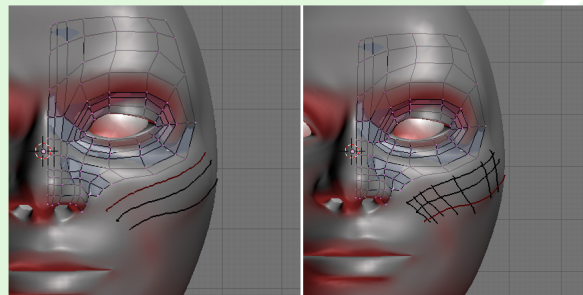


Fig.5

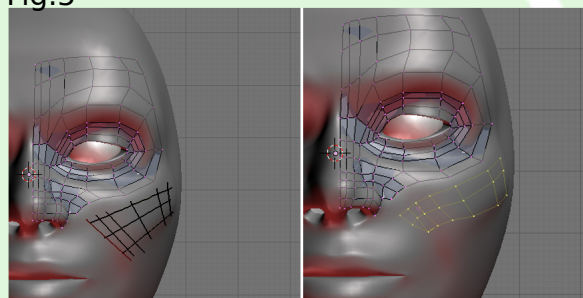
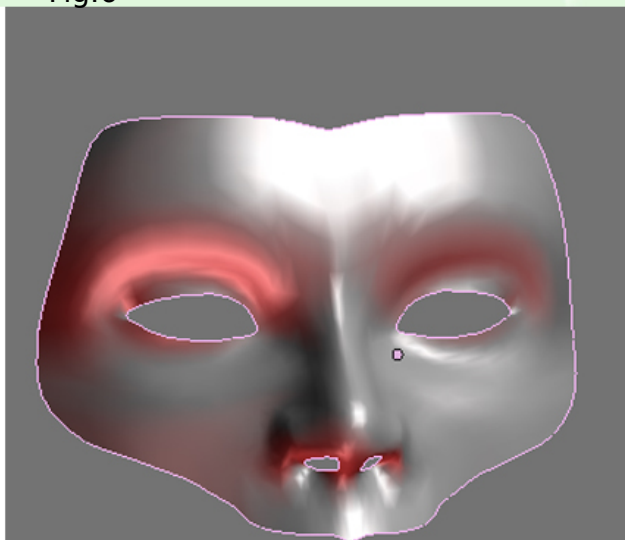
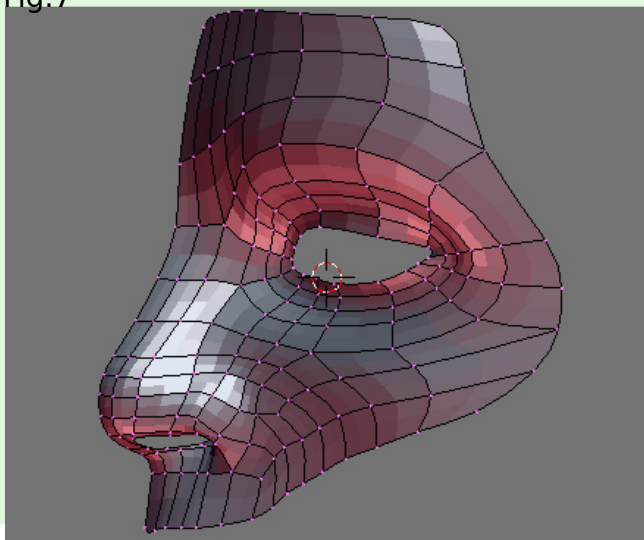


Fig.6

Fig.7





Passiamo ora alla tecnica che ci permette di praticare fori regolari su mesh curve o particolarmente complesse.

Aggiungiamo una sfera alla nostra scena, andiamo in object mode duplichiamola e in side view cancelliamone metà (fig.8). Selezioniamo tutti i vertici della semisfera e scaliamoli lungo l'asse Y con S key-Y key-0 key.

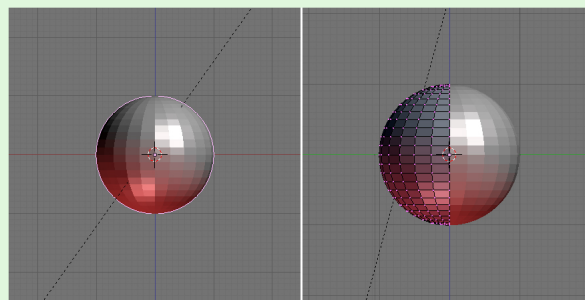


Fig.8

Spostiamo lungo l'asse Y il piano ottenuto in modo che sia staccato dalla prima sfera e selezioniamo alcuni vertici in corrispondenza del punto in cui vogliamo praticare il foro (fig.9).

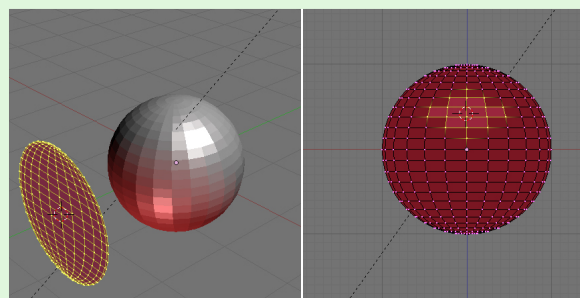


Fig.9

Una volta cancellati i vertici selezioniamo gli edges interni premiamo un paio di volte il tasto smooth, estrudiamo e scaliamo, un paio di colpi di smooth ancora estrudiamo e scaliamo un'ultima volta. Spostiamo il cursore al centro della selezione premendo SHIFT-S key e scegliendo "cursor to selection" e premiamo "to sphere 100%" nel pannello di edit.

A questo punto abbiamo il cerchio forato pronto per il "retopo" ma prima dobbiamo aggiungere un piano al centro della prima sfera di appoggio affinché i vertici della semisfera non vadano oltre la metà di essa(fig.10).

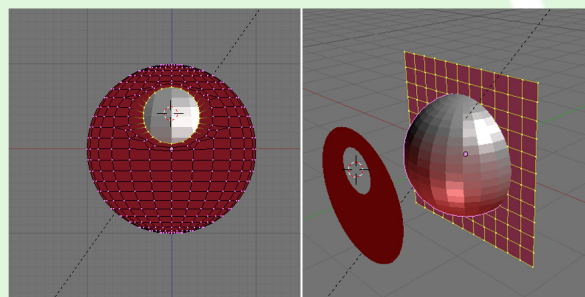


Fig.10

Selezioniamo ora il piano forato, in vista frontale e ortogonale selezioniamo tutti i vertici e premiamo "retopo" nel pannello edit.

La semisfera così ottenuta avrà bisogno di qualche ritocco (fig.11) e sarà sufficiente, dopo aver deselezionato "retopo", scalare a zero lungo l'asse Y i vertici evidenziati in fig.12.

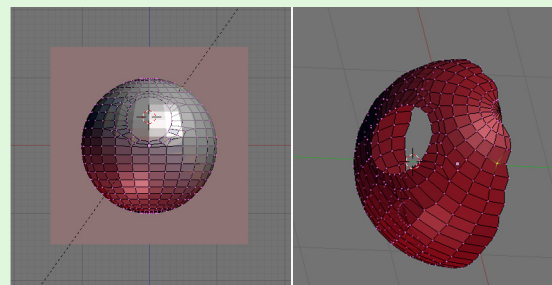


Fig.11

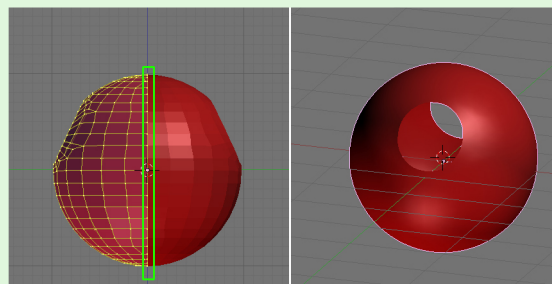


Fig.12

Una volta selezionato ed applicato il modificatore "mirror" avremo una sfera intera; allineando il cursore al centro e premendo to sphere 100% il lavoro sarà completo(fig.13).@

Irezumi

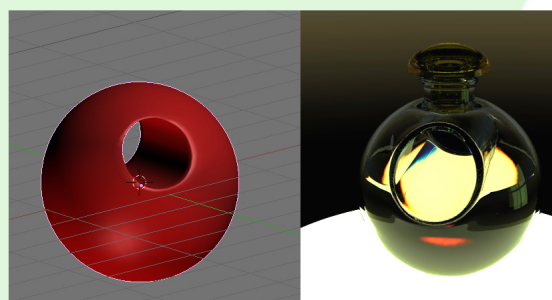


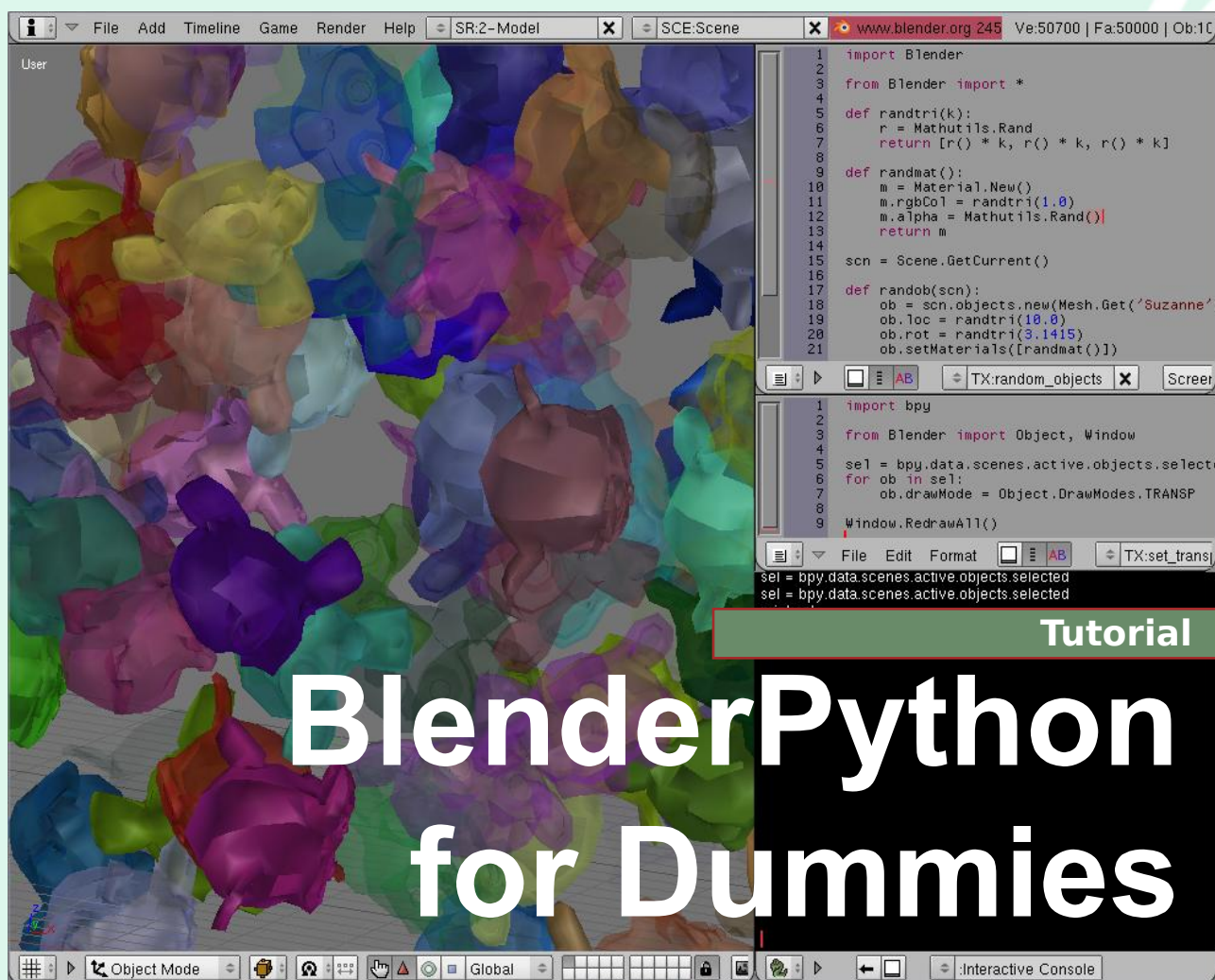
Fig.13



*Per molti utilizzatori di Blender, le possibilità offerte dal Python scripting restano avvolte da un alone di mistero e magia. Spesso, infatti, i timidi tentativi di capirci qualcosa sono frustrati o da documentazione specialistica, troppo criptica, o da tentativi di divulgazione che, pur in buona fede, prendono la questione troppo alla lontana cercando di insegnare come programmare in generale.*

*In questa rubrica vorrei tentare un approccio differente. Di volta in volta farò vedere dei brevissimi script, spiegandoli dal punto di vista dell'utente di Blender, lasciando in secondo piano gli aspetti relativi alla programmazione, relegandoli ad un apprendimento in gran parte mnemonico. Per chi ha intenzione di approfondire seriamente la programmazione in Python, si tratta sicuramente di un approccio insoddisfacente, ma fortunatamente in quel caso esistono molte altre risorse da cui partire. Spero invece che queste note siano utili per chi non è un programmatore né ha particolare interesse a diventarlo, ma vuole godere comunque di alcuni dei benefici offerti dallo scripting.*

Nota: prima di procedere alla lezione si consiglia di installare una versione completa di Python da [www.python.org](http://www.python.org).



**Lezione 1 - Modificare una proprietà per più oggetti**  
*di UncleZeiv*

Cominciamo, dunque! Nella presente puntata vedremo come sia possibile, con Python, supplire ad uno dei limiti dell'interfaccia di Blender, ovvero la difficoltà (talvolta impossibilità) di modificare una proprietà per più oggetti. Certo, esiste Copy Attributes (Ctrl+C), ma non sempre va bene. Vediamo, tanto per fare un esempio, come sia possibile abilitare la visualizzazione trasparente per ogni oggetto selezionato. Per farlo a mano, dovremmo cliccare "Transp" in F7 > Draw > Draw Extra per ognuno di essi: scoccante, quando il numero degli oggetti in questione è alto.

Proviamo a farlo in Python. Avviamo Blender e creiamo una decina di oggetti, posizionandoli casualmente nella scena e assegnando loro un materiale semitrasparente. Creiamo anche una finestra di tipo Scripts; qui, dallo header, clicchiamo sul menù Scripts, alla voce System, e invochiamo lo script Interactive Console. Si tratta di un ottimo ambiente dove cominciare a fare i propri esperimenti con Python. Adesso scriviamo le seguenti righe, copiando attentamente le scritte e la punteggiatura, rispettando maiuscole e minuscole e senza aggiungere spazi dove non ci sono, e ovviamente andando a capo alla fine della riga:

```
sel = bpy.data.scenes.active.objects.selected
```

Possiamo leggere questa riga così: d'ora in poi, con "sel" mi riferirò (leggendo da destra verso sinistra) ai selezionati, fra gli oggetti, della scena attiva, fra le scene presenti fra i dati di Blender (per ora leggete "bpy" come se rappresentasse Blender). Notate che se dopo aver premuto invio compare del testo rosso, vuol dire che la vostra operazione ha

Sarete probabilmente confusi dal fatto che alcune cose sono scritte in minuscolo ed altre in maiuscolo; che a volte servano le parentesi tonde, ed altre volte no, e via dicendo. Alcuni di questi aspetti hanno dietro delle motivazioni sensate, dettate da regole del linguaggio o da convenzioni ben precise; altre sono effettivamente scelte arbitrarie e talvolta incoerenti all'interno delle funzioni Python di Blender: si tratta del normale risultato di un'evoluzione durata anni durante i quali la coerenza dell'interfaccia non è stata messa al primo posto. Per adesso limitatevi a farci caso.

Interagendo con Blender tramite scripting, non è impossibile rovinare il file su cui si sta lavorando, ad esempio cancellando oggetti o cose simili. Per questo consiglio di fare i primi esperimenti in un file di test appositamente creato, e di salvare sempre i propri file prima di eseguire un proprio script (come anche quelli degli altri!).

prodotto un errore: nella maggior parte dei casi la causa è un errore di digitazione. Attenzione inoltre a distinguere fra oggetti attivi e oggetti selezionati: qui stiamo parlando solo di oggetti selezionati. È possibile però cambiare selezione e premere nuovamente invio per ottenere l'effetto desiderato sui nuovi oggetti selezionati. Continuiamo:

```
for ob in sel:  
    ob.drawMode =  
Object.DrawModes TRANSP
```

Se avete digitato tutto correttamente, il cambiamento sarà già andato in porto, ma probabilmente non è cambiato nulla nella 3D View. Questo perché nessuno ha ancora dato ordine a Blender di "ridisegnare" ciò che si vede sullo schermo, a differenza di quanto accade cliccando direttamente su un tasto. Per vedere le modifiche, basta cliccare nella 3D View; è possibile però ottenere questo stesso effetto direttamente da Python, scrivendo:

```
Window.RedrawAll()
```

Ma il lavoro vero lo fanno le due righe di prima. Semplificando, la prima riga chiede che la modifica successiva avvenga "per tutti gli oggetti selezionati all'interno della scena"; ovvero, per ogni oggetto "ob" (for ob) fra (in) i selezionati (sel). La seconda riga, intuitivamente, setta il "draw mode" di ogni oggetto a "TRANSP".

A questo punto, se siete riusciti a far funzionare tutto, potreste essere contenti ma anche perplessi; magari pensate che "da solo non ci sarei mai arrivato". Tranquilli: è proprio così! Una persona che ha dimestichezza con la programmazione ha sicuramente una sua idea sulla struttura da dare ad un programma, almeno a grandi linee, ma nessun ragionamento può portare a sapere, dalla pura logica, che per cambiare la draw mode di un oggetto si



debba scrivere proprio quella riga lì. Per ora, basta capire che queste quattro righe che abbiamo scritto forniscono una struttura generale per cambiare una stessa proprietà a più oggetti, e che di volta in volta basta cambiare la terza riga se ci interessa una proprietà diversa o un valore diverso. Ad esempio se volete aggiungere la visualizzazione wireframe, scrivete:

```
ob.drawMode = Object.DrawModes.WIRE
```

e se volete visualizzare anche il nome:

```
ob.drawMode = Object.DrawModes.NAME
```

La prossima riga, invece, consente di allineare la coordinata "x" di tutti gli oggetti selezionati ad un valore prestabilito:

```
ob.LocX = 27.3
```

Per questa volta può bastare. Consultate la tabella a fianco per vedere altre possibili variazioni. Un saluto e alle prossime puntate, dove vedremo alcuni consigli per consultare la documentazione di Blender Python e inizieremo a complicare la struttura di questo piccolo script. Data la natura sperimentale di questo approccio, ogni commento e suggerimento è particolarmente gradito.



*Per attivare/disattivare la visualizzazione degli assi locali:*

```
ob.axis = True  
ob.axis = False
```

*Per cambiare la modalità di visualizzazione degli oggetti selezionati:*

```
ob.drawType = \  
    Object.DrawTypes.SOLID  
ob.drawType = \  
    Object.DrawTypes.WIRE  
ob.drawType = \  
    Object.DrawTypes.SHADED  
ob.drawType = \  
    Object.DrawTypes.BOUNDBOX
```

*Per impostare o rimuovere un fake user (serve a conservare l'oggetto anche quando non è linkato a nessuna scena):*

```
ob.fakeUser = True  
ob.fakeUser = False
```

*Per impostare la visibilità dell'oggetto, ad esempio sul layer 2 o sui layer 3, 4 e 8:*

```
ob.layers = [2]  
ob.layers = [3,4,8]
```

*Per impostare la rotazione, in radianti, attorno ai tre assi:*

```
ob.rot = [1.2, 0.0, 1.15]
```

*Per impostare la scalatura dell'oggetto lungo i tre assi*

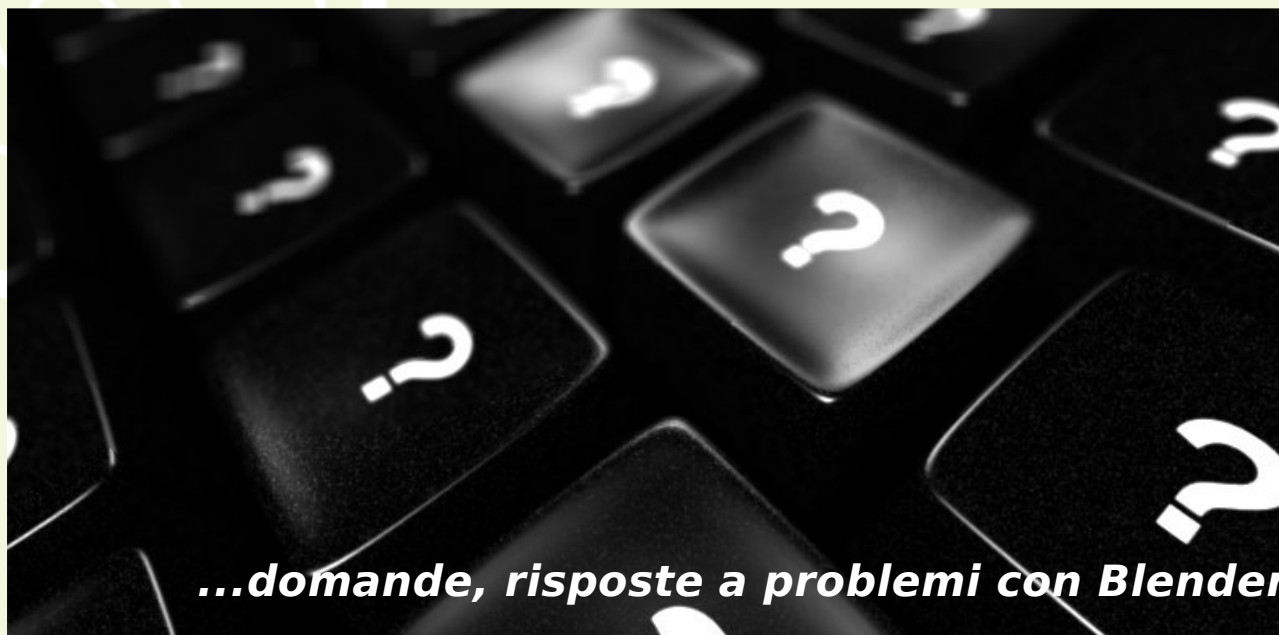
```
ob.size = [1.0, 1.5, 1.2]
```

Ovviamente, la Interactive Console è molto utile per sperimentare, ma molto scomoda una volta che avete messo a punto una sequenza di istruzioni che vi soddisfano e la volete utilizzare in seguito a più riprese. In questo caso, si può salvare lo script come file di testo. Da Blender, aprite la finestra del text editor, e scrivete le righe di cui sopra, precedute però da due righe per il momento "misteriose" (le spiegheremo in seguito):

```
import bpy  
from Blender import Object, Window  
sel = bpy.data.scenes.active.objects.selected  
for ob in sel:  
    ob.drawMode = Object.DrawModes.TRANSP  
Window.RedrawAll()
```

Inoltre, dovrete fare molta attenzione a indentare la riga (o le righe) che seguono la riga "for", con una tabulazione (preferibile) o degli spazi. In generale, non mettete spazi dove non sono richiesti. Infine, se volete eseguire lo script, digitate Alt+P nella finestra del text editor. Attenzione perché Ctrl+Z, oltre ad annullare la vostra modifica in 3dView, spesso annulla anche le ultime modifiche nel text editor.





*...domande, risposte a problemi con Blender*

# Dove si trova il Tasto SubSurf?

Parte questa rubrica di FAQ prese un po' in giro a caso sul forum di Blender.it . Anche se non siete nuovi al programma, una volta lo siete stati, a meno che non siate uno di quelli che l'ha scritto, in questo caso siete esentati dal leggere questa sezione :D. Quindi fate un tuffo nel passato rivivendo le care e vecchie domande che ci hanno fatto imparare molto e sbattere la testa sulla tastiera (alzi la mano chi non lo ha mai fatto). Se invece siete alle prime armi, forse qualche risposta potrà farvi andare a letto (visto che sono le 2 e domani dovete alzarvi alle 7.00) oppure, se la domanda non c'è e volete farla, scrivete scrivete scrivete, magari qualcuno vi risponde.

## Oggetti che spariscono ►

D: Il gatto è saltato sulla mia tastiera, non so che tasto abbia premuto, ma mi ha cancellato tutto e ora vedo solo la griglia vuota, devo cambiare gatto?

R: Tieni pure il tuo gatto, perché il lavoro non è andato perso, più semplicemente è stato cambiato il Layer di lavoro. Infatti, se premi i tasti da 1 a 0 sulla tastiera in alto, ti puoi spostare in un altro Layer. Di default si è sempre nel Layer 1, quindi ti basta ripremere 1 per tornare al tuo lavoro. In oltre puoi attivare contemporaneamente più Layer, basta tenere premuto il tasto SHIFT.

In alternativa c'è una pulsantiera sotto ogni finestra 3d, che contiene 20 layer diversi nei quali suddividere lavori di grandi proporzioni (un layer per le luci, uno per le camere, uno per le armature ecc...)



Fig1: Questo è il pannello dei layer, premendo il tasto SHIFT mentre si seleziona/deseleziona un pulsante, il valore degli altri rimane invariato.

D: Ho passato la giornata intera a unvrappare il mio mostro-scimmia-alieno-supercorazzato, poi gli ho applicato una Texture stramegafiga, solo che mentre nella finestra 3d riesco a vedere la texture, quando faccio il Rendering la Texture scompare, è un errore del programma?

R: Quando mettiamo una UVmap nell'UV/Image Editor e vogliamo che questa appaia sull'oggetto, dobbiamo creare un materiale per l'oggetto e procedere in due modi diversi:

- quello facile (ma superficiale), basta cliccare sul pulsante TexFace nelle opzioni del materiale;
- quello di difficile (più "Professional"), creare una nuova Texture, selezionarla come Image e caricare l'immagine che si vuole applicare. Usare la Texture come Map Input settato su UV, mentre Map To settato su Col.

Consiglio comunque una lettura del manuale nella sezione dei materiali, che da soli richiedono molto studio e approfondimento, in quanto da essi dipende il 50% della resa di un rendering (percentuale variabile a seconda del tipo di progetto a cui si sta lavorando).

D: Salve, sono Steven Spielberg ed ho appena finito il mio nuovo film di animazione 3d con Blender, ora, come faccio a salvare un Animazione?

R: Salve signor Spielberg, è un vero onore conoscerla. Per salvare un animazione fatta in Blender esiste un tasto nel pannello Scene (F10), denominato "Anim". Tutto in torno ci sono diverse opzioni che comprendono quali e quanti frame Renderizzare, quanti frame al secondo renderizzare, le dimensioni di ogni Frame espresse in Pixel e il formato in cui si vuol salvare il rendering. Blender può salvare direttamente in formato video (avi, mpeg, mov), ma è consigliabile salvare in formato immagine (jpg o TGA) e unire il tutto con un programma esterno (tipo Virtualdub). Questo perché è più semplice ripartire in caso di errori del programma, blackout ecc..., in quanto basta riprendere il rendering dall'ultimo fotogramma salvato.

Il salvataggio avviene di default in una cartella c:\tmp (per utenti windows), modificabile dalla scheda Output del pannello Scene (F10).

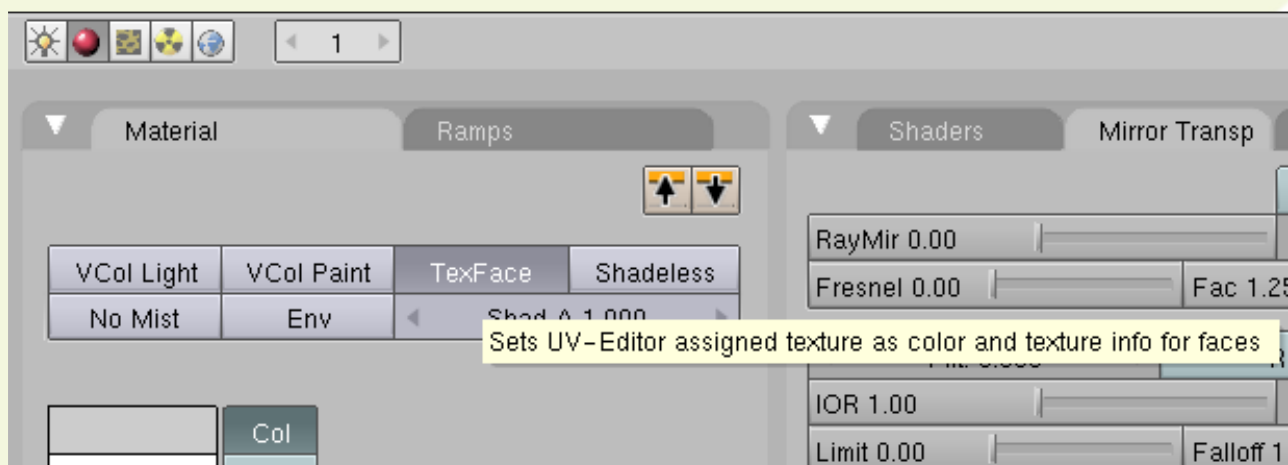


Fig2: L'immagine mostra la posizione esatta del pulsante TexFace di Blender 2.45.

### ... e l'Audio? ►

D: Chiedo scusa, ma sono sempre Spielberg, sono riuscito a renderizzare l'animazione, ma Blender non mi ha aggiunto l'audio, eppure io lo avevo caricato dalla Audio Window, dove sto sbagliando.

R: Purtroppo Blender non supporta a pieno il comparto audio, la finestra che lei ha usato serve principalmente per aiutare la sincronizzazione tra audio e animazione, soprattutto per i dialoghi. Si consiglia di unire audio e video con un programma esterno (tipo Virtualdub), in attesa che gli sviluppatori si occupino della faccenda.

### Vista Camera Impazzita ►

D: Mentre stavo lavorando, ho provato ad allineare la Camera con la vista 3d premendo la combinazione CTRL+ALT+Tn0, solo che ora mi ritrovo la vista dell'interno della mesh a cui stavo lavorando, nonostante la Camera sia da tutt'altra parte cosa ho sbagliato?

R: Blender permette di impostare come Camera, qualsiasi oggetto, basta selezionare tale oggetto e premere CTRL+Tn0, in pratica hai premuto erroneamente questa combinazione invece dell'altra. Per riselectare la Camera come principale, basta selezionarla e premere CTRL+Tn0 e tornare tranquillamente al proprio lavoro.

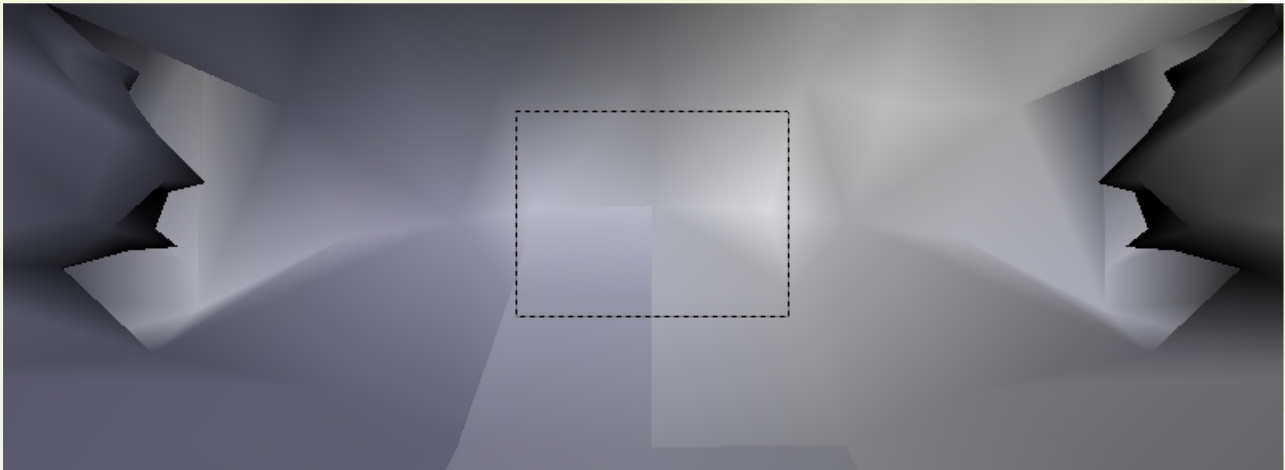


Fig3: Questa è la visuale di un oggetto adibito a Camera, non ho ancora ben chiara l'utilità di questa funzione, di certo non va usata con oggetti chiusi.

### Artefatti nel Rendering ►

D: Ho seguito passo passo le indicazioni di un tutorial sulla modellazione, tutto perfetto, solo che quando faccio il rendering su alcune facce appaiono delle sfaccettature, ma io non ho applicato nessun materiale, dov'è il problema

R: Capita a volte di avere delle facce doppie che durante la modellazione non si vedono, ma risultano solo quando si fa il rendering. Per eliminare l'inconveniente bisogna trovare le facce sovrapposte ed eliminarle.

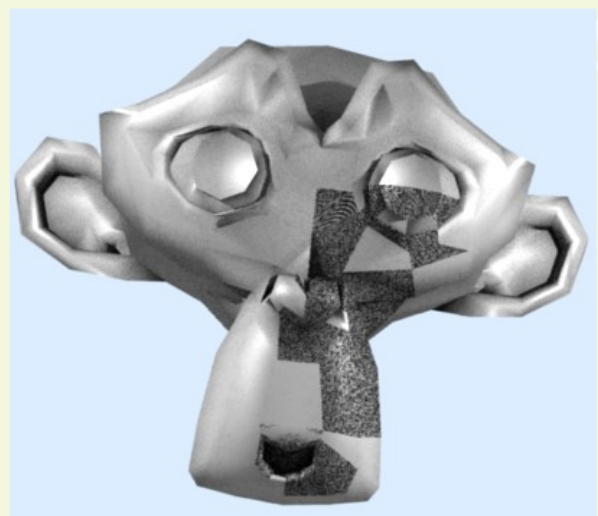


Fig4: Ecco cosa succede quando per sbaglio di duplicano delle facce.

Con questo termina questa rubrica, alla prossima... ah, ma che sbadato, ho dimenticato di dirvi sove hanno messo il tasto SubSurf, be' ecco vedete, basta andare.....





# Comunicazioni dallo Staff

## Contest Copertina ▶

Nuovo numero, nuova copertina, quindi nuovo contest.

Vogliamo lanciare un nuovo contest per la copertina del N2 di Blender Magazine Italia.

Il tema sarà: "Figura Femminile".

Come potete notare è un argomento molto ampio e abbastanza di libera interpretazione, quindi tutto ciò che avrà attinenza con esso sarà ammesso in gara.

Modalità di partecipazione e regolamento:

- Le immagini devono essere eseguite principalmente con Blender.
- Si possono eseguire anche più lavori, ma ricordate che conta la qualità non la quantità.
- Sarà possibile usare anche altri software open source, ma solo per post produzione texture e tutto quello che c'è di contorno nella realizzazione di un'immagine 3d.
- Le immagini devono essere in formato A4, ad una risoluzione moderata (1024 o giù di lì per il lato più corto).
- Le immagini possono essere postate come Wip su Blender.it o Kino3d.
- Sono ammessi anche lavori precedenti basta che rispettino le regole sopraelencate.
- Sarà aperto un topic su Blender.it al quale ci si potrà iscrivere ufficialmente, postare i lavori finiti e procedere in fine alla votazione dell'opera migliore.
- Il vincitore avrà l'onore di vedere pubblicata la sua immagine sulla copertina di Blender Magazine Italia.
- Il vincitore potrà scrivere un piccolo Making Of del suo lavoro che sarà pubblicato sulla stessa rivista elettronica. L'articolo non è obbligatorio.
- Ricordate di specificare il nome che volete usare per la firma (nick e/o

nome vero).

Aspettiamo numerosi lavori e quelli più belli saranno messi nella gallery del prossimo numero.

## We Want You ▶

Blender Magazine Italia cerca collaboratori per portare avanti questo progetto e far crescere la comunità italiana di Blender.

Questa rivista si basa non solo sul lavoro di due persone (Anfeo e Sinistar) che hanno provveduto all'impaginazione e la divulgazione di questo documento, ma anche da tutti gli articolisti che hanno scritto le varie rubriche e tutorial.

Quindi si può dire che è una rivista nata dalla comunità italiana di Blender raggruppata principalmente su Blender.it e Kino3d.

Ecco in sintesi di chi abbiamo bisogno:

**Articolisti**, persone che vogliono pubblicare un loro tutorial, una recensione o un articolo su una feature di Blender e dintorni.

**Grafici**, una o più persone che si occupino della grafica della rivista, sia interna che esterna.

**Collaboratori**, persone che vogliano aiutare la rivista dall'interno, aiutandoci a cercare articoli, news e immagini.

**Artisti**, persone che vogliono veder pubblicate le loro immagini sulla gallery che c'è al fondo di ogni numero.

Per chiunque fosse interessato ad uno di questi compiti, può contattarci su Blender.it via PM (ad Anfeo o Sinistar) oppure via E-mail a [anfeo@libero.it](mailto:anfeo@libero.it).

# Gallery



Andrea Fiore (waaf) - Ferrari - Modellata in Blender e render in Yafray



Andrea Fiore (waaf) - Hero - Modellata in Blender e render interno con AO

# Gallery



Anto - Merry Nightmare - 2° posto al contest di Natale 2007 di Blender.it



Dreamer - Moto futuristica



# Gallery



IZE - Tarda



Waz - Bathroom

# Gallery



tonipezz - Alitalia Boeing