

Numero 6 - Settembre 2009

# BLENDER

## Magazine Italia

GAME ENGINE

Come realizzare un platform interattivo

CAMERA TRACKING

Unire riprese dal vivo e animazioni 3d

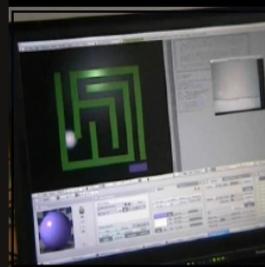
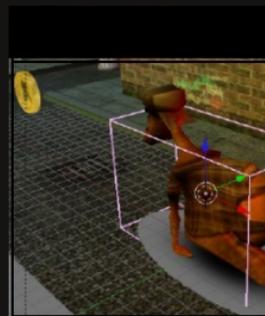
MAKING OF...

Scena architettonica di una galleria



# Indice

News .....	4
GE: Creare un platform .....	7
Object e laser tracking .....	12
BT: intervista all'admin .....	17
Making of "Galleria" .....	19
Colonna decorata .....	24
Simulare uccelli in volo .....	28
Camera Tracking .....	33
V-Ray Displacement .....	39
Contest copertina BMI7 .....	42
Gallery .....	43



**A**rticoli molto interessanti in questo nuovo numero di Blender Magazine Italia.

Dopo le consuete news, il primo articolo è un tutorial che spiega come realizzare un platform interattivo utilizzando il BGE. Viene descritto come far muovere Ernie, una lumaca antropomorfa, e come far in modo che raccolga dei bonus. Con accuratezza vengono mostrati i Logic Bricks che è necessario utilizzare. Un articolo tutto da leggere!

Per gli appassionati di programmazione, c'è l'interessante articolo sull'Object e il laser tracking in Blender.

Dario Caffoni racconta come è nata l'idea di creare quello che ora conosciamo come Blender-Tutorial .com. Inizialmente solo un blog ma ora una vivace community con diversi progetti attivi.

Ettore Mele ha scritto un tutorial in cui spiega i punti principali per creare una scena architettonica, in questo caso l'interno della Galleria Vittorio Emanuele II di Milano.

Rimaniamo in ambito architettonico. Il tutorial successivo illustra, attraverso 23 passi, come modellare una colonna decorata a spirale.

Come simulare con Blender uno stormo di uccelli in volo? Ce lo dice Anfeo in un tutorial di 16 passaggi in cui sfrutta le particelle e la fisica Boids. Il tutorial è utile, in generale, per comprendere una metodologia per muovere in gruppo grandi masse di oggetti.

Ancora Anfeo descrive cos'è il camera tracking e le regole da seguire nelle riprese per avere un tracking accurato. Infine specifica come ottenere il camera tracking utilizzando Voodoo Camera Tracking e come importarlo successivamente in Blender.

Boggio, che nel precedente numero di BMI, aveva parlato di come utilizzare V-Ray con Blender, ora spiega come sia possibile creare displacement map da utilizzare con V-Ray.

Infine vi proponiamo una ricca galleria di immagini tutte rigorosamente realizzate con Blender.

**Blender Magazine Italia**  
numero 6  
anno 2009

**Responsabili:**  
Luca Pinciani (Sinistar)  
Alfonso Annarumma (Anfeo)

**Collaboratori:**  
Gikkio

**Grafica:**  
Davide\_G

**Siti di riferimento:**  
[www.blender.it](http://www.blender.it)  
[www.kino3d](http://www.kino3d)  
[www.blender.org](http://www.blender.org)

**Software utilizzati:**  
Blender 2.45  
Pdftk  
The Gimp 2.4  
Openoffice  
Scribus 1.3.3



In Copertina:  
"Butterfly"  
di t0m0



© Tutte le immagini appartengono ai legittimi proprietari

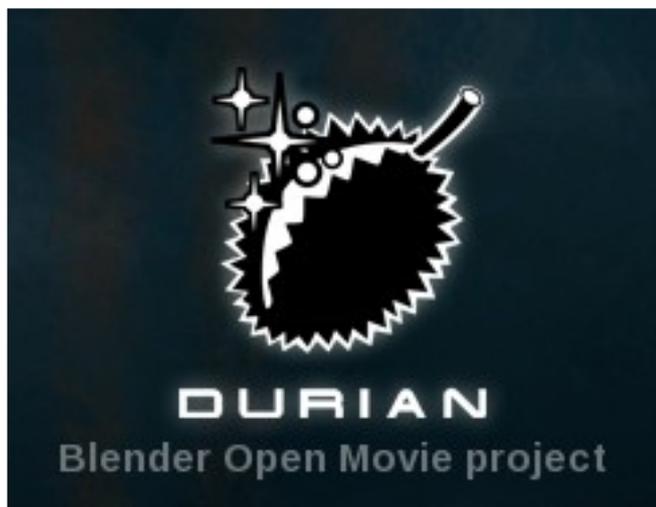
# Blender: le news

Ultime news dal mondo Blender

di Luca Pinciani

## Progetto "Durian"

La Blender Foundation ha annunciato lo sviluppo di un nuovo "open movie": il progetto ha come fine la realizzazione di un cortometraggio utilizzando Blender, come accaduto precedentemente con Elephants Dream e Big Buck Bunny.



Dopo un'attenta selezione sono già stati scelti gli artisti che svolgeranno il progetto. I membri del

gruppo sono: Colin Levy (regista), David Revoy (concept art), Ton Roosendaal (produttore), Angela Guenette (modellazione, rigging, animazione), Ben Dansie (3D in generale), Nathan Vegdahl (rigging, animazione), Lee Salvemini (animazione), Soenke Maeter (shading, illuminazione, compositing), Brecht van Lommel e Campbell Barton (supporto tecnico).

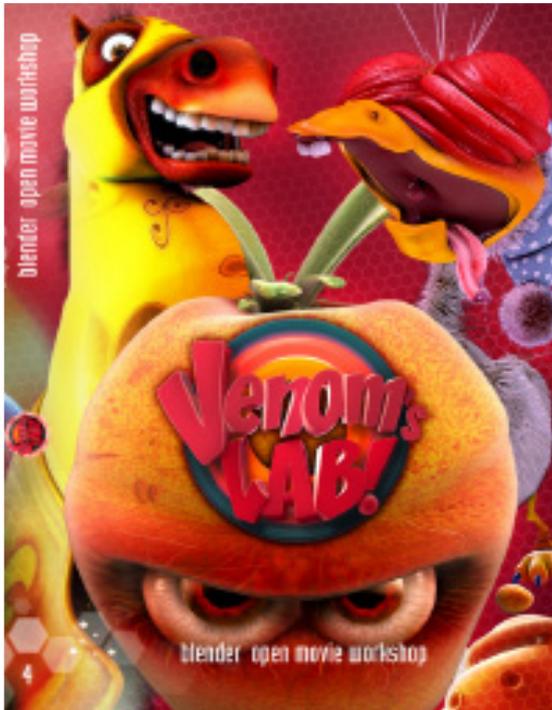
Dal sito ufficiale del progetto (<http://durian.blender.org/>) è possibile conoscere maggiori informazioni (come la timeline del progetto e alcuni concept art di ispirazione) e trovare i link alle singole pagine degli artisti del team.

## Venom Labs! pronto

Pablo Vazquez ha realizzato un dvd chiamato "Venom Labs!".

Pablo è un ragazzo sudamericano di 22 anni che da anni utilizza Blender (ha anche collaborato al

progetto YoFrankie!) ed è noto sui forum col nome di VenomGFX.



Il dvd da lui realizzato copre diversi aspetti della CG: modellazione, shading, illuminazione, sculpting, simulazione di tessuti, creazione di peli, rendering e compositing.

Il dvd è diviso in 10 capitoli: ognuno dedicato ad uno specifico argomento. Utilizzando uno dei personaggi creati da Pablo Vazquez, sarà possibile essere guidati passo passo attraverso i comandi e le impostazioni più importanti.

Per vedere un'anteprima, presentata dallo stesso Pablo, è possibile guardare il seguente video: <http://vimeo.com/5452577>

Per avere invece più informazioni, il sito ufficiale di VenomGFX è <http://www.venomgfx.com.ar/>

### Risultati del Blender F1 Challenge 2009

A fine giugno si è concluso il Blender F1 Challenge 2009, il concorso per il quale bisogna realizzare, con Blender, l'immagine di un mezzo di trasporto futuristico. Dalla prima edizione del contest, nel 2001, la qualità dei lavori proposti è migliorata ad ogni edizione. Anche quest'anno,

molte erano le possibili immagini vincitrici. Attraverso una votazione, la vittoria è andata a Zordan. Tutte le immagini presentate al contest possono essere viste dal sito ufficiale: <http://www.blenderf1.com/>



Come ogni anno, gli organizzatori hanno ringraziato gli utenti che hanno preso parte al concorso di quest'anno e hanno esortato gli utenti di Blender a partecipare all'edizione del 2010.



### Database per i materiali per LuxRender

LuxRender è un motore di render compatibile con Blender. Il sito ufficiale di questo engine ha realizzato un sezione in cui raccogliere materiali utilizzabili con il proprio motore di render.

Gli utenti possono scaricare materiali pronti da utilizzare o caricare i propri per condividerli con gli altri utenti.

Attualmente la collezione comprende oltre una trentina di materiali ma è destinata a crescere dato che il database è stato creato da poco.

La raccolta di materiali può essere raggiunta all'indirizzo:

<http://www.luxrender.net/lrmdb/en/material/>

Nel sito di LuxRender è anche possibile vedere una gallery con belle immagini che mostrano la potenzialità di questo motore:

<http://www.luxrender.net/gallery/main.php>



### Blender 2.49 e anticipazione su Blender 2.50

Poco tempo fa è stata rilasciata l'ultima versione Blender: la 2.49b. Come ad ogni nuova release, molte sono le novità. Quelle della 2.49 riguardano principalmente il game engine (aggiornamento del Bullet physics, dome rendering in tempo reale, possibilità di utilizzare video textures) ma non solo: texture nodes, forte miglioramento delle operazioni booleane, supporto per JPEG2000, possibilità di dipingere direttamente le texture sugli oggetti 3d, tecnica "sketching" per il rigging e molte altre. Per una lista più completa di tutte le novità della versione 2.49 è possibile consultare il release log: <http://www.blender.org/development/release-logs/blender-249/>

Per scaricarle, invece, è possibile trovare il link nell'apposita sezione del sito ufficiale di Blender.

Poco dopo l'uscita di Blender 2.49a, Jonathan Williamson ha realizzato un video di più di 20 minuti che mostra alcune novità della futura versione 2.50 di Blender: <http://www.cgcookie.com/articles/blender-25-overview/>. Successive notizie hanno comunicato che è prevista l'uscita di Blender 2.50 per ottobre 2009, circa nel periodo in cui si svolgerà l'ottava annuale Blender Conference ad Amsterdam.

Per i più curiosi, su internet si possono trovare altri video che mostrano alcune nuove features in sviluppo per la futura release di Blender. In particolare, all'indirizzo <http://www.youtube.com/watch?v=xCKv-WWO85A> è possibile vedere una stupefacente simulazione di fumo realizzata con una versione SVN di Blender. Un'altra attesa novità è l'introduzione delle BMesh: un video di preview è disponibile all'indirizzo <http://vimeo.com/6430167>.



# GE: Creare un platform

Tutorial su come realizzare un gioco interattivo utilizzando il GE

di Orathio89

Si ripensa sempre con nostalgia al primo incontro con Blender. Nel mio caso avvenne un'estate di otto anni fa, sul microscopico trafiletto di una rivista di quart'ordine, che lo descriveva come "software con strumenti professionali per la grafica 3d e la creazione di videogame".

La curiosità ebbe il sopravvento, ma la prima occhiata alla faticosa interfaccia è sempre scioccante, e subito desistetti. Fu solo in seguito che mi riavvicinai e cominciai, lentamente, a fare gavetta.

All'epoca i tutorial sul game engine non erano esattamente diffusissimi, e come molti fui costretto a sviscerare i fantomatici Regression files, ma era comunque un modo divertente e sempre stimolante di imparare.

E lo è ancora.

SI COMINCIA

In questo tutorial cercheremo di esplorare le basi del Blender game engine (BGE), scoprendo

come creare un semplicissimo "giochino" platform interattivo, con il classico personaggio seguito dalla telecamera che raccoglie bonus. In questo caso ci faremo aiutare da Ernie, l'irascibile lumaca.

Non sei contento, Ernie?



Sì, beh, h-ehm, la prima cosa da fare è procurarsi il file .blend su cui lavoreremo. Potete trovarlo in allegato a questo numero di Blender Magazine Italia. Nel file sono presenti due SCENE, selezionabili di solito dal piccolo menu in alto al centro della schermata. Nella scena FINALE trovate il lavoro finito, mentre TUTORIAL è la scena su cui agiremo.

## UN PO' DI TEORIA

Prima di entrare nel vivo è opportuno fare una panoramica sulle funzionalità dei LOGIC BRICKs, che di fatto sono la chiave di volta del BGE. Nella finestra BUTTONS, premete F4 o cliccate sull'icona con il faccino per visualizzare i pulsanti LOGIC. A sinistra, in violetto, ci sono i tasti relativi alla FISICA, mentre a destra troviamo tre colonne di pulsanti verdi, rispettivamente SENSORS, CONTROLLERS e ACTUATORS.

La dinamica fra queste tre colonne è molto simile a quelle che in programmazione sono le condizionali IF>THEN; Ogni qual volta si verifica una determinata condizione, espressa dal SENSOR, il CONTROLLER connette tale condizione ad un ACTUATOR, che attiva così una reazione. La connessione fra i tre elementi si stabilisce cliccando sui tondini gialli alle loro estremità e trascinandoli sui tondini dell'elemento al loro fianco: apparirà un segmento nero a testimoniare l'avvenuta unione.

Ecco un esempio:

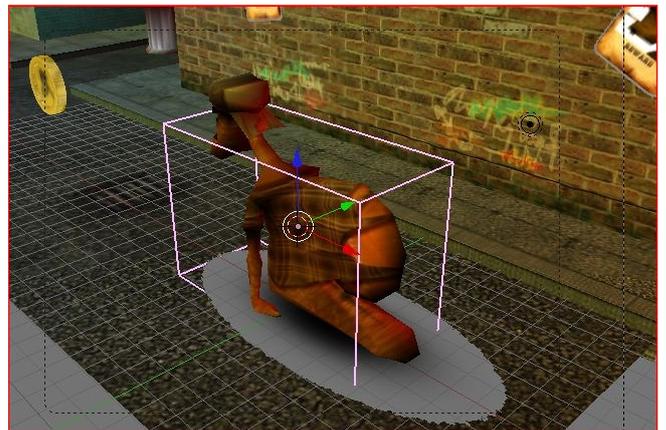


tipi diversi di sensor, controller e actuator, ognuno con una determinata funzione.

Guardiamoli dal vivo.

ERNIE CAMMINA (o quasi)

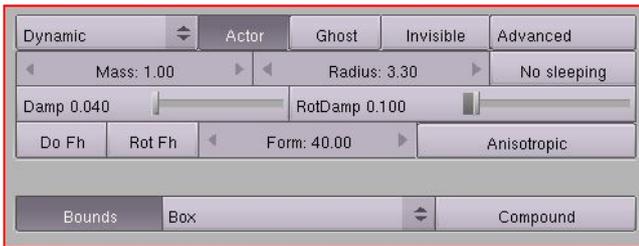
Appena aprirete il file noterete che Ernie è racchiuso in un cubo trasparente:



tranquilli, non è a causa dei suoi precedenti penali, semplicemente la mesh principale del personaggio è imparentata al cubo che la circonda, in questo modo il rilevamento delle collisioni sarà meno difficoltoso per il BGE: sarà infatti il cubo stesso a muoversi, e non la mesh vera e propria. Cominciamo a calibrarlo;

-Con il cubo selezionato (se lo deselectionate e non riuscite più a vederlo, tranquilli: basta passare alla visuale Wireframe) premete f4 per passare ai pulsanti LOGIC

-dal pulsante menu dei tasti violetti all'estrema sinistra, selezionate DINAMIC al posto di STATIC, regolate lo slider del valore RADIUS appena apparso a 3.30 (questo è il cerchio di collisione, visibile in wireframe), attivate, sotto, il pulsante BOUNDS e assicuratevi che a fianco sia selezionata la voce BOX.



ora si passa ai logic brick: cliccate tre volte su ADD nella colonna SENSORS per aggiungere tre sensor. Cliccando su ALWAYS potrete scegliere il tipo di sensor: convertiteli tutti e tre in KEYBOARD e poi in ognuno di essi cliccate sul campo vuoto accanto a KEY: così facendo potrete assegnare ad ogni sensor un comando della tastiera.

Assegnate FRECCIA SU al primo, FRECCIA SINISTRA al secondo e FRECCIA DESTRA al terzo: sono i pulsanti che useremo per muovere Ernie.

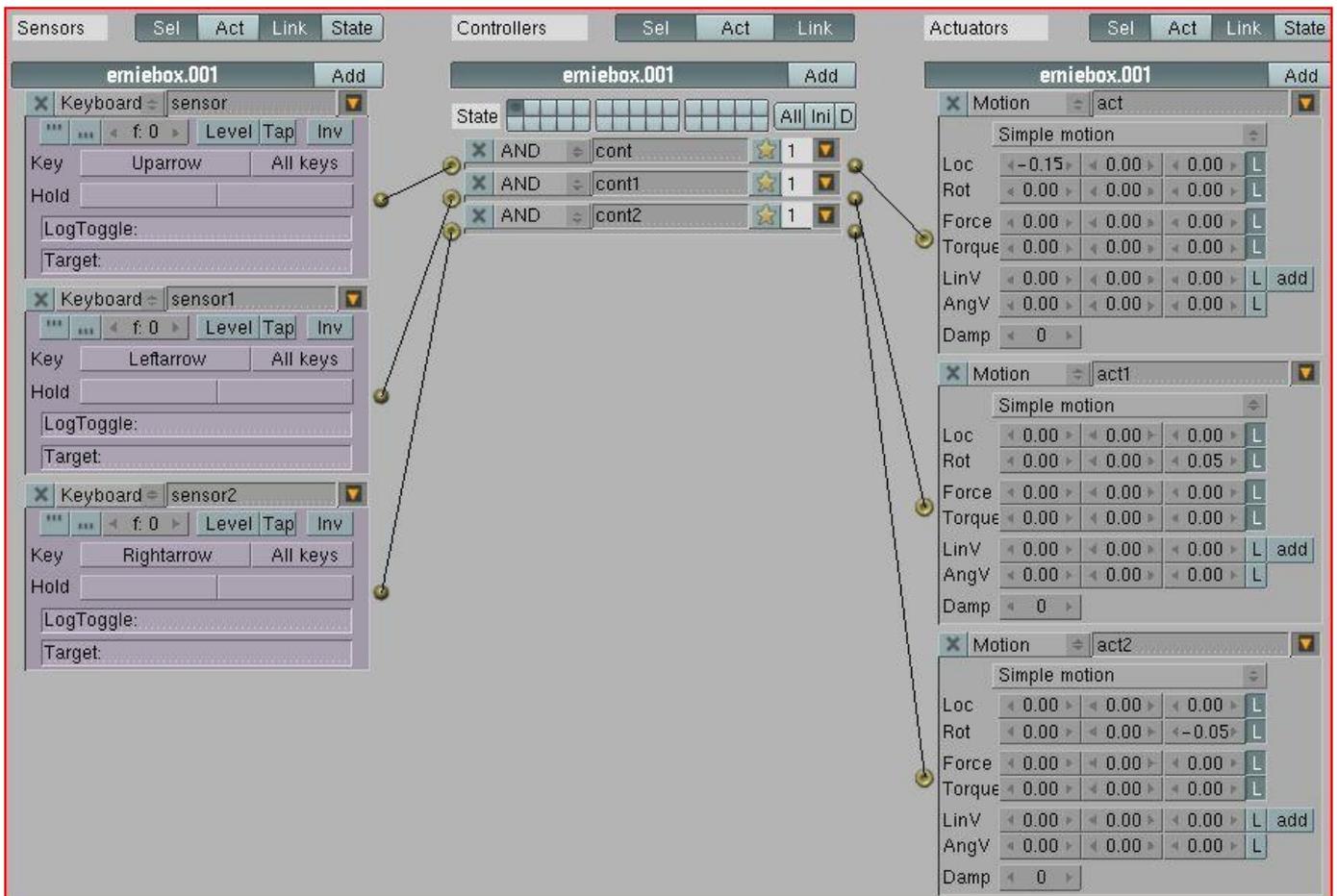
- ora aggiungiamo, allo stesso modo, tre CONTROLLER e tre ACTUATOR; lasciamo i controller così come sono, mentre convertiamo i tre actuator in MOTION. ogni actuator controllerà un movimento fisico di Ernie, e andrà poi associato al relativo sensor. Ogni actuator motion ha tre slider per ogni movimento (LOC, ROT, FORCE etc), ognuno relativo ad un asse spaziale (x,y,z).

Nel primo actuator regoliamo lo slider X di LOC su -15.0,

nel secondo lo slider Z di ROT su 0.05

e nel terzo lo slider Z di ROT su -0.05.

-una volta finito è il momento di collegare il tutto: connettete, come spiegato sopra, ogni sensor al suo actuator tramite i controller, seguendo l'ordine dall'alto verso il basso. dovreste avere qualcosa di simile:



Per l'occasione ho preparato una semplicissima animazione IPO da assegnare ad Ernie mentre cammina:

-selezionate la mesh di Ernie;

-aggiungete un sensor, un controller ed un actuator;

-convertite il sensor in KEYBOARD ed assegnategli FRECCIA SU;

-convertite l'actuator in IPO: gli actuator IPO hanno due slider, relativi ai fotogrammi di inizio e fine dell'animazione: portate END a 19 e cambiate PLAY in LOOP END.

-collegate il tutto come fatto precedentemente con i brick del cubo.

Ora premete P per far partire il gioco e preparatevi alla prova del fuoco: se tutto è andato bene riuscirete a controllare Ernie con le frecce direzionali!



### NEL MIRINO

Il BGE possiede un actuator apposito per regolare il comportamento della telecamera:

-premete 0 (zero) nella finestra 3d per andare in visuale camera, e fatto ciò selezionate la telecamera stessa;

-aggiungete anche qui un sensor, un controller ed un actuator. Connetteteli fin da subito fra di loro. convertite l'actuator nel tipo CAMERA. Nel campo OB: inserite il nome dell'oggetto che la telecamera dovrà seguire, in questo caso Ernie (ATTENTI a trascrivere bene i nomi degli oggetti, maiuscole comprese: su questo il BGE è imperativo).

-I restanti slider regolano il comportamento della telecamera: HEIGHT l'altezza a cui deve stare, MIN e MAX i limiti di distanza dall'oggetto seguito. Regoliamo HEIGHT a 6.45, MIN a 16.45 e MAX a 20.00



Ora proviamo di nuovo a far partire il gioco: la telecamera seguirà Ernie in maniera fluida e piacevole.

### PER UN PUGNO DI DOLLARI

Adesso è il momento di permettere ad Ernie di guadagnarsi il suo bonus quotidiano. Prima di tutto, dalla barra strumenti della finestra utente (quella in alto) andiamo su GAME e spuntiamo la voce SHOW DEBUG PROPERTIES. Questo ci permetterà di vedere il valore numerico del bonus.

Adesso selezioniamo la moneta e prepariamoci a calibrarla:

-nella finestra LOGIC(f4) è presente un grosso pulsante, ADD PROPERTY: premetelo per aggiungere una property. Le property possono essere utilizzate per fare quasi di tutto nel BGE, sono ottime come valori globali o come timer, e l'applicazione di una property ad un'oggetto lo distingue da un'altro in molti modi. Chiamiamo la property appena creata MONETA nel campo NAME:

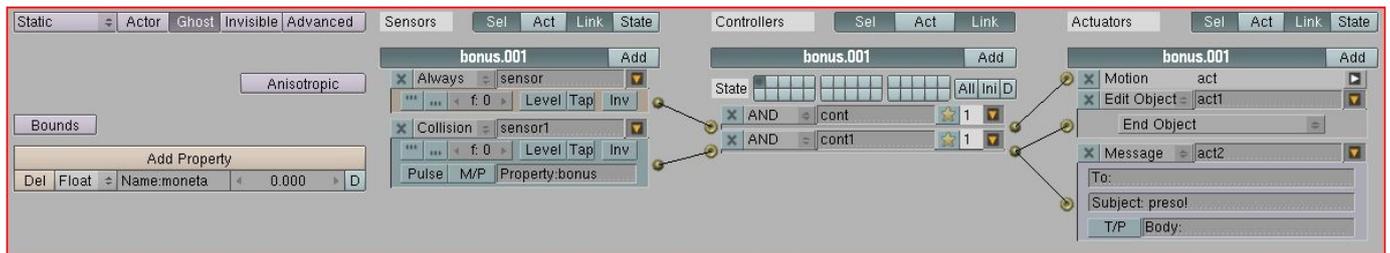
-già che ci siete, create anche qui un sensor, un controller ed un actuator e collegateli. Convertite l'actuator in MOTION e regolate lo slider Z di ROT a 0.10 ;in questo modo sarà più bella da vedere.

-fatto questo create un altro sensor, un altro controller e altri due actuators.

convertite il sensor in un COLLISION e nel campo PROPERTY scrivete BONUS; in questo modo la moneta reagirà agli oggetti con la property BONUS, che più avanti assegneremo ad Ernie.

-occupiamoci dei due actuators: convertiamo il primo in un EDIT OBJECT e cambiamo ADD OBJECT in END OBJECT, convertiamo il secondo in MESSAGE e scriviamo PRESO! nel campo SUBJECT; i MESSAGE sono actuator

molto utili, che possono trasmettere una condizione praticamente a tutti gli oggetti della Scene;  
-ora colleghiamo il tutto, facendo attenzione ad assegnare i due actuators allo stesso controller



così facendo la moneta sparirà al contatto con Ernie.

-ora, nella finestra 3d, duplicate la moneta: fatene quante volete, ma senza esagerare. ogni moneta varrà 5 punti.

Ora selezionate il cubo che contiene Ernie:

-create anche qui una property, al posto di FLOAT scegliete INT e chiamatela BONUS; assicuratevi che il piccolo tasto verde con la lettera D a destra dello slider sia attivato.

-aggiungete un nuovo sensor, un nuovo controller ed un nuovo actuator;

-convertite il sensor in un MESSAGE, e scrivete nel campo subject PRESO!, che corrisponde all'actuator MESSAGE della moneta.

-convertite l'actuator in un PROPERTY e cambiate ASSIGN in ADD; nel campo PROP scrivete il nome della property da editare (ossia BONUS), nel campo VALUE il punteggio di ogni singola moneta, ossia 5.

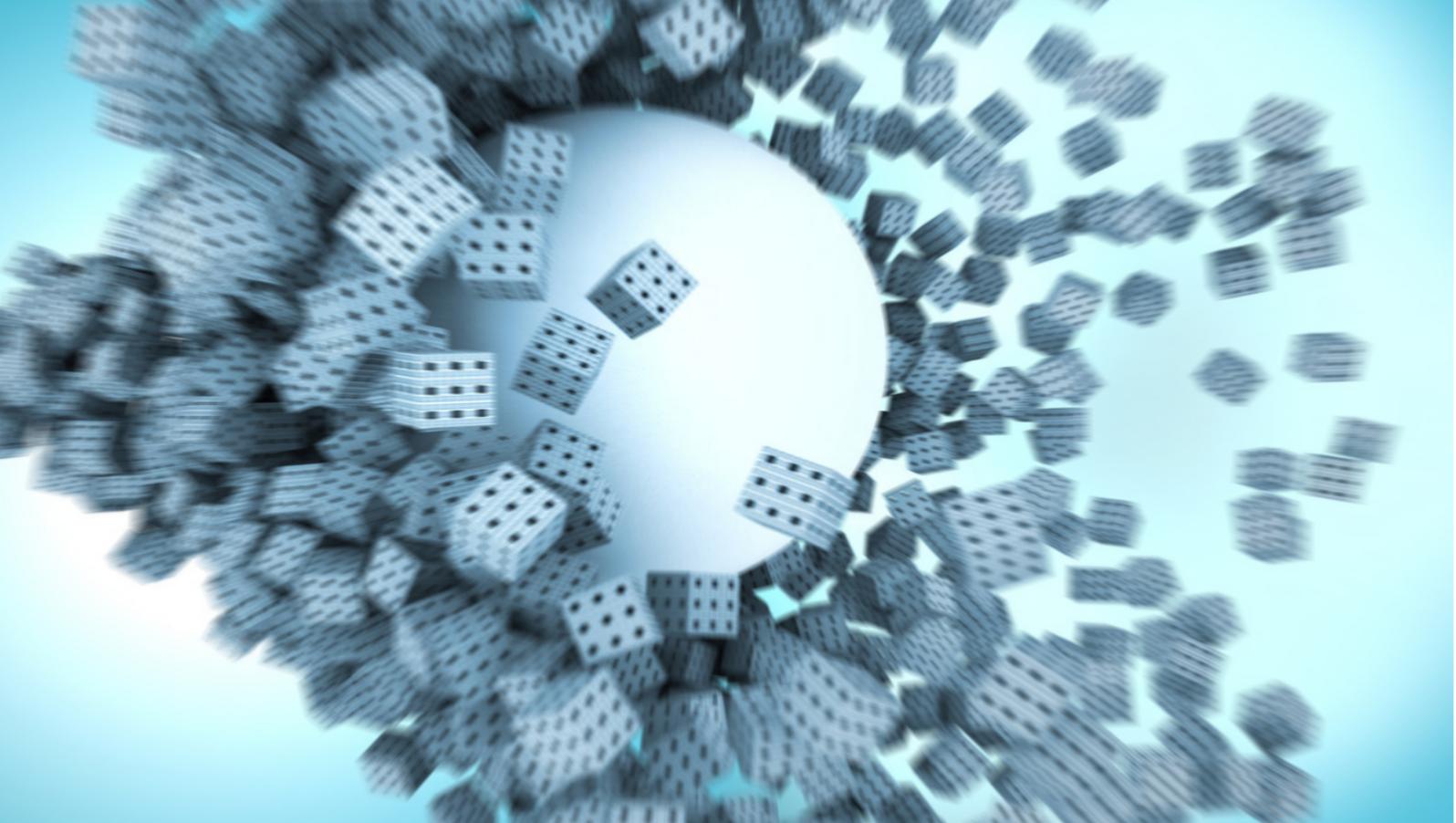
Ora premete P e provate il gioco: visualizzerete il valore della property BONUS in alto a sinistra sullo schermo; ogni volta che toccherete una moneta, questa sparirà e il valore BONUS incrementerà.

Beh, con questo era tutto.

Si chiude qui il nostro Tutorial: inutile che vi dica che ciò che abbiamo visto e provato in questa guida è soltanto la punta dell'iceberg. Le potenzialità del BGE sono praticamente illimitate e continuano a crescere di release in release, per non parlare poi degli orizzonti schiusi dall'implementazione con script python: l'ormai celebre progetto apricot è solo l'esempio più lampante.

Sperimentate liberamente gli altri logic bricks, non ponetevi limiti. Quello che vi serve è una buona dose di fantasia e, soprattutto, idee chiare.

Restate sintonizzati!



# Object e laser tracking

Approfondimento sull'object e sul laser tracking in Blender

di Marco Rapino

Uno dei miei hobby preferiti quando sono in pausa al lavoro, è setacciare il web in cerca di idee nuove che stimolino la mia creatività. Dire che il web sia un pozzo senza fondo di idee è dir poco, grazie alla rapida espansione della rete e il potenziamento dei motori di ricerca (soprattutto Google) oggi è molto semplice sapere se qualcosa che si ha in mente di fare sia stata già fatta, o eventualmente quali potrebbero essere i possibili approcci ai sistemi che si vogliono sviluppare. Come molti di voi probabilmente sapranno, poco tempo fa a Los Angeles c'è stato l'E3, la convention videoludica più importante dell'anno, che si svolge solitamente tra giugno e luglio negli States. Una delle grandi novità della convention è stato il progetto presentato da Microsoft, chiamato "Project Natal". Questo sistema attraverso una telecamera e algoritmi di motion tracking in realtime, permette al videogiocatore di interagire in maniera molto naturale con i giochi, semplicemente eseguendo i movimenti che vogliamo che il nostro avatar esegua all'interno del gioco.



Quindi per esempio in un gioco di tennis bisognerà muoversi come se stessi davvero effettuando un dritto o un rovescio, in un gioco di macchine dovremo simulare l'impugnatura di un volante e così via.

Seppur io sia abbastanza retrogamer e creda che al giorno d'oggi sia davvero difficile trovare qualità tecnica nei giochi, sono fortemente a favore dell'innovazione nel campo videoludico, ben vengano le evoluzioni. Così mi sono chiesto, ma nel mondo open source fin dove siamo arrivati in questo settore di ricerca e

innovazione? Da lì è partita la mia sfrenata e forsennata ricerca di articoli, librerie, mailing lists e spaccati codice. Conosceva già openCV in quanto nel centro di ricerca dove lavoro avevo avuto modo di usarlo nello sviluppo di sistemi di face recognition (riconoscimento di espressioni facciali, usati nell'HCI – Human Computer Interaction), quindi sapevo almeno da dove partire senza rischiare di perdersi nell'oceano di informazioni di cui Google dispone.

Da subito la scelta del linguaggio è stata chiara: Python. Pur avendo usato e usando tuttora diversi linguaggi di programmazione, il mio preferito resta sempre Python. Prototipare e sviluppare applicazioni in Python diventa davvero molto semplice dopo diverso tempo di esperienza da programmatore alle spalle. Soprattutto se sei passato per linguaggi davvero complessi nella loro sintassi quali C, C++, ObjC su tutti. Ovviamente usare Python porta degli svantaggi in termini di velocità (specie nelle computazioni matematiche), in quanto non è compilato, ma interpretato. Dunque i linguaggi C-oriented rimangono sempre i migliori se dovete lavorare con dei delay e tempi molto bassi. Una cosa fantastica che troviamo in Python è la semplicità con cui possiamo estendere il linguaggio creandoci librerie ad hoc, in C ad esempio o con altri linguaggi. Tornando però al nostro discorso, dopo un'attenta analisi di tutto il materiale trovato sulla rete, sono giunto alle seguenti conclusioni:

- Linguaggio: Python
- Librerie: openCV (la mia webcam non era supportata ho dovuto ripiegare con un workaround usando VideoCapture), PIL (Python Image Library), PyGame per la visualizzazione dell'output, Socket per la comunicazione interna con il BGE.
- Tipo di tracking: color-based, ovvero si “traccia” per colori e non per movimento che si ricava di conseguenza dal color-tracking

La differenza sostanziale tra color-tracking e motion-tracking è che nel primo si cercano oggetti colorati all'interno dell'immagine, mentre

nel secondo si comparano ad ogni step il frame attuale e quello precedente. Il confronto viene fatto cercando di capire come ogni pixel si è mosso da un frame all'altro. Per chi usa Voodoo Camera Tracker capirà cosa intendo se ha presente quelle “x” verdi e rosse che il programma genera durante il tracking.

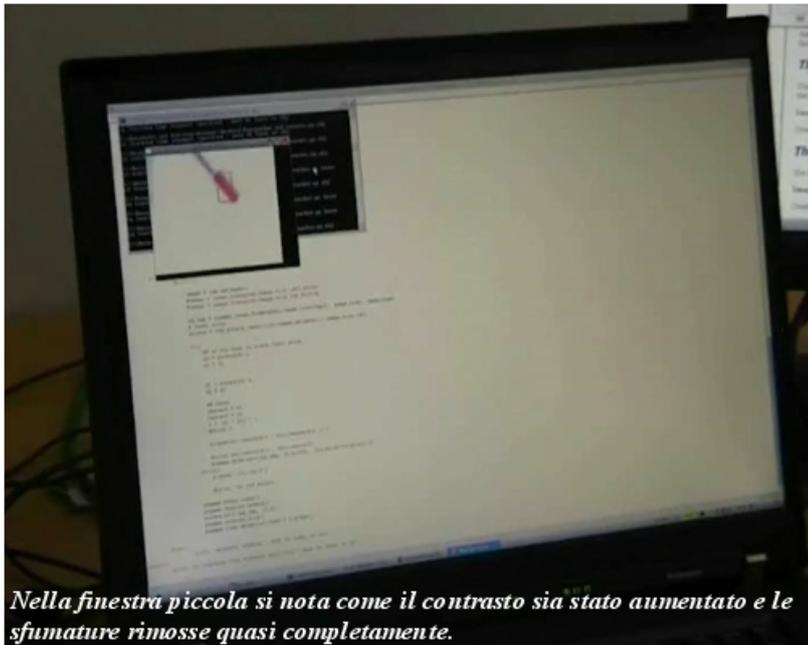
Un'altra domanda che ha trovato immediata risposta è stata: in quale ambito applicare tutto ciò? Ovviamente in Blender mi sono risposto e questo per due motivi:

- è l'applicativo che da 5 anni a oggi è parte integrante della mia vita
- volevo vedere fino a che punto era possibile dare il “la” ad una risposta open nei confronti di grandi progetti come quello Microsoft, pur essendo cosciente del fatto che competere con loro sarebbe ridicolo, ma cercare di dare piccoli segnali invece è sempre qualcosa di positivo

Spero di non essermi dilungato troppo finora, ma adesso scenderò nei dettagli tecnici di quella che è stata la realizzazione vera e propria. “Tracciare un colore” cosa vuol dire? Cerchiamo di immaginare cosa avviene nel nostro cervello quando abbiamo degli oggetti davanti e qualcuno ci chiede: qual è quello rosso? L'immagine che passa attraverso gli occhi, viene elaborata nel cervello e in seguito vengono trovate le zone rosse che a loro volta sono trasformate in coordinate che identificano la posizione dell'oggetto. Nel nostro programma dobbiamo replicare lo stesso meccanismo.

Cominciamo con un'immagine statica prima di passare ad un flusso quale può essere quello di una webcam. Gli step da eseguire sono i seguenti:

- Aprire l'immagine tramite il modulo PIL e ottenere la matrice di pixel
- Step obbligatorio per tracciare un oggetto, non necessario per il laser. Bisogna aumentare il contrasto dell'immagine e usare un filtro di tipo blur per togliere di mezzo tutte le sfumature fastidiose per stabilire il bounding box dell'oggetto.
- Una volta ottenuta la matrice la si può



*Nella finestra piccola si nota come il contrasto sia stato aumentato e le sfumature rimosse quasi completamente.*

analizzare in diversi modi, a seconda che si abbia un solo oggetto o più da tracciare. In questo articolo userò un solo oggetto, tuttavia se siete interessati a breve scriverò un nuovo articolo sul mio blog per il multi-tracking.

- In ogni caso andremo a tracciare il bounding box di ogni oggetto, ovvero il quadrato dell'area occupata dall'oggetto stesso. Io consiglio di usare una sfera in quanto l'area non cambia se la ruotiamo, questo può causare problemi nel caso in cui vogliamo tracciare la profondità.

- Stabilire una soglia oltre la quale decidiamo che il colore è da tracciare. La soglia si calcola usando la terna r,g,b dei pixel della matrice. Ad esempio nella mia applicazione ho voluto tener traccia

di un cappuccio di un pennarello rosso, dunque la mia soglia era:  $\text{threshold} = r - g - b > 30$ . Questo vuol dire che la componente rossa del pixel deve superare almeno di trenta la somma delle altre due. La soglia invece per tracciare il laser è la seguente:  $(r+g+b) / 3 > 220$ . Dal momento che il laser è un raggio luminoso, il suo centro sarà prevalentemente bianco.

- Una volta stabilita la soglia, nella mia applicazione ho deciso di tracciare un solo oggetto, in quanto avrebbe reso più semplice lo sviluppo e l'interazione con il BGE. Quindi ecco cosa ho fatto. Ho analizzato la matrice andando a cercare il primo pixel che fosse entro la soglia e l'ho salvato in una struttura dati temporanea ad hoc. Successivamente ho rovesciato la matrice (sotto forma di lista) e fatto lo stesso procedimento in modo tale da trovare l'ultimo elemento entro la soglia salvando anch'esso nella classe Point da me creata, dove salvo le coordinate del pixel e intensità del punto (il valore soglia).

- Passo i pixels a pygame che mi mostra l'immagine con un quadrato disegnato in overlay che ha come coordinate il primo e l'ultimo punto "rosso" da me tracciati.

A questo punto è facile immaginare che lo step successivo, ovvero quello di eseguire lo stesso procedimento su uno stream video, sia molto semplice. Difatti basta ripetere la procedura per ogni frame che otteniamo dalla webcam. Leggere i dati da una webcam è una riga di codice, quindi fate riferimento ai sorgenti per capire come fare. Un'ultima cosa che vi consiglio è di usare uno sfondo di colore uniforme che sia a contrasto con l'oggetto. Questo facilita notevolmente il tracking.

Una volta ottenuto il bounding box, useremo il suo centro come coordinate di input x,z nel BGE e la sua area per determinare la profondità y. Entrambe si calcolano con semplici operazioni matematiche di base:

Centro:  $(x1+x2)/2, (y1+y2)/2$

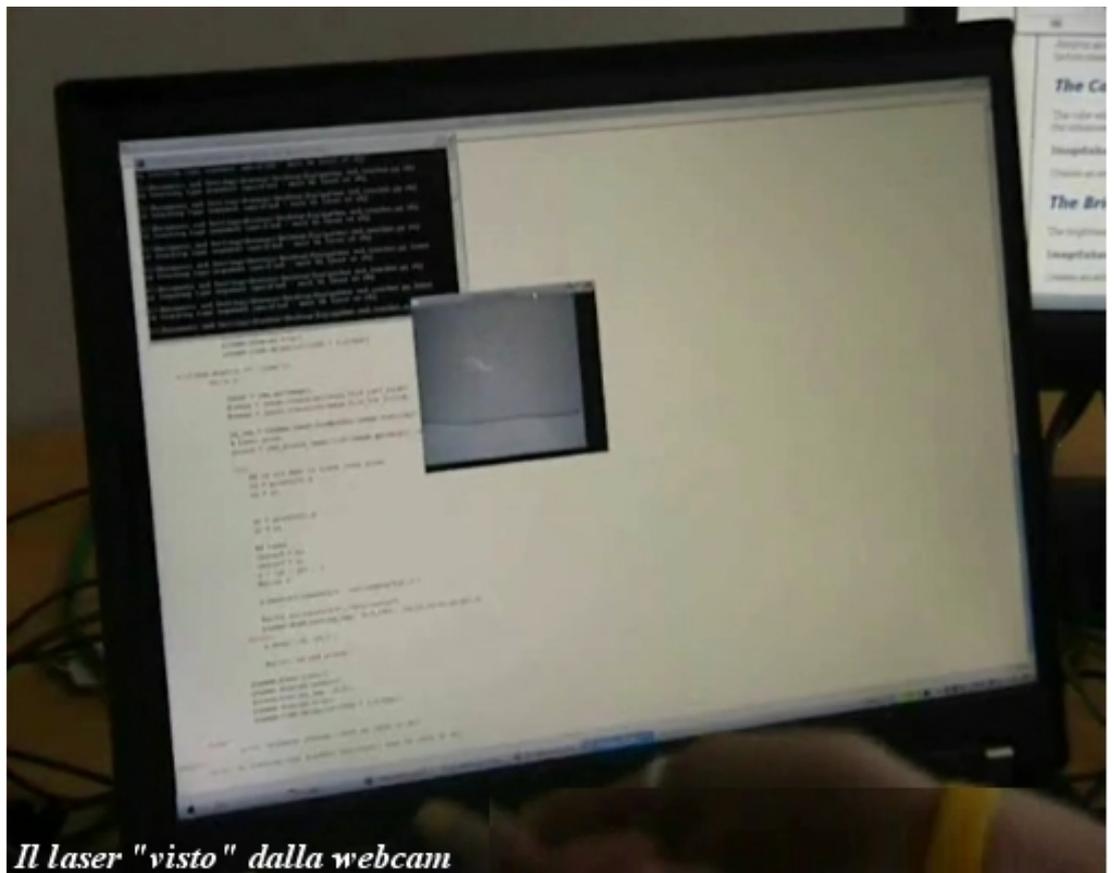
Area:  $(x2-x1)*(y2-y1)$

Dove  $x1,y1$  è l'angolo in alto a sinistra e  $x2,y2$  l'angolo in basso a destra.

Finora ho parlato solo di object tracking e non laser. La procedura è la stessa solo che come potete immaginare il laser ha un bounding box piccolissimo (praticamente 4-5 pixels) e costante. Dunque è molto più facile e preciso per tracciare lungo gli assi x e z in Blender, ma è impossibile ottenere la profondità in quanto l'area non cambia mai.



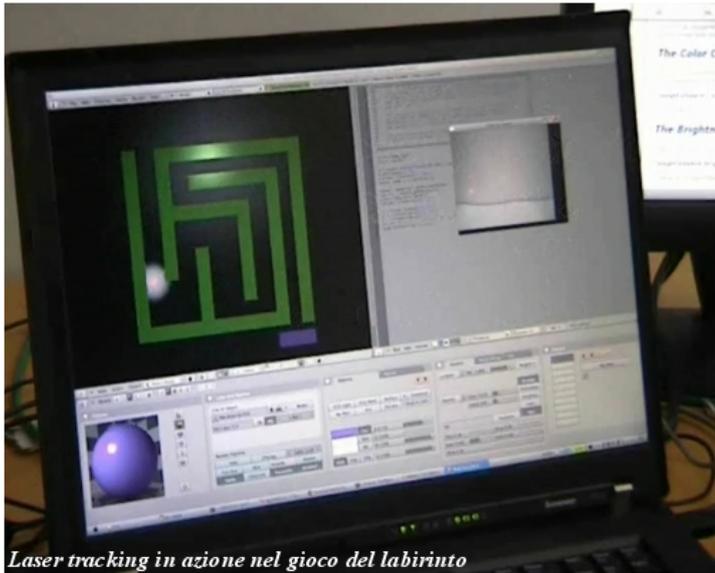
*Il laser proiettato sul muro. La webcam e' posizionata di fronte, sopra la scatola di cartone bianco.*



*Il laser "visto" dalla webcam*

Ora è arrivato il momento di integrare il tutto con Blender!

Per far ciò è necessario avere la parte di tracking come un thread separato per evitare lag nel BGE, quindi useremo dei socket UDP non bloccanti interni per far comunicare i due componenti. Nel BGE ciò che avviene è molto semplice. Ad ogni step riceviamo tramite socket i valori dal tracking-component e li usiamo per muovere il nostro cubo o altro oggetto nel BGE semplicemente richiamando la funzione `setPosition` con python. Il risultato lo potete vedere nel video che ho messo sul mio blog all'indirizzo <http://www.aktasway.com/blog>.



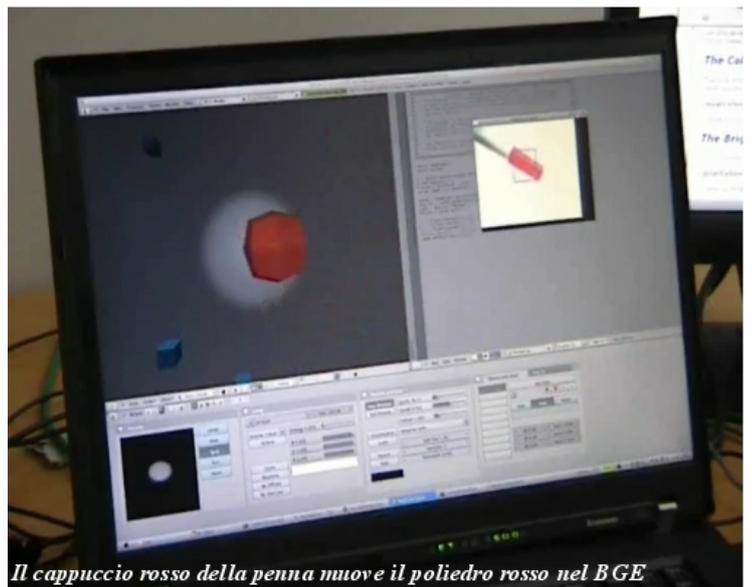
*Laser tracking in azione nel gioco del labirinto*

Ho realizzato due file blend. Il primo è un labirinto nel quale dovete muovervi senza toccare i bordi. Questo utilizza il laser. Invece il secondo è un solido rosso che interagisce con cubi blu nel BGE. Sia i cubi che il solido da noi comandato sono rigid bodies e potrete notare come tramite Bullet Engine sia possibile fisicamente interagire con essi.

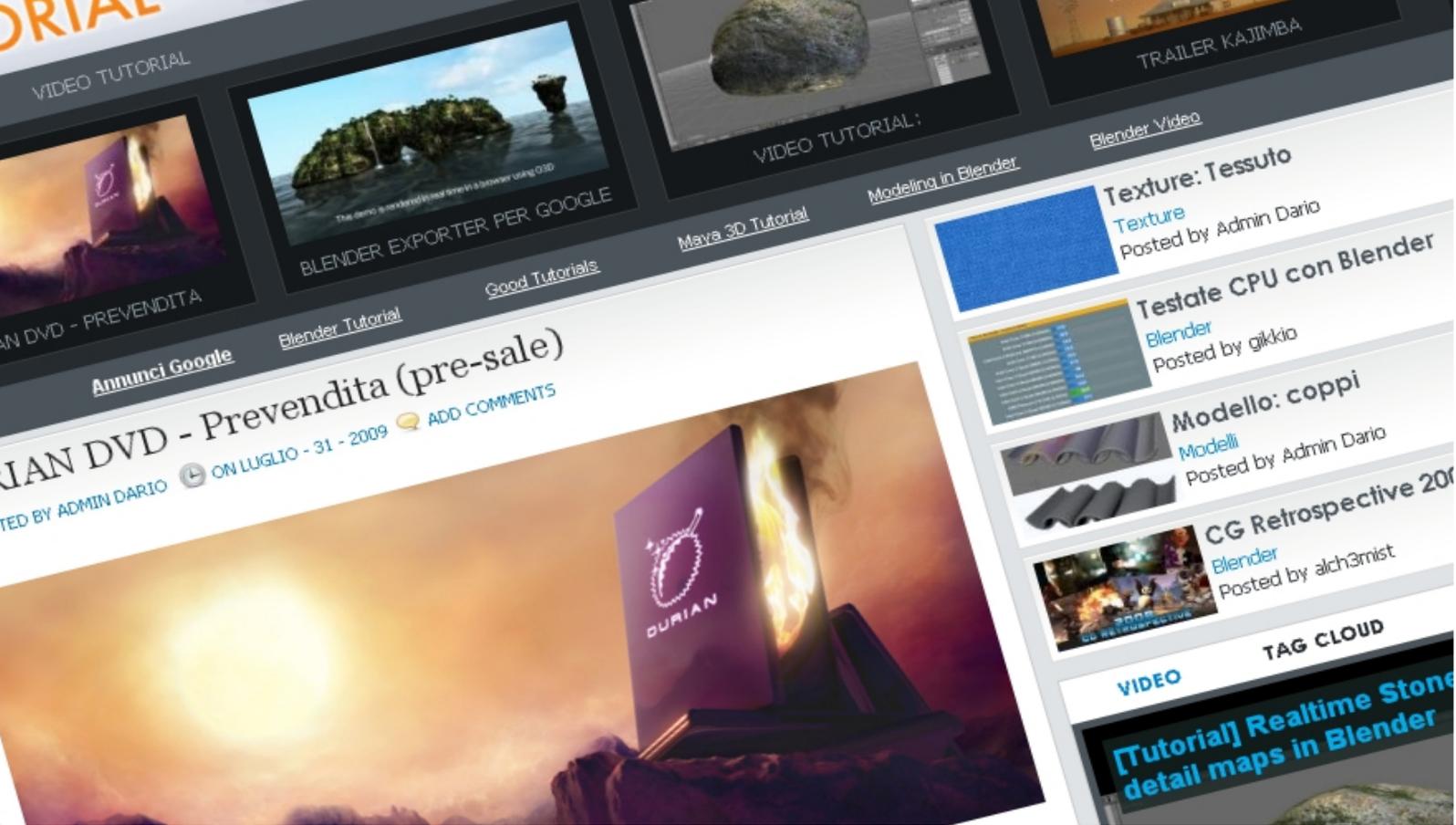
Da qui in poi sta tutto alla vostra creatività e fantasia. Blender ha potenzialità davvero infinite da questo punto di vista, dal momento che può essere esteso grazie alla presenza integrata di Python. Infatti sempre sul mio blog potrete trovare un esempio su come usare l'accelerometro del Nokia N95 nel BGE e non

solo. Sarebbe interessante vedere come la cosa si potrebbe evolvere, purtroppo a causa del mio lavoro e altri impegni personali non ho tempo a sufficienza per creare un sistema che vada oltre il prototipo, dunque mi affido agli altri programmatori e alla loro creatività per sviluppi futuri!

In conclusione ancora una volta abbiamo dimostrato come l'open source non è assolutamente da meno ai sistemi proprietari. Anzi molto spesso questi ultimi partono proprio da progetti open per poi svilupparsi a seconda delle esigenze dettate dalle aziende o dal mercato. Se non fosse stato per la filosofia dell'open e del condividere la propria conoscenza, molto probabilmente non sarei arrivato mai a sviluppare un'applicazione simile e tale sistema sarebbe rimasto nelle mani di "pochi" eletti, rallentando la crescita dell'innovazione e limitando l'accessibilità della conoscenza a persone "normali" come noi. Quindi, come Blender ogni giorno mette a disposizione strumenti potentissimi, gratuitamente, credo sia opportuno prenderlo di esempio e continuare a sviluppare rilasciando le proprie applicazioni sotto licenza Creative Commons.



*Il cappuccio rosso della penna muove il poliedro rosso nel BGE*



# BT: intervista all'admin

Blender-Tutorial: la nascita di una nuova community

di Dario Caffoni (admin di Blender-Tutorial.com)

Si parla ormai di più di un anno fa quando mi venne l'idea di creare un blog strettamente legato a Blender. Vi racconto la storia dal principio e quindi dalla "scoperta" da parte mia di questo splendido programma.

Mi sono imbattuto in Blender circa 2 anni fa, da prima ero un utente di cinema4d (ottimo programma a mio avviso), ma per esigenze lavorative cercavo un programma gratuito che mi permettesse di fare quel poco 3d che la mia azienda necessitava. Scoprii così Blender, programma open source che dalle caratteristiche sembrava proprio fare al caso mio.

Arrivando da un altro programma 3d non fu facile all'inizio riuscire a realizzare qualcosa che risultasse perlomeno guardabile, ma andando avanti con lo studio e l'applicazione del programma i primi risultati arrivarono nel giro di qualche settimana.

Scoprii ben presto che Blender è un programma fantastico, complesso ma completo, tutto quello che avevo in mente di fare lui lo permetteva.

Bisognava documentarsi a dovere e non sempre

era facile reperire il materiale in lingua italiana, ed è da qui che nasce prima di tutto "l'amore" per Blender ma anche l'esigenza e la voglia di fare qualcosa per aiutare chi come me aveva avuto bisogno di una guida, un tutorial, un aiuto per proseguire la sua avventura con Blender.



Decisi di realizzare da prima un Blog dove inserire Video Tutorial (realizzati personalmente con il mio poco sapere) riguardanti le basi di Blender, non nascondo che la mia fortuna è stata

quella di essere nel campo della web graphic e avere quindi a portata di mano diversi strumenti indispensabili (conoscenza grafica e web), nel giro di qualche giorno fu tutto pronto, dominio, wordpress con tema personalizzato installato e qualche video tutorial realizzato...e da qui che partì l'avventura con Blender-Tutorial

Agli inizi è stata davvero dura lavoravo tutti i giorni fino a notte tarda per organizzare i contenuti, crearne dei nuovi cercando nel contempo di farmi un po' di pubblicità nelle community già esistenti.



Erano tutti un tantino scettici (giustamente), le visite giornaliere erano davvero poche, ma io non mollavo anche perché mi piaceva fare quello che stavo facendo (che tutt'ora faccio) ed ero fiducioso che pian piano le cose sarebbero cambiate e così è stato, blender-tutorial piano piano si arricchiva di visitatori che ringraziavano per i video tutorial e chiedevano supporto, c'era un via vai continuo di persone, si stava pian piano animando insomma. Capii ben presto che si poteva fare di più, molto di più per gli utenti di

blender, partirono così diverse migliorie riguardanti il blog come l'aggiunta di una sezione delle news (la più ricca ora), una sezione sulle textures, modelli, materiali e in fine un forum dedicato.

Proprio per questo ho deciso di creare un nuovo forum e oggi posso dire che ne è valsa la pena Blender-Tutorial da quei giorni è cresciuto, un sacco d'iniziative nate dagli utenti sono state realizzate e molte altre sono in dirittura d'arrivo, ve ne indico alcune:

- Il BOMP (Blender Online Movies Project), gruppo di studio per la realizzazione di corti utilizzando blender, ideata e magistralmente amministrata da Alfonso Annarumma (Anfeo).
- Blender-Textures, portale fratello di blender-tutorial dove ci sarà una ricca raccolta aggiornata giornalmente di textures in alta risoluzione libere da diritti, il portale sarà in doppia lingua italiano ed inglese, qui mi avvalgo oltre che alla collaborazione dello staff di Blender-Tutorial (Gikkio e Seregost) anche di Max Puliero (MmAaXx).
- Le interviste di Andrea Fiocca (Gikkio) agli artisti di tutto il mondo che utilizzano Blender professionalmente con risultati davvero strabilianti.
- L'organizzazione di videocorsi online gratuiti con serate a tema (modellazione, Game Engine, Nodi, ecc) dove l'utente potrà comunicare con l'insegnante in tempo reale.
- La traduzione del Wiki di Blender dall'inglese all'italiano e successivi aggiornamenti...e molto altro che potrete scoprire voi stessi seguendo le attività del portale e del forum di Blender-Tutorial.

Siamo arrivati alla conclusione di questo articolo sulla nascita e futuro sviluppo di Blender-Tutorial, non mi resta che ringraziare tutte le persone che contribuiscono giornalmente a tenere viva ed interessante la community, in particolare Andrea Fiocca (Gikkio) e Damiano Buffin(Seregost).

Saluti



# Making of "Galleria"

Creazione di una scena architettonica con Blender e l'internal render

di hi\_volt (Ettore Mele)

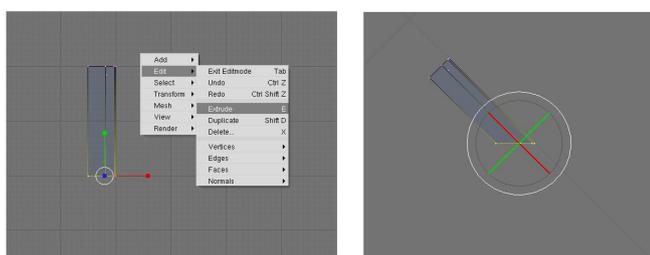
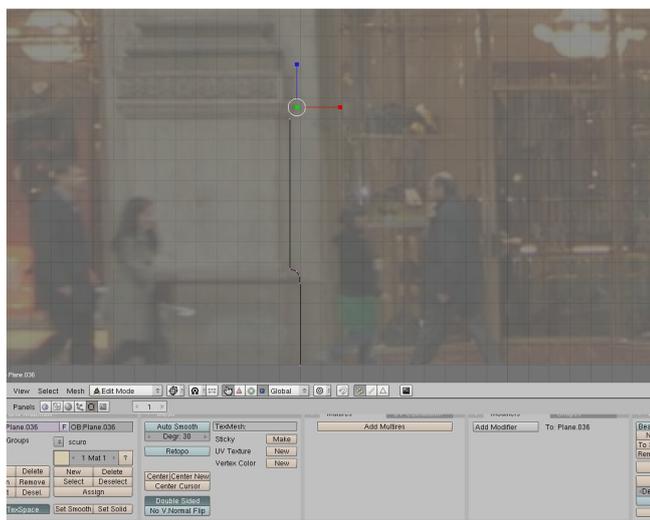
“Galleria” è stato per me un eccellente campo di prova per quel che riguarda la modellazione e l'illuminazione di Blender. Gli obiettivi del lavoro erano la realizzazione di una scena realistica con l'internal renderer e l'ottimizzazione del sistema di illuminazione per render veloci. Di seguito vi illustrerò in alcuni passaggi i punti salienti della scena e come sono stati realizzati. Questo articolo non è un vero e proprio tutorial e sottintende la conoscenza di alcuni comandi e meccanismi di Blender e Gimp:

1- Preparazione della scena: prima di iniziare a modellare ho cercato una foto su internet, una volta trovato un riferimento soddisfacente sono passato a Gimp per le fasi di raddrizzamento utilizzando le guide e lo strumento prospettiva.



In questa fase bisogna fare molta attenzione perché non è possibile fare un raddrizzamento preciso a causa della sfericità degli obiettivi fotografici. Inoltre gli elementi discostati dal piano di riferimento tendono a creare errori prospettici. L'importante è avere un riferimento abbastanza preciso per quel che riguarda le proporzioni e tenere conto dei difetti di questa tecnica comporta.

2 - Modellazione di base (facciata): In questa fase si procede con la modellazione di tutti gli elementi di base della scena, ossia: oggetti (elementi che fuoriescono), archi, balconate. Per modellare in Edit mode si traccia un profilo della colonna e si estrude dalla vista dall'alto. Si preme 3 volte il numpad 4 o 6 (a seconda di come si vuole ruotare la camera), si ruotano i vertici di 45 gradi (utilizzando CTRL per una maggiore precisione), si imposta il "3d transform manipulator" in modalità view e si usa il comando scala (sempre con CTRL premuto) sfruttando l'asse rosso per allineare i vari vertici. Da qui si può estrarre di nuovo e ripetere i passaggi precedenti per ricreare tutti gli oggetti della scena.



Per gli archi è stato utilizzato il comando spin (in Edit mode) fissando il cursore al centro dell'arco. Nel mio caso specifico era importante non dare un eccessivo numero di step per permettere di estrarre le facce inferiori e creare le decorazioni semplicemente scalando le facce estruse in modalità "individual centers" dei pivot. La stessa tecnica è stata utilizzata per realizzare la copertura della galleria. Al termine di questa fase è stato aggiunto un piano come oggetto separato per ricreare il muro di base. Per velocizzare il lavoro è stato modellato solo un modulo, per i successivi è stato aggiunto un modificatore Array.

3 - Elementi di dettaglio e decorazioni (prima fase): Giunti a questo punto si modellano elementi come le finestre ed i capitelli. Si segue l'immagine di riferimento ricalcandola con CTRL+ Click e spostando i vertici creati per dare profondità al modello. Creata la serie di vertici si può procedere alla creazione delle facce selezionando 4 vertici per volta ed utilizzando il comando F.

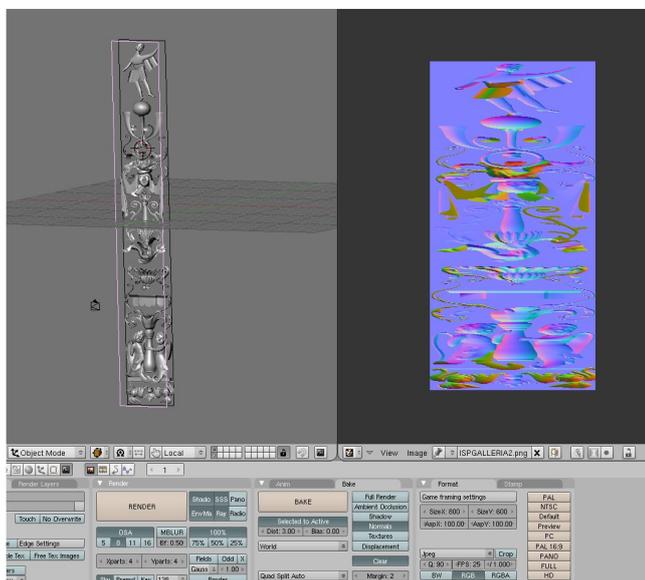


Un discorso a parte meritano i volti ed i fregi della balconata superiore che sono stati realizzati con lo sculpt. Per i volti è stato utilizzato un cubo suddiviso 4 volte e trasformato in sfera col comando "to sphere" (100%). Da qui si è iniziato a scolpire partendo dai dettagli più grossolani e affinando man mano. Al termine la parte retrostante del volto è stata eliminata per ottimizzare il numero di poligoni della scena. All'inizio avevo realizzato con questa tecnica anche la decorazione delle paraste, ma la scena era troppo pesante ed il risultato non

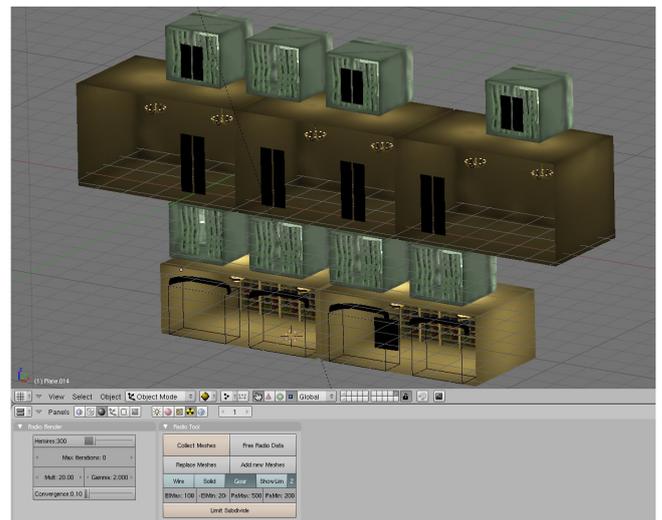
soddisfacente, quindi ho utilizzato la tecnica utilizzata nel passaggio successivo.



4 – Paraste: per la realizzazione di questa parte del modello un'immagine più dettagliata del particolare ha permesso di poter ricreare un modello 3d. Il bake è stato utilizzato per renderizzare displacement e normalmap. Sono state renderizzate entrambe poiché le normalmap considerano solo le variazioni di angolo, mentre le displace considerano l'altezza; in questo modo si ottiene un maggior quantitativo di informazioni e di conseguenza un maggior dettaglio. La modellazione in questa fase non è stata molto pulita, considerando che il risultato avrebbe svolto solo una parte marginale. Una volta terminata la modellazione il modello è stato mappato su un piano nel modello della facciata.



5 – Interni ed il radiosity: tutti gli interni sono scatole aperte dal lato finestra con normali invertite verso l'interno (altrimenti radiosity non riconosce le pareti), qualche oggetto d'arredo ed un materiale emit in corrispondenza delle fonti di luce. Per sfruttare la radiosity di blender si seleziona la scatola con tutto il contenuto, si accede al pannello radiosity, si imposta Gour invece di solid e si clicca su collect meshes, dopodiché si clicca su go e blender inizia il calcolo della radiosity. Essendo una tecnica che sfrutta il vertex color per creare l'illuminazione potrebbe essere necessario suddividere alcune volte gli elementi più semplici. Al termine del calcolo si clicca su "replace meshes" e su "free radio data" ed il risultato è una mesh contenente l'illuminazione. Le 3 stanze sono state copiate più volte. Se si intende utilizzare questa tecnica in un lavoro conviene copiare l'oggetto o salvare prima di procedere col calcolo della radiosity, poiché è una tecnica "distruttiva", le mesh vengono suddivise, scombinare e riunite in un unico oggetto. Ricordo inoltre che il numero massimo di materiali permesso per ogni calcolo di radiosity è pari a 16 (emitter inclusi).



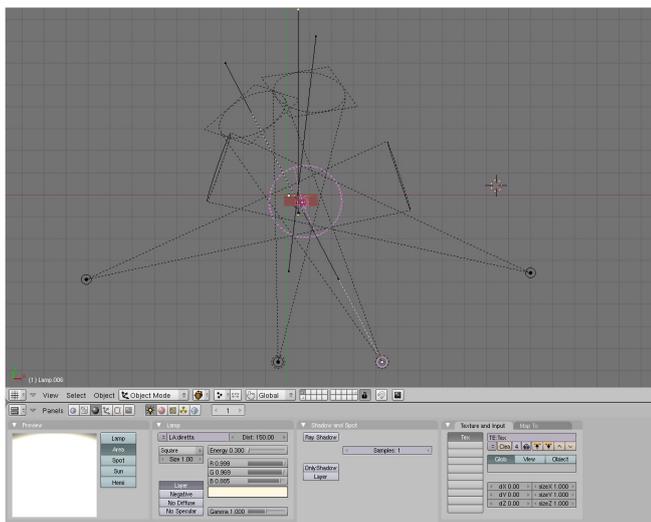
6- L'illuminazione: questa fase è stata in assoluto la più difficile di tutto il lavoro per i seguenti motivi: assenza di Ambient Occlusion per poter fare un render veloci e necessità di resa convincente, quasi fotorealistica. La scena è stata suddivisa in 3 layer, ciascuno illuminato in modo diverso:

il layer della facciata illuminato con 6 luci: 2 all'estrema destra e sinistra del modello per schiarire le parti più nascoste (spot senza ombre), una spot dall'alto blu con ombre per ricreare il cielo, altre 2 spot per creare le ombre principali e le secondarie ed una arealight senza ombre.

Il layer degli interni: anche se è già illuminato tramite radiosity, è stata aggiunta un'arealight per poter bilanciare le differenze tra l'illuminazione della facciata e degli interni per creare un'illuminazione più uniforme.

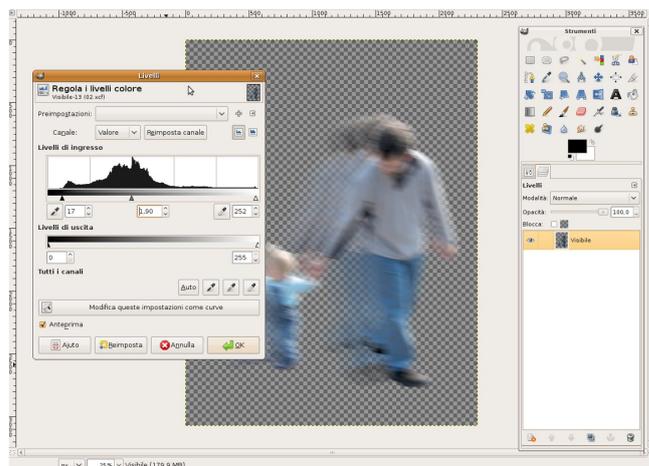
Il layer della copertura: questa parte ha solo una spot che proietta le ombre della galleria ed un piano utilizzato come ingombro. Nella realtà le ombre non sono accentuate come nel render, ma servivano per far capire che ci si trova dentro una galleria. E' più che altro una scelta "artistica" che fornisce informazioni su ciò che non è renderizzato direttamente.

(tutte le luci con ombre utilizzano shadowbuffer e non rayshadow che rallenta le fasi di rendering) Come spiegavo prima la difficoltà maggiore sta nel posizionare correttamente le luci e dare valori di intensità adeguati alla scena, per questo sono stati fatti molti test. Purtroppo non esiste una configurazione universale per l'illuminazione, ma ogni scena va a sé e dipende molto dal tipo di modello e dal tipo di illuminazione che si vuole creare, per questo motivo non entro ulteriormente in dettagli.



7 – Materiali: I materiali sono strettamente collegati all'illuminazione, perché reagiscono in

base a questa. Dopo aver scelto dei materiali di base per la scena la procedura è stata la taratura alle diverse altezze in base all'intensità dell'illuminazione. Dallo stesso materiale ne derivano così altri con alcuni valori leggermente modificati. Per il pavimento e le persone in movimento invece è stato utilizzato il linear workflow. In Gimp non esiste il comando gamma, o meglio è un po' nascosto, quindi spiegherò come trovarlo per applicare il linear workflow. Si accede dal pannello Colori – Livelli. A questo punto la gamma viene regolata dalla prima riga attraverso i 3 triangolini. Innanzitutto si spostano gli estremi verso l'interno se non ci sono dati (se il grafico è piatto), dopodiché si procede con lo slider centrale e si fissa un valore che in generale è compreso tra 1,8 e 2 (dipende dalla taratura del monitor). Io ho scelto 1,9.



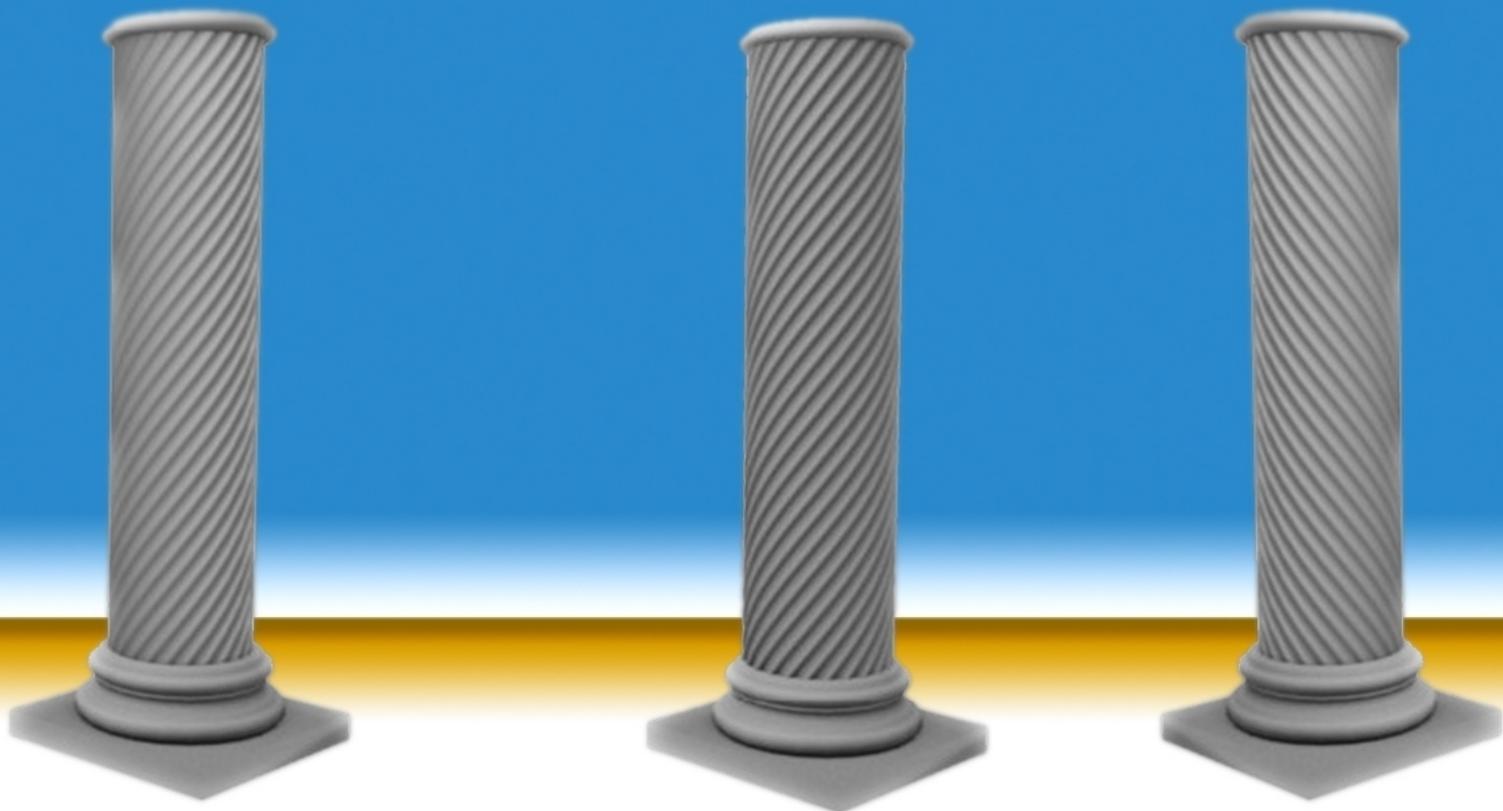
8 – I personaggi: Per dare un senso delle proporzioni degli edifici di solito vengono aggiunte delle figure umane nella scena. Esistono diversi metodi utilizzati da vari architetti, tuttavia nella maggior parte dei casi hanno un "effetto modellino". così ho deciso di affidarmi a tecniche fotografiche per ottenere una resa più realistica. Le immagini delle persone sono state sfocate per dare un senso di movimento e "dare vita" alla scena. La parte più complessa di questa fase riguarda il tipo di mappatura necessaria per ottenere un alpha sfumata in fase di render utilizzando soltanto una immagine png con valori di trasparenza inclusi. Dopo aver aggiunto un materiale e caricato la

texture si preme il pulsante "Use Alpha". Tornando al pannello materiali si imposta ztransp con valore di alpha pari a 0, il tipo di mappatura (nel mio caso uv). Nel pannello "map to" si seleziona "Col", "Alpha" e più in basso "Stencil". Quando verrà selezionato quest'ultimo si noterà immediatamente che nella preview il materiale cambierà assumendo l'aspetto desiderato.



9 – Conclusioni: Nonostante ci siano ancora margini di miglioramento (ci sono sempre nella grafica 3d, per fortuna!) l'obiettivo preposto è stato pienamente raggiunto, con render velocissimi per una scena del genere. In particolare alla risoluzione di 2400 x 1800 il render è avvenuto al di sotto dei 7 minuti su un quad core. Il che significa che anche su pc meno performanti i tempi di attesa sono comunque accettabili e dovrebbero attestarsi al di sotto dei 30 minuti. Inoltre con l'arrivo della versione 2.5 di blender sono previsti miglioramenti nel motore di rendering, quindi in futuro dovrebbe essere ancora più veloce!

Come nota conclusiva vorrei fare un ringraziamento particolare agli iscritti del forum che mi hanno aiutato e dato spunti interessanti per raggiungere questo livello. Cito i loro nick in ordine di post: Joystx\_ , johnj , GIKKIO , lssc , darkblender , Chrysophylax , EnV , LiberatePoliceto , UncleZeiv , Gillan. Ringrazio inoltre lo staff di Blender Magazine Italia per aver preso in considerazione la realizzazione di questo articolo.



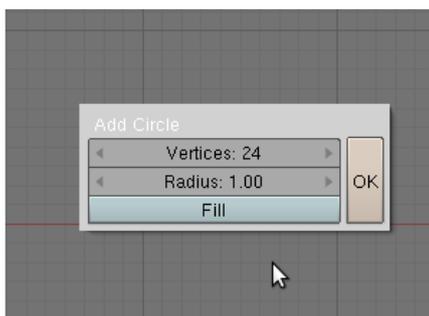
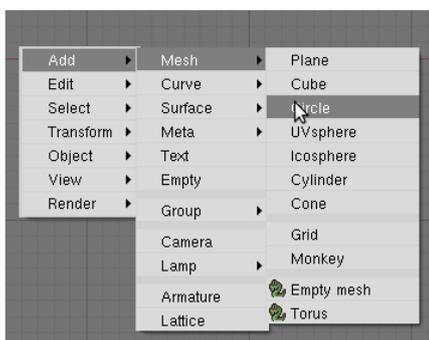
# Colonna decorata

Realizzare una colonna decorata a spirale in Blender

di Alessandro Battisti

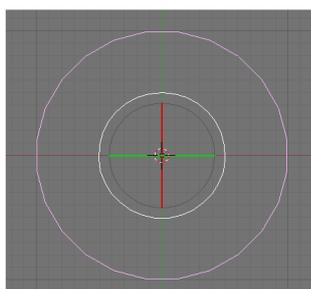
01

In vista top inserire un cerchio con 24 lati.



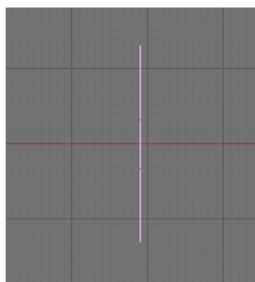
02

Ruotare di 7.5 gradi in modo da avere un lato parallelo all'asse y.



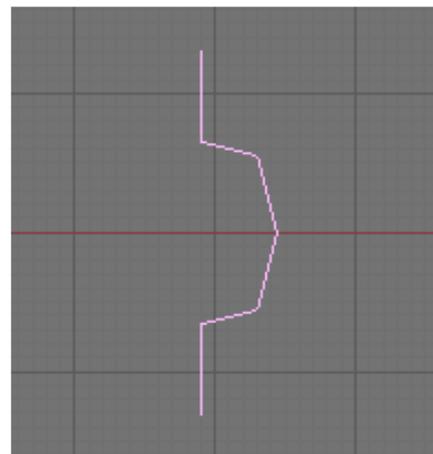
03

Cancellare tutte le facce lasciando solo una delle 2 parallele all'asse y.

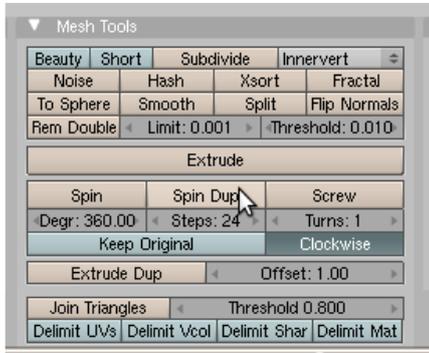


04

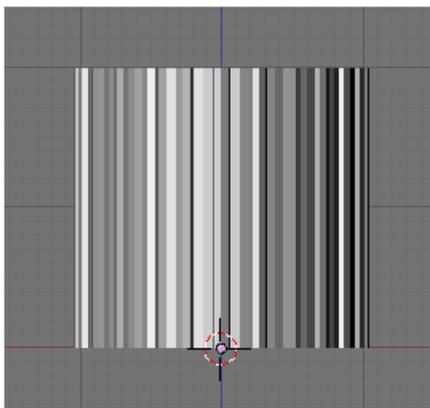
Suddividere il lato e muovere i vertici in modo da avere qualcosa di simile a questo.



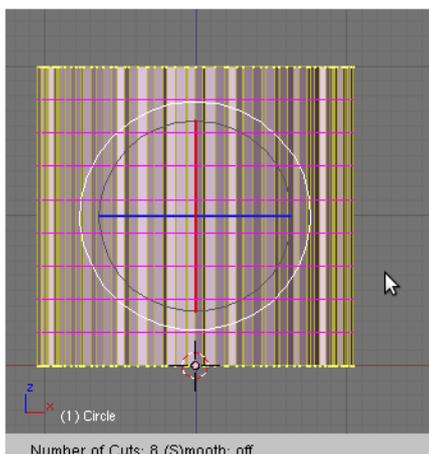
**05** In vista top effettuare uno spindub con il puntatore al centro del cerchio iniziale. 360° 24 steps.



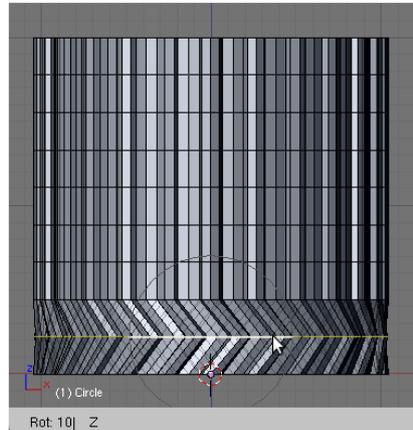
**06** In vista front eliminare i vertici doppi ed estrarre in alto di 2



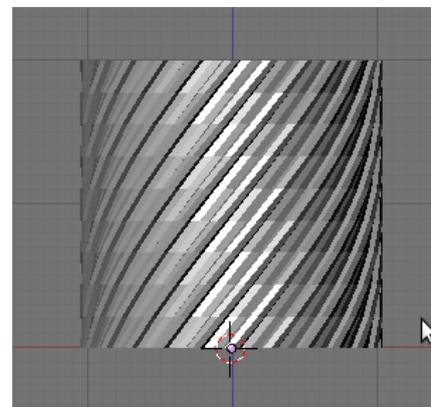
**07** Suddividere l'oggetto con Ctrl-R e muovendo in alto la rotellina del mouse fare 8 tagli



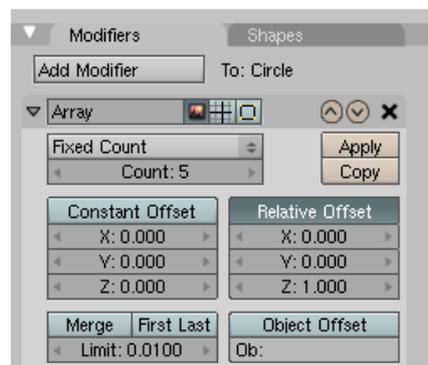
**08** Sempre in vista front selezionare il primo taglio con Shift-Alt tasto destro del mouse e ruotare di 10° sull'asse z.



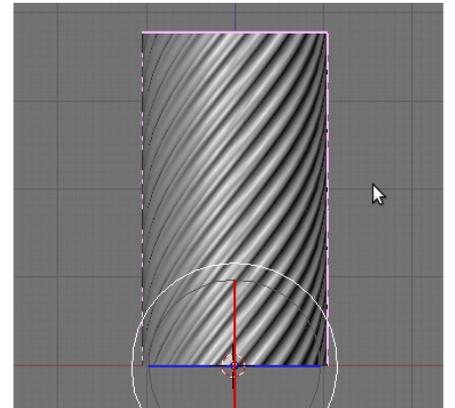
**09** Selezionare il secondo taglio e ruotare di 20° poi così via fino al margine superiore che va ruotato di 90°



**10** Applicare un modificatore array, Relative offset 1 sull'asse Z e quanti counts si desiderano.



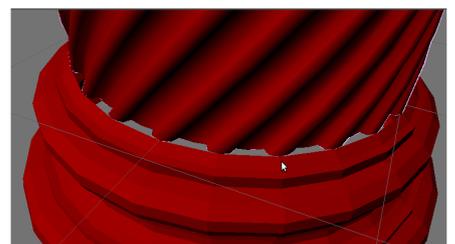
**11** Applicare il modificatore, rimuovere i vertici doppi. Ricalcolare le normali all'esterno, mettere lo smooth.



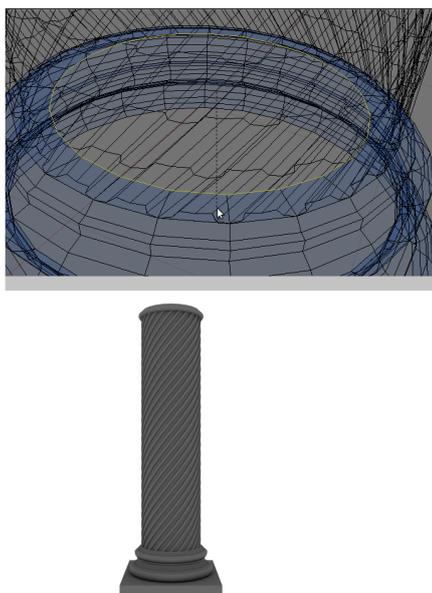
**12** Fare il profilo della base partendo dal vertice più esterno da vista front che dovrebbe essere a metà colonna in vista top



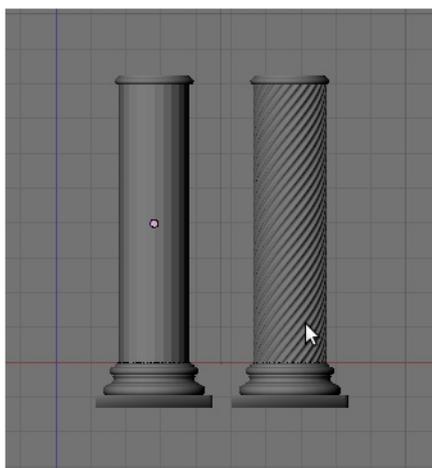
**13** Fare uno Spin. da vista top 360° 24 steps, eliminare poi i vertici duplicati. Si viene a creare però un buco tra colonna e base.



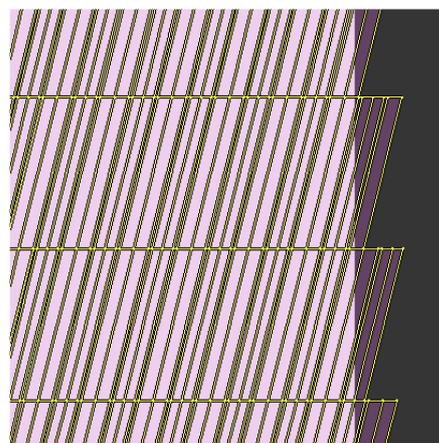
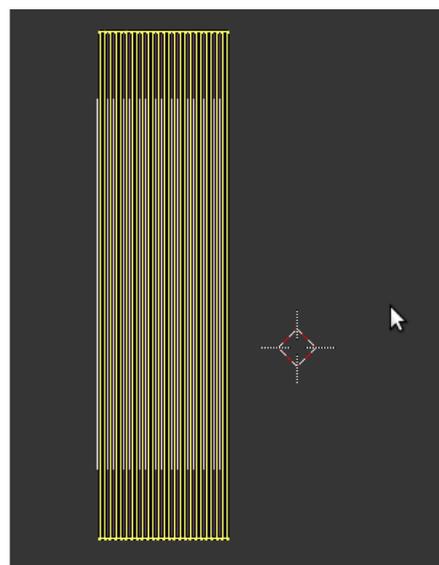
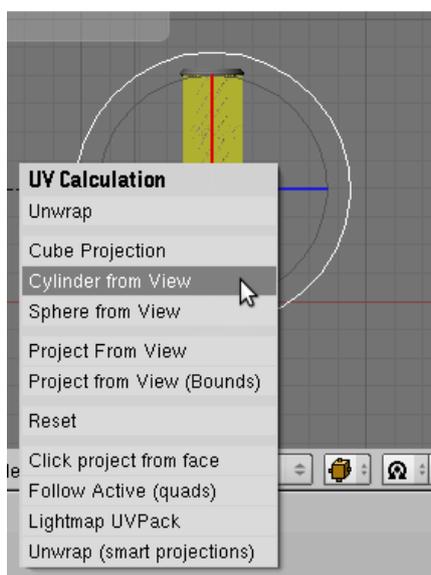
**14** Per riempirlo selezionare il margine superiore della base ed estrarlo verso l'interno premendo E e poi S. Per evitare problemi di smooth selezionare la base e farne un oggetto a parte con P. La colonna è completa.



**15** Per ridurre il numero dei poligoni si può utilizzare una normal map. Si crea un cilindro con 24 lati e lo si scala in modo che coincida con la colonna.



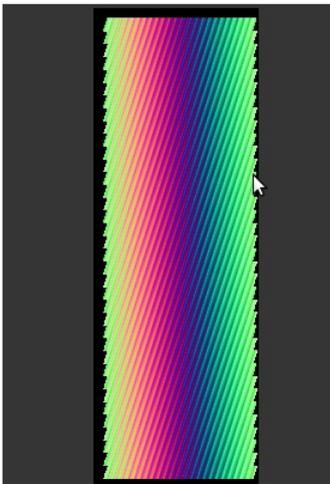
**16** A questo punto si fa l'unwrap della colonna decorata in vista front, nell'Image Editor si crea una nuova immagine di dimensione piuttosto allungata io ho messo 1000 di larghezza e 3000 di altezza. Poi in modalità edit mode, si selezionano tutte le facce, si preme U e si seleziona Cylinder from View.



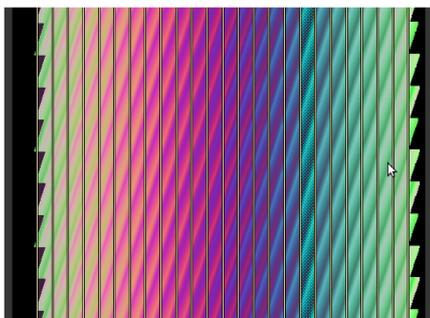
**17** Nell'Image Editor le facce sono molto più grandi dell'immagine per questo si scala in modo da far sì che siano contenute per intero nell'immagine, è necessario scalare sia sull'asse delle x (io di 0.9) e sull'asse delle y (io di 0.7) in quanto probabilmente una piccola parte delle facce esce anche sulla destra dell'immagine. È importante scalare con valori conosciuti perché successivamente sarà necessario scalare dello stesso fattore anche il cilindro.

**18** Ora si va a creare la mappa delle normali, nel pannello scene si va su Bake, si selezionano normals, e modalità object, in questo modo le normali saranno calcolate sulle coordinate dell'oggetto che potrà essere successivamente anche scalato rotato e spostato. Salviamo il risultato.

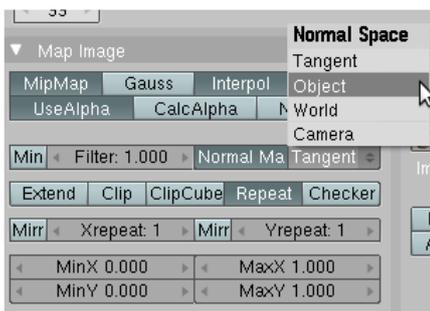




**19** Ora si seleziona il cilindro, nell'Image editor si apre la mappa delle normali appena salvata. In edit mode si fa l'unwrap del cilindro nello stesso modo di prima. In image editor si scalano le facce con le stesse proporzioni usate prima per la colonna decorata e il risultato dovrebbe essere più o meno questo.



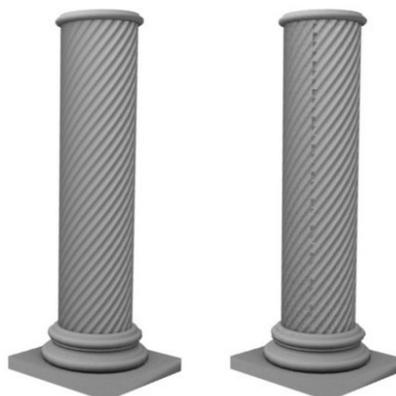
**20** Nel materiale del cilindro, si aggiunge come texture la mappa delle normali, selezionando il pulsante normal map e si seleziona object.



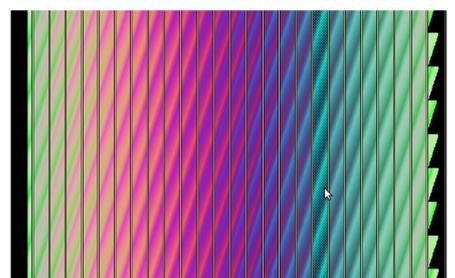
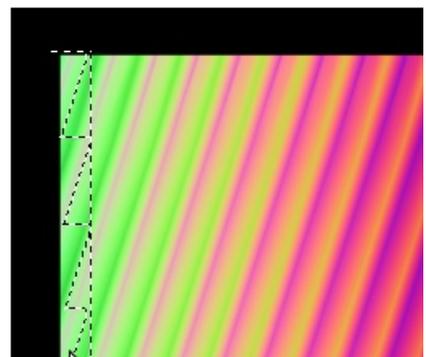
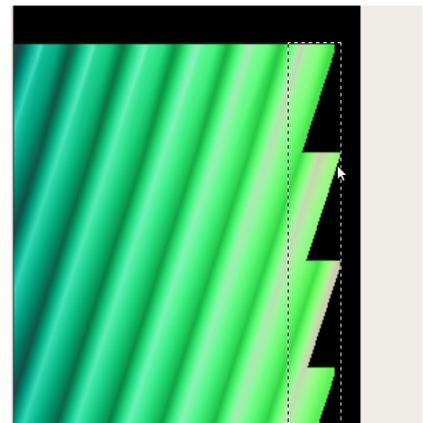
**21** Nel pannello delle texture in map input si seleziona UV e in map to si deselecta Col e si seleziona Nor inserendo un valore di nor di 11 (può essere anche variato ovviamente...).



**22** A questo punto se renderizziamo il cilindro dovrebbe risultare così. Se però lo giriamo ad un certo punto si vedranno una "cucitura". Se la colonna andrà visualizzata solo da un lato possiamo fermarci qui ma se in un'animazione essa si vedrà tutta bisogna effettuare un altro piccolo passaggio.



**23** Il problema sta in questo (Passo 19), come si vede la faccia più a destra del cilindro non è completamente coperta dalla texture. Per risolvere in Gimp si apre la mappa delle normali e si copia una piccola striscia della parte sinistra della mappa (Passo 23a) e la si incolla sulla parte destra facendola coincidere perfettamente (Passo 23b). Ora la salviamo, andiamo a vedere nell'Image editor e si dovrebbe aver ottenuto questo (Passo 23c). In questo modo il problema è completamente risolto e la cucitura non esisterà più.





# Simulare uccelli in volo

Le particelle per creare uno stormo di uccelli che volano

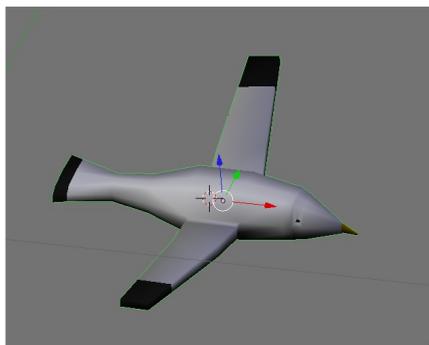
di Anfeo

Utilizziamo le particelle di Blender e la fisica Boids per simulare uno stormo di uccelli in volo.

Il nuovo, anche se non più tanto nuovo, sistema di particelle presente nelle ultime versioni di Blender (ora siamo alla 2.49), ci viene incontro in moltissimi casi in cui abbiamo grandi masse di oggetti da posizionare e spostare. Vediamo quindi come procedere per creare lo stormo.

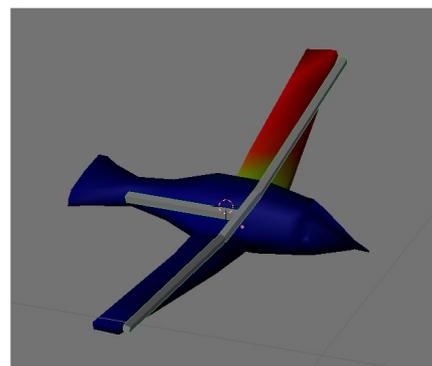
01

Prima di tutto abbiamo bisogno di un uccello, quindi modellatene uno del dettaglio che volete, ma ricordate che più saranno gli uccelli e maggiore saranno i poligoni in gioco. Quindi cercate una via di mezzo tra il dettaglio e quello che verrà effettivamente inquadrato



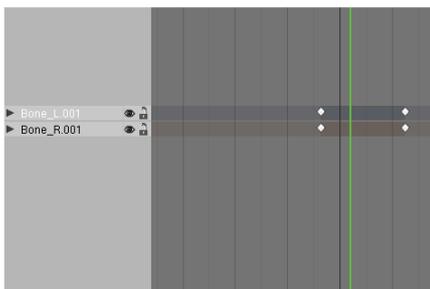
02

Creiamo una semplice armatura per far battere le ali, imparentiamola alla mesh e assicuriamoci che i pesi sulle ali siano come in figura, in modo che le ali possano muoversi quando l'osso viene ruotato. Potete eseguire anche Rig più complessi, il sistema di piumaggio di un'ala è davvero una grande sfida per un animatore.



**03** Passiamo nell'Action Editor e creiamo due keyframe per ala distanti 5 frame l'uno dall'altro, nel primo le ali saranno giù, mentre nel secondo le ali saranno su.

Proviamo l'animazione e vediamo subito che le ali battono una sola volta, vedremo più avanti come ripetere l'Action.



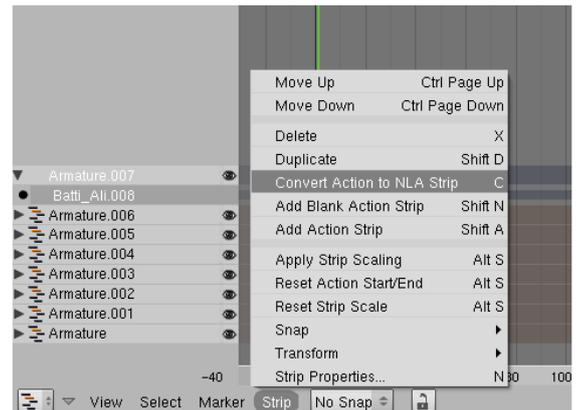
**04** Ora duplichiamo l'uccello e la sua armatura creando in tutto 8 copie. Queste copie ci serviranno a differenziare il battito delle ali dello stormo che se no risulterebbe tutto in sincrono e poco naturale. Il numero degli uccelli dipende da quanto vogliamo che siano diversificati i vari elementi dello stormo, volendo possiamo anche modificare leggermente le mesh o i colori degli uccelli, rendendoli più eterogenei, ma per ora concentriamoci solo sul movimento.



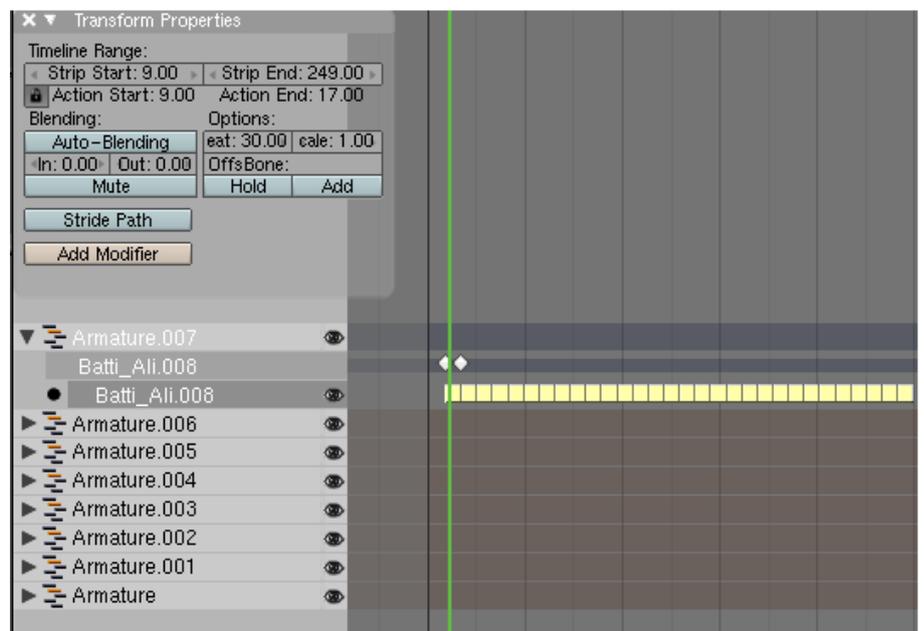
**05** Torniamo nell'Action Editor e selezioniamo un'armatura alla volta, assicuriamoci che ogni armatura abbia un action diversa dall'altra, in alternativa rendiamo univoca la action cliccando sul numero



**06** Ci occupiamo adesso di rendere il battito perpetuo per tutta la durata dell'animazione. Andiamo nell'NLA Editor e selezioniamo la prima Action, quindi convertiamola in una strip.

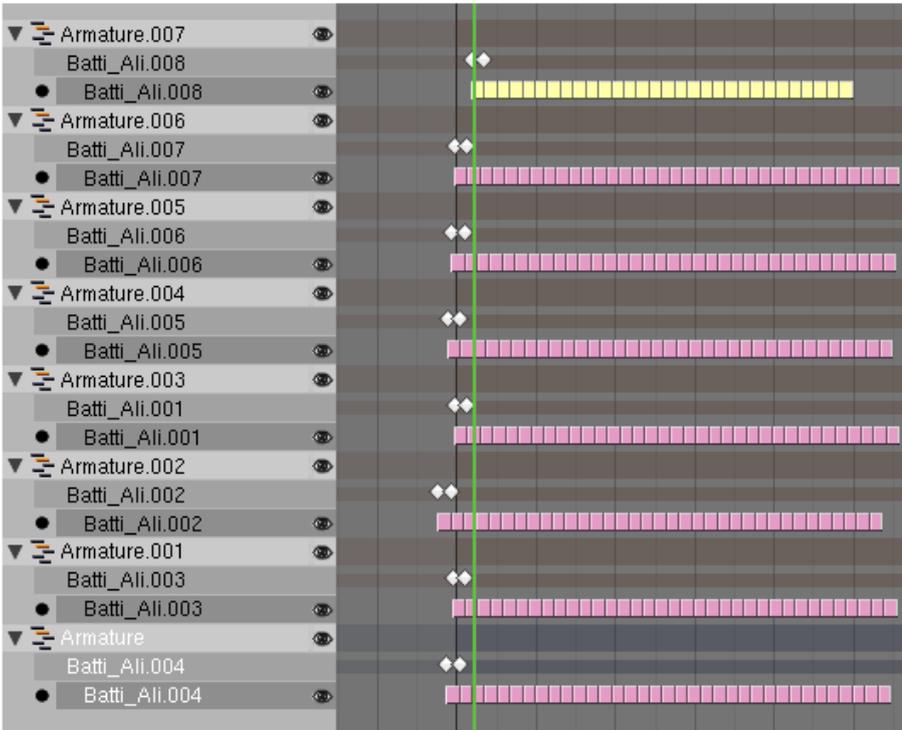


**07** Con la strip selezionata scegliamo Strip Properties dal menù Strip e aumentiamo il valore di Repeat a 30, in modo che la strip copra tutti i 250 frame di durata dell'animazione.



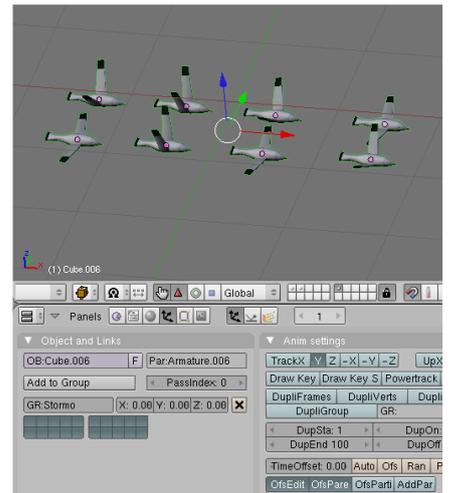
08

Facciamo lo stesso per le altre armature ripetendo gli step 6 e 7.



09

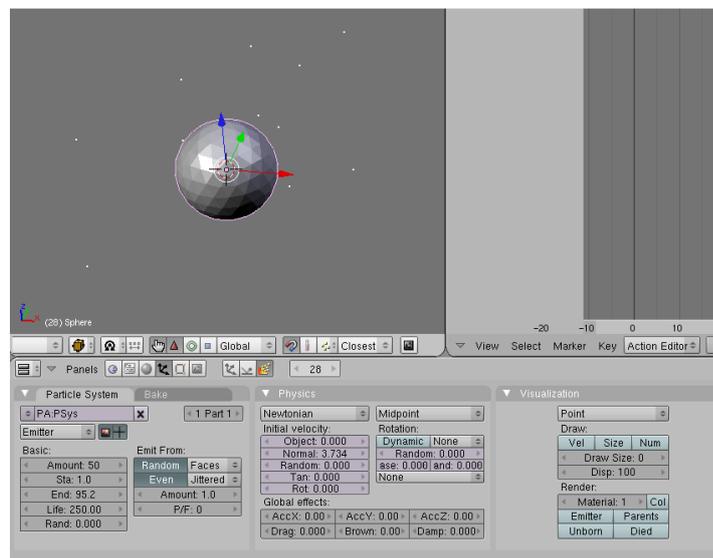
Torniamo alle nostre mesh di uccelli e selezioniamole una ad una tenendo premuto Shift, quindi premiamo CTRL+G e raggruppiamole. Poi chiamiamo il gruppo "Stormo" per comodità.



10

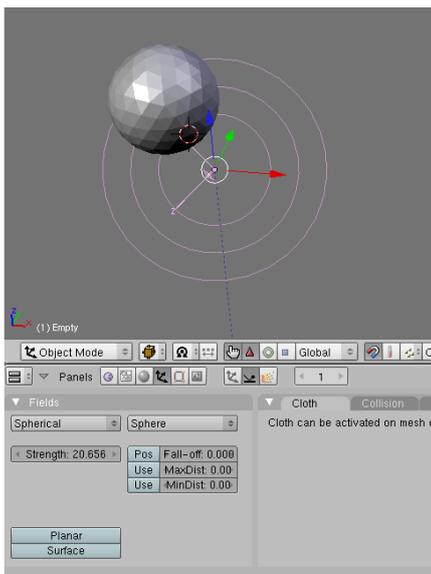
Mettiamo gli uccelli e le loro armature da parte in un livello separato e passiamo alle particelle. Inseriamo un Icosphere a livello 3, poi andiamo nel pannello Object (F7), selezioniamo "Particle Button" e creiamo un nuovo sistema di particelle facendo "add new". Settiamo quindi nella scheda "Particle System", i seguenti parametri:  
 Amount: 50 (numero di particelle/uccelli dello stormo)  
 Sta: 1.0 (inizio della generazione)  
 End: 95 (fine della generazione)  
 Life: 250 (la durata della visualizzazione, che deve essere uguale alla durata dell'animazione).

Mentre in Physics per ora settiamo solo Norma con un valore elevato, giusto per far volare via le particelle della sfera (che non verrà renderizzata alla fine). Lanciamo l'animazione per vedere l'andamento delle particelle.



**11** Tocca adesso direzionare le particelle, per farlo useremo la fisica Boids che simula appunto il movimento di massa. Nella fisica Boids le particelle sono attratte dai “fields”, quindi andremo a crearne uno in questo modo:

- aggiungiamo un empty
- andiamo nel pannello Object (F7) e scegliamo Physic Button vicino al pulsante delle particelle
- nella scheda Field scegliamo “Spherical” invece di “none” e settiamo come da figura:



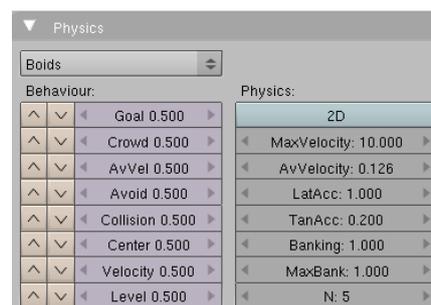
**12** Torniamo alla nostra Icosphere, selezioniamola e andiamo nel pannello del nostro sistema particellare, cambiamo quindi la fisica da Newtonian a Boids. Il pannello che ci appare (quello nella figura) è spazzante, ci sono un sacco di parametri da settare (per una spiegazione più approfondita vi rimando al wiki di Blender). Per ora prendiamo in considerazione la prima colonna a sinistra, questa rappresenta le

proprietà dello stormo, il loro ordine e il loro valore dipende dall'importanza che noi vogliamo attribuire ad ogni proprietà. Le prime della lista ha una priorità maggiore, man mano che si scende, l'importanza della proprietà viene meno e a volte addirittura ignorata, il tutto in base alla potenza del computer che si usa, il quale utilizzerà le risorse primarie per i primi della lista e le restanti per gli altri. Il modo migliore di imparare ad usare questo pannello è sperimentare una o due proprietà alla volta, settando tutte le altre a Zero.

Nel nostro caso ce ne servono soltanto due, andiamo per ordine di importanza:

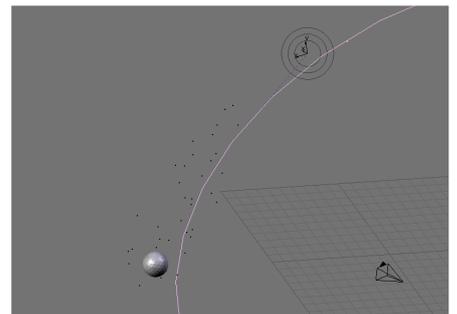
- Goal (le particelle seguono un oggetto con una forza, come un Field, maggiore è la forza, maggiore è l'attrazione)
- Crowd (questa crea l'effetto stormo, dando quella casualità tipica degli esseri viventi che si muovono in massa)

Cosa succede se invertiamo queste due proprietà? Provate a sperimentare, se mettiamo prima il Crowd, le particelle avranno ancora più casualità e molte si perderanno per strada. Capirete da soli che lavorando con tutti gli altri parametri si possono creare infinite combinazioni di movimenti.

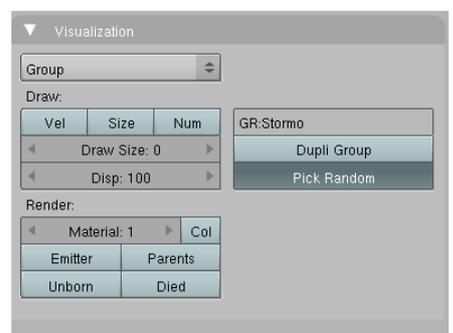


**13** Ora le nostre particelle seguono l'empty, non ci resta che farlo muovere lungo un percorso, inseriamo quindi una curva bezier e modifichiamola per farle seguire il percorso che vogliamo. Selezioniamo l'Empty, andiamo nel pannello “Object(F7) e questa volta clicchiamo sul primo pulsante sulla destra (Object Buttons), aggiungiamo una Constraint “Follow Path” e inseriamo il nome della curva bezier nel campo “OB:”.

Proviamo l'animazione e le particelle dovrebbero seguire l'empty che si muove.



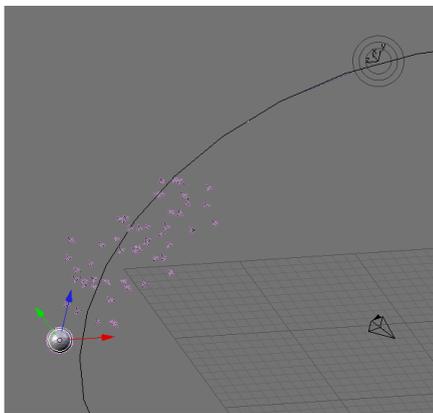
**14** Non ci resta che passare alla parte più succulenta di tutto il tutorial, selezioniamo quindi l'Icosphere e torniamo nel pannello del sistema di Particelle.



Nella scheda “Visualization” cambiamo il parametro “Point” con “Group”, scriviamo il nome del gruppo “Stormo” che abbiamo

creato al punto 9 nell'apposito campo e per finire clicchiamo su Pick Random (come in figura).

**15** Premiamo Alt+A per far partire l'animazione e vedremo partire degli uccelli dalla sfera e seguire l'empty per tutto il percorso (se gli uccelli non sbattono le ali, vuol dire che dovete rendere visibile il livello che contiene gli otto uccelli e le rispettive armature). Cosa abbiamo fatto risulta abbastanza chiaro, il sistema delle particelle pesca a caso nel gruppo Stormo e sostituisce ogni particella con un oggetto che trova nel gruppo. Se vogliamo aumentare il numero degli uccelli dello stormo, basta aumentare il numero delle particelle del sistema.



Potete visualizzare il video di esempio a questo indirizzo:  
<http://www.youtube.com/watch?v=GezjvL61q7Q>

### Conclusioni

Vi consiglio di approfondire il sistema di fisica Boids in quanto si tratta di un potente strumento per la gestione di masse di oggetti, non solo volanti in quanto attivando il pulsante 2d è possibile inserire un oggetto sul quale le particelle “cammineranno”, questo potrebbe essere utile per creare folle di persone o di insetti che camminano su terreni accidentati.



# Camera Tracking

Come fondere le nostre riprese dal vero con le animazioni 3d create in Blender

di Anfeo

Spieghiamo per bene cosa si intende per Camera Tracking. Tracciare il percorso di una videocamera, significa avere tradotto digitalmente il movimento che questa compie durante la ripresa di un video. Questo ci permette di poter sovrapporre un elemento creato digitalmente, che verrà ripreso da una camera digitale che emulerà la camera vera. In questo modo, l'oggetto sembrerà integrato perfettamente nella composizione finale.

Il Tracking del video girato può essere fatto in diversi modi e con diversi programmi. Principalmente lavorano tutti creando dei punti di riferimento fotogramma per fotogramma

utilizzando delle aree della ripresa come riferimento. Al variare della ripresa (e del fotogramma), si va a ricercare la stessa zona che c'era nel fotogramma precedente. Si avranno così le coordinate di movimento della videocamera. Il processo base è questo, ogni programma di Camera Tracking ha i suoi algoritmi che gli permettono un'approfondita analisi del video, per questo il processo di Tracking può a volte durare anche delle ore. Da questo si può capire che per facilitare il tracking bisogna seguire alcune regole.

Il programma che andremo ad usare, Voodoo Camera Tracking (<http://www.digilab.uni->

[hannover.de/download.html](http://hannover.de/download.html)), non ama molto le inquadrature libere, ma si possono ottenere dei buoni risultati tenendo a mente alcune cose:

- non fare riprese troppo lunghe;
- non fare movimenti bruschi;
- non fare zoommate;
- avere la videocamera su treppiede (tripod);
- avere punti di riferimento disposti in maniera casuale (evitare le scene piatte), per i Greenscreen di solito si usano delle palline da ping pong o degli adesivi di un colore che faccia contrasto con lo sfondo (che vengono tolti in post con Garbage Matte o Croma Key)
- avere video in HD
- avere ripresa da max 400

fotogrammi

– avere un oggetto di forma cubica (o parallelepipedo) di cui si conosce la distanza dalla videocamera e la sua dimensione reale.

N.B. Questi non sono proprio regole ferree, ma solo consigli per avere un Tracking più preciso.

Inoltre la qualità del video deve essere abbastanza buona e

soprattutto non devono esserci bruschi cambi di esposizione e di contrasti (Es. Passaggio da una stanza poco illuminata, all'esterno assolato).

Dopo questa premessa passiamo alla pratica vera e propria.

Ho eseguito dei test su di un video girato con una videocamera DV Pal. Si tratta di pochi secondi di una ripresa esterna fatta a mano libera.

Vedremo infatti che anche usando un video che non rispetta appieno le regole su citate, si possono ottenere buoni risultati.

NB: per motivi di spazio e dimensione dei file, troverete in allegato il file di trackin già pronto e la sequenza in formato AVI compresso.

NB: per la comprensione di questo tutorial è necessario avere una conoscenza media di Blender e di aver letto almeno i tutorial precedenti sui numeri 3,4 e 5 di BMI, relativi al Composite Node.

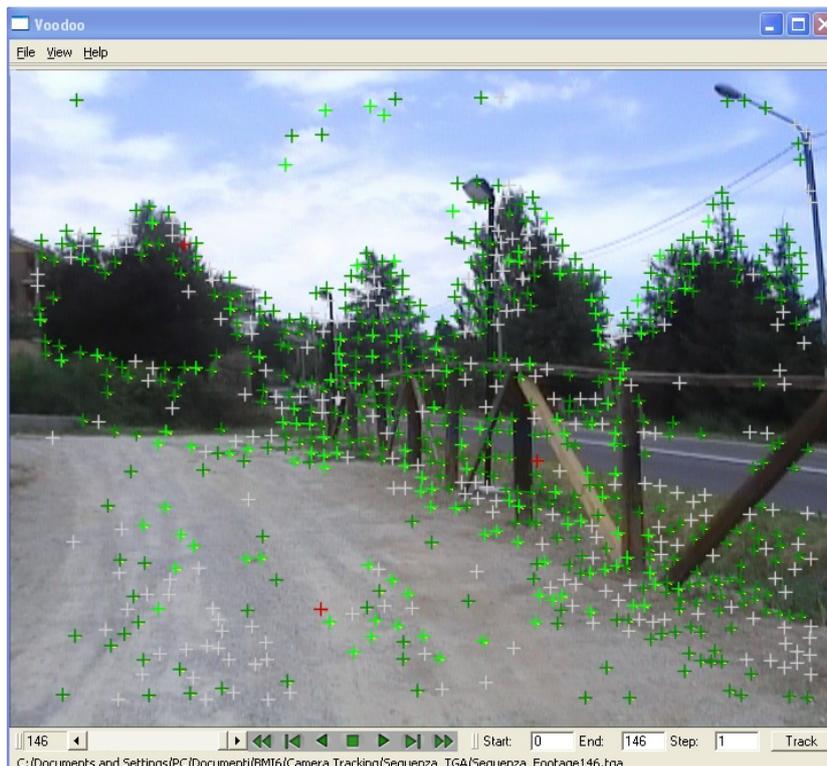
Potete trovare i file del progetto in allegato allo zip della rivista.

Per eventuali problemi e incomprensioni, non esitate a contattarmi a : [anfeo@libero.it](mailto:anfeo@libero.it).

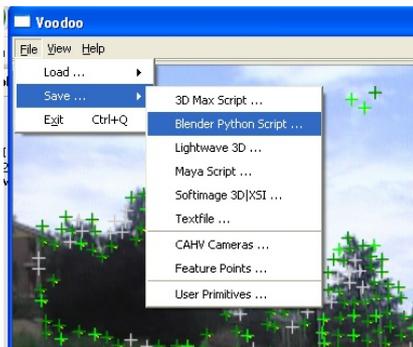
**01** Prima di tutto dobbiamo convertire il nostro video in formato “Sequenza di TGA”. Possiamo farlo tranquillamente da virtual dub, niente di complesso, apriamo il file e lo salviamo come sequenza di TGA.

**03** Lanciamo il tracking premendo sul pulsante Track e attendiamo. Alla fine del processo ci chiederà se vogliamo fare un ulteriore tracking più approfondito, noi accettiamo e aspettiamo che finisca. Il processo è abbastanza lento, quindi armatevi di pazienza e attendente

**02** Apriamo Voodoo e carichiamo la sequenza di TGA, selezioniamo movimento Free e lasciamo invariati gli altri parametri.



**04** Come potete vedere dall'immagine precedente, sono apparse delle croci che indicano tutti i punti tracking che il programma è riuscito a trovare. Alla fine del tracking salviamo il file Python scegliendo dal menù File->Save->Blender Python Script.

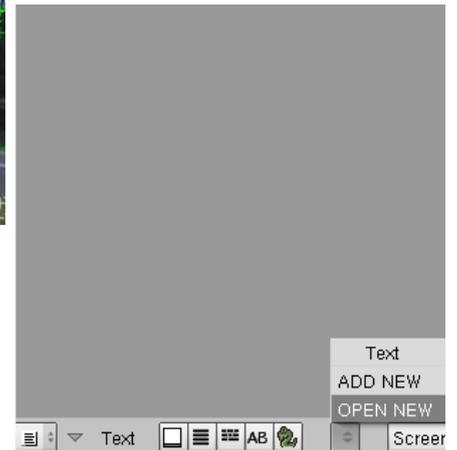


**05** Settiamo la finestra seguente scegliendo di esportare solo 500 punti di tracking, anche perché ci serviranno solo come riferimento nel file di Blender.

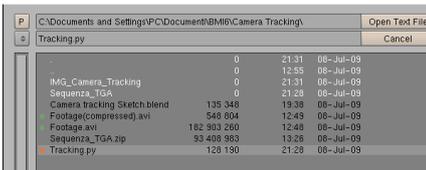


Da qui in poi è possibile usare direttamente il file di tracking allegato alla rivista in quanto è stato eseguito sulla sequenza di immagini non compressa e quindi risulta leggermente più preciso.

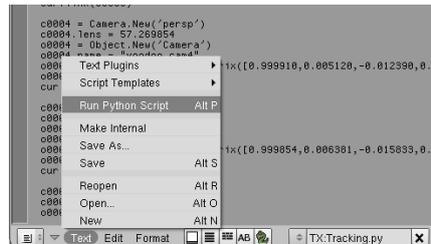
**06** Apriamo Blender, selezioniamo tutto quello presente nella finestra 3d e cancelliamolo (cubo, lamp e camera) e in una finestra a parte visualizziamo il Text Editor.



**07** Quindi carichiamo il file .py che contiene le informazioni del nostro tracking.

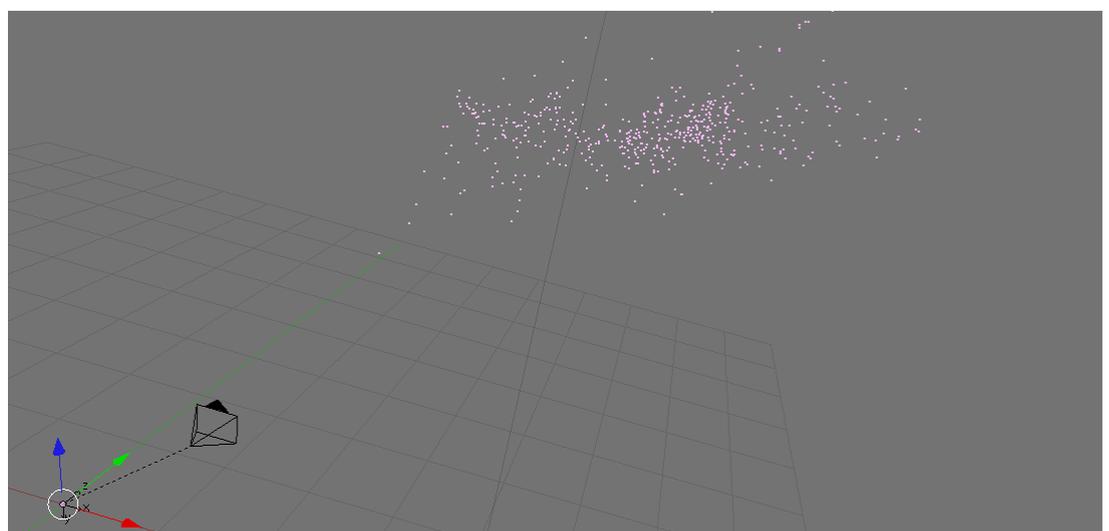


**08** Scegliamo dal menù "Text" il comando "Run Python Script"

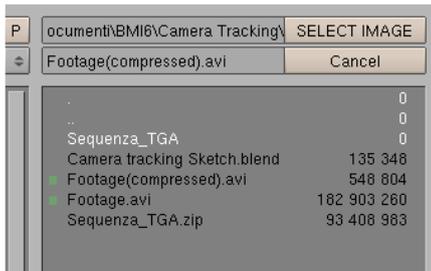


**09** Vediamo apparire magicamente una Camera nuova di zecca, imparenta ad una nuvola di puntini.

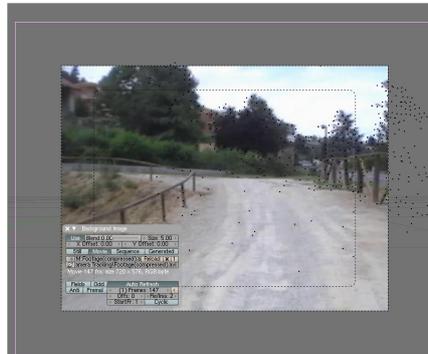
Proviamo a selezionarla e premere CTRL+Tn0 per selezionarla come camera di default, poi premiamo Tn0 per andare in vista Camera.



**10** Aggiungiamo la sequenza girata come sfondo scegliendo da View->Background Image e carichiamo il video Footage(compressed).Avi



**11** Impostiamo il pannello del Background Image come da figura.

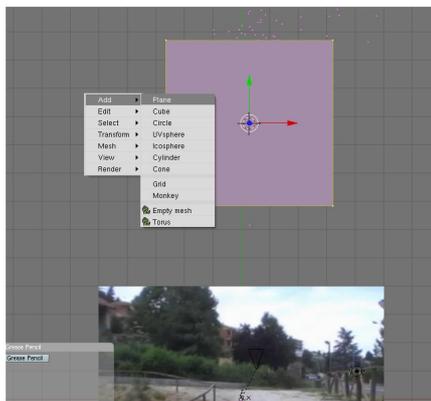


**12** Nel pannello di Rendering di Blender, impostiamo il numero di frame dell'animazione da 250 a 147.

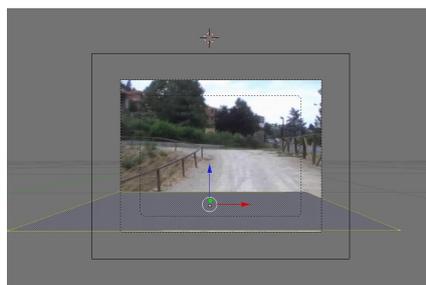
Proviamo l'animazione nella vista camera premendo Alt+A. Vedremo che i puntini cercano di seguire il video che cambia sullo sfondo. In realtà a muoversi non sono i puntini, ma la Camera importata tramite script.



**13** Spostiamoci in vista Top (Tn7) e inseriamo un Plane.



**14** Torniamo in vista Camera (Tn0), assicuriamoci di essere in Edit Mode e cerchiamo di coprire la piccola salita che si vede nel video con il nostro plane.

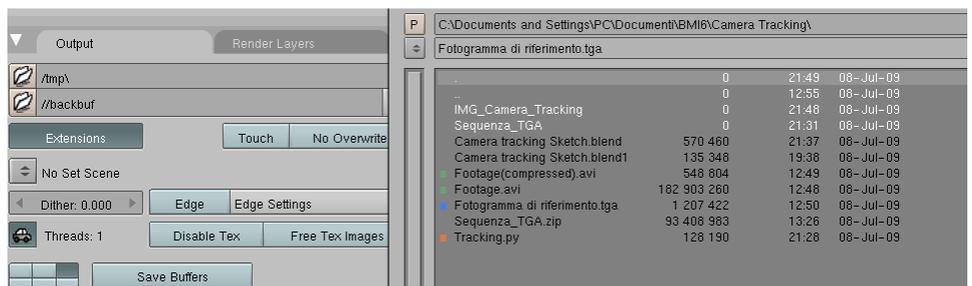


**15** In questa immagine ho aggiunto dei loop in corrispondenza del paletto per avere un punto di riferimento più preciso.



**16** Facciamo un veloce rendering di prova come spiegato nel tutorial sul compositing che trovate su BMI4. Carichiamo l'immagine del primo fotogramma (che potete trovare in allegato con il nome "Fotogramma di Riferimento.tga") andando nel pannello rendering e selezionando la sezione relativa al BackBuffer.

Ricordiamoci di dare la spunta al quadratino vicino al percorso.





**21** Questa è l'immagine risultante. Niente di particolare, ma si vede ancora troppo che è finta, manca una correzione del colore, del motion blur e un leggero rumore sui pixel, dovuto alla bassa qualità del sensore della videocamera.



**23** Ecco l'immagine finale, potete visualizzare il filmato su Youtube a questo indirizzo [http://www.youtube.com/watch?v=qfL6fMp\\_fac](http://www.youtube.com/watch?v=qfL6fMp_fac), mentre il file con tutta l'animazione, le texture e i materiali, potete trovarlo tra gli allegati dello zip della rivista.



**22** Ecco infatti dove vanno a finire gli altri svincoli del nodo "Render Layer", seguiamoli da sinistra a destra:

Il valore Z e Speed (che si attiva come render pass di Vec), vanno nel nodo di Vector Blur per simulare il motion blur.

L'uscita va a fondersi con un noise a colori (per maggiori info sulla texture utilizzata aprite il file in allegato). La fusione è leggera e di tipo "Color".

Quel gran casino di tentacoli, sono due nodi di Separate e Combine H, V, S, A (rispettivamente Hue, Value, Saturation e Alpha, per maggiori

**24** Come ultima chicca, vi voglio dare alcune dritte sul materiale usato per mappare il buco del video.

Nell'immagine potete prima di tutto in alto a sinistra il materiale Terriccio, che sarebbe quello dell'interno del buco, un semplice materiale con una texture.

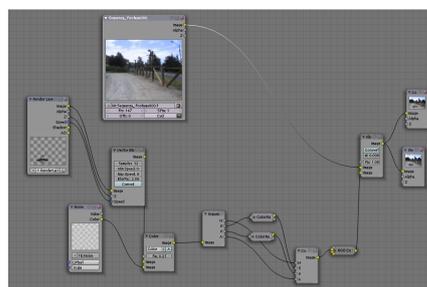
Questo materiale va a fondersi con il materiale OnlyShadow, ma in un modo molto particolare.

Il materiale OnlyShadow ci da in uscita solo le informazioni sulle ombre, ma queste informazioni sono date dal valore Alpha e non da quello Color. Quindi cosa accade? Siccome non possiamo applicare come Alpha la texture del buco, dobbiamo fondere i due valori con l'apposito nodo che appunto mixa i Valori, con una combinazione di ADD, in questo modo si preservano tutti i valori

dettagli vi consiglio un qualsiasi tutorial di fotoritocco).

Per agire su uno dei valori usiamo delle Ramp, che altro non fanno se non fornire dei valori di controllo, noi agiamo su Value (che sarebbe la luminosità) e la Saturazione, giusto per adattare il render all'immagine di sfondo.

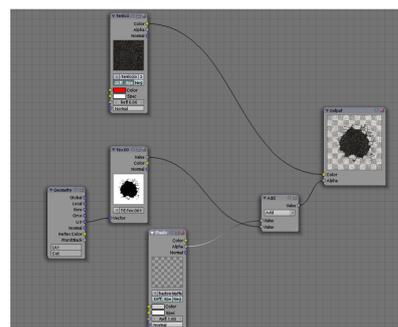
Un nodo finale di correzione colore con un RGB Curves.



che vogliamo.

Non ci resta che mettere sul nodo di uscita il valore Color del materiale Terriccio e come Alpha il valore del risultate mix dei valori di Alpha (il buco e le ombre). L'unico problema è che come valore di Ombre avremo non il classico nero, ma il colore del materiale Terriccio che si intravede da sotto.

A questo di può ovviare utilizzando il Composite Node e mixando le Ombre renderizzate come nel tutorial sul numero precedente di BMI.





# V-Ray Displacement

Displacement di alta qualità con gli algoritmi MTD (Micro-Triangle Displacement)

di Andrea Boggio "boggio95"

## Da Bump a Displacement

Una delle tecniche più utilizzate di sempre per dare l'illusione di profondità in un materiale è il Bump Mapping. Attraverso particolari calcoli vengono alterati gli angoli di incidenza dei raggi di luce per creare l'effetto di una superficie "ruvida", e non perfettamente liscia.

Ovviamente il Bump Mapping non agisce sulla mesh ma sulla luce.

Questo significa che la mesh non subisce deformazioni, ma è solamente la distribuzione della luce sulla texture che crea l'illusione: per questo problema la soluzione è il Displacement Mapping.

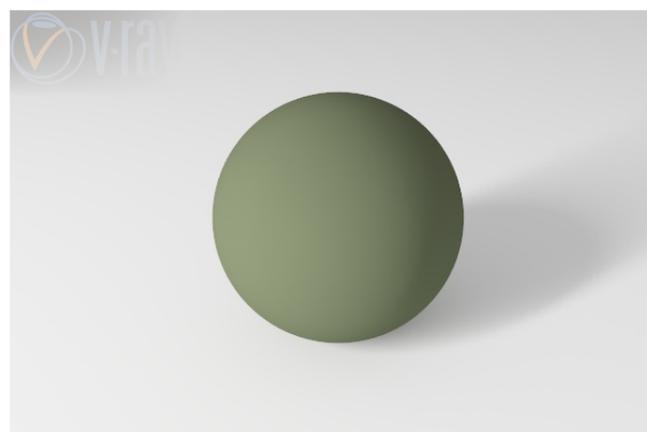
Questa tecnica agisce direttamente sull'oggetto, deformandolo a livello poligonale.

Anche Blender possiede un sistema di Displacement, ma non è preciso come altri (ad esempio V-Ray).

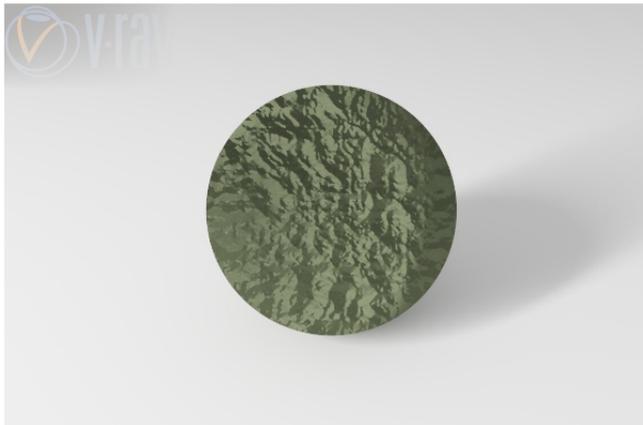
V-Ray utilizza degli algoritmi chiamati "Micro-Triangle Displacement" (MTD) che tassellano in triangoli sempre più piccoli la mesh, per poi applicare la mappa di displacement in base ai

valori di altezza per ogni pixel della mappa, tutto questo solo in fase di rendering. Con questo sistema si ottiene una precisione elevata senza "appesantire" la scena.

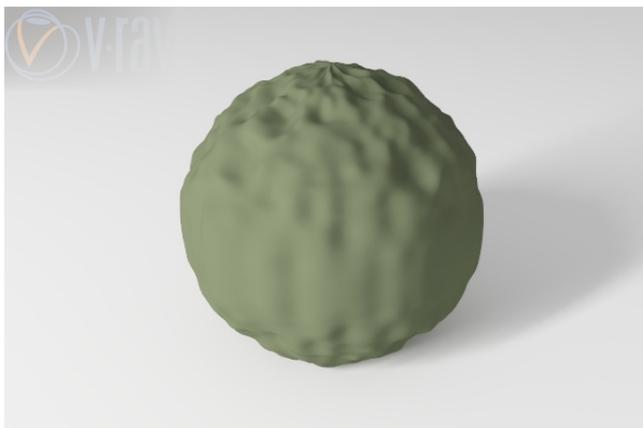
Ecco un confronto con valori di displacement simili e stessa texture:



**Sfera originale**



**Bump Mapping**



**Blender Displacement Mapping**



**V-Ray Micro-Triangle Displacement Mapping**

## Impostazioni per il Displacement

Vediamo come impostare Blender per il displacement di V-Ray.

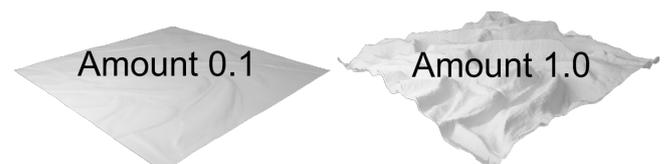
Andiamo nel pannello dei materiali e creiamo una nuova texture. Carichiamo la nostra Displacement Map e nel pannello "Map To" selezioniamo "Disp" (e deseleggiamo "Col"), come in figura:



Ora andiamo nel pannello Object dell'exporter (che ho spiegato nello scorso numero) e analizziamo le opzioni di Displacement:



- "Use Globals": se attivato, utilizza i valori comuni a tutti gli oggetti per il displace.
- "Disp": se attivato, utilizza il valore "Disp" del pannello "Map To" di Blender come "Amount" del Displace.
- "Amount": questo valore regola la profondità della mappa. Ecco un esempio:



- "Shift": è un valore di base di scostamento da aggiungere al valore applicato (Amount). A esempio: se "Amount" è impostato su 0.5 e "Shift" è impostato su 0.5, il valore del displacement finale sarà 1.0.

- Water Level: è il valore sull'asse z sotto la quale la mesh non viene renderizzata.

Ecco un esempio:



- Edge Length: è la lunghezza del lato di ogni micro-triangolo durante la fase di rendering.

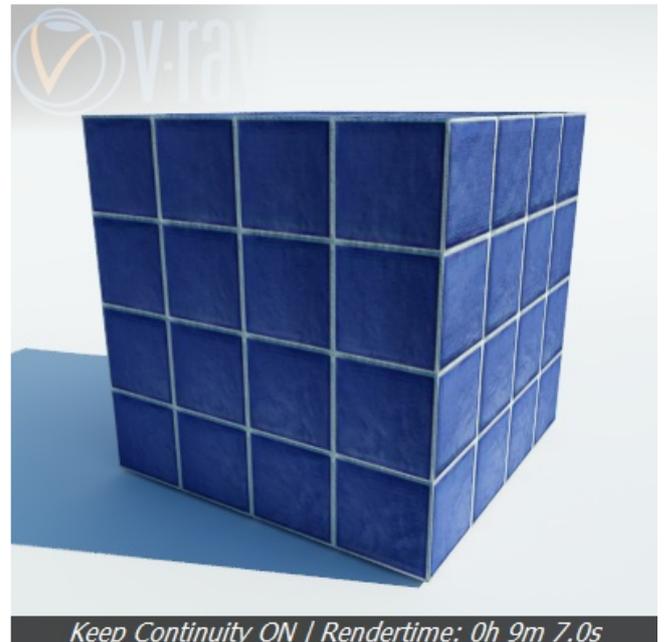
- Max subdivs: è il numero massimo di triangoli generati. Il numero massimo di suddivisione è dato dal valore immesso elevato al quadrato.

Ad esempio se il valore immesso è 256, i triangoli massimi generabili da V-Ray saranno 65.536, cioè 256x256.

Gli ultimi due pulsanti che troviamo sono "View Dependent", che attiva la suddivisione della mesh in base all'inquadratura, e "Keep Continuity", è una funzione che cerca di riparare ad un problema del displacement. Se la mesh è un cubo o un parallelepipedo, è possibile che tra le facce adiacenti vengano a crearsi degli spazi, un effetto simile allo strumento Bevel di Blender.

"Keep Continuity" invece evita queste imperfezioni ottimizzando l'effetto del displacement sull'oggetto, senza intaccare le prestazioni.

Ecco un esempio:



## Conclusioni

V-Ray si aggiudica ancora una volta un punto a suo favore nella "guerra" dei motori di rendering commerciali. Ovviamente le funzioni disponibili nel corrente exporter per Blender, non sono complete (come ad esempio nell'implementazione con 3ds Max), ma nei prossimi sviluppi dello script queste lacune verranno colmate.

Il displacement di V-Ray è sicuramente un'ottimo strumento, soprattutto facile da usare, e che compensa, nel caso di Blender, la mancanza di un sistema di Displacement preciso ed efficace.



# Immagine "Macro"

Contest per scegliere la prossima copertina:  
Macro: i grandi dettagli delle piccole cose

Come per ogni numero di Blender Magazine Italia, anche per il prossimo, l'immagine di copertina verrà scelta tramite un contest.

Il tema per il numero 7 di BMI è "Macro".

La macrofotografia è una tecnica fotografica che permette di ottenere tramite forti rapporti di ingrandimento immagini di soggetti molto piccoli.

Se non la conoscete o volete approfondire l'argomento, potete leggere un'esauriente spiegazione su Wikipedia:

<http://it.wikipedia.org/wiki/Macrofotografia>

e vedere alcuni esempi di foto macro su google:

<http://images.google.it/images?hl=it&um=1&sa=1&q=macrofotografia&btnG=Cerca+immagini&aq=0&oq=macrofot>

Il regolamento è sempre lo stesso:

- Le immagini devono essere eseguite principalmente con Blender.
- Si possono eseguire anche più lavori, ma ricordate che conta la qualità non la quantità.
- Sono permessi motori di rendering esterni.

- Sono permessi altri programmi esterni ma solo per ritocchi di postproduzione ed a patto che siano Open.

- Le immagini devono avere queste dimensioni: 1024\*1448 px o superiore, basta rispettare la proporzione A4.

- Le immagini possono essere postate come Wip su Blender.it (sezione immagini), Kino3d (sezione open lab) o Blender-Tutorial, con la dicitura [BMI7] prima del titolo.

- I lavori finiti devono essere mandati all'email [anfeo@libero.it](mailto:anfeo@libero.it) **entro il 29/11/2009**

- Ricordate di specificare il titolo dell'immagine e il nome che volete usare come firma (nick e/o nome vero).

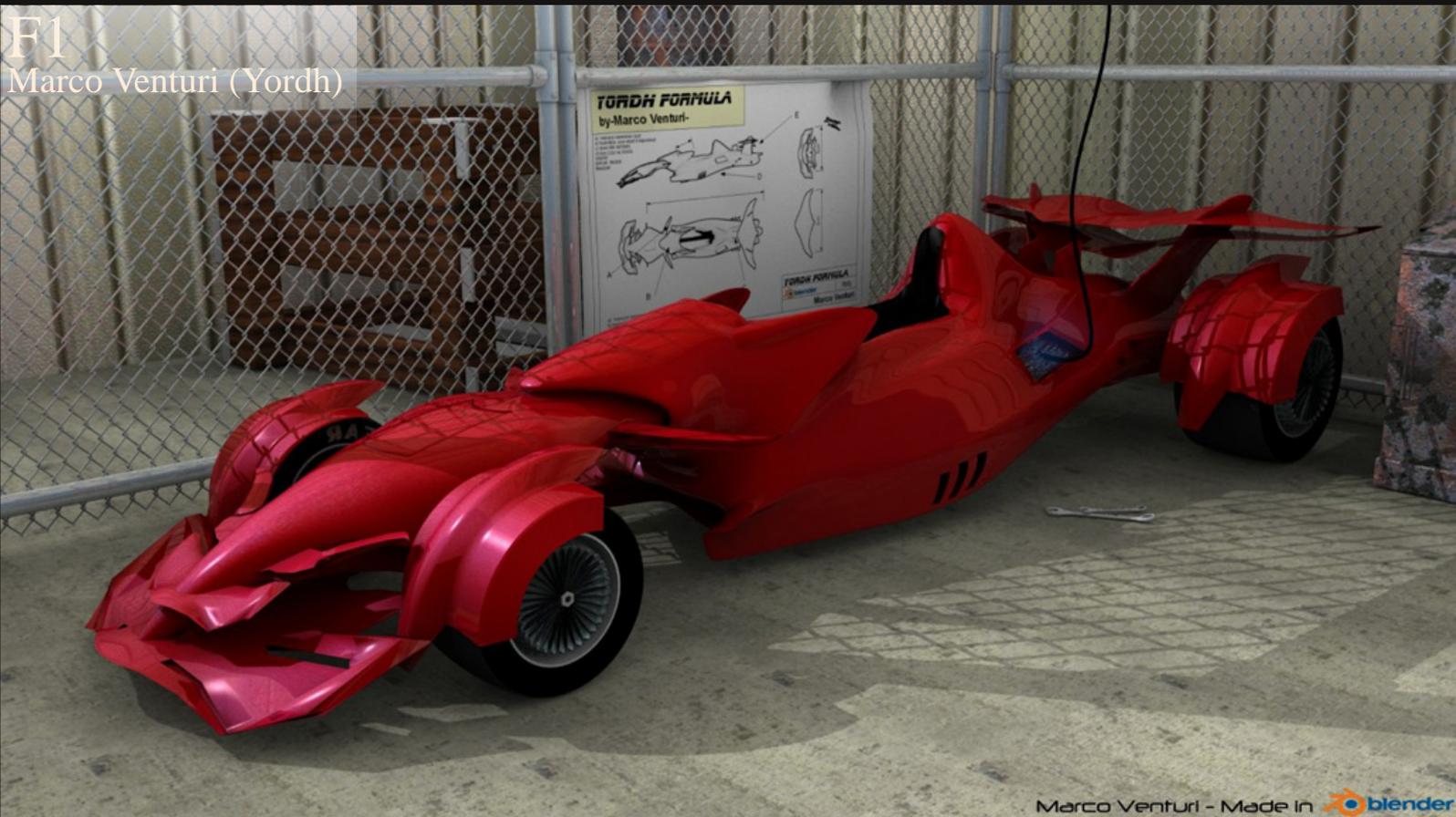
- Il vincitore avrà l'onore di vedere pubblicata la sua immagine sulla copertina di Blender Magazine Italia.

- Il vincitore potrà scrivere un piccolo Making Of del suo lavoro che sarà pubblicato sulla stessa rivista elettronica. L'articolo non è obbligatorio.

Al termine della data di consegna, verrà aperto un sondaggio affinché, chi vuole, possa votare, tra le immagini partecipanti, quella che vorrebbe vedere sul prossimo numero di Blender Magazine Italia.

# Gallery

F1  
Marco Venturi (Yordh)



Marco Venturi - Made in  blender

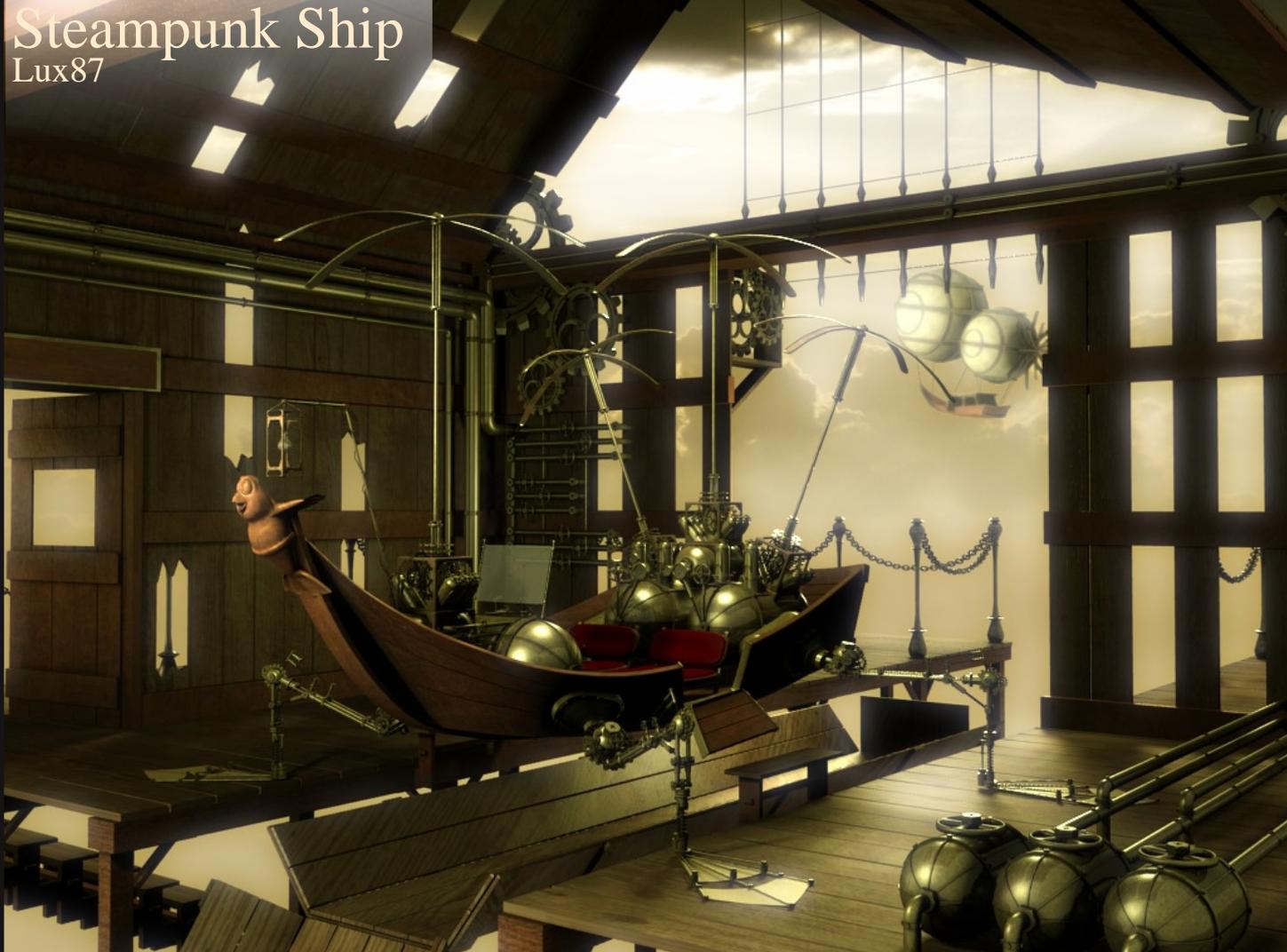


**DEVOTO 3D**

Iroplano  
Devoto

# Steampunk Ship

Lux87



# Samba Bus

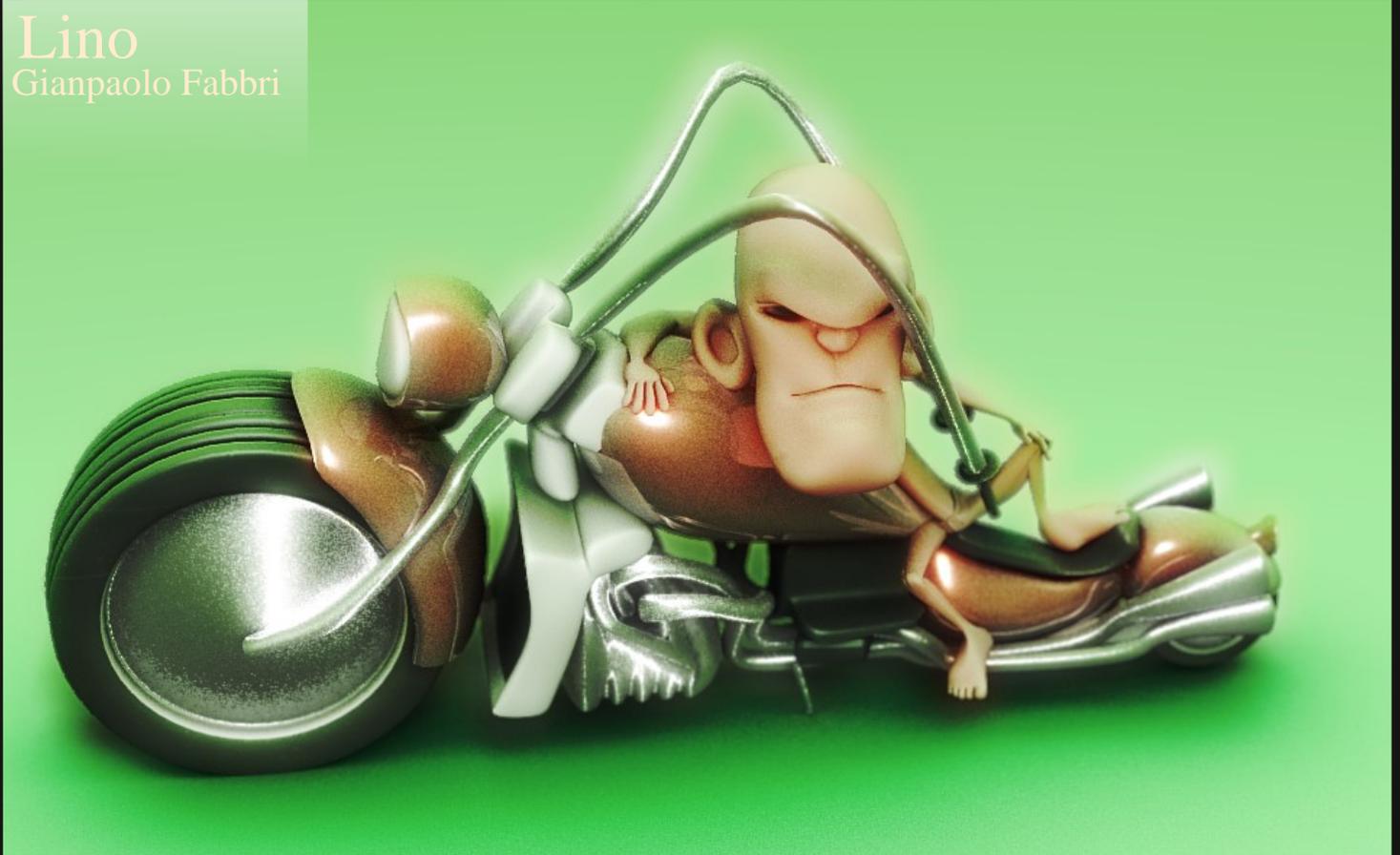
Salvatore Provvidenza (SmokeArch)



Kim  
Gianpaolo Fabbri



Lino  
Gianpaolo Fabbri



# Birra

Marco Alici (MarcoA)



# Foglie

Sandro Pizziolo



Ulivi al tramonto  
Albergo Giuseppe  
([www.blukiwi.org](http://www.blukiwi.org))



Maronide  
(Vecchia Ebra)  
Fabio Gagliardi  
(Liberate Policleto)

