

Numero 5 - Maggio 2009

BLENDER

Magazine Italia

LINEAR WORKFLOW

Introduzione al flusso di lavoro lineare

V-RAY E BLENDER

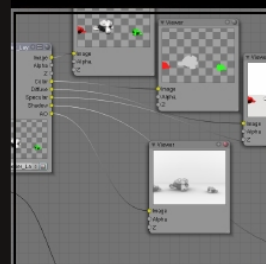
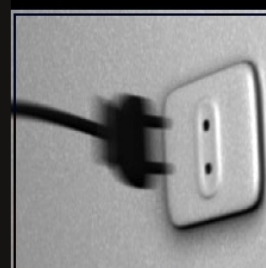
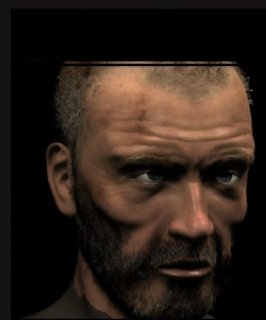
L'incontro fra commerciale e opensource

MAKING OF...

Scena Still Life: settare Blender
come uno studio fotografico



News	4
Appunti sul linear workflow	7
Le Shape Key Curve	11
V-Ray e Blender	14
3d Compositing	18
Creare una scena Still Life	23
Algoritmi di rendering	30
Ufo a mezzanotte	35
Contest copertina BMI6	43
Gallery	44



In questo numero, gli appassionati di Blender possono leggere le ultime news sul loro software preferito.

A seguire vi è un interessante articolo di Fernando Luceri che spiega il flusso di lavoro lineare, tecnica di fondamentale importanza per realizzare opere di qualità professionale, in quanto fornisce una metodologia che consente di ottenere render con colori più coerenti con le texture applicate al modello 3d.

Con un esempio pratico dettagliatamente illustrato, Dawidh insegna ai lettori l'uso delle Shape Key Curve, scaturito da una sua ricerca del modo per far deformare un percorso.

Nell'articolo successivo, di Andrea Boggio, viene detto come è possibile effettuare con Blender dei rendering utilizzando V-Ray, uno dei migliori motori di render biased. Ci si avvale di uno script realizzato dal russo Andrey Izrantsev. L'autore dell'articolo spiega come creare scene e settaggi di base per ottenere render di interni, esterni o in studio.

Anche in questo numero Anfeo approfondisce come inserire elementi 3d in ambienti reali usando i nodi. Con 20 passaggi guida il lettore alla scoperta di utili tecniche di compositing.

Rik, la cui firma appare per la prima volta su Blender Magazine Italia, dà il suo prezioso contributo per spiegare, in 28 passaggi, in che modo settare Blender come uno studio fotografico.

E non manca, in questo numero, uno degli interessanti articoli tecnici a cura di Fulvio Satta (YotaVGA), questa volta sugli algoritmi utilizzati comunemente nel rendering.

A seguire, vi è il resoconto di Fulvio Fontana dell'esperienza di un progetto didattico nel quale viene largamente utilizzato Blender.

Nella Gallery, infine, è possibile prendere visioni di immagini realizzate con Blender, negli ultimi tre mesi, da utenti italiani.

Blender Magazine Italia
numero 5
anno 2009

Responsabili:

Luca Pinciani (Sinistar)
Alfonso Annarumma (Anfeo)

Collaboratori:

Gikkio

Grafica:

Davide_G

Siti di riferimento:

www.blender.it
www.kino3d
www.blender.org

Software utilizzati:

Blender 2.45
PdfTk
The Gimp 2.4
Openoffice
Scribus 1.3.3



In Copertina:
"La caduta del cavaliere oscuro"
di Seregost



© Tutte le immagini appartengono ai legittimi proprietari

Blender: le news

Ultime news dal mondo Blender

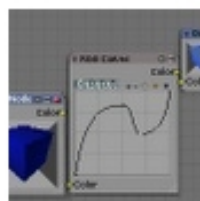
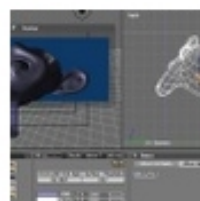
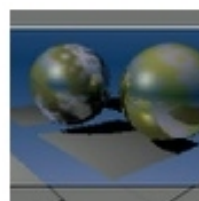
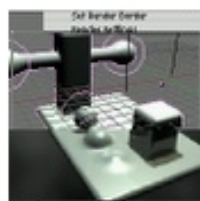
di Luca Pinciani

Rilasciato BlenderPocket versione 2.48a

BlenderPocket è una versione di Blender per i dispositivi mobili (cellulari, palmari, ecc.) che hanno un touch screen, poca memoria RAM, un processore lento e che mancano della graphic unit. BlenderPocket attualmente funziona correttamente su tutti i dispositivi che hanno Windows Mobile 2003 o sistemi operativi più recenti, ma dovrebbe comunque essere relativamente facile farlo funzionare su ogni altro dispositivo.

Ad inizio marzo 2009, BlenderPocket è stato aggiornato, passando dalla versione 2.42 alla 2.48. In questo modo, nella nuova versione si possono trovare le caratteristiche di Blender 2.48, ad eccezione dello Sculpt Mode che ancora non funziona.

BlenderPocket è realizzato da Salvatore Russo ed il suo sito web è <http://russose.free.fr/BlenderPocket>. Dal sito è possibile scaricare BlenderPocket, vedere alcuni screenshot o lasciare un messaggio all'autore.



Vincitori del contest "Yo Frankie! Level Design"

Pochi mesi fa, era stato indetto un concorso per la realizzazione di un livello aggiuntivo di Yo Frankie!, il videogioco sviluppato dall'Apricot Team. A metà marzo sono stati eletti i vincitori. I livelli realizzati possono essere scaricati e provati in Yo Frankie!

Il primo premio (250 euro) è andato Moraes Junior per aver creato un bellissimo livello con una catapulte, cascate di lava ed altre sorprese.



Il secondo premio, pari a 150 euro, è stato assegnato a Bruno Franci per il livello dal titolo "Faro", ambientato su un'isola.



Infine, il terzo posto è stato per Bradley Wabbersen per il suo livello non particolarmente grande ma realizzato con molta cura.



Tutti e tre i livelli possono essere scaricati e installati in Yo Frankie!

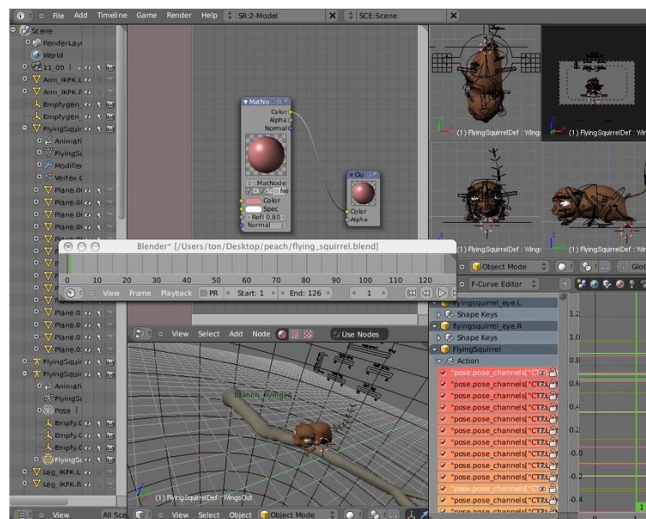
Per maggiori informazioni, visitare il sito <http://www.yofrankie.org/competition-winners-announced/>

Blender 2.50: nuovi aggiornamenti

Per tutto i curiosi che non hanno il tempo di seguire con costanza gli aggiornamenti che porteranno all'attesissima versione 2.50 di Blender, all'indirizzo <http://www.blender.org/dev/development/current-projects/blender-25-project/> è stato redatta una roadmap per documentare le successive fasi del progetto.

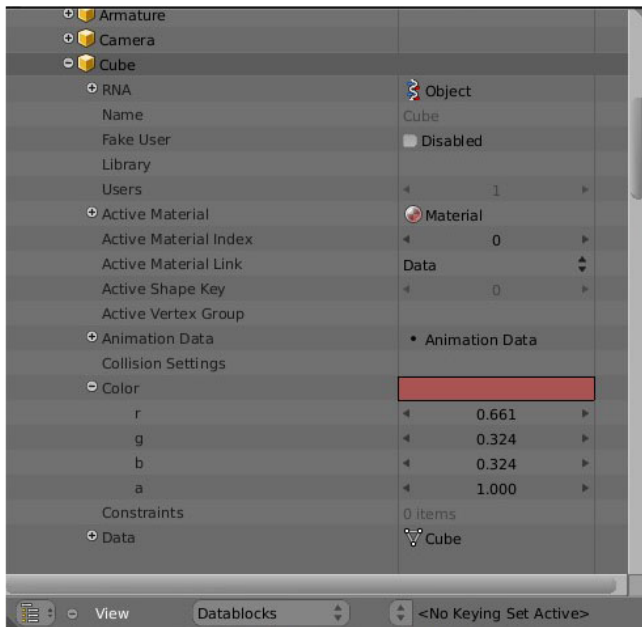
Attualmente alcuni di queste fasi sono quasi ultimate, mentre altri cambiamenti sono ancora da intraprendere.

Molte saranno le novità, eccone alcune:



Riscrittura del Window Manager che attualmente si basa su librerie grafiche vecchie di 10 anni.

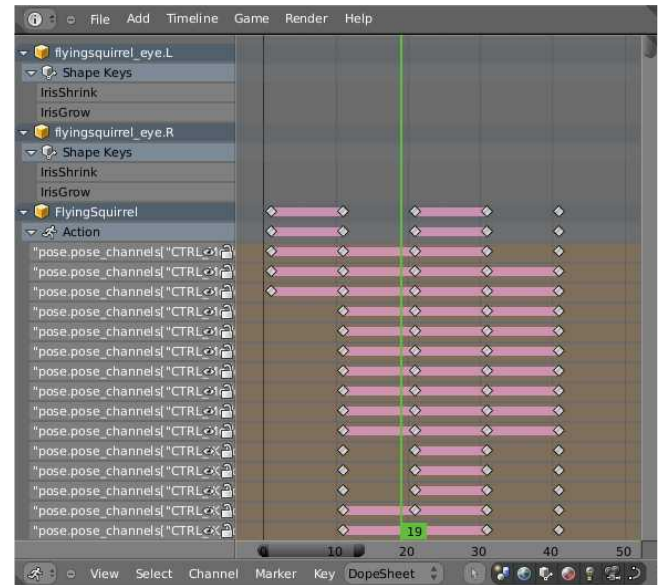
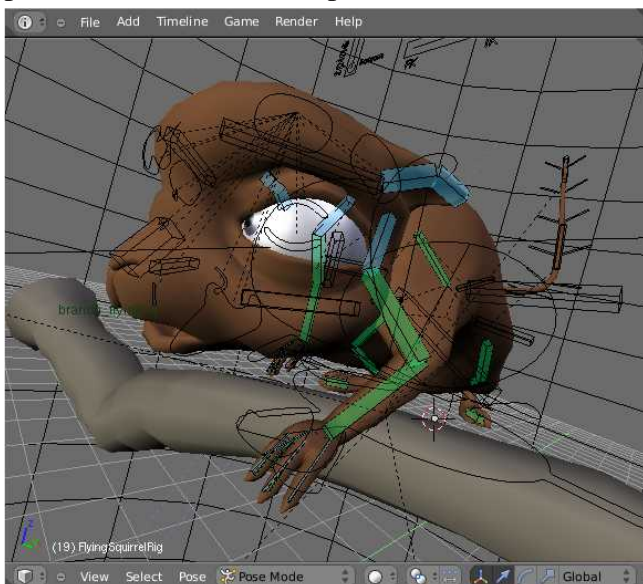
Rifacimento dei sistemi di gestione di eventi, strumenti e data.



Aggiunta della possibilità di personalizzare le hotkeys e nuovo design per l'interfaccia utente.

Nuovo File Browser, con segnalibri e preview. Potenziamento nell'inserimento e nella gestione di testo.

Miglioramento del sistema di animazione ed estensione dell'Action Editor per il controllo di più azioni allo stesso tempo.



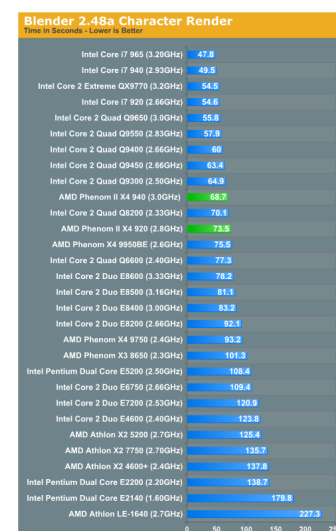
Miglioramento dello Sculpt grazie al progetto del Google Summer of Code. Infine migliorie al Game Engine.

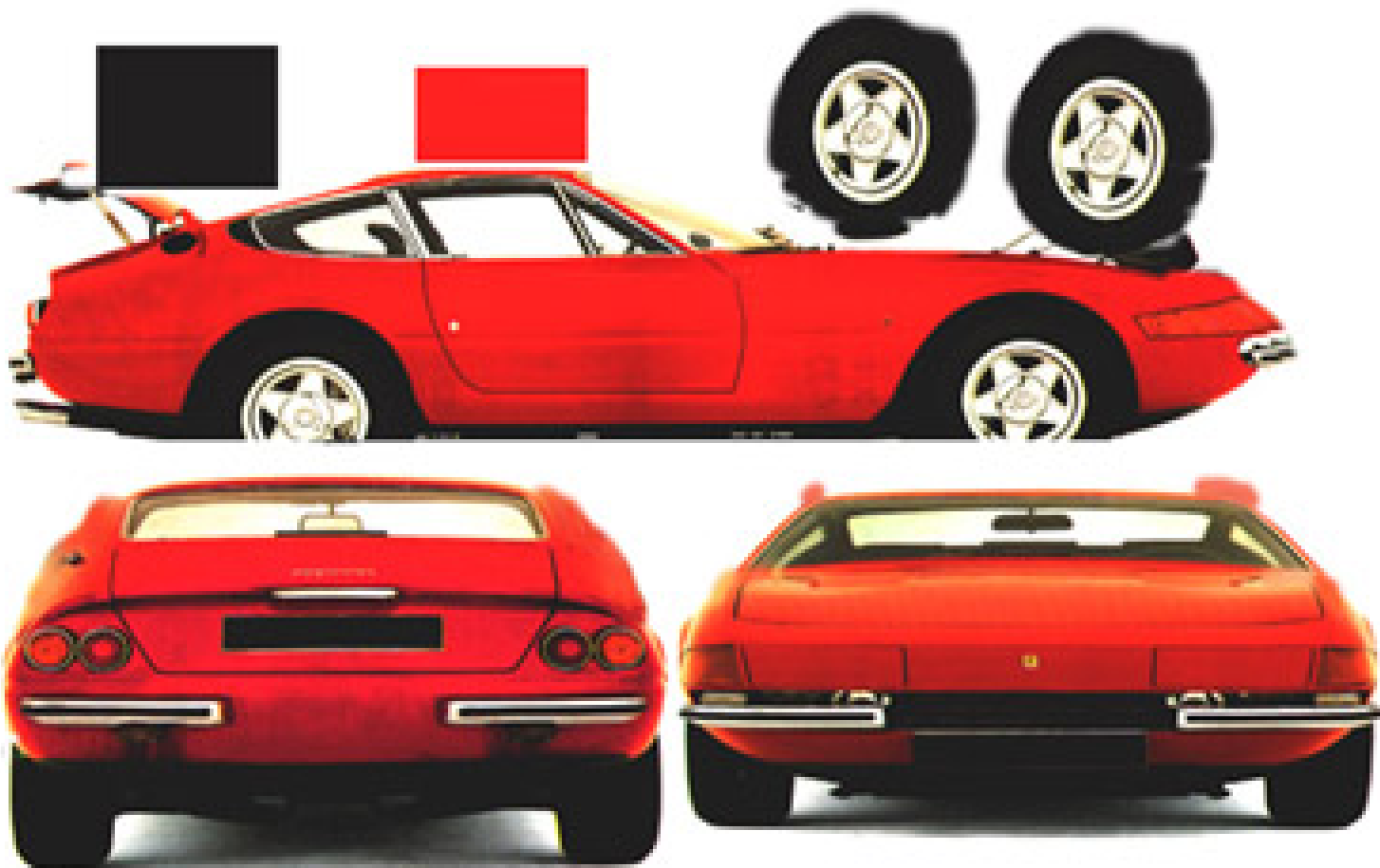
Inoltre, è assicurata la retrocompatibilità, ovvero progetti e file creati con versioni precedenti di Blender si potranno aprire senza problemi con la versione 2.50.

Test di velocità di CPU Intel e AMD

AnandTech ha messo a disposizione i risultati di alcuni test di velocità realizzati calcolando il tempo per renderizzare una scena di test con Blender 2.48a utilizzando diverse CPU, sia Intel che AMD. Il grafico riepilogativo si può trovare

all'indirizzo
<http://anandtech.com/cpuchipsets/showdoc.aspx?i=3492&p=16>

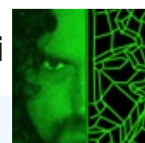




Appunti sul linear workflow

Veloce introduzione al misterioso mondo del Flusso di lavoro lineare.

Di Fernando (Gillan) Luceri



Read Me First

Per valutare correttamente la resa delle immagini prodotte è necessario calibrare correttamente il vostro monitor.

<http://www.calibrize.com/> (software gratuito per la calibrazione del monitor)

<http://tft.vanity.dk/> (online monitor test)

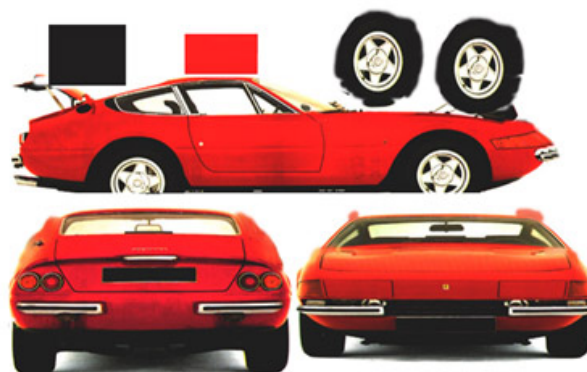
http://en.wikipedia.org/wiki/Linux_color_management (per gli utenti linux)

<http://www.normankoren.com/makingfineprints1A.html> (test di calibrazione specifici per la gamma)

Quando ho scoperto che i professionisti della grafica 3d utilizzavano un *linear workflow*, mi sono detto: “ Se voglio diventare un professionista devo assolutamente sapere di cosa stanno parlando”. Purtroppo però il primo approccio invece di essere chiarificatore è stato disorientante.

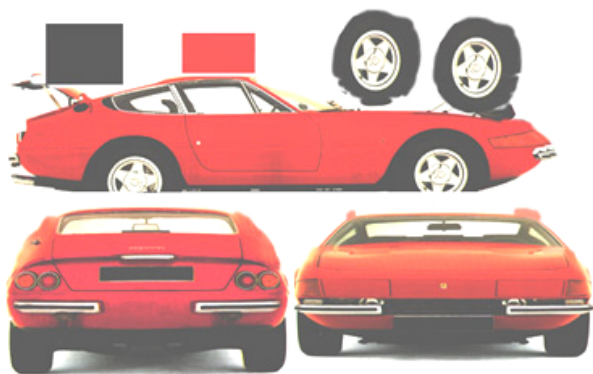
Si andava da uno script che faceva tutto lui ad una filosofia di comportamento, per cui mi scuso con i tecnici e i puristi ma questo mio articolo non vuole essere né esauriente né preciso (per chi ha voglia di approfondire vi rimando agli ottimi link in coda all'articolo) ma solo accompagnarvi a sperimentare nella pratica il linear workflow.

Partiamo da una texture, per esempio questa qui:



Avete faticato tanto in Gimp per dare alla carrozzeria quel bel rosso fiammante ed ai

pneumatici un bel grigio scurissimo. Cosa pensereste se vi dicessi che Blender la vede così?



Tutto da rifare eh? Ma come stanno davvero le cose? Il sensore di una macchina fotografica digitale rileva la quantità di luce in un dato punto e ne registra il valore numerico (in realtà nella maggior parte dei casi lo filtra attraverso un algoritmo, ma di questo parleremo dopo); se rileva una quantità di luce doppia, registra un valore numerico doppio. Questo è un metodo “lineare” di registrare il valore.

Il problema è che per un occhio umano il rapporto fra le due quantità di luce è decisamente diverso! Più del doppio se le due tonalità sono “scure”, meno del doppio se sono “chiare”. Questa errata percezione è data dal fatto che l'occhio è in grado di rilevare con più precisione le differenze fra le tonalità scure rispetto alle differenze fra quelle chiare. Per questo sono stati creati alcuni standard colore che danno molte più informazioni nella parte “scura” della palette rispetto a quella chiara. Lo standard più diffuso è lo **sRGB** (standard RGB) con questo o con suoi derivati sono codificate il 90% delle immagini digitali in circolazione. Le macchine fotografiche digitali convertono normalmente le immagini in questo standard prima di salvare il file, a meno che non si voglia mantenerlo nel formato **Raw** (grezzo) che salva i dati così come percepiti dal sensore.

Ma Blender non si comporta così, quando noi inseriamo una texture, lui la “vede” in modo lineare anche se in realtà molto probabilmente

quella immagine è stata codificata (e processata) nello spazio colore sRGB.

Questo nei nostri rendering potrebbe provocare un veloce decadimento della parte medio scura dell'immagine ed una desaturazione del colore all'aumentare della luminosità. Spesso vengono utilizzati dei trucchi per rimediare a questo: postproduzione, luci con decadimento lineare invece che quadratico, etc.

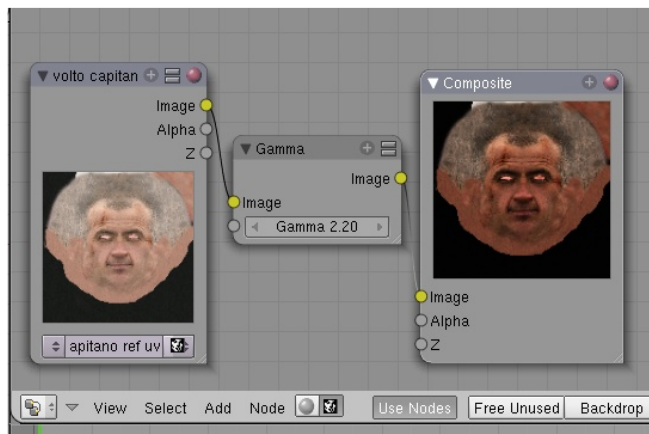
Ma il miglior risultato si otterrebbe se dessimo in pasto a Blender texture e colori nello spazio colore lineare e, solo alla fine, una volta renderizzata l'immagine, applicassimo un filtro (**gamma**) per portare l'immagine nello spazio colore sRGB a noi così familiare.

Facciamo un veloce esperimento che vi permetterà di saggiare gli effetti di questa metodologia. Partiamo da una nostra vecchia scena in blender. Io parto da questa:



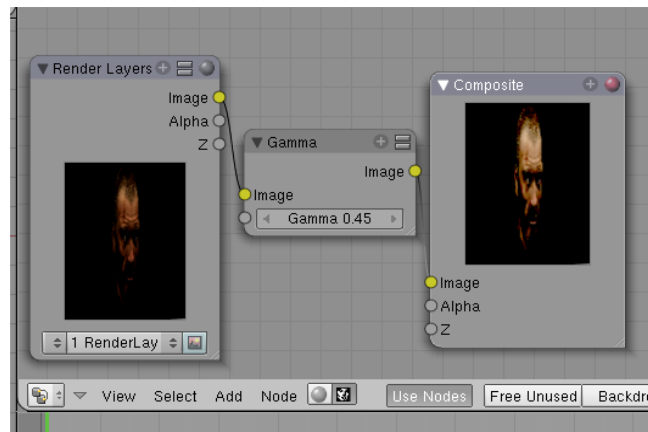
Un paio di considerazioni. Ho scelto un vecchio progetto perché volevo capire quanto potevo migliorare i miei rendering adottando il linear workflow, però c'è da ammettere che questo progetto era ottimizzato per il vecchio metodo. Luci, texture, materiali: tutto calibrato per dare il miglior risultato. La filosofia del linear workflow

pretenderebbe invece che, fin dall'inizio, si ragionasse in modo "lineare". Terremo conto di questo quando faremo i confronti finali.



La prima cosa da fare è portare le texture ed i colori dei materiali da sRGB a lineare. Questo procedimento normalmente viene affidato ad un programma di fotoritocco, ma purtroppo Gimp non è il massimo per questo perché lavorando a **8 bit** ci costringerebbe a perdere informazioni nella conversione (gli utenti Linux possono usare **digikam**). Il modo migliore che mi viene in mente è quello di usare **Blender** stesso. Pongo la texture come **input** nel **Node Compositor**, la modifico grazie al **nodo gamma** impostato su **2.2** (questo è il valore consigliato dai vari esperti nei siti che vi ho linkato) e come output una immagine con la stessa risoluzione in formato **.exr** (nessuna perdita di informazioni ma attenzione alla dimensione che aumenta enormemente).

Un sistema più rapido è quello di usare l'RGB curve direttamente nel node material di Blender. In questo caso però non abbiamo il valore numerico di riferimento ed il processo è meno preciso. In ogni caso sostituisco le vecchie texture con le nuove e regolo di conseguenza i colori dei materiali (purtroppo il preview diventa poco rappresentativo del risultato finale). Nel compositor correggo l'immagine grazie al nodo gamma (impostato sull'inverso del valore precedente in questo caso può andare bene **0.45**).



Il risultato, pur essendo interessante, mi sembra sbiadito. Questo, come dicevo prima, può essere dovuto al fatto che il linear workflow pretende un intervento in tutte le fasi della produzione e non, come in questo caso, a lavoro compiuto. Tento, così, di mediare le cose provando ad abbassare la gamma di correzione delle texture (min 1,5 max 1,8) e di conseguenza ad alzare quella di correzione dell'immagine finale (min 0,70 max 0.55). Il risultato dopo diverse prove mi piace davvero e mi convince dell'utilità del flusso di lavoro lineare.



Prima di tutto il decadimento della luce è molto più naturale, sono spariti gli hotspot ed i rapidi passaggi al nero, l'immagine nel complesso

appare molto più bilanciata e, soprattutto, le texture sembrano molto più dettagliate sia nelle zone illuminate sia in quelle in ombra, il tono e la saturazione sono diffusi in modo molto più uniforme e realistico. Oltretutto mi sono accorto che questa immagine reagisce molto bene alla post produzione anche spinta.

Mi piacerebbe che Blender avesse un pulsante nel quale impostare la Gamma e che tutto il resto avvenisse in automatico ma, molto più realisticamente, penso che sia molto utile un nodo gamma disponibile anche per il **node texture** ed una impostazione gamma del **preview**. Ho utilizzato la stesura di questi appunti per chiarire a me stesso cosa si intendesse per linear workflow spero che siano utili anche a voi.

<http://blip.tv/file/1879517> (videotutorial)

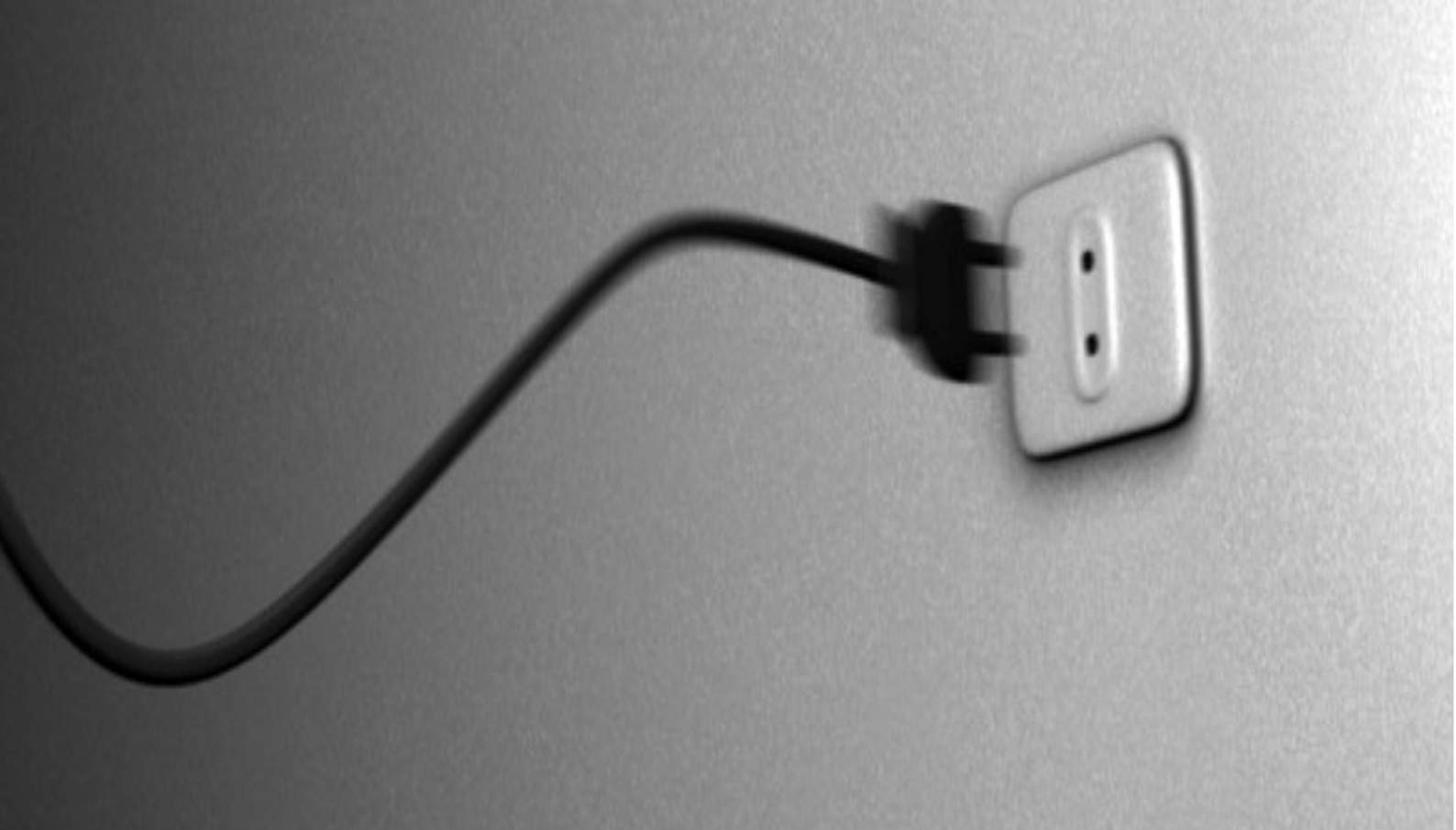
<http://www.lysator.liu.se/~zap/lwf/>

<http://www.ypoart.com/tutorials/tone/index.php>

<http://www.digikam.org/>

http://en.wikipedia.org/wiki/SRGB_color_space

<http://blenderartists.org/forum/showthread.php?t=119205&highlight=linear+workflow>

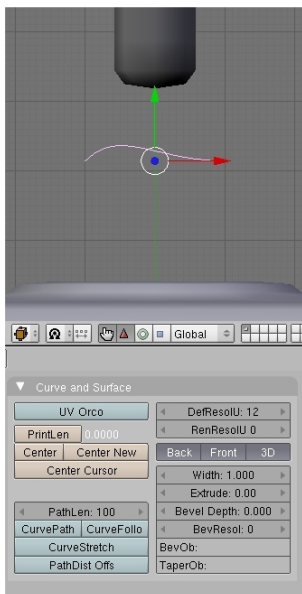
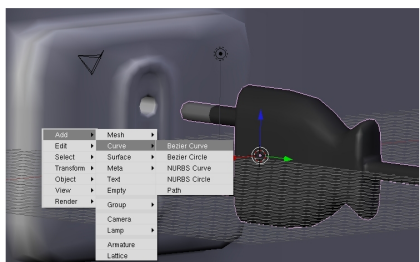
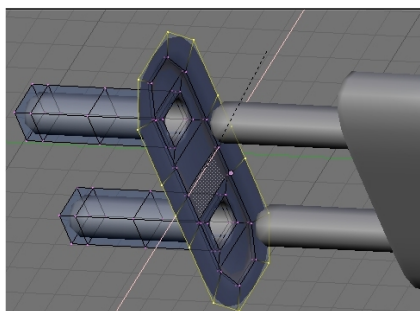


Le Shape Key Curve

Impariamo ad usare le Key Curve tramite un esempio pratico

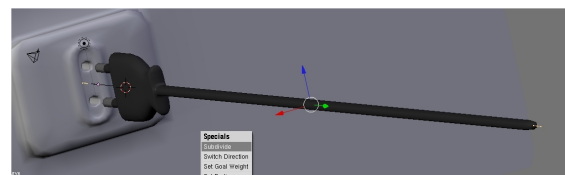
di Dawidh

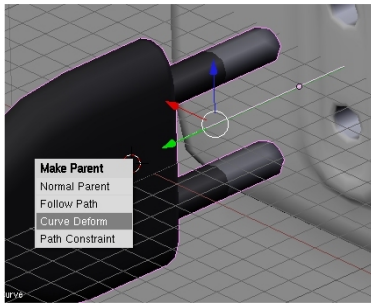
Un mio ultimo lavoro mi ha portato ad eseguire un po' di ricerca su come far deformare un percorso. Vi presento cosa ho scoperto sottoforma di una spina che viene strappata da una presa. Partendo da un cubo, comincio a modellarmi una spina, dopodiché la duplico senza spostarla, la lascio coincidere con l'originale, entro in Edit Mode e cancello tutti i vertici tranne quelli terminali ottenendo così i fori della presa. Seleziono i margini della mesh tagliata ed estraendo vado a dare forma alla mia presa e al muro.



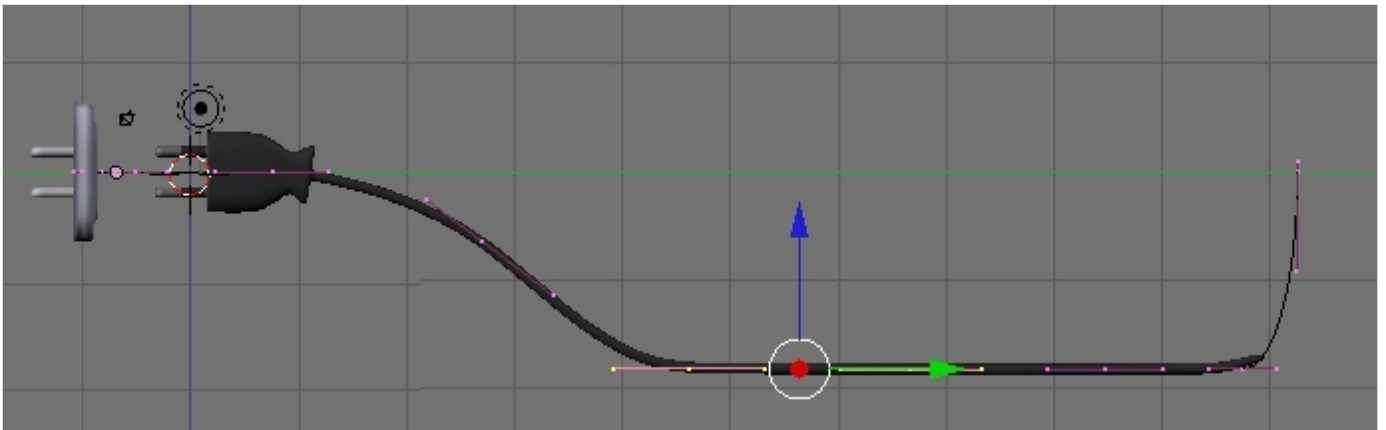
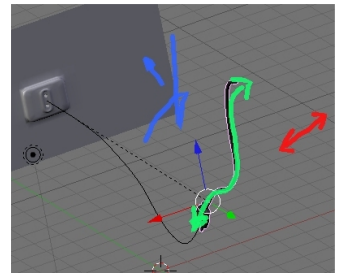
Fatto questo vado ad aggiungere una curva Beziér, che mi fungerà da percorso.

In Curve and Surface attivo il bottone 3D, entro in Edit Mode, vado a posizionare i vertici della curva lungo l'asse presa-spina-filo, seleziono tutti i vertici e premendo W la suddivido un paio di volte.

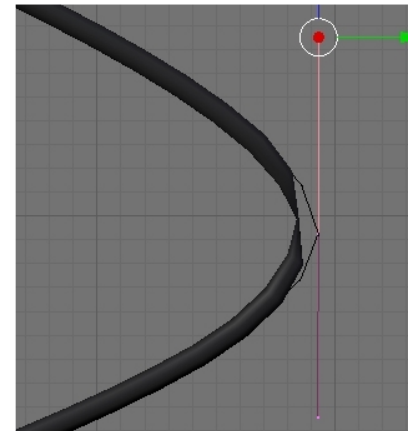




Imparento (Ctrl+P) la spina alla curva scegliendo, tra le opzioni, Curve Deform. Se tutto andrà bene, la mesh si adatterà al percorso e spostandola lungo l'asse y lo percorrerà senza problemi, altrimenti nei peggiori dei casi la mesh verrà allontanata dal percorso e dovrò cercare la giusta posizione manualmente, provando a spostarla e a ruotarla rispetto agli assi.

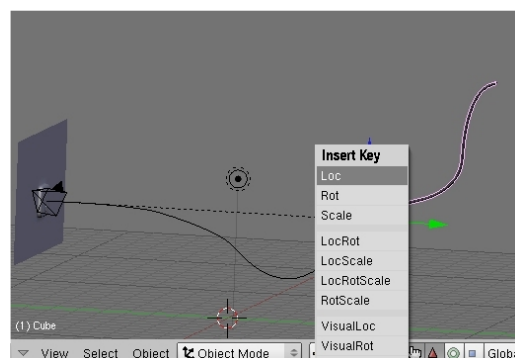
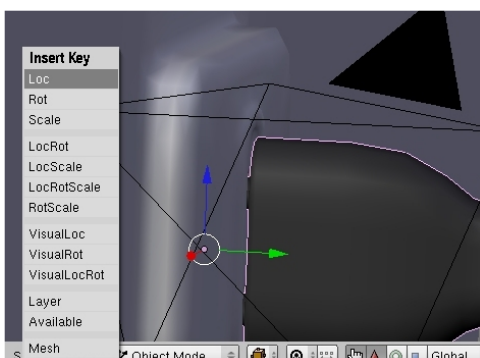


Comincio a dare forma al mio percorso spostando i vertici in Edit Mode. Se ho bisogno di un ulteriore segmento estrudo l'ultimo vertice ma potrebbe succedermi un effetto non voluto, la mesh percorrendo il percorso potrebbe implodere. Se cerco di recuperare il danno con le maniglie Beziér non ottengo grandi risultati riesco solo a spostare l'implosione. Sarebbe un bell'effetto, se stessi facendo un astronave intenta in un salto iperspaziale, ma è solo una spina. Faccio un tentativo spostando la mesh spina all'inizio del percorso, lontana dal vertice da estrarre, elimino il vertice che mi faceva implodere la mesh, ed estrudo l'ultimo vertice del percorso. Riprovo a spostare la mesh lungo il percorso e tutto sembra filare per il verso giusto.

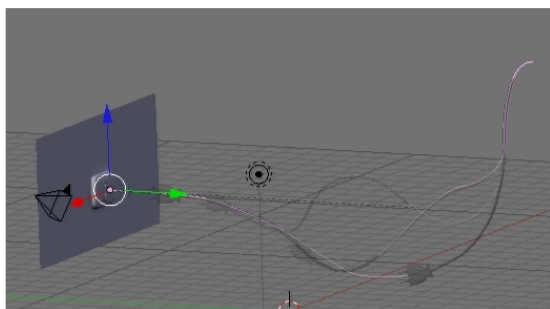
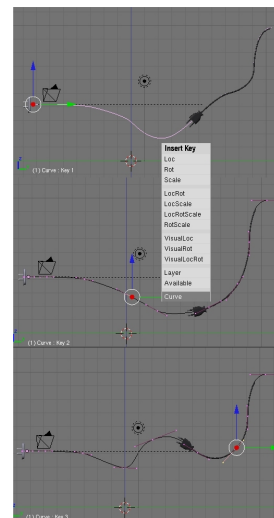
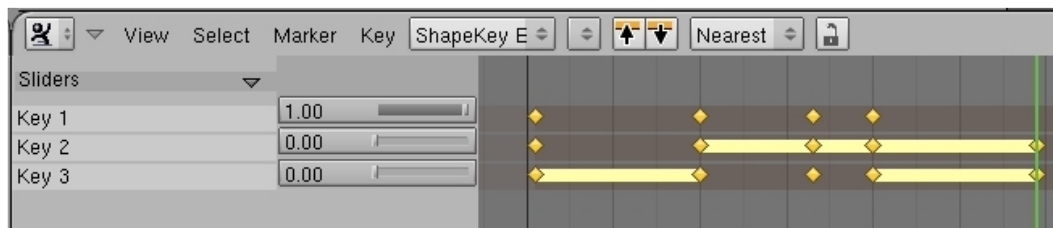


Non mi resta altro che inserire i Key frame per l'animazione.

Seleziono la mesh, al frame 0 posiziono la spina nella presa, premo I e assegno una chiave locale LOC; al frame 60 allontano lungo il percorso la spina dalla presa fino a raggiungere il capo opposto, ripremo I e assegno un'altra chiave LOC. Ritorno al frame 0 e con freccia destra mi revisiono il movimento, oppure nella finestra Timeline setto Sta:1 End: 60 e mando in play.

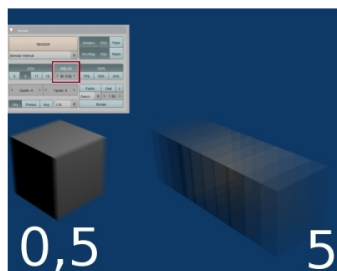
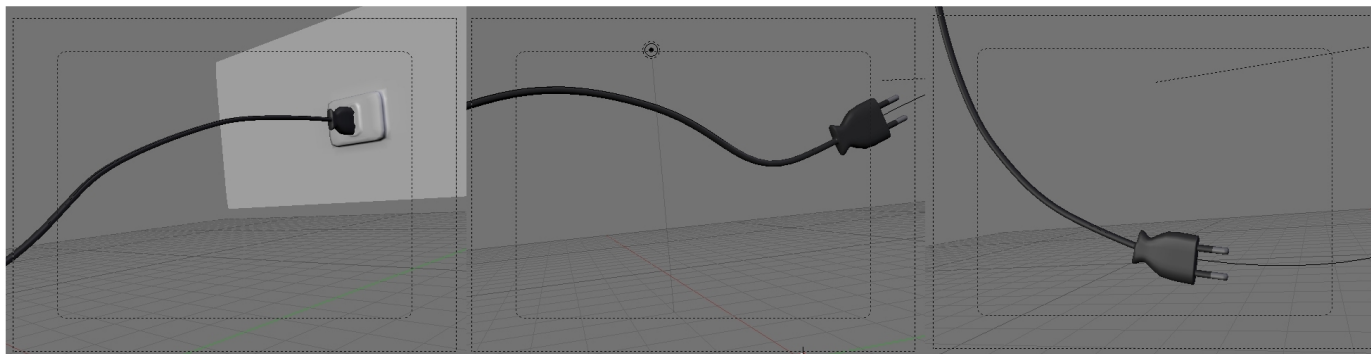


Ora è il momento di aggiungere qualche oscillazione al percorso. Seleziono il percorso, entro in Edit Mode, premo I e aggiungo una chiave Curve per la posizione di base e di nuovo premo I per la Key1. Poi modifico il percorso, aggiungo un'altra chiave Curve la Key2, modifico ancora la curva e aggiungo ancora una chiave Curve per la key3. E' possibile modificare ulteriormente le curve dopo aver assegnato le chiavi selezionando le chiavi direttamente dalla finestra IPO CURVE contesto SHAPE.



Nella finestra ACTION EDITOR, contesto SHAPEKEY EDITOR, estendo Sliders e vado a settare l'influenza delle varie chiavi lungo la time line. Se ora avvio il Play vedrò la mia spina staccarsi dal muro e oscillare contemporaneamente.

Vado a posizionare la telecamera e la animo in modo che segua la spina e che l'inquadri sempre.



Il lavoro potrebbe essere arricchito con una scintilla di sistemi particellari nell'atto dello strappo o anche solo da un motion blur selezionato nel pannello Render e impostato a 1.5 .

Alla prossima! :-)



V-Ray e Blender

L'incontro fra commerciale e open source

di Andrea Boggio (boggio95)

Cos'è V-Ray?

V-Ray è da sempre considerato il miglior rendering engine biased. La sua resa è comparabile effettivamente ad altri engine unbiased, come LuxRender o Maxwell. Inoltre è molto versatile e può essere usato per qualsiasi tipo di rendering, a volte (ma raramente) anche per l'animazione; ma questo è solo uno dei fattori che ha portato questo software al successo. Uno dei vantaggi di V-Ray è la sua ampia compatibilità con i più famosi software di grafica tridimensionale, tra cui 3ds Max, Maya, Sketchup, Cinema 4D e alla lista si è aggiunto anche Blender....

Dall'exporter all'integrazione

Il progetto è di Andrey Izrantsev (bdancer sul web), programmatore russo che ha deciso di esserne il pioniere; Andrey ha deciso di dividere il progetto in due parti: un exporter (quello attuale), e l'integrazione totale in Blender.

Impresa ardua, ma non impossibile....

Per ora l'exporter supporta molte delle funzioni di V-Ray Standalone, la versione ancora in beta di V-Ray che dovrebbe essere rilasciata proprio quest'anno. Le principali caratteristiche supportate dall'exporter sono Global Illumination (GI), Mappatura esponenziale dei colori, Illuminazione Sun+Sky, Physical Camera, ecc.... Ma non sto ad annoiarvi con questi dettagli, andiamo a vedere la parte pratica.

Il primo approccio

Prima di tutto procuriamoci il materiale necessario: da <http://blender.bevice.ru/> preleviamo l'ultima versione dell'exporter, e dal sito di Chaos Group scarichiamo la build 08529 di V-Ray for Maya (sezione Download - Necessaria la registrazione).

Ora installiamo la build appena scaricata: quando apparirà la schermata di selezione del percorso di installazione, consiglio di installare nella stessa cartella tutti e tre i componenti che

appariranno; ricordate che su Vista dovete avviare l'installer come amministratore. Infine copiate lo script nella cartella degli scripts di Blender.

Ora avviamo Blender e dividiamo lo schermo, in modo da avere da una parte la vista 3D e dall'altra una finestra per lo script.

In quest'ultima cambiamo pannello e andiamo su "Scripts Window". Dal menù "Script" selezioniamo "V-Ray/Blender" nella categoria "Render".

Vedremo il pannello dell'exporter apparire sullo schermo: analizziamo le varie voci.

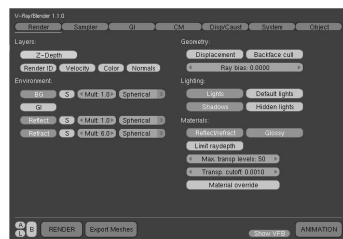
Iniziamo dal pannello Render: qui troviamo i settaggi delle mappe BG (Background),

GI (Global Illumination), Reflect e Refract. Inoltre possiamo anche scegliere i layers da renderizzare (in alto a sinistra) e attivare il Displacement.

Il pannello Sampler invece è molto intuitivo: troviamo i settaggi antialiasing, Image Sampler e le impostazioni del Sampler DMC (quello normalmente usato).

Ecco il pannello più importante: GI (ormai lo sapete, Global Illumination).

Qui troviamo tutti i settaggi come Irradiance Map, Light Cache, Brute Force, Photon Map, ecc....



Pannello Color Mapping: qua possiamo scegliere il tipo di calcolo dei colori (di solito si usa Exponential).

Il pannello Disp\Caus permette di attivare il Displacement sulla mesh selezionata.

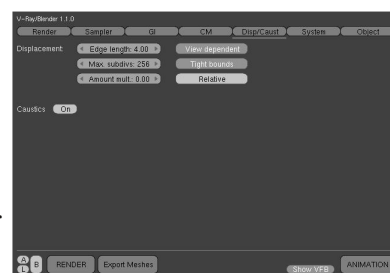
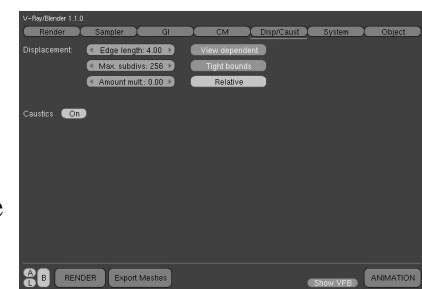
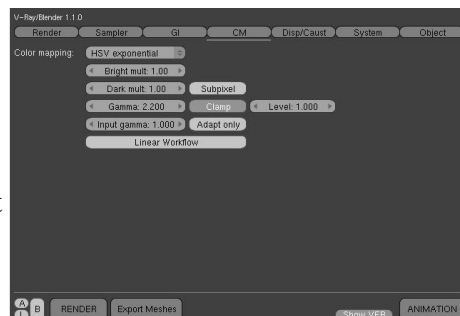
Inoltre nello stesso pannello si può attivare Caustic e regolare le impostazioni. Nel pannello system infine

dovremo inserire il percorso dell'eseguibile di V-Ray. Esplorate quindi le cartelle fino a trovarlo; selezionatelo e cliccate su save.

Ultimo pannello (ma non meno importante) e Object: qui possiamo regolare le impostazioni per ogni oggetto:

Se selezioniamo la camera potremo impostarla su Physical e regolarne ISO, Shutter Speed ecc... Se selezioniamo invece una luce sul pannello potremo sceglierne la potenza e impostare il sole della funzione Sun+Sky (se selezioniamo una luce Sun). Di seguito vi è la lista delle luci in corrispondenza con quelle di V-Ray:

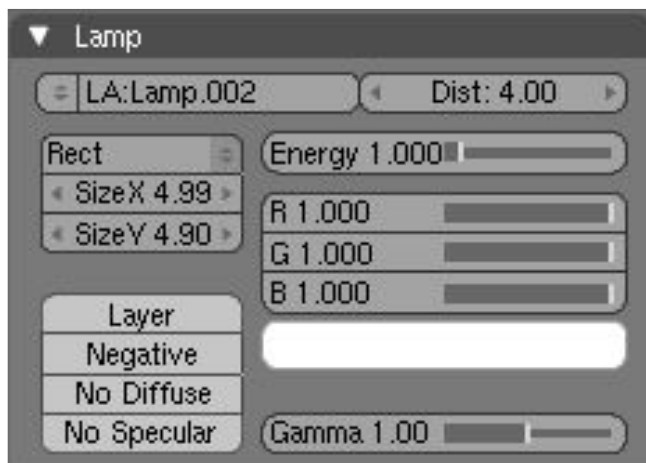
- Lamp - VRayLight Sphere
- Area - VRayLight Rectangle
- Spot - VRaySpotLight, VRayLightIES
- Sun - VRayDirectLight, VRaySun
- Hemi - VRayDomeLight



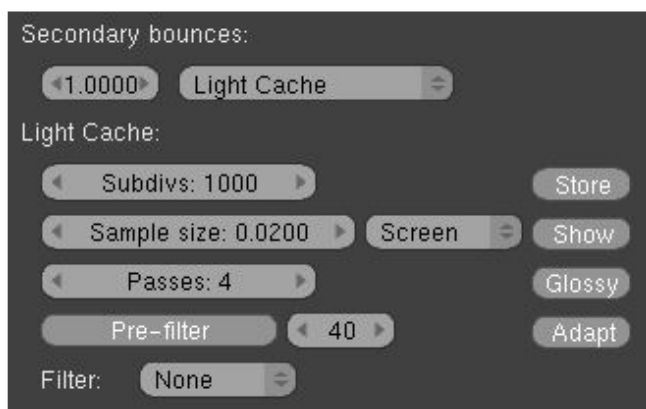
Scene e settaggi di base

Interno

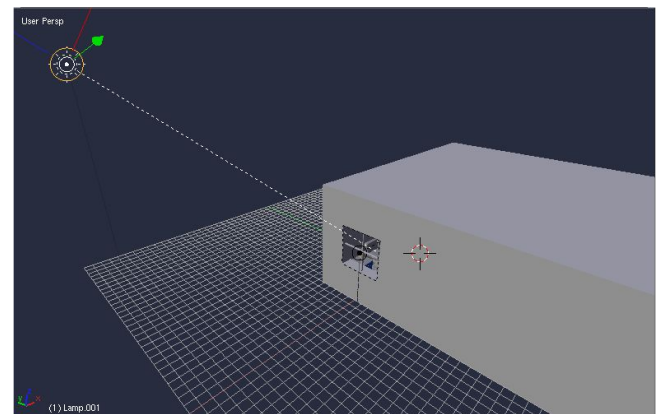
Impostare le luci è abbastanza semplice: bisogna aggiungere una lampada Sun che punti verso la finestra della stanza e che abbia Ray Shadow attivato; con questa lampada selezionata nel pannello object di V-Ray clicchiamo su "Sun". Poi basterà aggiungere una lampada Area che punti verso la finestra (nel pannello V-Ray settiamo su "Skylight Portal"), modificata adattandola alla forma della finestra con il parametro Rect nel pannello delle luci di Blender.



E anche secondary bounces:



Infine nel pannello CM impostate "Exponential"

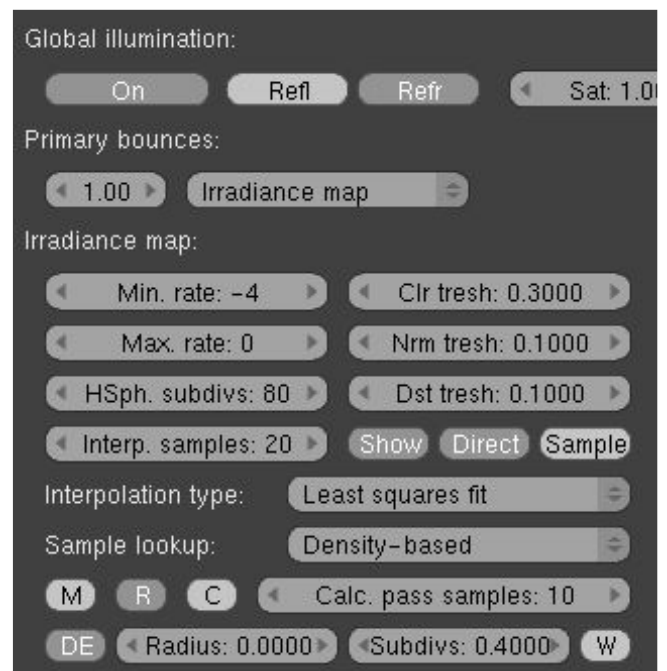


Ora andiamo nei pannelli V-Ray:

Nel pannello render attiviamo "BG" e "GI", e per ognuno possiamo scegliere se aggiungere un'immagine (deve essere una texture chiamata rispettivamente ENV_BG o ENV_GI nel pannello world di Blender) o usare il V-Ray Sunsky (basterà cliccare su "S" vicino alle opzioni "BG" o "GI").

Nel pannello Sampler scegliete DMC per un render più dettagliato e aggiungete un po' di AntiAliasing (Gaussiano va bene).

Per il pannello GI la faccenda è più complicata; semplicemente, settate come in figura:

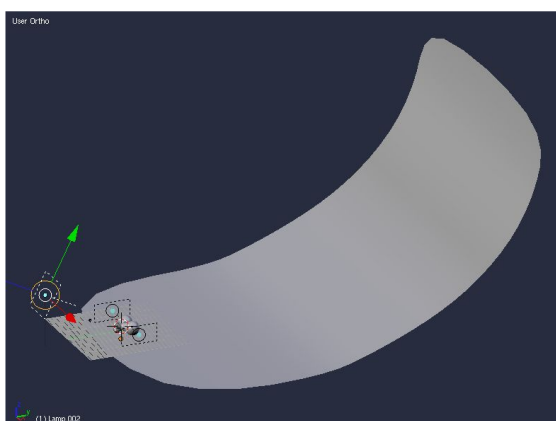


Esterno

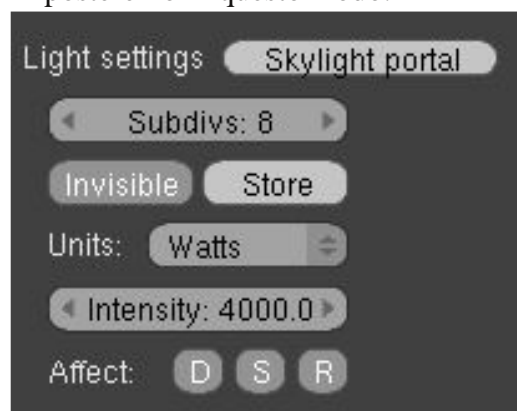
Per i render d'esterni i settaggi sono uguali ai quelli per gli interni, anche se con prove ed esperimenti potrete ottenere una resa più realistica. E' essenziale per una illuminazione basilare avere una lampada Sun con Ray Shadows attivato e "Sun" attivato nel pannello Objects di V-Ray.

Studio Lights

Per "Studio Lights" si intende un render di oggetti su sfondo bianco (o graduale in scala di grigi). Per renderizzare questo tipo di scena bisogna creare un piano come in figura



su cui poggeremo i nostri oggetti; poi ai lati dell'oggetto e sopra ad essa aggiungiamo delle lampade area che puntano verso lo stesso, e le imposteremo in questo modo:



Scegliamo il valore in Watt (o Lumen, lo potete selezionare da "Units") e rendiamo invisibile la lampada (in realtà questo elimina solo il rettangolo bianco generato dalla stessa).

Infine settate i vari pannelli di V-Ray come per gli altri tipi di settaggi. Io personalmente preferisco usare una lampada Sun invece che area che punti dall'alto verso l'oggetto (con i relativi settaggi Sun e Sky). Infine per avere una migliore resa sui materiali riflettenti possiamo aggiungere una mappa HDRI. Per farlo basta andare nel pannello Render di V-Ray e attivare "Reflect". Poi aggiungiamo una texture nel pannello World e chiamiamola ENV_REFLECT; in quest'ultima carichiamo la nostra HDRI e applichamola a Blend e a Hori.

Render

Ora che abbiamo settato la scena con le impostazioni basilari possiamo procedere con le impostazioni della Physical Camera. Selezioniamo la nostra telecamera e andiamo nel pannello Object di V-Ray: qui attiviamo Physical e impostiamo la sensibilità (ISO) su 100 (il valore di default utilizzato la maggior parte delle volte). Ora possiamo esportare la scena e renderizzare; clicchiamo quindi su "Export Meshes" e attendiamo che Blender finisca di esportare la scena. Se abbiamo impostato correttamente il pannello System clicchiamo su render e dopo pochi secondi la finestra di V-Ray dovrebbe apparire: se utilizzate la versione demo di V-Ray for Maya dovreste abbassare la risoluzione sotto i 600x450 (che è appunto il limite della demo). Se la finestra non apparirà guardate quale errore riporta la console di Blender (può essere un limite della demo, una versione errata di V-Ray, ecc....). Se tutto andrà come previsto vedremo l'immagine comporsi pian piano sullo schermo.

Conclusioni

Una soluzione che combina algoritmi avanzati di rendering e tempi di "cottura" contenuti. Un'esperienza da provare sicuramente.



3d Compositing

Come inserire elementi 3d in ambienti reali usando i nodi

di Anfeo

Possiamo iniziare questo tutorial utilizzando il file .blend ottenuto nella prima parte che potete trovare su BMI4. In oltre si richiederà una conoscenza base del sistema a Nodi e dei Render Pass, anch'essi già accennati in un tutorial su BMI3.

Questa serie di tutorial saranno finalizzati allo scopo di imparare i rudimenti di compositing 3D, color correction e altri principi che stanno alla base della postproduzione e del reparto VFX.

Preparazione

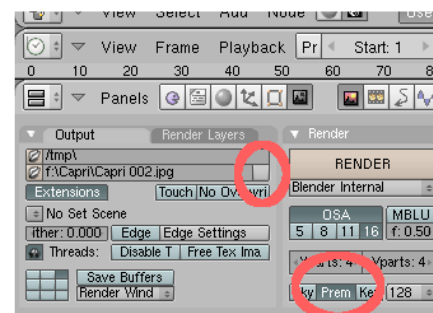
01

Iniziamo subito con l'aprire il file in allegato su BMI4 che mostra una scena di compositing 3d settata di luci e soprattutto di allineamento della camera.

NB: Il file che si trova allegato a questo numero della rivista è la scena finale di questo tutorial, evitiamo di confonderli.

In teoria potreste usare anche il file da voi creato grazie alla prima parte di questo tutorial, a voi la scelta.

Andiamo a cambiare qualche impostazione di rendering come da figura:

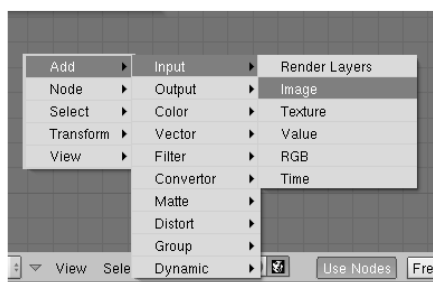


Noterete che abbiamo disattivato l'immagine di Backbuffer e soprattutto modificato il parametro di rendering dello sfondo da "Sky" a "Premul". Utilizzando Premul, lo sfondo risulterà completamente nero, ma soprattutto gli oggetti non presenteranno uno strano alone durante il compositing con i Nodi (provare per credere).

Composite Node

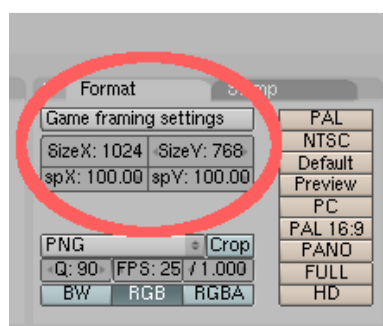
02 Ora apriamo una finestra del Composite Node, attiviamo il pulsante “Scenes” e il pulsante “Use Nodes”.

Ora aggiungiamo un nodo di Input “Image” e da esso carichiamo l’immagine di sfondo: “Capri002.jpg”.



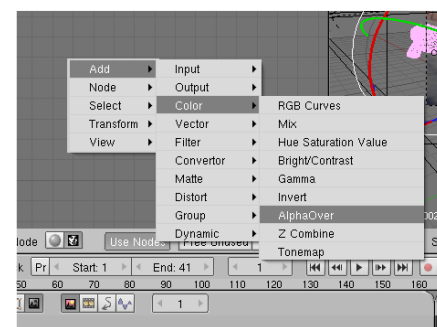
Adattiamo il rendering

03 L’immagine che abbiamo caricato è grande “1024x768 px” quindi anche il nostro rendering dovrà essere grande uguale, allora modifichiamo la grandezza del rendering nel pannello di controllo apposito:



Alpha Over

04 Torniamo alla finestra dei Nodi e aggiungiamo un nodo “Alpha Over”, situato sotto la categoria “Color”:



Sovrapponiamo le immagini

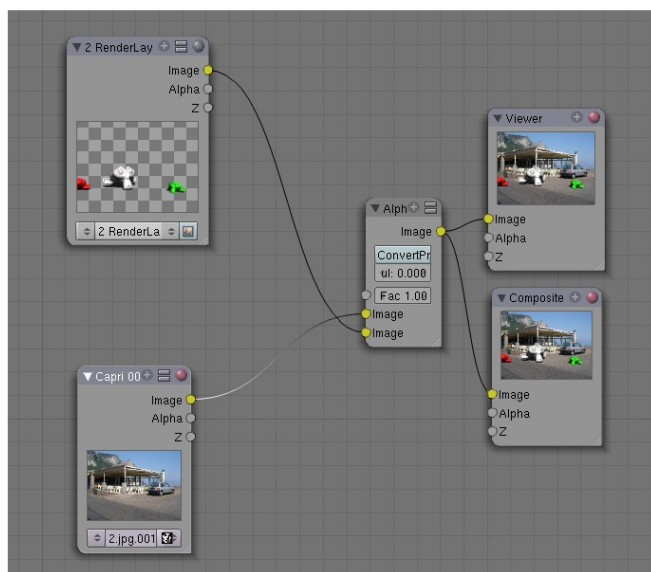
05 Uniamo i nodi come da immagine.

Ricordo che il nodo “Alpha Over” unisce due immagini, quella della prima casella, con sopra quella della seconda casella,

utilizzando in automatico l’Alpha dell’immagine come maschera.

La scelta del “Premul” in fase di rendering è apposita per questo nodo. NB: Potete provare a variare il valore del nodo “Alpha Over”, per vedere apparire un alone nero intorno

all’immagine renderizzata. Sperimentare con i nodi è sempre utile, possono essere trovate molteplici soluzioni ad un singolo problema.



Alpha Over

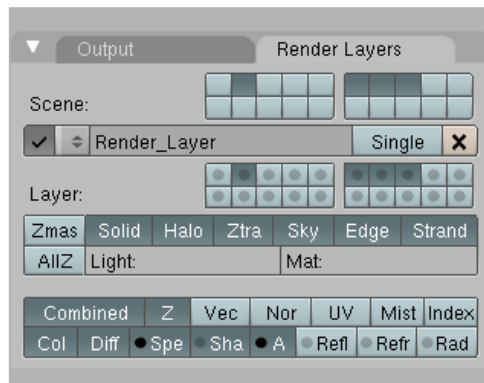
06 Facciamo un rendering e osserviamo l’immagine finale: Ora siamo allo stesso punto del



tutorial precedente, da qui in avanti le strada da prendere possono essere molte, la cosa più facile sarebbe quella di prendere l’uscita “Image” del nodo Render Layer e tramite nodi di correzione colore, aggiustare un po’ la tonalità e la saturazione per farla coincidere con quella della foto.

Render Pass

07 Noi però siamo qui per imparare e soprattutto vogliamo di più dalla nostra immagine, quindi ci avvaliamo dei “Render Pass” e agendo separatamente su certi output avremo più controllo dei singoli parametri dell’immagine, come Ombre, Specular, AO (che prima non abbiamo usato) e Colore. Settiamo il nostro Render Layer in questo modo:

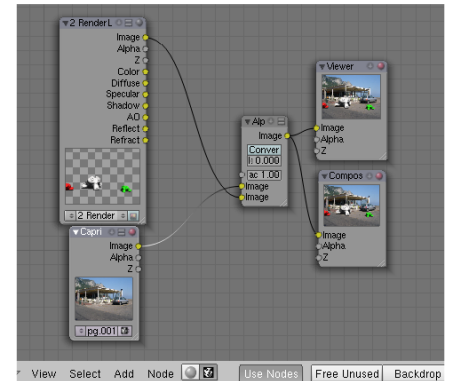


Dando una breve spiegazione, possiamo dire che i Pass: AO e Specular, sono stati esclusi (pallino nero,

cliccando e tenendo premuto il tasto CTRL) dal rimontaggio automatico dei Pass (Combined), questo per montarli manualmente e con qualche lieve modifica dopo.

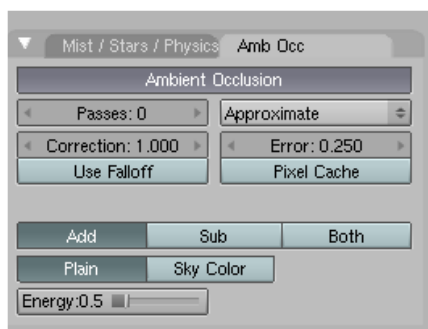
I Nodi dei Pass

08 Ecco come dovrebbe apparire il nodo Render Layer (non badate alle uscite Ref e Refr):



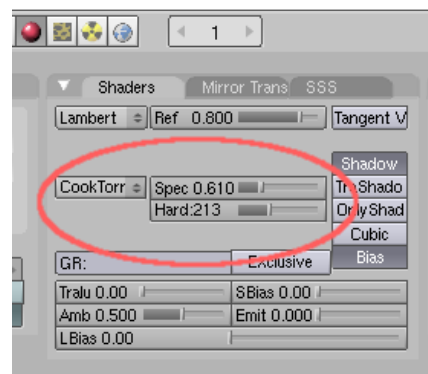
Ambient Occlusion

09 Prima di fare il rendering attiviamo l’AO con questi parametri:



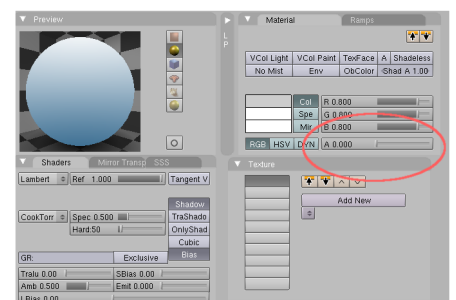
Aggiungiamo un po' di Specular

10 Volendo possiamo modificare il valore di Spec del materiale della scimmia grigia, giusto per provare il nodo di Spec:



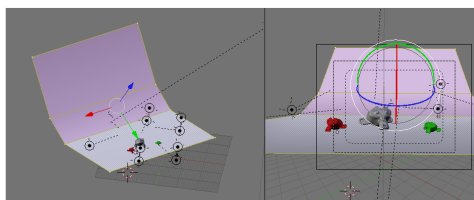
Il materiale del Plane

11 Dobbiamo modificare anche il materiale del piano di appoggio, in quanto il pulsante Only Shadow non funziona correttamente con l’AO, quindi bisogna disattivarlo e mettere l’alpha del materiale a 0:



La forma del Plane

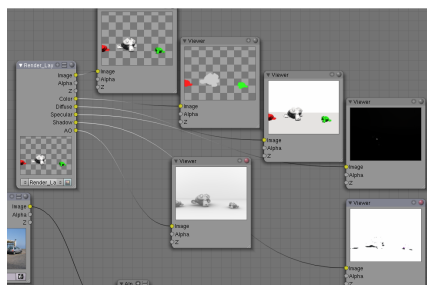
12 Ultima modifica al file, è richiesto che il piano di appoggio compia una curva verso lo sfondo, in maniera tale che sia il Pass di AO che quello di Shadow, non presentino la vista dello sfondo. In oltre il piano deve essere curvo in quanto angoli evidenti, non darebbero uno sfondo omogeneo per il compositing (si capirà meglio dopo guardando i Render Pass).



Che cosa è successo? In pratica il Pass Diffuse è uguale al Pass Image, ma quest'ultimo supporta il canale Alpha, allora, perché non usare questo? Per le ombre, rimane il problema che non si moltiplicano su un canale trasparente, quindi ci conviene moltiplicarle direttamente sull'immagine di sfondo, magari insieme al canale AO, per dare un po' di profondità alle ombre. Poi non resta che procedere come da tutorial, unire i restanti Pass e sovrapporre il tutto con il Nodo "Alpha Over".

I render Pass

13 Facciamo infine un rendering e vediamo i Render Pass ottenuti:

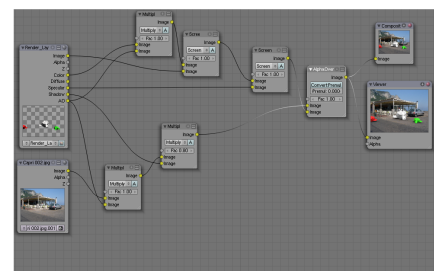


Come potete notare il Pass di Shadow è completamente Bianco, tranne che per le ombre. Se il piano fosse rimasto uguale, avremo avuto lo sfondo nero in corrispondenza della fine del piano.

Lo stesso discorso vale per il Passi di AO, in questo caso si nota ancora una divisione, eliminabile aumentando la curvatura del piano.

Problema sul Diffuse Pass

14 Seguendo il tutorial sui Render Pass, possiamo rimontare facilmente l'immagine finale, ma sorge subito un problema: il nodo Diffuse non supporta la trasparenza del materiale, quindi rimane impossibile fondere il rendering con la foto. Come risolvere? Vi invito a fare delle prove prima di andare a leggere la soluzione, ma per chi vuole andare avanti, ecco come si risolve:



L'immagine risultante

15 Osserviamo cosa è successo all'immagine risultante:



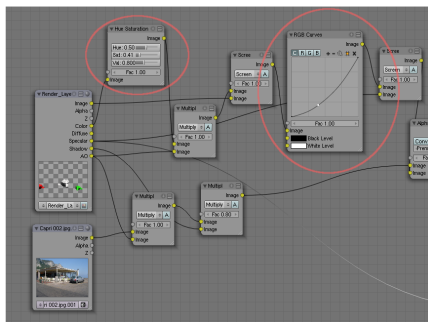
Questa immagine è molto più luminosa della prima composizione, spieghiamo perché: In pratica ora abbiamo anche l'AO attivo e questo conferisce un ulteriore fake GI, senza

considerare che il piano di appoggio è Alpha=0, quindi gli oggetti ricevono anche un po' di luce da sotto.

A questo possiamo rimediare in diversi modi, possiamo modificare un po' l'illuminazione o disattivare l'AO, ma come al solito vogliamo complicarci la vita, quindi correggiamo questi difetti con i nodi.

Agiamo sui colori

16 Aggiungiamo in serie due Nodi come da figura, uno “Hue Saturation Value” e uno di “RGB Curves”, entrambi si trovano sotto la voce “Color”. Osserviamo la loro posizione:



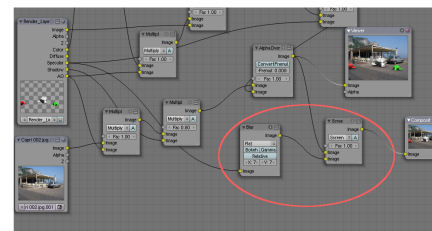
Il primo Nodo (Hue Saturation Value) è posizionato tra l’output “Color” e l’input sul primo “Multiply”. Questo nodo va a modificare il valore di saturazione

dei colori del rendering, abbassandolo, in oltre agisce anche sulla luminosità (Value), abbassando anche quella (vedere i valori in figura).

Il secondo nodo (RGB Curves), agisce sull’output “quasi finale” dell’immagine renderizzata, il punto in cui il 3d non è stato ancora compositato con lo Spec e con l’immagine di sfondo. Modificando la curva come da figura, otteniamo un notevole abbassamento della luminosità, in oltre questa curva può essere ritoccata per definire il giusto livello di luminosità e contrasto del rendering con l’immagine reale.

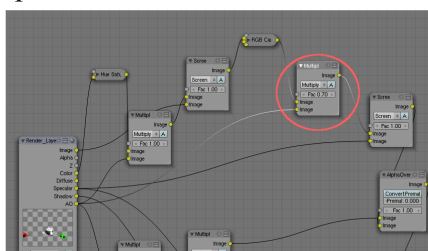
Un tocco di Glow Effeto

17 Aggiungiamo un po’ di Glow sfocando il Pass Specular e aggiungendolo come screen al nodo finale.



Scuriamo con un'aggiunta di AO

18 Volendo possiamo scurire ulteriormente le scimmie aggiungendo come “Multiply” il Pass AO a questo nodo:



L'immagine finale

19 Questa è l’immagine finale:



Possiamo continuare a regolare i valori per ottenere una perfetta fusione, si può aggiungere un ulteriore nodo “RGB Curves” per controllare le ombre o l’AO

Con oggetti e materiali diversi

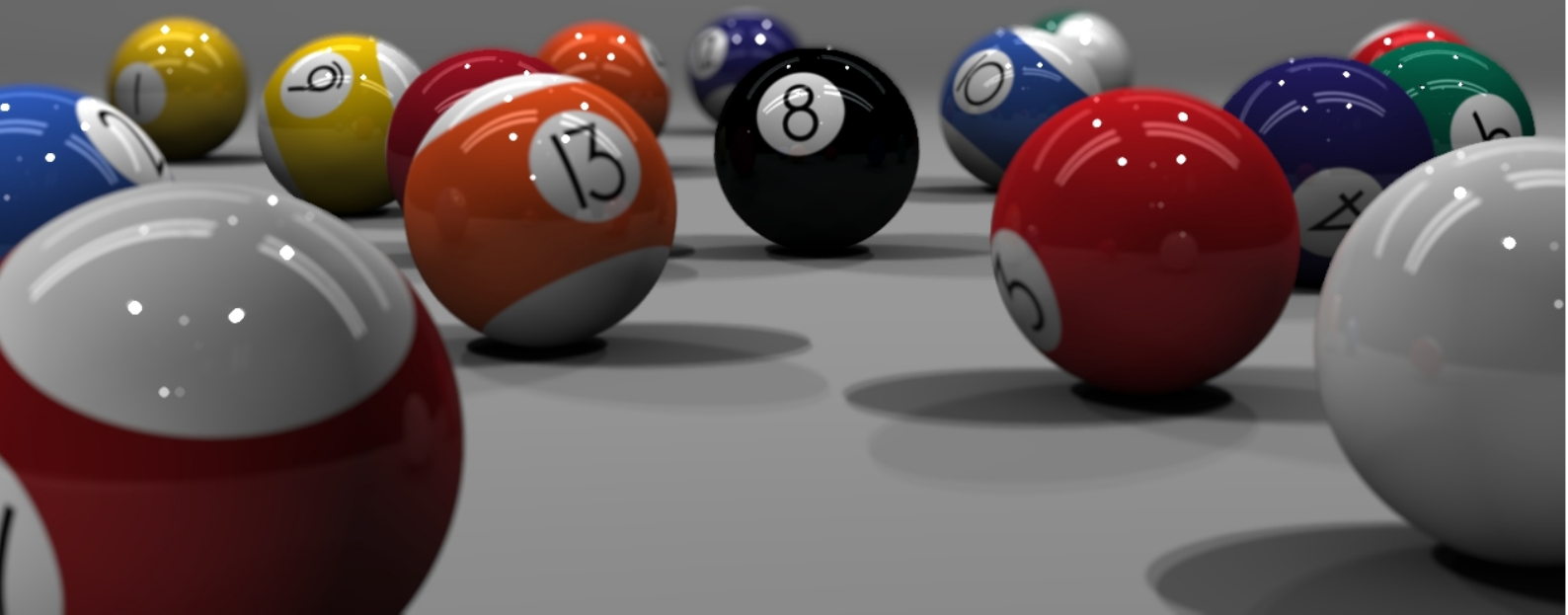
20 Ad esempio provando ad inserire un oggetto con una texture migliore, bisogna aggiustare la curva del nodo “RGB Curves” altrimenti si ha un’immagine troppo scura.



Potete sperimentare i risultati con il file completo “Tut_Fotoinserimento_Seconda parte_A1.blend” presente in allegato alla rivista.

Conclusioni

Con questo abbiamo segnato un’altra tappa nel percorso degli inserimenti 3d. Le basi, nel prossimo numero vedremo come applicare questa tecnica ad una ripresa video, concentrandoci di più sul tracking del video e della sua importazione in Blender.



Creare una scena Still Life

Settare Blender come uno studio fotografico

di t0m0 aka 'Rik'

Salve a tutti.

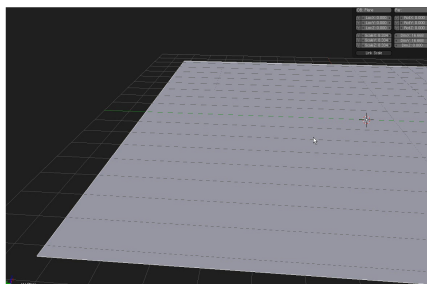
Questo è il primo tutorial che scrivo, nello specifico vi spiegherò come creare una scena da still life (o da studio fotografico) con il nostro amato frullatore.

Per chi volesse affrontare solo una delle due parti (chi non ha voglia o si ritiene abbastanza bravo da non aver bisogno della modellazione), può trovare in allegato il file .blend con tutta la scena pronta per essere renderizzata e compositata. Consiglio vivamente di seguire entrambe le parti perché non si sa mai che qualche cosa di nuovo capiti sotto gli occhi. Bando alle ciance e cominciamo!

Parte 1: preparare la scena

01

Apriamo blender, cancelliamo il cubo se presente e inseriamo un piano e scaliamolo circa come in figura:



02

Entriamo in edit mode, selezioniamo un edge e estrudiamolo come in figura. Aggiungiamoci anche un paio di loop come mostrato in fig 3 e 4.

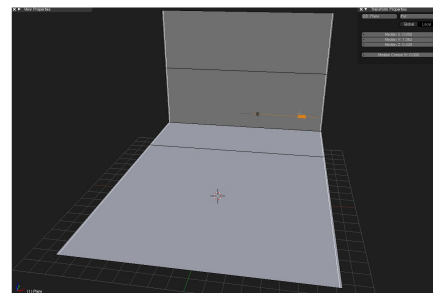


Fig 3

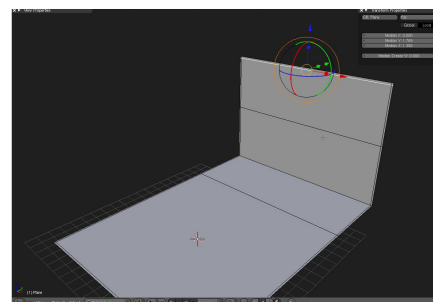


Fig 4

03 Assegniamoli un modificatore subsurf con livello 2 o 3, scaliamo il piano e dalla viewport cerchiamo un angolo di ripresa abbastanza basso. una volta trovata l'angolazione che più ci piace schiacciamo ctrl+alt+0 (del tastierino numerico) per allineare la camera alla nostra vista corrente. Andiamo nelle proprietà della camera e settiamo come lunghezza focale 50mm. (fig 5,6)

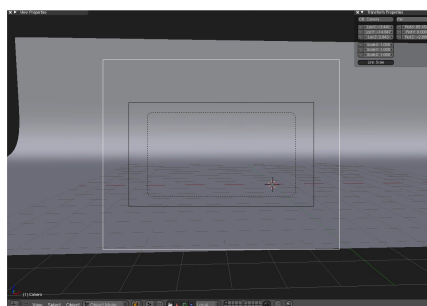
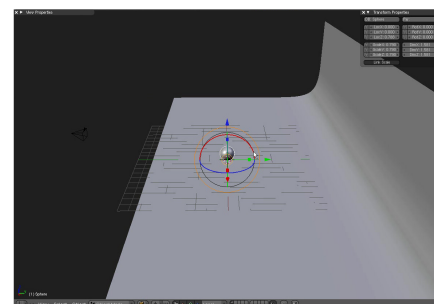


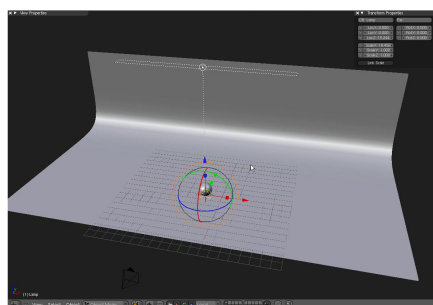
Fig 5 (sopra) e Fig 6 (Sotto)



04 A questo punto torniamo nella vista prospettica e aggiungiamo una sfera (le impostazioni di default vanno benissimo) al centro degli assi e la riposizioniamo in modo che appoggi sul nostro piano (eventualmente riscalamola leggermente).



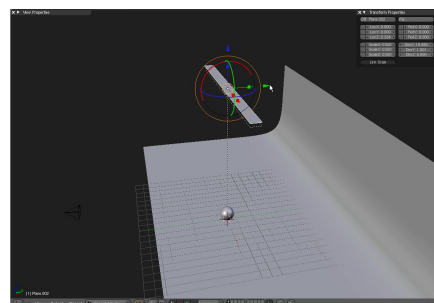
05 Ora aggiungiamo la prima luce del tipo Area (nel caso ci sia in scena quella di default eliminiamola), scaliamola e posizioniamola come in figura.



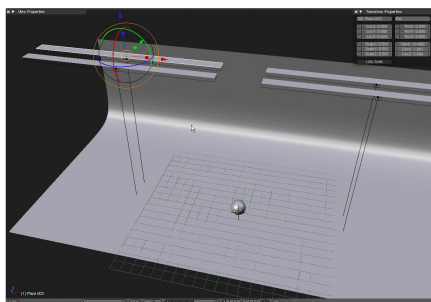
06 Mentre stavo creando i materiali, mi sono accorto che le area light non si vedono nei riflessi. per ovviare questo problema creiamo un piano e gli aggiungiamo un constraint del tipo ChildOf. Nell'apposito spazio scriviamo il nome del padre da cui vogliamo che erediti le trasformazioni, nel nostro caso la luce Lamp.



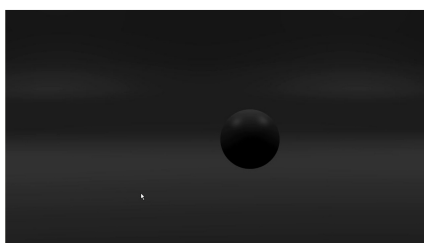
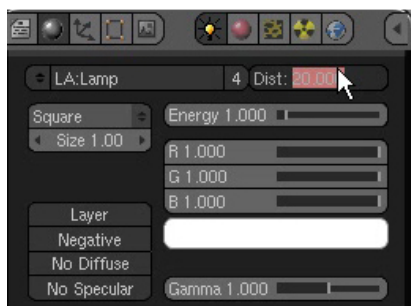
07 Una volta attivato il constraint, il nostro piano andrà a posizionarsi esattamente sulla nostra AreaLight. Andiamo a scalare la sua dimensione per fargli assumere le stesse proporzioni della luce e a spostarlo leggermente sull'asse Z, in modo che non sia sovrapposto alla luce



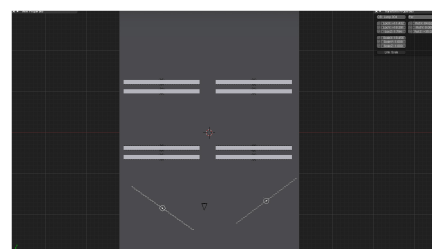
08 Duplichiamo la nostra luce con il tasto Alt+d (duplicate Linked, in modo che modificando un parametro della luce tutte le altre ereditano la modifica) e aggiungiamo altri piani per ottenere un setup come in figura.



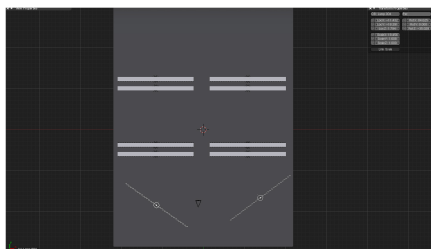
09 Andiamo ora nelle impostazioni della luce e modifichiamo il parametro dist, da 20 lo settiamo a 1 e facciamo un render di prova. (fig 12,13)



10 Nel render vediamo che il nostro limbo (il piano che abbiamo creato all'inizio) è colpito dalle luci, quindi andiamo a modificare le sue dimensioni come mostrato in figura e attiviamo sulle luci le ombre e aumentiamo la loro intensità a 3 (ricordo che basta farlo su una luce e tutte le altre ereditano la modifica).



11 Duplichiamo (alt+d) le nostre 4 luci e spostiamole un po' verso il fondo (aggiungiamoci i relativi pannelli). andiamo in oltre ad aggiungere 2 luci (questa volta le duplichiamo con la combinazione shift+d) con intensità 1, disattivandogli le ombre e andiamo a posizionare tutto come in figura :



12 Ora è tempo di lavorare sulla nostra sfera. Anzitutto dobbiamo calcolargli le UV, quindi selezioniamola, andiamo su una qualsiasi vista ortogonale (1,3,7 o 9 del tastierino numerico), entriamo in edit mode e selezioniamo tutti gli edge ('a' un paio di volte). a questo punto schiacciamo 'u' e scegliamo 'Sphere from View'. ok, le nostre Uv sono state create. (fig 16,17)

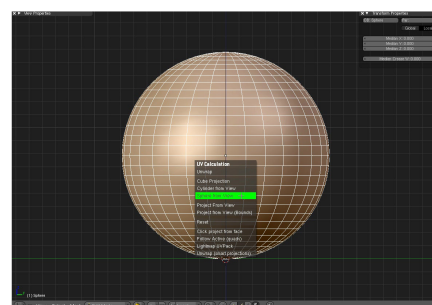
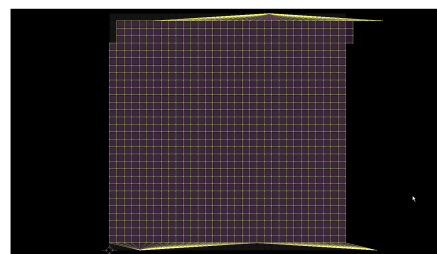


Fig. 16 (sopra) e Fig 17 (Sotto)



13 Prima di proseguire, organizziamo un attimo la scena. Selezioniamo il piano e schiacciamo 'm', ci si aprirà un box da dove poter scegliere un layer su cui spostare il nostro oggetto. selezioniamo il secondo in alto a sinistra. Ora selezioniamo tutte le luci, i piani corrispondenti e la camera e li spostiamo sul primo layer in basso a sinistra.



Fig19

A questo punto possiamo iniziare a creare il materiale per la sfera. Chiamiamolo '01' e usiamo le impostazioni come in figura. Carichiamo anche la texture 'Biliardo_01.jpg' e la mappiamo nel materiale sempre come in figura 18.

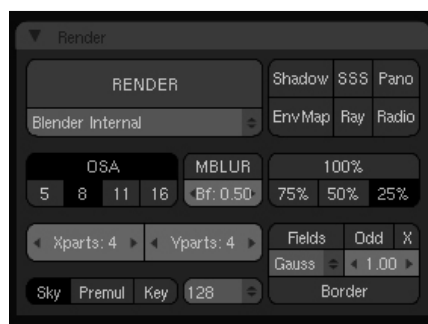


Fig18

16 Una volta completati i materiali dovreste ottenere un render simile a questo.



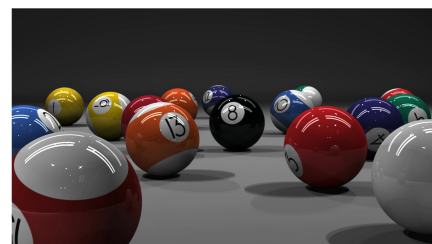
17 Ora non resta che ruotare randomicamente le nostre sfere. Io ho proceduto girandole a caso col tasto r, poi una volta che non erano tutte più orientate allo stesso modo, nel pannello del render ho disattivato tutti i calcoli e mettendo le dimensioni al 25%(vedi figura). In questo modo avevo un riscontro immediato del loro orientamento.



15 Ok, la scena è quasi pronta. Dobbiamo ora duplicare (shift+d) la nostra sfera e posizionare le copie sul nostro piano. Questa operazione conviene farla dalla vista top, in modo che abbiamo una visione più chiara della loro distribuzione. State attenti a non metterne più di 16 (le palle da biliardo sono 15 più la bianca :D) e a controllare sempre che siano tutte visibili da camera (0 sul tastierino).

Una volta posizionate nello spazio dovete assegnare a ogni sfera il materiale con la propria texture. Se avete fatto tutto correttamente, una volta che create un nuovo materiale, esso avrà le impostazioni di quello precedente, quindi non vi resta che sostituire la texture (e mi raccomando i nomi corretti..!)

18 Una volta che finito, il nostro render finale dovrebbe risultare più o meno così:



Parte 2: Render e compositing

19 Ok, ora che abbiamo creato la nostra scena con luci e materiali, andiamo a fare il setup dei render layer. Per una descrizione accurata dei singoli layer e del loro utilizzo generale, rimando all'articolo di Anfeo sul numero precedente di BMI (n°3, Ottobre 08).

Anzitutto dobbiamo capire quali pass ci possono tornare utili in fase di compositing, ovvero su quali attributi vogliamo avere il controllo in un secondo momento per poter rifinire meglio la nostra immagine.

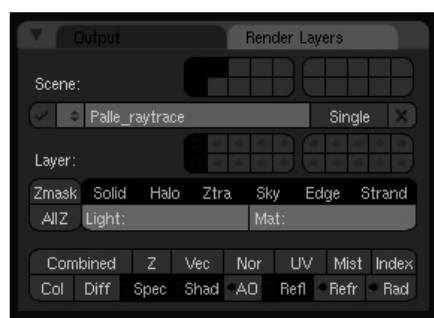
Sicuramente il pass delle texture, la nostra base su cui andremo a costruire l'albero. Poi avremo bisogno dello shading, per

poter controllare l'intensità delle nostre zone d'ombra sulle sfere; i riflessi per decidere quanto devono incidere sulla nostra immagine; il pass di index, per avere delle comode maschere dei nostri oggetti; il pass di z per poter controllare la sfocatura di campo; il pass delle ombre e l'Ambient Occlusion.

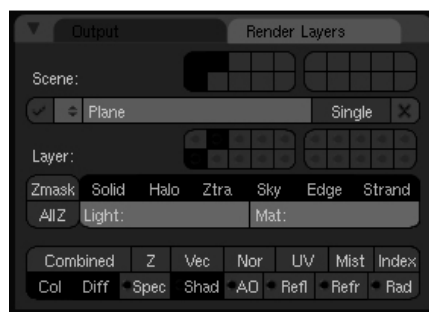
Anzitutto dobbiamo assegnare a ogni sfera un differente indice. per fare questo dobbiamo andare in modalità Object (sotto pannello Object and Links -> Pass Index) e assegnare un numero unico a ogni sfera. Per comodità io gli ho assegnato il numero corrispondente al materiale assegnato, dando alla sfera bianca il numero 16.



Fig. 23 (Sopra) e 24 (sotto)



21 Ora creiamo il render Layer per il piano e usiamo i seguenti settaggi:



20 Una volta fatto questo andiamo nel pannello di render e nel tab

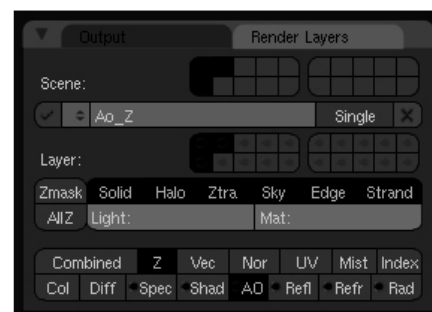
Render Layer.

Creiamo un nuovo render layer e lo chiamiamo 'Palle_color'.

Se notiamo, nel tab del render layer, abbiamo 2 'gruppi' di layer. Nel primo sono presenti i layer di visibilità, mentre nel secondo, quello più in basso, sono presenti i layer che vogliamo includere nel nostro layer di render.

Ho notato che se si mettono i riflessi in un layer che contiene anche un pass di texture, questi ultimi risultano 'sporcati' dalle texture. Quindi saremo costretti a dover ottenere 2 layer per le sfere, in uno (quello appena creato) andremo a metterci il color, il diffuse, l'index e uv, mentre nel secondo (lo chiameremo Palle_Raytrace) ci andremo a mettere le ombre, i riflessi e i punti di specularità. (fig 23,24)

22 Infine creeremo un Layer in cui andremo a mettervi l'Ambient Occlusion e lo Z-Depth

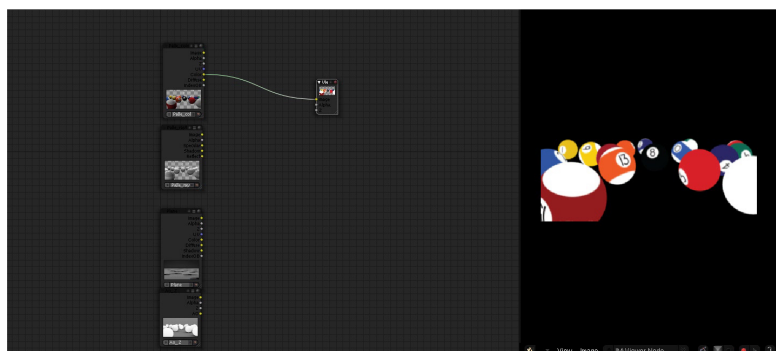


23 Come potete notare io ho disabilitato in tutti i layer il pass 'Combined' perché porta via tempo per in fase di render. Se volete avere un'immagine così com'era prima della separazione in Render Layer, basta lasciare inalterato il Layer di default (1RenderLayer). In oltre alla sinistra del nome del nostro layer vi è la possibilità, tramite un check, di render inattivo o riattivare un determinato layer, mentre alla destra potete renderizzare solo il layer attivo (Single). Bene, ora che abbiamo settato tutto come ci torna più utile possiamo renderizzare. Io tendo a fare un render al 25% delle

dimensioni per poter comporre in maniera più rapida. una volta poi che l'albero è stato creato faccio il render al 100% e ricontrollo che niente mi sia sfuggito.

Una volta che il render è finito, andiamo nel Node Editor e andiamo in modalità Composite Nodes.

Tasto dx->Input->Render Layer
Tasto dx->Output->Viewer
ora abbiamo un nodo di output, che ci permette di vedere qualsiasi output a cui lo attacchiamo, e un nodo di input. selezioniamolo e duplichimolo 4 volte e attiviamo in ogni nodo uno dei nostri layer:

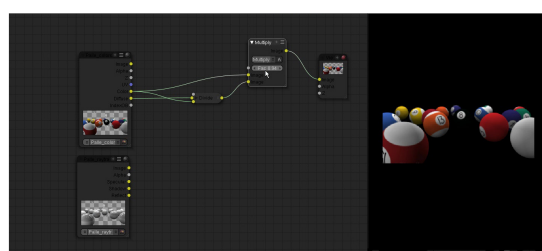
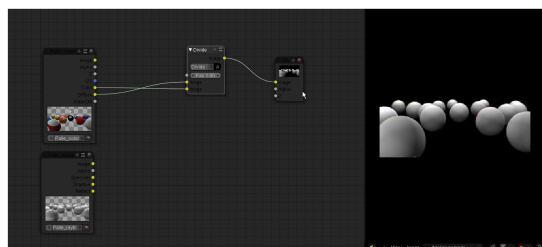


24 Come ho detto prima, io solitamente parto dal pass di colore a creare il mio albero. La prima cosa che andremo ad aggiungere al colore piatto è il diffuse delle pile. Per poter ottenere il contributo di diffuse 'pulito' dalle texture, è necessario dividere il pass di diffuse con quello di color. Quindi:

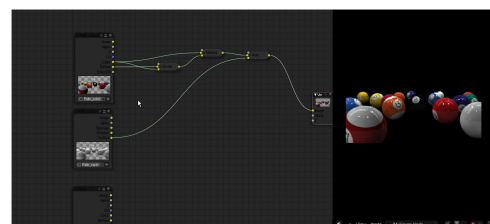
add->color->mix (settiamo su 'divide', fact 1)
input superiore: Diffuse
input inferiore: Color
In questo modo abbiamo ricavato il contributo in scala di grigi che andremo a rimoltiplicare sul color add->color->mix (settiamo su multiply; in questo caso non fa differenza in quale input andiamo a mettere le

immagini. In ogni caso dobbiamo connettere l'output del divide e il color)

Se andiamo a vedere il risultato di queste due operazioni notiamo che aumentando o diminuendo il 'fac' del nodo 'multiply', andiamo di fatto ad aumentare o diminuire l'intensità del diffuse sul colore.

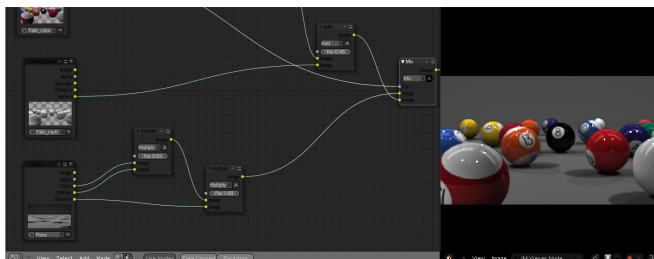


25 Ora aggiungiamo i riflessi:
add->color->mix
(settiamo su 'add')
input superiore: multiply
input inferiore: reflect (da Palle_raytrace)



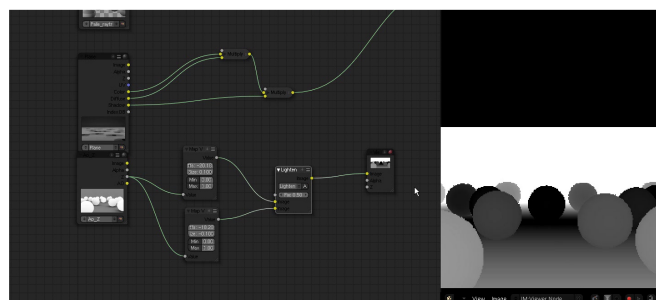
26

Lavoriamo sul piano. Moltiplichiamo il pass di color con quello di diffuse, poi sul risultato ci moltiplichiamo il pass di Shadow e in fine andiamo ad aggiungere il risultato alle nostre sfere. Quindi: add->color->mix (settiamo su multiply) input color e diffuse add->color->mix (settiamo su multiply) input Shadow (del piano) e risultato della moltiplicazione appena fatta add->color->mix (lasciamo settato su mix)



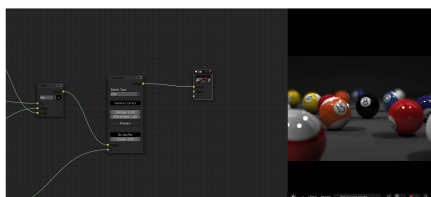
27

In fine settiamo il pass di Z. Per poterlo visualizzare dobbiamo mapparolo usando il nodo remap, decidere quale oggetto sarà a fuoco e implementare il risultato nel nodo defocus. add->vector->map value colleghiamolo con l'out Z (offs -20, size 0,1) duplichamolo e connettiamolo (questa volta con size a -0,1) add->color->mix (settiamo su Lighten) nei 2 input ci colleghiamo i 2 value di out



28

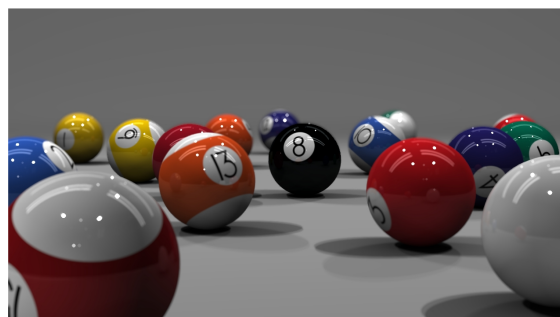
add->filter->defocus input image: la nostra immagine ottenuta fin qui input Z: l'output del nodo Lighten.



Conclusioni

Quello che abbiamo ottenuto è, di fatto, il nostro render di partenza con l'aggiunta dell'effetto di profondità. Allora a cosa serve tutto questo girar di nodi? Semplice: a poter migliorare l'immagine e averne il massimo controllo: non ci piacciono i riflessi (troppo accesi o troppo leggeri)? Basta cambiare un valore e si ottiene il risultato voluto. Le ombre sono troppo marcate? Basta aggiungere un po' di blur e il gioco è fatto. Ci si può spingere a tal punto che possiamo, con un semplice click, cambiare le texture delle nostre sfere..come? guardate (e provate) questo network (da utilizzare come color nel nostro albero...

Spero di essere stato d'aiuto a qualche d'uno nel muovere i primi passi con i nodi per il compositing. Con questa scena avete abbastanza materiale per potervi sbizzarrire e fare un po' di test, ma soprattutto capire la logica che ci sta dietro. Perché non c'è un modo universale per fare le cose, ogni situazione dev'essere interpretata. Un saluto
t0m0 aka 'Rik'





ALGORITMI DI RENDERING - PARTE 1

Uno sguardo al funzionamento dei motori di rendering

Di YotaVGA

Lo scopo di questo e dei prossimi articoli è di esporre gli algoritmi più comunemente utilizzati nel rendering. Lo studio sarà in qualche modo finalizzato ad un utilizzo grafico, senza perciò scendere eccessivamente nella matematica che ci sta sotto. Questa verrà quindi limitata il più possibile. In ogni caso non si potrebbe fare una trattazione completa senza scendere un po' nel dettaglio in qualche punto. Quindi cercherò di semplificare tutto il più possibile, ma senza perdere quei tecnicismi indispensabili per una comprensione più profonda degli algoritmi utilizzati.

In questo primo articolo si daranno alcune nozioni generiche, e si introdurranno i path tracer, ed in particolare i path tracer stocastici, che molti di voi conosceranno sotto il nome di unbiased. La scelta di trattarli per primi potrebbe

apparire azzardata, ma in realtà sono probabilmente concettualmente i più semplici, e comunque sono quelli che più naturalmente si abbinano con i concetti che vengono spiegati in questo articolo.

Raccomando una lettura del mio articolo pubblicato su BMI1 prima di leggere questo, per chi non l'avesse fatto. Alcuni concetti infatti non verranno riproposti, visto che sono già stati ampiamente trattati.

L'EQUAZIONE DI RENDERING E L'INTEGRAZIONE

Nel 1986, James Kajiya formalizza uno dei risultati più importanti di tutta la matematica che sta sotto ciò che viene spiegato in questa serie di articoli: l'equazione di rendering. Per chi fosse

interessato può guardare i link in calce, non è mia intenzione andare ad esaminare quest'equazione. Ciò che m'interessa e che ora cercherò di fare, è capire cosa vuol dire.

Per quanto possa sembrare banale, ciò che da un punto, verso un angolo, viene emesso, è esattamente quanto viene emesso dal materiale stesso (nel caso in cui si tratti di una sorgente luminosa), più quanto viene riflesso. Ciò che viene riflesso è la somma di tutti i contributi che arrivano da tutti gli angoli verso il materiale. Ogni angolo risponderà in un modo diverso, che dipende dal materiale. Questo calcolo dovrà essere eseguito per ogni lunghezza d'onda, in maniera che si consideri anche il colore, se si vuole essere davvero precisi.

Riassumendo, ed esponendo il tutto in una forma probabilmente più chiara, è necessario considerare ogni raggio, per ogni direzione, che parte da qualunque punto, e la somma di tutti quelli che partono da ogni punto sarà pari ai raggi che arrivano (da altre parti della scena) in quel punto, più le parti emesse dal punto stesso (se il materiale è una luce), meno la parte assorbita dal materiale. Il modo in cui la luce viene distribuita nelle varie direzioni dipende dal materiale.

Come si può evincere da quanto scritto prima, la somma è una componente indispensabile in un rendering corretto. Non solo bisogna sommare tutti i contributi che arrivano in un punto, ma bisogna anche sommare poi tutti i contributi che arriveranno su ogni pixel (una trattazione più completa su questo ultimo problema lo si trova nell'articolo sull'aliasing in BMI4). I componenti da sommare sono infiniti ed infinitamente piccoli, e questo tipo particolare di somma viene chiamata in matematica integrazione.

Per molti renderer il modo in cui viene eseguita l'integrazione, nella scena, stabilisce quasi interamente il modo in cui renderizzano. Più avanti verrà trattato, infatti, un particolare tipo di integrazione, detta Montecarlo, che quasi sempre viene impiegata nei path tracer stocastici.

PATH TRACING

Una classe davvero nutrita di algoritmi di rendering consiste nel lanciare da una qualche sorgente dei raggi luminosi, che poi verranno in qualche modo messi assieme per formare l'immagine finale. Questo viene chiamato path tracing. Va notato che non tutti chiamano i path tracer in questo modo. C'è chi preferisce ray tracer, o ray caster, e la confusione aumenta visto che ognuno con questi termini intende cose diverse, usandoli in contesti diversi, ma al di là della terminologia usata, il concetto alla base è sempre lo stesso. Io preferisco il termine path tracer perché è il meno equivocabile e rappresenta bene il concetto.

Secondo quanto detto dall'equazione di rendering, se fossimo in grado di lanciare infiniti raggi avremmo un rendering corretto. Ma ad oggi non ho ancora conosciuto nessuno che non si stufi aspettando un rendering infinito. La scelta quindi di quanti e quali raggi lanciare è una delle principali differenze tra i vari path tracer.

Un'altra differenza che ci permette di distinguere tra i vari path tracer è da dove partono i raggi. Potrebbero partire dalle sorgenti luminose, o dalla camera, oppure avere comportamenti differenti ed a volte bizzarri. Alcuni di questi metodi verranno esaminati nel dettaglio più avanti.

Una terza differenza è il metodo usato nell'integrazione dei raggi, e più avanti nell'articolo si vedranno due approcci di questo tipo.

Queste scelte non sono trascurabili, da questo dipende la precisione di un rendering, la quantità di rumore, e soprattutto la velocità di un renderer. Saper destreggiarsi a seconda dei casi in scelte simili influisce decisamente sulla propria efficienza nello scegliere renderer e parametri adatti ad un lavoro.

PATH TRACING STOCASTICO, INTEGRAZIONE MONTECARLO

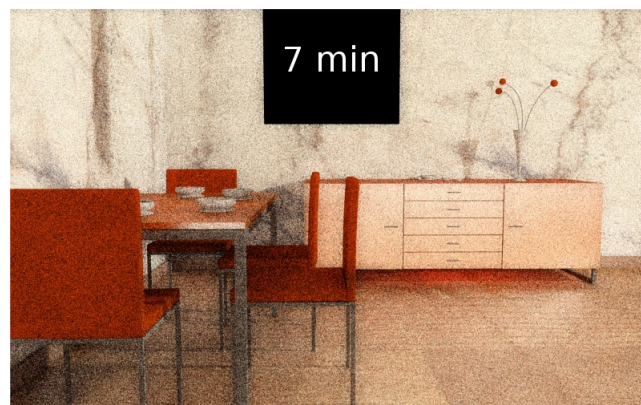
Se invece di prendere dei raggi ben precisi se ne scelgono alcuni casualmente, il rumore dovuto da aliasing si trasforma in rumore stocastico. In matematica la parola stocastico viene usata quando si ha a che fare con la probabilità. Ed in effetti i renderer stocastici sono renderer casuali. Infatti il risultato è casuale, ma come si vedrà fra poco il rendering comunque tenderà al risultato corretto.

Il problema della scelta dei raggi si manifesta, come visto poco fa, nell'integrazione. Avendo infiniti raggi, in un path tracer stocastico viene deciso quali prendere in maniera casuale (non per forza del tutto casuale, come si vedrà dopo). A questo punto bisogna fare una piccola parentesi per spiegare alcuni termini, attraverso un esempio.

Supponiamo di avere un numero molto grande di orologi. Gli orologi non mostrano l'orario esatto, o per lo meno non tutti. A questo punto facciamo la media tra tutti gli orologi che abbiamo. Otterremo un orario. Prendendo a caso gli orologi questo orario dovrebbe essere mediamente quello giusto. Se questa caratteristica è presente il sistema si dirà unbiased. Altrimenti si dirà bias. Chiaramente prendendo più orologi ci si avvicinerà sempre di più all'orario giusto se il sistema è unbiased, mentre questo non avviene con i bias (esiste un tipo particolare di tecniche bias, dette consistenti, che aumentando il numero di campioni comunque converge al valore corretto, ma ora non ci interessa).

Perciò se nell'integrazione si scelgono i raggi a caso, ma in un modo tale da rendere l'intera procedura unbiased, aumentando il numero di campioni aumenta la probabilità che l'errore si mantenga entro una data soglia. Questo è possibile tramite una tecnica chiamata integrazione Montecarlo. L'idea è in realtà molto

semplice, ma non è necessario che venga sviscerata. Gli interessati possono vedere i link in calce. Come ultima nota dico che in realtà i renderer unbiased non lo sono perché si compiono piccoli errori di calcolo, comunque decisamente trascurabili, rispetto al modello scelto.



A questo punto, poiché tutti gli integrali cercati hanno come media il valore ricercato, prendendo immagini diverse generate con questa tecnica e facendone la media, ci si avvicinerà sempre di più ad un rendering privo di rumore. Più questo

processo va avanti, e più ci si avvicinerà ad un rendering pulito. Il problema è che procedere in questo modo ha una convergenza quadratica inversa. Ovvero, per ridurre il rumore della metà bisogna stare quattro volte il tempo originale, per diminuirlo di un quarto bisogna starci sedici volte, e così via. Questo è il motivo per cui quasi immediatamente con path tracer stocastici si ottiene un'immagine dove si capisce la scena, ma prima di ottenere un'immagine pulita ci possono volere anche dei giorni.

Questo problema può essere ridotto grandemente da un altro tipo di integratore, detto quasi-Montecarlo. Questo integratore non prende raggi a caso, ma in un modo che sembra casuale. Questo effettivamente aggiunge aliasing, ma esistono schemi che lo rendono decisamente poco evidente. La quantità di rumore dei quasi-Montecarlo si riduce in maniera lineare, il che vuol dire che per dimezzare il rumore bisognerà dimezzare i tempi, perciò è decisamente molto più veloce, pur, come detto, scambiando il rumore stocastico in più noioso aliasing.

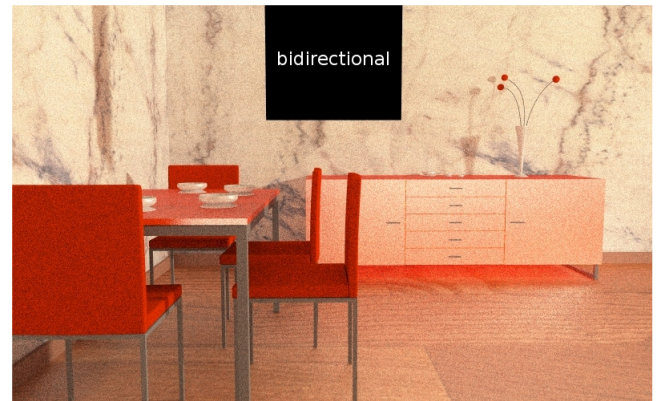
CAMERA E LUCI, PERCORSI E CAUSTICHE

Il modo più naturale di pensare ai raggi luminosi è quello di avere raggi che partono dalle sorgenti luminose ed arrivano alla camera. Questo tipo di path tracing si chiama forward path tracing, ed è tra tutti probabilmente il meno efficiente (il motivo è da ricercarsi nel fatto che gli ostacoli sono decisamente numerosi in genere quando si illumina qualcosa). Hanno comunque il vantaggio di convergere velocemente nel caso di caustiche.

Un modo più efficiente di generare i raggi, che conduce ad un risultato identico, è il backward path tracing, in cui i raggi partono dalla camera ed arrivano alle sorgenti luminose. Sebbene in genere molto più efficiente del precedente, anche questo metodo ha diversi difetti, ed in certe scene è persino più lento del forward path tracing,

specialmente in presenza di caustiche. Convergono molto più velocemente invece le penombre.

Una soluzione più intelligente a questi problemi è utilizzare quello che normalmente viene chiamato bidirectional path tracing. I raggi



partono sia dalla camera che dalle luci e si incontrano a metà strada. Sebbene l'implementazione sia abbastanza più complessa delle prime due, si tratta di una tecnica decisamente molto più efficiente, che unisce la velocità di convergenza delle due situazioni, in maniera da generare sia ombre che caustiche velocemente.

Un altro modo, più singolare, di generare raggi, che ultimamente sta prendendo molto piede, è MLT, Metropolis light transport. Questo metodo consiste nello scegliere raggi come viene fatto



nel bidirectional path tracing, ma una volta scelto il primo questo viene variato e trasformato (in maniera casuale) proporzionalmente al contributo che dà sull'immagine (di tanto in tanto, ma con probabilità piuttosto bassa, il raggio viene ricreato interamente). Quindi verranno prima di tutto esaminati i percorsi importanti, curando in seguito quelli meno importanti. Il che vuol dire un maggior rumore nelle parti scure, dove però non è possibile vederlo molto chiaramente. Non è ancora chiaro se questo algoritmo converga velocemente nel caso di molti percorsi importanti (o per lo meno non conosco nessuna ricerca in proposito).

Come si è visto è possibile lanciare raggi in maniera molto diversa, ma oltre che scegliere da dove originare i raggi, un altro modo di considerare il problema è in che modo i raggi sono distribuiti. Il jittering è un modo molto comune con cui distribuire un insieme di raggi. Sebbene casuali, i raggi vengono divisi in una griglia e da ogni elemento della griglia si lancia un raggio. Questo, come si può dimostrare, non peggiora in nessun caso la velocità di convergenza rispetto al numero di raggi.

UN'ANALISI AL RUMORE

Il rumore stocastico è ovunque. Qualunque cosa si faccia, se si usa una macchina fotografica, se si considera l'occhio umano, qualunque cosa è soggetta a rumore stocastico (comunque in quantità ridottissime). Perciò a parità tra aliasing e rumore stocastico, si nota molto di meno l'ultimo. In ogni caso, quando il rumore stocastico è molto, risulta comunque fastidioso. Imparare a controllarlo efficacemente ha quindi la sua importanza, visti anche i tempi di rendering.

Specialmente in MLT, le parti in ombra sono quelle che ricevono meno campioni, e quindi in

genere le più rumorose. Quindi se si ha fretta bisognerebbe illuminare tutte le parti considerevolmente, oppure rendere le ombre totalmente nette (che quindi non possono ricevere poca luce, non ne ricevono proprio...). Scene poi che hanno una grande velocità di cambiamento (quindi frequenze alte) avranno campioni molto diversi tra loro, e quindi convergeranno più lentamente.

I consigli dati prima su bidirectional, forward e backward path tracing si possono seguire, preferendo quasi sempre il bidirectional ray tracing. Mentre una scelta tra bidirectional path tracing ed MLT è in genere più complessa, spesso conviene preferire l'ultimo, ma non è sempre vero. Le nozioni contenute qui possono aiutare, ma alla fine l'esperienza è come sempre più preziosa.

Quanto al jittering, quando possibile è preferibile attivarlo e mettere valori piuttosto alti. Questo accelera quasi sempre la convergenza.

LINK

http://it.wikipedia.org/wiki/Equazione_di_rendering
http://en.wikipedia.org/wiki/Monte_Carlo_integration
[http://en.wikipedia.org/wiki/Ray_tracing_\(graphics\)](http://en.wikipedia.org/wiki/Ray_tracing_(graphics))



UFO alla scuola di mezzanotte

Effetti speciali e animatic alla scuola media

di Fulvio Fontana

Da quando ho iniziato ad insegnare Tecnologia alle medie ho maturato due propositi: far esplodere un pezzo dell'edificio scolastico e vedere gli alunni inseguiti da qualche creatura gigante (magari un ragno). Tutto questo, ovviamente, in grafica 3D e con l'aiuto di *Blender*, in modo da carpire qualche segreto ai maghi degli effetti speciali. Lo scorso anno scolastico è stato ricco di progetti legati a *Blender* (vedi box) e, tra una cosa e l'altra, anche l'esplosione ha preso forma all'interno di una delle scene del progetto video *UFO Halloween* (il ragno gigante dovrà aspettare). Il video voleva essere un po' il banco di prova di vari effetti speciali realizzati con *Blender* e *Voodoo Camera Tracking*.

L'esperienza si ricollegava ad un precedente video scolastico, *UFO Natale*, realizzato nella medesima sede scolastica (Varallo, VC) nell'anno 2003-04. *UFO Natale* rappresentò l'occasione di sperimentare vari trucchi, tutti rigorosamente 2D e basati in buona misura sul *chroma key*. L'animazione di alcuni elementi -quali

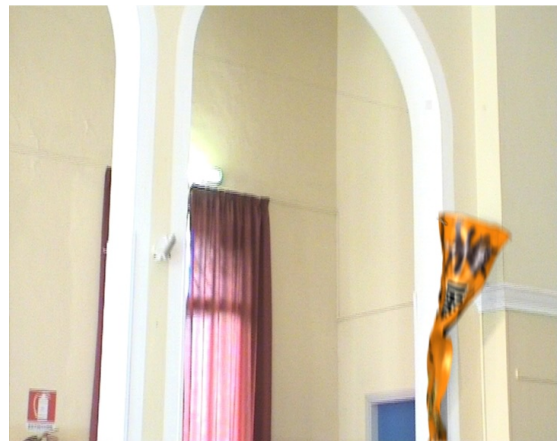
l'astronave che atterra nel piazzale della scuola e le scope volanti in stile *Harry Potter*- erano stati realizzati pazientemente tramite *morphing 2D*, applicato ad alcuni fotogrammi estratti da riprese dal vero. L'esperienza era stata abbastanza scioccante da convincermi a fare il grande passo verso la grafica 3D e scoprire quindi *Blender*. Tornare nella stessa sede scolastica dopo tre anni, con *Blender* ormai entrato a far parte della programmazione didattica, rappresentava l'occasione per riprendere l'esperimento da un nuovo punto di vista.

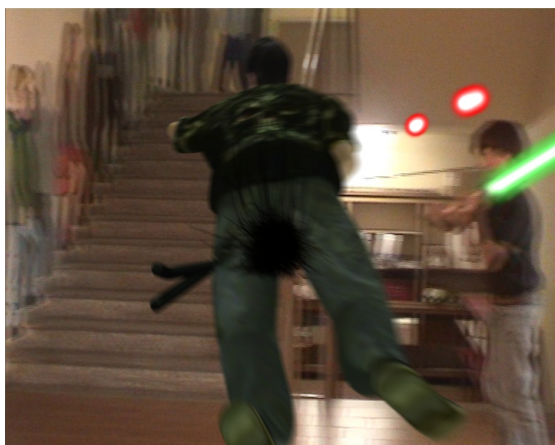
UFO Halloween

Lo spunto e il titolo per *UFO Halloween* arrivarono nel dicembre 2007, al termine della proiezione di *UFO Natale*, quando Francesca, di seconda media, suggerì la festa di *Halloween* come l'ambientazione ideale per il prequel. Trattandosi più che altro di una prova tecnica, *UFO Natale* non dava infatti molte spiegazioni agli avvenimenti narrati. All'interno della cornice

di un immaginario *TG Ragazzi* natalizio, venivano raccontati gli avvenimenti misteriosi che avevano obbligato a stravolgere la tradizionale scaletta fatta di auguri e buoni sentimenti. L'evento principale era rappresentato dall'astronave aliena atterrata sul piazzale della scuola (un modellino di *A-Wing*, da *Star Wars*), il secondo era la fuga degli ultimi quattro partecipanti al laboratorio pomeridiano *Harry Potter* (come i loro compagni prima di loro, i quattro avevano lasciato scuola non appena in grado di cavalcare decentemente le scope volanti). Sulla presenza degli alieni all'interno dell'istituto venivano fatte varie ipotesi, ma nessuna risolutiva, mentre le scope volanti venivano date semplicemente per scontate. Il *prequel* doveva tentare di dare coerenza al tutto e l'ambientazione di *Halloween*, con la possibilità di mascherare gli alieni in giro per la scuola, sembrava perfetta. In questo modo anche le scope volanti sarebbero state spiegate in termini di manufatti alieni e avrebbero costituito soltanto una delle svariate tipologie di oggetti infiltrati a scuola in occasione della festa di *Halloween*. I mascheramenti davano infatti via libera alla possibilità di portare nell'istituto tutta una serie di innocui giocattoli, come pistole e spade laser, che però si sarebbero presto rivelati perfettamente funzionanti. Allo stesso modo, gli insegnanti che avrebbero voluto indagare sui proprietari delle bizzarre maschere, indossate senza permesso prima della festa, avrebbero avuto una brutta sorpresa. Rimaneva il problema di giustificare il fine ultimo di tutti questi sotterfugi, che si concretizzò nel tentativo di fuga dei prigionieri relegati nel carcere alieno nascosto sotto la scuola.

Ripensandoci ora, il progetto si rivelò fin dalle origini troppo ambizioso: la sceneggiatura prevedeva 26 scene, metà delle quali ambientate nel carcere sotterraneo, da realizzare in 3D. Gli alieni sarebbero stati interpretati dai ragazzi, sfruttando i travestimenti e una specie di versione fantascientifica della "pozione polisucco" di *Harry Potter*, ma era necessario prevedere almeno un alieno "al naturale", da realizzare in *Blender*. La battaglia finale che





infuriava durante la festa prevedeva inoltre una serie di scene impegnative e culminava nell'irruzione di un gruppo di alieni carcerieri nell'aula in cui si erano asserragliati gli alieni criminali: era in quest'occasione che il muro divisorio tra le due aule veniva demolito da un colpo di *bazooka* in modo da creare un varco. L'eccesso di ambizioni, unito agli altri impegni (e problemi) dell'anno fecero ripetutamente slittare l'inizio del progetto. Solo verso le ultime settimane di scuola decidemmo di girare, in fretta e quasi di nascosto, qualche scena da montare in alcuni *trailer*.

Queste scene, nonostante il nuovo forzato cambio di sede, sono in lavorazione ancora adesso. Al momento sono in via di completamento due momenti della battaglia: l'inizio della sparatoria in sala mensa e l'alunno che vola per i corridoi della scuola a cavallo di una scopa. La sparatoria in sala mensa viene introdotta dallo striscione della festa, che cade colpito dai primi colpi di laser. Lo striscione e la sua caduta sono realizzati grazie al simulatore cloth di *Blender*, mentre i laser sono dei semplici materiali *halo*.

L'alunno sulla scopa spara a sua volta laser all'impazzata, ma viene fermato da un suo compagno armato di spada laser. Questa seconda scena è ricalcata su una sequenza della battaglia su *Endor* de *Il ritorno dello Jedi*, quando *Luke* taglia con la spada laser lo stabilizzatore di una moto volante imperiale che si schianta su una sequoia. La sostituzione della moto con la scopa crea un collegamento con il precedente *UFO Natale* e rappresenta il tentativo di superare i limiti del vecchio effetto 2D delle scope (poco riuscito, nonostante la fatica). In *UFO Halloween* infatti sia la scopa sia l'alunno sono stati realizzati e animati in *Blender*. Trattandosi del mio primo esperimento del genere il risultato in realtà è modesto e i difetti non mancano, soprattutto a livello di animazione del personaggio (anche il rilievo fotografico dell'alunno è stato fatto in fretta e in condizioni di luce infelici).

Grazie a *Voodoo Camera Tracking*, però, in queste nuove sequenze l'inquadratura è

finalmente libera di muoversi: possiamo così seguire la caduta dello striscione digitale fino alle due ragazze che duellano a colpi di spade laser, oppure svoltare l'angolo insieme all'alunno volante, per scoprire l'aggressore in agguato. Se a tutto questo si aggiunge la magia del pulsantino *motion blur*, che miscela ulteriormente

il movimento reale e quello digitale, il passaggio dalla forzata fissità degli effetti speciali della precedente esperienza è realmente liberatorio e permette di "toccare con mano" alcuni cambiamenti nel linguaggio cinematografico e televisivo contemporaneo. Guardando il dietro le quinte di *Jurassic Park* (1993) scopriamo come all'epoca l'unione di effetti speciali al computer e riprese in movimento era ancora qualcosa di assolutamente pionieristico e ottenuto solo grazie ad una serie di marcatori accuratamente posizionati sul set (e cancellati successivamente). Vediamo *Spielberg* che racconta di come non volle piegarsi a quello che gli venne inizialmente presentato come un limite tecnico invalicabile: quando i disosauri digitali entravano in scena, la telecamera doveva fermarsi. Spesso tendiamo a darlo per scontato, ma l'idea che qualcosa del genere sia oggi liberamente a disposizione dei nostri PC domestici è veramente incredibile!

The midnight school project

La lunga gestazione del trailer di *UFO Halloween* non è però dovuta solo all'eccesso di ambizioni, alla mia inesperienza e all'ennesimo trasferimento, ma anche ad un nuovo progetto video che ha preso vita con le classi seconde della *Scuola Media di Valduggia*. Il progetto ha visto ancora una volta protagonista *Blender*, ma



in una fase di lavorazione completamente differente. Questa volta l'obiettivo era realizzare la puntata di una *sit-com* per partecipare al concorso *Storie di classe* organizzato da *Disney Channel* e *Giffoni Experience* in collaborazione con *La Fabbrica*. I limiti erano precisi: l'episodio doveva

durare al massimo sei minuti (il modello di riferimento era la serie *Quelli dell'intervallo*) e la scadenza era fissata sabato 11 aprile 2009. Questa volta l'eccesso di ambizioni di *UFO Halloween* doveva essere rigorosamente bandito, altrimenti non saremmo riusciti neppure a partecipare al concorso, però volevo comunque utilizzare *Blender*, anche per presentare il programma ai nuovi alunni.

I dubbi sull'opportunità di buttarsi in un'ennesima iniziativa video, non avendo neppure terminato quella precedente, erano parecchi. Lo spunto era però interessante e i nuovi alunni sembravano incuriositi. Un paio di sessioni di *brainstorming* in classe confermarono l'impressione, riuscendo a fissare i punti fondamentali della nostra *sit-com*. Innanzitutto l'ambientazione doveva essere notturna (opposta quindi all'esperienza scolastica quotidiana, tipicamente diurna) e gli alunni che la frequentavano sarebbero stati ispirati ai mostri dei *film horror* classici. A ripensarci ora sembra strano, ma le caratteristiche "particolari" degli alunni, che sarebbero diventate un po' il punto di forza del progetto, nacquero solo per motivare lo strano orario delle lezioni (all'inizio pensavamo ad un più banale corso serale in cui accadevano cose misteriose). Il modello di riferimento era quindi diventato *La famiglia Addams*, ma ambientata a scuola. Non credo sia un caso, tra l'altro, che la serie dedicata alla più famosa

famiglia *horror* della TV si apra con la visita dell'ispettore scolastico, preoccupato del fatto che i due giovani *Addams* non frequentino la scuola pubblica. La scuola porta in questo modo una ventata di normalità all'interno di villa *Addams*, dando il via ai continui confronti con il "mondo esterno" che costituiscono l'anima della serie.

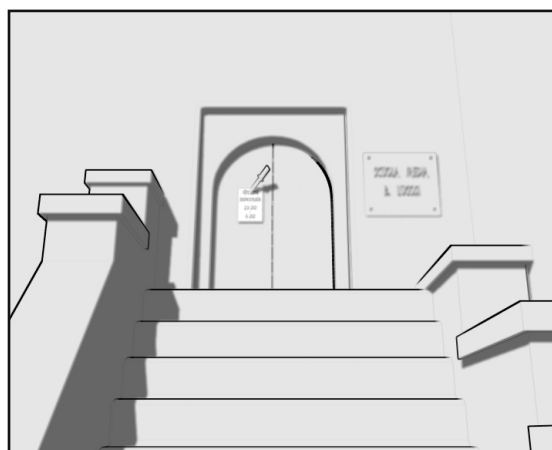
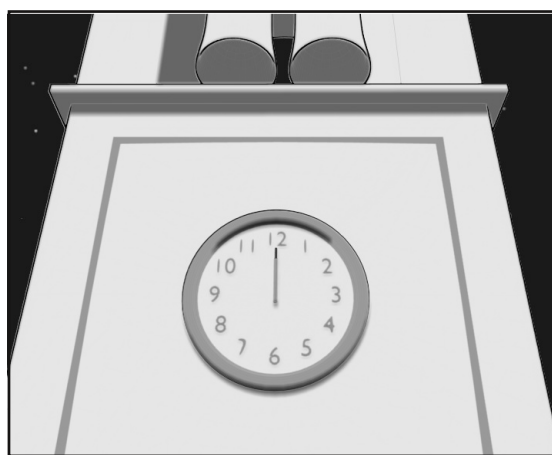
Nel nostro caso, la svolta *horror* ci permetteva di rileggere i tradizionali stereotipi scolastici (il bullo, la nuova arrivata, il secchione) in chiave mostruosa, osservandoli "dall'esterno", estremizzandoli e ridendoci sopra. Così la mitologica Steno -sorella di Medusa- diventa la bulla della classe, che si diverte a spaventare Francesca, la nuova arrivata (a sua volta meno inoffensiva di quanto sembri), mentre i professori si rifugiano dietro le spesse lenti di occhiali da scienziato pazzo che servono a tutto fuorché a vederci meglio.

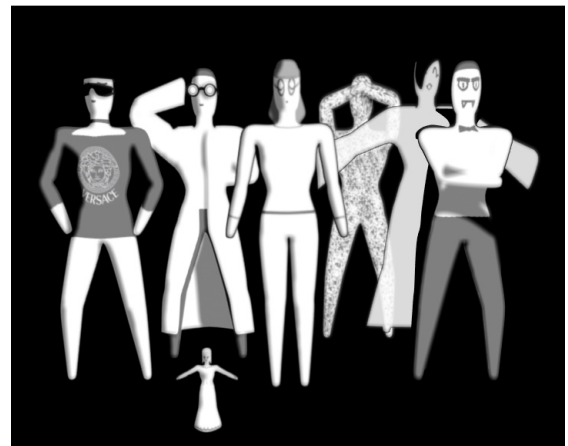
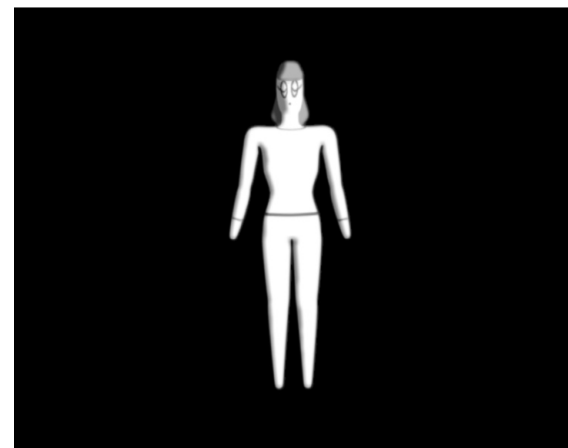
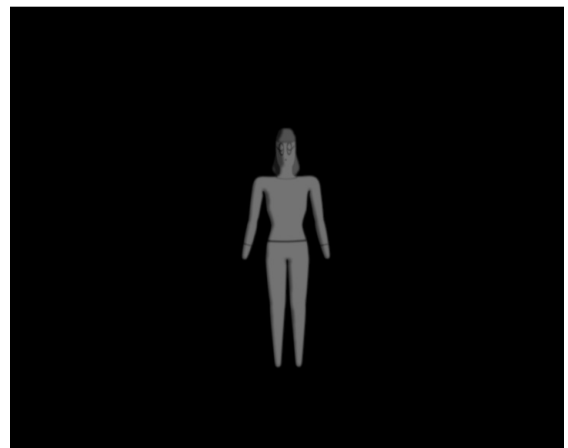
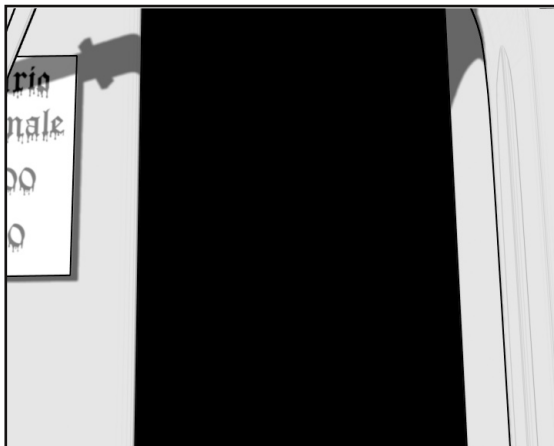
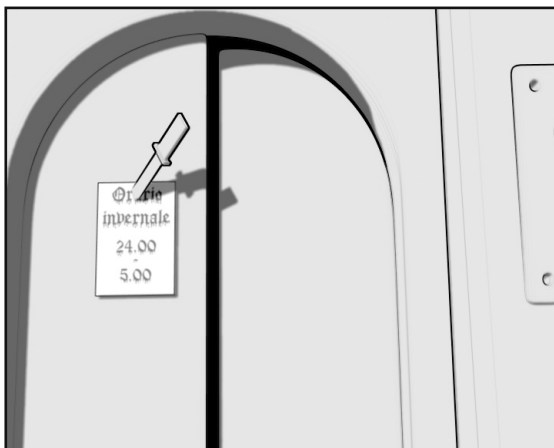
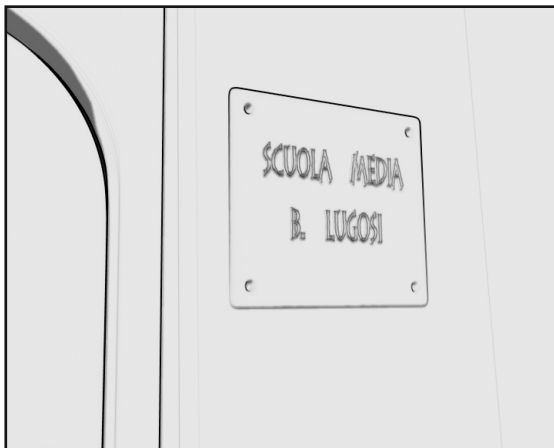
Cosa c'entra *Blender* in tutto questo? Moltissimo: anche per evitare di ricadere negli stessi errori del passato, preferii infatti pianificare accuratamente tutto, rimandando la possibilità di girare effettivamente la puntata tra le "varie ed eventuali". Il concorso infatti chiedeva innanzitutto un'idea di trama precisa, che poteva essere arricchita dalla sceneggiatura, dalla definizione dei personaggi, dalla scenografia e da un video. Il premio principale era per la trama e consisteva -tra l'altro- nella possibilità di vedere realizzato il proprio progetto grazie all'aiuto di una troupe di professionisti, mentre per gli altri elaborati erano previsti dei premi speciali. Riuscire a definire tutto, girare le scene e montare il video in tre mesi -con i vincoli dalla normale *routine* scolastica- era praticamente impossibile, per cui l'obiettivo principale diventò quello di definire una trama divertente e convincere la giuria che il tutto fosse effettivamente realizzabile. In sintesi, bisognava realizzare uno *storyboard* dettagliato dell'episodio, meglio ancora se in forma di *animatic*, per calcolare bene i tempi.

La possibilità di utilizzare efficacemente *Blender* per realizzare l'*animatic* fu rafforzata dai progetti del *BOMP* e in particolare dal progetto 2, *La*



L'animatic della sigla





zucca. Mi sembrava una buona idea per raccontare una storia senza dover affrontare la sfida di un'animazione vera e propria (sia per motivi di esperienza, sia di tempi). L'idea di una sit-com sullo stile dei classici *horror* anni '30-'40 ha portato come naturale conseguenza la scelta di realizzare l'animatic in bianco e nero, usando un semplice sistema di nodi. Questo ci ha ovviamente permesso anche di risparmiare un bel po' di lavoro e di tempi di *rendering*, riuscendo ad ottenere -speriamo!- una grafica in grado di aggiungere qualcosa all'atmosfera generale del prodotto. In fondo una delle caratteristiche fondamentali delle prime pellicole *horror* era proprio l'uso di un bianco e nero stilizzato, che permette ancora oggi di apprezzarle nonostante fondali ed ambientazioni a volte evidentemente "finti".

Software libero

Dal punto di vista tecnico, uno dei punti fondamentali del progetto era utilizzare esclusivamente programmi liberi. L'occasione era interessante perché il concorso consentiva la realizzazione di elaborati di vario genere (testi, immagini e video). La scelta fu ulteriormente rafforzata dall'insegnante di musica, **Bruno Tasso**, che, appassionato utente *Slackware*, accettò di comporre e realizzare la colonna sonora dell'episodio con *LMMS* (*Linux MultiMedia Studio*).

L'animatic invece è realizzato quasi completamente in *Blender* su *Ubuntu 8.04*. Questo è vero non solo per la modellazione e l'animazione delle singole scene, ma anche per il montaggio finale (sonoro compreso). *Blender* si è infatti dimostrato la scelta più immediata anche come editor video. Solo per intervenire su alcune tracce audio -in particolare per le dissolvenze di entrata ed uscita- è stato necessario utilizzare *Audacity*. Allo stesso modo, è stato necessario un passaggio in *CineLerra* per le transizioni video della sigla di chiusura. Il fido *GIMP* è stato invece indispensabile per le *texture* e per alcuni ulteriori ritocchi sui singoli fotogrammi, resi possibili dalla scelta di esportare le animazioni

come sequenze di PNG.

GIMP e *Scribus* sono stati i programmi fondamentali per realizzare e confezionare il materiale destinato alla stampa (trama, sceneggiatura, schede dei personaggi e scenografia). In più, *GIMP* si è rivelato prezioso anche per adattare le immagini per il web, rivelandosi quindi il programma più "trasversale" del progetto.

Dopo un po' di test e qualche tentennamento, *DVD Styler* è stato scelto per realizzare il *DVD* vero e proprio. Ancora una volta c'è stato naturalmente bisogno dell'intervento di *GIMP* e di *Blender* per le immagini dei menù. La masterizzazione vera e propria è stata infine affidata a *K3B*. Dagli esperimenti fatti fino ad ora, il risultato in termini di compatibilità è piuttosto buono: solo uno dei lettori su cui abbiamo avuto modo di provare i dischi ha dato dei problemi.

Midnight web

Sul sito del progetto sono al momento disponibili l'animatic della sigla di apertura e il brano musicale completo, insieme a qualche altra curiosità e le schede dei personaggi. Il sito ha avuto il doppio ruolo di servire come punto di riferimento durante la realizzazione del progetto e di permettere agli alunni di utilizzare le basi di HTML acquisite nei primi mesi dell'anno in un contesto reale.

Ora che il tutto è finalmente arrivato a conclusione, l'intenzione sarebbe quella di mettere in rete l'animatic dell'intero episodio, ma per questo dovremo almeno aspettare gli esiti del concorso. L'unico rimpianto è che -per ragioni di prudenza e *privacy*- non potremo pubblicare in rete i risultati delle "prove costumi", il che è un peccato, perché alcune interpretazioni sono state piuttosto sorprendenti. La scelta di privilegiare l'animatic rispetto al video finito era comunque stata pensata fin dall'inizio anche per poter pubblicare in rete il prodotto senza problemi. Nel frattempo non ci resta che attendere e sperare!

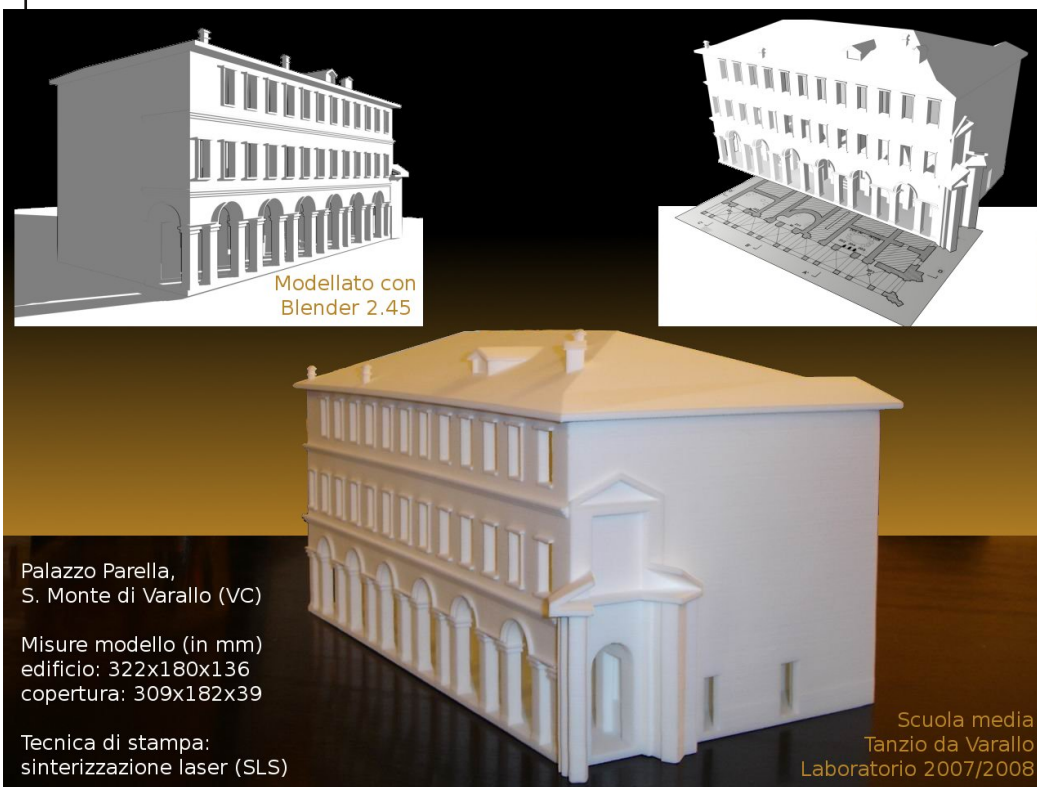
Non solo video: a scuola di prototipi 2

Sul secondo numero di BMI avevamo avuto l'opportunità di segnalare un'altra iniziativa dell'anno scolastico 2007/08 legata a *Blender*: la ricostruzione di alcuni edifici del complesso del *Sacro Monte di Varallo* (VC) finalizzata alla stampa solida. All'epoca avevamo stampato un solo edificio (mediante tecnica FDM) adattando un modello 3D già esistente, realizzato negli anni precedenti per un video. Il progetto vero e proprio doveva ancora iniziare.

Modellare appositamente per la stampa solida si sarebbe rivelato presto un'impresa impegnativa, ma alla fine ha dato i suoi risultati. Nella foto è visibile il modello stampato di *Palazzo Parella* (scala 1:100) realizzato da due ragazze di terza media che lo hanno portato all'esame come

lavoro finale di Tecnologia. La stampa è stata possibile anche in questo caso grazie all'associazione culturale **Nuova Regaldi**. Il nuovo modello era grande circa il doppio del precedente: per ragioni sia tecniche sia di costo, abbiamo quindi scelto la sinterizzazione laser, più economica e in grado di lavorare su dimensioni maggiori (con la tecnica FDM avremmo dovuto spezzare il modello in due parti).

L'edificio è stato parzialmente ricostruito anche all'interno: togliendo la copertura è infatti possibile vedere le volte a crociera del portico e del piano terra, che hanno rappresentato la sfida più impegnativa (bastava un vertice "scucito" e il tutto veniva considerato non prototipabile da *MiniMagic*, il software gratuito utilizzato per il controllo).



Due curiosità: le due ragazze che duellano nella ripresa dello striscione di *UFO Halloween* facevano anche parte del gruppo impegnato nella ricostruzione degli edifici del *Sacro Monte*; una delle due autrici del modello stampato in realtà non ha potuto portarlo all'esame orale per il semplice fatto che io quel giorno me lo ero dimenticato a casa!

Risorse e link citati nell'articolo

Storie di classe su Giffoni Experience: <http://www.giffoniff.it/pageview2.php?i=139>

BOMP: <http://www.blenderonlinemoviesproject.com/?cat=3>

The midnight school project, il sito: <http://midnightschool.altervista.org/>



Crea una Statua

Nuovo contest per scegliere la prossima copertina

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 6 di BMI è **"Statua"**.

Quindi mostrateci la vostra "Venere di Milo", la vostra "Pietà", la vostra "Statua della Libertà", i vostri "Bronzi di Riace" o, ancora meglio, una statua di vostra creazione. Ricordate che le statue possono essere di diversi materiali, non solo pietra, marmo, ma anche legno, bronzo, ceramica, ecc.

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: 1754*2480 px o superiore, basta rispettare la proporzione A4.

- Le immagini possono essere postate come Wip su Blender.it (sezione immagini) o Kino3d (sezione open lab), con la dicitura [BMI6] prima del titolo.

- I lavori finiti devono essere mandati all'email anfeo@libero.it **entro il 19/07/2009**

- 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.

Nascita... nel futuro

Dario Caffoni

(vincitrice del 3° Blender Tutorial Contest)



Western Park

Stefano Zanardi



Casa sulla Luna

Fabris89



Angolo di luce

Paolo Testiccioli





Stanza da letto
Andrea Boggio "boggio95"