

BLENDER

Magazine
Italia

ARTICOLI

BIACT

Perché crediamo nella
formazione di qualità

I CORTI DI BLENDER ITALIA

Open Movie
work in progress

TUTORIALS

NODEVEMBER

Magie e sparizioni
utilizzando i nodi

CREARE UN SERVIZIO DA TE

Puntata finale:
materiali e rendering

PYTHON

Realizziamo un
countdown in 3D

MORPHING

Trucchi e segreti di
un metodo alternativo

INTERVISTE: - Mike Pan, esperienze di un blenderista
- Alessio Cassaro, il pittore degli affreschi virtuali
- Luca Di Cecca, la genesi di Arturo e il gabbiano



Immagine di Luca Di Cecca

Scene 0 INTRO

Scene 0 INTRO

Editoriale



Eccoci di nuovo!

L'avevamo promesso e siamo riusciti a raccogliere notizie e tutorial per impacchettare il nuovo numero del BMI. A ridosso del Natale questo è il regalo che ci facciamo: vi raccontiamo delle attività di blender.it e dell'Associazione Blender Italia; abbiamo voluto sapere tutto della creatività e dei lavori di alcuni ottimi blenderisti, che riconoscerete dalle loro artistiche produzioni e non ci siamo fatti mancare degli eccellenti tutorial, che di certo non troveremmo in rete.

La nuova era di Blender 2,8x è in arrestabile ascesa: si susseguono incessanti le versioni aggiornate, potenziate e corrette, donandoci la possibilità di concentrare tutta la nostra attenzione sulla creatività, con puro divertimento.

Anche l'attività della redazione del BMI continua incessantemente e se non ci sentite durante gli altri periodi è perchè non smettiamo mai di darci da fare.

Insomma, buona lettura e buon Blender a tutti!

www.magazine.blender.it

Blender Magazine Italia

Numero 23
Anno 2019

Direttore Responsabile
Cristina Ducci

Redazione
Alfonso Annarumma
Andrea Campagnol
Alessandro Passariello
Enrico Lui
Riccardo Giovanetti
Carin Belmonte

Grafica e impaginazione
Alessandro Passariello

Copertina impaginazione
Andrea Campagnol

Siti
blender.it
magazine.blender.it

Contatti
magazine@blender.it

Software utilizzati
Blender
The Gimp
Krita
LibreOffice



Il Blender Magazine Italia è mantenuto e gestito dall'Associazione Blender Italia perché possa essere sempre disponibile per tutti gli utenti.

Associazione Blender Italia

In questo numero

News:

- BLENDER 2.8 - Pag.4
- BLENDER ANNUAL CONFERENCE 2019 - Pag.8
- BLENDER ITALIA CONFERENCE 2019 - Pag.9
- PREMIO ASIFA A BLENDER - Pag.10
- BLENDER ITALIA CONTEST 2019 - Pag.11
- BLENDER ITALIA COMPIE 17 ANNI - Pag.12
- BLENDER COMPIE 25 ANNI - Pag.13

Associazione:

- L'ASSOCIAZIONE E I BIACT - Pag.14
- I CORTI DI BLENDER ITALIA - Pag.16

Tutorial:

- COUNTDOWN ANIMATO IN PYTHON - Pag.18
- NODEVOMBER - Pag.26
- MORPHING HACK! ANIMARE SENZA FISICA - Pag.34
- CREARE UN SERVIZIO DA TE' PARTE III - Pag.38

Interviste:

- ALESSIO CASSARO - Pag.48
- LUCA DI CECCA - Pag.52
- MIKE PAN - Pag.54
- NUCCIO CANINO - Pag.56

Gallery:

- Le immagini scelte dallo Staff – Pag.68

Blender Magazine Italia non rappresenta una testata in quanto viene aggiornato senza alcuna periodicità. Non può considerarsi un prodotto editoriale ai sensi della legge n. 62 del 07/03/2001 in quanto è un prodotto di informazione. Gli autori non hanno alcuna responsabilità sui contenuti dei siti in collegamento, sulla qualità o correttezza dei dati. Essi si riservano la facoltà di rimuovere le informazioni, fornite da terzi, ritenute offensive o contrarie al buon costume. Le immagini sono correlate agli argomenti di cui si scrive.

Alcune sono provenienti da Internet e quindi valutate di pubblico dominio. Qualora i soggetti proprietari fossero contrari alla pubblicazione, non dovranno far altro che segnalarlo in modo da poter procedere ad una rapida eliminazione.





News

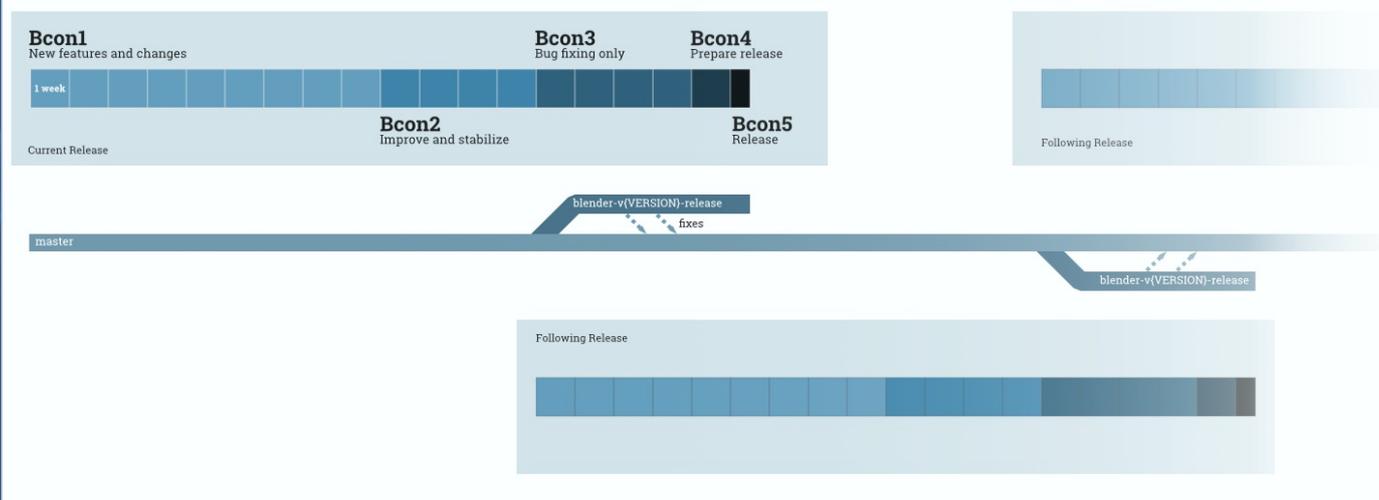
di Riccardo Giovanetti

Bentornati in questa rubrica di notizie e novità su Blender ed il mondo della computer grafica ed animazione.

BLENDER 2.8

Dopo il rilascio della versione 2.80 alla fine di luglio, a fine novembre è stata rilasciata la versione 2.81 di Blender ed ai primi di dicembre una versione correttiva di alcuni bug, la 2.81a. E' stato inoltre introdotto un nuovo ciclo di sviluppo che prevede la lavorazione della versione 2.83 Alfa durante le fasi di completamento dell'attuale versione 2.82 Beta. Questo nuovo approccio rende più efficiente il processo di sviluppo e permetterà di rilasciare una nuova versione di Blender ogni tre mesi, circa.

Blender Release Cycle

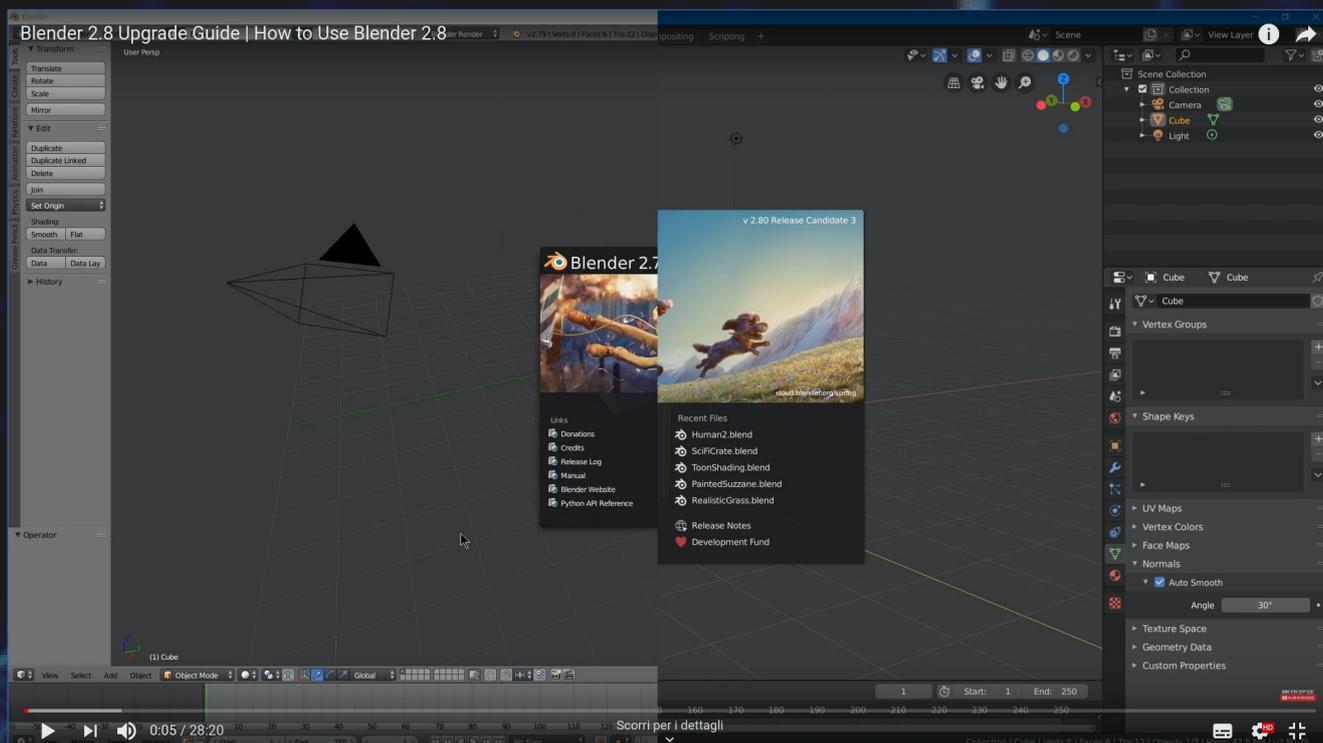


Per maggiori informazioni: <https://code.blender.org/2019/10/blender-release-cycle/>

Le novità introdotte con la versione 2.80, e successivamente sviluppate nella 2.81, sono veramente tante e per una descrizione più approfondita si rimanda alle note di rilascio ufficiali sul sito www.blender.org. Questa è una breve ed incompleta sintesi delle molte novità introdotte nelle versioni 2.80 e 2.81.

Interfaccia

Blender 2.80 presenta un'interfaccia utente ridisegnata per migliorare il flusso di lavoro ed il processo creativo. Sono stati introdotti un nuovo tema scuro e un set di icone più moderno. Il clic con il pulsante sinistro del mouse è ora il metodo di selezione predefinito. Il menu Quick Favorites consente di salvare ed accedere rapidamente agli strumenti utilizzati di frequente.



Are di lavoro (Workspace)

Una serie di aree di lavoro predefinite consentono di iniziare rapidamente a scolpire, dipingere, lavorare sui materiali o eseguire il motion-tracking e possono essere personalizzate per creare il proprio ambiente di lavoro efficiente.

Workbench

Grazie al nuovo motore di rendering Workbench la visualizzazione della scena in 3D è ottimizzata per supportare attività come il layout della scena, la modellazione e la scultura. E' possibile scegliere quali elementi visuali sovrapporre (Overlay) alla vista attuale, come il wireframe per esempio, e funziona anche in vista completamente renderizzata con Eevee e Cycles.

Strumenti e gizmo

La vista 3D e l'editor UV dispongono ora di nuovi strumenti interattivi e Gizmo, insieme a una nuova barra degli strumenti. Ciò rende più semplice per i nuovi utenti iniziare a utilizzare Blender e per gli utenti esistenti scoprire e utilizzare strumenti che in precedenza richiedevano oscure combinazioni di tasti. Oltre ai gizmo per gli strumenti, vari elementi come luci, fotocamera e l'immagine di sfondo di composizione ora hanno maniglie per regolare la loro forma o altri attributi.

Eevee

Eevee è un nuovo motore di rendering in tempo reale basato sulla fisica (sulla fisica che sta alla base del comportamento della luce n.d.r.). Funziona sia come renderer per creare singole immagini, sia come visualizzatore in tempo reale nel processo di creazione di asset. Ha funzionalità avanzate come volumetria, riflessi e rifrazioni, subsurface scattering, ombre morbide e di contatto, profondità di campo, motion blur e bloom.



Animazione 2D

Il Grease Pencil è ora un sistema di disegno e animazione 2D completo. Questa integrazione senza precedenti di strumenti 2D in un ambiente 3D consente di creare concept art, storyboard e animazioni di livello superiore.



Cycles

Cycles ora offre funzionalità come Cryptomatte, ombreggiatori BSDF per volume, capelli e subsurface scattering con tecnica random walk. Sono state apportate molte ottimizzazioni tra cui il rendering combinato CPU e GPU, tempi di avvio e rendering OpenCL molto migliorati e supporto CUDA per scene che non rientrano nella memoria GPU.

Altre novità introdotte con la versione 2.80: Principled Volume Shader; Bevel Shader; Ambient Occlusion Shader; IES Lights; Disk Area Lights; Vector Displacement.

La versione 2.81 prosegue lo sviluppo sull'onda della 2.80 ed introduce diverse importanti novità quali la revisione degli strumenti di scultura, il supporto per il ray tracing NVIDIA RTX in Cycles, il denoising di Intel Open Image, un miglior outliner, un nuovo file browser e molto altro.

Per ulteriori approfondimenti: <https://www.blender.org/download/releases/2-81/>



Blender Development Fund

Il sistema di finanziamento dello sviluppo di Blender, il Blender Development Fund, ha visto quest'anno la partecipazione di nomi illustri dell'industria come NVIDIA, AMD, EPIC GAMES, ADIDAS, UBISOFT, ed altri che hanno contribuito con somme anche ingenti, come gli 1.2 milioni di Dollari in tre anni. Un risultato importante che permetterà di investire in molti ambiti dello sviluppo di Blender ma soprattutto, come dice Ton Roosendaal: "Grazie alla sovvenzione effettueremo un investimento significativo nella nostra organizzazione del progetto per migliorare l'onboarding, il coordinamento e le migliori pratiche per la qualità del codice. Di conseguenza, prevediamo che un numero maggiore di collaboratori del settore aderirà ai nostri progetti".



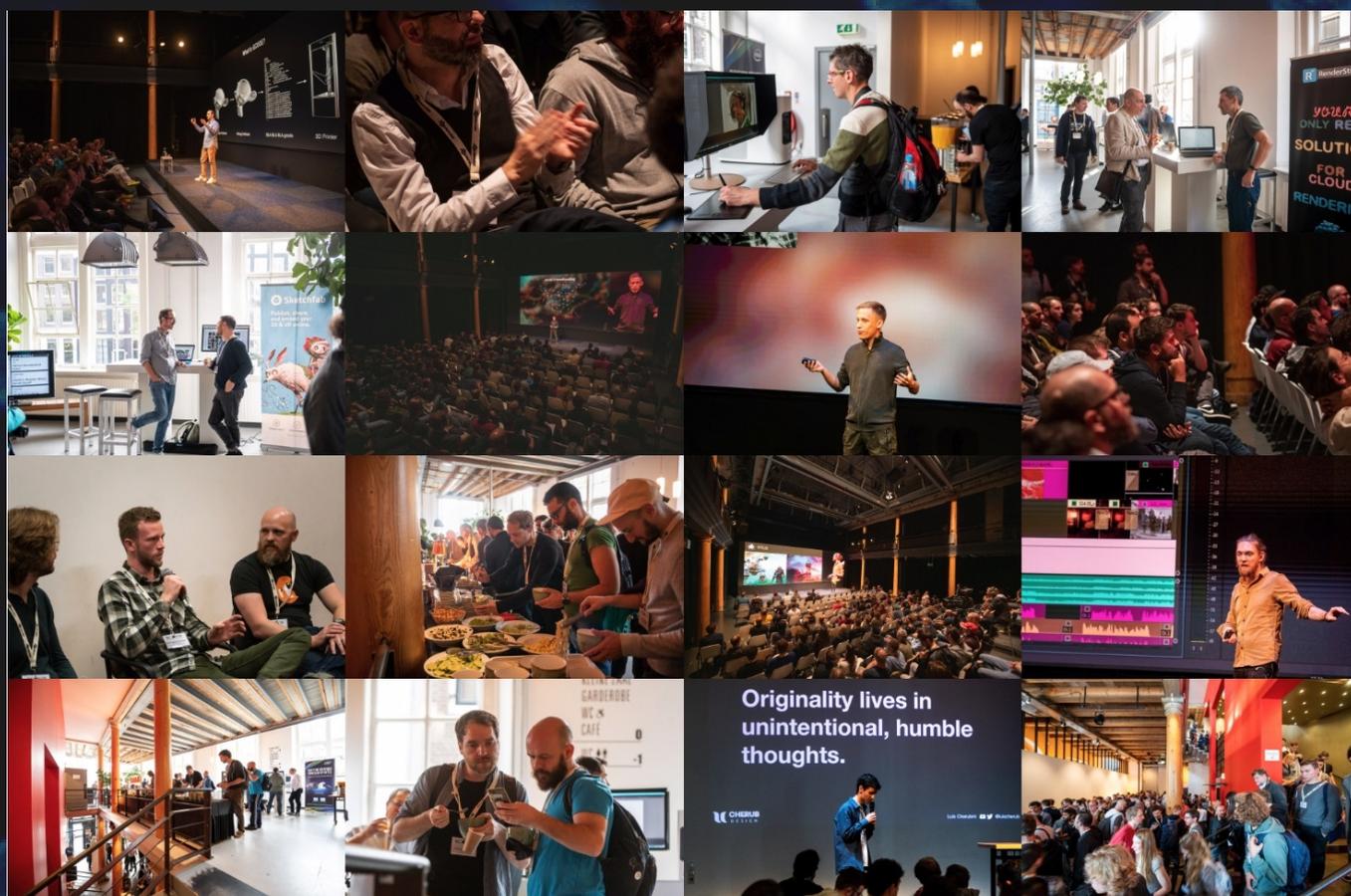
BLENDER ANNUAL CONFERENCE 2019

Anche quest'anno si è tenuta ad Amsterdam la conferenza su Blender organizzata dalla Blender Foundation e per poter accomodare il gran numero di partecipanti è stata scelta una nuova struttura per ospitare l'evento e tutte le attività collaterali, il Compagnie Theater nel centro della città.

La qualità degli interventi è cresciuta notevolmente e dimostra l'aumentata attenzione che il mondo rivolge a questo incredibile programma ed alla sua comunità così affiatata e coinvolgente.

Tutte le registrazioni delle presentazioni e delle sessioni sono visionabili sul canale YouTube di Blender: https://www.youtube.com/playlist?list=PLa1F2ddGya_8Wzpjwu1EtiS8E1Exm82S

<https://conference.blender.org/>



BLENDER ITALIA CONFERENCE 2019

L'edizione 2019 della conferenza annuale, la seconda dopo quella di Foligno nel 2018, si è tenuta presso la sede della Fondazione Marco Besso nel centro di Roma. La partecipazione è andata oltre le aspettative degli organizzatori e sono intervenuti in tanti per illustrare i loro progetti e realizzazioni che gravitano attorno al programma software Blender.

Maggiori informazioni sulle conferenze di Blender Italia sono reperibili sul sito dell'associazione:

<https://www.blender.it/bic/>

Le registrazioni degli interventi sono visionabili sul canale YouTube dell'associazione:
https://www.youtube.com/playlist?list=PLb4uLLumrU5J8k_ASQr5-sma_qafUZGrX



PREMIO ASIFA A BLENDER

La International Animated Film Society, ASIFA-Hollywood, ha assegnato a Ton Roosendaal e al software Open Source Blender il Ub Iwerks Award al 46 ° Annie Awards ™. Il Premio Ub Iwerks riconosce i progressi tecnici che hanno un impatto significativo sull'arte o sul settore dell'animazione. Questa è la prima volta che questo premio viene assegnato a un progetto software gratuito / open source.

<https://www.blender.org/news/ton-roosendaal-and-blender-to-receive-asifa-hollywood-ub-iwerks-award/>



BLENDER ITALIA CONTEST 2019

Proseguono anche quest'anno i contest artistici mensili grazie alla partecipazione di tanti concorrenti entusiasti di mettersi alla prova e divertirsi. Invitiamo quindi chiunque abbia passione e voglia di cimentarsi a partecipare. Questi sono i temi ed i vincitori al momento della pubblicazione:

- BIC 01 "L'isola misteriosa" - Akenaton
- BIC 02 "Tank Battle" – Poly Low (Luca Ferro Auditore)
- BIC 03 "Oggetto impossibile" - MirkoC (Stampa modello 3D)
- BIC 04 "Alieni pericolosi" - Enrivector
- BIC 05 "E' primavera!" - Akenaton
- BIC 06 "Vita da pompiere" -
- BIC 07-08 "Viaggio nel corpo umano" - Davide.C88
- BIC 09 "L'inebriante mondo del vino" - Akenaton
- BIC 10 "Computer vintage" - Alessandro Passariello
- BIC 11 "Colonie spaziali" - Akenaton



03/12/2019

Blender Contest "Natale in stile libero 2019"



04/11/2019

Blender Italia Contest "Colonie Spaziali"



09/07/2019

Blender Italia Contest "Viaggio nel corpo umano"



04/06/2019

Blender Italia Contest "Vita da Pompiere"



01/03/2019

Blender Italia Contest "Oggetto Impossibile" 2019



04/02/2019

Blender Italia Contest "Tank Battle"



04/11/2018

Blender Italia Contest "Città Apocalittiche"



04/10/2018

Blender Italia Contest "Trasporti Speciali"



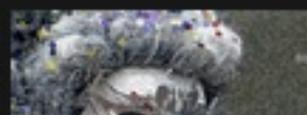
04/07/2018

Blender Italia Contest "Un'attenta progettazione"



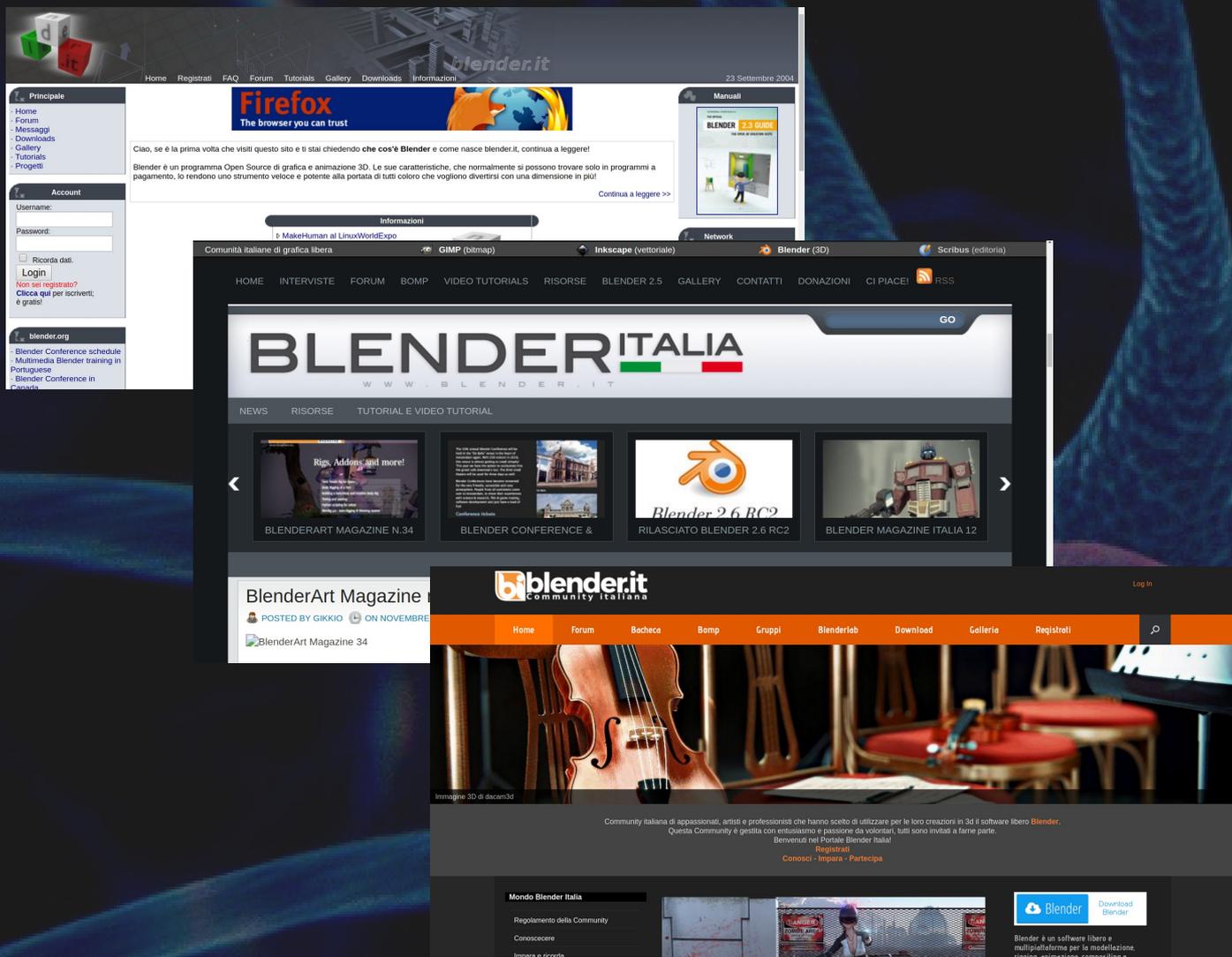
04/06/2018

Blender Italia Contest "Sotto la pioggia"



BLENDER ITALIA COMPIE 17 ANNI

Quest'anno Blender Italia compie 17 anni, tanto è trascorso dal 4 ottobre del 2002 quando veniva aperta al pubblico la prima homepage del sito blender.it. Tante cose sono cambiate nel frattempo ed il mondo è cambiato ma ci si è tenuti al passo, almeno questo è quello che speriamo di essere riusciti a fare.



BLENDER COMPIE 25 ANNI

Il 2 gennaio 1994 viene eseguita la prima versione di Blender una data memorabile - ora contrassegnata come il compleanno ufficiale di Blender - è definita da un vecchio backup della workstation SGI Indigo2 di Ton.

Dopo la creazione dei primi file, ci sono voluti 15 mesi prima che il software fosse pronto per la produzione come strumento interno nello studio di animazione NeoGeo nei Paesi Bassi. Blender è stato quindi rilasciato pubblicamente online il 1° gennaio 1998 come freeware SGI. Le versioni Linux e Windows sono state rilasciate poco dopo. Dopo lo scoppio della bolla di Internet nei primi anni 2000, il codice sorgente è diventato disponibile sotto GNU GPL il 13 ottobre 2002, grazie a una spettacolare campagna di raccolta fondi, che segna l'inizio di Blender come la famosa suite di creazione 3D gratuita e open source che è oggi.

The collage consists of three overlapping screenshots:

- Top Screenshot:** The Blender website homepage. The navigation menu includes: About Blender, News, Download, InfoCenter, Support, Community, Blender Foundation, E-shop. The main banner features the Blender logo and the text "BLENDER Open source 3D graphics creation". A secondary banner for "orange Extended DVD Edition" is visible with a "Buy now!" button.
- Middle Screenshot:** A section of the website with a filmstrip of 3D models. The navigation menu includes: Features & Gallery, Download, Education & Help, Community, Development, e-Shop. A "Download Now Blender 2.37" button is present, along with the "orange the open movie project" logo.
- Bottom Screenshot:** The Blender 2.69 software interface. The navigation menu includes: Features, Download, Support, Get Involved, About, Store. The interface shows a 3D model of a mechanical structure. A "Blender 2.69" download button is overlaid on the interface, along with the text "3D creation for everyone, free to use for any purpose. Free Software, great community, and 100% open." and "What's New? Older versions".

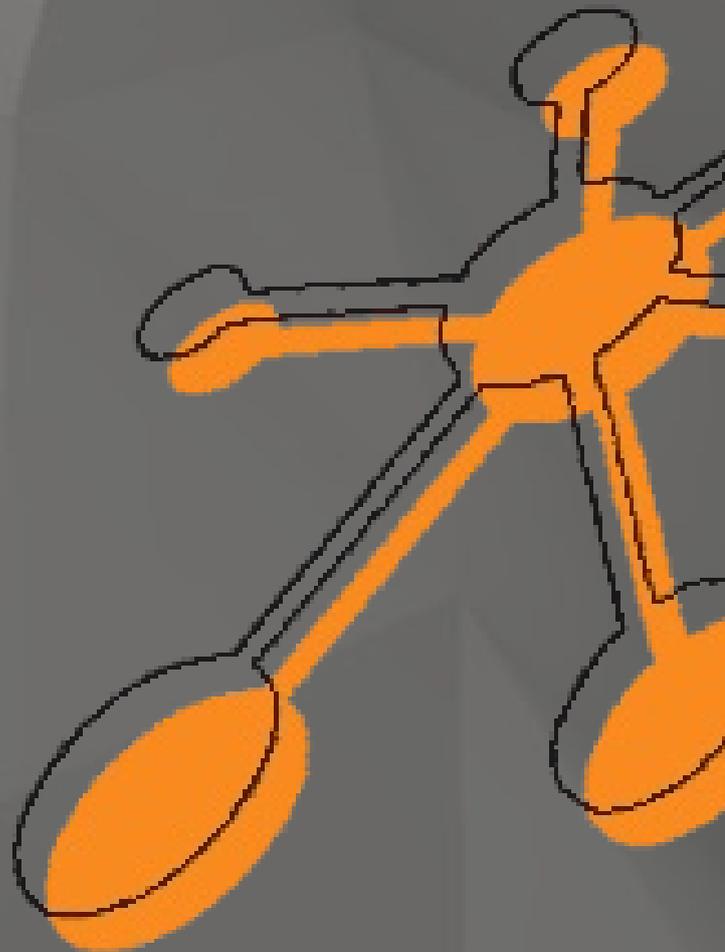


L'Associazione Blender Italia e I BIACT di Carin Belmonte

Nei primi giorni dello scorso ottobre Ton Roosendaal ha annunciato l'abolizione dei BFCT (Blender Foundation Certified Trainer), sistema che certificava i Formatori da parte della Blender Foundation.

L'associazione Blender Italia ha avuto indicazione di far nascere un sistema nazionale proprio che sostituisca in modo efficace il BFCT, portando alla nascita dei BIACT (Blender Italia Association Certified Trainer).

L'associazione Blender Italia, per dare maggiore credibilità ai professionisti che operano nel settore della formazione sul territorio nazionale, oltre ad aver istituito il titolo di Formatore accreditato, potrà d'ora in poi conferire la certificazione di BIACT a quei Formatori accreditati che dimostreranno capacità e competenze in Blender e di aver prodotto del materiale video, testi, etc, oltre aver maturato esperienza nel campo dell'insegnamento su Blender.



Chi è un Formatore BIACT?

Un Formatore certificato BIACT (Blender Italia Association Certified Trainer) è un Associato, impegnato nel campo della formazione, il quale si candida per diventare Formatore certificato BIACT ed ottiene con successo la certificazione da parte della Associazione Blender Italia.



Certificato BFCT | BIACT - Responsabile
Area Formazione

Francesco
Andresciani



Certificato BFCT | BIACT

Francesco Arena



Certificato BFCT | BIACT

Andrea Coppola

Sopra i primi BFCT italiani certificati dalla Associazione Blender Italia quali BIACT, Blender Italia Association Certified Trainer.



I corti di Blender Italia

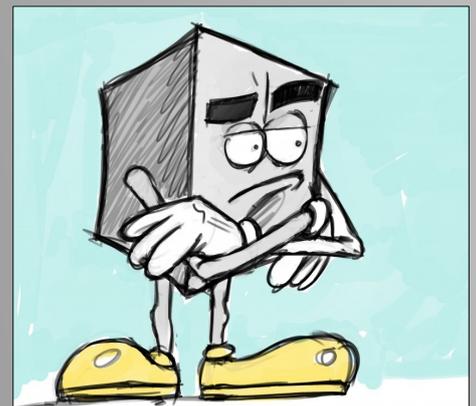
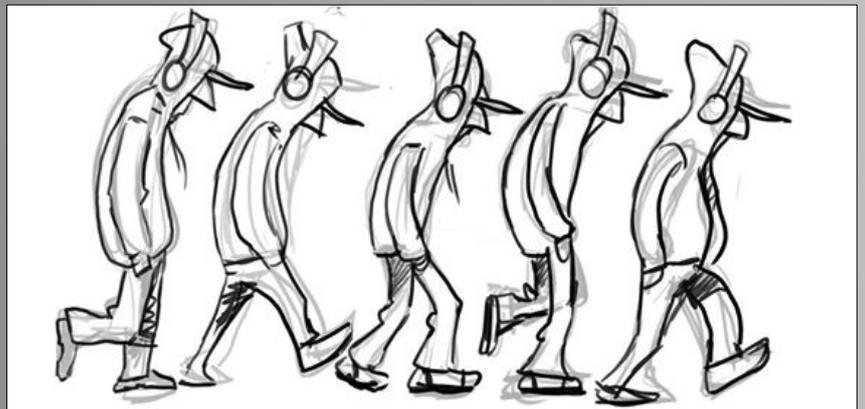
Open Movie work in progress di Carin Belmonte

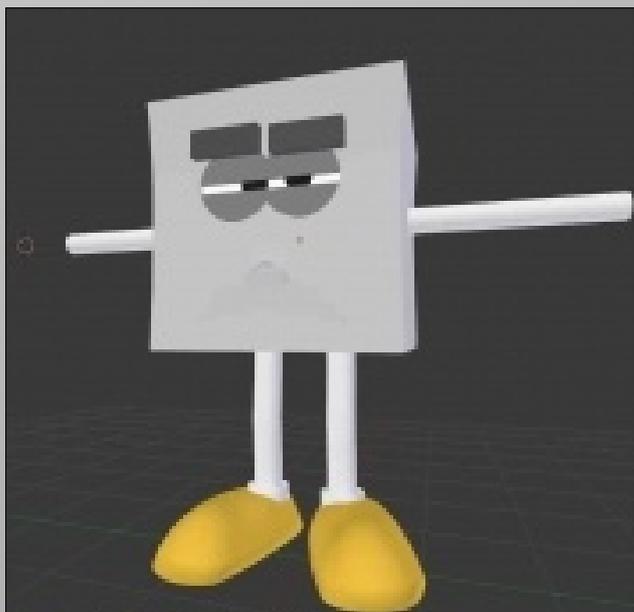
Salve Blenderisti, oggi voglio parlarvi di un progetto nato all'interno di Blender.it che, cooperando con l'Associazione Blender Italia, punta alla realizzazione di un open movie.

L'ambizioso scopo che perseguiamo come associazione è quello di riuscire a realizzare un cortometraggio da presentare alle future edizioni del Suzanne Awards, manifestazione tenuta ogni anno ad Amsterdam in concomitanza con la Blender Conference.

Il progetto è aperto a tutti gli utenti del sito Blender.it, ed è iniziato con la richiesta fatta agli utenti di scrivere e pubblicare sul sito un racconto inedito.

Tra le storie pubblicate, è stata scelta quella che in seguito ad un sondaggio ha ricevuto maggior numero di voti.

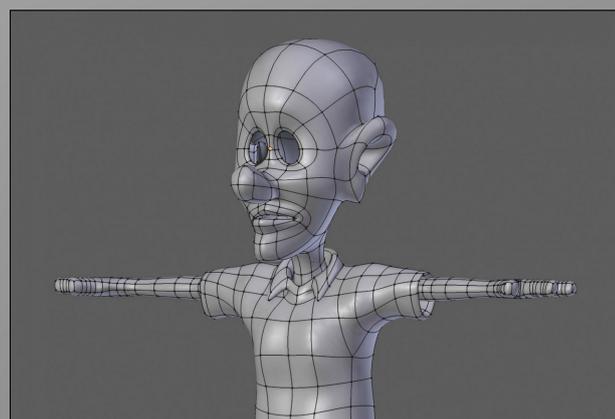




Dopo la scelta della storia, è iniziata la ricerca di tutte quelle figure necessarie alla realizzazione del corto, come ad esempio regista, direttore tecnici, disegnatori, modellatori, esperti di rigging etc etc.

Non essendo una casa di produzione ci siamo imbattuti da subito in una serie di problemi, dovuti all'inesperienza, ma che non sono stati affrontati come ostacoli, ma come stimoli, cercando di informarci e di formare affinché tali intoppi si potessero nel tempo superare.

Attualmente il corto in fase di realizzazione è "Cubes Game" tratto dall'omonimo racconto scritto da me (Carin Belmonte), con la regia di Michele Zampieri, Attualmente Cubes Game è l'unico Corto in realizzazione, ma a breve sarà possibile proporre nuovi corti da realizzare.





Countdown Animato in Python

Di Alfonso Annarumma

Introduzione

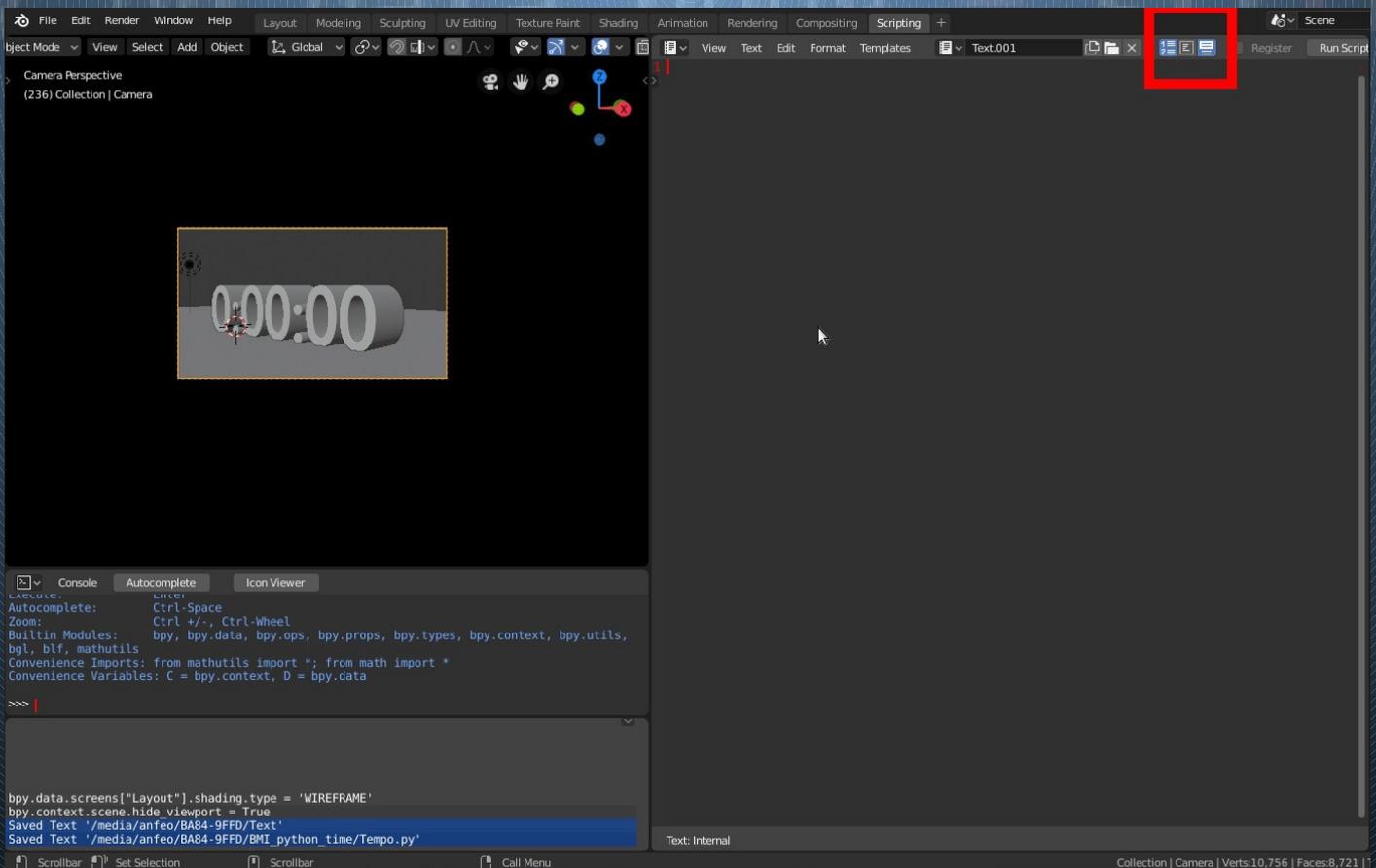
La programmazione in Python all'interno di Blender può agevolare in mille modi diversi e, per rendere più interessante possibile questo tutorial, ho deciso di creare direttamente un esercizio di medio livello, che potesse attirare più persone possibili e spingerle ad approfondire questo aspetto di Blender, molto spesso trascurato perché considerato solo per tecnici e matematici. Con un po' di impegno, e seguendo qualche tutorial di base, si possono fare cose molto interessanti e quella che segue è una di queste.

Il tutorial che vi presento è rivolto ad una tipologia di utenti che già conoscono blender e che hanno già familiarità con i linguaggi di programmazione, anche se una conoscenza di base di Python è quasi basilare, ma vi invito a seguire l'esercizio anche perché mostrerò dove reperire le informazioni che ci servono online e come applicarle ai nostri script.

Testo 3d

Diamo per scontato che sappiamo inserire un oggetto di tipo Text in scena e rinomiamolo in "Testo". In caso di difficoltà possiamo trovare il file pronto in download a questo link.

Entriamo nel workbench Scripting e aggiungiamo un nuovo file di testo nell'editor apposito. Ricordiamoci di attivare le proprietà "Line Numbers" e "Syntax HightLight" come mostrato nel riquadro rosso dell'immagine sottostante. In questo modo attiveremo la numerazione delle righe e la colorazione della sintassi dello script che ci aiuterà nella leggibilità.



Copia e Incolla

Banalmente possiamo copiare e incollare le seguenti linee di codice e quindi premere Run Script per vedere cosa succede in scena:

```
import bpy  
  
ob = bpy.data.objects["Testo"]  
  
ob.data.body = "PROVA"
```

Analizziamo le righe di codice e capiamo cosa è successo.

Python è un linguaggio interpretato come l'html o il php, in pratica non va compilato per generale un programma. Quindi quando premiamo Run Script viene eseguita ogni riga, una alla volta, e ogni riga fa qualcosa.

La prima riga non fa altro che importare il modulo chiamato bpy, cioè BlenderPy, in pratica è una libreria che contiene tutti i comandi che ci servono per far dialogare Python con Blender. Difatti nella seconda riga andiamo a creare una variabile che possiamo chiamare ob, nella quale, tramite il modulo bpy, andiamo a inserire l'oggetto Testo che abbiamo creato. Ci sono diversi modi per raggiungere un oggetto presente nel file .blend su cui stiamo lavorando. Se ne conosciamo il nome, la maniera più facile è arrivarci tramite il database, infatti scriveremo:

```
bpy.data
```

Mettendo il punto vicino al nome bpy (che abbiamo importato) accediamo a varie sezioni, una di queste è il database data.

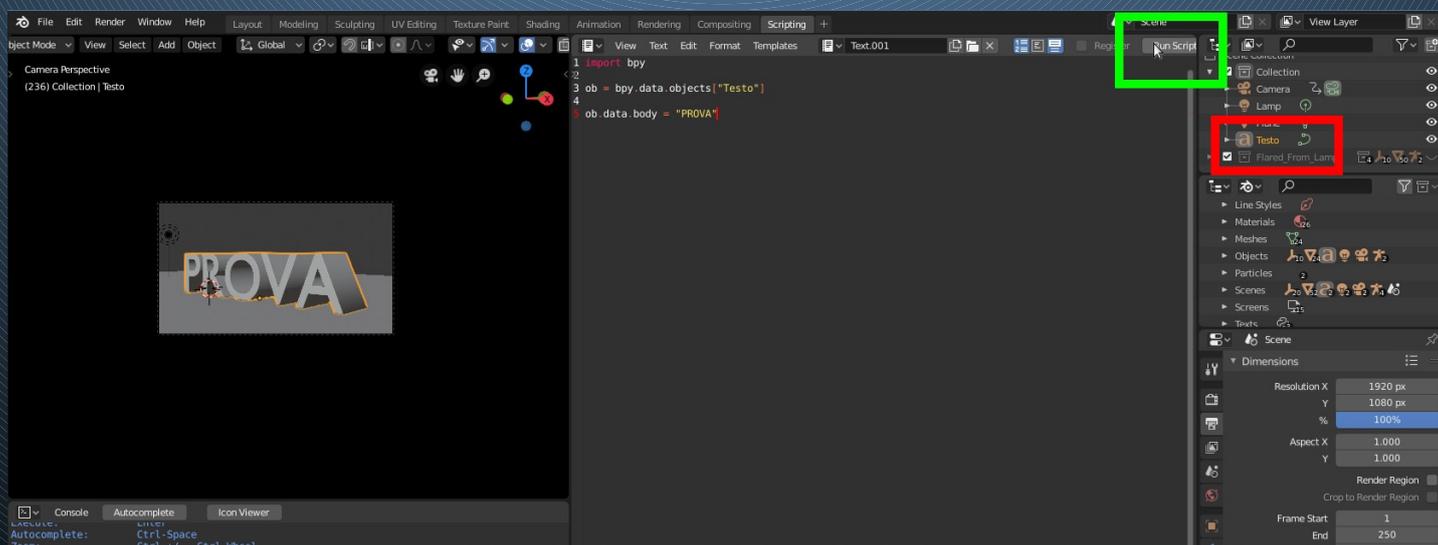
Mettendo ancora un punto vicino a data accediamo al database e scegliamo il contenitore di oggetti objects, quindi con le parentesi quadre andiamo ad inserire il nome dell'oggetto racchiuso tra virgolette.

Usare le virgolette "" equivale a dire a Python che quello che scriviamo all'interno di esse è una stringa, cioè un dato di tipo testo.

Assegnando un oggetto ad una variabile, nel nostro caso ob, possiamo accedere all'oggetto direttamente da essa.

Per accedere al campo dove inserire il testo dell'oggetto 3D text, dobbiamo navigare nell'oggetto attraverso il suo data e raggiungere il campo body, nel quale assegnamo una stringa "PROVA".

Facendo Run Script il testo nella 3D view diviene PROVA.



L'algoritmo

Con tre righe siamo riusciti a capire come andare ad editare tramite lo script il testo nella 3D view, ma come ricaviamo l'orologio che va all'indietro?

Sviluppare l'algoritmo di un programma è proprio quello, quindi va capito come fare. Le soluzioni possono essere diverse, possiamo semplicemente creare tanti testi per ogni numero e farli apparire in sequenza... mmm troppo lungo... possiamo sottrarre dal numero dieci un numero per ogni secondo che passa... ma c'è una strada più facile.

Cosa sappiamo?

- Sappiamo che possiamo far eseguire lo script ad ogni cambio di frame usando gli handlers.
- Sappiamo che possiamo prelevare il numero del frame corrente, il frame rate e il frame finale, cioè quanti frame è composta l'animazione.
- Possiamo quindi ricavare i secondi dai frame con la semplice operazione Fram/Framrate.
- Esiste una libreria che ha una funzione che restituisce l'orologio in formato HH:MM:SS se le diamo in pasto il numero di secondi.

Tutto qua, il resto dello script si scrive da solo.

Nota: Google è stato molto di aiuto per la realizzazione di questo script, non conoscevo la libreria datetime, alle volte basta veramente poco per togliersi da un impiccio, non sono un esperto di Python, ma ogni volta mi sorprende cosa può fare questo linguaggio di programmazione.

L'orologio

```
import bpy
import datetime

testo = datetime.timedelta(seconds=500)

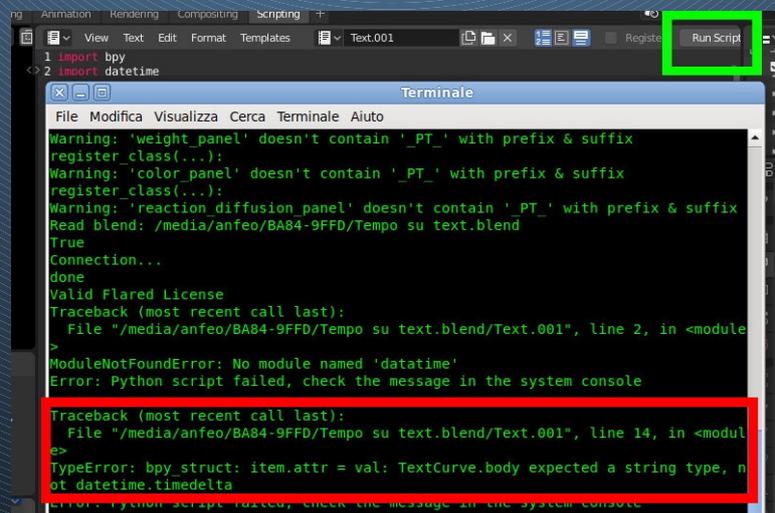
ob = bpy.data.objects["Testo"]
ob.data.body = testo
```

Aggiungiamo due righe al nostro script, la prima serve ad importare la libreria datetime, che contiene una funzione che andiamo a richiamare nella seconda riga, la funzione si chiama timedelta().

Riconosciamo una funzione dalle parentesi dopo il nome. Di solito tra le parentesi si inseriscono le variabili che servono alla funzione per eseguire i calcoli. Infatti andiamo ad usare la variabile seconds dove le assegnamo (per ora) 500 secondi per fare un test.

La funzione restituisce un valore che andiamo a immagazzinare nella variabile testo.

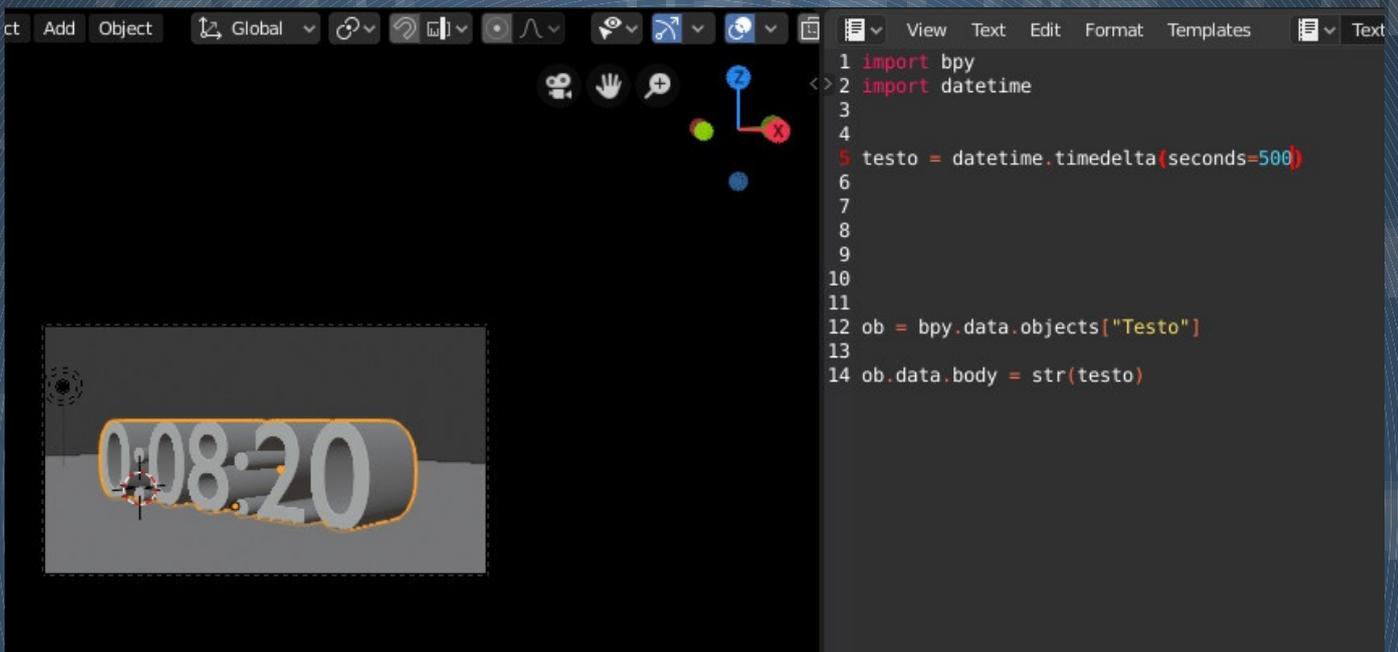
Proviamo a lanciare lo script e vediamo che ci restituisce un errore nel terminale.



L'errore ci dice che il campo `body` accetta solo elementi di tipo stringa. Quindi dobbiamo convertire la variabile `testo` in un formato stringa. Sostituiamo l'ultima riga dello script con questa:

```
ob.data.body = str(testo)
```

Abbiamo semplicemente richiamato una funzione nativa di Python che si chiama `str()` che in pratica va a trasformare un numero in una stringa. Se lanciamo di nuovo lo script avremo il risultato della figura sotto.



In pratica abbiamo trasformato i 500 secondi in formato orologio, cioè 8 minuti e 20 secondi.

Due Calcoli

```
import bpy
import datetime

fr = bpy.context.scene.frame_current
fps = bpy.context.scene.render.fps
end = bpy.context.scene.frame_end

testo = datetime.timedelta(seconds=500)

ob = bpy.data.objects["Testo"]
ob.data.body = str(testo)
```

C'è da fare qualche calcolo per ricavare il countdown in base al numero di frame in scena e il frame corrente in cui ci si trova.

Aggiungiamo le righe evidenziate al codice, queste non fanno altro che andare a pescare nel nostro file .blend le informazioni che ci servono, la prima preleva il frame corrente e lo mette nella variabile `fr`, la seconda va nelle impostazioni di render a prendere il parametro di frame rate (`fps`) e lo mette nella relativa variabile `fps` e l'ultima va a dirci qual'è il numero dell'ultimo frame, cioè sappiamo (presupponendo che lo start sia a 1) che la nostra animazione durerà quei frame lì, messi nella variabile `end`.

```
import bpy
import datetime

fr = bpy.context.scene.frame_current
fps = bpy.context.scene.render.fps
end = bpy.context.scene.frame_end

end_s = end/fps
secondi = end_s - (fr/fps)

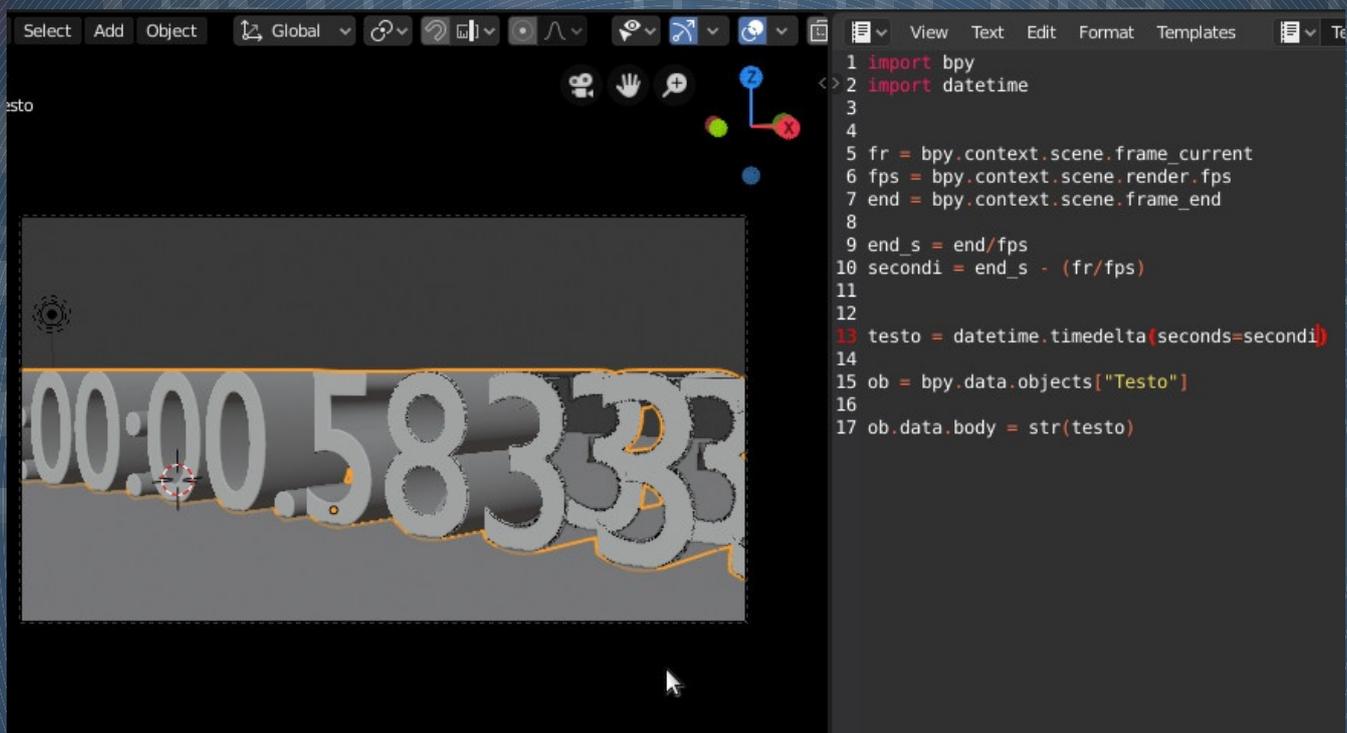
testo = datetime.timedelta(seconds=secondi)

ob = bpy.data.objects["Testo"]
ob.data.body = str(testo)
```

Le due righe aggiunte sopra sono i calcoli che vengono eseguiti:

Prima di tutto ricaviamo la durata totale in secondi dell'animazione, facendo Frame diviso Framerate, usando il numero totali di frame dell'animazione e gli `fps`. Il tutto va nella variabile `end_s`.

Quindi otteniamo i secondi restanti sottraendo ad `end_s` il secondo corrente, ricavato dalla formula usata sopra, ma questa volta usando il frame corrente invece che i frame totali, e mettiamo tutto nella variabile `secondi` che daremo in pasto al `timedelta()`. Premiamo Run Script e vediamo che succede in figura 5.



C'è un altro problema, il valore in secondi che si ottiene con la divisione non è intero, ma reale (cioè con la virgola). Anche in questo caso ci viene in aiuto una funzione di Python che si chiama `int()` che non fa altro che trasformare il numero in un intero, in pratica ci dà indietro solo quello che c'è prima della virgola. Non resta che applicarla ai secondi mentre li passiamo al `timedelta()`.

```
testo = datetime.timedelta(seconds=int(secondi))
```

Handlers

Blender dispone di certi contenitori chiamati Handlers che hanno al loro interno delle liste di funzioni che vengono eseguite in determinati momenti.

Le API di Blender sono ben strutturate e soprattutto ben documentate, qua possiamo trovare molte informazioni sugli handlers e soprattutto esempi pratici:

<https://docs.blender.org/api/current/bpy.app.handlers.html>

Il primo esempio è quello che fa proprio al caso nostro, in pratica esegue una funzione prima di passare da un frame ad un altro.

```
import bpy
import datetime

def tempo(pippo):
    fr = bpy.context.scene.frame_current
    fps = bpy.context.scene.render.fps
    end = bpy.context.scene.frame_end

    end_s = end/fps
    secondi = end_s - (fr/fps)

    testo = datetime.timedelta(seconds=int(secondi))

    ob = bpy.data.objects["Testo"]

    ob.data.body = str(testo)

bpy.app.handlers.frame_change_pre.append(tempo)
```

Racchiudiamo tutto quello che abbiamo scritto (a parte gli import) sotto una funzione.

Per creare una intestazione si usa questo comando:

```
def nome_funzione (variabili):
```

Poi tutto quello che è sotto la funzione va indentato, cioè spostato con il tasto TAB.

Nel nostro caso abbiamo creato una funzione che si chiama tempo e passato una variabile pippo. La variabile in realtà non viene usata, ma la passiamo perché le funzioni che carichiamo negli handlers devono averne una, anche se fittizia.

Alla fine dello script carichiamo la funzione con un append nell'handlers frame_change.

A questo punto se facciamo Run Script la funzione va nell'handlers e se premiamo Play sulla time line vediamo il tempo che cambia.

Conclusioni

Il percorso per chi inizia è lungo e tortuoso, però vedere che con poche righe ci si salva una mezza giornata di lavoro può diventare appagante per gli sforzi compiuti.

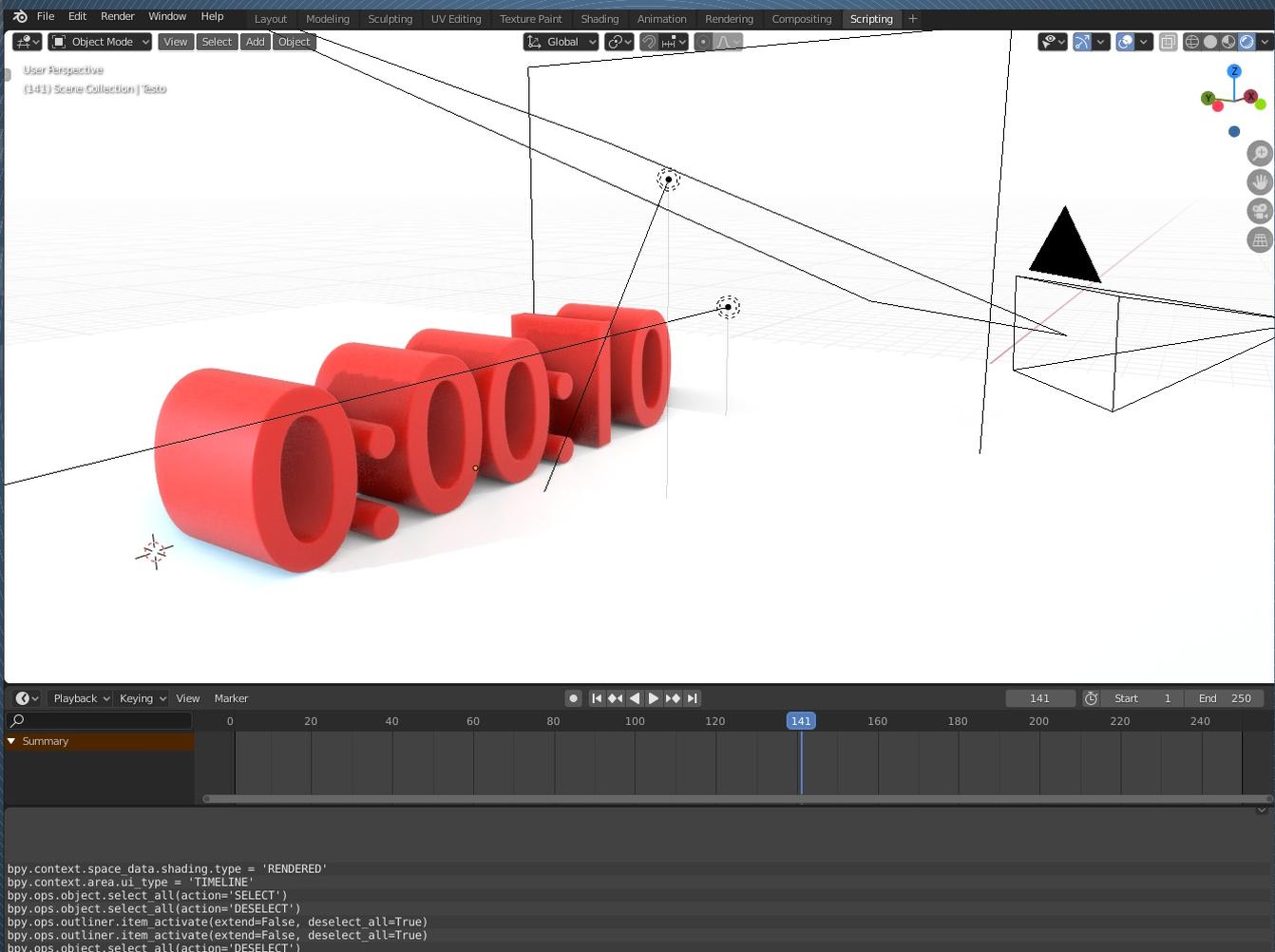
L'esercizio presente elementi che non sono proprio basilari, l'uso delle funzioni e gli handlers possono disorientare. Il mio consiglio è di dare per buono quanto scritto, ma avere l'accortezza di approfondire autonomamente gli aspetti che risultano più incomprensibili e scoprire tante altre possibilità che offrono questi semplici comandi.

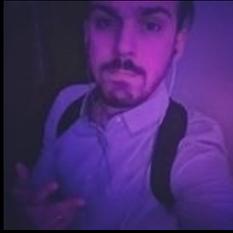
Conclusioni

Il percorso per chi inizia è lungo e tortuoso, però vedere che con poche righe ci si salva una mezza giornata di lavoro può diventare appagante per gli sforzi compiuti. L'esercizio presenta elementi che non sono proprio basilari, l'uso delle funzioni e gli handlers possono disorientare. Il mio consiglio è di dare per buono quanto scritto, ma avere l'accortezza di approfondire autonomamente gli aspetti che risultano più incomprensibili e scoprire tante altre possibilità che offrono questi semplici comandi.

Puoi scaricare il file e sperimentare direttamente.

Scarica il file





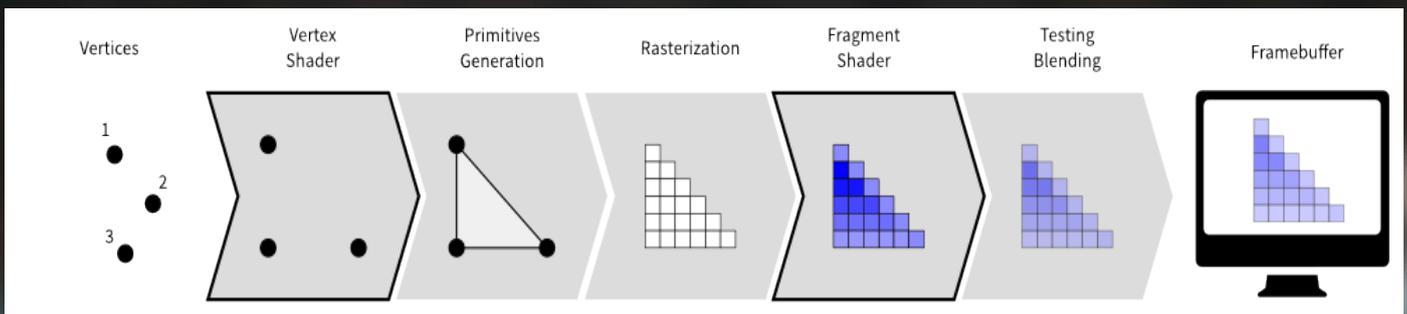
Nodevember

Di Isac Jack Amisano

Molti di voi non hanno nemmeno idea di chi io sia, ma giuro che 6 o 7 anni fa, quando ancora cercavo di trovare la mia strada (l'avrò poi trovata?), ero un habituè del forum di Blender Italia con il nick di "Soft House Graphics". Non molto tempo prima scaricavo la mia prima versione di Blender, 2.48b se non ricordo male, e partivo per la mia avventura con la modellazione di "Gus", l'omino di pan di zenzero con i bottoni colorati che sono sicuro abbia battezzato molti di voi. Blender rimarrà un fedele compagno di vita fino ad oggi, una nave pirata open source per scoprire le sfaccettature del mondo della CGI.

Ad ogni modo, salve a tutti, mi chiamo Isac (Isac Luca ndr), ho 24 anni, e dopo un "breve ma intenso" percorso di studi nell'animazione digitale mi trovo a lavorare come Junior Technical Artist in Ubisoft Milan, a Milano per l'appunto.

Ma passiamo subito al fulcro della questione... Shaders.



Che cosa sono gli Shaders? Forse sarebbe un qualcosa di estremamente delicato e complesso da trattare in un articolo, ma in una sintesi che definirei decisamente estrema e approssimativa sono dei microapplicativi che vengono eseguiti dalla GPU e che definiscono i singoli valori RGBA di ogni pixel sul nostro schermo.

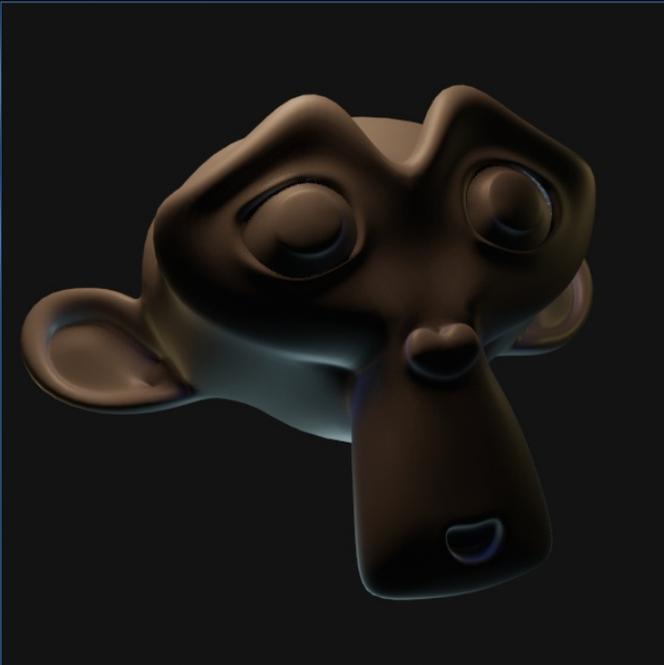
La gestione e la creazione degli Shaders sono una parte integrante del mio lavoro come tech artist e se compresi a fondo possono essere la soluzione per la stragrande maggioranza di problemi tecnici/artistici all'interno di una produzione (tentacoli organici che si deformano, esplodono, bruciano dall'interno e si dissolvono in piccoli frammenti atomici luminescenti che si spostano a tempo di valzer??).

Mi sto dilungando...

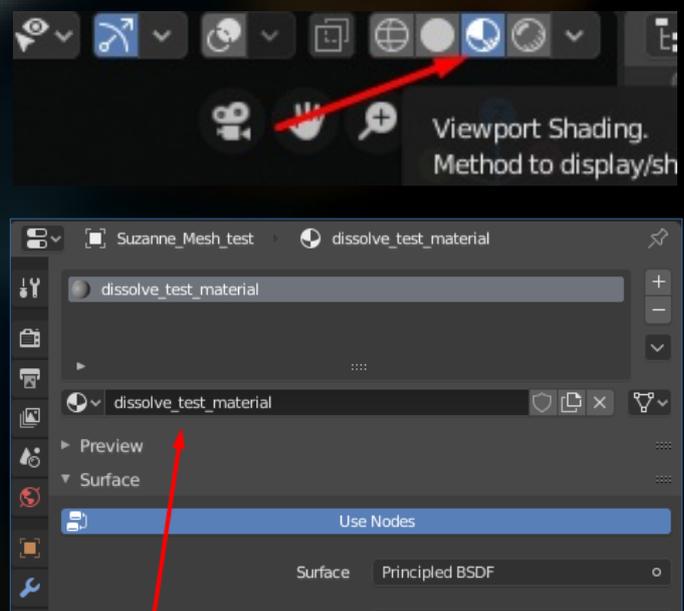
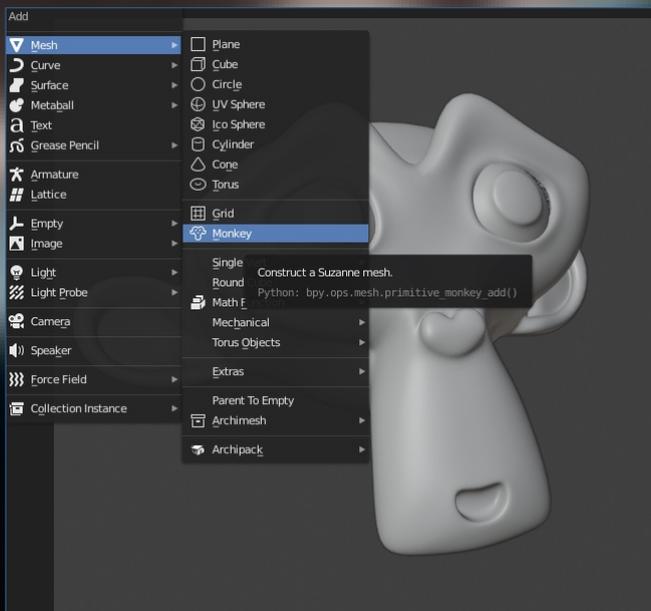
Con l'avvento di Eevee, il nuovo motore realtime per il "lookdev" in Blender 2.8, ho colto l'occasione per riavvicinarmi un po' al frullatore arancione e mi sono divertito a sperimentare alcune tecniche molto diffuse in ambito realtime. Nonostante lo shader node editor (soprattutto in termini di nodi di Utility) non fosse al pari di altri engine come Unreal o Snowdrop, ho visto in Eevee delle grandi possibilità e sono fiducioso sul fatto che continuerà a crescere anche su quest'aspetto nel prossimo futuro.

Uno degli esperimenti più veloci che ho voluto replicare è stato quello di Dissolvere una mesh agendo sulla sua visibilità all'interno dello shader, sfruttando il suo canale di Trasparenza e guidandolo attraverso una maschera che può essere animata e parametrizzata.

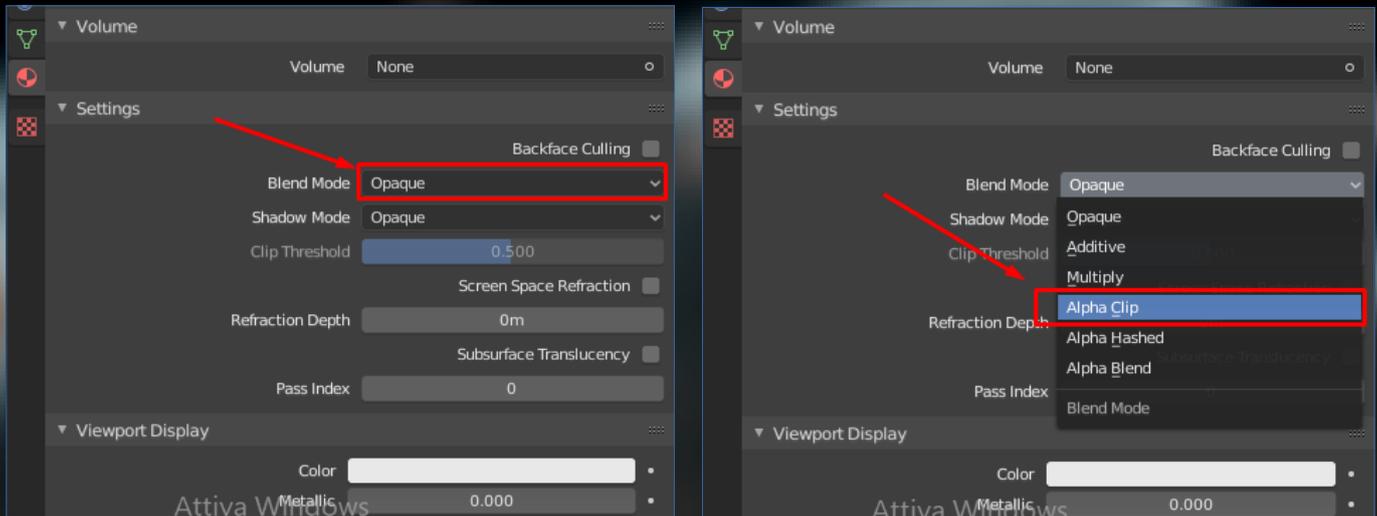
Tutto quello che ci servirà sarà un modello di partenza a cui andare ad assegnare il nostro shader. Per comodità userò la sempre disponibile Suzanne, dopodichè chiuderemo la nostra logica di dissolve in un compound node o nodegroup in modo che sia semplice e veloce replicarla in altri materiali.



La prima cosa da fare quindi è quella di andare a generare la nostra Suzanne, assicurarsi di avere Eevee impostato come motore di rendering e quindi attivo nella viewport, posizionare la nostra camera in modo da avere un buon framing e assegnare alla nostra mesh un nuovo materiale. Io ho preferito dargli anche un paio di livelli di suddivisione utilizzando il subdivision surface modifier per avere un modello più definito, ma non è cruciale per quello che andremo a fare.

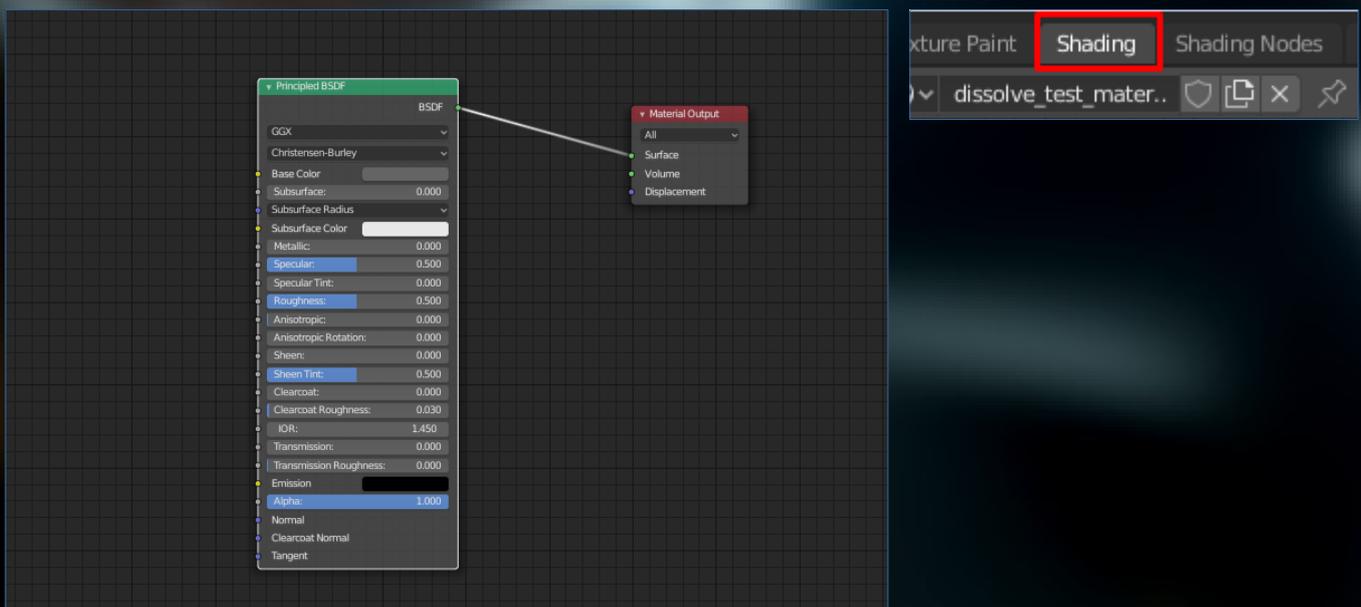


Prima di passare al node editor e alla composizione del nostro albero di nodi, è fondamentale definire come questo shader specifico andrà ad interpretare il canale di alpha nella nostra renderview. Per fare questo assicuratevi di avere selezionato Suzanne, poi nella tab contestuale "Material", nel menù "Settings" troveremo un sottomenù a tendina "Blend Mode", che di default sarà settato su "Opaque". Questo significa che il nostro shader non terrà conto in alcun modo della trasparenza eventuale che andremo ad assegnare al canale alpha e che la superficie sarà renderizzata sempre con un'opacità del 100%.

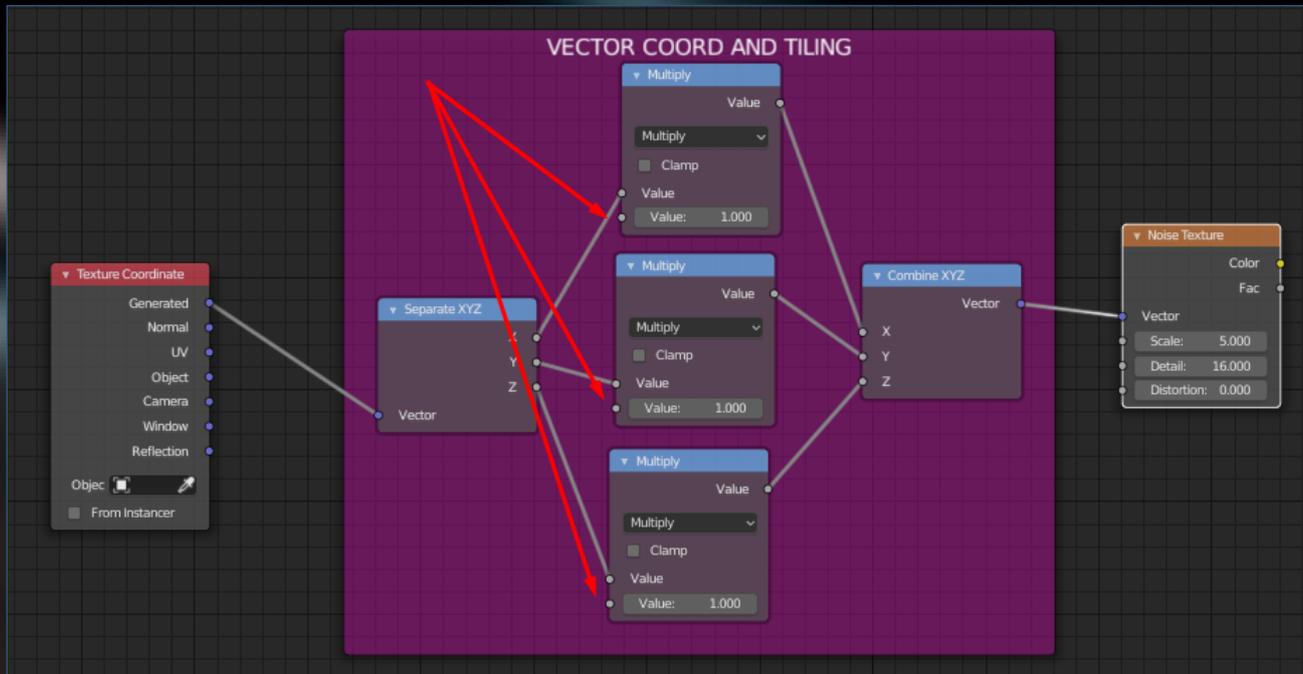


Cliccando sul menù troveremo diverse opzioni. Non spiegherò singolarmente i diversi metodi di blending che troverete per non allungare troppo il brodo, ma quello che ci servirà per il nostro shader sarà l'Alpha Clip (Binary visibility ndr), che sostanzialmente andrà a leggere i valori di opacità del nostro shader da 0 a 1. Tutti i valori sotto la soglia di 0,5 di alpha avranno visibilità pari a 0, viceversa, tutti quelli che supereranno questa soglia avranno opacità piena al 100%. Facile no?...

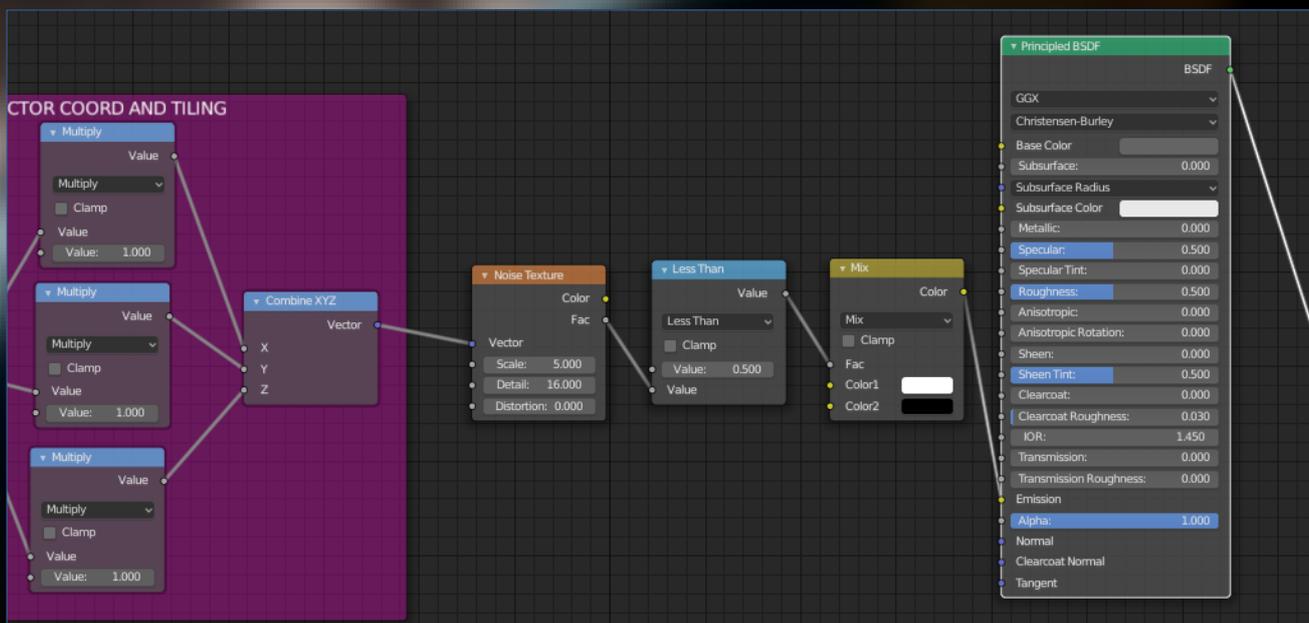
Bene, ora è il momento di spostarci nella finestra di Shading (che dovrebbe essere in alto nella vostra top bar) per iniziare a buttare giù la nostra logica a nodi per il dissolve da qui in poi parleremo principalmente di connessioni, input, output e operazioni matematiche di base.



Useremo un principled BSDF come output, quindi lasciamo il setup che ci troviamo davanti così com'è, e iniziamo ad aggiungere un nodo "Noise texture", che servirà per definire più o meno irregolarità nella transizione di visibilità. Colleghiamo un nodo "Texture Coordinates" al Vector input del noise e usiamo il "Separate XYZ" per agire sul tile delle coordinate come nell'immagine qui sotto. I valori dei multiply sui tre assi definiranno come il noise verrà proiettato sul modello 3D.

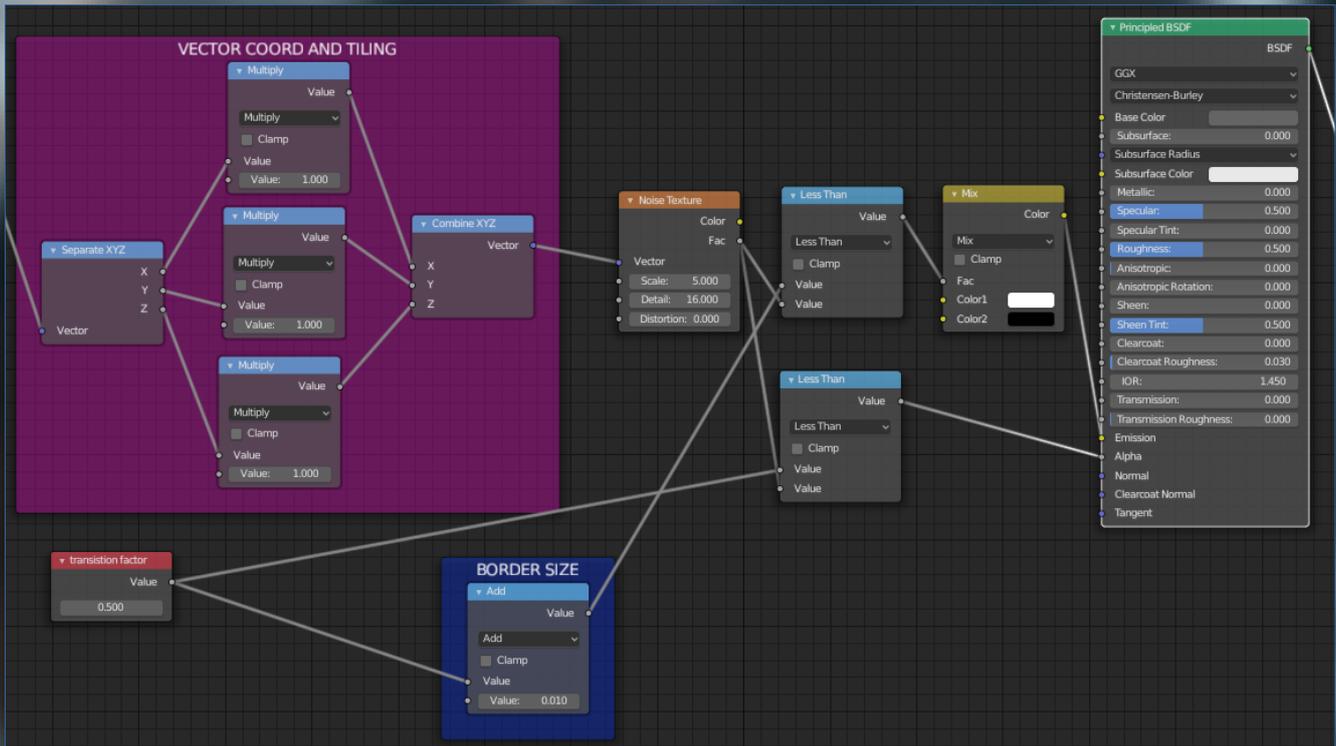


Ora che abbiamo più controllo quello che ci servirà sarà una funzione di "step" che andrà a leggere i valori in scala di grigio del noise che abbiamo creato e terrà conto solo di quelli oltre una determinata soglia che definiremo noi in seguito. Il nodo "Math" di "Less Than" è quello che fa al caso nostro e quindi colleghiamo il "Fac" output del noise al secondo input del "Less Than" lasciando momentaneamente scollegato il primo. Colleghiamo poi il tutto ad un nodo "MixRGB" che andrà a definire il colore del nostro bordo di dissolvenza. Infine, il mix node andrà al nostro input di "Emission" nel principled BSDF.

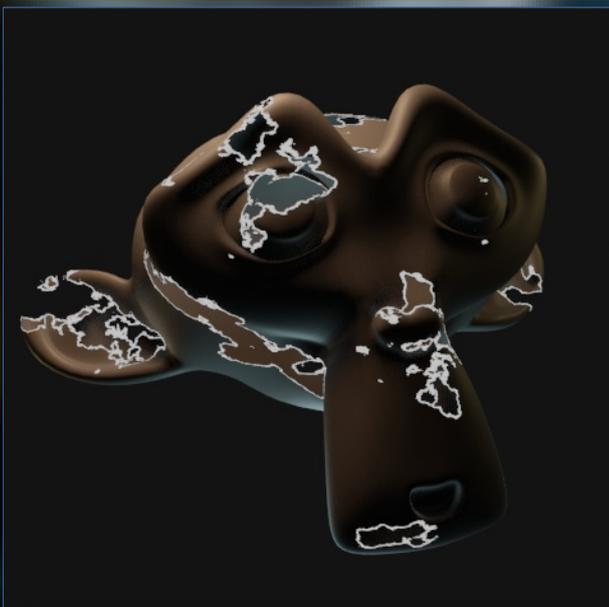


Ora è il momento di creare i valori che ci serviranno per animare la transizione e rendere visibile il bordo emissivo del noise. Per fare questo creiamo un nodo "Value" che corrisponderà al nostro fattore di transizione e colleghiamolo ad un "Math node" di tipo "Add", aggiungendo qualche valore molto basso (io ho usato 0.010). Questa semplice operazione matematica ci permetterà di avere il controllo sulla dimensione del bordo emissivo. Ora creiamo un altro nodo "Less Than" che guiderà il canale di Alpha del Principled BSDF. Colleghiamo il "Value" appena creato al primo input del "Less Than" e il noise al secondo input. Questo nuovo "Less than" andrà collegato all'Input di Alpha nel Principled Shader.

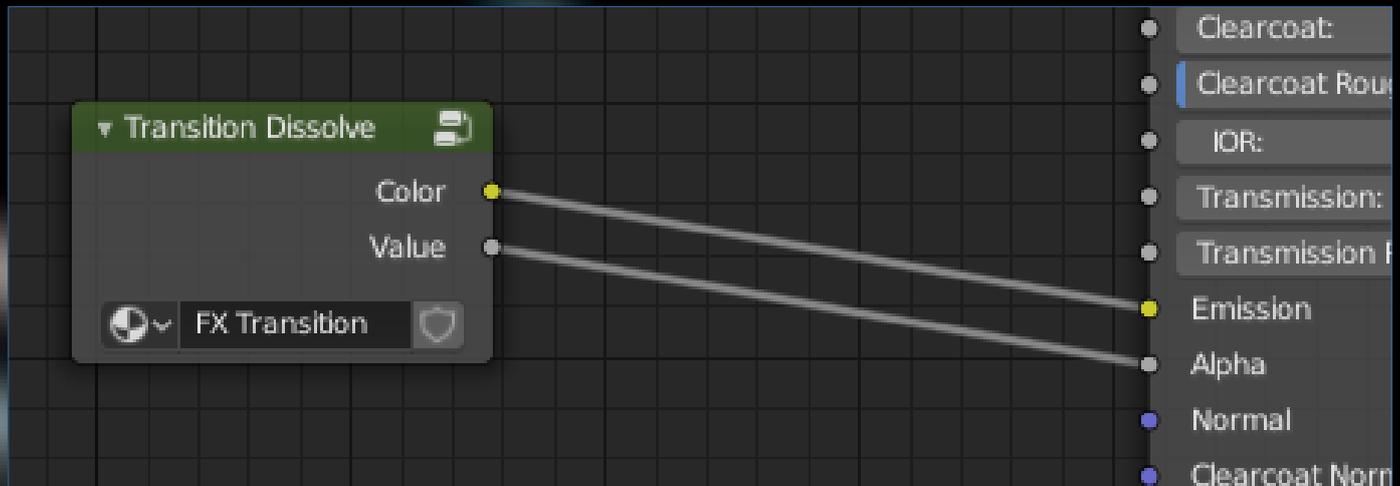
Il risultato della nostra operazione di Add invece andrà nel primo input del nodo di Less than creato in precedenza, così come nell'immagine.



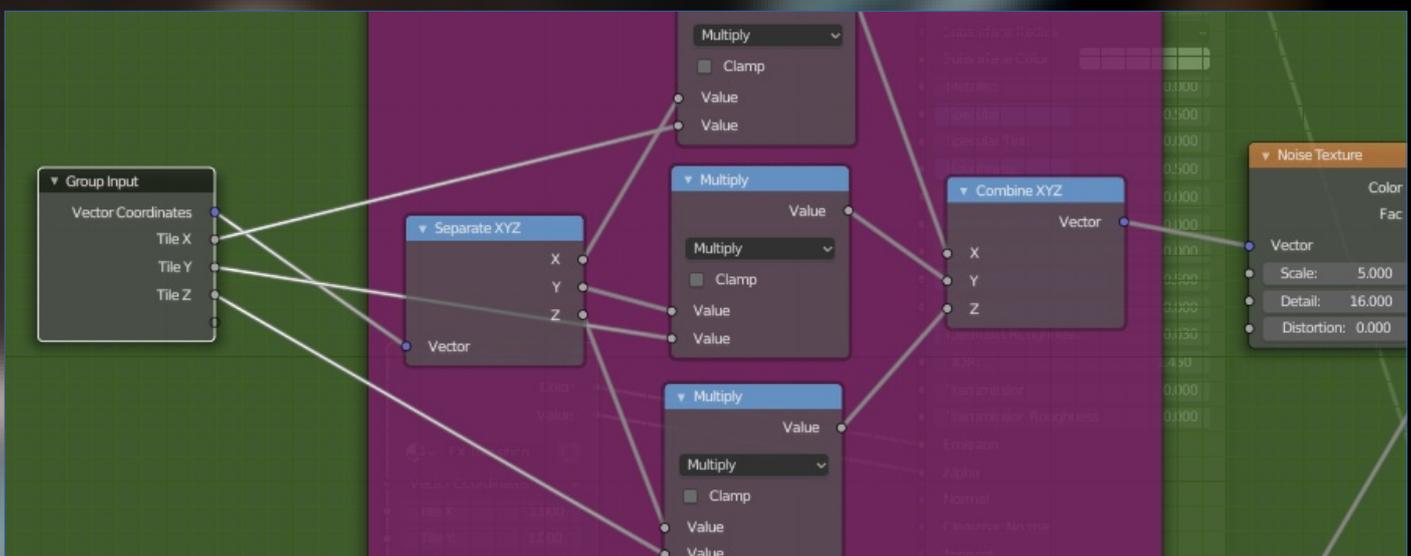
Nella viewport , se avete seguito tutti i passaggi correttamente, potrete già vedere un primo risultato.



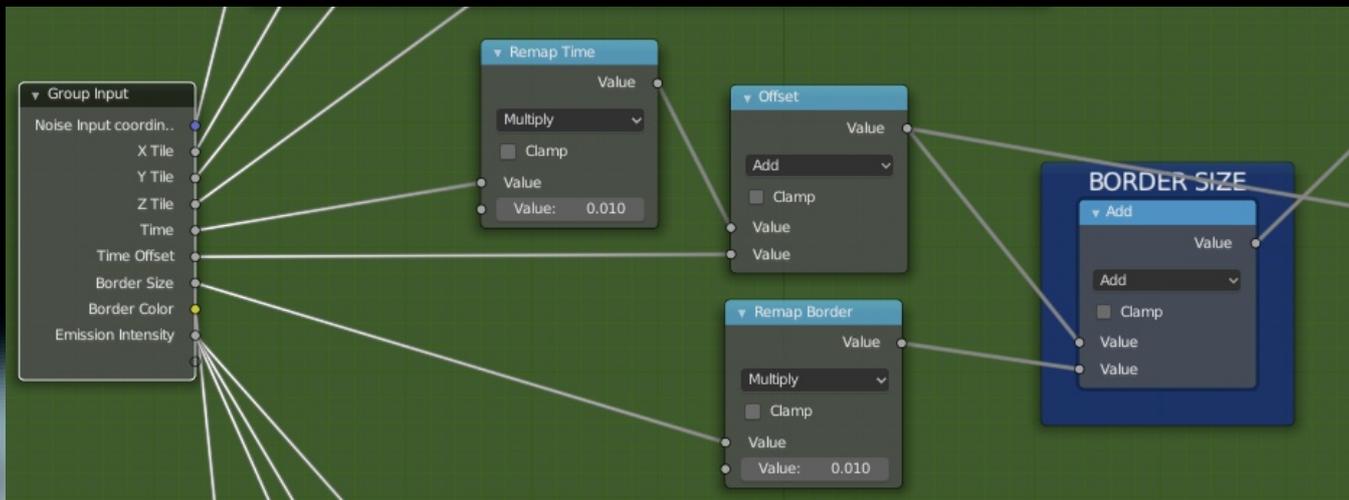
Ora però è giunto il momento di mettere tutto dentro un nodegroup, in modo da poter facilmente istanziare il nostro setup in altri shader e in modo da poter animare e regolare comodamente i nostri parametri. Selezioniamo dunque tutti i nostri nodi, ad eccezione del Principled BSDF, e con la shortcut CTRL+G li isoliamo all'interno di un nodegroup.



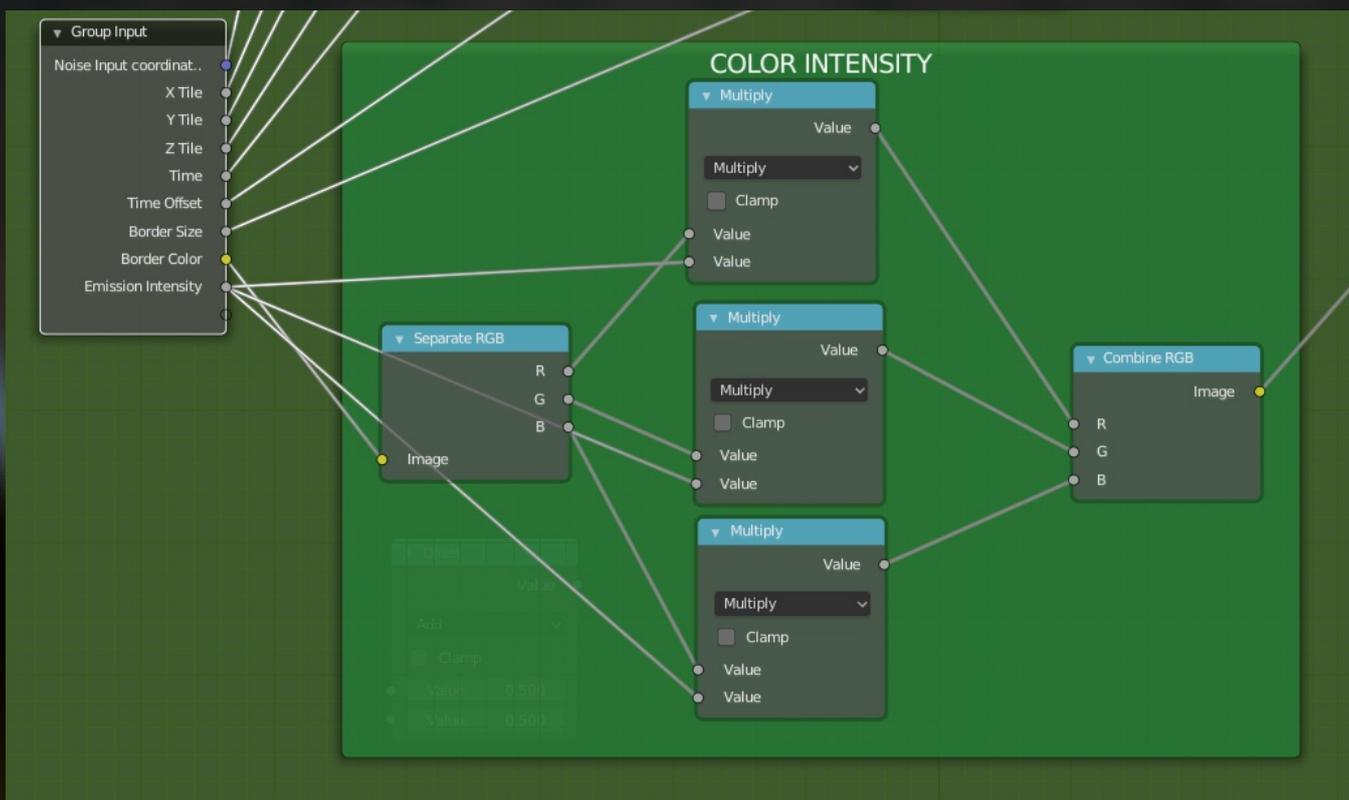
Selezionandolo e premendo TAB accederemo al setup dei nodi. Per prima cosa voglio eliminare il nodo di texture coordinates creato in precedenza, per avere la possibilità di cambiare a mio piacimento il metodo di projection del noise sulla mesh. Quindi lo esporremo come input dinamico, in modo da poterlo impostare al di fuori del nostro nodegroup. Come seconda cosa esporrò il valore dei tre moltiplicatori collegati alle coordinate XYZ, così potremo regolare il Tile del noise esternamente. Non dimenticate di dare dei nomi comprensibili ai parametri che vorrete esporre in modo da ritrovarvi quando lo utilizzerete nuovamente (O quando altri dovranno usarlo).



Un altro parametro che dovremmo esporre sarà il fattore di tempo che determina lo stato della nostra transizione. Quindi eliminiamo "Value" creato in precedenza e sostituiamolo con una connessione esterna al nodegroup. Io personalmente ho voluto inserirci anche un ulteriore nodo di Add per permettermi di gestire un offset esterno, oltre che due moltiplicatori per valori molto bassi per evitare che questi controlli siano troppo sensibili e difficili da regolare. Vi consiglio di fare lo stesso se volete evitare di dover giocare con gli "0,001" quando sarà il momento di settarli.

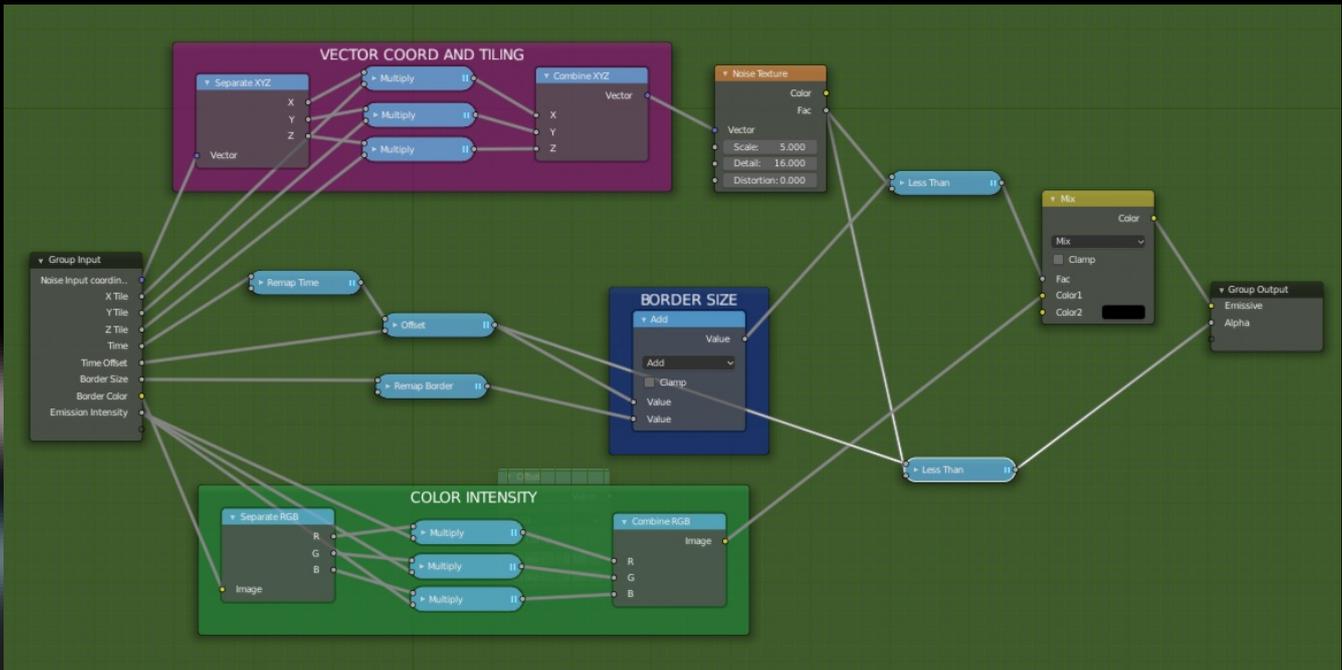


Facciamo la stessa cosa esponendo il value di Add a cui era collegato se vogliamo regolare esternamente la dimensione del nostro bordo emissivo. Esponiamo poi il "Color1" del nostro Mix Node (ricordate di lasciare impostato il nero sul secondo Input) per avere controllo sul colore del bordo e come aggiunta, possiamo separare i 3 canali e aggiungere dei moltiplicatori prima di ricombinarli in un vettore RGB, in modo da avere un controllo anche sull'intensità del valore di Emissive così come nell'immagine.

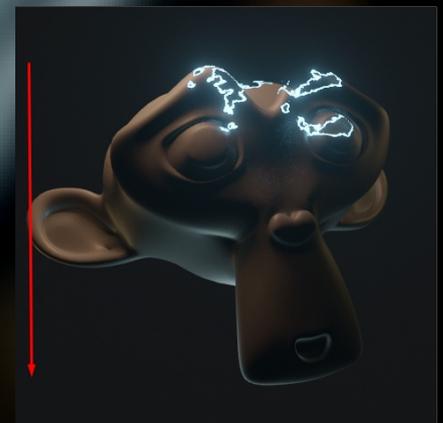
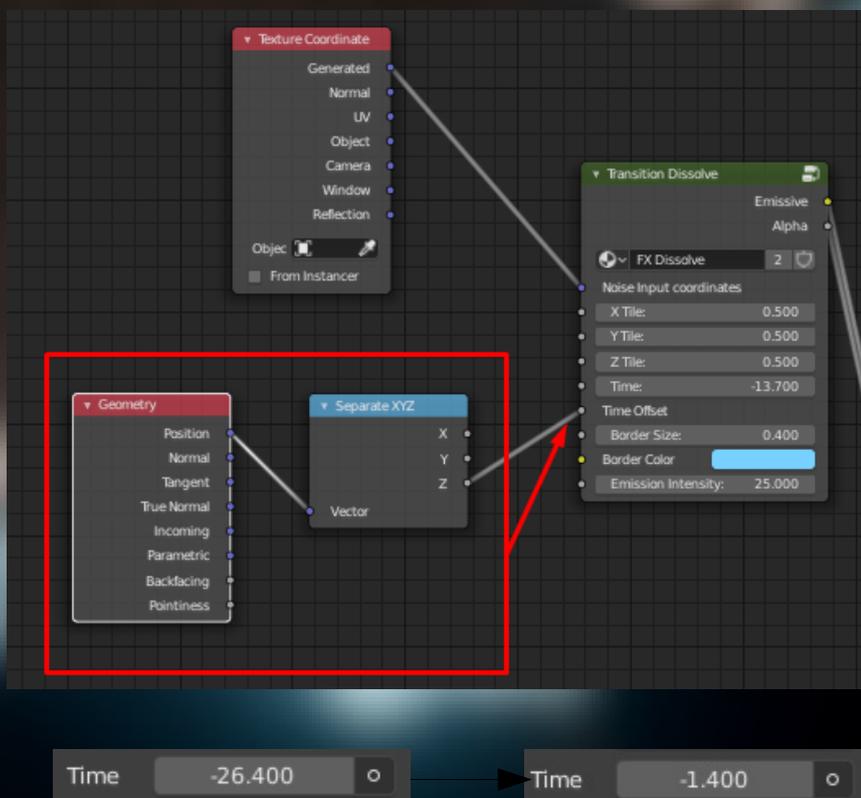


Rinominiamo infine anche gli output del nostro nodegroup, in modo da sapere sempre a quali input del Principled BSDF collegarli.

Ed ecco fatto, il nostro nodegroup di base per il dissolve delle mesh è pronto per essere istanziato ed utilizzato nei prossimi materiali, basterà chiamarlo dal menù di "add nodes", settare i vostri parametri preferiti e collegare gli output.



Ricordatevi di utilizzare in maniera creativa l'input esposto dell'offset del fattore di transizione. Agirà come una maschera di tempo a seconda di cosa gli collegherete. Qui per esempio ho collegato il canale Z del "Position Output" del nodo "Geometry" in modo da avere una transizione che scorre dall'alto verso il basso del mio modello, ma è solo un esempio, siate creativi e troverete un'infinità di modi per ottenere tutto quello che volete da uno Shader.





Morphing hack! Animare senza la fisica

Di Alessia Lorenzi

Introduzione

In questo articolo scoprirete un trucco di animazione che vi permetterà di avere un'alternativa semplice all'uso della fisica, spesso complessa e laboriosa da impostare e renderizzare.

Tuttavia sconsiglio ai beginner di usarlo per evitare di imparare la gestione della fisica. Saper gestire la fisica del proprio ambiente 3D è importante. Agli inizi è fondamentale sperimentare e arrivare a padroneggiare gli strumenti che Blender offre, in modo da poter essere sempre in grado di scegliere la tecnica più efficiente per ogni singolo caso.

Morphing, cos'è e quando usarlo

Si tratta di una tecnica che nell'ambito dell'animazione 3D è detta anche Morph target animation, per-vertex animation, shape interpolation o blend shapes.

Si parla di morphing quando avviene la trasformazione di una forma in un'altra (un quadrato che diventa un triangolo, una figura umana che diventa un lupo mannaro...).

Nel mondo dell'animazione 3D viene usata soprattutto per animare le espressioni facciali e per gestire le deformazioni muscolari (come la contrazione di un bicipite voluminoso piegando il braccio). La skeletal animation rappresenta il classico modo di animare un personaggio, viene infatti spesso associata alla morph animation per il controllo preciso delle espressioni e delle deformazioni della geometria o al motion tracking, il quale permette di ricevere informazioni riguardo i movimenti da un viso umano reale, conferendo naturalezza ai movimenti del personaggio digitale. In ogni caso, Blender permette di utilizzare tutte le tecniche elencate.

Come animare tramite Morph target Animation

I keyframe definiti in una timeline e le shape keys determinano l'animazione con l'uso del morphing. Definiamo con parole semplici questi concetti: i keyframe sono degli elementi che vanno inseriti in una timeline per fissare dei parametri in un momento preciso del video, tali parametri possono essere la posizione, la rotazione e un qualsiasi altro valore, tra cui anche le shape keys. Queste ultime sono appunto delle forme chiave, posizionando due shape keys diverse tra loro in due punti diversi della timeline otterremo un'animazione calcolata dal software in modo da trasformare la prima forma indicata nella seconda (da umano a licantropo, per esempio). "Trasformare" significa che non apparirà un nuovo oggetto nella scena, bensì la posizione dei vertici cambierà mutando l'aspetto dell'oggetto tridimensionale. Le 3 figure a destra rappresentano tre shape key di uno stesso oggetto: la geometria cambia ma l'oggetto rimane lo stesso.



Fisica, skeletal animation o morphing?

Il modo migliore per animare un personaggio è un agile compromesso tra i vantaggi sia dell'animazione con scheletro che della morph target animation. Si possono usare le potenzialità di entrambe le tecniche senza escluderne nessuna. Lo stesso vale per l'uso delle forze e della

fisica in generale. Il morphing permette di modellare con estrema precisione le trasformazioni desiderate. Per questo motivo il mio consiglio è di non perdere troppo tempo impostando i parametri della fisica (soft o rigid body, forze, ecc...) quando lo stesso risultato si può ottenere in minor tempo utilizzando la tecnica della morph target animation per simulare anche l'interazione tra oggetti (come impatti, superfici elastiche che vengono deformate o altro).

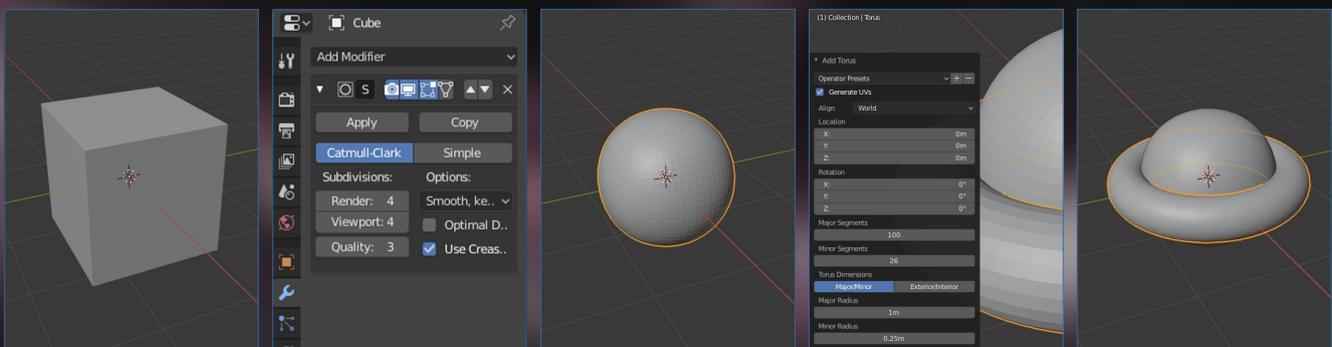


Step by step

Analizziamo passo per passo la realizzazione di quest'animazione che potete trovare nel mio canale youtube e nel mio sito www.alessialorenzi.it

1. **Otteniamo una sfera** applicando un Modifier al cubo di partenza: subdivision surface. Manteniamo l'opzione Catmull-Clark ma modifichiamo la suddivisione: Viewport 4 e Render 4.

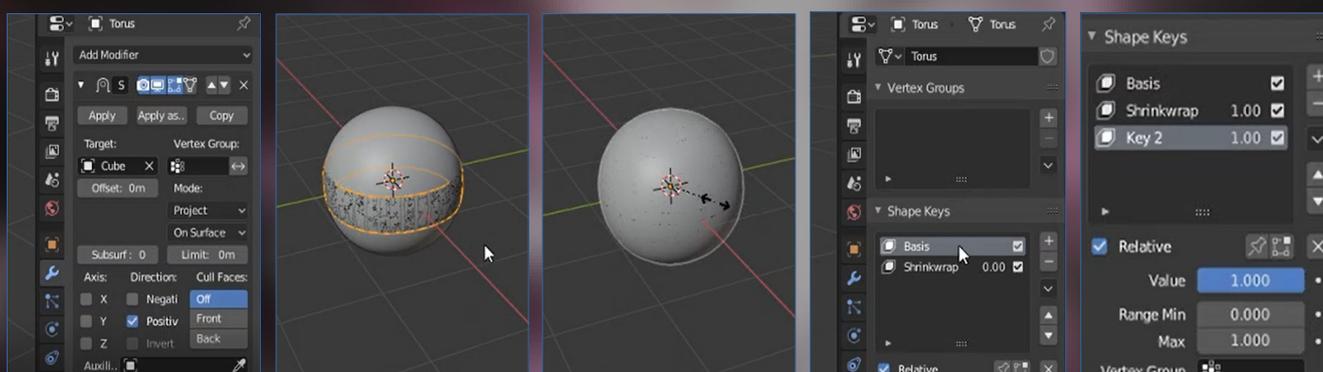
2. **Creiamo un anello**. Ho preferito crearlo con i seguenti parametri per quanto riguarda i segmenti: maggiori 100 e minori 26. Per preferenza ho applicato **Shade Smooth** a entrambi gli oggetti in modo da vedere figure smussate e pulite.



3. **Selezionare l'anello** e attivare **Shrinkwrap modifier**. Impostare come target il cubo e cambiare la modalità in "project" dai parametri del modifier. Spostare e ridimensionare l'anello fino ad adattarlo completamente alla forma sferica. A quel punto, cliccare su **Apply as Shape Key**. Nel pannello delle proprietà dell'oggetto **saranno apparse due shape keys**: basis (l'anello di partenza) e shrinkwrap (la sfera).

4. Creiamo ora una **terza shape key** cliccando sul simbolo "+", ossia il comando Add shape key.

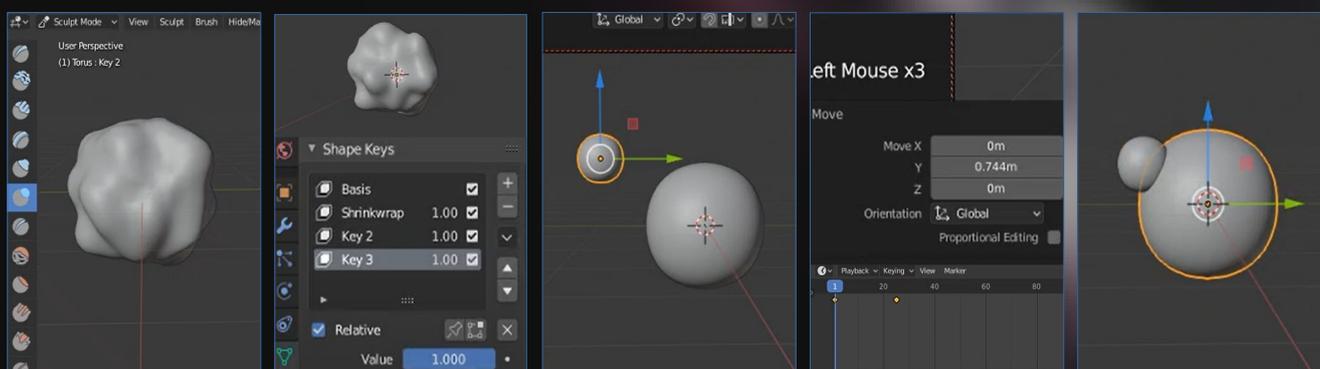
5. Portiamo il valore della shape key shrinkwrap e della nuova shape key da 0 a 1 in modo da poterla visualizzare e modificare in **Sculpt Mode**.



6. **Scolpiamola ottenendo una superficie increspata**. Per creare un movimento più interessante ho creato una quarta shape key con un altro tipo di increspatura. Abbiamo ora quattro shape key.

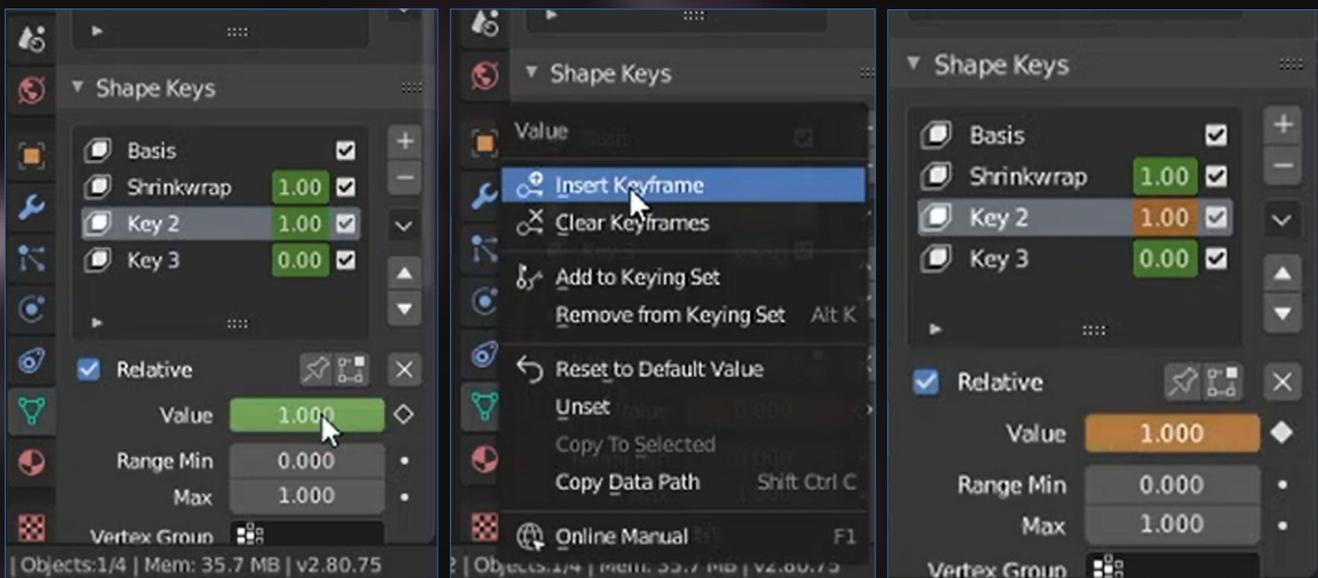
7. Utilizziamo il cubo di partenza (ora a forma di sfera). **Riduciamo la sua dimensione in modo da renderlo una piccola sfera**.

8. **Aggiungiamo un keyframe** in cui l'oggetto si trova lontano dalla sfera (l'anello di partenza con forma chiave shrinkwrap a valore 1) e un secondo keyframe in cui l'oggetto si trova dentro alla sfera. Così sembrerà lanciato verso di essa (con le curve di animazione si può creare un movimento più naturale correggendo l'accelerazione).



9. Quando la piccola sfera collide con la grande sfera, la si può far incresparsi fissando nella timeline i valori delle ultime due shape key create e modificandoli. Poi, continuando a giocare in questo modo con i valori delle shape keys, con la rotazione e con la posizione degli oggetti, si potrebbe espellere la piccola sfera da quella più grande mutando quest'ultima in anello e terminando così l'animazione.

L'intera animazione, collisione inclusa, è quindi risolta col controllo delle shape keys e della posizione nello spazio varianti tempo, semplici da impostare ed estremamente precise.





Creare un servizio da Tè – parte III

Di Francesco di Buono

Riprendiamo a lavorare sui materiali cominciando dal legno del tavolino, quello più complicato. Per questo materiale, useremo esclusivamente delle mappe – in luogo delle curve - per determinarne la specularità e misceleremo due textures legno diverse per creare un effetto invecchiato. Innanzitutto scegliamo una texture legno di tipo pregiato, del tipo noce o ciliegio (quelli lucidi tendenti al rosso) ed una più naturale, non lavorata, per le parti scrostate. Dopodiché ci servirà una terza texture, in bianco e nero, per determinare la visibilità dei due tipi di legno, ovvero quali saranno le parti invecchiate/scrostate e quali quelle nuove/pregiate. Sarà bene sceglierne quindi una che abbia dei graffi o comunque delle linee verticali sottili – questo tipo di texture in genere si può trovare sotto il nome di “dirt/grunge/scratches map” o “decal” - e ce ne sono di gratuite.

Della prima texture, quella pregiata, dovremo recuperare o ricavare anche le mappe normal e roughness. Ricordo che esistono vari programmi/plugin gratuiti per creare le varie mappe partendo da una semplice texture diffuse (io ho usato Awesome Bump).

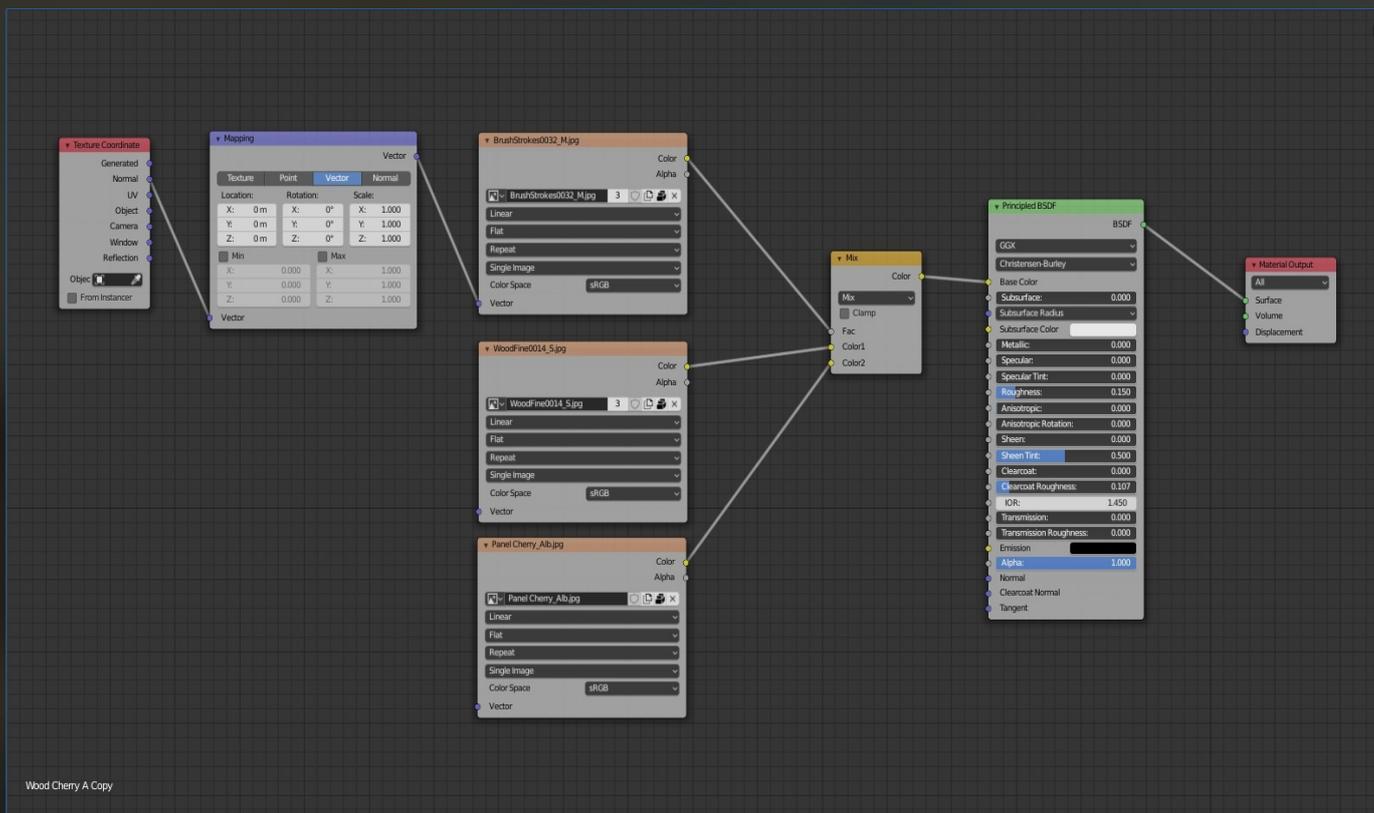
Nel node editor (ambiente shading nella 2.80) aggiungiamo un “image texture” e carichiamo quella principale - la diffuse del legno pregiato - e inseriamola nel color input (superiore se i graffi sono bianchi su sfondo nero, inferiore se i colori sono invertiti) di un nodo mix rgb; aggiungiamo un altro nodo con la texture del legno “grezzo” e inseriamola nell’ altro input, lasciando su “mix” il tipo di miscelazione. Ora usiamo un terzo image texture e carichiamo la “maschera” che dovrà guidare i due tipi diversi di legno collegandola all’ input “factor” del mix rgb. Questi costituirà il base color di un principled shader

Opzionale

Per poter avere più controllo sull’ ammontare dell’ effetto, si può passare l’ output color della texture per una color ramp. Questa ci permette di regolare la “nettezza” dello scarto tra le due textures, nonché il livello generale dell’ effetto: infatti se avviciniamo i due estremi dei colori bianco e nero, vedremo che il “taglio” sarà più deciso... Inoltre la gradazione di nero e bianco dei due estremi fissa una soglia minima e massima tra i due colori: se ad esempio la texture “invecchiata” dovesse risultare troppo diversa, staccata, da quella principale, si può aumentare il valore del nero verso il grigio, per farla risultare “stemperata” e più o meno sovrapposta a quest’ ultima.

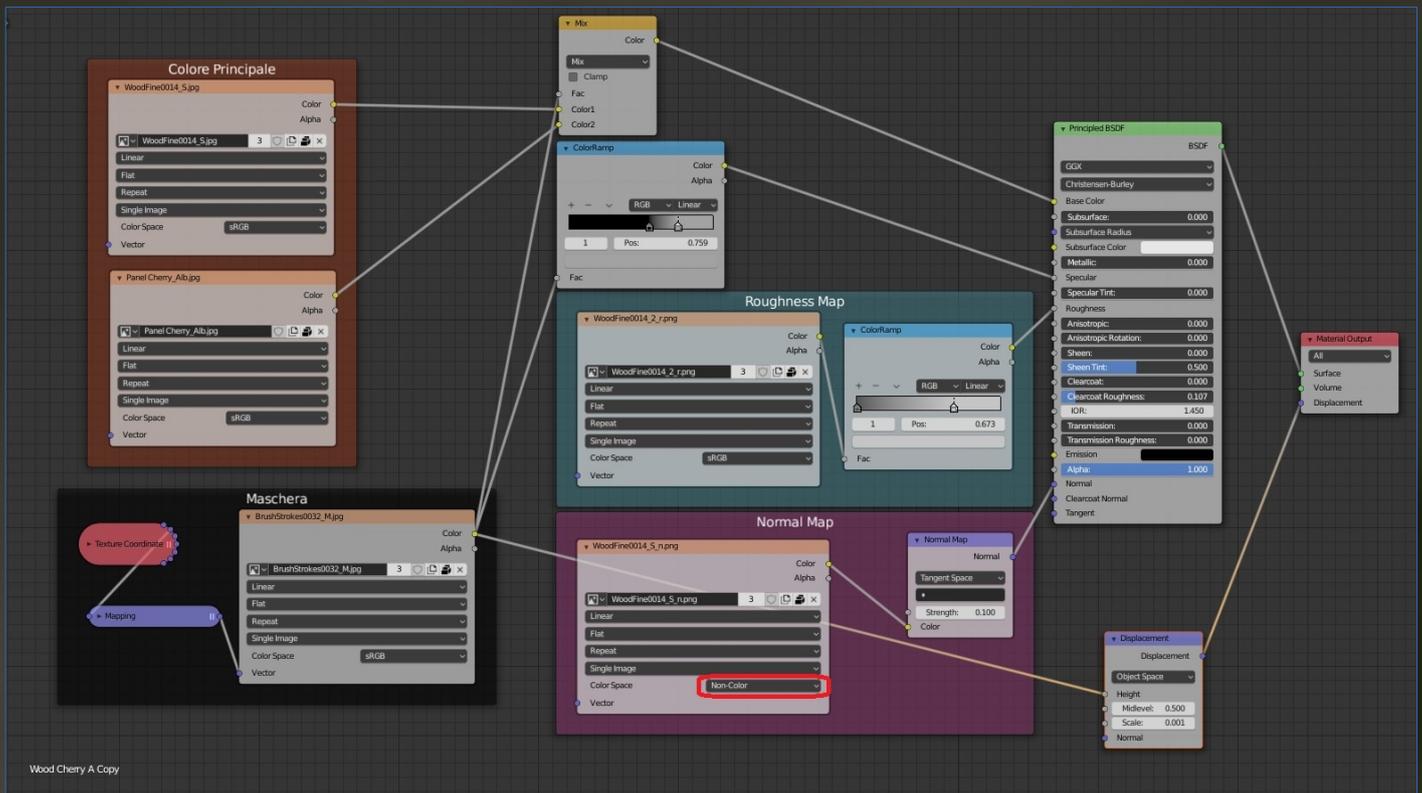
Ricordiamoci di effettuare la scucitura UV delle mesh del tavolino, selezionando tutte le facce (selezione face mode) in edit mode e poi premere “u” e scegliere “follow active quads” - eventualmente deselegionando e rielezionando una faccia per dargli il punto di riferimento attivo – per avere una scucitura allineata; potrebbe essere possibile dover scalare tutte le UV verso il basso per far corrispondere le venature del legno alla scala dell’ oggetto

Tutti i nodi image texture utilizzeranno automaticamente la scucitura UV come punto di riferimento per l’ applicazione delle textures, quindi le due immagini del legno avranno di default l’ effetto desiderato... Ma nel caso della nostra texture “maschera” potremmo non avere il risultato voluto: un buon modo per assicurarsi che le striature di legno scrostato seguano la direzione della geometria, è quello di utilizzare il vettore “normal” di un “texture coordinate”. Aggiungere quest’ ultimo, inserirlo in un nuovo nodo “mapping” e inserirlo nell’ input “vector” della nostra texture maschera. Nelle opzioni del nodo mapping selezionare “vector” in luogo di “point”

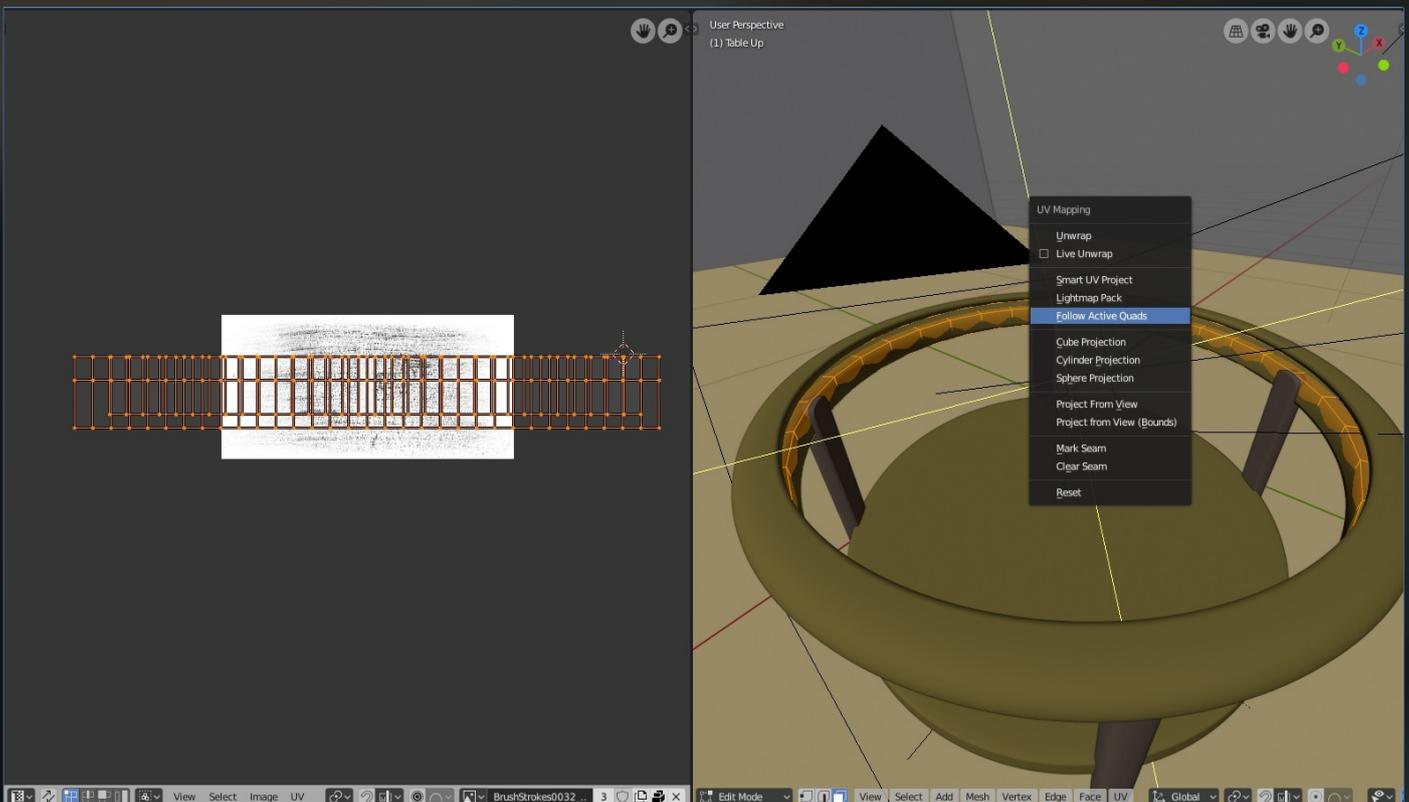


In questo modo la mappa di invecchiamento non sarà dipendente dalle UV, ma comparirà lungo le direttrici della geometria (funziona molto bene con curve digradanti o elementi con pochi spigoli).

Ora, il nostro nodo mix rgb sarà il “base color” di un principled shader; l’ output della texture maschera, invece, dovrà essere duplicato e fare da base per il displacement (nodo displacement, “height” input e inserirlo nel material output) perché il nostro legno è di tipo lucido, quindi nelle sue parti integre sarà liscio mentre nelle parti scrostate avrà una leggera rientranza: in pratica la nostra maschera funge anche da displacement. Per mitigarne l’ effetto, si dovrà abbassare la scala a 0.001: ora il nostro legno avrà dei leggerissimi avvallamenti nelle aree dove emerge la texture naturale. Parimenti, si dovrà sfruttare la maschera per creare delle differenze di specularità tra i due tipi di legno: inserire l’ output in una color ramp e portare l’ estremo di colore scuro verso il centro (parte non speculare) e alzarne il valore di qualche decimale, per non avere zone a specularità zero. Muovere verso sinistra anche l’ estremo bianco e abbassarne il valore per evitare aree a specularità totale. Carichiamo la nostra roughness map e usiamola nell’ input roughness – appunto – del principled, ma anche qui regoleremo gli estremi con una color ramp per evitare che la rugosità spazi da zero a totale. Infine, carichiamo la mappa delle normali (nel nodo image texture settare il color space su “non color data”) e, associandola ad un nodo normal map (abbassandone l’ intensità per un effetto più realistico), inseriamola nell’ input “normal” del principled shader.



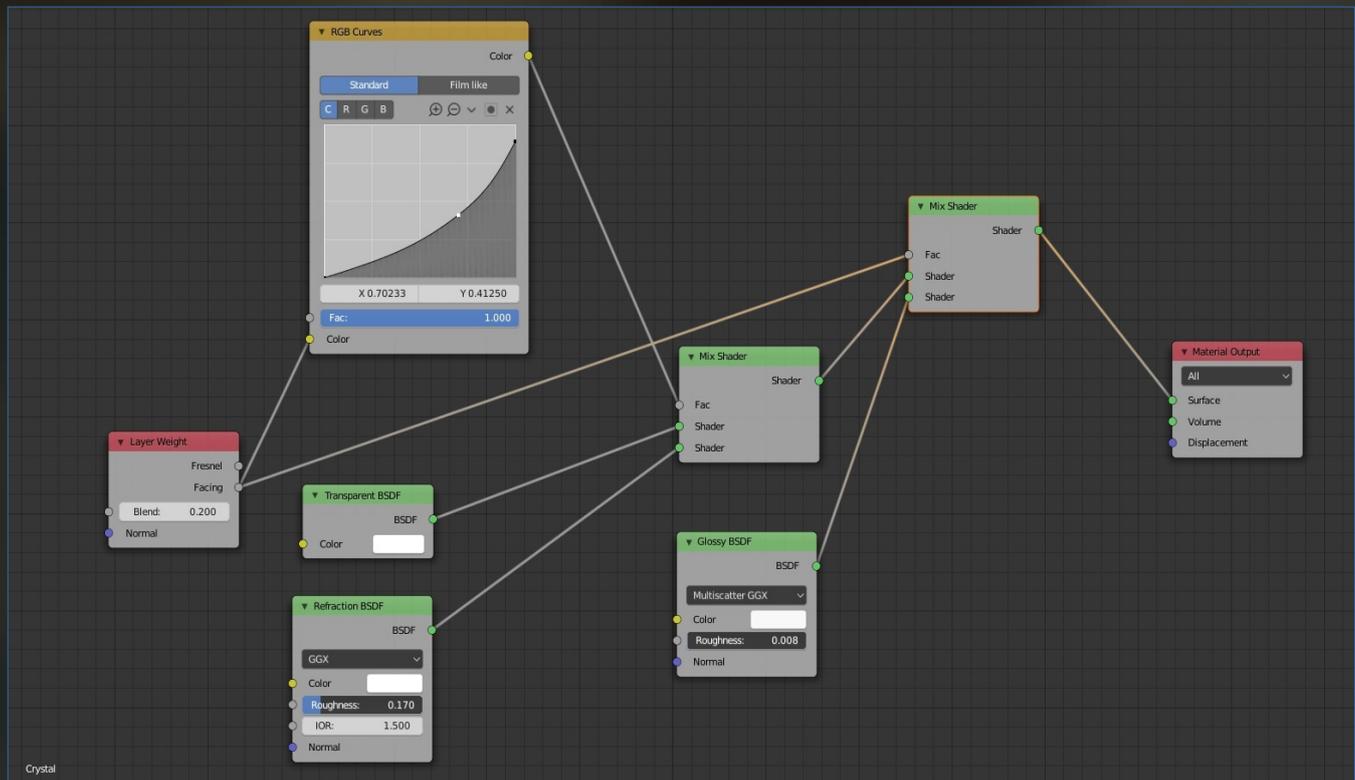
Applichiamo questo materiale a tutti gli elementi del tavolino, ma, nel caso della base, converrà utilizzare una scucitura UV diversa; il metodo più efficace che ho trovato è quello di selezionare le facce superiori e inferiori ed effettuare un unwrapping del tipo "project from view (bounds)", quindi selezionare le facce laterali e scegliere "cylinder projection" (eventualmente scalando): in questo modo le venature del legno saranno tutte perpendicolari ai piani.



Vetro

Per il vetro del tavolino NON ci serviremo né del principled né del glass shader, bensì di una combinazione di transparent e refraction. In realtà questo è il principio fisico con cui la luce interagisce con una superficie di questo tipo – a seconda dell' angolo di visuale, vedremo selettivamente luce riflessa o filtrante -

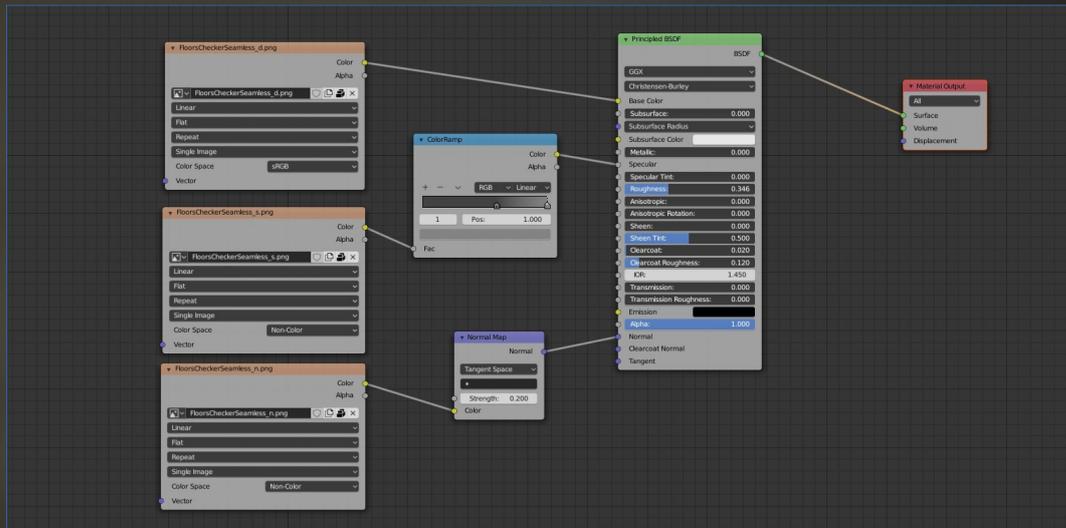
Creiamo quindi un mix shader in cui inseriremo questi due shader (transparent nell' input superiore, refraction nell' inferiore) e aggiungiamo un "layer weight" il cui blend imposteremo a 0.200. Aggiungiamo anche un "rgb curves" e disegniamo una curva con una concavità molto leggera, quindi uniamo l' output "facing" del layer weight e colleghiamo l' rgb curves con il mix shader. Lo shader transparent dovrà avere il color sul massimo valore di bianco (trasparenza totale) mentre il refraction dovrà avere la roughness entro i valori 0.100 - 0.160 (non completamente trasparente) e l' IOR intorno a 1.5 (il valore del vetro comune è circa 1.47, più è alto e più il vetro apparirà impuro). In questo modo, creiamo un leggero effetto densità che sarà dipendente dall' angolazione della visuale. Aggiungiamo un nodo glossy (roughness molto bassa, circa 0.00X) il cui output andrà nel nodo inferiore di un nuovo mix shader e inseriamo il gruppo precedente nell' input superiore di questo; nel mix colleghiamo l' output facing del layer weight (bypassando l' rgb curves). Ora abbiamo dei riflessi netti che tendono a lasciare il posto alla trasparenza quanto più si guarda il vetro perpendicolarmente.



Pavimento

Per il pavimento scegliamo una texture di un pattern classico, adatto ad un ambiente tipo salone/sala da the. Io ho optato per le classiche mattonelle a scacchi. Di questa, serviranno le mappe specular e normal. Essendo un semplice piano la scucitura UV potrà essere quella base (unwrap), cercando però di scalarla in modo che la grandezza del pattern sia coerente con le dimensioni degli oggetti.

Nel color di un nodo principled inserire la diffuse texture, quindi nell' input specular inserire la specular map frapponendo una color ramp per regolarne gli estremi: il valore sinistro dovrà tendere verso il grigio scuro ed essere spostato verso il centro, mentre quello destro dovrà tendere al grigio chiaro/medio e rimanere verso l' estremità. La roughness dovrà avere riflessi poco netti – piuttosto alta di diversi decimali, nell' esempio è 0.346 - perché anche un' eventuale lucidatura sarà affidata al glossy coating; infatti quest' ultimo dovrà essere superiore a zero (0.02) e con una roughness non troppo lucida (0.12) per evitare un effetto specchio. Per ultima carichiamo la normal map e procediamo come in precedenza impostando "non color data" e collegandola al vettore "normal map" prima di collegarla all' input "normal" del principled. Il valore di intensità dovrà essere piuttosto basso (0.100 - 0.200) anche perché, nel rendering finale, il pavimento sarà sfocato con il depth of field e non saranno visibili tutti i dettagli.

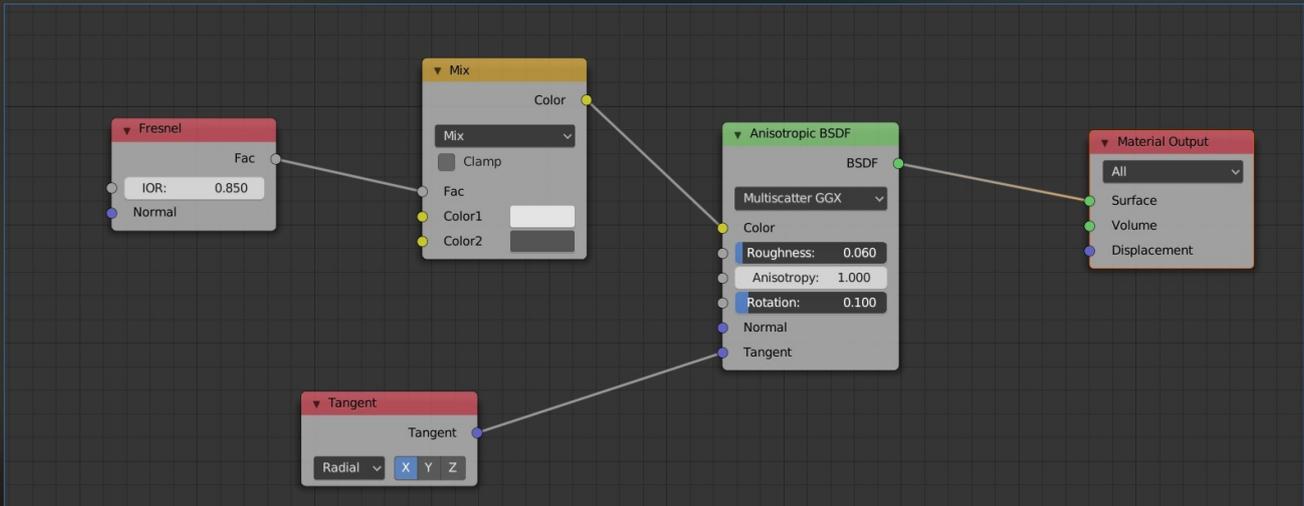


Cucchiaino

I materiali metallici sono tra i più difficili da rendere correttamente in Cycles, nei miei vari studi sull' argomento sono arrivato a un setup di nodi molto complesso che li rende in maniera soddisfacente, ma per lo scopo di questo tutorial ne creeremo una versione semplificata (sia per motivi di spazio che di focalizzazione). Creiamo un nodo anisotropic in un nuovo materiale e impostiamo la roughness su un valore molto basso (io ho usato 0.06 ma si possono usare valori diversi o anche utilizzare una mappa di "sporco" per simulare macchie di opacità).

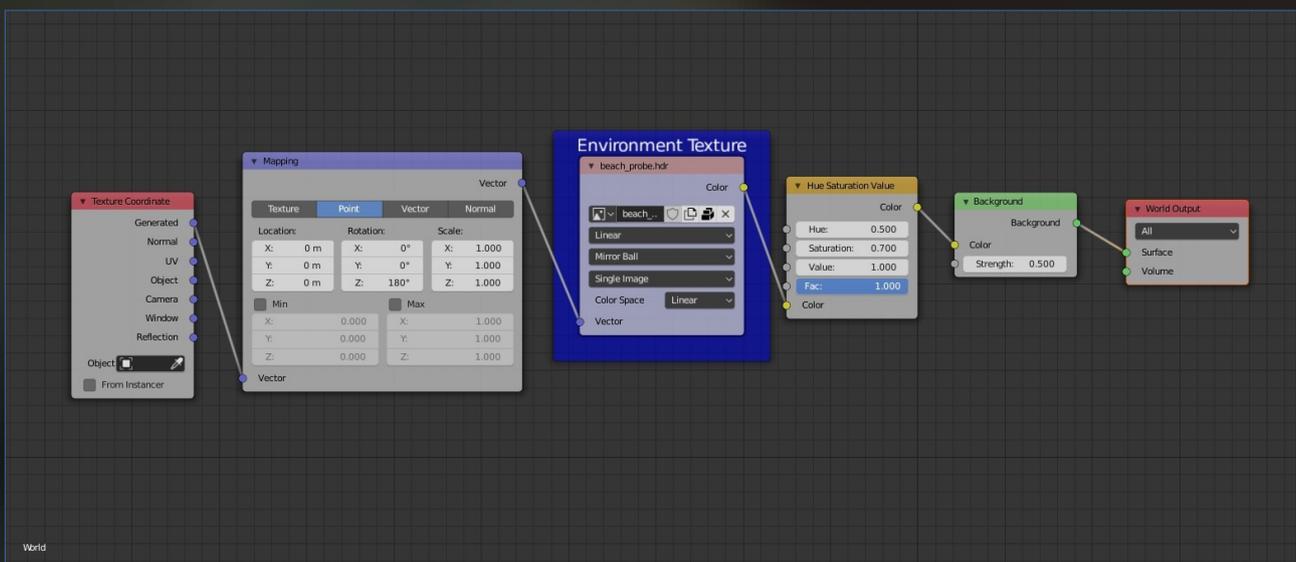
Alziamo l' anisotropy al massimo e la rotation a 0.1 - l' anisotropia è la "direzione" in cui viene incanalata la luce, fenomeno presente in molti metalli spazzolati – e scegliamo dal menù superiore dello shader l' algoritmo "multiscatter CGX".

Aggiungiamo poi un input fresnel e un mix RGB, scegliendo come colore superiore un bianco molto chiaro e come colore inferiore un grigio scuro e inseriamolo nell' input color dello shader anisotropic. L' Index Of Refraction del nodo fresnel dovrà essere inferiore a 1 (0.85 nel mio caso) per avere riflessi "negativi" rispetto all' angolo di incidenza visivo. E' importante notare che un materiale metallico non trattato/dipinto è sempre riflettente al 100% e che, a causa delle caratteristiche fisiche, "piega" la direzione dei riflessi (in realtà si tratta di un fenomeno di eccitazione - da parte dei fotoni della luce - degli elettroni) . Ora aggiungiamo anche un input "tangent", lasciamo la direzione su "radial" e impostiamo l' asse su "x", quindi colleghiamolo all' input "tangent" del nodo anisotropico. Il risultato dovrà essere la tipica curva scura che si vede nelle aree dei metalli angolate rispetto alla luce

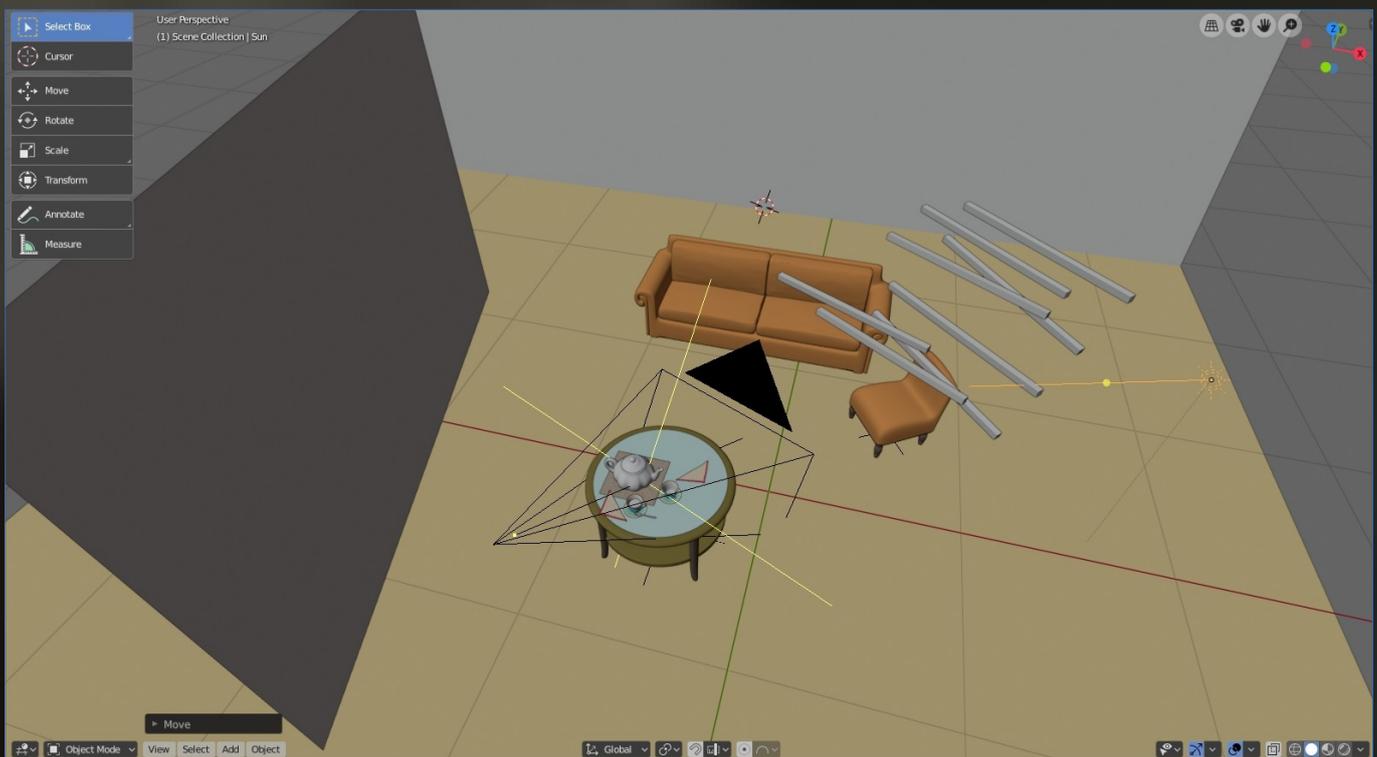


Render setup

Questo è uno dei passaggi più importanti perché una buona illuminazione può rendere "credibili" anche materiali non corretti e dare un "tono" all'intera scena. In questo caso si tratta di un rendering di interni, che tende ad essere molto scuro e richiede molti più samples, rispetto agli esterni, per ridurre al minimo il "rumore". Per ovviare a questi problemi, ho usato dei "trucchi" che funzionano molto bene e possono essere replicati anche con altre scene di interni. Innanzitutto cerchiamo un HDR dai colori neutri e dalla luce molto diffusa poichè le ombre non dovranno essere nette. Un'astuzia: spesso gli HDRi hanno una dominante molto forte o cmq colori troppo accesi che conferiscono un tono "sintetico" alla scena; per ovviare a questo problema, interponiamo un nodo "hue/saturation" tra il nodo "environment texture" e quello "background" dal node editor (modalità "world") e abbassiamo la "saturation". Aggiungiamo anche i nodi "texture coordinate" e vector=>"mapping" collegandoli attraverso l'output "generated" del primo e susseguentemente nell'input "vector" dell' environment texture. E' essenziale perdere un po' di tempo a scegliere l' HDR giusto, perchè la nostra scena di interni sarà una scena esterna! Infatti cercheremo di riprodurre l'illuminazione tipica della luce riflessa dalle pareti (e conseguenti ombre di oggetti non visibili nella scena) in modo artificiale

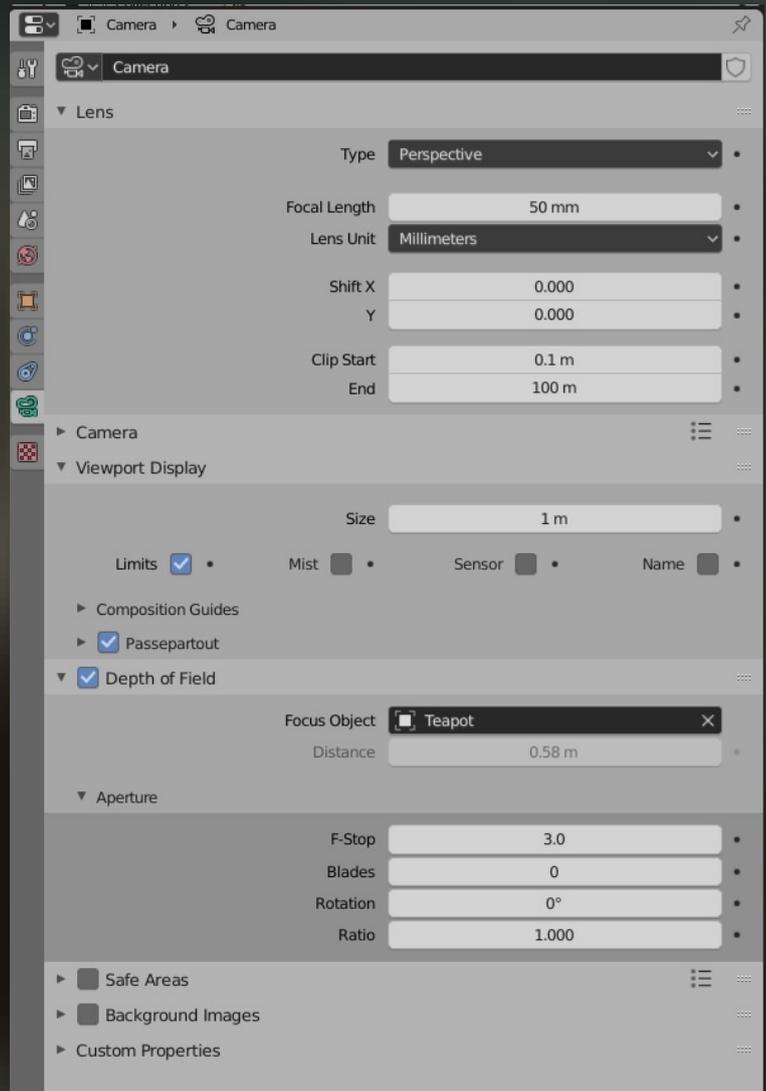


Dopo aver trovato un HDR soddisfacente aggiungiamo una luce di tipo "sun" le cui ombre siano piuttosto nette (10cm 2.79 - 11 gradi 2.80) e facciamo combaciare l'angolo di illuminazione di questa, con quello dell' HDR: nascondiamo i layer/collection in cui sono presenti mesh e visualizziamo in render mode la viewport (shift+z 2.79 - z,8 2.80) per individuare il punto in cui si trova la luce solare o l'area più luminosa della environment texture; eventualmente si può ruotare l' HDR sull'asse z del nodo mapping e regolare successivamente la posizione della luce sun. Da notare che la "strength" del background deve essere piuttosto bassa (0.500 nell'esempio) e anche la luce solare non dovrà superare il valore di default. Per quanto riguarda l'occlusione della luce, il pavimento dovrà essere molto più grande del tavolo per non far giungere luci da sotto, e si deve anche inserire una parete (materiale di default) nel lato opposto alla telecamera per coprirne lo sfondo. A questo punto, siamo pronti per creare finti riflessi e ombre. Creiamo un piano perpendicolare al pavimento e che sia fuori della visuale della camera, unwrappiamolo con il comando di default e aggiungiamo un nuovo materiale e carichiamogli una texture; questa dovrà essere una scena di interni o di mobili, dai colori dominanti rosso scuro/marrone ed avere abbondanza di toni neutri (varie tonalità di grigio/bianco) per essere compatibile con il nostro scenario. Il materiale sarà semplicemente un transparent shader, il cui input colore sarà l'output della nostra texture: verificare che vi siano abbondanti zone in trasparenza o semitrasparenza, e posizionarla in modo da influenzare il colore della luce nella scena. Se necessario, creare più di un piano con texture. Successivamente, creare e posizionare davanti alla luce sun una serie di parallelepipedi stretti e larghi (come delle travi di legno per intenderci) che creino delle ombre, assicurandosi che non blocchino mai completamente la luce diretta.



Per finire, posizionare dei "mobili" nella visuale della telecamera, che facciano da sfondo all'inquadratura; sono sufficienti anche forme estremamente semplici, con materiali dal colore base (ad esempio dei semplici cubi con un bevel modifier e un principled di default dal colore bordeaux) per simulare divani, sedie poltrone ecc... Non devono essere preponderanti perchè l'attenzione deve essere sul set da the e in più li sfoccheremo successivamente.

Raffiniamo l'inquadratura cercando di avvicinare quanto più possibile la camera al soggetto, abbassando la lunghezza focale (50mm) per accentuare l'effetto sfocatura, e regolando l'angolazione in modo da creare quello che si chiama "casual shot" (inquadratura spontanea) per ricreare un'atmosfera di quotidianità - nello specifico: inquadrare gli oggetti dall'alto verso il basso, evitando angoli perpendicolari e "tagliare" fuori alcuni elementi, come uno dei tovaglioli per dare l'impressione di "attività" al di fuori dell'inquadratura, inoltre si deve cercare di dare "profondità" alla scena, inquadrando gli oggetti in modo che siano posizionati a distanze diverse dalla telecamera (anche di poco, l'importante è che non siano allineati in fila di fronte allo spettatore) - e, per finire, nella tab proprietà della camera scegliere come focus l'oggetto teiera, selezionare l'apertura di tipo f-stop e abbassarla drasticamente (3.0, ma abbassandola ancora si può accentuare la sfocatura) finché solo il tavolo e gli oggetti su di esso siano a fuoco







Francesco di Buono

Utente di Blender dal 2013, ho lavorato nel campo dell'archviz e nella stampa 3d. Attualmente lavoro come modellatore freelance per visualizzazioni interattive di architettura in Unity e Unreal



Alessio Cassaro

Intervista di Cristina Ducci



Ciao Alessio, raccontaci qualcosa di te e della tua professione

Mi chiamo Alessio Cassaro e sono Direttore di Antaless Visual Design.

Lavoro nel campo della grafica animata da oltre 10 anni e da alcuni anni mi sono specializzato in video mapping.

Ho creato grafiche animate per molti spettacoli e concerti per tutta Europa.

Dopo aver terminato gli studi ho lavorato in molti teatri come tecnico luci-audio-video.

Successivamente, ho lavorato per diverse società di video mapping conosciute in tutto il mondo.

Nel 2013 ho fondato Antaless Visual Design con lo scopo di condividere le mie emozioni con il mondo dell'arte visiva.-

Come hai iniziato ad occuparti di video mapping?

Questa specializzazione è iniziata come un hobby, per divertimento piuttosto che per lavoro, ma ne sono rimasto completamente rapito e ho deciso di farne una professione.-

Quali sono i softwares che utilizzi?

Dipende molto dalla richiesta che arriva da parte della committenza, ma in genere cerco di utilizzare Blender in tutte le fasi della produzione, credo fermamente in questo software, e nelle potenzialità della community per lo sviluppo dei suoi strumenti.-

Come è nata la tua passione per Blender?

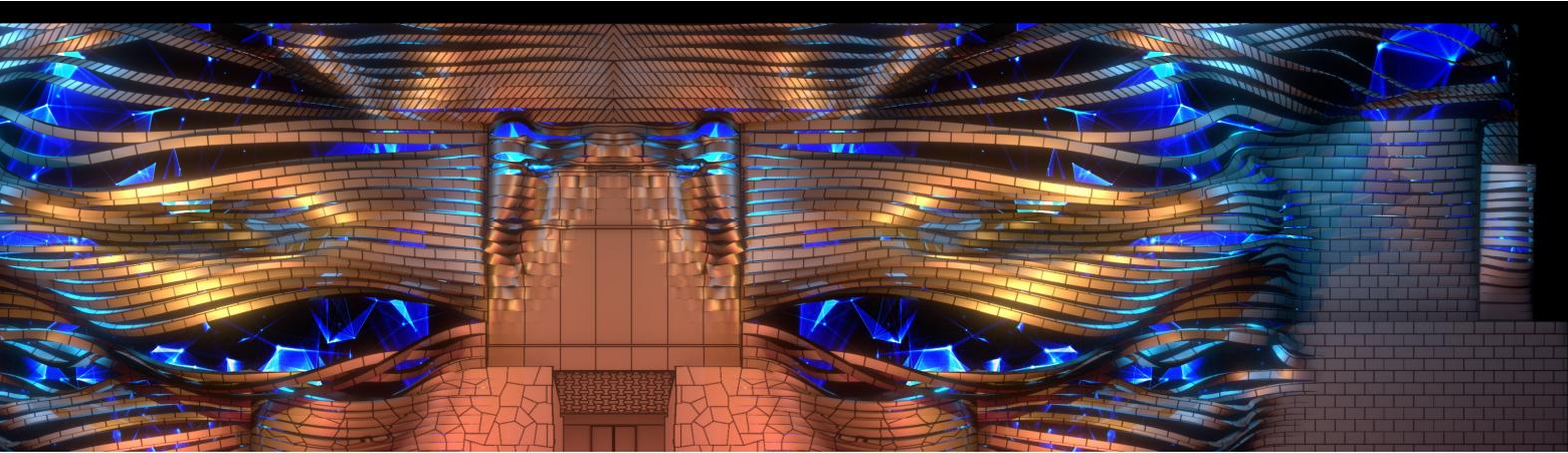
Tanti anni fa, ero davvero alle prime armi, feci un colloquio presso uno studio grafico.

Mi fu richiesto espressamente l'utilizzo del software Autodesk 3ds max, poichè le licenze erano state acquistate ed avrebbero assunto solo gente che sapeva usarlo e che non comportasse ulteriori costi di licenza per l'azienda.

Ovviamente non continuai il colloquio, ma rimasi determinato a esprimere la mia creatività.

Decisi di studiare Blender per non avere altri problemi di discriminazione sul software.

In seguito rimasi totalmente affascinato e ad oggi sono dodici anni che lo uso.-



Pensi che Blender sia un buono strumento per la realizzazione dei tuoi video?

Dal mio punto di vista ritengo che Blender non sia soltanto un ottimo strumento per il video mapping ma in generale un potente alleato per ogni creazione CGI.

Non ha nulla da invidiare a nessuno dei software 3d in commercio, semplicemente ci stiamo evolvendo secondo il mood BFoundation ed Eevee ne è uno degli esempi più concreti.-

Quanto è importante la parte creativa rispetto a quella tecnica?

Sinceramente credo siano direttamente proporzionali, l'una incide sull'altra e senza una l'altra risulterebbe scarna.

La creatività è molto importante per elevare ad un livello più alto la tecnica-

Ci racconti a grandi linee il processo per la realizzazione di un video?

Posso dire che in realtà è molto semplice da spiegare.

Si parte da una "matrice" ovvero una maschera di proiezione su cui basare le proporzioni, la distanza e la risoluzione della proiezione video spesso sono ricavate dalla stessa fotografia dell'architettura su cui avverrà lo show.

In seguito, per coloro che lavorano in 3d come nel mio caso, attraverso un lavoro di fotogrammetria e modellazione si dovrà riuscire a creare il modello tridimensionale della stessa architettura.

Stabilite le animazioni si prosegue con renderizzare le scene e compositarle secondo il proprio gusto o scelta artistica.-

Per finire vorremmo sapere se sei legato in particolar modo a qualcuno dei concept che hai proposto negli anni

Sicuramente, avendo scritto ogni singolo concept personalmente sono legato a tutti in modo indissolubile.

Ognuno di questi ha rappresentato per me un momento creativo ed ha richiesto un particolare studio ed approfondimento.

Posso dirvi però che sono particolarmente legato all'ultimo ENTELECHY (Entelechia), presentato a Bucarest in Romania il mese scorso. Esso, oltre a descrivere il senso che ho voluto dare al progetto, incarna lo spirito creativo di Antaless Visual Design.

Vorrei concludere questa interessante intervista lasciandovi alla lettura di questo concept ed alla visione della registrazione video compresa di video nativo (produzione originaria)

ENTELECHY

Il progetto grafico si basa sulle idee filosofiche di Aristotele.

In particolare sul concetto di Entelechia, cioè la perfetta attuazione da potenza ad atto.

Materia e potenza si equivalgono nell'idea di Aristotele, essendo la materia per definizione facoltà di assumere forme.

Il termine Entelechia esprime la possibilità propria di qualcosa di trasformarsi in qualcos'altro, attraverso un viaggio fatto di continui mutamenti, con il fine di raggiungere il suo pieno potenziale.

"La luce emessa dalla proiezione, rendendoli visibili, trasforma in atto i colori e le forme dell'architettura, dallo stato potenziale in cui si trovano nell'oscurità"



ARCADIA

Il progetto grafico descrive una visione personale e geometrica dell'Arcadia antica, una regione storica della Grecia che si presenta montuosa, disabitata per via della sua topografia.

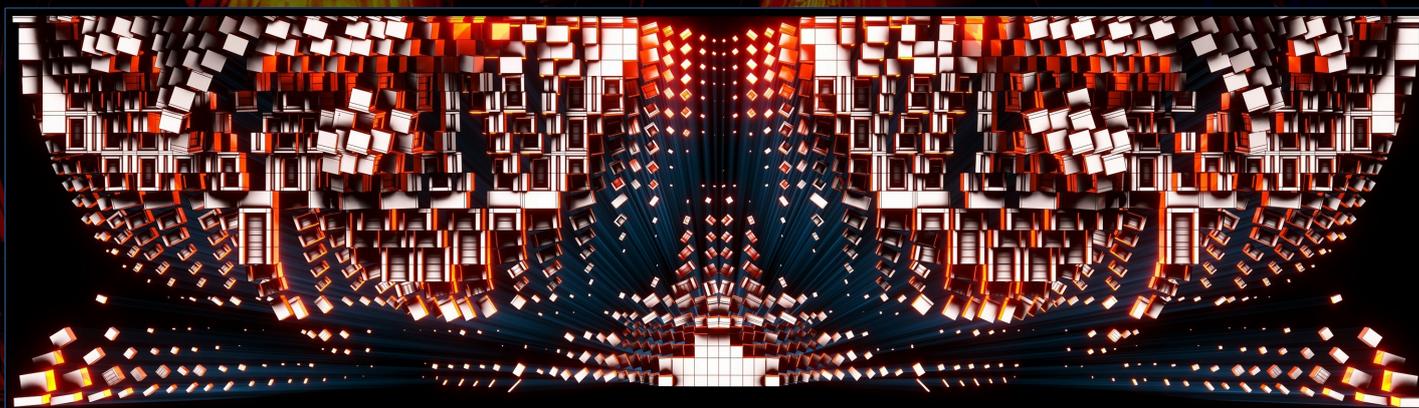
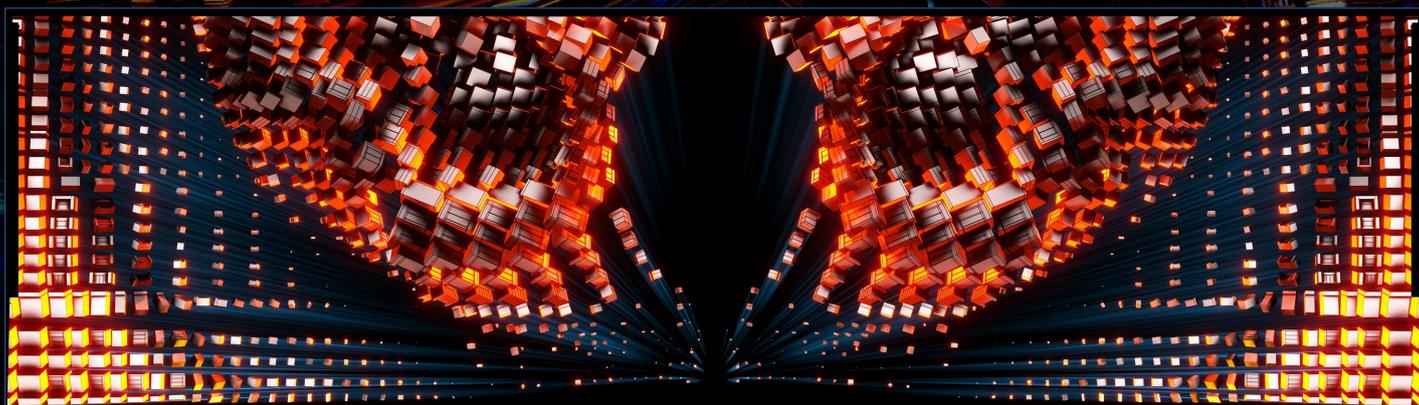
L'Arcadia è stata oggetto delle arti visuali e della letteratura, tanto da esser stata elevata a topos letterario.

Viene descritta come una sorta di paradiso terrestre, un mondo idilliaco abitato però solamente da entità sovranaturali; a costeggiarla, immagini di bellissime ninfe che giocano e corrono in una rigogliosa foresta.

Fonte d'ispirazione per pittori, scultori e letterati anche nei secoli successivi, l'Arcadia, in quanto emblema del classicismo e quindi del buon gusto, diventa una risposta all'ampollosa letteratura barocca attraverso la fondazione, nel 1690, dell'Accademia dell'Arcadia di Roma.

Il progetto parte dallo studio della topografia dell' Arcadia antica tradotta nella topologia del modello 3D della facciata del Palazzo della Farnesina e prende ispirazione dall'immagine del Bosco Parrasio, sede storica dell' Accademia dell'Arcadia di Roma.

Sono così i colori, le luci e le ombre a diventare entità sovranaturali che ridisegnano parte di un mondo illusorio che ho voluto rappresentare.



<http://www.antalessvisualdesign.com/>

<https://www.facebook.com/antalessvisualdesign/>

<https://www.youtube.com/user/AntalessGroup>



Luca Di Cecca

Intervista di Nicolas D'Amore

Arturo e il gabbiano è un cortometraggio in animazione realizzato interamente in Blender. Il corto è stato diretto da Luca Di Cecca, utente blender da diversi anni e in squadra con lui Nicolas D'Amore e Umberto Salerni che si sono occupati principalmente dell'animazione.

Luca, da dove nasce l'idea di un cortometraggio con Blender?

Fin da quando studiavo all'università, con amici e colleghi di studio, sono stato sempre affascinato dall'animazione.

Già durante gli studi avevo realizzato corti in animazione, principalmente in stop motion.

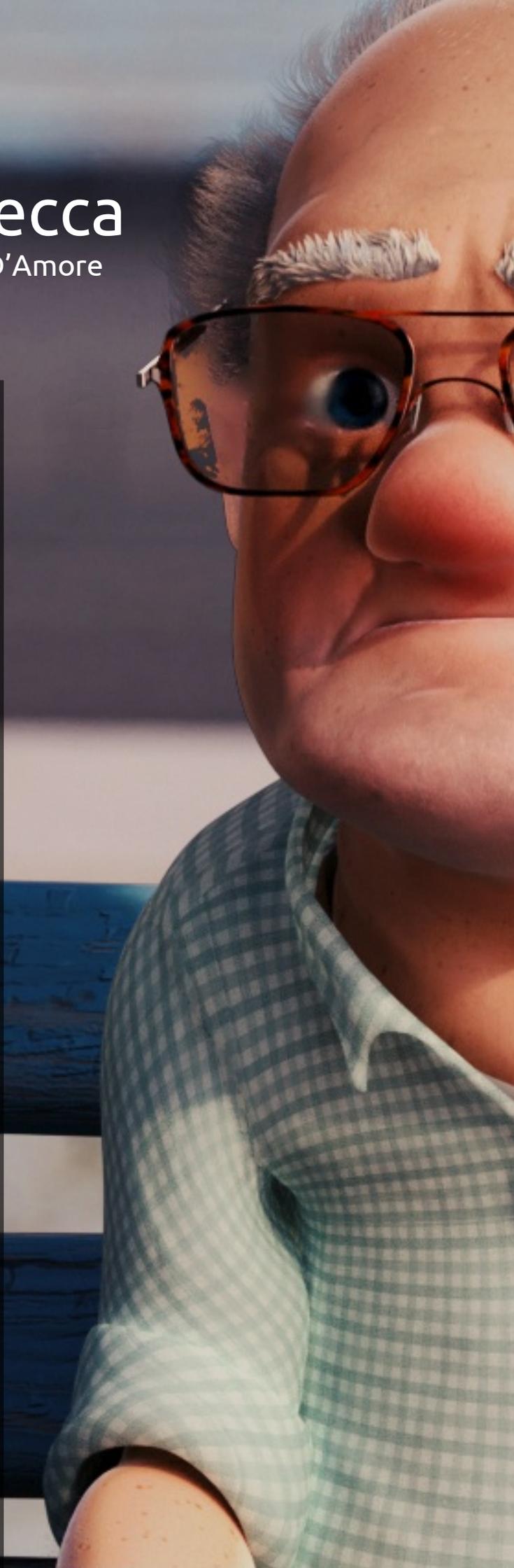
Dall'animazione stop motion poi la Cgi, dapprima altri software, poi l'amore per Blender.

Questo corto in animazione ha origine tempo addietro: avevo chiesto a un mio amico, Alessandro Izzi, critico cinematografico e saggista, di scrivere un piccolo soggetto per un corto in animazione. L'idea era subito sfumata perché all'epoca ero totalmente incapace di gestire una intera produzione. Poi, nel corso degli anni, con i lavori con la Light and Color in particolare, i tempi sono diventati maturi per la realizzazione.

Quali sono stati i primi step nella realizzazione del corto?

Dapprima sviscerare la sceneggiatura, cercando di semplificare al massimo, in base alle possibilità produttive (ho fatto la maggior parte del lavoro con un laptop).

Storyboard e concept art e poi tanto, tantissimo tempo sugli asset e look Dev. Circa metà della produzione è andata via nella realizzazione di 2 personaggi, 5 scenografie e centinaia di props.





Come hai gestito la produzione del cortometraggio?

E' stata una gestione molto lineare. Ho delineato un Gantt di massima e in base a questo schema produttivo ho cercato di capire di quali figure professionali avevo bisogno.

La fortuna è che lavorando nel campo dell'animazione da svariati anni molti dei miei amici fanno questo mestiere ed è stato semplice trovare e chiedere una mano ad amici modellatori, animatori, musicisti ecc ecc.

Qual'è stata la difficoltà più grande?

La difficoltà più grande è stata mantenere viva la voglia per un anno e mezzo di produzione. Vedere il risultato solo dopo così tanto tempo, immaginarlo per più di un anno e non demordere mai.

Sicuramente avere chiara la motivazione ha aiutato molto, ma soprattutto gli amici che mi hanno aiutato in questo VIAGGIO e che hanno sentito il progetto loro e lo hanno arricchito.

La presentazione al talk di Amsterdam è stato un giusto punto di arrivo, una piccola ciliegina sulla torta per concludere questo progetto.

Il cortometraggio è ora in distribuzione nei festival e sarà reso on-line nel prossimo ottobre.



Mike Pan

Intervista di
Nicolas D'Amore

Qualcosa su Mike:

Mike si diletta nell'arte 3D, nella programmazione e nella creazione di giochi. Con 12 anni di esperienza in Blender e un vivo interesse per la grafica 3D, ha lavorato per l'Università della British Columbia in Canada, l'Institute of Clinical Physiology (CNR Pisa) in Italia e la Harvard Medical School.

È supervisore VFX per il cortometraggio "Sunny and Gerd in Spring Cleaning", e puoi anche vedere i suoi lavori in programmi TV come Man in the High Castle e Silicon Valley.

Come hai cominciato con il 3D?

Ho iniziato con il 3D a metà degli anni 2000, in un momento in cui i personal computer stavano diventando abbastanza potenti da poter davvero farlo.

Penso che sia stata una combinazione di visualizzazioni di Winamp, sfondi di Digital Blasphemy e film animati come Toy Story che mi hanno interessato alla computer grafica. Ero solo un adolescente che voleva ricreare il suo proprio sfondo!

Come hai conosciuto Blender?

Ad essere sincero, non ricordo come ho scoperto Blender. La sua leggerezza nel suo download e le sue prestazioni veloci hanno reso molto più semplice iniziare (Blender 2.27 era di poco più di 2 MB!)

Il mio lavoro con Blender ha alimentato il mio desiderio per un computer sempre più veloce, da lì mi sono interessato alla grafica in tempo reale, il che mi ha attirato ulteriormente a ciò che è possibile con fare con la CGI.





Come hai imparato quello che sai oggi?

Sono autodidatta, ho iniziato con molti tutorial online e ho trascorso molto tempo sui forum della community quando ho iniziato, questo mi ha permesso di imparare rapidamente Blender. Pubblicare il mio lavoro per ricevere feedback è ciò che mi ha motivato ad andare avanti. Più tardi, ho iniziato a espandermi e ad imparare da altre comunità CG, il che mi ha permesso di capire davvero come funziona l'industria nel suo insieme. Quando ho iniziato, Blender aveva una piccola comunità, quindi era abbastanza facile essere mediocri ma essere ancora riconoscibili. Oggi, la quantità di talenti che utilizza Blender è cresciuta probabilmente di 1000 volte. Anche il campo CGI è diventato molto più complesso, quindi l'apprendimento è infinito.

Quali consigli daresti a chi inizia oggi?

Impara ad essere efficiente. Non pensare di poterti far strada "forzando le cose" attraverso un modello o una scena complessa. Potresti - ma non è quello che la gente si aspetta che tu faccia nel mondo reale. Se c'è qualcosa che ti ritrovi a fare molto, cerca come gli altri affrontano lo stesso problema. O prova a pensare a modi alternativi. Essere un artista 3D significa fingere di ricreare la realtà, imbrogliare l'occhio, quindi trova il modo di farlo.

Perché Blender e quali altri software utilizzi?

Come studio individuale, mi piace la filosofia di fare tutto con Blender. La sua completa integrazione significa anche che c'è meno pipeline da gestire, posso modellare, assegnare i materiali, animare e renderizzare tutto attraverso un'unica interfaccia in modo abbastanza efficiente. Uso anche Substance Painter e Houdini quando necessario. Uso Blender semplicemente perché credo sia lo strumento migliore per quello che devo fare. Il software è veloce, non si blocca mai in modo imprevisto e ha un supporto straordinario da parte degli sviluppatori.



Nuccio Canino

Intervista di Irriverender. Arch. Bonni

Oggi abbiamo l'onore di intervistare Nuccio Canino in merito al lavoro fatto, con alcuni software e addon tra cui Blender, su "L'uomo del Labirinto", nuovo film con Dustin Hoffman. Tramite un appassionante storytelling, Nuccio ci racconterà la genesi del film da un punto di vista tecnico (la scelta dei software, degli addon e delle tecniche adeguate), ma non solo. Ci narrerà anche come la "sua città", multiforme e densa di citazioni cinematografiche e fumettistiche, ha preso forma grazie a Blender, e di come sia riuscito a darle vita da innumerevoli prospettive.

Panoramica

L'intento del discorso è dimostrare come un piccolo team, con l'ausilio di un software completamente gratuito (OpenSource), possa portare a compimento un grosso progetto cinematografico con più di 200 shots di effetti digitali complessi.

Il Film

Il film è **L'uomo del Labirinto**, film tratto dall'omonimo best seller di **Donato Carrisi**. Già dalle prime riunioni il regista aveva ben in mente il mood preciso per il suo secondo lungometraggio. Le **ambientazioni** dovevano essere un non luogo, un non tempo, dovevano rendere spaesati gli attori e chi vedeva il film.

Una città **di giorno in preda al caldo soffocante e disabitata, di notte viva e popolata da gente dormiente**. Si trattò subito di una sfida tra le idee, le immagini del regista e io che cercavo di recuperare visioni dal mio bagaglio culturale.

Prima delle riprese i produttori avevano proposto di girare molti esterni a Macao o Hong Kong. Col passare del tempo, però, il budget fu ridimensionato di molto. Vennero da me e mi chiesero **un'intera città in 3D**.

Avevo in mente il look e i primi giorni sul set ero già pronto con molte **previsualizzazioni**. Una cosa ho capito negli anni: bisogna sempre mostrare i muscoli alla gente con cui si lavora. E' il modo migliore per essere ascoltati. E feci così.

Andai sul set con delle tavole di quella che era la mia città per il film. I miei incendi, le mie campagne desolate. Bastò far vedere le tavole alle persone giuste e in poco tempo quello che doveva essere il film stava prendendo piede sul set, tra macchinisti elettricisti, fino ai runners. Tutti avevano un'idea più chiara del mondo in cui si svolgeva la storia.

Blender, in questo, è stato il software perfetto. Facevamo sopralluoghi tornavo in ufficio e potevo allestire i set e renderizzarli col suo motore in tempo reale.



Un giorno andammo in una via anonima dell'Olgiata. Voleva che quella fosse la **via del rapimento, del fatto più importante del film**. Era una semplice via come tante...

Quella sera, dopo due ore di sopralluoghi tornai a casa, ripresi dei modelli di palazzi che avevo, negli anni, modellato o comprato e...



Questa era la forza di Blender. Lo sky generator di **Fusion**, **Eevee** il suo motore di rendering in tempo reale, **Krita** e il giorno dopo il regista poteva mostrare sul set agli altri collaboratori dove sarebbe stato il rapimento.

Intanto, il regista mi lanciava nuove sfide. Devo ammettere che non ho lavorato al film per dedizione professionale ma l'ho vissuto come **una continua sfida alla mente creativa di un grande scrittore**.

Da copione molti dei camera car erano descrittivi, nessuno di essi era un semplice tramite tra shots. Il mondo là fuori doveva vedersi attraverso il lunotto della Saab del protagonista.

Questa è stata la sfida più grande: **rappresentare tutto un mondo fuori dal lunotto, tra i passanti del tettuccio e i finestrini.**

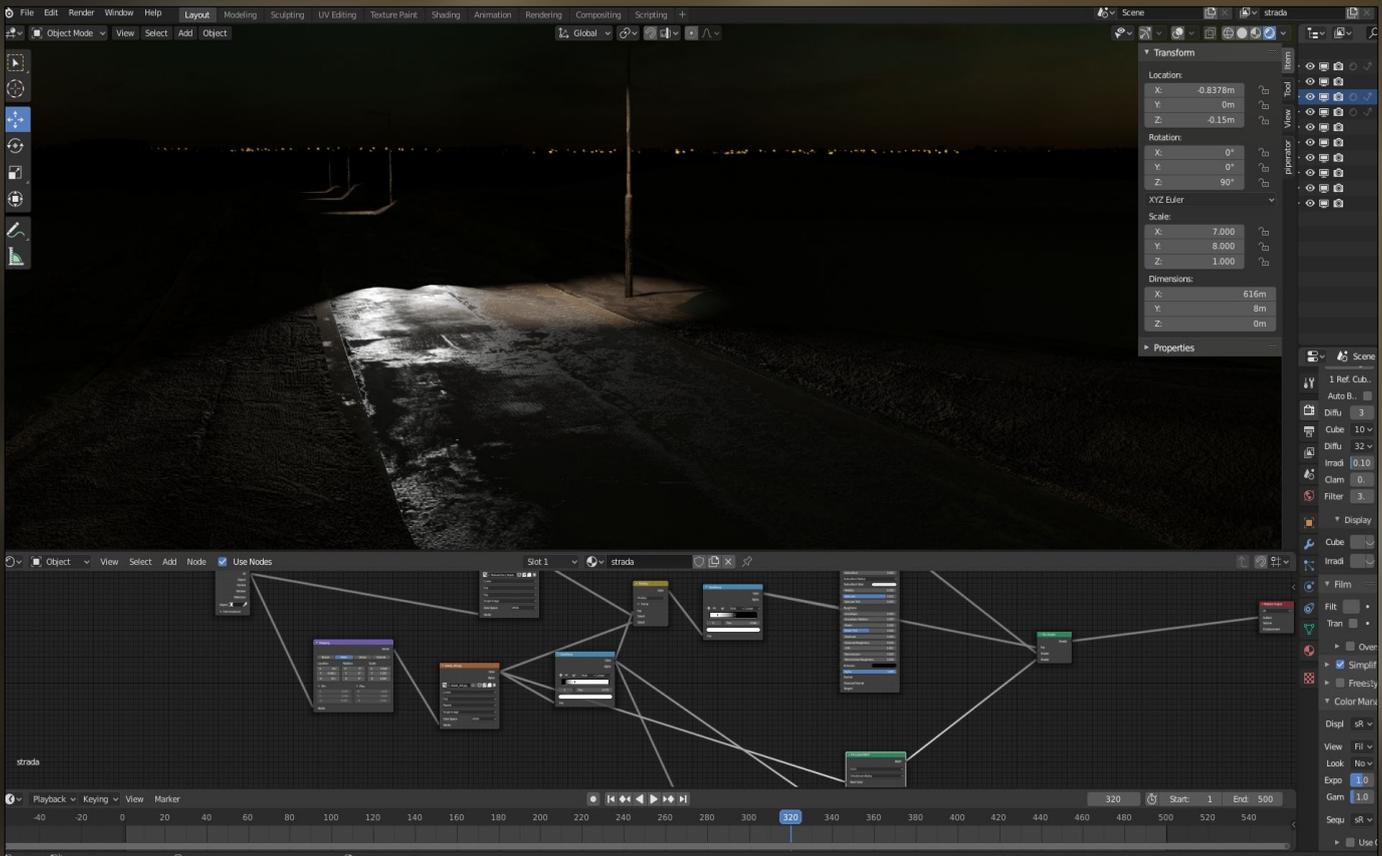
Essendo stato ridimensionato il budget, i camera car dovevano essere molto pochi e tutti laterali. Al regista non bastava, ovviamente. L'ultimo giorno delle riprese fu il giorno della Saab; furono girate oltre 50 inquadrature e l'attore in macchina ripreso lateralmente, frontalmente e da dietro. Lì la mia ansia saliva, perché non avevo preventivato che la mia città fosse inquadrata così a fondo e da ogni angolo. Era l'ultimo giorno di riprese, la scena di fine film peggiore della mia vita.

Il giorno dopo mi misi subito a pensare come risolvere il problema dei camera car. Comprai dei moduli, dei props di nuclei abitativi e con il sistema particellare di Blender e il suo potentissimo array avrei costruito interi palazzi proceduralmente e dal peso irrisorio per la renderfarm usando le istanze di rendering. La prima volta che allestii la città fu per un camera car **in notturna lungo una via in aperta campagna.**



Tutto per queste scene era digitale, la strada, l'erba ai margini, gli alberi lontano. **Tutto era renderizzato sulla mia sola macchina**, con *eevee* in tempo reale e tutto funzionò.

Nessun costo di licenze, e con la farm della mia azienda che poteva continuare a lavorare su altri progetti



Questa era una delle prime visualizzazioni in tempo reale che filmavo con il cellulare e mandavo al regista per farmi approvare il **grado di "romanità" delle strade**. Potevo muovermi in quell'environment fatto di displacement, reflection map e far vedere come sarebbe stato il suo **viale verso la città**.

Quella via di campagna arrivava alla città, non potevo sfuggire, dovevo allestirla e, per farlo, usai le stesse metodiche di quando anni prima creai in *3dsMax* la Roma dei Gladiatori. Tramite *openMap* (una sorta di map di google open source), tracciai le linee di un qualsiasi quartiere di **New York**, da quella porzione mi ero estratto le mappe di elevazione e sostituii i cubi con dei proxy più evoluti dei palazzi. In teoria potevo spostarmi su un enorme quadrante di 5 km in lungo e in largo, poter girare l'angolo e ritrovarmi altri palazzi proxy.

La prima volta della città

Era fatta! al regista piaceva, al DOP piaceva molto, persino agli attrezzisti che il giorno delle riprese mi chiedevano cosa avessi messo al posto del verde.

Era la mia città, un mix perfetto tra **Ghost in the Shell** e **Vanilla Sky**.

E altre inquadrature nacquero, dei lunghi laterali della città vista dal finestrino, delle infilate viste dal lunotto posteriore. Il film era stato girato in anamorfico e nemmeno questo fermava Blender, impostazioni di rendering 2:1.

Settaggi:

- camera alexa e lenti Cooke
- boke e lens anamorfici.
- Tutto in camera
- tutto in rendering con motion blur settando lo shutter come la MDP reale e DOF.



Il regista non si fermò. Volle usare dei camera car notturni per far vedere allo spettatore come **potesse essere quella città di notte**, non più vista da lontano, ma **tra le sue vie**.

Provai prima a fare una **correzione colore** della città di giorno. Provai con *ID* di selezione a **illuminare i vetri dei palazzi**. Non funzionava. Un fine settimana passato a studio. Dovevo illuminare quelle vie in **modo procedurale**, non avevo il tempo di mettere luci.

Lavorai di shading.

Avevo nominato con accortezza ogni materiale, e ogni materiale aveva il proprio indice; duplicai parte dei materiali e tutti quelli che erano **vetri di finestre, cartellonistica, vetri di lampade spente** li aggiunsi a uno shader emissivo.

I colori non erano casuali ma, con il nodo di color ramp, creai delle palette colore usando immagini di città come generatori. Avevo delle color ramp di diverso tipo, stile **Gotham City, New York, Hong kong, Ghost in the shell, città di perdizione e prostituzione**. Scelsi l'ultima palette ovviamente.

Le sfide ovviamente non erano finite. Nella mente del regista, sin dalla prima stesura del film c'era **un grande incendio, lontano sopra le colline**.

Quando andammo a vedere la location feci un video dal finestrino della macchina. Con il mio cellulare.

Avrei dovuto provare e trovare un modo per **fare incendi in mezzo agli alberi senza nessun green back**. Pensai subito che il modo migliore era togliere gli alberi reali e farli in 3D, più grandi, e che mi potessero servire per tenere il grande incendio nascosto per 3 inquadrature fino alla scoperta finale dietro ai grandi alberi su una curva. Lì dietro lo spettatore avrebbe visto per la prima volta **come stava bruciando il mondo**.

La scena venne ripresa al **tramonto**; mi serviva la luce per poter trackare il movimento del furgone da cui avevamo fatto le riprese. Il trak lo avevo trasferito alla clip della *saab* girata ferma su greenback, un lieve offset sulle chiavi e l'illusione che la *saab* fosse in quel posto era totale.

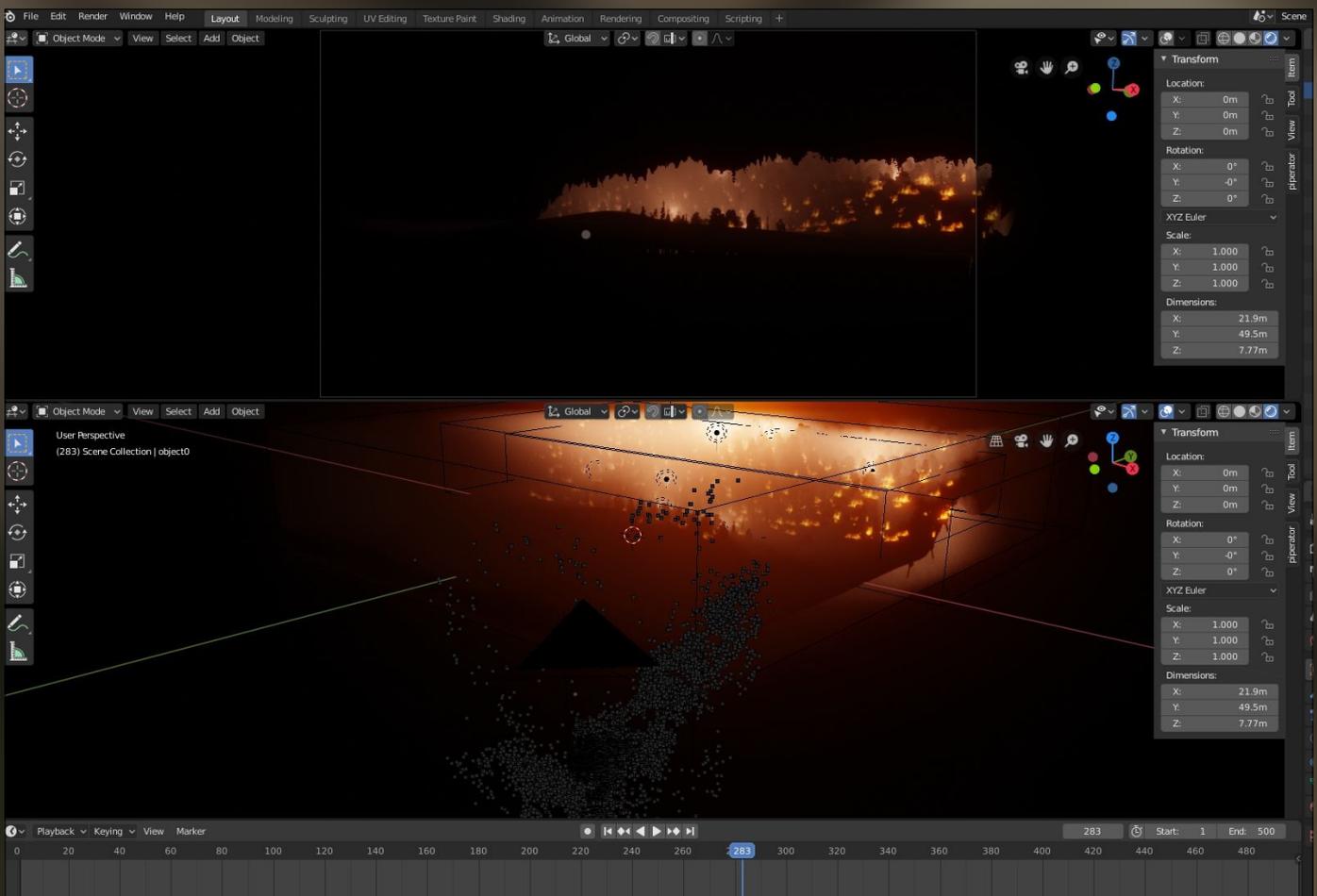
Intanto la scena girata con il cellulare il giorno dello shooting mi serviva a **convincere il regista che quello era il posto ideale**, mentre lui ne preferiva un'altro.



Questa era il compositing della scena fatta con il cellulare. Il regista era entusiasta: se era così credibile fatta col cellulare, già pregustava come sarebbe venuta col girato finale.

Quegli **incendi sulle colline** erano in tempo reale, in eevee, potevo girarci attorno, andar avanti e indietro sulla timeline e spostare in tempo reale gli alberi istanziati con particelle e clips di incendi che nascevano da mappe spennellate sulla mesh.

Mediante il track incluso in Blender avevo estratto dalle riprese del cellulare una **nuvola di punti** che passata in *Meshroom* (altro software openSource) mi aveva generato una geometria proxy su cui scatterizzare alberi e clip di incendi. Mediante il *weightPaint* generavo mappe diverse che diventavano zone definite con cui distribuivo alberi e incendi.



Questa era la scena dentro Blender, volumi e fog erano in real time la geometria proxy sempre più rifinita con il suo sculpt interno e via via avevo allestito decine di alberi ai lati della macchina che usavo per nascondere il grosso incendio in lontananza fino poi a scoprirlo dietro una curva di alberi in 3D.

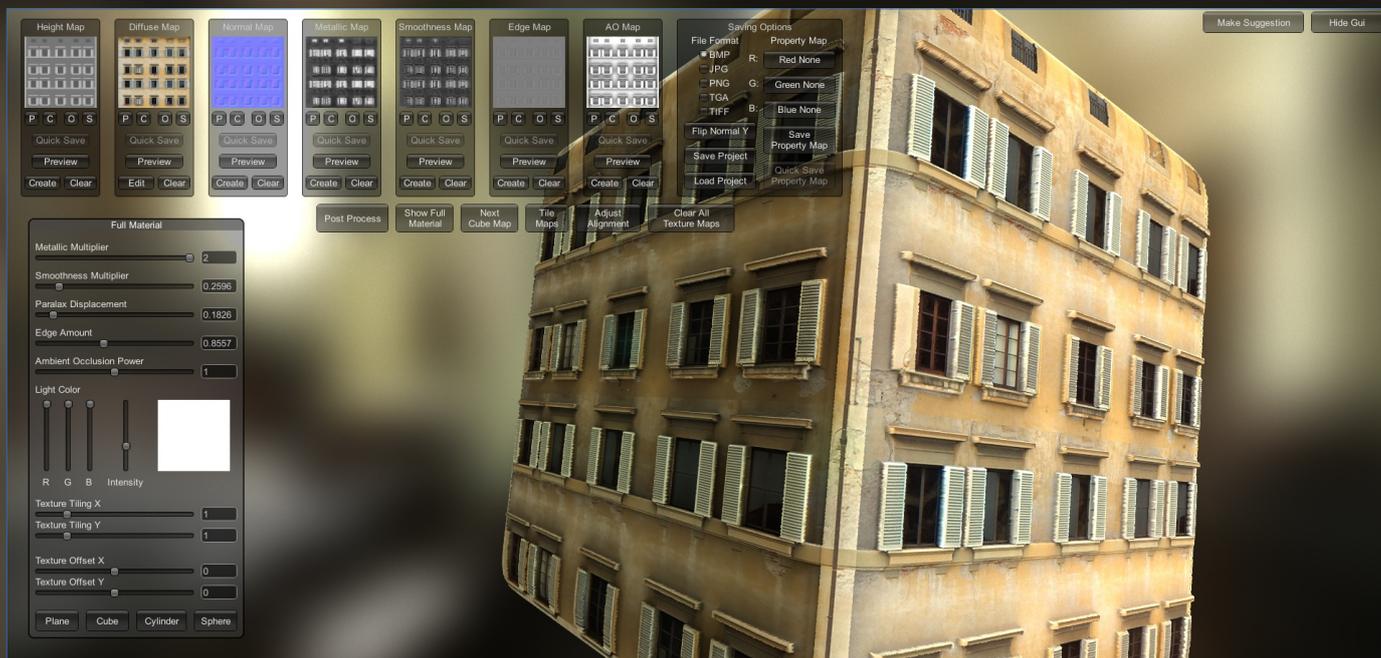
Gli interventi sul film non erano finiti. Oltre i grandi *environments* per i camera car dovevo allestire anche la città vista da dietro le finestre delle varie abitazioni.

Vi era **l'appartamento del detective**, che abitava in periferia, e la città doveva apparire al di là di un grande fiume. Vi era **l'appartamento di una prostituta** in pieno centro con la città vista dal 22esimo piano e vi erano delle enormi vetrate in un albergo a piani molto più bassi.

Ho dovuto **allestire la città in varie viste, condizioni di luce e situazioni.**

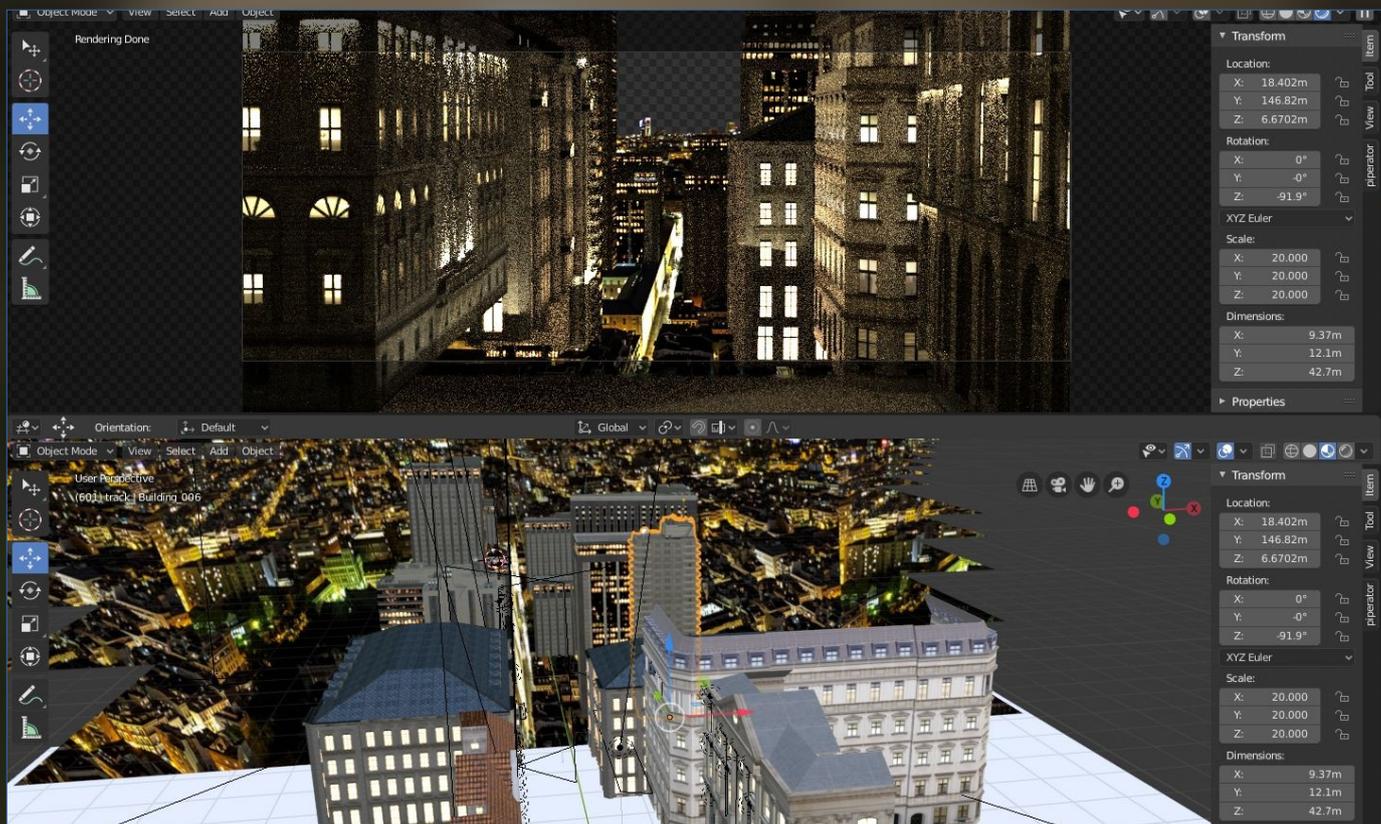


Per la stanza dell'albergo avevo diverse facciate neoclassiche nate da foto ortogonali e poi proiettate su mesh estruse a dovere, le avevo usate molte altre volte in passato e anche per questo film hanno funzionato benissimo... ma ovviamente non bastavano. Quando il film è **anamorfico**, bisogna abituarsi a **inquadrature sempre molto larghe**, a **campi lunghi** che fanno vedere a km di distanza. Usai Materialize un software Open Source per generare **mappe di displacement** e normali.

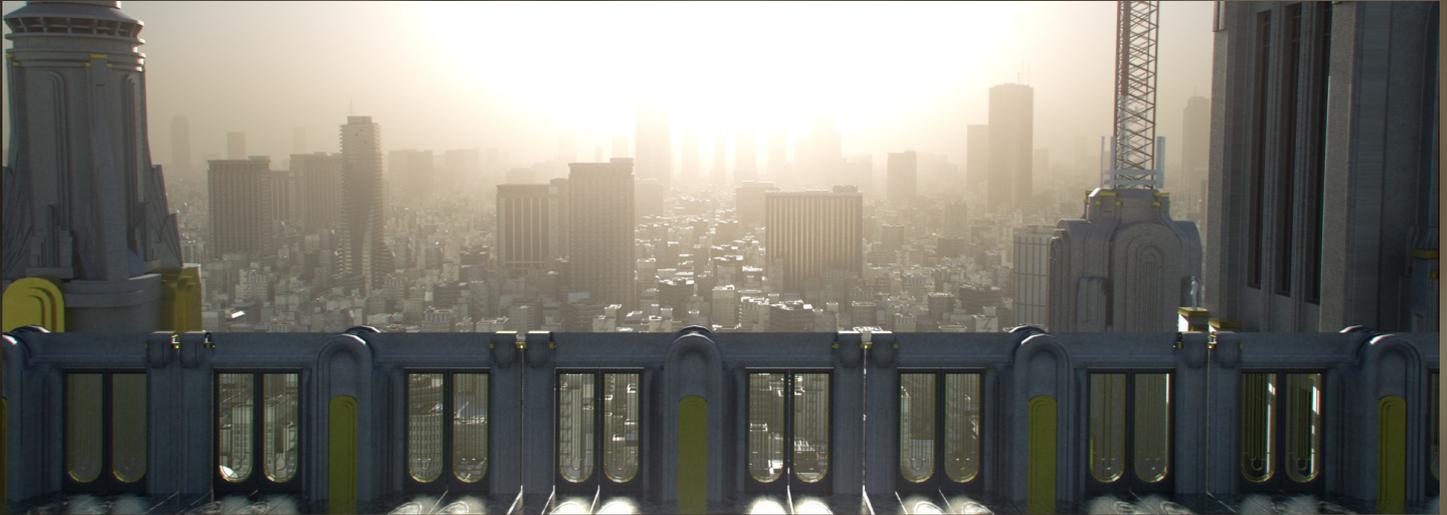


Altri palazzi erano pronti. Al regista facevo scegliere le facciate dal sito CGtextures e poi diventavano modelli di allestimento nella sua città mediante displacement e normal map.

La città così prese corpo, molto newyorkese, labirintica caotica che contrastava col silenzio dentro e dietro le grandi vetrate.

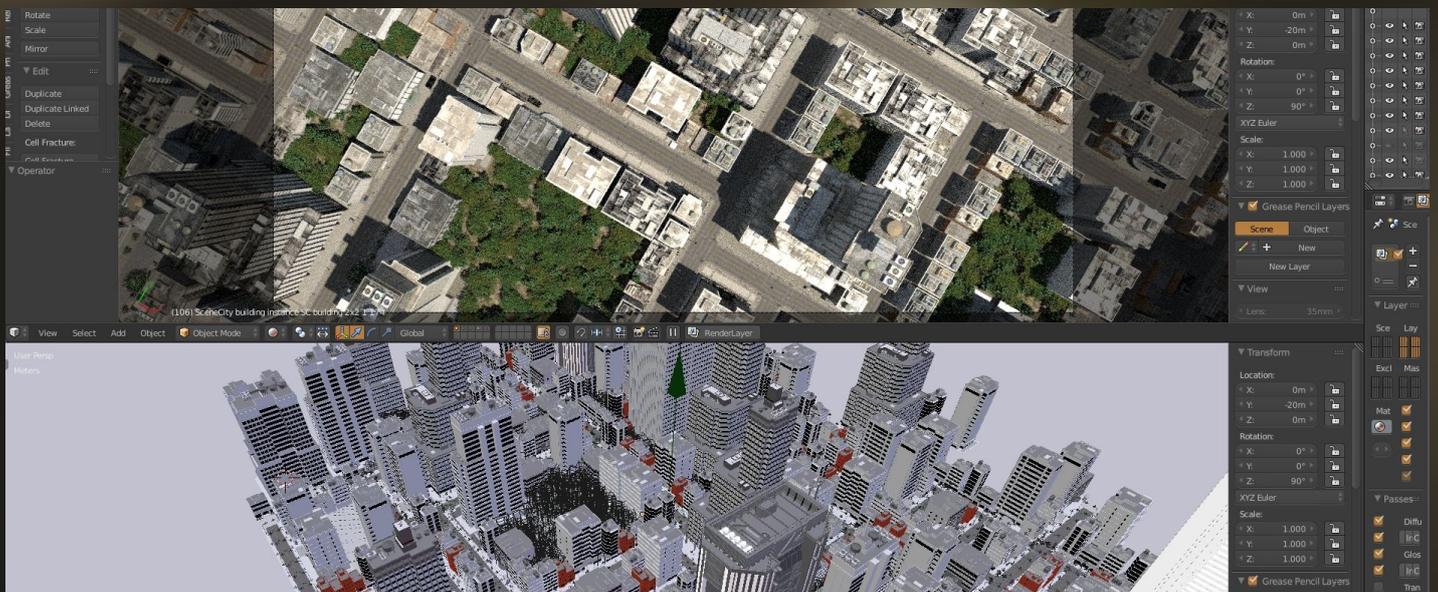


Ennesima sfida era **la città vista dalla casa della prostituta**. Dal **22esimo piano** di un palazzo nella zona deko della City. Modellai delle balaustre come se guardassi lo strapiombo e comprai dei modelli di guglie che dovetti solo "shedare". La città oltre le guglie la tirai su mediante delle mappe disegnate in Krita, vie, **box di quartieri**, **incroci e tante particelle che distribuivano** **randomicamente i palazzi in un container con shading volumetrico**.



Non avevo tempo per renderizzare la città in tutte le inquadrature, e ho quindi creato un **allestimento di 270 gradi attorno alla camera**. Ho renderizzato un latLong (un rendering VR) e in nuke potevo spostare la camera tranquillamente e l'environment fuori si comportava sempre correttamente con **una città che si muoveva coerentemente con parallasse dietro le finestre**.

Sembrava potessi mettere a riposo Blender e dedicarmi al compositing delle scene a due settimane dalla color, ma non fu così. Il regista mi chiamò al telefono dicendomi che aveva bisogno della **città in una vista top**. Gli serviva per i titoli di testa per **farne sentire la grandezza**. Le varie inquadrature della città **funzionavano ad altezza uomo** ma **l'urbanizzazione dei palazzi non era sufficiente per vederla dall'alto**. Cercai modi diversi e chiesi sui vari forum di blender e qualcuno mi diede la soluzione. **SceneCity** un potete addon da 70 euro che mi permise di **finalizzare la città come la voleva il regista**.







Si tratta di un potentissimo plugin che permette in modo nodale e procedurale di tracciare delle spline come tracce lungo cui far scorrere strade. La sua potenza è che mi ha permesso di distribuire i palazzi in base a delle mappe colore e potendo usare i miei palazzi. Con lo stesso approccio potevo distribuire alberi e piccoli props tipo lampioni e segnali stradali.

Il *lighting* lo feci con una sun che ruotava stile timelapse, segnando il passare delle ore il tutto animando in compositing un cielo procedurale nato in fusion e poi usato in blender per il *tonal mapping*.

Ovviamente gli interventi non erano finiti. Ho dovuto **ricostruire in eevee una parete di un ospedale** perché in montaggio non funzionava quella reale e persino una porta che al regista non piaceva.

L'uso di **Blender, Krita, MeshRoom, Materialize** è stata una sfida mia nei confronti della società in cui lavoro. **Un solo operatore 3D**, il sottoscritto, **due compositor** in sede e **due remoti** per poter portare a compimento **un film di oltre 230 shots** di cui oltre la metà con interventi 3D. Le mie **capacità da generalista** sono state fondamentali sul set, nelle scelte fatte col regista e il DOP, e Blender è stato lo strumento ideale.

Conosco altri software, ma **con nessuno di questi sarei riuscito a ottenere tutto questo**. Inoltre la comunità online è davvero molto attiva, per ogni problema trovavo una rapida soluzione nei vari forum.

Ma la cosa più importante è stata quella di **non rallentare la farm** e gli altri progetti dell'azienda. Ho renderizzato molto in eevee e solo sulla mia macchina mentre, **per i rendering classici**, ho usato **solo sei macchine su 200 in nostro possesso**.

Con i suoi moduli di *sculpt, paint textures, fx, shading nodale e procedurale* non ho mai dovuto usare altro che non fosse blender.

E' il primo film fatto con Blender, ma non sarà l'ultimo.

Sto finalizzando un altro film per ragazzi in cui ho fatto uso massiccio di Blender in ogni sua forma e incarnazione.

Ah, dimenticavo una cosa non da poco... Blender **era in beta** durante tutta la produzione!

Gallery



Immagine di Fiori Andrea



Immagine di Dario Demi



Immagine di Luca Di Cecca



Immagine di Alessandro Passariello



Immagine di Marco Iacopetti



Immagine di Cristina Ducci



Immagine di Davide Picardi



Immagine di M. Antonio Renzulli



Immagine di Max Puliero



Immagine di Cristina Ducci



Immagine di Davide Picardi



Immagine di Cassettari Roberto



Immagine di Zecca Davide



Immagine di Liguori Aldo



Immagine D'Onofrio Manuel



Immagine Valenti Enrico

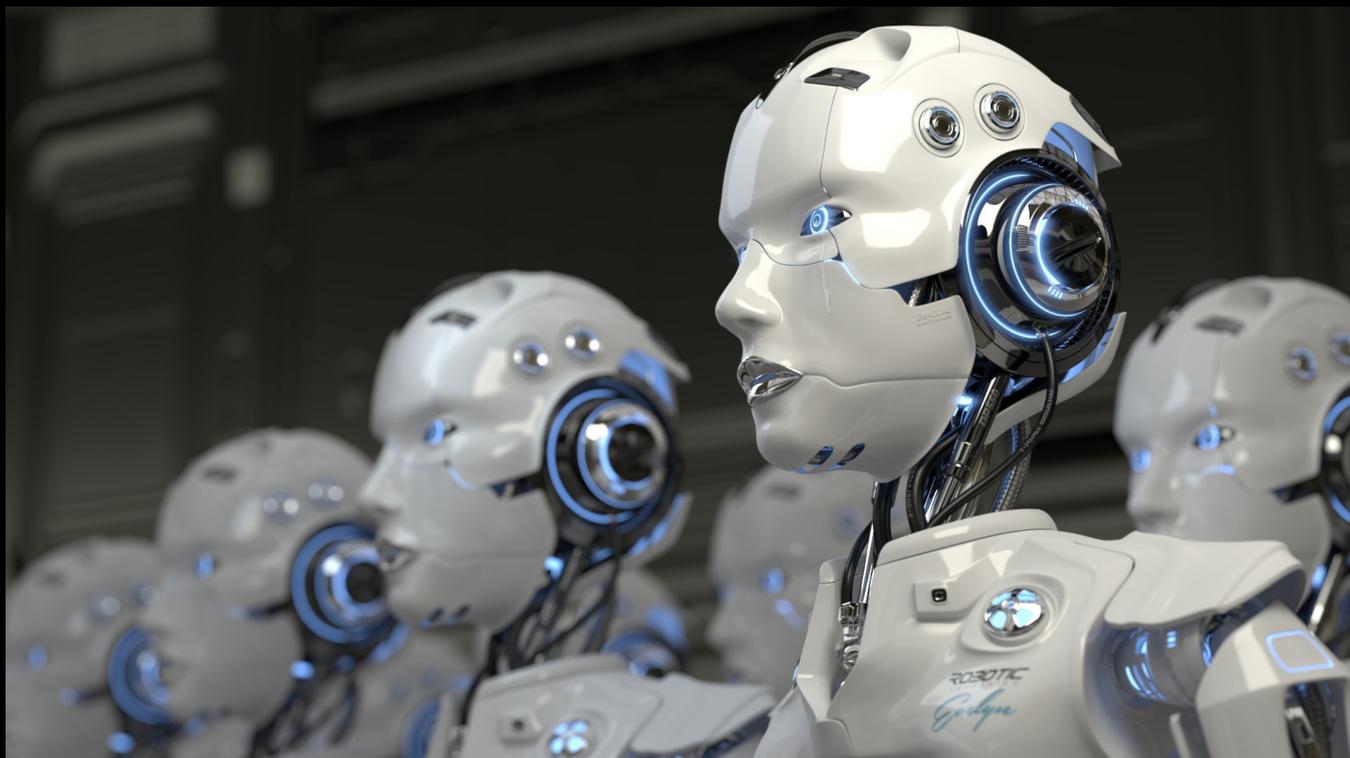


Immagine Marco Iacopetti



Immagine Valenti Enrico

Numero 23
Anno 2019

Direttore Responsabile
Cristina Ducci

Redazione
Alfonso Annarumma
Andrea Campagnol
Alessandro Passariello
Enrico Lui
Riccardo Giovanetti
Carin Belmonte

Grafica e impaginazione
Alessandro Passariello

Copertina impaginazione
Andrea Campagnol

Siti
blender.it
magazine.blender.it

Contatti
magazine@blender.it

Software utilizzati
Blender
The Gimp
Krita
LibreOffice



Il Blender Magazine Italia è mantenuto e gestito dall'Associazione Blender Italia perchè possa essere sempre disponibile per tutti gli utenti.

Associazione Blender Italia

Hanno collaborato a questo numero di **BMI**:

Nicolas D'Amore
Irriverender Bonni
Francesco di Buono
Alessia Lorenzi
Isac Jack Amisano

Vuoi collaborare con **BMI**?

Puoi contribuire scrivendo articoli e tutorials.

Articoli - news relative a Blender come le nuove features, gli aggiornamenti, i nuovi progetti; i "making of" dei tuoi lavori più belli; l'applicazione di Blender in vari ambiti lavorativi.

Interviste - a persone che grazie a Blender sono famose o che hanno fatto di Blender lo strumento del loro lavoro o della loro arte.

Tutorials - sulla modellazione, la creazione, il texturing, l'illuminazione, l'animazione, il rigging, ecc...

Scrivi un documento di testo senza immagini ma con i riferimenti per inserire l'immagine corretta nella posizione esatta
Crea una cartella con le immagini e nominale come scritto nel testo
Crea un'immagine per l'header dell'articolo
Contatta **magazine@blender.it**

grazie a tutti

Il presente numero del Blender Magazine Italia è pubblicato in rete in proprio dalla Associazione Blender Italia sul sottodominio del sito che ospita l'associazione www.magazine.blender.it. Non costituisce testata, non ha carattere periodico ed è aggiornato secondo la disponibilità e la reperibilità dei materiali. La responsabilità di quanto pubblicato è esclusivamente dei singoli Autori. L'Associazione Blender Italia, fondata con atto costitutivo del 10 maggio 2017, ha l'attuale sede sociale in Cosenza (CS), via S. Allende, 2. Il Presidente della associazione è Alessandro Passariello, il Direttore del Magazine è Cristina Ducci. Tutti i riferimenti su www.associazione.blender.it, www.blender.it

www.magazine.blender.it



Immagine di D'Onofrio Manuel