

BLENDER libertà di creare



tesi di laurea di **Riccardo Carozzi**

relatore Fabrizio Valpreda

Laurea in Design e Comunicazione Visiva A.a. 2021/2022 Sessione di Laurea di Dicembre 2022

Blender: la libertà di creare

Creazione di un filmato interamente prodotto utilizzando strumenti gratuiti

Indice

0	Introduzione	9
1 1.1 1.2 1.3 1.4 1.5	Storia e business model della Blender Foundation Cosa è Blender Ton Roosendaal: la sua vision Neo Geo: creatività e tecnica Sostenere un'azienda con un prodotto gratuito Esplosione della "dot-com bubble"e la campagna "Free Blender"	10 10 10 11 12 12
2	Il software libero e l'open source	14
2.1 2.2	Cosa significa software libero? La GNU GPL, Richard Stallman e la Free Software foundation	14 15
3	Casi studio	16
3.1 3.2 3.3 3.4	Filmografia Ricerca Contenuti multimediali Software	16 26 28 30
4	Pre – produzione	34
4.1 4.2	Identificazione del concept: pro e contro di Blender Approfondimenti sul tema: moodboard e brainstorming	34 35
4.3 4.4	Definizione delle linee guida e stesura dello script Scrittura storyboard	37 38
5	Pipeline di produzione	40
5.1 5.2 5.3 5.4 5.5 5.6	Pianificazione del file e introduzione alle collezioni Modellazione environment Le simulazioni fisiche I geometry nodes Progettazioni shader Inserimento luci	40 41 43 45 50 53
6	Pipeline di animazione	54
6.1 6.2 6.3 6.4 6.5	Introduzione all'animazione I keyframe Inserimento camere: animazioni su curve Sviluppo finale shader e illuminazioni Rendering e render engine	54 55 56 59 59
7	Post – produzione	64
7.1 7.2 7.3	Raccolta del materiale Compositing Editing	64 65 68
8	Conclusioni	72
	Bibliografia e sitografia	



Introduzione

L'ambito progettuale è ampio, astratto e complesso. Ogni progettista nel suo percorso che sia di studi, lavorativo o personale, si troverà di fronte a numerose difficoltà e problemi: una di queste è acquisire familiarità con i mezzi software a sua disposizione.

La poca versatilità di numerosi e costosi programmi che cercano di scavarsi la loro nicchia nell'ambito professionale e il continuo flusso di nuovi formati di file proprietari che ostacolano ulteriormente il dialogo tra discipline sono solo alcuni dei molti ostacoli che si possono incontrare. Lo scopo di questa tesi è quello di illustrare come un programma gratuito come Blender possa inserirsi in maniera organica e completa all'interno di un programma complesso come quello della progettazione.

L'elaborato consiste in una discussione dei vantaggi che porta la filosofia del software libero, la storia di Blender e dei visionari che ci sono dietro la sua nascita e di un video completamente ideato, progettato e creato seguendo questa ideologia.

Storia e business model della Blender Foundation

1.1 Cosa è Blender

"Just do nice things for people"

-Ton Roosendaal

3D rilasciato gratuitamente dalla Blender Foundation. Nasce il 2 gennaio del 1994 dalla tastiera di Ton Roosendaal, come un'applicazione del suo software NeoGeo (nome assegnato poco prima del rilascio della celeberrima console omonima giapponese) con lo scopo di renderlo meno frustrante nelle fasi di revisione con l'eventuale commissione, garantendo modifiche più rapide e repentine anche grazie alla sua struttura altamente personalizzabile.

Blender è alla base un software di modellazione

Nel gennaio 1995 lancerà Blender 1.0, già accompagnato dalla sua più grande caratteristica: una singola finestra centrale completamente personalizzabile dall'utente. A ridosso degli anni 2000 lo studio della UX era ancora nelle sue fasi iniziali e i programmi dell'epoca tendevano a seguire il layout di Windows, usando un'organizzazione "a finestre" virtualmente impilate tra di loro, un approccio efficace se inserito in un sistema operativo, ma macchinoso e inefficace quando all'interno di un singolo programma.

Con l'aumento in popolarità e potenza di Blender l'azienda decide nel 2005 di iniziare una tendenza che verrà portata avanti fino ad oggi: dimostrare la potenza e la versatilità del programma creando un cortometraggio in 3D fatto unicamente con Blender. Nasce così "Elephants Dream" il primo di una serie di cortometraggi che segna la nascita del Blender Institute, la casa di produzione interna all'azienda.

1.2Ton Roosendaal: "I was never interested in money, money doesn't la sua vision

mean anything, it's not interesting [...] the real satisfaction is making something". È così che si giustifica Ton Roosendaal, informatico olandese e lead developer di Blender, in una sua celebre intervista con il creator Andrew Price (aka Blender Guru) quando gli viene chiesto come mai ha deciso di rendere il suo software gratuito.

Certamente non una decisione semplice, anche inserendola nel panorama di crisi che stava affrontando il mondo dell'informatica in quegli anni ma che segue la vision che ha reso Roosendaal tanto amato dalla community: pensare prima di tutto al creare qualcosa di interessante, e nel caso servano soldi per realizzarla solo allora pensare a quelli.

1.3 Neo Geo: creatività e tecnica Dopo aver frequentato il corso di laurea di Industrial Design a Eindhoven, Roosendaal decide all'età di 29 anni di fondare la sua prima azienda, con lo scopo di ottenere dei fondi con cui finanziare i suoi studi da Computer Graphics Designer: Neo Geo.

L'azienda procede in poco tempo a diventare una delle più grandi aziende di produzione media dei Paesi Bassi, grazie ad essa Roosendaal acquista nel 1991 per la cifra di 30.000\$ un computer Silicon Graphics, ai tempi l'unico ad avere una scheda video dedicata per il wireframe in tempo reale.

La Silicon Graphics (conosciuta poi come SGI) fu un'azienda nata per produrre quelli che negli anni Novanta erano considerati i computer più potenti che un' industria potesse comprare, sia per software design che per aziende specializzate in grafiche 3D. Calcolatori del genere hanno partecipato nello sviluppo di pietre miliari di numerosi media, tra cui la celeberrima console Nintendo 64 (33 milioni di copie vendute, aveva all'interno una GPU Silicon Graphics RPC) e produzioni Hollywoodiane come Jurassic Park e Toy Story (primo lungometraggio di animazione realizzato completamente in CGI).

Con la compagnia al suo picco e clienti multinazionali del carico di Philips, Roosendaal ritiene che Blender, diffusosi anche grazie alla compatibilità con Linux, FreeBSD e Windows (le tre alternative al sistema operativo nativo SGI: IRIS), abbia le potenzialità per essere un prodotto autonomo. Decide quindi nel 1998 (3 anni dopo il rilascio di Blender 1.0) di vendere la società e fondare Not a Number (NaN).

1.4 Sostenere un'azienda con un prodotto gratuito	Blender non è sempre stato un prodotto interamente gratuito, durante il primo anno di NaN il software utilizzava quella che viene definita una strategia freemium: offrire cioè il programma base in maniera gratuita ma accompagnato da due chiavi, nel caso di Blender denominate A e B (i cui costi erano rispettivamente 50\$ e 100\$) che permettevano di sbloccare più funzioni.
	Con la diffusione del software e i guadagni ottenuti dal precedente business model Roosendaal presenta Blender alla SIGGRAPH (Special Interest Group on GRAPHics and Interactive Techniques) una delle maggiori conferenze internazionali esclusivamente dedicata alla computer grafica, dove ottiene due round di finanziamenti per un totale di 5.5 milioni di dollari.
1.5 Esplosione della dot-com bubble e la campagna "Free Blender"	Per poter giustificare un investimento così grande per quella che era una singola azienda di produzione media dei Paesi Bassi, oltretutto in un periodo storico (quello che va dagli anni 90 all'anno 2000) in cui l'interesse per il 3D a livello commerciale era quasi nullo bisogna considerare l'investimento in una prospettiva temporale.
	Se si guarda infatti il periodo storico l'economia mondiale si trovava in quella che venne successivamente definita la "dot-com bubble": una bolla speculativa dovuta all'eccessiva quantità di investimenti nel settore tecnologico e informatico, causati in gran parte dall'entusiasmo portato dall'incredibile successo avuto da aziende tecnologiche diventate in quegli anni leader mondiali (ad esempio Amazon, le cui azioni ai tempi raggiunsero il picco di 107\$ dollari ad azione per poi passare a 7\$).
	In questo periodo "d'oro" della società Blender si espande a livello internazionale, raggruppa quasi 50 dipendenti da tutto il mondo e, per stessa ammissione di Roosendaal, brucia tutti gli investimenti in conferenze e riunioni per ottenere

investimenti in conferenze e riunioni per ottenere altri investimenti, senza però riuscire a dare una base solida all'azienda su cui mantenersi e da cui avere dei profitti.

Quando nel marzo 2000 esplode la bolla speculativa il gruppo di investitori che aveva finanziato

Blender abbandona il progetto e qualsiasi operazione dentro Nan.

Roosendaal, che oramai possedeva solo il 10% di Blender, decide quindi di sfruttare la community ormai estesa per trovare il capitale da usare per riacquistare i diritti. Nel maggio del 2002 decide quindi di fondare una organizzazione non-profit: la Blender Foundation.

L'estate dello stesso anno marca l'inizio della campagna Free Blender, uno dei primi eventi di crowd-funding della storia, e la raccolta nell'arco di sette settimane di 100.000 dollari. Con questa somma Roosendaal convince gli investitori a rilasciare Blender sotto la licenza GNU GPL (vedere capitolo 2.1.1) dando inizio al nuovo approccio libero di Blender.

"Sometimes you need it to do something big, so you have to work, make money, and then you can do something with it. But the real satisfaction is in making something"

-Ton Roosendaal

2

Il software libero e l'open source

2.1 Cosa significa software libero?

L'esigenza del software libero è stata una diretta conseguenza dello sviluppo tecnologico mondiale: con la diminuzione dei prezzi dell'hardware e l'aumento di complessità dei software iniziata a fine anni 70, quando oramai l'uso dei computer era diffuso al di fuori dagli ambienti accademici, in particolar modo in quelli industriali, il software ha iniziato a guadagnare un forte valore economico.

Questo mancato guadagno, sempre meno assorbito dalle vendite dell'hardware, ha portato le aziende a spostare i costi sui prodotti software, prima infatti era comune offrire gratuitamente il programma con anche il codice sorgente, l'acquirente avrebbe poi deciso da sé se modificarlo e aggiustarlo alle proprie esigenze.

Questa tendenza vede massima espressione nei giorni nostri, dove molti software, in particolar modo quelli altamente complessi e specializzati come quelli per il cinema e la modellazione tridimensionale possono raggiungere prezzi estremamente proibitivi.

Una voce fuori dal coro è la filosofia del software libero, ogni programma o prodotto sotto la licenza GNU GPL deve infatti rispettare quelle che vengono chiamate "le quattro libertà" nella comunità dei cosiddetti "white hat" (nome dato all'insieme di persone che usa le proprie profonde competenze informatiche nell'ambito della sicurezza informatica per prevenire attacchi dei pirati informatici):

"-la libertà di usare il software per qualunque scopo,

-la libertà di modificare il software per adattarlo ai propri bisogni,

-la libertà di condividere il software con gli amici e i vicini,

-la libertà di condividere le modifiche ffettuate."

2.2 La GNU GPL, Richard Stallman e la Free Software Foundation

La GNU GPL (Gnu is Not Unix General Public License) è la licenza principale che viene data ai software liberi, nasce nel 1989 dal desiderio di Richard Stallman di fornire un'alternativa gratuita al sistema operativo Unix brevettato dalla AT&T Corporation (il più grande provider telefonico americano). La particolarità di questa licenza è che obbliga chiunque modifichi o ridistribuisca in qualche modo il programma che la utilizza ad adottarla, quindi distribuirlo in maniera gratuita creando una catena di software gratuiti.

La Free Software Foundation è l'organizzazione non a scopo di lucro fondata da Stallman nel 1895 e si occupa della tutela della licenza GNU GPL, impegnandosi anche a livello legale nella causa tra SCO e IBM, difendendo la distribuzione del famosissimo sistema operativo Linux, anch'esso sotto GNU GPL.

"Free software is software that respects your freedom and the social solidarity of your community. So it's free as in freedom." -Richard Stallman 3



Elephants Dream



Blender Studio



Primo film dello studio della Blender Foundation, il tema di questo cortometraggio è il confine tra immaginazione e realtà.

I due protagonisti esplorano infatti un mondo immaginario e caotico, con un emozionante conclusione che porta a riflettere sul significato del corto.

Primo media in Europa ad essere distribuito in HD DVD

3.1 Filmografia



Figura 3.1 - Elephants dream -2005



Figura 3.2 - Elephants dream frame -2005



Figura 3.3 – Big Buck Bunny –2008

Big Buck Bunny



Blender Studio

2008

Corto nato con lo scopo di dimostrare le capacità di simulare il pelo in Blender.

La trama si evolve attorno a un grosso coniglio disturbato nel suo rituale mattutino da un terzetto di animaletti.



Figura 3.4 – Big Buck Bunny frame –2008





Blender Studio



Primo film dello studio che gode di un budget consistente, il quale permette di creare un video lungo il doppio dei precedenti e con un miglioramento generale nella progettazione dello stesso, in particolar modo nello storytelling.

Sintel è una ragazza guerriera che affronta un viaggio per ritrovare un vecchio amico perduto, con un triste svolgimento finale accompagnato da un sottile messaggio di speranza.





Figura 3.6 - Sintel frame- 2010



Figura 3.7 -Tears of steel - 2012

Tears of steel



Blender Studio

2012

Progetto nato con lo scopo di dimostrare la potenza del programma anche nell'utilizzo di VFX compositing (green screen), reso possibile da un finanziamento del Netherlands Film Fund e da una settimana di registrazione con sistemi e attori professionisti. La trama segue le vicende di un mondo probabilmente post apocalittico, circondato da una lieve vena umoristica che addolcisce un mondo in lotta contro i robot.



Figura 3.8 – Tears of Steel frame – 20012



Cosmos Laundromat



Blender Studio



Pilot di un serie dallo stampo fortememente metaforico che si concentra su Frank: una pecora sola e triste a cui viene data la possibilità di vivere vite nuove al di fuori della sua piccola isola.



Figura 3.9 - Cosmos Laundromat - 2015



Figura 3.10 - Cosmos Laundromat frame- 2015



Figura 3.11 - Agent 327 - 2017

Agent 327 operation barbershop

Blender Studio



Altro trailer per un film attualmente in realizzazione nello studio di Blender.

Nel trailer si può vedere l'agente 327 seguire una pista che lo porta a un barbiere di Amsterdam



Figura 3.12 - Agent 327 frame- 2017

Hero Hero

Blender Studio

2018

Progetto nato per illustrare la versatilità e la potenza del grease pencil, il nuovo strumento creato dalla Blender foundation per poter unire l'arte digitale al 3D in maniera fluida.



Figura 3.13 - Hero - 2018



Figura 3.14 – Hero frame – 2018

Spring



Blender Studio



Cortometraggio al momento di maggior successo del Blender studio (8.3 Mln di visualizzazione su youtube, seguito da agente 327 con 7.8 Mln).

Il cortometraggio è interamente gestito con la nuova versione di blender 2.8, una delle versioni ad aver portato i maggiori cambiamenti nei 20 anni di storia del software.



Figura 3.15 - Spring - 2019



Figura 3.16 – Spring frame – 2019



Figura 3.17 – Spoon – 2022

Corto vincitore del premio "Best Director" al Nikon Best Film Festival 2022, racconta le difficoltà di una madre alle prese con una famiglia impegnativa e il suo desiderio di fuggire dalla realtà per trovare un po' di pace.

Spoon

2022

Arthur Chays

O



Figura 3.18 - Spoon frame - 2022





Figura 3.19 - Dynamo Dream - 2021

Cortometraggio diretto da una delle figure più importanti del panorama di Blender: Ian Hubert. Diventato famoso per il suo approccio creativo al contempo minuzioso ed efficiente e il suo World Building estremamente creativo.

Dynamo Dream è un progetto lungo 4 anni (ancora in corso) che dipinge un mondo fantascientifico sull` orlo del diventare una distopia.



Figura 3.20 - Dynamo dream frame - 2021

3.2 Ricerca



3D Modelling and Visualisation of Heterogeneous Cell Membranes in Blender



Ghaffar, Mehmood, Biere, Niklas, Jäger, Daniel, Klein, Karsten, Schreiber, Falk, Kruse, Olaf, Sommer, Björn



Ricerca sulla virtualizzazione tridimensionale attraverso l'utilizzo di Blender di una serie di cellule dell'alga Chlamydomonas reinhardtii, già soggetto di studi per il loro potenziale utilizzo come forma di produzione di energia.



The comparison Between 3D studio Max and Blender Based on Software Qualities



Hendriyani, Yeka, Amrizal, Vania Amanda



Comparazione tra Blender e il software al momento industry standard all'interno dell'industria dell'animazione: 3Ds Max. Lo scopo della tesi è quello di dimostrare che i vantaggi nell'utilizzo di Blender non sono esclusivamente economici e basati sulla natura open del programma, ma anche sulle sue caratteristiche tecniche.



Design of Virtual Reality Application for Taharah Using 3D Blender



D. R. Anamisa, M. Yusuf, F. A. Mufarroha and N. Rohmah



Ricerca sull'utilizzo di Blender nel creare un' applicazione di supporto alla pratica religiosa di puricazione detta Taharah.



Creating 3D Game Art for the iPhone with Unity





Analisi e guida sul come Blender possa inserirsi nella pipeline di lavorazione Unity creando rigging e animazioni dei personaggi di un'applicazione per Iphone.



O Determinants of Blender 3D (open source) usage in Malaysian animation industry



Mohamed Afique Bin Mohamed Azi-yen, Elsadig Musa Ahmed



Analisi della diffusione dell'animazione come economia all'interno della Malesia, in particolar modo sulla diffusione di Blender all'interno della stessa, analizzando quali siano i fattori che possano determinare una maggiore diffusione del software.



Immersive 3D modeling with Blender and off-the-shelf hardware



Takala, Tuukka M., Makarainen, Meeri, Hamalainen, Perttu



Progettazione e messa in uso di un sistema di modellazione 3D basato su Blender e sull'utilizzo di elettronica facilmente reperibile (controller PlayStation Move).

Il sistema ottenuto consente di creare bozze 3D con sufficiente accuratezza e in maniera molto intuitiva grazie all'utilizzo di un display 3D.

3.3 Contenuti multimediali



Riproduzione dell'iconica scena di Jurassic Park interamente realizzata in Blender.



Figura 3.21 - Recreating Jurassic Park in Blender - 2021



Figura 3.22 - Blender 2.9 beginner tutorial - 2021



Primo video della serie di tutorial di Blender più conosciuta, tale da rendere la ciambella il simbolo del primo approccio al programma.



Creature creation - Modeling, Sculpting, Texturing, Rigging





) 3 milioni

Timelapse della creazione di una creatura fantasy attraverso la tecnica dello sculpting: uno degli approcci permessi dal programma per poter creare forme organiche in maniera "tradizionale" agendo su un blocco di creta virtuale.



Figura 3.23 Creature creation - 2015



Figura 3.24 - Product animation in Blender - 2019



Serie di tutorial dedicata alla modellazione, texturing e animazione di una presentazione di product design.

3.4 Contenuti multimediali







Figura 3.25 - VLC Media Player - 2001



Uno dei più celebri lettori multimediali gratuiti esistenti al mondo. Sviluppato inizialmente come streaming server per la comunità scolastica di una scuola francese 5 anni dopo la sua nascita viene completamente riprogrammato e distribuito sotto la licenza GPL



Figura 3.26 – Telegram 2013





Nikolaj e Pavel Durov



Secondo client di messaggistica al mondo dopo whatsapp, nasce con l'obbiettivo di essere un'alternativa al software di proprietà di Zuckerberg portando maggiore attenzione alla privacy e l'integrazione di bot e canali estremamente versatili.



The Krita Team



Programma di elaborazione di immagini che si pone come alternativa a GIMP (altro software sotto licenza GPL) e Adobe Photoshop.

Nonostante sia un programma meno maturo e con meno budget dei suoi due competitor la sua interfaccia più moderna lo rende lo strumento preferito di molti artisti.





Figura 3.28 - Libre Office - 2011



The Document Foundation

2011

Alternativa free al celeberrimo software di produttività di casa Microsoft, la suite è composta da diversi programmi che permettono di creare presentazioni, grafiche, fogli di calcolo e stesure di testi.





Spencer Kimball e Peter Mattis





Figura 3.29 – Gimp – 1996

Altro programma di disegno digitale nato come progetto di tesi per un'università della California. Una volta concluso il percorso universitario i due creatori del programma entrano nel progetto GNU (GIMP sta per GNU Image Manipulation Program).









Matt Mullenweg e Mike Little



Fork del CMS (content management system) b2evolution che negli anni è diventato il più utilizzato al mondo grazie alla sua semplicità di utilizzo e alla possibilità di creare un proprio sito web in maniera gratuita e autonoma.

Pre-produzione

4.1 Identificazione del concept: pro e contro di Blender

• gratuito

- tutti i tool seguono lo stesso sistema di navigazione
- interfaccia pulita e moderna
- · limiti compensati dai plug-in
- enormi risorse didattiche
- aggiornato e supportato
- versioni a Long Term Support (LTS)
- node editor/workflow procedurale
- motore fisico integrato e semplice da usare
- render più veloci
- possibilità di rimanere in un solo software per tutto il processo creativo
- ripida curva di apprendimento
- rigging organico più difficoltoso
- mancanza di sculpting layer
- texturing non in pari coi competitors
- editing video e compositor limitati
- mancanza di figure estremamente specializzate
- mancanza di un workflow professionale stabilito
- non integrato nell'industria

Dopo aver analizzato lo scenario in cui ci troviamo è necessario stabilire un concept efficace al nostro scopo: come abbiamo già detto l'obbiettivo di questa tesi è "quello di illustrare come un programma gratuito come Blender possa inserirsi in maniera organica e completa all'interno di un programma complesso come quello della progettazione" [introduzione; pag. 9].

Per fare ciò però è necessario analizzare quali sono i pro e i contro di questo programma, sono stati quindi fatte delle interviste nelle varie piattaforme social (www.facebook.com ; https:// www.blender.org/community/; https://www. blender.org/get-involved/developers/) a vari user e pro-user di Blender, i dati sono quindi stati organizzati nella tabella a fianco .

Possiamo quindi avere un'idea di quali siano i punti di forza da valorizzare e quali quelli deboli da compensare.

Il modo più efficiente ed utilizzato sia da aziende che da liberi professionisti per dimostrare le capacità di un programma (o nell'utilizzare lo stesso) è il formato dello "showreel": un montaggio ad alto potenziale visivo che mostri, come un portfolio, il livello di rappresentazione che si può raggiungere con il programma che si vuole utilizzare. Lo si può definire come il "trailer cinematografico" di un prodotto.

Il concept del video nasce dal voler creare un montaggio che invogli lo spettatore a provare Blender: il target deve quindi essere il più ampio possibile, dal libero professionista che ha già esperienza nel 3D sia qualcuno che si sta approcciando al mondo del 3D. Il messaggio principale deve essere quello dell'essere un free software che offre completa libertà creativa e dell'essere allo stesso livello dei competitor. Per poter esprimere nella maniera più organizzata il messaggio che vogliamo trasmettere è necessario fare delle ulteriori analisi utilizzando due strumenti caratteristici delle fasi di progettazione: il moodboard e il brainstorming.

4.2 Approfondimento sul tema e ricerca del linguaggio comunicativo Questi due strumenti sono fondamentali per poter trascrivere e riconoscere dei pattern all'interno dei valori che vogliamo trasmettere, è necessario quindi analizzare in maniera più approfondita il tema raccogliendo immagini (per la moodboard) [vedi figura 4.2] e concetti (per il brainstorming) [vedi figura 4.1] che rimandino in maniera istintiva al concept.



Figura 4.1 – Brainstorming



Figura 4.2 – Moodboard
4.3 Definizione e linee guida dello script Ora che abbiamo analizzato lo scenario e trovato un tema per il concept è necessario stabilire delle linee guida in base ai dati che abbiamo raccolto.

Sappiamo che l'obbiettivo è quello di attrarre più persone possibile ad utilizzare Blender, quindi è necessario prendere un target molto ampio, il messaggio principale che vogliamo esaltare è la possibilità di scaricare Blender in maniera totalmente gratuita, altri punti fondamentali sono il mostrare il profondo legame tra la community e la filosofia del software libero ed utilizzare materiale il più possibile variegato per poter dimostrare la versatilità del programma.

Arrivati a questo stadio della progettazione possiamo definire uno script che faccia da base allo storyboard:

La scena si apre in una strada di una città del futuro in notevole stato di degrado, la povertà dilaga e le strade sono tremendamente pericolose, un uomo attraversa una strada tra i palazzi illuminati da luci e insegne e si trova davanti un particolare distributore automatico, se vive in un posto del genere difficilmente potrà comprarsi nulla e il piccolo display a lato recita "please insert credit" a ripetizione, ma tra le varie offerte una risulta disponibile, e gratuita.

Il protagonista quindi si avvicina e seleziona l'opzione gratuita con il simbolo di Blender sopra, il piccolo display a lato risponde con un inizialmente titubante "Are you sure? Ok! Let's go!"

A quel punto la camera si avvicina a un altro display fino a prendere l'intero schermo, inizia quindi il vero e proprio showreel con una musica accattivante, si susseguono spezzoni del materiale raccolto [vedi capitolo 7 per approfondire]

Una volta concluso il montaggio appare all'interno dello schermo nero esclusivamente lo schermo con cui si è dialogato precedentemente, con la scritta "Another shot? No credit needed"

Teniamo a mente che essendo questo uno script bozza è probabile che nel risultato finale vi saranno delle differenze, anche importanti.

4.4 Scrittura dello storyboard

Una volta stabilito lo script da cui partire è necessario creare uno storyboard, in modo da poter organizzare nella maniera più efficace possibile il lavoro di modellazione e rendering.

Per fare ciò ci avvaliamo del software "storyboarder" un programma gratuito e che permette di creare, editare e condividere storyboard in maniera professionale.

Una caratteristica molto utile del programma è la sua interazione con i programmi di modellazione, possiamo quindi modellare una semplice bozza dell'environment su Blender e poi importarla dentro storyboarder.



Schermo nero

il logo di blender entra in camera e rimane fermo al centro dello schermo per mezzo secondo, per poi uscire





camera zooma sul distributore simulando una camminata



"please insert credit and choose your software of choice

distributore si rivolge alla camera



"are you sure? yes - no"

viene selezionato "yes"



"OK! Let's go" "-)"





schermo spento e transizione a showreel

viene selezionato "yes"

SHOWREEL





SHOWREEL

showreel

showreel





showreel

5A

showreel

zoom out schermo si spegne





"Another shot? No credit needed"

":)"

Camera sale verso il display primario

Con la stesura dello storyboard completata abbiamo concluso la fase di preproduzione del filmato. Si può quindi passare alla fase di produzione, partendo dalla modellazione dell'ambiente che andremo ad utilizzare. 5

Pipeline di produzione

5.1 Pianificazione del file e introduzione alle collezioni Per poter modellare nella maniera più efficace possibile è necessario affrontare uno step di organizzazione della scena: sappiamo che il nostro ambiente sarà una strada con una vista frontale su un edificio, la scena sarà molto affollata per dare un senso di sovrappopolazione e povertà quindi questo passaggio è ancora più importante.

La scena quindi si suddividerà in 3 macroambienti:

- edificio principale

- sfondo (composto da edifici secondari e la strada)

 - il distributore automatico (che essendo il focus principale sarà più dettagliato e quindi necessità di essere separato)

L'organizzazione dei file su Blender segue un ragionamento simile alle cartelle di windows: un sistema gerarchico molto flessibile e personalizzabile, ogni oggetto, luce e camera presente nella scena ha un suo nome (nel caso i nomi siano ripetuti sarà presente un suffisso numerato partendo da .001) e più oggetti possono essere raggruppati in cartelle, dette collezioni. Un insieme di collezioni forma una scena.

Ogni collezione può poi essere suddivisa in sotto-collezioni che possono essere spostate da una "cartella" all'altra.

Per quanto riguarda la fase di modellazione le collezioni che ci interessano sono le prime due: "Environment" che richiude al suo interno l'edificio principale e lo sfondo, più i vari oggetti accessori, mentre "Distributore automatico" ha al suo interno il soggetto principale.

Blender inoltre mostra dopo il nome delle collezioni quali tipologie di elementi possiede al suo interno: prendiamo come esempio la

- collezione environment che contiene in ordine -un oggetto empty -più di novantanove mesh -due curve -due testi
- -quattro sottocartelle

5.2 Modellazione environment

Ora che abbiamo stabilito delle macrocategorie da seguire per essere il più ordinati possibile passiamo alla modellazione vera e propria.

La casa al centro dell'inquadratura, per essere in linea con il concept, deve avere forme semplici ma ben definite, deve essere quindi il meno organica possibile per dare l'idea di un ambiente tecnologicamente avanzato, partendo quindi da un cubo ne modifichiamo ed estrudiamo le facce per ottenere le "ossa" di quella che sarà la casa, ottenendo questo risultato [vedi figura 5.1]

Una volta raggiunto un livello di dettaglio soddisfacente aggiungiamo dei dettagli, in questo caso una struttura a lato, delle griglie, delle veneziane, una porta e delle barre davanti alla finestra, in modo da aumentare ancora di più la sensazione di degrado e pericolo.

Tutto ciò viene fatto modificando la struttura di forme base come piani, cubi e cilindiri. [vedi figura 5.2]

Nota: per poter avere un'integrazione più semplice con i materiali e le luci, tutti questi lavori sono fatti in scala usando come unità di misura un modello umano molto semplificato, che è poi stato scartato una volta definita la dimensione corretta.

Ora che l'edificio è completo possiamo passare alla definizione dell'area circostante: composta dalla strada, dagli edifici secondari e dagli impianti di illuminazione, essendo tutti questi elementi fuori dal focus della camera, sia per poter rendere il procedimento più veloce ma anche per non caricare di dettagli inutili la scena, è buona norma usare un livello di dettaglio sempre più basso man mano che ci si allontana dal punto di interesse (con l'aggiunto beneficio di guidare lo sguardo dello spettatore verso parti più importanti). Sempre seguendo il metodo della modifica e dell'estrusione delle forme base raggiungiamo questo modello.



Figura 5.2



Figura 5.1

[vedi figura 5.3]

Definita la scena principale possiamo salire con i livelli di dettaglio ed aggiungere degli elementi di contestualizzazione, come delle ringhiere sul bordo del marciapiede e dei pali della luce. [vedi figura 5.4]

Adesso che abbiamo definito una scena in cui ambientare il corto di introduzione possiamo modellare il distributore automatico, nonostante il processo sia particolarmente lungo a causa dei numerosi dettagli i principi che utilizziamo saranno sempre gli stessi, modificare ed estrudere delle forme base, raggiungiamo quindi questo risultato: [vedi figura 5.5]

Il prossimo passo necessario è l'ultima fase della modellazione: essendo questo un ambiente come detto povero è consono aggiungere alla scena numerosi "props", cioè elementi a basso livello di dettaglio che però aumentino la generale sensazione di immersione all'interno della scena. Modelliamo quindi numerosi oggetti di scena ottenendo la scena completa. [vedi figura 5.6]



Figura 5.5







Figura 5.4



5.3 Le simulazioni fisiche

Una parentesi degna di nota è rappresentata dai vari teli che si possono vedere nella scena, quest'ultimi infatti non sono fatti secondo il metodo che abbiamo visto della modifica di forme base, ma andiamo ad utilizzare quella che è una simulazione fisica. Blender include infatti numerose possibilità simulative, in questo caso noi andremo ad utilizzare quella dei tessuti, progettata per questa particolare esigenza, vediamo quindi una spiegazione un po' più approfondita di come eseguire una simulazione.

Innanzitutto è necessaria una forma da cui partire, in questo caso la più semplice è un piano, quest'ultimo, come ogni rettangolo, è composto da quattro vertici, quattro lati e una faccia (per il momento è consigliabile lavorare in 2 dimensioni, useremo poi uno strumento per dare spessore al telo), ma dato che la simulazione calcola le forze sui vertici di una forma è necessario aumentarne il numero per avere una simulazione il più possibile accurata, usiamo quindi la funzione "subdivide" che permette di dividere il piano in quattro facce più piccole, usandolo più volte dividiamo queste quattro facce in altre quattro e così via finché non otteniamo un numero di vertici consono (tenendo a mente però che più vertici otteniamo più pesante sarà la simulazione). [vedi figura 5.7]

Una volta preparato il "telo" dobbiamo preparare



Figura 5.7



Figura 5.8







Figura 5.10

gli elementi con cui questo telo interagisce: il processo è molto semplice, prendiamo tutti gli elementi che andranno a toccare il telo guando lo faremo cadere e attiviamo la funzione "collision" in modo che Blender lo consideri come un oggetto solido e permetta al telo di interagirci.

Abbiamo quindi preparato tutto ciò che serve, possiamo far partire la simulazione con i valori di default (essendo questa una simulazione molto semplice, non è necessario modificarli, quando si raggiunge volumi di dettagli molto più importanti sarà necessario ottimizzarli per rendere la simulazione credibile) e una volta conclusa scegliere il frame che riteniamo più adatto. [vedi figura 5.8]

Come già anticipato però il telo al momento è bidimensionale e molto "frammentato" infatti non è possibile deformare delle facce e dobbiamo aggirare questo limite (nota: definirlo limite non è propriamente corretto, essendo la base di come il programma gestisce le forme, è una caratteristica simile a tutti i software di modellazione mesh come Blender) utilizzando due modificatori.

Con modificatori si intende dei procedimenti automatizzati che possono essere "assegnati" agli oggetti e possono interagire sia tra di essi che con l'oggetto in questione, per quanto riguarda la frammentazione usiamo un modificatore con il nome di "subdivision surface" che permette di aumentare considerevolmente il numero di vertici in maniera intelligente ottenendo una superfice più morbida [vedi figura 5.9], altro vantaggio di usare questo approccio rispetto a fare una modellazione come quelle che abbiamo usato precedentemente è la non distruttibilità del processo, cambiando infatti un singolo valore possiamo ad esempio decidere la quantità di suddivisioni, senza modificarlo permanentemente, basta infatti eliminare il modificatore e ci troveremo davanti il telo originale non modificato.

Per poter dare spessore al telo useremo il modificatore "solidify" che serve appunto a dare massa a oggetti bidimensionali. [vedi figura 5.10] Questo procedimento viene quindi riutilizzato sui vari teli della nostra scena.

5.4 I geometry nodes L'ultimo (e più recentemente implementato) metodo di modellazione che andiamo ad utilizzare è quello dei Geometry Nodes. Questo metodo esclusivo di Blender è stato introdotto al pubblico in maniera stabile il 25 febbraio 2021 ed è quindi ancora poco utilizzato, ma possiede un potenziale inimmaginabile.

> I geometry nodes sono un metodo di modellazione procedurale, prevede quindi l'utilizzo di formule matematiche e non della modifica di mesh per creare un oggetto, portando quindi a un metodo completamente diverso e molto complesso. In questo progetto lo utilizzeremo per creare delle insegne a led interattive, che possono quindi mostrare dei testi in movimento ed essere facilmente riutilizzabili e modificabili.

> Dato che stiamo approcciando un metodo nuovo e molto complesso, può essere difficile interfacciarcisi le prime volte proprio perché segue dei principi completamente diversi dalla modellazione tradizionale. Ed è qui che uno dei grandi vantaggi della community così attiva di Blender ci viene in aiuto: possiamo infatti semplicemente cercare delle guide online (su youtube o anche su molti altri siti) ed avere un infarinatura dell'argomento e, in alcuni casi, dei file già pronti e privi di qualsiasi licenza da poter usare anche commercialmente.

> In questo caso andremo a studiare questo video in particolare (https://www.youtube.com/ watch?v=_TYD-CRDGXI) e, una volta compreso le meccaniche di base dei geometry nodes andremo a ricreare l'oggetto in questione riadattandolo alle nostre esigenze.

Una volta creato un oggetto e assegnatogli un gruppo di geometry nodes ci troviamo davanti a questa schermata: [vedi figura 5.11]

Il primo passo è quello di trovare una geometria di base da cui partire, nel nostro caso penseremo ad una insegna led come ad una griglia di pixel, quindi partiamo da una griglia, aggiungiamo quindi un nodo "grid" (che possiamo trovare sotto il pannello mesh primitive).

Aggiungiamo poi un nodo "transform" che permette di applicare delle modifiche alla posizione della geometria nello spazio, in modo da poterlo ruotare di 90 gradi (e quindi appenderlo virtualmente ad un eventuale muro). Raggiungiamo



Figura 5.11



Figura 5.12



Figura 5.13

quindi questo risultato: [vedi figura 5.12]

Una volta ottenuta la griglia (di cui possiamo modificare dimensioni e densità a piacimento) possiamo passare allo step successivo, e cioè assegnare a ogni vertice della griglia una geometria che ci farà da led, per fare ciò è necessario aggiungere un nodo "instances to point", lo scopo di quest' ultimo è infatti di creare un' istanza di un oggetto su ogni punto della griglia, lo inseriamo quindi tra il nodo trasform e il nodo output e gli indichiamo come oggetto da utilizzare due sfere, una che rappresenta un led acceso e una che rappresenta un led spento (usando dei materiali appropriati, per approfondimenti vedi capitolo successivo).

Nasce qui un problema facilmente risolvibile: dato che abbiamo sostituito la griglia con degli oggetti (il nodo instances usa la griglia come riferimento di posizione, rimuovendo quindi la geometria assegnatale) usiamo un nodo "join geometry" che ci permette di unire sia i led disposti sulla griglia che la griglia stessa.

Una volta che abbiamo raccolto in una collezione a parte le due sfere possiamo usare la collezione come riferimento per il nodo instance on point, attiviamo poi le opzioni "separate children" (consente di usare come output più oggetti), "reset children" (resetta eventuali trasformazioni fatte agli oggetti) e "pick istance" [vedi figura 5.13]. Quest' ultima funzione è particolarmente importante perché ci permette di scegliere se mostrare sulla griglia una lampadina accesa o spenta in base alla posizione di un testo secondo un metodo che ora spiegamo.

Una volta che abbiamo ottenuto la griglia con i led possiamo aggiungere i nodi che ci permetteranno di usare un testo come indicatore di posizione dei led accessi, allo stesso modo in cui funziona un display reale.

Il metodo che utilizziamo può essere immaginato nel seguente modo: diciamo al programma di creare dei raggi che escano in maniera perpendicolare (o normale, vedremo a breve il collegamento) ai pixel del display virtuale, nel caso in cui questo raggio venga interrotto da un oggetto, nel nostro caso un testo, il pixel a cui è assegnato quel raggio in particolare si accenderà, in caso contrario resterà spento. Per fare ciò ci



Figura 5.14





Figura 5.16

avvaliamo del nodo "Raycast" (che in inglese sta a rappresentare questo nostro ragionamento: ray, infatti, significa raggio e cast significa lanciare) che colleghiamo al nodo instance, in questo modo i valori che questo nodo porterà saranno usati dalla funzione pick instance che abbiamo visto poco fa. Il nodo di base però non è sufficiente, dobbiamo infatti dirgli quale geometria debba considerare come "solida" e che quindi non venga attraversata dai raggi e in che direzione proiettare quest' ultimi, poiché la direzione di default non è quella adatta al nostro caso, portiamo quindi all' interno del pannello un oggetto testuale che abbiamo creato appositamente e lo colleghiamo alla funzione "target geometry", per indicare la direzione usiamo il nodo "normal" per indicare appunto la direzione normale/perpendicolare. [vedi figura 5.14]

L' oggetto base è quindi concluso e questo è il risultato: [vedi figura 5.15]

Possiamo però aggiungere alcuni miglioramenti all' oggetto, in particolar modo poter modificare la posizione del testo con un solo slider e rendere più accessibili le opzioni di modifica.

Per fare ciò inseriamo tra il nodo object info e il nodo raycast un nodo transorm, in modo fa poter spostare i raggi virtuali di cui abbiamo parlato prima a nostro piacimento, usando quindi lo slider della traslazione orizzontale possiamo muovere il testo ed animarlo grazie ai keyframe [vedi capitolo 6.2]; ora il nostro oggetto avanzato è concluso. [vedi figura 5.16]

Manca però una questione che nel nostro caso può essere ignorata (cosa che non faremo), ma una volta che il lavoro di modellazione diventa professionale o addirittura scala a produzioni cinematografiche diventa fondamentale: la facilità d' uso.

Lo studio della progettazione ci insegna infatti come prima cosa il significato della parola stessa: proiectare, gettare (rielaborabile come guardare) avanti, e quindi pensare in anticipo alle problematiche future, questo approccio vale anche nella creazione di contenuti video. Quando le scene diventano molto complesse avere un accesso veloce e semplificato a dei parametri diventa di fondamentale importanza,



Figura 5.17

al pari della qualità del lavoro che si porta. Risparmiare tempo nelle produzioni media, così in come molti altri lavori, equivale a un risparmio di denaro. Dettagli anche piccoli si ammassano molto velocemente e rischiano di creare una situazione ingestibile, se poi consideriamo che spessissimo una produzione di alto livello come quella di uno studio richiede l'interazione di diverse figure professionali che devono lavorare con lo stesso file si moltiplica ulteriormente.

È quindi di fondamentale importanza tenere sempre a mente questa ideologia, sia nella modellazione (ad esempio nominando in maniera consona i vari oggetti, creare codici colore per le collezioni, organizzare il lavoro dal principio) che nello shading (organizzando in maniera ordinata le texture) che in qualsiasi altro step della pipeline di produzione.

Spiegato quindi il motivo per cui anche dettagli così piccoli sono fondamentali andiamo a vedere come migliorare in pochissimo tempo il nostro pannello led. Tenendo a mente che questo è lo stato attuale del nostro oggetto. [vedi figura 5.17] Considerando quanto sia ritenuto basico ciò che abbiamo fatto non è difficile immaginare quanto velocemente una visualizzazione a nodi di questo tipo (per quanto esistano modi di organizzarla) possa diventare troppo confusionaria. Ci viene però in aiuto la versatilità di Blender. Il programma, infatti, come abbiamo detto nei primi capitoli, ha come filosofia di base la perfetta interazione di tutte le sue parti. Basta quindi prendere un nodo denominato "group input", che permette di estrapolare le variabili più importanti, mostrarle nel pannello delle proprietà e collegarci gli slider o nodi che riteniamo più importanti. Questo procedimento ci permette quindi di modificare forma, definizione e posizione del testo da un pannello molto accessibile e più semplificato. Decidiamo quindi di collegare i valori size x e size y (responsabili delle dimensioni del pannello), verticles x e verticles y (responsabili della densità dei led) e la funzione transform del testo. Quest' ultimo per poter funzionare correttamente però necessita di un nodo in più: un "combine xyz". Se noi collegassimo transform direttamente al group input lo slider che ne consegue modificherebbe la posizone in tutti e tre gli assi contemporaneamente, il combine xyz permette di separare i tre assi e quindi collegare solo l'asse x (che si occupa dello scorrimento orizzontale) al



8	<u>م</u>	~
ł۲	🔳 LED Board > 🍋 GeometryNodes	Ń
ĉ	Add Modifier	~
8	🗸 🍋 GeometryNodes 🚏 🛄 🙍 🗸 🗙	
	🕄 🗸 Led Panel 💟 🖸	X
6	Size X 1.08 m	•
S	Size Y 0.1 m	•
-	Vertices X 131	•
	Vertices Y 18	•
2	X -0.004	•
12	> Output Attributes	
۲	> Internal Dependencies	
0		
\$		
•		
88		
	Figura 5.19	

Figura 5.18

group input.

Ora il pannello a led è concluso in tutte le sue parti, a dimostrazione di ciò che abbiamo detto poco fa questa è la visualizzazione a nodi: [vedi figura 5.18]

Mentre questo è il pannello che ci fornisce il group input:

[vedi figura 5.19]

La fase di modellazione è quindi ufficialmente conclusa.

5.5 Progettazione shader

	DCD
	BSDI
GGX	~
Random Walk	Ý
Base Color	
Subsurface	0.000
Subsurface Radius	~
Subsurface Colo	
Subsurface IOR	1.400
Subsurface Anisotropy	0.000
Metallic	0.000
Specular	0.500
Specular Tint	0.000
Roughness	0.500
Anisotropic	0.000
Anisotropic Rotation	0.000
Sheen	0.000
Sheen Tint	0.500
Clearcoat	0.000
Clearcoat Roughness	0.030
IOR	1.450
Transmission	0.000
Transmission Roughness	0.000
Emission	
Emission Strength	1.000
Alpha	1.000
Normal	

Figura 5.20

La creazione di materiali è un procedimento molto complesso, esistono programmi dedicati esclusivamente alla creazione di questi materiali per quanto il discorso sia ampio e discuterne nel dettaglio va oltre lo scopo di questa tesi, possiamo però fare un'introduzione al metodo. Blender permette di utilizzare principalmente due metodologie per creare dei materiali, il PBR e il texturing. PBR sta per "Physically Based Rendering" e cioè un materiale che viene creato seguendo un' imitazione virtuale delle proprietà fisiche del materiale che si vuole rappresentare, il texturing invece prevede di usare delle immagini da usare sopra un determinato oggetto. Questi due metodi, grazie all'utilizzo dei nodi (un sistema di visualizzazione schematico che permette di organizzare le varie proprietà di un materiale) possono coesistere, nel nostro caso in alcuni casi useremo il texturing in altri il PBR, ed in casi particolari entrambi.

Ogni oggetto in Blender può possedere uno o più materiali, e ogni materiale può essere costituito da un numero potenzialmente infinito di nodi, ma il più diffuso è il Principled BSDF: questo nodo, infatti, racchiude tutte le principali caratteristiche fisiche di un materiale, ad esso poi potremo collegare altri nodi in caso ci serva una maggiore personalizzazione.

ll nodo di default appare così: [vedi figura 5.20]

Ecco una legenda delle funzioni più importanti del nodo:

Base color: colore della superfice dell'oggetto
Metallic: questo slider permette di decidere se il materiale è o no metallico, essendo l'interazione con la luce diversa tra metalli e non metalli (nota bene: i metalli non rifrangono la luce ma la riflettono esclusivamente, perché tutta la luce assorbita dal metallo viene dispersa in calore)
Roughness: decide quanto un materiale sia o no riflettente, un valore più alto porterà a un oggetto più ruvido e quindi meno riflettente
Emission: opzione da utilizzare solo se il materiale che vogliamo usare emette luce [vedi capitolo

successivo] -Normal: nodo che definisce la direzione dei vari pixel del materiale, per dare quindi un'immagine

più realistica di un oggetto che nella realtà non è mai perfettamente piatto; per poterlo utilizzare è necessario mapparlo non essendo un valore singolo ma variabile.

51

Da questo punto possiamo decidere l'approccio da utilizzare in base al materiale, se è molto semplice e non è in primo piano possiamo semplicemente modificare valori come il base color, il metallic e la roughness per dare un'idea di solidità all'oggetto senza appesantire la scena, rimanendo quindi all'interno dell'approccio PBR. Essendo infatti un materiale procedurale, e cioè definito da regole matematiche, in maniera analoga alla grafica vettoriale può essere adattato a qualsiasi dimensione e definizione senza perdita di qualità o necessità di maggiore potenza di calcolo.

Questo approccio basato sulla fisica è però estremamente complesso quindi lo utilizzeremo solo per le luci [vedi capitolo successivo] e qualche materiale in cui sarà necessario avere più controllo possibile.

Per quanto riguarda i materiali principali useremo delle texture, in questo caso la community di Blender ci viene nuovamente in aiuto: il lavoro di creazione di texture è infatti estremamente complesso, lungo e costoso, impensabile per un libero professionista e anche per studi minori. Ma proprio grazie alla enorme diffusione di questo programma e la filosofia del free software, esistono numerosi e amplissimi cataloghi online completamente gratuiti e privi di qualsiasi restringimento di diritto d'autore a cui possiamo accedere da un qualsiasi browser, basterà quindi scaricare le varie texture (o direttamente il materiale già applicato ad un oggetto, trasferendolo poi ai nostri oggetti di scena) ed applicarli ai nostri oggetti. Vediamo quindi il procedimento.

Una volta scaricato il materiale ci troviamo davanti queste immagini:

[vedi figura 5.21]



Paper005_4K_Col

or.png







Paper005_4K_Rou ghness.png



W.jpg

Paper005_4K_Nor malGL.png placement.png

Abbiamo quindi:

-una pre-visualizzazione di come sarà il materiale una volta applicato -una texture che definisca il colore (che colleghiamo al base color) -una texture che definisce la roughness -una texture che definisce le normali (unica che non verrà collegata direttamente al modulo Principled, serve infatti un tramite detto normal map per convertire l'immagine in valori) -una texture che viene detta displacement, questa texture serve per dare geometria aggiuntiva all'oggetto a cui applichiamo il materiale, ma noi non la adopereremo poiché come funzione non è compatibile con il motore grafico che usiamo. Importiamo quindi le varie immagini con un semplice drag and drop e facciamo i collegamenti che abbiamo precedentemente discusso ottenendo questo risultato. [vedi figura 5.22]

(nota bene: i due nodi a sinistra servono a mappare le immagini, specificando dove e come distribuirle sul mesh)

Applichiamo poi lo stesso principio a tutti i materiali ottenendo questo risultato: [vedi figura 5.23]





Figura 5.22

5.6 Inserimento luci



Figura 5.24

Il discorso delle luci nel nostro caso è molto semplice, anche se richiede una certa attenzione nelle fasi più avanzate [vedi capitolo 6.4], la nostra scena essendo localizzata in una città di notte prevede un illuminazione strettamente artificiale, più semplice da emulare che una naturale, ed è garantita da numerosi oggetti cilindrici a cui applichiamo un materiale molto semplice, sostituiamo infatti il nodo principled BSDF e usiamo una nuova versione semplificata fatta appositamente per oggetti che emettono luce, un nodo "emission", da cui basterà prendere il colore e l'intensità che desideriamo.

L'organizzazione del materiale sarà quindi la seguente:

[vedi figura 5.24]

Aggiungiamo inoltre un piano che emette una luce neutra per risaltare meglio i contrasti della scena. Otteniamo quindi questo risultato: [vedi figura 5.25]



6

Pipeline di animazione

6.1 Introduzione all' animazione

"Nothing's occurring in animation – you manufacture everything." – Gore Verbinski La nostra scena è pronta: abbiamo una idea generale di come gestire il prodotto finale e abbiamo tutti gli elementi grafici necessari pronti, entriamo quindi nello step dell'animazione.

Possiamo suddividere questa fase in due principali step: animare la scena e rendering del prodotto.

Il primo step prevede di modificare nel tempo posizione, forma, colore e molti altri fattori degli oggetti presenti nella scena per poter dare "vita" alla stessa: inserire personaggi e creare interazioni tra oggetti sono solo alcune delle possibilità infinite date dal media del 3D, il limite più grande rimane la capacità e la creatività dell'autore.

La fase di rendering è invece responsabile del trasportare da un ambiente tridimensionale (il nostro software di modellazione) ad uno bidimensionale (lo schermo da cui è fruito il prodotto che stiamo realizzando).

Per fare ciò, come in molte altre situazioni, i programmi di modellazione 3D si ispirano alla realtà e prevedono l'utilizzo di camere virtuali, progettate per imitare ogni variabile del corrispettivo reale. È possibile quindi utilizzare e personalizzare delle vere e proprie videocamere, con parametri analoghi a quelli che potremmo trovare in qualsiasi altra camera, con cui "filmare" ciò che desideriamo all'interno del nostro ambiente. È quindi possibile animare, e cioè muovere, queste camere all'interno della scena come qualsiasi altro oggetto presente all'interno di essa.

Come ogni altro elemento discusso fino ad ora, queste due fasi sono parti fondamentali della produzione, argomenti con così tante tecnicità e sfaccettature che spesso negli studi anche di medie dimensioni sono necessarie diverse figure specializzate nelle varie sottocategorie di rendering e animazione per poter gestire la mole di lavoro e tecnicismi necessari al portare a termine in maniera efficace queste fasi. Essendo però la nostra tesi uno specchio sulla realtà di un possibile libero professionista o addirittura qualcuno che si è appena approcciato al mondo della modellazione 3D, affronteremo questi step in maniera basilare, concentrandoci sulla teoria base e sulle metodologie che più facilmente vengono utilizzate.

E proprio in questo momento che il lavoro di progettazione dello storyboard che abbiamo affrontato nel capitolo 4.4 porta i suoi frutti: avendo già una mappa mentale delle posizioni della camera e di come approcciare il "filmato" possiamo usare come base ciò che abbiamo disegnato nello storyboard e poi nel caso modificarlo in base a ciò che il nostro senso estetico ritiene appropriato.

Prima di inserire le camere però, è doveroso introdurre della teoria su ciò che rappresenta la chiave di volta dell'animazione: i keyframe.

6.2 I keyframe I keyframe (dall' inglese, frame chiave) sono lo strumento principale sui cui è basato tutto il procedimento di animazione all'interno del nostro programma.

Dal manuale online di Blender (https://docs. blender.org/manual/en/latest/animation/ keyframes/introduction.html) i keyframe sono definiti come: "a marker of time which stores the value of a property"

Possiamo quindi pensare i keyframe come dei "punti fermi" che Blender usa come riferimento nello stabilire come si comporta un determinato valore. Se noi, ad esempio, prendiamo un intervallo di 60 frame in cui, nel frame numero 1, associamo alla posizione di un cubo un keyframe in una posizione A, mentre nel frame 60 in una posizione C, Blender calcolerà (più tecnicamente interpolerà, e cioè definirà in base a una curva definita dall'utente) la posizione di questo cubo nei frame dal 2 al 59; se decidessimo di inserire un nuovo keyframe in una posizione non precedentemente prevista dal programma in un frame intermediario Blender ricalcolerà un percorso che comprenda il keyframe che abbiamo inserito.

Nota bene: abbiamo fatto un esempio utilizzando la posizione come valore variabile, ma quasi ogni valore può essere associato a un keyframe e

6.3 Inserimento camere: animazione su curve

ලැ v Camera.002		\bigcirc
∨ Lens		
Туре	Perspective	•
Focal Length		•
Lens Unit	Millimeters	•
Shift X		•
		•
Clip Start		•
End		•
> 🗹 Depth of Field		
> Camera	:	Ξ
> Safe Areas	:	Ξ
> Background Images		
> Viewport Display		::::
> Camera Shakify		::::
> Custom Properties		

Figura 6.1

quindi "animato", gli esempi più comuni possono essere scala, forma, colore, trasparenza, densità ma anche ogni singolo valore dei nodi che abbiamo visto nei capitoli precedenti può essere associato a dei keyframe, le opzioni sono quindi all'atto pratico infinite.

Ora che ci è più chiara la teoria di base dell'animazione, possiamo inserire delle camere. Come è intuibile, essendo i parametri delle camere in tutto comparabili a quelli di una telecamera reale (è infatti possibile anche utilizzare dei preset di singole camere o singoli obbiettivi, tale è il livello di realismo) le decisioni che vanno prese sono numerose a tal punto da richiedere una figura professionale dedicata e la collaborazione di un eventuale regista o art director, fortunatamente per un utilizzo come il nostro sarà sufficiente utilizzare i valori di default e modificare solo due valori: la lunghezza focale (responsabile dell'angolo di campo, e cioè quanto di ciò che effettivamente stiamo inquadrando riusciremo a vedere attraverso l'obbiettivo) e l ' F-stop (responsabile della messa a fuoco). [vedi figura 6.1]

Mentre l'F-stop lo vedremo a breve è buona norma già definire una lunghezza focale corretta: questo parametro è fondamentale perché in base a ciò che scegliamo cambierà drasticamente il linguaggio del nostro prodotto e, a livello più pratico, la posizione corretta della camera. Nel nostro caso useremo una lunghezza focale di 35 mm, questa è infatti considerata un po' un "tuttofare" nell'ambiente fotografico ma è anche particolarmente indicata per la street photography, e cioè il caso del nostro filmato introduttivo.

Inserire una camera all'interno di Blender è molto semplice: basta infatti premere shift+A ed aggiungerla come qualsiasi altro elemento fatto fino ad ora, prendiamo poi il valore focal length dal pannello delle proprietà e inseriamo 35 mm. Questa camera al momento non è molto utile, è infatti ferma e puntata verso il basso, per risolvere questi due problemi dobbiamo affrontarli separatamente.

Esistono molti metodi per gestire le camere all'interno di Blender, noi ne useremo uno un po' più avanzato del semplice traslare, ruotare ed associare keyframe ai movimenti.

~	Follow Path						0	××	::::
		Target	🔲 Bez	ierCurve.	002			×	
		Offset							•
		Forward Axis		Y					•
		Up Axis					Z	Z	•
			Fixed	Position					•
			Curve	e Radius					•
			Follow	w Curve					•
2,				mate Pat					
		1-0			1.00	0			
		Influence			1.00	0		8	•

Figura 6.2

~ ,•	Track To			○ ∨ X	::::
	Target	Empty.004		×	
	Track Axis			Y Z	
	Up		Y		
		Target Z			
	Target	World Space			· •
	Owner	World Space			•
	Influence		1.000	6	•
		Figure C	2		

Figura 6.3

Questo approccio richiede infatti un po' più di preparazione ma è molto più versatile e porta a un risultato molto più realistico: l'animazione su curve. In Blender è infatti possibile inserire, tra le molte forme bidimensionali e tridimensionali, anche delle curve, quest'ultime però, a differenza delle altre forme che abbiamo utilizzato precedentemente, hanno una caratteristica fondamentale: possiedono una direzione. È possibile quindi usarle come dei "binari" a cui agganciare la nostra telecamera e muoverla all'interno dello spazio tridimensionale modificando solo un paio di valori facilmente accessibili.

Per fare ciò aggiungiamo una curva che posizioniamo in maniera consona al percorso che desideriamo la nostra camera segua ed aggiungiamo alla camera un constraint "follow path", la camera sarà quindi obbligata a seguire la curva e il valore "offset", a cui associamo i keyframe desiderati, ci permetterà di modificarne la posizione all'interno della stessa. [vedi figura 6.2]

Otteniamo quindi una videocamera in movimento, ma che ancora punta verso il basso. Entra qui in gioco un altro modificatore da associare alla camera: il "track to". Come dice il nome stesso questa funzione ci permette di chiedere alla camera di seguire con l'inquadratura un oggetto in particolare, di cui noi poi andremo ad animare la posizione sempre con una curva. [vedi figura 6.3]

L'oggetto in questione è un altro elemento particolare, denominato "empty": questo tipo di oggetto è infatti solo la rappresentazione grafica di una coordinata. In questo modo potremo usarlo come "gancio" senza che appaia sempre al centro della inquadratura.

Ora che abbiamo agganciato la visuale ad un oggetto andiamo nelle opzioni della camera e attiviamo la spunta su "depth of field", questa opzione ci permette di scegliere un oggetto che la telecamera terrà sempre a fuoco, a discapito del resto (nota bene: se questa opzione non viene attivata, la camera metterà a fuoco l'intera scena, diminuendo in maniera considerevole il senso di realismo). Selezioniamo quindi come oggetto da tenere a fuoco il nostro empty (che posizioneremo nella scena anche tenendo in considerazione che sarà la distanza a cui la nostra videocamera mantiene il focus) e inseriamo poi un valore di O.9 (più il valore sarà basso, più velocemente gli oggetti lontani dall'empty andranno fuori fuoco) Il risultato che otteniamo ci permetterebbe già da ora di fare il rendering del nostro prodotto finale, ottenendo però un video forzato, poco organico e non immersivo. Questo problema nasce dal fatto che il nostro video vuole porsi come una narrazione in prima persona, ma il movimento di camera è troppo stabile, mancano infatti delle piccole imperfezioni che ci permettano di ricreare le sensazioni di un video fatto con il proprio cellulare, se non direttamente ciò che vediamo con i nostri stessi occhi.

Come abbiamo già detto precedentemente Blender considera ogni animazione come una variazione di valori nel tempo definita da una funzione, e che viene rappresentata graficamente in una curva di tale funzione in un piano cartesiano. Questi grafici sono però anch'essi modificabili attraverso i modificatori, dato che abbiamo detto il problema essere l'eccessiva stabilità dei movimenti della camera la soluzione è molto semplice: andremo a "sporcare" il grafico della rotazione e della posizione dell'empty (che oltre a decidere la direzione in cui punta la videocamera, è anche responsabile della rotazione dell'inquadratura) creando dei micromovimenti che rendano il video più immersivo. Il procedimento è estremamente semplice, basta infatti aprire il pannello del graph editor e selezionare i valori dell'empty che ci interessano (nel nostro caso la rotazione sull' asse Y e la posizione su asse X e Z) e inserire il modificatore "noise" [vedi figura 6.4]. Il "noise" (dall'inglese "rumore") permette infatti di creare una variazione casuale di un dato valore di una curva, seguendo però la tendenza della curva stessa. Se noi quindi applichiamo questo modificatore possiamo diminuirne l'intensità (i valori massimi e minimi che gli è concesso raggiungere) e la scala (quanto rapidamente suddette variazioni avvengono) ottenendo un filmato decisamente più realistico ed avvincente in poco tempo.

Ci stiamo addentrando nelle fasi finali del processo di produzione, abbiamo infatti una scena completa e una camera propriamente impostata. C'è però ancora un modo per aumentare il livello di realismo della scena: animare l'illuminazione.

Replace ~
130.000
0.800
0.000
1.000
0
1.000
1.000
480.000
0.000
24.000

Figura 6.4

<u>59</u>

Come ben sappiamo nel mondo reale, soprattutto in città, le fonti luminose sono innumerevoli, mutabili e spesso anche in movimento. Dato che ci troviamo in una ipotetica città del futuro è consono aggiungere dei dettagli al nostro sistema di illuminazione, in questo capitolo vedremo quindi come aggiungere delle macchine, i cui fari illumineranno brevemente tutta la scena, e dello sfarfallio in alcuni dei led che abbiamo inserito precedentemente. Entrambi sono procedimenti molto semplici e alla portata di tutti, per raggiungere gli effetti desiderati useremo infatti due strumenti che già precedentemente abbiamo utilizzato: i keyframe e i modificatori delle curve di valori. Partendo dai fari delle macchine aggiungiamo tre luci fuori dalla camera, scegliendo colore e intensità consoni a quelli che potremmo trovare nel mondo reale, basterà poi scegliere degli intervalli di tempo in cui desideriamo queste luci si "animino" ed applicare un modificatore noise (di cui modificheremo scala ed intensità in base a ciò che la nostra intuizione ritiene come realistico) alla posizione sull'asse Z in modo da simulare le asperità del terreno. Per quanto riguarda lo sfarfallio delle luci, il procedimento è ancora più semplice, non cambiando la posizione, andremo semplicemente a inserire un numero arbitrario di keyframe di vari valori di intensità variabile nel nodo emission (nota bene: volendo è possibile usare direttamente due keyframe dello stesso valore, ma il risultato sarebbe meno realistico) e ad inserire nuovamente un modificatore noise (anch'esso propriamente modificato in scala ed intensità). Ora che abbiamo ufficialmente concluso la fase di realizzazione e modifica del file si ouò passare
all'ultimo step della produzione: il rendering.
conversione dall'ambiente 3D ad un media 2D, che esso sia foto o video, fruibile al di fuori del programma di modellazione. Per fare ciò è necessario un render engine, un software quindi che calcoli per ogni dato frame ciò che la camera inquadra e lo trasporti in un'immagine. Blender prevede tre engine differenti:

Blender prevede tre engine differenti: -cycles: basato sul path tracing (una simulazione fisicamente accurata del comportamento della luce in relazione alla posizione della camera), estremamente realistico e cinematografico ma molto più pesante a livello di calcolo, principalmente usato per produzioni cinematografiche ad alto budget -eevee: basato sul rendering in tempo reale, più complicato di Cycles se usato per raggiungere effetti realistici ma molto più leggero -workbench: è l'engine usato nell'area di lavoro di Blender, nonostante sia selezionabile non è inteso da usare come motore finale, solo a riprodurre in un'immagine ciò che vediamo all'interno del software

Nella realizzazione della nostra introduzione allo showreel useremo Eevee, essendo quello più a portata di un utente medio dato che richiede potenze di calcolo decisamente minori. Per quanto Eevee permetta di risparmiare numerose risorse per poter raggiungere livelli di realismo accettabili è necessario un po' di lavoro preparativo in più: dobbiamo infatti preparare degli oggetti di supporto denominati "light probes".

Questi oggetti, non visibili nel render finale, permettono a Eevee di calcolare in maniera più accurata e veloce (dopo uno step di registrazione dell'illuminazione, detto "baking") vari tipi di illuminazione in base al light probe usato, ne esistono infatti di tre tipi:

-Irradiance volumes: responsabile della memorizzazione dell'illuminazione indiretta diffusa dalla scena, la porzione di luce quindi non assorbita dal materiale e riflessa all'interno della scena ad angoli differenti da quelli del raggio di luce incidente alla superfice

-Reflection Cubemap: responsabile della memorizzazione dell'illuminazione indiretta riflessa della scena

-Reflection Planes: usato solo in caso di riflessioni chiarissime come specchi o acqua all'interno della scena, essendo un piano che riflette l'intera scena e la costruisce con la camera capovolta (proprio per simulare una riflessione quasi totale) è molto pesante e incide fortemente sui tempi di rendering.

Nel nostro caso essendo un ambiente molto buio e particolarmente privo di superfici riflettenti andremo ad usare uno solo di questi volumi, e cioè l'irradiance volume, che ci permetterà di ottenere un risultato molto più realistico senza intaccare eccessivamente i tempi di rendering. Posizioneremo quindi un irradiance volume che contenga tutta la scena e attiveremo il baking, questo processo è molto pesante e può consumare del tempo, che però guadagneremo in tempo di rendering successivamente e ci permetterà di ottenere un risultato decisamente più realistico.

Oltre all'utilizzo di questi volumi per poter usare al meglio questo render engine dovremo modificare alcuni dei valori di default, più precisamente attivare delle opzioni che aumenteranno il tempo di calcolo ma ci permetteranno di ottenere un risultato estremamente più realistico. Vediamo quali sono e di cosa si occupano:

-ambient occlusion: simulazione dell'attenuazione luminosa nel caso in cui due oggetti siano molto vicini tra di loro

-bloom: simulazione di come reagisce una camera reale a luci molto forti, creando una "corona" di pixel illuminati attorno alle fonti di luce particolarmente intense che illuminano nella direzione di ripresa

-screen space reflection: effetto simile al reflection plane che abbiamo visto precedentemente, ma più preciso, realistico e localizzato al singolo oggetto riflettente -motion blur: simulazione dell'effetto di perdita di nitidezza dovuta ai tempi di esposizione della videocamera, creando quindi un effetto di sfocatura, di intensità modificabile, sugli oggetti in movimento rispetto alla camera. [vedi figura 6.5]

Siamo quindi giunti alle fasi finali della fase di produzione, prima però di fare il rendering della scena è necessario definire i parametri del video che ne risulterà.

Lasceremo la definizione standard di 1920x1080 pixel poiché il video sarà principalmente fruito attraverso schermi in full HD, useremo un framerate (cioè da quante immagini sarà formato un secondo di video) da 24 frame al secondo essendo lo standard cinematico e il livello di fluidità necessario perché il cervello umano riconosca un oggetto come in movimento. Usare una definizione o framerate maggiore è possibile ma impatterebbe in maniera direttamente proporzionale i tempi di rendering, poiché dovremo effettivamente calcolare la stessa quantità di dati moltiplicata per l'aumento che

6	notte		52
	Render Engine	evee	~
	Sampling		
	Ambient Occlusion		
	🛃 Bloom		
	Depth of Field		::::
	Subsurface Scattering		
	Screen Space Reflections		::::
	🗹 Motion Blur		
	Volumetrics		
	Performance		
	Curves		
	Shadows		
	Indirect Lighting		
	Film		
	Simplify		
	Grease Pencil		
	Freestyle		::::
	Color Management		

Figura 6.5

abbiamo definito. Per esempio, se decidessimo di avere un output di 30 frame al secondo dovremmo calcolare il 20 % in più di frame, mentre se decidessimo di scalare la risoluzione al 4K (3840x2160) Blender dovrebbe calcolare quattro volte la quantità di pixel presenti nel full HD (1920x1080), rendendo in tempi di rendering estremamente più lunghi.

Per quanto riguarda il tipo di output che ci fornirà Blender non faremo il rendering di un unico video ma faremo il singolo render di ogni frame che poi metteremo insieme successivamente nella fase di post-produzione.

Questo approccio ci permetterà di agire sul singolo frame senza dover rinunciare a tutto il rendering fatto nel resto del video e in caso di crash del sistema avremo già ottenuto una parte dei frame necessari nel disco rigido del nostro computer; in caso di errore potremo quindi riprendere da dove ci siamo fermati, se invece facessimo direttamente il rendering di un video perderemmo il file e dovremmo rifare da capo. Una volta premuto CTRL+F12 il nostro computer inizierà il rendering frame by frame ed avremo concluso la fase di produzione.

Post - produzione

7.1 Raccolta materiale

Volendo essere questa tesi uno specchio della comunità attorno a Blender e al mondo dell'open software è consono raccontarne uno dei vantaggi meno quantificabili ma sicuramente più impattanti nell'utilizzo di questo software: i rapporti tra persone.

Per quanto la stesura di una tesi universitaria sia un lavoro che richiede molto impegno e dedizione, tutti gli studenti che hanno affrontato in passato questo percorso sanno quanto la parte in assoluto più difficoltosa è anche quella che meno è sotto il proprio controllo: la raccolta di dati. Che siano essi sotto forma di interviste o questionari, anche compilabili in pochi minuti se non addirittura in meno di un minuto, convincere degli estranei a dedicare del tempo a un'altra persona può essere un'esperienza profondamente sconfortante.

Ed è proprio in queste situazioni che si può toccare con mano l'importanza di una comunità così ampia, dedicata e generosa come quella di Blender (simile in questi aspetti a tante altre comunità di free-software, ma sicuramente tra le più accoglienti). A testimonianza di quanto chi utilizza questo software sia pronto ad aiutare chiunque, all'interno dell'arco di tempo dedicato esclusivamente a questa tesi – e cioè all'incirca 3 mesi- la fase di raccolta del materiale è durata un singolo pomeriggio.

In questo brevissimo arco di tempo è stato infatti concesso il permesso di utilizzare l'intera filmografia di Blender Studio e l'accesso ai file di progetto di vari professionisti del settore da ogni parte del mondo.

Oltre a quanto detto va anche considerato l'accesso gratuito e a portata di tutti a numerosissimi materiali, guide e file di progetto che si sono rivelati fondamentali in primis per imparare ad utilizzare questo software, ma anche per la realizzazione della intro che abbiamo analizzato in questa tesi.

7.2 Compositing Nonostante la fase di effettiva progettazione della nostra introduzione si possa dire conclusa, c'è un passaggio che non abbiamo ancora considerato: il compositing.

Essendo il nostro progetto studiato in un ambiente tridimensionale è fisiologico che nella trasposizione da 3D a 2D si possa avere delle perdite di qualità dovute a moltissimi fattori, che siano inesperienza, errori di progettazione o semplicemente scelte funzionanti nell'ambito 3D, ma meno efficaci una volta che si fruisce il prodotto finito. Portando quindi a un risultato al di sotto delle aspettative.

Così come nella fotografia la qualità delle foto può essere migliorata da una modifica post-scatto, nella produzione media 3D possiamo migliorare la qualità del nostro render con delle modifiche postume: questa fase è detta di compositing.

Il termine compositing sta ad indicare l'idea di "comporre" più immagini e fonti all'interno di una singola immagine (nota bene: mentre l'editing prevede di mettere "in coda" più elementi ma distribuiti nel tempo, il compositing prevede di unirli negli stessi intervalli di frame).

Questo passaggio avviene nel compositor di Blender, una finestra che prenderà ogni frame della nostra animazione e, attraverso la metodologia dei nodi che già abbiamo discusso, ci permetterà di modificarla a nostro piacimento.

Nella moltitudine di modifiche possibili (come ogni altro aspetto visto fino ad ora, la figura del compositor è un professionista a sé stante, spesso anche non pratico di 3D ma con una formazione artistica a parte) noi faremo delle modifiche molto basilari: aumenteremo infatti un po' il contrasto, inseriremo una leggera distorsione e inseriremo una leggerissima grana, che seppur quasi invisibile cambierà profondamente l'impatto dei nostri frame e quindi del nostro video generale.

Il procedimento che useremo è molto semplice: useremo come input il frame che abbiamo appena renderizzato e vi applicheremo vari filtri sotto forma di nodi.

Per poter aumentare il contrasto di un'immagine e renderla più piacevole alla vista diminuiremo la luminosità delle ombre (e cioè "aumenteremo i livelli di nero", nonostante questa espressione sia più intuibile non è tecnicamente corretta, essendo il buio la sola mancanza di luce infatti un colore meno luminoso è per forza di cose più scuro). Prenderemo quindi il nodo della nostra immagine e collegheremo al polo "shadow", che gestisce le ombre, ed "AO", responsabile dell'ambient occlusion, due copie di uno dei nodi più utilizzati nel workflow 3D: il colorRamp. Questo nodo prevede infatti uno slider che permette di gestire la quantità di dati colori all'interno di un dato nodo, oggetto o materiale. Essendo molte delle informazioni usate da Blender sotto forma di bianco e nero (che rispettivamente rappresentano l'1 e lo 0 del codice binario) questo nodo permette di modificare in maniera estremamente precisa numerosissime proprietà di altrettanti numerosi nodi, nel caso delle ombre la loro luminosità: più sarà presente il nero all'interno dello slider meno le ombre saranno luminose. Per quanto riguarda la grana e la distorsione, entrambi effetti usati per simulare la ripresa di una camera (il primo simula il rumore di fondo dovuto alla zona poco luminosa, il secondo una lieve distorsione a bordo dell'immagine dovuta alla concavità delle lenti della camera) si possono applicare in maniera molto semplice: il nodo per la distorsione è esclusivo solo per questa funzione e prende quindi il nome "lens distorsion", per la grana invece inseriremo sopra l'immagine una texture di rumore estremamente opacizzata in modo da risultare quasi impercettibile.

Se volessimo però comparare i risultati di questo processo al frame non lavorato non troveremo differenze, questo è dovuto al fatto che i nodi che abbiamo collegato all'immagine input devono poi essere collegati anche all'output sovrapponendoli all'immagine, in modo analogo a come funzionano i filtri di un qualsiasi telefono, per fare ciò useremo un altro nodo di nome "overlay", come il nome descrive grazie ad esso potremo sovrapporre i vari filtri, ne useremo quindi uno per ogni modifica che abbiamo discusso, creando una catena di nodi.

Due dettagli che meritano una certa attenzione: il nodo lens distorsion non è accoppiato a un nodo overlay ma è direttamente inserito nella catena che parte dall'input all'output, questo perché non è un'immagine come gli altri nodi (se infatti li inserissimo al posto degli overlay, sostituirebbero direttamente l'immagine invece di sovrapporsi) ma modifica direttamente l'immagine di base, creando appunto la distorsione.

Altro dettaglio più tecnico, e perciò ancora più degno di attenzione, risiede nel fatto che in questo caso l'ordine dei nodi non modificherebbe il risultato, a differenza dei materiali, infatti, in cui tutti i nodi tendono ad interagire strettamente tra di loro in questo caso stiamo semplicemente sovrapponendo vari filtri ed immagini per ottenere un risultato più piacevole, ogni interazione dei filtri è relegata al primo nodo di input in maniera separata alla catena principale. Bisogna quindi tenere a mente che questo è un caso isolato e non lo standard, in cui l'ordine dei nodi è fondamentale. Qui mostrata una comparazione dello stesso render prima (a sinistra) e dopo (a destra) la fase di compositing [vedi figura 7.1]



Figura 7.1

7.3 Editing, VFX, e motion graphics



Figura 7.2

Ora che abbiamo raccolto tutto il materiale necessario e la nostra introduzione non necessita ulteriori interventi possiamo unire tutto ciò che abbiamo raccolto in un unico filmato. Per fare ciò conviene aprire un nuovo file di Blender esclusivo per il video-editing, i file di montaggio video possono essere molto pesanti, e in generale tenere le due parti separate è una buona abitudine per evitare delle interferenze, soprattutto in fase di rendering. Selezioniamo quindi il layout dedicato al video editing e ci troviamo di fronte a questa schermata: [vedi figura 7.2]

Il primo passo consiste nell'importare tutti i file sia audio che video. Per quanto riguarda la nostra introduzione useremo la funzione "importa strisce di immagini", selezioneremo quindi tutti i nostri frame e Blender li unirà automaticamente in una striscia video, per quanto riguarda tutti gli altri elementi un singolo drag and drop dalla cartella del nostro computer sarà sufficiente, Blender in automatico inserirà i file e ne creerà ciò che viene definito come "proxy".

l proxy sono copie del file sorgente a qualità minore e vengono usati all'interno dell'editor per due principali vantaggi:

-renderli modificabili senza intaccare il file importato

-aumentare notevolmente le prestazioni dell'editor Una volta inseriti tutti i file che abbiamo deciso di utilizzare li esamineremo uno ad uno e con la funzione "trim" accessibile premendo il tasto K isoleremo alcuni frammenti ad alto impatto visivo da utilizzare nel video finale. Sposteremo poi tutti i file ritenuti più interessanti in un canale a parte e disattiveremo, premendo il tick a sinistra, i vari canali dedicati alle varie fonti video. Per quanto riguarda l'ultimo step preparatorio, seppur non fondamentale ma fortemente consigliato, selezioniamo le varie tracce audio e dal pannello a destra (in caso sia nascosto, renderlo visibile premendo N) andremo a

selezionare la spunta "display waveform", questo ci permetterà di vedere sopra la nostra striscia una rappresentazione grafica della traccia audio, molto utile per poter montare a tempo di musica i vari frammenti di video.

Ora che abbiamo raccolto e organizzato tutti i file attiviamo la funzione snap premendo il simbolo



Figura 7.3

della calamita in alto (questo ci permetterà di collegare le varie strip senza rischiare di saltare dei frame) e possiamo semplicemente spostare nell'ordine che desideriamo i vari frammenti video.

Come dettaglio aggiuntivo premiamo tasto destro del mouse su ognuna delle fonti audio e inseriamo grazie alla funzione omonima un "fade" (ovvero una sfocatura) in entrata (fade in) ed in uscita (fade out) in modo che il volume dell'audio non sia immediatamente al massimo ma aumenti in maniera graduale, rendendolo più piacevole. Otterremo alla fine la seguente schermata: [vedi figura 7.3]

Il nostro progetto sta quindi volgendo al termine: abbiamo progettato ed ottenuto il materiale necessario e lo abbiamo elaborato e riorganizzato in un unico video fruibile da qualsiasi dispositivo elettronico.

L'ultimo step necessario per completare il nostro elaborato e la fase di inserimento di grafiche aggiuntive ed effetti speciali, detti VFX (=visual effects)

Nonostante Blender comprenda gli strumenti necessari alla creazione di effetti speciali, fondamentali per produzioni ad alto o altissimo budget come quelle cinematografiche, nel nostro caso non ne aggiungeremo, gli effetti speciali sono infatti una branca del 3D estremamente complessa, ramificata e molto diversa dalla modellazione ed animazione 3D, tanto da non prevedere normalmente l'intervento di una sola figura professionale ma direttamente il coinvolgimento di studi cinematografici dedicati solo all'inserimento di effetti speciali. Caso simile, ma in intensità minore, è la motion graphics, termine che sta ad indicare l'animazione che ha come soggetto principale i testi, in altre parole lo sviluppo animato di una progettazione grafica, normalmente affidata alla figura del motion graphic designer. Nel nostro elaborato faremo un uso estremamente limitato del graphic design, poiché fuoriesce dall'argomento principale della tesi essendo principalmente un lavoro ideato per workflow bidimensionali, ma anche in questo caso la flessibilità di Blender permette, con le giuste attenzioni, di utilizzarlo anche per questo scopo. Inseriremo quindi semplicemente una piccola introduzione in cui il logo del Politecnico di Torino e di Blender appariranno a schermo, prima che il

filmato inizi.

Nota bene: prima di affrontare questa fase apriremo un nuovo file Blender, questo approccio di separare in vari file alcune fasi del processo è una mia preferenza personale, ma non è strettamente necessario né fortemente consigliato, in Blender è infatti possibile organizzare le cosiddette "scene" dal pannello in alto a sinistra, e cioè creare nuove istanze di Blender completamente vuote (o, volendo, una esatta copia di quella già costruita) completamente indipendenti l'una dall'altra. Entrambi gli approcci sono validi, creare più scene permette, ad esempio, di copiare e incollare materiali e oggetti da una scena all'altra in maniera più veloce e diretta che da un file ad un altro, mentre avere più file separati permette di avere file più leggeri (con un vantaggio di prestazioni in caso di macchine più deboli). Nel mio caso preferisco la seconda scelta non per necessità di potenza ma per una semplice preferenza personale, trovo infatti che usare più scene se non si fa attenzione può diventare molto confusionario in breve tempo.

Prima di iniziare il, molto semplice, procedimento che andiamo ad affrontare però è necessario attivare un'add-on già presente, ma non attivo, all'interno di Blender. Apriamo quindi il software, premiamo su edit e su preferences e nella finestra che ci si pone davanti andiamo nella sezione degli add-on, digitiamo quindi nella barra di ricerca "import images as planes" e lo attiviamo. Questo componente aggiuntivo ci permetterà di importare immagini sotto forma di piani già preparati, ed utilizzando i nodi che abbiamo visto fino ad ora è compatibile con le trasparenze dei file .png. Usare questo add-on ci permetterà di non dover importare file .svg (scalable vector graphics), dato che lavorare con suddetti file può rivelarsi a volte inutilmente complicato e nel nostro caso non necessario.

Possiamo quindi preparare la scena: impostiamo gli stessi parametri di render del nostro filmato (in particolar modo stessa definizione e stesso framerate), creiamo poi una camera che punti sull'asse Y. Una volta fatto ciò creiamo due semplici piani perpendicolari alla camera e che abbiamo una grandezza superiore a quella dei loghi che importeremo attraverso il comando images>image as plane, anch'essi perpendicolari alla camera. I piani serviranno a nascondere i loghi vicino al centro dell'immagine, dato che non vogliamo che appaiano in movimento a bordo schermo ma solo al centro.

Ci si pone davanti un problema: il materiale dei piani. Dato che Blender interpreta i materiali attraverso una simulazione delle regole della fisica, come possiamo creare un materiale completamente buio, solido e che non interagisca in nessun modo con la luce? La soluzione è molto semplice, scolleghiamo il materiale di default e non ne assegniamo nessuno ai piani, in questo modo Blender mostrerà nel risultato finale i piani, ma come due oggetti totalmente neri non riuscendo ad elaborare il "non-materiale" assegnatogli. Impostiamo una semplice luce che illumini la scena e inseriamo i keyframe desiderati, la nostra animazione è quindi pronta.

Una volta finito il render della nostra breve animazione basterà inserirla nei frame desiderati del progetto principale e il nostro progetto può dirsi ufficialmente concluso.

Conclusioni

Giunti alla fine di questo percorso è consono riassumere ciò che abbiamo affrontato e dimostrato.

Siamo partiti da un'analisi della storia di Blender e dell'ideologia del free software, da dove e grazie a chi è nata, a cosa ha portato e che cosa può dare in più della normale ideologia fornitore-consumatore.

Abbiamo scoperto quanto fondamentale sia il senso di comunità che nasce proprio da questa ideologia e abbiamo scoperto come Blender, con la giusta guida e il giusto approccio possa sì essere alla portata di tutti ma anche e soprattutto uno strumento professionale e completo, che porti a risultati degni dei competitor che seguono il modello classico. Ci siamo resi conto dell'infinite possibilità e metodologie diverse che permette Blender e abbiamo potuto toccare con mano il valore e la fondamentale importanza dell'approccio progettuale e l'importanza di una mentalità "da designer" nell'affrontare il non sapere cosa fare, come fare ciò che si sa e tutti i problemi che ne conseguono.

Questa tesi è sì la dimostrazione di quanto detto poco sopra, ma è una piccola, infinitesima, parte di ciò che può fare questo software, la quantità di sbocchi professionali che offre, la profonda interazione con industrie completamente diverse da loro, la possibilità modificare ogni singolo pixel del nostro lavoro e renderlo nostro. Tutto questo completamente gratuito.
Bibliografia e sitografia

Free Software, Free Society: Selected Essays of Richard M. Stallman Timing for Animation- john halas 3D Modelling and Visualisation of Heterogeneous Cell Membranes (Ghaffar, Mehmood, Biere, Niklas, Jäger, Daniel, Klein, Karsten, Schreiber, Falk, Kruse, Olaf, Sommer, Björn –2018) The comparison Between 3D studio Max and Blender Based on Software Qualities (Hendriyani, Yeka, Amrizal, Vania Amanda-2019) Design of Virtual Reality Application for Taharah Using 3D Blender (D. R. Anamisa, M. Yusuf, F. A. Mufarroha and N. Rohmah -2020) Creating 3D Game Art for the iPhone with Unity (McDermott, Wes - 2020) Determinants of Blender 3D (open source) usage in Malaysian animation industry (Mohamed A¬que Bin Mohamed Aziyen, Elsadig Musa Ahmed - 2020) Immersive 3D modeling with Blender and off-the-shelf hardware (Takala, Tuukka M., Makarainen, Meeri, Hamalainen, Perttu - 2013) La storia di Blender, disponibile da https://www.blender.org/about/history/ Intervista a Ton Roosendaal, di creative commons, disponibile a https://creativecommons. org/2010/10/27/ton-roosendaal-sintel-producer-and-head-of-blender-institute/ "Quick guide to gplv3", di Brett Smith, disponibile a https://www.gnu.org/licenses/quick-guidegplv3 Intervista a Ton Roosendaal (Blender), di Sabine Niederer, disponibile da https://vimeo. com/3836064 "Novell Wins Again – Jury Rules Copyrights didn't go to SCO!", disponible a http://www.groklaw. net/article.php?story=20100330152829622 "GNU, general Public License", di Katie Terrell Hanna disponibile a https://www.techtarget.com/ searchdatacenter/definition/GNU-General-Public-License-GNU-GPL-or-simply-GPL "Copyleft: idealism pragmatic", di Richard Stallman, disponibile a https://www.gnu.org/philosophy/pragmatic.it.html "Blender Open movies", disponibile a https://www.blender.org/about/studio/ "Sprite Fright", disponibile a https://www.youtube.com/watch?v=UKQiQvAYoNU "Coffee Run", disponibile a https://www.youtube.com/watch?v=PVGeM4OdABA "Spring", disponibile a https://www.youtube.com/watch?v=WhWc3b3KhnY "Cosmos Laundromat", disponibile a https://www.youtube.com/watch?v=Y-rmzhOPI3c "Tears of steel", disponibile a https://www.youtube.com/watch?v=R6MlUcmOul8 "Spoon", by Artur Chays, disponibile a https://www.youtube.com/watch?v=wXhGd3lu2sO "Hero", disponibile a https://www.youtube.com/watch?v=pKmSdY56VtY

Ringraziamenti

E siamo arrivati alla parte preferita di noi studenti, i ringraziamenti. Molti dei miei amici hanno iniziato il lavoro di tesi proprio con i ringraziamenti, per darsi la giusta motivazione, io invece ho preferito tenermeli alla fine, così me li godo di più.

Sarà mainstream, sarà scontato, ma per me non lo è mai, prima di chiunque altro vorrei ringraziare mia Mamma Barbara (la M maiuscola è voluta e necessaria), l'arco che mi ha scoccato e che mi protegge sempre e comunque, le parole non possono descrivere quanto ti sia grato ogni secondo della mia vita e quanto io ti ammiri nella tua forza, non posso dilungarmi a dirti quanto ti amo perché se no so già che finisce in lacrime e la tesi si bagna!

Un grazie a tutta la mia famiglia: Valeria, che mi ispira ad essere la versione migliore possibile di me stesso, Massimo, il mio compagno di nerdate e battute e non un fratello minore ma alla pari, Stella, la mia compagna d'arte e la prima sorella che ho potuto coccolare fin da appena nata, Nicola, il mio orgoglio di fratello maggiore e senza ombra di dubbio la persona più interessante che io conosca, Elena, la più piccola della famiglia ma sicuramente il mio amore più grande, nonno Mimmo, per tutte le pizze offerte e mangiate in compagnia, ed Alessandro, membro acquisito, che ha il merito più faticoso, calzare le scarpe di padre ad honorem.

Vorrei ringraziare Alice, la mia ragazza (anche se definirti solo così è riduttivo), anche per te servirebbe una tesi intera per dirti quanto ti amo e quanto il tuo aiuto sia stato fondamentale non solo per la tesi, ma per tutto ciò che ho affrontato insieme a te negli ultimi sette anni. Grazie per tutti i meravigliosi momenti che abbiamo passato insieme, grazie per esserci sempre, per avermi fatto crescere, per conoscermi come nessuno, per spronarmi e consolarmi nei momenti più bui, grazie per tutti i pranzi offerti, per tutti i vizi che mi concedi, per la tua infinita pazienza e grazie semplicemente di essere la mia compagna...aishiteru.

Grazie a Silvia, la persona che più mi ha ispirato in questi anni di università, che più mi ha sopportato quando non avevo voglia di fare niente e quando non sapevo fare niente. Che mi ha dimostrato quanto importante sia l'intraprendenza e che solo l'essere diventato suo amico ha più valore di qualsiasi altra cosa mi abbia insegnato questa università. Silvia sei destinata a grandissime cose e nonostante ora non riesca più a respirare senza pensare che faccio troppo rumore sono orgoglioso di essere tuo amico e non vedo l'ora di vedere cosa combinerai in futuro.

Grazie al mio relatore, Fabrizio Valpreda, per aver creduto in me e al mio progetto quando nessun'altro era disposto a darmi man forte, che nonostante tutte le difficoltà che questo progetto ha portato ad affrontare è rimasto al mio fianco rispondendo alle mie miriadi (davvero miriadi) di mail in pochissimo tempo, cosa che non è per niente scontata.

Grazie a Nicola e Daniele, i miei due compari di mangiate e di ca... va beh c'è la rima sapete completare, i miei due "bro", grazie perché con voi posso essere me stesso senza aver paura di essere giudicato, vorrei avervi conosciuto molti anni prima perché siete speciali.

Un altro grazie speciale a Ton Roosendaal, per aver creato questo meraviglioso software che mi sta cambiando la vita. Grazie ai miei amici vicini e lontani, grazie a Sacchy per tutte le serate e i gossip insieme, a Federica con cui posso parlare di one piece tutto il tempo che mi pare, a Paolo che anche se non ci sentiamo mai so che rimane come un fratello, grazie a Riccardo (non io) per avermi preso come tirocinante nel momento più disperato della mia carriera universitaria, grazie ad Arianna e Tommaso, i veri "og" come dicono gli americans, grazie a Sofia e a tutto il gruppo del discord, tenervi sul groppone giocando ad Overwatch mi ha reso forte e sano e lo devo a voi, grazie a Luca, il mio coinquilino e compagno di torte, grazie a Eleonora, compagna di classe ad Hogwarts, e infine grazie a Silente per i punti a Grifondoro completamente a caso, anche se avrei preferito dei CFU, grazie a POWERPIZZAPODCASK111222333 (il podcast migliore del mondo secondo Mattarella) che mi ha fatto ridere nelle infinite giornate dedicate a questa tesi, grazie a Giulia (Sulia per gli amici) per avermi intrattenuto con le sue live di Eurotruck simulator, grazie alla Riot Games, per avermi fatto capire che alla fine fare la tesi non è così male, poteva andare peggio, potevo giocare a League Of Legends.

Jungle Diff- cioè volevo dire Ad Maiora Si dice così giusto? Giusto??? Va beh.

Riccardo